IOT and Bluetooth-Based Pick and Place Robotic Arm Vehicle

Code :

#include "BluetoothSerial.h"

#include <ESP32Servo.h>

// Bluetooth and servo objects

BluetoothSerial SerialBT;

Servo gripper; // S1

Servo y; // S2

Servo x; // S4

Servo base; // S3

// Motor pins

#define EN1 25

#define EN2 14

// Motor 1 (Back Left)

#define M1A 33

#define M1B 32

// Motor 2 (Back Right)

#define M2A 26

#define M2B 27

// Servo control pins

#define S1 13

#define S2 12

#define S3 15

#define S4 2

// Bluetooth device name and pin

String device\_name = "ESP32-BT-Slave";

const char \*pin = "1234"; // Change this to a more secure PIN

char BluetoothData;

// Motor control functions

void forward() {

// Motor 1

digitalWrite(M1A, 1);

digitalWrite(M1B, 0);

// Motor 2

digitalWrite(M2A, 1);

digitalWrite(M2B, 0);

}

void backward() {

// Motor 1

digitalWrite(M1A, 0);

digitalWrite(M1B, 1);

// Motor 2

digitalWrite(M2A, 0);

digitalWrite(M2B, 1);

}

void left() {

// Motor 1

digitalWrite(M1A, 0);

digitalWrite(M1B, 1);

// Motor 2

digitalWrite(M2A, 1);

digitalWrite(M2B, 0);

}

void right() {

// Motor 1

digitalWrite(M1A, 1);

digitalWrite(M1B, 0);

// Motor 2

digitalWrite(M2A, 0);

digitalWrite(M2B, 1);

}

void stop() {

// Motor 1

digitalWrite(M1A, 1);

digitalWrite(M1B, 1);

// Motor 2

digitalWrite(M2A, 1);

digitalWrite(M2B, 1);

}

void setup() {

// Initialize serial communication

Serial.begin(115200);

gripper.attach(S1);

y.attach(S2);

x.attach(S4);

base.attach(S3);

SerialBT.begin(device\_name); // Bluetooth device name

Serial.printf("The device with name \"%s\" is started.\nNow you can pair it with Bluetooth!\n", device\_name.c\_str());

// Optionally use a PIN for Bluetooth pairing

#ifdef USE\_PIN

SerialBT.setPin(pin);

Serial.println("Using PIN");

#endif

// Set motor control pins as output

int pins[] = {EN1, EN2, M1A, M1B, M2A, M2B};

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

pinMode(pins[i], OUTPUT);

}

// Enable motors

digitalWrite(EN1, 1);

digitalWrite(EN2, 1);

}

void loop() {

if (Serial.available()) {

SerialBT.write(Serial.read());

}

if (SerialBT.available()) {

BluetoothData = SerialBT.read();

Serial.println(BluetoothData);

// Motor control commands

if (BluetoothData == 'F') {

forward();

Serial.print("Forward");

} else if (BluetoothData == 'B') {

backward();

Serial.print("Backward");

} else if (BluetoothData == 'L') {

left();

Serial.print("Left");

} else if (BluetoothData == 'R') {

right();

Serial.print("Right");

} else if (BluetoothData == 'S') {

stop();

Serial.print("Stop");

}

// Servo control commands

else if (BluetoothData == 'g') {

int angle = SerialBT.parseInt();

gripper.write(angle);

Serial.println(angle);

} else if (BluetoothData == 'x') {

int angle = SerialBT.parseInt();

x.write(angle);

Serial.println(angle);

} else if (BluetoothData == 'y') {

int angle = SerialBT.parseInt();

y.write(angle);

Serial.println(angle);

} else if (BluetoothData == 'b') {

int angle = SerialBT.parseInt();

base.write(angle);

Serial.println(angle);

}

}

delay(20);

}