ECGR 4161/5196 LAB 06 REPORT

GROUP: 29

NAMES: Ajay Chundi, and Anyaegbu Somtochukwu C.

VIDEO LINK:

https://drive.google.com/drive/folders/1_VY_7_gvVfYqvrWaiej_D4SC0oum6WCW?usp=sharing

https://drive.google.com/file/d/1l3x5d7X849k9CuIMYevAdGXowVPSdwYm/view?usp=sharing

OBJECTIVE: The main objective of this lab is to implement and demonstrate (using video) a line-following program on the TI-RSLK robot. The robot drives forward following a line and stops when it gets to a perpendicular line.

COMMENTARY:

- **Introduction**

  This is a one-part lab which is aimed at enabling students gain practical understanding of how IR LED/phototransistor works. Also, interfaced with the TI-RSLK robot the sensor (QTR-8A) is used to determine the robot's drive path.

  The robot's drive path is 20cm straight line (1.9cm wide) followed by a 45-degree curve and another 20cm of straight line. The drive path's end is indicated using a 10cm perpendicular line. This was implemented using 3 sheets of 8.5" by 11" paper. The robot should drive through the path without a variance of more than 2cm from center of the line.

- **Materials Required**
  - MSP432P401R MCU
  - TI-RSLK Chassis Board v1.0
  - QTR-8A Reflectance sensor array with 8 IR LED/phototransistor
  - Romi Black chassis kit
  - Energia 1.8.11E23
- **Results**

  One problem encountered had to do with the robot not being able to stop at the end of the track at the thick finish line. This issue was fixed by writing more code to include a Boolean variable and an "if" statement to decide when the motors would stop. At the end of the experiment, the desired outcome was achieved, and the robot properly traversed the line on the course, and came to a halt at the end, just as required. This lab was successful.

CODE:

```
/*
 * Energia Robot Library for Texas Instruments' Robot System Learning Kit (RSLK)
 * Line Following Example
 *
 * Summary:
 * This example has the TI Robotic System Learning Kit (TI RSLK) follow a line
```

```
 * using a basic line following algorithm. This example works on a dark floor with
 * a white line or a light floor with a dark line. The robot first needs to be calibrated
 * Then place the robot on the hit the left button again to begin the line following.
 *
 * How to run:
 * 1) Push left button on Launchpad to have the robot perform calibration.
 * 2) Robot will drive forwards and backwards by a predefined distance.
 * 3) Place the robot center on the line you want it to follow.
 * 4) Push left button again to have the robot begin to follow the line.
 *
 * Parts Info:
 * o Black eletrical tape or white electrical tape. Masking tape does not work well
 *   with IR sensors.
 *
 * Learn more about the classes, variables and functions used in this library by going
to:
 * https://fcooper.github.io/Robot-Library/
 *
 * Learn more about the TI RSLK by going to http://www.ti.com/rslk
 *
 * created by Franklin Cooper Jr.
 * Modified by Somto Anyaegbu & Ajay Chundi
 *
 * This example code is in the public domain.
 */

#include "SimpleRSLK.h"

uint16_t sensorVal[LS_NUM_SENSORS];
uint16_t sensorCalVal[LS_NUM_SENSORS];
uint16_t sensorMaxVal[LS_NUM_SENSORS];
uint16_t sensorMinVal[LS_NUM_SENSORS];

void setup(){
  Serial.begin(115200);

  setupRSLK();
  /* Left button on Launchpad */
  setupWaitBtn(LP_LEFT_BTN);
  /* Red led in rgb led */
  setupLed(RED_LED);
  clearMinMax(sensorMinVal,sensorMaxVal);
}

void floorCalibration() {

  /* Place Robot On Floor (no line) */
  delay(2000);
  String btnMsg = "Push left button on Launchpad to begin calibration.\n";
  btnMsg += "Make sure the robot is on the floor away from the line.\n";
  /* Wait until button is pressed to start robot */
  waitBtnPressed(LP_LEFT_BTN,btnMsg,RED_LED);

  delay(1000);

  Serial.println("Running calibration on floor");
  simpleCalibrate();
  Serial.println("Reading floor values complete");

  btnMsg = "Push left button on Launchpad to begin line following.\n";
```

```
  btnMsg += "Make sure the robot is on the line.\n";
  /* Wait until button is pressed to start robot */
  waitBtnPressed(LP_LEFT_BTN,btnMsg,RED_LED);
  delay(1000);

  enableMotor(BOTH_MOTORS);
}

void simpleCalibrate() {
  /* Set both motors direction forward */
  setMotorDirection(BOTH_MOTORS,MOTOR_DIR_FORWARD);
  /* Enable both motors */
  enableMotor(BOTH_MOTORS);
  /* Set both motors speed 20 */
  setMotorSpeed(BOTH_MOTORS,20);

  for(int x = 0;x<100;x++){
    readLineSensor(sensorVal);
    setSensorMinMax(sensorVal,sensorMinVal,sensorMaxVal);
  }

  /* Disable both motors */
  disableMotor(BOTH_MOTORS);
}

bool isCalibrationComplete = false;
void loop() {

  static bool done = false; //boolean variable to determine when motors stop

  // Wait for button press after a run so we can run again
  if (done) {
    disableMotor(BOTH_MOTORS);
    waitBtnPressed(LP_LEFT_BTN,"Waiting for Button Press",RED_LED);
    delay(2000);
    enableMotor(BOTH_MOTORS);
    done = false;
  }



  //Count number of sensors reading value to determine
  //end of line.
  //Line present if sensor value > 750
  int sensorCount = 0;
  for (int i = 0; i < LS_NUM_SENSORS; ++i) {
    if (sensorCalVal[i] > 750) {
      ++sensorCount;
    }
  }



  uint16_t normalSpeed = 10;
  uint16_t fastSpeed = 20;

  /* Valid values are either:
   *  DARK_LINE  if your floor is lighter than your line
   *  LIGHT_LINE if your floor is darker than your line
   */
  uint8_t lineColor = DARK_LINE;
```

```
  /* Run this setup only once */
  if(isCalibrationComplete == false) {
    floorCalibration();
    isCalibrationComplete = true;
  }

  readLineSensor(sensorVal);
  readCalLineSensor(sensorVal,sensorCalVal,sensorMinVal,sensorMaxVal,lineColor);

  uint32_t linePos = getLinePosition(sensorCalVal,lineColor);
        delay(10);  Serial.println(linePos);

  if(linePos > 0 && linePos < 3000) {
    setMotorSpeed(LEFT_MOTOR,normalSpeed);
    setMotorSpeed(RIGHT_MOTOR,fastSpeed);
  } else if(linePos > 3500) {
    setMotorSpeed(LEFT_MOTOR,fastSpeed);
    setMotorSpeed(RIGHT_MOTOR,normalSpeed);
  } else {
    setMotorSpeed(LEFT_MOTOR,normalSpeed);
    setMotorSpeed(RIGHT_MOTOR,normalSpeed);
  }

  //If the robot derails off the line
  //Or reaches end, motors are disabled
  //number of sensors reading values >= 5 used
  //to determine end of line
  //linePos == 0 used to monitor when robot derails off course
   if (sensorCount >= 5 || linePos == 0) {
    Serial.println("End");
    done = true;
    return;
   }

}
```