LAB 19

TEC 284

USING THE ESP32 REDBOARD

OBJECTIVES:

* Use Bluetooth LE to remotely control an Arduino
* Familiarize yourself with the ESP32 ecosystem

SUPPLIES:

1x SparkFun IoT Redboard, 1x Adafruit car chassis from previous lab

INTRODUCTION:

*In this revisit of the race car lab, we are going to use the ESP32 to add some wireless capability to your race cars. The ESP32 is capable of WiFi and Bluetooth communication. We are going to be combining a lot of resources together to make this lab happen.*

THE EASY STUFF

1. Let’s start with the easy parts of this lab. One lab partner will need to download a free app that is available on the iOS and Google Play store. The app is called Bluefruit Connect. It is completely free and doesn’t require an account.
2. You’ll also need to grab a copy of your code from the race car lab.
3. Alright, that’s the easy stuff. Let’s move to the setup!

SETTING UP THE ARDUINO IDE

1. The great thing about the ESP32 is that it can be programmed just like an Arduino in the Arduino IDE. We just need to download the correct board definitions.
2. Open the Arduino’s board manager. You will need to search for and download the “ESP32” board definitions, as shown below:

A screenshot of a computer

Description automatically generated

1. This might take a while to download. Once it finishes, we can tell the Arduino IDE that we are using a “Sparkfun ESP32 IoT Redboard” on whatever COM port.

A screenshot of a computer

Description automatically generated

STARTING THE BLE SERVER

1. The other great thing about this lab? A lot of it is written for you! In fact, we only have to make a few, tiny changes.
2. Open the example sketch located at: File > Examples > BLE > UART
3. Since all of you will be making a Bluetooth server, we need to personalize your own server so that you can tell it apart from your classmates’.
4. On line 71, change the name of the BLE server from “UART Service” to something you can remember. “<Your name> BLE Server” works fine.
5. You may need to change line 42 if you get an error message:

std::string rxValue = pCharacteristic->getValue().c\_str();

1. For now, let’s upload this and run through some things with the Bluefruit app.
2. Uploading will take longer than you’re used to. Once it finishes, open a serial monitor. You’ll need to change your baud rate to 115200.
3. Open the Bluefruit Connect app. You should see your BLE server. Connect to your server.

Screens screenshot of a computer

AI-generated content may be incorrect.

1. On the following screen, click on “Controller”, and then “Control Pad.” You probably see where this is going! Press some of the directional arrows and watch your IDE’s serial monitor. If you set it up correctly, you should see something akin to: “Received Value: !B804”.
2. You’ll have a total of 8 values for the directional arrows. There will be a unique value for pressing and releasing each button. What are those values? You’ll need them in the next step. **Include the exclamation point**.

|  |  |
| --- | --- |
| **Action** | **Value** |
| Up Arrow Pressed | !B516 |
| Up Arrow Released | !B507 |
| Left Arrow Pressed | !B714 |
| Left Arrow Released | !B705 |
| Right Arrow Pressed | !B813 |
| Right Arrow Released | !B804 |
| Down Arrow Pressed | !B615 |
| Down Arrow Released | !B606 |

1. Now we can begin to write some code. Make a new line after line 62. You can actually remove line 62, as it clogs up the serial port. Make some skeleton IF statements (or test / case statements, whatever you’re more comfortable with) with the values above. Here’s an example of what it should look like:

if (rxValue == "!B516") {

      // GO FORWARD

        } else if (rxValue == "!B615") {

      // GO BACKWARD

        }

1. Don’t worry about actually writing anything in those IF statements for now. We’ll come back to that soon.

COMBINING PROGRAMS

1. Now comes the hard part. Open your code from the race car lab. If you remember, in that lab, we had a distance sensor that we used to tell our car to move either left or right if it detected an obstacle. If you did the lab correctly, you should have a bunch of functions already written. You’ve already *done* the work, now we need to combine the programs together.
2. Essentially, everything except for the distance function (and its two associated variables and pins) needs to be copied over. Keep it in the same section of code, so that stuff in the loop() in one program stays in the loop() in the other. Remember that functions need to be written outside of the loop().
3. Remember all of those IF statements you wrote on the previous page? That’s where you’ll call your functions. If you detect a right arrow press, you call the function to go right.
   1. We never really covered “stopping” the car. If you want to do this, the code to do so is:

rightMotor->setSpeed(0);

leftMotor->setSpeed(0);

rightMotor->run(RELEASE);

leftMotor->run(RELEASE);

1. You have a number of ways to do this. You can have it so that the car moves so long as the arrow is pressed and held down. You can have each arrow press correspond to 3 or 5 seconds of motion, or you can do it however you want.
2. Make sure that your function definitions come before the class MyCallbacks definition! Otherwise, you’ll get an error message.

CONCLUSION

1. When all of your IF statements are working, you should have a fully remote-controlled car! You can take it for a drive now, you’ve earned it!
2. Since this is the last “lab” that we do in class, I want you to think back to when you first built this race car in lab 5. Think about how much you have learned and created within these few, short weeks! The world of electronics, automation, and tinkering are full of fun things like these labs. There is so much to learn about this field, and it is constantly growing and expanding as new technology emerges. I hope that, if anything, this class has shown you a glimpse of this world, and I hope you have had fun and learned something. That’s all for now!

FINISHED PROGRAM

1. When you have finished, submit your code to your GitHub. You may want to include instructions on how to install / use the Bluefruit Connect app.