2.12: Introduction to Robotics
Lab 2:
Robot Assembly and Validation*

Spring 2023

Assigned on: 16th Feb 2023 Due by: 22nd Feb 2023

Instructions:

1. **We will be present to answer questions, and help you to debug any wiring issues. Please make sure to attend.**
2. **You will need to be able to successfully run the Arduino code in order to validate your system. There is nothing to submit for this lab.**

1 Introduction

This week, we'll be assembling the 2-link robot arm and wiring up the system (Figure 1). You will also run some provided Arduino code, to confirm that everything has been properly connected. The code that we'll be using for the class can be found in the Lab 2 GitHub: `https://github.com/mit212/lab2_2023`.

The lab tasks consist three parts:

1. Assemble the hardware
2. Validate the encoder connection
3. Validate the motor controller connection

2 Assemble Hardware

For instructions on how to assemble the hardware, please refer to the "Assembly Instructions.pdf" file under the Lab 2 folder of Canvas.

*

1. Version 1 - 2020: Rachel Hoffman
2. Version 2 - 2021: Phillip Daniel
3. Version 3 - 2023: Ravi Tejwani, Kentaro Barhydt

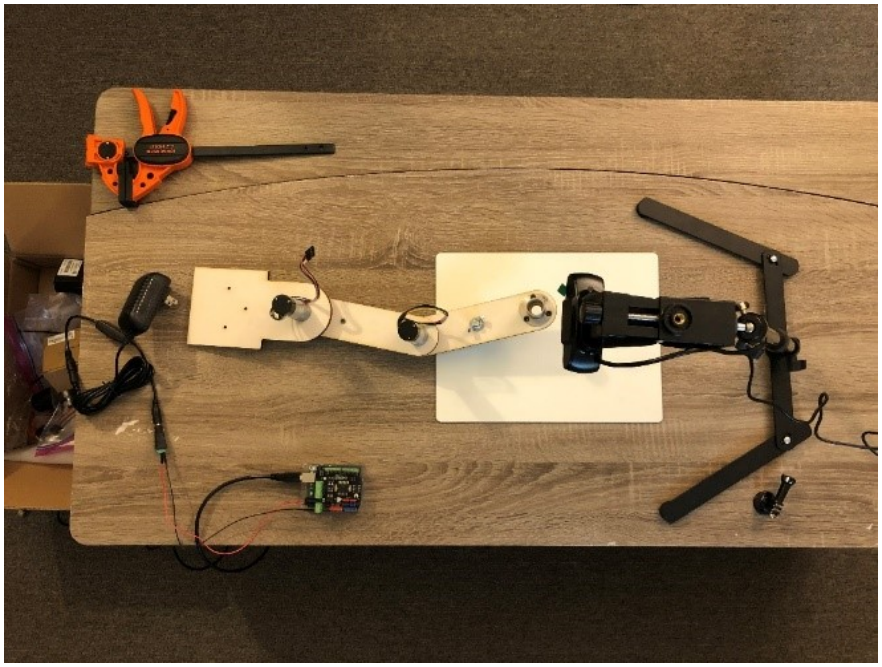

Figure 1: Overhead view of the assembled kit.

3 Validate Encoder

In order to validate the encoder, you will need to install the Arduino IDE onto your laptop:

3.1 Install the Arduino IDE:

1. Go to the Arduino website in the VM: `https://www.arduino.cc/en/software`. Download the Arduino IDE depending on your OS.

2. Install the Arduino IDE.

3. Connect the Arduino to your computer using the included USB cable

4. Download the diagnostic files from the Lab 2 GitHub:

```
cd ~                    # note: make sure we are at home folder
git clone https://github.com/mit212/lab2_2023.git
```

5. Open the Arduino IDE:

```
arduino
```

6. Add the encoder library to the Arduino installation by selecting: Tools>Manage Libraries. Search for "Encoder," and select it. If you can't find it through the search, then scroll down until you find the library labeled "Encoder" by Paul Stoffrefen. Install version 1.4.2 of the library by hovering over the title of the library, and clicking the button "Install." Now close the library management tool.

3.2 Run the validation scripts:

Use the Arduino IDE to open the file "encoderCheck". Refer to the website (`https://www.arduino.cc/reference/en/`) if you wish to learn more about the syntax of the code, although this is not required to complete the lab.

1. Ensure that the computer that will be connected to the Arduino via the USB is plugged into the wall. The computer must share the same ground as the motor controller board
2. Ensure that the motor controller is powered off. Do this by unplugging the power cable at the emergency stop.
3. Connect the Arduino to your computer using the provided USB cable.
4. Follow the instructions here (`https://www.arduino.cc/en/Guide/ArduinoUno`), starting at the line "**Select your board type and port**," in order to load the file 'encoderCheck.ino' onto the Arduino. The steps are summarized below.
   (a) You'll need to select the entry in the Tools > Board menu that corresponds to your Arduino board.
   (b) Select the serial device of the board from the Tools — Serial Port menu. To find out what device to select, you can disconnect your board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.
   (c) Now, simply click the "Upload" button in the environment. Wait a few seconds - you should see the RX and TX leds on the board flashing. If the upload is successful, the message "Done uploading." will appear in the status bar.
      i. If you see a "permission denied" error when uploading, for windows change the permissions by right clicking the file and give full permission. For linux and mac machines, run the below command in the terminal window of linux and then re-upload the script to your Arduino Uno:

```
sudo chmod a+rw /dev/ttyACM0
```

5. Open a serial port monitor via: **Tools>Serial Monitor**. Confirm that the tool reads "9600 baud" in the drop down menu.
6. Move the joints of the robot by hand. You should observe that counter clock-wise motion increases each encoder's count, while clockwise motion decreases them. A quarter turn should be 1176 counts. If you don't observe this, correct your setup. Ask the teaching staff for help if you get stuck. Note, you can reset the encoder count to zero by pressing the Arduino's "Reset" button. This button is next to the USB port on the Arduino.
7. Once you have confirmed proper operation of the encoders, close the file 'encoderCheck.ino'

4 Validate Motor

Once you have confirmed that the encoders operate correctly, you can confirm that the motors are properly connected. Use Arduino's GUI to open the file "motorCheck.ino"

To validate the motors' connections:

1. Ensure that the computer that is connected to the Arduino via the USB is plugged into the wall. The computer must share the same ground as the motor controller board
2. Be sure that the area in the robot's workspace is clear of obstacles.
3. Straighten each of the robot's link out, as shown in Figure 1. This is the "home position"
4. Load the file "motorCheck.ino" onto the Arduino.
5. Connect the motor controller board to power using the emergency stop cable that is provided. If your system begins to behave erratically, press the button to stop power to the motors.
   (a) Proper function will result in each joint moving back and fourth around its initial position. If you need to re-home the arm, you can do so by pressing the reset button on the Arduino