#include <ESP8266WiFi.h>

#include <espnow.h>

#include<wire.h>

const long interval = 50;

unsigned long previousMillis = 0;

typedef struct struct\_message {

int x;

int y;

} struct\_message;

struct\_message data1;

int xAxis;

int yAxis;

void OnDataRecv(uint8\_t \* mac, uint8\_t \*incomingData, uint8\_t len) {

memcpy(&data1, incomingData, sizeof(data1));

// Serial.print("Bytes received: ");

// Serial.println(len);

xAxis = data1.x;

yAxis = data1.y;

}

void printIncomingReadings(){

// Display Readings in Serial Monitor

// Serial.println("INCOMING READINGS");

Serial.print(xAxis);

Serial.print(" - ");

Serial.println(yAxis);

}

void setup() {

// put your setup code here, to run once:

Serial.begin(115200);

//xet up wifi mode

WiFi.mode(WIFI\_STA);

WiFi.disconnect();

if (esp\_now\_init() != 0) {

Serial.println("Error initializing ESP-NOW");

return;

}

esp\_now\_set\_self\_role(ESP\_NOW\_ROLE\_COMBO);

esp\_now\_register\_recv\_cb(OnDataRecv);

//set up for i2c

Wire.begin(2);

Wire.onRequest(requestEvent);

}

void loop() {

// put your main code here, to run repeatedly:

printIncomingReadings();

delay(100);

//transmit struct incoming reading to arduino2

}

void requestEvent()

{

Serial.print("sending ("); Serial.print(sizeof data1); Serial.println(" bytes)");

Wire.write((byte \*)&data1, sizeof data1);

}

Code of arduino2

#include <Wire.h>

#define in1 5 //3

#define in2 6 //4

#define in3 7 //5

#define in4 8 //6

#define enA 1 //2

#define enB 2 //7

struct TransmitData

{

int x;

int y;

};

TransmitData data2;

void setup() {

Wire.begin();

Serial.begin(9600);

pinMode(enA, OUTPUT);

pinMode(enB, OUTPUT);

pinMode(in1, OUTPUT);

pinMode(in2, OUTPUT);

pinMode(in3, OUTPUT);

pinMode(in4, OUTPUT);

}

void loop() {

Serial.print("requesting ("); Serial.print(sizeof data2); Serial.print(" bytes)... ");

if (Wire.requestFrom(2, sizeof data2)) {

Wire.readBytes((byte\*) &data2, sizeof data2);

Serial.println("done");

xAxis=data2.x;

yAxis=data2.y;

Serial.println(xAxis);

Serial.println(yAxis);

}

else {

Serial.println("could not connect");

}

delay(100);

if (yAxis < 470) {

// Set Motor A backward

digitalWrite(in1, HIGH);

digitalWrite(in2, LOW);

// Set Motor B backward

digitalWrite(in3, HIGH);

digitalWrite(in4, LOW);

// Convert the declining Y-axis readings for going backward from 470 to 0 into 0 to 255 value for the PWM signal for increasing the motor speed

motorSpeedA = map(yAxis, 470, 0, 0, 255);

motorSpeedB = map(yAxis, 470, 0, 0, 255);

}

else if (yAxis > 550) {

// Set Motor A forward

digitalWrite(in1, LOW);

digitalWrite(in2, HIGH);

// Set Motor B forward

digitalWrite(in3, LOW);

digitalWrite(in4, HIGH);

// Convert the increasing Y-axis readings for going forward from 550 to 1023 into 0 to 255 value for the PWM signal for increasing the motor speed

motorSpeedA = map(yAxis, 550, 1023, 0, 255);

motorSpeedB = map(yAxis, 550, 1023, 0, 255);

}

// If joystick stays in middle the motors are not moving

else {

motorSpeedA = 0;

motorSpeedB = 0;

}

if (xAxis < 470) {

// Convert the declining X-axis readings from 470 to 0 into increasing 0 to 255 value

int xMapped = map(xAxis, 470, 0, 0, 255);

// Move to left - decrease left motor speed, increase right motor speed

motorSpeedA = motorSpeedA + xMapped;

motorSpeedB = motorSpeedB - xMapped;

// Confine the range from 0 to 255

if (motorSpeedA < 0) {

motorSpeedA = 0;

}

if (motorSpeedB > 255) {

motorSpeedB = 255;

}

}

if (xAxis > 550) {

// Convert the increasing X-axis readings from 550 to 1023 into 0 to 255 value

int xMapped = map(xAxis, 550, 1023, 0, 255);

// Move right - decrease right motor speed, increase left motor speed

motorSpeedA = motorSpeedA - xMapped;

motorSpeedB = motorSpeedB + xMapped;

// Confine the range from 0 to 255

if (motorSpeedA > 255) {

motorSpeedA = 255;

}

if (motorSpeedB < 0) {

motorSpeedB = 0;

}

}

// Prevent buzzing at low speeds (Adjust according to your motors. My motors couldn't start moving if PWM value was below value of 70)

if (motorSpeedA < 70) {

motorSpeedA = 0;

}

if (motorSpeedB < 70) {

motorSpeedB = 0;

}

digitalWrite(enA, motorSpeedA); // Send PWM signal to motor A

digitalWrite(enB, motorSpeedB); // Send PWM signal to motor B

}