

Aplikacja turystyczna #2

Sprawozdanie

Niedziela, 11:45

Skład zespołu:

Dominik Cegiełka (lider) 224478 224478@edu.p.lodz.pl

Kamil Zarych 224546 224546@edu.p.lodz.pl

Aktualnie wykorzystywany sprzęt: Smartfon Xiaomi Mi 5.

Zakres projektu:

Założona funkcjonalność	Stopień zaawansowania	Osoba odpowiedzialna
Kompas	Działa	Dominik Cegiełka
Odczyt współrzędnych geograficznych i wysokości	Działa	Kamil Zarych
Odczyt ciśnienia atmosferycznego	Działa	Kamil Zarych
Obsługa latarki	Działa	Dominik Cegiełka
Obsługa wibratora	Działa	Dominik Cegiełka
Zmiana motywu aplikacji	Działa	Dominik Cegiełka
Obsługa czytnika linii papilarnych	Działa	Kamil Zarych
Nadawanie sygnału SOS	Działa	Kamil Zarych

Wykorzystane funkcjonalności:

akcelerometr, magnetometr, barometr, GPS, UART, wibrator, czytnik linii papilarnych, czujnik światła, lampa błyskowa, interfejs I²C, interfejs SPI.

Spis treści

1. Dokumentacja użytkownika	3
1.1. Wymagania techniczne aplikacji	3
1.2. Instrukcja użytkownika	3
2. Opis algorytmów	5
2.1. Sygnał SOS	5
2.2. Motyw aplikacji	5
2.3. Ciśnienie atmosferyczne	6
2.4. Współrzędne geograficzne i wysokość	6
2.5. Kompas	7
3. Funkcjonalności	8
3.1 Akcelerometr	9
3.1.1 Obsługa przez interfejs SPI	10
3.1.2 Obsługa przez interfejs I ² C	11
3.1.3 Rejestry wewnętrzne SPI/I ² C	12
3.2 Magnetometr	13
3.2.1 Tryby pracy	13
3.2.2 Zakresy pomiarowe	14
3.2.3 Rejestry wewnętrzne SPI/I ² C	15
3.3 Barometr	15
3.3.1 Tryby zasilania	15
3.3.2 Rejestry	16
3.3.3 Pomiary ciśnienia	17
3.3.4 Zakresy pomiarowe	17
3.4 Czytnik linii papilarnych	17
3.4.1 Tryby pracy	18
3.4.2 Polecenia	18
3.4.3 Rejestry	19
3.5 Czujnik światła	20
3.5.1 Zakresy pomiarowe	21
3.5.2 Polecenia i rejestry	21
3.6 GPS	22
3.6.1 Obsługa przez UART	23
3.6.2 Protokół komunikacyjny NMEA	23

1. Dokumentacja użytkownika

TouristAPP to aplikacja, którą można wykorzystać np. podczas wędrowki. Dostarcza nam ona takie funkcjonalności jak:

- Odczyt współrzędnych geograficznych,
- Kompas,
- Odczyt ciśnienia atmosferycznego,
- Odczyt wysokości n.p.m,
- Włączenie/wyłączenie latarki,
- Uruchomienie świetlnego sygnału SOS,
- Zmianę motywu na podstawie jasności otoczenia lub przycisku.

1.1. Wymagania techniczne aplikacji

Aby skorzystać z wszystkich dostępnych funkcjonalności aplikacji smartfon powinien spełniać następujące wymagania:

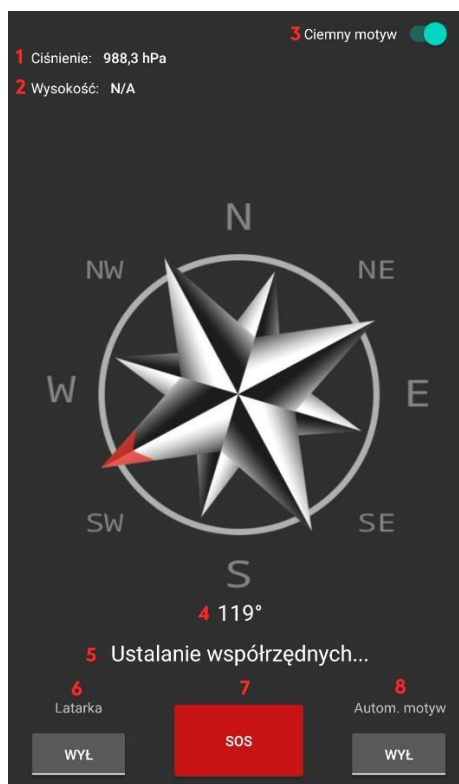
- System operacyjny Android wersja 6.0 (API 23) lub nowsza, *
- Czujnik ciśnienia atmosferycznego (barometr),
- Akcelerometr, *
- Magnetometr,
- Dioda LED,
- GPS,
- Wibrator,
- Czujnik światła.

* - **wymagania minimalne**

1.2. Instrukcja użytkownika

W celu instalacji aplikacji należy posiadać smartfon spełniający minimalne wymagania aplikacji określone w pkt. 1.1. Instalację przeprowadzić można na 2 sposoby:

- Instalacja z pliku apk (TouristAPP.apk)
- Instalacja poprzez Sklep Google Play
<https://play.google.com/store/apps/details?id=com.sw.touristapp>



Rysunek 1. Zrzut okna aplikacji z dostępnymi funkcjonalnościami (1:8)

Podczas pierwszego uruchomienia aplikacji lub gdy odbierzemy aplikacji uprawnienie do lokalizacji urządzenia zostaniemy o nie poproszeni.

Aplikacja przy uruchomieniu sprawdza dostępność i uprawnienia do czujników:

- gdy nie jest włączona dokładna lokalizacja GPS, aplikacja poprosi o jej włączenie,
- gdy nie jest dostępny barometr, wartość ciśnienia ustawiana jest na "N/A" (rys. 1 pkt. 1).

Na środku okna aplikacji znajduje się dynamiczna róża wiatrów, która wskazuje kierunek północny. Poniżej wyświetlona jest wartość azymutu (rys. 1 pkt. 4).

Po ustaleniu lokalizacji GPS przez urządzenie zostanie wyświetlona wysokość nad poziomem morza (rys. 1 pkt. 2) oraz współrzędne geograficzne (rys. 1 pkt. 5).

Aplikacja umożliwia zmianę motywu za pomocą przełącznika (rys. 1 pkt. 3). Dostępna jest również opcja włączenia automatycznej zmiany motywu (rys. 1 pkt. 8).

Ponadto aplikacja udostępnia funkcjonalność latarki (gdy jest dostępna w urządzeniu), którą można włączyć/wyłączyć za pomocą przycisku (rys. 1 pkt. 6). W przypadku braku latarki nie mamy możliwości uruchomienia sygnału SOS.

Mamy również możliwość uruchomienia świetlnego sygnału SOS (rys. 1 pkt 7). Po wciśnięciu czerwonego przycisku umieszczonego w dolnym środkowym miejscu zostaniemy poproszeni o autoryzację za pomocą odcisku palca lub innego zabezpieczenia, które jest dostępne w naszym smartfonie. Następnie po udanej autoryzacji zostaje

uruchomiony dwukrotnie świetlny sygnał SOS (sekwencja – 3 szybkie, 3 krótkie, 3 szybkie mrugnięcia lampą błyskową).

2. Opis algorytmów

2.1. Sygnał SOS

Metoda wykonująca sygnał SOS przy pomocy latarki przyjmuje jako argument ilość powtórzeń sygnału SOS. Sygnał jest zapisany jako ciąg znaków (wartości logiczne) określający nadawanie sygnału świetlnego, gdzie "0" – latarka nie świeci, "1" – latarka świeci. Ciąg znaków nadawany jest z częstotliwością 250 ms. Przed nadaniem kolejnego powtórzenia sygnału SOS występuje przerwa 1500 ms.

```
/**
 * Metoda służąca do uruchomienia sygnału SOS z wykorzystaniem latarki.
 * @param step ilość powtórzeń sygnału SOS.
 */
private void turnOnSOSFlashSignal(final int step) {
    String signal = "1010100100100100101010";
    int delay = 0;
    Handler handler = new Handler();
    for (int i = 0; i < step; i++) {
        for (int j = 0; j < signal.length(); j++) {
            if (signal.charAt(j) == '1') {
                handler.postDelayed(new Runnable() {
                    @Override
                    public void run() {
                        flashlight.switchFlashLight( status: true);
                    }
                }, (delay += 250));
            }

            handler.postDelayed(() -> {
                flashlight.switchFlashLight( status: false);
            }, (delay += 250));
        }
        final int finalI = i;
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                if (finalI == step - 1) {
                    flashlight.flashlightButton.setEnabled(true);
                    sosButton.setEnabled(true);
                }
            }
        }, delay += 1500);
    }
}
```

Rysunek 2. Metoda wykonująca świetlny sygnał SOS

2.2. Motyw aplikacji

Zmiana motywu na ciemny następuje w momencie, gdy odczytana wartość jasności otoczenia z czujnika światła spadnie poniżej 50lx (luks), natomiast kiedy wartość wynosi powyżej 50lx aktywny jest jasny motyw.

```

/**
 * Metoda zostaje wywołana gdy zmieni się wartość czujnika. Kiedy wartość odczytana jest poniżej 50Lx (Lüks) uruchamiany jest ciemny motyw,
 * w przeciwnym razie pozostaje włączony motyw jasny.
 * @param event zdarzenie sensora
 */
@Override
public void onSensorChanged(SensorEvent event) {
    if(event.sensor.getType() == Sensor.TYPE_LIGHT){
        if(event.values[0] < 50f) {
            if(autoThemeButton.isChecked()) {
                nightThemeSwitch.setChecked(true);
            }
        }
        else {
            if(autoThemeButton.isChecked()) {
                nightThemeSwitch.setChecked(false);
            }
        }
    }
}
}

```

Rysunek 3. Metoda obsługująca automatyczną zmianę motywu

2.3. Ciśnienie atmosferyczne

Po odczytaniu ciśnienia przez barometr zostaje ono wyświetlone w oknie aplikacji z dokładnością do jednego miejsca po przecinku w jednostce hektopaskal.

```

/**
 * Metoda zostaje wywołana gdy zmieni się wartość czujnika.
 * @param event zdarzenie sensora
 */
@Override
public void onSensorChanged(SensorEvent event) {
    pressureValue = event.values[0];
    pressureTextView.setText(String.format("%.1f hPa", pressureValue));
}

```

Rysunek 4. Metoda odczytująca oraz wyświetlająca wartość ciśnienia atmosferycznego

2.4. Współrzędne geograficzne i wysokość

Po ustaleniu lokalizacji GPS przez urządzenie wartość współrzędnych i wysokość zostaje ustalona i wyświetlona.

```

/**
 * Metoda wywoływana po zmianie lokalizacji.
 * @param location zaaktualizowana lokalizacja
 */
@Override
public void onLocationChanged(@NonNull Location location) {
    double latitude = location.getLatitude();
    double longitude = location.getLongitude();
    coordinatesTextView.setText(String.format("%.4f, %.4f", latitude, longitude));
    altitudeTextView.setText(String.format("%.1f m n.p.m", location.getAltitude()));
}

```

Rysunek 5. Metoda odczytująca oraz wyświetlająca wartość współrzędnych oraz wysokość n.p.m

2.5. Kompas

Koncepcyjnie czujnik przyspieszenia określa przyspieszenie, które jest przyłożone do urządzenia, mierząc siły działające na sam czujnik przy użyciu zależności:

$$A_D = -\left(\frac{1}{mass}\right) \sum F_S$$

Jednak siła grawitacji zawsze wpływa na mierzone przyspieszenie zgodnie z następującą zależnością:

$$A_D = -g - \left(\frac{1}{mass}\right) \sum F_S$$

Gdy urządzenie jest położone na stole (i nie przyspiesza), akcelerometr odczytuje wartość $g = 9.81 \text{ m/s}^2$. Podobnie, gdy urządzenie spada swobodnie, a zatem gwałtownie przyspiesza w kierunku ziemi z prędkością 9.81 m/s^2 , akcelerometr wskazuje wielkość $g = 0 \text{ m/s}^2$. Aby zmierzyć rzeczywiste przyspieszenie urządzenia, z danych akcelerometru należy usunąć udział siły grawitacji. Można to osiągnąć stosując filtr górnoprzepustowy. Natomiast, aby odizolować siłę grawitacji możemy użyć filtr dolnoprzepustowy, który zaimplementowaliśmy w poniższym kodzie.

```
/**
 * Metoda zostaje wywołana gdy zmieni się wartość czujnika.
 * @param event zdarzenie sensora
 */
@Override
public void onSensorChanged(SensorEvent event) {
    final float alpha = 0.96f;
    synchronized (this) {
        if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
            gravity[0] = alpha * gravity[0] + (1 - alpha) * event.values[0];
            gravity[1] = alpha * gravity[1] + (1 - alpha) * event.values[1];
            gravity[2] = alpha * gravity[2] + (1 - alpha) * event.values[2];
        }

        if (event.sensor.getType() == Sensor.TYPE_MAGNETIC_FIELD) {
            geoMagnetic[0] = alpha * geoMagnetic[0] + (1 - alpha) * event.values[0];
            geoMagnetic[1] = alpha * geoMagnetic[1] + (1 - alpha) * event.values[1];
            geoMagnetic[2] = alpha * geoMagnetic[2] + (1 - alpha) * event.values[2];
        }

        if (SensorManager.getRotationMatrix(rotationMatrix, null, gravity, geoMagnetic)) {
            final float[] orientation = new float[3];
            SensorManager.getOrientation(rotationMatrix, orientation);
            app.runOnUiThread(() -> {
                changeUI((float) -toDegrees(orientation[0]));
            });
        }
    }
}
```

Rysunek 6. Metoda ustalająca azymut

W naszym programie zastosowaliśmy stałą α , aby utworzyć filtr dolnoprzepustowy.

$$\alpha = \frac{t}{(t + dT)}$$

Stała filtra jest wyprowadzana ze stałej czasowej (t), która jest przybliżoną reprezentacją opóźnienia, które filtr dodaje do zdarzeń czujnika, oraz szybkości dostarczania zdarzeń przez czujnik (dT). W naszej aplikacji wartość α jest ustawiona na 0.96, przez co uzyskaliśmy stabilność wskazówki kompasu.

Gdy urządzenie leży płasko na stole w swoim naturalnym położeniu obowiązują następujące warunki:

- Gdy urządzenie jest przesuwane w lewą stronę (tak, aby poruszało się w prawo), wartość przyspieszenia x jest dodatnia,
- Gdy urządzenie jest przesuwane do dołu (tak, aby odsuwało się użytkownika), wartość przyspieszenia y jest dodatnia,
- Gdy przesuniesz urządzenie w kierunku nieba z przyspieszeniem a , wartość przyspieszenia z będzie równa $a + 9.81$, co odpowiada przyspieszeniu urządzenia ($a - 9.81 \text{ m/s}^2$).

Czujnik pola magnetycznego umożliwia monitorowanie zmian pola magnetycznego Ziemi. Czujnik dostarcza dane natężenia pola (w μT) dla każdej z trzech osi współrzędnych.

W naszej aplikacji użyliśmy akcelerometru i czujnika pola geomagnetycznego (jeśli jest dostępny) w połączeniu z macierzą obrotu uzyskiwaną metodą [`getRotationMatrix\(\)`](#). Metoda ta przekształca wektor z układu współrzędnych urządzenia do globalnego układu współrzędnych. Następnie na podstawie uzyskanej macierzy obrotu metoda [`getOrientation\(\)`](#) zwraca tablicę wartości, gdzie pierwsza wartość określa wartość azymutu w radianach. Następnie wartość ta zamieniana jest na kąt mierzony w stopniach (0-360°) i przekazywana do UI.

3. Funkcjonalności

Poniżej wymienione zostały wykorzystane funkcjonalności urządzenia oraz opis ich zastosowania.

Device Info HW	
OGÓLNE SOC SYSTEM PAMIĘĆ KAMERA BATERIA	
Urządzenie	Xiaomi MI 5
Rozdzielczość	1920x1080
Platforma	qcom
Android	8.0.0
Kernel	3.18.71-perf-g5edcf86 #1 SMP PREEMPT Wed Oct 23 18:37:20 CST 2019
LCM	jdi_j1_fhd_cmd_incell_dsi_panel
Ekran dotykowy	synaptics_dsx_i2c
Akcelerometr	LSM6DS3
Als/ps	CM36686
Magnetometr	AK09915
Żyroskop	LSM6DS3
Barometr	BMP280
Ładowarka	smb1351-charger
NFC	pn548
Audio	tfa98xx
Fingerprint	fpc1020
Inny	at24 drv2604 ti_tusb320 lp5562
Wi-Fi	qca6174
Dźwięk	msm8996tashamtp
RAM	3 GB LPDDR4
Flash	UFS

Rysunek 7. Zrzut ekranu z aplikacji *Device Info HW*.

3.1 Akcelerometr

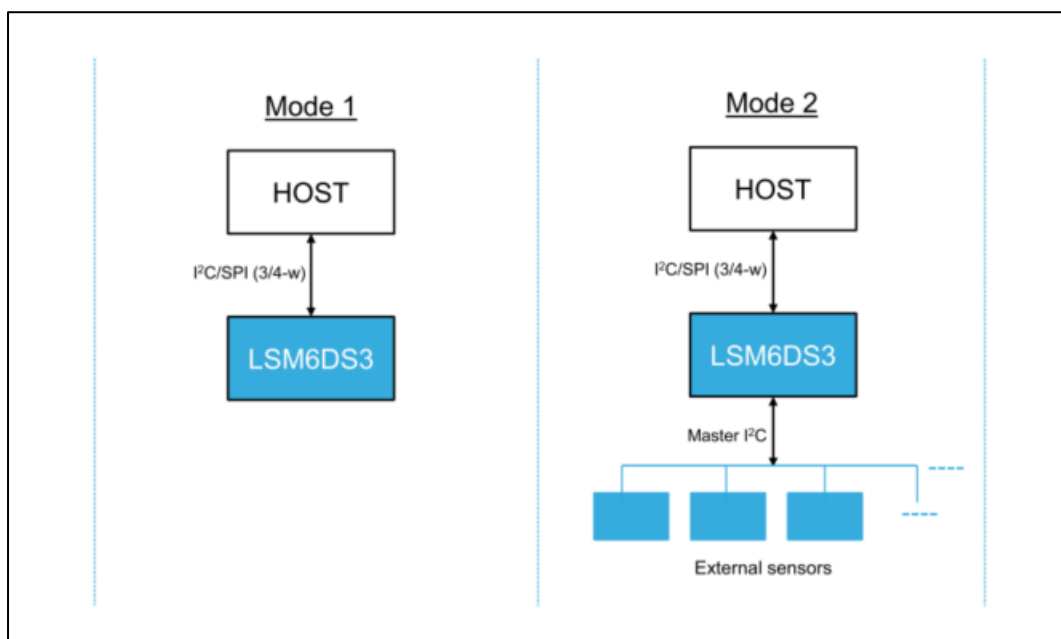
Akcelerometr *LSM6DS3* to cyfrowy trójosiowy akcelerometr i żyroskop w pakiecie z cyfrowym standardowym wyjściem interfejsu I²C/SPI. Oferuje on inteligentne funkcje uśpienia i przebudzenia, które umożliwiają zaawansowane oszczędzanie energii. Urządzenie posiada dynamicznie wybierane przez użytkownika cztery zakresy pomiarowe przyspieszenia ziemskiego w pełnej skali $\pm 2/\pm 4/\pm 8/\pm 16$ g oraz zakres prędkości kątowej $\pm 125/\pm 250/\pm 500/\pm 1000/\pm 2000$ dps (dps == stopnie na sekundę). Akcelerometr został zaprojektowany w celu implementacji funkcji ruchu, pochylenia, krokomierza. *LSM6DS3* ma zintegrowany inteligentny bufor FIFO (first-in-first-out) o wielkości do 8 kB. Umożliwia on dynamiczne grupowanie znaczących danych takich jak: czujniki zewnętrzne, licznik kroków, znacznik czasu i temperatury).

LSM6DS3 zapewnia trzy możliwe konfiguracje operacyjne:

- Tylko akcelerometr aktywny i żyroskop w trybie wyłączenia,
- Tylko żyroskop aktywny i akcelerometr w trybie wyłączenia,
- Zarówno akcelerometr jak i żyroskop aktywne z niezależnym ODR.

Oraz dwa tryby pracy:

- **Tryb 1:** Dostępny jest interfejs slave I²C lub interfejs szeregowy SPI,
- **Tryb 2:** Dostępny jest interfejs slave I²C lub interfejs szeregowy SPI oraz interfejs I²C master do podłączenia czujników zewnętrznych.



Rysunek 8. Tryby połączenia LSM6DS3

Ważną kwestią jest dostęp do rejestrów umieszczonych w LSM6DS3, który można uzyskać zarówno poprzez interfejs I²C jak i SPI. Ponadto interfejs SPI może zostać skonfigurowany do pracy w trybie interfejsu 3-przewodowego lub 4-przewodowego. Korzystanie z obu interfejsów jest możliwe poprzez mapowanie #12 pinu CS naprzemiennie/zamiennie przez każdy z interfejsów. Interfejs SPI aktywuje się w stanie niskim CS "0", natomiast I²C w stanie wysokim CS "1". Urządzenie jest kompatybilne z trybami interfejsu SPI 0 i 3.

Nie jesteśmy w stanie określić, z którego interfejsu korzysta wykorzystywany przez nas smartfon przy komunikacji z akcelerometrem *LSM6DS3*.

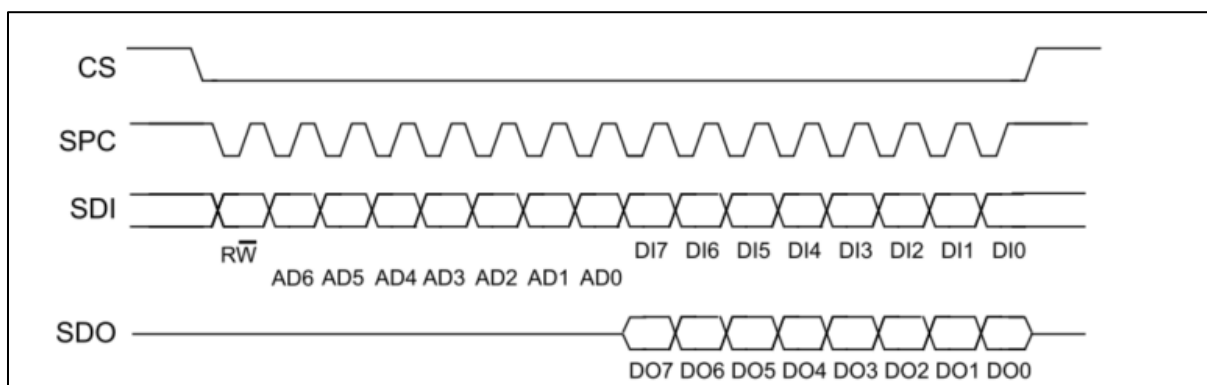
3.1.1 Obsługa przez interfejs SPI

Aby komunikacja odbywała się poprzez interfejs SPI należy ustawić na dedykowanym GPIO CS "0".

Wymagania czasowe interfejsu¹:

- Cykl zegara (SPI) – 100 ns,
- Częstotliwość zegara (SPI) – 10 MHz,
- Czas przygotowawczy (CS) – min 5 ns,
- Czas zatrzymania (CS) – min 20 ns.

¹ Powyższe wartości czasowe są gwarantowane przy częstotliwości zegara SPI 10 MHz w trybach interfejsu 3 i 4-przewodowym.



Rysunek 9. Odczyt i zapis rejestrów w interfejsie SPI (tryb 3).

Interfejs magistrali SPI w urządzeniu LSM6DS3 działa w trybie slave oraz umożliwia zapisywanie i odczytywanie rejestrów urządzenia. Interfejs szeregowy komunikuje się z aplikacją przy pomocy 4 przewodów: **CS**, **SPC**, **SDI** i **SDO**.

Tryb master interfejsu SPI odpowiada za kontrolowanie CS, który umożliwia włączenie portu szeregowego. Na początku transmisji CS spada nisko, a na koniec rośnie wysoko. Zegar portu szeregowego SPC jest również kontrolowany przez master SPI. Gdy CS jest wysoko (brak transmisji), SPC zatrzymuje się w stanie wysokim.

SDI oraz SDO to odpowiednio dane wejściowe i wyjściowe portu szeregowego.

Zarówno odczyt jak i zapis rejestru jest wykonywany w **16** cyklach zegara lub w wielokrotności **8** w sytuacji odczytu/zapisu wielu bajtów. Czas trwania bitu to jeden okres SPC.

Bit 0: RW bit. Kiedy bit jest równy 0 DI(7:0), dane są zapisywane w urządzeniu, kiedy bit jest równy 1 DO(7:0), dane są odczytywane z urządzenia.

Bit 1-7: adres AD(6:0) pole adresowe indeksowanego rejestru.

Bit 8-15: DI(7:0) (tryb zapisu), dane które są zapisywane w urządzeniu.

Bit 8-15: DO(7:0) (tryb odczytu), dane które są odczytywane z urządzenia.

3.1.2 Obsługa przez interfejs I²C

Aby komunikacja odbywała się poprzez interfejs I²C należy ustawić na dedykowanym GPIO CS "1".

Parametr	Tryb standardowy		Tryb szybki		Jednostka
	MIN	MAX	MIN	MAX	
Częstotliwość zegara SCL	0	100	0	400	kHz
Niski czas zegara SCL	4.7	-	1.3	-	μs

Wysoki czas zegara SCL	4.0	-	0.6	-	
Czas konfiguracji SDA	250	-	100	-	ns
Czas przechowywania danych SDA	0	3.45	0	0.9	μs

Tabela 1. Wymagania czasowe interfejsu I²C

Interfejs magistrali I²C działa w trybie slave, służy do zapisywania danych w rejestrach, które można również odczytać ponownie. Z magistralą I²C połączone są dwa sygnały (SCL i SDA). SDA jest linią kierunkową i służy do wysyłania i odbierania danych do/z interfejsu. Interfejs I²C jest również implementowany w trybie szybkim (400 kHz).

Transakcja poprzez interfejs magistrali I²C rozpoczyna się sygnałem START (ST). Warunek START (początkowy) jest zdefiniowany jako przejście ze stanu wysokiego do niskiego w linii danych, podczas gdy linia SCL jest utrzymywana w stanie wysokim. Po przesłaniu danych przez urządzenie w trybie master, magistrala uważana jest za zajęta. Następny bajt danych przesyłanych po spełnieniu warunku początkowego zawiera adres urządzenia podrzędnego w pierwszych 7 bitach, a 8 bit informuje czy urządzenie odbiera/przesyła dane z/do urządzenia podrzędnego. Po przesłaniu adresu, każde urządzenie w systemie porównuje pierwsze 7 bitów po spełnieniu warunku początkowego z jego adresem. Jeśli się zgadzają, urządzenie uważa się zaadresowane przez urządzenie nadrzędne.

3.1.3 Rejestry wewnętrzne SPI/I²C

Interfejsy **SPI** i **I²C** posiadają wspólne rejestry wewnętrzne 8/16 bitowe osadzone w urządzeniu z odpowiadającymi im adresami.

Wybrane istotne wykorzystywane rejestry wewnętrzne:

- WHO_AM_I (0x0F) – jego wartość jest ustalona na (0x69), służy do testu poprawności komunikacji, tryb tylko do odczytu
- OUTX_L_XL (0x28), OUTX_H_XL (0x29) – para rejestrów 8-bitowych przechowująca wartość wyjściową (16 bitową) z czujnika przyspieszenia liniowego (akcelerometru) w osi X, tryb tylko do odczytu
- OUTY_L_XL (0x2A), OUTY_H_XL (0x2B) – wartość wyjściowa z akcelerometru w osi Y, tryb tylko do odczytu
- OUTZ_L_XL (0x2C), OUTZ_H_XL(0x2D) – wartość wyjściowa z akcelerometru w osi Z, tryb tylko do odczytu
- CTRL1_XL (0x10) – rejestr kontrolny czujnika akcelerometru, zawiera informację o szybkości przesyłania danych oraz trybie zasilania (4 najstarsze bity – domyślna wartość 0000). Na kolejnych dwóch bitach zawarte są informacje o wybranym zakresie pomiarowym (00: ±2 g (domyślny); 01: ±16 g; 10: ±4 g; 11: ±8 g). Na ostatnich dwóch

bitach zawarte są informacje o wybranej przepustowości filtra wygładzającego (00: 400 Hz (domyślny); 01: 200 Hz; 10: 100 Hz; 11: 50 Hz).

3.2 Magnetometr

AK09915 to 3-osiowy elektroniczny kompas IC z wysokoczułym czujnikiem Halla. Zawiera czujniki magnetyczne do wykrywania ziemskiego magnetyzmu w osi X, osi Y i osi Z, obwód sterujący czujnika, wzmacniacz sygnału oraz funkcje autotestu. Dzięki niewielkim rozmiarom możliwe jest zastosowanie go w smartfonach i tabletach, w którym umożliwia korzystanie z nawigacji pieszej z wykorzystaniem GPS.

Urządzenie to umożliwia pracę z interfejsem [I²C](#) lub [SPI](#) (tryb 4-przewodowy). Nie jesteśmy w stanie określić, z którego interfejsu korzysta wykorzystywany przez nas smartfon przy komunikacji z magnetometrem AK09915.

3.2.1 Tryby pracy

- **Tryb wyłączenia** – zasilanie prawie wszystkich obwodów wewnętrznych jest wyłączone. Wszystkie rejestry są dostępne w trybie wyłączenia. Dane przechowywane w rejestrach do odczytu/zapisu pozostają.
- **Tryb pojedynczego pomiaru** – kiedy ustawiony jest tryb pojedynczego pomiaru (bity MODE[4:0] = "00001"), rozpoczyna się pomiar czujnika magnetycznego. Po zakończeniu pomiaru i przetworzeniu sygnału, dane pomiarowe zapisywane są w rejestrach (HXL:HZH), a następnie urządzenie automatycznie przechodzi w tryb wyłączenia. Po przejściu do trybu wyłączenia zmieniają się bity (MODE[4:0] = "00000").
- **Tryb ciągłego pomiaru** – po ustawieniu trybu pomiaru ciągłego (1-6) pomiar czujnika magnetycznego rozpoczyna się okresowo z częstotliwością 10 Hz, 20 Hz, 50 Hz, 100 Hz, 200 Hz, 1Hz. Po zakończeniu pomiaru i przetworzeniu sygnału, dane pomiarowe zapisywane są w rejestrach (HXL:HZH), a wszystkie obwody z wyjątkiem obwodu minimalnego wymaganego do zliczenia długości cyklu są wyłączane (PD). Kiedy nastąpi kolejny okres pomiarowy AK09915 wybudza się automatycznie z trybu PD i ponownie rozpoczyna pomiar. Tryb pomiaru ciągłego kończy się, gdy ustawiony jest tryb wyłączenia (MODE[4:0] = "00000").

Tryb pracy	Ustawienia rejestru [4:0] bity	Częstotliwość pomiaru [Hz]
Tryb ciągłego pomiaru 1	00010	10
Tryb ciągłego pomiaru 2	00100	20
Tryb ciągłego pomiaru 3	00110	50
Tryb ciągłego pomiaru 4	01000	100
Tryb ciągłego pomiaru 5	01010	200
Tryb ciągłego pomiaru 6	01100	1

Tabela 2. Tryby pomiaru ciągłego

- **Tryb autotestu** – służy do sprawdzenia czy czujnik magnetyczny działa normalnie. Gdy zostanie ustawiony tryb autotestu (MODE[4:0] = "10000") pole magnetyczne jest generowane przez wewnętrzne źródło magnetyczne i wykonywany jest pomiar czujnika magnetycznego. Dane pomiarowe są zapisywane w rejestrach danych pomiarowych (HXL:HZH), a następnie AK09915 automatycznie przechodzi do trybu wyłączenia. Sekwencja odczytu danych i funkcje rejestrów tylko do odczytu w trybie autotestu są takie same jak w trybie pojedynczego pomiaru.

Po włączeniu zasilania AK09915 znajduje się w trybie wyłączenia. Gdy określona wartość jest ustawiona na bity MODE[4:0], AK09915 przechodzi do określonego trybu. Gdy użytkownik chce zmienić tryb pracy, należy najpierw przejść do trybu wyłączenia, a następnie do innego trybu. Po ustawieniu trybu wyłączenia wymagane jest co najmniej 100µs (mikrosekund) przed ustawieniem innego trybu.

Ponadto urządzenie może wybrać jeden z dwóch trybów: „niski pobór mocy” lub „niski poziom szumów”. Tryb niskiego poboru mocy służy do oszczędzenia energii, a tryb niskiego poziomu szumów do zmniejszenia szumów. W trybie niskiego poboru mocy (bit SDR = "0") średni pobór energii przy częstotliwości 100 Hz zmniejsza się z 1.8 mA do 0.9 mA. Natomiast w trybie niskiego poziomu szumów (bit SDR = "1"), szum magnetycznych danych wyjściowych jest mniejszy niż w trybie niskiego poboru mocy (około 70% z trybu niskiego poboru mocy).

Bit SDR można zmienić tylko w trybie wyłączenia. Domyślnie bit SDR jest ustawiony na „0” czyli w trybie niskiego poboru mocy.

3.2.2 Zakresy pomiarowe

Parametr	Symbol	Warunki	Min.	Typ.	Max.	Jednostka
Bit wyjściowy danych pomiarowych	DBIT	-	-	16	-	bit
Czas pomiaru	TSM	Tryb pojedynczego pomiaru SDR bit = "0"		4.5		ms
		SDR bit = "1"		8.5		
Czułość czujnika magnetycznego	BSE	Ta = 25 °C	0.1425	0.15	0.1575	µT/LSB
Zakres pomiarowy czujnika magnetycznego	BRG	Ta = 25 °C	±4670	±4912	±5160	µT
Początkowe przesunięcie czujnika magnetycznego		Ta = 25 °C	-2000		+2000	LSB

Tabela 3. Zakresy pomiarowe urządzenia AK09915

AK09915 ma ograniczenie zakresu pomiarowego. Suma wartości bezwzględnych każdej osi X, Y, Z powinna być mniejsza niż 4912 µT.

$$|X| + |Y| + |Z| < 4912 \mu T$$

Gdy pole magnetyczne przekroczy to ograniczenie, dane pomiarowe są nieprawidłowe. Zdarzenie to określane jest przepełnieniem czujnika magnetycznego.

3.2.3 Rejestry wewnętrzne SPI/I²C

Interfejsy **SPI** i **I²C** posiadają wspólne rejestry wewnętrzne 8 bitowe osadzone w urządzeniu z odpowiadającymi im adresami. Urządzenie ma 29 rejestry.

Wybrane istotne wykorzystywane rejestry wewnętrzne:

- HXL (0x11), HXH (0x12) – para rejestrów 8-bitowych przechowująca wartość wyjściową (16 bitową) z czujnika magnetycznego w osi X, tylko do odczytu
- HYL (0x13), HYH (0x14) – wartość wyjściowa z czujnika magnetycznego w osi Y, tylko do odczytu
- HZL (0x15), HZH (0x16) – wartość wyjściowa z czujnika magnetycznego w osi Z, tylko do odczytu
- CNTL2 (0x31) – rejestr kontrolny czujnika magnetycznego, zawiera informację o trybie pracy (5 najmłodszych bitów – domyślna wartość 00000). 7 bit określa SDR (domyślnie „0”). Najstarszy bit określa włączenie/wyłączenie kolejki FIFO dostępnej tylko w [trybach ciągłego pomiaru](#).

3.3 Barometr

Urządzenie BMP280 to czujnik ciśnienia atmosferycznego zaprojektowany specjalnie do zastosowań mobilnych. Urządzenie to umożliwia precyzyjny pomiar ciśnienia atmosferycznego z dokładnością względną wynoszącą ± 0.12 hPa, co odpowiada ± 1 m różnicy wysokości. Urządzenie to umożliwia również pomiar temperatury (nasz smartfon nie obsługuje tej funkcjonalności).

Urządzenie to umożliwia pracę z interfejsem [I²C](#) lub [SPI](#) (tryb 3 lub 4-przewodowy). Nie jesteśmy w stanie określić, z którego interfejsu korzysta wykorzystywany przez nas smartfon przy komunikacji z barometrem BMP280.

3.3.1 Tryby zasilania

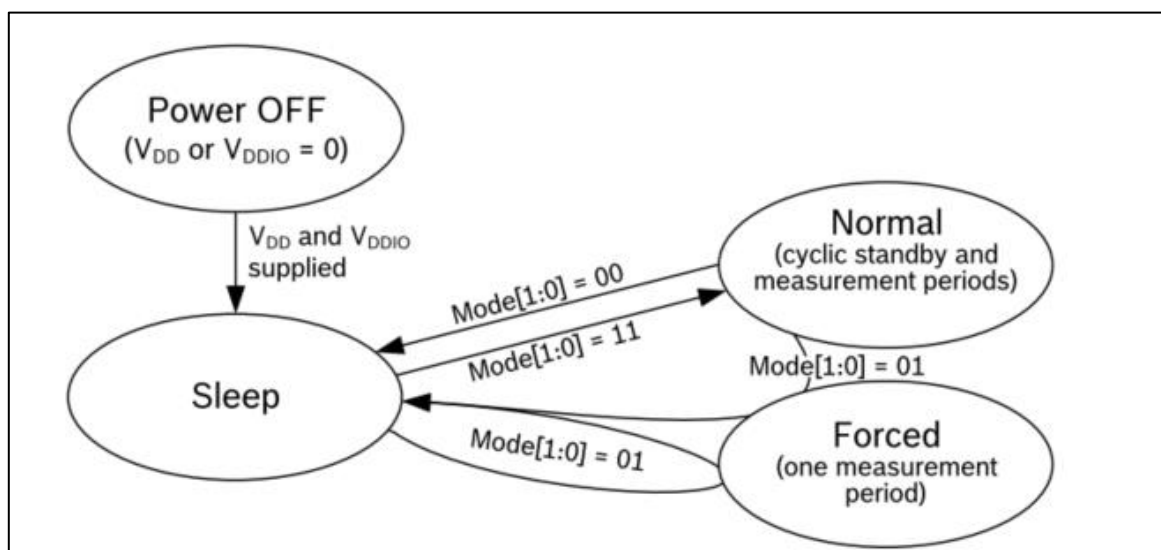
BMP280 oferuje 3 tryby zasilania, które można wybrać za pomocą bitów MODE[1:0] w rejestrze kontrolnym 0xF4.

Mode[1:0]	Tryb
00	Tryb uśpienia

01 i 10	Tryb wymuszony
11	Tryb normalny

Tabela 4. BMP280 tryby zasilania

- **Tryb uśpienia** – tryb ten jest ustawiany domyślnie po zresetowaniu zasilania. W trybie uśpienia nie są wykonywane żadne pomiary, a zużycie energii jest minimalne, natomiast wszystkie rejestry są dostępne.
- **Tryb wymuszony** – w trybie tym wykonywany jest pojedynczy pomiar zgodnie z wybranymi opcjami pomiaru i filtrowania. Po zakończeniu pomiaru czujnik powraca do trybu uśpienia, a wyniki pomiarów można odczytać z rejestrów danych. Do następnego pomiaru należy ponownie wybrać tryb wymuszony. Tryb wymuszony zalecany jest do aplikacji wymagających niskiej częstotliwości próbkowania lub synchronizacji na hoście.
- **Tryb normalny** – prąd w okresie gotowości (IDDSB) jest nieco wyższy niż w trybie uśpienia. Po ustawieniu opcji trybu, pomiaru i filtra, ostatnie wyniki pomiarów można uzyskać z rejestrów danych bez potrzeby dalszego dostępu do zapisu. Tryb normalny jest zalecany podczas korzystania z filtra IIR i jest przydatny w aplikacjach, w których należy filtrować zakłócenia krótkotrwałe (np. dmuchanie).



Rysunek 10. Schemat zmian trybów

3.3.2 Rejestry

Interfejsy **SPI** i **I²C** posiadają wspólne rejestry wewnętrzne 8 bitowe osadzone w urządzeniu z odpowiadającymi im adresami.

Wybrane rejestry wewnętrzne:

- rejestr 0xD0 "id" – zawiera numer identyfikacyjny układu chip_id[7:0] (0x58), tylko do odczytu.

- rejestr 0xE0 "reset" – jeśli zostanie zapisana wartość 0xB6, urządzenie zostanie zresetowane przy użyciu pełnej procedury resetowania. Zapisywanie innej wartości niż 0xB6 nie ma wpływu. Wartość odczytu wynosi zawsze 0x00.
- rejestry 0xF7, 0xF8, 0xF9 "press" – zawierają nieprzetworzone dane wyjściowe pomiaru ciśnienia z czujnika ciśnienia atmosferycznego, tylko do odczytu.
- Rejestr 0xF4 „ctrl_meas” – rejestr ustawia opcje akwizycji danych urządzenia, odczyt/zapis.

3.3.3 Pomiary ciśnienia

Pomiar ciśnienia można włączyć lub pominąć. Pominięcie pomiaru może być przydatne, jeśli BMP280 jest używany jako czujnik temperatury. Po włączeniu istnieje kilka opcji nadpróbkowania. Każdy etap nadpróbkowania zmniejsza szum i zwiększa rozdzielczość wyjściową o jeden bit. Włączenie / wyłączenie ustawień pomiaru i nadpróbkowania wybiera się za pomocą bitów osrs_p[2:0] w rejestrze kontrolnym 0xF4.

Ustawienie nadpróbkowania	Nadpróbkowanie ciśnienia	Typowa rozdzielczość ciśnienia
Pominięty pomiar ciśnienia	Pominięte	-
Bardzo niska moc	x1	16 bit / 2.62 Pa
Niska moc	x2	17 bit / 1.31 Pa
Standardowa rozdzielczość	x4	18 bit / 0.66 Pa
Wysoka rozdzielczość	x8	19 bit / 0.33 Pa
Bardzo wysoka rozdzielczość	x16	20 bit / 0.16 Pa

Tabela 5. Ustawienia nadpróbkowania pomiaru ciśnienia osrs_p

3.3.4 Zakresy pomiarowe

Parametr	Min	Max	Dokładność	Jednostka
Ciśnienie atmosferyczne ²	300	1100	± 0.12	hPa
	+9000	-500	± 1	m n.p.m. / p.p.m
Temperatura	-40	+85	0.01	°C

Tabela 6. Zakresy pomiarowe urządzenia BMP280

3.4 Czytnik linii papilarnych

FPC1020 to pojemnościowy dotykowy czytnik linii papilarnych o niskim zużyciu energii, opracowany i zoptymalizowany dla urządzeń mobilnych. Czujnik ten oferuje producentom smartfonów, tabletów i komputerów PC niespotykaną dotąd wydajność, a także możliwość

² Dokładność dla zakresu ciśnienia 700-900 hPa przy temperaturze pomiaru 25°C

odciążenia konsumentów od ciężaru używania kodów PIN i haseł do weryfikacji użytkownika. Czytnik FPC1020 w naszym smartfonie do komunikacji z hostem i szybkiego odczytu danych obrazu (odcisk palca) wykorzystuje interfejs SPI (4-przewodowy) i sygnał przerwania.

3.4.1 Tryby pracy

- **Zapytanie palcem (ang. Finger Query)** – wykonywana jest migawka wszystkich 12 podobszarów. Rejestr stanu (fngerPresentStatus) jest aktualizowany, aby wskazać, które obszary są w danym momencie objęte palcem.
- **Czekaj na palec (ang. Wait for Finger)** – procedura ta jest powtarzana, dopóki co najmniej jeden podobszar nie zostanie objęty w jednej z migawek i żadne nowe objęte obszary nie zostaną zarejestrowane w następnej migawce. Rejestr (fngerPresentStatus) jest aktualizowany w celu wskazania, które obszary są w tym momencie objęte palce i ustawiane jest przerwanie. Okres bezczynności pomiędzy powtarzającymi się zapytaniami można konfigurować.
- **Tryb uśpienia (ang. Sleep)** – czujnik budzi się w cyklicznych odstępach czasu, wykonując pomniejszony pomiar, który sprawdza tylko czy obszary 5 i 6 są pokryte. Jeśli na czujniku zostanie wykryty palec, system obudzi się i zostanie ustawione przerwanie. Okres bezczynności pomiędzy wybudzeniami jest konfigurowalny.

3.4.2 Polecenia

Poniższa tabela zawiera mapowanie adresów dla istotnych poleceń i rejestrów, które mogą zostać użyte do kalibracji i obsługi czytnika FPC1020.

Polecenie	Adres	Opis
Przechwyć obraz	0xC0	Przechwyć nowy obraz. Dostęp do jednego bajtu. Przesyłane jest tylko polecenie.
Odczyt danych obrazu	0xC4	Prawidłowe dane są odbierane po raz pierwszy po poleceniu z bajtem zastępczym. Odczyt jest kontynuowany, dopóki csN nie zostanie cofnięty.
Odczyt przerwania bez czyszczenia	0x18	Odczyt rejestru przerwań. Rejestr nie jest czyszczony.
Odczyt przerwania z czyszczeniem	0x1C	Odczyt rejestru przerwań. Rejestr jest czyszczony.
Sprawdzenie obecności palca	0x20	Sprawdza, czy palec jest obecny. Dostęp do jednego bajtu, przesyłane jest tylko polecenie.
Oczekuj na odcisk palca	0x24	Kontynuacja sprawdzania palca, aż palec będzie obecny. Dostęp do jednego bajtu, przesyłane jest tylko polecenie.

Tabela 7. Wybrane polecenia czytnika FPC1020

3.4.3 Rejestry

Wybrane rejestry wewnętrzne:

- rejestr 0xD8 "fngDetThrsh" [7:0] – określa wartość progową dla wykrycia palca. Umożliwia odczyt/zapis. Wartość progowa jest porównywana z sumą pikseli, obliczaną jako:

$$Pixelsum = \frac{1}{2} \sum_x \sum_y \frac{Pixel(x,y)}{8}$$

- rejestr 0xDC "fngDetCntr" [15:8], [7:0] – umożliwia odczyt/zapis, składa się z dwóch bajtów:
 - bajt 1 [15:8] "WaitFngDetCntr" – czas między zapytaniami dotyczącymi wykrywania palców w oczekiwaniu na tryb zapytania palcem. Czas oczekiwania obliczany jest na podstawie wzoru:

$$time = \frac{255 \cdot WaitFngDetCntr}{f_{clkSys}},$$

gdzie f_{clkSys} wynosi 10 MHz.

- bajt 0 [7:0] "SleepFngDetCntr" - czas między zapytaniami o wykrycie palca w trybie uśpienia. Czas uśpienia jest obliczany na podstawie wzoru:

$$time = \frac{64 \cdot SleepFngDetCntr}{f_{OscLo}},$$

gdzie f_{OscLo} określony jest w rejestrze oscTrim.

- rejestr 0xD4 "fngPresentStatus" – umożliwia tylko odczyt, określa status obecności palca. Ma dostęp do trzech bajtów: jednego bajtu adresu i dwóch bajtów danych (bajty 1:0):
 - [15:12] – resetowanie wartości,
 - [11:0] "fngPresentStatus"- każdy bit wskazuje obecność palca w jednym podobszarze, gdzie bit 0 jest podobszarem 0.
- rejestr 0x18 lub 0x1C "fpcInterrupts" – umożliwia odczyt/odczyt z czyszczeniem, ma dostęp do dwóch bajtów: jednego bajtu adresu i jednego bajtu odczytu. Odczyt z adresem 0x1C usunie rejestr przerw, natomiast odczyt z adresem 0x18 nie wyczyści rejestru przerw.
 - bit 0 – przyłożenie palca do czytnika,
 - bit 2 – wystąpienie błędu,
 - bit 5 – nowe dane obrazu dostępne w FIFO,
 - bit 7 – polecenie wykonane.

- rejestr 0x7C "clkBISTResult" – tylko do odczytu, umożliwia obliczanie częstotliwości oscylatora. Ma dostęp do pięciu bajtów: jednego bajtu adresu i czterech bajtów danych:
 - bajty 3:2:
 - bit 31 – resetowanie wartości,
 - bit 30 "doneSPI" – zero wskazuje, że test się nie powiódł,
 - bity [29:16] "resultSPI" – wynik testu oscylatora w stanie niskim. Wartość „1” lub „0” oznacza przepełnienie lub błąd.
 - bajty 1:0:
 - bit 15 – resetowanie wartości,
 - bit 14 "doneOscHi/doneclkSys" – zero wskazuje, że test się nie powiódł,
 - bity [13:0] "resultOscHi/resultclkSys" – wynik testu clkSys. Wartość „1” lub „0” oznacza przepełnienie lub błąd.
- rejestr 0x94 "oscTrim" – umożliwia odczyt/zapis. Rejestr ten wykorzystywany jest do kalibracji oscylatora, ma dostęp do 3 bajtów: jednego bajtu adresu i dwóch bajtów danych. Bieżąca zawartość rejestru jest odczytywana, gdy dane są zapisywane w rejestrze.
 - bajt 1:
 - bity [15:12] – reset lub odczytywana jest ostatnia wartość zapisana w tych bitach,
 - bity [11:8] – wartość przycinania wysokich częstotliwości oscylatora.
 - bajt 0:
 - bity [7:6] – reset lub odczytywana jest ostatnia wartość zapisana w tych bitach,
 - bity [5:0] – wartość przycinania niskich częstotliwości oscylatora:
 - bit 5 = 0 -> częstotliwość nominalna 16KHz,
 - bit 5 = 1 -> częstotliwość nominalna 8 KHz,

3.5 Czujnik światła

Ze względu na brak dostępności pełnej dokumentacji czujnika światła CM36686 Capella wykorzystywanego w naszym urządzeniu, do opisu wykorzystaliśmy dostępną dokumentację podobnego czujnika światła tej firmy CM32181 Capella.

CM32181 to zaawansowany czujnik światła otoczenia, który jest obsługiwany za pomocą interfejsu I²C. CM32181 zawiera fotodiode, wzmacniacze i obwody analogowe w jednym chipie. Zawiera doskonałą kompensację temperatury, solidne ustawienie częstotliwości odświeżania nie wymaga zewnętrznego filtra dolnoprzepustowego RC. Dostępny jest tryb wyłączenia oprogramowania, który zmniejsza zużycie energii do mniej niż 0.5µA. Napięcie robocze mieści się w zakresie od 2.5 V do 3.6 V i zużywa tylko 1uA. Maksymalna moc światła detektywistycznego to ponad 167000 Lux.

3.5.1 Zakresy pomiarowe

Parametr	Min	Max	Jednostka
Wykrywalna intensywność światła	0.005	167000	Lux

Tabela 8. Zakresy pomiarowe urządzenia CM32181 Capella

3.5.2 Polecenia i rejestry

CM32181 zawiera rejestr poleceń zapisany przez magistralę I²C. Wszystkie operacje mogą być kontrolowane przez rejestr poleceń. Prosta struktura poleceń umożliwia użytkownikowi łatwe programowanie ustawień pracy i pozyskiwanie danych z czujnika światła.

Wybrane polecenia oraz objaśnienia definicji rejestrów:

- Polecenie 0x00:
 - bity [12:11] "ALS_SM" – rejestr umożliwia zapis, określa tryb czułości czujnika światła (ALS),
 - bity [9:6] "ALS_IT" – rejestr umożliwia zapis, określa czas integracji ALS, czyli jak długo aktualizowana jest wartość odczytu,
 - bity [5:4] "ALS_PERS" – rejestr umożliwia zapis, definiuje ustawienie trwałości przerwania ALS,
 - bit 1 – rejestr umożliwia zapis, definiuje czy włączyć opcję przerwania, czy nie. 0 – wyłączone, 1 – włączone,
 - bit 0 – rejestr umożliwia zapis, określa sposób włączenia i wyłączenia czujnika. 0 – włączony, 1 – wyłączony.
- Polecenie 0x01 "ALS_WH" – 16 bitowy rejestr ustawienia okna wysokiego progu czujnika światła,
- Polecenie 0x02 "ALS_WL" – 16 bitowy rejestr ustawienia okna niskiego progu czujnika światła,
- Polecenie 0x03:
 - bity [2:1] "PSM" – rejestr umożliwia zapis, określa tryb oszczędzania energii, który zapewnia różne kombinacje zużycia energii, czułości i czasu odświeżania.
 - bit 0 – rejestr umożliwia zapis, określa działanie trybu oszczędzania energii. 0 – włączony, 1 – wyłączony.
- Polecenie 0x04 "ALS" – rejestr umożliwia odczyt 16-bitowych danych wyjściowych z czujnika światła.

Czas odświeżania CM32181 można określić za pomocą ustawień PSM i ALS_IT w aktywnym trybie oszczędzania energii (PSM). Przy pomocy poleceń i odpowiednich ustawień rejestrów, możliwe jest elastyczne definiowanie czasu, poboru mocy i czułości zbierania danych z czujnika światła.

ALS_SM	PSM	ALS_IT	Czas odświeżania (ms)	Pobór prądu (uA)	Czułość (Lux/bit)
01	00	0000	600	6	0.042
01	01	0000	1100	3.5	0.042
01	10	0000	2100	2	0.042
01	11	0000	4100	1	0.042
01	00	0001	700	10	0.021
01	01	0001	1200	6	0.021
01	10	0001	2200	3.5	0.021
01	11	0001	4200	2	0.021
01	00	0010	900	15	0.01
01	01	0010	1400	10	0.01
01	10	0010	2400	6	0.01
01	11	0010	4400	3.5	0.01
01	00	0011	1300	21	0.005
01	01	0011	1800	15	0.005
01	10	0011	2800	10	0.005
01	11	0011	4800	6	0.005

Tabela 9. Czas odświeżania, pobór prądu i czułość w określonych trybach oszczędzania energii

Automatyczne zapamiętywanie

CM32181 może zapamiętać ostatnie dane dotyczące natężenia światła otoczenia przed wyłączeniem i zachować te dane przed wybudzeniem. Gdy CM32181 jest w trybie wyłączenia, host może swobodnie czytać te dane bezpośrednio za pomocą polecenia odczytu. Gdy CM32181 się obudzi, dane zostaną odświeżone przez nową detekcję.

3.6 GPS

GPS jest systemem nawigacji satelitarnej stworzonym i utrzymywanym przez Departament Obrony USA, który swoim zasięgiem pokrywa całą kulę ziemską. Jego działanie polega na pomiarze czasu propagacji sygnału radiowego nadawanego przez satelity i odbieranego w dowolnym punkcie na Ziemi. Znając prędkość propagacji fal elektromagnetycznych, czas w jakim sygnał został nadany oraz pozycje co najmniej czterech nadajników, możliwe jest ustalenie położenia odbiornika.

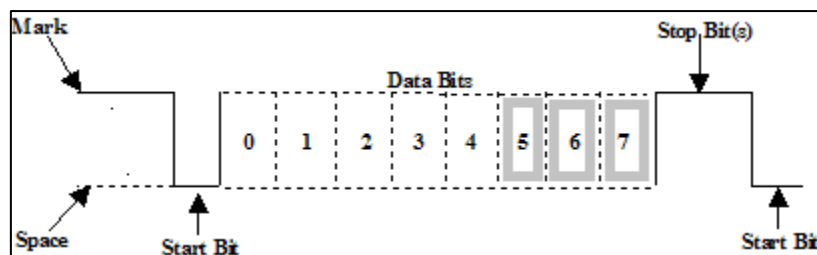
Sygnał nadawany z satelitów GPS zawiera almanach oraz efemerydę – informacje o położeniu i torze lotu satelitów na orbicie oraz dokładny czas odmierzany przez zegary atomowe – cezowe i rubidowe umieszczone w satelitach.

Satelity nadają swój sygnał w paśmie mikrofalowym, równocześnie na dwóch częstotliwościach nośnych – 1575,42 MHz oraz 1227,6 MHz, stosując modulację CDMA – nadając dla sygnału jedynki i zera kod rozpraszający, unikalny dla pojedynczej satelity i ortogonalny względem pozostałych kodów, tak aby nadajnik mógł rozróżnić sygnały nadawane przez inne satelity i jednocześnie nasłuchiwać na tylko jednej (lub dwóch) częstotliwościach.

Wszystkie niespecjalistyczne urządzenia GPS komunikują się z komputerami (smartfonami) za pomocą protokołu komunikacyjnego NMEA przy użyciu UART.

3.6.1 Obsługa przez UART

UART, czyli Universal Asynchronous Receiver and Transmitter, to urządzenie w postaci układu scalonego, które umożliwia komunikację, poprzez asynchroniczny przekaz i odbiór danych, za pośrednictwem portu szeregowego.



Rysunek 11. Schemat ramki UART

Transmisję informacji złączem szeregowym inicjuje tzw. bit startowy. W każdym przypadku jest to logiczne zero. W dalszej kolejności przesłana zostaje informacja w postaci 7, 8 lub 9 kolejnych bitów (w zależności od ustalonej konfiguracji urządzenia). Za zakończenie transmisji odpowiedzialny jest tzw. bit stopu, który zawsze jest równy logicznej jedynce. Całość tworzy tzw. ramkę UART, która zawiera w sobie kompletną informację.

W przypadku GPS komunikacja odbywa się przy pomocy protokołu NMEA 0183, przy odpowiedniej konfiguracji UART:

- Szybkość transmisji: 4800 bodów
- Liczba bitów danych: 8
- Brak kontroli parzystości
- Bity stopu: 1 (lub więcej)
- Komputerowa wymiana potwierdzeń (ang. Handshake): nie

3.6.2 Protokół komunikacyjny NMEA

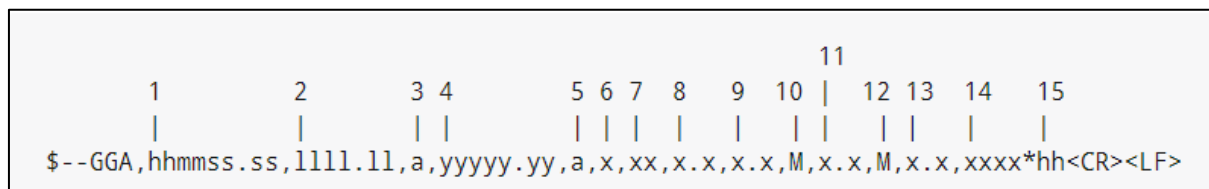
NMEA opublikowany przez National Marine Electronics Association protokół komunikacji między morskimi urządzeniami elektronicznymi. Ma on powszechne zastosowanie w elektronice nawigacji morskiej oraz urządzeniach GPS.

Dane są transmitowane w postaci „zdań” zapisanych kodem ASCII. Pojedyncza seria ramek standardu NMEA, składa się z maksymalnie 480 znaków ASCII, a długość każdej ramki ograniczona jest do 82 znaków. Znakiem zaczynającym dane w protokole jest „\$”, dalej następuje identyfikator zdania i pola danych oddzielone przecinkami, a na końcu znajduje się ciąg dwóch znaków CRLF.

Protokół NMEA może być wykorzystywany przez różne urządzenia. Można go zidentyfikować za pomocą tak zwanego identyfikatora nadawcy. W przypadku wykorzystania urządzenia jako GPS identyfikator to „GP” – Global Positioning System (GPS).

Protokół NMEA udostępnia wiele standardowych sekwencji (zdań). W przypadku GPS urządzenia najczęściej wykorzystują sekwencje RMC (Recommended Minimum Navigation Information), GLL (Geographic Latitude/Longitude) lub GGA (Global Positioning System Fix Data). Dostarczają one odbiornikowi GPS dane związane z czasem i pozycją.

Opis sekwencji GGA



Rysunek 12. Schemat sekwencji GGA

1. Czas UTC raportu o pozycji (aktualność danych)
2. Szerokość geograficzna (latitude)
3. N lub S (Północ lub Południe)
4. Długość geograficzna (longitude)
5. E lub W (Wschód lub Zachód)
6. Jakość pomiaru GPS
 - 0 – ustalanie pozycji GPS niedostępne
 - 1 – ustalanie pozycji GPS
 - 2 - Differential GPS fix (values above 2 are 2.3 features)
7. Ilość śledzonych satelitów, 00 - 12
8. Horyzontalna dokładność pozycji (w metrach)
9. Wysokość anteny powyżej / poniżej średniego poziomu morza (geoidy) (w metrach)
10. Separacja geoidalna, różnica między elipsoidą Ziemi WGS-84 a średnim poziomem morza (geoidą), "-" oznacza średni poziom morza poniżej elipsoidy
11. Jednostki separacji geoidalnej (w metrach)
12. Wiek różnicowy danych GPS, czas w sekundach od ostatniej aktualizacji SC104 typu 1 lub 9, wartość zerowa, gdy DGPS nie jest używany
13. Różnicowa stacja referencyjna ID, 0000-1023
14. Suma kontrolna

Przykład:

\$GPGGA,001043.00,4404.14036,N,12118.85961,W,1,12,0.98,1113.0,M,-21.3,M,,*47