

# Your FIRST LabVIEW Program

In all of the languages there is a basic framework to start from that does all the background tasks you need to get a complete running system.

While you can write your project from scratch, there is some basic communication to the Driver Station that has to be setup first. And there are field states (Disable, Autonomous, Teleop) that your code has to be capable of responding to.

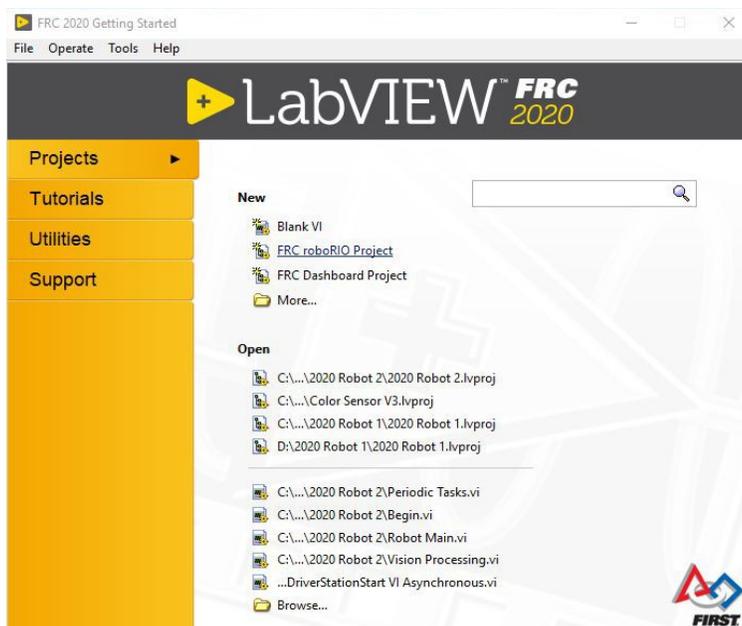
That background framework is handed to you, so that you don't have to understand everything all at once.

It also organizes your code structure in a common way making it easier for all of us to help each other.

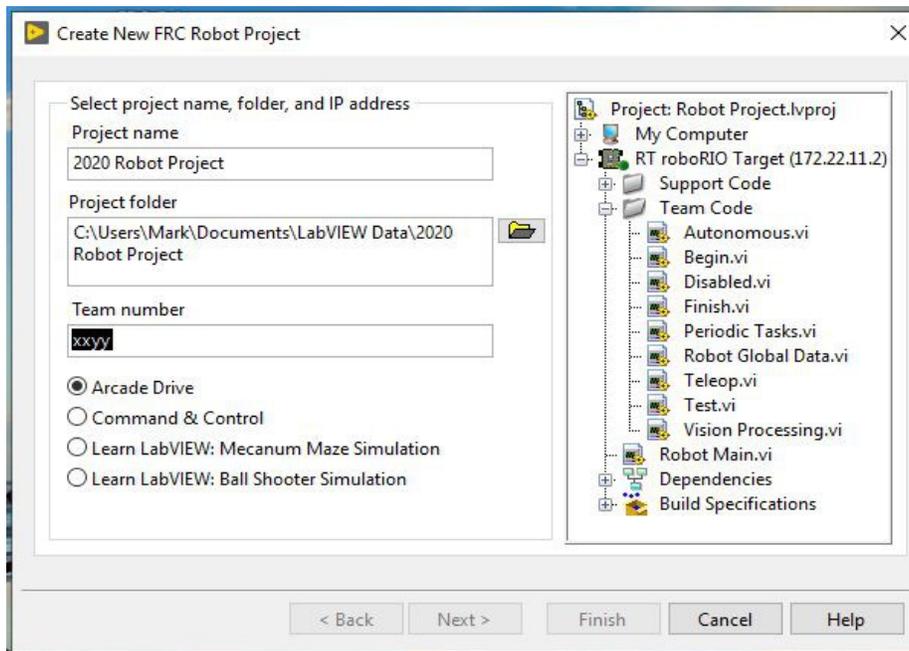
---

To start let's create a default Arcade Drive project:

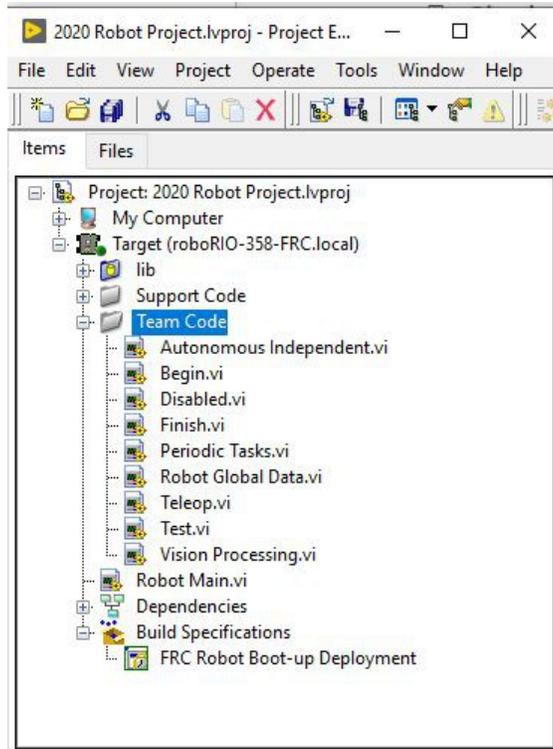
1. Open LabVIEW and you should see this window if everything has been properly installed (Note: all that stuff under OPEN is just a list of the most recent files that I had opened previously on my PC here at home):



- 2) Choose FRC roboRIO Project under NEW and you will get this:



- 3) Enter your team number in the highlighted field and click FINISH to get your default Arcade Project. Your Team number will identify your robot to the competition field and to your Driver Station app running on your PC. There will be a lot of robots and the team number is how we tell them apart.



Expand the Target and Team Code directories and you will see the vi's that you should be placing your new code in.

The most important of these are:

- Autonomous Independent.vi will contain your code running during the Autonomous period at the beginning of each match.
- Begin.vi is where you should open all the devices you will be using, e.g., joysticks, motors, sensors.
- Periodic Tasks.vi this is intended to contain user code that you always want running in the background.
- Teleop.vi runs during the Teleop period of each match and is where you typically map your driver controls to actions on the robot, like driving.

This project is a complete user code, as is, and can be downloaded to run your roboRIO. That's a good first test anyway.

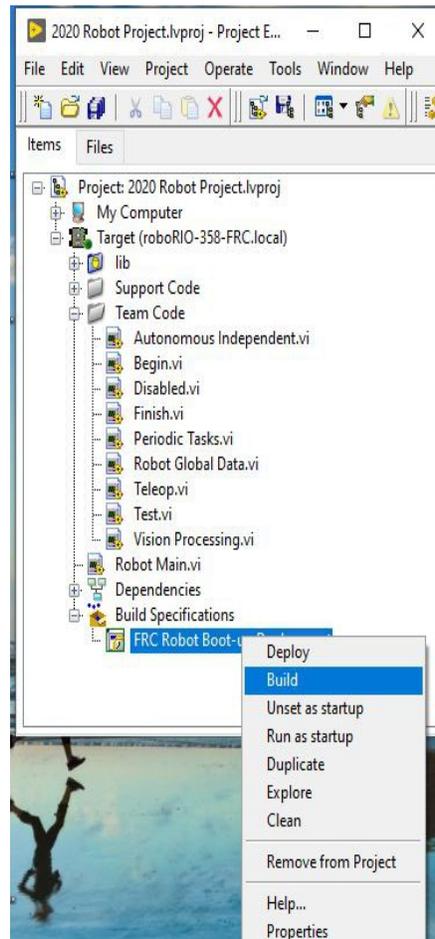
Whenever you work on your robot project, always open the whole project so you get everything necessary to run. That's a file with the name of your project and the file extension *.lvproj*.

Now, to complete the steps in beginning operation, compile and download your unchanged project to you roboRIO and test it.

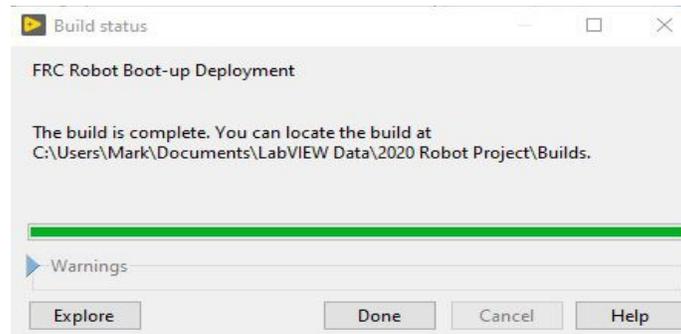
A) At the very bottom of your project file list is “Build Specifications”.

Expand that to see the default build spec “FRC Robot Boot-up Deployment”.

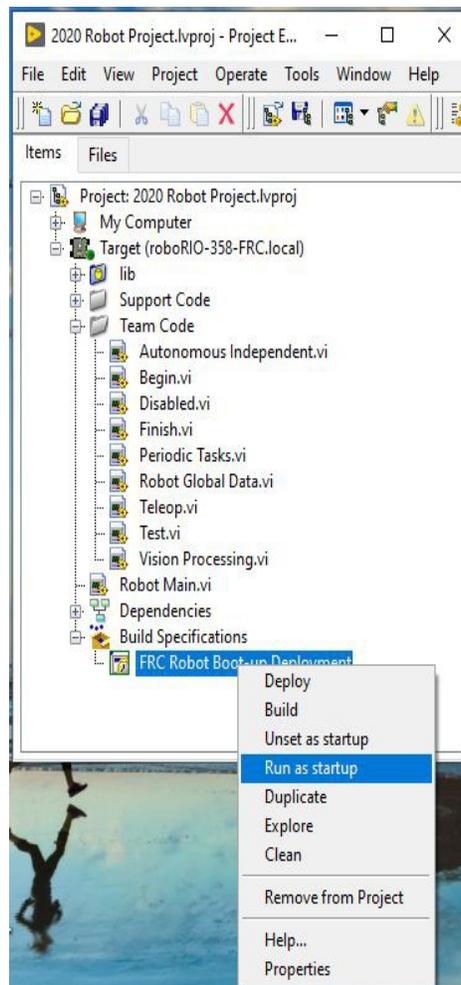
Right-click on that and choose “Build” to compile your program.



B) That will show a bunch of pop-ups as it compiles and checks your default program. When it is done it will stop on this pop-up and just click Done. After the first time you do this the roboRIO will always have a running program on it, so in the future there will be an additional popup warning you that there is already a program running on the roboRIO and that you will be overwriting it. When you see this just confirm that you want to replace the current roboRIO code with your new code.



C) Then connect a USB cable to your roboRIO and you can download your compiled program. Right-click on that build specification again to choose “Run as startup” to start the transfer.



That's the process of getting your code onto the roboRIO permanently.  
Now go test drive it.

This default Arcade project assumes you are driving using a single joystick (either arcade joystick or a game controller style will work).

It assumes the joystick is in the first (0) position on the PC Driver Station USB tab, so check that.

It also assumes PWM0 and PWM1 are connected to the motor controllers.

---

To operate your FRC robot requires the Driver Station app be running, so it can Enable the robot.

For safety, a Driver Station must be in communication with the robot, so it can be easily stopped in an emergency or if the robot becomes dangerously uncontrollable.