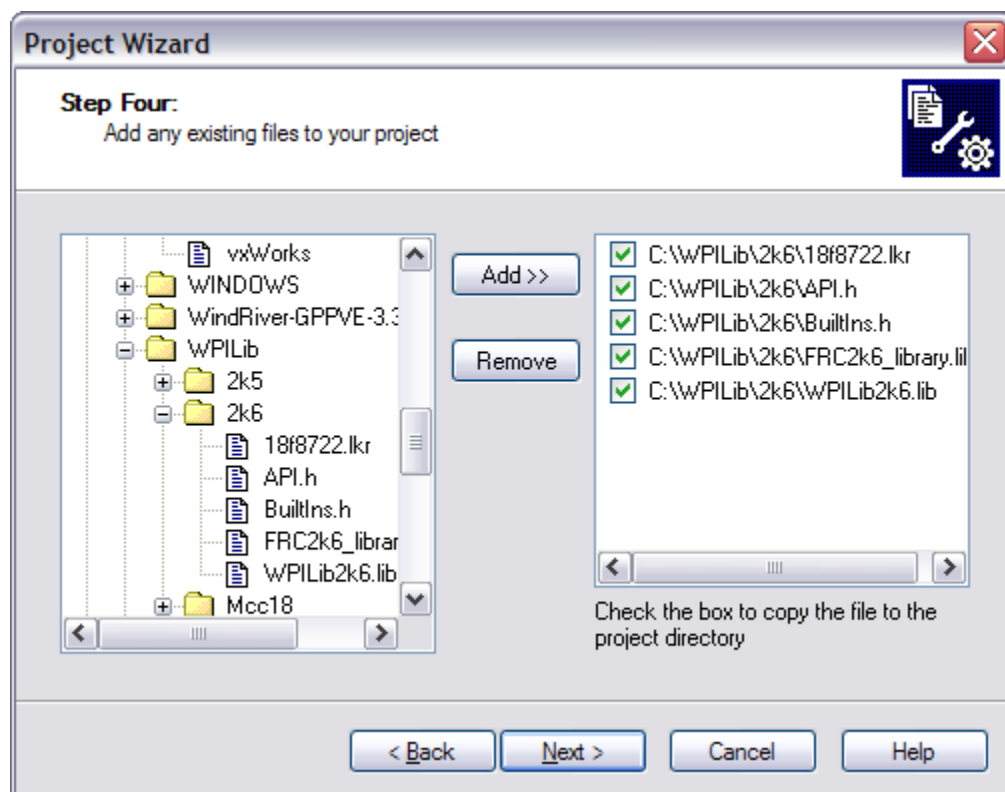
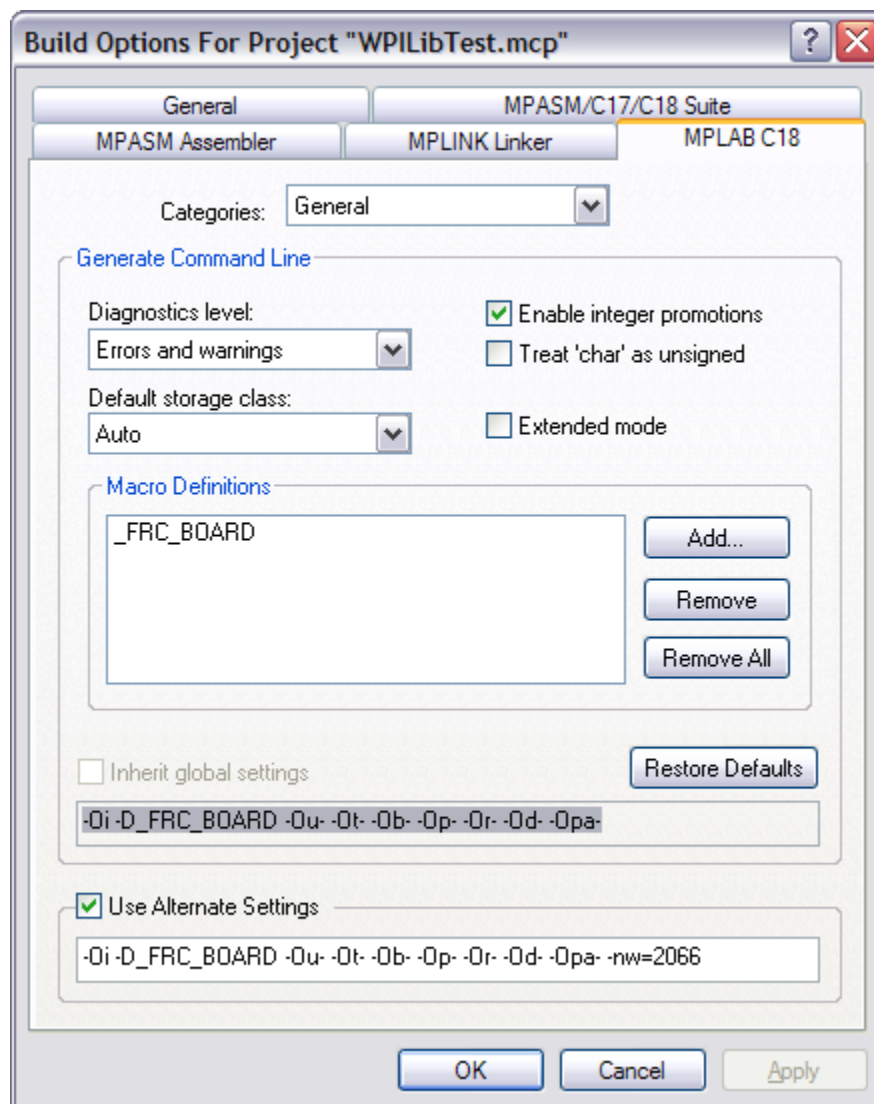
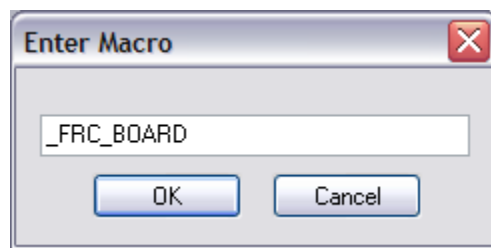


## Using WPILib with Microchip tools

1. Install MPLab taking all the defaults
2. Install C18 Compiler – check all boxes in Configuration Options dialog
3. Install IFI Loader
4. Start MPLab
5. Start Project Wizard (Project / Project wizard...)
6. Select 18F722 for the device type (FRC)
7. Select “Microchip C18 Toolsuite” for the Active Toolsuite
8. On the Step 3 dialog window, enter a project name and directory
9. On the Step 4 dialog window, add the WPILib files and check the boxes to cause the files to be copied into the project directory. These should come from the 2k6 directory.

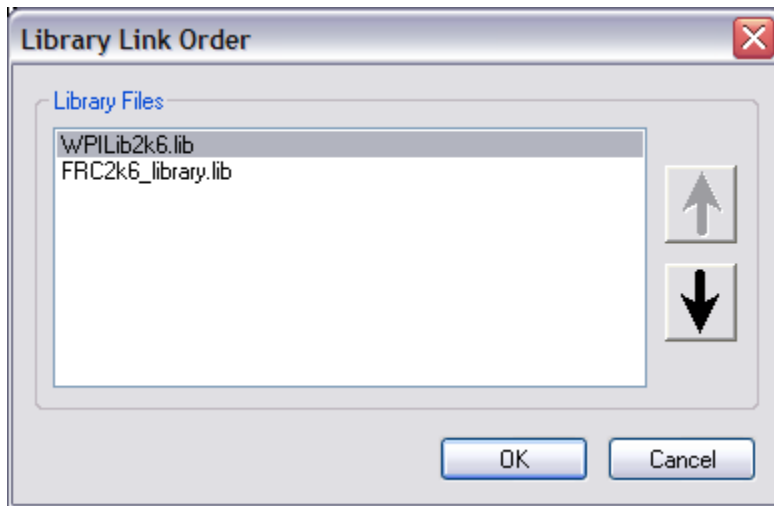


10. Hit Finish when prompted.
11. Add a source file to the project. I'd suggest starting with the template
12. In the Project build options add `_FRC_BOARD` as a macro definition, enable Integer Promotions, and add the `nw=2066` option to suppress warnings for `printf` function calls.



13. On the MPLink Linker tab, be sure to select "Supress COD-file generation".

14. And, last, right-click on the Library Files and make sure the link order is as shown here. If it isn't, make adjustments by using the up/down arrows.



15. Now, Build All should generate a valid .hex file that can be loaded into the robot controller using the IFI Loader.

Here is a sample program that you can start with. If you run this program with a dongle attached to the competition port you'll see (in sequence):

Autonomous, disabled: Initialize function running

Autonomous, enabled: Autonomous function running

Operator, enabled: Operator Control function running

```
#include "BuiltIns.h"
```

```
/*  
 * This function must be here for a competition project. It is  
 * automatically referenced by WPILib at startup and run. At that  
 * point the SetCompetitionMode function sets the competition  
 * mode. Basically, a mode of 0 is the default (without the  
 * IO_Initialization function) and runs a main function only.  
 * SetCompetitionMode(1) runs a competition project as shown.  
 */
```

```
void IO_Initialization(void)  
{  
    SetCompetitionMode(1);  
}
```

```
/*  
 * Initialize is run immediately when the robot is powered on  
 * regardless of the field mode.  
 */
```

```
void Initialize(void)  
{
```

```

    int i;
    for (i = 0; i < 10; i++)
    {
        printf("Initialize %d\r", i);
        Wait(500);
    }
}

/*
 * Autonomous is run as soon as the field controls enable the
 * robot. At the end of the autonomous period, the Autonomous
 * function will end (note: even if it is in an infinite loop
 * as in the example, it will be stopped).
 */
void Autonomous(void)
{
    while (1)
    {
        printf("In autonomous\r");
        Wait(500);
    }
}

/*
 * The OperatorControl function will be called when the field
 * switches to operator mode. If the field ever switches back
 * to autonomous, then OperatorControl will automatically exit
 * and the program will transfer control to the Autonomous
 * function.
 */
void OperatorControl(void)
{
    while (1)
    {
        printf("In OperatorControl\r");
        Wait(500);
    }
}

/*
 * the main program is not used, but needs to be here to
 * statisfy a reference to it. This requirement will probably
 * go away in the next version of WPILib.
 */
void main(void)
{
}

```