

PROJEKT

ROBOTY MOBILNE

Dokumentacja

Linefollower

LF

Skład grupy:

Cyprian HRYNIUK, 235512

Tomasz MASŁOŃ, 235827

Termin: wtTP15

Prowadzący:

mgr inż. Michał BŁĘDOWSKI

8 czerwca 2019

Spis treści

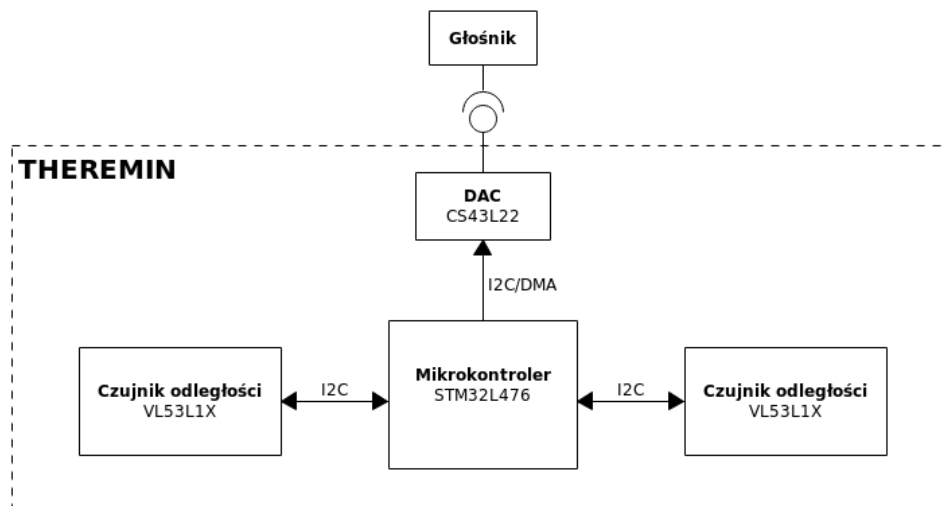
1	Opis projektu	2
2	Konfiguracja mikrokontrolera	3
2.1	Konfiguracja pinów	5
2.2	I2C	5
2.3	UART	5
2.4	DMA	5
3	Harmonogram pracy	6
3.1	Podział pracy	6
4	Podsumowanie	6
	Bibilografia	7

1 Opis projektu

Projekt ma na celu stworzenie linefollowera - robota klasy k(2,0) poruszającego się po wyznaczonej linii. Prace projektowe zawierać w sobie będą projekt i wykonanie układu mechanicznego i układu elektronicznego, wytworzenie oprogramowania oraz montaż robota.

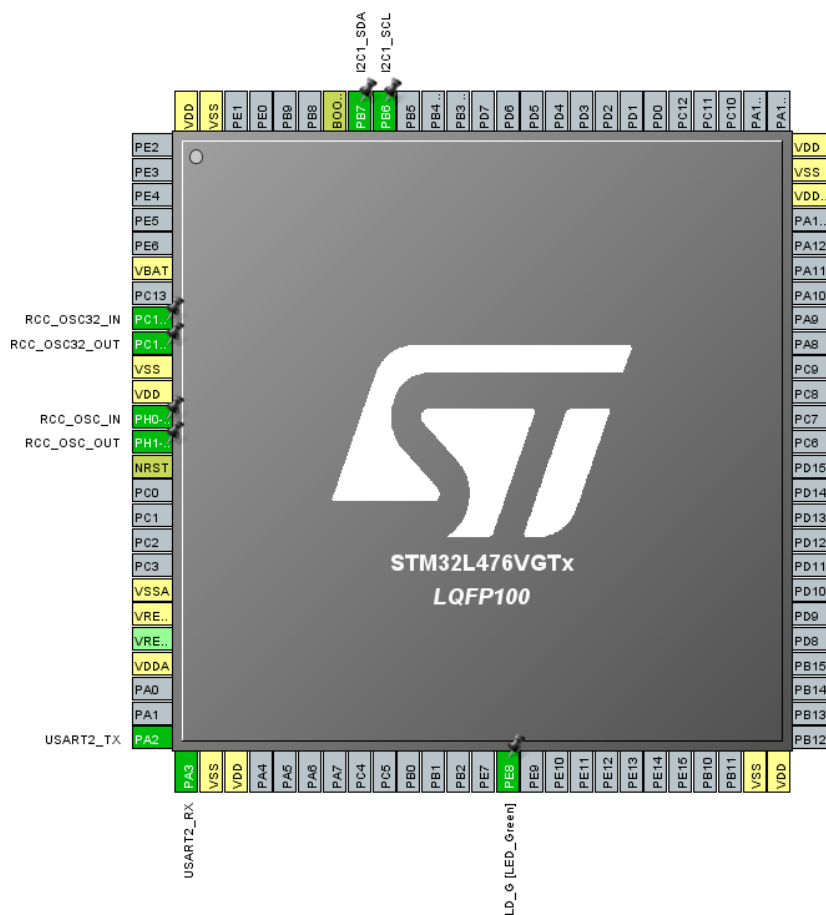
Robot będzie wykrywał linię za pomocą zestawu czujników odbiciowych. Sygnał z czujników przekazywany będzie do mikrokontrolera, który w oparciu o dane z czujników, będzie sterował silnikami prądu stałego pełniącymi rolę napędową kół. Całość zasilana będzie akumulatorem litowo-polimerowym.

Sterowanie silnikami odbywać się będzie przy pomocy mostków-H. Algorytm sterujący wykorzystywać będzie regulator PD (w którym błąd będzie określany na podstawie sygnałów z czujników i odzwierciedlać będzie odbieganie toru jazdy od aktualnego kierunku, w którym linefollower się porusza), aby płynnie reagować na zmiany toru jazdy.



Rysunek 1: Architektura systemu

2 Konfiguracja mikrokontrolera



Rysunek 2: Konfiguracja wyjść mikrokontrolera w programie STM32CubeMX

2.1 Konfiguracja pinów

Numer pinu	PIN	Tryb pracy	Funkcja/etykieta
8	PC14	OSC32_IN	RCC_OSC32_IN
9	PC15	OSC32_OUT	RCC_OSC32_OUT
12	PH0	OSC_IN	RCC_OSC_IN
13	PH1	OSC_OUT	RCC_OSC_OUT
39	PE8	GPIO_Output	LD_Green
86	PD5	USART2_TX	USART2_TX
87	PD6	USART2_RX	USART2_RX
92	PB6	I2C1_SCL	I2C1_SCL
93	PB7	I2C1_SDA	I2C1_SDA

Tabela 1: Konfiguracja pinów mikrokontrolera

2.2 I2C

Parametr	Wartość
I2C Speed Mode	Standard Mode
I2C Speed Frequency	100kHz

Tabela 2: Konfiguracja peryferium I2C

2.3 UART

Peryferium skonfigurowano na potrzeby debugowania.

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

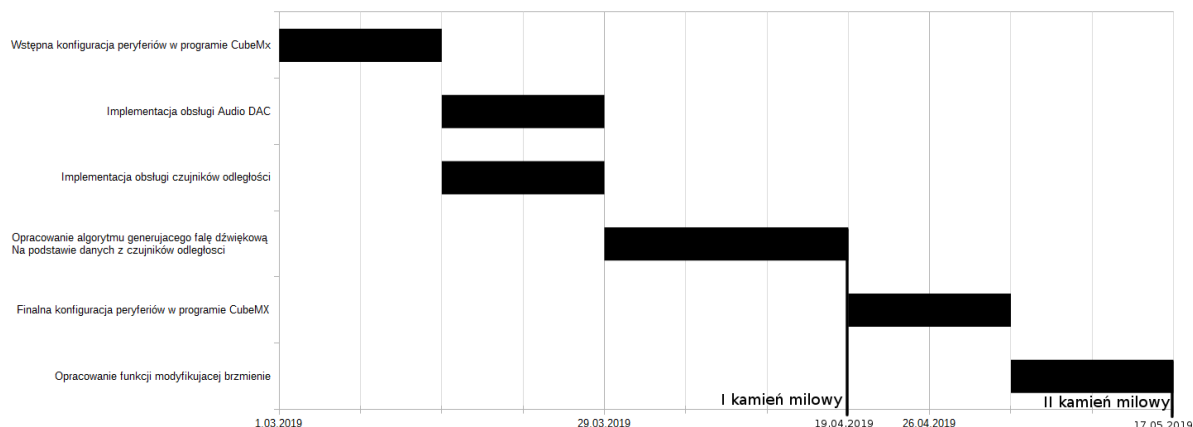
Tabela 3: Konfiguracja peryferium USART

2.4 DMA

Parametr	Wartość
DMA request	DAC_CH1
Stream	DMA1_Channel3
Direction	Memory To Peripheral

Tabela 4: Konfiguracja peryferium DMA

3 Harmonogram pracy



Rysunek 4: Diagram Gantta

Kolejne kamienie milowe pokrywają się z etapami wyszczególnionymi w podrozdziale Podział pracy.

3.1 Podział pracy

Tomasz Masłoń	%	Cyprian Hryniuk	%
Wstępna konfiguracja peryferiów w programie CubeMx		Wstępna konfiguracja peryferiów w programie CubeMx	
Implementacja obsługi Audio DAC		Implementacja obsługi czujników odległości	
Opracowanie algorytmu generującego falę dźwiękową na podstawie danych z czujników odległości		Opracowanie algorytmu generującego falę dźwiękową na podstawie danych z czujników odległości	

Tabela 5: Podział pracy – Etap II

Tomasz Masłoń	%	Cyprian Hryniuk	%
Finalna konfiguracja peryferiów w programie CubeMX		Finalna konfiguracja peryferiów w programie CubeMX	
Opracowanie funkcji modyfikującej brzmienie		Opracowanie funkcji modyfikującej brzmienie	

Tabela 6: Podział pracy – Etap III

4 Podsumowanie

Projekt dotyczy stworzenia instrumentu muzycznego - theremina, przy wykorzystaniu cyfrowych czujników odległościowych oraz Audio DAC.

Literatura

- [1] CirrusLogic. CS43L22 Datasheet. 2010.
- [2] STMicroelectronics. AN3126 Application note Audio and waveform generation using the DAC in STM32 microcontrollers. Maj 2017.
- [3] STMicroelectronics. UM1879 User manual Discovery kit with STM32L476VG MCU. Maj 2017.
- [4] STMicroelectronics. STM32L476xx Datasheet. Maj 2018.
- [5] STMicroelectronics. VL53L1X Datasheet. 2018.