

PROJEKT

STEROWNIKI ROBOTÓW

---

# Raport

## ModułAntykradzieżowy SMS

## MASS

---

*Skład grupy (5):*

Krzysztof DĄBEK, 218549

Maciej FLIS, 218543

*Termin:* srTP11

*Prowadzący:*

mgr inż. Wojciech DOMSKI

17 maja 2017

# Spis treści

## 1 Główne założenia projektowe

## 2 Harmonogram

## 3 Opis zrealizowanych prac

## 4 Konfiguracja mikrokontrolera

## 5 Konfiguracja peryferiów

|     |                |  |
|-----|----------------|--|
| 5.1 | I2C . . . . .  |  |
| 5.2 | UART . . . . . |  |
| 5.3 | NVIC . . . . . |  |
| 5.4 | GPIO . . . . . |  |

## 6 Opis układów zewnętrznych

|     |                        |  |
|-----|------------------------|--|
| 6.1 | Akcelerometr . . . . . |  |
| 6.2 | GSM i GPS . . . . .    |  |

## 7 Opis działania programu

## 8 Zadania niezrealizowane

## 9 Bibliografia

# 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 → USB.

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

- 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.

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.

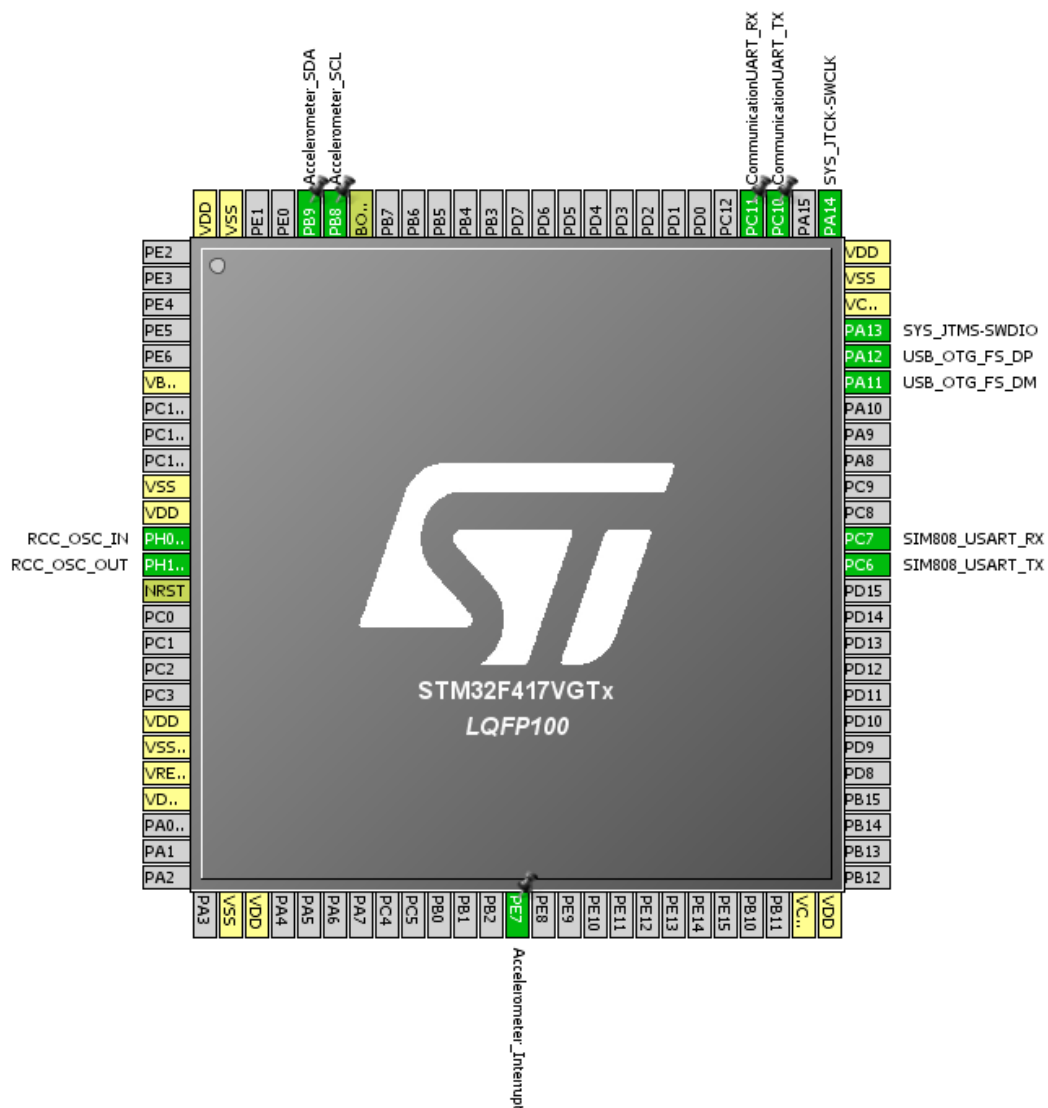
W projekcie wykorzystywana jest płytka rozwojowa ZL41ARMF4 z mikrokontrolerem STM32F417VGT6. Na płycie wyprowadzone są wszystkie piny mikrokontrolera, może być zasilana przez port miniUSB. Wygląd płytki został przedstawiony na rysunku 1.



W konfiguracji pinów mikrokontrolera przewidziano:

- I2C (PB8 SCL, PB9 SDA) oraz wejście zewnętrznego przerwania (PE7) dla modułu z akcelerometrem.
- 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 oraz w tabeli 1.

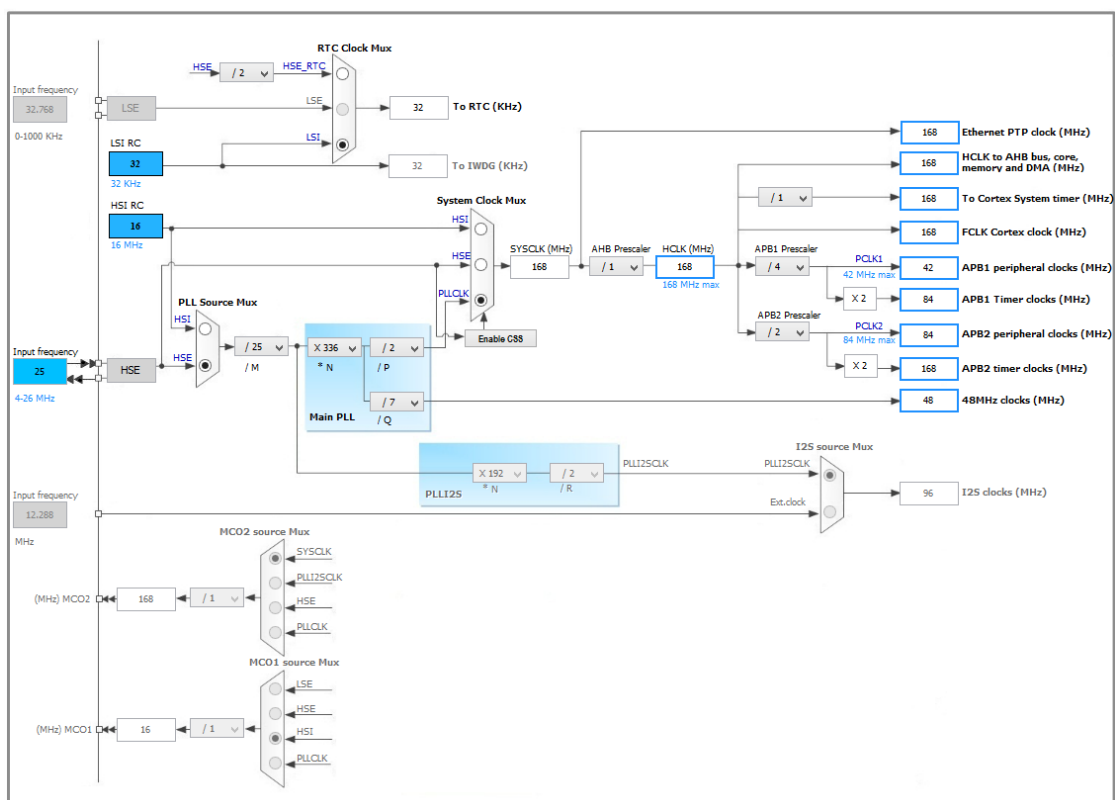


Rysunek 2: Konfiguracja mikrokontrolera (pinout)

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

Tabela 1: Konfiguracja pinów mikrokontrolera

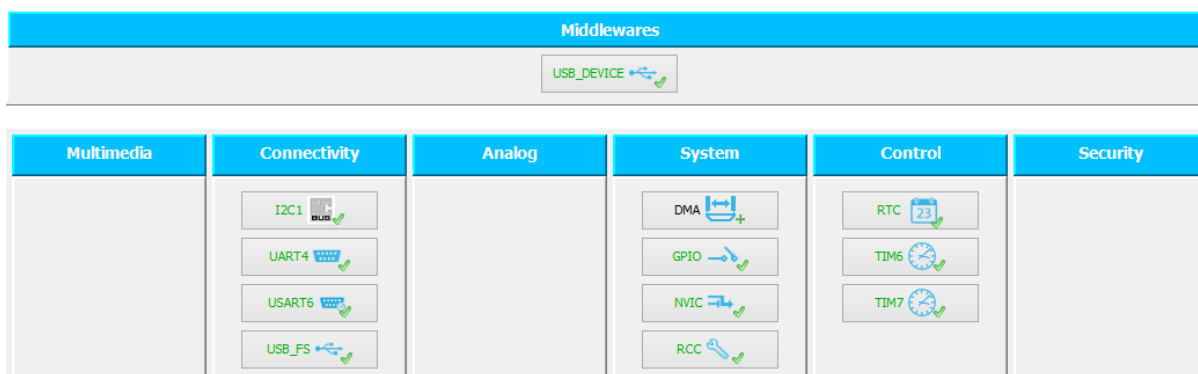
| Pin Number<br>LQFP100 | Pin Name<br>(function after<br>reset) | Pin Type | Alternate<br>Function(s) | Label                   |
|-----------------------|---------------------------------------|----------|--------------------------|-------------------------|
| 6                     | VBAT                                  | Power    |                          |                         |
| 10                    | VSS                                   | Power    |                          |                         |
| 11                    | VDD                                   | Power    |                          |                         |
| 12                    | PH0-OSC_IN                            | I/O      | RCC_OSC_IN               |                         |
| 13                    | PH1-OSC_OUT                           | I/O      | RCC_OSC_OUT              |                         |
| 14                    | NRST                                  | Reset    |                          |                         |
| 19                    | VDD                                   | Power    |                          |                         |
| 20                    | VSSA                                  | Power    |                          |                         |
| 21                    | VREF+                                 | Power    |                          |                         |
| 22                    | VDDA                                  | Power    |                          |                         |
| 27                    | VSS                                   | Power    |                          |                         |
| 28                    | VDD                                   | Power    |                          |                         |
| 38                    | PE7                                   | I/O      | GPIO_EXTI7               | Accelerometer_Interrupt |
| 49                    | VCAP_1                                | Power    |                          |                         |
| 50                    | VDD                                   | Power    |                          |                         |
| 63                    | PC6                                   | I/O      | USART6_TX                | SIM808_USART_TX         |
| 64                    | PC7                                   | I/O      | USART6_RX                | SIM808_USART_RX         |
| 70                    | PA11                                  | I/O      | USB_OTG_FS_DM            |                         |
| 71                    | PA12                                  | I/O      | USB_OTG_FS_DP            |                         |
| 72                    | PA13                                  | I/O      | SYS_JTMS-SWDIO           |                         |
| 73                    | VCAP_2                                | Power    |                          |                         |
| 74                    | VSS                                   | Power    |                          |                         |
| 75                    | VDD                                   | Power    |                          |                         |
| 76                    | PA14                                  | I/O      | SYS_JTCK-SWCLK           |                         |
| 78                    | PC10                                  | I/O      | UART4_TX                 | CommunicationUART_TX    |
| 79                    | PC11                                  | I/O      | UART4_RX                 | CommunicationUART_RX    |
| 94                    | BOOT0                                 | Boot     |                          |                         |
| 95                    | PB8                                   | I/O      | I2C1_SCL                 | Accelerometer_SCL       |
| 96                    | PB9                                   | I/O      | I2C1_SDA                 | Accelerometer_SDA       |
| 99                    | VSS                                   | Power    |                          |                         |
| 100                   | VDD                                   | Power    |                          |                         |



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 akcelerometru została przedstawiona w wyciągu z raportu (rysunek 5).

### 5.1. I2C1

#### I2C: I2C

##### 5.1.1. Parameter Settings:

###### Master Features:

I2C Speed Mode

I2C Clock Speed (Hz)

Fast Mode Duty Cycle

###### Fast Mode \*

400000

Duty cycle Tlow/Thigh = 2

Rysunek 5: Konfiguracja I2C

## 5.2 UART

Konfiguracja UART dla GSM, GPS oraz komunikacji z komputerem została przedstawiona w wyciągu z raportu (tabela 6).

### 5.7. UART4

#### Mode: Asynchronous

##### 5.7.1. Parameter Settings:

###### Basic Parameters:

Baud Rate

115200

Word Length

8 Bits (including Parity)

Parity

None

Stop Bits

1

###### Advanced Parameters:

Data Direction

Receive and Transmit

Over Sampling

16 Samples

Rysunek 6: Konfiguracja UART



## 5.3 NVIC

Konfiguracja przerwań NVIC została przedstawiona w wyciągu z raportu (tabela 2).

Tabela 2: Konfiguracja przerwań

| Interrupt Table  | Enable | Preenmption Priority | SubPriority |
|--|--------|----------------------|-------------|
| Non maskable interrupt   | true   | 0                    | 0           |
| Hard fault interrupt   | true   | 0                    | 0           |
| Memory management fault  | true   | 0                    | 0           |
| Pre-fetch fault, memory access fault                           | true   | 0                    | 0           |
| Undefined instruction or illegal state                         | true   | 0                    | 0           |
| System service call via SWI instruction                        | true   | 0                    | 0           |
| Debug monitor  | true   | 0                    | 0           |
| Pendable request for system service                            | true   | 0                    | 0           |
| System tick timer  | true   | 0                    | 0           |
| EXTI line[9:5] interrupts                                      | true   | 0                    | 0           |
| UART4 global interrupt   | true   | 0                    | 0           |
| TIM6 global interrupt, DAC1 and DAC2 underrun error interrupts | true   | 0                    | 0           |
| TIM7 global interrupt  | true   | 0                    | 0           |
| USB On The Go FS global interrupt                              | true   | 0                    | 0           |
| USART6 global interrupt  | true   | 0                    | 0           |
| PVD interrupt through EXTI line 16                             | unused |                      |             |
| Flash global interrupt   | unused |                      |             |
| RCC global interrupt   | unused |                      |             |
| I2C1 event interrupt   | unused |                      |             |
| I2C1 error interrupt   | unused |                      |             |
| FPU global interrupt   | unused |                      |             |

## 5.4 GPIO

Konfiguracja GPIO została przedstawiona w wyciągu z raportu (tabela 3).

Tabela 3: Konfiguracja GPIO

| IP         | Pin         | Signal         | GPIO mode  | GPIO pull/up pull down      | Max Speed   | User Label              |
|------------|-------------|----------------|--|-----------------------------|-------------|-------------------------|
| I2C1       | PB8         | I2C1_SCL       | Alternate Function Open Drain                              | Pull-up                     | Very High * | Accelerometer_SCL       |
|            | PB9         | I2C1_SDA       | Alternate Function Open Drain                              | Pull-up                     | Very High * | Accelerometer_SDA       |
| RCC        | PH0-OSC_IN  | RCC_OSC_IN     | n/a  | n/a                         | n/a         |                         |
|            | PH1-OSC_OUT | RCC_OSC_OUT    | n/a  | n/a                         | n/a         |                         |
| SYS        | PA13        | SYS_JTMS-SWDIO | n/a  | n/a                         | n/a         |                         |
|            | PA14        | SYS_JTCK-SWCLK | n/a  | n/a                         | n/a         |                         |
| UART4      | PC10        | UART4_TX       | Alternate Function Push Pull                               | Pull-up                     | Very High * | CommunicationUART_TX    |
|            | PC11        | UART4_RX       | Alternate Function Push Pull                               | Pull-up                     | Very High * | CommunicationUART_RX    |
| USART6     | PC6         | USART6_TX      | Alternate Function Push Pull                               | Pull-up                     | Very High * | SIM808_USART_TX         |
|            | PC7         | USART6_RX      | Alternate Function Push Pull                               | Pull-up                     | Very High * | SIM808_USART_RX         |
| USB_OTG_FS | PA11        | USB_OTG_FS_DM  | Alternate Function Push Pull                               | No pull-up and no pull-down | Very High * |                         |
|            | PA12        | USB_OTG_FS_DP  | Alternate Function Push Pull                               | No pull-up and no pull-down | Very High * |                         |
| GPIO       | PE7         | GPIO_EXTI7     | External Interrupt Mode with Rising edge trigger detection | No pull-up and no pull-down | n/a         | Accelerometer_Interrupt |

## 6 Opis układów zewnętrznych

### 6.1 Akcelerometr

Jako akcelerometr został wykorzystany moduł MPU6050 (rysunek 7), 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 +/- 16g. Bity



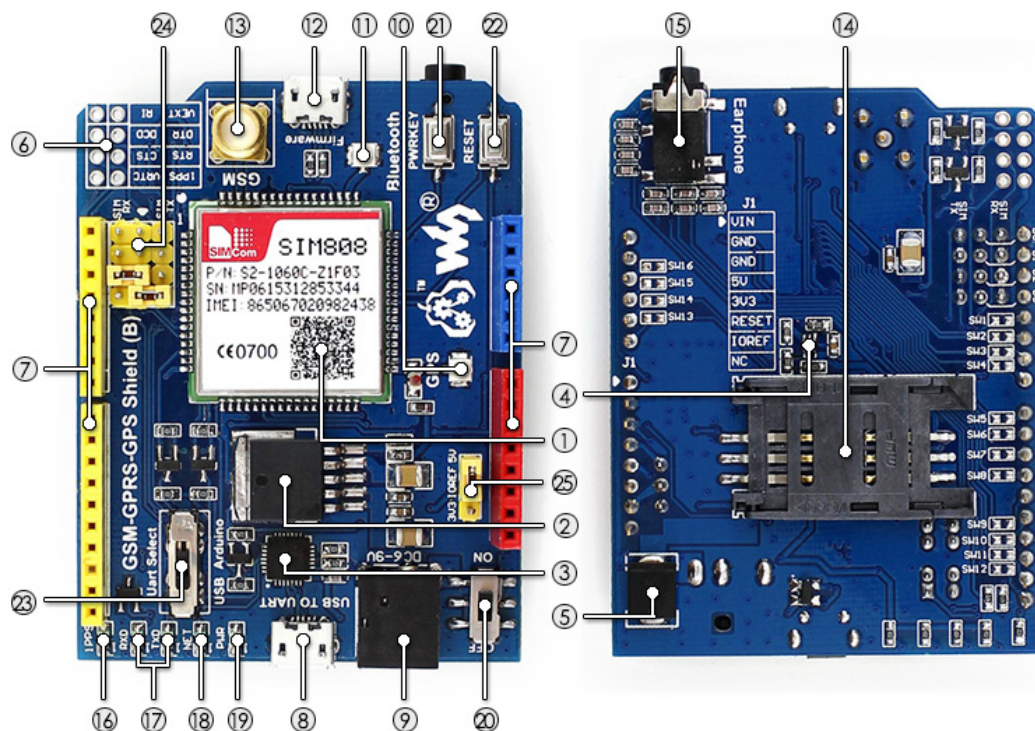
Rysunek 7: Moduł MPU6050 z akcelerometrem

konfiguracyjne ustawiane w rejestrach akcelerometru zostały pokazane poniżej.

```
1 /*
2  * No Reset , Sleep Mode off , CycleMode on , x ,
3  * Temperature Sensor disabled , Clock Source internal
4  */
5 uint8_t g_MPU6050_PowerManagement1Settings[1] = { 0b00101000 };
6 /*
7  * Wake up 20 Hz,
8  * Accelerometer Axis Stand-by Mode: X off , Y off , Z off
9  * Gyroscope Axis Stand-by Mode: X off , Y off , Z off
10 */
11 uint8_t g_MPU6050_PowerManagement2Settings[1] = { 0b10000000 };
12
13 /*
14  * Self Axis Test: X disabled , Y disabled , Z disabled
15  * Full scale range for Accelerometer: +/- 16g
16 */
17 uint8_t g_MPU6050_AccelerometerSettings[1] = { 0b00011000 };
18
19 /*
20  * Active high , push-pull , pulse mode off ,
21  * Clear on any read ,
22 */
23 uint8_t g_MPU6050_InterruptSettings[1] = { 0b00110000 };
24
25 /*
26  * Motion Interrupt enabled , other disabled
27 */
28 uint8_t g_MPU6050_InterruptEnableSettings[1] = { 0b01000000 };
29
30 uint8_t g_MPU6050_MotionInterruptThreshold[1] = { ACC_MOTION_THRESHOLD_mg /
31   ACC_MOTION_THRESHOLD_RESOLUTION_mg };
32
33 uint8_t g_MPU6050_MotionInterruptDuration[1] = { ACC_MOTION_DURATION_ms /
34   ACC_MOTION_DURATION_RESOLUTION_ms };
```

## 6.2 GSM i GPS

W projekcie wykorzystano moduł Waveshare sim808 (rysunek 8), 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 8: 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.

### 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 przerwai 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 historie lokalizacji oraz czas z RTC.
- Pobieranie danych o lokalizacji z modułu GPS.
- Działający program modułu antykradzieżowego.

## 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