#include <FEHLCD.h>

#include <FEHIO.h>

#include <FEHUtility.h>

#include <FEHServo.h>

#include <FEHMotor.h>

void straight(float time);

void start();

void leverSwitch();

void stopMotors();

void back(float time);

void left(float time);

void right(float time);

void read();

int lightTest();

void slowStraight(float time);

void slowRight(float time);

void slowLeft(float time);

void backLeft(float time);

void backRight(float time);

void toCoin(float time);

void lineFollower(float color,float time);

void dropCoin();

void toRamp();

//declare motors, servos, microswitches, and cdsCell

//front motor in motor port 0

FEHMotor rightMotor(FEHMotor::Motor0, 7.2);

//back motor in motor port 1

FEHMotor leftMotor(FEHMotor::Motor1, 7.2);

//arm servo in servo port 0

FEHServo armServo(FEHServo::Servo2);

//cds cell in pin P1\_7

AnalogInputPin cdsCell(FEHIO::P1\_7);

//sets servo for wheel orientation

FEHServo rotServo(FEHServo::Servo0);

//sets microservo for arm

FEHServo microServo(FEHServo::Servo4);//Need to find min and max for microServo

//set bump switches for front of bot and on top of arm

DigitalInputPin rightBump(FEHIO::P0\_0);

DigitalInputPin leftBump(FEHIO::P0\_2);

DigitalInputPin armBump(FEHIO::P0\_7);

DigitalInputPin frontBump(FEHIO::P0\_4);

//Set inputs for line readers

AnalogInputPin rightOpto(FEHIO::P1\_0);

AnalogInputPin midOpto(FEHIO::P1\_2);

AnalogInputPin leftOpto(FEHIO::P1\_4);

#define LINE\_ON\_RIGHT 0

#define ON\_LINE 1

#define LINE\_ON\_LEFT 2

int main(void)

{

LCD.Clear();

//read();

//set mins and maxes for the all servos

armServo.SetMin(1200);

armServo.SetMax(2200);

rotServo.SetMin(700);

rotServo.SetMax(2350);

microServo.SetMin(1470);

microServo.SetMax(2270);

armServo.SetDegree(0);

microServo.SetDegree(50);

rotServo.SetDegree(124);

//check to see if light is on and start moving once detected

start();

float time = 2.0;

slowStraight(time);

time = 0.5;

slowLeft(time);

////function that uses line following to line itself up with the front of the coin slot

toCoin(time);

////uses microservo to drop coin into slot-may have arm servo rotate as well

dropCoin();

//should line robot up directly with the ramp

toRamp();

//This is the same code we used from PT1, could need to be updated

straight(time);

time = 0.1;

right(time);

time = 0.75;

straight(time);

time = 0.75;

leverSwitch();

}

//function to stop motors

void stopMotors(){

rightMotor.Stop();

leftMotor.Stop();

}

//function to move to lever

void straight( float time)

{

float rightMotorPercent=70.0;

float leftMotorPercent=-70.0;

rightMotor.SetPercent(rightMotorPercent);

leftMotor.SetPercent(leftMotorPercent);

Sleep(time);

stopMotors();

}

//function to detect start light

void start()

{

//declare maximum value for light being off

float thresholdValue=1.0;

//keep checking to see if start light is on

bool check=true;

while(check)

{

//check to see if light is turned on

if(cdsCell.Value()<thresholdValue)

{

check=false;

}

}

}

void leverSwitch()

{

float leverAngle = 180.00;

float postLever = 2.00;

armServo.SetDegree(leverAngle);

Sleep(postLever);

}

void back( float time)

{

float rightMotorPercent=-70.0;

float leftMotorPercent=70.0;

rightMotor.SetPercent(rightMotorPercent);

leftMotor.SetPercent(leftMotorPercent);

Sleep(time);

stopMotors();

}

void left( float time){

float rightMotorPercent=60.0;

float leftMotorPercent=0.0;

rightMotor.SetPercent(rightMotorPercent);

leftMotor.SetPercent(leftMotorPercent);

Sleep(time);

stopMotors();

}

void read(){

LCD.SetFontColor(WHITE);

while(true){

LCD.Clear(BLACK);

LCD.WriteLine(midOpto.Value());

Sleep(50);

}

}

void right(float time){

float rightMotorPercent=0.0;

float leftMotorPercent=-60.0;

rightMotor.SetPercent(rightMotorPercent);

leftMotor.SetPercent(leftMotorPercent);

Sleep(time);

stopMotors();

}

int lightTest()

{

int test =0;

float time = 1.0;

if(cdsCell.Value()>0.63) //Testing for Blue value and will move straight before turning right into button

{

LCD.Clear(BLUE);

time = 1.2;

slowStraight(time);

time = 1.5;

slowRight(time);

stopMotors();

test = 2;

}else //Testing for red value and will turn right directly into button

{

LCD.Clear(RED);

time = 2.0;

slowRight(time);

time = 6.0;

//slowStraight(rightMotor,leftMotor,time);

stopMotors();

test = 1;

}

return test;

}

void slowStraight( float time)

{

float rightMotorPercent=20.0;

float leftMotorPercent=-20.0;

rightMotor.SetPercent(rightMotorPercent);

leftMotor.SetPercent(leftMotorPercent);

Sleep(time);

stopMotors();

}

void slowRight( float time)

{

float rightMotorPercent=0.0;

float leftMotorPercent=-30.0;

rightMotor.SetPercent(rightMotorPercent);

leftMotor.SetPercent(leftMotorPercent);

Sleep(time);

stopMotors();

}

void slowLeft(float time)

{

float rightMotorPercent=30.0;

float leftMotorPercent=0.0;

rightMotor.SetPercent(rightMotorPercent);

leftMotor.SetPercent(leftMotorPercent);

Sleep(time);

stopMotors();

}

void backRight(float time){

float rightMotorPercent=0.0;

float leftMotorPercent= 30.0;

rightMotor.SetPercent(rightMotorPercent);

leftMotor.SetPercent(leftMotorPercent);

Sleep(time);

stopMotors();

}

void backLeft( float time){

float rightMotorPercent=-60.0;

float leftMotorPercent=0.0;

rightMotor.SetPercent(rightMotorPercent);

leftMotor.SetPercent(leftMotorPercent);

Sleep(time);

stopMotors();

}

void toCoin(float time)

{

time = 0.05;

double Lv=leftOpto.Value();

double Mv=midOpto.Value();

double Rv=rightOpto.Value();

//got yellow value from earlier exploration, may need to be retaken

float yellow = 2.0;

int check = 0;

while (check ==0)

{

if(leftOpto.Value()>yellow&&midOpto.Value()>yellow&&rightOpto.Value()>yellow){

slowStraight(time);

}else{

LCD.Clear(YELLOW);

check++;

}

}

time = 1.0;

lineFollower(yellow,time);

Sleep(time);

stopMotors();

}

void lineFollower(float color, float time)

{

int state = ON\_LINE;

double Lv=leftOpto.Value();

double Mv=midOpto.Value();

double Rv=rightOpto.Value();

float t\_now;

t\_now=TimeNow();

while(frontBump.Value()){

Lv=leftOpto.Value();

Mv=midOpto.Value();

Rv=rightOpto.Value();

if(Lv<color){

LCD.Clear(GREEN);

state = LINE\_ON\_LEFT;

}

if(Mv<color){

LCD.Clear(WHITE);

state = ON\_LINE;

}

if(Rv<color){

LCD.Clear(RED);

state = LINE\_ON\_RIGHT;

}

switch(state) {

// If the line is on my right...

case LINE\_ON\_RIGHT:

leftMotor.SetPercent(-25.0);

rightMotor.SetPercent(0.0);

break;

// If I am on the line

case ON\_LINE:

leftMotor.SetPercent(-20.0);

rightMotor.SetPercent(20.0);

break;

// If the line is on my left...

case LINE\_ON\_LEFT:

leftMotor.SetPercent(0.0);

rightMotor.SetPercent(25.0);

break;

}

}

LCD.Clear();

Sleep(time);

stopMotors();

}

void dropCoin()

{

float leverAngle = 60.00;//need to test this value, Completely guessing here

float postLever = 2.00;

float k = 0.0;

while (k < leverAngle) {

armServo.SetDegree(k);

Sleep(7);

k += 1;

}

Sleep(postLever);

armServo.SetDegree(0);

}

void toRamp(){

//rotate until robot turns 90 degrees

float time = 0.15;

back(time);

time = 1.0;

backRight(time);

//while loop to run straight until both bump switches are pressed

while((rightBump.Value()==true)&&(leftBump.Value()==true)){

time = 0.1;

slowStraight(time);

}

time=1.0;

//hopefully rotates 90 degrees so bot lines up with ramp

backLeft(time);

}