COMPSCI 503
Project 2: Duckie Town – Sensor/Motor Interface

Tiffany Liu, Rod Grupen
{tliu, grupen} @cs.umass.edu

October 1, 2019

DUE: Friday, October 11, 2019

1 Main Embedded Computing

For the remaining projects in this course, you will be using the Raspberry Pi 3 Model B+ and Arduino Uno as the on-board computer for the Duckie Town vehicle you will be building. The goal of this project is to familiarize and set-up the Pi-Arduino computing interface with all the peripherals necessary for Duckie Town.

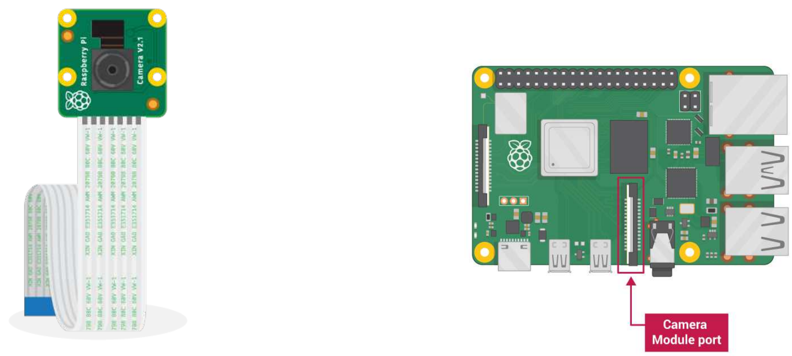
1.1 Raspberry Pi and Camera

This handout will provide you with most of the information you will need on the Raspberry Pi for this project. Should you find the need for more information, check out "The Official Raspberry Pi Beginner's Guide."

The Raspberry Pi comes equipped with ports that enable you to attach a monitor, keyboard, and mouse. While these will be useful for debugging issues should you need to, your primary mode of interacting with your Pi will be through remote ssh. We have provided your Pi with an 8GB micro SD card with the Raspbian OS all ready installed and set-up. **NOTE:** *The micro SD card is very susceptible to data corruption, so be sure that the Pi has been safely shut down before disconnecting it from power*—more on this later (Section 3.1).

The camera we will be using is the Raspberry Pi Camera v2 (Figure 1a). Locate the Camera Module port on your Pi (Figure 1b). While the Pi is **not powered**, gently pull up on the edges of the port's plastic clip. Insert the Camera Module ribbon cable into the port such that the metal tips of the ribbon cable are facing the metal parts in the port. Once you have ensured that the ribbon cable has been inserted properly, push the plastic clip back down—you may have to hold the ribbon cable in place while pushing down on the clip.

Find the 5V Pi power adapter in your kit and attach the micro USB to the corresponding port on the Pi. You will notice there is a Wide Input Shim attached to your Pi; this will



(a) Camera module. (b) Camera module port.

Figure 1: Raspberry Pi Camera v2.

be used in later projects to let you power your Pi with input voltages from 3V to 16V. If the Pi has been powered, you should see the red power LED light come on followed by the blinking green data LED light. If neither of these lights are on, toggle the switch on the 5V Pi power adapter cable. Once the green data LED light blinks at a steady rate, you should be able to connect to your Pi through remote ssh.

To ssh in to your Pi, you will need a laptop connected to the CICS network. If you do not have direct access to CICS, you will need to first ssh to the elnux.cs.umass.edu systems using your edlab account. If you do not have an edlab account, let us know, and we can get one set up for you. The username, hostname, and default password for ssh-ing are written and attached to the box your Pi came in.

Once you have successfully ssh-ed in to your Pi, determine a new team password and reset the password using the command: `sudo passwd`. You will be prompted to enter the current password before being prompted to enter the new password followed by a confirmation. After changing your password, you must report your new password to Tiffany.

Using your favorite text editor in your ssh session, open a new file and save it as `camera.py`. **NOTE:** it is important that you **never save the file as `picamera.py`**. Enter the following code in your file:

```
from picamera import PiCamera
from time import sleep

camera = PiCamera()

camera.start_preview()
sleep(5)
camera.capture('~/Desktop/image.jpg')
camera.stop_preview()

```

NOTE: it is important to `sleep` for at least 2 seconds before capturing an image because this gives the camera enough time to sense the light levels.

Run your `camera.py` program to take a team photo and include it in your project report. For more on working with the Pi camera, visit the Raspberry Pi project page on getting started with the Pi camera.

1.2 Raspberry Pi-Arduino Interface

As mentioned previously, your Duckie Town vehicle will use both the Raspberry Pi and the Arduino Uno as the on-board computer. To start, go to the Arduino guide home page and familiarize yourself with the basic care and feeding of Arduino-based applications. At this stage, your Arduino should receive power from the serial port.

IMPORTANT: In general, it is good practice to flash your Arduino with known working code before powering it to run. So to get into this practice, set up your Arduino to run the Analog Read Serial tutorial that is built-in to the IDE example code library.

Once you are comfortable working with the Arduino, follow this tutorial to set up your Pi and Arduino to send a message to the Pi, compute something, and send the proper reply back to the Pi.

2 Peripheral Sensor/Motor Frontend

For the last part of this project, your team will complete the interfaces to peripheral sensors and motors necessary for subsequent projects by running the following peripheral tasks using the Pi-Arduino interface you have set-up previously.

2.1 Ping Ultrasonic Range Finder

Read the entire tutorial on the Ping Ultrasonic Range Finder. Cable the device to your Arduino using 22 gauge solid core wire following the color convention presented in the tutorial. Install and run the demo code on your Arduino. **For your report, perform the following tasks**:

- Determine the speed of sound in the embedded systems lab in meters per second.
- Plot the range data from the Ping versus ground truth.
- Measure the sensitivity of the range measurement to misalignment of the reflecting surface.

2.2 QTR Reflectance Sensors

Read the application note on QTR reflectance sensors and the tutorial on quadrature encoders in their entirety. Cable two of your reflectance sensors and connect them to your Arduino as you have learned from the guide and tutorial to implement a quadrature encoder. Install and run the demo code on your Arduino. **For your report, perform the following tasks**:

- Explain how a quadrature encoder works.

- Design an experiment to test your encoder.

2.3 Motors and Pololu Dual MC33926 Motor Driver Shield

Read the entire user's guide on the Pololu Dual Motor Driver Shield and verify that the jumper connecting the power on the motor shield to the power on the Arduino is **NOT** installed. Before continuing, you must attend a tutorial from Tiffany on the procedure for charging the 3S 11.1V LiPo battery included in your team kit and give your battery a full charge (it is potentially **DANGEROUS** to charge these batteries incorrectly). Once you have a fully charged battery, build interface cables to connect both motors and power from the LiPo battery to the motor shield.

There is an Arduino library for the Pololu motor shield under the "Resources" tab in the Arduino IDE. Install and run on your Arduino the demo code that ramps up and down the speed of your motors. **For your report, perform the following tasks**:

- Determine the voltage your battery produces when fully charged.
- Explain why you must make sure that the jumper is **NOT** installed.
- Measure and report the resistances of your motors. Do they match the spec sheet?
- Determine the motor's torque constant, stall torque, and speed constant.

3 USEFUL RASPBERRY PI TIPS

3.1 Safely Powering Down Raspberry Pi

The green data LED light on the Pi indicates that the Pi is currently in the process of reading and/or writing data to the disk. If power is interrupted while data is being read or written, the disk will likely become corrupted. To avoid disk corruption, you are strongly advised to follow the protocol for properly shutting down and powering down the Pi. To shut down your Pi, in your ssh terminal, run `sudo shutdown 0`. This will kick you out of your ssh session, however, the final indication for whether or not your Pi has completed shut down is the green data LED light turning off. When the green data LED light is completely off, it is then safe to disconnect your Pi from power (when using the 5V Pi power adapter, first toggle the switch on the cable, then unplug the adapter from the power outlet; when using other power sources through the Wide Input Shim, simply disconnect the cable from the port on the shim)—doing so will then turn off the red power LED light.

3.2 Reclaiming Raspberry Pi Disk Space

Given that your Pi's are only equipped with an 8GB micro SD, **we recommend that you install libraries sparingly and as needed**. We have removed the Wolfram engine and all LibreOffice packages that come pre-installed on the OS, but you may find that you still need more disk space in the future. Here are a couple of useful commands for determining which packages to uninstall to gain disk space.

To display the disk usage, run

```
df -h
```

To list the installed packages by size, excluding required system packages, run

```
dpkg-query -Wf '${Installed-Size}\t${Package}\t${Priority}\n' | egrep '\s(
    optional|extra)' | cut -f 1,2 | sort -nr | less
```

Once you have determined which packages to uninstall, run the following commands to remove them:

```
sudo apt-get purge [unwanted package name]
sudo apt-get clean
sudo apt-get autoremove
```

3.3 Troubleshooting Raspberry Pi

For common problems with the Pi, visit the Embedded Linux Wiki page on R-PI Troubleshooting.