**WOJSKOWA AKADEMIA TECHNICZNA**

im. Jarosława Dąbrowskiego

**WYDZIAŁ ELEKTRONIKI**


# PRACA DYPLOMOWA

## Energooszczędny system mikroprocesorowy do rejestracji i szyfrowania danych na karcie microSD

*(temat pracy dyplomowej)*

**inż. Bartłomiej Polkowski s. Dariusza**

*(stopień wojskowy, tytuł zawodowy, imiona i nazwisko, imię ojca dyplomanta)*

## ELEKTRONIKA I TELEKOMUNIKACJA

*(kierunek studiów)*

**Systemy telekomunikacyjne**

*(specjalność)*

**Studia niestacjonarne drugiego stopnia - magisterskie**

*(forma i rodzaj studiów)*)*

**ppłk dr inż. Tadeusz Sondej**

*(stopień wojskowy, tytuł i stopień naukowy, imię i nazwisko promotora pracy dyplomowej)*

**WARSZAWA 2021**

# Spis treści

# 1. Wstęp

Urządzenia przenośne w obecnym świecie stały się codziennością. Ich funkcjonalność oraz ilość zapisywanych danych stale się powiększa. Wraz ze wzrostem ilości gromadzonych danych wzrasta również pobór energii elektrycznej. Optymalizacja zużycia prądu staje się ważnym aspektem w projektowaniu urządzeń. Zwiększenie energooszczędności pozwala na redukcje kosztów utrzymania produktów poprzez obniżenie opłat. Ma również niemały wpływ na środowisko naturalne przykładając się do obniżenia śladu węglowego. Może to odgrywać role w zadbaniu o wizerunek wielu światowych koncernów. Należy również zaznaczyć, że rozmiar oraz pojemność baterii urządzeń przenośnych w dalszym ciągu stanowi słabe ogniwo rozwiązań przenośnych.

Aby obniżyć poziom zużycia energii jednym z rozwiązań jest wprowadzenie systemu w tryb niskiego poboru energii między kolejnymi instrukcjami oprogramowania lub obniżenie zegara taktującego.

Oprócz wydajności oraz niskich kosztów eksploatacji w ostatnich latach unaocznił się problem bezpiecznego przechowywania danych. Wraz z rozwojem technologii rozwija się dziedzina cyberprzestępczości. Newralgiczne dane w niepowołanych rękach mogą poważnie wpłynąć, zarówno na życie osób prywatnych, jak i funkcjonowanie dużych firm. W elektronicznych systemach wojskowych bezpieczne zbieranie, agregacja oraz przechowywanie danych może mieć ogromne znaczenie na funkcjonowanie Sił Zbrojnych lub nawet życie i zdrowie żołnierzy.

Obecnie projektowane systemy mikroprocesorowe powinny zapewniać zarówno bezpieczeństwo zapisywanych danych jak i obniżenie kosztów użytkowania. W niniejszej pracy zaprojektowano oraz wykonano energooszczędny system mikroprocesorowy, do bezpiecznego gromadzenia danych akcelerometru na karcie microSD. Celem pracy jest porównanie istniejący rozwiązań energooszczędnych mikrokontrolerów oraz charakterystyk kart microSD. Wykonanie różnych metod odczytu, szyfrowania i zapisu danych na karcie pamięci oraz przeanalizowanie pod kątem wpływu na pobór energii elektrycznej.

# 2. Charakterystyka mikrokontrolerów o niskim zużyciu energii.

Ze względu na szerokie zastosowanie urządzeń zasilanych bateryjnie producenci mikrokontrolerów dążą do jak najmniejszego zużycia energii elektrycznej. Standardowo w swojej pracy urządzenia czekają przez długi czas na zarejestrowanie zdarzenia, które wymaga obsługi. W momencie oczekiwania mikrokontrolery charakteryzują się przejściem w jeden z trzech trybów: uśpienia, zatrzymania lub czuwania [1]. W zależności jakich funkcjonalności wykorzystuje system mikroprocesorowy dokonywany jest wybór którego z trybów użyć. Tryb uśpienia rdzeń mikrokontrolera jest wyłączony natomiast zegar oraz peryferia pozostają aktywne. Wyjście z tego trybu trwa najkrócej. Tryb uśpienia wyłącza oscylatory wysokich częstotliwości oraz większość bloków peryferii. Rdzeń wchodzi w stan głębokiego uśpienia. Tryb czuwania również korzysta z mechanizmów głębokiego uśpienia rdzenia oraz wyłączenia wewnętrznego stabilizatora napięcia 1,8 V. Ostatni z trybów jest najbardziej energooszczędny kosztem długiego czasu powrotu do normalnej pracy.

Analizy energooszczędności można dokonać na podstawie not katalogowych producentów. Podawane w nich wartości mogą być uzyskiwane w bardzo korzystnych warunkach do ich uzyskania. Istnieje wiele metod pomiaru prądu doświadczalnie jednak wiąże się to wysokimi kosztami zakupu aparatury pomiarowej oraz badanych mikrokontrolerów. Rozwiązaniem może być użycie analiz porównawczych wykonywanych przez firmy zewnętrzne.

## 2.1. Embedded Microprocessor Benchmark Consortium Benchmark

Na rynku komercyjnym istnieje wiele rozwiązań służących analizie porównawczej mikrokontrolerów. Spośród nich wyróżnia się testy funkcjonalności ale również obniżonego poboru mocy. Wiodącym na rynku jest rozwiązanie konsorcjum EEMBC (Embedded Microprocessor Benchmark Consortium). Jest to organizacja non-profit założona w 1997 roku. Składa się między innym z firmy członkowskie, współpracujące środowiska akademickiego oraz instytutów szkolnictwa wyższego [2]. Jest finansowana z opłat członkowskich i licencyjnych.

W zakresie działalności można znaleźć między innymi analizy [3]:

- Obliczeń heterogenicznych – skoncentrowanych na rdzeniach 8 i 16 bitowych;
- Wydajności jednordzeniowego procesora;

- Symetrycznej wydajności wielordzeniowych procesorów;
- Wydajności przeglądarek internetowych telefonów i tabletów;
- Obniżonego poboru mocy mikroprocesor – Ultra-Low Power and Internet Of Things.

Ostatni z podpunktów został dodatkowo podzielony ze względu na zastosowanie i poddany testom pod względem [4]:

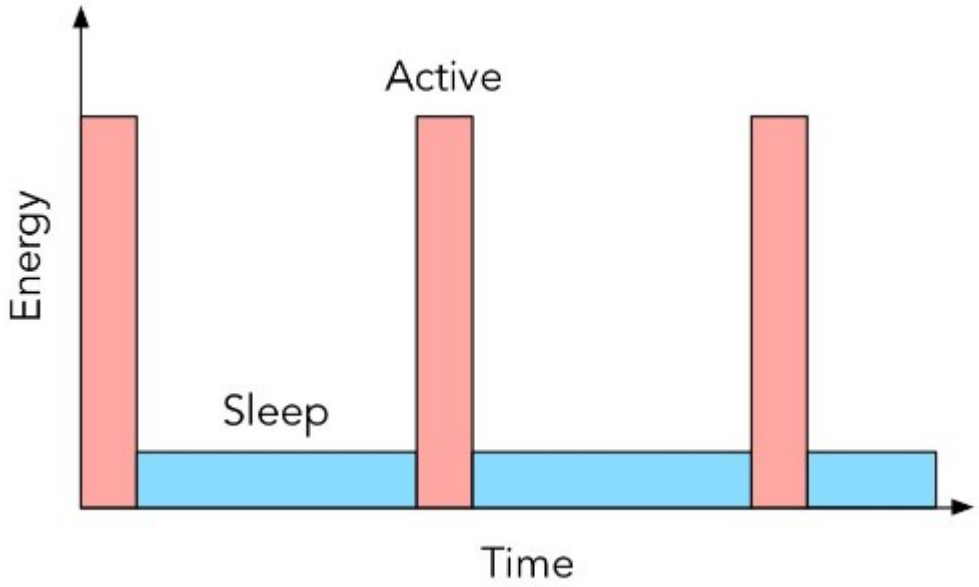
- ULPMark – obniżonego poboru mocy;
- IoTMark - wydajności węzła brzegowego urządzenia Internetu rzeczy
- SecureMark - kosztów energii ponoszonych przez implementacje algorytmów bezpieczeństwa.

Dalsza część pracy koncentruje się na testach porównawczych ULPMark. Są to profile analizy mocy i energii pobieranej przez MCU. W Tabela Charakterystyka mikrokontrolerów o niskim zużyciu energii..1 został zawarty zestaw funkcjonalności testów.

*Tabela Charakterystyka mikrokontrolerów o niskim zużyciu energii..1. Funkcjonalności ULPMark.*

| **Wariant ULPMark** | **Wartość pomiaru** |
|---|---|
| ULPMark-CoreProfile | Koszt energii w trybie głębokiego uśpienia |
| ULPMark-PeripheralProfile | Wpływ urządzeń peryferyjnych na pobór mocy |
| ULPMark-CoreMark | Moc czynnie obciążonego procesora |
| ULPMark-ML (Zostanie wprowadzone w przyszłości) | Moc czynna przy użyciu uczenia maszynowego. |

CoreProfile symulujący pracę w warunkach zbliżonych do rzeczywistego układu. Koncentruje się głównie na poborze mocy w czasie głębokiego uśpienia oraz przejścia do trybu aktywnej pracy. Polega na naprzemiennym załączaniu jednosekundowego obciążenia i wydłużonym względem czasu pracy tryb uśpienia. Na Rysunek Charakterystyka mikrokontrolerów o niskim zużyciu energii..1 przedstawiono sekwencje działania.




*Rysunek Charakterystyka mikrokontrolerów o niskim zużyciu energii..1. Wykres obrazujący zużycie energii przez mikroprocesor w czasie.*

PeripheralProfile jest przeprowadzana we współdziałaniem mikroprocesora z układami peryferyjnymi. Są nimi, zegar czasu rzeczywistego(RTC), magistrala komunikacyjna SPI, generator sygnału PWM oraz konwerter analogowo-cyfrowy(ADC). W testach uwzględniane są również inne parametry takie jak częstotliwość pracy procesora oraz porównanie czasu i poboru mocy w stanie standardowej pracy oraz trybu uśpienia. Benchmark jest złożony z 10 kroków przedstawionych w Tabela Charakterystyka mikrokontrolerów o niskim zużyciu energii..2.

*Tabela Charakterystyka mikrokontrolerów o niskim zużyciu energii..2. Sekwencja działania PeripheralProfile*

| Krok | ADC | PWM | SPI | RTC |
|---|---|---|---|---|
| 1 | Odczyt: 64 próbki<br>Częstotliwość: 1 kHz | Impulsów: 20<br>Okres: 255<br>Wypełnienie: 10%<br>Częstotliwość: 32,768Hz<br>Zbocze: stałe | Włączony | Ustawienie i uruchomienie |
| 2 | Odczyt: 64 próbki<br>Częstotliwość: 1 kHz<br>Przeliczenie danych w buforze | Impulsów: 40<br>Okres: 255<br>Wypełnienie 20%<br>Częstotliwość 32,768Hz<br>Zbocze: narastające | Włączony | Włączony |
| 3 | Odczyt: 1 próbka<br>Częstotliwość: 1 Hz | Impulsów: 40<br>Okres: 255<br>Wypełnienie: 30%<br>Częstotliwość: 32,768Hz<br>Zbocze: stałe | Włączony | Włączony |
| 4 | Odczyt: 1 próbka<br>Częstotliwość: 1 Hz | Impulsów: 100<br>Okres: 255<br>Wypełnienie: 40%<br>Częstotliwość: 32,768Hz<br>Zbocze: stałe | Nadanie 128 Bajtów | Włączony |
| 5 | Odczyt: 1 próbka<br>Częstotliwość: 1 Hz | Impulsów: 20<br>Okres: 255<br>Wypełnienie 50%<br>Częstotliwość 32,768Hz<br>Zbocze: stałe | Odebranie ostatniego nadanego Bitu i nadanie 128 Bajtów | Włączony |
| 6 | Odczyt: 1 próbka<br>Częstotliwość: 1 Hz | Impulsów: 20<br>Okres: 255<br>Wypełnienie: 60%<br>Częstotliwość: 32,768Hz<br>Zbocze: stałe | Odebranie ostatniego nadanego Bajtu i nadanie 128 Bajtów | Włączony |
| 7 | Odczyt: 1 próbka<br>Częstotliwość: 1 Hz | Impulsów: 20<br>Okres: 255<br>Wypełnienie: 70%<br>Częstotliwość: 32,768Hz<br>Zbocze: stałe | Odebranie ostatniego nadanego Bajtu i nadanie 128 Bajtów | Włączony |
| 8 | Odczyt: 1 próbka<br>Częstotliwość: 1 Hz | Impulsów: 20<br>Okres: 255<br>Wypełnienie: 80%<br>Częstotliwość: 32,768Hz<br>Zbocze: stałe | Odebranie ostatniego nadanego Bajtu i nadanie 128 Bajtów | Włączony |
| 9 | Odczyt: 1 próbka<br>Częstotliwość: 1 Hz | Impulsów: 30<br>Okres: 10000<br>Wypełnienie: 10%<br>Częstotliwość: 1 MHz<br>Zbocze: stałe | Odebranie ostatniego nadanego Bajtu i nadanie 128 Bajtów | Włączony |
| 10 | Wyłączenie i sprawdzenie dancyh | Wyłączony | Odebranie ostatniego nadanego Bajtu | Zatrzymanie i sprawdzenie |

Po wykonanej sekwencji urządzenie przechodzi w tryb uśpienia. Przez to wydajniejsze sprzętowo urządzenia mogą szybciej przejść w tryb obniżonego poboru mocy.

CoreMark jest Benchmarkiem pomagającym w dokonaniu wyboru miedzy wydajnością a energooszczędnością. Efektem wykonania testów jest określenie punktu pracy na osi wydajności w funkcji zużycia energii. Test ukazuje trzy konfiguracje ważne do uwzględnienia podczas projektowania systemu: Najwydajniejsza konfiguracja peryferii, najoptymalniejsza energetycznie konfiguracja przy najniższym i 3 V napięciu zasilania.

Do przeglądu dostępnych na rynku energooszczędnych mikrokontrolerów w niniejszej pracy został wykorzystany benchmark CoreProfile(3.0V). Konsorcjum EMBC zawiera największą ilość przetestowanych w ten sposób mikrokontrolerów.

## 2.2. NXP Kinetis „L”

Firma NXP w swojej ofercie posiada sześć rodzin zapewniających mechanizmy „ultra low-power”. Rodzina mikrokontrolerów Kinetis jest oznaczona literą „L”. Każda z serii używa rdzenia ARM Cortex-M0+ i jest dedykowana konkretnym funkcjonalnością przedstawionym w Tabela Charakterystyka mikrokontrolerów o niskim zużyciu energii..3.

*Tabela Charakterystyka mikrokontrolerów o niskim zużyciu energii..3. Właściwości serii mikrokontrolerów NXP*

| Rodzina | Maksymalna częstotliwość | Pamięć | Funckjonalności |
|---|---|---|---|
| KL8x | 96 MHz | Flash: 128 KB<br>SRAM: 96 KB | Mechanizmy bezpieczenstwa |
| KL4x | 48 MHz | Flash: 128-256 KB<br>SRAM: 16-32 KB | USB + Wyświetlacz segmentowy LCD |
| KL3x | 48 MHz | Flash: 32-256 KB<br>SRAM: 4-32 KB | Wyświetlacz segmentowy LCD |
| KL2x | 48 MHz/72MHz | Flash: 32-512 KB<br>SRAM: 4-128 KB | USB |
| KL1x | 48 MHz | Flash: 32-256 KB<br>SRAM: 4-32 KB | Powszechne funkcjonalności |
| KL0x | 48 MHz | Flash: 8-32 KB<br>SRAM: 1-4 KB | Podstawowe funkcjonalności |

Seria mikrokontrolerów KL0x posiada minimalne zasoby sprzętowe mogące stanowić alternatywę dla mikrokontrolerów 8-bitowych. Kolejna seria oznaczona KL1x jest wyposażona w rozszerzony zestaw peryferii między innymi I2S. Ilość pamięci pozwalających zastosować je w projektach powszechnego użycia. Linia KL2x posiada w swoim standardzie USB-FullSpeed oraz sprzętowe mechanizmy bezpieczeństwa. KL3x zapewnia wsparcie wyświetlaczy segmentowych LCD natomiast następna w kolejności KL4x łączy funkcjonalności dwóch poprzednich serii z pominięciem mechanizmów bezpieczeństwa. Ostatnia z serii oznaczona

KL8x zawiera wsparcie dla wszystkich funkcjonalności swoich poprzedników i charakteryzuje się największą ilością pamięci RAM. Zawiera w sobie mechanizm awaryjnego kasowania informacji oraz oprogramowania wbudowanego pod nazwą „TamperDetection” [5]

Wartym zaznaczenia jest rozwiązanie zastosowane w rodzinie LPC5411x. Mikrokontrolery tak jak przedstawiony na Rysunek Charakterystyka mikrokontrolerów o niskim zużyciu energii..2. Mikrokontroler LPC54114 składają się z dwóch rdzeni Cortex-M4 oraz Cortex-M0+. Drugi z wymienionych jest koprocesorem umożliwiający całkowite wyłączenie pierwszego w celu maksymalnego obniżenia pobieranej energii w trybie uśpienia. [6]

*Rysunek Charakterystyka mikrokontrolerów o niskim zużyciu energii..2. Mikrokontroler LPC54114*

## 2.3. SiliconLabs EFM32

W ofercie firmy SiliconLabs znajduję się rodzina mikrokontrolerów EFM32 dedykowanych do zasilania bateryjnego. Mikrokontrolery wykorzystują mikroprocesory ARM z rdzeniem Cortex-M0+, Cortex-M3, Cortex-M4. W swoich szeregach zawiera serie [7]:

- Giant Gecko G11
- Pearl Gecko PG12
- Pearl Gecko PG1
- Happy Gecko
- Zero Gecko
- Tiny Gecko

- Gecko

Funkcjonalności poszczególnych serii zostały przedstawione w Tabela Charakterystyka mikrokontrolerów o niskim zużyciu energii..4.

*Tabela Charakterystyka mikrokontrolerów o niskim zużyciu energii..4. Funkcjonalności poszczególnych serii mikrokontrolerów SiliconLabs*

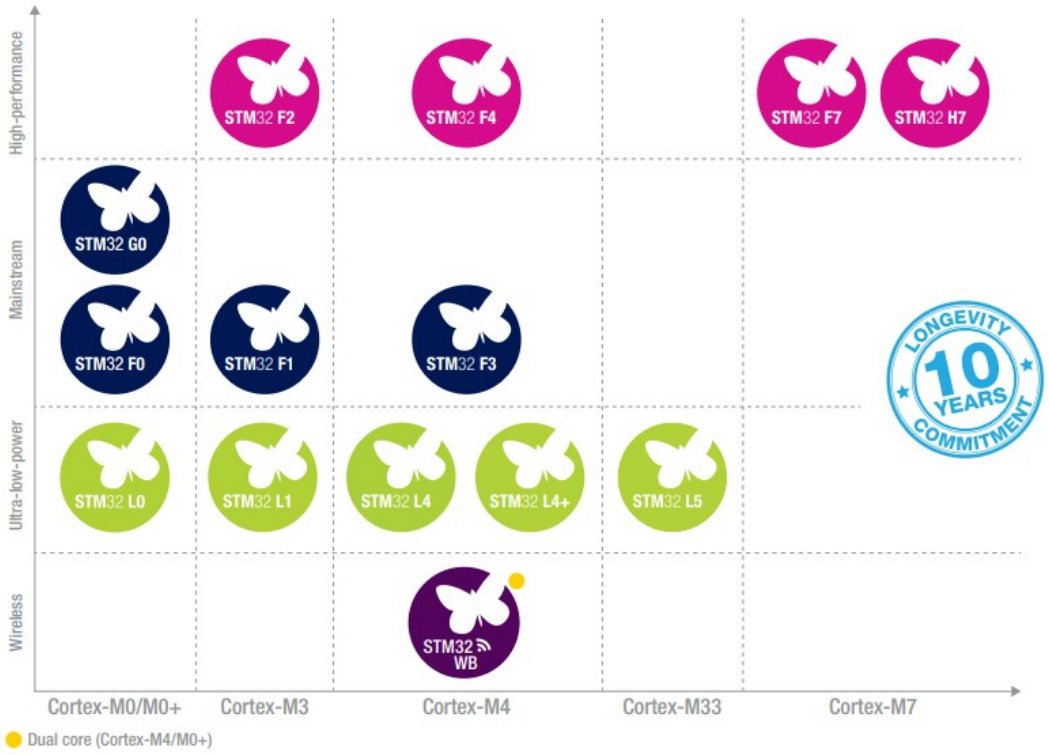
<table>
<tr><th>Nazwa Serii</th><th>Rdzeń ARM</th><th>Pamięć flash [kB]</th><th>Wsparcie mechanizmów bezpieczeństwa</th><th>Wyświetlacz LCD</th><th>USB</th></tr>
<tr><td>Giant Gecko G11</td><td rowspan="3">Cortex-M4</td><td>2048</td><td rowspan="6">Tak</td><td>Tak</td><td>Tak</td></tr>
<tr><td>Pearl Gecko PG12</td><td>1024</td><td rowspan="4">Nie</td><td rowspan="2">Nie</td></tr>
<tr><td>Pearl Gecko PG1</td><td>128-256</td></tr>
<tr><td>Happy Gecko</td><td rowspan="2">Cortex-M0+</td><td>32-64</td><td>Tak</td></tr>
<tr><td>Zero Gecko</td><td>4-32</td><td rowspan="3">Nie</td></tr>
<tr><td>Tiny Gecko</td><td rowspan="2">Cortex-M3</td><td>4-32</td><td rowspan="2">Tak</td></tr>
<tr><td>Gecko</td><td>16-128</td></tr>
</table>

Każda z serii mikrokontrolerów jest wyposażona w akcelerator kryptograficzny dla szyfrów AES, ECC oraz SHA. Posiadają również moduł zarządzania bezpieczeństwem, który zapewnia uprzywilejowany dostęp do wybranych urządzeń peryferyjnych [8]. Serie Happy Gecko, Zero Gecko oraz Tiny Gecko charakteryzują się najmniejszą ilością komórek pamięci oraz wbudowanych mechanizmów. Zarazem jest to najekonomiczniejsze rozwiązanie. Do segmentu rozwiązań uniwersalnych zalicza się platformy z serii Gecko oraz Pearl Gecko. Są one wyposażone w większą ilość pamięci oraz szersze spektrum modułów peryferyjnych. Najwięcej możliwości zawiera seria Giant Gecko GG11. W tej serii znajduje się mikrokontroler SLSTK3701A GG11 umieszczony na Rysunek Charakterystyka mikrokontrolerów o niskim zużyciu energii..3. Zawiera on wbudowany debugger SEGGER J-Link oraz zaawansowany system monitorowania energii w czasie rzeczywistym [9].


*Rysunek Charakterystyka mikrokontrolerów o niskim zużyciu energii..3. Płytka developerska EFM32 GIANT GECKO S1 SLSTK3701A*

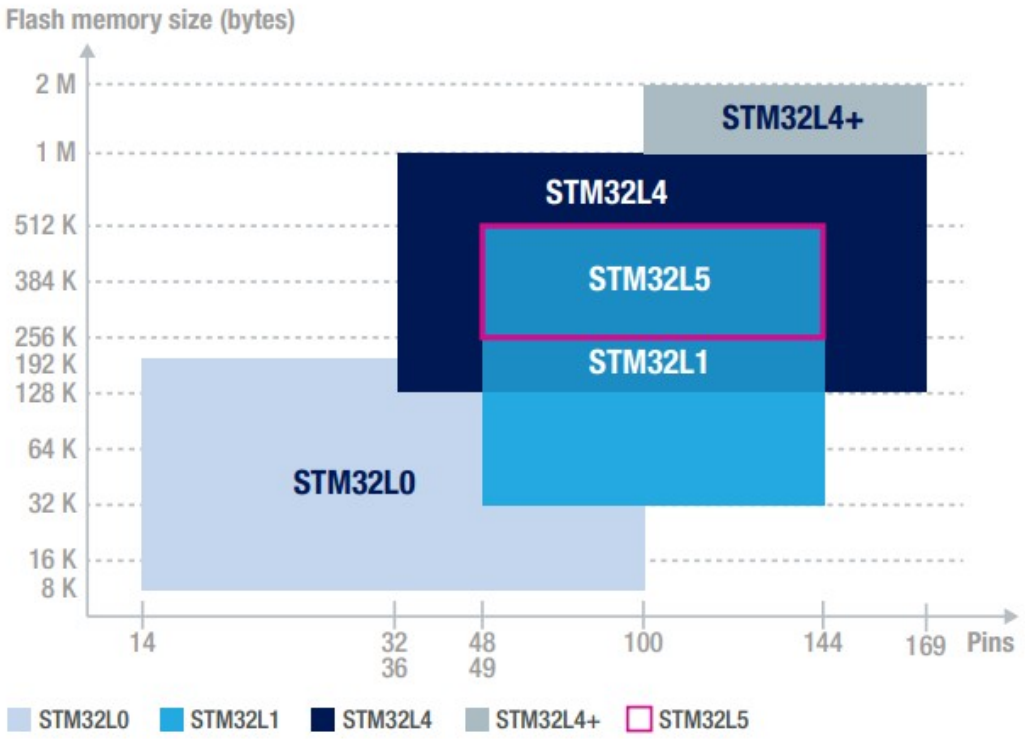
## 2.4. STMicroelectronics STM32Lx

Spośród rozwiązań oferowanych przez STMicroelectronics możemy znaleźć wiele rodzin specjalizujących się w różnych dziedzinach. Na Rysunek Charakterystyka mikrokontrolerów o niskim zużyciu energii..4 przedstawiono na bazie jakich rdzeni zostały zbudowane poszczególne rodziny. Mikrokontrolery skupione na wydajności prądowej oznaczane są literą „L” [10]. Producent proponuje swoje rozwiązania w branży fitness jako mierniki aktywności fizycznej. Inteligentne mierniki gazu, wody lub zarządzania „Smart Home”. Również w urządzeniach odtwarzających i rejestrujących dźwięk. Firma kompleksowo wspiera konstruowanie energooszczędnych systemów mikroprocesorowych. W swojej ofercie posiadają wsparcie sprzętowe w postaci modułu STM32 Power Shield X-Nucleo-LPM01A do pomiaru poboru mocy średniej oraz w czasie rzeczywistym. Oprogramowanie STM32CubeMonitor-Power zapewnia wsparcie programowe wartości zmierzonych modułem rozszerzeniowym.




*Rysunek Charakterystyka mikrokontrolerów o niskim zużyciu energii..4. Rodziny mikrokontrolerów STM*

Rysunek Charakterystyka mikrokontrolerów o niskim zużyciu energii..5 przedstawia podział rodziny na poszczególne serie ze względu na posiadane zasoby pamięci oraz ilość wyprowadzeń. Mikroprocesory serii L0 zbudowana na bazie rdzenia Cortex-M0/0+ posiadają najmniejszą ilość pamięci Flash oraz wyprowadzeń jednak zastosowany rdzeń pozwala na osiąganie bardzo dobrze zoptymalizowanych prądowo rozwiązań. Seria L1 powstała jako pierwsza z rozwiązań energooszczędnych firmy posiadając ten sam rdzeń Cortex-M3 co powszechnie używane mikrokontrolery z serii F1. Cortex-M4 pozwala na zbudowanie najbardziej wymagających aplikacji o szerokim zestawie bloków peryferyjnych na bazie którego zostały zbudowane serie L4 i L4+.




*Rysunek Charakterystyka mikrokontrolerów o niskim zużyciu energii..5. Zasoby mikroprocesorów z serii L*

Najbardziej zoptymalizowanym poborem mocy w stosunku do posiadanych możliwości cechuje się seria L5. Rdzeń Cortex-M33 posiadające wewnętrzne zabezpieczenia w postaci technologii TrustZone zapewnia dodatkowe mechanizmy bezpieczeństwa.

Najnowszym produktem firmy nie ujętym na poprzednich rysunkach jest seria U5. Jest również oparta o Cortex-M33 i dedykowana jest rozwiązaniom dotyczących mocy, wydajności i bezpieczeństwa [11]. Grupa docelowa tego rozwiązania to sektor medyczny, przemysłowy i automatyki domowej. Seria ta wprowadza istotne zmiany w kwestiach cyberbezpieczeństwa. Szyfrowanie AES oraz akcelerator klucza publicznego zostały uodpornione na ataki kanału bocznego. Wewnętrzne dane zostały zabezpieczone unikalnym kluczem sprzętowym [12]. Pamięć flash wzrosła do 2048 kB, wsparcie dla interfejsów USB-Type-C z kontrolerem zasilania oraz dwa nowoczesne interfejsy Octo-SPI. Wprowadzono mechanizmy energooszczędnego bezpośredniego dostępu do pamięci (LPDMA Low Power Direct Memory Access) zapewniające działanie modułów peryferyjnych w trybie zatrzymania.

## 2.5. Porównanie wybranych mikrokontrolerów

Przegląd rozwiązań pozwala na zestawienie wybranych mikrokontrolerów wymienionych firm oraz porównanie ich energooszczędności. W tym celu posłużono się benchmarkami ULPMark CoreProfile i PeripherialProfile przedstawionymi w Tabela

Charakterystyka mikrokontrolerów o niskim zużyciu energii..5. Dodatkowo w Tabela Charakterystyka mikrokontrolerów o niskim zużyciu energii..6 zestawiono wybrane właściwości z kart katalogowych mikrokontrolerów. Dane te mają istotny wpływ podczas dokonywania wyboru urządzenia na etapie projektowania systemu mikroprocesorowego. Spośród wielu produktów na rynku mikrokontrolerów do analizy zostały wybrane mikrokontrolery firm przedstawionych w poprzednich rozdziałach, koncentrując się na 32-bitkowych mikrokontrolerach zbudowanych na bazie rdzenia ARM Cortex-M.

*Tabela Charakterystyka mikrokontrolerów o niskim zużyciu energii..5. Wyniki benchmarków EMBC*

| Producent | Rdzeń | Seria | ULP-Mark | |
|---|---|---|---|---|
| | | | CoreProfile (3 V) | PeripherialProfile (3 V) |
| NXP | Cortex-M0+ | KL0x | 116 | - |
| NXP | Cortex-M0+ | KL1x | 116 | - |
| NXP | Cortex-M0+ | KL2x | 116 | - |
| NXP | Cortex-M0+ | KL3x | 116 | - |
| NXP | Cortex-M0+ | KL4x | 116 | - |
| NXP | Cortex-M0+ | KL8x | 116 | - |
| NXP | Cortex-M4/M0+ | LPC5411x | 116 | - |
| SiliconLabs | Cortex-M0+ | Happy Gecko | 101 | 8 |
| SiliconLabs | Cortex-M0+ | Zero Gecko | 113 | 8 |
| SiliconLabs | Cortex-M3 | Gecko | - | - |
| SiliconLabs | Cortex-M3 | Tiny Gecko | 97 | 64 |
| SiliconLabs | Cortex-M4 | Pearl Gecko PG1 | 106 | - |
| SiliconLabs | Cortex-M4 | Pearl Gecko PG12 | 72 | - |
| SiliconLabs | Cortex-M4 | Giant Gecko G11 | 82 | 51 |
| STM | Cortex-M0+ | L0 | 244 | 95 |
| STM | Cortex-M3 | L1 | 155 | - |
| STM | Cortrx-M4 | L4 | 447 | 167 |
| STM | Cortrx-M5 | L4+ | 233 | 56 |
| STM | Cortex-M33 | L5 | 402 | 56 |
| STM | Cortex-M34 | U5 | 535 | 149 |

Wyniki firmy należy interpretować według zasady najwyższy wynik jest najlepszy. Wartości są przeliczane według następującego wzoru:

ULPMark-CP/PP = 1000 / mediana z 5 przebiegów testowych trwających 10 sekund.

Niestety nie zostały przeprowadzone wszystkie mikrokontrolery ujęte w przeglądzie. Zwłaszcza badań modułów peryferyjnych. Najlepsze wyniki w obydwóch kategoriach osiągnęła seria STM32U5 oraz STM32L4.

*Tabela Charakterystyka mikrokontrolerów o niskim zużyciu energii..6. Zestawienie danych z not katalogowych*

| Producent | Rdzeń | Seria | Najniższy pobór prądu [µA/MHz] | Pobór prądu w stanie norlnej pracy [µA] | Czas wybudzenia [µs] |
|---|---|---|---|---|---|
| NXP | Cortex-M0+ | KL0x | 115 | 0,07 | 152 |
| NXP | Cortex-M0+ | KL1x | 98 | 0,09 | 152 |
| NXP | Cortex-M0+ | KL2x | 36 | 0,09 | 188 |
| NXP | Cortex-M0+ | KL3x | 50 | 0,07 | 113 |
| NXP | Cortex-M0+ | KL4x | 120 | 0,18 | 113 |
| NXP | Cortex-M0+ | KL8x | 148 | 0,27 | 138 |
| NXP | Cortex-M4/M0+ | LPC5411x | 102 | 0,36 | 1200 |
| SiliconLabs | Cortex-M0+ | Happy Gecko | 132 | 0,02 | 163 |
| SiliconLabs | Cortex-M0+ | Zero Gecko | 114 | 0,02 | 163 |
| SiliconLabs | Cortex-M3 | Gecko | 180 | 0,02 | 163 |
| SiliconLabs | Cortex-M3 | Tiny Gecko | 150 | 0,02 | 282 |
| SiliconLabs | Cortex-M4 | Pearl Gecko PG1 | 63 | 0,04 | 290 |
| SiliconLabs | Cortex-M4 | Pearl Gecko PG12 | 64 | 0,06 | 290 |
| SiliconLabs | Cortex-M4 | Giant Gecko G11 | 77 | 0,11 | 163 |
| STM | Cortex-M0+ | L0 | 166 | 1,95 | 50 |
| STM | Cortex-M3 | L1 | 177 | 1,33 | 50 |
| STM | Cortrx-M4 | L4 | 31 | 0,08 | 40 |
| STM | Cortrx-M5 | L4+ | 55 | 0,03 | 40 |
| STM | Cortex-M33 | L5 | 60 | 0,03 | 40 |
| STM | Cortex-M33 | U5 | 19 | 0,11 | 40 |

Przegląd deklarowanych przez producenta wartości wykazuje podobną zależność jak ocena ULPMark z poprzedniej tabeli. Najniższym poborem mocy w stanie najgłębszego uśpienia, pracy normalnej oraz czasu wybudzenia osiągnęły serie STM32L4 oraz STM32U5.

# 3. Cechy i parametry kart pamięci microSD

Systemy mikroprocesorowe potrzebują nieulotnych pamięci typu flash do przechowywania oraz przetwarzania danych. Do tego celu mogą być wykorzystane specjalne układy scalone zamontowane na stałe do jednej z płytek urządzenia. Pamięć taka może być również przenośna na co pozwalają karty pamięci. Na początku XXI z inicjatywy firm: Panasonic, SanDisk i Toshiba [13] zostało założone stowarzyszenie Security Data Card Association. Ma ono na celu standaryzacje promowanie oraz rozwój kart SD (ang Security Data). Odpowiada ono na rosnące zapotrzebowania na energooszczędne, bezpieczne, pojemne i zajmujące mało miejsca pamięci przenośne. Standard SD jest on dostępny w czterech wersjach podzielonych ze względu na pojemność oraz system plików przedstawionych w Tabela Cechy i parametry kart pamięci microSD.7.

*Tabela Cechy i parametry kart pamięci microSD.7. Standardy kart SD [14]*

| Nazwa | Pojemność | System plików |
|---|---|---|
| SD<br>(ang. Security Digital) | Do 2GB | FAT12<br>FAT16 |
| SDHC<br>(ang. Secure Digital High Capacity) | Od 2GB do 32GB | FAT 32 |
| SDXC<br>(ang. Secure Digital eXtended Capacity) | Od 32GB do 2TB | exFAT |
| SDUC<br>(ang. Secure Digital Ultra Capacity) | Od 2TB do 128TB | exFAT |

Można również dokonać podziału ze względu na rozmiar standardowy, mini oraz micro. W szczególności szerokie zastosowanie znalazły kary o wymiarach micro 11 x 15 x 1 mm. Mały rozmiar, szybki transfer danych oraz łatwość implementacji w systemie okazały się przeważającą zaletą w zastosowaniach mobilnych. Łatwość przenoszenia kart pozwala na ich szybką wymianę w wypadku zapełnienia. Dodatkowym atutem jest szybki

odczyt danych na innych systemach mobilnych takich jak smartphone lub laptop przy pomocy odpowiedniego konwertera. Odczyt ten można przeprowadzić poza miejscem działania systemu zbierającego dane na kartę.

Do zapisu i odczytu danych na kartę początkowo była używana magistrala o szybkości 12.5 MB/s. W związku z rozwojem fotografii cyfrowej kolejna wersja interfejsu nazwana High Speed i pozwalał a osiągnąć prędkość do 25 MB/s. Przez szybki rozwój kamer sportowych wzrosło zapotrzebowanie na karty pamięci zdolne zapisywać dużej pojemności pliki w bardzo szybkim czasie. W ten sposób powstał interfejs Ultra High Speed w wersji I używający całego rzędu wyprowadzeń. Pozwoliło to osiągnąć prędkości do 104 MB/s. Obecnie jest to najpowszechniej wykorzystywany. Interfejsy Ultra High Speed w wersji drugiej i trzeciej wykorzystują dwa rzędy wyprowadzeń. Wersje te pracują w dwóch trybach Full-Duplex wykorzystując jeden rząd do komunikacji z karty do hosta, a drugi w odwrotnym kierunku. Tryb Half-Duplex wykorzystuje dwie linie transmisji w jednym kierunku. Najnowszy interfejs SD Express oferuje najszybsze prędkości transferu danych. Wykorzystuje on interfejs PCIe generacji 4 oraz protokół aplikacyjny NVMe Właściwości poszczególnych interfejsów oraz ich właściwości przedstawiono w Tabela Cechy i parametry kart pamięci microSD.8. Interfejsy komunikacyjne kart SD

*Tabela Cechy i parametry kart pamięci microSD.8. Interfejsy komunikacyjne kart SD [15]*

| Interfejs magistrali | Typ karty | Prędkosć [MB/s] |
|---|---|---|
| Default Speed | SD, SDHC, SDXC, SDUC | 12,5 |
| High Speed | | 25 |
| UHS-I | SDHC, SDXC, SDUC | 104 |
| UHS-II | | 156 Full Duplex<br>312 Half Duplex |
| UHS-III | | 312 Full Duplex<br>624 Half Duplex |
| SD Express | | 3940 |

Producenci stosują różne rozwiązania sprzętowe dla swoich produktów. Z tego powodu organizacja SD Association zdefiniowało klasy szybkości kart. Pozwala to użytkownikowi na szybką identyfikacje, który z produktów zapewni jego zapotrzebowanie. Ma to szczególnie ważne znaczenie w urządzeniach nagrywających filmy video.

Konieczny jest sprzęt umożliwiający szybki zapis materiału bez utraty jakości. W Tabela Cechy i parametry kart pamięci microSD.9 zestawiono klasy szybkości.

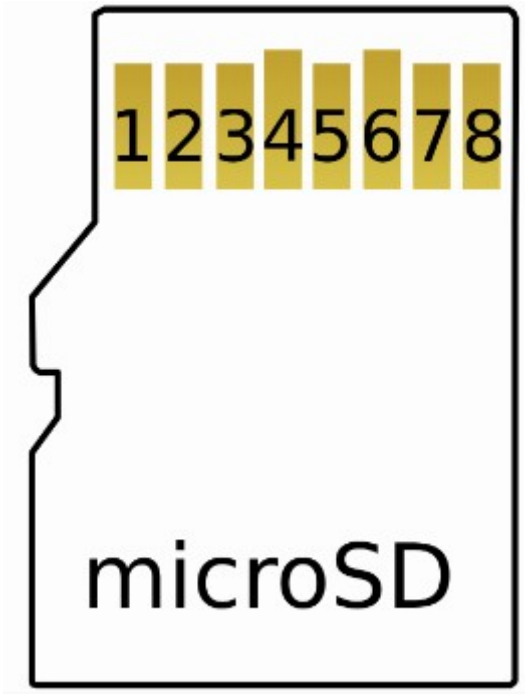
*Tabela Cechy i parametry kart pamięci microSD.9. Klasy szybkości kart micorSD. [16]*

| Prędkość zapisu [MB/s] | Klasa szybkości | | |
|---|---|---|---|
| | Speed Class | UHS Class | Video Speed Class |
| 90 | - | - | V90 |
| 60 | - | - | V60 |
| 30 | - | 3 | V30 |
| 10 | 10 | 1 | V10 |
| 6 | 6 | - | V6 |
| 4 | 4 | - | - |
| 2 | 2 | - | - |

W obecnym wydaniu karty pamięci nie tylko są wykorzystywany jako nośnik danych ale również pamięć dla aplikacji systemowych. Zapotrzebowanie to wynikło ze stałej ilości pamięci smartfonów i stale rosnącej wielkości aplikacji przez nie przetwarzanych. Telefony z systemem Android po wprowadzeniu funkcji Adoptable Storage Device pozwalają na używanie aplikacji na kartach SD. W związku z tym została wprowadzona klasa wydajności A1 pozwalająca edytować i aktualizować dane bez udziału użytkownika. Druga wersja wydajności A2 wprowadziła szybszy dostęp dzięki funkcją kolejkowania instrukcji.

W systemach wbudowanych wykorzystuje się dwa interfejsy do komunikacji z kartą. Tryb SPI korzysta z interfejsu SPI w który wyposażone są powszechnie używane mikrokontrolery. Interfejs składa się z linii adresowej CS, sygnału taktującego SCLK, linii odbierania danych przez urządzenie nadrzędne i nadawana danych przez urządzenie podrzędne MISO oraz linii odbierania danych przez urządzenie podrzędne i wysyłania danych przez urządzenie nadrzędne MOSI. Obecnie są też stosowane wersje Quad-SPI wykorzystujące cztery linie transmisyjne. Drugi z interfejsów nazywany SD może być wykorzystywany w trybie 1-bitowym w którym standardowo karta jest uruchamiania oraz 4-bitowy w który można przełączyć po uruchomieniu. Tryb SD pracuje w częstotliwości 50 MHz natomiast SPI maksymalnie do 40 MHz. Dzięki temu że tryb SD wykorzystuje 4 linie danych szybkość działania w tym trybie jest czterokrotnie szybsza. Karty microSD posiadają 8 wyprowadzeń przedstawionych wraz z numeracja na Rysunek Cechy i

parametry kart pamięci microSD.6. Ich zastosowanie przedstawiono w Tabela Cechy i parametry kart pamięci microSD.10 ta dla obu trybów pracy.




*Rysunek Cechy i parametry kart pamięci microSD.6. Rozmieszczeni i numeracja wyprowadzeń w karcie micro SD*

*Tabela Cechy i parametry kart pamięci microSD.10. Przeznaczenie wyprowadzeń w karcie microSD dla dwóch interfejsów. [17]*

| Numer wyprowadzenia | SD | SPI |
|---|---|---|
| 1 | DAT2 | - |
| 2 | CD/DAT3 | CS |
| 3 | CMD | MOSI |
| 4 | VDD | VDD |
| 5 | CLK | SCLK |
| 6 | GND | GND |
| 7 | DAT0 | MISO |
| 8 | DAT1 | - |

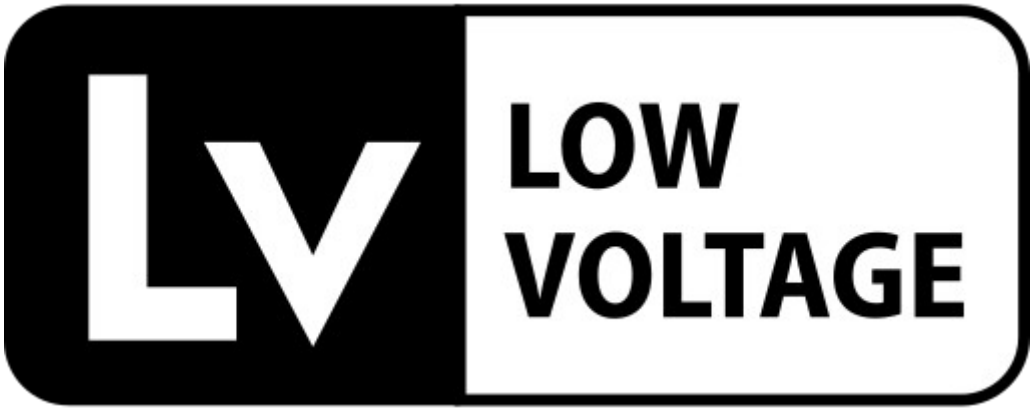
Do komunikacji z kartą stosowany jest zestaw komend w trybie SPI. Pozwalają one na między innymi zainicjalizowanie karty sprawdzenie jej diagnozę oraz ewentualny odczyt i zapis danych. Znaczenie komend oraz ich format przedstawiono w Tabela Cechy i parametry kart pamięci microSD.11.

*Tabela Cechy i parametry kart pamięci microSD.11. Polecenia karty w trybie SPI [18]*

| Indeks | Argument | Potwierdzenie | Opis |
|---|---|---|---|
| CMD0 | - | R1 | Zerowanie karty |
| CMD1 | - | R1 | Rozpoczęcie inicjalizacji |
| CMD9 | - | R1 | Przesłanie rejestru CSD |
| CMD10 | - | R1 | Przesłanie rejestru CID |
| CMD12 | - | R1b | Zakończenie transmisji podczas odczytu |
| CMD13 | - | R2 | Przesłanie statusu |
| CMD16 | [31:0] długość bloku | R1 | Ustawienie długości bloku do zapisu/odczytu |
| CMD17 | [31:0] adres | R1 | Odczytanie jednego bloku |
| CMD18 | [31:0] adres | R1 | Ciągłe odczytywanie bloków do wysłania komendy CMD12 |
| CMD24 | [31:0] adres | R1 | Zapis bloku pamięci długości określonej przez CMD16 |
| CMD25 | [31:0] adres | R1 | Ciągłe odczytywanie bloków do odebrania bitu „stop transsmision command” |
| CMD27 | - | R1 | Zmiana bitów rejestru CSD |
| CMD28 | [31:0] adres | R1b | Pierwszy adres bloków objętych protekcją |
| CMD29 | [31:0] adres | R1b | Ostatni adres bloków objętych protekcją |
| CMD30 | [31:0] zapis adresu protekcji | R1 | Stan bitów protekcji |
| CMD32 | [31:0] adres | R1 | Adres pierwszego bloku do skasowania |
| CMD33 | [31:0] adres | R1 | Adres ostatniego bloku do skasowania |
| CMD38 | [31:0] bez znaczenia | R1b | Kasowanie grupy bloków zaznaczonych przez komendy CMD32 i CD33 |
| CMD55 | [31:0] | R1 | Następna komenda typu ACMD |
| CMD56 | [31:0] bit[0] RD/WR | R1 | |
| CMD58 | - | R3 | Oczytanie rejestru OCR |
| CMD59 | [31:0] bez znaczenia, bit [0] CRC | R1 | Bit[0] Włącz/wyłącz CRC |

W energooszczędnych systemach mikroprocesorowych istotną role odgrywa pobór prądu czytnika karty pamięci jak i samej karty. Trzema najistotniejszymi czynnikami jest pobór prądu przez kartę w czasie inicjalizacji, zapisu oraz odczytu. Kolejny ze standardów SD Association definiuje karty do urządzeń mobilnych z zasilaniem

bateryjnym. Jako napięcie logiczne przyjęto 1,8 V. Charakteryzuje się mniejszymi zakłóceniami elektromagnetycznymi oraz szybszym czasem narostu oraz opadania. Na przedstawiono oznaczenie kart w standardzie Low Voltage Signaling.




*Rysunek Cechy i parametry kart pamięci microSD.7. oznaczenie krat pamięci w standardzie Low Voltage Signaling*

## 3.1. System plików FAT

Przechowywanie danych w pamięciach nieulotnych wymaga standaryzacji. Jednym z najpopularniejszych systemów plików jest FAT (ang. File Allocation Table). System organizacji pliku polega na wyznaczeniu z pamięci obszaru w postaci tablicy informującego między innymi o rozmieszczeniu i rozmiarze przechowywyanych plików. System w drodze rozwoju technologicznego i pojemności nośników danych ewaluował. W Tabela Cechy i parametry kart pamięci microSD.12. przedstawiono różnice między poszczególnymi typami. Nazewnictwo oparte jest o numeracje po nazwie FAT. Liczba ta odpowiada liczbie bitów na których można zakodować klastrów.

*Tabela Cechy i parametry kart pamięci microSD.12. Typy systemu FAT*

| | FAT12 | FAT16 | FAT32 | exFAT (FAT64) |
|---|---|---|---|---|
| Maksymalna wielkość pliku [GB] | 0,016 | 2 | 4 | 1073741824 |
| Maksymalna liczba klastrów | 4077 | 65517 | 268435445 | 4294967295 |
| Wielkość klastra [kB] | 12 | 16 | 32 | 64 |

Proces ewolucji systemu trwa od końca lat 70 XX wieku i trwa do dzisiaj Najnowsza typ exFAT nazywany również FAT64 został wprowadzony w 2006 roku. Pozwala on na obsługę nośników pamięci dużo większych niż są obecnie powszechnie używane. W systemach embedded najczęściej stosowanymi typami jest FAT16 oraz FAT32.

Najmniejszą ilością danych jest sektor. Jest to najmniejsza ilość danych przekazaną do odczytu lub zapisu. W większości typów jest to 512 bajtów. Zapisywane pliki składają się z klastrów a te z sektorów. Każdy z klastrów jest przypisany tylko do jednego pliku. Jeśli plik zawiera tylko niewielką część klastra pozostała jego część mimo że jest nieużywana jest do niego przypisana. Nośniki danych posiadają kilka zarezerwowanych sektorów przedstawionych na Rysunek Cechy i parametry kart pamięci microSD.8.


*Rysunek Cechy i parametry kart pamięci microSD.8. Podział pamięci nośnika w systemie FAT*

Obszar zarezerwowany to pierwszy sektor zawierający informacje na temat nośnika. Przechowuje między innymi informacje o typie FAT, ilości sektorów jednego klastra oraz rozmiarze pojedynczego sektora. Może znajdować się również program umożliwiający wyczytanie systemu operacyjnego. Następnym obszarem jest tablica alokacji klastrów. Informuje o rozmieszczeniu ich w nośniku oraz jak są wykorzystywane. Zaraz za tablicą znajduje się jej kopia. Katalog główny to pierwszy katalog utworzony w obszarze danych składający się z 32 sektorów. Za nim znajduje się przestrzeń do wykorzystania.

## 3.2. Producenci kart microSD

Wiele firm na rynku oferuje kart microSD o szerokim zastosowaniu i w różnych standardach wymienionych w poprzednich rozdziałach. Jednym z największych jest firma SanDisk mająca w swoich szeregach aż sześć serii z dedykowanym zastosowaniem. Zestawienie możliwości poszczególnych serii przedstawiono w Tabela Cechy i parametry kart pamięci microSD.13. Pierwsza serii Micro jest dedykowana urządzeniom takim jak smartfon i tablet z systemem operacyjnym Android. Kolejna z serii Ultra jest powiększona o dodatkową pojemność. Micro Gaming dedykowana jest konsolą mobilnym firmy Nintendo. Jej zaletą jest zwiększona szybkość wymiany danych. Wysoką wytrzymałością cechuje się kolejna z serii dedykowana rejestratorom samochodowym i kamerą monitoringu przemysłowego. Ostatnie dwie serie znajdują zastosowanie w kamerach sportowych zapisujących z wysoką prędkością obraz rozdzielczości nawet 8K.

*Tabela Cechy i parametry kart pamięci microSD.13. Oferta kart microSD firmy SanDisk*

| Seria | SanDisk | | | | | | |
|---|---|---|---|---|---|---|---|
| | Standard pamięci | System plików | Prędkość odczytu [MB/s] | Prędkość zapisu [MB/s] | Magistrala | Klasa prędkości | Pojemność [GB] |
| Micro | SDHC/SDXC | FAT32 | 100 | - | UHS I | V10 | do 128 |
| Micro Ultra | SDHC/SDXC | FAT32 | 120 | - | | V10 | do 512 |
| Micro Gaming | SDXC | exFAT | 100 | 60 | | V30 | do 256 |
| Micro Endurance | SDXC | exFAT | 100 | | | V10 | do 256 |
| Micro Extreme | SDHC/SDXC | exFAT | 100 | 60 | | V30 | do 100 |
| Micro Extreme Pro | SDHC/SDXC | exFAT | 100 | 90 | | V30 | do 100 |

Firma Samsung oferuje cztery rodziny kart microSD:

- EVO Plus – do szerokiego zastosowania zapisu danych,
- PRO Endurance – dla rejestratorów samochodowych oraz monitoringu,
- EVO – wzmocnionej wytrzymałości mechanicznej,
- EVO Select - do kamer sportowych,

Z racji że produkuje również smartfony tablety oraz inne urządzenia mobilne, karty microSD są dedykowane ich własnym produktom.

Ostatnim z porównanych najpopularniejszych producentów kart microSD jest Kingston. W swojej ofercie posiada pięć rodzin kart:

- Canvas Select Plus – dla urządzeń z systemem Android oraz aparatów fotograficznych,
- Canvas GO! Plus – dla kamer sportowych, tworzenia filmów w rozdzielczości 4K,
- High-Endurance – dla rejestratorów samochodowych oraz monitoringu

- Klasa przemysłowa – dla zastosowań w środowisku przemysłowym o szerokim spektrum zmiany temperatur,
- Canvas React Plus – dla filmów w rozdzielczości 8K oraz do wykorzystania w bezzałogowych statkach powietrznych.

Ostatnia z serii posiada jako jedyna z wymienionych dwie szyny wyprowadzeń z zaimplementowanym interfejsem UHS-II. Pozwala to na odczyt z prędkością 285 MB/s oraz zapis 165 MB/s.

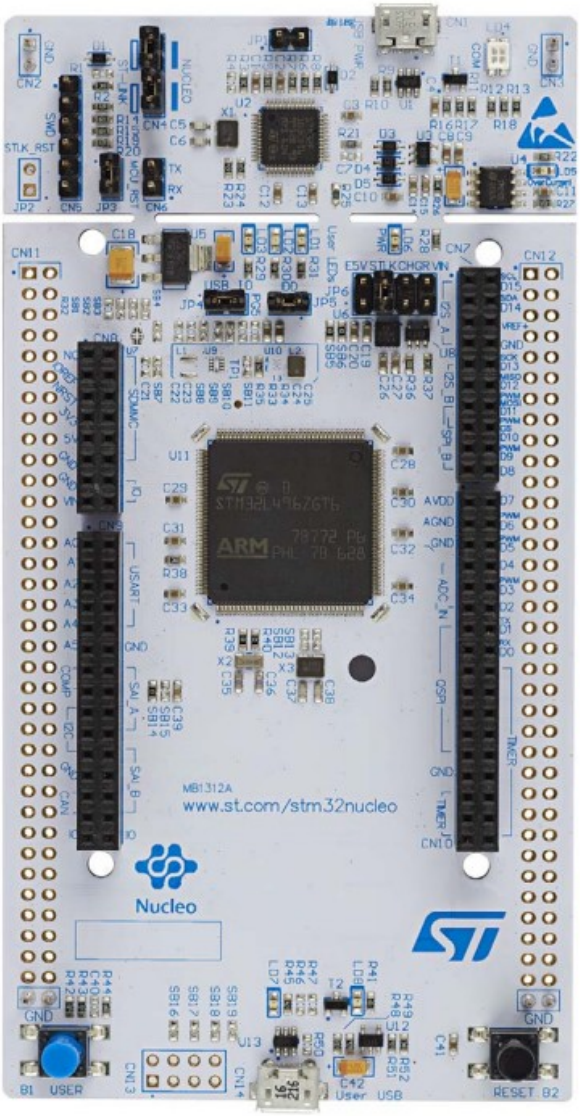
# 4. Projekt systemu mikroprocesorowego

Zadaniem systemu mikroprocesorowego jest zapisywanie szyfrowanych danych na karcie microSD przy jak najmniejszym zużyciu energii elektrycznej. Bazą systemu jest mikrokontroler z rdzeniem posiadającym mechanizmy zapewniające niski pobór energii. Sensorem pomiarowym jest trójosiowy akcelerometr. Zastosowanie tego sensora pozwoli na szybki zapis dużych ilości danych. Poprzez poruszanie czujnikiem można w łatwy sposób wpłynąć na różnorodność wyników pomiarowych. Kolejnym elementem systemu są czytniki kart pamięci. Działają one w dwóch różnych interfejsach w celu porównania ich ze sobą bez konieczności zmiany konfiguracji systemu. Komunikacja z mikrokontrolerem odbywa się na dwa sposoby. Pierwszym jest dedykowana klawiatura z sześcioma przyciskami oraz wyświetlaczem. Na ekranie wyświetlane jest menu pozwalające wybrać na wybór karty pamięci do zapisu, metody zapisu oraz wysłania komendy o rozpoczęciu lub zakończeniu pomiaru. Drugim sposobem jest komunikacja przy pomocy magistrali UART. W ten sposób można w łatwy sposób wysłać polecenia do mikrokontrolera oraz odczytać zawartość polików z wynikami na karcie microSD. Całość została zaimplementowana w płytce rozszerzeniowej celem łatwego przeprowadzenia pomiaru.

## 4.1. Wybrany mikrokontroler

Po porównaniu mikrokontrolerów w punkcie 2.5. został wybrany mikrokontroler STM32L4A6ZG. Produkty firmy STMicroelectronics są powszechnie używane na świecie oraz posiadają szeroki zasób wsparcia zarówno sprzętowego jak i programowego. Spośród porównanych pozycji osiągnął on bardzo dobre wyniki w benchmarku EMBC i posiada korzystne właściwości związane z niskim poborem energii. Mikrokontrolery z rodziny STMU5, które osiągnęły najlepsze wyniki w momencie projektowania systemu nie były dostępne na rynku. Mikrokontroler posiada aż 144 wyprowadzenia pozwalające na

swobodne prace prototypowe. W na płytce MB1312 przedstawionej na Rysunek Projekt systemu mikroprocesorowego.9 oferowanej przez producenta znajdują się również trzy diody LED, dwa przyciski oraz USB-C możliwe do wykorzystania w dalszych pracach nad nośnikami danych. W łatwy sposób można wprowadzić mikrokontroler w jeden z trzech stanów niskiego poboru energii. Zawiera dwa interfejsy komunikacyjne SD oraz SPI. Jest to istotne ze względu na możliwość łatwego zaimplementowania modułów peryferyjnych do obsługi kart microSD. Dzięki temu można było przeprowadzić pomiary dla obydwóch trybów bez konieczności zmiany platformy. Platforma wspiera sprzętowe rozwiązania kryptograficzne pozwalające na implementacje szyfru AES w wielu trybach z kluczem 128 oraz 256 bitowym. Pozwala również na wykorzystanie algorytmów funkcji skrótu; HMAC, MD5, oraz SHA.

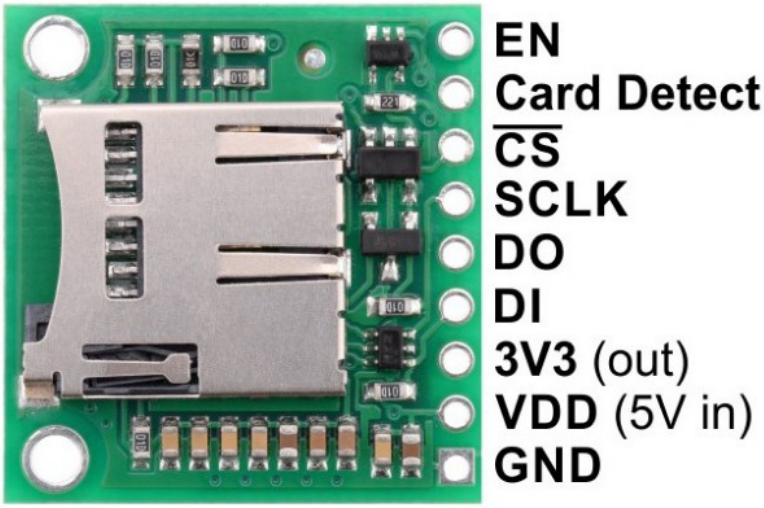



*Rysunek Projekt systemu mikroprocesorowego.9. Płytka deweloperska MB1312.*

## 4.2. Moduły peryferyjne

Moduły peryferyjne mają za zadanie realizację założeń pracy w postaci zbierania parametrów fizycznych z czujnika na jednym z dwóch czytników karty pamięci micro-SD. Ułatwiają one komunikację człowieka z urządzeniem przez wyświetlacz oraz przyciski. Dodatkowo pozwala to na łatwiejsze przeprowadzenie pomiarów w różnych warunkach. System zawiera łącznie pięć modułów peryferyjnych.

Pierwszym z nich jest czytnik kart micro-SD firmy Pololu przedstawiony na Rysunek Projekt systemu mikroprocesorowego.10. Czytnik jest ustawiony w trybie transmisji SPI a jego wyprowadzenia przedstawiono w Tabela Projekt systemu mikroprocesorowego.14. Czytnik posiada wbudowany detektor karty. Jest to kontakt który jest zwierany do linii zasilającej gdy kata jest włożona do czytnika. Powoduje to zmianę linii CD na wysoki co informuje mikrokontroler o możliwości podjęcia działań związanych z inicjalizacją karty.




*Rysunek Projekt systemu mikroprocesorowego.10. Czytnik kart SPI firmy Pololu.*

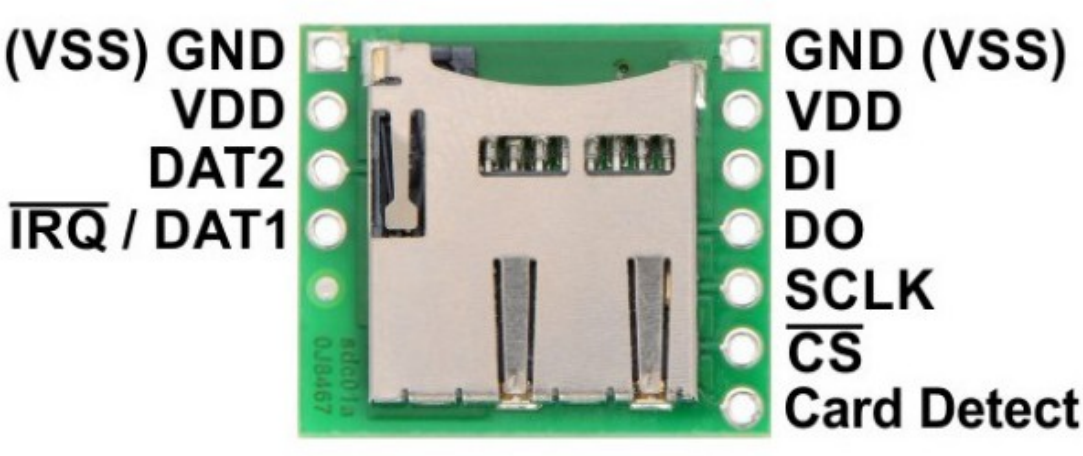
Ten czytnik kart posiada napięcie zasilania 5 V. Zasilanie kart pamięci to najczęściej napięcie 3,3 V, dlatego na karcie na module jest umieszczony stabilizator napięcia.

*Tabela Projekt systemu mikroprocesorowego.14. Wyprowadzenia czytnika kart SPI*

| Pin | Zastosowanie |
|---|---|
| EN | Linia uruchamiająca stabilizator napięcia |
| CD | Linia detekcji karty w czytniku. Stan wysoki karta obecna, stan niski karata nieobecna |

| | |
|---|---|
| CS | Linia adresowa SPI |
| SCLK | Linia zegara taktującego SPI |
| DO | Linia wyjściowa MISO interfejsu SPI |
| DI | Linia wejściowa MOSI interfejsu SPI |
| 3V3 | Wyjściowe napięcie 3,3 V |
| GND | Masa sygnałów |
| VDD | Napięcie zasilania 5 V |

Drugi z czytników przedstawiony na Rysunek Projekt systemu mikroprocesorowego.11 obsługuje zarówno tryb komunikacji SPI jak i SD. W Tabela Projekt systemu mikroprocesorowego.15 przedstawiono wyprowadzenia oraz ich zastosowania w obydwóch interfejsach. W tym czytniku również zaimplementowana jest możliwość detekcji karty w czujniku. Jednak w przeciwieństwe do poprzedniego, stan niski zwarty do masy informuje o obecności karty w czytniku.



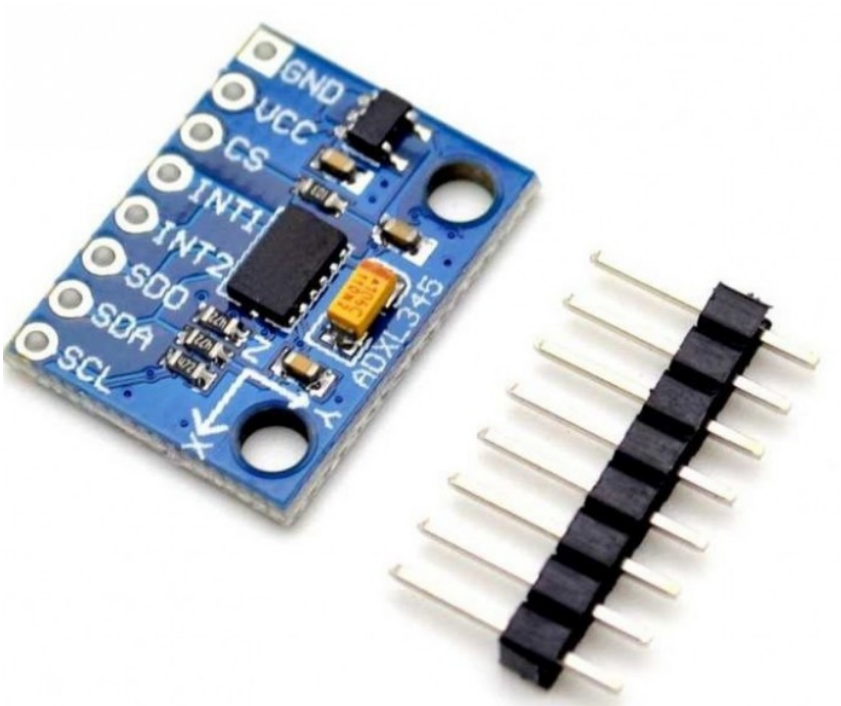

*Rysunek Projekt systemu mikroprocesorowego.11. Czytnik kart firmy Pololu z interfejsami SPI, SD*

W projekcie skonfigurowana czytnik do pracy w interfejsie SD. Jest to obecnie najszerzej stosowany interfejs komunikacyjny w urządzeniach mobilnych z powodu szybkości działania. Posiada on dwa tryby z wykorzystaniem jednej lub czterech linii transmisji danych. Podobnie jak SPI potrzebuje sygnału zegarowego z urządzenia nadrzędnego.

*Tabela Projekt systemu mikroprocesorowego.15. Wyprowadzenia czytnika kart z interfejsami SD i SPI*

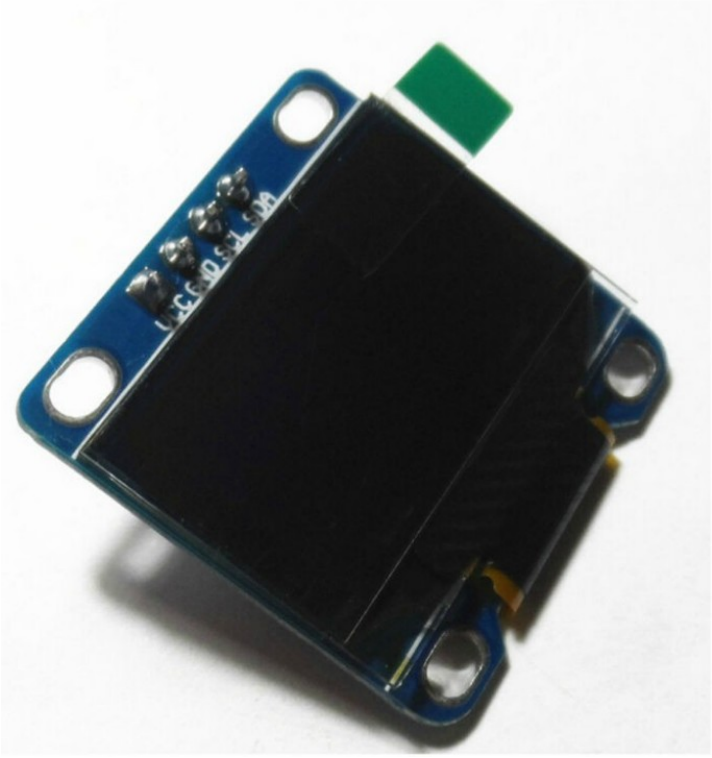
| PIN | Zastosowanie | |
|---|---|---|
| | SPI | SD |
| GND | Masa sygnałów | |
| VDD | Zasilanie 3,3 V | |
| CD | Detekcja karty | |
| DI | Wejście danych MOSI | Linia CMD |
| DO | Wyjście danych MISO | Linia Data0 |
| SCLK | Lina zegarowa | |
| CS | Linia adresowa | Linia Data3 |
| IRQ/ DAT1 | Linia przerwania | Linia Data1 |
| DAT2 | - | Linia Data2 |

Elementem pozyskiwanych danych w systemie jest trójosiowy akcelerometr ADXL345 przedstawiony na Rysunek Projekt systemu mikroprocesorowego.12. W jednym cyklu próbkowania można pobrać trzy dane na temat orientacji sensora w przestrzeni. Posiada on dwa interfejsy komunikacyjne SPI oraz I2C. Do komunikacji z systemem został wybrany interfejs SPI z powodu większej szybkości transmisji danych. Sama komunikacja SPI może być używanych w dwóch trybach 3 i 4 [19]. Tryb z trzema liniami używa jednej linii do nadawania i odbierania danych. W drugim trybie linie wejścia i wyjścia są rozłączne na dwie osobne. Maksymalna prędkość pobierania danych to 3200 Hz. Dane pobierane są z wysoką rozdzielczością od 10 do 13 bitów. Akcelerometry są szeroko wykorzystywane w urządzeniach z zasilaniem bateryjnym jak smartfony i gimbale.




*Rysunek Projekt systemu mikroprocesorowego.12. Akcelerometr ADXL345 z interfejsem komunikacyjnym SPI*

Monochromatyczny wyświetlacz OLED ze sterownikiem SDD1306 ukazany na Rysunek Projekt systemu mikroprocesorowego.13 pełni role komunikacji użytkownika z systemem. Matryca o wielkości 0,96 cala jest monochromatyczna. Posiada rozdzielczość 128x64 pikseli w kolorze niebieskim. Komunikacja z systemem odbywa się za pomocą magistrali I2C. Na ekranie wyświetlane jest menu główne systemu pozwalające użytkownikowi na wybranie metody zapisu na jedną z dwóch dostępnych kart pamięci. Informuje również o szczegółach trwającego zapisu.



*Rysunek Projekt systemu mikroprocesorowego.13. Wyświetlacz OLED ze sterownikiem SDD1306*

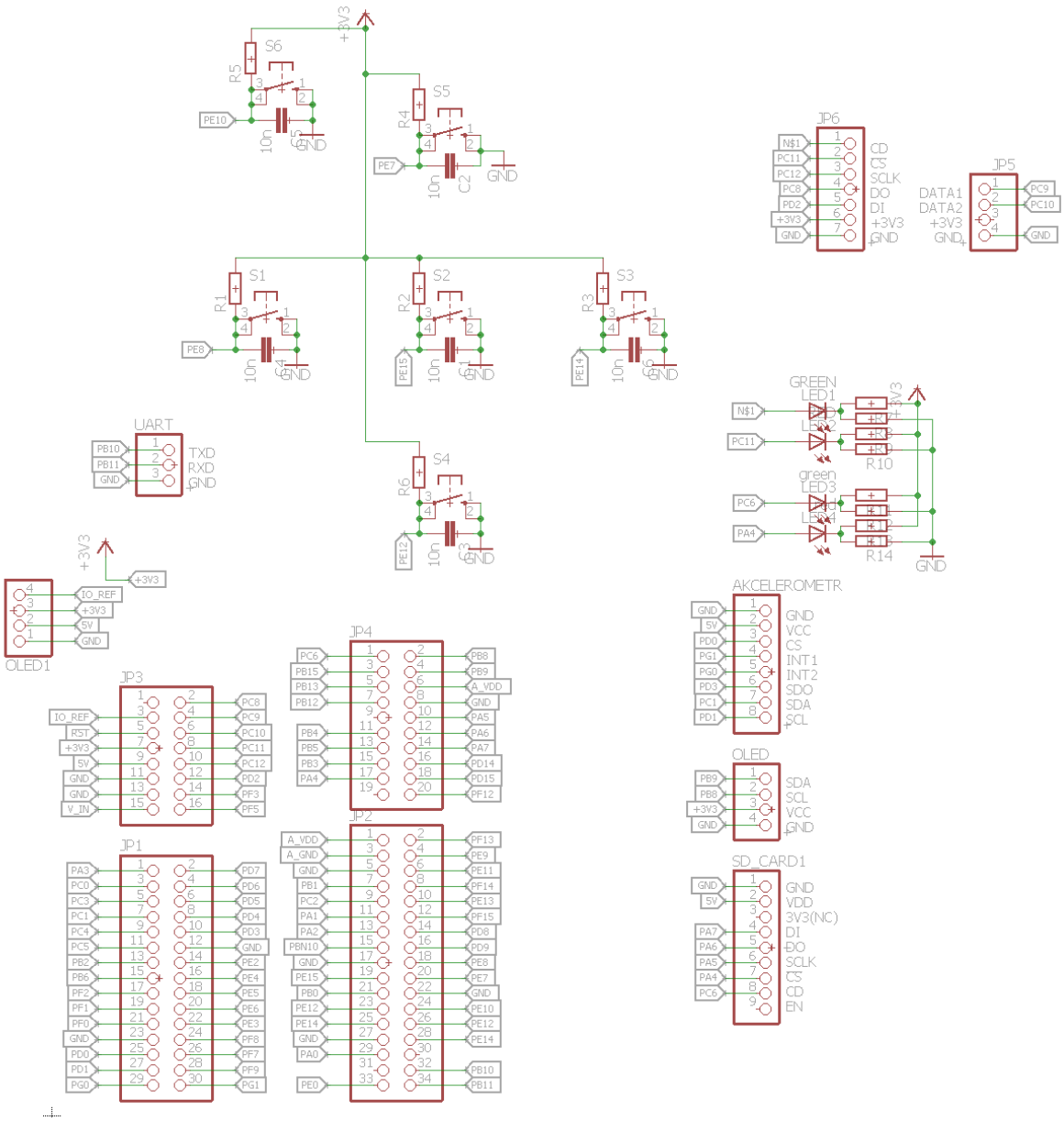
## 4.3. Projekt i wykonanie płytki rozszerzeniowej

Z powodu złożoności układu elektrycznego projektowanie oraz testowanie układu było znacznie utrudnione na płytce stykowej, zdecydowano się na zaprojektowanie oraz zbudowanie płytki rozszerzeniowej dla układu. Jest to płytka typu „shield” stanowiąca nakładkę na płytkę mikrokontrolera STM32L4A6ZG. Znajdują się na niej moduły peryferyjne opisane w poprzednim rozdziale. Została zaprojektowana w programie Eagle 7.4.0. Schemat został zobrazowany na Rysunek Projekt systemu mikroprocesorowego.15. Dodatkowo zostały umieszczone cztery diody. Dwie czerwone diody informują użytkownika o wprowadzeniu w stan niski linii adresowej konkretnego czytnika karty pamięci. Oznacza to rozpoczęcie transmisji na tej karcie pamięci. Zielone diody wykorzystywane są do sygnalizacji detekcji karty w czytniku kart pamięci. Do sterowania menu na wyświetlaczu zostało dołączone sześć przycisków. Ich rozmieszczenie odpowiada odpowiednio funkcją:

- Góra;
- Dół;
- Prawo;
- Lewo;
- Zatwierdź;

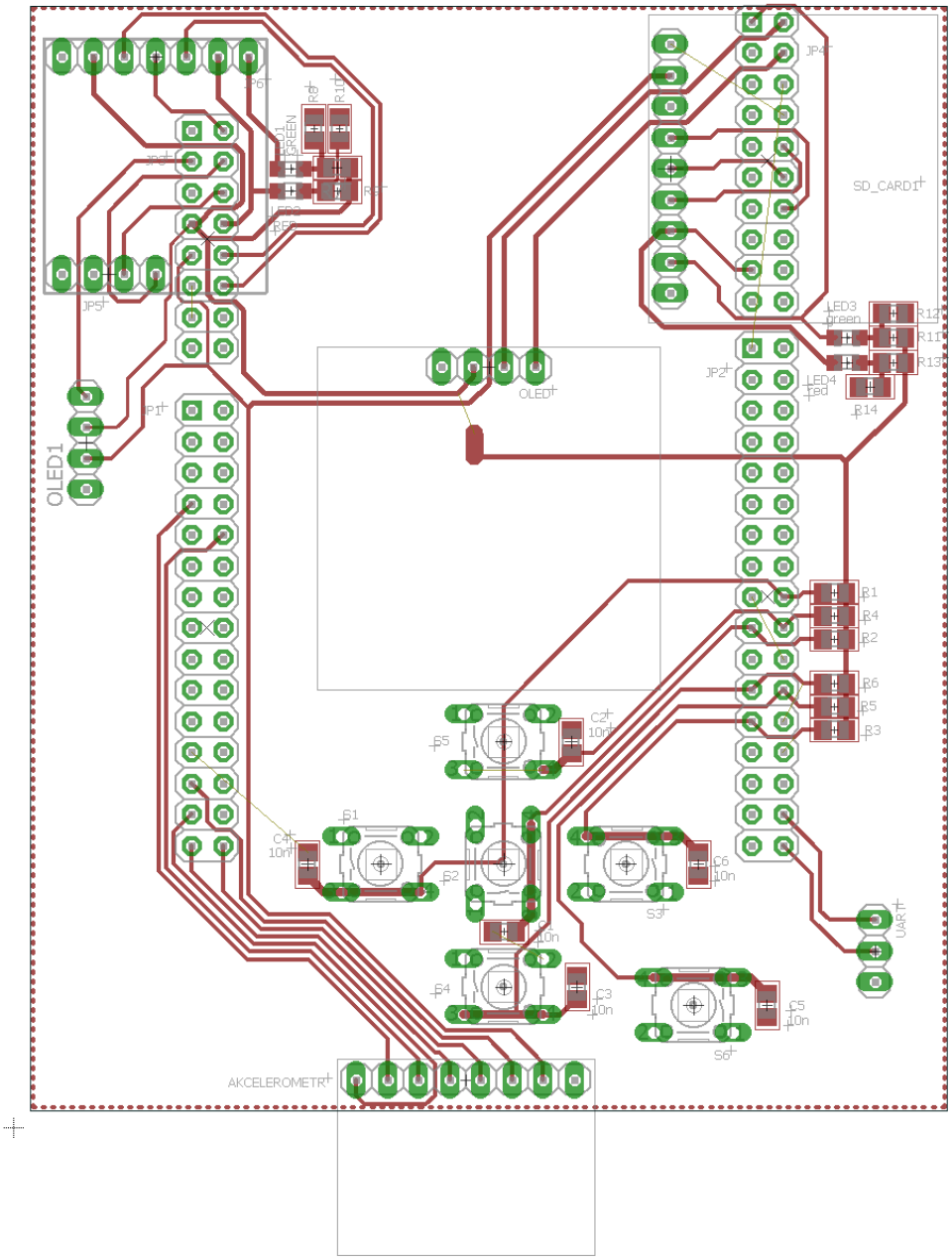
- Cofnij.

Każdy z nich jest wyposażony w dodatkowe rezystory oraz kondensatory eliminujące zjawisko drgania styków, uniemożliwiając wielokrotne wykonanie instrukcji wybranych przez użytkownika. Mogło to prowadzić to wykrycia na porcie kilku nadmiarowych impulsów. Wyprowadzone zostały również piny zasilania 3,3 V, 5 V, masa sygnałowa oraz wyprowadzenie na zewnętrzne zasilanie 5 V. Dzięki nim można przeprowadzić pomiary zużycia energii. Mogą one zostać wykorzystane również do zasilania dodatkowych układów dołączonych do systemu w przyszłości. Ostatnim z wymienionych rozszerzeń jest wyprowadzenie transmisji sygnałów interfejsu UART. Służy on do rozszerzenia komunikacji użytkownika np. z komputerem. Przez ten interfejs są wysyłane rozszerzone dane dotyczące przeprowadzanych testów, zawartości plików umieszczonych na kartach micro-SD.




*Rysunek Projekt systemu mikroprocesorowego.14. Schemat elektryczny płytki rozszerzeniowej*

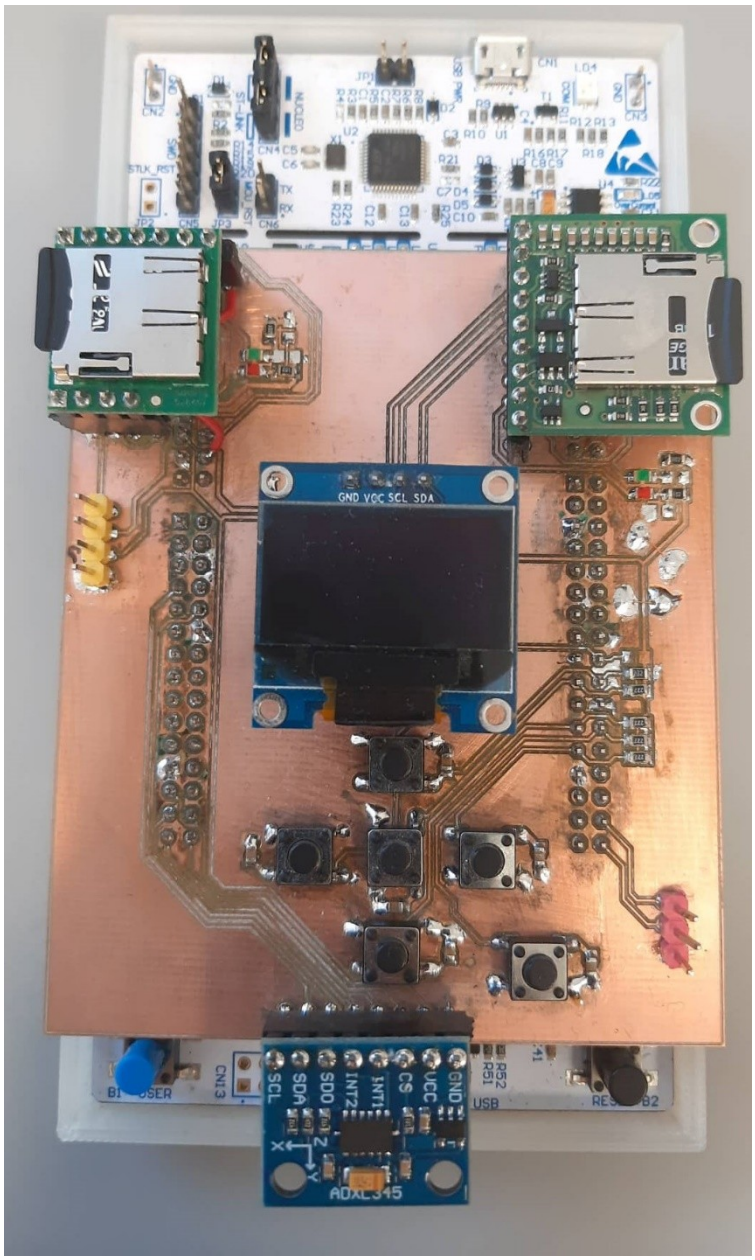
Rozmieszczenie elementów systemu zostało przedstawione Rysunek Projekt systemu mikroprocesorowego.15. Jest to ściśle powiązane z rozmieszczeniem dostępnych wyprowadzeń mikrokontrolera. W lewym górnym rogu znajduje się czytnik kart pamięci wykorzystujący interfejs SD. Po jego prawej stronie umieszczone zostały diody sygnalizujące jego stan pracy. Pod czytnikiem znajdują się wyprowadzenia służące pomiarowi zużycia energii. W prawym górnym rogu umieszczony został czytnik kart pamięci z interfejsem SPI i sygnalizacyjnymi diodami. W środkowej części znajduje się wyświetlacz OLED ze ścieżkami do interfejsu I2C oraz przyciski sterujące. Pod nimi znajdują się wyprowadzenia akcelerometru i ścieżki sterujące SPI. W prawym dolnym rogu znajdują się dwa piny RX oraz TX magistrali UART.




*Rysunek Projekt systemu mikroprocesorowego.15. Projekt płytki rozszerzeniowej programu Eagle*

Płytka została wytworzona z laminatu miedzianego FR4 o grubości to 1,5 minimetra. Z powodu dużej ilości miejsca została wykonana w technologii jednostronnej. Ścieżki zostały wytworzone za pomocą frezarki mechanicznej. Zastosowanie „rozlanej masy" można uzasadnić długotrwałym procesem obróbki mechanicznej usuwania nadmiaru miedzi. Na płytce zostały przylutowane przy pomocy lutownicy kolbowej i cyny bezołowiowej części składowe. Elementy pasywne rezystory kondensatory oraz diody zostały wykorzystane w technologii SMD. Wyprowadzenia do połączenia z mikrokontrolerem, akcelerometrem, czytnikami pamięci, interfejsu UART oraz wyświetlacza stanowią elementy przewlekane. Finalne wykonanie oraz umiejscowienie na

mikrokontrolerze zostało przedstawione na Rysunek Projekt systemu mikroprocesorowego.16




*Rysunek Projekt systemu mikroprocesorowego.16. Wykonana płytka rozszerzeń*

# 5. Opracowanie algorytmów do rejestracji i szyfrowania danych

Sensorem gromadzącym dane w systemie jest moduł akcelerometru trójosiowego ADXL345. Dane po pobraniu z akcelerometru są gromadzone w buforze do zapisu. Posiada on wewnętrzny licznik zliczający ilość zapisów do bufora. Gdy licznik jest równy jego wielkości następuje zapis na kartę. W przypadku szyfrowania każda z wartości w buforze zostaje zaszyfrowana przed zapisem na kartę micro-SD. Odczytane wartości nie są

przeliczane na jednostki w celu szybszego zbierania danych. Surowe dane mogę zostać przeliczone i odszyfrowane w innym systemie niemającym rygoru energooszczędności. Metoda ta pozwala na większość niezawodność zbieranych danych. Zastosowanie kryptografii ma na celu zapewnienie bezpieczeństwa przetwarzanych danych. Do tego celu mogą być wykorzystane algorytmy w odmiennych trybach oraz z różną długością klucza. Zadaniami algorytmów są między innymi:

- Usługi dotyczące integralności danych tak by uniemożliwić ich modyfikacje;
- Usługi poufności ograniczające dostęp do danych jedynie osobą do tego powołanym;
- Ustalenie odbiorcy oraz nadawcy;

Szyfrowanie jest realizowane za pomocą kluczy niejawnych zmieniających tekst jawny w postać zaszyfrowaną. Przywrócenie szyfrogramu do postaci czytelnej odbywa się w kryptografii symetrycznej za pomocą tego samego klucza. Do szyfrowania danych został wykorzystany szyfr blokowy AES ( ang. Advanced Encryption Standard). W algorytmie są wykorzystywane są klucze o długości 128, 192, 256 bitów. Pojedynczy blok wejściowy oraz wyjściowy jest wielkości 128 bitów. Liczba rund jest uzależniona od długości klucza. Dla 128 bitowego klucza jest to 10 rund natomiast dla 256 bitowego 14 rund.

## 5.1. Algorytm zapisu danych z akcelerometru

Wartości odczytywane z akcelerometru są rozdzielczości od 10 do 13 bitów. Następnie są formatowane i uzupełnione do 16 bitów by można było zapisać odczyt za pomocą typu języka programowania C *uint16_t*. Pozwala to na zapis wartości całkowitej dziesiętnej do 65535 [20]. Może ona być obniżona do prędkości 12,5 Hz lub podwyższona do 3200 Hz. Wyniki pobierane są z częstotliwością 1000 próbek na sekundę z trzech osi. Do odliczania czasu służy wewnętrzny licznik mikrokontrolera generujący przerwanie z częstotliwością 1000 Hz. Zapis jest kontynuowany do momentu przerwania przez użytkownika lub upływie zadanego czasu pomiaru. Czas pomiaru jest ustawiany za pomocą komendy oraz przeliczany z licznika SysTick. Licznik ten jest inkrementowany co 1ms. Wyznaczenie wzoru na prędkość pobieranych danych z akcelerometru została przedstawiona na Równanie Opracowanie algorytmów do rejestracji i szyfrowania danych.1.

$$V = a \cdot f \cdot c$$

*Równanie Opracowanie algorytmów do rejestracji i szyfrowania danych.1. Prędkość odczytu danych z akcelerometru.*

gdzie:

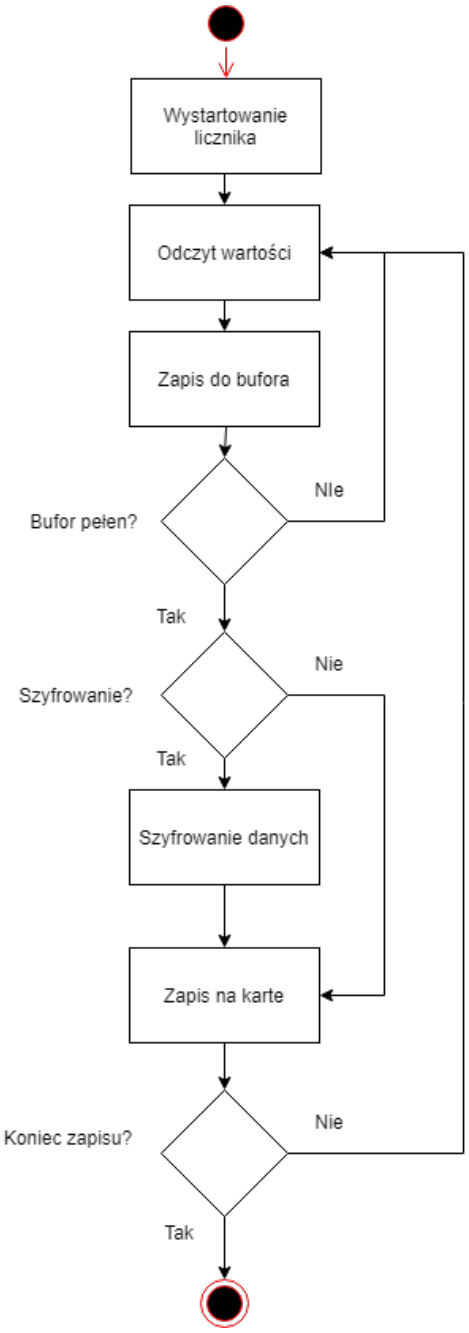
V – prędkość zapisu [B/s]

f – częstotliwość próbkowania w [Hz]

a – liczba osi z których są pobierane dane

c – rozmiar typu zmiennej [B]

W przypadku zbierania danych z trzech osi z prędkością 1000 Hz do zmiennej typu uint_16t otrzymujemy wynik 48000 bitów na sekundę. Taka wartość pozwala na obserwacje w wysokiej rozdzielczości zmian wywołanych przemieszczaniem się akcelerometru oraz zaobserwowania drgań własnych w stanie spoczynku. Zapis takich ilości danych na kartę micro-SD pozwoli na efektywne jej zapełnienie w niedużym czasie.. Całość algorytmu zapisu przedstawiono na




*Rysunek Opracowanie algorytmów do rejestracji i szyfrowania danych.17. Procedura zapisu na kartę pamięci.*

## 5.2. Biblioteka kryptograficzna w mikrokontrolerach STM32

Firma STMicoelectronics udostępnia bibliotekę kryptograficzną pozwalającą na implementację szerokiej gamy algorytmów szyfrowania, funkcji skrótu, uwierzytelnienia wiadomości, podpisu cyfrowego oraz generacji liczb losowych. Wśród algorytmów szyfrowania znajdują się mechanizmy szyfrowania AES. Został on przyjęty NIST jako standard FIPS-197. Tryby pracy możliwe do wykorzystania w mikrokontrolerach STM32L4 to [21]:

1. ECB – jest to najprostszy tryb w którym wiadomość jest dzielona na bloki, a następnie każdy z nich jest zaszyfrowany oddzielnie. Ten tryb nie zapewnia ukrywania wzorców danych. Używanie tego trybu jest niezalecane dla więcej niż jednego bloku danych.
2. CBC – tryb szyfrowania łańcucha bloków. Każdy z bloków tekstu jawnego jest poddany operacji XOR przez zaszyfrowany poprzedni blok. Aby wiadomość była unikatowa należy w pierwszej kolejności dodać wektor inicjalizujący IV o długości bloku.
3. CTR – tryb licznikowy który zmienia szyfr blokowy w strumieniowy. Szyfruje kolejne wartości licznika następnie przeprowadza operacje XOR na jawnym lub zaszyfrowanym strumieniu danych.
4. GCM GMAC – jest oparty o tryb licznikowy dodatkowo zapewniając autentykacje wiadomości.
5. CMAC – jest to kod uwierzytelnienia liczony za pomocą funkcji skrótu zaszyfrowanej symetrycznym kluczem blokowy.

Ponadto dostępne są algorytmy szyfrowania ARC4, DES, 3DES w trybach ECB oraz CBC. Dodatkowo można użyć funkcji skrótu HMAC:

- MD5;
- SHA-1;
- SHA-224;
- SHA-256;
- SHA-384;
- SHA-512.

W projekcie został wykorzystany tryb szyfrowania AES CTR ze 128 bitowym kluczem. Do wykorzystania tego trybu konieczny jest wygenerowanie klucza oraz wektora inicjalizującego. Ich wartości zostały wygenerowane losowo przez generator kluczy online [22] oraz zostały podane poniżej:

Wektor inicjalizujący: 4F6835DAC50F2133FF4DE91112F4F454

Klucz: 2E562CE90BF473F07CC9D050372FA3F8

Dane wyjściowe przed zapisaniem do pamięci karty micro-SD zostają zaszyfrowane.

# 6. Opracowanie oprogramowania dla mikrokontrolera

Oprogramowanie systemu mikroprocesorowego zostało napisane w języku programowania C. Środowiskiem programistycznym było CubeIDE dedykowane mikrokontrolerom firm STMicroelectronics. Głównym zadaniem jest: konfiguracja, inicjalizacja oraz obsługa modułów peryferyjnych. W skład programu wchodzą moduły:

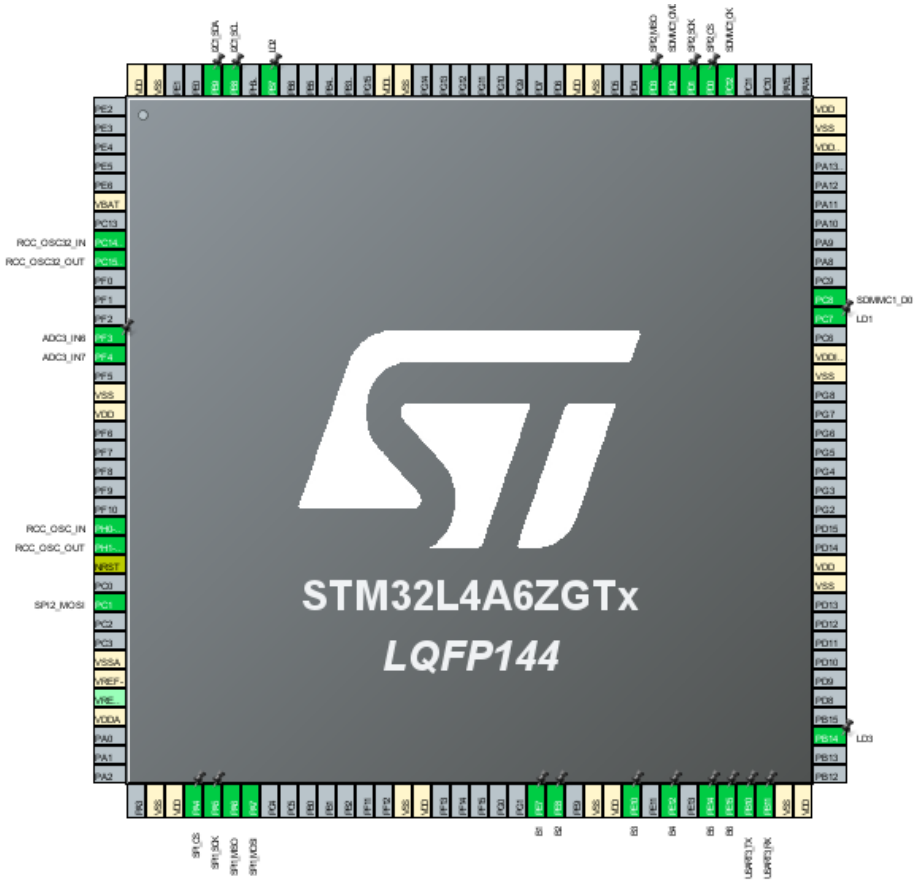
- Obsługi wyświetlacza OLED przy pomocy magistrali I2C;
- Obsługi akcelerometru ADXL345 przy pomocy magistrali SPI;
- Biblioteki kryptograficznej;
- Obsługa klawiatury przez porty GPIO;
- Moduł komunikacyjny z komputerem przez interfejs UART;
- Sterowników czytników kart pamięci przez magistrale SD oraz SPI;
- Bibliotekę obsługi pilików w systemie FAT;
- Przechodzenie w stany obniżonego poboru mocy;

Oprogramowanie po uruchomieniu inicjalizuje wszystkie interfejsy oraz sygnalizuje gotowość do rozpoczęcia pracy poprzez wyświetlenie odpowiedniego komunikatu. Następnie zostają wyświetlone możliwości zapisu danych z akcelerometru przy pomocy jednego z dwóch czytników pamięci oraz metoda zapisu. Z możliwością szyfrowania oraz bez. Po wybraniu jednej z opcji rozpoczyna się zapisywanie danych z wykorzystaniem biblioteki FatFs oraz wyświetlany jest czas zapisu. Podgląd danych dotyczących zapisu oraz rozszerzone możliwości sterowania są przesyłane przez interfejs UART.

## 6.1. Środowisko programowania CubeIDE i konfiguracja mikrokontrolera

CubeIDE jest to środowisko programowania zalecane przez producenta wybranego mikrokontrolera firmy ST. Zaawansowana platforma programistyczna języków C i C++ umożliwia konfiguracje urządzeń peryferyjnych, generowanie, kompilacje oraz debugowanie kodu. Składa się ono z konfiguratora systemu CubeMX oraz wbudowanego środowiska programistycznego Eclipse. Do programowania wykorzystuje narzędzia GCC natomiast do debugowania GDB. Samo środowisko pozwala na wygenerowanie dokumentacji informującej o konfiguracji projektu. Całość dokumentacji została dodana do załączników niniejszej pracy natomiast kluczowe elementy zostały przedstawione w rozdziałach poniżej. Na  Rysunek Opracowanie oprogramowania dla mikrokontrolera.18 przedstawiono konfiguracje wyprowadzeń mikrokontrolera. Na zielono zostały zaznaczone

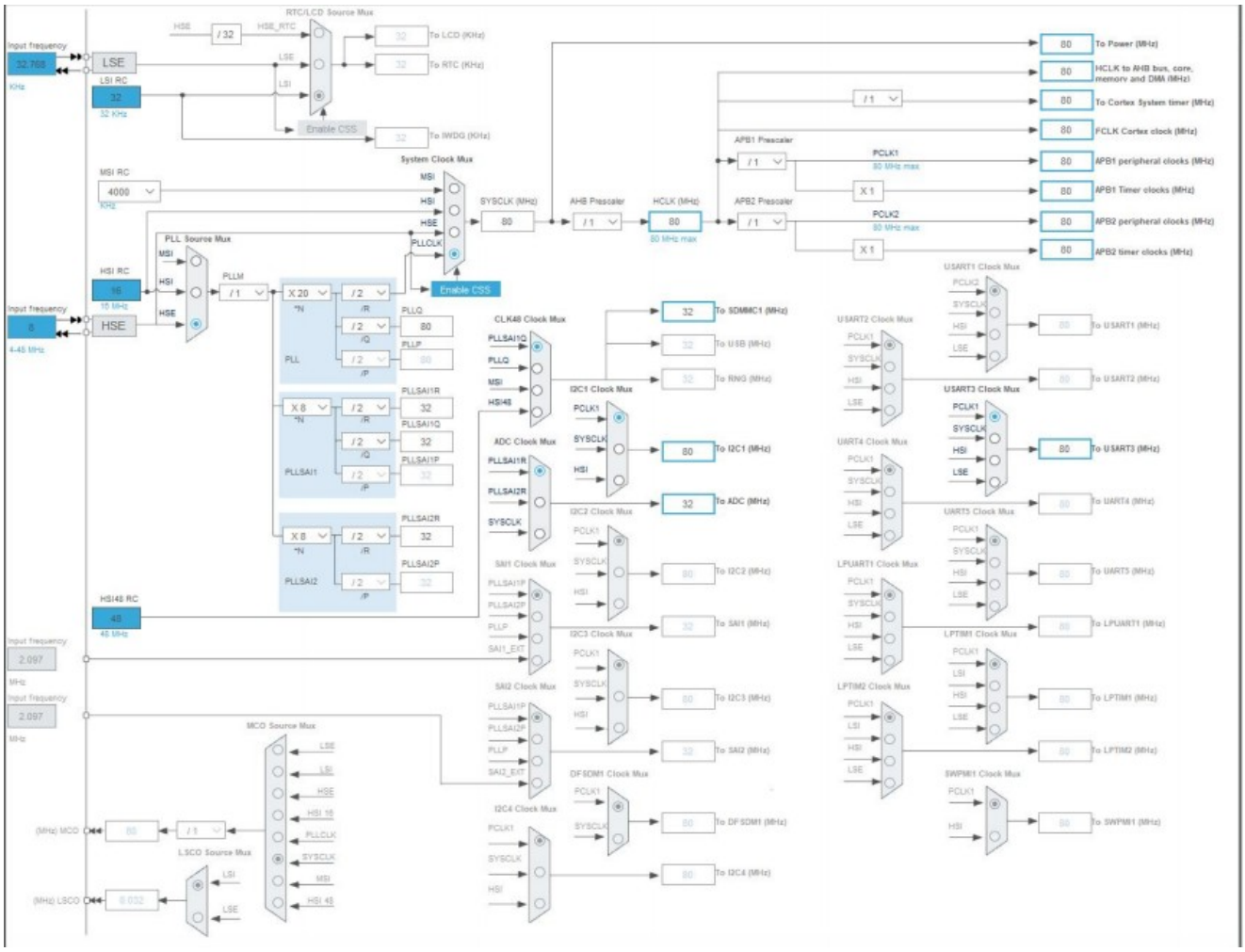
wykorzystane piny oraz ich etykiety informujące o przeznaczeniu. Do projektowania wykorzystano bibliotekę wsparcia dla mikrokontrolerów serii L4 w wersji 1.16.1. Oprogramowanie zostało podzielone na odrębne pliki z rozszerzeniami .h i .c zawierające oprogramowanie dedykowane konkretnym modułom peryferyjnym systemu.




*Rysunek Opracowanie oprogramowania dla mikrokontrolera.18. Konfiguracja wyprowadzeń mikrokontrolera STM32L46ZG*

### 6.1.1. Konfiguracja zegarów

W pierwszej kolejności został skonfigurowany sygnał taktujący. Na Rysunek Opracowanie oprogramowania dla mikrokontrolera.19 przedstawiono częstotliwości taktowania poszczególnych elementów systemu w środowisku CubeIDE. Zegar jest taktowany z wewnętrznego oscylatora HSI. Zapewnia on bardzo bezpieczne taktowanie zapewniające większą energooszczędność od zewnętrznego oscylatora kwarcowego HSE. Standardowo po uruchomieniu system jest taktowany z zegara HSI. Przełączenie na taktowanie zewnętrzne dodaje niepotrzebną zwłokę czasową, a dokładność taktowania nie jest kluczowa w projekcie. Główny zegar syste4mu HCLK działa z częstotliwością 80 MHz.




*Rysunek Opracowanie oprogramowania dla mikrokontrolera.19. Konfiguracja zegarów*

Kod wynikowy konfiguracji zegarów przedstawiono na Rysunek Opracowanie oprogramowania dla mikrokontrolera.20.Rysunek Charakterystyka mikrokontrolerów o niskim zużyciu energii..1

```
void SystemClock_Config(void)
{
  RCC_OscInitTypeDef RCC_OscInitStruct = {0};
  RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
  RCC_PeriphCLKInitTypeDef PeriphClkInit = {0};

  /** Initializes the RCC Oscillators according to the specified parameters
  * in the RCC_OscInitTypeDef structure.
  */
  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
  RCC_OscInitStruct.HSIState = RCC_HSI_ON;
  RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
  RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
  RCC_OscInitStruct.PLL.PLLM = 1;
  RCC_OscInitStruct.PLL.PLLN = 10;
  RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
  RCC_OscInitStruct.PLL.PLLQ = RCC_PLLQ_DIV2;
  RCC_OscInitStruct.PLL.PLLR = RCC_PLLR_DIV2;
  if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
  {
    Error_Handler();
  }
  /** Initializes the CPU, AHB and APB buses clocks
  */
  RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                              |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
  RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
  RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
  RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
  RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

  if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_4) != HAL_OK)
  {
    Error_Handler();
  }
  PeriphClkInit.PeriphClockSelection = RCC_PERIPHCLK_USART3|RCC_PERIPHCLK_I2C1
                                     |RCC_PERIPHCLK_SDMMC1;
  PeriphClkInit.Usart3ClockSelection = RCC_USART3CLKSOURCE_PCLK1;
  PeriphClkInit.I2c1ClockSelection = RCC_I2C1CLKSOURCE_PCLK1;
  PeriphClkInit.Sdmmc1ClockSelection = RCC_SDMMC1CLKSOURCE_PLLSAI1;
  PeriphClkInit.PLLSAI1.PLLSAI1Source = RCC_PLLSOURCE_HSI;
  PeriphClkInit.PLLSAI1.PLLSAI1M = 1;
  PeriphClkInit.PLLSAI1.PLLSAI1N = 8;
  PeriphClkInit.PLLSAI1.PLLSAI1P = RCC_PLLP_DIV2;
  PeriphClkInit.PLLSAI1.PLLSAI1Q = RCC_PLLQ_DIV4;
  PeriphClkInit.PLLSAI1.PLLSAI1R = RCC_PLLR_DIV2;
  PeriphClkInit.PLLSAI1.PLLSAI1ClockOut = RCC_PLLSAI1_48M2CLK;
  if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit) != HAL_OK)
  {
    Error_Handler();
  }
```

*Rysunek Opracowanie oprogramowania dla mikrokontrolera.20. Kod konfiguracji zegarów mikrokontrolera.*

### 6.1.2. Komunikacja przy pomocy UART

Ustawienia konfiguracyjne magistrali UART zostały przedstawione na Rysunek Opracowanie oprogramowania dla mikrokontrolera.21. Komunikacja odbywa się z prędkością 115200 w trybie asynchronicznym. Magistrala komunikuje się z użytkownikiem informując go aktualnym stanie urządzenia oraz odbiera w trybie przerwaniowym polecenia dla systemu.

### 7.9. USART3

**Mode: Asynchronous**

7.9.1. Parameter Settings:

| | |
|---|---|
| **Basic Parameters:** | |
| Baud Rate | 115200 |
| Word Length | 8 Bits (including Parity) |
| Parity | None |
| Stop Bits | 1 |
| **Advanced Parameters:** | |
| Data Direction | Receive and Transmit |
| Over Sampling | 16 Samples |
| Single Sample | Disable |
| **Advanced Features:** | |
| Auto Baudrate | Disable |
| TX Pin Active Level Inversion | Disable |
| RX Pin Active Level Inversion | Disable |
| Data Inversion | Disable |
| TX and RX Pins Swapping | Disable |
| Overrun | Enable |
| DMA on RX Error | Enable |
| MSB First | Disable |

*Rysunek Opracowanie oprogramowania dla mikrokontrolera.21. Konfiguracja UART*

Głównym zadaniem jest oczekiwanie na odebranie znaku polecenia od użytkownika. Po uruchomieniu systemu wyświetlane jest menu którego kod znajduje się na Rysunek Opracowanie oprogramowania dla mikrokontrolera.22. Przesłanie odpowiedniego numeru rozpocznie wykonywanie jednej z ośmiu funkcji. Odpowiadają one za przeprowadzenie całkowitej oraz wolnej pamięci, testów szyfrowania oraz rozpoczęcia zapisu przy pomocy jednego z dwóch czytników kart pamięci.

```
void uart_menu()
{
    send_uart("1. Sprawdź pamięć SPI\n");
    send_uart("2. Sprawdź pamięć SD\n");
    send_uart("3. Sprawdź szyfrowanie SPI\n");
    send_uart("4. Sprawdź Szyfrowanie SD\n");
    send_uart("5. Zapis bez szyfrowania SPI\n");
    send_uart("6. Zapis bez szyfrowania SD\n");
    send_uart("7. Zapis z szyfrowaniem SPI\n");
    send_uart("8. Zapis z szyfrowaniem SD\n");
    send_uart("Wybierz numer: \n");
}
```

*Rysunek Opracowanie oprogramowania dla mikrokontrolera.22. Menu programu przesyłane przez UART*

### 6.1.3. Komunikacja oraz obsługa akcelerometru przy pomocy SPI

Komunikacja z akcelerometrem odbywa się za pomocą jednej z dwóch wykorzystywanych magistral SPI. Wyprowadzenia magistrali zostały przedstawione na Rysunek Opracowanie oprogramowania dla mikrokontrolera.24. Wykorzystuje on cztery wyprowadzenia zawierające komunikacje wejściową, wyjściową, linię zegarową oraz adresową.

| | | | | | | |
|---|---|---|---|---|---|---|
| SPI2 | PC1 | SPI2_MOSI | Alternate Function Push Pull | No pull-up and no pull-down | Very High * | |
| | PD1 | SPI2_SCK | Alternate Function Push Pull | No pull-up and no pull-down | Very High * | |
| | PD3 | SPI2_MISO | Alternate Function Push Pull | No pull-up and no pull-down | Very High * | |
| | PD0 | GPIO_Output | Output Push Pull | No pull-up and no pull-down | High * | SPI2_CS |

*Rysunek Opracowanie oprogramowania dla mikrokontrolera.23. Wyprowadzenia magistrali SPI do komunikacji z akcelerometrem*

Jego konfiguracje przedstawiono na Rysunek Opracowanie oprogramowania dla mikrokontrolera.24. Format ramki danych został ustawiony w standardzie Motorola oraz jej wielkość 8 Bitów. Najwyższą prędkością transmisji obsługiwaną przez akcelerometr jest 5 MB/s. W tym celu zegar taktujący został przy pomocy preskalera przeskalowany do prędkości 1,25 MB/s. Dodatkowo wg. dokumentacji bity CPOL oraz CPHA zostały ustawione na jeden.

**7.6. SPI2**

**Mode: Full-Duplex Master**

7.6.1. Parameter Settings:

| | |
|---|---|
| **Basic Parameters:** | |
| Frame Format | Motorola |
| Data Size | **8 Bits *** |
| First Bit | MSB First |
| **Clock Parameters:** | |
| Prescaler (for Baud Rate) | **32 *** |
| Baud Rate | **1.25 MBits/s *** |
| Clock Polarity (CPOL) | **High *** |
| Clock Phase (CPHA) | **2 Edge *** |
| **Advanced Parameters:** | |
| CRC Calculation | Disabled |
| NSS Signal Type | Software |

*Rysunek Opracowanie oprogramowania dla mikrokontrolera.24. Konfiguracja SPI do komunikacji z akcelerometrem*

Obsługa akcelerometru odbywa się przy pomocy czterech funkcji: inicjalizacji zapisu do rejestru, odczytu rejestru oraz odczytu wartości, których kod został przedstawiony na Rysunek Opracowanie oprogramowania dla mikrokontrolera.25. Funkcja *adxl_write* przyjmuje dwie wartość adres rejestru oraz wartości. Funkcja w pierwszej kolejności wartość adresu poddaje operacji logicznej OR z wartością 0x40. Informuje to akcelerometr, że zostanie przesłane więcej niż jeden bajt danych. Następnie ustawia stan niski linii adresowej a następnie przesyła wartość na podany adres. Po zakończonej transmisji linia adresowa zmieniana jest na stan wysoki. Przy pomocy tej funkcji odbywa się funkcja inicjalizacji *adxl_init.* Wysyła ona trzy polecenia na trzy adresy. W pierwszej kolejności ustawia format na ±4 G. Następnie zeruje wszystkie bity rejestru trybu pracy a następnie ustawia go w tryb pracy ciągłej. Odczytanie wartości przy pomocy funkcji *adxl_read* odbywa się w podobny sposób jak zapis. W pierwszej kolejności przeprowadzane są operacje XOR na adresie by poinformować akcelerometr o tym że chcemy odczytać wiele bajtów danych. W tym celu szósty i ostatni bit są ustawione w stan wysoki. W dalszej kolejności zmieniany jest stan szyny adresowej na niski. Przesyłane są wcześniej przygotowane informacje oraz odczytywana wartość oraz zmiana linii adresowej na stan wysoki. Ostatnia funkcja odczytuje dane oraz przypisuje konkretnym zmiennym również przeliczając na jednostkę przyspieszenia ziemskiego. W tym celu odczyt przemnożony jest przez wartość 0,0078.

```
void adxl_write (uint8_t address, uint8_t value)
{
    uint8_t data[2];
    data[0] = address|0x40;
    data[1] = value;
    HAL_GPIO_WritePin (GPIOD, GPIO_PIN_0, GPIO_PIN_RESET);
    HAL_SPI_Transmit (&hspi2, data, 2, 100);
    HAL_GPIO_WritePin (GPIOD, GPIO_PIN_0, GPIO_PIN_SET);
}

void adxl_read (uint8_t address)
{
    address |= 0x80;
    address |= 0x40;
    HAL_GPIO_WritePin (GPIOD, GPIO_PIN_0, GPIO_PIN_RESET);
    HAL_SPI_Transmit (&hspi2, &address, 1, 100);
    HAL_SPI_Receive (&hspi2, data_rec, 6, 100);
    HAL_GPIO_WritePin (GPIOD, GPIO_PIN_0, GPIO_PIN_SET);
}

void adxl_init (void)
{
    adxl_write (0x31, 0x01);
    adxl_write (0x2d, 0x00);
    adxl_write (0x2d, 0x08);
}

void adxl_read_value(uint8_t address)
{
    adxl_read(address);
    x = ((data_rec[1]<<8)|data_rec[0]);
    y = ((data_rec[3]<<8)|data_rec[2]);
    z = ((data_rec[5]<<8)|data_rec[4]);

    // Convert into 'g'

    xg = x*.0078;
    yg = y*.0078;
    zg = z*.0078;
}
```

*Rysunek Opracowanie oprogramowania dla mikrokontrolera.25. Funkcje obsługi akcelerometru.*

### 6.1.4. Biblioteka kryptograficzna AES

Biblioteka kryptograficzna została skonfigurowana w sposób przedstawiony na Rysunek Opracowanie oprogramowania dla mikrokontrolera.26. Tryb szyfrowania ustawiono na tryb licznikowy CTR. Długość klucza została ustawiona na 128 bitów. Wartość klucza oraz wektora inicjalizującego została skonfigurowana zgodnie z podpunktem Biblioteka kryptograficzna w mikrokontrolerach STM32.

**7.1. AES**
**mode: Activated**
7.1.1. Parameter Settings:

| | |
|---|---|
| **Algorithm:** | |
| Data encryption type | **AES CTR *** |
| **Parameters:** | |
| Data type | **16b(half-word swapping) *** |
| KeySize | 128b |
| Encryption/Decryption key | **2E 56 2C E9 0B F4 73 F0 7C C9 D0 50 37 2F A3 F8 *** |
| Operating Mode | Encryption mode |
| Chaining Mode | Counter mode chaining algorithm (CRYP_CHAINMODE_AES_CTR) |
| Key Write Flag | Enable decryption key writing |
| Initialization vector | **4F 68 35 DA C5 0F 21 33 FF 4D E9 11 12 F4 F4 54 *** |

*Rysunek Opracowanie oprogramowania dla mikrokontrolera.26. Konfiguracja AES*

### 6.1.5. Obsługa czytnika kart interfejs SD

Czytnik kart microSD wykorzystuje interfejs SD zarówno w trybie jedno jak i cztero-liniowym. Konfiguracja trybu jednoliniowego została przedstawiona na Rysunek Opracowanie oprogramowania dla mikrokontrolera.27. Wprowadzenie PD2 zostało zadeklarowane jako linia adresowa, PC12 jako linia zegara taktującego oraz linia danych PC8.

```
void HAL_SD_MspInit(SD_HandleTypeDef* sdHandle)
{

  GPIO_InitTypeDef GPIO_InitStruct = {0};
  if(sdHandle->Instance==SDMMC1)
  {
  /* USER CODE BEGIN SDMMC1_MspInit 0 */

  /* USER CODE END SDMMC1_MspInit 0 */
    /* SDMMC1 clock enable */
    __HAL_RCC_SDMMC1_CLK_ENABLE();

    __HAL_RCC_GPIOC_CLK_ENABLE();
    __HAL_RCC_GPIOD_CLK_ENABLE();
    /**SDMMC1 GPIO Configuration
    PC8     ------> SDMMC1_D0
    PC12     ------> SDMMC1_CK
    PD2     ------> SDMMC1_CMD
    */
    GPIO_InitStruct.Pin = GPIO_PIN_8|GPIO_PIN_12;
    GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
    GPIO_InitStruct.Alternate = GPIO_AF12_SDMMC1;
    HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);

    GPIO_InitStruct.Pin = GPIO_PIN_2;
    GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
    GPIO_InitStruct.Alternate = GPIO_AF12_SDMMC1;
    HAL_GPIO_Init(GPIOD, &GPIO_InitStruct);

  /* USER CODE BEGIN SDMMC1_MspInit 1 */

  /* USER CODE END SDMMC1_MspInit 1 */
  }
}
```

*Rysunek Opracowanie oprogramowania dla mikrokontrolera.27. Konfiguracja SD 1-linowy*

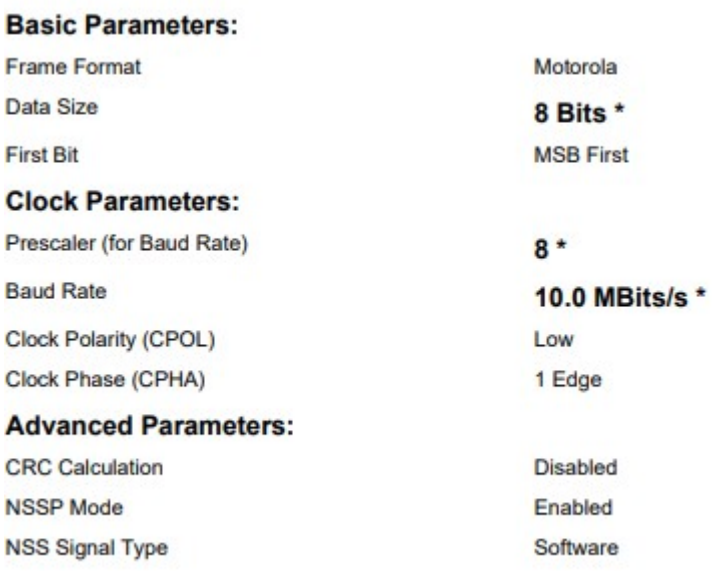
Zasadniczą różnicą między dwoma trybami jest liczba linii komunikacyjnych. Konfiguracja trybu czteroliniowego przedstawiona na Rysunek Opracowanie oprogramowania dla mikrokontrolera.28. Dodatkowo zostały zadeklarowane wyprowadzenia PC9, PC10 oraz PC11 jako linie wymiany danych. Po uruchomieniu oprogramowania interfejs jest inicjalizowany w wybranym przez użytkownika trybie. Obsługa karty odbywa się przy pomocy bibliotek obsługi karty wbudowanych w standardowy zestaw bibliotek CubeIDE.

```
void HAL_SD_MspInit(SD_HandleTypeDef* sdHandle)
{

  GPIO_InitTypeDef GPIO_InitStruct = {0};
  if(sdHandle->Instance==SDMMC1)
  {
  /* USER CODE BEGIN SDMMC1_MspInit 0 */

  /* USER CODE END SDMMC1_MspInit 0 */
    /* SDMMC1 clock enable */
    __HAL_RCC_SDMMC1_CLK_ENABLE();

    __HAL_RCC_GPIOC_CLK_ENABLE();
    __HAL_RCC_GPIOD_CLK_ENABLE();
    /**SDMMC1 GPIO Configuration
    PC8     ------> SDMMC1_D0
    PC9     ------> SDMMC1_D1
    PC10     ------> SDMMC1_D2
    PC11     ------> SDMMC1_D3
    PC12     ------> SDMMC1_CK
    PD2     ------> SDMMC1_CMD
    */
    GPIO_InitStruct.Pin = GPIO_PIN_8|GPIO_PIN_9|GPIO_PIN_10|GPIO_PIN_11
                          |GPIO_PIN_12;
    GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
    GPIO_InitStruct.Alternate = GPIO_AF12_SDMMC1;
    HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);

    GPIO_InitStruct.Pin = GPIO_PIN_2;
    GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
    GPIO_InitStruct.Alternate = GPIO_AF12_SDMMC1;
    HAL_GPIO_Init(GPIOD, &GPIO_InitStruct);

  /* USER CODE BEGIN SDMMC1_MspInit 1 */

  /* USER CODE END SDMMC1_MspInit 1 */
  }
}
```

*Rysunek Opracowanie oprogramowania dla mikrokontrolera.28. Konfiguracja SD 4-liniowy*

### 6.1.6. Obsługa czytnika kart interfejs SPI

Drugi z czytników kart wykorzystuje interfejs SPI. Został on skonfigurowany według parametrów przedstawionych na Rysunek Opracowanie oprogramowania dla mikrokontrolera.29. Wartość preskalera zegara taktującego została zmieniona na 8. Jest to spowodowane koniecznością obniżenia prędkości komunikacji do 10 MB/s. Jest to ograniczenie sterownika do komunikacji kart microSD przez interfejs SPI. Kod konfiguracji został przedstawiony na Rysunek Opracowanie oprogramowania dla mikrokontrolera.30.



| | |
|---|---|
| **Basic Parameters:** | |
| Frame Format | Motorola |
| Data Size | **8 Bits *** |
| First Bit | MSB First |
| **Clock Parameters:** | |
| Prescaler (for Baud Rate) | **8 *** |
| Baud Rate | **10.0 MBits/s *** |
| Clock Polarity (CPOL) | Low |
| Clock Phase (CPHA) | 1 Edge |
| **Advanced Parameters:** | |
| CRC Calculation | Disabled |
| NSSP Mode | Enabled |
| NSS Signal Type | Software |

*Rysunek Opracowanie oprogramowania dla mikrokontrolera.29. Konfiguracja SPI do komunikacji z czytnikiem kart pamięci.*

Interfejs wykorzystuje cztery wyprowadzenia. Do wyprowadzenia zegara taktującego został wykorzystany pin PA5. Dane wejściowe są odbierane przez pin PA6 natomiast wyjściowe przez PA7. Linią adresową dla tego interfejsu jest pory wyjściowy PA4.

```
GPIO_InitTypeDef GPIO_InitStruct = {0};
if(spiHandle->Instance==SPI1)
{
/* USER CODE BEGIN SPI1_MspInit 0 */

/* USER CODE END SPI1_MspInit 0 */
  /* SPI1 clock enable */
  __HAL_RCC_SPI1_CLK_ENABLE();

  __HAL_RCC_GPIOA_CLK_ENABLE();
  /**SPI1 GPIO Configuration
  PA5     ------> SPI1_SCK
  PA6     ------> SPI1_MISO
  PA7     ------> SPI1_MOSI
  */
  GPIO_InitStruct.Pin = GPIO_PIN_5|GPIO_PIN_6|GPIO_PIN_7;
  GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
  GPIO_InitStruct.Alternate = GPIO_AF5_SPI1;
  HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

/* USER CODE BEGIN SPI1_MspInit 1 */

/* USER CODE END SPI1_MspInit 1 */
}
```

*Rysunek Opracowanie oprogramowania dla mikrokontrolera.30. Kod konfiguracji SPI*

### 6.1.7. Biblioteka FatFs

*Do obsługi systemu plików FAT została wykorzystana biblioteka FatFs. Jest ona włączona w standardowy zestaw bibliotek dostępnych w CubeIDE. Została ona skonfigurowana według parametrów przedstawionych na*

.

| | |
|---|---|
| **Version:** | |
| FATFS version | R0.12c |
| **Function Parameters:** | |
| FS_READONLY (Read-only mode) | Disabled |
| FS_MINIMIZE (Minimization level) | Disabled |
| USE_STRFUNC (String functions) | Enabled with LF -> CRLF conversion |
| USE_FIND (Find functions) | Disabled |
| USE_MKFS (Make filesystem function) | Enabled |
| USE_FASTSEEK (Fast seek function) | Enabled |
| USE_EXPAND (Use f_expand function) | Disabled |
| USE_CHMOD (Change attributes function) | Disabled |
| USE_LABEL (Volume label functions) | Disabled |
| USE_FORWARD (Forward function) | Disabled |
| **Locale and Namespace Parameters:** | |
| CODE_PAGE (Code page on target) | Latin 1 |
| USE_LFN (Use Long Filename) | **Enabled with static working buffer on the BSS *** |
| MAX_LFN (Max Long Filename) | 255 |
| LFN_UNICODE (Enable Unicode) | ANSI/OEM |
| STRF_ENCODE (Character encoding) | UTF-8 |
| FS_RPATH (Relative Path) | Disabled |
| **Physical Drive Parameters:** | |
| VOLUMES (Logical drives) | 2 |
| MAX_SS (Maximum Sector Size) | **4096 *** |
| MIN_SS (Minimum Sector Size) | 512 |
| MULTI_PARTITION (Volume partitions feature) | Disabled |
| USE_TRIM (Erase feature) | Disabled |
| FS_NOFSINFO (Force full FAT scan) | 0 |
| **System Parameters:** | |
| FS_TINY (Tiny mode) | **Enabled *** |
| FS_EXFAT (Support of exFAT file system) | **Enabled *** |
| FS_NORTC (Timestamp feature) | Dynamic timestamp |
| FS_REENTRANT (Re-Entrancy) | Disabled |
| FS_TIMEOUT (Timeout ticks) | **3000 *** |
| FS_LOCK (Number of files opened simultaneously) | 2 |

*Rysunek Opracowanie oprogramowania dla mikrokontrolera.31. Konfiguracja FatFs*

Został on zdefiniowany do obsługi dwóch dysków logicznych jednocześnie. Pierwszy tryb został powiązany z funkcjami obsługi karty przez interfejs SD i został on oznaczony cyfrą 0. Drugi został powiązany z zestawem bibliotek obsługujących czytnik kart przez interfejs SPI. Biblioteka wykorzystuje zestaw kodowy UTF-8 w pełni

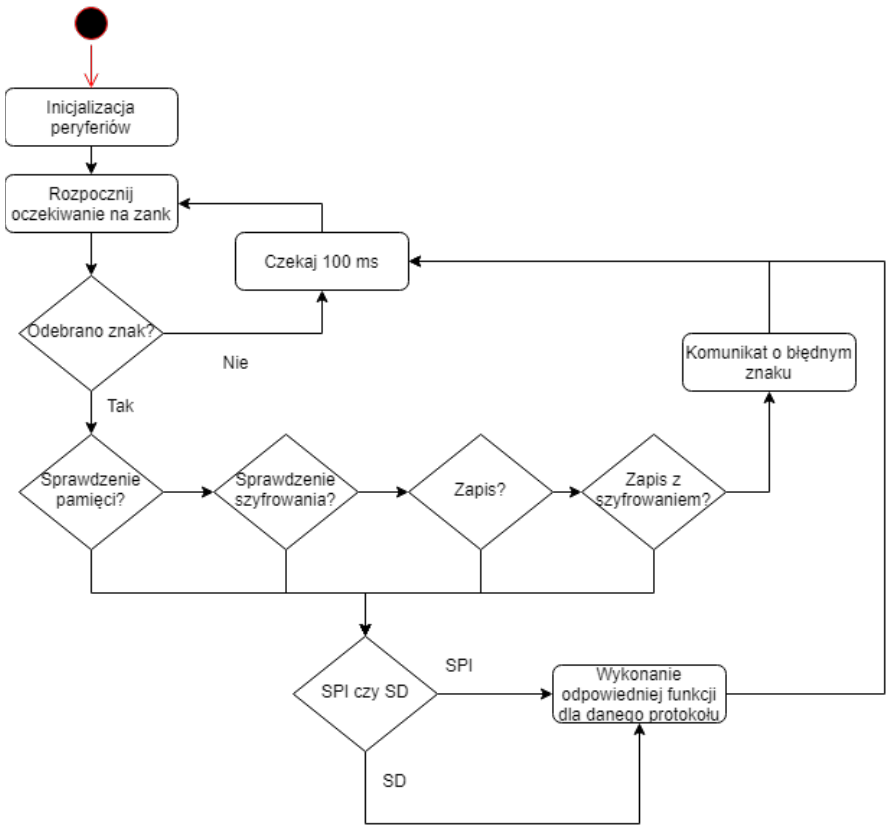
kompatybilny z ASCII. Maksymalna wielkość odczytanego sektora została zmieniona na wartość 4096, możliwość otworzenia dwóch plików jednocześnie oraz obsługę systemu plików exFAT.

Najważniejszymi funkcjami biblioteki są:

- *f_mount* – inicjalizacja dysku logicznego w systemie,
- *f_open* – otwarcie lub utworzenie pliku,
- *f_close* – zamknięcie pliku,
- *f_read* – odczytanie zawartości pliku,
- *f_write* – wpisanie zawartości do pliku,
- *f_puts* – wprowadzenie do pliku linijki tekstu,
- *f_Iseek* – przesunięcie wskaźnika zapisu lub odczytu w zadeklarowane miejsce.

## 6.2. Oprogramowanie pomiarowe

Głównym zadaniem oprogramowania jest przeprowadzenie pomiaru według zadanych przez użytkownika parametrów. Diagram UML przedstawiający kolejność wykonywanych działań został przedstawiony na Rysunek Opracowanie oprogramowania dla mikrokontrolera.32.



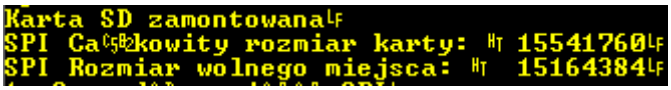

*Rysunek Opracowanie oprogramowania dla mikrokontrolera.32. Diagram funkcji głównej.*

W pierwszej kolejności program wykonuje inicjalizacji modułów peryferyjnych. Następnie przechodzi do nasłuchiwania wysłania jednego znaku przez interfejs UART oraz wchodzi do nieskończonej pętli głównej programu. W pętli sprawdzany jest warunek czy został odebrany znak. Jeśli został odebrany następuje sprawdzenie czy odpowiada on wykonaniu jednej z ośmiu instrukcji. W przypadku odebrania znaku „1” lub „2” zostanie wykonanie sprawdzenie pamięci karty microSD przez odpowiedni czytnik kart pamięci przy pomocy funkcji *MemChc()*. Kod funkcji przedstawiono poniżej:

```
    void MemChc(char *mode)
{
    char *path;
    toupper(mode);
    if(strcmp(mode,”SPI”)==0)
    path = „1:/”;
    else if(strcmp(mode,”SDIO”)==0)
    path = „0:/”;
    fresult = f_mount(&fs, path, 1); if(fresult != FR_OK)
    send_uart(„Błąd w montowaniu karty\n”);
    else
       {
    send_uart(„Karta SD zamontowana\n”);
    f_getfree(path, &fre_clust, &pfs);
    total = (uint32_t)((pfs->n_fatent-2)* pfs->csize * 0.5);
sprintf(buffer, „%s Całkowity rozmiar karty: \t%lu\n”,mode, total);
send_uart(buffer);
bufclear();
free_space = (uint32_t)(fre_clust * pfs->csize * 0.5);
    sprintf(buffer, „%s Rozmiar wolnego miejsca: \t%lu\n”, mode, free_space);
send_uart(buffer);
bufclear();
        }
    fresult = f_mount(NULL, path, 1);
}
```

Przyjmuje ona jako parametr wejściowy wskaźnik znakowy informujący w którym z interfejsów przeprowadzić sprawdzenie pamięci. W zależności od wpisanej wartości ścieżka zostaje ustawiona na odpowiadającą interfejsowi wartość w bibliotece FatFs. Kolejnym krokiem jest zamontowanie dysku logicznego przy pomocy funkcji f_mount. W razie niepowodzenia wysyłany jest komunikat o błędzie oraz funkcja kończy działanie. W przeciwnym razie poprzez odwołanie się do poszczególnych wartości struktury nośnika danych uzyskuje oraz prezentuje dane na temat pojemności i wolnego miejsca. Rezultat wykonania funkcji przedstawiono na Rysunek Opracowanie oprogramowania dla mikrokontrolera.33. W tym przypadku został przeprowadzony

pomiar karty o pojemności 16 GB zamontowanej w czytniku kart obsługiwanym przy pomocy interfejsu SPI.



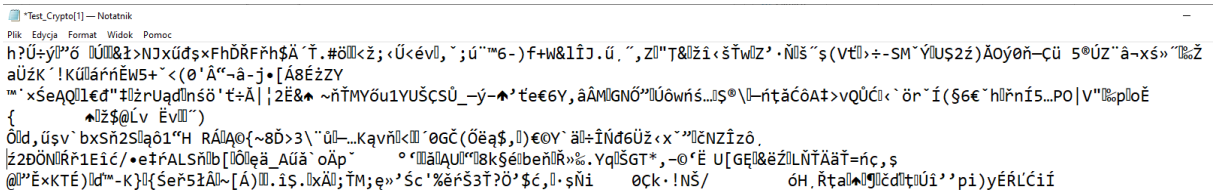

*Rysunek Opracowanie oprogramowania dla mikrokontrolera.33. Rezultat wykonanej funkcji MemChc.*

Aby przetestować poprawność działania algorytmów szyfrujących napisano funkcję *CryptoTest.* Zostaje wywołana w przypadku wysłania przez użytkownika znaku 3 lub 4 w zależności od wybranej karty pamięci do przetestowania. Funkcja ta powinna być wykonywana przed rozpoczęciem zapisu danych na karcie microSD w celu sprawdzenia poprawności działania algorytmów bezpieczeństwa. Kod funkcji znajduje się poniżej:

```
    void Crypto_test(char *mode)
{
    char path[80];
    toupper(mode);
    if(strcmp(mode,”SPI”)==0)
    strcpy(path,”1:/”);
    else if(strcmp(mode,”SDIO”)==0)
    strcpy(path, „0:/”);
    strcat(path, „Test_Crypto.txt”);
    fresult = f_mount(&fs, path, 1);
    sprintf(buffer, „Test do zaszyfrowana\n”);
    fresult = f_open(&fil, path, FA_OPEN_ALWAYS | FA_READ | FA_WRITE);
 if(fresult != FR_OK)
    send_uart(„ Error in monting SD CARD\n”);
HAL_CRYP_AESCTR_Encrypt(&hcryp, &buffer, sizeof(buffer), &buffercrypto, 10);
fresult = f_puts(buffercrypto, &fil);
f_close(&fil);
fresult = f_open(&fil, path ,FA_READ);

    f_gets(buffercrypto, sizeof(buffercrypto), &fil);
    f_close(&fil);
    HAL_CRYP_AESCTR_Decrypt(&hcryp, &buffercrypto, sizeof(buffercrypto),
 &buffer, 10);
    f_unlink(path);
    if(strcmp(buffer,”Test do zaszyfrowana\n”)==0)
    {
    send_uart(„Szyfrowanie OK!\n”);

    }
    fresult = f_mount(NULL, path, 1);
              }
```

Przyjmuje wskaźnik znakowy informujący na którym czytniku kart przeprowadzić test. W pierwszej kolejności czy nośnik pamięci jest poprawnie zamontowany w czytniku i inicjalizuje go w systemie. W razie niepowodzenia wysyła informację przez interfejs UART informacje o błędzie w inicjalizacji. Następnie do bufora

wpisywany jest szyfrogram o treści „Test do zaszyfrowania” ze znakiem końca linii „\n” na końcu. Następnie otwierany jest plik tekstowy „test_crypto.txt” do odczytu lub zapisu. Jeśli taki plik nie istnieje na nośniku pamięci to zostaje utworzony. Następnie tekst zostaje zaszyfrowany oraz wpisany do *buffercrypto* który następnie jest umieszczany w pliku i zamknięty. Następnie otwarty jest tylko z możliwością odczytu oraz wpisany do *buffercrypto,* a plik zostaje zamknięty. Odczytana wartość jest odszyfrowywany oraz wpisana do bufora. Nośnik zostaje wymontowany z systemu, a tekst odszyfrowany zostaje porównany z jego pierwotną wersją. W przypadku powodzenia całej operacji wysyłany jest komunikat przez UART informujący o powodzeniu szyfrowania. Zawartość pliku Test_crypto.txt przedstawiono na Rysunek Opracowanie oprogramowania dla mikrokontrolera.34.Error: Reference source not found




*Rysunek Opracowanie oprogramowania dla mikrokontrolera.34. Efekt działania testu szyfrowania.*

W wypadku wybrania przez użytkownika znaków od 5 do 8 zostanie wykonany zapis danych z akcelerometru na jedną z kart pamięci. Wybór polecenia 5 lub 7 rozpocznie zapis na karcie podłączonej do systemu przez interfejs SPI z opcją szyfrowania lub bez. Kod funkcji został przedstawiony poniżej:

```
    case 5:
    start = aktual;
HAL_TIM_Base_Start_IT(&htim17);
while(aktual<=(start+60000))
                        {

    while(cnt<=512)
                                {
    if(flaga_zapis==0)
                                        {
valx[cnt] = xg;
valy[cnt] = yg;
valz[cnt] = zg;
cnt++;
flaga_zapis=1;
                                        }
                                }
HAL_TIM_Base_Stop_IT(&htim17);
```

```
zapis(„SPI”);
HAL_TIM_Base_Start_IT(&htim17);
cnt=0;

                                }
HAL_TIM_Base_Stop_IT(&htim17);
send_uart(„Koniec zapisu SPI\n”);
uart_menu();
            break;
```

W pierwszej kolejności pobierana jest wartość zmiennej *aktual* i zapisana w zmiennej *start.* Jest liczba milisekund, które upłynęły od włączenia systemu. Wartość ta jest inkrementowana każdorazowo przez przerwanie systemowe *SysTick_Handler()* wywoływane co jedną milisekundę. W następnej kolejności jest uruchamiany licznik *TIM17.* Został on tak skonfigurowany by wywoływał przerwanie z częstotliwością 1 kHz. Obsługę przerwania opisuje kod poniżej:

```
    void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
    if(htim->Instance == TIM17)
    {

    if(flaga_zapis)
            {
adxl_read_value(0x32);
flaga_zapis=0;
            }

    }
}
```

Jeśli flaga_zapis jest równa 1 co oznacza gotowość do zapisu nowej porcji danych do bufora wykonywany jest odczyt wartości z akcelerometru. Po zakończonym pomiarze flaga zmienia stan na 0 informując o nowej paczce danych.

Pierwsza z pętli *while* wykonuje swoją zawartość do czasu aż aktualny czas działania programu nie zrówna się z czasem przez który chcemy wykonywać pomiar. W powyższym przypadku będzie to 60 000 milisekund czyli jedną minutę. Pętla główna programu wchodzi w pętle wykonującą się aż do zapełnienia bufora. Funkcja wewnętrzna sprawdza czy wartości pobrane w przerwaniu są aktualne. W przypadku wartości 0 wpisuje nowe wartości do bufora, przesuwa zapisu następnej wartości oraz informuje o gotowości na pobranie następnej zmieniając stan flagi na 1. Proces powtarza się do zapełnienia bufora. W momencie jego przepełnienia licznik przerwania zostaje zatrzymany oraz uruchamiana jest procedura zapisu paczki danych na karcie przy pomocy funkcji *zapis(),* która przyjmuje

jako wartość wskaźnik, do której z kart ma zostać wykonany zapis. Kod funkcji umieszczono poniżej:

```
    void zapis(char *mode)
{
    char path[80];
    toupper(mode);
    if(strcmp(mode,”SPI”)==0)
    strcpy(path,”1:/”);
    else if(strcmp(mode,”SDIO”)==0)
    strcpy(path,”0:/”);
    strcat(path, „zapis_1.txt”);
    fresult = f_mount(&fs, path, 1); if(fresult != FR_OK)
    send_uart(„Błąd w montowaniu karty\n”);
    else
    {
    fresult = f_open(&fil, path, FA_OPEN_ALWAYS | FA_READ | FA_WRITE);
    if(fresult != FR_OK)
    send_uart(„Bład otwarcia pliku\n”);
    fresult=f_lseek(&fil, f_size(&fil));
    for(int i=0; i<512; i++)
            {
    sprintf(buffer, „%f\t%f\t%f\n”,valx[i],valy[i],valz[i]);
    f_puts(buffer, &fil);
            }
    f_close(&fil);
    }

    fresult = f_mount(NULL, path, 1);
}
```

Tak jak w przypadku funkcji sprawdzającej pamięć w pierwszej kolejności montowany jest dysk logiczny w systemie. W razie niepowodzenia tej operacji wysyłany jest komunikat o błędzie. Tworzona jest ścieżka do pliku zapisu oraz otwarcie pliku tekstowego do zapisu. W przypadku gdy plik nie istnieje to zostanie utworzony w przeciwnym razie nadpisany. Funkcja wewnętrzna *f_Iseek* przesuwa wskaźnik rozpoczęcia zapisu/odczytu na koniec pliku. Umożliwia to dopisywanie kolejnych paczek danych do jednego pliku w celu utrzymania ciągłości zapisu. Następnie zawartość bufora kopiowana jest do pliku tekstowego. Po zakończonej operacji plik oraz nośnik jest logicznie wymontowywany z systemu.

Po powrocie do funkcji nadrzędnej licznik przerwań ponownie zostaje uruchomiony oraz zerowany jest licznik wypełnienia bufora, a program wykonuje ponownie instrukcje, aż do upłynięcia czasu pomiaru. W przypadku zakończenia czasu pomiaru licznik przerwań zostanie zatrzymany a program wysyła informacje o zakończeniu zapisu. Program wraca do pętli głównej oraz oczekuje kolejnych poleceń.

Przesłanie znaku 7 lub 8 rozpocznie zapis na kartę szyfrowanych danych. W odróżnieniu od przypadku opisanego powyżej przed zapisem informacje zostają zaszyfrowane. Fragment kodu przedstawiono poniżej:

```
    if(flaga_zapis==0)
    {
HAL_CRYP_AESCTR_Encrypt(&hcryp, &xg, sizeof(xg), &xc, 10);
HAL_CRYP_AESCTR_Encrypt(&hcryp, &yg, sizeof(yg), &yc, 10);
HAL_CRYP_AESCTR_Encrypt(&hcryp, &zg, sizeof(zg), &zc, 10);
valx[cnt] = xc;
valy[cnt] = yc;
valz[cnt] = zc;
cnt++;
flaga_zapis=1;
    }
```

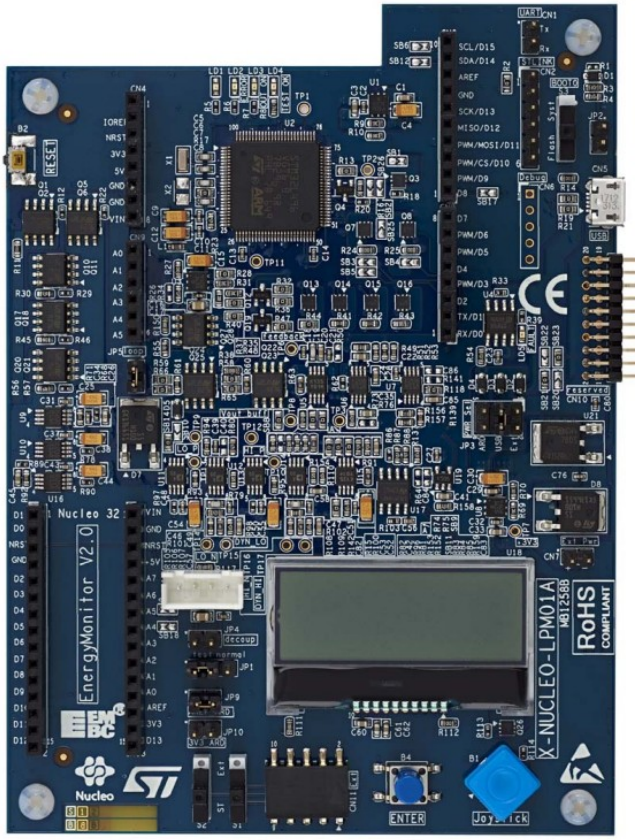
Po odczycie wartość jest zaszyfrowywana funkcją biblioteki HAL w trybie CTR, a następnie trafia do bufora zapisu. Następne kroki wykonywania procedury są takie same jak w poprzednim przypadku.

# 7. Wykonanie badań testowych i analiza zużycia energii

Założeniem badań testowych było sprawdzenie, który z trybów zapisu danych na karcie microSD, oraz która z przetestowanych kart pamięci osiąga najlepsze wyniki pod względem wydajności energetycznej. Do tego celu zostały przetestowane cztery karty podczas zapisu danych w trybach SD oraz SPI. Dodatkowo zmierzono wpływ szyfrowania na pobór mocy systemu mikroprocesorowego.

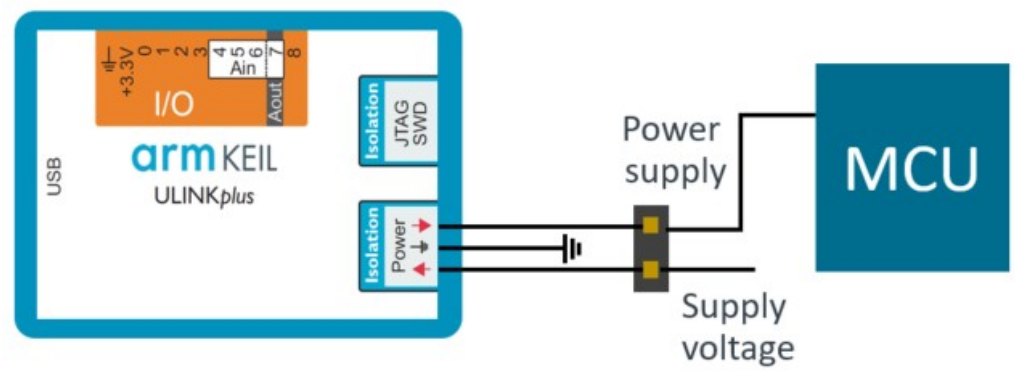
## 7.1. Porównanie urządzeń do pomiaru mocy

Pomiar poboru mocy przez mikrokontroler można zmierzyć poprzez wiele narzędzi do tego dedykowanych. Najczęściej są one źródłem zasilania dla mikrokontrolera. Jednym z rozwiązań jest rozszerzenie do pomiaru mocy X-NUCLEO-LPM01A dedykowane mikrokontrolerom z rodziny STM32 NUCLEO. Płytka rozszerzeniowa przedstawiona na Rysunek Wykonanie badań testowych i analiza zużycia energii.35. pozwala na pomiar mocy średniej do 200 mA lub w czasie rzeczywistym do 50 mA. Uniwersalne złącza Arduino ułatwiają szybki montaż w urządzeniach wyposażonych w ten standard. Urządzenie może działać zarówno współdziałając z komputerem jak i autonomicznie.



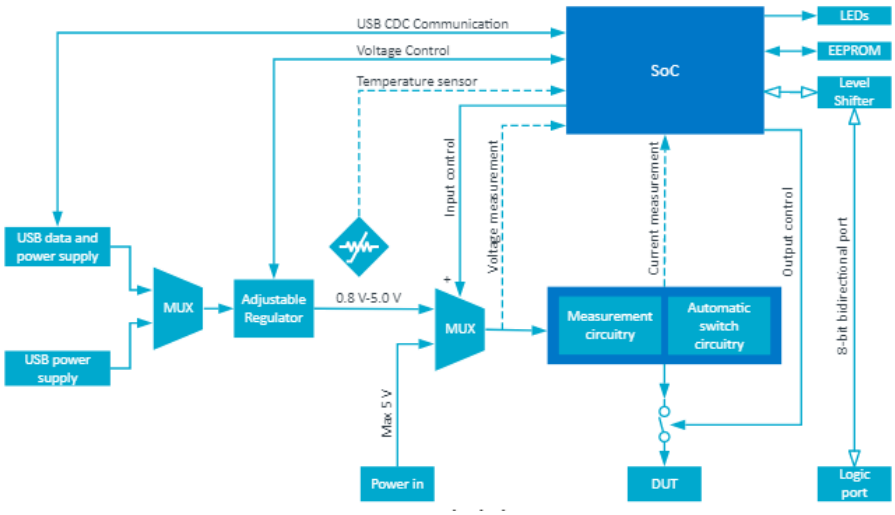

*Rysunek Wykonanie badań testowych i analiza zużycia energii.35. Płytka rozszerzeniowa X-NUCLEO-LPM01A*

Następnym z możliwych do wykorzystania rozwiązań jest adapter debuger Keil ULINKplus przedstawiony na . Jest to dedykowane urządzenie dla mikrokontrolerów z rdzeniem ARM. Pozwala nie tylko pomiar poboru mocy, ale również na gromadzenie szerokiego spektrum danych statystycznych podczas wykonywania programu.. Ułatwia to projektantowi systemu mikroprocesorowego optymalizacje zużycia energii na każdym etapie budowy. Rysunek Wykonanie badań testowych i analiza zużycia energii.36 przedstawia schemat podłączenia urządzenia do mikrokontrolera Istotnym ograniczeniem jest kompatybilność jedynie ze środowiskiem programowania oferowanym przez producenta urządzenia Keil.




*Rysunek Wykonanie badań testowych i analiza zużycia energii.36. Schemat pomiaru przy pomocy ULINKplus*

Kolejnym z dostępnych na rynku urządzeń jest zestaw Power Profiler Kit II. Jest to autonomiczny zestaw do pomiaru poboru mocy urządzeń embedded. Urządzenie może być używane w trybie amperomierza oraz źródła zasilania. Deklarowany przez producenta zakres pomiarowy od 200nA do 1A. Schemat podłączenia układu przedstawiono na Rysunek Wykonanie badań testowych i analiza zużycia energii.37. Do zakresu pomiarowego od 500mA do 1 A potrzebne są dwa przewody zasilające USB 5 V. Dodatkowo posiada osiem wejść cyfrowych, które mogą posłużyć jako analizator stanów logicznych. Firma Nordic Semicondutros wspiera swoje rozwiązania zestawem narzędzi Online Moc Profiler. Oprogramowanie szacuje zużycie energii urządzeń wykorzystujących protokoły komunikacyjne Internetu Rzeczy, Bluetooth Low Energy oraz technologii LTE-M.



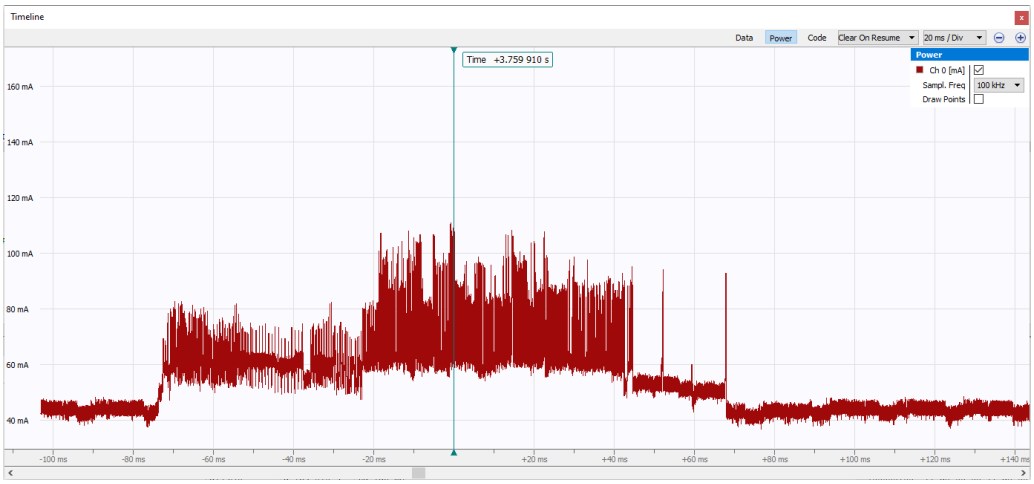

*Rysunek Wykonanie badań testowych i analiza zużycia energii.37. Schemat układu Power Profiling Kit II*

Do programowania oraz przeprowadzenia testów systemu mikroprocesorowego wykorzystano emulator USB – JTAG/SWD J-Link Ultra + firmy Segger przedstawiony na Rysunek Wykonanie badań testowych i analiza zużycia energii.38. Urządzenie to jest dedykowane mikrokontrolerom z rdzeniem ARM7/9, Cortex oraz Renesans RX. Szybkość

programowania sięga 720 kB/s. Urządzenie to zostało wybrane ze względu na możliwość pomiaru mocy. Oferuje ono wysoką częstotliwość próbkowania 100 kHz z dokładnością do 50 µA. Podłączany jest przy pomocy interfejsu USB do komputera PC natomiast do badania wykorzystywane jest dwudziestopinowe wyprowadzenie JTAG/SWD.

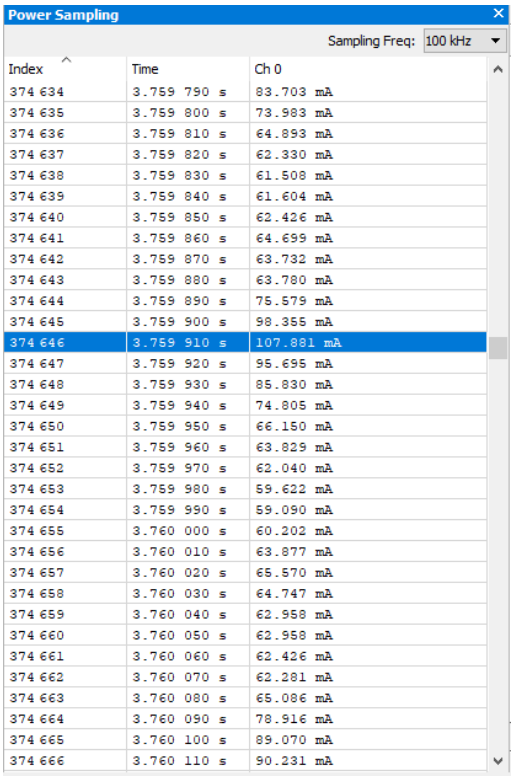

*Rysunek Wykonanie badań testowych i analiza zużycia energii.38. J-Link Ultra+*

Wsparciem dla narzędzi sprzętowy jest oprogramowanie Ozone. Jest to debugger programowy dla aplikacji urządzeń embedded napisanych w języku C/C++. Oferuje przyjazne użytkownikowi środowisko graficzne. Kluczowymi komponentami jest okno „Timeline” pokazane na rysunku . W czasie rzeczywistym tworzone są wykresy zmiennych oraz poboru mocy urządzenia. Próbkowanie pomiaru pomocy może odbywać się z częstotliwością od 1 kHz do 100 kHz. Wykres przedstawiony na zawiera informacje o poborze prądu całego systemu w czasie wykonywania funkcji *MemChc* katy pamięci umieszczonej w czytniku obsługiwanym przez interfejs SPI.




*Rysunek Wykonanie badań testowych i analiza zużycia energii.39. Wykres poboru prądu funkcji sprawdzenia pamięci przez interfejs SPI.*

Po za graficznym przedstawieniem przebiegu tworzone są statystyki pomiaru. Przedstawione na Rysunek Wykonanie badań testowych i analiza zużycia energii.40. okno „Power Sampling” tworzy tabelę z trzema kolumnami. W pierwsze umieszczony jest index pomiaru, w drugiej czas liczony w sekundach od rozpoczęcia pomiaru. W ostatnie wartość natężenia prądu w danej próbce liczona w miliamperach. Częstotliwość próbkowania została ustawiona na 100 kHz. Po zakończonym pomiarze tabele można eksportować do pliku z rozszerzeniem csv. Pozwala to na łatwe przekonwertowanie pliku do arkusza Excel oraz przeliczenia najważniejszych właściwości takich jak średni, minimalny czy maksymalny pobór mocy przez system.




| Index | Time | Ch 0 |
|---|---|---|
| 374 634 | 3.759 790 s | 83.703 mA |
| 374 635 | 3.759 800 s | 73.983 mA |
| 374 636 | 3.759 810 s | 64.893 mA |
| 374 637 | 3.759 820 s | 62.330 mA |
| 374 638 | 3.759 830 s | 61.508 mA |
| 374 639 | 3.759 840 s | 61.604 mA |
| 374 640 | 3.759 850 s | 62.426 mA |
| 374 641 | 3.759 860 s | 64.699 mA |
| 374 642 | 3.759 870 s | 63.732 mA |
| 374 643 | 3.759 880 s | 63.780 mA |
| 374 644 | 3.759 890 s | 75.579 mA |
| 374 645 | 3.759 900 s | 98.355 mA |
| 374 646 | 3.759 910 s | 107.881 mA |
| 374 647 | 3.759 920 s | 95.695 mA |
| 374 648 | 3.759 930 s | 85.830 mA |
| 374 649 | 3.759 940 s | 74.805 mA |
| 374 650 | 3.759 950 s | 66.150 mA |
| 374 651 | 3.759 960 s | 63.829 mA |
| 374 652 | 3.759 970 s | 62.040 mA |
| 374 653 | 3.759 980 s | 59.622 mA |
| 374 654 | 3.759 990 s | 59.090 mA |
| 374 655 | 3.760 000 s | 60.202 mA |
| 374 656 | 3.760 010 s | 63.877 mA |
| 374 657 | 3.760 020 s | 65.570 mA |
| 374 658 | 3.760 030 s | 64.747 mA |
| 374 659 | 3.760 040 s | 62.958 mA |
| 374 660 | 3.760 050 s | 62.958 mA |
| 374 661 | 3.760 060 s | 62.426 mA |
| 374 662 | 3.760 070 s | 62.281 mA |
| 374 663 | 3.760 080 s | 65.086 mA |
| 374 664 | 3.760 090 s | 78.916 mA |
| 374 665 | 3.760 100 s | 89.070 mA |
| 374 666 | 3.760 110 s | 90.231 mA |

*Rysunek Wykonanie badań testowych i analiza zużycia energii.40. Okno Power Sampling*

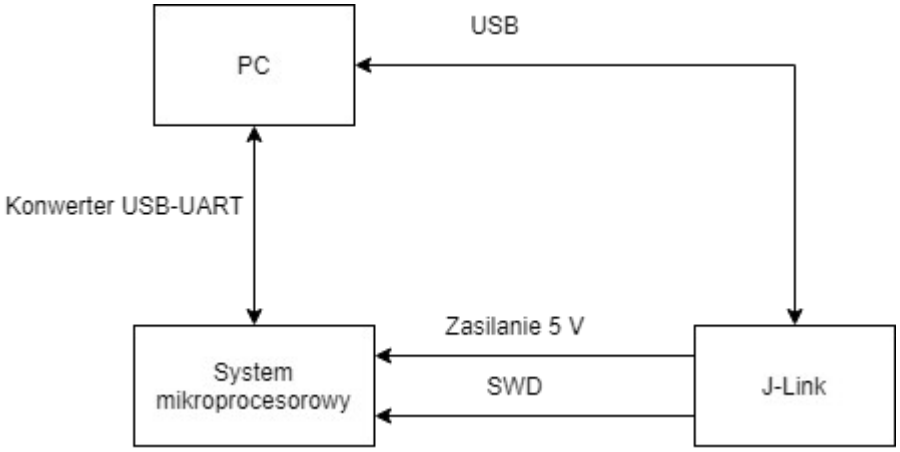
Aby umożliwić pomiar poboru prądu należy w konsoli programu Ozone, przed rozpoczęciem pomiaru wykonać polecenie:

Edit.SysVar(VAR_TARGET_POWER_ON,1);

Jest to polecenie dla urządzenia J-Link by zasilić włączyć zasilanie urządzenia docelowego na wyprowadzeniu numer 19.

## 7.2. Układ pomiarowy

Schemat pomiarowy przedstawiony na Rysunek Wykonanie badań testowych i analiza zużycia energii.41 składa się z trzech części. Komputer PC jest urządzeniem sterującym dla systemu mikroprocesorowego. Polecenia wysyłane są przez konwertera UART USB. Urządzenie pomiarowe w postaci J-Link Ultra plus przy pomocy przewodu USB. Natomiast emulator podłączony jest do urządzenia przez interfejs SWD oraz jedno wyprowadzenie zasilające.




*Rysunek Wykonanie badań testowych i analiza zużycia energii.41. Schemat układu pomiarowego.*

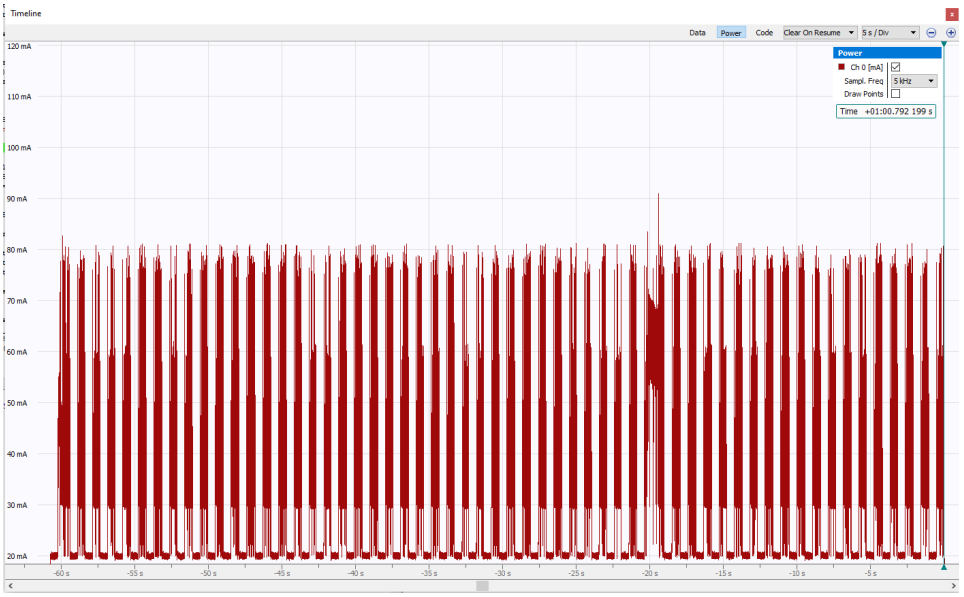
Szczegółowe połączenia urządzenia pomiarowego z systemem zostało przedstawione w Tabela Wykonanie badań testowych i analiza zużycia energii.16. Ponadto na płytce mikrokontrolera STM32 Nucleo należy przełączyć zworkę JP6 w położenie E5V. Pozwoli to na zasila systemu z programatora J-Link przez wyprowadzenie numer 19.

*Tabela Wykonanie badań testowych i analiza zużycia energii.16.Połączenie J-Link z Nucleo*

| Numer wyprowadzenia J-Link | Wyprowadzenie J-Link | Numer wyprowadzenia złącza CN4 Nucleo | Wyprowadzenie Nucleo |
|---|---|---|---|
| 1 | Vtref | 1 | VDD_Target |
| 4 | GND | 3 | GND |
| 7 | TNS | 4 | SWDIO |
| 9 | TCK | 2 | SWCKL |
| 13 | TDO | 6 | SWO |
| 15 | Reset | 5 | NRST |
| 19 | 5V-Supply | - | E5V |

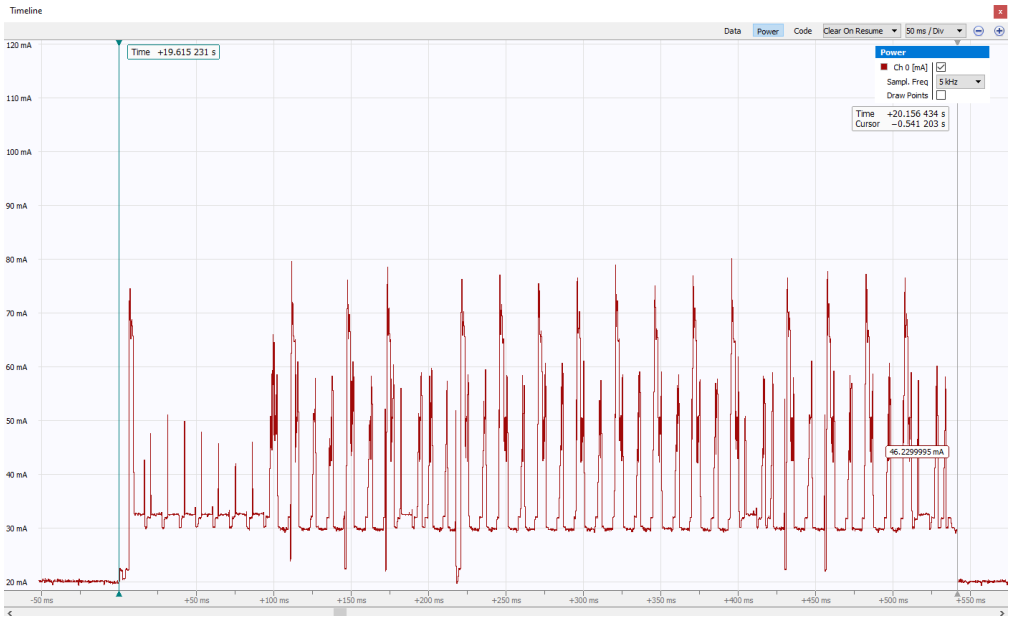
## 7.3. Pomiary przez SPI

Do pierwszego pomiaru przy pomocy interfejsu SPI został wykorzystana karta pamięci firmy SanDisck serii Edge o pojemności 16 GB. Posiada ona standard pamięci SD-HC oraz klasę szybkości Speed Class 4. Czas próbkowania został ustawiony na 5 kHz. Na Rysunek Wykonanie badań testowych i analiza zużycia energii.42 został przedstawiony zrzut ekranu wykresu pomiaru prądu w czasie zapisu prowadzonego przez jedną minutę.




*Rysunek Wykonanie badań testowych i analiza zużycia energii.42. Pomiar karty SanDisc 16Gb Speed Class 4*

Wyraźne skoki napięcia jest to moment inicjalizacji karty pamięci oraz zapisania paczki danych. Przybliżenie pomiaru jednego zapisu przedstawiono na Rysunek Wykonanie badań testowych i analiza zużycia energii.43. Na podstawie kursorów dostępnych w oprogramowaniu zmierzono czas zapisu jednej paczki danych wynosi 0,541 s.

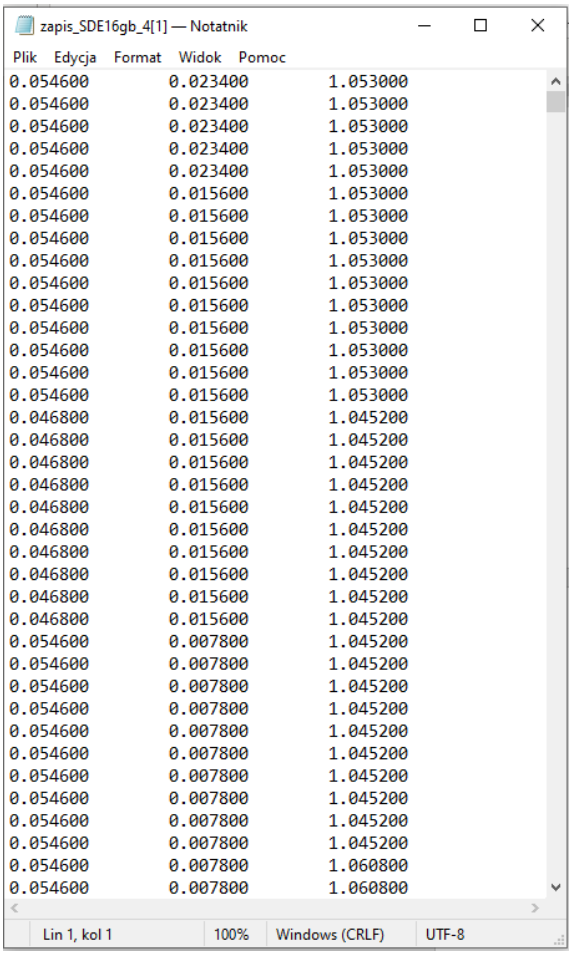



*Rysunek Wykonanie badań testowych i analiza zużycia energii.43. Zapis jednej paczki danych.*

Na podstawie wyeksportowanych danych ze środowiska Ozone obliczono najważniejsze wartości pomiaru. W czasie zapisu najwyższa odnotowana wartość poboru

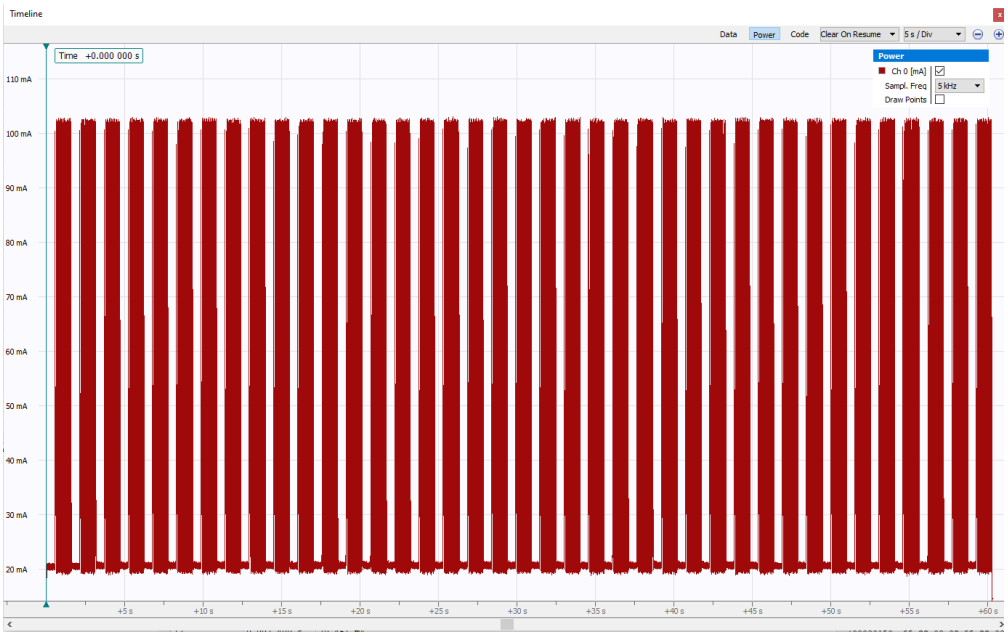
prądu to 91,05 mA natomiast najniższa to 15,706 mA. Średni pobór prądu podczas zapisu to 29,29 mA.

Plik którego fragment załączono na Rysunek Wykonanie badań testowych i analiza zużycia energii.44 zawiera 29696 wierszy danych. Łączna wielkość pliku tekstowego to 812 KB.



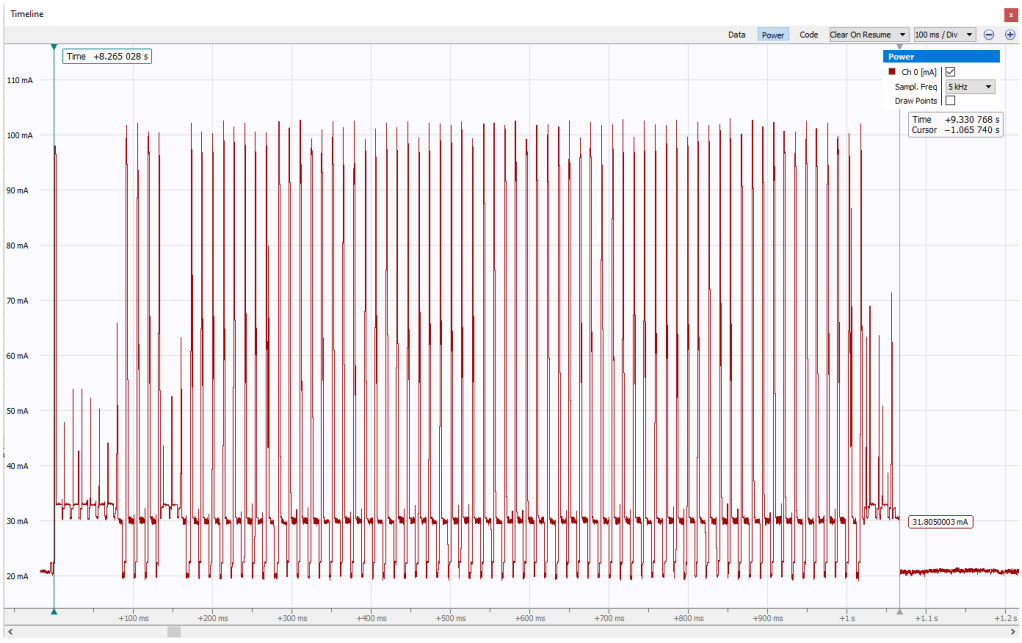

*Rysunek Wykonanie badań testowych i analiza zużycia energii.44. Fragment zapisu danych z akcelerometru.*

Kartę tą również wykorzystano do pomiaru z funkcją szyfrowania danych. Przebieg pomiaru przedstawiono na Rysunek Wykonanie badań testowych i analiza zużycia energii.45. Średni pobór prądu to 31,269 mA maksymalny 103,115 mA a minimalny 14,183.




*Rysunek Wykonanie badań testowych i analiza zużycia energii.45 Zapis szyfrowanych danych na karcie SanDisck ExtremePro 64 GB*

Przebieg zapisu pojedynczej paczki danych przedstawiono na Rysunek Wykonanie badań testowych i analiza zużycia energii.46. Czas zapisu to 1,065 s. Plik zawierające zaszyfrowane dane zawiera 19456 wierszy tekstu oraz zajmuje 1,24 MB pamięci. Tabela Wykonanie badań testowych i analiza zużycia energii.17 przedstawia przykładowe wartości zaszyfrowane i umieszczone w pliku.



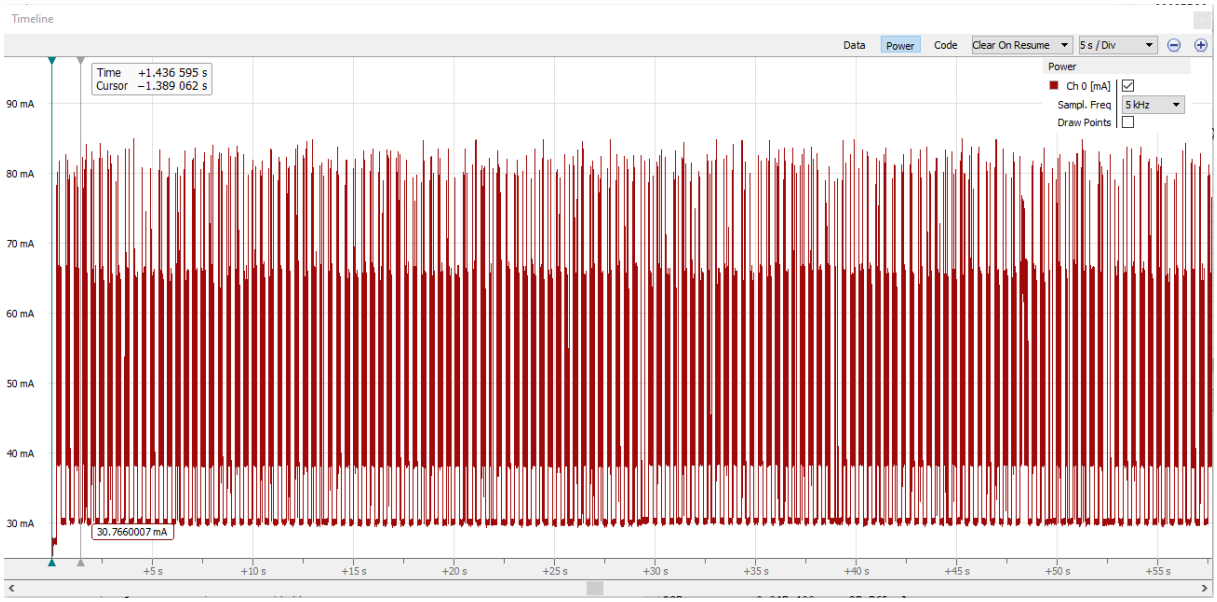

*Rysunek Wykonanie badań testowych i analiza zużycia energii.46. Zapis pojedynczej paczki zaszyfrowanych danych na karcie SanDisck ExtremePro 64 GB*

*Tabela Wykonanie badań testowych i analiza zużycia energii.17. Przykładowe zaszyfrowane wartosci zapisu.*

| | | |
|---|---|---|
| 1013.999756 | 36634168769812784742<br>4.000000 | -163418695249428480.000000 |
| 366345909822778507264.000000 | 0.000000 | -<br>320170945981609402351914844<br>16.000000 |
| 0.000000 | -<br>69619771506688.000000 | 4214116057088.000000 |

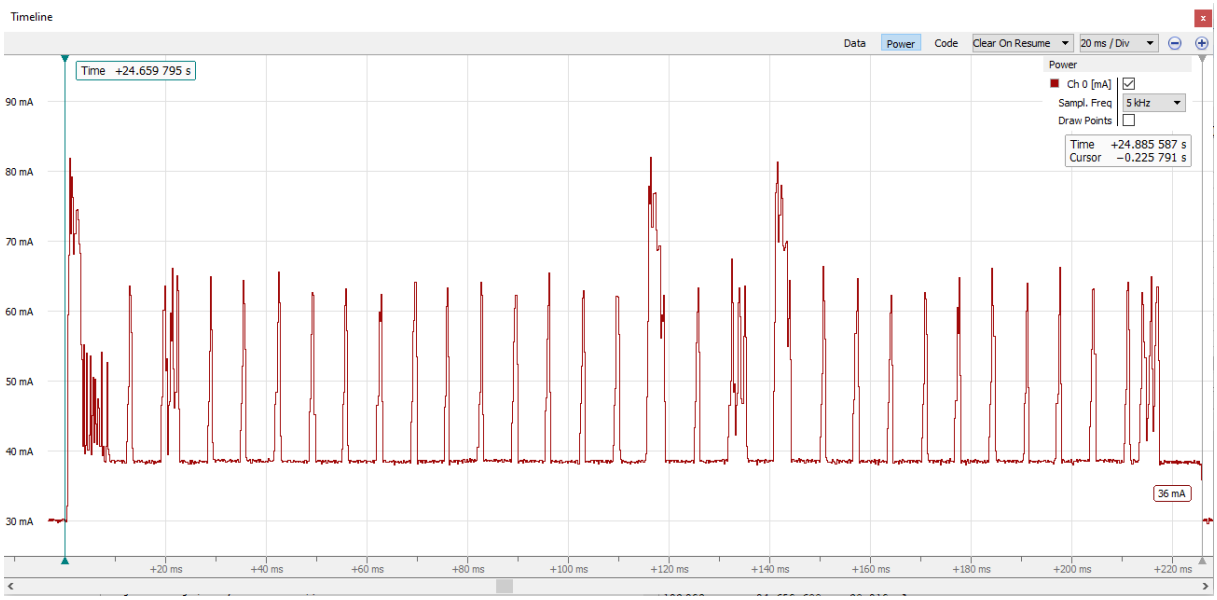
## 7.4. Pomiar bez przez SD

Pomiary przy pomocy interfejsu SD przeprowadzono z wykorzystaniem jednej linii transmisji danych. Częstotliwość próbkowania sygnału to 5 kHz. Przykładowy przebieg poboru prądu podczas zapisu przedstawiono na Rysunek Wykonanie badań testowych i analiza zużycia energii.47. Pomiar przeprowadzono na czterech kartach firmy SanDisk o różnych właściwościach.



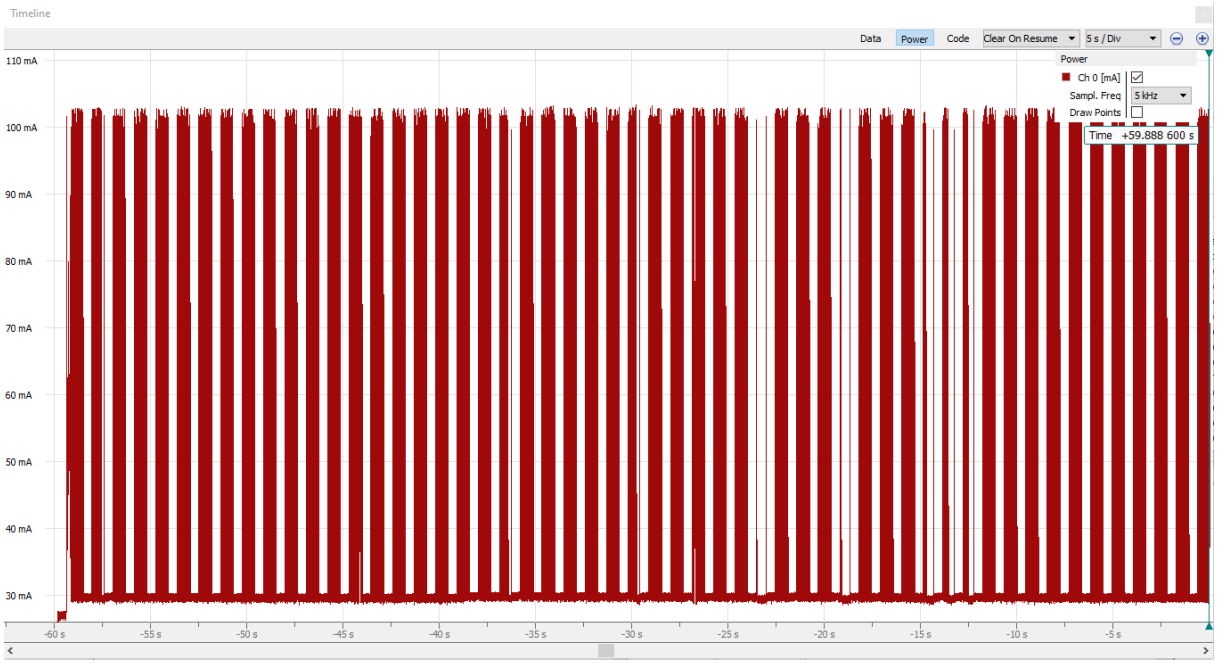

*Rysunek Wykonanie badań testowych i analiza zużycia energii.47. Wykres zapisu na kracie SanDisk 16 GB SDHC*

Rysunek Wykonanie badań testowych i analiza zużycia energii.48 przedstawia zapis jednej paczki danych na kartę pamięci. Zapis ten trwa około 0,225 sekundy co zmierzono kursorami narzędzia Ozone. Został on przeprowadzony na karcie SanDisk o pojemności 16 GB w standardzie SDHC oraz klasie szybkości Speed Class 10.




*Rysunek Wykonanie badań testowych i analiza zużycia energii.48. Zapis jednej paczki danych przez interfejs SD 1-kanał*

W celu porównania wydajności na karcie SanDisk Extreme Pro przeprowadzono zapis szyfrowanych danych przy pomocy interfejsu SD z jednym kanałem transmisji.




*Rysunek Wykonanie badań testowych i analiza zużycia energii.49. Zapis szyfrowanych danych interfejs SD 1-kanał*

## 7.5. Interpretacja wyników

Pomiary dla dwóch interfejsów przeprowadzono w takich samych warunkach działania całego systemu mikroprocesorowego. Tabela Wykonanie badań testowych i analiza zużycia energii.18. Wyniki pomiaru dla interfejsu SPIprzedstawia wyniki pomiarów czterech kart podczas używania interfejsu SPI. Dodatkowo został przeprowadzony pomiar zapisu szyfrowanych danych na karcie Extreme Pro. Wyniki

pomiaru zostały zawarte w ostatnim wierszu tabeli. Kary uszeregowano od najwolniejszej klasy szybkości do najwyższej. Różnica średniego poboru prądu między najlepiej wypadającą kartą Extreme, a najbardziej energochłonną Edge Class 10 wyniosła 5,377 mA. Istotne znaczenie ma ilość zapisanych danych. Karta o najlepszym wyniku energooszczędności zapisała tylko 16 kB mniej w tym samym czasie trwania pomiaru. Zapis szyfrowanych danych różnił się jedynie o 11 µA przy wzroście ilości zapisanych danych o 349 kB.

*Tabela Wykonanie badań testowych i analiza zużycia energii.18. Wyniki pomiaru dla interfejsu SPI*

| Firma | Seria | Pojemność [GB] | Klasa szybkości | Standard SD | Średni pobór prądu [mA] | Maksymalny pobór prądu [mA] | Wielkość bufora [B] | Czas pomiaru [s] | CPU CLK [MHz] | Wielkość pliku po zapisie [MB] |
|---|---|---|---|---|---|---|---|---|---|---|

W Tabela Wykonanie badań testowych i analiza zużycia energii.19 przedstawiono wyniki pomiaru dla interfejsu SD z jedną linią transmisji danych. Pomiar został przeprowadzony dla tych samych kart co dla interfejsu SPI. W porównaniu do poprzednich wyników średni pobór prądu wzrósł dla wszystkich badanych przypadków. Był to wzrost średnio o 4,229 mA. Objętość plików również wzrosła średnio o 362 kB. Największą energooszczędnością wykazała się karta Extreme Pro osiągając wynik średniego poboru prądu 34,327 mA. Zarazem podczas tego pomiaru została zapisana największa ilość danych spośród wszystkich pomiarów w tych samych warunkach. Zapis szyfrowanych zwiększył swoją objętość o 0,453 MB, natomiast różnica w poborze prądu zwiększyła się o 6,001 mA w stosunku do interfejsu SPI.

*Tabela Wykonanie badań testowych i analiza zużycia energii.19. Wyniki pomiaru dla interfejsu SD 1-liniowy*

| Firma | Seria | Pojemność [GB] | Klasa szybkości | Standard SD | Średni pobór prądu [mA] | Maksymalny pobór prądu [mA] | Wielkość bufora [B] | Czas pomiaru [s] | CPU CLK [MHz] | Wielkość pliku po zapisie [MB] | Ty szyfr ni |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SanDisk | Edge | 16 | Class 4 | SDHC | 35,809 | 86,8 | 6144 | 60 | 80 | 1,171 | - |
| | Edge | 16 | Class 10 | SDHC | 34,841 | 95,19 | | | | 1,252 | |
| | Extreme | 64 | Video 30 | SDXC | 34,538 | 106,34 | | | | 1,263 | |
| | Extreme Pro | 64 | Video 60 | SDXC | 34,327 | 104,151 | | | | 1,315 | |
| | Extreme Pro | 64 | Video 60 | SDXC | 37,27 | 103,643 | | | | 1,73 | AES |

# 8. Podsumowanie

# Bibliografia


[1] K. Paprocki, Mikrokontrolery STM32 w praktyce, Legionowo: BTC, 2011.

[2] Embedded Microprocessor Benchmark Consortium, About EEMBC, https://www.eembc.org/about/, 15.03.2021.

[3] Embedded Microprocessor Benchmark Consortium, „EEMBC Benchmarks," 15 03 2021. [Online]. Available: https://www.eembc.org/products/.

[4] Embedded Microprocessor Benchmark Consortium, „ULPMark-CM Scores," EMBC, 15 03 20201. [Online]. Available: https://www.eembc.org/ulpmark/.

[5] NXP, „Kinetis® L Series: Ultra-Low Power Microcontrollers (MCUs) based on Arm® Cortex®-M0+ Core," 16 03 2021. [Online]. Available: https://www.nxp.com/products/processors-and-microcontrollers/arm-microcontrollers/general-purpose-mcus/kl-series-cortex-m0-plus:KINETIS_L_SERIES.

[6] NXP, „LPC541XX: Low-Power Microcontrollers (MCUs) Based on Arm® Cortex®-M4 Cores With Optional Cortex®-M0+ Co-processor," 16 03 2021. [Online]. Available: https://www.nxp.com/products/processors-and-microcontrollers/arm-microcontrollers/general-purpose-mcus/lpc54000-cortex-m4-/low-power-microcontrollers-mcus-based-on-arm-cortex-m4-cores-with-optional-cortex-m0-plus-co-processor:LPC541XX.

[7] Silicon Labs, „Microcontrollers (MCUs) for Battery Operated Embedded Devices," 16 03 2021. [Online]. Available: https://www.silabs.com/solutions/battery-operation.

[8] Silicon Labs, „32-bit Microcontrollers," 16 03 2021. [Online]. Available: https://www.silabs.com/mcu/32-bit.

[9] Silicon Labs, „SLSTK3701A EFM32 Giant Gecko S1, GG11 Starter Kit," 16 03 2021. [Online]. Available: https://www.silabs.com/development-tools/mcu/32-bit/efm32gg11-starter-kit.

[10] STMicroelectronics, STM32L series Ultra-low-power 32-bit MCUs, life.augmented, Styczeń 2019.

[11] STMicroelectronics, „STM32U5 series of ultra-low-power MCUs," 17 03 2021. [Online]. Available: https://www.st.com/content/st_com/en/products/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus/stm32-ultra-low-power-mcus/stm32u5-series.html#overview.

[12] „Microvisor and STM32U5, TheBest Performance-per-WattMCU,1st to Support aNew IoT Development Paradigm," 4 03 2021. [Online].

[13] M. Galewski, STM32 Aplikacja i ćwiczenia w języku C z biblioteką HAL, Legionowo: BTC, 2019.

[14] SD Association, „Capacity (SD/SDHC/SDXC/SDUC)," [Online]. Available: https://www.sdcard.org/developers/sd-standard-overview/capacity-sd-sdhc-sdxc-sduc/. [Data uzyskania dostępu: 23 03 2021].

[15] SD Association, „Capacity (SD/SDHC/SDXC/SDUC)," [Online]. Available: https://www.sdcard.org/developers/sd-standard-overview/capacity-sd-sdhc-sdxc-sduc/. [Data uzyskania dostępu: 25 03 2021].

[16] S. Association, „Speed Class," [Online]. Available: https://www.sdcard.org/developers/sd-standard-overview/speed-class/. [Data uzyskania dostępu: 23 03 2021].

[17] SD Association, „SD Specifications Part 1 Physical Layer Simplified Specification," 23 9 2020. [Online]. [Data uzyskania dostępu: 25 3 2021].

[18] T. Jabłoński, „Obsługa kart pamięciowych SD, część 3," *Elektronika Praktyczna*, p. 6, 2 2008.

[19] Analog Devices, „Digital Accelerometer ADX345”.
[20] S. Prata, Język C Szkoła programowania, Helion, 2014.
[21] STMicroelectronics, „UM0586 STM32 Cryptographic Library,” wrzesień 2013. [Online]. Available: https://www.st.com/resource/en/user_manual/cd00208802-stm32-cryptographic-library-stmicroelectronics.pdf. [Data uzyskania dostępu: 2021 kwiecień 16].
[22] „A Security site,” [Online]. Available: https://asecuritysite.com/.