Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Section: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Names: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| LABORATORY CHECK OFFS | |
| **Section 1.1 Connecting IR sensor**  IR remote causes RGB led colors to change  **Section 1.2 Showing the actual IR codes**  IR codes are printed to serial monitor  **Section 2.1 Interfacing Bluetooth module**  Change RGB led colors from your Android application  **Section 2.2 Display Bluetooth data**  Display the data received (ASCII) via the serial port  **Section 2.3 Control robot navigation**  Control mobile robot navigation from your Android application | \_\_\_\_  \_\_\_\_  \_\_\_\_  \_\_\_\_  \_\_\_\_ |
| Points | \_\_\_\_ |

Prelab:

No prelab work required.

Learning Outcomes:

By the end of this lab you will be able to:

1. Control the robot’s RGB led using an infrared remote control
2. Control robot’s navigation and state transitions using Bluetooth module

Overview:

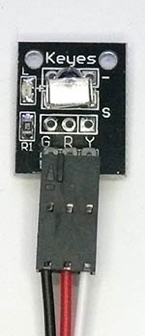
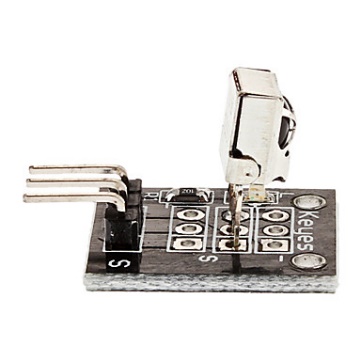
IR (InfraRed) remote controls are widely used for consumer devices such as televisions, set top boxes, air conditioners and other home devices. In this lab, you will add an IR sensor to your robot and use a program that detect IR remote control codes and sets the state of your robot (e.g. STOPPED, FORWARD, LEFT\_TURN, etc).

Your kit includes an IR sensor packaged on a small circuit board. The sensor datasheet is posted on MyCourses. The sensor detects infrared light modulated at 38kHz. Codes are send from a remote by pulsing the IR light on and off for varying periods of time. For more info on how IR remote control code work, see:

<http://www.sbprojects.com/knowledge/ir/index.php>

and

<http://arduino-info.wikispaces.com/IR-RemoteControl>

****

**SECTION 1.1 – Connecting the IR remote sensor**

The IR sensor has three lead, +5V power is in the center, ground(-) is on one side and, signal (s) is on the other.

**Procedure:**

1) Connect the IR sensor to your Arduino power and ground. Connect the signal to Arduino analog input pin A3. Note that in this case, we will configure Arduino pin A3 as a digital input.

2) Next, take out the remote control unit as pictured below.



3) Check that your remote has a coin cell battery installed and that the clear plastic battery saver tab is removed.

4) Connect the RGB led to your Arduino according to the following pin definitions:

#define LED\_DATA\_PIN 12

#define LED\_CLOCK\_PIN 11

5) Download the code file named: Lab9\_section1\_IRremoteRGBledYDR\_r1.ino from mycourses. Read through the code to get an idea of what it does. Upload the code to your Arduino. You will need the IRremote.h library which is installed on the lab machines, but also available on MyCourses.

6) Try pressing different buttons on the remote and observe the change in led color. At the same time look at the serial port monitor and notice what state names are printed and correspond to each color.

**SECTION 1.2 – Showing the actual IR codes**

The IR remote codes can be printed out directly to the serial monitor

**Procedure:**

1) Modify the code so that SHOWCODES is true instead of false.

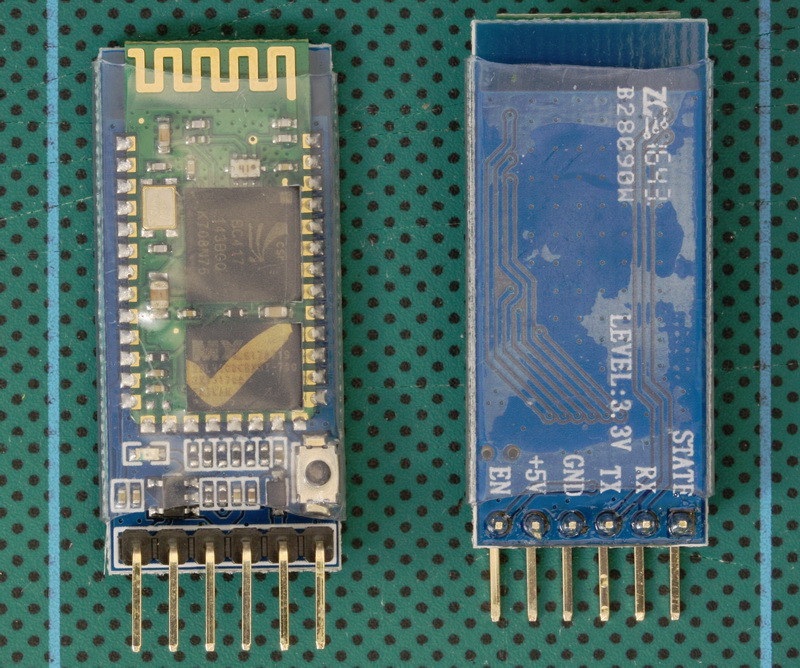
2) Run the code and observe the serial monitor printing out the code as Hex values.

Extra, Optional: try using your new remote control capability to control your robot.

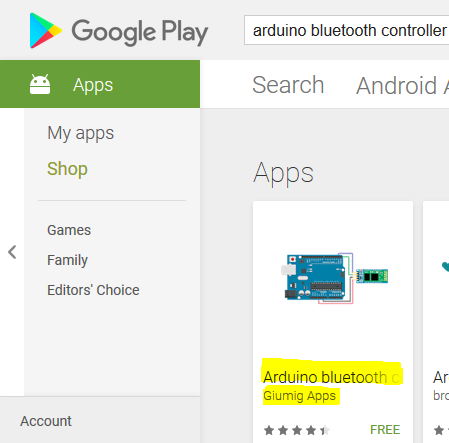
2. Bluetooth Communication using HC-05 module

Overview:

Bluetooth is a wireless communication standard developed originally developed as an alternative to wired RS-232 serial communication. Bluetooth has been widely adopted by hobbyist and consumer technology companies alike. In this lab, you will use the HC-05 Bluetooth module provided in your kit to control the robot’s motion and state transitions. You can read more on the HC-05 Bluetooth module in the file on mycourses: arduino-info - BlueTooth-HC05-HC06-Modules-How-To.pdf



HC-05 can be interfaced to an Arduino board to receive commands from another Bluetooth device. In this Lab you will interface a HC-05 Bluetooth module to your RoboRED board, and use a ‘Bluetooth Controller’ app running on an Android phone to send commands to your robot platform. You will have to download the following app (‘Android Bluetooth Controller’) or equivalent from the [Google Play store](https://play.google.com/store/apps/details?id=com.giumig.apps.bluetoothserialmonitor):

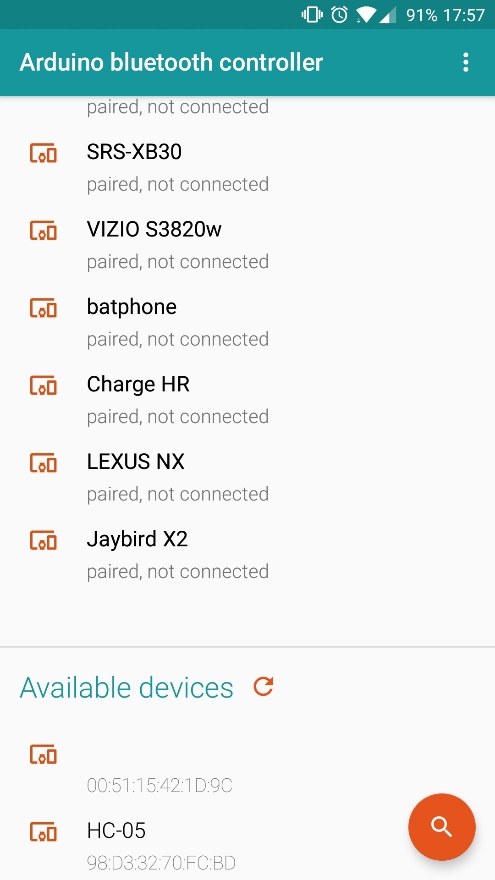


The application needs to be setup to send desired commands over to your robot, as described later.

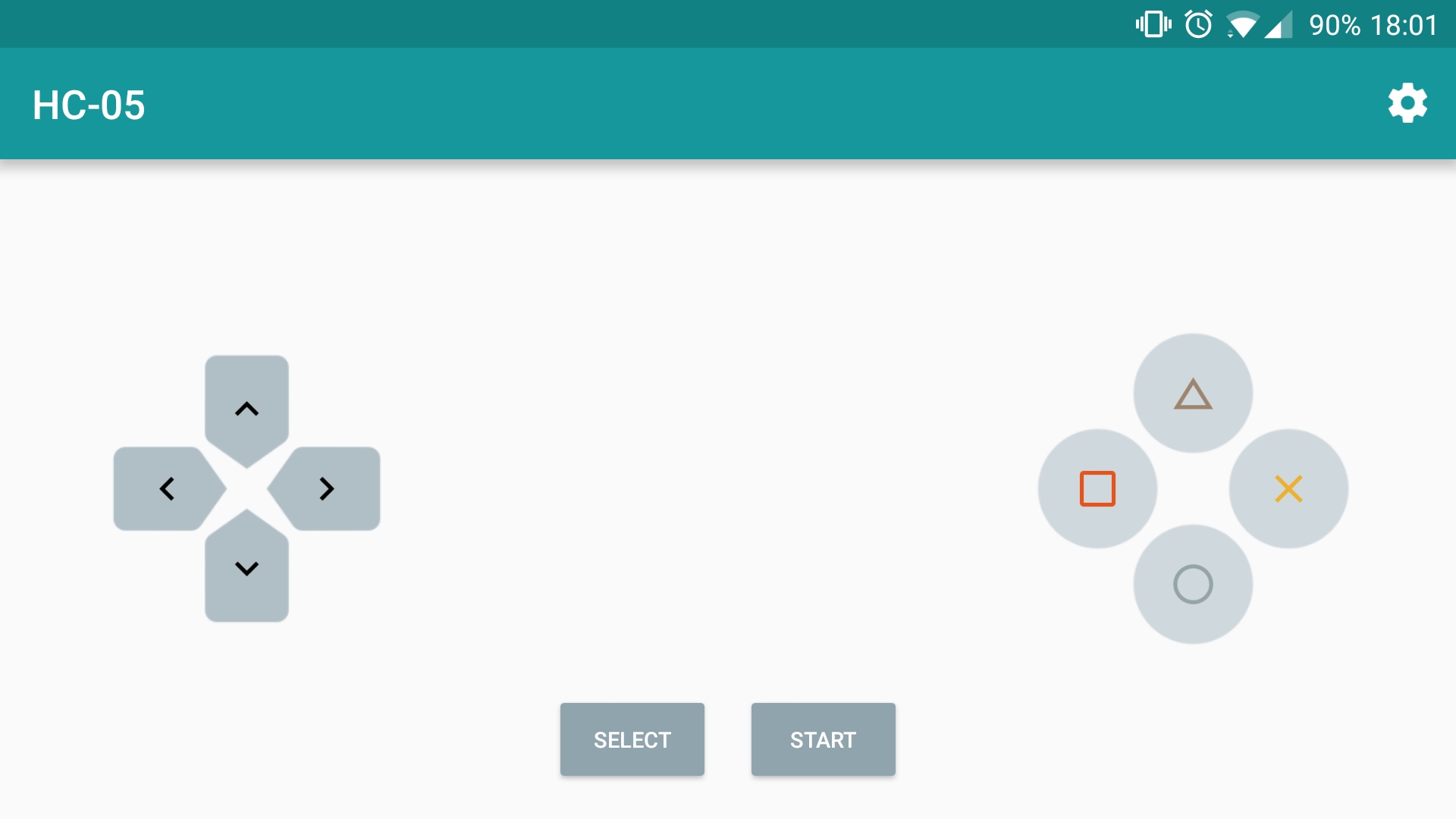
Procedure:

**SECTION 2.1 – Wiring up the Bluetooth module**

1. Connect the Pin ‘TX’ found on HC-05 module to pin ‘A0’ on your Arduino. Connect the ‘RX’ on HC-05 module to ‘A1’. Connect the pins ‘+5V’ and ‘GND’ on HC-05, to ‘VCC’ and ‘GND’ on your Arduino respectively.
2. On powering up your Arduino, you should see a blinking LED on your HC-05 module.
3. Download and install the ‘’Android Bluetooth Controller’ app on your Android device (refer above for app store link). *(If you have an I-phone, look for an equivalent Bluetooth controller app.)*
4. Open the app, and accept the permissions prompt (if any). Launch the app and scroll down to ‘Available devices’ part (Bluetooth should be ON and in ‘Discover’ mode on your Android). Look for ‘HC-05’ to be listed under this section.

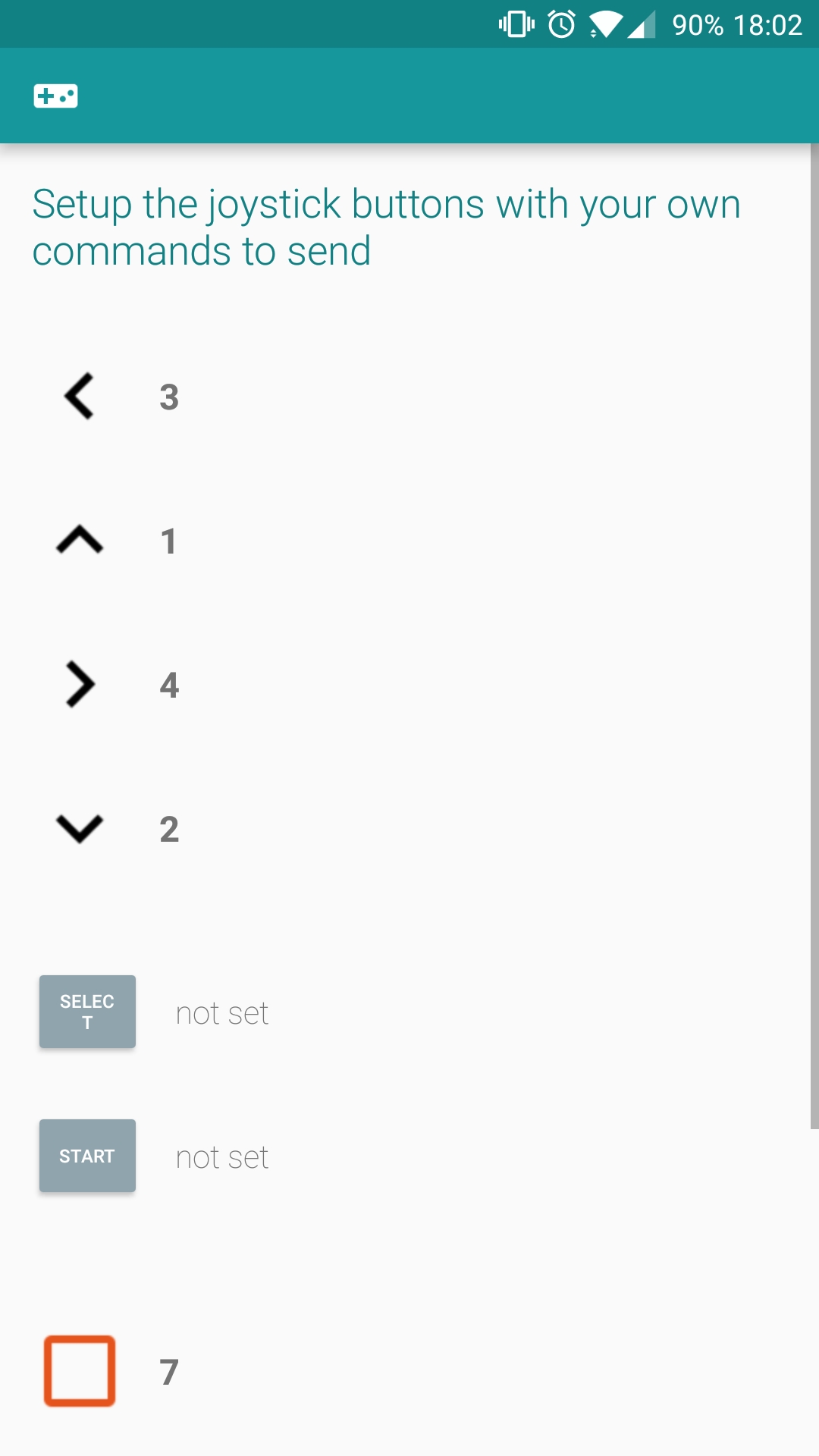


1. Select the device to pair. If prompted for password – use ‘1234’ as the password. Once paired, click on ‘HC-05’. This should bring prompt ‘Connect in’ menu – select ‘Controller mode’. You should have a game controller on your display:



1. When you press a button on the keypad e.g. “Square”, the app will send a decimal number to the HC-05 controller. For each button, you will have to set-up the ‘command’ that you wish to send to your HC-05 module from the controller. Do this by clicking on the ‘settings/gear’ icon on the top right corner and entering the values from the table below.

|  |  |
| --- | --- |
| Controller Button | Value to be set |
| UP | 1 |
| DOWN | 2 |
| LEFT | 3 |
| RIGHT | 4 |
| TRIANGLE | 5 |
| X\_BUTTON | 6 |
| SQUARE | 7 |
| CIRCLE | 8 |



1. Download the program ‘Lab9\_section2\_BTControlledRGBled.ino’ from myCourses. This is a skeleton program that lets you transition states and light-up the RGB LED strip with different colors using Bluetooth communication module. Read through the code and understand the operation of receiving commands from a transmitter. The program is built using the ‘SoftwareSerial.h’ Arduino library that allows your program to open a virtual serial port using pins ‘A0’ and ‘A1’ to transmit and receive data.
2. Upload the program to Arduino. You will need to have your ‘HC-05’ module paired with the android application, and have the app in ‘controller mode’. Try pressing different buttons and observe the change in LED color and notice the state transitions printed in the serial monitor.

Section 2.2: Display Bluetooth data

During the set-up process in the Android application, you would have set each game controller key with a numeric value. This is the command we wish to send to our robot, but Bluetooth will transmit the ASCII values of the numerical values assigned to each key. These ASCII values are provided as pre-processor directives in the program. You can change the SHOWCODES value to print the ASCII values received by the module on serial monitor.

*Note: The ASCII character ‘1’ is represented by a byte value of 49 when transmitted over the Bluetooth connection. ‘2’ is represented by a byte value of 50.*

#define UP 49 // ASCII Character 1. Forwards

#define DOWN 50 // ASCII Character 2. Backwards

Procedure:

1. Modify the pre-processor directive SHOWCODES to be able to print the data received by your Bluetooth module. Get a sign off for showing the codes.

Section 2.3: Control robot navigation

Procedure:

1. The switch-case portion of your program can be modified to control the navigation of your mobile robot platform. Modify the states by updating the ‘entry housekeeping’, ‘state businesses’, and ‘exit housekeeping’ enabling your robot to successfully move FORWARD, BACKWARD, LEFT, RIGHT, and to STOP.
2. Note: You can use the code from previous labs to make your robot move (with the gyro sensor providing feedback). Make sure to add the corresponding library files when using code from previous labs.