



Instytut Teleinformatyki



Wydział Fizyki, Matematyki i Informatyki
Politechnika Krakowska

Mikroprocesory i Mikrokontrolery

„Dostęp do portów mikrokontrolera ATmega32 – język C”

laboratorium: 10
autorzy: dr hab. Zbislaw Tabor, prof. PK
mgr inż. Paweł Pławiak

Kraków, 2015

Spis treści

Spis treści	2
1. Wiadomości wstępne	3
1.1. Mikrokontroler ATmega32	3
1.2. Listing 1	4
1.3. Programowanie zestawu ZL15AVR	5
1.4. Zagadnienia do przygotowania	6
2. Przebieg laboratorium	8
2.1. Zadanie 1. Na ocenę 3.0 (dst)	8
2.2. Zadanie 2. Na ocenę 4.0 (db)	8
2.3. Zadanie 3. Na ocenę 5.0 (bdb)	8

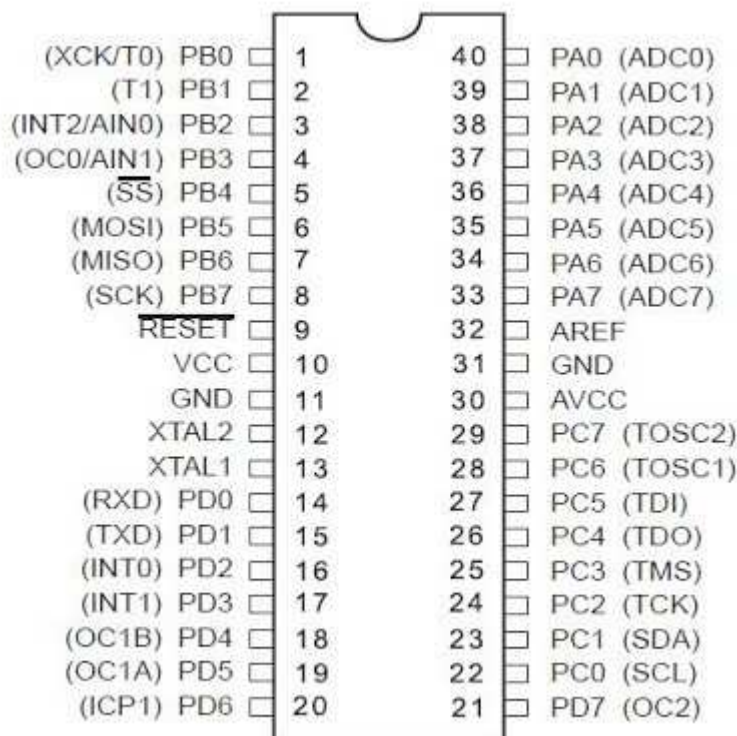
1. Wiadomości wstępne

Pierwsza część niniejszej instrukcji zawiera podstawowe wiadomości teoretyczne dotyczące omawianego tematu. Poznanie tych wiadomości umożliwi prawidłowe zrealizowanie praktycznej części laboratorium.

1.1. Mikrokontroler ATmega32

Szczegółowe informacje o architekturze mikrokontrolera ATmega32 są zawarte w nocie katalogowej na stronie producent – <http://www.atmel.com/Images/doc2503.pdf>

Do obsługi portów mikrokontrolera wykorzystywane są 8-bitowe rejestry PORTA, PORTB, PORTC i PORTD. Każdy pin każdego z portów może pracować w trybie wejścia (np. obsługa klawiatury) lub wyjścia (np. obsługa LED). Kierunek pinu PORTA określa skojarzony z nim bit rejestru DDRA (analogicznie dla pozostałych portów). Ustawiony bit rejestru DDRx oznacza, że skojarzony z nim pin portu pracuje jako wyjście, a wyczyszczony bit konfiguruje pin portu jako wejście. Po resecie wszystkie piny są skonfigurowane jako wejścia i ustawiony jest na nich nieokreślony stan napięcia tzw. "Hi-Z" - dlatego przed użyciem należy ustawić na portach wymagane poziomy napięć 0 lub 1. Jeżeli dany pin portu PORTX jest skonfigurowany jako wejście, jego stan należy odczytywać z wykorzystaniem rejestru PINX. Wyjścia mikrokontrolera Atmega32 mogą pełnić różne funkcje alternatywne (**Rys. 1**), które są aktywowane przez włączanie odpowiednich modułów. Na przykład ustawienie odpowiednich bitów w rejestrze kontrolującym Timer0 powoduje, że na wyjście PB3 (OC0) podawany jest sygnał prostokątny. Z kolei włączenie modułu USART (poprzez ustawienie odpowiednich bitów odpowiednich rejestrów) powoduje, że wyjścia PD1 i PD0 pełnią rolę linii nadawczej i odbiorczej dla transmisji szeregowej.



Rys. 1 Alternatywne funkcje pinów mikrokontrolera ATmega32

1.2. Listing 1

Na **Listingu 1** przedstawiono prosty program wywołujący mruganie diod dołączonych do Portu A mikrokontrolera ATmega32 (zestaw ewaluacyjny **ZL15AVR** - <http://www.kamami.pl/index.php?productID=46782>). Po uruchomieniu **AVR Studio** zarządzanie projektami odbywa się z poziomu menu **Project**. Przy tworzeniu nowego projektu należy wybrać język programowania (C lub assembler) i określić lokalizację plików projektu. Opcje utworzonego projektu można kontrolować wybierając polecenie **Project -> Configuration options**: przede wszystkim na karcie **General** należy określić rodzaj mikrokontrolera (atmega32) i częstotliwość (1MHz). Na karcie tej należy również zaznaczyć opcję **Create Hex File**, ponieważ plik w tym formacie zostanie zapisany w pamięci Flash mikrokontrolera. Karta **Custom options** pozwala wybrać narzędzia budowania (AVR toolchain lub zewnętrzny kompilator i make). Do budowania aplikacji przeznaczono menu **Build** (skrót **F7**). Przedstawiony na Listingu 1 program wywołuje miganie diod zestawu po połączeniu zestawu pinów Con9 z Con15 kablem 1 do 1.

Listing 1:

```
#include <avr/io.h>
#include <util/delay.h>

int main(void)
{
    char i;
    DDRA = 0xFF; // Wszystkie linie wyjściowe

    do{
        PORTA = 0b10101010;
        for(i= 0; i < 25; i++)
            _delay_ms(10);
        PORTA = 0b01010101;
        for(i= 0; i < 25; i++)
            _delay_ms(10);
    }while(1);
    return 0;
}
```

1.3. Programowanie zestawu ZL15AVR

Przed fizycznym zaprogramowaniem mikrokontrolera warto prześledzić jego pracę w trybie debugowania.

Programowanie zestawu ZL15AVR w trybie ISP wymaga zastosowania dedykowanego programatora (np KamPROG - <http://www.kamami.com/index.php?productID=50609>) Programator dostarczany jest z oprogramowaniem - należy najpierw zainstalować oprogramowanie programatora, a następnie wykonać aktualizację firmware sprzętu do wymaganej przez oprogramowanie programatora wersji (1.5). Poprawnie zainstalowane oprogramowanie potrafi rozpoznać wersję firmware i rozpoznać mikrokontroler po podłączeniu zestawu ewaluacyjnego. Ustawienia oprogramowania programatora są przedstawione na **Rys. 2**. Zaprogramowanie kontrolera wymaga wybrania pliku HEX - zapis do pamięci FLASH następuje po kliknięciu przycisku **Write** w sekcji **Flash programming**. Po zainstalowaniu programatora z opcją pluginu dla AVR Studio programowanie może się odbywać z poziomu środowiska AVR Studio (**Tools->Kamami AVR programmer->Connect**).



Rys. 2 Ustawienia programatora KAMprog for AVR v.2.1

1.4. Zagadnienia do przygotowania

Przed przystąpieniem do realizacji laboratorium należy zapoznać się z zagadnieniami dotyczącymi:

- o mikrokontrolera AVR ATmega32
- o zestawu ZL15AVR
- o AVR Studio
- o programatora KAMprog

Literatura:

- [1] Rafał Baranowski, „Mikrokontrolery AVR ATmega w praktyce”
- [2] Tomasz Francuz, „Język C dla mikrokontrolerów AVD od podstaw do zaawansowanych aplikacji”
- [3] Nota katalogowa mikrokontrolera – <http://www.atmel.com/Images/doc2503.pdf>
- [4] Dokumentacja zestawu ewaluacyjnego ZL15AVR – http://dl.btc.pl/kamami_wa/zl15avr.pdf
- [5] AVR Studio - http://www.atmel.com/microsite/avr_studio_5/
- [6] KAMprog - http://dl.btc.pl/kamami_com/kamprogavr_en.pdf

2. Przebieg laboratorium

Druga część instrukcji zawiera zadania do praktycznej realizacji, które demonstrują zastosowanie technik z omawianego zagadnienia.

2.1. Zadanie 1. Na ocenę 3.0 (dst)

Proszę napisać program, który włącza i wyłącza diodę D0 klawiszem SW0. Urządzenia zewnętrzne mają być sterowane 6 i 7 bitem portu PORTC.

2.2. Zadanie 2. Na ocenę 4.0 (db)

Licznik naciśnięć klawiszy. Proszę napisać program, który po każdym wciśnięciu klawisza SW0 wyświetla na pierwszym module wyświetlacza 7-segmentowego cyfry 1,2,3,4 itd., zawijając wyświetlanie do 0 po dojściu do 10. Przy starcie na wyświetlaczu ma być wyświetlona cyfra 0. Wyświetlenie segmentu LED danego modułu wymaga wyczyszczenia bitu sterującego danym modułem i segmentem.

2.3. Zadanie 3. Na ocenę 5.0 (bdb)

Proszę napisać program, który umożliwi sterowanie diodami przy użyciu joysticka (reset, przewijanie w lewo, prawo, zwiększ/zmniejsz szybkość przewijania).