

Interdyscyplinarny Projekt Zespołowy

Grupa Manipulator

Łącuch komunikacyjny:

- Enkoder AS5600
- Mikrokontroler ESP32 – manipulator
- Baza danych – PythonAnywhere
- Mikrokontroler ESP32 – robot

Program do komunikacji na ESP32:

```
#include <Wire.h>

#define AS5600_ADDR 0x36 // Adres enkodera AS5600 do I2C

// Pinouty pinów zasilających poszczególnych enkoderów
#define enkoder_1_power 14 // Przykładowe numery GPIO dla pinów zasilających
#define enkoder_2_power 15
#define enkoder_3_power 16
#define enkoder_4_power 17

void setup() {
    Wire.begin(); // Inicjalizacja interfejsu I2C
    Serial.begin(9600); // Inicjalizacja komunikacji szeregowej
}

double read(int enkoder_power_pin) {
    digitalWrite(enkoder_power_pin, HIGH); // Włącz zasilanie enkodera
    Wire.beginTransmission(AS5600_ADDR); // Rozpocznij transmisję do enkodera
    Wire.write(0x0E); // Adres rejestru odczytu kąta
    Wire.endTransmission(); // Zakończ transmisję
    Wire.requestFrom(AS5600_ADDR, 2); // Odczytaj 2 bajty danych z enkodera
    if (Wire.available() == 2) {
        int angle_LSB = Wire.read(); // Odczytaj mniej znaczący bajt
        int angle_MSB = Wire.read(); // Odczytaj bardziej znaczący bajt
        int angle = (angle_MSB << 8) | angle_LSB; // Oblicz wartość kąta
        digitalWrite(enkoder_power_pin, LOW); // Wyłącz zasilanie enkodera
        return(angle);
    }
    else {
        digitalWrite(enkoder_power_pin, LOW); // Wyłącz zasilanie enkodera w
        // przypadku błędu
        return(-1);
    }
}

void loop() {
    // Zmienne mierzonych kątów
    double angle_1;
    double angle_2;
    double angle_3;
    double angle_4;
```

```

// Odczyt położenia wszystkich enkoderów
angle_1 = read(enkoder_1_power);
angle_2 = read(enkoder_2_power);
angle_3 = read(enkoder_3_power);
angle_4 = read(enkoder_4_power);

// Ewentualna transformacja na bardziej użyteczne wartości

// Transmisja do bazy danych

// Testowy odczyt wartości
Serial.print("Pozycja enkodera 1: ");
Serial.println(angle_1);
Serial.print("/n");

delay(100); // Poczekać przed kolejnym odczytem
}

```

Tworzenie bazy danych:

```

// TWORZENIE BAZY DANYCH
CREATE TABLE motor(
    id integer PRIMARY KEY AUTO_INCREMENT,
    angle integer
);

CREATE TABLE point(
    id integer PRIMARY KEY AUTO_INCREMENT,
    x double,
    y double,
    z double,
    angle double
);

CREATE TABLE path(
    id integer PRIMARY KEY AUTO_INCREMENT,
    points_id integer,
    FOREIGN KEY(points_id) REFERENCES point(id)
);

CREATE TABLE angle_point(
    id integer PRIMARY KEY AUTO_INCREMENT,
    motor_id integer,
    FOREIGN KEY(motor_id) REFERENCES motor(id)
);

// DODAWANIE DANYCH DO TABEL
INSERT INTO motor(angle) VALUES(45);
INSERT INTO motor(angle) VALUES(90);
INSERT INTO point(x, y, z, angle) VALUES(3.0, 4.0, 5.0, 120.0);

INSERT INTO point(x, y, z, angle) VALUES(1.0, 2.0, 3.0, 45.0);
INSERT INTO point(x, y, z, angle) VALUES(2.0, 3.0, 4.0, 90.0);
INSERT INTO point(x, y, z, angle) VALUES(3.0, 4.0, 5.0, 120.0);

--Dodanie obiektu typu motor
INSERT INTO angle_point(motor_id) VALUES(motor_id);

// PRZYKŁADOWY ODCZYT DANYCH Z TABELI
SELECT* FROM point;

```

Obsługa HTTP w kodzie mikrokontrolera ESP32:

```
#include <WiFi.h>
#include <HTTPClient.h>

const char* ssid = "your-ssid";
const char* password = "your-password";
const char* server_address = "your-server-address"; // Adres serwera baz danych
const int server_port = 80; // Port serwera baz danych

void setup() {
    Serial.begin(9600);
    delay(100);

    WiFi.begin(ssid, password);
    Serial.println("Connecting to WiFi");

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("WiFi connected");
}

void loop() {
    // Tutaj umieść kod odczytu danych z enkoderów

    // Przykładowe dane
    int angle = 90;
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        http.begin(String("http://") + server_address + "/add_data"); // Adres
        // endpointu do dodawania danych
        http.addHeader("Content-Type", "application/x-www-form-urlencoded");

        // Przygotowanie danych
        String data = "angle=" + String(angle);

        int http_code = http.POST(data);

        if (http_code > 0) {
            String response = http.getString();
            Serial.println("Response: " + response);
        }
        else {
            Serial.println("Error on HTTP request");
        }
        http.end();
    }
    delay(5000); // Przykładowa pauza przed kolejnym przesyłaniem danych
}
```

REST API w PythonAnywhere:

```
from flask import Flask, request, jsonify
from flask_sqlalchemy import SQLAlchemy

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] =
'mysql://username:password@hostname/database_name'
db = SQLAlchemy(app)

class Data(db.Model):
    id = db.Column(db.Integer, primary_key = True)
    angle = db.Column(db.Integer)

@app.route('/add_data', methods = ['POST'])
def add_data():
    if request.method == 'POST':
        data = request.get_json()
        angle = data['angle']

        new_data = Data(angle = angle)
        db.session.add(new_data)
        db.session.commit()

        return jsonify({ 'message': 'Data added successfully' }), 201

if __name__ == '__main__':
    app.run()
```

ESP32 – przesył danych do REST API:

```
#include <WiFi.h>
#include <HTTPClient.h>

const char* ssid = "your-ssid";
const char* password = "your-password";
const char* server_address = "your-pythonanywhere-api-address"; // Adres REST API
na PythonAnywhere
const int server_port = 80; // Port REST API

void setup() {
    Serial.begin(9600);
    delay(100);

    WiFi.begin(ssid, password);
    Serial.println("Connecting to WiFi");

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("WiFi connected");
}
```

```

void loop() {
    // Tutaj umieść kod odczytu danych z enkoderów

    // Przykładowe dane
    int angle = 90;

    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        http.begin(String("http://") + server_address + "/add_data"); // Adres
        // endpointu do dodawania danych
        http.addHeader("Content-Type", "application/x-www-form-urlencoded");

        // Przygotowanie danych
        String data = "angle=" + String(angle);

        int http_code = http.POST(data);

        if (http_code > 0) {
            String response = http.getString();
            Serial.println("Response: " + response);
        }
        else {
            Serial.println("Error on HTTP request");
        }

        http.end();
    }

    delay(5000); // Przykładowa pauza przed kolejnym przesyłaniem danych
}

```