



# **box293\_check\_sensehat**

## **A Nagios Plugin To Monitor Raspberry Pi Sense HAT Temperature & Humidity**

### **Table of Contents**

Overview .....	2
Setup Instructions.....	3
Download and extract files .....	3
Install Modules .....	3
Create Nagios User And Directories .....	4
Transfer the box293_check_sensehat plugin to the Pi .....	4
Create Sudoers Entry.....	5
Test Plugin - Nagios Mode .....	5
Test Plugin - Display Mode.....	6
Configure Cron Job .....	6
Nagios Steps.....	7

Finish Configuring The Pi .....	7
Configure Nagios server.....	7
Create Certificates .....	7
Plugin Test From Nagios Host .....	9
Thresholds .....	9
Create Monitoring Objects In Nagios.....	9
Check Nagios .....	10
Plugin Syntax .....	11
Plugin Arguments.....	11
Modes.....	15
How Did You.....	15
Subscribe For Update Notifications .....	<b>Error! Bookmark not defined.</b>
Subscribe .....	<b>Error! Bookmark not defined.</b>
Un-subscribe.....	<b>Error! Bookmark not defined.</b>
My Nagios Projects .....	<b>Error! Bookmark not defined.</b>
My Nagios Website .....	<b>Error! Bookmark not defined.</b>
Nagios XI Discount Code .....	<b>Error! Bookmark not defined.</b>
Support .....	16
Feedback and Contributions.....	16
About Me .....	<b>Error! Bookmark not defined.</b>

## Overview

This is the manual for the Nagios Plugin called "box293\_check\_sensehat". The plugin is designed to be running on a Raspberry Pi device that has the Sense HAT module connected to it. The Sense HAT module provides many different functions, this plugin utilises the temperature and humidity sensors along with the 8x8 LED Matrix.

The plugin serves two purposes:

1. Monitoring the temperature and humidity to allow Nagios to alert when thresholds have been exceeded
2. Scroll the temperature and humidity on the 8x8 LED Matrix

## Target Audience

Considering the purpose of a Raspberry Pi is to provide kids with the ability to experiment with computing, the instructions in this guide should allow any user of any age or skill level to implement the plugin and make it work.

Nagios is not the most straight forward monitoring system for a beginner to setup. However steps on how to do this are found in the following guide:

<https://sites.google.com/a/box293.com/nagios/guides/installing-nagios/4-1-x/centos-6-7>

If you follow that guide to setup Nagios in a virtual machine, or on a spare computer, the rest of this guide should work 100%.

NOTE: If you don't care about Nagios, and simply want to use this plugin to display the temperature and humidity on the 8x8 LED Matrix, this guide can still be followed, just skip the “Nagios server” steps and it will work as expected.

## About The Instructions

- The steps listed below should cater to all types of end users, the goal is to make it as easy as possible
- When an ssh session is mentioned, this just means you are opening a secure shell, basically a remote terminal window on the Pi. This is handy as it allows you to do the steps without being directly connected to the Pi. However all steps can also be performed in a terminal session on the Pi directly.
  - Windows Users:
    - You can ssh to the Pi using [Putty](#)
    - You can transfer the files to the Pi using [WinSCP](#)

### Linux Users

- You can ssh to the Pi using the terminal program that comes with your Linux Distribution
  - You can transfer the files to the Pi using using the command **scp** (*part of openssh-clients*)
- There are some commands which are presented as follows:
  - Type **mkdir .ssh** and press Enter
  - The **bold text** is what you need to type into the window
  - **Copying** and **Pasting** the text can save spelling mistakes and it makes the process pretty simple (*this is why I like doing things in an SSH session*)
  - Some commands start with **sudo**, this means you are elevating your privileges to run the command. The first time you use **sudo** you will be prompted for the password for the user you are currently logged in as, this is simply a security measure. The next command that uses sudo won't prompt you to type the password again, it rememebers you just typed it (*for a limited time window*).

## Setup Instructions

These are the steps required to get the box293\_check\_sensehat plugin up and running.

These steps were created using:

- Raspberry Pi Model B
- Raspberry Pi Sense HAT module
- NOOBS 1.9.0
- Raspbian GNU/Linux 8 (jessie)

Raspberry Pi IP Address: 192.168.100.10

Nagios Server IP Address: 192.168.100.20

Your Raspberry Pi will need to be able to access the Internet to download and install Python libraries.

## Download and extract files

Download the box293\_check\_sensehat.zip file to a location on your PC and extract the files.

## Install Modules

Power on the Pi.

Initially you'll notice the 8x8 LED Matrix is on with a pretty gradient.

- Establish an **SSH session** to the **Pi** (192.168.100.10)
  - login as the user: **pi**
  - Type **sudo apt-get update** and press Enter
  - *Wait while the command runs*
  - Type **sudo apt-get install sense-hat libjpeg-dev** and press Enter
  - *Wait while the command runs*
  - Type **sudo pip install pillow** and press Enter
  - *Wait while the command runs*
  - Type **sudo reboot** and press Enter
  - *Wait while the Pi reboots*

After the reboot the 8x8 LED Matrix will no longer be lit up with the pretty gradient, this is expected.

## **Create Nagios User And Directories**

We are going to create a user called nagios that the plugin will execute as.

- Establish an **SSH session** to the **Pi** (192.168.100.10)
  - login as the user: **pi**
  - Type **sudo adduser --home /home/nagios --gecos "" nagios** and press Enter
  - You will be prompted for a password (*for example: A Str0ng P@ssw0rd*)
  - Type **sudo mkdir -p /usr/local/nagios/libexec** and press Enter
  - Type **sudo chown nagios:nagios /usr/local/nagios/libexec** and press Enter
  - Type **sudo chmod 770 /usr/local/nagios/libexec** and press Enter
  - We'll come back to this window after transferring the files

## **Transfer the box293\_check\_sensehat plugin to the Pi**

- Windows Users:
  - Establish a **WinSCP session** to the **Pi** (*for example 192.168.100.10*)
    - login as: **nagios**
    - Password: The one you previously provided (*for example: A Str0ng P@ssw0rd*)
    - In the **left pane** of WinSCP navigate to the directory you **extracted** box293\_check\_sensehat.zip
    - In the **right pane** of WinSCP:
      - Click the **icon** that looks like a **folder with a /** (this changes to the / directory on the Pi)
      - Double click **usr**
      - Double click **local**
      - Double click **nagios**
      - Double click **libexec**
    - In the **left pane** drag **box293\_check\_sensehat.py** to the **right pane**
    - *This copies the box293\_check\_sensehat.pl plugin to the Pi*
    - In the **left pane** drag **box293\_check\_sensehat\_cron.sh** to the **right pane**
    - *This copies the box293\_check\_sensehat\_cron.sh script to the Pi*

- **Close** the WinSCP session
- Linux Users:
  - The following command, executed from your workstation could do this if you were in the directory that the files are located:
  - Type **scp box\* nagios@192.168.100.10:/usr/local/nagios/libexec/** and press Enter
  - Type the password you previously provided (*for example: A Str0ng P@ssw0rd*)
  - This copies both files to the Pi

## Create Sudoers Entry

Now we need to add an entry to the sudoers file as the plugin requires elevated privileges to run. This means the plugin can run without having to prompt for a password.

- Return to your **SSH session** to the **Pi**
- Type **sudo visudo** and press Enter
  - This puts you in an editor, most likely *nano*
  - Add this line to the bottom of the file:
    - **nagios ALL = NOPASSWD:/usr/local/nagios/libexec/box293\_check\_sensehat.py \*** ○
- Save and exit (**Ctrl + X** and then **y** and then **Enter**)
- We'll continue in this window in the next step

## Test Plugin - Nagios Mode

At this point we can test the plugin.

- Return to your **SSH session** to the **Pi**
- First we need to become the nagios user
- Type **sudo su nagios** and press Enter
  - You are now the nagios user
  - Type **cd ~** and press Enter
  - You are now the nagios user home directory */home/nagios/*
  - Type **sudo /usr/local/nagios/libexec/box293\_check\_sensehat.py** and press Enter ○

Which should output something like:

- OK: {Temperature 31.1 Degrees Celsius} {Humidity 44.6 %} | 'Temperature Degrees Celsius'=31.1 'Humidity %'=44.6%
- If you're from a part of the world that uses Fahrenheit, type this instead:
- Type **sudo /usr/local/nagios/libexec/box293\_check\_sensehat.py -d f** and press Enter ○

Which should output something like:

- OK: {Temperature 88.5 Degrees Fahrenheit} {Humidity 44.6 %} | 'Temperature Degrees Fahrenheit'=88.5 'Humidity %'=44.6%

Great so we know it works. FYI you'll also notice that every time the plugin runs, it creates/updates the file `box293_check_sensehat.txt`. This file is a record of the last measurements taken, and if warning or critical thresholds were exceeded (see Modes chapter later in the manual for more information about this).

This file is used by the display mode, which is covered next. This file is created in the directory you were in when you executed the plugin. If you followed the steps above exactly, the file will be created in **/home/nagios/**.

## Test Plugin - Display Mode

At this point we can test the plugin.

- Return to your **SSH session** to the **Pi**, still as the nagios user
    - Type **sudo /usr/local/nagios/libexec/box293\_check\_sensehat.py --mode display** and press Enter
- You should see something like the following scroll across the 8x8 LED matrix display:



- This message will scroll continually, and keep up to date (see Modes chapter later in the manual for more information about this)
- You'll notice that you haven't been returned to the command prompt. This is because the display mode runs in an infinite loop. Press **CTRL + C** to stop and return to the command prompt
  - The screen will stop scrolling and may leave whatever was last on the screen lit up. This is OK, it will be cleared the next time it runs.
- If you're from a part of the world that uses Fahrenheit, type this instead:
- Type **sudo /usr/local/nagios/libexec/box293\_check\_sensehat.py --mode display -d f** and press Enter
  - Press **CTRL + C** to stop and return to the command prompt

## Configure Cron Job

You noticed in the previous step that when you run the plugin in display mode, it scrolled the message continually. This is great, but we now need to create a cron job that will run the display mode in the background, and continue to make sure it is running. This will also ensure that when you reboot the Pi, the message will resume scrolling after the reboot.

- Return to your **SSH session** to the **Pi**, still as the nagios user
  - Type **crontab -u nagios -e** and press Enter
  - You will be prompted to choose an editor, just press **enter** to accept the defaults (nano)
    - Add this line to the bottom of the file:
    - **\* \* \* \* \* /usr/local/nagios/libexec/box293\_check\_sensehat\_cron.sh > /dev/null 2>&1**
- *The plugin defaults to Celsius, so if you want the temperature to be Fahrenheit, use this line instead:*
  - **\* \* \* \* \* /usr/local/nagios/libexec/box293\_check\_sensehat\_cron.sh '-d f' > /dev/null 2>&1**
  - Save and exit (**Ctrl + X** and then **y** and then **Enter**)

Within one minute the display should start to scroll with the temperature and humidity. This cronjob will ensure the display is always scrolling the text. Even after a reboot it will start scrolling within one minute.

Read the Modes chapter that goes into more detail as to how the nagios mode updates into the display mode.

We'll continue in this SSH window in the next step.

## Nagios Steps

All these remaining steps show you how to configure Nagios to monitor the Pi. If you don't want to do anything with Nagios, you don't need to follow these steps.

### Finish Configuring The Pi

We have one last step to perform on the Pi.

- Return to your **SSH session** to the **Pi**, still as the nagios user
  - Type **cd ~** and press Enter
  - Type **mkdir .ssh** and press Enter
  - Type **chmod 0700 .ssh/** and press Enter
  - Type **exit** and press Enter
- You are now back as the pi user.

The next steps are performed on the Nagios server.

### Configure Nagios server

**NOTE** The following example is performed a Nagios XI server which is running on CentOS.

- If you followed this guide:
  - <https://sites.google.com/a/box293.com/nagios/guides/installing-nagios/4-1-x/centos-6-7>
  - Everything should be straight forward.
- If you are using a different OS then some of these commands may be a little different. The main requirements here are:
  - openssh-clients (required for SSH)
  - Nagios Plugins 1.5 or later (this contains the check\_by\_ssh plugin which is required)
  - If you know you already have these components installed jump down to **Create Certificates** section
- Establish an **SSH session** (using Putty) to your **Nagios Server**
  - login as: **root**
  - Password: *You should know your root password*
  - You will now be logged in and presented with: [root@localhost ~]#
- **Installing openssh-clients**
  - Type **yum -y install openssh-clients** and press Enter
  - *Wait while the files are downloaded and installed*
  - Leave this SSH session open, you will use it in the next step

### Create Certificates

- **Create the certificates to use with Pi**
  - *You are creating a certificate that allows the Nagios server to establish an SSH session with the Pi without using credentials*

- Type **su nagios** and press Enter (*this means the following steps will be performed as the 'nagios' user, as this is what account is used when the Nagios Monitoring Engine Executes the box293\_check\_sensehat pluin*)
  - You will now be presented with: [nagios@localhost]\$
  - Type **cd ~** and press Enter (*puts you in the home directory of the nagios user*)
  - Type **ls -la ~/.ssh/id\_dsa.pub** and press Enter
    - If you get the message:
      - "No such file or directory", continue with the following steps
    - Otherwise skip to the **Transfer certificates to the Pi** steps
  - Type **ssh-keygen -t dsa** and press Enter
    - Enter file in which to save the key (/home/nagios/.ssh/id\_dsa):
      - You will use the default location so **press Enter**
    - Enter passphrase (empty for no passphrase):
      - You will use an empty passphrase so **press Enter**
    - Enter same passphrase again:
      - You will use an empty passphrase so **press Enter**
    - A randomart image is displayed, the certificate has been created
  - Leave this SSH session open, you will use it in the next step
- **Transfer certificates to the Pi**
- The next command will transfer the certificate file to the **Pi** (192.168.100.10)
- Type **cat ~/.ssh/id\_dsa.pub | ssh nagios@192.168.100.10 'umask 077; cat >>~/.ssh/authorized\_keys'** and press Enter (*that command is all on one line*)
  - An authenticity message is displayed
  - Are you sure you want to continue connecting (yes/no)?
    - Type **yes** and press Enter
  - You are prompted for the **nagios** password on the Pi (*for example: A Str0ng P@ssw0rd*)
    - Type *the password* and press Enter
  - The certificate file is transferred
- Leave this SSH session open, you will use it in the next step



## Plugin Test From Nagios Host

- This will test the plugin in exactly the same way the Nagios monitoring engine would execute a check and confirm everything is configured correctly.
- Continue on using your SSH session to your Nagios host (*which is currently the nagios user*)
  - Type `/usr/local/nagios/libexec/check_by_ssh -E 1 -l nagios -H 192.168.100.10 -C "sudo /usr/local/nagios/libexec/box293_check_sensehat.py"` and press Enter
  - If the command works as expected, you should get a response like:
    - OK: {Temperature 32.0 Degrees Celsius} {Humidity 35.2 %}|'Temperature Degrees Celsius'=32.0 'Humidity %'=35.2%
  - This means you have configured the Nagios Server to execute a check on the Pi host using `box293_check_sensehat.py` without being prompted for credentials
- Testing complete
  - Type `exit` and press Enter
- You are now back to your SSH session on your Nagios host as the user root
- Leave this SSH session open, you will use it in the next step

That completes all the configuration steps required to get the `box293_check_sensehat` plugin working. The next steps will be to configure Nagios to use the plugin and create your hosts and services.

## Thresholds

Before continuing it's worthwhile talking about thresholds.

Right now, all of our testing has not used any thresholds, so this plugin will always be in an OK state in Nagios.

What we really want is that if the temperature gets too high, make the plugin go into a WARNING or CRITICAL state.

This can be done by adding the arguments like so:

- `sudo /usr/local/nagios/libexec/box293_check_sensehat.py --warning_temperature 40 --critical_temperature 50`

That means if the temperature goes over 40, return a warning status. If the temperature goes over 50, return a critical status.

There are also thresholds for humidity, these are explained further on in the manual.

## Create Monitoring Objects In Nagios

This will create the monitoring objects in a config file that Nagios will use when it starts. This file has been included with the plugin, so you can copy it to the Nagios server which should make things easier (you'll still need to edit it so it has the correct IP address in the host object and adjust your warning and threshold values.

- Continue on using your SSH session to your Nagios host (*which is currently the root user*)
- Type `vi /usr/local/nagios/etc/objects/pi.cfg` and press Enter
- You are now in the vi text editor
  - Press `i` on the keyboard to enter *insert* mode
  - Type/paste the following into the window:

```

define command { command_name
    box293_check_sensehat
    command_line    $USER1$/check_by_ssh -E 1 -l nagios -H $HOSTADDRESS$
-C "sudo /usr/local/nagios/libexec/box293_check_sensehat.py $ARG1$"
}

define host {
    use                linux-server
    host_name          Pi
    address             192.168.100.10
}

define service {
    use local-service host_name Pi
    service_description Environment Sensors
    check_command       box293_check_sensehat! --warning_temperature
40 --critical_temperature 50
    max_check_attempts 3 check_interval
        1 retry_interval    1
}

```

- The only setting you will need to change is:
  - address 192.168.100.10
  - This should be the IP Address of your Pi
- Press **Escape** on the keyboard to exit *insert* mode
- Type **:wq** and press Enter
- Now we need to tell Nagios to use this config file
- Type **echo 'cfg\_file=/usr/local/nagios/etc/objects/pi.cfg' >> /usr/local/nagios/etc/nagios.cfg** and press Enter
- Finally, restart Nagios
- Type **service nagios restart** and press Enter

If you wanted Fahrenheit, simply change the check command line to include **-d f** like so:

```

check_command box293_check_sensehat! --warning_temperature 40 --
critical_temperature 50 -d f

```

## Check Nagios

Now that Nagios has been configured to monitor the Pi, lets have a look at this in the Nagios Web Interface.

- Open a web browser to <http://192.168.100.20/nagios>
- Login as **nagiosadmin**
- In the left pane, in the Quick Search window type **pi** and press Enter
- You should see something like:

General

Home  
Documentation

Current Status

Tactical Overview  
Map (Legacy)  
Hosts  
Services  
Host Groups  
Summary  
Grid

Service Groups  
Summary  
Grid

Problems  
Services (Unhandled)  
Hosts (Unhandled)  
Network Outages

Quick Search:  
pi

Current Network Status

Last Updated: Sun May 8 19:14:10 AEST 2016  
Updated every 90 seconds  
Nagios® Core™ 4.1.2-Pre1 - www.nagios.org  
Logged in as nagiosadmin

View History For This Host  
View Notifications For This Host  
View Service Status Detail For All Hosts

Host Status Totals

Up	Down	Unreachable	Pending
1	0	0	0

All Problems All Types

0	1
---	---

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
1	0	0	0	0

All Problems All Types

0	1
---	---

## Service Status Details For Host 'Pi'

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
Pi	Environment Sensors	OK	05-08-2016 19:13:35	0d 0h 2m 35s	1/3	OK: {Temperature 30.6 Degrees Celsius} {Humidity 48.4 %}

Results 1 - 1 of 1 Matching Services

- Click the word service Environment Sensors • You should see something like:

Service State Information

Current Status: OK (for 0d 0h 6m 14s)

Status Information: OK: {Temperature 31.4 Degrees Celsius} {Humidity 47.2 %}

Performance Data: 'Temperature Degrees Celsius'=31.4;40.0;50.0 'Humidity %'=47.2%

Current Attempt: 1/3 (HARD state)

Last Check Time: 05-08-2016 19:17:35

Check Type: ACTIVE

Check Latency / Duration: 0.000 / 0.952 seconds

Next Scheduled Check: 05-08-2016 19:18:35

Last State Change: 05-08-2016 19:11:35

Last Notification: N/A (notification 0)

Is This Service Flapping? NO (0.00% state change)

In Scheduled Downtime? NO

Last Update: 05-08-2016 19:17:46 ( 0d 0h 0m 3s ago)

Active Checks: ENABLED

Passive Checks: ENABLED

Obsessing: ENABLED

Notifications: ENABLED

Event Handler: ENABLED

Flap Detection: ENABLED

Service Commands

✗ Disable active checks of this service

🕒 Re-schedule the next check of this service

❓ Submit passive check result for this service

✗ Stop accepting passive checks for this service

✗ Stop obsessing over this service

✗ Disable notifications for this service

📧 Send custom service notification

🕒 Schedule downtime for this service

✗ Disable event handler for this service

✗ Disable flap detection for this service

Congratulations, you are now monitoring your Pi in Nagios!

You will of course want to look into setting up a contact so a notification is sent when the warning or critical thresholds are triggered, but that is not covered in this guide.

## Plugin Syntax

box293\_check\_sensehat.pl <arguments as required>

## Plugin Arguments

- m
- mode
  - Define the mode
    - nagios

- Runs as a Nagios plugin
- display
  - Outputs to Sense HAT 8x8 LED Matrix Display
- Default: nagios • Example:
  - -m display
  - --mode display

-d

--degrees\_unit

- Define the degree unit used for temperature
  - c = Celsius
  - f = Fahrenheit
- Default: c
- If you want display mode to show Fahrenheit you'll need to define this in the cron job command
- Example:
  - -d c
  - --degrees\_unit f

-w

--warning\_temperature

- Allows you to provide a warning threshold for temperature
- No default defined
- Does not require a critical threshold to also be defined
- Example:
  - -w 30
  - --warning\_temperature 30

-c

--critical\_temperature

- Allows you to provide a critical threshold for temperature
- No default defined
- Does not require a warning threshold to also be defined
- Example:
  - -c 30
  - --critical\_temperature 40

-f

--warning\_humidity

- Allows you to provide a warning threshold for humidity
- Humidity is a % value (don't provide the % sign in the threshold)
- No default defined
- Does not require a warning threshold to also be defined
- Example:
  - -s 30
  - --warning\_humidity 30

-s

#### `--critical_humidity`

- Allows you to provide a critical threshold for humidity
- Humidity is a % value (don't provide the % sign in the threshold)
- No default defined
- Does not require a warning threshold to also be defined
- Example:
  - `-s 30`
  - `--critical_temperature 30`

#### `-p`

##### `--pickle_dictionary_location`

- The location to store the pickle dictionary file for display mode
- (nagios mode) will save to the file, (display mode) will read the file
- Default location is the directory you are in when the plugin executes
  - Normally `/home/nagios/` when executed by Nagios via SSH
- Example
  - `-p /tmp`
  - `--pickle_dictionary_location /usr/local/nagios/libexec/`

#### `-r`

##### `--rotate_led`

- Define the Sense HAT 8x8 LED Matrix Display rotation when (display) mode will scroll the output  
Default is 1 = 90
- Other options are:
  - 0 = 0
  - 2 = 180
  - 3 = 270
- Example
  - `-r 2`
  - `--rotate_led 0`

#### `-h`

##### `--help`

- Display the help
- To see the help type:
  - `box293_check_sensehat.pl --help | more`

#### `-g`

##### `--gnu_license`

- Display the GNU General Public License
- To see the license type:
  - `box293_check_sensehat.pl --gnu_license | more`

#### `-v`

- Set verbosity level
- Allows you too see what is happening when the plugin executes, handy for troubleshooting
- Example ◦ `-v`

- -vvv

--version

- Reports the plugin version

## Modes

When you run the plugin without any arguments, it runs in nagios mode. Every time the plugin runs in the nagios mode, it creates/updates the file `box293_check_sensehat.txt`. This file is a record of the last measurements taken, and if warning or critical thresholds were exceeded.

When you run the plugin in display mode the following things happen:

- Checks for the existence of the file `box293_check_sensehat.txt`
- If it exists, and it is less than 2 minutes old it will:
  - Use the values in this file to display the output
  - It will also report if the current state of the values have exceeded the user defined warning or critical thresholds
  - If the thresholds have been exceeded, the colour of the LED's will reflect that state (or green if OK)
  - The colours are the same as in the Nagios interface (I used a screen capture program with a colour picker to determine the RGB values)
    - If the file doesn't exist, or is older than 2 minutes it will:
- Obtain the values from the sensors directly
- Everything will be in an OK state and the colour of the LED's will be green
  - It runs in an infinite loop
- This loop ensures that it is regularly checking the file to get the most recent results
- If you want to kill this infinite loop:
  - If running the plugin in display mode manually:
    - Press **CTRL + C** to kill it
  - If running the plugin in display mode via the cron job:
    - Type **ps -ef | grep box293 | grep -v grep** and press Enter
    - You'll get two results, the line with **sudo** in it has the **pid** number you'll use in the next command
    - Type **kill 7665** and press Enter
      - Where 7665 is the pid number you just identified
    - The cron job will then restart display mode at the next minute (it runs every minute to ensure it's running)

The point of using the file was to make sure you could see recent results. It is important that if Nagios crashed and wasn't causing the plugin to be executed, the file wouldn't be updated and you would be seeing incorrect results on the display.

## How Did You ...

First let me start off saying that this is the first time I've coded in Python, so a big thank you goes out to all the information I found on the Internet that helped make this happen.

I bet you have some questions about how some of the stuff is done in the plugin.

## How to change the colour of the LED text

It can be defined in the `show_message` function:

```
sense.show_message("H: {} {}%" . format(humidity_state, humidity_value), text_colour=humidity_colour)
```

```
text_colour=humidity_colour
```

You notice this is a variable. The colour is relative to the state the service is in. It's just an RGB value which could be:

```
text_colour=(255, 0, 0)
```

Something like that.

## How to change the rotation of the LED text

It can be defined in the `set_rotation` function:

```
sense.set_rotation(rotation)
```

You notice this is a variable. This can be one of four angles (0, 90, 180, 270):

```
sense.set_rotation(180)
```

## Support

Please contact us for support via the email address:

- [devteam@nagios.com](mailto:devteam@nagios.com)

## Feedback and Contributions

Any feedback or contributions are welcome as it improves the plugin for others.