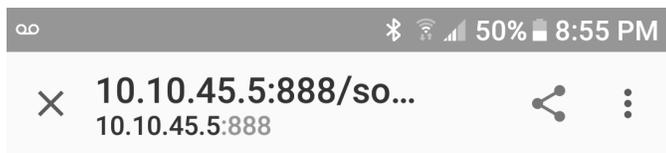


ESP 9266 Control of 433MHz Switches

Lets look at what it does first:



Temperature: 22.10°C

Humidity: 55.70%

1 - Guest Room	<input type="checkbox"/> ON	<input type="checkbox"/> OFF
2 - Living Room	<input checked="" type="checkbox"/> ON	<input type="checkbox"/> OFF
3 - Family Room	<input type="checkbox"/> ON	<input type="checkbox"/> OFF
4 - Landing	<input type="checkbox"/> ON	<input type="checkbox"/> OFF
5 - SB Bedroom	<input type="checkbox"/> ON	<input type="checkbox"/> OFF
6 - Car Tent	<input type="checkbox"/> ON	<input type="checkbox"/> OFF
7 - Tube Led	<input type="checkbox"/> ON	<input type="checkbox"/> OFF
8 - Pond Pump	<input type="checkbox"/> ON	<input type="checkbox"/> OFF
9 - Radio	<input type="checkbox"/> ON	<input type="checkbox"/> OFF
10 - Fan	<input checked="" type="checkbox"/> ON	<input type="checkbox"/> OFF

SSID: Steele

PW: Piwehavelotsof.

10.10.45.5:888 to connect to the esp8266 web server.

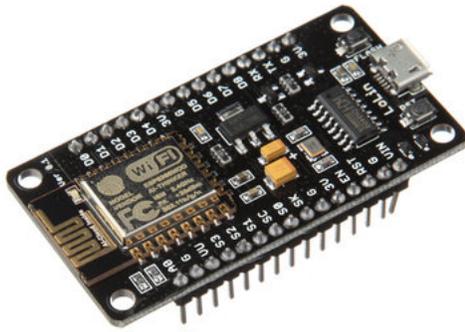
ID: light

PW: makeBright

Hardware to make it work

Parts

- ESP8266



<https://www.aliexpress.com/item/New-Wireless-module-CH340-NodeMcu-V3-Lua-WIFI-Internet-of-Things-development-board-based-ESP8266>

Is the module I am using, and I don't recommend it as it is too wide to leave hook up space on a standard breadboard.

[aliexpress...](#)

- 433MHz Receiver/Transmitter Pair



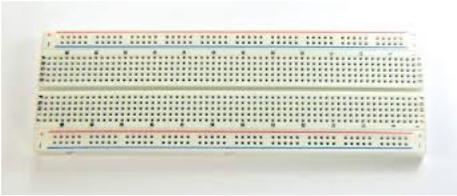
<http://a.co/dRFPVea>

- 433MHz Switches



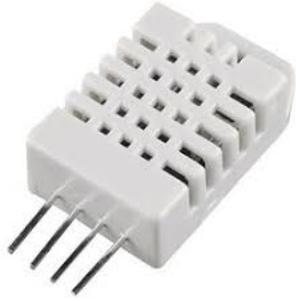
<http://a.co/2PJQtqQ>

- Breadboard.



<http://a.co/dFNjGuB>

- DHT Temperature and Humidity Sensor. [Covered in a different presentation](#) - ask questions later if you wish.



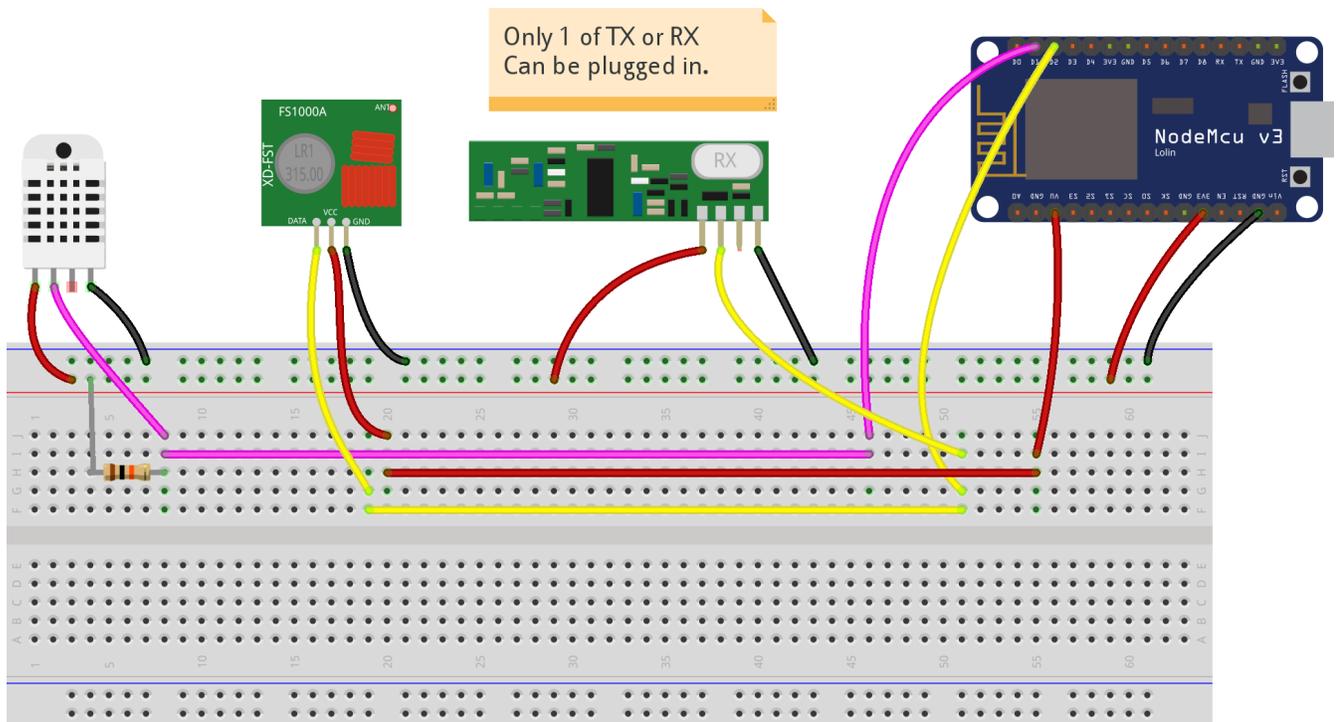
<https://www.aliexpress.com/item/1pcs-Free-Shipping-DHT22-AM2302-Digital-Temperature-and-Humidity-Sensor/1041496689.html?spm=a2g0s.9042311.0.0.fu5Tiz>

- 10K Resistor



<http://a.co/0pVdGy7>

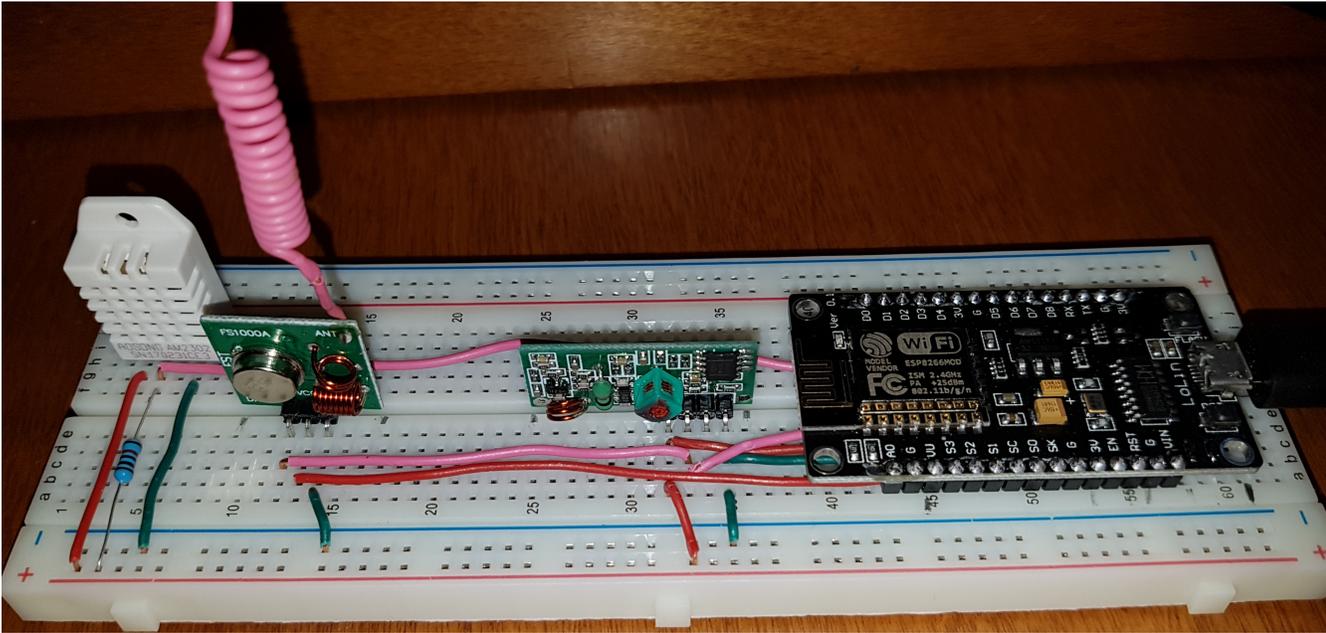
Circuit



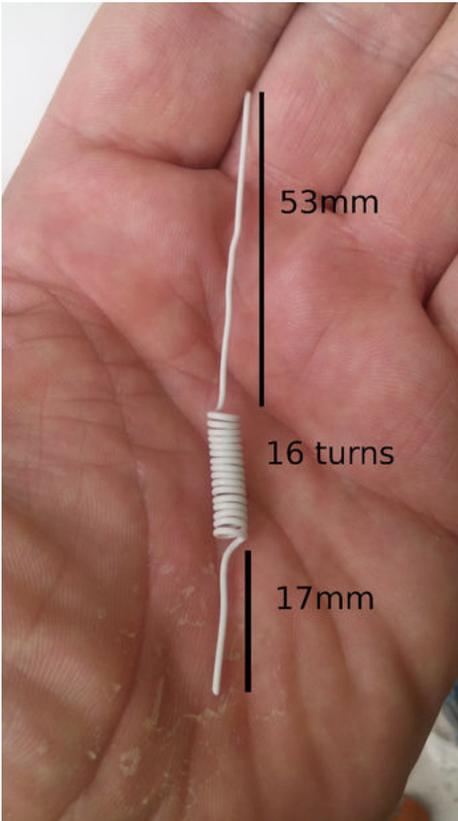
Note that the Plus (VCC) supply is 3.3 volts for the receiver and 5 volts for the transmitter. This is because the ESP8266 expects 3.3 volts on its pins so the voltage supplied to the input pin reading the received data needs to be 3.3 volts. 5 Volts would fry the 8266.

We can give the transmitter 5 Volts on the supply as this voltage does not get to the 8266. The greater voltage gives the transmitter more power thus more range.

Perhaps a picture makes more sense.



Antenna



<https://www.instructables.com/id/433-MHz-Coil-loaded-antenna/> and

[Original PDF](#)

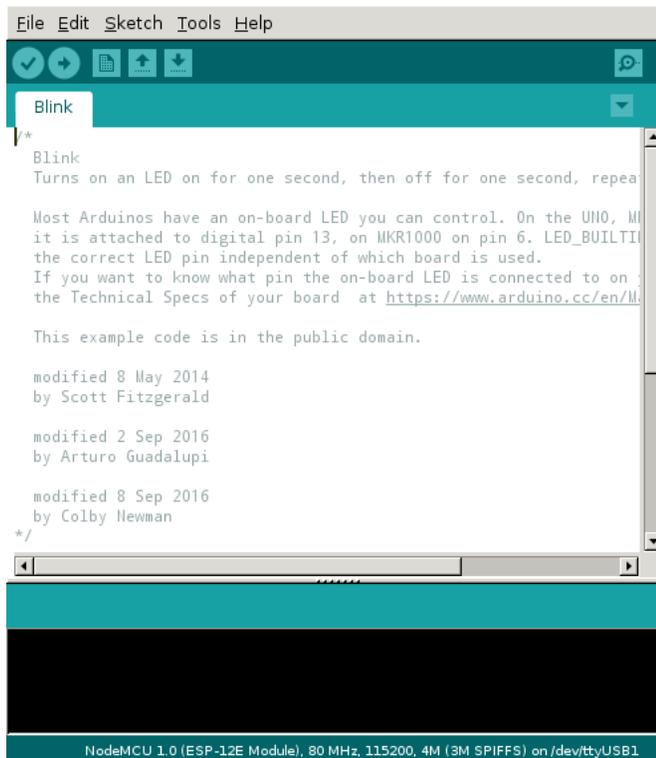
Software to make it work - Transmitter and web server software.

Arduino IDE

1. You can do an apt-get install arduino, but it is an older version. It is better to install from the source:
2. <https://www.arduino.cc/en/Main/Software>
3. Download the ARM Version if you want to run this on the Pi. They will you ask you to donate. You may want to consider doing this.
4. Put the file somewhere where you usually put downloaded executables. I put mine in a directory uniquely called "arduino".
5. Un-compress it using tar. `tar -xvf`.
6. This will create a new directory, arduino-1.8.3. Or perhaps something larger as time passes.
7. The contents of this directory are:

```
-rwxr-xr-x  1 pi pi      946 May 31 16:58 arduino
-rwxr-xr-x  1 pi pi 3293964 May 31 16:58 arduino-builder
drwxr-xr-x 13 pi pi    4096 May 31 16:58 examples
drwxr-xr-x  4 pi pi    4096 May 31 16:58 hardware
-rwxr-xr-x  1 pi pi   9713 May 31 16:58 install.sh
drwxr-xr-x  4 pi pi    4096 Apr  7 16:10 java
drwxr-xr-x  4 pi pi    4096 May 31 16:58 lib
drwxr-xr-x 21 pi pi    4096 May 31 16:58 libraries
drwxr-xr-x  6 pi pi    4096 May 31 16:58 reference
-rw-r--r--  1 pi pi  82397 May 31 16:58 revisions.txt
drwxr-xr-x  4 pi pi    4096 May 31 16:58 tools
drwxr-xr-x  3 pi pi    4096 Mar 20 17:30 tools-builder
-rwxr-xr-x  1 pi pi     86 May 31 16:58 uninstall.sh
```

8. You can run the install script `sudo install.sh` if you wish, but you don't have to. You can just run the arduino program. `./arduino` and after a bit you will get something like the following.



ESP8266 Board Manager for Arduino IDE

- Install the ESP8266 Board Manager:
 - In the Arduino IDE click File, Preferences and enter/paste the following URL into the "Additional Boards Manager URLs" box:
`http://arduino.esp8266.com/stable/package_esp8266com_index.json`
 - Then press OK.
 - Go to Tools > Board > Boards Manager... Then select "esp8266 by ESP8266 Community"

Libraries we need

These are part of the ESP 8266 base - we don't need to add these.

<ESP8266WiFi.h> Supports the wifi connection

<WiFiClient.h> Supports a wifi connected client.

<ESP8266WebServer.h> Web Server Module

We need to add these:

<RCSwitch.h> 433 MHz receiver/transmitter

- <https://github.com/sui77/rc-switch> // 433MHz library.

<DHT.h> Temperature/humidity sensor - [ask about this later if you care.](#)

- <https://github.com/adafruit/DHT-sensor-library> // Adafruit DHT Humidity & Temperature Unified Sensor Library
- https://github.com/adafruit/Adafruit_Sensor // Adafruit Unified Sensor Driver

Add library by: (<https://www.arduino.cc/en/Guide/Libraries>)

- Download the desired library and save it somewhere.
- In the Arduino IDE go to: sketch > Include Library > Add .ZIP Library...
- Select the file you downloaded.

Getting the Switch Codes

Normally we simply push a button but as seen above we have software to do this for us. But what does the software have to send



The RCSwitch.h library comes with an example that you will find in the directory:

~/Arduino/libraries/rc-switch-master/examples/ReceiveDemo_Advanced

There are two file in here:

- output.ino
- ReceiveDemo_Advanced.ino

The ReceiveDemo_Advanced program calls the output code. If you run this on the Arduino, the Receive... program seems to find the output program. On the ESP8266 it does not. So I just combined the two into a single file. That works.

[receive.c.pdf](#)

Output from the receive program

I used the binary values. Bolded here.

To get these I held the remote next to the receiver and pushed a button and got:

Decimal: 5147919 (24Bit) Binary: 010011101000110100001111 Tri-State: not applicable PulseLength: 308 microseconds
Protocol: 1
Raw data:
9570,306,883,880,303,295,890,294,888,876,310,871,313,869,315,285,897,870,314,288,892,293,888,295,889,293,890,874,31

Decimal: 5147915 (24Bit) Binary: 010011101000110100001011 Tri-State: not applicable PulseLength: 308 microseconds
Protocol: 1
Raw data:
9566,308,878,888,294,304,881,303,884,883,301,881,303,880,303,295,887,880,304,295,890,293,889,294,887,296,891,876,30

Copy the binary values and put them into the transmit program discussed above. The array variable sockets contains the codes.

```
char sockets[20][25] = {  
  "010101101000110100001111", // On  
  "010101101000110100001110", // Off ...  
  "010101101000110100001101",  
  "010101101000110100001100",  
  "010101101000110100001011",  
  "010101101000110100001010",  
  "01010110100011010000111",  
  "01010110100011010000110",  
  "01010110100011010000101",  
  "01010110100011010000100",  
  "010011101000110100001111",  
  "010011101000110100001110",  
  "010011101000110100001101",  
  "010011101000110100001100",  
  "010011101000110100001011",  
  "010011101000110100001010",  
  "01001110100011010000111",  
  "01001110100011010000110",  
  "01001110100011010000101",  
  "01001110100011010000100",  
};
```

There are 10 switches in the system and 20 codes. On and off for each switch.

About the temperature/humidity sensor. It is an DHT22. [More in another presentation.](#)

Reference

