#include <FEHLCD.h>

#include <FEHIO.h>

#include <FEHUtility.h>

#include <FEHServo.h>

#include <FEHMotor.h>

void straight(FEHMotor rightMotor, FEHMotor leftMotor, float time);

void start(AnalogInputPin cdsCell);

void leverSwitch(FEHServo arm);

void stopMotors(FEHMotor rightMotor, FEHMotor leftMotor);

void back(FEHMotor rightMotor, FEHMotor leftMotor, float time);

void left(FEHMotor rightMotor, FEHMotor leftMotor, float time);

void right(FEHMotor rightMotor, FEHMotor leftMotor, float time);

void read(AnalogInputPin cdsCell);

int lightTest(AnalogInputPin cdsCell,FEHMotor rightMotor, FEHMotor leftMotor);

void slowStraight(FEHMotor rightMotor, FEHMotor leftMotor, float time);

void slowRight(FEHMotor rightMotor, FEHMotor leftMotor, float time);

void backLeft(FEHMotor rightMotor, FEHMotor leftMotor, float time);

void backRight(float time);

//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::Servo0);

//cds cell in pin P1\_7

AnalogInputPin cdsCell(FEHIO::P1\_7);

//sets servo for wheel orientation

FEHServo botServo(FEHServo::Servo1);

int main(void)

{

LCD.Clear();

//read(cdsCell);

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

start(cdsCell);

armServo.SetMin(1200);

armServo.SetMax(2200);

armServo.SetDegree(0);

float time = 0.25;

//move to the ddr first light with cds cell centered on the first light

straight(rightMotor,leftMotor,time);

time=0.6;

right(rightMotor,leftMotor, time);

time=0.40;

straight(rightMotor,leftMotor, time);

//read light color and press correct button

int i=0;

int test=0;

//read(cdsCell);

time = 1.0;

Sleep(time);

while(i==0){

test = lightTest(cdsCell, rightMotor,leftMotor);

if(test!=0)

i=1;

}

//have light test to return int that can be used to determine the color

if (test = 2){ // run if loop to bring bot back if test is blue

time = 0.2;

back(rightMotor,leftMotor,time);

backRight(time);

time =1.2;

back(rightMotor,leftMotor,time);

stopMotors(rightMotor, leftMotor);

}

if (test = 1){ //run if loop to bring bot back if test is red

time = 0.2;

back(rightMotor,leftMotor,time);

backRight(time);

stopMotors(rightMotor, leftMotor);

}

//get bot on top of the ramp

time =0.5;

left(rightMotor,leftMotor,time);

time = 1.0;

straight(rightMotor,leftMotor,time);

time =0.5;

left(rightMotor,leftMotor,time);

time= 7.0;

straight(rightMotor,leftMotor,time);

}

//function to stop motors

void stopMotors(FEHMotor rightMotor, FEHMotor leftMotor){

rightMotor.Stop();

leftMotor.Stop();

}

//function to move to lever

void straight(FEHMotor rightMotor, FEHMotor leftMotor, float time)

{

float rightMotorPercent=-70.0;

float leftMotorPercent=-67.0;

rightMotor.SetPercent(rightMotorPercent);

leftMotor.SetPercent(leftMotorPercent);

Sleep(time);

stopMotors(rightMotor, leftMotor);

}

//function to detect start light

void start(AnalogInputPin cdsCell)

{

//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(FEHServo arm)

{

float leverAngle = 180.00;

float postLever = 2.00;

arm.SetDegree(leverAngle);

Sleep(postLever);

}

void back(FEHMotor rightMotor, FEHMotor leftMotor, float time)

{

float rightMotorPercent=70.0;

float leftMotorPercent=70.0;

rightMotor.SetPercent(rightMotorPercent);

leftMotor.SetPercent(leftMotorPercent);

Sleep(time);

stopMotors(rightMotor, leftMotor);

}

void left(FEHMotor rightMotor, FEHMotor leftMotor, float time){

float rightMotorPercent=-60.0;

float leftMotorPercent=0.0;

rightMotor.SetPercent(rightMotorPercent);

leftMotor.SetPercent(leftMotorPercent);

Sleep(time);

stopMotors(rightMotor, leftMotor);

}

void read(AnalogInputPin cdsCell){

LCD.SetFontColor(WHITE);

while(true){

LCD.Clear(BLACK);

LCD.WriteLine(cdsCell.Value());

Sleep(50);

}

}

void right(FEHMotor rightMotor, FEHMotor leftMotor, float time){

float rightMotorPercent=0.0;

float leftMotorPercent=-60.0;

rightMotor.SetPercent(rightMotorPercent);

leftMotor.SetPercent(leftMotorPercent);

Sleep(time);

stopMotors(rightMotor, leftMotor);

}

int lightTest(AnalogInputPin cds,FEHMotor rightMotor, FEHMotor leftMotor)

{

int test =0;

float time = 1.0;

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

{

LCD.Clear(BLUE);

time = 1.2;

slowStraight(rightMotor,leftMotor,time);

time = 1.5;

slowRight(rightMotor,leftMotor,time);

stopMotors(rightMotor, leftMotor);

test = 2;

// time = 2.0;

//slowStraight(rightMotor,leftMotor,time);

// time = 1.0;

//slowRight(rightMotor,leftMotor,time);

//Sleep(time);

//time = 6.0;

//slowStraight(rightMotor,leftMotor,time);

//stopMotors(rightMotor, leftMotor);

//test = 1;

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

{

LCD.Clear(RED);

time = 2.0;

slowRight(rightMotor,leftMotor,time);

time = 6.0;

//slowStraight(rightMotor,leftMotor,time);

stopMotors(rightMotor, leftMotor);

test = 1;

}

return test;

}

void slowStraight(FEHMotor rightMotor, FEHMotor leftMotor, float time)

{

float rightMotorPercent=-20.0;

float leftMotorPercent=-20.0;

rightMotor.SetPercent(rightMotorPercent);

leftMotor.SetPercent(leftMotorPercent);

Sleep(time);

stopMotors(rightMotor, leftMotor);

}

void slowRight(FEHMotor rightMotor, FEHMotor leftMotor, float time)

{

float rightMotorPercent=-5.0;

float leftMotorPercent=-30.0;

rightMotor.SetPercent(rightMotorPercent);

leftMotor.SetPercent(leftMotorPercent);

Sleep(time);

stopMotors(rightMotor, leftMotor);

}

void backRight(float time){

float rightMotorPercent=0.0;

float leftMotorPercent= 60.0;

rightMotor.SetPercent(rightMotorPercent);

leftMotor.SetPercent(leftMotorPercent);

Sleep(time);

stopMotors(rightMotor, leftMotor);

}

void backLeft(FEHMotor rightMotor, FEHMotor leftMotor, float time){

float rightMotorPercent=60.0;

float leftMotorPercent=0.0;

rightMotor.SetPercent(rightMotorPercent);

leftMotor.SetPercent(leftMotorPercent);

Sleep(time);

stopMotors(rightMotor, leftMotor);

}