Introduction

Welcome to learning programming with the Romi robot! In this course, you'll learn how to program robots for fun or for competition, and how to write software, which you'll be able to do for robots, and for other uses as well. This course assumes no prior experience of writing software. If you have some prior experience and know the basics, you will still find value in this course as it relates to programming robots and the Romi in particular. If you're familiar with FIRST Robotics and want to learn how to program robots that compete in FIRST competitions, this is the perfect course for you. By the time you finish this course you'll have built up skills to contribute to any team you're a part of and created a foundation for yourself for future learning, leading you on a path to becoming an expert software developer. In this first lesson, we'll focus on setting up the Romi and making it move.

Setting up your robot and getting the first program to run can be the most frustrating part of programming. Unfortunately, there is no way around this. But, if you follow the steps here, you will succeed, and you'll be driving by the end of this lesson. If something fails the first time you try it, don't give up – once you make it through this lesson, the succeeding lessons will be more interactive and fun. Also note that although this lesson seems long by page count, most of the pages are filled up with images, so it will be faster than you might think.

Necessary Materials

To complete this course, you'll need the following things:

- A Romi robot (of course.) You do NOT need the arm attachment.
- The data USB cable that comes with the Romi kit. It's important to use this specific cable, as some USB cables are for power only and will NOT work.
- One or more joysticks/controllers for driving the robot. This course uses a gamepad-style controller (such as an Xbox controller, Logitech F310, or similar) for the examples, but you can use any controller recognized by your computer, as long as it has at least two joystick axes. The course explains how to account for differences in controllers, although there are some instances where driving may feel uncomfortable if you only have a single joystick.
- A supply of AA batteries. You will likely want rechargeable batteries because you will go through a lot of batteries in this course, and it will be expensive and wasteful if you can't recharge them. The Romi uses six batteries at once and a full charge will last up to a few hours under normal use where you spend more time writing code than driving but can run out faster than that if you spend a lot of time driving. The Romi comes with six batteries, so you can get started on the first couple lesson with those batteries while waiting for a rechargeable kit to kit to arrive, if you purchase a kit online.
- A computer with wireless internet access. If you have a computer without wireless access, you can buy a wireless adapter for less than $20. You need wireless internet to connect to the Romi to deploy code and drive.
- A computer with a wired (ethernet) internet port, and an ethernet cable. This can be the same computer as the one with wireless, or a different one. You will need to plug an ethernet cable from your computer directly into the Romi during setup, but do not need to access the internet with a wired connection at any point. If your computer doesn't have an ethernet port, you can use an ethernet USB pigtail to plug an ethernet cable into a USB port instead. (You can search Amazon for "ethernet USB pigtail" if you need to purchase one – they're typically $10-20.)

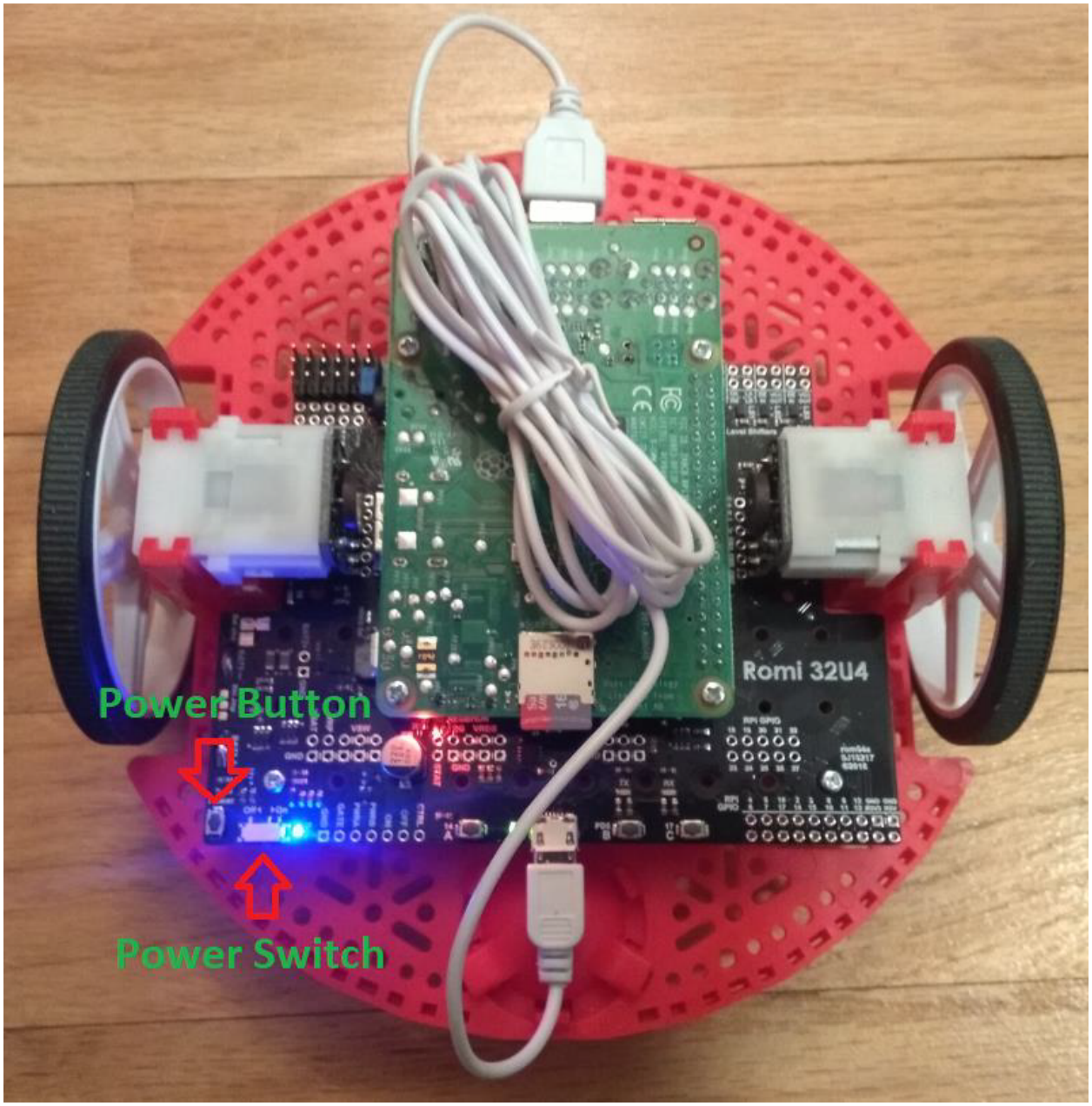
• An internet connection. In addition to connecting to the Romi, this course has links to external materials you'll need to access. In this lesson, you'll need to download two files which are each about a gigabyte, which can take a few minutes on slower connections. These are the only large downloads in the course.
• An SD card reader/writer. The Romi comes with an SD card and you will need to install firmware on this card before your robot will work. If you're using a laptop, there's a good chance it already has a slot for SD cards.

Setting Up Your Romi

There are two main steps in getting started, both of which are explained in detail below: installing VSCode with the WPILib library, and installing firmware onto your Romi. If you know that one or both of these steps have been done for you already, you can skip that step accordingly. If you aren't sure, then do not skip the step.

Turning your Romi On and Off & the Low Battery Signal

The Romi has two power switches – one is a switch, and one is just a button. To turn it on, you simply have to flip the switch to the "On" position. But to turn it off, you have to flip the switch to "Off" and then push the power button. If you don't do this, the Romi will remain on and eventually drain its batteries, so make sure you fully turn it off whenever you're not using it.
When the Romi is low on batteries, it will start playing noises. If you hear it playing noises (you'll know it when you hear it, it's loud and annoying), that means the batteries are low and need to be changed. Change them right away as it will stop functioning correctly if you don't.



Video Tutorial

If you prefer videos over text instruction, you can follow this link to see a video of how to set up a Romi. If you prefer to follow a list of steps, or are unclear about any part of the video, you can follow the text steps outlined below. Video: https://www.youtube.com/watch?v=mop51tpWWcA

Step 1 – Installing VSCode and the WPILib

Programming the Romi is done using a library known as “WPILib”. WPILib is built into an installation of an application called “VSCode”, which is what you will use to write and deploy

code to the Romi. There is documentation for the WPILib which includes guides on how to get started. To begin, follow the guide linked below to download VSCode. Note that the VSCode download and the Romi images are 1-2 gigabytes each and may take a few minutes to download.

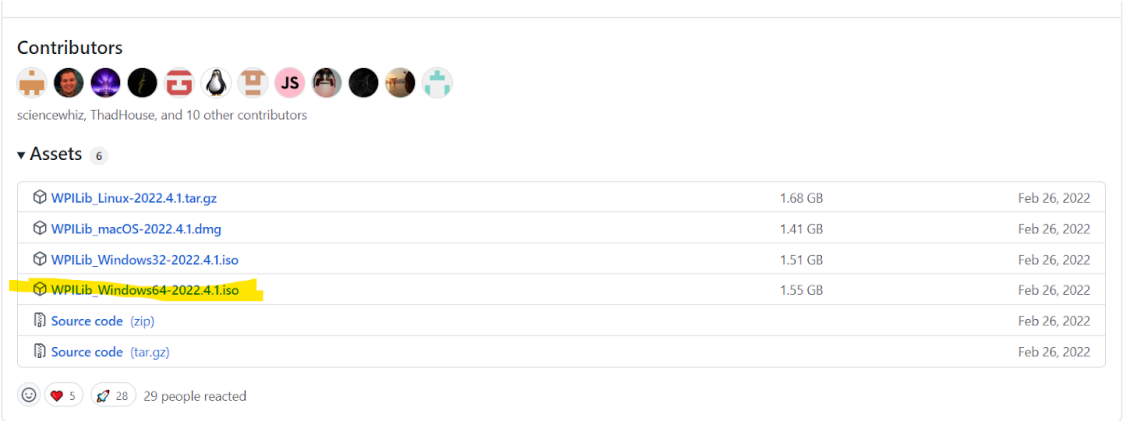
Installing VSCode

1. Start by opening this link:

https://docs.wpilib.org/en/stable/docs/zero-to-robot/step-2/wpilib-setup.html

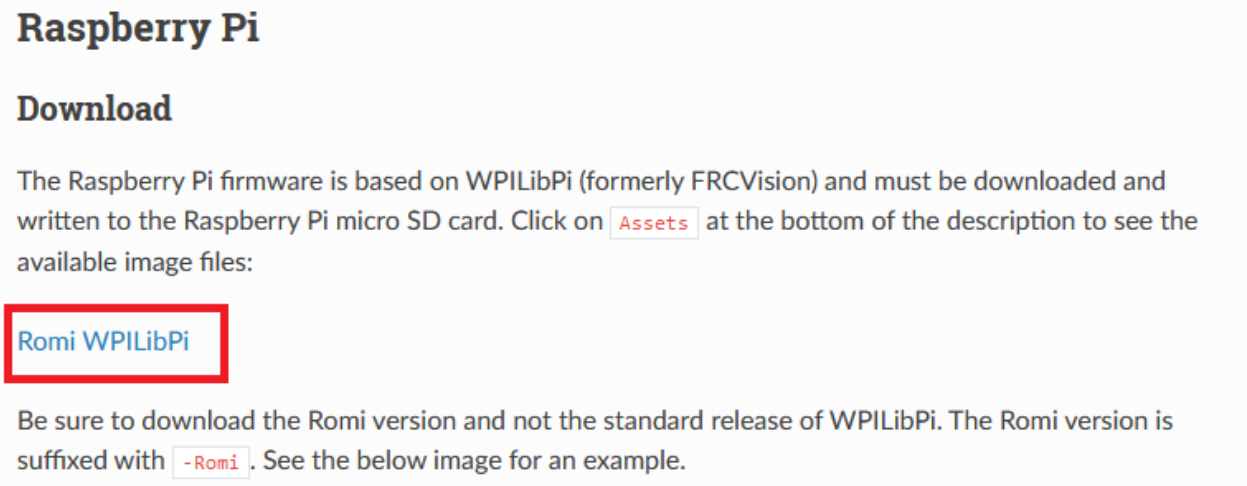
You'll be using Java.

2. There will be a link to GitHub which you need to follow:
3. Once on GitHub, the actual downloads are at the bottom of the page; you'll need to select the proper one based on which operating system you're using (Windows, Mac, or Linux) and whether your system is 32-bit or 64-bit. You don't need to read any of the other text on this page, you just need to find the right download. If you're on Windows and unsure which one to download, look at sub-point underneath the image below.



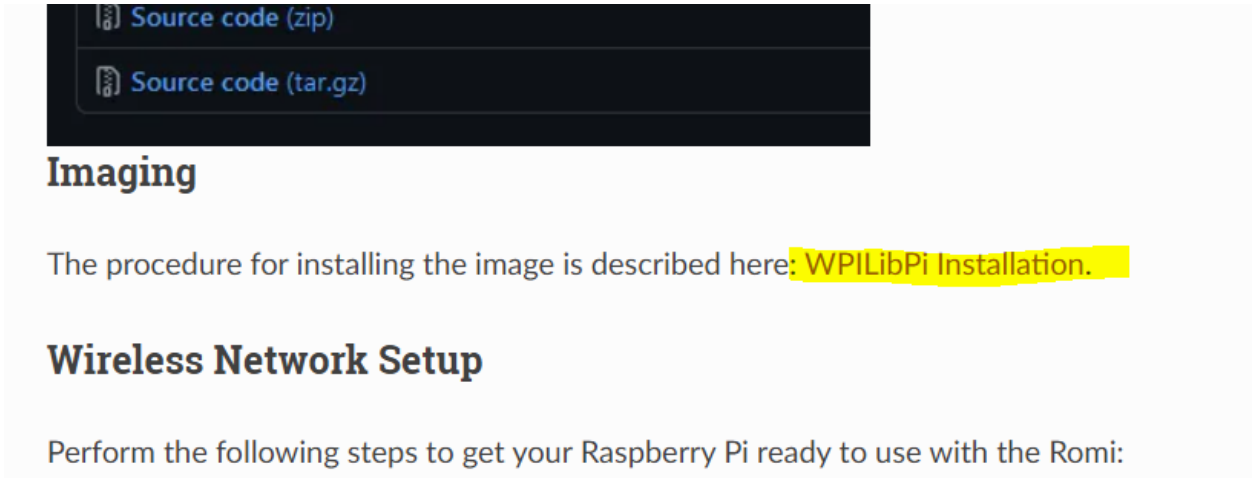
4. Once you have the right file downloaded from GitHub, go back to the WPILib documentation using the same link as before:
https://docs.wpilib.org/en/stable/docs/zero-to-robot/step2/wpilib-setup.html 5. In the documentation, follow all the steps on the page, starting with the "Extracting the Installer" step. There are images to guide you. If you are unsure what options to choose at any point through the setup, follow the default recommendations that the documentation gives you. If you're unsure about "Install for this User" or "Install for all Users", you can choose "Install for this User".
6. Partway through the page you'll notice a green warning similar to this: If this is your first time installing VSCode, you probably don't need to worry about this warning too much, but if for any reason you have a version of VSCode installed your computer already, pay attention to this.

Only the new version that you're installing will work – any prior installation will NOT work, even if it was done for a robotics team. 7. Make sure to read through the page thoroughly while you follow the steps. It is easy to make a mistake, but if you do follow the steps exactly, it will work. If it does not work, you can attempt to backtrack and see what step you missed. Starting over from scratch can sometimes be an easier option as well. Step 2 – Setting Up Your Romi To set up your Romi, you need to download and install firmware on it. You can think of firmware as kind of like an operating system for the Romi. It needs to have firmware installed in order to execute the programs that you write. There are a lot of details in the documentation during this section, but you don't need to read all of them. Follow the steps listed here to successfully install the firmware and configure your robot for use. 1. Start by following this link: https://docs.wpilib.org/en/stable/docs/romi-robot/imaging-romi.html You don't actually need to read the information on this page, but keep this link as a reference because if you want to do more advanced things with the Romi after reading through this course, you can refer back to it. This page is important for ALL users, regardless of whether your Romi came pre-assembled. The first think you'll need to do is download the RomiWPILibPi image. The steps to do this are a little confusing so follow closely: a. Click on the "Romi WPILibPi" link which takes you to a GitHub page:

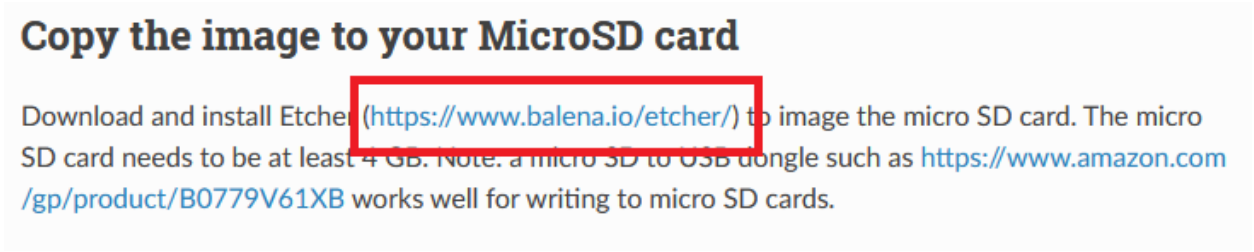


b. There will be a large link on the top of this page that looks similar to the image below, although the exact numbers and dates may have changed relative to the image. Make sure you're at the top of the page and that you did not scroll down. Before clicking on this link, look at the scrollbar in the upper right corner of your browser – it should be quite small. After looking at it and visualizing how large it is, click the big WPILibPi link. It may appear as though nothing on your screen has changed.
C.

d. Once you've downloaded this zip file, go back to the WPILib documentation, on the page called "Imaging your Romi": https://docs.wpilib.org/en/stable/docs/romi-robot/imaging-romi.html
e. Under the heading "Imaging", there is a link for a WPILibPi Installation guide. Click that link:
f.

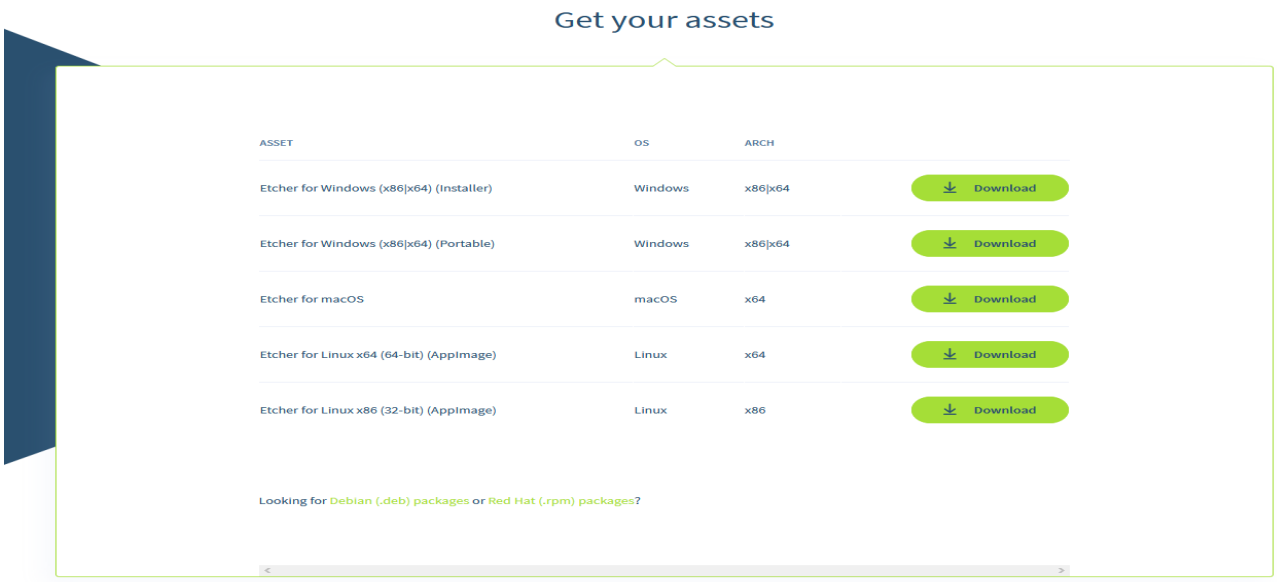


f. On this new page, you can ignore the first link to a WPILibPi repository – that's where you just were, and what you just downloaded. Read through the instructions and you'll get to the heading "Copy the image to your MicroSD card". Then you'll see a link to installing a program called Etcher. Follow that link:

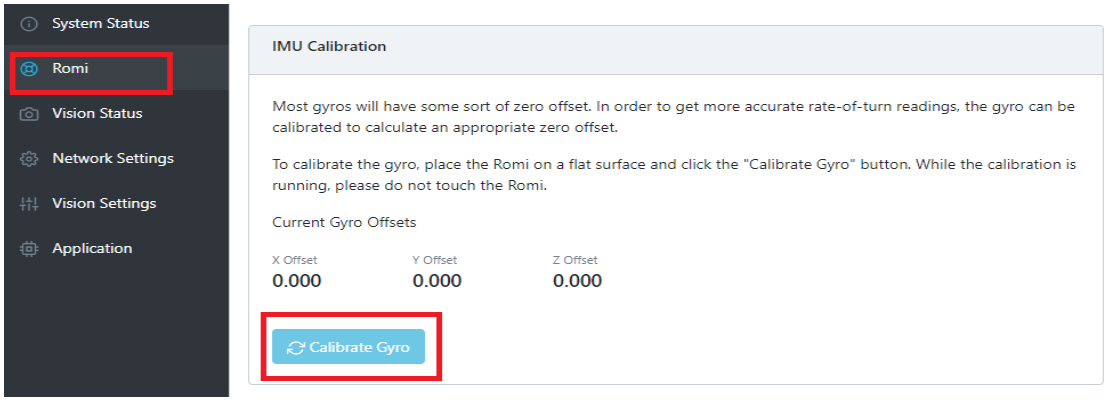


g. On this page, scroll down to the heading "Get your assets" and select the proper download for your operation system:

h.

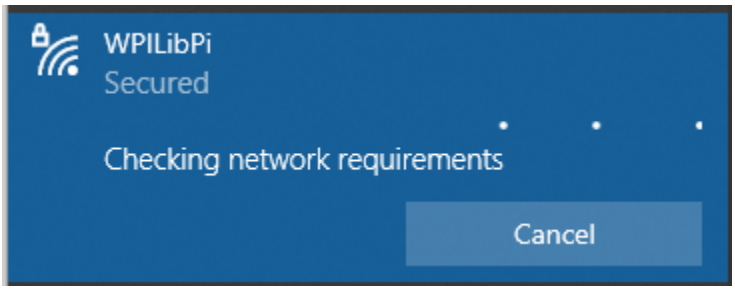


h. When you're done downloading Etcher, head back to the imaging page in the WPILib documentation: https://docs.wpilib.org/en/stable/docs/software/vision-processing/wpilibpi/installing-the-image-to-your-microsd-card.html
i. Follow the instructions on this page where you left. It explains how to flash the MicroSD card with the image using Etcher. Note how it states: "Expect the process to take about 3 minutes on a fairly fast laptop." If the flashing process happens very quickly and does not take a couple minutes, it did not flash correctly, but Etcher may still say that the flash is complete. If this happens to you, double check that you selected your downloaded zip file (do NOT unzip it first), and check that you selected your MicroSD card as the target. When it flashes correctly, it will go through a few different phases. First, decompressing. Then it will pause for a moment. Then, flashing, then validating, then finishing, where again it will pause for a moment. You can refer to the images below – click "Flash from file", select the zipped image, select the target as your SD card, wait for it to finish, and then it should show the "Flash Complete!" Message after flashing successfully.
j. After you've successfully flashed the SD card, head back to this page: https://docs.wpilib.org/en/stable/docs/software/vision-processing/wpilibpi/installing-the-image-to-your-microsd-card.html and scroll down to the bottom, to the "Testing the Raspberry PI" section, and continue following the steps.
k. For step 2, plug an ethernet cable directly from your computer into the Romi and navigate to the URL provided.
m. Finish completing the steps on this page and click "Next".
l. After clicking next, you get a long page with a lot of info. You can ignore a lot of this info, although you can read it if you're interested. Read from the start of the page up until the "System status" heading, at which point you can stop if you want.
m. In the web interface, navigate to "Network Settings". Click "Writable" at the top. On this page, you can set the name of your Romi's wifi network if you want to in the SSID field. You can also leave it as WPILibPi if you want to, but if you're in a class or lab with other people you might want to choose something unique. You can also enter a wifi security code in the WPA2 Passphrase field if you want.
n. Lastly, click on the Romi tab, scroll down to the "IMU Calibration" section, read the instructions (lay the Romi on a flat surface and don't touch it while calibrating), and click "Calibrate Gyro". When you're done, click the "Read-Only" button at the top.

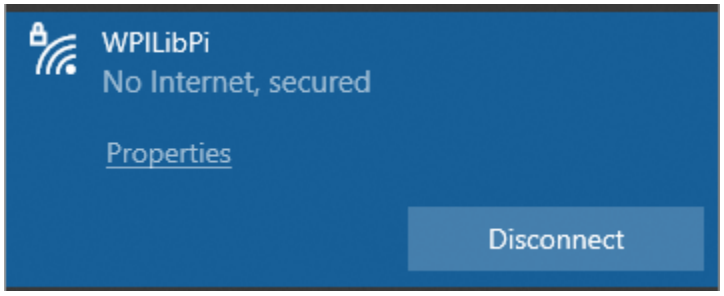


o.

o. You're now done with the network configuration. You can unplug the ethernet cable. Now you can connect to your Romi wirelessly the same way you would connect to any other wifi network on your computer. The network name will be whatever you set it to, or if you did not change it, it will be whichever network starts with "WPILibPi". There are a couple things to note about connecting to the Romi over wifi. First, when you do so, you will disconnect from any other wifi networks you're connected to, meaning you will probably lose internet access if you're connected to your internet over wifi. You can maintain internet access and connect to the Romi at the same time if you connect to your internet with an ethernet cable, but if you're not connecting to the internet with a cable, be prepared to lose access. Third, sometimes on Windows the network connection screen might look like this while you are connected:



If your screen sits with scrolling white dots, you're probably connected after a few seconds. It might also look like this:



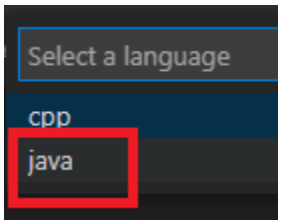
In this case, it says "No Internet", but you're still connected to the Romi – you just don't have access to the internet, because your Romi is just a robot, not a router that connects you to the outside world like you're usually connected to with wifi.

Congratulations! You're finally done setting up your Romi. It's time to make it move!

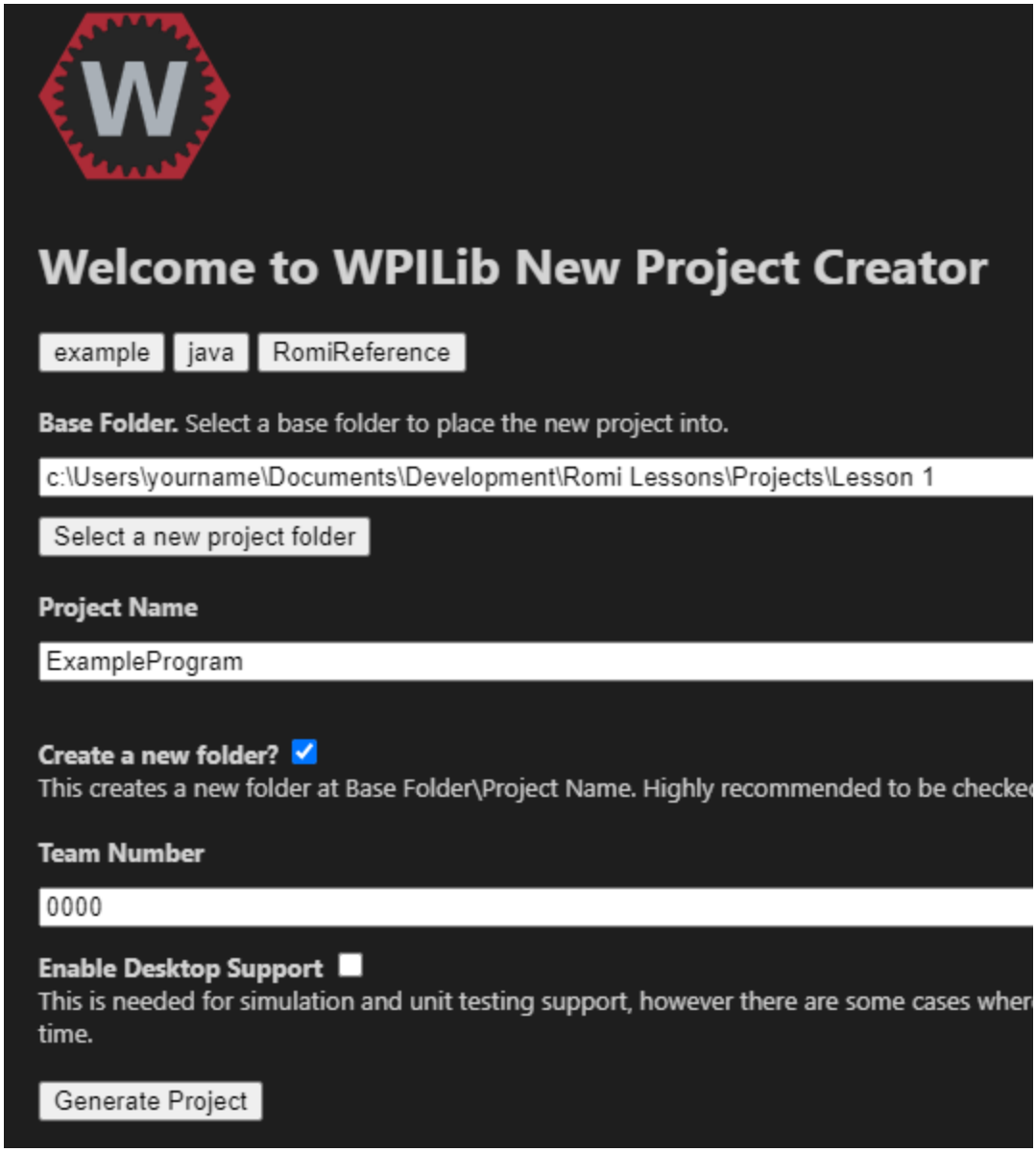
Part 2 - Running Your First Program

Navigate to this page in your browser:
https://docs.wpilib.org/en/stable/docs/romi-robot/programming-romi.html and open VSCode. Make sure you're connected to your Romi through your wifi network.

Follow the steps in the linked page all the way to the end. There are a couple items that you'll need to specify that are not covered by the linked page. First, when it asks you to select a language, pick Java:



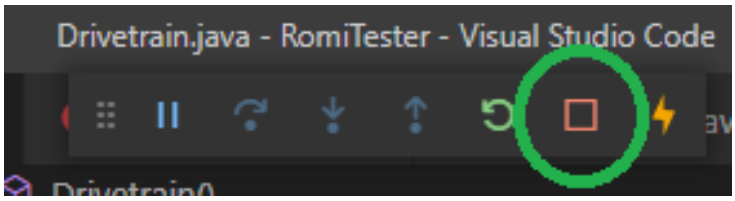
For the "Base Folder" field, you can choose whatever you want in your filesystem, but it might be a good idea to create a folder called "Development" in an easy-accessible location, where you put all your Romi projects. For the project name, you can again enter whatever you want. "ExampleProgram" might be a good name for your first project. Leave the "Create a new folder?" checkbox checked. For "Team Number", if you belong to an FRC team, you can enter your team's number. If you do not, or you don't know what an FRC team is, you can enter "0000". Do not check the "Enable Desktop Support" checkbox. Here's an example:



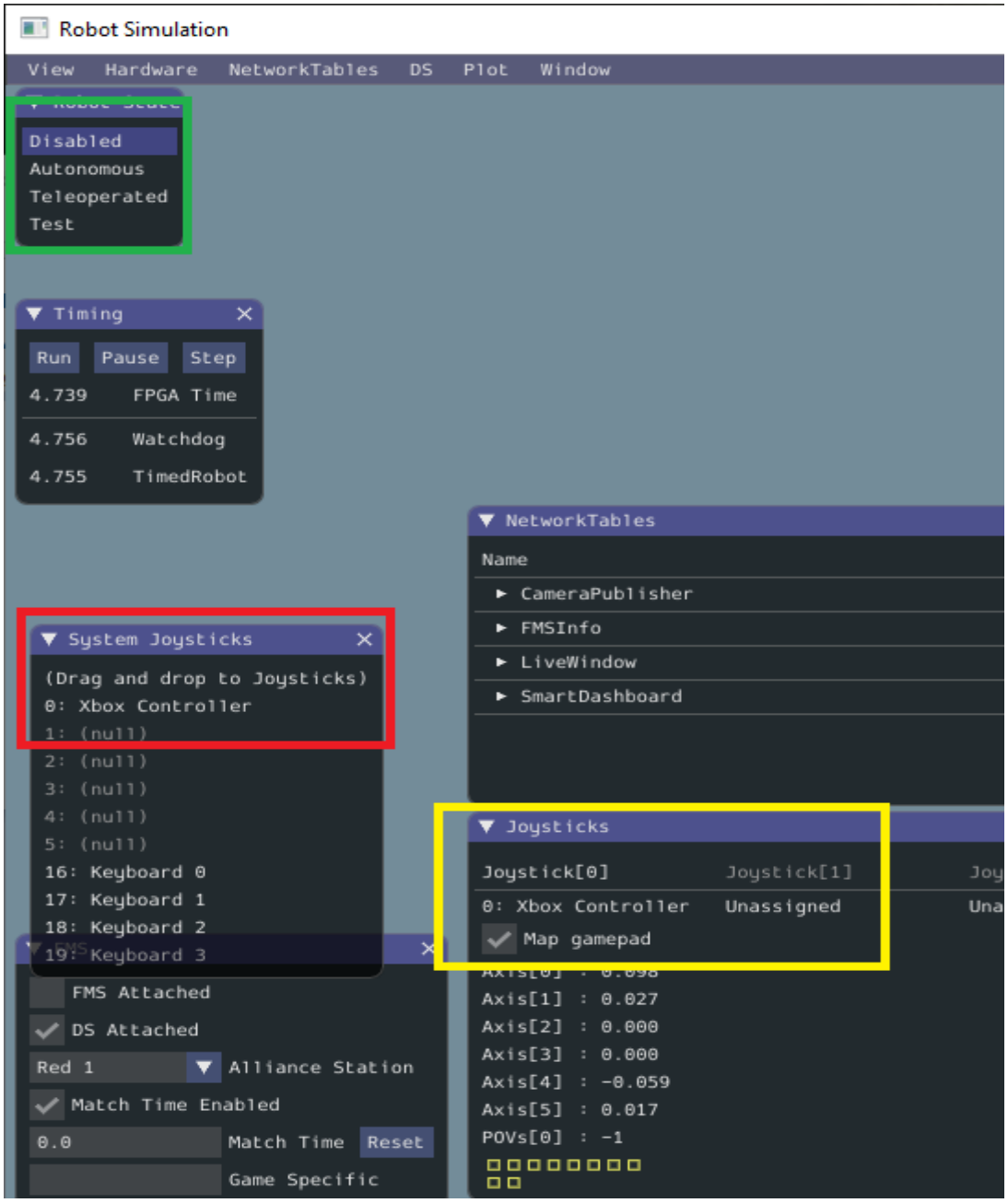
Back on the “Programming the Romi” webpage, you can ignore the paragraph right near the end that says “If you changed the Romi network settings…”. This means the last thing you’ll do in the instructions is press F5, which will deploy the example project. Go ahead and do this!

If when you press F5 you get any sort of build error (you will see red text near the bottom of your screen saying build failed), try rebooting VSCode and your Romi, re-connecting to your Romi’s wifi (remember that will take a minute for the wifi network to pop up), and deploying again. Sometimes VSCode will fail to contact the Romi even if there is nothing wrong with your code, but rebooting solves this.

After you deploy your code, VSCode will bring up a new interface (called the “Robot Simulation” interface) that looks something like this. Note 1: Sometimes the Robot Simulation window will not open the first time you press F5. If this happens to you, simply click the red square to end your program, and press F5 again.



Note 2: If you have the FRC Driver Station application installed on your computer, you may not see the Robot State box in the upper left of the image below. Instead, you will use the FRC Driver Station app to enable/disable your robot and select modes. If you don't know what the FRC Driver Station is, don't worry about this note.



This interface lets you control the Romi and see various output from the onboard sensors. But what we're most interested in right now is making it move. The simplest way to do this with the sample program is to enable its autonomous program. Enabling the autonomous program will cause the Romi to drive forward about ten inches, turn around, drive back to its original location, and then turn around again. Make sure you have the Romi somewhere where it can safely do this (not at the edge of a desk or table) and be ready to enable it. Look at the photo above, and find the "Robot State" box, in the upper left, highlighted in green. Here you will see a list of four robot states – Disabled, Autonomous, Teleoperated, and Test. As soon as you click Autonomous, the Romi will enable and immediately begin its autonomous routine. When you're ready, give it a try! After the Romi finishes moving, click the Disabled option to disable it again. Although the Romi is lightweight and low power, making sure robots are disabled when not in direct use is a critical safety practice.

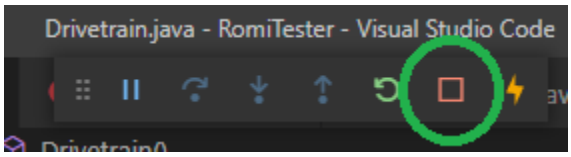
Now that you've made the Romi move on its own, it's time to try driving it yourself. If you look at the window called "System Joysticks", highlighted in red, you will see any joysticks that you have plugged in to your computer. In the example, there is an Xbox controller plugged in, but you can use anything that your computer recognizes as a joystick. There's an additional window called simply "Joysticks", highlighted in yellow, and these are the joysticks that your code and Romi are looking at for input. You may not see any joysticks listed in this window. If that is the case, you can simply drag and drop your joysticks from the System Joysticks window, to the Joysticks window. In the example photo, the Xbox Controller has been dragged to joystick 0. Do likewise – drag whatever joystick you want to use as your controller to the 0 slot. If you have multiple joysticks plugged in to your computer, you may need to do some quick tests to figure out which one is which. How to do this is explained next.

After dragging your joystick so the Joysticks window, try moving your joystick around and pressing some buttons. You will see the yellow squares at the bottom of the Joysticks window fill/unfill as you press and release buttons, and you can see the axis values change in value as you move the sticks around. This is how you can test which joystick in the System Joysticks is which if you are not sure. If you have one joystick dragged to the joysticks window and pressing buttons/moving the sticks doesn't change any of the values, you probably dragged a different stick.

Once you've gotten your joystick to register, and you've confirmed it's in the Joystick[0] slot, you're ready to enable your robot! Look back at the robot state box, and when you're ready, click Teleoperated. This will enable the robot. Now you can start trying to drive it around! If you press various buttons and/or sticks on your joystick/controller, you should be able to make the robot move. However, it may or may not control as you expect. For example, by default, if you're using an Xbox controller, the left joystick will control the robot moving forward and backward, but the left trigger will control turning, and you'll only be able to turn it one direction since the left trigger can only register positive values (as opposed to a joystick, which can register positive or negative values, e.g. if you push it forward or backward.) Now disable the robot. Your first task for you to start programming will be to fix the controls so it can drive normally. If your robot does drive as expected, which could happen for some controllers, then you can still follow the lesson. You'll simply change it to drive abnormally, and then fix it again.

Ending the Executing Program

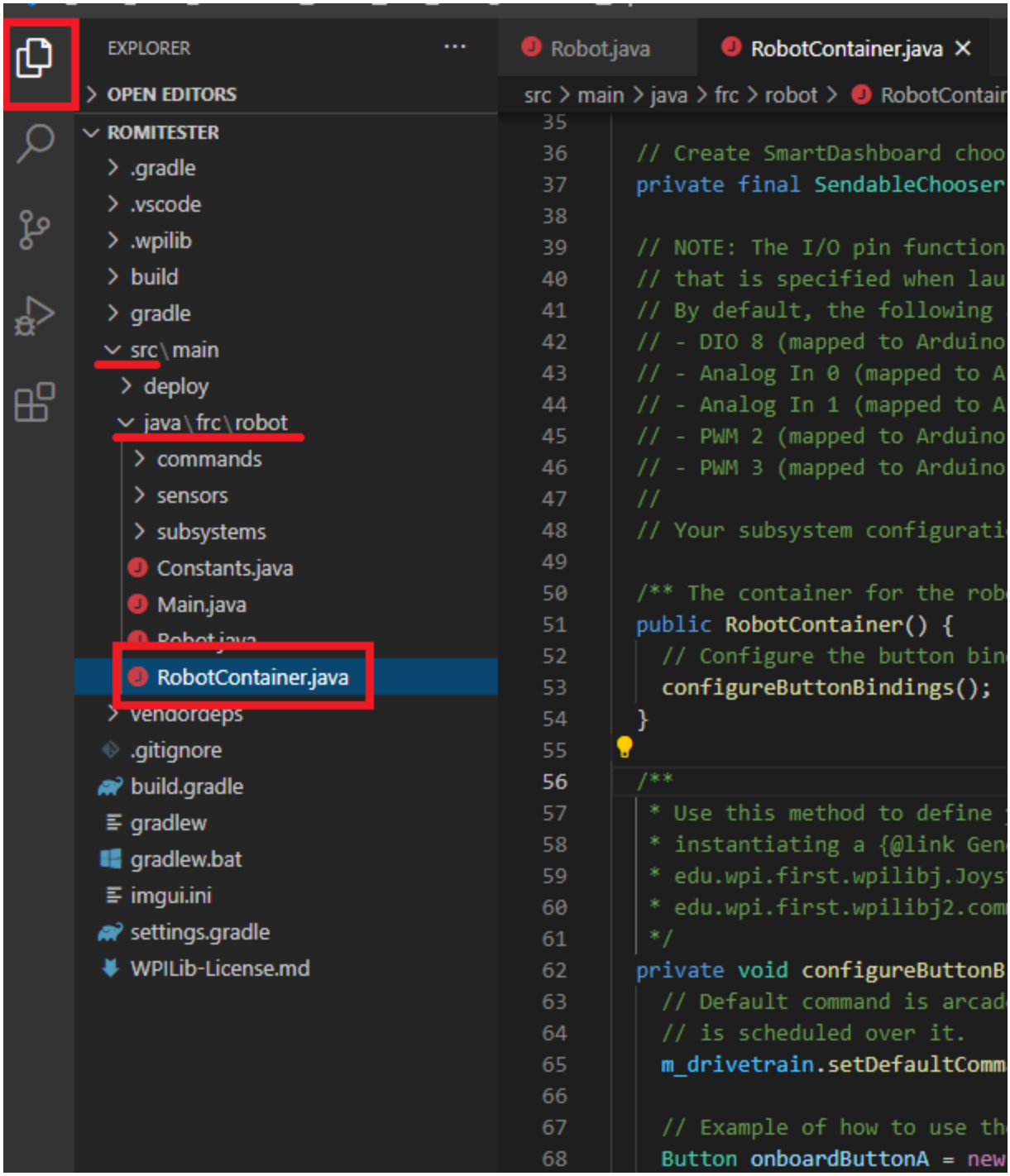
After you press deploy your code, a small box with a few buttons will pop up in the top of VSCode. While this box is visible, you may not be able to deploy additional code to the Romi – VSCode may give you an error. When you want to deploy new code, simply click the red square in this box first, and then you can deploy your code:



Fixing the Controls

As discussed above, you can look at the robot simulation interface, and see changes in the joystick section as you move the joysticks or press buttons. Decide which joystick axis you want to control turning. For example if you're using an Xbox controller, this might be the left joystick being pushed left or right, or the right joystick being pushed left or right. If you're using a single standalone joystick, you don't have as many options. But either way, decide what you want, and then start pushing the joystick that way. On the joysticks interface you will see a value change. For example, it might move from 0 to 1 or -1 depending on which direction you push it. The important thing here is to note which axis is changing. The axes are labeled from 0 to 5 in brackets, like so: Axis[4]. Determine which axis is changing, and remember this number.

Now let's dig into the code. Back in VSCode, open the "Explorer" (highlighted in red, upper left, see the picture below) and then click on src, java, frc, robot, and then finally double click on RobotContainer.java to open it.



If you're brand new to programming, there is a lot to take in here – but don't worry about that. We'll discuss everything in time, but for now let's get those controls fixed. Scroll down in RobotContainer.java until you see line number 97. The lines are numbered on the left side of the code. On line 97, you'll see this:

```
94    */
95   public Command getArcadeDriveCommand() {
96     return new ArcadeDrive(
97         m_drivetrain, () -> -m_controller.getRawAxis(1), () -> m_controller.getRawAxis(2));
98   }
99 }
100
```

At the end of the line, you'll see "m_controller.getRawAxis(2))", highlighted in red above. This line sets which axes of the joystick are being used to control the robot. Change the 2 to whatever axis you identified when testing the controls. If your controls were already correct, try changing it to something else, following the next step, and then changing it back. Once you do this, you can press ctrl+s to save, and F5 to deploy your code to the Romi again. Once the code finishes deploying, you can enable it again and try out your new controls. If you put the right axis, you'll be able to drive your robot comfortably! If the controls aren't what you expected, double check your axes again when you push on them using the joysticks interface, and make sure you changed the correct value in the code. Now that you have your Romi driving around, have some fun with it! You've taken your first steps into writing code and you already have a robot driving around.