

Uputstvo za rad sa EasyAVR6 sistemom i programiranje pomoću CodeVision alata

1. Pregled alata za pisanje programa i programiranje EasyAVR6 ploče

Na lokaciji "E:/TelekomunikacionaElektronika" se nalaze svi potrebni dokumenti i programi koje ćemo koristiti na ovom predmetu.

CodeVisionAVR je alat za editovanje i kompajliranje koda za Atmel mikrokontrolere.

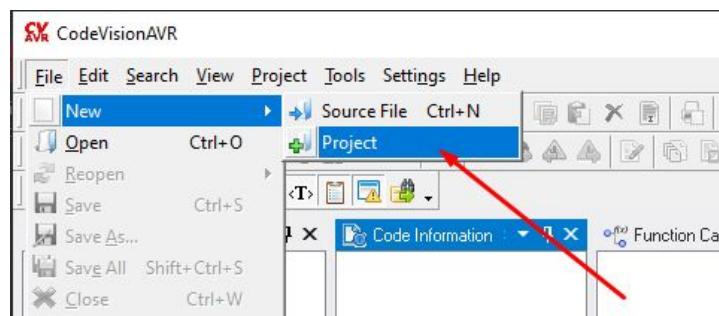
AVR Flash je alat pomoću kojeg se izvršni fajl upisuje u mikrokontroler.

CodeVisionAVR sa sobom nosi **CodeWizard** koji omogućava jednostavno konfigurisanje periferija na mikrokontroleru (konfiguracije tajmera, ADC-a, prekidnih rutina i slično).

2. Kako napraviti novi projekat?

- Otvoriti CodeVisionAVR
- Napraviti novi projekat kao na sledećim slikama:

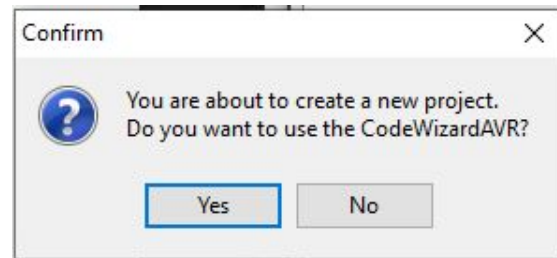
Korak 1:



Vežba 1: uputstvo za rad sa EasyAVR i CodeVision-om

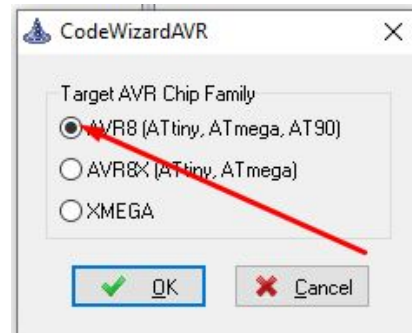
Korak 2:

- CodeWizard je alat koji nam omogućava lakše konfigurisanje periferija
- Uglavnom kada pravimo novi projekat želimo i da izgenerišemo osnovni kod, pa ćemo uglavnom odabrati **Yes**



Korak 3:

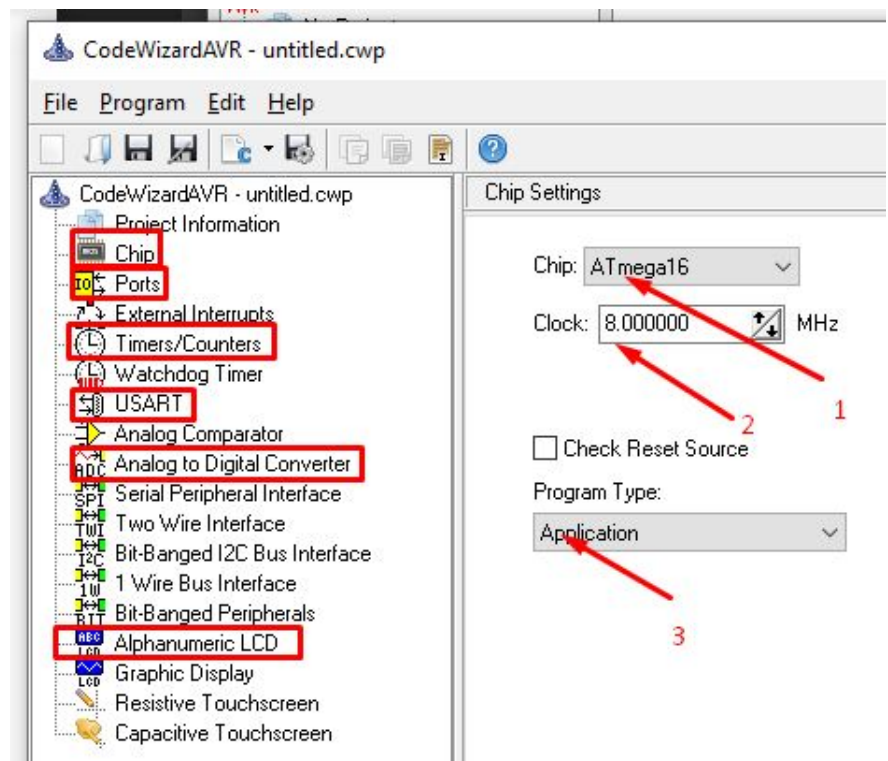
- Mikrokontroler na kojem radimo je AVR8 familije mikrokontrolera, tako da ćemo to i odabrati



Korak 4:

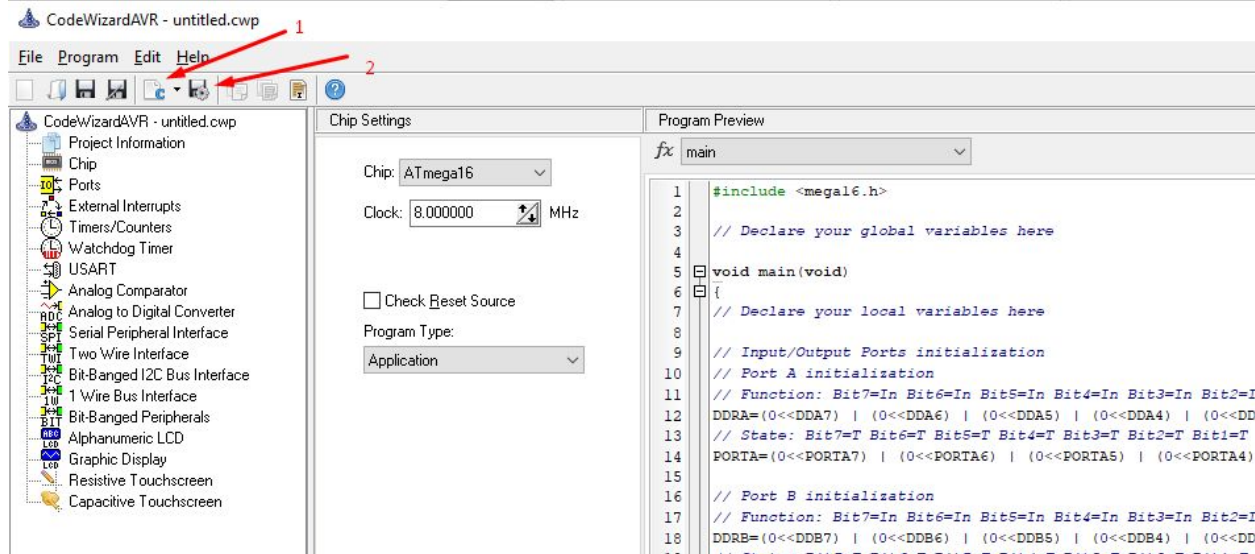
Početni prozor je obavezno potrebno postaviti kako je prikazano na slici.

Periferije koje će biti rađene na vežbama su uokvirene.



Vežba 1: uputstvo za rad sa EasyAVR i CodeVision-om

Korak 5:



Nakon odabrane konfiguracije, pritiskom na dugme (1) se generiše u desnom prozoru izgled koda.

Pritiskom na dugme (2) se taj kod generiše u vidu jednog “<ime>.c” fajla, na kraju dati tražene nazive i projekat će biti izgenerisan.

3. Kratak primer “LED Blink”

Kompajliranje:

Kopirati naredni deo koda u .c fajl koji je trenutno otvoren. Prethodni sadržaj može biti u celosti obrisan.

```
#include <mega16.h>

void my_delay(int num)
{
    int i, j;
    for( i = 0; i < num; i++)
        for( j = 0; j < num; j++);
}

void main(void)
```

Vežba 1: uputstvo za rad sa EasyAVR i CodeVision-om

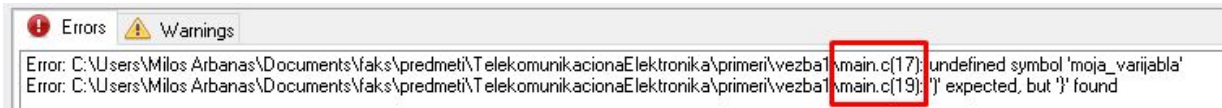
```
{
    DDRA |= (1 << DDA0);
    while(1)
    {
        PORTA |= (1 << PORTA0);
        my_delay(1000);
        PORTA &= ~(1 << PORTA0);
        my_delay(1000);
    }
}
```

Nakon kopiranja koda pokrenuti opciju **Project -> Build All** i sačekati da se kod iskompajlira.

Ako je kompajliranje uspešno, na prozoru koji se pojavi trebalo bi da se nađe sledeća poruka:

```
Build: 7
763 line(s) compiled
No errors
No warnings
```

Ako, pak, ima nekih grešaka, potrebno ih je potražiti i ukloniti. Primer prijave greške se nalazi na slici ispod:

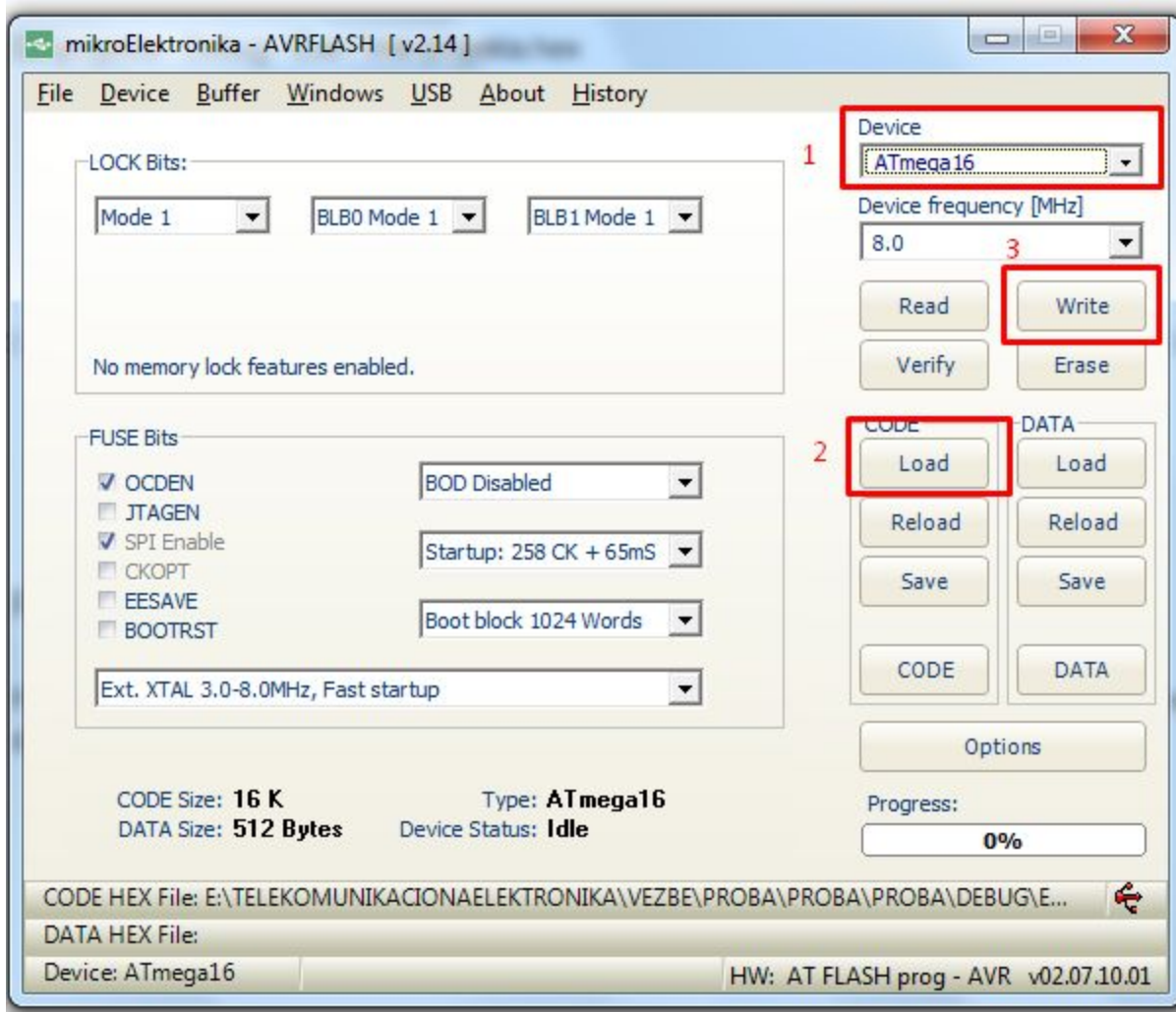


Alat ispisuje koja je greška, kao i na kojoj je liniji (uokvireno).

Nakon uspesno završenog kompajliranja, izvršni ".hex" fajl će biti upisan na sledeću lokaciju:
<Dir-projekta> / Debug / Exe / <ime-projekta>.hex

Prebacivanje izvršnog fajla na mikrokontroler

Pomoću programa AVRFLASH, izvršni fajl prebacujemo u memoriju kontrolera, koji potom može da izvršava željene instrukcije.



Prvo odaberemo kontroler na koji želimo da spustimo kod i njegovu frekvenciju rada, označeno na slici brojem 1.

Potom učitamo .hex fajl pomoću Load dugmeta označenim brojem 2.

Na kraju dugmetom Write upisujemo .hex fajl u mikrokontroler.

Zadaci:

- 1) Napisati, iskompajlirati i pokrenuti program koji na svaku sekundu menja stanja LED dioda na portu B
- 2) Napraviti softverski PWM kojim je moguće smanjiti intenzitet kojim diode sijaju