

Dokumentacja projektu

# **Li-Fi – transmisja danych światłem**

Szymon Bortel, Damian Czajka

Elektronika 3 rok

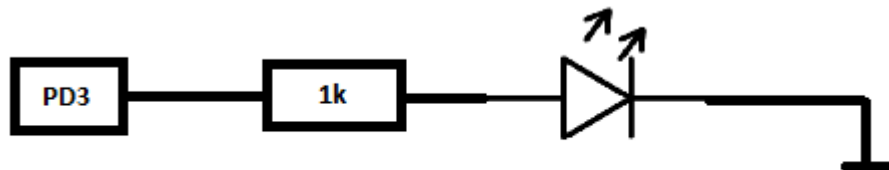
## 1. Opis funkcjonalny

Głównym zadaniem układu była transmisja danych przy pomocy światła widzialnego oraz podczerwonego emitowanego z diody LED. Nadajnik, korzystając z GPIO oraz wyprowadzenia PIN nadaje informacje na diodę. W skład ramki wchodzi: sekwencja startowa, końcowa oraz tablica danych, którą chcemy przesłać. Logiczna jedynka występuje, gdy dioda emituje światło, logiczne zero w przeciwnym wypadku. Dane kodowane są kodowaniem Manchester, które daje nam dużą odporność na zmiany szybkości transmisji. Dodatkowo eliminuje składową stałą. W nadajniku wykorzystaliśmy licznik PIT, który wysyła ramkę w czasie, który jest modyfikowalny przez użytkownika. Użytkownik może również zmieniać prędkość transmisji.

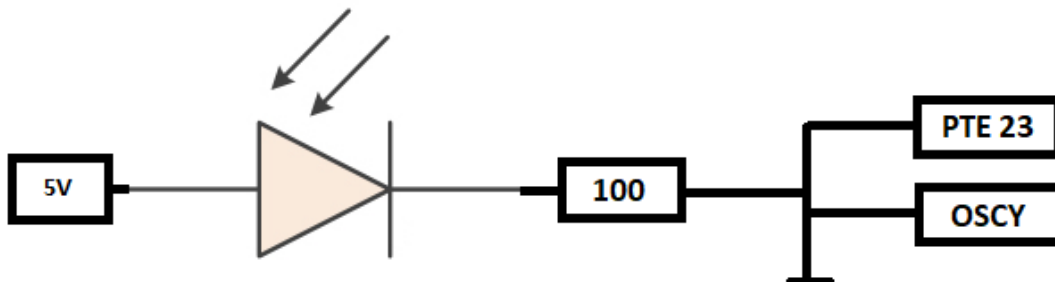
Odbiornik wykorzystuje fotodiodę, ADC, liczniki SysTick oraz PIT. Przetwornik odpowiedzialny jest za odczyt wartości napięć z rezystora znajdującego się za fotodiodą. Wykorzystywany jest w każdym etapie odbierania danych, przy odzyskiwaniu zegara, przy zdefiniowaniu napięcia referencyjnego oraz przy odbieraniu danych. Na podstawie danych odczytanych z przetwornika dane są interpretowane w funkcji – czy odbieramy logiczne „0” czy „1”. Wartości napięć są wyświetlane na wyświetlaczu LCD.

## 2. Schemat zaproponowanego przez nas układu do transmisji danych światłem:

Nadajnik:



Odbiornik:



### 3. W skład projektu wchodzi następujące pliki:

- Dla nadajnika:
  - Main.c
  - Transmitter.c
  - Leds.c
- Dla odbiornika:
  - Main.c
  - Receiver.c
  - Leds.c
  - Slcd.c

### 4. Opis plików nadajnika:

- **Main.c** – wykonuje jednorazowo 5 funkcji zdefiniowanych w transmitter.c, które opisane są poniżej. Brak pętli nieskończonej, działa tylko i wyłącznie na przerwaniu z PIT.
- **Leds.c** – obsługa led, wykorzystana w celach kontrolowania, kiedy kończy się ramka.
- **Transmitter.c** :
  - *Initialize\_transmitter()* – inicjalizacja zegara, ustawienie pinów jako GPIO, nałożenie masek
  - *filldatatab()* – uzupełnienie tablicy danych
  - *ramkaopen()* – uzupełnienie ramki początkowej sekwencją 1010
  - *ramkaclose()* – uzupełnienie ramki końcowej sekwencją 0101
  - *senddata()* – sprawca wartości w tablicach, następnie koduje je w kodzie Manchester
  - *initialize\_clockt()* – inicjalizacja CLK dla modułu PIT
  - *initialize\_pitt()* – inicjalizacja PIT z nadpisaniem wartości LDVAL
  - *delay\_mc(uint32\_t value)* – opóźnienie o value ms.
  - *PIT\_IRQHandler()* – senddata, gdy skończy się zliczanie w dół wartości LDVAL, czyszczenie flag

### 5. Opis plików odbiornika

- **Receiver.c**
  - *initializeReceiver()* – inicjalizacja niezbędnych pinów – dołączenie zegara, ustawienie multiplexera oraz masek,
  - *ADC\_init()* – inicjalizacja przetwornika ADC,
  - *getVoltage()* – odczytana wartość napięcia w Voltach przechowywana w zmiennego voltage
  - *initialize\_pit()* – inicjalizacja licznika PIT, ustawienie wartości LDVAL

- *compare()* – ustalanie napięcia referencyjnego na podstawie danych wcześniej zgromadzonych
- *clkrestore()* – funkcja odzyskująca zegar – na początku transmisji odzyskuje zegar. Zlicza czas do momentu logicznej jedynki
- *openingframe()* – funkcja wyświetla dane odbierane z nadajnika, dodatkowo, gdy wystąpi sekwencja ramki początkowej zaświeca zieloną diodę
- *data()* – odbiera dane, wyświetla je na czerwonej diodzie
- *closeframe()* – gdy występuje sekwencja kończąca ramkę gasi zieloną diodę
- *receive()* – wykonuje wcześniej opisane funkcje – *getVoltage*, *compare*, *openingframe*, *data*, *closeframe*
- *PIT\_IRQHandler* – służy do ustalania napięcia referencyjnego co określony przez programistę czas (wartość LDVAL) oraz służy do odzyskiwania zegara
- **Main.c** – składa się z inicjalizacji PIT, ADC, odbiornika, wyświetlacza LCD. Dodatkowo wyposażony jest w SysTick, który wyświetla wartość napięcia na wyświetlaczu LCD z określoną przez programistę częstotliwością. W pętli nieskończonej znajduje się opisana wyżej funkcja *receive*.
- **Leds.c** – obsługa diód LED
- **Slcd.c** – obsługa wyświetlacza LCD

## 6. Wykorzystane elementy:

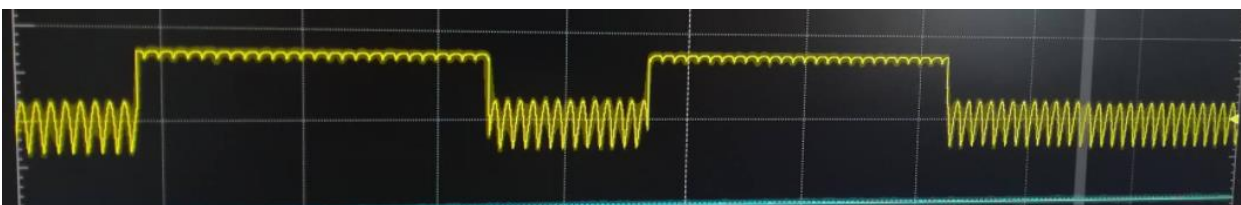
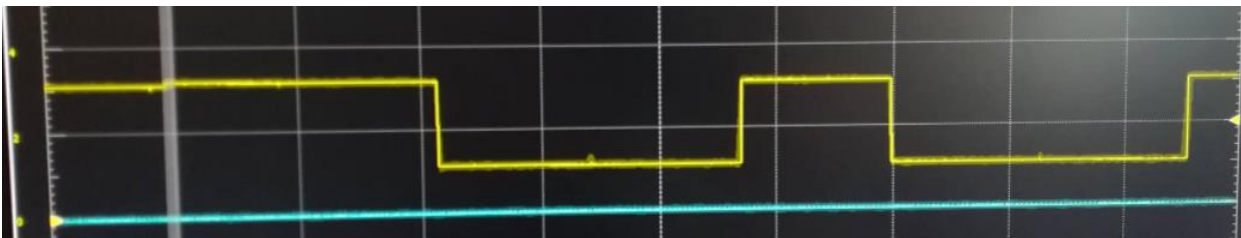
- 2x KL46,
- fotodioda SFH206K,
- diody LED różnych kolorów,
- dioda HSDL4220 (podczerwień),
- rezystory 100 i 1k.

## 7. Wnioski, porównanie z założeniami teoretycznymi

W trakcie projektu spotkaliśmy się z kilkoma problemami – pierwszym z nich była transmisja danych przy użyciu jednej płytki – chcą odbierać i wysyłać dane jednocześnie bez użycia dodatkowych funkcji dostępnych na płytce (DMA) tracilibyśmy dane. Finalnie postanowiliśmy zrealizować projekt na dwóch osobnych płytkach, gdzie jedna jest odpowiedzialna za nadawanie a druga za odbieranie danych.

Drugim problemem było ustalanie napięcie referencyjnego oraz odzyskiwanie zegara w taki sposób, aby nie tracić danych – udało nam się rozwiązać ten problem przy użyciu przerwań.

Trzecim problemem były zakłócenia spowodowane światłem zewnętrznym. Gdy natężenie światła zewnętrznego było zbyt duże poziom napięcia referencyjnego nie ustalał się prawidłowo.



Zdjęcie powyżej przedstawia wpływ świetłówki na sygnał

#### 8. Datasheety użytych elementów:

- <https://www.tme.eu/pl/Document/8515b57d717bb959693dac540126ef61/SFH206K.pdf>
- <https://www.tme.eu/pl/Document/5599a5d144a1f762f5969adda9005080/HSDL-4220.pdf>