



Instytut Teleinformatyki



Wydział Fizyki, Matematyki i Informatyki
Politechnika Krakowska

Mikroprocesory i mikrokontrolery

„Obsługa portów wyjścia procesora AVR”

laboratorium: 06
autor: mgr inż. Katarzyna Smelcerz

Kraków, 2015

Spis treści

Spis treści	2
1. Wiadomości wstępne	3
1.1. Niezbędne wiadomości	3
1.2. Założenie nowego projektu w środowisku AVR Studio4	3
1.3. Konfiguracja portów	5
2. Przebieg laboratorium	6
2.1. Zadanie 1. Na ocenę 3.0 (dst)	6
2.2. Zadanie 2. Na ocenę 4.0 (db)	7
2.3. Zadanie 3. Na ocenę 5.0 (bdb)	7

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. Niezbędne wiadomości

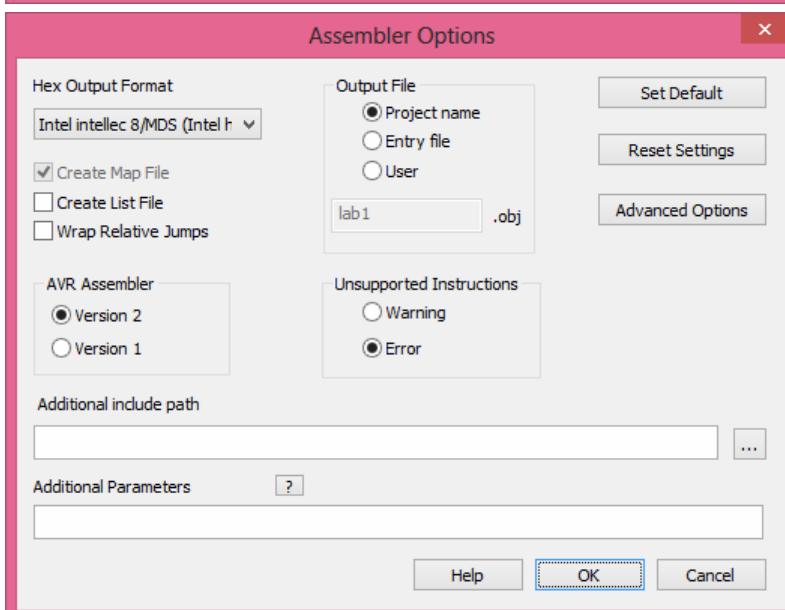
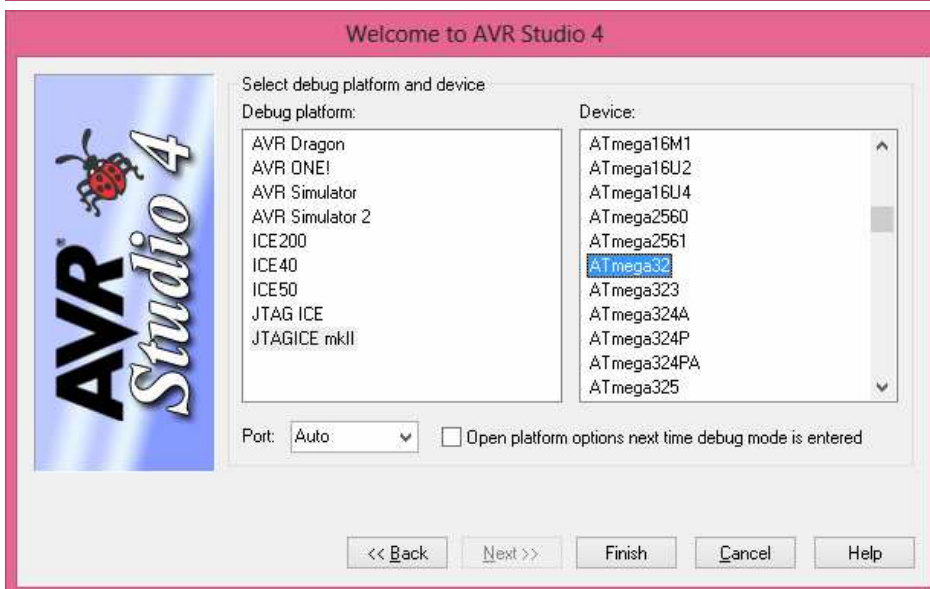
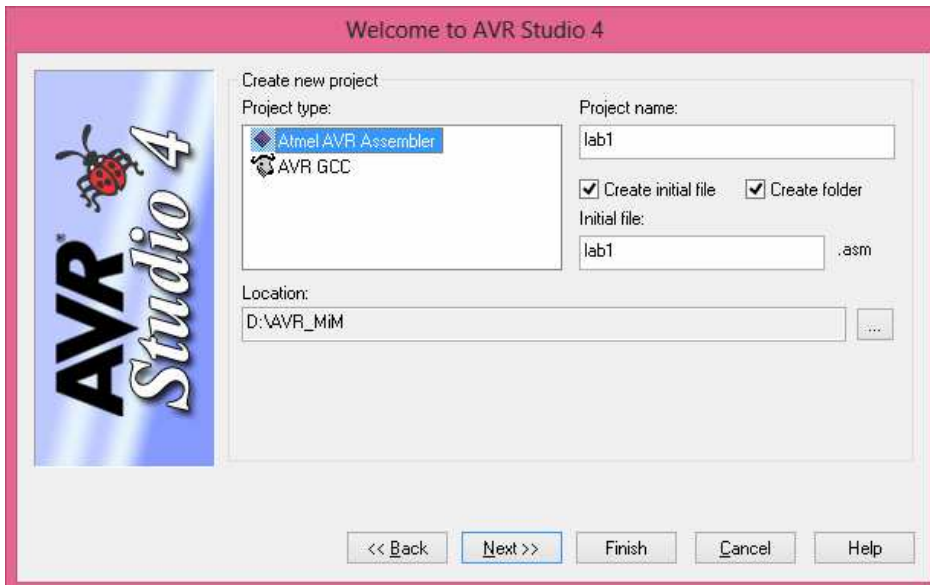
Pamięć programu, wewnętrzna pamięć RAM, mapa pamięci, rejestry, tryby adresowania oraz lista instrukcji, dostępna w dokumentacji:
<http://www.atmel.com/images/doc2503.pdf>



Schemat płytki ewaluacyjnej AVRZL15
<http://www.cyfronika.com.pl/uruchomieniowe/zl15avr.pdf>

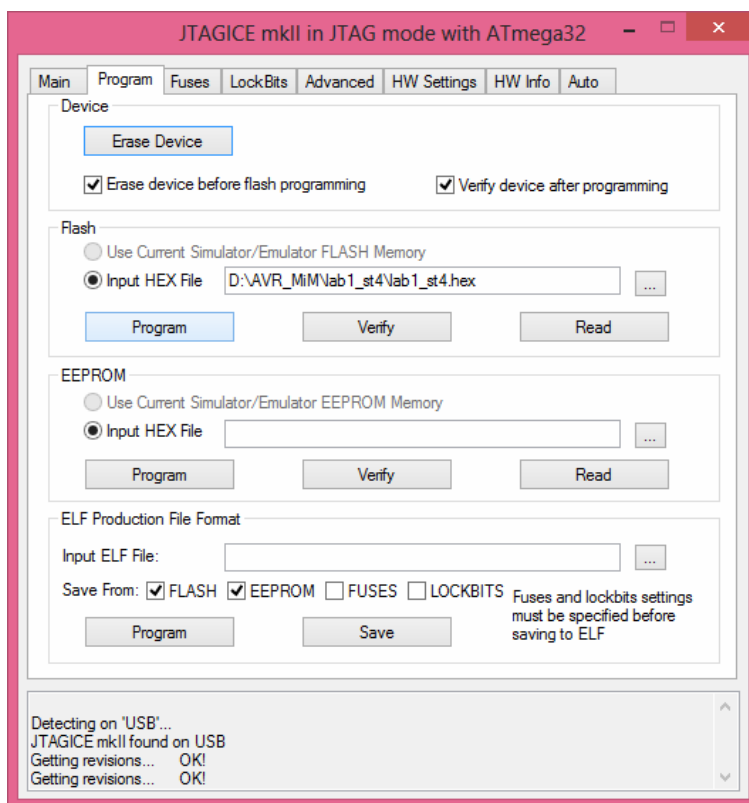
Bieżące laboratorium jest przeprowadzane w środowisku AVR Studio4. Do fizycznego zaprogramowania mikrokontrolera używany jest programator (np. KamPROG)

1.2. Założenie nowego projektu w środowisku AVR Studio4

Po uruchomieniu oprogramowania, należy wybrać opcję "New Project", następnie wybrać typ, jako Atmel AVR Assembler, oraz nadać nazwę. **Dobrze aby projekt był założony w lokalizacji, w której nie występują spacje** (w podanym przykładzie poniżej, jest to katalog AVR_MiM, zamiast np "AVR MiM"). Kolejnym krokiem jest wybranie platformy służącej do Debugowania (**w moim przykładzie, jest to programator JTAGICE mkII, jednak w sali laboratoryjnej, znajduje się inny, dlatego proszę wprowadzić typ, zgodny ze stanem sprzętu**), oraz typ mikrokontrolera: ATmega32. W oknie dialogowym Project/Assembler Options proszę sprawdzić konfigurację, zgodnie z ostatnim PrtSc poniżej.



Proces kompilacji odbywa się za pomocą skrótu F7 . Po jego przeprowadzeniu należy wybrać połączenie z programatorem  podłączonym do zestawu ewaluacyjnego AVRZL15 oraz nagrać program poprzez wybranie pliku .hex i wybranie opcji "Program" Flash.



1.3. Konfiguracja portów

W mikrokontrolerach AVR na początku należy zawsze określić, jaką funkcję ma pełnić port, tzn. czy ma być wejściem/wyjściem oraz, jaki stan ma przyjąć początkowo: "0"/"1". Dodatkowo nie wolno bezpośrednio przekazać wartości liczbowej do portu, trzeba się posłużyć tzw. rejestrem pomocniczym. Jako rejestry pomocnicze, definiujemy rejestry od R16-R31.

Za stan portów odpowiadają następujące rejestry:

DDRx, PORTx i PINx (gdzie x to: A, B, C lub D). Aby port był wyjściem, bity rejestru DDRx należy ustawić w "1", dla wejścia analogicznie w "0". Jeżeli każdy bit rejestru PORTx ustawiony będzie miał stan "1", to wyjścia lub wejścia (w zależności od stanu DDRx) będą podciągnięte do stanu "1" i odwrotnie.

Listing 1 Przykładowy fragment programu ustawiający port B jako wyjście

```
LDI R16, 0xFF      ;R16 zapisany w "1"
OUT DDRB, R16     ;port B ustawiony jako wyjście
OUT PORTB, R16    ;wszystkie wyjście portu B ustawione w stan wysoki
```

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)

Należy przeanalizować pracę mikrokontrolera poprzez wykonanie symulacji krokowej kodu przedstawionego na **Listingu 2**, a następnie uruchomić program na płytce ZL15AVR. Następnie proszę zapalić diody od D0-D3, pozostałe mają pozostać wyłączone. Proszę zwrócić uwagę na schemat płytki – wyjście LED na złączu Con9 musi być połączone 1:1 z odpowiednim portem (patrz kod programu).

Listing 2

```
.nolist
.include "m32def.inc"
.list
.listmac
.cseg
.org 0
rjmp ResetProcessor ;

ResetProcessor :
    CLI
    LDI R16, LOW(RAMEND)
    OUT SPL, R16
    LDI R16, 0xFF
    OUT DDRB, R16
    LDI R16, 0xFF
    OUT PORTB, R16

.exit
```

2.2. Zadanie 2. Na ocenę 4.0 (db)

Program „*Wędrujące zero oraz wędrująca jedynka*”. Proszę, modyfikując program z Listingu 2, wprowadzić takie zmiany, które umożliwią przesuwanie jednej zgaszonej diody od miejsca 0-7, następnie jednej zapalanej na tle pozostałych zgaszonych. Przykład przebiegu programu, dla wartości bitowych portu obsługującego diody:

1. 1111 1110
2. 1111 1101
3. 1111 1011
4. 1111 0111
5. ...itd.

Proszę zwrócić uwagę na schemat płytki – wyjście LED na złączu Con9 musi być połączone 1:1 z odpowiednim portem.

2.3. Zadanie 3. Na ocenę 5.0 (bdb)

Proszę stworzyć program „*Wędrująca jedynka*”, który umożliwi przesuwanie jednej zgaszonej diody od miejsca 0-7 (patrz przykład – zadanie 2.). Następnie proszę nagrać program na płytkę przy wykorzystaniu programatora, oraz przetestować go w czasie rzeczywistym w trybie pracy krokowej.