ÓBUDAI EGYETEM
KANDÓ KÁLMÁN VILLAMOSMÉRNÖKI KAR

Nemzeti Tehetség Program

Beágyazott ipari vezérlők Szakszeminárium

ESP32 alapok. Analóg és digitális jelek feldolgozása mikrovezérlővel.

Kandó Kálmán Villamosmérnöki Szakkollégium

Borsos Döníz borsos.doniz@uni-obuda.hu

KANDÓ KÁLMÁN SZAKKOLLÉGIUM

MINISZTERELNÖKSÉG

ÓBUDAI EGYETEM
KANDÓ KÁLMÁN VILLAMOSMÉRNÖKI KAR

KANDÓ KÁLMÁN SZAKKOLLÉGIUM

BEÁGYAZOTT IPARI VEZÉRLŐK

Szakszeminárium
2022.11.21-22.

A Kandó Kálmán Villamosmérnöki Szakkollégium 16 órás szakszemináriumot szervez Beágyazott ipari vezérlők témakörben.

Témakörök:

- Beágyazott ipari vezérlők
- Mérésadatgyűjtés, szenzorok
- IoT, Ipar 4.0
- ESP32
- Bluetooth, WiFi
- GUI fejlesztése
  - Mobiltelefonos applikáció
  - Böngészőből futtatható GUI
- Miniprojektek

ÜTEMEZÉS

Első nap - 2022.11.21.

- Elméleti alapok: 09.00-13.00
- Ebédszünet: 13.00-13.30
- Gyakorlati alapok: 13.30-17.30

Második nap - 2022.11.22.

- Miniprojektek: 9.00-11.00
- Bemutatók, vizsga: 11.00-12.00
- Értékelés: 12.00-12.30
- Oklevélátadó: 12.30-13.00

A jelentkezők további tájékoztatást kapnak a részletekről!

ESP32

Kapcsolat:
borsos.doniz@uni-obuda.hu
sandor.tamas@uni-obuda.hu

Blynk

Jelentkezés: https://kando-szakkoli.uni-obuda.hu/

A program megvalósulását támogatja a Nemzeti Tehetség Program és a Miniszterelnökség, az Emberi Erőforrás Támogatáskezelő által kiírt "Szakkollégiumok tehetséggondozó programjainak támogatása" című pályázata.

Nemzeti Tehetség Program

MINISZTERELNÖKSÉG

Nemzeti Tehetség Program

A szakszeminárium megvalósulását támogatja a Nemzeti Tehetség Program és a Miniszterelnökség, az Emberi Erőforrás Támogatáskezelő által kiírt "Szakkollégiumok tehetséggondozó programjainak támogatása" című pályázata.

MINISZTERELNÖKSÉG

Tartalom

- Espressif - ESP32
- ESP32 NodeMCU
- Fejlesztői környezetek
- Programozás
- Arduino alapfüggvények
- Mintaprojektek

Espressif – ESP32

- https://www.espressif.com/

ESP32

Robust Design

Ultra-Low Power Consumption

High Level of Integration

Hybrid Wi-Fi & Bluetooth Chip

ESP32 jellemzői (ESP32WROOM32E)

- MCU
  - dual-core 32-bit LX6 microprocessor
  - up to 240 MHz
  - 448 KB ROM
  - 520 KB SRAM
  - 16 KB SRAM in RTC
- WiFi
  - 802.11b/g/n
  - Bit rate: 802.11n up to 150 Mbps
- Bluetooth®
  - Bluetooth V4.2 BR/EDR
  - Bluetooth LE specification

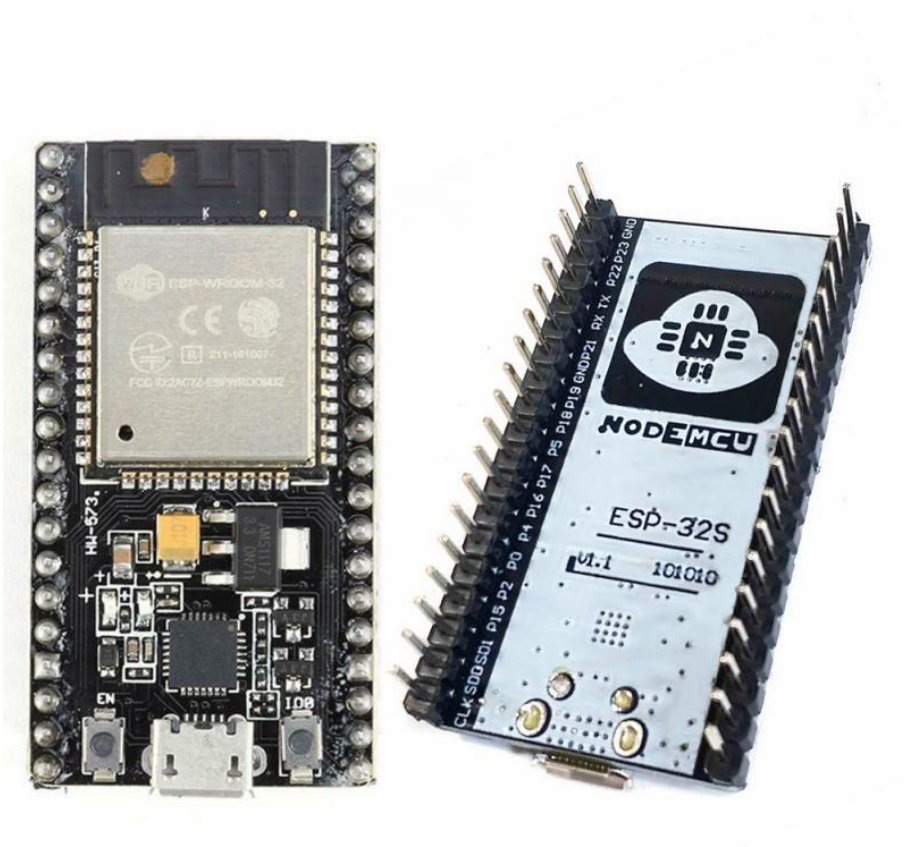
https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf

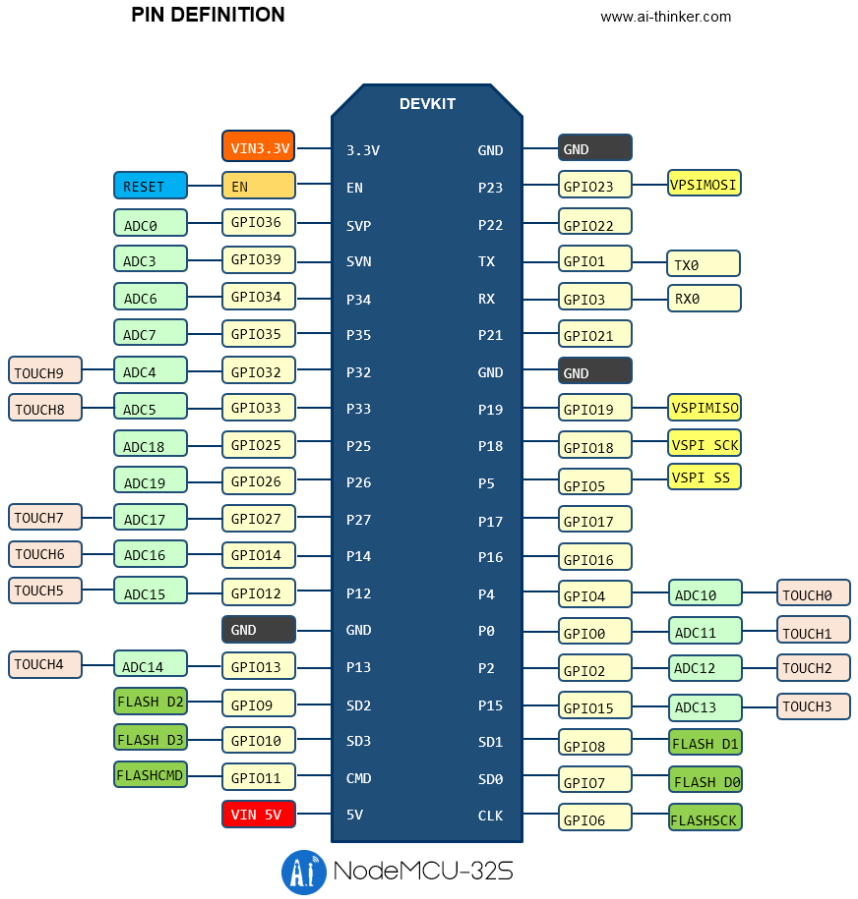
ESP32 jellemzői (ESP32WROOM32E)

- Interfészek
  - SD card
  - UART
  - SPI
  - SDIO
  - I2C
  - LED PWM, Motor PWM
  - pulse counter
  - GPIO
  - capacitive touch sensor
  - ADC, DAC
  - Two-Wire Automotive Interface
- További jellemzők
  - 40 MHz crystal oscillator
  - 4 MB SPI flash
  - Operating voltage/Power: 3.0-3.6 V
  - Operating temperature range: –40-85 °C

https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf

ESP32 NodeMCU



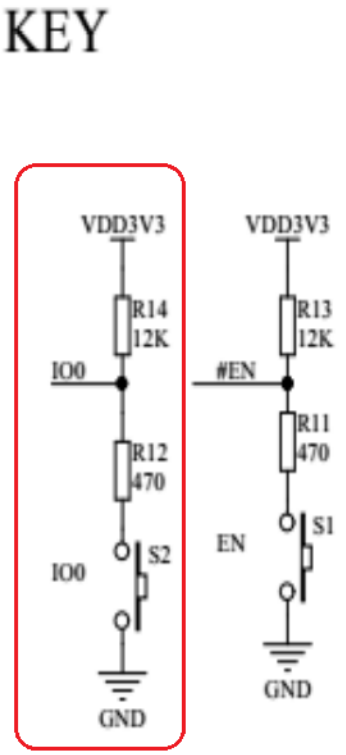


https://docs.ai-thinker.com/_media/esp32/docs/nodemcu-32s_product_specification.pdf

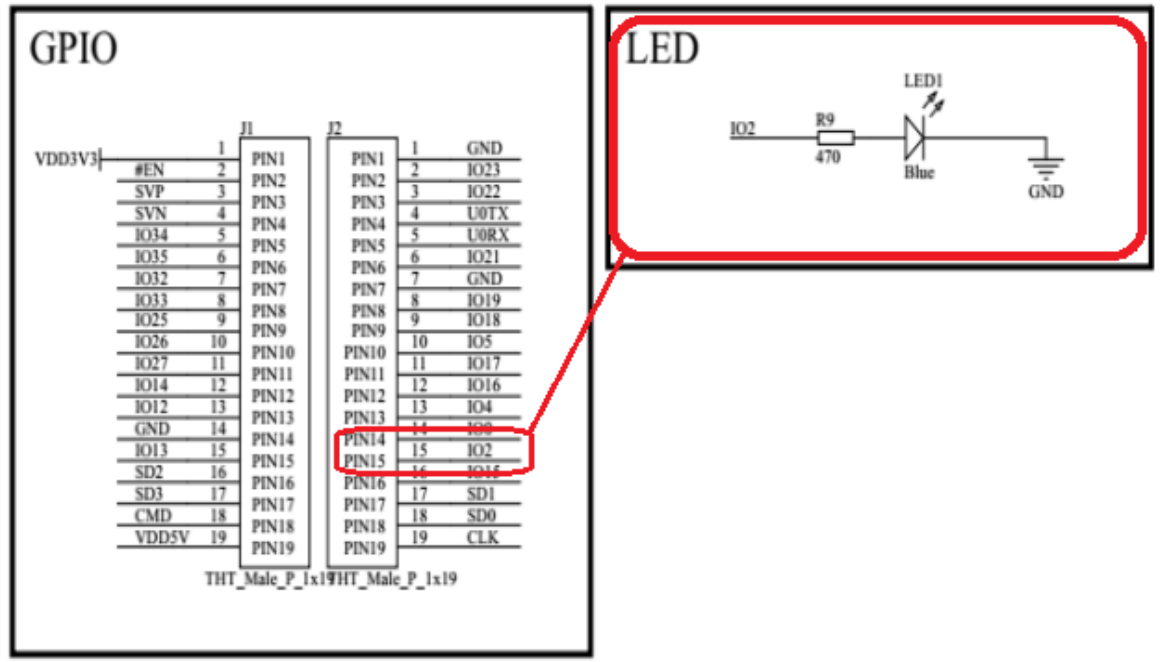
ESP32 NodeMCU - Jellemzők

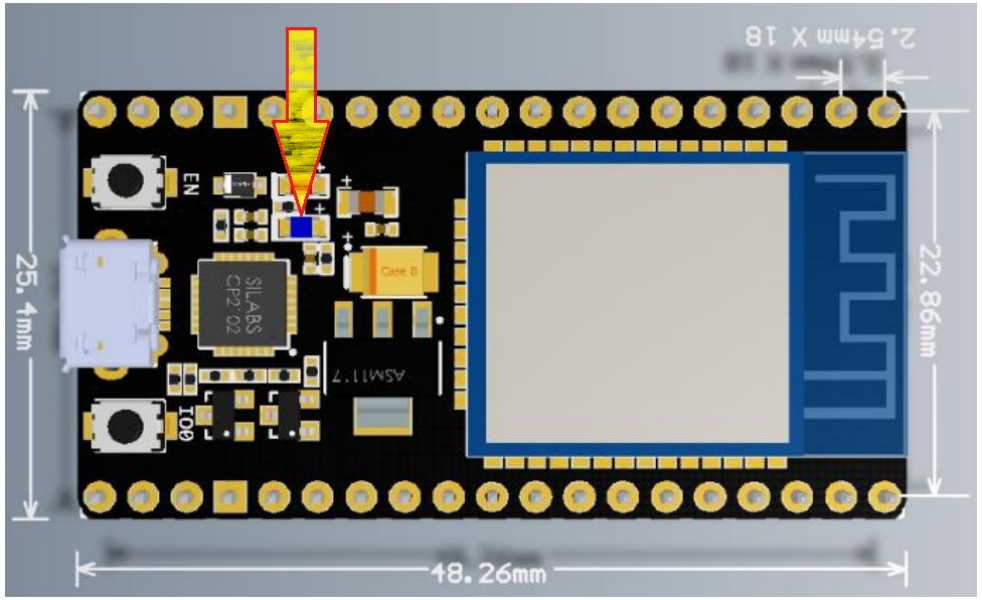
Module model	ESP-WROOM-32s
Size	25.4*48.26*3mm(±0.2mm)
Certification	FCC/CE-RED/IC/TELEC/KCC/SRRC/NCC/ BQB/ RoHS/REACH
SPI Flash	32Mbit(default)
Support interface	UART/GPIO/ADC/DAC/SDIO/SD card /PWM/I2C/I2S
Integrated crystal oscillator	40MHz Crystal oscillator
IO Port	38
Antenna	Onboard antenna
Power Supply	Voltage 3.0V ~ 3.6V, Typical 3.3V, Current >500mA
Operating Temperature	-40 ℃ ~ 85 ℃
Storage Environment	-40 ℃ ~ 120 ℃

ESP32 NodeMCU – Boot gomb (IO0)



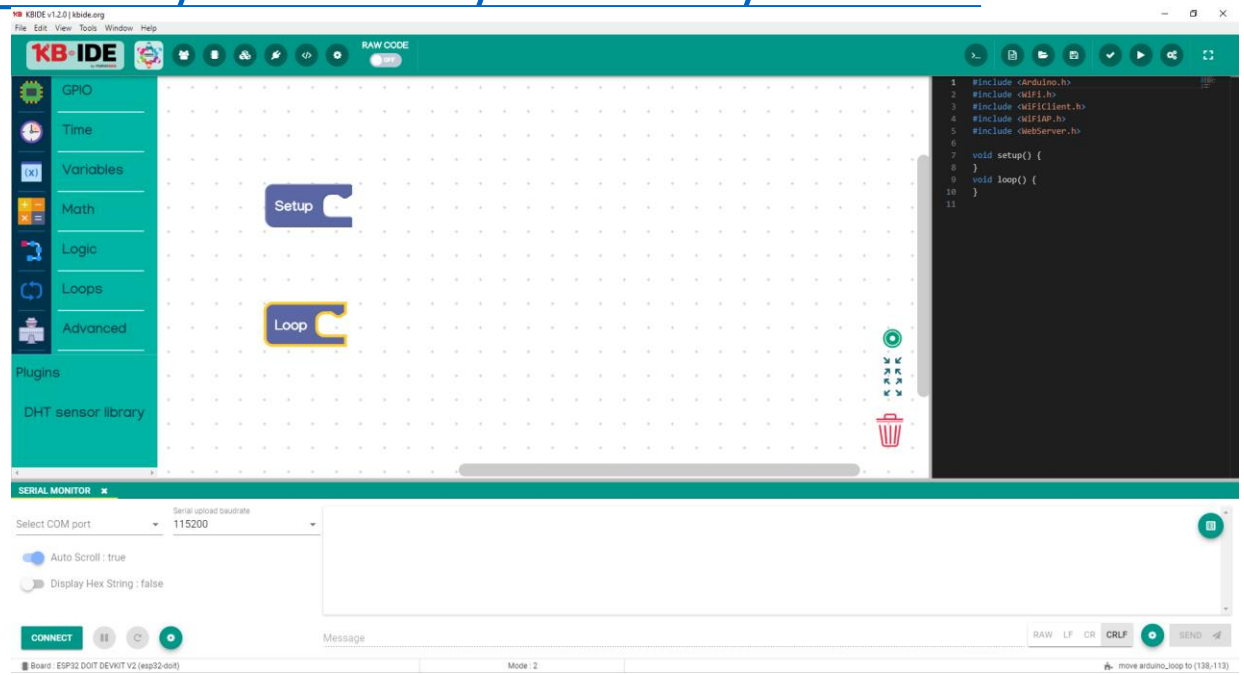
ESP32 NodeMCU - LED





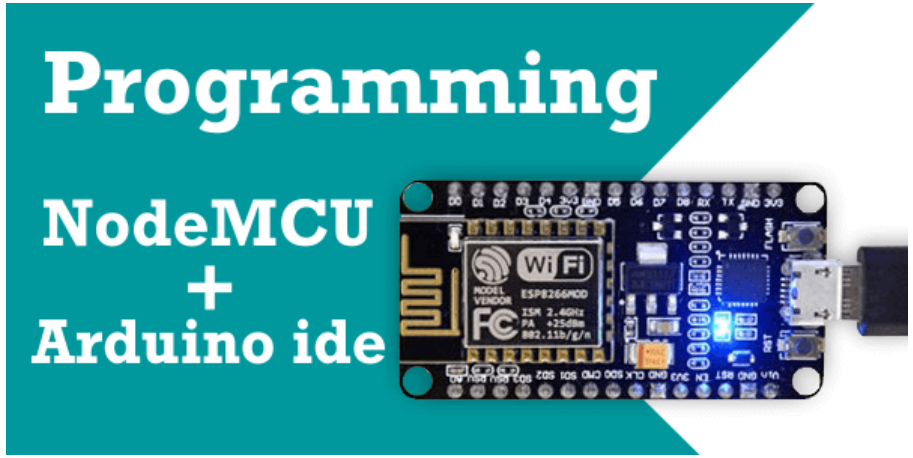
Fejlesztői környezetek

- KB IDE
  - https://kbide.org/
  - https://github.com/MakerAsia/KBProIDE/releases



Fejlesztői környezetek

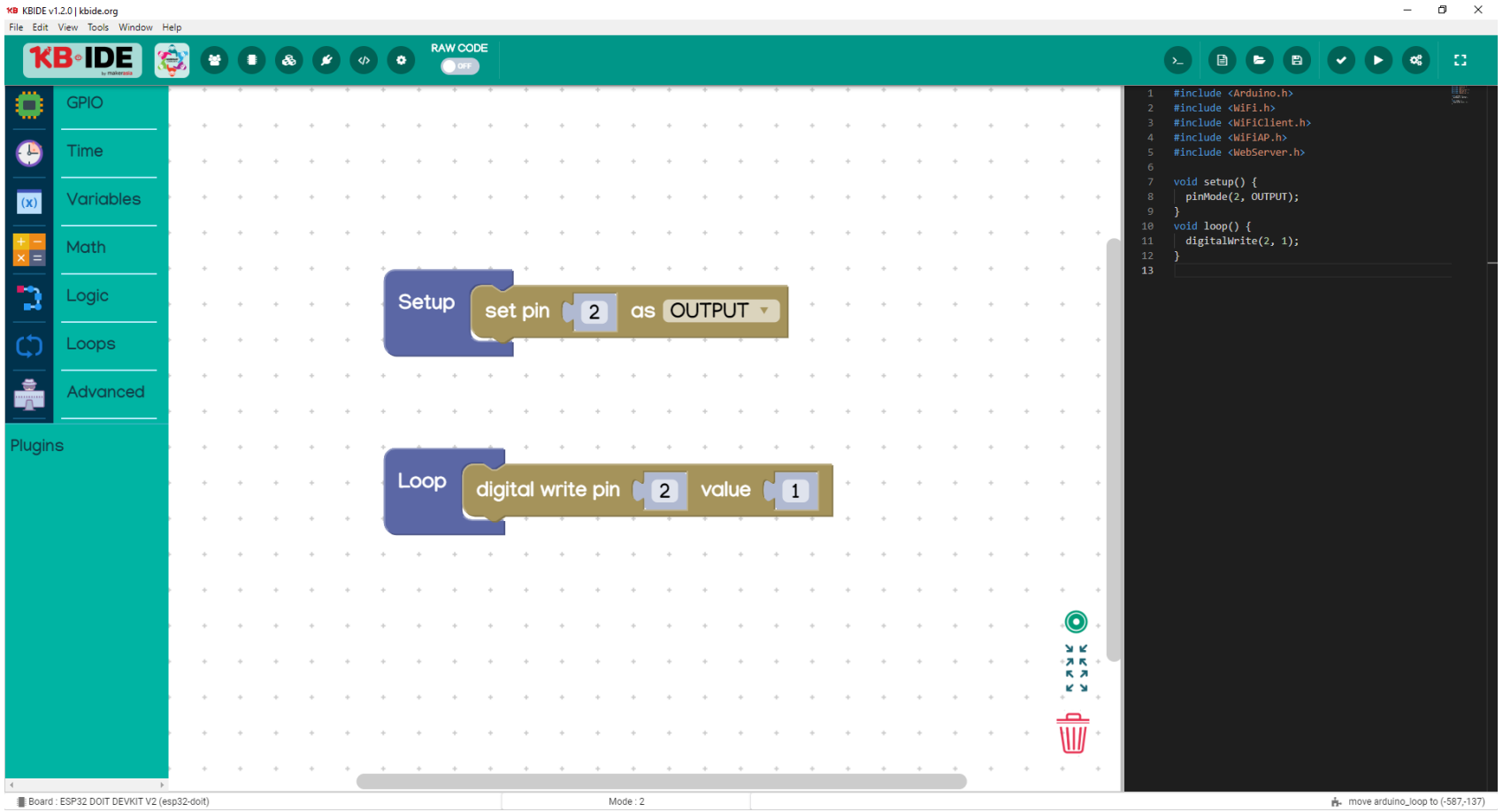
- Arduino IDE
  - https://www.arduino.cc/en/software
  - https://dl.espressif.com/dl/package_esp32_index.json, http://arduino.esp8266.com/stable/package_esp8266com_index.json



Fejlesztői környezetek

- ESP-IDF
  - https://www.espressif.com/en/products/sdks/esp-idf
  - https://github.com/espressif/esp-idf

KB IDE – LED bekapcsolása

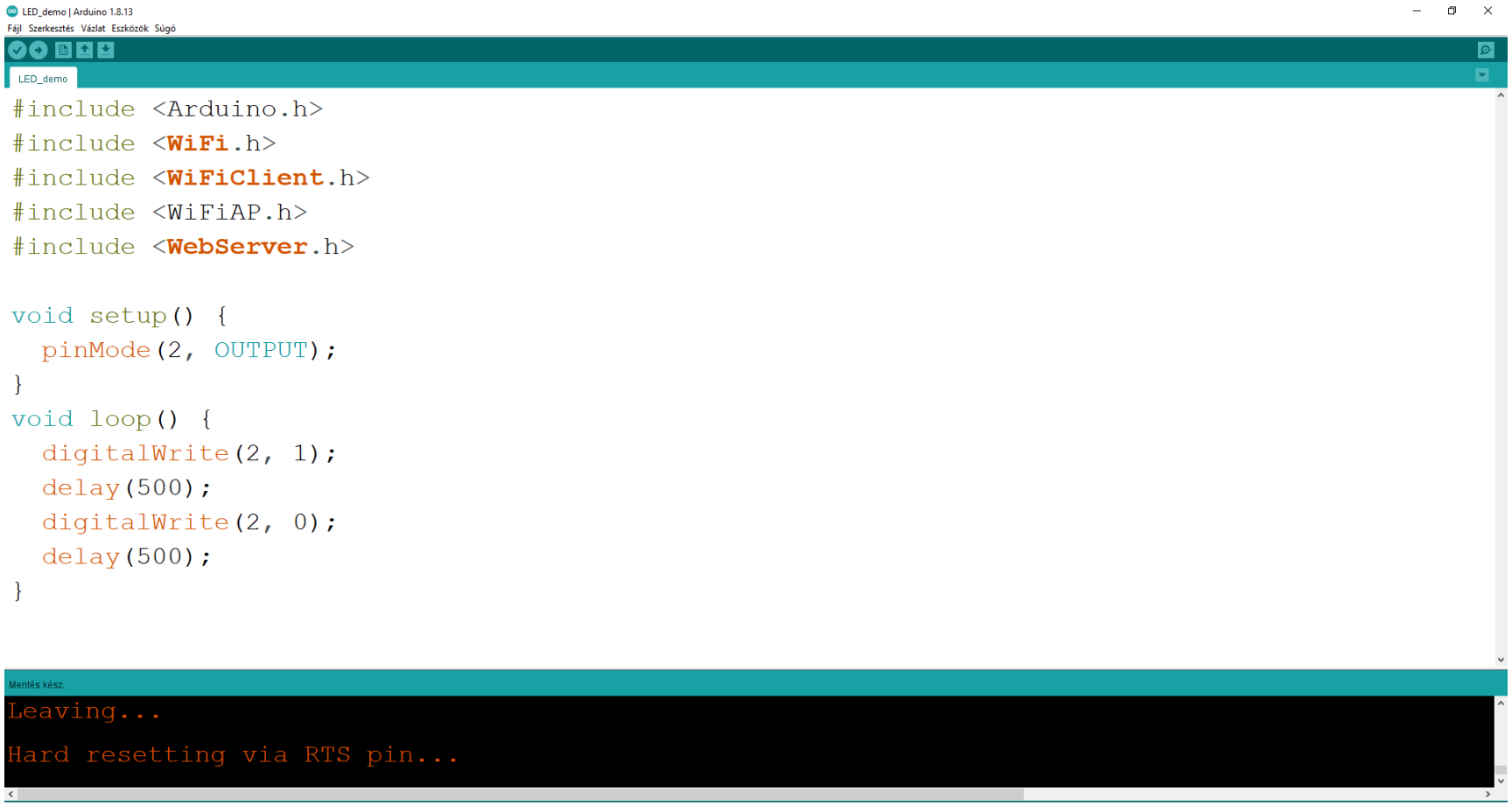


Arduino alapfüggvények – (Digitális I/O)

- pinMode()
  - pinMode(pin, mode)
    - pin: the Arduino pin number to set the mode of.
    - mode: INPUT, OUTPUT, INPUT_PULLUP
  - Példa:
    - pinMode(13, OUTPUT); // sets the digital pin 13 as output
- digitalWrite()
  - digitalWrite(pin, value)
    - pin: the Arduino pin number.
    - value: HIGH or LOW.
  - Példa:
    - digitalWrite(13, HIGH); // sets the digital pin 13 on
- digitalRead()
  - digitalRead(pin)
    - pin: the Arduino pin number you want to read
    - Returns: HIGH or LOW
  - Példa:
    - val = digitalRead(inPin); // read the input pin

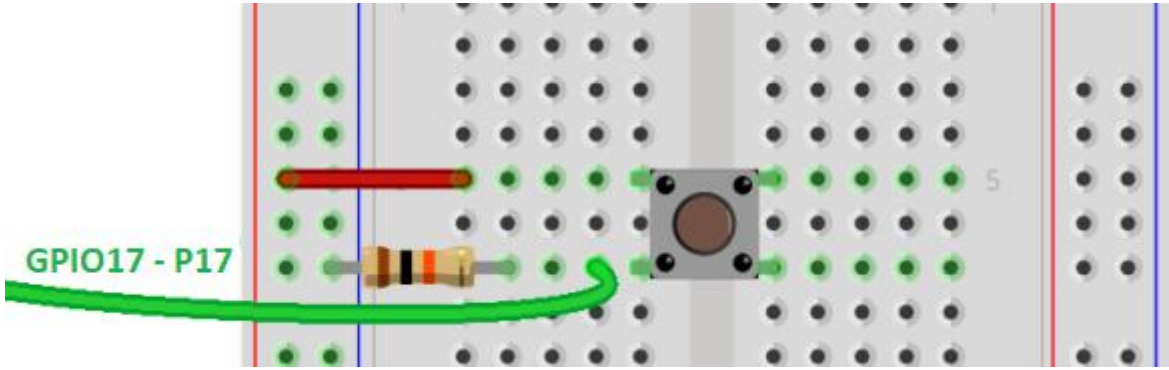
https://www.arduino.cc/reference/en/

Arduino IDE – LED villogtatása



Arduino IDE – GOMB olvasása

```
#include <Arduino.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <WiFiAP.h>
#include <WebServer.h>

void setup() {
  pinMode(17, INPUT);
  pinMode(2, OUTPUT);
}

void loop() {
  digitalWrite(2, digitalRead(17));
}
```



Arduino alapfüggvények – (Analóg IN)

- analogRead()
  - analogRead(pin)
    - pin: the name of the analog input pin to read from
    - Returns: The analog reading on the pin. (0-1023 for 10 bits or 0-4095 for 12 bits)
  - Példa:
    - val = analogRead(analogPin); // read the input pin

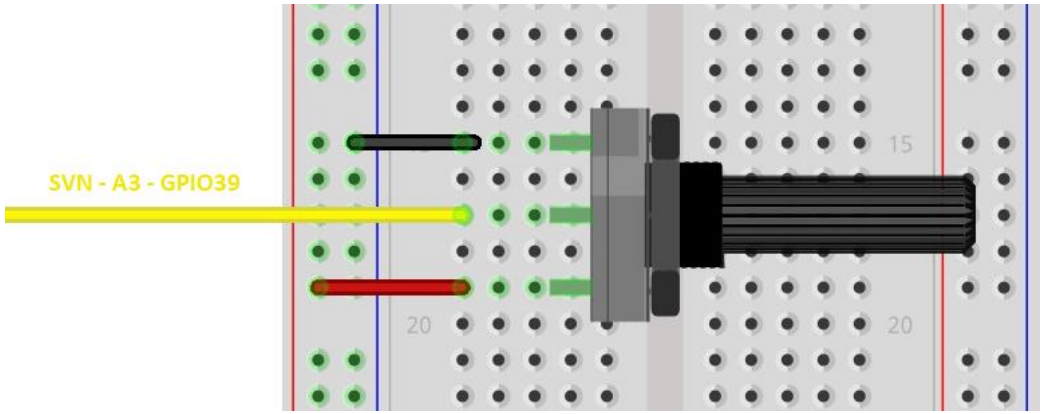
- 0-4095 = 0-3300mV
- val*(3300.0/4096.0) [mV]

- Értékek kijelzéséhez további függvények
  - Serial.begin(9600); // setup serial -> INIT
  - Serial.println(val); // debug value

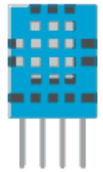
https://www.arduino.cc/reference/en/

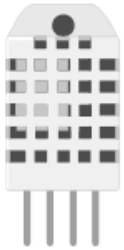
Arduino IDE – Potméter (analóg PIN olvasása)

```
#include <Arduino.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <WiFiAP.h>
#include <WebServer.h>

int analogValue = 0;

void setup() {
  Serial.begin(9600);
}

void loop() {
  analogValue = analogRead(A3);
  Serial.print(float(analogValue)*(3300.0/4096.0));
  Serial.print(" mV");
}
```



Mintaprojekt – DHT11





	DHT11	DHT22
Operating Voltage	3 to 5V	3 to 5V
Max Operating Current	2.5mA max	2.5mA max
Humidity Range	20-80% / 5%	0-100% / 2-5%
Temperature Range	0-50°C / ± 2°C	-40 to 80°C / ± 0.5°C
Sampling Rate	1 Hz (reading every second)	0.5 Hz (reading every 2 seconds)
Body size	15.5mm x 12mm x 5.5mm	15.1mm x 25mm x 7.7mm
Advantage	Ultra low cost	More Accurate

Example 1: 40 data is received:

0011 0101	0000 0000	0001 1000	0000 0000	0100 1101
High humidity 8	Low humidity 8	High temp. 8	Low temp. 8	Parity bit

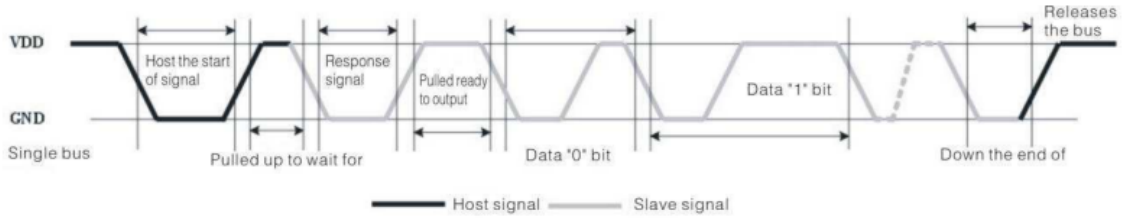
Calculate：

0011 0101+0000 0000+0001 1000+0000 0000= 0100 1101

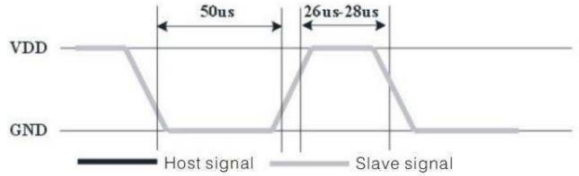
Received data is correct：

Humidity：0011 0101=35H=53%RH

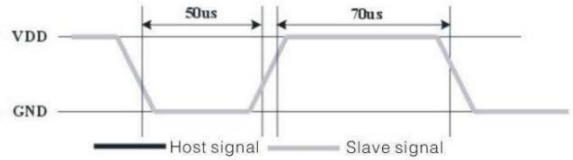
Temperature：0001 1000=18H=24℃



Data Timing Diagram



Bit data "0" bit format

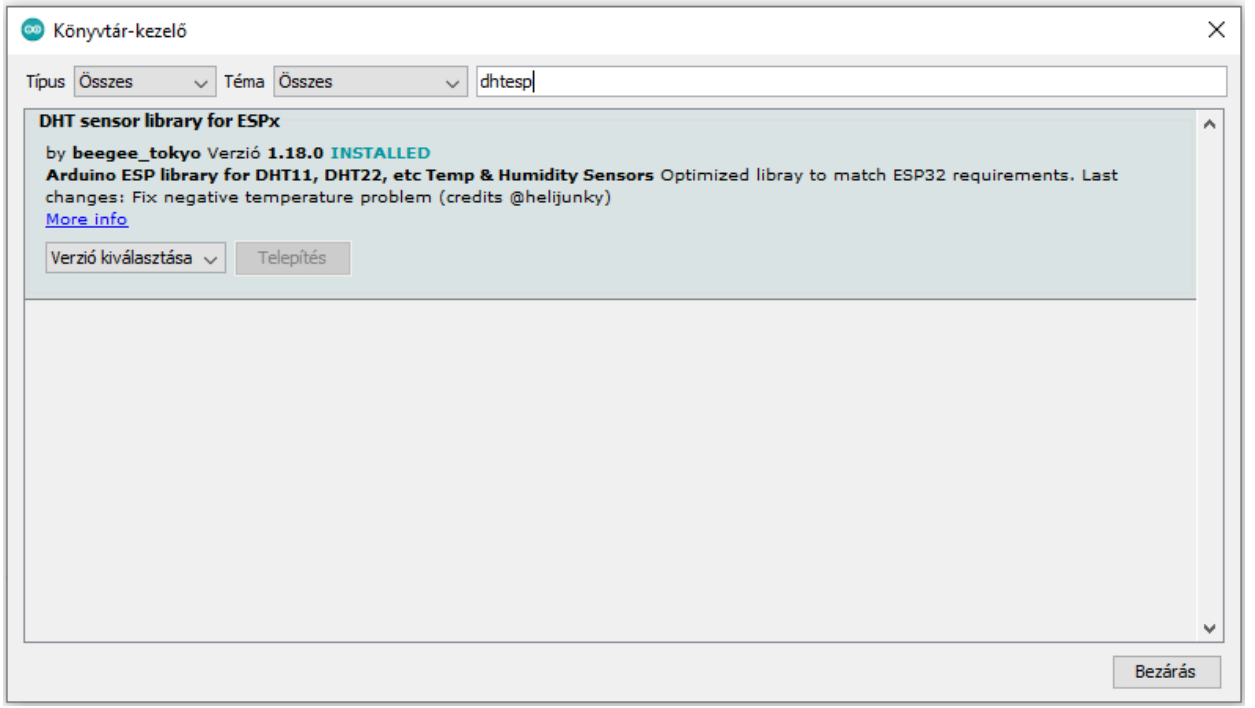


Bit data "1" bit format

https://www.hestore.hu/prod_getfile.php?id=12543

Mintaprojekt – DHT11

```
uint8_t c=0,I_RH,D_RH,I_Temp,D_Temp,CheckSum;

void Request()                          /* Microcontroller send start pulse/reques
{
        DDRD |= (1<<DHT11_PIN);
        PORTD &= ~(1<<DHT11_PIN);       /* set to low pin */
        _delay_ms(20);                  /* wait for 20ms */
        PORTD |= (1<<DHT11_PIN);        /* set to high pin */
}

void Response()                         /* receive response from DHT11 */
{
        DDRD &= ~(1<<DHT11_PIN);
        while(PIND & (1<<DHT11_PIN));
        while((PIND & (1<<DHT11_PIN))==0);
        while(PIND & (1<<DHT11_PIN));
}

uint8_t Receive_data()                  /* receive data */
{
        for (int q=0; q<8; q++)
        {
                while((PIND & (1<<DHT11_PIN)) == 0);  /* check received bit
                _delay_us(30);
                if(PIND & (1<<DHT11_PIN))/* if high pulse is greater than 30
                c = (c<<1)|(0x01);      /* then its logic HIGH */
                else                    /* otherwise its logic LOW */
                c = (c<<1);
                while(PIND & (1<<DHT11_PIN));
        }
        return c;
}
```



https://www.electronicwings.com/avr-atmega/dht11-sensor-interfacing-with-atmega16-32 (AVR C kód a logika miatt)

Mintaprojekt – DHT11

DHTesp DhtSensor1;

`void setup(uint8_t pin, DHT_MODEL_t model=AUTO_DETECT);`

- Call to initialize the interface, define the GPIO pin to which the sensor is connected and define the sensor type.
  Valid sensor types are:
  - AUTO_DETECT Try to detect which sensor is connected (default if 2nd parameter is not used)
  - DHT11
  - DHT22
  - AM2302 Packaged DHT22
  - RHT03 Equivalent to DHT22

  `void resetTimer();`
- Reset last time the sensor was read

DhtSensor1.setup(16, DHTesp::DHT11);

`float getTemperature();`

- Get the temperature in degree Centigrade from the sensor
  Either one of `getTemperature()` or `getHumidity()` or `getTempAndHumidity()` initiates reading a value from the sensor if the last reading was older than the minimal refresh time of the sensor.
  See example `DHT_ESP32.ino` or `DHT_Test.ino`

`float getHumidity();`

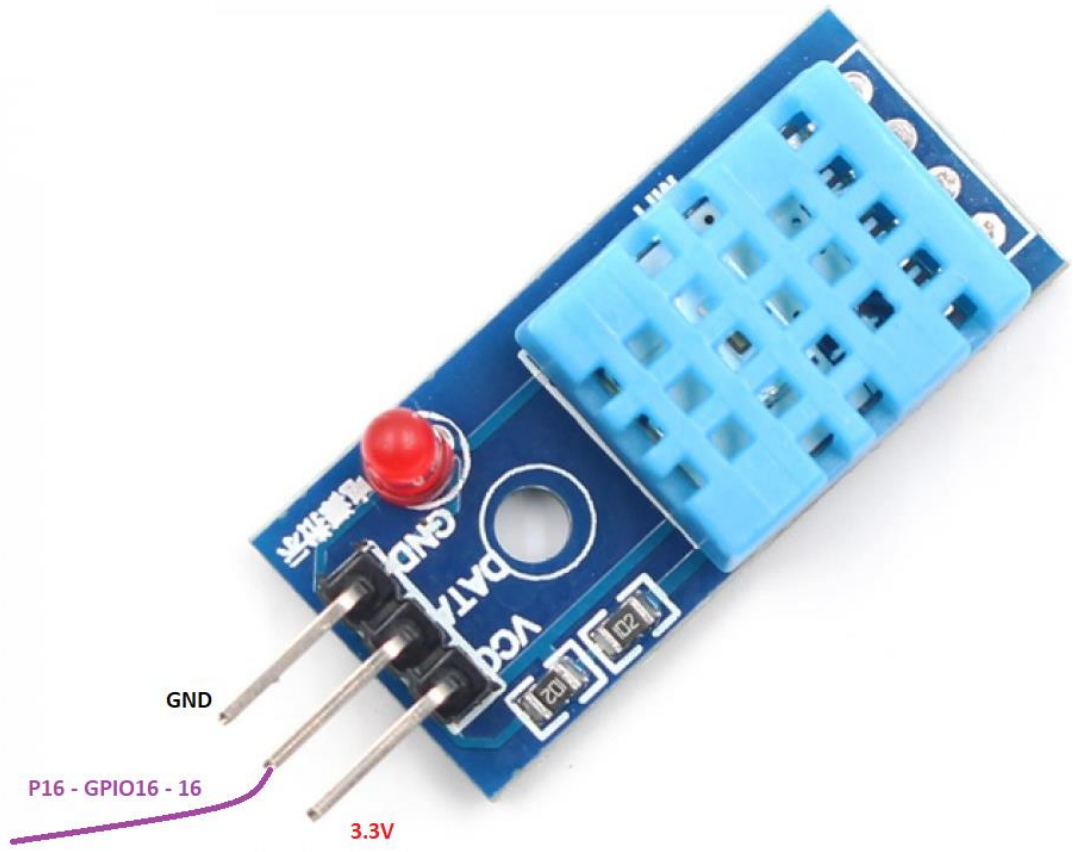
- Get the humidity from the sensor
  Either one of `getTemperature()` or `getHumidity()` or `getTempAndHumidity()` initiates reading a value from the sensor if the last reading was older than the minimal refresh time of the sensor.
  See example `DHT_ESP32.ino` or `DHT_Test.ino`

https://github.com/beegee-tokyo/DHTesp

float temp = DhtSensor1.getTemperature();
float hum = DhtSensor1.getHumidity();

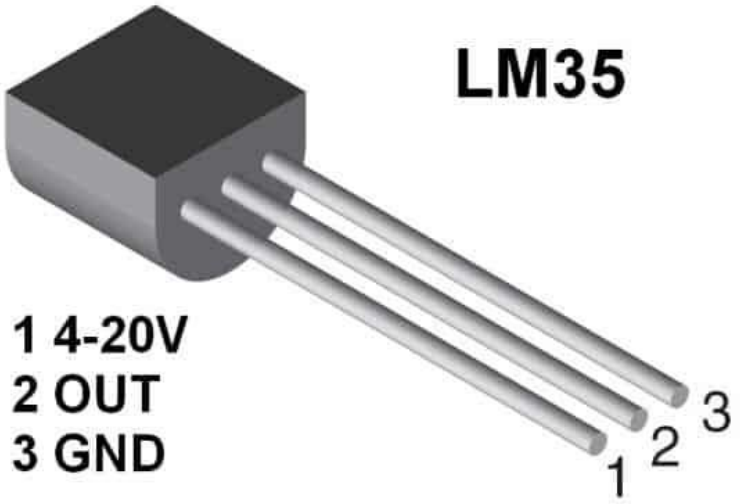
Mintaprojekt – DHT11

```
#include <Arduino.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <WiFiAP.h>
#include <WebServer.h>
#include "DHTesp.h"

DHTesp DhtSensor1;
float temp = 0;
float hum = 0;

void setup() {
  DhtSensor1.setup(16, DHTesp::DHT11);
  Serial.begin(9600);
}

void loop() {

  temp = DhtSensor1.getTemperature();
  hum = DhtSensor1.getHumidity();


  Serial.print("Temperature = ");
  Serial.println(temp);
  Serial.print("Humidity = ");
  Serial.println(hum);

  delay(2000);
}
```

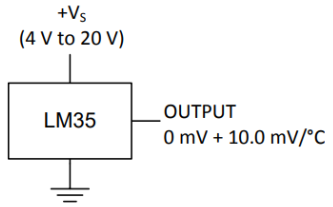


Mintaprojekt – LM35

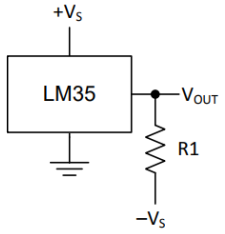
- Calibrated Directly in Celsius (Centigrade)
- Linear + 10-mV/°C Scale Factor
- 0.5°C Ensured Accuracy (at 25°C)
- Rated for Full −55°C to 150°C Range
- Operates From 4 V to 30 V



Basic Centigrade Temperature Sensor (2°C to 150°C)



Full-Range Centigrade Temperature Sensor

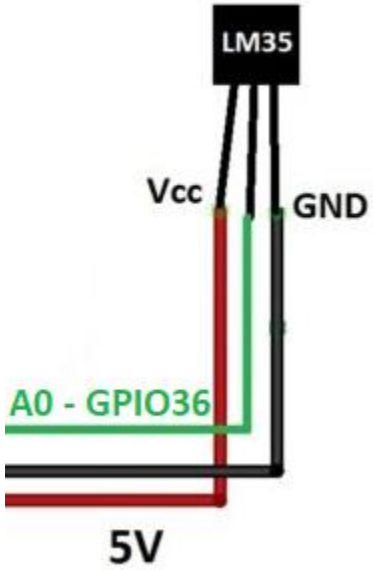


Choose $R_1 = -V_S / 50\ \mu A$
$V_{OUT} = 1500$ mV at 150°C
$V_{OUT} = 250$ mV at 25°C
$V_{OUT} = -550$ mV at −55°C

https://www.ti.com/lit/ds/symlink/lm35.pdf

Mintaprojekt – LM35

```
#include <Arduino.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <WiFiAP.h>
#include <WebServer.h>


void setup() {

  Serial.begin(9600);
}

void loop() {

  int val = analogRead(A0);    //A0 36
  float mv = val*(3300.0/4096.0);
  float cel = mv/10;
  Serial.print("Temp = ");
  Serial.println(cel);    // the raw analog reading

  delay(1000);
}
```



Nemzeti Tehetség Program

Q&A

MINISZTERELNÖKSÉG

Nemzeti Tehetség Program

Köszönöm a figyelmet!

MINISZTERELNÖKSÉG