

(v)

IoT Workshop



Piotr Bochyński

Michał Bogacz

(x)

INSTALL ARDUINO IDE

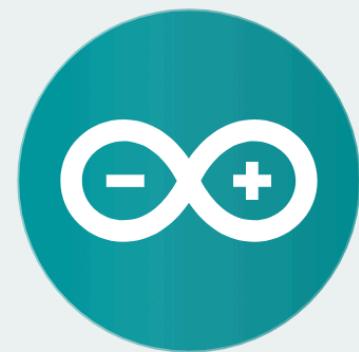
<https://www.arduino.cc/en/Main/Software>



<https://www.arduino.cc/en/Main/Software>

(x)

Download the Arduino IDE



ARDUINO 1.8.5

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

Windows Installer

Windows ZIP file for non admin install

Windows app



Mac OS X 10.7 Lion or newer

Linux 32 bits

Linux 64 bits

Linux ARM

[Release Notes](#)

[Source Code](#)

[Checksums \(sha512\)](#)

Agenda

(x)

- Introduction
- Part 1: Hardware
- Part 2: Using sensors
- Part 3: Connecting to cloud
- Conclusion



What is IoT?

(x)

“The Internet of things (IoT) is the network of physical devices, vehicles, and other items embedded with electronics, software, sensors, actuators, and network connectivity which enable these objects to collect and exchange data.”

[Source: https://en.wikipedia.org/wiki/Internet_of_things](https://en.wikipedia.org/wiki/Internet_of_things)

What we want to accomplish?

(x)

Device reads temperature and send it to cloud.

Setup (x)

- Chat: tlk.io/workshop
- Code: github.com/dshop-gliwice/code-week-iot
- Arduino IDE & Board
- Libraries
- USB-Driver

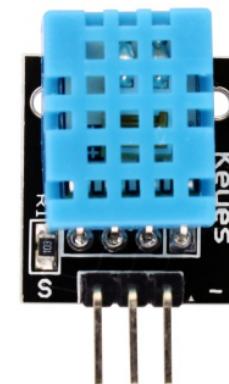
(v)

Part 1 - Hardware

- ESP8266
- DHT-11



(v)



Hardware - Core



ESP 8266



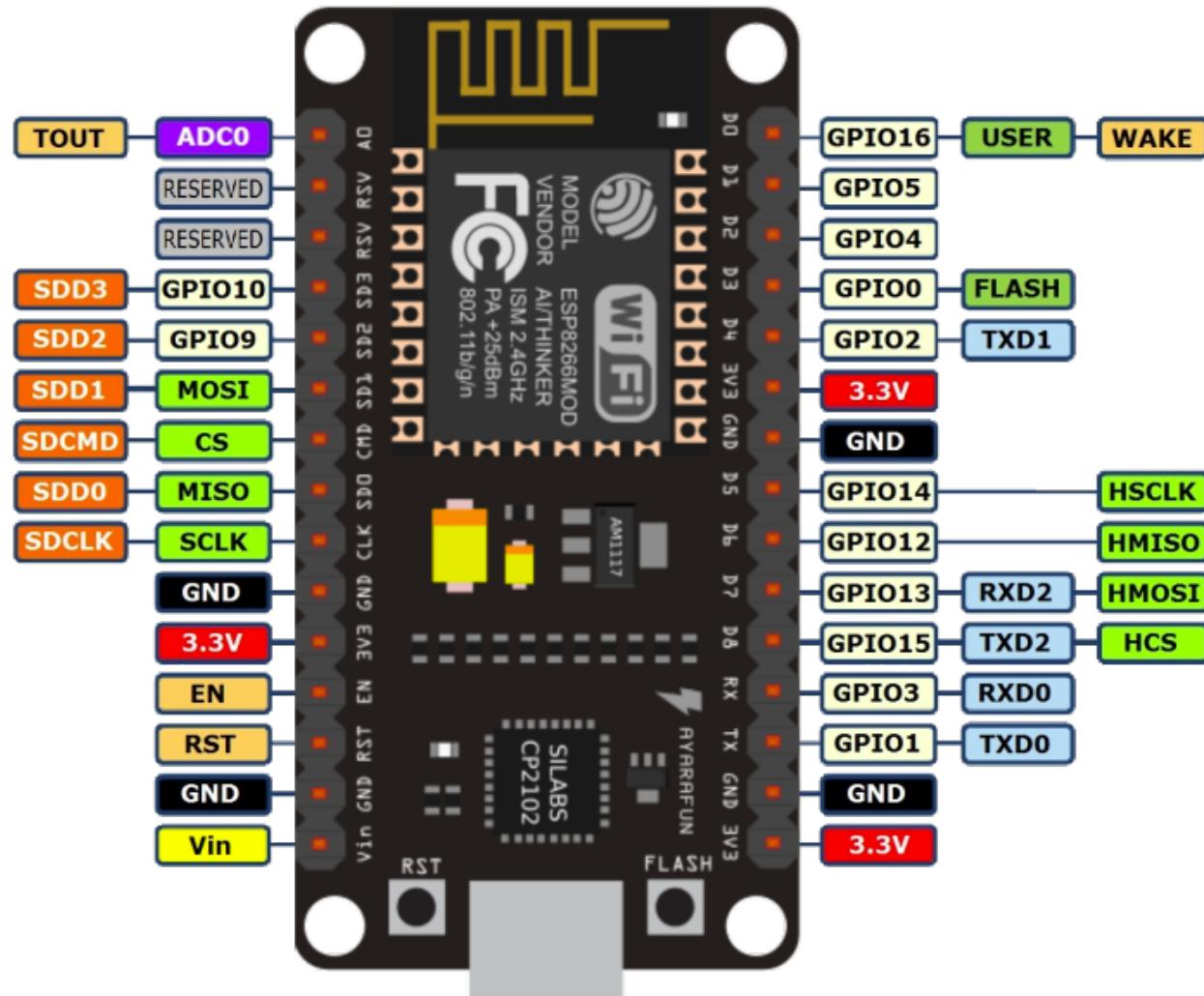
NodeMCU DEVKIT



- 32-bit RISC CPU 80 MHz
- 64 KiB of instruction RAM, 96 KiB of data RAM
- 4 MiB flash
- Wi-Fi a/b/g/n
- 16 GPIO pins
- Interfaces: SPI, I²C, UART, 10-bit ADC
- NodeMCU: a Lua-based firmware.
- **Arduino Core** - <https://github.com/esp8266/Arduino>

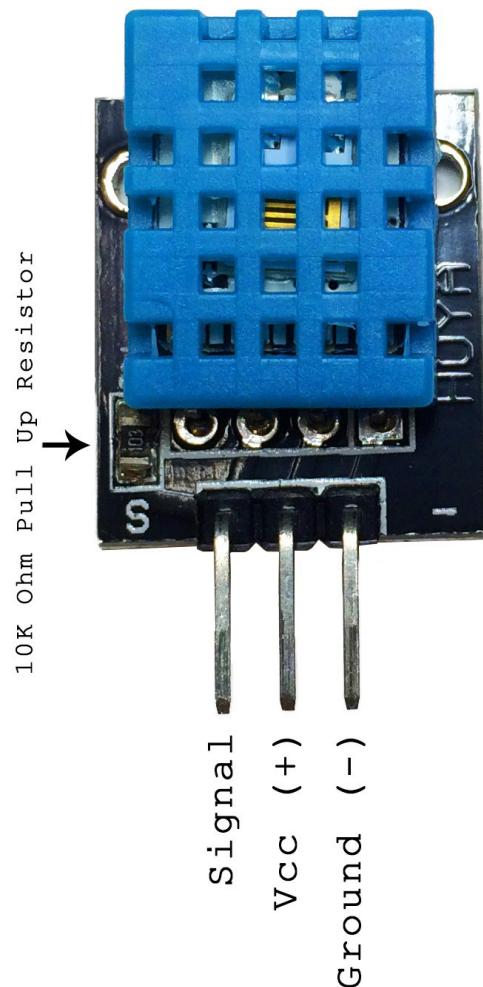
NodeMCU devkit – pinout

(v)



Sensor - DHT11

(v)



- **Interface – 1wire**

- **Temperature**

- Measurement range: 0 - 50 °C
- Accuracy: 1 °C

- **Humidity:**

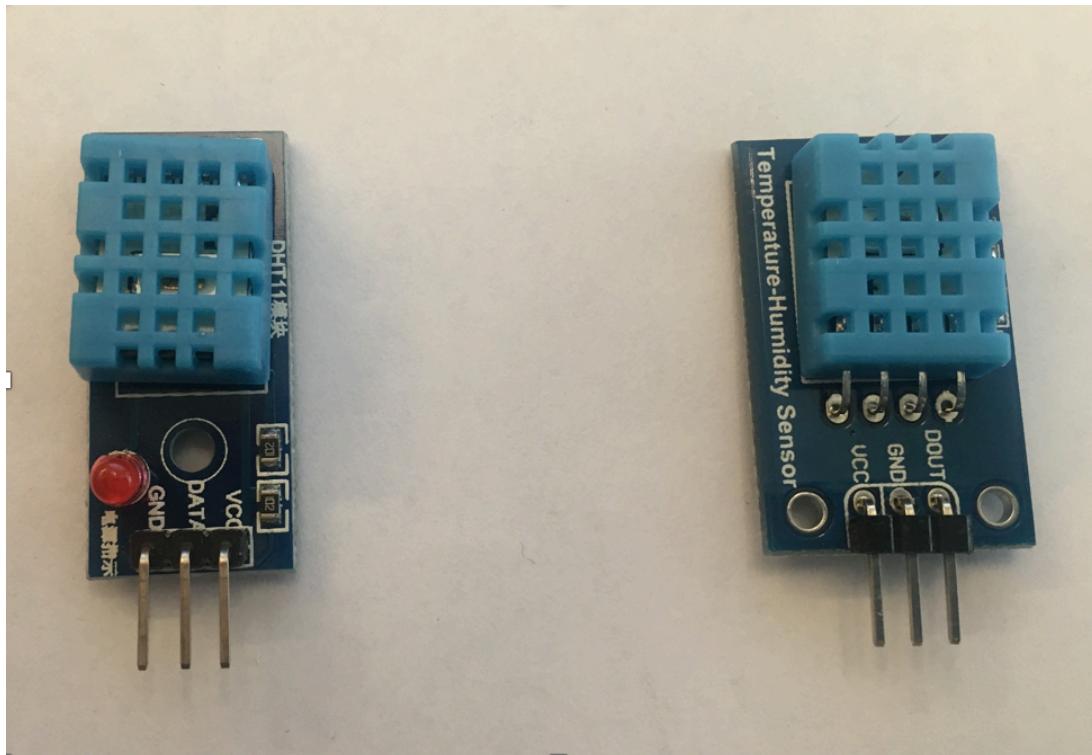
- Measurement range: 20 - 90 %RH
- Accuracy ± 4 RH* (at 25 °C)

Need more?

Use **DS18B20, DHT22, BME280**

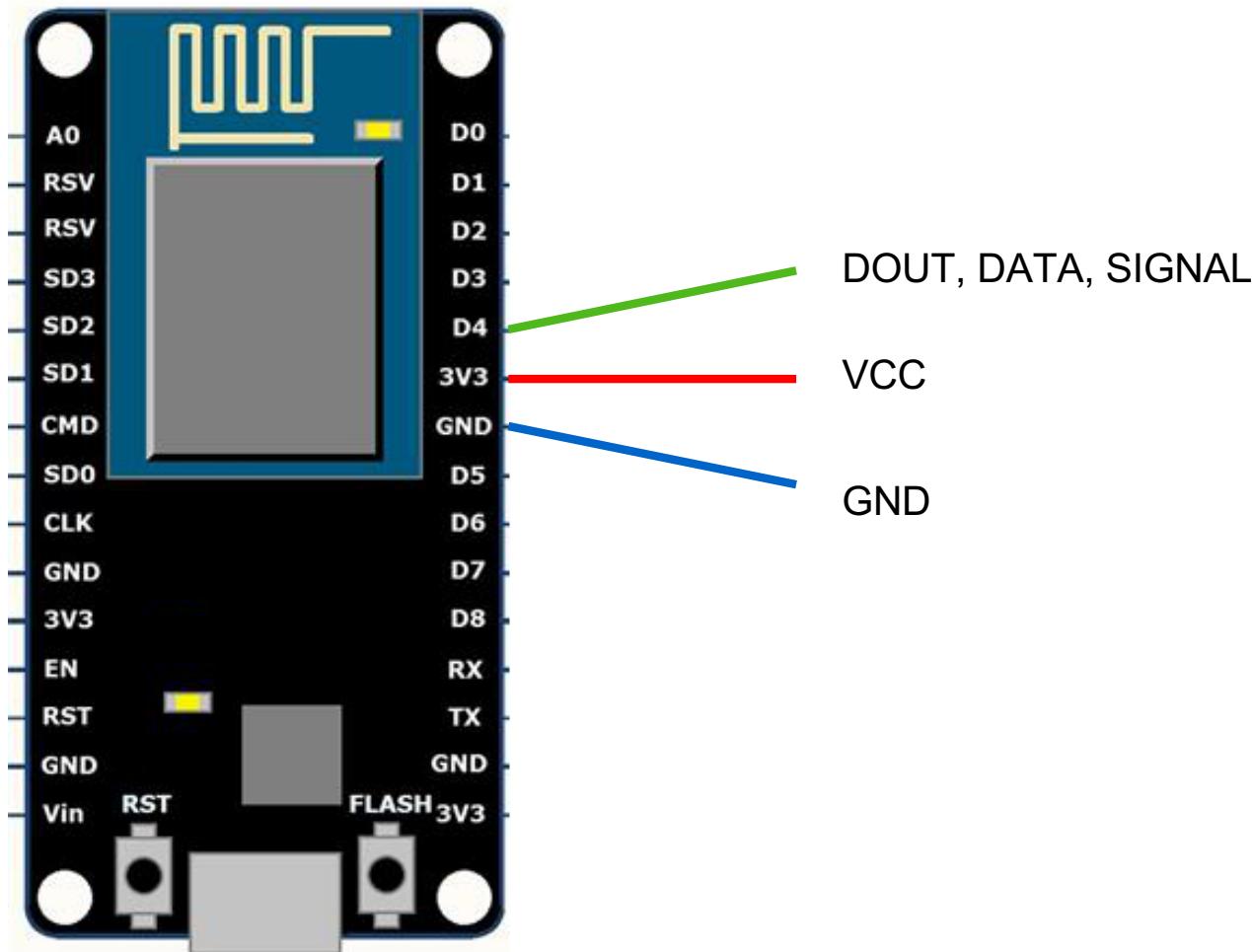
Sensor - pinout

(v)

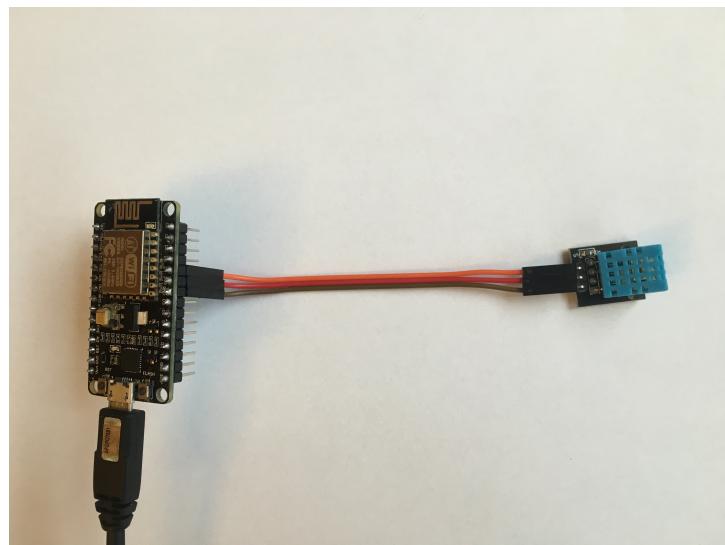


Wiring

(v)



Wiring - example



(x)

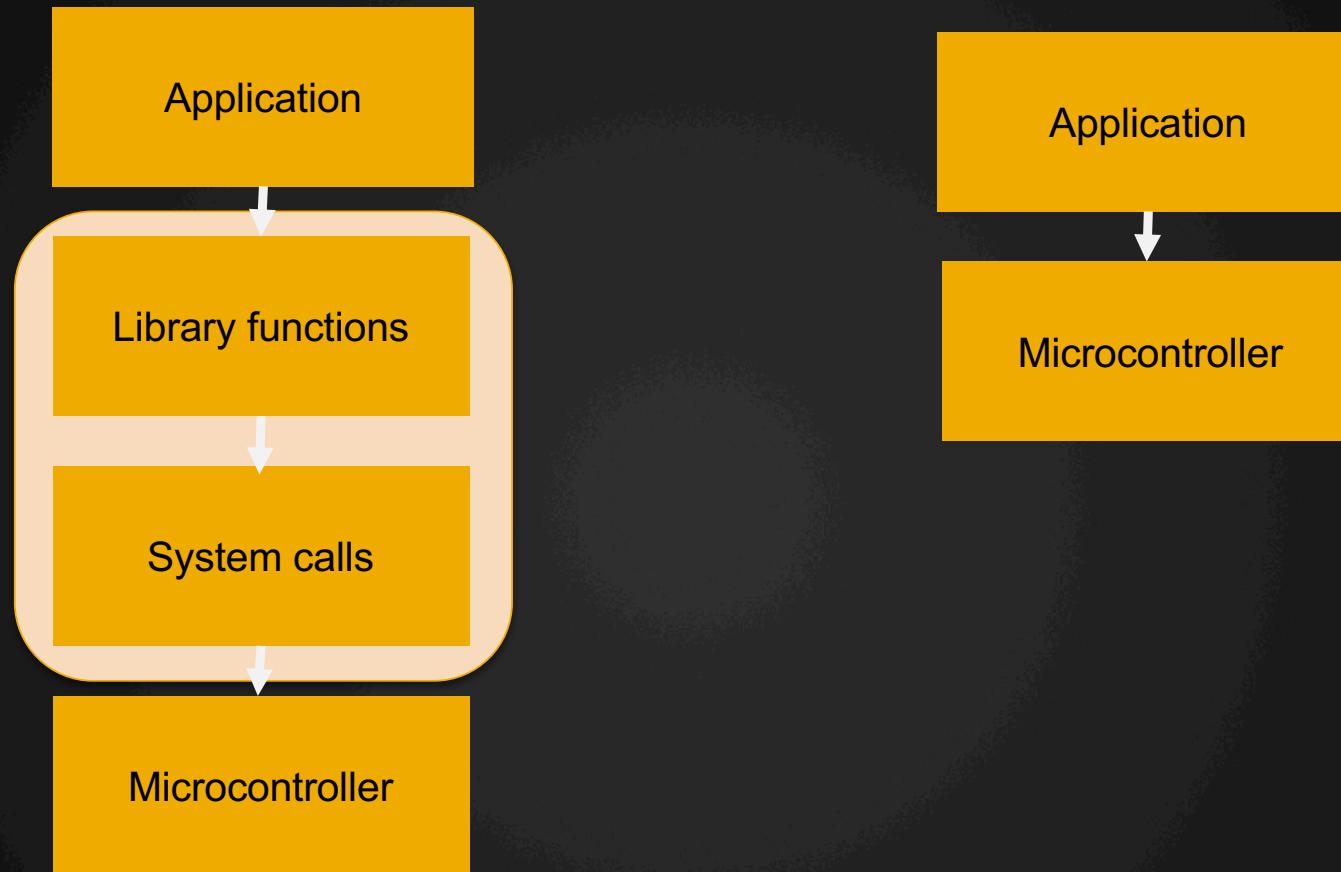
Part 2 - Using sensors

- Introduction to Arduino
- Examples LED + DHT sensor
- Task



Forget about Operating System

(x)



Arduino

(x)



The screenshot shows the Arduino IDE interface with the title bar "blink | Arduino 1.6.11". The main window displays the "blink" sketch code. The code is as follows:

```
/*
 * Blink
 * Turns on the onboard LED on for one second, then off for one second, repeatedly.
 * This uses delay() to pause between LED toggles.
 */

void setup() {
    pinMode(BUILTIN_LED, OUTPUT); // initialize onboard LED as output
}

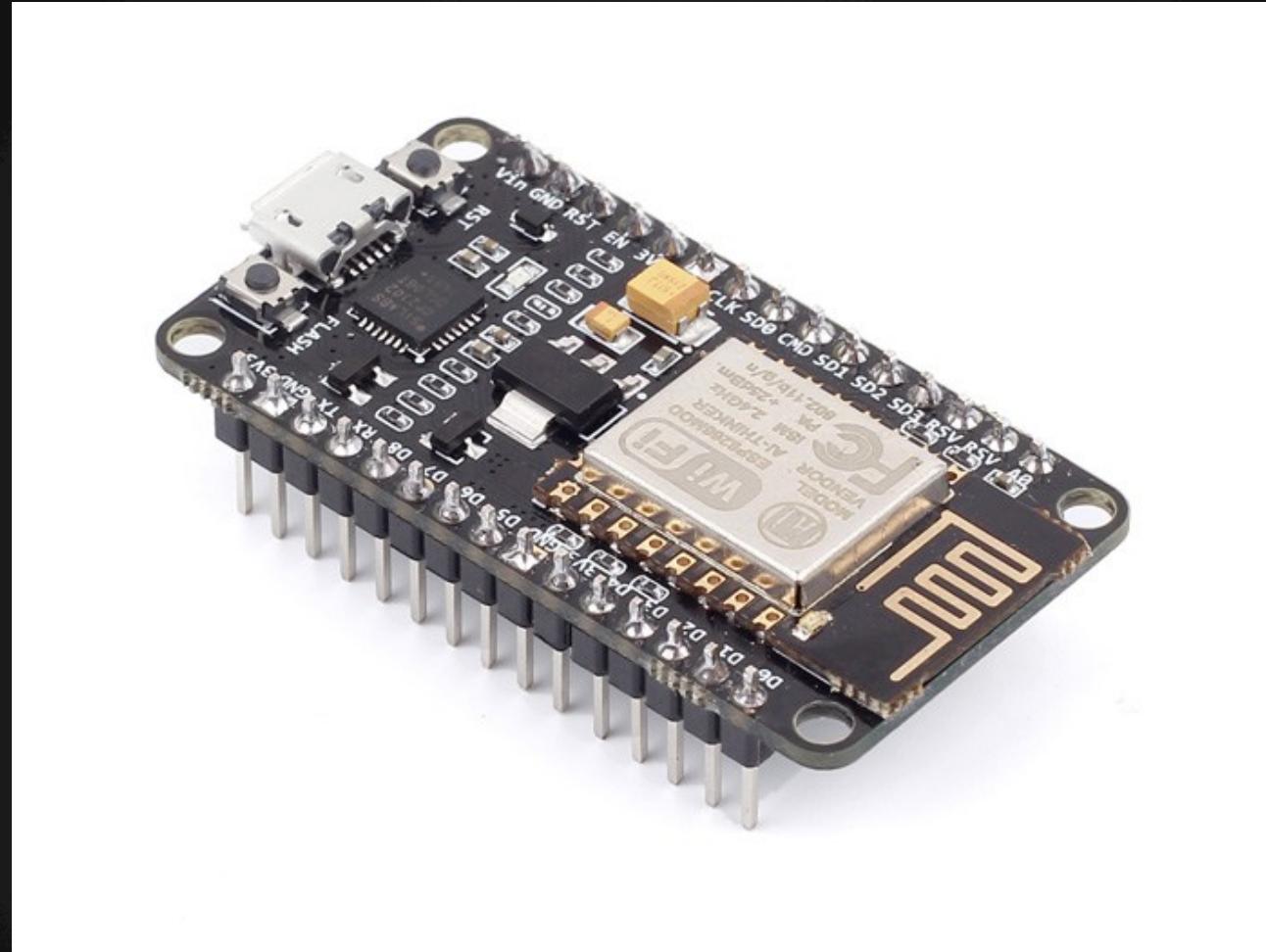
void loop() {
    digitalWrite(BUILTIN_LED, HIGH); // turn on LED with voltage HIGH
    delay(1000); // wait one second
    digitalWrite(BUILTIN_LED, LOW); // turn off LED with voltage LOW
    delay(1000); // wait one second
}
```

At the bottom of the IDE, the status bar shows the number "5" and the text "NodeMCU 1.0 (ESP-12E Module), 80 MHz, 115200, 4M (1M SPIFFS) on /dev/cu.usbmodem1884371".



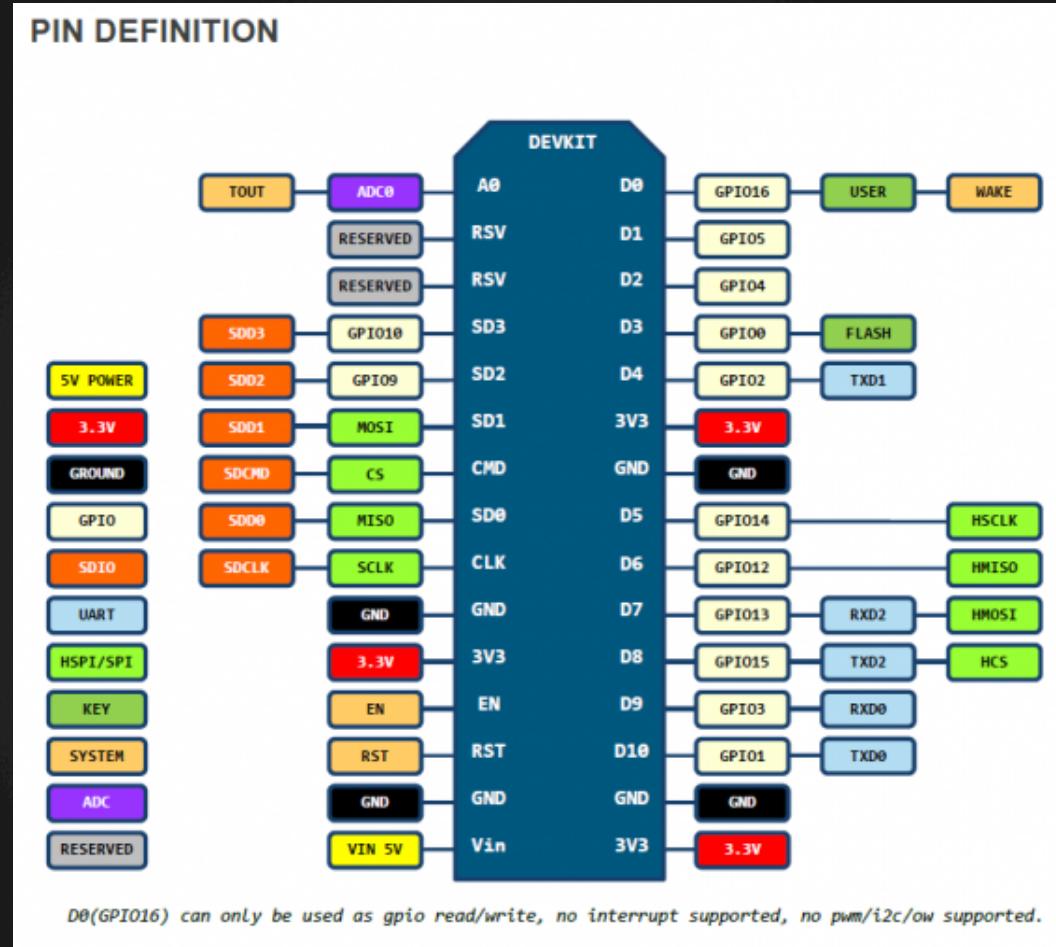
Focus on pins

(y)



Focus on pins

(y)



Arduino IDE

Compile

Compile+

Upload

Serial Monitor

(x)

blink | Arduino 1.6.11

```
/*
 * Blink
 * Turns on the onboard LED on for one second, then off for one second, repeatedly.
 * This uses delay() to pause between LED toggles.
 */

void setup() {
    pinMode(BUILTIN_LED, OUTPUT); // initialize onboard LED as output
}

void loop() {
    digitalWrite(BUILTIN_LED, HIGH); // turn on LED with voltage HIGH
    delay(1000); // wait one second
    digitalWrite(BUILTIN_LED, LOW); // turn off LED with voltage LOW
    delay(1000); // wait one second
}
```

5 NodeMCU 1.0 (ESP-12E Module), 80 MHz, 115200, 4M (1M SPIFFS) on /dev/cu.usbmodem1884371

Arduino Code

(x)

setup()

loop()

delay(ms)

```
/*
 * Blink
 * Turns on the onboard LED on for one second, then off for one second, repeatedly.
 * This uses delay() to pause between LED toggles.
 */

void setup() {
  pinMode(BUILTIN_LED, OUTPUT); // initialize onboard LED as output
}

void loop() {
  digitalWrite(BUILTIN_LED, HIGH); // turn on LED with voltage HIGH
  delay(1000); // wait one second
  digitalWrite(BUILTIN_LED, LOW); // turn off LED with voltage LOW
  delay(1000); // wait one second
}
```

5 NodeMCU 1.0 (ESP-12E Module), 80 MHz, 115200, 4M (1M SPIFFS) on /dev/cu.usbmodem1884371

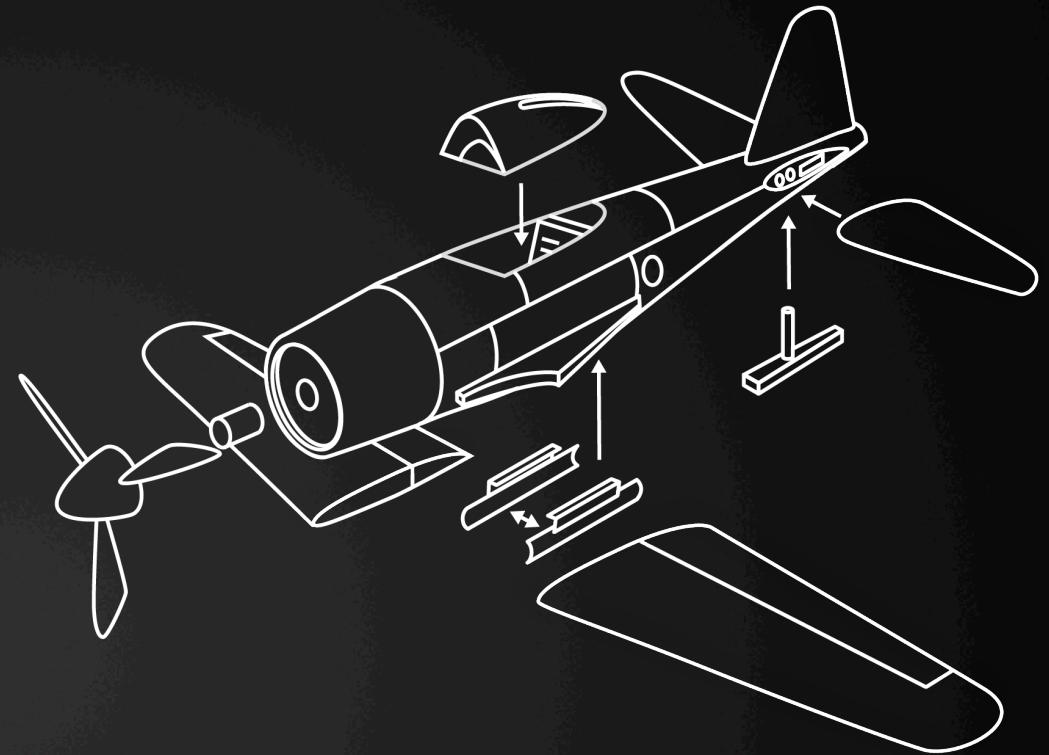
Device configuration

Tools:

- Board: NodeMCU 1.0 (ESP-12E Module)
- Flash Size: 4M (1M SPIFFS)
- CPU Frequency: 160 MHz
- Upload Speed: 115200
- Port: /dev/cu.SLAB_USBtoUART / COM?

Example

Blink LED



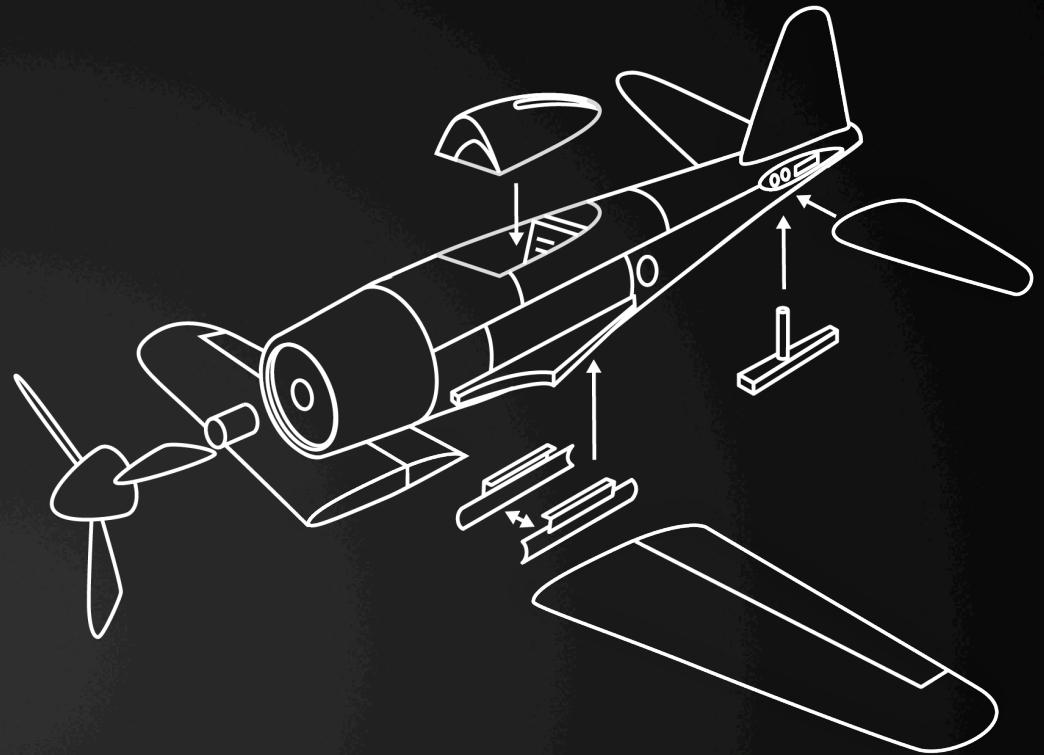
step2 -> blink

Example

Temperature and
humidity sensor

DHT-11

step2 -> dht-sensor



Task

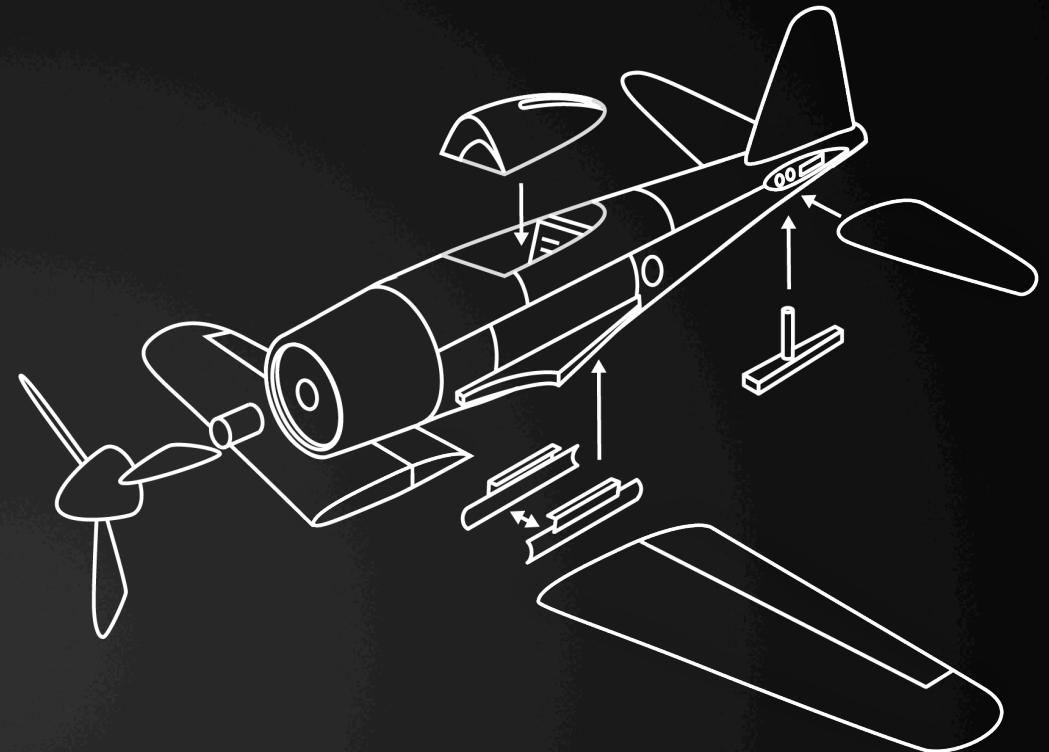
Turn on LED

when temperature

is over

degrees 28 Celsius

(turn off when lower)



(x)

Part 3 – Connecting to the cloud



WIFI connection



HTTP requests



A pinch of security

WIFI connection

```
WiFi.mode(WIFI_STA);  
WiFi.begin(ssid, password);  
while (WiFi.status() != WL_CONNECTED) {  
    delay(250);  
    Serial.print(".");  
}  
Serial.println("READY");
```



(x)



thingspeak.com



codeweek

(x)

Channel ID: 136124

codeweek iot workshop

Author: pibipi

Access: Public

Private View

Public View

Channel Settings

Sharing

API Keys

Channel Settings

Percentage complete 50%

Channel ID 136124

Name codeweek

Description codeweek iot workshop

Field 1 t

Field 2 h

SAP®



codeweek

Channel ID: 136124

codeweek iot workshop

Author: pibipi

Access: Public

Private View

Public View

Channel Settings

Sharing

API Keys

Data Import / Export

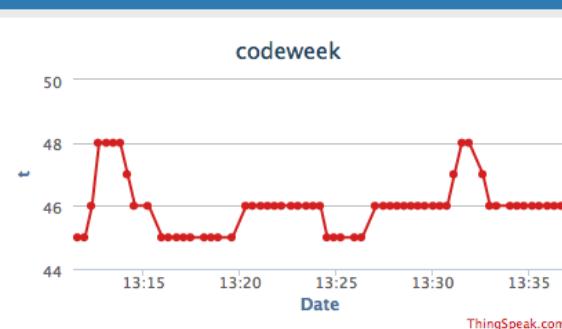
[+ Add Visualizations](#)[Data Export](#)[MATLAB Analysis](#)[MATLAB Visualization](#)

Channel Stats

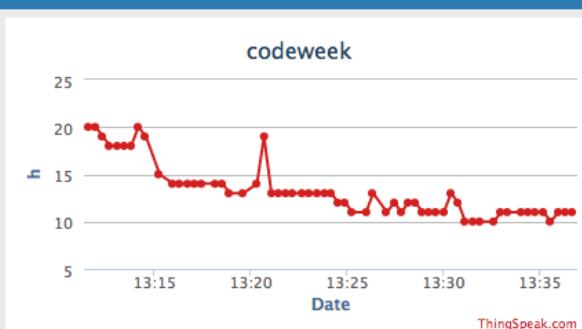
Created: [about a year ago](#)Updated: [less than a minute ago](#)Last entry: [less than a minute ago](#)

Entries: 208

Field 1 Chart



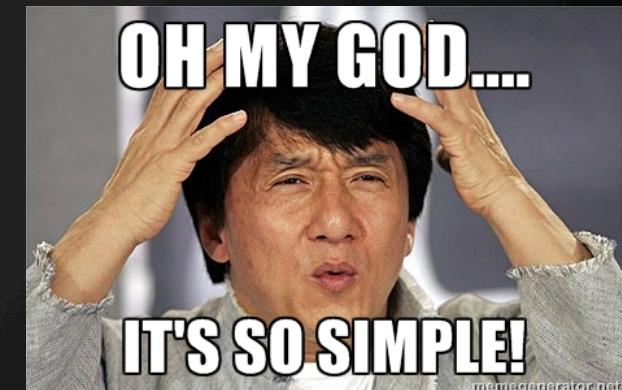
Field 2 Chart



HTTPS request

(x)

```
void updateChannel(float t, float h) {  
    HTTPClient client;  
    client.begin("https://api.thingspeak.com/update?api_key="  
        + API_KEY + "&field1=" + String(t) + "&field2=" + String(h),  
        "78 60 18 44 81 35 BF DF 77 84 D4 0A 22 0D 9B 4E 6C DC 57 2C");  
    // SHA-1 Fingerprint  
  
    int httpCode = client.GET();  
    Serial.println("Request sent");  
  
    if (httpCode == 200) {  
        Serial.print("OK");  
    }  
    client.end();  
}
```



What's the magic string?



(x)

DigiCert High Assurance EV Root CA
↳ DigiCert SHA2 High Assurance Server CA
↳ *thingspeak.com

Extension Authority Key Identifier (2.5.29.35)
Critical NO
Key ID 51 68 FF 90 AF 02 07 75 3C CC D9 65 64 62 A2 12 B8 59 72 3B

Extension Subject Alternative Name (2.5.29.17)
Critical NO
DNS Name *.thingspeak.com
DNS Name thingspeak.com

Extension Certificate Policies (2.5.29.32)
Critical NO
Policy ID #1 (2.16.840.1.114412.1.1)
Qualifier ID #1 Certification Practice Statement (1.3.6.1.5.5.7.2.1)
CPS URI <https://www.digicert.com/CPS>

Extension CRL Distribution Points (2.5.29.31)
Critical NO
URI <http://crl3.digicert.com/sha2-ha-server-q4.crl>
URI <http://crl4.digicert.com/sha2-ha-server-q4.crl>

Extension Certificate Authority Information Access (1.3.6.1.5.5.7.1.1)
Critical NO
Method #1 Online Certificate Status Protocol (1.3.6.1.5.5.7.48.1)
URI <http://ocsp.digicert.com>
Method #2 CA Issuers (1.3.6.1.5.5.7.48.2)
URI <http://cacerts.digicert.com/DigiCertSHA2HighAssuranceServerCA.crt>

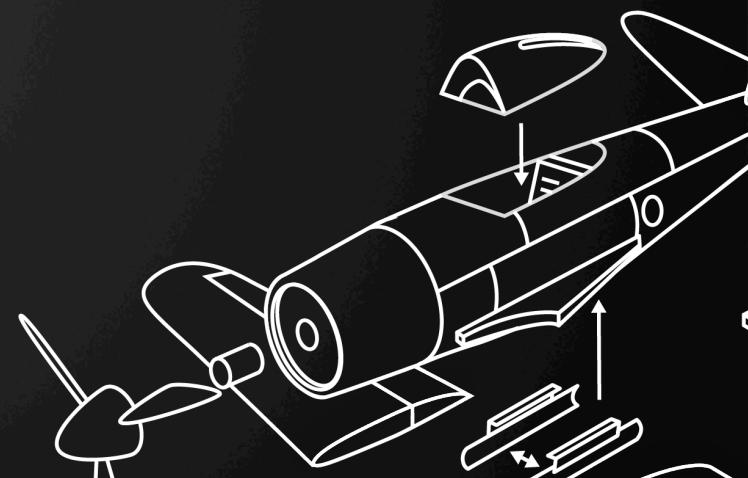
Fingerprints

SHA-256 74 13 12 8D 0B 7D 1E C8 11 0F 50 4F 34 00 19 75 A0 89 7B 12 70 A 3D 45 42 E3 AC 5E 81 77 8C
SHA-1 78 60 18 44 81 35 BF DF 77 84 D4 0A 22 0D 9B 4E 6C DC 57 2C

OK

Conclusion

- Arduino code for ESP8266 <https://github.com/esp8266/Arduino>
- Hackig at SAP Hybris <http://hackingat.hybris.com>
- Andreas Spiess YouTube channel:
https://www.youtube.com/channel/UCu7_D0o48KbfhpEohoP7YSQ
- IFTTT <https://ifttt.com/>
- Alternative to Arduino <https://mongoose-os.com>
- ESP32 <http://espressif.com/en/products/hardware/esp32/overview>



(Y)
THANK YOU!