1   Overview

As mentioned in the project document, there are a lot of parameters that your robot must know about in order to complete the competition. This includes knowing which corner it starts in, where the red and green zones are, and so on. This information will be sent over WiFi to your robot at the start of the round, after which your robot can localize and do its job.

The software that we provide to do this is split into two parts: a server and a client. This zip file contains code for both parts.

1.1   The Server

The server receives and responds to requests from a client, in exactly the same way your computer is a client that connects to servers running websites. In this case, your robot is the client and you can test with the server on your laptop. In the final competition, the TA/professor will run the server and they will decide what data to send to your robot.

The server code is in the `Server` folder, provided to you as a JAR.

1.2   The Client

The EV3 client code we provide can be split into two sections. First is the `WifiConnection` class, which handles all communication to and from the server. The second is the `WiFiExample` class, which provides an example of how to use the `WifiConnection` class. Once it has received data from the server, it prints a few select parameters to the screen. It is up to you to add the `WifiConnection` class to your project and process the data sent appropriately.

The client code is in the `Client` folder, provided to you as a JAR. The example code is in the `Client Example` folder.

2   Using the WiFi

To help you get started, we provide some step-by-step instructions on setting up the WiFi and getting the provided example code to work. As mentioned in Section 1.2, it is up to you to learn from the example and get the WiFi working with the rest of your code.

2.1   Importing into Eclipse

The example code provided in the zip file has been given to you as an Eclipse project so that you can import it easily.

1. Copy the `EV3WiFiExample` folder from the `Client Example` folder to your Eclipse workplace

2. In Eclipse, go to `File→Import→General→Existing Projects into Workspace`

3. Browse and select the `EV3WiFiExample` folder

4. Make sure the `EV3WiFiExample` project shows up

5. Click Finish

When it's done importing, right click on the `EV3WiFiExample` project and select `leJOS EV3`→`Convert to leJOS project`

You can now run the WiFi code as an EV3 project on your robot and the server as a regular Java program on your laptop. That said, the WiFi still needs to be set up in order to do anything useful.

2.2 WiFi Setup

1. Connect to the WiFi network with your laptop. There is an access point running in the lab (it will be the same during competition):

   - SSID: DPM
   - Password: dddpppmmm

2. Connect to the WiFi network with the EV3. First, plug in the USB WiFi adapter you were given at the start of the semester and either reboot leJOS or start it if your robot was off. Once leJOS has booted up, navigate to the `Wifi` menu, select the `DPM` network, and enter the password. It should connect, and an IP address in the format `192.168.2.x` will appear on your home screen, where `x` is a number unique to your robot, at least until you turn it off. From now on, whenever you start the robot it should connect automatically to the WiFi network.

3. You now need to configure your robot to connect to your laptop. To do this, first find out what IP address you were assigned by the DPM router.

   - On Windows 10, you can do this by clicking on `Properties` on the `DPM` network once you are connected to it. The resulting setting page will have your IP address listed at the bottom as `IPv4 adress`.

   - On Linux, the `ifconfig` utility will show it under `inet addr` for a network likely to be named something along the lines of `wlan0`. On more recent versions, you may have to run `ip a` instead of `ifconfig` and/or look for an interface name starting with `wlx`. Your GUI of choice (GNOME, KDE, etc.) will also display this information in one way or another.

   - On Mac, open the `Network` tool from `System Preferences` and look for the `IP Address` field of your WiFi adapter.

   Regardless of your OS, there are plenty of online guides that will show you how to do this if you're not sure where to look.

2.3　Running The Example Code

1. With your laptop's IP address in hand from the previous section, modify the string `SERVER_IP` in the example code (`WiFiExample.java`) to match. You will also need to modify the `TEAM_NUMBER` variable as appropriate. Keep in mind that in the competition or beta demo, you will connect to the TA or professor's laptop using the address `192.168.2.3`.

2. Run the server program as a normal Java program on your computer. You can do so by either:

   (a) Navigating to the `Server` folder using a command line console and executing

   ```
   java -jar "DPM Server Fall 2018.jar"
   ```

   (b) Double clicking on the JAR file (in most operating systems).

   Keep in mind that no matter how you run it the file `layout.xml` must be in the same folder as the JAR.

3. While the sever program is running, launch the `EV3WiFiExample` program on your robot; it should connect to your laptop and wait for data. You should see a message appear in the GUI saying *Team X connected*. You can now enter numbers into the GUI or use the `Fill` button, which lets you specify an XML file to load data from. There is an example XML file included alongside the server JAR file named `example_data_fill.xml`.

4. Once the data is entered in the text boxes, click `Start` on the GUI. You should see a bunch of messages appear detailing the program's communication with the EV3 and if the data was successfully sent or not. Note that if you leave one of the team number boxes empty (or set it to 0), the program will understand there is only one team and won't complain about not being able to connect to some other team that doesn't exist.

5. If all goes well, your EV3 should now display the data you just entered. The program will end if a button is pressed after the data is received. Generally speaking, if anything goes wrong the program will simply print an error message, allowing you to quit by pressing any button. If necessary, you can tell the `WifiConnection` class to stop trying to talk to the server by pressing the `Back` button.

3　Integrating with your code

Once you have confirmed the example code works, you have to integrate the `WifiConnection` class into your own code. To do so:

1. Open Eclipse

2. Right click on your project in the *Package Explorer* and click on `New→Folder`

3. Name the new folder `lib`

4. Copy the file `EV3WiFiClient.jar` from the `Client` folder to the new `lib` folder you just created

5. Right click on your project in the *Package Explorer* and click on `Properties`

6. Navigate to `Java Build Path`→`Libraries`

7. Click on `Add JARs...` and select the `EV3WiFiClient.jar` file inside the `lib` folder.

8. You can now import the `WifiConnection` class as shown in the example code and use it in your project.

Note that in the `EV3WiFiExample` project, these steps have already been done for you.

4   Important Points

Please keep the following in mind:

- We expect you to use the `WifiConnection` class to communicate with our server.

- The parameters sent are the same as described in the project description document, with the addition of _x or _y for coordinates. For example, if BRR_LL is $(5, 7)$, then your robot will receive two parameters named BRR_LL_x and BRR_LL_y with values 5 and 7 respectively.

- In the beta demo and final competition:

  - The TA or professor will run the same server code as provided to you.
  - Recall that you will need to change the `SERVER_IP` variable to `192.168.2.3`.
  - The professors will decide what parameters the server will send to your robot. As a result, your code must be able to handle any valid input as specified in the project description.
  - The very first thing your robot should do is connect to the sever and download the parameters. Your robot should only start moving *after* having received data.

5   Useful information

- The WiFi test code uses `System.out.println` statements that print to both the screen and, if connected, the `EV3Control` console (over WiFi/Bluetooth/USB). This is particularly useful if you need to debug as reading output is much easier on your laptop than on the LCD screen. There is a boolean variable `debugPrint` in the constructor of the `WifiConnection` class that will disable printing of the messages if you don't want to see them.

- The file `layout.xml` in the `Server` folder contains a list of all possible parameters. You may wish to refer to it when writing code to process the data sent over WiFi.

- It is possible to use the `WiFiConnection` code over USB, Bluetooth or your own WiFi hotspot if you use the IP address for the correct interface on your laptop. Feel free to do this when testing, but make sure it actually works over WiFi during the demo and final competition.

- Post on the discussion board if you have questions about the code or using the WiFi in general. You may also email me at tharsan.ponnampalam@mail.mcgill.ca.