Projekt

Sterowniki robotów

Raport

Robot mobilny typu MicroMouse RAMZES

Skład grupy:
Agata Smoląg, 235279
Termin: srTP15

Prowadzący:
mgr inż. Wojciech Domski

5 czerwca 2019

Spis treści

1 Opis projektu

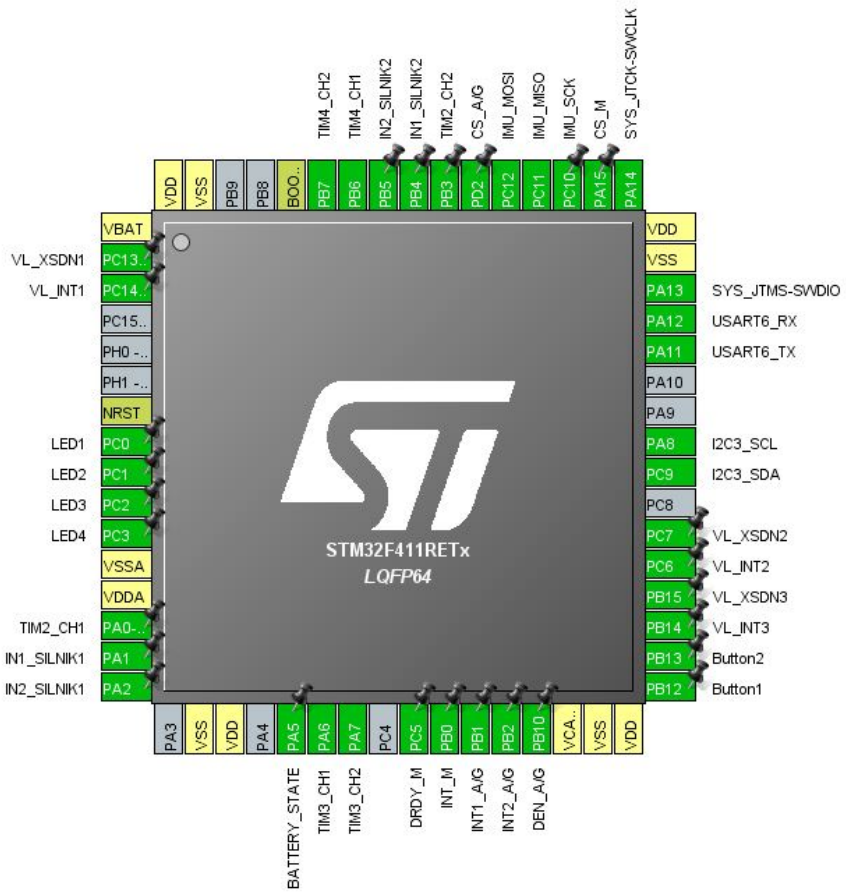
Głownym celem projektu jest budowa robota mobilnego klasy micromouse. Jest to platforma mobilna klasy (2,0) mogąca w przyszłości startować w zawodach robotycznych. Do lokalizacji w labiryncie robot będzie wykorzystywał system czujników takich jak: enkodery, IMU [4], dalmierze laserowe [6].

1.1 Założenia projektowe

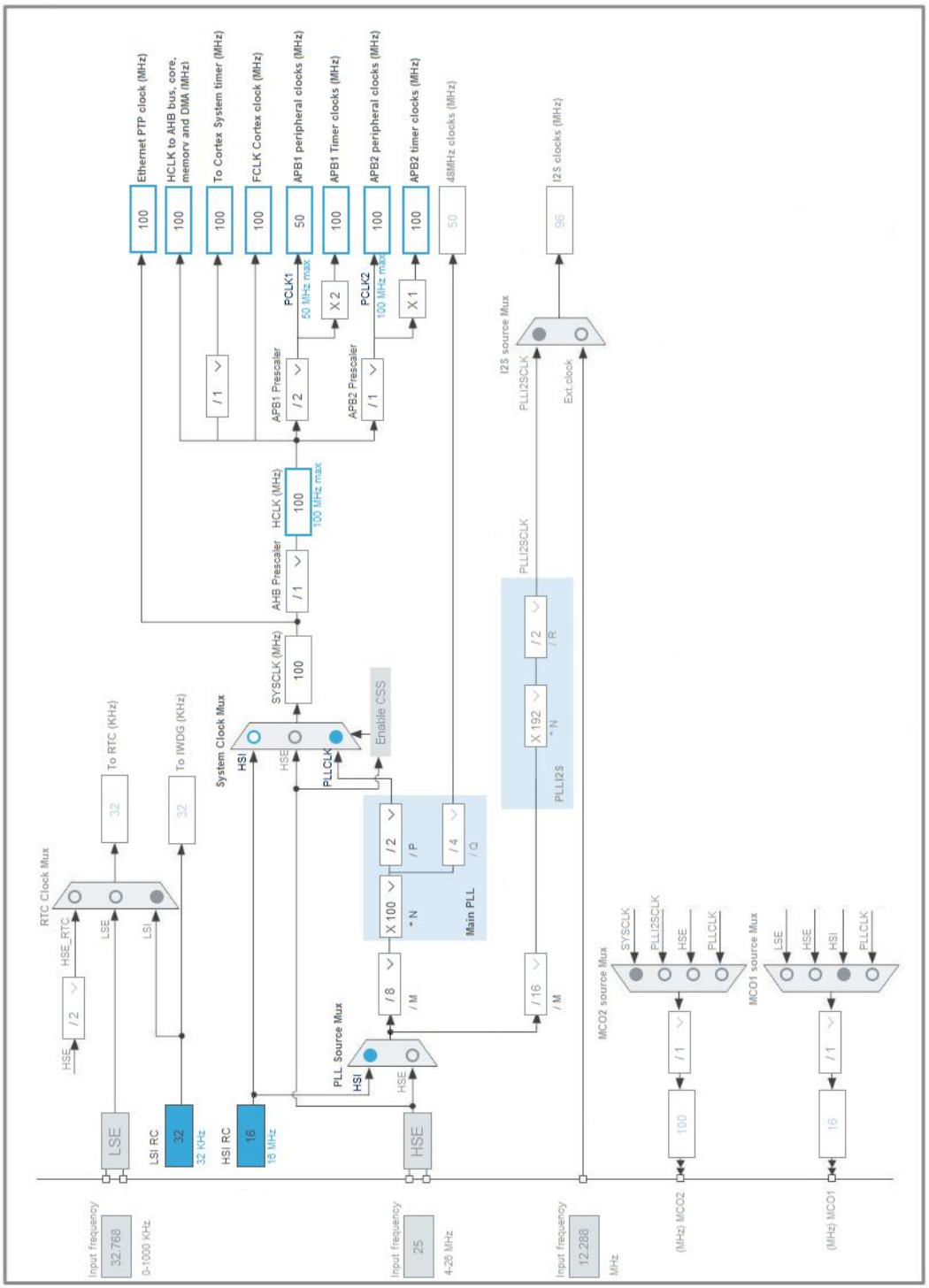
- Wykorzystanie mikrprocesora STM32F411RET6.
- Wykorzystanie enkoderów, IMU, dalmierzy laserowych do określenia położenia robota w labiryncie.
- Maksymalna szerokość robota w konkurencji micromouse nie może przekraczać 110mm.
- Optymalizacja wagi robota.

2 Konfiguracja mikrokontrolera

Do projektu wybrano mikrokontroler STM32F411RET6 [5] w obudowie LQFP64. Wybór podyktowany został tym, że posiada on wystarczającą moc obliczeniową, ilość potrzebnych peryferiów oraz pamieć flash. Na zdjęciu poniżej umieszczono ostateczna konfigurację mikroprocesor na podstawie której mozna wygenerować pierwszy podstawowy program.



Rysunek 1: Konfiguracja wyjść mikrokontrolera w programie STM32CubeMX



Rysunek 2: Konfiguracja zegarów mikrokontrolera

2.1 Konfiguracja pinów

Numer pinu	PIN	Tryb pracy	Funkcja/etykieta
2	PC13-ANTI_TAMP	GPIO_Input	VL_XSDN1
3	PC14-OSC32_IN	GPIO_Input	VL_INT1
8	PC0	GPIO_Output	LED1
9	PC1	GPIO_Output	LED2
10	PC2	GPIO_Output	LED3
11	PC3	GPIO_Output	LED4
14	PA0-WKUP	TIM2_CH1	
15	PA1	GPIO_Output	IN1_SILNIK1
16	PA2	GPIO_Output	IN2_SILNIK1
21	PA5	ADC1_IN5	BATTERY_STATE
22	PA6	TIM3_CH1	
23	PA7	TIM3_CH2	
25	PC5	GPIO_Input	DRDY_M
26	PB0	GPIO_EXTI0	INT_M
27	PB1	GPIO_EXTI1	INT1_A/G
28	PB2	GPIO_EXTI2	INT2_A/G
29	PB10	GPIO_Output	DEN_A/G
33	PB12	GPIO_Input	Button1
34	PB13	GPIO_Input	Button2
35	PB14	GPIO_EXTI14	VL_INT3
36	PB15	GPIO_Input	VL_XSDN3
37	PC6	GPIO_EXTI6	VL_INT2
38	PC7	GPIO_Input	VL_XSDN2
40	PC9	I2C3_SDA	
41	PA8	I2C3_SCL	
44	PA11	USART6_TX	
45	PA12	USART6_RX	
46	PA13	SYS_JTMS-SWDIO	
49	PA14	SYS_JTCK-SWCLK	
50	PA15	GPIO_Output	CS_M
51	PC10	SPI3_SCK	IMU_SCK
52	PC11	SPI3_MISO	IMU_MISO
53	PC12	SPI3_MOSI	IMU_MOSI
54	PD2	GPIO_Output	CS_A/G
55	PB3	TIM2_CH2	
56	PB4	GPIO_Output	IN1_SILNIK2
57	PB5	GPIO_Output	IN2_SILNIK2
58	PB6	TIM4_CH1	
59	PB7	TIM4_CH2	

Tabela 1: Konfiguracja pinów mikrokontrolera

2.2 ADC1

To peryferium jest wykorzystywane do pomiaru napięcia na baterii.

ADC_Settings:	**Wartość:**
Clock Prescaler	PCLK2 divided by 4
Resolution	12 bits (15 ADC Clock cycles)
Data Alignment	Right alignment
Scan Conversion Mode	Disabled
Continuous Conversion Mode	Disabled
Discontinuous Conversion Mode	Disabled
DMA Continuous Requests	Disabled
End Of Conversion Selection	EOC flag at the end of single channel conversion
ADC_Regular_ConversionMode:	
Number Of Conversion	1
External Trigger Conversion Source	Regular Conversion launched by software
External Trigger Conversion Edge	None
Rank	1
Channel	Channel 5
Sampling Time	3 Cycles
ADC_Injected_ConversionMode:	
Number Of Conversions	0
WatchDog:	
Enable Analog WatchDog Mode	false

Tabela 2: Konfiguracja peryferium ADC1

2.3 I2C

To peryferium jest wykorzystywane do komunikacji mikrokontrolera z dalmierzami laserowymi.

Parametr:	**Wartość:**
Master Features:	
I2C Speed Mode	Standard Mode
I2C Clock Speed (Hz)	100000
Slave Features:	
Clock No Stretch Mode	Disabled
Primary Address Length selection	7-bit
Dual Address Acknowledged	Disabled
Primary slave address	0
General Call address detection	Disabled

Tabela 3: Konfiguracja peryferium I2C

2.4 SPI3

To peryferium jest wykorzystywane do komunikacji mikrokontrolera z IMU.

Parametr:	**Wartość:**
Basic Parameters:	
Frame Format	Motorola
Data Size	8 Bits
First Bit	MSB First
Clock Parameters:	
Prescaler (for Baud Rate)	**32 ***
Baud Rate	**1.5625 MBits/s ***
Clock Polarity (CPOL)	Low
Clock Phase (CPHA)	1 Edge
Advanced Parameters:	
CRC Calculation	Disabled
NSS Signal Type	Software

Tabela 4: Konfiguracja peryferium SPI3 Full-Duplex

2.5 TIM2

To peryferium jest wykorzystywane do zadawania sygnału PWM na silniki.

Parametr:	**Wartość:**
Counter Settings:	
Prescaler (PSC - 16 bits value)	**4**
Counter Mode	Up
Counter Period (AutoReload Register - 32 bits value)	**999**
Internal Clock Division (CKD)	No Division
Trigger Output (TRGO) Parameters:	
Master/Slave Mode (MSM bit) Disable (Trigger input effect not delayed)	
Trigger Event Selection	Reset (UG bit from TIMx_EGR)
PWM Generation Channel 1:	
Mode PWM mode	1
Pulse (32 bits value)	0
Fast Mode	Disable
CH Polarity	High
PWM Generation Channel 2:	
Mode PWM mode	1
Pulse (32 bits value)	0
Fast Mode	Disable
CH Polarity	High

Tabela 5: Konfiguracja peryferium TIM2

2.6 TIM3 i TIM4

To peryferium jest wykorzystywane do odczytywania informacji z enkoderów.

Parametr:	**Wartość:**
Counter Settings:	
Prescaler (PSC - 16 bits value)	0
Counter Mode	Up
Counter Period (AutoReload Register - 16 bits value)	**65535**
Internal Clock Division (CKD)	No Division
Trigger Output (TRGO) Parameters:	
Master/Slave Mode (MSM bit)	Disable (Trigger input effect not delayed)
Trigger Event Selection	Reset (UG bit from TIMx_EGR)
Encoder:	
Encoder Mode	**Encoder Mode TI1 and TI2**
_____ Parameters for Channel 1 _____	
Polarity	Rising Edge
IC Selection	Direct
Prescaler Division Ratio	No division
Input Filter	**15**
_____ Parameters for Channel 2 _____	
Polarity	Rising Edge
IC Selection	Direct
Prescaler Division Ratio	No division
Input Filter	**15**

Tabela 6: Konfiguracja peryferium TIM3 oraz TIM4

2.7 USART6

To perfyferium zostało wykorzystane do komunikacji przez bluetooth.

Basic Parameters:	
Baud Rate	115200
Word Length	8 Bits (including Parity)
Parity	None
Stop Bits	1
Advanced Parameters:	
Data Direction	Receive and Transmit
Over Sampling	16 Samples

Tabela 7: Konfiguracja peryferium USART6

3 Urządzenia zewnętrzne

3.1 LSM9D1

Czujnik LSM9D1 jest połączeniem 3-osiowego cyfrowego żyroskopu, akcelerometru i magnetometru. Pozwala na pomiar przyspieszeń, pola magnetycznego oraz prędkości kątowej w konfigurowalnych zakresach. Będzie wykorzystywany do wyznaczenia lokalizacji robota w labiryncie. Do filtracji pomiarów wykorzystywany jest filtr Kalmana. Komunikuje się poprzez magistralę I2C bądź SPI. W projekcie została wybrana komunikacja wykorzystująca interfejs SPI.

3.2 VL53L1X

VL53L1x jest to cyfrowy czujnik łączący pomiar zbliżeniowy oraz zdolność mierzenia światła otoczenia. Pomiar odległości od ścianki labiryntu odbywa się przy pomocy metody czasu przelotu światła (time of flight). Polega ona na wykorzystaniu diody nadawczej do generowania bardzo krótkiego impulsu światła czerwonego lub podczerwonego o wąskim spektrum, które odbija się od obiektu docelowego i z powrotem do czułego, laserowego detektora energii, zwanego również diodą odbiorczą. Dokładna i precyzyjna elektronika w czujniku może mierzyć czas przejścia światła i wykorzystywać stałą dla prędkości światła do obliczania odległości obiektu od czujnika. Czujnik VL53L1X umożliwia dzięki tej technice

dokonywanie dokładnych pomiarów niezależnie od koloru czy odbicia światła od obiektu. Działa w zakresie do 400 cm z rozdzielczością 1 mm. Komunikuje się z mikroprocesorem poprzez magistralę I2C. Są one wolniejsze od czujników odległości (diody IR i fototranzystora), ale jako wynik zwracają dokładny pomiar odległości w milimetrach, charakteryzują się także większym zasięgiem.

3.3 Enkodery optyczne

Określanie prędkości kół robota umożliwiają enkodery optyczne firmy Pololu przystosowane do wybranych silników. Rozdzielczość enkoderow wynika z zastosowanych łopatek. Wybrano nasadki z 5 łopatkami, które dają 20 impulsów na obrót silnika, co w połaczeniu z przekładnią 30:1 daje rozdzielczość 600 impulsów na obrót.

3.4 Silniki

Robot będzie posiadał dwa silniki wysokiej mocy Pololu HPCB z przekładnia 30:1. Posiadają one wytrzymałe, karbonowe szczotki. Prędkość obrotowa to 1000 obr/min, a moment obrotowy wynosi 0.6 kg*cm (0,059 Nm). Silniki będą przykrecone do płytki za pomocą dedykowanych do nich mocowań krótkich firmy Pololu.

3.5 Mostek H

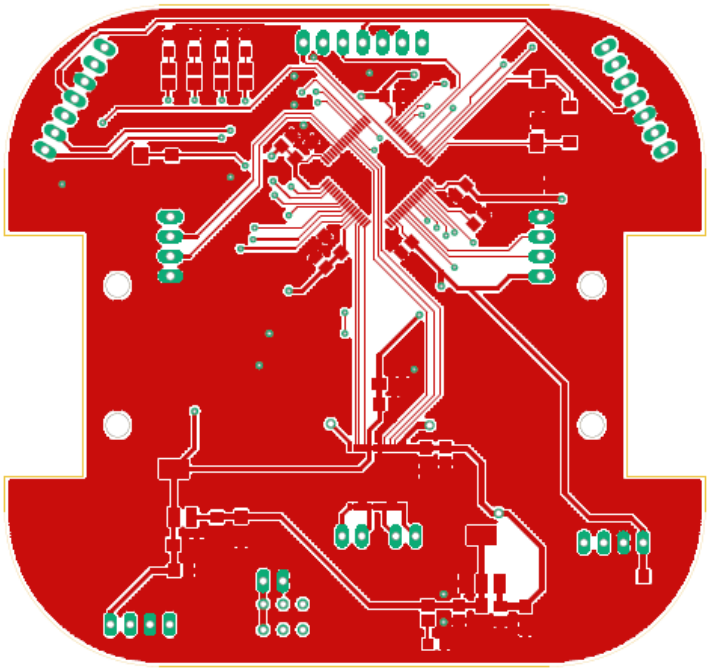
Do sterowania silnikami wykorzystano mostek H TB6612 firmy Toshiba. Pozwala on na sterowanie dwoma silnikami osobno. Jest on w zupełności wystarczający do sterowania tego typu robotem zakładając, że nie będzie on wywierał zbyt dużych przeciążeń na silnikach.

4 Projekt elektroniki

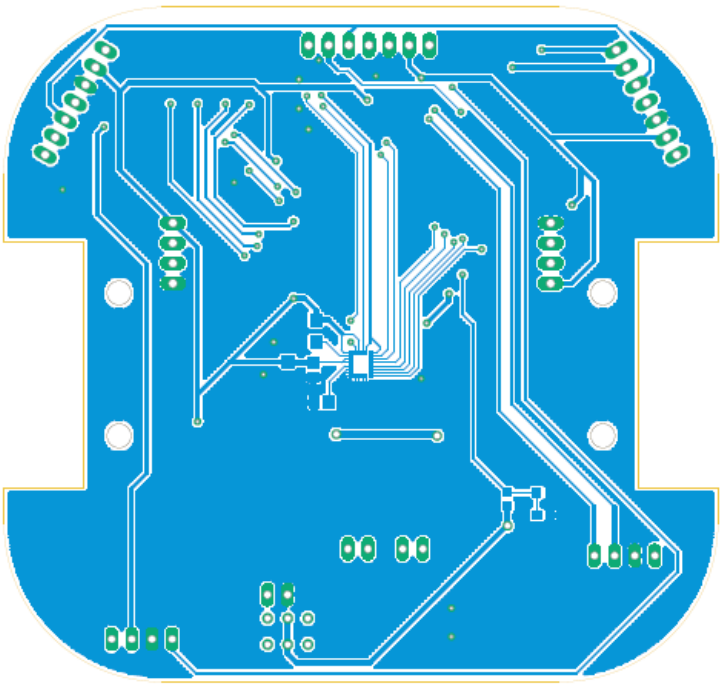
W programie Eagle wykonano schemat płytki, który został dołączony jako dodatek do dokumentu. W schemacie zostały umieszczone następujace elementy:

- mikroprocesor STM32F411RET6
- LSM9D1 moduł IMU
- podwójny mostek H TB6612
- stabilizatory napięcia LM1117S 5V i 3V3
- enkodery optyczne
- dalmierze laserowe VL53L1X
- mocowania do silników Pololu długie
- diody LED
- rezystory, kondensatory
- przełącznik suwakowy, tact switch

Widok wygenerowanej płytki z zachowaniem skali został umieszczony poniżej.

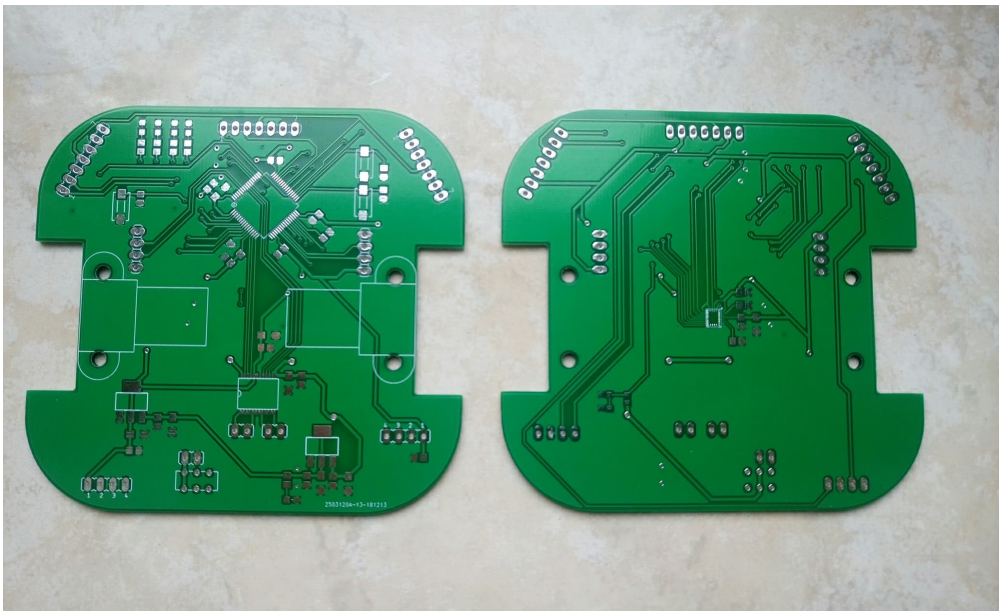


Rysunek 3: Widok górnej warstwy płtyki PCB



Rysunek 4: Widok dolnej warstwy płytki PCB

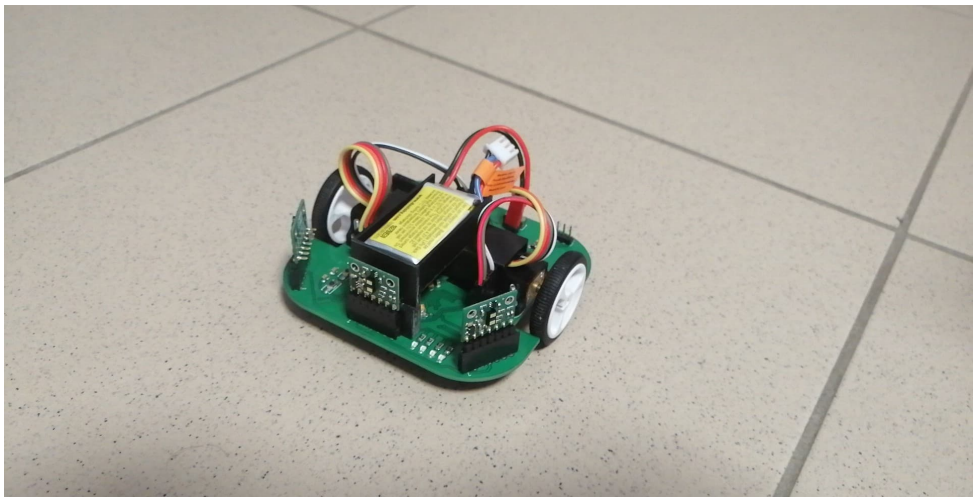
Płytka zostala wytworzona przez firmę JLCPCB. Zdjęcie gotowej płytki widać poniżej.



Rysunek 5: Widok gotowej płytki

5 Konstrukcja mechaniczna

Gotowa konstrukcja robota składa się z płytki PCB, przymocowanych do niej za pomocą mocowań silników oraz ślizgacza podporowego plastkiowej kulki umieszonego w przedniej części robota.Wykorzystano koła pololu 32x7 mm. Na płytce znajduje się nakładka wykonana z polistyrenu, która odgradza od siebie enkodery oraz stanowi półkę dla akumulatora. Widok zmontowanego robota widoczny jest poniżej.



Rysunek 6: Widok robota

6 Oprogramowanie

6.1 Filtr Kalmana oraz obsługa IMU

Na podstawie [3] opracowano projekt realizacji filtru Kalmana. Jego zadaniem jest dynamiczna filtracja mierzonych wartości. Filtr Kalmana jest wykorzystany przy pomiarach wartości kąta, o który będzie się odchylać robot. Do pomiarów wykorzystane są akcelerometr oraz żyroskop, a miarę odchylenia kątowego można przedstawić równieniem:

$$\theta_k = \theta_{k-1} + (\omega_{k-1} - g_{bias})dt \tag{1}$$

θ - odchylenie kątowe; ω - prędkość kątowa zmierzona żyroskopem; g_{bias} - dryft żyroskopu.

Ogólne równania opisujące problem prezentują się następująco:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$$

$$z_k = Hx_k + v_k$$

A - macierz stanu; B - macierz sterowania; w_k - szum procesu; v_k - zakłócenia pomiaru; z_k - pomiar akcelerometru; x_k - wektor stanu.

Tym samym, uwzględniając równanie 1, można zapisać:

$$\begin{bmatrix} \theta_k \\ g_{bias} \end{bmatrix} = \begin{bmatrix} 1 & -dt \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_{k-1} \\ g_{bias} \end{bmatrix} + \begin{bmatrix} dt \\ 0 \end{bmatrix} \omega_{k-1}$$

$$z_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \theta_k \\ g_{bias} \end{bmatrix} + v_k$$

Wykorzystując te zależności,stworzono projekt fitru Kalmana. W ogólności filtr kalmana składa się z 2 etapów, przy założeniu warunków początkowych: $\hat{x}_{k-1}$ oraz P_{k-1}, gdzie P to macierz kowariancji.

1. **Aktualizacja czasu ("Przewidywanie")**

$$\hat{x_k}^- = A\hat{x}_{k-1} + Bu_{k-1}$$

$$P_k^- = AP_{k-1}A^T + Q$$

2. **Aktualiacja pomiarów ("Korekcja")**

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$$

$$\hat{x_k} = \hat{x}_{k-1} + K_k(z_k - H\hat{x_k}^-$$

$$P_k = (I - K_k H)P_k^-$$

gdzie Q-wariancja szumu procesowego; R-wariancja zakłóceń.

W main rozpoczyna się inicjalizacja Kalmana. Funkcja przypisuje odpowiednie wartości parametrów filtru. Zostały one dobrane na podstawie eksperymentów prowadzonych w ramach innego projektu. Najlepsze rezultaty otrzymano dla $R = 1$ oraz $q = 0.05$.

Komunikacja z IMU odbywa się przez interfejs SPI. Ustawiamy oba piny odpowiadające za chip select - jedne od akcelerometru i żyroskopu, drugi od magnetometru są ustawiane na początku na 1.

W celu uruchomienia żyroskopu rejestr CTRL_REG1_G ustawiamy na wartość 11111000 co oznacza częstotliwość 476Hz oraz zakres pomiarowy w stopniach na sekundę 2000dps. Transmisja SPI odbywa się po ustwieniu chip select na 0, poprzez funckję HAL_SPI_TransmitReceive. Przyjmuje ona 5 parametrów.Pierwszym jest wskaźnik do struktury opisującej wykorzystywany interfejs SPI, następnie bufor nadawczy, później odbiorczy, wielkość bufora oraz maksymalny czas komunikacji. Po jej zakończeniu pin chip select zmienia stan 1.

W celu odczytania danych z akcelerometru rejestr CTRL_REG6_XL ustawiamy na 00011000 co oznacza zakres pracy akceleromtru 8g. Transmisja odbywa się tak jak wyżej.

Tak uzyskane dane są następnie filtrowane. Uzyskany wyniki zapisywany jest w zmiennej stan.theta.

Dane z akcelerometru służą do obliczenia kąta obrotu wokół osi X, zgodnie ze wzorem:

$$\alpha = arctg\Big(\frac{A_x}{\sqrt{A_y^2 + A_z^2}}\Big) \cdot \frac{180}{\pi}$$

oraz wokół osi Y zgodnie ze wzorem:

$$\beta = arctg\Big(\frac{A_y}{\sqrt{A_x^2 + A_z^2}}\Big) \cdot \frac{180}{\pi}$$

Obliczony tak kąt obrotu oraz prędkość kątową wokół odpowiedniej osi (pomiar żyroskopu) wykorzystano do przefiltrowania uzyskanych pomiarów. Porównanie wyniku z akceleromeru, żyroskopu i po filtracji odybwało się przy użyciu STMStudio. Filtr działa poprawnie, redukuje powstałe piki.

6.2 Dalmierze laserowe

Komunikacja z dalmierzami VL53L1X odbywa się poprzez interfejs I2C. W programie wykorzystano bibliotekę dostarczoną przez producenta. Dodano funkcje, które powodują, że kod staje się bardziej przejrzysty. Aby móc komunikować się z trzema czujnikami korzystając z jednej magistrali I2C należy zaraz po połaczeniu z czujnikiem zmienić jego adres. odbywa się to w funckji private_VL_init umieszconej poniżej.

```
1  void private_VL_init(VL53L1_DEV vl_dev, GPIO_TypeDef *xsPort, uint16_t xsPin, int
       newAddress){
2    vl_dev->I2cHandle = &hi2c3;
3    vl_dev->I2cDevAddr = 0x52;
4    HAL_GPIO_WritePin(xsPort,xsPin,GPIO_PIN_SET);
5    HAL_Delay(2);
6    VL53L1_SetDeviceAddress(vl_dev, newAddress);
7    vl_dev->I2cDevAddr = newAddress;
8    VL53L1_WaitDeviceBooted( vl_dev );
9    VL53L1_DataInit( vl_dev );
10   VL53L1_StaticInit( vl_dev );
11   VL53L1_SetDistanceMode( vl_dev, VL53L1_DISTANCEMODE_LONG );
12   VL53L1_SetMeasurementTimingBudgetMicroSeconds( vl_dev, 50000 );
13   VL53L1_SetInterMeasurementPeriodMilliSeconds( vl_dev, 500 );
14   VL53L1_StartMeasurement( vl_dev );
15 }
16 }
```

Korzystając z funkcji VL53L1_GetRangingMeasurmentData otrzymujemy odleglość robota od przeszkody podana w milimetrach. Posiadając 3 czujniki jeden z przodu po srodku i dwa pod kątem $-45°$ oraz $45°$. Pozwala to w porawny sposób określić gdzie znajduje się przeszkoda i skutecznie ją ominać.

6.3 Sterowanie silnikami

Sterowanie silnikami odbywa się poprzez podwójny mostek H. Tabela prawdy mostka znajduje się w tabeli poniżej.

Input				Output		
IN1	IN2	PWM	STBY	OUT1	OUT2	Mode
H	H	H/L	H	L	L	Short brake
L	H	H	H	L	H	CCW
		L	H	L	L	Short brake
H	L	H	H	H	L	CW
		L	H	L	L	Short brake
L	L	H	H	OFF (High impedance)		Stop
H/L	H/L	H/L	L	OFF (High impedance)		Standby

Rysunek 7: Tabela prawdy mostka H

Ustawiamy piny wejsciowe do sterownika jeden na set a drugi na reset. W ten sposób ustawiamy kierunek obrotów silnika. Następnie startujemy PWM. Początkowo wypełnienie ustawione jest na 20%. Działające enkodery pozwalają na korekcje ruchu i w ten sposob wyeliminowanie błędu poślizgu kół. W momencie wykrycia przeszkody robot zmienia kierunek ruchu. Prosty profiler prędkości został pokazany poniżej.

```
1 VL_getDistance(&vl_distance);
2     if(HAL_GetTick()>10000){
3       Left_Motor(0);
```

```
4           Right_Motor(0);}
5       else{
6         if(vl_distance[1]<500){
7           Left_Motor(200);
8           Right_Motor(-200);
9         }else{
10        Left_Motor(200);
11        Right_Motor(200);
12      }
13      }
14  }
```

7 Podsumowanie

Udalo się zrealizować wszystkie postawione cele. Robot porusza się, zmienia kierunek po wykryciu przeszkody.

Literatura

[1] H. Grzegorczyk. Algorytm lokalizacji robota mobilnego w labiryncie.

[2] A. Kurczyk. *Mikrokontrolery STM32 dla początkujących*. Wydawnictwo btc, Legionowo, 2019.

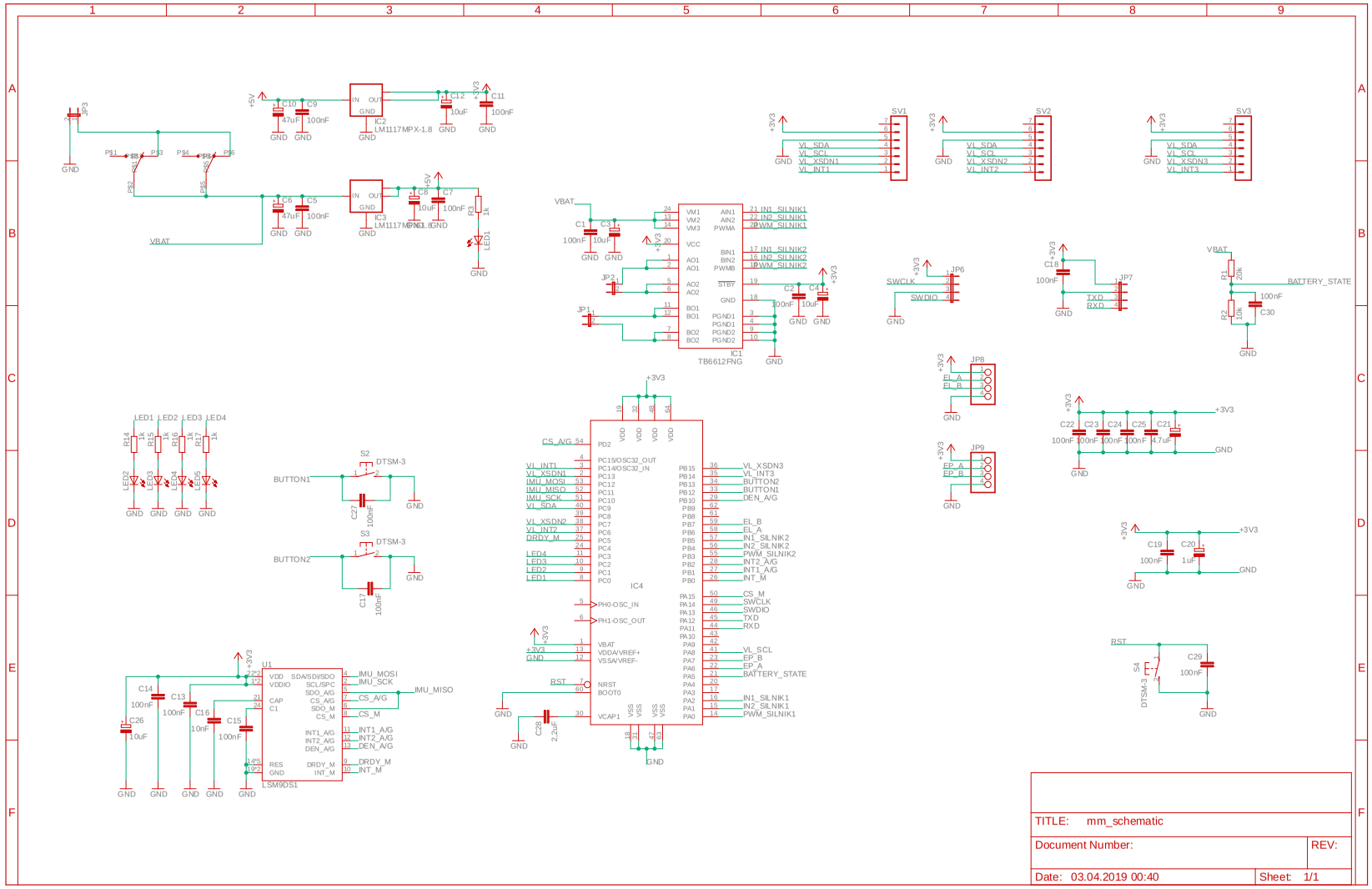
[3] J. Kędzierski. Filtr kalmana - zastosowania w prostych układach sensorycznych. Paz. 2009.

[4] STMicroElectronics. *LSM9DS1 datasheet*.

[5] STMicroElectronics. *STM32F411RET6 datasheet*.

[6] STMicroElectronics. *VL53L1X datasheet*.

[7] Toshiba. *TB6612FNG datasheet*.


1 2 3 4 5 6 7 8 9
A B C D E F
JP3
PB1 PB3 PB4 PB6 PB2 PB5
GND
VBAT
+5V
C10 47uF
C9 100nF
IC2 LM1117MPX-1.8
C12 10uF
C11 100nF
+3V3
C6 47uF
C5 100nF
IC3 LM1117MPX-1.8
C8 10uF
C7 100nF
R3 1K
LED1
SV1
SV2
SV3
VL_SDA
VL_SCL
VL_XSDN1
VL_INT1
VL_XSDN2
VL_INT2
VL_XSDN3
VL_INT3
IC1 TB6612FNG
C1 100nF
C3 10uF
JP2
JP1
C2 100nF
C4 10uF
JP6
SWCLK
SWDIO
C18 100nF
JP7
TXD
RXD
R1 20k
R2 10k
C30 100nF
BATTERY_STATE
JP8
EL_A
EL_B
JP9
EP_A
EP_B
C22 100nF
C23 100nF
C24 100nF
C25 100nF
C21 4.7uF
C19 100nF
C20 1uF
LED1 LED2 LED3 LED4
R14 1k
R15 1k
R16 1k
R17 1k
LED2 LED3 LED4 LED5
S2 DTSM-3
BUTTON1
C27 100nF
S3 DTSM-3
BUTTON2
C17 100nF
IC4
PD2
PC15/OSC32_OUT
PC14/OSC32_IN
PC13
PC12
PC11
PC10
PC9
PC8
PC7
PC6
PC5
PC4
PC3
PC2
PC1
PC0
PB15
PB14
PB13
PB12
PB11
PB10
PB9
PB8
PB7
PB6
PB5
PB4
PB3
PB2
PB1
PB0
PA15
PA14
PA13
PA12
PA11
PA10
PA9
PA8
PA7
PA6
PA5
PA4
PA3
PA2
PA1
PA0
PH0-OSC_IN
PH1-OSC_OUT
VBAT
VDDA/VREF+
VSSA/VREF-
NRST
BOOT0
VCAP1
C28 2.2uF
CS_A/G
VL_INT1
VL_XSDN1
IMU_MOSI
IMU_MISO
IMU_SCK
VL_SDA
VL_XSDN2
VL_INT2
DRDY_M
LED4
LED3
LED2
LED1
VL_XSDN3
VL_INT3
BUTTON1
BUTTON2
DEN_A/G
INT1_A/G
INT2_A/G
INT_M
CS_M
SWCLK
SWDIO
TXD
RXD
VL_SCL
EP_B
EP_A
BATTERY_STATE
IN1_SILNIK1
IN2_SILNIK1
PWM_SILNIK1
IN1_SILNIK2
IN2_SILNIK2
PWM_SILNIK2
U1 LSM9DS1
C14 100nF
C13 100nF
C16 10nF
C15 100nF
C26 10uF
IMU_MISO
RST
S4 DTSM-3
C29 100nF
TITLE: mm_schematic
Document Number:
REV:
Date: 03.04.2019 00:40
Sheet 1/1