



**Politechnika  
Śląska**

## **PROJEKT INŻYNIERSKI**

Język programowania dla mikrokontrolerów AVR

**Maciej WALERYN**

Nr albumu: 296441

**Kierunek:** Informatyka

**Specjalność:** Grafika komputerowa

**PROWADZĄCY PRACĘ**

**dr hab. inż. Krzysztof Simiński**

**KATEDRA Algorytmiki i Oprogramowania**

**Wydział Automatyki, Elektroniki i Informatyki**

**OPIEKUN, PROMOTOR POMOCNICZY**

**⟨stopień naukowy imię i nazwisko⟩**

**Gliwice 2025**



**Tytuł pracy**

Język programowania dla mikrokontrolerów AVR

**Streszczenie**

Celem dyplomu jest zaprojektowanie języka programowania oraz napisanie kompilatora dla mikrokontrolera w architekturze AVR. Język ten ma spełniać cechę kompletności Turinga, posiadać znane dla współczesnych języków programowania funkcjonalności, systemy zapobiegające powstawaniu błędów w programie (np. *garbage collector*) i podstawowe algorytmy optymalizacji zmniejszające kod wynikowy oraz czas wykonania programów.

**Słowa kluczowe**

język programowania, kompilator, mikrokontrolery AVR

**Thesis title**

Programming language for AVR microcontrollers

**Abstract**

(Thesis abstract – to be copied into an appropriate field during an electronic submission – in English.)

**Key words**

programming language, compiler, AVR microcontrollers



# Spis treści



# Rozdział 1

## Wstęp

Wytwarzanie oprogramowania dla systemów wbudowanych wymaga wiedzy specjalistycznej na wysokim poziomie. W przypadku tworzenia projektów amatorskich, przodującymi technologiami są płytki rozwojowe oparte o mikrokontrolery AVR serii ATMega. Ograniczenia tej architektury, tj. mała ilość pamięci operacyjnej i programowej, skłaniają programistów do korzystania z niskopoziomowych języków oraz bibliotek. Te czynniki są dużym utrudnieniem dla nowicjuszy, co w dużej mierze prowadzi do zniechęcenia do rozwoju w kierunku systemów wbudowanych, wydłuża czas realizacji projektów i zwiększa liczbę powstających błędów w wytwarzanym oprogramowaniu.

Celem tej pracy jest opracowanie języka programowania, posiadającego zalety języków wysokopoziomowych, ułatwiającego pracę z mikrokontrolerami AVR. Docelowym układem, dla którego generowany oraz testowany będzie kod wynikowy, jest mikrokontroler ATMega328, znany szeroko z występowania w płytkach rozwojowych Arduino Uno. Ze względu na rozmiar rodziny mikrokontrolerów AVR, kompilator będzie umożliwiał wprowadzenie parametrów konfiguracyjnych, pozwalając tym samym na wsparcie dla większości członków tej rodziny mikrokontrolerów.

Opis rozdziałów będzie gotowy po bliższym przygotowaniu listy rozdziałów :)

- wprowadzenie w problem/zagadnienie
- osadzenie problemu w dziedzinie
- cel pracy
- zakres pracy
- zwięzła charakterystyka rozdziałów
- jednoznaczne określenie wkładu autora, w przypadku prac wieloosobowych – tabela z autorstwem poszczególnych elementów pracy





# Rozdział 2

## Języki programowania dla mikrokontrolerów AVR

### 2.1 Mikrokontrolery AVR

Platforma mikrokontrolerów AVR, utworzona przez firmy Atmel i aktualnie będąca własnością Microchip, jest jedną z najbardziej popularnych platform wykorzystywanych w budowaniu urządzeń wbudowanych. Ośmio-bitowa architektura tych mikrokontrolerów powstała w roku 1996, na rynek wprowadzona została w roku 1997. Jej najważniejszymi cechami stały się prostota wytwarzania dla niej oprogramowania, niski pobór mocy oraz przystępna cena układów.

Mikrokontrolery AVR zostały oparte na ośmio-bitowym procesorze RISC w zmodyfikowanej architekturze Harvardzkiej z autorskim zestawem instrukcji. W zależności od rodziny, rdzeń procesora może pracować z zakresie częstotliwości 1-20 MHz lub 32 MHz dla rodziny XMEGA. W podstawowej wersji architektury, dyspozycji programisty zostały przekazane:

- pamięć Flash, wykorzystywana jako pamięć programu,
- pamięć SRAM, służąca przechowywaniu zmiennych,
- pamięć EEPROM, umożliwiającą przechowywanie dużych wartości statycznych,
- zbiór rejestrów wewnętrznych kontrolujących pracę mikrokontrolera oraz służących do wykonywania instrukcji,
- rejestry portów wejścia/wyjścia,
- w zależności od modelu: liczniki zegarowe, konwertery analogowo-cyfrowe i cyfrowo-analogowe, sprzętowe interfejsy dedykowane dla protokołów tj. TWI, UART, SPI, USB, Ethernet.