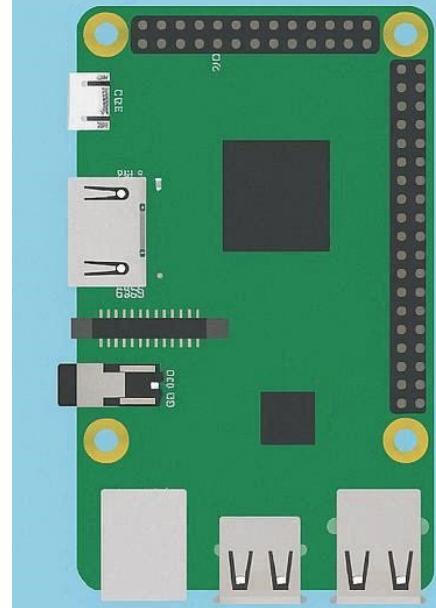


# Vegyestüzelésű kazán vízhőfokának és a kazánház hőmérsékletének mérése.

ESP32 + HOME ASSISTANT + MQTT  
INTEGRÁCIÓ

KÉSZÍTETTE: BARNA SZILVESZTER,  
BERNÁTH DÁVID ÉS KOVÁCS KORNÉL



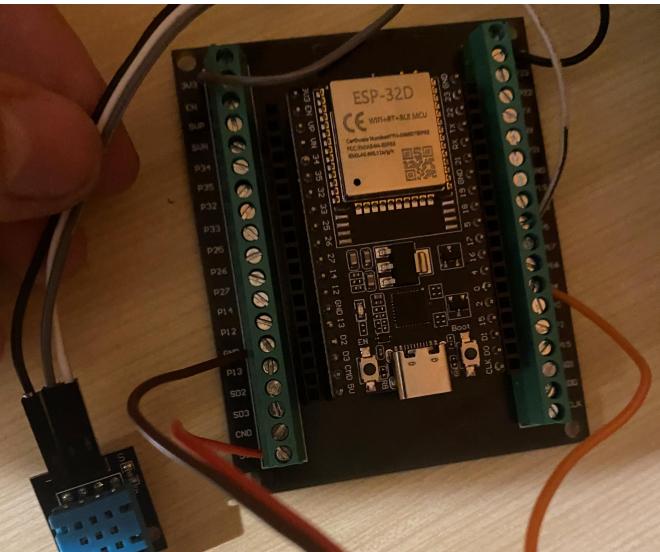
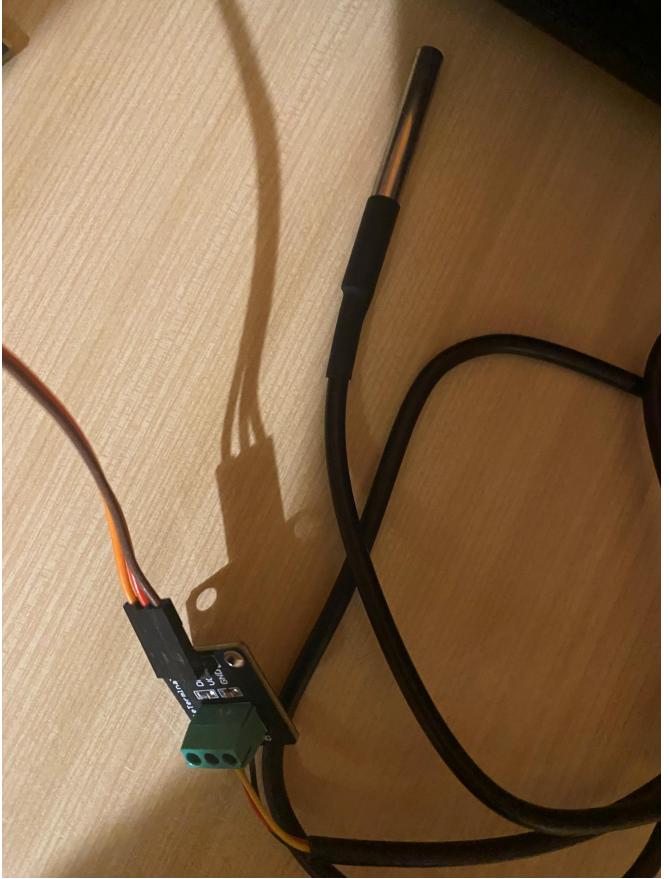
Home  
Assistant



# Szükséges szoftver és hardver komponensek

- Home Assistant OS:  
Otthonautomatizálási operációs rendszer
- Mosquitto broker: MQTT szerver (add-on formájában)
- Arduino IDE: ESP32 programozási környezet
- PubSubClient: MQTT kliens könyvtár

Komponens	Leírás	Funkció
<b>Raspberry Pi 3B</b>	Egylapos ARM processzoros Linux optimalizált miniszámítógép WiFi modullal szerelt egylapos mikrokontroller digitális és analóg portokkal	Home Assistant OS + Mosquitto MQTT broker futtatása Szenzor adatok gyűjtése és MQTT-n
<b>ESP32 mikrokontroller</b>	OneWire digitális hőmérő Digitális hőmérséklet és páratartalom szenzor	Kazán vízhőmérséklet mérése Műhely hőmérséklet és Páratartalom mérése
<b>DS18B20</b>		
<b>DHT11</b>		
<b>WiFi hálózat</b>	ESP32 és Home 2.4 GHz-es vezeték nélküli kommunikációs	Assistant közötti kommunikáció



# EESP-32D bekötése

# Arduino IDE EEESP32 kód part 1

The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** ESP32 Dev Module
- File Explorer:** Shows the file 'kazanhomero.ino' is selected.
- Code Editor:** Displays the C++ code for an ESP32 project. The code includes configurations for WiFi (WIFI\_SSID, WIFI\_PASSWORD), MQTT (MQTT\_SERVER, MQTT\_PORT, MQTT\_USER, MQTT\_PASS), and various topics for publishing sensor data (TOPIC\_DS18B20\_TEMP, TOPIC\_DHT\_TEMP, TOPIC\_DHT\_HUM). It also defines GPIO pins for DS18B20 (DS18B20\_PIN) and DHT (DHT\_PIN) sensors, and specifies the DHT type (DHT11). A measurement interval of 10000ms (10 seconds) is set. The code uses WiFiClient and PubSubClient libraries to handle network communication, and OneWire and DallasTemperature libraries for the DS18B20 sensor. A DHT object is created for the DHT11 sensor. Global variables include lastMeasure, wifiConnected, and mqttConnected. A connectWiFi() function is defined to print a message and set WiFi mode to STA.
- Status Bar:** Shows the line number (Ln 43, Col 21) and the connection status (ESP32 Dev Module on /dev/cu.usbserial-0001 [not connected]).

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

#include <OneWire.h>
#include <DallasTemperature.h>

#include <DHT.h>

// ----- ÁLLÍTHATÓ ADATOK -----
const char* WIFI_SSID      = "MamaNet_IoT";
const char* WIFI_PASSWORD   = "PannaReka2019";

const char* MQTT_SERVER    = "192.168.1.34";
const uint16_t MQTT_PORT    = 1883;

const char* MQTT_USER      = "esp32mqtt";
const char* MQTT_PASS      = "esp32mqtt";

const char* TOPIC_DS18B20_TEMP = "home/esp32/temperature_ds18b20";
const char* TOPIC_DHT_TEMP    = "home/esp32/temperature_dht";
const char* TOPIC_DHT_HUM     = "home/esp32/humidity_dht";

// GPIO-k
#define DS18B20_PIN 4
#define DHT_PIN      5
#define DHT_TYPE     DHT11

const unsigned long MEASURE_INTERVAL = 10000; // 10 s
// ----

WiFiClient espClient;
PubSubClient client(espClient);

OneWire oneWire(DS18B20_PIN);
DallasTemperature ds18b20(&oneWire);

DHT dht(DHT_PIN, DHT_TYPE);

unsigned long lastMeasure = 0;
bool wifiConnected = false;
bool mqttConnected = false;

void connectWiFi() {
  Serial.println("WiFi csatlakozasi kiserlet elindult...");
  WiFi.mode(WIFI_STA);
```

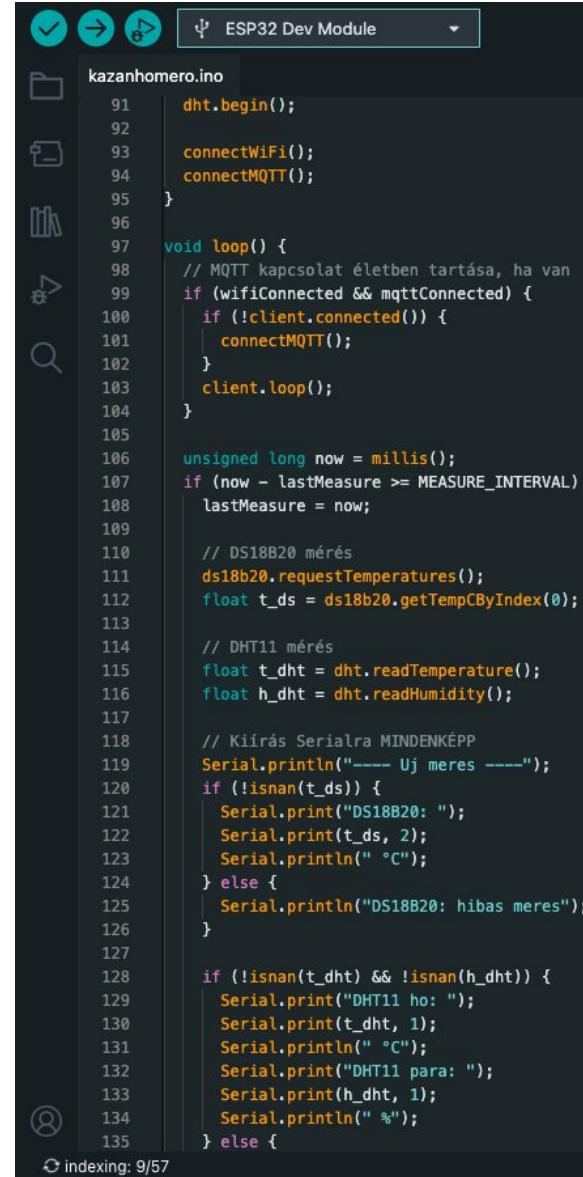
# Arduino IDE EEESP32 kód part 2

ESP32 Dev Module

```
kazanhomero.ino
46     WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
47
48     unsigned long start = millis();
49     while (WiFi.status() != WL_CONNECTED && millis() - start < 10000) { // max 10 s
50         delay(500);
51         Serial.print(".");
52     }
53     Serial.println();
54
55     if (WiFi.status() == WL_CONNECTED) {
56         wifiConnected = true;
57         Serial.print("WiFi kapcsolat sikeresen felelt, IP: ");
58         Serial.println(WiFi.localIP());
59     } else {
60         wifiConnected = false;
61         Serial.println("WiFi NINCS csatlakozva, offline mod (csak Serial.)");
62     }
63 }
64
65 void connectMQTT() {
66     if (!wifiConnected) {
67         mqttConnected = false;
68         return;
69     }
70
71     Serial.println("MQTT csatlakozasi kiserlet elindult...");
72     client.setServer(MQTT_SERVER, MQTT_PORT);
73
74     if (client.connect("esp32-sensors", MQTT_USER, MQTT_PASS)) {
75         mqttConnected = true;
76         Serial.println("MQTT OK");
77     } else {
78         mqttConnected = false;
79         Serial.print("MQTT hiba, rc=");
80         Serial.println(client.state());
81     }
82 }
83
84 void setup() {
85     Serial.begin(115200);
86     delay(1000);
87     Serial.println();
88     Serial.println("ESP32 DS18B20 + DHT11 teszt indul");
89
90     ds18b20.begin();
91 }
```

Ln 43, Col 21 ESP32 Dev Module on /dev/cu.usbserial-0001 [not connected]

# Arduino IDE EEESP32 kód part 3



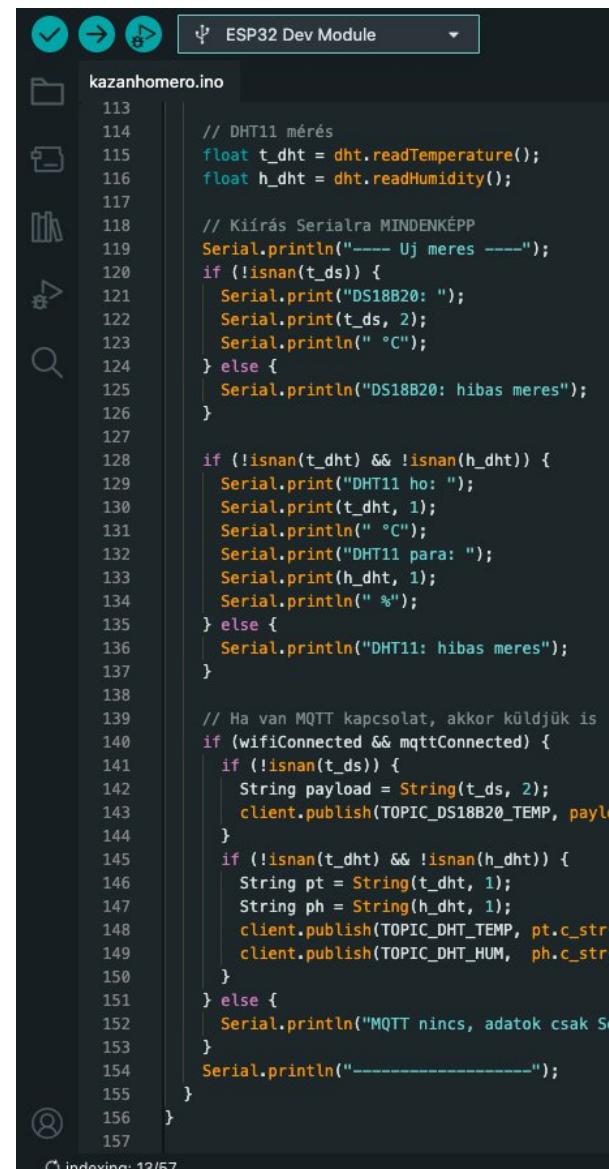
```
ESP32 Dev Module
kazanhomero.ino

91 dht.begin();
92
93 connectWiFi();
94 connectMQTT();
95 }
96
97 void loop() {
98 // MQTT kapcsolat életben tartása, ha van
99 if (wifiConnected && mqttConnected) {
100   if (!client.connected()) {
101     connectMQTT();
102   }
103   client.loop();
104 }
105
106 unsigned long now = millis();
107 if (now - lastMeasure >= MEASURE_INTERVAL) {
108   lastMeasure = now;
109
110   // DS18B20 mérés
111   ds18b20.requestTemperatures();
112   float t_ds = ds18b20.getTempCByIndex(0);
113
114   // DHT11 mérés
115   float t_dht = dht.readTemperature();
116   float h_dht = dht.readHumidity();
117
118   // Kiírás Serialra MINDENKÉPP
119   Serial.println("---- Uj meres ----");
120   if (!isnan(t_ds)) {
121     Serial.print("DS18B20: ");
122     Serial.print(t_ds, 2);
123     Serial.println(" °C");
124   } else {
125     Serial.println("DS18B20: hibas meres");
126   }
127
128   if (!isnan(t_dht) && !isnan(h_dht)) {
129     Serial.print("DHT11 ho: ");
130     Serial.print(t_dht, 1);
131     Serial.println(" °C");
132     Serial.print("DHT11 para: ");
133     Serial.print(h_dht, 1);
134     Serial.println(" %");
135   } else {
```

indexing: 9/57

Ln 43, Col 21 ESP32 Dev Module on /dev/cu.usbserial-0001 [not connected]

# Arduino IDE EEESP32 kód part 4



```
// DHT11 mérés
float t_dht = dht.readTemperature();
float h_dht = dht.readHumidity();

// Kiírás Serialra MINDENKÉPP
Serial.println("---- Uj meres ----");
if (!isnan(t_ds)) {
    Serial.print("DS18B20: ");
    Serial.print(t_ds, 2);
    Serial.println(" °C");
} else {
    Serial.println("DS18B20: hibas meres");
}

if (!isnan(t_dht) && !isnan(h_dht)) {
    Serial.print("DHT11 ho: ");
    Serial.print(t_dht, 1);
    Serial.println(" °C");
    Serial.print("DHT11 para: ");
    Serial.print(h_dht, 1);
    Serial.println(" %");
} else {
    Serial.println("DHT11: hibas meres");
}

// Ha van MQTT kapcsolat, akkor küldjük is
if (wifiConnected && mqttConnected) {
    if (!isnan(t_ds)) {
        String payload = String(t_ds, 2);
        client.publish(TOPIC_DS18B20_TEMP, payload.c_str(), true);
    }
    if (!isnan(t_dht) && !isnan(h_dht)) {
        String pt = String(t_dht, 1);
        String ph = String(h_dht, 1);
        client.publish(TOPIC_DHT_TEMP, pt.c_str(), true);
        client.publish(TOPIC_DHT_HUM, ph.c_str(), true);
    }
} else {
    Serial.println("MQTT nincs, adatok csak Serialon.");
}
Serial.println("-----");
```

⌚ indexing: 13/57

Ln 43, Col 21 ESP32 Dev Module on /dev/cu.usbserial-0001 [not connected]

---

## Mosquitto Add-on Telepítése:

- 1. Nyissuk meg a Home Assistant webes felületét
- 2. Menjünk a Beállítások > Kiegészítők > Kiegészítő bolt menübe
- 3. Keresük ezt: mosquitto
- 4. Válasszuk ki ezt: Mosquitto broker (az első)
- 5. Kattintsunk az Telepítés gombra
- 6. Várjuk meg, hogy befejeződik
- 7. Kapcsoljuk be: kattintsunk az Indítás gombra
- 8. Pipáljuk be az Indítás rendszerrel opciót

## MQTT Felhasználó Létrehozása:

- 1. Home Assistant: Beállítások > Személyek > Felhasználók
- 2. Kattintsunk az Felhasználó hozzáadása gombra
- 3. Kitöltés:
  - Név: MQTT ESP32 (tetszőleges)
  - Felhasználónév: esp32mqtt
  - Jelszó: esp32mqtt (vagy másik)
  - Bejelentkezhet: (pipa)
  - Admin: (nincs pipa)
- 4. Kattintsunk a Létrehozás gombra

## MQTT Integráció Hozzáadása:

- 1. Beállítások > Eszközök és szolgáltatások
- 2. Jobb alsó sarokban: + Integráció hozzáadása gomb
- 3. Keressük meg ezt: MQTT
- 4. A varázsló automatikusan kitölti:
  - Broker: core-mosquitto
  - Port: 1883
- 5. Kattintsunk a Küldés gombra
- 6. Befejezés

## Szenzorok felvétele:

- 1. Beállítások > Kiegészítők > Kiegészítő bolt
- 2. Keressük meg ezt: file editor
- 3. Telepítés és indítás
- 4. Ha sikeresen települt a bal oldali menüben megjelenik a File editor ikon
  - configuration.yaml Szerkesztése
  - 1. A File editorban nyissuk meg ezt: configuration.yaml
  - 2. Menjünk a fájl végére
  - 3. Adjuk hozzá az alábbiakat (vigyázzunk az indentálásra!)
  - 4. Kód beírása
- 5. Beállítások > Rendszer > Újraindítás

```
mqtt:  
  sensor:  
    - name: "Muhely homereklet (DHT)"  
      state_topic: "home/esp32/temperature_dht"  
      unit_of_measurement: "°C"  
  
    - name: "Muhely paratartalom (DHT)"  
      state_topic: "home/esp32/humidity_dht"  
      unit_of_measurement: "%"   
  
    - name: "Kazan vizhofoky (DS18B20)"  
      state_topic: "home/esp32/temperature_ds18b20"  
      unit_of_measurement: "°C"
```

The screenshot shows a Home Assistant dashboard with several cards:

- Nappali** card: Shows Google TV (off), Lexmark MS510dn (Tétlen), and Lexmark MS510dn Black (86% battery).
- Mama szobája** card: Shows Samsung TV (UE43RU7102KXXH) off, Wireless Router TL-WR1043ND Feltöltési sebesség 0,0 KiB/s, and Wireless Router TL-WR1043ND Letöltési sebesség 0,0 KiB/s.
- KK**, **BD**, **BS** cards: Status cards for KK (Otthon), BD (Ismeretlen), and BS (Ismeretlen).
- sensor** card: Lists Kazán vízhőfok (DS18B20) at 26,25 °C, Műhely hőmérséklet (DHT) at 20,3 °C, and Műhely páratartalom (DHT) at 30,0%.
- Felhős** card: Weather forecast for Otthon: 6,3 °C, 95% chance of rain.

# Home Assistant konfiguráció