

PROJEKT

STEROWNIKI ROBOTÓW

Założenia projektowe

Wykrywacz kradzieży

WK

Skład grupy:

Sylwester KOZIEJA, 235798

Paula LANGKAFEL, 235373

Termin: środa TN 13

Prowadzący:

mgr inż. Wojciech DOMSKI

19 maja 2019

Spis treści

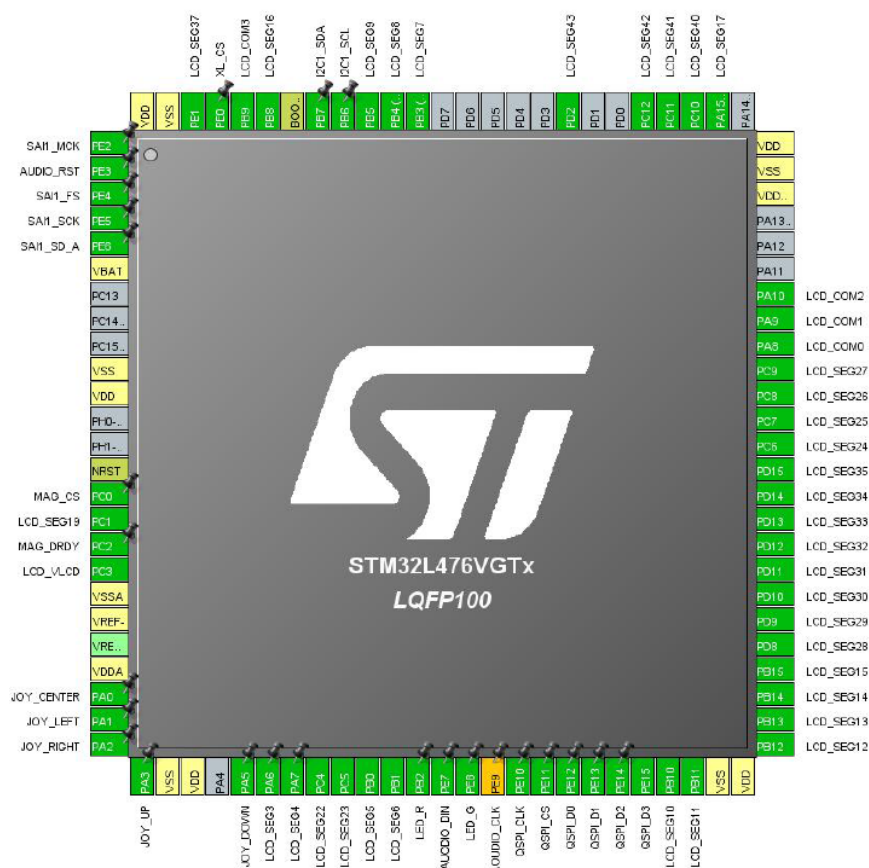
1	Opis projektu	2
2	Konfiguracja mikrokontrolera	2
2.1	Konfiguracja pinów mikrokontrolera	4
2.2	QUADSPI	5
2.3	LCD	5
2.4	GPIO	5
2.5	SPI	5
2.6	SAI	6
3	Urządzenia zewnętrzne	6
3.1	Akcelerometr – LSM303C	6
4	Opis działania programu	7
5	Założenia projektowe	7
6	Harmonogram pracy	7
6.1	Zakres prac	7
6.2	Kamienie milowe	7
6.3	Diagram Gantta	8
6.4	Podział pracy	8
7	Podsumowanie etapu II	8
7.1	Akcelerometr	8
7.2	Wyświetlacz z menu i zegarem RTC	9
7.3	Opis menu	9
	Bibilografia	10

1 Opis projektu

Celem projektu jest stworzenie urządzenia, które poprzez komunikację z akcelerometrem wykrywa niepożądany ruch. Użytkownik będzie miał możliwość wybrania sposobu otrzymywania komunikatów. Jednym z założeń projektu jest wybór opcjonalnego interfejsu audio. Urządzenia ma zawierać menu dające możliwość podstawowej konfiguracji, takiej jak załączenie alarmu i wybór sposobu komunikowania się oraz ustawianie poziomu załączania alarmu. Dodatkową opcją jest wizualizowanie poziomu rejestrowanych przyspieszeń.

Na zakres prac składa się oprogramowanie zewnętrznej pamięci Flash , skonfigurowanie akcelerometru, a także realizacja komunikacji z interfejsami.

2 Konfiguracja mikrokontrolera



Rysunek 1: Konfiguracja wyjść mikrokontrolera w programie STM32CubeMx

2.1 Konfiguracja pinów mikrokontrolera

Numer pinu	Pin	Tryb pracy	Funkcja/ etykieta
1	PE2	SAI1_MCLK_A	SAI1_MCK
2	PE3	GPIO_Output	AUDIO_RST
3	PE4	SAI1_FS_A	SAI1_FS
4	PE5	SAI1_SCK_A	SAI1_SCK
5	PE6	SAI1_SD_A	
15	PC0	GPIO_Output	MAG_CS
16	PC1	LCD_SEG19	
17	PC2	GPIO_Input	MAG_DRDY
18	PC3	LCD_VLCD	
23	PA0	GPIO_EXTI0	JOY_CENTER
24	PA1	GPIO_EXTI1	JOY_LEFT
25	PA2	GPIO_EXTI2	JOY_RIGHT
26	PA3	GPIO_EXTI3	JOY_UP
30	PA5	GPIO_EXTI5	JOY_DOWN
31	PA6	LCD_SEG3	
32	PA7	LCD_SEG4	
33	PC4	LCD_SEG22	
34	PC5	LCD_SEG23	
35	PB0	LCD_SEG5	
36	PB1	LCD_SEG6	
37	PB2	GPIO_Output	LED_R
38	PE7	SAI1_SD_B	AUDIO_DIN
39	PE8	GPIO_Output	LED_G
40	PE9*	SAI1_FS_B	AUDIO_CLK
41	PE10	QUADSPI_CLK	QSPI_CLK
42	PE11	QUADSPI_NCS	QSPI_CS
43	PE12	QUADSPI_BK1_IO0	QSPI_D0
44	PE13	QUADSPI_BK1_IO1	QSPI_D1
45	PE14	QUADSPI_BK1_IO2	QSPI_D2
46	PE15	QUADSPI_BK1_IO3	QSPI_D3
47	PB10	LCD_SEG10	
48	PB11	LCD_SEG11	
51	PB12	LCD_SEG12	
52	PB13	LCD_SEG13	
53	PB14	LCD_SEG14	
54	PB15	LCD_SEG15	
55	PD8	LCD_SEG28	
56	PD9	LCD_SEG29	
57	PD10	LCD_SEG30	
58	PD11	LCD_SEG31	
59	PD12	LCD_SEG32	
60	PD13	LCD_SEG33	
61	PD14	LCD_SEG34	
62	PD15	LCD_SEG35	
63	PC6	LCD_SEG24	
64	PC7	LCD_SEG25	
65	PC8	LCD_SEG26	
66	PC9	LCD_SEG27	
67	PA8	LCD_COM0	
68	PA9	LCD_COM1	
69	PA10	LCD_COM2	
77	PA15	(JTDI) LCD_SEG17	
78	PC10	LCD_SEG40	

Tabela 1: Konfiguracja pinów mikrokontrolera

Numer pinu	Pin	Tryb pracy	Funkcja/ etykieta
79	PC11	LCD_SEG41	XL_CS
80	PC12	LCD_SEG42	
83	PD2	LCD_SEG43	
89	PB3	(JTDO-TRACESWO) LCD_SEG7	
90	PB4	(NJTRST)LCD_SEG8	
91	PB5	LCD_SEG9	
92	PB6	I2C1_SCL	
93	PB7	I2C1_SDA	
95	PB8	LCD_SEG16	
96	PB9	LCD_COM3	
97	PE0	GPIO_Output	
98	PE1	LCD_SEG37	

Tabela 2: Konfiguracja pinów mikrokontrolera

2.2 QUADSPI

Interfejs użyty do obsługi pamięci Flash zapewniający dużą przepustowość dzięki czterem liniom danych (PE12:15) oraz dwóm liniom sterującym (PE10 i PE11).

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

2.3 LCD

Interfejs wyświetlacza umożliwiający komunikację z urządzeniem poprzez zaprojektowane menu. Korzysta z dużej ilości pinów, które opisane są na rysunku z konfiguracją. Ustawienia standardowe, parametr Duty Selection ustawiony na 1/4.

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

2.4 GPIO

Prosty interfejs do sterowania cyfrowymi wejściami/wyjściami. Użyty do obsługi LED i joystick'a. 5 pinów dla joystick'a (PA0:3 i PA5) oraz pin dla diody LED (PB2).

2.5 SPI

Akcelerometr, który będzie podstawą projektu, służy do pomiaru przyspieszenia na którym będzie bazowało kryterium stwierdzenia kradzieży.

Parametr	Wartość
Frame Format	Motorola
Data Size	8 Bits *
First Bit	MSB First
Prescaler (for Baud Rate)	16 *
Baud Rate	5.0 MBits/s *
Clock Polarity (CPOL)	Low
Clock Phase (CPHA)	1 Edge
CRC Calculation	Disabled
NSSP Mode	Enabled
NSS Signal Type	Software

2.6 SAI

Interfejs do obsługi wyjścia audio. Na jego podstawie będzie wysyłany komunikat o rzekomej kradzieży.

Parametr	Wartość
SAI A	
Synchronization Inputs	Asynchronous
Audio Mode	Master Transmit
Output Mode	Stereo
Companding Mode	No companding mode
SAI SD Line Output Mode	Driven
Protocol	I2S Standard
Data Size	16 Bits
Number of Slots (only Even Values)	2
Master Clock Divider	Enabled
Audio Frequency	192 KHz
Real Audio Frequency	35.714 KHz *
Error between Selected	-81.39 % *
Fifo Threshold	Empty
Output Drive	Disabled
SAI B	
Synchronization Inputs	Asynchronous
Protocol	SPDIF
Audio Mode	Master Transmit
Output Mode	Stereo
Companding Mode	No companding mode
Audio Frequency	48 KHz
Real Audio Frequency	142.857 KHz *
Fifo Threshold	Empty
Output Drive	Disabled

3 Urządzenia zewnętrzne

3.1 Akcelerometr – LSM303C

Akcelerometr został wykorzystany do pomiaru przyspieszenia liniowego na tej podstawie określano czy załączyć alarm.

Rejestr	Wartość
CTRL_REG4_A (0x23)	0xFD
CTRL_REG1_A (0x20)	0x4F

4 Opis działania programu

5 Założenia projektowe

1. Projekt będzie wykonywany w oparciu o płytkę STM32L476 Discovery wypożyczoną od prowadzącego kurs
2. Pomiar przyspieszenia będzie odbywał się przez wbudowany moduł z akcelerometrem
3. W przypadku wykrycia alarmu urządzenie podejmie określone kroki.
4. Menu sterowane z joystick’iem zapewni możliwość wybór ustawień sygnalizacji alarmu(dioda oraz głośnik).
5. Zbieranie danych o alarmach i przechowywanie w pamięci Flash.
6. Wykorzystanie zegara RTC do umiejscowienia zdarzenia alarmu w jego lokalnym czasie.
7. Badania dotyczące wykrywania progów alarmu i sprawdzenie funkcjonalności zaprojektowanego urządzenia.

6 Harmonogram pracy

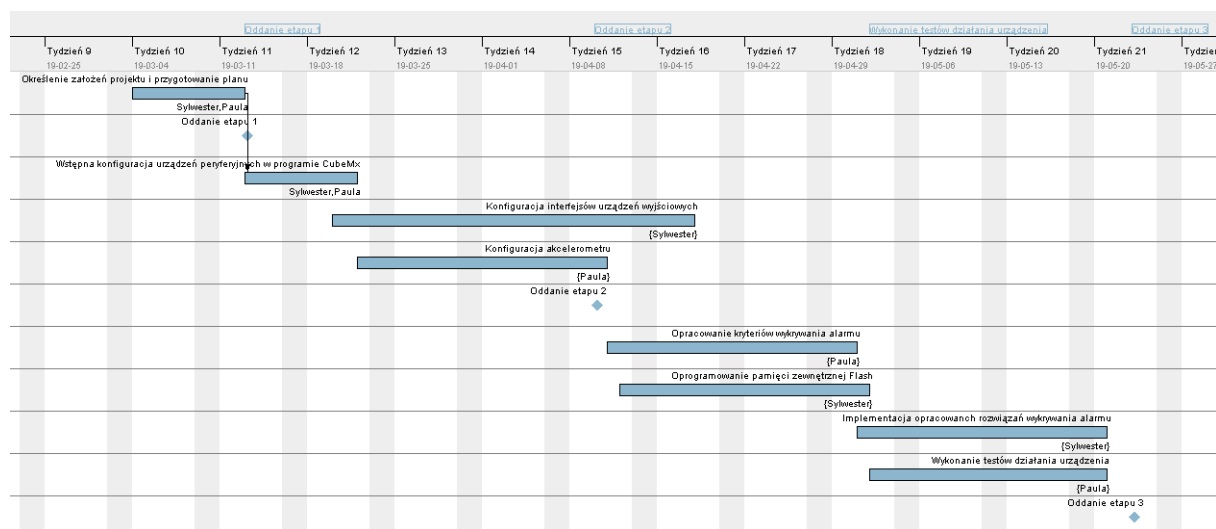
6.1 Zakres prac

Zapoznanie się z mikrokontrolerem, konfiguracja peryferiów. Implementacja obsługi zarówno pamięci Flash, jak i akcelerometru. Skonfigurowanie zegara RTC w celu późniejszej implementacji przekazywania godziny nieplanowanego ruchu. Zapoznanie z literaturą i poruszonym problemem. Przeprowadzenie badań na temat progów i optymalizacji działania urządzenia.

6.2 Kamienie milowe

1. Oddanie I etapu projektu. Projekt powinien zawierać założenia oraz plan co będzie podstawą do rozpoczęcia prac.
2. Oddanie II etapu projektu. Konfiguracja peryferiów powinna być już sfinalizowana, a przynajmniej na etapie pozwalającym rozpoczęcie kolejnego etapu związanego z badaniem przyspieszeń które powinny aktywować alarm.
3. Oddanie III etapu gdzie projekt powinien być już kompletny. Wg planu projekt powinien zakończyć się tydzień przed ostatecznym terminem złożenia pracy u prowadzącego.

6.3 Diagram Gantta



Rysunek 3: Diagram Gantta

6.4 Podział pracy

Oboje uczestnicy projektu zajmą się wstępną konfiguracją peryferiów odbywającą się za pomocą programu CubeMx. Zostanie zaimplementowana obsługa pamięci Flash, a także akcelerometru. W tym czasie opracowany zostanie również sposób przechowywania danych na zewnętrznej pamięci Flash. Projekt menu ma zakładać możliwość wyboru sygnału uruchamiającego alarm.

Sylwester Kozieja	%	Paula Langkafel	%
Wstępna konfiguracja peryferiów w programie CubeMx		Wstępna konfiguracja peryferiów w programie CubeMx	
Wstępny projekt menu		Implementacja obsługi akcelerometru	
Konfiguracja zegara RTC			

Tabela 3: Podział pracy – Etap II

Sylwester Kozieja	%	Paula Langkafel	%
Oprogramowanie zewnętrznej pamięci Flash oraz opracowanie sposobu przechowywania danych		Opracowanie kryteriów wykrywania alarmu	
Implementacja opracowanych rozwiązań wykrywania alarmu		Wykonanie testów urządzenia	
Sygnalizacja audiowizualna za pomocą Audio DAC oraz diody LED			

Tabela 4: Podział pracy – Etap III

7 Podsumowanie etapu II

7.1 Akcelerometr

Z niewiadomych przyczyn akcelerometr nie działa, na wyjściu mamy do czynienia z wartością 255, czyli maksimum zakresu. Próby oprogramowania były zgodne z dokumentacją, komunikacja odbywa się

w trybie half duplex master. Piny dotyczące innych peryferiów w pewien sposób związanych z używanymi zostały wprowadzone w stan wysoki.

7.2 Wyświetlacz z menu i zegarem RTC

Pierwszy commit miał na celu stworzenie pierwszego prostego programu, który sprawdzał komunikację z diodami LED oraz wyświetlaczem. W drugim commicie zostało zaimplementowane menu wraz z podstawowymi opcjami i zegar RTC będący podstawą do obliczania czasu co jest niezbędne do umiejscowienia alarmu w czasie. Urządzenie inicjowane jest stałą datą i godziną, ale istnieje możliwość ustawienia czasu w menu. Również w menu można ustawić flagi aktywności sygnalizacji alarmu dźwiękiem i diodą.

7.3 Opis menu

Opcje menu:

- Start - uruchamia procedurę rejestrowania alarmów
- Ustaw godzinę
- Ustaw datę
- Audio - ustawienie sygnalizacji alarmu przez sygnał Audio
- LED -ustawienie sygnalizacji alarmu diodą LED
- Wyświetl alarmy - pobiera z pamięci zarejestrowane alarmy i wyświetla w postaci listy

Literatura

- [1] *User manual - Getting started with STM32L476G discovery kit software development tools*, Sierpień 2015.
- [2] *UM1928 User manual - Getting started with STM32L476G discovery kit software development tools*, Wrzesień 2018.
- [3] W. Dowski. Sterowniki robotów, Laboratorium – Wprowadzenie, Wykorzystanie narzędzi STM32CubeMX oraz SW4STM32 do budowy programu mrugającej diody z obsługą przycisku. Marzec 2017.
- [4] Marius Bazu, Lucian Galateanu, Virgil Emil Ilian, Jerome Loicq, Serge Habraken, Jean-Paul Collette. Quantitative accelerated life testing of mems accelerometers. *Sensors*, Listopad 2007.
- [5] J. L. Suryadiputra Liawatimena. Vehicle Tracker with a GPS and Accelerometer Sensor System in Jakarta. *Internetworking Indonesia Journal*, Styczeń 2017.