



Middle East Technical University
Department of Computer Engineering

PIDE

Emulator and Development Environment for

CEng Embedded System Card

USER MANUAL

by

SimSys Corporation

Table of Contents

1.	INTRODUCTION.....	3
2.	GRAPHICAL USER INTERFACE.....	4
2.1	GENERAL VIEW.....	4
2.2	MENUS.....	5
3.	ASM++ LANGUAGE FORMAT.....	8
3.1	GENERAL SPECIFICATIONS.....	8
3.2	IDENTIFIERS.....	8
3.3	RESERVED WORDS.....	9
3.4	DELIMITERS AND OPERATORS.....	9
3.5	VARIABLES.....	9
3.6	NUMBERS.....	10
3.7	EXPRESSIONS.....	10
3.8	STATEMENTS.....	11
3.9	COMMENTS.....	13
3.10	EXTENDED INSTRUCTION SET.....	13
4.	START USING PIDE.....	15

1. INTRODUCTION

PIDE is a complete integrated development and simulation environment designed for CENG336 Embedded Systems Board by SimSys Corporation. This document gives information about the usage of the program, describes the graphical user interface and introduces a new language ASM++.

2. GRAPHICAL USER INTERFACE

2.1 GENERAL VIEW

Below in Figure 2.1, the GUI of the PIDE program, showing the menus, toolbars, tabs, workspace view and the status bar can be found. It shows the case with an opened project, and two opened asm files. The workspace view is also present on the left hand side. The program is able to handle multiple opened files using a tabbed view

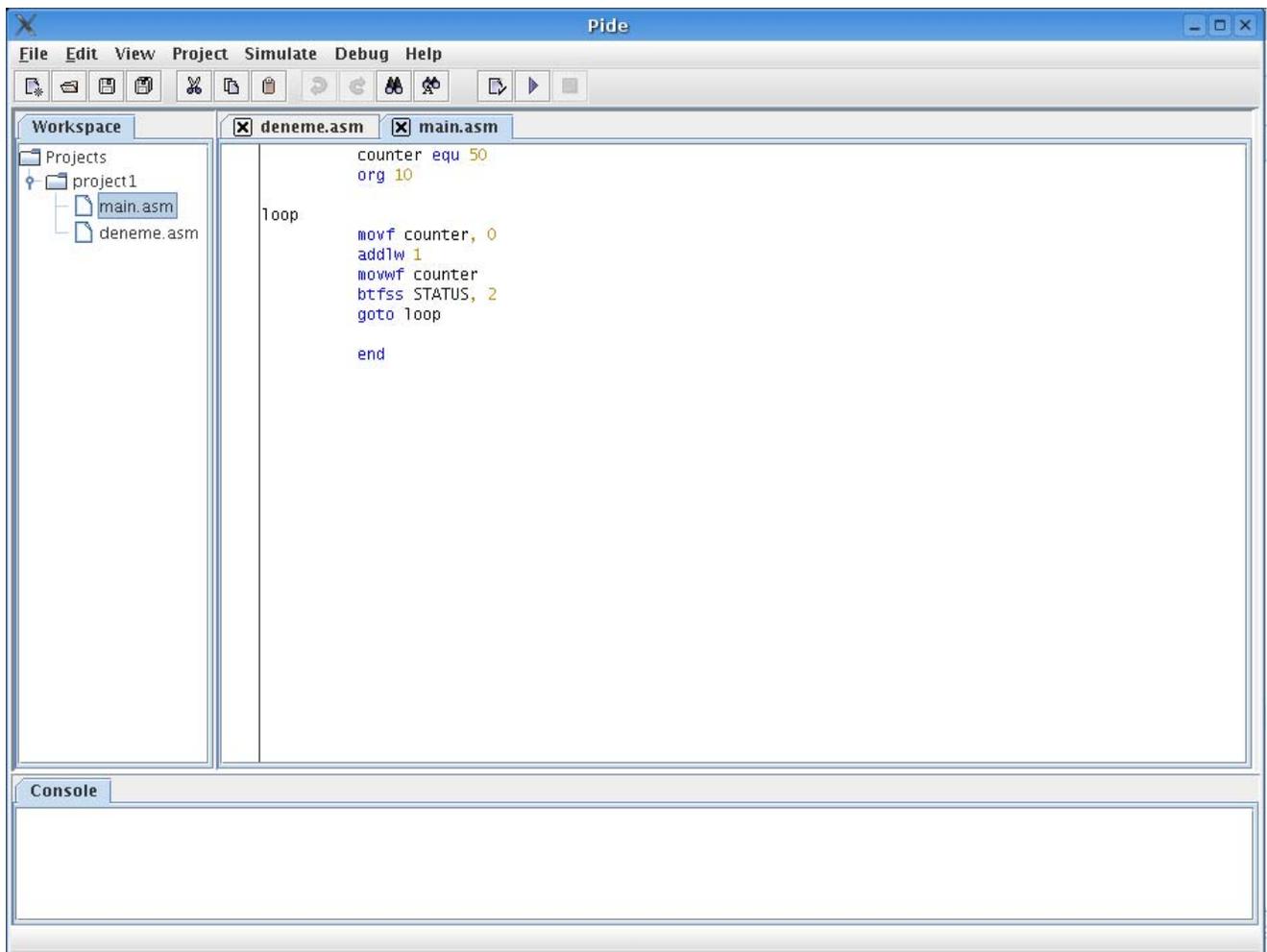


Figure 2.1

In Figure 2.2, the menu bar of the PIDE is shown. The menu items will be explained in detail in the following sections.



Figure 2.2

In Figure 2.3, the toolbar of the PIDE is shown. Here exist shortcuts of the frequently used operations in the menu bar.

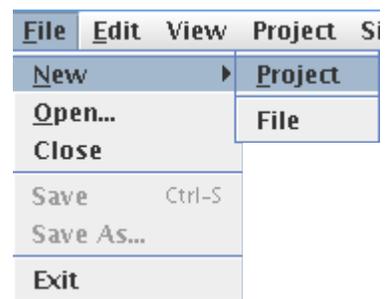


Figure 2.3

2.2 MENUS

FILE MENU

New	Project	Create a new project.
	ASM File	Create a new ASM file.
	ASM++ File	Create a new ASM PlusPlus file.
Open...		Open an existing file.
Close		Close the current file.
Save		Save the current file.
Save As...		Save the current file with a different name or save to a different place.
Exit		Quit from the program.



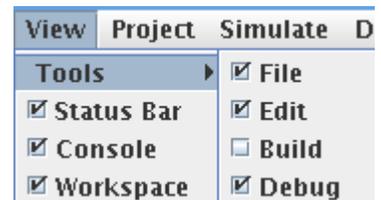
EDIT MENU

Undo	Undo the last action.
Redo	Redo the last undo action.
Cut	Cut the selected item.
Copy	Copy the selected item.
Paste	Paste the last cut or copied item.
Select All	Select and highlight the whole text in the active tab.
Find/ Replace	Find and replace given word in the current file.



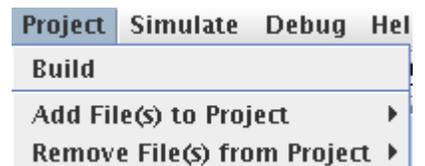
VIEW MENU

Tools	Show/Hide the toolbars of File, Edit, Build and Debug menus.
Status Bar	Show/Hide the status bar.
Console	Show/Hide the console view.
Workspace	Show/Hide the workspace view.



PROJECT MENU

Build	Build the current project.
Add File to Project	Add a new file to the current project.
Remove File from Project	Remove a file from the current project.



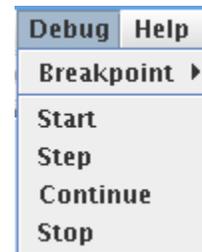
SIMULATE MENU

Run	Open the simulator window.
Start	Start the simulation.
Pause	Pause the simulation.
Continue	Continue the simulation.
Stop	Stop the simulation.



DEBUG MENU

Breakpoint	Add or Remove breakpoints.
Start	Start the debugging process.
Step	Execute one instruction.
Continue	Execute until the next breakpoint.
Stop	Stop the debugging process



HELP MENU

About PIDE...	Show brief information about the program.
----------------------	---



3. ASM++ LANGUAGE FORMAT

ASM++ is simply an improvement on assembly language, including some new instructions and introducing some high level language concepts such as control and repetitive structures and variable definitions. The name of the language is ASM++ (ASM plus plus), and the file extension is “.asmpp”.

3.1 GENERAL SPECIFICATIONS

- ASM and ASM++ in PIDE are case sensitive. Instructions and other reserved words (if, else, while, etc.) should be written in lower-case letters.
- PIC 16F877 constants (STATUS, PORTA, etc.) are recognized and must be used in upper-case.
- White spaces (space character, tab character and end-of-line) serve to separate tokens; otherwise, they are ignored.
- No token can extend past end-of-line.
- Spaces may not appear inside any token except character and string literals.
- A comment begins with two forward slashes (as in C++) or with a semicolon (as in assembly language) and extends to end of line.
- There cannot be more than one statement in a line.
- No semicolons exist at the end of statements. (In fact, that does not matter since, after a semicolon, the rest of the line is considered as comment.)

3.2 IDENTIFIERS

Identifiers start with a letter and contain letters and digits.

3.3 RESERVED WORDS

The following keywords are reserved in ASM++ in addition to the instructions in ASM:

addff	subff	addwff	subwff	swapff	
iorwff	andwff	xorwff	movff		
if	else	for	while	do	var

3.4 DELIMITERS AND OPERATORS

One-character delimiters: : ; , () EOF

One-character operators: ! < = > '

Two-character delimiters: //

Two-character operators: == != >= <= && ||

3.5 VARIABLES

Variables are introduced by the declaration of the form:

```
var var_name var_address
```

This declaration reserves the given var_address for that variable. It is the user's responsibility to use the variable in the correct bank.

Example:

```
var var1 0x121
```

3.6 NUMBERS

A constant consists of a sequence of one or more digits in decimal, binary or hexadecimal format. Example usage is shown below.

Binary number: Starts with a "B" or "b" (which stands for binary) and contains digits 1, 0 enclosed in single quotes.

Ex. B`10011101` or b`10011101`

Hexadecimal number: Starts with "0x" and contains 0-9 and A-F.

Ex. 0x45AF

Decimal number: Contains digits 0-9.

Ex. 45 or 127

3.7 EXPRESSIONS

In ASM++, expressions are defined as below:

<expr> : <label> == <label> | <label> != <label> |
<label> > <label> | <label> < <label> |
<label> >= <label> | <label> <= <label>

<label> : <variable> | <number>

<variable> is the variable declared using **var** keyword,

<number> is the number represented in binary, decimal or hexadecimal format.
The numbers defined with **equ** keyword are also included here.

endif

- **Loop Statements**

The compiler supports **while**, **do-while** and **for** loops. The curly braces are compulsory regardless of the number of statements inside the loop. The syntaxes of the loop statements are as follows:

```
while (hede)
```

```
{
```

```
.....
```

```
.....
```

```
}
```

```
for (expr1; expr2; expr3)
```

```
{
```

```
.....
```

```
}
```

```
do
```

```
{
```

```
.....
```

```
}
```

```
while (hede)
```

3.9 COMMENTS

The comments are specified by a semicolon or two forward slashes. It will comment out the characters until the end of line.

3.10 EXTENDED INSTRUCTION SET

ASM++ provides a bunch of new instructions together with the basic PIC instruction set. Using these new instructions, it will be possible to do arithmetic operations between two file registers without using the working register WREG in between.

These instructions are:

addff v1 v2 : (V1 ← V1 + V2)

Adds the value of v2 to v1, and writes the result back to v1.

subff v1 v2 : (V1 ← V1 - V2)

Subtracts the value of v2 from v1, and writes the result back to v1.

addwff : (W ← V1 + V2)

Adds the value of v2 to v1, and writes the result to WREG.

subwff : (W ← V1 - V2)

Subtracts the value of v2 from v1, and writes the result to WREG.

swapff : (Temp ← V1, V1 ← V2, V2 ← Temp)

Swaps the values of v1 and v2.

iorwff : (V1 ← V1 OR V2)

Takes the OR of v1 and v2, and writes the result to WREG.

andwff : (V1 ← V1 AND V2)

Takes the AND of v1 and v2, and writes the result to WREG.

xorwff : (V1 ← V1 XOR V2)

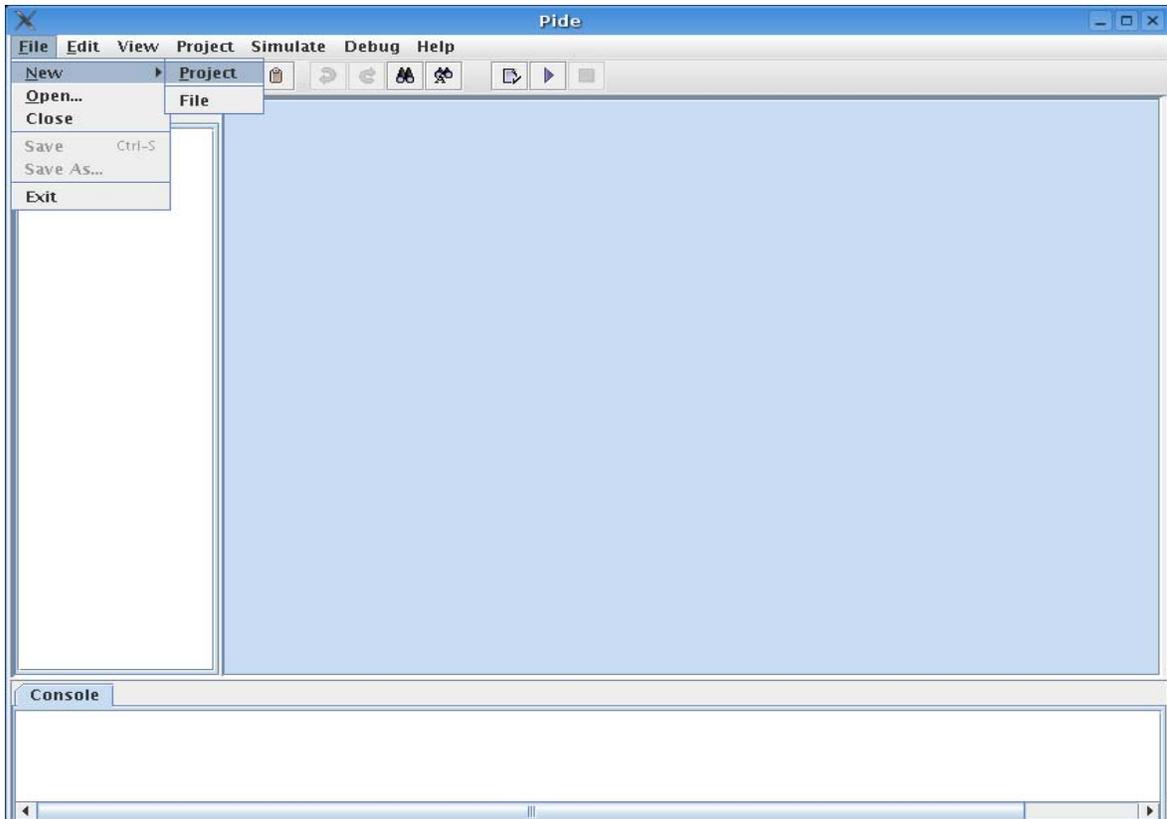
Takes the XOR of v1 and v2, and writes the result to WREG.

movff : (V2 ← V1)

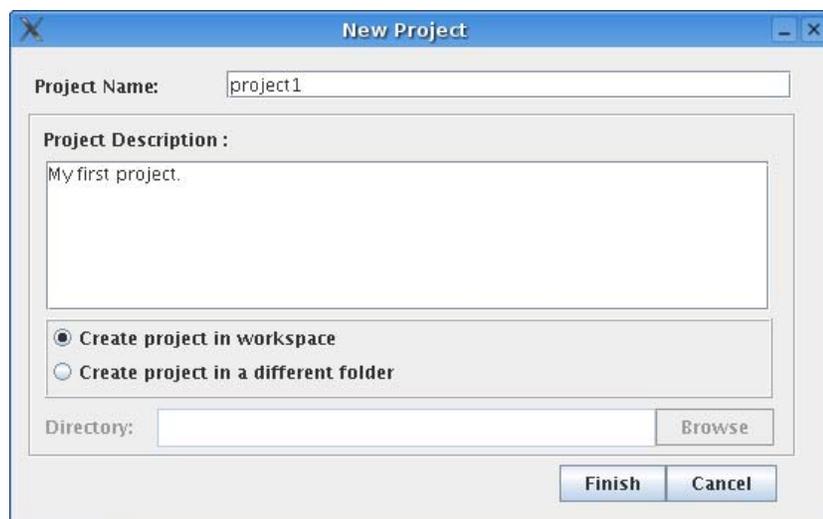
Copies the value of v1 to v2.

4. START USING PIDE

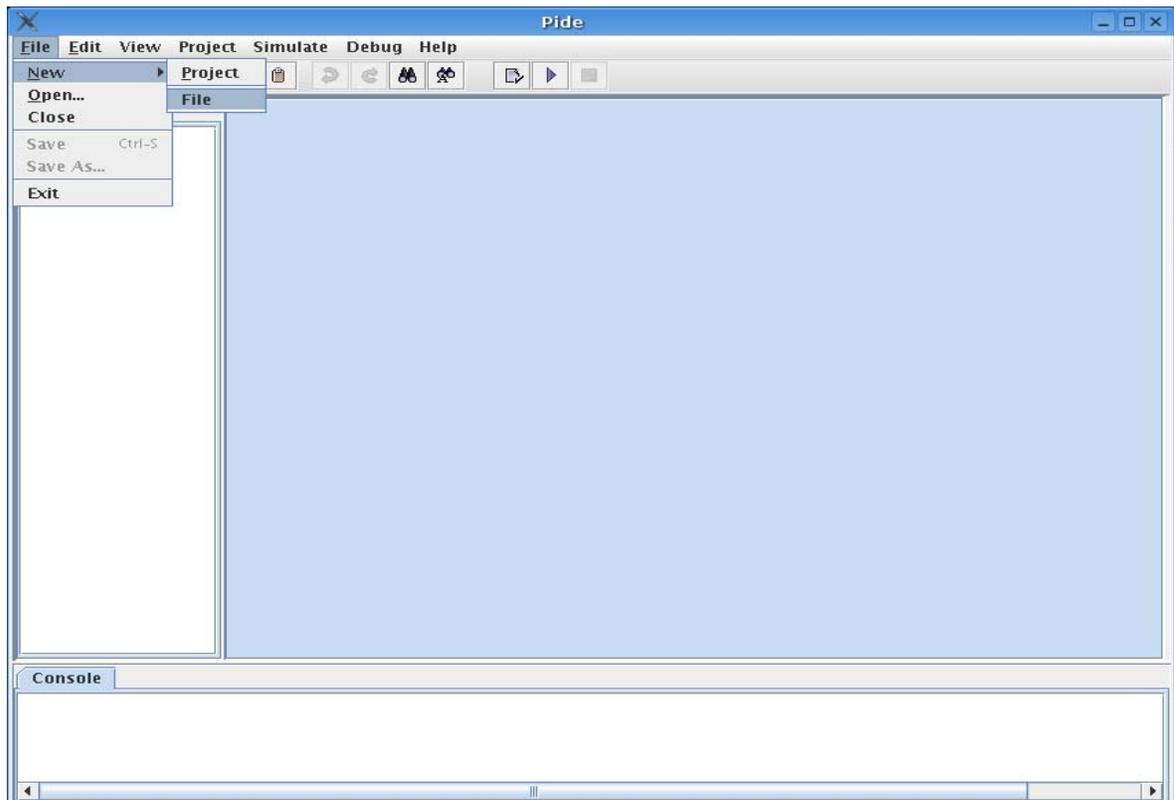
- First, create a new project through *File* → *New* → *Project*.



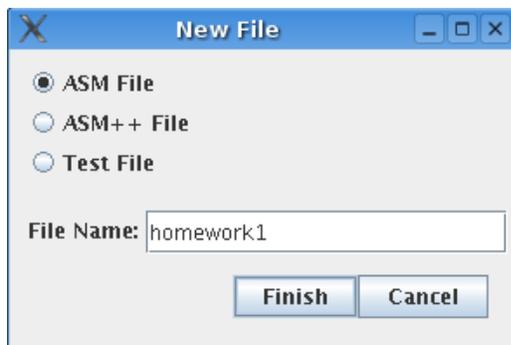
- Enter the project name and description, and choose whether to create the project in workspace or in a different folder.



- Create a new file through *File* → *New* → *File*.

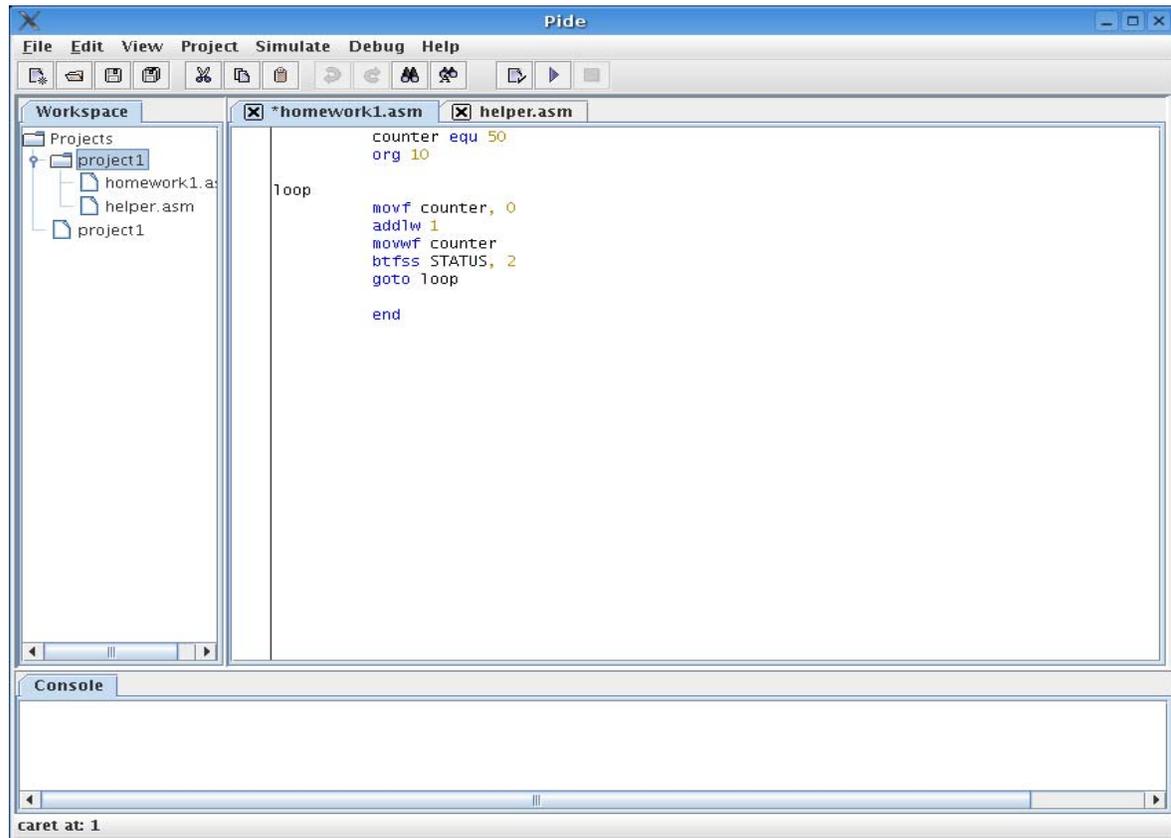


- Choose ASM or ASM++ file and enter a name.

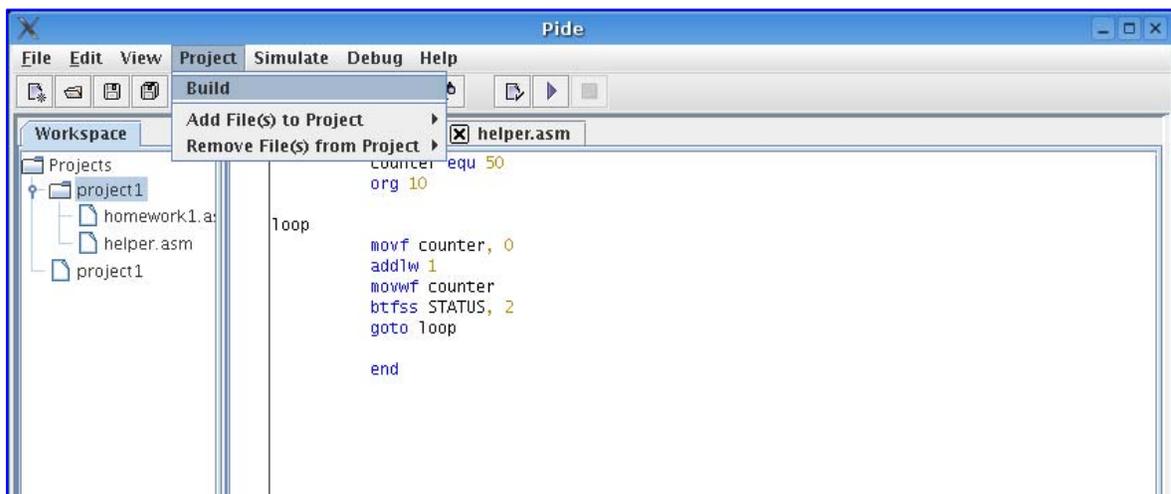


- Multiple files can be created which will be shown in a tabbed manner. A closed file can be opened by double-clicking or by right-clicking on its name on the workspace pane.

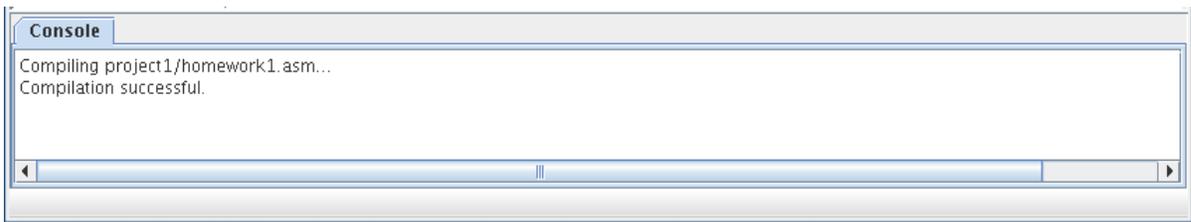
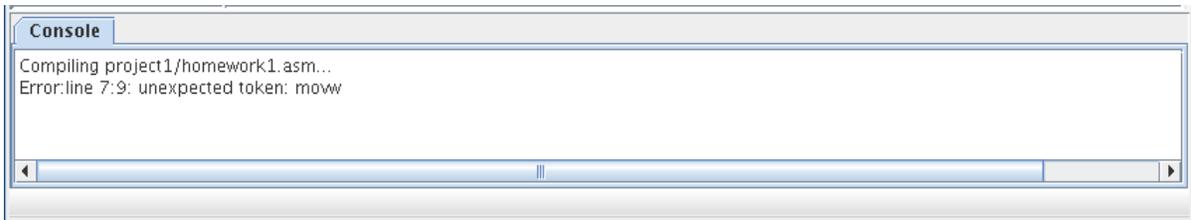
- Write the source code, which will automatically be colored.



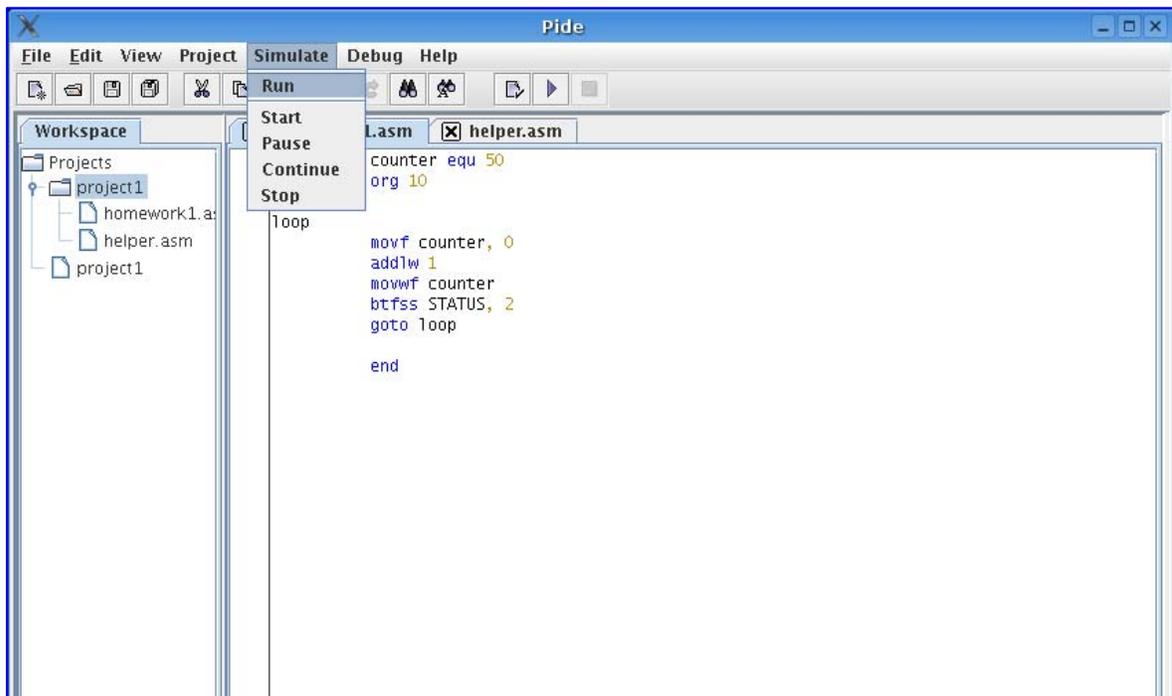
- Compile the file through *Project* → *Build*.



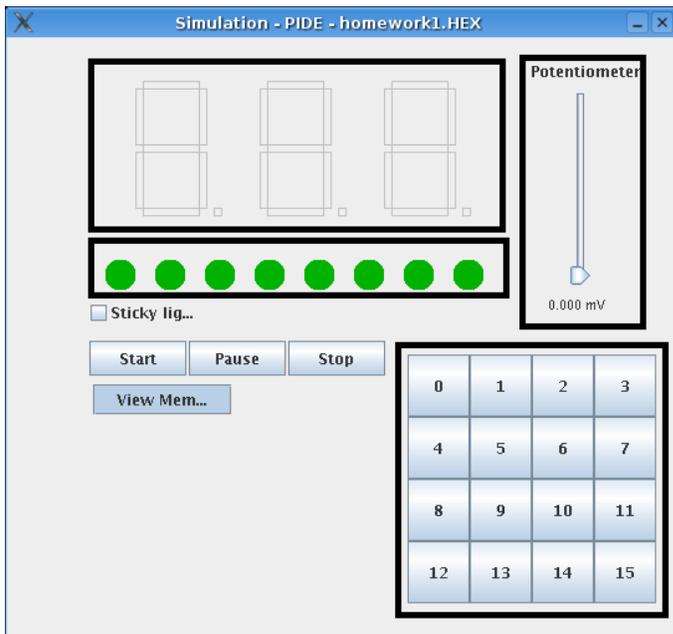
- If the file contains errors, it can be seen on the console pane. Otherwise, "Compilation Successful" message will be seen.



- Simulate the code through *Simulate* → *Run*.



- The board view and the memory view are opened. *Start/Pause/Stop* the simulation as intended. Console pane shows the simulated instructions.



Address	Value	Address	Value	Address	Value	Address	Value		
001h	TMR0	0	081h	OPTIO...	0	101h	TMR0	0	
002h	PCL	0	082h	PCL	0	102h	PCL	0	
003h	STATUS	0	083h	STATUS	0	103h	STATUS	0	
004h	FSR	0	084h	FSR	0	104h	FSR	0	
005h	PORTA	0	085h	TRISA	0	105h	UNIMPLEME...	185h	UNIMPLEME...
006h	PORTB	0	086h	TRISB	0	106h	PORTB	0	
007h	PORTC	0	087h	TRISC	0	107h	UNIMPLEME...	187h	UNIMPLEME...
008h	PORTD	0	088h	TRISD	0	108h	UNIMPLEME...	188h	UNIMPLEME...
009h	PORTE	0	089h	TRISE	0	109h	UNIMPLEME...	189h	UNIMPLEME...
00Ah	PCLATH	0	08Ah	PCLATH	0	10Ah	PCLATH	0	
00Bh	INTCON	0	08Bh	INTCON	0	10Bh	INTCON	0	
00Ch	PIR1	0	08Ch	PIE1	0	10Ch	EEDATA	0	
00Dh	PIR2	0	08Dh	PIE2	0	10Dh	EEADR	0	
00Eh	TMR1L	0	08Eh	PCON	0	10Eh	EEDATH	0	
00Fh						10Fh			
010h						110h			
011h						111h			
012h						112h			
013h						113h			
014h						114h			
015h						115h			
016h						116h			
017h						117h			
018h						118h			
019h						119h			
01Ah						11Ah			
01Bh						11Bh			
01Ch						11Ch			
01Dh						11Dh			
01Eh						11Eh			
01Fh						11Fh			
020h						120h			
021h						121h			
022h						122h			
023h						123h			
024h						124h			
025h						125h			
026h						126h			
027h						127h			
028h						128h			
029h						129h			
02Ah						12Ah			
02Bh						12Bh			
02Ch						12Ch			
02Dh						12Dh			
02Eh						12Eh			
02Fh						12Fh			
030h						130h			
031h						131h			
032h						132h			
033h						133h			
034h						134h			
035h						135h			
036h						136h			
037h						137h			
038h						138h			
039h						139h			
03Ah						13Ah			
03Bh						13Bh			
03Ch						13Ch			
03Dh						13Dh			
03Eh						13Eh			
03Fh						13Fh			
040h						140h			
041h						141h			
042h						142h			
043h						143h			
044h						144h			
045h						145h			
046h						146h			
047h						147h			
048h						148h			
049h						149h			
04Ah						14Ah			
04Bh						14Bh			
04Ch						14Ch			
04Dh						14Dh			
04Eh						14Eh			
04Fh						14Fh			
050h						150h			
051h						151h			
052h						152h			
053h						153h			
054h						154h			
055h						155h			
056h						156h			
057h						157h			
058h						158h			
059h						159h			
05Ah						15Ah			
05Bh						15Bh			
05Ch						15Ch			
05Dh						15Dh			
05Eh						15Eh			
05Fh						15Fh			
060h						160h			
061h						161h			
062h						162h			
063h						163h			
064h						164h			
065h						165h			
066h						166h			
067h						167h			
068h						168h			
069h						169h			
06Ah						16Ah			
06Bh						16Bh			
06Ch						16Ch			
06Dh						16Dh			
06Eh						16Eh			
06Fh						16Fh			
070h						170h			
071h						171h			
072h						172h			
073h						173h			
074h						174h			
075h						175h			
076h						176h			
077h						177h			
078h						178h			
079h						179h			
07Ah						17Ah			
07Bh						17Bh			
07Ch						17Ch			
07Dh						17Dh			
07Eh						17Eh			
07Fh						17Fh			
080h						180h			
081h						181h	OPTIO...	0	
082h						182h	PCL	0	
083h						183h	STATUS	0	
084h						184h	FSR	0	
085h						185h	UNIMPLEME...	UNIMPLEME...	
086h						186h	TRISB	0	
087h						187h	UNIMPLEME...	UNIMPLEME...	
088h						188h	UNIMPLEME...	UNIMPLEME...	
089h						189h	UNIMPLEME...	UNIMPLEME...	
08Ah						18Ah	PCLATH	0	
08Bh						18Bh	INTCON	0	
08Ch						18Ch	EECON1	0	
08Dh						18Dh	EECON2	0	
08Eh						18Eh	RESERVED		
08Fh						18Fh			
090h						190h			
091h						191h			
092h						192h			
093h						193h			
094h						194h			
095h						195h			
096h						196h			
097h						197h			
098h						198h			
099h						199h			
09Ah						19Ah			
09Bh						19Bh			
09Ch						19Ch			
09Dh						19Dh			
09Eh						19Eh			
09Fh						19Fh			
0A0h						1A0h			
0A1h						1A1h			
0A2h						1A2h			
0A3h						1A3h			
0A4h						1A4h			
0A5h						1A5h			
0A6h						1A6h			
0A7h						1A7h			
0A8h						1A8h			
0A9h						1A9h			
0Aah						1Aah			
0Abh						1Abh			
0ACh						1ACh			
0ADh						1ADh			
0AEh						1AEh			
0AFh						1AFh			
0B0h						1B0h			
0B1h						1B1h			
0B2h						1B2h			
0B3h						1B3h			
0B4h						1B4h			
0B5h						1B5h			
0B6h						1B6h			
0B7h						1B7h			
0B8h						1B8h			
0B9h						1B9h			
0BAh						1BAh			
0Bbh						1Bbh			
0BCh						1BCh			
0BDh						1BDh			
0BEh						1BEh			
0BFh						1BFh			
0C0h						1C0h			
0C1h						1C1h			
0C2h						1C2h			
0C3h						1C3h			
0C4h						1C4h			
0C5h						1C5h			
0C6h						1C6h			
0C7h						1C7h			
0C8h						1C8h			
0C9h						1C9h			
0CAh						1CAh			
0CBh						1CBh			
0CCh						1CCh			
0CDh						1CDh			
0CEh						1CEh			
0CFh						1CFh			
0D0h						1D0h			
0D1h						1D1h			
0D2h						1D2h			
0D3h						1D3h			
0D4h						1D4h			
0D5h						1D5h			
0D6h						1D6h			
0D7h						1D7h			
0D8h						1D8h			
0D9h						1D9h			
0DAh						1DAh			
0DBh						1DBh			
0DCh						1DCh			
0DDh						1DDh			
0DEh						1DEh			
0DFh						1DFh			
0E0h						1E0h			
0E1h						1E1h			
0E2h						1E2h			
0E3h						1E3h			
0E4h						1E4h			
0E5h									