

PROJEKT

STEROWNIKI ROBOTÓW

---

## Raport końcowy

Świejące opaski na nadgarstek zgodne z  
CIR wraz z kontrolerem do zastosowań  
scenicznych

[ PMod ]

---

*Skład grupy:*

Maciej BOWSZYS, 278353

*Termin:* wtTN15

*Prowadzący:*

dr inż. Wojciech DOMSKI

3 lutego 2026

## Spis treści

<b>1</b>	<b>Opis ogólny</b>	<b>2</b>
<b>2</b>	<b>Moduł opaski</b>	<b>2</b>
2.1	Zasada działania . . . . .	2
2.2	Projekt elektroniki . . . . .	3
2.3	Protokół komunikacyjny . . . . .	3
2.4	Realizacja . . . . .	4
2.4.1	IR . . . . .	4
2.4.2	Interpretacja ramek . . . . .	4
2.4.3	Odtwarzanie animacji . . . . .	4
2.4.4	LED . . . . .	5
2.4.5	UART . . . . .	5
2.4.6	Stanowisko robocze - ATTiny45 . . . . .	5
2.4.7	Płytką Drukowaną . . . . .	6
2.5	Testy . . . . .	7
<b>3</b>	<b>Kontroler</b>	<b>8</b>
3.1	Zasada działania . . . . .	8
3.2	Projekt elektroniki . . . . .	8
3.3	Realizacja . . . . .	9
3.3.1	IR . . . . .	9
3.3.2	Art-Net . . . . .	9
3.3.3	Obudowa . . . . .	9
3.4	Testy . . . . .	10
<b>4</b>	<b>Harmonogram prac</b>	<b>10</b>
4.1	Postęp prac . . . . .	11
4.1.1	Komentarz do postępu prac . . . . .	12
	<b>Bibilografia</b>	<b>13</b>

## 1 Opis ogólny

Od kilku lat coraz częściej, podczas największych tras koncertowych świata wśród konwencjonalnego wyposażenia technicznego - nagłośnienia, oświetlenia ruchomego i statycznego - pojawia się oświetlenie osobiste (rys. 1). Są to albo świecące przedmioty (zwykle w formie "berła") lub świecące opaski. O tyle, o ile "berła" łączą się zwykle z telefonem użytkownika przez Bluetooth, zapewniając synchronizację z pozostałym oświetleniem przez internet, o tyle opaski są standardowo zdalnie sterowane bezpośrednio poprzez centralny nadajnik znajdujący się na terenie wydarzenia. Zależnie od producenta, napotkamy różne metody komunikacji - urządzenia PixMob<sup>1</sup> korzystają z podczerwieni, CrowdLED<sup>2</sup> z komunikacji radiowej etc.

Podczerwień wbrew pozorom, jest w tych okolicznościach dobrym rozwiązaniem - prosty protokół komunikacyjny, proste konstrukcyjnie i tanie układy odbiorcze i małe zajęcie medium dają dużą swobodę projektową urządzeń i organizacyjną wydarzeń. Z tego właśnie powodu zdecydowałem się, przygotowując swoje opaski tego typu, idąc w ślady PixMob, użyć do rozgłaszania komunikatów podczerwieni - stąd (od PixMob) skrócona nazwa projektu PMod.



Rysunek 1: Opaski PixMob wykorzystane podczas koncertu Imagine Dragons w 2018r.<sup>a</sup>

<sup>a</sup>File:Imagine Dragons at LoveLoud 2018 (43364933435).jpg [Wikimedia.org]

## 2 Moduł opaski

### 2.1 Zasada działania

Opaska składa się z paska (opcjonalny), białej, częściowo przepuszczającej światło obudowy i płytki drukowanej z odpowiednimi elementami elektronicznymi i bateriami. W zamierzonym zastosowaniu nie będzie potrzeby wyłączania układu, więc aby nie dopuścić do przypadkowego

<sup>1</sup>PixMob: Immersive event lighting solutions for fan-favorite experiences

<sup>2</sup>Crowdled: Light Up Your Crowd

wyłączenia, na płytce nie znajduje się wyłącznik. Podczas pracy układ nieustannie nasłuchuje na linii danych z odbiornika podczerwieni w oczekiwaniu na przybywające komunikaty [2]. Komunikat zawiera kod efektu oświetleniowego, dwa kody kolorów i sumę kontrolną. Po odebraniu pakietu polecenie w nim zawarte wykonywane jest w pętli aż do skutecznego odkodowania kolejnego komunikatu. Dwa kody kolorów pozwalają zaimplementować niektóre efekty stochastyczne.

## 2.2 Projekt elektroniki

Ze względu na przewidziane zastosowanie, moduł opaski składa się z:

- mikroprocesora - ze względu na małe rozmiary i niską cenę wybrany został układ ATTiny45-20SU [1]
- scalonego odbiornika podczerwieni - OSRB38C9BB [3]
- dwóch diod RGB - z przyczyn praktycznych będą to diody adresowane - RF-W7SA50TSA51-IC w obudowie 5050 [4]
- stabilizatora napięcia - w następstwie specyfikacji pozostałych elementów wybrane zostało napięcie 4V - układ AP2210N-4.0TRG1
- dwie baterie CR2032

Komunikacja z odbiornikiem podczerwieni odbywa się za pośrednictwem protokołu producenta - przypomina on sygnał PWM, jednak ze stałym czasem trwania stanu wysokiego. Komunikacja z diodami (szeregowo) odbywa się poprzez protokół WS2812B.

## 2.3 Protokół komunikacyjny

Przygotowany protokół komunikacyjny przewiduje przesył ramek danych o długości 14 bitów: 4 bity kodu efektu, 4 bity kodu koloru głównego, 4 bity koloru "tła", 2 bity sumy kontrolnej.

Efekty:

1. (0000) Światło ciągłe, kolor główny
2. (0001) Pulsowanie koloru głównego (do koloru tła i z powrotem), 0.33Hz
3. (0010) Pulsowanie koloru głównego (do koloru tła i z powrotem), 1Hz
4. (0011) Pulsowanie koloru głównego (do koloru tła i z powrotem), 2Hz
5. (0100) Pulsowanie koloru głównego (do koloru tła i z powrotem), 5Hz
6. (0101) Stroboskop, kolor główny (do koloru tła i z powrotem), 0.33Hz
7. (0110) Stroboskop, kolor główny (do koloru tła i z powrotem), 1Hz
8. (0111) Stroboskop, kolor główny (do koloru tła i z powrotem), 2Hz
9. (1000) Stroboskop, kolor główny (do koloru tła i z powrotem), 5Hz
10. (1001) Stroboskop, losowy kolor, 0.33Hz
11. (1010) Stroboskop, losowy kolor, 1Hz
12. (1011) Stroboskop, losowy kolor, 2Hz

13. (1100) Stroboskop, losowy kolor, 5Hz
14. (1111) Świecenie niezależne - po wydarzeniu

Zarówno paleta kolorystyczna kolorów głównych, jak i kolorów "tła" może być zdefiniowana osobno dla każdego wydarzenia.

Jako algorytm do wyznaczania sumy kontrolnej wybrany został wstępnie mechanizm Cyklicznej Kontroli Nadmiarowej<sup>3</sup> (CRC).

## 2.4 Realizacja

Pierwsze testy prototypu próbowałem przeprowadzić używając gotowych bibliotek do obsługi protokołów CIR i WS2812B. Okazało się jednak, że nawet ich minimalistyczne wersje są zbyt duże, aby zmieściły się w dostępnej pamięci ATTiny45 (4kB). Zastosowanie wymagało więc ręcznego napisania możliwie najprostszych i najszybszych modułów obsługujących te protokoły.

### 2.4.1 IR

Odczyt danych nadawanych przez podczerwień jest równoznaczny z pomiarem długości impulsów zwracanych przez scalony odbiornik podczerwieni. Impuls krótki to "0", a impuls długi to "1". Aby zrealizować to zadanie, naturalnym jest użycie licznika - tutaj jednak pojawia się problem - ATTiny45 ma tylko liczniki ośmiobitowe. Jak pokazały testy, nie wystarczają one do pomiaru parametrów sygnału z wystarczającą rozdzielczością. Skutecznym rozwiązaniem okazało się być połączenie dwóch takich liczników w "łańcuch". Pierwszy, szybki, licznik taktowany jest podzielonym przez osiem sygnałem zegarowym systemu, a drugi przerwaniami z pierwszego. Pozwala to na niezależny od głównej pętli programu pomiar czasu trwania impulsów.

Po pomiarze czasu trwania impulsu, zwrócona wartość jest klasyfikowana, jako "0", "1", błąd (impuls za krótki) lub koniec kodu (impuls bardzo długi) - one z kolei odpowiednio modyfikują wartość zmiennej code i - po poprawnym wykryciu końca kodu - wywołują funkcję interpretującą go.

### 2.4.2 Interpretacja ramek

Po skutecznym odebraniu komunikatu, usuwany jest ostatni bit (bit kontrolny będący zawsze jedyneką) oraz jest on dzielony na cztery segmenty: identyfikator efektu, identyfikator koloru głównego, identyfikator koloru tła i sumę kontrolną. [Obliczanie sumy kontrolnej nie zostało jeszcze zaimplementowane, przygotowane jest jednak po obu stronach komunikacji miejsce na jej obsługę.]

Po podziale komunikatu, na podstawie identyfikatorów efektu i kolorów, ustawiane są odpowiednie zmienne globalne mówiące o parametrach obecnie odtwarzanej animacji i wywoływane jest odświeżenie stanu animacji.

### 2.4.3 Odtwarzanie animacji

Z tego względu, że zastosowanie przewiduje nie tylko świecenie statyczne, ale też odtwarzanie prostych animacji, kod musiał zawierać okresowe wywoływanie funkcji odświeżającej stan diod. Jak wykazały testy, czas wykonania pętli głównej programu jest zbyt długi i zbyt zmienny, aby można było z jej poziomu wywoływać funkcję odświeżającą. Naturalnym rozwiązaniem byłoby tu generowanie przerw z licznika - my jednak wykorzystaliśmy już oba standardowe liczniki do

---

<sup>3</sup>Cykliczny kod nadmiarowy [Wikipedia]

obsługi modułu IR. Tu pojawia się nieco niekonwencjonalne rozwiązanie, które okazało się być bardzo skutecznym - do obsługi animacji w ATTiny użyty został WatchDog Timer. Pracuje on na sygnale taktującym o nieco niższej częstotliwości, ale w tym wypadku nie stanowi to problemu, ponieważ wymagana częstotliwość odświeżania animacji to raptem ok. 10Hz.

Podczas wywołania funkcji odświeżającej, sprawdzany jest aktualny typ efektu (STILL, STROBE, PLUSE, AUTO) i aktualny kolor główny i kolor tła. Typ efektu determinuje, w jaki sposób realizowane jest odświeżanie animacji (głównie sposób przechowywania stanu animacji - dla STROBE używamy stanu binarnego flip, a dla PLUSE zmiennej zmiennoprzecinkowej będącej postępowaniem animacji).

Implementacja obsługi animacji nie została dokończona ze względu na problemy z obsługą obliczeń zmiennoprzecinkowych być może wywołanych kończącą się przestrzenią w pamięci.

#### 2.4.4 LED

Protokół WS2812B przewiduje nadawanie komunikatów w formie trzech jednobajtowych wartości, gdzie "0" reprezentowane jest ponownie przez impuls krótki, a "1" przez impuls długi. W porównaniu z obsługą scalonego odbiornika podczerwieni, dużym problemem okazała się być standardowa częstotliwość pracy diod - wymagają one, aby impuls długi trwał typowo 900ns, a impuls krótki zaledwie 300ns. Mikrokontroler ATTiny45 obsługuje na szczęście taktowanie do 8MHz, co jest minimum niezbędnym do realizacji tego zadania. Ograniczona częstotliwość taktowania okazała się być problemem do tego stopnia, że standardowe funkcje oczekiwania (np. `_delay_us`) już nie wystarczyły - odnalezionym wyjątkowo ciekawym rozwiązaniem okazało się wstawienie pomiędzy ustawieniem pinu w stan wysoki, a przełączeniem w stan niski, bloku instrukcji Assembly - jedna instrukcja *nop* to jeden takt oczekiwania. Natomiast o dopasowaniu częstotliwości taktowania na styk świadczy choćby to, że do generowania impulsu krótkiego nie mogłem zastosować żadnego opóźnienia - sama dwukrotna zmiana stanu portu zajmuje ok. 250ns.

#### 2.4.5 UART

Podczas prac nad odczytem danych przez CIR, pojawiła się potrzeba możliwości odczytywania odbieranych kodów sygnałów CIR w celu debugowania dekodera. Układ ATTiny45 nie posiada jednak domyślnie kontrolera magistrali UART. Przygotowana została więc minimalistyczna implementacja UART pozwalająca na nadawanie komunikatów (z przekierowaniem `printf`) przez pin PB0 (współdzielony z linią MOSI programatora) - dzięki temu można było odczytywać nadawane przez mikroprocesor komunikaty.

#### 2.4.6 Stanowisko robocze - ATTiny45

Układ programowany jest przy pomocy klasycznego edytora tekstowego (tu: NeoVim), kompilatora `avr-gcc` i mikrokontrolera SeeedStudio Seeeduno Xiao ESP32-C3 z wgranym programem programatora ArduinoISP. Jako konwerter USB-UART użyty został drugi mikrokontroler Xiao ESP32-C3.

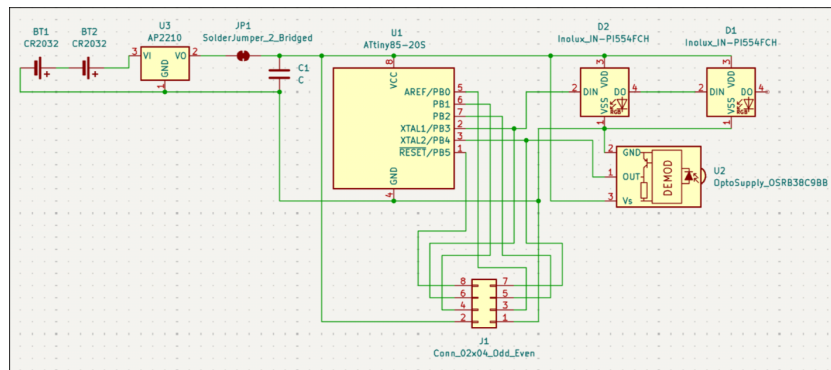
Poniżej widoczny jest wykaz pinów mikroprocesora wraz z nadanymi im funkcjami:

1. PB5 - Programator D6 (RESET)
2. PB3 - Diody Adresowane (DigOut)
3. PB4 - Odbiornik Podczerwieni (PCINT4)
4. GND

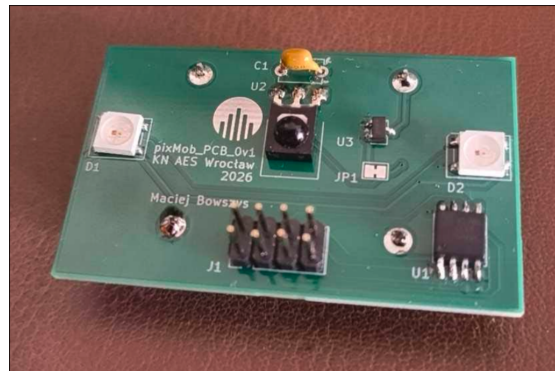
5. PB0 - Programator D10 (MOSI), soft\_UART (TX)
6. PB1 - Programator D9 (MISO)
7. PB2 - Programator D8 (SCK)
8. VCC

#### 2.4.7 Płytką Drukowaną

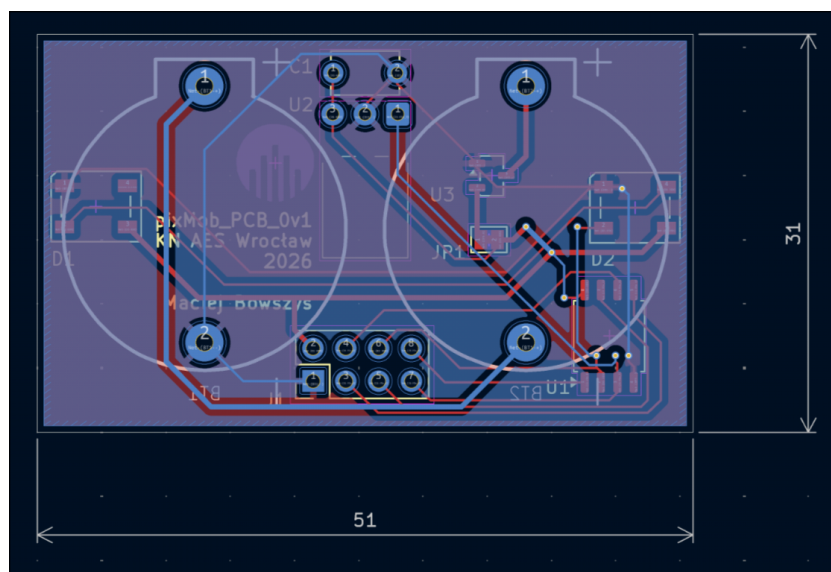
Płytką drukowaną powstała, jako idealne odzwierciedlenie układu elektronicznego przygotowanego w ramach prototypu (rys. 2). Aby umożliwić programowanie i ułatwić debugowanie, wszystkich osiem pinów układu ATTiny wyprowadzonych zostało na płytce w postaci złącz Gold-pin 2.54mm. Elementy elektroniczne zostały rozmieszczone tak, aby możliwie zmniejszyć objętość gotowego układu, aby noszenie opaski pozostało wygodnym. Na rys. 3 i rys. 4 zobaczyć można kolejno projekt płytki i zmontowany układ.



Rysunek 2: Schemat układu elektronicznego



Rysunek 4: Układ elektroniczny zmontowany na płytce drukowanej



Rysunek 3: Projekt płytki drukowanej

## 2.5 Testy

Do sprawdzenia poprawności działania mechanizmu odbierania komunikatów używane były podczas trwania projektu jednobarwne diody LED, diody LED adresowane oraz wyjście UART. Z testów wynika, że ok. 70% komunikatów odbierana jest poprawnie i w całości. Przy wczesnych testach z użyciem pilota na podczerwień, problemem był głównie pilot - po przytrzymaniu danego przycisku, w miarę upływu czasu zmieniały się powoli natężenie i częstotliwość sygnału. Problem ten zniknął w momencie przejścia na programowo generowane komunikaty.

Ze względu na niewielką rozdzielczość taktowania procesora, obsługa diod okazała się być wyzwaniem - nawet niewielkie fluktuacje napięcia zasilania i okresowe niekontaktowanie połączeń na płytce stykowej powodowało miganie diod.

Płytką drukowaną pomimo (sprawdzonej wielokrotnie) poprawności połączeń i pełnej funk-

cyjności peryferiów, nie mogła zostać przetestowana na czas, ponieważ z niezidentyfikowanych przyczyn niepowodzeniem kończyły się próby wgrania programu sterującego do nowych mikrokontrolerów.

Pomimo niemożności pełnego przetestowania układu w docelowej konfiguracji, udało się przeprowadzić test żywotności pracy przy zasilaniu bateryjnym. Przy pełnej jasności diod i światła ciągłym, moduł pracował poprawnie przez ok. 3h, później napięcie zaczęło powoli spadać, co przełożyło się na zmianę koloru światła (ze względu na różne spadki napięcia na diodach różnych kolorów składowych). Po ok. 10h diody nadal świeciły, ale już na wyraźny czerwono-pomarańczowy kolor.

Przeprowadzony został też uproszczony test zasięgu nadajnika - po warunkiem, że moduł opaski odwrócony był od nadajnika o nie więcej, niż ok. 80deg, komunikaty były poprawnie interpretowane nawet w odległości ok. 10m, większe odległości nie były testowane.

Zgodnie z wynikami testów, aby zachować skuteczną pracę, komunikaty nie powinny być nadawane częściej niż co ok. 400ms.

## 3 Kontroler

### 3.1 Zasada działania

Po podłączeniu do zasilania, układ automatycznie się uruchomi, określi swój adres IP i zacznie nasłuchiwać na wejściowym porcie komunikacyjnym - Z przyczyn praktycznych wybrane zostało połączenie Ethernet. Standardem branżowym komunikacji urządzeń scenicznych są protokoły DMX-512 i AtrNet. Pierwszy z nich jest protokołem szeregowym opartym na RS-485, a drugi korzysta z komunikacji w sieci LAN w trybie multicast za pomocą protokołu UDP. Są to protokoły ekwiwalentne (Art-Net pozwala na przesył większych ilości danych na większe odległości). Po odebraniu pakietu od zewnętrznej konsoli oświetleniowej, pakiet zostaje zinterpretowany (wartości na różnych kanałach, jako kolory, tryby pracy), przepisany do pakietu w komunikacji IR i nadany za pośrednictwem diod IR. Sposób sterowania pozwala wykorzystać wiele nadajników, aby stworzyć niezależnie adresowane strefy przestrzenne.

### 3.2 Projekt elektroniki

Urządzenie składa się z zewnętrznego zasilacza, mikrokontrolera, modułu komunikacyjnego i układu nadawczego z diodami IR wysokiej mocy.

Wymagane elementy:

- mikrokontroler z modułem Ethernet - WizNet W5500-EVB-PICO
- diody IR - TSAL6200 (20x 0.21W)
- tranzystor do sterowania diodami - IRL540NPBF
- zasilacz sieciowy 9V

Pierwotnym planem było użycie modułu komunikacyjnego ENC28J60, okazało się jednak, że nie istnieją do nich sterowniki zintegrowane z wbudowanymi funkcjami obsługi komunikacji sieciowej dla układu RP2040, zdecydowałem więc o zmianie sterownika na W5500-EVB-PICO.

## 3.3 Realizacja

### 3.3.1 IR

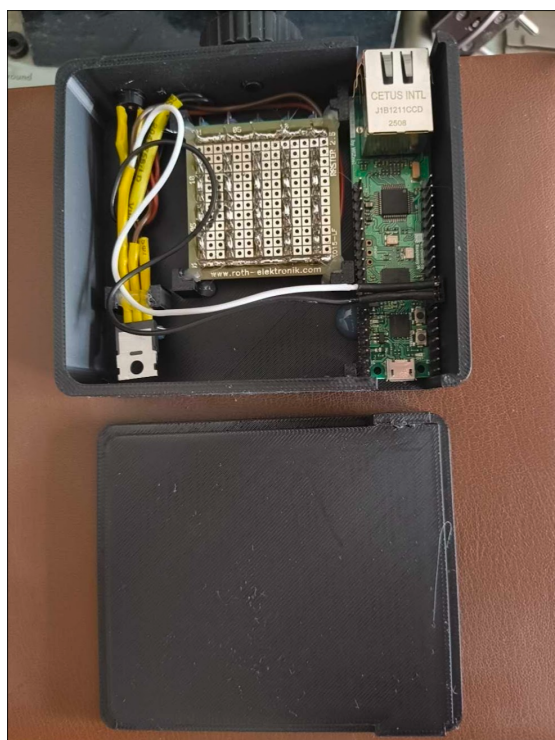
Generowanie sygnałów sterujących dla modułów opasek okazało się być bardzo proste - wymagane jest generowanie sygnału prostokątnego o częstotliwości 38kHz, modulowanego binarnie impulsami reprezentującymi "jedynek" i "zera". Bardzo dobrze w tym zastosowaniu sprawdził się dostępny generator sygnału PWM ustawiony na częstotliwość 38kHz i wypełnienie 50%. Modulacja została zrealizowana poprzez zmianę wypełnienia sygnału z 50% do 0%. Powstał moduł *lib\_pixmod.py* generujący i nadający w opisany wyżej sposób komunikaty sterujące.

### 3.3.2 Art-Net

Po ramowym zapoznaniu się z dokumentacją standardu Art-Net, przygotowany został moduł *lib\_artnet.py* obsługujący przychodzące, rozgłaszane w sieci lokalnej ramki Art-Net. Inicjalizowana jest komunikacja za pośrednictwem wbudowanego modułu Ethernet, po odebraniu wiadomości na wskazanym porcie, treść jest rozpakowywana, dzielona na wartości kanałów i zapisywana w pamięci obiektu. Bezpośrednio po otrzymaniu aktualizacji stanu magistrali, wywoływana jest przez obiekt funkcja odwoławcza, w której nadawany jest komunikat przez podczerveń.

### 3.3.3 Obudowa

Dla wygody użytkownika przygotowana została dla kontrolera obudowa. Model wykonany został w oprogramowaniu Autodesk Fusion 360, a następnie wydrukowany z PETG (rys. 5). Obudowa posiada otwory na złącza, diody LED IR i otwór do montażu na statywie fotograficznym.



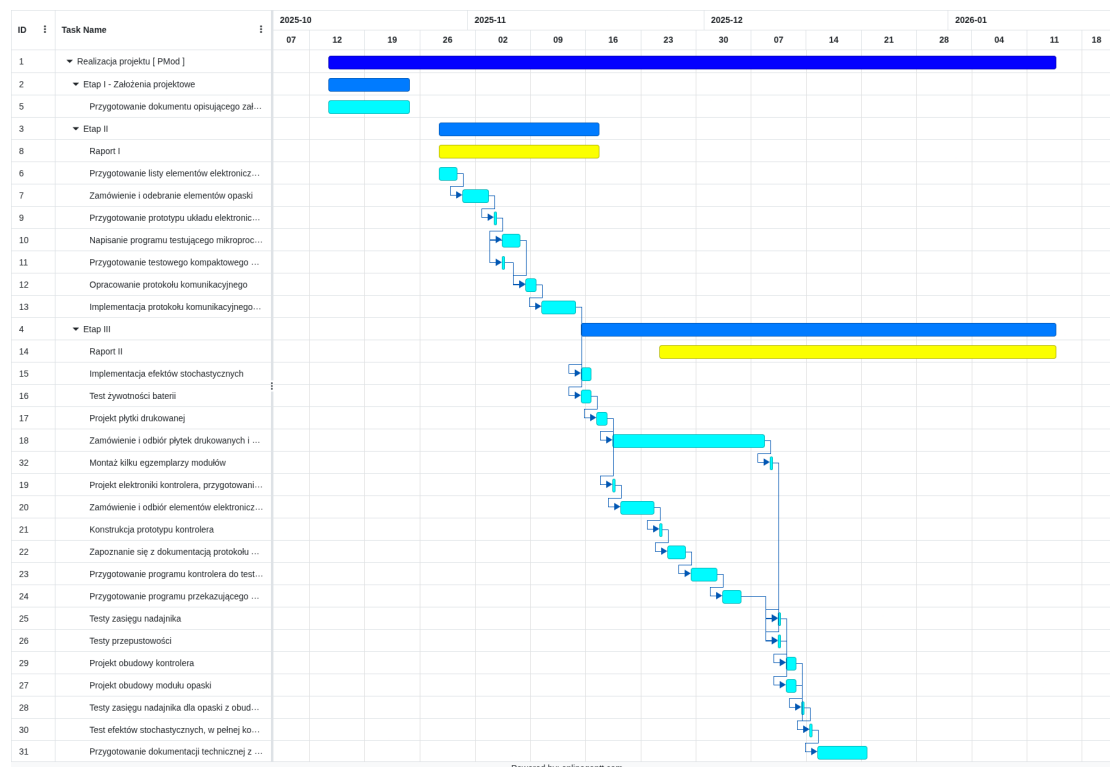
Rysunek 5: Obudowa z gotowym układem elektronicznym kontrolera

### 3.4 Testy

Wstępne testy przebiegły pomyślnie - statyczny program sterujący opaską działał z ok. 90% skutecznością. Integracja nadajnika z protokołem Art-Net dała rozsądne, ale nie zadowalające rezultaty - najprawdopodobniej mikrokontroler opaski nie nadąża za szybkim tempem aktualizacji danych w protokole Art-Net. Jednak ok. 1/3 efektów i podstawowe kolory działają poprawnie. Podczas testów, do generowania komunikatów Art-Net wykorzystane zostało oprogramowanie QLC+.

## 4 Harmonogram prac

Dla projektu stworzony został diagram Gantta (rys. 6) oraz, na jego podstawie, wyznaczona ścieżka krytyczna - minimalny czas prac to 55 dni, a więc w zapasie pozostaje ok. miesiąc. W warunkach rzeczywistych okazało się, że dostępny czas nie wystarczył - pozostało kilka nierozwiązanych problemów.



Rysunek 6: Diagram Gantta projektu

Proponowane kamienie milowe to natomiast:

1. Gotowy prototyp układu opaski
2. Działający prototyp modułu opaski
3. Gotowy prototyp kontrolera
4. Działający prototyp kontrolera
5. Gotowe i działające moduły opasek i kontroler
6. Dokumentacja techniczna z instrukcją obsługi

#### 4.1 Postęp prac

Tabele 1 i 2 przedstawiają ramowy harmonogram prac podczas realizacji projektu.

Przygotowane oprogramowanie i inne materiały dostępne są w poniższym repozytorium:

[https://git.make-more.cloud/mbowszys/PWr\\_AiR\\_s5\\_SR\\_P\\_pixmod](https://git.make-more.cloud/mbowszys/PWr_AiR_s5_SR_P_pixmod)

Nr	Zadanie	Postęp [%]
1	Przygotowanie listy elementów elektronicznych dla modułu opaski	100
2	Zamówienie i odebranie elementów opaski	100
3	Przygotowanie prototypu układu elektronicznego opaski	100
4	Napisanie programów testujących mikroprocesor i peryferia	100
5	Przygotowanie testowego kompaktowego nadajnika podczerwieni	100
6	Opracowanie protokołu komunikacyjnego	100
7	Implementacja protokołu komunikacyjnego bez efektów stochastycznych	100

Tabela 1: Plan prac - Etap II

Nr	Zadanie	Postęp [%]
8	Implementacja efektów stochastycznych	0
9	Test żywotności baterii	100
10	Projekt płytki drukowanej	100
11	Zamówienie i odbiór płytek drukowanych i dodatkowych elementów elektronicznych	100
12	Montaż kilku egzemplarzy modułów	100
13	Projekt elektroniki kontrolera, przygotowanie listy elementów	100
14	Zamówienie i odbiór elementów elektronicznych do kontrolera	100
15	Konstrukcja prototypu kontrolera	100
16	Zapoznanie się z dokumentacją protokołu Art-Net	100
17	Przygotowanie programu kontrolera do testów podzespołów	100
18	Przygotowanie programu przekazującego komunikaty z Art-Net do CIR	100
19	Testy zasięgu nadajnika	100
20	Testy przepustowości	100
21	Projekt obudowy kontrolera	100
22	Projekt obudowy modułu opaski	30
23	Testy zasięgu nadajnika dla opaski z obudową	0
24	Test efektów stochastycznych, w pełnej konfiguracji sprzętowej	0
25	Przygotowanie dokumentacji technicznej z instrukcją obsługi	50

Tabela 2: Plan prac - Etap III

#### 4.1.1 Komentarz do postępu prac

8 - Nie wystarczyło czasu na implementację, może być ona utrudniona lub niemożliwa, ze względu na ograniczoną pamięć i częstotliwość pracy układu (generowanie wartości pseudolosowych).

22 - Rysunek odręczny

23,24 - Nie przeprowadzone, ze względu na problem z programowaniem układów w modułach z płytką PCB.

25 - Nie została przygotowana instrukcja obsługi.

## Bibliografia

- [1] Atmel. *Dokumentacja Atmel 8-bit AVR Microcontroller with 2/4/8K Bytes In-System Programmable Flash*. [https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ProductDocuments/DataSheets/Atmel-2586-AVR-8-bit-Microcontroller-ATtiny25-ATtiny45-ATtiny85\\_Datasheet.pdf](https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ProductDocuments/DataSheets/Atmel-2586-AVR-8-bit-Microcontroller-ATtiny25-ATtiny45-ATtiny85_Datasheet.pdf). Accessed: 2025-10-26.
- [2] Elliot V John. *Introducing infrared remote control protocols*. en. <https://hackaday.com/2025/04/13/introducing-infrared-remote-control-protocols/>. Accessed: 2025-10-25. Kw. 2025.
- [3] OptoSupply. *Dokumentacja OSRB38C9BB*. <https://www.tme.eu/Document/3a22487aaecfd54815d45be1bce15028/OSRB38C9BB.pdf>. Accessed: 2025-10-26.
- [4] REFOND. *Dokumentacja RF-W7SA50TS-A51-IC*. <https://www.tme.eu/Document/0211cf53e906994e2bdeefa32e6c2fb0/RF-W7SA50TS-A51-IC.pdf>. Accessed: 2025-10-26.