

AGH

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

WYDZIAŁ INFORMATYKI ELEKTRONIKI I TELEKOMUNIKACJI

KATEDRA TELEKOMUNIKACJI

Praca magisterska

*Opracowanie i ocena bezprzewodowego, miniaturowego czujnika
zasilanego bateryjnie*

*Implementation and evaluation of long-lasting, battery powered,
wireless sensor*

Autor:

Paweł Kazimierowicz

Kierunek studiów:

Teleinformatyka

Opiekun pracy:

dr. inż. Jarosław Bułat

Kraków, 2019

Uprowadzony(-a) o odpowiedzialności karnej na podstawie art. 115 ust. 1 i 2 ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2018 r. poz. 1191 z późn. zm.): „Kto przywłaszcza sobie autorstwo albo wprowadza w błąd co do autorstwa całości lub części cudzego utworu albo artystycznego wykonania, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 3. Tej samej karze podlega, kto rozpowszechnia bez podania nazwiska lub pseudonimu twórcy cudzy utwór w wersji oryginalnej albo w postaci opracowania, artystyczne wykonanie albo publicznie zniekształca taki utwór, artystyczne wykonanie, fonogram, wideogram lub nadanie.”, a także uprowadzony(-a) o odpowiedzialności dyscyplinarnej na podstawie art. 307 ust. 1 ustawy z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce (Dz. U. z 2018 r. poz. 1668 z późn. zm.) „Student podlega odpowiedzialności dyscyplinarnej za naruszenie przepisów obowiązujących w uczelni oraz za czyn uchybiający godności studenta.”, oświadczam, że niniejszą pracę dyplomową wykonałem(-am) osobiście i samodzielnie i nie korzystałem(-am) ze źródeł innych niż wymienione w pracy.

Jednocześnie Uczelnia informuje, że zgodnie z art. 15a ww. ustawy o prawie autorskim i prawach pokrewnych Uczelni przysługuje pierwszeństwo w opublikowaniu pracy dyplomowej studenta. Jeżeli Uczelnia nie opublikowała pracy dyplomowej w terminie 6 miesięcy od dnia jej obrony, autor może ją opublikować, chyba że praca jest częścią utworu zbiorowego. Ponadto Uczelnia jako podmiot, o którym mowa w art. 7 ust. 1 pkt 1 ustawy z dnia 20 lipca 2018 r. – Prawo o szkolnictwie wyższym i nauce (Dz. U. z 2018 r. poz. 1668 z późn. zm.), może korzystać bez wynagrodzenia i bez konieczności uzyskania zgody autora z utworu stworzonego przez studenta w wyniku wykonywania obowiązków związanych z odbywaniem studiów, udostępniać utwór ministrowi właściwemu do spraw szkolnictwa wyższego i nauki oraz korzystać z utworów znajdujących się w prowadzonych przez niego bazach danych, w celu sprawdzania z wykorzystaniem systemu antyplagiatowego. Minister właściwy do spraw szkolnictwa wyższego i nauki może korzystać z prac dyplomowych znajdujących się w prowadzonych przez niego bazach danych w zakresie niezbędnym do zapewnienia prawidłowego utrzymania i rozwoju tych baz oraz współpracujących z nimi systemów informatycznych.

Spis treści

Wstęp	7
1. Wprowadzenie	9
1.1. Przegląd rozwiązań dostępnych na rynku.....	9
1.2. Architektura rozwiązania.....	11
1.3. Rozważane platformy sprzętowe.....	13
1.3.1. AVR ATmega	13
1.3.2. ESP8266.....	14
1.3.3. ESP32.....	14
1.4. Dostępne rozwiązania komunikacji radiowej.....	16
1.4.1. WiFi.....	16
1.4.2. Bluetooth Low Energy	16
1.4.3. Moduły radiowe pracujące w paśmie nielicencjonowanym.....	17
1.4.4. Komunikacja w oparciu o sieci telekomunikacyjne - NarrowBand-IoT.....	18
1.5. Sensory temperatury, wilgotności, ciśnienia	20
1.6. Zasilanie bezprzewodowych sensorów.....	21
1.6.1. Wybrane rodzaje baterii	22
1.7. Ocena stanu naładowania baterii / akumulatorów	24
2. Prototypy platformy pomiarowej	27
2.1. Architektura rozwiązania, wybór konfiguracji sprzętowych.....	27
2.1.1. Wybrane konfiguracje sprzętowe prototypów.....	28
2.2. Wykonanie prototypów.....	29
2.2.1. Prototypy oparte o mikrokontroler AVR ATmega	29
2.2.2. Prototyp z komunikacją WiFi - ESP8266	32
2.2.3. Sensor z komunikacją NarrowBand-IoT.....	33
2.3. Sensor pomiarowy	36
2.4. Zasilanie układu.....	36
2.5. Oprogramowanie prototypów	36

3. Metodologia badawcza i przeprowadzenie testów poboru energii	39
3.1. Czynniki decyzyjne w porównaniu prototypów.....	39
3.2. Środowisko pomiarowe.....	40
3.3. Optymalizacja działania prototypów.....	41
3.4. Konfiguracja usypiania ATmega328p oraz sensora BME280.....	41
3.5. Prototyp ATmega328p z modułem komunikacyjnym RFM69.....	44
3.6. Prototyp ATmega328p z modułem komunikacyjnym RFM75.....	45
3.7. Prototyp ATmega328p z modułem Narrowband-IoT u-blox Sara N211.....	48
3.8. Komunikacja WiFi.....	50
3.8.1. Asocjacja z siecią WiFi.....	50
3.8.2. Przesyłanie pomiarów.....	53
3.9. Podsumowanie wyników przeprowadzonych pomiarów.....	55
3.9.1. Wybór platformy pomiarowej do dalszej pracy.....	55
4. Wykonanie sensora w docelowej wersji	57
4.1. Dodatkowe narzędzie do testowania - brama dostępowa.....	57
4.2. Ostateczna konfiguracja sprzętowa urządzenia.....	59
4.3. Optymalizacja zużycia energii finalnej konfiguracji sprzętowej.....	61
4.4. Zasilanie opracowanego czujnika.....	66
4.5. Finalna estymacja czasu pracy opracowanego czujnika na baterii.....	67
4.6. Kierunki dalszego rozwoju.....	68
Wnioski	71
Bibliografia	73

Wstęp

Koncept bezprzewodowej kontroli urządzeń elektronicznych znany jest już od wczesnych lat dwudziestych[1], jednak to właśnie w ostatnich latach zaobserwować można największy rozwój tego obszaru. Dzięki szybkiemu rozwojowi technologii coraz więcej zastosowań staje się osiągalnych i posiadających wdrożeniową wartość. Szacuje się, że ilość urządzeń podłączonych do Internetu wzrośnie do ponad 75 miliardów[2] w 2025 roku. Coraz więcej mówi się o tzw. Inteligentnych Miastach, których podstawą jest właśnie koncept Internetu Rzeczy[3], które służyć mają do optymalizacji funkcjonowania miast i systemów.

Podstawowym elementem koniecznym do funkcjonowania tego rodzaju rozwiązań są wszelkiego rodzaju czujniki dostarczające informacji o warunkach atmosferycznych, funkcjonowaniu maszyn i systemów, ruchu pojazdów i osób. Wyzwaniem technologicznym tego zagadnienia jest jednak zasilanie, którego - przy tak dynamicznym rozwoju - będzie wymagać ogromna ilość urządzeń. Wiele z nich znajduje się w miejscach odległych, szczególnie w telemetrii, śledzeniu zasobów, w których nie ma dostępu do stałego zasilania. Rozwiązaniem tej sytuacji, ale jednocześnie jednym z istotniejszych wyzwań Internetu Rzeczy jest opracowanie urządzeń, w których nie jest konieczna wymiana baterii, lub następuje to w dużych odstępach czasu[4].

Autor postanowił opracować bezprzewodowy czujnik warunków atmosferycznych niewielkich rozmiarów, zasilany bateryjnie, nie wymagający wymiany źródła zasilania częściej niż raz na dziesięć lat. W rozdziale pierwszym opisane są poszczególne komponenty, które mogą posłużyć do skonstruowania takiego urządzenia, oraz konfiguracje sprzętowe wybrane do przetestowania. W rozdziale drugim opisano proces przygotowania prototypów w oparciu o różne mikrokontrolery i interfejsy komunikacyjne. Wykonane zostały dedykowane płyty PCB dla każdej z konfiguracji, zawierające komponenty potrzebne do ich przetestowania w możliwie optymalny sposób - pozbawione dodatkowych elementów, które mogłyby wpłynąć negatywnie na poziom zużycia energii.

W rozdziale trzecim przedstawiono metodologię pomiarową i kryteria porównawcze zużycia energii przez prototypy. Przeprowadzono pomiary zużycia energii przez konfiguracje testowe oraz dokonano optymalizacji oprogramowania i peryferiów celem uzyskania możliwie dobrych wyników.

Po podsumowaniu i porównaniu wyników pomiarów wybrano najbardziej efektywną konfigurację z modułem radiowym w paśmie poniżej jednego gigaherca, a w rozdziale czwartym przedstawiono finalny projekt czujnika oraz wskazano drogi dalszego rozwoju projektu.

1. Wprowadzenie

W rozdziale pierwszym wykonany został przegląd dostępnych na rynku czujników atmosferycznych, celem zidentyfikowania potrzebnych funkcjonalności i formy dla budowanego rozwiązania.

Następnie, po określeniu architektury rozwiązania i koniecznych do jego opracowania komponentów, rozważono możliwe scenariusze.

1.1. Przegląd rozwiązań dostępnych na rynku

Celem zidentyfikowania rynkowych potrzeb, które miałyby zaspokajać projektowane urządzenie, sporządzony został przegląd dostępnych na rynku sensorów parametrów atmosferycznych pozwalających na przesyłanie danych do usługi chmurowej.

Produkty dedykowane konsumentom

Istnieje wiele konsumenckich produktów działających w oparciu o domową sieć WiFi lub komunikujących się bezpośrednio ze smartfonem użytkownika za pomocą interfejsu Bluetooth Low Energy.

Przykładem jednego z nich jest *Temp Stick*¹ - sensor temperatury i wilgotności pracujący w oparciu o WiFi, przesyłający dane do dedykowanej chmury producenta pozwalającej na monitorowanie i ustawianie alertów. Urządzenie zasilane jest standardowymi paluszkami AA, producent deklaruje 6-12 miesięcy czasu pracy na bateriach. Jego cena jest jednak wysoka (149 USD na stronie producenta²), a samo urządzenie ma format czarnego prostopadłościanu 8.9 x 3.8 x 3.18 cm.

Kolejne dostępne rozwiązanie to *Xiaomi Mi Smart Temp Sensor*. Produkt ten jest zdecydowanie bardziej estetyczny (biały dysk o średnicy 8cm i wysokości 1.5cm), a jego cena zdecydowanie niższa (od 8 do 16 USD zależnie od sklepu Internetowego). Wymaga on jednak do funkcjonowania dedykowanej bramy (w cenie około 30 dolarów), gdyż sam sensor komunikuje się za pomocą *ZigBee*³. Produkt ten posiada pewne ograniczenia, nie ma możliwości konfiguracji (np. interwału pomiarów), a dane przesyłane są bezpośrednio do usługi chmurowej producenta.

¹<https://tempstick.com/>

²Wszystkie ceny wspomniane w tej części pracy są to ceny obowiązujące 07.08.2018

³Zamknięty protokół przesyłu danych w sieciach bezprzewodowych w paśmie ISM

W Internecie znaleźć można wzmianki o polskim produkcie Clime, obiecującym pomiar bezprzewodowy temperatury w formie miniaturowych kostek przyklejanych do ściany. Produkt ten jednak nie odniósł sukcesu i nie jest dostępny na rynku, a jedyny ślad to opis na stronie biura projektowego⁴.

Autorowi wydaje się, że w 2014 roku, kiedy to Clime zadebiutował, dostępna technologia mogła być nie wystarczająca by dotrzymać obietnic, które ten produkt składał.

Produkty do zastosowań profesjonalnych

Istnieją również produkty o typowo przemysłowym zastosowaniu, np. firmy Monnit⁵, jednak ich ceny są wysokie - od 50 do 150 dolarów, zależnie od wersji zasilania oraz modułu komunikacyjnego. Jeszcze wyższa jest cena bramy dedykowanej sensorom pracującym w paśmie ISM sub-GHz - 230 dolarów⁶.

Autorowi znana jest również polska firma Efento⁷, produkująca rejestratory parametrów fizycznych (m. in. warunków atmosferycznych). Rozwiązanie w formie prostopadłościanu 71 x 71 x 27 mm, zasilane bateriami litowymi o deklarowanym czasie pracy do 2 lat komunikuje się za pomocą Bluetooth Low Energy. Bramę do zbierania danych może stanowić smartfon lub dedykowane urządzenie. Urządzenie z sensorem temperatury dostępne jest w cenach od 330 do 700 PLN, w zależności od tego czy urządzenie jest certyfikowane i czy cena uwzględnia dostęp do usługi chmurowej. Brama dostępowa dedykowana rozwiązaniu dostępna jest w cenie 700 PLN.

Motywacja dla budowy własnego rozwiązania

Podsumowując, urządzenia dostępne na rynku dzielą się na produkty typowo konsumenckie - o ograniczonych możliwościach, czasie życia baterii, zróżnicowanych rozmiarach - oraz rozwiązania profesjonalne, o większej dokładności i bogatszej funkcjonalności, jednak o bardzo wysokich cenach. Celem autora było opracowanie platformy sensorowej pozwalającej na adaptację do konkretnych zastosowań (w podstawowej opcji do pomiaru parametrów atmosferycznych, takich jak temperatura czy wilgotność powietrza), o czasie pracy na baterii minimum dwóch lat, przy zachowaniu niskiego kosztu komponentów.

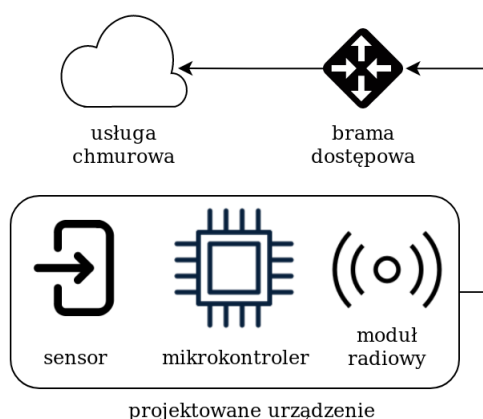
⁴<http://vorm.io/works/clime/>

⁵<https://www.monnit.com/Product/MNS-8-W1-TS-ST>

⁶<https://www.monnit.com/Products/Gateways/Ethernet-Gateway/Wireless-Ethernet-Gateways>

⁷<https://efento.pl>

1.2. Architektura rozwiązania



Rys. 1.1. Diagram architektury proponowanego rozwiązania

Na rysunku 1.1 zaprezentowano architekturę tworzonego rozwiązania, na które składać będą się następujące komponenty:

- **sensor** - układ pomiarowy dla danego parametru fizycznego (np. sensor warunków atmosferycznych, akcelerometr, sensor światła),
- **mikrokontroler** - scalony układ elektroniczny obsługujący logikę urządzenia, dokonujący okresowych pomiarów oraz odpowiedzialny za transmisję danych,
- **interfejs komunikacyjny** - moduł radiowy wraz z dedykowanym protokołem,
- **brama dostępową** - urządzenie przekazujące dane przesyłane przez sensory do usługi chmurowej,
- **usługa chmurowa** - usługa do przechowywania i analizy danych przesyłanych przez sensory.

Rozwiązania należące do nurtu Internetu Rzeczy mierzą wiele różnych parametrów fizycznych,

Sensor

Rozwiązania należące do nurtu Internetu Rzeczy mierzą wiele różnych parametrów fizycznych, które mogą pozwolić na automatyzację procesów czy też wykrywanie awarii. W przypadku monitorowania maszyn przemysłowych może być to zmiana pola magnetycznego, drgania, natężenie dźwięku. W przypadku systemów bezpieczeństwa będzie to ruch czy przerwanie obwodu elektrycznego przez kontaktron.

Dla potrzeb pracy jako przykład wybrano pomiar warunków atmosferycznych (temperatura, wilgotność, ciśnienie), gdyż jest to najbardziej powszechnie wykorzystywany zestaw danych - który zastosowany może zostać zarówno w przypadku rozwiązań konsumenckiego Inteligentnego Domu, jak również do monitorowania warunków przechowywania czy też optymalizacji użycia energii na ogrzewanie / chłodzenie biur i budynków przemysłowych.

Należy jednak podkreślić, że projektowana platforma pomiarowa może zostać rozszerzona o innego rodzaju sensory i zaadaptowana do potrzeb szczególnego przypadku. Przykłady alternatywnych zastosowań autor przedstawi w ostatnim rozdziale pracy.

Mikrokontroler

Centralną częścią budowanej platformy pomiarowej jest mikrokontroler - miniaturowy układ scalony wyposażony w odpowiednie oprogramowanie. Układ ten pobiera dane z sensora (lub wielu sensorów), odpowiednio przygotowuje ramkę danych oraz przesyła poprzez interfejs komunikacyjny.

Interfejs komunikacyjny

Na interfejs komunikacyjny składa się moduł radiowy i dedykowany rozwiązaniu protokół opisujący format przesyłanych danych. Odpowiednia enkapsulacja przesyłanych danych pomiarowych jest bardzo istotna z punktu widzenia zużycia energii, gdyż w przypadku niektórych modułów radiowych czas aktywności modułu rośnie liniowo wraz z długością przesyłanej ramki danych - a jednocześnie podobnej zależności podlega zużycie energii. Należy rozważyć odpowiedni balans pomiędzy efektywnością energetyczną a zachowaniem gwarancji dostarczenia danych do bramy dostępowej.

Brama dostępowa

Urządzenie lub oprogramowanie, które tłumaczyć będzie protokół komunikacyjny projektowanego urządzenia na format wygodny do dalszej integracji i analizy. Krótkie, zoptymalizowane datagramy opisywane będą w wygodnym i czytelnym formacie JSON (*JSON, ang. JavaScript Object Notation - tekstowy format danych komputerowych*), oraz przesyłane dalej do usługi chmurowej.

Usługa chmurowa

Dostępne jest wiele usług chmurowych dla Internetu Rzeczy, o zróżnicowanych funkcjonalnościach. Dedykowane narzędzia można znaleźć w usłudze Microsoft Azure⁸, Amazon Web Services⁹ czy też IBM Watson¹⁰. Obszar ten nie wchodzi w zakres zainteresowania tej pracy, gdyż wybór konkretnej usługi zależy od konkretnego zastosowania rozwiązania. Inaczej będzie to wyglądać jeżeli celem opomiarowania jest np. Predictive Maintenance¹¹, a inaczej gdy sterować ma domowym termostatem.

⁸<https://azure.microsoft.com/en-us/services/iot-hub/>

⁹<https://aws.amazon.com/iot/>

¹⁰<https://www.ibm.com/internet-of-things>

¹¹ang. serwisowanie urządzeń przemysłowych oparte o przewidywanie ich uszkodzeń, na podstawie warunków pracy

1.3. Rozważane platformy sprzętowe

Producenci układów scalonych posiadają w swoich portfoliach bardzo dużą ilość mikrokontrolerów, które mogłyby posłużyć do opracowania miniaturowego bezprzewodowego urządzenia pomiarowego. Są to zarówno układy mniejsze, o ograniczonej ilości peryferii, pamięci ram i zegarach, jak i również mikrokontrolery trzydziestodwubitowe pozwalające na wykonywanie złożonych operacji arytmetycznych.

Autor pracy zdecydował się wykorzystać i przetestować kilka popularnych układów scalonych, dla których dostępne jest wsparcie w wygodnym w użyciu i darmowym środowisku PlatformIO¹². Autor wykluczył układy, których programowanie za pomocą narzędzi Wolnego Oprogramowania jest utrudnione lub niemożliwe.

Dodatkowo, wszystkie wykorzystane układy zachowują zgodność z platformą Arduino¹³. Pozwoliło to na wielokrotne użycie wspólnego kodu dla różnych platform, ograniczając w ten sposób czas implementacji. Środowisko Arduino udostępnia wspólną, czytelną warstwę abstrakcji (taką jak np. funkcje *digitalWrite()* czy *analogRead()*), dla wielu układów scalonych.

Zwrócono jednak uwagę iż wykorzystane środowisko posiada pewne ograniczenia i możliwy narzut w stosunku do wykorzystania natywnego zestawu deweloperskiego, założono więc konieczność analizy możliwych optymalizacji dla finalnej konfiguracji.

W szczególności - znane są autorowi przypadki, w których dostępne otwarte biblioteki opracowane są w sposób taki, by zadziałały zawsze jednak w sposób zupełnie nie optymalny. Przykładowo, jedna z bibliotek do obsługi sensora temperatury DS18B20 (wspomnianego w kolejnej sekcji), podczas każdego pomiaru przeszukuje całą magistralę, zamiast odwoływać się do adresu sensora, co drastycznie wydłuża czas wykonywanego pomiaru.

W pracy rozważono wykorzystanie następujących mikrokontrolerów:

- **AVR ATmega328p** - tani ośmiobitowy mikrokontroler umożliwiający pracę z dowolnym z testowanych modułów radiowych,
- **ESP8266** - SoC (*ang. System on Chip*) z komunikacją bezprzewodową WiFi,
- **ESP32** - następca ESP8266 zapewniający usprawnienia pozwalające na szybsze połączenie WiFi, oraz komunikację Bluetooth.

1.3.1. AVR ATmega

Układy z rodziny AVR ATmega są ośmiobitowymi mikrokontrolerami spopularyzowanymi przez szerokie występowanie w płytach rozwojowych platformy Arduino. Są łatwo dostępne, dobrze udokumentowane oraz przygotowane dostępne są na tą platformę biblioteki do większości popularnych modułów, sensorów oraz układów scalonych.

¹²PlatformIO jest to zintegrowane środowisko do programowania systemów wbudowanych z poziomu edytora Atom

¹³Arduino jest to zintegrowane środowisko do programowania mikrokontrolerów oraz zbiór układów rozwojowych i rozszerzeń

Do pracy nad projektem wykorzystany został najpopularniejszy układ ATmega328p w wersji SMD, przez wzgląd na jego łatwą dostępność, niską cenę, oraz szerokie zastosowanie. Do uruchomienia tego układu nie jest wymagana duża ilość komponentów zewnętrznych, a jedynie jeden rezystor podciągający (dla linii RESET) oraz oscylator kwarcowy z kondensatorami filtrującymi. Możliwe jest również wykorzystanie z wewnętrznego oscylatora RC, co ogranicza ilość komponentów zewnętrznych do jednego rezystora podciągającego, należy mieć jednak na uwadze znaczny dryf zegara wraz z temperaturą. Jest to istotne z punktu widzenia pomiaru czasu między wybudzeniami sensora, oraz może mieć negatywny wpływ na funkcjonowanie protokołów komunikacyjnych bez linii zegarowej.

Alternatywą mogłyby być również układy z rodziny ATtiny, np. ATtiny84, które posiadają dobrze udokumentowane funkcjonalności oszczędzania energii i głębokiego uśpienia, jednak zdecydowano iż wykorzystanie ATmega328p będzie bardziej uniwersalne (przez większą ilość wyprowadzeń i dostępnych interfejsów), a zarazem tańsze (układ ATmega328 dostępny jest w cenie trzykrotnie niższej niż ATtiny84, zakładając zakup kilkunastu sztuk). Zgodnie z dokumentacją wartości poboru prądu przez te dwa układy są porównywalne.

1.3.2. ESP8266

ESP8266 jest to SoC (*ang. System on Chip*) początkowo dostępny na rynku jako prosty moduł komunikacyjny WiFi obsługiwany za pomocą komend AT. Jednak dzięki udostępnionemu przez producenta SDK (*ang. Software Development Kit*) oraz bardzo atrakcyjnej cenie (około dwóch dolarów) - stał się bardzo popularny zarówno wśród hobbystów jak i użytkowników profesjonalnych, dzięki czemu powstała wokół niego bogata społeczność.

Układ wspierany jest przez środowisko Arduino, w ramach którego możliwe jest skorzystanie z większości funkcjonalności, w tym interesujących autora - obsługi WiFi oraz głębokiego uśpienia układu. Możliwe jest również agresywne zmodyfikowanie zachowania się układu by umożliwić rozgłaszanie komunikatów na warstwie fizycznej 802.11, bez asocjacji z punktem dostępowym. Taka konfiguracja nie została jednak wzięta pod uwagę, przez wzgląd na wysoce nieprzewidywalne zachowanie układu.

Układ występuje w wielu formach, różniących się anteną (mikropaskowa, ceramiczna, gniazdo UFL), oraz sposobem montażu (przewlekany, powierzchniowy). Wybrany został, zdaniem autora, najwygodniejszy w zastosowaniu układ w wersji ESP-12F. Jest to aktualna wersja modułu w wersji do montażu powierzchniowego z wyprowadzonymi 22 pinami oraz zintegrowaną anteną mikropaskową. W przypadku dodatkowej bramy MQTT opisanej w rozdziale trzecim wykorzystany został moduł w wersji ESP-07S, który posiada gniazdo pozwalające na wyprowadzenie anteny poza obudowę urządzenia.

1.3.3. ESP32

Jest to układ będący pewnego rodzaju następcą ESP8266, o większej pamięci oraz ilości peryferiów. z punktu widzenia projektu istotne jest występowanie w tym układzie ULP (Ultra Low Power Coprocessor), który między innymi umożliwia podtrzymanie połączenia WiFi w stanie głębokiego uśpienia, by

zmniejszyć czas potrzebny na reasocjację z punktem dostępowym po wybudzeniu - a zarazem ograniczyć czas potrzebny na przesłanie danych.

Układ ESP32 również jest wspierany przez środowisko Arduino, jednak w sposób ograniczony, przewidywane jest skorzystanie z ESP-IDF¹⁴, które umożliwi autorowi skorzystanie z zaawansowanych możliwości układu.

¹⁴ESP-IDF jest to zestaw bibliotek deweloperskich udostępniony przez producenta układów ESP8266/ESP32

1.4. Dostępne rozwiązania komunikacji radiowej

Komunikacja dla bezprzewodowego sensora zrealizowana może być na wiele sposobów, między którymi istnieją znaczne różnice w kontekście zasięgu, zużycia energii i przepustowości. Istotną różnicą między rozważanymi możliwościami jest też konieczność stworzenia dedykowanej infrastruktury (w przypadku modułów radiowych w paśmie ISM), możliwość wykorzystania zwykle już istniejącej (dla WiFi) oraz wykorzystanie infrastruktury operatorskiej (Nb-IoT / LTE-M), co wiąże się z koniecznością wykupienia abonamentu.

Autor rozważając poszczególne możliwości miał na uwadze budowę średniej wielkości sieci sensorowej korzystającej ze wspólnego interfejsu komunikacyjnego oraz jednego punktu styku usługą chmurową (brama dostępową).

1.4.1. WiFi

Popularnym interfejsem komunikacyjnym wykorzystywanym w Internecie Rzeczy jest WiFi, w szczególności w instalacjach domowych oraz amatorskich. Jest to technologia o łatwej do zbudowania, często już dostępnej architekturze, jednak nie jest ona optymalna ze względu na dość znaczne zużycie energii wynikające z konieczności każdorazowej asocjacji z punktem dostępowym.

Autor założył wykorzystanie WiFi jako przykład konfiguracji często wykorzystywanej w amatorskich instalacjach, jednak spodziewając się, że każdorazowe przesyłanie danych pomiarowych z sensora nie będzie optymalne, przez duży narzut energetyczny asocjacji z punktem dostępowym.

Jednocześnie, zwrócono uwagę, że w przypadku układu ESP8266 oraz ESP32 wspierane jest jedynie pasmo 2.4GHz, które jest mocno zawodne i obserwuje się jego wyłączenie w sieciach WiFi.

1.4.2. Bluetooth Low Energy

Bluetooth w wersji czwartej obejmuje standard Bluetooth Low Energy (w skrócie BLE), o który często opierane są urządzenia z przeznaczeniem do pracy przez dłuższą ilość czasu na baterii (m. in. popularne *Beacon*¹⁵). Protokół umożliwia przesyłanie w sposób bezpołączeniowy ramek rozgłoszeniowych, w których umieścić można pięć bajtów danych (tabela 1.1). Odpowiednio dwie dwu bajtowe liczby mające na celu identyfikację urządzenia, oraz siłę sygnału która pozwolić ma na ocenę odległości urządzenia odbiorczego (np. telefonu komórkowego) od nadajnika. Znane są autorowi aplikacje w

Major-1	Major-0	Minor-1	Minor-0	RSSI
---------	---------	---------	---------	------

Tabela 1.1. Struktura danych w ramce rozgłoszeniowej iBeacon

których format ramki iBeacon jest wykorzystywany niezgodnie ze standardem, a zamiast standardowo

¹⁵Miniaturowe urządzenia nadawcze BLE mające na celu umożliwienie orientacyjnej geolokalizacji użytkownika i dostarczania informacji kontekstowych.

określonych danych umieszczanych są w niej dane użytkownika (np. pomiar sensora). W ramach Bluetooth Low Energy możliwe jest również otworzenie klasycznego połączenia szeregowego, co pozwolić może na zebranie danych historycznych z sensora (np. pomiary z ostatniej godziny).

Technologia ta nie została jednak wzięta pod uwagę przy projektowaniu urządzenia, podobnie jak w przypadku WiFi przez wzgląd na znaczny stopień zajętości pasma 2.4GHz w przestrzeniach biurowych oraz mieszkalnych, powodujący bardzo dużą zawodność i ograniczony zasięg. Autor spodziewał się, że w celu uzyskania odpowiedniej niezawodności dostarczania danych sensorycznych, konieczne byłoby stosowanie wielokrotnych powtórzeń, co zaś znacznie zwiększyło by zużycie energii.

1.4.3. Moduły radiowe pracujące w paśmie nielicencjonowanym

Kolejną możliwością komunikacji radiowej są moduły pracujące w ogólnodostępnym paśmie ISM (*ang. Industrial, Scientific, Medical* – "przemysłowe, naukowe, medyczne), pozwalające na przesyłanie surowych ramek danych i optymalizację protokołu we własnym zakresie. Autor skupił się na modułach z serii RFM, producenta HopeRF (wykorzystującego wzory układów Semtech) przez wzgląd na atrakcyjne ceny, łatwą dostępność u polskich dystrybutorów i wcześniejsze doświadczenia autora z tymi modułami. Należy jednak zwrócić uwagę iż istnieje wiele alternatywnych modułów spełniających podobną rolę.

RFM75 - 2.4GHz

Są to moduły pracujące w paśmie 2.4GHz korzystając z protokołu ESB (*ang. Enhanced Shock Burst*) opracowanego przez Nordic Semiconductor. Są one kompatybilne z popularnymi modułami *nrf24l01+*, w których przypadku spotyka się wiele klonów, spotkać można negatywne komentarze dotyczące stabilności ich funkcjonowania. Dodatkowo nie są one dostępne w wersji do montażu powierzchniowego, co w przypadku RFM75 nie jest problemem. Należy jednak zwrócić uwagę, że moduły *nrf24l01+* tolerują napięcie 5V na magistrali, a w przypadku RFM75 nie należy przekraczać poziomów logicznych 3.3V.

Wykorzystane moduły mają niewielkie rozmiary (13 x 17 mm), wymagają doprowadzenia zasilania (1.9 - 3.6 V) oraz magistrali SPI, wraz z dodatkowym pinem kontrolnym (CSN). Umożliwiają przesyłanie pakietów w rozmiarze od jednego do trzydziestu dwóch bajtów.

Jest to układ często spotykany w peryferiach osobistych takich jak klawiatury czy myszy bezprzewodowe, podobnie jednak jak w przypadku WiFi oraz Bluetooth - pasmo 2.4GHz nie jest optymalne do przesyłania danych na większe odległości.

RFM69 - 433/868/915MHz

Jest to seria modułów komunikacji bezprzewodowej w paśmie poniżej 1GHz, występują one w trzech wersjach, których stosowanie możliwe jest zależnie od regionu. W Europie bez licencji można posługiwać się modułami pracującymi przy na częstotliwościach 433Mhz oraz 868MHz. Autor zdecydował się wykorzystać moduł pracujący w paśmie 868MHz, przez wzgląd na fakt, że częstotliwość 433MHz

wydaje się być powszechniej wykorzystywana w zabawkach, pilotach bram garażowych, szlabanów, stacjach pogodowych.

Moduły te nie posiadają zintegrowanej anteny, wymagają podłączenia zewnętrznej anteny odpowiedniej dla danego pasma. W przypadku sensorów wykorzystane zostaną miniaturowe anteny ceramiczne, a w przypadku bramy dostępowej - zewnętrzna antena o większym wzmocnieniu, dedykowana dla tego pasma.

Moduły te posiadają wbudowane mechanizmy pozwalające na obliczanie i sprawdzanie sum kontrolnych oraz szyfrowanie AES. Komunikują się z mikrokontrolerem za pomocą SPI.

Zużycie energii waha się od 16 do 130 mA podczas transmisji oraz poniżej 1 uA w uśpieniu. Dostępny jest również tryb spoczynku w którym zużywane jest średnio 1.5 uA, jednak w tym stanie co pewien czas odbiornik jest wybudzany celem odbioru danych, jeżeli coś do niego jest transmitowane.

Układy te występują w kilku wersjach, różniących się dostępną mocą nadawania. Autor zdecydował się na wykorzystanie mocniejszej wersji *RFM69HCW*, ponieważ możliwa jest kontrola mocy nadawczej przez oprogramowanie. Na etapie pomiarów i optymalizacji pracy sensora autor rozważył możliwość automatycznego regulowania mocy nadawczej tak by uzyskać minimalny poziom zużycia energii gwarantujący dostarczenie pakietu do bramy dostępowej.

1.4.4. Komunikacja w oparciu o sieci telekomunikacyjne - NarrowBand-IoT

Alternatywą dla sieci o niskim zasięgu (WiFi, BLE) czy też rozwiązań wymagających budowę własnej infrastruktury (komunikacja w paśmie ISM) jest skorzystanie z dostępnych sieci operatorów telekomunikacyjnych. W projekcie wykluczono wykorzystanie klasycznych modułów komunikacyjnych GSM, przez wzgląd na bardzo dużą konsumpcję energii (typowo nawet do 2A przez kilka sekund podczas transmisji).

W odpowiedzi na potrzeby rynku IoT powstał standard *NarrowBand-IoT*, który jest bardzo efektywny pod względem zużycia energii[5]. W przypadku tej technologii mówi się o wieloletnim czasie działania na baterii. Uzyskane to zostało dzięki uproszczonej modulacji, PSM (*ang. Power Saving Mode*) - automatycznym trybie oszczędzania energii oraz eDRX (*ang. Extended Idle-Mode Discontinuous Reception*) - wydłużeniu czasu bezczynności terminala (około czterokrotnie w stosunku do LTE).

Technologia ta dedykowana jest przesyłaniu bardzo małych ilości danych, w dużych odstępach czasowych - pojedynczy rozmiar datagramu, który wspierają terminale NB-IoT to 512 bajtów. Wspierany jest protokół UDP, często stosuje się CoAP (*ang. Constrained Application Protocol*) jako protokół warstwy wyższej. Choć technicznie możliwe nie zalecane są protokoły oparte o TCP, przez wzgląd na znaczny narzut ilości przesyłanych danych.

Technologia ta pracuje w ramach standardowych komórek LTE, wymagana jest jedynie odpowiednia konfiguracja oprogramowania. Możliwe jest zarówno wydzielenie fragmentu pasma w ramach częstotliwości zarezerwowanych dla komórki LTE, jak i również na skraju pasma (tak zwana komórka *Guard-Band*).

Do pracy z NB-IoT konieczne jest wykorzystanie dedykowanego terminala wykorzystującego standard z odpowiednimi poprawkami. Urządzenia w tym standardzie dostępne są u wielu producentów (m.in. Quectel, u-blox, Telit), zwrócono jednak uwagę, że często różnice między odpowiadającymi sobie modelami terminali są marginalne - opierają się one o ten sam chipset radiowy.

Autor zdecydował się na wykorzystanie terminala u-blox Sara N211, przez wzgląd na dostępność czytelnej i kompletnej dokumentacji, która pozwoliła na zaimplementowanie terminala we własnym urządzeniu. Wstępne testy przeprowadzone zostały za pomocą zestawu ewaluacyjnego producenta, jednak z jego wykorzystaniem nie możliwe było przeprowadzenie pomiarów zużycia energii, gdyż posiada on dużą ilość nadmiarowych komponentów oraz diod LED sygnalizujących stan urządzenia.

Komunikacja z terminalem odbywa się za pomocą tzw komend AT poprzez UART, charakterystycznych dla modemów GSM. Terminal ten wspiera poziom logiczny 3.3V, dzięki czemu nie było konieczne wykorzystanie konwertera poziomów logicznych.

Finalnie ten rodzaj komunikacji został odrzucony, gdyż pomimo znacznej optymalizacji zużycia energii w porównaniu do klasycznej sieci komórkowej, zalecane są do niego dosyć duże gabarytowo baterie, o dużej wydajności prądowej. Jest to technologia odpowiednia w przypadku gdy konieczne jest uruchomienie pojedynczego urządzenia w danej lokalizacji (np. w telemetrii dla energetyki czy gazownictwa), jednak w przypadku budowania sieci sensorowej rozsądniejsze wydają się bardziej energooszczędne moduły w paśmie ISM w zestawie z pojedynczą bramą dostępową.

1.5. Sensory temperatury, wilgotności, ciśnienia

Najpopularniejsze układy wykorzystywane w amatorskich rozwiązaniach IoT do pomiaru temperatury i innych parametrów atmosferycznych to:

- **DS18B20** - scalony czujnik temperatury z magistralą one wire,
- **DHT11** oraz **DHT22** - zintegrowane czujniki temperatury i wilgotności różniące się dokładnością.

Wartości deklarowanego poboru prądu podczas pomiaru wydają się pozornie niskie (odpowiednio 1.5mA dla DS18B20 oraz DHT11, 2.5mA dla DHT22), jednak w przypadku projektowanego rozwiązania konieczne było wyszukanie sensora o zdecydowanie niższym poborze prądu. Dodatkowo dla układów tych typowe napięcie zasilania to 5V - co uniemożliwiłoby optymalne wykorzystanie baterii.

Autor zdecydował się na wykorzystanie rzadziej spotykanego układu **BME280** - scalonego sensora temperatury, wilgotności, ciśnienia atmosferycznego. Wybór umotywowany został przez wzgląd na wysoką dokładność oraz deklarowane bardzo niskie zużycie energii (3.6uA podczas pomiaru temperatury, wilgotności i ciśnienia z częstotliwością odświeżania 1Hz, oraz 0.1uA w uśpieniu)[6].

By porównać praktyczną konsumpcję energii przez opisane sensory konieczne byłoby zmierzenie czasu trwania pomiarów, jednak autor zwrócił na kolosalną różnicę między tymi wartościami - niemal trzech rzędów wielkości między BME280 a DHT22 oraz gorszą dokładność pozostałych sensorów.

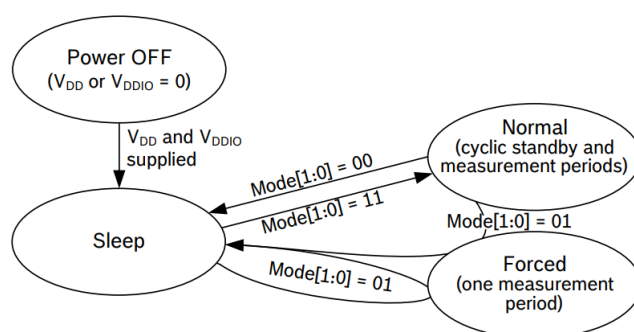
Dodatkowo zakres dopuszczalnych napięć zasilania sensora BME280 to 1.71V do 3.6V - dzięki czemu nie będzie konieczna dodatkowa regulacja napięcia. Porównanie popularnych sensorów oraz wybranego przez autora przedstawiono w tabeli 1.2.

	pobór prądu	dokładność temperatury	dokładność wilgotności	zakres napięć zasilania
BME280	3.6 uA	0.5°C	1%	1.71 - 3.6V
DS18B20	1500 uA	0.5°C	brak	3.0 - 5.5 V
DHT11	300 uA	2°C	5%	3.5 - 5.5 V
DHT22	2500 uA	0.5°C	2%	3.3 - 6V

Tabela 1.2. Porównanie parametrów pracy popularnych sensorów

Zarządzanie zasilaniem sensora

Na rysunku 1.2 opisane są tryby pracy sensora BME280. Po podłączeniu zasilania automatycznie przechodzi on w stan uśpienia, posiada dwa tryby pracy: normalny oraz wymuszony. W trybie normalnym pomiary są wykonywane cyklicznie, w wymuszonym wykonywany jest tylko jeden pomiar a następnie układ przechodzi z powrotem w stan uśpienia.



Rys. 1.2. Tryby pracy układu BME280, źródło: nota katalogowa

1.6. Zasilanie bezprzewodowych sensorów

W tabeli 1.3 podsumowano dopuszczalne napięcia pracy poszczególnych elementów projektowanego rozwiązania. Należy zwrócić uwagę że w przypadku mikrokontrolerów AVR wartość niższa (1.8V) podana jest dla częstotliwości zegara poniżej 4MHz, która może być niewystarczająca do poprawnej pracy z różnego rodzaju peryferiami (np. UART).

	minimalne	typowe	maksymalne
ATmega328	1.8 / 2.7V		5.5V
ATtiny84(V)	1.8 / 2.7V		5.5V
ESP32	2.3V	3.3V	3.6
ESP8266	2.5V	3.3V	3.6V
RFM75	1.9V	3.3V	3.6
RFM69	1.8V	3.3V	3.6
BME280	1.71V	3.3V	3.6V

Tabela 1.3. Napięcia pracy poszczególnych elementów rozwiązania

Po przeanalizowaniu potrzeb poszczególnych elementów systemu założono trzy możliwe konfiguracje zasilania:

- **zasilanie bezpośrednio** - zarówno w przypadku baterii litowych jak i litowo żelazowo fosforanowych ich napięcia pracy są odpowiednie do bezpośredniego zasilania rozwiązania,
- **stabilizacja napięcia do 3.3V** - akceptowalne dla wszystkich konfiguracji,
- **stabilizacja napięcia do 1.8V** - w przypadku mikrokontrolerów AVR z modułem RFM69 należy rozważyć napięcie z dolnego zakresu dopuszczalnych, dzięki czemu przy wykorzystaniu odpowiedniej przetwornicy można wykorzystać niemal całą energię zawartą w klasycznych bateriach alkalicznych.

Baterie alkaliczne

Jest to rodzaj baterii najpowszechniej wykorzystywany w urządzeniach wykorzystywanych okazjonalnie, o małym poborze prądu (piloty, zegarki). Ich napięcie nominalne to 1.5V, można je rozładować aż do 0.8V. Przy rozładowaniu bardzo małymi prądami typowa bateria w rozmiarze LR6 potrafi przechować ponad 2500mAh energii[7].

Baterie litowe

Baterie litowe występują w różnych formach:

- **baterie pastylkowe** - o pojemnościach od 30mAh do 600mAh zależnie od rozmiaru,
- **baterie cylindryczne** - np. CR123A (popularna bateria foto) pojemności ok 1500mAh, CR-V3 kształt baterii LR6 pojemności ok 3000mAh.

Ich napięcie nominalne to 3V lub 3.6V, rozładować można je do 2V.

Ogniwa litowo żelazowo fosforowe

Są to ogniwa o szczególnie atrakcyjnym dla rozwiązań IoT zakresie napięć pracy. Nominalnie przeznaczone są do pracy przy napięciu 3.3V w pełni naładowane utrzymują napięcie 3.6 V, a rozładować można je nawet do 2 V [8]. W projekcie rozważone zostały zostały ogniwa APR18650 producenta A123Systems o pojemności 1100 mAh, jako znane i dostępne na rynku polskim ogniwa LiFePo4 w klasycznej formie cylindrycznej 18650. Mając jednak na względzie ich pojemność względem rozmiaru obudowy, nie jest to odpowiedni wybór do projektu.

Ogniwa litowo jonowe oraz litowo polimerowe z przetwornicą

Ogniwa litowo jonowe oraz litowo polimerowe są łatwo dostępne i dosyć tanie. Ich znacznym plusem jest mnogość rozmiarów w których są dostępne, od klasycznych cylindrycznych ogniw 18650 wykorzystywanych do zasilania komputerów przenośnych - po ogromną ilość rozmiarów baterii płaskich o pojemnościach od kilkudziesięciu po tysiące miliamperogodzin.

Mając jednak na uwadze zakres napięć z którymi pracują tego rodzaju akumulatory konieczne by było jednak stabilizowanie napięcia do niższego - odpowiedniego dla komponentów sensora, co wiązałoby się z pewnymi stratami energii.

1.6.1. Wybrane rodzaje baterii

Po porównaniu dostępnych baterii (porównanie parametrów w tabeli 1.4), ich pojemności oraz napięć pracy założono iż wykluczone zostaną akumulatory Litowo Jonowe, przez wzgląd na konieczność stabilizacji napięcia oraz zjawisko samoczynnego rozładowywania się.

Najbardziej odpowiednim rozwiązaniem dla projektowanego rozwiązania są baterie litowe, osiągające bardzo duże pojemności przy rozładowywaniu niskimi prądami. Jednocześnie mając na uwadze utrudnioną dostępność tego rodzaju baterii dla konsumenta (użytkownika projektowanego rozwiązania), rozważone zostanie również wykorzystanie tradycyjnych baterii alkalicznych, jako zamiennika, który jest powszechnie dostępny.

	minimalne	nominalne	maksymalne
Li-Fe-Po4	2.0 V	3.3 V	3.6 V
Li-Ion	2.7 V	3.7 V	4.2 V
Litowe pastylkowe	2.0 V	3.0 V	3.2 V
Litowe cylindryczne	2.0 V	3.6 V	3.62 V
Alkaliczne	0.8 V	1.5V	1.5V

Tabela 1.4. Napięcia charakterystyczne dla poszczególnych rodzajów baterii

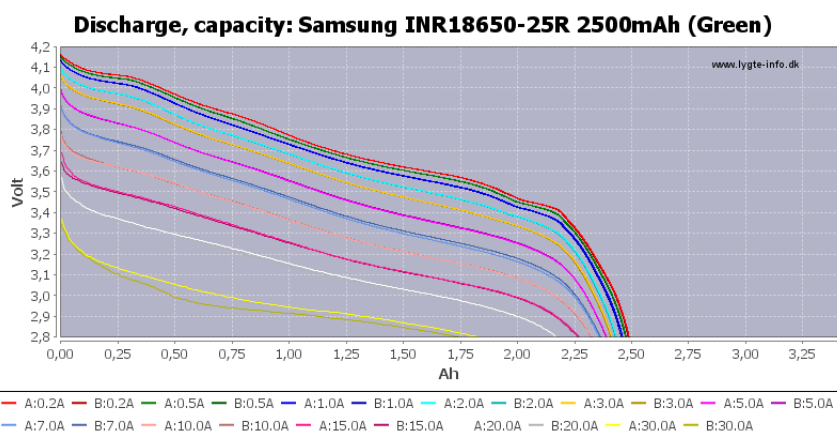
1.7. Ocena stanu naładowania baterii / akumulatorów

W przypadku sensora zasilanego bateryjnie, który może być zainstalowany w trudno dostępnym miejscu istotną funkcjonalnością wydaje się być przesyłanie wraz z pomiarami informacji o stanie naładowania baterii. Wszystkie testowane układy mikrokontrolerów posiadają zintegrowane przetworniki analogowo cyfrowe, więc pomiar napięcia akumulatora / baterii jest technicznie wykonalny.

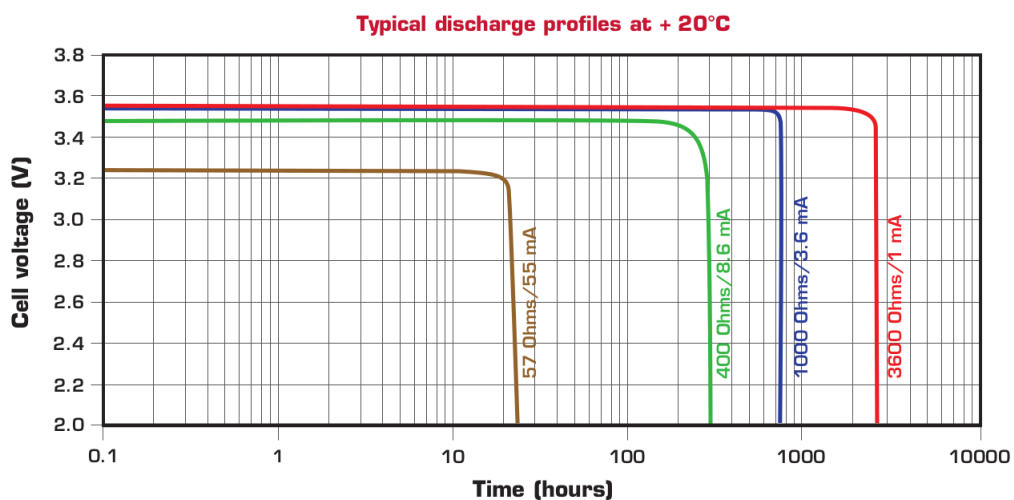
Jednak przez wzgląd na charakterystyki rozładowania poszczególnych rodzajów baterii, użyteczną wiedzę o stanie akumulatora taki pomiar dostarczyć może jedynie w przypadku akumulatorów litowo-jonowych, których charakterystyka rozładowania (Rysunek 1.3) w pewnym stopniu pozwala na ocenę stopnia naładowania poprzez pomiar napięcia.

W przypadku baterii litowych oraz akumulatorów litowo-żelazowo-fosforowych, których charakterystyki są niemalże płaskie, w szczególności dla niskich prądów rozładowania (odpowiednio rysunki 1.4 oraz 1.5) - jedyną możliwością wiarygodnej oceny pojemności akumulatora jest pomiar prądu pobieranego przez układ oraz zliczanie zużytej energii. Istnieją scalone układy służące jako kulombometry stosowane w urządzeniach przenośnych do oceny energii pozostałej w akumulatorze. Jednym z nich jest układ MAX17055 [9], który posiada interfejs I2C i dostarcza stan naładowania akumulatora zarówno w procentach jak i miliamperogodzinach. Należy mieć jednak na względzie pobór prądu przez taki układ, który choć jest marginalny (7uA według noty katalogowej producenta), w przypadku projektowanego rozwiązania może być kilkukrotnie wyższy niż pobór energii przez cały układ w stanie uśpienia.

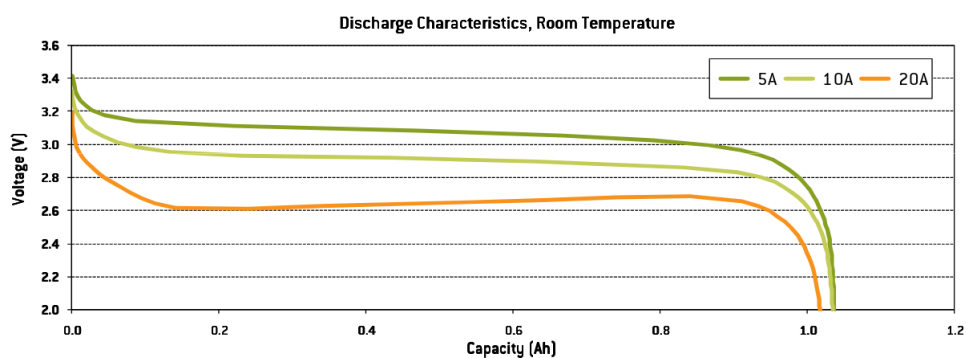
Wobec tego zdecydowano się na zastosowanie jedynie uproszczonej informacji o stanie naładowania akumulatora - bazując na charakterystykach każdego z rodzajów baterii wyznaczono napięcie progowe dla którego należy założyć konieczność wymiany baterii.



Rys. 1.3. Charakterystyka rozładowania akumulatorów litowo-jonowych Samsung INR18650, źródło: niezależny portal testujący akumulatory [10]



Rys. 1.4. Charakterystyka rozładowania baterii litowych Saft LS14500, źródło: nota katalogowa [11]



Rys. 1.5. Charakterystyka rozładowania baterii litowo żelazowo fosforowych APR18650, źródło: nota katalogowa [8]

2. Prototypy platformy pomiarowej

W tym rozdziale opisane zostaną wykonane prototypy układu sensorowego w różnych konfiguracjach. W pierwszej części opisane zostaną wybrane konfiguracje sprzętowe do testowania. Następnie przedstawione zostaną schematy oraz szczegóły wykonania prototypów, wraz ze wstępną wersją oprogramowania.

Opisane w tym rozdziale prototypy podlegać będą dalszym modyfikacjom i optymalizacji równolegle z wykonywanymi pomiarami poboru prądu - co zostanie przedstawione w kolejnym rozdziale.

2.1. Architektura rozwiązania, wybór konfiguracji sprzętowych

Założono iż prototypy powinny być łatwo rekonfigurowalne, by zapewnić elastyczność podczas optymalizacji warunków pracy poszczególnych kombinacji mikrokontrolerów, układów zasilania i modułów radiowych. Przykładowo, wszystkie wykonane płyty PCB mają możliwość montażu przetwornicy napięcia, jak i również zasilania bezpośredniego. Podobnie - dla układów ATmega328p przygotowano miejsce na rezonator kwarcowy, jednak prawdopodobnie ze względu na oszczędność energii wykorzystany zostanie wewnętrzny oscylator RC o niższym taktowaniu.

Dodatkowo, wykonane płyty prototypowe posiadają pełne wyprowadzenia do programowania, czy dodatkowe punkty testowe, co nie jest optymalne ze względu na rozmiar samego urządzenia, jednak na etapie prototypowania pozwala to na szybsze przeprogramowywanie mikrokontrolera, czy też podłączenie dodatkowych komponentów.

Każda z konfiguracji testowych składać będzie się z czterech elementów:

- **Mikrokontroler** - układ z mikrokontrolerem umożliwiający głębokie usypianie,
- **Moduł komunikacyjny** - moduł radiowy z anteną lub gniazdem antenowym,
- **Sensor** - sensor BME280 podłączony za pomocą magistrali I2C,
- **Zasilanie** - układ zasilania (przetwornica z kondensatorami filtrującymi) lub adapter dla baterii podłączanej bezpośrednio.

Tam gdzie to możliwe przygotowane zostały PCB posiadające zintegrowany komplet peryferiów

2.1.1. Wybrane konfiguracje sprzętowe prototypów

Po przeanalizowaniu założeń, dostępnych komponentów wybrano następujące konfiguracje sprzętowe:

- **ESP8266 z komunikacją WiFi**, jako przykład rozwiązania wykorzystującego już dostępną infrastrukturę w postaci sieci bezprzewodowych WiFi, oparty o łatwo dostępną i tanią architekturę sprzętową,
- **ESP32 z komunikacją WiFi**, by przetestować usprawnienia wprowadzone przez układ ESP32 w postaci skrócenia czasu połączenia z siecią WiFi dzięki wykorzystaniu koprocatora podtrzymującego asocjację z punktem dostępowym w stanach uśpienia,
- **ATMega328p oraz RFM75**, dla zweryfikowania efektywności modułów komunikacyjnych działających w paśmie 2.4GHz z protokołem ESB,
- **ATMega328p oraz RFM69**, jako najbardziej obiecująca konfiguracja wykorzystująca moduły radiowe pracujące w paśmie 868MHz o bardzo niskim zużyciu energii,
- **ATMega328p oraz terminal Nb-IoT u-blox Sara N211**, by przetestować możliwości budowy rozwiązania sensorycznego w oparciu o komercyjną infrastrukturę operatora komórkowego.

Wszystkie konfiguracje sprzętowe testowane będą z sensorem temperatury, wilgotności oraz ciśnienia atmosferycznego BME280, jednoznacznie wybranym jako najlepszy z punktu widzenia zużycia energii oraz dokładności pomiarów.

Dodatkowo autor zdecydował się też na przetestowanie konfiguracji ESP8266 z sensorem DHT11, jako najbardziej powszechnie przedstawianą we wszelkiego rodzaju instrukcjach, warsztatach, filmach instruktażowych - jednak zupełnie nie optymalną z punktu widzenia zużycia energii.

W przypadku każdej z konfiguracji po wykonaniu wstępnych pomiarów rozważone zostaną możliwości optymalizacji zużycia energii, poprzez odpowiednią konfigurację stanów uśpienia oraz oszczędzania energii, podmianę komponentów pasywnych, modyfikację układu zasilania.

2.2. Wykonanie prototypów

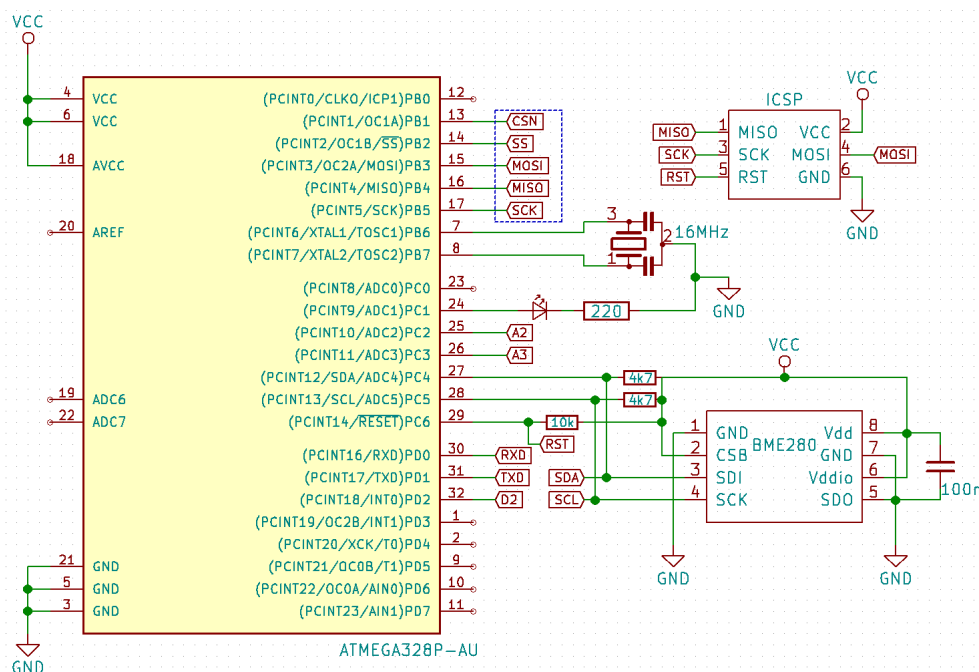
2.2.1. Prototypy oparte o mikrokontroler AVR ATmega

W przypadku prototypów opartych o mikrokontroler ATmega przygotowane zostały PCB zawierające część wspólną układu ATmega328p-au oraz sensora BME280 (rysunek 2.1). Jest to podstawowa aplikacja mikrokontrolera wraz z opcjonalnym oscylatorem 16MHz, złączem do programowania ICSP (*ang. In Circuit Serial Programming*) oraz wyprowadzoną magistralą UART, by ułatwić testowanie układu.

Do układu ATmega podłączony został sensor BME280 za pomocą magistrali I2C, wraz z rezystorami podciągającymi, zalecanym kondensatorem filtrującym oraz odpowiednią konfiguracją (stan wysoki na pinie *CSB* by sensor korzystał z magistrali I2C, stan niski na pinie *SDO* - by adres układu pozostał niezmieniony).

Do podłączenia modułów radiowych wykorzystana została magistrala SPI, wraz z dodatkowymi pinami cyfrowymi służącymi sygnalizacji stanu pracy modułów. W przypadku RFM75 jest to kombinacja *SS* oraz *CSN*, a w przypadku układu RFM69 *SS* oraz port przerwania *D2*.

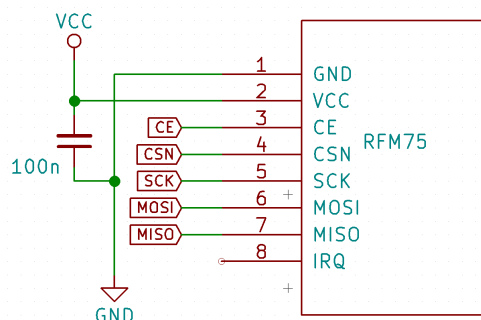
Dodatkowo wyprowadzone zostały punkty kontrolne - porty magistrali UART (*TXD*, *RXD*), dwa piny cyfrowe (*A2*, *A3*), oraz dołączono diodę LED na jednym z pinów cyfrowych (*A1*). Porty te nie zostaną wykorzystane w finalnej konfiguracji, a magistrala UART będzie wyłączona by zminimalizować zużycie energii - są to jednak wyprowadzenia, które mogą być pomocne podczas prototypowania i rozwiązywania ewentualnych problemów.



Rys. 2.1. Schemat układu ATmega328p z sensorem BME280

Moduł radiowy RFM75

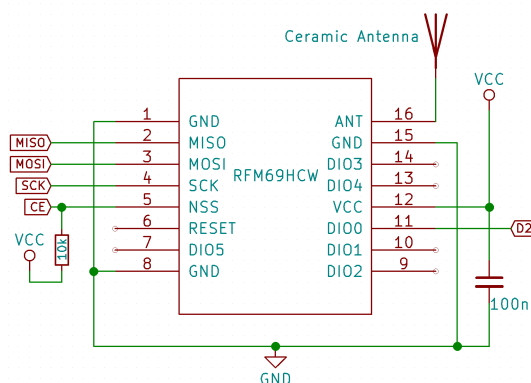
Moduł radiowy RFM75 został bezpośrednio podłączony do magistrali SPI, z wykorzystaniem dodatkowej linii sygnalizacyjnej *CSN*. Wyjście przerwań nie zostało wykorzystane, gdyż nie jest ono wymagane do funkcjonowania modułu w trybie nadawania, a nie przewiduje się możliwości wybudzania mikrokontrolera sygnałem radiowym. Tuż przy module umieszczony został dodatkowy kondensator filtrujący, zgodnie z zaleceniem producenta.



Rys. 2.2. Schemat aplikacyjny dla modułu radiowego RFM69HCW

Moduł radiowy RFM69HC

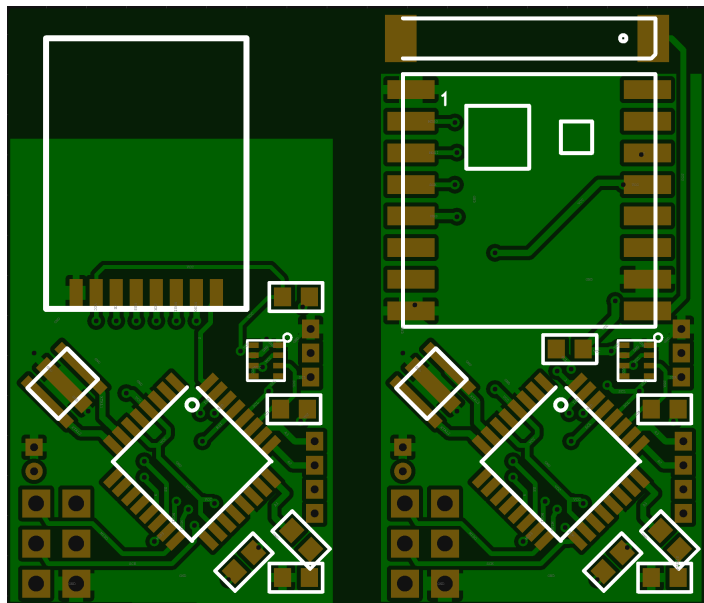
Moduł radiowy RFM69 również został podłączony do magistrali SPI, wraz z dodatkowym pinem sygnalizującym *DIO0* obsługiwanym po stronie mikrokontrolera przez przerwania - jest on wymagany do funkcjonowania modułu, nawet gdy używa się go jako nadajnika. Moduł ten nie posiada zintegrowanej anteny, zdecydowano się więc na umieszczenie na PCB miniaturowej anteny ceramicznej dedykowanej pasmu 868MHz. Poza standardowym kondensatorem filtrującym napięcie konieczne było również umieszczenie dodatkowego rezystora podciągającego linię *CE*, tak by podczas programowania mikrokontrolera moduł radiowy pozostawał w uśpieniu i nie zaburzał komunikacji podczas programowania, która współdzieli linie *MISO*, *MOSI* oraz *CLK*.



Rys. 2.3. Schemat aplikacyjny dla modułu radiowego RFM69HCW

Wykonane płyty PCB

Aby przetestować prototypy oparte o AVR ATmega autor zaprojektował dwie płyty PCB których większość opiera się o opisany schemat 2.1. Płyty różnią się jedynie aplikacją modułu radiowego.

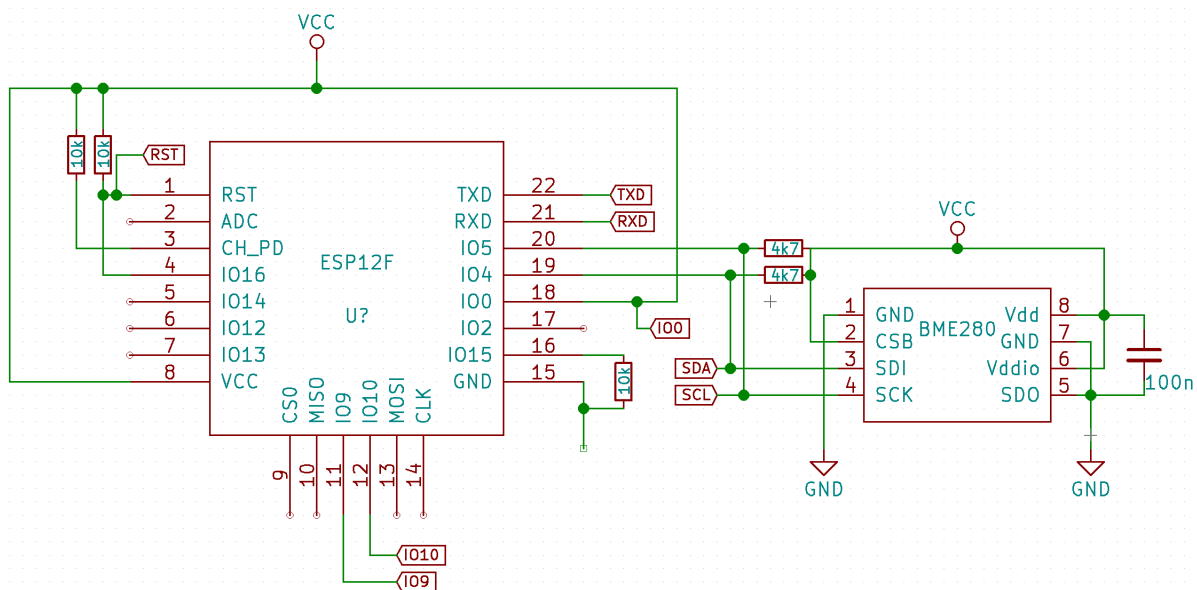


Rys. 2.4. Widok płyt PCB z ATmega328p-au, sensorem BME280 oraz odpowiednio RFM75 i RFM69HCW

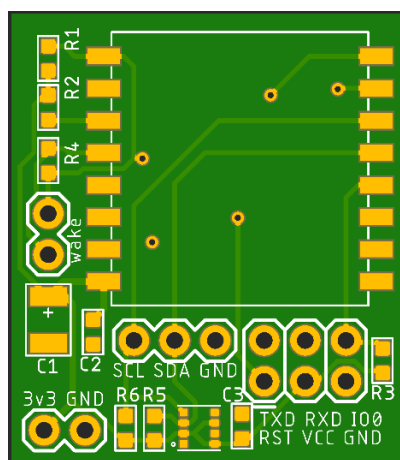
2.2.2. Prototyp z komunikacją WiFi - ESP8266

Układ oparty o ESP8266 jest to podstawowa aplikacja tego modułu, z odpowiednimi rezystorami podciągającymi oraz sensorem BME280 podłączonym za pomocą magistrali I2C. Programowanie odbywa się poprzez magistralę UART, dodatkowo wyprowadzono pole *RST* do resetowania modułu oraz *IO0*, którego stan logiczny decyduje o przejściu w stan programowania lub normalnej pracy.

Dodatkowo, by umożliwić wykorzystanie stanu głębokiego uśpienia konieczne było bezpośrednie połączenie *IO16* z polem *RST*, za pomocą którego następuje wybudzenie modułu po upływnięciu ustawionego licznika.



Rys. 2.5. Schemat aplikacyjny prototypu sensora opartego o moduł ESP8266 w obudowie ESP12F oraz sensor BME280



Rys. 2.6. Widok płyty PCB wykonanej dla ESP8266 oraz sensora BME280

2.2.3. Sensor z komunikacją NarrowBand-IoT

Prototyp wykorzystujący komunikację za pomocą protokołu NarrowBand-IoT, poprzez komercyjną sieć komórkową - wykonano w oparciu o mikrokontroler ATmega328 oraz modem uBlox Sara N211. Jest to dwuzakresowy terminal Nb-IoT mogący pracować w pasmach LTE dwudziestym oraz ósmym. Komunikacja między mikrokontrolerem oraz terminalem następuje poprzez magistralę UART.

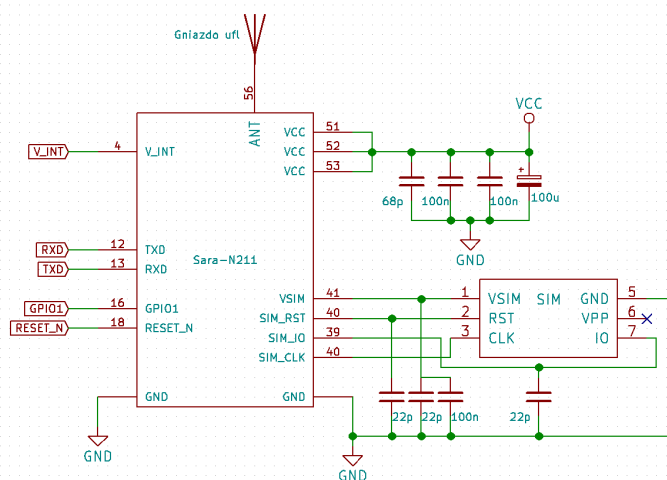
Układ z mikrokontrolerem zaprojektowany został w sposób wysoce zbliżony do wspomnianych wcześniej dla modułów radiowych w paśmie ISM (zgodnie ze schematem 2.7). Dodatkowo ponieważ rozmiar płyty PCB podyktowany został rozmiarem modemu oraz gniazda kart SIM zdecydowano się na umieszczenie dodatkowych wyprowadzeń, umożliwiających dalsze prototypowanie z wykonaną płytą.

Modem Sara N211 posiada zintegrowane większość komponentów wymaganych do działania. Zgodnie z zaleceniem producenta podłączone zostały odpowiednie kondensatory filtrujące tuż przy gnieździe zasilającym.

Karta SIM została podłączona do dedykowanego interfejsu. Zmodyfikowano natomiast wartości kondensatorów filtrujących na magistrali karty SIM. Wartość wskazana przez producenta w instrukcji integracyjnej [12] - 47pF okazała się zbyt duża. Podczas uruchamiania modemu okazało się iż modem nie wykrywa karty SIM. Po zaobserwowaniu za pomocą oscyloskopu iż przebiegi na liniach *CLK* oraz *IO* odbiegają od pożądanych - prostokątnych autor zdecydował się na wymianę kondensatorów na 22pF, co rozwiązało problem.

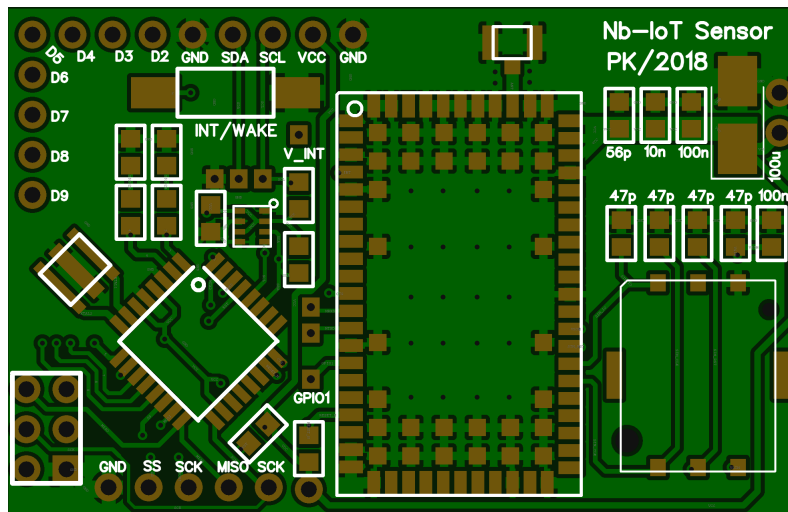
Zrezygnowano z zabezpieczeń ESD przez wzgląd na wysoką cenę komponentów i dodatkowe skomplikowanie układu. Zgodnie z zaleceniem producenta wyprowadzono dodatkowe punkty kontrolne do ewentualnej analizy stanu modemu - *V_INT* oraz *GPIO1*.

Schemat aplikacji modemu przedstawiony został na rysunku 2.7.

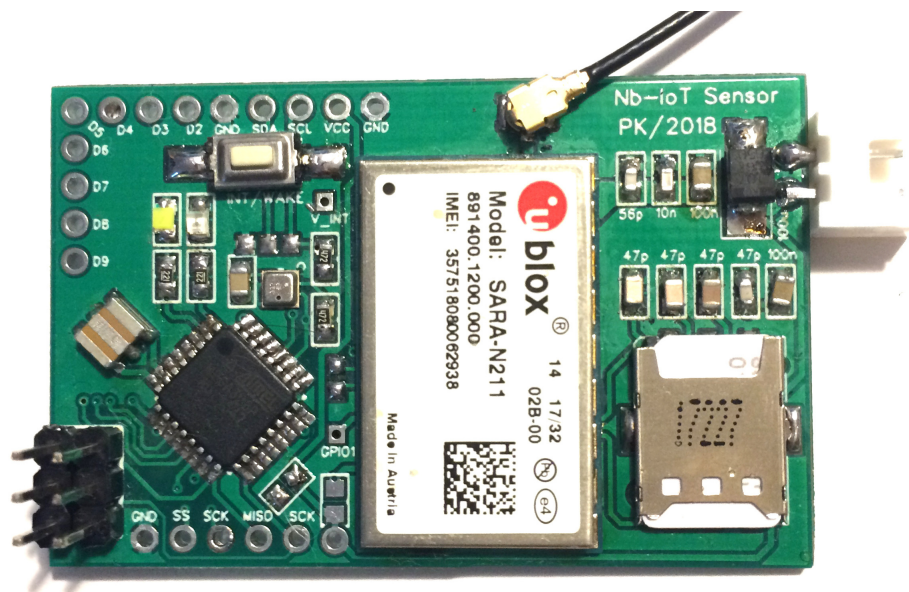


Rys. 2.7. Schemat aplikacyjny dla modemu ublox Sara N211

Płytkę PCB (rysunek 2.8) dla tego prototypu zaprojektowano mając na uwadze wskazania przedstawione przez producenta modemu oraz planowany sposób montażu. Wszystkie komponenty zostały umieszczone na wierzchu płytki, dzięki czemu możliwy był montaż poprzez podgrzanie PCB od spodu z wykorzystaniem pasty lutowniczej. Dla ułatwienia montażu i ze względu na planowane wykorzystanie wykonanej płytki do prototypowania - zdecydowano się na warstwę opisową z wartościami komponentów oraz oznaczeniami wyprowadzonych portów. Wykonany prototyp przedstawiony jest na zdjęciu 2.9.



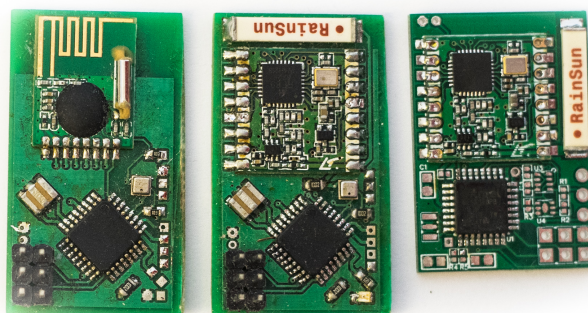
Rys. 2.8. Widok płytki PCB wykonanej dla ATmega328p oraz terminala u-blox Sara N211



Rys. 2.9. Prototyp wykonany w oparciu o ATmega328p oraz terminal Nb-IoT u-blox Sara N211

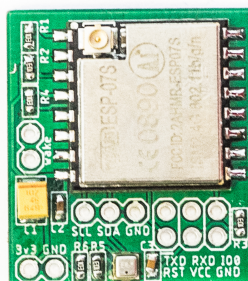
2.2.3.1. Montaż prototypów

Montaż prototypów wykonano kombinacją lutownicy na gorące powietrze oraz kolbowej stacji lutowniczej. Gorące powietrze posłużyło do montażu sensora BME280, którego styki umieszczone są pod spodem układu, reszta elementów przylutowana została ręcznie. Po wykluczeniu ewentualnych zwarców lub braków ciągłości na liniach komunikacyjnych oraz optycznej inspekcji płytek dokonano uruchomienia prototypów - podłączono programator za pomocą złącza ICSP oraz ustawiono odpowiednie bity konfiguracyjne, definiujące źródło i częstotliwość zegara. Układy z mikroprocesorem ATmega328p w procesie prototypowania wykonywano w kilku wersjach (rysunek 2.10), różniących się fizycznym umiejscowieniem komponentów, wyprowadzeniem interfejsu do wgrывania oprogramowania, czy też dodatkowymi polami do testowania układu.



Rys. 2.10. Kilka wersji prototypu czujnika z mikroprocesorem AVR ATmega

Prototyp z układem ESP8266 wykonany został w jednej wersji zawierającej odpowiednie wyprowadzenia do wgrania oprogramowania oraz ewentualnego dodania zewnętrznego czujnika na magistrali I2C (rysunek 2.11).

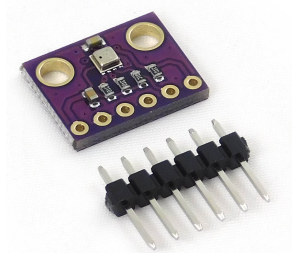


Rys. 2.11. Prototyp czujnika z układem ESP8266

2.3. Sensor pomiarowy

Na etapie testowania sensora, jeszcze przed zintegrowaniem go na wykonywanych płytach drukowanych wykorzystany został gotowy moduł (zdjęcie 2.12) z układem BME280 zawierający jedynie rezystory podciągające i kondensatory filtrujące.

Zasilanie nie jest stabilizowane, wyprowadzone są magistrale I2C oraz SPI.



Rys. 2.12. Moduł z układem BME280

Mając na uwadze różnice między zużyciem energii przez poszczególne magistrale[13] założono przetestowanie modułu korzystając z SPI na etapie optymalizacji finalnego rozwiązania.

2.4. Zasilanie układu

Podczas pomiarów zużycia energii przez prototypy sensorów zasilanie dostarczane będzie z urządzenia pomiarowego Power Monitor pozwalającego na regulowanie napięcia zasilania. Początkowo - wszystkie prototypy zasilane będą napięciem 3.3V, by łatwiej było porównać uzyskane wyniki.

2.5. Oprogramowanie prototypów

Oprogramowanie prototypów przygotowane zostało z wykorzystaniem zintegrowanego środowiska PlatformIO, które pozwala na programowanie wielu różnych platform sprzętowych, w oparciu o różne zestawy bibliotek i narzędzi deweloperskich - między innymi Arduino, mbed, Espressif IDF. Środowisko to opiera się o edytor tekstu Atom, oraz system skonteneryzowanych środowisk do budowania projektów - dzięki czemu nie ma potrzeby konfiguracji poszczególnych środowisk dla mikrokontrolerów w systemie, a jedynie zdefiniowanie odpowiednich ustawień w pliku konfiguracyjnym - brakujące komponenty i biblioteki zostaną zainstalowane przed kompilacją projektu.

Wszystkie wykorzystywane platformy są zgodne ze środowiskiem Arduino. Jest to otwarte i darmowe środowisko szeroko wykorzystywane przez hobbystów do szybkiego prototypowania i pisania oprogramowania na platformy wbudowane. Na platformę Arduino składa się również wiele gotowych układów i płyt deweloperskich pozwalająca na wygodne eksperymentowanie bez przygotowywania własnych PCB.

Zwrócono jednak uwagę na wspomniane w rozdziale pierwszym często występujące ograniczenia w środowisku oraz gotowych bibliotekach - pisane one są często w sposób nadmiarowy, tak by objąć jak najwięcej konfiguracji sprzętowych i sytuacji, nie koniecznie w sposób najbardziej efektywny.

Założono wykorzystanie gotowych rozwiązań celem szybkiego prototypowania i testowania, jednak na etapie optymalizacji konkretnego rozwiązania konieczna będzie weryfikacja, czy tego rodzaju ograniczenia nie zostały wprowadzone przez autorów bibliotek obsługujących wykorzystujące komponenty.

3. Metodologia badawcza i przeprowadzenie testów poboru energii

W tym rozdziale opisany zostanie sposób w jaki przeprowadzone zostało porównanie efektywności prototypów platformy pomiarowej. Określone zostaną parametry, które zostaną wzięte pod uwagę, sposób ich pomiaru oraz kryteria według których dokonane zostało porównanie.

Równoległe z pomiarami wykonywanymi dla każdej z platform, Wykonane zostaną próby optymalizacji oprogramowania oraz określenie ich wpływu na zużycie energii.

3.1. Czynniki decyzyjne w porównaniu prototypów

Celem autora było rzetelne oszacowanie czasu pracy danej platformy pomiarowej zasilanej z baterii o określonej pojemności. Celem takiej oceny konieczne było określenie zużycia energii podczas pracy sensora w każdym z możliwych stanów (głębokiego uśpienia, dokonywania pomiaru oraz transmisji danych), oraz czasu ich trwania.

Założeniem projektowanych testów jest pomiar poboru prądu zarówno podczas pracy, w uśpieniu oraz rzeczywistego średniego zużycia energii by oszacować czas pracy na baterii lub akumulatorze o znanej pojemności. Testom poddane zostaną opracowane platformy sprzętowe - sensor, mikrokontroler oraz medium komunikacyjne.

W pierwszej kolejności wzięte pod uwagę zostaną warunki pracy mikrokontrolera AVR ATmega328p oraz sensora BME280 - wspólnego dla większości proponowanych rozwiązań. Dodatkowo w przypadku platform opartych o WiFi rozważony zostanie aspekt asocjacji z punktem dostępowym oraz jego optymalizacja - będąca głównym powodem nadmiernej konsumpcji energii.

Rozważono przeprowadzenie dłuższych pomiarów cyklicznej transmisji danych, jednak przez wzgląd na niedokładność wykorzystywanego narzędzia pomiarowego (opisane poniżej) w zakresie niskich prądów (poniżej 500 mikroamperów) - wpływ tej niedokładności byłby zbyt duży by przeprowadzić rzetelne pomiary.

Kalkulacja czasu pracy poszczególnych konfiguracji wykonana zostanie w rozdziale czwartym, gdzie zestawione zostaną wyniki dokonanych pomiarów.

3.2. Środowisko pomiarowe

Do pomiarów na tym etapie pracy wykorzystano dedykowane urządzenie pomiarowe *Power Monitor* producenta Monsoon Solutions, model *FTA22D*. Jest to urządzenie pomiarowe pozwalające na zasilanie mierzonego układu dowolnym napięciem z zakresu 2.1V do 4.5V, mogące dostarczyć do 3A prądu. W urządzeniu zostały zastosowane dwa zakresy pomiarowe - pomiędzy 30mA a 4.5A o rozdzielczości 286uA, oraz dokładny do 40mA o rozdzielczości 2.86uA - jest to narzędzie pomiarowe dedykowane pracy z terminalami bezprzewodowymi (m.in. telefonami komórkowymi), które zwykle większość czasu trwają w uśpieniu by okresowo używać więcej energii podczas transmisji.

Urządzenie komunikuje się z komputerem za pomocą magistrali USB, a w połączeniu z dedykowaną aplikacją pozwala na rejestrację pomiarów celem późniejszej analizy. Dla wszystkich testów ustalono stałe napięcie pracy 3.30V, rejestrowano po kilkanaście cykli pomiarowo-transmisyjnych, by ocenić czy wynik jest powtarzalny. Następnie analizowano pojedynczy przebieg wysyłania danych równoległe starając się zoptymalizować warunki pracy.

Przez wzgląd na niedokładność pomiaru bardzo małych prądów urządzeniem *Power Monitor*, wykonano dodatkowe pomiary zużycia energii w trybie głębokiego uśpienia za pomocą multimetru cyfrowego *VoltCraft VC170-1*. Charakteryzuje się on bardzo dużą dokładnością oraz małym błędem (rozdzielczość 0.1uA w zakresie do 400uA, z błędem do 2.3%, gdzie dla *Power Monitor* jest to 2.86 uA rozdzielczości i dokładność 50uA)

Na dalszym etapie pracy (w rozdziale kolejnym - skupiającym się na optymalizacji wybranego rozwiązania) wykorzystano nowe narzędzie pomiarowe wcześniej niedostępne autorowi - *Otii Arc*. Jest to urządzenie o formie i funkcjonalności podobnej do urządzenia *Power Monitor*, jednak o większej dokładności (1% + 0.5uA w całym zakresie pracy urządzenia do 5A)¹

Urządzenie to posiada również szereg udogodnień jak dodatkowa możliwość pomiaru napięcia analogowego, stanów cyfrowych, czy też zestawienie wykresu poboru prądu z komunikatami wysyłanym przez urządzenie na UART. Co pozwala na bardziej dokładne zrozumienie co dokładnie dzieje się w testowanym urządzeniu, oraz który z procesów zużywa określoną ilość energii (np. dokonywanie pomiaru, transmisja bezprzewodowa).

¹<https://www.qoitech.com/products/techspec>

3.3. Optymalizacja działania prototypów

Wspomnianej optymalizacji parametrów pracy platform pomiarowych podlegać będą dwa konteksty:

- **wykonywanie pomiaru oraz transmisja danych** - czas w którym urządzenie jest aktywne powinien być jak najkrótszy, a energia pobierana przez moduł radiowy jak najniższa; jednocześnie konieczne jest zapewnienie wiarygodności pomiaru oraz jego dostarczenie do modułu odbiorczego,
- **stan głębokiego uśpienia** - stan nieaktywności urządzenia, w którym konieczne jest możliwe ograniczenie poboru prądu, poprzez wyłączenie nieużywanych peryferiów, modułu transmisyjnego, sensora.

3.4. Konfiguracja usypiania ATmega328p oraz sensora BME280

Przed rozpoczęciem pracy i pomiarów dla prototypów platform pomiarowych wykorzystujących mikrokontroler ATmega328p oraz sensor BME280 jako część wspólną - skupiono się na optymalizacji ich cyklu pracy oraz stanów głębokiego uśpienia.

Główną pętlę aplikacji w trybie domyślnym przedstawiono w tabeli 3.1.

```
1 void loop() {  
2     _sensorData.temperature = bme.readTemperature();  
3     _sensorData.pressure    = bme.readPressure();  
4     _sensorData.humidity    = bme.readHumidity();  
5     _sensorData.systemVCC   = readVcc();  
6  
7     delay(4000);  
8 }
```

Tabela 3.1. Pętla główna oprogramowania dla AVR w domyślnej konfiguracji

W pierwszej kolejności zdecydowano się na skonfigurowanie głębokiego usypiania mikrokontrolera (tabela 3.2).

Następnie przekonfigurowano tryb odczytów danych z sensora na wymuszony, pojedynczy (tabela ??).

```

1 void loop() {
2     _sensorData.temperature = bme.readTemperature();
3     _sensorData.pressure    = bme.readPressure();
4     _sensorData.humidity    = bme.readHumidity();
5     _sensorData.systemVCC   = readVcc();
6
7     LowPower.powerDown(SLEEP_4S, ADC_OFF, BOD_OFF);
8 }

```

Tabela 3.2. Pętla główna oprogramowania dla AVR z włączonym głębokim usypianiem mikrokontrolera

```

1 void setup() {
2     bme.begin();
3
4     bme.setSampling(Adafruit_BME280::MODE_FORCED,
5     Adafruit_BME280::SAMPLING_X1, // temperature
6     Adafruit_BME280::SAMPLING_X1, // pressure
7     Adafruit_BME280::SAMPLING_X1, // humidity
8     Adafruit_BME280::FILTER_OFF );
9 }

```

Tabela 3.3. Konfiguracja trybu pojedynczych pomiarów dla sensora BME280

Zwrócono uwagę iż wyłączenie filtrowania danych z sensora może sprzyjać zakłóceniom i niedokładnościom pomiaru, jednak założono iż jest to optymalna konfiguracja pozwalająca na rzetelną cenę różnic między różnymi modułami komunikacyjnymi.

Mikrokontroler skonfigurowano by przechodził do stanu głębokiego usypienia. Przez wzgląd na rozmiar wewnętrznych liczników, maksymalny czas na który mikrokontroler może zostać uspijony to osiem sekund. Wywołano więc funkcję *powerDown* w pętli ośmiokrotnie (tabela ??) by uzyskać 64 sekundy stanu usypienia.

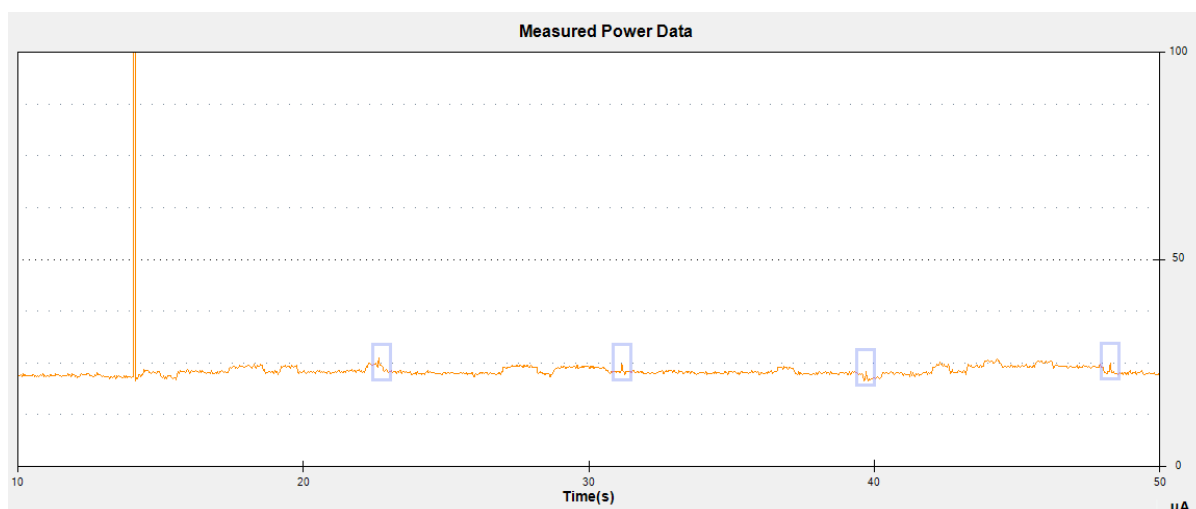
```

1     void loop() {
2         _sensorData.temperature = bme.readTemperature();
3         _sensorData.pressure    = bme.readPressure();
4         _sensorData.humidity    = bme.readHumidity();
5         _sensorData.systemVCC   = readVcc();
6
7         for(uint8_t i = 0; i < 8; i++)
8             LowPower.powerDown(SLEEP_8S, ADC_OFF, BOD_OFF);
9     }

```

Tabela 3.4. Pętla główna oprogramowania dla AVR z usypieniem na 64s

W efekcie uzyskano następujący profil zużycia energii jak na rysunku 3.1, o poborze 2.2uA w trybie głębokiego uśpienia.

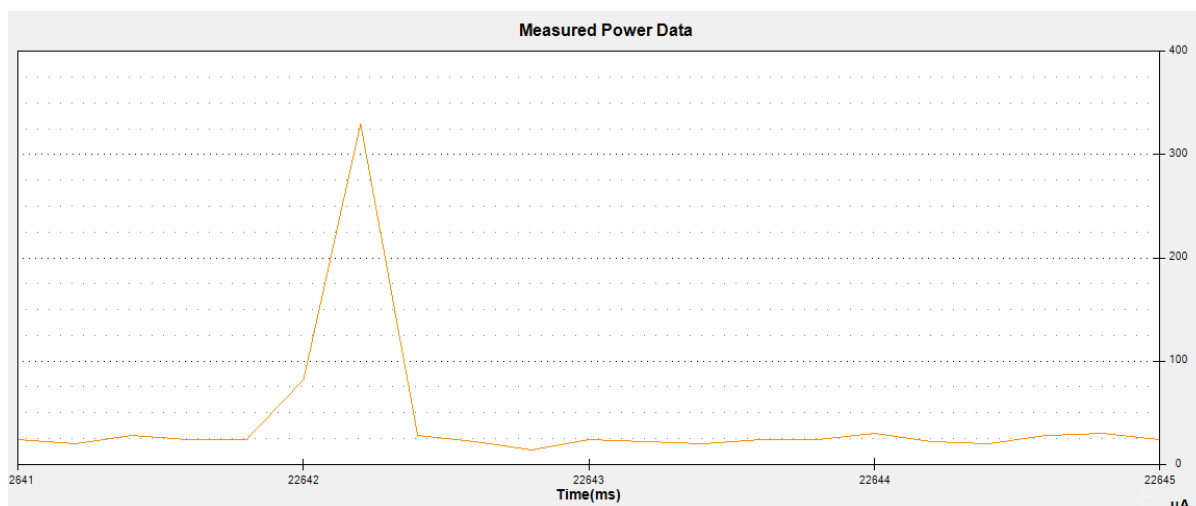


Rys. 3.1. Minimalny pobór prądu przy głębokim uśpieniu.

Na rysunku 3.1 dostrzec można też okresowe wybudzanie się procesora co osiem sekund. Spróbowano przeanalizować konsumpcję energii w tych momentach, jednak na rysunku 3.2 widoczna jest zbyt mała szybkość próbkowania by móc dokładnie przeanalizować zjawisko.

Zarejestrowany został szczytowy pobór energii 0.33mA, trwający około 0.5ms, autor założył więc, że zjawisko wybudzania między stanami uśpienia nie będzie miało istotnego wpływu na czas pracy urządzenia na baterii.

Dokładniejszą analizę przeprowadzono w rozdziale czwartym, za pomocą urządzenia pomiarowego o większej szybkości próbkowania.



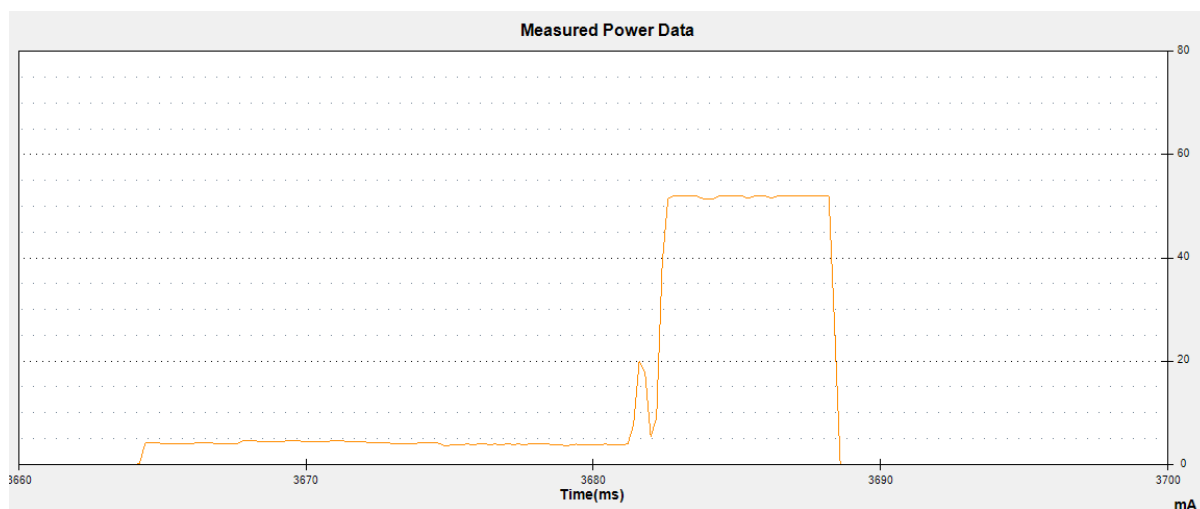
Rys. 3.2. Zbliżenie przebiegu w chwili wybudzania się układu

3.5. Prototyp ATmega328p z modułem komunikacyjnym RFM69

Rozpoczynając pracę nad sensorem opartym o ATmega328p-au oraz RFM69 wykorzystano optymalną konfigurację mikrokontrolera opisaną powyższej części pracy. Zaimplementowano obsługę modułu radiowego RFM69 w oparciu o bibliotekę autorstwa *LowPowerLabs*[14].

Zauważono iż pojedynczy pomiar oraz transmisja odbywają się w czasie około 25 milisekund, każda transmisja konsumuje około 0.12uAh.

Na rysunku 3.3 widoczny też jest charakterystyczny rozkład konsumpcji energii, w którym to transmisja przez moduł radiowy stanowi największą jego część.



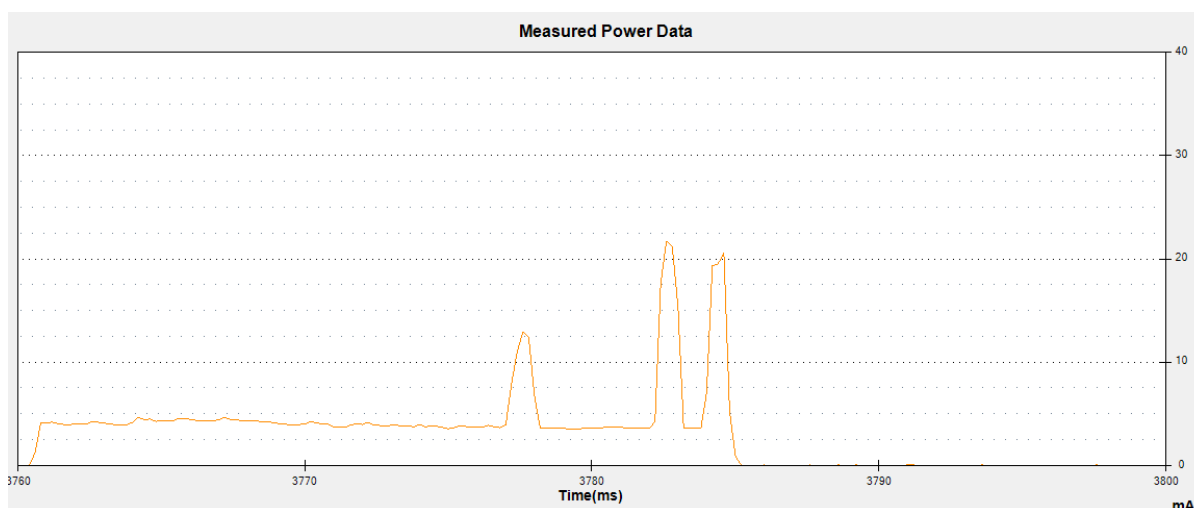
Rys. 3.3. ATmega328p z RFM69 - pojedynczy cykl pomiaru oraz wysłania danych

Zużycie energii w głębokim uśpieniu przez mikrokontroler i moduł radiowy

Korzystając z multimetru cyfrowego dokonano również pomiaru zużycia energii w stanie głębokiego uśpienia zarówno modułu radiowego oraz mikrokontrolera - uzyskując wynik 5.2uA.

3.6. Prototyp ATmega328p z modułem komunikacyjnym RFM75

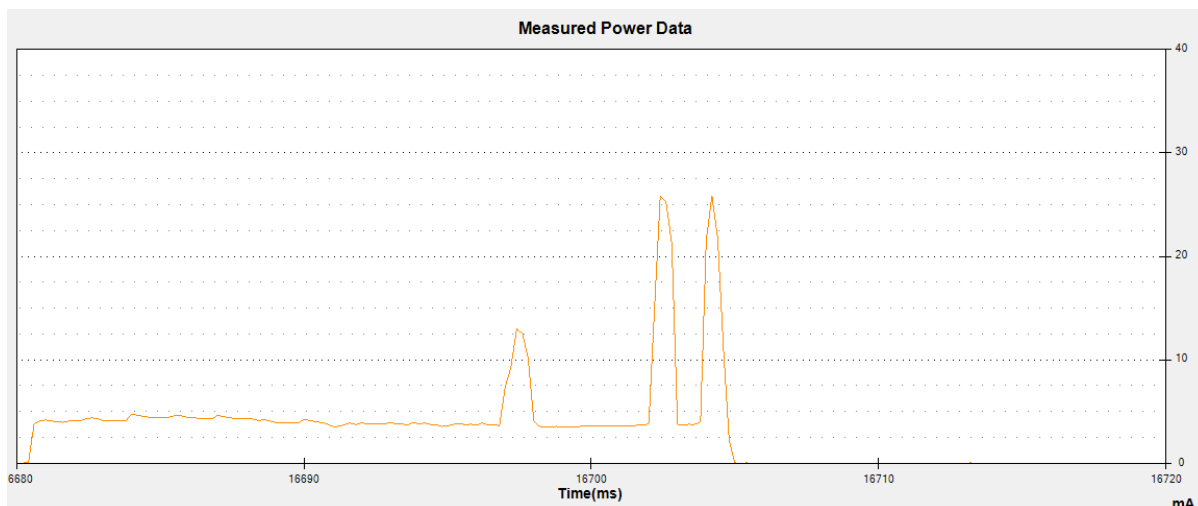
W podobny sposób zaimplementowano obsługę modułu komunikacyjnego RFM75, wstępnie ustawiając poziom sygnału na niski. Uzyskano przebieg jak na rysunku 3.4, czas trwania transmisji porównywalny do prototypu z modułem RFM69 na poziomie 25 milisekund, jednak koszt energetyczny transmisji był niższy - 0.03uAh (rysunek 3.4).



Rys. 3.4. ATmega328p z RFM75 - pojedynczy przebieg wysyłania danych przy ustawieniu mocy nadawczej na niską

Maksymalna moc nadawcza modułu radiowego

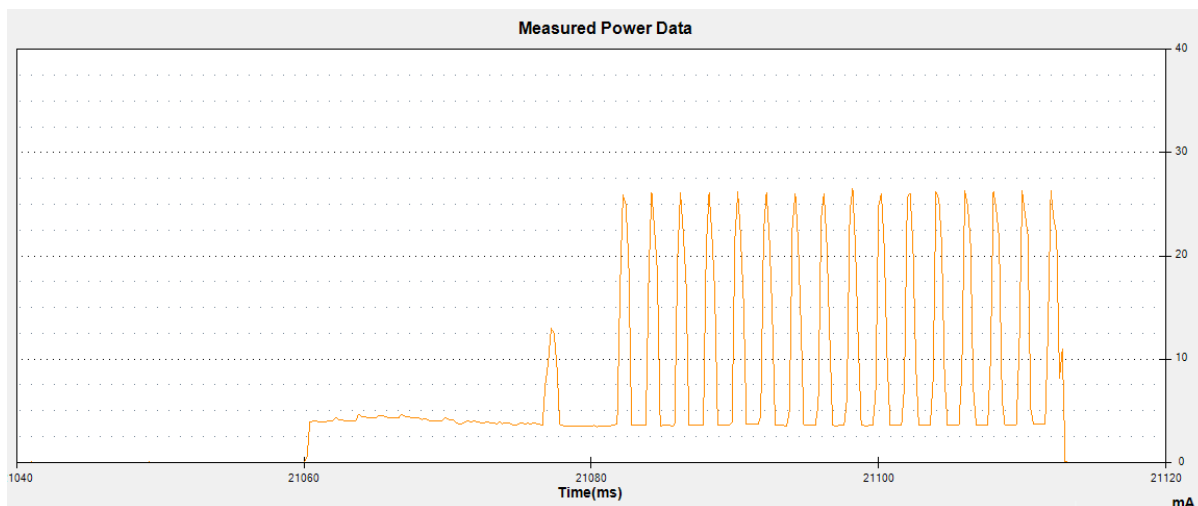
Przetestowano również maksymalną moc nadawczą modułu, jednak wpływ tego ustawienia na konsumpcję energii był ograniczony (rysunek 3.5), czas transmisji pozostał ten sam, jednak choć w szczytowych punktach pobór prądu wzrósł o około 25%, jednak mając na uwadze krótki czas trwania transmisji, nie niesie to wielkich konsekwencji dla ogólnego zużycia energii przez urządzenie.



Rys. 3.5. ATmega328p z RFM75 - pojedynczy przebieg wysyłania danych przy ustawieniu mocy nadawczej na maksimum

Powtórzenia transmisji

Zwrócono natomiast uwagę na znaczne wydłużenie czasu transmisji w przypadku gdy urządzenie odbiorcze było wyłączone (rysunek 3.6), co można jednak zniwelować ustawiając odpowiednio ilość dopuszczalnych powtórzeń.

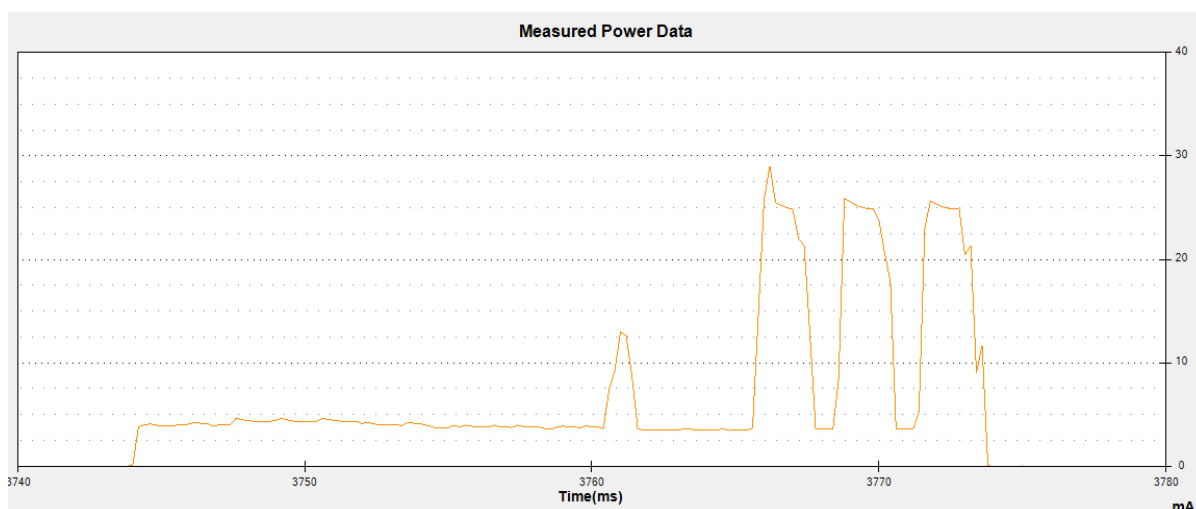


Rys. 3.6. ATmega328p z RFM75 - wpływ powtórzeń na zużycie energii

Ograniczenie przepustowości modułu radiowego

Ponieważ moduł komunikacyjny RFM75 ma możliwość ustawienia prędkości transmisji (od 250kbps do 2Mbps - domyślnie jest najwyższa), sprawdzono zachowanie modułu przy ograniczeniu prędkości transmisji do 250kbps, przebieg takiej sytuacji widoczny jest na rysunku 3.7. Czas pełnego cyklu wydłużył się do 30ms, a ogólny koszt energetyczny do 0.06uAh. Znane autorowi są jednak sytuacje w których

obniżenie prędkości nadawczej zdecydowanie poprawiało zasięgi uzyskiwane w oparciu o te moduły radiowe, jest to konfiguracja warta rozważenia.



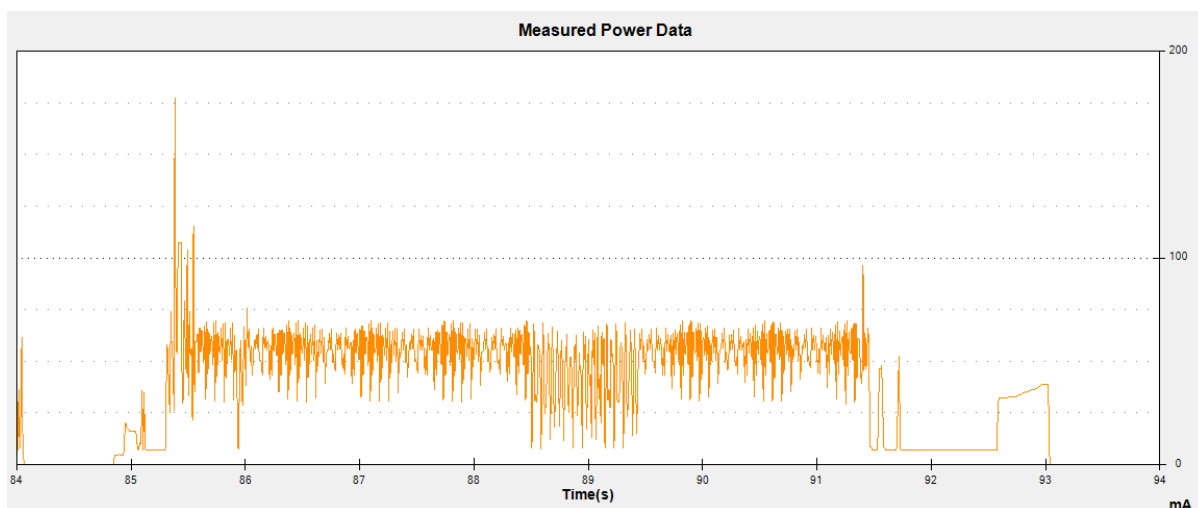
Rys. 3.7. ATmega328p z RFM75 - ograniczona przepustowość modułu radiowego

Zużycie energii w głębokim uśpieniu mikrokontrolera i modułu radiowego 2.4GHz

Korzystając z tej samej procedury jak dla poprzedniego modułu wykonano pomiar zużycia energii mikrokontrolera oraz modułu radiowego w stanie głębokiego uśpienia, uzyskując wynik: 8.4uA.

3.7. Prototyp ATmega328p z modułem Narrowband-IoT u-blox Sara N211

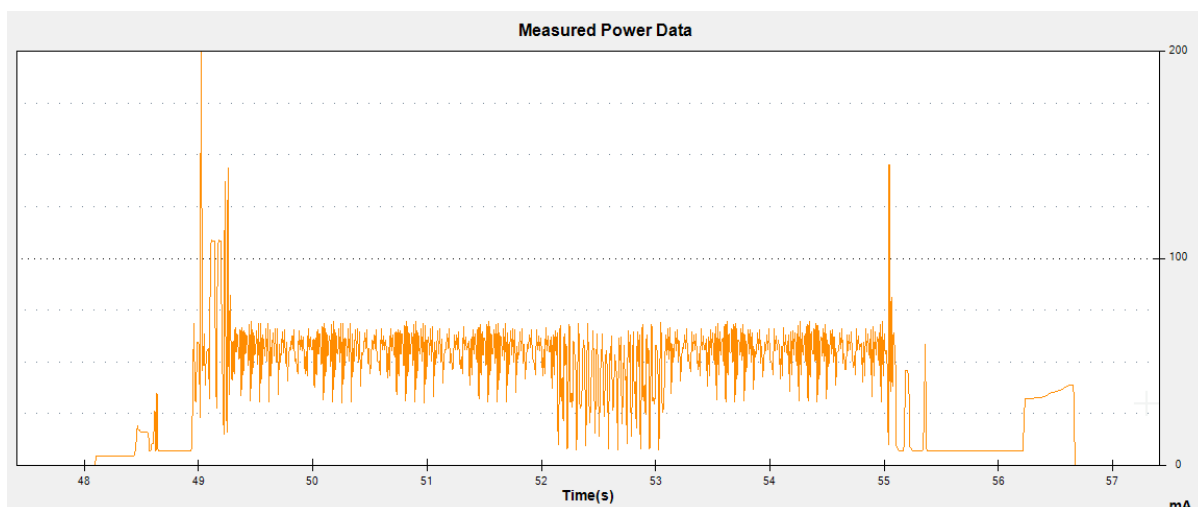
Test w podobnej konfiguracji wykonano również dla prototypu z modemem NB-IoT, najpierw zwiualizowano pobór energii podczas przesłania pojedynczego pomiaru bezpośrednio do serwera przez sieć Internet, za pomocą protokołu UDP (rysunek 3.8).



Rys. 3.8. ATmega328p z Narrowband-IoT - przesłanie pojedynczego pomiaru

Przesłanie wielu pomiarów w jednej transmisji

Autor zwrócił jednak uwagę na fakt, że sama transmisja danych jest jedynie marginalną częścią czasu aktywności modemu, możliwe więc jest w porównywalnym czasie i koszcie energetycznym przesłanie większej ilości danych. Wykonano pomiar (rysunek 3.9) dla przesłania dziesięciu pomiarów, którego czas trwania i konsumpcja energii były na zbliżonym poziomie.

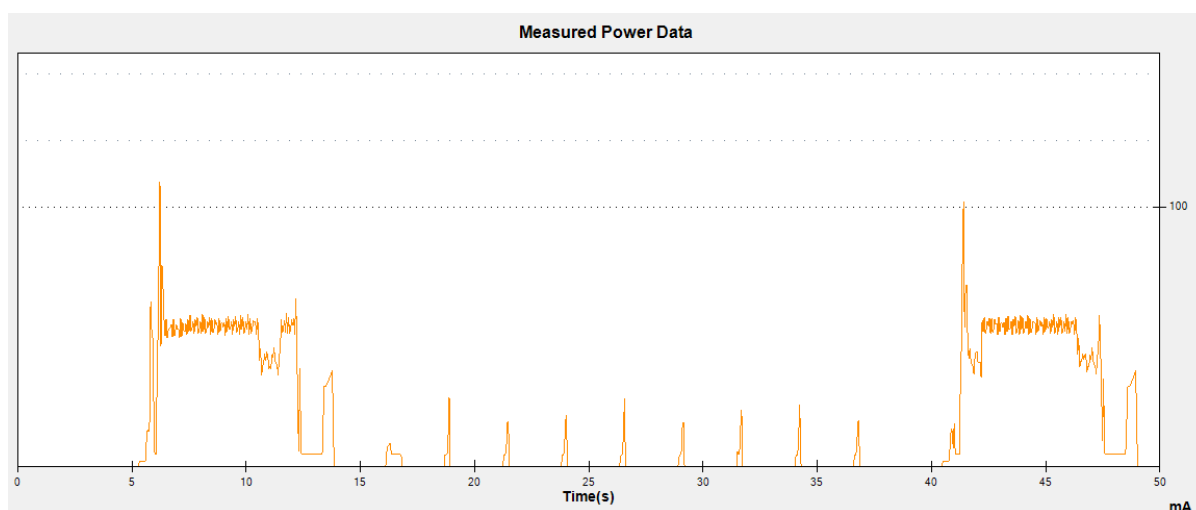


Rys. 3.9. ATmega328p z Narrowband-IoT - przesłanie dziesięciu pomiarów

Pomiar zużycia energii w głębokim uśpieniu

Zużycie energii w trybie uśpienia jest trudne do zmierzenia dla tej konfiguracji urządzenia. Pomiedzy pomiarami następuje wiele wybudzeń modemu wynikających z potrzeb synchronizacji z siecią komórkową. Możliwe jest ograniczenie tego zjawiska poprzez mechanizm *Relase Assistance* - po transmisji wychodzącej, jeżeli nie jest spodziewana żadna komunikacja przychodząca, połączenie może zostać zwolnione - nie jest to jednak wspierane w każdej z sieci. Autor zna również przypadki urządzeń, w których modem jest zupełnie wyłączany z zasilania, by nie aktywował się między transmisjami. Wtedy jednak procedura połączenia jest dłuższa, gdyż modem za każdym razem musi ponownie zestawić połączenie z siecią komórkową.

Wykorzystano narzędzie Power Monitor z dedykowanym oprogramowaniem by zmierzyć zużycie energii uśrednione w ciągu 10 minut pomiędzy transmisjami. Uzyskano wynik 220uAh, na rysunku 3.10 widoczne są okresowe wybudzenia modemu.



Rys. 3.10. ATmega328p z Narrowband-IoT - wybudzenia modemu pomiędzy pomiarami

Dzięki wykonanym pomiarom autor upewnił się w przekonaniu, że ilości energii konsumowane przez modem NarrowBand-IoT podczas transmisji oraz w trybie oszczędzania energii (PSM), pomimo iż są o wiele niższe niż dla klasycznych sieci komórkowych, są jednak zdecydowanie za duże dla budowanego rozwiązania.

3.8. Komunikacja WiFi

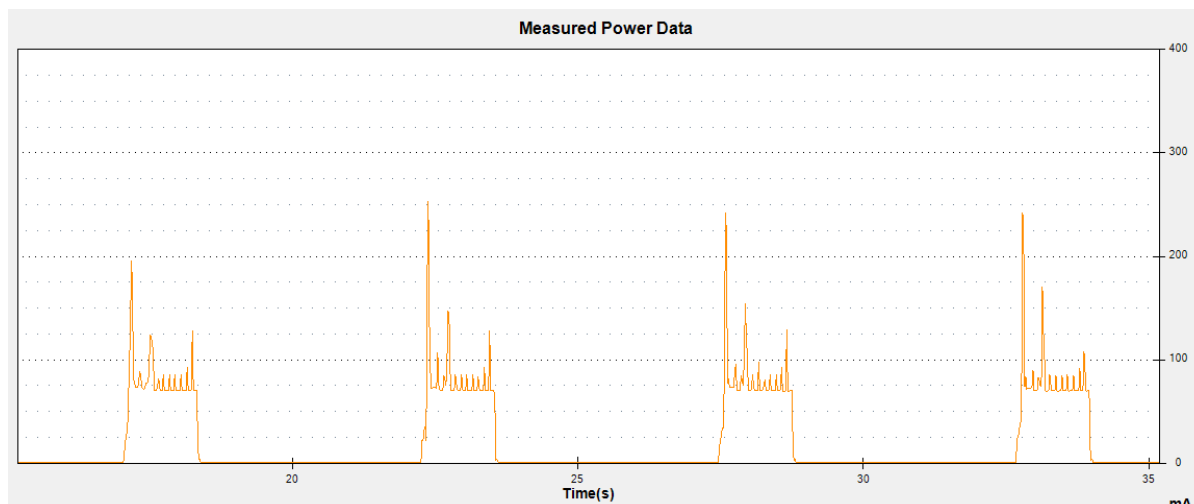
W przypadku prototypów platform pomiarowych opartych o ESP8266 oraz ESP32 z komunikacją WiFi, położono szczególny nacisk na optymalizację pracy części radiowej modułu, która konsumować będzie najwięcej energii elektrycznej.

3.8.1. Asocjacja z siecią WiFi

Przygotowano proste oprogramowanie (tabela 3.5), w którym mikrokontroler jedynie wykonuje asocjację z siecią WiFi, a następnie błyskawicznie przechodzi w tryb głębokiego uśpienia na cztery sekundy. Zaobserwowano przebieg zużycia energii (rysunek 3.11), na którym widoczne są asocjacje z punktem dostępowym trwające niemal półtorej sekundy, konsumujące około 30uAh energii.

```
1 #include <Arduino.h>
2 #include <ESP8266WiFi.h>
3
4 #define WIFI_SSID "testWiFi"
5 #define WIFI_PASSWORD "wifitest123"
6
7 void setup_wifi() {
8     WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
9     while (WiFi.status() != WL_CONNECTED) {
10         delay(3);
11     }
12 }
13 void setup() {
14     setup_wifi();
15 }
16 void loop() {
17     ESP.deepSleep(4e6);
18 }
```

Tabela 3.5. Minimalistyczny szkic dla ESP8266



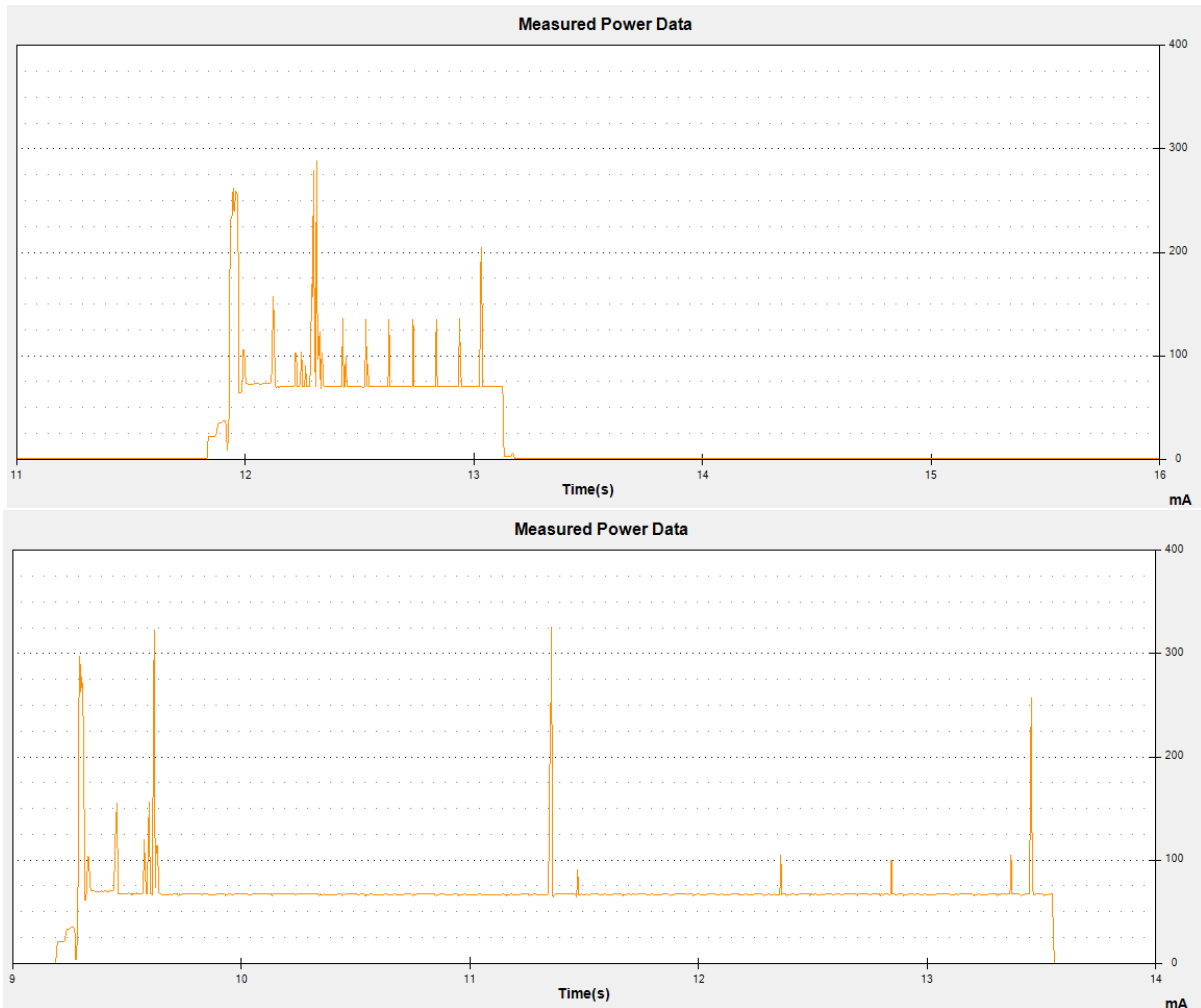
Rys. 3.11. Periodyczna asocjacja z siecią WiFi

Zwrócono jednak uwagę, że na czas asocjacji może mieć dodatkowo wpływ stopień obciążenia punktu dostępowego, jak i również jego wydajność. Wykonano porównanie między czasem asocjacji z routerem dedykowanym, do którego łączyło się tylko testowane urządzenie, a konsumenckim routerem będącym w użytku przez 5 innych urządzeń. Obydwa urządzenia skonfigurowane były w ten sam sposób (ta sama nazwa sieci, sposób zabezpieczeń), umieszczone zostały w takiej samej odległości od urządzenia testowanego.

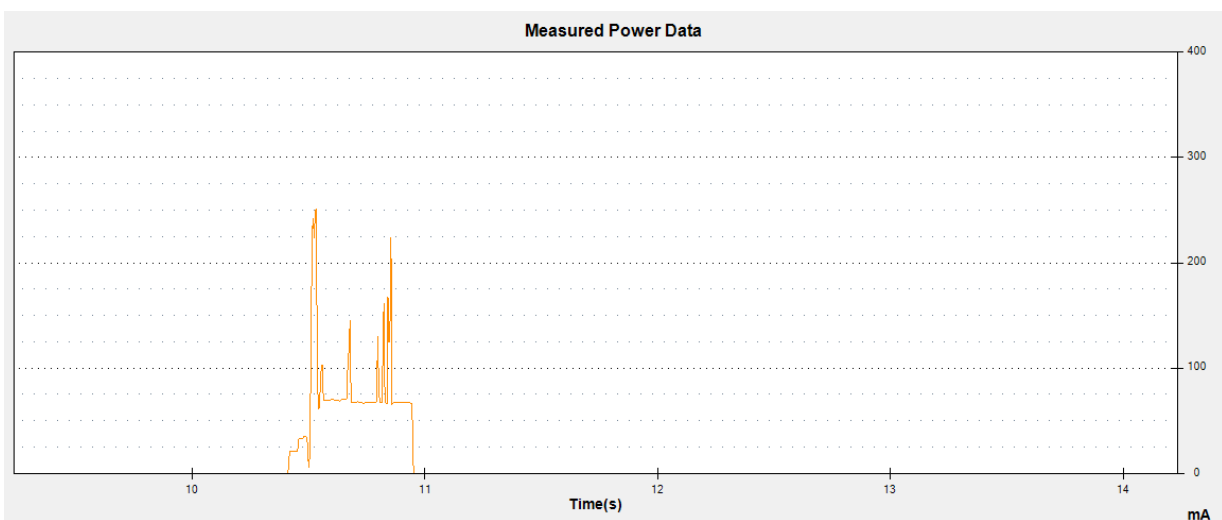
Przy dedykowanym punkcie dostępowym czas asocjacji wynosił około 1.36s, o koszcie energetycznym ok 30uAh gdzie testując konsumenckim urządzeniu końcowym dostarczonym przez operatora, obciążonym wieloma innymi urządzeniami ten czas wydłużał się do ok 4.46s, konsumując nawet 80uAh energii. Porównanie widoczne jest na rysunku 3.12.

Skrócenie czasu asocjacji - statyczna adresacja IP

Podjęto próbę skrócenia czasu asocjacji z siecią poprzez ustawienie urządzeniu statycznej adresacji IP (tabela 3.6). W efekcie uzyskano powtarzalny czas asocjacji około 600ms, o koszcie energetycznym około 12uAh (rysunek 3.13).



Rys. 3.12. Porównanie czasu asocjacji (u góry: dedykowany punkt dostępowy, na dole: domowy router z którego korzysta więcej użytkowników)



Rys. 3.13. Pojedyncza asocjacja z siecią WiFi przy ustawionym statycznym adresie IP

```
1 void setup_wifi() {
2     IPAddress ip(172, 18, 0, 200);
3     IPAddress gw(172, 18, 0, 1);
4     IPAddress snet(255, 255, 255, 0);
5
6     WiFi.config(ip, gw, snet);
7
8     WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
9
10    while (WiFi.status() != WL_CONNECTED) {
11        delay(3);
12    }
13 }
```

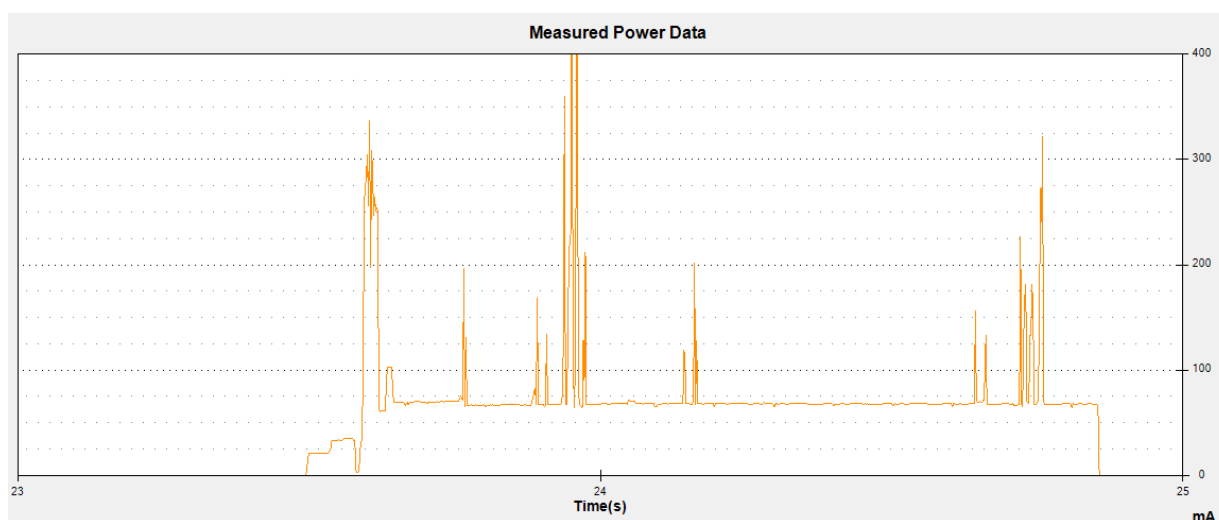
Tabela 3.6. Konfiguracja statycznej adresacji dla ESP8266

3.8.2. Przesyłanie pomiarów

Powszechnym wyborem w przypadku Internetu Rzeczy opartego o sieci TCP/IP jest protokół MQTT (*ang. MQ Telemetry Transport*), który zapewnia dużą wygodę i elastyczność w przesyłaniu danych.

Zaimplementowano przesyłanie danych w formie binarnej w oparciu o ten protokół z wykorzystaniem biblioteki *PubSubClient*². Ponieważ biblioteka nie wspiera sygnalizacji zakończenia transmisji danych, konieczne było dodanie opóźnienia przed przejściem w stan głębokiego uśpienia, o wartości 50ms - dobrane doświadczalnie.

Zaobserwowano przebieg widoczny na rysunku 3.14 trwający około 1.4s, o koszcie energetycznym 28uAh.

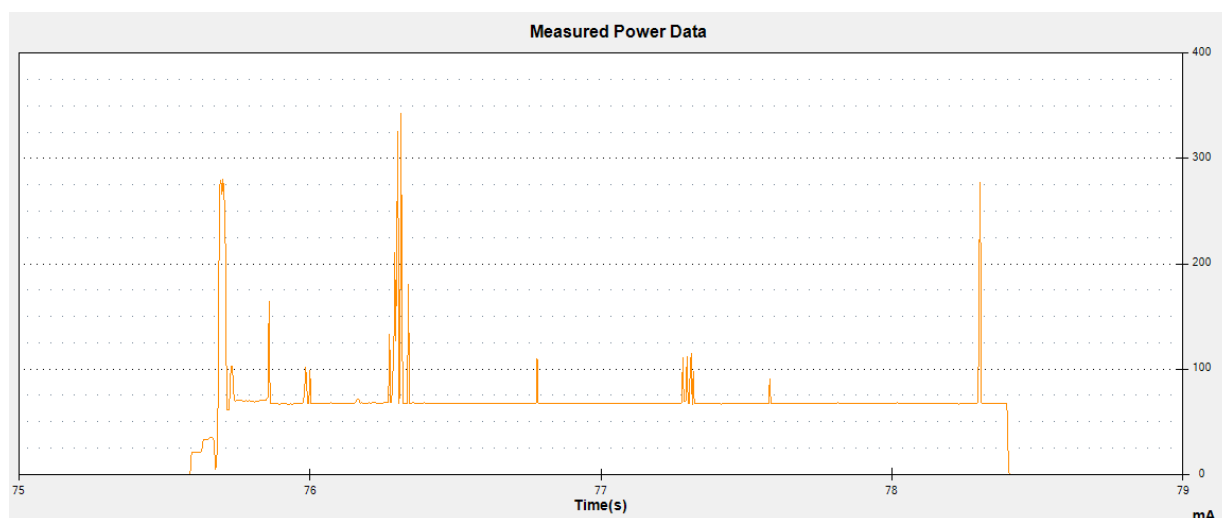


Rys. 3.14. Przesyłanie danych z użyciem MQTT

²<https://github.com/knolleary/pubsubclient>

Zauważywszy znaczny narzut czasowy i energetyczny zastosowanego protokołu MQTT rozważono możliwość ograniczenia poboru energii poprzez zastosowanie prostego protokołu opartego o surowy socket TCP, jednak nie zauważono żadnej znaczącej różnicy w stosunku do użycia protokołu MQTT.

Mając na względzie dodatkowy narzut związany z mechanizmami zapewniania niezawodności TCP, rozważono też zastosowanie komunikacji na gniazdach UDP. Jednak efekt był przeciwny do zamierzonego - czas transmisji wzrósł do 2.86s, a zużyta energia do 55.07uAh (rysunek 3.15). Wynika to prawdopodobnie z niedoskonałości implementacji biblioteki UDP dla ESP8266.



Rys. 3.15. Przesyłanie danych z użyciem gniazda UDP

Pobór prądu w stanie uśpienia układu ESP8266

Następnie urządzenie zostało skonfigurowane by czas uśpienia wynosił 30 sekund pomiędzy pomiarami i wykonano pomiar multimetrem cyfrowym podobnie jak dla prototypów z mikroprocesorem AVR. Uzyskano wynik 126.4uA.

3.9. Podsumowanie wyników przeprowadzonych pomiarów

Wyniki przeprowadzonych pomiarów zestawiono w tabeli 3.7. Na podstawie danych pomiarowych oszacowano czas pracy urządzenia na akumulatorze o pojemności 1000mAh, założywszy przesyłanie pomiaru co 10 minut.

	czas trwania transmisji	koszt energetyczny	stan uśpienia
ATmega328 + RFM75	30ms	0.06uAh	8.4uA
ATmega328 + RFM69	25ms	0.12uAh	5.2uA
ATmega328 + Sara N211	8.46s	100uAh	220uA
ESP8266 + MQTT	1.4s	28uAh	126.4uA

Tabela 3.7. Wyniki pomiarów

Obliczono koszt energetyczny doby pracy urządzenia, uzyskano wynik w mikroamperogodzinach.

$$dobaPracy = \left(\frac{60}{10} [minut] \times kosztTransmisji[uAh] + trybCzuwania[uA] \times 1[h]\right) \times 24[h] \quad (3.1)$$

Następnie pojemność baterii w mikroamperogodzinach podzielono przez uzyskany koszt pracy przez 24 godziny, by uzyskać wynik w dniach:

$$czasPracy = \frac{10^6}{dobaPracy} \frac{[uAh]}{d} \quad (3.2)$$

	koszt energetyczny / dzień	czas pracy w dniach	czas pracy w latach
ATmega328 + RFM75	210.24 uAh	4756	13
ATmega328 + RFM69	142.08 uAh	7038	19
ATmega328 + Sara N211	19680 uAh	50	0.14
ESP8266 + MQTT	7065.60 uAh	141	0.4

Tabela 3.8. Wyniki pomiarów

3.9.1. Wybór platformy pomiarowej do dalszej pracy

Zastanawiając się nad wyborem platformy pomiarowej do wykonania docelowej wersji sensora w oparciu o wyniki przedstawione w tabeli 3.8 autor wstępnie wykluczył opcję komunikacji WiFi oraz NarrowBand-IoT, przez wzgląd na niewspółmiernie wyższe zużycie energii w porównaniu do modułów radiowych w paśmie ISM. Rozwiązania te są atrakcyjne w przypadku budowy pojedynczego urządzenia, mającego działać niezależnie - czy to w oparciu o dostępną infrastrukturę WiFi, czy też sieć operatora. Konieczne byłoby też stosowanie dość dużych baterii (jakie z resztą są zalecane w przypadku modułów Nb-IoT przez producentów).

Kierując się zamysłem opracowania miniaturowej platformy pomiarowej, która umożliwi budowanie sieci sensorowej o miniaturowych rozmiarach, zdecydowany nacisk położono na rozwiązanie opierające się o transmisję w paśmie ISM.

W przypadku modułów RFM69 oraz RFM75, autor zauważył iż pomimo tego że moduł w paśmie 868MHz (RFM69) konsumuje dwukrotnie więcej energii, jednocześnie charakteryzuje się niższym poborem energii w stanie głębokiego uśpienia. Dodatkowo autor ma świadomość, że w paśmie 2.4GHz pracuje bardzo dużo różnych urządzeń elektronicznych, spodziewać się może więc zakłóceń, mających negatywny wpływ na zasięg i sprawne funkcjonowanie urządzenia.

Ostatecznie wybrano kombinację mikroprocesora ATmega328p z modułem radiowym RFM69.

4. Wykonanie sensora w docelowej wersji

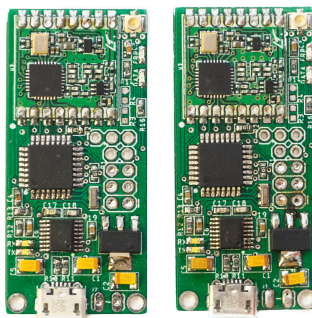
W ostatnim rozdziale pracy opisany został proces wykonania finalnej wersji budowanego sensora, optymalizacji jego pracy oraz dodatkowych akcesoriów koniecznych do jego testowania i użytkowania. W tym rozdziale wykorzystane zostało urządzenie Otii Arc do precyzyjnego zmierzenia zużycia energii przez zaprojektowany układ.

4.1. Dodatkowe narzędzie do testowania - brama dostępowa

Mając na uwadze konieczność wygodnego testowania budowanego rozwiązania opracowano dwie wersje bramy dostępowej, które posłużyły do testowania budowanego rozwiązania i zbierania danych z czujnika w komputerze.

Moduł USB z radiem RFM69HCW

Aby wygodnie podłączyć moduł RFM69HCW autor zaprojektował i wykonał proste urządzenie z interfejsem USB (widoczne na rysunku 4.1). Składa się ono z konwertera USB do UART, mikrokontrolera ATmega328p, modułu RFM69HCW oraz koniecznych elementów pasywnych. Urządzenie zasilane jest z portu USB poprzez regulator stabilizujący napięcie pracy do 3.3V.

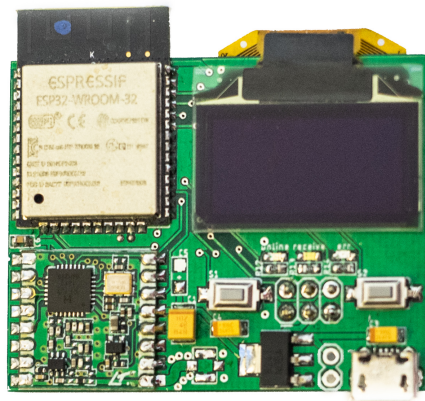


Rys. 4.1. Wykonane moduły USB z radiem RFM69HCW

Dane odbierane przez moduł radiowy przekazywane są do komputera poprzez magistralę szeregową w formie heksadecymalnej co umożliwi zarówno powtarzalne odczytanie danych poprzez oprogramowanie, jak i również jest to format czytelny dla człowieka, co pozwala na szybką ocenę otrzymanych wyników.

Bramka komunikacyjna WiFi z radiem RFM69HCW w oparciu o ESP32

Mając na względzie budowę sieci sensorowej opartej o zwielokrotnienie budowanego urządzenia opracowano również bramę dostępową z WiFi oraz wyświetlaczem OLED (zdjęcie 4.2). Jest to prototyp urządzenia, które mogłoby stanowić zarówno bramę komunikacyjną dla sieci sensorowej do Internetu, jak i również interfejs prezentujący dane zbierane z czujników.

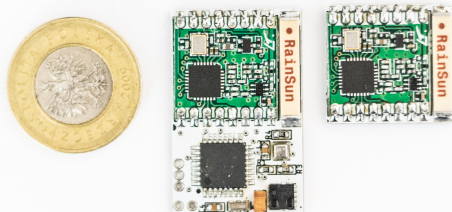


Rys. 4.2. Bramka komunikacyjna w oparciu o ESP32 i radio RFM69HCW

4.2. Ostateczna konfiguracja sprzętowa urządzenia

Autor zdecydował się na wykonanie czujnika w dwóch wersjach (zdjęcie 4.3):

- podstawowej, o standardowym rozmiarze, z dodatkowym układem pomiarowym VCNL4010 - sensorem światła i odległości, który charakteryzuje się niską konsumpcją energii - około 1.5uA w trybie uśpienia, oraz 2.5uA podczas wykonywania dwóch pomiarów natężenia światła na sekundę[15],
- miniaturowej, wykorzystującej układ ATMege w mniejszej obudowie, oraz z elementami umieszczonymi na obu stronach płyty PCB.

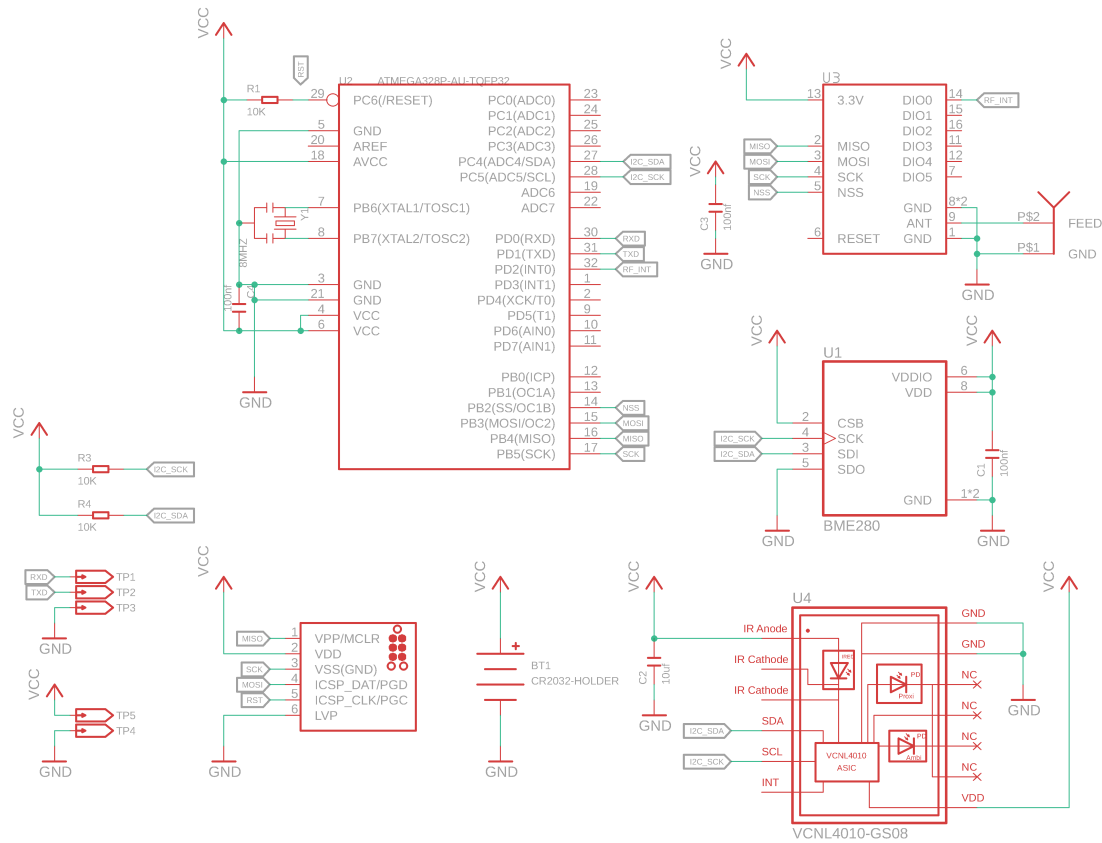


Rys. 4.3. Dwie wersje wykonanego czujnika

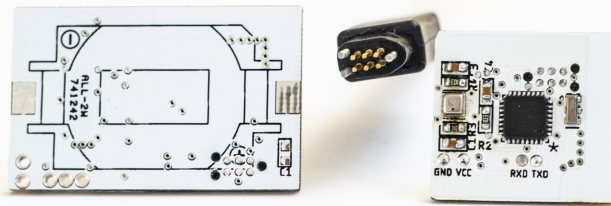
Obydwa urządzenia wykonane zostały na bazie tego samego schematu (rysunek 4.4) podobnego do przedstawionego w rozdziale trzecim, różnią się jedynie wersją obudowy układu ATMege oraz w wersji miniaturowej autor zrezygnował z dodatkowego czujnika światła.

Złącze programowania układu

By ułatwić wgrywanie oprogramowania mikrokontrolera, oraz oszczędzić miejsce na PCB wykorzystano sześciopinowe złącze *Tag-Connect*, jest to kabel ze specjalną końcówką z sześcioma elastycznymi pinami oraz trzema stałymi bolcami do pozycjonowania złącza. Na pola kontaktowe wyprowadzony został standardowy interfejs ICSP wraz z zasilaniem (rysunek 4.5). W kombinacji z odpowiednią podstawką pod płytę PCB umożliwiłoby to seryjne wgrywanie oprogramowania do urządzeń, bez konieczności lutowania dużych wtyków goldpin jak we wcześniejszych prototypach przedstawionych w rozdziale drugim.



Rys. 4.4. Schemat finalnej konfiguracji sprzętowej

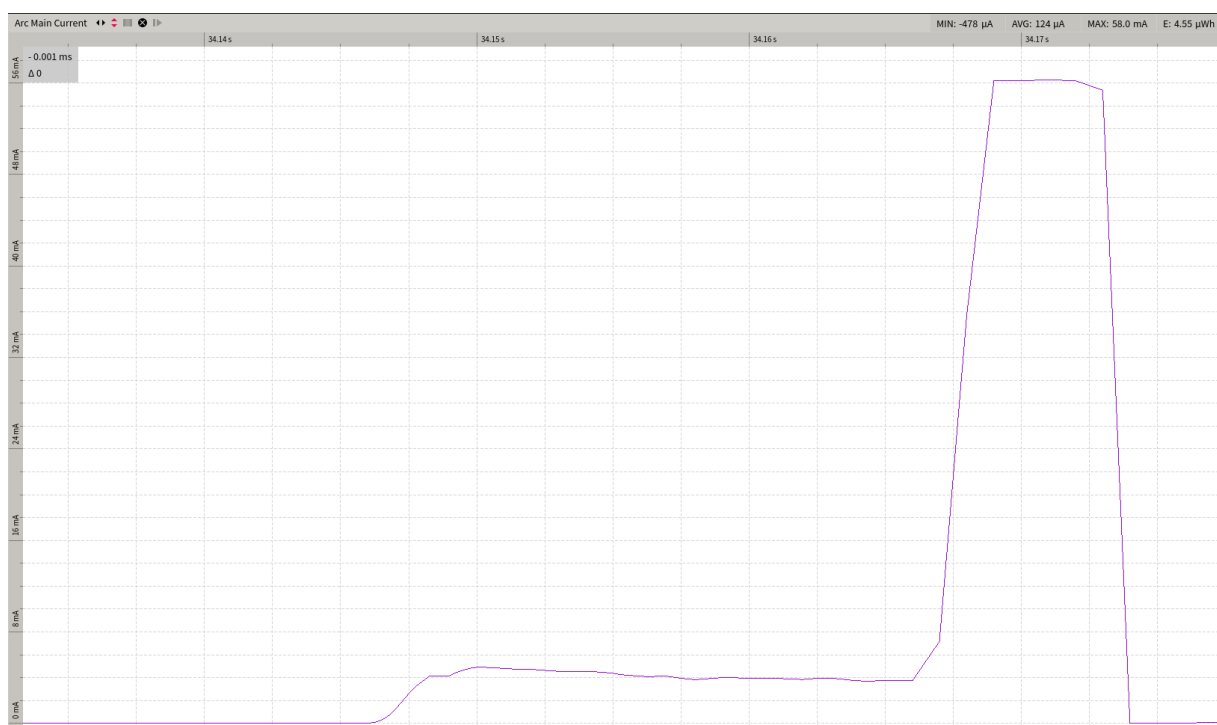


Rys. 4.5. Złącza do wgrzywania oprogramowania na dolnej warstwie wykonanych PCB oraz przewód z dedykowaną końcówką

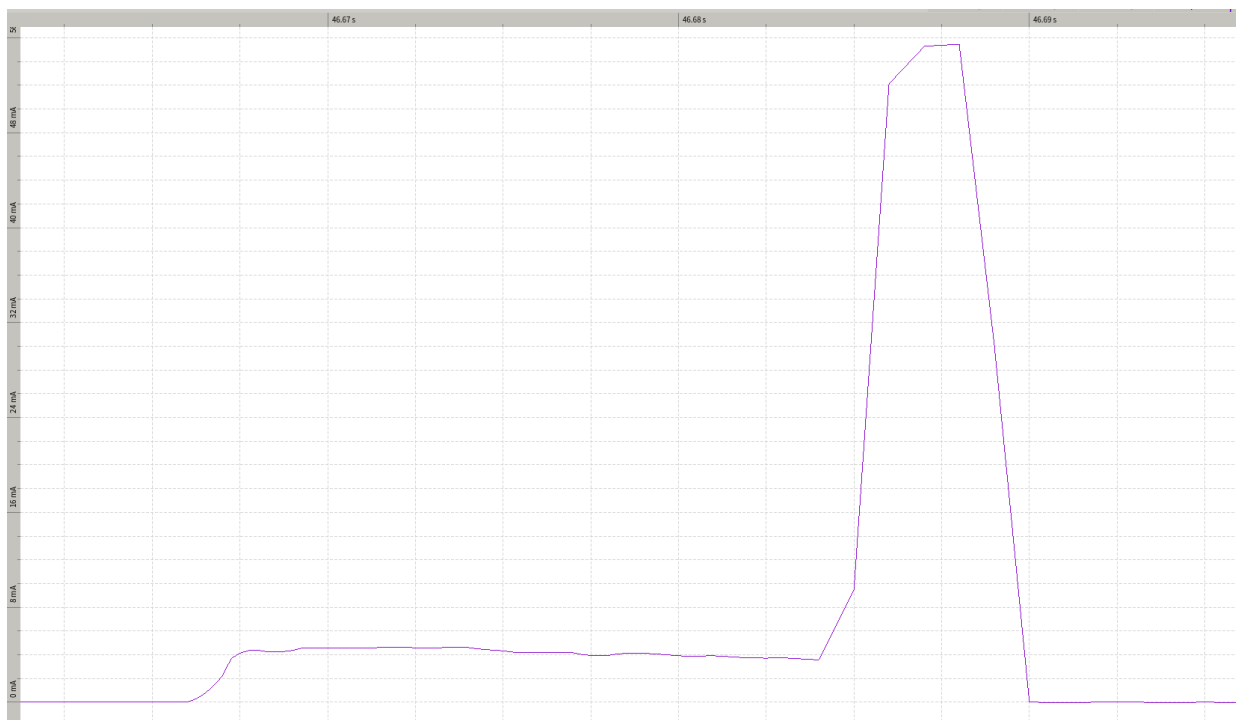
4.3. Optymalizacja zużycia energii finalnej konfiguracji sprzętowej

Wykorzystując narzędzie Otii Arc wykonano optymalizację konsumpcji energii przez finalną konfigurację sprzętową opracowanego sensora. Na rysunku 4.6 przedstawiono przebieg zużycia energii przez urządzenie bez żadnych modyfikacji. Zaobserwowano wyniki zbliżony do wcześniejszego prototypu - czas transmisji to około 26 milisekund, koszt energetyczny na poziomie 370nWh, przy napięciu 3.3V, czyli wartość ok 0.11uAh. Uzyskany wynik pokrywa się z pomiarem wykonanym za pomocą narzędzia PowerMonitor

Następnie zmodyfikowano działanie biblioteki do obsługi sensora, by nie wywoływać wielokrotnie pomiaru temperatury (wymagany jest do kompensacji odczytu wartości ciśnienia atmosferycznego i wilgotności), wyłączono też zbędną magistralę UART. W efekcie uzyskano ostateczny pomiar zużycia prądu przez cykl pomiarowy czujnika (rysunek 4.7) - pojedynczy pomiar oraz transmisja danych trwa ok 24 milisekund i konsumuje ok 300nWh, czyli ok 0.09uAh.



Rys. 4.6. Zużycie energii podczas dokonywania pomiaru i transmisji danych przez finalną konfigurację sprzętową



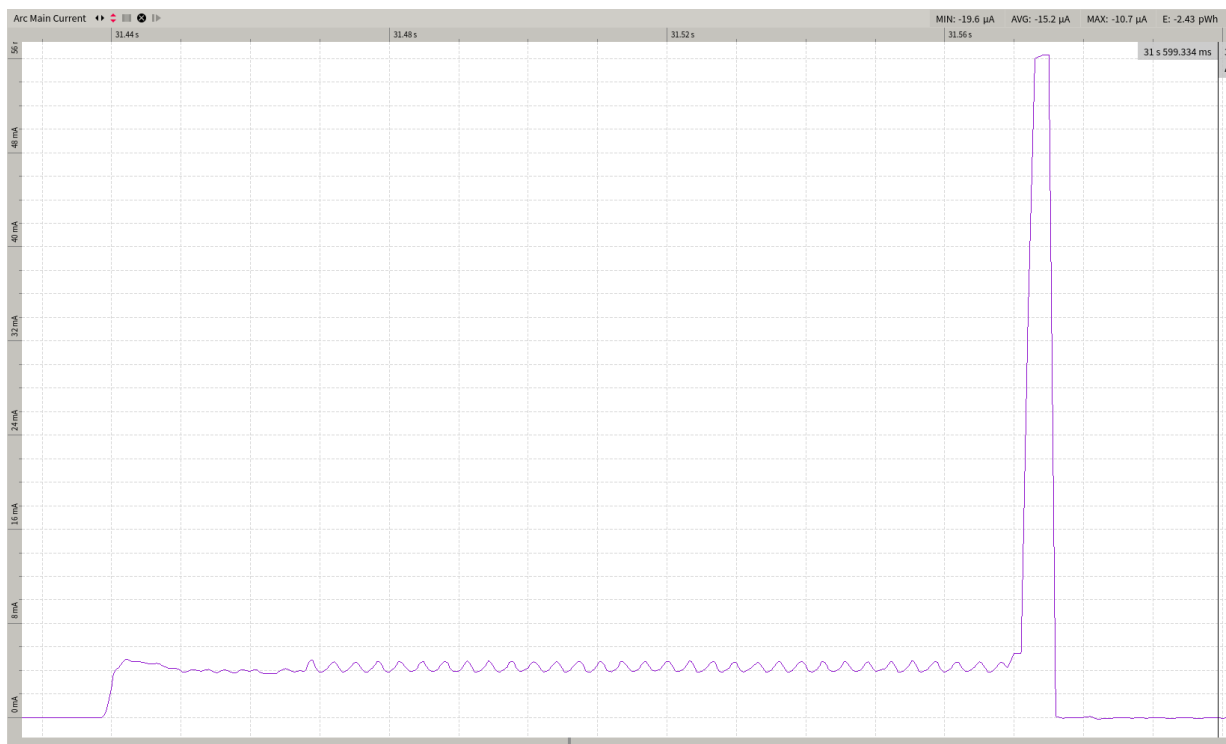
Rys. 4.7. Uzyskany przebieg zużycia energii przez czujnik po optymalizacji

Włączenie dodatkowego sensora światła

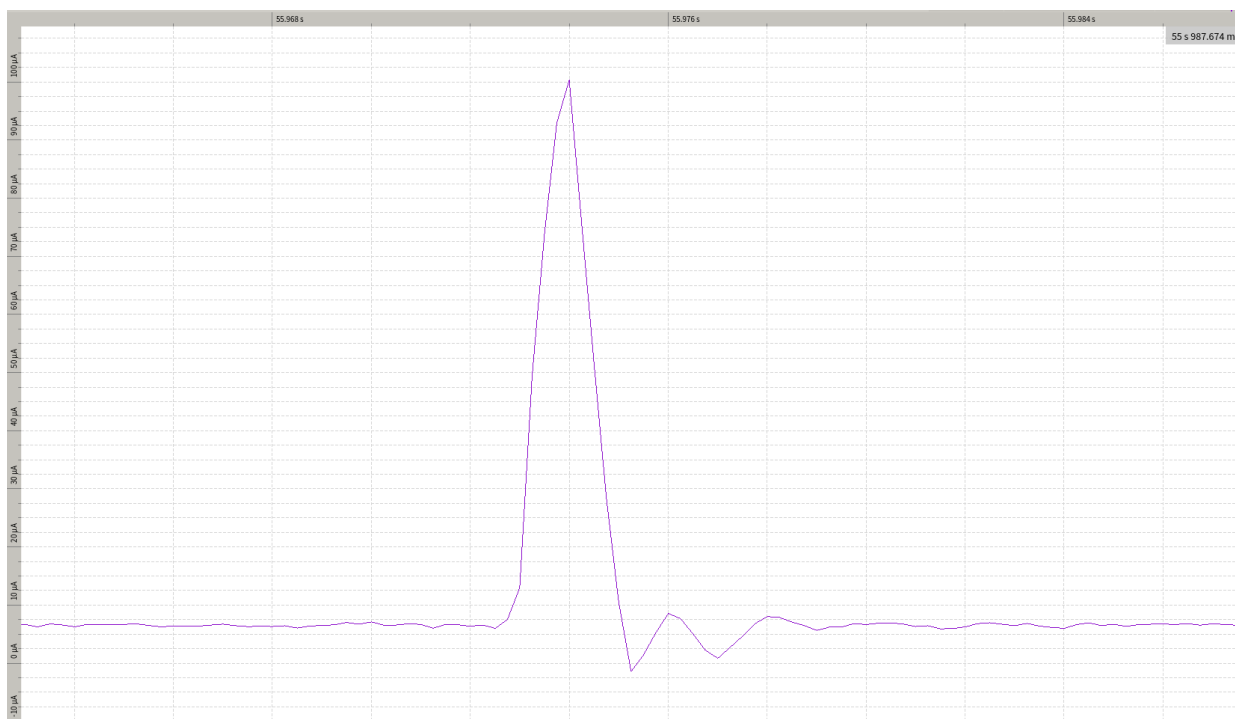
W związku z uzyskanym wynikiem rozważono rozszerzenie funkcjonalności urządzenia o dodatkowy sensor światła, co uwzględniono w projekcie PCB finalnej wersji. Dodano do oprogramowania wsparcie dla tego czujnika, odczytując z niego jedynie wartość natężenia światła otoczenia. Na przebiegu zużycia energii (rysunek 4.8) znaczne wydłużenie czasu aktywności mikrokontrolera do około 140 milisekund, niemalże podwajając koszt energetyczny do wartości 800nWh, co zgodne jest z notą aplikacyjną[16] czujnika VCNL4010 - pojedynczy pomiar trwa około stu milisekund. Dla takiej kombinacji sensorów wynik nadal jest zadowalający, czas pracy urządzenia na baterii oszacowano na około siedemnaście lat.

Weryfikacja zużycia energii przez okresowe wybudzenia mikroprocesora w stanie uśpienia

Jak opisano w rozdziale trzecim, mikrokontroler ATmega328p może utrzymywać stan uśpienia maksymalnie przez osiem sekund, cykl wybudzeń wykonywany jest więc wielokrotnie, by uzyskać oczekiwany czas pomiędzy wykonywanymi pomiarami. Wykorzystywane wcześniej urządzenie pomiarowe Power Monitor nie pozwoliło na obserwację tego zjawiska, przez niewystarczającą szybkość próbkowania. Wykorzystując nowe urządzenie pomiarowe ponowiono pomiar (przedstawiony na rysunku 4.9) potwierdzając zaniechwalną konsumpcję energii w momentach wybudzenia mikroprocesora.



Rys. 4.8. Wydłużenie czasu aktywności mikrokontrolera podczas wykonywania pomiaru przez sensor światła



Rys. 4.9. Zużycie energii podczas chwilowych wybudzeń mikrokontrolera

Sprawdzenie funkcjonowania urządzenia przy obniżonym napięciu zasilania

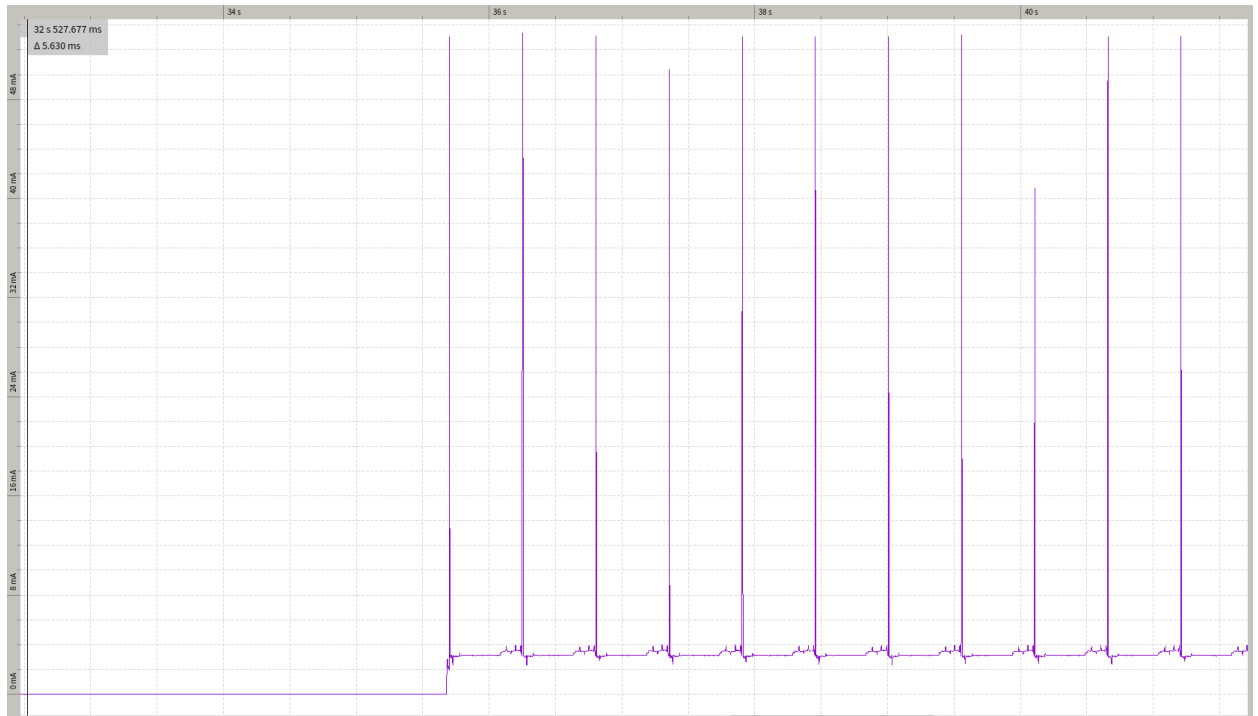
Ponieważ zarówno mikrokontroler jak i moduł radiowy wspierają zasilanie napięciem 1.8V, sprawdzono zachowanie urządzenia i konsumpcję energii przy obniżonym napięciu zasilania. Jednak jak widać na rysunku 4.10, urządzenie nie zainicjalizowało poprawnie peryferiów i nie potrafiło przejść do stanu uśpienia"



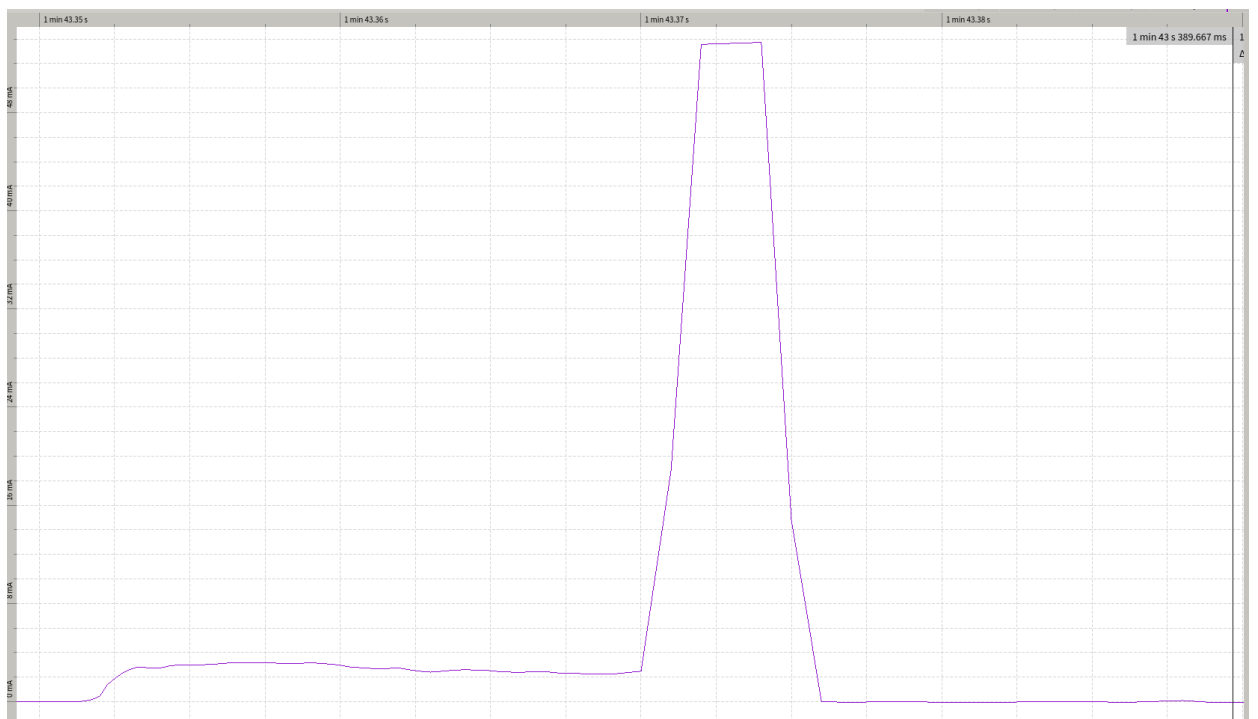
Rys. 4.10. Błędne funkcjonowanie urządzenia przy napięciu zasilania 1.8V

Podniesiono więc napięcie zasilania do 2V, jednak to nie pozwoliło na przesłanie danych (rysunek 4.11) pomimo poprawnego zainicjalizowania urządzenia.

Dopiero przy napięciu 2.2V urządzenie funkcjonowało poprawnie (4.12) i z sukcesem transmitowało odczyty. Uzyskano przy tym zdecydowanie mniejszą konsumpcję energii - na poziomie 150nWh. Świadczy to o tym, że możliwe będzie wykorzystanie niemalże całej energii zgromadzonej w bateriach litowych (napięcie odcięcia 2V).



Rys. 4.11. Błędne funkcjonowanie urządzenia przy napięciu zasilania 2.0V

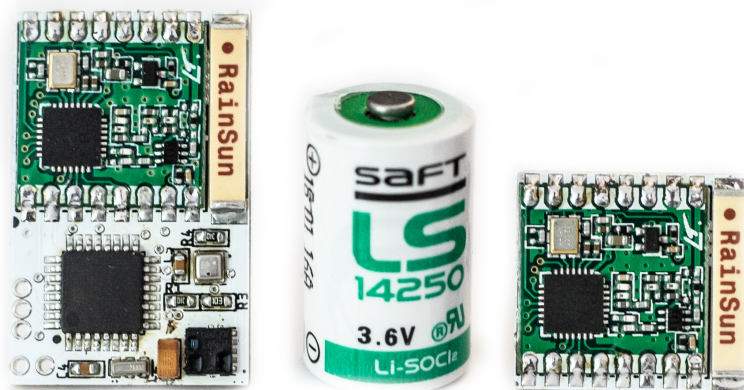


Rys. 4.12. Zużycie energii przy napięcia 2.2V

4.4. Zasilanie opracowanego czujnika

Ponieważ wykonane urządzenie pobiera w szczytowych momentach (podczas transmisji danych) około 55mA, nie było możliwe wykorzystanie baterii pastylkowych, dla których zalecany jest pobór prądu do kilku miliamper.

Jako finalne źródło zasilania autor wykorzystał baterie litowe Saft LS14250 w kształcie cylindra o szerokości 14.5mm oraz wysokości 24.5mm, o pojemności nominalnej 1200mAh[17], rozmiar ten adekwatny jest do rozmiaru elektroniki czujnika co przedstawiono na zdjęciu 4.13. Nominalny prąd rozładowania tej baterii to tylko 35mA, jednak producent dopuszcza chwilowe skoki poboru prądu nawet do 100mA przez 100ms, tak więc szczytowy pobór prądu na poziomie 55mA przez około 5 milisekund podczas transmisji nie powinien wpłynąć negatywnie na czas pracy baterii.



Rys. 4.13. Bateria wykorzystana do zasilania czujnika

4.5. Finalna estymacja czasu pracy opracowanego czujnika na baterii

Napięcie nominalne wykorzystanej baterii to 3.6V, konieczne było więc ponowienie pomiarów zużycia energii przez finalne wersje czujników dla tego napięcia.

Następnie posługując się zależnościami 3.1, 3.2 z rozdziału trzeciego dokonano estymacji czasu pracy opracowanego czujnika zasilanego zastosowaną baterią. Założywszy wykorzystanie całej energii zgromadzonej w baterii, oraz dokonywanie pomiarów co dziesięć minut otrzymano bardzo optymistyczne estymaty przedstawione w tabeli 4.1.

	zużycie energii jednej transmisji	zużycie energii w stanie uśpienia	czas działania w latach
czujnik warunków atmosferycznych	0.08uAh	5.2uA	24 lata
czujnik warunków atmosferycznych oraz natężenia światła	0.27uAh	6.5uA	16 lat

Tabela 4.1. Szacunkowy czas działania

Należy zwrócić jednak uwagę na fakt samoczynnego rozładowania się akumulatora (około 1% rocznie), przy testach prowadzonych przez dziesięć lat[18]. Jednocześnie broszury instruktażowe producentów baterii wspominają maksymalny czas życia urządzeń do dwunastu lat [19]. Na samoczynne rozładowanie ogniw ma również temperatura otoczenia, co może zostać wykorzystane do oceny starzenia ogniw [20], oraz cykl rozładowania baterii[21].

Mając na względzie wyżej wspomniane aspekty autor założył konieczność wymiany baterii w opracowanym czujniku raz na dziesięć lat.

4.6. Kierunki dalszego rozwoju

Budowa czujnika bez baterii - odzyskiwanie energii

Mając na uwadze jak niskie średnie zużycie energii udało się uzyskać w opracowanym czujniku, autor rozważa dalszy rozwój urządzenia z wykorzystaniem odzyskiwania energii by zbudować czujnik zupełnie pozbawiony klasycznej baterii. Możliwe jest pozyskiwanie energii z fal elektromagnetycznych sygnałów radiowych[22], zarówno ze źródeł dedykowanych zasilaniu jak i również z sygnałów występujących w środowisku. Istnieją układy specjalizowane pozyskiwanie energii z fal elektromagnetycznych i przechowywaniu ich w superkondensatorach, jak na przykład układ AEM30940, który poza interfejsem do odzyskiwania energii (już od 3uW) posiada dwa regulatory napięcia (niskonapięciowy 1.8V oraz drugi regulowany 1.8-4.1V mogący dostarczyć do 80mA), który mógłby posłużyć do zasilenia oddzielnie mikrokontrolera oraz modułu radiowego w czujniku [23].

Budowa sieci sensorowej z czujnikami o różnych parametrach

W oparciu o opracowaną konfigurację sprzętową możliwa jest budowa wielu różnych bezprzewodowych czujników, które mogłyby posłużyć do budowy sieci sensorowej na potrzeby na przykład inteligentnego domu. Dzięki uzyskanemu dużemu zasięgowi możliwe byłoby zastosowanie pojedynczej bramy dostępowej łączącej sieć sensorową z Internetem na potrzeby mieszkania czy jednokondygnacyjnego domu.

Nietrudno sobie wyobrazić wyposażenie czujnika w hallotron do wykrywania otwartych okien, czujnika drgań do wykrywania czy maszyna na której zostanie umieszczony pracuje, czy też energooszczędnego czujnika ruchu. Dzięki temu, że czas między uśpieniem a transmisją danych jest bardzo krótki, możliwe jest też wykorzystanie tej konfiguracji sprzętowej jako urządzenie wejściowe dla użytkownika do sterowania innymi urządzeniami - mikrokontroler i transmisja danych mogłaby być wywoływana przyciskiem.

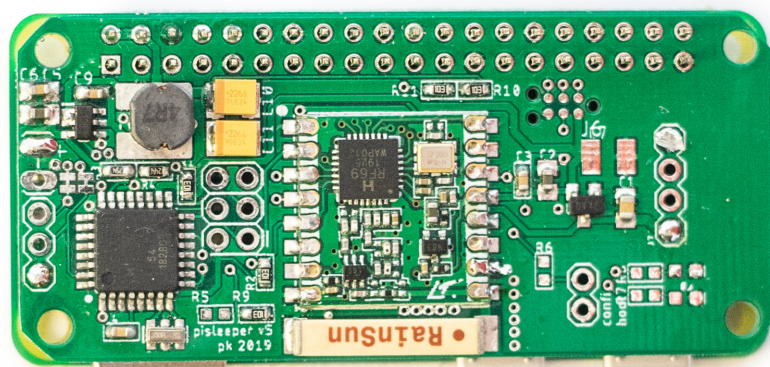
Ograniczenie konsumpcji energii w stanie uśpienia

Ponieważ urządzenie większość czasu przebywa w stanie głębokiego uśpienia, istnieje potencjał optymalizacji tego czasu poprzez zupełne odłączenie komponentów od zasilania (np. modułu radiowego) lub wykorzystanie na przykład innego mikrokontrolera o niższej konsumpcji energii. Jak wskazują podobne eksperymenty[24], istnieją mikrokontrolery o zużyciu energii w stanie uśpienia na poziomie ułamków mikroamperów. Umożliwiłoby to wykorzystanie bardziej energochłonnych komponentów pomiarowych utrzymując zbliżony czas pracy na baterii. W artykule "*Powering the Internet of Things*"[24] autorzy sugerują iż możliwe jest wykorzystanie dodatkowego zegara czasu rzeczywistego o bardzo niskiej konsumpcji energii (na poziomie 100nA) do zewnętrznego wybudzania innych układów, by wykonały pomiar i transmitowały dane.

Alternatywne zastosowanie wypracowanej konfiguracji sprzętowej - jako urządzenie wykonawcze

Autor wykonał również urządzenie wykonawcze w oparciu o opracowaną konfigurację sprzętową. Korzystając z mikrokontrolera AVR ATmega328p oraz radia RFM69HCW zaprojektowano urządzenie zarządzające zasilaniem mikrokomputera RaspberryPi Zero W, by umożliwić jego pracę jako urządzenie zasilane z akumulatorów.

Mikrokomputer ten nie posiada możliwości zarządzania zasilaniem, a w czasie aktywności konsumuje setki miliamperów przy napięciu 5V. Konieczne jest więc zewnętrzne zarządzanie jego zasilaniem. Opracowane rozwiązanie steruje przetwornicą stabilizującą napięcie zasilania mikrokomputera by zasilac go tylko wtedy gdy powinien on wykonać jakieś działania.



Rys. 4.14. Urządzenie wykonawcze - zarządzające zasilaniem mikrokomputera Raspberry Pi

Wnioski

Miniaturowe czujniki są podstawą funkcjonowania złożonych systemów Internetu Rzeczy, zarówno w obszarach czysto konsumenckich (Inteligentny Dom), jak i również przemysłowych (automatyka i przewidywanie awarii urządzeń oraz maszyn).

W ramach pracy opracowano przykładowy czujnik mierzący warunki atmosferyczne oraz natężenie światła, zasilany małą cylindryczną baterią litową, której wymiana konieczna będzie nie częściej niż raz na dziesięć lat założony, że czujnik będzie dokonywać pomiary i transmitować je co dziesięć minut.

W ostatecznej wersji urządzenia uzyskano zużycie energii na poziomie 0.08uAh na każdy pomiar trwający około 25 milisekund oraz 5.2uA w trybie głębokiego uśpienia pomiędzy pomiarami. Co w połączeniu z małą baterią litową w kształcie cylindra 14x25mm o pojemności nominalnej 1200mAh powinno pozwolić na dwadzieścia lat pracy urządzenia bez wymiany baterii.

Opracowanie zostało poprzedzone przeglądem dostępnych platform sprzętowych i komponentów, wykonane zostały odpowiednie prototypy oraz przeprowadzone pomiary zużycia energii podczas pomiaru oraz transmisji danych w różnych konfiguracjach. Wykonano odpowiednie optymalizacje, tak by z każdego prototypu uzyskać możliwie najlepsze wyniki.

Wykonany czujnik został poddany dokładniejszej ewaluacji, co umożliwiło przeprowadzenie kilku dodatkowych optymalizacji wydłużając teoretyczny czas pracy czujnika w podstawowej wersji do nawet ponad dwudziestu lat.

Na zakończenie wskazano kierunki możliwego dalszego rozwoju projektu, jak i również przedstawiono już opracowany przykład wykorzystania tej samej konfiguracji sprzętowej jako urządzenia wykonawcze do zarządzania zasilaniem mikrokomputera.

Bibliografia

- [1] P. Suresh i in. "A state of the art review on the Internet of Things (IoT) history, technology and fields of deployment". W: *2014 International Conference on Science Engineering and Management Research (ICSEMR)*. List. 2014.
- [2] *Statystyka ilości urządzeń Internetu Rzeczy do 2025 roku.*
- [3] S. P. Mohanty, U. Choppali i E. Kougianos. "Everything you wanted to know about smart cities: The Internet of things is the backbone". W: *IEEE Consumer Electronics Magazine* 5.3 (2016), s. 60–70.
- [4] Yen-Kuang Chen. "Challenges and opportunities of internet of things". W: (sty. 2012), s. 383–388. DOI: [10.1109/ASPDAC.2012.6164978](https://doi.org/10.1109/ASPDAC.2012.6164978).
- [5] Grant Svetlana. "3GPP Low Power Wide Area Technologies - GSMA White Paper". W: *GSMA* (2016).
- [6] *Nota katalogowa sensora BME280.*
https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BME280_DS001-12.pdf.
[Dostęp 05 Maj 2018].
- [7] *Nota katalogowa baterii Energizer LR6.*
<http://data.energizer.com/pdfs/alk-power-aa.pdf>. [Dostęp 10 Lipiec 2019].
- [8] *Nota katalogowa akumulatorów LiFePo4 APR18650.*
<http://www.batteryspace.com/prod-specs/6612.pdf>. [Dostęp 05 Maj 2018].
- [9] *Nota katalogowa układu MAX17055.*
<https://datasheets.maximintegrated.com/en/ds/MAX17055.pdf>. [Dostęp 05 Maj 2018].
- [10] *Test baterii INR18650.*
<https://lygte-info.dk/review/batteries2012/Samsung\%20INR18650-25R\%202500mAh\%20\%28Green\%29\%20UK.html>. [Dostęp 05 Maj 2018].
- [11] *Nota katalogowa baterii litowej Saft LS14500.*
. [Dostęp 05 Maj 2018].

- [12] *Instrukcja integracyjna dla modemu ublox SaraN211*.
https://www.u-blox.com/sites/default/files/SARA-N2_SysIntegrManual_%28UBX-17005143%29.pdf. [Dostęp 6 Maj 2018].
- [13] Jouni Tervonen Konstantin Mikhaylov. *Evaluation of Power Efficiency for Digital Serial Interfaces of Microcontrollers*. [Dostęp 25 Luty 2017].
- [14] *Biblioteka do obsługi modułu RFM69 autorstwa LowPowerLabs*.
<https://github.com/LowPowerLab/RFM69/>. Dostęp 03 Kwiecień 2019.
- [15] *Nota katalogowa sensora światła i odległości VCNL4010*.
- [16] *Nota aplikacyjna sensora VCNL4010*.
- [17] *Nota katalogowa baterii Saft 14250*.
<https://www.tme.eu/Document/01f99316651ed8ac4d3762ffd2728b6b/LS-14250.pdf>.
- [18] *Broszura baterie litowe Tadiran*.
<https://tadiranbatteries.de/pdf/Technical-Brochure-LTC-Batteries.pdf>. Dostęp 01 Wrzesień 2019.
- [19] *Broszura dotycząca wyboru baterii litowych*.
http://alpha-energy.ru/D/0000008967/CS_Saft_Li_B383703873_200303_en.pdf. Dostęp 01 Wrzesień 2019.
- [20] Hans W. Papenguth Kimberly D. Weaver David E. Weigand Donald B. Hardy i Michael R. Overstreet. *Capacity Loss Predictions for Active Lithium Primary Cells Using Accelerated Aging*.
- [21] *Modelling, Aging and Optimal Operation of Lithium-ion Batteries*.
- [22] Mahapatra Chinmaya Sheng Zhengguo Mirabbasi Shahriar Leung Victor C M Kamalinejad Pouya i Yong Liang Guan. "Wireless Energy Harvesting for Internet of Things". W: *IEEE Communications Magazine* 6.53 (2015).
- [23] *Nota katalogowa układu AEM30940 do odzyskiwania energii*.
https://www.fujitsu.com/uk/Images/DS_AEM30940_REV101.pdf. Dostęp 01 Wrzesień 2019.
- [24] Woo Suk Lee Arnab Raha Younghyun Kim Hrishikesh Jayakumar Kangwoo Lee i Vijay Raghunathan. "Powering the Internet of Things". W: *Proceedings of the International Symposium on Low Power Electronics and Design* (sierp. 2014).