Akademia Górniczo-Hutnicza
im. Stanisława Staszica
w Krakowie.
Wydział Informatyki, Elektroniki i Telekomunikacji

Tomasz Bednorz

Operating Systems for Embedded Systems

Raport z projektu

5 rok, Systemy wbudowane, Elektronika i telekomunikacja

Temat:

Monitor zużycia energii modułu embedded

Spis treści

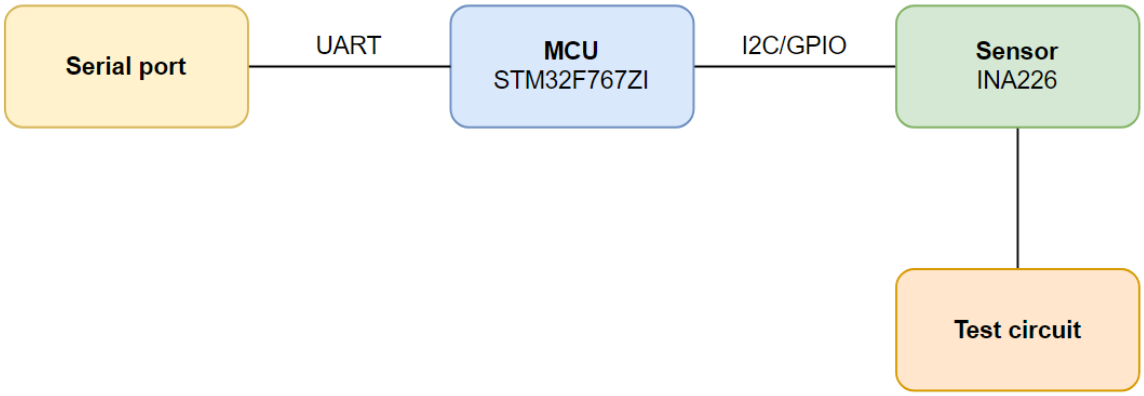
1. Opis projektu

1.1 Cel projektu

Celem projektu jest stworzenie aplikacji umożliwiającej monitorowanie zużycia energii modułu embedded. Poniżej przedstawione są funkcjonalności zrealizowanego systemu:

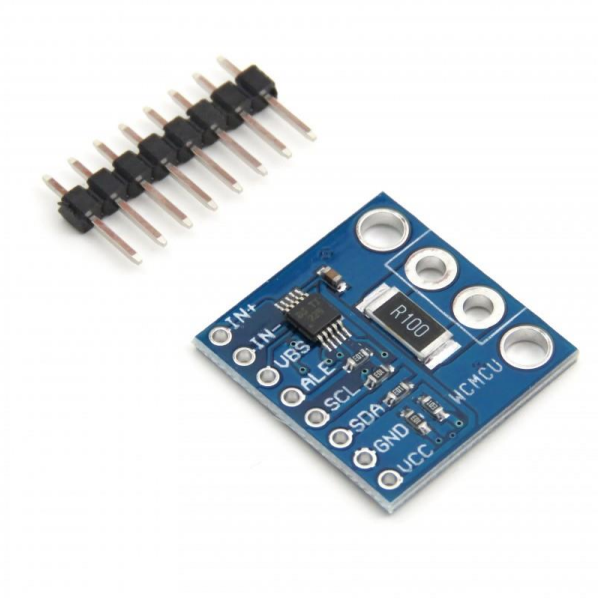
- odbiór danych z urządzenia będącego monitorem prądu/napięcia/mocy za pomocą mikrokontrolera,
- przetworzenie odebranych danych,
- transmisja przetworzonych informacji poprzez port szeregowy.

1.2 Hardware



Rys 1. Schemat blokowy warstwy sprzętowej

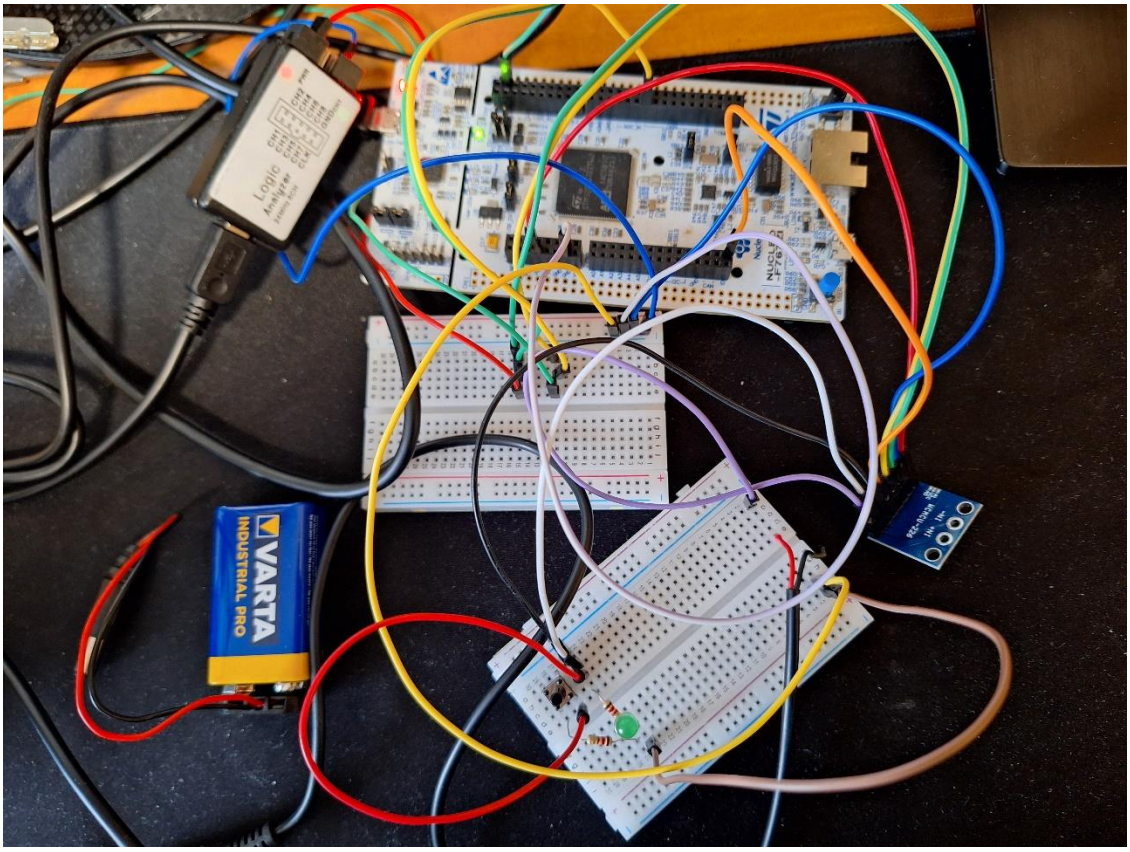
Głównym układem odpowiedzialnym za komunikację oraz przetwarzanie danych jest płytka rozwojowa NUCLEO z mikrokontrolerem STM32F767ZI.
Komunikacja z sensorem INA226 odbywa się za pomocą magistrali I2C oraz GPIO odbierającego informacje z pinu ALERT.
Moduł INA226 połączony jest z obwodem testowym z którego pobierane są dane.
Układ mikroprocesorowy przetwarza uzyskane informacje, takie jak: napięcie, prąd oraz moc, a następnie dokonuje dalszej transmisji poprzez port szeregowy.



Rys 2. Czujnik INA226

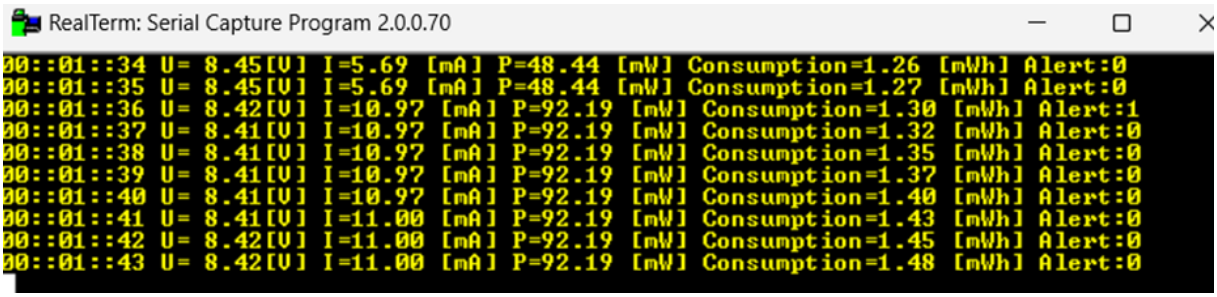
Zastosowany układ przedstawiony na rysunku 2 pozwala na pomiar z wysoką dokładnością napięcia do 36V oraz wysokich wartości prądu – w zależności od zastosowanego rezystora mocy – w używanym module jest to 0.1 Ohm ‘a, więc potencjalny maksymalny prąd mógłby wynosić 360A.

1.3 Testy systemu



Rys 3. Układ testowy

Testy systemu przeprowadzono z użyciem baterii 9V, diody LED oraz dwóch rezystorów 1.2k, gdzie jeden był ciągle podłączony pomiędzy zasilaniem, a diodą, a drugi był dołączany równolegle za pomocą aktywacji przycisku – umożliwiło to wzrost poboru prądu oraz mocy. W pliku konfiguracyjnym został ustawiony próg mocy na 80mW, którego przekroczenie powoduje wystawienie stanu niskiego na pin ALERT poprzez moduł INA226.



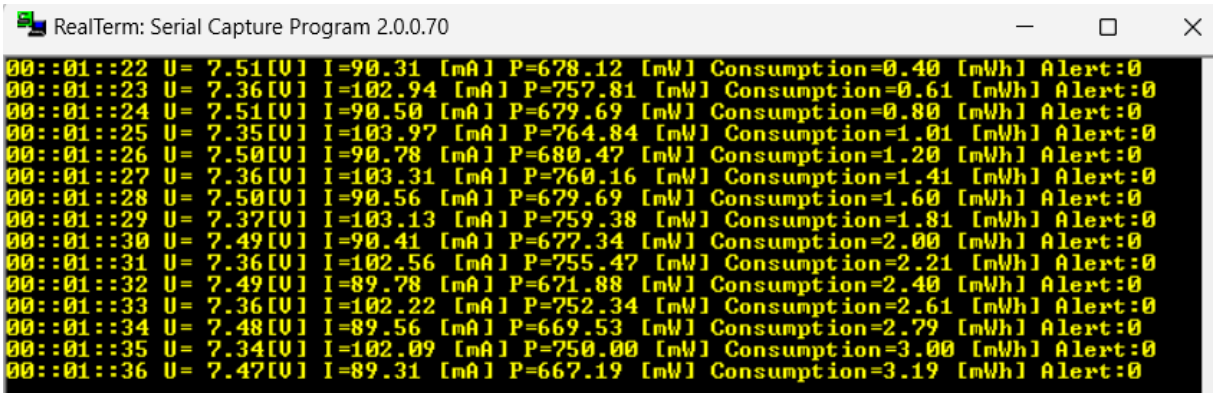
Rys 4. Transmisja poprzez port szeregowy cz.1

Rysunek 4 przedstawia dane transmitowane poprzez port szeregowy. Sa to:

- czas od uruchomienia systemu, format hh::mm::ss,
- napięcie w V,
- pobór prądu w mA,
- moc w mW,
- zużycie energii w mWh (miliwatogodziny)
- informacja o aktywacji alertu.

Na rysunku można zauważyć efekt aktywacji przycisku. Wzrasta pobór prądu, czego efektem jest większa moc, która po przekroczeniu 80 mW spowodowała, że nastąpiła aktywacja alertu. Zauważalny jest również wzrost zużycia energii. Dodatkowo następuje aktywacja diody LED na płytce rozwojowej NUCLEO.

Aby dokonywać pomiaru zużycia energii, np. innego systemu mikroprocesorowego wystarczy wpiąć piny IN+ oraz IN- modułu INA226 pomiędzy pin zasilający badany układ, a zasilanie – pozwoli to na pomiar spadku napięcia na rezystorze pomiarowym.

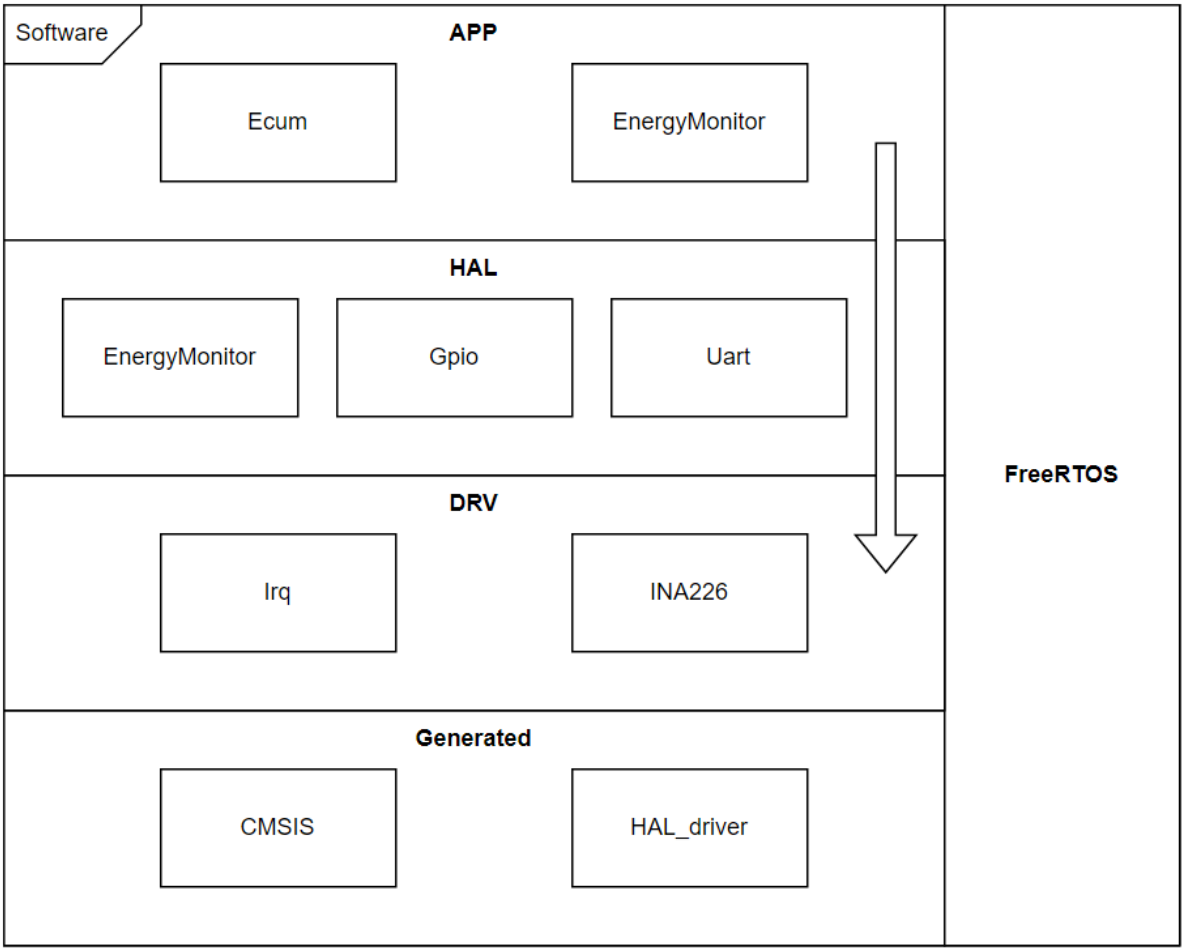


Rys 5. Transmisja poprzez port szeregowy cz.2

Rysunek 5 prezentuje pomiar poboru energii innej płytki NUCLEO, która miga diodą.

2. Oprogramowanie

2.1 Architektura oprogramowania



Rys 6. Schemat blokowy architektury oprogramowania

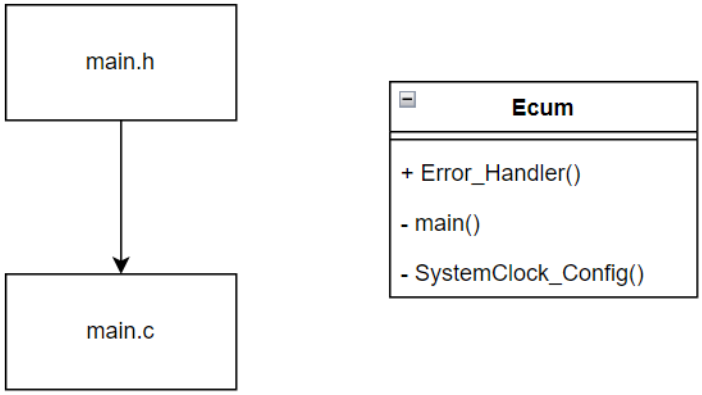
Rysunek 6 przedstawia schemat blokowy architektury oprogramowania na platformę wbudowaną. Aplikacja posiada cztery warstwy:
- warstwę aplikacji,
- warstwę abstrakcji sprzętu,
- sterowniki,
- warstwa wygenerowanego oprogramowania.

Oprogramowanie powinno być możliwe do przeportowania na inne urządzenia, ponieważ:
- zastosowany został podział na warstwy oprogramowania – warstwy niższe nie zaciągają plików z warstw wyższych,
- moduły posiadają pliki konfiguracyjne.

2.2 Szczegółowy opis modułów

2.2.1 APP

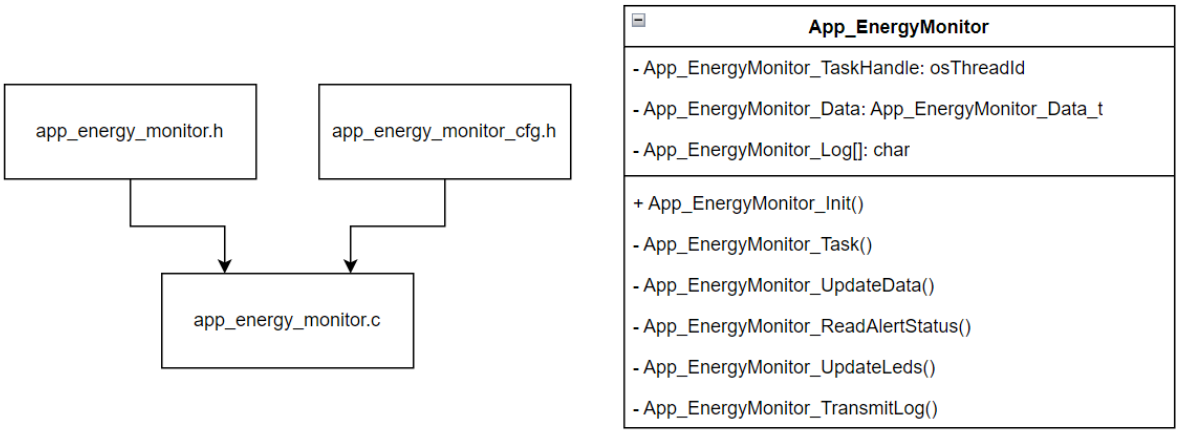
2.2.1.1 Ecum



Rys 7. Struktura oraz interfejsy (+: publiczne, -: prywatne) modułu Ecum

Moduł Ecum odpowiada za inicjalizację peryferiów, zegarów, systemu operacyjnego oraz modułów z wszystkich warstw oprogramowania. Zawiera funkcję main().

2.2.1.2 EnergyMonitor

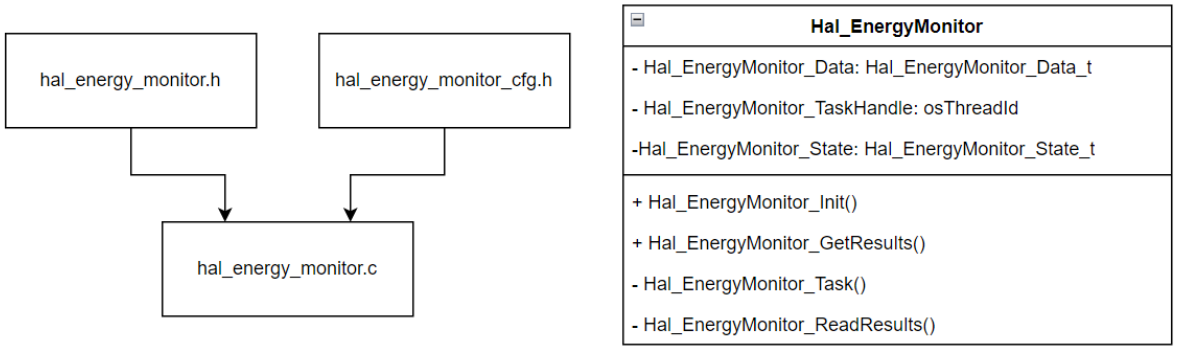


Rys 8. Struktura oraz interfejsy modułu EnergyMonitor

Moduł EnergyMonitor w warstwie aplikacji odpowiada za:
- pobieranie danych z niższych warstw oprogramowania,
- włączanie/wyłączanie diod LED,
- transmisja danych poprzez port szeregowy.

2.2.2 HAL

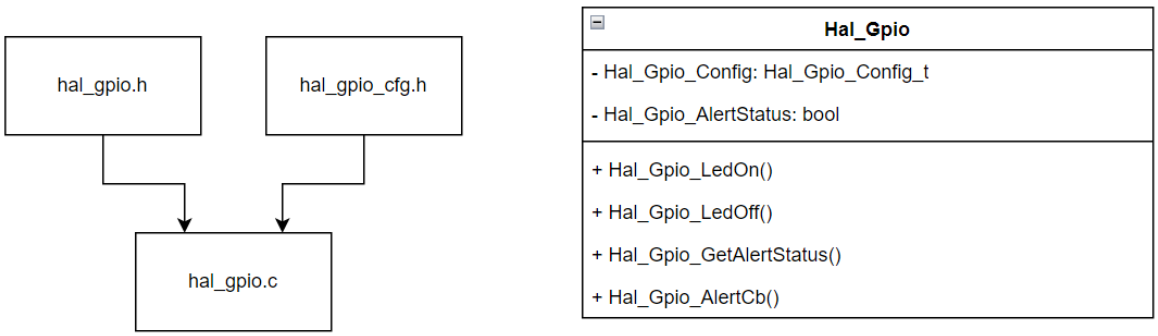
2.2.2.1 EnergyMonitor



Rys 9. Struktura oraz interfejsy modułu EnergyMonitor

Moduł EnergyMonitor w warstwie HAL odpowiada za cykliczny odbiór danych z sensora INA226. Jest warstwą abstrakcji dla drivera.

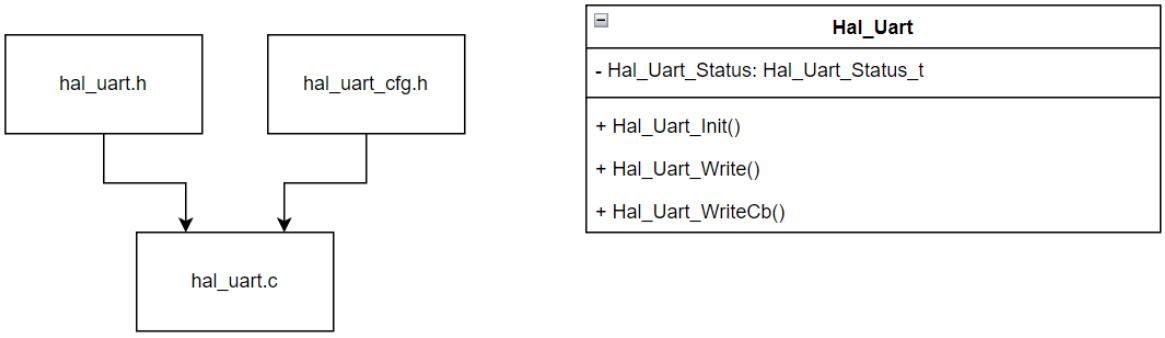
2.2.2.2 Gpio



Rys 10. Struktura oraz interfejsy modułu Gpio

Moduł Gpio odpowiada za sterowanie diodami LED oraz obsługą pinu alert sensora INA226. Jest warstwą abstrakcji dla drivera.

2.2.2.3 Uart

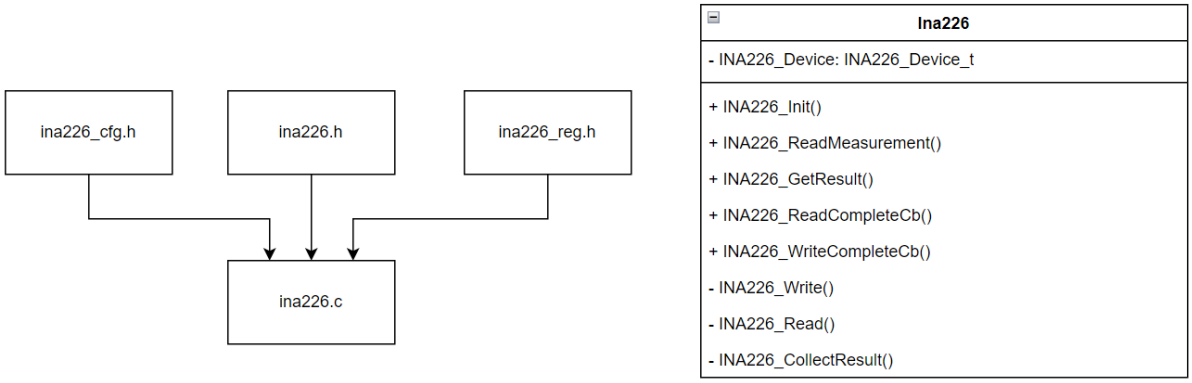


Rys 11. Struktura oraz interfejsy modułu Uart

Moduł Uart odpowiada za transmisję danych poprzez port szeregowy. Jest warstwą abstrakcji dla drivera.

2.2.3 DRV

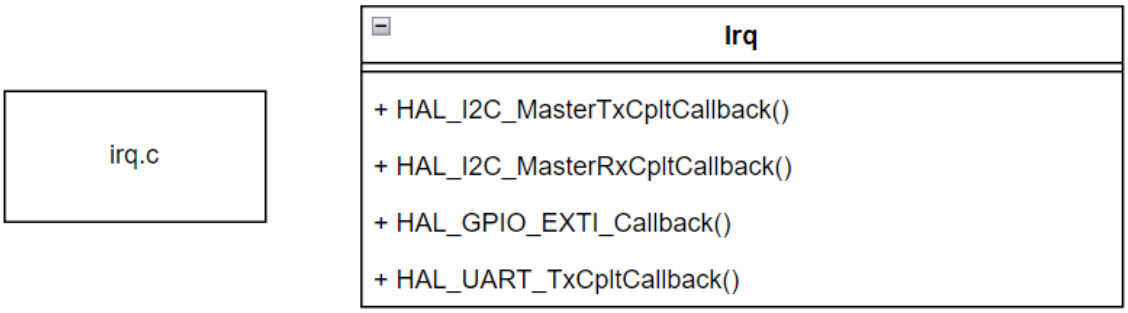
2.2.3.1 INA26



Rys 12. Struktura oraz interfejsy modułu INA226

Driver INA226 odpowiedzialny jest za komunikację z sensorem. Plik konfiguracyjny pozwala podpiąć odpowiednie funkcje transmisyjne oraz dokonać konfiguracji sensora zgodnie z wymaganiami użytkownika.

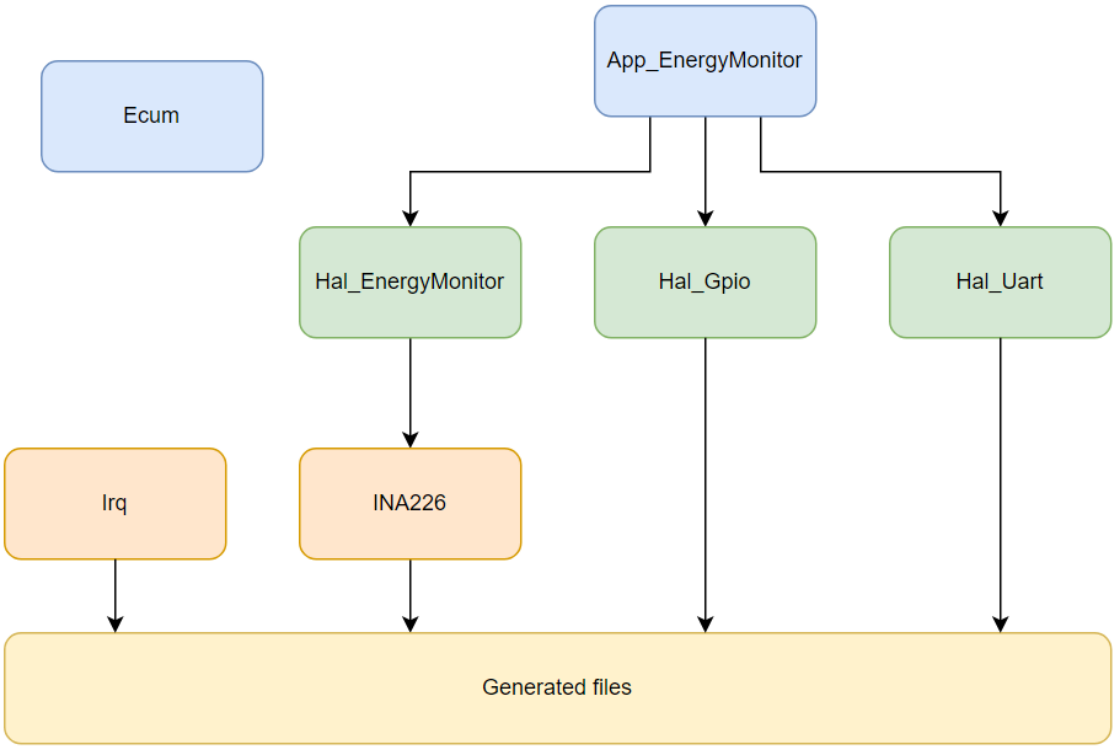
2.2.3.2 Irq



Rys 13. Struktura oraz interfejsy modułu Irq

Moduł Irq odpowiedzialny jest za obsługę callback'ów przerwań.

2.3 Przepływ informacji



Rys 13. Interakcja pomiędzy poszczególnymi modułami oprogramowania

2.4 Kod źródłowy

Github: https://github.com/TomaszBednorz/energy_monitor