

VEX Competitions

Using ROBOTC for VEX Competitions

Table of Contents

1. Update VEX and ROBOTC to Competition-ready versions
 - a. Install or Upgrade ROBOTC version 1.08
 - b. Update the VEX Master Code (Master Firmware) to Version 7 or higher
 - c. Update the VEX User Code (Firmware) to Version 7.21 or higher
2. Before Programming
 - a. Competition Programming vs. Normal Programming?
3. Begin Programming!
 - a. "VEX_Competition_Template.c" Competition Template (sample code folder)
 - b. About Competition Programming
4. Example Competition Program

Note: Find the latest version of this document and other resources at <http://www.robotc.net/ifi/comp>

Upgrading ROBOTC to Version 1.08

ROBOTC for IFI version 1.08 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 VEX Master Firmware in ROBOTC

ROBOTC for IFI version 1.08 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 VEX Master Firmware:

1. Open ROBOTC for IFI. Ensure that your VEX controller is connected and turned on.
2. Open the "Window" menu and choose "Menu Level". Select "Expert"
3. Open the "Robot" menu and select "Download IFI Master CPU Firmware"
4. An Open File Dialog box will open. Select the "VEX_MASTER_V7.bin" (or higher)
5. The Master Firmware will begin to download. Once this is completed, download the user firmware before doing anything else.

VEX Competitions

How to update VEX User Firmware in ROBOTC

ROBOTC for IFI version 1.08 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 VEX User Firmware:

1. Open ROBOTC for IFI. Ensure that your VEX controller is connected and turned on.
2. Open the "Robot" menu and select "Download Firmware"
3. An Open File Dialog box will open. Select the "VEX VM 0721.hex" (or higher)
4. The User Firmware will begin to download. Once this is completed, power cycle (turn off and then on) your VEX to get started using ROBOTC for IFI.

Competition Programming vs. Normal Programming?

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

```
19 void pre_auton()  
20 {  
21  
22 }
```

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 VEX 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 pre_auton** – For running code that the robot needs to get "set up" before the competition begins
- **task autonomous** – For code the robot should run during the autonomous period (20 seconds by default)
- **task user_control** – For code the robot should run during the user control period (120 seconds by default)

The template and its accompanying "include" file handle the background work for the competition, such as monitoring the duration of each phase during a match, and communicating with the Vex Field Controller. All you need to do is tell the robot what to do in each phase.

Note: No Jumper Pins are required. All competition control is software controlled.

VEX Competitions

To Open a Competition Template:

1. Go to File, and select Open Sample Program.
2. Select the "Vex_Competition_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.

Setting Up Competition Control and Timing

```
6 #pragma competitionControl(Competition)
7 #pragma autonomousDuration(20)
8 #pragma userControlDuration(120)
```

With ROBOTC, the competition control is completely user defined and controlled via software, so there are no jumpers required for testing. By changing these settings, the template can be adapted to work with any competition. The three lines above set the VEX controller into the "competition" mode and specify the timing for the match.

#pragma competitionControl(competition_mode) – Controls the Competition mode that the Vex robot will function in. There are four different competition_modes that you can pass to this statement:

- **OFF** – Standard setting. No competition controls.
- **Competition** – Competition mode. The VEX will respond to Field Controller commands and switch between Autonomous and User Control modes at the competition-specified times. Use this mode for competitions.
- **autonomousTest** – Autonomous Test mode. Executes only the autonomous portion of the code. For testing outside of the competition arena.
- **UserTest** – Executes only the User Control portion of the code. For testing outside of the competition arena.

#pragma autonomousDuration(time_in_seconds) - Defines the duration of the autonomous phase of a VEX competition. Poor performance may result if this setting does not match the Field Controller settings.

#pragma userControlDuration(time_in_seconds) - Defines the duration of the user control phase of a VEX competition. Poor performance may result if this setting does not match the Field Controller settings.

VEX Competitions

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

```
19 void pre_auton()  
20 {  
21  
22 }
```

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

```
25 task autonomous()  
26 {  
27  
28 }
```

Place your Autonomous Period code inside this function.

During the autonomous period, the robot performs actions autonomously for the length of time specified by the "autonomousDuration(time)" setting. While the robot cannot accept commands from the RF Transmitter during this time, it still requires that the RF Transmitter's signal be present, as a safety precaution. You cannot skip the autonomous period during a competition by shutting the RF Transmitter off. If you shut the RF Transmitter off during the autonomous period, the VEX's internal timers will pause, potentially causing your robot to enter the User Control period later than it should.

VEX Competitions

Operator Controlled

```
31 task user_control()  
32 {  
33  
34 }
```

Place your User Controlled Period code inside this function.

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

If you shut the RF Transmitter off during this period you will not increase the length of the Operator Controller period as the Field Control System determines and controls the length of the human player period.

Testing your code:

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

1. Download code to VEX controller
2. Disconnect the programming cable from the VEX controller
3. Turn both the VEX and the RF Transmitter off (if not already)
4. Place the VEX robot on a flat surface
5. Turn the VEX controller on
6. Wait 5 seconds
7. Turn the FM Transmitter on... your Autonomous code should begin, and your User Control code should follow after the correct amount of time has elapsed.

VEX Competitions

Code Example

```

Auto const tSensors touch1      = (tSensors) S1;
Auto /*!!CLICK to edit 'wizard' created sensor
1
2 #pragma platform(VEX)
3
4 #pragma competitionControl(Competition)
5 #pragma autonomousDuration(20)
6 #pragma userControlDuration(120)
7
8 #include "Vex_Competition_Includes.c"
9
10 void pre_auton()
11 {
12     bMotorReflected[port1] = true;
13 }
14
15 task autonomous()
16 {
17     while(SensorValue[touch1] == 0)
18     {
19         motor[port1] = 127;
20         motor[port2] = 127;
21     }
22     motor[port1] = 0;
23     motor[port2] = 0;
24 }
25
26 task usercontrol()
27 {
28     motor[port1] = vexRT[Ch2];
29     motor[port2] = vexRT[Ch3];
30 }

```

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

Pragma Statements –

Four statements:

- Setting the platform for this program to "VEX"
- Setting the competition type to "Competition"
- Setting the Autonomous Period Duration to 20 seconds
- Setting the User Control Duration to 120 seconds

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

Void Pre_Auton – Initialization code – Reflecting the Motor attached to motor port 1.

Task Autonomous – Moving 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.

Task UserControl – Ch2 and Ch3 from the RF Transmitter are directly mapped to the two motors on the robot.