

---

# RAPORT

## PROJEKT MODUŁU ANTYKRADZIEŻOWEGO

---

PROJEKT REALIZOWANY W RAMACH KURSU STEROWNIKI  
ROBOTÓW NA POLITECHNICE WROCŁAWSKIEJ

**Nazwa kursu:** STEROWNIKI ROBOTÓW  
**Nazwa projektu:** MODUŁ ANTYKRADZIEŻOWY SMS  
**Akronim projektu:** MASS  
**Termin zajęć:** ŚRODA TP 11:15  
**Autor 0:** MACIEJ FLIS 218543  
**Autor 1:** KRZYSZTOF DĄBEK 218549  
**Numer grupy:** 5  
**Prowadzący:** MGR INŻ WOJCIECH DOMSKI

WEDNESDAY 17<sup>TH</sup> MAY, 2017

## 1 Główne założenia projektowe

- Stworzenie software'owego projektu modułu antykradzieżowego na gotowej płytce, zaprojektowanej przez Macieja Flisa, opartej na mikrokontrolerze STM32F746VGT6.
- Napisanie programu z użyciem biblioteki HAL generowanej przez dedykowane IDE od firmy ST (STM32 toolchain: STM32CubeMX, Atollic TruStudio).
- Wgrywanie programu na płytkę za pomocą konwertera USB → UART.
- Wykorzystanie modułu GPS do uzyskania lokalizacji płytki.
- Wykorzystanie akcelerometru do uruchamiania wszystkich modułów w momencie poruszenia płytką.
- Przesyłanie informacji na żądanie w wiadomości SMS z użyciem modułu GSM.
- Emulacja pamięci EPROM w pamięci flash za pomocą bufora cyklicznego i zapisywanie w niej kolejnych lokalizacji modułu (po każdym przejściu w stan czuwania oraz co ustalony czas w stanie aktywnym).
- Uruchomienie zegara czasu rzeczywistego (RTC) i zapisywanie informacji o czasie przeprowadzenia pomiaru wraz z lokalizacją w symulowanej pamięci EPROM w formie logu.
- Wysyłanie danych z logu do komputera przez port USB.

## 2 Harmonogram

- (20.03. - 26.03.) Konfiguracja peryferiów mikrokontrolera, wygenerowanie kodu, wgranie programu na płytkę przez UART.
- (27.03. - 2.04.) Oprogramowanie komunikacji USB i akcelerometru.
- (3.04. - 23.04.) Uruchomienie modułu GSM, uruchomienie modułu GPS.
- (24.04. - 7.05.) Oprogramowanie RTC, emulacja pamięci EPROM.
- (8.05. - 14.05.) Juwenalia :)
- (15.05. - 26.05.) Przygotowanie raportu, przedstawienie pierwszego etapu prac nad projektem.

- (27.05. - 11.06) Przygotowanie gotowej aplikacji, testy, rozwiązywanie problemów.
- (12.06. - 22.06) Przygotowanie dokumentacji technicznej, przedstawienie gotowego projektu.

### 3 Opis zrealizowanych prac

Zadania zrealizowane na płytce stworzonej jako moduł antykradzieżowy opartej o mikrokontroler STM32F746VGT6:

- Konfiguracja mikrokontrolera i peryferiów.
- Wgrywanie kodu na płytkę przez UART oraz SWD, debuggowanie.
- Implementacja komunikacji przez USB.
- Obsługa akcelerometru FXOS8700CQ.
- Próby nawiązania połączenia z urządzeniem GSM Quectel M12 (na płytce).
- Próby nawiązania połączenia z urządzeniem GPS Quectel L70 (na płytce).
- Próby nawiązania połączenia z zewnętrznym modułem GSM.
- Przesyłanie danych przez konwerter UART  $\rightarrow$  USB.

Okazało się, że z powodu błędów projektowych związanych z hardwarem płytki, trzeba było dokonać zmiany koncepcji i używanych urządzeń. Zdecydowano się na:

- Płytkę rozwojową Kamami ZL41ARMF4 z mikrokontrolerem STM32F417VGT6.
- Moduł z akcelerometrem MPU6050.
- Moduł GSM, GPS Waveshare Sim808.

Zadania zrealizowane na modułach oraz płytce rozwojowej ZL41ARMF4:

- Pobieranie danych z GPS, komendy wysyłane z terminala komputera.
- Wysyłanie SMS przez GSM, komendy wysyłane z terminala komputera.
- Oprogramowanie akcelerometru MPU6050 (nietestowane).

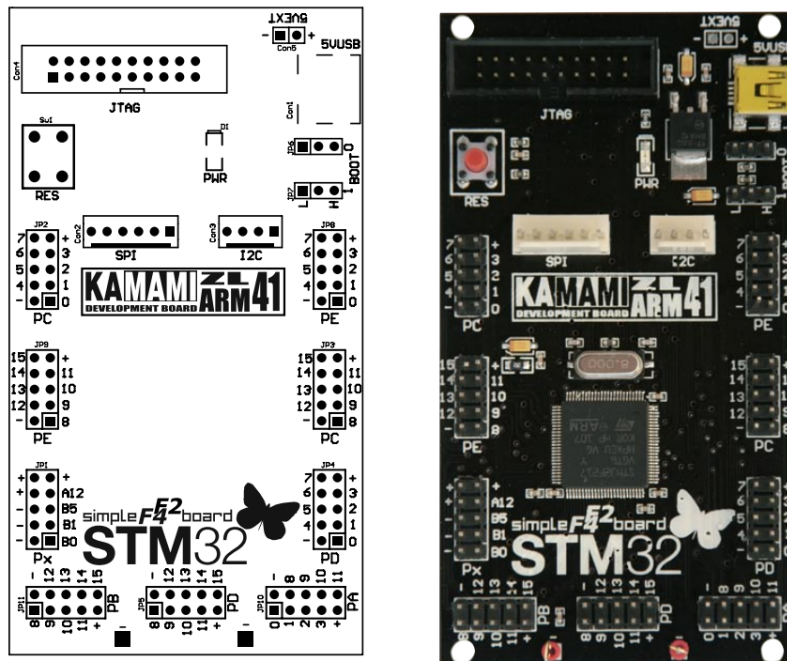
- Implementacja komunikacji przez USB.

Wykorzystany moduł Waveshare Sim808 prawdopodobnie został uszkodzony i przestał odpowiadać na komendy "AT", co uniemożliwiło dalsze prace związane z projektem oraz wykonanie wszystkich zadań zgodnie z harmonogramem.

W planach na najbliższe tygodnie jest zakupienie nowego modułu (modułów) GPS, GSM oraz nawiązanie z nim komunikacji i oprogramowanie tego na mikrokontrolerze. Następnie sprawdzenie i poprawienie obsługi akcelerometru, implementacja RTC oraz emulowanej pamięci EEPROM.

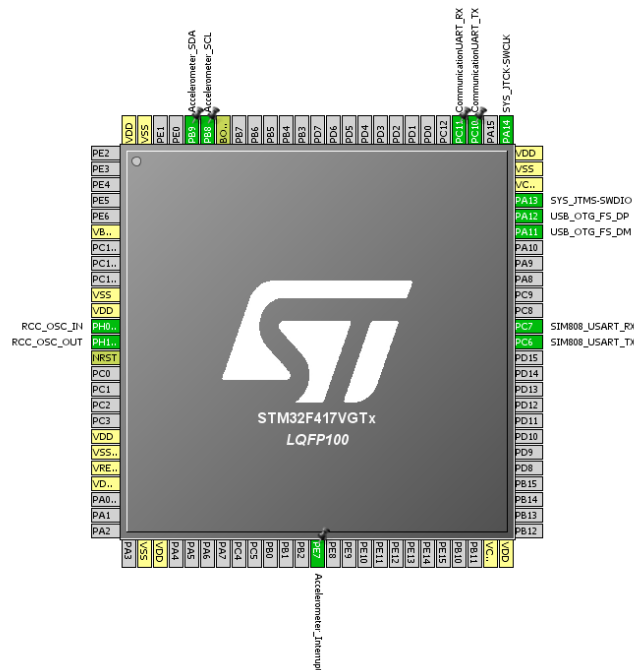
## 4 Konfiguracja mikrokontrolera

W projekcie wykorzystywana jest płytką rozwojową ZL41ARMF4 z mikrokontrolerem STM32F417VGT6. Na płytce wyprowadzone są wszystkie piny mikrokontrolera, może być zasilana przez port miniUSB. Wygląd płytki został przedstawiony na rysunku 1.



- UART (PC6 Tx, PC7 Rx) dla modułu SIM808 (GSM oraz GPS).
- UART (PC10 Tx, PC11 Rx) dla komunikacji z komputerem przez zewnętrzny konwerter UART→ USB.
- USB (PA11 otg\_fs\_dm, PA12 otg\_fs\_dp) w trybie device do komunikacji przez Virtual COM Port.
- RCC (PH0 osc\_in, PH1 osc\_out) zewnętrzny oscylator 8MHz dla US.
- SWD (PA13 swdio, PA14 swclk) szeregowy port do debuggowania.

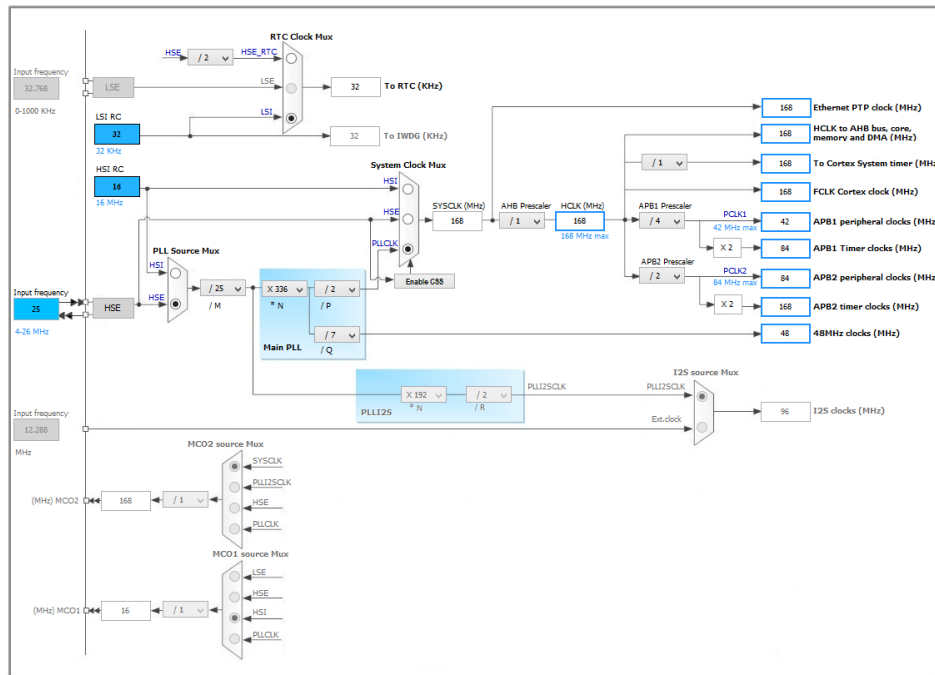
Konfiguracja pinów mikrokontrolera została przedstawiona na rysunku 2.



Rysunek 2: Konfiguracja mikrokontrolera (pinout)

Konfiguracja zegara mikrokontrolera zakłada maksymalne możliwe taktowanie timerów oraz taktowanie 48MHz z wykorzystaniem HSE (oscylatora 8MHz) do obsługi USB.

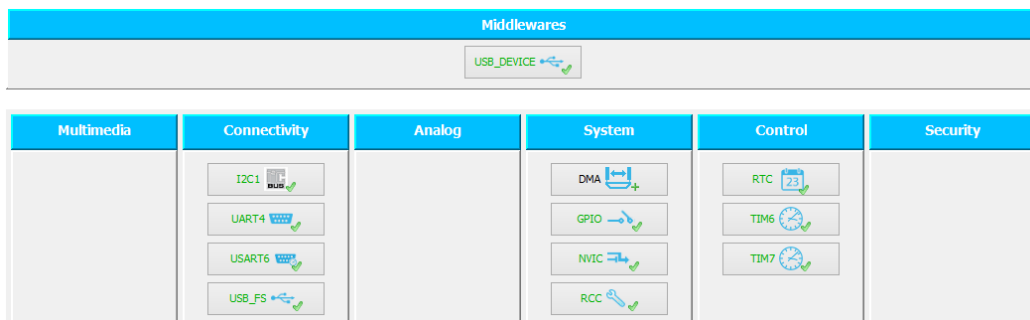
Konfiguracja zegara mikrokontrolera została przedstawiona na rysunku 3



Rysunek 3: Konfiguracja mikrokontrolera (zegar)

## 5 Konfiguracja peryferiów

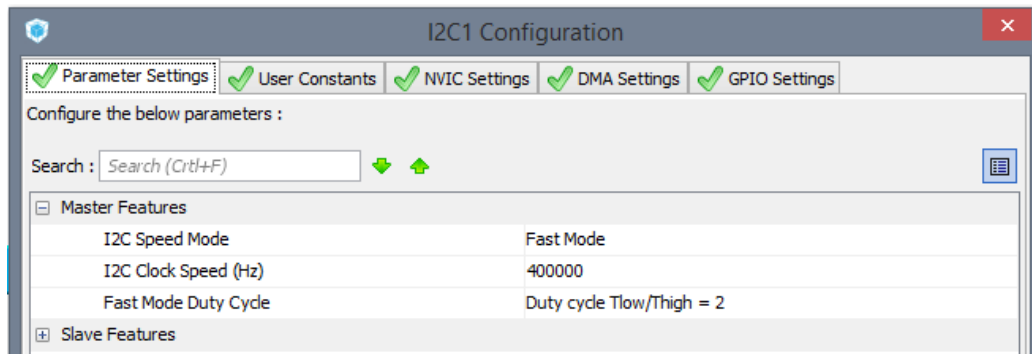
Konfiguracja peryferiów (pokazanych na rysunku 4) została dostosowana do wymagań układów zewnętrznych.



Rysunek 4: Konfiguracja peryferiów

## 5.1 I2C

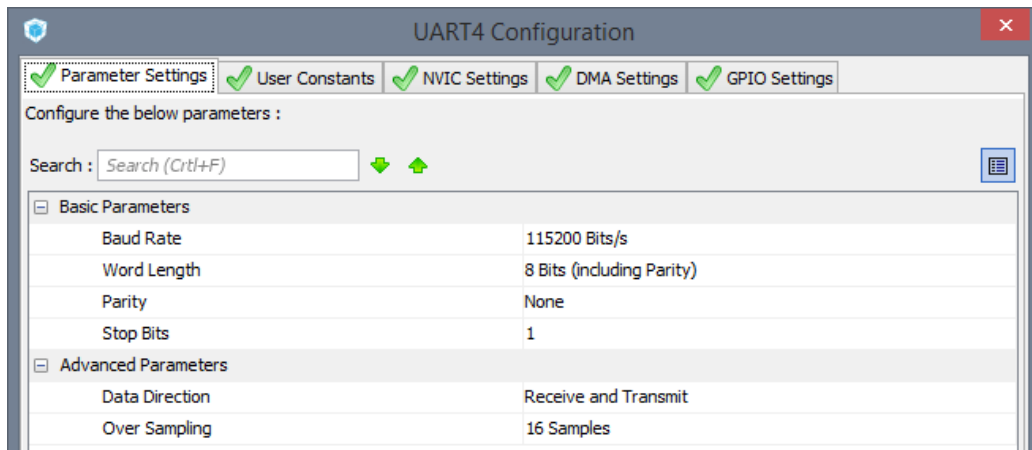
Konfiguracja I2C została przedstawiona na rysunku 5



Rysunek 5: Konfiguracja I2C

## 5.2 UART

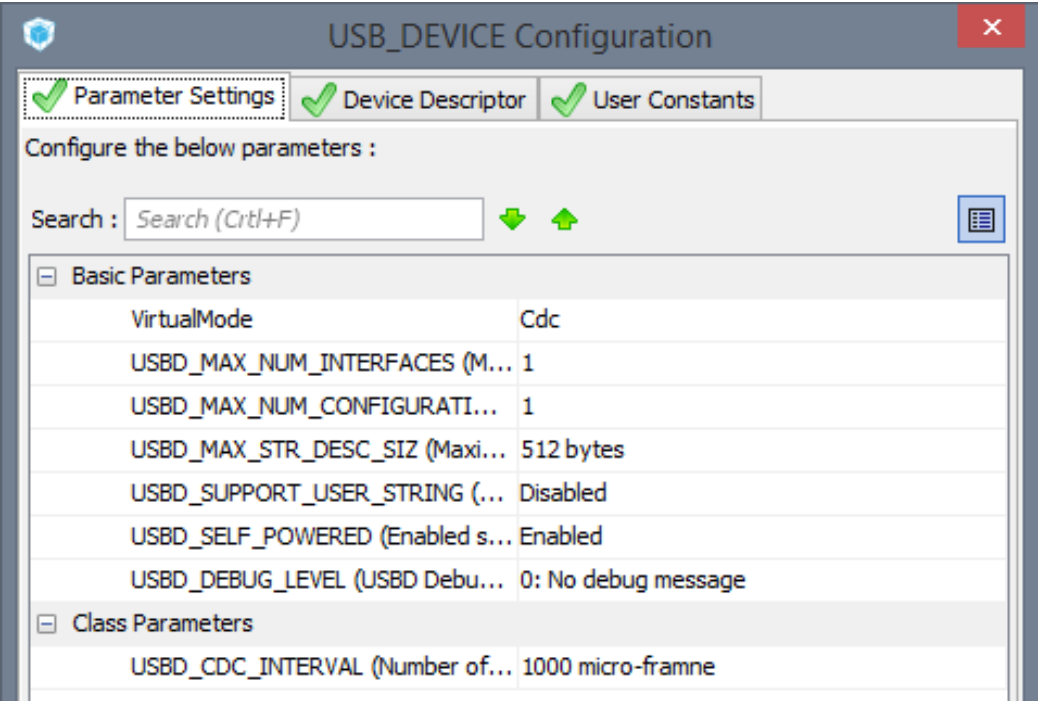
Konfiguracja UART została przedstawiona na rysunku 6. Oba peryferia UART zostały skonfigurowane jednakowo



Rysunek 6: Konfiguracja UART

5.3 USB

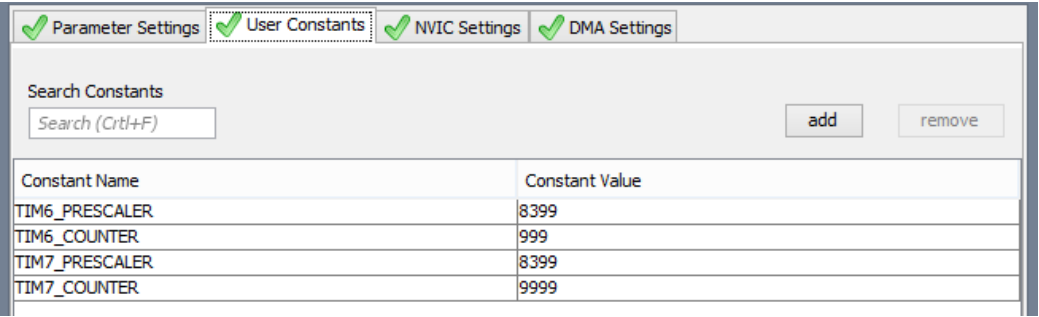
Konfiguracja USB wraz z Middleware CDC została przedstawiona na rysunku 7



Rysunek 7: Konfiguracja USB

5.4 Timery

Konfiguracja timerów została przedstawiona na rysunku 8

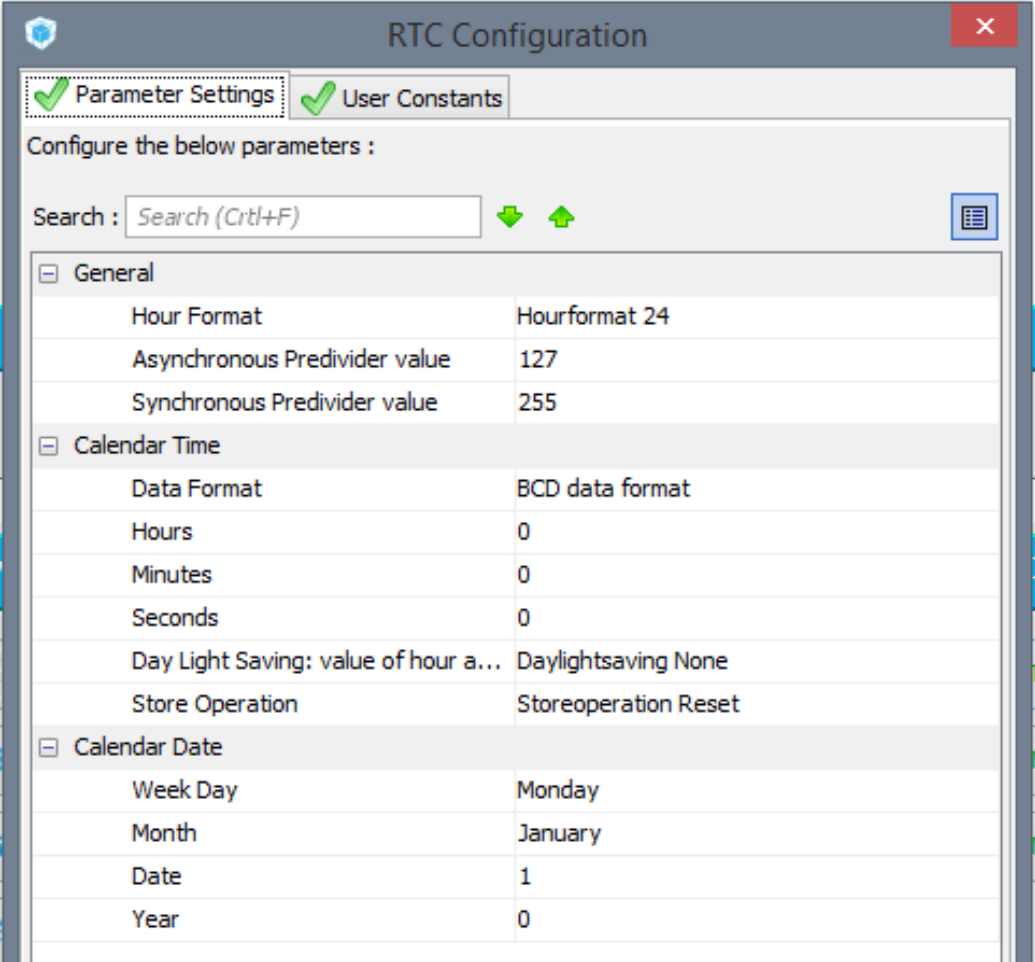


Rysunek 8: Konfiguracja timerów



## 5.5 RTC

Konfiguracja RTC została przedstawiona na rysunku 9



The screenshot shows the 'RTC Configuration' window with two tabs: 'Parameter Settings' (active) and 'User Constants'. Below the tabs, it says 'Configure the below parameters :'. There is a search bar with the text 'Search (Ctrl+F)' and two green arrow buttons. The configuration is organized into three expandable sections: 'General', 'Calendar Time', and 'Calendar Date'. Each section contains a table of parameters and their values.

General	
Hour Format	Hourformat 24
Asynchronous Predivider value	127
Synchronous Predivider value	255

Calendar Time	
Data Format	BCD data format
Hours	0
Minutes	0
Seconds	0
Day Light Saving: value of hour a...	Daylightsaving None
Store Operation	Storeoperation Reset

Calendar Date	
Week Day	Monday
Month	January
Date	1
Year	0

Rysunek 9: Konfiguracja RTC

## 5.6 Przerwania (NVIC)

Konfiguracja NVIC została przedstawiona na rysunku 10

**NVIC Configuration**

☒ NVIC ☒ Code generation

Priority Group: 4 bits for pre-emption priority 0 bits for subpriority ☐ Sort by Preemption Priority and Sub Priority

Search:    ☐ Show only enabled interrupts

Interrupt Table	Enabled	Preemption Priority	Sub Priority
Non maskable interrupt	<input checked="" type="checkbox"/>	0	0
Hard fault interrupt	<input checked="" type="checkbox"/>	0	0
Memory management fault	<input checked="" type="checkbox"/>	0	0
Pre-fetch fault, memory access fault	<input checked="" type="checkbox"/>	0	0
Undefined instruction or illegal state	<input checked="" type="checkbox"/>	0	0
System service call via SWI instruction	<input checked="" type="checkbox"/>	0	0
Debug monitor	<input checked="" type="checkbox"/>	0	0
Pendable request for system service	<input checked="" type="checkbox"/>	0	0
Time base: System tick timer	<input checked="" type="checkbox"/>	0	0
PVD interrupt through EXTI line 16	<input type="checkbox"/>	0	0
Flash global interrupt	<input type="checkbox"/>	0	0
RCC global interrupt	<input type="checkbox"/>	0	0
EXTI line[9:5] interrupts	<input checked="" type="checkbox"/>	0	0
I2C1 event interrupt	<input type="checkbox"/>	0	0
I2C1 error interrupt	<input type="checkbox"/>	0	0
UART4 global interrupt	<input checked="" type="checkbox"/>	0	0
TIM6 global interrupt, DAC1 and DAC2 underrun error interrupts	<input checked="" type="checkbox"/>	0	0
TIM7 global interrupt	<input checked="" type="checkbox"/>	0	0
USB On The Go FS global interrupt	<input checked="" type="checkbox"/>	0	0
USART6 global interrupt	<input checked="" type="checkbox"/>	0	0
FPU global interrupt	<input type="checkbox"/>	0	0

☐ Enabled Preemption Priority  Sub Priority

Rysunek 10: Konfiguracja NVIC

## 6 Opis układów zewnętrznych

### 6.1 Akcelerometr

Jako akcelerometr został wykorzystany moduł MPU6050 (rysunek 11), komunikujący się po I2C z częstotliwością 400kHz oraz generujący przerwania np. wykrywania ruchu urządzenia (Motion Interrupt). Akcelerometr jest trzyosiowy i jest w stanie mierzyć przyspieszenia o maksymalnej wartości  $\pm 16g$ . Bity konfiguracyjne ustawiane w rejestrach akcelerometru zostały pokazane na rysunku 12.



Rysunek 11: Moduł MPU6050 z akcelerometrem

```

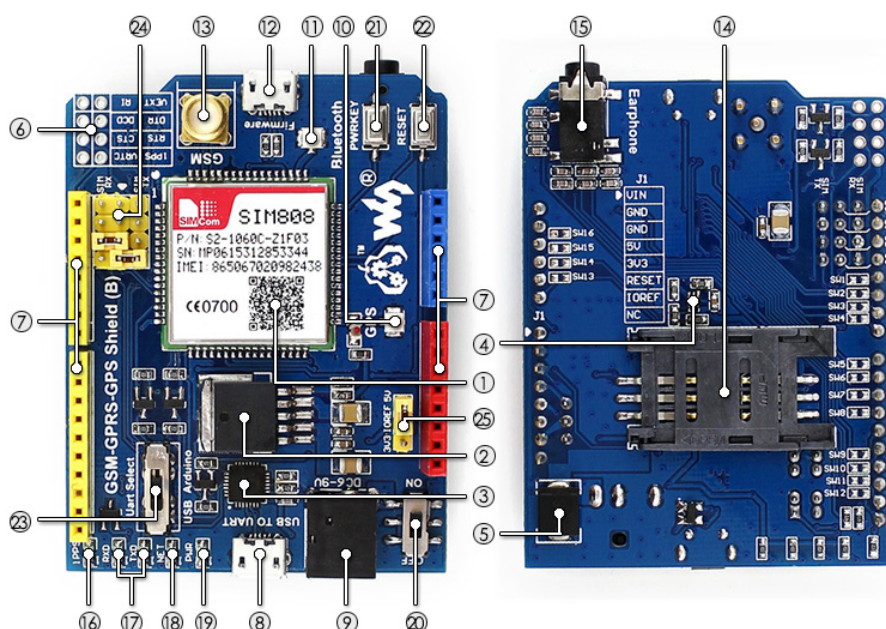
14@/*
15 * No Reset, Sleep Mode off, CycleMode on, x,
16 * Temperature Sensor disabled, Clock Source internal
17 */
18 uint8_t q_MPU6050_PowerManagement1Settings[1] = { 0b00101000 };
19@/*
20 * Wake up 20 Hz,
21 * Accelerometer Axis Stand-by Mode: X off, Y off, Z off
22 * Gyroscope Axis Stand-by Mode: X off, Y off, Z off
23 */
24 uint8_t q_MPU6050_PowerManagement2Settings[1] = { 0b10000000 };
25
26@/*
27 * Self Axis Test: X disabled, Y disabled, Z disabled
28 * Full scale range for Accelerometer: +/- 16g
29 */
30 uint8_t q_MPU6050_AccelerometerSettings[1] = { 0b00011000 };
31
32@/*
33 * Active high, push-pull, pulse mode off,
34 * Clear on any read,
35 */
36 uint8_t q_MPU6050_InterruptSettings[1] = { 0b00110000 };
37
38@/*
39 * Motion Interrupt enabled, other disabled
40 */
41 uint8_t q_MPU6050_InterruptEnableSettings[1] = { 0b01000000 };
42
43 uint8_t q_MPU6050_MotionInterruptThreshold[1] = { ACC_MOTION_THRESHOLD_mg / ACC_MOTION_THRESHOLD_RESOLUTION_mg };
44
45 uint8_t q_MPU6050_MotionInterruptDuration[1] = { ACC_MOTION_DURATION_ms / ACC_MOTION_DURATION_RESOLUTION_ms };

```

Rysunek 12: Konfiguracja Akcelerometru

## 6.2 GSM i GPS

W projekcie wykorzystano moduł Waveshare sim808 (rysunek 13), z którym można komunikować się za pomocą interfejsu UART oraz przez USB, co zapewnia wbudowany konwerter USB  $\rightarrow$  UART. Moduł może być zasilany z zewnętrznego zasilacza, posiada wbudowaną antenę GSM, slot na kartę SIM, oraz możliwość podłączenia anteny GPS. Komunikacja z GSM, a także GPS odbywa się za pomocą komend "AT". Weryfikacja poprawności działania po wysłanych komendach odbywa się za pomocą wiadomości zwracanych przez urządzenie ("OK" oraz "ERROR").



Rysunek 13: Moduł Waveshare Sim808

### Komendy komunikacji z GSM:

- **AT** – Sprawdzenie poprawności działania urządzenia.
- **AT+CLCK="SC",1,"pin"** – Odblokowanie karty SIM.
- **AT+CPIN="pin"** – odblokowanie karty SIM.
- **AT+CMGF=1** – Podstawowa konfiguracja SMS.

- **AT+CMGS="phone number"** – Ustawienie numeru telefonu, na który wysłać wiadomość oraz wysłanie wiadomości.
- **0x1A** – Terminator wiadomości. Znak kończący wiadomość SMS.

Komendy komunikacji z GPS:

- **AT+GSV** – Sprawdzenie poprawności działania urządzenia.
- **AT+CGNSPWR=1** – Uruchomienie urządzenia GPS.
- **AT+CGNSTST=1** – Reset GPS. Po tej komendzie GPS zwraca dane co sekundę.
- **AT+CGNSINF** – Pobranie informacji o aktualnej pozycji z GPS.
- **AT+CGPSSTATUS** – Status GPS.

## 7 Opis działania programu

1. Inicjalizacja Akcelerometru (wpisanie do rejestrów urządzenia ustawień ogólnych, poboru mocy oraz przerwania wykrywających ruch).
2. Inicjalizacja GSM (sprawdzenie poprawności działania urządzenia, odblokowanie karty SIM).
3. Inicjalizacja GPS (sprawdzenie poprawności działania urządzenia, wyłączenie urządzenia).
4. Oczekiwanie na przerwanie:
  - Odebranie komendy "log" lub "LOG" przez virtualny port COM USB.
    - (a) Wysłanie wszystkich pozycji zapisanych w emulowanej pamięci EEPROM przez USB.
  - Przerwanie od akcelerometru (wykrycie ruchu urządzenia)
    - (a) Uruchomienie GPS
    - (b) Cykliczne czytanie pozycji urządzenia z GPS i zapisywanie w emulowanej pamięci EEPROM.
    - (c) Uruchomienie timera TIM7, który co określony czas wysyła dzięki GSM, SMS z aktualną pozycją urządzenia.

## 8 Zadania niezrealizowane

- Emulacja pamięci EEPROM w pamięci flash mikrokontrolera.
- Oprogramowanie i korzystanie z RTC.
- Stworzenie struktury w pamięci EEPROM przechowującej historię lokalizacji oraz czas z RTC.
- Pobieranie danych o lokalizacji z modułu GPS.
- Działający program modułu antykradzieżowego.

## Spis treści

<b>1</b>	<b>Główne założenia projektowe</b>	<b>1</b>
<b>2</b>	<b>Harmonogram</b>	<b>1</b>
<b>3</b>	<b>Opis zrealizowanych prac</b>	<b>2</b>
<b>4</b>	<b>Konfiguracja mikrokontrolera</b>	<b>3</b>
<b>5</b>	<b>Konfiguracja peryferiów</b>	<b>5</b>
5.1	I2C . . . . .	6
5.2	UART . . . . .	6
5.3	USB . . . . .	7
5.4	Timery . . . . .	7
5.5	RTC . . . . .	8
5.6	Przerwania (NVIC) . . . . .	9
<b>6</b>	<b>Opis układów zewnętrznych</b>	<b>10</b>
6.1	Akcelerometr . . . . .	10
6.2	GSM i GPS . . . . .	11
<b>7</b>	<b>Opis działania programu</b>	<b>12</b>
<b>8</b>	<b>Zadania niezrealizowane</b>	<b>13</b>
<b>9</b>	<b>Bibliografia</b>	<b>14</b>

## 9 Bibliografia

- <https://edu.domski.pl/kursy/sterowniki-robotow/sr-projekt/>
- G510-GSM-GPRS-command.pdf – Spis i wyjaśnienie komend "AT"
- G510-GSM-GPRS-manual.pdf – Dokumentacja modułu GSM, GPS
- quectel-l70.pdf – Dokumentacja urządzenia GPS Quectel L70
- quectel-m12.pdf – Dokumentacja urządzenia GSM Quectel M12
- STM32F4ReferenceManual – Dokumentacja mikrokontrolera STM32F417VGT6
- STM32F7ReferenceManual – Dokumentacja mikrokontrolera STM32F746VGT6
- FXOS8700CQManual – Dokumentacja akcelerometru FXOS8700CQ
- MPU6050Manual – Dokumentacja modułu akcelerometru MPU6050
- MPU6050RegisterMap – Spis rejestrów modułu akcelerometru MPU6050
- <https://botland.com.pl/arduino-shield-komunikacja/5623-waveshare-gsmgprsgps-sim808-shield-nakladka-na-arduino.html> – Opis i materiały do modułu Waveshare Sim808
- [http://www.waveshare.com/wiki/GSM/GPRS/GPS\\_Shield\\_%28B%29](http://www.waveshare.com/wiki/GSM/GPRS/GPS_Shield_%28B%29) – Przewodnik użytkownika modułu Waveshare Sim808