



AKADEMIA GÓRNICZO-HUTNICZA

Dokumentacja do projektu

**Biblioteka do obsługi peryferiów
mikrokontrolera (GPIO)**

z przedmiotu

Języki programowania obiektowego

EiT, III rok

Jakub Marszałek

piątek 15:00

prowadzący: Jakub Zimnol

08.01.2025

Wprowadzenie

Projekt ten stanowi bibliotekę obsługi wejść/wyjść GPIO (General Purpose Input/Output) dla mikrokontrolerów AVR. Jego głównym celem jest ułatwienie pracy z portami mikrokontrolera poprzez udostępnienie obiektowego interfejsu pozwalającego na wykonywanie operacji takich jak:

- Ustawianie kierunku pracy pinu (wejście/wyjście).
- Odczyt stanu logicznego pinu.
- Zapis stanu logicznego pinu.
- Włączanie rezystora podciągającego (pull-up).
- Eliminacja drgań styków (debouncing).
- Generowanie sygnałów PWM na wybranych pinach.
- Miganie diodą LED z określoną częstotliwością.

Biblioteka została zaprojektowana w oparciu o podejście obiektowe, co pozwala na łatwe rozszerzanie jej funkcjonalności oraz ponowne wykorzystanie kodu w różnych projektach.

Struktura projektu

Projekt składa się z trzech głównych klas:

1. Klasa **GPIO**

Jest to klasa bazowa definiująca interfejs dla obsługi GPIO. Zawiera virtualne metody:

- `void setDirection(bool inOut)` – Ustawia kierunek pracy pinu (wejście/wyjście).
- `void write(bool state)` – Zapisuje stan logiczny na pinie.
- `bool read() const` – Odczytuje stan logiczny pinu.

Klasa ta jest abstrakcyjna i służy jako podstawa do implementacji specyficznych funkcji GPIO.

2. Klasa **GPIOPort**

Jest to klasa pochodna, związana z funkcjonalnością portów

Funkcjonalności:

- Wybór odpowiednich rejestrów (DDR, PORT, PIN) na podstawie nazwy portu (B, C, D).

Konstruktor:

- `GPIOPort(char portName, uint8_t pinNr)` – Inicjalizuje obiekt GPIO dla wybranego portu i numeru pinu. W przypadku podania nieprawidłowych danych, program zatrzymuje się w nieskończonej pętli.

3. Klasa **GPIOPin**

Klasa dziedziczy po `GPIOPort` i rozszerza jej funkcjonalności o dodatkowe metody. implementuje funkcjonalności zdefiniowane w klasie `GPIO` związane z pinami

Funkcjonalności:

- `setDirection(bool inOut)` – Ustawia kierunek pracy pinu poprzez modyfikację rejestru DDR.

- `write(bool state)` – Zapisuje stan logiczny na pinie poprzez rejestr PORT.
- `read() const` – Odczytuje stan logiczny pinu z rejestru PIN.
- `void pullUp(bool on)` – Włącza lub wyłącza rezystor podciągający (pull-up).
- `void toggle()` – Zmienia stan logiczny pinu na przeciwny.
- `void blink(uint16_t delay, uint8_t times)` – Powoduje miganie diodą LED z podanym opóźnieniem (w milisekundach) i liczbą powtórzeń.
- `bool debounced()` – Eliminacja drgań styków poprzez odczyt stanu pinu w odstępie czasowym 50 ms.
- `void configurePWM(uint8_t timer, uint8_t fill, char channel, uint8_t prescaler)` – Konfiguruje funkcję PWM na wybranym pinie, z użyciem podanego timera, współczynnika wypełnienia, kanału (A/B) oraz preskalera.

Obsługa PWM:

- Timer 0, 1, 2 obsługiwane w trybach Fast PWM z możliwością konfiguracji kanałów A i B.
- Rejestry OCR, TCCR oraz preskalery są używane do precyzyjnego sterowania sygnałem PWM.

Kluczowe elementy projektu

- **Rejestry AVR:** Klasa `GPIOPort` wykorzystuje rejestry `DDRX`, `PORTX` oraz `PINX`, gdzie `X` oznacza nazwę portu (B, C, D).
- **Debouncing:** Klasa `GPIOPin` implementuje prosty mechanizm eliminacji drgań styków poprzez dwukrotny odczyt stanu pinu z opóźnieniem.
- **PWM:** Klasa `GPIOPin` pozwala na łatwe skonfigurowanie sygnałów PWM na wybranych pinach, co może być wykorzystane np. do sterowania jasnością LED lub prędkością silnika DC.

Zastosowanie

Projekt może być wykorzystany w różnych aplikacjach mikrokontrolerowych, takich jak:

- Sterowanie diodami LED.
- Odczyt stanu przycisków z eliminacją drgań styków.
- Generowanie sygnałów PWM
- Obsługa wejść/wyjść cyfrowych w złożonych systemach embedded.

Podsumowanie

Projekt GPIO dla AVR jest elastycznym i łatwym w użyciu narzędziem dla programistów pracujących z mikrokontrolerami AVR. Obiektowe podejście pozwala na intuicyjne zarządzanie pinami GPIO, a zaawansowane funkcje, takie jak PWM czy debouncing, znacznie rozszerzają możliwości aplikacji. Biblioteka może być łatwo dostosowana do specyficznych potrzeb projektu.