

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 Schemat systemu	6
3 Urządzenie zbudowane w oparciu o mikrokontroler	7
3.1 Przegląd różnych mikrokontrolerów i porównanie, a także przedstawienie wybranego rozwiązania	7
3.2 Schemat	7
3.3 Przegląd gotowych rozwiązań z zakresu ładowania i magazynowania energii w tego typu urządzeniach	7
3.4 Konkretnie rozwiązanie "problemu zasilania" zastosowane w moim projekcie	7
3.5 Obudowa	7
3.6 Przedstawienie protokołu Lora	7
4 Aplikacja na mikrokontroler	9
4.1 Wstęp	9
4.2 Algorytm	9
4.3 Kod aplikacji	9
4.4 Krytyczna analiza aplikacji	9
5 Aplikacja mobilna	11
5.1 Wprowadzenie - użyte narzędzia	11
5.2 Aplikacja - kod, opis, działanie	11
5.3 Krytyczna analiza aplikacji mobilnej	11
6 Efektywność zastosowanego rozwiązania	13
6.1 Przegląd sposobów badania efektywności	13
6.2 Faktyczne badanie	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 [6] 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.

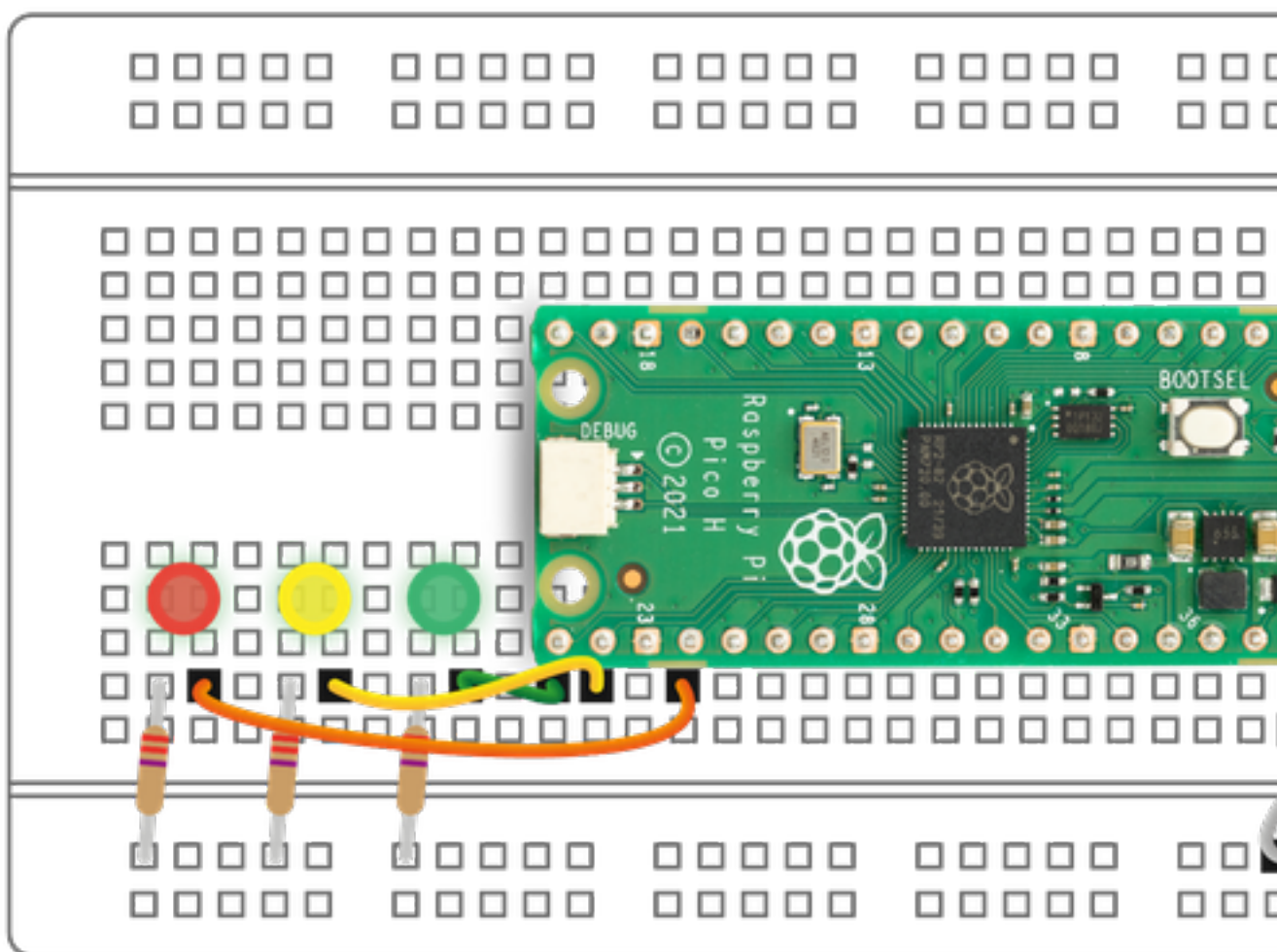
1.3 Szablon

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

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 Schemat systemu

3 Urządzenie zbudowane w oparciu o mikrokontroler

3.1 Przegląd różnych mikrokontrolerów i porównanie, a także przedstawienie wybranego rozwiązania

3.2 Schemat

3.3 Przegląd gotowych rozwiązań z zakresu ładowania i magazynowania energii w tego typu urządzeniach

3.4 Konkretne rozwiązanie "problemu zasilania" zastosowane w moim projekcie

3.5 Obudowa

3.5.1 Przegląd różnych technik wykonania obudów do podobnych urządzeń

3.5.2 Przedstawienie wybranego rozwiązania - druk 3D

3.5.3 Schemat obudowy wykonany w oprogramowaniu typu CAD

3.6 Przedstawienie protokołu 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.

4 Aplikacja na mikrokontroler

4.1 Wstęp

4.2 Algorytm

4.3 Kod aplikacji

4.4 Krytyczna analiza aplikacji

5 Aplikacja mobilna

5.1 Wprowadzenie - użyte narzędzia

5.2 Aplikacja - kod, opis, działanie

5.3 Krytyczna analiza aplikacji mobilnej

6 Efektywność zastosowanego rozwiązania

6.1 Przegląd sposobów badania efektywności

6.2 Faktyczne badanie

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. 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

Spis tabel

Załączniki

Spis załączników

A	Tekst źródłowy programu	25
B	Model VHDL projektu	27

A Tekst źródłowy programu

```
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  elseif (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 Model VHDL projektu

```
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 (
10    clk, reset, load : in  std_logic;
       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;
```