//  ROCKRadioLineFollowEncoder 3-19-2023

#include <Servo.h>

#include <MINDSi.h>

#include "Encoder.h"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/ Radio auxilliary Ch 5 & 6 are mapped to switches A & B.

/ Reads radio Ch 5 & 6 pulse widths on pins 7 & 8 and

/ chooses functions depending on radio switchs A & B positions.

/ Encoder is on digital pins 0 (A blue) and 1 (B yellow)

/ If the order is reversed, going backwards would read as positive rpm and vice versa

/ Wheel diameter is about 4.45 in.; circumference is pi\*dia

/ Moving 10 feet requires (120/wheel circumference) revolutions

/ 3:1 gear ratio

/ drive motor is on pin 4 and steering is on pin 5

/ winch is on pin 6, and is controlled by radio channel 4,

/ left joystich (L/R) on arduino pin 9.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Servo steer, drive, winch;

unsigned long duration7; //pulse width on pin 7

unsigned long duration8; //pulse width on pin 8

const int inchesTravel = 120;

const float wheelDia = 4.45;

float shaftRevs = inchesTravel/(3.14 \* wheelDia) \* 3;

float rev;

float revStop;

const int encoderPins[2] = {0, 1};  // an array

const int threshold = 2000;  // QTI threshold

bool funcB = false;

void setup() {

//  Serial.begin(9600);

  pinMode(11, OUTPUT);  // green LED indicating radio drive

  pinMode(12, OUTPUT); // red LED indicating autonomous mode

  digitalWrite(11, LOW);

  digitalWrite(12, LOW);

  pinMode(7, INPUT);  // radio channel 5

  pinMode(8, INPUT);  // radio channel 6

  encoder::begin(encoderPins[0], encoderPins[1]);

  drive.attach(4);

  steer.attach(5);

  drive.write(90);

  steer.write(90);

  winch.attach(6);

  winch.write(55);  // 55 sets winch "off"

// if needed, winch can be disabled by attaching it to an unused pin

  delay(2000); //wait for 2 seconds to arm the ESC

}

void loop() {

  duration7 = pulseIn(7, HIGH);

  duration8 = pulseIn(8, HIGH);

if ((duration7 < 1250) && (duration8 < 1250)) {functionA();}

if ((duration7 > 1750) && (duration8 < 1250)) {functionB();}

if ((duration7 < 1250) && (duration8 > 1250) && (duration8 < 1750 )) {functionC();}

if ((duration7 > 1750) && (duration8 > 1250) && (duration8 < 1750 )) {functionD();}

if ((duration7 < 1250) && (duration8 > 1750)) {functionE();}

if ((duration7 > 1750) && (duration8 > 1750)) {functionF();}

//  Serial.print(duration7);

//  Serial.print("\t");

//  Serial.println(duration8);

}

void functionA() {

// radio drive

  digitalWrite(12, LOW);

  digitalWrite(11, HIGH);

  drive.write(getRadio(2));

  steer.write(getRadio(3));

  winch.write(getRadio(9));

  funcB = false;

}

void functionB() {

// line follower

  digitalWrite(11, LOW);

  digitalWrite(12, HIGH);

  rev = encoder::getRev();

  revStop = rev + shaftRevs;

  //run along line until rev exceeds revStop

  while ((rev < revStop) && !funcB)  {

    drive.write(100);

    // follow black line

    if (QTI(A0) > threshold) {

      steer.write(65);

    } else if (QTI(A2) > threshold) {

      steer.write(115);

    } else if (QTI(A1) > threshold) {

      steer.write(90);

    }

  rev = encoder::getRev();

  duration7 = pulseIn(7, HIGH);  // check if radio control changed

  duration8 = pulseIn(8, HIGH);

  if ((duration7 < 1250) && (duration8 < 1250)) {funcB = true;}

  }

// stop motor

drive.write(90);

delay(100);

funcB = true;

}

void functionC() {Serial.println("function C");}

void functionD() {Serial.println("function D");}

void functionE() {Serial.println("function E");}

void functionF() {Serial.println("function F");}