

PROJEKT

STEROWNIKI ROBOTÓW

Założenia projektowe

Theremin THR

Skład grupy:

Cyprian HRYNIUK, 235512

Tomasz MASŁOŃ, 235827

Termin: srTN17

Prowadzący:

mgr inż. Wojciech DOMSKI

12 marca 2019

Spis treści

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

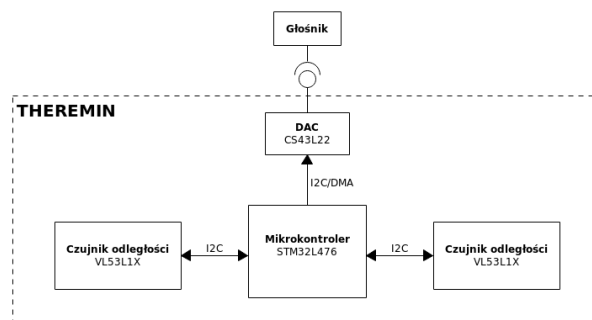
1 Opis projektu

Celem projektu jest stworzenie theremina opartego na płytce rozwojowej STM32L476G-DISCO i dwóch cyfrowych czujnikach odległości VL53L1X korzystających z technologii Time-of-Flight. Dane z jednego czujnika będą wpływać na częstotliwość dźwięku a dane z drugiego na głośność. Aby uzyskać zamierzony efekt niezbędne będzie zaprogramowanie obsługi czujników oraz Audio DAC zawartego na płytce.

Głównym zadaniem jest napisanie pełnego oprogramowania na MCU STM32L476VGT6 pozwalającego na generowanie fali dźwiękowej (sinusoidea). Modyfikacje do generowanej fali będą wprowadzane ruchami rąk (bliżej-dalej). Informacja na temat odległości będzie podawana przez czujniki VL53L1X. Na początku zostanie zaimplementowana funkcja generująca podstawową sinusoidę, funkcję tę później rozwinieemy tak, aby generowała ciekawe brzmienie.

Czujniki będą się komunikować z MCU za pomocą protokołu I^2C . Ustawienia Audio DAC również będą realizowane za pomocą protokołu I^2C . Przekazanie próbek do Audio DAC odbędzie się za pomocą DMA.

Projekt będzie tworzony w języku C korzystając ze środowiska Atollic TrueSTUDIO for STM32. Konfiguracja peryferiów zostanie zrealizowana przy pomocy programu STM32CubeMX. Wszelkie dokumenty dotyczące projektu będą powstawać w systemie składu tekstu \LaTeX . Na potrzeby projektu zostało założone zdalne repozytorium w hostingowanym serwisie internetowym *github.com* korzystającym z systemu kontroli wersji *git*. Repozytorium jest publiczne i dostępne pod linkiem: <https://github.com/CypHry/Theremin>

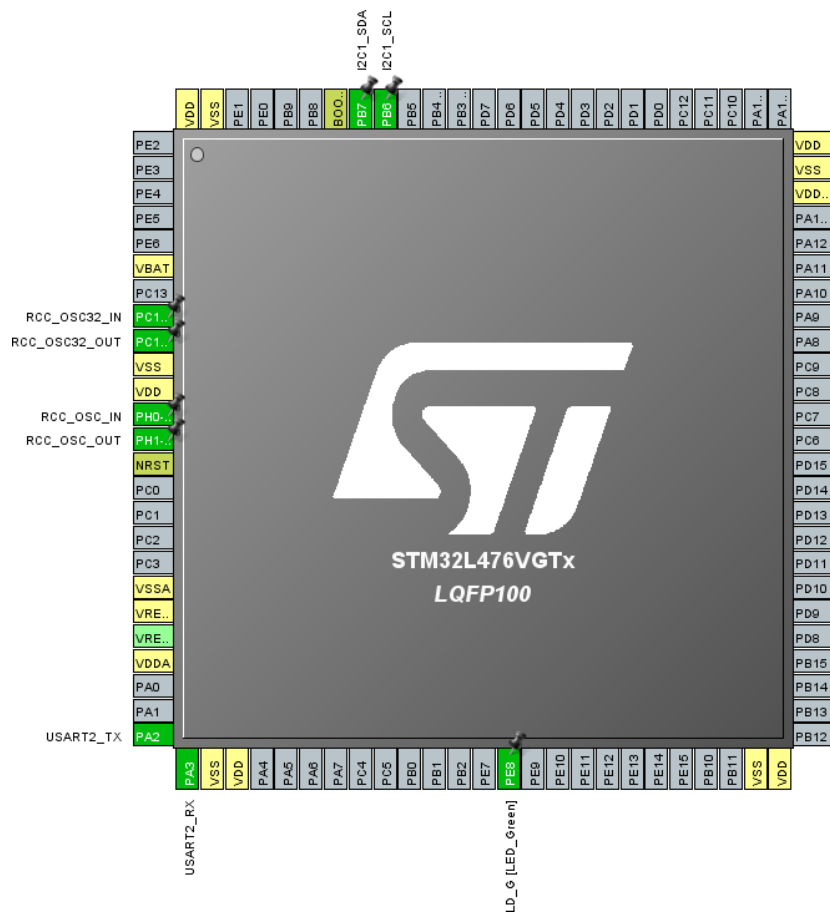


Rysunek 1: Architektura systemu

2 Konfiguracja mikrokontrolera

Tutaj powinna znaleźć się konfigurację poszczególnych peryferiów mikrokontrolera – jeśli wykorzystywany jest np. ADC to należy podać jego konfigurację nie zapominając o DMA jeśli jest wykorzystywane. Proszę wzorować się na raporcie wygenerowanym z programu STM32CubeMx (plik PDF i TXT, Project -> Generate Report Ctrl+R). W pliku PDF jest to rozdział *IPs and Middleware Configuration*. Należy umieścić uproszczoną konfigurację peryferiów w formie tabel (najistotniejsze parametry + parametry zmienione, pogrubione). Dodatkowo w pliku tekstowym (TXT) znajduje się konfiguracja pinów mikrokontrolera, którą również należy zamieścić w raporcie.

W przypadku, gdy projekt zakłada wykorzystanie większej liczby modułów sekcję tą należy podzielić na odrębne podsekcje.



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_G [LED_Green]
47	PB10	I2C2_SCL	
48	PB11	I2C2_SDA	
86	PD5	USART2_TX	USART_TX
87	PD6	USART2_RX	USART_RX
92	PB6	I2C1_SCL	
93	PB7	I2C1_SDA	

Tabela 1: Konfiguracja pinów mikrokontrolera

2.2 I2C

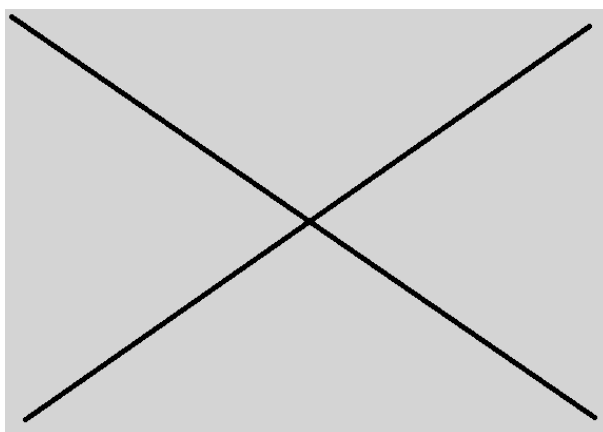
Przykładowa konfiguracja peryferium interfejsu szeregowego. Należy opisać do czego będzie wykorzystywany interfejs. Zmiany, które odbiegają od standardowych w programie CubeMX powinny być zaznaczone innym kolorem, jak to zostało pokazane w tabeli 2.

Parametr	Wartość
Baud Rate	11520
Word Length	8 Bits (including parity)
Parity	None
Stop Bits	1

Tabela 2: Konfiguracja peryferium USART

3 Harmonogram pracy

Należy wstawić diagram Gantta oraz określić ścieżkę krytyczną. Ponadto zaznaczyć i opisać kamienie milowe.



Rysunek 4: Diagram Gantta

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 modulującego falę dźwiękową na podstawie danych z czujników odległości		Opracowanie algorytmu modulującego falę dźwiękową na podstawie danych z czujników odległości	

Tabela 3: 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 4: Podział pracy – Etap III

4 Podsumowanie

Krótkie podsumowanie projektu

Literatura