

Dokumentacja do projektu

Stacja pogodowa oparta na sensorze BME280

Mikołaj Zielniok
Damian Kozikowski

Prowadzący: **mgr inż. Jakub Zimnol**

1 Cel projektu

Celem projektu jest stworzenie stacji pogodowej, a dokładniej utworzenie sterowników do sensora BME280 podłączonego do mikroprocesora ESP32. W tym celu użyto protokołów SPI i I2C do komunikacji między sensorem, a mikrokontrolerem.

2 Struktura projektu

Projekt został podzielony na 2 foldery:

-components - zawierający pliki źródłowe i nagłówkowe do protokołów:

- BME280
- SPI
- I2C
- GPIO
- common

-main - zawierający plik główny:

- main.cpp - zawiera inicjalizację wyżej wspomnianych protokołów

3 Opis stworzonych bibliotek

- BME280 BME280 to czujnik środowiskowy, który mierzy temperaturę, wilgotność względną i ciśnienie atmosferyczne. W projekcie czujnik wykorzystuje protokół SPI do komunikacji z ESP32.

Cechy:

- Zakres temperatur: -40°C do +85°C.
- Zakres wilgotności: 0
- Zakres ciśnienia: 300 hPa do 1100 hPa.
- Możliwość pracy z interfejsami SPI lub I2C.
- SPI (Serial Peripheral Interface) SPI to synchroniczny protokół komunikacyjny stosowany w projekcie do wymiany danych między ESP32 a BME280.

Linie magistrali SPI:

- SCK (Serial Clock): Zegar synchronizujący transmisję.
- MOSI (Master Out Slave In): Dane przesyłane z ESP32 do BME280.
- MISO (Master In Slave Out): Dane przesyłane z BME280 do ESP32.
- CS (Chip Select): Wybór aktywnego urządzenia.

- GPIO (General Purpose Input/Output)

GPIO w ESP32 jest używane do przypisania fizycznych pinów dla interfejsu SPI oraz do obsługi dodatkowych urządzeń.

W projekcie:

Piny SPI:

- GPIO_MISO: Dane z czujnika do ESP32.
- GPIO_MOSI: Dane z ESP32 do czujnika.
- GPIO_SCK: Zegar SPI.
- GPIO_CS: Wybór urządzenia na magistrali. Możliwość konfiguracji GPIO do innych zastosowań, np. obsługi diod LED.

Przykład przypisania pinów w kodzie:

- `#define GPIO_MISO 19`
- `#define GPIO_MOSI 23`
- `#define GPIO_SCK 18`
- `#define GPIO_CS 5`

GPIO w ESP32 jest elastyczne, co pozwala na łatwą integrację dodatkowych funkcji w projekcie.

- common

Klasa ComProtocol zawarta w folderze common pełni rolę interfejsu dla komunikacji z urządzeniami, takimi jak czujniki lub moduły peryferyjne. Definiuje zestaw funkcji wirtualnych, które muszą być zaimplementowane w klasach pochodnych. Metoda write służy do zapisywania danych w określonym rejestrze urządzenia, a metoda read pozwala na odczyt pojedynczego bajtu danych. Dodatkowo, readWord umożliwia odczyt większej ilości danych, na przykład 16-bitowych wartości. Klasa zawiera także wskaźnik rx_buf, który może być wykorzystywany do przechowywania odebranych danych. Dzięki wirtualnym funkcjom klasa pozwala na stworzenie uniwersalnego rozwiązania, które można dostosować do różnych protokołów komunikacyjnych, takich jak SPI czy I2C.