

PROJEKT

STEROWNIKI ROBOTÓW

Raport

**Humanistycznie upośledzony robot
akrobatyczny**

HURA

Skład grupy:

Albert LIS, 235534
Michał MORUŃ, 235986

Termin: sr TP15

Prowadzący:

mgr inż. Wojciech DOMSKI

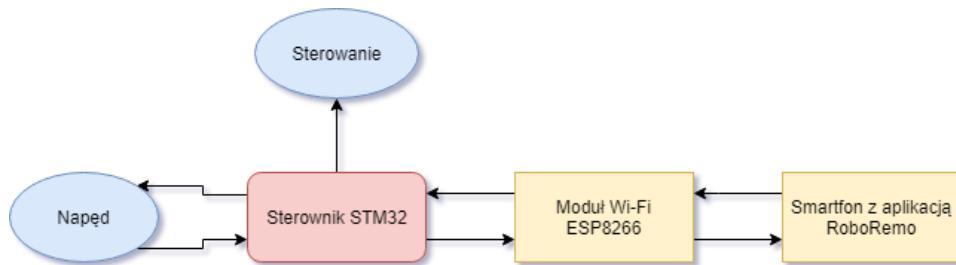
27 maja 2019

Spis treści

1 Opis projektu	2
2 Konfiguracja mikrokontrolera	3
2.1 Konfiguracja pinów	5
2.2 ADC 1	5
2.3 Timer 2	5
2.4 Timer 4	6
2.5 Timer 6	6
3 Projekt elektroniki	6
3.1 Schemat elektryczny	6
3.2 Regulacja położenia wertykalnego	6
4 Konstrukcja mechaniczna	7
5 Opis działania programu	8
5.1 Schemat działania programu	8
5.2 Funkcja obsługująca przerwanie timera 6	8
5.3 Funkcja obsługująca przerwanie ADC	8
6 Zrealizowane zadania	8
6.1 Wizualizacja	8
7 Urządzenia zewnętrzne	9
8 Podsumowanie	9
Bibliografia	10

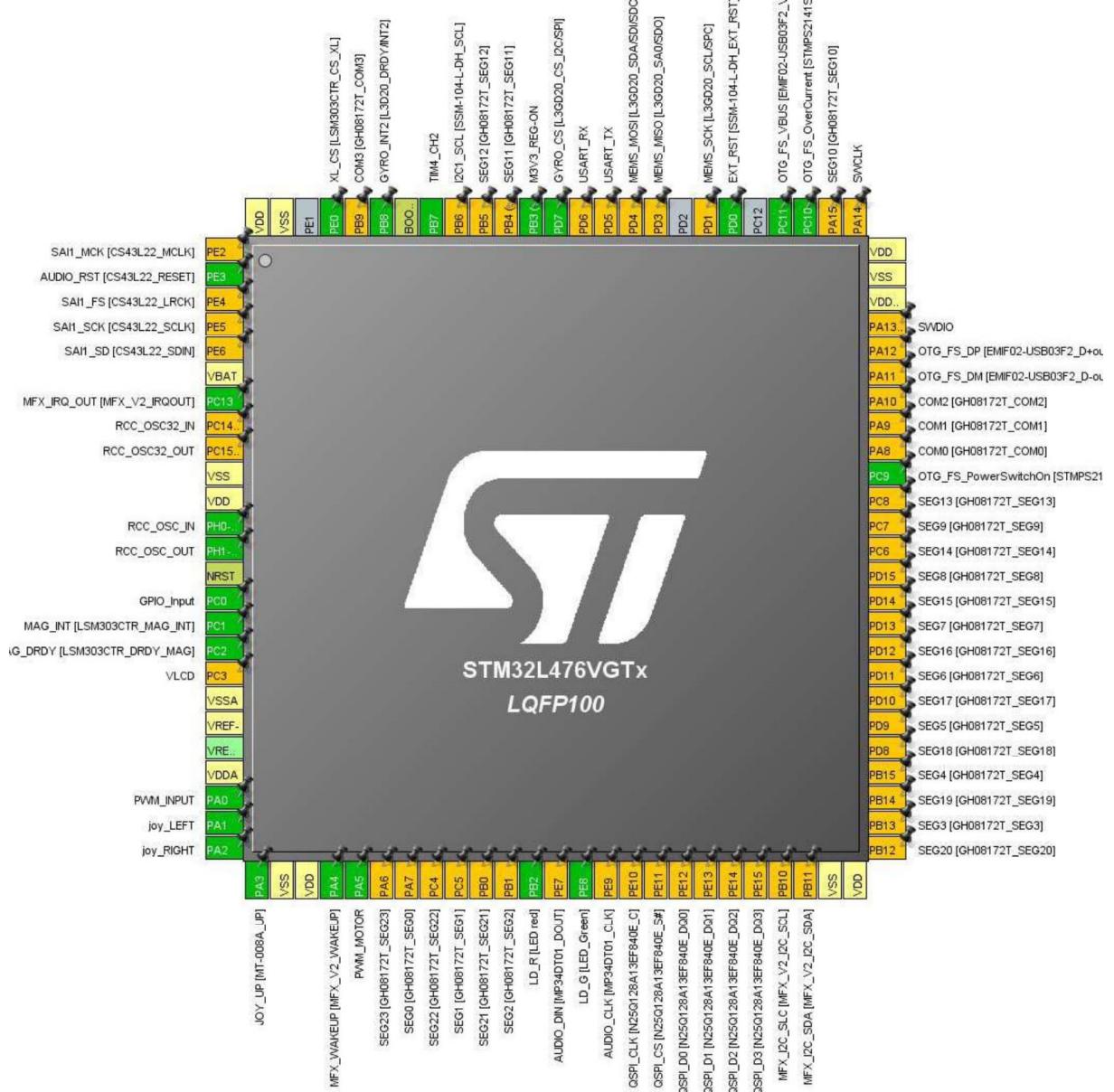
1 Opis projektu

Celem projektu jest zbudowanie zdalnie sterowanego robota jezdniego. Robot będzie sterowany za pomocą akcelerometru w telefonie. Dane będą przesyłanie za pomocą Wi-Fi lub Bluetooth. Regulacja prędkości będzie się odbywać za pomocą regulatora PID. Dane o prędkości będą pobierane z enkoderów znajdujących się w kołach robota. Opcjonalnie robot będzie wyświetlał szczegółowe dane o swoim stanie wewnętrznym za pomocą wbudowanego w płytke z mikrokontrolerem wyświetlacza LCD.

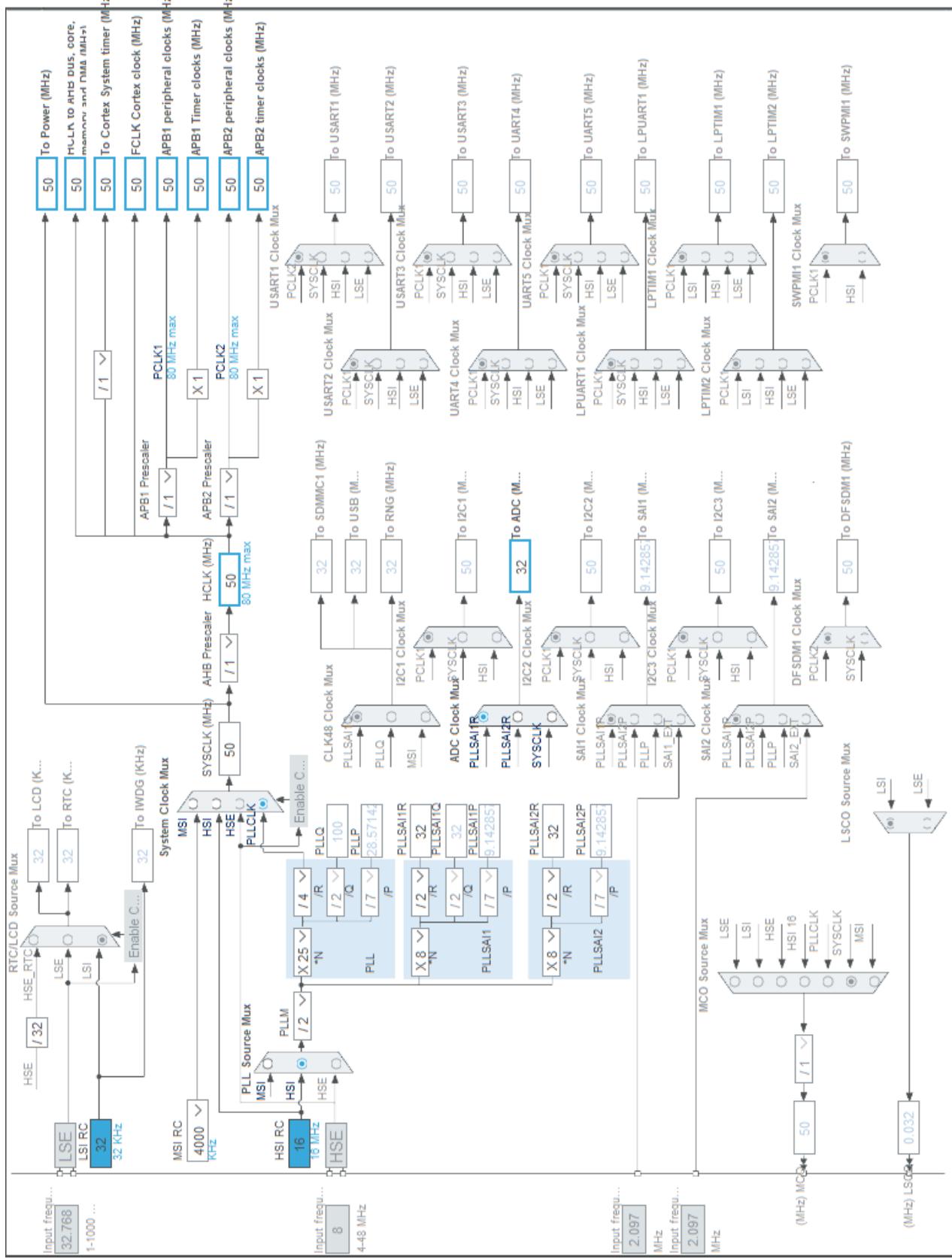


Rysunek 1: Architektura systemu

2 Konfiguracja mikrokontrolera



Rysunek 2: Konfiguracja wyjść mikrokontrolera w programie STM32CubeMX



Rysunek 3: Konfiguracja zegarów mikrokontrolera

2.1 Konfiguracja pinów

PIN	Tryb pracy	Funkcja/etykieta
PC14	OSC32_IN*	RCC_OSC32_IN
PC15	OSC32_OUT*	RCC_OSC32_OUT
PH0	OSC_IN*	RCC_OSC_IN
PH1	OSC_OUT*	RCC_OSC_OUT
PD5	USART2_TX	USART_TX
PD6	USART2_RX	USART_RX
PA0	ADC1_IN5	PWM_INPUT
PA1	GPIO_Input	JOY_LEFT
PA2	GPIO_Input	JOY_RIGHT
PA3	GPIO_Input	JOY_UP
PA4	GPIO_Input	JOY_DOWN
PA5	TIM2_CH1	PWM_MOTOR
PB7	TIM4_CH2	PWM_SERVO

Tabela 1: Konfiguracja pinów mikrokontrolera

2.2 ADC 1

Parametr	Wartość
Resolution	ADC 12-bit resolution
DMA Continuous Requests	Enabled
Data Alignment	Right alignment
Continuous Conversion Mode	Disabled
Channel	Channel 5
Sampling Time	92.5 Cycles

Tabela 2: Konfiguracja peryferium ADC

2.3 Timer 2

Parametr	Wartość
Clock Source	Internal Clock
Channel1	PWM Generation CH1
Prescaler	PWM_PRESC
Counter Mode	Up
Counter Period	PWM_PERIOD
Internal Clock Division	No Division
Mode	PWM mode 1
CH Polarity	High

Tabela 3: Konfiguracja peryferium Timer 2

2.4 Timer 4

Parametr	Wartość
Clock Source	Internal Clock
Channel	PWM Generation CH2
Prescaler	999
Counter Mode	Up
Counter Period	999
Internal Clock Division	No Division
Mode	PWM mode 1
CH Polarity	High

Tabela 4: Konfiguracja peryferium Timer 4

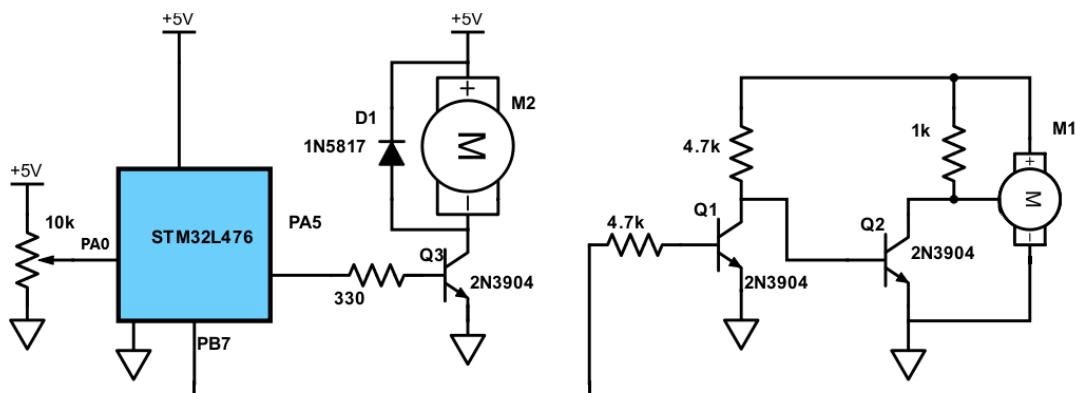
2.5 Timer 6

Parametr	Wartość
Prescaler	TIM6_PRESC
Counter Mode	Up
Counter Period	TIM6_PERIOD
Trigger Event Selection	Update Event

Tabela 5: Konfiguracja peryferium Timer 6

3 Projekt elektroniki

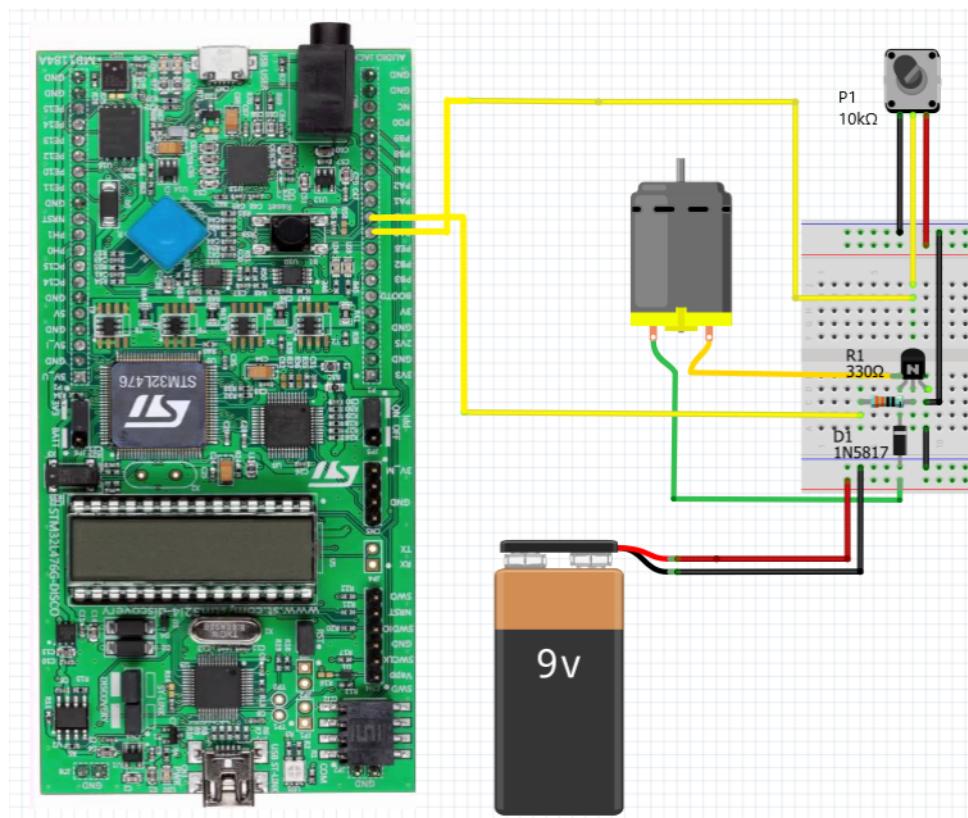
3.1 Schemat elektryczny



Rysunek 4: Schemat elektryczny

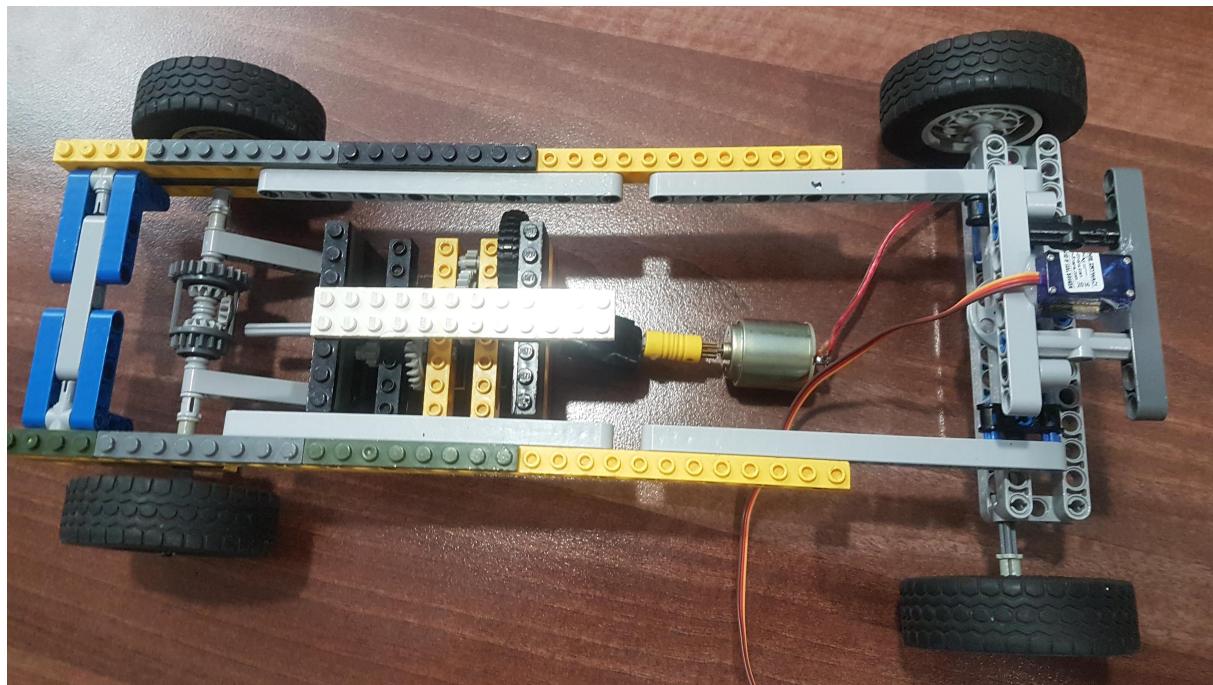
3.2 Regulacja położenia wertykalnego

Do regulacji położenia wertykalnego robota wykorzystano serwo zamontowane na platformie. Jest ono sterowane za pomocą sygnału PWM z Arduino. Obok silnika krokowego zapewnia poruszanie się robota w kierunku światła. Serwo odpowiada za dokładne ustawienie prostopadle do słońca, podwyższając lub obniżając ścianki na których są zamontowane fotorezystory, natomiast silnik krokowy obraca postawę platformy w kierunku światła.

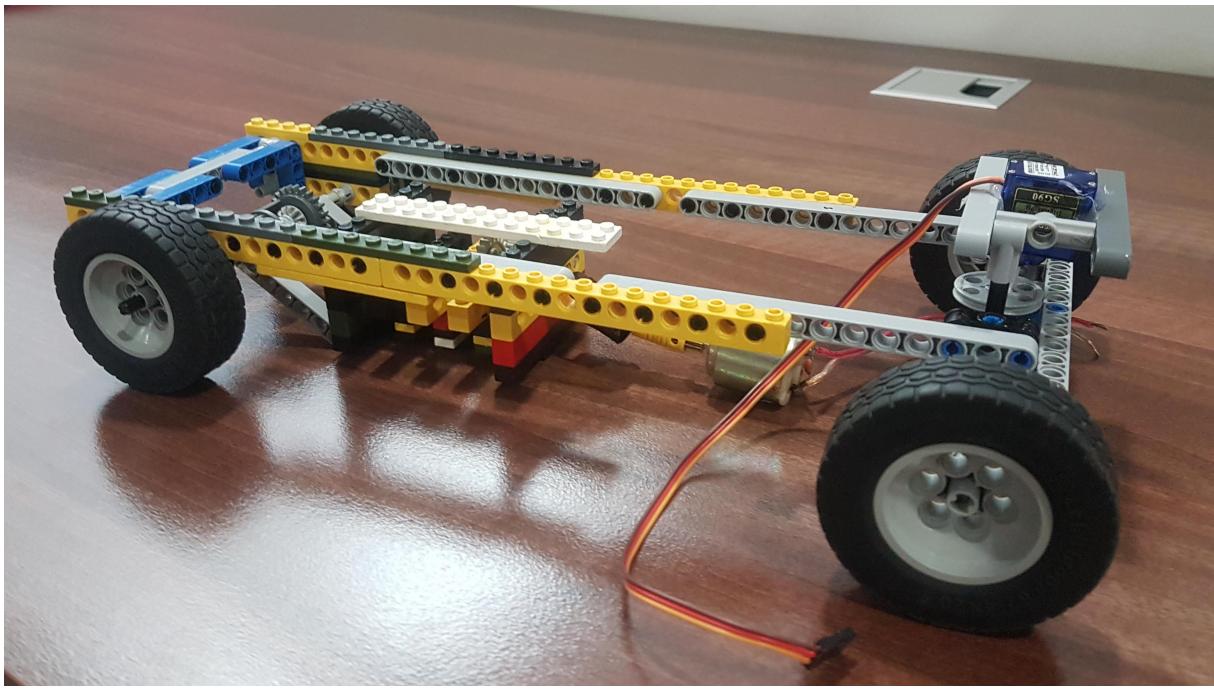


Rysunek 5: Schemat poglądowy regulacji prędkości obrotowej silnika

4 Konstrukcja mechaniczna



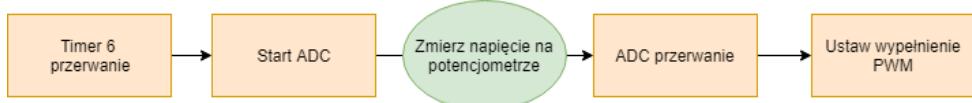
Rysunek 6: Zdjęcie części mechanicznej nr 1



Rysunek 7: Zdjęcie części mechanicznej nr 2

5 Opis działania programu

5.1 Schemat działania programu



Rysunek 8: Schemat działania programu

5.2 Funkcja obsługująca przerwanie timera 6

```

1 void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
2 {
3     if (htim->Instance == TIM6)
4         HAL_ADC_Start_DMA(&hadc1, (uint32_t *)&adc_value, 1);
5 }
  
```

5.3 Funkcja obsługująca przerwanie ADC

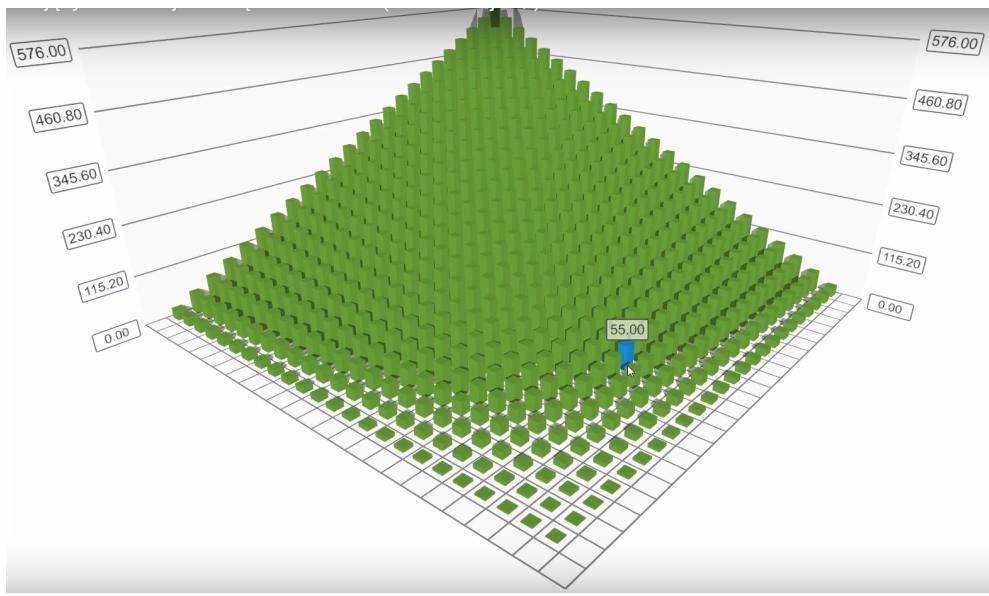
```

1 void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc)
2 {
3     //pid_output = pid_calc(&pid, adc_value, set_value);
4     __HAL_TIM_SET_COMPARE(&htim2, TIM_CHANNEL_1, adc_value);
5 }
  
```

6 Zrealizowane zadania

6.1 Wizualizacja

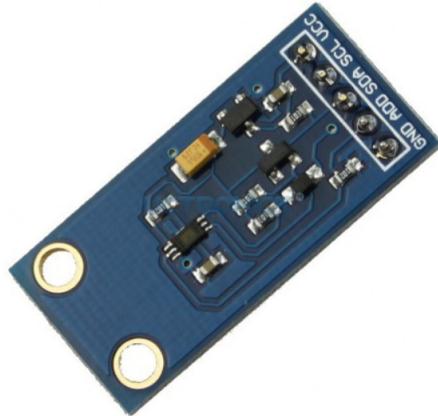
Udało się zrealizować wizualizację ukazującą natężenie światła w określonym miejscu w przestrzeni zrealizowaną dla przykładowych danych. Wizualizacja zakłada, że robot będzie poruszał się po polu prostokąta raz przy razie zbierając w ramach możliwości jak najdokładniejsze dane.



Rysunek 9: Przykładowa wizualizacja

7 Urządzenia zewnętrzne

Wykorzystywanym w projekcie urządzeniem zewnętrznym jest czujnik natężenia światła – GY-30-BH1750. Czujnik został zamontowany na górze platformy z fotorezystorami w celu zbierania informacji potrzebnych do wizualizacji.



Rysunek 10: Czujnik wykorzystany na platformie

8 Podsumowanie

Udało się zrealizować większość zadań. Nastąpiły drobne zmiany koncepcyjne jak użycie potencjometru do regulacji prędkości obrotowej napędu. To będzie wymagać mniejszej ingerencji gdy będziemy projektować regulator PID.

Literatura