

## **Using ROBOTC for FRC Competitions – Table of Contents**

1. Update FRC and ROBOTC to Competition-ready versions
  - a. Install or Upgrade ROBOTC version 1.13
  - b. Update the FRC Master Code (Master Firmware) to Version 15d or higher
  - c. Update the FRC User Code (Firmware) to Version 7.24 or higher
2. Before Programming
  - a. Competition Programming vs. Normal Programming?
3. Begin Programming!
  - a. “FRC\_Comp\_Template.c” Competition Template (sample code folder)
  - b. About Competition Programming
4. Example Competition Program

Note: The Competition port on the Operator Interface is required to activate the “Disabled” and “Autonomous” modes in the Competition template.

## **Updating FRC and ROBOTC to Competition-ready versions**

### **Upgrading ROBOTC to Version 1.13**

*ROBOTC for IFI* version 1.13 is a free upgrade for all existing *ROBOTC for IFI* users. You can download the latest version of ROBOTC from <http://www.robotc.net/ifi/>. To install the updated version:

1. Download the latest version of *ROBOTC for IFI* from <http://www.robotc.net/ifi/>
2. Uninstall your previous version of *ROBOTC for IFI*
3. Install the new version of *ROBOTC for IFI*

### **How to update FRC Master Firmware in ROBOTC**

*ROBOTC for IFI* version 1.13 has the Innovation FIRST Firmware Loader capabilities built-in. This allows you to update both the master and user firmware without having to leave ROBOTC. To update the FRC Master Firmware:

1. Open *ROBOTC for IFI*. Ensure that your Robot Controller is connected to the computer via serial.
2. Connect the Operator Interface to the Robot Controller together via a tether (red cable) connection.
3. Press and hold the “PROG” button on the Robot Controller until the “Program State” light is orange. This will enter the Robot Controller into programming mode.
4. Return to ROBOTC. Open the “Robot” menu and choose “Platform Type”, then select “Innovation First (IFI) FRC” as the platform type.
5. Open the “Window” menu and choose “Menu Level”. Select “Expert”
6. Open the “Robot” menu and select “Download IFI Master CPU Firmware”
7. An Open File Dialog box will open. Select the “FRC\_MASTER\_V15d.BIN” (or higher, if available)
8. The Master Firmware will begin to download. Once this is completed, Power Cycle the Robot Controller. You will now need to download the user firmware before doing anything else.

## How to update FRC User Firmware in ROBOTC

*ROBOTC for IFI* version 1.13 has the Innovation FIRST Firmware Loader capabilities built-in. This allows you to update both the master and user firmware without having to leave ROBOTC. To update the FRC User Firmware:

1. Open *ROBOTC for IFI*. Ensure that your Robot Controller is connected to the computer via serial.
2. Press and hold the “PROG” button on the Robot Controller until the “Program State” light is orange. This will enter the Robot Controller into programming mode.
3. Return to ROBOTC. Open the “Robot” menu and choose “Platform Type”, then select “Innovation First (IFI) FRC” as the platform type.
4. Open the “Robot” menu and select “Download Firmware”
5. An Open File Dialog box will open. Select the “FRC VM 0724.hex” (or higher, if available)
6. The Firmware will begin to download. Once this is completed, Power Cycle the Robot Controller. You can now use ROBOTC for IFI with the FRC platform.

## Competition Programming vs. Normal Programming?

In normal ROBOTC programming, every program is based around a “task main” function, as shown below:

```
task main()  
{  
  
}
```

**In a competition program, things are a little different.** To keep things simple, however, a Competition Template program is built into ROBOTC for use in FRC Competitions. The template provides a common starting point for teams when competing in an event. Rather than a single “main” task, the template has three sections, each matched to a specific section of the competition:

- void Initialization() – For running code that the robot uses to “set up” before the competition begins
- task Autonomous() – For code the robot will run during the autonomous period.
- task Human\_Control() – For code the robot will run during the user control period

The template and its accompanying “include” file handle the background work for the competition. All you need to do is tell the robot what to do in each phase.

## **To Open a Competition Template:**

1. Go to File, and select Open Sample Program.
2. Select the “FRC\_Comp\_Template.c”, and select Open.
  - Note: The competition template is a read-only file, so be sure to save a copy of the file somewhere else on your computer.

## **Programming and Testing Competition Code:**

Note: With competition control enabled, your robot will not begin to execute code until after it receives valid transmitter data.

### **Pre-Competition**

```
void Initialization()  
{  
  
}  
}
```

Place your initialization code inside this function.

During the pre-competition section, code can be executed to configure your robot before the competition begins. Valid code for this section includes tasks such as configuring sensors, setting up reflected motors and setting initial servo positions.

This code will execute only once and will run before the competition begins. There may be a 2 second delay from the point that you power your robot on and the code executes.

### **Autonomous**

```
task Autonomous()  
{  
  
}  
}
```

Place your Autonomous Period code inside this function.

During the autonomous period, the robot will execute the code inside of the Autonomous task while the autonomous period of the competition is active. During this time, your program will run, but no commands from the Operator Interface will be sent to the Robot Controller (i.e. remote control signals).

You can simulate Autonomous mode by connecting a switch to pins 5 and 8 on the Operator Interface's Competition Port. See this diagram for more information - <http://www.ifirobotics.com/docs/competition-port-pinout-guide-reva.pdf>

## Operator Controlled

```
task Human_Control( )  
{  
  
}
```

Place your User Controlled Period code inside this function.

During the operator controlled period, the robot can accept commands from the Operator Interface. This segment of code executes immediately after the autonomous period ends.

The Human\_Control task will only execute once, so it is advisable to have the code in a loop (such as a *while(true)*) to ensure that any commands that need to run continuously will not stop until the end of the match.

### **Testing your code:**

To test your competition code without a field control kit, follow these steps:

1. Follow the Instructions on the IFI website to connect the switch for both “Autonomous” and “Disabled” to the Operator Interface.
2. Load the Master and User firmware onto the Robot Controller.
3. Write a program and download the program to the Robot Controller (note: the Operator Interface must be connected via tether or radio to download a program to the Robot Controller)
4. After the program is downloaded, reset the Robot Controller (By the “Robot Reset” or simply power cycling the robot). Once reset, the program will start instantly into the human control phase. You can begin testing the Human Control phase instantly.
5. To test the autonomous phase, close the switch you put on the Operator Interface. The Robot will detect it is in autonomous mode and instantly begin executing the Autonomous code. The Autonomous code will execute until the Disabled switch is closed, or the Autonomous switch is opened.

## Code Example:

This example shows all three sections of the competition code being used. Here's an explanation of everything is happening.

Include Statement – Default include file. This line is required for all competition programs written with ROBOTC.

Initialization() – This code will execute before the competition even begins. You should avoid having motor commands, and only use this area to setup functionality for use inside of the competition. This program's initialization function has two statements inside of it:

1. `bMotorReflected[port2] = true` – This will cause all motor commands to port 2 to be inverted. This allows you to send the same motor commands as your other motor, even those they are mounted opposing one another.
2. `frDigitalIODirection[pio1] = dirInput;` - This command will set the Digital I/O port #1 as an input port. We will be using a touch sensor on this port, so it needs to be setup as a digital input.

Autonomous – The code will execute as soon as the autonomous period has begun. This portion of the code will not respond to joystick controls or values.

In this program the robot to move forward until a touch sensor attached to sensor port 1 is pressed. After the sensor is pressed, the robot will stop until the end of autonomous.

Human Control – Once the Autonomous period is over, the Human Control period starts. The Human Control code will execute forever until the robot loses a radio signal or is disabled by the competition port.

In this program the robot will be controlled by the y-axis on both the port 1 and port 2 joysticks. This gives basic steering functionality.

```

#include "FRC_Comp_Include.c"

void Initialization()
{
    bMotorReflected[port2] = true;
    frcDigitalIODirection[piol] = dirInput;
}

task Autonomous()
{
    while(frcDigitalIOValue[piol] == 0)
    {
        motor[port1] = 30;
        motor[port2] = 30;
    }
    motor[port1] = 0;
    motor[port2] = 0;
}

task Human_Control()
{
    while(true)
    {
        motor[port1] = frcRF[p1_y];
        motor[port2] = frcRF[p2_y];
    }
}

```