# How to monitor
# Temperature & Humidity in beehives
# for under $20 per hive

By Ron Zickefoose

This paper sets out to describe how to set up a remote temperature and humidity monitoring system for a beehive.  A major limitation is the need for wireless internet available at the hive location. Additionally power may become an issue as I have not tested the longevity of the AAA battery setup in this configuration and some beekeepers may want to use a more permanent power source.

## Parts Needed

| Parts List | | Qty | | Each |
|---|---|---|---|---|
| 1 | 5pk ESP8266 Serial Wireless Transceiver Module | 1 | $15.99 | $3.20 |
| 1 | Ch340 USB to ESP8266 Serial Wireless board | 1 | $8.99 | $8.99 |
| 1 | 5pk DHT-11 Temperature Humidity Sensor | 1 | $8.99 | $1.80 |
| 1 | 10pk 6mm push button switch | 1 | $5.29 | $0.53 |
| 1 | 25pk 10K ohm Resistors | 1 | $5.80 | $0.23 |
| 1 | 10pk 5x7 prototype pcb board | 1 | $5.99 | $0.60 |
| 1 | 6pk 2x1.5v aaa battery holder with leads | 1 | $7.09 | $1.18 |
| | Cost Per Unit | | | $16.53 |

Other various parts include wire, solder a soldering iron, a breadboard, & breadboard jumpers for testing.

ESP 8266 – The ESP8266 ins a wi-fi enabled microcontroller about the size of a quarter.  You can find detailed information about the product at http://esp8266.net/

CH340 USB to ESP8266 Adaptor – This is an adapter with a uart and usb port allowing you to communicate with the microcontroller.



DHT-11 Sensor- This slick little sensor about the size of a dime does one simple job. it reads temperature and humidity and serially transfers it to the middle pin for collection by the microcontroller.



-

Push Button Switch – This is required to modify the CH340 so the microcontroller can be put into a programmable state.
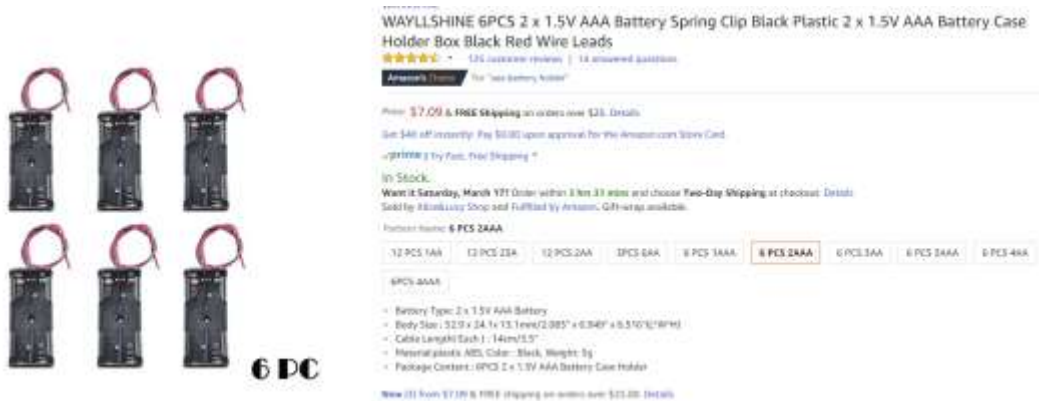
10K Resistors – The resistors are used to "Pull up" certain inputs on the microcontroller limiting current flow and providing a definite logic state.



E-Projects 2SEPS1410K0 10K Ohm Resistors, 1/4 W, 5% (Pack of 25)

Prototype Board – used for the final build, I've also used a breadboard which allows for rapid changes as connections are not soldered as they will be with the prototype board.



10Pcs 5 x 7 cm DIY Prototype Paper PCB Universal Board New
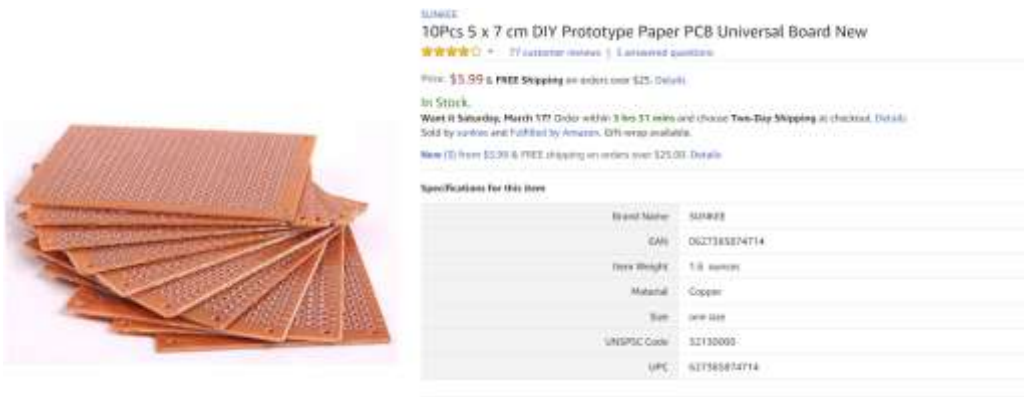
Battery Holder – simply holds 2 batteries providing 3volts. The microcontroller really would prefer 3.3volts which can be provided by a more permanent ac adaptor power supply.



WAYLLSHINE 6PCS 2 x 1.5V AAA Battery Spring Clip Black Plastic 2 x 1.5V AAA Battery Case Holder Box Black Red Wire Leads

## The Build

Here is the pinout for the esp8266, you will refer to this often during the build.

## ESP8266 PINOUT
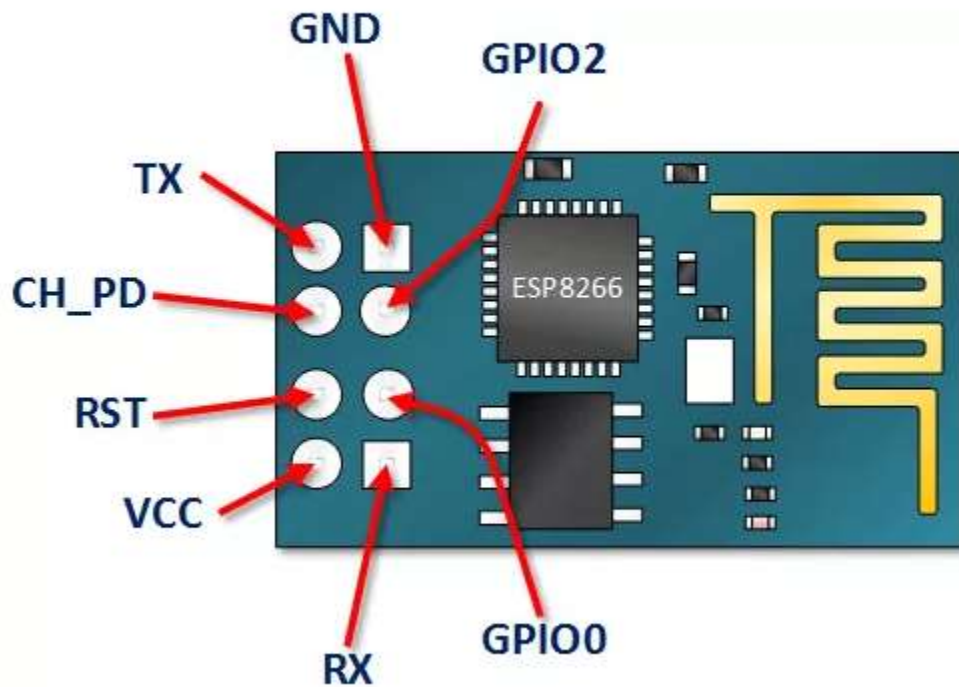


The CH340 does not work to program the ESP8266 out of the box because the GPIo0 pin needs to be momentarily taken to ground on power up for the device to enter a state where it can be programmed. (EEROM in Write Mode).  I solved this by adding the small switch between the pins that connect to GPIO0 and GND.

Here is the proper orientation of the esp8266 on the CH340 board.
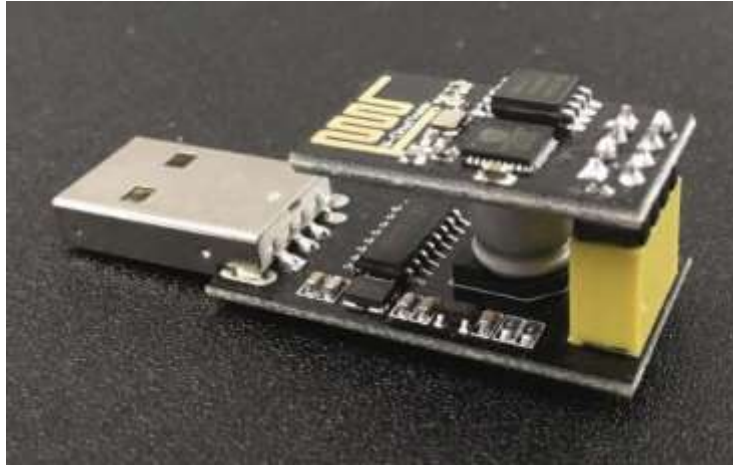
Now we are ready to insert it into your computer's usb port.

If windows does not find the drivers you can download and install them found here
http://sparks.gogo.co.nz/assets/_site_/downloads/CH340_WINDOWS.zip

Go to the device manager to determine what com port the ch340 is operating on as we will need that information to flash the rom on the esp8266

Now we are ready to install the Easyesp software.

The software is found at http://www.letscontrolit.com/downloads/ESPEasy_R120.zip . Alternatively download the most recent stable release here:
https://www.letscontrolit.com/wiki/index.php/ESPEasy#Loading_firmware

unzip the file and run the **flash** batch file It will first ask you for the com port which you got from device manager in the previous step. It then asks the memory size which is 1024. Next question is the Build # which was 120 for flash ESPEasy_R120_1024 file.

Once the file is installed, restart the device by unplugging and plugging it back in. Make sure that the switch you installed is not depressed this time.

The unit will now be broadcasting as a wi-fi  access point named ESP_0 with a password of configesp.

Connect to the device and ignore any messages that you are not connected to the internet.

Open a browser and you will see the espeasy wizard. Select the wifi network that your device will be connecting to and enter the password so it can connect.  It will then give you an ip address on that network which you can use to connect for completing configuration.

Your esp8266 is now programmed to connect to your wifi and we can continue the rest of the build.

Now it's time to build the circuit. You may want to use a breadboard as I did to test. Wire and solder the circuit as follows: VCC to 3.3volts, GND to Ground. GPIO2 to the middle pin of the DHT-11. VCC and GND being the other pins on the DHT-11 that need connected to 3.3volts and ground accordingly. You will need to tie both CH_PD and RST to VCC via the 10K ohm resistors. Connect the batteries and you are ready to finish configuring the unit.
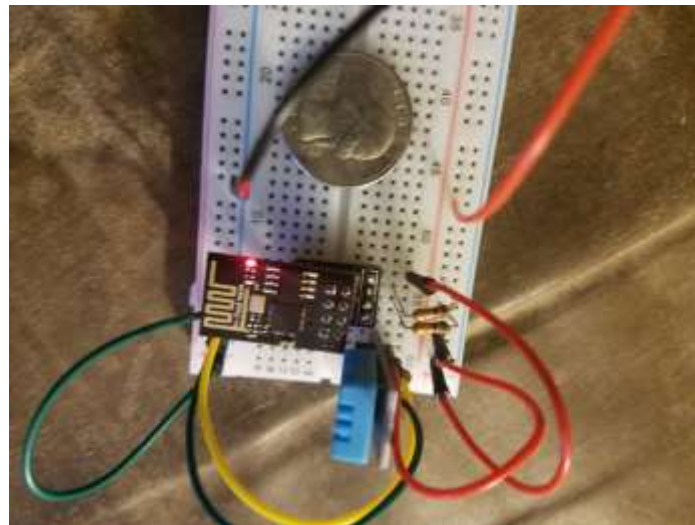


## Configuring

Connect you laptop or phone back to the wi-fi being used by the device and use the ip address to access the configuration page of the module through your browser.

Go to the tools tab and select I2CScan and it should find the DHT-11 sensor. Edit the device and select DHT as the device type. Add a device name, an index number , and value names as the field names to be used to upload the data. Click on the devices tab and you should be able to see your real time temperature and humidity readings.
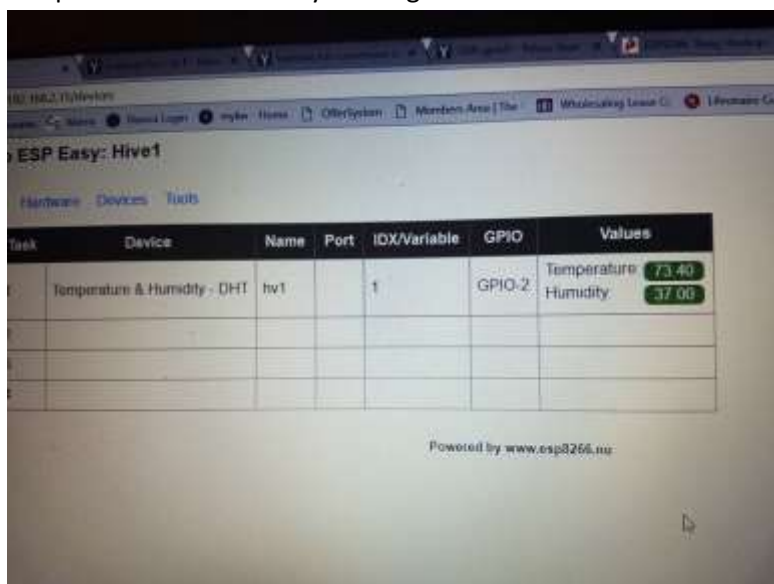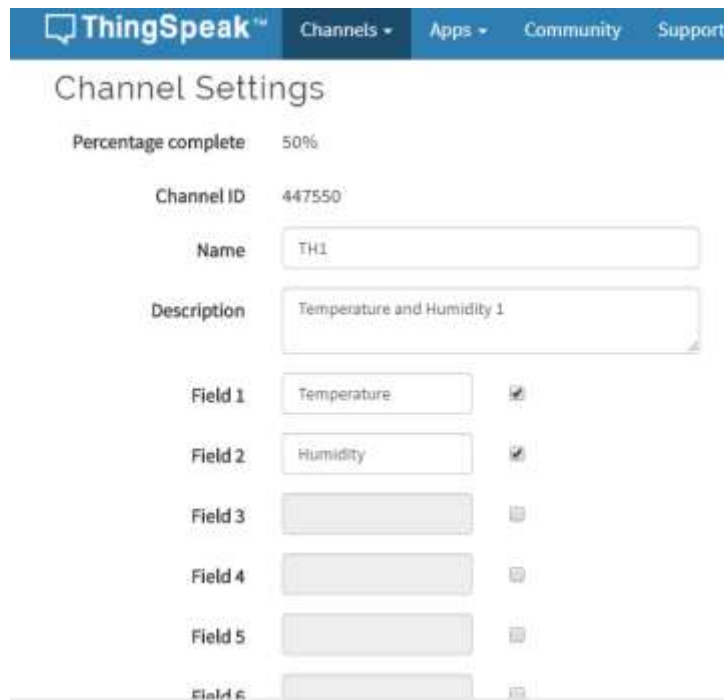
Next we are going to configure uploading data to the web.

Back on the edit devices section there is a check box for send data which you will want checked as it configures to send the data to Thingspeak.

Thingspeak is a free cloud service that allows you to upload sensor data and visually display it again in a browser.  So you will need to sign up for a free Thingspeak account at https://thingspeak.com/users/sign_up

Once you are logged in you will need to create a channel for the data with name and field names used earlier.



Then Navigate to the API KEY tab and record the Channel ID as well as the Write and Read API KEYS.

Back on the configuration page you will select Thingspeak as the protocol set controller hostname to api.thingspeak.com and enter the write API Key you generated on the thingspeak account as the password.

| | |
|---|---|
| Protocol: | ThingSpeak ▼ ? |
| Locate Controller: | Use Hostname ▼ |
| Controller Hostname: | api.thingspeak.com |
| Controller Port: | 80 |
| Controller Password: | IZU7NDU5HXOLW7UI |

The system defaults to reading the temperature in Celsius. If you would like Fahrenheit you will need to add the following formula in the edit device page: (%value%*9/5)+32

| **Optional Settings** | | |
|---|---|---|
| Formula temp: | (%value%*9/5)+32 | De |
| Formula hum: | | De |
| Formula press: | | De |
| Value Name 1: | temp | |
| Value Name 2: | hum | |

Back on the Thinkspeak web site go to private view and you will see the graphs of your data.
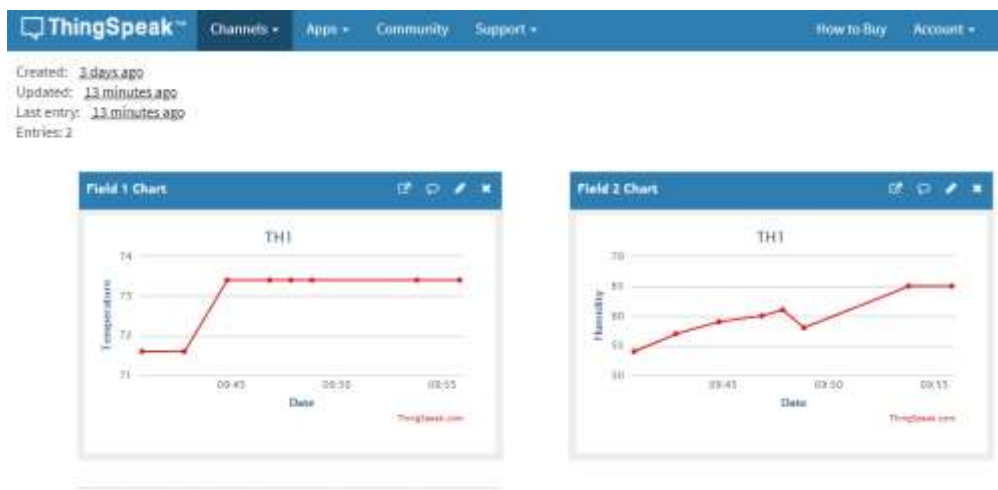
You will want to play with the sensor delay on the config page to set how often data is sampled.

## Conclusion

This paper describes the step by step setup of a temperature and humidity sensor for a beehive for under $20. Many disciplines including electronics and computer configuration are used but we were able to eliminate the need for programming by using the easyesp flash and framework.  I hope you find this project useful and it helps you create other projects which improve the survivability and sustainability of beekeeping.