

PROJEKT

STEROWNIKI ROBOTÓW

Dokumentacja

Monitor hałasu

MonHa

Skład grupy:

Szymon LEWANDOWSKI, 235203

Filip ADAMCEWICZ, 235130

Termin: srTP15

Prowadzący:

mgr inż. Wojciech DOMSKI

8 czerwca 2020

Spis treści

1	Opis	2
2	Konfiguracja mikrokontrolera	2
2.1	Konfiguracja pinów	4
2.2	QSPI i pamięć flash	4
2.3	RTC	5
2.4	Timer	5
2.5	USART2	5
3	Urządzenia zewnętrzne	6
3.1	LCD	6
3.2	Mikrofon MEMS SAI_1	6
3.3	FLASH	7
3.4	Joystick	7
4	Opis działania programu	7
4.1	Menu	8
4.2	Wykrywanie przekroczenia progu	8
4.3	Okresowe pomiary	8
5	Podsumowanie	9
	Bibilografia	9

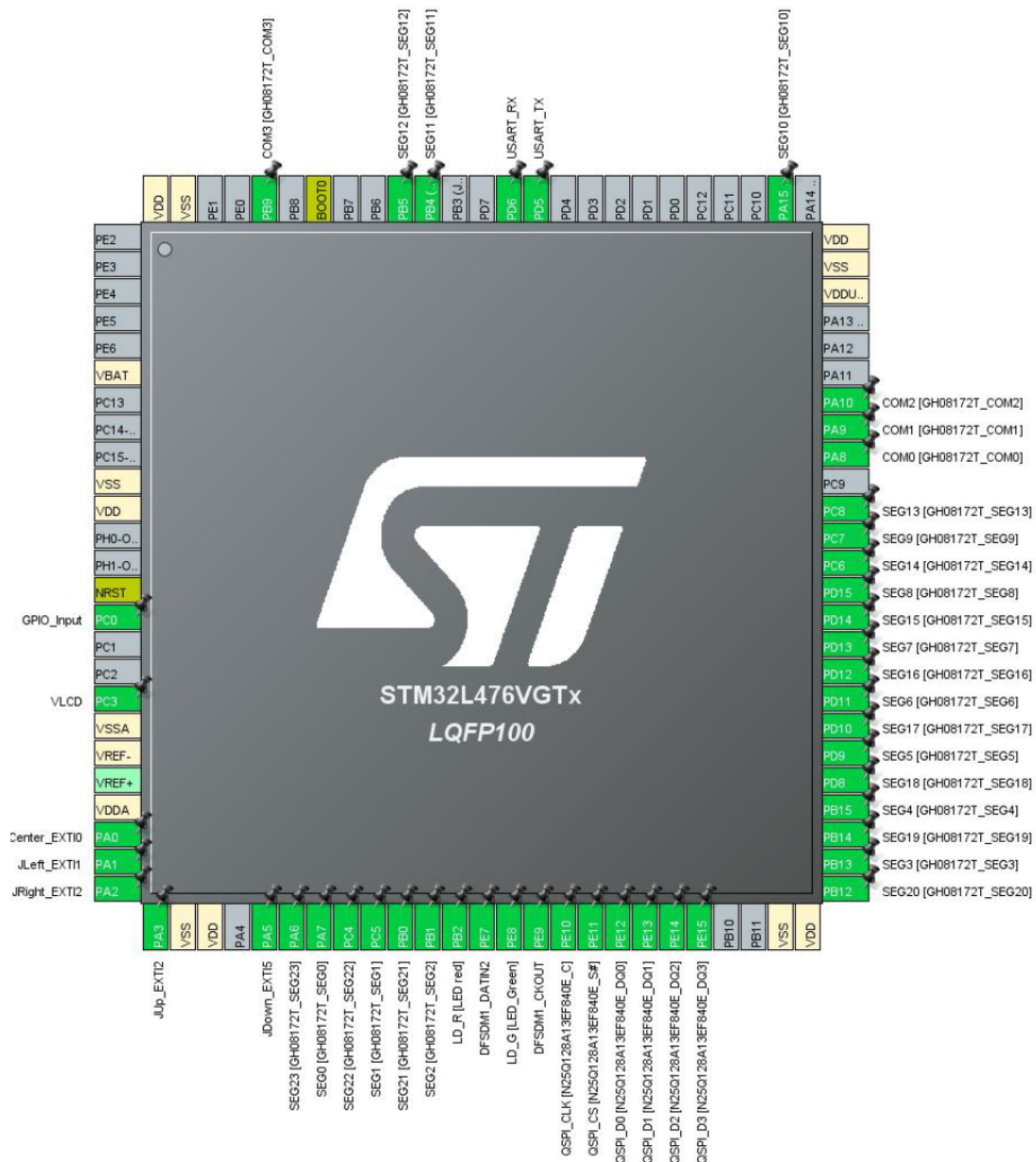
1 Opis

Zadaniem projektu jest stworzenie układu którego zadaniem będzie monitorowanie poziomu hałasu oraz zapisywanie zdarzeń do zewnętrznej pamięci FLASH. Planowane są dwa tryby pracy loggera:

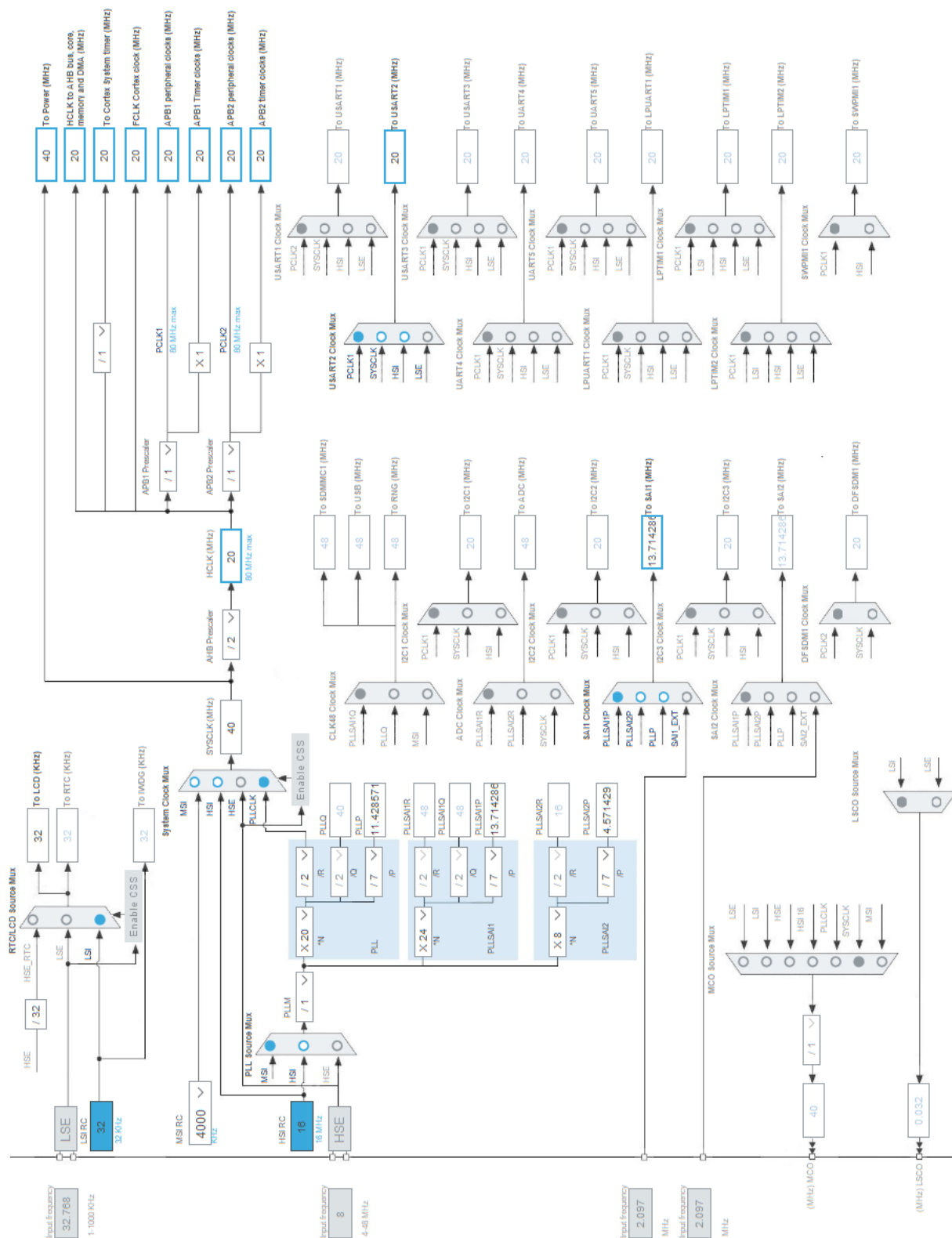
- zapisywanie kiedy poziom hałasu przekroczy wyznaczony próg
- okresowe mierzenie próbek i prowadzenie ich statystyki

Menu programu oraz informacje będą wyświetlane na wyświetlaczu LCD. Wybór opcji menu joystickiem na płycie.

2 Konfiguracja mikrokontrolera



Rysunek 1: Konfiguracja wyjść mikrokontrolera w programie STM32CubeMX



Rysunek 2: Konfiguracja zegarów mikrokontrolera

2.1 Konfiguracja pinów

Numer pinu	PIN	Tryb pracy	Funkcja/etykieta
15	PC0	GPIO_Input	VLCD
18	PC3	LCD_VLCD	
23	PA0	GPIO_EXTI0	JCenter_EXTI0
24	PA1	GPIO_EXTI1	JLeft_EXTI1
25	PA2	GPIO_EXTI2	JRight_EXTI2
26	PA3	GPIO_EXTI3	JUp_EXTI2
30	PA5	GPIO_EXTI5	JDown_EXTI5
31	PA6	LCD_SEG3	SEG23 [GH08172T_SEG23]
32	PA7	LCD_SEG4	SEG0 [GH08172T_SEG0]
33	PC4	LCD_SEG22	SEG22 [GH08172T_SEG22]
34	PC5	LCD_SEG23	SEG1 [GH08172T_SEG1]
35	PB0	LCD_SEG5	SEG21 [GH08172T_SEG21]
36	PB1	LCD_SEG6	SEG2 [GH08172T_SEG2]
37	PB2	GPIO_Output	LD_R [LED red]
38	PE7	DFSDM1_DATIN2	
39	PE8	GPIO_Output	LD_G [LED_Green]
40	PE9	DFSDM1_CKOUT	
41	PE10	QUADSPI_CLK	QSPI_CLK [N25Q128A13EF840E_C]
42	PE11	QUADSPI_NCS	QSPI_CS [N25Q128A13EF840E_S#]
43	PE12	QUADSPI_BK1_IO0	QSPI_D0 [N25Q128A13EF840E_DQ0]
44	PE13	QUADSPI_BK1_IO1	QSPI_D1 [N25Q128A13EF840E_DQ1]
45	PE14	QUADSPI_BK1_IO2	QSPI_D2 [N25Q128A13EF840E_DQ2]
46	PE15	QUADSPI_BK1_IO3	QSPI_D3 [N25Q128A13EF840E_DQ3]
51	PB12	LCD_SEG12	SEG20 [GH08172T_SEG20]
52	PB13	LCD_SEG13	SEG3 [GH08172T_SEG3]
53	PB14	LCD_SEG14	SEG19 [GH08172T_SEG19]
54	PB15	LCD_SEG15	SEG4 [GH08172T_SEG4]
55	PD8	LCD_SEG28	SEG18 [GH08172T_SEG18]
56	PD9	LCD_SEG29	SEG5 [GH08172T_SEG5]
57	PD10	LCD_SEG30	SEG17 [GH08172T_SEG17]
58	PD11	LCD_SEG31	SEG6 [GH08172T_SEG6]
59	PD12	LCD_SEG32	SEG16 [GH08172T_SEG16]
60	PD13	LCD_SEG33	SEG7 [GH08172T_SEG7]
61	PD14	LCD_SEG34	SEG15 [GH08172T_SEG15]
62	PD15	LCD_SEG35	SEG8 [GH08172T_SEG8]
63	PC6	LCD_SEG24	SEG14 [GH08172T_SEG14]
64	PC7	LCD_SEG25	SEG9 [GH08172T_SEG9]
65	PC8	LCD_SEG26	SEG13 [GH08172T_SEG13]
67	PA8	LCD_COM0	COM0 [GH08172T_COM0]
68	PA9	LCD_COM1	COM1 [GH08172T_COM1]
69	PA10	LCD_COM2	COM2 [GH08172T_COM2]
77	PA15 (JTDI)	LCD_SEG17	SEG10 [GH08172T_SEG10]
86	PD5	USART2_TX	USART_TX
87	PD6	USART2_RX	USART_RX
90	PB4 (NJTRST)	LCD_SEG8	SEG11 [GH08172T_SEG11]
91	PB5	LCD_SEG9	SEG12 [GH08172T_SEG12]
96	PB9	LCD_COM3	COM3 [GH08172T_COM3]

Tabela 1: Konfiguracja pinów mikrokontrolera

2.2 QSPI i pamięć flash

Quad-SPI służy do komunikacji z układem scalonym zawierającym pamięć flash. Konfiguracje przedstawiono w tabeli 2.

Parametr	Wartość
clock prescaler	255
FIFO threshold	1
Sample Shifting	No Sample Shifting
Flash size	1
Chip Select High Time	1 Cycle
Clock Mode	Low

Tabela 2: Konfiguracja peryferium QSPI

2.3 RTC

Wykorzystywany do kontroli czasu niezbędnej do prawidłowego notowania czasu zdarzeń. Konfigurację RTC wykonano z pomocą kursu ze strony *Forbot* [2].

Parametr	Wartość
Hour Format	Hourformat 24
Asynchronous Predivider value	127
Synchronous Predivider value	255
Data Format	Binary data format*
Hours	23*
Minutes	59*
Seconds	50*
Day Light Saving	daylightsaving none
Store Operation	storeoperation reset
Week Day	Wednesday*
Month	April*
Date	24*
Year	19*

Tabela 3: Konfiguracja RTC

2.4 Timer

W celu odmierzenia czasu podczas trybu okresowego mierzenia próbek, ustawiono *timer 6* na częstotliwość 1 Hz. Konfigurację przedstawiono w tabeli 4. [1]

Parametr	Wartość
Prescaler (PSC)	9999*
Counter Mode	Up
Period (ARR)	1999*
auto-reload preload	Disabled

Tabela 4: Konfiguracja Timera TIM6

2.5 USART2

USART2 używany jest do przekazywania komunikatów do komputera. Zależnie od potrzeb przekazywane są nim różne teksty, wartości. Konfigurację interfejsu przedstawiono w tabeli 5.

Parametr	Wartość
Baud Rate	115200 Bits/s
Word Length	8 Bits
Parity	None
Stop Bits	1
Data Direction	Receive and Transmit
Over Sampling	16 Samples

Tabela 5: Konfiguracja peryferium USART2

3 Urządzenia zewnętrzne

3.1 LCD

Ekran umożliwiający poruszanie się po menu oraz wyświetlanie wyników logowania.

Parametr	Wartość
Mode	1/4 duty cycle
Clock Prescaler	1
Synchronous Predivider value	16
Duty Selection	1/4
Bias Selector	1/4
Multiplex mode	Disable
Voltage Source Selection	Internal
Contrast Control	2.60 V
Dead Time Duration	No dead Time
High Drive	Disable
Pulse ON duration	0 pulse
Blink mode	Disabled
Blink Frequency	fLCD/8

Tabela 6: Konfiguracja LCD

3.2 Mikrofon MEMS SAI_1

Wykorzystywany do pobierania próbek głośności otoczenia przez mikrofon SAI B:

Parametr	Wartość
Mode	1/4 duty cycle
Synchronization Inputs	Asynchronous
Protocol	free
Audio Mode	Master Transmit
Frame Length	64 bits *
Data Size	16 Bits *
Slot Size	DataSize
Output Mode	Stereo
Companding Mode	No companding mode
SAI SD Line Output Mode	Driven
First Bit	MSB First
Frame Synchro Active Level Length	32 *
Frame Synchro Definition	Channel Identification *
Frame Synchro Polarity	Active Low
Frame Synchro Offset	First Bit
First Bit Offset	0
Number of Slots (only Even Values)	4 *
Slot Active Final Value	0x0000FFFF *
Slot Active	All *
Master Clock Divider	Enabled
Audio Frequency	16 KHz *
Real Audio Frequency	16.071 KHz *
Error between Selected	0.44*
Clock Strobing	Falling Edge
Fifo Threshold	Empty
Output Drive	Disabled

Tabela 7: Konfiguracja LCD

3.3 FLASH

W pamięci flash są zapisywane odpowiednio czasy wystąpienia przekroczenia progu lub zarejestrowane poziomy dźwięku. Do obsługi pamięci i komunikacji z nią wykorzystano bibliotekę BSP, załączoną do CubeMX.

3.4 Joystick

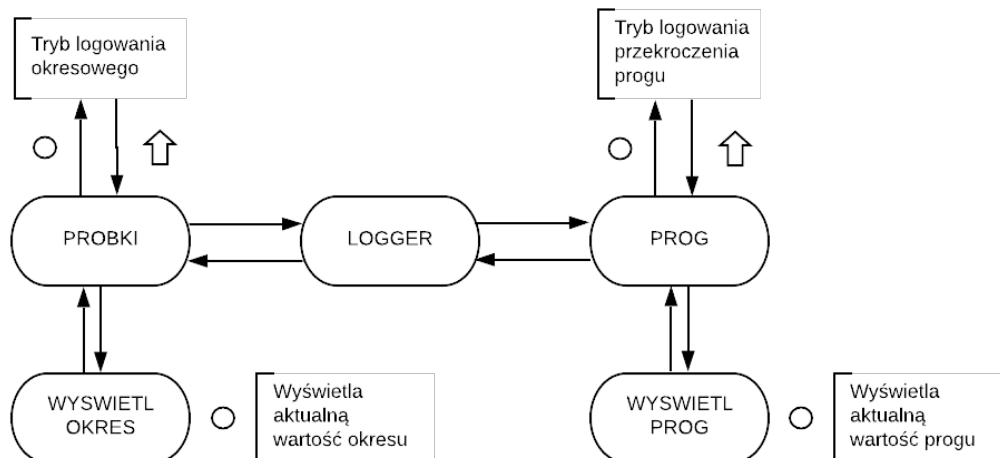
Joystick służy do poruszania się po menu. Przesunięcie w odpowiednim kierunku (zbocze narastające na przycisku) powoduje przerwanie, którego flaga aktywuje odpowiednie funkcje. Na podstawie dokumentacji płytki STM [3], należy w Cubie piny joysticka są w trybie *Pull-down*.

4 Opis działania programu

Podczas rozpoczęcia programu następuje inicjalizacja zegara RTC, LCD, komunikacji QPSI oraz mikrofon. Następnie wchodzi się w pętlę opartą na menu. Przeszczanie się po programie opiera się na menu oraz joysticku. Każda pozycja i funkcja menu wyświetla informacje na LCD. Funkcje loggera realizowane są jako funkcje odpowiednich pozycji menu.

Pomiędzy pozycjami na jednym poziomie przemieszcza się wciskając prawo/lewo, do podpoziomów przechodzi się w dół. Aktywowanie funkcji środkowym przyciskiem. Przejście do górnego poziomu lub wyjście z działającej funkcji menu za pomocą przycisku w górę.

Schemat ideowy przedstawiono na rys. 3. Strzałki przedstawiają faktyczny sposób poruszania się pomiędzy pozycjami, chyba że figura pomiędzy blokami przekazuje inaczej. Kółko oznacza wciśnięcie środkowego przycisku.



Rysunek 3: Schemat ideowy menu

4.1 Menu

Menu opiera się na liście dwustronnej. Każda pozycja jest opisana odpowiednią relacją z sąsiadami, ma swoją nazwę oraz może mieć przypisaną funkcję. Strukturę pozycji *Menu*, przedstawiono na listingu poniżej.

```

1 struct Menu{                                // wzorowane na liście dwustronnej
2     const char* Nazwa_Pozycji;
3     Pozycja_Menu* Nastepna;
4     Pozycja_Menu* Poprzednia;
5     Pozycja_Menu* Rodzic;
6     Pozycja_Menu* Dziecko;
7     void (*Funkcja_Pozycji)(void); // wskaznik na funkcje danej Pozycji
8 };

```

Menu wzorowane na artykułach z blogu *EmbeddedDev* [4].

4.2 Wykrywanie przekroczenia progu

Program działa w pętli, aż zostanie wciśnięty przycisk do góry. Najpierw do flash zostaje zapisana data z RTC.

Za pomocą DFSDM (Digital Filter for Sigma Delta Modulation) przetwarzany jest sygnał PDM pochodzący z mikrofonu cyfrowego na sygnał 'analogowy'.

Kolejnie pomiary są filtrowane za pomocą filtra FIR środkowo-przepustowego dla częstotliwości 300Hz-300kHz i na podstawie maksimów obecnych w przefiltrowanym sygnale ustalane jest maksymalne natężenie dźwięku.

Każde przekroczenie progu, powoduje zapisanie czasu zegara do pamięci flash.

Kiedy pętla zostaje przzerwana, logi są wyświetlane na LCD, w sposób 'LOG X', gdzie X to numer logu, a następnie wartości czasu wystąpienia przekroczenia. Oba napisy są wyświetlane przez określony czas.

4.3 Okresowe pomiary

Program działa w pętli, aż zostanie wciśnięty przycisk do góry. Program pobiera aktualny czas i w zależności czy minął czas pojedynczej próbki lub czas równy okresowi próbkowania, odpowiednio zapisuje do tablicy statystyki zmierzoną wartość lub wylicza na podstawie wartości w tablicy statystyki wartość średnią, zapisuje ją do pamięci i wyświetla za pomocą ekranu LCD.

Kiedy pętla zostaje przzerwana, logi są wyświetlane na LCD, w sposób 'XXXXEY', gdzie XXXX to pierwsze 4 cyfry pomiaru a Y to eksponenta pomiaru.

5 Podsumowanie

Projekt zakończony pozytywnie.

Możliwe rozszerzenia projektu uwzględniają:

- Rozbudowanie logowania o aktywowanie alarmu dla użytkownika
- Dodanie zabezpieczenia przy zbliżaniu się do limitu pamięci flash, bloku, lub całościowej.
- Zmianę wartości progu i okresu jako opcję w menu
- Zamianę liczbowych wartości wyświetlanych poziomu głośności na przyjaźniejsze użytkownikowi, typu 'cicho', 'średnio', 'głośno'

Potencjalne zastosowanie monitora hałasu może być w:

- Sprawdzaniu obecności niepożądanych osób, lub zdarzeń, w pomieszczeniu, podczas nieobecności użytkownika
- Monitorowanie stanu osób/obiektu, np. czy dziecko nie płacze, albo głośnej awarii maszyny
- Rejestr poziomu głośności w pomieszczeniu na przestrzeni czasu, np. w szkolnej klasie

Link do repozytorium na Githubie

Literatura

- [1] W. Domski. Sterowniki robotów, Laboratorium – Wprowadzenie, Wykorzystanie narzędzi STM32CubeMX oraz SW4STM32 do budowy programu mrugającej diody z obsługą przycisku. Mar. 2017.
- [2] FORBOT. Kursy STM32.
- [3] STMicroelectronics. User_Manual_1879. 2018.
- [4] Łukasz Łaguna. EmbeddedDev - menu na wyświetlaczu LCD.