



“Cloud Computing in the European schools”

Project: 2017-1-ES01-KA202-038471



Co-funded by the
Erasmus+ Programme
of the European Union



“PAAS activity”

Cloud Temperature - Humidity logger

Tutorial



Index

Σκοπός του μαθήματος	3
Απαιτούμενος Χρόνος: 4 διδακτικές ώρες	3
Χώρος εργασίας: Εργαστήριο Η/Υ	3
Για τις ανάγκες των μαθημάτων θα χρειαστούμε:	3
Υλοποίηση	4
Μέρος 1ο	4
Υλοποίηση της κατασκευής με τη βοήθεια breadboard	4
Προετοιμασία του ARDUINO IDE για τον προγραμματισμό του ESP8266	5
Μέρος 2ο	8
Λογικό διάγραμμα του προγράμματος	8
Το πρόγραμμα φαίνεται στο παρακάτω link	8
2. Γνωριμία και Ρυθμίσεις Thingspeak	9



Δραστηριότητα PAAS

Χρήση της πλατφόρμας Thingspeak για την καταγραφή και διαδικτυακή παρουσίαση της θερμοκρασίας και της υγρασίας

Σκοπός του μαθήματος

Σκοπός της ενότητας είναι να χρησιμοποιήσετε την πλατφόρμα thingspeak για την καταγραφή δεδομένων ενός αισθητήρα στο διαδίκτυο με τη μορφή γραφήματος.

Μέρος 1

Για τις ανάγκες της υλοποίησης θα χρησιμοποιήσουμε το μικροελεγκτή ESP8266, και τον αισθητήρα θερμοκρασίας υγρασίας DHT11. Το υπολογιστικό σύστημα παίρνει τιμές της θερμοκρασίας και της υγρασίας και θα στέλνει τα δεδομένα κάθε πέντε λεπτά .

Στο πρώτο μέρος θα συνδέσουμε τα εξαρτήματα στο breadboard και θα προετοιμάσουμε το περιβάλλον προγραμματισμού ARDUINO IDE για την εγγραφή του κώδικα στο ESP8266.

Μέρος 2

Προγραμματισμός του συστήματος και μελέτη της πλατφόρμας Thingspeak. Το σύστημα θα χρησιμοποιήσει της cloud υπηρεσίες της πλατφόρμας Thingspeak για την παρουσίαση και επεξεργασία των δεδομένων.

Απαιτούμενος Χρόνος: 4 διδακτικές ώρες

Χώρος εργασίας: Εργαστήριο Η/Υ

Για τις ανάγκες των μαθημάτων θα χρειαστούμε:

Ηλεκτρονικά Εξαρτήματα:

Μικροελεγκτής:	ESP8266	x	1
Αισθητήρας Θερμοκρασίας Υγρασίας:	DHT	x	1
Αντιστάσεις:	1 Kohm	x	1
	2.2 Kohm	x	1
Σταθεροποιητής τάσεις 3,3 V ή αντίστ.	LD1117	x	1
Τροφοδοτικό usb 5v ()		x	1
Breadboard		x	1
σειριακός μετατροπέας USB ή ARDUINO Uno		x	1

Ηλεκτρονικός Υπολογιστής

Εγκατεστημένα προγράμματα:
Arduino IDE

Fritzing

Χώρος εργασίας: Εργαστήριο Η/Υ

Υλοποίηση

Υπόθεση : Θέλουμε να κατασκευάσουμε ένα έξυπνο υπολογιστικό σύστημα που να μετράει τη θερμοκρασία και την υγρασία και να καταγράφει τα δεδομένα στο Cloud. Η υλοποίηση θα γίνει με τη χρήση της πλατφόρμας Thingspeak.

Μέρος 1ο

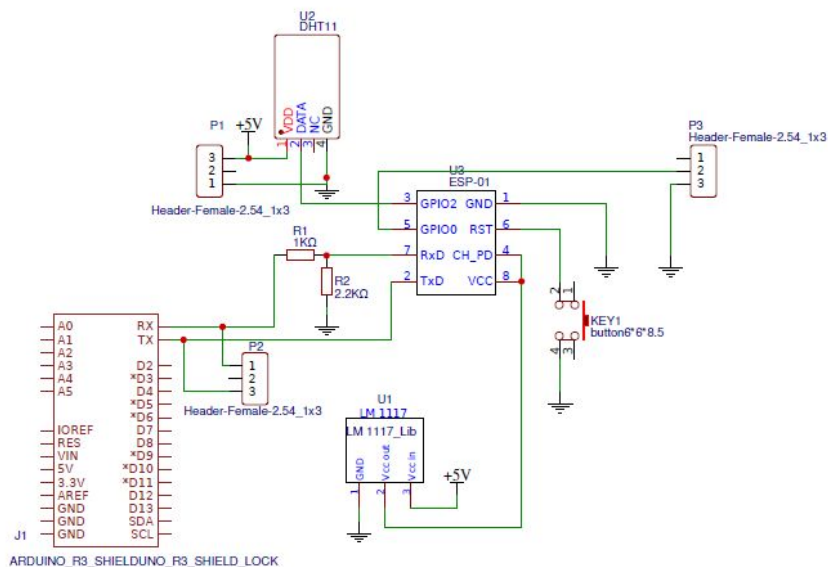
Βήματα

1. Υλοποίηση της κατασκευής με τη βοήθεια breadboard

- Χρόνος υλοποίησης: 1 διδακτική ώρα
- Απαιτούμενες γνώσεις : Ηλεκτρονικό Σχέδιο
- Υλικά που θα χρειαστείς:

Μικροελεγκτής:	ESP8266	x	1τεμ
Αισθητήρας Θερμοκρασίας Υγρασίας:	DHT	x	1τεμ
Αντιστάσεις:	R1=1 Kohm	x	1τεμ
	R2=2.2 Kohm	x	1τεμ
Σταθεροποιητής τάσεις 3,3 V ή αντίστ.	LD1117	x	1τεμ
Button push on		x	1τεμ
Ακροδέκτες pin header 1x2		x	3τεμ
Τροφοδοτικό usb 5v ()		x	1τεμ
Breadboard		x	1τεμ
σειριακός μετατροπέας USB ή ARDUINO Uno		x	1τεμ

Το παρακάτω σχέδιο απεικονίζει το κύκλωμα του υπολογιστικού συστήματος.





Η καρδιά του κυκλώματος είναι το ολοκληρωμένο U3, ο μικροελεγκτής esp8266-01 (για περισσότερα δεξ το σύνδεσμο [esp8266](#))

Ο μικροελεγκτής έχει ενσωματωμένο WiFi επομένως μπορεί να συνδέεται με τον κατάλληλο προγραμματισμό στο διαδίκτυο.

Τα δεδομένα θερμοκρασίας και υγρασίας συλλέγονται από τον αισθητήρα DHT11 (για περισσότερα δεξ το σύνδεσμο [DHT11](#))

Ενδεικτικά αναφέρουμε ότι:

- Τροφοδοσία (Vcc) : 3-5V
- Μέγιστο ρεύμα (Imax): 2.5mA
- Υγρασία: 20-80%, ακρίβεια 2-5%
- Θερμοκρασία: 0 to 50°C, ακρίβεια ±0.5°C

Τα δεδομένα (data) θερμοκρασίας και υγρασίας βγαίνουν στον ακροδέκτη 2 σε ψηφιακή μορφή και διαβάζονται από τον μικροελεγκτή στον ακροδέκτη 5.

Ο μικροελεγκτής με τον κατάλληλο προγραμματισμό αναλαμβάνει να στείλει τα δεδομένα στην ψηφιακή πλατφόρμα thingspeak, όπως θα δούμε παρακάτω.

Αναλυτικά οι συνδέσεις του μικροελεγκτή:

1. **GND**, Ground (0 V) - **Γείωση**.
2. **TX**, Transmit data bit - Στην περίπτωση της εφαρμογής μας χρησιμοποιείται για την **αποστολή δεδομένων** κατά τον προγραμματισμό.
3. **GPIO 2**, General-purpose input/output No. 2
4. **CH_PD**, Chip power-down - Συνδέεται με τη γείωση όταν θέλουμε να προγραμματίσουμε τον μικροελεγκτή.
5. **GPIO 0**, General-purpose input/output No. 0 - Ακροδέκτης εισόδου/εξόδου
6. **RST**, Reset - **Επανεκκίνηση** του μικροελεγκτή **συνδέεται με το button 1**
7. **RX**, Receive data bit - Στην περίπτωση της εφαρμογής μας χρησιμοποιείται για τη **λήψη δεδομένων** κατά τον προγραμματισμό.
8. **VCC**, Voltage (+3.3 V; can handle up to 3.6 V) - **Τροφοδοσία 3,3V** από το **LM1117**.

Στη θέση του Arduino μπορούμε να χρησιμοποιήσουμε οποιοδήποτε σειριακό μετατροπέα σε usb όπως αυτούς που δείχνουμε στον παρακάτω σύνδεσμο. <https://goo.gl/btLsJw>.

Ο ακροδέκτης Rx συνδέεται στο δίκτυωμα αντιστάσεων R1, R2 και μετά στο Tx του μετατροπέα USB.

Ο ακροδέκτης Tx συνδέεται απευθείας στο Rx του μετατροπέα USB.

Προσοχή: Στην περίπτωση που θα συνδέσουμε τον ESP8266 για τον προγραμματισμό του με το Arduino (όπως το παραπάνω σχέδιο) η συνδεσμολογία είναι Rx - RX και Tx - Tx.

2. Προετοιμασία του ARDUINO IDE για τον προγραμματισμό του ESP8266

Για τις ανάγκες του προγραμματισμού θα χρησιμοποιήσουμε το αναπτυξιακό πρόγραμμα ανάπτυξης κώδικα του ARDUINO ([Arduino IDE](#)).



Στο Arduino IDE για να υποστηριχθεί ο ESP8266 πρέπει να γίνουν οι παρακάτω αλλαγές:

- 1) Άνοιξε από τα menu την επιλογή Αρχείο → Προτιμήσεις



2) Βάλτε την επιλογή https://dl.espressif.com/dl/package_esp32_index.json στη επιλογή: Επιπλέον διαχειριστής πλακετών Urls: (“Additional Board Manager URLs”) όπως φαίνεται στην παρακάτω εικόνα . Αφού συμπληρώσεις το κείμενο με το παραπάνω URL πάτησε το κουμπί “εντάξει”.

```

#include <SPI.h>
#include <Wire.h>
#include <Arduino.h>

// Pin definitions
const int LED_PIN = 13;
const int Buzzer_PIN = 12;

// Variables
int ledState = LOW;
int buzzerState = LOW;

// Function prototypes
void setup();
void loop();

// Main function
int main() {
  setup();
  loop();
}

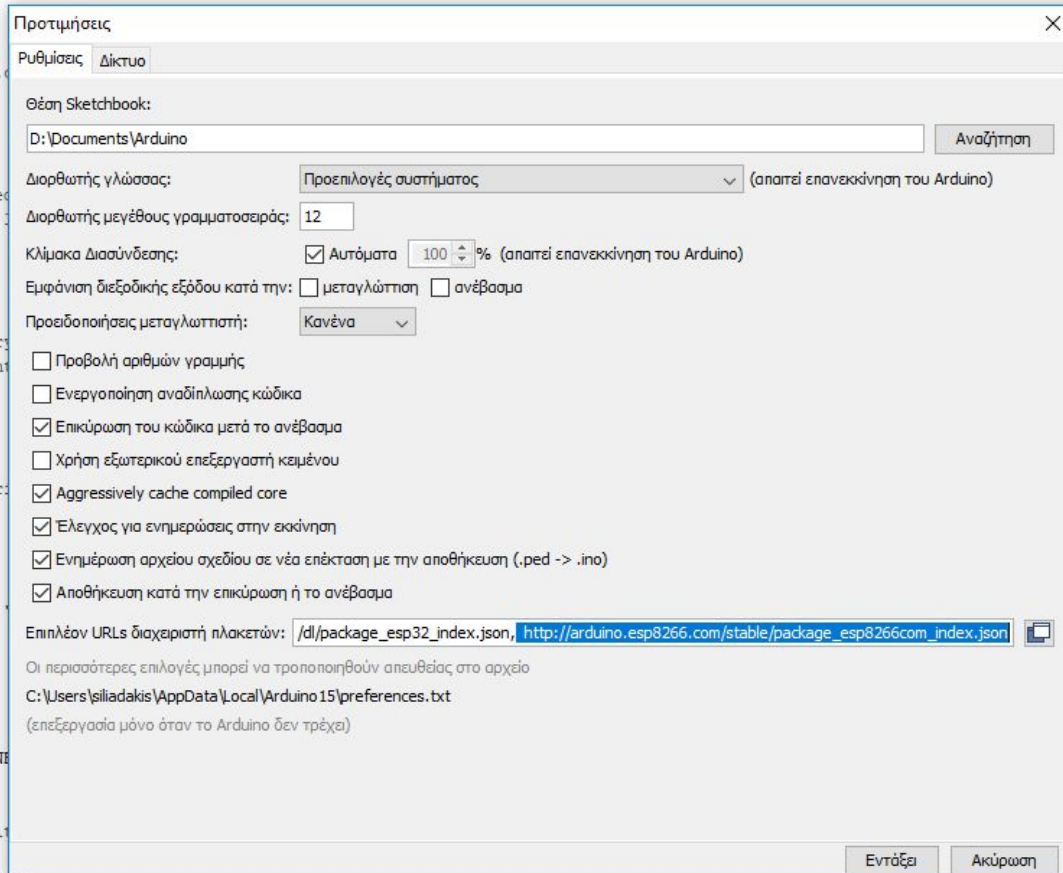
// Setup function
void setup() {
  pinMode(LED_PIN, OUTPUT);
  pinMode(Buzzer_PIN, OUTPUT);
}

// Loop function
void loop() {
  // Toggle LED
  ledState = !ledState;
  digitalWrite(LED_PIN, ledState);

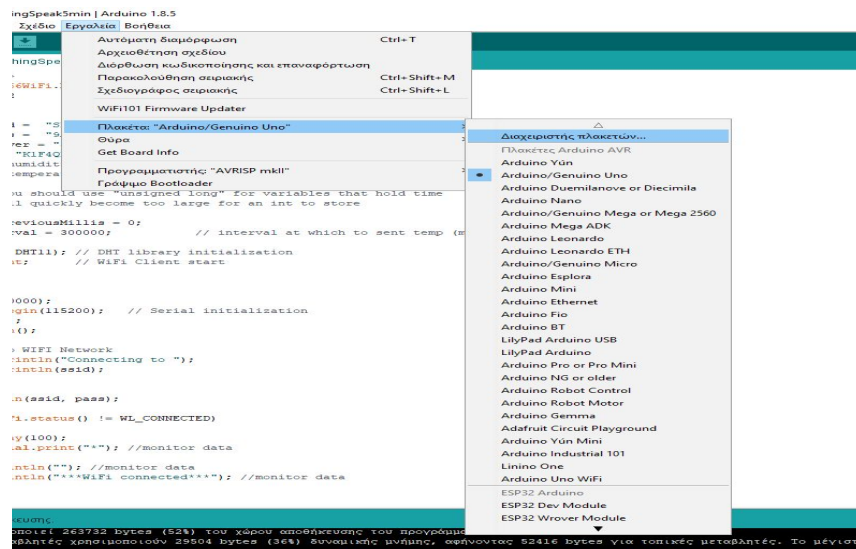
  // Toggle Buzzer
  buzzerState = !buzzerState;
  digitalWrite(Buzzer_PIN, buzzerState);

  delay(1000);
}

```



3) Άνοιξε από το μενού εργαλεία → την επιλογή “πλακέτα” → “Διαχειριστής πλακετών”





4) Εκεί αναζητάμε το **ESP8266** και το εγκαθιστούμε με την επιλογή **εγκατάσταση**. Το αποτέλεσμα είναι όπως αυτό που φαίνεται στη παρακάτω εικόνα.

The screenshot shows an IDE with the following code:

```

DHT11ESP8266ThingSpeak5min $
#include <DHT.h> // DHT library
#include <ESP8266WiFi.h> //ESP8266 library
#define DHTPIN 2 //DHT11 is connected to GPIO Pin 2

const char* ssid = "SKInet"; // Enter your WiFi Network's SSID
const char* pass = "9AB7B742BE"; // Enter your WiFi Network's Password
const char* server = "api.thingspeak.com";
String apiKey = "K1F4QBVEKXD6JNSL"; // Enter your Write API key from ThingSpeak
float humi; //humidity variable
float temp; //temperature variable

// Generally, you should use "unsigned long" for variables that hold time
// The value will quickly become too large for an int to store

unsigned long previousMillis = 0;
const long interval = 300000;

DHT dht(DHTPIN, DHT11); // DHT library
WiFiClient client; // WiFi Client

void setup()
{
  delay(180000);
  Serial.begin(115200); // Serial
  delay(10);
  dht.begin();

  //Connection to WIFI Network
  Serial.println("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, pass);

  while (WiFi.status() != WL_CONNECTED)
  {
    delay(100);
    Serial.print(" "); //monitor
  }
  Serial.println(""); //monitor data
  Serial.println("***WiFi connected***"); //monitor data
}

void loop()
  
```

Overlaid on the code is a window titled "Διαχειριστής πλακετών" (Component Manager). The search type is set to "Όλα" (All) and the search term is "esp8266". The results show "esp8266 by ESP8266 Community έκδοση 2.4.2 INSTALLED". Below this, a list of compatible boards is provided, including Generic ESP8266 Module, ESP8285 Module, ESPDuino (ESP-13 Module), Adafruit Feather HUZZAH ESP8266, Invent One, XinaBox CW01, ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Olimex MOD-WIFI-ESP8266(-DEV), SparkFun ESP8266 Thing, SparkFun ESP8266 Thing Dev, SweetPea ESP-210, LOLIN(WEMOS) D1 R2 & mini, LOLIN(WEMOS) D1 mini Pro, LOLIN(WEMOS) D1 mini Lite, WeMos D1 R1, ESPino (ESP-12 Module), ThatEasyElec's ESPino, WiFiInfo, Arduino, 4D Systems gen4 IoT Range, Digistump Oak, WiFiDuino, Amperka WiFi Slot, Seed Wio Link, ESPectro Core. There are links for "Online help" and "More info". A "Κλείσιμο" (Close) button is at the bottom right.

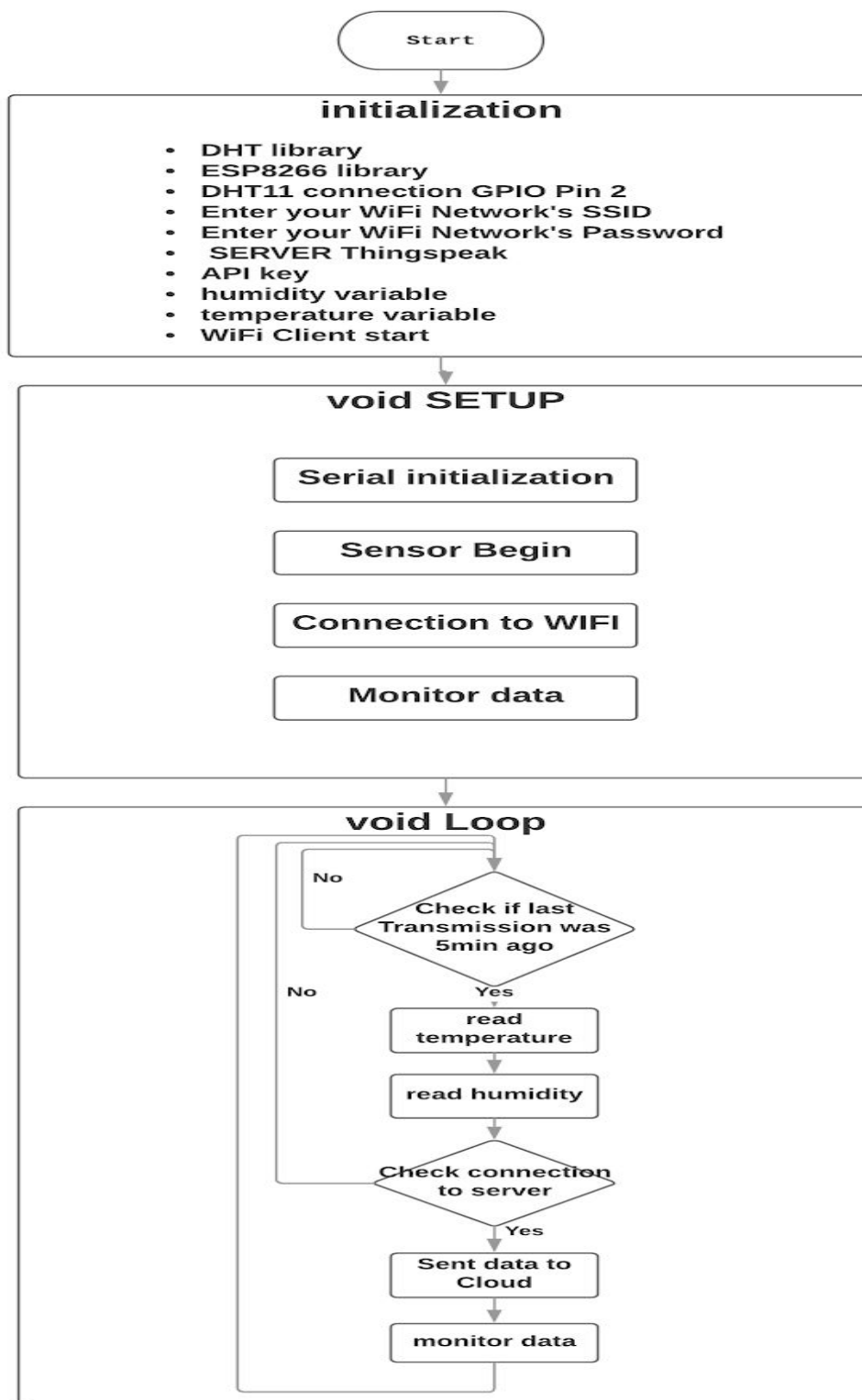
At the bottom of the IDE, a status bar shows: "Ολοκλήρωση αποθήκευσης. Το σχέδιο χρησιμοποιεί 263732 bytes (52%) του χώρου αποθήκευσης του προγράμματος. Το μέγιστο είναι 499696 bytes. Οι καθολικές μεταβλητές χρησιμοποιούν 29504 bytes (36%) δυναμικής μνήμης, αφήνοντας 52416 bytes για τοπικές μεταβλητές. Το μέγιστο είναι 81920 bytes."

Είμαστε έτοιμοι για να προγραμματίσουμε τον ESP8266.



Μέρος 2ο

1. Λογικό διάγραμμα του προγράμματος



Το πρόγραμμα φαίνεται στο παρακάτω link

<https://create.arduino.cc/editor/siliadakis/46c44fdc-adf8-480a-9724-cfdc8e058321/preview>



Στην αρχικοποίηση το προγράμματος πρέπει να βάλουμε τα παρακάτω στοιχεία.

Όνομα WiFi και κωδικός εισόδου

```
ssid = "XXXXXXXXXXXX"; // Enter your WiFi Network's SSID  
pass = "XXXXXXXXXXXX"; // Enter your WiFi Network's Password  
καθώς και το Write API Key από το κανάλι μας.  
apiKey = "xxxxxxxxxxxxxxxx"; // Enter your Write API key from ThingSpeak
```

2. Γνωριμία και Ρυθμίσεις Thingspeak

1. <https://docs.google.com/presentation/d/1b4bV6q9nmYt55q5gGrw6kC3kExGNvWvzEcw8tZICVs/edit?usp=sharing>
2. Δημιουργία λογαριασμού.

3. Use of Channels , Thingspeak API

Reference

https://www.electronicshub.org/dht11-humidity-sensor-with-esp8266/#A_Brief_Note_on_DH_T11_Sensor