

PROJEKT

ROBOTY MOBILNE

Raport

Mobilny manipulator, GrabIO MMG

Skład grupy:

Adrian URBAN, 241558
Kewin GAŁUSZKA, 241624

Termin: ptTN11

Prowadzący:

dr inż. Wojciech DOMSKI

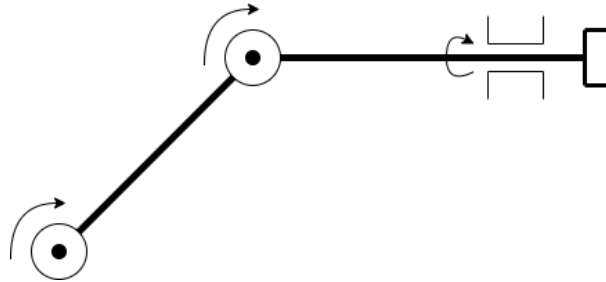
31 marca 2021

Spis treści

1	Opis projektu	2
1.1	Założenia projektowe	2
2	Konfiguracja mikrokontrolera	3
2.1	Konfiguracja pinów	5
2.2	TIM1	5
2.3	TIM2	6
2.4	TIM3 TIM4	6
2.5	TIM5	7
2.6	TIM9	7
2.7	USART	7
3	Projekt elektroniki	8
3.1	Zasilanie	8
3.2	Serwomechanizmy	8
3.3	Moduł Bluetooth	8
3.4	Sterownik silnika DC	8
3.5	Silnik DC z enkoderem	8
4	Konstrukcja mechaniczna	10
5	Opis działania programu	11
5.1	Tryb ręczny	11
5.2	Tryb automatyczny	11
6	Podsumowanie	13

1 Opis projektu

Celem projektu jest zbudowanie mobilnego manipulatora o 3 stopniach swobody (RRR) z efektoorem w postaci chwytaka.



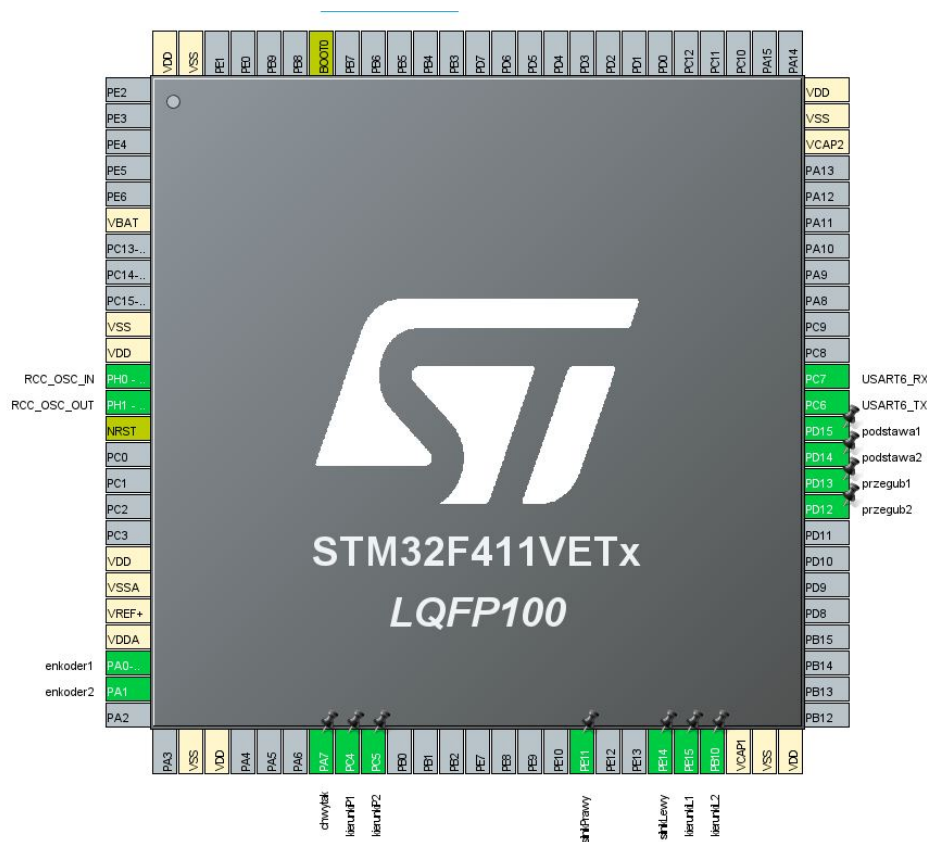
Rysunek 1: Schemat manipulatora

1.1 Założenia projektowe

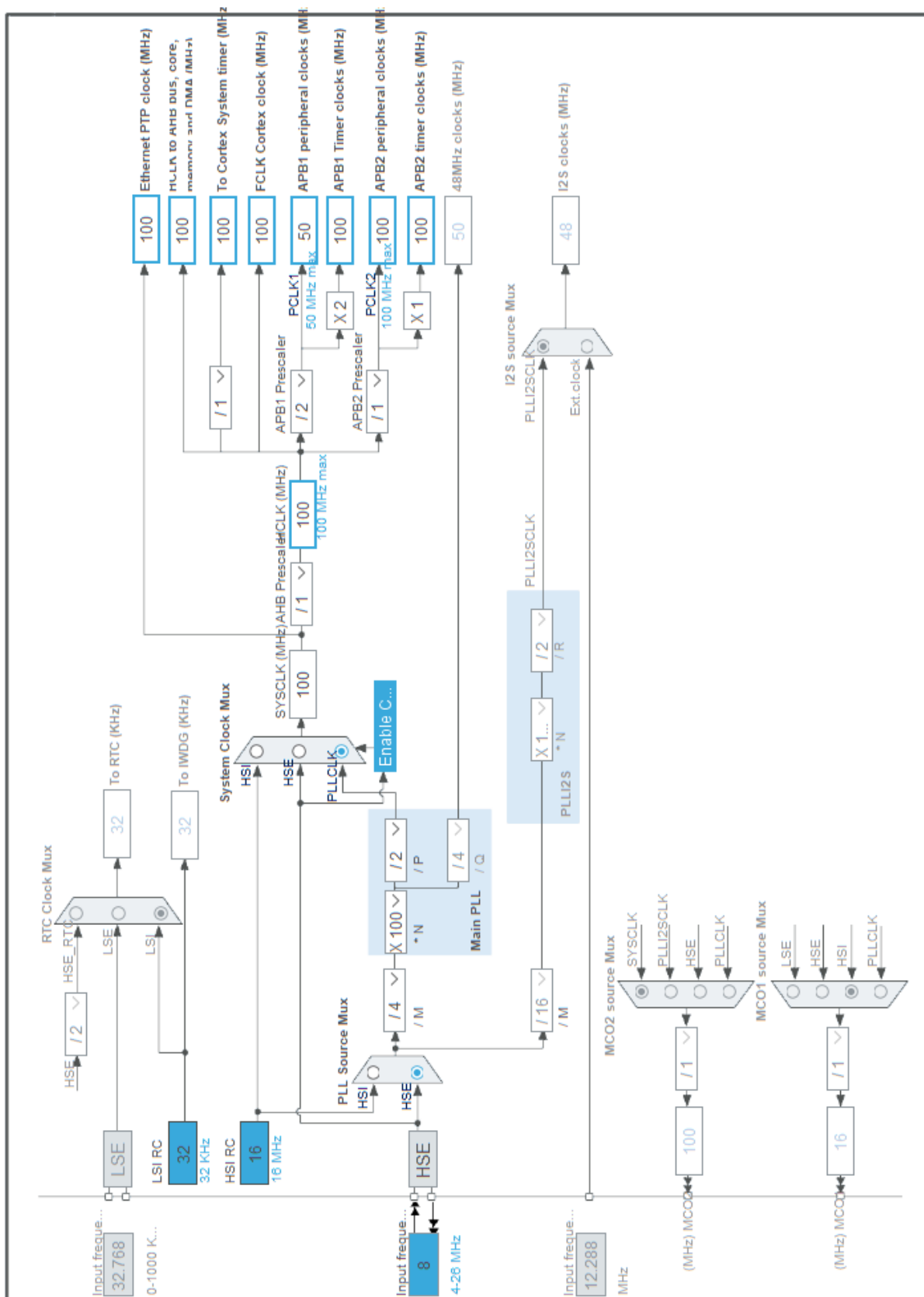
- Szkielet robota stanowić będą elementy wytworzone przy użyciu drukarki 3D.
- Ruch w przegubach oraz zaciskanie chwytaka umożliwią serwomechanizmy.
- Możliwość przemieszczania się umożliwi platforma z dwoma silnikami DC (z mostkiem H). Okrągła podstawa wraz z enkoderami w kołach pozwoli na obrót całej konstrukcji o zadany kąt wokół własnej osi.
- Sterownikiem obsługującym całość będzie płytki deweloperska DISCOVERY STM32F411VE wraz z zestawem peryferiów (silniki silników krokowych, elementy wykonawcze) wraz z sekcją zasilania.
- Praca robota w 2 trybach:
 - Automatyczny** — zadane współrzędne zewnętrzne.
 - Ręczny** — użycie kontrolera (w postaci aplikacji na system Android komunikującej się z robotem przez protokół bluetooth).
- Zasilanie będzie oparte na akumulatorze żelowym lub litowo-jonowym oraz przetwornicy DC-DC

2 Konfiguracja mikrokontrolera

Konfiguracja portów mikrokontrolera (rysunek 2) oraz zegarów (rysunek 3) przy użyciu programu STM32CubeMX.



Rysunek 2: Konfiguracja wyjść mikrokontrolera w programie STM32CubeMX



Rysunek 3: Konfiguracja zegarów mikrokontrolera

2.1 Konfiguracja pinów

Numer pinu	PIN	Tryb pracy	Funkcja/etykieta
12	PH0	RCC_OSC_IN	
13	PH1	RCC_OSC_OUT	
23	PA0	TIM5_CH1	enkoder1
23	PA1	TIM5_CH2	enkoder2
32	PA7	TIM3_CH2	chwytak
33	PC4	GPIO_Output	kierunkiP1
34	PC5	GPIO_Output	kierunkiP2
42	PE11	TIM1_CH2	silnikPrawy
45	PE14	TIM1_CH4	silnikLewy
46	PE15	GPIO_Output	kierunkiL1
47	PB10	GPIO_Output	kierunkiL2
59	PD12	TIM4_CH1	przegub2
60	PD13	TIM4_CH2	przegub1
61	PD14	TIM4_CH3	podstawa2
62	PD15	TIM4_CH4	podstawa1
63	PC6	USART1_TX	
64	PC7	USART1_RX	

Tabela 1: Konfiguracja pinów mikrokontrolera

Pin PE11 i PE14 odpowiada za prędkość obrotu silników napędowych. PC4, PC5, PE15, PB10 podłączone do mostka H obsługują kierunek obrotu silników. Następnie PA0 i PA1 to piny podłączone do tego samego enkodera, dzięki nim otrzymujemy informacje o obrocie podstawy. PC6 oraz PC7 (USART) podłączone do modułu HC05 pozwalają na komunikację poprzez bluetooth. Piny PD12-PD15 oraz PA7 sterują obrotem serwo mechanizmów dzięki sygnałowi PWM.

2.2 TIM1

Timer 1 generuje sygnały PWM które sterują prędkością obrotów silników DC. Częstotliwość została dopasowana do mostka H – 1298n dla którego optymalna wartość to 100Hz.

Parametr	Wartość
Prescaler(PSC - 16 bits value)	9999
Counter Period (AutoReload Register — 16 bits value)	Up
Counter Period	99
Internal Clock Division (CKD)	No Division
auto-reload preload	Disable
Master/Slave Mode (MSM bit)	Disable (Trigger input effect not delayed)
Trigger Event Selection	Reset (UG bit from TIMx_EGR)
PWM Generation Channel 2/4	
Mode	PWM mode 1
Pulse	0
Output compare preload	Enable
Fast mode	Disable
CH Polarity	High
CH Idle State	Reset

Tabela 2: Konfiguracja TIM1

2.3 TIM2

Timer 2 generuje przerwania z częstotliwością 25Hz dzięki którym inkrementowane bądź dekrementowane są wartości wypełnień sygnałów PWM sterujących serwomechanizmami. Dzięki temu ramie robota pracuje płynnie co zmniejsza siłę bezwładności która wpływa na manipulator.

Parametr	Wartość
Prescaler(PSC - 16 bits value)	3999
Counter Period (AutoReload Register — 16 bits value)	Up
Counter Period	999
Internal Clock Division (CKD)	No Division
auto-reload preload	Disable
Master/Slave Mode (MSM bit)	Disable (Trigger input effect not delayed)
Trigger Event Selection	Reset (UG bit from TIMx_EGR)

Tabela 3: Konfiguracja TIM2

2.4 TIM3 TIM4

Timer 3 oraz 4 generują sygnały PWM sterujące serwomechanizmami. Ich parametry zostały skonfigurowane identycznie. Zgodnie z notą katalogową, częstotliwość sygnału wynosi 50Hz. Wówczas zmiana jego wypełnienia od 0.5ms do 2.5ms pozwala na zmianie kąta obrotu serwomechanizmu.

Parametr	Wartość
Prescaler(PSC - 16 bits value)	1999
Counter Period (AutoReload Register — 16 bits value)	Up
Counter Period	999
Internal Clock Division (CKD)	No Division
auto-reload preload	Disable
Master/Slave Mode (MSM bit)	Disable (Trigger input effect not delayed)
Trigger Event Selection	Reset (UG bit from TIMx_EGR)
PWM Generation Channel 1/2/3/4	
Mode	PWM mode 1
Pulse	0
Output compare preload	Enable
Fast mode	Disable
CH Polarity	High
CH Idle State	Reset

Tabela 4: Konfiguracja TIM3 oraz TIM4

2.5 TIM5

Kanały 1 oraz 2 timera 5 zostały skonfigurowane w trybie *Encoder Mode*. Pobierają one informacje z enkodera umieszczonego na wale silnika. Jego rozdzielczość została ustawiona na 14000 (obrót całej podstawy o kąt 180 stopni wokół własnej osi wynosi ponad 6000 impulsów)

Parametr	Wartość
Prescaler(PSC - 16 bits value)	0
Counter Period (AutoReload Register — 16 bits value)	Up
Counter Period	14000
Internal Clock Division (CKD)	No Division
auto-reload preload	Disable
Master/Slave Mode (MSM bit)	Disable (Trigger input effect not delayed)
Encoder Mode	Encoder Mode TI1 and TI2
Parameters for Channel 1/2	
Polarity	Rising Edge
IC Selection	Direct
Prescaler Division Ratio	No division
Input Filter	15

Tabela 5: Konfiguracja TIM5

2.6 TIM9

Timer 9 generuje przerwania które sprawdzają aktualny kąt o który obróciła się platforma. Jeśli osiągnięto zadany kąt silniki zostają wstrzymane. Częstotliwość licznika wynosi 2000 Hz.

Parametr	Wartość
Prescaler(PSC - 16 bits value)	499
Counter Period (AutoReload Register — 16 bits value)	Up
Counter Period	99
Internal Clock Division (CKD)	No Division
auto-reload preload	Disable

Tabela 6: Konfiguracja TIM9

2.7 USART

Interfejs USART służy do komunikacji z aplikacją poprzez moduł bluetooth.

Parametr	Wartość
Baud Rate	115200
Word Length	8 Bits (including parity)
Parity	None
Stop Bits	1

Tabela 7: Konfiguracja peryferium USART

3 Projekt elektroniki

Część elektroniczną wykonano na płytce uniwersalnej. Płytkę deweloperską STM32F411VET6 umieszczono na listwach goldpin celem uproszczenia procesu montażu i stworzenia możliwości użycia jej ponownie do innych projektów.

3.1 Zasilanie

Układ zasilany jest z akumulatora o napięciu 6V i pojemności 4500mAh. Na płytce znajduje się przetwornica step-down 5V. Serwomechanizmy oraz część wykonawcza sterownika L298N zasilane są bezpośrednio z akumulatora. Część logiczna układu L298N, enkodery, moduł HC-05 oraz płytka deweloperska są zasilane z przetwornicy napięciem 5V.

3.2 Serwomechanizmy

Serwomechanizmy MG996R podłączone są za pomocą taśm do goldpinów umieszczonych na płytce. Trzecim przewodem dostarczany jest sygnał PWM. Pobór prądu uzależniony jest wychyleń od środka masy manipulatora.

3.3 Moduł Bluetooth

Moduł Bluetooth HC-05 podłączony jest do 2 pinów mikrokontrolera TX oraz RX. Komunikuje się z nim poprzez protokół USART. Umożliwia komunikację np. z telefonem.

3.4 Sterownik silnika DC

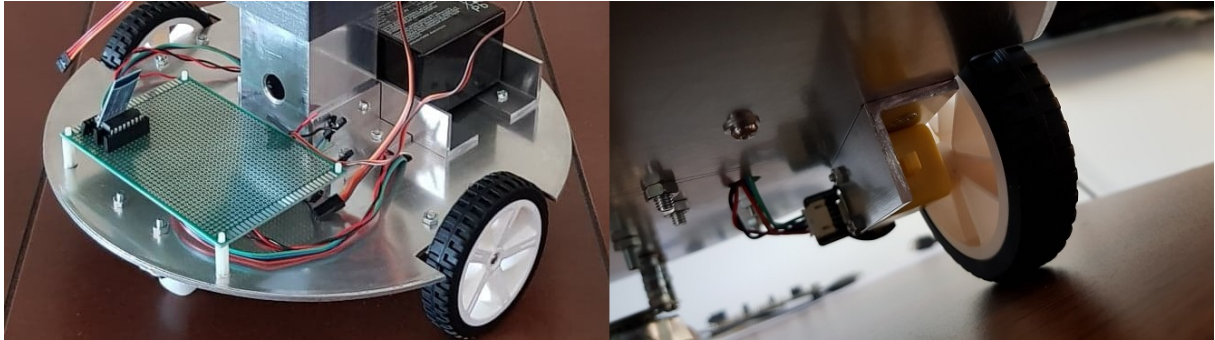
Dwukanałowy sterownik silników L298N umożliwia regulację prędkości i kierunku obrotu silnika. Do pinów *Enable* podłączony jest sygnał PWM, a do pinów *IN1-4* stan wysoki bądź niski w zależności od pożądanego kierunku. W momencie podania stanu wysokiego na oba piny, sterownik stara się wyhamować silnik.

3.5 Silnik DC z enkoderem

Silnik SJ01 podłączony jest do wyjść sterownika L298N. Moduł wyposażony jest w enkoder który umożliwia generowanie impulsów generowanych przy obrocie silnika. W tym celu 2 przewody podłączone bezpośrednio do mikrokontrolera wysyłają sygnał kwadraturowy przesunięty w fazie.

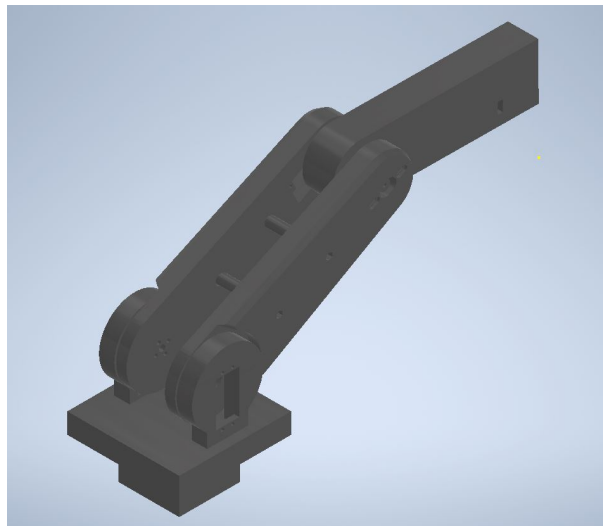
4 Konstrukcja mechaniczna

Szkielet podstawy został wykonany ręcznie, w całości z aluminium. Wszystkie elementy okrągłej platformy zostały przymocowane za pomocą śrub i nakrętek. Przy wcięciach w podstawie zostały umieszczone silniki wraz z kołami napędowymi. Pozostałe 2 koła obrotowe, jedno z przodu drugie z tyłu, zapobiegają przewróceniu się konstrukcji. Całą elektronikę od metalowych części konstrukcji oddzielają nylonowe dystanse. Pionowy profil na środku jest trzonem manipulatora. Z tyłu kątowniki stanowią kieszeń dla akumulatora.



Rysunek 4: Konstrukcja podstawy

Ramię zostało w całości wykonane w technologii druku 3D. Dzięki stronie Thingiverse oraz użytkownikowi WonderTiger skorzystano z modelu [6] na mocy licencji CC BY-NC 4.0, który następnie przerobiono dla własnych potrzeb.



Rysunek 5: Model 3D

5 Opis działania programu

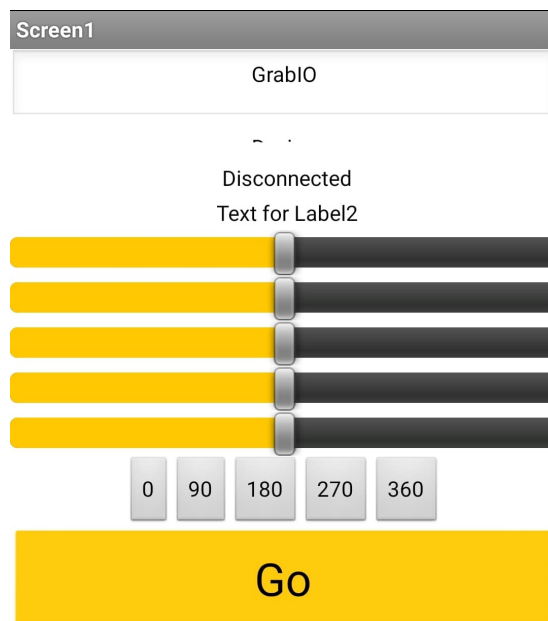
Program na początku inicjalizuje wszystkie nastawy w przegubach (modyfikuje wypełnienie sygnałów PWM), tak aby ramie było możliwie jak najbardziej złożone, po to by serwomechanizmy pobierały jak najmniejszy prąd. Następnie włącza nasłuchiwanie USART w trybie przerwań. Gdy moduł bluetooth odbierze sygnał o 4 znakach (każda operacja w aplikacji generuje 4 znakowy kod), mikrokontroler go dekoduje i w zależności od odebranych znaków podejmuje określone działanie.

5.1 Tryb ręczny

Pierwsze 3 suwaki w aplikacji mobilnej wysyłają sygnał dla serwomechanizmów manipulatora, które odpowiadają za obrót poszczególnych przegubów o zadany kąt. Mikrokontroler przelicza stopnie na wypełnienie sygnału PWM. Wszystkie przeguby poruszają się płynnie poprzez inkrementację wypełnienia w przerwaniach timera. Czwarty suwak wraz z kolejnym serwomechanizmem otwiera bądź zamyka chwytak (tutaj wartości podawane są w %). Piąty suwak o wartościach od 0 do 360 stopni odpowiada za obrót podstawy wokół własnej osi. Bezpośrednio za to działanie odpowiedzialne są silniki DC. W zależności od otrzymanego kąta program przelicza go na szacowaną pozycję enkodera i uruchamia silniki (zadając im odpowiednią prędkość sygnałem PWM oraz kierunek obrotu poprzez stan wysoki na odpowiednich wyjściach). Następnie wartość szacowana jest porównywana z wartością rzeczywistą. W momencie otrzymania większej lub mniejszej wartości, w zależności od kierunku obrotu, silniki są zatrzymywane. Oprócz suwaka aplikacja posiada przyciski dla wielokrotności kąta prostego, obrotu podstawy, które są w ten sam sposób interpretowane przez mikrokontroler, jak wcześniej wspomniany suwak. Wciśnięcie przycisku *Go* umożliwia ruch manipulatora do przodu ze stałą prędkością, a jego puszczenie zatrzymuje robota.

5.2 Tryb automatyczny

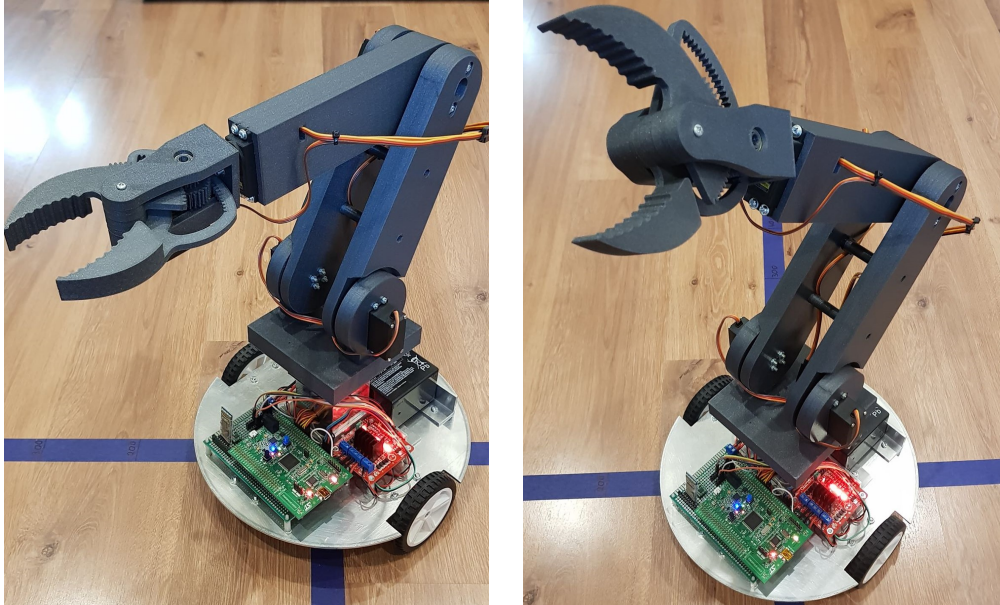
Aby umożliwić manipulatorowi pracę w zadanych współrzędnych zewnętrznych na mikrokontrolerze został zaimplementowany algorytm kinematyki odwrotnej jakim jest FABRIK. Dzięki niemu otrzymujemy informacje w jakich punktach w przestrzeni mają znaleźć się poszczególne przeguby. Znając długość ramion robota, przy pomocy przekształceń trygonometrycznych, jesteśmy w stanie wyznaczyć kąty w przegubach. Następnie program wykorzystuje wcześniej zaimplementowane rozwiązania trybu ręcznego w którym wybieramy kąty obrotu poszczególnego ramienia. Do mobilnej aplikacji zostały dodane rubryki do wpisania zadanych współrzędnych (4 znaki, długości wyrażone w mm), których edycje zatwierdzamy przyciskiem *SET*. Wówczas manipulator rozpocznie ruch w kierunku zadanego punktu, najpierw obrót podstawy a następnie ruch samego ramienia. Oprócz tego został dodany przycisk *move to home* pozwalający ustawić manipulator w konfiguracji początkowej, dla której serwomechanizmy pobierają najmniejszy prąd. Jeśli żądany przez nas punkt leży poza obszarem przestrzeni roboczej manipulatora ten nie wykona żadnego ruchu.



Rysunek 6: Aplikacja

6 Podsumowanie

Wykonano wszystkie zaplanowane zadania projektowe. Efektem pracy jest zdalnie sterowany manipulator mobilny. Użytkownik ma możliwość sterowania manipulatorem z użyciem aplikacji mobilnej na system Android przez podanie współrzędnych zewnętrznych bądź ręcznie.



Rysunek 7: Końcowy wygląd projektu

Linki do repozytoriów na platformie Github
[Raport – Github]
[Projekt STM32CubeIDE – Github]

Literatura

- [1] Billjamshedji. MeArm Robotic Arm. Feb. 2017.
- [2] D. Cook. *Robot Building for Beginners*. Technology in action series. Apress, 2010.
- [3] Dejan. DIY Arduino Robot Arm with Smartphone Control. Sept. 2018.
- [4] W. Dowski. Sterowniki robotów, Laboratorium – Wprowadzenie, Wykorzystanie narzędzi STM32CubeMX oraz SW4STM32 do budowy programu mrugającej diody z obsługą przycisku. Mar. 2017.
- [5] C. Nethmal. Designing a Micro Robot on a PCB. Dec. 2019.
- [6] M. Neuhaus. Robot Arm. Sept. 2016.
- [7] STMicroelectronics. User manual Discovery kit with STM32F411VE MCU . Dec. 2014.
- [8] K. Tchoń. *Manipulatory i roboty mobilne: modele, planowanie ruchu, sterowanie*. Problemy Współczesnej Nauki, Teoria i Zastosowania. Robotyka. Akademicka Oficyna Wydawnicza PLJ, 2000.
- [9] X. Treker, wikinc1. Mostek H (H-bridge) - kompendium dla robotyka. Sept. 2006.
- [10] L. Tyoan. Magician Mobile Robot Arm. Oct. 2014.
- [11] M. A. K. Yusoff, R. E. Samin, and B. S. K. Ibrahim. Wireless mobile robotic arm. *Procedia Engineering*, 41:1072 – 1078, 2012. International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012).