



Instytut Informatyki Politechniki Śląskiej  
Zespół Mikroinformatyki i Teorii Automatów  
Cyfrowych



Rok akademicki:

Rodzaj studiów\*: SSI/NSI/NSM

Przedmiot (Języki  
Asemblerowe/SMiW):

Grupa

Sekcja

**2019/2020**

**SSI**

**SMiW**

**1**

**5**

Imię:

Dawid

Prowadzący:

OA/JP/KT/GD/BSz/GB

Nazwisko:

Jaraczewski

**KT**

## ***Raport końcowy***

Temat projektu:

**System lokalizacji pojazdu**

Data oddania:  
dd/mm/rrrr

**21.06.2020**

# 1. Temat projektu i założenia

System lokalizowania pojazdu konfigurowany przez WiFi udostępniający położenie pojazdu cyklicznie lub poprzez wywołanie.

Układ powinien być zasilany ogniwem, które będzie ładowane w trakcie użytkowania pojazdu wykorzystując jego instalację elektryczną.

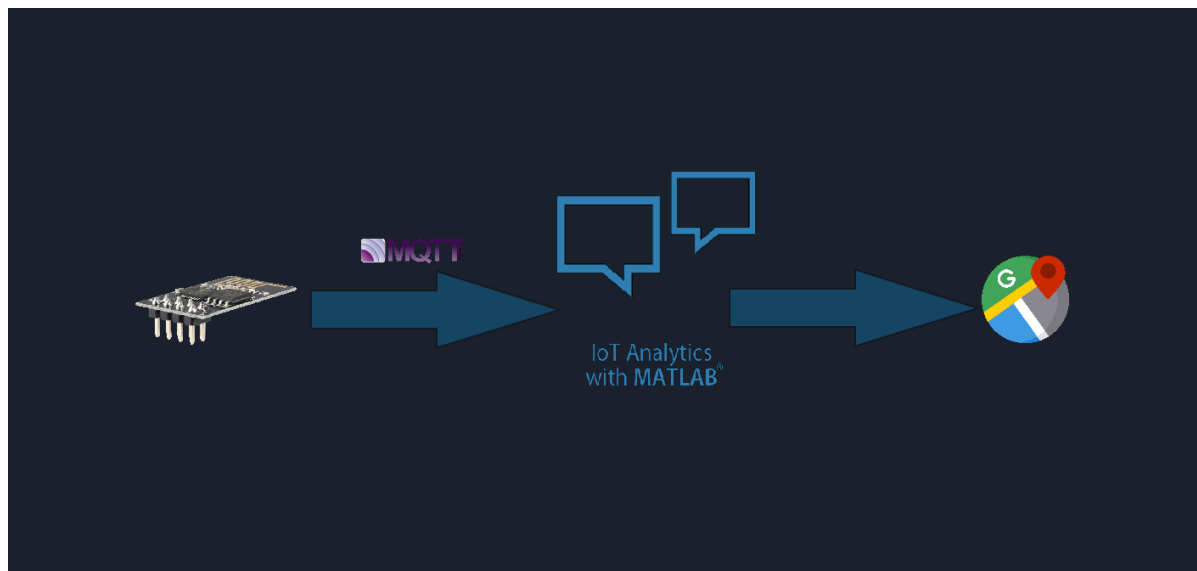
Konfiguracja urządzenia za pomocą WiFi:

- numer telefonu
- APN operatora
- konfiguracja MQTT

Cykliczne wysyłanie pakietu MQTT z lokalizacją urządzenia do brokera w chmurze (<https://Thingspeak.com>).

Wysyłanie za pomocą wiadomości tekstowej ostatniej znanej lokalizacji na wywołanie przez sms o odpowiedniej treści.

Idea:



## 2. Wybrane komponenty

Mikrokontroler: ESP8266-12S

Moduł GSM/GPS: Ai-thinker A9G

Kontroler ładowania: MCP73831

Akumulator SAMSUNG INR18650-35E | 3.6V 3500mAh 13A

Konwerter USB-UART: CH340C

Liniowy regulator napięcia LDO: AP2112

Antena GPS

Antena GSM

### 3. Specyfikacja wewnętrzna

Oprogramowanie mikrokontrolera:

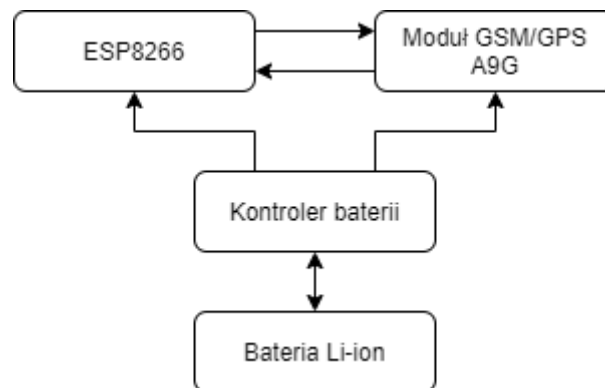
- PlatformIO + biblioteki Arduino (C++)

Konfiguracja i prezentacja danych:

- HTML
- CSS
- JavaScript

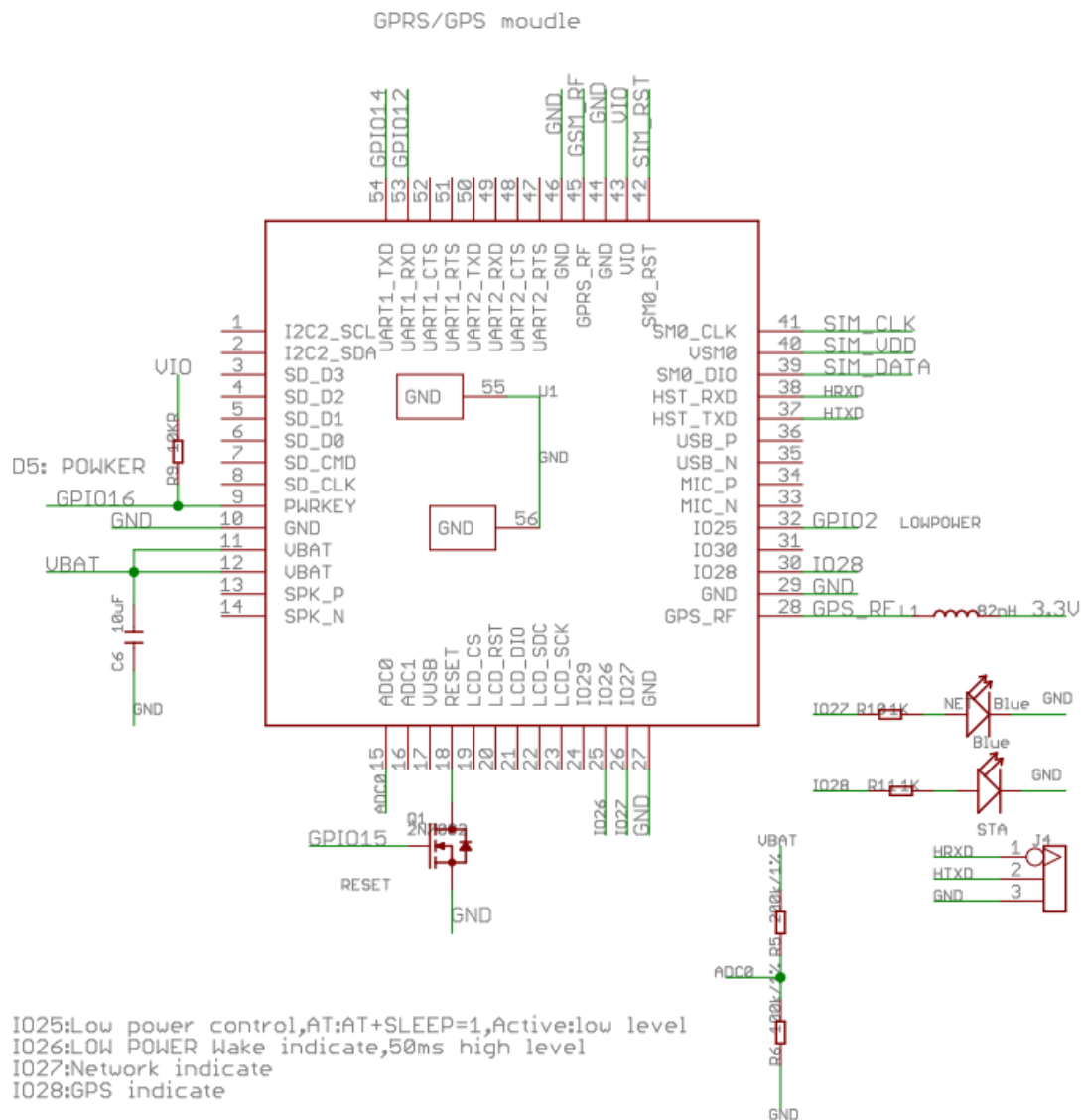
Broker MQTT: ThingSpeak

### 4. Schemat blokowy urządzenia

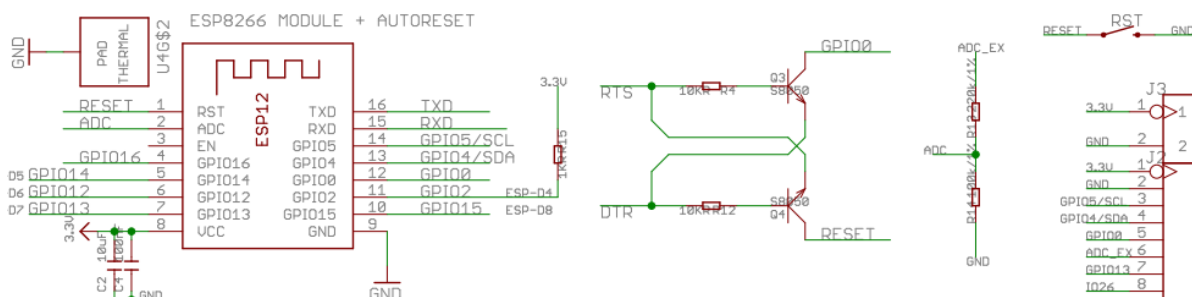


Schemat blokowy urządzenia

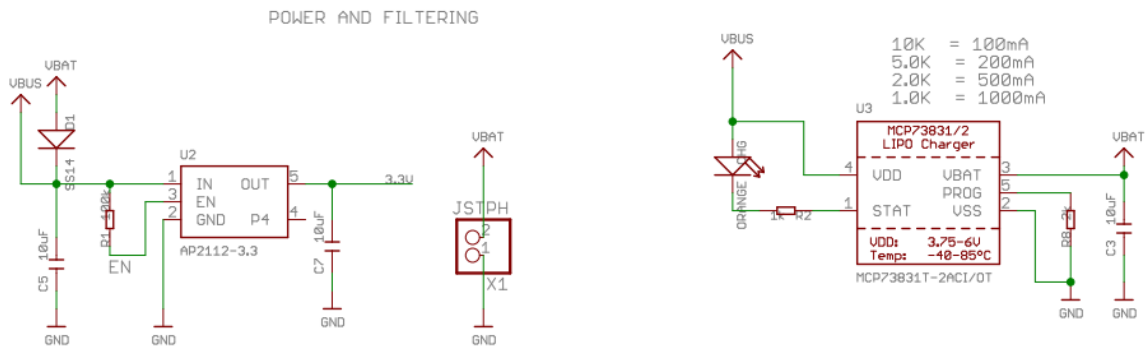
## 5. Schemat ideowy urządzenia



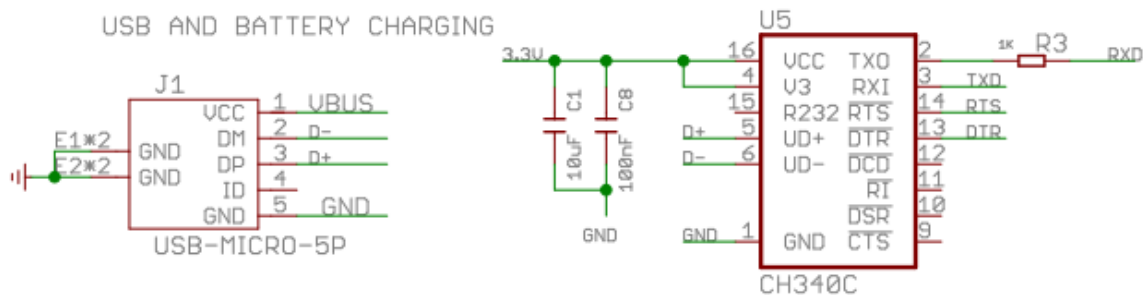
### Schemat modułu GPRS/GPS



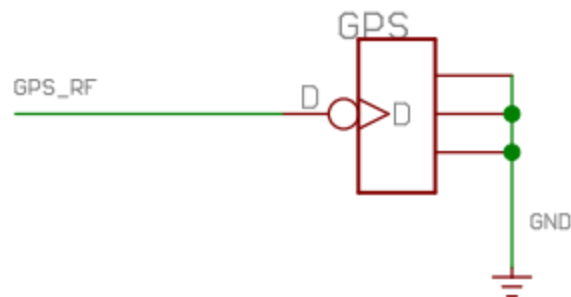
## Moduł ESP8266 + Autoreset



Układ zasilania i jego filtrowanie



Złącze microUSB



Antena GPS



Antena GSM

## 6. Opis funkcji poszczególnych bloków układu

Akumulator Li-ion dostarcza energię do całego układu przy braku zasilania z zewnątrz oraz służy do zasilania modułu GPRS/GSM. Moduł GPRS/GSM służy do uzyskiwania informacji o lokalizacji urządzenia, komunikacji między użytkownikiem a modułem ESP8266 na dużą odległość oraz wysyłania danych do serwera MQTT. Moduł ESP8266 pełni funkcję zarządzającą, przechowuje dane oraz umożliwia konfigurację urządzenia tworząc punkt dostępowy.

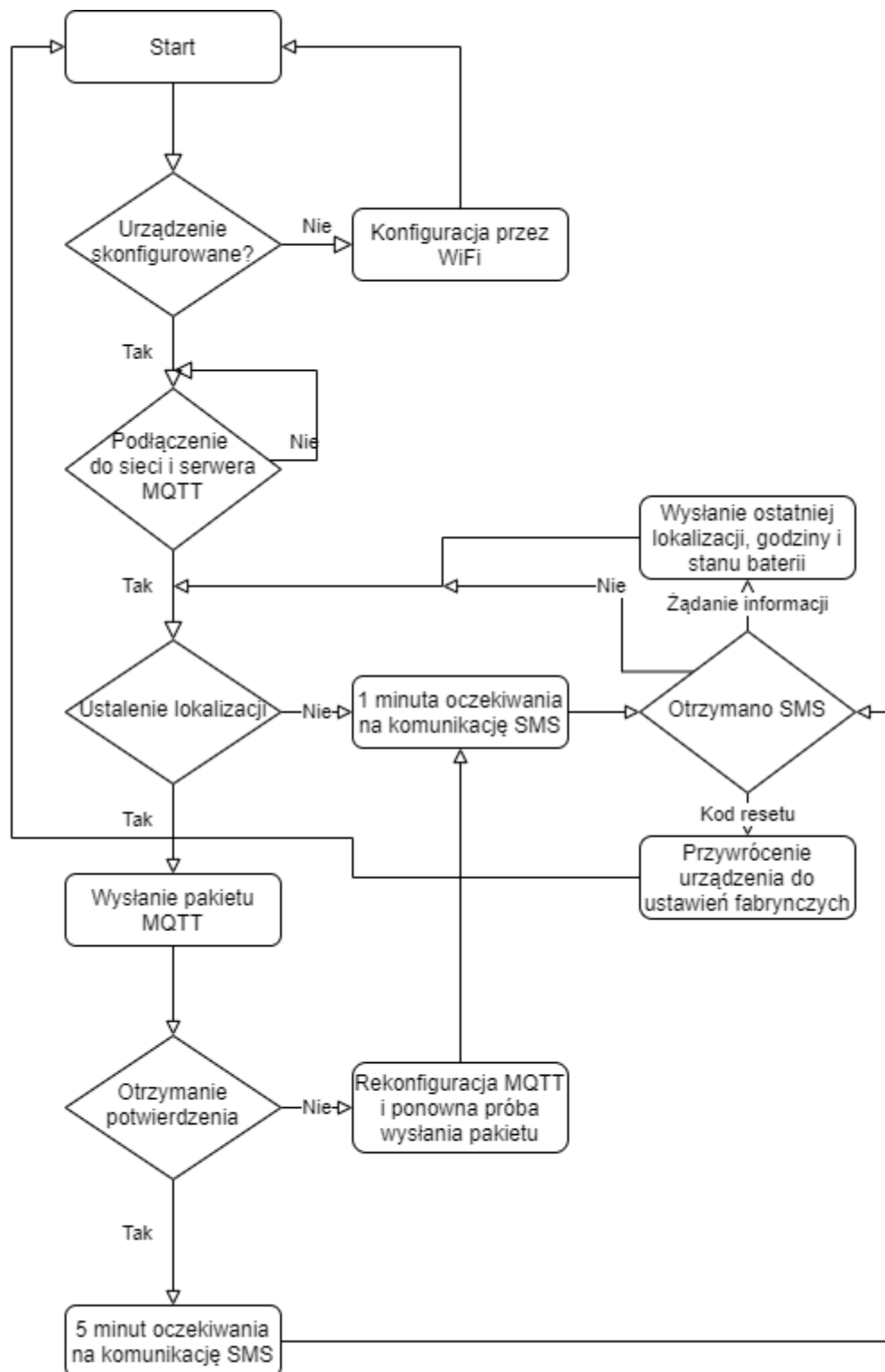
Złącze microUSB służy zarówno do zasilania układu (w tym także ładowania baterii) jak i programowania modułu ESP8266 (transceiver USB na UART CH340C).

## 7. Szczegółowy opis działania ważniejszych elementów układu

Moduł ESP8266 wraz z modulem GPRS/GPS są głównymi blokami wykonawczymi urządzenia. Moduł ESP8266 posiada awaryjny przycisk resetu, komunikuje się z modulem GPRS/GPS poprzez UART na pinach GPIO12 i GPIO14 podłączonych do portów UART1\_TXD i UART1\_RXD modułu GPRS/GPS. Moduł zasilania posiada filtr zakłóceń przewodzonych (zastosowane kondensatory odsprężające).

## 8. Algorytm oprogramowania urządzenia

Graf poniżej ilustruje uproszczone działanie algorytmu oprogramowania urządzenia:



## 9. Opis kodu programu

Kod programu został szczegółowo skomentowany i udostępniony na platformie GitHub [[https://github.com/dawidjar97/projekt\\_smiw\\_2020](https://github.com/dawidjar97/projekt_smiw_2020)], poniżej deklaracje głównych funkcji programu oraz funkcja główna programu(loop):

```
int A9GPOWERON(); //Funkcja włączająca moduł GPRS/GPS
void A9GMQTTCONNECT(); //Funkcja inicjująca połączenie do serwera MQTT
void a9gCommunication(String command, const int timeout); //Funkcja do komunikacji dwustronnej z modułem GPRS/GPS
void clearSms(); //Funkcja do usuwania wiadomości tekstowych z pamięci modułu GPRS/GPS

void loop()
{
    if(espReboot) //restart urządzenia po konfiguracji
    {
        delay(5000);
        server.end();
        WiFi.softAPdisconnect();
        ESP.restart();
    }
    if(!config) //normalny tryb pracy
    {
        a9gCommunication("AT+CBC?",1000); //Stan baterii
        a9gCommunication("AT+CCLK?",1000); //Aktualna data i czas
        a9gCommunication("AT+LOCATION=2",1000); //Lokalizacja
        /* sprawdzenie czy zwrócono lokalizację */
        if(a9gAnswer.indexOf(",") >= 0 && a9gAnswer.indexOf("GPS NOT FIX")==-1)
        {
            lastLocationTime=dateTime; //czas ostatniej lokalizacji
            separator=a9gAnswer.indexOf(",");
            lastLocation=(a9gAnswer.substring(separator-9,separator))+",
"+(a9gAnswer.substring(separator+1,separator+10)); //separacja współrzędnych i zapis do zmiennej
            if(!saveConfiguration(SPIFFS, nrTel, writeAPIKey, channelID, lastLocation, lastLocationTime))
                //zapis do pamięci
            {
                #if DEBUG
                    Serial.println("Something went wrong.");
                    pinMode(INTERNAL_LED, OUTPUT);
                    digitalWrite(INTERNAL_LED, HIGH);
                #endif
            }
            #if DEBUG
                Serial.println(lastLocation);
            #endif
            /* Komenda do przesłania MQTT */
            command="AT+MQTTPUB=\"channels/" + channelID + "/publish/" + writeAPIKey + "\"+ ","\"field1="
+ (a9gAnswer.substring(separator-9,separator)) + "&field2=" +
(a9gAnswer.substring(separator+1,separator+10)) + "&status=MQTTPUBLISH" + "\""+",0,0,0";
            a9gCommunication(command,5000); //Wysłanie pakietu MQTT

            if( a9gAnswer.indexOf("OK") >= 0 ) //Jeżeli wysłano pakiet poprawnie
            {
                #if DEBUG
                    Serial.println("MQTT sent");
                #endif
                a9gCommunication("",240000); //4 minuty oczekiwania na SMS
            }
            else //Jeżeli wystąpił błąd
            {
                config=1; //tryb konfiguracji
                A9GMQTTCONNECT(); //ponowne połączenie do serwera MQTT
                a9gCommunication(command,5000); //próba ponownego wysłania pakietu MQTT
            }
        }
        #if DEBUG
            Serial.println("60 seconds break");
        #endif
        a9gCommunication("",60000); //Minuta oczekiwania na SMS
        #if DEBUG
            Serial.println("End of loop");
        #endif
    }
}
```

## 10. Integracja oprogramowania z układem elektronicznym

Moduł ESP8266 komunikuje się z modułem GPRS/GSM poprzez zrealizowany programowo interfejs UART. Moduł ESP8266 uruchomiony w trybie debugowania wysyła także komunikaty na konsolę(gdy ta jest podłączona) poprzez sprzętowy interfejs UART. Moduł GPRS/GSM mierzy napięcie zasilania baterijnego i zamienia je na procentowy stan baterii.

## 11. Opis reakcji oprogramowania na zdarzenia zewnętrzne

Otrzymanie wiadomości tekstowej SMS o treści:

- Bateria - moduł wysyła wiadomość ze stanem baterii na zapisany w pamięci numer telefonu
- Lokalizuj - moduł wysyła wiadomość z ostatnią lokalizacją na zapisany w pamięci numer telefonu
- Info - moduł wysyła wiadomość z ostatnią lokalizacją, jej czasem, a także stanem baterii urządzenia na zapisany w pamięci numer telefonu
- ?R3ST4RT!. - moduł usuwa dotychczasową konfigurację i po upływie 500 ms resetuje się

## 12. Skrócona instrukcja obsługi urządzenia

Moduł należy podłączyć do zasilania poprzez złącze microUSB, a następnie podłączyć się dowolnym urządzeniem obsługującym WiFi 2.4GHz do punktu dostępowego "Lokalizator ESP12S" hasłem: 12345678.

Kolejnym krokiem jest połączenie się do urządzenia za pomocą przeglądarki internetowej(adres IP urządzenia to 192.168.4.1), a następnie konfiguracja podstawowych parametrów potrzebnych do działania (numer telefonu, writeAPIKey, channelID).

Przebieg konfiguracji:



Strona powitalna



# KONFIGURACJA

Potrzebujemy kilku informacji wymaganych do konfiguracji Lokalizatora GPS

Numer telefonu

+48515303896

MQTT WriteAPIKey

HRDDRV3GPRP

MQTT channelID

1048994



Zapisz

Właściwa konfiguracja

# GOTOWE

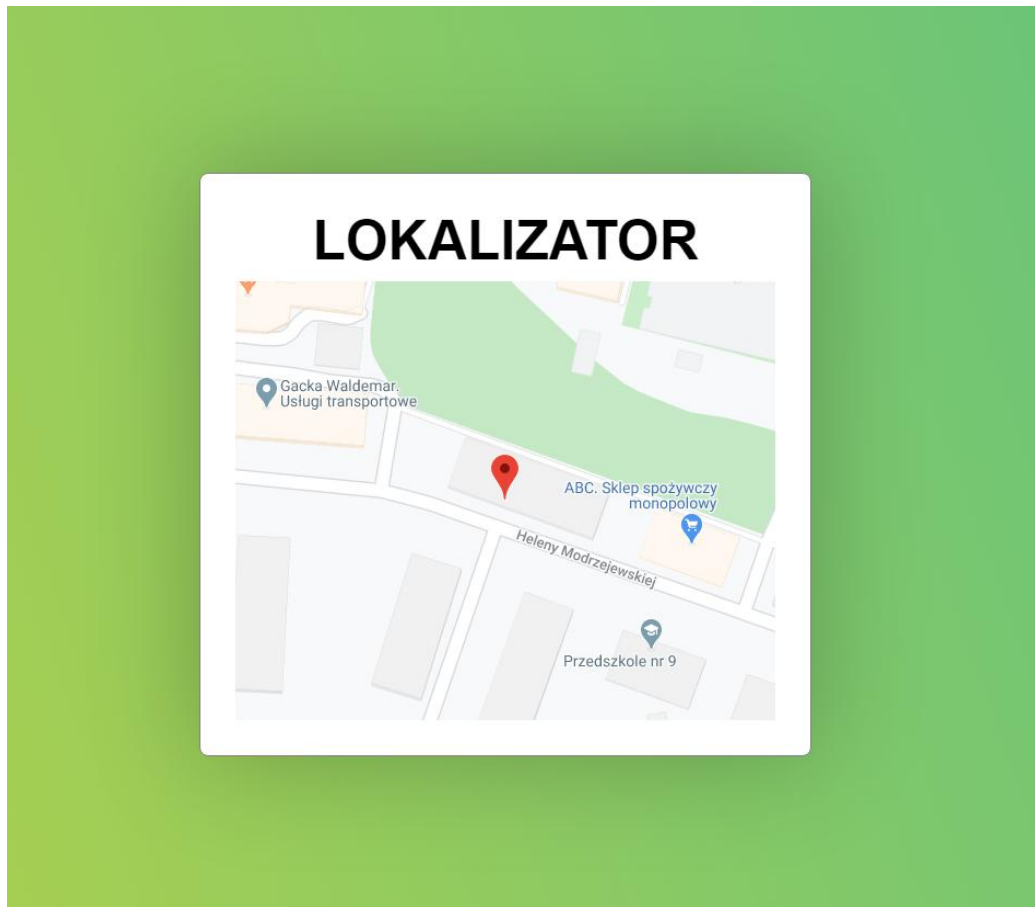


Urządzenie zostało skonfigurowane.

By zmienić ustawienia, zresetuj urządzenie zgodnie z instrukcją

Zakończenie konfiguracji

Po podaniu prawidłowych danych urządzenie zresetuje się i połączy z siecią. Dodatkowo utworzono stronę, która pobiera ostatnią lokalizację z brokera MQTT ThingSpeak oraz poprzez GoogleAPI pokazując lokalizację na mapie:



Zlokalizowane urządzenie

## 13. Opis montażu i uruchamiania

Płytkę wraz z koszykiem na baterię, baterią, anteną GSM i anteną GPS zostały zamontowane w zaprojektowanej wcześniej, a następnie wydrukowanej na drukarce 3D obudowie. Urządzenie zostało złożone zgodnie z przedstawionym wcześniej schematem, a przewody do przewodów zasilających została przylutowana wtyczka zasilania kompatybilna z tą umieszczoną na płycie.



Rozmieszczenie elementów w obudowie



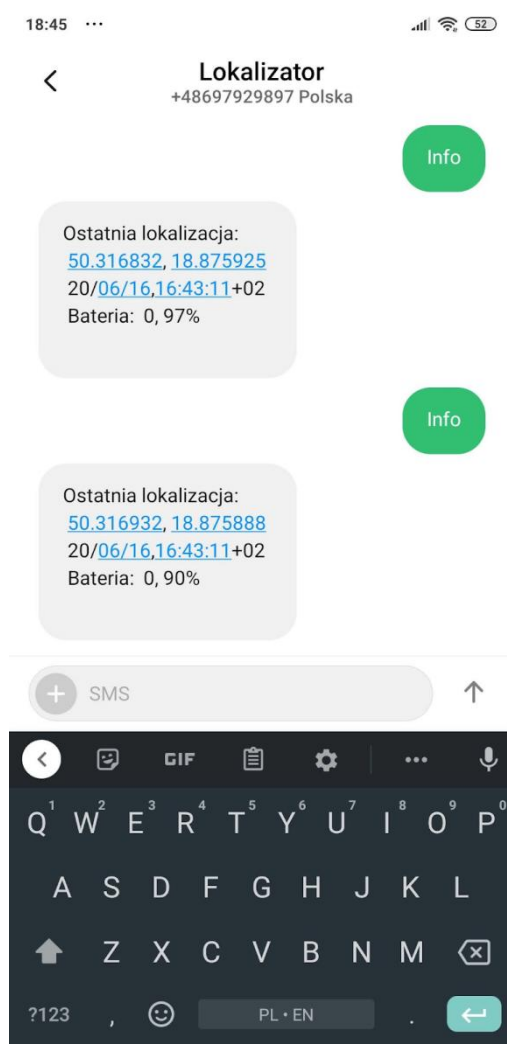
Urządzenie w zamkniętej obudowie

## 14. Jakie problemy wystąpiły podczas montażu i uruchamiania i jak zostały rozwiązane

Po kilku pierwszych testach urządzenie przestało się uruchamiać. Okazało się, iż bateria nie ładuje się i nie daje napięcia. Źródłem problemu okazał się wadliwy styk baterii w koszyku na ogniwo 18650. Problem rozwiązano regulując sprężynę dociskającą ogniwo do złącza.

## 15. Jakie przeprowadzono testy poprawności działania urządzenia

Urządzenie było kilkakrotnie testowane w warunkach przeznaczenia (umieszczone w skrytce w samochodzie osobowym), a także w bardziej wymagających warunkach (umieszczone w plecaku podczas ekstremalnej jazdy na rowerze MTB). Testowano także zdolność do lokalizacji urządzenia w pomieszczeniach zamkniętych.



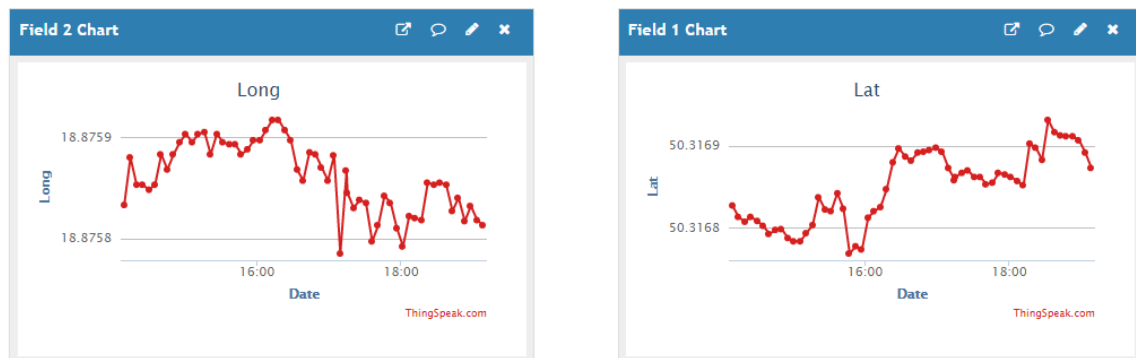
Komunikacja z urządzeniem poprzez wiadomości tekstowe

## 16. Wnioski z uruchamiania i testowania

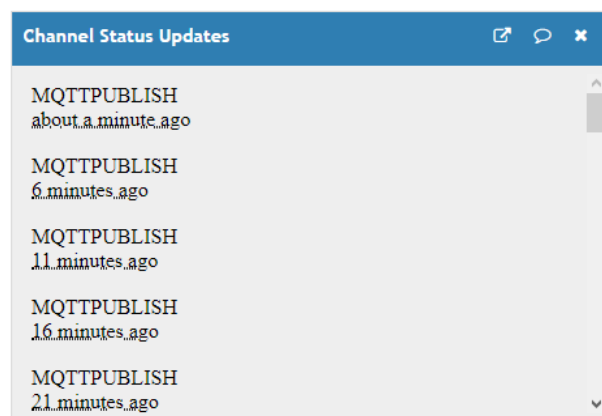
Pierwsze problemy zostały zauważone przy wysyłaniu wiadomości SMS, a także przy komunikacji między modułem ESP8266 a modułem GPRS/GPS - problem objawiał się w postaci przekłamanych wiadomości. Powodowało to liczne utraty połączenia z serwerem MQTT, a także wysyłanie przekłamanych wartości. Problem udało się rozwiązać zmniejszając baudrate komunikacji między modułami (limit programowo realizowanego seriala w ESP8266 to 115200 baud).

Kolejny problem wystąpił przy ostatnich testach, urządzenie nagle przestało odpowiadać na wysyłane komunikaty SMS. Po podłączeniu do konsoli, okazało się, że urządzenie w ogóle nie otrzymuje wysyłanych SMSów z powodu przepełnionej pamięci na wiadomości tekstowe. Problem rozwiązano implementując systematyczne czyszczenie pamięci na wiadomości tekstowe.

Okazało się również, że urządzenie często jest w stanie zlokalizować się w pomieszczeniach zamkniętych, szczególnie jeśli wcześniej miało dostęp do satelitu GPS.



Dane odbierane przez broker MQTT



Okresowość danych odbieranych przez broker MQTT

## 17. Literatura

1. <https://github.com/IOT-MCU/ESP-12S-A9-A9G-GPRS-Node-v1.0/wiki>
2. [https://www.espressif.com/sites/default/files/documentation/esp8266-technical\\_reference\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf)
3. <https://platformio.org/lib/show/306/ESP%20Async%20WebServer>