

# 1<sup>ο</sup> ΕΠΑ.Λ. ΔΡΑΜΑΣ

## 2<sup>η</sup> Δράση – «Φτιάξε το δικό σου μετεωρολογικό σταθμό»

### 2.4 Ψηφιακό τεχνικό εγχειρίδιο κατασκευής ενός μετεωρολογικού σταθμού

#### Απαιτούμε υλικά

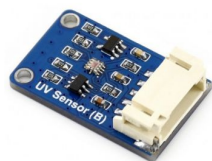
Οι μετεωρολογικοί σταθμοί απαιτούν εξειδικευμένους αισθητήρες για την ακριβή μέτρηση και αναφορά των καιρικών συνθηκών, όπως:

- Adafruit Αισθητήρας Βαρομετρικής Πίεσης/Υψομέτρου
- Waveshare Αισθητήρας Υπεριώδους Ακτινοβολίας
- βροχόμετρο
- ανεμόμετρο
- ανεμοδείκτης

Επίσης απαιτείται το ESP32-DEVKITC-32D. Πρόκειται για μια μικρή και βολική πλακέτα ανάπτυξης με εγκατεστημένη μονάδα ESP32-WROOM-32, κεφαλίδες ακίδων και ελάχιστα πρόσθετα εξαρτήματα. Περιλαμβάνει διεπαφή προγραμματισμού USB σε σειριακό, που παρέχει επίσης τροφοδοσία για την πλακέτα. Διαθέτει κουμπιά για επαναφορά της πλακέτας και θέση σε λειτουργία αποστολής.



Αισθητήρας Βαρομετρικής Πίεσης/Υψομέτρου



Αισθητήρας Υπεριώδους Ακτινοβολίας



Βροχόμετρο



Ανεμοδείκτης



Ανεμόμετρο

#### Συναρμολόγηση του μετεωρολογικού σταθμού

Για να ξεκινήσετε, βρείτε τους δύο μεταλλικούς σωλήνες και συναρμολογήστε τους.

Στη συνέχεια, προσθέστε τον οπλισμό στην κορυφή του σετ σωλήνων. Βεβαιωθείτε ότι έχετε ευθυγραμμίσει το άκρο στον οπλισμό με την εγκοπή στο σωλήνα.

Χρησιμοποιήστε μία από τις παρεχόμενες βίδες και παξιμάδια για να το ασφαλίσετε στη θέση του.



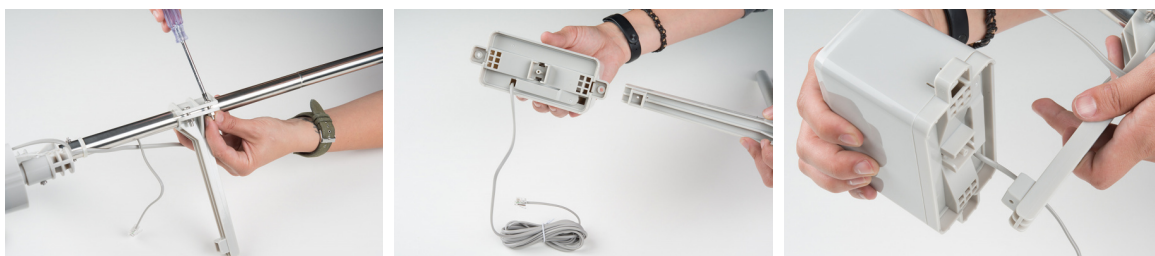
Στη συνέχεια, θα τοποθετήσουμε το ανεμόμετρο στη μία πλευρά του οπλισμού.



Για την τοποθέτηση του ανεμοδείκτη, θα ακολουθήσετε την ίδια διαδικασία με την εγκατάσταση του ανεμόμετρου.



Για να στερεώσουμε το βροχόμετρο, χρειαζόμαστε και δευτερεύοντα οπλισμό. Χρησιμοποιείται για να κρατήσει το βροχόμετρο μακριά από τους άλλους αισθητήρες για να διασφαλίσει ότι μπορεί να λάβει ακριβή μέτρηση. Εάν το βροχόμετρο είναι τοποθετημένο κάτω από τον ανεμοδείκτη ή το ανεμόμετρο, μπορεί να εμποδίσουν τη βροχόπτωση στο μανόμετρο ή να αντισταθμίσουν τις ενδείξεις λόγω απορροής.



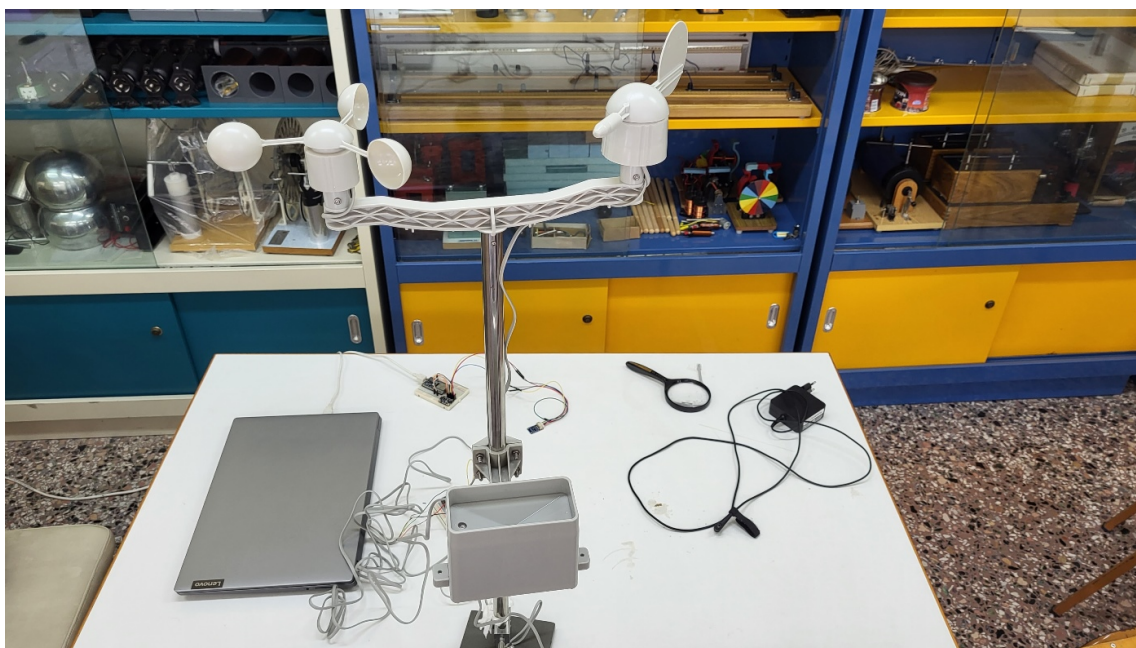
Ξετυλίξτε τα καλώδια από το ανεμόμετρο και τον ανεμοδείκτη. Στην κάτω πλευρά του οπλισμού, θα δείτε κλιπ για να κρατάτε αυτά τα καλώδια στη θέση τους. Σύρετε το καλώδιο από κάθε αισθητήρα σε αυτά. Μόλις συνδέσετε και τα δύο καλώδια, θα παρατηρήσετε ότι το καλώδιο του ανεμομέτρου είναι πολύ πιο κοντό από το ανεμοδείκτη. Θα χρειαστεί να συνδέσετε το καλώδιο του ανεμόμετρου στον ανεμοδείκτη. Περάστε όλα τα υπόλοιπα καλώδια στους μεταλλικούς σωλήνες.



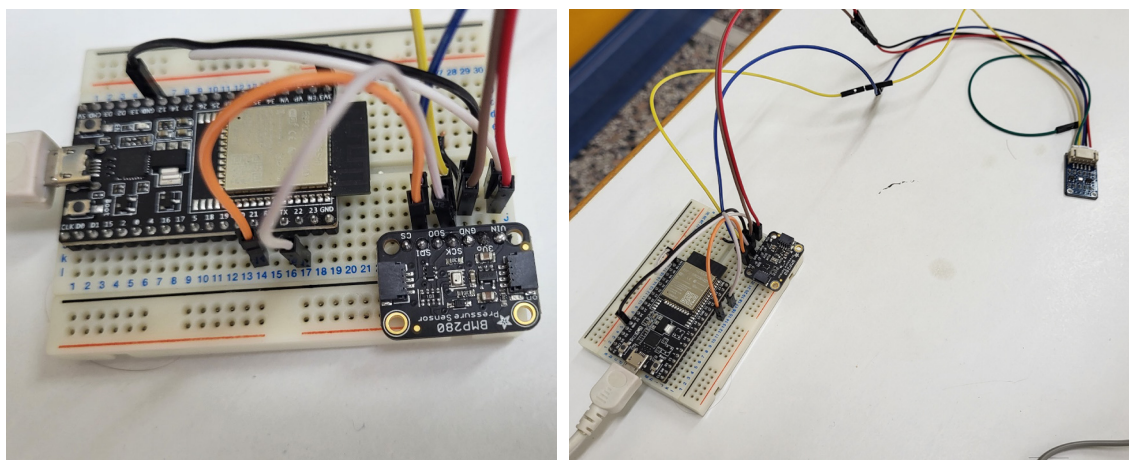


Τώρα έχετε συναρμολογημένο μετρητή καιρού! Βεβαιωθείτε ότι το βροχόμετρο σας είναι μακριά από το ανεμόμετρο και το ανεμοδείκτη και ότι δεν μετατοπίστηκε κατά τη συναρμολόγηση.

Ο μετρητής καιρού σας θα μοιάζει με την παρακάτω εικόνα.



Τώρα μπορείτε να το τοποθετήσετε, να το συνδέσετε σε διάφορα ηλεκτρονικά αισθητήρια στην πλακέτα ESP32-DEVKITC-32D.



## Πρότυπος Μετεωρολογικός Σταθμός

Ο μετεωρολογικός σταθμός στηρίζεται στον μικροελεγκτή esp32, ο οποίος έχει τη δυνατότητα να ελέγχει πλήθος αισθητήρων , να συνδέεται στο δίκτυο WiFi και να στέλνει τα δεδομένα σε ένα Server.

### Μέτρηση θερμοκρασίας και ατμοσφαιρικής πίεσης.

Η μέτρηση της θερμοκρασίας και της ατμοσφαιρικής πίεσης γίνεται με τον αισθητήρα BMP280. Ο αισθητήρας συνδέεται με τον μικροελεγκτή με το πρωτόκολλο I2C στα pin 21 (SDA) και 22(SCL).

Ο κώδικας:

```
#include <Wire.h>

#include <SPI.h>

#include <Adafruit_BMP280.h>

Adafruit_BMP280 bmp; // I2C

void setup() {

  Serial.begin(9600);

  while ( !Serial ) delay(100); // wait for native usb

  Serial.println(F("BMP280 test"));

  unsigned status;

  //status = bmp.begin(BMP280_ADDRESS_ALT, BMP280_CHIPID);

  status = bmp.begin();

  if (!status) {

    Serial.println(F("Could not find a valid BMP280 sensor, check wiring or "

      "try a different address!"));

    Serial.print("SensorID was: 0x"); Serial.println(bmp.sensorID(),16);

    Serial.print("    ID of 0xFF probably means a bad address, a BMP 180 or BMP 085\n");

    Serial.print("    ID of 0x56-0x58 represents a BMP 280,\n");

    Serial.print("    ID of 0x60 represents a BME 280.\n");

    Serial.print("    ID of 0x61 represents a BME 680.\n");

    while (1) delay(10);

  }

}
```

```

/* Default settings from datasheet. */

bmp.setSampling(Adafruit_BMP280::MODE_NORMAL, /* Operating Mode. */

    Adafruit_BMP280::SAMPLING_X2, /* Temp. oversampling */

    Adafruit_BMP280::SAMPLING_X16, /* Pressure oversampling */

    Adafruit_BMP280::FILTER_X16, /* Filtering. */

    Adafruit_BMP280::STANDBY_MS_500); /* Standby time. */

}

void loop() {

    Serial.print(F("Temperature = "));

    Serial.print(bmp.readTemperature());

    Serial.println(" *C");

    Serial.print(F("Pressure = "));

    Serial.print(bmp.readPressure());

    Serial.println(" Pa");

    Serial.print(F("Approx altitude = "));

    Serial.print(bmp.readAltitude(1013.25)); /* Adjusted to local forecast! */

    Serial.println(" m");

    Serial.println();

    delay(5000);

```

### Μέτρηση υπεριώδους ακτινοβολίας (δείκτης UV)

Για την μέτρηση της υπεριώδους ακτινοβολίας χρησιμοποιούμε τον αισθητήρα UV Sensor της Waveshare. Η σύνδεση γίνεται με το πρωτόκολλο I2C.

Ο κώδικας:

```
#include <Wire.h>
```

```

#include "Si1145.h"

Si1145 uv = Si1145();

void setup() {

  Serial.begin(9600);

  if (! uv.begin()) {

    Serial.println("Didn't find Si1145 !\r\n");

    while (1);

  }

  Serial.println("Si1145 Init success !\r\n");

}

void loop() {

  Serial.println("=====");

  Serial.print("Vis: "); Serial.println(uv.readVisible());

  Serial.print("IR: "); Serial.println(uv.readIR());

  // Uncomment if you have an IR LED attached to LED pin!

  //Serial.print("Prox: "); Serial.println(uv.readProx());

  float UVindex = uv.readUV();

  // the index is multiplied by 100 so to get the

  // integer index, divide by 100!

  UVindex /= 100.0;

  Serial.print("UV: "); Serial.println(UVindex);

  delay(5000);

}

```

## Μέτρηση ταχύτητας ανέμου

Για την μέτρηση της ταχύτητας του ανέμου χρησιμοποιήσαμε το “Weather Meter Kit” της Sparkfun. Κατά την περιστροφή του ανεμόμετρου υπάρχει ένας μαγνήτης που μας δίνει ένα σήμα. Με το σήμα αυτό δημιουργούμε ένα interrupt στο pin 26 και καταγράφουμε τη χρονική στιγμή. Μετρούμε δύο διαδοχικά σήματα και από τη χρονική διαφορά υπολογίζουμε την ταχύτητα του ανέμου σε km/h.

Ο Κώδικας:

```
unsigned long t1=millis();
```

```
float windSpeed=0;
```

```
void IRAM_ATTR windSpeedFun() {  
  
    unsigned long t2=millis();  
  
    unsigned long t=t2-t1;  
  
    if (t>100) { windSpeed=2400.0/t; t1=t2;}  
  
}
```

```
void setup() {  
  
    Serial.begin(9600);  
  
    pinMode(26,INPUT_PULLUP);  
  
    attachInterrupt(26, windSpeedFun, FALLING);  
  
}
```

```
void loop() {  
  
    Serial.println(windSpeed);  
  
    delay(2000);  
  
}
```

## Κατεύθυνση ανέμου

Το Weather Meter kit περιέχει και αισθητήρα διεύθυνση ανέμου. Σε κάθε κατεύθυνση αντιστοιχεί και μία τιμή αντίστασης, οπότε στο pin 25 με την εντολή analogRead(25) πραγματοποιούσαμε την μέτρηση.

Ο κώδικας:

```
int vane=0;

void setup() {

  Serial.begin(9600);

  pinMode(25,INPUT);

}

void loop() {

  vane=analogRead(25);

  if (vane<=400)Serial.println("E");

  else if (vane>400 && vane<=790) Serial.println("SE");

  else if (vane>790 && vane<=1340) Serial.println("S");

  else if (vane>1340 && vane<=2010) Serial.println("NE");

  else if (vane>2010 && vane<=2630) Serial.println("SW");

  else if (vane>2630 && vane<=3220) Serial.println("N");

  else if (vane>3200 && vane<=3700) Serial.println("NW");

  else if (vane>3700) Serial.println("W");

  delay(1000);

}
```

### Μέτρηση ύψους βροχής

Στο kit υπάρχει και αισθητήρας μέτρησης ύψους βροχής. Όταν στον αισθητήρα πέσει βροχή ύψους 0,28 mm μας δίνει ένα ηλεκτρικό σήμα. Με το σήμα αυτό δημιουργούμε ένα interrupt στο pin 27. Κάθε φορά που έχουμε ένα interrupt αθροίζονται 0,28 mm βροχής σε έναν αθροιστή, που αντιστοιχεί στο συνολικό ύψος βροχής.

Ο κώδικας.

```
float rainHigh=0;

unsigned long t1=0;

unsigned long t2=0;
```



```

void IRAM_ATTR rainFunct() {

    t2=millis();

    if ((t2-t1)>100){t1=t2; rainHigh=rainHigh+0.28; }

}

void setup() {

    Serial.begin(9600);

    pinMode(27,INPUT_PULLUP);

    attachInterrupt(27, rainFunct, FALLING);

}

void loop() {

    int i=0;

    for (i=0 ; i<255; i++){

        Serial.println(i);

        Serial.print("Rain ");

        Serial.println(rainHigh);

        delay(1000);

    }

}

```

### Αποστολή μετεωρολογικών δεδομένων στον Server του Weather Underground

Για την αποστολή των μετεωρολογικών δεδομένων στο Weather Underground συνδέουμε το esp32 στο δίκτυο WiFi και με http commands στέλνουμε τα δεδομένα.

Κώδικας για σύνδεση στο WiFi

```
WiFi.begin(ssid, password);
```

```
while (WiFi.status() != WL_CONNECTED) {

    delay(500);

    Serial.print(".");
```

### Κώδικας για αποστολή δεδομένων στον Server

```
wuurl=wuurl+"&humidity="+String(h,0)+"&tempf="+String(tf)+"&baromin="+String(barinhg,3)+
"&dewptf="+String(td,1)+"&action=updateraw";
```

```
HTTPClient http;

http.begin(wuurl);

int httpResponseCode=http.GET();

Serial.print("HTTP Response code: ");

Serial.println(httpResponseCode);

String payload = http.getString();

Serial.println(payload);

http.end();
```

Τα δεδομένα του σταθμού είναι προσβάσιμα μέσω της ιστοσελίδας του weather underground ή μέσω της εφαρμογής android: «Μετεωρολογικός Σταθμός»

