SANTA CLARA UNIVERSITY	Mechatronics 2024	Andy Wolfe

Lab #8 – Line Follower

I. Objectives

- Use the line-follower sensor to follow a straight or curved line.
- Determine how the geometry of your robot impacts this task.

II. Pre-Lab

Preparation:

- Study the documents related to the line follower: **QTRX-MD-08RC Reflectance Sensor Array (digital output).** **Make sure you have the right one!** A user's guide in the datasheet directory on google drive. A software library guide is there as well.
- The best way to install the library is to:
  - In the Arduino IDE, open the "Tools" menu, select "Manage Libraries...".
  - Search for "QTRSensors".
  - Click the QTRSensors entry in the list.
  - Click "Install".
- Determine how you will mount the sensors to your platform for this lab and obtain the necessary materials. The stock screws, nuts, and standoffs will be available in lab (#2 and #4 size) as well as some velcro.
- The QTRX-MD-08RC works best at a specific range of distances from the ground. Determine how far your sensor needs to be from your robot platform and where you will mount it.
- Design some test patterns – 1" wide black lines on white paper. Print them out or bring paper to lab and use black masking tape. You should be able to follow a curve with any radius 200mm or larger, a straight line, and up to a 45° angle.
- Print out or copy a ruler on paper that you can glue/tape to your robot. It should measure 100mm to the left and right of a center point.

- Review the line-follower library and your prior motor code. https://pololu.github.io/qtr-sensors-arduino/
  Remember to include a calibration process in your system. Also note that there are several routines provided to read the sensors. You can use any or all of them. These may be most useful.

```
void          read (uint16_t *sensorValues, QTRReadMode mode=QTRReadMode::On)
void          readCalibrated (uint16_t *sensorValues, QTRReadMode mode=QTRReadMode::On)
uint16_t      readLineBlack (uint16_t *sensorValues, QTRReadMode mode=QTRReadMode::On)
```

- Think about and discuss an appropriate control algorithm and how to calibrate the control parameters.

Pre-Lab Report:

- Create an accurate measured drawing that represents the position of each of the drive wheels and each of the line-follower photosensors on your robot. You can hand draw or use CAD. A top-projection is acceptable.
- Using these measurements, calculate how the orientation of your robot changes (i.e. the angle), when the right wheel moves forward 1cm and the left wheel simultaneously moves backwards 1cm.
- Using these measurements, calculate how the orientation of your robot changes (i.e. the angle), when the right wheel moves forward 1cm and the left wheel remains still.
- Using these measurements, calculate how the position of each of the 8 photosensors changes (i.e. ΔX and ΔY), when the right wheel moves forward 1cm and the left wheel simultaneously moves backwards 1cm.
- Using these measurements, calculate how the position of each of the 8 photosensors changes (i.e. ΔX and ΔY), when the right wheel moves forward 1cm and the left wheel remains still.
- Wish you had studied harder in Trigonometry.
- Turn in all of your calculations (with explanations) and your drawings.
- Include a selfie of your team's reflection from your planning meeting.

III. Lab Procedure

Line Follower

Task 1:

Line Sensor Operation:

The QTRX-MD-08RC uses an R/C discharge circuit to create a digital timing signal that corresponds to the phototransistor measurement. Luckily, someone has already written the low-level library to interface this device. *See* https://pololu.github.io/qtr-sensors-arduino/class_q_t_r_sensors.html. This library can be used to read and calibrate the sensors. Make sure you understand the calibration procedure from the documentation.

Mount the QTRX-MD-08RC in accordance with your preparation and connect it to the Arduino. Suitable cables are in the lab. Don't power anything unless you understand what you are doing, or you will break stuff. If you need to drill, see the instructor. If your QTRX-MD-08RC does not have connector pins – see the instructor.

Set up part of a 10-wide LED display **with proper current-limiting resistors**. Connect these LEDs to the Arduino such that you can turn 8 adjacent LEDs on and off individually.

Using the QTR library – your program must do the following:

1. Print a prompt on the serial port monitor to calibrate for black. When you press a button or type in something (your choice), perform the black-level calibration.
2. Print a prompt on the serial port monitor to calibrate for white. When you press a button or type in something (your choice), perform the white -level calibration.
3. Enter a loop where you read the 8 sensors using the previous calibration. Compare each reading to a threshold value that determines whether you are seeing black or white. On the 8 LEDs (corresponding to the 8 sensors in order), light the LED's to show white detection and leave the LEDs dark for black detection.

Get black masking tape and white paper from the instructors and make a line. Place the line under the sensor. As you move it left and right, the dark band indication on the LEDs should move as well. (Demo #1)

Include your code and a photo of the system in operation in your lab report.

Task 2:

Do any construction, coding, or circuit design necessary for this task.

1. Mount the line-follower sensor and wheels in the permanent position and orientation that you plan to use for your final project.
2. Develop and test line-following code using the test patterns you developed. You should be able to follow a curve with any radius 200mm or larger and an angle of up to 45°. Initially test with the wheels elevated by moving the black line back and forth and seeing if you get the proper response, then test on the ground/bench.
3. Demo the line follower following at least one non-straight test pattern you developed. (Demo #2)
4. Measure the speed at which your line follower operates (meters/s or feet/s). Include in your report.
5. Measure the maximum deviation from the line. Mark the center of your robot at the front with a pen and mount the paper ruler along the front edge. Use it to estimate the maximum deviation from the line. One good way to do this is to make a cell-phone video from right above, then view it frame by frame on your computer to find the maximum.
6. Lab report should include the measurements from steps 4 and 5 and working code.
7. Demo your line-following with no USB cable on the test courses I will supply. If your line follower works, it should follow my lines as well. (Demo #3)
8. The final project requires that you follow the actual line on the test field. You need to separately demo that for final project credit.