//Lab 5 Demo 1

//Matt Russell, Joshua Ramayrat

//Calibrates reflectance sensor for white and black surroundings

#include <QTRSensors.h>

//must be above TIMEOUT

#define MINGUESS 5000

QTRSensorsRC qtrrc((unsigned char[]) {

41, 42, 43, 44, 45, 46, 47, 48

}, 8);

unsigned int maxVal[] = {0, 0, 0, 0, 0, 0, 0, 0};

unsigned int minVal[] = {MINGUESS, MINGUESS, MINGUESS, MINGUESS, MINGUESS, MINGUESS, MINGUESS, MINGUESS};

unsigned int sensorValues[8];

unsigned int sensor1[100];

unsigned int sensor2[100];

unsigned int sensor3[100];

unsigned int sensor4[100];

unsigned int sensor5[100];

unsigned int sensor6[100];

unsigned int sensor7[100];

unsigned int sensor8[100];

volatile double avg1 = 0.0, avg2 = 0.0, avg3 = 0.0, avg4 = 0.0, avg5 = 0.0, avg6 = 0.0, avg7 = 0.0, avg8 = 0.0;

void setup() {

Serial.begin(9600);

pinMode(41, INPUT);

pinMode(42, INPUT);

pinMode(43, INPUT);

pinMode(44, INPUT);

pinMode(45, INPUT);

pinMode(46, INPUT);

pinMode(47, INPUT);

pinMode(48, INPUT);

pinMode(22, OUTPUT);

attachInterrupt(digitalPinToInterrupt(18), whitecalibration, RISING);

attachInterrupt(digitalPinToInterrupt(19), blackcalibration, RISING);

}

void loop() {

}

//calibrates sensor for white surroundings

void whitecalibration() {

delay(500);

Serial.println("Calibration for white started, finding Max, Min, and Mean");

digitalWrite(22, HIGH);

//Find Max

for (int j = 0; j < 100; j++) {

qtrrc.read(sensorValues);

//Store raw data

sensor1[j] = sensorValues[0];

sensor2[j] = sensorValues[1];

sensor3[j] = sensorValues[2];

sensor4[j] = sensorValues[3];

sensor5[j] = sensorValues[4];

sensor6[j] = sensorValues[5];

sensor7[j] = sensorValues[6];

sensor8[j] = sensorValues[7];

}

//Data processing: 4000 reading is noise in white case, give it nonsensical value

for (int e = 0; e < 100; e++) {

if (sensor1[e] == 4000) {

sensor1[e] = 5000;

}

if (sensor2[e] == 4000) {

sensor2[e] = 5000;

}

if (sensor3[e] == 4000) {

sensor3[e] = 5000;

}

if (sensor4[e] == 4000) {

sensor4[e] = 5000;

}

if (sensor5[e] == 4000) {

sensor5[e] = 5000;

}

if (sensor6[e] == 4000) {

sensor6[e] = 5000;

}

if (sensor7[e] == 4000) {

sensor7[e] = 5000;

}

if (sensor8[e] == 4000) {

sensor8[e] = 5000;

}

if (sensor1[e] != 5000) {

avg1 = avg1 + sensor1[e];

}

if (sensor2[e] != 5000) {

avg2 = avg2 + sensor2[e];

}

if (sensor3[e] != 5000) {

avg3 = avg3 + sensor3[e];

}

if (sensor4[e] != 5000) {

avg4 = avg4 + sensor4[e];

}

if (sensor5[e] != 5000) {

avg5 = avg5 + sensor5[e];

}

if (sensor6[e] != 5000) {

avg6 = avg6 + sensor6[e];

}

if (sensor7[e] != 5000) {

avg7 = avg7 + sensor7[e];

}

if (sensor8[e] != 5000) {

avg8 = avg8 + sensor8[e];

}

}

avg1 = avg1 / 100;

avg2 = avg2 / 100;

avg3 = avg3 / 100;

avg4 = avg4 / 100;

avg5 = avg5 / 100;

avg6 = avg6 / 100;

avg7 = avg7 / 100;

avg8 = avg8 / 100;

Serial.println("Averages: ");

Serial.print(avg1);

Serial.print('\t');

Serial.print(avg2);

Serial.print('\t');

Serial.print(avg3);

Serial.print('\t');

Serial.print(avg4);

Serial.print('\t');

Serial.print(avg5);

Serial.print('\t');

Serial.print(avg6);

Serial.print('\t');

Serial.print(avg7);

Serial.print('\t');

Serial.println(avg8);

for (int t = 0; t < 100; t++) {

if (maxVal[0] < sensor1[t] && sensor1[t] != 5000) {

maxVal[0] = sensor1[t];

}

if (maxVal[1] < sensor2[t] && sensor2[t] != 5000) {

maxVal[1] = sensor2[t];

}

if (maxVal[2] < sensor3[t] && sensor3[t] != 5000) {

maxVal[2] = sensor3[t];

}

if (maxVal[3] < sensor4[t] && sensor4[t] != 5000) {

maxVal[3] = sensor4[t];

}

if (maxVal[4] < sensor5[t] && sensor5[t] != 5000) {

maxVal[4] = sensor5[t];

}

if (maxVal[5] < sensor6[t] && sensor6[t] != 5000) {

maxVal[5] = sensor6[t];

}

if (maxVal[6] < sensor7[t] && sensor7[t] != 5000) {

maxVal[6] = sensor7[t];

}

if (maxVal[7] < sensor8[t] && sensor8[t] != 5000) {

maxVal[7] = sensor8[t];

}

if (minVal[0] > sensor1[t]) {

minVal[0] = sensor1[t];

}

if (minVal[1] > sensor2[t]) {

minVal[1] = sensor2[t];

}

if (minVal[2] > sensor3[t]) {

minVal[2] = sensor3[t];

}

if (minVal[3] > sensor4[t]) {

minVal[3] = sensor4[t];

}

if (minVal[4] > sensor5[t]) {

minVal[4] = sensor5[t];

}

if (minVal[5] > sensor6[t]) {

minVal[5] = sensor6[t];

}

if (minVal[6] > sensor7[t]) {

minVal[6] = sensor7[t];

}

if (minVal[7] > sensor8[t]) {

minVal[7] = sensor8[t];

}

}

Serial.println("Maximum values");

for (int k = 0; k < 8; k++) {

Serial.print(maxVal[k]);

Serial.print('\t');

}

Serial.println();

Serial.println("Minimum values");

for (int l = 0; l < 8; l++) {

Serial.print(minVal[l]);

Serial.print('\t');

}

}

void blackcalibration() {

delay(500);

Serial.println("Calibration for black started, finding Max, Min, and Mean");

digitalWrite(22, HIGH);

//Find Max

for (int j = 0; j < 100; j++) {

qtrrc.read(sensorValues);

//Store raw data

sensor1[j] = sensorValues[0];

sensor2[j] = sensorValues[1];

sensor3[j] = sensorValues[2];

sensor4[j] = sensorValues[3];

sensor5[j] = sensorValues[4];

sensor6[j] = sensorValues[5];

sensor7[j] = sensorValues[6];

sensor8[j] = sensorValues[7];

}

//Since 4000 is sensical value for black line we don't need to perform any data processing

for (int r = 0; r < 100; r++) {

avg1 = avg1 + sensor1[r];

avg2 = avg2 + sensor2[r];

avg3 = avg3 + sensor3[r];

avg4 = avg4 + sensor4[r];

avg5 = avg5 + sensor5[r];

avg6 = avg6 + sensor6[r];

avg7 = avg7 + sensor7[r];

avg8 = avg8 + sensor8[r];

}

avg1 = avg1 / 100;

avg2 = avg2 / 100;

avg3 = avg3 / 100;

avg4 = avg4 / 100;

avg5 = avg5 / 100;

avg6 = avg6 / 100;

avg7 = avg7 / 100;

avg8 = avg8 / 100;

Serial.println("Averages: ");

Serial.print(avg1);

Serial.print('\t');

Serial.print(avg2);

Serial.print('\t');

Serial.print(avg3);

Serial.print('\t');

Serial.print(avg4);

Serial.print('\t');

Serial.print(avg5);

Serial.print('\t');

Serial.print(avg6);

Serial.print('\t');

Serial.print(avg7);

Serial.print('\t');

Serial.println(avg8);

for (int t = 0; t < 100; t++) {

if (maxVal[0] < sensor1[t]) {

maxVal[0] = sensor1[t];

}

if (maxVal[1] < sensor2[t]) {

maxVal[1] = sensor2[t];

}

if (maxVal[2] < sensor3[t]) {

maxVal[2] = sensor3[t];

}

if (maxVal[3] < sensor4[t]) {

maxVal[3] = sensor4[t];

}

if (maxVal[4] < sensor5[t]) {

maxVal[4] = sensor5[t];

}

if (maxVal[5] < sensor6[t]) {

maxVal[5] = sensor6[t];

}

if (maxVal[6] < sensor7[t]) {

maxVal[6] = sensor7[t];

}

if (maxVal[7] < sensor8[t]) {

maxVal[7] = sensor8[t];

}

if (minVal[0] > sensor1[t]) {

minVal[0] = sensor1[t];

}

if (minVal[1] > sensor2[t]) {

minVal[1] = sensor2[t];

}

if (minVal[2] > sensor3[t]) {

minVal[2] = sensor3[t];

}

if (minVal[3] > sensor4[t]) {

minVal[3] = sensor4[t];

}

if (minVal[4] > sensor5[t]) {

minVal[4] = sensor5[t];

}

if (minVal[5] > sensor6[t]) {

minVal[5] = sensor6[t];

}

if (minVal[6] > sensor7[t]) {

minVal[6] = sensor7[t];

}

if (minVal[7] > sensor8[t]) {

minVal[7] = sensor8[t];

}

}

Serial.println("Maximum values");

for (int k = 0; k < 8; k++) {

Serial.print(maxVal[k]);

Serial.print('\t');

}

Serial.println();

Serial.println("Minimum values");

for (int l = 0; l < 8; l++) {

Serial.print(minVal[l]);

Serial.print('\t');

}

}