/\*

 \* Coded by: Hunter Burnett & Jaskin Kabir

 \* Date: 02/15/2024

 \* Class: ECGR 4161 | Intro to Robotics Spring 2024

 \* Lab03

 \* This program will first test the robot's ability to drive straight 1.5 meters

 \* with no motor ramping with the press of the left button

 \* Another press of the left button will make the robot drive straight for 3

 \* meters with the motors ramping up their speed

 \* This is the combined code from part A and B for lab 3 ECGR 4161

 \*

\*/

// This code includes various libraries and defines constants and variables used in controlling a robot.

#include <Bump\_Switch.h>

#include <Encoder.h>

#include <GP2Y0A21\_Sensor.h>

#include <QTRSensors.h>

#include <Romi\_Motor\_Power.h>

#include <RSLK\_Pins.h>

#include <SimpleRSLK.h>

// Constants defining time intervals and speeds for the robot's motors.

#define MOTOR\_DELAY           11132 // Motor delay changed to

                                    // 10000 ms to travel 1.5 meters

#define STOP\_MOTOR\_DELAY      1000

#define MOTOR\_DELAY1          5700 // <--- Adjust delay

#define MOTOR\_DELAY2          750  // <--- Adjust delay

bool started = false;

// Default PWM signals (percentage-% of power 0-100) for both RSLK motors.

int DEFAULT\_LEFT\_SPEED = 2;   // <--- Adjust main speed

int DEFAULT\_RIGHT\_SPEED = 3;  // <--- Adjust main speed

// Variables to store initial motor speeds for reset.

int RESET\_LEFT\_SPEED = DEFAULT\_LEFT\_SPEED;

int RESET\_RIGHT\_SPEED = DEFAULT\_RIGHT\_SPEED;

// Default PWM signals (percentage-% of power 0-100) for both RSLK motors.

int LEFT\_MOTOR\_SPEED = 16;     // Speed percentage

int RIGHT\_MOTOR\_SPEED = 17;    // Speed percentage

void setup() {

  // Initialize serial communication and setup various components of the robot.

  Serial.begin(9600);

  setupRSLK();

  setupWaitBtn(LP\_LEFT\_BTN);

  setupLed(RED\_LED);

  setupLed(GREEN\_LED);

}

void loop() {

  // Main loop of the program.

  startProgram();

  // Move the robot forward, then stop.

  moveForward();

  stopMotors();

  startProgram();

  // Ramp up motor speeds, move forward at a constant speed, then ramp

  // down motor speeds and stop.

  rampUpMotors();

  moveForward2();

  blinkLED(100); //Blink indicating ramp up stopping

  delay(MOTOR\_DELAY1-200);

  blinkLED(100); //Blink indicating ramp down starting

  rampDownMotors();

  stopMotors();

  resetMotorSpeeds();

}

// Function to ramp up motor speeds over a period of time.

void rampUpMotors(){

  for(int i = 0; i < 10; i++){

    DEFAULT\_LEFT\_SPEED += 2;

    DEFAULT\_RIGHT\_SPEED += 2;

    moveForward2();

    delay(MOTOR\_DELAY2);

  }

}

// Function to ramp down motor speeds over a period of time.

void rampDownMotors(){

  for(int i = 10; i > 0; i--){

    DEFAULT\_LEFT\_SPEED -= 2;

    DEFAULT\_RIGHT\_SPEED -= 2;

    moveForward2();

    delay(MOTOR\_DELAY2);

  }

}

// Function to reset motor speeds back to default values.

void resetMotorSpeeds(){

  DEFAULT\_LEFT\_SPEED = RESET\_LEFT\_SPEED;

  DEFAULT\_RIGHT\_SPEED = RESET\_RIGHT\_SPEED;

}

// Function to move the robot forward for a specified time and then stop.

void moveForward(){

  enableMotor(BOTH\_MOTORS);

  setMotorDirection(BOTH\_MOTORS,MOTOR\_DIR\_FORWARD);

  setMotorSpeed(LEFT\_MOTOR,LEFT\_MOTOR\_SPEED);

  setMotorSpeed(RIGHT\_MOTOR,RIGHT\_MOTOR\_SPEED);

  delay(MOTOR\_DELAY);

}

// Function to move the robot forward at set motor speeds.

void moveForward2(){

  enableMotor(BOTH\_MOTORS);

  setMotorDirection(BOTH\_MOTORS,MOTOR\_DIR\_FORWARD);

  setMotorSpeed(LEFT\_MOTOR,DEFAULT\_LEFT\_SPEED);

  setMotorSpeed(RIGHT\_MOTOR,DEFAULT\_RIGHT\_SPEED);

}

// Function to stop the motors.

void stopMotors(){

  disableMotor(BOTH\_MOTORS);

  delay(STOP\_MOTOR\_DELAY);

}

// Function to start the program upon pressing a button.

void startProgram(){

  String btnMsg = "Push left button on Launchpad to start lab program.\n";

  waitBtnPressed(LP\_LEFT\_BTN,btnMsg,RED\_LED);

  started = true;

  blinkLED(1000);

  blinkLED(1000);

}

// Function to blink an LED for a specified period.

int blinkLED(int period){

  int pause = period/2;

  digitalWrite(GREEN\_LED, HIGH);

  delay(pause);

  digitalWrite(GREEN\_LED, LOW);

  delay(pause);

}