

Application Note - MPLAB IDE Guide

Creating a new project

The easiest way to create a new project is by using the Project Wizard. The followings are the step by step instructions of how to create a new project for your iMicro Controller board.

	1.	From	the	project	menu.	choose	Proi	iect	Wizard	Ι.
--	----	------	-----	---------	-------	--------	------	------	--------	----

MPLAB IDE v	7.10	
File Edit View	Project Debugger Programmer Tools Configure Window Hel	p
🗋 🗅 🚅 🗔 🛛	Project Wizard	
Untitled Wo	New Open Close Set Active Project	
	Quickbuild (no .asm file)	
	Clean Build Options	
	Find in Project Files	
	Save Project Save Project As Add Files to Project	



2. Press Next and Select the device. For iMicro Controller board, the device is PIC18F4620.

roject Wizard Step One: Select a device		ا پر
	De <u>v</u> ice: ■ PTC18746201	
	< <u>B</u> ack <u>N</u> ext > Cancel	Help

3. Then select the language toolsuite as Microchip C18 Toolsuite.

roject Wizard Step Two: Select a language toolsuite	Select Language Toolsuite by MPLAB-C18
Active Toolsuite: Microchip C18 Toolsuite	
Toolsuite Contents MPASM Assembler (mpasmwin.exe) MPLINK Object Linker (mplink.exe) MPLAB C18 C Compiler (mcc18.exe) MPLIB Librarian (mplib.exe)	Highlight each Toolsuite to confirm the path of utilities
Location C:\Program Files\Microchip\MPASM Suite\MPAsmWin.exe Browse.	
Help! My Suite Isn't Listed!	Select Compiler installation path
< <u>B</u> ack <u>N</u> ext > Cancel H	elp



4. Name your project and specify the project directory.

Step Three: Name your project	Ē
Project Name newProj	
Project Directory	
c:\newPro	Browse
	<pre></pre>

5. Add all the .c, .h, .lib and .lkr files from iMicro11 folder into your newProj.

Project Wizard	×
Step Four: Add any existing files to your project	/ _@
Micro11 Add >> CMPS03.c CMPS03.h Micro1.lib Micro1.mcp Micro1.mcs Micro1.mcw Micro1.mcw Micro1.mcw Micro1.mcw Micro1.mcw Micro1.mcw Micro1.mcw MotionCtrl.h Check the box to copy the file to the project directory. Click the filename to edit the name of the local copy.	>
< Back Next > Cancel Help	





6. Check all the checkboxes so that all the files are copied into a newProj folder.

roject Wizard Step Four: Add any existing files to your proj	ect	<u>ک</u> چر
MotionCtrl.h RC4620.h RC4620i.lkr RCServo.h FF.c RF.h serial_com.h untitled.mcw MCC18 Minimon_v2228b Minimon_v2228b	Add >> Remove	 ✓ ADC.h C:\iMicro11\ADC.h ✓ CMPS03.c C:\iMicro11\CMP ✓ CMPS03.h C:\iMicro11\CMP ✓ iMicro1.lib C:\iMicro11\IMicro ✓ Init.c C:\iMicro11\Init.c ✓ main.c C:\iMicro11\Init.c ✓ MotionCtrl.h C:\iMicro11\Mot ✓ RC4620.h C:\iMicro11\RC4E ✓ PC4620:lkc C:\iMicro11\RC4E ✓ Check the box to copy the file to the project directory. Click the filename to edit the name of the local copy.
	Back Next	Cancel Help

7. Press Next> and a new project is successfully created.

Project Wizard		×		
1473	Summary			
2:0	Click 'Finish' to create the project with these parameters.			
Ver	Project Parameters			
Device: PIC18F4620				
	Toolsuite: Microchip C18 Toolsuite			
	File: c:\newProj\newProj.mcp			
	A new workspace will be created, and the new project added to that workspace.			
	<pre> < Back Finish Cancel Help</pre>			



8. Press Finish and you will see the project window as below with all the files under their respective folder.



9. Then Compile the program by pressing Ctrl+F10 or press the button and you should be able to see the output window as below.





After creating a new project using the above simple steps, you can start your software adventure by adding your own .c and .h files into your newly built project. For all the useful functions provided by iMicro1.lib, please refer to the iMicro_Software Guide.

How to edit the linker file

In our iMicro software, 2 sections of data memory are used to store the variables. As you can see in the linker file (RC4620i.lkr), it is initialized as below:

SECTION NAME=LIB_DATA RAM=gpr1 SECTION NAME=MAIN_DATA RAM=gpr3 Gpr1 is used for data relating to the iMicro library and the grp3 is used to store the user variables.

Declaring the variables

New variables can be declared in the MAIN_DATA section and the proper way of declaring those variables can be seen in main.c as below:

#pragma udata MAIN_DATA
 char HeartCount;
 char DelayCounter;

. //your new variables

#pragma udata

As mentioned above, each data bank can store 256 bytes of data. So if your variable declaration exceeds 256 bytes, the linker will give an error as follow :

MPLINK 4.06, Linker

Copyright (c) 2006 Microchip Technology Inc. Error - section 'MAIN_DATA' has a memory 'gpr3' which can not fit the section. Section 'MAIN_DATA' length=0x000004b2. Errors : 1

In such case, you will need to create another data section for your extra variables. For example, if grp4 is to used as a new section to store the extra data, you will need to add the following code into the linker file. SECTION NAME=EXTRA_DATA RAM=gpr4

After initialized the section, all the extra data can be declared in the EXTRA_DATA section as follow:

```
#pragma udata EXTRA_DATA
    int test_data[10];
    char sample;
```

//your extra variables

#pragma udata