

Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH



Instytut Mikroelektroniki i Optoelektroniki

Praca dyplomowa inżynierska

na kierunku Elektronika i Telekomunikacja
w specjalności Inżynieria Komputerowa

Dedykowany system komunikacji z wykorzystaniem protokołu Lora

Emil Michalik

Numer albumu 280293

promotor

dr inż. Marek Niewiński

WARSZAWA 2024

Podziękowania

Jestem bardzo wdzięczny mojemu promotorowi za bezcenne rady. Mojej babci Apolonii za wiarę we mnie. Natomiast moim rodzicom za niekończące się i przepastne pokłady cierpliwości.

Streszczenie

Tytuł: Dedykowany system komunikacji z wykorzystaniem protokołu Lora

Obecnie dostępne systemy komunikacji wykorzystują w dużej mierze Internet, a co w sytuacji gdy go zabraknie? Na to pytanie postaram się odpowiedzieć w mojej pracy.

W mojej pracy starałem się przybliżyć różne aspekty konstruowania, wykonania, bądź działania systemów komunikacji. Ponadto opisałem działanie gotowego systemu mojego autorstwa

Mój system składa się (od strony) oprogramowania z dwóch elementów: aplikacji na mikrokontroler i aplikacji mobilnej napisanej na urządzenia z systemem operacyjnym Android.

Natomiast od strony sprzętowej jest to mikrokontroler Raspberry Pi Pico (RP2040), moduł Bluetooth i moduł do komunikacji na częstotliwości sub-1 Ghz

Wiele kwestii dotyczących budowy takich - dedykowanych systemów komunikacji jest kwestią umowną. Ja natomiast postarałem się skupić na sprawdzonych, powiedziałbym nawet kuloodpornych rozwiązaniach.

Słowem końcowym, nie możemy zapomnieć o sytuacji geopolitycznej, obecne działania na terytorium Ukrainy pokazują nam jak ważna jest komunikacja (choćby na względnie krótki dystans), a moim osobistym życzeniem jest zapewnienie do niej dostępu wszystkim, w miarę przystępnej realizacji.

Słowa kluczowe: *Android, raspberry pico, mikrokontroler, lora*

Abstract

Title: Dedicated communication system using Lora protocol

Currently available communication systems are using Internet, to the great extent. What if there is no Internet? I am going to answer to this question in my thesis.

In my work, I have tried to introduce various aspects of the construction, implementation, or operation of communication systems.

My system is composed (from the side of) software of two components: application working on microcontroller and mobile application written to operate on devices with Android operation system.

On the hardware side, however, it is a raspberry pico microcontroller, a Bluetooth module and a module for communication at sub-1 GHz frequency

Many issues concerning the construction of such - dedicated communication systems are a matter of convention. I, however, have tried to focus on proven, I would even say bulletproof solutions.

In a final word, we must not forget about the geopolitical situation, the current actions on Ukrainian territory show us how important communication is (if only for a relatively short distance), and my personal wish is to provide access to it for all, with a relatively affordable implementation.

Keywords: *Android, raspberry pico, microcontroller, lora.*

Tę kartkę należy zastąpić oświadczeniem o autorstwie pracy.

Spis treści

Spis treści	1
1 Wstęp	3
1.1 Tło	3
1.2 Cel	3
1.3 Szablon	3
2 Tło teoretyczne	5
2.1 Mikrokontroler	5
2.2 Lora	5
2.3 Bluetooth LE	5
2.4 Micropython	6
3 Projekt urządzenia i oprogramowania	7
3.1 Schemat systemu	7
4 Badanie efektywności	9
4.1 Symulacja numeryczna	9
4.2 Badanie terenowe	10
5 Podsumowanie	13
Bibliografia	15
Spis rysunków	17
Spis tabel	19
Spis załączników	21

1 Wstęp

1.1 Tło

Podstawą dla rozwoju społeczeństwa niewątpliwie jest przekazywanie wiedzy, ale jaką możliwość przekazywania wiedzy mielibyśmy bez komunikacji. Najnowsze badania [5] wskazują, że jej rozwój nastąpił w okresie od 20 milionów lat temu do 200 tys. lat temu. To bardzo dawno temu. Natomiast, telegraf [2] wynaleziono raptem w XVIII w. Na koniec tego wieku możemy datować początek komunikacji długodystansowej. Wolno, wręcz powolnie i kilkaset lat później mamy już Internet [3], którego początki datujemy na lata 60 XX w. Następnym etapem rozwoju tej pięknej dziedziny jest niewątpliwie zwiększanie przepustowości światłowodów, które następuje skokowo jak i przygotowanie do kolonizacji odległych (na początek tylko w obrębie naszego Układu Słonecznego) planet z którymi oczywiście musimy utrzymywać kontakt.

1.2 Cel

Kontakt, stanowi on sedno niniejszej pracy. Pragnę w jej ramach przygotować, przetestować, opisać i zaprezentować system komunikacji. Mój system komunikacji ma działać na odległości do 5-10 km w idealnych warunkach, przy wykorzystaniu technologii Lora [8] w paśmie poniżej 1 GHz. Jest to zadanie wymagające pod względem technicznym, jak i praktycznym. Trzeba zgrać ze sobą różne moduły, przygotować funkcjonalną obudowę, stworzyć aplikację mobilną i napisać kod na mikrokontroler, a na koniec wszystko przetestować i opisać. Cały system musi działać niezawodnie, bo w jaki inny sposób możemy mówić o komunikacji, jeśli nie jest ona niezawodna. Mamy wtedy raptem nie w pełni sprawną protezę kompletnego rozwiązania, czego bardzo nie chcę i poprzez odpowiednie testy udowodnię, że moje rozwiązanie jest kompletne od strony zarówno hardware'u jak i software'u.

1.3 Szablon

Niniejsza praca podzielona jest na następujące rozdziały i podrozdziały:

1. Wstęp - omówienie celu jak i zakresu pracy.
2. Tło teoretyczne - omówienie koncepcji i technologii używanych przez mikrokontroler, moduł lora i moduł bluetooth.
3. Projekt urządzenia i oprogramowania - zaprezentowanie narzędzi i komponentów wykorzystanych do stworzenia od strony hardware'owej jak i software'owej systemu i jego kluczowych funkcji.
4. Badanie efektywności - sprawdzenie efektywności zaprezentowanego rozwiązania
5. Podsumowanie - podsumowanie pracy
6. Bibliografia
7. Spisy rysunków, tabel i załączników (tutaj znajdują się kody źródłowe)

2 Tło teoretyczne

2.1 Mikrokontroler

Mikrokontroler który wykorzystano to Raspberry Pico (RP2040) [1] w rozmiarze (z ang. form factor) 21 mm x 51 mm, z dwu-rdzeniowym procesorem Arm Cortex-M0+, z zegarem o maksymalnym taktowaniu 133 MHz. Ten "mini-komputer" posiada 264 kB SRAM-u i 2 MB pamięci QSPi, 26 wielofunkcyjnych pinów GPIO, włączając w to 3 wejścia analogowe. Ponadto co jest szczególnie ważne w kwestii przyłączania zewnętrznych modułów mikrokontroler wyposażono w 2 UART, i co mniej ważne 2 SPI, 2 I2C i 16 kanałów PWM. Do wgrywania programów udostępniono kontroler USB w wersji 1.1, z opcją hosta. Obsługiwane napięcie wejściowe to od 1,8 V do 5,5 V DC. Temperatura pracy to od -20 st. C do +85 st. C.

2.2 Lora

Jak podaje artykuł naukowy [4] LoRa to bezprzewodowy system telekomunikacyjny o dużym zasięgu, małej mocy i niskiej przepływności, promowany jako rozwiązanie infrastrukturalne dla Internetu rzeczy: urządzenia końcowe wykorzystują LoRa w pojedynczym przeskoku bezprzewodowym, aby komunikować się z bramą (bramkami), podłączonymi do Internetu, które działają jako przezroczyste mosty i przekazują wiadomości między tymi urządzeniami końcowymi a centralnym serwerem sieciowym. W artykule przedstawiono przegląd LoRa i dogłębną analizę jego funkcjonalnych komponentów. W moim zastosowaniu nie będzie typowych bram, będą po prostu 2 urządzenia działające przy wykorzystaniu tego systemu. LoRa jest ukierunkowana na zastosowania, w których urządzenia końcowe mają ograniczoną ilość energii (na przykład zasilane z baterii), w których urządzenia końcowe nie muszą przysyłać więcej niż kilka bajtów na raz i w których ruch danych może być inicjowany przez urządzenie końcowe lub przez podmiot zewnętrzny, który chce się z nim skomunikować. Charakter dalekiego zasięgu i niskiego poboru mocy LoRa sprawia, że jest to idealny kandydat do wykorzystania w tym projekcie. LoRa zapewnia komunikację na duże odległości do 5 km w obszarach miejskich i do 15 km w obszarach wiejskich (w linii wzroku). Protokół ten umożliwia tworzenie urządzeń, które na zasilaniu bateryjnym mogą działać nawet przez 10 lat.

2.3 Bluetooth LE

Jest to bezprzewodowa sieć osobista (PAN), zaprojektowana i stworzona przez Bluetooth Special Interest Group. Wykorzystuje się ją w opiece zdrowotnej, w branży fitness, w beaconach, bezpieczeństwie i urządzeniach domowej rozrywki. Jest niezależna od klasycznego Bluetooth i nie jest z nim kompatybilna, jednakże te dwie technologie mogą współdziałać w ramach pojedynczego urządzenia. Sieć ta działa na częstotliwości 2,4 GHz, tak jak klasyczny Bluetooth. Nominalny zasięg to poniżej 100 m, prędkość od 125 kbit/s do 2 Mbit/s, ilość urządzeń typu slave zależy od implementacji, do zabezpieczenia transmisji wykorzystywany jest 128-bitowy AES. Sieć ta posiada aktywny frequency hopping, leniwe wiązanie, 24-bitowy klucz CRC i 32-bitowe sprawdzanie integralności wiadomości. Ze stanu niepołączonego wybudza się w 6 ms, natomiast minimalny czas na wysłanie wiadomości to 3 ms. Dostępna topologia to z ang. Scatternet. Moc w zależności od przypadku użycia to 0,01-0,50 W. Szczytowe natężenie prądu w czasie pracy to mniej niż 15 mA. Główne zastosowania to telefony mobilne, gaming, inteligentny dom, urządzenia typu wearables, motoryzacja,

komputery, bezpieczeństwo, urządzenia zbliżeniowe, ochrona zdrowia, sport, fitness i zastosowania przemysłowe [6].

2.4 Micropython

Jest to podzbiór języka Python 3 przeznaczony do zastosowania w środowiskach o ograniczonych możliwościach pod względem pamięci operacyjnej, mocy procesora czy też maksymalnego zużycia energii elektrycznej. Nie zaimplementowano w nim całego standardu Python 3, jednak jest to dość obszerna implementacja. Nie posiada on własnego środowiska programistycznego (z ang. Integrated Development Environment), zamiast tego zarówno do jego instalacji jak i debugowania stworzonych w nim programów wykorzystuje się aplikację Thonny. Jest on zoptymalizowany do wykorzystywania na mikrokontrolerach takich jak np. Raspberry Pi Pico, Arduino lub inne. Kod źródłowy tego oprogramowania dostępny jest na licencji MIT. [7]

3 Projekt urządzenia i oprogramowania

3.1 Schemat systemu

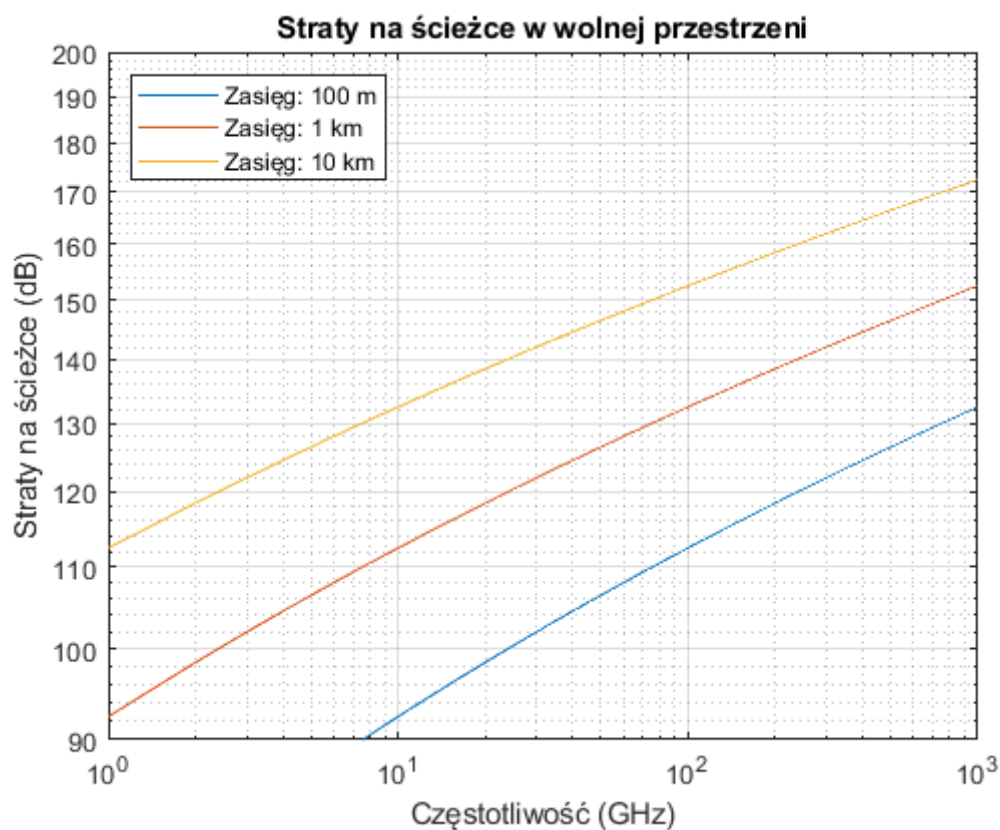
Mikrokontroler wkładany jest do gniazda w płytce z modułem Lora, wykorzystuje to 1 UART, do odrębnego UART-u podpinany jest z kolei moduł Bluetooth LE, którego wadą jest brak obsługi tzw. bonding (modułu nie można powiązać przy użyciu PIN-u z pojedynczym urządzeniem, co jest pewną niedogodnością pod względem bezpieczeństwa). Poniżej znajduje się grafika przedstawiająca urządzenie, jeszcze bez obudowy i zasilania bateryjnego.

4 Badanie efektywności

Badanie efektywności odbyło się dwójako. Przy użyciu symulacji numerycznej w programie Matlab, jak i poprzez eksperymenty w terenie.

4.1 Symulacja numeryczna

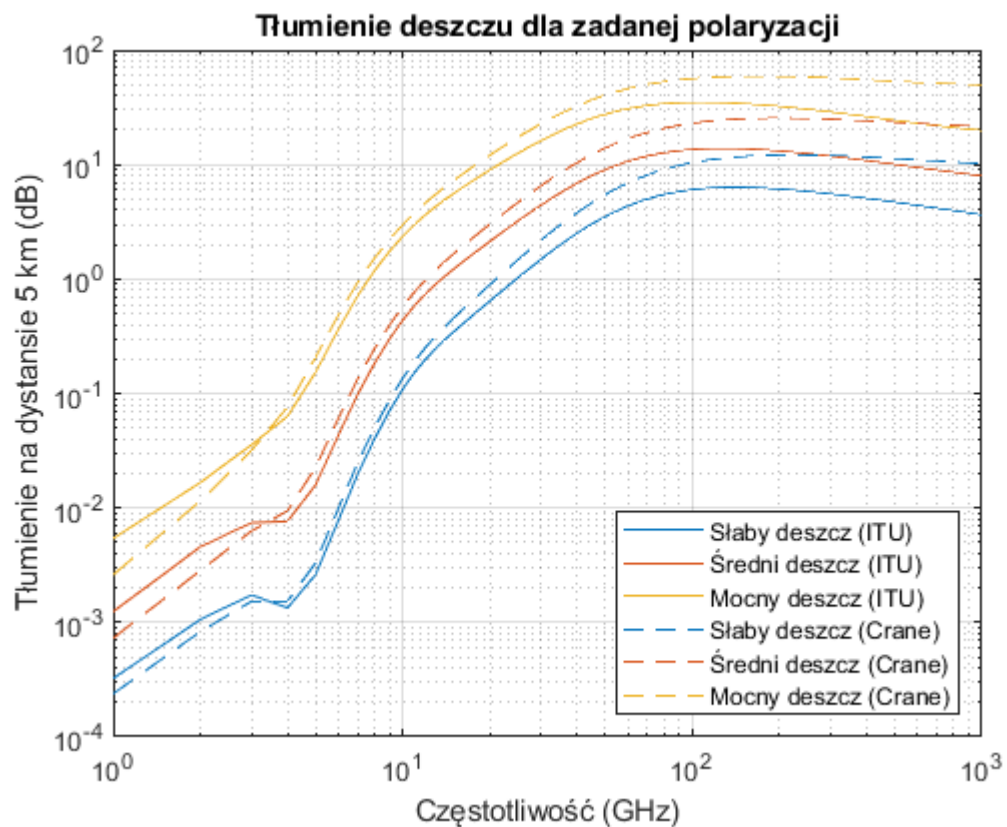
Wszystkie kody symulacji wykorzystane w pracy znajdują się w dodatku C. Na poniższej grafice widać jak propaguje się sygnał na ścieżce wolnej, interesuje nas w szczególności część wykresu dla częstotliwości w okolicach 1 GHz, czyli takiej jaką wykorzystuje protokół Lora.



Rys. 4.1. Straty na ścieżce w wolnej przestrzeni

W rzeczywistości jednak sygnały nie poruszają się w próżni, więc straty na ścieżce wolnej opisują tylko część tłumienia sygnału. Sygnały interferują z cząsteczkami w powietrzu i tracą energię na drodze propagacji. Straty różnią się w zależności od różnych czynników takich jak: ciśnienie, temperatura, opad atmosferyczny, rodzaj i gęstość opadu, zachmurzenie lub jego brak.

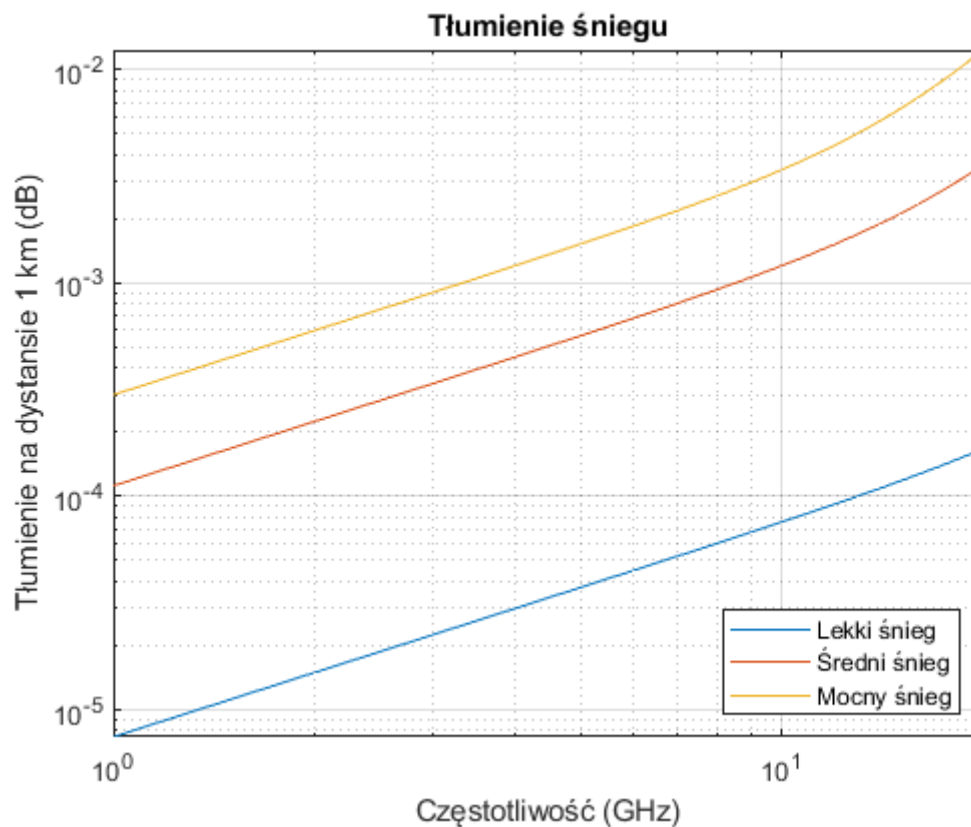
Poniżej znajduje się wykres przedstawiający tłumienie podczas opadu deszczu, na dystansie 5 km. Interesuje nas w szczególności część wykresu na jego początku, właśnie w takim zakresie częstotliwości, w jakim działa Lora.



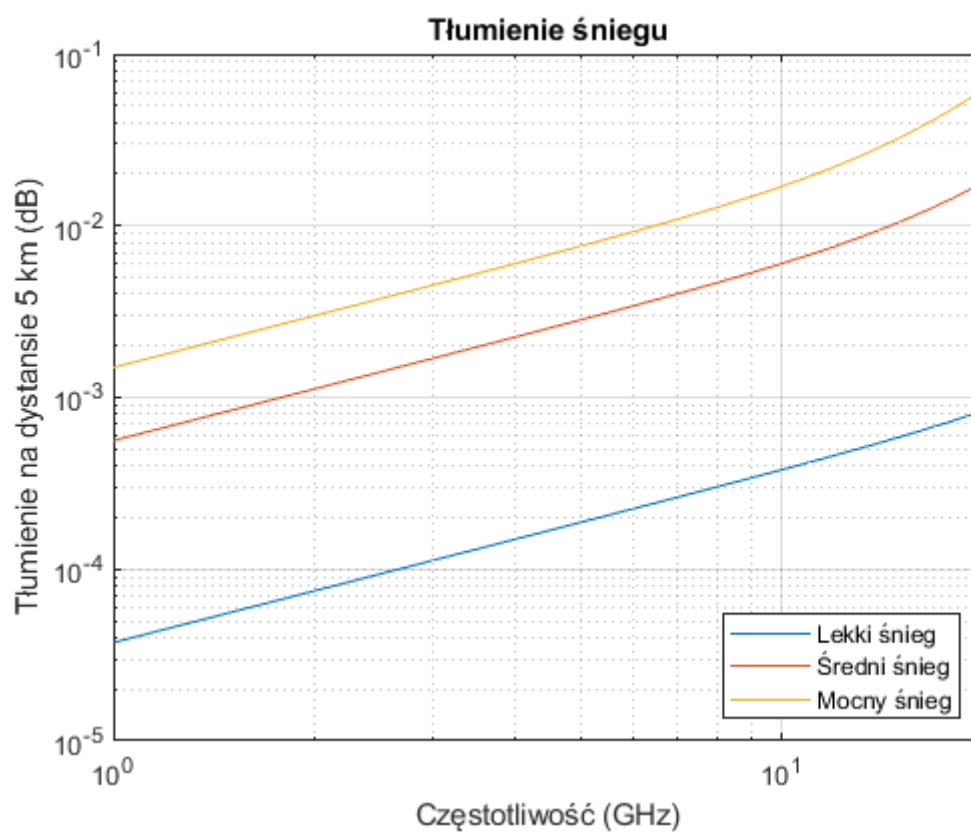
Rys. 4.2. Tłumienie deszczu dla zadanej polaryzacji na dystansie 5 km

Ponownie, poniżej wykresy, tym razem podczas opadu atmosferycznego w postaci śniegu, w 3 stopniach nasilenia, dla 2-óch odległości.

4.2 Badanie terenowe



Rys. 4.3. Tłumienie śniegu na dystansie 1 km



Rys. 4.4. Tłumienie śniegu na dystansie 5 km

5 Podsumowanie

Udało się osiągnąć zadawalające rezultaty zarówno pod względem zużycia energii elektrycznej, co w tym wypadku jest kluczowe albowiem mamy do czynienia z urządzeniem zasilanym z baterii, jak i pod względem tzw. user experience. Z testów wynika, że urządzenie są w stanie obsługiwać osoby o różnej sprawności, praktycznie bez względu na wiek użytkownika. Jeżeli ktoś jest w stanie obsłużyć smartfon, to bez problemu poradzi sobie z systemem komunikacji zaprezentowanym w tej pracy.

Bibliografia

- [1] Raspberry pi pico(rp2040). URL: <https://www.raspberrypi.com/products/raspberry-pi-pico/>.
- [2] Telegraf. [online], marzec 2022. [dostęp: 2022-10-31 14:00Z].
- [3] Internet. [online], wrzesień 2022. [dostęp: 2022-10-31 14:07Z].
- [4] Aloÿs Augustin, Jiazi Yi, Thomas Clausen, and William Townsley. A study of LoRa: Long range & low power networks for the internet of things. 16(9):1466. doi:10.3390/s16091466.
- [5] Louis-Jean Boë, Thomas R. Sawallis, Joël Fagot, Pierre Badin, Guillaume Barbier, Guillaume Captier, Lucie Ménard, Jean-Louis Heim, and Jean-Luc Schwartz. Which way to the dawn of speech?: Reanalyzing half a century of debates and data in light of speech science. *Science Advances*, 5(12):eaaw3916, 2019. URL: <https://www.science.org/doi/abs/10.1126/sciadv.aaw3916>, arXiv:<https://www.science.org/doi/pdf/10.1126/sciadv.aaw3916>, doi:10.1126/sciadv.aaw3916.
- [6] Wikipedia contributors. Bluetooth low energy — Wikipedia, the free encyclopedia. [Online; accessed 7-April-2024]. URL: https://en.wikipedia.org/w/index.php?title=Bluetooth_Low_Energy&oldid=1211216789.
- [7] Wikipedia contributors. Micropython — Wikipedia, the free encyclopedia. <https://en.wikipedia.org/w/index.php?title=MicroPython&oldid=1214292564>. [Online; accessed 8-April-2024].
- [8] Wikipedia contributors. Lora — Wikipedia, the free encyclopedia, 2022. [Online; accessed 31-October-2022]. URL: <https://en.wikipedia.org/w/index.php?title=LoRa&oldid=1112695195>.

Spis rysunków

4.1	Straty na ścieżce w wolnej przestrzeni	9
4.2	Tłumienie deszczu dla zadanej polaryzacji na dystansie 5 km	10
4.3	Tłumienie śniegu na dystansie 1 km	11
4.4	Tłumienie śniegu na dystansie 5 km	11

Spis tabel

Załączniki

Spis załączników

A	Tekst źródłowy programu na mikrokontroler	25
B	Kod źródłowy aplikacji na telefon z systemem operacyjnym Android	27
C	Kody źródłowe symulacji numerycznych w programie Matlab	29
C.1	Symulacja na ścieżce w przestrzeni wolnej	29
C.2	Symulacja na dystansie 5 km podczas opadu deszczu	29
C.3	Symulacja na dystansie 1 km podczas opadu śniegu	29
C.4	Symulacja na dystansie 5 km podczas opadu śniegu	30

A Tekst źródłowy programu na mikrokontroler

```
1  library ieee;
   use ieee.std_logic_1164.all;
   use ieee.std_logic_unsigned.all;
   use ieee.std_logic_arith.all;
5
   entity wave is
   port (
       clk, reset : in std_logic;
       t_l, t_h   : in std_logic_vector(31 downto 0);
10  w           : out std_logic);
   end wave;

   architecture rtl of wave is
   type FSM is (IDLE, PH, PL);
15  signal state, next_state: FSM;
   signal timer : std_logic_vector(31 downto 0);
   signal resetc: std_logic;

   begin
20  RS_PROC:
   process (clk, reset)
   begin
       if (reset='0') then
           state <= IDLE;
25  elsif (rising_edge(clk)) then
           state <= next_state;
       end if;
   end process;

30  NS_PROC:
   process (state, t_l, t_h, timer)
   begin
       case state is
35  when idle =>
           if (t_l = 0 or t_h = 0) then
               next_state <= idle;
           else
               next_state <= PH;
40  end if;

           when PH =>
               if (timer < t_h - 1) then
                   next_state <= PH;
45  else
                   next_state <= PL;
               end if;

           when PL =>
50  if (timer < t_l - 1) then
                   next_state <= PL;
               else
                   next_state <= PH;
               end if;
55  end case;
   end process;
```

```

DW_PROC:
60  process ( state )
    begin
        if      ( state = PH) then    w <= '1';
        elsif  ( state = PL) then    w <= '0';
        end if;
65  end process;
    end if;
    end process;
    end rtl;

```

B Kod źródłowy aplikacji na telefon z systemem operacyjnym Android

```
1  library ieee;
   use ieee.std_logic_1164.all;
   use ieee.std_logic_unsigned.all;
   use ieee.std_logic_arith.all;
5
   entity shift_reg is
   generic(N: integer range 0 to 32 := 8);
   port (
   clk, reset, load : in std_logic;
10  pos, reg_in      : in std_logic_vector(N-1 downto 0);
   reg_out          : out std_logic_vector(N-1 downto 0));
   end shift_reg;

   architecture rtl of shift_reg is
15  signal rejestr: std_logic_vector(N-1 downto 0);

   begin
   process (clk, reset, load, reg_in)
   begin
20     if (reset = '0') then
         rejestr <= (others => '0');
         elsif (load = '0') then
             rejestr <= reg_in;
             elsif (rising_edge(clk)) then
25         rejestr (N-1 downto conv_integer(pos)) <= rejestr(N-1-conv_integer(pos) downto 0);
             rejestr (conv_integer(pos) downto 0) <= (others => '0');
         end if;
     end process;

30  reg_out <= rejestr;
   end rtl;
```


C Kody źródłowe symulacji numerycznych w programie Matlab

Wszystkie kody źródłowe, z drobnymi zmianami i polonizacją pochodzą stąd: <https://www.mathworks.com/help/radar/ug/modeling-the-propagation-of-rf-signals.html>

C.1 Symulacja na ścieżce w przestrzeni wolnej

```
1 c = physconst('lightspeed');
  R0 = [100 1e3 10e3];
  freq = (1:1000).'*1e9;
  apathloss = fspl(R0,c./freq);
5 loglog(freq/1e9, apathloss);
  grid on;
  ylim([90 200]);
  legend('ęZasig: 100km', 'ęZasig: 1km', 'ęZasig: 10km', 'Location', 'northwest');
  xlabel('ęCzstotliwo (GHz)');
10 ylabel('Straty na ścieżce (dB)');
  title('Straty na ścieżce w wolnej przestrzeni');
```

C.2 Symulacja na dystansie 5 km podczas opadu deszczu

```
1 R0 = 5e3; % 5 km ęzasig
  rainrate = [1 4 20]; % łsia opadu w mm/h
  el = 0; % 0 stopni wznoszenia
  tau = 0; % polaryzacja horyzontalna
5 for m = 1:numel(rainrate)
  rainloss_itu(:,m) = rainpl(R0,freq,rainrate(m),el,tau)';
  rainloss_crane(:,m) = cranerainpl(R0,freq,rainrate(m),el,tau)';
end
10 loglog(freq/1e9, rainloss_itu);
  hold on;
  set(gca, 'ColorOrderIndex', 1); % reset color index for better comparison
  loglog(freq/1e9, rainloss_crane, '--');
  hold off;
15 grid on;
  legend('łSaby deszcz (ITU)', 'Średni deszcz (ITU)', 'Mocny deszcz (ITU)', ...
  'łSaby deszcz (Crane)', 'Średni deszcz (Crane)', 'Mocny deszcz (Crane)', ...
  'Location', 'SouthEast');
  xlabel('ęCzstotliwo (GHz)');
20 ylabel('łTumienie na dystansie 5km (dB)');
  title('łTumienie deszczu dla zadanej polaryzacji');
```

C.3 Symulacja na dystansie 1 km podczas opadu śniegu

```
1 freq = (1:20)*1e9;
  R0 = 1e3; % 1 km dystansu
  snowrate = [0.1 1.5 4]; % odpowiednik opadu deszczu mm/h
```

```

5  for m = 1:numel(snowrate)
    snowloss(:,m) = snowpl(R0,freq,snowrate(m));
    end
    loglog(freq/1e9,snowloss);
    grid on;
10 legend('Lekki_śnieg','Średni_śnieg','Mocny_śnieg', ...
    'Location','SouthEast');
    xlabel('ęćCzstotliwo_(GHz)');
    ylabel('Tumienie_na_dystansie_1_km_(dB)')
    title('Tumienie_śniegu');

```

C.4 Symulacja na dystansie 5 km podczas opadu śniegu

```

1  freq = (1:20)*1e9;
    R0 = 5e3; % 5 km dystansu
    snowrate = [0.1 1.5 4]; % odpowiednik opadu deszczu mm/h

5  for m = 1:numel(snowrate)
    snowloss(:,m) = snowpl(R0,freq,snowrate(m));
    end
    loglog(freq/1e9,snowloss);
    grid on;
10 legend('Lekki_śnieg','Średni_śnieg','Mocny_śnieg', ...
    'Location','SouthEast');
    xlabel('ęćCzstotliwo_(GHz)');
    ylabel('Tumienie_na_dystansie_1_km_(dB)')
    title('Tumienie_śniegu');

```