



AKADEMIA GÓRNICZO-HUTNICZA

Dokumentacja do projektu

**Generyczna biblioteka do protokołów
komunikacyjnych mikrokontrolera
AVR**

z przedmiotu

Języki programowania obiektowego

EiT rok III

Miłosz Mynarczuk

piątek 13.15

prowadzący: Jakub Zimnol

9.01.2025

1. Opis i zastosowanie projektu

Celem projektu było stworzenie biblioteki do protokołów komunikacyjnych AVR, którą można wykorzystać do komunikacji mikrokontrolera z innymi urządzeniami. Obsługiwane są 3 standardy komunikacji: I2C, SPI i UART. Każdy protokół pozwala na ustawianie różnych parametrów danych standardów komunikacji a także na zapisy i odczyty danych. W projekcie zawarta jest także obsługa czujnika BME280 protokołem I2C, a dane z tego czujnika wysyłane są za pomocą UART.

2. Struktura projektu:

W skład projektu wchodzi:

CommunicationProtocol.h – zawiera się w nim klasa bazowa

SPI.h SPI.cpp – plik nagłówkowy i implementacyjny odpowiadający za SPI

I2C.h I2C.cpp – plik nagłówkowy i implementacyjny odpowiadający za I2C

UART.h UART.cpp – plik nagłówkowy i implementacyjny odpowiadający za UART

Communication.h – plik zbiorczy

3. Zastosowane klasy i funkcje

CommunicationProtocol:

Jest to klasa bazowa to "CommunicationProtocol", zwarta w niej jest funkcja wirtualna init(), po tej klasie dziedziczą trzy klasy:

SPI:

Wewnątrz tej klasy konstruktor pozwala na ustawienie pinów do komunikacji, co pozwala na uniwersalność na różnych modelach mikrokontrolerów AVR, w tym konstruktorze można także ustawić czy urządzenie ma działać jako master albo slave. Posiada również funkcje odpowiedzialne za odczyt i zapis bajtów, a także odczyt i zapis danych z rejestrów.

I2C:

Wewnątrz tej klasy konstruktor pozwala na ustawienie prędkości zegara oraz prescalingu. Posiada funkcję startu i stopu potrzebnych w tym protokole, które następnie są używane w następnych funkcjach. Podobnie jak w SPI jest tu odczyt i zapis Bajtów, a także danych z rejestrów, z różnicą że jest wzięty pod uwagę adres urządzenia slave. Dodatkowo dodany jest zapis i odczyt ciągu rejestrów.

UART:

Wewnątrz tej klasy konstruktor pozwala na ustawienie którego interfejsu USART chcemy używać oraz prędkości komunikacji. Posiada funkcję do przesyłania i odczytywania bajtu, a także do przesyłania i odczytywania stringa.