

EK HTTP Server

Martin Simov

16.1.2024

Einführung

Dieses Projekt implementiert einen einfachen HTTP-Server auf einem Arduino mit Hilfe eines Ultraschallsensors. Der Server gibt die gemessene Distanz in Zentimetern als HTTP-Antwort zurück. Dies kann beispielsweise für eine Einparkhilfe genutzt werden.

Projektbeschreibung

Das Projekt verwendet einen Ultraschallsensor, um die Entfernung zu einem Objekt zu messen. Ein Arduino-Board mit WiFi-Modul wird als Server eingesetzt. Der Server erfasst die Distanz und gibt diese Informationen über eine HTTP-Anfrage zurück. Die Informationen können dann auf einer Webseite angezeigt werden.

Theorie

Der Ultraschallsensor sendet einen Schallimpuls aus und misst die Zeit, die der Impuls benötigt, um vom Objekt zurückzukehren. Die gemessene Zeit wird verwendet, um die Distanz zum Objekt zu berechnen.

Arbeitsschritt

Der Code liest die gemessene Distanz vom Ultraschallsensor aus und gibt diese Informationen über eine einfache HTML-Seite zurück. Der Server wird auf dem Arduino gestartet, der mit dem WLAN verbunden ist. Die HTML-Antwort enthält die gemessene Distanz und wird alle Sekunde aktualisiert, um kontinuierliche Messungen anzuzeigen.

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

// Pin-Nummern für den Ultraschallsensor
int trigPin = 2;
int echoPin = 4;

// Variablen zur Messung der Distanz
int duration;
int distance;

// WLAN-Zugangsdaten
const char* ssid = "TGM-3"; // SSID des WLANs eintragen
const char* password = "123456789"; // Passwort des WLANs eintragen

WiFiServer server(80);
WiFiClient client;
```

```

// Funktion zur Behandlung des HTTP-Root-Anfragen
void handleRoot() {
    // Laufzeitmessung + Distanzberechnung durchführen
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    distance = duration * 0.034 / 2; // Lauzeit * Schallgeschwindigkeit(= 343m/

    // HTML-Antwort senden
    String response = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n";
    response += "<html><body>";
    response += "<h1>Einparkhilfe</h1>";
    response += "<p id=\"distance\">Distanz: " + String(distance) + " cm</p>";
    response += "<script>setTimeout(function() { location.reload(); }, 1000);</";
    response += "</body></html>";
    client.print(response);
}

void setup() {
    Serial.begin(9600);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);

    // WLAN-Verbindung herstellen
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(1000);
        Serial.println("Connecting to WiFi...");
    }
    Serial.println("Connected to WiFi");

    // Server starten
    server.begin();
    Serial.println("Server started");

    // IP-Adresse des Servers in der seriellen Konsole anzeigen
    Serial.print("IP address: ");
    Serial.println(WiFi.localIP());
}

void loop() {
    // Auf Verbindungen warten
    client = server.available();
    if (client) {

```

```
    Serial.println("New client connected");
    handleRoot();
    delay(200);
    client.stop();
    Serial.println("Client disconnected");
}

// Laufzeitmessung + Distanzberechnung durchführen
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = duration * 0.034 / 2;

delay(1000);
}
```

Zusammenfassung

Das Arduino-Programm implementiert einen einfachen HTTP-Server, der die Distanz von einem Ultraschallsensor misst und diese Information über eine HTML-Seite zurückgibt. Das Projekt könnte für Anwendungen wie eine Einparkhilfe verwendet werden.

Quellen

Keine spezifischen Quellen für dieses Beispiel. Der Code wurde auf Basis allgemeiner Arduino- und WiFi-Bibliotheken erstellt.