

Monitair System Software

Version 1.0 – Rob Miles

This document describes the installation and configuration of the Monitair system software. The software can be obtained from GitHub in source and binary form.

Monitair Hardware Platform

The software will run without modification on a device with the following configuration:

- Wemos D1 Mini 8266 processor
- Bosch BM280 temperature sensor connected to I2C
- NeoPixel ring (optional) for air quality visualisation and status display
- SDS011 or ZPH01 air quality sensor (the sensor type is automatically detected)

The configuration of the peripheral devices along with the network and MQTT settings can be performed within the software.

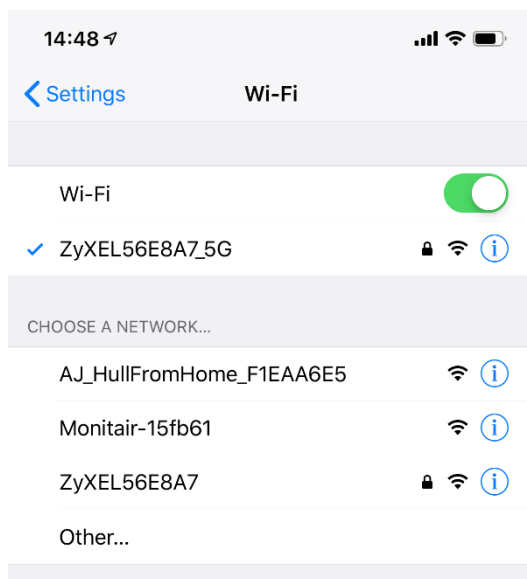
Getting started

A new device will need to be configured. At the very minimum you will need to set the name of the WiFi network, the password for that network and the password for the Connected Humber MQTT broker. There are two ways that you can configure these settings:

- You can set the Monitair device into *configuration* mode. In this mode it runs an access point that you can connect to with any WiFi enabled device. You can then enter the settings via a browser on your device.
- You can connect the Monitair device to a computer via a usb connection and use a serial console to enter the settings.

Which approach you use depends entirely on how you want to work, and what devices you have available. If you are just using your mobile phone you can use the configuration mode. If you already have a PC and a connecting cable you might find it easier to use the serial connection. Note that this means that you can use the serial console to configure a device directly after you have loaded the Arduino code into it. You could even write a simple program to configure the device automatically.

Using Configuration Mode



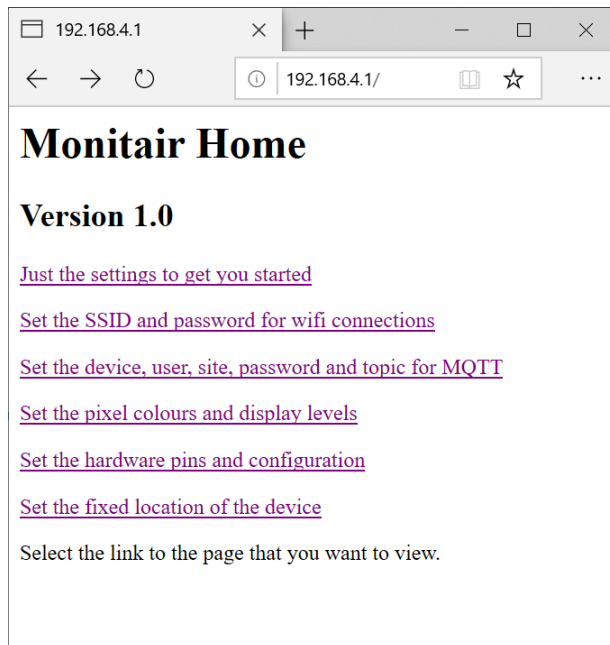
A Monitair device is fitted with a hardware button that can be pressed to view the device status. This button can also be used to select configuration mode. If the device is powered on with this button held down it will enter configuration mode. In this mode it will display a ring of magenta pixels on the neopixel display (if connected) and then display two green pixels as the WiFi and the configuration web server start running.

You can then connect to the device as you would any other WiFi host. The screenshot on the left shows how I would connect to a sensor using my iPhone. The device has the name Monitair followed by a unique set of hex values.

You can use the equivalent WiFi menu on your computer to connect to the sensor. There is no password on the network connection.

Once you have connected to the Monitair device you can adjust the settings of the device from your browser. Start your browser and navigate to the address:

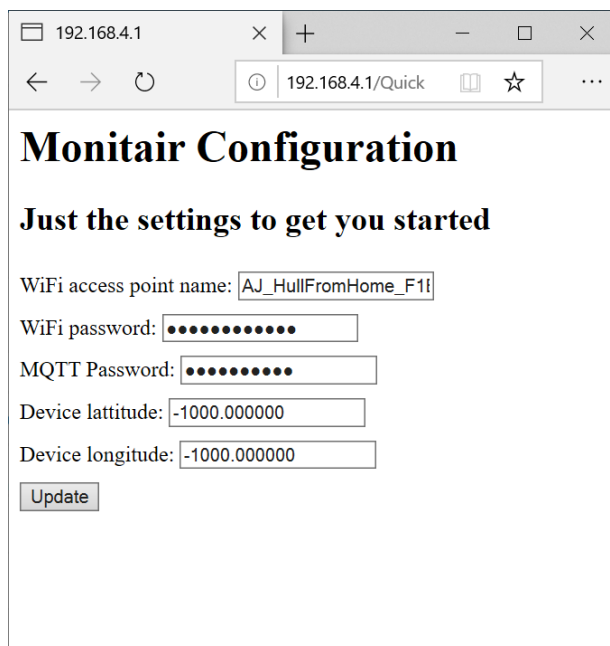
`http://192.168.4.1`



The home page for the device provides links to several configuration pages for different settings on the device.

For simple configuration select the link "Just the settings to get you started".

You can use this to set just the options that you need to connect to the Connected Humber MQTT service and start uploading data.

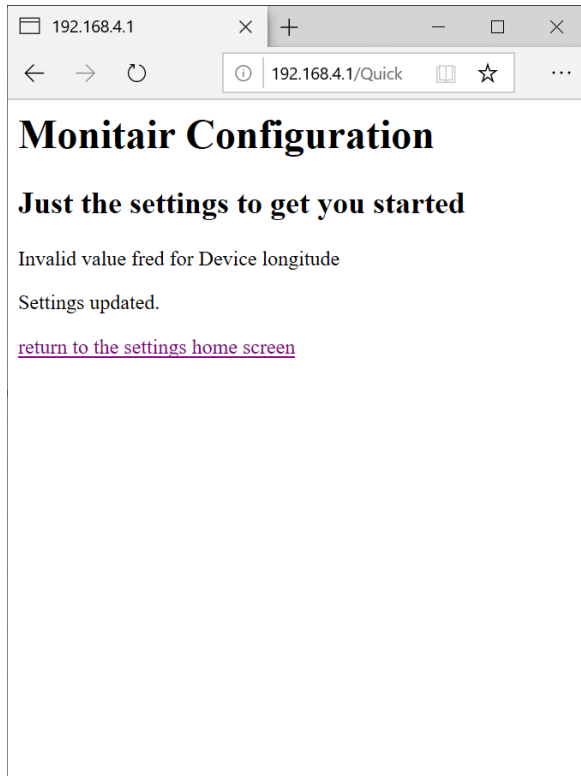


The simple settings page lets you configure the access point and password of the network the Monitair device will be connected to.

It also lets you set the password for the Connected Humber MQTT broker. This is the default broker for our devices.

Finally you can set the latitude and longitude of the device. This can be used by our servers to place readings from your device on the map for the area.

When you have entered these values press the Update button to store the values in the device.



If you make a mistake entering a value the Monitair device will display an error message when Update is pressed identifying the invalid values.

Note that all the other values that you have entered will have been stored. There is no need to enter these again.

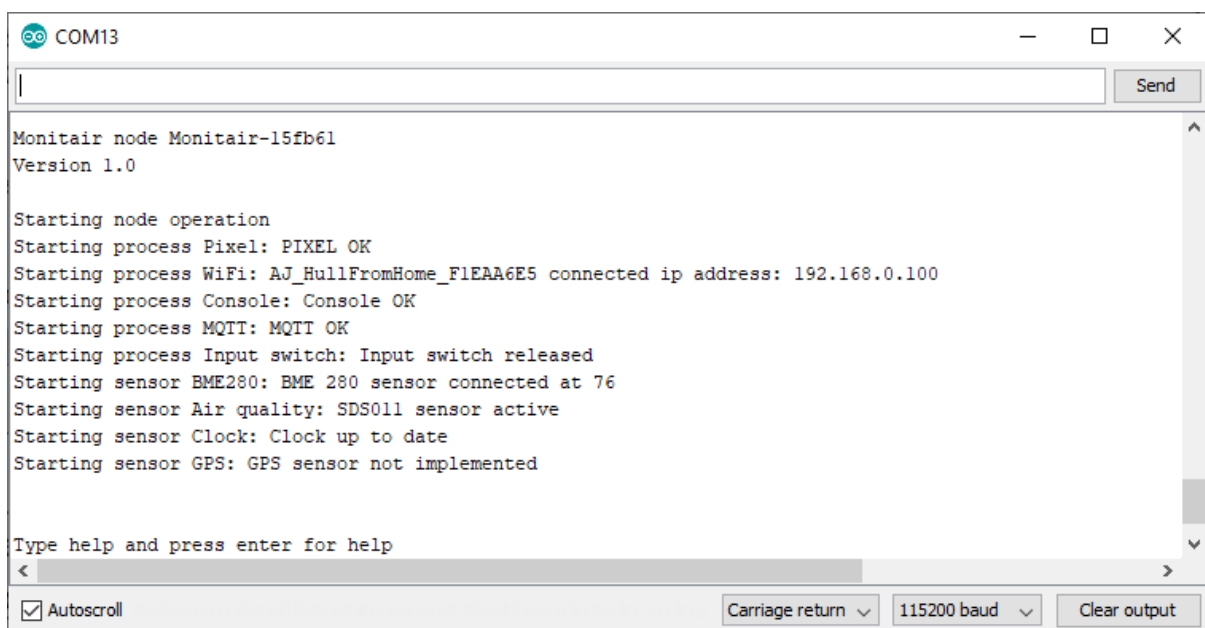
You can press the link to return to the settings home screen if you want to enter new settings.

When you have finished entering values you need to turn the Monitor node off and then on again to return to normal operation.

Configuration and management via the serial console

A Monitair device can be configured using a serial connection to the Wemos processor. Connect a serial terminal to the device and set the baud rate to 115200. The screenshots were created using the Serial Monitor which is part of the Arduino SDK but you can use any terminal that you like.

When you reset the device the following startup message is displayed:

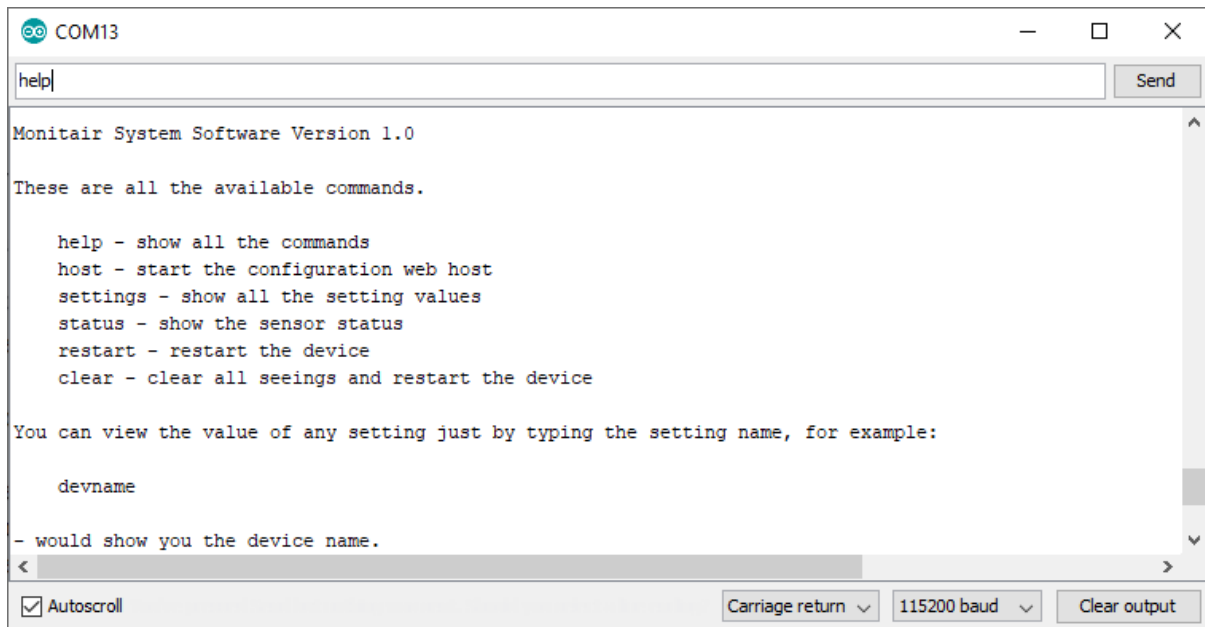


The device is managed as several internal processors and sensors. When the device is powered up it will display each of these as they are started. This is a useful way to confirm that all the connected peripherals are working correctly. Note that the GPS software process has not yet been implemented.

Once the processors and sensors have been started the sensor will then begin running and respond to commands that are entered via the console. At this point the sensor is active and will be uploading values to the MQTT broker.

Serial commands

You can send commands to the device using the console. To find out what commands are available you can use the command `help`.



Note that you need to enter the command at the top of the screen and press the SEND button. Note also that you need to make sure that commands you enter are terminated by a carriage return or a newline when they are sent to the device. You set this option using the combo-box at the bottom of the terminal

host command

The host command starts an internal web server that can be used to configure large numbers of device settings easily. The interface for the server is the same as that provided for Access Point configuration described above. It can be useful to configure settings on an active device. To connect to the web server hosted by the device you use the IP address that is displayed when the device starts.

Settings command

The settings command allows you to view all the internal settings and their names. The settings are displayed in groups. An example of the complete setting output is given at the end of this document. Each setting item is displayed in the following form:

```
Device name [devname]: Monitair-15fb61
```

The text at the start of the message gives a long text description of the setting. The text enclosed in square brackets [] is the *internal name* of that setting. In the example above the setting is called `devname`.

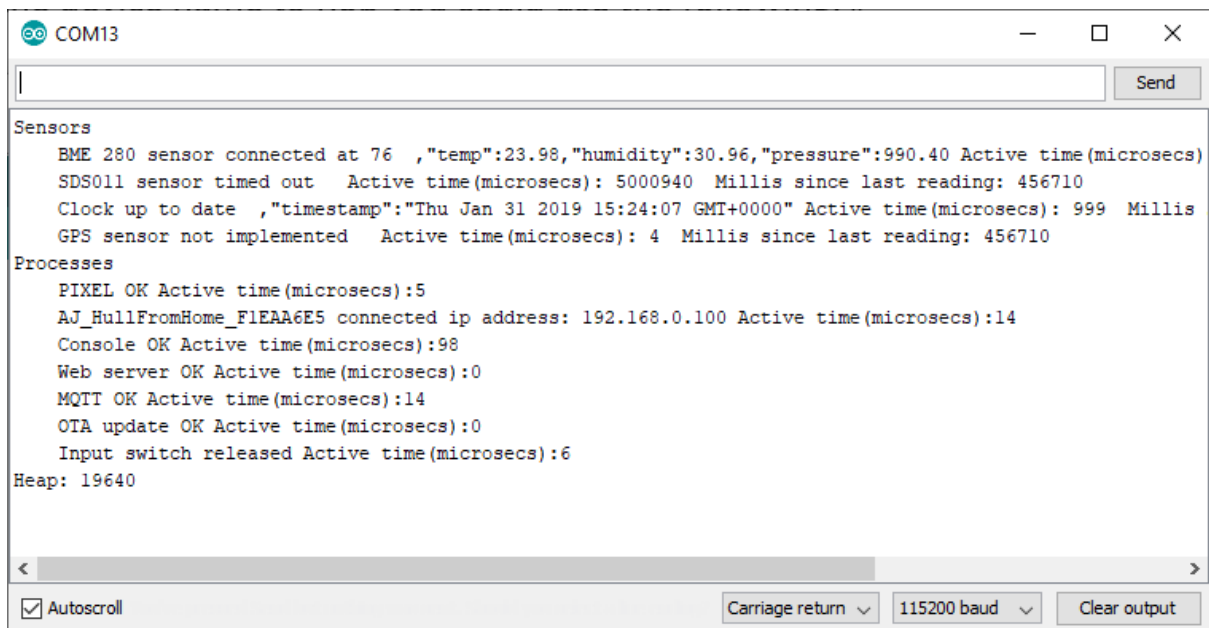
You can view the value of a single setting by just entering the internal name of that setting as a command. You can also set the value of a setting by adding an equals character followed by the new value. For example, to set the device name to Rob you could use the following:

```
devname=Rob
```

If the text that is entered is invalid (for example letters where a number is expected) an error is displayed and the setting is updated.

status command

The status command shows the status for each of the internal processors and sensors:



```
COM13
|
Send

Sensors
  BME 280 sensor connected at 76 , "temp":23.98, "humidity":30.96, "pressure":990.40 Active time(microsecs)
  SDS011 sensor timed out Active time(microsecs): 5000940 Millis since last reading: 456710
  Clock up to date , "timestamp": "Thu Jan 31 2019 15:24:07 GMT+0000" Active time(microsecs): 999 Millis
  GPS sensor not implemented Active time(microsecs): 4 Millis since last reading: 456710

Processes
  PIXEL OK Active time(microsecs):5
  AJ_HullFromHome_F1EAA6E5 connected ip address: 192.168.0.100 Active time(microsecs):14
  Console OK Active time(microsecs):98
  Web server OK Active time(microsecs):0
  MQTT OK Active time(microsecs):14
  OTA update OK Active time(microsecs):0
  Input switch released Active time(microsecs):6
Heap: 19640

Autoscroll Carriage return 115200 baud Clear output
```

For each sensor the sensor status is displayed along with a json fragment that makes up the latest reading sent to the server. The output also shows the number of microseconds that sensor was active in the last timeslice and finally the number of milliseconds since the last reading was taken from that device.

For each process the status is displayed, followed by the number of microseconds the process took in the last timeslice.

Finally the size of the heap is displayed.

restart command

The restart command restarts the device. It is equivalent to pressing the reset button, or turning the device off and on again.

clear command

The clear command returns a device to factory settings and then restarts it. **Note that there is no confirmation for this command, it happens immediately. Be sure that this is what you want to do before you do it.**

Settings

These are all the settings that can be managed in the device.

Wifi

Device name [devname]: Monitair-15fb61
WiFiSSID1 [wifissid1]: AJ_HullFromHome_F1EAA6E5
WiFiPassword1 [wifipwd1]: *****
WiFiSSID2 [wifissid2]:
WiFiPassword2 [wifipwd2]: *****
WiFiSSID3 [wifissid3]:
WiFiPassword3 [wifipwd3]: *****
WiFiSSID4 [wifissid4]:
WiFiPassword4 [wifipwd4]: *****
WiFiSSID5 [wifissid5]:
WiFiPassword5 [wifipwd5]: *****
WiFi On [wifion]: on

MQTT

MQTT Host [mqtttthost]: mqtt.connectedhumber.org
MQTT port number [mqtttport]: 1883
MQTT secure sockets (on or off) [mqtttsecure]: off
MQTT Active [mqtttactive]: on
MQTT UserName [mqtttuser]: connectedhumber
MQTT Password [mqtttpwd]: *****
MQTT Publish topic [mqtttpub]: test/data
MQTT Subscribe topic [mqtttsub]: airquality/command/Monitair-15fb61
MQTT Reporting topic [mqtttreport]: airquality/report/Monitair-15fb61
MQTT Seconds per update [mqttsecsperupdate]: 5
MQTT Seconds per retry [mqttsecsperretry]: 10

Pixel

Pixel red (0-255) [pixelred]: 128
Pixel green (0-255) [pixelgreen]: 128
Pixel blue (0-255) [pixelblue]: 128
AirQ Low Limit [airqlowlimit]: 1
AirQ Low Warning Limit [airqlowwarnlimit]: 2
AirQ Mid Warning Limit [airqmidwarnlimit]: 3
AirQ High Warning Limit [airqhighwarnlimit]: 4
AirQ High Warning Limit [airqhighalertlimit]: 9

Hardware

AirQ Sensor type (0 = not fitted 1=SDS011, 2=XPH01) [airqsensortype]: 1
AirQ Seconds for sensor warmup [airqsensorwarmup]: 30
AirQ RX Pin [airqrxpinno]: 12
BME 280 fitted [bme280fitted]: yes
Power Control fitted [powercontrolfitted]: yes
Power Control Pin [powercontrolpin]: 2
Control Input Pin [controlinputpin]: 14
Control Input Active Low [controlinputlow]: yes
GPS fitted [gpsfitted]: no
GPS RX Pin [gpsrxpin]: 13
Number of pixels (0 for pixels not fitted) [noofpixels]: 12
Pixel Control Pin [pixelcontrolpin]: 15

Location

Fixed location [fixedlocation]: yes
Device lattitude [lattitude]: -1000.00
Device longitude [longitude]: -1000.00