

Configure of Espress for HeaterMeter:

requirements: NodeMCU/ESP12-E with 4MB flash size.

You'll need a flasher for the NodeMCU if you don't have it yet. Here's a link

<https://github.com/nodemcu/nodemcu-flasher>

It has compiled win32/win64 binaries within

Before you begin, download the Espress Module for HeaterMeter ZIP file and extract all contents to a temp folder. You should read this entire document over before starting....

For now I'll just release compiled BIN files until there's a real release...and I get a chance to organize the source files to put on github

STEP #1 Get the Espress Module(ESP) to connect to your internet router.

You do not connect the ESP to the HeaterMeter(HM) yet....

grab the BIN file from zip file....the BIN file is designed for nodemcu or ESP12-E chips with 4MB flash sizes....make sure you know the size of flash on your NODEMCU/esp chip. They range, but it's important you flash the correct bin file to the correct size nodemcu.

use the nodemcu flasher to burn the bin file to the device....and reboot device.

..once it restarts the ESP will start in AccessPoint mode(AP)...use tablet/phone/notebook to connect to the AP(WIFI) that will start up (you should see SSID like espHeater1545...

Once you are connected to the ESP you can bring up your browser and go to <http://setup.com> (or "anything".com since all websites should land you on the wifi config screen). If you have any problems connecting via <http://setup.com> you can try accessing directly using <http://192.168.4.1>

If you connect a wifi config page should be displayed. Fill in the DEVICE NAME (no spaces, call it something simple and short or leave the default myesp). Set your SSID(wifi) and your wifi password. You can leave all other items as default. Press Save settings....you should power down/on the ESP module for a clean boot.

You will know you are successful if after a few seconds after boot up the local AP of the ESP is no longer there. The local AP is only on when there's an error connecting to your internet.

STEP #1 Connecting to ESP via your desktop computer.

Physically connect the ESP to your HM...(see section on connecting).

Once Step 1 is complete...we no longer will connect the Espress Module via its local AP. We will now only access it via your local lan.

You should now use a desktop computer and setup the ESP and thingspeak/cloud/HeaterMeter stuffs.

To make life easier I've implemented mDNS protocol for ESP...that way you can to your browser and just type `http://myESP.local` to access the webpages.

Accessing the ESP through lan can be tricky though because or different ways different browsers/OS connect to mDNS (that is the protocol the ESP uses to broadcast its name).

Let's say you called your devicename myESP from step one: You will be able to access your ESP by typing `http://myESP.local` from your browser. It will work for some browers in windows and will always work on iOS/macs. If your system doesn't connect via the .local address you will have to access it via it's local lan ip like: `http://192.168.1.xx` . BTW: when you reboot the HM/ESP connected together after a few seconds the LCD will display a message Connected with it's local IP or a message that says cannot connect.

I have tried on a few systems and here's what I found out so far.

Windows 10 with chrome browser works (ie doesn't work)

Windows 7 doesn't work...

Ipads/Iphones/Macs work

Android doesn't work....

It's been said that if you have windows box with itunes installed it will work as well (mDNS was invented by apple)...again, if your desktop doesn't work with the .local address just access it directly via the IP from the HM lcd screen on bootup.

Once you connect via desktop the first page on the ESP is a file browser startup mode (this is the page that you get when it's waiting for config files).

STEP #3 Uploading config files/apps to the ESP module.

Once you connect via desktop and see the filebrowser you are now ready to copy the needed files over to the ESP.

We now just need to copy all the files from within the html folder of zip to the root folder of the ESP module.

Press the Choose Files button from the top of the page and go to the html folder from ZIP file. Go into the html folder and hilite/select all the files and press OK/open. It will upload all the html files into the root of the ESP. Ensure that all files are copied (html file transfer isn't always 100%)...once you are sure all files got upload the RESET the ESP/HM.

If everything worked you should now access the <http://myesp.local> web address and be displayed a proper index page to start configuring the entire system.....

STEP #4 Setup ThingSpeak.

You'll need to setup a thingspeak account/channel and setup the channel like the attached image. Then inside thingspeak there's an API KEYS page that you grab the writekey and put in into the esp webconfig section.

It's important that you keep the field order the same....field1 must be pittemp, or something similar like BBQTemp...

Channel Settings

Percentage complete 35%

Channel ID

Name

EspHeaterMeter

Description

BBQ Control Temp

Field 1

PitTemp



Field 2

FoodTemp1



Field 3

FoodTemp2



Field 4

AmbTemp



Field 5

FanAvg



Field 6

Fan



Field 7

SetPoint



Field 8

LidOpen



Metadata

Help

Thing

Channels store eight fields of status data, and visualization

Channel

- Channel
- Description
- Field#
- Channel
- Metadata
- Tags:
- Latitude
- Longitude
- Elevation
- Make
- URL: I
- Video

You will need to setup a talkback App on thingspeak to support setting temperatures and settings alarms....

inside thingspeak there's an APPs page that you want to add a talkback(name doesn't matter) and grab the ID and the key for it as well....put that in the esp cloud config...

Apps / TalkBack / SetTemp

Edit TalkBack

Name: SetTemp

TalkBack ID: 3910

API Key:

Regenerate API Key

Created: Mon, Nov 2 at 1:58 PM

Logged to Channel: EspHeaterMeter

Commands

Add a new command

NOTE:!! Espress Module doesn't read any current setting from your HM currently....configuration is used/saved on the ESPress so you'll have to setup your HM again within the ESP webpages....it will then send them to your HM...

note: thingspeak is a cloud based system so that user doesn't need to configure stuffs locally. however, thingspeak is opensource and you can actually install it locally so that you aren't 'stuck' with a certain provider. Oh and the 15/second update gone if you use your own server locally...there's a website that shows how to turn a raspberrypi into a thingspeak server...so that's cool too to try. (in esp webconfig you can change the server ip)

Connecting ESPress to HM.

The preferred method is to make a small daughter/prototype board that holds the nodemcu upsidedown and has male headers. You will need to connect 5 wires (power,ground,tx,rx,reset) between the nodemcu module and the HM board.

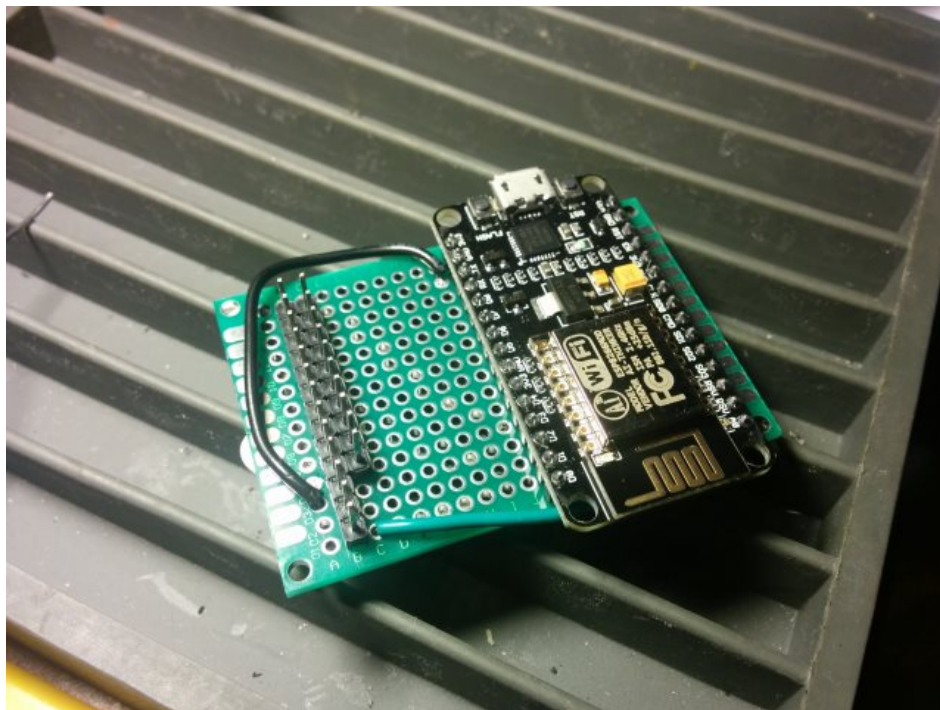
Pin Connection:

HeaterMeter		NodeMcu
5v (pin 26)	<->	Vin (5v pin)
ground (pin 22)	<->	ground
RX (pin 20)	<->	D1 (gpio5)
TX (pin 18)	<->	D2 (gpio4)
Reset (pin 6)	<->	D5 (gpio14)

Sample Photos:

You'll see that I had to cut D0 and A0 pins off the nodemcu to fit the small board...I used full row of headers so that connection to HM is stronger and no chance of shifting pins and blowing up stuff ;)

The node mcu rests on the 2 plastic humps on the LCD panel so it seems there's no chance of shorting the LCD traces....but you may feel more safe to put a layer of something between the two (anti-conducting!)



Dashboard for ESPress Module.

The dashboard you can see what's going and and set your alarms/temperature from anywhere (even outside you home!).

currently its just a javascript file that you can store on your local desktop or somewhere on the web. (no php,or server side scripting needed)

you can link to my website directly

<http://nailbuster.com/thingspeak>

Once you load it up you will need to go to config tab and setup your thingspeak settings for it to work.

it's an early proof of concept.....

Make your own Dashboard tech info:

you can examine the source of the sample dashboard file from above to get an idea on reading the values.

to set temp we need a javascript button that sends temperature like this: (http post not get!)

remember the APIkey is the talkback KEY NOT the standard writekey for channel.

```
$.post(https://api.thingspeak.com/talkbacks/3910/commands?apikey=EBJXXXXXZH&command_string=$SETPOINT,'+document.getElementById("command").value);
```

to set alarms:

```
$.post('https://api.thingspeak.com/talkbacks/3910/commands?apikey=EBJXXXXXZH&command_string=$ALARM,'+document.getElementById("command").value);
```

input boxes to get 8 values (4 probes with low/high points).

command_string looks like \$ALARM,200,300,200,300,200,300,200,300

(negative values will disable alarm permanent)

ThingSpeak React

With thingspeak react you can do things when alarms from HM are triggered. See thingspeak website for all the features on React.

For example, you can tweet something, or you can you send a HTTP request to a service for all sorts of things. Here's a sample of thingspeak react config.

Apps / React / Edit

Help

React Name

React 1

Condition Type

Status

Test Frequency

On Data Insertion

Condition

If channel

EspHeaterMeter (62709)

contains

Pit Alarm!

Action

ThingTweet

then tweet

%%trigger%%

using Twitter account

thingpeak

Options

☐ Run action only the first time the condition is met

☒ Run action each time condition is met

Save React

React Settings

- **React Name:** Enter a u
- **Condition Type:** Selec
- **Test Frequency:** Choo
- **Condition:** Select a ch
- **Action:** Select ThingTv
- **Options:** Select when

[Learn More](#)

For twitter...this is how I get notifications: Setup a new twitter account for your BBQ, link that new account to thingtweet inside thingspeak. Setup like picture above. On your regular twitter account, go

and find your BBQ twitter account and follow it. You will also click the 'star' on the account so that any new tweets from BBQ you will be notified on phone/tablet/etc...ensure twitter notifications is enabled on your mobile device.

***** Notes:

When using the filebrowser utility from the esp webmenu all your cloud services are disabled. This is by design and is needed for stable file transfers. Once you are finished inside the file browser there is a button on bottom of screen to 'reboot esp'...use that so that all cloud services we be re-enabled.

For AVR flashing via WiFi(esp) the atmega chip needs to have optiboot bootloader installed. This is beyond the scope of this document on how to get a bootloader onto your atmega chip. Google is your friend....