Basic ProView Software Tutorial

Written by Sepehr Jafari Naimi (BIHE)

Introduction

This tutorial will teach you how to write, compile, and trace a simple program in ProView. For more information, you can see ProView's help.

Opening ProView

1. Open the ProView from the start menu.



Creating a new File

2. Select *New* from the *File* menu.

🖉 PV 32	
File Edit Search	Project Tool View Debug Options Window Help
New Ctrl+N Open Ctrl+O	
Close Ctrl+F4 Save Ctrl+S	
Save all Save as	
Print Ctrl+P Print Setup	
Exit Alt+F4	
Create a new docume	ent de la constance de la const

3. If you want to write your code in Assembly you should select *Assembler files*, and if you want to write you code in C, you should choose *C Files*. As we want to write the first program in assembly, choose *Assembler files*.



Writing the source code

4. Write the following code. (It is copied from *The 8051 Microcontroller and Embedded Systems* written by *Mazidis*.)

```
ORG OH ;start (origin) at location O
MOV R5,#25H ;load 25H into R5
MOV A. #0 :load 34H into P7
        MOV A, #0
                        ;load 34H into R7
        MOV A, R5
                        ;load 0 into A
                         ; now A = A + R5
        ADD A, R7
                         ;add contents of R7 to A
                        ; now A = A + R7
        ADD A, #12H
                        ;add to A value 12H
                        ;now A = A + 12H
HERE:
        SJMP HERE
                        ;stay in this loop
                          ;end of asm source file
        END
```

Saving

5. Press *Ctrl* + *S* or choose *Save* from the *File* menu.

🗶 PV32	
File Edit Search Project Tool View	Debug Options Window Help
New Ctrl+N Open Ctrl+O Close Ctrl+F4	
Save Ctrl+S ntitled4 asm	
Save all Save as ORG OH MOV R5.#25H	start (origin) at location 0 :load 25H into R5
Print Ctrl+P MOV A, #0 Print Setup MOV A, R5	;load 34H into R7 ;load 0 into A
Exit Alt+F4 ADD A, R7	;now A = A + RS ;add contents of R7 to A ;now A = A + R7
ADD A, #12H	;add to A value 12H ;now A = A + 12H
HERE: SJMP HERE END	;stay in this loop ;end of asm source file
Save this source file	

6. Name your file. Then press *Save*.

Save As				? 🔀
Save in: 🔀	temp	•	← 🗈	💣 🎟 -
File name:	prog2_1			Save
Save as type:	Assembler Files (*,asm;*,a51;*,inc)		•	Cancel

Compiling

7. Press *F9* or select *Make* from the *Project* menu.

Tracing and Debugging

8. Press *Ctrl* + *D* or select *Start* from *Debug*.



9. As it is the first time you debug the file, the following dialog appears. Press Ok.

Debug Options:	
Tool: • Virtual Machine (Simulator)	<u>O</u> k
C <u>R</u> eal Machine (Emulator)	<u>C</u> ancel
C Other Tool	<u>H</u> elp
80C52 Kicrocontroller: Frequency: Crystal (MHz) 11.059	Advanced Options
,	Port Settings

10. Some new windows will be opened:

🧟 PV32				
File Edit Search Project Tool View Debug Options Window Help				
ピュッメ++Gi × 注意: Ei ゅる 🖬 O ,ゑ,ゑ 男 P	≥ J			
C:\\prog2_1.asm ORG OH ;start (origin) at location 0 MOV R5,#25H ;load 25H into R5 MOV A, #0 ;load 34H into R7 MOV A, R5 ;load 0 into A ;now A = A + R5 ADD A, R7 ;add contents of R7 to A ;now A = A + R7 ADD A, #12H ;add to A value 12H	Main Registers (prog2_1) Image: CPU Bank Data Hardware PC 00002 RB 00 @R0 00 P0 FF ACC 00 R0 00 @R1 00 P1 FF PSW 00 R1 00 @DPTR FF P2 FF SP 07 R2 00 X@R0 FF P3 FF DPTRJ0000 R3 00 X@R1 FF TCON 00			
Z Code (prog2_1)	B JO0 H4 JO0 SPX XX THLU JO000 C 0 R5 25 XAREA XX THLU JO000 EA 0 R6 00 Task XX THL2 JO000			
Address Symbol Code Mnemonic ##_2 MOV R5,#25H ;load 25H into R5 Image: Code Image: Code 0000: 7D25 MOV R5,#25 ##_3 MOV A, #0 ;load 34H into R7 00002: 7400 MOV A,#00				
Message Running L51 on C:\Documents and Settings\Sepehr\Desktop\temp\prog2 1.asm				
T: 0:0:0 ms.1 26:85 ns	3:7 INS NUM CAPS			

• Main Registers: This window shows the contents of all of the registers, at the current time.

📕 Code	e (prog2_1)			
Address	Symbol	Code	Mnemonic	
##_2	MOV R5,#2	5H ;load 25H into	R5 2	~
0000:	5	7D25 4	MOV R5,#25	
##_3	MOV A, #0	load 34H into R7		
0002:		7400	MOV A,#00	
## 4	MOV A, R5	;load 0 into A 👘		~

- **Code:** This window shows the contents of the flash memory. In the window:
 - 1) The blue line shows the instruction which will be executed in the next cycle.
 - 2) The violet texts represent your source code.
 - 3) The Mnemonic column represents the assembly equivalent instruction of your code. When your code is in assembly the mnemonic is the same as your code. But, when your code is in C language, it is not.
 - 4) The code column shows the machine equivalent of the instruction.
 - 5) The address column shows where the instruction is located.

11. Press *F7* or select *step into* from *Debug* menu. It executes the blue instruction and updates the values of registers in *Main Registers* window. Press *F7* a few times and see the changes of registers.

🖉 PV32				
File Edit Search Project Tool View Debug Optic	ns Window Help			
	Ctrl+D			
	Ctl+F9			
C:\\prog2_1.asm Step into	F7			
ORG OH ;star Step over	F8	👗 💒 Main Registers (prog2_1) 🛛 🗖 🔀		
 MOV R5,#25H ;load Reset 	Ctrl+F2	CPU Bank Data Hardware		
MOV A, #0 ;load Reset time	ition point Ctrl+E3	PC 0002 RB 00 @R0 00 P0 FF		
* NUV A, RS ;10ac Show exect		ACC 100 HU 00 @H1 00 P1 FF		
ADD A, R7 ;add Evaluate/M	odify Ctrl+M			
now Add watch	F6	DPTB0000 B3 00 X@B1 FF TCON 00		
* ADD A, #12H ;add Toggle brea	knoint ES			
Code (preg 2, 1) Toggle trac	e F4	C 0 R5 25 XAREA X THL10000		
		EA 0 R6 00 Task XX THL20000		
Address Sympol Code Mine	emonic	IE 00 R7 00 TaskP XX PCON 00		
##_2 MOV R5,#25H ;load 25H into R5	W DE #2E			
## 3 MOV A, #0 ;load 34H into R7	74 HJ,#23			
0002: 7400 MC)V A,#00			
## 4 MOV A, R5 ;load 0 into A				
and the second se				
🚺 Message				
Running L51 on C:\Documents and Settings\Se	pehr\Desktop\temp\prog	rog2_1.asm		
Execute the current statement stepping into function calls				

12. Press *Ctrl* + *D* or select *Terminate* from the *Debug* menu to terminate debugging.

🗶 PV32		
File Edit Search Project Tool View	Debug Options Window H	elp
	Terminate Ctrl+	
Code (nrog2_1)	Run Ctl+F Step into F7 Step over F8 Reset Ctrl+ Reset time Show execution point Show execution point Ctrl+ Evaluate/Modify Ctrl+ Add watch F6 Inspect Toggle breakpoint F5 Toggle trace F4	9 F2 F3 M F2 F3 M F2 F3 M F2 F3 M F2 F3 M F2 F3 M F4 F5 F3 F3 F4 F5 F5 F5 F5 F5 F5 F5 F5 F5 F5
Address Symbol Code	Mnemonic	EA 0 R6 00 Task XX THL20000
000A: FF 000B: FF 000C: FF 000D: FF 000D: FF	db OFF db OFF db OFF db OFF db OFF db OFF	
Message 🗗 🗆 🗙 Start the Debugging Mode		

Creating a new Project

When you write a big program, you would better, split your code into some files and put the relevant functions in the same files. To do so, you should make a new project and add your files to the project.



13. Select *New* from the *Project* menu.

14. If you want to change the name of the project, type another name in front of *Name*. If you want to change the path, where the project file will be stored, press the *Browse* button.

💹 New Pro	ject 🛛 🔀
Name:	nts and settings\sepehr\desktop\temp\noname.pr
Directory:	c:\\temp\
Type:	
	C 80C51XA
ОК	Cancel Browse Help

15. Press Ok.

16. Select Add j	file from the P	<i>roject</i> menu.
------------------	-----------------	---------------------

ើ PV32- tst		
File Edit Search	Project Tool View Debug Options Window Help	
	Project Tool View Debug Options Window Heip New Open Image: Close Image: Close Image: Close Image: Close Save Type of Target Add file Alt+Ins Image: Close Image: Close Delete file Alt+FP Add file Alt+FP Make F9 Build all Shift+F9 Link Ctrl+F9	
C:\Docume	xdocuments and settings\sepehr\desktop\temp\tst.prj NTS AND SETTINGS\SEPEHR\DESKTOP\TEMP\TST.PRJ {8051} code=10 external data=0 internal data=8.0	

17. Select the ASM file which you have created in the previous steps. Then press *Open*.

📕 PV32- tst			
File Edit Search Project Tool View Debug Options Window Help			
Add File			
Look in: 🗁 temp 🔽 🔶 🖻 💣 🖽 -			
prog2_1.asm			
File name: prog2_1.asm Open			
Files of type: Source(",c,",h,",c51,",asm,",a51,",inc)			
Project - tst.prj			
TST.PRJ {8051} code=10 external data=0 internal data=8.0			
1:0 INS NUM CAPS			

18. The file has been added to the project. Now open your source file by double clicking on the name of it.



19. Debug and trace the project by pressing Ctrl + D and F7 if you like to.

Making Hex File

ProView does not normally make hex files. But you sometimes need it; For example, when we want to feed the Proteus software. To make the ProView software, provide the hex file, go through the following steps:

- 20. Select *Project* from the *Options* menu. The following window appears.
- 21. Click on *L51*.
- 22. Click on Linker.
- 23. Activate Intel Hex.
- 24. Press OK.

Options			
Options: Environment Directories C51 A51 Off Bank switching Kernel Listing Flash Monitor	Libraries: RTX-51 Kerner C C51x.LIB RTX-51 Monitor Starting addresses: Code [0-FFFF[: 0000 Idata [0- Xdata [0-FFFF[: 0000 Data [0-	Al Misc: Ram Size: 128 Intel Hex 3 Debug Info Intel Hex 3 FF[: 0000 Bit [0-FF[: 0000 FF[: 0000 Bit [0-FF[: 0000	
OK Defaults Cancel Help			