

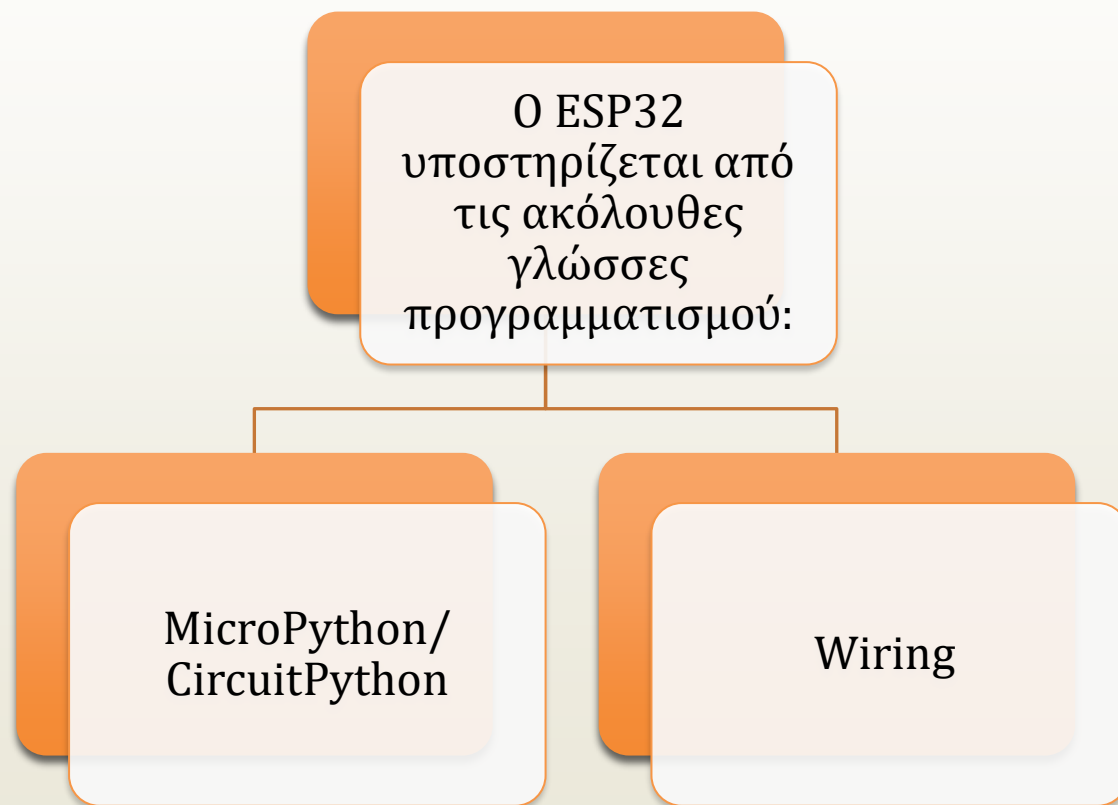


# Η οικογένεια μικροελεγκτών ESP32

- Ανήκει στο portfolio της εταιρίας ESPRESSIF
- Απόγονος της οικογένειας μικροελεγκτών ESP8266
- Βασικά χαρακτηριστικά
  - Δυνατότητα διασύνδεσης μέσω των ενσωματωμένων Wi-Fi και Bluetooth
  - Λειτουργίες χαμηλής κατανάλωσης
  - Υποστήριξη πολλών περιφερειακών
  - Υποστήριξη πολλών πρωτοκόλλων ασφάλειας
- Συμβατός με πολλά Arduino examples
- Δυνατότητα υλοποίησης πολλών εφαρμογών
- Μεγάλο community support



# Προγραμματισμός ESP32



# Το μοντέλο ESP32-C3- DevKitC-02

Ολοκληρωμένο  
υποσύστημα Wi-Fi

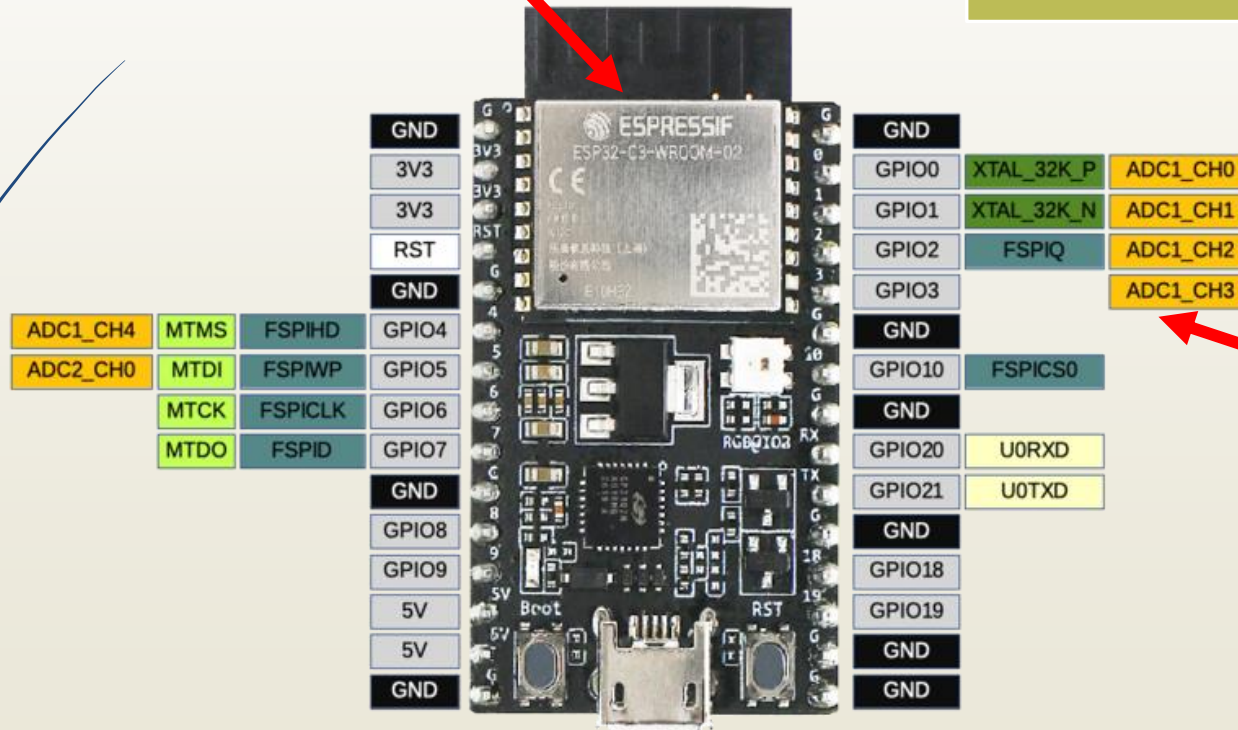
Υποσύστημα  
Bluetooth Low  
Energy και Bluetooth  
Mesh

32-bit αρχιτεκτονική  
RISC-V single core  
processor (up to 160  
MHz)

ESP32-C3-WROOM-02

Χωρητικότητα  
αποθήκευσης  
(400KB SRAM,  
384KB ROM)

Υποστήριξη πολλών  
περιφερειακών  
διεπαφών



Μερικά pin έχουν  
περισσότερες από μια  
λειτουργίες!



# Arduino IDE

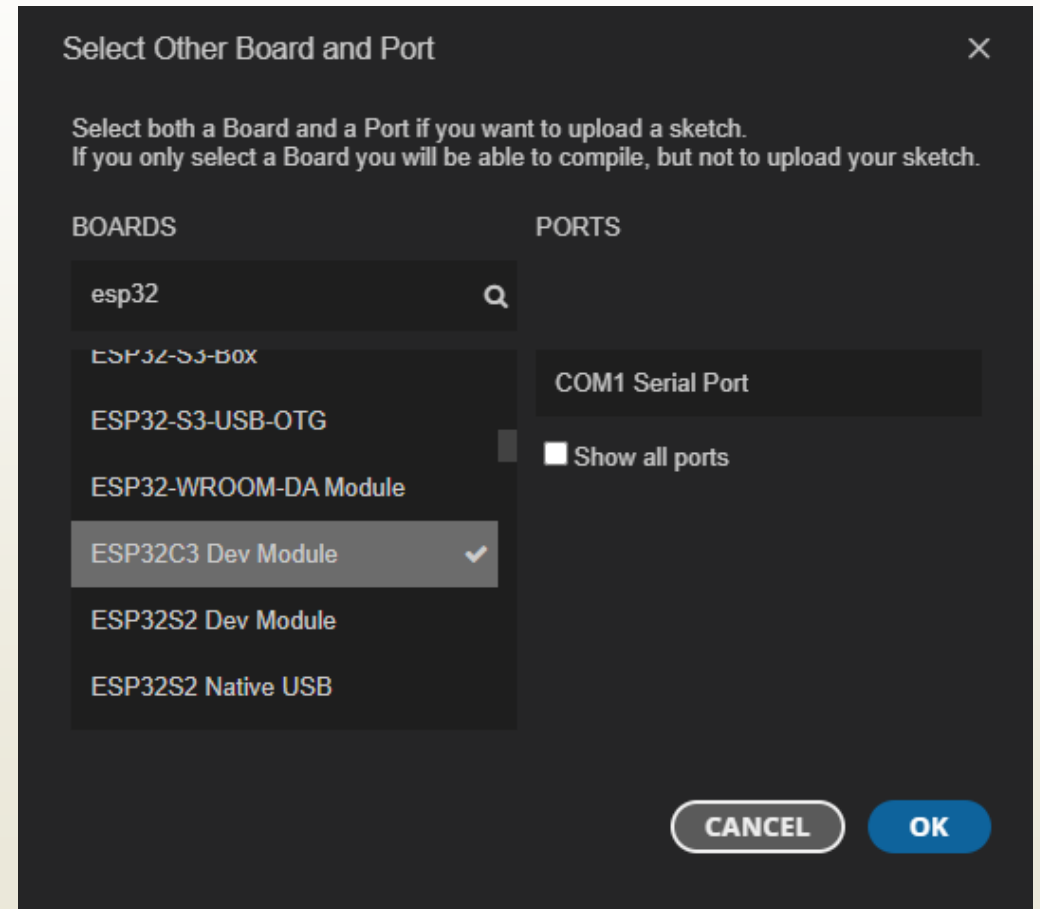
Οι οικογένεια ESP32 είναι συμβατή με το περιβάλλον Arduino IDE και υποστηρίζεται όπως και η σειρά ESP8266.

Κατά τη διεξαγωγή των εργαστηρίων θα χρησιμοποιηθεί η έκδοση του Arduino 2.0 (ή νεότερη)



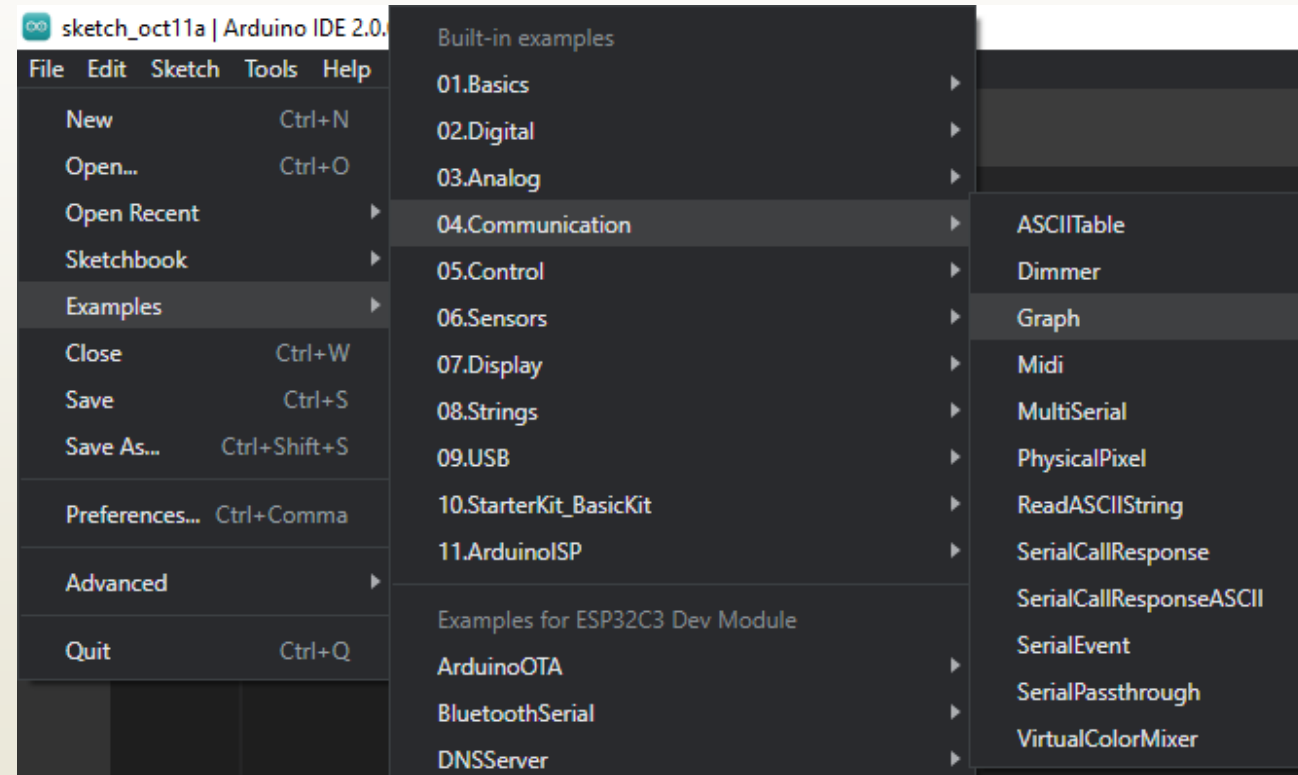
# Ανέβασμα ενός κώδικά (1/2)

Επιλέγουμε το μοντέλο του μικροελεγκτή καθώς και τη θύρα στην οποία είναι συνδεδεμένος



# Ανέβασμα ενός κώδικα (2/2)

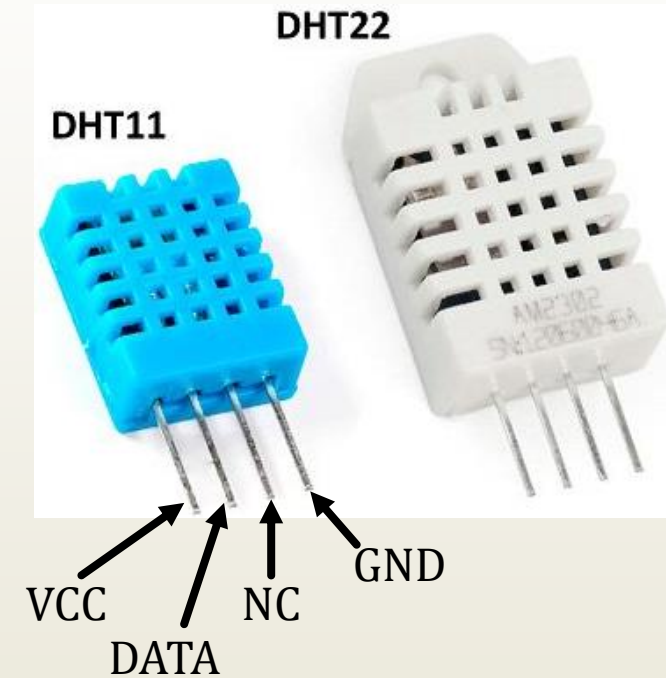
- Αφού ολοκληρωθεί η ανάπτυξη του κώδικα ή δοκιμάζοντας κάποιο από τα έτοιμα παραδείγματα όπως το File-> Examples->04.Communication-> Graph
- Πατώντας το κουμπί «Upload», ο κώδικας μεταγλωττίζεται και μεταφέρεται “ανεβαίνει” στο μικροελεγκτή



# Ο Αισθητήρας DHT11 / DHT22

- ▶ Ψηφιακός αισθητήρας
- ▶ Χαμηλού κόστους
- ▶ Μετρά θερμοκρασία και υγρασία

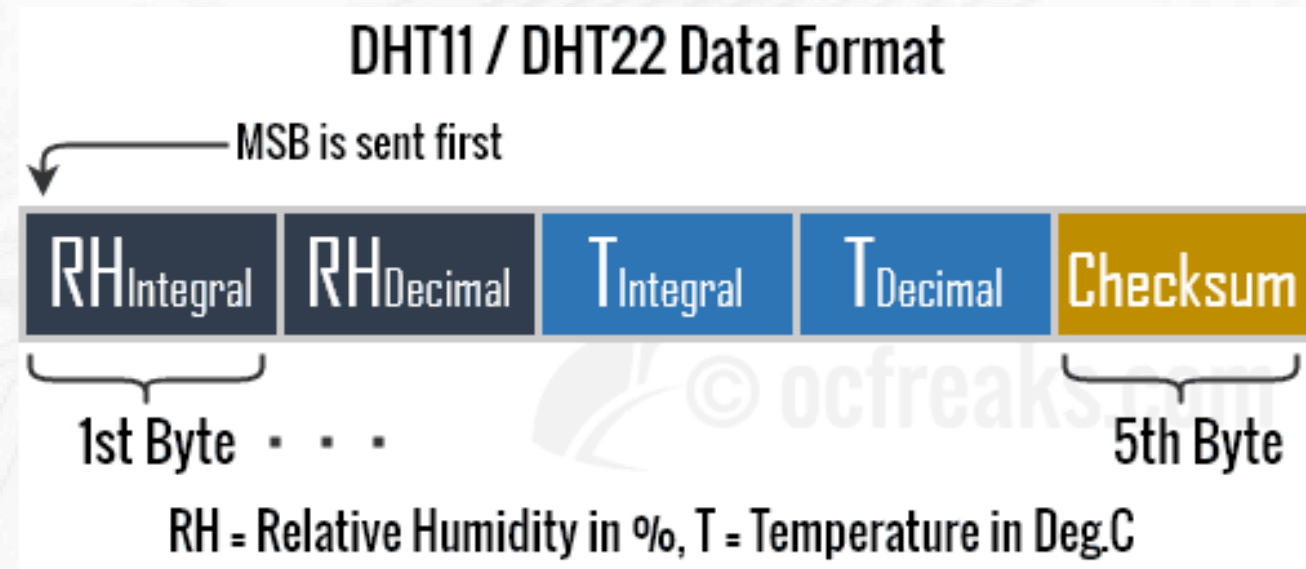
| Parameter            | DHT11      | DHT22       |
|----------------------|------------|-------------|
| Temperature Range    | 0 to 50°C  | -40 to 80°C |
| Temperature accuracy | ±1         | ±0.5        |
| Humidity range       | 30% to 90% | 0% to 100%  |
| Humidity accuracy    | ±4%        | ±2%         |
| Operating voltage    | 3.3V to 5V | 3.3V to 5V  |
| Resolution           | 8-bit      | 16-bit      |
| Sampling period      | ≥1sec      | ≥2sec       |



# DHTxx

## Dedicated library

- Οι αισθητήρες της οικογένειας DHT χρησιμοποιούν ένα dedicated πρωτόκολλο σε αντίθεση με τα κοινά I<sup>2</sup>C, SPI ή OneWire



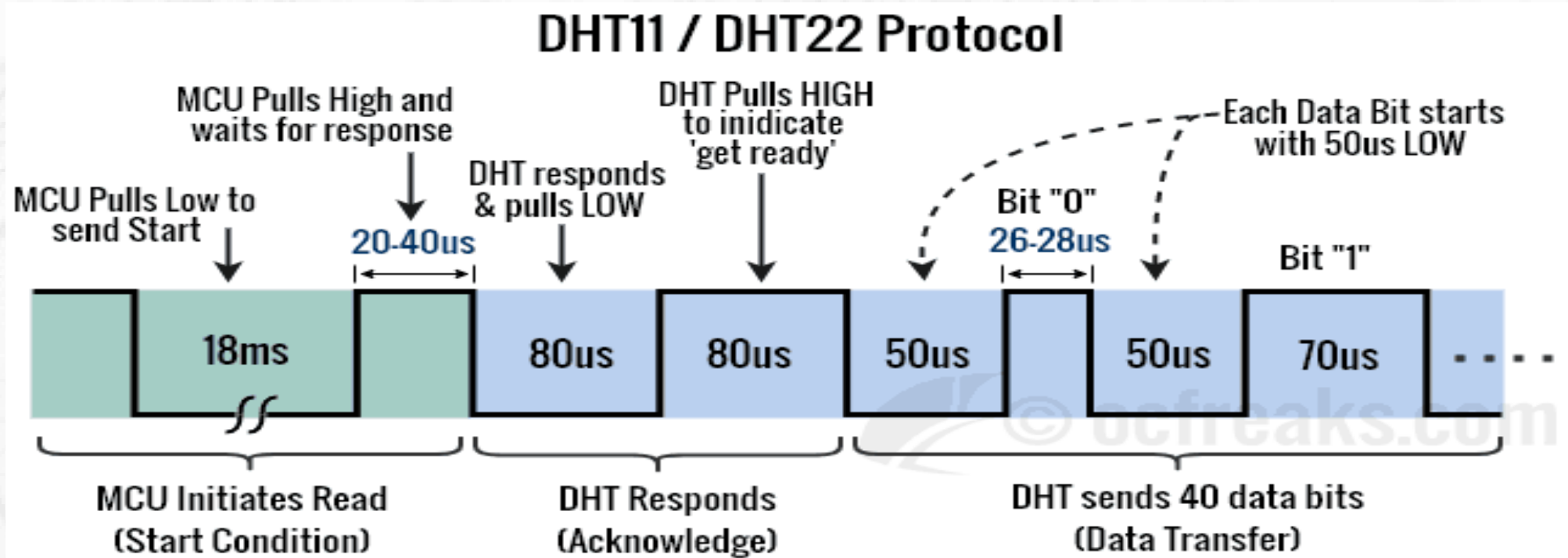
- 1st Byte: Relative Humidity Integral Data in % (Integer Part)
- 2nd Byte: Relative Humidity Decimal Data in % (Fractional Part) – Zero for DHT11
- 3rd Byte: Temperature Integral in Degree Celsius (Integer Part)
- 4th Byte: Temperature in Decimal Data in % (Fractional Part) – Zero for DHT11
- 5th Byte: Checksum (Last 8 bits of {1st Byte + 2nd Byte + 3rd Byte+ 4th Byte})



# DHT protocol

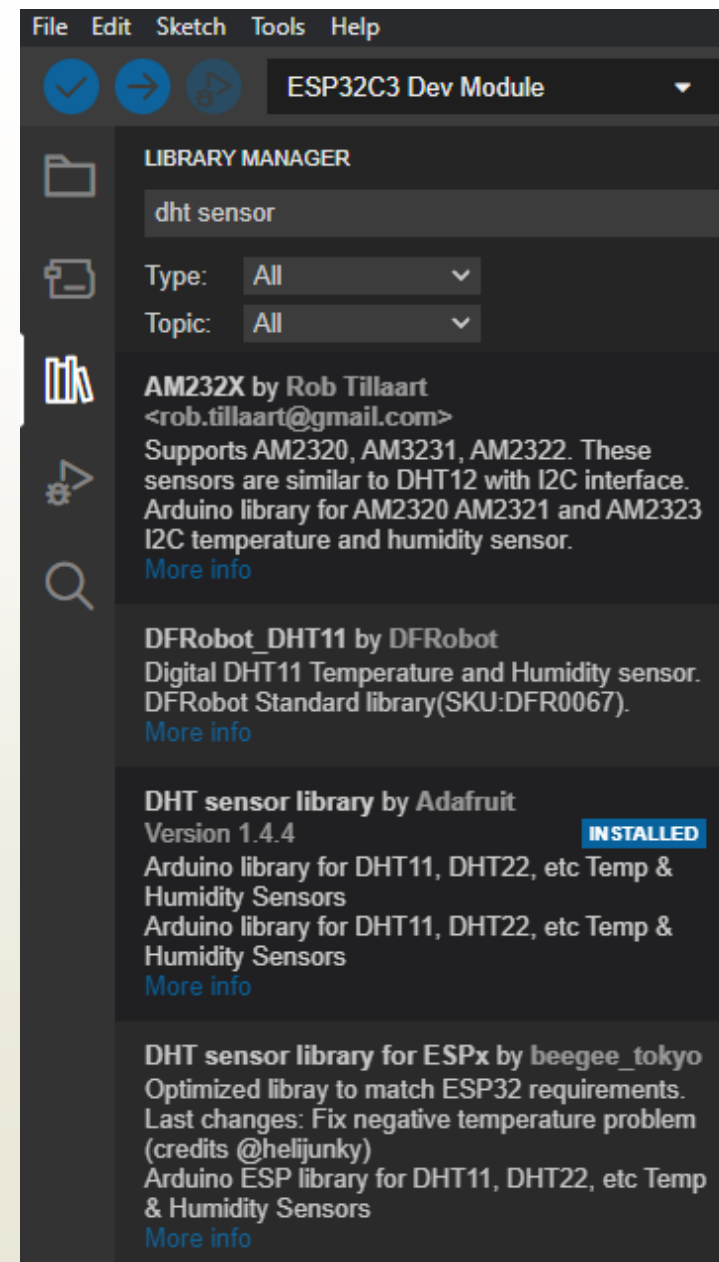
## Communication Process:

1. When the Line is IDLE the microcontroller pulls it to a LOW for 18ms.
2. After this MCU pulls it HIGH for around 20 to 40 $\mu$ s.
3. DHTxx will detect it as a START from the MCU and responds by pulling the line LOW for 80 $\mu$ s.
4. Next, DHTxx will pull it HIGH for 80 $\mu$ s which indicates that it is ready to send data or "get ready".
5. Next it will send 40 bits of Data. Each bit starts with a 50 $\mu$ s LOW followed by 26-28 $\mu$ s for a "0" or 70 $\mu$ s for a "1".
6. After communication ends, the Line is pulled HIGH by the pull-up resistor and enters IDLE state.



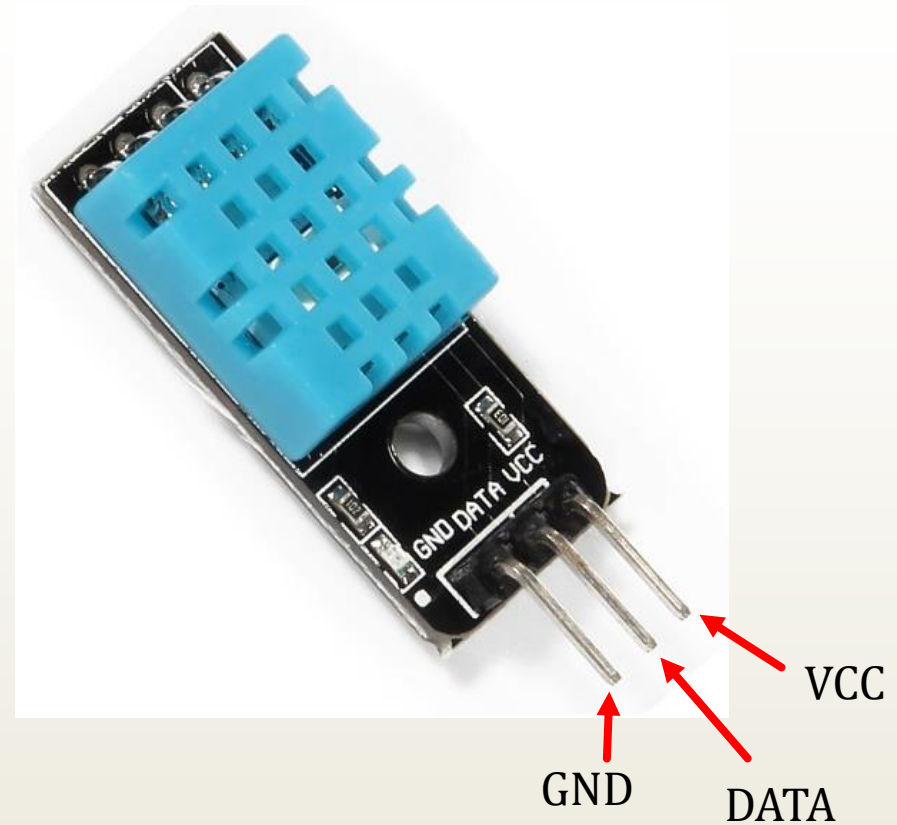
# Εγκατάσταση της βιβλιοθήκης

- ▶ Για να χρησιμοποιηθεί ο αισθητήρας χρειάζεται να γίνει εγκατάσταση της βιβλιοθήκης του
- ▶ Επιλέγοντας **Tools > Manage Libraries...** πληκτρολογούμε “dht sensor”. Από τις διαθέσιμες βιβλιοθήκες επιλέγουμε την επίσημη βιβλιοθήκη που παρέχει η εταιρία Adafruit, όπως και την επίσημη βιβλιοθήκη της Adafruit Unified Sensor



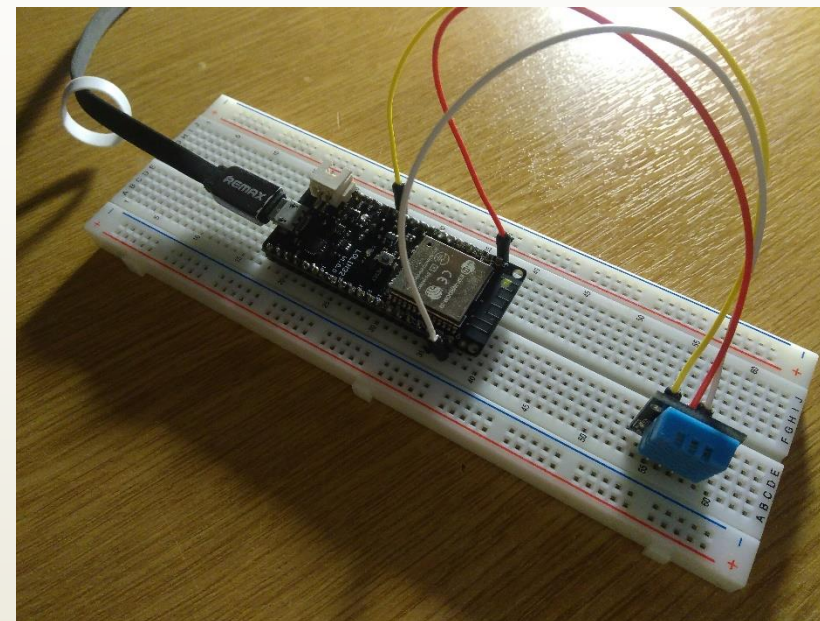
# DHT11 Σύνδεση σε Breadboard

- Προσέχουμε κατά την τοποθέτηση του αισθητήρα στο breadboard, το GND pin να βρίσκεται πάντα στην αριστερή πλευρά.
- Όλες οι συνδέσεις γίνονται με το μικροελεγκτή εκτός τροφοδοσίας για την αποφυγή βραχυκυκλώματος!
- Όταν βεβαιωθούμε ότι έχουμε συνδέσει σωστά τον αισθητήρα, τροφοδοτούμε τον μικροελεγκτή.



# Σύνδεση ενός ESP32 με τον αισθητήρα DHT11

| DHT11 PINS | ESP32-C3-DEVKITC-02 PINS |
|------------|--------------------------|
| DATA       | DIGITAL PIN 4            |
| Vcc        | 3V3                      |
| Ground     | GND                      |



Παρατηρήστε προσεκτικά τη σύνδεση με το Breadboard μεταξύ του DHT11 & ESP32!

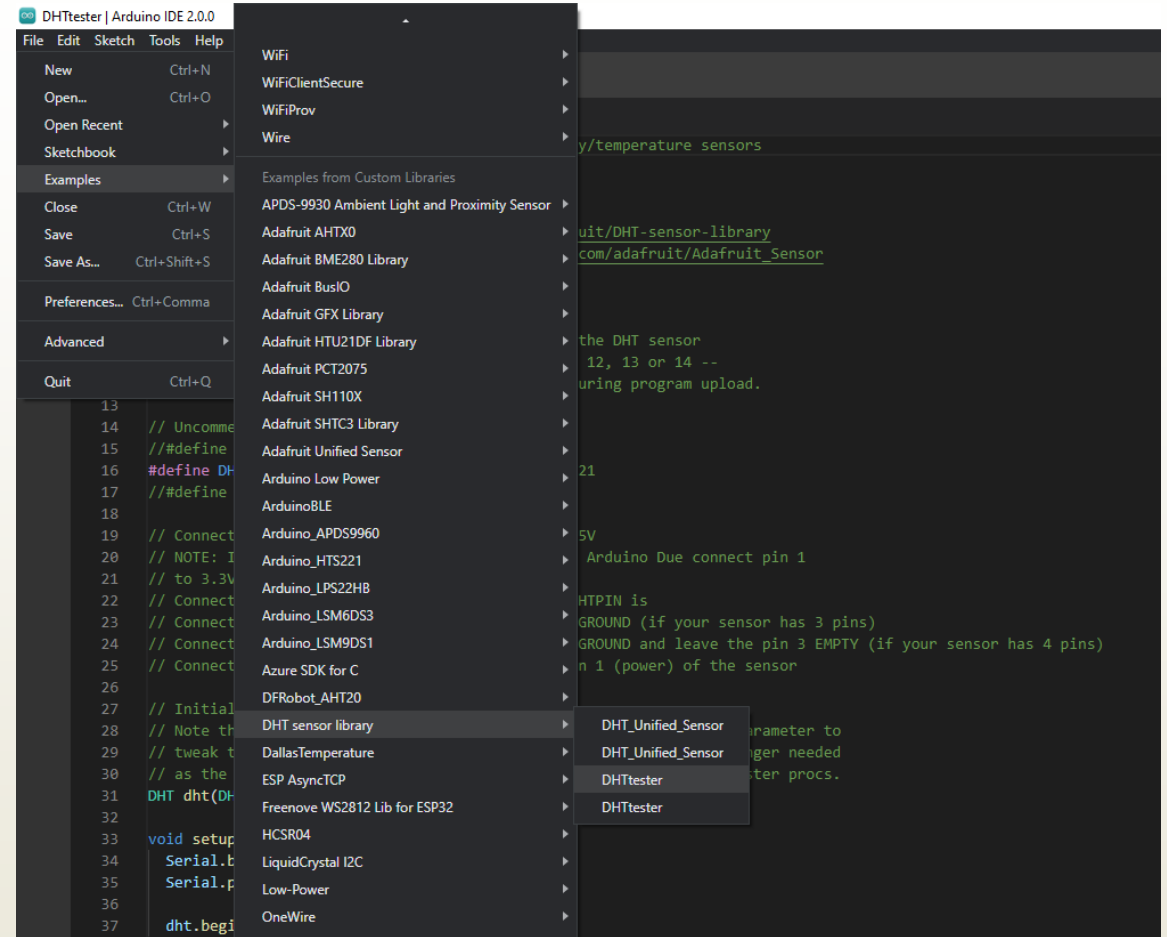


# Pre-made example DHT11 with ESP32

- File > Example > DHT sensor library > DHTtester
- Στη γραμμή 10 αλλάζουμε τον αριθμό του PIN στο αντίστοιχο που έχουμε κάνει τη σύνδεση
- Τελική μορφή εντολής:  
`#define DHTPIN 4`

Αποτέλεσμα:

- Εκτύπωση των τιμών θερμοκρασίας και υγρασίας στη σειριακή κονσόλα



# 16x2 Liquid Crystal Display

- Οθόνη υγρών κρυστάλλων
  - 16 στηλών
  - 2 γραμμών
  - Υποστήριξη αλφαριθμητικών χαρακτήρων
  - Ρύθμιση φωτεινότητας μέσω potentiometer
- Δύο εκδόσεις
  - Διασύνδεση με χρήση 11 ακίδων I/O
  - Διασύνδεση μέσω I<sup>2</sup>C



Έκδοση I<sup>2</sup>C

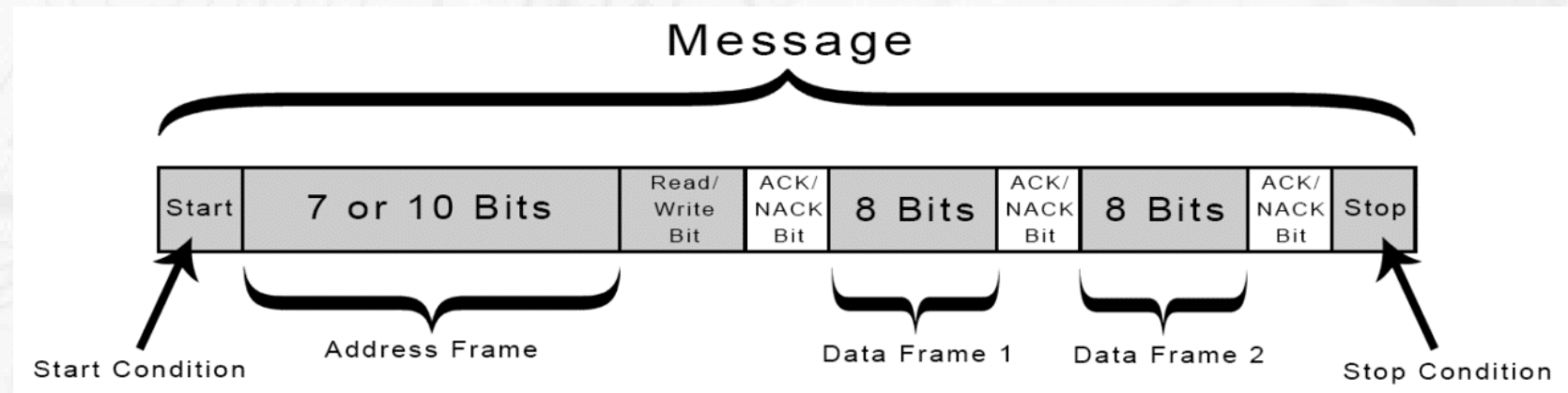


Standard έκδοση



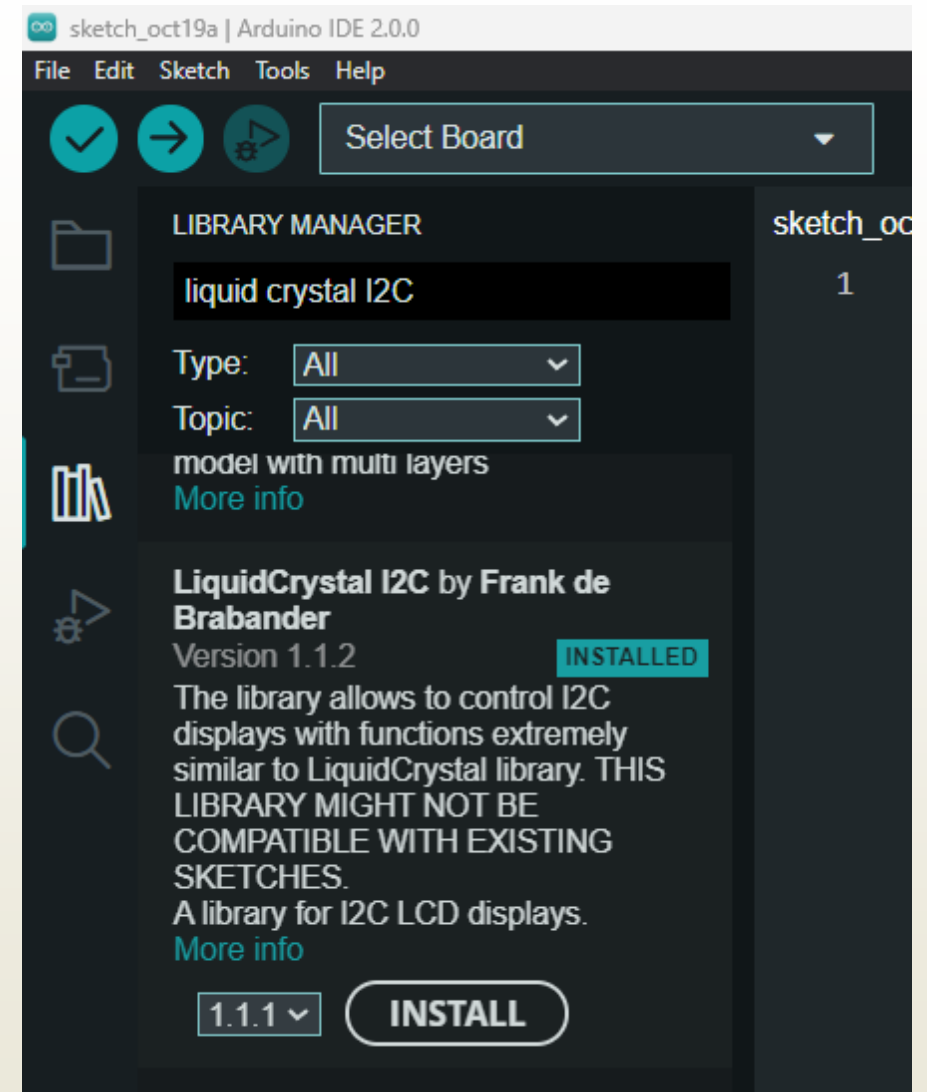
# I<sup>2</sup>C Protocol

- **Start Condition:** The SDA line switches from a high voltage level to a low voltage level *before* the SCL line switches from high to low.
- **Address Frame:** A 7 or 10 bit sequence unique to each slave that identifies the slave when the master wants to talk to it.
- **Read/Write Bit:** A single bit specifying whether the master is sending data to the slave (low voltage level) or requesting data from it (high voltage level).
- **ACK/NACK Bit:** Each frame in a message is followed by an acknowledge/no-acknowledge bit. If an address frame or data frame was successfully received, an ACK bit is returned to the sender from the receiving device.
- **Stop Condition:** The SDA line switches from a low voltage level to a high voltage level *after* the SCL line switches from low to high.



# Εγκατάσταση βιβλιοθήκης I<sup>2</sup>C LCD Οθόνης

- Θα χρειαστεί η βιβλιοθήκη LiquidCrystal I2C από τον Frank de Brabander.
- Επειδή η βιβλιοθήκη δεν έχει ενημερωθεί θα μας εμφανιστεί μήνυμα ότι μπορεί να μην είναι συμβατή με τον ESP32C3 DevKitC02 που θα προγραμματιστεί.
- Διαθέσιμη για κατέβασμα εδώ:  
<https://www.arduino.cc/reference/en/libraries/liquidcrystal-i2c/>
- Μπορεί να χρειαστεί να κάνουμε restart το IDE σε περίπτωση που δεν αναγνωρίζεται η βιβλιοθήκη



# Σύνδεση της LCD 16x2 I<sup>2</sup>C

Για τη σύνδεση της LCD οθόνης με τον μικροελεκτή μέσω I<sup>2</sup>C χρειάζεται:

- SDA PIN (I<sup>2</sup>C)
- SCL PIN (I<sup>2</sup>C)
- 5V
- Ground



Πιο συγκεκριμένα θα πρέπει για να γίνει η  
σύνδεση:

| LCD 16x2 | ESP32C3 DEVKITC-02 |
|----------|--------------------|
| Vcc      | 5V                 |
| Gnd      | DIGITAL GND        |
| SDA      | Digital Pin 8 (8)  |
| SCL      | Digital Pin 9 (9)  |



# No display troubleshooting

- ▶ Σε περίπτωση που δεν εμφανίζεται τίποτα μπορούμε να τρέξουμε «έτοιμο» κώδικα του εργαστηρίου 3 - "LCD\_ADDRESS\_FINDER" ώστε να αναγνωριστεί η διεύθυνση της οθόνης και να αντικαταστήσουμε την απαραίτητη γραμμή κώδικα.

```
35 // set LCD address, number of columns and rows
36 // if you don't know your display address, run an I2C scanner sketch
37 LiquidCrystal_I2C lcd(0x27, lcdColumns, lcdRows);
38
```



# Παράδειγμα χρήσης της LCD οθόνης

- Αφού ολοκληρωθεί σωστά η συνδεσμολογία της οθόνης στο μικροελεγκτή
- Τρέχουμε τον κώδικα LCD\_initialization
- Μετά τον επιτυχή προγραμματισμό του μικροελεγκτή, στην οθόνη εμφανίζεται το μήνυμα

“IoT  
Lesson 3”



# Συνδυασμός αισθητήρα DHT και LCD οθόνης

Συνδέουμε τη οθόνη και τον αισθητήρα με τον μικροελεγκτή ακολουθώντας την παρακάτω συνδεσμολογία

| DHT11  | ESP32-C3-DEVKITC-02 |
|--------|---------------------|
| Data   | DIGITAL PIN 4       |
| VCC    | 3V3                 |
| Ground | GND                 |

| LCD 16x2 | ESP32C3 DEVKITC-02 |
|----------|--------------------|
| VCC      | 5V                 |
| GND      | GND                |
| SDA      | Digital PIN 8      |
| SCL      | Digital PIN 9      |



# Ανέβασμα ολοκληρωμένου κώδικα

- ▶ Ανοίγουμε τον κώδικα "LCD\_DHT11.ino"
- ▶ Κάνουμε Upload τον κώδικα

Αποτέλεσμα:

- ▶ Ανάγνωση και εμφάνιση της τιμής της θερμοκρασίας κάθε 10δ/πτα και εμφάνιση στην LCD Οθόνη!



# Άσκηση για το σπίτι

- 1) Ανάγνωση κάθε 3 δ/πτα και εκτύπωση της τιμής της υγρασίας με τη μορφή  
RH%:  
<xx> (όπου xx: 0..100%)
- 2) Ταυτόχρονη ανάγνωση και εκτύπωση της θερμοκρασίας και της υγρασίας με τη μορφή  
RH: <xx> %, Temp: <yy> C (όπου xx: 0..100, yy:-40..80)
- 3) Ταυτόχρονη ανάγνωση και εναλλαγή εκτύπωσης της θερμοκρασίας και της υγρασίας κάθε 3 δευτερόλεπτα με τη μορφή  
RH: <xx> %, ή Temp: <yy> C (μια γραμμή μόνον)



# Formatting

- "%c" returns char
- "%s" returns string
- "%d", "%i" returns integer
- "%f", "%e" returns floating point number
- "%o" returns octal integer
- "%x" returns hexadecimal integer

