Lab 9: Control of Robot via SmartPhone App

EECE-3015 Introduction to Microcontrollers Lab

Cris Ababei
Dept. of Electrical and Computer Engineering, Marquette University

1. Objective

The objective of this lab is to develop a remote controller for the self-balancing robot. We use a webpage ran within a browser on the iPhone – which looks like a smartphone app that implements a joystick. The joystick is used to generate control commands, such as left, right, forward, back. We will test the remote controller with a test program that requires a separate hardware setup based on the Arduino Pro Mini and the BlueTooth module HM-10.

2. The Main Idea of the Remote Controller: Command Byte

We will remotely control the robot by regularly sending via BlueTooth (BT) a **command** generated based on the x-y position of a joystick. The command is always a byte (8 bits) and captures how the joystick is panned or tilted or a combination of that. The logic of creating such a command is described by the following code:

```
int command_byte = 0b00000000; // captures joystick position
if (tilt < 0) command_byte |= 0b00000100; // forward when tilt is negative
if (tilt > 0) command_byte |= 0b00001000; // back
if (pan > 0)  command_byte |= 0b00000001; // left when tilt is negative
if (pan < 0)  command_byte |= 0b00000010; // right
```

The remote controller will send "**command_byte**" as the periodic (e.g., every 100 ms) controlling command. The robot's BT module will receive that and use it inside the main program running on the Arduino Pro Mini of the robot.

3. SmartPhone Apps as Remote Controllers

Android:
NOTE: *You should read and do this section only if you are an Android (as opposed to iPhone) smartphone user and if you would like to get into Android app development.*
For a simple example of an Android App (installed on a Samsung Galaxy smartphone) that implements a joystick and which sends via BlueTooth the basic control commands discussed in the previous section, check out the following video starting at **minute 1:30**:
https://www.youtube.com/watch?v=pj5hVGT7MBc
The complete Android Studio project that implements that app is available on GitHub at:
https://github.com/eigenpi/YARY

iPhone:
For an iPhone smartphone, we could implement a remote controller joystick using an approach/workflow described below (to avoid implementing from scratch native iPhone apps):

- Create an HTML/JavaScript joystick
  - Touch position → pan/tilt → **command byte**
- Publish it using GitHub Pages (or similar)
  - Free HTTPS website with a shareable URL

- Open URL in Bluefy (a free iPhone browser that you need to install on your iPhone)
    - No custom iOS app installation or developer account
- Send one byte commands over Bluetooth LE (BLE) every say 100 ms
    - Robot's BT module must be paired to iPhone and receive command
- Robot's Arduino Pro Mini microcontroller decodes the **command byte**
    - Controls the stepper motors

I developed a specific implementation of this approach with the help of AI tools –this helped me circumvent the need for direct html coding. I verified and tested it and it works pretty well – as it will be demoed in class. You can try it yourself – first install Bluefy on your iPhone, then, open the following URL link:
https://eigenpi.github.io/HM-10-Joystick-Remote-for-iPhone/index.html
You should get the online joystick as shown in Fig.1 below. You could do this from a laptop computer as well.

We will use this to connect to the BT module and use the joystick to send control commands. This is demonstrated for now in the example presented in the next section.



Figure 1: Online web joystick loaded on iPhone via Bluefy web browser.

4. Arduino Test

The Arduino sketch example to test the remote controller is provided in the .zip file for this lab:
lab9_files/test2_bt_module/**test2_bt_module.ino**

To test it, you must first create the **hardware set-up separately**. Do not attempt to run this example test with the Arduino board and the HM-10 module plugged into the PCB board. You must

unplug the Arduino board and the HM-10 module and wire them up separately (Arduino board free in the air and the HM-10 module plugged into the breadboard; use jumper wire to create all required connections). Create the connections between the Arduino board and the HM-10 module as shown in Figure 2.

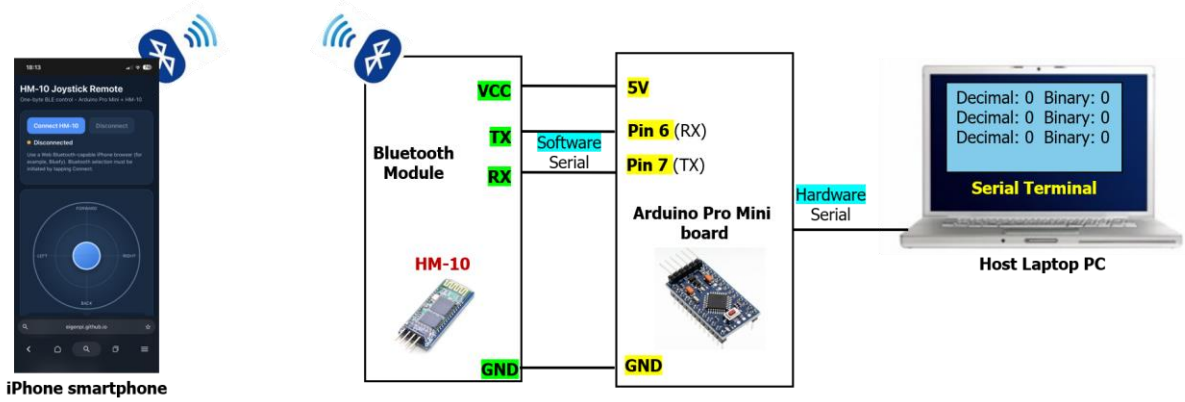


Figure 2: Experimental set-up for the example program.

Compile the sketch and program the Arduino board.
Open the serial monitor on your computer.
Launch the joystick remote controller described in the previous section on your iPhone.
Use the joystick to send commands.
Observe the serial monitor on your laptop computer to make sure/confirm those commands are received correctly!

5. Assignment

1)Follow through all the steps described in this lab and as demonstrated in class for the iPhone approach. If your smartphone is an Android phone, you can do the Android approach instead. Take clean/nice photos of your experimental setup to include in your report.

2) **Extending the iPhone Bluetooth Joystick Controller.** Because it is not the purpose of this lab/course to cover in-depth webpage development, you are allowed/encouraged to use AI tools for this assignment. Starting from the provided index.html that implements the joystick you must add additional features as described next.

Part 1: Modify the existing joystick webpage index.html to support additional commands, using two of the four currently unused upper bits of the command byte. The existing protocol uses only the least significant four bits to construct the command byte:
b7: 0
b6: 0
b5: 0
b4: 0
b3: B
b2: F
b1: R
b0: L

Where: B = backward, F = forward, R = right, L = left. The four upper bits are currently unused. You must modify the index.html to add two new buttons or switches to the webpage – placed **right underneath the joystick**. These buttons will be used to set bits b4 and b5 of the command byte:

Bit	Mask	New Function
b4	0x10	Turn headlights ON/OFF
b5	0x20	Sound horn

Part 2: Modify the given **test2_bt_module.ino** sketch to turn on/off the user LED on the Arduino Pro Mini when the received command byte includes "Turn headlights ON/OFF" and to blink 10 times/second the same LED if the received command byte includes "Sound horn". Verify the correct operation of the modified sketch with a simple hardware setup using the breadboard. Use only the Arduino Pro Mini connected via jumper wires and the HM10 BT module plugged into the breadboard. **Read the comments inside the provided sketch for connections between the Arduino and the BT module!**

Test your newly modified index.html – use those new buttons to send remote controls to the Arduino.

3)You must write a lab report. It should contain the following sections:

- Lab title, Your name
- **Summary** – a brief description of the activities you did in this lab.
- **Given Examples** – describe what examples you did. Make sure you take photos of your hardware setup and include in your report to show your work.
- **Extending the iPhone Bluetooth Joystick Controller** - describe what you did. Include photos to document work done.
- **Conclusion** – discuss your main takeaways from this lab and its assignment. Discuss any issues you encountered and how you solved them.
- **Appendix** – Include and fill to the best of your ability information in the following table, titled "Work Distribution and AI use" (the info in this table is hypothetical example):

Item	**Contribution of team member 1: Name**	**Contribution of team member 2: Name**	**AI use: Percentage and Details**
Given joystick example	50%	50%	0%, No AI use
Extension to add features	50%	50%	90%, We totally relied on AI to learn about HTML, CSS, and JavaScript and to implement required features.
Report writing	40%	60%	10% We used Grammarly to fix grammar and spelling errors. We wrote the complete initial draft and had AI improve its presentation.

Upload your report named **lab9_report_lastname.pdf** in a .zip file that shall include also your project(s) code that you created for the assignment, if any.

6. References and Credits

[1] https://www.youtube.com/watch?v=pj5hVGT7MBc
[2] https://eigenpi.github.io/HM-10-Joystick-Remote-for-iPhone/index.html