#include <SPI.h>

#include <Pixy.h>

Pixy pixy;

////////////////////////////////////////////////////////

// ENA IN1 IN2 IN3 IN4 ENB

int myPins[6] = {5, 6, 7, 8, 9, 10};

float deadZone = 0.15;

//int baseSpeed = 130;

////////////////////////////////////////////////////////

int cont = 0;

int signature, x, y, width, height;

float cx, cy, area;

void setup() {

Serial.begin(9600);

Serial.print("Starting...\n");

pixy.init();

for (int i = 0; i < 6; i++) {

pinMode(myPins[i], OUTPUT);

}

}

void loop() {

float turn = pixyCheck();

if (turn > -deadZone && turn < deadZone) {

turn = 0;

}

if (turn < 0) {

moveRobot(-80, 170);

}

else if (turn > 0) {

moveRobot(170, -80);

}

else {

moveRobot(70, 70);

}

delay(1);

}

float pixyCheck() {

static int i = 0;

int j;

uint16\_t blocks;

char buf[32];

// grab blocks!

blocks = pixy.getBlocks();

// If there are detect blocks, print them!

if (blocks)

{

signature = pixy.blocks[0].signature;

height = pixy.blocks[0].height;

width = pixy.blocks[0].width;

x = pixy.blocks[0].x;

y = pixy.blocks[0].y;

cx = (x + (width / 2));

cy = (y + (height / 2));

cx = mapfloat(cx, 0, 320, -1, 1);

cy = mapfloat(cy, 0, 200, 1, -1);

area = width \* height;

// Serial.print("sig: ");

// Serial.print(signature);

// Serial.print(" x:");

// Serial.print(x);

// Serial.print(" y:");

// Serial.print(y);

// Serial.print(" width: ");

// Serial.print(width);

// Serial.print(" height: ");

// Serial.print(height);

// Serial.print(" cx: ");

// Serial.print(cx);

// Serial.print(" cy: ");

// Serial.println(cy);

}

else {

cont += 1;

if (cont == 100) {

cont = 0;

cx = 0;

}

}

return cx;

}

float mapfloat(long x, long in\_min, long in\_max, long out\_min, long out\_max)

{

return (float)(x - in\_min) \* (out\_max - out\_min) / (float)(in\_max - in\_min) + out\_min;

}

void moveRobot(int leftSpeed, int rightSpeed)

{

if (leftSpeed >= 0) {

digitalWrite(myPins[1], 0);

digitalWrite(myPins[2], 1);

}

else {

digitalWrite(myPins[1], 1);

digitalWrite(myPins[2], 0);

}

if (rightSpeed >= 0) {

digitalWrite(myPins[3], 0);

digitalWrite(myPins[4], 1);

}

else {

digitalWrite(myPins[3], 1);

digitalWrite(myPins[4], 0);

}

analogWrite(myPins[0], abs(leftSpeed));

analogWrite(myPins[5], abs(rightSpeed));

}