

PROJEKT

STEROWNIKI ROBOTÓW

Dokumentacja techniczna

Interfejs zdalnego sterowania robotem
klasy minisumo

InZdaSter (2,0)

Skład grupy (VI):

Dawid MARSZAŁKIEWICZ, 218665

Piotr PORTASIAK, 218594

Termin: srTP11

Prowadzący:

mgr inż. Wojciech DOMSKI

20 czerwca 2017

Spis treści

1	Opis funkcjonalności urządzenia	2
2	Konfiguracja mikrokontrolerów	2
2.1	Mikrokontroler STM32F7	2
2.1.1	Ustawienia zegara	4
2.1.2	Ustawienia DMA	4
2.1.3	Konfiguracja dodatkowych peryferii	4
2.1.4	Peryferia dotyczące GUI	5
2.1.5	Peryferia dotyczące modułu rozszerzającego (shield)	7
2.2	Mikrokontroler STM32F3	8
2.2.1	Podstawowe funkcjonalności	8
2.2.2	Konfiguracja peryferii	9
2.2.3	Konfiguracja RCC i NVIC	9
3	Wykorzystywane układy zewnętrzne	11
4	Opis działania programu	12
4.1	Projekt interfejsu graficznego	13
4.2	Pętla obsługi zdarzeń	14
5	Projekt modułu rozszerzającego	15
5.1	Opis funkcjonalności	15
5.2	Wykaz elementów elektronicznych	16
	Bibliografia	16

1 Opis funkcjonalności urządzenia

Projekt zakładał budowę interfejsu zdalnej komunikacji, diagnostyki oraz sterowania robotem minisumo w postaci bezprzewodowego pilota z wyświetlaczem LCD. W ramach jego realizacji uzyskano wszystkie zakładane funkcjonalności urządzenia:

- Sterowanie robotem klasy (2,0) poprzez moduł Bluetooth.
- Sterowanie (jazda do przodu, do tyłu oraz obroty) odbywa się poprzez przyciski na panelu dotykowym lub joysticku znajdujący się na dodatkowej płytce.
- Gdy wyświetlacz znajduje się do góry, joystick jest nieaktywny.
- W przeciwnym wypadku to wyświetlacz nie reaguje na dotyk, co jest zabezpieczeniem przed niekontrolowanym użyciem.
- Komenda głosowa 'go!' otwiera flagi, natomiast komenda 'stop!' je zamyka.
- Do poprawnego działania komend głosowych wymagana jest obecność karty SD wraz z wzorcowymi transformatami

2 Konfiguracja mikrokontrolerów

Do realizacji projektu powstały dwie osobne konfiguracje mikrokontrolerów:

- STM32F746G-DISCOVERY – zdalne sterowanie robotem (nadajnik)
- STM32F3-DISCOVERY – symulacja pracy robota (odbiornik).

2.1 Mikrokontroler STM32F7

Na rysunku 1 znajduje się konfiguracja pinów mikrokontrolera STM32F746, który znajduje się w obudowie TFBGA216. Zostały już skonfigurowane peryferia, które nie są jeszcze wykorzystywane w celu wykrycia ewentualnych konfliktów.

W tabeli 3 znajduje się spis wykorzystywanych pinów wraz z informacją jaką pełnią funkcję oraz do jakiego peryferium należy.

Tabela 1: Tabela z konfiguracją pinów mikrokontrolera STM32F746

Peryferium	Sygnal	Pin
ADC3	ADC3_IN6	PF8
	FMC_SDCKE0	PC3
	FMC_SDNE0	PH3
	FMC_BA0	PG4
	FMC_BA1	PG5
	FMC_A0	PF0
	FMC_A1	PF1
	FMC_A2	PF2
	FMC_A3	PF3
	FMC_A4	PF4

	FMC_A5	PF5
	FMC_A6	PF12
	FMC_A7	PF13
	FMC_A8	PF14
	FMC_A9	PF15
	FMC_A10	PG0
	FMC_SDCLK	PG8
	FMC_SDNCAS	PG15
	FMC_SDNRAS	PF11
	FMC_SDNWE	PH5
	FMC_A11	PG1
	FMC_D0	PD14
	FMC_D1	PD15
	FMC_D2	PD0
	FMC_D3	PD1
	FMC_D4	PE7
	FMC_D5	PE8
	FMC_D6	PE9
	FMC_D7	PE10
	FMC_D8	PE11
	FMC_D9	PE12
	FMC_D10	PE13
	FMC_D11	PE14
	FMC_D12	PE15
	FMC_D13	PD8
	FMC_D14	PD9
	FMC_D15	PD10
I2C3	I2C3_SCL	PH7
	I2C3_SDA	PH8
LTDC	LTDC_R0	PI15
	LTDC_R1	PJ0
	LTDC_R2	PJ1
	LTDC_R3	PJ2
	LTDC_R4	PJ3
	LTDC_R5	PJ4
	LTDC_R6	PJ5
	LTDC_R7	PJ6
	LTDC_G0	PJ7
	LTDC_G1	PJ8
	LTDC_G2	PJ9
	LTDC_G3	PJ10
	LTDC_G4	PJ11
	LTDC_G5	PK0
	LTDC_G6	PK1
	LTDC_G7	PK2
	LTDC_B0	PE4
	LTDC_B1	PJ13
	LTDC_B2	PJ14

	LTDC_B3 LTDC_B4 LTDC_B5 LTDC_B6 LTDC_B7 LTDC_HSYNC LTDC_VSYNC LTDC_CLK LTDC_DE	PJ15 PG12 PK4 PK5 PK6 PI10 PI9 PI14 PK7
RCC	RCC_OSC_IN RCC_OSC_OUT	PH0/OSC_IN PH1/OSC_OUT
SAI2	SAI2_SD_A SAI2_SCK_A SAI2_FS_A SAI2_MCLK_A SAI2_SD_B	PI6 PI5 PI7 PI4 PG10
SDMMC1	SDMMC1_CK SDMMC1_CMD SDMMC1_D0	PC12 PD2 PC8
SPI2	SPI2_MISO SPI2_MOSI SPI2_SCK	PB14 PB15 PI1
USART6	USART6_RX USART6_TX	PC7 PC6

2.1.1 Ustawienia zegara

Na rysunku 2 znajduje się konfiguracja zegara mikrokontrolera znajdującego się na płytce rozwojowej *32F746GDISCOVERY*, który korzysta z zewnętrznego rezonatora kwarcowego.

2.1.2 Ustawienia DMA

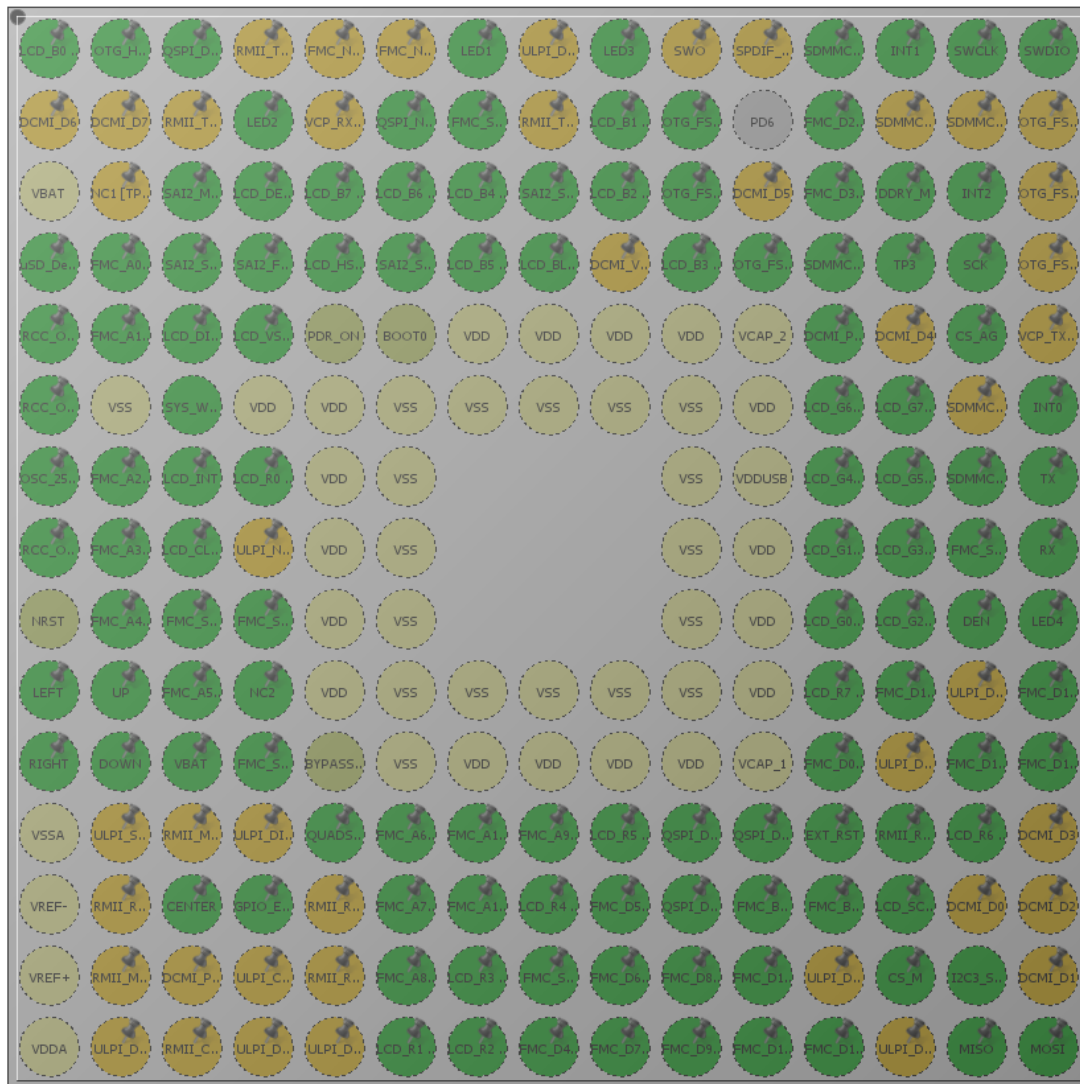
W tabeli 2 znajduje się wykaz konfiguracji kontrolera DMA

Tabela 2: Konfiguracja DMA

Żądanie DMA	Strumień	Kierunek
MEMTOMEM	DMA2_Stream0	Memory To Memory
I2C3_RX	DMA1_Stream1	Peripheral To Memory
I2C3_TX	DMA1_Stream4	Memory To Peripheral

2.1.3 Konfiguracja dodatkowych peryferii

Poniżej znajduje się opis skonfigurowanych peryferii na płytce *32F746GDISCOVERY*. Na obecnym etapie realizacji projektu można wyróżnić dwie główne grupy peryferiali:



STM32F746NGHx
TFBGA216

Rysunek 1: Konfiguracja pinów STM32F746

- związane z obsługą GUI,
- związane z dodatkowym modulem rozszerzającym (shield).

2.1.4 Peryferia dotyczące GUI

Do poprawnego działania bibliotek STemWin wymagana jest konfiguracja opisanych poniżej peryferiów.

FMC - Flexible Memory Controller Peryferium to udostępnia interfejs do pamięci SDRAM. Sposób sterowania pamięcią SDRAM został zdefiniowany przez połączenia znajdujące się na płytce deweloperskiej, natomiast konfiguracja czasowa SDRAM została stworzona na podstawie dokumentacji układu MT48LC4M32B2B5. Poniżej znajduje się spis najważniejszych konfiguracji.

- Bank: SDRAM Bank 1

- Liczba bitów adresu wiersza: 12 bitów

LTDC Sterownik wyświetlaczy LCD-TFT (LTDC) został skonfigurowany w 24-bitowym formacie danych z ustawieniami synchronizacji zgodnymi z dokumentacją wyświetlacza.

I2C3 Wyświetlacz LCD znajdujący się na płycie deweloperskiej posiada kontroler panelu dotykowy, z którym można się komunikować poprzez I2C. Peryferium I2C3 zostało ustawione w standardowym trybie prędkości z głównym adresem slave wynoszącym 112. Zdeklarowano również dwa kanały DMA odpowiedzialne za nadawanie (*Peripheral to Memory*) oraz odbieranie (*Memory to Peripheral*).

SAI Serial Audio Interface to interfejs wspierający szeroką gamę protokołów audio. Mikrokontroler STM32F7 komunikuje się dzięki temu peryferium z kodekiem audio WM8994, który z kolei komunikuje się z dwoma mikrofonami, które dostarczają informację o natężeniu dźwięku w formacie PCM.

- Tryb audio: Master Receive
- Długość ramki: 64 bity
- Rozmiar danych: 16 bitów

SDMMC1 Peryferium, które dostarcza interfejs do komunikacji z kartą SD. Zapis oraz odczyt z karty SD jest realizowany przy pomocy systemu plików FatFS, w którym ustawiono stały stempel czasowy.

CRC oraz DMA2D Dodatkowo zostały aktywowane peryferia DMA2D (układ graficzny pozwalający na szybkie kopiowanie strumienia danych oraz konwersje pomiędzy formatami pikseli) oraz CRC (ze względu na wymagania biblioteki STemWin).

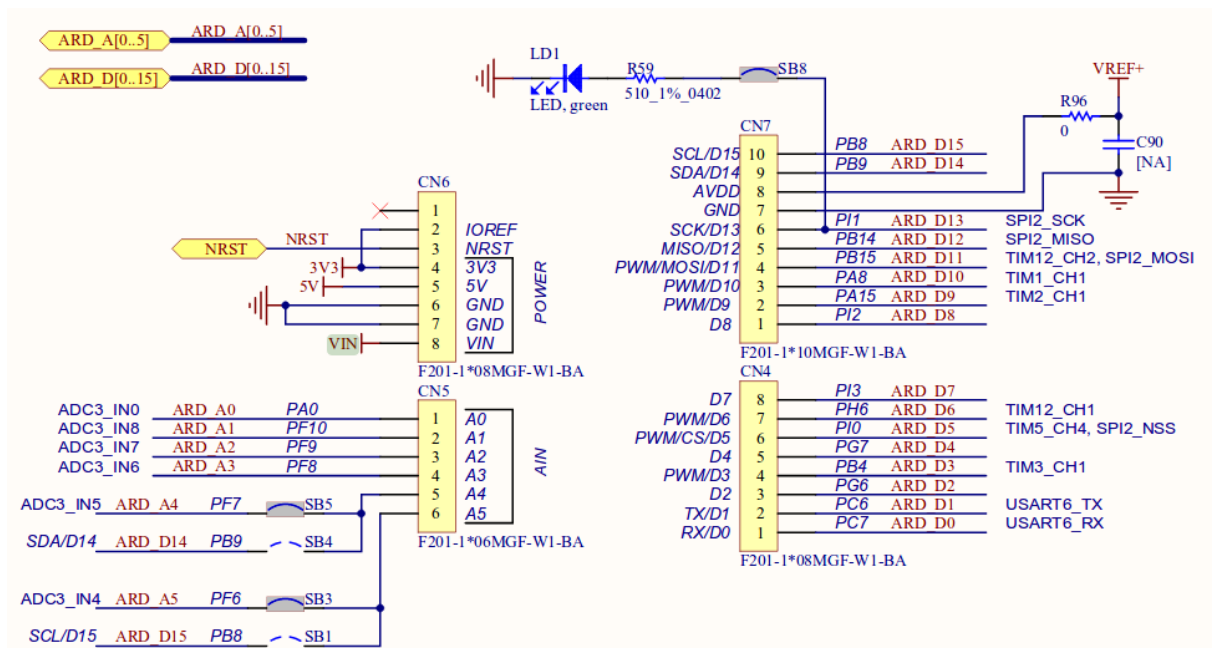
2.1.5 Peryferia dotyczące modułu rozszerzającego (shield)

Wykorzystywana w projekcie płytką deweloperską *STM32F746G-DISCOVERY* została wyposażona w złącza zgodne ze specyfikacją Arduino Uno (rys. 3). Pozwala to na zainstalowanie własnego modułu rozszerzeń (shield) przystosowanego do napięcia logiki mikroprocesora *STM32* (3.3V). Ponadto złącze V_{in} pozwala na dołączenie zewnętrznego źródła napęcia (7-12V) np. akumulatora Li-poly (7.4V), co zostało ujęte w projekcie układu elektronicznego shieldu.

UART W celu bezprzewodowej wymiany informacji pomiędzy pilotem a robotem został skonfigurowany interfejs portu szeregowego UART do obsługi modułu Bluetooth *HC-05*. Komunikacja pomiędzy urządzeniami odbywa się w trybie transparentnym, z ustalonymi parametrami połączenia .

SPI Do obsługi inercyjnej jednostki pomiarowej *LSM9DS1* przygotowano kanał komunikacyjny – SPI.

ADC Wykorzystywany do pomiaru napięcia akumulatora Li-poly.



Rysunek 3: Układ wyprowadzeń *Arduino Shield*

GPIO Pozostałe wyprowadzenia shieldu wykorzystano do obsługi:

- 5-pozycyjnego joysticka – jako linie wejściowe *GPIO_Input* z obsługą zgłaszania zewnętrznego przerwania
- diod sygnalizacyjnych kierunek (LED) – jako linie wyjściowe *GPIO_Output*

2.2 Mikrokontroler STM32F3

Z uwagi na trwające prace nad budową robota mobilnego minisumo, (który ma być docelowym obiektem sterowanym za pomocą opracowywanego pilota) postanowiono przygotować układ symulujący jego działanie. Do tego celu wykorzystano moduł rozwojowy *STM32F3-DISCOVERY*.

2.2.1 Podstawowe funkcjonalności

Podstawowa konfiguracja (rys. 4) pozwala na nawiązanie bezprzewodowego połączenia Bluetooth oraz interpretację odbieranych komend typu:

- $\langle F \rangle$ - jedź do przodu
- $\langle B \rangle$ - jedź do tyłu
- $\langle L \rangle$ - skreć w lewo
- $\langle R \rangle$ - skreć w prawo
- $\langle S \rangle$ - stop
- $\langle V:X \rangle$ - ustaw X% maksymalnej prędkości V

Dodatkowo zostają wyświetlone komunikaty w postaci świecących diod (LED), informujące o kierunkach poruszania się, hamowania, prędkości robota.

2.2.2 Konfiguracja peryferii

Wykorzystane peryferia mikrokontrolera STM32F3 to:

- USART - obsługa modułu Bluetooth HC-05
 - Baudrate: 115200
 - Liczba bitów: 8
 - Kontrola parzystości: brak
 - Bity stopu: 1
- SWD - wykorzystanie SWV oraz przekierowania funkcji *printf()*, w celu śledzenia komunikatów odbieranych przez moduł Bluetooth oraz innego rodzaju debugowania.
- GPIO - do sterowania LEDami sygnalizacyjnymi

Tabela 3: Tabela z konfiguracją pinów mikrokontrolera STM32F303

Peryferium	Sygnal	Pin
SYS	TRACESWO	PB3
USART4	UART4_RX UART4_TX	PC11 PC10
GPIO	GPIO_Output LD4 [Blue Led] GPIO_Output LD3 [Red Led] GPIO_Output LD7 [Green Led] GPIO_Output LD10 [Red Led] GPIO_Output LD6 [Green Led]	PE8 PE9 PE11 PE13 PE15

2.2.3 Konfiguracja RCC i NVIC

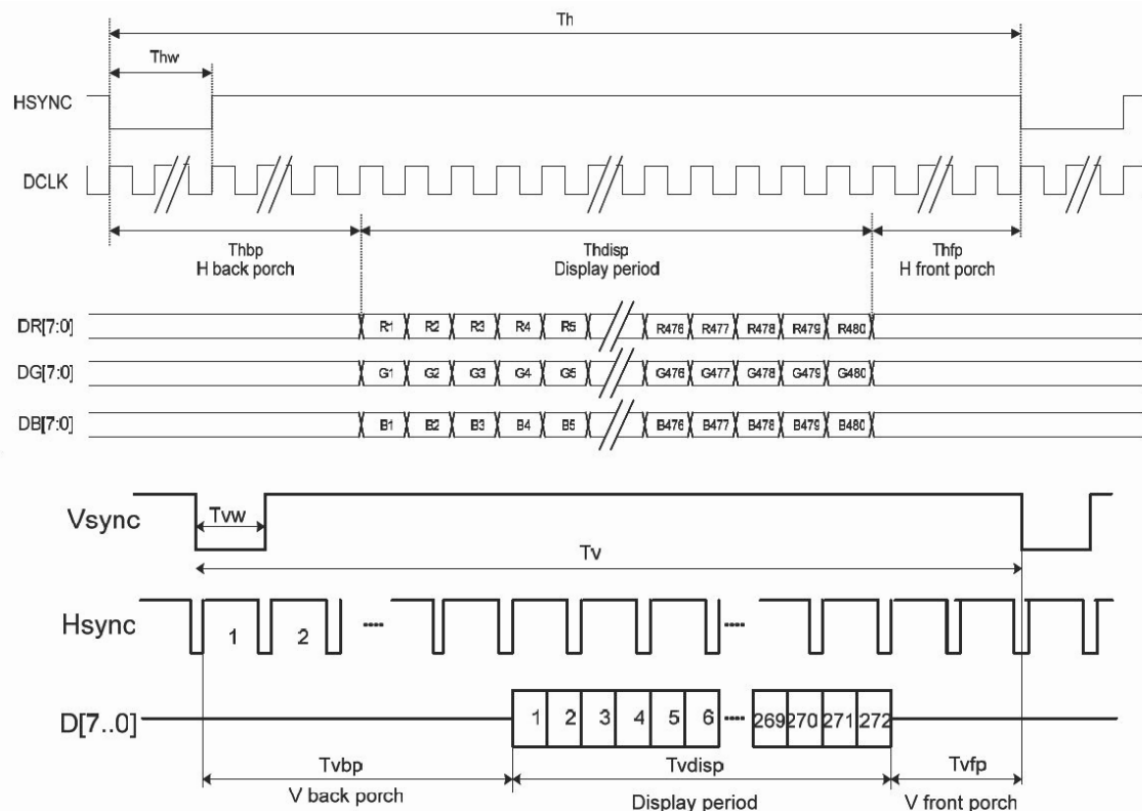
Rdzeń mikrokontrolera taktowany jest sygnałem zegarowym o częstotliwości 48 MHz (rys. 5), którego źródłem jest wewnętrzny oscylator pracujący w pętli PLL. Natomiast magistrala z którą współpracuje USART4 działa z częstotliwością 24 MHz.

Port szeregowy został obsłużony w trybie z przerwaniem. Ramki danych przychodzące z nadajnika są odpowiednio parsowane w każdym wywołaniu funkcji zwrotnej zgłaszanej przez linię USART_RX. Funkcja odpowiadająca za to zadanie została przedstawiona w poniższym listingu:

```
1 uint8_t Flaga;  
2 uint8_t Predkosc;  
3 uint8_t Bufor;  
4  
5 void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)  
6 {  
7  
8     if (!Flaga)  
9         switch (Bufor)  
10        {  
11            case 'f':  
12                HAL_GPIO_WritePin(GPIOE, LD3_Pin, GPIO_PIN_SET);  
13                break;  
14            case 'b':  
15                HAL_GPIO_WritePin(GPIOE, LD10_Pin, GPIO_PIN_SET);
```


Diagram czasowy przedstawiający komunikację z wyświetlaczem znajduje się na rysunku 6. W górnej części jest przedstawiony sposób wyświetlania jednej linii (480 pikseli), natomiast w dolnej części znajduje się opis wyświetlania 272 linii. Należy zwrócić uwagę na sygnały synchronizacyjne HSYNC oraz VSYNC, których stan należy ustawić na niski w trakcie przejścia do nowej linii (w przypadku HSYNC) lub rozpoczęciu wyświetlania nowego ekranu (w przypadku VSYNC).

- kontroler panelu dotykowego FT5336 - komunikacja poprzez I2C.
- Czujnik inercyjny 9DoF LSM9DS1 - komunikacja poprzez SPI. W opisywany projekcie został wykorzystany tylko akcelerometr w roli inklinometru.



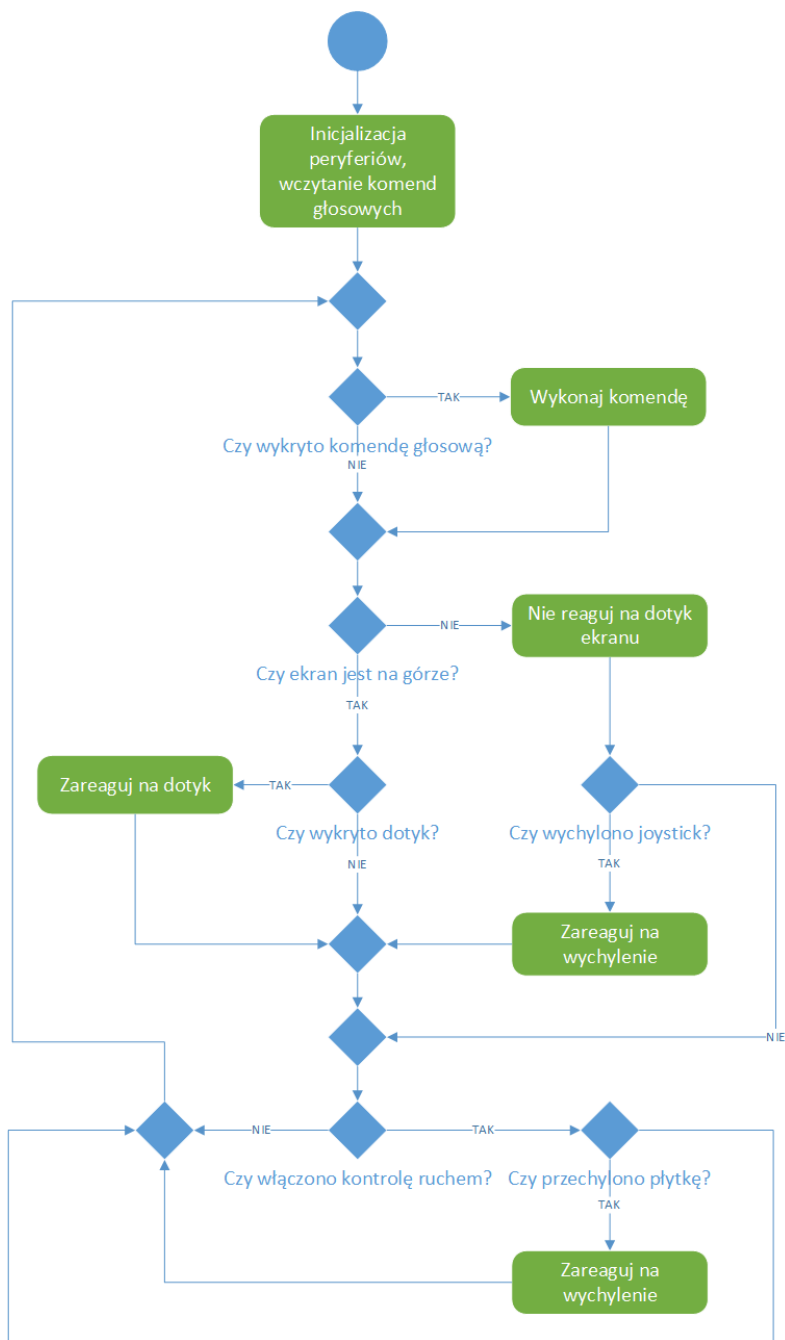
Rysunek 6: Diagram czasowy LTDC

4 Opis działania programu

Po inicjalizacji wszystkich peryferiów sterowanie zostaje w całości oddane do pętli obsługi zdarzeń obsługiwanej przez bibliotekę STemWin. Zapewnienie ciągłości pracy GUI wymaga inkrementacji zmiennej typu GUI_TIMER_OS co kwant czasu, w tym przypadku wykorzystano do tego przerwanie generowane przez SysTicka.

Kolejną rzeczą jaką należy dostarczyć do GUI jest stan panelu dotykowego. Wykonuje się to w przerwaniu od timera, które wywołuje się co 10ms.

Diagram działania programu jest widoczny na rysunku 7.



Rysunek 7: Diagram przepływu

4.1 Projekt interfejsu graficznego

Interfejs graficzny został zaprojektowany w dedykowanym do tego oprogramowaniu narzędziowym GUIBuilder. Projekt interfejsu znajduje się na rysunku 8.

Przyciski widoczne w lewym dolnym rogu ekranu służą do sterowania kierunkiem jazdy robota. Suwak znajdujący się po prawej stronie ustawia prędkość jazdy robota. Checkbox opisany jako "Sterowanie ruchem" będzie aktywować sterowanie za pomocą ruchu pilota.



Rysunek 8: GUI

4.2 Pętla obsługi zdarzeń

Po wystąpieniu zdarzenia do funkcji przekazanej przy inicjalizacji GUI wysyłana jest wiadomość o fakcie wystąpienia zdarzenia, jakiego elementu to zdarzenie dotyczy oraz jakiego typu zdarzenie się wywołało.

Fragmenc kodu odpowiedzialny za obsługę zdarzeń dotyczących suwaka wygląda następująco:

```

1   case ID_SLIDER_0: // Notifications sent by 'velocity'
2       switch(NCode) {
3           case WM_NOTIFICATION_CLICKED:
4               // USER START (Optionally insert code for reacting on notification message)
5               // USER END
6               break;
7           case WM_NOTIFICATION_RELEASED:
8               // USER START (Optionally insert code for reacting on notification message)
9               HAL_UART_Transmit(&huart6, velCommand, 2, 20);
10              // USER END
11              break;
12          case WM_NOTIFICATION_VALUE_CHANGED:
13              // USER START (Optionally insert code for reacting on notification message)
14              hItem = WM_GetDlgItem(pMsg->hWin, ID_SLIDER_0);
15              velocityValue = SLIDER_GetValue(hItem);
16              sprintf(velocityLabel, "Predkosc: %d", velocityValue);
17              velCommand[1] = velocityValue;
18              hItem = WM_GetDlgItem(pMsg->hWin, ID_TEXT_1);
19              TEXT_SetText(hItem, velocityLabel);
20              // USER END
21              break;

```

Linie 12-19 pokazują w jaki sposób jest dokonywana aktualizacja wyświetlanego tekstu podczas ruchu suwaka.

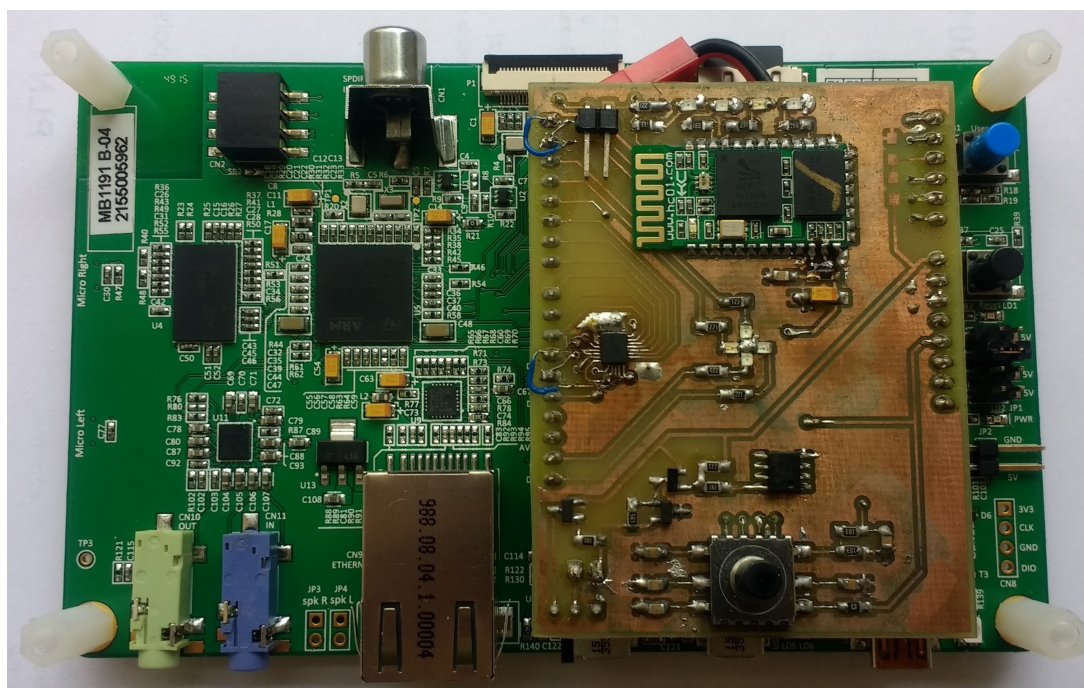
5 Projekt modułu rozszerzającego

Elektronika shieldu w całości została zaprojektowana w programie *KiCAD*, który pozwolił na przygotowanie projektu PCB (rysunek 10) na podstawie schematu ideowego.

5.1 Opis funkcjonalności

Moduł rozszerzający przedstawiony poniżej (rysunek 10) oraz na schemacie ideowym (na końcu dokumentu) składa się z następujących elementów:

- sekcji zasilania
 - złącze zasilania (akumulator Li-poly 7.4V)
 - zabezpieczenie przed odwrotną polaryzacją
 - zabezpieczenie przed spadkiem napięcia akumulatora
- układu pomiaru napięcia akumulatora Li-poly poprzez ADC
- układu komunikacji bezprzewodowej Bluetooth (moduł HC-05)
- inercyjnej jednostki pomiarowej (IMU) o 9 stopniach swobody w postaci pojedynczego układu scalonego LSM9DS1 produkcji firmy ST
- Joysticka 5-pozycyjnego (do przodu, do tyłu, prawo, lewo, stop)
- Świecące diody (LED) sygnalizujące kierunek poruszania się robota w przypadku sterowania z joysticka (do przodu, do tyłu, prawo, lewo)



Rysunek 9: Zdjęcie modułu rozszerzającego dla STM32F746G-DISCOVERY

5.2 Wykaz elementów elektronicznych

Poniżej (w tabeli 4) przedstawiono listę wykorzystywanych elementów do budowy modułu rozszerzającego (shieldu)

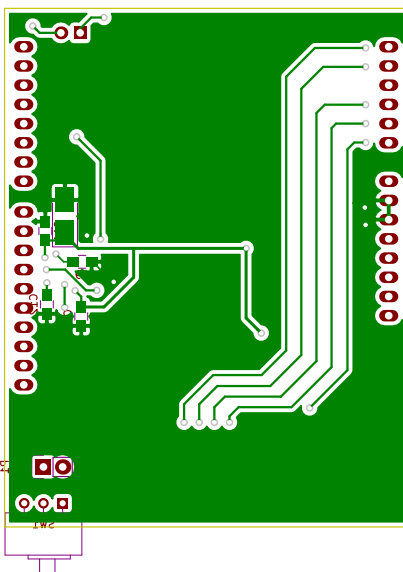
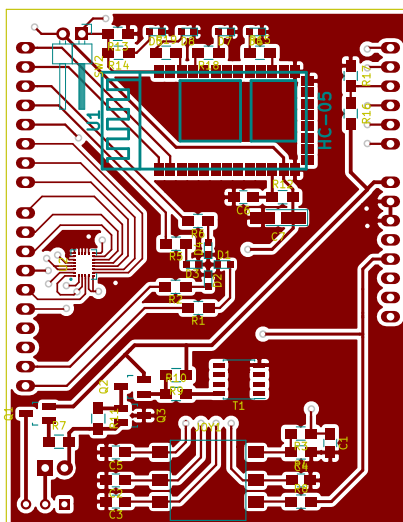
Tabela 4: Lista elementów elektronicznych (BOM)

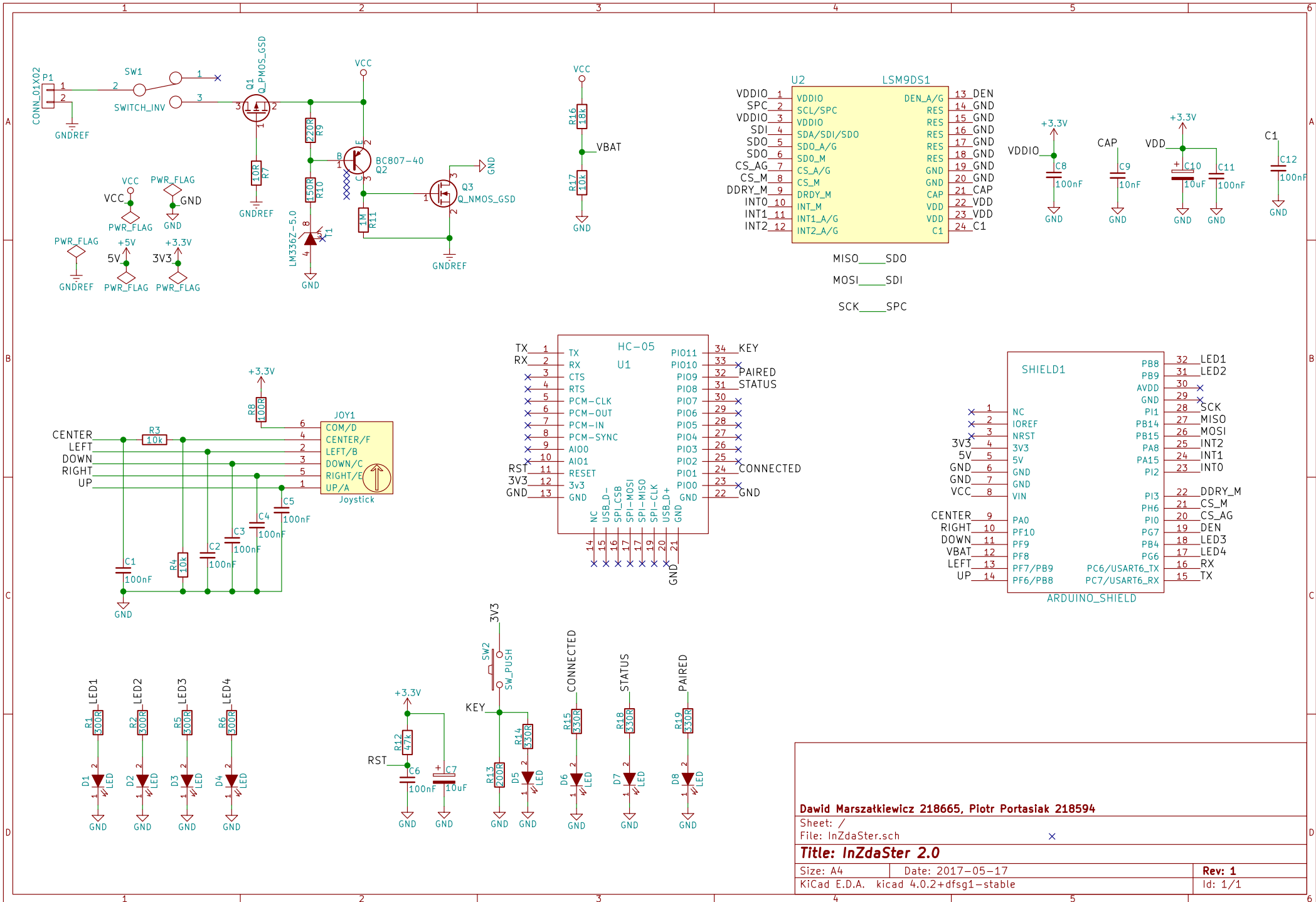
Symbol	Wartość	Footprint
P2	BLUETOOTH	Pin_Headers:Pin_Header_Straight_1x06
U1	LSM9DS1	warsztaty:LGA-24L_3x3.5mm_Pitch0.43mm
C2	10nF	Capacitors_SMD:C_0805_HandSoldering
C4	100nF	Capacitors_SMD:C_0805_HandSoldering
C1	100nF	Capacitors_SMD:C_0805_HandSoldering
C3	10uF	Capacitors_SMD:CP_Tantalum_Case-A_EIA-3216-18
SHIELD1	ARDUINO_SHIELD	arduino_shields:ARDUINO_SHIELD_2_040pins
SW1	SWITCH_INV	misc:przelacznik_smd_2
P1	CONN_01X02	Pin_Headers:Pin_Header_Straight_1x02
R2	220R	Resistors_SMD:R_0805_HandSoldering
R3	150R	Resistors_SMD:R_0805_HandSoldering
R4	1M	Resistors_SMD:R_0805_HandSoldering
Q1	Q_PMOS_GSD	TO_SOT_Packages_SMD:SOT-23_Handsoldering
R1	10R	Resistors_SMD:R_0805_HandSoldering
Q3	Q_NMOS_GSD	TO_SOT_Packages_SMD:SOT-23_Handsoldering
R6	10k	Resistors_SMD:R_0805_HandSoldering
R5	18k	Resistors_SMD:R_0805_HandSoldering
Q2	BC807-40	TO_SOT_Packages_SMD:SOT-23_Handsoldering
T1	LM336Z-5.0	Housings_SOIC:SOIC-8_3.9x4.9mm_Pitch1.27mm

Literatura

- [1] Dokumentacja techniczna: *biblioteki Segger emWin* (dostęp: 20 czerwca 2017)
- [2] Dokumentacja techniczna: *FT5336* (dostęp: 20 czerwca 2017)
- [3] Dokumentacja techniczna: *MT48LC4M32B2* (dostęp: 20 czerwca 2017)
- [4] Dokumentacja techniczna: *STM32F746xx* (dostęp: 20 czerwca 2017)
- [5] User Manual: *Discovery kit for STM32F7 Series with STM32F746NG MCU* (dostęp: 20 czerwca 2017)
- [6] Nota aplikacyjna: *biblioteki STemWin* (dostęp: 20 czerwca 2017)
- [7] Strona internetowa: <http://microelk.azurewebsites.net/> (dostęp: 20 czerwca 2017)
- [8] Adam Pyka. *Zabezpieczenie akumulatora Li-Poly*. KoNaR, Wrocław, 2011. (dostęp: 20 czerwca 2017)
- [9] Dokumentacja techniczna: *ST LSM9DS1 IMU (9 DoF)* (dostęp: 20 czerwca 2017)
- [10] Dokumentacja techniczna: *Moduł Bluetooth (HC-05)* (dostęp: 20 czerwca 2017)

Rysunek 10: Widok PCB





Dawid Marszatkiewicz 218665, Piotr Portasiak 218594

Sheet: /

File: InZdaSter.sch

Title: InZdaSter 2.0

Size: A4 Date: 2017-05-17

KiCad E.D.A. kicad 4.0.2+dfsg1-stable

Rev: 1

Id: 1/1