

# UVOD V PROGRAMIRANJE MIKROKRMILNIKOV

Programirno okolje  
Microchip MPLAB - Simulator

Janez Pogorelc  
UM-FERI, maj 2006

---



Univerza v Mariboru  
Fakulteta za elektrotehniko,  
računalništvo in informatiko



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



Inštitut za Robotiko

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

<http://www.hipulab.uni-mb.si/>

## 1. Uporaba programskega orodja Microchip MPLAB SIM

*Spoznali boste programska orodja integriranega programskega okolja Microchip MPLAB. Naučili se boste kreirati projekt, konfigurirati Simulator ter sprožiti postopek generiranja izvršljivega programa za MPLAB Simulator na osebem računalniku.*

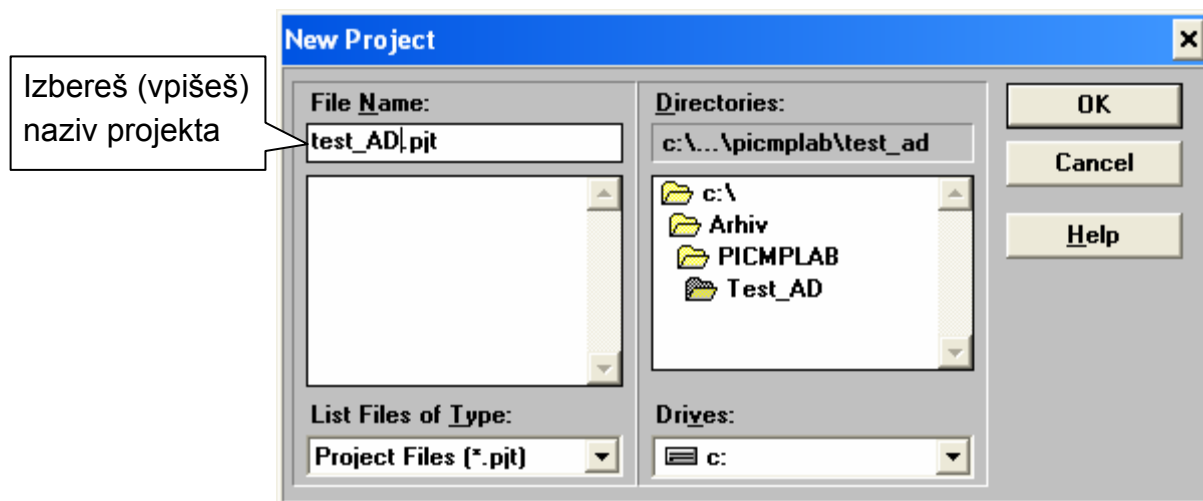
**Opomba:** Opis se nanaša na programsko orodje **Microchip MPLAB V5.70** (zadnje različice, ki še omogoča delo v kombinaciji z **ICD1**). V kolikor ima uporabnik namen izključno uporabljati **Simulator** oz. možnost dela z **ICD2** (ali naprednejšim razvojnim sistemom), je priporočljiva uporaba novejših različic **MPLAB** iz serije npr. **V7.X**.

**V nadaljevanju bo podrobno opisan postopek od kreiranja projekta do stopnje, ko lahko program izvajamo z orodjem MPLAB Simulator na osebem računalniku, tako da posnemamo ciljni mikrokrmilnik (npr.: MPU-PIC16F876).**

Poženite program [10][9][16] **Microchip MPLAB** (**Start** → **Programs** → **Microchip MPLAB** → **MPLAB**) in prikaže se osnovno okno **MPLAB** programirnega okolja.

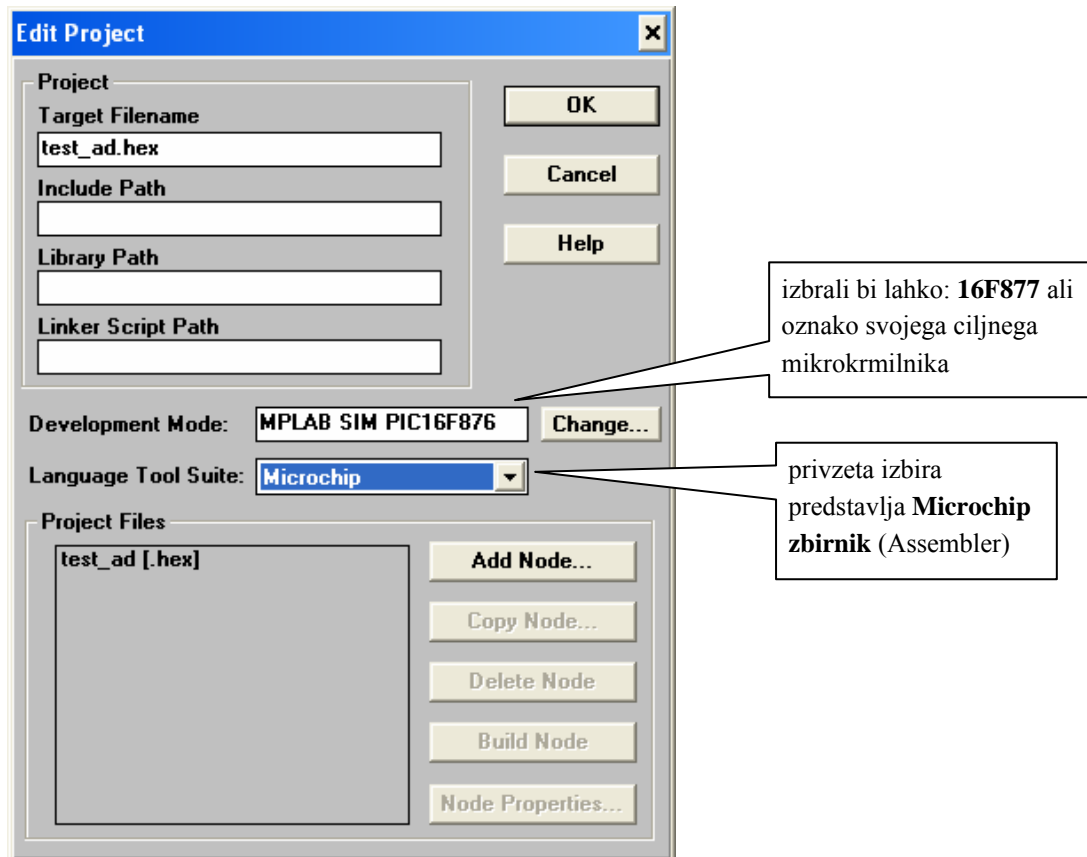
### 1.1. Konfiguriranje MPLAB programskega okolja

Odprite (Slika 1-1) novi projekt (**Project** → **New Project ...**); projekt imenujte poljubno; obvezna je končnica **.pjt**, pot (direktorij) odprete v svojem delovnem imeniku (v istem imeniku hranite tudi **.asm** in **.hex** datoteke):



Slika 1-1: Okno za »nov projekt«

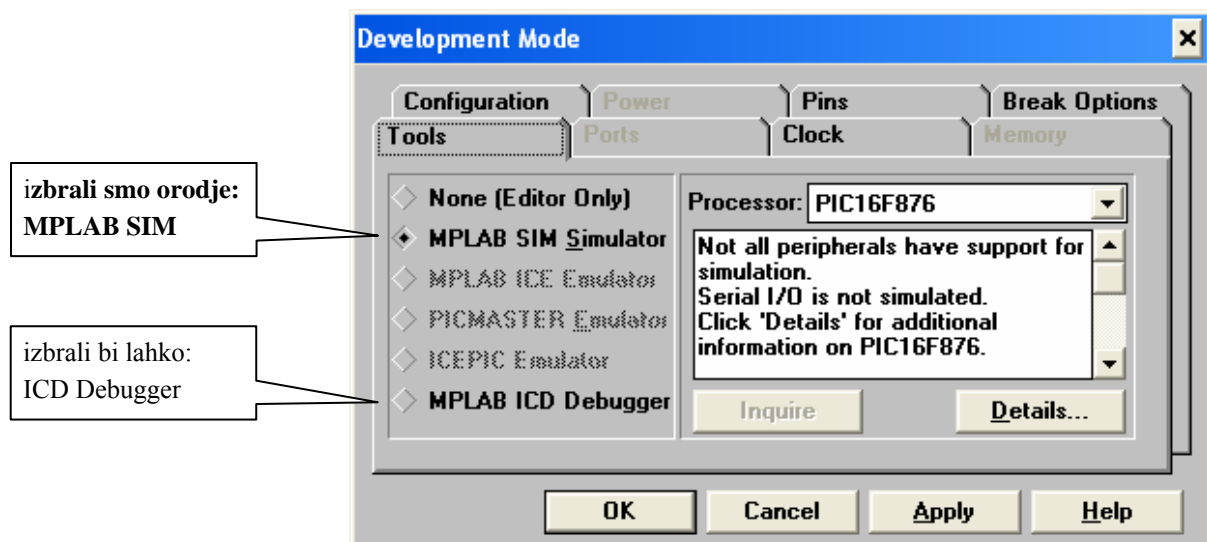
Pritisnite OK, odpre se (Slika 1-2) okno **Edit Project**.



Slika 1-2: Okno »urejanje lastnosti projekta«

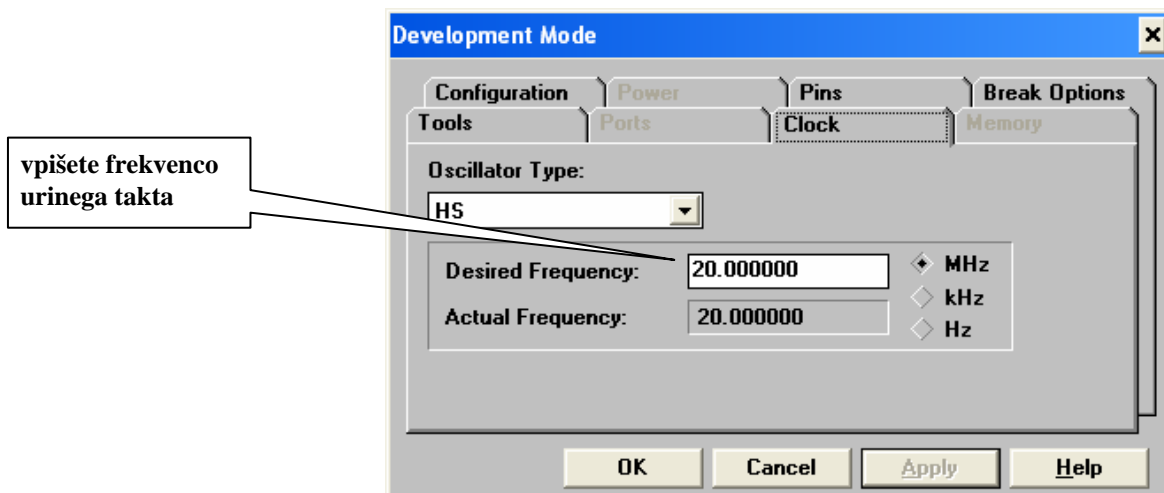
Ustvari se datoteka s končnico **.hex**, z imenom projekta npr.: *test\_ad.hex* (tovrstna datoteka predstavlja preveden - izvršljiv program, ki bo generiran kasneje).

Kliknite **Change ...**, odpre se (Slika 1-3) okno **Development Mode**, kjer nastavite opcije, kot so prikazane na spodnji sliki:



Slika 1-3: Okno za izbiro razvojnega okolja

Izberete zavihek **Clock** in prikaže se (Slika 1-4) okno **MPLAB-ICD**, kjer lahko poljubno spremenite frekvenco urinega takta procesorja:

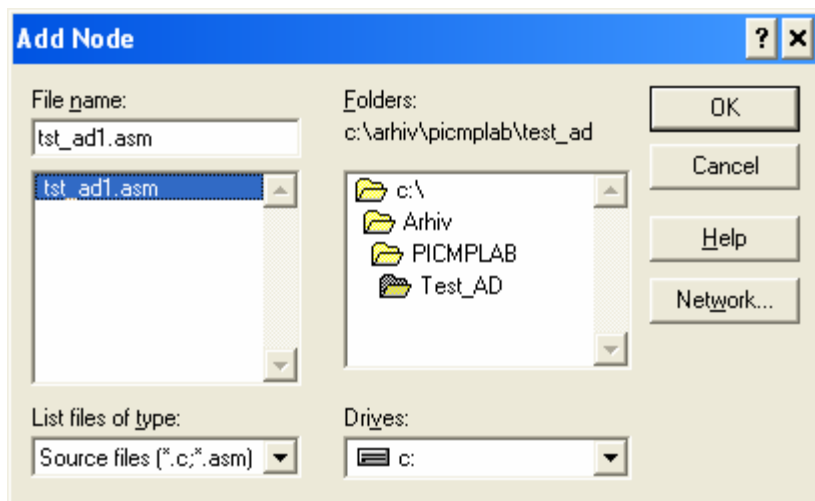


Slika 1-4: Okno za izbiro COM vrat ICD modula

**Opomba:** Največja možna frekvenca mikrokrmilnika **PIC 16F876** je **20 MHz**, kar predstavlja **200 ns** čas izvajanja običajnega ukaza. Glede na čim bolj verno posnemanje ciljnega sistema, lahko zberemo tudi nižjo frekvenco (privzeto je 4 MHz).

## 1.2. Nalaganje izvornega programa v ciljni sistem in testiranje

V oknu **Edit Project** kliknete še **Add Node** in vključite datoteko (Slika 1-5), v kateri se nahaja program s končnico **.asm** (izvirni program v zbirnem jeziku, npr.: *tst\_ad1.asm*; le-to lahko kasneje urejate v vsebovanem urejevalniku **MPLAB Editor**):

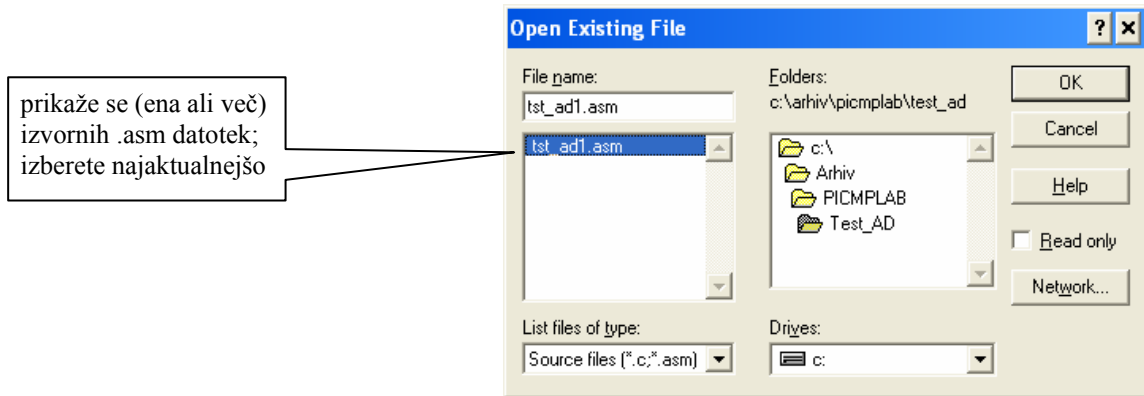


Slika 1-5: Okno za izbiro obstoječega ali vpis novega imena izvorne datoteke

Kliknete **OK** in nato tudi v oknu **Edit Project** storite enako - **OK** (s tem se obe okni zapreta).

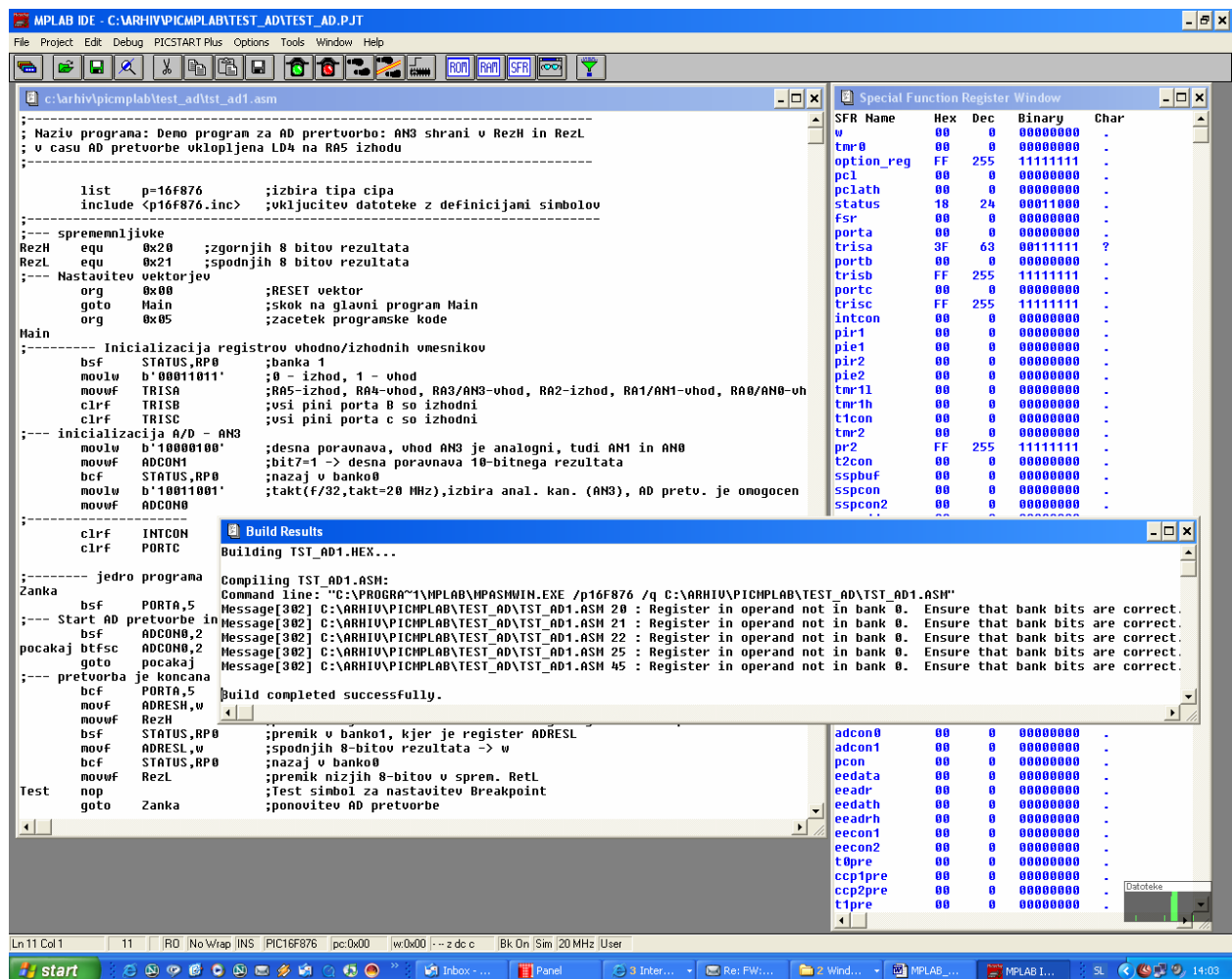
Zatem lahko odprete novo datoteko (**File** → **New**), napišete (ali modificirate) izvorni program v zbirnem jeziku (Assembler) ter ga shranite s (**File** → **Save as...**) (v našem primeru **vaja1.asm**).

V kolikor imate v imeniku že aktualno izvorno datoteko, jo odprete s (**File** → **Open**.) ter jo tako omogočite bodisi za urejanje, bodisi za testno izvajanje (Slika 1-6).



Slika 1-6: Okno za odpiranje datoteke izvornega programa

Program je oblikovan v zbirnem jeziku (Assembler) in vsebuje obvezno končnico **\*.asm**. Odprete lahko tudi okno posebnih datotečnih registrov - **SFR** (Slika 1-7).

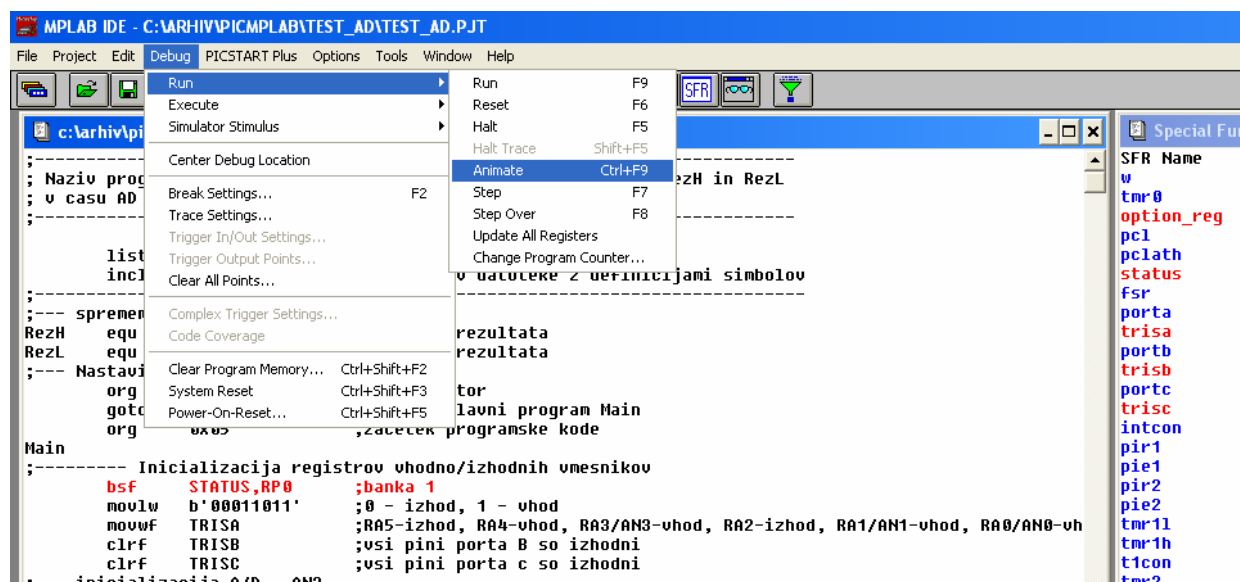


Slika 1-7: Zaslonska slika MPLAB okolja z odprtim izvornim programom in prikazom SFR registrov

**Prevajanje** (aktiviranje **zbornika ASM**) izvedete tako, da v osnovnem oknu **MPLAB** menija **Project** izberete opcijo **Build All** (bližnjica **Ctrl+F10** ali klik na **skrajno desno ikono v orodni vrstici**).

V kolikor se v oknu **Build Results** izpiše: »*Build completed successfully*«, lahko nadaljujete s testiranjem programa, sicer morate najprej odpraviti sintaktične napake v izvornem programu .asm in postopek ponavljati, dokler ne odpravite vse izpisane napake (**Error**). Na morebitna opozorila (**Warning**) se ni potrebno ozirati. Zatem lahko okno **Build Results** zapremo.

**Na voljo so bogate možnosti kontroliranega simulacijskega izvajanja programa** (Slika 1-8) v meniju (**Debug** → **Run** → ...): start programa (**Run F9**), ponovna priprava na start (**Reset F6**), ustavitev (**Halt F5**), izvajanje po korakih - vrsticah (**Step F7**), animirano izvajanje programa (**Animate F6**).



Slika 1-8: Okno za »razhroščanje« (kontrolirano izvajanje programa)

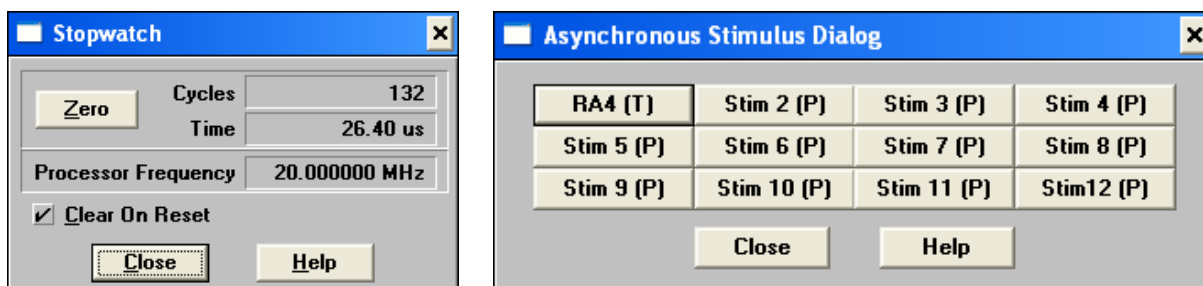
Pomembne lastnosti pri simulacijskem izvajanju programov so: vstavljanje prekinitvenih točk (**Break Settings F2**), stimuliranje bitnih vhodov (**Asynchronous Stimulus**), sledenje vsebine oz. vrednosti registrov (**SFR**) in spremenljivk (**Watch Windows**) ter merjenje časa v ciklih in realni časovni enoti (**Stopwatch**).

Prekinitveno točko (**Breakpoint**) lahko vstavimo/odstranimo tako, da kliknemo na željeno programsko vrstico in z desno tipko miške potrdimo: **Break Point(s)**, pri čemer se vrstica z aktivno prekinitveno točko obarva rdeče.

Od ene do druge prekinitvene točke lahko merimo čas z vgrajeno uro - štoparico, ki jo aktiviramo iz menija (**Window** → **Stopwatch**). S tem se odpre opazovalno okno (Slika 1-9). Vrednost lahko kadarkoli postavimo na nič (Zero).

Bitne vhode (**RA0** do **RA5**, **RB0** do **RB7**, **RC0** do **RC7**) lahko stimuliramo (spreminjamo vrednosti med 0 in 1) tako, da izberemo meni: (**Debug** → **Simulator Stimulus** → **Asynchronous Stimulus** ...). Pojavi se okno (Slika 1-9), v katerem lahko

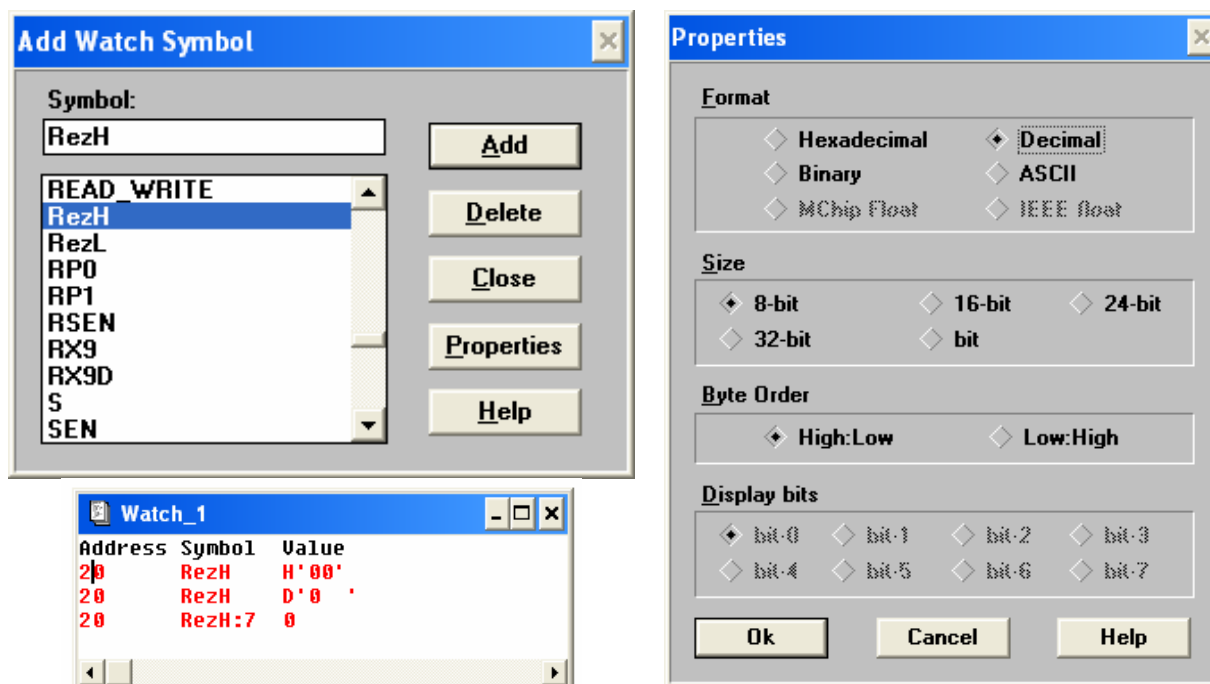
nastavimo do 12 stimulacijskih točk. Če želimo preklapljati vhod **RA4**, izberemo npr. **Stim 1** (klik z desno tipko), kliknemo na **Assign...** in zatem izberemo **RA4** vhod ter z dvoklikom potrdimo izbiro. Iz privzetega pulznega načina stimuliranja lahko preidemo na preklopni (**Toggle**), tako da s ponovnim desnim klikom na gumb **RA4 (P)** izberemo **Toggle** in oznaka gumba se spremeni v **RA4 (T)**.



Slika 1-9: Okno za štoparico in okno za stimuliranje bitnih vhodov

**Vsebinsko posebnih datotečnih registrov** lahko spremljamo v oknu **SFR**, ki si ga namestimo npr. na desno stran. **Vsaka sprememba vsebine registra** med izvajanjem programa se odrazi v **rdeči barvi vrstice** z imenom registra ter 8-bitno vrednostjo v šestnajstiški, desetiški, binarni, in ASCII obliki.

Če želimo spremljati vrednosti poljubnih spremenljivk (simboličnih imen), si moramo aktivirati opazovalno okno v meniju: (**Window** → **Watch Windows** → **New watch window ...**). Izberemo definirano simbolično ime npr. **RezH**, potrdimo s klikom na **Add**, s klikom na **Properties** pa lahko spremenimo izpis vrednosti iz (privzeto 8-bitne šestnajstiške) na 1-, 8-, 24-, 16- ali 32-bitno vrednost v desetiški celoštevilčni ali ASCII (znakovni) notaciji). Tako lahko spremenljivko (pomnilniško lokacijo) izpišemo v oknu Watch\_1 v šestnajstiški (H'), desetiški (D') in binarni obliki (Slika 1-10).



Slika 1-10: Postopek za konfiguriranje opazovalnega okna spremenljivk

Informacijsko koristna je tudi **spodnja - statusna vrstica MPLAB**, ki med drugim prikazuje številko vrstice izvornega programa (**LnXY**), trenutno vrednost programskega števca (**pc:0xYYYY**), trenutno vrednost delovnega registra (**w**) in vrednosti zastavic (**z, dc, c**).

**Opozorilo:** »Razhroščevalni način« izvajanja programa (**Debug Mode**) v simulacijskem okolju **MPLAB SIM** na osebem računalniku ima **določene pomanjkljivosti**: ni mogoče vplivati na napetost na analognem vhodu, PWM vmesnik ne generira signala, **program ne teče v realnem času**, ...). Prave razmere pri testiranju programov se pokažejo šele, ko uporabljamo ciljni mikrokontroler, bodisi v načinu programiranja **FLASH ROM** in poganjanja programa v ciljnem mikrokontroler s priključenimi vhodno/izhodnimi napravami, bodisi v **ICD** »razhroščevalnem načinu«. Simulator lahko s pridom uporabljamo pri učenju in spoznavanju ukazov in programskih postopkov, pri testiranju programskega toka in merjenju časovnih intervalov.

**Za razumevanje tematike je potrebno tudi predznanje** iz osnov programiranja v zbirnem jeziku, kar je na voljo med drugim v [2][4][6][9]. **Pomembna lastnost** Microchip MPLAB programskega okolja je, da lahko **programe izvajamo** (z določenimi omejitvami) tudi v **simulacijskem načinu** (brez uporabe ciljnega sistema) zgolj na osebem računalniku povprečnih zmogljivosti [10].

Vprašanja za utrjevanje:

1. *Kateri tipi datotek (končnice) so nujno potrebni pri kreiranju projekta za programiranje v zbirnem jeziku ?*
2. *Kakšno končnico ima datoteka – strojni (izvršljiv) program v zbirnem jeziku?*
3. *S katerim orodjem pretvarjamo izvorne programe v zbirnem jeziku v strojne programe ?*
4. *Katere možnosti testiranja (poganjanja) programov imamo v MPLAB Simulatorju?*
5. *Na kakšen način izpiše štoparica (Stopwatch) čas izvajanja programa ?*
6. *V katerih številskih formatih lahko spremljamo vrednost spremenljivke v opazovalnem oknu (Watch Window) ?*



## 2. Literatura

- [1] James M. Sibigroth: Understanding small microcontrollers; Prentice Hall, 1993; ISBN 0-13-089129-0, 250 strani
- [2] Nebojša Matić: PIC mikrokontroleri; Mikroelektronika, Beograd 2002; ISBN 86-902189-4-7; 276 strani
- [3] Matjaž Colnarič: Osnove digitalne tehnike v računalništvu; UM-FERI, Maribor, 2002; ISBN 86-435-0435-8; 138 strani
- [4] Miran Rodič, Janez Pogorelc: Navodila za delo z modulom MPU-PIC16F876 (interno gradivo), [[http://www.ro.feri.uni-mb.si/predmeti/mikro\\_el/nav\\_mpu\\_pic.pdf](http://www.ro.feri.uni-mb.si/predmeti/mikro_el/nav_mpu_pic.pdf)]; UM-FERI, Maribor, 2002
- [5] Darko Dužanec: Programiranje PIC-ev v C-ju (serija člankov), Svet elektronike, maj 2004 do julij 2004, številke 109-111, ISSN 1318-4679
- [6] Jernej Škvarč: PIC od začetka (serija 9 člankov), Svet elektronike, marec 2003 do januar 2004, številke 96-105, ISSN 1318-4679
- [7] Microchip: spletna stran proizvajalca PIC mikrokontrolerov, [<http://www.microchip.com/>]
- [8] Microchip: Priročnik za mikrokontroler PIC16F87x (angl.), [[http://www.ro.feri.uni-mb.si/predmeti/mikro\\_el/FTP/PIC16F876\\_30292b.pdf](http://www.ro.feri.uni-mb.si/predmeti/mikro_el/FTP/PIC16F876_30292b.pdf)], 2000, 200 strani
- [9] Microchip: Priročnik za MPLAB ICD (angl.), [[http://www.ro.feri.uni-mb.si/predmeti/mikro\\_el/FTP/ICD\\_51184d.pdf](http://www.ro.feri.uni-mb.si/predmeti/mikro_el/FTP/ICD_51184d.pdf)], 2000, 104 strani
- [10] Microchip: MPLAB V5.70 integrirano programsko okolje (program), [[http://www.ro.feri.uni-mb.si/predmeti/mikro\\_el/FTP/mp57full.zip](http://www.ro.feri.uni-mb.si/predmeti/mikro_el/FTP/mp57full.zip)], 2003, 13 MB
- [11] Brian W. Kernicham, Dennis M. Ritchie: Programski jezik C (slovenski prevod), 1990, UM-FERI Ljubljana, 240 strani
- [12] HI-TECH Software: spletna stran proizvajalca C-prevajalnika, [<http://www.htsoft.com/>]
- [13] HI-TECH Software: priročnik C-prevajalnika HI-TECH PICC Lite (angl.), [<http://www.htsoft.com/files/demo/piccliteman.zip>], 2002, 358 strani, 1,4 MB
- [14] HI-TECH Software: C-prevajalnik HI-TECH PICC Lite (program), [<http://www.htsoft.com/files/demo/picclite-setup.exe>], 2,4 MB
- [15] Miran Rodič: Uvod v programski jezik C, (interno gradivo) [[http://www.ro.feri.uni-mb.si/predmeti/sis\\_meh/vaje/UvodC.pdf](http://www.ro.feri.uni-mb.si/predmeti/sis_meh/vaje/UvodC.pdf)], UM-FERI, 2001, 14 strani
- [16] Hobby Electronics: spletna navodila za uporabo PIC mikrokontrolerov (angl.), [[http://www.interq.or.jp/japan/se-inoue/e\\_pic.htm](http://www.interq.or.jp/japan/se-inoue/e_pic.htm)]
- [17] Hobby Electronics: spletni primeri uporabe PIC mikrokontrolerov (angl.), [[http://www.interq.or.jp/japan/se-inoue/e\\_pic6.htm](http://www.interq.or.jp/japan/se-inoue/e_pic6.htm)]
- [18] MicrochipC.com: spletni primeri C-programov za PIC-e (angl.), [<http://www.microchipc.com/>]
- [19] MicrochipC.com: Bootloader – najpreprostejši programator PICev (angl.), [<http://www.microchipc.com/PIC16bootload/>]
- [20] A. Tetičkovič, B. Brečko, B. Gračner: Mobilni robot za sledenje črte, Uvodni seminar skupine študentov 3.I. Mehatronika, [[http://www.ro.feri.uni-mb.si/predmeti/skup\\_sem/projektu/MobRob\\_Tet\\_Bre\\_Gra.pdf](http://www.ro.feri.uni-mb.si/predmeti/skup_sem/projektu/MobRob_Tet_Bre_Gra.pdf)], UM-FERI, 2003
- [21] J. Horvat, I. Kodrič: Poročilo o izdelavi mobilnega robota, Uvodni seminar skupine študentov 3.I. Mehatronika, [[http://www.ro.feri.uni-mb.si/predmeti/skup\\_sem/projektu/protokol\\_Kodric\\_meha-rudi.pdf](http://www.ro.feri.uni-mb.si/predmeti/skup_sem/projektu/protokol_Kodric_meha-rudi.pdf)], UM-FERI, 2003, 14 strani
- [22] Tekmovanje RoboT200X: spletni članki z opisi mini mobilnih robotov na osnovi PIC mikrokontrolerov, [[http://www.ro.feri.uni-mb.si/tekma/dodatne\\_informacije\\_nasveti.htm](http://www.ro.feri.uni-mb.si/tekma/dodatne_informacije_nasveti.htm)]