

Politechnika Wrocławska

Wydział Elektroniki, Fotoniki i Mikrosystemów

KIERUNEK: Automatyka i Robotyka (AIR)

PRACA DYPLOMOWA INŻYNIERSKA

TYTUŁ PRACY:
Zastosowania technologii Internetu Rzeczy
w inteligentnych budynkach

AUTOR:
Krystian Matusiak 249460

PROMOTOR:
dr inż. Wojciech Dowski, K29W12ND02

Spis treści

1	Wstęp	3
1.1	Teza pracy	6
1.2	Zakres pracy	6
2	Internet Rzeczy	7
2.1	Transmisja danych	8
2.1.1	Modulacje analogowe	8
2.1.2	Modulacje cyfrowe	10
2.2	Topologie sieci	10
2.3	Protokoły komunikacyjne	11
2.4	LoRaWAN	12
3	Architektura i model fizyczny systemu	15
3.1	Zastosowane urządzenia	15
3.1.1	Moduły sterujące	15
3.1.2	Struktura instalacji w pomieszczeniach	16
3.1.3	Elektronika i układy sensoryczne	17
4	Realizacja oprogramowania	19
4.1	Implementacja programu na mikrokontrolerze STM	19
4.1.1	Systemy czasu rzeczywistego	20
4.1.2	Architektura urządzenia wykonawczego	22
4.1.3	Komunikacja z podmodułami	23
4.1.4	Regulator PID w lokalizatorze słonecznym	24
5	System zarządzający	27
5.1	Graficzny interfejs użytkownika	27
6	Algorytmy korekcyjne	31
6.1	Kod Hamminga	32
7	Testy użytkownika	33
8	Podsumowanie	37
	Bibilografia	38
	Załącznik A	41
	Załącznik B	43

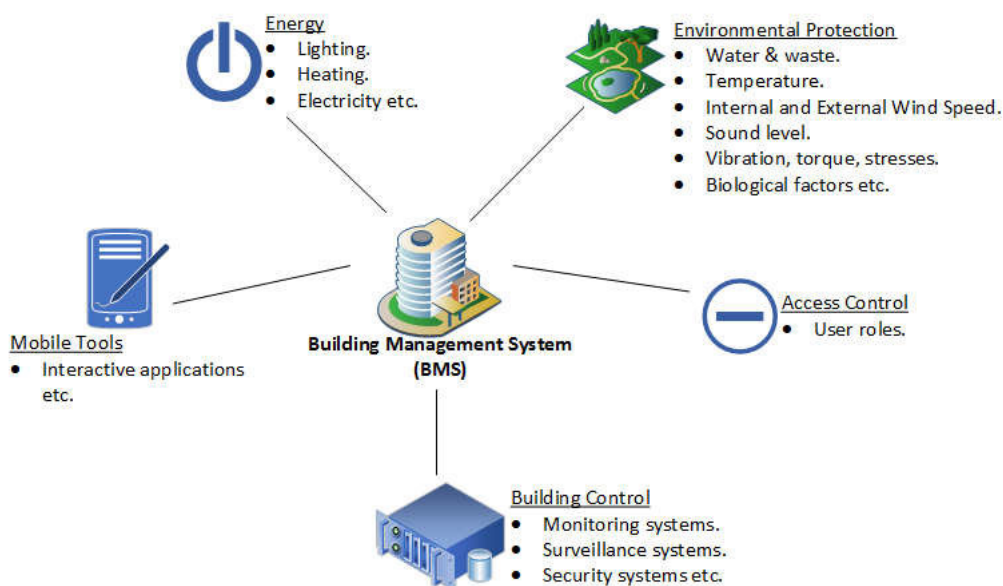
Rozdział 1

Wstęp

Podczas rozwoju przemysłu i globalnej industrializacji, przy jednoczesnym postępie technologicznym coraz to bardziej mnożyły się potrzeby optymalizacji środowisk pracy i modernizacji technik wytwarzania w zakładach przemysłowych. Początkowo starano się zapewnić komfort i dogodne warunki pracy.

Niesprzyjające warunki takie jak skrajne temperatury doprowadziły do opracowania systemów sterowania ogrzewaniem i klimatyzacją. Zwiększenie ilości miejsc pracy i rozbudowa fabryk poprzez nowsze i liczniejsze urządzenia zwiększyły prawdopodobieństwo wystąpień pożarów, co również przyczyniło się do poszerzenia listy systemów, usprawniających prace przemysłowe o system przeciwpożarowy. Co więcej, przez ryzyko włamań ludzie zdecydowali się na stworzenie systemów alarmowych.

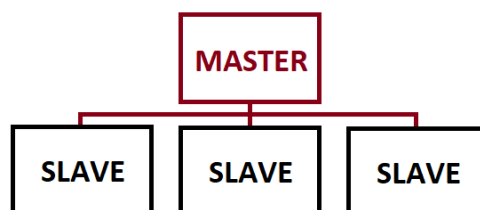
Na przestrzeni lat liczba dostępnych instalacji zwiększyła się na tyle, że powstały prace nad jednym systemem, spajającym pracę innych systemów; składającym je w całość. Był to system zarządzania budynkiem BMS (z ang. *Building Management System*) [28]. Na rysunku 1.1 można zaobserwować przykładowy schemat ww. systemu. Widać, że może on koordynować pracę systemów energetycznych, kontroli danych, bezpieczeństwa i ochrony środowiska. Oprócz integracji instalacji odpowiadał on za nadzorowanie i monitorowanie ich pracy. W przypadku awarii któregoś z urządzeń układ miał powiadomić o napotkanej usterce i o wszelkich nieprawidłowościach.



Rysunek 1.1 System zarządzania budynkiem (BMS) [26]

W momencie, gdy większość urządzeń elektronicznych stała się upowszechniona i ogólnodostępna, ludzie mogli sobie pozwolić na zastosowanie systemów inteligentnych budynków w prywatnych domach. To przeniknięcie z płaszczyzny przemysłowej do codziennego życia ludzi doprowadziło do jeszcze większego rozwoju ww. gałęzi technologii, ponieważ wymagania systemowe rosną w miarę potrzeb. Z roku na rok ludzie stawiali systemom coraz to nowsze zadania, co przyciągało inżynierów i zwiększało ich pole do działania. Jako że musieli oni sprostać wymaganiom użytkowników, rodziły się nowe pomysły na usprawnienia i funkcjonalności, takie jak sterowanie nagłośnieniem, oświetleniem, urządzeniami audio–video oraz systemami pogodowymi i „Home Assistant”, w konsekwencji czego umożliwiły wzbogacenie zastosowań automatyki budynkowej.

Obecne systemy inteligentnych budynków oparte są na modelu master-slave, którego uproszczony schemat zamieszczono na rysunku 1.2. Polega on na tym, że jedno urządzenie jest nadrzędne – zarządza pozostałymi urządzeniami. Ponadto służy jako medium, synchronizuje prace i zapewnia komunikację.



Rysunek 1.2 Model komunikacji master-slave

W zależności od potrzeb i złożoności budynku, liczba urządzeń podrzędnych może się różnić. Jednakże najczęstsze wykorzystywane instalacje to sterowanie ogrzewaniem, wentylacją i klimatyzacją *HVAC* (z ang. *Heating, Ventilation and Air Conditioning*), systemy sygnalizacji włamania i napadu *SSWN* (mogą być rozszerzane o symulowanie obecności domowników), sterowanie oświetleniem, roletami, żaluzjami oraz systemy przeciwpożarowe.

Wykorzystywanym sterownikiem do automatyzacji budynku w większości przypadków jest programowalny sterownik logiczny PLC [17]. Pracuje on w sposób cykliczny, tzn. wykonuje rozkazy kolejno po sobie. Są to powszechność zawdzięcza niezawodności oraz przystępności, ze względu na łatwy w nauce język drabinkowy. Z uwagi na fakt, że używane są od dawna (pierwszy sterownik PLC powstał w 1969 roku) na przestrzeni lat powstało wiele rozwiązań, które zostały ustandaryzowane, co zapewniło łatwość w zintegrowaniu różnego rodzaju urządzeń.

Komunikacje między poszczególnymi modułami można pogrupować na dwie kategorie: przewodowa oraz bezprzewodowa [13]. Stosowanie komunikacji przewodowej wiąże się z licznymi uniedogodnieniami. Aby nie narażać przewodów na warunki zewnętrzne najczęściej umieszcza się je w ścianach budynku co sprawia, że ewentualne modyfikacje bądź eliminacja awarii stają się trudne do wykonania. Dlatego też coraz częstsze okazują się zastosowania komunikacji radiowej. Nie mniej istnieją interfejsy przewodowe, których korzyści przeważają niedoskonałości i wady, co czyni je wykorzystywanymi do dzisiaj.

Przykładem takiego interfejsu jest standard Ethernet. Należy on do bardzo szybkich i niezawodnych sposobów komunikacji. Dodatkowo charakteryzuje się niską podatnością na zakłócenia. Z tego tytułu jest najpopularniejszym standardem w sieciach lokalnych.

Innym używanym interfejsem jest Profibus. Wykorzystuje technikę przekazywania prawa dostępu do sieci między stacjami nadrzędnymi typu master. Tworzy to topologię pierścienia. Stacje podrzędne nie mają aktywnego dostępu do medium. Oznacza to, że nie

mogą wysyłać zapytań. Zamiast tego mogą odpowiadać tylko i wyłącznie urządzeniu nadrzędnemu.

Oprócz wyżej wymienionych interfejsów stosuje się także komunikację elektroenergetyczną siecią rozdzielczą – PLC (z ang. *Power line communication*). Wykorzystuje sieć elektroenergetyczną budynku jako medium transmisyjne. Z napięciem zasilającym z sieci przesyła się równolegle sygnał z danym o częstotliwości znacznie wyższej niż panujące w sieci. Korzyści, jakie płyną z użycia takiego rozwiązania jest oszczędność, ze względu na wykorzystanie istniejącej już instalacji elektrycznej.

Podobnym do wyżej wymienionego interfejsu jest HomePNA. Różnica jednak polega na tym, że wykorzystana została infrastruktura okablowania telefonicznego i/lub przewodu koncentrycznego.

W pewnym momencie rozwoju techniki ludzie zaczęli zastanawiać się nad integracją urządzeń nie tylko lokalnie, ale żeby była możliwość połączenia się z danym modulem na większych odległościach. Niestety komunikacja przewodowa na dalsze odległości traci swój urok ze względu na niską energooszczędność (dłuższe przewody wiążą się z dużymi stratami energii w postaci ciepła), nieekonomiczność (większe koszty przez materiały wykonawcze oraz roboczo-godziny elektromonterów) ale przede wszystkim kluczową wadą jest podatność na zakłócenia. Dlatego też inżynierzy tych czasów (przełom XX i XXI wieku) skierowali swoją uwagę i zainwestowali swój czas w rozwój technik radiowych i telekomunikacyjnych.

Był to okres rozkwitu transmisji danych radiowych. Powstały nowe standardy i pomysły na protokoły, opierające się na paśmie ISM [19] (z ang. *industrial, scientific, medical*). Oznacza to, że musiały one spełniać konkretne wymagania, takie jak nadawanie informacji z częstotliwością mieszczącą się w danym zakresie. Początkowo były one zarezerwowane głównie do celów przemysłowych, naukowych i medycznych, jednak z czasem rozszerzono zakres zastosowań.

Wśród nowych protokołów pojawiły się takie nazwy jak Bluetooth, Zigbee czy WiFi. Były one brakującym ogniwem w rozwoju inteligentnych domów. Sprzęty elektroniczne, takie jak klawiatura, komputer, mysz skomunikowano siecią Bluetooth, ze względu na krótki zasięg działania lecz zadowalającą przepustowość. Są to urządzenia, które nie mogą komunikować się na dużej odległości między sobą, dlatego też można było pozwolić na większą przepustowość i energooszczędność kosztem mniejszego zasięgu działania. Przy bardziej odległych modułach (sterowanie bramą, roletami, ogrzewaniem) zaczęto stosować protokół Z-Wave, ze względu na szerszy promień nadawania i odbioru. Do sprzętów gospodarstwa domowego, takich jak lodówki czy pralki stosuje się również moduły, które łączą możliwości komunikacji przez Bluetooth oraz Zigbee. Koncepcję łączenia ze sobą urządzeń w sposób bezprzewodowy określono mianem Internetu rzeczy.

Nie ustalono jednego, stałego sposobu na skoordynowanie prac systemów. Dobór odpowiedniej technologii ma być w pełni zależny od wymagań użytkownika i cech otoczenia, o czym świadczą powyższe przykłady.

Atutem korzystania z technologii Internetu rzeczy w inteligentnych budynkach [31] jest wysoka energooszczędność. Moduły odpowiadające za komunikację bezprzewodową mają niewielki pobór prądu, co idealnie sprawdza się w instalacjach, które potrzebują ciągłego zasilania. Mimo zadowalającej ekonomiczności, prace nad doskonaleniem kryterium wciąż są kontynuowane [2]. Do optymalizacji kosztów poboru mocy zaczęto stosować nawet rozwiązania uczenia maszynowego [6].

Przykładową rodziną wykorzystywanych algorytmów jest uczenie przez wzmacnianie (z ang. *Reinforcement Learning*). Służyły głównie do wyboru ścieżki przesyłu oraz doboru opóźnień w technice usypiania zadań. Polegały one na wprowadzaniu do środowiska

(w tym przypadku do sieci modułów nadawczo-odbiorczych) agentów, monitorujących pracę węzłów. Informacją, na której bazowały był sygnał wzmocnienia, który odpowiednio otrzymywał dużą wartość (tzw. nagrodę), podejmując dobrą decyzję bądź niską wartość przy błędnej decyzji. Nagrodą w przypadku komunikacji bezprzewodowej jest podjęcie energooszczędnej operacji. Zebranyymi informacjami posługują się w dalszym kroku algorytmy heurystyczne, aby opracować sposób na utrzymanie minimalnego możliwego poboru mocy.

Jak widać technologia wciąż się rozwija. Często jedna gałąź jest zależna od drugiej, tak jak to było można zauważyć w przypadku stosowania komunikacji radiowej przy inteligentnych budynkach. Optymalizację można dostrzec nawet w sytuacjach, gdy ulepszany wskaźnik już jest na zadowalającym poziomie, czego przykładem jest kryterium energooszczędności. Ciągłe dążenie do doskonałości jest wciąż obecne.

1.1 Teza pracy

Celem niniejszego opracowania jest wykonanie symulacji oraz wizualizacji pracy inteligentnego budynku i jego podrzędnych modułów przy pomocy wybranego protokołu komunikacyjnego.

1.2 Zakres pracy

Zakres pracy obejmuje zaprojektowanie układów sterujących, pomiarowych oraz symulujących pracę konkretnych modułów instalacji domowych (takich jak detektor zalania mieszkania), zbudowanie ich fizycznej reprezentacji oraz sterowanie nimi przez mikrokontroler firmy STMicroelectronics. Oprócz zaimplementowania kodu dla sterowników należało oprogramować główną jednostkę sterującą, pełniącą rolę BMS, wykorzystując platformę komputerową Raspberry Pi. Dodatkowo należało zrealizować aplikację dostępową w postaci graficznego interfejsu użytkownika, odpowiedzialnego za wizualizację pracy wszystkich modułów. Poszczególne węzły sieci komunikują się ze sobą poprzez sieć bezprzewodową wykorzystującą do tego celu opracowany protokół. Dla dodatkowego usprawnienia zastosowano algorytm korekcji danych.

Oprogramowanie przeznaczone na mikrokontrolery STM32 przygotowano przy pomocy środowiska projektowego STM32CubeIDE. Aby umożliwić równoległe wykonywanie określonych zadań przez ww. mikrokontrolery wykorzystano system czasu rzeczywistego FreeRTOS [23].

Do realizacji aplikacji interfejsu graficznego wykorzystano framework Qt. Opracowanie protokołu komunikacyjnego obejmuje dobór odpowiedniego sposobu komunikacji bezprzewodowej oraz przygotowanie ramki danych, która jest przesyłana drogą radiową. Korekcję danych zrealizowano w oparciu o kodowanie Hamminga.

Rozdział 2

Internet Rzeczy

Określenie Internetu Rzeczy jako pierwszy podał w 1999 roku brytyjski przedsiębiorca Kevin Ashton, definiując je jako cyfrową sieć połączonych ze sobą przedmiotów, przy jednoczesnym połączeniu ze światem fizycznym [15]. Ponadto nie jest możliwe doświadczenie bezpośredniego rozwoju koncepcji Internetu Rzeczy. Intensyfikacja tej technologii objawia się w rosnącej ilości urządzeń, które w swoich funkcjonalnościach mają zawartą komunikację między sobą.

Synchronizacja urządzeń Internetu Rzeczy z przechowywaniem danych i Big Data, oparta na energooszczędności oraz tanim oprogramowaniu i sprzęcie sprzyja interoperacyjności otwartych standardów. Obecnie stosowanie zintegrowanych urządzeń i systemów można zauważyć na wielu płaszczyznach w różnych sferach życia.

Jako przykład można podać logistykę transportu. Dokonane przez systemy rozproszone usprawnienia dotyczą nie tylko przepływu towarów ale i globalnego pozycjonowania oraz automatycznej identyfikacji tych towarów. Oczekuje się, że powiększone zostaną zyski z łańcuchów dostaw poprzez inteligentne zarządzanie i ruch produktów handlowych dzięki ciągłej koordynacji i synchronizacji danych oraz śledzenia czasu i drogi obiektów.

Kolejnym przykładem zastosowań są inteligentne budynki. Przede wszystkim zyskają pewnego rodzaju „świadomość”, tzn. będą gromadzić informacje o zasobach (ile prądu bądź wody zużyto) oraz o stanie bezpieczeństwa (czy nie dokonano włamania bądź czy nie wzniesiono pożaru). Ponadto będą w stanie podnieść poziom komfortu stosując różnego rodzaju instalacje.

Idea zautomatyzowanego budynku została nawet rozszerzona do pojęcia inteligentnego miasta. O ile w pełni zmodernizowanych miast na świecie nie ma, o tyle powstają osiedla, których infrastruktura telekomunikacji przewyższa nie jedną miejscowość. Dodatkowo coraz częściej wprowadzane są elektryczne pojazdy miejskie takie jak hulajnogi, rowery czy nawet auta. Cały system zarządzający wypożyczaniem i lokalizacją środków transportu oparty jest na ideologii Internetu Rzeczy.

Idąc dalej w stronę usprawnień i technik optymalizacji, stosując systemy rozproszone można udoskonalić techniki wytwarzania i w pełni zautomatyzować procesy przemysłowe w inteligentnych fabrykach. Podczas, gdy we wczesnych etapach rozwoju techniki tworzone pojedyncze, bezobsługowe stanowiska tak obecnie, dzięki zintegrowaniu wielu urządzeń przemysłowych bezprzewodowo, można opracowywać systemy, które będą odpowiedzialne za wytworzenie całego przedmiotu od początku do końca. Ponadto w globalnym łańcuchu dostaw firmy są w stanie śledzić produkty dzięki znacznikom identyfikacji radiowej RFID, co skutkuje zmniejszeniem kosztów operacyjnych. Dodatkowym atutem jest ułatwiona konserwacja maszyn przez połączenie ze sobą bezprzewodowo czujników, które na bieżąco monitorują stan i wydajność wyposażenia fabrycznego w czasie rzeczywistym.

Ogólnie można ująć, że Internet Rzeczy zapewni automatyczne procedury, które drastycznie zmniejszą liczbę potrzebnych osób zatrudnionych. Pracownicy będą zastąpieni przez skanery kodów kreskowych, czytniki, czujniki a przede wszystkim przez roboty.

Powyższe przykłady pokazują jak powstanie i rozwój internetu rzeczy wpłynęły na wiele płaszczyzn. Idea zintegrowanych i skoordynowanych ze sobą urządzeń błyskawicznie przeniknęła do różnych sfer życia i znalazła swoje globalne zastosowanie.

Zintegrowanie tak dużej ilości urządzeń przyciąga osoby, które chciałyby mieć dostęp do tych instalacji, dlatego też starano się wprowadzić różnego rodzaju zabezpieczenia. Jednym z najlepiej sprawdzonych sposobów ochrony danych okazał się Blockchain [5]. Jest to współdzielona cyfrowa księga, zaprojektowana tak, aby każda modyfikacja była wykrywalna. Przypomina bazę danych, która jest współużytkowana, replikowana i zsynchronizowana między członkami zdecentralizowanej sieci. Technologia ta składała się z łańcucha sekwencyjnie połączonych bloków, gdzie każdy blok zawierał listę transakcji. Struktura bloku zaś składa się z sekcji nagłówka i treści. W nagłówku znajdowały się wartości „zahaszowane” poprzedniego oraz aktualnego bloku, natomiast w treści zamieszczone były informacje o transakcjach. Dzięki technice szyfrowania technologia Blockchain staje się coraz częściej stosowanym zabezpieczeniem w urządzeniach Internetu Rzeczy.

2.1 Transmisja danych

Internet Rzeczy opiera swoją komunikację na protokołach komunikacji zarówno przewodowej jak i bezprzewodowej. Przeważającą grupą zaś są standardy przesyłu danych drogą radiową. Sama transmisja radiowa polega na wygenerowaniu fali nośnej w postaci sygnału okresowego (najczęściej jest to przebieg sinusoidalny) o konkretnej częstotliwości. Aby uwzględnić dodatkowo w tym sygnale daną informację, jaką zamierza się przesłać klucze okazują się być modulacje [4]. Najprościej mówiąc są to zmiany charakterystyk fali nośnej na podstawie właściwości sygnału informacyjnego (modulującego). Ów przekształcane charakterystyki to następujące wielkości: amplituda, częstotliwość oraz faza.

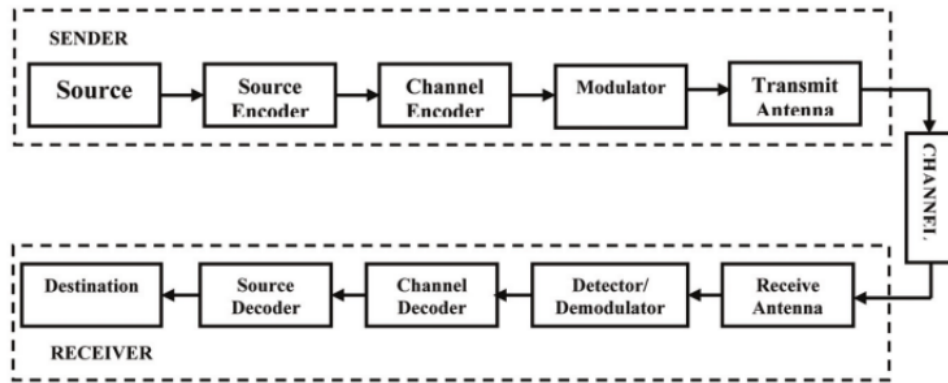
Modulacje można podzielić na dwie kategorie: cyfrową i analogową. Różnią się one typem sygnału, które mają być przesyłane. W przypadku modulacji analogowych informacje muszą być w postaci analogowej, np. dźwięk, nagranie wideo. Modulację cyfrową natomiast można zauważyć przy przesyłaniu danych w formie binarnej, np. obrazy lub pliki tekstowe.

Każdą kategorię można pogrupować na kolejny zestaw modulacji [11]. Jest to efekt tego, że większość przypadków transmisji danych wiąże się z nieuniknioną obecnością szumu i zniekształceń, na które różne modulacje są w mniejszym lub większym stopniu wrażliwe. Dlatego też opracowywano rozmaite przekształcenia sygnałów w odpowiedzi na konkretne zakłócenia.

Po udanej transmisji, po stronie odbiorcy dokonuje się odtworzenia oryginalnego sygnału przy pomocy procesu odwrotnego, określanego demodulacją. Ze schematu przykładowego przesyłu danych na rysunku 2.1 można zauważyć, że kolejne kroki nadawania wiadomości odpowiadają odwrotnej kolejności kroków odbioru.

2.1.1 Modulacje analogowe

Tę rodzinę modulacji stosuje się w przypadkach, kiedy sygnałem informacyjnym jest sygnał ciągły tzn. taki, którego wartość należy do przedziału ciągłego oraz można ją określić w dowolnej chwili czasu. Najczęściej wykonywanymi przekształceniami są:



Rysunek 2.1 Schemat systemu komunikacji [4]

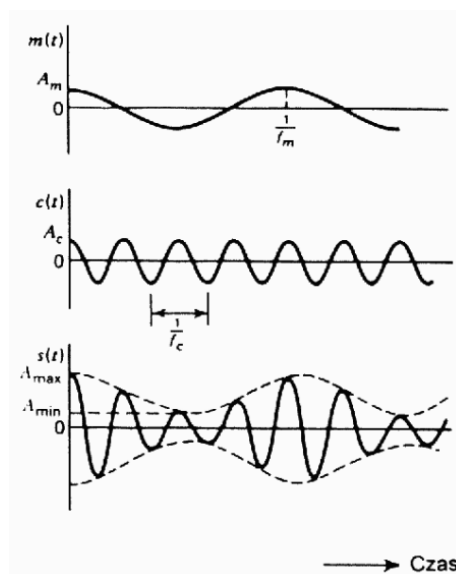
- modulacja amplitudy AM (z ang. *Amplitude Modulation*) – amplituda fali nośnej zmienia się w takt sygnału modulującego. Zakładając następującą postać fali nośnej

$$c(t) = A_c \cos(2\pi f_c t + \phi_c), \quad (2.1)$$

gdzie: A_c – amplituda, f_c – częstotliwość, ϕ_c – faza oraz przyjmując, że $m(t)$ oznacza przebieg sygnału modulującego fala zmodulowana przyjmuje postać

$$s(t) = A_c[1 + k_a m(t)] \cos(2\pi f_c t + \phi_c), \quad (2.2)$$

gdzie k_a oznacza czułość amplitudową modulatora odpowiedzialnego za generację sygnału zmodulowanego. Wykresy fal omówionej modulacji można zaobserwować na rysunku 2.2.



Rysunek 2.2 Ilustracja procesu modulacji amplitudy pojedynczym tonem prostym: a) fala modulująca, b) fala nośna, c) fala zmodulowana [11]

- modulacja częstotliwości FM (z ang. *Frequency Modulation*) – częstotliwość fali nośnej zmienia się pod wpływem sygnału informacyjnego. Zakładając, że sygnałem $m(t)$ modulujemy sygnał (2.1) uzyskujemy

$$s(t) = A_c \cos\left(2\pi f_c t + 2\pi k_f \int_0^t m(t) dt\right), \quad (2.3)$$

gdzie k_f wyraża czułość częstotliwościową modulatora.

- modulacja fazy PM (z ang. *Phase Modulation*) – przesunięcie fazowe sygnału jest zależne od sygnału modulującego. Dla przykładu fali nośnej oznaczonej wzorem (2.1) odpowiednio fala zmodulowana sygnałem informacyjnym $m(t)$ wyrażana jest jako

$$s(t) = A_c \cos(2\pi f_c t + k_p m(t)), \quad (2.4)$$

gdzie k_p jest czułością fazową.

Zestawiając powyższe przypadki przekształceń można zauważyć, iż modulacja częstotliwości różni się tym, że sygnał zmodulowany jest zdefiniowany równaniem nieliniowym, co utrudnia analizę widmową tego sygnału.

2.1.2 Modulacje cyfrowe

W sytuacji, gdy informacja jaką chcemy przesłać staje się konkretną wartością (reprezentowaną przez ciąg bitów) rozpatruje się modulacje cyfrowe. Sam proces przekształceń fali nośnej jest zbliżony do przypadku przesyłu fali ciągłej. Różnica jednak polega na tym, że nie uwzględniane są stany pośrednie w wartościach sygnału wynikowego (wielkości fizyczne przyjmują tylko dwie wartości odpowiadające sygnałowi binarnemu). Powszechnie używa się kluczowania amplitudy, fazy oraz częstotliwości.

W sytuacji, gdy dokonujemy kluczowania amplitudy fala zmodulowana przyjmuje postać

$$s(t) = \begin{cases} A_0 \cos(\omega_0 t) & \text{gdy } m_t = 1 \\ A_1 \cos(\omega_0 t) & \text{gdy } m_t = 0 \end{cases}, \quad (2.5)$$

gdzie A_0, A_1 – amplitudy, ω_0 – pulsacja fali nośnej. Dla tego konkretnego rodzaju modulacji można powiedzieć, że sygnał został poddany kwantyzacji.

2.2 Topologie sieci

Oprócz doboru odpowiedniej modulacji do przesyłanych danych istotną kwestią, jaką należy wziąć pod uwagę jest sposób komunikacji urządzeń, a dokładniej ich konfiguracja. W zależności od tego jak bardzo moduły radiowe mogą być podatne na zakłócenia oraz jak często generują błędy przy komunikacji należy dobrać odpowiednią konfigurację tych urządzeń. Właściwie dobrana budowa sieci może zoptymalizować przepływ danych na wielu płaszczyznach.

W miarę rozwoju telekomunikacji, ale także i innych gałęzi technologii powstawały coraz to nowsze pomysły na struktury połączeń między różnymi urządzeniami [25]. Wiodące topologie wśród bezprzewodowych sieci komunikacyjnych znalazła się topologia gwiazdy, pierścienia oraz siatki.

Topologia gwiazdy polega na tym, że wszystkie węzły są połączone z jednym węzłem komunikacyjnym, gdzie wszystkie węzły poza centralnym nie mogą się bezpośrednio

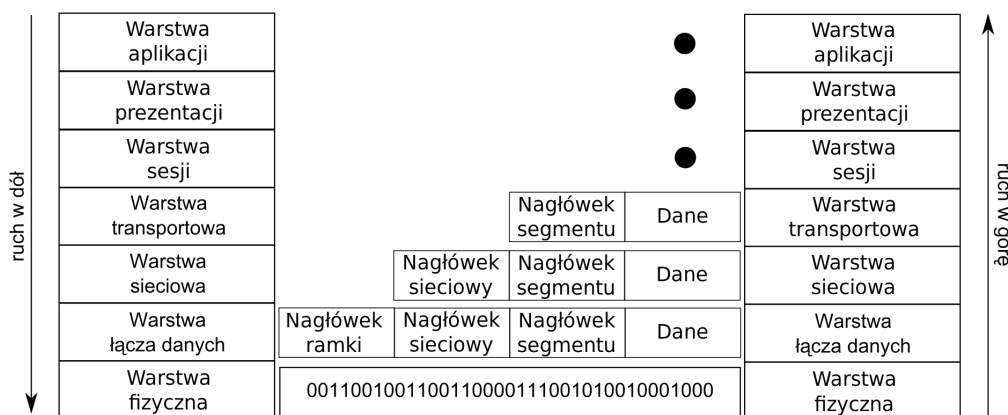
między sobą komunikować. Koordynacja wszystkich modułów zależy od jednostki nadrzędnej. Zaletą tej sieci jest możliwość łatwej rozbudowy, dodania nowego elementu oraz namierzenia potencjalnej awarii. Topologia pierścienia zaś zakłada, że każdy węzeł ma dokładnie dwóch sąsiadów. Komunikacja wygląda w taki sposób, że wiadomość jest wysyłany w jedną stronę, zaś odpowiedź przychodzi z przeciwnego kierunku. Główną wadą stosowania tej topologii jest podatność na awarie. Wystarczy jeden niedziałający poprawnie moduł aby cała transmisja nie powiodła się. Topologia siatki zaś opiera się na bezpośrednim połączeniu jak największej ilości węzłów. Daje to wysoką niezawodność oraz brak kolizyjności. Implementacja ta bywa jednak kosztowna i często skomplikowana.

Na podstawie wyżej wymienionych topologii można zaobserwować wiele korzyści, ale też i niedogodności. W zależności od tworzonego systemu dobiera się odpowiednią architekturę komunikacji, maksymalizując parametry, na których w danej sytuacji zależy najbardziej. Istnieją także przypadki hybrydowe, tzn. struktury zawierające cechy różnych architektur. Są to połączenia wyżej wymienionych topologii.

Sieci można również pogrupować ze względu na ich wielkość. Przykładem mogą być sieci lokalne LAN (z ang. *Local Area Network*), sieci miejskie MAN (z ang. *Metropolitan Area Network*) oraz sieci rozległe WAN (z ang. *Wide Area Network*).

2.3 Protokoły komunikacyjne

Wiele kwestii, takich jak sposób transmisji czy wybór struktury komunikacji zostało już ujętych w gotowych standardach, powstałych na przestrzeni lat. W zależności od potrzeb uformowały się protokoły komunikacyjne [30], czyli zbiory zasad i standardów, które wyznaczają wcześniej wspomniane aspekty. Opracowano również sposób opisu systemów komunikacyjnych – model ISO/OSI (z ang. *Open Systems Interconnection*). Składa się z siedmiu warstw komunikacji (rys. 2.3).



Rysunek 2.3 Model ISO/OSI [1]

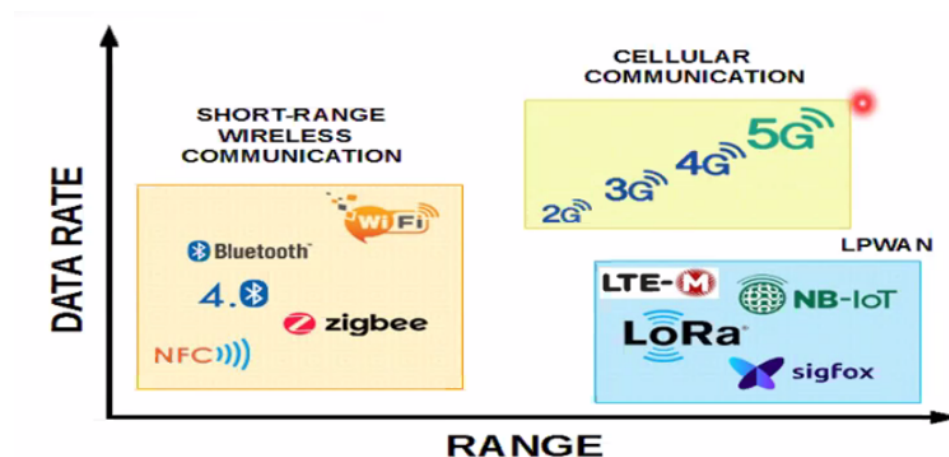
Protokół komunikacji bezprzewodowej najczęściej implementowany jest w warstwie łącza danych (modulacje zaś opisane są w warstwie fizycznej). Protokół odpowiada za kwestie formowania ramki, transmisji radiowej, nadzorowania jakości przesyłu oraz wykrywania ewentualnych błędów. Wśród najczęściej wykorzystywanych standardów komunikacji bezprzewodowej w produktach Internetu Rzeczy można znaleźć [20]:

- WiFi – najbardziej popularny typ komunikacji wśród dostępnych rozwiązań. Wiele urządzeń dostępnych na rynku może łączyć się ze smartfonami za pomocą technologii WiFi. Jednakże powszechność tego sposobu komunikacji może nieść zarówno

korzyści jak i zagrożenia. Każde kolejne urządzenie dołączone do sieci może powodować coraz to większe zakłócenia z konkretnego pasma częstotliwości (2,4 GHz oraz 5GHz),

- Zigbee – najczęściej działa w topologii siatki (z ang. *mesh*), tj. wykorzystuje urządzenia do przekazywania sygnałów dalszym urządzeniom, rozszerzając sieć. Węzły łączą się bezpośrednio oraz niehierarchicznie. Może być wbudowany w gotowe systemy (np. termostaty i zamki do drzwi). Jego przestrzeń operacyjna dochodzi do 100 metrów. Szeroki zakres pasma powoduje, że nie ma całkowitej pewności udanego skomunikowania ze sobą dwóch urządzeń,
- Z-Wave – podobnie jak Zigbee, Z-Wave pracuje w sieci typu siatka. Odróżnia się niższą przepustowością danych, jednakże wymaga znacznie mniej energii od komunikacji poprzez protokół Zigbee,
- Bluetooth – jego główną zaletą jest bezpieczeństwo oraz niski pobór mocy, jednakże posiada bardzo krótki zasięg działania. Technologia ta może obsługiwać zarówno strumienie asynchroniczne jak i synchroniczne,
- LoRaWAN – cechuje się bardzo niskim poborem mocy oraz szerokim zasięgiem. Coraz więcej pojawia się jako rozwiązanie w zastosowaniach inteligentnego miast. Najczęściej pracuje w topologii gwiazdy.

Różnorodność powyższych przykładów świadczy o tym, że w różnych sytuacjach mogą się sprawdzać inne protokoły. Analizując takie wymagania systemowe jak energooszczędność, zasięg, przepustowość czy awaryjność można dobrać najodpowiedniejszy sposób komunikacji. Na rysunku 2.4 został przedstawiony poglądowy wykres z wybranymi technologiami komunikacji radiowej, osadzonymi w zależności od przepustowości danych oraz zakresu operacyjnego.



Rysunek 2.4 Zakresy pracy podanych protokołów [7]

2.4 LoRaWAN

Do jednej z głównych technologii sieci rozległych o niskim poborze mocy LPWAN (z ang. *Low Power Wide Area Network*) należy LoRa. Głównym powołaniem tego protokołu było utrzymanie połączenia między urządzeniami Internetu Rzeczy na dużych

odległościach. Urządzenia komunikujące się tym protokołem najczęściej konfiguruje się w topologii gwiazdy.

Stosuje on technikę modulacji widma rozproszonego [14], dzięki czemu cechuje się niską podatnością na zniekształcenia. Technologia ta opiera się na wykorzystaniu przepustowości przekraczającej minimalną przepustowość niezbędną do wysłania sekwencji danych. Wąskie pasma sygnałów są łatwe do wykrycia, dlatego też zastosowanie sygnału szerokopasmowego uczyniło transmisję ciężko wykrywalną. Takie działanie znalazło zastosowanie w komunikacji wojskowej.

Oprócz zastosowań militarnych technologia LoRaWAN znalazło również zastosowanie w medycynie, inteligentnych miastach, przemyśle oraz w rolnictwie. Możliwości dalekiego zasięgu zapewniają protokołowi znacznie szerszy wachlarz zastosowań niż w przypadku takich protokołów jak Bluetooth czy Zigbee. Duża rozległość wiąże się niestety z niższą przepustowością (maksymalna szybkość transmisji dochodzi do 50 kb/s). Na taką niedogodność mogą sobie pozwolić systemy, które nie uwzględniają przesyłu obszernych pakietów danych. Wyżej wymienione zastosowania spełniają te kryteria, gdyż większość przesyłanych informacji w tych systemach oparte są na odczytach z czujników, detektorów czy innych przyrządów pomiarowych.

Rozdział 3

Architektura i model fizyczny systemu

Po przestudiowaniu licznej literatury, związanej z wyżej wymienionymi aspektami i analizie obecnych rozwiązań postanowiono wykonać model domu jednorodzinnego oraz zbadać efektywność zastosowania koncepcji Internetu Rzeczy. Makieta budynku została wyposażona została w najczęściej spotykane instalacje automatyki budynkowej, których praca jest nadzorowana przez jedno urządzenie nadrzędne. Narzuconym wymaganiem na główny moduł kontrolny jest wysoka wydajność sprzętowa, aby zminimalizować prawdopodobieństwo wystąpienia nieprzewidzianych okoliczności oraz aby umożliwić wykonanie bardziej wymagających oraz kosztownych obliczeniowo zadań.

Do skoordynowania ze sobą oddzielnych modułów wykorzystano protokół LoRaWAN, ze względu na jego daleki zasięg oraz niskie wymagania systemu, dotyczące ilości przesyłanych danych (przeważają dane pomiarowe). Zdecydowano się na użycie konfiguracji gwiazdy do integracji modułów, ze względu na fakt, iż wyżej wymieniony protokół najczęściej odnajduje się w strukturach o topologii gwiazdy. Ponadto hierarchia urządzeń oparta została na modelu master-slave.

Do przetestowania projektu postanowiono, aby jednostka główna komunikowała się z dwoma modułami podrzędnymi. Każdy z nich odpowiada za obsługę instalacji w trzech pokojach. Pierwszy zarządza systemami w salonie, łazience oraz sypialni, zaś drugi moduł odpowiedzialny jest za obsługę kuchni, holu oraz garażu.

3.1 Zastosowane urządzenia

Aby zasymulować prace poszczególnych instalacji inteligentnego budynku wykorzystano liczne moduły oraz układy elektroniczne, które imitują czujniki bądź nimi faktycznie są. Rzeczywistych układów użyto do pomiaru ciśnienia oraz temperatury, natomiast pozostałe czujniki są symulowane zostały w większości przez przełączniki monostabilne. W celu ułatwienia późniejszego testowania detekcji zalania mieszkania zastosowano potencjometr, którego manipulacja powodowała zmianę poziomu wilgotności.

3.1.1 Moduły sterujące

Wcześniej wspomnianej jednostce nadrzędnej narzucono odpowiednią wydajność sprzętową. Spowodowane jest to zaplanowaną implementacją graficznego interfejsu użytkownika. Dlatego też urządzenie musi sprostać wymaganiom podtrzymania aplikacji graficznej w stałym, zadowalającym stanie (bez opóźnień i błędów w synchronizacji z rzeczywistym modelem).

Potencjalnymi platformami, które były brane pod uwagę do doboru jednostki głównej były BeagleBone Black Rev. C [29] oraz Raspberry Pi 3 B+ [22]. Są to przykłady komputerów jednopłytkowych SBC (z ang. *Single-Board Computer*), tzn. że na ich pokładzie między różnymi peryferiami i pamięcią znajduje się też procesor z systemem operacyjnym. Całość mieści się na pojedynczej płytce drukowanej.

Pomimo posiadania przez platformę Beaglebone przetwornika analogowo-cyfrowego oraz akceleratora grafiki 3D ostatecznie zdecydowano, że Raspberry Pi zostało użyte do realizacji jednostki centralnej. Posiada dwukrotnie większą pamięć operacyjną RAM – 1GB. Ponadto taktowanie procesora wynosi 1,4GHz (w przypadku pierwszego modułu wynosi 1GHz) co może znacznie wpłynąć na usprawnienie pracy całego systemu.

Moduły podrzędne zrealizowane zostały oparte o mikrokontrolery firmy STMicroelectronics. Podjęto taką decyzję ze względu na ich dostępność, posiadanie licznych peryferii oraz łatwość w programowaniu. Dobór odpowiednich wersji mikrokontrolerów był oparty na funkcjonalnościach, za które rzeczono mikrokontrolery miały być odpowiedzialne. Przede wszystkim brana pod uwagę była ilość cyfrowych wyprowadzeń wejściowo-wyjściowych GPIO. Musiała ona być wystarczająca, aby obsłużyć wszystkie zaplanowane elementy elektroniczne. Ponadto mikrokontroler musiał być wyposażony w odpowiednie interfejsy transmisji szeregowej, aby były w stanie komunikować się z gotowymi modułami, takimi jak czujnik ciśnienia bądź układ nadawczo-odbiorczy protokołu LoRa. Pamięć operacyjna również brana była pod uwagę ze względu na liczebność elementów, jakie musiały zostać obsłużone.

Ostatecznie wybrano mikrokontroler STM32F303K8 umieszczony na płytce ewaluacyjnej NUCLEO do integracji systemów w salonie, łazience oraz sypialni. Moduły w pozostałych pomieszczeniach skoordynowane zostały przy pomocy płytki deweloperskiej STM32F411E DISCOVERY, z uwagi na większą ilość wyprowadzeń i peryferiów, które idealnie nadają się do bardziej wymagających instalacji i operacji, takich jak jednoczesna obsługa kilku interfejsów szeregowych.

3.1.2 Struktura instalacji w pomieszczeniach

Podstawowym wyposażeniem każdego pokoju jest czujnik ruchu, wychwytyjący obecność osób w pomieszczeniu. Dodatkowo, wszystkie pomieszczenia zawierają najczęściej spotykaną i typową dla danej części budynku instalację. Dla łazienki jest to detekcja zalania mieszkania. Działa ona w oparciu o czujnik wilgotności. Gdy osiągnie on odpowiedni próg wartości wyśle sygnał do sterownika, aby wyłączyć dopływ prądu w łazience. Takie zastosowanie zapewnia bezpieczeństwo dla pobliskich urządzeń, ale przede wszystkim dla domowników.

Z uwagi na fakt, że w garażu znajduje się samochód, w coraz większym stopniu zaczęto stosować systemy alarmowe, sygnalizujące kradzież pojazdu. Dlatego też w modelu uwzględniono tę funkcjonalność.

Sypialnia wykorzystuje zaawansowany system budzika. Za pośrednictwem aplikacji graficznej w jednostce centralnej ustawiana jest czas załączenia budzika. Następnie urządzenie główne wysyła informacje o aktualnym czasie oraz dniu i godzinie, kiedy ma zostać załączona pobudka. Moduł podrzędny odbiera informacje i odpowiednio je przetwarza – aktywuje budzik we wcześniej ustalonym czasie.

W podobny sposób działa układ wentylacji w salonie. Na podstawie dostarczanych przez moduł centralny informacji o temperaturze, układ podejmuje decyzje o włączeniu wiatraka sufitowego.

W kuchni pierwszorzędną kwestią, o jaką trzeba zadbać jest bezpieczeństwo w przypadku pożaru. Zainstalowany czujnik czadu po wykryciu tlenku węgla aktywuje alarm w postaci sygnału dźwiękowego oraz czerwonego światła.

Hol, oprócz czujnika obecności i instalacji sprawdzającej zamknięcie budynku, wyposażony jest w zestaw czujników – temperatury oraz ciśnienia. Dodatkowo podłączono zegar czasu rzeczywistego, dostarczający dane dotyczące godziny oraz daty. Informacje z tych czujników przesyłane są do jednostki nadrzędnej, która następnie dzieli się zdobytymi informacjami z inną jednostką podrzędną (daje sygnał wiatrakowi sufitowemu w salonie oraz budzikowi w sypialni kiedy mają się załączyć).

Ostatnią instalacją jest poruszający się moduł paneli słonecznych. Przy pomocy czujników światłoczułych instalacja jest w stanie podążać za źródłem światła, co daje optymalną pozycję panelu fotowoltaicznego.

Jak widać rozkład instalacji został zorganizowany w taki sposób, aby móc przetestować komunikację nie tylko z modułem centralnym ale i pośrednią komunikację między jednostkami podrzędnymi.

3.1.3 Elektronika i układy sensoryczne

Do realizacji wcześniej omówionych instalacji w większości przypadków wykorzystano liczne układy elektroniczne imitujące ich pracę. W przypadku czujników binarnych najczęściej pojawiał się przełącznik monostabilny. Przytrzymanie danego przycisku powodowało zasymulowanie pracy konkretnego czujnika.

Znajdujący się w załączniku B schemat (rys. 8.1) przedstawia połączenia elektryczne dla pierwszego modułu podrzędnego. Pracę wiatraka sufitowego zasymulowano poprzez włączenie do obwodu prostego silnika prądu stałego. Gdy moduł otrzyma informację o temperaturze w domu mikrokontroler sprawdzi czy wartość nie przekracza zadanego progu. W przypadku spełnienia warunku silnik załączy się. Czujnik zalania pomieszczenia działa w oparciu na pomiar wilgotności, której wartość można ręcznie symulować. Układ składa się z dwóch rezystorów. Pierwszy charakteryzuje się stałym oporem, z kolei drugi jest rezystorem regulowanym – dwa wyprowadzenia połączone są ścieżką o stałym oporze, natomiast trzecie znajduje się na tak zwanym ślizgaczu, którego położenie jest zmienne względem ścieżki. Cała operacja imitacji detektora tlenku węgla bazuje na działaniu dzielnika napięcia [3].

Przy pomocy przetwornika analogowo-cyfrowego mierzone jest napięcie, jakie odłoży się na wyprowadzeniu potencjometru. Łatwo oszacować jaką wartość zostanie odczytana w skrajnych ustawieniach potencjometru. Przy całkowitym zwarciu wyprowadzenia z masą otrzyma się wartość napięcia 0 woltów, natomiast przy maksymalnym wysterowaniu potencjometru wartość ta przyjmie 1.65V.

Zastosowany przetwornik jest 8 bitowy, zatem zakres wartości pomiarowych wynosi [0, 255]. Pomiar wykonywany jest względem napięcia referencyjnego ADC (z ang. *Analog to Digital Converter*) wynoszącego 3.3 V, które jest identyczne jak napięcie wejściowe. Dlatego też maksymalna wartość będzie wynosiła połowę całego zakresu, tj. 127. Rezultaty badania napięcia na potencjometrze są odpowiednio skalowane, aby otrzymać zakres typowy dla czujnika wilgotności (od $\sim 20\%$ do $\sim 90\%$).

System inteligentnej pobudki posiłkując się danymi przesłanymi od jednostki centralnej uaktywnia budzik. W momencie, gdy aktualna data i godzina zrówna się z zadaną mikrokontroler załącza tranzystor, co powoduje przepływ prądu przez buzzer. Aby wyłączyć budzik należy nacisnąć przycisk. Sygnał wyłączenia buzzera zostaje wysłany do modułu głównego w celu późniejszego odnotowania tego zdarzenia w aplikacji graficznej.

W podobny sposób działa czujnik czadu w drugim module podrzędnym. Naciśnięcie odpowiedniego przełącznika odbierane będzie przez system jako wykrycie tlenku węgla, w efekcie czego załączony zostanie tranzystor uruchamiając sygnał dźwiękowy.

Najbardziej złożonym i skomplikowanym układem jest moduł poruszający panelem fotowoltaicznym. Sprzężeniem zwrotnym jest pomiar natężenia światła padającego na moduł. W załączniku B na schemacie elektrycznym (rys. 8.2) widać, że informacje te pozyskiwane są dzięki fotorezystorom, które odgrywają tę samą rolę co potencjometr w instalacji detektora zalania mieszkania. W im większej ciemności czujniki te się znajdują tym wyższy opór stawiają, natomiast przy wykryciu światła opór się zmniejsza, dzięki czemu napięcie odłożone na fotorezystorze w układzie zmniejsza się. Działa to na zasadzie takiej jak w przypadku termistora typu NTC – naświetlanie podwyższa temperaturę co w efekcie zmniejsza rezystancję opornika. Znajac zakres rezystancji oszacowano maksymalną oraz minimalną wartość jaką wskaże przetwornik analogowo–cyfrowy. Dalsze przetwarzanie sygnału z pętli sprzężenia zwrotnego wyznacza dane, które służą do wygenerowania sterowania dla serwomechanizmów poruszających całą platformą.

Rozdział 4

Realizacja oprogramowania

Jak już wcześniej wspomniano projekty inteligentnych budynków najczęściej realizowane są w oparciu o programowalne sterowniki logiczne PLC. Algorytmy integrujące wszystkie instalacje implementowane są w języku drabinkowym, który jest intuicyjny i przystępny. Jednakże w niniejszym opracowaniu należało utworzyć własny protokół komunikacji stosując transmisję radiową. Dlatego też zdecydowano użyć języka programowania niższego poziomu, aby móc zrealizować wyżej wymienione zadanie. Idealnym rozwiązaniem okazuje się być język C/C++, gdyż umożliwia on obsługę sterownika różnych peryferiów. Ponadto zaplanowane moduły sterujące są programowalne w wyżej wymienionym języku, a jego dodatkowym atutem jest szereg bibliotek umożliwiających tworzenie graficznych interfejsów. W związku z tym finalnie wykonano jeden program odpowiedzialny zarówno za aplikację dostępową jak i obsługę modułu komunikacji LoRa. Całość stanowi jeden system wbudowany.

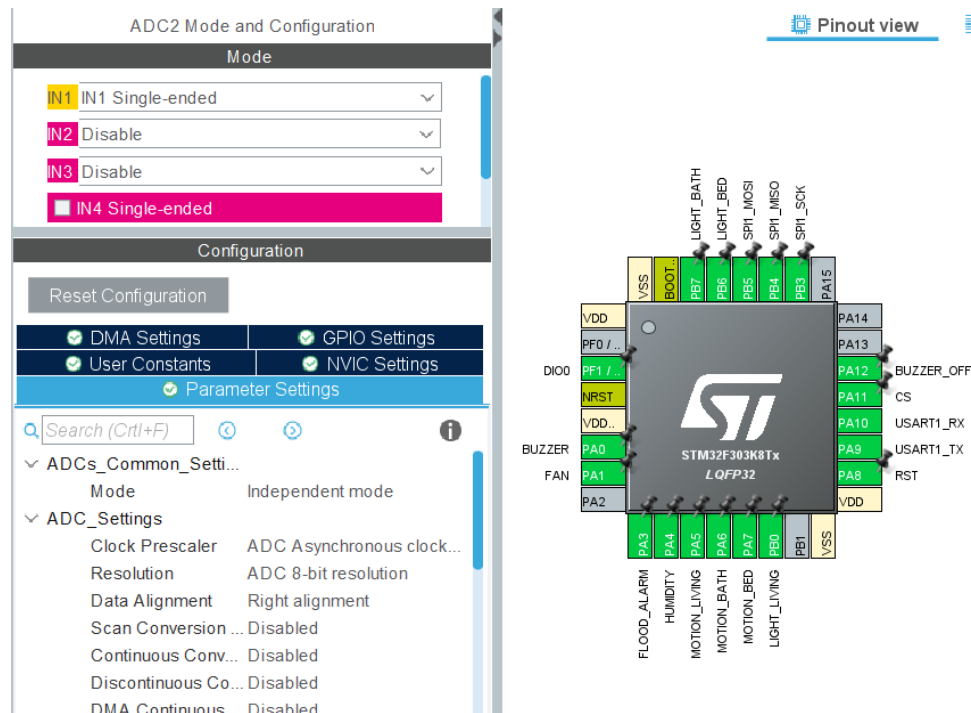
Obecne rozwiązania automatyki budynkowej zawierają gotowe moduły realizujące komunikację bezprzewodową z opracowanymi już protokołami, dlatego częściej stosowanym jest język drabinkowy, ze względu na brak potrzeby tworzenia nowych rozwiązań.

4.1 Implementacja programu na mikrokontrolerze STM

Mikrokontrolery firmy STMicroelectronics programuje się w języku C/C++. Aby ułatwić obsługę i implementowanie funkcjonalności wykorzystano środowisko STM32CubeIDE. Umożliwia ono automatyczne generowanie kodu, które konfiguruje odpowiednie rejestry mikrokontrolera w celu spełnienia zamysłu programisty.

Chcąc ustawić konkretne wyprowadzenie, aby działało na przykład jako przetwornik analogowo-cyfrowy wystarczy aktywować dany pin oraz ustawić jego typ jako ADC. Jak widać na rysunku 4.1 zawsze wypisane są możliwe opcje konfiguracji. W czasie generowania kodu przez aplikację rejestry ustawiane są w taki sposób, aby dane peryferium mikrokontrolera spełniało wymagania ustawione w procesie konfiguracji.

Ze względu na fakt, iż każdy moduł podrzędny w założeniu ma imitować pracę kilku instalacji, koniecznym jest zagwarantowanie, aby zadania były wykonywane jednocześnie. W przeciwnym razie praca poszczególnych systemów może nie działać poprawnie. Dlatego należało zapewnić jednoczesne wykonywanie się zadań. Najlepszym sposobem aby to osiągnąć jest zastosowanie systemu czasu rzeczywistego [27].



Rysunek 4.1 Przykład konfiguracji mikrokontrolera w środowisku STM32CubeIDE

4.1.1 Systemy czasu rzeczywistego

Część oprogramowania w mikrokontrolerach działa w sposób sekwencyjny i jednowątkowy – istnieje jedno zadanie, które obsługuje różne czujniki i dokonuje różnych operacji. Natomiast w sytuacji, gdy projekt się rozbudowuje, a ilość potrzebnych do wykonania zadań wzrasta pomocnym okazuje się być wsparcie systemu operacyjnego. Można je podzielić na dwa rodzaje:

- system operacyjny ogólnego przeznaczenia GPOS (z ang. *General Purpose Operating System*) – stosowany do realizacji wielu zadań jednocześnie na bardziej złożonych sprzętowo urządzeniach, na przykład serwerach, komputerach lub urządzeniach mobilnych,
- system operacyjny czasu rzeczywistego RTOS (z ang. *Real Time Operating System*) – przydatny w procesach, które mają zakończyć się w konkretnym czasie.

Druga z wymienionych rodzin została opracowana w taki sposób, aby mniej skomplikowane sprzętowo elementy (takie jak mikrokontrolery) o ograniczonych zasobach i pamięci były w stanie utrzymać ich pracę. Ponadto systemy czasu rzeczywistego dostarczają odpowiednie środki, aby zadania zostały wykonane w określonych ramach czasowych, dlatego też idealnie nadają się do systemów wbudowanych.

Pierwotne prace nad systemem, który miał przeprowadzić dany proces w określonym czasie doprowadziły do utworzenia się dwóch zbliżonych do siebie rodzajów systemów:

- miękki system czasu rzeczywistego (z ang. *soft RTOS*) – okazjonalnie zdarza się, że zadanie nie wykona się w określonym czasie. Jest to zachowanie niepożądane, lecz dopuszczalne, gdyż opóźnienia nie wprowadzają znacznych, negatywnych konsekwencji. Generalnie system stara się wykonać zadania jak najszybciej, nie znając najgorszego czasu wykonania procesu.

- twardy system czasu rzeczywistego (z ang. *hard RTOS*) – stosowane w sytuacjach, kiedy akcja musi bezwzględnie zakończyć się w określonym czasie. Ponadto znany jest najdłuższy czas wykonania zadania.

Ogólne określenie równoległego wykonywania zadań jest pozorne – procesor jest w stanie wykonywać tylko jedną operację w danej chwili. Jednakże wykonuje je tylko przez pewien czas i okresowo przełącza się między zadaniami. Dodatkowo biorąc pod uwagę fakt, że procesor dokonuje tych operacji niezwykle szybko cały mechanizm przełączania zadań daje wrażenie równoległości.

Zarówno przydzielanie czasu procesora dla poszczególnych zadań oraz sposób ich zarządzania jest koordynowane przez moduł zwany dyspozytorem (z ang. *scheduler*), czasem określany jako planista. Ponadto posiada możliwość przerywania danej operacji i udzielenia zasobów innemu procesowi (takie działanie nazywane jest wywłaszczaniem). Rozwiązuje on głównie problemy szeregowania zadań.

W zależności od systemu zaimplementowany jest inny sposób przydzielania czasu procesora. Jednym z przykładowych algorytmów jest SJF (z ang. *Shortest Job First*). Udziela on zasobów procesora zadaniom, które szacuje się, że będą trwały najkrócej. Wadą tego algorytmu jest możliwość częstego występowania sytuacji zwanej głodzeniem (z ang. *starvation*), to jest takiej, w której proces nie otrzymuje wystarczającego czasu procesora. Pojawić się może on wtedy, gdy nadchodzące zadania będą klasyfikowane jako krótkotrwałe. W takich okolicznościach dłuższe procesy mogą nigdy się nie wykonać do końca.

Często stosowanym algorytmem jest planowanie rotacyjne (z ang. *Round Robin*), które cechuje się nie uwzględnianiem priorytetów oraz przydzielaniem zasobów sprawiedliwie każdemu procesowi. Po określonym kwancie czasu zadanie zostaje wywłaszczone i ustawione na końcu kolejki.

W przypadku twardych systemów czasu rzeczywistego opracowano deterministyczne algorytmy szeregowania, takie jak algorytm EDF (z ang. *Earliest Deadline First*) oraz RMS (z ang. *Rate Monotonic Scheduling*) [18].

Zadania mogą przebywać w jednym z czterech stanów:

- stan gotowości – stan, w którym zadanie wysyła informacje planiście o tym, że mogą mu być przydzielone zasoby,
- stan wykonywania – zadanie jest w trakcie wykonywania swoich operacji,
- stan wstrzymania – zadanie zostało najprawdopodobniej wstrzymane przez inny proces i czeka na ponowne wznowienie (dyspozytor nie uwzględnia tego zadania w planowaniu),
- stan zablokowania – zadanie oczekuje na pojawienie się jakiegoś zdarzenia (na przykład odblokowania po upływie zadanego opóźnienia).

Ze względu na fakt, iż w niniejszym projekcie nie jest wymagane, aby zadania były zrealizowane w ściśle określonym czasie zdecydowano się na użycie systemu FreeRTOS, który jest typem soft RTOS. Obsługuje on tryb wywłaszczania zadań o wyższym priorytecie oraz stosuje algorytm karuzelowy (planowanie rotacyjne) w przypadkach zadań o tym samym priorytecie. Dodatkowym atutem rzeczonego systemu jest fakt, że jest zaimplementowany w języku C.

4.1.2 Architektura urządzenia wykonawczego

W każdym drugorzędnym urządzeniu została zaimplementowana liczba zadań odpowiadająca liczbie instalacji jakie mają być imitowane przez dany moduł. Priorytety tych zadań ustawiono na jednym poziomie, aby zmniejszyć prawdopodobieństwo pojawienia się zagłódzeń. Ponadto procesy związane z komunikacją radiową we wszystkich modułach podrzędnych działa w ten sam sposób, a mianowicie zadanie odpowiedzialne za nasłuchiwanie na samym początku działania wstrzymuje zadanie transmisji. W momencie, gdy odebrana zostanie informacja ze strony jednostki głównej zadanie transmisji zostaje wznowione, zaś odbiór wstrzymany. W chwili zakończenia procesu przesyłu bezprzewodowego zadanie aktualne jest wstrzymywane, natomiast odbiornik ponownie wznawiany. Aby zapewnić brak możliwości obsługi nadawania przed nasłuchiowaniem w momencie startu systemu, proces transmisji danych jest dodatkowo blokowany poprzez dodanie funkcji opóźniającej dane zadanie.

W przeciwny sposób działa jednostka centralna. Moduł główny zaczyna pracę od transmisji danych i przechodzi w stan uśpienia. Jednocześnie załącza się tryb nasłuchu informacji. W momencie odebrania informacji układ na nowo załącza tryb nadawania sygnału. Dodatkowo wprowadzono zabezpieczenie transmisji przed błędem transmisji urządzeń drugoplanowych – jeżeli czas trwania nasłuchu przekroczy zadaną wartość, automatycznie realizowane jest wyjście z trybu odbiornika.

Zadania dla dla układu podrzędnego prezentują się następująco:

- nasłuchiwanie,
- transmisja,
- detekcja zalania mieszkania,
- inteligentny budzik,
- obsługa wentylatora sufitowego,
- wykrywanie obecności w salonie,
- wykrywanie obecności w łazience,
- wykrywanie obecności w sypialni.

Utworzono je stosując funkcję *xTaskCreate()*, która jest dostępna z poziomu systemu FreeRTOS. Jednym z argumentów jest uchwyt do zadania. Za jego pośrednictwem można ingerować w pracę danego zadania. Jest to funkcjonalność, która posłuży do spełnienia założeń architektury, związanych z komunikacją radiową. W momencie odebrania sygnału zadanie odpowiedzialne za nasłuchiwanie wznawia zadanie transmisji używając funkcji *vTaskResume(TaskHandle_t xTaskToResume)*.

W przypadku drugiego urządzenia drugoplanowego lista zadań jest bardziej złożona i przyjęła następującą formę:

- nasłuchiwanie,
- transmisja,
- pomiar ciśnienia,
- odczyt informacji z zegara czasu rzeczywistego,

- kontrola zamkniętych drzwi od domu,
- kontrola obecności pojazdu w garażu,
- czujnik czadu w kuchni,
- wykrywanie obecności w kuchni,
- wykrywanie obecności w holu,
- wykrywanie obecności w garażu,
- obsługa lokalizatora słonecznego.

Poza obsługą przełączników monostabilnych narzucono zapewnienie komunikacji z innymi urządzeniami, co zwiększyło oczekiwania odnośnie modułu. Przykładem takich zadań jest obsługa czujnika temperatury, ciśnienia oraz zegara czasu rzeczywistego. Ponadto wprowadzona została obsługa licznika sprzętowego w celu wygenerowania sygnału PWM, który steruje serwomechanizmami lokalizatora słonecznego.

4.1.3 Komunikacja z podmodułami

Oprócz imitowania pracy różnych czujników zastosowano również rzeczywiste układy pomiarowe. Komunikacja z każdym z nich jest zależna od tego w jaki sposób zostało zaprojektowane urządzenie. W przypadku czujnika ciśnienia i temperatury BMP280 [24] jest to interfejs szeregowy SPI (z ang. *Serial Peripheral Interface*). Jego działanie opiera się na modelu master-slave. Urządzenie podrzędne najczęściej jest zaprogramowane i skalibrowane w jeden sposób. Aby urządzenie nadrzędne zdołało się z nim skomunikować musi odpowiednio obsłużyć magistrale danych dedykowaną techniką.

Wydzielić można rodzaj transmisji typu Full-duplex oraz Half-duplex. Pierwszy zakłada, że dane są wysyłane jednocześnie w obu kierunkach, podczas gdy w drugiej urządzenie naprzemiennie nadaje i odbiora wiadomości. W podstawowej wersji stosuje się przesył danych naprzemiennie rozdzielonymi liniami.

Aby zapewnić stałą komunikację wykorzystuje się cztery linie. MOSI (z ang. *Master Output Slave Input*) to szyna, po której wysyłane są informacje od modułu nadrzędnego do podrzędnego. Dane na linii MISO (z ang. *Master Input Slave Output*) wychodzą z układu peryferyjnego do nadrzędnego. Dzięki linii SS (z ang. *Slave Select*) moduł nadrzędny rozpoczyna komunikację z urządzeniem. Ustawiając stan niski na szynie wysyłany jest sygnał o podjęciu transmisji z danym modulem. Na szynie SCLK wysyłany jest sygnał zegarowy.

W sytuacji, gdy liczba urządzeń podrzędnych jest większa niż jeden należy jedynie zwiększyć proporcjonalnie liczbę linii SS. To dzięki nim jednostka nadrzędna zarządza z którym urządzeniem w danym momencie chce się skomunikować. Do pozostałych linii danych mogą być podpięte wszystkie urządzenia.

W ten sam sposób realizowane jest wysyłanie danych między sterownikiem a układem nadawczo-odbiorczym LoRa RFM95W [12]. Ponadto, aby odpowiednio obsłużyć proces nadawania informacji radiowo producent wprowadził dodatkowe wyprowadzenia cyfrowe wejście-wyjściowe (DIO0-DIO5). W nocie katalogowej modułu opisane jest ich przeznaczenie.

Kod obsługujący moduł LoRa został zaimplementowany inspirowany się [9] oraz [10]. Wprowadzoną modyfikacją było zastosowanie algorytmu korekcyjnego oraz opracowanie własnej ramki danych (rys. 4.2).

Transmitter Device Id	Receiver Device Id	Main message
--------------------------	-----------------------	--------------

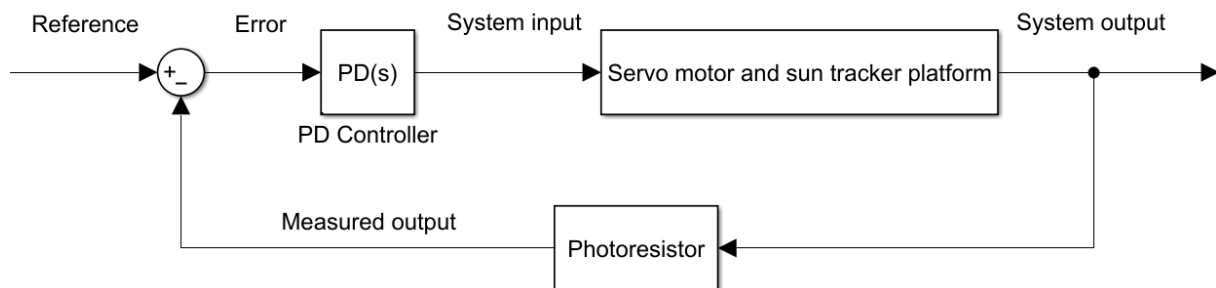
Rysunek 4.2 Model zaprojektowanej ramki danych

Pierwsze dwie informacje pakietu to numery identyfikacyjne urządzeń. Pierwszy wskazuje na moduł wysyłający wiadomość, zaś drugi wskazuje na urządzenie, do którego dana informacja ma zostać wysłana. Trzecia część ramki zawiera właściwą treść, jaką moduł zamierza przekazać. W zależności od urządzenia wiadomość może mieć inną długość. Wprowadzając każdą kolejną jednostkę podrzędną należy jedynie skonfigurować układ nadrzędny, aby był przygotowany na otrzymanie i interpretację danych z nowego modułu. Urządzenie główne w sposób iteracyjny wysyła kolejno jednostkom podrzędnym pozwolenie na transmisję zebranych danych z czujników. W przypadku, gdy jednostka nadrzędna nie otrzyma informacji zwrotnej w określonym czasie kontynuuje dalszą komunikację. Dana dotycząca numeru identyfikującego mieści się na jednym bajcie co oznacza, że do systemu może być podpięte 255 urządzeń podrzędnych (zerowy indeks jest zarezerwowany).

Innym modulem, z którym należało się skomunikować był zegar czasu rzeczywistego DS1302 [8]. Wymiana danych z tym urządzeniem różniła się od poprzednich tym, że działanie linii zegarowej należało zainicjować, co również miało miejsce w przypadku przesyłu danych.

4.1.4 Regulator PID w lokalizatorze słonecznym

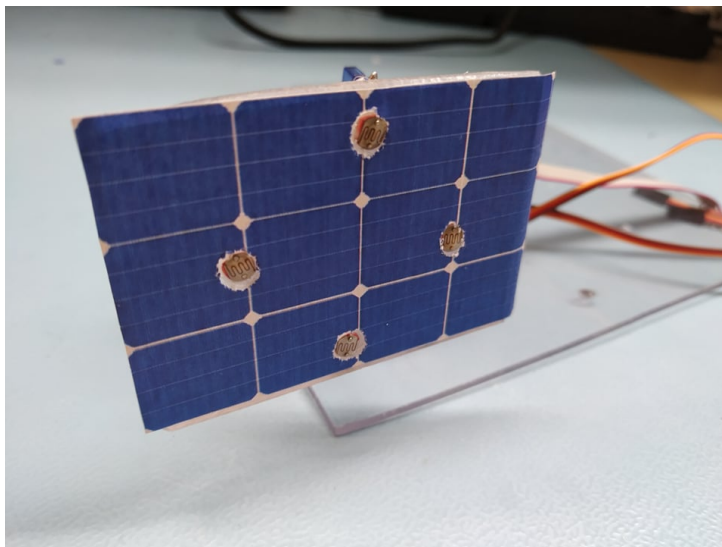
Sterowanie modulem odpowiedzialnym za manipulację platformy z panelem fotowoltaicznym działa w oparciu o regulację PID [16], która należy do najpopularniejszych i do najczęściej używanych regulacji w zastosowaniach przemysłowych. Jest to spowodowane między innymi prostotą i zadowalającą wydajnością. Regulator PID składa się z trzech członów – proporcjonalnego, różniczkującego oraz całkującego.



Rysunek 4.3 Model układu lokalizatora słonecznego [16]

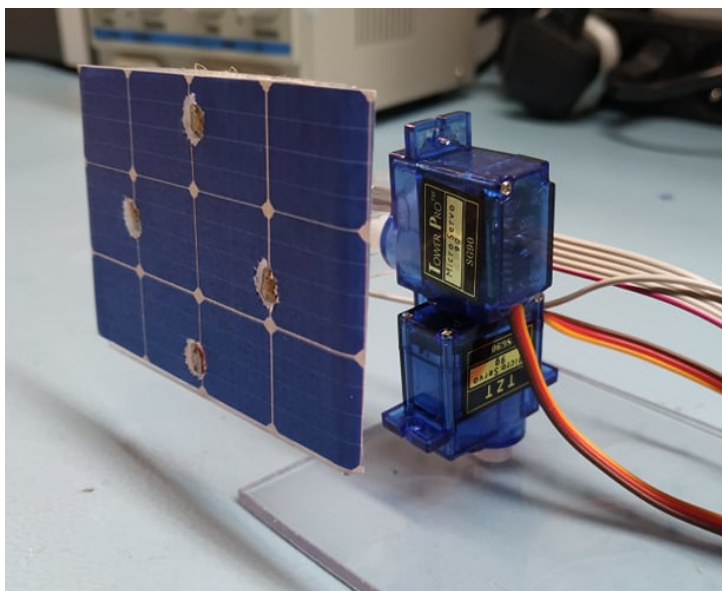
Rysunek 4.3 przedstawia przybliżony model układu oraz umiejscowiony w nim regulator PD. Jak widać wejściem regulatora zawsze jest uchyb regulacji. Celem sterowania jest zredukowanie wartości uchybu do zera, co w rzeczywistości oznacza, że obiekt zachowuje się w pożądanym sposób (śledzi wartość zadaną).

W przypadku lokalizatora słonecznego wartość zadana wynosi zero, zaś wartościami ze sprzężenia zwrotnego są wyniki pomiarów z fotorezystorów. Rozmieszczenie czujników na platformie widać na rysunku 4.4.



Rysunek 4.4 Rozmieszczenie czujników światła na platformie

Ich rozlokowanie nie jest przypadkowe – zorientowanie fotorezystorów góra–dół posłużyło do zdecydowania czy moduł ma się unieść bądź opuścić. Dodatkowe czujniki na bokach wykorzystano do zorientowania platformy w płaszczyźnie poziomej. Znak różnicy wyników pomiarów odpowiednich czujników decydował o kierunku ruchu platformy, zaś od wartości tej różnicy zależała prędkość przesuwania się modułu. Cały układ dążył do tego, aby źródło światła znajdowało się idealnie między czujnikami. Oznacza to taką samą wartość napięć odłożoną na fotorezystorach co wiąże się z tym, że ich różnica musi być równa zero (stąd też wartość sygnału zadanego).



Rysunek 4.5 Widok całego lokalizatora słonecznego

Serwomechanizmy są ustawione w taki sposób, aby platforma z czujnikami mogła osiągać możliwie największy zakres ruchu. Na rysunku 4.5 przedstawiono sposób ich połączenia.

Rozdział 5

System zarządzający

Elementem odpowiedzialnym za poprawną pracę systemu zarządzającego jest mini-komputer Raspberry Pi. Jego atutem jest fakt, iż korzysta on z dystrybucji Linuksa opartego na Debianie. Daje to szereg możliwości, takich jak korzystanie z interfejsu graficznego oraz brak ograniczenia w postaci wymogu kompilacji skrośnej (z ang. *cross-compilation*). Oznacza to, że program może zostać zbudowany na maszynie docelowej, w przeciwieństwie do mikrokontrolera STM32, do którego zbudowany kod należy wysłać. Ponadto dla różnych języków programowania powstały biblioteki umożliwiające obsługę wyprowadzeń i peryferiów urządzenia. Cechy te powodują, że platforma idealnie nadaje się do projektów zawierających zadania złożone oraz wymagające obliczeniowo.

Zdecydowano, aby jeden kod był odpowiedzialny za wszystkie funkcjonalności, do których należy obsługa interfejsu graficznego, zarządzanie modułami podrzędnymi oraz komunikacja z układem nadawczo-odbiorczym LoRa. Odbiór i wysyłanie wiadomości radiowych odbywa się w ramach jednego wątku, natomiast przetwarzanie otrzymanych danych jak i wizualizacja dokonują się w oddzielnym wątku utworzonym z poziomu biblioteki Qt.

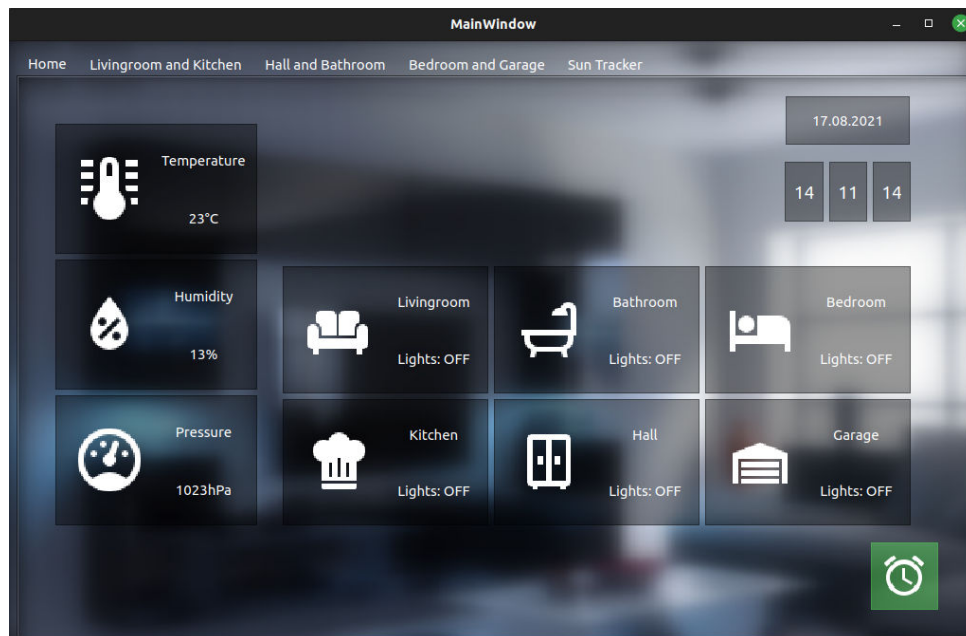
W celu skomunikowania się z modułem transmisyjnym wykorzystano gotową bibliotekę *WiringPi*, która posiada zadeklarowane oraz zdefiniowane funkcje obsługujące peryferia urządzenia. W znacznym stopniu ułatwiło to zainicjowanie komunikacji przez interfejs SPI.

5.1 Graficzny interfejs użytkownika

Do stworzenia aplikacji dostępowej wykorzystano framework Qt. Jest to zestaw narzędzi w postaci bibliotek, które umożliwiają budowę graficznych interfejsów zarówno dla programów komputerowych jak i urządzeń mobilnych. Pozwalają na optymalne tworzenie kodu poprzez implementację różnych funkcjonalności w sposób obiektowy. Oznacza to, że poszczególne rozwiązania są ściśle związane z konkretnym obiektem. Ponadto aspekt obiektowości pozwala na rozszerzenie danych funkcjonalności dzięki tworzeniu klas pochodnych, dziedziczących po istniejących już klasach.

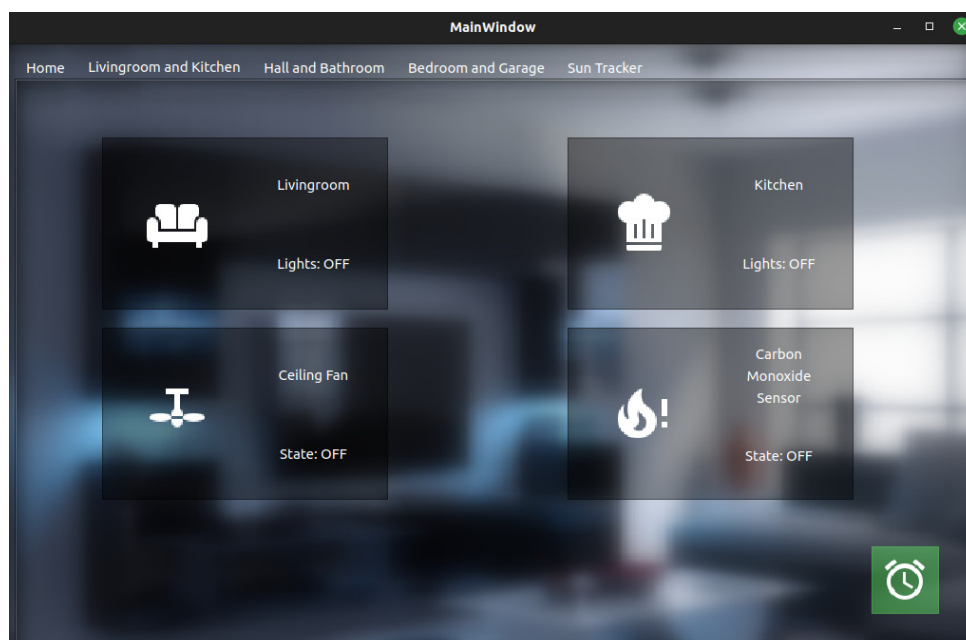
Aktualna wersja bibliotek Qt oprócz możliwości związanych z wizualizacją posiada również szereg udogodnień z innych obszarów. Przykładem tego jest biblioteka *QtSql*, która jest swojego rodzaju interfejsem między aplikacją a bazą danych. Stanowi to podstawę do inicjatywy rozszerzenia niniejszego projektu o zastosowanie bazy danych i zaimplementowanie kodu odpowiedzialnego za komunikację z tą bazą.

Aplikacja graficzna składa się z pięciu dostępnych okien, na których wyświetlane są dane pochodzące z czujników. Podstawowym jest widok główny (rys. 5.1).



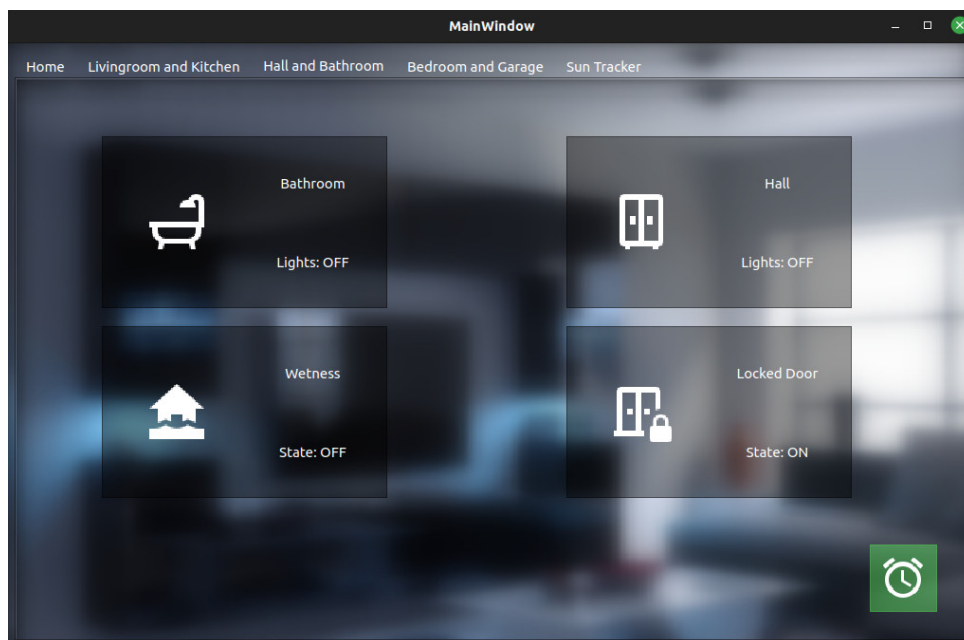
Rysunek 5.1 Główne okno interfejsu graficznego

Widoczne są na nim podstawowe informacje takie jak aktualna data, temperatura w mieszkaniu oraz stan oświetlenia poszczególnych pokojach. W dolnym prawym rogu widnieje ikona budzika. Pojawia się on na każdym z dostępnych okien. Jest ona podświetlona na zielono w przypadku nieaktywnego budzika. W sytuacji uruchomienia alarmu podświetla się ona na czerwono. Kolejne trzy okna związane są z poszczególnymi pomieszczeniami. Ich cechą wspólną jest informowanie o zapalonych światłach w każdym z pokoi, co wiąże się z obecnością domowników.



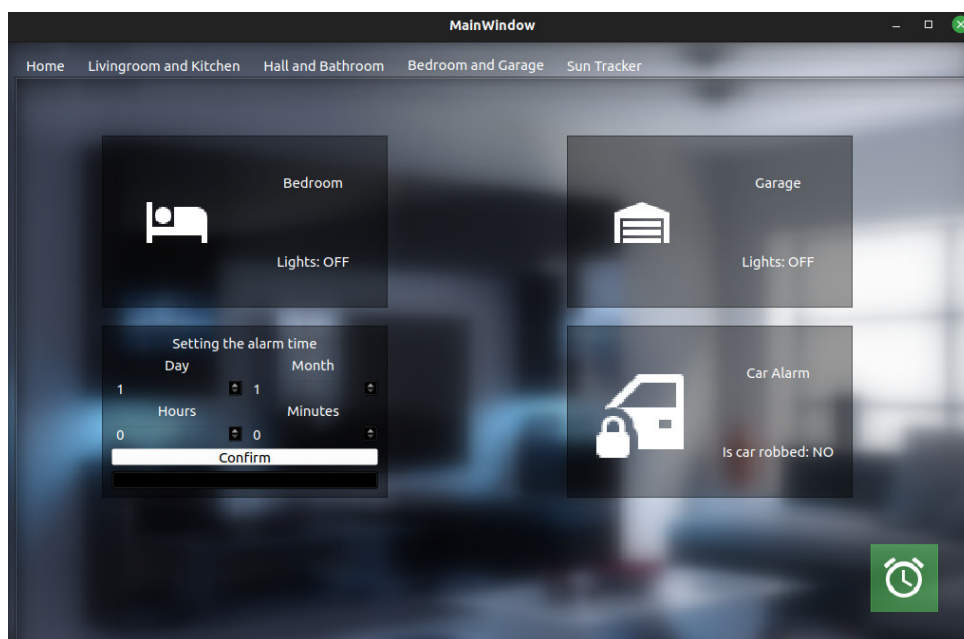
Rysunek 5.2 Wizualizacja pracy instalacji w salonie i kuchni

Rysunek 5.2 przedstawia okna udostępniające dane o salonie i kuchni. Informacje dotyczą instalacji wykorzystanych w poszczególnych pokojach. Sprawdzane jest czy załączony



Rysunek 5.3 Wizualizacja pracy systemów w holu i łazience

został wiatrak sufitowy oraz czy czujnik czadu nie wykrył tlenku węgla. Na kolejnym oknie (rys. 5.3) widnieje informacja o tym czy łazienka nie jest zalana oraz czy zamknięte są drzwi zewnętrzne.

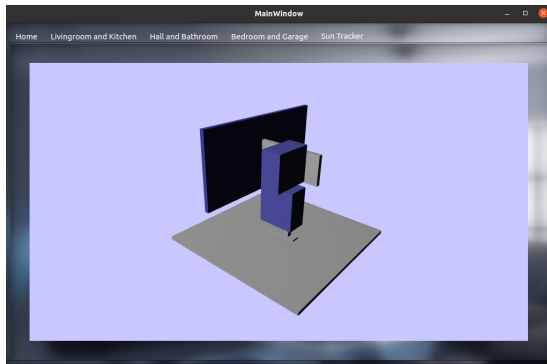


Rysunek 5.4 Graficzne przedstawienie instalacji w sypialni oraz w garażu

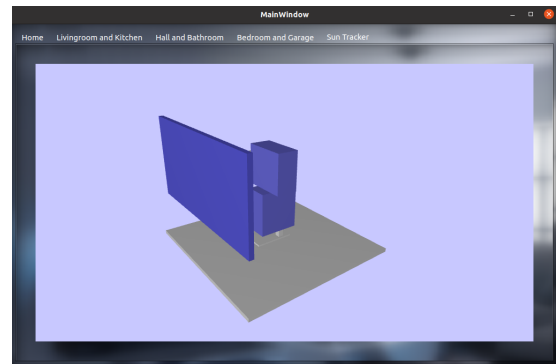
Szczególną uwagę należy zwrócić na okno dotyczące systemów w sypialni oraz garażu (rys. 5.4). Oprócz wizualizacji alarmu samochodowego widać ramkę, w której realizowane jest ustawianie inteligentnego budzika. W sposób klarowny można nastawić go i uruchomić z poziomu aplikacji graficznej.

Ostatnie okno (rys. 5.5) odpowiada za wizualizację orientacji lokalizatora słonecznego. Prezentuje uproszczony model całej platformy, aby skupić uwagę jedynie na ustawieniu

i wychyleniu poszczególnych serwomechanizmów. Do wygenerowania figur trójwymiarowych wykorzystano bibliotekę *Qt3D*.



(a)



(b)

Rysunek 5.5 Okna ilustrujące pracę lokalizatora słonecznego

Rozdział 6

Algorytmy korekcyjne

Podczas bezprzewodowej komunikacji cyfrowej, tak jak w przypadku sygnałów analogowych, istnieje prawdopodobieństwo wystąpienia szumów, które w mniejszym lub większym stopniu wpływają na jakość transmisji. Gdy dane wejściowe nie są dopasowane z wyjściowymi określa się to mianem błędu. W transmisji cyfrowej błędy mogą dotyczyć pojedynczych bitów, to jest najmniejszej porcji przesyłanej informacji.

Wpływ błędu na urządzenie można podzielić na dwa typy – błąd łagodny oraz twardy. Pierwszy rodzaj charakteryzuje się tym, że nie ma trwałego wpływu na zdolność przechowywania i przetwarzania danych konkretnego urządzenia. Takie błędy mogą być spowodowane na przykład promieniowaniem kosmicznym. Częstość występowania takich błędów jest zależna od konstrukcji danego urządzenia oraz od miejsca jego działania.

Błędem twardym określa się błędy, które powodują trwałe uszkodzenie modułu. Podczas odczytu wartość będzie zawsze wynosiła stałą wartość (binarne 0 bądź 1), niezależnie od zapisanej wartości podczas nadawania. Przyczyną tego typu błędów są wady produkcyjne, nieprawidłowe lub długotrwałe działanie układu. Ponadto błędy mogą występować pojedynczo bądź wielokrotnie.

Wywiera to ogromny wpływ na jakość przesyłanych danych. Z tego powodu coraz bardziej powszechne stawało się opracowywanie różnych technik wykrywania i korygowania błędów [21]. Proces identyfikacji błędów występujących w przesyłanych danych nazywany jest wykrywaniem błędów. Polega na użyciu nadmiarowych bitów, na podstawie których określa się gdzie nastąpił błąd.

Podstawową techniką jest wykrywanie błędów parzystości, która polega na liczeniu cyfry „1” w danej ramce. W tym celu dodaje się dodatkowy bit zwany bitem parzystości, którego wartość ustawia się 0 lub 1, aby liczba wszystkich jedynek była parzysta w słowie. Gdy odbiornik wykryje nieparzystą ich ilość oznaczać to będzie wystąpienie błędu w przesyłach. Wśród najpopularniejszych technik można znaleźć:

- jednowymiarowa kontrola parzystości – zakłada sprawdzenie ilości jedynek w jednym ciągu danych.
- dwuwymiarowa kontrola parzystości – polega na podziale ramki danych na równe odcinki, skonstruowanie z nich macierzy oraz sprawdzenie parzystości zarówno dla wierszy jak i kolumn
- cykliczna kontrola nadmiarowości CRC (z ang. *Cyclic Redundancy Check*) – działa w oparciu na obliczaniu reszty z dzielenia ciągu danych przez $(n + 1)$, gdzie n – ilość dodatkowych bitów.

Wadą technik parzystości niestety jest to, że pozwalają na wykrycie jedynie pojedynczej zmiany.

6.1 Kod Hamminga

Kodowanie Hamminga jest algorytmem korekcyjnym, który wykrywa błędy jedno- oraz dwubitowe. Redundantne bity parzystości dodawane do sekwencji danych umieszczane są na pozycjach o numerach kolejnych potęg liczby dwa. Każdy bit parzystości sprawdza informacje znajdujące się na odpowiednich pozycjach ramki danych. Na rysunku 6.1 widać przykład sytuacji, w której jedna wartość została zmieniona w trakcie działania przesyłu. Utworzony błąd zaburzył parzystość przez co został wychwycony przez bity parzystości, które były odpowiedzialne za kontrolę bitu w pozycji 11. Dodanie do siebie pozycji tych bitów parzystości, których kontrola zakończyła się niepowodzeniem zwraca pozycję błędu (w przypadku przytoczonego przykładu suma jest równa 11).

Pozycja	1	2	3	4	5	6	7	8	9	10	11	Rezultat kontroli
Binarnie	1	10	11	100	101	110	111	1000	1001	1010	1011	
Odebrany bit	1	1	1	0	1	0	1	0	1	0	0 (błąd)	
Bit parzystości:1	1		1		1		1		1		0	wykryty błąd
Bit parzystości:2		1	1			0	1			0	0	wykryty błąd
Bit parzystości:3				0	1	0	1					brak błędu
Bit parzystości:4								0	1	0	0	wykryty błąd

Tabela 6.1 Tabela wyników dla testów komunikacji radiowej

Jest to sposób do wykrywania najczęściej pojedynczych błędów. Jak widać na powyższym przykładzie sposób implementacji nie należy do najtrudniejszych oraz nie powoduje znacznego zwiększenia rozmiaru ramki danych. Pozycja bitów parzystości rośnie w sposób wykładniczy, co w efekcie daje tym mniejszy stosunek ich ilości do całego zakresu danych im większą ramkę zamierza się wysłać.

Wymiana informacji między użytymi modułami LoRa wykorzystuje sumy kontrolne CRC, natomiast zdecydowano się na nadmiarowe użycie omówionego algorytmu korekcyjnego ze względu na dodatkową kontrolę transmisji danych niskim kosztem złożoności pakietu oraz na możliwość wprowadzenia korekty w przypadku pojawienia się uszkodzonych danych.

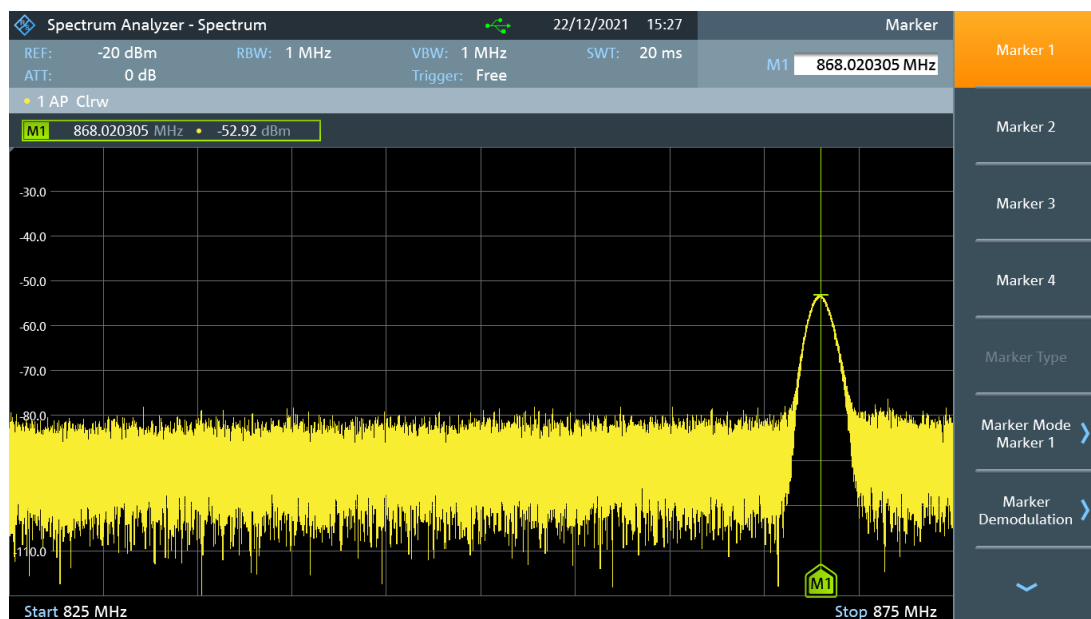
Rozdział 7

Testy użytkownika

Po zintegrowaniu wszystkich modułów (sterowników oraz systemów imitujących poszczególne instalacje inteligentnego budynku) przetestowano wszystkie funkcjonalności. Do najważniejszych kwestii, które należało przetestować zaliczają się:

- sprawdzenie czy moduły nadawczo–odbiorcze komunikują się ze sobą,
- skontrolowanie czy podstawowe funkcjonalności (takie jak czujniki obecności) działają,
- przetestowanie czy aplikacja graficzna wizualizuje pracę konkretnych modułów,
- sprawdzenie czy istnieje pośrednie połączenie między urządzeniami podrzędnymi,
- przetestowanie poprawności kodowania Hamminga.

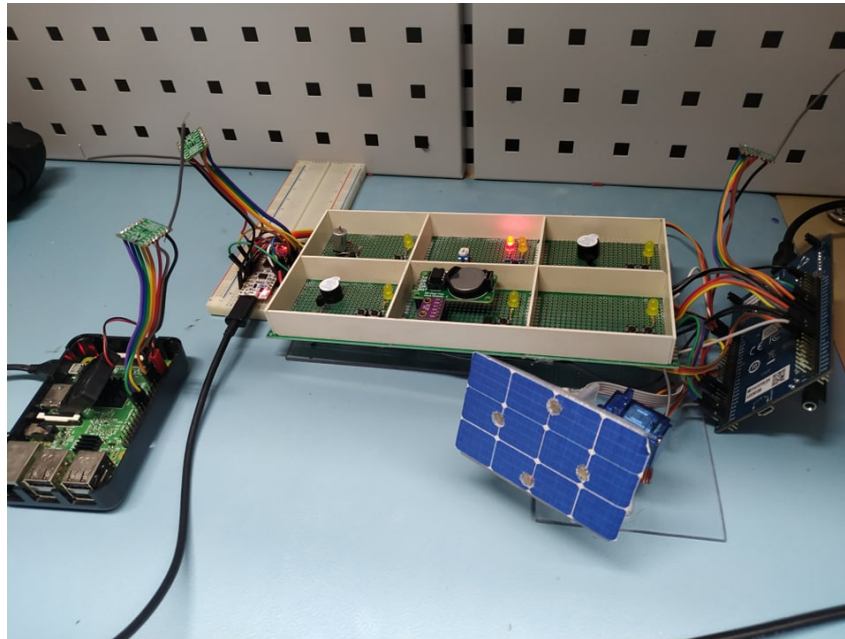
Używając analizatora widma można w prosty sposób sprawdzić czy nadajnik LoRa działa poprawnie. Wiedząc, że moduł ten dokonuje transmisji na częstotliwości 868MHz ustawiono pomiar na zakres od 825MHz do 875MHz.



Rysunek 7.1 Detekcja sygnału na częstotliwości 868 MHz

Na rysunku 7.1 widać widmo sygnału, które jest odbierany przez urządzenie. Dostrzegalny jest nagły wzrost wartości natężenia sygnału dla szukanej częstotliwości, podczas gdy w pozostałej części wykresu znajdują się zebrane przez układ szumy z otoczenia. Świadczy to o poprawności skalibrowania modułu.

Na ilustracji 7.2 widoczny jest system podczas testów. Ze względu na prostotę wykonania układów symulujących pracę instalacji budynkowych w łatwy sposób można było przetestować ich działanie. Włączając przycisk symulujący pracę czujnika obecności pomyślnie załączono diodę LED, która z kolei imitowała oświetlenie w pomieszczeniu. Na takiej samej podstawie skontrolowano pozostałe podsystemy automatyki budynkowej, jednocześnie sprawdzając czy aktualny stan się zgadza z tym co jest prezentowane w aplikacji graficznej. Testy tych funkcjonalności zakończył się pozytywnie.



Rysunek 7.2 Widok układów podczas testów

Aby ocenić czy urządzenia podrzędne komunikują się między sobą należało sprawdzić czy silnik symulujący pracę wiatraka sufitowego załączy się w miarę podgrzewania czujnika temperatury. Po wstępnym przetestowaniu stwierdzono, że układ zachowuje się zgodnie z przewidywaniami.

Do bardziej złożonych testów należało skontrolowanie działania kodowania Hamminga. Polegało na załączeniu dwóch układów komunikujących się modułem nadawczo-odbiorczym LoRa przez określony czas. Podczas komunikacji należało wykrywać i składować informacje o ilości wystąpień błędów, przy których udało się dokonać korekcji oraz o ilości błędów, których nie dało się poprawić (przekłamanie dla więcej niż jednego bitu danych).

Sprawdzając poprawność kodowania Hamminga jednocześnie skontrolowano jakość wysyłania pakietów danych w zależności od warunków zewnętrznych. Dokonując transmisji z okesem co 2 sekundy przesyłano wiadomości przez pół godziny, co daje łącznie 900 pakietów danych. Jest to ilość wystarczająca, aby dokonać analizy i sformułować wnioski.

Parametrami w przeprowadzonym teście była długość ramki danych oraz dystans między urządzeniami. Wygenerowano pakiety losowych danych przeznaczonych do przesyłu. Pierwsza długość wiadomości wynosiła 12 bajtów, co implikuje wykorzystanie sześciu bitów parzystości, które mieszczą się na jednym bajcie. Dlatego ostatecznie ramka posiadała

13 bajtów. W przypadku drugiej długości wygenerowano ciąg 126 znaków, dla których przypada 9 bitów parzystości. Po zaokrągleniu wartości w górę długość wynosiła 2 bajty. Stąd otrzymano wiadomość składającą się ze 128 bajtów. Dla każdej długości ramki wykonano testy komunikacji przy dystansie między urządzeniami równym 1 metr oraz 30 metrów. Wyniki zaprezentowano w tabeli 7.1.

Kryteria	13 bajtów, 1m odległości	13 bajtów, 30m odległości	128 bajtów, 1m odległości	128 bajtów, 30m odległości
1 zmieniony bit	0	0	0	0
2 zmienione bity	0	0	0	0
3 zmienione bity	0	0	0	0
4+ zmienione bity	1	1	2	2
suma błędów	1	1	2	2
Poprawne pakiety	899	898	896	898
Dostarczone pakiety	900	901	898	900
Utracone pakiety	0	1	2	0
Wysłane pakiety	900	900	900	900
Dodatkowe pakiety (zakłócenia)	0	2	0	0

Tabela 7.1 Tabela wyników dla testów komunikacji radiowej

Analizując powyższe rezultaty można dojść do kilku wniosków. Przede wszystkim ilość pojawiających się błędów jest stosunkowo niska (w najgorszym przypadku stosunek wynosi 0.22%). Ponadto ilość ta utrzymuje pewną stałość niezależnie od wielkości ramki oraz dystansu między urządzeniami.

Zastosowana większa odległość została dobrana w taki sposób, aby sprawdzić czy moduły komunikacyjne spełnią swoje zadanie w warunkach charakterystycznych w automatyce budynkowej, to znaczy czy urządzenia będą w stanie przesyłać informacje mimo znacznego dystansu oraz występujących przeszkód na przykład w postaci ścian. W przypadku nieznacznego wzrostu ilości błędów przy wprowadzeniu urządzenia do powyżej opisanych warunków można wywnioskować, że moduł LoRa idealnie nadaje się do transmisji danych między instalacjami inteligentnego budynku.

Nie można wykluczyć sytuacji, w której system utraci pewne informacje podczas przesyłu danych bądź odbierze nieoczekiwane dane, spowodowane napotkanym szumem.

Rozpatrując rodzaj napotkanych błędów można zauważyć, że w żadnym z testów nie napotkano błędu, w którym wystąpiłaby zmiana pojedynczego bitu. Może to świadczyć o redundancji użycia kodowania Hamminga. Dla łącznej ilości 3600 pakietów informacji nie zarejestrowano ani jednego przypadku, w którym algorytm korekcyjny musiałby dokonać korekty danych.

Zakładając, że system automatyki budynkowej ma działać w czasie rzeczywistym, co oznacza większą częstość wysyłania danych błędy mogą okazać się niezauważalne, zaś kodowanie Hamminga zbędne.

Rozdział 8

Podsumowanie

Celem pracy było wykonanie fizycznego modelu inteligentnego budynku, zaprogramowanie modułów imitujących pracę instalacji automatyki budynkowej oraz przeprowadzenie testów poprawności działania instalacji. Ponadto należało opracować sposób komunikacji między poszczególnymi modułami. Założenia projektu pomyślnie zrealizowano.

Na początku pracy zostały przytoczone przykłady istniejących systemów wchodzących w skład głównego systemu zarządzania budynkiem oraz dotychczasowe rozwiązania, dotyczące używanych urządzeń i możliwości skomunikowania ich między sobą. Następnie podano architekturę systemu, opisując strukturę i hierarchię urządzeń w projekcie oraz w jaki sposób moduły zostały ze sobą połączone. Podano zastosowane urządzenia, które dostosowano do wymagań systemowych – przeanalizowano złożoność poszczególnych zadań podsystemów i na ich podstawie dobrano odpowiednie urządzenia. Rozplanowano przykładowy rozkład pomieszczeń w budynku oraz dopasowano do nich sterowniki. W dalszym etapie omówiono w jaki dokładnie sposób dokonano imitacji poszczególnych instalacji. Zdecydowano się wykorzystać przetwornik analogowo–cyfrowy oraz wyprowadzenia cyfrowych wejść–wyjść dostępnych w strukturze mikrokontrolera.

Omówiono również jak wygląda obsługa podsystemów od strony oprogramowania, przedstawiono w jaki sposób podzielono zadania dla każdego sterownika oraz omówiono obsługę modułu nadawczo–odbiorczego. Opisano sposób realizacji kodowania Hamminga oraz przedstawiono implementację aplikacji graficznej.

Po zbudowaniu, zaimplementowaniu i zintegrowaniu wszystkich podzespołów przeprowadzono testy weryfikujące działanie wszystkich funkcjonalności. Owe testy zakończyły się pomyślnie w szczególności testy komunikacji oraz użycia kodowania Hamminga. Kluczową ich konkluzją jest fakt, iż zintegrowanie systemów automatyki budynkowej z wykorzystaniem LoRa jest możliwe i efektywne. Ilość pojawiających się błędów mimo znacznego dystansu między modułami oraz przeszkód w postaci ścian jest na tyle niska, że użycie algorytmów korekcyjnych staje się zbędne.

Jednym z możliwych usprawnień może być zaprojektowanie płytki drukowanej. Takie rozwiązanie zmniejszyłoby prawdopodobieństwo wystąpień błędów wynikających z źle wykonanych połączeń lutowniczych. Ponadto stosując mikrokontrolery o większej ilości pamięci operacyjnej możliwe jest rozszerzenie obsługi o dodatkowe zadania. Architektura systemu została dobrana w taki sposób, aby bezproblemowo móc dołączyć elementy wykonawcze nie wprowadzając znacznych zmian w oprogramowaniu. Jak już wcześniej wspomniano możliwe jest uruchomienie bazy danych połączonej z aplikacją graficzną w celu gromadzenia danych i prowadzenia statystyk. Sam graficzny interfejs użytkownika można również skonfigurować, aby działał na urządzeniu mobilnym. W tym celu należałoby dodatkowo wykorzystać moduł komunikacji Wi-Fi w jednostce głównej, aby móc łączyć się

z nią przy pomocy telefonu komórkowego.

Jak widać możliwości rozwoju są ogromne. Sam projekt może być punktem wyjścia do dalszej rozbudowy oraz inspiracją do wykorzystania zaproponowanego sposobu komunikacji, ramki danych oraz architektury systemu w innych projektach wymagających bezprzewodowej komunikacji na odległość.

Literatura

- [1] Model OSI. <https://ejix.weebly.com/model-osi.html>, 2021.
- [2] M. Alhaisoni. IoT Energy Efficiency through Centrality Metrics. *Annals of Emerging Technologies in Computing*, strony 14–21, 2019.
- [3] D. Arbutina, A. Vasic, U. Kovacevic. Probabilistic analysis of voltage divider ratios. *FME Transactions*, strony 624–633, 2019.
- [4] M. Bhagyaveni, R. Kalidoss, K. Vishvakshan. *Introduction to Analog and Digital Communication*. River Publishers Series in Communications. River Publishers, 2016.
- [5] M. Brohi. Integration of IoT and blockchain. *Technium Romanian Journal of Applied Sciences and Technology*, strony 32–41, 2021.
- [6] M. Chincoli, A. Liotta. Self-Learning Power Control in Wireless Sensor Networks. *Sensors*, strona 375, 2018.
- [7] Connect Data - Sintel. IoT – Internet des Objets. <https://www.connectdata.fr/solution/iot-et-m2m/>, 2021.
- [8] Dallas Semiconductor. *DS1302 datasheet*, 2021.
- [9] W. Domski. LoRa driver for Raspberry Pi with Python wrapper. 2021. <https://blog.domski.pl/lora-driver-for-raspberry-pi-with-python-wrapper/>.
- [10] W. Domski. STM32 HAL driver for LoRa SX1278 wireless module. 2021. <https://blog.domski.pl/stm32-hal-driver-for-lora-sx1278-wireless-module/>.
- [11] S. Haykin, B. Galiński, G. Hahn. *Systemy telekomunikacyjne: 1*. Wydawnictwa Komunikacji i Łączności, 1998.
- [12] HOPERF Electronics. *RFM95/96/97/98(W) – Low Power Long Range Transceiver Module: datasheet*, 2021.
- [13] J. Hoque, M. Ahmed. Smart Home: An Empirical Analysis of Communication Technological Challenges. *European Journal of Engineering Research and Science*, strony 571 – 575, 2020.
- [14] V. Joe, S. Sugi. Spread Spectrum Modulation Techniques using MATLAB. *International Journal of Innovative Research in Technology*, strona 5, 2015.
- [15] T. Kramp, R. Kranenburg, S. Lange. *Introduction to the Internet of Things*, strony 1–10. 2013.

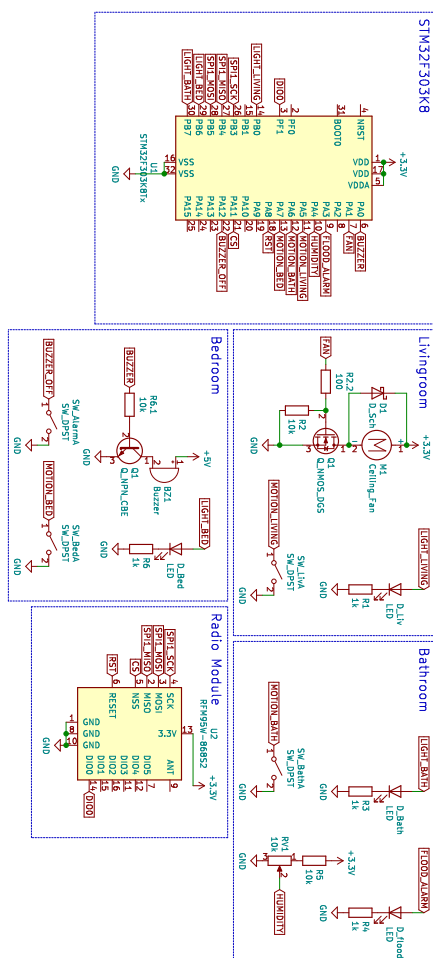
- [16] E. Köse. Optimal Control of AVR system with Tree Seed Algorithm-based PID Controller. *IEEE Access*, strony 89457–89467, 2020.
- [17] M. Latif, A. Dafid, V. Widyaningrum, A. Romadhon, S. Wahyuni. The Control of Home Appliances Using Voice Command Based on Speech Recognition. *Advanced Science Letters*, strony 12417–12419, 2017.
- [18] T. Li, D. P. Baumberger, S. Hahn. Efficient and scalable multiprocessor fair scheduling using distributed weighted round-robin. *PPoPP '09*, 2009.
- [19] N. Minhas, D. Kumar. Performance Analysis of ISM Band Antennas: A Survey. *International Journal of Advanced Computer Research*, 2018.
- [20] R. Mubashar, M. Abu Bakar, A. Rehman, A. Asad, A. Rasool. Comparative performance analysis of short-range wireless protocols for wireless personal area network. *Iran Journal of Computer Science*, 2021.
- [21] P. Prakash, A. Manoj, D. S. Mukunda. Math model on error detection and correction. *International Research Journal of Engineering and Technology*, 2020.
- [22] Raspberry Pi Foundation. *Raspberry Pi 3 Model B+: datasheet*, 2021.
- [23] Real Time Engineers Ltd. FreeRTOS™ Real-time operating system for microcontrollers, 2021.
- [24] Robert Bosch GmbH. *BMP280 Digital Pressure Sensor: datasheet*, 2021.
- [25] D. Sharma, S. Verma, K. Sharma. Network topologies in wireless sensor networks: A review. *Int. J. Electron. Commun. Technol.*, strony 93–97, 2013.
- [26] E. Skondras, T. Mavroeidakos, D. Vergados. The Convergence of Blockchain, Internet of Things (IoT) and Building Information Modeling (BIM): The smart museum case. 2019.
- [27] A. Tanenbaum, H. Bos. *Modern Operating Systems*. Always learning. Pearson, 2015.
- [28] W. Tariq, S. Khan, A. Mustafa. Building Management System. *ASIAN JOURNAL OF ENGINEERING, SCIENCES TECHNOLOGY*, strona 106, 2012.
- [29] Texas Instruments. *BeagleBone Black System Reference Manual: datasheet*, 2021.
- [30] D. Tomaszewski. Standardy komunikacji bezprzewodowej dla urządzeń IoT. <https://elektronikab2b.pl/technika/49873-standardy-komunikacji-bezprzewodowej-dla-urzadzen-iot>, 2018.
- [31] M. C. Yuen, S. Y. Chu, W. Chu, H. Cheng, H. Ng, S. Yuen. A low-cost IoT smart home system. *International Journal of Engineering and Technology*, 2018.

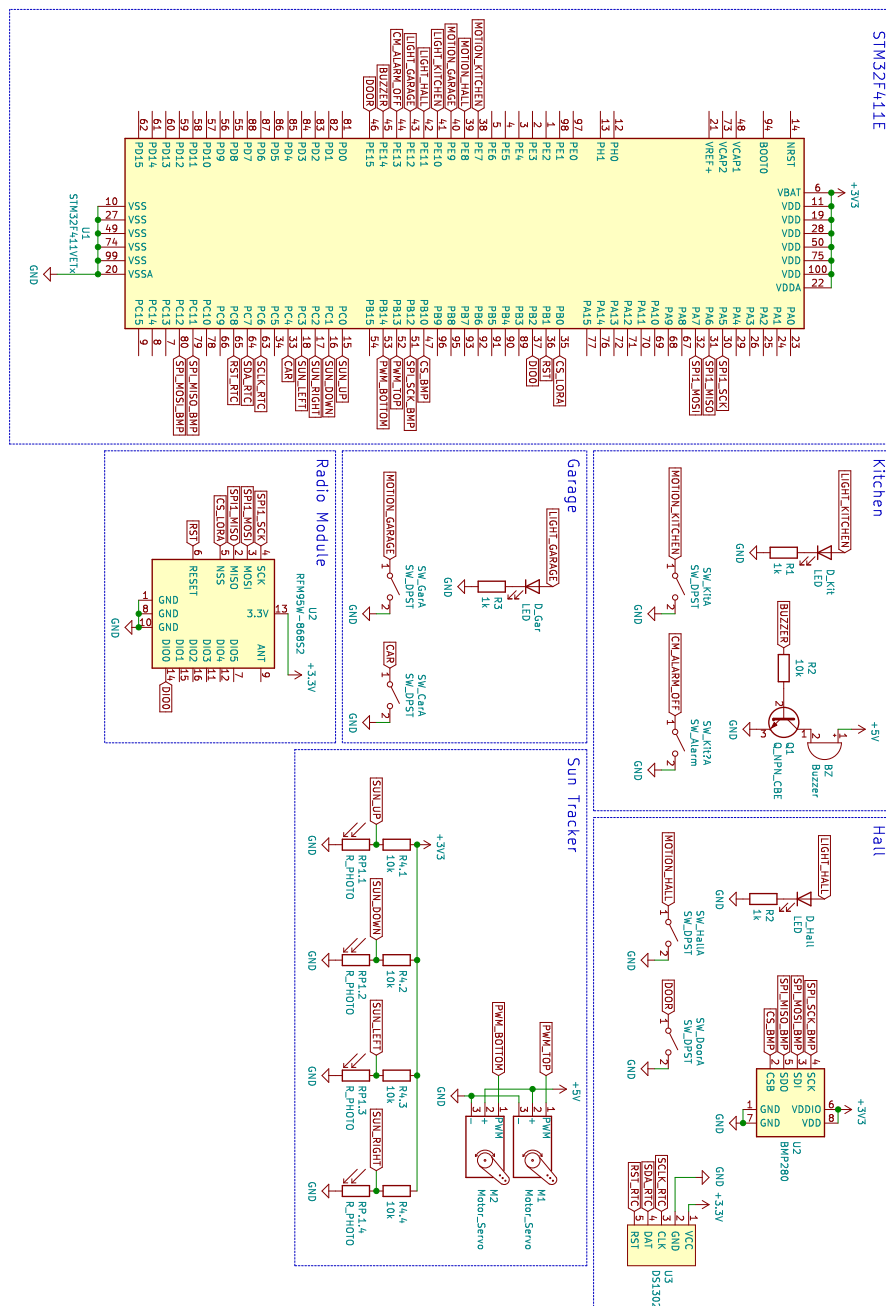
Załącznik A

Do pracy załączono płytę CD zawierającą poszczególne pliki i katalogi:

- /Praca_inzynierska.pdf – wersja cyfrowa pracy dyplomowej,
- /Schematy.zip – schematy elektryczne układów,
- /Kod_zrodlowy_stm – kod źródłowy oprogramowania do sterowników STM32,
- /Kod_zrodlowy_rpi – kod źródłowy oprogramowania do platformy Raspberry Pi.

Rysunek 8.1 Schemat elektryczny sterownika oraz instalacji dla salonu, łazienki oraz sypialni.





Rysunek 8.2 Schemat elektryczny sterownika oraz instalacji dla kuchni, holu i garażu.