Matthew Russell, Joshua Ramayrat

Professor Wolfe

ELEN 123, Mechatronics

Laboratory 5: Line Follower and Accelerometer

**Task 1:**

It was asked to accomplish 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 calibration parameters. 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.*

The arduino program for the setup is as follows:

*#include <QTRSensors.h>*

*#include <TimerOne.h>*

*#define NUM\_SENSORS 8 //number of sensors used*

*#define NUM\_SAMPLES\_PER\_SENSOR 4 //average of ...*

*#define EMITTER\_PIN 2 //emitter is controlled by digital pin 2*

*volatile char calibrationColor;*

*int enA = 10; //Motor controller 1 pinouts connected to PWM.*

*int in1 = 9;*

*int in2 = 8;*

*int enB = 5; //Motor controller 2 pinouts connected to PWM.*

*int in3 = 7;*

*int in4 = 6;*

*volatile int count = 0, sw = 0, dist = 0; //distance attach interrupts*

*QTRSensorsRC qtrrc((unsigned char[]) {41, 42, 43, 44, 45, 46, 47, 48},*

*NUM\_SENSORS, NUM\_SAMPLES\_PER\_SENSOR, EMITTER\_PIN);*

*//sensors to 41-48 digital input*

*unsigned int sensorValues[NUM\_SENSORS];*

*void setup()*

*{*

*Serial.begin(9600);*

*pinMode(A0, INPUT\_PULLUP);*

*pinMode(A1, INPUT\_PULLUP);*

*pinMode(A2, INPUT\_PULLUP);*

*pinMode(A3, INPUT\_PULLUP);*

*pinMode(A4, INPUT\_PULLUP);*

*pinMode(A5, INPUT\_PULLUP);*

*pinMode(A6, INPUT\_PULLUP);*

*pinMode(A7, INPUT\_PULLUP);*

*pinMode(A8, INPUT\_PULLUP);*

*int i;*

*calibrationColor = Serial.read();*

*Serial.println("Click send to calibrate for black");*

*if (calibrationColor == '\n') //user inputs b, calibrate for black*

*{*

*Serial.println("Calibrating for black...");*

*for (int i = 0; i < 400; i++)*

*{*

*qtrrc.calibrate(); //calibrating the values of the sensor readings.*

*delay(20);*

*}*

*for (int i = 0; i < NUM\_SENSORS; i++)*

*{*

*Serial.print("Minimum calibrated value for sensor ");*

*Serial.print(i);*

*Serial.print(": ");*

*Serial.println(qtrrc.calibratedMinimumOn[i]);*

*Serial.println(' ');*

*}*

*}*

*calibrationColor = Serial.read();*

*Serial.println("Click send to calibrate for white");*

*if (calibrationColor == '\n') //user inputs w, calibrate for white*

*{*

*Serial.println("Calibrating for white...");*

*for (int i = 0; i < 400; i++)*

*{*

*qtrrc.calibrate(); //calibrating the values of the sensor readings.*

*delay(20);*

*}*

*for (int i = 0; i < NUM\_SENSORS; i++)*

*{*

*Serial.print("Maximum calibrated value for sensor ");*

*Serial.print(i);*

*Serial.print(": ");*

*Serial.println(qtrrc.calibratedMaximumOn[i]);*

*Serial.println(' ');*

*}*

*}*

*delay(100);*

*Serial.println("Beginning sensor readings...");*

*}*

*void loop(){*

*// read calibrated sensor values and obtain a measure of the line position from 0 to 5000*

*// To get raw sensor values, call:*

*// qtra.read(sensorValues); instead of unsigned int position = qtra.readLine(sensorValues);*

*unsigned int position = qtrrc.readLine(sensorValues);*

*// print the sensor values as numbers from 0 to 1000, where 0 means maximum reflectance and*

*// 1000 means minimum reflectance, followed by the line position*

*for (unsigned char i*

*#include <QTRSensors.h>*

*#include <TimerOne.h>*

*#define NUM\_SENSORS 8 //number of sensors used*

*#define NUM\_SAMPLES\_PER\_SENSOR 4 //average of ...*

*#define EMITTER\_PIN 2 //emitter is controlled by digital pin 2*

*volatile char calibrationColor;*

*int enA = 10; //Motor controller 1 pinouts connected to PWM.*

*int in1 = 9;*

*int in2 = 8;*

*int enB = 5; //Motor controller 2 pinouts connected to PWM.*

*int in3 = 7;*

*int in4 = 6;*

*int interruptPin1 = 11;*

*int interruptPin2 = 12;*

*volatile int count = 0, sw = 0, dist = 0; //distance attach interrupts*

*QTRSensorsRC qtrrc((unsigned char[]) {41, 42, 43, 44, 45, 46, 47, 48},*

*NUM\_SENSORS, NUM\_SAMPLES\_PER\_SENSOR, EMITTER\_PIN);*

*//sensors to 41-48 digital input*

*unsigned int sensorValues[NUM\_SENSORS];*

*void setup()*

*{*

*Serial.begin(9600);*

*pinMode(A0, INPUT\_PULLUP);*

*pinMode(A1, INPUT\_PULLUP);*

*pinMode(A2, INPUT\_PULLUP);*

*pinMode(A3, INPUT\_PULLUP);*

*pinMode(A4, INPUT\_PULLUP);*

*pinMode(A5, INPUT\_PULLUP);*

*pinMode(A6, INPUT\_PULLUP);*

*pinMode(A7, INPUT\_PULLUP);*

*pinMode(A8, INPUT\_PULLUP);*

*Timer1.initialize(250000);*

*Timer1.attachInterrupt(digitalPinToInterrupt(18), calibrationBlack, HIGH);*

*Timer1.attachInterrupt(digitalPinToInterrupt(19), calibrationWhite, HIGH);*

*int i;*

*void setup() {*

*// put your setup code here, to run once:*

*}*

*void loop() {*

*// put your main code here, to run repeatedly:*

*}*

*void calibrationWhite(){*

*Serial.println("Switch to 18 is HIGH, calibrating for white...");*

*{*

*Serial.println("Calibrating for white...");*

*for (int i = 0; i < 400; i++)*

*{*

*qtrrc.calibrate(); //calibrating the values of the sensor readings.*

*delay(20);*

*}*

*for (int i = 0; i < NUM\_SENSORS; i++)*

*{*

*Serial.print("Minimum calibrated value for sensor ");*

*Serial.print(i);*

*Serial.print(": ");*

*Serial.println(qtrrc.calibratedMinimumOff[i]);*

*Serial.println(' ');*

*}*

*}*

*}*

*}*

*void calibrationBlack(){*

*Serial.println("Switch to 18 is HIGH, calibrating for black...");*

*if (calibrationColor == 'b') //user inputs b, calibrate for black*

*{*

*Serial.println("Calibrating for black...");*

*for (int i = 0; i < 400; i++)*

*{*

*qtrrc.calibrate(); //calibrating the values of the sensor readings.*

*delay(20);*

*}*

*for (int i = 0; i < NUM\_SENSORS; i++)*

*{*

*Serial.print("Maximum calibrated value for sensor ");*

*Serial.print(i);*

*Serial.print(": ");*

*Serial.println(qtrrc.calibratedMaximumOn[i]);*

*Serial.println(' ');*

*}*

*}*

*}*

*}*

*= 0; i < NUM\_SENSORS; i++)*

*{*

*Serial.print(sensorValues[i]);*

*Serial.print('\t');*

*}*

*//Serial.println(); // uncomment this line if you are using raw values*

*//Serial.println(position); // comment this line out if you are using raw values*

*delay(100);*

*}*

**Task 2**

1. Mount the ADXL335 board onto your platform.

2. Connect the ADXL335 to your Arduino. Don’t power anything unless you understand what you

are doing or you will break stuff. If you need to drill, arrange to go over to the Maker lab.

3. Using your gravitational reference, determine the sensor readout for at least 2 known gravitation values for each of the 3 axes. Write your own program to do this (and turn it in with your report). Use this information to create a linear response model for the sensor in each of 3 dimensions.

Sample the sensor approximately every 10ms. Create a program that prints out the force on your sensor in 3 dimensions in units of G once when the program starts and again whenever the

readings change by more than approximately .05G. (Demo #2)

The calibration sketch for the ADXL335 is uploaded below

*const int xInput = A0;*

*const int yInput = A1;*

*const int zInput = A2;*

*const int buttonPin = 2;*

*// Raw Ranges:*

*// initialize to mid-range and allow calibration to*

*// find the minimum and maximum for each axis*

*int xRawMin = 512;*

*int xRawMax = 512;*

*int yRawMin = 512;*

*int yRawMax = 512;*

*int zRawMin = 512;*

*int zRawMax = 512;*

*// Take multiple samples to reduce noise*

*const int sampleSize = 10;*

*void setup()*

*{*

*analogReference(EXTERNAL);*

*Serial.begin(9600);*

*}*

*void loop()*

*{*

*int xRaw = ReadAxis(xInput);*

*int yRaw = ReadAxis(yInput);*

*int zRaw = ReadAxis(zInput);*

*if (digitalRead(buttonPin) == LOW)*

*{*

*AutoCalibrate(xRaw, yRaw, zRaw);*

*}*

*else*

*{*

*Serial.print("Raw Ranges: X: ");*

*Serial.print(xRawMin);*

*Serial.print("-");*

*Serial.print(xRawMax);*

*Serial.print(", Y: ");*

*Serial.print(yRawMin);*

*Serial.print("-");*

*Serial.print(yRawMax);*

*Serial.print(", Z: ");*

*Serial.print(zRawMin);*

*Serial.print("-");*

*Serial.print(zRawMax);*

*Serial.println();*

*Serial.print(xRaw);*

*Serial.print(", ");*

*Serial.print(yRaw);*

*Serial.print(", ");*

*Serial.print(zRaw);*

*// Convert raw values to 'milli-Gs"*

*long xScaled = map(xRaw, xRawMin, xRawMax, -1000, 1000);*

*long yScaled = map(yRaw, yRawMin, yRawMax, -1000, 1000);*

*long zScaled = map(zRaw, zRawMin, zRawMax, -1000, 1000);*

*// re-scale to fractional Gs*

*float xAccel = xScaled / 1000.0;*

*float yAccel = yScaled / 1000.0;*

*float zAccel = zScaled / 1000.0;*

*Serial.print(" :: ");*

*Serial.print(xAccel);*

*Serial.print("G, ");*

*Serial.print(yAccel);*

*Serial.print("G, ");*

*Serial.print(zAccel);*

*Serial.println("G");*

*delay(500);*

*}*

*}*

*//*

*// Read "sampleSize" samples and report the average*

*//*

*int ReadAxis(int axisPin)*

*{*

*long reading = 0;*

*analogRead(axisPin);*

*delay(1);*

*for (int i = 0; i < sampleSize; i++)*

*{*

*reading += analogRead(axisPin);*

*}*

*return reading/sampleSize;*

*}*

*//*

*// Find the extreme raw readings from each axis*

*//*

*void AutoCalibrate(int xRaw, int yRaw, int zRaw)*

*{*

*Serial.println("Calibrate");*

*if (xRaw < xRawMin)*

*{*

*xRawMin = xRaw;*

*}*

*if (xRaw > xRawMax)*

*{*

*xRawMax = xRaw;*

*}*

*if (yRaw < yRawMin)*

*{*

*yRawMin = yRaw;*

*}*

*if (yRaw > yRawMax)*

*{*

*yRawMax = yRaw;*

*}*

*if (zRaw < zRawMin)*

*{*

*zRawMin = zRaw;*

*}*

*if (zRaw > zRawMax)*

*{*

*zRawMax = zRaw;*

*}*

*}*

4. Now connect up your motors so that you can drive forward and stop. 5. Set up a 10-wide LED display with proper current-limiting resistors. Connect at least 8 of these

LEDs to the Arduino such that you can turn them on and off individually.

6. I have boards that are 36” long, 16” wide, and approximately ¾” thick. Starting from the middle of the board, you need to drive to the end (approximately 18”) and detect when you fall off the

edge. Once you fall off the edge, you need to stop and flash all 8 lights 3 times. (I only have 6 boards so you may need to do some testing without the boards first.) Document for your repo

**Task 3**

**Task 1**

**#include <QTRSensors.h>**

**#include <TimerOne.h>**

**#define NUM\_SENSORS 8 //number of sensors used**

**#define NUM\_SAMPLES\_PER\_SENSOR 5 //average of ...**

**#define EMITTER\_PIN 2 //emitter is controlled by digital pin 2**

**volatile char calibrationColor;**

**int enA = 10; //Motor controller 1 pinouts connected to PWM.**

**int in1 = 9;**

**int in2 = 8;**

**int enB = 5; //Motor controller 2 pinouts connected to PWM.**

**int in3 = 7;**

**int in4 = 6;**

**int interruptPin1 = 11;**

**int interruptPin2 = 12;**

**volatile int count = 0, sw = 0, dist = 0; //distance attach interrupts**

**QTRSensorsRC qtrrc((unsigned char[]) {41, 42, 43, 44, 45, 46, 47, 48}, 8);**

**//NUM\_SENSORS, NUM\_SAMPLES\_PER\_SENSOR, EMITTER\_PIN);**

**//sensors to 41-48 digital input**

**unsigned int sensorValues[NUM\_SENSORS];**

**void calibrationBlack(){**

**Serial.println("Switch to 18 is HIGH, calibrating for black...");**

**for (int i = 0; i < 400; i++)**

**{**

**qtrrc.calibrate(); //calibrating the values of the sensor readings.**

**delay(20);**

**}**

**for (int i = 0; i < NUM\_SENSORS; i++)**

**{**

**Serial.print("Minimum calibrated black value for sensor ");**

**Serial.print(i);**

**Serial.print(": ");**

**Serial.println(qtrrc.calibratedMinimumOn[i]);**

**Serial.println(' ');**

**}**

**for (int i = 0; i < NUM\_SENSORS; i++)**

**{**

**Serial.print("Maximum calibrated black value for sensor ");**

**Serial.print(i);**

**Serial.print(": ");**

**Serial.println(qtrrc.calibratedMaximumOn[i]);**

**Serial.println(' ');**

**}**

**Serial.println("Done calibrating.");**

**}**

**void calibrationWhite(){**

**Serial.println("Switch to 19 is HIGH, calibrating for white...");**

**for (int i = 0; i < 400; i++)**

**{**

**qtrrc.calibrate(); //calibrating the values of the sensor readings.**

**delay(20);**

**}**

**for (int i = 0; i < NUM\_SENSORS; i++)**

**{**

**Serial.print("Minimum calibrated white value for sensor ");**

**Serial.print(i);**

**Serial.print(": ");**

**Serial.println(qtrrc.calibratedMinimumOn[i]);**

**Serial.println(' ');**

**}**

**for (int i = 0; i < NUM\_SENSORS; i++)**

**{**

**Serial.print("Maximum calibrated white value for sensor ");**

**Serial.print(i);**

**Serial.print(": ");**

**Serial.println(qtrrc.calibratedMaximumOn[i]);**

**Serial.println(' ');**

**}**

**Serial.println("Done Calibrating");**

**}**

**void setup()**

**{**

**Serial.begin(9600);**

**pinMode(A0, INPUT\_PULLUP);**

**pinMode(A1, INPUT\_PULLUP);**

**pinMode(A2, INPUT\_PULLUP);**

**pinMode(A3, INPUT\_PULLUP);**

**pinMode(A4, INPUT\_PULLUP);**

**pinMode(A5, INPUT\_PULLUP);**

**pinMode(A6, INPUT\_PULLUP);**

**pinMode(A7, INPUT\_PULLUP);**

**pinMode(A8, INPUT\_PULLUP);**

**Timer1.initialize(250000);**

**attachInterrupt(digitalPinToInterrupt(18), calibrationBlack, HIGH);**

**attachInterrupt(digitalPinToInterrupt(19), calibrationWhite, HIGH);**

**}**

**void loop() {**

**// put your main code here, to run repeatedly:**

**}**