

A G H

**Akademia Górniczo-Hutnicza
im. Stanisława Staszica
w Krakowie**

Wydział Informatyki, Elektroniki i Telekomunikacji

Złożone Systemy Cyfrowe

2018/2019

**SYSTEM STEROWANIA CYFROWEGO
MAKIETĄ KOLEJOWĄ**

Mateusz Buta
Student III roku
Informatyka WIET

SPIS TREŚCI

Temat	3
Elementy systemu	3
Mikrokontroler na płytce Arduino Leonardo	3
Pulpit sterujący	3
Układy PCF8574	3
Napędy rozjazdów i semaforów	4
Detektor pociągu	4
Etapy realizacji projektu	5
Implementacja magistrali I2C	5
Implementacja obsługi układu PCF8574	5
Implementacja obsługi serwomechanizmów na bazie przerwań.	6
Wstęp	6
Inicjacja magistrali	6
Implementacja funkcji reagującej na przerwania	7
Dobór parametrów	7
Testy implementacji	8
Implementacja obsługi serwomechanizmów na bazie układu PCF8574	8
Detektor pociągu	9
Dwa fotorezystory SMD tworzą jeden detektor, dzięki temu w dalszej fazie rozwoju projektu będzie można wykryć kierunek ruchu przejeżdżającego taboru:	9
Realizacja detekcji	10
Prototyp systemu	11
Harmonogram	13
Bibliografia	13

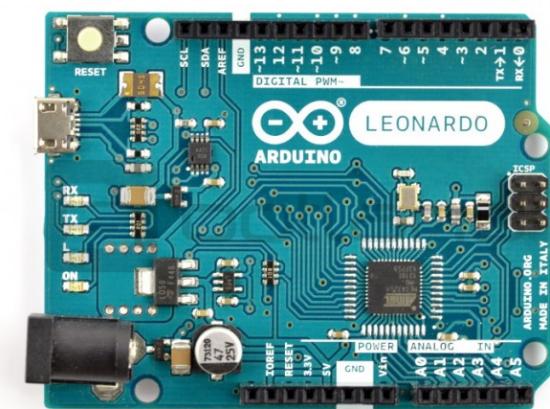
1. TEMAT

Tematem projektu jest zbudowanie prototypu systemu sterowania makietą kolejową opartego na mikrokontrolerze Arduino. Głównym elementem systemu będzie pulpit sterujący w którym umieszczony będzie mikrokontroler. Sterowanie elementami wykonawczymi takimi jak rozjazdy, semafory będzie odbywać się magistralą I2C przy wykorzystaniu portów równoległych na bazie układu PCF8574 oraz mikrokontrolerów pracujących w trybie Slave.

2. ELEMENTY SYSTEMU

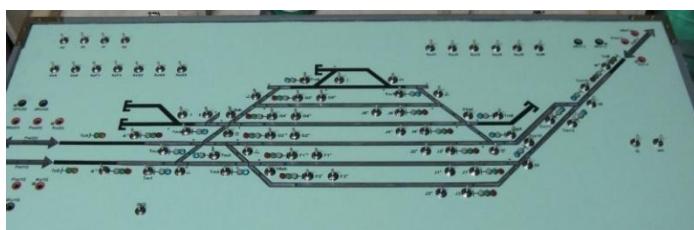
MIKROKONTROLER NA PŁYTCE ARDUINO LEONARDO

Główny element systemu. Komunikuje się za pomocą magistrali i2c z układami wykonawczymi, w tym również z innymi mikrokontrolerami pracującymi w trybie Slave.



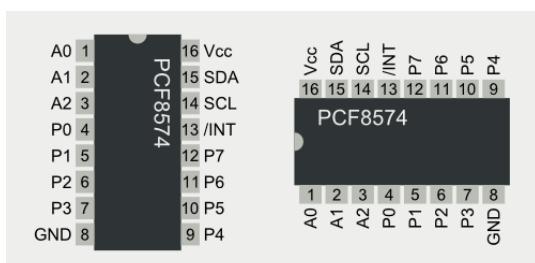
PULPIT STERUJĄCY

Umożliwia sterowanie napędami rozjazdów przy pomocy przycisków. Podświetlenie pozwala śledzić przejazd pociągu.



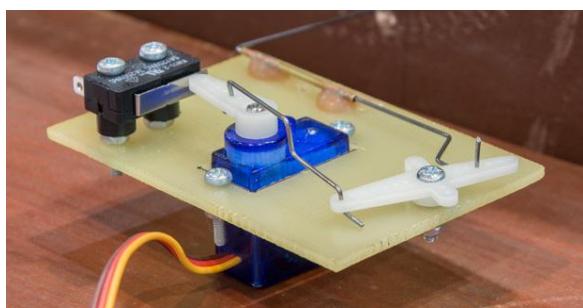
UKŁADY PCF8574

Porty równoległe sterowane magistralą i2c, prowadzoną pod makietą, obsługujące proste lokalnie urządzenia wykonawcze, takie jak przyciski i diody.



NAPĘDY ROZJAZDÓW I SEMAFORÓW

Przekładanie rozjazdów realizowane przy pomocy serwomechanizmów obsługiwanych przez układy wykonawcze.



DETEKTOR POCIĄGU

Układ wykrywający przejazd pociągu, zrealizowany na bazie fotorezystorów. Detekcja pozwala pokazać na pulpicie przejazd pociągu.



ETAPY REALIZACJI PROJEKTU

1. IMPLEMENTACJA MAGISTRALI I2C

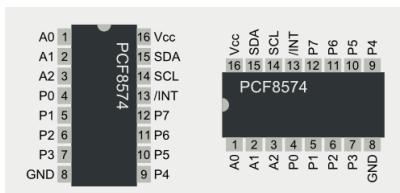
Magistrala I2C została zaimplementowana przy użyciu wsparcia sprzętowym w postaci rejestrów. Poza funkcjami podstawowymi do obsługi magistrali: *twiStart*, *twiAddress*, *twiWrite*, *twiRead*, *twiStop*, zaimplementowane też zostały funkcje przygotowane do obsługi prostych układów: *twiSend* i *twiReceive* służące do wysyłania i odbierania jednego bajtu informacji z podanego adresu.

Do przetestowania działania zaimplementowana została funkcja *twiScan* skanująca magistralę I2C w poszukiwaniu aktywnych urządzeń i wypisującą na porcie szeregowym znalezione adresy. Pozwoliło to przetestować poprawną inicjalizację magistrali.

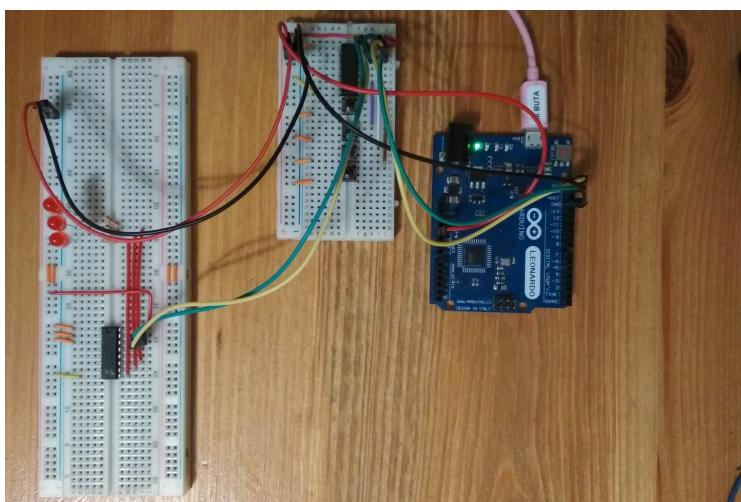
```
Address 20 100000 is present
Address 29 101001 is present
Address 38 111000 is present
```

2. IMPLEMENTACJA OBSŁUGI UKŁADU PCF8574

Na bazie przygotowanej implementacji magistrali I2C zaimplementowałem bibliotekę do obsługi układu PCF8574. Biblioteka udostępnia wiele funkcji znanych z obsługi zwykłych pinów Arduino takich jak *pinMode*, *digitalWrite*, *write*, *digitalRead*, oraz kilka dodatkowych: *read*, *toggle*, *clear*, *set*. Pozwalają one na elastyczną obsługę układu.



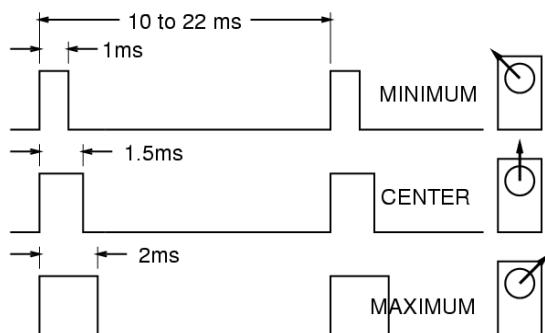
Przetestowanie implementacji na układzie prototypowym. Informacja o stanie przycisków jest odczytywana przez mikrokontroler od układu PCF8574. Mikrokontroler analizuje informacje i komunikuje się z drugim układem PCF8574 zapalając odpowiednie diody.



3. IMPLEMENTACJA OBSŁUGI SERWOMECHANIZMÓW NA BAZIE PRZERWAŃ.

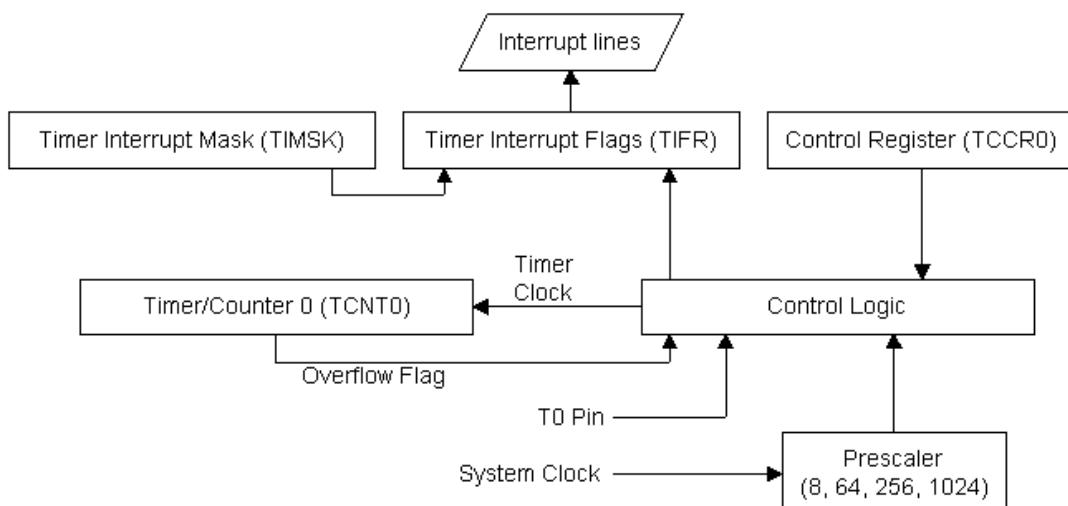
WSTĘP

Do obsługi serwomechanizmów trzeba wygenerować sygnał PWM o odpowiedniej częstotliwości:



INICJACJA MAGISTRALI

Częstotliwość taktowania zegara wynosi 16kHz, zatem ustalając preskaler na 1:8 i ustawiając licznik od 0 do 20, otrzymujemy przerwania co 10us. Do wywoływanego przerwań posłużyłem się zegarem Timer3.



```
cli(); // Disable interrupts.  
TCCR3A = 0;  
TCCR3B = 0;  
  
TCCR3B |= (1 << WGM32); // Turn on CTC mode.  
TCCR3B |= (1 << CS31); // Set a 1:8 prescaler.  
  
TCNT3 = 0; // Count from 0 to 20.  
OCR3A = 20;  
  
TIMSK3 |= (1 << OCIE3A); // Enable timer compare interrupt.  
sei(); // Enable interrupts.
```

IMPLEMENTACJA FUNKCJI REAGUJĄCEJ NA PRZERWANIA

Funkcja operuje na statycznej tablicy `servoPins`, która pozwala na obsługę kilku serwomechanizmów jednocześnie. Przerwania wywoływanie są co 10us. Główny licznik `counter` zlicza od 0 do `SERVO_PWM_PERIOD = 2000`, dzięki czemu uzyskany okres sygnału wynosi 20ms. Odliczanie rozpoczyna się ze stanem wysokim na każdym wyjściu, jeśli `counter` osiągnie wartość `servoHighTimes[idx]` na odpowiednim wyjściu `servoPins[idx]` podawany jest sygnał niski.

```
ISR(TIMER3_COMPA_vect)
{
    counter++;
    if(counter >= SERVO_PWM_PERIOD){
        counter=0;
        for(uint8_t idx=0;idx<servoNumber;idx++)
            digitalWrite(servoPins[idx],HIGH);
    }else{
        for(uint8_t idx=0;idx<servoNumber;idx++)
            if(counter == servoHighTimes[idx])
                digitalWrite(servoPins[idx],LOW);
    }
}
```

DOBÓR PARAMETRÓW

Obserwacje zachowania serwomechanizmu i wykresu z oscyloskopu pozwoliło poprawić parametry generowanego sygnału:

```
#define SERVO_PWM_PERIOD 1900
#define MIN_PULSE_WIDTH 54
#define DEFAULT_PULSE_WIDTH 150
#define MAX_PULSE_WIDTH 240
```

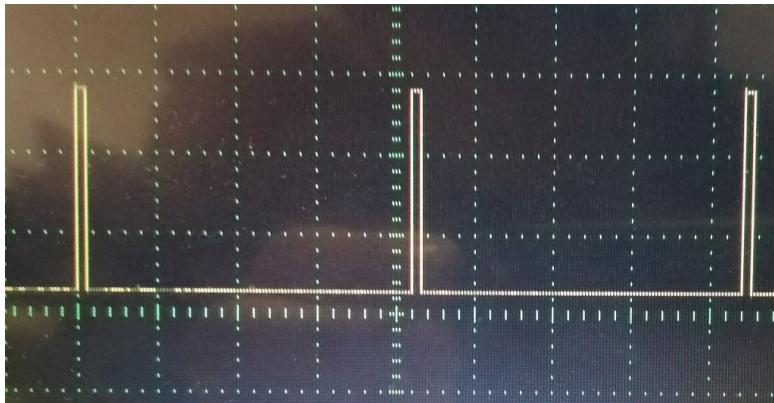
USTAWIENIE SERWOMECHANIZMU W ŻĄDANEJ POZYCJI

Polega na mapowaniu wartości kąta od 0 do 180 stopni na długość pulsu. Wpisanie nowej wartości do rejestru wymaga chwilowej blokady obsługi przerwań.

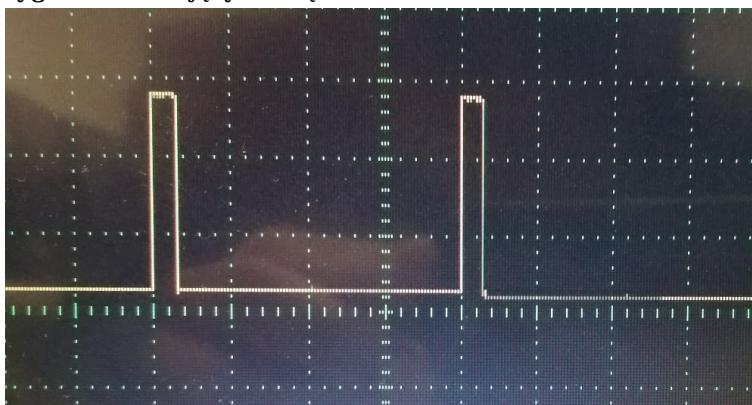
```
void Servo::setPosition(uint16_t angle)
{
    pulse_width = map(angle, 0, 180, MIN_PULSE_WIDTH, MAX_PULSE_WIDTH);
    TIMSK3 &= ~(1 << OCIE3A); // disable timer compare interrupt
    servoHighTimes[_idx] = pulse_width;
    TIMSK3 |= (1 << OCIE3A); // enable timer compare interrupt
}
```

TESTY IMPLEMENTACJI

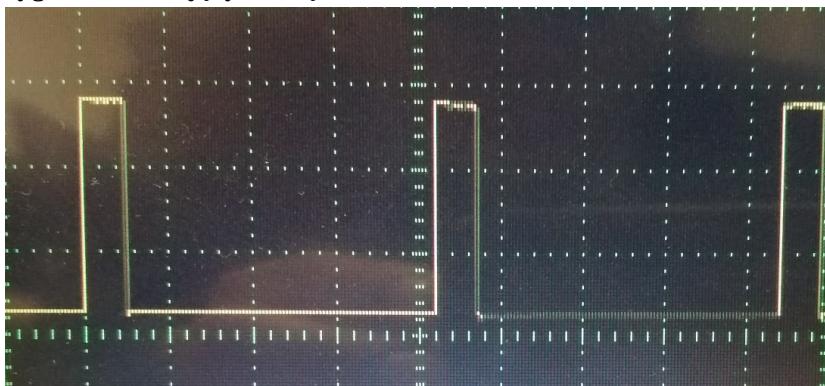
Sygnał ustawiający ramię serwomechanizmu na 0° :



Sygnał ustawiający ramię serwomechanizmu na 90° :



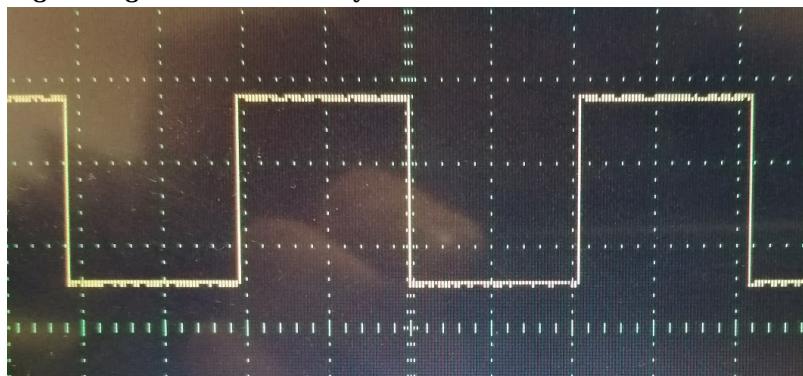
Sygnał ustawiający ramię serwomechanizmu na 180° :



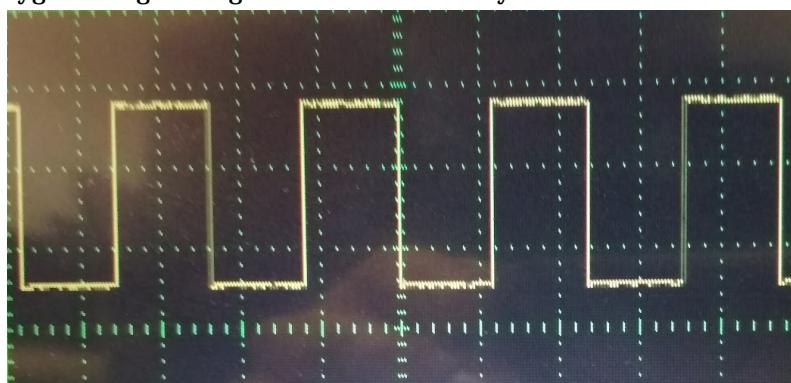
4. IMPLEMENTACJA OBSŁUGI SERWOMECHANIZMÓW NA BAZIE UKŁADU PCF8574

Okazała się niemożliwa w realizacji, ponieważ czas przesyłania sygnału do układu wynosi około 20ms, czyli tyle ile trwa okres sygnału PWM sterującego położeniem serwomechanizmu. Po podniesieniu częstotliwości sysgnału zegarowego SCL z 100kHz do 400kHz czas przesyłania sygnału zmalał do 12ms. Dalsze podniesienie częstotliwości magistrali I2C jest nie możliwe, ponieważ układ PCF8574 nie obsługuje większych częstotliwości. Uniemożliwia to poprawne generowanie sygnału PWM układach PCF8574.

Czas potrzebny na przesłanie informacji do PCF8574 przy częstotliwości sygnału zegarowego SCL 100kHz wynosi 20ms:



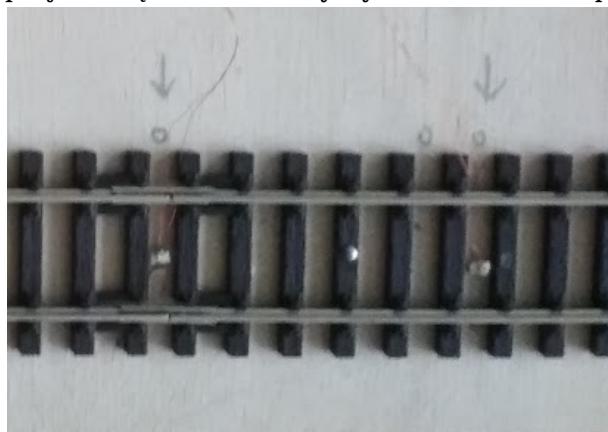
Czas potrzebny na przesłanie informacji do PCF8574 po podniesieniu częstotliwości sygnału zegarowego SCL do 400kHz wynosi 12ms:



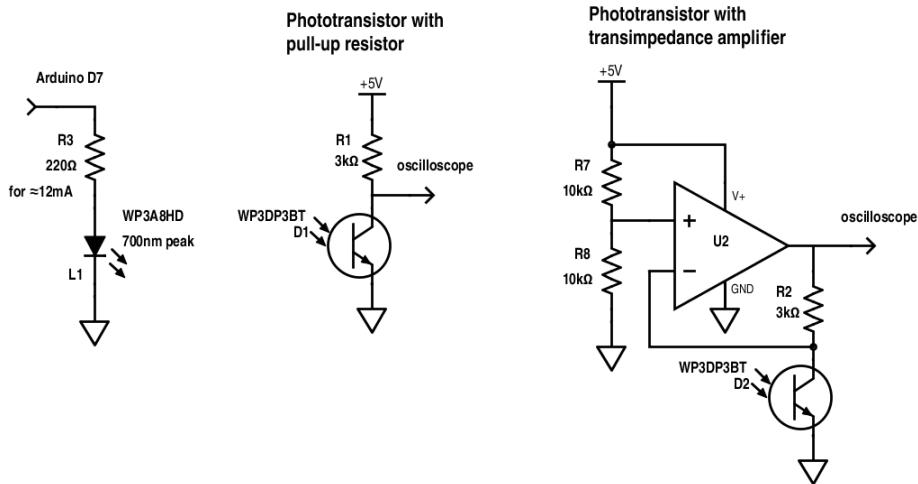
5. DETEKTOR POCIĄGU

Realizowany przy pomocy fotorezystorów SMD umieszczonych pomiędzy podkładami toru kolejowego, tak aby przejeżdżający tabor przysłaniał fotorezystor. Obok toru umieszczony będzie fotorezystor referencyjny, mierzący bazowe natężenia światła. Wystąpienie dużej różnicy pomiędzy fototranzystora w torze kolejowym, a fotorezystorem referencyjnym będzie oznaczało przejazd taboru.

Dwa fotorezystory SMD tworzą jeden detektor, dzięki temu w dalszej fazie rozwoju projektu będzie można wykryć kierunek ruchu przejeżdżającego taboru:



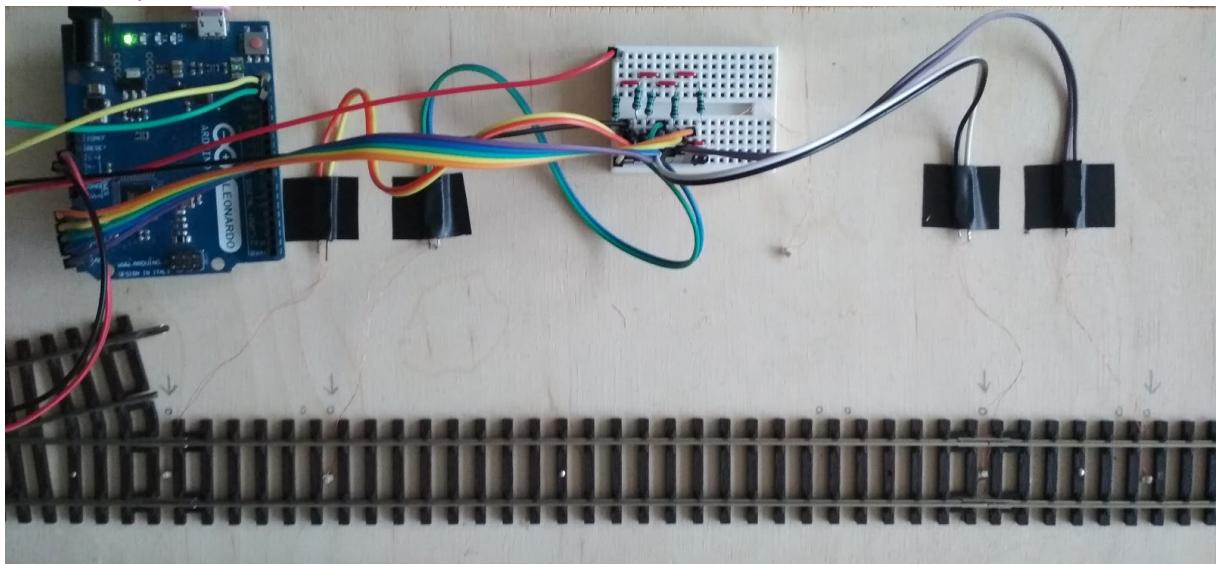
W trakcie realizacji okazało się że funkcjonalność fotorezystora można realizować korzystając z fototranzystora:



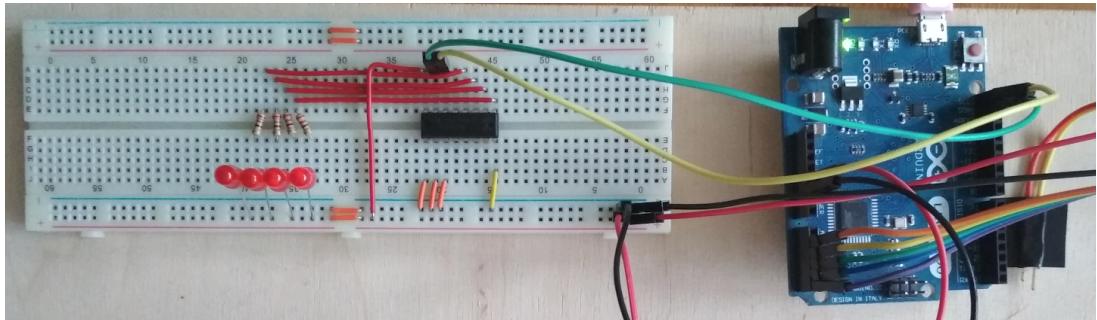
6. REALIZACJA DETEKCIJ

Tor testowy z pięcioma fotorezystorami - cztery detektory umieszczone są w torach, piąty jest detektorem referencyjnym.

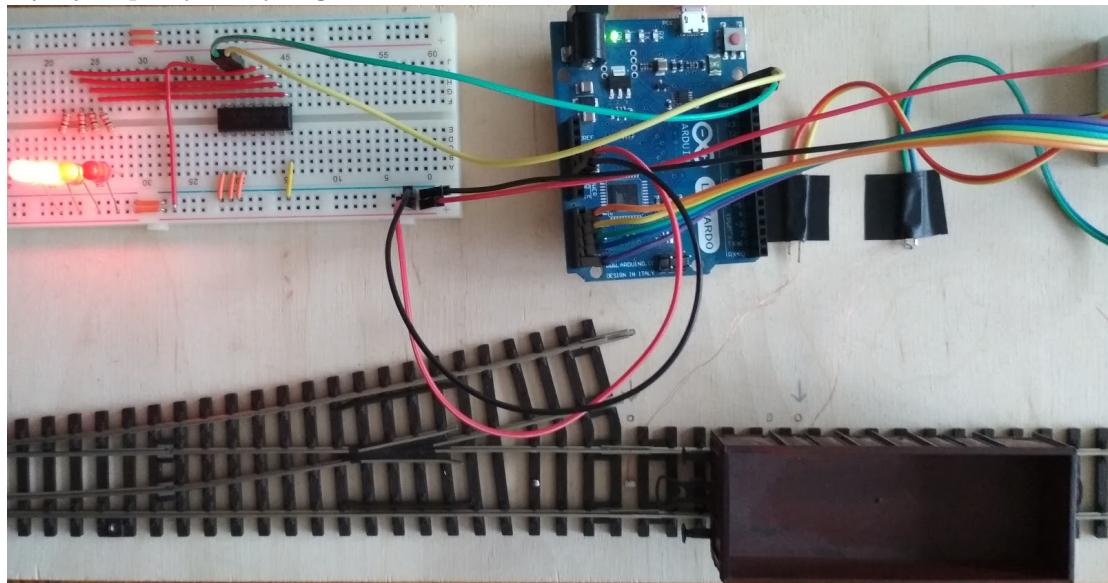
Tor testowy:



Pulpit prezentujący stan czujników:



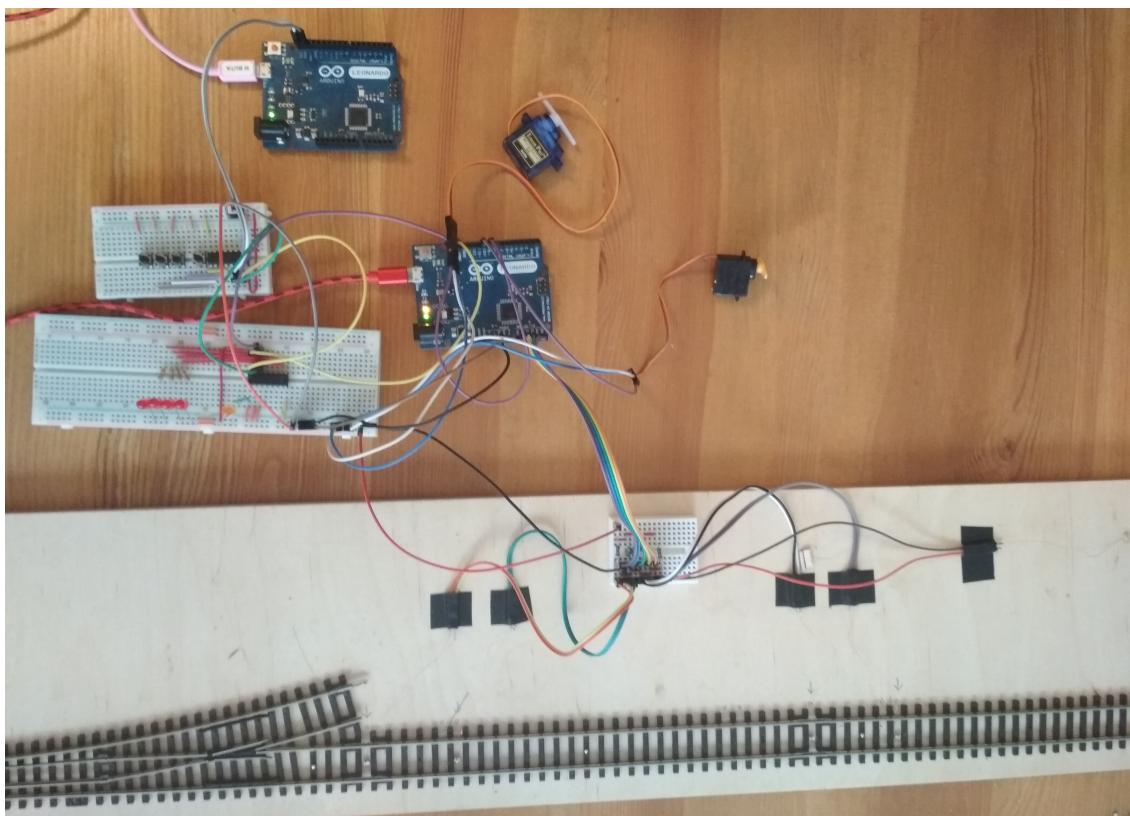
Wykrycie przejeżdżającego taboru:



7. PROTOTYP CAŁEGO SYSTEMU

Zbudowanie funkcjonującego prototypu. W jego skład wchodzą:

1. Mikrokontroler główny:
Obsługuje komunikację między pulpitem i układami wykonawczymi.
2. Mikrokontroler wykonawczy:
Obsługuje detektory pociągu i serwomechanizmy przekładające rozjazdy.
3. Pulpit:
Reprezentowany przez układy PCF8574 i podpięte do nich przyciski i diody.



HARMONOGRAM

- 10.10.2018 Prezentacja ogólnego tematu projektu.
15.10.2018 Dopełnianie tematu projektu.
31.10.2018 Prezentacja implementacji magistrali I2C i implementacja układu PCF8574.
14.11.2018 Prezentacja implementacji obsługi serwomechanizmu na bazie Timera.
16.11.2018 Pomiar czasu przesyłania sygnału magistralą I2C na oscyloskopie.
28.11.2018 Prezentacja prac nad detektorem oraz programu prezentującego zajętość toru.
12.12.2018 Prezentacja układu z detektorami.
17.12.2018 Testowanie detektora na oscyloskopie.
09.01.2018 Poprawienie detekcji i zajętości toru.
16.01.2018 Prezentacja prototypu systemu oraz oddanie dokumentacji.

BIBLIOGRAFIA

1. Dokumentacja użytkownika Symulatora prowadzenia ruchu kolejowego ISDR
<https://www.symulator.isdr.pl/download/dokumentacja.pdf>
2. Dokumentacja mikrokontrolera Atmega32u4
<https://www.pjrc.com/teensy/atmega32u4.pdf#page=240&zoom=100,0,173>
3. How to use I2C / TWI (Two Wire Interface) in AVR ATmega32
<https://www.engineersgarage.com/embedded/avr-microcontroller-projects/atmega32-twi-two-wire-interface>
4. Obsługa interfejsu TWI (I2C) na mikrokontrolerze Atmega16 – Cezary Klimasz
http://ktc.wieik.pk.edu.pl/kurs_avr/avr_twiobsługa.pdf
5. Dokumentacja ekspandera PCF8574
<https://www.nxp.com/docs/en/data-sheet/PCF8574A.pdf>
<http://www.ti.com/lit/ds/symlink/pcf8574.pdf>
6. Nota katalogowa serwomechanizmu Tower Pro Micro Servo SG90
http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf
7. Słownik terminologii angielskiej sterowania ruchem kolejowym.
<https://www.bsk.isdr.pl/slownik.php?l1=PL&l2=EN>
8. How to get reliable readings from photoresistors - Robert Hongpu Ma
<https://www.cs.sfu.ca/CourseCentral/433/bfraser/other/2015-student-howtos/ReadPhotoresistors.pdf>
9. Zastosowania wzmacniaczy operacyjnych
http://www.mif.pg.gda.pl/homepages/sylas/students/mt/is/wzmacniacze_operacyjne.pdf
10. Dokumentacja fotorezystora PT15-21C/TR8
<https://www.tme.eu/pl/Document/bb4010342c0e235fd0f2a8009a74bb5c/ELPT15-21C.pdf>