POLITECHNIKA ŚLĄSKA

Wydział Automatyki, Elektroniki
i Informatyki

ROZPRAWA DOKTORSKA

mgr inż. Jakub Hyla

Efektywne kodowanie korekcyjne dla systemów transmisji w Internecie rzeczy

Promotor
dr hab. inż. Wojciech Sułek, prof. PŚ

Gliwice, 2024

Przedmowa

Rozprawa doktorska ma formę spójnego tematycznie cyklu publikacji, uzupełnionego o poszerzone streszczenie. W tekście streszczenia zastosowano odwołania do artykułów z wykorzystaniem następującej numeracji:

- **Publikacja 1.** W. Sułek i J. Hyla, „Aplikacje kodów korekcyjnych LDPC we współczesnych systemach radiokomunikacyjnych", w Elektronika, telekomunikacja, mobilność, t. 870, J. Izydorczyk, Red. 2020, s. 179–189.

- **Publikacja 2.** J. Hyla, W. Sułek, W. Izydorczyk, L. Dziczkowski, i W. Filipowski, „Efficient LDPC Encoder Design for IoT-Type Devices", Applied Sciences-Basel, t. 12, Art. nr 5, 2022, doi: 10.3390/app12052558.

- **Publikacja 3.** J. Hyla i W. Sułek, „Dekoder LDPC implementowany w mikrokontrolerze dla systemów Internetu rzeczy", Przegląd Elektrotechniczny, t. 99, Art. nr 4, 2023, doi: 10.15199/48.2023.04.23.

- **Publikacja 4.** J. Hyla i W. Sułek, „Energy-efficient Raptor-like LDPC coding scheme design and implementation for IoT communication systems", Energies, t. 16, Art. nr 12, 2023, doi: 10.3390/en16124697.

- **Publikacja 5.** J. Hyla i W. Sułek, „Niebinarne kodowanie LDPC dla systemów Internetu rzeczy", Przegląd Elektrotechniczny, Art. nr 4, 2024, doi: 10.15199/48.2024.03.44.

- **Publikacja 6** w trakcie recenzji (ID: Access-2024-04215). J. Hyla i W. Sułek, „Short Blocklength Nonbinary Raptor-Like LDPC Coding Systems Design and Simulation", IEEE Access.

Spis treści

1 Streszczenie

Rola Internetu rzeczy (IoT) w obecnym życiu społecznym i gospodarczym jest niezwykle istotna i prawdopodobnie będzie nadal wzrastać w przyszłości. Badania nad protokołami transmisji, włączając w to kodowanie korekcyjne, mają duże znaczenie dla efektywnego funkcjonowania Internetu rzeczy. Urządzenia IoT operują w różnorodnych warunkach środowiskowych i komunikują się przez różne typy sieci, korzystając z różnych protokołów, w szczególności w łączach do węzłów końcowych sieci, np. typu WSN (ang. *Wireless Sensor Network*). Efektywne protokoły transmisji, wraz z mechanizmami korekcji błędów, są niezbędne dla zapewnienia niezawodnej komunikacji, minimalizacji opóźnień oraz optymalizacji zużycia energii.

Prace badawcze przeprowadzone w ramach rozprawy skupiały się na analizie oraz ocenie wydajności zaawansowanego kodowania korekcyjnego, w szczególności kilku wariantów kodowania LDPC (Low-Density Parity-Check), w kontekście ich zastosowania w protokołach komunikacyjnych dedykowanych dla systemów Internetu rzeczy. Kodowanie LDPC, ze względu na swoje możliwości korekcyjne, jest uznawane za jedną z najbardziej efektywnych znanych i stosowanych metod. Badania obejmowały strukturę i parametry tych kodów, metody konstrukcji, kodowania, i implementacji w układach stanowiących bazę typowych systemów wbudowanych urządzeń IoT.

W ramach prac badawczych opracowano efektywne energetycznie rozwiązania algorytmów kodowania i dekodowania LDPC, zoptymalizowane pod kątem implementacji w mikrokontrolerach. Istotnym aspektem było uwzględnienie ograniczonych zasobów obliczeniowych i pamięciowych charakterystycznych dla sprzętu w urządzeniach, zwłaszcza tych związanych z Internetem rzeczy. Uzyskane wyniki wskazują, że kody LDPC, typowo stosowane w zaawansowanych systemach z implementacją sprzętową w dedykowanych układach cyfrowych, mogą znaleźć zastosowanie także w urządzeniach o silnie ograniczonych zasobach, szczególnie przy transmisji w kierunku w górę (ang. *uplink*). Mogą przynieść istotne korzyści w zakresie efektywności energetycznej oraz skuteczności korekcji błędów, w porównaniu z tradycyjnymi kodami blokowymi.

Następnie starano się wskazać możliwości dodatkowego zwiększenia efektywności energetycznej poprzez zaproponowanie i implementację adaptacyjnego schematu kodowania. Analizowano zastosowanie adaptacyjnych strategii kodowania

dla kodów QC-LDPC. Obiecującym kierunkiem wydały się kody typu Raptor (ang. RL – *Raptor Like*). Badania obejmowały rozwój architektury systemu z elastyczną konstrukcją kodów RL QC-LDPC oraz implementację w urządzeniu IoT. Eksperymenty oceniły wydajność transmisji z urządzenia IoT do bramki, ze szczególnym uwzględnieniem efektywności energetycznej adaptacyjnego kodowania, z redundancją przyrostową. Wykazano, że krótkie bloki QC-RL-LDPC mogą osiągnąć wyższą efektywność energetyczną niż kody LDPC o stałej długości.

Kolejne prace badawcze ukierunkowano na zastosowania kodów niebinarnych (NB). Przeprowadzono analizę potencjalnego wykorzystania kodów NB-LDPC, wykonano implementacje koderów, zaproponowano oryginalne metody dekodowania kodów typu Raptor oraz opracowano algorytm tworzenia macierzy kontrolnych kodów NB QC-RL-LDPC. Podkreślono istotność wydajności implementacji algorytmu kodowania, sugerując, że skuteczne wdrożenie kodów niebinarnych może przynieść dodatkowe korzyści, a koder może być efektywnie zaimplementowany w układzie mikrokontrolera. Wnioski z przeprowadzonych badań wskazują na obiecujące perspektywy zastosowania kodów NB-LDPC, zarówno pod względem efektywności energetycznej, jak i ogólnej wydajności systemów komunikacyjnych. W ramach eksperymentów porównano zależności czasowe dla różnych kodów nad ciałami Galois (GF), co pozwoliło ocenić wpływ wyboru kodu na potencjalne zużycie energii. Określono możliwości korekcyjne i pokazano, że opracowane kody, wraz z algorytmem dekodowania, dają lepsze możliwości niż binarne kody standardu 5G, a jednocześnie jest możliwa efektywna implementacja kodera, która może być skutecznie wykorzystana w transmisji w łączu w górę.

2 Abstract

The role of the Internet of Things (IoT) in current social and economic life is extremely significant and is likely to continue growing in the future. Research on transmission protocols, including error correction coding, is crucial for the efficient operation of the Internet of Things. IoT devices operate in diverse environmental conditions and communicate through various types of networks, using different protocols, especially in links to end nodes of the network, e.g., Wireless Sensor Networks (WSNs). Effective transmission protocols, along with error correction mechanisms, are essential to ensure reliable communication, minimize delays, and optimize energy consumption.

The research conducted as part of the dissertation focused on the analysis and evaluation of the performance of advanced error correction coding, particularly several variants of Low-Density Parity-Check (LDPC) coding, in the context of their application in communication protocols dedicated to Internet of Things systems. LDPC coding, due to its error correction capabilities, is considered one of the most efficient methods known and used. The studies encompassed the structure and parameters of these codes, methods of construction, encoding, and implementation in systems forming the basis of typical embedded IoT devices.

As part of the research activities, energy-efficient solutions for LDPC encoding and decoding algorithms were developed, optimized for implementation in microcontrollers. An important aspect was to consider the limited computational and memory resources characteristic of hardware in devices, especially those associated with the Internet of Things. The obtained results indicate that LDPC codes, typically used in advanced systems with hardware implementation in dedicated digital circuits, can also be applied in devices with severely constrained resources, especially in the uplink direction of transmission. They can bring significant benefits in terms of energy efficiency and error correction effectiveness compared to traditional block codes.

Next, efforts were made to identify opportunities for further increasing energy efficiency by proposing and implementing an adaptive coding scheme. The application of adaptive encoding strategies for QC-LDPC codes was analyzed. Raptor-like (RL) codes emerged as a promising direction. The research included the development of a system architecture with a flexible design of RL QC-LDPC codes

and implementation in an IoT device. Experiments evaluated the transmission performance from the IoT device to the gateway, with a particular focus on the energy efficiency of adaptive coding with incremental redundancy. It was demonstrated that short blocks of QC-RL-LDPC codes can achieve higher energy efficiency than fixed-length LDPC codes.

Subsequent research efforts were directed towards the applications of non-binary (NB) codes. An analysis of the potential use of NB-LDPC codes was conducted, implementations of encoders were carried out, original decoding methods for Raptor-like codes were proposed, and an algorithm for creating check matrix of NB QC-RL-LDPC codes was developed. The importance of implementation efficiency of the encoding algorithm was emphasized, suggesting that effective deployment of non-binary codes could bring additional benefits, and the encoder could be efficiently implemented in a microcontroller circuit. The conclusions drawn from the conducted research indicate promising prospects for the application of NB-LDPC codes, both in terms of energy efficiency and overall performance of communication systems. Time dependencies for different codes over Galois fields (GF) were compared in experiments, allowing for the assessment of the impact of code selection on potential energy consumption. Correction capabilities were determined, and it was shown that the developed codes, along with the decoding algorithm, provide better opportunities than binary 5G standard codes, while still allowing for an efficient implementation of the encoder, which can be effectively utilized in uplink transmission.

3 Poszerzone streszczenie

3.1 Wprowadzenie

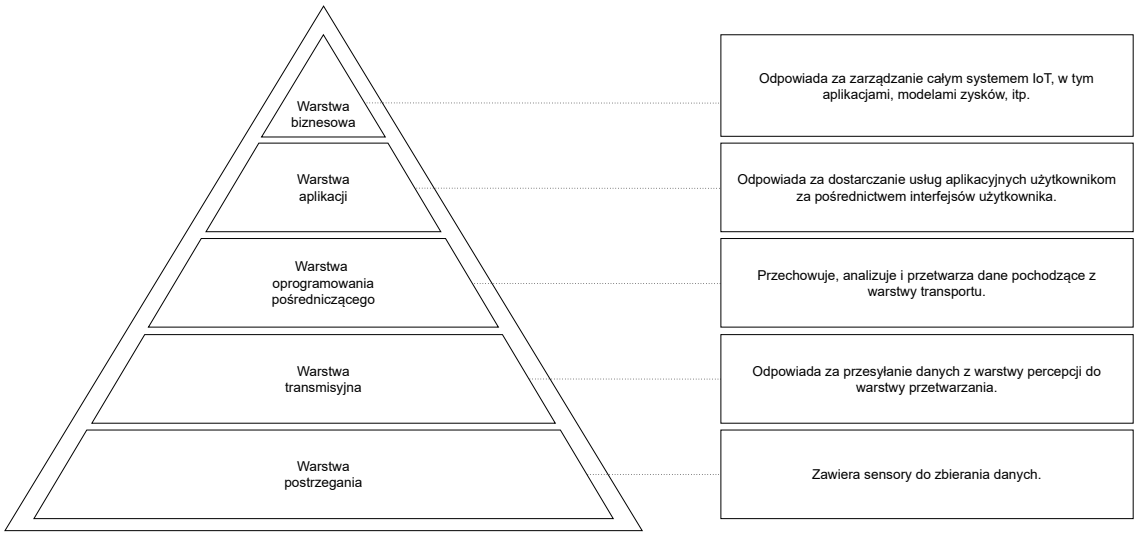
Niniejsza praca jest rezultatem badań zrealizowanych w ramach doktoratu wdrożeniowego, we współpracy z firmą TKH Technology Poland Sp. z.o.o., będącą członkiem grupy TKH, specjalizującą się w konstruowaniu urządzeń i systemów szeroko pojętego Internetu rzeczy (IoT). W prowadzonych w TKH Technology zaawansowanych projektach sieci IoT, jedną z kluczowych ról odgrywają łącza komunikacyjne, przy czym wykorzystywane są różnorodne rozwiązania i protokoły. Przeprowadzone prace badawcze są elementem zgłębiania wiedzy i potencjału związanego z innowacyjnymi rozwiązaniami w obszarze protokołów dla IoT, w kontekście szerokiej gamy urządzeń i złożonych projektów, rozwijanych przez firmę. Efekty prac zostały wdrożone w jednym z projektów, a ich opublikowanie wnosi wkład w rozwój wiedzy o możliwościach efektywnej korekcji kanałowej w urządzeniach posiadających specyficzne cech węzłów sieci IoT, implementujących kodowanie kanałowe.

Internet rzeczy (IoT), coraz bardziej ewoluujący w stronę Internetu inteligentnych rzeczy, jest istotnym elementem współczesnej technologii w krajobrazie biznesowym, ale także związanym z szeregiem aspektów badawczych. IoT to sieć obiektów fizycznych, wyposażonych w czujniki, elementy wykonawcze i oprogramowanie pozwalające na interakcję z otoczeniem, innymi obiektami i ludźmi, jak również układy umożliwiające łączenie się i wymianę danych z innymi węzłami za pośrednictwem infrastruktury sieciowej, w tym internetowej [5, 62]. Spektrum obiektów i urządzeń kategoryzowanych jako elementy IoT jest bardzo szerokie, jednakże wspólną cechą charakterystyczną jest obecność elektronicznego systemu wbudowanego, zwykle z centralnym elementem w postaci mikrokontrolera [86].

Stworzenie wszechobecnego środowiska (ekosystemu IoT), ze zróżnicowanymi obiektami świadczącymi inteligentne usługi, daje rozmaite korzyści dla ludzkości, w obszarach takich jak inteligentne domy [132] przemysł 4.0 [103] inteligente miasta [61] , zdrowie [90], rolnictwo [69] i wiele innych [101]. Rynek konsumencki wpływa na kształtowanie się nowego modelu biznesowego, z istotną rolą usług udostępnianych przez sieci typu IoT. Niezbędnym elementem postępu technologicznego umożliwiającego te przemiany jest skuteczna komunikacja pomiędzy

urządzeniami i systemami, wymagająca odpowiednio dobranych technologii i protokołów. Postęp w tej dziedzinie jest wynikiem ukierunkowania prac naukowych na ulepszanie technik transmisji danych. Wraz z upływem czasu, obserwuje się ciągły wzrost mocy obliczeniowej i ilości pamięci w urządzeniach elektronicznych. To zjawisko jest ściśle związane z Prawem Moore'a, które odzwierciedla stały progres możliwości układów scalonych i mikroprocesorowych [20, 129].

Urządzenia typu IoT są jednak w wielu rozwiązaniach oparte o energooszczędne mikrokontrolery, ze stosunkowo ograniczonymi zasobami, gdyż masowość wymaga ograniczania kosztów oraz zużycia energii [34, 106]. Wiele rozwiązań IoT wykorzystuje zasilanie bateryjne lub inne źródła zasilania o ograniczonej wydajności. Istotną kwestią w konstrukcji takich urządzeń jest minimalizacja zużycia energii poprzez zastosowanie efektywnych energetycznie rozwiązań, w celu np. przedłużenia żywotności baterii lub wydłużenia czasu pomiędzy ładowaniami lub wymianami baterii. Ten cel można osiągnąć nie tylko na etapie projektowania urządzeń wbudowanych, ale także na etapie projektowania algorytmów i ich implementacji w układach mikrokontrolerów, w tym algorytmów związanych z protokołami komunikacyjnymi. W tym kontekście, korzystne jest dokonywanie optymalizacji wykorzystania zasobów obliczeniowych i pamięci.



Rysunek 1: Model warstwowy architektury IoT.

Funkcjonowanie systemów IoT można ująć w model warstwowy [113], np. zawierający pięć warstw, jak pokazano na rysunku 1. Warstwa postrzegania reprezentuje fizyczne urządzenia i obiekty, np. zbierające dane o środowisku, takie jak temperatura, ciśnienie, dane o lokalizacji, ruchu, czy też zanieczyszczeniu powietrza. Warstwa transmisyjna odpowiada za bezpieczne przekazywanie danych i informacji pomiędzy urządzeniami lub z urządzeń do scentralizowanego systemu przetwarzania informacji [47]. Medium transmisyjne może być przewodowe lub bezprzewodowe, a wykorzystane może być całe spektrum technologii, takich jak sieci mobilne 4G [35], 5G [67, 27], w przyszłości 6G [97, 130], Wifi [35], WiMax [72], Bluetooth [48, 66], ZigBee [7, 45], LoRa [30, 8], NB-IoT [19, 140], LTE-Cat-M1 [133, 93], transmisja światłowodem [110, 114, 94] i inne. Warstwa oprogramowania pośredniczącego reprezentuje część architektury umożliwiającej połączenie niejednorodnych typów urządzeń, zapewniając pośrednictwo pomiędzy różnymi komponentami systemu, umożliwiając im komunikację, współpracę i integrację. Można tu także umiejscowić przetwarzanie danych zgromadzonych przez urządzenia IoT, zarówno lokalnie, jak i w chmurze, z zastosowaniem różnych metod analizy danych, a także podejmowanie autonomicznych decyzji i inicjowania działań. Warstwa aplikacji zapewnia globalne zarządzanie aplikacjami, dostarcza interfejsy użytkownika i aplikacje obsługujące konkretne przypadki użycia IoT. Warstwa biznesowa odpowiada za zarządzanie ekosystemem IoT, włączając w to aplikacje, usługi i tworzenie modeli biznesowych.

Stos protokołów warstwy transmisyjnej stanowi normatywny kanon komunikacji, który podlega stałemu rozwojowi w związku z postępującym rozwojem technologicznym. Możliwość jego rozszerzania o nowe protokoły wynika z nieustannego poszerzania horyzontów technicznych. Ponadto, obserwuje się istnienie dedykowanych protokołów, które są tworzone z myślą o konkretnej specyfice wymagań. Przykładem są wdrożenia w rozwiązaniach firmy TKH Technology, we współpracy z którą zrealizowana została niniejsza praca, w której projektowane są systemy dostosowane do zróżnicowanych potrzeb odbiorców. W przypadku, gdy powszechnie stosowane protokoły komunikacyjne, takie jak wymienione wyżej, spełniają wymagania klienta, wykorzystuje się je w celu zapewnienia komunikacji między urządzeniami IoT. Jednakże istnieją sytuacje, gdy szczególne założenia i potrzeby można osiągnąć skuteczniej korzystając z dedykowanych, opracowywanych wewnętrznie rozwiązań w warstwie transmisyjnej [9] .

Zastosowanie własnych protokołów komunikacyjnych w kontekście Internetu rzeczy (IoT) może dać szereg korzyści, wynikających z możliwości dostosowania wykorzystanych protokołów do specyficznych wymagań i warunków danego projektu. Można wskazać między innymi następujące korzyści:

- Możliwość optymalizacji wykorzystania zasobów: precyzyjne dostosowanie parametrów komunikacyjnych, takich jak przepustowość, oraz kontrolę zużycia energii i pamięci, co jest istotne dla efektywnego zarządzania ograniczonymi zasobami.
- Elastyczność i skalowalność: elastyczne dostosowanie rozwiązań do zmieniających się wymagań projektu, a także możliwość sprawnej rozbudowy i skalowalność systemu.
- Brak zależności od standardów zewnętrznych, co pozwala na swobodniejsze kształtowanie rozwiązań, eliminując potencjalne ograniczenia narzucane przez standardy.
- Możliwość optymalizacji obciążenia łącz: dedykowane protokoły umożliwiają swobodne dostosowanie wielkości paczek przesyłanych danych, jak również szybkości transmisji, a co za tym idzie – obciążenia poszczególnych połączeń oraz opóźnień w transmisji.
- Możliwość wykorzystania dedykowanych technik adaptacyjnych w celu dopasowania do zmieniających się warunków transmisyjnych.
- Wykorzystanie niesymetryczności łącza. W sieciach o topologii gwiazdy występuje odmienny charakter urządzeń po dwóch stronach łącza, a węzeł centralny posiada zwykle znacznie większe zasoby obliczeniowe i energetyczne niż węzeł końcowy. Wówczas istnieje możliwość zastosowania zróżnicowanych technik transmisji i protokołów w łączach w górę (*uplink*) i w dół (*downlink*), co pozwala na dodatkową minimalizacją zużycia zasobów w urządzeniu wbudowanym.

Prace badawczo-wdrożeniowe, przedstawione w niniejszej pracy, są związane z jednym z istotnych elementów dedykowanych protokołów w warstwie transmisyjnej, a mianowicie z kodowaniem korekcyjnym w łączu komunikacyjnym. Niezależnie od wykorzystanego medium czy technologii transmisji, detekcja i/lub korekcja błędów jest niezbędnym elementem protokołu. Silnie ograniczone zasoby

systemów urządzeń końcowych sieci IoT sugerują wykorzystanie stosunkowo niezłożonych metod korekcji, z dekodowaniem tzw. twardo-decyzyjnym, takich jak np. CRC do detekcji błędów [104, 13], kody Hamminga [51, 18], ewentualnie nieco bardziej złożone kody blokowe: BCH [21, 92] lub RS [107, 111].

Wykonane prace pokazały jednak, że zastosowanie zaawansowanych metod korekcji, w szczególności kodowania LDPC (ang. *Low-Density Parity-Check*) [87] z dekodowaniem miękko-decyzyjnym, niebinarnego kodowania LDPC [29], kodowania z macierzami quasi-cyklicznymi QC-LDPC [41], jak również kodowania typu Raptor [25], ma uzasadnienie w możliwości uzyskania zwiększonej efektywności energetycznej układu kodera implementowanego w mikrokontrolerze i układu transmisji zabezpieczonej tymi kodami. W szczególności, można tu wykorzystać wspomnianą niesymetryczność, z zaawansowanym kodowaniem zastosowanym tylko w łączu w górę, dzięki któremu można obniżyć moc sygnałów transmitowanych przez urządzenie końcowe IoT. Operacja dekodowania, która zwykle cechuje się dużą złożonością, jest przeprowadzana w węźle centralnym, w którym limit zasobów obliczeniowych i energetycznych jest na znacznie wyższym poziomie. Kody LDPC znalazły zastosowanie w wielu zaawansowanych standardach komunikacyjnych, w których kodeki, ze względu na dużą wymaganą przepustowość, są zazwyczaj implementowane w specjalizowanych układach scalonych (ICs) lub sprzętowych układach programowalnych (FPGA). Tematyka implementacji programowej w mikrokontrolerach nie była w istotnym zakresie podejmowana przez badaczy.

Urządzenia elektroniczne, zwłaszcza te zasilane bateryjnie, wymagają efektywnych strategii zarządzania energią, aby przedłużyć ich żywotność. Stan aktywny i stan uśpienia są dwoma podstawowymi stanami pracy układu, które charakteryzują działanie systemów wbudowanych, na których oparte są urządzenia IoT. W stanie aktywnym urządzenie jest w pełni funkcjonalne i wykonuje zadania, w tym gromadzenie i przetwarzanie danych, oraz przygotowanie bloku danych do transmisji, uwzględniające kodowanie korekcyjne. W przeciwieństwie do tego, w stanie uśpienia urządzenie przechodzi w stan ograniczonej aktywności, aby zminimalizować zużycie energii. Wykorzystywanie stanu uśpienia jest szczególnie istotne w urządzeniach zasilanych bateryjnie, gdzie długa żywotność baterii jest kluczowa. Przejścia pomiędzy stanem aktywnym a stanem uśpienia mogą być wyzwalane różnymi czynnikami, takimi jak harmonogramy czasowe, zdarzenia

wywołujące, czy też niski poziom naładowania baterii. Implementacje zarządzania energią dla urządzeń IoT mogą różnić się w zależności od specyfiki konkretnego systemu, jednak celem zawsze jest osiągnięcie optymalnego balansu między wydajnością a efektywnością energetyczną,

Z punktu widzenia efektywności energetycznej operacji kodowania, istotne jest zapewnienie jak najkrótszego czasu przygotowania zakodowanego bloku, tak, aby wymagany stan aktywności układu obliczeniowego był jak najkrótszy. W tym celu, zaproponowano wykorzystanie podklasy kodów quasi-cyklicznych (QC-LDPC) wraz z opracowanym algorytmem przetwarzania bitów w blokach, o rozmiarze nie większym niż rozmiar podmacierzy macierzy kontrolnej kodu QC-LDPC. Dodatkowo, można wskazać wymienione niżej przesłanki, które zastosowano stawiając cele, które zostaną przedstawione w kolejnym podrozdziale oraz w wyborze rozwiązań prowadzących do osiągnięcia tych celów.

- Istotny jest odpowiedni dobór długości słów kodowych. Skrócenie bloków ogranicza możliwości korekcyjne kodów, lecz jednocześnie umożliwia zmniejszenie czasu kodowania i czasu transmisji, redukując zużycie energii. Co więcej, komunikaty transmitowane w rozważanych systemach IoT są zwykle stosunkowo krótkie. Eksperymenty przeprowadzone w ramach niniejszej pracy skoncentrowano na krótkich do średnich długościach bloków kodowych, w zakresie od ok. 64b do ok. 1024b.

- Opracowany schemat kodowania powinien być skuteczny w różnych warunkach transmisji, z różnymi poziomami zakłóceń. Zastosowanie schematów z redundancją inkrementalną, np. kodowania typu Raptor, pozwala na dynamiczną adaptację, jak również sterowanie zależnością pomiędzy sprawnością kodowania (a w konsekwencji jednostkowym zużyciem energii) a możliwościami korekcyjnymi.

- Wykorzystanie zaawansowanych kodów dedykowanych krótkim blokom, takich jak niebinarne kody LDPC, pozwala na osiągnięcie zwiększonych możliwości korekcyjnych lub możliwości dodatkowej redukcji energii, pod warunkiem opracowania odpowiednio efektywnego kodera implementowanego w mikrokontrolerze. Duża złożoność obliczeniowa dekodowania niebinarnego nie stanowi przeszkody w zastosowaniu takich schematów w łączu w górę.

- Optymalizacja parametrów kodowania oraz opracowanie uniwersalnego algorytmu tworzenia kodów typu Raptor pozwala na skuteczne wdrożenie rozwiązań zoptymalizowanych pod względem efektywności energetycznej układu transmisji wykorzystujących krótkie bloki danych.

- Punktem odniesienia do wyników prac mogą być klasyczne metody kodowania blokowego (kody BCH i kody RS), jak również adaptacyjny schemat kodowania LDPC ze standardu 5G.

W opisanym kontekście realizowano prace badawczo-wdrożeniowe, których nadrzędnym celem było zaproponowanie oraz efektywna energetycznie implementacja algorytmów korekcji danych w zakłóconym kanale komunikacyjnym dla urządzeń typu IoT, opartych na mikrokontrolerze. Zdecydowano się na zastosowanie metod kodowania i dekodowania LDPC, binarnego i niebinarnego, które są znane ze swojej znakomitej skuteczności w korekcji błędów, jak również kodowania adaptacyjnego typu Raptor. Wybór tych konkretnych metod był podyktowany ich zdolnością do efektywnej redukcji błędów transmisji danych przy zachowaniu relatywnie niskiego zużycia zasobów obliczeniowych, co jest istotne w kontekście ograniczonych zasobów dostępnych w urządzeniach IoT.

3.2 Istotne pojęcia

Liniowy kod blokowy $\mathcal{C}(N,K)$ o długości wektora informacyjnego K, długości wektora kodowego $N > K$ i liczbie słów kodowych 2^K jest zbiorem wszystkich wektorów $\mathbf{c}_{1\times K}$ spełniających równanie macierzowe $\mathbf{H}\cdot\mathbf{c}^T = \mathbf{0}$ w ciele $GF(q)$ (dla kodów binarnych $q = 2$), gdzie $\mathbf{H}$ to **macierz kontrolna** kodu.

Kody **LDPC (*Low-Density Parity-Check*)** to rodzina liniowych kodów blokowych, w których macierz kontrolna jest macierzą rzadką, tzn. zdecydowana większość jej elementów jest zerami [87]. Kody LDPC mogą być dekodowane jednym z algorytmów BP (*Belief Propagation*) [87, 42], czyli iteracyjnych algorytmów propagacji przybliżonych wartości prawdopodobieństw, które pozwalają na wykorzystanie kodów o długich blokach, przy złożoności obliczeniowej liniowo zależnej od długości bloku N.

Graf kodu (graf Tannera) to graf dwudzielny, w którym jeden podzbiór wierzchołków odpowiada równaniom kontrolnym kodu (czyli wierszom macierzy $\mathbf{H}$),

a drugi podzbiór – bitom słowa kodowego (kolumnom macierzy **H**). Graf Tannera definiuje sposób propagacji wiadomości w algorytmach BP.

Możliwości korekcyjne kodów, w tym LDPC, są zwykle opisywane poprzez zależność prawdopodobieństwa błędu na wyjściu dekodera (estymowanego przez BER – **bitową stopę błędów**) w funkcji parametru kanału, zwykle **stosunku sygnału do szumu (SNR)** kanału **AWGN (*Additive White Gaussian Noise*)**. Możliwości korekcyjne kodu LDPC zależą od długości bloku kodu oraz od zawartości macierzy kontrolnej, a w szczególności związanych z tym własności strukturalnych skojarzonego grafu Tannera, np. [52, 53, 68, 71, 17].

Kody LDPC **typu Raptor** (RL – *Raptor-Like coding*) [25] to jedna z metod kodowania z redundancją przyrostową, pochodząca od kodowania Raptor [115], która łączy blokowy kod bazowy LDPC z dodatkowymi porcjami redundancji, definiowanymi przez dodatkowe równania kontrolne o niskiej gęstości. W takim schemacie kodowania, jeśli początkowa transmisja słowa kodowego LDPC o dużej sprawności nie zostanie skutecznie zdekodowana, dosyłane są dodatkowe bloki parzystości. W efekcie, jest to transmisja z tzw. **redundancją przyrostową** (IR – *Incremental Redundancy*) [83], w której dodatkowe nadmiarowe porcje symboli są dosyłane po zaobserwowaniu błędów.

Kody QC-LDPC (*Quasi-Cyclic LDPC*) [41] to podklasa kodów LDPC o macierzy kontrolnej składającej się z podmiacierzy kwadratowych, w których kolejne wiersze są cyklicznym przesunięciem pierwszego wiersza podmacierzy. Kody zastosowane w standardzie 5G [4] mogą być sklasyfikowane jako rodzina binarnych kodów QC-LDPC typu Raptor. Kody QC-LDPC charakteryzują się możliwością efektywnej implementacji sprzętowej dekoderów iteracyjnych [82, 100, 128].

Niebinarne kody LDPC (NB-LDPC) [29] to uogólnienie binarnych kodów LDPC, w którym macierz kontrolna oraz równanie kontrolne jest definiowane nad ciałem $GF(q > 2)$. Pozwala to na osiągnięcie zwiększonych możliwości korekcyjnych, szczególnie w zakresie stosunkowo krótkich do średnich długości bloku (kilkadziesiąt do kilkaset symboli) [29, 33, 17], kosztem istotnie zwiększonej złożoności obliczeniowej dekodowania [31, 75].

Kody Hamminga [51, 18] to liniowe kody blokowe, które umożliwiają wykrywanie i korekcję pojedynczych błędów bitowych. Kody **BCH (Bose-Chaudhuri-Hocquenghem)** [21, 92] to rodzina korekcyjnych kodów blokowych bazujących na wielomianach nad ciałem Galois, oferująca elastyczność

w dostosowywaniu zdolności korekcyjnej do konkretnych wymagań systemu komunikacyjnego. **Kody RS (Reed-Solomon)** [107, 111] to rodzina niebinarnych liniowych kodów blokowych. Zastosowanie kodów BCH i RS obejmuje różne dziedziny, od przechowywania danych na nośnikach magnetycznych i nośnikach Flash, po transmisję w komunikacji satelitarnej.

3.3 Teza badawcza, cel i zakres pracy

W opisanym kontekście oraz w związku z postawionymi celami, które zostaną przedstawione poniżej, sformułowano następującą tezę.

W przypadku implementacji koderów LDPC w systemach mikroprocesorowych można wskazać konstrukcję kodera oraz klasę kodów skutkujących uzyskaniem szczególnie efektywnych energetycznie systemów transmisji danych. Implementacje te należą do najefektywniejszych w całej klasie kodów blokowych, a ich efektywność można dalej zwiększać przez zastosowanie techniki kodowania typu Raptor.

Zaproponowana teza związana jest z głównymi celami projektu, które były następujące:

- obniżenie poboru energii w systemach wbudowanych w urządzenia typu IoT wykorzystujących niestandardowe protokoły komunikacji w zakłóconych łączach bezprzewodowych lub przewodowych, poprzez zastosowanie nowoczesnych kodów dekodowanych iteracyjnie: binarnych QC-LDPC, niebinarnych QC-LDPC do zabezpieczania transmisji, zamiast klasycznych kodów blokowych;
- zapewnienie skutecznej i efektywnej energetycznie korekcji błędów transmisji w sieciach IoT, wykorzystujących niestandardowe protokoły komunikacji i zróżnicowane media komunikacyjne, w szczególności w łączności pomiędzy węzłami o niesymetrycznym charakterze: od węzłów końcowych dysponujących stosunkowo niewielkimi zasobami obliczeniowymi, do bramek, o znacznie większych zasobach obliczeniowych i energetycznych.

Dla osiągnięcia założonych celów oraz wykazania postawionej tezy, zaplanowano i wykonano następujące prace badawczo-wdrożeniowe:

- opracowanie programowych modeli układów kodowania i dekodowania LDPC i NB-LDPC, kanałów i innych elementów środowiska numerycznego umożliwiającego statystyczne określanie możliwości korekcyjnych badanych systemów transmisji;
- implementacja koderów kodów LDPC oraz niebinarnych kodów LDPC w CPU typowych współczesnych mikrokontrolerów, konkurencyjnych do koderów

klasycznych metod kodowania (np. BCH, RS) pod względem szybkości działania i zużycia energii;

- utworzenie eksperymentalnych macierzy kontrolnych podklasy kodów efektywnie kodowanych (QC) o różnych rozmiarach bloków i sprawnościach, które umożliwiają transmisję stosunkowo krótkich wiadomości charakterystycznych dla sieci typu IoT;
- opracowanie i eksperymentalna weryfikacja systemu kodowania dla łącz o nieznanych i zmiennych w czasie parametrach: wykorzystanie redundancji przyrostowej (IR – *Incremental Redundancy*) i kodowania typu Raptor (RL – *Raptor-Like*), z efektywnie kodowanymi macierzami typu LDPC, RL-QC-LDPC;
- badania eksperymentalne: implementacja efektywnego kodera w CPU, określenie jego parametrów czasowych i zużycia energii; oszacowanie całościowego zużycia energii przez koder i nadajnik radiowy przy stosowaniu różnych metod kodowania; porównanie zużycia energii na jeden przesłany bit w systemach z kodowaniem LDPC o stałej sprawności z systemem z przyrostową redundancją i kodowaniem typu Raptor;
- określenie możliwości uzyskania dodatkowej oszczędności energii w układzie nadawczym systemu dla łącz o nieznanych i zmiennych w czasie parametrach, dla zaproponowanych kodów i implementacji, jak również kodów standardu 5G, implementowanych w CPU mikrokontrolera;
- zaproponowanie metody tworzenia kodów niebinarnych, NB QC-RL-LDPC o krótkich blokach i porównanie z kodowaniem binarnym standardu 5G, z wykorzystaniem systemu z redundancją przyrostową;
- implementacja kodera kodów niebinarnych LDPC w układzie mikrokontrolera i eksperymentalne określenie jego parametrów czasowych i zużycia energii;
- przeprowadzenie badań eksperymentalnych i uzyskanie: wykresów stopy błędów / jednostkowego zużycia energii w systemie transmisyjnym / efektywnej przepustowości, dla utworzonych kodów o krótkich blokach, binarnych i niebinarnych LDPC, w tym kodów typu Raptor.

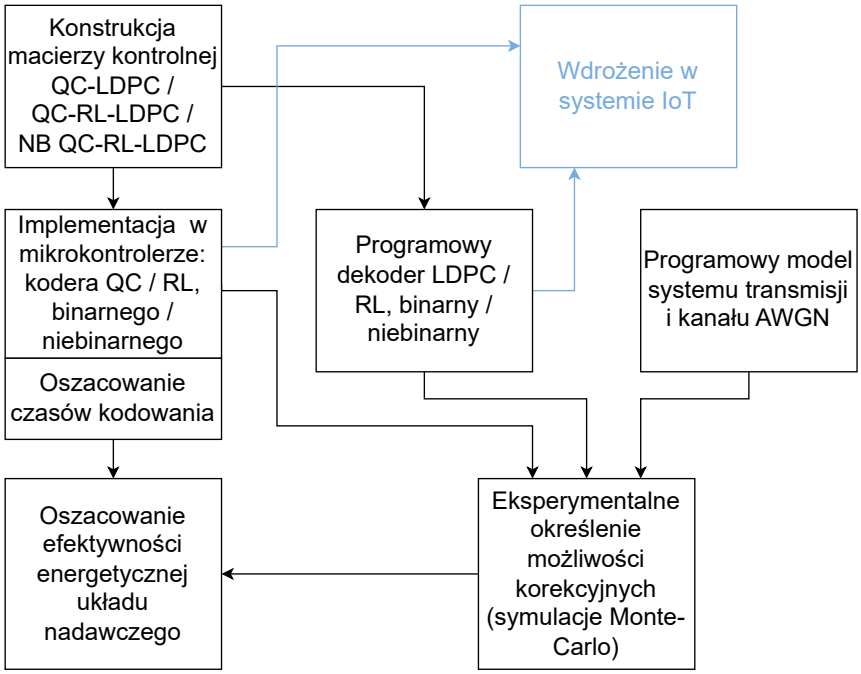
3.4 Wykonane prace badawcze i uzyskane wyniki

W prowadzonych pracach wykorzystano współczesne układy mikrokontrolerowe rodziny STM32 do implementacji różnych kodów pochodzących z różnych klas, takich jak LDPC, QC-LDPC, NB-LDPC oraz kody typu Raptor. Opracowane rozwiązania programowe mogą być przy niewielkim wysiłku wykorzystane w układach wbudowanych bazujących na innym sprzęcie. Pomiary przeprowadzono dla szeregu parametrów, włączając w to czas kodowania/dekodowania, zajętość pamięci oraz zużycie energii. Do pomiarów wykorzystano odpowiednio dobrany sprzęt pomiarowy, w tym multimetry z wysoką dokładnością do pomiaru prądu, oscyloskop, analizator stanów logicznych, jak również środowisko programistyczne pozwalające na debuggowanie opracowywanych rozwiązań. Te narzędzia umożliwiły precyzyjną ocenę różnych parametrów oraz analizę zachowania układów mikrokontrolerowych podczas implementacji koderów i dekoderów.

Drugim elementem warsztatu badawczego były modele programowe przygotowane w środowisku Matlab, obejmujące kodery i dekodery dla różnych rodzajów kodów, jak LDPC (zarówno binarne, jak i niebinarne) oraz kody typu Raptor, modele modulatorów i demodulatorów QPSK i QAM, model kanału AWGN oraz oprogramowanie pozwalające na symulacje Monte-Carlo systemu tranmisji. Pozwoliło ono na analizę możliwości korekcyjnych kodów w zależności od poziomu zakłóceń w kanale, jak również zostało wykorzystane do przygotowania środowiska umożliwiającego symulacje różnych warunków opracowanie graficzne wyników. Modele pozwoliły określać możliwości korekcyjne kodów w zależności od poziomu zakłóceń w kanale. Fuzja wyników pomiarów: zużycia energii, czasu kodowania oraz możliwości korekcyjnych pozwoliła na oszacowanie efektywności energetycznej.

Prace rozpoczęto od wykonania przeglądu istniejących metod kodowania korekcyjnego, w szczególności pod kątem zastosowania w systemach typu IoT. Rozważano kody, które mają możliwie najlepsze własności korekcyjne przy krótkich blokach kodowych i akceptowalnej złożoności operacji kodowania. W dalszych pracach skupiono się na kodach LDPC (Low-Density Parity-Check). Są one uznawane za jedną z najlepszych metod korekcji błędów w sensie wydajności korekcji i są wykorzystywane w znaczących standardach radiokomunikacyjnych, takich jak WiFi, WiMAX, DVB-T2, DVB-S2 i sieciach mobilnych 5G. Dodatkowo, niebinarne kody LDPC mają znakomite możliwości korekcyjne dla krótkich bloków. Założone prace

obejmowały w pierwszej kolejności dokonanie analizy i przeglądu charakterystycznych cech podklas kodów LDPC, w tym w szczególności QC-LDPC (Quasi-Cyclic LDPC), podklasy dającej możliwość efektywnej implementacji. Został opracowany programowy model kodera, kanału i dekodera, umożliwiający eksperymentalne określanie stopy błędów systemu kodowania. Wykonany przegląd kodów stosowanych w standardach komunikacyjnych oraz ich eksperymentalnie zweryfikowane możliwości korekcyjne przedstawiono w **Publikacji 1**. Przegląd istniejących rozwiązań i wyników badań był kontynuowany w kierunku kodów niebinarnych NB-LDPC, kodów polarnych, metod kodowania adaptacyjnego, w tym kodowania typu Raptor, algorytmów tworzenia kodów, algorytmów kodowania i dekodowania oraz implementacji koderów. Wyniki tego przeglądu znajdują się w rozdziałach wstępnych w kolejnych publikacjach 2-6 cyklu, a lista najistotniejszych pozycji znajduje się w rozdziale **Literatura**.



Rysunek 2: Podsumowanie metodologii prac badawczo-wdrożeniowych.

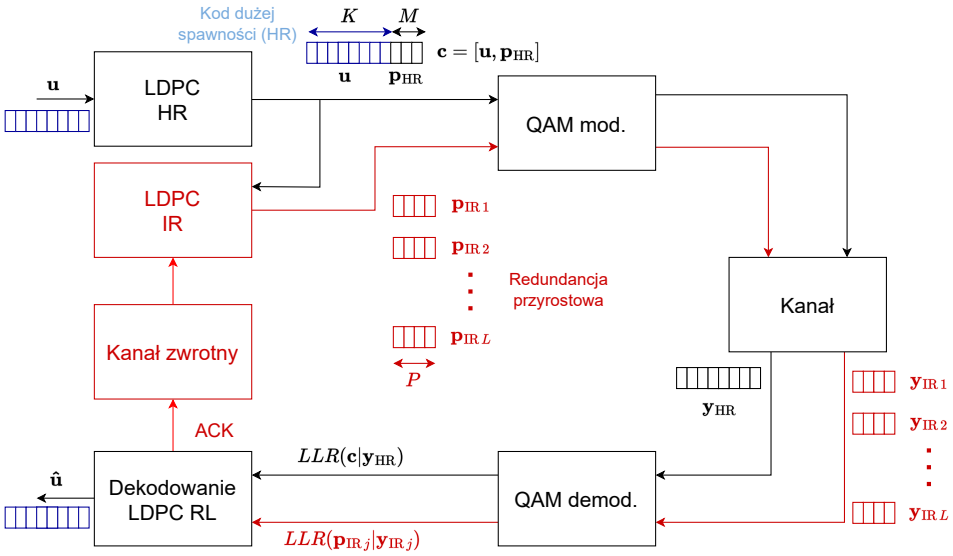
W następnym etapie podjęto próbę wykazania, że schemat oparty na kodowaniu LDPC może być z powodzeniem zaimplementowany w układzie nadawczym systemu z silnie ograniczonymi zasobami obliczeniowymi. W **Publikacji 2** przedstawiono implementację w urządzeniu IoT opartym o mikrokontroler oraz zamieszczono wyniki badań eksperymentalnych. Opracowana implementacja jest

przeznaczona dla systemu typu Internetu rzeczy (IoT), w którym niskowydajne urządzenie końcowe koduje komunikaty przesyłane do bramki. Wykorzystano efektywny algorytm kodowania LDPC Richardsona-Urbanke i pokazano, jak operacje elementarne tego algorytmu mogą być dla podklasy kodów QC-LDPC zwektoryzowane w programowej implementacji oraz zaprogramowane w formie operacji przesunięć cyklicznych i dodawania w ciele Galois (GF(2)), bezpośrednio odpowiadających instrukcjom CPU. Przedstawiono także efektywny zapis macierzy parzystości i organizację pamięci dla przechowywania wyników obliczeń pomocniczych.

Wyniki eksperymentalne wykazują znaczący wzrost efektywności w zakresie zużycia pamięci i czasu kodowania w porównaniu z kodowaniem przy użyciu bezpośredniej reprezentacji macierzy parzystości. W artykule przeanalizowany został problem złożoności obliczeniowej kodera oraz omówiono wyzwanie związane z przetwarzaniem dużej liczby danych przez urządzenia o ograniczonych możliwościach obliczeniowych. **Publikacja** 2 zawiera porównania eksperymentalne z innymi znanymi kodami blokowymi, takimi RS i BCH, ukazując, że wymagania pamięciowe dla kodów LDPC nie są większe niż dla standardowych kodów blokowych. Jednocześnie wskazano możliwości uzyskania obniżonego czasu kodowania względem innych standardowych koderów blokowych, w tym koderów cyklicznych takich jak BCH bez wektoryzacji operacji kodowania w CPU, co prowadzi także do obniżenia zużycia energii. Dodatkowo zweryfikowano eksperymentalnie zysk kodowania LDPC, który przy zastosowaniu dekodera z miękkimi decyzjami, jest większy niż w przypadku wspomnianych kodów BCH i RS, porównywalnych rozmiarem bloku. Niniejsze badanie wnosi wkład w poszerzanie zastosowań kodowania LDPC w systemach o ograniczonych zasobach, prezentując jego zalety w zakresie efektywności i korekcji błędów w porównaniu z tradycyjnymi kodami blokowymi, implementowanymi programowo. Wyniki płynące z efektywnej implementacji algorytmu kodowania są bardzo obiecujące, szczególnie dla wykorzystania w urządzeniach IoT zasilanych bateriami. Rozwiązanie zostało wykorzystane w urządzeniu przemysłowym IoT opartym na mikrokontrolerze o ograniczonych możliwościach obliczeniowych.

Założona koncepcja systemu, z kodowaniem LDPC w kierunku w górę, nie zakłada konieczności konstrukcji dekodera w układzie programowym o silnie ograniczonych zasobach. Tym niemniej, w ramach prac komplementarnych, opracowano

także także implementację iteracyjnego dekodera LDPC w układzie mikrokontrolera. Ta część prac została opisana w **Publikacji 3**. Przedstawiono wyniki dotyczące zajętości pamięci, czasu dekodowania bloku informacyjnego, przepustowości dekodera oraz wyników dekodowania w funkcji liczby błędów w dyskretnym kanale z zakłóceniami. Pokazano, że możliwa jest implementacja dekodera kodów LDPC oraz QC-LDPC, także nieregularnych, w układzie mikrokontrlera, mimo znacznych wymagań dotyczących mocy obliczeniowej dla kodów LDPC. Wskazano, jaka przepustowość może być uzyskana przy pełnym obciążeniu typowego mikrokontrolera oraz zaproponowano przybliżony wzór obliczania przepustowości w takim układzie.



Rysunek 3: Proponowany model komunikacji w górę dla Internetu rzeczy (IoT) z wykorzystaniem schematu korekcyjnego błędów z redundancją przyrostową. [**Publikacja 4** – kody binarne, **Publikacja 6** – kody niebinarne]

Jednym z podstawowych celów prowadzonych prac było zaproponowanie schematów kodowania o jak najlepszej efektywności energetycznej, w sensie energii jednostkowej, czyli energii zużywanej na transmisję 1 bitu informacyjnego. W **Publikacji 4** pokazano, że istnieje możliwość uzyskania zwiększonej efektywności energetycznej transmisji względem transmisji z kodowaniem QC-LDPC o stałej sprawności, przy zastosowaniu kodowania typu Raptor (RL – *Raptor Like*), opartego o kod bazowy QC-LDPC (QC-RL-LDPC).

Uproszczony schemat takiego systemu przedstawiony jest na rys. 3. Wykonane prace dotyczące binarnego schematu kodowania typu Raptor, opisane w **Publikacji 4**, obejmowały implementację kodera QC-RL-LDPC w układzie mikrokontrolera, zaproponowanie oryginalnej przyrostowej metody dekodowania opartej na algorytmie dekodowania BP kodów LDPC (rys. 5 w **Publikacji 4**), rozwinięcie znanych metod konstrukcji kodów LDPC na kody typu QC-RL-LDPC oraz wykonanie badań eksperymentalnych, obejmujących określenie skuteczności korekcji, w formie obserwowanej sprawności bitowej (ang. *goodput*) w funkcji stosunku do szumu w kanale (SNR) oraz efektywności energetycznej urządzenia nadawczego, czyli jednostkowego zużycia energii przez koder i nadajnik. Przeprowadzono porównanie znormalizowanych statystyk zużycia energii z przypadkiem kodowania LDPC o stałej długości.

W obliczeniach oczekiwanego zużycia energii wykorzystano wyniki pomiarów poboru energii przez typowy mikrokontroler realizujący operację kodowania oraz pomiary czasów kodowania. Założono, że mikrokontroler jest w stanie aktywnym tylko w trakcie kodowania. Następnie oszacowano wymaganą moc transmisji w typowym systemie, uwzględniając eksperymentalne wyniki związane z bitową stopą błędów w funkcji stosunku sygnału do szumu w kanale. Wyniki eksperymentów, między innymi te pokazane na rys. 9, 12, 13 w **publikacji 4**, wskazują, że:

- w porównaniu z kodowaniem o stałej sprawności, możliwe jest uzyskanie mniejszego jednostkowego zużycia energii dla wysokich poziomów stosunku sygnału do szumu (SNR),
- możliwa jest skuteczna transmisja z korekcją dla niskich poziomów SNR, w których możliwości korekcyjne kodu o stałej sprawności nie pozwalają już na skuteczną korekcję,
- wygenerowane kody są konkurencyjne do kodów wykorzystywanych w standardzie 5G wykorzystujących podobny schemat redundancji przyrostowej.

W kolejnym etapie prac badawczych, opracowane rozwiązania kodowania typu Raptor rozwinięto na kodowanie niebinarne (NB), zastosowane jako kodowanie bazowe, jak również kodowanie dodatkowych równań kontrolnych. Opracowano programową implementację kodera NB-LDPC, z wykorzystaniem efektywnego algorytmu kodowania, która może być z powodzeniem wykorzystana także dla

niebinarnych kodów typu Raptor oraz zaimplementowano taki koder w mikrokontrolerze. Prace te zostały opisane w **Publikacji 5**. Podkreślono istotne aspekty związane z wydajnością implementacji algorytmu kodowania, sugerując, że skuteczne wdrożenie kodów niebinarnych może przynieść znaczne korzyści, zwłaszcza dla urządzeń IoT zasilanych bateryjnie. Wnioski z przeprowadzonych badań wskazują na obiecujące perspektywy zastosowania kodów niebinarnych, w łączu w górę systemów IoT, ze względu na dobrą efektywność energetyczną układu nadawczego. W ramach eksperymentów dokonano porównania zależności czasowych dla kodów nad różnymi rzędami ciał Galois (GF), co pozwoliło na zbadanie wpływu wyboru kodu na potencjalne zużycie energii.

W ramach prac nad niebinarnym kodowaniem typu Raptor opracowano także algorytmiczną metodę konstrukcji kodów NB QC-RL-LDPC, jak również metodę mapowania niebinarnych symboli na konstelacje modulacji QAM o dowolnym rzędzie i programowy dekoder będący rozwinięciem dekodera kodów binarnych typu Raptor. Te prace oraz uzyskane wyniki zawarto w manuskrypcie oznaczonym jako **Publikacja 6**. Zaproponowano procedurę konstruowania krótkich kodów NB QC-RL-LDPC, dostosowanych do efektywnej implementacji kodera o niskiej złożoności i zoptymalizowanych strukturach grafu kodu. Opracowana procedura, która została zaprogramowana w środowsiku Matlab, opiera się na optymalizacji grafu kodu bazowego poprzez minimalizację liczby cykli, a następnie oryginalnej metodzie określania pozycji symboli w dodatkowych równaniach kontrolnych redundancji przyrostowej, wspieranej analizą grafu oraz symulacjami Monte Carlo. Eksperymenty numeryczne potwierdziły skuteczność algorytmu. Uzyskane kody niebinarne, wraz z opracowanym algorytmem dekodowania, cechują się zwiększoną efektywnością korekcji w porównaniu z binarnym kodowaniem 5G z podobnym schematem redundancji przyrostowej. Pokazano także, że różnicowanie rzędów modulacji QAM poprzez kombinację modulacji o wyższym rzędzie, na przykład ($X = 64$)-QAM dla transmisji początkowej, z modulacją o niższym rzędzie, na przykład ($X_{\mathrm{IR}} = 4$)-QAM dla redundancji przyrostowej, może być korzystne pod względem wydajności w zakresie niskich wartości SNR (poniżej 0 dB i nieznacznie powyżej). Elastyczność łączenia kodów LDPC nad ciałem GF o dowolnym rzędzie z modulacjami o dowolnym rzędzie zapewnia szeroką adaptacyjność, oferując szeroki zakres możliwych efektywności widmowych. Uzyskany zysk kodowania

może się bezpośrednio przekładać na zysk energetyczny w układzie nadawczym, poprzez możliwość transmisji ze zmniejszoną mocą.

3.5 Podsumowanie

Opracowanie systemu kodowania korekcyjnego, które pozwala na skuteczną korekcję błędów przy jak najlepszej efektywności energetycznej układu wbudowanego, o ograniczonych zasobach obliczeniowych, jest zadaniem nietrywialnym, które podjęto w ramach prac badawczo-wdrożeniowych, opisanych w złączonym cyklu publikacji.

Implementacja różnych rodzajów kodów, w tym LDPC, QC-LDPC, NB LDPC oraz kodów typu Raptor, na współczesnych mikrokontrolerach umożliwiła analizę ich efektywności energetycznej oraz wydajności w warunkach ograniczonych zasobów obliczeniowych. Badania obejmowały pomiary czasu kodowania i dekodowania, zajętości pamięci oraz zużycia energii, co pozwoliło na precyzyjną ocenę parametrów rozwiązań kodowania korekcyjnego w układach wbudowanych. Opracowano modele programowe w środowisku Matlab, obejmujące różnorodne rodzaje kodów, co umożliwiło analizę ich możliwości korekcyjnych w zależności od poziomu zakłóceń w kanale. Te modele pozwoliły również na symulację różnych warunków transmisji oraz opracowanie wyników badań.

Przedstawione rozwiązania i wyniki stanowią istotny wkład w rozwój technik kodowania LDPC w kontekście zastosowań w urządzeniach typu IoT. Kompleksowe opisanie skutecznych implementacji kodów LDPC, QC-LDPC, QC-RL-LDPC, NB-LDPC, a także procedur konstrukcji takich kodów oraz skutecznego dekodowania w systemach z redundancją przyrostową i kodowaniem LDPC, poszerza zrozumienie możliwości wykorzystania nowoczesnych metod kodowania korekcyjnego w środowisku IoT, gdzie zasoby są ograniczone, a wymagania dotyczące efektywności energetycznej są wysokie. Implementacja kodera i dekodera w urządzeniu IoT stanowi krok w rozwoju programowych rozwiązań kodowania korekcyjnego. Dzięki temu możliwe jest wykorzystanie zalet kodów LDPC w aplikacjach IoT, nawet w przypadku urządzeń o ograniczonych zasobach obliczeniowych i energetycznych.

Adaptacyjne kodowanie typu Raptor, dla którego opracowano i rozwijano rozwiązania implementacji i konstrukcji kodów, stanowi metodę, która pozwala na elastyczną reakcję na zmiany warunków transmisji, co może być niezwykle korzystne w dynamicznym środowisku IoT. Wprowadzenie niebinarnego kodowania LDPC oraz niebinarnego kodowania typu Raptor może dać dodatkowe korzyści związane ze zwiększonymi możliwościami korekcji, w szczególności przy stosunkowo

krótkich blokach kodowych, które są charakterystyczne w systemach IoT. Analiza efektywności w warunkach ograniczonych zasobów obliczeniowych stanowi istotny krok w dostosowywaniu tej technologii do specyficznych wymagań systemów wbudowanych. Dzięki temu możliwe jest wykorzystanie zalet kodów LDPC w aplikacjach IoT.

Kluczowe osiągnięcia i wyniki, przedstawione w cyku publikacji, w kontekście sformułowanej tezy oraz celów pracy, w mniemaniu autora są następujące:

- W **publikacji 2** opisano oryginalne rozwiązanie efektywnej implementacji binarnego kodera QC-LDPC w mikrokontrolerze oraz przedstawiono wyniki eksperymentalne pokazujące istotne ograniczenie jednostkowego zużycia energii opracowanego rozwiązania względem kodera RS, BCH oraz LDPC bez wektoryzacji obliczeń.

- W **publikacji 4** opisano oryginalne rozwiązanie efektywnej implementacji kodera w schemacie redundancji przyrostowej, QC-RL-LDPC, oryginalny algorytm dekodowania kodów QC-RL-LDPC i środowisko symulacji kodowania RL oraz przedstawiono wyniki eksperymentalne, wskazujące możliwość uzyskania obniżonego jednostkowego zużycia energii w porównaniu z kodowaniem bez adaptacji sprawności oraz możliwość transmisji przy skrajnie niskich poziomach stosunku mocy sygnału do mocy szumu systemu.

- W **publikacji 5** uogólniono metodę kodowania na niebinarne kody LDPC nad ciałami $GF(2^2) \ldots GF(2^8)$ oraz pokazano, że także niebinarny koder LDPC może być efektywnie zaimplementowany w układzie mikrokontrolera.

- W **publikacji 6** zaproponowano oryginalną algorytmiczną metodę konstrukcji niebinarnych kodów QC-RL-LDPC dostosowanych do efektywnej implementacji kodera, uogólniono algorytm dekodowania przyrostowego na kody niebinarne oraz pokazano, jak niebinarne kody skutecznie łączyć z modulacjami QAM o zmiennym rzędzie. Przedstawiono wyniki pokazujące zwiększoną efektywność korekcji w porównaniu z binarnym kodowaniem LDPC standardu 5G, dające potencjalnie możliwość obniżenia energii transmitowanych sygnałów.

Wskazane kluczowe wyniki wykazują słuszność sformułowanej tezy, a założone cele pracy zostały osiągnięte. Macierze kontrolne kodów wykorzystanych w eksperymentach zamieszczone są w repozytorium `https://github.com/JakubHyla93/LDPC_Simulations`. Uzyskanie dodatkowego zysku kodowania przez skonstruowane niebinarne kody QC-RL-LDPC, względem kodów 5G oraz jednocześnie pokazanie efektywnej implementacji kodera, pozwala stwierdzić, że niebinarne kody, przy odpowiednim ich wykorzystaniu, mają potencjał dodatkowego zwiększenia efektywności energetycznej.

Opracowane rozwiązania i implementacje zostały częściowo wdrożone jako istotny element projektów prowadzonych przez grupę TKH Group związanych z Internetem rzeczy oraz mają potencjał dalszego wykorzystania w innych projektach i ulepszania produktów, w szczególności pod kątem efektywności energetycznej. Otwierają nowe perspektywy zarówno dla działań badawczo-rozwojowych, jak i biznesowych. Osiągnięte zwiększenie wydajności transmisji danych w systemach IoT jest znaczące i stanowi cenny wkład w rozwój dyscypliny naukowej, przyczyniając się jednocześnie do poprawy jakości usług, optymalizacji kosztów i zwiększenia niezawodności systemów IoT.

Wiedza i umiejętności zdobyte w trakcie realizacji pracy są dla autora bardzo cenne w dalszym rozwoju i będą stanowić motywację do dalszych prac, w szczególności w zakresie zastosowania, aplikacji i implementacji niebinarnego kodowania LDPC, które jak dotychczas nie jest szeroko wykorzystywane w istniejących standardach telekomunikacyjnych, a ma znakomity potencjał w sytuacji, gdy duża złożoność obliczeniowa operacji dekodowania jest dopuszczalna. W tym zakresie, najbliższe prace będą koncentrowały się na efektywnych implementacjach niebinarnego dekodera w procesorach graficznych GPU, w powiązaniu z projektowaniem kodów, które potencjalnie będą efektywnie dekodowane w takich układach. Inne kierunki dalszych prac to rozwijanie metod algorytmicznego tworzenia kodów oraz wykorzystanie algorytmów uczenia maszynowego w dekodowaniu kodów o krótkich blokach.

4 Extended summary

4.1 Introduction

This work is a result of research conducted as part of an industrial doctorate, in collaboration with TKH Technology Poland Sp. z.o.o., a member of the TKH Group specializing in the design of devices and systems for the Internet of Things (IoT).

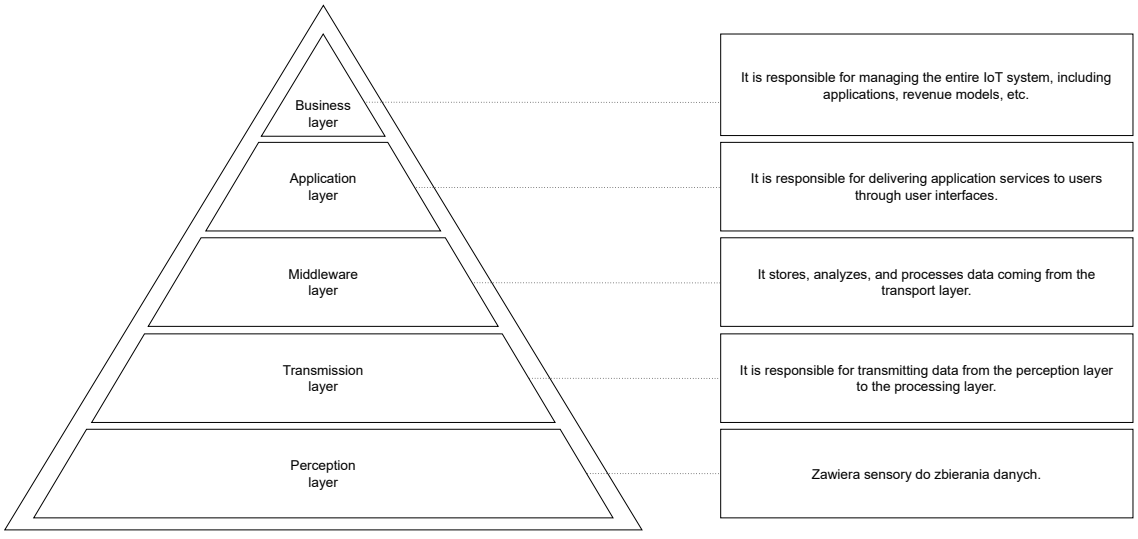
Effective communication links are pivotal in the sophisticated IoT network initiatives carried out at TKH, where a diverse array of solutions and protocols are developed. The conducted research serves as a means to deepen knowledge and potential related to innovative solutions in the area of protocols for IoT, within the context of a wide range of devices and complex projects developed by the company. The outcomes of the work have been implemented in one of the projects, and the publications contribute to the advancement of knowledge regarding the possibilities of effective channel coding in devices with specific characteristics in IoT network nodes.

The concept of the Internet of Things (IoT), progressing towards the Internet of Intelligent Things, stands as a focal point in contemporary discussions and scholarly investigations within both business and scientific research realms. IoT constitutes a network of physical objects equipped with sensors, components, and software enabling interaction with the environment, other objects, and humans, as well as systems facilitating connection and data exchange with other nodes through network infrastructure, including the internet [5, 62]. The spectrum of objects and devices categorized as IoT elements is extensive, but a common characteristic is the presence of an embedded electronic system, typically with a central component in the form of a microcontroller [86].

The creation of a ubiquitous Internet of Things (IoT) ecosystem, comprising diverse objects offering intelligent services, brings various benefits to humanity across domains such as smart homes [132], Industry 4.0 [103], smart cities [61], healthcare [90], agriculture [69], and many others [101]. The consumer market influences the shaping of a new business model, with a significant role played by services provided through IoT-type networks. An essential element for technological progress enabling these transformations is effective communication between devices and systems, requiring appropriately chosen technologies and protocols.

Progress in this field is the result of scientific efforts aimed at improving data transmission techniques. Over time, there is a continuous increase in computational power and memory capacity in electronic devices. This phenomenon is closely related to Moore's Law, formulated by the co-founder of Intel Corporation, Gordon Moore, in 1965, predicting that the number of transistors on an integrated circuit would double approximately every two years [20, 129].

Nonetheless, IoT devices in numerous solutions rely on energy-efficient microcontrollers with relatively limited resources. This is because mass production demands reductions in cost and energy consumption [34, 106]. Many IoT solutions utilize battery power or other power sources with limited efficiency. An important aspect in the design of such devices is the minimization of energy consumption through the use of energy efficient solutions, aiming to prolong battery life or extend the time between charges or battery replacements. This objective can be achieved not only during the design stage of embedded devices but also during the design of algorithms and their implementation in microcontroller systems, including algorithms related to communication protocols. In this context, optimizing the utilization of computational resources and memory is beneficial.



Rysunek 4: Model warstwowy architektury IoT.

The operation of IoT systems can be encapsulated within a layered model [113], such as one comprising five layers, as depicted in Figure 4. The perception layer

represents physical devices and objects, such as environmental sensors collecting data like temperature, pressure, location, movement, or air pollution. The transmission layer is responsible for securely transmitting data and information between devices or from devices to a centralized information processing system [47]. The transmission medium can be wired or wireless, utilizing a wide spectrum of technologies such as 4G mobile networks [35], 5G [67, 27], future 6G [97, 130], WiFi [35], WiMax [72], Bluetooth [48, 66], ZigBee [7, 45], LoRa [30, 8], NB-IoT [19, 140], LTE-Cat-M1 [133, 93], fiber optic transmission [110, 114, 94], and others.

The middleware layer serves as an integral component of the architecture, facilitating the linkage of diverse device types by mediating between various system components. It enables communication, collaboration, and integration among these components. This layer can also encompass the processing of data gathered by IoT devices, both locally and in the cloud, using various data analysis methods, as well as autonomous decision-making and action initiation. The application layer provides global application management, delivering user interfaces and applications for specific IoT use cases. The business layer is responsible for managing the IoT ecosystem, including applications, services, and business model development.

The protocol stack at the transmission layer forms a standard framework for communication, subject to ongoing development to align with technological progress. The possibility of expanding it with new protocols arises from the continual broadening of technnology and research. Furthermore, specialized protocols are developed to fulfill specific requirements. An illustration of this can be found in the implementations within solutions by TKH Technology, the collaborator in this study, where systems are tailored to address a wide range of customer demands. In cases where commonly used communication protocols, such as those mentioned above, meet the client's requirements, they are utilized to ensure communication between IoT devices. However, there are situations where particular assumptions and needs can be more effectively addressed by utilizing dedicated, internally developed solutions at the transmission layer [9].

The use of custom communication protocols in the context of the Internet of Things (IoT) can yield a variety of benefits, stemming from the ability to tailor the protocols used to the specific requirements and conditions of a given project. Some of the benefits include:

- The ability to optimize resource utilization: precise adjustment of communication parameters, such as bandwidth, and control over energy and memory consumption, which is essential for efficient management of limited resources.
- Independence from external standards, allowing for more flexibility in shaping solutions, eliminating potential limitations imposed by standards.
- The ability to optimize link load: dedicated protocols enable the flexible adjustment of packet sizes and transmission speeds, thereby optimizing the load on individual connections and transmission delays.
- The ability to utilize dedicated adaptive techniques to adjust to changing transmission conditions.
- Utilizing link asymmetry involves taking advantage of the varying characteristics of devices within star topology networks. Typically, the central node possesses significantly more computational and energy resources compared to the end node. In such cases, it is possible to employ different transmission techniques and protocols for the uplink and downlink connections, allowing for additional minimization of resource consumption in the embedded device.

The research and deployment efforts presented in this work are related to one of the significant components of dedicated protocols at the transmission layer, namely error correction coding in the communication link. Regardless of the medium or transmission technology used, error detection and/or correction are essential protocol elements. The highly constrained resources of end-device systems in IoT networks suggest the use of relatively straightforward error correction methods, with hard-decision decoding, such as CRC for error detection [104, 13], Hamming codes [51, 18], or potentially slightly more complex block codes: BCH [21, 92] or RS [107, 111].

However, the coducted research findings indicate that employing advanced correction methods, such as LDPC (Low-Density Parity-Check) coding with soft-decision decoding, non-binary LDPC coding with quasi-cyclic matrices, and Raptor coding, can lead to enhanced energy efficiency in both the encoder system implemented in the microcontroller and the transmission system secured by these codes. These methods justify their application by demonstrating potential improvements in energy consumption and overall system performance. In particular, the

aforementioned asymmetry can be utilized, with advanced coding applied only in the uplink direction, thereby reducing the power of signals transmitted by the IoT end device. The decoding operation, which typically involves high complexity, is carried out at the central node, where the limit of computational and energy resources is at a significantly higher level. LDPC codes find application in many advanced communication standards, where codecs are usually implemented in specialized integrated circuits (ICs) or hardware programmable devices (FPGAs). The topic of software implementation in microcontrollers has not been extensively addressed.

Electronic devices, especially those powered by batteries, require efficient energy management strategies to prolong their lifespan. The active state and sleep state are two fundamental operating states of the system that characterize the operation of embedded systems upon which IoT devices are based. In the active state, the device is fully functional and performs tasks, including data collection and processing, as well as preparing data blocks for transmission, incorporating error correction coding. In contrast, in the sleep state, the device enters a state of reduced activity to minimize energy consumption. Utilizing the sleep state is particularly important in battery-powered devices, where long battery life is crucial. Transitions between the active state and the sleep state can be triggered by various factors, such as time schedules, triggering events, or low battery levels. In practice, energy management implementations for IoT devices may vary depending on the specifics of the particular system, but the goal is always to achieve an optimal balance between performance and energy efficiency.

From the perspective of energy efficiency of encoding operations, it is essential to ensure the shortest possible time for preparing the encoded block, so that the required computational activity is minimized. To accomplish this objective, a proposed approach involves utilizing a subclass of quasi-cyclic codes (QC-LDPC) alongside a developed algorithm designed to process bits in blocks. These blocks are no larger than the size of the submatrix of the QC-LDPC code's parity-check matrix. Additionally, the following premises, listed below, were employed in selecting solutions to achieve the stated objectives, which will be presented in the subsequent subsection.

- The appropriate selection of codeword lengths is crucial. Shortening the blocks diminishes the error correction capabilities of the codes. However, it

also facilitates reduced encoding and transmission times, consequently lowering energy consumption. Moreover, messages transmitted in the considered IoT systems are typically relatively short. Therefore, experiments conducted in this work focused on short to medium codeword lengths, ranging from approximately 64b to approximately 1024b.

- The developed encoding scheme should be effective under various transmission conditions, with different levels of interference. Utilizing schemes with incremental redundancy, such as Raptor coding, allows for dynamic adaptation, as well as control over the trade-off between encoding efficiency and error correction capabilities.
- Utilizing dedicated codes tailored for short blocks, such as non-binary LDPC codes, offers increased error correction capabilities or the potential for further energy reduction, contingent upon the development of an efficiently designed encoder implemented in a microcontroller. The high computational complexity of non-binary decoding is not a barrier to applying such schemes in the uplink.
- Optimizing encoding parameters and developing a universal algorithm for creating Raptor-like codes enables the effective implementation of solutions optimized for energy efficiency of transmission systems using short data blocks.
- The reference point for the results of the work includes classical block coding methods (BCH codes and RS codes), as well as the adaptive LDPC coding scheme from the 5G standard.

In the described context, research and deployment work was carried out with the primary goal of proposing an energy efficient implementation of error correction coding algorithms in a disturbed communication channel for IoT devices based on microcontrollers. The decision was made to apply LDPC (Low-Density Parity-Check) coding and decoding methods, binary or non-binary, as well as Rateless coding, known for their effectiveness in error correction in communication systems. The choice of these specific methods was dictated by their ability to effectively reduce data transmission errors while maintaining relatively low computational resource consumption, which is crucial in the context of limited resources available in IoT devices.

4.2 Key Concepts

Linear block code $\mathcal{C}(N, K)$ of length K information vector, length $N > K$ codeword vector, and 2^K codewords is the set of all vectors $\mathbf{c}_{1\times K}$ satisfying the matrix equation $\mathbf{H} \cdot \mathbf{c}^T = \mathbf{0}$ in the field $GF(q)$ (for binary codes $q = 2$), where $\mathbf{H}$ is the **code parity-check matrix**.

LDPC (*Low-Density Parity-Check*) codes are a family of linear block codes in which the parity-check matrix is sparse, i.e., the vast majority of its elements are zeros [87]. LDPC codes can be decoded using one of the **Belief Propagation (BP)** algorithms [87, 42], which are iterative algorithms for propagating approximate probabilities, allowing for the utilization of long block codes with computational complexity linearly dependent on the block length N.

The **code graph** (Tanner graph) is a bipartite graph in which one subset of vertices corresponds to the code's parity-check equations (i.e., rows of matrix $\mathbf{H}$), and the other subset corresponds to the bits of the codeword (columns of matrix $\mathbf{H}$). The Tanner graph defines the message propagation in BP algorithms.

The error-correcting capabilities of codes, including LDPC, are typically described by the dependency of the error probability at the decoder output (estimated by BER – **bit error rate**) as a function of the channel parameter, usually the **signal-to-noise ratio (SNR)** of the **AWGN (Additive White Gaussian Noise)** channel. The error-correcting capabilities of LDPC codes depend on the code block length and the content of the parity-check matrix, particularly the associated structural properties of the Tanner graph, e.g., [52, 53, 68, 71, 17].

Raptor-Like (RL) LDPC codes [25] are a method of coding with incremental redundancy, which combines a base LDPC block code with additional redundancy portions defined by additional low-density parity-check equations. In such a coding scheme, if the initial transmission of the LDPC codeword with high efficiency is not successfully decoded, additional parity blocks are sent. As a result, there is transmission with so-called **incremental redundancy** (IR) [83], where additional redundant portions of symbols are sent after observing errors.

Quasi-Cyclic LDPC (QC-LDPC) codes [41] are a subclass of LDPC codes with a control matrix consisting of square submatrices, where successive rows are cyclic shifts of the first row of the submatrix. Codes used in the 5G standard [4] can be classified as a family of binary QC-LDPC codes of the Raptor-Like type. QC-LDPC

codes are characterized by the possibility of efficient hardware implementation of iterative decoders [82, 100, 128].

Nonbinary LDPC codes (NB-LDPC) [29] are a generalization of binary LDPC codes, where the parity-check matrix and control equation are defined over a field $GF(q > 2)$. This allows for increased error correction capabilities, especially for relatively short to medium block lengths (from tens to hundreds of symbols) [29, 33, 17], at the cost of significantly increased decoding computational complexity [31, 75].

Hamming codes [51, 18] are linear block codes that enable the detection and correction of single-bit errors. **BCH (Bose-Chaudhuri-Hocquenghem) codes** [21, 92] are a family of error-correcting block codes based on polynomials over a Galois field, offering flexibility in adjusting error-correction capability to specific communication system requirements. **RS (Reed-Solomon) codes** [107, 111] are a family of nonbinary linear block codes. The application of BCH and RS codes spans various domains, from data storage on magnetic and flash media to satellite communication transmission.

4.3 Research Thesis, Objective, and Scope

In the presented context, in connection with the set of the goals outlined below, the following thesis has been formulated.

In the case of LDPC encoder implementations in microprocessor systems, it is possible to indicate the encoder's construction and the class of codes resulting in particularly energy-efficient data transmission systems. These implementations belong to the most efficient within the entire class of block codes, and their efficiency can be further enhanced by employing Raptor coding techniques.

The proposed thesis is related to the main objectives of the project, which were as follows:

- to reduce energy consumption in embedded systems within IoT devices utilizing, custom communication protocols in noisy wireless or wired links by employing modern iteratively decoded codes: binary QC-LDPC, non-binary QC-LDPC, for transmission security instead of classical block codes.
- to ensure effective and energy-efficient error correction of transmission errors in IoT networks utilizing custom communication protocols and diverse communication media, particularly in communication between nodes of asymmetric nature: from end nodes with relatively limited computational resources to gateways with significantly greater computational and energy resources.

To achieve the stated objectives and prove the proposed thesis, the following research and implementation work was planned and executed:

- development of software models for LDPC and NB-LDPC encoding and decoding systems, channels, and other elements of the numerical environment enabling statistical determination of the correction capabilities of the investigated transmission systems;
- implementation of LDPC and non-binary LDPC encoders in the CPUs of typical modern microcontrollers, competitive with typical block coding methods (e.g., BCH, RS) in terms of speed and energy consumption;
- creation of experimental parity check matrices for the subclass of efficiently encoded (Quasi-Cyclic – QC) codes of various block sizes and efficiencies,

enabling the transmission of relatively short messages characteristic to the IoT networks;

- development and experimental verification of an encoding system for links with unknown and time-varying parameters: utilizing incremental redundancy (IR – *Incremental Redundancy*) and Raptor-like coding (RL – *Raptor-Like*), with efficiently encoded LDPC and RL-QC-LDPC matrices;
- experimental research: implementation of an efficient encoder in the CPU, determination of its time parameters and energy consumption; estimation of the overall energy consumption by the encoder and radio transmitter; comparison of energy consumption per transmitted bit in LDPC encoding systems with constant efficiency to systems with incremental redundancy and Raptor-Like coding;
- determining the potential for achieving additional energy savings in the transmitter system for links with unknown and time-varying parameters, for the proposed codes and implementations, as well as standard 5G codes implemented in the microcontroller's CPU;
- proposing a method for creating non-binary codes, NB QC-RL-LDPC, with short blocks and comparing them with binary coding of the 5G standard, utilizing a system with incremental redundancy;
- implementation of a non-binary LDPC encoder in a microcontroller system and experimental determination of its time parameters and energy consumption.
- conducting experimental research and obtaining: error rate/energy consumption per unit in the transmission system/ effective throughput charts for created short block codes, both binary and non-binary LDPC, including Raptor-like codes.

4.4 Research Work and Research Findings

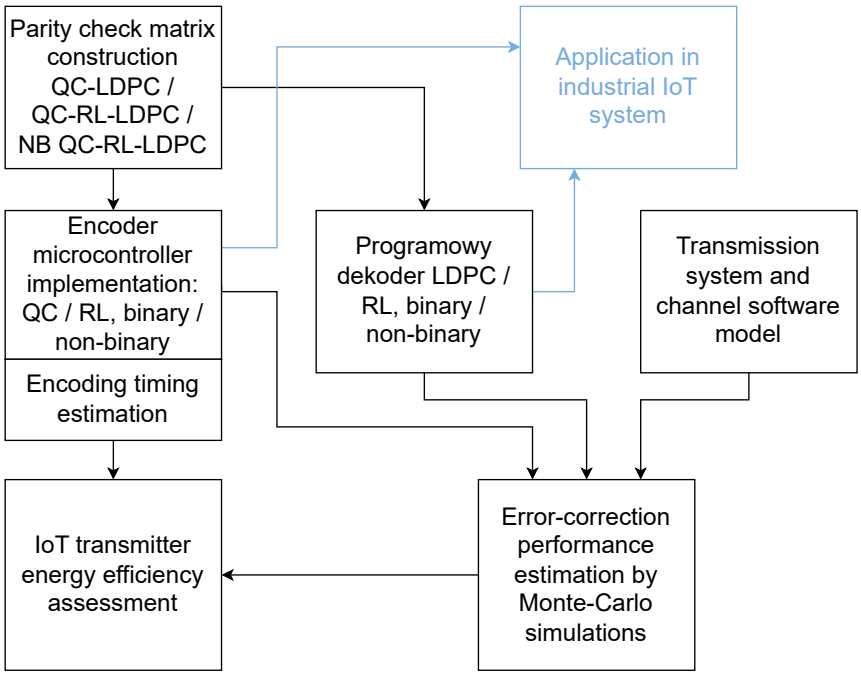
The conducted work utilized modern microcontroller units from the STM32 family to implement various types of codes, such as LDPC, QC-LDPC, NB LDPC, and Raptor codes. The developed software solutions can be easily adapted to embedded systems based on different hardware with minimal effort. Measurements were carried out for a range of parameters, including encoding/decoding time, memory usage, and energy consumption. For measurements, appropriately selected measuring equipment was used, including high-precision multimeters for current measurement, oscilloscope, logic state analyzer, as well as a programming environment enabling debugging of developed solutions. These tools facilitated precise evaluation of various parameters and analysis of the behavior of microcontroller units during the implementation of encoders and decoders.

The second component of the research toolkit consisted of software models prepared in the Matlab environment, encompassing encoders and decoders for various code types such as LDPC (both binary and non-binary) and Raptor. They allowed for the analysis of the error correction capabilities of codes depending on the level of disturbances in the channel, as well as their utilization for preparing an environment enabling simulations of various conditions and the possibility of result analysis.

The models enabled the determination of the error correction capabilities of codes depending on the level of disturbances in the channel. The fusion of measurement results: energy consumption, encoding time, and error correction capabilities allowed for the estimation of the transmitter energy efficiency.

The work began with a review of existing error correction coding methods, particularly focusing on their applicability in IoT systems. Codes with the best error correction properties for short block lengths and acceptable encoding complexity were considered. Subsequently, attention was focused on LDPC (Low-Density Parity-Check) codes. They are recognized as one of the best error correction methods in terms of correction performance and are utilized in significant radio communication standards such as WiFi, WiMAX, DVB-T2, DVB-S2, and 5G mobile networks. Additionally, non-binary LDPC codes exhibit excellent error correction capabilities for short blocks. The planned work initially involved analyzing and reviewing the characteristic features of LDPC code subclasses, particularly QC-LDPC (Quasi-Cyclic LDPC), a subclass enabling efficient implementation. A

software model of the encoder, channel, and decoder was developed, enabling experimental determination of the coding system's error rate. The review of codes used in communication standards and their experimentally verified error correction capabilities is presented in **Publication 1**.



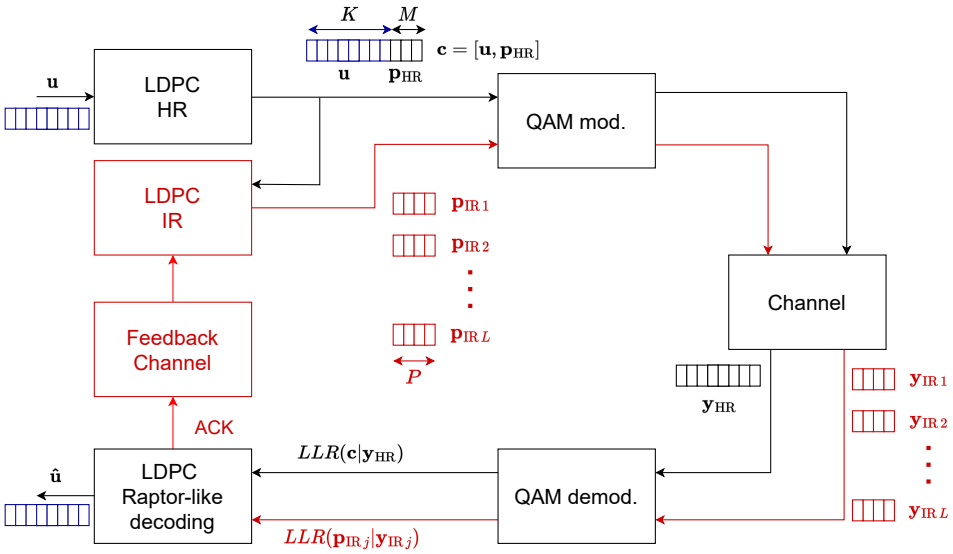
Rysunek 5: Summary of the Research and Implementation Methodology.

In the next stage, an attempt was made to demonstrate that a scheme based on LDPC coding can be successfully implemented in the transmitting device of a system with severely limited computational resources. In **Publication 2**, an implementation in an IoT device based on a microcontroller was presented, along with the results of experimental studies. The developed implementation is intended for an Internet of Things (IoT) system, where a low-power end device encoded messages are transmitted to the gateway. An efficient LDPC coding algorithm by Richardson and Urbanke was utilized, and it was shown how elementary operations of this algorithm could be vectorized for the QC-LDPC code subclass in a software implementation and programmed as cyclic shift and Galois field addition operations (GF(2)) directly corresponding to CPU instructions. Additionally, an efficient parity matrix representation and memory organization for storing auxiliary computation results were presented.

Experimental results demonstrate a significant increase in efficiency in terms of memory usage and encoding time compared to coding using a direct representation of the parity check matrix. The article analyzed the computational complexity of the encoder and discussed the challenge of processing a large amount of data by devices with limited computational capabilities. **Publication 2** includes experimental comparisons with other well-known block codes, such as Reed-Solomon (RS) and Bose-Chaudhuri-Hocquenghem (BCH), showing that memory requirements for LDPC codes are not greater than for standard block codes. At the same time, the possibility of achieving shorter encoding time than with traditional block codes created using a generator matrix was indicated, which also leads to reduced energy consumption. Furthermore, the benefits of LDPC coding have been experimentally verified to be greater when using a soft-decision decoder compared to the mentioned block-size-comparable BCH and RS codes. This study contributes to expanding the applications of LDPC coding ito the resource-limited systems, presenting its advantages in terms of efficiency and error correction compared to traditional block codes implemented in software. The outcomes resulting from the efficient implementation of the coding algorithm show great promise, aligning well with the trend observed in battery-powered IoT devices. One of the main benefits is achieving significant energy savings. The solution was utilized in an IoT device based on a microcontroller with limited computational capabilities.

The proposed concept of the system, with LDPC coding in the uplink direction, does not require the construction of a decoder in software for systems with severely limited resources. Nevertheless, as part of complementary work, an implementation of an iterative LDPC decoder in a microcontroller system was also developed. This part of the work is described in **Publication 3**. The results regarding memory usage, decoding time of an information block, decoder throughput, and decoding performance as a function of the number of errors in a discrete channel with interference were presented. It was shown that it is possible to implement LDPC and QC-LDPC decoder, including irregular ones, in a microcontroller system, despite significant computational requirements for LDPC codes. The achievable throughput under full load of a typical microcontroller was indicated, and an approximate formula for calculating the throughput in such system was proposed.

One of the primary goals of the conducted research was to propose coding schemes with optimized energy efficiency, in terms of the energy required per



Rysunek 6: Proposed Uplink Communication Model for the Internet of Things (IoT) Utilizing an Error Correction Scheme with Incremental Redundancy. [**Publication 4** – Binary Codes, **Publication 6** – Non-binary Codes]

information unit (1 bit) to be transmitted. In **Publication 4**, it was demonstrated that increased energy efficiency in transmission can be achieved compared to QC-LDPC encoding with constant efficiency, by employing Raptor-like coding (RL - Raptor Like), based on QC-LDPC base codes (QC-RL-LDPC).

The simplified diagram of such a system is illustrated in Fig. 6. The efforts towards the binary Raptor-like coding scheme, outlined in **Publication 4**, involved several key steps. These steps included implementing a QC-RL-LDPC encoder on a microcontroller, proposing an innovative incremental decoding approach utilizing the LDPC BP decoding algorithm (Fig. 5 in **Publication 4**), refining existing methods for constructing LDPC codes into QC-RL-LDPC codes, and conducting experimental studies. These studies revolved around assessing the correction effectiveness in relation to the observed bit throughput (goodput), considering varying signal-to-noise ratios (SNR), and evaluating the energy efficiency of the transmitter device. This evaluation included measuring the unit energy consumption by the encoder and transmitter. Additionally, a comparison of normalized energy consumption statistics was carried out, contrasting it with the scenario of LDPC coding of fixed length.

In the calculations of the expected energy consumption, the results of energy consumption measurements by a typical microcontroller performing encoding

operations and encoding time measurements were utilized. It was assumed that the microcontroller is only active during encoding. Subsequently, the required transmission power in a typical system was estimated, taking into account experimental results related to the bit error rate as a function of the signal-to-noise ratio in the channel.

The results of experiments, including those shown in Fig. **??**, indicate that:

- in comparison to fixed-rate coding, achieving lower unit energy consumption is possible for high signal-to-noise ratio (SNR) levels,
- effective transmission is possible for low SNR levels, where the error correction capabilities of fixed-rate coding are no longer effective,
- the developed codes are competitive with those used in the 5G standard employing a similar rateless redundancy scheme.

In the next stage of the research work, the developed Raptor coding solutions were expanded to non-binary (NB) coding, used as the base coding, as well as for coding incremental parity equations. A software implementation of the NB-LDPC encoder was developed, using an efficient coding algorithm, which can also be successfully utilized for non-binary Raptor codes, and such an encoder was implemented in a microcontroller. These efforts were described in **Publication 5**. Significant aspects related to the performance of the coding algorithm were emphasized, suggesting that effective deployment of non-binary codes could bring substantial benefits, especially for battery-powered IoT devices. The conclusions drawn from the conducted research indicate promising prospects for the application of non-binary codes in the uplink of IoT systems, due to the good energy efficiency of the transmitter unit. In the experiments, a comparison of time dependencies for codes over different orders of Galois fields (GF) was conducted, allowing for an examination of the impact of code selection on potential energy consumption.

Within the work on non-binary Raptor coding, a method for mapping non-binary symbols to QAM modulation constellations of any order was also developed, as well as a software decoder based on the expansion of binary Raptor code decoders, and an algorithmic method for constructing NB QC-RL-LDPC codes. These efforts and the obtained results are included in the manuscript identified as **Publication 6**. A procedure for constructing short NB QC-RL-LDPC codes

tailored to efficient implementation of low-complexity encoders and optimized code graph structures was proposed. The proposed method is based on optimizing the base code graph by minimizing the number of cycles and then determining the positions of symbols in additional parity equations for incremental redundancy, supported by Monte Carlo simulations. Numerical experiments confirmed the effectiveness of the algorithm. The obtained non-binary codes, along with the developed decoding algorithm, exhibit increased error correction efficiency compared to binary 5G coding with a similar incremental redundancy scheme. It was also shown that differentiating QAM modulation orders through a combination of higher-order modulation, for example, ($X = 64$)-QAM for initial transmission, with lower-order modulation, for example, ($X_{\mathrm{IR}} = 4$)-QAM for incremental redundancy, can be advantageous in terms of performance at low SNR values (below 0 dB and slightly above). The flexibility of combining LDPC codes over GF of any order with modulations of any order ensures broad adaptability, offering a wide range of possible spectral efficiencies. The coding gain obtained can directly translate into energy gain in the transmitter system by enabling transmission with reduced power.

4.5 Summary

Developing error correction coding systems that allow for effective error correction with the best possible energy efficiency in embedded systems with limited computational resources is a non-trivial task, which was undertaken as part of research and implementation efforts described in the attached series of publications.

The implementation of various code types, including LDPC, QC-LDPC, NB LDPC, and Raptor codes, on modern microcontrollers enabled the analysis of their energy efficiency and performance under conditions of limited computational resources. The research involved measurements of encoding and decoding time, memory usage, and energy consumption, allowing for a precise assessment of the parameters of error correction coding solutions in embedded systems. Software models were developed in the Matlab environment, encompassing diverse code types, enabling the analysis of their error correction capabilities depending on the level of channel interference. These models also allowed for the simulation of various transmission conditions and the analysis of research results.

The presented solutions and results represent a significant contribution to the development of LDPC coding techniques in the context of IoT device applications. A comprehensive description of effective implementations of LDPC, QC-LDPC, QC-RL-LDPC, NB-LDPC codes, as well as procedures for constructing such codes and effective decoding in systems with incremental redundancy and LDPC coding, expands the understanding of the possibilities of utilizing modern error correction coding methods in the IoT environment, where resources are limited, and energy efficiency requirements are high. Implementing encoder and decoder in IoT devices represents a step forward in the development of software-based error correction coding solutions. Consequently, it becomes possible to leverage the advantages of LDPC codes in IoT applications, even in the case of devices with limited computational and energy resources.

Adaptive Raptor coding, for which implementation and construction solutions have been developed and refined, represents a method that allows for flexible adaptation to changes in transmission conditions, which can be extremely advantageous in the dynamic IoT environment. Introducing non-binary LDPC coding and non-binary Raptor coding may provide additional benefits associated with increased correction capabilities, especially for relatively short code blocks, which are characteristic in IoT systems. Analyzing efficiency under conditions of limited

computational resources is a significant step in adapting this technology to the specific requirements of embedded systems. Consequently, it becomes possible to leverage the advantages of LDPC codes in IoT applications.

Kluczowe osiągnięcia i wyniki, przedstawione w cyku publikacji, w kontekście sformułowanej tezy oraz celów pracy, w mniemaniu autora są następujące:

- **Publication 2** describes an original solution for the efficient implementation of a binary QC-LDPC encoder on a microcontroller and presents experimental results demonstrating a significant reduction in energy consumption per unit compared to RS, BCH, and LDPC encoders without computation vectorization.

- **Publication 4** describes an original solution for the efficient implementation of an encoder in the incremental redundancy scheme, QC-RL-LDPC. It also introduces an original decoding algorithm for QC-RL-LDPC codes and a simulation environment for RL encoding. The experimental results presented indicate the possibility of achieving reduced energy consumption per unit compared to coding without efficiency adaptation and the ability to transmit at extremely low signal-to-noise ratio levels.

- In **Publication 5**, a method for encoding non-binary LDPC codes over fields $GF(2^2)$ to $GF(2^8)$ was generalized, demonstrating that non-binary LDPC encoding can also be effectively implemented in a microcontroller system.

- In **Publication 6**, an original algorithmic method for constructing non-binary QC-RL-LDPC codes tailored for efficient encoder implementation was proposed. The incremental decoding algorithm was generalized for non-binary codes, and results were presented demonstrating increased correction efficiency compared to binary LDPC coding standards used in 5G, potentially enabling a reduction in transmitted signal energy.

The indicated key results confirm the validity of the formulated thesis, and the set goals of the work have been achieved. The attainment of additional coding gain through the constructed non-binary QC-RL-LDPC codes compared to 5G codes, along with demonstrating efficient encoder implementation, illustrates that non-binary codes, when appropriately utilized, have the potential to further enhance energy efficiency.

The developed solutions and implementations have been partially deployed as a significant component of projects conducted by the TKH Group team related to the Internet of Things and hold potential for further utilization in other projects and product enhancements, particularly concerning energy efficiency. They open new perspectives for both research and development activities and business endeavors. The achieved increase in data transmission efficiency in IoT systems is significant and constitutes a valuable contribution to the advancement of the scientific discipline, while simultaneously contributing to improving service quality, cost optimization, and increasing the reliability of IoT systems.

The knowledge and skills acquired during the course of this work are highly valuable to the author's further development and will serve as motivation for future endeavors, particularly in the realm of application, utilization, and implementation of non-binary LDPC coding, which has not been widely employed in existing telecommunication standards but holds significant potential in situations where high computational complexity of decoding operations is permissible. In this regard, forthcoming efforts will focus on efficient implementations of non-binary decoders on graphics processing units (GPUs), coupled with the design of codes that could potentially be effectively decoded on such platforms. Other directions for future work include advancing methods for algorithmic code generation and leveraging machine learning algorithms in decoding short-block codes.

Literatura

[1] Generic binary bch encoding/decoding library. available online: https://github.com/parrot-developers/bch.

[2] RSCODE project. `http://rscode.sourceforge.net`.

[3] 5G; NR; Multiplexing and channel coding (3GPP TS 38.212 version 15.2.0 Release 15), 2018. Version 15.2.0 Release 15.

[4] 5G; NR; Multiplexing and channel coding (3GPP TS 38.212 version 16.2.0 Release 16), 2020. Version 16.2.0 Release 16.

[5] A. Aagaard, M. Presser, and T. Andersen. Applying iot as a leverage for business model innovation and digital transformation. In *2019 Global IoT Summit (GIoTS)*, pages 1–5, 2019.

[6] K. M. Al-Obaidi, M. Hossain, N. A. M. Alduais, H. S. Al-Duais, H. Omrany, and A. Ghaffarianhoseini. A review of using iot for energy efficient buildings and cities: A built environment perspective. *Energies*, 15(16), 2022.

[7] A. I. Ali, S. Z. Partal, S. Kepke, and H. P. Partal. Zigbee and lora based wireless sensors for smart environment and iot applications. In *2019 1st Global Power, Energy and Communication Conference (GPECOM)*, pages 19–23, 2019.

[8] L. Angrisani, P. Arpaia, F. Bonavolontà, M. Conti, and A. Liccardo. Lora protocol performance assessment in critical noise conditions. In *2017 IEEE 3rd International Forum on Research and Technologies for Society and Industry (RTSI)*, pages 1–5, 2017.

[9] ANTHONY, A. KUMAR, J. HUSSAIN, and CHUN. *Connecting the Internet of Things: IoT Connectivity Standards and Solutions*. APRESS, 2023.

[10] E. Arikan. Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels. *IEEE Trans. Inf. Theory*, 55:3051–3073, July 2009.

[11] J. H. Bae, A. Abotabl, H.-P. Lin, K.-B. Song, and J. Lee. An overview of channel coding for 5g nr cellular communications. *APSIPA Transactions on Signal and Information Processing*, 8:e17, 2019.

[12] S. Belhadj and M. L. Abdelmounaim. On error correction performance of ldpc and polar codes for the 5g machine type communications. In *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, pages 1–4, 2021.

[13] P. S. Bere and M. Z. A. Khan. Low complexity, diversity preserving hard decision decoder for crc codes with iot applications. In *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*, pages 1–5, 2022.

[14] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes. In *Proc. (IEEE) International Conference on Communications*, pages 1064–1070, Geneva, Switzerland, May 1993.

[15] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes. In *Proc. (IEEE) International Conference on Communications*, pages 1064–1070, Geneva, Switzerland, 1993.

[16] I. E. Bocharova, B. Kudryashov, and R. Johannesson. Searching for Binary and Nonbinary Block and Convolutional LDPC Codes. *IEEE Trans. Inf. Theory*, 62:163–183, January 2016.

[17] I. E. Bocharova, B. D. Kudryashov, E. P. Ovsyannikov, V. Skachek, and T. Uustalu. Design and analysis of nb qc-ldpc codes over small alphabets. *IEEE Trans. Commun.*, 70(5):2964–2976, 2022.

[18] Z. BOHUSLAVEK and I. MASIK. Error characteristics and their prediction in zigbee transmission at coexistence conditions. 16, Mar 2017.

[19] R. Boisguene, S.-C. Tseng, C.-W. Huang, and P. Lin. A survey on nb-iot downlink scheduling: Issues and potential solutions. In *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 547–551, 2017.

[20] P. Bondyopadhyay. Moore's law governs the silicon revolution. *Proceedings of the IEEE*, 86(1):78–81, 1998.

[21] R. C. Bose, D. K. Ray-Chaudhuri, and A. Hocquenghem. A class of error-correcting codes in binary space and their applications. *IEEE Transactions on Information Theory*, 6(2):459–470, 1960.

[22] E. Boutillon, L. Conde-Canecia, and A. A. Ghouwayel. Design of a GF(64)-LDPC Decoder Based on the EMS Algorithm. *IEEE Trans. Circuits Syst. I*, 60:2644–2656, October 2013.

[23] J. Chen, A. Dholakia, E. Eleftheriou, M. Fossorier, and X.-Y. Hu. Reduced-complexity decoding of ldpc codes. *IEEE Transactions on Communications*, 53(8):1288–1299, 2005.

[24] J. Chen and M. P. C. Fossorier. Density Evolution for Two Improved BP-Based Decoding Algorithms of LDPC Codes. *IEEE Communications Letters*, 6, No. 5:208–210, May 2002.

[25] T.-Y. Chen, K. Vakilinia, D. Divsalar, and R. D. Wesel. Protograph-Based Raptor-Like LDPC Codes. *IEEE Trans. Commun.*, 63(5):1522–1532, May 2015.

[26] X. Chen and C.-L. Wang. High-Throughput Efficient Non-Binary LDPC Decoder Based on the Simplified Min-Sum Algorithm. *IEEE Trans. Circuits Syst. I*, 59:2784–2794, November 2012.

[27] L. Chettri and R. Bera. A comprehensive survey on internet of things (iot) toward 5g wireless systems. *IEEE Internet of Things Journal*, 7(1):16–32, 2020.

[28] C. Condo. VLSI Implementation of a Multi-ModeTurbo/LDPC Decoder Architecture. *IEEE Trans. Circuits Syst. I*, 60:1441–1454, 2013.

[29] M. C. Davey and D. MacKay. Low-Density Parity Check Codes over GF(q). *IEEE Commun. Lett.*, 2:165–167, June 1998.

[30] J. de Carvalho Silva, J. J. P. C. Rodrigues, A. M. Alberti, P. Solic, and A. L. L. Aquino. Lorawan — a low power wan protocol for internet of things: A review and opportunities. In *2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*, pages 1–6, 2017.

[31] D. Declercq and M. Fossorier. Decoding Algorithms for Nonbinary LDPC Codes Over GF(q). *IEEE Trans. Commun.*, 55:633–643, April 2007.

[32] D. Divsalar, S. Dolinar, C. R. JoJones, and K. Andrews. Capacity-Approaching Protograph Codes. *IEEE J. Sel. Areas Commun.*, 27:876–888, August 2009.

[33] L. Dolecek, D. Divsalar, Y. Sun, and B. Amiri. Non-Binary Protograph-Based LDPC Codes: Enumerators, Analysis, and Designs. *IEEE Trans. Inf. Theory*, 60:3913–3941, July 2014.

[34] V. A. Dubal and Y. S. Rao. A low-power high-performance sensor node for internet of things. In *IEEE Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 607–612, Madurai, India, June 2018.

[35] A. Elnashar and M. A. El-saidny. *IoT Evolution Towards a Super-connected World*, pages 310–381. 2018.

[36] U. Erez, M. D. Trott, and G. W. Wornell. Rateless coding for gaussian channels. *IEEE Trans. Inf. Theory*, 58(2):530–547, February 2012.

[37] ETSI. Digital video broadcasting (dvb); second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications (dvb-s2). european standard etsi en 302–307 v1.4.1. 2014.

[38] ETSI Standard: EN 302 307 v1.1.1, *Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications*, 2005.

[39] Y. Fang, G. Bi, Y. L. Guan, and F. C. M. Lau. A Survey on Protograph LDPC Codes and Their Applications. *IEEE Commun. Surveys Tuts.*, 17:1989–2016, 2015.

[40] O. Ferraz, S. Subramaniyan, R. Chinthala, J. Andrade, J. R. Cavallaro, S. K. Nandy, V. Silva, X. Zhang, M. Purnaprajna, and G. Falcao. A survey on high-throughput non-binary ldpc decoders: Asic, fpga, and gpu architectures. *IEEE Commun. Surveys Tuts.*, 24(1):524–556, 2022.

[41] M. P. C. Fossorier. Quasi-Cyclic Low-Density Parity-Check Codes from Circulant Permutation Matrices. *IEEE Trans. Inf. Theory*, 50, No. 8:1788–1793, August 2004.

[42] M. P. C. Fossorier, M. Mihaljevic, and H. Imai. Reduced Complexity Iterative Decoding of Low-Density Parity Check Codes Based on Belief Propagation. *IEEE Transactions on Communications*, 47, No. 5:673–680, May 1999.

[43] R. Gallager. Low-density parity-check codes. *IRE Transactions on Information Theory*, 8(1):21–28, 1962.

[44] R. G. Gallager. Low-Density Parity-Check Codes. *IRE Transactions on Information Theory*, IT-8:21–28, January 1962.

[45] V.-D. Gavra and O. A. Pop. Usage of zigbee and lora wireless technologies in iot systems. In *2020 IEEE 26th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, pages 221–224, 2020.

[46] M. Gholami and G. Raeisi. Large Girth Column-Weight Two and Three LDPC Codes. *IEEE Commun. Lett.*, 18:1671–1674, October 2014.

[47] A. Glória, F. Cercas, and N. Souto. Comparison of communication protocols for low cost internet of things devices. In *2017 South Eastern European Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, pages 1–6, 2017.

[48] S. Gowrishankar, N. Madhu, and T. G. Basavaraju. Role of ble in proximity based automation of iot: A practical approach. In *2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, pages 400–405, 2015.

[49] P. Hailes, L. Xu, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo. A survey of fpga-based ldpc decoders. *IEEE Commun. Surveys Tuts.*, 18:1098–1122, 12 2015.

[50] P. Hailes, L. Xu, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo. A survey of fpga-based ldpc decoders. *IEEE Commun. Surveys Tuts.*, 18:1098–1122, 12 2016.

[51] R. W. Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147–160, 1950.

[52] X. Y. Hu, E. Eleftheriou, and D. M. Arnold. Regular and Irregular Progressive Edge-Growth Tanner Graphs. *IEEE Trans. Inf. Theory*, 51:386–398, 2005.

[53] J. Huang, L. Liu, W. Zhou, and S. Zhou. Large-Girth Nonbinary QC-LDPC Codes of Various Lengths. *IEEE Trans. Commun.*, 58:3436–3447, December 2010.

[54] J. Huang, S. Zhou, and P. Willett. Nonbinary LDPC Coding for Multicarrier Underwater Acoustic Communication. *IEEE J. Sel. Areas Commun.*, 26:1684–1696, December 2008.

[55] J. Hyla, W. Sułek, W. Izydorczyk, L. Dziczkowski, and W. Filipowski. Efficient ldpc encoder design for iot-type devices. *Applied Sciences*, 12(5), 2022.

[56] J. Hyla, W. Sułek, W. Izydorczyk, L. Dziczkowski, and W. Filipowski. Efficient ldpc encoder design for iot-type devices. *Applied Sciences*, 12(5), 2022.

[57] IEEE Standard: IEEE P802.11n=D10. Draft IEEE Standard for Local Metropolitan Networks – Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC), and Physical Layer (PHY) specifications: Enhancements for Higher Throughput, March 2006.

[58] IEEE Standard: IEEE P802.16. Local and Metropolitan Area Networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems, 2006.

[59] 2016. IEEE 802.11-2016. IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

[60] IEEE Std 802.16e, *IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems. IEEE Std 802.16e*, 2006.

[61] M. Ilyas. Iot applications in smart cities. In *2021 International Conference on Electronic Communications, Internet of Things and Big Data (ICEIB)*, pages 44–47, 2021.

[62] M. Ishaq, M. H. Afzal, S. Tahir, and K. Ullah. A compact study of recent trends of challenges and opportunities in integrating internet of things

(iot) and cloud computing. In *2021 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*, pages 1–4, 2021.

[63] J. Izydorczyk, W. Sułek, and P. Zawadzki. *Kody i szyfry*. Wydawnictwo Politechniki Śląskiej, 2017.

[64] S. Jayasooriya, M. Shirvanimoghaddam, L. Ong, and S. J. Johnson. Analysis and design of raptor codes using a multi-edge framework. *IEEE Trans. Commun.*, 65(12):5123–5136, 2017.

[65] S. Jayasooriya, M. Shirvanimoghaddam, L. Ong, and S. J. Johnson. Raptor codes for higher-order modulation using a multi-edge framework. *IEEE Wireless Commun. Lett.*, 7(1):110–113, 2018.

[66] K. E. Jeon, J. She, P. Soonsawad, and P. C. Ng. Ble beacons for internet of things applications: Survey, challenges, and opportunities. *IEEE Internet of Things Journal*, 5(2):811–828, 2018.

[67] S. Kar, P. Mishra, and K.-C. Wang. 5g-iot architecture for next generation smart systems. In *2021 IEEE 4th 5G World Forum (5GWF)*, pages 241–246, 2021.

[68] M. Karimi and A. H. Banihashemi. On the Girth of Quasi-Cyclic Protograph LDPC Codes. *IEEE Trans. Inf. Theory*, 59:4542–4552, July 2013.

[69] R. Karthiga, C. L. B. Devi, R. Janaki, C. Gayathri, V. S. Pandi, and D. Shobana. Iot farm: A robust methodology design to support smart agricultural system using internet of things with intelligent sensors association. In *2023 7th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 1332–1337, 2023.

[70] C. A. Kelley and S. Deepak. Pseudocodewords of Tanner Graphs. *IEEE Transactions on Information Theory*, 53, No. 11:4013–4038, November 2007.

[71] I. Kim and H.-Y. Song. Some new constructions of girth-8 qc-ldpc codes for future gnss. *IEEE Commun. Lett.*, 25(12):3780–3784, 2021.

[72] A. Kumar, D. Saxena, N. Sharma, and A. Gankotiya. Shorting pin and srr loaded multi-band antenna for 4g/5g/wi-fi/wimax and iot applications. In *2023 International Conference on Artificial Intelligence and Smart Communication (AISC)*, pages 303–307, 2023.

[73] A. Lazarenko. Fpga design and implementation of dvb-s2/s2x ldpc encoder. In *IEEE International Conference on Electrical Engineering and Photonics*, pages 98–102, St. Petersburg, Russia, 2019.

[74] H. D. Le, V. V. Mai, C. T. Nguyen, and A. T. Pham. Throughput analysis of incremental redundancy hybrid arq for fso-based satellite systems. In *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, pages 1–5, Honolulu, Hawaii, USA, September 2019.

[75] T. Lehnigk-Emden and N. Wehn. Complexity Evaluation of Non-binary Galois Field LDPC Code Decoders. In *Proc. (IEEE) 6th International Symposium on Turbo Codes & Iterative Information Processing*, pages 53–57, Brest, France, September 2010.

[76] H. Li, B. Bai, X. Mu, J. Zhang, and H. Xu. Algebra-assisted construction of quasi-cyclic ldpc codes for 5g new radio. *IEEE Access*, 6:50229–50244, 2018.

[77] J. Li, K. Liu, S. Lin, and K. Abdel-Ghaffar. Algebraic Quasi-Cyclic LDPC Codes: Construction, Low Error-Floor, Large Girth and a Reduced-Complexity Decoding Scheme. *IEEE Trans. Commun.*, 62:2626–2637, August 2014.

[78] M. Li, V. Derudder, K. Bertrand, C. Desset, and A. Bourdoux. High-Speed LDPC Decoders Towards 1 Tb/s. *IEEE Trans. Circuits Syst. I*, 68:2224–2233, 5 2021.

[79] S. Liao, Y. Zhan, Z. Shi, and L. Yang. A high throughput and flexible rate 5g nr ldpc encoder on a single gpu. In *IEEE International Conference on Advanced Communications Technology (ICACT)*, pages 29–34, Virtual, February 2021.

[80] E. A. Likhobabin, A. A. Ovinnikov, R. S. Goriushkin, P. B. Nikishkin, and E. I. Khokhryakov. High throughput fpga implementation of ldpc decoder architecture for dvb-s2x standard. In *2022 International Conference on Information, Control, and Communication Technologies (ICCT)*, Nanjing, China, November 2022.

[81] C.-F. Lin. Ufmc-based underwater voice transmission scheme with ldpc codes. *Applied Sciences*, 11:art. no. 1818, 2021.

[82] J. Lin, J. Sha, Z. Wang, and L. Li. Efficient Decoder Design for Nonbinary Quasicyclic LDPC Codes. *IEEE Trans. Circuits Syst. I*, 57:1071–1082, May 2010.

[83] S. Lin and D. J. Costello, Jr. *Error Control Coding: Fundamentals and Applications, 2nd Edition*. Prentice-Hall, Inc., Upper Saddle River, New Jersey 07458, 2004.

[84] B. Liu, R. Liu, Z. Liu, and L. Zhao. An efficient implementation of ldpc decoders on arm processors. In *2018 IEEE International Workshop on Signal Processing Systems (SiPS)*, pages 1–5, 2018.

[85] H. Luo, Y. Zhang, W. Li, L.-K. Huang, J. Cosmas, D. Li, and C. Maple. Low latency parallel turbo decoding implementation for future terrestrial broadcasting systems. *IEEE Trans. Broadcast.*, 64:96–104, 3 2018.

[86] Z. Ma, M. Xiao, Y. Xiao, Z. Pang, H. V. Poor, and B. Vucetic. High-reliability and low-latency wireless communication for internet of things: Challenges, fundamentals, and enabling technologies. *IEEE Internet of Things Journal*, 6(5):7946–7970, 2019.

[87] D. J. C. MacKay. Good Error-Correcting Codes Based on Very Sparse Matrices. *IEEE Trans. Inf. Theory*, 45:399–431, March 1999.

[88] D. J. C. MacKay and R. M. Neal. Near Shannon Limit Performance of Low Density Parity Check Codes. *Electronics Letters*, 32:1645–1646, August 1996.

[89] A. Mahdi and V. Paliouras. A low complexity-high throughput qc-ldpc encoder. *IEEE Trans. Signal Process.*, 62(10):2696–2708, 2014.

[90] T. Malapane, W. Doorsamy, and B. Paul. An intelligent iot-based health monitoring system. In *2020 International Conference on Intelligent Data Science Technologies and Applications (IDSTA)*, pages 95–100, 2020.

[91] M. Marques da Silva, R. Dinis, and G. Martins. On the performance of ldpc-coded massive mimo schemes with power-ordered noma techniques. *Applied Sciences*, 11:art. no. 8684, 2021.

[92] D. Merget, G. Smietanka, and J. Goetze. Uhf rfid transmission with soft-input bch decoding. In *2014 IEEE World Forum on Internet of Things (WF-IoT)*, pages 199–202, 2014.

[93] M. Mikulasek, R. Dvorak, M. Stusek, P. Masek, R. Mozny, P. Mlynek, and J. Hosek. Nb-iot vs lte cat m1: Demystifying performance differences under varying radio conditions. In *2022 14th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pages 133–138, 2022.

[94] R. Miyatake, H. Katsurai, Y. Fukada, M. Sekiguchi, and T. Yoshida. Ultra-low-power optical network unit driven by optical power supply using single-mode fiber. *IEEE Photonics Technology Letters*, 35(16):874–877, 2023.

[95] S. Myung, K. Yang, and J. Kim. Quasi-Cyclic LDPC Codes for Fast Encoding. *IEEE Trans. Inf. Theory*, 51, No. 8:2894–2901, August 2005.

[96] J. Nadal and A. Baghdadi. Parallel and flexible 5g ldpc decoder architecture targeting fpga. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 29(6):1141–1151, 2021.

[97] M. Ndiaye, A. M. Saley, K. Niane, and A. Raimy. Future 6g communication networks: Typical iot network topology and terahertz frequency challenges and research issues. In *2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, pages 1–5, 2022.

[98] T. T. B. Nguyen, T. Nguyen Tan, and H. Lee. Efficient qc-ldpc encoder for 5g new radio. *Electronics*, 8:art. no. 668, 2019.

[99] T. T. B. Nguyen, T. N. Tan, and H. Lee. Low-complexity high-throughput qc-ldpc decoder for 5g new radio wireless communication. *Electronics*, 10:1–18, 1 2021.

[100] T. T. Nguyen-Ly, V. Savin, D. Declercq, F. Ghaffari, and O. Boncalo. Analysis and Design of Cost-Effective, High-Throughput LDPC Decoders. *IEEE Trans. VLSI Syst.*, 26:508–521, March 2018.

[101] F. Noorbehbahani, M. Aminazadeh, H. H. Esfahani, and S. Bajoghli. Business models for the internet of things: An in-depth categorization, design and innovation tools, challenges, and solutions. In *2023 7th International Conference on Internet of Things and Applications (IoT)*, pages 1–11, 2023.

[102] S. I. Park, Y. Wu, H. M. Kim, N. Hur, and J. Kim. Raptor-like rate compatible ldpc codes and their puncturing performance for the cloud transmission system. *IEEE Transactions on Broadcasting*, 60(2):239–245, 2014.

[103] K. Pawar, K. Kasat, A. P. Deshpande, and N. Shaikh. Adoption of industry 4.0 technologies for performance management. In *2023 1st DMIHER International Conference on Artificial Intelligence in Education and Industry 4.0 (IDICAIEI)*, volume 1, pages 1–5, 2023.

[104] W. W. Peterson and D. T. Brown. Cyclic codes for error detection. *Proceedings of the IRE*, 49(1):228–235, 1961.

[105] V. L. Petrović, D. M. El Mezeni, and A. Radošević. Flexible 5g new radio ldpc encoder optimized for high hardware usage efficiency. *Electronics*, 10(9), 2021.

[106] C. Rajasekaran and K. Raguvaran. Microcontroller based reconfigurable iot node. In *IEEE 4th International Conference on Frontiers of Signal Processing*, pages 12–16, Poitiers, France, September 2018.

[107] I. S. Reed and G. Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960.

[108] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke. Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes. *IEEE Trans. Inf. Theory*, 47:619–637, February 2001.

[109] T. J. Richardson and R. L. Urbanke. Efficient Encoding of Low-Density Parity-Check Codes. *IEEE Trans. Inf. Theory*, 47:638–656, February 2001.

[110] P. Sangmahamad, T. Pechrkool, and P. Thiamsinsangwon. An optical fiber monitoring and alert system for a passive optical network based on iot. In *2021 Research, Invention, and Innovation Congress: Innovation Electricals and Electronics (RI2C)*, pages 258–261, 2021.

[111] I. A. Shah, F. A. Malik, and S. A. Ahmad. Enhancing security in iot based home automation using reed solomon codes. In *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pages 1639–1642, 2016.

[112] E. A. Shammar and A. T. Zahary. The internet of things (iot): a survey of techniques, operating systems, and trends. *Library Hi Tech*, 38(1):5–66, 2020.

[113] M. S. Sharbaf. Iot driving new business model, and iot security, privacy, and awareness challenges. In *2022 IEEE 8th World Forum on Internet of Things (WF-IoT)*, pages 1–4, 2022.

[114] K. Shimizu and H. Suzuki. Proposal for optical fiber network monitoring system using iot technology. In *2022 Tenth International Symposium on Computing and Networking Workshops (CANDARW)*, pages 421–425, 2022.

[115] A. Shokrollahi. Raptor codes. *IEEE Trans. Inf. Theory*, 52(6):2551–2567, 2006.

[116] M. Stark, J. Lewandowsky, and G. Bauch. Information-bottleneck decoding of high-rate irregular ldpc codes for optical communication using message alignment. *Appl. Sciences*, 10:1–17, 8 2018.

[117] W. Sulek. Pipeline processing in low-density parity-check codes hardware decoder. *Bull. Pol. Ac.: Tech.*, 59:149–155, February 2011.

[118] W. Sulek. Non-binary LDPC Decoders Design for Maximizing Throughput of an FPGA Implementation. *Circuits, Systems, and Signal Processing*, 35:4060–4080, November 2016.

[119] W. Sulek. Quasi-Regular QC-LDPC Codes Derived From Cycle Codes. *IEEE Commun. Lett.*, 20:1705–1708, 2016.

[120] W. Sulek. Protograph Based Low-Density Parity-Check Codes Design with Mixed Integer Linear Programming. *IEEE Access*, 7:1424–1438, 2019.

[121] W. Sulek and M. Kucharczyk. Partial parallel encoding and algorithmic construction of non-binary structured IRA codes. *China Communications*, 13:103–116, 2016.

[122] S. N. Swamy and S. R. Kota. An empirical study on system level aspects of internet of things (iot). *IEEE Access*, 8:188082–188134, 2020.

[123] R. M. Tanner. A Recursive Approach to Low Complexity Codes. *IEEE Transactions on Information Theory*, IT-27:533–547, September 1981.

[124] C. Tarver, M. Tonnemacher, H. Chen, J. Zhang, and J. R. Cavallaro. Gpu-based, ldpc decoding for 5g and beyond. *IEEE Open J. Circuits Syst.*, 2:278–290, 1 2021.

[125] A. Tasdighi, A. H. Banihashemi, and M.-R. Sadeghi. Efficient Search of Girth-Optimal QC-LDPC Codes. *IEEE Trans. Inf. Theory*, 62:1552–1564, April 2016.

[126] A. Tasdighi, A. H. Banihashemi, and M.-R. Sadeghi. Symmetrical Constructions for Regular Girth-8 QC-LDPC Codes. *IEEE Trans. Commun.*, 52, No. 8:1242–1247, August 2017.

[127] D. Theodoropoulos, N. Kranitis, and A. Paschalis. An efficient ldpc encoder architecture for space applications. In *IEEE 22nd International Symposium on On-Line Testing and Robust System Design (IOLTS)*, pages 149–154, Sant Feliu de Guixols, Spain, July 2016.

[128] T. Thi Bao Nguyen, T. Nguyen Tan, and H. Lee. Low-complexity high-throughput qc-ldpc decoder for 5g new radio wireless communication. *Electronics*, 10(4):516, 2021.

[129] R. R. Tummala. Moore's law for packaging to replace moore's law for ics. In *2019 Pan Pacific Microelectronics Symposium (Pan Pacific)*, pages 1–6, 2019.

[130] P. Upadhyaya, S. Dutt, Ruchi, and S. Upadhyaya. 6g communication: Next generation technology for iot applications. In *2021 First International Conference on Advances in Computing and Future Communication Technologies (ICACFCT)*, pages 23–26, 2021.

[131] S. Usman and M. M. Mansour. Fast column message-passing decoding of low-density parity-check codes. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 68(7):2389–2393, 2021.

[132] S. K. Vishwakarma, P. Upadhyaya, B. Kumari, and A. K. Mishra. Smart energy efficient home automation system using iot. In *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, pages 1–4, 2019.

[133] S.-Y. Wang, J.-E. Chang, H. Fan, and Y.-H. Sun. Performance comparisons of nb-iot, lte cat-m1, sigfox, and lora moving at high speeds in the air. In *2020 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6, 2020.

[134] Y. Wang, S. C. Draper, and J. S. Yedidia. Hierarchical and High-Girth QC LDPC Codes. *IEEE Trans. Inf. Theory*, 59:4553–4583, July 2013.

[135] H. Wu and H. Wang. A high throughput implementation of qc-ldpc codes for 5g nr. *IEEE Access*, 7:185373–185384, 2019.

[136] H. Xiao and A. H. Banihashemi. Improved Progressive-Edge-Growth (PEG) Construction of Irregular LDPC Codes. *IEEE Commun. Lett.*, 8, No. 12:715–717, December 2004.

[137] H. Xu, D. Feng, R. Luo, and B. Bai. Construction of Quasi-Cyclic LDPC Codes via Masking With Successive Cycle Elimination. *IEEE Commun. Lett.*, 20:2370–2373, December 2016.

[138] W. Yu, K. Narayanan, J. Cheng, and J. Wu. Raptor codes with descending order degrees for awgn channels. *IEEE Commun. Lett.*, 24(1):29–33, 2020.

[139] F. Zarkeshvari and A. H. Banihashemi. On Implementation of Min-Sum Algorithm for Decoding Low-Density Parity-Check (LDPC) Codes. In *IEEE Globecom*, pages 1349–1353, Taipei, Taiwan, August 2002.

[140] H. Zhang, J. Li, B. Wen, Y. Xun, and J. Liu. Connecting intelligent things in smart hospitals using nb-iot. *IEEE Internet of Things Journal*, 5(3):1550–1560, 2018.

[141] S. Zhao and X. Ma. Construction of High-Performance Array-Based Non-Binary LDPC Codes With Moderate Rates. *IEEE Commun. Lett.*, 20:13–16, January 2016.

[142] Z. Zhao, X. Jiao, J. Mu, and Y.-C. He. Randomly penalized symbol flipping decoding of non-binary ldpc codes. *IEEE Trans. Veh. Technol.*, 70(1):639–654, 2021.

[143] Y. Zhu, Z. Xing, Z. Li, Y. Zhang, and Y. Hu. High area-efficient parallel encoder with compatible architecture for 5g ldpc codes. *Symmetry*, 13(4):700, 2021.

5 Publikacje będące podstawą rozprawy doktorskiej

Sumaryczny Impact Factor: 6,9

Suma punktów MNiSW: 400

Numer Publikacji	Tytuł	Punktacja	Impact Factor
1	Aplikacje kodów korekcyjnych LDPC we współczesnych systemach radiokomunikacyjnych	20	-
2	Efficient LDPC Encoder Design for IoT-Type Devices	100	2.7
3	Dekoder LDPC implementowany w mikrokontrolerze dla systemów Internetu Rzeczy	70	0.5
4	Energy-efficient Raptor-like LDPC coding scheme design and implementation for IoT communication systems	140	3.2
5	Niebinarne kodowanie LDPC dla systemów Internetu rzeczy	70	0.5

Tabela 1: Publikacje

Numer	Tytuł	Punktacja	Impact Factor
6	Short Blocklength Nonbinary Raptor-Like LDPC Coding Systems Design and Simulation	100	4.1

Tabela 2: Manuskrypty w trakcie recenzji.

5.1 Publikacja 1: Aplikacje kodów korekcyjnych LDPC we współczesnych systemach radiokomunikacyjnych.

Wojciech SUŁEK
Politechnika Śląska, Wydział Automatyki, Elektroniki i Informatyki, Gliwice
Jakub HYLA
TKH Technology Poland
Politechnika Śląska, Wydział Automatyki, Elektroniki i Informatyki, Gliwice

APLIKACJE KODÓW KOREKCYJNYCH LDPC WE WSPÓŁCZESNYCH SYSTEMACH RADIOKOMUNIKACYJNYCH

Streszczenie. Kody LDPC stanowią jedną z najlepszych metod kodowania korekcyjnego. W artykule przedstawiono przegląd charakterystycznych cech podklas kodów LDPC zastosowanych w istotnych współcześnie standardach radiokomunikacyjnych, takich jak WiFi, WiMAX, DVB-S2 i 5G. Przedstawiono także wyniki symulacji tych systemów transmisji dające porównanie poszczególnych aplikacji kodów LDPC.

Słowa kluczowe: kodowanie korekcyjne**,** kody blokowe, kody LDPC, sieci 5G.

LDPC CODES APPLICATIONS IN TELECOMMUNICATION SYSTEMS

Summary. LDPC codes are one of the best error correction coding methods. In this article, we review the characteristic features of LDPC subclasses applied in important radiocommunication standards, such as WiFi, WiMAX, DVB-S2 and 5G. Then, we provide a simulation results of selected codes taken from these standards, presenting comparison between different configurations.

Keywords: error correction coding, block coding**,** LDPC codes, 5G networks.

1. Wprowadzenie

Rola szybkiej i skutecznej komunikacji cyfrowej w postępie technologicznym jest nie do przecenienia. Jednym z podstawowych czynników napędzających rozwój możliwości przesyłania informacji jest nieustanny progres szybkości i niezawodności systemów transmisji danych cyfrowych. Progres ten jest podtrzymywany przez prace badawcze ukierunkowane na

ulepszanie technik transmisji danych, w tym technik kodowania pozwalającego na detekcję i korekcję błędów transmisji.

Jedną z najbardziej skutecznych znanych i szeroko opisanych metod kodowania korekcyjnego jest kodowanie LDPC (ang. Low-Density Parity-Check). Jest to rodzina kodów blokowych, po raz pierwszy opisanych w 1962 roku, ale ze względu na zbyt dużą złożoność obliczeniową jak na tamte czasy – na lata zapomnianych. Możliwe do praktycznego zastosowania algorytmy kodowania i dekodowania LDPC zostały przedstawione w latach dziewięćdziesiątych przez D.J.C. MacKaya [5] i od tego czasu cieszą się dużym zainteresowaniem, zarówno świata naukowego [1]-[9], jak i organizacji standaryzujących technologie radiokomunikacyjne [10]-[13]. Swoje zastosowanie znajdują w transmisji sygnału telewizji cyfrowej DVB-T2 oraz DVB-S2, bezprzewodowych sieciach lokalnych WLAN, sieciach WiMAX, jak również w sieciach komórkowych piątej generacji (5G). W niniejszej publikacji zawarto przegląd aktualnych zastosowań kodów LDPC, wraz z krótkim wprowadzeniem przywołującym podstawowe definicje związane z kodami LDPC. Zamieszczono także wyniki symulacji własnych, w postaci zależności bitowej stopy błędów (BER) od poziomu zakłóceń w kanale radiokomunikacyjnym, dla różnych konfiguracji kodów o macierzach kontrolnych zaczerpniętych z poszczególnych standardów.

2. Kodowanie LDPC

Kodowanie blokowe polega na sukcesywnym gromadzeniu *K*-elementowych bloków symboli informacyjnych i dołączaniu do każdego bloku pewnej liczby symboli kontrolnych, które służą do detekcji i korekcji błędów, np. po stronie cyfrowego odbiornika sygnału transmitowanego drogą radiową. W danym systemie kodowania blokowego określane jest jednoznaczne przyporządkowanie każdego z możliwych wektorów informacyjnych $\mathbf{u} = [u_1, u_2, : : : , u_K]$ do wektora kodowego $\mathbf{c} = [c_1, c_2, : : : , c_N]$, gdzie $N = K + M$, gdzie M jest liczbą symboli kontrolnych (nadmiarowych). Kolejne słowa, w których zawarta jest informacja **u**, są przekształcane w zakodowane informacje **c** niezależnie od siebie. Taki kod jest określany kodem $C(N,K)$ [1].

Sprawność kodu $R = K/N$ jest parametrem, który wskazuje poziom nadmiarowości danego kodu i zawiera się w przedziale $0 < R < 1$. Warto zwrócić uwagę na to, że im większa nadmiarowość kodu, tym mniejsza sprawność, ale za to większe możliwości korekcyjne. Klasa kodów LDPC zawiera kody o dowolnej sprawności, która może być dobrana do danego systemu / zastosowania.

Elementy wektorów **u**, **c** mogą należeć do dowolnego ciała skończonego, czyli ciała Galois *GF(q)*. Powszechne jest stosowanie kodowania binarnego, nad ciałem *GF(2)*, dla którego $u_k, c_n \in \{0, 1\}$. Istotną kwestią jest również fakt, że znaczenie mają w zasadzie tylko kody

liniowe, dla których suma dwóch dowolnych wektorów kodowych $\mathbf{c}_1$ $\mathbf{c}_2$ daje wektor $\mathbf{c}_3$, który jest również prawidłowym słowem kodowym.

Wyznaczając poszczególne elementy c_n wektora kodowego **c** generuje się słowa kodu *C(N,K)*, zgodnie z równaniem:

$$\mathbf{c} = \mathbf{u} \cdot \mathbf{G} \tag{2}$$

gdzie macierz generująca kodu **G** ma rozmiar $K \times N$, a mnożenie jest iloczynem macierzowym z operacjami w ciele *GF(2)*. Kolumny **G** zawierają wagi, za pomocą których wyznacza się kolejne elementy c_n wektora kodowego. Macierz generująca jednoznacznie określa konkretny linowy kod blokowy.

Można wykazać, że dla dowolnej macierzy generującej **G** o rozmiarze, istnieje macierz **H** o rozmiarze $(N\text{-}K) \times N$, w której wszystkie wiersze są ortogonalne do wierszy **G**, co wyraża zależność:

$$\mathbf{G} \cdot \mathbf{H}^{\mathbf{T}} = \mathbf{0} \tag{3}$$

gdzie **0** to wektor zerowy. Stąd:

$$\mathbf{c} \cdot \mathbf{H}^{\mathbf{T}} = \mathbf{u} \cdot \mathbf{G} \cdot \mathbf{H}^{\mathbf{T}} = \mathbf{0} \tag{4}$$

Macierz **H** jest nazywana macierzą kontrolną kodu, natomiast $\mathbf{cH^T = 0}$ albo równoważnie $\mathbf{Hc^T = 0}$ nazywane jest równaniem kontrolnym. Daje ono możliwość sprawdzenia, czy dany wektor **c** jest prawidłowym słowem kodowym, pozwalając na detekcję ewentualnych błędów.

Korzystając z przedstawionych informacji można przedstawić jedną z form definicji kodu blokowego. Binarny liniowy kod blokowy *C(N,K)* o długości słowa *N* i liczbie słów kodowych 2^K jest zbiorem wszystkich wektorów $\mathbf{c}_{1\times K}$ spełniających równanie macierzowe w ciele GF(2):

$$\mathbf{H} \cdot \mathbf{c}^{\mathbf{T}} = \mathbf{0},$$

gdzie $\mathbf{H}_{MxN}$ jest macierzą kontrolną kodu, a $M = N - K$.

Kody LDPC to kody blokowe, których macierz kontrolna **H** jest macierzą posiadającą małe zagęszczenie niezerowych elementów, czyli jest to tzw. macierz rzadka. Dany kod LDPC jest zatem definiowany za pomocą jego macierzy kontrolnej. Wyróżnia się dwa typy kodów LDPC – regularne i nieregularne. Regularność oznacza tu jednakową liczbę elementów niezerowych w każdej kolumnie macierzy kontrolnej, co w szczególności upraszcza konstrukcję sprzętowych kodeków. Jednakże kody nieregularne o odpowiednio skonstruowanej macierzy kontrolnej **H** (w tym odpowiednio dobranej dystrybucji niezerowych elementów w kolumnach), mają w ogólności lepsze własności korekcyjne od kodów regularnych [7]. Możliwe jest uzyskanie kodów (o dużej długości bloku), mających możliwości korekcyjne zbliżone do teoretycznej granicy Shannona [4], [5], [7], [9]. W większości znanych

praktycznych aplikacji, w tym w wymienionych we wstępie standardach radiokomunikacyjnych, stosowane są kody nieregularne.

Oprócz cechy nieregularności, stosowane w tych standardach macierze kontrolne mają zwykle dodatkowe wspólne cechy strukturalne, w szczególności związane z usprawnieniem wymagań obliczeniowych i architektury szybkich sprzętowych modułów koderów i dekoderów [2]:

- Macierz kontrolna składa się z podmacierzy kwadratowych, gdzie każda z podmacierzy jest macierzą zerową lub pewną permutacją kolumn macierzy jednostkowej, co umożliwia organizację dostępów jednostek obliczeniowych do pamięci w sprzętowej implementacji, dlatego też takie kody nazywane są kodami zorientowanymi na implementację (ang. Architecture Aware – AA-LDPC). Najważniejszą podklasą kodów AA-LDPC są kody QC-LDPC (ang. Quasi-Cyclic) [6], w których podmacierze kwadratowe są podmacierzami przesunięcia cyklicznego kolumn macierzy jednostkowej, dzięki czemu kod ma własność quasi-cykliczności, dającą dodatkową możliwość uproszczenia budowy kodera i dekodera.
- Macierz kontrolna ma formę prawie-dolnotrójkątną, co umożliwia wykorzystanie efektywnego algorytmu kodowania opisanego w [8]. W wielu aplikacjach [10], [11], [12] macierz jest nie tylko dolnotrójkątna, ale dodatkowo dwu-diagonalna (ang. dual-diagonal) [6], dzięki czemu wyeliminowany jest krok algorytmu, którego zależność złożoności obliczeniowej od długości bloku jest nieliniowa. W efekcie, przy odpowiednim doborze wartości przesunięć cyklicznych w podmacierzach (jak pokazano w [6]), złożoność obliczeniowa algorytmu kodowania jest liniowo zależna od długości bloku.
- Wszystkie znaczące standardy radiokomunikacyjne zawierają mechanizmy konfigurowalności długości bloku (N) i sprawności kodu (R), albo inaczej: mechanizmy dostosowania możliwości korekcyjnych do aktualnego stanu kanału komunikacyjnego. Najczęściej takie dostosowanie odbywa się na zasadzie wyboru kodu z predefiniowanego katalogu [10], [11], [12], ale np. schemat kodowania w 5G NR [13] dodatkowo udostępnia mechanizmy: dziurkowania (ang. puncturing), skracania (ang. shortening) i powtarzania (ang. repetition) elementów bloków informacyjnych [3].

Stosowanie kodów QC-LDPC w wymienionych standardach umożliwia także zwięzłą prezentację ogromnych macierzy (o liczbie kolumn liczonej w tysiącach lub dziesiątkach tysięcy). Na przykład w dokumentach standaryzacyjnych [10],[11] zamieszczone są tabele prezentujące tylko położenie niezerowych podmacierzy w **H** oraz liczbowe wartości przesunięć cyklicznych. Każda podmacierz jest zatem przedstawiana za pomocą jednej liczby – cyklicznego przesunięcia macierzy jednostkowej, lub też symbolu (umownie: wartość ujemna, -1 albo symbol "-" jak w [11]) wskazującego podmacierz zerową, gdyż 0 reprezentuje podmacierz o zerowym przesunięciu cyklicznym, nie może więc wskazywać podmacierzy zerowej.

Wykorzystywane w praktyce algorytmy dekodowania należą do grupy iteracyjnych algorytmów propagacji wiadomości. Dogodny sposób prezentacji tych algorytmów wykorzystuje tzw. graf Tannera [5], który jest alternatywnym dla **H** (bądź **G**) sposobem definiowania kodu. Dla liniowego kodu blokowego o macierzy kontrolnej **H**, graf Tannera tego kodu to graf dwudzielny, w którym jeden podzbiór wierzchołków (kontrolnych) odpowiada równaniom kontrolnym kodu (czyli wierszom **H**), a drugi podzbiór – bitom słowa kodowego (kolumnom **H**). Wierzchołek bitowy *n* jest połączony krawędzią z wierzchołkiem kontrolnym o numerze *m* wtedy i tylko wtedy, gdy element h_{mn} macierzy **H** jest jedynką.

Wierzchołki grafu reprezentują elementarne obliczenia realizowane w procesie dekodowania, natomiast krawędzie wskazują sposób przesyłania wyników tych obliczeń (wiadomości). Wiadomości reprezentują przybliżone wartości prawdopodobieństw, że dany bit jest równy 0 bądź 1 (wiadomości z wierzchołków bitowych do kontrolnych) oraz przybliżonych wartości prawdopodobieństw, że dane równanie kontrolne jest spełnione, przy określonej wartości danego bitu (wiadomości z wierzchołków kontrolnych do bitowych). Idea dekodowania polega na iteracyjnym uzyskiwaniu coraz lepszych przybliżeń wartości tych prawdopodobieństw. Taki algorytm dekodowania znany jako algorytm BP (ang. Belief Propagation), czyli algorytm propagacji poziomów wiarygodności. Danymi inicjalizującymi algorytm są prawdopodobieństwa *a priori*, czyli wartości (tzw. miękkie decyzje wejściowe) otrzymane z układów demodulatora sygnału radiowego. Szczegółowe zależności matematyczne definiujące algorytm dekodowania przedstawiane są w licznej literaturze, np. [1], [4], [5].

W implementacjach dekoderów, dla zapewnienia większej przepustowości wykorzystywane są obliczenia równoległe w celu zapewnienia odpowiedniej przepustowości algorytmu dekodowania [2].

3. Przegląd aplikacji kodów LDPC

W rozdziale przedstawiono przegląd prawdopodobnie najistotniejszych współczesnych aplikacji kodów LDPC, którymi są WiFi 802.11 [11], WiMAX 802.16e [10], telewizja cyfrowa DVB-S2 [12] oraz sieci mobilne piątej generacji 5G [13]. Zbiorcze zestawienie parametrów kodów LDPC zastosowanych w tych standardach znajduje się w Tabeli 1.

3.1. IEEE Std 802.11, znany jako WiFi [11]

Standardy z oznaczeniem IEEE 802.11 są rodziną standardów bezprzewodowych sieci lokalnych WLAN (ang. Wireless Local Area Network) [11]. Początki sięgają roku 1997, jednakże przez lata aż do chwili obecnej powstają nowe specyfikacje, z coraz to wyższymi szybkościami transmisji. Początkowo do korekcji błędów transmisji wykorzystywany był kod

splotowy, jednakże obecne normy zalecają kody LDPC w wielu produktach WLAN ze względu na wysoką wydajność korekcji błędów. Między innymi standard IEEE 802.11ac specyfikuje wykorzystanie kodów LDPC do korekcji błędów transmisji i umożliwia przesyłanie danych z teoretyczną szybkością bitową rzędu kilku Gbps.

Zastosowany katalog kodów LDPC obejmuje kody QC-LDPC, o strukturze dwu-diagonalnej, różnych długościach bloku kodowego i bloku informacyjnego oraz różnych rozmiarach podmacierzy, które są oznaczane w [11] literą *Z*.

3.2. IEEE Std 802.16e, znany jako WiMAX [10]

WiMAX jest nieco mniej znanym standardem komunikacji bezprzewodowej, zaprojektowanym do korzystania na niskich i średnich dystansach w zabudowaniach miejskich (typowe dystanse to np. 10km). Podstawowy standard definiuje szybkości transmisji do 75 Mbps. Podobnie do Wi-Fi, zastosowano dwu-diagonalne kody QC-LDPC, o nieco większej długości słowa kodowego niż w WiFi (w zakresie od 576 do 2304 bitów), co jest naturalne dla standardu transmisji na nieco większe odległości.

3.3. Telewizja cyfrowa, DVB-S2 i DVB-T2 [12]

Standardy telewizji cyfrowej są wykorzystywane na całym świecie do rozsiewczej (broadcastowej) transmisji dźwięku i obrazów do domowych odbiorników telewizyjnych. Standardy te mają zaimplementowane tryby dające możliwość optymalizacji wykorzystania przepustowości poprzez dynamiczne zmiany parametrów transmisji. Porównując standard DVB-S2 do DVB-S, wzrost wydajności kształtuje się na poziomie około 30%.

Zastosowane kody LDPC są również kodami klasy QC-LDPC, ale w porównaniu z WiMAX i WiFi, długości bloków kodowych są znacznie większe. Wynika to z faktu, że w jednokierunkowej, rozsiewczej transmisji sygnału telewizyjnego, opóźnienie przetwarzania (ang. Latency) kodera i dekodera, które jest zależne od długości bloku kodowego, nie ma istotnego znaczenia. Natomiast z drugiej strony, długi blok pozwala na uzyskanie znakomitych własności korekcyjnych.

3.4. Sieci mobilne 5G [13]

Piąta generacja (5G) szerokopasmowych sieci mobilnych stanowi duży krok w ewolucji dzisiejszych sieci LTE czwartej generacji. 5G jest projektowane do tego, aby sprostać nieustannemu wzrostowi potrzeb szybkiej transmisji danych oraz liczby połączeń, w tym zapewnić działanie Internetu Rzeczy z miliardami połączonych ze sobą urządzeń, a także umożliwić przyszłe innowacje.

Kodowanie LDPC w 5G NR (ang. New Radio) wykorzystywane jest w kanałach danych użytkowych. Specyfika sieci komórkowych, zwłaszcza heterogenicznych sieci nowych

generacji, wymaga zastosowania takich technik przetwarzania sygnałów, które mogą być dostosowane do bardzo zróżnicowanych warunków transmisji. Mnogość scenariuszy, w tym wielkości komórek, warunków propagacji, kategorii urządzeń mobilnych, mocy transmisji, rozmiarów paczek transmitowanych danych itd., wymaga także elastycznego sposobu kodowania, który cechuje się bardzo szerokim zakresem konfigurowalności długości bloku i sprawności kodu. W związku z tym, zamiast katalogu predefiniowanych kodów, sporządzono specyfikację dwóch macierzy bazowych (ang. base matrix 1 and base matrix 2) oraz zaproponowano schemat generowania wektorów kodowych o konfigurowalnych długościach, przy macierzy kontrolnej będącej określonym fragmentem jednej z macierzy bazowych. Macierz bazowa jest jedno-diagonalną macierzą typu QC-LDPC, w której wyróżniono kolumny skojarzone z bitami informacyjnymi (ang. systematic bits) i bitami nadmiarowymi (ang. parity bits). Poprzez wybór odpowiedniej części macierzy, możliwa jest zmiana nie tylko długości słowa kodowego, ale także sprawności. Jest to tzw. struktura kodu z uzgadnianiem sprawności (ang. rate-compatible structure). Dodatkową elastyczność parametrów systemu zapewniają operacje dziurkowania, skracania i powtarzania.

3.5. Zestawienie parametrów kodów LDPC w aplikacjach radiokomunikacyjnych

W Tabeli 1 przedstawiono zbiorcze zestawienie podstawowych parametrów kodów QC-LDPC zastosowanych w poszczególnych standardach. Zwraca uwagę szczególnie duży rozmiar bloku kodów DVB-S2/T2, jak również szeroki zakres konfiguracji parametrów kodów dla 5G, co jest związane ze wspomnianą wyżej elastycznością warstwy fizycznej 5G NR. Waga kolumny macierzy kontrolnej to liczba niezerowych elementów w tej kolumnie. Maksymalna waga kolumny (w Tabeli 1) w pośredni sposób pokazuje poziom nieregularności macierzy definiującej kod nieregularny.

Tabela 1

Podstawowe parametry kodów LDPC stosowanych w istotnych standardach radiokomunikacyjnych

	Liczba bitów w bloku (min – max) (codeword lenght)	Sprawność (code rate)	Wielkość podmacierzy (submatrix size)	Maks. waga kolumny macierzy
WiFi 802.11n	648 – 1944	$\frac{1}{2} - \frac{5}{6}$	27 – 81	12
WiMAX 802.16e	576 – 2304	$\frac{1}{2} - \frac{5}{6}$	24 – 96	6
DVB-S2 DVB-T2	16200 – 64800	$\frac{1}{5} - \frac{9}{10}$	360	13
5G NR	544 – 8448	$\frac{10}{52} - \frac{948}{1024}$	2 – 384	12

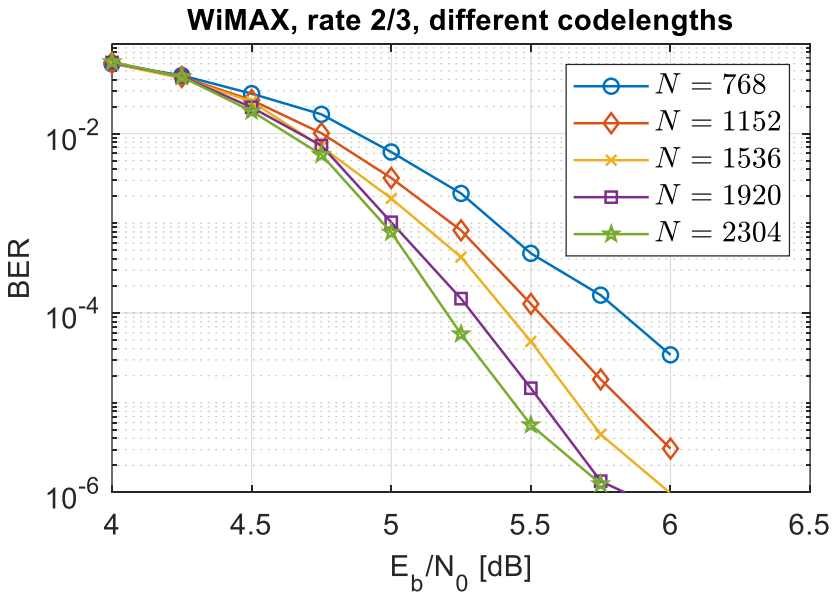
Źródło: Opracowanie własne na podstawie [10], [11], [12], [13].

3.6. Możliwości korekcyjne wybranych kodów

Możliwości kodów korekcyjnych są zwykle przedstawiane w formie wykresów stopy błędów w funkcji parametru charakteryzującego kanał komunikacyjny. Bitowa stopa błędów (BER) wskazuje oczekiwany stosunek liczby błędnych bitów po zdekodowaniu (korekcji w dekoderze LDPC) do liczby wszystkich bitów transmitowanych przez kanał.

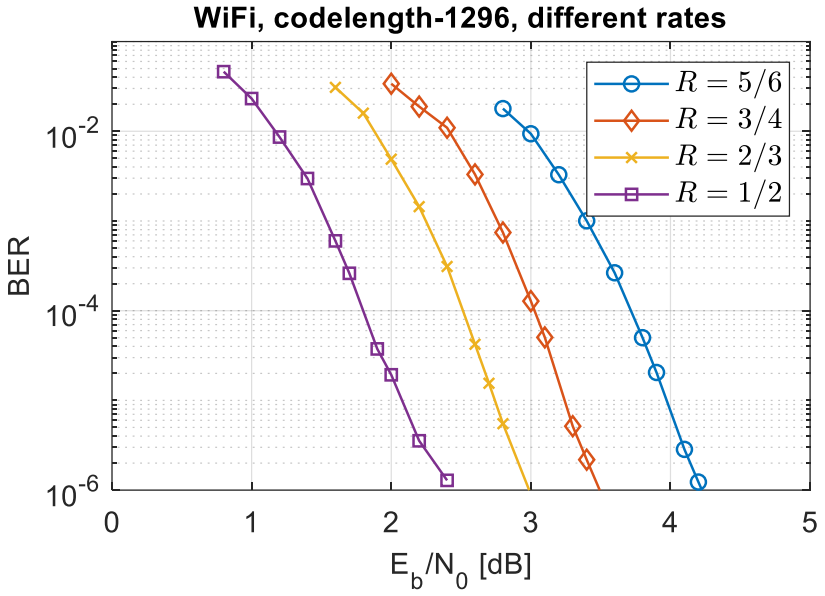
W niniejszym podrozdziale przedstawiono wykresy stopy błędów uzyskane w wyniku symulacji odpowiednio dużej liczby wektorów testowych. Symulacja polega na losowej generacji ciągu bitów gromadzonych w bloki informacyjne, kodowaniu (LDPC), modulacji (zastosowano modulacje takie jak w standardach, QPSK oraz QAM), propagacji przez model kanału (typu AWGN – Additive White Gaussian Noise) oraz dekodowaniu (algorytmem BP [5]). Tego typu system symulacyjny został opracowany w środowisku Matlab. Parametrem kanału AWGN jest stosunek mocy sygnału do mocy szumu (SNR), lub też efektywny SNR, w którym uwzględniono liczbę bitów informacyjnych przypadających na symbol QAM. Taki efektywny SNR [4] to wartość E_b/N_0 gdzie E_b to energia przypadająca na bit informacyjny, a N_0 to stała gęstość widmowa szumu.

Na Rys.1 przedstawiono wyniki uzyskane dla kodów IEEE 802.16e [10] o sprawności 2/3, przy różnych długościach bloków kodowych i modulacji QAM-16. Wyniki te pokazują typową poprawę zdolności korekcyjnych przy zwiększaniu długości bloku. Z drugiej strony, dłuższy blok oznacza większe opóźnienie przetwarzania sygnału oraz większe wymagania sprzętowe.



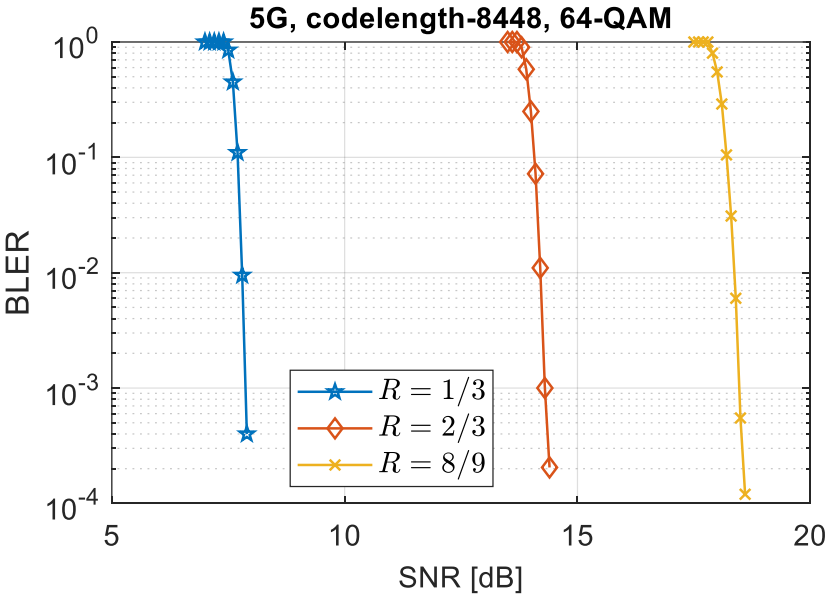
Rys. 1. Bitowa stopa błędów (BER) w funkcji efektywnego stosunku sygnału do szumu, dla transmisji sygnału QAM-16 z kodowaniem LDPC WiMAX przez kanał AWGN

Fig. 1. Bit Error Rate (BER) versus effective signal to noise ratio, for QAM-16 transmission with LDPC WiMAX coding over AWGN channel



Rys. 2. Bitowa stopa błędów (BER) w funkcji efektywnego stosunku sygnału do szumu, dla transmisji sygnału QPSK z kodowaniem LDPC WiFi przez kanał AWGN

Fig. 2. Bit Error Rate (BER) versus effective signal to noise ratio, for QPSK transmission with LDPC WiFi coding over AWGN channel



Rys. 3. Blokowa stopa błędów (BLER) w funkcji stosunku sygnału do szumu, dla transmisji sygnału 64-QAM z kodowaniem LDPC 5G przez kanał AWGN

Fig. 3. Block Error Rate (BLER) versus signal to noise ratio, for QAM-64 transmission with LDPC 5G coding over AWGN channel

Na Rys. 2 przedstawiono wyniki uzyskane dla kodów IEEE 802.11n [11] o długościach bloku 1296 bitów, uzyskane dla symulacji transmisji QPSK przez kanał AWGN, przy różnych

sprawnościach kodu. Wyniki pokazują możliwość dostosowania systemu kodowania do aktualnego poziomu zakłóceń w kanale. Zmniejszenie sprawności umożliwia poprawną korekcję przy większym poziomie zakłóceń, czyli mniejszym stosunku sygnału do szumu, jednakże kosztem jest zmniejszenie efektywności widmowej transmisji (mniejsza liczba bitów informacyjnych w bloku kodowym).

Na Rys. 3 przedstawiono blokową stopę błędów (BLER) dla kodów 5G NR [3] [13], przy transmisji z modulacją 64-QAM. Tym razem na osi poziomej znajduje się wartość SNR w kanale. Warto zauważyć, do jak szerokiego zakresu SNR może być dostrajany system kodowania, tylko poprzez zmianę kodu. Dodatkowe możliwości dostrajania daje zmiana modulacji (QPSK/ 16-QAM / 64-QAM / 256-QAM) oraz operacje dziurkowania i skracania kodu.

4. Podsumowanie

Ze względu na znakomite własności korekcyjne przy rozsądnej złożoności obliczeniowej kodeka, klasa kodów LDPC jest prawdopodobnie najchętniej stosowaną metodą kodowania we współczesnych zaawansowanych systemach transmisji danych cyfrowych. Tym niemniej, ze względu na zróżnicowane wymagania, kody stosowane w poszczególnych systemach dosyć istotnie różnią się ze względu na pewne charakterystyczne parametry. W tym artykule przedstawiono zestawienie prawdopodobnie najistotniejszych aplikacji kodów LDPC, uwypuklając charakterystyczne cechy i różnice. Przedstawiono również wyniki pokazujące możliwości korekcyjne, korzystając z autorskiego środowiska symulacji systemów transmisji.

Bibliografia

1. Izydorczyk J., Sułek W., Zawadzki P.: Kody i szyfry. Wydawnictwo Politechniki Śląskiej, Gliwice 2017.
2. Hailes P., Maunder R.G., Al-Hashimi B.M., Hanzo L.: A Survey of FPGA-Based LDPC Decoders. "IEEE Communications Surveys & Tutorials", vol. 18, 2016, pp. 1098-1122.
3. Li H., Bai B., Mu X., Zhang J., Xu H.: Algebra-Assisted Construction of Quasi-Cyclic LDPC Codes for 5G New Radio. "IEEE Access", vol. 6, 2018, pp. 50229–50244.
4. Lin S., Costello D.J.: Error Control Coding. Pearson Prentice Hall, 2nd edition, 2004.
5. MacKay D.J.C.: Good Error-Correcting Codes Based on Very Sparse Matrices. "IEEE Transactions on Information Theory", vol. 45, 1999, pp. 399–431.

6. Myung S., Yang K., Kim J.: Quasi-Cyclic LDPC Codes for Fast Encoding. "IEEE Transactions on Information Theory", vol. 51, 2005, pp. 2894–2901.
7. Richardson T. J., Shokrollahi M. A., Urbanke R. L.: Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes. "IEEE Transactions on Information Theory", vol. 47, 2001, pp. 619–637.
8. Richardson T. J., Urbanke, R. L.: Efficient Encoding of Low-Density Parity-Check Codes. " IEEE Transactions on Information Theory", vol. 47, 2001, pp. 638–656.
9. Sułek W.: Protograph Based Low-Density Parity-Check Codes Design with Mixed Integer Linear Programming. "IEEE Access", vol. 7, 2019, pp. 1424-1438.
10. IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems. IEEE Std 802.16e-2005.
11. IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012), 2016.
12. Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2). European Standard ETSI EN 302–307 V1.4.1, 2014.
13. 5G New Radio; Multiplexing and channel coding. ETSI TS 138 141-2 V15.3.0 (3GPP TS 38.141-2 version 15.3.0 Release 15), 2019.

Abstract

LDPC codes are one of the best error correction coding methods. In this article, we review the characteristic features of LDPC subclasses applied in important radiocommunication standards, such as WiFi, WiMAX, DVB-S2 and 5G. Then, we provide simulation results of selected codes taken from these standards, presenting comparison between different configurations.

5.2 Publikacja 2: Efficient LDPC Encoder Design for IoT-Type Devices.

applied sciences

MDPI

Article

Efficient LDPC Encoder Design for IoT-Type Devices

Jakub Hyla [1,2], Wojciech Sułek [1,*], Weronika Izydorczyk [1], Leszek Dziczkowski [1] and Wojciech Filipowski [1]

[1] Silesian University of Technology, 44-100 Gliwice, Poland; jakubhyla93@gmail.com (J.H.); weronika.izydorczyk@polsl.pl (W.I.); leszek.dziczkowski@polsl.pl (L.D.); wojciech.filipowski@polsl.pl (W.F.)
[2] TKH Technology Poland Sp. z.o.o, 64-100 Leszno, Poland
* Correspondence: wojciech.sulek@polsl.pl

Abstract: Low-density parity-check (LDPC) codes are known to be one of the best error-correction coding (ECC) schemes in terms of correction performance. They have been utilized in many advanced data communication standards for which the codecs are typically implemented in custom integrated circuits (ICs). In this paper, we present a research work that shows that the LDPC coding scheme can also be applied in a system characterized by highly limited computational resources. We present a microcontroller-based application of an efficient LDPC encoding algorithm with efficient usage of memory resources for the code-parity-check matrix and the storage of the results of auxiliary computations. The developed implementation is intended for an IoT-type system, in which a low-complexity network node device encodes messages transmitted to a gateway. We present how the classic Richardson–Urbanke algorithm can be decomposed for the QC-LDPC subclass into cyclic shifts and GF(2) additions, directly corresponding to the CPU instructions. The experimental results show a significant gain in terms of memory usage and decoding timing of the proposed method in comparison with encoding with the direct parity check matrix representation. We also provide experimental comparisons with other known block codes (RS and BCH) showing that the memory requirements are not greater than for standard block codes, while the encoding time is reduced, which enables the energy consumption reduction. At the same time, the error-correction performance gain of LDPC codes is greater than for the mentioned standard block codes.

Keywords: error correction; ECC; low-density parity-check codes; LDPC; QC-LDPC; IoT systems

Citation: Hyla, J.; Sułek, W.; Izydorczyk, W.; Dziczkowski, L.; Filipowski, W. Efficient LDPC Encoder Design for IoT-Type Devices. *Appl. Sci.* **2021**, *12*, 2558. https://doi.org/10.3390/app12052558

Academic Editor: Yosoon Choi

Received: 21 January 2022
Accepted: 23 February 2022
Published: 28 February 2022

1. Introduction

Internet of things (IoT) systems, which are composed of a large number of cheap, energy-efficient terminals, are some of the emerging elements of the recent landscape of information technology. The role of IoT has grown in numerous areas, such as smart homes, smart cities, Industry 4.0, agriculture, smart grids, public safety, etc. [1]. The general concept of IoT involves numerous possibilities in which a system's hardware and software can be designed. However, the IoT nodes, which constitute the bottom level of a system, are usually devices containing a set of sensors/actuators and a low-power processor with limited processing capabilities (a microcontroller) (e.g., [2,3]). Machine-to-machine communications over the Internet in an IoT system employ a variety of last-mile communication types, with wired, wireless, or fiber-optic mediums and numerous possible stacks of protocols for the physical layer. Since nodes are usually battery powered, the energy efficiency of the protocol and its implementation is an important issue. Despite the limited resources of IoT processor devices, the implementation of efficient communication protocols is required in order to enable communication with decent power efficiency.

One of the important elements of any communication protocol stack is an error-control coding (ECC) mechanism, which has a direct influence on the effectiveness of the communication. However, the implementation of modern ECC methods with great coding gain, such as Turbo coding [4], Polar coding [5], and low-density parity-check (LDPC) coding [6,7], is known to be computationally demanding. In this paper, we present

how one of the mentioned modern and power-efficient LDPC coding schemes can be implemented in a low-power microcontroller-based IoT device.

1.1. Relevant Research Works

While numerous hardware implementations of modern ECC codecs, including LDPC codecs, have already been developed, typically they are devoted to high-throughput demanding communication standards; thus, they are based on high-volume custom ICs [8–10], FPGAs [11–14], or GPU devices [15,16]. On the other hand, the typical physical-layer protocol stack for an IoT-type device is developed with low memory, device complexity and processing time requirements as the most important goals.

LDPC codes [6,7] are a class of linear block error-correcting codes that are known to achieve a performance near the capacity limit [17]. Extensive research in the field of LDPC coding includes issues such as graph–theoretic representations of LDPC codes [18,19], parity-check matrix design and construction methods [20–24], efficient encoding [25] and decoding [26–28] algorithm development, ECC coding system design for different use cases [29,30], hardware implementations in ASIC and FPGA equipment [9,11,13,31,32], and the design of nonbinary codes and codecs [22,33–36]. While LDPC codes have already been applied in a number of communication standards, such as WiMAX [37], Wi-Fi [38], and 5G [39], in this article, we show that LDPC coding can also be considered for devices with highly constrained computational resources. Despite the recent advances in digital circuit technologies, the process of codec design is still not trivial.

The research on LDPC efficient encoding implementation is quite rich already, with a number of issues that have been considered. For example, in [40], an efficient FPGA architecture for subclass of CCSDS (Consultative Committee for Space Data Systems) codes, recommended for channel coding in near-Earth and deep-space communications, is presented. The proposed architecture is based on a series of convolutional encoders, leveraging the QC structure inherent parallelism. In [41], LDPC encoding for a 5G new radio (NR) codes is considered, and a high-throughput encoder architecture is proposed. In [42], an encoding scheme is developed, involving pruning the full-base matrix of 5G NR code for a computationally efficient encoding in a high-throughput hardware architecture. The 5G NR code encoders are also considered in [16], where a GPU implementation is developed.

1.2. This Research Contributions

In the research presented in this paper, we investigate how one of the modern and power-efficient coding schemes—namely, a quasi-cyclic (QC) subclass of LDPC codes—can be applied in a device equipped with a microcontroller. The great error-correction performance of LDPC codes can then guarantee a power-efficient usage of the communications channel, regardless of the transmission medium and protocols used.

While the number of research works, including the mentioned above, involve high-throughput hardware implementations, there is a lack of LDPC codec development for the case of computationally constrained, low-power CPU devices. In this context, the research presented in this paper was initiated.

In this work, an IoT system is considered, in which the energy efficiency, due to the utilization of an advanced ECC coding, is mostly desired in an uplink direction, from the microcontroller-based IoT node, which is usually battery powered, to the IoT gateway. Therefore, the focus is on the LDPC encoding procedure, which is formulated and implemented in a low-cost CPU device. We present an efficient encoding scheme developed for a microcontroller implementation, with elementary operations corresponding directly to a CPU instructions, as well as the experimental results showing the computing time and energy reduction that can be then achieved. The contributions can be listed as follows:

1. We show how the Richardson–Urbanke encoding algorithm [43] utilized for QC-LDPC codes can be decomposed into simple, repeated operations of cyclic shifts and GF(2) additions.

2. We present a CPU implementation of this encoding with a QC-LDPC matrix representation that can be efficiently stored in the system's RAM.
3. We provide the experimental results of the implementation in comparison with other classic block ECC schemes: the RS (Reed–Solomon) and BCH (Bose–Chaudhuri–Hocquenghem) codes. Comparisons are made in terms of memory usage, block encoding time, energy savings and error-correction performance.

For an IoT system, short-to-moderate block length codes are likely to be utilized, due to the short message lengths; therefore, we do not consider the application of state-of-the-art 5G NR codes, but instead we base the experimental part on the proprietary codes designed with a graph optimization algorithm [24].

The paper is organized as follows. In Sections 2 and 3, we provide preliminary information about our system model of ECC in the context of IoT systems and basic definitions of LDPC coding. In Section 4, the Richardson–Urbanke encoding algorithm with a formulation specifically for QC-LDPC codes is presented. Then, in Section 5, we provide the details of the proposed microcontroller application. Finally, in Section 6, we provide the numerical results, and the paper is concluded in Section 7.

2. Error Correction for IoT End-Node Devices

We assume a communication model (Figure 1) in which a number of low-complexity IoT devices communicate with a central node—the gateway. Since we consider the gateway to possess great computational resources, an ECC scheme with a high decoding complexity can be considered for data blocks sent from devices to the gateway (the uplink direction). Therefore, LDPC coding seems to be an attractive option because of its great correction performance, which has the potential to save transmission power. However, in such a model, the LDPC encoder still needs to be implemented in an IoT node. In this article, we present the results of an investigation that aims to answer the question of how to optimize the LDPC encoder's implementation in a typical microcontroller, as well as to find out what the memory requirements and timing parameters of such a solution are.

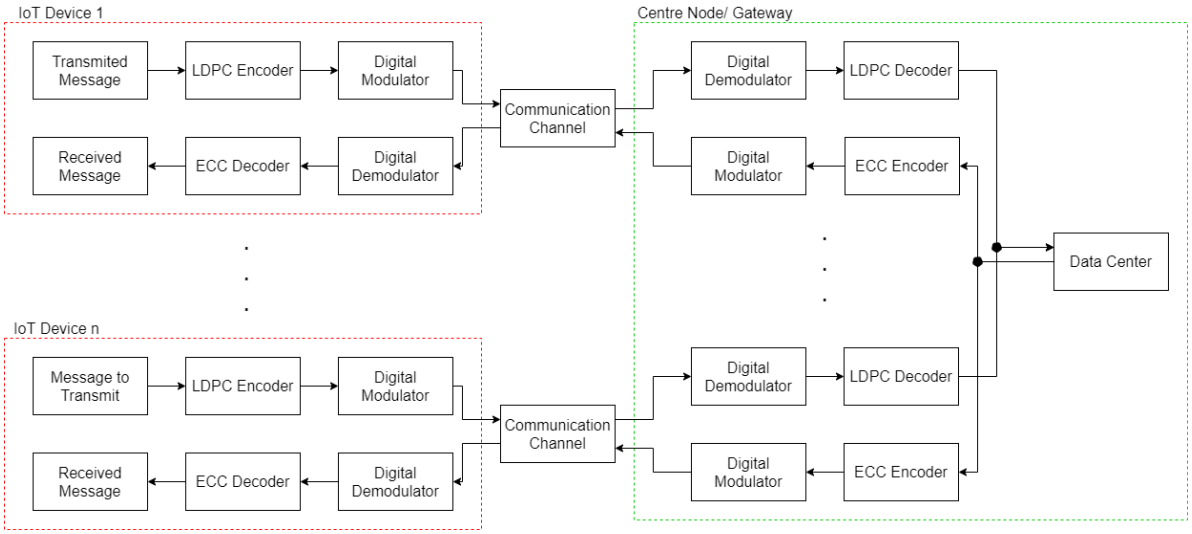


Figure 1. General architecture of IoT communication using LDPC codes for the error correction.

As shown in Figure 1, the presented solution is devoted to a microcontroller-based IoT device with any physical layer, wired or wireless, requiring computationally efficient and energy-efficient ECC. Therefore, the details of the protocol are abstracted as a digital modulation and communication channel in the system model considered. The developed coding system is utilized in an industrial implementation with a fiber-optic communication

link, with highly constrained optical power. This is an example of practical application of the presented ideas.

IoT device nodes are typically based on small microcontrollers with limited resources and peripherals, such as a universal asynchronous receiver–transmitter (UART), SPI and I2C serial interfaces, etc. Some of these peripherals are configured to work with very-low-power modes; most of the time, IoT devices are in sleep mode, waiting for a wake-up, and then send a message upon the occurrence of some event. Considering ECC message encoding, the energy efficiency of such a message delivery is dependent on two factors:

- The encoding time, which is connected with encoding complexity;
- The coding scheme gain, which is the possible reduction in transmission power due to the application of an effective ECC scheme.

These two factors are likely contradictory because an effective ECC scheme that enables transmission power reduction usually has a high encoding—and especially decoding—complexity. Therefore, we propose the utilization of an LDPC coding scheme in the uplink direction, and we present how an LDPC encoder can be efficiently realized with a microcontroller. As presented in Figure 1, the assumed system model does not fix any specific ECC scheme for the downlink direction (from the gateway to the IoT node).

3. LDPC Codes

3.1. Preliminaries

A binary (N, K) LDPC code is defined by its binary parity-check matrix $\mathbf{H}$ of size $M \times N$, where N is the code vector length, $M = N - K$ is the number of parity-check bits, and $K < N$ is the length of an information vector. The code redundancy is indicated by the code rate, which is defined as $R = K/N$. In the systematic encoding process, M parity bits are added to the information vector $\mathbf{u} = \{u_1, u_2, \ldots, u_K\}$, forming the code vector (code word) $\mathbf{c} = \{c_1, c_2, \ldots, c_N\}$. In the mathematical formulation, the vectors are over the Galois field GF(q)—specifically, GF(2) for the binary codes considered in this paper.

In the decoder, the received data in a row vector $\hat{\mathbf{c}}$ of length N are recognized as a correct (error-free) vector if, and only if, they satisfy the parity-check equation $\mathbf{H}\hat{\mathbf{c}}^T = \mathbf{0}_{M\times 1}$ with GF(2) arithmetic. If the parity-check equation is not satisfied, error-correction decoding is required, which is applied by means of the iterative belief propagation algorithm [7].

To be effectively and efficiently utilized in an IoT system, the LDPC code should usually satisfy requirements regarding the code rate $R = K/N$, the code-word length N, decent error correction performance, and the possibility of implementing an encoder using very limited resources. Our proposed encoding scheme is versatile; that is, a broad range of N and R can be engaged, and we will present the results for a number of parity-check matrices with different N and R. Unlike the frequently considered advanced communication devices based on custom ICs and FPGAs, we consider an IoT device with highly limited resources: limited memory and only a single computation core (CPU).

3.2. QC-LDPC Codes

The quasi-cyclic subclass of low-density parity check codes (QC-LDPC) [44] is known to possess a desirable feature of organized message routing in hardware implementations of a partially parallel decoder. QC-LDPC codes are used for most LDPC coding system designs, industrial applications [11,22,37,39,44], etc. Their characteristic feature is the block-circulant structure of $\mathbf{H}$. The parity-check matrix $\mathbf{H}$ of a QC-LDPC code consists of a grid of square $P \times P$ submatrices, with each submatrix being the following:

- An all-zero zero matrix;
- An identity matrix;
- An identity matrix with circularly shifted columns.

Additionally, the implementation of a low-complexity encoder using the classic Richardson–Urbanke method [43] requires an ALT (almost lower triangular) form, preferably a dual-diagonal structure of $\mathbf{H}$ [45]. Let us denote a $P \times P$ circulant permutation matrix

(CPM) as $\mathbf{P}^s$, which is obtained by cyclically shifting the identity matrix $\mathbf{I}_{P\times P}$ by s positions to the right. In this article, a class of codes with the following structure of the parity-check matrix $\mathbf{H}$ is considered:

$$\mathbf{H} = \begin{bmatrix} \mathbf{P}^{s_{1,1}} & \mathbf{P}^{s_{1,2}} & \dots & \mathbf{P}^{s_{1,J}} \\ \mathbf{P}^{s_{2,1}} & \mathbf{P}^{s_{2,2}} & \dots & \mathbf{P}^{s_{2,J}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}^{s_{I,1}} & \mathbf{P}^{s_{I,2}} & \dots & \mathbf{P}^{s_{I,J}} \end{bmatrix}, \tag{1}$$

where $s_{i,j} \in \{0, 1, \dots, P-1, \infty\}$, where $\mathbf{P}^{\infty} = \mathbf{0}$ is used to denote the all-zero matrix [45]. Codes associated with the parity-check matrix as in (1) are (N, K) QC-LDPC codes with code vector length $N = JP$, information vector length $K = (J-I)P$, and rate $R = (J-I)/J$.

Alternatively to the matrix representation, the LDPC codes can also be represented by the associated bipartite Tanner graph [7,46]. In the Tanner graph, variable nodes representing data bits are associated with columns of $\mathbf{H}$, and check nodes representing parity checks are associated with rows of $\mathbf{H}$. The nonzero elements in $\mathbf{H}$ are represented by the edges in the code graph. The Tanner graph defines the passage of the message through the iterative decoding algorithm [7], but it is also involved in numerous methods of code construction with optimized performance [20,47,48].

4. Richardson–Urbanke Encoding Algorithm

A basic encoding method for any linear block codes can be based on a generator matrix. However, in general, the generator matrix of an LDPC code is a dense $N \times K$ matrix, which in itself is a great amount of data for storing in limited memory, and it requires dense matrix operations. Therefore, it is desirable to employ a more efficient LDPC encoding method, as introduced by Richardson and Urbanke [43], who directly utilized the parity-check matrix $\mathbf{H}$, which is sparse by definition. This method requires an $\mathbf{H}$ matrix in an ALT form. Let us shortly introduce the Richardson–Urbanke method in the context of QC-LDPC codes.

We assume that by making use of some of the known methods [20,24], the parity-check matrix is constructed in (or converted into) an ALT form, with $P \times P$ submatrices of the QC-LDPC structure. Let the $\mathbf{H}$ in ALT form be partitioned as follows:

$$\mathbf{H} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{T} \\ \mathbf{C} & \mathbf{D} & \mathbf{E} \end{bmatrix}, \tag{2}$$

where $\mathbf{A}$ is of size $(I-g)P \times (J-I)P$, $\mathbf{B}$ is $(I-g)P \times gP$, $\mathbf{T}$ is $(I-g)P \times (I-g)P$, $\mathbf{C}$ is $gP \times (J-I)P$, $\mathbf{D}$ is $gP \times gP$, and $\mathbf{E}$ is $gP \times (I-g)P$, where g is the gap size, that is, the size of the non-triangular part. The structure of the $\mathbf{T}$ matrix is essential, which should be lower triangular, that is,

$$\mathbf{T} = \begin{bmatrix} \mathbf{P}^{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{P}^{s_{2,(J-I+g+1)}} & \mathbf{P}^{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{P}^{s_{3,(J-I+g+1)}} & \mathbf{P}^{s_{3,(J-I+g+2)}} & \mathbf{P}^{0} & & \mathbf{0} \\ \vdots & & \ddots & & \vdots \\ \mathbf{P}^{s_{(I-g),(J-I+g+1)}} & \mathbf{P}^{s_{(I-g),(J-I+g+2)}} & \dots & \mathbf{P}^{s_{(I-g),(J-1)}} & \mathbf{P}^{0} \end{bmatrix} \tag{3}$$

where $\mathbf{P}^0$ is an identity matrix of size $P \times P$. An important feature is that $\mathbf{T}^{-1}$ multiplication by any vector can be performed with linear complexity through a back-substitution operation [43].

The systematic encoding process of an information vector $\mathbf{u} = \{u_1, u_2, \dots, u_K\}$ is based on determining two parity sub-vectors $\mathbf{p}_1$ and $\mathbf{p}_2$ with lengths gP and $(I-g)P$, respectively, such that the obtained code vector is $\mathbf{c} = [\mathbf{u}, \mathbf{p}_1, \mathbf{p}_2]$. Considering (2), the parity-check equation $\mathbf{H}\mathbf{c}^T = \mathbf{0}$ can be divided into two parts:

$$\mathbf{A}\mathbf{u}^T + \mathbf{B}\mathbf{p}_1^T + \mathbf{T}\mathbf{p}_2^T = \mathbf{0}, \quad (4)$$

$$\mathbf{C}\mathbf{u}^T + \mathbf{D}\mathbf{p}_1^T + \mathbf{E}\mathbf{p}_2^T = \mathbf{0}. \quad (5)$$

After simple algebraic transformations with GF(2) arithmetic, the expressions for calculating $\mathbf{p}_1$ and $\mathbf{p}_2$ are obtained:

$$\mathbf{p}_1^T = \mathbf{\Phi}^{-1}(\mathbf{E}\mathbf{T}^{-1}\mathbf{A}\mathbf{u}^T + \mathbf{C}\mathbf{u}^T), \quad (6)$$

$$\mathbf{p}_2^T = \mathbf{T}^{-1}(\mathbf{A}\mathbf{u}^T + \mathbf{B}\mathbf{p}_1^T), \quad (7)$$

where $\mathbf{\Phi} = \mathbf{E}\mathbf{T}^{-1}\mathbf{B} + \mathbf{D}$. The multiplication by the sparse sub-matrices **A**, **B**, **C**, **E**, and $\mathbf{T}^{-1}$, which is realized through back-substitution, is an operation with complexity that is linearly dependent on the code length N. More importantly, as will be presented below, for QC-LDPC matrices, as in (1), the encoding can be implemented with just two elementary operations: the GF(2) addition and circular shifting of sub-vectors.

However, straightforward multiplication by $\mathbf{\Phi}^{-1}$ can be problematic. The first reason is that $\mathbf{\Phi}$ is not guaranteed to be an invertible matrix. In such a case, in order to use the Richardson–Urbanke encoding method, **H** must be rearranged by row and column permutations such that an invertible $\mathbf{\Phi}$ is achieved. The second reason is that $\mathbf{\Phi}^{-1}$ is not sparse in general, and the complexity of the multiplication is quadratic on N. This is (for large block sizes) the most computationally demanding part of the encoder. However, it was shown that for QC-LDPC codes with a dual diagonal structure, $\mathbf{\Phi}$ can be reduced to an identity matrix, eliminating the need for multiplication by $\mathbf{\Phi}^{-1}$ [45]. The additional constraint on the **T** and **E** submatrices is that only nonzero elements (submatrices in QC-LDPC) are organized in a dual diagonal in **T** [45]. In the article [45], it was shown how circular shifts $s_{i,j}$ in matrices **B** and **D** of the dual-diagonal QC-LDPC **H** could be calculated to make $\mathbf{\Phi}$ an identity matrix. Dual-diagonal codes are used in several industrial standards, e.g., WiMAX [37] and Wi-Fi [38].

5. LDPC Encoder Design for a Microcontroller Implementation

The application of dual-diagonal QC-LDPC codes with Richardson–Urbanke encoding can be an attractive option for the transmission system under consideration. In this section, we present a developed implementation of an efficient QC-LDPC encoder for a computational core of a typical microcontroller device.

Similarly to the majority of embedded software solutions, the experimental application is implemented in the C language using the GCC (GNU compiler collection) compiler delivered with an IDE (independent development environment) from STMicroelectronics, version for STM32–9-2020-q2-update. The encoder is compiled for Cortex-M4 core, with optimization for size (-Os flag), which increases the possible size of applicable parity check matrices. All drivers for HAL (hardware abstraction level) were delivered by ST company. The experimental hardware is based on the STM32L476 microcontroller; however, the proposed solution can be adapted for other devices from numerous vendors. The device utilized for experiments has 1 MB of flash memory and 128 KB of static RAM, operating with an 80 MHz clock frequency.

5.1. Elementary Computations

In this section, we present how the encoding process can be decomposed into elementary operations that are suitable for a typical microcontroller CPU, as well as how the code specification, i.e., the structured parity-check matrix, can be efficiently memorized.

An efficient sparse matrix representation of **H** can utilize indexes of nonzero elements, $\mathrm{ind}(\mathbf{H})$, which is a matrix with the same number of rows as **H**, but only a few columns with indexes of nonzero elements in every row of **H**. For regular codes, with every row of **H** that has weight d_c, this representation requires the storage of $M \times d_c$ index

values. However, for the QC-LDPC codes applied in our research, an even more efficient representation is possible, similar to hardware implementation of, for example, [32], in which **H** is represented by the following:

- Indexes of nonzero submatrices $\mathbf{P}^{s_{i,j}}$ in (1);
- Circular shift values $s_{i,j}$ in (1) gathered in matrix representation **S**.

This way, the storage is reduced to $(M/P) \times d_c$ indexes plus $(M/P) \times d_c$ circular shift values, which gives, in total, $2(M/P) \times d_c$ integer values, which can often fit into an 8-bit (`uint8_t`) representation if $P < 256$ and $N/P < 256$.

In our implementation, the Richardson–Urbanke algorithm, according to expressions (6) and (7), is partitioned into steps that are realized sequentially in a CPU, as we illustrate in Figure 2.

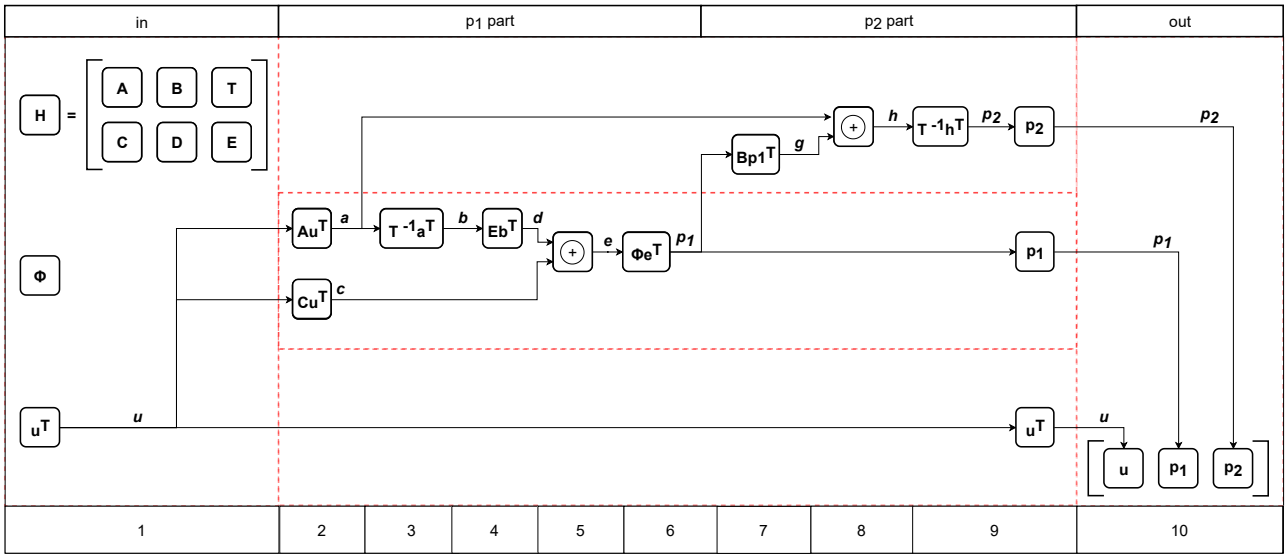


Figure 2. Block diagram presenting consecutive steps in the efficient encoding algorithm.

The first step is the preparation of the input information vector $\mathbf{u}^T$, that is, buffering the incoming data into a vector $\mathbf{u}^T$ of length K. Then, in the following steps, intermediate-result vectors denoted by $\mathbf{a}, \mathbf{b}, \ldots, \mathbf{h}$, as well as final parity bit vectors $\mathbf{p}_1, \mathbf{p}_2$ are calculated sequentially. The required partial operations can be categorized into three types:

- Sparsely structured matrix by vector multiplication over GF(2), e.g., $\mathbf{a} := \mathbf{A}\mathbf{u}^T$;
- Inverse of a lower-triangular sparse matrix by vector multiplication over GF(2), e.g., $\mathbf{b} := \mathbf{T}^{-1}\mathbf{a}^T$;
- Addition over GF(2), e.g., $\mathbf{e} := \mathbf{c} + \mathbf{d}$.

Let us discuss these types of operations, beginning with the sparsely structured matrix by vector multiplication. The expression is

$$\mathbf{a} := \mathbf{A}\mathbf{u}^{\mathbf{T}} \tag{8}$$

For the QC-LDPC codes with a structured parity-check matrix, as in (1), this can be formulated as follows:

$$\begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_{I-L}^T \end{bmatrix} := \begin{bmatrix} \mathbf{P}^{s_{1,1}} & \cdots & \mathbf{P}^{s_{1,J}} \\ \mathbf{P}^{s_{2,1}} & \cdots & \mathbf{P}^{s_{2,J}} \\ \vdots & \ddots & \vdots \\ \mathbf{P}^{s_{I-L,1}} & \cdots & \mathbf{P}^{s_{I-L,J}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{u}_1^T \\ \mathbf{u}_2^T \\ \vdots \\ \mathbf{u}_{I-L}^T \end{bmatrix} = \begin{bmatrix} \mathbf{P}^{s_{1,1}}\mathbf{u}_1^T & \cdots & \mathbf{P}^{s_{1,J}}\mathbf{u}_{I-L}^T \\ \mathbf{P}^{s_{2,1}}\mathbf{u}_1^T & \cdots & \mathbf{P}^{s_{2,J}}\mathbf{u}_{I-L}^T \\ \vdots & \ddots & \vdots \\ \mathbf{P}^{s_{I-L,1}}\mathbf{u}_1^T & \cdots & \mathbf{P}^{s_{I-L,J}}\mathbf{u}_{I-L}^T \end{bmatrix} \tag{9}$$

where $\mathbf{u}_1$, $\mathbf{u}_2, \ldots$ and $\mathbf{a}_1$, $\mathbf{a}_2, \ldots$ are the subvectors of, respectively, $\mathbf{u}$ and $\mathbf{a}$ of size $1 \times P$. In (9), every subvector calculation of

$$\mathbf{a}_i^T := \mathbf{P}^{s_{i,1}}\mathbf{u}_1^T + \mathbf{P}^{s_{i,2}}\mathbf{u}_2^T + \ldots + \mathbf{P}^{s_{i,J-I}}\mathbf{u}_{J-I}^T \tag{10}$$

requires the performance of elementary operations of the type $\mathbf{P}^{s_{i,j}}\mathbf{u}_j^T$, as well as additions over GF(2). Every $\mathbf{P}^{s_{i,j}}$ for the QC-LDPC code is one of the following:

- A zero matrix ($s_{i,j} = \infty$); then, $\mathbf{P}^{s_{i,j}}\mathbf{u}_j^T = \mathbf{0}$.
- A circular shift of an identity matrix by $s_{i,j}$; then it can easily be verified that $\mathbf{P}^{s_{i,j}}\mathbf{u}_j^T$ is a circular shift of $\mathbf{u}_j^T$ by $s_{i,j}$ positions.

Therefore, computing (10) for QC-LDPC codes requires elementary operations of circular shifts of subvectors of $\mathbf{u}^T$ of length P, as well as modulo-2 additions of subvectors of length P, which are equivalent to additions over GF(2). Similar elementary operations can be indicated for the other sparsely structured matrices by vector multiplications in Figure 2.

The $\mathbf{T}^{-1}$ by vector multiplication, e.g., $\mathbf{b^T} := \mathbf{T}^{-1}\mathbf{a}^T$, can be realized due to the equivalency:

$$\mathbf{b}^T = \mathbf{T}^{-1}\mathbf{a}^T \Leftrightarrow \mathbf{T}\mathbf{b}^T = \mathbf{a}^T \tag{11}$$

which, in the matrix form, is expressed as

$$\begin{bmatrix} \mathbf{P}^0 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{P}^{s_{2,(J-I+g+1)}} & \mathbf{P}^0 & \mathbf{0} & \cdots & 0 \\ \vdots & & \ddots & \vdots & \\ \mathbf{P}^{s_{(I-g),(J-I+g+1)}} & \mathbf{P}^{s_{(I-g),(J-I+g+2)}} & \mathbf{P}^{s_{(I-g),(J-I+g+3)}} & \cdots & \mathbf{P}^0 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{b}_1^T \\ \mathbf{b}_2^T \\ \vdots \\ \mathbf{b}_{I-L}^T \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_{I-L}^T \end{bmatrix} \tag{12}$$

Examining (12), it can be noticed that subvectors $\mathbf{b}_1, \mathbf{b}_2, \ldots$ can be subsequently computed. First, $\mathbf{b}_1$ is computed according to:

$$\mathbf{P}^0\mathbf{b}_1^T = \mathbf{a}_1^T \;\Rightarrow\; \mathbf{b}_1^T = \mathbf{a}_1^T \tag{13}$$

Then, $\mathbf{b_2}$ is computed according to:

$$\mathbf{P}^{s_{2,(J-I+g+1)}}\mathbf{b_1^T} + \mathbf{P}^0\mathbf{b_2^T} = \mathbf{a_2^T} \;\Rightarrow\; \mathbf{b_2^T} = (\mathbf{P}^{s_{2,(J-I+g+1)}})\mathbf{b_1^T} + \mathbf{a_2^T} \tag{14}$$

and then every subsequent ith subvector $\mathbf{b}_i^T$ is computed according to the following expression:

$$\begin{gathered} \mathbf{P}^{s_{i,(J-I+g+1)}}\mathbf{b}_1^T + \mathbf{P}^{s_{i,(J-I+g+2)}}\mathbf{b}_2^T + \cdots + \mathbf{P}^0\mathbf{b}_i^T = \mathbf{a}_i^T \;\Rightarrow \\ \mathbf{b}_i^T = \mathbf{P}^{s_{i,(J-I+g+1)}}\mathbf{b}_1^T + \mathbf{P}^{s_{i,(J-I+g+2)}}\mathbf{b}_2^T + \cdots + \mathbf{P}^{s_{i,(J-I+g+i-1)}}\mathbf{b}_{i-1}^T + \mathbf{a}_i^T \end{gathered} \tag{15}$$

This last expression shows that computing every subsequent subvector $\mathbf{b}_i^T$ requires only circular shift operations of previously calculated subvectors $\mathbf{b}_1^T, \ldots, \mathbf{b}_{(i-1)}^T$ and additions over GF(2).

The subvector additions over GF(2) are equivalent to the vector-wise binary exclusive-or (XOR) operations. The last step of encoding, as shown in Figure 2, is assembling a code vector **c** as follows:

$$\mathbf{c} = \begin{bmatrix} \mathbf{u} & \mathbf{p_1} & \mathbf{p_2} \end{bmatrix} \tag{16}$$

We can conclude that the whole encoding process can be partitioned into the following elementary operations:

- Circular shift operations of subvectors of length P;
- Subvector additions over GF(2) realized with XOR operations.

We remark that a classic encoding that employs a generator matrix can also be realized with just GF(2) additions, but the generator matrix representation is not sparse. Meanwhile, the presented formulation operates directly on a sparse parity-check matrix; therefore, many of the submatrices are all-zero, thus significantly reducing the computational complexity.

All vectors computed as intermediate results and their sizes are summarized in Table 1. During the operation of the encoding algorithm, the system RAM can be dynamically interchanged for the storage of the intermediate vectors; for example, after step 2, vectors **a** and **c** are memorized, after step 3, vectors **a**, **b**, and **c** are memorized, and so on, as can be easily read in Figure 2. It can be verified that the greatest memory consumption is after step 7, when the following vectors are memorized: **a**, **g**, $\mathbf{p}_1$, and **u**. The memory utilization is quantified by experimental results in the next section.

Table 1. Comparison of intermediate computation vector sizes.

Vector	Length	Vector	Length
a	$(I-g)P$	**u**	$(J-I)P$
b	gP	**g**	$(I-g)P$
c	gP	**h**	$(I-g)P$
d	gP	$\mathbf{p}_1$	gP
e	gP	$\mathbf{p}_2$	$(I-g)P$

5.2. Parity-Check Matrix Representations in Memory

Basically, three different representation methods can be considered. The first method is the direct representation with a `Boolean` variable table storing the binary parity-check matrix, as shown on the left side of Figure 3 (basic representation). Direct parity-check matrix storage would require an unnecessarily large amount of memory, e.g., 524,288 B for a 512×1024 matrix.

The second method is an index representation, in which indexes of nonzero elements in every row are memorized, which is shown in the middle of Figure 3, where -1 indicates a lack of nonzero indexes in the macro-column. The third method is the representation of circular shift values, as shown on the right side of Figure 3 (efficient QC-LDPC), where all-zero matrices are indicated by -1 and non-zero CPMs are represented by their cyclic shift values, which are stored in an `integer` table.

We claim that the third (shift values) representation is the most suitable for our implementation, not only because it utilizes memory efficiently, but also because it explicitly defines the circular shift values introduced in the previous section as the components of elementary encoding operations in the described algorithm.

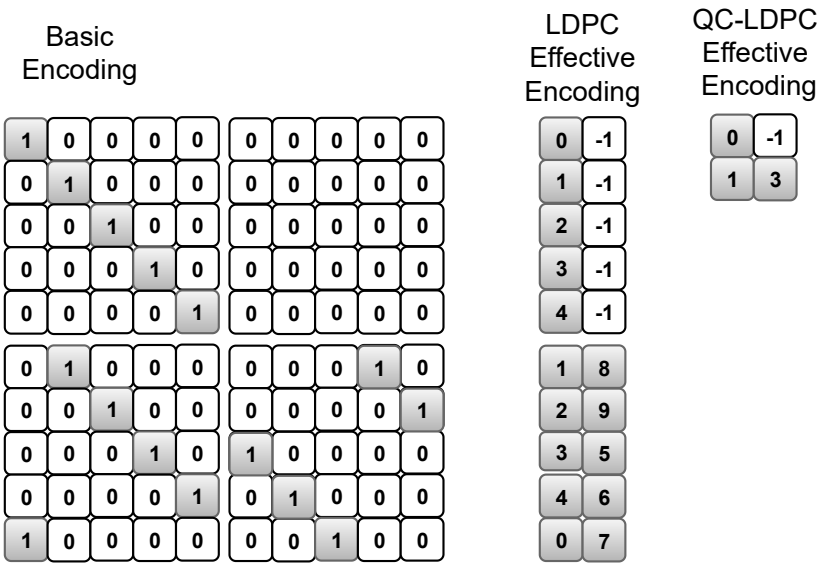


Figure 3. Illustration of the size of the **H** matrix with three different representations in the microcontroller memory.

5.3. Implementation of Elementary Operations

The elementary operations of the circular shift of QC-LDPC encoding algorithm can be implemented in the C language by using a bit-wise shift operation and bit-wise OR operation, as in the following example:

```
output = (input << offset) | (input >> (size - offset));
```

where `offset` is the cyclic shift value and `size` is equivalent to the submatrix size P.

Using this instruction for the circular shift, the input value needs to be of the `uint32_t` type (if $P \leq 32$), while the `size` and `offset` can usually be of the `uint8_t` type. The output value is in the same format as the input. An example for shifting 8-bit values with `offset` 2 is presented in Figure 4.

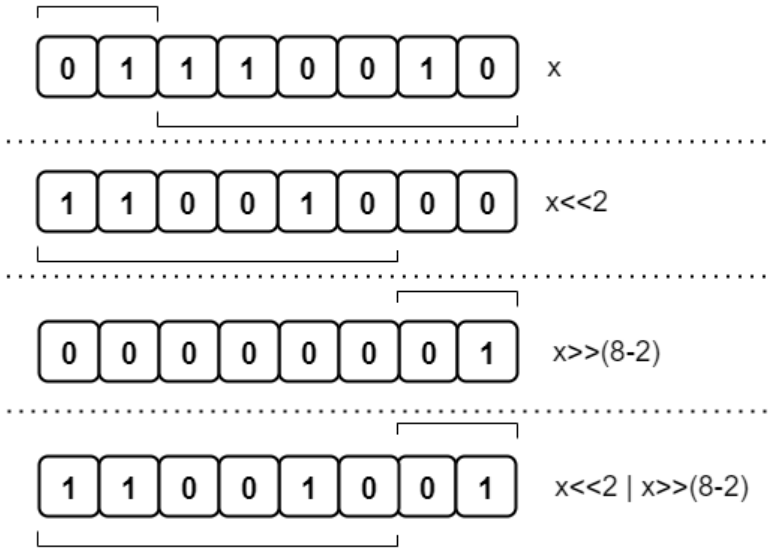


Figure 4. Example of a circular shift operation.

The possible circular shift is limited to $P \leq 32$, because the typical microcontrollers use 32-bit cores, and hence the elementary operations are efficiently executed for up to 32b

subvector sizes. However, this limitation does not influence the achievable performance, because the LDPC codes with short-to-moderate submatrix sizes can be optimized at least as effectively as the codes with greater submatrix sizes if the codeword length remains the same [24].

The second elementary operation, the addition of the GF(2) subvectors, can be realized directly with an XOR operator in the C language applied to variables representing the subvectors.

5.4. Microcontroller Implementation Issues

We implemented the encoder for LDPC codes with a broad range of code vector and information vector sizes (N, K). The encoder operates according to the block diagram shown previously in Figure 2.

The algorithm implemented in the microcontroller (the low-power STM32L476 mentioned above was used in the experimental design) was divided into four parts: input data preparation, two parts for calculating the $\mathbf{p}_1$ and $\mathbf{p}_2$ vectors, and the output assembly. The parity-check matrix **H** was divided into **A**, **B**, **T**, **C**, **D**, and **E** for the LDPC and QC-LDPC encoding, and this way, it was stored in the memory. For the experimental comparison, we implemented encoding for both LDPC codes with index representation and for QC-LDPC with cyclic shift representation. The **H** matrix was a constant initialized variable and was stored in flash memory.

In the first part of the algorithm, data were prepared to start the encoding procedure. The information vector **u** is a parameter of the encoding function and stored in the stack; likewise, every buffer and variable during the encoding process is stored in stack memory in the RAM. The available RAM memory in the current microcontroller is 128 KB; thus, the operations needed to use memory efficiently. The buffer usage was minimized; in most cases, one main and one temporary buffer were used to execute the operations.

6. Numerical Results

Our research is supported by the experimental results, which cover three aspects: memory utilization, encoding time, and error-correction performance. The optimized memory enables the use of smaller, lower-cost, and more energy-efficient microcontroller devices; optimized timing can potentially decrease energy consumption, and likewise, the error-correction performance, which is greater than in the case of typical block codes, can enable lower transmission power in wired or wireless transmission systems.

The QC-LDPC codes used for encoder verification and system performance simulation were obtained with a code graph optimization method, consisting of two phases:

1. Base graph construction, that is, a graph corresponding to the $I \times J$ base matrix **W** indicating non-zero submatrices in **H** in (1).
2. Cyclic lifting of the base graph, with cyclic shift values $s_{i,j}$ in (1) obtained with an iterative search algorithm.

The code construction algorithm is summarized as Algorithm 1.

Several different parity-check matrices (**H**1, . . . , **H**8) with diversified block lengths, regular and irregular, as listed in Table 2, were used for the experimental study. For example, the parity-check matrix of a regular $(3, 6)$, $(1024, 512)$ code with submatrix size $P = 32$ is provided in Figure 5. This matrix, referred to as H1 in Table 2, corresponds to an optimized graph, free of length-4 and length-6 cycles and a minimized number of length-8 cycles.

Algorithm 1: QC-LDPC parity-check matrix construction.

Input: Base matrix size $I \times J$, submatrix size P, variable nodes degree distribution $\boldsymbol{\lambda} = [\lambda_2, \lambda_3, \ldots, \lambda_{d_{V\max}}]$

Output: Binary matrix $\mathbf{H}_{IP \times JP}$ consisting of $I \times J$ circulants of size $P \times P$

{Base matrix construction}

If possible (that is, if $\lambda_2 \geq I/J$), insert dual diagonal at the right side of the base matrix $\mathbf{W}$.

Utilizing the PEG algorithm [20], progressively add edges into the base graph (ones into the remaining part of $\mathbf{W}$), subject to the degree distribution $\boldsymbol{\lambda}$.

{Cyclic shift values determination}

If $\mathbf{W}$ is not dual diagonal, then by row and column permutations, convert it to the ALT (almost lower triangular) form.

If $\mathbf{W}$ is dual diagonal, then fix the cyclic shift values on dual diagonal such that $\boldsymbol{\Phi} = \mathbf{I}$, according to the expressions in [45].

In the base graph corresponding to $\mathbf{W}$, identify all short closed walks (if the length of the shortest cycle is g, then closed walks up to length at least $g + 4$ should be identified).

For every closed walk, formulate a condition involving cyclic shift values $s_{i,j}$ such that corresponding cycles in the lifted code graph do not exist [24].

For every condition, iteratively modify associated $s_{i,j}$ values in order to satisfy the greatest possible number of conditions for the non-existence of corresponding cycles.

{Matrix construction}

From the obtained $\mathbf{W}$ and $s_{i,j}$, $i = 1, \ldots, I$, $j = 1, \ldots, J$, construct $\mathbf{H}$ as in (1).

The parameters of the RS and BCH codes, which are counterparts of LDPC $\mathbf{H}1, \ldots, \mathbf{H}8$, are also presented in Table 2. These codes were used for an experimental comparison of the developed QC-LDPC scheme with classical block codes. Every RS code is clearly defined by its (N, K) parameters and for BCH codes, we also provide the correction capability parameter t_{BCH}, defined as in [49]. The generator polynomials of RS and BCH codes are obtained as described in [49].

		0			7		6		0		2							0												
11	4						15			1		0								0										
0					3			8								6		4			0									
						8							1						31	8	20	0									
				13								4		0		2			5				0							
	4	2	0														11				19			0						
0											12											0	29	5	0					
			24			0				19								3							8	0				
	0												4		4								28			1	0			
		5						0		2				4													26	0			
									14						0		11			1							29		0		
			5		4							7	4															0	0	
				1				0																11	29			9		0
							1							10								27				6			10	29
						1			1						6	3											31			2
				4						5						3	1									8		3		

Figure 5. Parity-check matrix of a (1024,512) QC-LDPC code with submatrix size $P = 32$. The cyclic shift values are provided for non-zero submatrices.

Table 2. Parameters of the experimental parity-check matrices **H** for LDPC. The λ_d represents the fraction of degree-d variable nodes in the code graph.

Matrix	K	N	P	R	Distribution	$N_{RS,BCH}$	K_{RS}	R_{RS}	K_{BCH}	t_{BCH}	R_{BCH}
H1	512	1024	32	$\frac{1}{2}$	Reg., $\lambda_3 = 1$	1023	511	$\approx \frac{1}{2}$	-	-	-
H2	512	1024	32	$\frac{1}{2}$	Irreg., $\lambda_{2,3,7}$	1023	511	$\approx \frac{1}{2}$	-	-	-
H3	256	512	32	$\frac{1}{2}$	Reg., $\lambda_3 = 1$	511	255	$\approx \frac{1}{2}$	259	30	$\approx \frac{1}{2}$
H4	256	512	32	$\frac{1}{2}$	Irreg., $\lambda_{2,3,6}$	511	255	$\approx \frac{1}{2}$	259	30	$\approx \frac{1}{2}$
H5	320	512	32	$\frac{5}{8}$	Reg., $\lambda_3 = 1$	511	321	$\approx \frac{5}{8}$	322	22	$\approx \frac{5}{8}$
H6	384	512	32	$\frac{3}{4}$	Reg., $\lambda_3 = 1$	511	383	$\approx \frac{3}{4}$	385	14	$\approx \frac{3}{4}$
H7	144	288	16	$\frac{1}{2}$	Reg., $\lambda_3 = 1$	255	143	$\approx \frac{1}{2}$	147	14	$\approx \frac{1}{2}$
H8	144	288	16	$\frac{1}{2}$	Irreg., $\lambda_{2,3,6}$	255	143	$\approx \frac{1}{2}$	147	14	$\approx \frac{1}{2}$

The RS encoder we use for comparison is a byte-size (8 bit) LSFR (linear-feedback shift register) implementation, following the source codes provided in [50], while the BCH encoder we base on the description provided in [51].

The encoding process in each algorithm is based on three steps: data preparation, encoding algorithm execution and storing the results. The encoding in all cases is based mostly on operations: XOR, OR and register shift. The basic LDPC, RS and BCH algorithms need to execute a similar order of the number of operations. Meanwhile, the inherent advantage of QC-LDPC is due to the following reasons:

- The encoding combines operation for a blocks of P bits because multiplication by a whole submatrix can be done in a single shift operation;
- The QC-LDPC parity-check matrix, which can be used directly for encoding, is by definition sparse; therefore, the number of multiplications by submatrices is also relatively small.

This explains the reduction in time processing, which is shown in the following results.

6.1. Memory Utilization

Figure 6 shows the memory utilization observed in our experimental framework for eight different LDPC parity-check matrices (H1, . . . , H8), as well as their Reed–Solomon (RS) and Bose–Chaudhuri–Hocquenghem (BCH) counterpart block codes (with similar block lengths and code rates R). Three LDPC encoder implementation methods were investigated. The basic implementation refers to the direct **H** matrix stored in the Boolean table. The memory requirement is too huge to be applicable with the STM32 device mentioned here due to the stack overflow error.

The second solution, referred to as an efficient LDPC, is based on the index representation of the **H** matrix and the developed encoding algorithm written in the C programming language. The algorithm is versatile. The memory utilization is roughly 20 times less than with the basic representation.

Finally, the third solution, marked with the blue bar in Figure 6, involves the developed QC-LDPC encoding algorithm and the circular shift value representation of **H**. This algorithm is available to encode only QC-LDPC parity-check matrices, but with a broad range of code lengths and rates. The significant memory reduction in the proposed solution was confirmed by the experimental results. It can also be observed that the memory utilization was not greater than that of the RS encoder counterpart implemented with

the same microcontroller equipment. In all the presented charts, the proposed encoder is shortly denoted as QC-LDPC, while the index representation solution is denoted as LDPC.

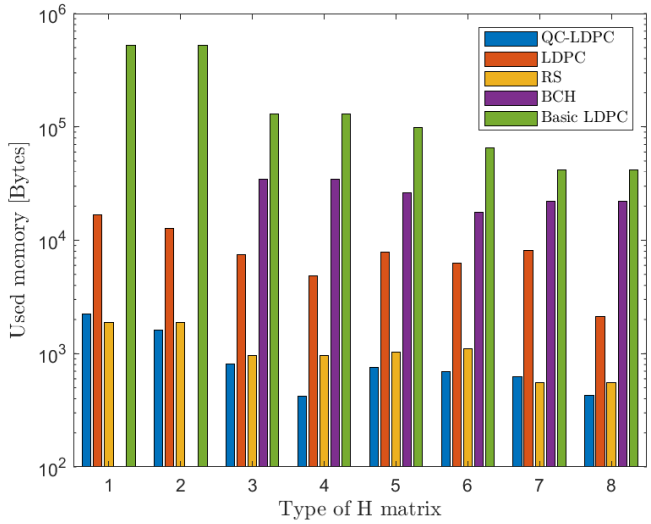


Figure 6. Memory utilization of the implemented encoders.

6.2. Block Encoding Time

A similar comparison was made for the encoding time measurements. Figure 7 presents the total encoding time of a single block for different parity-check matrices. The measured time involved the data preparation, calculation of parity bits $\mathbf{p_1}$, $\mathbf{p_2}$, and gathering of the output vector. We included results for standard LDPC encoding with the index representation (orange bar), as well as the proposed efficient QC-LDPC encoding (blue bar). A significant encoding time reduction was visible in the case of the developed efficient QC-LDPC algorithm implementation in comparison with the LDPC with the index matrix representation, as well as the Reed–Solomon (RS) encoders. This implementation thus has potential for the reduction in energy consumption in solutions that use sleep modes after the communication of a message from an IoT node. The relation of the working mode time to the sleep mode time can be reduced if the block encoding time is reduced.

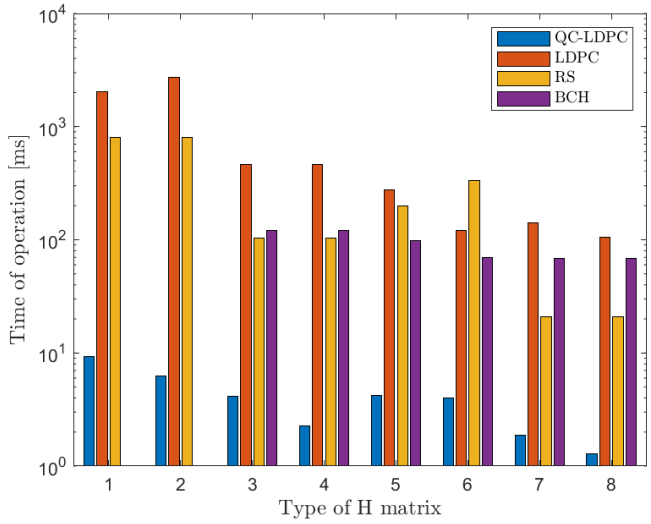


Figure 7. Single block encoding time periods.

Basing on the timing measurements, we estimated that energy saving that can be achieved for the proposed encoder in comparison with the Reed–Solomon encoding. For this estimation, the following assumptions were made, reflecting real operation of the microcontroller STM32 used for experiments:

- The voltage supplied to the circuit is 3.6 V.
- The current in the active mode with 80 MHz clock is 33.5 mA.

We calculated the normalized energy needed for encoding a single bit of the QC-LDPC codes as well as the other reference encoders. The results are presented in Figure 8. Then, we subtracted the results, obtaining the potential saving that can be achieved, expressed in [µJ/bit]. These results are presented in Table 3.

Table 3. Energy savings per bit, achieved for QC-LDPC encoding, in comparison with counterpart Reed–Solomon encoding.

Matrix ID	Energy Saved [µJ/bit]	Matrix ID	Energy Saved [µJ/bit]
H1	196.81	H2	197.55
H3	48.65	H4	49.59
H5	76.29	H6	108.94
H7	16.75	H8	17.26

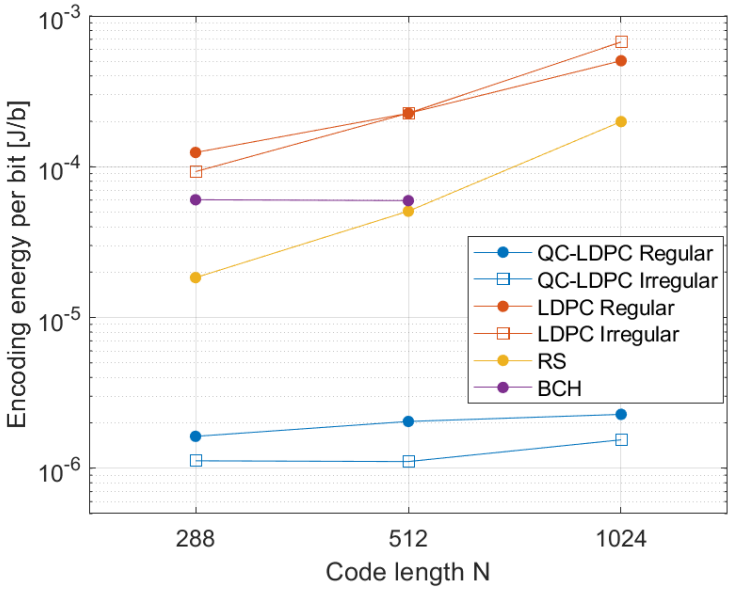


Figure 8. Encoding energy needed per bit vs. code length.

6.3. Throughput

The trade-offs between code parameters and the achieved encoder features can be also illustrated by means of the achieved maximum encoding throughput. This is visualized in Figure 9. We can observe the advantage of the proposed efficient QC-LDPC encoder realization over the basic LDPC, BCH and RS encodings. The throughput developed realization is rather not sensitive to the block length. The throughput in general increases with the code rate, because the portion of information bits in the code blocks increases.

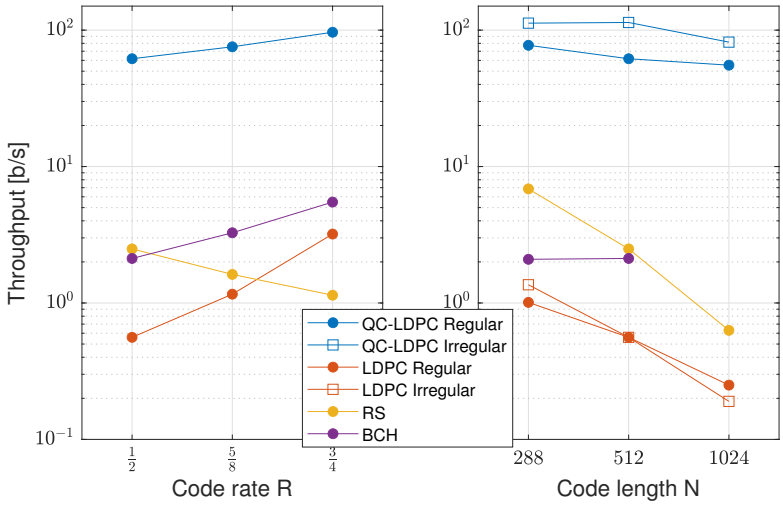


Figure 9. Throughput of the encoders vs. code parameters.

6.4. Error-Correction Performance

In order to show the coding gain that can be achieved with the QC-LDPC codes employed in comparison with typical classic block codes, we performed Monte Carlo simulations for the QC-LDPC codes, as well as for their RS and BCH counterparts. RS codes with rate $R = 0.5$, BCH and QC-LDPC codes with different block lengths were used for this comparison. A QPSK modulator and an AWGN channel model were used in the communication system models. The LDPC codes were decoded with a belief propagation (BP) decoder, the BCH codes were decoded with the hard-decision Berlekamp algorithm, and the RS codes were decoded using the Berlekamp–Massey decoding algorithm.

Figure 10 presents the results of these experiments in the form of the bit error rate curve versus the effective signal-to-noise ratio (E_b/N_0) in an AWGN channel for three selected cases: (1024, 512), (512, 256), and (512, 320) LDPC codes and their counterparts, as shown in Table 2. An additional coding gain of about 2 dB for the QC-LDPC codes is visible in all of the presented cases. This result represents the amount of transmission power that can be saved when using iteratively decoded QC-LDPC codes instead of standard block codes.

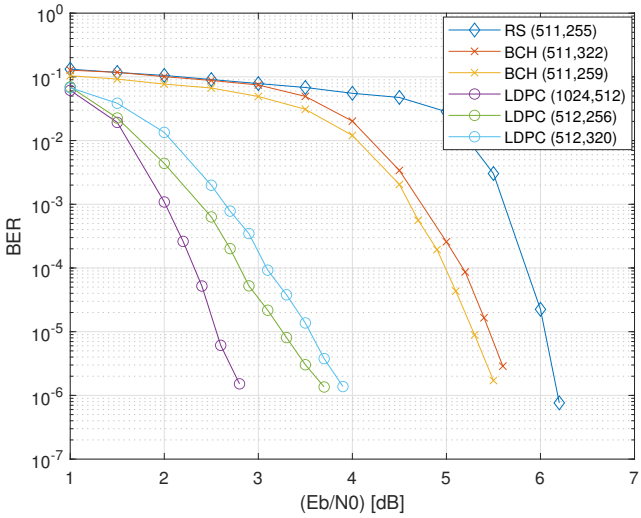


Figure 10. Error correction performance of the experimental codes.

7. Discussion

The ECC-encoding procedure presented in this paper is devoted to error protection in IoT node devices constructed with microcontroller-type equipment and without strict throughput requirements, but with the energy efficiency being an important factor. We showed that QC-LDPC codes, which are usually considered for high-throughput communication systems and advanced protocols can also be considered for these types of computationally constrained devices. We provided Richardson–Urbanke LDPC encoding expressed for QC-LDPC codes, and we showed how the encoding can be decomposed into elementary operations that are suitable for implementation in a simple microcontroller device.

The whole encoding algorithm was implemented and verified for a few QC-LDPC parity-check matrices. Comparisons with other known block codes (RS and BCH) are also provided, showing that the memory requirements are not greater than for standard block codes, while the achievable throughput is increased and the encoding time is reduced, which enables the energy consumption reduction. At the same time, the error-correction performance of LDPC codes is significantly better than for standard block codes. This provides the potential for the reduction in the energy consumption of the communication links used to connect IoT nodes with IoT gateways.

Author Contributions: Conceptualization, J.H. and W.S.; methodology, J.H. and W.S.; software, J.H.; validation, J.H., W.S., W.I., L.D. and W.F.; experimental investigation, J.H., W.S., W.I., L.D. and W.F.; data acquisition, J.H., W.S., W.I., L.D. and W.F.; writing, J.H., W.S., W.I., L.D. and W.F.; supervision, W.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was co-financed by the Ministry of Education and Science of Poland under grant No. DWD/3/7/2019 as well as SUBB.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Swamy, S.N.; Kota, S.R. An Empirical Study on System Level Aspects of Internet of Things (IoT). *IEEE Access* **2020**, *8*, 188082–188134. [CrossRef]
2. Dubal, V.A.; Rao, Y.S. A Low-Power High-Performance Sensor Node for Internet of Things. In Proceedings of the IEEE Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 14–15 June 2018; pp. 607–612.
3. Rajasekaran, C.; Raguvaran, K. Microcontroller Based Reconfigurable IoT Node. In Proceedings of the IEEE 4th International Conference on Frontiers of Signal Processing, Poitiers, France, 24–27 September 2018; pp. 12–16.
4. Berrou, C.; Glavieux, A.; Thitimajshima, P. Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes. In Proceedings of the (IEEE) International Conference on Communications, Geneva, Switzerland, 23–26 May 1993; pp. 1064–1070.
5. Arikan, E. Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels. *IEEE Trans. Inf. Theory* **2009**, *55*, 3051–3073. [CrossRef]
6. Gallager, R. Low-density parity-check codes. *IRE Trans. Inf. Theory* **1962**, *8*, 21–28. [CrossRef]
7. MacKay, D.J.C. Good Error-Correcting Codes Based on Very Sparse Matrices. *IEEE Trans. Inf. Theory* **1999**, *45*, 399–431. [CrossRef]
8. Condo, C. VLSI Implementation of a Multi-ModeTurbo/LDPC Decoder Architecture. *IEEE Trans. Circuits Syst. I* **2013**, *60*, 1441–1454. [CrossRef]
9. Nguyen, T.T.B.; Tan, T.N.; Lee, H. Low-Complexity High-Throughput QC-LDPC Decoder for 5G New Radio Wireless Communication. *Electronics* **2021**, *10*, 1–18.
10. Zhu, Y.; Xing, Z.; Li, Z.; Zhang, Y.; Hu, Y. High Area-Efficient Parallel Encoder with Compatible Architecture for 5G LDPC Codes. *Symmetry* **2021**, *13*, 700 . [CrossRef]
11. Hailes, P.; Xu, L.; Maunder, R.G.; Al-Hashimi, B.M.; Hanzo, L. A Survey of FPGA-Based LDPC Decoders. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 1098–1122. [CrossRef]
12. Luo, H.; Zhang, Y.; Li, W.; Huang, L.K.; Cosmas, J.; Li, D.; Maple, C. Low Latency Parallel Turbo Decoding Implementation for Future Terrestrial Broadcasting Systems. *IEEE Trans. Broadcast.* **2018**, *64*, 96–104. [CrossRef]
13. Lazarenko, A. FPGA Design and Implementation of DVB-S2/S2X LDPC Encoder. In Proceedings of the IEEE International Conference on Electrical Engineering and Photonics, St. Petersburg, Russia, 17–18 October 2019; pp. 98–102.
14. Petrović, V.L.; El Mezeni, D.M.; Radošević, A. Flexible 5G New Radio LDPC Encoder Optimized for High Hardware Usage Efficiency. *Electronics* **2021**, *10*, 1106 . [CrossRef]

15. Tarver, C.; Tonnemacher, M.; Chen, H.; Zhang, J.; Cavallaro, J.R. GPU-Based, LDPC Decoding for 5G and Beyond. *IEEE Open J. Circuits Syst.* **2021**, *2*, 278–290. [CrossRef]
16. Liao, S.; Zhan, Y.; Shi, Z.; Yang, L. A High Throughput and Flexible Rate 5G NR LDPC Encoder on a Single GPU. In Proceedings of the IEEE International Conference on Advanced Communications Technology (ICACT), Pyeongchang-gun, Korea, 7–10 February 2021; pp. 29–34.
17. Richardson, T.J.; Shokrollahi, M.A.; Urbanke, R.L. Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes. *IEEE Trans. Inf. Theory* **2001**, *47*, 619–637. [CrossRef]
18. Kelley, C.A.; Deepak, S. Pseudocodewords of Tanner Graphs. *IEEE Trans. Inf. Theory* **2007**, *53*, 4013–4038. [CrossRef]
19. Divsalar, D.; Dolinar, S.; JoJones, C.R.; Andrews, K. Capacity-Approaching Protograph Codes. *IEEE J. Sel. Areas Commun.* **2009**, *27*, 876–888. [CrossRef]
20. Hu, X.Y.; Eleftheriou, E.; Arnold, D.M. Regular and Irregular Progressive Edge-Growth Tanner Graphs. *IEEE Trans. Inf. Theory* **2005**, *51*, 386–398. [CrossRef]
21. Gholami, M.; Raeisi, G. Large Girth Column-Weight Two and Three LDPC Codes. *IEEE Commun. Lett.* **2014**, *18*, 1671–1674. [CrossRef]
22. Bocharova, I.E.; Kudryashov, B.; Johannesson, R. Searching for Binary and Nonbinary Block and Convolutional LDPC Codes. *IEEE Trans. Inf. Theory* **2016**, *62*, 163–183. [CrossRef]
23. Tasdighi, A.; Banihashemi, A.H.; Sadeghi, M.R. Symmetrical Constructions for Regular Girth-8 QC-LDPC Codes. *IEEE Trans. Commun.* **2017**, *52*, 1242–1247. [CrossRef]
24. Sulek, W. Protograph Based Low-Density Parity-Check Codes Design with Mixed Integer Linear Programming. *IEEE Access* **2019**, *7*, 1424–1438. [CrossRef]
25. Sulek, W.; Kucharczyk, M. Partial parallel encoding and algorithmic construction of non-binary structured IRA codes. *China Commun.* **2016**, *13*, 103–116. [CrossRef]
26. Fossorier, M.P.C.; Mihaljevic, M.; Imai, H. Reduced Complexity Iterative Decoding of Low-Density Parity Check Codes Based on Belief Propagation. *IEEE Trans. Commun.* **1999**, *47*, 673–680. [CrossRef]
27. Li, J.; Liu, K.; Lin, S.; Abdel-Ghaffar, K. Algebraic Quasi-Cyclic LDPC Codes: Construction, Low Error-Floor, Large Girth and a Reduced-Complexity Decoding Scheme. *IEEE Trans. Commun.* **2014**, *62*, 2626–2637. [CrossRef]
28. Stark, M.; Lewandowsky, J.; Bauch, G. Information-Bottleneck Decoding of High-Rate Irregular LDPC Codes for Optical Communication Using Message Alignment. *Appl. Sci.* **2018**, *10*, 1–17. [CrossRef]
29. Marques da Silva, M.; Dinis, R.; Martins, G. On the Performance of LDPC-Coded Massive MIMO Schemes with Power-Ordered NOMA Techniques. *Appl. Sci.* **2021**, *11*, 8684 . [CrossRef]
30. Lin, C.F. UFMC-Based Underwater Voice Transmission Scheme with LDPC Codes. *Appl. Sci.* **2021**, *11*, 1818. [CrossRef]
31. Sulek, W. Pipeline processing in low-density parity-check codes hardware decoder. *Bull. Pol. Acad. Sci. Tech. Sci.* **2011**, *59*, 149–155.
32. Mahdi, A.; Paliouras, V. A Low Complexity-High Throughput QC-LDPC Encoder. *IEEE Trans. Signal Process.* **2014**, *62*, 2696–2708. [CrossRef]
33. Huang, J.; Zhou, S.; Willett, P. Nonbinary LDPC Coding for Multicarrier Underwater Acoustic Communication. *IEEE J. Sel. Areas Commun.* **2008**, *26*, 1684–1696. [CrossRef]
34. Boutillon, E.; Conde-Canecia, L.; Ghouwayel, A.A. Design of a GF(64)-LDPC Decoder Based on the EMS Algorithm. *IEEE Trans. Circuits Syst. I* **2013**, *60*, 2644–2656. [CrossRef]
35. Sulek, W. Non-binary LDPC Decoders Design for Maximizing Throughput of an FPGA Implementation. *Circuits Syst. Signal Process.* **2016**, *35*, 4060–4080. [CrossRef]
36. Zhao, S.; Ma, X. Construction of High-Performance Array-Based Non-Binary LDPC Codes With Moderate Rates. *IEEE Commun. Lett.* **2016**, *20*, 13–16. [CrossRef]
37. *IEEE Standard: IEEE P802.16*. Local and Metropolitan Area Networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems. IEEE: Piscataway, NJ, USA, 2006.
38. *IEEE 802.11-2016*. IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE: Piscataway, NJ, USA, 2016.
39. 3GPP TS 38.212 Version 16.2.0 Release 16. *Multiplexing and Channel Coding; 5G; NR*; ETSI: Sophia Antipoli, France, 2020.
40. Theodoropoulos, D.; Kranitis, N.; Paschalis, A. An efficient LDPC encoder architecture for space applications. In Proceedings of the IEEE 22nd International Symposium on On-Line Testing and Robust System Design (IOLTS), Sant Feliu de Guixols, Spain, 4–6 July 2016; pp. 149–154.
41. Nguyen, T.T.B.; Nguyen Tan, T.; Lee, H. Efficient QC-LDPC Encoder for 5G New Radio. *Electronics* **2019**, *8*, 668. [CrossRef]
42. Wu, H.; Wang, H. A High Throughput Implementation of QC-LDPC Codes for 5G NR. *IEEE Access* **2019**, *7*, 185373–185384. [CrossRef]
43. Richardson, T.J.; Urbanke, R.L. Efficient Encoding of Low-Density Parity-Check Codes. *IEEE Trans. Inf. Theory* **2001**, *47*, 638–656. [CrossRef]
44. Fossorier, M.P.C. Quasi-Cyclic Low-Density Parity-Check Codes from Circulant Permutation Matrices. *IEEE Trans. Inf. Theory* **2004**, *50*, 1788–1793. [CrossRef]

45. Myung, S.; Yang, K.; Kim, J. Quasi-Cyclic LDPC Codes for Fast Encoding. *IEEE Trans. Inf. Theory* **2005**, *51*, 2894–2901. [CrossRef]
46. Tanner, R.M. A Recursive Approach to Low Complexity Codes. *IEEE Trans. Inf. Theory* **1981**, *IT-27*, 533–547. [CrossRef]
47. Tasdighi, A.; Banihashemi, A.H.; Sadeghi, M.R. Efficient Search of Girth-Optimal QC-LDPC Codes. *IEEE Trans. Inf. Theory* **2016**, *62*, 1552–1564. [CrossRef]
48. Xu, H.; Feng, D.; Luo, R.; Bai, B. Construction of Quasi-Cyclic LDPC Codes via Masking With Successive Cycle Elimination. *IEEE Commun. Lett.* **2016**, *20*, 2370–2373. [CrossRef]
49. Lin, S.; Costello, D.J., Jr. *Error Control Coding: Fundamentals and Applications*, 2nd ed.; Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 2004.
50. RSCODE Project. Available online: http://rscode.sourceforge.net (accessed on 3 October 2021).
51. Generic Binary BCH Encoding/Decoding Library. Available online: https://github.com/Parrot-Developers/bch. (accessed on 14 January 2022).

5.3 Publikacja 3: Dekoder LDPC implementowany w mikrokontrolerze dla systemów Internetu rzeczy.

Jakub HYLA[1 2], Wojciech SUŁEK[1]

Silesian University of Technology (1), TKH Technology Poland Sp. z.o.o. (2)

doi:10.15199/48.2023.04.23

Dekoder LDPC implementowany w mikrokontrolerze dla systemów Internetu Rzeczy

Streszczenie. *Artykuł dotyczy projektowania systemów kodowania korekcyjnego dla protokołów komunikacyjnych w Internecie Rzeczy, które są implementowane na platformach o mocno ograniczonych zasobach obliczeniowych. W artykule zaproponowano wykorzystanie nowoczesnych kodów LDPC (Low Density Parity Check), zaprezentowano algorytm dekodujący oraz przedstawiono wyniki eksperymentalne implementacji w układzie mikrokontrolera. Przeprowadzono testy dla różnych wielkości słów kodowych oraz zebrano wyniki związane z czasem dekodowania, przepustowością, jak również liczbą iteracji potrzebną do zdekodowania jednego bloku.*

Abstract. *The article concerns the design of correction coding systems for communication protocols in the Internet of Things, which are implemented on platforms with very limited computing resources. The article proposes the use of modern LDPC (Low Density Parity Check) codes, presents the decoding algorithm and presents the experimental results of the implementation in a microcontroller device. Experiments were performed for different codeword sizes and the results were collected concerning the decoding time, throughput as well as the number of iterations needed to decode one block.* ***(LDPC decoder for Internet of Things systems)***

Słowa kluczowe: IoT, LDPC, QC-LDPC, dekodowanie
Keywords: IoT, LDPC, QC-LDPC, decoding

Wprowadzenie

Współczesny rozwój technologiczny determinuje wzrost znaczenia poprawnego przesyłania danych w wielu rozmaitych systemach teleinformatycznych. Wymagana jest transmisja coraz większej liczby informacji, w coraz krótszym czasie, przy wykorzystaniu jak najmniejszych porcji energii, co skutkuje nieuniknionym zjawiskiem błędów w transmisji. Tymczasem poprawność danych przesyłanych jest jednym z kluczowych elementów w wielu sektorach. Dzięki kodowaniu kanałowemu możliwe jest wykrywanie błędów, a także automatyczna rekonstrukcja prawidłowych bloków danych (korekcja). Jest to realizowane poprzez dołączanie do bitowych bloków informacyjnych pewnej liczby symboli (bitów) nadmiarowych (kontrolnych), które mogą być wykorzystane w odbiorniku do detekcji i korekcji błędów.

Kody LDPC (Low Density Parity Check) są obecnie jednymi z najlepszych metod kodowania blokowego o bardzo dobrych własnościach korekcyjnych. Kody LDPC dzięki swoim własnościom znajdują się blisko granicy Shannona [11]. Historia kodów LDPC sięga roku 1962, gdzie zostały zaproponowane przez R. G. Gallagera [5]. W ówczesnych czasach poziom technologiczny nie pozwalał na ich zastosowanie. Tym samym straciły one na znaczeniu aż do roku 1999, gdzie D.J.C. MacKay zwrócił uwagę na znakomite własności kodów LDPC [10]. Od tego czasu zdecydowanie zyskały one na popularności, czego przykładami są zastosowania takie jak: DVB-S2 i DVB-T2 [4], WiFi [1], WiMAX [7], sieci 5G [13, 2, 12, 15].

Projekt badawczy, którego częścią są wyniki przedstawione w niniejszym artykule, ma na celu pokazanie, że kody LDPC mogą być z powodzeniem stosowane także w systemach o silnie ograniczonych zasobach sprzętowych, którymi zwykle cechują się urządzenia systemów Internetu rzeczy (ang. IoT). W ramach projektu powstała efektywna implementacja kodera LDPC w układzie mikrokontrolera, która została opisana w [6], a także implementacja dekodera, oparta na koncepcji opisanej w niniejszym artykule. Artykuł dotyczący kodera opisuje problem złożoności obliczeniowej kodera LDPC oraz porusza problem związany z przetwarzaniem dużych liczby danych przez urządzenia o niewielkich możliwościach obliczeniowych. Jednocześnie autorzy proponują zmianę podejścia do zapisu tak zwanych macierzy rzadkich, czyli takich, w których występuje niewiele elementów niezerowych na poczet wykorzystania innych metod zapisu tych samych informacji. Analiza i rozwiązanie tego problemu znalazło zastosowanie w urządzeniu typu Internetu Rzeczy bazującym na mikrokontrolerze o ograniczonych możliwościach obliczeniowych. Wyniki liczbowe zajętości pamięci, czasu potrzebnego na zakodowanie bloku informacyjnego, jak również przepustowość zostały podsumowane w wynikach cytowanego artykułu. Należy zwrócić uwagę, że zyski związane z efektywną implementacją algorytmu kodowania są według autorów bardzo dobre i wpisują się w trend urządzeń typu Internetu Rzeczy zasilanych bateryjne ze względu na zyski związana z energooszczędnością.

Kody LDPC - podstawowe definicje

W systemie systematycznego kodowania blokowego, określony jest wektor informacyjny $\mathbf{u} = [u_1, u_2, ...u_K]$ będący częścią wektora kodowego $\mathbf{c} = [c_1, c_2, ...c_N]$, gdzie $N = K + M$, a M oznacza liczbę bitów nadmiarowych. Taki kod blokowy jest oznaczany $C(N, K)$, a kolejne słowa strumienia transmitowanej informacji – wektory informacyjne $\mathbf{u}$ są przekształcane w zakodowane wektory $\mathbf{c}$. Elementy wektorów $\mathbf{u}$, $\mathbf{c}$ mogą w ogólności należeć do ciała Galois $GF(q)$, jednakże powszechnie jest stosowane kodowanie binarne nad $GF(2)$, gdzie $u_k, c_n \in \{0, 1\}$.

Pojęcie sprawności kodu $R = K/N$ definiuje poziom nadmiarowości kodowania. Sprawność zawsze zawiera się w przedziale $0 < R < 1$. Zwiększenie liczby elementów nadmiarowych poprawia własności korekcyjne kodu, jednocześnie pogarszając sprawność kodu.

Dany kod LDPC definiowany jest przez macierz kontroli parzystości $\mathbf{H}$, która jest macierzą rzadką, natomiast $\mathbf{Hc}^T = 0$ to równanie kontrolne, które pozwala na weryfikację czy dany wektor $\mathbf{c}$ jest prawidłowy. Jeśli nie, to iteracyjny algorytm dekodowania LDPC pozwala na korekcję (pewnej liczby) błędów.

W przypadku kodów LDPC macierz kontrolna $\mathbf{H}$ jest macierzą rzadką – o małej liczbie elementów niezerowych. Kody LDPC dzielą się na dwa typy – kody regularne i nieregularne. Odróżnia je stała bądź też zmienna w przypadku kodów nieregularnych liczba elementów niezerowych w każdej kolumnie $\mathbf{H}$. W przypadku kodów regularnych jako zaletę można wskazać uproszczenie struktury sprzętowego kodera, natomiast w przypadku kodów nieregularnych o odpowiednio dobranej macierzy $\mathbf{H}$, zaletą w ogólności są lepsze własności korekcyjne od kodów regularnych [14]. W większości standardów telekomunikacyjncych wykorzysty-

wane są kody nieregularne.

Algorytmy dekodowania wykorzystywane w praktyce są algorytmami iteracyjnymi. Alternatywnym sposobem reprezentacji kodu dla macierzy $\mathbf{H}$, na potrzeby prezentacji algorytmu dekodowania, jest graf Tannera [10]. Graf Tannera dla liniowego kodu blokowego o macierzy kontrolnej $\mathbf{H}$ jest grafem dwudzielnym. Składa się on z wierzchołków kontrolnych i bitowych. W procesie dekodowania elementarne obliczenia reprezentowane są przez wierzchołki, natomiast sposób przesyłania wyników tych obliczeń reprezentowany jest poprzez krawędzie. Wierzchołki kontrolne odpowiadają wierszom macierzy $\mathbf{H}$ – czyli równaniom kontrolnym. Wierzchołki bitowe odpowiadają kolumnom macierzy $\mathbf{H}$ – czyli bitom słowa kodowego. Wartości prawdopodobieństw są reprezentowane przez wiadomości – poziomy wiarygodności (ang. beliefs), że dany bit jest równy 0 lub 1 (wiadomości z wierzchołków bitowych do kontrolnych) oraz informacje o spełnieniu równania kontrolnego, przy określonej wartości bitu (wiadomości z wierzchołków kontrolnych do bitowych).

Algorytmy iteracyjne polegają na zbliżaniu się do prawidłowego rozwiązania z każdą iteracją dekodowania. Inaczej nazywane są algorytmami propagacji poziomów wiarygodności BP (ang. Belief Propagation) i szczegółowo opisane w licznej literaturze [8], [10].

Model systemu IoT

Obecnie urządzenia systemów tzw. Internetu rzeczy są coraz bardziej popularne i zazwyczaj są postaci małego urządzenia zasilanego bateryjnie, z bardzo ograniczonymi funkcjami, lecz czasem także znacznie bardziej rozbudowanego urządzenia posiadającego system operacyjny, dającymi możliwość bezpośredniej wymiany danych z chmurą [9]. W niniejszym artykule uwagę skupiamy a urządzeniach bazujących na prostych mikrokontrolerach (np. STM32L4), o ograniczonych zasobach obliczeniowych. Dla wspomnianego mikrokontrolera producent (STM) daje do wykorzystania narzędzie wspomagające programowanie, które wykorzystano w opisanych pracach badawczych oraz biblioteki do obsługi peryferiów.

Model przykładowego rozwiązania komunikacji w systemie IoT jest przedstawiony na Rys. 1. Urządzenie końcowe Internetu Rzeczy wysyła / odbiera komunikaty, np. w aplikacji chmurowej, jako węzeł pośredni wykorzystując urządzenie centralne – bramkę IoT. Poziom oznaczony na czerwono, związany z urządzeniem IoT, jest poziomem o ograniczonych zdolnościach obliczeniowych, reprezentującym zwykle niskokosztowe i energooszczędne urządzenia końcowe.

Urządzenie IoT wysyła komunikaty w kierunku tzw. w górę (ang. uplink) oraz odbiera w kierunku tzw. w dół (ang. downlink). W przypadku komunikacji uplink urządzenie IoT gromadzi dane, przetwarza je, koduje oraz zamienia na odpowiedni sygnał dostosowany do kanału komunikacyjnego. Dane kodowane są za pomocą kodera LDPC, które zostało zaprojektowane dla urządzeń typu IoT, wyposażonych w mikrokontroler [6]. Komunikacja w kierunku downlink realizowana jest od bramki do urządzenia IoT, więc urządzenie odbiera sygnał, przetwarza go do postaci bitów lub tzw. miękkich decyzji (demodulator), a następnie wykorzystuje programowy dekoder do korekcji błędów.

Realizacja algorytmów kodeka LDPC w urządzeniach IoT

Algorytmy kodowania i dekodowania mają wymagania co do pojemności pamięci, jak również mocy obliczeniowej. Algorytm kodowania zrealizowany dla mikrokontrolera w urządzeniu typu IoT został opisany w artykule [6]. Na potrzeby urządzeń o ograniczonych zasobach pamięci i mocy obliczeniowej został zaproponowany algorytm kodujący oparty o indeksy niezerowych elementów macierzy $\mathbf{H}$. Możliwe jest również zrealizowanie algorytmu kodowania w postaci programowej wymagającego mniejszej liczby pamięci mikrokontrolera. Taki algorytm może realizować dekodowanie na potrzeby kodów LDPC jak i QC-LDPC (ang. Quasi Cyclic LDPC), jak przedstawiono w [6]

Kolejnym krokiem jest opracowanie realizacji programowej dekodera kodów LDPC dla układu mikrokontrolera. Do implementacji wybrano wersję algorytmu BP (ang Be-

Algorithm 1: Dekodowanie LDPC [16] dla układu mikrokontrolera.

Input:
– Macierz kontroli parzystości $\mathbf{H}$ zapisana w postaci indeksów niezerowych elementów.
– Maksymalna liczba iteracji dekodowania.
Output:
– Informacja o powodzeniu lub niepowodzeniu dekodowania.
– Wektor zdekodowanych danych.

1 Inicjalizacja - Przypisanie wartości wejściowych. Są one prawdopodobieństwami LLR (ang. Log-Likelyhood Ratio) (dla wszystkich $m \in M(n)$ i $n \in (1, M)$)

$$Q_{nm} := L(c_n|y_n) = \ln\left(\frac{P(c_n = 0|y_n)}{P(c_n = 1|y_n)}\right) \quad (1)$$

2 Krok horyzontalny, w którym wyznaczane są wierzchołki kontrolne. Wyznaczanie wartości minimalnych (dla wszystkich $n \in N(m)$ i $m \in (1, N)$):

$$R_{mn} := \left[\prod_{k \in N(m)\setminus n} \operatorname{sgn}(Q_{km})\right] \min_{k \in N(m)\setminus n} |Q_{km}| \quad (2)$$

Algorytm dekodujący wykorzystany w implementacji to algorytm Normalized Min-Sum

3 Krok wertykalny, w którym wyznaczane są wierzchołki bitowe. Sumowanie wartości minimalnych i wejściowych LLR (dla wszystkich $m \in M(n)$ i $n \in (1, M)$):

$$Q_{nm} := L(c_n|y_n) + \sum_{k \in M(n)\setminus m} R_{kn} \quad (3)$$

4 Dekodowanie wstępne Wyznaczanie prawdopodobieństw pseudo-posteriori (dla wszystkich n):

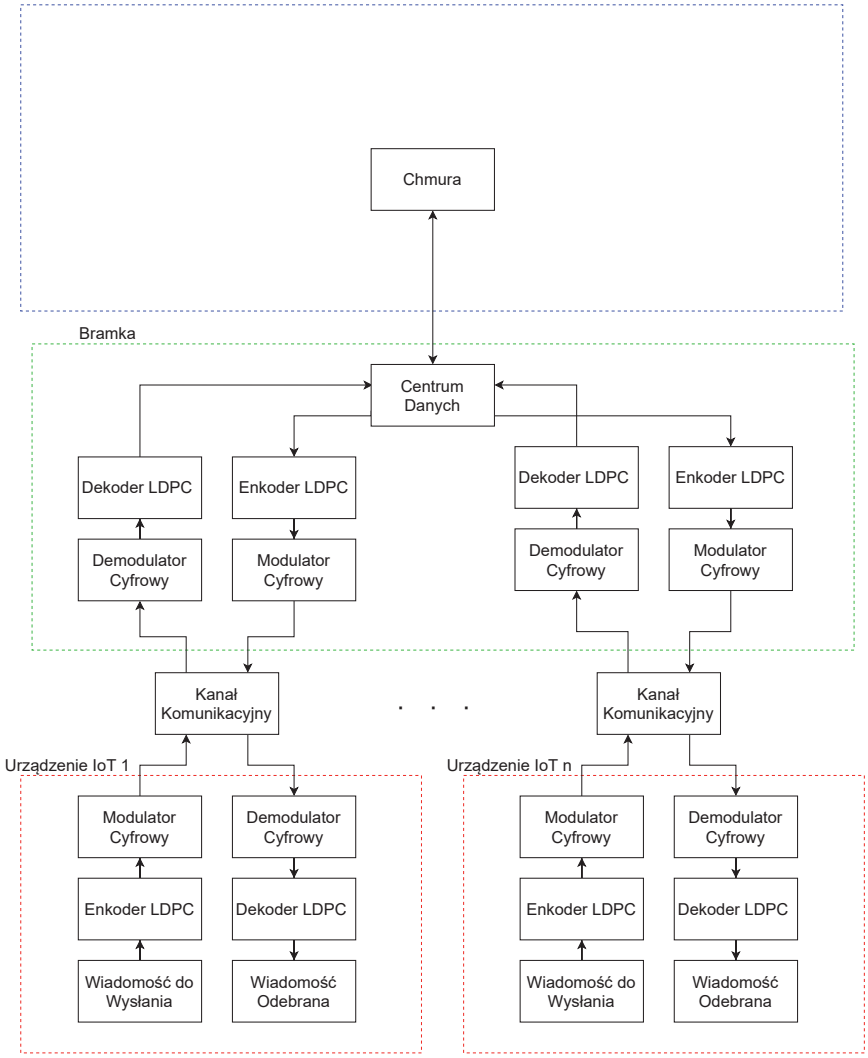
$$Q_n := L(c_n|y_n) + \sum_{k \in M(n)} R_{kn} \quad (4)$$

5 Wyznaczenie twardych decyzji.Podejmowanie próbnych, twardych decyzji (dla wszystkich $n \in (1, M)$):

$$\hat{c}_n := \begin{cases} 1 & \text{gdy } Q_n < 0 \\ 0 & \text{gdy } Q_n \geq 0 \end{cases} \quad (5)$$

6 Warunek zakończenia

$$\mathbf{H}\hat{c}^T = 0 \quad (6)$$



Rys. 1. Uproszczony model komunikacji z wykorzystaniem kodowania korekcyjnego LDPC w systemie IoT o topologii gwiazdy

lief Propagation) znaną jako Normalized Min-Sum [3], która jest przedstawiona jako algorytm 1. Przedstawione kroki powtarzane są iteracyjnie tak długo, aż zostanie osiągnięty warunek zakończenia (6) lub zostanie osiągnięta maksymalna liczba iteracji (co oznacza dekodowanie zakończone niepowodzeniem). Schemat blokowy algorytmu pokazany jest na Rys. 2. Należy ona do klady iteracyjnych algorytmów propagacji wartości wiarygodności BP.

W opisie algorytmu 1 wykorzystano następujące oznaczenia:

- $\mathbf{x} = [c_1, c_2, \ldots, c_n]$ jest wektorem kodowym,
- $\mathbf{y} = [y_1, y_2, \ldots, y_n]$ jest wektorem wartości odebranych (miękkie decyzje),
- $L(c_n|y_n)$ - LLR prawdopodobieństw *a priori* dla bitu n-tego,
- $N(m)$ - numery kolumn w macierzy kontrolnej $\mathbf{H}$ zawierających jedynkę w wierszu m-tym,
- $M(n)$ - numery wierszy w macierzy kontrolnej $\mathbf{H}$ zawierających jedynkę w kolumnie n-tej,
- Q_{nm} - wartość LLR (*Log-Likelihood Ratio*) wiarygodności z n-tego wierzchołka bitowego do m-tego wierzchołka kontrolnego grafu Tannera,
- R_{mn} - wiadomość z m-tego wierzchołka kontrolnego do n-tego wierzchołka bitowego,
- $\hat{c}_n$ - wektor zdekodowany.

Implementacja programowa algorytmu dokowania na urządzeniu IoT

Zaproponowany algorytm został zaimplementowany z wykorzystaniem języka C, przy czym opis utworzono optymalizując dla kompilacji do układu mikrokontrolera. Znajduje on zastosowanie w dekodowaniu danych dla macierzy kontroli parzystości $\mathbf{H}$ regularnych oraz nieregularnych kodów LDPC, w tym klasy QC-LDPC. Jest to zatem uniwersalne rozwiązanie, które wymaga jedynie odpowiedniego zapisu macierzy $\mathbf{H}$, z wykorzystaniem indeksów elementów niezerowych tej macierzy.

Pierwszym krokiem algorytmu dekodowania jest przekazanie danych do procedury inicjalizującej zmienne oraz bufory do przechowywania danych tymczasowych. Przy wywołaniu funkcji dekodowania trafią one na stos w obszarze pamięci RAM. Pamięć RAM mikrokontrolera zwykle ograniczona jest do kilkudziesięciu kilobajtów w zależności od typu mikrokotrolera. W tym kroku przypisywane są wartości wejściowe (LLR prawdopodobieństw skojarzonych z poszczególnymi bitami), dla wszystkich węzłów bitowych. Na tym etapie wywoływana jest również funkcja odpowiadająca za podjęcie tzw. twardych decyzji, czyli określenie na podstawie odebranych symboli najbardziej prawdopodobnych wartości poszczególnych bitów (0 / 1). Tworzona i inicjalizowana jest macierz (dwuwymiarowa

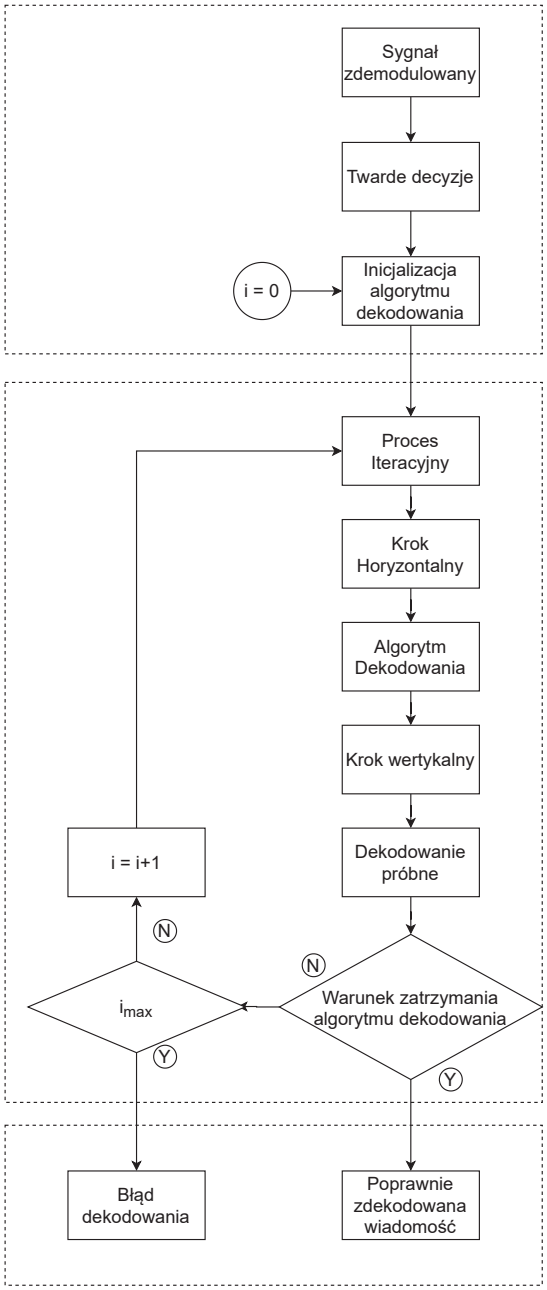
Macierz	K	N	P	R	Rodzaj
H1	512	1024	32	$\frac{1}{2}$	Regularna
H2	512	1024	32	$\frac{1}{2}$	Nieregularna
H3	256	512	32	$\frac{1}{2}$	Regularna
H4	256	512	32	$\frac{1}{2}$	Nieregularna
H5	320	512	32	$\frac{5}{8}$	Regularna
H6	384	512	32	$\frac{3}{4}$	Regularna
H7	144	288	16	$\frac{1}{2}$	Regularna
H8	144	288	16	$\frac{1}{2}$	Nieregularna

Tablica 1. Parametry eksperymentalnych macierzy kontroli parzystości $\mathbf{H}$ dla kodowania LDPC.

tablica), która przechowuje wiadomości z wierzchołków kontrolnych do bitowych (macierz $\mathbf{rllr}$ oraz macierz (tablica) przechowująca wiadomości z wierzchołków bitowych do kontrolnych (macierz $\mathbf{qllr}$).

Następnie rozpoczynana jest pętla o ograniczonej liczbie iteracji, w której realizowany jest algorytm dekodowania. Ostatnim krokiem pętli jest zawsze sprawdzanie warunku zakończenia (spełnienia wszystkich równań kontrolnych) i w zależności od rezultatu możliwe jest przerwanie obliczeń.

Drugim krokiem jest krok horyzontalny, w którym obliczane są wartości wierzchołków kontrolnych oraz wykonywana jest realizacja algorytmu Normalized Min-Sum za pomocą którego dane są dekodowane. Dla każdego wiersza macierzy kontrolnej zapisywane są w pamięci indeksy elementów niezerowych macierzy kontrolnej $\mathbf{H}$. Patrząc na graf Tannera zapisywane są w pamięci indeksy wierzchołków bitowych połączonych krawędziami z wierzchołkami kontrolnymi. Następnie zapisywane w pamięci są wartości bezwzględne twardych decyzji oraz ich znak w osobnych tablicach pomocniczych. Algorytm iteruje po wszystkich wierszach macierzy (H), gdzie dla każdego wiersza obliczana jest wartość minimalna z elementów o indeksach innych niż aktualny, natomiast bazując na znakach obliczany jest iloczyn znaków wektora wejściowego bez brania pod uwagę elementu o obliczanym indeksie lub innymi słowy jest to XOR bitów znaku. Obliczona wartość minimalna jest normalizowana poprzez przemnożenie ich przez wartość w zakresie 0.5-1. Na potrzeby demonstracyjne przyjęto doświadczalnie wartość 0.75. Po normalizacji wartości zaokrąglane są do liczb całkowitych. W trakcie pracy nad algorytmem zauważono, że wartość parametru mniejsza od 1 wpływa pozytywnie, gdy jest dużo błędów i algorytm działa na granicy swoich możliwości. W analogicznym przypadku algorytm dekodujący Min-Sum z parametrem 1 nie potrafił zdekodować danych, natomiast algorytm dekodujący Normalized Min-Sum z parametrem 0.75 dekodował je poprawnie. Końcowym etapem jest połączenie obliczonego znaku z wartością modułem. Tym samym wraz kolejnymi iteracjami aktualizowane są wartości macierzy $\mathbf{rllr}$.



Rys. 2. Parameters of the experimental parity-check matrices $\mathbf{H}$ for LDPC. The λ_d represents the fraction of degree-d variable nodes in the code graph.

Trzecim krokiem jest krok wertykalny, w którym obliczane są wierzchołki bitowe. Wykonywana jest ona zgodnie z Tab. 1 N razy, czyli tyle ile mamy kolumn macierzy $\mathbf{H}$. Początkowo wybieramy indeksy niezerowych elementów w pierwszej kolumnie macierzy $\mathbf{H}$. Bazując na macierzy $\mathbf{rllr}$ obliczanej w poprzednim kroku wyciągany jest jej odpowiedni fragment bazując na indeksach niezerowych elementów kolumn macierzy $\mathbf{H}$. Następnie wybrany fragment jest sumowany bez brania pod uwagę elementu o obliczanym indeksie, do niego dodawana jest wartość inicjująca algorytm. Obliczony wektor tymczasowy jest ograniczany do co

wartości po to, aby wartości nie dążyły do +/-∞, a w przypadku obliczeń komputerowych przekroczenia zakresu zmiennej.

Bazując na wyznaczonych wartościach macierzy $\mathbf{qllr}$ oraz $\mathbf{rllr}$ realizowane jest wstępne dekodowanie bazujące na wartości tymczasowej z poprzedniego kroku algorytmu. Na jego podstawie wyznaczane są twarde decyzje bazując na znaku. Końcowym etapem jest obliczenie równania kontrolnego oraz sprawdzenie warunku zakończenia. Jeżeli równanie kontrolne jest spełnione oznacza to, że zostało zakończone dekodowanie tego słowa.

W przypadku, równanie kontrolne nie zostanie spełnione oraz zostanie osiągnięta maksymalna liczba iteracji algorytm przerwie działanie i zwróci błąd. Wtedy jest wiadome, że słowo jest niepoprawne i obarczone błędami.

Wyniki liczbowe działania algorytmu dokowania na urządzeniu IoT

Implementacja dekodera została zrealizowana z wykorzystaniem mikrokontrolera firmy ST o oznaczeniu STM32L476, należącym do serii o bardzo niskim poborze mocy. Mikrokontroler jest oparty o wydajny 32-bitowy rdzeń Arm Cortex M4, pracujący z częstotliwością do 80MHz i posiada wbudowane 1MB pamięci Flash oraz 128KB pamięci SRAM. Dzięki swoim możliwościom obliczeniowym oraz trybom niskiego poboru energii, układ jest dobrym rozwiązaniem dla urządzeń Internetu Rzeczy, stosowanym w wielu przemysłowych rozwiązaniach.

Z wykorzystaniem zaprezentowanego mikrokontrolera przeprowadzono analizy dla różnych macierzy $\mathbf{H}$ w dwóch wariantach:

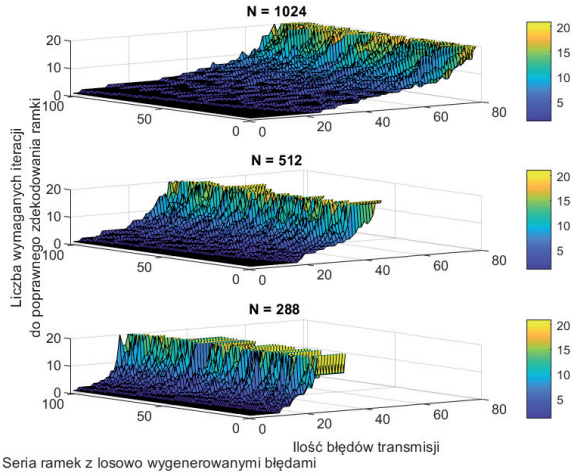
- stała sprawność kodu o różne długości słowa
- stała długość słowa o różnej sprawności

Stała sprawność

Porównano, jak zachowuje się algorytm dekodowania w zależności od liczby błędnych bitów w odebranym słowie kodowym. Błędy były wprowadzane w sposób pseudo-losowy, w liczbie od 0 do 80 bitów. Dla każdej macierzy $\mathbf{H}$ wykonano 100 prób. Na rys. 3 przedstawiono porównanie wyników dekodowania dla sprawności $\frac{1}{2}$, dla różnych długości słowa kodowego N, przy maksymalnej liczbie iteracji dekodowania ustalonej na 20. Przedstawiono liczbę iteracji potrzebnych do zdekodowania poszczególnych słów kodowych. Osiągnięcie limitu 20 iteracji wskazuje bloki, które nie zostały prawidłowo zdekodowane. Do uzyskania tych wyników eksperymentalnych wykorzystano macierze $\mathbf{H2}$, $\mathbf{H4}$, $\mathbf{H8}$, wymienione w tab. 1. Zauważyć można, jak długość bloku kodowego, a tym samym liczba bitów nadmiarowych wpływa na możliwości dekodera. Najdłuższy blok z 512 bitami nadmiarowymi potrafi skorygować do ok. 80 błędów, blok z 256 bitami nadmiarowymi potrafi skorygować do około 40 błędów, natomiast blok z 144 bitami nadmiarowymi potrafi skorygować do około 30 błędów.

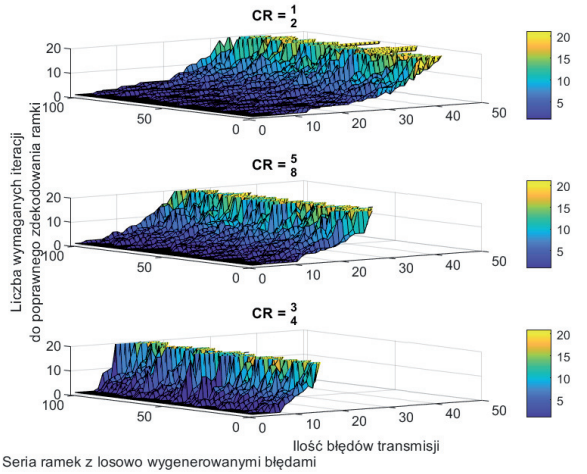
Stała długość słowa

Porównano, jak zachowuje się algorytm dekodowania w zależności od liczby błędów transmisji. Błędy dla każdego wektora testowego były wprowadzane w sposób pseudo-losowy, w liczbie od 0 do 50. Dla każdej macierzy $\mathbf{H}$ wykorzystano 100 wektorów testowych. Na wykresie 4 przedstawiono porównanie dla długości słowa $N = 512$ dla różnych sprawności kodu R. Do porównania wykorzystano macierze $\mathbf{H3}$, $\mathbf{H5}$, $\mathbf{H6}$ z tab. 1. Zauważalne jest jak liczba bitów nadmiarowych wpływa na możliwości dekodera.



Rys. 3. Wyniki procedury iteracyjnego algorytmu dekodowania LDPC dla stałej sprawności

Kod o sprawności $\frac{1}{2}$ potrafi osiągnąć najmniejszą liczba iteracji dla tej samej liczby błędów. Potrafi zdekodować nawet 50 błędów transmisji w zadanej liczbie iteracji. Kod o sprawności $\frac{5}{8}$ potrafi zdekodować do około 25 błędów transmisji. Kod o sprawności $\frac{3}{4}$ potrafi zdekodować do 15 błędów transmisji.



Rys. 4. Wyniki procedury iteracyjnego algorytmu dekodowania LDPC dla stałej długości słowa

Wyniki czasowe oraz przepustowość algorytmu dekodowania

Przepustowość dekodera (ang. throughput – TH) można określić z zależności (1), gdzie K jest długością słowa informacyjnego, a T_{exec} to czas działania dekodera. Czas dekodowania można rozłożyć na 3 elementy: czas inicjalizacji T_{init}, czas dekodowania T_{decode}, czas generowania informacji wyjściowej T_{out}, jak w (2). Według testów proces inicjalizacji może zostać wykonany raz na początku, a czas generowania informacji wyjściowej jest znikomo mały w porównaniu do czasu dekodowania. Zatem $T_{exec} \approx T_{decode}$. Czas dekodowania obejmuje czas wykonywania instrukcji oraz sumaryczny czas dostępów do pamięci, przy czym oba czasy są proporcjonalne do liczby niezerowych elementów w macierzy kontrolnej. Przedstawiono to za pomocą zależności (4), gdzie C określa średnią liczbę elementów niezerowych w każdym wierszu macierzy $\mathbf{H}$, natomiast I to maksymalna

liczba iteracji. Wreszcie przepustowość można wyznaczyć równaniem (6), gdzie α jest określonym eksperymentalnie współczynnikiem proporcji złożoności obliczeń i opóźnienia dostępu do pamięci. Wynika z tego, że przepustowość zależna jest od współczynnika α, średniej liczby niezerowych elementów w każdym wierszu macierzy $\mathbf{H}$, liczby iteracji oraz sprawności kodu. Współczynnik α został wyznaczony doświadczalnie na podstawie czasu dekodowania jednej iteracji i dla każdej z macierzy $\mathbf{H}$ jest zbliżony. Średnia wartość współczynnika α dla eksperymentów przeprowadzonych na wybranych macierzach wyniosła $1,83 * 10^{-6}$. Korzystając z tej wartości współczynnika α, błąd określenia przepustowości z zależności (6) nie przekracza 5% dla wykorzystanych w pracach badawczych macierzy kontrolnych.

$$TH = \frac{N}{T_{exec}} \tag{1}$$

$$T_{exec} = T_{init} + T_{decode} + T_{out} \tag{2}$$

$$T_{decode} = T_{operate} + T_{memory} \tag{3}$$

$$T_{decode} \approx\propto \times N \times C \times I \tag{4}$$

$$TH = \frac{N}{\alpha \times C \times I \times N} = \frac{1}{\alpha \times C \times I} \tag{5}$$

Typ macierzy H	Czas dekodowania jednej iteracji [ms]	Przepustowość [bit/s]
H1	230	4452
H2	245	4180
H3	115	4452
H4	118	4339
H5	150	3012
H6	185	2116
H7	65	4431
H8	65	4431

Tablica 2. Wyniki czasowe oraz przepustowość algorytmu dekodowania

Typ macierzy H	Długość słowa kodowego N	Średnia liczba elementów niezerowych w każdym wierszu C	Współczynnik złożoności obliczeniowej α
H1	1024	6	1,87E-06
H2	1024	6,5	1,84E-06
H3	512	6	1,87E-06
H4	512	6,5	1,77E-06
H5	512	9	1,84E-06
H6	512	13	1,82E-06
H7	288	6	1,88E-06
H8	288	6,5	1,74E-06

Tablica 3. Parametry potrzebne do wyznaczenia przepustowości algorytmu dekodowania

Istotnym jest, że przepustowość nie jest zależna od wielkości słowa kodowego, a jedynie od liczby elementów niezerowych w kolumnie, co dla algorytmu bazującego na indeksach elementów niezerowych jest poprawne, a jedynie czas dekodowania jest zależny od wielkości słowa kodowego. Dla kodów nieregularnych zastosowano wartość średnią obliczoną na podstawie liczby jedynek w każdym wierszu.

Wnioski

W artykule przedstawiono implementację nowoczesnego kodowania korekcyjnego LDPC w układzie mikrokontrolera, które może być wykorzystane w protokołach komunikacyjnych przetwarzających niewiele danych, przy ograniczonych zasobach obliczeniowych, co jest charakterystyczne dla urządzeń typu IoT. W szczególności skupiono się na układzie dekodera. Z przeprowadzonych eksperymentów można wyciągnąć pewne praktyczne wnioski projektowe. W szczególności zauważono, że kody nieregularne mają lepsze własności korekcyjne. Dla tych samych parametrów macierzy $\mathbf{H}$ w przypadku macierzy regularnej oraz nieregularnej, różnica jest znaczna i kształtuje się na poziomie około od 50% do nawet 90% większej liczby zdekodowanych błędów w tej samej liczby iteracji. Pokazano, że możliwa jest implementacja dekodera kodów LDPC oraz QC-LDPC, także nieregularnych, w układzie mikrokontrlera, mimo znacznych wymagań dotyczących mocy obliczeniowej dla kodów LDPC. Wskazano, jaka przepustowość może być uzyskana oraz zaproponowano przybliżony wzór obliczania przepustowości dekodera zaimplementowanego w typowym mikrokontrolerze.

***Autorzy**: Ph.D. Wojciech Sułek, Jakub Hyla Institute of Automatic Control and Robotics, Electrinics and Telefomunication, Silesian University of Technology ul. Akademicka 2a, 44-100 Gliwice, Poland, email: jakubhyla93@gmail.com,*

This research was co-financed by the Ministry of Education and Science of Poland under grant No. DWD/3/7/2019.

LITERATURA

[1] Ieee 802.11-2016. ieee standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications, 2016.

[2] Salima Belhadj and Moulay Lakhdar Abdelmounaim. On error correction performance of ldpc and polar codes for the 5g machine type communications. In *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, pages 1–4, 2021.

[3] Xiaoheng Chen and Chang-Li Wang. High-Throughput Efficient Non-Binary LDPC Decoder Based on the Simplified Min-Sum Algorithm. *IEEE Trans. Circuits Syst. I*, 59:2784–2794, November 2012.

[4] ETSI Standard: EN 302 307 v1.1.1, *Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications*, 2005.

[5] Robert G. Gallager. Low-Density Parity-Check Codes. *IRE Transactions on Information Theory*, IT-8:21–28, January 1962.

[6] Jakub Hyla, Wojciech Sułek, Weronika Izydorczyk, Leszek Dziczkowski, and Wojciech Filipowski. Efficient ldpc encoder design for iot-type devices. *Applied Sciences*, 12(5), 2022.

[7] IEEE Standard: IEEE P802.11n=D10. Draft IEEE Standard for Local Metropolitan Networks – Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC), and Physical Layer (PHY) specifications: Enhancements for Higher Throughput, March 2006.

[8] Shu Lin and Daniel J. Costello, Jr. *Error Control Coding: Fundamentals and Applications, 2nd Edition*. Prentice-Hall, Inc., Upper Saddle River, New Jersey 07458, 2004.

[9] Bing Liu, Rongke Liu, Zhanxian Liu, and Ling Zhao. An efficient implementation of ldpc decoders on arm processors. In *2018 IEEE International Workshop on Signal Processing Systems (SiPS)*, pages 1–5, 2018.

[10] David J. C. MacKay. Good Error-Correcting Codes Based on Very Sparse Matrices. *IEEE Trans. Inf. Theory*, 45:399–431, March 1999.

[11] David J. C. MacKay and Radford M. Neal. Near Shannon Limit Performance of Low Density Parity Check Codes. *Electronics Letters*, 32:1645–1646, August 1996.

[12] Jérémy Nadal and Amer Baghdadi. Parallel and flexible 5g ldpc decoder architecture targeting fpga. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 29(6):1141–1151, 2021.
[13] Vladimir L. Petrović, Dragomir M. El Mezeni, and Andreja Radošević. Flexible 5g new radio ldpc encoder optimized for high hardware usage efficiency. *Electronics*, 10(9), 2021.
[14] Thomas J. Richardson, M. Amin Shokrollahi, and Rüdiger L. Urbanke. Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes. *IEEE Trans. Inf. Theory*, 47:619–637, February 2001.
[15] Chance Tarver, Matthew Tonnemacher, Hao Chen, Jianzhong Zhang, and Joseph R. Cavallaro. Gpu-based, ldpc decoding for 5g and beyond. *IEEE Open Journal of Circuits and Systems*, 2:278–290, 2021.
[16] Saleh Usman and Mohammad M. Mansour. Fast column message-passing decoding of low-density parity-check codes. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 68(7):2389–2393, 2021.

5.4 Publikacja 4: Energy-efficient Raptor-like LDPC coding scheme design and implementation for IoT communication systems.

energies

MDPI

Article

Energy-Efficient Raptor-like LDPC Coding Scheme Design and Implementation for IoT Communication Systems

Jakub Hyla [1] and Wojciech Sułek [2,*]

[1] TKH Technology Poland Sp. z.o.o., 64-100 Leszno, Poland; j.hyla@tkhtechnology.com
[2] Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology, 44-100 Gliwice, Poland
* Correspondence: wojciech.sulek@polsl.pl

Abstract: The large number of inexpensive and energy-efficient terminals in IoT systems is one of the emerging elements of the recent landscape of information and communication technologies. IoT nodes are usually embedded systems with limited processing power devices and strict requirements on energy consumption. In this paper, we consider the design and implementation of a part of the IoT communication uplink stack, namely the error correction coding scheme, for energy-efficient operation. We examine how an efficient rate-adaptive coding scheme, namely the Raptor-like (RL) quasi-cyclic (QC) subclass of low-density parity check (LDPC) codes, can be applied. We present an encoding algorithm designed for an embedded CPU; a respective QC-RL-LDPC decoding scheme, based on typical LDPC iterative decoding; and a combined design procedure for construction of the QC-RL-LDPC parity check matrices. Next, we conduct experiments to explore the time intervals for implemented encoding and the goodput of the coding system with incremental redundancy. We provide the statistical results by combining the measured energy consumption of the encoder and the simulated radio transmitter energy consumption. We demonstrate the comparison of normalized energy consumption statistics with the fixed-rate LDPC coding case. As conclusion, we note that under an unknown channel corruption level, the short block QC-RL-LDPC code implemented in the CPU can achieve higher energy efficiency per information bit compared to a fixed-rate LDPC code. Future work will include an extension of the work to nonbinary QC-RL-LDPC coding design.

Keywords: error correction; ECC; low-density parity check codes; LDPC; QC-LDPC; IoT systems

Citation: Hyla, J.; Sułek, W. Energy-Efficient Raptor-like LDPC Coding Scheme Design and Implementation for IoT Communication Systems. *Energies* **2023**, *16*, 4697. https://doi.org/10.3390/en16124697

Academic Editors: Daniele D. Giusto and Matteo Anedda

Received: 30 April 2023
Revised: 7 June 2023
Accepted: 9 June 2023
Published: 14 June 2023

1. Introduction

The Internet of Things, or IoT, is typically perceived as a system of a large number of interrelated computing devices, often constituting a kind of cyberphysical system. It is often made up of elements (IoT nodes) that are inexpensive terminals, with an important feature of optimized energy efficiency of operation [1]. The general concept of the IoT includes many possibilities for designing the hardware and software of the system. There is no single unified IoT architecture, and there is no common definition of communication protocols for IoT parts. While developers use existing technologies to build the IoT, research groups are still working to adapt protocols and improve communication quality and energy efficiency [1,2].

The IoT nodes that make up the bottom level of the system are typically battery-powered devices based on embedded low-power processors (microcontrollers) with limited processing capabilities. When the system requires a wireless communication protocol, the energy efficiency of such embedded devices could be highly dependent on the energy efficiency of the data communication protocol. Since the messages sent from/to constrained IoT nodes are typically relatively short, the important factor is also the activity time of the CPU and radio modem, in relation to the inactivity time (when the device is in sleep mode). Due to the limited resources of the processing units of the IoT nodes,

efficient communication protocols are desired to enable communication with reasonable energy efficiency.

The effectiveness and reliability of communication depend heavily on the error control coding (ECC) mechanism used. Numerous effective coding schemes, such as LDPC coding [3], turbo coding [4] and polar coding [5], have been developed in recent years. One problem with the implementation of these schemes is that they they require computationally demanding iterative algorithms for decoding. These coding schemes are typically utilized in complex communication protocols such as WiFi, WiMAX and 5G, with high-performance computing hardware required for implementation.

In the case of a time-varying communication channel, additional improvement in the uplink transmission efficiency can be obtained if a properly designed rate-adaptive or rateless system is employed, because of the two limitations of the traditional fixed-rate coding, inherently designed for a specific channel noise level (or SNR, signal-to-noise ratio). First, if the SNR is significantly higher than the level that the fixed-rate code has been designed for, then the rate-adaptive code can correct error with higher rate, which means a lower number of coded bits must be encoded and delivered. Second, if, on the other hand, the SNR is significantly lower than expected, then the fixed-rate code fails to provide sufficient redundancy to combat channel corruptions, while the rate-adaptive code can adapt and provide such additional redundancy.

Rateless codes are a class of ECC that can generate an unlimited number of coded bits given a finite number of information bits. It is the core of Hybrid ARQ (HARQ) protocols, which allows for efficient transmission of signals over a channel that changes over time. The system's transmitter first sends the high-rate codeword of the necessary size over the channel. If the receiver cannot decode the received noisy codeword, the transmitter then sends repeated or additional parity symbols to the receiver to allow the receiver to attempt another round of decoding, but now with a higher redundancy level. The process continues until the decoder successfully decodes the received codewords. HARQ can be classified into two main types: type-I with chase-combining (CC) and type-II with incremental redundancy (IR). The type-II IR provides superior throughput performance to CC-HARQ, due to the retransmission of only incremental parity bits, when required [6,7]; therefore, we consider IR HARQ in this research.

The important and well-investigated case of rateless codes is Raptor coding [8]. A Raptor code is a concatenation of a high-rate LDPC outer code and a rateless Luby transform (LT) inner code. Raptor codes are shown to achieve the capacity of the binary erasure channel (BEC), while research to design them for the additive white Gaussian noise (AWGN) channel is already quite advanced [9–11] . While LT codes and Raptor codes share some similarities with LDPC codes, more closely related to LDPC are protograph-based Raptor-Like (PBRL) LDPC codes [12,13]. Unlike the Raptor codes, in the PBRL scheme, a high-rate LDPC codeword is transmitted directly as the initial block, and additional parity symbols are sent only if necessary. PBRL-LDPC codes are well suited for low-complexity encoder implementation, because the well-known LDPC and QC-LDPC efficient encoding schemes can be adapted [14,15].

The general rate-adaptive communication system model that we consider in our research is presented in Figure 1. We investigate a system based on a low-complexity embedded device that acts as an IoT node and can function in a variety of noisy environments, regardless of the communication channel. The communication scheme automatically adapts to the channel; hence, the system can work properly in a wide range of noisy environments. The developed system is classified as a type II hybrid automatic repeat request (HARQ) [6], which involves an initial transmission of a length-K block, followed by transmissions of shorter length-P chunks. The receiver combines these chunks in a decoding scheme, which signalizes the successful decoding by sending positive acknowledge (ACK), after receiving as many chunks as are needed for correct decoding. The coding scheme that we employ is PBRL-LDPC, constrained to quasi-cyclic (QC) submatrices, which we refer to as QC Raptor-like (RL) coding (QC-RL-LDPC).

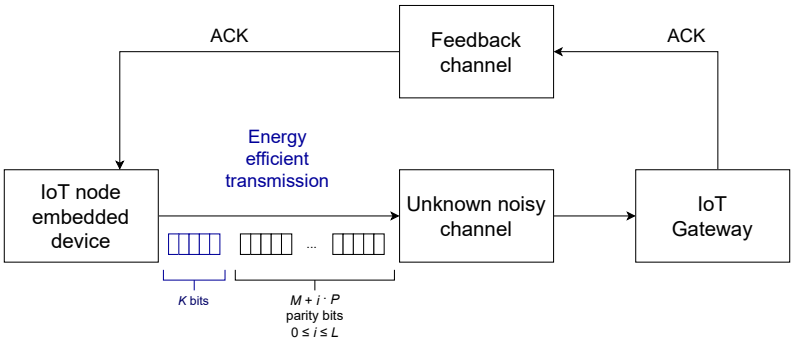


Figure 1. Considered model of the IoT uplink communication with incremental redundancy ECC scheme.

Since the gateway in Figure 1 is assumed to possess significantly more computational resources, it is possible to consider an ECC scheme with high decoding complexity for data blocks sent in an uplink direction. Therefore, we claim RL-LDPC coding to be a good choice due to its great correction performance, which can potentially also reduce the transmission power needed, and in consequence, enhance the transmitter energy efficiency. However, in the proposed system, the QC-RL-LDPC encoder must still be implemented with an embedded CPU in an IoT node.

In this context, the research presented in this paper is initiated. Although there have been many research efforts in the implementation of LDPC and Raptor-like encoding and decoding, most of them involve high-throughput FPGA or ASIC hardware implementations, while CPU codec investigations are missing. However, we show that with some effort, the encoding side can be designed in a way that also enables implementation in low-power processors with limited processing capabilities. Moreover, for the system designed according to the outline in Figure 1, the energy efficiency of the IoT node operation is very important. We note that the energy efficiency of the node radio transmitter can be enhanced by using a modern and well-performing ECC, such as the LDPC coding scheme, and additional enhancement in the case of unknown channel corruption can be achieved by a rate-adaptive scheme devoted from LDPC.

Therefore, this paper concerns the design of the entire system, including the construction of flexible QC-RL-LDPC codes and the implementation of the encoding scheme in the IoT node of Figure 1, as well as the experimental investigation of the uplink transmission system performance and the energy efficiency, especially by comparing the incremental redundancy system with the standard fixed-rate coding. This comparison is made in terms of energy efficiency of message transmissions, which is dependent on both the encoding energy consumption and the transmitter energy consumption, which are experimentally assessed in this research.

Our focus in energy efficiency assessment is on the IoT node (transmitter) procedures that have been formulated and implemented in a microcontroller CPU device. Other details of the communication protocol are abstracted as a typical digital modulation scheme and noisy communication channels; however, for complete system description, we also provide the decoding algorithm, based on modified message-passing flow. The detailed particular contributions of this paper can be listed as follows.

1. With regard to implementation, we show how the QC-RL-LDPC encoding can be efficiently implemented in a CPU device, by decomposing the encoding into simple cyclic shifts and GF(2) addition operations, in direct correspondence to the CPU program instructions.
2. We verify the designed CPU implementation of the encoder, with several QC-RL-LDPC matrix representations, measuring the encoding time and providing measurement results.

3. We provide a concise procedure for a QC-RL-LDPC binary code design, which is based on the PEG construction of base graph [16] of the HR code; an IR part optimization with reciprocal channel approximation, as in [13]; and then combined lifting of protographs with a proprietary computer search algorithm.
4. For the receiver side of the system, we provide an effective decoding scheme, based on LDPC belief propagation (BP) decoding, with incremental redundancy.
5. We perform a number of experiments of the entire system operation, providing the results in the form of simulated transmission goodput, which is defined as the ratio of the number of information bits transmitted to the total number of modulation symbols utilized in the IR, required for successful decoding. Several different designed QC-RL-LDPC codes with different code lengths are experimentally tested. Moreover, we also simulate rate-adaptive coding based on state-of-the-art 5G codes [17,18] and compare them with our designed codes.
6. Taking into account the timing results and system simulation goodput results, we assess the energy consumption of the encoder plus radio transmitter in a typical IoT wireless transmission scenario, and provide the results in the form of energy consumption normalized per transmitted information bit, which allows for an energy efficiency comparison with a fixed-rate coding system, as well as with a 5G coding system. The experimental results show the energy efficiency gain that can be achieved in comparison to fixed-rate coding.

2. Research Methods

The QC-RL-LDPC encoder is implemented in low-power microcontroller unit from the STM32 family, as will be presented in Sections 5 and 8.1. All the remaining parts of the communication model considered, presented in Figure 1, are implemented as software models developed in the Matlab environment. The research methodology is outlined in Figure 2.

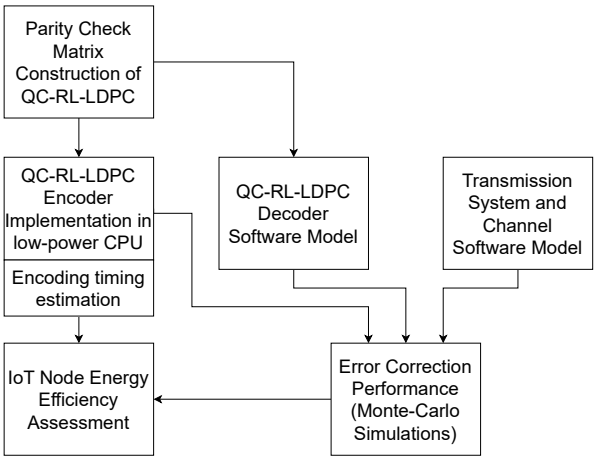


Figure 2. Outline of research methods.

The parity check matrix construction is a flexible algorithm that enables construction of QC-RL-LDPC code with any code length, base rate, submatrix size and chunk size, as is presented in Section 7. The developed decoding algorithm is presented in Section 6, and in general, it can be used in software or hardware implementations; however, for this research, the software model created in Matlab is used. Similarly, the transmission system with QAM/QPSK modulation and communication channel are modeled with software.

The models are then used in Monte-Carlo simulations that provide, as a result, the estimated error correction performance of the system, in the function of channel corruption level represented by signal-to-noise ratio (SNR). The simulation results are provided in Section 8.3. The second part of the experimental design is the encoding timing estimation, which is conducted by injection into the encoder source code, a time measurement unit based on one of the hardware timers in the STM32 microcontroller. The timer is executed for as long as the encoding and the measurement are read by an interface with PC computer and then converted into time units. The results of these experiments are presented in Section 8.1.

Finally, both the system performance results and the encoding timing results are used to assess energy consumption of the coding and transmission module in an IoT node, and the results are provided in Section 8.4.

3. LDPC Coding—Preliminaries

The low-density parity check coding was initially proposed by Robert Gallager in [19], and later recalled by MacKay [3,20], and demonstrated that irregular LDPC codes can asymptotically approach the Shannon limit [21] of lossless transmission. In the last decade, LDPC codes have become very popular as a powerful error correction coding method with the possibility of algorithm parallelization that enables the development of hardware codecs with optimized algorithms and very high throughput [22–25], typically for high-speed industrial communication standards [26,27].

The particular LDPC code is defined by its parity check matrix (PCM) $\mathbf{H}$, which defines the specific parity checks involving the transmitted message bits. The important feature of PCM, which is its sparseness, allows LDPC codes to be iteratively decoded using low-complexity message passing algorithms [3]. Moreover, low-complexity encoding algorithms have also been well recognized [14]. Recent research in the field also involves nonbinary coding algorithms, as well as code design and implementation [28–30].

3.1. LDPC Codes

The binary parity check matrix $\mathbf{H}$ of a binary (N, K) LDPC code has a size $M \times N$, where N is the length of the code vector (encoded message), $K < N$ is the length of an information vector, and $M = N - K$ represents the number of parity bits in the code vector. The measure of coding redundancy is the code rate, defined as $R = K/N$. In the systematic encoding process, M parity bits are added to the information vector $\mathbf{u} = \{u_1, u_2, \ldots, u_K\}$, forming the code vector $\mathbf{c} = \{c_1, c_2, \ldots, c_N\}$. All vectors are binary, and it is convenient to consider them to be defined over the Galois finite field GF(2).

In the decoder, the data received $\mathbf{y}$, which is a vector of length N, are corrupted by channel imperfections. If—and only if—the hard decisions made from $\mathbf{y}$ meet the parity check equation, it will be recognized as a correct, (error-free) vector $\hat{\mathbf{c}}$. The parity check equation is expressed as $\mathbf{H}\hat{\mathbf{c}}^T = \mathbf{0}_{M\times 1}$, where the matrix operation is in GF(2) arithmetic. If the equation is not satisfied, the iterative decoding algorithm is started that can correct errors, with soft decisions as the algorithm input, which are usually in the form of log-likelihood ratio (LLR) vector:

$$LLR(\mathbf{c}|\mathbf{y}) = \left[\log\frac{P(c_1 = 0|\mathbf{y})}{P(c_1 = 1|\mathbf{y})}, \log\frac{P(c_2 = 0|\mathbf{y})}{P(c_2 = 1|\mathbf{y})}, \cdots, \log\frac{P(c_N = 0|\mathbf{y})}{P(c_N = 1|\mathbf{y})}\right]. \quad (1)$$

Alternatively to matrix representation, LDPC codes can also be represented by the associated bipartite Tanner graph [3,31]. In the Tanner graph, the variable nodes (VN) representing codeword bits are associated with columns of $\mathbf{H}$ and the check nodes (CN) representing parity checks are associated with rows of $\mathbf{H}$. The nonzero elements in $\mathbf{H}$ correspond to edges in the code graph. This means that the graph nodes are divided into two distinct sets (VNs and CNs), and that the edges only connect two different types of nodes. For an (N, K) LDPC code with block length N, the graph consists of N variable nodes and $M = N - K$ check nodes.

The most popular LDPC decoding algorithms are all algorithms involving belief propagation (BP), including sum product (SPA) algorithms and several simplified versions operating on LLR beliefs, such as min-sum algorithms (MSA), offset min-sum algorithms (OMSA), or normalized min-sum algorithms (NMSA) [32–34].

Since the message length in our assumed IoT system is assumed to be relatively short, the block lengths of codes that are considered and investigated in this article are short-to-medium. We investigate such codes constructed with a developed procedure that will be briefly presented in one of the next sections.

3.2. Quasi-Cyclic LDPC Codes

Many communication systems use the quasi-cyclic (QC) subclass of LDPC codes [15]. These codes have properties that make them suitable for efficient encoder and decoder designs, as well as for industrial implementations [17,24,35,36]. The PCM of QC-LDPC codes has a distinct structure composed of circulant permutation matrices (CPMs):

$$\mathbf{H} = \begin{bmatrix} \mathbf{P}^{s_{1,1}} & \mathbf{P}^{s_{1,2}} & \dots & \mathbf{P}^{s_{1,C}} \\ \mathbf{P}^{s_{2,1}} & \mathbf{P}^{s_{2,2}} & \dots & \mathbf{P}^{s_{2,C}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}^{s_{R,1}} & \mathbf{P}^{s_{R,2}} & \dots & \mathbf{P}^{s_{R,C}} \end{bmatrix}, \tag{2}$$

where $\mathbf{P}^{s_{r,c}}$ is either a $P \times P$ all-zero submatrix (by convention, it is indicated by $s_{r,c} = \infty$) or an $s_{r,c}$-CPM of size $P \times P$, that is, a matrix obtained by cyclically shifting every row of an identity matrix $\mathbf{I}_{P\times P}$ by $s_{r,c}$ positions to the right [37]. The particular code can be unambiguously defined by its size and the values of $s_{r,c} \in \{0, 1, \dots, P-1, \infty\}$, for macrorows $r = 1, \dots, R$, and macrocolumns $c = 1, \dots, C$.

The design of QC-LDPC codes can be treated as an important special case of the protograph-based LDPC code construction process [38]. In essence, this construction process is based on the copying and edge permuting of a smaller graph (protograph, base graph) to achieve a larger Tanner graph of a QC-LDPC code.

3.3. Raptor-like LDPC Codes

In contrast to block coding with a fixed code rate, rateless coding is an error correction scheme that, given a fixed information vector, generates a (potentially unconstrained) series of redundant bits [39], making use of IR decoding on the receiver side. However, constructing good rateless codes with the possibility of efficient encoding and decoding requires some effort. One of the recently achieved solutions is protograph-based Raptor-like (PBRL) LDPC coding [12,13].

The PBRL-LDPC code can be considered to be a concatenation of a high-rate code (HRC), which is an LDPC code with a parity check matrix $\mathbf{H}_{\mathrm{HR}}$ of size $M \times N$ and an incremental redundancy code (IRC) characterized by a low-density generator matrix (LDGM) $\mathbf{G} = [\mathbf{I}, \mathbf{H}_{\mathrm{IR}}^T]$ [13]. In this paper, we employ the QC subclass of PBRL-LDPC codes, which will be called QC-RL-LDPC codes.

The matrices defining the QC-RL-LDPC code $\mathbf{H}_{\mathrm{HR}}$ and $\mathbf{H}_{\mathrm{IR}}$ both have a structure consisting of CPMs, as in (2). The size of matrix $\mathbf{H}_{\mathrm{IR}}$ is $LP \times N$, which means that it has L macrorows of $P \times P$ circulants, giving a total of LP parity checks in the incremental redundancy scheme. The Tanner graph of such a QC-RL-LDPC code can be illustrated as presented in Figure 3, where the blocks denoted by $\mathbf{H}_{\mathrm{HR}}$ and $\mathbf{H}_{\mathrm{IR}}$ are edge permutators according to the matrices' structures.

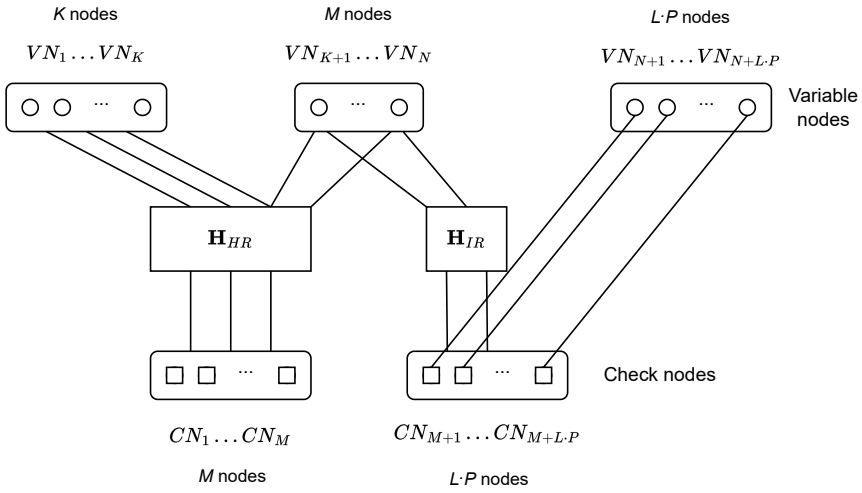


Figure 3. Tanner graph of a QC-RL-LDPC code.

4. The Developed System

This coding system is specifically created for a low-complexity IoT embedded device that communicates using a radio transmission protocol, with an energy-efficient ECC. The IoT node must meet certain criteria, such as being low-complexity and embedded-MCU-based, requiring a wireless communication system for various conditions and prioritizing energy efficiency of the radio transmitter during the design process. With these considerations in mind, it is suggested that a modern incremental-redundancy-based transmission could be advantageous.

The model of the proposed communication system employing incremental redundancy is presented in Figure 4. On the transmitter side, the HR (high-rate) and IR (incremental redundancy) parts of coded message are distinguished. The HR parity bits vector $\mathbf{p}_{HR}$ is an element of the codeword $\mathbf{c}$ that is initially transmitted, with a chosen transmission method. In the model presented in Figure 4, we assume that QAM modulation over a wireless communication channel is used; however, other transmission techniques and mediums can also be supported. After the initial transmission of $\mathbf{c}$, the transmitter generates and subsequently transmits IR parity bits in vectors $\mathbf{p}_{IR_j}$, $j = 1, \dots, L$ as long as it receives an acknowledgment (ACK) of successful decoding. The IR vectors are of length P, and will also be called IR chunks.

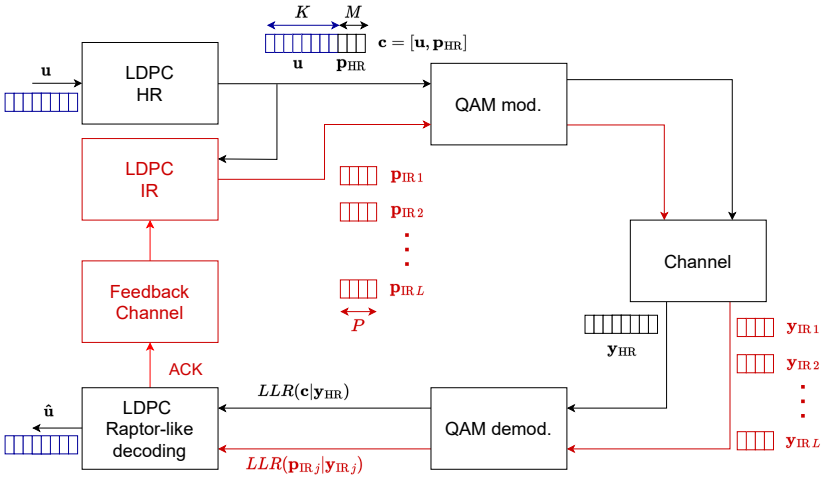


Figure 4. Communication system model with the developed QC-RL-LDPC coding.

The developed system is a kind of adaptive-rate scheme, in which the coding rate of the transmission system is variable in the range $R_{\max} \geq R \geq R_{\min}$, with the highest rate corresponding to the HR code being as in (3), and the lowest rate (4) reflecting the situation where the maximum number of chunks is required for correct transmission. If decoding is still unsuccessful after sending L IR chunks, the message can be retransmitted, either immediately or scheduled for later retransmission.

$$R_{\max} = \frac{K}{N}, \qquad N = K + M \tag{3}$$

$$R_{\min} = \frac{K}{K + M + L \cdot P} \tag{4}$$

We performed an analysis of the energy efficiency of a developed transmitter embedded in an IoT device, with microcontroller (MCU) hardware. The receiver is assumed to be implemented with any device (CPU/GPU/hardware) with significantly greater computational power. It is assumed that either:

- The computational resources of the receiver (gateway) are far greater than the resources of the transmitter (IoT node), to the extent that IR decoding is performed in a fraction of the encoding time, as well as the feedback channel latency can be neglected; or
- The CPU that implements encoding in the transmitter, after encoding a $\mathbf{p}_{IR_j}$ chunk enters a sleep state for a specified time needed to receive a feedback message, in which state the energy consumption is negligible.

If one of these points is satisfied, the ACK can be assumed to be received virtually immediately after transmitting only as many chunks as are needed for correct decoding. This assumption is important for the energy efficiency calculations we made.

In the following sections, we provide the details of the whole designed system: the encoder design for a CPU; the decoding algorithm that was developed based on LDPC iterative decoding; and the code construction algorithm that can be used to design QC-RL-LDPC codes of arbitrary lengths and submatrix sizes. We provide experimental results in the form of encoding and decoding timings, and then energy consumption. We show that, in the case of unknown channel SNR, the implemented scheme can achieve a greater transmission energy efficiency than a fixed-rate coding system.

We also present the developed decoding scheme, based on the iterative LDPC decoding algorithm, that can be utilized in an incremental redundancy QC-RL-LDPC system. While the receiver (gateway) hardware is undefined but usually possesses a great computing power, a QC-RL-LDPC decoder in a CPU has also been developed for possible use in the reverse (gateway to IoT node) direction. We also provide the results of the decoder implementation for the same CPU as on the encoder side.

5. QC-RL-LDPC Encoding

The encoding of QC-RL-LDPC codes, as shown in Figure 4, always involves the HR part and then as many chunks in the IR part as are required to efficiently decode the message in the receiver.

The encoding of the HR part, that is, determining $\mathbf{p}_{\mathrm{HR}}$, can be performed with one of the known efficient LDPC encoding algorithms, utilizing the QC feature. In [36], we presented a CPU-based QC-LDPC encoder implementation, which can also be used as an HR part encoder implementation in the QC-RL-LDPC system presented in this article. If this implementation is used, for the sake of code construction, it is important to note that the parity check matrix $\mathbf{H}_{\mathrm{HR}}$, in addition to being in QC form, should follow the ALT (almost lower-triangular) form, as in the Richardson–Urbanke encoding [14], that is,

$$\mathbf{H}_{\mathrm{HR}} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{T} \\ \mathbf{C} & \mathbf{D} & \mathbf{E} \end{bmatrix} \tag{5}$$

where $\mathbf{T}$ is a lower-triangular matrix and all submatrices are CPMs. Cyclic changes in CPMs should preferably be selected so that $\mathbf{\Phi} = \mathbf{E}\mathbf{T}^{-1}\mathbf{B} + \mathbf{D}$ is an identity matrix, $\mathbf{\Phi} = \mathbf{I}_{g\times g}$, which can be achieved by fulfilling a condition originally proposed in [37]. As a result of the encoding of the HR information vector $\mathbf{u}$, the code vector $\mathbf{c} = [\mathbf{u}, \mathbf{p}_{\mathrm{HR}}]$ is obtained. The Details of the encoding algorithm for QC-LDPC and its microcontroller implementation can be found in [14,36,37].

The IR encoding is based on the $\mathbf{H}_{\mathrm{IR}}$ parity check matrix, which can be expressed as follows:

$$\mathbf{H}_{\mathrm{IR}} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \\ \vdots \\ \mathbf{H}_L \end{bmatrix}, \qquad \mathbf{H}_l = \left[\mathbf{P}^{s_{R+l,1}}, \mathbf{P}^{s_{R+l,2}}, \ldots, \mathbf{P}^{s_{R+l,C}}\right], \quad l = 1, 2, \ldots, L \tag{6}$$

where each macrorow $\mathbf{H}_l$ defines the parity checks (7) in GF(2) arithmetic for the lth chunk of incremental redundancy, $\mathbf{p}_{\mathrm{IR}l}$.

$$\mathbf{H}_l \cdot \mathbf{c}^T + \mathbf{P}_{IR_l} = \mathbf{0} \tag{7}$$

The consecutive chunks for $l = 1, 2, \ldots, L$ can be computed as in the following expression:

$$\mathbf{P}_{IR_l} = \mathbf{H}_l \cdot \mathbf{c}^T = \mathbf{P}^{s_{R+l,1}}\mathbf{c}_1^T + \mathbf{P}^{s_{R+l,2}}\mathbf{c}_2^T + \ldots + \mathbf{P}^{s_{R+l,C}}\mathbf{c}_C^T \tag{8}$$

Every chunk calculation as in (8) requires elementary operations of the type $\mathbf{P}^{s_{R+l,j}}\mathbf{c}_j^T$, as well as additions over GF(2), where every $\mathbf{P}^{s_{R+l,j}}$ is either:

- a zero matrix, in which case the product $\mathbf{P}^{s_{R+l,j}}\mathbf{c}_j^T$ is $\mathbf{0}$;
- or a circular shift of an identity matrix by $s_{R+l,j}$ positions, in which case it can be easily verified that the product $\mathbf{P}^{s_{R+l,j}}\mathbf{c}_j^T$ is a circular shift of $\mathbf{c}_j^T$ by $s_{R+l,j}$ positions.

Therefore, computing (8) for the QC case requires elementary operations of circular shifts of binary length-P subvectors in $\mathbf{c}^T$, as well as modulo-2 additions of length-P subvectors. Implementation in the microcontroller can be developed in C language, in which modulo-2 additions are realized with an "^" operator, and the circular shifts can be realized by using a bitwise shift operation and a bitwise OR operation, as in the following example:

```
output = (input << offset) | (input >> (size - offset));
```

where `offset` is the cyclic shift value and `size` is equivalent to the submatrix size P. Using this instruction for the circular shift, the input value can be represented in `uint32_t` type (if $P \leq 32$), while the `size` and `offset` can be represented in any unsigned type, for example, `uint8_t`. The output value is in the same format as the input.

6. Raptor-like Decoding of QC-RL-LDPC Codes

The decoding can be based on the belief propagation (BP) scheme with messages (beliefs) calculated as in the LDPC decoding with an SPA algorithm, or one of its LLR variants. In accordance with the proposal in [13], the overall concept is to decode the HRC and IRC parts together. The joint parity check matrix of both parts, corresponding to the whole Tanner graph of the QC-RL-LDPC code (as in Figure 3), is given as

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{\mathrm{HR}} & \mathbf{0} \\ \mathbf{H}_{\mathrm{IR}} & \mathbf{I} \end{bmatrix} \tag{9}$$

where $\mathbf{I}$ is an identity matrix. BP generally proceeds according to the graph in Figure 3. The decoding algorithm calculates the LLR messages for every edge of the graph in every iteration. However, it can be noted that it is inefficient to initialize the decoding process with a priori beliefs every time the decoder receives a new IR chunk. Therefore, we have developed the decoding procedure, in which the iterative process is suspended if the correct

vector is not found after some number of iterations and then resumed after receiving a new chunk with LLR-input IR information. Other than IR, beliefs (messages exchanged between nodes) remain as obtained in the previous iteration. The decoding algorithm is presented in Figure 5, where the processing sequence as well as the data flow (in blue lines) are shown. The algorithm can be summarized as follows.

1. Soft input decisions of the HR part, $LLR(\mathbf{c}|\mathbf{y})$, are delivered first to the decoder, which proceeds with a number of I_{HR} iterations, computing variable node messages, denoted as Q_{nm}, and check node messages, denoted as R_{mn}, for $n = 1, \ldots, N$ and $m = 1, \ldots, M$.
2. In every iteration, based on the total beliefs for the HR codeword symbols Q_n, the error syndrome $\mathbf{s}$ is checked, and if a correct codeword is obtained, the decoding ends with success.
3. After I_{HR} iterations, the decoder waits for the next IR chunk. The chunk counter is denoted as j, while the iteration counter is i in Figure 5. After every subsequent chunk reception, the decoding is resumed for I_{IR} iterations, but the initial check-to-variable messages R_{nm} are taken from the previously suspended decoding part.

The computations of beliefs R_{mn}, Q_{nm}, for $n = 1, \ldots, N$ and $m = 1, \ldots, M$, can be done with one of the well-known BP message calculation methods, such as min-sum or normalized min-sum [34]. In our experiments, we used a computer model of the corrected min-sum algorithm [34]. An important choice of the number of iteration limits I_{HR} and I_{IR} can be made experimentally. The experimental results of the decoding performance will be provided in Section 8.

For a highly erroneous initial HR code block, the information calculated during initial decoding iterations is not enough to correct errors, but all iteratively enhanced beliefs are saved. Based on it, the next iterations are started, with additional soft information received. The point is that there is no need to calculate the beliefs from the beginning after every chunk transmission.

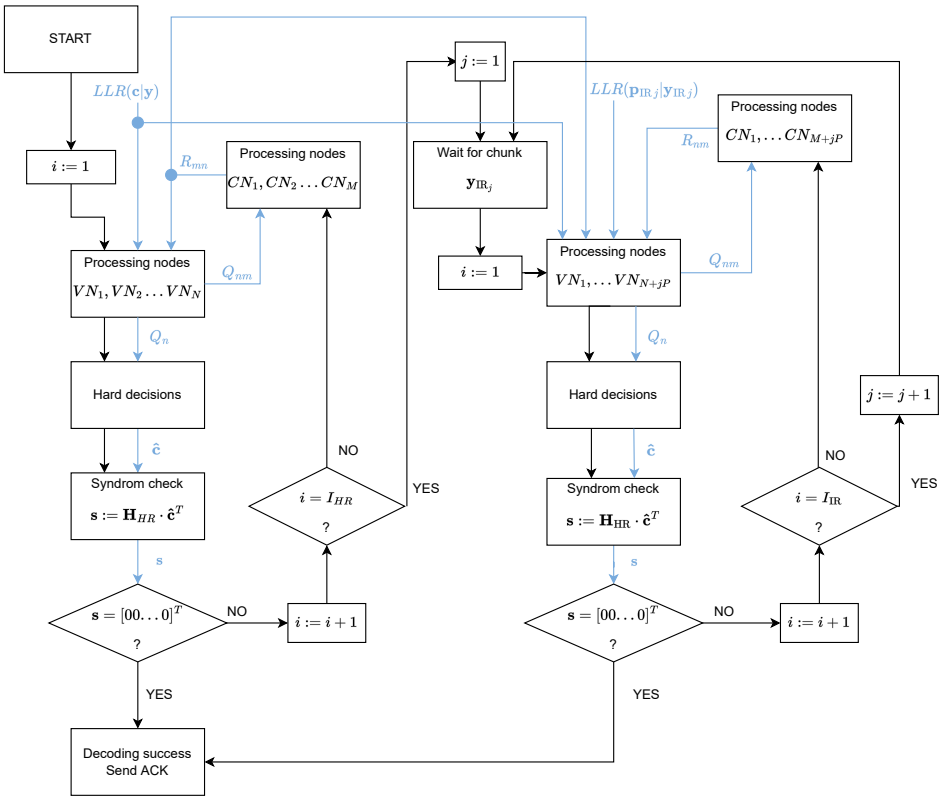


Figure 5. The developed algorithm for Raptor-like decoding.

7. QC-RL-LDPC Codes Construction for the Proposed System

Numerous methods for the construction of QC-LDPC have already been proposed, and a majority of finite code designs are centered around creating a Tanner graph structure with advantageous features, for example, in [40–44]. However, in this investigation, in order to design the entire QC-RL-LDPC coding scheme, we integrated the base graph construction by progressive edge growth (PEG, [40]) with protograph optimization for PBRL codes using density evolution [13] and proprietary algorithms to determine CPMs [36]. This allowed us to create a single, comprehensive method that gives both $\mathbf{H}_{\mathrm{HR}}$ and $\mathbf{H}_{\mathrm{IR}}$ as a result.

The proposed approach to the construction of the QC-RL-LDPC can be summarized as in the following steps:

1. The HRC base graph, that is, a graph corresponding to the $R \times C$ base matrix $\mathbf{W}_{\mathrm{HR}}$ indicating nonzero submatrices in $\mathbf{H}_{\mathrm{HR}}$ in (2), is constructed with an improved PEG algorithm [16]. By row and column permutations, $\mathbf{W}_{\mathrm{HR}}$ is converted to ALT (almost lower-triangular) form.
2. The rows of the IR code base matrix $\mathbf{W}_{\mathrm{IR}}$ are determined subsequently with a greedy optimization that uses density evolution with the reciprocal channel approximation, the method described in [13].
3. The parity check matrix of the quasi-cyclic HR code $\mathbf{H}_{\mathrm{HR}}$ and the IR part $\mathbf{H}_{\mathrm{IR}}$ are initialized with CPMs with zero shifts in the positions indicated by ones in $\mathbf{W}_{\mathrm{HR}}$ and $\mathbf{W}_{\mathrm{IR}}$, respectively.
4. The cyclical shift values $s_{r,c}$ in $\mathbf{H}_{\mathrm{HR}}$, together with the matrix $\mathbf{H}_{\mathrm{IR}}$, at positions corresponding to the values of 1 in $\mathbf{W}_{\mathrm{HR}}$ and $\mathbf{W}_{\mathrm{IR}}$ are subsequently modified in the entire structure $\mathbf{H}$ (9) simultaneously, with an algorithmic search method similar to the one proposed in [36,45] for LDPC codes. The algorithm seeks a Tanner graph with minimized existence of small cycles with low external connectivity, measured by the extrinsic message degree (EMD) feature.

The crucial characteristic of the designed algorithm is that after determining base matrices (protographs) for both parts $\mathbf{H}_{\mathrm{HR}}$ and $\mathbf{H}_{\mathrm{IR}}$, with appropriate methods previously known for rate-fixed LDPC codes, the whole graph of the QC-RL-LDPC coding scheme is optimized with a joint algorithm.

8. System Implementation Experimental Studies

Our research has been validated through experimental studies that examine various aspects of the developed QC-RL-LDPC system. These include encoding time; memory utilization of the encoder implemented with a microcontroller; investigation of decoding time and decoding parameter selection; error correction performance; and finally, normalized energy consumption estimation of the developed QC-RL-LDPC system. Due to the specifics of the design objective, which is an energy-efficient IoT uplink transmission scheme design, we investigated relatively short codes of length within $100 \ldots 1000$. We provide the microcontroller implementation results and a performance simulation for the channel with an unknown corruption level, represented by a wide range of SNRs (signal-to-noise ratios).

Several different codes ($\mathbf{H}1, \ldots, \mathbf{H}5$) with diversified block lengths, chunk sizes and minimum rates, as listed in Table 1, were designed and used for this experimental study. As an example, Figure 6 shows a whole parity check matrix $\mathbf{H}$ of the code $\mathbf{H}1$.

The parts of the designed communication system were implemented specifically with an STM32L476 microcontroller, for which we present the implementation results, including a case study of the power consumption and the energy efficiency calculations.

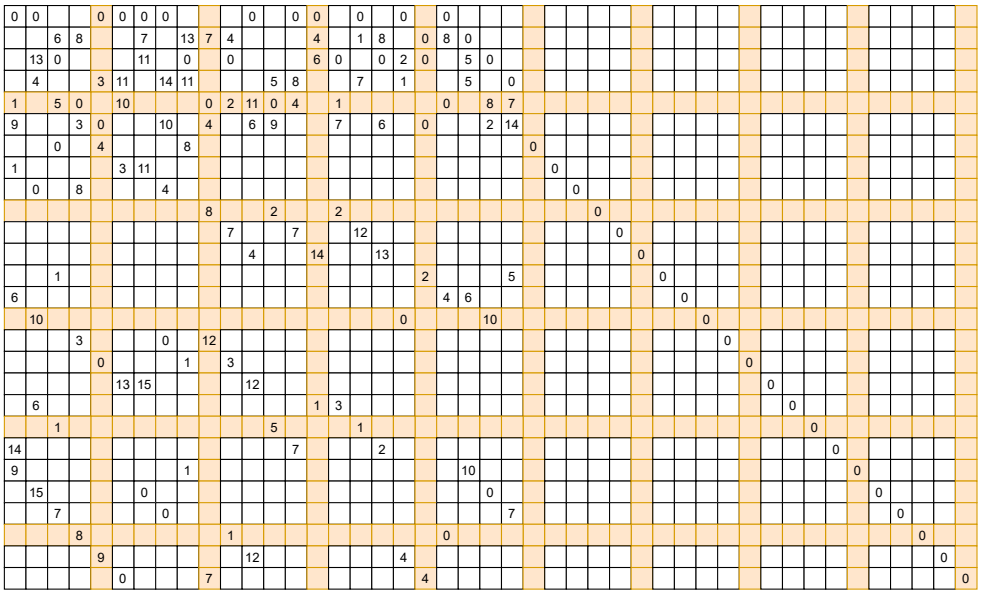


Figure 6. Parity check matrix of a QC-RL-LDPC code **H**1, both HR and IR parts, with submatrix size $P = 16$. The cyclic shift values are provided for nonzero submatrices.

Table 1. Parameters of the QC-RL-LDPC codes used for numerical experiments.

Matrix	K	$N_{\min}$	$N_{\max}$	$R_{\max}$	$R_{\min}$	P
H1	288	384	720	0.75	0.4	16
H2	256	384	896	0.667	0.286	32
H3	144	208	464	0.692	0.310	16
H4	192	240	600	0.8	0.320	8
H5	384	480	1200	0.8	0.320	16

8.1. Encoder Implementation

The encoder was designed with a C language implementation of the encoding algorithm presented in Section 5. Parts of the parity check matrix $\mathbf{H}_{\mathrm{HR}}$ and $\mathbf{H}_{\mathrm{IR}}$ are stored in the microcontroller flash memory in the form of positions (indexes) of nonzero submatrices together with corresponding circular shifts, as illustrated by an example in Figure 6.

We measured the encoding time in the QC-RL-LDPC system with the developed encoder, which was compiled and implemented in the STM32 low-power microcontroller unit. The results are presented in Figure 7. Note that the first part of the encoding (the HR part, shown as chunk-0 in Figure 7) takes a significantly longer time period than the subsequent encoding iterations of the following chunks (the IR part). We also implemented encoders for selected 5G codes, with Base-Graph-1 (BG1, see [17]) and submatrix sizes of 6 and 12, giving codes with information block lengths of 132 and 264, which are similar to our designed codes **H**3, **H**2 and **H**5, respectively. The presented time measurements were then used to assess the energy consumption of the encoder, which is presented in Section 8.4.

The QC-RL-LDPC encoder stores the parity check matrices in memory using static allocation. Its memory utilization is comparable to the efficient QC-LDPC encoder previously presented by us in [36]. The maximum memory requirement for QC-RL codes is related to the maximum code size, with all chunks. The utilized microcontroller is able to allocate the needed memory and process the encoding.

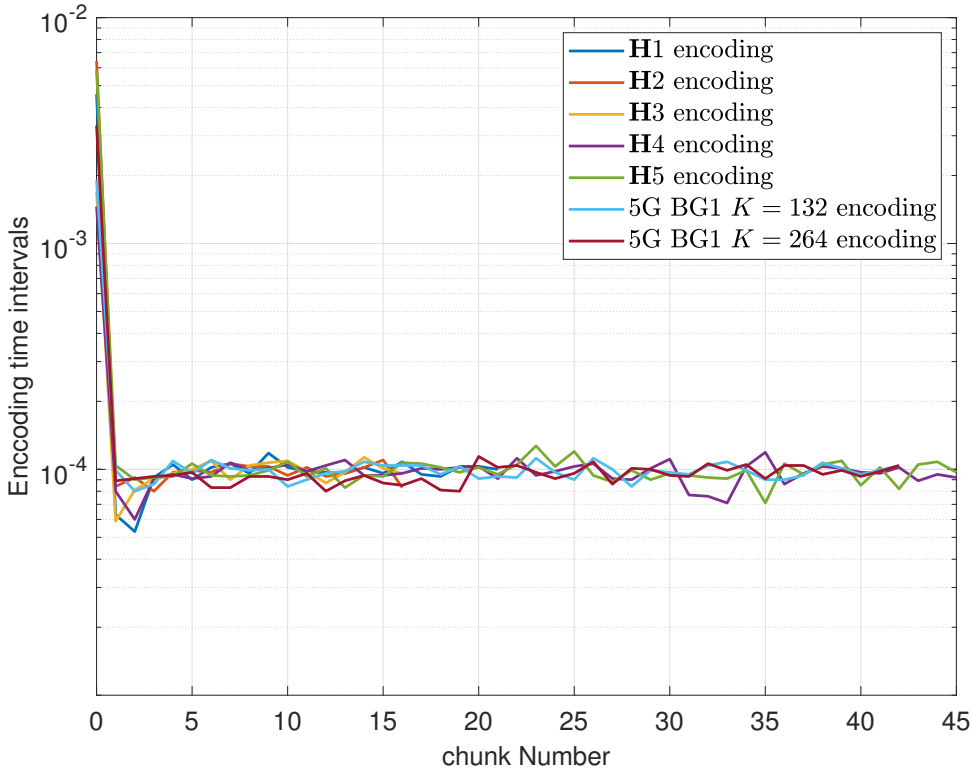


Figure 7. Observed encoding time intervals for initial HR block (chunk No. 0) and IR chunks $1, \ldots, L$.

8.2. Decoder Implementation

The developed Raptor-like decoding algorithm, presented in Section 6, can be implemented on any computing platform utilized in the system gateway (Figure 1), including CPU, GPU, ASIC or FPGA. For the the experimental investigation in this paper, Matlab software implementation on a personal computer was used. However, additionally, we developed the C language implementation for the STM32 microcontroller, the same as that used for the encoder. In Figure 8, we present the decoding time measurements of this implementation.

The microcontroller with the developed decoding algorithm is able to decode information portions efficiently, with lower computing time requirements if the received signal is of high quality (with small number of errors) and higher computing time consumption if the received signal is of low quality. In case of an erroneous received signal, the algorithm adapts and continues decoding until the correct result is achieved. During decoding, subsequent parity checks are used from $\mathbf{H}_{\mathrm{IR}}$. With every additional chunk, the total (cumulative) decoding time increases, as shown in Figure 8. The figure provides averaged measured cumulative decoding time versus the number of chunks required for successful decoding. This means that the first (leftmost) point in every curve presents the time of decoding just the core HR part (with 0 additional chunks), and then the subsequent points present the total decoding times if the number of required chunks increases.

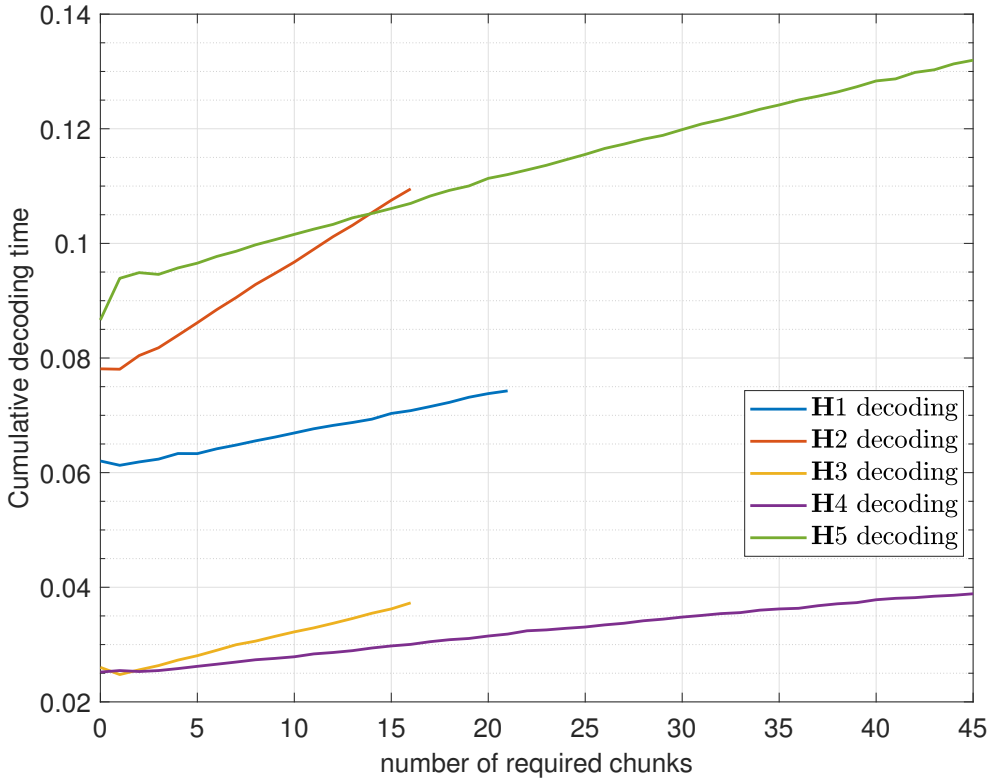


Figure 8. Mean of observed cumulative decoding time.

8.3. System Simulation Results

The experimental verification of error decoding performance that can be achieved with the proposed QC-RL-LDPC implementation was carried out by simulating a system as in Figure 4. In the results presented in this section, we assume a QPSK modulation (two bits per symbol) and an additive white Gaussian noise (AWGN) channel, with a broad range of simulated signal-to-noise ratios (SNRs). Correction performance is evaluated with Monte Carlo simulations, with the number of messages such that at least 100 of them are in error after HR part decoding, for every SNR.

In Figure 9, we provide simulation results in the form of goodput performance, which is defined as the ratio of the number of information bits correctly delivered to the total number of QPSK symbols transmitted, required for successful decoding [11]. Obviously, the higher the SNR, the lower the number of average transmitted chunks needed, which means the goodput is higher. The upper limit of goodput for QPSK is $2R_{\max}$, since if only the initial HR coded vector is transmitted, then $2R_{\max}$ information bits are successfully delivered per every QPSK symbol. The chunk size is assumed to be equal to P, because the encoding is performed in blocks of P bits.

We obtained the parity check matrix $\mathbf{H}1$ by lifting the protograph taken from [13] (Equations (11) and (12) therein), with submatrix size $P = 16$. Since the minimum achievable rate of this code is relatively high, $R_{\min} = 0.4$, the correction performance of the code $\mathbf{H}1$ in the low-SNR regime can be improved by proposing codes with more chunks (or rows in the IR part). To address this, we constructed matrix $\mathbf{H}2$ using the procedure outlined in Section 7, with an increased number of chunks and lower $R_{\min}$. This code can successfully correct errors with SNR well below 0dB, which is shown in Figure 9. Some other matrices that we also designed are $\mathbf{H}3$, which can be used for shorter messages, with information vector length $K = 144$ but slightly underperforming $\mathbf{H}2$; as well as $\mathbf{H}4$ and $\mathbf{H}5$, with an increased initial rate $R_{\max}$, which provides a greater maximum goodput of 1.6.

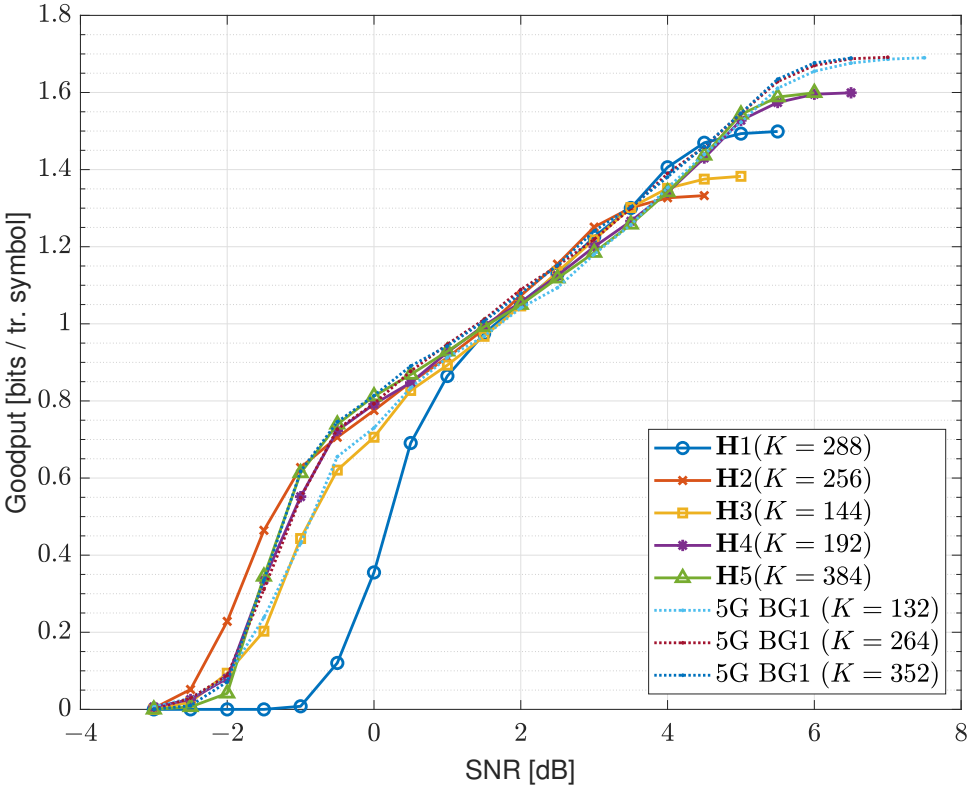


Figure 9. Goodput of the designed codes obtained by Monte Carlo simulation with AWGN channel and QPSK modulation.

As a point of comparison to other methods, we conducted simulations of a rate-adaptive scheme that utilizes 5G codes. Specifically, we examined BG1 with information block lengths of 132, 264 and 352. It can be observed in Figure 9 that the performance is similar to our designed codes **H**3, **H**2 and **H**5, respectively. This proves that the code construction part of this research provides codes that are not worse than state-of-the-art industrial LDPC codes. However, the advantage of the proprietary QC-RL-LDPC design is that the code parameters of HR code rate, information block length and submatrix size are flexible and can be freely chosen, while the 5G code family offers only two HR code configurations (BG1/BG2) and a discrete set of submatrix size/block length pairs.

The designed QC-RL-LDPC codes successfully achieve the objective of the rate self-adaptability processing. As shown in Figure 10, the number of chunks required for error-free reception increases as the SNR decreases. The code should be selected for a specific use case, and the algorithm developed can design codes for diversified needs of information block length K, rates $R_{\max}$, $R_{\min}$ and chunk sizes P.

Another decision that needs to be made in the system design is choosing the iteration limits for decoding, I_{HR} and I_{IR} (as defined in Section 6). This selection involves a trade-off between computational burden and decoding performance. However, as illustrated in Figure 11, which shows results for the same set of received input vectors, decoded with different parameters I_{HR} and I_{IR}, the choice of these parameters does not have a very significant impact on the performance. For the system simulation in this paper, we used $I_{\mathrm{HR}} = 20$ and $I_{\mathrm{IR}} = 7$.

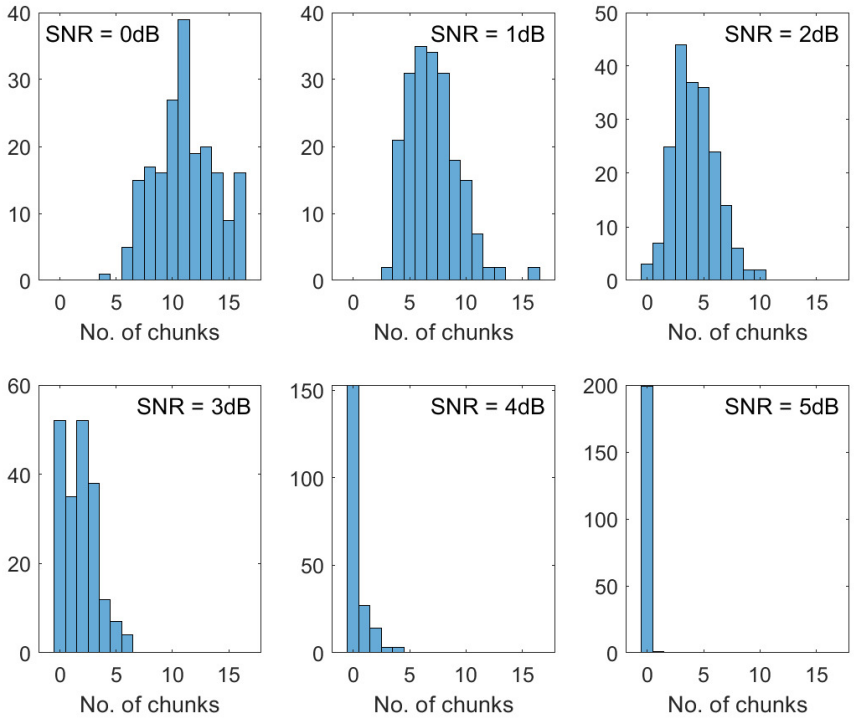


Figure 10. Number of chunks transmitted for different SNR levels for code **H**3 in simulation of transmission of 200 messages.

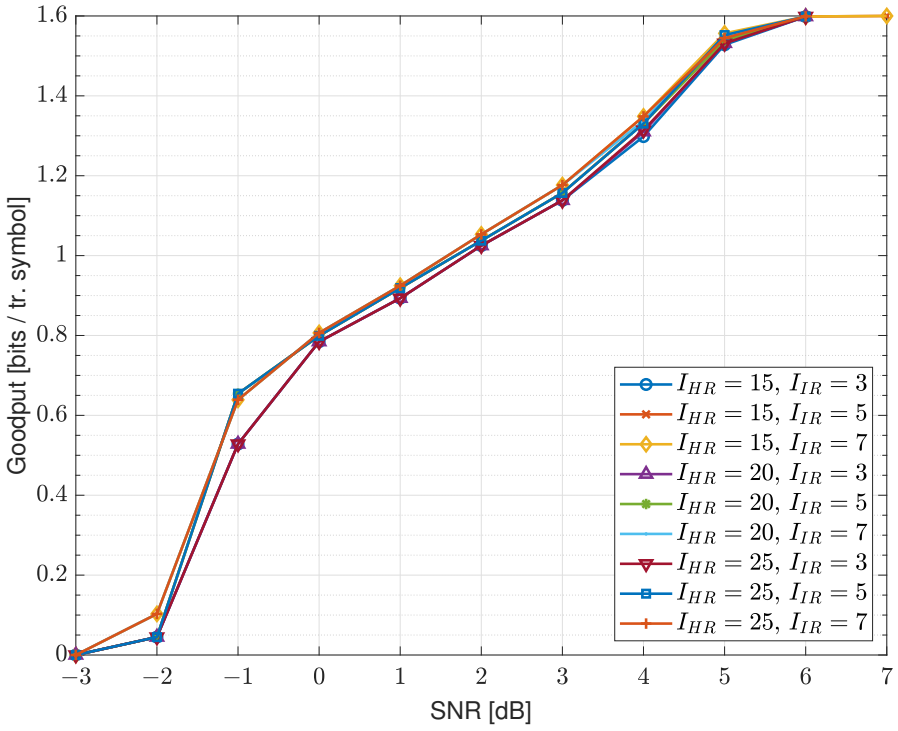


Figure 11. Goodput of the code **H**5 with different decoding iteration limits.

8.4. Energy Consumption Case Study

Based on the experimentally determined transmission performance (goodput) and encoding efficiency (time required for encoding) presented in previous subsections, we aimed to compare the designed system with the fixed-rate coded transmission system, as well as a system employing selected 5G coding configurations, in terms of the IoT transmitter energy efficiency. Since the calculations presented in this section are based on several assumptions, they should be treated as assessments of the energy consumption that can be expected in a real implementation. However, the results illustrate well the gains that can be achieved in the rate-adaptive system.

We assess the portion of the transmitter energy consumption that depends on the utilized coding system, that is, the sum of consumption by the encoding performed in a microcontroller plus consumption by the transmitter module. In the calculations, we have made the following assumptions:

1. The encoder is implemented in the STM32L476 microcontroller, with a supply voltage of 3.3 V and an active mode current draw of 10.7 mA.
2. The radio signal power is 25 mW (equal to the uplink limit in LoRa radio), the RF (radio frequency) module power efficiency is 50% and the transmission rate is 12,000 QPSK symbols per second (also comparable to LoRa limits).
3. The RF transmitter is active only during initial message bits and chunk bits transmissions.

By adding up the radio transmit energy and the encoding energy, factoring in the average number of chunks for the given SNR, we can determine the overall transmitter energy consumption. In the QC-RL-LDPC system, the energy consumption of the transmitter is dependent on the SNR, because with the change in the number of chunks needed, the total encoding time and the radio signal transmission time are also changing. This is illustrated by the evaluation results in Figures 12 and 13 for our designed codes of information block lengths of $K = 144$ and $K = 256$, respectively, as well as codes taken from 5G standard [17] with BG1 and similar block lengths of $K = 132$ and $K = 264$, respectively. The energy consumption provided is normalized and presented as one energy per one correctly decoded information bit.

In contrast, the energy consumption of the fixed-rate encoding is independent of the SNR. In Figures 12 and 13 we also include the energy consumption of the LDPC coding system, calculated with the same assumptions, for the same information block lengths $K = 144$ and $K = 256$, respectively, and the coding rate $R = 0.5$. It can be observed that:

- In the high-SNR regime (higher than about 2 dB), the normalized energy consumption is lower for the QC-RL-LDPC coding than for LDPC fixed-rate coding.
- In the regime just above 0 dB, the LDPC fixed-rate coding is more energy-efficient than QC-RL-LDPC coding. This is the regime for which an LDPC rate-0.5 code would be used under conditions that it was designed for; therefore, its advantage in this regime is as expected.
- For SNRs lower than about 1 dB for for $K = 144$ and lower than about 0 dB for $K = 256$, the fixed-rate LDPC code can no longer correct errors with reasonable BER, while QC-RL-LDPC consumes more energy, but can still be used for transmission and correctly decoded.
- The developed transmission system, with the same IR redundancy scheme, but utilizing the state-of-the-art 5G codes, has very similar energy efficiency to the developed QC-RL-LDPC codes in the case of $K = 256$ (264) block length (Figure 13), and a little worse energy efficiency in the case of $K = 144$ (132) bits (Figure 12).

The justification for this slight difference between the designed QC-RL-LDPC codes and 5G codes is that the code $\mathbf{H}3$ has the submatrix size $P = 16$, while the 5G code has $P = 6$, which means that more smaller chunks need to be encoded for a given parity vector size; the efficient QC encoding in a CPU can compute a chunk with similar speed, for different chunk sizes. This result emphasizes the flexibility of the submatrix size selection that is given by the proprietary QC-RL-LDPC design framework.

It can be inferred that the QC-RL-LDPC system exhibits higher energy efficiency compared to fixed-rate LDPC in high-SNR scenarios. This is due to its reduced transmission rate and encoding time. Additionally, it surpasses LDPC in low-SNR scenarios (below approximately 0 dB) since it can still effectively correct errors while utilizing more parity bit chunks. It should be especially desired to use rate-adaptive coding in situations where the channel SNR is not known in advance. It then improves not only the error correction possibilities, but also the energy efficiency.

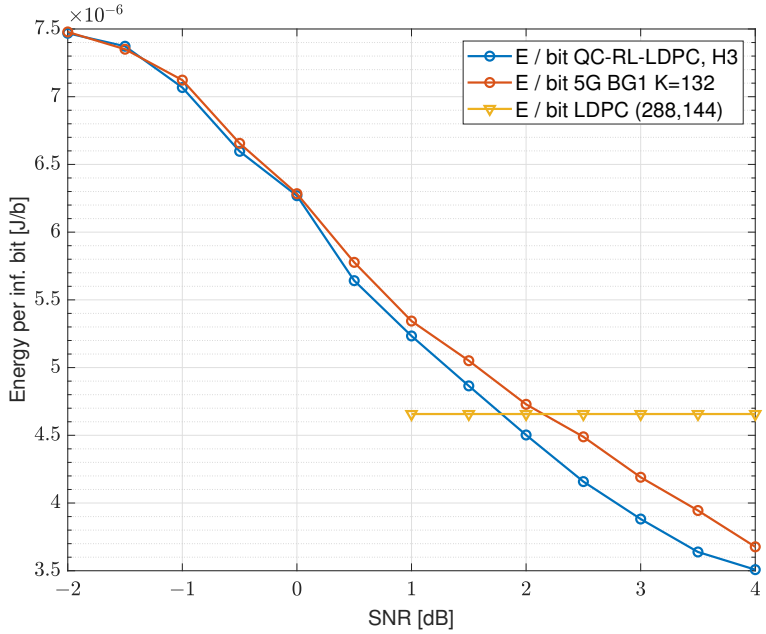


Figure 12. Transmitter energy consumption per information bit—codes of information block length 144b.

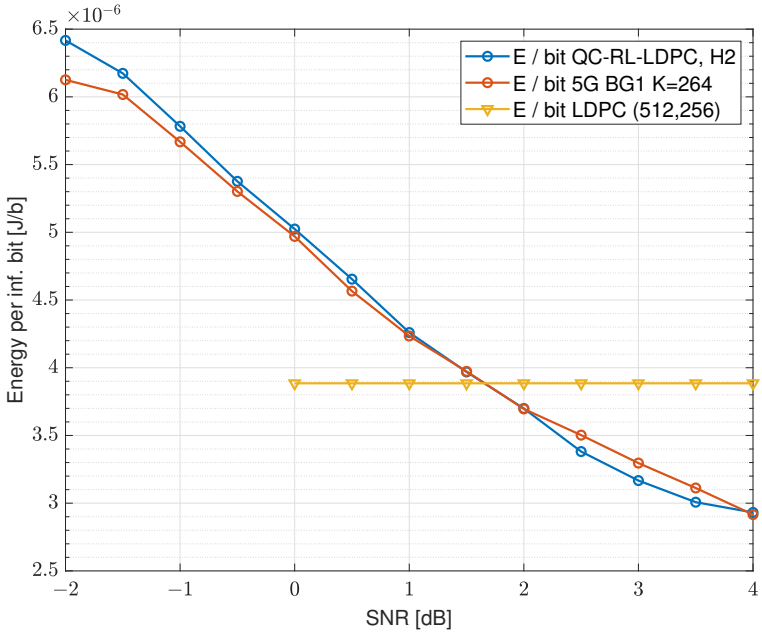


Figure 13. Transmitter energy consumption per information bit—codes of information block length 256b.

9. Concluding Remarks

This paper discusses a complete design procedure and provides implementation results for a QC-RL-LDPC coding system, specifically tailored for IoT uplink transmission. Our focus is on creating an energy-efficient encoding procedure for IoT nodes built with microcontroller devices, without strict throughput requirements, but where the energy efficiency is an important factor. To achieve this, we implemented a short-block-length Raptor-like LDPC scheme that is efficiently encodable. We showed that LDPC coding, commonly used in high-throughput communication systems and advanced protocols, is also applicable to such computationally constrained devices. Our experimental results showed that Raptor-like QC-RL-LDPC coding can be beneficial in terms of energy efficiency, in comparison with fixed-rate LDPC coding, and not worse that state-of-the-art 5G coding. Then, the developed design procedure, including QC-RL-LDPC code design, encoder implementation and the proposed decoding scheme for Raptor-like decoding, can be used in a rate-adaptive uplink transmission, with flexible parameter selection: block length, code rate and submatrix size.

The entire proposed system was implemented and validated for several designed binary QC-RL-LDPC parity check matrices, as well as compared with a similar design based on 5G codes. The comparison shows that the designed codes have at least similar efficiency to the 5G codes, while in some configurations, the flexible submatrix size choice can provide a slight energy efficiency improvement, even over 5G codes. The coding system can still evolve, and future work will be concentrated on the very promising extension of the proposed concepts to the nonbinary (NB) case, that is, the development of NB QC-RL-LDPC codes and the respective codec implementations.

Author Contributions: Conceptualization, J.H. and W.S.; Methodology, W.S.; Software, J.H.; Validation, J.H.; Investigation, J.H.; Supervision, W.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was co-financed by the Ministry of Education and Science of Poland under grant No. DWD/3/7/2019 as well as research subsidy for Silesian University of Technology, No. 02/160/BK_23/0009.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Al-Obaidi, K.M.; Hossain, M.; Alduais, N.A.M.; Al-Duais, H.S.; Omrany, H.; Ghaffarianhoseini, A. A Review of Using IoT for Energy Efficient Buildings and Cities: A Built Environment Perspective. *Energies* **2022**, *15*, 5991. [CrossRef]
2. Shammar, E.A.; Zahary, A.T. The Internet of Things (IoT): A survey of techniques, operating systems, and trends. *Libr. Hi Tech* **2020**, *38*, 5–66. [CrossRef]
3. MacKay, D.J.C. Good Error-Correcting Codes Based on Very Sparse Matrices. *IEEE Trans. Inf. Theory* **1999**, *45*, 399–431. [CrossRef]
4. Berrou, C.; Glavieux, A.; Thitimajshima, P. Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes. In Proceedings of the ICC '93—IEEE International Conference on Communications, Geneva, Switzerland, 23–26 May 1993; pp. 1064–1070.
5. Arikan, E. Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels. *IEEE Trans. Inf. Theory* **2009**, *55*, 3051–3073. [CrossRef]
6. Lin, S.; Costello, D.J., Jr. *Error Control Coding: Fundamentals and Applications*, 2nd ed.; Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 2004.
7. Le, H.D.; Mai, V.V.; Nguyen, C.T.; Pham, A.T. Throughput Analysis of Incremental Redundancy Hybrid ARQ for FSO-Based Satellite Systems. In Proceedings of the 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), Honolulu, HI, USA, 22–25 September 2019; pp. 1–5.
8. Shokrollahi, A. Raptor codes. *IEEE Trans. Inf. Theory* **2006**, *52*, 2551–2567. [CrossRef]
9. Jayasooriya, S.; Shirvanimoghaddam, M.; Ong, L.; Johnson, S.J. Analysis and Design of Raptor Codes Using a Multi-Edge Framework. *IEEE Trans. Commun.* **2017**, *65*, 5123–5136. [CrossRef]
10. Jayasooriya, S.; Shirvanimoghaddam, M.; Ong, L.; Johnson, S.J. Raptor Codes for Higher-Order Modulation Using a Multi-Edge Framework. *IEEE Wirel. Commun. Lett.* **2018**, *7*, 110–113. [CrossRef]
11. Yu, W.; Narayanan, K.; Cheng, J.; Wu, J. Raptor Codes with Descending Order Degrees for AWGN Channels. *IEEE Commun. Lett.* **2020**, *24*, 29–33. [CrossRef]
12. Park, S.I.; Wu, Y.; Kim, H.M.; Hur, N.; Kim, J. Raptor-Like Rate Compatible LDPC Codes and Their Puncturing Performance for the Cloud Transmission System. *IEEE Trans. Broadcast.* **2014**, *60*, 239–245. [CrossRef]

13. Chen, T.Y.; Vakilinia, K.; Divsalar, D.; Wesel, R.D. Protograph-Based Raptor-Like LDPC Codes. *IEEE Trans. Commun.* **2015**, *63*, 1522–1532. [CrossRef]
14. Richardson, T.J.; Urbanke, R.L. Efficient Encoding of Low-Density Parity-Check Codes. *IEEE Trans. Inf. Theory* **2001**, *47*, 638–656. [CrossRef]
15. Fossorier, M.P.C. Quasi-Cyclic Low-Density Parity-Check Codes from Circulant Permutation Matrices. *IEEE Trans. Inf. Theory* **2004**, *50*, 1788–1793. [CrossRef]
16. Xiao, H.; Banihashemi, A.H. Improved Progressive-Edge-Growth (PEG) Construction of Irregular LDPC Codes. *IEEE Commun. Lett.* **2004**, *8*, 715–717. [CrossRef]
17. 3GPP Technical Specification TS 38.212 Version 16.2.0 Release 16. 5G; NR; Multiplexing and Channel Coding, 2020-07. Available online: https://www.etsi.org/deliver/etsi_ts/138200_138299/138212/16.02.00_60/ts_138212v160200p.pdf (accessed on 7 June 2023).
18. Bae, J.H.; Abotabl, A.; Lin, H.P.; Song, K.B.; Lee, J. An overview of channel coding for 5G NR cellular communications. *APSIPA Trans. Signal Inf. Process.* **2019**, *8*, e17. [CrossRef]
19. Gallager, R. Low-density parity-check codes. *IRE Trans. Inf. Theory* **1962**, *8*, 21–28. [CrossRef]
20. MacKay, D.J.C.; Neal, R.M. Near Shannon Limit Performance of Low Density Parity Check Codes. *Electron. Lett.* **1996**, *32*, 1645–1646. [CrossRef]
21. Richardson, T.J.; Shokrollahi, M.A.; Urbanke, R.L. Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes. *IEEE Trans. Inf. Theory* **2001**, *47*, 619–637. [CrossRef]
22. Stark, M.; Lewandowsky, J.; Bauch, G. Information-Bottleneck Decoding of High-Rate Irregular LDPC Codes for Optical Communication Using Message Alignment. *Appl. Sci.* **2018**, *10*, 1884. [CrossRef]
23. Nguyen, T.T.B.; Tan, T.N.; Lee, H. Low-Complexity High-Throughput QC-LDPC Decoder for 5G New Radio Wireless Communication. *Electronics* **2021**, *10*, 516. [CrossRef]
24. Hailes, P.; Xu, L.; Maunder, R.G.; Al-Hashimi, B.M.; Hanzo, L. A Survey of FPGA-Based LDPC Decoders. *IEEE Commun. Surv. Tutorials* **2016**, *18*, 1098–1122. [CrossRef]
25. Li, M.; Derudder, V.; Bertrand, K.; Desset, C.; Bourdoux, A. High-Speed LDPC Decoders Towards 1 Tb/s. *IEEE Trans. Circuits Syst. I* **2021**, *68*, 2224–2233. [CrossRef]
26. Wu, H.; Wang, H. A High Throughput Implementation of QC-LDPC Codes for 5G NR. *IEEE Access* **2019**, *7*, 185373–185384. [CrossRef]
27. Likhobabin, E.A.; Ovinnikov, A.A.; Goriushkin, R.S.; Nikishkin, P.B.; Khokhryakov, E.I. High Throughput FPGA Implementation of LDPC Decoder Architecture for DVB-S2X Standard. In Proceedings of the 2022 International Conference on Information, Control, and Communication Technologies (ICCT), Astrakhan, Russia, 3–7 October 2022.
28. Zhao, Z.; Jiao, X.; Mu, J.; He, Y.C. Randomly Penalized Symbol Flipping Decoding of Non-Binary LDPC Codes. *IEEE Trans. Veh. Technol.* **2021**, *70*, 639–654. [CrossRef]
29. Bocharova, I.E.; Kudryashov, B.D.; Ovsyannikov, E.P.; Skachek, V.; Uustalu, T. Design and Analysis of NB QC-LDPC Codes Over Small Alphabets. *IEEE Trans. Commun.* **2022**, *70*, 2964–2976. [CrossRef]
30. Ferraz, O.; Subramaniyan, S.; Chinthala, R.; Andrade, J.; Cavallaro, J.R.; Nandy, S.K.; Silva, V.; Zhang, X.; Purnaprajna, M.; Falcao, G. A Survey on High-Throughput Non-Binary LDPC Decoders: ASIC, FPGA, and GPU Architectures. *IEEE Commun. Surv. Tutorials* **2022**, *24*, 524–556. [CrossRef]
31. Tanner, R.M. A Recursive Approach to Low Complexity Codes. *IEEE Trans. Inf. Theory* **1981**, *27*, 533–547. [CrossRef]
32. Chen, J.; Dholakia, A.; Eleftheriou, E.; Fossorier, M.; Hu, X.Y. Reduced-complexity decoding of LDPC codes. *IEEE Trans. Commun.* **2005**, *53*, 1288–1299. [CrossRef]
33. Chen, J.; Fossorier, M. Density evolution for two improved BP-Based decoding algorithms of LDPC codes. *IEEE Commun. Lett.* **2002**, *6*, 208–210. [CrossRef]
34. Zhao, J.; Zarkeshvari, F.; Banihashemi, A.H. On Implementation of Min-Sum Algorithm and Its Modifications for Decoding Low-Density Parity-Check (LDPC) Codes. *IEEE Trans. Commun.* **2005**, *53*, 549–554. [CrossRef]
35. Bocharova, I.E.; Kudryashov, B.; Johannesson, R. Searching for Binary and Nonbinary Block and Convolutional LDPC Codes. *IEEE Trans. Inf. Theory* **2016**, *62*, 163–183. [CrossRef]
36. Hyla, J.; Sułek, W.; Izydorczyk, W.; Dziczkowski, L.; Filipowski, W. Efficient LDPC Encoder Design for IoT-Type Devices. *Appl. Sci.* **2022**, *12*, 2558. [CrossRef]
37. Myung, S.; Yang, K.; Kim, J. Quasi-Cyclic LDPC Codes for Fast Encoding. *IEEE Trans. Inf. Theory* **2005**, *51*, 2894–2901. [CrossRef]
38. Fang, Y.; Bi, G.; Guan, Y.L.; Lau, F.C.M. A Survey on Protograph LDPC Codes and Their Applications. *IEEE Commun. Surv. Tutorials* **2015**, *17*, 1989–2016. [CrossRef]
39. Erez, U.; Trott, M.D.; Wornell, G.W. Rateless Coding for Gaussian Channels. *IEEE Trans. Inf. Theory* **2012**, *58*, 530–547. [CrossRef]
40. Hu, X.Y.; Eleftheriou, E.; Arnold, D.M. Regular and Irregular Progressive Edge-Growth Tanner Graphs. *IEEE Trans. Inf. Theory* **2005**, *51*, 386–398. [CrossRef]
41. Wang, Y.; Draper, S.C.; Yedidia, J.S. Hierarchical and High-Girth QC LDPC Codes. *IEEE Trans. Inf. Theory* **2013**, *59*, 4553–4583. [CrossRef]
42. Tasdighi, A.; Banihashemi, A.H.; Sadeghi, M.R. Efficient Search of Girth-Optimal QC-LDPC Codes. *IEEE Trans. Inf. Theory* **2016**, *62*, 1552–1564. [CrossRef]
43. Sulek, W. Quasi-Regular QC-LDPC Codes Derived From Cycle Codes. *IEEE Commun. Lett.* **2016**, *20*, 1705–1708. [CrossRef]

44. Kim, I.; Song, H.Y. Some New Constructions of Girth-8 QC-LDPC Codes for Future GNSS. *IEEE Commun. Lett.* **2021**, *25*, 3780–3784. [CrossRef]
45. Sulek, W. Protograph Based Low-Density Parity-Check Codes Design with Mixed Integer Linear Programming. *IEEE Access* **2019**, *7*, 1424–1438. [CrossRef]

5.5 Publikacja 5: Niebinarne kodowanie LDPC dla systemów Internetu rzeczy.

Jakub HYLA[1 2] Wojciech SUŁEK[1]

Silesian University of Technology (1), TKH Technology Poland Sp. z.o.o. (2)

doi: 10.15199/48.2024.03.44

Niebinarne kodowanie LDPC dla systemów Internetu rzeczy

Streszczenie. *Artykuł dotyczy implementacji niebinarnego kodera LDPC dla urządzeń IoT. W artykule zaproponowano wykorzystanie efektywnego algorytmu kodowania LDPC (Low Density Parity Check) w urządzeniach o mocno ograniczonych zasobach – pamięci oraz mocy obliczeniowej. Wskazano algorytm kodujący uogólniony do kodów niebinarnych nad ciałem $GF(2^q)$ oraz przedstawiono wyniki eksperymentalne implementacji w układzie typowego mikrokontrolera. W publikacji pokazano porównanie zależności czasowych dla kodów nad różnymi rzędami ciał GF. Pokazano wpływ wyboru kodu na potencjalne zużycie energii.*

Abstract. *The article concerns the implementation of a non-binary LDPC encoder for IoT devices. The article proposes the use of an effective LDPC (Low Density Parity Check) coding algorithm in devices with very limited resources – memory and computing power. A coding algorithm generalized to non-binary codes over the $GF(2^q)$ is indicated and experimental results of implementation in a typical microcontroller system are presented. The publication shows a comparison of time dependencies for codes over different orders of GF fields. The impact of code selection on potential energy consumption is also shown.* ***(Non-Binary LDPC coding for Internet of Things)***

Słowa kluczowe: IoT, LDPC, QC-LDPC, kodowanie
Keywords: IoT, LDPC, QC-LDPC, coding

Wprowadzenie

Współczesny postęp technologiczny ma ścisły związek z wysokiej jakości transmisją danych w systemach teleinformatycznych, w tym systemach kategoryzowanych jako Internet rzeczy. Oczekuje się przesyłania coraz większej liczby informacji w coraz krótszym czasie, przy minimalnym zużyciu energii [18]. To z kolei rodzi problemy związane z błędami w przesyle danych, które są nieuniknione, ale mogą być wykrywane i korygowane odpowiednimi technikami [11].

W celu eliminacji wspomnianych błędów, stosuje się techniki kodowania kanałowego. Te metody umożliwiają skuteczne wykrywanie oraz korekcję błędów występujących w danych po stronie odbiorcy informacji. Kodowanie kanałowe wymaga dodawania nadmiarowych bitów kontrolnych do danych, co zwykle odbywa się w blokach o jednakowej długości [17].

Kody LDPC (Low Density Parity Check) stanowią obecnie jedną z najskuteczniejszych metod kodowania blokowego i charakteryzują się znakomitymi zdolnościami korekcyjnymi. Ich historia sięga roku 1962 [13], ale z różnych powodów nie były szeroko wykorzystywane w tamtych czasach [6]. Dopiero w 1999 roku zyskały popularność i obecnie znajdują zastosowanie w różnych systemach, takich jak DVB-S2, DVB-T2 [4], WiFi [1], WiMAX[10] i sieci komórkowe 5G[15, 2, 14, 19].

Projekt badawczy, z którym związany jest niniejszy artykuł, ma na celu pokazanie, że kody LDPC mogą być skutecznie stosowane także w systemach o ograniczonych zasobach sprzętowych, takich jak urządzenia z obszaru Internetu Rzeczy (IoT). W ramach tego projektu stworzono wydajną implementację kodera LDPC dla mikrokontrolerów, a także dekodera, co przedstawiono w publikacjach [9],[7]. Omówiono wyzwania związane ze złożonością obliczeniową kodera LDPC, proponując zmiany w sposobie zapisu macierzy kontrolnych kodów, co znalazło zastosowanie w urządzeniach typu IoT opartych na mikrokontrolerach o ograniczonych możliwościach obliczeniowych [3, 16]. Efektywna implementacja algorytmu kodowania może przynieść znaczne korzyści, szczególnie w przypadku urządzeń IoT zasilanych bateryjne, co pozwala na oszczędność energii [8].

Kontynuacja prac dotyczy wykorzystania niebinarnych (NB) kodów LDPC (Low-Density Parity-Check) w kontekście kodowania bloków informacyjnych. Autorzy zwrócili uwagę na istotne aspekty związane z wydajnością implementacji algorytmu kodowania, sugerując, że skuteczne wdrożenie może przynieść znaczne korzyści. Szczególnie podkreślono potencjalne korzyści dla urządzeń Internetu Rzeczy (IoT) zasilanych bateryjnie, gdzie efektywne kodowanie może istotnie przyczynić się do oszczędności energii. Wnioski z badań wskazują na obiecujące perspektywy zastosowania kodów NB-LDPC w praktyce, zarówno pod względem efektywności energetycznej, jak i ogólnej wydajności systemów komunikacyjnych. W artykule omówiono wyniki związane ze zużyciem pamięci, czasem potrzebnym do zakodowania bloku informacyjnego oraz zużyciem energii.

Podstawowe definicje

Kody LDPC to klasa kodów korekcyjnych, liniowych kodów blokowych, które cechują się niskim stopniem zagęszczenia macierzy kontroli parzystości. Znaczna większość elementów macierzy kodu jest zerami. W związku z tym możliwe jest wykorzystanie efektywnych iteracyjnych algorytmów dekodowania, a jednocześnie kody te charakteryzują się dużymi możliwościami korekcyjnymi [12].

Macierz kontroli parzystości, lub też macierz kontrolna, jest to macierz, która opisuje zależności między bitami informacyjnymi a bitami parzystości (nadmiarowymi) w kodzie. Macierz ta definiuje konkretny kod i w przypadku kodów LDPC jest macierzą rzadką. Wartości różne od zera w macierzy parzystości wskazują, które bity informacyjne są powiązane z którymi bitami parzystości w danym wektorze kodowym.

Graf Tannera, znany również jako graf wiadomości (message-passing graph) lub graf dekodowania, jest strukturą używaną w kontekście dekodowania kodów korekcyjnych, w tym kodów LDPC (Low-Density Parity-Check) oraz kodów BCH (Bose-Chaudhuri-Hocquenghem). Graf Tannera jest używany do reprezentacji i analizy procesu dekodowania w algorytmach korzystających z iteracyjnych propagacji wiadomości (BP – Belief Propagation).

Stopień wierzchołka oznacza liczbę krawędzi incydentnych do danego wierzchołka w grafie, który reprezentuje kod LDPC. Rozkład stopni wierzchołków jest istotnym parametrem w analizie i projektowaniu kodów LDPC, ponieważ wpływa z jednej strony na ich własności korekcyjne, a z drugiej strony – złożoność obliczeniową.

Sprawność kodu LDPC określa stosunek liczby bitów informacyjnych do ogólnej liczby bitów w wektorze kodowym. Jest to jeden z podstawowych parametrów, który można dostosować, aby uzyskać kod LDPC o określonych możliwościach korekcyjnych.

Pojemność informacyjna kanału odnosi się do maksy-

malnej liczby informacji, jaką można przesłać przez dany kanał komunikacyjny przy założeniu pewnego poziomu zakłóceń. Kody LDPC osiągają możliwości zbliżone do tej pojemności, co oznacza, że są one w stanie przesyłać dane z dużą wydajnością przy minimalnych błędach [17].

Operacje arytmetyczne w ciele Galois

Algorytm kodowania wymaga wykonywania operacji dodawania i mnożenia w ciele $GF(2^q)$. W niniejszej sekcji, zostaną omówione te operacje, w kontekście ich implementacji w układzie mikrokontrolera.

×	0	1
0	0	0
1	0	1

Rys. 1. Mnożenie w GF(2)

Ciało Galois o najniższym możliwym rzędzie, $GF(2)$, nazywane jest również ciałem binarnym [5] i składa się z dwóch elementów: 0 i 1. Operacje dodawania i mnożenia w ciele GF(2) są zdefiniowane w sposób - dodawanie jest tożsame do operacji OR , zgodnie z odpowiadającym fragmentem macierzy dodawania z Rys. 4, mnożenie jest tożsame do operacji AND, zgodnie z Rys. 1.

Ciało GF(2) jest używane w różnych dziedzinach, takich jak informatyka, elektronika, kryptografia i teoria kodowania korekcyjnego, ze względu na swoją prostotę i możliwość bezpośredniej reprezentacji danych binarnych.

×	1	2	3
1	1	2	3
2	2	3	1
3	3	1	2

Rys. 2. Mnożenie w GF(2^2)

Ciało $GF(2^q)$ nazywane jest rozszerzonym ciałem Galois, gdzie q jest liczbą całkowitą większą od 1. To ciało składa się z 2^q elementów, a każdy element może być reprezentowany przez wielomian $P(x)$ stopnia $q-1$ ze współczynnikami 0 lub 1. Operacje dodawania i mnożenia w $GF(2^q)$ są nieco bardziej skomplikowane niż w przypadku ciała $GF(2)$. Ciała rozszerzone $GF(2^q)$ są używane w zaawansowanych aplikacjach, takich jak kodowanie korekcyjne Reeda-Solomona, np. w zapisie danych na płytach CD i DVD, a także w algorytmach kryptograficznych.

Operacje dodawania w ciele $GF(2^q)$ dla $q = 1, 2, \ldots$ odbywają się na q-bitowych wartościach binarnych przy użyciu operacji bitowego XOR. Dla $q = 4$ zaprezentowano tablicę dodawania na rys. 4, natomiast dla niższych rzędów niż $q = 4$, należy wykorzystać fragment tej tablicy (odpowiednią lewą górną jej część).

Elementy ciała $GF(2^2)$ można reprezentować przez liczby dziesiętne z zakresu $0 \ldots 3$, lub też słowa 2-bitowe. Mnożenie można także przedstawić w formie tabeli, jak na Rys. 2. Kolumnę i wiersz z wartościami zerowymi pominięto dla czytelności tabeli, ponieważ mnożenie przez zero zawsze daje wynik równy zero.

Elementy ciała $GF(2^3)$ można reprezentować przez liczby z zakresu $0 \ldots 7$, co odpowiada słowom 3-bitowym. Mnożenie w $GF(2^3)$ można zdefiniować jako mnożenie wielomianów, po którym następuje redukcja iloczynu modulo $P(x)$. Co ważne z punktu widzenia implementacji, operacje

×	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7
2	2	4	6	3	1	7	5
3	3	6	5	7	4	1	2
4	4	3	7	6	2	5	1
5	5	1	4	2	7	3	6
6	6	7	1	5	3	2	4
7	7	5	2	1	6	4	3

Rys. 3. Mnożenie w GF(2^3)

mnożenia można przedstawić w postaci tablicy, jak na rys. 3.

Mnożenie w ciele GF to operacja, która może interpretowana jako dzielenie z redukcją (modulo), tzn. dzielenie przy użyciu wielomianu redukującego jako dzielnika.

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	0	3	2	5	4	7	6	9	8	B	A	D	C	F	E
2	2	3	0	1	6	7	4	5	A	B	8	9	E	F	C	D
3	3	2	1	0	7	6	5	4	B	A	9	8	F	E	D	C
4	4	5	6	7	0	1	2	3	C	D	E	F	8	9	A	B
5	5	4	7	6	1	0	3	2	D	C	F	E	9	8	B	A
6	6	7	4	5	2	3	0	1	E	F	C	D	A	B	8	9
7	7	6	5	4	3	2	1	0	F	E	D	C	B	A	9	8
8	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7
9	9	8	B	A	D	C	F	E	1	0	3	2	5	4	7	6
A	A	B	8	9	E	F	C	D	2	3	0	1	6	7	4	5
B	B	A	9	8	F	E	D	C	3	2	1	0	7	6	5	4
C	C	D	E	F	8	9	A	B	4	5	6	7	0	1	2	3
D	D	C	F	E	9	8	B	A	5	4	7	6	1	0	3	2
E	E	F	C	D	A	B	8	9	6	7	4	5	2	3	0	1
F	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0

Rys. 4. Dodawanie w GF(2^4)

Wartości w ciele $GF(2^4)$ to czterobitowe wartości, które obejmują zakres dziesiętny [0..15]. Dodawanie odbywa się na tych czterobitowych wartościach za pomocą operacji logicznego XOR. Na przykład: 5 + 6 = (0101) + (0110) = (0011) = 3 (zaznaczone na rys. 4).

Każde zwiększenie stopnia wielomianu o 1 prowadzi do dwukrotnego zwiększenia każdego z wymiarów tablicy dodawania i tablicy mnożenia, co skutkuje czterokrotnie większym rozmiarem w pamięci. Operacja dodawania może być wykonana za pomocą funkcji XOR bitów w reprezentacji wielomianowej, zatem jest obliczana na bieżąco w czasie trwania algorytmu, bez korzystania z tablicy operacji. Obliczenie wyniku mnożenia w ciele Galois wyższych rzędów wymagałoby bardziej złożonych obliczeń. W związku z tym, jeśli pojemność pamięci programu na to pozwala, lepszym rozwiązaniem może być wykorzystanie tablicy wyników mnożenia, co zostało wykorzystane w niniejszych pracach.

×	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	2	4	6	8	A	C	E	3	1	7	5	B	9	F	D
3	3	6	5	C	F	A	9	B	8	D	E	7	4	1	2
4	4	8	C	3	7	B	F	6	2	E	A	5	1	D	9
5	5	A	F	7	2	D	8	E	B	4	1	9	C	3	6
6	6	C	A	B	D	7	1	5	3	9	F	E	8	2	4
7	7	E	9	F	8	1	6	D	A	3	4	2	5	C	B
8	8	3	B	6	E	5	D	C	4	F	7	A	2	9	1
9	9	1	8	2	B	3	A	4	D	5	C	6	F	7	E
A	A	7	D	E	4	9	3	F	5	8	2	1	B	6	C
B	B	5	E	A	1	F	4	7	C	2	9	D	6	8	3
C	C	B	7	5	9	E	2	A	6	1	D	F	3	4	8
D	D	9	4	1	C	8	5	2	F	B	6	3	E	A	7
E	E	F	1	D	3	2	C	9	7	6	8	4	A	B	5
F	F	D	2	9	6	4	B	1	E	C	3	8	7	5	A

Rys. 5. Mnożenie w GF(2^4)

Autorzy zdecydowali się skorzystać z gotowych tablic mnożących dla branych pod uwagę przypadków, a eksperymenty przeprowadzono dla ciał od $GF(2)$ do $GF(2^6)$.

Model systemu IoT

W ogólnej architekturze rozpatrywanego w artykule systemu IoT (Rys. 6) można wyróżnić różne kategorie (warstwy) urządzeń. W warstwie urządzeń końcowych IoT, istotną klasą są urządzenia zbierające dane pomiarowe w postaci surowej lub przetworzonej, z wykorzystaniem na przykład filtrów cyfrowych. Mogą to być czujniki parametrów środowiskowych, pogodowych, poziomu wody, wilgotności gleby, itp. W niniejszym artykule rozpatrywana jest komunikacja pomiędzy tymi urządzeniami a centralnymi węzłami – bramkami. Bramki agregują dane z wielu podłączonych do nich urządzeń – ich zadaniem jest zbieranie danych od urządzeń IoT i przekazywanie ich do chmury. Komunikacja ta może odbywać się z wykorzystaniem różnych rozwiązań – protokołów sieciowych, jak również różnych mediów – bezprzewodowych, przewodowych, światłowodowych. W założeniu urządzenia końcowe mogą być zasilane bateryjnie, natomiast bramka jest zasilana z sieci energetycznej. W związku z tym, szczególnego znaczenia nabiera efektywność energetyczna łącza w kierunku w górę (od urządzenia końcowego do bramki). Zagadnienia implementacji kodowania korekcyjnego mają tu istotne znaczenie z dwóch powodów: 1) im lepsze własności korekcyjne zastosowanego kodu, tym niższa może być moc nadawanego sygnału, 2) najlepsze własności korekcyjne są osiągane dla kodów o stosunkowo dużej złożoności obliczeniowej implementacji, co wymaga obciążenia układu obliczeniowego urządzenia, także istotnym zużyciem energii. Warto zastosować jak najlepsze kody korekcyjne, które zapewniają znakomite własności korekcyjne, w szczególności dla krótkich bloków, gdyż dane transmitowane w sieciach IoT mają często charakter krótkich bloków. Jednocześnie czas kodowania powinien być możliwie jak najkrótszy dla zapewnienia jak najkrótszego czasu w trybie aktywnym układu procesora.

Biorąc to wszystko pod uwagę, wydaje się, że niebinarne kody LDPC, pomimo ich dużej złożoności algorytmu dekodowania, mogą mieć potencjał zastosowania w łączu w górę w tego typu sieci, ponieważ znane są ich znakomite własności w szczególności dla krótkich i średnich rozmiarów bloków, a złożoność operacji kodowania może zostać utrzymana na poziomie konkurencyjnym do podobnych kodów binarnych. W niniejszym artykule przedstawiamy wyniki implementacji kodera kodów niebinarnych w układzie mikrokontrolera, typowej jednostce centralnej urządzenia końcowego. Przedstawiamy algorytm, implementację oraz eksperymentalnie określone czasy kodowania dla różnych rzędów ciała Galois $GF(q)$. Przeprowadzamy także analizę rozmiaru pamięci RAM wymaganej do pomieszczenia podstawowych zmiennych algorytmu.

Algorytm kodowania niebinarnego LDPC

Wykorzystano efektywny algorytm Richardsona–Urbanke [17], uogólniając go dla kodów nad niebinarnymi ciałami $GF(2^q)$. Wykorzystywana jest macierz kontrolna w postaci prawie-dolnotrójkątnej:

$$\mathbf{H} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{T} \\ \mathbf{C} & \mathbf{D} & \mathbf{E} \end{bmatrix}, \tag{1}$$

gdzie $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, $\mathbf{E}$, $\mathbf{T}$ to podmacierze $\mathbf{H}$ nad ciałem$GF(2^q)$, przy czym $\mathbf{T}$ jest macierzą dolnotrójkątną. Można wykazać, że zakodowanie słowa informacyjnego $\mathbf{u}$ w słowo kodowe $\mathbf{c} = [\mathbf{u}, \mathbf{p}_1, \mathbf{p}_2]$ można wyrazić z wykorzystaniem dwóch członów bloku parzystości:

$$\mathbf{p}_1^T = \mathbf{\Phi}^{-1}(\mathbf{E}\mathbf{T}^{-1}\mathbf{A}\mathbf{u}^T + \mathbf{C}\mathbf{u}^T), \tag{2}$$

$$\mathbf{p}_2^T = \mathbf{T}^{-1}(\mathbf{A}\mathbf{u}^T + \mathbf{B}\mathbf{p}_1^T) \tag{3}$$

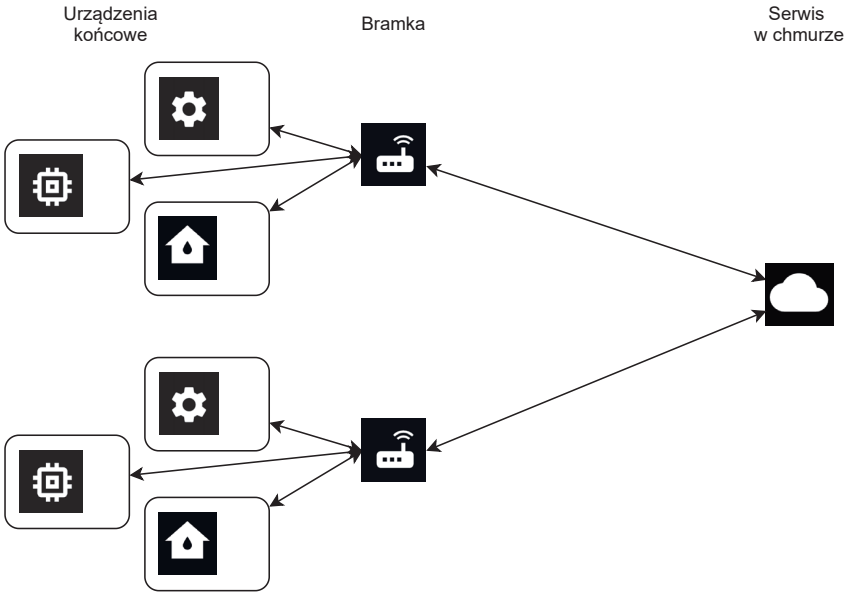
Operacje elementarne, na które można rozłożyć (2)–(3), to mnożenie przez macierz rzadką, dodawanie macierzy rzadkich oraz tzw. operacja wstawiania wstecz [9], do której można sprowadzić mnożenie przez $\mathbf{T}^{-1}$. Jedynie $\mathbf{\Phi} = \mathbf{E}\mathbf{T}^{-1}\mathbf{B} + \mathbf{D}$ jest w ogólności macierzą gęstą. Efektywnie dekodowane kody, które opracowano na cele przedstawionych prac, cechują się tym, że $\mathbf{\Phi}$ jest macierzą jednostkową, $\mathbf{\Phi} = \mathbf{I}$.

Implementacja kodowania NB-LDPC w urządzeniach IoT

Rozwiązanie oparte o binarne kody LDPC, wraz z algorytmem kodowania i jego implementacją, zostało opisane w [9]. Przedstawiono efektywne kodowanie z wykorzystaniem kodów LDPC oraz QC-LDPC, w ciele $GF(2)$. Prace przedstawione w niniejszym artykule są kontynuacją związaną z implementacją kodera dla kodów niebinarnych, definiowanych nad ciałami $GF(2^2)$–$GF(2^6)$.

W implementacji algorytmu wykorzystano reprezentację wektorową współczynników wielomianów reprezentujących elementy ciała. Mnożenie elementów ciała jest wykonywane poprzez indeksację tablicy wartości definiujących mnożenie (rys. 2,3,5 itd.), dla rzędów od 2^2 do 2^6. Macierz kontroli parzystości jest zapisana w pamięci w postaci szeregu indeksów elementów niezerowych oraz szeregu wartości tych elementów, w ciele GF. Należy zauważyć, że zwiększanie rzędu ciała, przy zachowaniu wielkości bloku kodowego (wyrażonego w bitach) oraz sprawności kodu, daje proporcjonalnie mniejsze macierze kontroli parzystości.

Prace eksperymentalne przeprowadzono uwzględniając dwa kryteria optymalizacji, tj. zajętość pamięci RAM przez struktury danych algorytmu oraz czas kodowania. Starano



Rys. 6. Uproszczony model komunikacji z wykorzystaniem kodowania korekcyjnego LDPC w systemie IoT o topologii gwiazdy

się wskazać rząd ciała, który daje najlepszy kompromis pomiędzy złożonością (definiowaną przez dwa wymienione kryteria) oraz możliwościami korekcyjnymi (które rosną wraz z rzędem ciała). Badania skoncentrowano na stosunkowo krótkich blokach danych, nie większych niż 1000 bitów, co jest uzasadnione dla typowo krótkich wiadomości transmitowanych w sieciach IoT.

W artykule podjęto analizę wpływu rzędu ciała Galois na liczbę przeprowadzanych obliczeń oraz zużycie energii, wyrażone skalą zużycia baterii, przy poczynionych założeniach co do sprzętu oraz trybów pracy (aktywny i uśpienia). Przeprowadzono również ocenę wpływu kodów niebinarnych na czas kodowania w warunkach ograniczonych zasobów obliczeniowych. Przedstawione zostaną zalety i wady tychże kodów w kontekście zastosowania w systemach Internetu rzeczy.

Szacowanie zużycia energii oraz dobór baterii

Dobór odpowiedniej baterii dla urządzeń IoT stanowi istotne wyzwanie, mając kluczowy wpływ na zapewnienie deklarowanego czasu pracy zgodnie z określonymi parametrami. Długość działania urządzenia zasilanego bateryjnie jest determinowana przez szereg czynników, takich jak:

- napięcie nominalne,
- pojemność baterii,
- maksymalny, chwilowy i średni prąd pobierany z baterii,
- temperatura baterii w trakcie eksploatacji.

To zagadnienie jest złożone, a jego analiza utrudniona przez wielość parametrów oraz wzajemne zależności między nimi. Na przykład urządzenie monitorujące, które jest wykorzystywane w różnych warunkach pogodowych, np. pełni funkcję monitorowania zapełnienia pojemników na śmieci, w Polsce w roku 2022 funkcjonowałoby w zakresie temperatur od minimalnej na poziomie -18,6 stopni Celsjusza do maksymalnej na poziomie 38,3 stopni Celsjusza.

Temperatura ma bezpośredni wpływ na pojemność baterii i napięcie, istnieje także zależność pomiędzy pobieranym prądem a napięciem oraz pojemnością. Dodatkowo, utrudnieniem w dokładnej ocenie jest pozostała pojemność baterii, wynikająca z charakterystyki napięciowej w trakcie procesu rozładowywania. W praktyce, szacuje się czas pracy na baterii dla stałych warunków oraz przeprowadza testy dla konkretnych punktów pracy, mające na celu symulację określonego scenariusza w skróconym okresie czasu.

Urządzenia będące węzłami IoT, jeśli są zasilane bateryjnie, w celu zwiększenia żywotności baterii powinny działać w schemacie opartym o dwa tryby pracy, zmieniane periodycznie:

- Tryb uśpienia – urządzenie zużywa minimalną liczbę energii, konieczną do podtrzymania zawartości pamięci oraz ewentualnych innych energooszczędnych peryferiów.
- Tryb aktywny – pomiaru i wysyłania danych – urządzenie jest wybudzone i w możliwie krótkim czasie agreguje i wysyła dane do węzła sieci i/lub infrastruktury chmurowej.

W takim schemacie działania, szybkość zużycia baterii podlega wpływowi następujących parametrów urządzenia:

- częstotliwość pomiarów i wysyłania informacji,
- czas pracy w stanie aktywnym,
- prąd zużywany w trybie uśpienia,
- prąd zużywany w trybie aktywnym.

Aby przeprowadzić pomiar i komunikację, należy ustalić oczekiwaną liczbę aktywacji. Przykładowo, urządzenie może dokonywać pomiarów i przesyłać dane co godzinę, co wymaga jednokrotnego wybudzenia w każdej godzinie, a w pozostałej części godziny może pozostawać w trybie uśpienia.

Szacunkowa metoda określenia średniego prądu pobieranego przez układ

Szacunkowe pomiary czasu kodowania zostały wykonane z wykorzystaniem oscyloskopu oraz analizatora stanów logicznych, poprzez pobudzanie wybranych portów wyjściowych jednostki w kluczowych momentach realizacji algorytmu i monitorowaniu zmian stanów logicznych na wyjściach. Przy użyciu przyrządów pomiarowych oraz odpowiednich sond i przystawek można mierzyć zależności czasowe w układzie. Taka metoda może być używana do obserwacji i analizy aktywności układu przy różnych warunkach obciążenia. Możliwe jest śledzenie zmian sygnałów, co umożliwia identyfikację i pomiar czasu trwania poszczególnych trybów (uśpienia / aktywne), a stąd – szacowania średniej wartości pobieranego prądu. Oczywiście realizacja protokołu

		Liczba elementów niezerowych	Rozmiar w pamięci	Rozmiar w pamięci macierzy mnożącej	
Macierz	Rząd ciała Galois		indeks - 2 BAJTY wartość - 2 BAJTY	wartość - 1 BAJT	Suma
HN1	$GF(2)$	2880	11520	4	11524
$R = 0.5$	$GF(2^2)$	1272	5088	16	5104
$N_b = 960$	$GF(2^3)$	816	3264	64	3328
	$\mathbf{GF(2^4)}$	**576**	**2304**	**256**	**2560**
	$GF(2^5)$	440	1760	1024	2784
	$GF(2^6)$	352	1408	4096	5504
	$GF(2^7)$	284	1136	16384	17520
	$GF(2^8)$	240	960	65536	66496
HN2	$GF(2)$	1440	5760	4	5764
$R = 0.5$	$GF(2^2)$	636	2544	16	2560
$N_b = 480$	$GF(2^3)$	404	1616	64	1680
	$\mathbf{GF(2^4)}$	**288**	**1152**	**256**	**1408**
	$GF(2^5)$	220	880	1024	1904
	$GF(2^6)$	176	704	4096	4800
	$GF(2^7)$	142	568	16384	16952
	$GF(2^8)$	120	480	65536	66016
HN3	$GF(2)$	1920	7680	4	7684
$R = 0.75$	$GF(2^2)$	848	3392	16	3408
$N_b = 640$	$GF(2^3)$	540	2160	64	2224
	$\mathbf{GF(2^4)}$	**384**	**1536**	**256**	**1792**
	$GF(2^5)$	292	1168	1024	2192
	$GF(2^6)$	236	944	4096	5040
	$GF(2^7)$	196	784	16384	17168
	$GF(2^8)$	160	640	65536	66176
HN4	$GF(2)$	960	3840	4	3844
$R = 0.75$	$GF(2^2)$	424	1696	16	1712
$N_b \approx 320$	$GF(2^3)$	270	1080	64	1144
	$\mathbf{GF(2^4)}$	**192**	**768**	**256**	**1024**
	$GF(2^5)$	148	592	1024	1616
	$GF(2^6)$	118	472	4096	4568
	$GF(2^7)$	98	392	16384	16776
	$GF(2^8)$	80	320	65536	65856

Tablica 1. Rozmiar macierzy w zależności od rzędu wielomianu ciała Galois.

komunikacji wymaga wykonania szeregu operacji, nie tylko kodowania kanałowego, jednakże w niniejszym artykule skupiono się na analizie czasu operacji kodowania i wynikających z tego konsekwencji.

Celem badania było oszacowanie średniego prądu pobieranego przez koder przy periodycznych zmianach trybu (aktywny / uśpienia), a w konsekwencji szacowanie czasu, w którym układ może być zasilany bez wymiany baterii, przy założonych warunkach (mikrokontroler, bateria, okres pomiędzy transmisjami wiadomości). Metoda pomiarowa umożliwia przeprowadzanie eksperymentów przy użyciu całego spektrum sprzętu, jednak wybów mikrokontrolera nie powinien mieć zasadniczego wpływu na wnioski: porównanie kodów o różnych rzędach ciała GF.

Mikrokontroler	STM32L476RGT6
Zestaw ewaluacyjny	Nucleo L476RZ
Urządzenia pomiarowe	Analizator stanów logicznych Saleae Oscyloskop cyfrowy
Zasilanie	Bateria 3.6V Fanso ER26500M

Tablica 2. Wykaz sprzętu wykorzystanego w eksperymentach

Wyniki

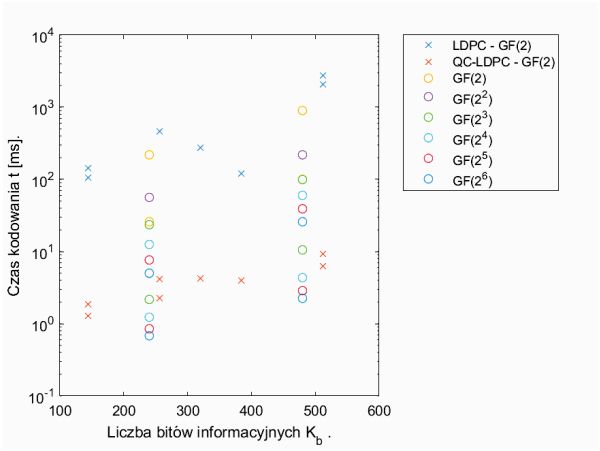
Badania eksperymentalne wykonano z wykorzystaniem sprzętu wymienionego w tab. 2.

Eksperymenty przeprowadzono dla 4 różnych szeregów kodów, oznaczanych dalej HN1...HN4, których parametry podano w tab. 1: rozmiar bloku (N_b – wyrażony w bitach) oraz sprawność kodu (R). W każdym szeregu zawarte są kody o takich samych rozmiarach bloków N_b (lub bardzo zbliżonych, w szczególności dla $q = 7$) i sprawności R, ale skonstruowane nad ciałami Galois o różnych rzędach: $GF(2) \dots GF(2^8)$. Wykorzystano macierze kontrolne kodów binarnych regularnych oraz niebinarnych semiregularnych. Większa sprawność kodu wymaga mniejszej liczby obliczeń przy kodowaniu, natomiast możliwości korekcji błędów są wtedy mniejsze. Długość bloku kodowego w ciele $GF(2^q)$ wynosi dla poszczególnych kodów $N = N_b/q$, a co za tym idzie – liczba elementów niezerowych w macierzy maleje wraz ze wzrostem q.

Każdy element niezerowy jest zapisany w pamięci w reprezentacji macierzy rzadkiej, zawierającej indeks elementu (2 bajty) oraz wartość elementu (2 bajty). Biorąc

Macierz	Rząd	Wykorzystanie baterii w skali roku	Prąd średni [μA]
HN1	$GF(2)$	17,34%	69,49
	$GF(2^2)$	4,22%	33,55
	$GF(2^3)$	1,91%	27,24
	$GF(2^4)$	1,16%	25,17
	$GF(2^5)$	0,75%	24,06
	$GF(2^6)$	0,50%	23,37
HN2	$GF(2)$	4,22%	33,56
	$GF(2^2)$	1,08%	24,97
	$GF(2^3)$	0,45%	23,24
	$GF(2^4)$	0,24%	22,66
	$GF(2^5)$	0,15%	22,40
	$GF(2^6)$	0,10%	22,26
HN3	$GF(2)$	1,93%	27,28
	$GF(2^2)$	0,50%	23,37
	$GF(2^3)$	0,20%	22,56
	$GF(2^4)$	0,08%	22,23
	$GF(2^5)$	0,06%	22,15
	$GF(2^6)$	0,04%	22,12
HN4	$GF(2)$	0,18%	23,37
	$GF(2^2)$	0,10%	22,26
	$GF(2^3)$	0,04%	22,11
	$GF(2^4)$	0,02%	22,06
	$GF(2^5)$	0,02%	22,04
	$GF(2^6)$	0,01%	22,04

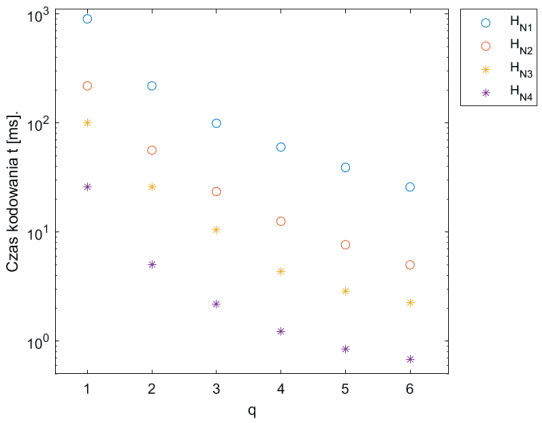
Tablica 3. Wykorzystanie baterii w skali roku dla algorytmów kodujących w ciele Galois rzędu wyższych rzędów $GF(2)$ - $GF(2^6)$. Dane dotyczą samego procesu kodowania przy założeniu jednej wiadomości na godzinę.



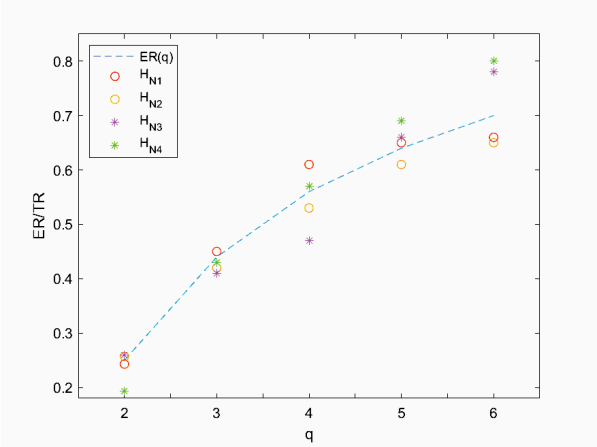
Rys. 7. Wykres zależności czasu kodowania od wielkości macierzy kontroli parzystości dla różnych schematów kodowania - binarne LDPC, QC-LDPC oraz niebinarne LDPC

to pod uwagę, określono i sprawdzono eksperymentalnie rozmiar pamięci potrzebny na zapisanie macierzy kontrolnej, co przedstawiono w tab. 1.

W tabeli podano także rozmiary tablicy wykorzystywanej w operacji mnożenia, która rośnie wraz ze wzrostem q,



Rys. 8. Wykres zależności czasu kodowania od rzędu q ciała Galois kodera dla różnych macierzy kontroli parzystości kodów niebinarnych.



Rys. 9. Teoretyczne i eksperymentalne wyniki stosunków czasów kodowania.

Po zsumowaniu rozmiarów pamięci wymaganej do zapisania macierzy kontrolnej oraz tablicy mnożącej, uzyskano wartości przedstawione w ostatniej kolumnie tabeli. Przedstawione wyniki pozwalają na sformułowanie interesującego wniosku. Otóż zauważono, że można określić taki rząd ciała, dla którego łączna wymagana pamięć ma minimalną wartość. Dla wszystkich szeregów kodów, minimum to przypada na $q = 4$, a zatem wykonane eksperymenty wskazują, że implementacja kodera w mikrokontrolerze będzie się cechowała najmniejszą wymaganą pamięcią dla kodów na ciałem $GF(2^4)$.

Drugą konsekwencją faktu, że rozmiar macierzy $\mathbf{H}$ zmniejsza się wraz ze zwiększaniem rzędu ciała GF, jest zmniejszenie liczby operacji w algorytmie kodującym. Proporcjonalnie mniej jest operacji, ale nad proporcjonalnie wyższym rzędem ciała, co jednakże w implementacji w mikrokontrolerze, z mnożeniem tablicowym, pozwala na znaczące zmniejszenie czasu kodowania. Wyniki dotyczące eksperymentalnych pomiarów czasu przedstawiono na rys. 7 oraz rys. 8, przy czym spektrum badanych kodów obejmuje kilka dodatkowych, oprócz HN1...HN4, badano także kody o innych rozmiarach macierzy.

Poczyniono także spostrzeżenie, że istnieje związek pomiędzy stosunkiem liczby elementów dwóch macierzy o kolejnych wartościach rzędu ciała q, $q+1$, oznaczonym $ER(q)$, a stosunkiem określonych eksperymentalnie czasów

kodowania $TR(q)$ dla tych samych rzędów. Wspomniane wartości zdefiniowano we wzorach (4) oraz (5), a na rys. 9 zobrazowano związek pomiędzy tymi wartościami: linia przerywana obrazuje wyliczone $ER(q)$, natomiast eksperymentalne wyniki stosunków czasów $TR(q)$ są zaprezentowane wykresami punktowymi.

Niech M_q oraz N_q oznaczają – odpowiednio – liczbę wierszy i kolumn kodu nad ciałem $GF(2^q)$, więc dla określonego szeregu kodów o stałej długości słowa informacyjnego i kodowego N_b wyrażonego w bitach łatwo pokazać, że: $M_{q+1}/M_q = N_{q+1}/N_q = q/(q+1)$. Stąd stosunek liczby elementów w macierzach $GF(2^q)$ oraz $GF(2^{q+1})$ wynosi:

$$ER(q) = \frac{M_{q+1}}{M_q} \cdot \frac{N_{q+1}}{N_q} = (\frac{q}{q+1})^2 \tag{4}$$

Punkty na wykresie na rys. 9 oznaczają stosunek czasu kodowania zgodnie ze wzorem 5:

$$TR(q) = \frac{t_{q+1}}{t_q} \tag{5}$$

gdzie t_q jest czasem kodowania kodu nad ciałem $GF(2^q)$.

Eksperymentalnie określono i porównano czasy kodowania w odniesieniu do rozmiaru macierzy kontroli parzystości, a tym samym do rozmiaru słowa informacyjnego. Zauważono zależność związaną ze zmniejszaniem się czasu kodowania wraz ze zmniejszaniem się wielkości macierzy kontroli parzystości. Wyniki czasowe zostały zaprezentowane dla różnych macierzy na rys. 7. Poddane analizie zostały macierze z poprzedniej publikacji autorów [9] oraz porównane z kodami i kodowaniem niebinarnym. Krzyżykami zaznaczone zostały wyniki wcześniejsze [9]: w kolorze niebieskim – wyniki dla różnych macierzy kontroli parzystości dla ciała $GF(2)$ i efektywnego algorytmu kodującego LDPC; kolorem czerwonym – wyniki dla różnych macierzy kontroli parzystości dla ciała $GF(2)$ i efektywnego algorytmu kodującego QC-LDPC. Okręgami zostały oznaczone wyniki dla kolejnych rzędów ciała $GF(2^q)$ dla różnych rozmiarów macierzy. Rząd 6 ciała, $GF(2^6)$ pozwala na uzyskanie najlepszych wyników czasowych i zbliża się do poziomu algorytmu efektywnego kodowania binarnego QC-LDPC. Jednocześnie, kod niebinarny oferuje większe możliwości korekcyjne, co przekłada się na potencjał dodatkowej oszczędności energii w transmisji.

Zużycie energii

W celu zobrazowania poziomu energii zużywanej przez układ kodera implementowanego w mikrokontrolerze oraz porównania poszczególnych kodów nad różnymi rzędami ciała, przeprowadzono szacunki zużycia energii i czasu pracy baterii przy założeniu stałej długości ramki danych i przechodzeniu w stan aktywności co określony czas – wyniki w tej publikacji przedstawiono dla aktywności z okresem jednej godziny. Założono wartości prądów i napięć takie jak w środowisku eksperymentalnym (wymienionym w tab. 2):

- W trybie aktywnym urządzenie zużywa stały prąd na poziomie 190mA, a w trybie uśpienia – 22μA.
- Napięcie baterii to 3.6, natomiast napięcie zasilania układu mikrokontrolera to 3.3V

Energię można wyrazić wzorem:

$$E = U \cdot Q = U \cdot I \cdot t \tag{6}$$

gdzie E to energia elektryczna wyrażona w [J], U to napięcie elektryczne wyrażone w [V], Q to ładunek elektryczny wyrażony w [C], I to natężenie prądu wyrażone w [A] oraz t to czas [s].

Zgromadzone eksperymentalnie dane są sprowadzane do jednostek podstawowych i korzystając ze wzoru 6 określana jest energia. Należy także określić pojemność baterii: wybrano baterię firmy Fanso ER26500M, o napięciu znamionowym 3.6V oraz pojemności 2200mAh, wyrażona w $[As]$ wynosi:

$$E_{bat}[J] = 3.6[V] \cdot 2.2[A] \cdot 3600[s] \tag{7}$$

Z 7 energia baterii wynosi $E_{bat} = 28512[J]$. Jest to energia zgromadzona w baterii i dostępna do wykorzystania. Urządzenie działające w trybie uśpienia z poborem prądu na poziomie 22μA w skali roku pobiera około 2300J energii z baterii co stanowi około 8% jej nominalnej pojemności. Przedstawione rozważania są uproszczone i nie zakładają wpływu prądu obciążenia i temperatury na pojemność oraz efektów starzenia baterii.

Obliczenia dotyczą przypadku, gdy urządzenie wysyła dane raz na godzinę, co daje 8760 wiadomości rocznie. W zależności od wybranej macierzy kontroli parzystości oraz stopnia wielomianu ciała Galois, poziom wykorzystania baterii w skali jednego roku może osiągać od około 0,01% do nawet ponad 50%. Widoczna jest zależność wprost proporcjonalna do czasu kodowania. Zmniejszenie macierzy $\mathbf{H}$ powoduje znaczne zmniejszenie liczby elementarnych operacji wymaganych do zakodowania danych.

Przeliczono wykorzystanie baterii na periodyczne kodowanie wiadomości, a wyniki zgromadzono w formie prądu uśrednionego (łącznie w trybach aktywnym i uśpienia) oraz zużycia baterii w perspektywie rocznej, w tabeli 3. Wpływ kodera na zużycie baterii można zmniejszyć nawet kilkudziesięciokrotnie wykorzystując kody niebinarne wyższych rzędów. Rozwiązanie to przynosi wymierny zysk energetyczny. Należy pamiętać, że czas kodowania jest ułamkiem całej procedury pomiarowo-transmisyjnej na która mogą składać się: wykonanie pomiaru, opracowanie wyniku, przygotowanie wiadomości, zakodowanie wiadomości, wysłanie wiadomości. Nie zmienia to faktu, że należy dążyć do możliwie jak najkrótszego czasu kodowania, aby nie wprowadzać dodatkowych opóźnień w transmisji danych oraz ograniczyć zużycie energii.

Wnioski

W artykule zaprezentowano implementację programową nowoczesnych metod kodowania korekcyjnego NB-LDPC oraz wyniki implementacji w typowych układach wbudowanych – mikrokontrolerach. Przeprowadzone badania otwierają perspektywy wykorzystania tych metod w komercyjnych rozwiązaniach Internetu Rzeczy, co powinno pozwolić na ograniczenie zużycia energii w realizacji protokołów najniższych warstw komunikacyjnych. Zaproponowane algorytmy poszerzają aktualną wiedzę dotyczącą metod kodowania w systemach o ograniczonych mocach obliczeniowych, przy minimalizacji opóźnienia i wymaganej pamięci. Mogą być efektywnie zastosowane w systemach zasilanych bateriami. Zaprezentowano potencjał kodów niebinarnych, które po stronie kodera (nadajnika) mogą mieć lepsze parametry czasowe niż kody binarne, jednocześnie zapewniając lepsze możliwości korekcyjne. Problemem może pozostać kwestia zwiększonej złożoności obliczeniowej dekodera, lecz w systemach z centralnym węzłem o dużych możliwościach

obliczeniowych może to być ograniczeniem nieistotnym, a zaprezentowany potencjał kodów NB wskazuje na dalsze możliwe kierunki badań i rozwoju w tej tematyce.

Autorzy:
- Jakub Hyla
TKH Technology Poland Sp. z.o.o.
email: jakubhyla93@gmail.com,
- Ph.D. Wojciech Sułek
Institute of Automatic Control and Robotics, Electrinics and Telefomunication, Silesian University of Technology ul. Akademicka 2a, 44-100 Gliwice, Poland,

This research was co-financed by the Ministry of Education and Science of Poland under grant No. DWD/3/7/2019.

LITERATURA

[1] Ieee 802.11-2016. ieee standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications, 2016.

[2] Salima Belhadj and Moulay Lakhdar Abdelmounaim. On error correction performance of ldpc and polar codes for the 5g machine type communications. In *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, pages 1–4, 2021.

[3] Vishal A. Dubal and Y. Srinivasa Rao. A low-power high-performance sensor node for internet of things. In *IEEE Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 607–612, Madurai, India, June 2018.

[4] ETSI Standard: EN 302 307 v1.1.1, *Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications*, 2005.

[5] Marc P. C. Fossorier, Miodrag Mihaljevic, and Hideki Imai. Reduced Complexity Iterative Decoding of Low-Density Parity Check Codes Based on Belief Propagation. *IEEE Transactions on Communications*, 47, No. 5:673–680, May 1999.

[6] Robert G. Gallager. Low-Density Parity-Check Codes. *IRE Transactions on Information Theory*, IT-8:21–28, January 1962.

[7] Jakub Hyla and Wojciech Sułek. Dekoder ldpc implementowany w mikrokontrolerze dla systemów internetu rzeczy. *Przedląd Elektrotechniczny*, (04/2023):133, 2023.

[8] Jakub Hyla and Wojciech Sułek. Energy-efficient raptor-like ldpc coding scheme design and implementation for iot communication systems. *Energies*, 16(12), 2023.

[9] Jakub Hyla, Wojciech Sułek, Weronika Izydorczyk, Leszek Dziczkowski, and Wojciech Filipowski. Efficient ldpc encoder design for iot-type devices. *Applied Sciences*, 12(5), 2022.

[10] IEEE Standard: IEEE P802.11n=D10. Draft IEEE Standard for Local Metropolitan Networks – Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC), and Physical Layer (PHY) specifications: Enhancements for Higher Throughput, March 2006.

[11] Shu Lin and Daniel J. Costello, Jr. *Error Control Coding: Fundamentals and Applications, 2nd Edition*. Prentice-Hall, Inc., Upper Saddle River, New Jersey 07458, 2004.

[12] David J. C. MacKay. Good Error-Correcting Codes Based on Very Sparse Matrices. *IEEE Trans. Inf. Theory*, 45:399–431, March 1999.

[13] David J. C. MacKay and Radford M. Neal. Near Shannon Limit Performance of Low Density Parity Check Codes. *Electronics Letters*, 32:1645–1646, August 1996.

[14] Jérémy Nadal and Amer Baghdadi. Parallel and flexible 5g ldpc decoder architecture targeting fpga. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 29(6):1141–1151, 2021.

[15] Vladimir L. Petrović, Dragomir M. El Mezeni, and Andreja Radošević. Flexible 5g new radio ldpc encoder optimized for high hardware usage efficiency. *Electronics*, 10(9), 2021.

[16] C Rajasekaran and K Raguvaran. Microcontroller based reconfigurable iot node. In *IEEE 4th International Conference on Frontiers of Signal Processing*, pages 12–16, Poitiers, France, September 2018.

[17] Thomas J. Richardson and Rüdiger L. Urbanke. Efficient Encoding of Low-Density Parity-Check Codes. *IEEE Trans. Inf. Theory*, 47:638–656, February 2001.

[18] S. Narasimha Swamy and Salomon Raju Kota. An empirical study on system level aspects of internet of things (iot). *IEEE Access*, 8:188082–188134, 2020.

[19] Chance Tarver, Matthew Tonnemacher, Hao Chen, Jianzhong Zhang, and Joseph R. Cavallaro. Gpu-based, ldpc decoding for 5g and beyond. *IEEE Open Journal of Circuits and Systems*, 2:278–290, 2021.

5.6 Publikacja 6 w trakcie recenzji: Short Blocklength Nonbinary Raptor-Like LDPC Coding Systems Design and Simulation.

Short Blocklength Nonbinary Raptor-Like LDPC Coding Systems Design and Simulation

Journal:	*IEEE Access*
Manuscript ID	Access-2024-04215
Manuscript Type:	Regular Manuscript
Date Submitted by the Author:	31-Jan-2024
Complete List of Authors:	Sulek, Wojciech; Politechnika Slaska HYLA, JAKUB; TKH Group NV
Keywords: Please choose keywords carefully as they help us find the most suitable Editor to review:	5G mobile communication, Block codes, Communications technology, Digital communication, Digital modulation, Error correction, Error correction codes, Parity check codes
Subject Category Please select at least two subject categories that best reflect the scope of your manuscript:	Communications technology, Signal processing
Additional Manuscript Keywords:	

AUTHOR RESPONSES TO IEEE ACCESS SUBMISSION QUESTIONS

Author chosen manuscript type:	Research Article
Author explanation /justification for choosing this manuscript type:	This article presents results of a research on a new algorithmic codes design methods.
Author description of how this manuscript fits within the scope of IEEE Access:	The article is a research in the communications technology field.
Author description detailing the unique contribution of the manuscript related to existing literature:	1) we develop a transmission scheme with NB code vectors over GF(q) orders mapped to the 2^X-QAM modulation symbols, where q is not necessarily equal to X; 2) we present an algorithm for constructing efficiently encodable NB-RL-LDPC codes over any GF(q), based on graph optimization and Monte-Carlo performance simulations; 3) we provide experimental results that show the performance improvement of NB-RL-LDPC codes over their binary counterparts, especially 5G codes; 4) we combine the designed codes with different modulation orders and experimentally investigate a system with fall-backing to QPSK modulation during subsequent IR transmissions.

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier xx.xxxx/ACCESS.2024.xxxxxxx

Short Blocklength Nonbinary Raptor-Like LDPC Coding Systems Design and Simulation

JAKUB HYLA[1], WOJCIECH SUŁEK[2], (Member, IEEE)
[1]TKH Technology Poland Sp. z.o.o., Leszno, Poland (e-mail: j.hyla@tkhtechnology.com)
[2]Silesian University of Technology, Gliwice, Poland (e-mail: wojciech.sulek@polsl.pl)

Corresponding author: Wojciech Sułek (e-mail: wojciech.sulek@polsl.pl).

This research was co-financed by the Ministry of Education and Science of Poland under grant No. DWD/3/7/2019 and the research subsidy for Silesian University of Technology, No. 02/160/BK_24/0226.

ABSTRACT This paper explores an efficient rate-adaptive error correction coding scheme with a nonbinary (NB) $\mathrm{GF}(2^q)$ extension of Raptor-like (RL) quasi-cyclic (QC) subclass of Low-Density Parity-Check (LDPC) codes. In the NB RL scheme, the high rate (HR) code vectors are supplemented by the incremental redundancy (IR) symbols, generated by the parity check equations over $\mathrm{GF}(2^q)$. We provide a procedure for constructing short NB QC-RL-LDPC codes with low encoding complexity and optimized graph structures. The proposed method is based on the graph optimization of the HR code and Monte Carlo simulation-supported placement of additional NB parity checks in the IR graph. Numerical simulations validate the approach, demonstrating the effectiveness of the constructed short NB QC-RL-LDPC codes, compared to the 5G standard binary coding with IR. The experimental framework developed further explores NB QC-RL-LDPC coding over $\mathrm{GF}(2^q)$ combined with 2^X-ary digital modulations, enabling high bandwidth utilization. The flexibility of combining LDPC codes over any field with modulations of arbitrary order provides wide adaptability, offering a broad range of possible spectral efficiencies.

INDEX TERMS Error Correction Coding, LDPC, Nonbinary LDPC, Raptor Coding, Modulation mapping

I. INTRODUCTION

Error correction coding (ECC) is an important part of the physical layer of any data transmission system and plays a crucial role in the latency and reliability advancements of modern transmission standards, including the new generations of broadband mobile networks. Binary Low-Density Parity-Check (LDPC) [1] codes are excellent ECC schemes that achieve performance close to the Shannon capacity benchmark. The extension of commonly applied binary LDPC codes (e.g., in 5G [2]) to nonbinary (NB) Galois fields, $\mathrm{GF}(2^q)$ [3], has been demonstrated to offer even improved error correction performance, particularly for short to moderate data block lengths. Nonbinary LDPC (NB LDPC) codes have recently been the subject of active research in various aspects, including code design, reduced complexity decoding algorithms, and implementation issues. This includes studies found in literature, such as [4]–[10].

In the case of a time-varying communication channel, additional improvement in transmission efficiency can be achieved by implementing a well-designed rate-compatible or rateless coding scheme [11], [12]. This is particularly pertinent owing to inherent limitations in traditional fixed-rate coding schemes, tailored for specific channel noise levels or Signal-to-Noise Ratios (SNR). When confronted with an SNR considerably higher than the level for which the fixed-rate code was designed, rate-adaptive coding can benefit by employing higher transmission rates. Conversely, when the SNR is significantly lower than anticipated, the fixed-rate code fails to provide sufficient redundancy. In such instances, rate-adaptive code demonstrates its adaptability by automatically lowering the rate to mitigate the impact of channel distortions.

Rate-compatible codes are at the core of systems with incremental redundancy (IR), which operate in phases. First, the transmitter sends the highest-rate codeword of the necessary size over the channel. If the receiver cannot decode the received noisy codeword, the transmitter sends repeated or additional parity symbols to allow the receiver to attempt another round of decoding, but now with a higher redundancy level. For instance, in 5G New Radio (NR), rate-adaptive coding employs a scheme based on IR transmissions and a family of parity check matrices that can be classified as Quasi-Cyclic (QC) Protograph-Based Raptor-Like (PB-RL) LDPC codes [13]. These codes leverage the structured and flexible nature

of protograph-based codes [14] while incorporating rateless fountain code characteristics inspired by Raptor codes [12], [15]. QC PB-RL codes [16], [17] have parity-check matrices composed of circulant permutation matrices (CPMs), which permit low-complexity decoder implementations and efficient message routing in hardware decoders.

Research on rate compatibility based on binary LDPC coding has reached a high level of maturity. Some of the proposed solutions start with low-rate code and utilize puncturing to obtain higher rates [18], [19]. However, it has been observed that code families obtained using puncturing low-rate code do not perform competitively at high rates [11], [16]. A higher range of well performing code-rates can be achieved using the extending method, which constructs the highest-rate code and then adds redundant symbols as combinations of existing symbols to obtain lower-rate codes [13], [15], [20]–[22].

However, the exploration of generalization to nonbinary Raptor-Like LDPC schemes remains very limited. Therefore, in this study we investigate the potential enhancement in the coding performance of RL-LDPC schemes, especially for short block lengths, by incorporating nonbinary extensions, with coding defined over $\mathrm{GF}(2^q)$ Galois fields with $q > 1$.

We introduce an NB PB-RL-LDPC code construction methodology and conduct an experimental investigation with a rate matching scheme, in which a high-rate NB-LDPC codeword is transmitted directly as the initial block, and additional NB parity symbols are sent in subsequent chunks, only if necessary. This scheme, along with an iterative decoding, is an application of the IR scheme with NB codes. The developed code construction consists of two steps: HR code (HRC) design and IR code (IRC) design, which is a lowering rate extension. We elected to use the Monte-Carlo simulation approach in IRC design instead of threshold optimization as previously proposed for binary codes (e.g. in [13]), because the threshold characterizes the asymptotic performance of the ensemble of codes, whereas our approach is aimed at QC codes for concrete short blocklength realizations, which can be directly obtained, verified and compared with the developed simulation framework. The proposed approach is compact and easy to apply.

We also note that, given the retransmissions are required in an adverse channel conditions, it may prove advantageous to reduce the modulation order for subsequent chunk transmissions. We combine coding over $\mathrm{GF}(2^q)$ with the QAM transmission and detection model. Our developed system allows for varied QAM modulation orders, also deviating from the traditional alignment with $\mathrm{GF}(2^q)$ order, to optimize the overall system performance. We found that transitioning the system to a lower-order, such as QPSK, during transmissions of IR symbols proved to be beneficial in the experiments we conducted.

The research presented in this paper can be summarized as follows: 1) we develop a transmission scheme with NB code vectors over $\mathrm{GF}(2^q)$ orders mapped to the 2^X-QAM modulation symbols, where q is not necessarily equal to X; 2) we present an algorithm for constructing efficiently encodable NB-RL-LDPC codes over any $\mathrm{GF}(2^q)$, based on graph optimization and Monte-Carlo performance simulations; 3) we provide experimental results that show the performance improvement of NB-RL-LDPC codes over their binary counterparts; we use 5G codes as well as IR schemes based on Polar codes as a reference; 4) we combine the designed codes with different modulation orders and experimentally investigate a system with fall-backing to QPSK modulation during subsequent IR transmissions.

II. NB-LDPC CODING PRELIMINARIES

Nonbinary LDPC codes are a class of linear block error correcting codes defined over the Galois Field, $\mathrm{GF}(2^q)$ with $q > 1$. The encoding process for an (N, K) code of rate $R = K/N$ adds $M = N - K$ redundant elements to the information vector $\mathbf{u} = \{u_1, u_2, \ldots, u_K\}$ which forms the code vector (codeword) $\mathbf{c} = \{c_1, c_2, \ldots, c_N\}$, where the vector elements are within the field $\mathrm{GF}(2^q)$.

The NB-LDPC code is specified by its low-density parity check matrix $\mathbf{H}_{M\times N}$ with $\mathrm{GF}(2^q)$ entries. In the decoder, a row vector $\hat{\mathbf{c}}$ of length N is recognized as the correct codeword if and only if it satisfies the parity check equation $\mathbf{H}\hat{\mathbf{c}}^T = \mathbf{0}_{M\times 1}$ in $\mathrm{GF}(2^q)$ field arithmetic. Every row of $\mathbf{H}$ defines one parity check in $\mathrm{GF}(2^q)$. When the check equation is not satisfied, error correction decoding can be executed using an iterative belief propagation (BP) algorithm [1], [9].

A. BOCK-STRUCTURED PARITY CHECK MATRICES

Similar to the QC-LDPC binary codes, deeply studied [23]–[25] and used in several industrial standards [2], [26], the block structure of the NB parity check matrix can support the development of efficient hardware implementations for both encoding and decoding. The parity check matrix $\mathbf{H}$ of a structured code consists of a set of square circulant permutation matrices (CPM) and square zero matrices. A CPM $\mathbf{P}^s$ is a $P \times P$ matrix obtained by cyclically shifting rows of the identity matrix $\mathbf{I}_{P\times P}$ to the right by $(s \mod P)$ positions.

For the general structured case, the matrix $\mathbf{H}$ can be expressed as follows:

$$\mathbf{H} = \begin{bmatrix} z_{1,1}\mathbf{P}^{s_{1,1}} & z_{1,2}\mathbf{P}^{s_{1,2}} & \cdots & z_{1,C}\mathbf{P}^{s_{1,C}} \\ z_{2,1}\mathbf{P}^{s_{2,1}} & z_{2,2}\mathbf{P}^{s_{2,2}} & \cdots & z_{2,C}\mathbf{P}^{s_{2,C}} \\ \vdots & \vdots & \ddots & \vdots \\ z_{R,1}\mathbf{P}^{s_{R,1}} & z_{R,2}\mathbf{P}^{s_{R,2}} & \cdots & z_{R,C}\mathbf{P}^{s_{R,C}} \end{bmatrix}, \quad (1)$$

where $s_{i,j} \in \{0, 1, \ldots, P-1\}$ define the circulant shift values (CSV) and $z_{i,j}$ indicates the positions of nonzero submatrices within $\mathbf{H}$. For nonbinary codes, $z_{i,j} \in \mathrm{GF}(2^q)$ also indicates the $\mathrm{GF}(2^q)$ coefficients of the parity checks.

Matrix $\mathbf{H}$ in Equation (1) has a size of $RP \times CP$ and defines an NB structured code over $\mathrm{GF}(2^q)$. The coefficients $z_{r,c}$ can be collected jointly, forming the code base matrix $\mathbf{Z}$, which is of size $R \times C$. Similarly, the $s_{r,c}$ form a matrix of CSVs, $\mathbf{S}_{R\times C}$. Every structured code is fully defined by the base matrix $\mathbf{Z}$, the submatrix size P, and the CSV matrix $\mathbf{S}$.

B. CODE GRAPH REPRESENTATION

A different approach for representing the code involves a bipartite Tanner graph [1]. This graph comprises variable nodes corresponding to the data symbols associated with the columns of $\mathbf{H}$, check nodes corresponding to parity checks associated with the rows of $\mathbf{H}$, and edges corresponding to the positions of nonzero entries within $\mathbf{H}$. Similarly, a base graph representation of the base matrix $\mathbf{Z}$ can be defined.

The performance of short blocklength code is directly dependent on its key graph structural properties. In particular, the presence of short cycles within a code graph substantially influences the performance of the corresponding LDPC code. For practical codes of short to medium lengths, the independence assumption required for BP decoding is not met if cycles exist. With this assumption unmet, the message passing in BP decoding is suboptimal, although it would suffice for practical implementations.

Therefore, a search method for high performance finite length codes involves reducing the existence and influence of short cycles in the code graph. The issues of code construction by graph optimization have been thoroughly considered in the rich literature on the subject [6], [8], [27], [28]. A well known and effective algorithm is the Progressive Edge Growth (PEG) [29], [30], which is a greedy edge placement construction method that is applied to an initially edge-empty graph to obtain a Tanner graph representation of an LDPC code. PEG inserts edges into the graph such that, when a cycle inevitably arises, its length takes the maximum possible value within the current graph structure. The PEG algorithm is versatile and convenient, but does not support direct QC-LDPC code design.

In addition to the structural properties of the graph, like the existence of the shortest cycles, the error correction capabilities of an LDPC code are dependent on the node degrees in the graph. The node degree distribution of a code graph plays an important role as a code parameter. This parameter represents the distribution of the column weights in matrix $\mathbf{H}$, which is given by $\boldsymbol{\lambda} = [\lambda_2, \lambda_3, \lambda_4, \ldots]$, where λ_i denotes the number of columns of weight i in $\mathbf{H}$. If only one element in $\boldsymbol{\lambda}$ is nonzero, the code is called regular, otherwise it is called irregular.

A typical LDPC code construction algorithm starts by determining the distribution of column weights $\boldsymbol{\lambda}$. It is known that for short NB codes over higher-order $\mathrm{GF}(2^q)$ fields, well performing degree distributions contain two nonzero elements: λ_2 and λ_3 [31], [32]. Such codes include only weigth-2 and weigth-3 columns in $\mathbf{H}$ and are called quasi-regular or semiregular codes.

C. DUAL DIAGONAL STRUCTURE FOR EFFICIENT ENCODING

Structured matrix $\mathbf{H}$ is suitable for the implementation of fast semiparallel decoders. In contrast, efficient LDPC encoding, using the method introduced by Richardson and Urbanke [33], requires the parity check matrix to take an approximately lower triangular form. For a QC-LDPC code with a submatrix size $P \times P$, this form can be represented as

$$\mathbf{H} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{T} \\ \mathbf{C} & \mathbf{D} & \mathbf{E} \end{bmatrix}, \tag{2}$$

where $\mathbf{T}$ is a lower triangular matrix of size $(R-g)P \times (R-g)P$, where g is a positive integer (the so-called gap).

The original Richardson-Urbanke encoding [33] requires multiple sparse matrix multiplications and a single multiplication by the dense matrix $\boldsymbol{\Phi} = \mathbf{E}\mathbf{T}^{-1}\mathbf{B} + \mathbf{D}$. Matrix $\boldsymbol{\Phi}$ is a dense matrix with size $gP \times gP$. However, as was later shown by Myung et al. [34], if the $\mathbf{H}$ of a QC-LDPC code is dual-diagonal, then $\boldsymbol{\Phi}$ can be converted to a $P \times P$ identity matrix. This dual-diagonal construction does not require multiplication by $\boldsymbol{\Phi}$ during the encoding process, which is utilized in certain industrial standards, such as WiMAX [26] and 5G [2], and can be generalized to NB codes. The code construction algorithm developed in this study provides dual-diagonal codes with an identity $\boldsymbol{\Phi}$.

D. RAPTOR-LIKE (RL) NB-LDPC CODING

Unlike block coding, which maintains a fixed code rate, rateless coding is an error correction method that generates a controlled stream of redundant bits from a given information vector. It typically employs Incremental Redundancy (IR) decoding at the receiver's end. This concept with binary LDPC codes is applied, for example, in the physical layer of the 5G mobile communication standard [2]. The 5G NR LDPC coding consists of a concatenation of a binary LDPC High-Rate (HR) code and low-density generator matrix (LDGM) code [35] for IR. This coding scheme is commonly referred to as Raptor-Like (RL) LDPC coding [13]. In the LDGM (IR) part of the graph, additional variable nodes are the extension degree-one variable nodes, each connected to a unique check node. These check nodes, in turn, have other variable node neighbors selected from the core HR code.

This research tries to investigate in an experimental setup, whether the binary coding of 5G can be outperformed by using nonbinary LDPC coding scheme, with IR based on nonbinary LDGM. The NB RL-LDPC code is defined as a concatenation of an NB High-Rate Code (HRC) and an NB incremental redundancy code (IRC). HRC is a structured NB-LDPC code with a parity check matrix $\mathbf{H}_{\mathrm{HR}}$ of size $M \times N$ following the form presented in (1). IRC defines additional parity check symbols in the IR codeword by means of a generator matrix: $\mathbf{G} = [\mathbf{I}, \mathbf{H}_{\mathrm{IR}}^T]$ over $\mathrm{GF}(2^q)$.

In our research, the matrices that define the coding scheme $\mathbf{H}_{\mathrm{HR}}$ and $\mathbf{H}_{\mathrm{IR}}$ are both built of CPMs with nonbinary $\mathrm{GF}(2^q)$ coefficients, as in (1). The size of the matrix $\mathbf{H}_{\mathrm{IR}}$ is $LP \times N$, which means that it has L nonbinary macrorows of $P \times P$ circulants, giving a total of LP parity checks in the IR scheme. The joint parity check matrix of both parts of NB-RL-LDPC coding can be expressed as follows:

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{\mathrm{HR}} & \mathbf{0} \\ \mathbf{H}_{\mathrm{IR}} & \mathbf{I} \end{bmatrix} \tag{3}$$

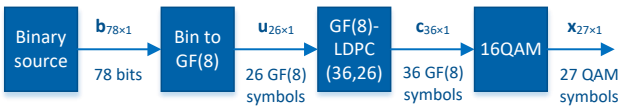


FIGURE 1. Example of short (36,26) nonbinary GF(8) LDPC HR code block combined with 16-QAM modulation.

where $\mathbf{I}$ is an identity matrix and $\mathbf{H}_{\mathrm{HR}}$ follows an efficiently encodable structure, as in (2), with dual-diagonal $\mathbf{T}$ and $\mathbf{0}$ is an all-zero matrix.

III. THE DEVELOPED TRANSMISSION SYSTEM MODEL

A. NONBINARY LDPC CODES COMBINED WITH QAM MODULATION OF ARBITRARY ORDER

In our communication system model, the $\mathrm{GF}(2^q)$ coding, $q > 1$, is combined with 2^X-ary digital modulations. The modulation scheme can be BPSK for $X = 1$, QPSK for $X = 2$ or 2^X-QAM with a square constellation for $X = 4, 6, 8, \ldots$. We do not constrain our investigation to the simplest and most often considered case of equal modulation and code order ($q = X$). This means that in our model, an LDPC code over any field can be combined with a modulation of any order, which provides spectral efficiency flexibility and a broad range of spectral efficiencies possible.

To clarify this, an example of $\mathrm{GF}(8)$ combined with 16-QAM is shown in Fig. 1. In this case, the conversion of 36 $\mathrm{GF}(8)$ code symbols into 16-QAM modulation symbols is performed by using a mapping from a 4-tuples of $\mathrm{GF}(8)$ symbols into a 3-tuples of 16-QAM symbols, which is illustrated by the constellations in Fig. 2. Every value of Symbol-1 in the $\mathrm{GF}(8)$ tuple is represented by a pair of 16-QAM values, as shown in the Figure, where the eight possible values of $\mathrm{GF}(8)$ are denoted by $\{0, 1, \ldots, 7\}$, which are decimal representations of the GF elements polynomials. The 2 values in every pair are then distinguished by Symbol-2 membership in either a $\{0, 2, 4, 6\}$ or a $\{1, 3, 5, 7\}$ subset. Each element of both subsets is then mapped to one of the possible subsets of the 2nd modulation symbol in a tuple. Therefore, $\mathrm{GF}(8)$ Symbol-2 is mapped jointly to the 1st and 2nd symbols in a 16-QAM tuple. Similarly, Symbol-3 is mapped to the 2nd and 3rd symbols in a 16-QAM tuple and Symbol-4 is mapped to the 3rd symbol in a 16-QAM 3-tuple, as illustrated in Fig. 2.

Obviously, many other mappings are possible, but the one shown in Fig. 2 results from bit-level Gray encoding [36], which means that the neighboring possibilities on the constellation diagrams are distinguished by just one bit in the binary representation of the tuples. This is exemplified by the binary representation of symbol-1 shown in Fig. 2.

In general, the NB-LDPC codeword is partitioned into A-tuples of $\mathrm{GF}(q = 2^p)$ symbols that are converted into B-tupples of 2^X-QAM symbols, where:

$$A = \frac{X}{\gcd(p, X)}, \qquad B = \frac{p}{\gcd(p, X)}, \tag{4}$$

where gcd denotes the greatest common divisor.

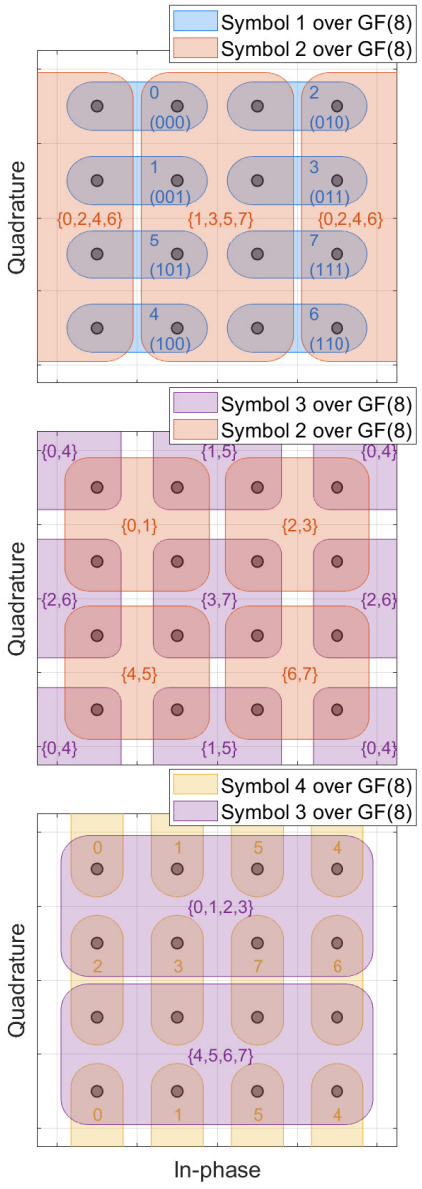


FIGURE 2. Code symbols over GF(8) mapping to 16-QAM constellation: a sequence of 4 GF(8) symbols is mapped to a sequence of 3 16-QAM symbols, represented by the 3 presented constellations.

The ratio of the number of system input (information) bits to the number of output (modulation) symbols can be expressed as:

$$\gamma = (Kp) \,/\, (pN/X) = R \cdot X \tag{5}$$

It reflects the number of information bits carried by a single modulation symbol. With the common assumption of signal shaping with 1 Hz of spectrum needed per every 1 symb/sec transmission rate, γ is equal to the system spectral efficiency expressed in $\left[\frac{\mathrm{b/s}}{\mathrm{Hz}}\right]$.

B. BLOCK DIAGRAM OF THE TRANSMISSION SYSTEM MODEL

The model of the investigated communication system employing NB IR coded QAM modulation is presented in Fig. 3. On the transmitter side, the encoded HR codeword $\mathbf{c}$ of size $N \times 1$ over $\mathrm{GF}(2^q)$ is converted to a vector of 2^X-QAM symbols $\mathbf{x}$ of size $(qN/X) \times 1$, according to the mapping method discussed in Section III-A. After the initial transmission of $\mathbf{c}$, the transmitter generates and subsequently transmits IR parity bits in vectors $\mathbf{p}_{\mathrm{IR}}(l)$, $l = 1, \ldots, L$, until an acknowledgment (ACK) of successful decoding is received. The IR transmissions use $2^{X_{\mathrm{IR}}}$-QAM modulation, not necessarily the same as the initial 2^X-QAM used for the HR part. The IR subvectors $\mathbf{p}_{\mathrm{IR}}(l)$ are of length P as a result of QC encoding, however several subvectors can be combined into a single transmitted chunk, depending on the specific requirements and implementation. The acknowledgements (ACKs) in the feedback channel controls the retransmissions. Detection and decoding operations on the receiver side are discussed in the next section.

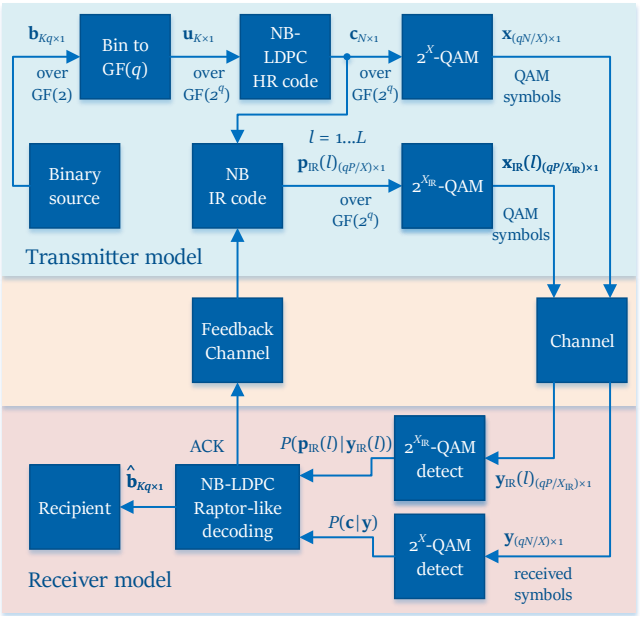


FIGURE 3. Transmission system model with NB RL-LDPC coding.

C. DETECTION AND DECODING OF RL-NB-LDPC CODING WITH QAM

At the receiver, the modulated symbols x_i, $i = 1, \ldots, qN/X$ are corrupted by channel impairments, which for a baseband AWGN channel model can be expressed as:

$$y_i = x_i + \zeta_i = x_i + (\zeta_{i,re} + j\zeta_{i,im}) \tag{6}$$

where $\zeta_{i,re}$ and $\zeta_{i,im}$ are two independent Gaussian noise samples with variance σ^2. The received values $\mathbf{y} = [y_1, \ldots, y_{qN/X}]$ are used to compute the probabilities $P(c_n = a|\mathbf{y})$ that initialize the decoding process, for $n = 1, \ldots, N$ and

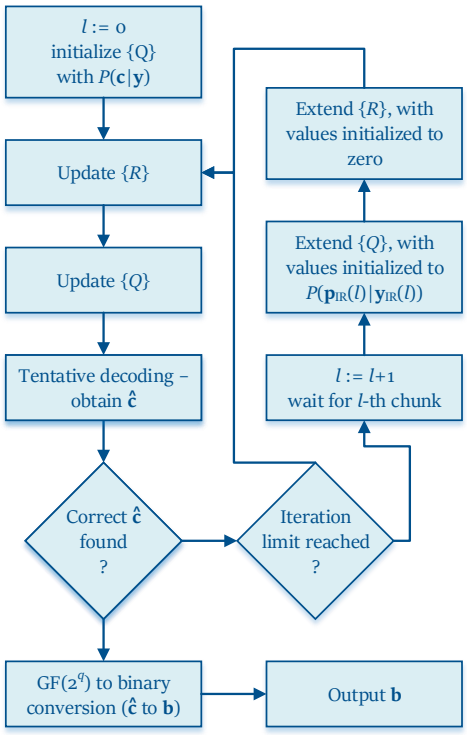


FIGURE 4. Decoding algorithm outline.

for every $a \in \mathrm{GF}(2^q)$. Since $P(x_i|y_i) \propto \exp\left(-\frac{|y_i - x_i|^2}{\sigma^2}\right)$, then the expression for computing the decoder input soft decisions can be given as:

$$P(c_n = a|\mathbf{y}) = \alpha_n \prod_{i=I_1(c_n)}^{I_2(c_n)} \left(\beta_i \sum_{j \in \Psi_{i,n}(a)} \exp\left(-\frac{|y_i - x_j|^2}{\sigma^2}\right) \right) \tag{7}$$

where $I_1(c_n)$ and $I_2(c_n)$ are indexes of the first $y_{I_1(c_n)}$ and last modulation symbol $y_{I_2(c_n)}$ involved in c_n transmission; $\Psi_{i,n}(a)$ is the set of symbols representing $c_n = a$ in y_i; α_n is the normalization factor chosen such that $\sum_{a \in \mathrm{GF}(2^q)} P(c_n = a|\mathbf{y}) = 1$; and β_i is the normalization factor chosen such that:

$$\sum_{a \in \mathrm{GF}(2^q)} \sum_{j \in \Psi_{i,n}(a)} \exp\left(-\frac{|y_i - x_j|^2}{\sigma^2}\right) = 1 \tag{8}$$

which ensures the probability distribution for ith QAM symbol is correctly normalized. Collectively, the matrix of all soft decisions $P(c_n = a|\mathbf{y})$ for $n = 1, \ldots, N$ and $a \in \mathrm{GF}(2^q)$, with a slight abuse of notation, is denoted by $P(\mathbf{c}|\mathbf{y})$ in Figs. 3–4.

Similarly, soft decisions for the lth chunk of IR transmissions are calculated and stored in a matrix that is denoted $P(\mathbf{p}_{\mathrm{IR}}(l)|\mathbf{y}_{\mathrm{IR}}(l))$.

The decoding algorithm that we apply is illustrated in Fig. 4. Every iteration consists of updating check-to-symbol messages $\{R\}$, updating symbol-to-check messages $\{Q\}$ and tentative decoding, which can end the algorithm if it gives the

correct codeword $\hat{\mathbf{c}}$. We use $\{R\}$ to denote a set of check-to-symbol messages $R^a_{m,n}$, which are beliefs that the mth check is satisfied if c_n is considered fixed at the symbol $a \in \mathrm{GF}(2^q)$, and $\{Q\}$ to denote a set of symbol-to-check messages $Q^a_{m,n}$, which are beliefs that the c_n is the symbol a, given the information obtained from checks excluding the mth check, and the initial probabilities $P(\mathbf{c}|\mathbf{y})$. These beliefs can represent either estimated likelihoods or logarithms of likelihoods or LLRs and they can be computed with one of the well known methods, including BP, FFT-BP, Min-Max, Min-Sum, EMS, [3], [9], [10]. In this study, we have used the FFT-BP decoding algorithm [9].

When the iteration limit for HR decoding is reached without the correct tentative decoding result, the process is stopped and resumed as soon as the new IR chunk is received. Then for each subsequent chunk, the message sets $\{R\}$ and $\{Q\}$ are extended with additional elements, corresponding to a part of the matrix $\mathbf{H}_{\mathrm{IR}}$, which are initialized accordingly, and subsequent decoding iterations are processed.

IV. NONBINARY QC-RL-LDPC CODES EXPERIMENTAL DESIGN METHOD

In this section, we present an experimental framework for the design of NB QC-RL-LDPC codes, that is, finding the optimized HRC and IRC components of the QC-RL-LDPC code. The optimization criteria include the existence of cycles in the code graph, which is known to have an indirect influence on the code capabilities, as well as directly on the BER performance obtained by Monte-Carlo simulations of the system. We elected to use the simulation approach in IRC design instead of threshold optimization as previously proposed for binary codes (e.g. in [13]), because: 1) the threshold characterizes the asymptotic performance of the ensemble of codes, while our approach is aimed at short blocklength QC codes concrete realizations, which can be directly obtained and compared with the proposed method; 2) generalization of the RCA method [13] to the NB code is not straightforward and still gives only an approximate threshold; and 3) the proposed approach is compact and easy to apply in an effective computer search algorithm.

The method consists of two steps: HR code (HRC) and IR code (IRC) design. The Monte-Carlo simulation framework in the design procedures follows the model in Fig. 3 with AWGN channel and QPSK (4-QAM) modulation.

A. OPTIMIZING THE HIGHEST RATE CODE (HRC)

In general, the HRC is defined by a parity check matrix of the NB-LDPC code, which can be designed using any well-known method, usually based on graph optimization procedures. Although semi-regular codes are known to be a good choice for the NB case, the optimal column weight distributions $[\lambda_2, \lambda_3]$ for NB RL-LDPC codes are not obviously the same as those for rate-fixed quasi-regular NB-LDPC codes [8]. Therefore, the selection of $[\lambda_2, \lambda_3]$ are the design choices that are supported by the coding system simulation in the proposed algorithm. The specific distribution is obtained as a side result of HR code optimization, with the algorithm 1 provided below.

Algorithm 1: HRC design

Input: R,C,P
$\lambda_2 := R$, $\lambda_3 := C - \lambda_2$;
do
  Construct the base matrix $\mathbf{Z}_{R\times C}$ subject to $[\lambda_2, \lambda_3]$ distribution, and the dual-diagonal structure constraint (2), utilizing known graph construction method: PEG with concentrated row distribution [29], [37];
  if *Exhaustive search for CSVs is feasible* **then**
    From all possible combinations of CSVs in $\mathbf{S}_{\mathrm{HR}}$, select one with best cycles distribution in associated graph of $\mathbf{H}_{\mathrm{HR}}$;
  else
    In the base graph associated with $\mathbf{Z}$, identify closed walks;
    Find CSVs of size$P \times P$ in $\mathbf{S}_{\mathrm{HR}}$ with non-exhaustive heuristic search optimizing the identified closed walks elimination, e.g. as proposed in [5], [6], [8];
  For every row in $\mathbf{S}_{\mathrm{HR}}$, select a set of coefficients over $\mathrm{GF}(2^q)$ by a computer search to minimize the irresolvable cycles, see [38];
  According to obtained $\mathbf{S}_{\mathrm{HR}}$, construct the $\mathbf{H}_{\mathrm{HR}}$ part;
  Construct the $\mathbf{H}_{\mathrm{IR}}$ part;
  Simulate the obtained NB QC-RL-LDPC coding system for a selected set of SNRs. Save the code, if it is better than the best previously obtained;
  $\lambda_2 := \lambda_2 + 1$, $\lambda_3 := C - \lambda_2$
while $\lambda_2 \leq C$;
Output the saved, best performing $\mathbf{H}_{\mathrm{HR}}$.

The algorithm starts with $\lambda_2 = R$, which is the number of weight-2 columns required to achieve a dual-diagonal form of $\mathbf{H}_{\mathrm{HR}}$. Then, it proceeds by increasing λ_2 (and decreasing λ_3), up to a maximum $\lambda_2 = C$. The number of choices considered is quite small for short codes that we design, and if it was is too large, then the search could be constrained to the λ_2 values that are expected to be good choice for fixed-rate NB-LDPC codes [8].

B. OPTIMIZING THE INCREMENTAL REDUNDANCY CODE (IRC)

Because the aim of IRC is to provide additional parity checks if decoding is not successful with HRC, the rows of $\mathbf{H}_{\mathrm{IR}}$ are subsequently used in IR transmissions, in portion sizes equal to the IR chunk size. Therefore, the optimization of $\mathbf{H}_{\mathrm{IR}}$ is also performed from top to bottom, similar to the binary code design in [13], but using a Monte-Carlo simulation approach instead of an approximated asymptotic threshold.

At the core of IRC optimization is a greedy algorithm that finds consecutive macro-rows in $\mathbf{H}_{\mathrm{IR}}$, selecting symbols that should be combined in the next parity check transmitted. Some of the proposed HARQ schemes for binary codes [39], [40] successfully use an approach, in which each subsequent transmission resends a single bit (or a few bits) selected to help the decoder as much as possible given its current decoding state. These are the bits that are expected to be corrupted the most often in the current decoding state. We further developed this concept to apply it to the NB QC-RL-LDPC experimental construction.

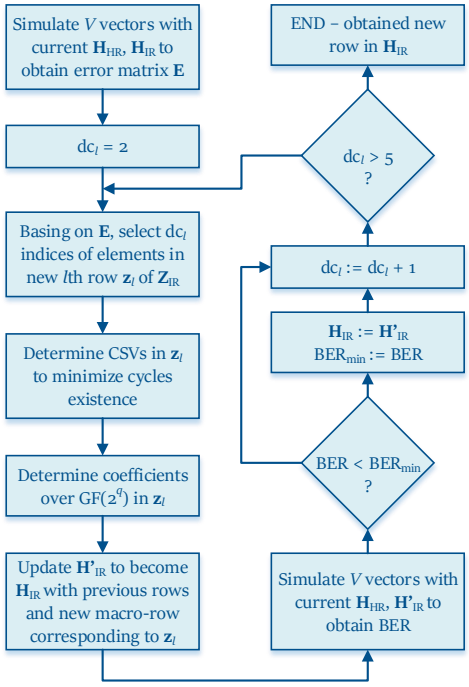


FIGURE 5. **Procedure for adding an *l*th macro-row to $\mathbf{H}_{\mathrm{IR}}$.**

Given the HR code obtained as presented in the previous section, the $\mathbf{H}_{\mathrm{IR}}$ is constructed by subsequent macro-rows generation, according to Algorithm 2 presented below.

Algorithm 2: IRC design

Input: $\mathbf{H}_{\mathrm{HR}}$,L

Iteration number $l := 1$;
do
  Add a new row $\mathbf{z}_l$ to the base matrix $\mathbf{Z}_{\mathrm{IR}}$ and a corresponding macro-row (P rows) to $\mathbf{H}_{\mathrm{IR}}$, according to the procedure outlined in Fig. 5;
  $l := l + 1$
while $l \leq L$;

In the procedure shown in Fig. 5, realizing the algorithm single iteration, the crucial part is the selection of dc_l indices of nonzero elements in a new row $\mathbf{z}_l$. The basic HARQ mentioned above, in which the symbols with the lowest reliability are resent, corresponds to $dc_l = 1$, because then every new parity check is dependent only on one previously sent symbol. For symbol selection, we use the error matrix $\mathbf{E}_{V\times N}$, which contains indications of erroneous symbols in the decoding with current (partially designed) $\mathbf{H}_{\mathrm{IR}}$, that is $e_{v,n}$ element ($n = 1 \ldots N$, $v = 1 \ldots V$) of $\mathbf{E}$ is either:

- 0 if symbol c_n of simulated vector v is correct,
- 1 if symbol c_n of simulated vector v is in error.

Therefore, a higher number of ones in the nth column of $\mathbf{E}$ indicates a higher probability of the symbol c_n being in error, according to the Monte Carlo simulations performed. Therefore, the symbol index selection policy is as follows: calculate the column weights in $\mathbf{E}$ and select n with the highest column weight, if $dc_l = 1$. The corresponding column in $\mathbf{z}_l$ is then marked as nonzero, and the algorithm proceeds to the subsequent steps.

We extended this concept to allow for higher row weights $dc_l > 1$. In our selection process, we follow this rationale: if more than one symbol is included in the check equation, it is desirable that only one symbol in this equation has low reliability. Therefore, for $dc_l > 1$, in the new check it is proposed to include one symbol corresponding to the highest column weight in $\mathbf{E}$ and $dc_l - 1$ symbols with the lowest column weights in $\mathbf{E}$. Finally, the new check is selected by additional simulations, as presented in Fig. 5 and from all considered choices, the best performing (in terms of BER) new row is added to $\mathbf{H}_{\mathrm{IR}}$.

Next, the coefficients over $\mathrm{GF}(2^q)$ are chosen from a precomputed collection of coefficient sets, as determined according to method proposed in [38]. The algorithm selects from the stored collection of candidate sets of coefficients for particular dc_l values. Then, a greedy computer search is executed, which iterates over all cycles that exist after the lifting process, and tries to modify: 1) the coefficient set choices, 2) the orders of the coefficients in respective rows, and 3) the multiplicative factor, in order to make as many cycles as possible irresolvable [6], [38].

V. RESULTS

To validate the transmission scheme and the NB code construction method developed, we evaluated the model using numerical experiments. We opted to compare the designed nonbinary schemes with a 5G binary LDPC configuration that employs Base Graph 2 (BG2) according to the standard [2], because BG2 is particularly optimized for scenarios involving small information block sizes and lower code rates in comparison to BG1. The first selected configuration uses the lifting size $P = 8$ of the 5G code. With this setting, the information block length is $K = 80$ bits, the HR code block length is $N = 112$, and the highest achievable rate – $R = 0.7143$. In our experimental evaluation, we conducted a comparative analysis with designed NB codes that have identical or closely similar highest rates and information block lengths expressed in bits (Kq bits), namely:

- codes over GF(16), with $N = 28, K = 20, P = 2$,

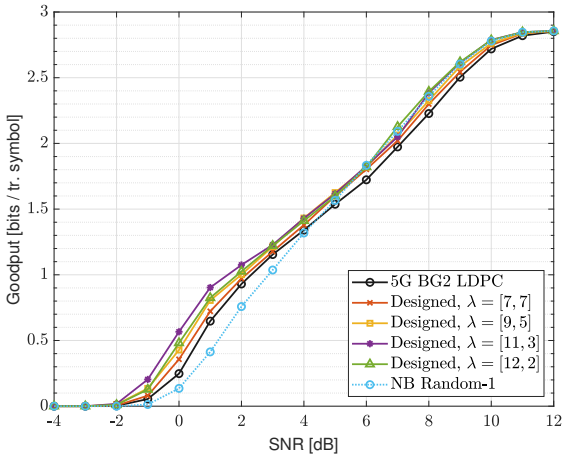


FIGURE 6. Results: comparison of goodput of the 16-QAM IR transmissions with the designed GF(16) codes and 5G BG2 codes, with information block length $K = 80$ bits and different column weigths of of semi-regular HRC codes.

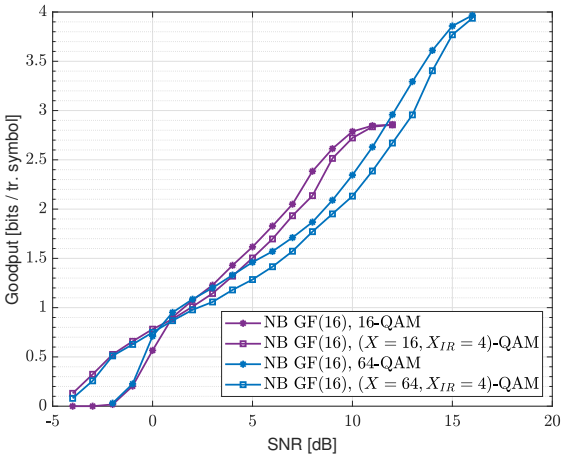


FIGURE 8. Results: comparison of goodput of the IR transmissions with 16-QAM and and a mix of QAM orders diversified for HR and IR transmissions, (X, X_{IR}).

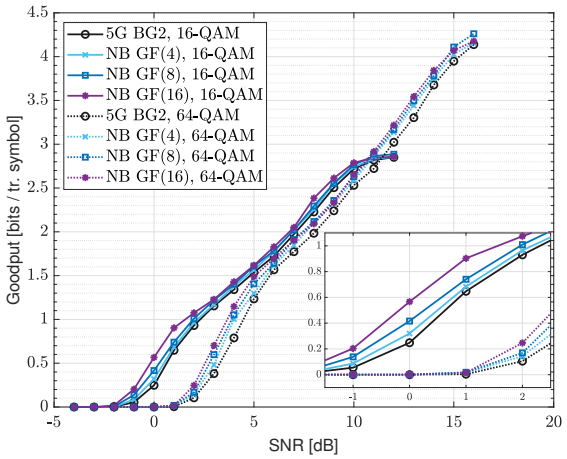


FIGURE 7. Results: comparison of goodput of the IR transmissions with the designed codes and 5G BG2 codes, with information block length $K = 80$ bits and diversified GF(2^q) and QAM modulation orders.

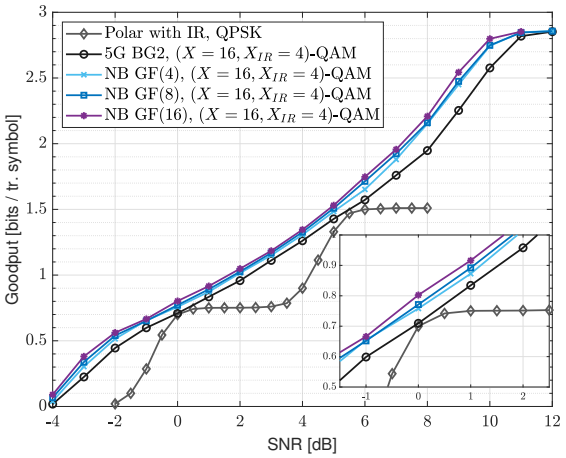


FIGURE 9. Results: comparison of the goodput of the IR transmissions with the designed codes, information block length $K = 180$, 5G BG2 codes with $K = 180$ as well as Polar Coded IR proposed in [41], with similar information block length $K = 192$.

- codes over GF(4), with $N = 56, K = 40, P = 4$,
- codes over GF(8), with $N = 36, K = 26, P = 2$, which involves minor deviation of the highest rate to $R = 0.7222$.

For the sake of simplicity in our comparison, the experimental setup omitted puncturing, and retransmissions were conducted in small chunks, each consistently comprising 12 bits. This choice aligns with the integer number of symbols over GF(4), GF(8) and GF(16), and the mappings of these segments to either six QPSK symbols, three 16-QAM symbols, or two 64-QAM symbols. Our simulations were carried out on an AWGN channel with varying SNR levels, and we evaluated the results in terms of transmission goodput. This metric is defined as the ratio of the number of information bits delivered successfully to the total count of QAM symbols transmitted, necessary for successful decoding [15].

In Fig. 6 we provide results of simulations of 16-QAM modulated AWGN transmissions, with several NB QC-RL-LDPC codes over GF(16) generated by applying the subsequent steps of Algorithm 1. This approach results in codes that exhibit diverse column weight distributions $\boldsymbol{\lambda} = [\lambda_2, \lambda_3]$ for the HRC, whereas the IRC is consistently designed using Algorithm 2 in each case. Among these codes, the one characterized by the distribution $\boldsymbol{\lambda} = [11, 3]$ shows slightly superior performance compared to the others, as well as superior performance to the 5G codes, simulated in the same environment and conditions. The $\boldsymbol{\lambda} = [11, 3]$ configuration was then used for further experiments with GF(16) codes. We also evaluated an NB code constructed using the same HRC, with $\boldsymbol{\lambda} = [11, 3]$, but with randomly selected checks in the IRC, with each row having a weight of $dc_l = 3$ for every l. This code configuration is referred to as "Random-

1.” The carefully designed IRC configurations obtained with Algorithm 2 significantly outperformed the Random-1 code of the same length.

In Fig. 7 we provide the experimental outcomes for a system employing NB QC-RL-LDPC codes over different fields: GF(16), GF(8) and GF(4), designed utilizing Algorithms 1 and 2. Results for fixed-order 16-QAM and 64-QAM constellations are provided. Overall, all the designed NB codes outperform the counterpart 5G binary solution in the IR system, with the best performance observed for the code over the highest field order, GF(16), as expected. We further explored the performance gap by calculating the ratios of the designed codes goodputs to the 5G code goodputs, for selected SNRs, and provided them in Tab. 1. For example, the ratio 1.1 means 10% more bits are delivered per transmitted symbol with a given NB code than 5G code. It can be observed that for higher SNRs, the gain is around several percent, while for lower SNRs, the gain expressed in this manner is significantly greater.

TABLE 1. Performance advantage of NB codes, expressed by ratios of goodputs for NB codes versus their 5G counterparts. Codes of information size $K = 80$ bits are combined with 16-QAM, while codes of information size $K = 180$ are combined with $(X = 16, X_{IR} = 4)$-QAM.

NB Code	SNR [dB]					
	−3.0	−2.0	−1.0	0.0	1.0	2.0
	Goodput in relation to 5G code					
$K = 80$, GF(4)	–	–	1.59	1.28	1.06	1.04
$K = 80$, GF(8)	–	–	2.8	1.66	1.14	1.09
$K = 80$, GF(16)	–	–	4.1	2.24	1.38	1.16
$K = 180$, GF(4)	1.35	1.16	1.08	1.08	1.03	1.05
$K = 180$, GF(8)	1.5	1.22	1.08	1.09	1.06	1.07
$K = 180$, GF(16)	1.69	1.27	1.1	1.14	1.1	1.09

The need for retransmission arises when the channel is in poor condition. In such cases, transitioning the system to a lower-order modulation, such as QPSK, during retransmissions could prove beneficial. We conducted experiments to assess the feasibility of such an approach. In Fig. 8, we present the results for a Raptor-like transmission scheme that employs a mix of QAM orders, combining the initial modulation of either $(X = 64)$-QAM or $(X = 16)$-QAM with retransmissions using $(X_{\mathrm{IR}} = 4)$-QAM (QPSK). The performance gain of this approach is visible in the low SNR regime.

The second configuration for the numerical experiments refers to the 5G BG1 code with lifting size $P = 18$, which gives a higher information block length of $K = 180$ bits, a code block length of HR code $N = 252$, and the highest achievable rate $R = 0.7143$. We performed a comparative analysis with designed NB codes possessing identical highest rates and information block lengths expressed in bits – Kq, that is:

- codes over GF(4), with $N = 126, K = 90, P = 6$,
- codes over GF(8), with $N = 84, K = 60, P = 6$,
- codes over GF(16), with $N = 63, K = 45, P = 3$.

In addition to comparison with a 5G code, we also take as a reference the an IR-HARQ scheme with Raptor extension of Polar codes, as proposed in [41]. From this publication we have taken the results for a scheme with information size $K = 192$ and QPSK modulation. The results of the comparison with the designed schemes with $K = 180$ are presented in Fig. 9. A significant gain over polar coded IR-HARQ is observed, while the performance advantage of the codes over 5G IR-HARQ, is similar to the previous case of shorter codes, which is also shown in Tab. 1.

VI. CONCLUSIONS

In this paper, we introduced a framework for the numerical construction and experimental performance verification of an NB extension of a QC-LDPC coding scheme with incremental redundancy, as applied in 5G. We have presented the NB coding combined with arbitrary QAM modulation order transmission system model and the iterative decoding, and we have designed the NB QC-RL-LDPC codes construction algorithm. Based on the experimental results discussed in the previous section, the following conclusions can be drawn:

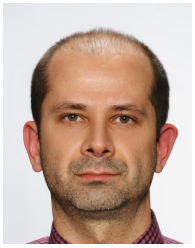
- TheMonte-Carlo simulation based NB QC-RL-LDPC construction method provides IRC codes that significantly outperform randomly constructed incremental checks.
- The comprehensive algorithm provides codes that satisfy the efficient encoding requirements and are structured for efficient decoding.
- The Raptor-Like NB coding exhibits a performance improvement over counterpart binary codes from 5G standard and an example Polar coded IR.
- Diversifying the modulation orders by mixing higher order modulations, for example $(X = 64)-$QAM for initial transmission, with a lower order, for example $(X_{\mathrm{IR}} = 4)-$QAM for IR, can be beneficial in terms of goodput for the lower-SNR regime (below 0dB and slightly above).

REFERENCES

[1] D. J. C. MacKay, “Good Error-Correcting Codes Based on Very Sparse Matrices,” *IEEE Trans. Inf. Theory*, vol. 45, pp. 399–431, March 1999.
[2] “3gpp ts 38.212 version 16.2.0 release 16. 5g; nr; multiplexing and channel coding,” 2020.
[3] M. C. Davey and D. MacKay, “Low-Density Parity Check Codes over GF(q),” *IEEE Commun. Lett.*, vol. 2, pp. 165–167, June 1998.
[4] L. Dolecek, D. Divsalar, Y. Sun, and B. Amiri, “Non-Binary Protograph-Based LDPC Codes: Enumerators, Analysis, and Designs,” *IEEE Trans. Inf. Theory*, vol. 60, pp. 3913–3941, July 2014.
[5] I. E. Bocharova, B. Kudryashov, and R. Johannesson, “Searching for Binary and Nonbinary Block and Convolutional LDPC Codes,” *IEEE Trans. Inf. Theory*, vol. 62, pp. 163–183, January 2016.
[6] W. Sulek, “Protograph Based Low-Density Parity-Check Codes Design with Mixed Integer Linear Programming,” *IEEE Access*, vol. 7, pp. 1424–1438, 2019.
[7] V. Wijekoon, E. Viterbo, Y. Hong, R. Micheloni, and A. Marelli, “A Novel Graph Expansion and a Decoding Algorithm for NB-LDPC Codes,” *IEEE Trans. Commun.*, vol. 68, pp. 1358–1369, March 2020.
[8] I. E. Bocharova, B. D. Kudryashov, E. P. Ovsyannikov, V. Skachek, and T. Uustalu, “Design and analysis of nb qc-ldpc codes over small alphabets,” *IEEE Trans. Commun.*, vol. 70, no. 5, pp. 2964–2976, 2022.
[9] D. Declercq and M. Fossorier, “Decoding Algorithms for Nonbinary LDPC Codes Over GF(q),” *IEEE Trans. Commun.*, vol. 55, pp. 633–643, April 2007.

[10] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urard, "Low-Complexity Decoding for Non-Binary LDPC Codes in High Order Fields," *IEEE Trans. Commun.*, vol. 58, pp. 1365–1375, May 2010.
[11] M. Yazdani and A. Banihashemi, "On construction of rate-compatible low-density Parity-check codes," *IEEE Commun. Lett.*, vol. 8, no. 3, pp. 159–161, March 2004.
[12] A. Shokrollahi, "Raptor codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.
[13] T.-Y. Chen, K. Vakilinia, D. Divsalar, and R. D. Wesel, "Protograph-Based Raptor-Like LDPC Codes," *IEEE Trans. Commun.*, vol. 63, no. 5, pp. 1522–1532, May 2015.
[14] D. Divsalar, S. Dolinar, C. R. JoJones, and K. Andrews, "Capacity-Approaching Protograph Codes," *IEEE J. Sel. Areas Commun.*, vol. 27, pp. 876–888, August 2009.
[15] W. Yu, K. Narayanan, J. Cheng, and J. Wu, "Raptor codes with descending order degrees for awgn channels," *IEEE Commun. Lett.*, vol. 24, no. 1, pp. 29–33, 2020.
[16] S. V. S. Ranganathan, D. Divsalar, and R. D. Wesel, "Quasi-Cyclic Protograph-Based Raptor-Like LDPC Codes for Short Block-Lengths," *IEEE Trans. Inf. Theory*, vol. 65, no. 6, pp. 3758–3777, 2019.
[17] F. Amirzade, M.-R. Sadeghi, and D. Panario, "Quasi-Cyclic Protograph-Based Raptor-Like LDPC Codes With Girth 6 and Shortest Length," in *2021 IEEE International Symposium on Information Theory (ISIT)*, Melbourne, Australia, July 2021, pp. 368–373.
[18] J. Kim, A. Ramamoorthy, and S. W. Mclaughlin, "The design of efficiently-encodable rate-compatible LDPC codes," *IEEE Trans. Commun.*, vol. 57, no. 2, pp. 365–375, February 2009.
[19] S. I. Park, Y. Wu, H. M. Kim, N. Hur, and J. Kim, "Raptor-like rate compatible ldpc codes and their puncturing performance for the cloud transmission system," *IEEE Transactions on Broadcasting*, vol. 60, no. 2, pp. 239–245, 2014.
[20] J. Fang, Y. L. Guan, G. Bi, L. Wang, and F. C. M. Lau, "Rate-Compatible Root-Protograph LDPC Codes for Quasi-Static Fading Relay Channels," *IEEE Trans. Veh. Technol.*, vol. 65, pp. 2741–2747, April 2016.
[21] H. Ro and H. Park, "Protograph-Based Raptor-Like LDPC Codes with Edge Addition for URLLC," in *ICC 2022 – IEEE International Conference on Communications*, May 2022, pp. 2930–2935.
[22] Z. He, K. Peng, and J. Song, "Multi-Layer Progressive Tree-Structured Edge Growth Algorithm for Nested Lifting Design of 5G-NR-Like LDPC Codes," *IEEE Commun. Lett.*, vol. 26, no. 12, pp. 2836–2840, 2022.
[23] M. P. C. Fossorier, "Quasi-Cyclic Low-Density Parity-Check Codes from Circulant Permutation Matrices," *IEEE Trans. Inf. Theory*, vol. 50, No. 8, pp. 1788–1793, August 2004.
[24] J. Li, K. Liu, S. Lin, and K. Abdel-Ghaffar, "Algebraic Quasi-Cyclic LDPC Codes: Construction, Low Error-Floor, Large Girth and a Reduced-Complexity Decoding Scheme," *IEEE Trans. Commun.*, vol. 62, pp. 2626–2637, August 2014.
[25] ——, "A Matrix-Theoretic Approach to the Construction of Non-Binary Quasi-Cyclic LDPC Codes," *IEEE Trans. Commun.*, vol. 63, pp. 1057–1068, April 2015.
[26] "Ieee standard: Ieee p802.16. local and metropolitan area networks part 16: Air interface for fixed and mobile broadband wireless access systems," 2006.
[27] T. Tian, C. Jones, J. D. Villasenor, and R. D. Wesel, "Selective Avoidance of Cycles in Irregular LDPC Code Construction," *IEEE Trans. Commun.*, vol. 52, No. 8, pp. 1242–1247, August 2004.
[28] H. Xu and B. Bai, "Superposition Construction of Q-Ary LDPC Codes by Jointly Optimizing Girth and Number of Shortest Cycles," *IEEE Commun. Lett.*, vol. 20, pp. 1285–1288, July 2016.
[29] X. Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and Irregular Progressive Edge-Growth Tanner Graphs," *IEEE Trans. Inf. Theory*, vol. 51, pp. 386–398, 2005.
[30] D. Vukobratovic and V. Senk, "Generalized ACE Constrained Progressive Edge-Growth LDPC Code Design," *IEEE Commun. Lett.*, vol. 12, pp. 32–34, 2008.
[31] G. Li, I. J. Fair, and W. A. Krzymien, "Density Evolution for Nonbinary LDPC Codes Under Gaussian Approximation," *IEEE Trans. Inf. Theory*, vol. 55, pp. 997–1015, March 2009.
[32] W. Sulek, "Quasi-Regular QC-LDPC Codes Derived From Cycle Codes," *IEEE Commun. Lett.*, vol. 20, pp. 1705–1708, 2016.
[33] T. J. Richardson and R. L. Urbanke, "Efficient Encoding of Low-Density Parity-Check Codes," *IEEE Trans. Inf. Theory*, vol. 47, pp. 638–656, February 2001.
[34] S. Myung, K. Yang, and J. Kim, "Quasi-Cyclic LDPC Codes for Fast Encoding," *IEEE Trans. Inf. Theory*, vol. 51, No. 8, pp. 2894–2901, August 2005.
[35] T. Richardson and S. Kudekar, "Design of low-density parity check codes for 5g new radio," *IEEE Commun. Mag.*, vol. 56, no. 3, pp. 28–34, 2018.
[36] H. Alasti and S. Gazor, "Vectored Implementation of Hierarchical 2^{2n} QAM," *IEEE Trans. Circuits Syst. II*, vol. 62, pp. 1103–1107, November 2015.
[37] H. Chen and Z. Cao, "A Modified PEG Algorithm for Construction of LDPC Codes with Strictly Concentrated Check-Node Degree Distributions," in *IEEE Wireless Communications and Networking Conference*, Kowloon, China, March 2007, pp. 564–568.
[38] C. Poulliat, M. Fossorier, and D. Declercq, "Design of Regular (2,dc)-LDPC Codes over GF(q) Using Their Binary Images," *IEEE Trans. Commun.*, vol. 56, pp. 1626–1635, October 2008.
[39] K. Vakilinia, S. V. S. Ranganathan, D. Divsalar, and R. D. Wesel, "Optimizing transmission lengths for limited feedback with nonbinary ldpc examples," *IEEE Trans. Commun.*, vol. 64, no. 6, pp. 2245–2257, 2016.
[40] A. Ahmed, A. Al-Dweik, Y. Iraqi, H. Mukhtar, M. Naeem, and E. Hossain, "Hybrid automatic repeat request (harq) in wireless communications systems and standards: A contemporary survey," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 4, pp. 2711–2752, 2021.
[41] L. Ma, J. Xiong, Y. Wei, and J. Ming, "An incremental redundancy HARQ scheme for polar code," *arXiv preprint arXiv: 1708.09679*, 2017.

JAKUB HYLA obtained his MSc in the discipline of Automatic Control, Electronics, and Computer Science from Silesian University of Technology, Gliwice, Poland, in 2018. Currently, he is employed at TKH Technology Poland Sp. z.o.o., part of the TKH Group. Concurrently, Jakub is actively pursuing a doctoral degree at Silesian University of Technology. His Ph.D. thesis focuses on IoT based LDPC codes encoder and decoder implementations. In his research, he explores rateless and nonbinary codes, and he employs them in practical applications on embedded devices. The outcomes of his research in LDPC codes design and implementation have been disseminated through several conference and journal papers.



WOJCIECH SUŁEK obtained his Ph.D. in Electronics from the Silesian University of Technology in Gliwice, Poland, in 2009. Since then, he has served as an assistant professor in the Department of Telecommunications and Teleinformatics at the same university. His Ph.D. thesis focused on the construction and implementation of Architecture-Aware LDPC codes. Wojciech's research interests include modern coding theory and hardware design for coding systems, which continue to be prominent in his work to this day. The results of his research in the field of LDPC codes design and implementation have been published in various conferences and journals, including the IEEE Transactions on Communications, IEEE Communications Letters, and IEEE Access.

6 Załączniki

6.1 Oświadczenia autorów o udziale w publikacjach

Gliwice, 25.08.2023 r.

Udział procentowy autorów w publikacji

W. Sułek i J. Hyla, „Aplikacje kodów korekcyjnych LDPC we współczesnych systemach radiokomunikacyjnych", w Elektronika, telekomunikacja, mobilność, t. 870, J. Izydorczyk, Red. 2020, s. 179–189.

Jakub Hyla – 40% .....

Wojciech Sułek – 60% ...

Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki
Katedra Telekomunikacji i Teleinformatyki
ul. Akademicka 16, pok. 916, 44-100 Gliwice
+48 32 237 11 57, wewn. 42
wojciech.sulek@polsl.pl

NIP 631 020 07 36
ING Bank Śląski S.A. o/Gliwice 60 1050 1230 1000 0002 0211 3056

Gliwice, 25.08.2023 r.

Oświadczenie autorów

W sprawie wkładu w przygotowanie publikacji:

W. Sułek i J. Hyla, „Aplikacje kodów korekcyjnych LDPC we współczesnych systemach radiokomunikacyjnych", w Elektronika, telekomunikacja, mobilność, t. 870, J. Izydorczyk, Red. 2020, s. 179–189.

Oświadczam, że mój wkład procentowy został oceniony na 40%. Polegał on na sporządzeniu przeglądu literaturowego oraz udziale w przygotowaniu tekstu manuskryptu.

....
mgr inż. Jakub Hyla

Oświadczam, że mój wkład procentowy został oceniony na 60%. Polegał on na przygotowaniu materiałów i tekstu manuskryptu.

....
dr hab. inż. Wojciech Sułek

Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki
Katedra Telekomunikacji i Teleinformatyki
ul. Akademicka 16, pok. 916, 44-100 Gliwice
+48 32 237 11 57, wewn. 42
wojciech.sulek@polsl.pl

NIP 631 020 07 36
ING Bank Śląski S.A. o/Gliwice 60 1050 1230 1000 0002 0211 3056

Gliwice, 25.08.2023 r.

Udział procentowy autorów w publikacji

J. Hyla, W. Sułek, W. Izydorczyk, L. Dziczkowski, i W. Filipowski, „Efficient LDPC Encoder Design for IoT-Type Devices", Applied Sciences-Basel, t. 12, Art. nr 5, 2022, doi: 10.3390/app12052558.

Jakub Hyla – 70% .....

Wojciech Sułek – 15% .....

Weronika Izydorczyk – 5% .....

Leszek Dziczkowski – 5% .....

Wojciech Filipowski – 5% .....

Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki
Katedra Telekomunikacji i Teleinformatyki
ul. Akademicka 16, pok. 916, 44-100 Gliwice
+48 32 237 11 57, wewn. 42
wojciech.sulek@polsl.pl

NIP 631 020 07 36
ING Bank Śląski S.A. o/Gliwice 60 1050 1230 1000 0002 0211 3056

Gliwice, 25.08.2023 r.

Oświadczenie autorów

W sprawie wkładu w przygotowanie publikacji:

J. Hyla, W. Sułek, W. Izydorczyk, L. Dziczkowski, i W. Filipowski, „Efficient LDPC Encoder Design for IoT-Type Devices", Applied Sciences-Basel, t. 12, Art. nr 5, 2022, doi: 10.3390/app12052558.

Oświadczam, że mój wkład procentowy został oceniony na 70%. Polegał on na przygotowaniu przeglądu literaturowego, implementacji algorytmów, opracowaniu wyników, przygotowaniu manuskryptu oraz odpowiedzi dla recenzentów.

.....
mgr inż. Jakub Hyla

Oświadczam, że mój wkład procentowy został oceniony na 15%. Polegał on na udziale w przygotowaniu manuskryptu oraz konsultacji w pracach nad algorytmami kodowania oraz odpowiedzi dla recenzentów.

.....
dr hab. inż. Wojciech Sułek

Oświadczam, że mój wkład procentowy został oceniony na 5%. Polegał on na udziale w przygotowaniu przeglądu literatury oraz ilustracji w manuskrypcie.

.....
dr inż. Weronika Izydorczyk

Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki
Katedra Telekomunikacji i Teleinformatyki
ul. Akademicka 16, pok. 916, 44-100 Gliwice
+48 32 237 11 57, wewn. 42
wojciech.sulek@polsl.pl

NIP 631 020 07 36
ING Bank Śląski S.A. o/Gliwice 60 1050 1230 1000 0002 0211 3056

Oświadczam, że mój wkład procentowy został oceniony na 5%. Polegał on na pracach koncepcyjnych oraz udziale w korekcie manuskryptu.

dr hab. inż. Leszek Dziczkowski

Oświadczam, że mój wkład procentowy został oceniony na 5%. Polegał on na on na udziale w korekcie manuskryptu oraz przygotowywaniu ilustracji.

dr ~~hab~~. inż. Wojciech Filipowski

Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki
Katedra Telekomunikacji i Teleinformatyki
ul. Akademicka 16, pok. 916, 44-100 Gliwice
+48 32 237 11 57, wewn. 42
wojciech.sulek@polsl.pl
NIP 631 020 07 36
ING Bank Śląski S.A. o/Gliwice 60 1050 1230 1000 0002 0211 3056

Gliwice, 25.08.2023 r.

Udział procentowy autorów w publikacji

J. Hyla i W. Sułek, „Dekoder LDPC implementowany w mikrokontrolerze dla systemów Internetu Rzeczy", Przegląd Elektrotechniczny, t. 99, Art. nr 4, 2023, doi: 10.15199/48.2023.04.23.

Jakub Hyla – 60% .....

Wojciech Sułek – 40% ...

Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki
Katedra Telekomunikacji i Teleinformatyki
ul. Akademicka 16, pok. 916, 44-100 Gliwice
+48 32 237 11 57, wewn. 42
wojciech.sulek@polsl.pl

Gliwice, 25.08.2023 r.

Oświadczenie autorów

W sprawie wkładu w przygotowanie publikacji:

J. Hyla i W. Sułek, „Dekoder LDPC implementowany w mikrokontrolerze dla systemów Internetu Rzeczy", Przegląd Elektrotechniczny, t. 99, Art. nr 4, 2023, doi: 10.15199/48.2023.04.23.

Oświadczam, że mój wkład procentowy został oceniony na 60%. Polegał on na przygotowaniu algorytmów dekodowania LDPC, opracowaniu wyników, przygotowaniu manuskryptu oraz odpowiedzi dla recenzentów.

mgr inż. Jakub Hyla

Oświadczam, że mój wkład procentowy został oceniony na 40%. Polegał on na przygotowaniu przeglądu literaturowego, konsultacji w zakresie dekodowania LDPC, udziale w przygotowaniu i korekcie manuskryptu.

.....
dr hab. inż. Wojciech Sułek

Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki
Katedra Telekomunikacji i Teleinformatyki
ul. Akademicka 16, pok. 916, 44-100 Gliwice
+48 32 237 11 57, wewn. 42
wojciech.sulek@polsl.pl
NIP 631 020 07 36
ING Bank Śląski S.A. o/Gliwice 60 1050 1230 1000 0002 0211 3056

Gliwice, 25.08.2023 r.

Udział procentowy autorów w publikacji

J. Hyla i W. Sułek, „Energy-efficient Raptor-like LDPC coding scheme design and implementation for IoT communication systems", Energies, t. 16, Art. nr 12, 2023, doi: 10.3390/en16124697.

Jakub Hyla – 70% .....

Wojciech Sułek – 30%

Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki
Katedra Telekomunikacji i Teleinformatyki
ul. Akademicka 16, pok. 916, 44-100 Gliwice
+48 32 237 11 57, wewn. 42
wojciech.sulek@polsl.pl

NIP 631 020 07 36
ING Bank Śląski S.A. o/Gliwice 60 1050 1230 1000 0002 0211 3056

Gliwice, 25.08.2023 r.

Oświadczenie autorów

W sprawie wkładu w przygotowanie publikacji:

J. Hyla i W. Sułek, „Energy-efficient Raptor-like LDPC coding scheme design and implementation for IoT communication systems", Energies, t. 16, Art. nr 12, 2023, doi: 10.3390/en16124697.

Oświadczam, że mój wkład procentowy został oceniony na 70%. Polegał on na opracowaniu modelu systemu, implementacji algorytmów kodowania adaptacyjnego, opracowaniu wyników, przygotowaniu manuskryptu oraz odpowiedzi dla recenzentów.

mgr inż. Jakub Hyla

Oświadczam, że mój wkład procentowy został oceniony na 30%. Polegał on na zaproponowaniu metody, udziale w opracowaniu wyników, udziale w przygotowaniu ilustracji oraz tekstu manuskryptu i korekty po recenzjach.

....
dr hab. inż. Wojciech Sułek

Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki
Katedra Telekomunikacji i Teleinformatyki
ul. Akademicka 16, pok. 916, 44-100 Gliwice
+48 32 237 11 57, wewn. 42
wojciech.sulek@polsl.pl

NIP 631 020 07 36
ING Bank Śląski S.A. o/Gliwice 60 1050 1230 1000 0002 0211 3056

Gliwice, 06.03.2024 r.

Udział procentowy autorów w publikacji

J. Hyla i W. Sułek, „Niebinarne kodowanie LDPC dla systemów Internetu rzeczy", Przegląd Elektrotechniczny, Art. nr 3, 2024, doi: 10.15199/48.2024.03.44.

Jakub Hyla – 70% ...

Wojciech Sułek – 30% ....

Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki
Katedra Telekomunikacji i Teleinformatyki
ul. Akademicka 16, pok. 916, 44-100 Gliwice
+48 32 237 11 57, wewn. 42
wojciech.sulek@polsl.pl

NIP 631 020 07 36
ING Bank Śląski S.A. o/Gliwice 60 1050 1230 1000 0002 0211 3056

Politechnika Śląska
UCZELNIA BADAWCZA

Wydział Automatyki, Elektroniki i Informatyki
Katedra Telekomunikacji i Teleinformatyki

Gliwice, 06.03.2024 r.

Oświadczenie autorów

W sprawie wkładu w przygotowanie publikacji:

J. Hyla i W. Sułek, „Niebinarne kodowanie LDPC dla systemów Internetu rzeczy", Przegląd Elektrotechniczny, Art. nr 3, 2024, doi: 10.15199/48.2024.03.44.

Oświadczam, że mój wkład procentowy został oceniony na 70%. Polegał on na przeglądzie literaturowym, przygotowaniu algorytmów kodowania niebinarnego LDPC, opracowaniu wyników, przygotowaniu manuskryptu oraz odpowiedzi dla recenzentów.

mgr inż. Jakub Hyla

Oświadczam, że mój wkład procentowy został oceniony na 30%. Polegał on na przeglądzie literaturowym, konsultacji w trakcie prac nad algorytmem kodowania niebinarnego LDPC, korekcie manuskryptu.

.....

dr hab. inż. Wojciech Sułek

Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki
Katedra Telekomunikacji i Teleinformatyki
ul. Akademicka 16, pok. 916, 44-100 Gliwice
+48 32 237 11 57, wewn. 42
wojciech.sulek@polsl.pl

NIP 631 020 07 36
ING Bank Śląski S.A. o/Gliwice 60 1050 1230 1000 0002 0211 3056

HR EXCELLENCE IN RESEARCH

6.2 Zaświadczenie opiekuna z przemysłu o wdrożeniu rozwiązania IoT.

TKH TECHNOLOGY

Let's create remarkable products. Together.

Leszno, 29.08.2023

Zaświadczenie o Wdrożeniu Rozwiązania IoT

Potwierdzam, że praca doktorska pt. "Efektywne kodowanie korekcyjne dla systemów transmisji w Internecie rzeczy", którą z powodzeniem zakończył doktorant Jakub Hyla, przyczynił się do wdrożenia rozwiązania w ramach projektu prowadzonego przez grupę TKH Group. Rozwiązanie to, oparte na efektywnym kodowaniu korekcyjnym, zostało skutecznie zaimplementowane jako kluczowy element projektu związanego z technologią Internetu rzeczy.

Rozwiązanie, którego koncept został opracowany w ramach pracy doktorskiej, okazało się nie tylko obiecującym teoretycznym podejściem, ale również praktycznym narzędziem zdolnym do zrewolucjonizowania systemów transmisji w obszarze IoT. Wdrożenie tego rozwiązania w ramach projektu grupy TKH Group otworzyło nowe perspektywy zarówno dla naszych działań badawczo-rozwojowych, jak i dla naszych partnerów biznesowych.

Warto podkreślić, że doktorant Jakub Hyla zaangażował się w pełni w proces wdrożenia, wykazując pozytywne zaangażowanie i profesjonalizm. Jego zdolności techniczne, umiejętność przystosowania teoretycznych koncepcji do praktycznych wymagań oraz determinacja w rozwiązywaniu ewentualnych problemów technicznych przyczyniły się w istotny sposób do sukcesu wdrożenia rozwiązania.

Rozwiązanie to nie tylko zyskało aprobatę i uznanie wewnętrzne w ramach grupy TKH Group, lecz także ma ogromny potencjał do wykorzystania w innych projektach naszej organizacji. Jego wpływ na zwiększenie niezawodności i wydajności transmisji danych w systemach IoT jest znaczący i stanowi cenny wkład w rozwój naszych produktów oraz usług.

Z pełnym przekonaniem potwierdzam, że rozwiązanie "Efektywne kodowanie korekcyjne dla systemów transmisji w Internecie rzeczy" zostało wdrożone jako integralny element projektu w ramach grupy TKH Group. Jego potencjał w poprawie jakości usług, optymalizacji kosztów i zwiększeniu niezawodności systemów IoT jest nie do przecenienia, a wkład doktoranta Jakuba Hyla w ten proces jest godny najwyższego uznania.

Z poważaniem,

TKH Technology Poland Sp. z o.o.
Piotr Niemczyk

Prezes Zarządu

TKH Technology Poland Sp. z o.o.
17 stycznia 119
64-100 Leszno, Poland

www.tkhtechnology.com
hello@tkhtechnology.com

NIP: 6972300690
KRS: 0000391310
REGON: 301805423