Energy-Meter

Urządzenie do pomiaru poboru mocy

Operating systems for embedded systems

Autorzy:
Krzysztof Półchłopek
Michał Tomacha

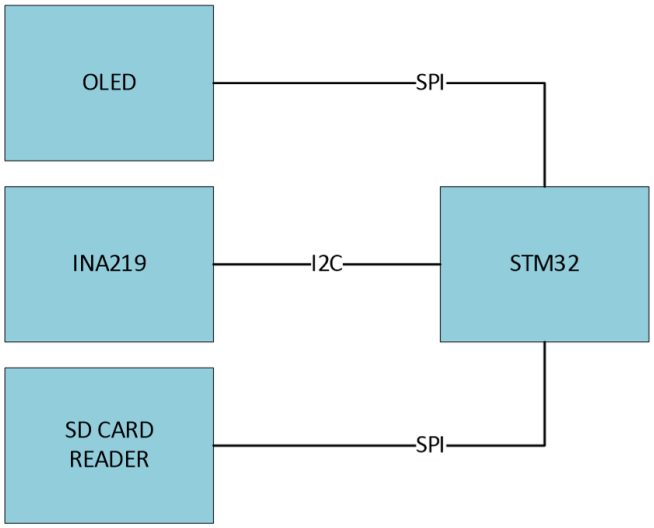
Github: https://github.com/krzpch/Energy-Meter

1. Założenia projektowe

Celem tego projektu było stworzenie aplikacji do wykonywania pomiarów poboru mocy podłączonego urządzenia. Urządzenie wykorzystuje czujnik INA219 do wykonywania potrzebnych pomiarów a następnie wyświetla te pomiary na wyświetlaczu. Istnieje również możliwość zapisywania tych pomiarów na kartę SD.

Główną częścią całego projektu jest mikrokontroler STM32L432KC na którym uruchomiony został system operacyjny FreeRTOS z warstwą API CMSIS.

Poniżej przedstawiony został schemat blokowy urządzenia:



2. Wykorzystane narzędzia

Do wykonania tego projektu wykorzystane zostały następujące narzędzia programistyczne:

- Visual Studio Code – uniwersalny edytor kodu,
- Docker – kontenery w celu ujednolicenia środowiska programistycznego pomiędzy deweloperami,
- Cmake - zarządzanie procesem kompilacji kodu na mikrokontroler,
- Git, GitHub - systemy kontroli wersji dla tego projektu,
- STM32CubeMX – wstępna konfiguracja mikrokontrolera.

Ponadto projekt przygotowany został do obsługi testów w środowisku Google Test. Jednak z powodu ograniczeń czasowych nie byliśmy w stanie ich wykorzystać.

W kontenerze Dockera instalowane są kompilatory ARM zarówno dla języka C jak i C++, make, CMake oraz kompilator C i C++ na platformy x86 w celu umożliwienia pisania testów.

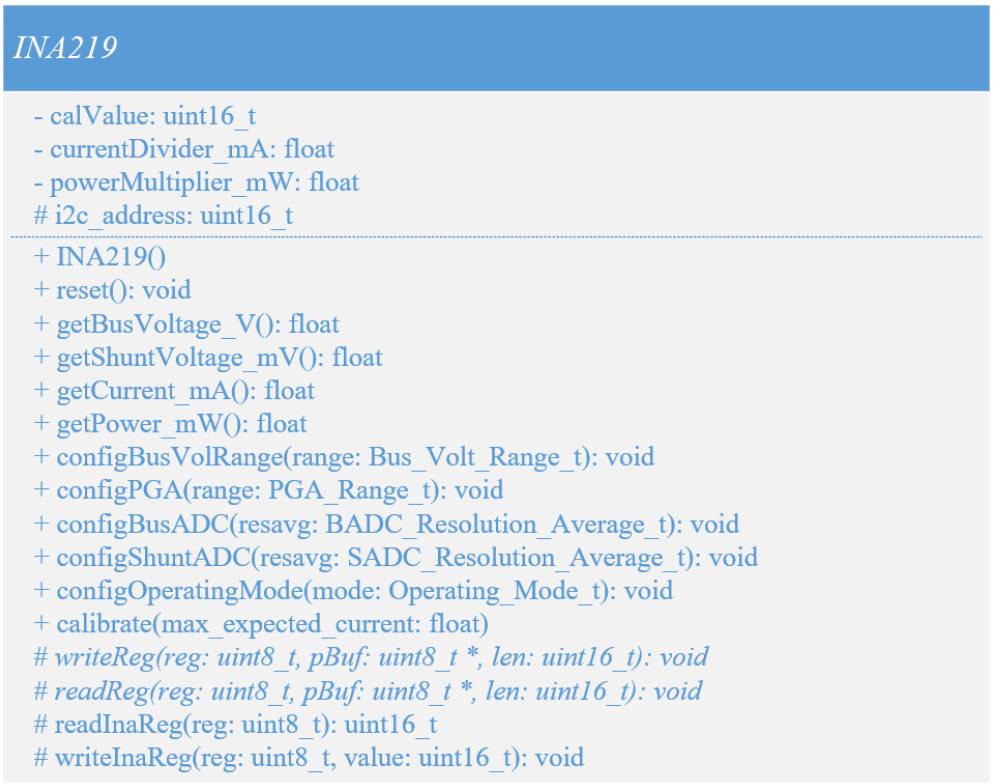
3. Oprogramowanie

Projekt został podzielony na dwie części. Pierwszą z nich są biblioteki konieczne do obsługi peryferiów. Składają się na nie poniższe elementy:

3.1. Biblioteki

3.1.1. Biblioteka do obsługi czujnika INA219

Biblioteka obsługi czujnika została napisana w języku C++. Główna funkcjonalność opiera się na abstrakcyjnej klasie INA219. Dzięki takiemu rozwiązaniu użytkownik może samemu zaimplementować metody do komunikacji I2C.



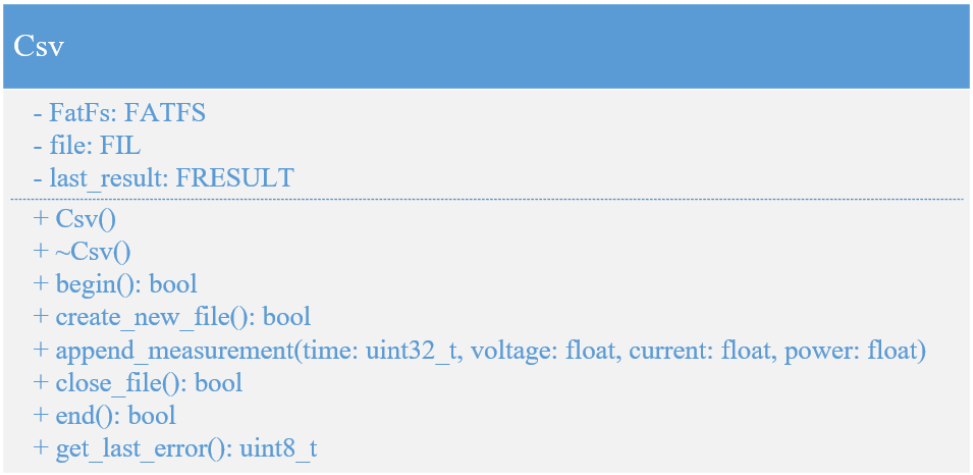
Klasa ta posiada metody odpowiedzialne za inicjalizację czujnika, jego konfigurację oraz odczyt i kalibrację zmierzonych wartości:

- INA219() - Konstruktor ustawiający adres I2C czujnika,
- reset() - Przywraca czujnik do domyślnej konfiguracji,
- getBusVoltage_V() - Odczytuje mierzone napięcie,
- getShuntVoltage_mV(vid) - Odczytuje napięcie na rezystorze pomiarowym,
- getCurrent_mA() - Odczytuje zmierzony pobór prądu,
- getPower_mW() - Odczytuje zmierzony pobór mocy,

- configBusVolRange() - Konfuguruje zakres mierzonego napięcia,
- configPGA() - Konfiguruje wzmocnienie wbudowanego wzmacniacza,
- configBusADC() - Konfiguruje ADC do pomiaru napięcia,
- configShuntADC() - Konfiguruje ADC do pomiaru napięcia na rezystorze pomiarowym,
- configOperatingMode() - Konfiguruje tryby pracy sensora,
- calibrate() - Kalibruje pomiar poboru prądu i mocy,
- readInaReg() - Odczytuje rejestr sensora,
- writeInaReg() - Zapisuje do rejestru sensora,
- writeReg() - W pełni wirtualna metoda do zapisu do rejestru sensora (zależna od sprzętu),
- readReg() - W pełni wirtualna metoda do odczytu rejestru sensora (zależna od sprzętu).

3.1.2. Biblioteka do obsługi plików CSV

Biblioteka obsługi plików została napisana w języku C++. Opiera się na bibliotece FatFs, która zarządza systemem plików na karcie SD, komunikując się z nią poprzez SPI. Główna funkcjonalność opiera się na klasie Csv.



Klasa ta posiada metody odpowiedzialne za inicjalizację systemu plików na karcie SD, tworzenie nowych plików i zapisywanie zmierzonych wartości:

- Csv() - Konstruktor klasy, alokuje pamięć dla systemu plików,
- ~Csv() - Destruktor klasy, uwalnia zaalokowaną pamięć,
- begin() - Inicjalizuje system plików na karcie SD,
- create_new_file() - Tworzy nowy plik csv na karcie SD,
- append_measurement() - Dopisuje nowy pomiar do otwartego pliku,
- close_file() - Zamyka otwarty plik i zapisuje pomiary,
- end() - Deinicjalizuje system plików na karcie SD,
- get_last_error() - Zwraca ostatni błąd systemu plików.

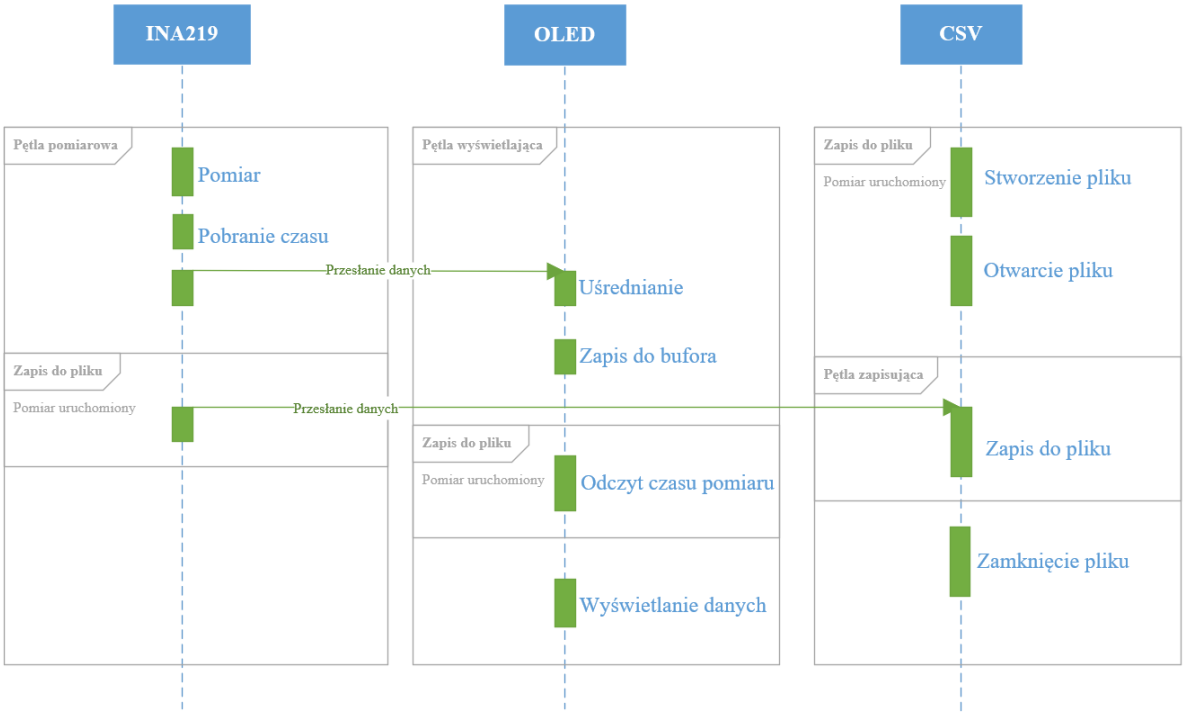
3.1.3. Biblioteka do obsługi wyświetlacza OLED

Biblioteka ta składa się z klasy DISPLAY, która dziedziczy po klasie U8g2 autorstwa użytkownika olikraus. Rodzic zapewnia wszystkie metody konieczne do sterowania wyświetlaczem. Jest ona przystosowana do obsługi bardzo dużej ilości sterowników. W celu uproszczenia inicjalizacji urządzenia odpowiednią funkcją inicjalizacyjną odpowiednią dla sterownika wyświetlacza, w klasie córce konstruktor odpowiada wyłącznie za stworzenie odpowiedniego obiektu na potrzeby tego projektu.

3.2. Funkcje wątków

Drugą częścią tego projektu były funkcje zarządzające poszczególnymi wątkami projektu. Zostały one podzielone na obsługę każdego z peryferiów osobno. Komunikacja pomiędzy nimi odbywa się z wykorzystaniem kolejek. Ponadto do uruchamiania zapisu pomiarów wykorzystany został przycisk obsługiwany z wykorzystaniem przerwań.

Poniższy diagram opisuje uproszczony schemat działania poszczególnych wątków wraz z przesyłanymi pomiędzy nimi wiadomościami.

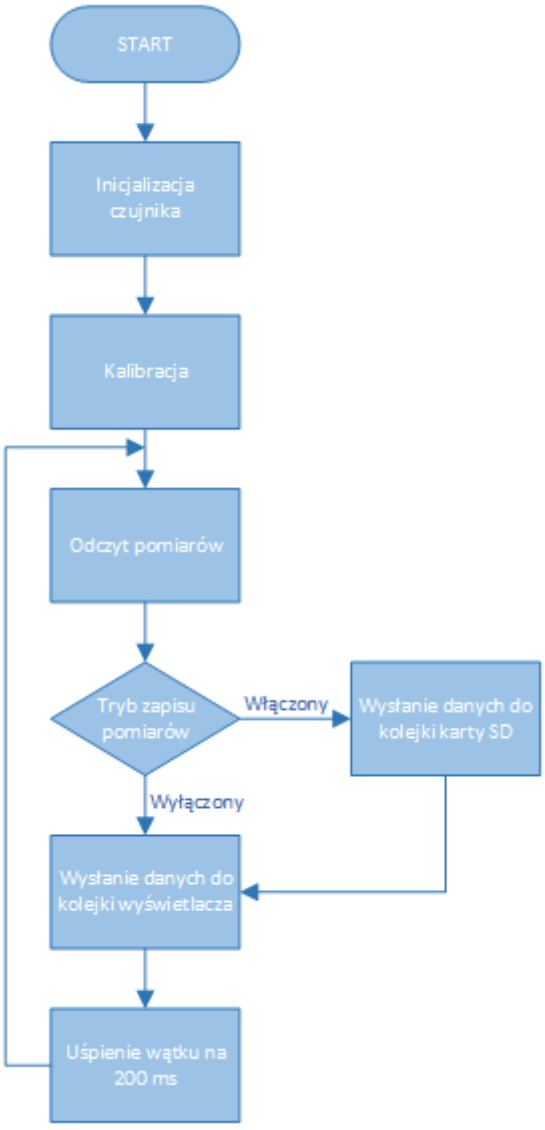


Poniżej przedstawione zostały opisy wraz z diagramami opisującymi działanie wszystkich wątków.

3.2.1. Wątek miernika poboru prądu

Wątek ten odpowiedzialny jest na obsługę miernika INA219. Działanie tego wątku rozpoczyna się poprzez inicjalizację czujnika oraz jego kalibrację. Czujnik ten skalibrowany został do pomiaru maksymalnego prądu o wartości 1 A lecz podczas testów mierzył on poprawnie wartości rzędu kilku mA.

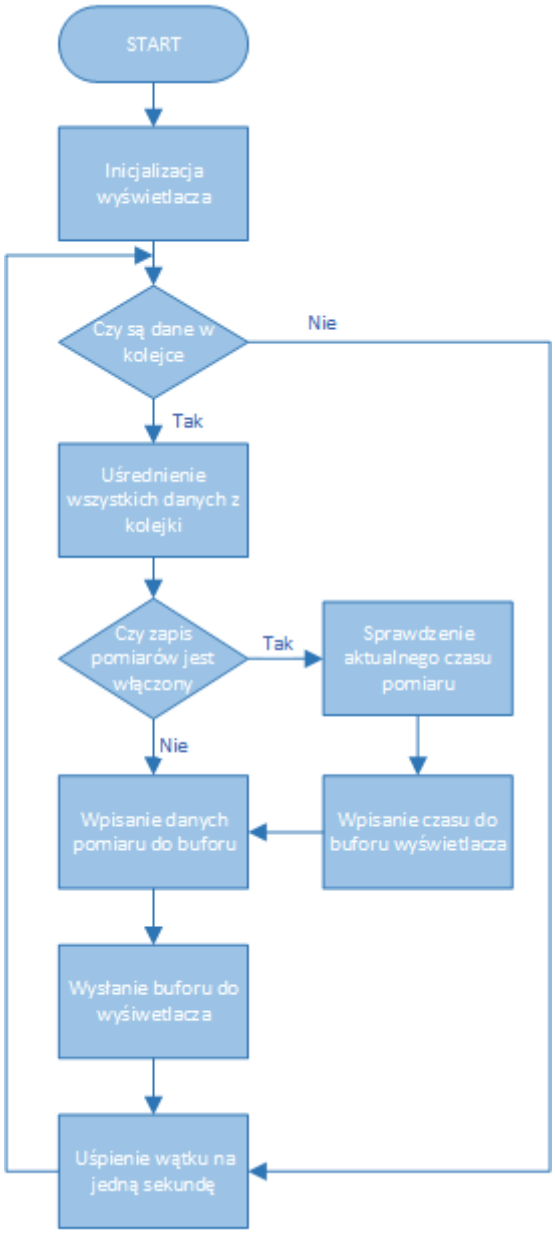
Poniżej przedstawiony został schemat blokowy działania tego wątku.



3.2.2. Wątek wyświetlacza OLED

Wątek ten ma za zadanie wyświetlać na bieżąco aktualne wyniki pomiarów wraz z czasem aktualnie mierzonego pomiaru w przypadku jego uruchomienia. Ponadto wyświetlane są na nim również informacje o problemach z kartą SD.

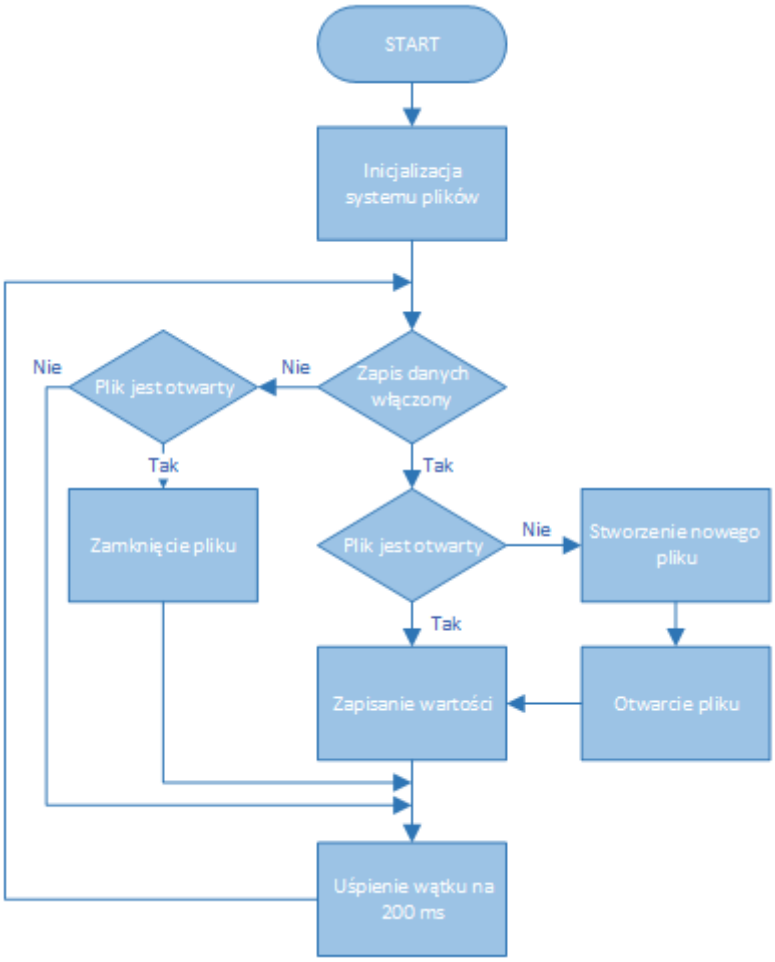
Poniżej przedstawiony został schemat blokowy przedstawiający proces działania tego wątku.



3.2.3. Wątek zapisu danych do pliku

Wątek ten odpowiedzialny jest za zapis aktualnie przeprowadzanych pomiarów na kartę SD. Zapis ten rozpoczynany jest naciśnięciem przycisku. Proces zapisu rozpoczyna się od stworzenia nowego pliku CSV. W pliku zapisywane są timestamp aktualnego pomiaru, napięcie, prąd oraz moc. Pozwala to na późniejszą analizę danych. W celu zakończenia zapisu należy ponownie wcisnąć przycisk. W przypadku gdy program napotka przeszkodę w obsłudze karty SD, na wyświetlaczu pojawia się informacja o błędzie.

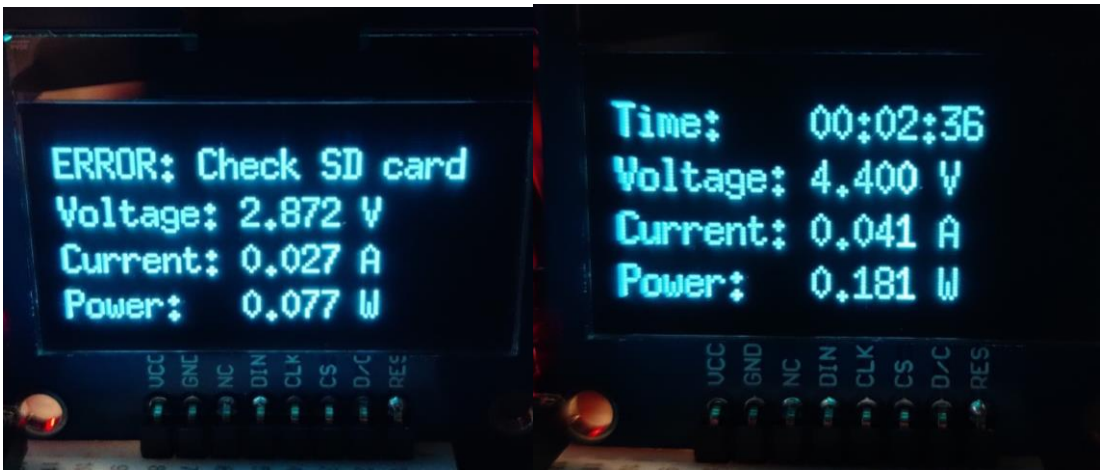
Poniżej przedstawiony został schemat blokowy przedstawiający proces działania tego wątku.



4. Prezentacja działania

Poniżej przedstawione zostały zdjęcia działającego projektu. Na lewym zdjęciu zauważyć można, wyświetlający się błąd karty SD. Najczęstszą przyczyną tego błędu jest brak karty SD w momencie próby rozpoczęcia zapisu pomiarów.

Zdjęcie po prawej przedstawia działanie układu w trybie zapisywania pomiarów.



Poniżej przedstawiona została zawartość przykładowego pliku z pomiarami.

	A	B	C
1	time,voltage,current,power		
2	0,4.076000,0.037000,0.150812		
3	67,4.084000,0.041000,0.167444		
4	201,4.084000,0.038000,0.155192		
5	268,4.084000,0.038000,0.155192		
6	402,4.080000,0.038000,0.155040		
7	469,4.084000,0.039000,0.159276		
8	603,4.080000,0.040000,0.163200		
9	670,4.084000,0.040000,0.163360		