/A\ughdon Breslin & Sal Montero

Learning Module #6 - Controlling a Roomba

Initial Proposal

Aughdon and Sal’s goal for the end of this learning module is to control a Roomba robot through an Android phone app. We want to start by familiarizing ourselves with the components required to make this happen, so we will probably sift through some YouTube videos and websites, looking for guides/tutorials. We’ll need to know the language that will interact with the app, the role of the Arduino within the Roomba, how to connect the Arduino to the app via bluetooth, among plenty of other things we’ll run into along the way.

We will both be involved with researching guides/helpful tips and developing a plan to control this Roomba, as well as writing up the code and setting up the circuitry required. Days 1 and 2 will primarily be focused on researching the components of a Roomba that will allow us to interact with it, while days 3 and 4 will be geared towards figuring out which of the language’s functions can be used to move the Roomba. Days 5, 6, and 7 will be based on following the schematic of the Roomba, and disassembling parts of the Roomba to place in our Arduino. On days 8, 9, and 10 we will be coding the robot and connecting it to the app. Days 11 and 12 we will probably be revising the code and optimizing our set up.

1. Day 1 (1/4/2019): We viewed several websites and found the parts required for our project. We also decided that Audi will bring in the Roomba on Monday, so we can start working with it.
   1. <https://www.hackster.io/mjrobot/controlling-a-roomba-robot-with-arduino-and-android-device-56970d>
   2. <https://create.arduino.cc/projecthub/mjrobot/controlling-a-roomba-robot-with-arduino-and-android-device-56970d>
   3. <https://github.com/Mjrovai/Roomba_BT_Ctrl#readme>
   4. <https://boredomprojects.net/index.php/projects/controlling-roomba>
      1. (1/6/19) Aughdon looked into what can be accessed with the roomba. I disassembled it as much as I could without breaking anything, and uncovered a lot of dust, cleaning tools, and batteries. Nothing looked like a serial port, but apparently it’s supposed to be on the top of the Roomba (7:00 - 8:30).
2. Day 2 (1/7/2019): We got a LOT more accomplished than we thought we would today. We managed to connect all the wires to the roomba without much trouble at all. We found the serial port (we had to snap off the top cover, which was pretty scary), and then it all went very smoothly. We plan to connect everything and possibly start coding next time.
3. Day 3 (1/9/2019): Sal was preoccupied with choir today so Aughdon solo’d it, mainly looking at what functions we would need to operate the Roomba. Such functions include turning on, driving forward, stopping, rotating left/right, and turning off. I wrote the basis for these functions.
4. Day 4 (1/10/19): We continued working on writing the Arduino functions today. We figured out that the way to compile all of the functions to the arduino was to condense everything into two files, the .ino file and a header file for variable declarations. Just as we were about to plug in the arduino for our first test, we realized that the roomba is dead… So we’ll test it tomorrow once it’s charged :)
   1. (1/10/19) 2 hours @ home : Sal managed to clean up all of the code that we wrote today during class, and fixed a little error with a variable never being declared before use…
5. Day 5 (1/11/19): IT WORKED! Took about 20 minutes for Sal’s computer to finally start, but the code worked and we’re set to move onto bluetooth.
   1. (1/14/19) ~2 hours @ home : Sal setup all that’s needed to connect using bluetooth, and setup sounds to be played. MADE GITHUB REPOSITORY AND UPLOADED CODE <https://github.com/inlineskater720/roombaProject>
6. Day 6 (1/15/19): Now that we got the motor controls for the Roomba working, we started setting up bluetooth capability. After going over Program or be Programmed Ch6 essays and getting back LM#5s, we searched for a Bluetooth module to hook into the Arduino and eventually operate the Arduino from a mobile app.
   1. (1/15/19)Aughdon looked up Arduino apps that would connect to bluetooth, turns out only Android phones are compatible because iOS is trash. We want to see whatever ways it can be controlled, so we’ll need to download Arduino Bluetooth apps with multiple controllers. I downloaded several onto my tablet, I may have downloaded malware, but I found apps with 6-10 different controls (8:30-10:00).
7. Day 7 (1/16/19): Today we got the bluetooth connected to Xander’s (android) phone and were able to send inputs from a downloaded app to the Arduino via a tilt controller. We plan on basing the controls on the change in x, y, and z dimensions. For example, if x > 3, move forwards, if x < -3, move backwards, if it’s in between, stop.
8. Day 8 (1/17/19): We were able to control the Roomba today. Using an emulated game controller, we bound our inputs to characters that activated functions that moved the Roomba. We were also able to make the Roomba emit notes and sing tunes.
9. Day 9 (1/22/19): We discussed the Amazon vs Apple Article-Response essay for a while. Then we figured out how to get and order data input from three different variables simultaneously using “structs” and if statements.
10. Day 10 (1/23/19): We used the structs to organize the data and from there we were able to check the value of each variable in a given struct. By the way a struct is basically a space to hold other values, so we use structs to collect the information and then look into the struct for the values of x, y, and z, and based on the magnitude of these variables we were able to control the Roomba via tilting the phone various directions.
11. Day 11 (1/24/19): We spent today making sure the Roomba could be controlled by both tilt control and gamepad controller simultaneously. It was actually pretty annoying setting up both in the same program, and ran into problems distinguishing between the controller modes. Both still work, but getting to each controller setting is currently not working.
    1. Hour @ home : Sal fixed the code that wasn’t working from day 11 :) Accidentally nested an if statement inside another one and the code would never reach that certain point....
12. Day 12 (1/28/19): We have both modes set in one program, but we just have to manually change the value of roombaMode from “tilt” to “gamepad” if we want to go between. So we cannot use both controls in one go without reconnecting the roomba to the computer and uploading the code, just changing roombaMode’s value. Nevermind we’re amazing/stupid lul. We just set extreme values for the tilt control as the change to gamepad, and changed one of the buttons on gamepad to change the mode to tilt.

Final Report

This project was presented by Aughdon Breslin and Salvatore Montero. We saw a video of a guy controlling a Roomba using an xbox controller, and thought it was pretty cool and that we could do better. So, we set off thinking out how we could go about controlling a roomba, and went in immediately. We didn’t have a super specific goal in mind yet, which is pretty similar to how we’ve started other learning modules, but we were looking to explore our options. Our options expanded a lot when we found a serial port where we could hook up an Arduino.

At the beginning of the module, we mostly relied on hackster.io [Day 1 Log (a)] to set up the Roomba with the Arduino. After we hooked up the Arduino, we pretty scarcely used the site, with the exception of replugging in the Arduino’s wires to the Roomba’s port. More directly related to the Roomba, we had to use an Android phone for its compatibility software, so we downloaded some apps (both sketchy and professional) hoping to find one to emulate a controller without too many problems. These apps were all called along the lines of “Bluetooth Arduino Controller.”

While the project did go mostly smoothly, we had some problems (worries) at first of how much we’d need to dig into the Roomba to be able to work with it. It took us a while to figure out the snap-off top, and pulling that off, just hoping it wouldn’t break each time, was scary. We had trouble finding apps that wouldn’t just download malware (luckily it was Xander’s/Rob’s phones and not ours), and one of the apps we end up needing to use still spams us with ads. We simply had to deal with this because of a lack of play store regulations, so we bit our tongues and downloaded quite a few apps and sought out the best ones. Our main problems from there on out was coding (rookie) mistakes. I assume Sal wrote most of his code pretty deep into the night, because most of the errors were simple fixes. We did have trouble switching between tilt controls and gamepad controls at first, until we realized we were stupid and could just find some condition that normally wouldn’t get activated. All of our problems for this module were overcome as we were able to accomplish our initial goal of controlling the Roomba through gamepad controller. However we did not try connecting the Kinect to the Roomba because we were intimidated by the C# language the Kinect was in and were not up to learn it in the remaining days of the learning module.

We ended up being able to control the Roomba through tilt and gamepad controls. The gamepad was straightforward as a button would correlate to an input, and an input could be programmed to a function. The tilt was a bit more complex, because instead of it input it was a constant feed of information for three variables, so we had to organize the three and set boundaries where if a value was high/low enough it would set off a function.

We learned quite a bit from this module. We learned that iRobot is completely programmer-friendly, that Sal really likes to jump right into coding, that Audie is super forgetful, and that projects can be relaxing. We were super hyped when we saw the Roomba working, but we weren’t really stressing beforehand. A smooth project is a blessing. The more familiar you are with Arduino, the more easily you’ll be able to adapt the Arduino to do literally whatever you want, even if it to just manually control an automatically-controlled vacuum cleaner.

Because of how smooth the project went, I really wouldn’t do much differently if we were to start again. We’d maybe pace out the coding so when errors popped up we weren’t brigaded with many at once, but we didn’t run into many errors that weren’t typos or syntax.

However, we do have suggestions. One, do not be intimidated by Detrick telling you that you might get in trouble for doing this project. You must work past this fear and break down your project as far as possible while still being able to reverse it before you can start giving into that fear. Breaking is only breaking if it no longer works, so nobody needs know about a little scratching. If you want to buy a Roomba though, they have “educational purpose” discounts for like $40 when we applied. This was still outside our price range as $120 is more than double our combined bank accounts. However, working with Roombas is fun because they’re built to bump into stuff and can make particular strings of notes.

In the future, we’d like to try working through the Kinect route. I’m not sure how Roomba-related it would be, but controller-wise the Kinect or perhaps VR-setups would be a really cool to work with, and we could have a lot of fun with them.

Overall, our experience was pretty relaxed. This is especially because Aughdon didn’t have to worry too much about coding as Sal wrote in all the functions literally the night we got the Arduino hooked up. For the majority of the learning module we just had to troubleshoot and make small additions as we expanded the program’s capabilities. In the future, Aughdon would like to actually slow Sal down so he can get in some more Arduino hours, but he can’t complain too much as Sal is the Arduino master.