

PROJEKT

STEROWNIKI ROBOTÓW

Raport

Manipulator

-

Skład grupy (1):

Aleksander BOJDA, 218675

Piotr STOPYRA, 218552

Termin: wtTN13

Prowadzący:

mgr inż. Wojciech DOMSKI

20 czerwca 2017

Spis treści

1	Opis projektu	2
2	Konfiguracja mikrokontrolera	3
3	Konfiguracja peryferiów	5
3.1	ADC1	5
3.2	I2C1	5
3.3	RCC	5
3.4	SYS	5
3.5	TIM3	5
3.6	TIM15	6
3.7	TIM16	6
3.8	UART5	6
3.9	USB	6
3.10	USB_DEVICE	6
4	Opis funkcji pinów	7
5	Opis wykorzystanych układów zewnętrznych	8
6	Opis działania programu	9
6.1	Sterowanie joystickiem we współrzędnych wewnętrznych	9
6.2	Sterowanie joystickiem we współrzędnych zewnętrznych	9
6.3	Sterowanie poprzez Bluetooth (z wykorzystaniem protokołu UART)	9
6.4	Sterowanie z aplikacji komputerowej (interfejs USB)	9
7	Algorytm kinematyki odwrotnej	10
8	Mechanika	11
8.1	Konstrukcja	11
8.2	Napęd	11
9	Elektronika	12
9.1	Mikrokontroler	12
9.2	Zasilanie	12
9.3	Peryferia zewnętrzne	12
9.4	Schemat ideowy	12
9.5	Płytki PCB	12
10	Realizacja zadań	15
10.1	Mechanika	15
10.2	Elektronika	15
10.3	Program	15
11	Podsumowanie	15
	Bibliografia	17

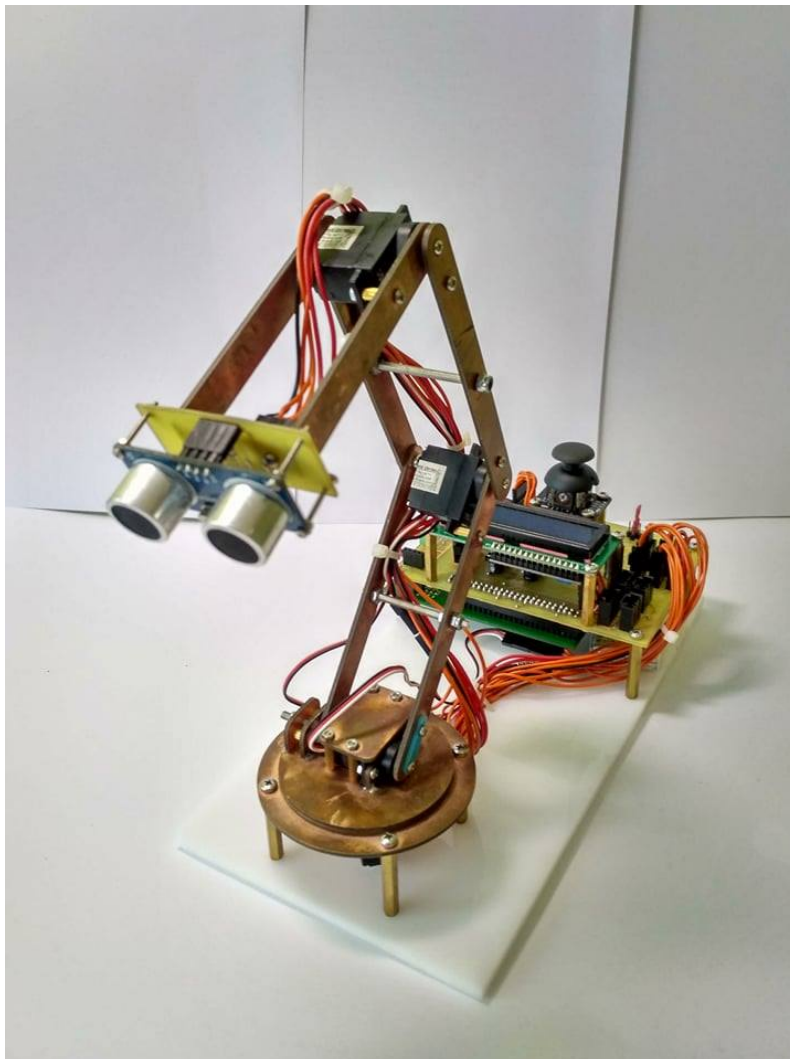
1 Opis projektu

W ramach projektu powstał manipulator szeregowy o czterech stopniach swobody. Konstrukcja wykonana została z laminatu miedzianego, a jej napęd oparty jest o 9-gramowe serwa modelarskie. Moduł sterowania wyposażony jest w płytke deweloperską STM32F3Discovery mocowaną na płytce rozszerzeniowej zawierającej m. in. ekran LCD, joystick, sekcję zasilania oraz złącza przygotowane do zamontowania serw, modułu Bluetooth, enkoderów oraz efektora.

Cały układ zasilany jest z akumulatora LiPo 2S. Serwomechanizmy oraz część logiczna układu zasilane są z napięć 6V i 5V uzyskiwanych przy użyciu stabilizatorów LM7806 oraz LM1117.

Urządzenie może pracować w jednym z czterech trybów:

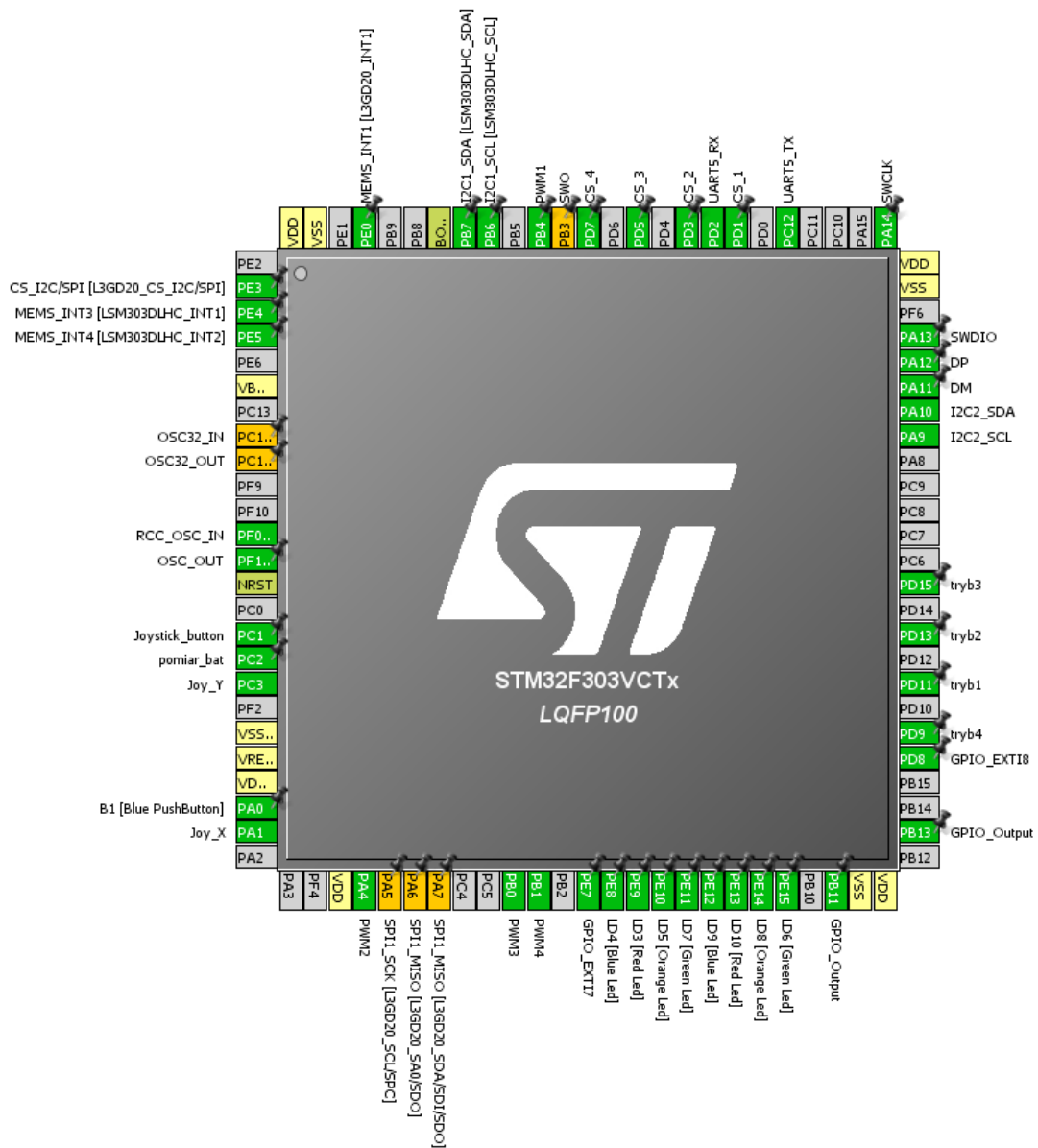
- Sterowanie joystickiem we współrzędnych wewnętrznych
- Sterowanie joystickiem we współrzędnych zewnętrznych
- Sterowanie poprzez Bluetooth (z wykorzystaniem protokołu UART) z aplikacji mobilnej
- Sterowanie z aplikacji komputerowej (interfejs USB)



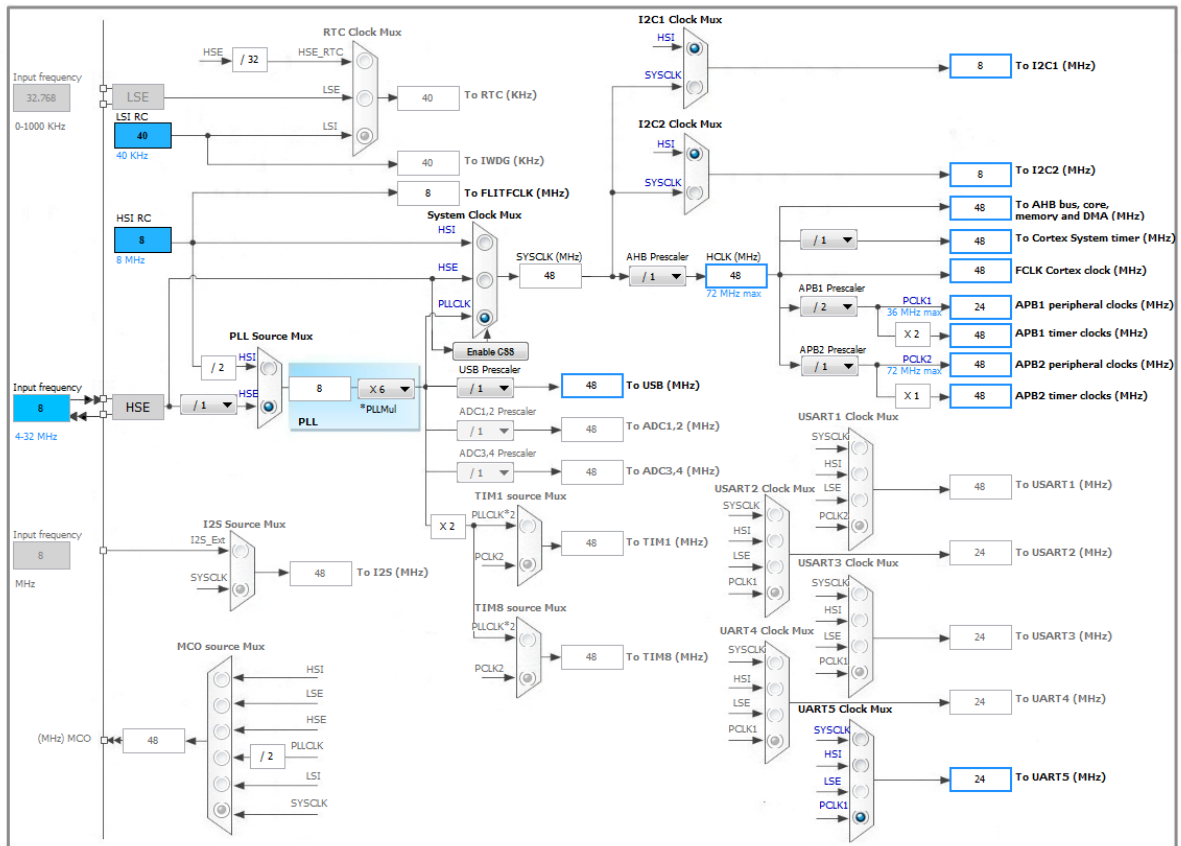
Rysunek 1: Manipulator

2 Konfiguracja mikrokontrolera

Konfiguracja portów MCU wygenerowana w programie STM32CubeMX widoczna jest na rysunku 2. Konfiguracja zegarów przedstawiona jest na rysunku 3.



Rysunek 2: Konfiguracja portów MCU



Rysunek 3: Konfiguracja zegarów

3 Konfiguracja peryferiów

Poniższe tabele przedstawiają konfigurację wszystkich wykorzystywanych w projekcie peryferiów.

3.1 ADC1

Kanały 2 i 9 wykorzystywane są do obsługi joysticka. Skonfigurowano je w trybie *Single-ended* z wykorzystaniem DMA.

Tabela 1: ADC1 - uproszczona konfiguracja

Mode	Independent mode
Clock Prescaler	Synchronous clock mode divided by 4
Resolution	ADC 12-bit resolution
Data Alignment	Right alignment
Scan Conversion Mode	Enabled
Contionous Conversion Mode	Enabled
DMA Continous Requests	Enabled
End of Conversion Selection	End of single conversion
Number of Conversions	2
Sampling time - Channel 2	181.5 Cycles
Sampling time - Channel 9	181.5 Cycles

3.2 I2C1

Wykorzystywany do komunikacji z ekranem LCD.

Tabela 2: I2C1 - uproszczona konfiguracja

I2C Speed Mode	Standard mode
I2C Speed Frequency	100 kHz

3.3 RCC

High Speed Clock (HSE): Crystal/Ceramic Resonator

3.4 SYS

Tabela 3: SYS - uproszczona konfiguracja

Debug	Serial Wire
Timebase Source	SysTick

3.5 TIM3

Wykorzystywany do sterowania serwomechanizmami - korzysta z DMA

Channel1: PWM Generation CH1

Channel2: PWM Generation CH2

Channel3: PWM Generation CH3

Channel4: PWM Generation CH4

Tabela 4: TIM3 - uproszczona konfiguracja

Prescaler	959
Counter Period (ARR)	999
Channel 1 - mode	PWM mode 1
Channel 2 - mode	PWM mode 1
Channel 3 - mode	PWM mode 1
Channel 4 - mode	PWM mode 1

3.6 TIM15

Wykorzystywany jako podstawa czasowa do wyboru trybu pracy

Tabela 5: TIM15 - uproszczona konfiguracja

Prescaler	479
Counter Period (ARR)	9999

3.7 TIM16

Wykorzystywany jako podstawa czasowa do sterowania joystickiem

Tabela 6: TIM16 - uproszczona konfiguracja

Prescaler	479
Counter Period (ARR)	999

3.8 UART5

Wykorzystywany do komunikacji z aplikacją mobilną poprzez moduł Bluetooth

Tabela 7: UART5 - uproszczona konfiguracja

Baud Rate	9600
Word Length	8 Bits (including parity)
Parity	None
Stop Bits	1
Data Direction	Receive and Transmit

3.9 USB

Wykorzystywany do komunikacji z aplikacją komputerową

Tabela 8: USB - uproszczona konfiguracja

Mode	Device (FS)
Speed	Full Speed 12MBit/s

3.10 USB_DEVICE

Wykorzystywany do komunikacji z aplikacją komputerową

Tabela 9: USB_DEVICE - uproszczona konfiguracja

Class	Communication Device Class (Virtual Port Com)
--------------	---

4 Opis funkcji pinów

Tabela 10: Tryby działania pinów oraz ich funkcje

Numer pinu:	PIN:	Tryb pracy:	Funkcja:
16	PC1	GPIO EXTI1	Przycisk Joysticka
17	PC2	ADC2 IN8	Pomiar napięcia na baterii
18	PC3	ADC1 IN9	Oś Y Joysticka
24	PA1	ADC1 IN2	Oś X Joysticka
29	PA4	TIM3 CH2	Sterowanie drugim serwem
35	PB0	TIM3 CH3	Sterowanie trzecim serwem
36	PB1	TIM3 CH4	Sterowanie czwartym serwem
38	PE7	GPIO EXTI7	Obsługa efektora
48	PB11	GPIO Output	Obsługa efektora
52	PB13	GPIO Output	Obsługa efektora
56	PD9	GPIO EXTI9	Wybór trybu 4
58	PD11	GPIO EXTI11	Wybór trybu 1
60	PD13	GPIO EXTI13	Wybór trybu 2
62	PD15	GPIO EXTI15	Wybór trybu 3
68	PA9	I2C2 SCL	Linia SCL wyświetlacza LCD
69	PA10	I2C2 SDA	Linia SDA wyświetlacza LCD
80	PC12	UART5 TX	Linia TX UART5
83	PD2	UART5 RX	Linia RX UART5
90	PB4	TIM3 CH1	Sterowanie pierwszym serwem

5 Opis wykorzystanych układów zewnętrznych

W projekcie wykorzystane zostały następujące układy zewnętrzne:

- Ekran LCD - Zgodny z HD44780 [1], wyposażony w ekspander PCF8574AT [2], pozwalający na obsługę wyświetlacza przy pomocy interfejsu I2C. Poniżej widoczny jest fragment kodu wraz z komentarzami, przedstawiający inicjalizację wyświetlacza:

```
1 #define LCD_ADDRESS 0x38 // Adres wyświetlacza LCD
2 #define EN_HIGH 0x04 // Włączenie pinu EN na wyświetlaczu
3 #define LIGHT_ON 0x08 // Włączenie podświetlenia na wyświetlaczu
4
5 void LCD_Init()
6 {
7     /* Tablica przechowuje komendy wyświetlacza LCD */
8     /* 1 2 3 4 5 6 7 8 9 10 11 12 */
9     uint8_t tab[12] = {0x30, 0x30, 0x30, 0x20, 0x20, 0x80, 0x00, 0x80, 0x00, 0x60, 0x00, 0
10                        xC0};
11     uint8_t tmp[12][2];
12
13     for(int i = 0; i < 12; i++)
14     {
15         tmp[i][0] = tab[i]|EN_HIGH|LIGHT_ON;
16         tmp[i][1] = tab[i]|LIGHT_ON;
17     }
18     HAL_Delay(100);
19
20     /* Inicjalizacja wyświetlacza */
21     for(int i = 0; i < 12; i++)
22     {
23         HAL_I2C_Master_Transmit(&hi2c2, LCD_ADDRESS << 1, tmp[i], 2, 100);
24         HAL_Delay(5);
25     }
26
27     /* 1,2,3 - wejście w tryb programowania (D4 i D5 włączone)
28     * 4 - tryb 4 bitowy (D5 włączone)
29     * 5 - 2 linie (D5 włączone)
30     * 6 - rozmiar 5x7 (D7 włączone)
31     * 7 - (0)
32     * 8 - kursor, wyświetlacz i miganie kursora wyłączony (D7 włączone)
33     * 9 - (0)
34     * 10 - kursor inkrementuje z lewej do prawej, wyświetlacz nie przesuwa się (D5 i D6
35            włączone)
36     * 11 - (0)
37     * 12 - włączenie wyświetlacza (D7 i D6 włączone)
38     *
39     * D7 - 1000 0000 - 0x80
40     * D6 - 0100 0000 - 0x40
41     * D5 - 0010 0000 - 0x20
42     * D4 - 0001 0000 - 0x10
43     * BL - 0000 1000 - 0x08
44     * EN - 0000 0100 - 0x04
45     * RW - 0000 0010 - 0x02
46     * RS - 0000 0001 - 0x01
47     */
48 }
```

- Joystick z dwoma wyjściami analogowymi (obsługiwanymi przez przetwornik ADC) oraz przyciskiem
- Moduł Bluetooth (komunikacja przez protokół UART) - wykorzystywany do komunikacji z aplikacją mobilną
- Ultradźwiękowy czujnik odległości HC-SR04 wykorzystywany w roli przykładowego efektora

6 Opis działania programu

Dostępne są cztery tryby działania, pozwalające na obsługę manipulatora. Aby wybrać dany tryb należy włożyć zworkę w odpowiednie, podpisane miejsce na płytce PCB. Sprawdzanie aktualnego trybu działania odbywa się z częstotliwością 10Hz w przerwaniu czasowym od timera 15.

6.1 Sterowanie joystickiem we współrzędnych wewnętrznych

Za pomocą przesuwania joysticka w pionie dokonywany jest wybór przegubu, przy przesuwaniu w poziomie w zależności od kierunku wychylenia wartość kąta danego przegubu zwiększa się lub zmniejsza. Aktualny przegub oraz wartość jego kąta widoczne są na wyświetlaczu LCD.

6.2 Sterowanie joystickiem we współrzędnych zewnętrznych

Za pomocą przesuwania joysticka w pionie dokonywany jest wybór osi według której dokonywana jest zmiana położenia, zaś przesuując joystick w poziomie zmienia się wartość przesunięcia w danej osi. Aktualna oś oraz wartość przesunięcia widoczne są na wyświetlaczu LCD.

6.3 Sterowanie poprzez Bluetooth (z wykorzystaniem protokołu UART)

Pozwala na sterowanie we współrzędnych wewnętrznych i zewnętrznych korzystając z aplikacji na telefonie.

6.4 Sterowanie z aplikacji komputerowej (interfejs USB)

Pozwala na sterowanie we współrzędnych wewnętrznych i zewnętrznych za pomocą aplikacji na komputerze.

7 Algorytm kinematyki odwrotnej

W celu uzyskania możliwości operowania efektorom we współrzędnych zewnętrznych został zaimplementowany iteracyjny algorytm FABRIK [4] (Forward And Backward Reaching Inverse Kinematic). Został on wybrany ze względu na dużą szybkość działania w porównaniu do innych algorytmów, co jest szczególnie istotne przy stosunkowo małej mocy obliczeniowej mikrokontrolera. Porównanie działania popularnych algorytmów kinematyki odwrotnej oraz algorytmu FABRIK widoczne jest na rysunku 4.

Reconstruction comparison. Average results (over 20 runs).

	Number of Iterations	Median time †	Time per iteration †	Median Error (mm)
FABRIK	65	1.6	0.0246	58.68
CCD	67	20.5	0.3060	69.99
J.Transpose	1352	1928.0	1.4334	137.42
J.DLS	804	1533.0	1.9067	84.84
J.SVD-DLS	723	1494.0	2.0664	83.73

† This is a MATLAB executable time in msec

Rysunek 4: Porównanie algorytmów kinematyki odwrotnej [4]

Algorytm FABRIK zwraca współrzędne położenia każdego z przegubów w ten sposób, aby między przegubami zachowane były odpowiednie odległości odpowiadające długościom ogniw, a także położenie efektora zgadzało się (z pewną zadaną tolerancją) z położeniem docelowym. Po uzyskaniu współrzędnych każdego z przegubów przeliczane są wartości kątów, w ten sposób, żeby przeguby przemieściły się z aktualnej pozycji do docelowej.

8 Mechanika

8.1 Konstrukcja

Konstrukcja robota wykonana została z laminatu miedzianego o grubości 1.6mm. Materiał ten został wybrany ze względu na łatwość obróbki, wytrzymałość oraz małą wagę. Z laminatu zostały wykonane wszystkie główne elementy konstrukcyjne robota takie jak obrotowa podstawa, ogniwa oraz uchwyty montażowe. Dodatkowo jako łączniki wykorzystane zostały metalowe elementy dystansowe. Całość konstrukcji została połączona za pomocą śrub M2 i M3 oraz za pomocą cyny lutowniczej. Projekt konstrukcji mechanicznej został wykonany w programie Autodesk Inventor Professional 2015 i jest widoczny na rysunku 5.

8.2 Napęd

Manipulator jest napędzany za pomocą czterech serwomechanizmów. Wybrano micro serwomechanizmy Redox S90 MG posiadające metalowe tryby oraz stosunkowo duży moment siły (2,55 kg*cm dla zasilania 4,8V) w porównaniu do ich niewielkiej masy (10g).



Rysunek 5: Model manipulatora

9 Elektronika

9.1 Mikrokontroler

Jednostką sterującą jest mikrokontroler STM32F303VC [3] znajdujący się na płytce deweloperskiej Discovery F3. Najważniejsze parametry mikrokontrolera widoczne są w tabeli 11.

Tabela 11: Parametry mikrokontrolera STM32F303VC

Napięcie zasilania:	2.0 - 3.6V
Pamięć Flash:	256kB
Maksymalna częstotliwość taktowania:	72MHz
Ilość wejść/wyjść:	87
Ilość timerów:	13
Interfejsy komunikacyjne:	1x CAN, 2x I2C, 5x UART, 3x SPI, 1x USB

9.2 Zasilanie

Manipulator zasilany jest za pomocą akumulatora Li-Po o napięciu znamionowym 7.4V. Napięcie zasilania obniżane jest do napięć 5V (LM1117) oraz 6V (LM7806), każde napięcie dostarczane jest przez połączone ze sobą równolegle dwa stabilizatory w celu zwiększenia wydajności prądowej. Płytkę deweloperską zasilana jest napięciem 5V. W celu zabezpieczenia układu zasilania zastosowano tranzystor N-MOSFET chroniący układ przed podłączeniem odwrotnej polaryzacji. W celu ochrony baterii oraz układu przed zwarcieniem zastosowano bezpiecznik polimerowy o prądzie znamionowym równym 2A.

9.3 Peryferia zewnętrzne

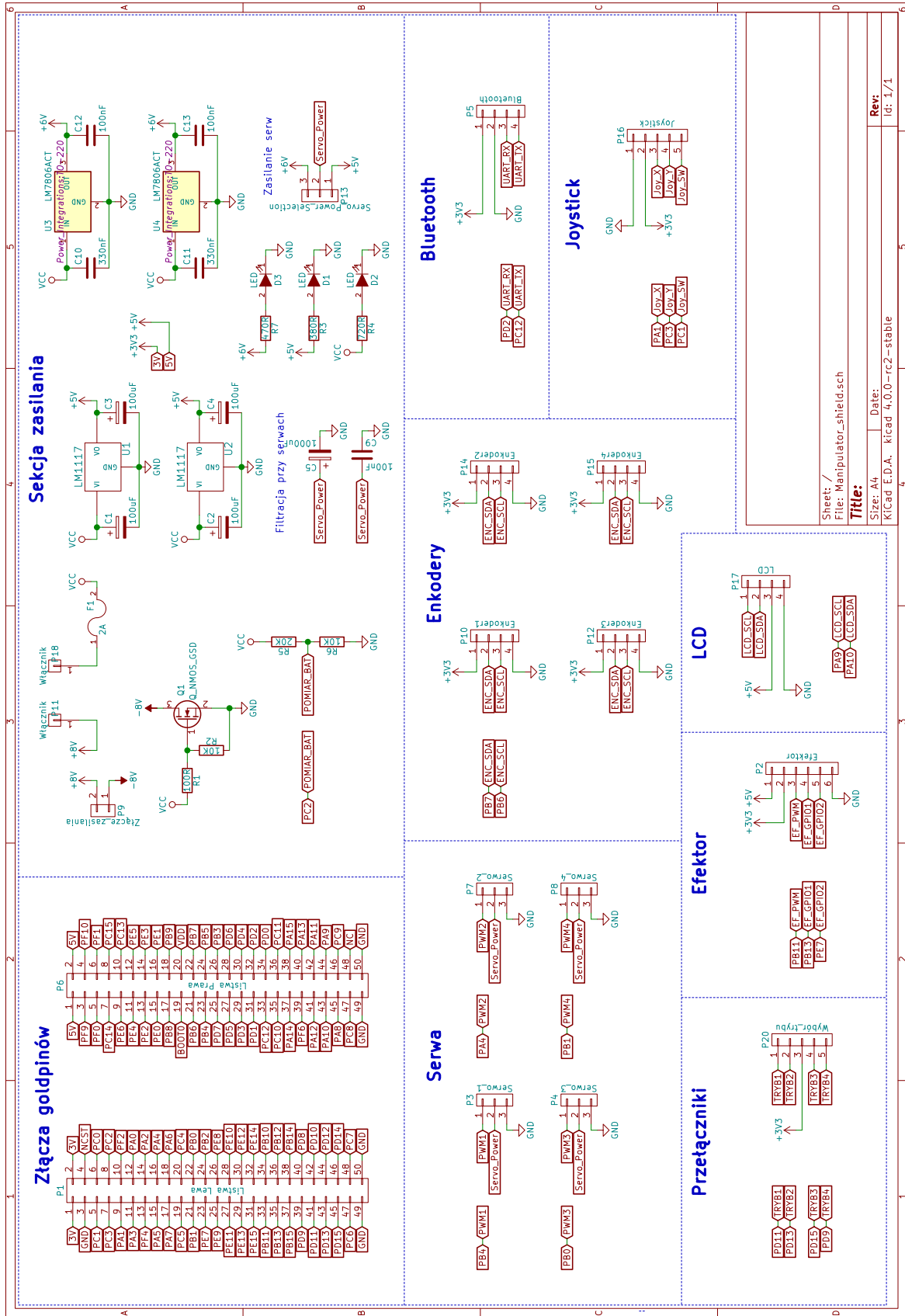
Peryferia zewnętrzne podłączane są za pomocą złącz z kluczem, zapobiegającym odwrotnemu podłączeniu lub wypdaniu złącza z gniazda oraz zwykłych gniazd goldpin.

9.4 Schemat ideowy

Schemat ideowy układu widoczny jest na rysunku 6.

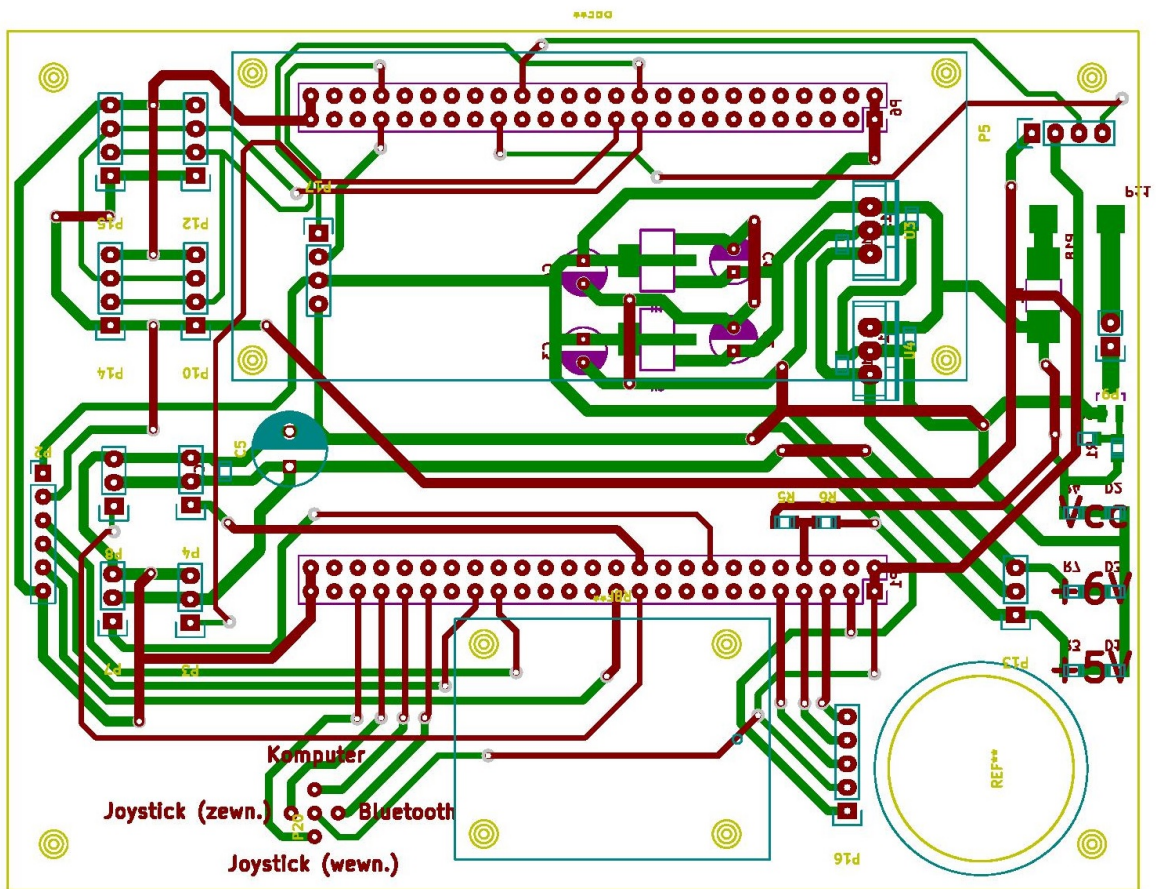
9.5 Płytki PCB

Projekt płytki PCB widoczny jest na rysunku 7.



Sheet: /
 File: Manipulator_shield.sch
Title:
 Size: A4 Date:
 Kicad E.D.A. kicad 4.0.0-r2-stable
 Rev:
 Id: 1/1

Rysunek 6: Schemat ideowy



Rysunek 7: Schemat płytki PCB

10 Realizacja zadań

10.1 Mechanika

Część mechaniczna została ukończona w 100%.

Zadania zrealizowane:

- Projekt mechaniki w programie Autodesk Inventor
- Wycięcie elementów konstrukcyjnych z laminatu
- Montaż części mechanicznej

10.2 Elektronika

Część elektroniczna została ukończona w 100%.

Zadania zrealizowane:

- Zaprojektowanie schematu ideowego
- Zaprojektowanie płytki PCB
- Wytrawienie płytki PCB
- Lutowanie elementów elementów elektronicznych

10.3 Program

Proces programowania został ukończony w 100%.

Zadania zrealizowane:

- Konfiguracja peryferiów w STM32CubeMX
- Powstanie podstawowej pętli sterowania i wyboru trybu
- Sterowanie serwomechanizmami
- Obsługa joysticka
- Obsługa ekranu LCD
- Komunikacja Bluetooth z aplikacją mobilną
- Implementacja algorytmu odwrotnej kinematyki

11 Podsumowanie

Modułowość całej konstrukcji zapewnia łatwość wymiany elementów w razie ich zużycia lub zepsucia oraz umożliwia montaż różnych efektorów wykorzystywanych do realizacji różnych zadań.

Po ukończeniu zajęć projektowych planowany jest dalszy rozwój konstrukcji uwzględniający m. in.:

- Montaż enkoderów i wykorzystanie ich do dokładniejszego sterowania pozycją
- Stworzenie kilku wymiennych efektorów (np. chwytak)
- Wykorzystanie zewnętrznego układu inercyjnego do sterowania pozycją efektora we współrzędnych zewnętrznych

Literatura

- [1] HD47780 Datasheet. <https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>.
- [2] PCF8574 Datasheet. <http://www.ti.com/lit/ds/symlink/pcf8574.pdf>.
- [3] STM32F303VC Datasheet. <http://www.st.com/content/ccc/resource/technical/document/datasheet/f2/1f/e1/41/ef/59/4d/50/DM00058181.pdf/files/DM00058181.pdf/jcr:content/translations/en.DM00058181.pdf>.
- [4] Andreas Aristidou and Joan Lasenby. FABRIK: a fast, iterative solver for the Inverse Kinematics problem. *Graphical Models*, 2011.