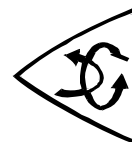




Univerza v Mariboru,
Fakulteta za elektrotehniko, računalništvo in
informatiko



INŠTITUT ZA ROBOTIKO

Delo z razvojnim orodjem eZ430-F2013 in razširitveno kartico ez430f2013-ext

Navodila za delo

Maribor, julij, 2008

Avtor: Miran Rodič

Podatki o avtorju

Avtor: dr. Miran Rodič, dipl. inž. el.

Naslov: Inštitut za robotiko
Fakulteta za elektrotehniko, računalništvo in informatiko
Univerza v Mariboru
Smetanova ulica 17, SI-2000 Maribor, Slovenija, EU

Tel.: (+386 2) 220 7308

Fax: (+386 2) 220 7315

e-mail: miran.rodic@uni-mb.si

www: <http://www.ro.feri.uni-mb.si/~miranro/>

Kazalo

1. Uvod.....	4
2. Opis strojne opreme	5
3. Delo s programskim orodjem.....	7
3.1. Zagon programskega orodja.....	7
3.2. Ustvarjanje novega projekta v programskem jeziku C	7
4. Opis funkcij v knjižnici fn_pze1.lib.....	15
4.1. init_ez430f2013_ext.....	15
4.2. LED2_pwm.....	15
4.3. pwm_LED2.....	15
4.4. beri_adc.....	16
4.5. beri_S1	17
4.6. beri_S2	17
4.7. LED0_vklop.....	17
4.8. LED0_izklop.....	18
4.9. LED1_vklop.....	18
4.10. LED1_izklop.....	18
4.11. LED2_vklop.....	19
4.12. LED2_izklop.....	19
4.13. LED3_vklop.....	19
4.14. LED3_izklop.....	20
5. Literatura.....	21
Varstvo avtorskih pravic	22

1. Uvod

Modul eZ430-F2013 je namenjen uporabi za razvoj aplikacij, lahko pa ga uporabimo tudi v pedagoške namene. Zgrajen je z uporabo mikrokrmilnika MSP430F2013.

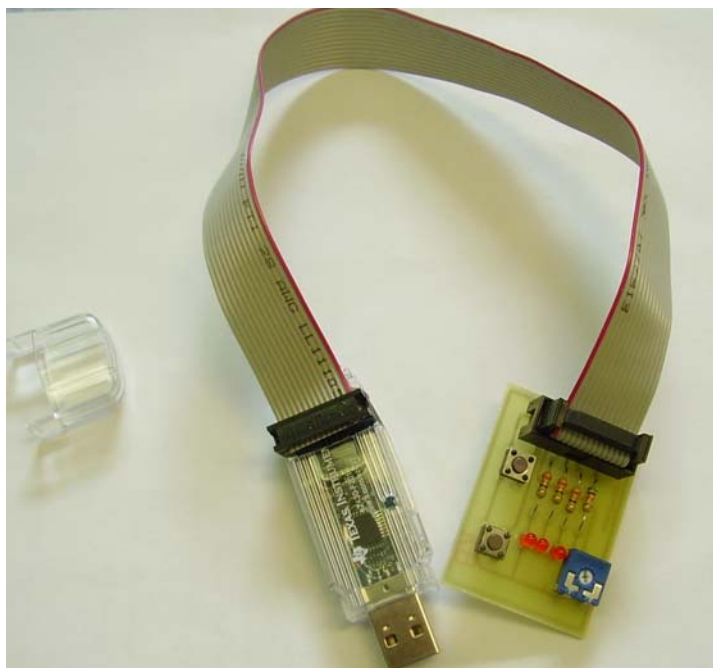
Z namenom uporabe v pedagoškem procesu je bil izdelan dodaten modul, kartica **ez430f2013-ext**, ki študentu omogoča preprost dostop do nekaterih komponent strojne opreme. Na ta način lahko piše programe, ki uporabljajo vhode in izhode mikrokrmilnika in neposredno preizkuša njihovo delovanje.

V nadaljevanju bo opisan modul in njegova uporaba v kombinaciji s programskim orodjem **IAR Workbench Kickstart**.

Podrobnejše informacije o mikrokrmilniku, programiranju in podobnem so podane v dokumentu »Delo z razvojnim orodjem eZ430-F2013« [4].

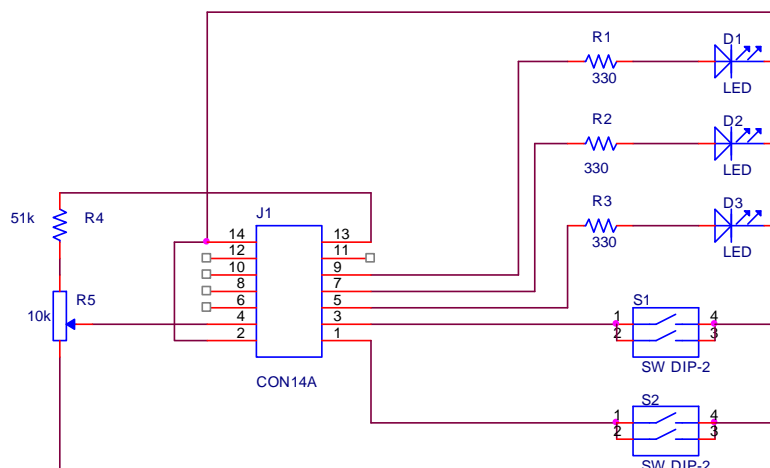
2. Opis strojne opreme

Kartico ez430f2013-ext kaže Slika 1. Deluje le v kombinaciji z razvojnim orodjem eZ430-F2013, saj nima lastnega napajanja in mikrokrmilnika.

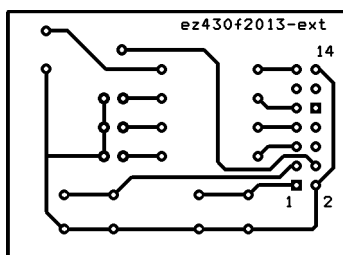


Slika 1: Razvojno orodje

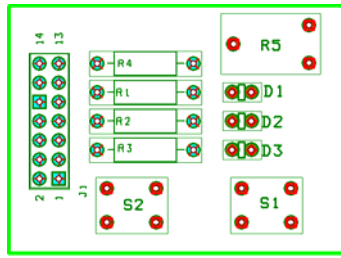
Shemo karte podaja Slika 2, tiskano vezje Slika 3, položaj komponent pa Slika 4 in Slika 5.



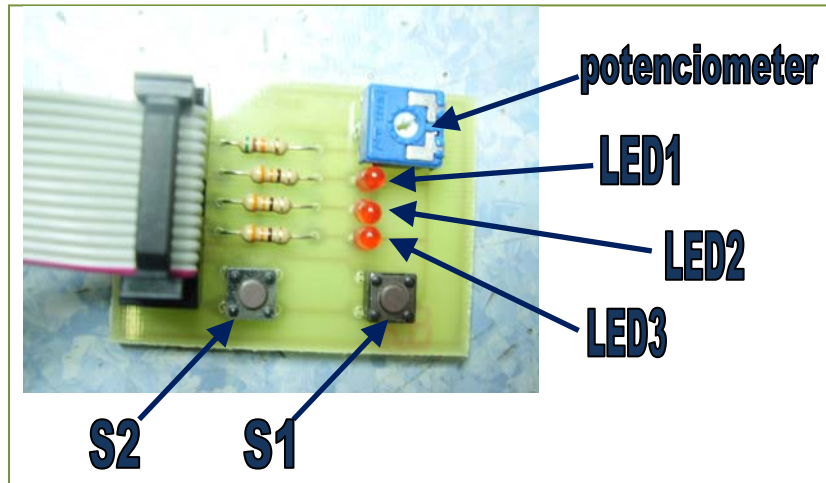
Slika 2: Ciljna karta ez430f2013-ext



Slika 3: Tiskano vezje ez430f2013-ext



Slika 4: Komponente ez430f2013-ext



Slika 5: Fotografija ez430f2013-ext z oznakami tipk, LED in potenciometra

Preko razširitvene kartice ez430f2013-ext imamo dostop do:

- Digitalnih izhodov:
 - Svetleča dioda LED1, povezana na pin P1.1 mikrokrmilnika MSP430F2013
 - Svetleča dioda LED2, povezana na pin P1.2 mikrokrmilnika MSP430F2013
 - Svetleča dioda LED3, povezana na pin P1.3 mikrokrmilnika MSP430F2013
- Digitalnih vhodov:
 - Tipka S1, povezana na pin P1.4 mikrokrmilnika MSP430F2013
 - Tipka S2, povezana na pin P1.5 mikrokrmilnika MSP430F2013
- Analognega vhoda:
 - Potenciometer, katerega drsnik je povezan na pin A3+ mikrokrmilnika MSP430F2013, masa pa na pin A3- mikrokrmilnika MSP430F2013

Svetlečo diodo LED2 je mogoče prižigati oz. ugašati tudi preko pulzno-širinsko moduliranega signala.

Dodatno je na samem razvojnem orodju eZ430-F2013 dostopen še en digitalni izhod:

- Svetleča dioda LED0, povezana na pin P1.0 mikrokrmilnika MSP430F2013

3. Delo s programskim orodjem

Za delo z razvojnim okoljem eZ430-F2013 uporabljamo programsko orodje IAR Embedded Workbench IDE za mikrokrmilnike družine 430 proizvajalca Texas Instruments. Orodje je prosto dostopno na internetu, omejena verzija je na voljo brezplačno. Uporabimo ga lahko v okolju Windows (2000 in XP).

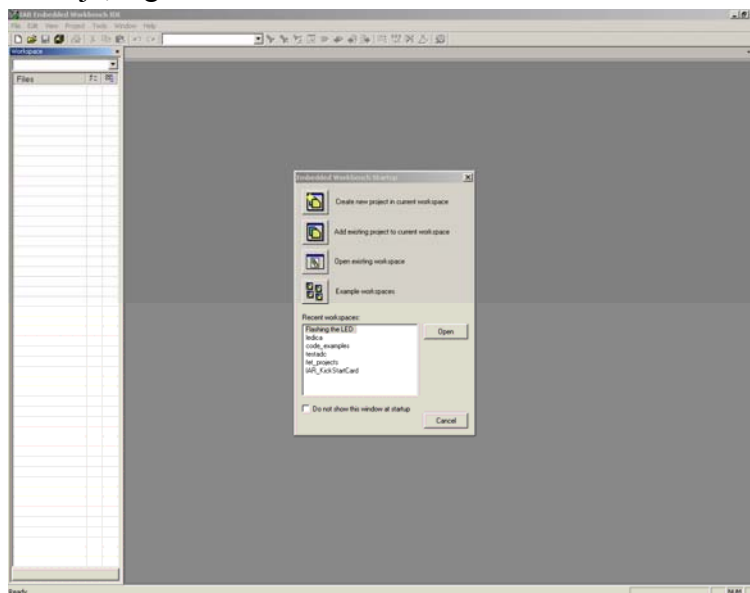
Ta dokument je napisan za verzijo 4.10. Ikone in oblika zaslona so lahko v novejših verzijah nekoliko drugačni. Natančnejša navodila so podana v [4], zato bo na tem mestu podan samo primer, kako v program vključimo knjižnico, ki vsebuje funkcije za delo z razširitveno kartico ez430f2013-ext.

Delo s programskim orodjem bo opisano po korakih.

3.1. Zagon programskega orodja

Razvojno okolje poženemo z izbiro *Start → Programs → IAR Systems → IAR Embedded Workbench Kickstart for MSP430 V4 → IAR Embedded Workbench*.

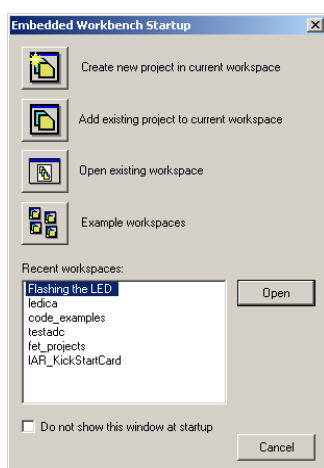
Odre se programsko okolje, izgled kaže Slika 6.



Slika 6: Programsko okolje IAR Embedded Workbench

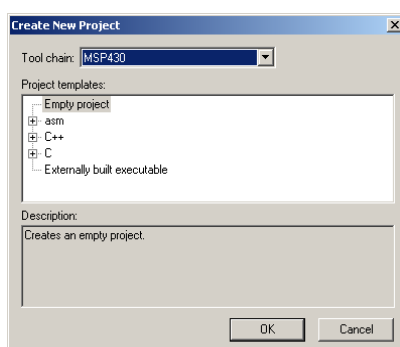
3.2. Ustvarjanje novega projekta v programskem jeziku C

Nov projekt v programskem jeziku C ustvarimo tako, da v oknu *Embedded Workbench Startup* izberemo opcijo *Create new project in current workspace* s čemer poženemo čarovnika za ustvarjanje projektov.



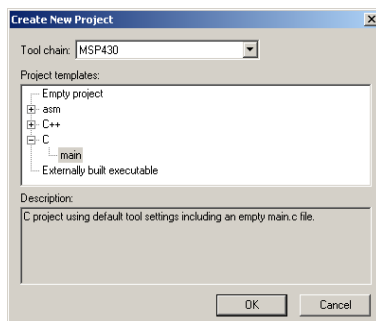
Slika 7: Ustvarjanje novega projekta v programskem jeziku C

Pokaže se okno, ki ga kaže Slika 8.



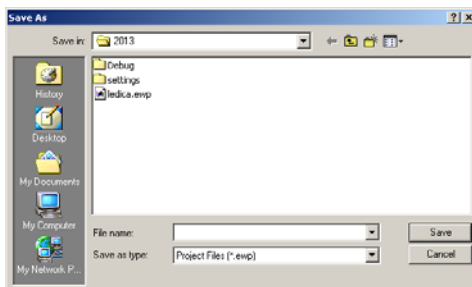
Slika 8: Uvodno okno Create New Project

Ker želimo ustvariti nov program v jeziku C izberemo opcijo C in kliknemo main (Slika 9).



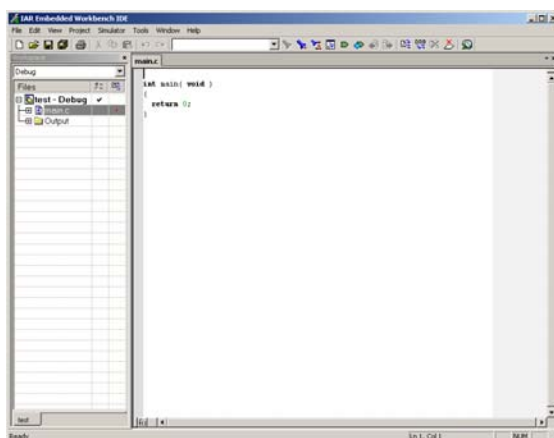
Slika 9: Izbira opcije C v oknu *Create New Project*

Kliknemo **OK** in odpre se okno za izbiro imena projekta in mesta, kamor ga bomo shranili.



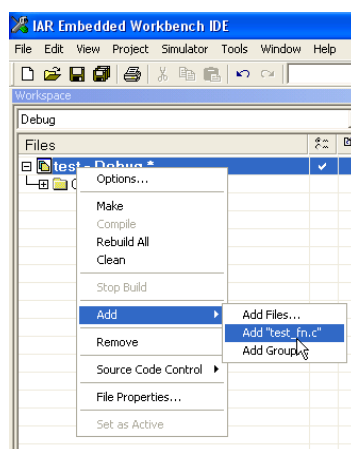
Slika 10: Izbira imena projekta in njegove lokacije na disku

Izberemo ime, npr **test** in kliknemo **Save** ter se tako pomaknemo na okno programa. Odpre se začetna datoteka **main.c**.



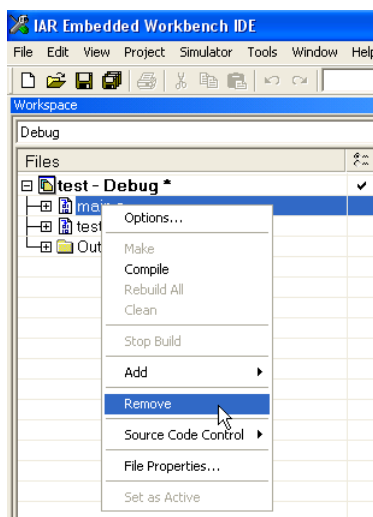
Slika 11: Okno *IAR Embedded Workbench IDE* po nastavitvi imena projekta

Zdaj lahko začnemo s pisanjem programa, lahko pa tudi naložimo drugo kodo, npr. *test_fn.c*, ki je priložen.



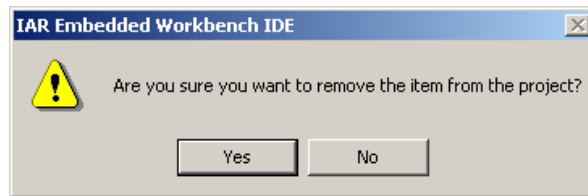
Slika 12: Dodajanje nove datoteke v projekt

V tem primeru je potrebno seveda odstraniti program *main.c*, ki je bil ustvarjen samodejno. To naredimo tako, da z desno tipko na miški kliknemo *main.c* in izberemo opcijo **Remove**.



Slika 13: Odstranjevanje datoteke *main.c* iz projekta

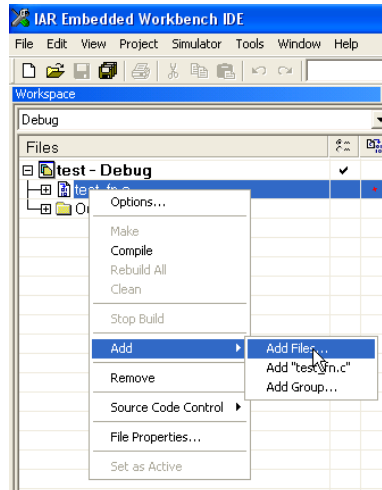
Odpre se okno, s katerim nam program da še eno možnost, da si premislimo.



Slika 14: Okno za potrditev odločitve o odstranitvi datoteke *main.c* iz projekta

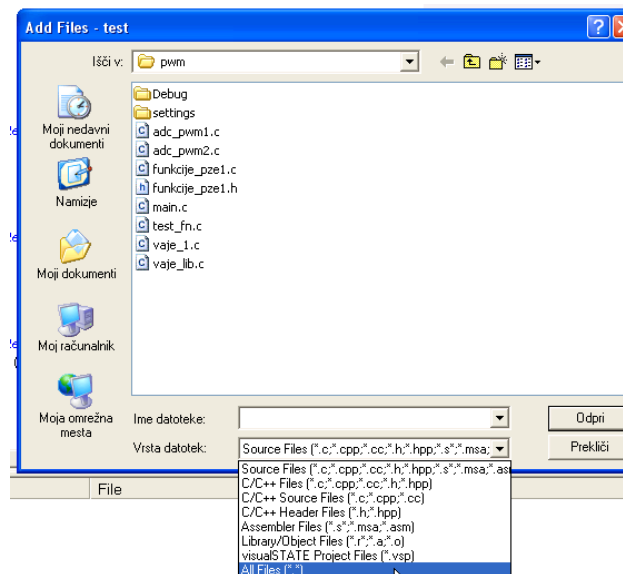
Kliknemo *Yes* in datoteke *main.c* ni več v projektu.

Vključiti moramo še priloženo prevedeno knjižnico *fn_pze1.lib*, ki vsebuje funkcije za delo z razširitveno karto *ez320f2013-ext*. Z desno tipko na miški kliknemo ime projekta in izberemo opcijo *Add Files ...*



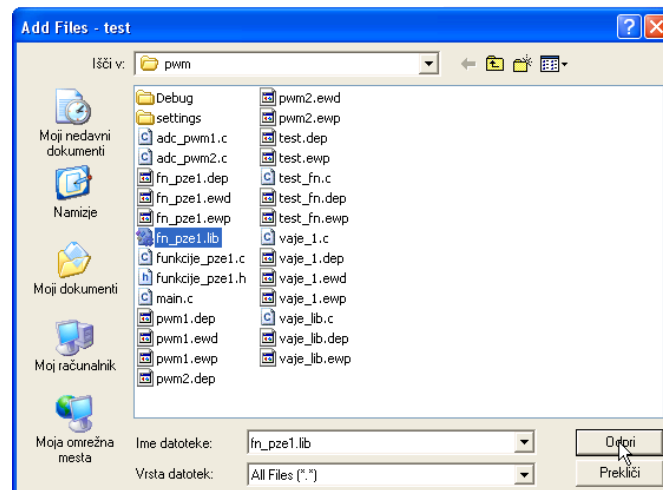
Slika 15: Dodajanje knjižnice *fn_pze1.lib* v projekt (prvi korak)

Ker končnice *.lib* ni na spisku možnih, izberemo *All Files (*.*)*. Nato poiščemo datoteko *fn_pze1.lib*.



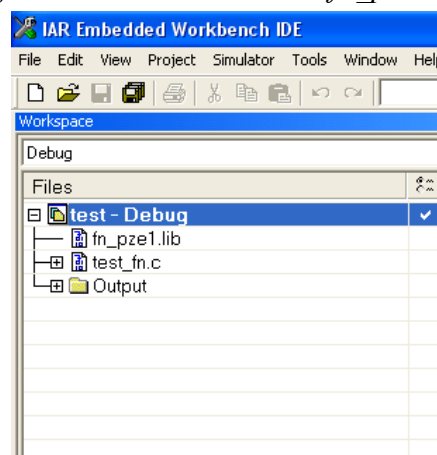
Slika 16: Dodajanje knjižnice *fn_pze1.lib* v projekt (drugi korak)

Ko najdemo datoteko (pomaga, če vemo kam smo jo predhodno shranili), jo izberemo in kliknemo na gumb *Odpri*.



Slika 17: Dodajanje knjižnice *fn_pze1.lib* v projekt (tretji korak – izbira datoteke)

Po opravljenem vključevanju knjižnice mora biti datoteka *fn_pze.lib* vključena v projekt.



Slika 18: Dodajanje knjižnice *fn_pze1.lib* v projekt (prikaz po vključitvi knjižnice)

Vsebina datoteke *test_fn.c* je:

```

//*****
// test_fn.c - Test tipk S1 in S2 ter LED0, LED1, LED2 IN LED3
//
// Opis: Primer programa za vaje pri predmetu Programiranje za elektrotehnike
//       1 z uporabo razširitvene karte ez430f2013_ext.
//
//           MSP430x2013
//           -----
//           /\|          XIN|-
//           | |          |
//           --|RST      XOUT|-
//           |           |
//           -->|A3+/-    P1.0|-->LED0
//           |           |
//           S1 -->|P1.4    P1.1|-->LED1
//           |           |
//           S2 -->|P1.5    P1.2|-->LED2 (PWM1)
//           |           |
//           |           P1.3|-->LED3
//           |           |
//
// M. Rodič

```

```

// UM-FERI
// Julij 2008
// Built with IAR Embedded Workbench Version: 3.40A
//*****

#include "msp430x20x3.h"
#include "funkcije_pze1.h"

//*****
// FUNKCIJA:
// void main()
//
// Opis: Glavni program (funkcija).
//*****
void main(void)
{
    volatile unsigned int i;          // volatile, da preprecimo optimizacijo

    init_ez430f2013_ext();           // inicializacija periferije za ez430f2013_ext
    LED2_pwm(0);                     // LED2 je priključena na digitalni izhod

    for (;;)
    {

        P1OUT ^= 0x01;                // Preklaplajaj P1.0 z uporabo
                                     // ekskluzivnega ALI

        // Prižiganje LED1 glede s tipko S1
        if(beri_S1())                 // Če je S1 pritisnjena
            LED1_vklop();             //   prižgi LED1
        else                           // sicer
            LED1_izklop();            //   ugasni LED1

        // Prižiganje LED2 glede s tipko S2
        if(beri_S2())                 // Če je S2 pritisnjena
            LED2_vklop();             //   prižgi LED2
        else                           // sicer
            LED2_izklop();            //   ugasni LED2

        // Prižiganje LED3 glede s tipko S1 in S2
        if(beri_S1() && beri_S2())     // Če sta S1 in S2 pritisnjeni
            LED3_vklop();             //   prižgi LED3
        else                           // sicer
            LED3_izklop();            //   ugasni LED3

        i = 3000;                     // Programska zakasnitev
        do i--;
        while (i != 0);

    }
}

```

Za deklaracijo funkcij potrebujemo še zaglavje *funkcije_pze1.h*, katerega vsebina je:

```

//*****
// funkcije_pze1.h - Deklaracije funkcij za delo z razširitveno kartico
//                   ez430f2013_ext.
//
// Opis: Deklaracije funkcij programov za vaje pri predmetu Programiranje za
//       elektrotehniko 1 z uporabo razširitvene karte ez430f2013_ext.
//

```

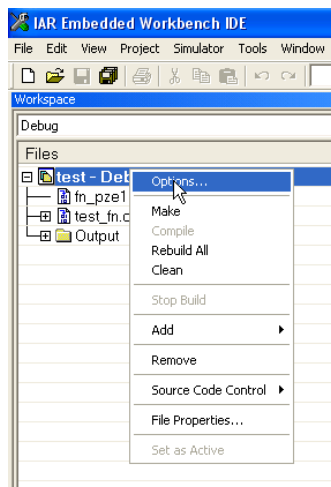
```

//          MSP430x2013
//          -----
//          /|\|          XIN|-
//          | |          |
//          --|RST        XOUT|-
//          |          |
//          -->|A3+/-      P1.0|-->LED0
//          |          |
//          S1 -->|P1.4      P1.1|-->LED1
//          |          |
//          S2 -->|P1.5      P1.2|-->LED2 (PWM1)
//          |          |
//          |          P1.3|-->LED3
//          |          |
//
// M. Rodič
// UM-FERI
// Julij 2008
// Built with IAR Embedded Workbench Version: 4.10E
//*****

#ifndef FN_PZE1_H
#define FN_PZE1_H
#include "msp430x20x3.h"
void init_ez430f2013_ext();
void LED2_pwm(int pwm);
void pwm_LED2(unsigned int perioda, unsigned int cas_vklop);
unsigned int beri_adc();
unsigned int beri_S1();
unsigned int beri_S2();
void LED0_vklop();
void LED0_izklop();
void LED1_vklop();
void LED1_izklop();
void LED2_vklop();
void LED2_izklop();
void LED3_vklop();
void LED3_izklop();
#endif

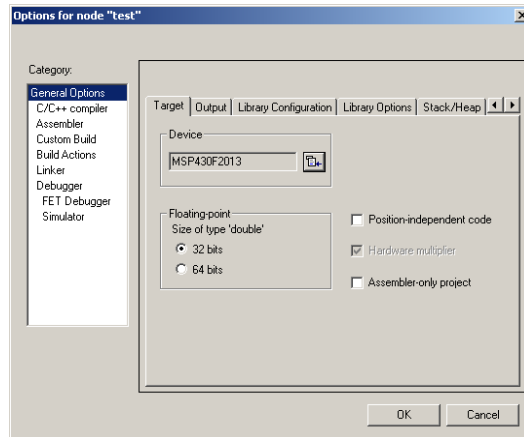
```

Zdaj je potrebno nastaviti parametre delovanja orodja in razhroščevalnika. Z desno tipko miške kliknemo na projekt in izberemo opcijo *Options...*



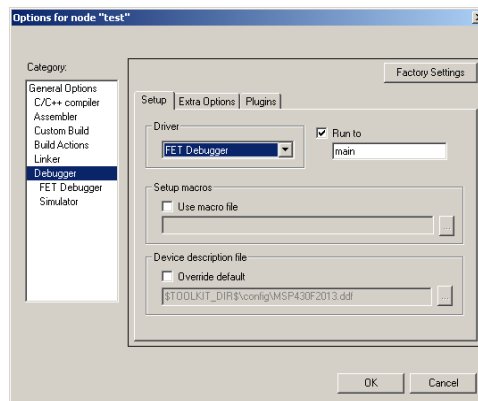
Slika 19: Nastavljenje opcij projekta

Najprej v *General options* v zavihku *Target* izberemo ciljni procesor, v našem primeru je to *MSP430F2013*.



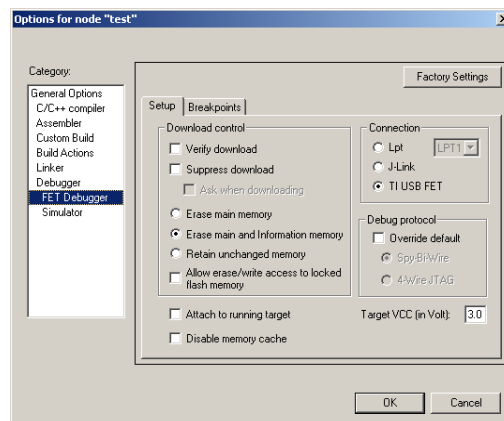
Slika 20: Izbira procesorja

Naslednji korak je nastavitve lastnosti razhoščevalnika. Pod opcijo *Debugger* v zavihku *Setup* nastavimo *Driver* na *FET Debugger*.



Slika 21: Uporaba stroje opreme – *FET Debugger*

V naslednjem koraku pod opcijo *Debugger* → *FET Debugger* v zavihku *Setup* nastavimo način priklopa naprave, pod *Connection* izberemo *TI USB FET*.



Slika 22: Izbira načina priklopa opreme

Zdaj kliknemo *OK*. S tem je program pripravljen za prevajanje in razhoščevanje.

4. Opis funkcij v knjižnici *fn_pze1.lib*

Nabor funkcij v knjižnici zajema funkcije za nastavljanje načinov delovanja strojne in programske opreme (inicializacijo), za branje digitalnih in analognih vhodov ter postavljanje digitalnih izhodov, od katerih lahko enega krmilimo tudi s pulzno širinsko modulacijo (PWM). Opisane bodo v nadaljevanju.

4.1. *init_ez430f2013_ext*

Deklaracija:

```
void init_ez430f2013_ext();
```

Klic:

```
init_ez430f2013_ext();
```

Opis:

Inicializacija (postavitev stanja registrov) procesorja za delo z razširitveno karto *ezf2013_ext*.

Vhodi:

-

Izhod:

-

4.2. *LED2_pwm*

Deklaracija:

```
void LED2_pwm(int pwm);
```

Klic:

```
LED2_pwm(pwm);
```

Opis:

Če je *pwm* 1, je LED2 vklapljana preko PWM, sicer preko digitalnega izhoda.

Vhodi:

- *int pwm* – parameter za izbiro

Izhod:

-

4.3. *pwm_LED2*

Deklaracija:

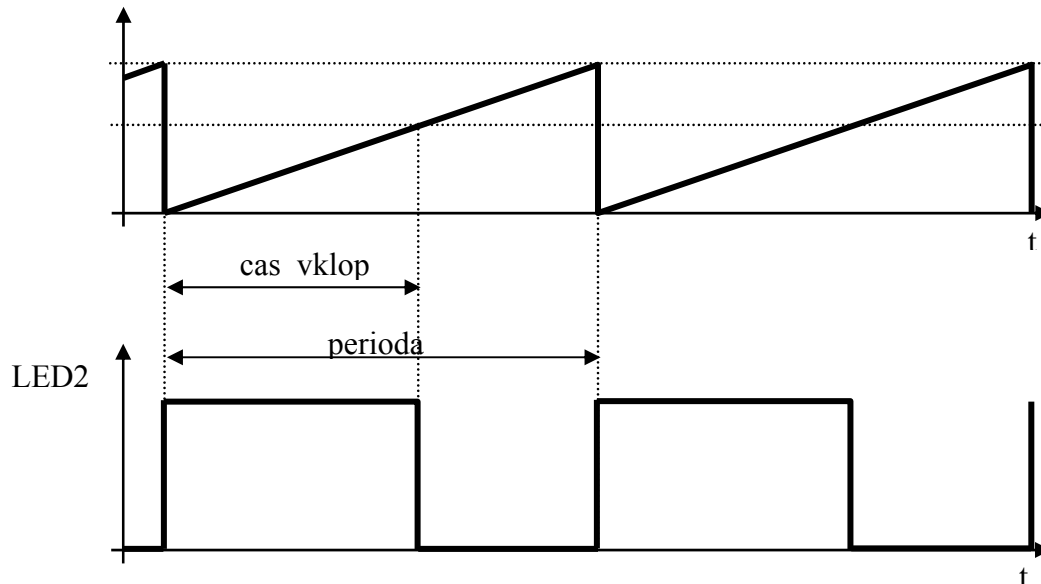
```
void pwm_LED2(unsigned int perioda, unsigned int cas_vklop);
```

Klic:

```
pwm_LED2(perioda, cas_vklopa);
```

Opis:

Pulzno širinska modulacija (PWM) izhodnega signala za LED2. Perioda je določena s spremenljivko *perioda*, čas vklopa (delež periode, ko je izhod na 1), pa s spremenljivko *cas_vklop*. Grafični opis kaže Slika 23.



Slika 23: Pulzno širinska modulacija (PWM) – grafični opis

Vhodi:

- unsigned int perioda – čas trajanja periode
- unsigned int cas_vklop – čas, ko je izhod na 1

Izhod:

-

4.4. beri_adc

Deklaracija:

```
unsigned int beri_adc();
```

Klic:

```
a = beri_adc();
```

Opis:

Branje vrednosti napetosti (16b) na analognem vhodu A3 (potenciometer).

Vhodi:

-

Izhod:

- Digitalna vrednost napetosti na analognem vhodu A3 (potenciometer).

4.5. beri_S1

Deklaracija:

```
unsigned int beri_S1();
```

Klic:

```
d = beri_S1();
```

Opis:

Branje stanja tipke S1 (ali je pritisnjena). Če je pritisnjena, funkcija vrne 1, sicer vrne 0.

Vhodi:

-

Izhod:

- Digitalna vrednost na vhodu P1.4.

4.6. beri_S2

Deklaracija:

```
unsigned int beri_S2();
```

Klic:

```
d = beri_S2();
```

Opis:

Branje stanja tipke S2 (ali je pritisnjena). Če je pritisnjena, funkcija vrne 1, sicer vrne 0.

Vhodi:

-

Izhod:

- Digitalna vrednost na vhodu P1.5.

4.7. LED0_vklop

Deklaracija:

```
void LED0_vklop();
```

Klic:

```
LED0_vklop();
```

Opis:

Funkcija vklopi LED0.

Vhodi:

-

Izhod:

-

4.8. LED0_izklop

Deklaracija:

```
void LED0_izklop();
```

Klic:

```
LED0_izklop();
```

Opis:

Funkcija izklopi LED0.

Vhodi:

-

Izhod:

-

4.9. LED1_vklop

Deklaracija:

```
void LED1_vklop();
```

Klic:

```
LED1_vklop();
```

Opis:

Funkcija vklopi LED1.

Vhodi:

-

Izhod:

-

4.10. LED1_izklop

Deklaracija:

```
void LED1_izklop();
```

Klic:

```
LED1_izklop();
```

Opis:

Funkcija izklopi LED1.

Vhodi:

-

Izhod:

-

4.11. LED2_vklop

Deklaracija:

```
void LED2_vklop();
```

Klic:

```
LED2_vklop();
```

Opis:

Funkcija vklopi LED2.

Vhodi:

-

Izhod:

-

4.12. LED2_izklop

Deklaracija:

```
void LED2_izklop();
```

Klic:

```
LED2_izklop();
```

Opis:

Funkcija izklopi LED2.

Vhodi:

-

Izhod:

-

4.13. LED3_vklop

Deklaracija:

```
void LED3_vklop();
```

Klic:

```
LED3_vklop();
```

Opis:

Funkcija vklopi LED3.

Vhodi:

-

Izhod:

-

4.14. LED3_izklop

Deklaracija:

```
void LED3_izklop();
```

Klic:

```
LED3_izklop();
```

Opis:

Funkcija izklopi LED3.

Vhodi:

-

Izhod:

-

5. Literatura

- [1] *eZ430-F2013 User's Guide (Rev. B)*, <http://focus.ti.com/docs/toolsw/folders/print/iar-kickstart.html>
- [2] *FET User's Guide (IAR Workbench Ver 3.x) (Rev. D)*,
<http://www-s.ti.com/sc/techlit/slau138>
- [3] *MSP430F2xx Family User's Guide (Rev. B)*,
<http://www-s.ti.com/sc/techlit/slau144>
- [4] *Miran Rodič, Delo z razvojnim orodjem eZ430-F2013, Univerza v Mariboru, september 2007*

Varstvo avtorskih pravic

Vse pravice so pridržane. Noben del gradiv ne sme biti reproduciran, spremenjen ali distribuiran brez predhodnega dovoljenja lastnikov avtorskih pravic. Gradivo je last Inštituta za robotiko, Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru, in njegovih avtorjev.