

PROJEKT

PROJEKT ZESPOŁOWY

Założenia projektowe

Złożone urządzenie ogólnorozwojowe

ZuO

Skład grupy:

Krzysztof KURNIK, xxx

Paula LANGKAFEL, xxx

Albert LIS, 235534

Maciej MARSZAŁEK, xxx

Michał MORUŃ, xxx

Termin: śr 8:15

Prowadzący:

mgr inż. Krzysztof ARENT

3 marca 2019

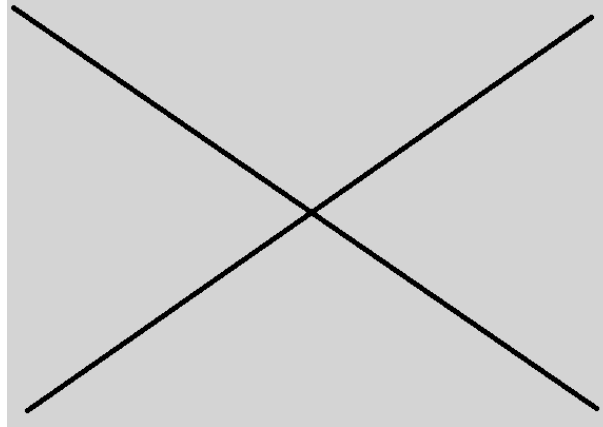
Spis treści

1	Opis projektu	2
2	Konfiguracja mikrokontrolera	2
2.1	Konfiguracja pinów	4
2.2	USART	4
3	Urządzenia zewnętrzne	4
3.1	Akcelerometr – LSM303C	4
4	Projekt elektroniki	5
4.1	Czujniki	5
5	Konstrukcja mechaniczna	5
6	Opis działania programu	5
7	Harmonogram pracy	6
7.1	Podział pracy	6
8	Zadania niezrealizowane	6
9	Podsumowanie	7
	Bibilografia	8

1 Opis projektu

Krótki opis projektu czego będzie on dotyczył.

W przypadku, gdy projekt dotyczy systemu wielomodułowego należy dołączyć diagram, który będzie prezentował architekturę systemu:

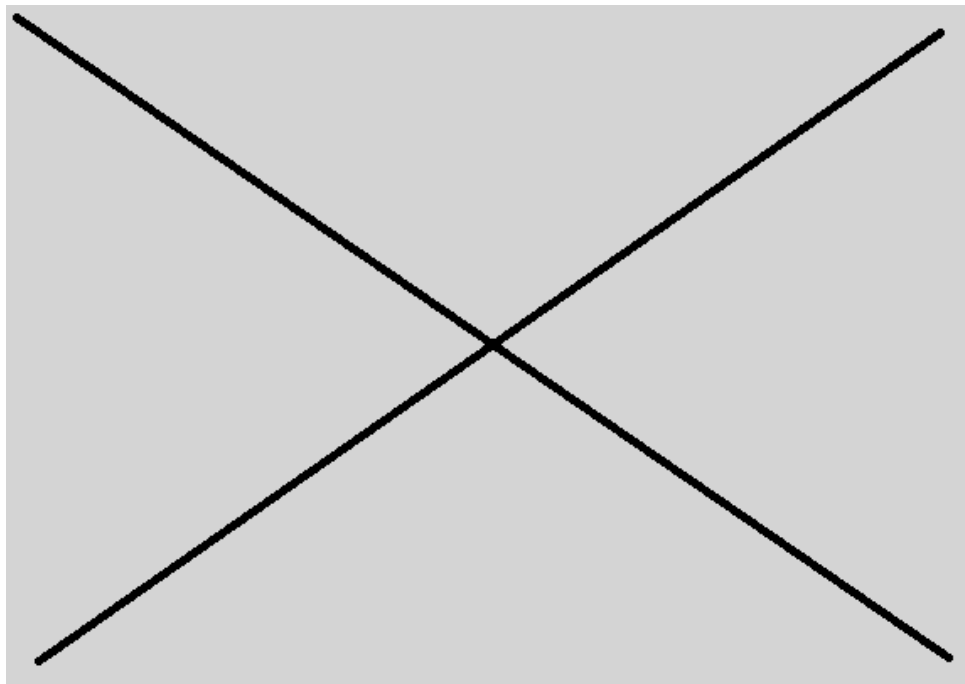


Rysunek 1: Architektura systemu

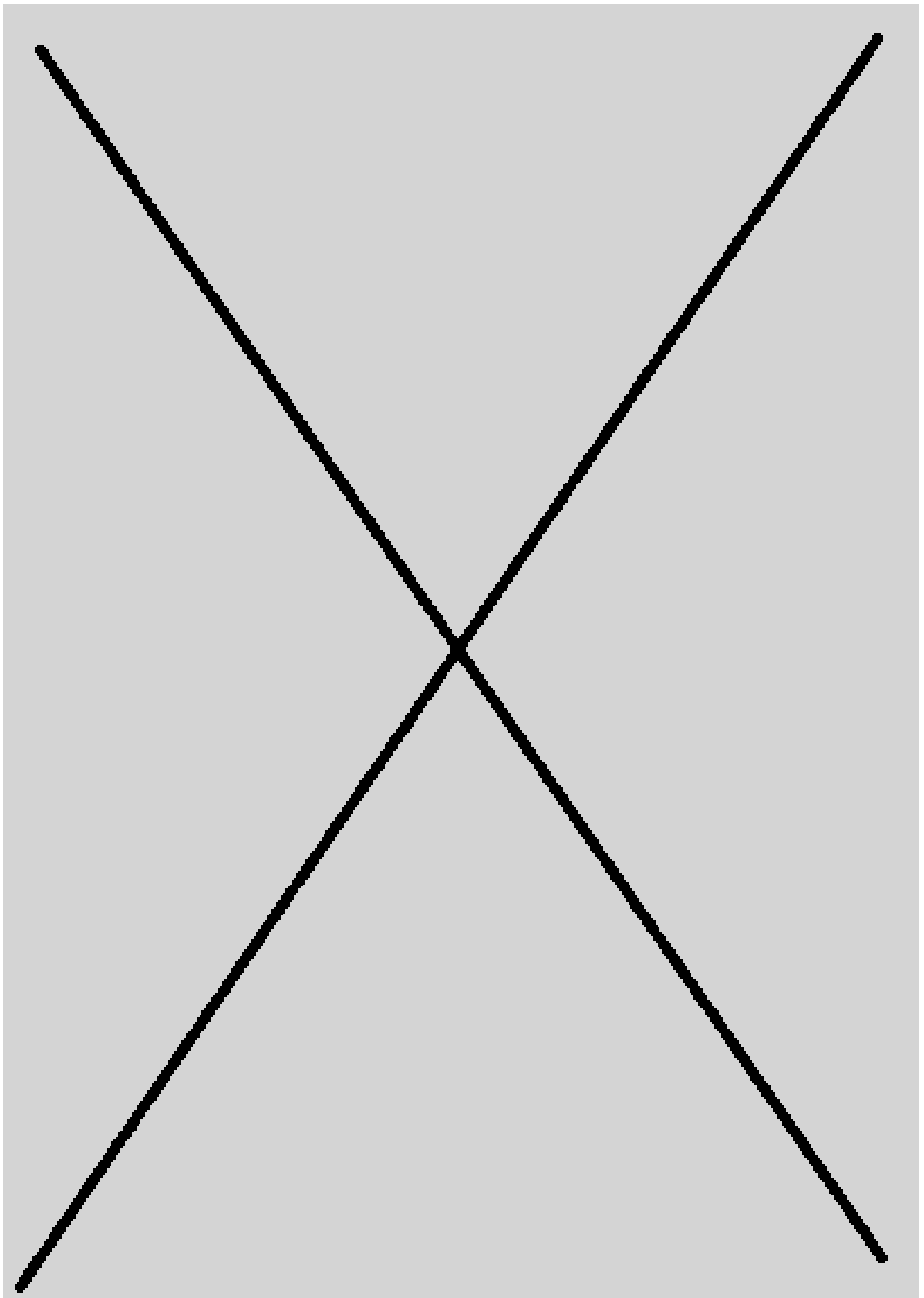
2 Konfiguracja mikrokontrolera

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 tabelek (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



Rysunek 3: Konfiguracja zegarów mikrokontrolera

2.1 Konfiguracja pinów

Numer pinu	PIN	Tryb pracy	Funkcja/etykieta
2	PC13	ANTI_TAMP GPIO_EXTI13	B1 [Blue PushButton]
3	PC14	OSC32_IN* RCC_OSC32_IN	
4	PC15	OSC32_OUT* RCC_OSC32_OUT	
5	PH0	OSC_IN* RCC_OSC_IN	
6	PH1	OSC_OUT*	RCC_OSC_OUT
16	PA2	USART2_TX	USART_TX
17	PA3	USART2_RX	USART_RX
21	PA5	GPIO_Output	LD2 [Green Led]
29	PB10	I2C2_SCL	I2C_SCL
41	PA8	TIM1_CH1	PWM1
46	PA13*	SYS_JTMS-SWDIO	TMS
49	PA14*	SYS_JTCK-SWCLK	TCK
55	PB3*	SYS_JTDO-SWO	SWO
62	PB9	I2C2_SDA	I2C_SCL

Tabela 1: Konfiguracja pinów mikrokontrolera

2.2 USART

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 Urządzenia zewnętrzne

Rozdział ten powinien zawierać opis i konfigurację wykorzystanych układów zewnętrznych, jak np. akcelerometr.

3.1 Akcelerometr – LSM303C

Akcelerometr został wykorzystany do ...

Konfiguracja rejestrów czujnika została zaprezentowana w ... Wpisanie tych wartości do rejestrów urządzenia ... powoduje ...

Rejestr	Wartość
CTRL_REG2 (0x21)	0x12
CTRL_REG3 (0x22)	0x13

Tabela 3: Konfiguracja peryferium USART

4 Projekt elektroniki

4.1 Czujniki

Wielkość mierzona	Symbol czujnika
Wilgotność + Temperatura	SHT11/30/31
Ciśnienie	Bmp085/180
Smog	GP2Y1010AU0F
Dym	MQ-2/MQ-9
Tlenek węgla (CO)	MQ-7
Alkohol	MQ-3/MQ-135
Dwutlenek węgla (CO ₂)	DFRobot SEN219 / MH-Z19
Temperatura (opcjonalnie)	DS18B20

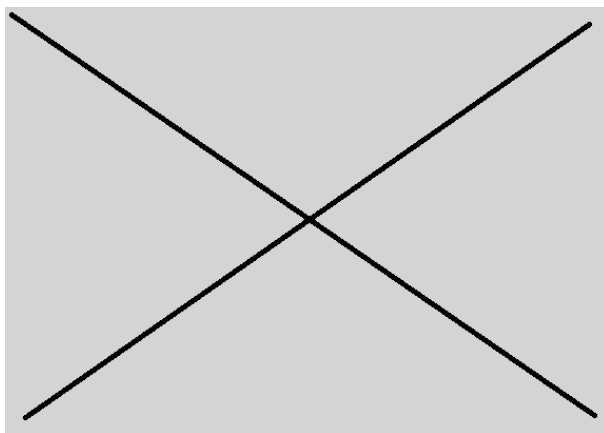
Tabela 4: Użyte czujniki

5 Konstrukcja mechaniczna

W przypadku, w którym projekt uwzględnia zastosowanie mechaniki to wówczas jej opis powinien znaleźć się tutaj. Nie należy dzielić rysunków mechaniki na poszczególne rzuty, wystarczy zamieścić wyrenderowane modele 3D. Można również dołączyć zdjęcia wykonanej mechaniki po uprzednim skompresowaniu, aby wynikowy rozmiar skompilowanego dokumentu nie był za duży.

6 Opis działania programu

Należy zawrzeć tutaj opis działania programu. Mile widziany diagram prezentujący pracę programu.



Rysunek 4: Diagram przepływu

Sekcję tą można podzielić na dodatkowe podsekcje w miarę potrzeb. Do tego celu należy wykorzystać *subsection*.

W przypadku, dodania istotnego fragmentu kodu należy posłużyć się środowiskiem `lstlisting`:

```
1 int foo(void){  
2 return 2;  
3 }
```

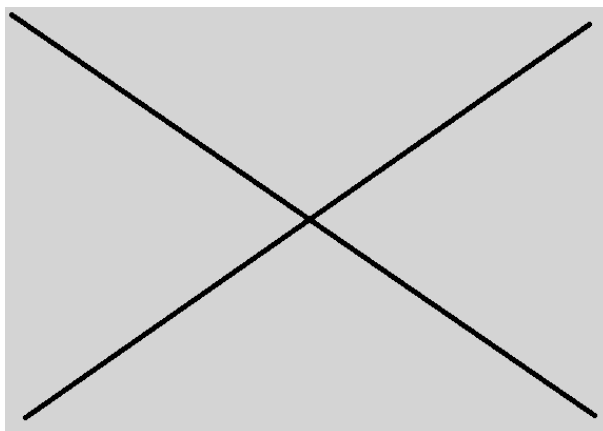
Przykładowy wzór (1):

$$\Theta = \int_t^{t+dt} \omega dt. \quad (1)$$

Przykładowa pozycja bibliograficzna [1] znajduje się w pliku `bibliografia.bib`.

7 Harmonogram pracy

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



Rysunek 5: Diagram Gantta

7.1 Podział pracy

Każdy z członków grupy powinien w każdym etapie mieć wymienione od 2 do 4 zadań. Przykładowa tabela podziału zadań dla etapu II (Tab. 5) oraz dla etapu III (Tab. 6) zostały przedstawione poniżej. Przy podziale prac nie uwzględniamy tworzenia dokumentacji projektu!

Przykładowy podział prac dla projektu pod tytułem "Automatyczny dyktafon rozmowy":

Adam Babacki	%	Bartłomiej Cabacki	%
Wstępna konfiguracja peryferiów w programie CubeMx		Wstępna konfiguracja peryferiów w programie CubeMx	
Implementacja obsługi mikrofonu		Opracowanie algorytmu automatycznej detekcji rozmowy	
Opracowanie sposobu przechowywania danych na zewnętrznej pamięci FLASH		Oprogramowanie testujące obsługę mikrofonu	
Odtwarzanie dźwięku za pomocą Audio DAC			

Tabela 5: Podział pracy – Etap II

Adam Babacki	%	Bartłomiej Cabacki	%
Finalna konfiguracja peryferiów w programie CubeMX		Finalna konfiguracja peryferiów w programie CubeMX	
Zapisywanie dźwięku na pamięć zewnętrzną FLASH		Integracja modułów	
Obsługa wyświetlacza ciekłokrystalicznego		Obsługa joysticka	
		Interfejs użytkownika	

Tabela 6: Podział pracy – Etap III

8 Zadania niezrealizowane

Jeśli wszystkie zadania zostały realizowane to wówczas ta sekcja powinna być usunięta w całości. W przeciwnym razie należy zawrzeć tutaj, jakie zadania zostały nie zrealizowane oraz jaka była tego przyczyna.

9 Podsumowanie

Krótkie podsumowanie projektu

Literatura

- [1] W. Domski. Sterowniki robotów, Laboratorium – Wprowadzenie, Wykorzystanie narzędzi STM32CubeMX oraz SW4STM32 do budowy programu mrugającej diody z obsługą przycisku. Mar. 2017.