

FLAMINGICE™ SERIES

GETTING STARTED

© 2008 AIS Cube. All rights reserved.

The FlamingICE(FI) and FIDE are either registered trademarks or trademarks of AIS Cube in Singapore and/or other countries.

Microsoft, Windows, Visual Basic and Visual C# are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Adobe and Acrobat are trademarks of Adobe Systems Incorporated.

FOREWORD

The “FlamingICE™ Series - Getting Started” aims to serve as a walkthrough from writing your first program in FIDE and downloading it to the FI Board, to controlling the E-Modules available for integration with the FI. These mini examples should provide enough resources for you to get a jump start in completing your projects.

NOTE: This walkthrough assumes that you have no previous knowledge or experience in using Microcontrollers or our FI Series what so ever. Should you require only specific examples, feel free to jump and skip sections.

SETTING UP YOUR WORKSTATION

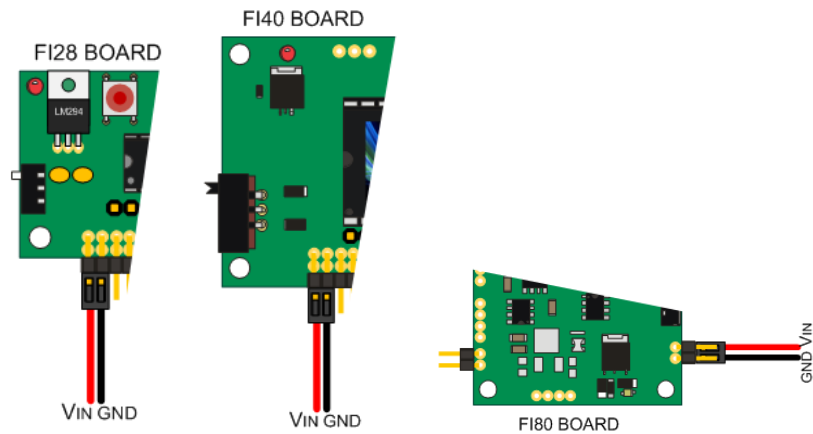
To perform the most basic of functions, you would require:

HARDWARE:

- Either an FI80/FI40/FI28 Board
- A 6V-12V DC power supply
- FTDI USB Comm.Key

SOFTWARE:

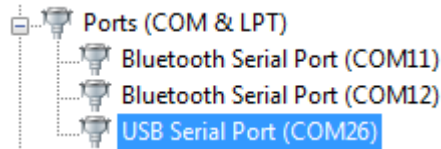
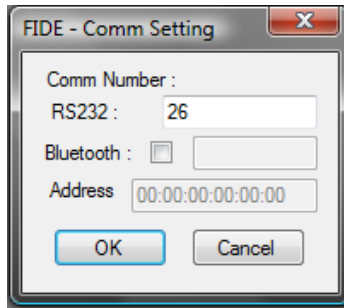
- FIDE
- CommKey Driver
- .NET Platform



1. Power up your FI Board according to the diagrams above for your specific board model. The power switch is located next to the power source.
2. Connect your FI Download Port to any USB Port on your PC. The download port is the JST connector located at the outermost edge of the board for all 3 FI boards. Please refer to the FICE Hardware Documentation for more details.
NOTE: Please install the Comm.Key driver BEFORE connecting the Comm.Key to the PC. If you have not already done so, please install the latest version of FIDE, you may need to install the .NET Platform if your PC does not already have it in the system. The driver and software can be obtained from our website.
3. Launch FIDE.


LAUNCHING FIDE

Upon launching the FIDE (FlamingICE Integrated Development Environment), this dialog should pop up requesting the Comm Number. This comm. Number can be obtained from your Device Manager when you have plugged in you Comm.Key.



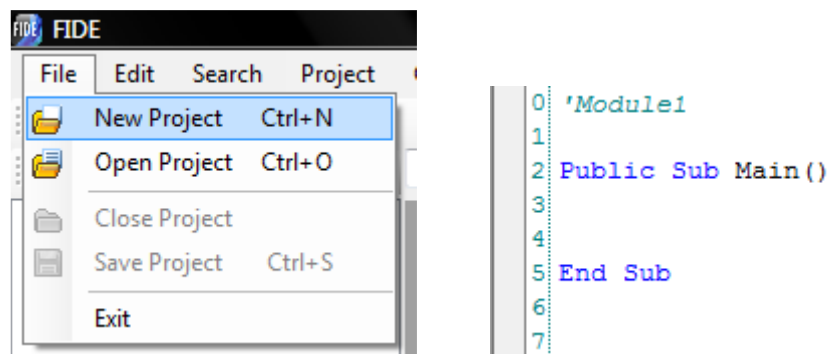
You can get the port numbers through:
Control Panel > System > Device Manager > Ports

In this case, the port number is COM26.

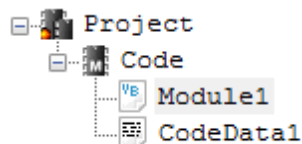
Press *OK* when you're done and the port should automatically open if the port number is valid. There are indicators on the bottom tool-strip stating "COM26 Open" and the icon just above the debug window should look like this .

CREATING A NEW PROJECT

To create a new project, go to **File > New Project**. Keyboard Short-Cut: **CTRL + N**.




A new project will be automatically created for you, along with the standard structures of a program in the editor.



A project consists of your code, which is contained in Modules. You will be writing your program in Modules. The Module which contains the subroutine 'Main', is your Main Module. Every Program MUST have a subroutine called Main(), and each subroutine has to have a unique name.

TIP: You can rename your modules to whatever you want to call it. This can be done under: **Project > Rename**.

SAVING YOUR PROJECT

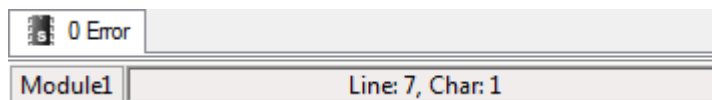
Before continuing, let's save this project. This can be done under **File > Save Project** or clicking on the  Save Icon on the toolbar. The Save Dialog should appear. Type in the project name that you would like to save as, e.g. "My Project" and hit the save button when you are done.

Keyboard Short-Cut: **CTRL + S**.


COMPILING YOUR PROJECT

You can now compile your project by selecting **Project > Compile** or clicking on the  Compile Icon on the toolbar. Keyboard Short-Cut: **F4**.

Compiling the project checks for any syntax errors in your program, you should not encounter any errors here. You can tell the number of errors you currently have, from the bottom indication in FIDE. For now, you should see something as follow.



COMPILING AND DOWNLOADING YOUR PROJECT

Since there are no errors in our project, let's download it into the FI. From the FIDE menu choose: **Project > Compile And Download** or press the  icon on the tool bar.

Keyboard Short-Cut: **F5**.

You may have to wait for a moment for the download process to complete. Your download is successful if there are no errors reported.

If you get an error message, such as: "**Download Error**"

Check the following:

- Make sure your power supply is on. The power indicator on your FI board should light up.
- Make sure that the Comm.Key is connected properly to the FI board and to your PC.
- Make sure you have the right Comm Port setting.

HELLO WORLD PROGRAM

Congratulations, you have successfully created, compiled and downloaded your first FI Project. Now for more VB coding.

The language you will be using is VB (Visual Basic); there are a number of differences as you will see when you work more with FIDE. One of which is that the language is streamlined to fit the .NET platform and style instead of the conventional VB language.

In the FIDE Editor, add in


```
0 'Module1
1
2 Public Sub Main()
3
4 Debug.Print "HELLO WORLD"
5
6
7 End Sub
8
9
```

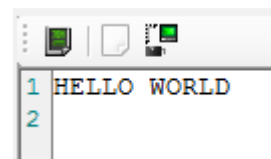
Debug.Print "HELLO WORLD"

Note : It is best to leave your Caps Lock on on your PC. The Editor will automatically format your text as you type.

Debug.Print is a FI command that sends the message between the quotes back to the PC. In this case it sends "HELLO WORLD" to the PC.

Now Download and Compile your project.

(Reminder: you can Download and Compile your project by pressing F5 or clicking the  icon.)



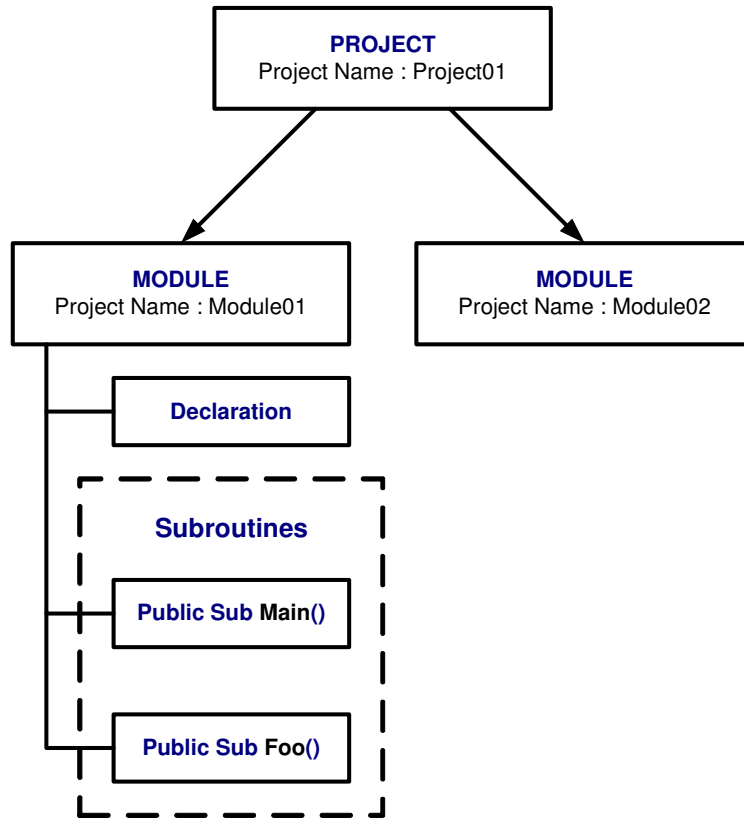
The screenshot shows a small window with a toolbar at the top containing icons for file operations and a gear icon. Below the toolbar, the text "HELLO WORLD" is displayed on the first line, and the second line is empty.

```
1 HELLO WORLD
2
```

After Compiling and Downloading your Project, you should see the message "HELLO WORLD" appearing at the debug window. The debug window is located to the right of your editor.

THE PROGRAM (HELLO WORLD)

A Project consists of one or more modules. Each Module consists of a declaration and one or more Subroutine. Every Project must have the Main() subroutine.



Between the Public Sub Main() and the End Sub is where you write your program.

In our Project (Project01), we have 1 Module (Module01) and 1 Subroutine call Main(). There are no declarations in our Project. The first line 'Module01 is a commented line which means that the compiler does not execute the line. A commented line is in green.

```
0 'Module1
1
2 Public Sub Main()
3
4 Debug.Print "HELLO WORLD"
5
6 End Sub
7
8 |
```

Commented Line →

Line Numbers →

Subroutine Main() →

FI Command ←

BLINKING LED PROGRAM

Here's another simple program. If you have an LED module, You can turn on and turn off the LED by using the following command High and Low.

HIGH

Set logic 1 at the specified pin.

HIGH [pin no.]

Example:

'Assuming an LED is attached to the specified pin, this code will turn 'the LED OFF.

HIGH 3

LOW

Set logic 0 at the specified pin.

LOW [pin no.]

Example:

'Assuming an LED is attached to the specified pin, this code will turn 'the LED ON.

LOW 3

NOTE: I/O pins are tied low to the ground on the hardware therefore explaining the inverse result as opposed to the logic statements.

Write the following program and Compile and Download it.

```
0 'Module1
1
2 Public Sub Main()
3
4 Debug.Print "HELLO WORLD"
5
6 Low(x)
7 Delay(200)
8 High(x)
9
10 End Sub
11
```

Replace 'x' with the pin number assigned to the IO pins on the FI Board to which you connected the LED. You can obtain these numbers by referring to the FICE Hardware reference for your respective FI Board.

Hint : You can restart the program by pressing the reset button on the FI board. After downloading the program, you will find that the LED blinks once for 200 msec.

THE PROGRAM (BLINKING LED)

```
0 'Module1
1
2 Public Sub Main()
3
4 Debug.Print "HELLO WORLD"
5
6 Low(x) 'TURN ON LED
7 Delay(200) 'WAIT 200MS
8 High(x) 'TURN OFF LED
9
10 End Sub
11
12
```

The Program starts by sending “HELLO WORLD” to the PC. Then the LED will be turned on. The program waits for 200 msec (or 0.2 sec) before turning the green LED off.

BASIC STRUCTURE : DO LOOP

In the last program we will need to type 3 times to blink the green LED 3 times. Or maybe we can just cut and paste the 3 lines. What do we need for the FI to instruct the LED to blink indefinitely?

What we need is to implement is a structure called Do...Loop.

Type in the following program and read the explanation before downloading it.

```
0 'Module1
1
2 Public Sub Main()
3
4 Debug.Print "HELLO WORLD"
5
6 Do
7     Low(x) 'TURN ON LED
8     Delay(200) 'WAIT 200MS
9     High(x) 'TURN OFF LED
10    Delay(200) 'WAIT 200MS
11 Loop
12
13 End Sub
14
15
```

The Do Loop repeats all commands between the Do and the Loop.