



**AKADEMIA GÓRNICZO-HUTNICZA**

**AGH**

Dokumentacja do projektu

# **Biblioteka generyczna do protokołów komunikacyjnych**

z przedmiotu

**Języki Programowania Obiektowego**

EiT 3 rok

*Antonia Zdziebko*

Środa 13:15

prowadzący: Jakub Zimnol

13.01.2026

## **1. Opis projektu**

Projekt jest biblioteką generyczną do obsługi protokołów komunikacyjnych: UART, I2C i SPI. Została zaprojektowana tak, aby była niezależna od mikrokontrolera. Wykorzystano projektowanie obiektowe, bazując na abstrakcyjnej klasie bazowej Protocol.

Celem projektu jest wspólny interfejs dla różnych protokołów komunikacyjnych oraz oddzielenie logiki protokołów od warstwy sprzętowej (aby była ogólna i uniwersalna). Dzięki temu rozbudowa projektu jest łatwiejsza, a chęć wykorzystania biblioteki w pełnej sprawności sprzętowej można osiągnąć pisząc odpowiedni adapter do mikrokontrolera.

Działanie kodu, zostało przedstawione na mikrokontrolerze ESP32 w formie demonstracyjnej. Nie wykorzystano pełnej implementacji protokołów, lecz użyto logi pozwalające pokazać zastosowanie biblioteki. Dodatkowo przedstawiają mechanizmy kontrolne i konfiguracyjne, a także zarządzanie stanami. Zachowuje to generyczność biblioteki i pokazuje możliwości po zastosowaniu warstwy sprzętowej.

## **2. Opis klas**

Protocol:

Główną klasą abstrakcyjną jest klasa Protocol. Stanowi ona podstawę do konkretnych protokołów komunikacyjnych. Znajdują się w niej podstawowe typy, które muszą należeć do klas pochodnych. Zawiera metody send i recv, czyli tylko to co powinno być ale czemu? Czemu tu nie init?

UART:

Klasa UART jest klasą dziedziczącą po Protocol. Jest implementacją samego protokołu UART. Zawiera pola konfiguracyjne typowe dla niej jak np. odpowiednie piny. Wykorzystuje zarządzanie swoim stanem, który został zdefiniowany w Protocol. Nadpisuje metody send i recv, tak aby spełniały założenia przesyłu ramek tak jak w protokole UART.

I2C:

Klasa I2C reprezentuje działanie protokołu komunikacyjnego I2C. Tak samo jak UART dziedzicy po Protocol i nadpisuje te same metody. Zarządza konfiguracją i pinami zgodnie z założeniami samego protokołu.

SPI:

Klasa SPI również jest klasą pochodną i realizuje protokół komunikacyjny SPI. Ma zdefiniowane pola potrzebne do przesyłu danych właśnie tym protokołem. Sprawdza poprawność konfiguracji pól i zarządza stanami obiektu.