

## Programming Guide (ROBOTC®)

### Introduction

In this guide, code created with the FTC® templates will be explored and then run using the Field Control Software (FCS). An FTC game begins with an autonomous period where only autonomous code can be run on the robot. A TeleOp phase follows, where teams can run a program that allows them to operate their robot with a joystick for the remainder of the game. To simulate a FTC game, an autonomous program and a TeleOp program will be deployed onto the NXT brick and run with the FCS. This guide is for use with the ROBOTC® programming language.

### Review:

- To review how to install the FCS, review the documentation in the Samantha Module extension.

### Importance of FCS and Programming Templates:

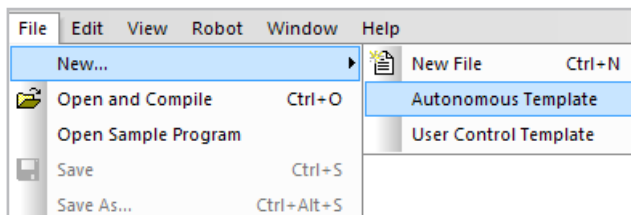
The Samantha Field Control Software is used to set up matches. It keeps track of time and is also responsible for relaying commands from joysticks to the Samantha Modules.

The Programming Templates are necessary to pass the software inspection. These templates contain specific commands, without which the robot will not be able to correctly respond to the commands from the FCS.

### Autonomous Template:

#### Access and Understand the Autonomous Template:

1. To open the autonomous template, click **File** and then navigate to **New...** and choose **Autonomous Template** from the list provided.



2. Save the file with a new name into an appropriate directory.
3. The autonomous template is very well commented and easy to understand. At the beginning of the template there are ROBOTC compiler-specific commands that can be edited using the **Motors and Sensors Setup** option, found in the **Robot** menu of ROBOTC.

```
#pragma config(Hubs, S1, HTMotor, HTServo, none, none)
#pragma config(Motor, mtr_S1_C1_1, , tmotorNone, openLoop)
#pragma config(Motor, mtr_S1_C1_2, , tmotorNone, openLoop)
/*!!Code automatically generated by 'ROBOTC' configuration wizard !!*/

/////////////////////////////////////////////////////////////////
//                               Autonomous Mode Code Template
//
// This file contains a template for simplified creation of an autonomous
// program for a TETRIX® robot competition.
//
// You need to customize two functions with code unique to your specific robot.
//
/////////////////////////////////////////////////////////////////
```

## Programming Guide (ROBOTC®)

4. The next line is the command to include the **JoystickDriver.c** file, which allows the robot to interact with the FCS software and receive joystick commands. The FCS will be discussed later in this guide.

```
#include "JoystickDriver.c"    //Include file to "handle" the Bluetooth™ messages.
```

5. The inclusion of the **JoystickDriver.c** is followed by the **initializeRobot()** function, where the user can add commands that are responsible for the initialization of the robot. The commands within this function will direct the robot to attain the initial position, so that it is ready before the match begins.

```
////////////////////////////////////
//
//          initializeRobot
//
// Prior to the start of autonomous mode, you may want to perform some
// initialization on your robot.
// Things that might be performed during initialization include:
// 1. Move motors and servos to a preset position.
// 2. Some sensor types take a short while to reach stable values during which time it is best
//    that robot is not moving. For example, gyro sensor needs a few seconds to obtain
//    the background "bias" value.
//
// In many cases, you may not have to add any code to this function and it will remain "empty".
//
////////////////////////////////////

void initializeRobot()
{
    // Place code here to initialize servos to starting positions.
    // Sensors are automatically configured and setup by ROBOTC®. They may need a brief
    // time to stabilize.

    return;
}
```

6. The command to follow the **initializeRobot()** function is the beginning of the **main()** function. The very first command within the main function is a call to the **initializeRobot()** function.

```
task main()
{
    initializeRobot();
}
```

**Note:** When the **initializeRobot()** command is processed, the **main()** function calls the **initializeRobot()** function and the commands within this function are executed. Once the commands within the function have been executed, the processor returns to the **main()** function to execute the commands that follow the **initializeRobot()** command.

## Programming Guide (ROBOTC®)

- The initializeRobot() function call is followed by the **waitForStart()** function. The waitForStart() function refrains the processor from executing the rest of the code until it receives a signal from the FCS to start executing the next command.

```
waitForStart(); // Wait for the beginning of autonomous phase.
```

```

////////////////////////////////////
////////////////////////////////////
////                               ////
////   Add your robot specific autonomous code here.   ////
////                               ////
////////////////////////////////////
////////////////////////////////////

```

- The only statements left in the autonomous template after the waitForStart() function are an empty infinite loop and the closing braces. This is where the commands that the robot must execute during the autonomous mode are placed.

```

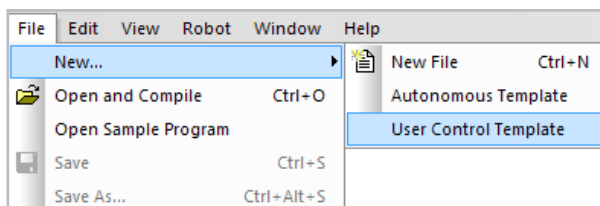
while (true)
{
}

```

### TeleOp Template:

#### Access and Understand the TeleOp Template:

- To open the TeleOp template, click **File** and then navigate to **New...** and click the **User Control Template** option.



- Save the file with a new name into an appropriate directory.
- The TeleOp template contains the same commands and functions as the autonomous template, all of which have been discussed in the Autonomous Template section of this programming guide.

**Note:** The JoystickDriver.c file is extremely important for the TeleOp mode, since this file is responsible for handling the commands sent by the user via the joystick. The robot then acts according to the commands from the joystick and the code in the TeleOp program.

### Samantha Field Control System

The Field Control System (FCS) is the software provided by FTC® to setup matches, to keep a record of time, as well as to send messages from the joystick to the robot.

It is helpful to note that the Samantha Field Control System Application window generally changes each season, depending on the format of the game for that season, as well as any new technologies that have been introduced.

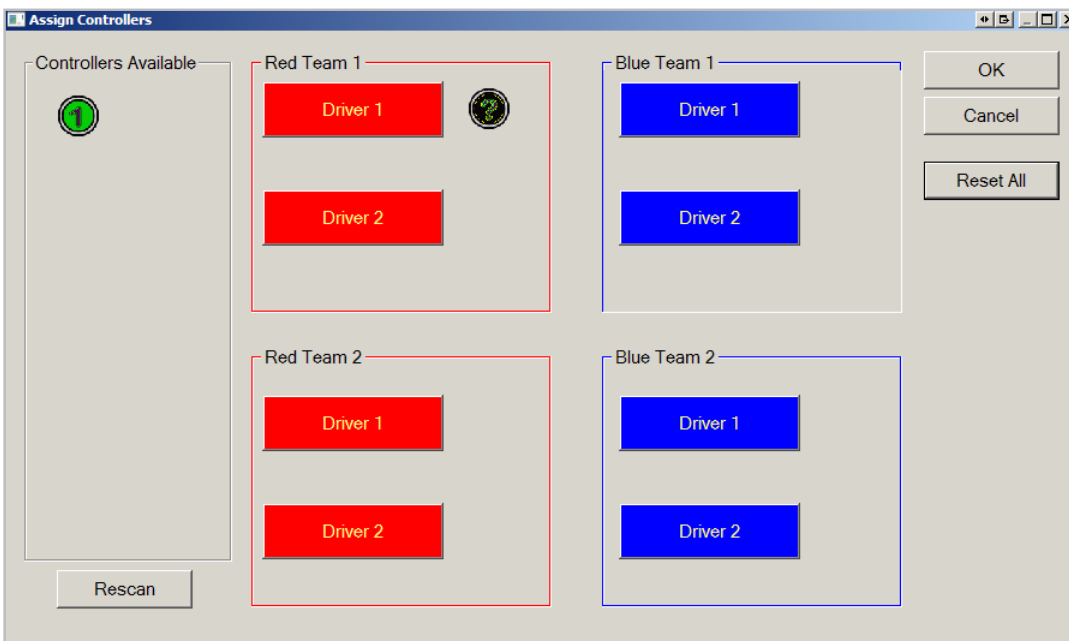
Teams participating in the FIRST® Tech Challenge must have the FCS installed and running on their computers to pass the software inspection.

## Programming Guide (ROBOTC®)

### Connect to the FCS:

1. Before connecting to the FCS, it is necessary to change the name of the NXT Brick on the robot. This can be done in either LabVIEW™ or ROBOTC®. Any name besides 'NXT' will suffice.
2. Open the Samantha Field Control System software.
3. The first window that appears will be the **Assign Controllers** window. Each joystick connected to the computer will appear in the **Controllers Available** box on the left. To select a joystick, push any button when the question mark on the screen is flashing next to the correct team and driver. To change the team and driver, simply click one of the other Driver buttons. If a controller has been incorrectly assigned, click the **Driver 1** or **Driver 2** button next to that particular controller to undo the assignment. Click the **Reset All** button to undo all assignments, and then click the **OK** button.

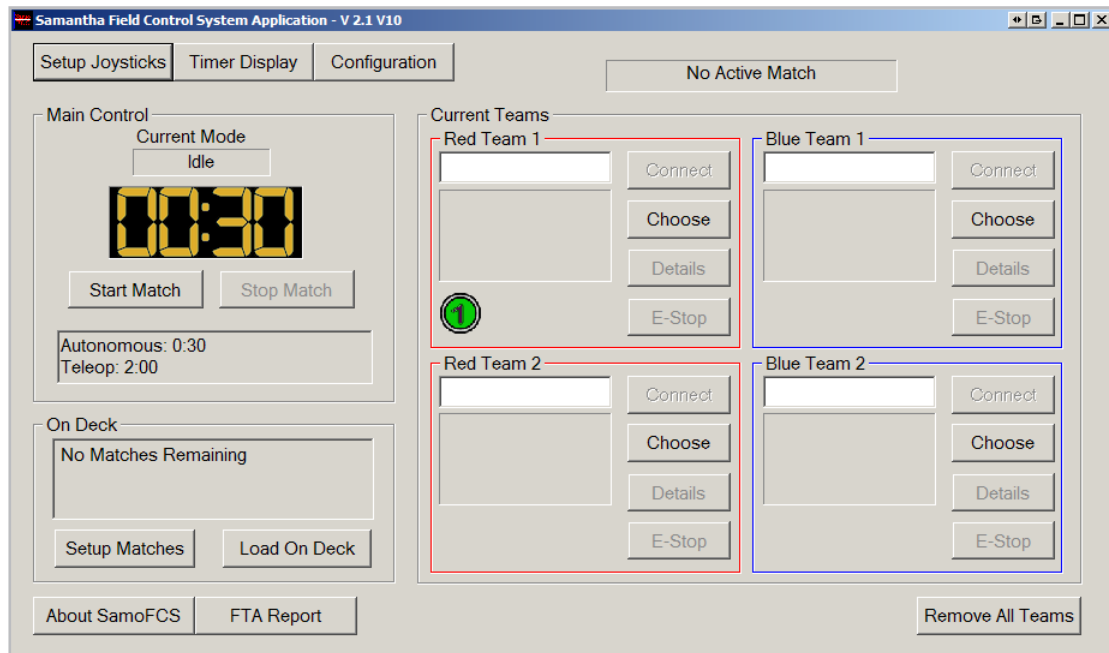
**Note:** If a new joystick has been connected to the computer and does not appear in Controllers Available panel, click the **Rescan** button.



4. The Samantha Field Control System Application window will appear. Notice that the joystick numbers appear under the selected team names in the Current Teams box on the right side of the window.

**Note:** The **Setup Joysticks** button can be used to return to the Assign Controllers window.

## Programming Guide (ROBOTC®)

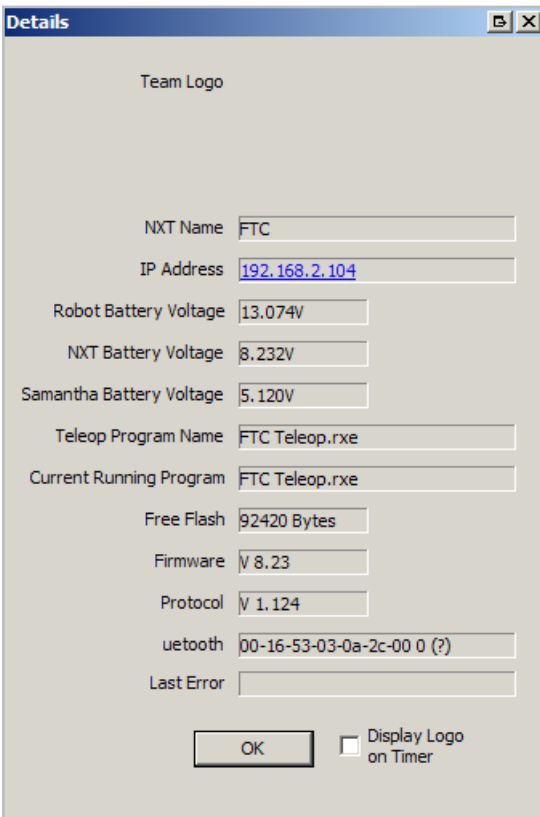


5. Ensure that the Samantha Module and NXT Brick are both powered on, and that the NXT is connected to the USB port on the Samantha Module. Select the **Choose** button in any of the Current Team boxes where a joystick has been assigned.
6. From the window that appears, select the correct NXT name and then press the **OK** button. The text box below the team name will contain the NXT name. The space below the NXT name will be yellow and will display information about the robot. The NXT selected is now connected to the FCS.



## Programming Guide (ROBOTC®)

7. Click the **Details** button to view relevant information about the Samantha Module in the Details window.



The 'Details' window displays the following information:

| Team Logo                |                               |
|--------------------------|-------------------------------|
| NXT Name                 | FTC                           |
| IP Address               | <a href="#">192.168.2.104</a> |
| Robot Battery Voltage    | 13.074V                       |
| NXT Battery Voltage      | 8.232V                        |
| Samantha Battery Voltage | 5.120V                        |
| Teleop Program Name      | FTC Teleop.rxe                |
| Current Running Program  | FTC Teleop.rxe                |
| Free Flash               | 92420 Bytes                   |
| Firmware                 | V 8.23                        |
| Protocol                 | V 1.124                       |
| uetooth                  | 00-16-53-03-0a-2c-00 0 (?)    |
| Last Error               |                               |

At the bottom, there is an 'OK' button and a checkbox labeled 'Display Logo on Timer' which is currently unchecked.

8. For additional details about the Wi-Fi network that Samantha is using, click the **IP Address** link. After clicking the IP Address link, detailed information will appear in the default web browser window and can be studied as needed. Notice that the list of files stored in the NXT will not be available while the Samantha module is connected to the FCS and can be obtained by refreshing the page after Samantha Module has been disconnected.

## Programming Guide (ROBOTC®)



### Samantha Information

VERSION: V1.49/Nov 2 2010

|            | SSID      | AUTHENTICATION |
|------------|-----------|----------------|
| PRIMARY:   | FTC_FIELD | [PSK]          |
| SECONDARY: | FTC_FIELD | [PSK]          |
| OTHER:     |           |                |

|                                 |                   |
|---------------------------------|-------------------|
| HOSTNAME:                       | NXTFTC            |
| DHCP CHECKED:                   | Enabled           |
| IP ADDRESS:                     | 192.168.2.104     |
| GATEWAY:                        | 192.168.2.1       |
| SUBNET MASK:                    | 255.255.255.0     |
| DNS1:                           | 192.168.2.1       |
| MAC ADDRESS:                    | 00:1E:Co:01:0A:53 |
| ACCEPTING FCS CONNECTIONS FROM: | 192.168.2.186     |
| FCS CLIENT IP ADDRESS:          | 192.168.2.186     |
| BRICK STATUS:                   | Connected         |
| BATTERY VOLTAGE:                | 13.033V           |
| USB VOLTAGE:                    | 5.118V            |
| LAST ERROR:                     |                   |

| Total Message Stats |       |       |
|---------------------|-------|-------|
|                     | COUNT | BYTES |
| RECEIVED            | 480   | 984   |
| SENT                | 479   | 6708  |

| Joystick Message Stats |   |
|------------------------|---|
| WAIT AUTONOMOUS:       | 0 |
| AUTONOMOUS:            | 0 |
| WAIT TELEOP:           | 0 |
| TELEOP:                | 0 |

### Lego Mindstorm Brick Information

|                |             |
|----------------|-------------|
| BATTERY LEVEL: | 8.183 volts |
| BLUETOOTH:     | On          |

File list is unavailable while FCS is connected

### Lego Mindstorm Brick Information

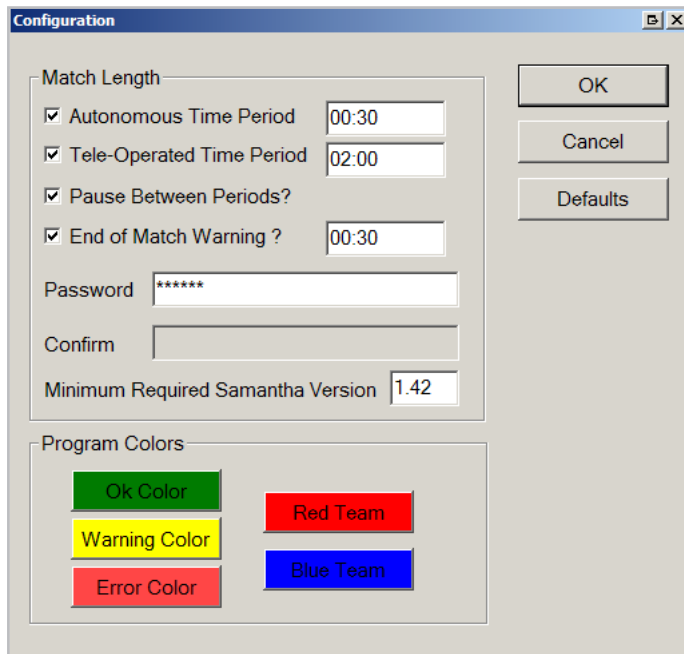
|                |             |
|----------------|-------------|
| BATTERY LEVEL: | 8.172 volts |
| BLUETOOTH:     | On          |

| #  | FILE NAME           | FILE SIZE |
|----|---------------------|-----------|
| 1  | NVConfig.sys        | 56        |
| 2  | FTCConfig.txt       | 14        |
| 3  | Advanced (i).rxe    | 1961      |
| 4  | FTC Teleop.rxe      | 1581      |
| 5  | FTC Autonomous.rxe  | 1542      |
| 6  | Autonomous.rxe      | 800       |
| 7  | Program Chooser.rtm | 1044      |
| 8  | Woops.rso           | 4699      |
| 9  | !Startup.rso        | 8161      |
| 10 | !Click.rso          | 451       |
| 11 | !Attention.rso      | 1755      |
| 12 | faceopen.ric        | 316       |
| 13 | faceclosed.ric      | 316       |
| 14 | Touch.rtm           | 576       |
| 15 | Sound.rtm           | 560       |
| 16 | SONAR.rtm           | 1507      |
| 17 | Light.rtm           | 522       |

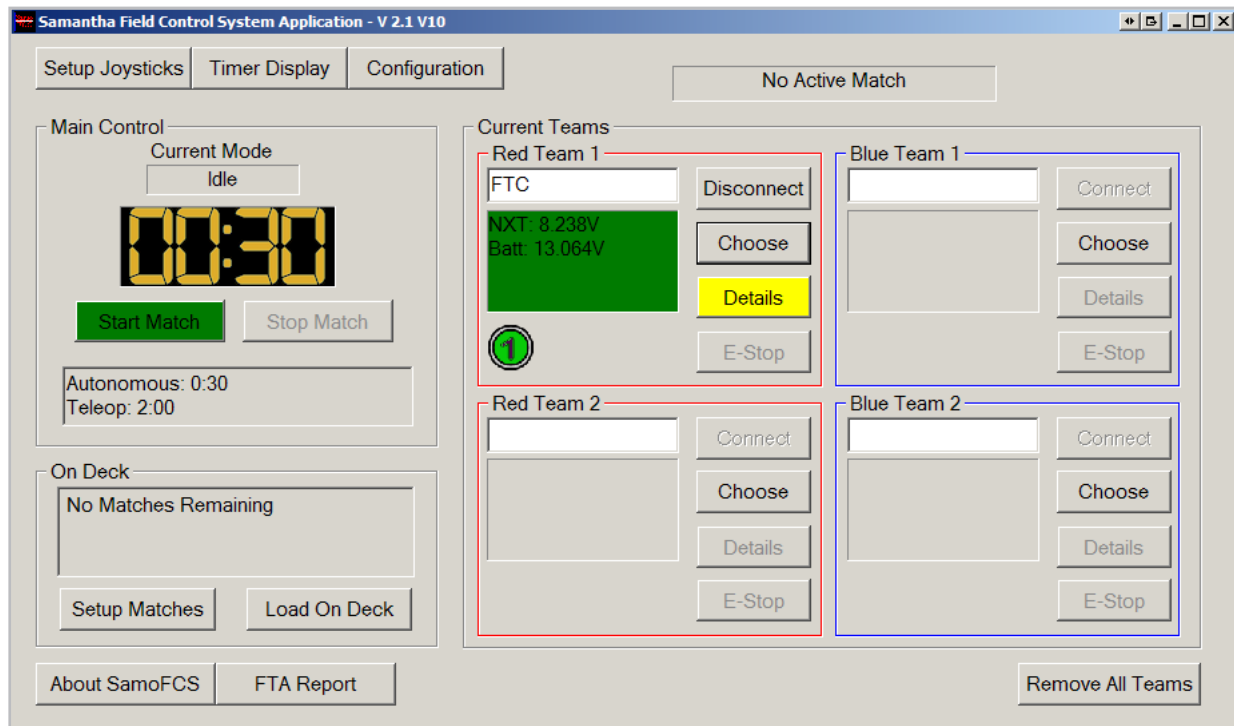
## Programming Guide (ROBOTC®)

### Set Up and Execute a Match:

1. From the top of the Samantha Field Control System Application main window, click the **Configuration** button.
2. In the window that appears, under the **Match Length** heading, it is possible to select or deselect different options and change the length of each portion of the match. The colors selected to represent Warnings and Errors can also be changed. When the desired match settings have been set, click the **OK** button. To return to the default match configuration, click the **Defaults** button at any time.



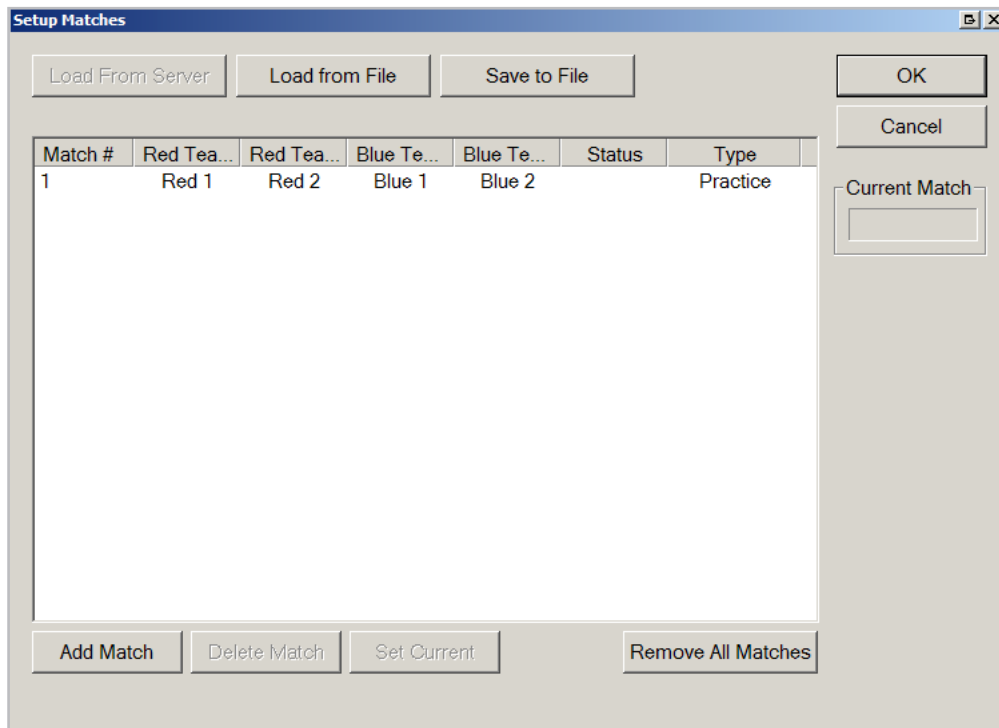
3. In the **Samantha Field Control System Application** window, the **Main Control** box on the right contains all the information and buttons needed to execute and control a match. Before beginning a match, ensure that each team is connected to a robot, and each has selected a program to run. The box in each team's window will turn green when they have successfully selected their program to run.



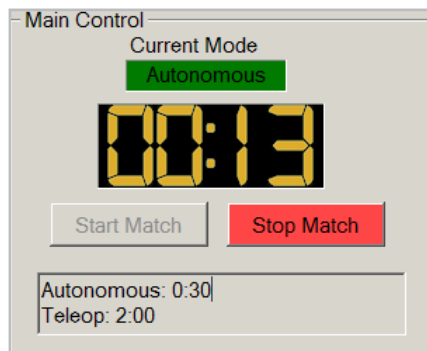


## Programming Guide (ROBOTC®)

4. A useful feature in the FCS is the **Setup Matches** window, which can be accessed using the **Setup Matches** button. This window allows the user to load matches from a file, save them to a file, and change team information by clicking the field and then typing in the required values or information. Click **OK** after setting up the matches to exit the window.



5. Press the **Start Match** button to begin the match. A sound will play to signal the beginning of the match, and each team's autonomous program will begin. This program will run on each NXT until the autonomous portion of the match is complete. The TeleOp portion of the match will follow, where each team will control their robot with the connected joystick. Depending on the specific match configuration settings, it may be necessary to select the **Continue** button to continue on to the TeleOp portion. The current mode of the match is displayed just above the timer.

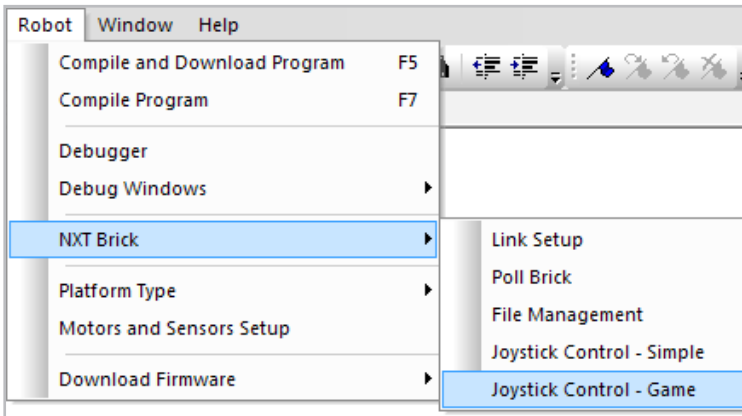


6. To display the match timer at any point, click the **Timer Display** button at the top of the window.
7. To stop the match before the time has run out, click the **Stop Match** button.

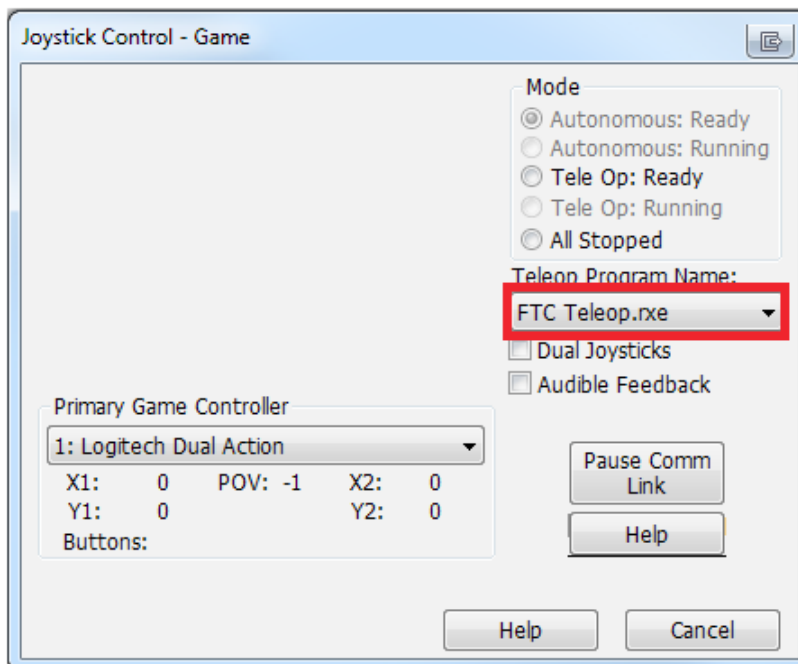
## Programming Guide (ROBOTC®)

### Select a TeleOp Program:

1. To select a TeleOp program, connect the NXT Brick and the Joystick to the computer.
2. From the **Robot** menu at the top of the window, navigate to **NXT Brick**, and from that menu, select **Joystick Control - Game**.



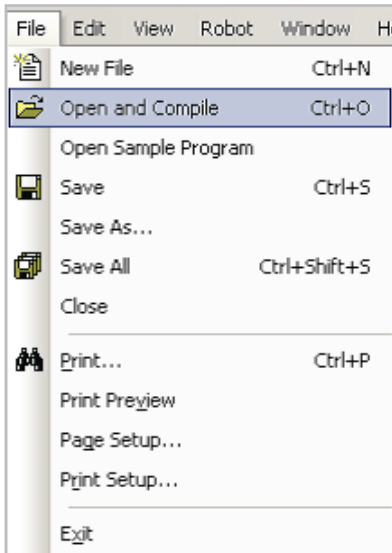
3. In the Joystick Control – Game window that appears, click the down arrow under the Teleop Program Name. A drop down menu will appear, including a list of files that are currently loaded on the NXT. Click the program that needs to run in the TeleOp mode of the match.



## Programming Guide (ROBOTC®)

### Open and Compile a File:

1. Open and save the ROBOTC® files specific to FTC Programming Extension on the computer.
2. Click the **File** menu in ROBOTC and then click **Open and Compile**. Locate the **ROBOTC** file and click the **Open** button.
3. To transfer the file to the NXT Brick, connect the NXT Brick to the computer and navigate to the **Robot** menu. Click



### Compile and Download.

**Note:** Once the NXT Brick has been connected to the computer, press **F5** to transfer the file to the NXT Brick.

