

# ANAVI Miracle Controller

---

**An ESP8266-powered, open source, Wi-Fi dev board to control two 5 V or 12 V LED strips**

---

## DISCLAIMER

---

ANAVI, the ANAVI logo and combinations thereof, are registered trademarks of Leon Anavi. Other product names may be trademarks of others and the rights belong to their respective owners.

The information in this document is provided in connection with Anavi products. No license, express or implied or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Anavi products.

This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0). this license, visit <https://creativecommons.org/licenses/by-sa/4.0/> (<https://creativecommons.org/licenses/by-sa/4.0/>).

ANAVI Miracle Controller hardware design is licensed under a Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0).

The software examples are released under MIT and the rest of the software is available under GPLv3.

It is possible that the pictures in this manual differ from the latest revision of the board.

The product described in this document is subject to continuous development and improvements. All particulars of the product and its use contained in this document are given by Anavi in good faith. However all warranties implied or expressed including but not limited to implied warranties of merchantability or fitness for purpose are excluded. This document is intended only to assist the reader in the use of the product. Anavi shall not be liable for any loss or damage arising from the use of any information in this document or any error or omission in such information or any incorrect use of the product.

This evaluation board/kit is intended for use for engineering development, demonstration, or evaluation purposes only and is not considered by Anavi to be a finished end-product fit for general consumer use. People handling the product must have electronics training and observe good engineering practice standards. As such, the goods being provided are not intended to be complete in terms of required design-, marketing-, and/or manufacturing-related protective considerations, including product safety and environmental measures typically found in end products that incorporate such semiconductor components or circuit boards.

There is no warranty for the design materials and the components used to create ANAVI Miracle Controller. There are considered suitable only for ANAVI Miracle Controller.

---

# CHAPTER 1: Overview

---

## Introduction

ANAVI Miracle Controller is an open source hardware Wi-Fi development board powered by the ESP8266 and designed to control two 5 V and 12 V addressable LED strips simultaneously. It supports popular LEDs including Neopixel, WS2811, WS2812B, TM1809, etc. It also has a dedicated slot for a mini OLED I<sup>2</sup>C display and slots for up to three additional I<sup>2</sup>C sensor modules. Furthermore, there are UART pins for easy flashing of custom software and an extra GPIO for connecting external peripherals. All these features make ANAVI Miracle Controller appropriate for developers, makers, students, and open source enthusiasts interested in home automation.

ANAVI Miracle Controller is designed with the free and open source electronics design automation suite KiCAD (<http://kicad-pcb.org/>). No soldering is required. You can assemble ANAVI Miracle Controller with your bare hands and a screwdriver.

## Features & Specifications

- **Microcontroller:** Tensilica L106 32-bit processor (ESP8266)
- **Wireless:** Wi-Fi 802.11 b/g/n
- **Voltage Connectivity:** Simultaneous support for two 5 V or 12 V addressable LED strips
- **LED Compatibility:** Open source Arduino sketch with FastLED library will support various strips: WS2812, WS2812B, WS2811, TM1804, etc.
- **Peripheral Support:** Mini OLED display, UART pins, button, three slots for I<sup>2</sup>C sensors
- **Software Compatibility:** Arduino IDE, PlatformIO, Home Assistant, OpenHAB 2, MQTT, and any modern web browser
- **Open Source Certification:** Open Source Hardware Association (OSHW) BG000050
- **Dimensions:** 75 mm x 40 mm
- **Upgradable:** The software can be upgraded, either using an USB to UART module, or (if configured) over-the-air

## Target Market

ANAVI Miracle Controller is a certified open source hardware development board for customers interested in home automation, software development and Internet of Things. The board is appropriate for embedded programming enthusiasts, open source supporters, students as well as web and/or mobile app developers. The main usage of the board is home automation.

## Board Version

Revision 1.1 of ANAVI Miracle Controller was used while writing this document. It is possible that it is outdated so it is always recommended to check the latest sources from the GitHub page of the board.

---

# CHAPTER 2: Getting Started

---

## Electrostatic Warning

ANAVI Miracle Controller is shipped in a protective bag. It must **NOT** be exposed to high electrostatic potentials. A grounding strap or similar protective device should be worn when handling the board. Avoid touching the component pins or any other metallic element.

## Requirements

In order to setup ANAVI Miracle Controller the following items are required:

- 1 or 2 addressable LED strips, supported by the popular FastLED Arduino library (with three wires), such as WS2812, WS2812b, WS2811, TM1804, etc.
- 5V or 12V center positive power supply matching the requirements of the LED strips. Set the pin jumper on ANAVI Miracle Controller according to the voltage of the power supply. Mismatching voltage will damage ANAVI Miracle Controller and/or the LED strips

It is recommended to use 2A (2000mA) or higher power supply. For **example** for the 5V WS2812B LED strips included in the kits: MEAN WELL GST18E05-P1J (<https://www.meanwell-web.com/en-gb/ac-dc-industrial-wall-mount-adaptor-output-5vdc-at-gst18e05--p1j>) .

## Supported Peripherals

ANAVI Miracle Controller has a terminal block for attaching up to 2 addressable LED strips, I2C slot for mini OLED display as well as up to 3 slots I2C sensors.

### LED strips

All addressable LED strips currently supported by the popular FastLED Arduino library (<http://fastled.io/>) (with three wires) are supported by ANAVI Miracle Controller, including WS2812, WS2812b, WS2811, and TM1804. To prevent spikes, a 300 to 500 Ohm choke resistor might be required at the data wire. Some LED strips, including those provided with our kits, already have this resistor built-in.

Strips provided in ANAVI Miracle Controller kits are WS2812B strips with ten LEDs each, appropriate connectors, and built-in 470 Ohm choke resistor.

You can attach 1 or 2 addressable LED strips to ANAVI Miracle Controller. Both attached strips must be of the same kind.

Non-addressable LED strips are not appropriate for ANAVI Miracle Controller. If you need to control 12V non-addressable RGB LED strip have a look at our other open source product ANAVI Light Controller.

## Sensors

You can attach various I2C sensor modules to ANAVI Miracle Controller. The officially supported I2C add-on sensor modules are for:

- Light (BH1750)
- Temperature and humidity (HTU21D)
- Color and gesture recognition (APDS-9960)
- Temperature and barometric pressure (BMP180)

You may also attach any other I2C sensors but you will have to take care of their software integration.

### Light Sensor

The official light I2C sensor for ANAVI Miracle Controller is BH1750.

Using 4 Dupont jumper wires connect BH1750 to one of the 3 I2C slots on ANAVI Miracle Controller as follows:

BH1750	ANAVI Miracle Controller
VCC	3.3V
GND	GND
SCL	SCL
SDA	SDA

### I2C External Temperature & Humidity Sensor

The official external temperature and humidity add-on sensor for ANAVI Miracle Controller is HTU21 (SHT21). This is I2C sensor capable of measuring both humidity and temperature.

Using 4 Dupont jumper wires connect HTU21 to one of the 3 I2C slots on ANAVI Miracle Controller as follows:

HTU21	ANAVI Miracle Controller
VIN	3.3V

HTU21	ANAVI Miracle Controller
GND	GND
SCL	SCL
SDA	SDA

### Color & Gesture Recognition Sensor

The official I2C sensor for RGB color and gesture detection on ANAVI Ligth pHAT is APDS-9960.

Using 4 Dupont jumper wires connect APDS-9960 to one of the 3 I2C slots on ANAVI Miracle Controller as follows:

APDS-9960	ANAVI Miracle Controller
VIN	3.3V
GND	GND
SCL	SCL
SDA	SDA

### Temperature & Barometric Pressure Sensor

The official temperature and barometric pressure sensor for ANAVI Miracle Controller is BMP180. This is I2C sensor capable of measuring both temperature and barometric pressure.

Using 4 Dupont jumper wires connect BMP180 to one of the 3 I2C slots on ANAVI Miracle Controller as follows:

BMP180	ANAVI Miracle Controller
VIN	3.3V
GND	GND
SCL	SCL
SDA	SDA

## Assembly

You can assemble ANAVI Miracle Controller with your bare hands and a screw driver following the steps below:

- Add addressable LED strips to your ANAVI Miracle Controller.
- Optionally, add I2C sensor modules.
- Optionally, add I2C mini OLED display.
- Optionally, you may also assemble the simple acrylic case included in the advanced and developer kit by removing the protective film and mounting screws.
- Use 5V or 12V power supply with standard 2.1x5.5 mm DC barrel jack matching the requirements of the addressable LED strips. **Set the jumper pin on ANAVI Miracle Controller to 5V or 12V depending on the power supply.**

## Powering ANAVI Miracle Controller

ANAVI Miracle Controller has been tested using various 5V power supplies and USB to microUSB cables. It is recommended to use 5V power supply with 1A or higher current output. Make sure that you are using a 5V power supply from a trusted supplier. Cheap, untested power supplies can be risky and unreliable.

## Configure ANAVI Miracle Controller

Video tutorial for getting started with ANAVI Light Controller, same procedure for ANAVI Miracle Controller ([https://www.youtube.com/watch?v=Y\\_81CuuGm0Y](https://www.youtube.com/watch?v=Y_81CuuGm0Y))

To turn on ANAVI Miracle Controller, set the jumper pin to 5 V or 12 V and plug appropriate power supply (matching the jumper pin configuration) in to the standard 2.1x5.5 mm DC barrel jack. As soon as the board is turned on for the very first time it will create a temporary WiFi access point. Connect to it from your computer, smartphone or tablet.

The default software of ANAVI Miracle Controller has a captive portal which guides you through the configurations. As you see in the video you have to select your WiFi network and enter a password if it is not open.

Our open source software relies on the machine to machine communication protocol MQTT to report data from the sensors. You can connect to your own MQTT broker or just leave the default configurations and connect to the public broker as shown in the video.

Recent versions of the firmware support Home Assistant MQTT Discovery (<https://www.home-assistant.io/docs/mqtt/discovery/>) which makes it very convenient to add ANAVI Miracle Controller to Home Assistant. For this to work, you need to enter the sensor name (such as "Master Bedroom") that will be used for this Thermometer on the setup screen. See Chapter 5 for more details.

Recent versions of the firmware support Over-the-Air upgrades of the software. To enable this, you need to specify the IP address or DNS domain name of a web server that hosts the firmware. See Chapter 3 for more details.

It very important to copy the machine ID. Later it is needed to identify your device.

When you are ready just click save. If you have entered valid credentials in a moment ANAVI Miracle Controller will connect to your WiFi network and the configured MQTT broker. This

way the configuration is complete and ANAVI Miracle Controller will turn off its temporary WiFi access point. After that your device, for example smartphone as in the video, will automatically connect again to your WiFi network.

To verify that ANAVI Miracle Controller is up and running open a modern web browser and visit [demo.anavi.technology](http://demo.anavi.technology). Enter your machine ID. Check the advanced settings only if you are not using the default public MQTT broker. Click connect.

## Reset to Factory Defaults

If you decide to change the settings of ANAVI Miracle Controller you need to reset the board and configure it again.

To reset ANAVI Miracle Controller press the button and hold it for approximately 6 seconds until the red light stops blinking. After resetting the board there will be a steady red light that indicates that the temporary WiFi access point is on and you can proceed with the configuration again. The reset button works once the ANAVI Miracle Controller has successfully connected to a WiFi network. If you are running a recent enough version of the firmware, it also works for 2 seconds after you power on the device (while the red LED is blinking quickly).

---

# CHAPTER 3: Software Development

---

## Default Firmware

By default ANAVI Miracle Controller comes with this free and open source Arduino sketch (<https://github.com/AnaviTechnology/anavi-miracle-controller-sw>) .

## USB to UART Module

For uploading firmware to ANAVI Miracle Controller you need USB to UART module. All kits include CP2102 which out of the box on GNU/Linux distributions. Drivers for MS Windows and Mac OS X are available at [silabs.com](http://silabs.com) (<https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>) .

## Setting up the Arduino IDE

1. Install the Arduino IDE following the instructions from from <https://www.arduino.cc/en/Main/Software> (<https://www.arduino.cc/en/Main/Software>)
2. Add the ESP8266 board package: In **File > Preferences** input [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json) ([http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)) into the Additional Board Manager URLs field.

3. In **Tools > Board ... > Boards manager** find and add the **ESP8266** package. Now "Generic ESP8266" should be an option in the **Tools > Boards** menu.
4. In **Tools > Flash Size:** select **4M (1M SPIFFS)**
5. Go to **Sketch > Include Library > Manage Libraries...** and include the following dependencies of the default firmware for ANAVI Miracle Controller:
6. WiFiManager by tzapu (version 0.12.0)
7. ArduinoJson by Benoit Blanchon (version 6.11.2)
8. PubSubClient by Nick O'Leary (version 2.7.0)
9. Adafruit HTU21DF Library by Adafruit (version 1.0.1)
10. Adafruit APDS9960 Library by Adafruit (version 1.0.5)
11. U8g2 by oliver (version 2.23.18)
12. Adafruit Unified Sensor by Adafruit (version 1.0.2)
13. Adafruit BMP085 Unified (version 1.0.0)
14. FastLED (version 3.2.6)

*Note: Issues might be experienced if using different versions of the libraries.*

## Flashing Custom Firmware

Follow the steps below to compile and flash custom firmware on ANAVI Miracle Controller from Arduino IDE:

1. To flash the firmware from Arduino IDE select **Tools > Generic ESP8266 Module** (Flash mode: DIO, Flash frequency: 40MHz, CPU frequency: 80MHz, Flash size: 4M, Debug port: Disabled, Debug level: None, Reset method: ck, Upload speed: 115200, Port: /dev/ttyUSB0). Set the flash size to 4M (1M SPIFFS). You might need to adjust the port if your USB to serial debug cable is connected on a different port.
2. After that press load an Arduino sketch. A simple blinking LED example is available at [GitHub](https://github.com/AnaviTechnology/anavi-examples/blob/master/anavi-light-controller/anavi-blinking-led/anavi-blinking-led.ino) (<https://github.com/AnaviTechnology/anavi-examples/blob/master/anavi-light-controller/anavi-blinking-led/anavi-blinking-led.ino>)
3. In Arduino IDE click Verify/Compile (Ctrl+R)
4. Connect ANAVI Miracle Controller to the USB to serial debug board: GND to GND, TX cable to RX of ANAVI Miracle Controller and RX cable to TX of ANAVI Miracle Controller.
5. In Arduino IDE click Upload (Ctrl+U)
6. Press and **hold** the RESET button on ANAVI Miracle Controller. Plug the 5V power supply in the jack of ANAVI Miracle Controller (without releasing the RESET button).
7. In Arduino IDE verify that the upload has been started. Hold RESET until the upload completes.

The output in Arduino IDE for successful flashing is:



```

[-] Archiving built core (caching) in: /tmp/arduino_cache_954939/core/core_esp8266_esp
8266_generic_CpuFrequency_80,FlashFreq_40,FlashMode_dio,UploadSpeed_115200,FlashSize_512K64,ResetMethod_ck,Debug_Disabled,DebugLevel_None_____1c2aa2b3da66da225b39c9bf
ab6531e5.a
Sketch uses 224949 bytes (51%) of program storage space. Maximum is 434160 bytes.
Global variables use 31756 bytes (38%) of dynamic memory, leaving 50164 bytes for
local variables. Maximum is 81920 bytes.
Uploading 229104 bytes from /tmp/arduino_build_904122/anavi-blinking-led.ino.bin to
flash at 0x00000000
..... [
35% ]
..... [
71% ]
..... [
100% ]

```

If you have flashed the blinking LED example, D1 on ANAVI Miracle Controller will start blinking.

Please have a look at the YouTube video that shows the exact steps for compiling and uploading an Arduino sketch to ANAVI Miracle Controller (<https://www.youtube.com/watch?v=HMikPuz0Zjs>).

**Note:** you have to be quick between step 5 and 6. Remember to press and **hold** RESET until the upload completes.

## OTA Firmware Upgrade

---

Recent versions of the official firmware also support OTA firmware updates by putting a firmware binary on a web server and then triggering the update via a MQTT message. This can be very convenient, as you do not have to connect the device to your computer.

### Check support

If the captive portal contained a "OTA server" field, your software supports OTA upgrades.

If you are unsure if your current firmware supports OTA update, you can first check if your ANAVI Miracle Controller is already setup to receive MQTT commands.

To do this, send a MQTT message to the topic "cmnd/[deviceId]/line1" with message body "test", for example using mosquitto\_pub:

```

[-] mosquitto_pub -h [mqttserver] -p 1883 -t cmnd/b5cfb5cfb5cfb5cfb5cfb5cfb5cf/line1 -u [mqttuser] -P [mqttpass] -m "test"

```

When this message causes the first line of your ANAVI Miracle Controller display to change to "test", OTA updates should be supported. If not (after double checking that you used the

correct topic and deviceId), please first update to a more recent firmware version.

(To reset the line again to the default display, just send an empty message to the same topic).

## Enabling OTA upgrades

For security reasons, OTA upgrades are disabled by default. There are two ways to enable them: either enter an IP address or domain name on the captive portal screen (see Chapter 2), or edit the official source code and define OTA\_SERVER as explained in the comment. (You can even do both, if you want the Thermometer to accept upgrades from two sources.)

The web server can be located anywhere, but for improved security it is best if it is located on your local network, and no untrusted persons should be able to edit any files at all on the server. Don't host your firmware images on a shared server!

## Build binary firmware image

Using the arduino build setting above, instead of clicking Upload (Ctrl+U), use Menu Sketch -> Export compiled binary (CTRL+ALT+S). A .bin file will be created in the same folder as the .ino file. Copy this file to the HTTP webserver you specified when you enabled OTA upgrades.

## Trigger update

The device listens to MQTT messages in topic "cmd/[deviceId]/update" and expects messages to be in the format

```
⌘ {"file":"/[filePath]", "server": "[server IP or DNS name]"}
```

For example, if you put the file at <http://192.168.100.46/anavi.bin> (<http://192.168.100.46/anavi.bin>) :

```
⌘ {"file":"/anavi.bin", "server": "192.168.100.46"}
```

If your server does not listen on port 80, you can provide the port, for example for <http://192.168.100.46:8080/anavi.bin> (<http://192.168.100.46:8080/anavi.bin>) :

```
⌘ {"file":"/anavi.bin", "server": "192.168.100.46", "port": 8080}
```

Example for sending an update message using mosquitto\_pub:

```
⌘ mosquitto_pub -h [mqttserver] -p 1883 -t cmd/b5cfb5cfb5cfb5cfb5cfb5cfb5cf/upd  
ate -u [mqttuser] -P [mqttpass] -m "{\"file\":\"anavi.bin\", \"server\":\"192.168.  
100.46\", \"port\": 8080}"
```

**Note:** To ensure successful update, please specify the file with leading slash (/), use an HTTP web server, not HTTPS, and -- when run from the command line -- escape quote characters

On receipt of this message, ANAVI Miracle Controller will download the file from the specified server, update the firmware and restart. As with the normal firmware update, your saved settings such as WiFi and MQTT server settings should remain as configured.

**Note:** *After the update, you may find that readings from the DHT22 sensor no longer work and temperature and humidity are shown as "o". In this case, please powercycle the device (unplug power until display turns off, then plug power in again)*

## Using signed builds

If you do OTA upgrades, and especially if you do so over the Internet, you are strongly encouraged to use Signed Updates ([https://arduino-esp8266.readthedocs.io/en/2.5.0/ota\\_updates/readme.html#advanced-security-signed-updates](https://arduino-esp8266.readthedocs.io/en/2.5.0/ota_updates/readme.html#advanced-security-signed-updates)) . Follow the link to see all the details.

If you are running the Arduino IDE on Linux or Mac it is very easy to enable automatic signing. Just run these commands in the sketch directory:

```
❏ openssl genrsa -out private.key 2048
  openssl rsa -in private.key -outform PEM -pubout -out public.key
```

Once this is done, builds will be automatically signed. In addition to the .bin file, a .bin.signed file will also be created. You should see messages similar to these when you compile your sketch:

```
❏ Enabling binary signing
  Signed binary: /tmp/arduino_build_710055/sketch_apr03a.ino.bin.signed
```

The exact file name will change depending on your installation. Use the signed binary instead of the .bin file. Once you have upgraded your ANAVI Miracle Controller using a signed build, OTA upgrades will reject any attempt to use an unsigned build, a build signed using the wrong key, or a build that has been tampered with. (If necessary, you can still load an unsigned firmware using the USB to UART module, and then you will be able to use unsigned builds for OTA upgrades as well.)

Be sure the back up the private.key file, and make sure nobody can get hold of it.

---

## CHAPTER 4: Temperature scale: Celsius vs Fahrenheit

---

The temperature scale of the default open source firmware of ANAVI Miracle Controller can be configured at the initial setup of the device. By default it is set to Celsius. Manually type in **fahrenheit** to change it.

After the initial setup the temperature scale can be still changed with a single MQTT message

with topic **cmnd//tempformat** (where must be replaced with the unique value for your board) and a JSON payload:

```
➤ { "scale": "fahrenheit" }
```

To switch back to Celsius, send the same message and just edit the JSON payload by changing **fahrenheit** to **celsius**.

For example, if using the mosquitto\_pub application from a command-line terminal and machine id 11111111111111111111111111111111:

```
➤ mosquitto_pub -h iot.eclipse.org -d -p 1883 -t  
  "cmnd/11111111111111111111111111111111/tempformat" -m '{ "scale":  
  "fahrenheit" }'
```

## CHAPTER 5: Home Automation Platforms

---

This chapter provide guidelines how to connect ANAVI Miracle Controller to popular home automation platforms like Home Assistant, OpenHab 2, Thinkspeak, etc.

### Home Assistant

Home Assistant (<https://home-assistant.io/>) is a free and open-source home automation platform running on Python 3 with more than 1200 components for integration with popular Internet of Things.

ANAVI Miracle Controller can be easily integrated in Home Assistant using the component MQTT Light Component (<https://www.home-assistant.io/integrations/light.mqtt/>) . This component supports JSON in the payload of the MQTT messages. To use it, in **configuration.yaml** specify MQTT broker and register the sensor with the corresponding MQTT topic, for example:

- Configure MQTT broker:

```
➤ mqtt:  
  broker: 127.0.0.1
```

- Register the MQTT component:

```
➤
```

MQTT Discovery can simplify this process. To use this, specify this in **configuration.yaml**:

```
➤ mqtt:  
  broker: 127.0.0.1  
  discovery: true  
  discovery_prefix: homeassistant
```

# OpenHAB 2

OpenHAB (<https://www.openhab.org/>) is a popular free and open source home automation software. Major version 2 of OpenHAB is significantly different for version 1. It is written in the JAVA programming language using the Eclipse Smart Home framework. Version 2.4 of OpenHAB adds several new MQTT features (<https://www.openhab.org/blog/2018-12-16-mqtt-arrives-in-the-modern-openhab-2-x-architecture.html>) . This YouTube video explains the exact steps how to integrate ANAVI Miracle Controller in OpenHAB 2 (<https://www.youtube.com/watch?v=tjuXcqKG1Kc>) .

Follow the steps below to integrate ANAVI Miracle Controller in OpenHAB 2:

## OpenHAB Step 1: Install MQTT Binding

Open the **Paper UI** web interface of OpenHAB 2. Go to **Add-ons > Binding**. Find **MQTT Binding** and click **Install**

## OpenHAB Step 2: Configure MQTT Broker

Again in Paper UI of OpenHAB 2, go to **Configuration > Things**. Click the button to add a new thing. Select **MQTT Thing Binding**. After that click **ADD MANUALLY**. Click **MQTT Broker**. Enter the domain or the IP address of the MQTT broker, the port and any other needed credentials.

After adding the MQTT Broker to OpenHAB go to **Configuration > Things** and verify that the broker appears as online. If it is not online double check the configuration and try again.

## OpenHAB Step 3: Install JavaScript Transformations

Go to **Add-ons > Transformations**. Find **JavaScript Transformation** and click **Install**

## OpenHAB Step 4: Download JavaScript Files

Download **temperature.js** and **humidity.js** from GitHub (<https://github.com/AnaviTechnology/anavi-examples/tree/master/openhab2/transform>) and deploy both files in directory **\$OPENHAB\_CONF/transform**.

For example, on Raspberry Pi with OpenHABian variable **\$OPENHAB\_CONF** is set to **/etc/openhab2** so the whole path is **/etc/openhab2/transform**:

```
❏ cd $OPENHAB_CONF/transform
   wget https://raw.githubusercontent.com/AnaviTechnology/anavi-examples/master/openhab2/transform/temperature.js
   wget https://raw.githubusercontent.com/AnaviTechnology/anavi-examples/master/openhab2/transform/humidity.js
```

**Note:** If you prefer the temperature in Fahrenheit, set variable **celsius** to **false** in the first line of **temperature.js**. This way the JavaScript will automatically convert the temperature from Celsius to Fahrenheit when showing it in the user interfaces of OpenHAB.

## OpenHAB Step 5: Add ANAVI Miracle Controller as MQTT Generic Thing

In **Paper UI** web interface go to **Configuration > Things**. Click the button to add a new thing and select **MQTT Thing Binding**.

On the next screen it is very important to click **ADD MANUALLY**. After that select **Generic MQTT Thing**. Set the name, for example to **ANAVI Miracle Controller**. Set the MQTT broker configured on step 2 for **Bridge Selection**.

---

## CHAPTER 6: Schematics

---

### Pinout

The components of ANAVI Miracle Controller relies on ESP8266 (ESP-12 module) and utilizes the following pins:

Component	Pins	Arduino Pin ID
I2C	13, 14	
Indication LED (D1)	4	16
LED1 data	6	12
LED2 data	5	14
Reset button (SW1)	12	0
Extra GPIO	11	2
UART	15, 16	

### I2C

The sensors that can be connected to ANAVI Miracle Controller communicate with a host microcontroller via a communications standard called **I2C** (Inter-Integrated-Circuit). I2C uses two wires, labelled SDA (Serial Data) and SCL (Serial Clock). To function properly, I2C requires a pullup resistor on each of those lines therefore ANAVI Miracle Controller includes two 4.7kohm resistors labelled as R2 and R3. If for one reason or another you need to disable the I2C pullup resistors remove R2 and R3.

---

## CHAPTER 7: Frequently Asked Questions

# (FAQ)

---

## What power supply do I need?

ANAVI Miracle Controller supports 5 V or 12 V input voltages. To turn on the board, set the jumper pin to 5 V or 12 V and plug appropriate power supply in to the standard 2.1x5.5 mm DC barrel jack. **Only use power supply matching the configuration of the jumper and the attached addressable LED strip.** Mismatching the power supply voltage will damage the board and the LED strips.

ANAVI Miracle Controller has been tested using 5 V power supplies from MEAN WELL GST60A05-P1J (<https://www.meanwell-web.com/en-gb/ac-dc-industrial-desktop-adaptor-output-5vdc-at-6a-gst60a05--p1j>) and 12 V power supplies from SUNNY Computer Technology Europe such as SYS1530-1212-W2E (<https://www.sunny-euro.com/en/products/sys1530-1212-w2e-europe-2-1x5-5x11-s-2wc-1-4m-4-5ft>) .

Make sure that you are using a 5V or 12 V power supply from a trusted supplier. Cheap, untested power supplies can be risky and unreliable.

## Why isn't a power supply included?

We were unable to source power supplies with universal plugs for the US/EU/UK at an acceptable. Nowadays, 5V power supplies with a microUSB connector are a commodity item because of most smartphone and tables, so you should be able to easily find a suitable unit online or at your local electronics store.

## Which addressable LED strips are included in the kits?

Strips provided in ANAVI Miracle Controller kits are 5V WS2812B strips with ten LEDs each, appropriate connectors, and built-in 470 Ohm choke resistor. Use these WS2812B strips only with 5V power supply and set the jumper on ANAVI Miracle Controller to 5V!

## Can I remotely control ANAVI Miracle Controller from a web browser on my smartphone, tablet, or laptop?

Yes, you can use our demo website (<http://demo.anavi.technology/>) or easily integrate ANAVI Miracle Controller in your instance of the popular open source platform Home Assistant (<https://home-assistant.io/>) as an MQTT Sensor component.

## Is ANAVI Miracle Controller an open source project?

Yes, ANAVI Miracle Controller is an open source hardware project powered and created with free and open source software. The hardware designs are available at GitHub under CC BY-SA 4.0 license (<https://github.com/AnaviTechnology/anavi-thermometer>) . All schematics, documents, and source code files are available at our GitHub repositories (<https://github.com/AnaviTechnology/>) .

## Is ANAVI Miracle Controller certified?

Yes, ANAVI Miracle Controller revision 1.1 has been certified by the Open Source Hardware

### **Does ANAVI Miracle Controller use the ESP8266?**

Yes, ANAVI Miracle Controller is based on the ESP8266.

### **Can I flash different firmware to ANAVI Miracle Controller?**

Yes, using a USB to serial cable, you can flash custom firmware built from your own source code.

### **Is ANAVI Miracle Controller compatible with Arduino IDE?**

Yes, ANAVI Miracle Controller is compatible with Arduino IDE. You can easily upload your own Arduino sketches to the board.

### **How can I get involved and help?**

Buy any of the available perks, get your hands on the ANAVI Miracle Controller, contribute to our GitHub repositories, and become part of our open source community!

---

## **CHAPTER 8: Revision History**

---

### **Document Revision**

Date	Changes	Modified pages	Author
21 December 2019	Initial release	All	Leon Anavi

### **ANAVI Miracle Controller Revision**

Revision	Notable changes
1.1	Stable product

### **See Also**

For more information please visit [anavi.technology](http://anavi.technology/) (<http://anavi.technology/>) and our GitHub repositories (<https://github.com/AnaviTechnology>) . If you have any questions or enquiries please contact us through Facebook (<https://www.facebook.com/AnaviTechnology/>) , Twitter (<https://twitter.com/AnaviTechnology>) or email (<mailto:info@anavi.technology>) .

---



