Raspberry PI 'How-To' Series

AOSONG AM2315 Temperature Sensor Implementation Guide Update

Written by: Sopwith
Revision 2.0
November 12, 2017
sopwith@ismellsmoke.net

Introduction

It is hard to believe I wrote the original AM2315 “How-To’ document over three years ago. A lot has changed in the Pi world since then. Over the last several weeks, I have been receiving Emails from folks having trouble getting my old instructions to work. There are two reasons for this – hardware and software.

All kidding aside, that is exactly the root cause. The Pi hardware has certainly changed. Most folks are on the Pi-3 and Pi-Zero hardware these days. The Pi-Zero is especially popular for use in weather stations. The *quickwire* libraries I used in the original “How-To” has been a dead project for years. It also requires the use of Python3 which befuddled many hackers since all their other Python code was on version 2.7.

The good news? Implementing an AM2315 sensor on your Pi is now easier than ever. The Raspberry Pi organization has released a terrific Raspian OS (Stretch) that works on every Pi ever built. This is great news for us because you can use an AM2315 on any Pi using a single and simple install process.

This document walks you through the steps to get your AM2315 temperature sensor working.

Step-by-Step

There are twelve (6) steps in the implementation of an AM2315.

1. Get your Pi up and running with *Raspian Stretch*.
2. Install required support software.
3. Wire up your AM2315.
4. Configure i2c and test the AM2315 hardware.
5. Install the *lexruee* i2c library.
6. Test the AM2315 software.

Step-1 - Get your Pi up and running with Raspian Stretch.

The first thing you need to do is get your Pi running. Head out to *raspberrypi.org* and download *Raspian Stretch* or *Stretch Lite.* The former has a Window GUI while the latter is a command-line only OS. Either one works.

Once the download completes, burn the image to an SD card and fire up your Pi. If you need help with this step, there is a ton of resources on the web to help you. When you are up and running and connected to the Internet, be sure to update your Pi with the latest patches.

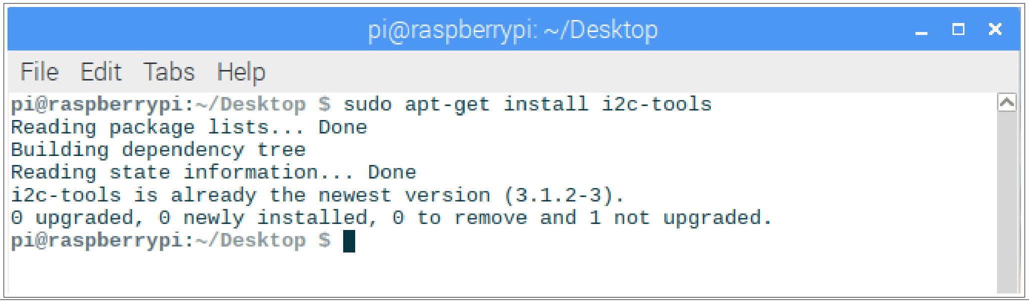
```
$sudo apt-get update
$sudo apt-get upgrade
$sudo reboot
```

Step-2 - Install required support software.

There are three (3) software packages that must be installed for the i2c interface to work correctly. Follow the below steps to get them installed.

Open a command window and type in the command: $*sudo apt-get install i2c-tools.*
You will probably get the output shown in Figure-1 below.

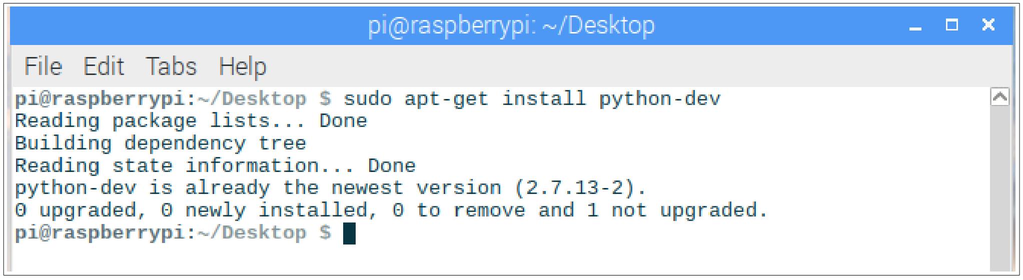
Figure-1



```
pi@raspberrypi:~/Desktop $ sudo apt-get install i2c-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
i2c-tools is already the newest version (3.1.2-3).
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
pi@raspberrypi:~/Desktop $ 
```

Next, run the command: $*sudo apt-get install python-dev*.
You will probably get the output shown in Figure-2 below.

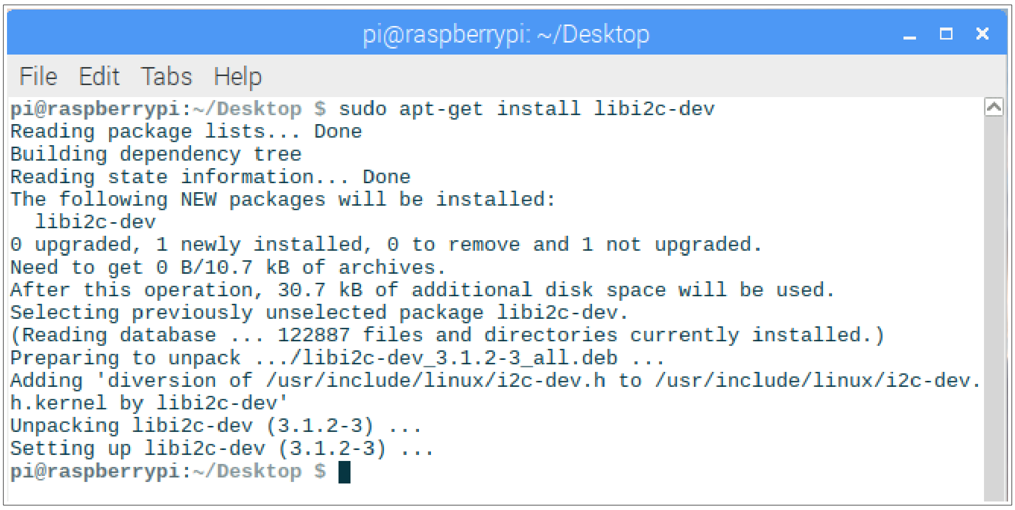
Figure-2



```
pi@raspberrypi:~/Desktop $ sudo apt-get install python-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-dev is already the newest version (2.7.13-2).
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
pi@raspberrypi:~/Desktop $ 
```

Finally, run the command: $*sudo apt-get install libi2c-dev.*
This package is probably not installed on your Pi so the output is going to be different. The package will be installed as shown in Figure-3 below.

Figure-3



```
pi@raspberrypi:~/Desktop $ sudo apt-get install libi2c-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  libi2c-dev
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 0 B/10.7 kB of archives.
After this operation, 30.7 kB of additional disk space will be used.
Selecting previously unselected package libi2c-dev.
(Reading database ... 122887 files and directories currently installed.)
Preparing to unpack .../libi2c-dev_3.1.2-3_all.deb ...
Adding 'diversion of /usr/include/linux/i2c-dev.h to /usr/include/linux/i2c-dev.
h.kernel by libi2c-dev'
Unpacking libi2c-dev (3.1.2-3) ...
Setting up libi2c-dev (3.1.2-3) ...
pi@raspberrypi:~/Desktop $ 
```

You now have all the software you need to get your AM2315 hardware wired up and running.

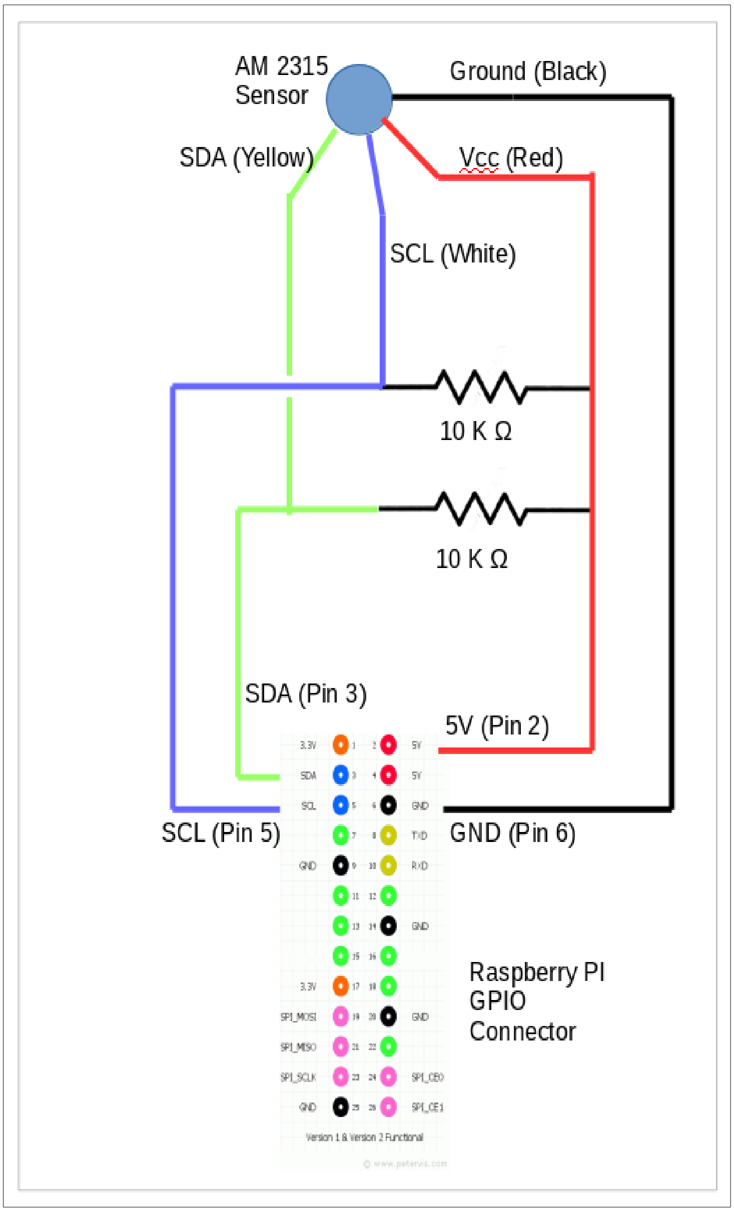
Step-3 - Wire up your AM2315

Shut down your Pi and remove the power lead while you wire up your AM2315 sensor. One of the really nice thing about the Pi design is the consistency in the GPIO pins across models. The GPIO pins used by the AM2315 are the same across all Pi versions.

The AM2315 *must* have two pull-up resistors connected to the device SCL and SDA leads. I use two 10k resistors. If you fail to wire these resistors correctly, your device will not work. The resistors are placed between the AM2315 SDA and SCL leads and the power lead.

Wire your device the same way shown in Figure-4 below.

Figure-4



Source: http://www.adafruit.com/forums/viewtopic.php?f=45&t=48285&start=30

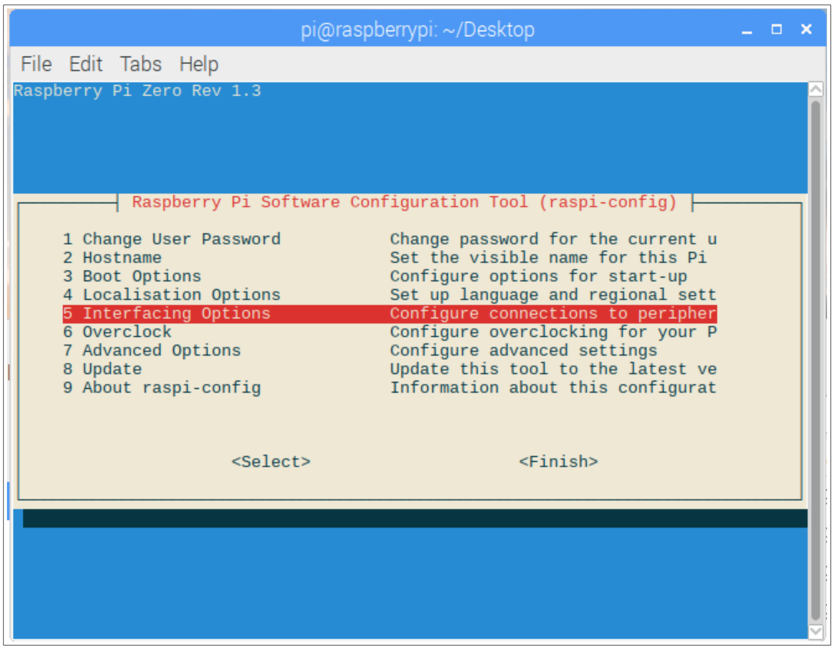
Step-4 - Configure i2c and test the AM2315 hardware

Before you begin this step, take a minute and look over your wiring to ensure each lead is on the right GPIO pin, the correct AM2315 lead is connected to the right GPIO pin, and the resistors are in place where they belong. When everything looks good, fire up your Pi and open a command window.

The first thing to do is enable the i2c interface on your Pi. This is done with the *raspi-config* utility. From the command window run the command: $*sudo raspi-config.*

You will see the window shown in Figure-2.

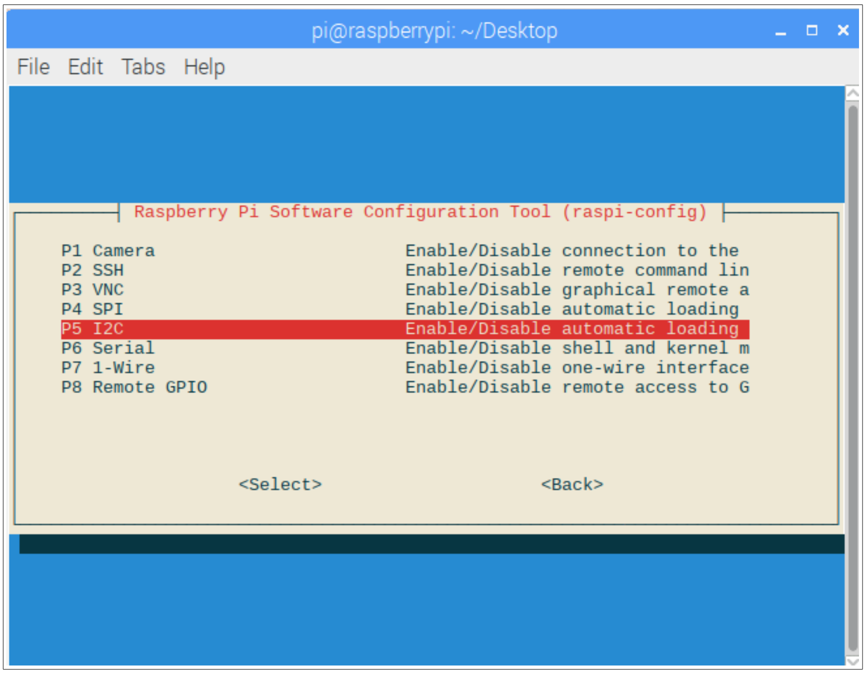
Figure-2.



```
Raspberry Pi Zero Rev 1.3

         ┤ Raspberry Pi Software Configuration Tool (raspi-config) ├

    1 Change User Password           Change password for the current u
    2 Hostname                       Set the visible name for this Pi
    3 Boot Options                   Configure options for start-up
    4 Localisation Options           Set up language and regional sett
    5 Interfacing Options            Configure connections to peripher
    6 Overclock                      Configure overclocking for your P
    7 Advanced Options               Configure advanced settings
    8 Update                         Update this tool to the latest ve
    9 About raspi-config             Information about this configurat

                   <Select>                     <Finish>
```

Press <Enter> and you will see the window shown in Figure-3.

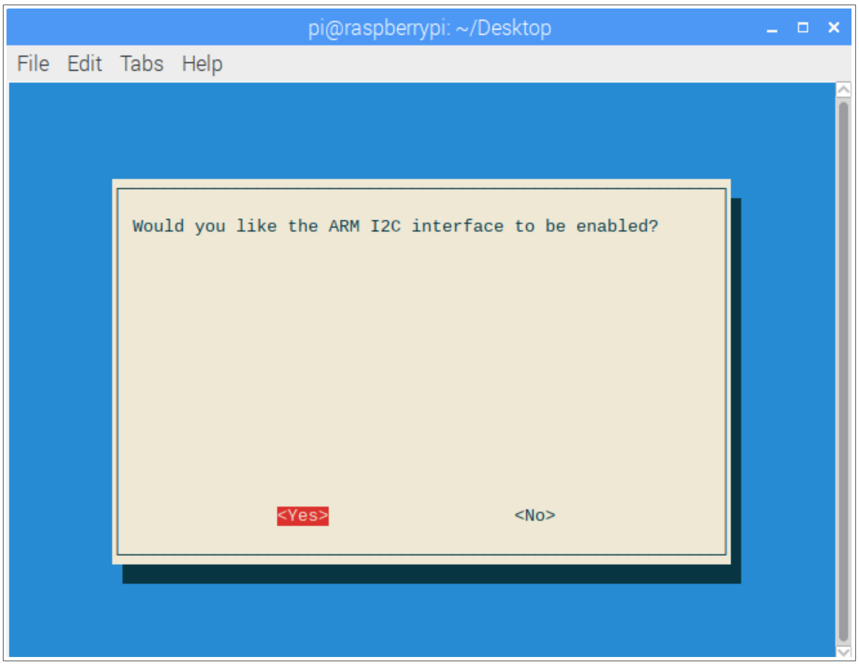
Figure-3



```
          ┤ Raspberry Pi Software Configuration Tool (raspi-config) ├

    P1 Camera                        Enable/Disable connection to the
    P2 SSH                           Enable/Disable remote command lin
    P3 VNC                           Enable/Disable graphical remote a
    P4 SPI                           Enable/Disable automatic loading
    P5 I2C                           Enable/Disable automatic loading
    P6 Serial                        Enable/Disable shell and kernel m
    P7 1-Wire                        Enable/Disable one-wire interface
    P8 Remote GPIO                   Enable/Disable remote access to G

                   <Select>                      <Back>
```

Press <Enter> and you will see the window shown in Figure-4.

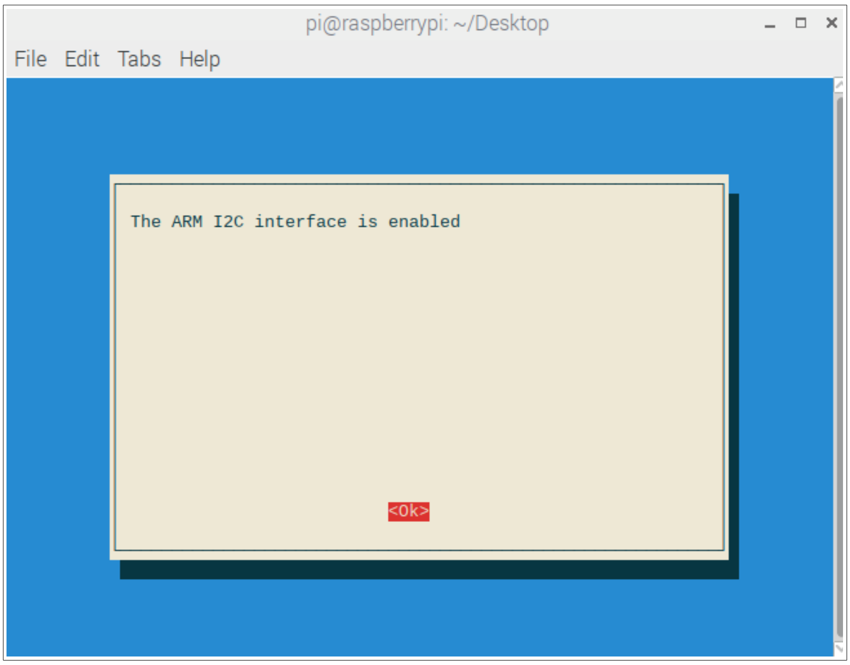
Figure-4



```
Would you like the ARM I2C interface to be enabled?
```

Press <Enter> and you will see the window shown in Figure-5.

Figure-5



```
The ARM I2C interface is enabled
```

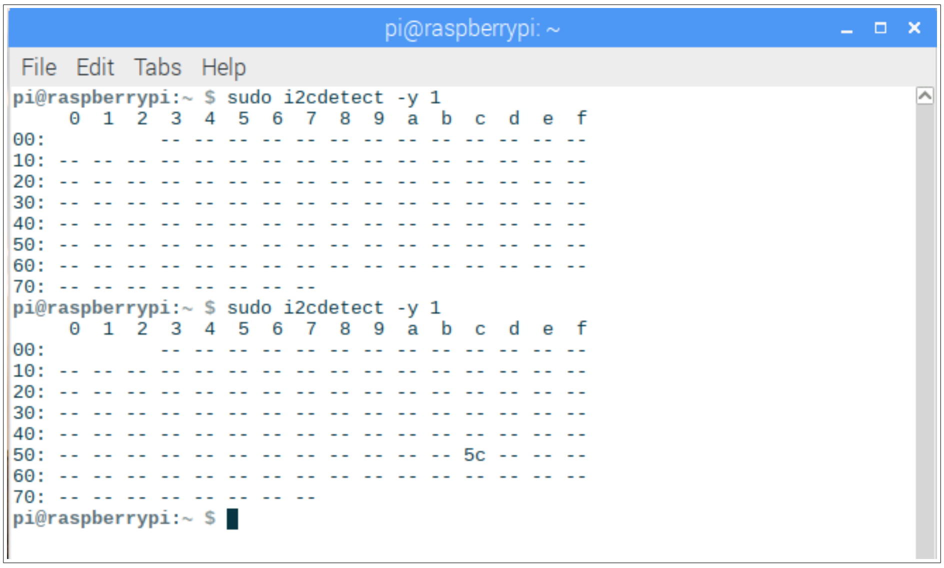
Exit *raspi-config.* Just to be safe, reboot your Pi: $*sudo reboot*.

Once you Pi is back online, let's make sure the AM2315 is wired correctly. Open a command window and type the command: $*sudo i2cdetect -y 1.*

You will need to enter this command two times about 1 second apart. This is due to the "sleep" mode of the device. You have to send it a signal to wake up, and then the command you want it to perform.

The first time you run the command, you will see output shown in the top ½ of Figure-6. No device was found. Running it a second time you should see the device address (0x5c) in the output. This is shown in the bottom ½ of Figure-6.

Figure-6



```
pi@raspberrypi:~ $ sudo i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
pi@raspberrypi:~ $ sudo i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- 5c -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
pi@raspberrypi:~ $ 
```

You will have to experiment with the delay between the two *i2cdetect* commands. It the delay is too long or too short you will not detect the device. On my Pi, a delay of about 1 second works. *Hint*: You can use the up-arrow key to quickly repeat commands entered in a console window without retyping.

If you cannot detect the device (0x5c never appears) then you must go back and review your wiring. The most common problem is the mis-wiring of the resistors. The second most common is using the wrong GPIO pins on the Pi. Do not continue with any more steps until you get this working. Do not blame the device. They are quite hardy and seldom fail unless you smoke them with too much voltage.

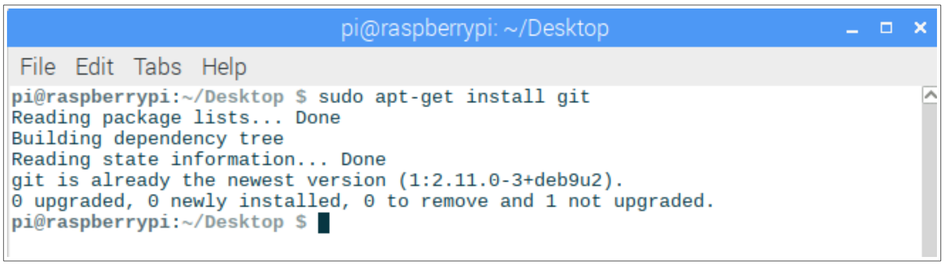
Step-5 - Install the *lexruee* i2c library

One of the major changes to the implementation steps of the AM2315 is the use of a better i2c software library. *Lexruee* has created a robust library that works with both Python and C. Installing it is a snap.

Open a command window and move to the *Download* folder: $cd *Download*.

Next, make sure *git* is installed on your Pi. Enter the command: $*sudo apt-get install git.*
You should see the output shown below in Figure-7.

Figure-7



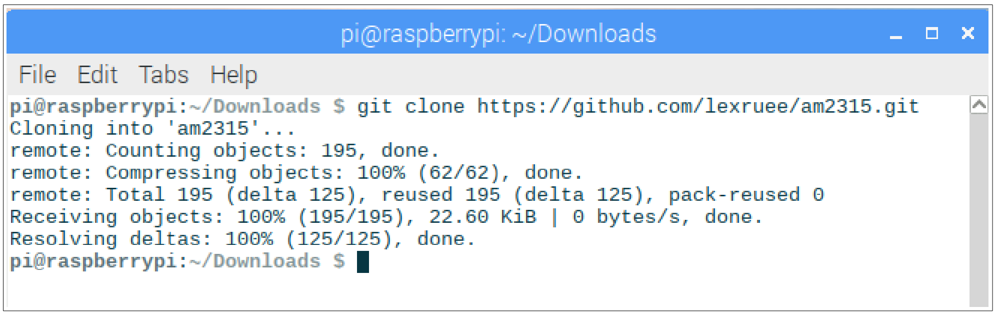
```
pi@raspberrypi:~/Desktop $ sudo apt-get install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.11.0-3+deb9u2).
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
pi@raspberrypi:~/Desktop $ 
```

If *git* is not installed on your Pi, the above command will install it.

Next, enter the command: $*git clone https://github.com/lexruee/am2315.git*.
This will download the source code for the *lexruee* i2c library.

You should see the output shown in Figure-8.

Figure-8

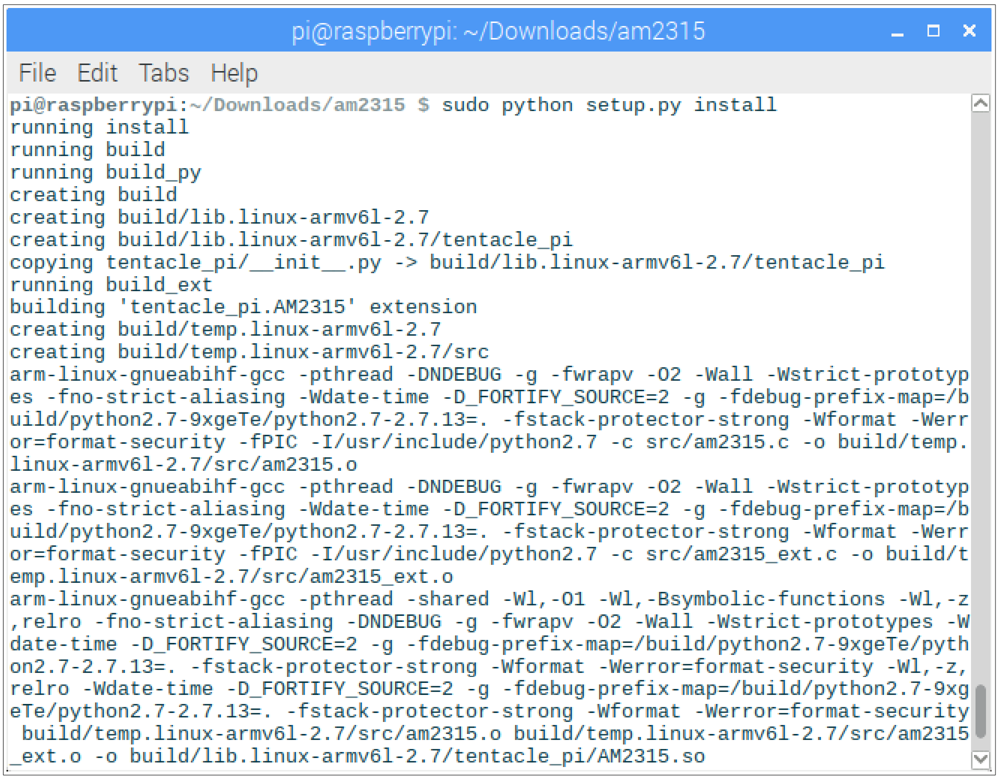


```
pi@raspberrypi:~/Downloads $ git clone https://github.com/lexruee/am2315.git
Cloning into 'am2315'...
remote: Counting objects: 195, done.
remote: Compressing objects: 100% (62/62), done.
remote: Total 195 (delta 125), reused 195 (delta 125), pack-reused 0
Receiving objects: 100% (195/195), 22.60 KiB | 0 bytes/s, done.
Resolving deltas: 100% (125/125), done.
pi@raspberrypi:~/Downloads $ 
```

The above command will create a folder named *am2315.* Move into that folder using the command: $*cd am2315.*

Next, build and install the library using the command: $*sudo python setup.py install*.
The build will start as shown in Figure-9.

Figure-9



```
pi@raspberrypi:~/Downloads/am2315 $ sudo python setup.py install
running install
running build
running build_py
creating build
creating build/lib.linux-armv6l-2.7
creating build/lib.linux-armv6l-2.7/tentacle_pi
copying tentacle_pi/__init__.py -> build/lib.linux-armv6l-2.7/tentacle_pi
running build_ext
building 'tentacle_pi.AM2315' extension
creating build/temp.linux-armv6l-2.7
creating build/temp.linux-armv6l-2.7/src
arm-linux-gnueabihf-gcc -pthread -DNDEBUG -g -fwrapv -O2 -Wall -Wstrict-prototyp
es -fno-strict-aliasing -Wdate-time -D_FORTIFY_SOURCE=2 -g -fdebug-prefix-map=/b
uild/python2.7-9xgeTe/python2.7-2.7.13=. -fstack-protector-strong -Wformat -Werr
or=format-security -fPIC -I/usr/include/python2.7 -c src/am2315.c -o build/temp.
linux-armv6l-2.7/src/am2315.o
arm-linux-gnueabihf-gcc -pthread -DNDEBUG -g -fwrapv -O2 -Wall -Wstrict-prototyp
es -fno-strict-aliasing -Wdate-time -D_FORTIFY_SOURCE=2 -g -fdebug-prefix-map=/b
uild/python2.7-9xgeTe/python2.7-2.7.13=. -fstack-protector-strong -Wformat -Werr
or=format-security -fPIC -I/usr/include/python2.7 -c src/am2315_ext.c -o build/t
emp.linux-armv6l-2.7/src/am2315_ext.o
arm-linux-gnueabihf-gcc -pthread -shared -Wl,-O1 -Wl,-Bsymbolic-functions -Wl,-z
,relro -fno-strict-aliasing -DNDEBUG -g -fwrapv -O2 -Wall -Wstrict-prototypes -W
date-time -D_FORTIFY_SOURCE=2 -g -fdebug-prefix-map=/build/python2.7-9xgeTe/pyth
on2.7-2.7.13=. -fstack-protector-strong -Wformat -Werror=format-security -Wl,-z,
relro -Wdate-time -D_FORTIFY_SOURCE=2 -g -fdebug-prefix-map=/build/python2.7-9xg
eTe/python2.7-2.7.13=. -fstack-protector-strong -Wformat -Werror=format-security
 build/temp.linux-armv6l-2.7/src/am2315.o build/temp.linux-armv6l-2.7/src/am2315
_ext.o -o build/lib.linux-armv6l-2.7/tentacle_pi/AM2315.so
```

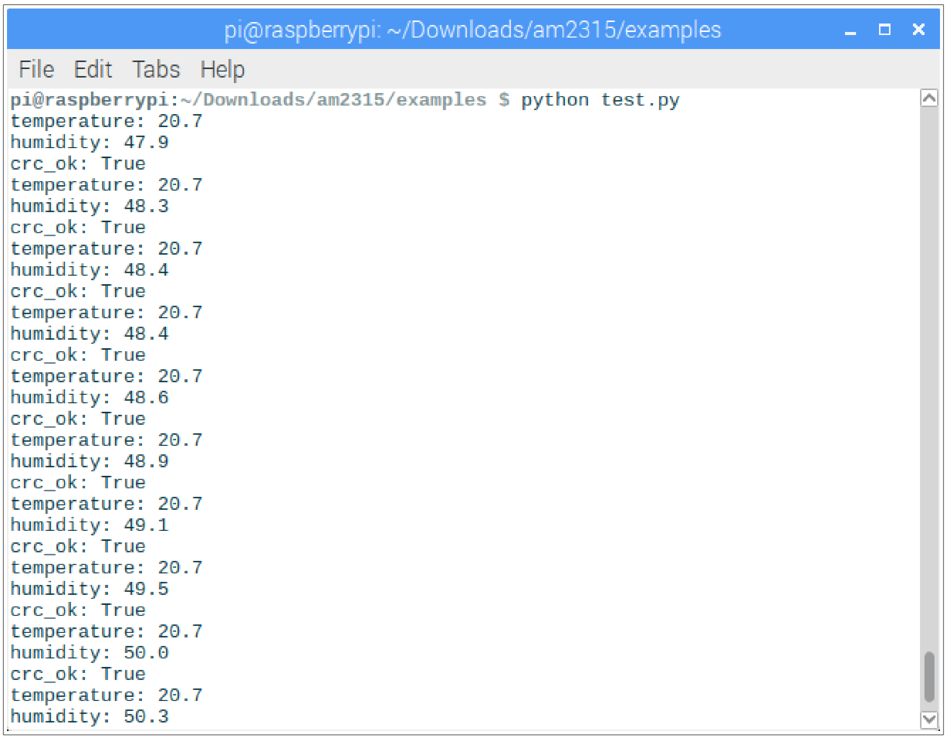
Once this step completes, you are ready to read data off the sensor.

Step-6 - Test the AM2315 software

The *lexruee* library comes with a small test program, so why not use it. In your command window change to the examples directory using the command: $*cd examples*.
Next, enter the command: $*python example.py.*

If the sensor is working correctly, you should see a 100 iteration loop of sensor data. This is shown in Figure-10 below.

Figure-10



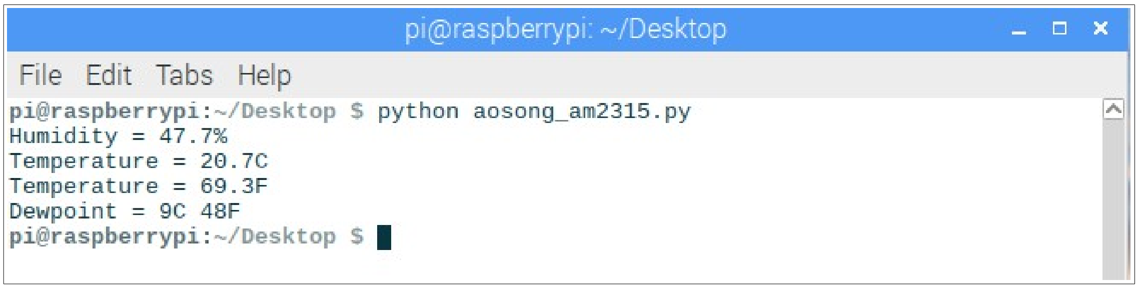
```
pi@raspberrypi:~/Downloads/am2315/examples $ python test.py
temperature: 20.7
humidity: 47.9
crc_ok: True
temperature: 20.7
humidity: 48.3
crc_ok: True
temperature: 20.7
humidity: 48.4
crc_ok: True
temperature: 20.7
humidity: 48.4
crc_ok: True
temperature: 20.7
humidity: 48.6
crc_ok: True
temperature: 20.7
humidity: 48.9
crc_ok: True
temperature: 20.7
humidity: 49.1
crc_ok: True
temperature: 20.7
humidity: 49.5
crc_ok: True
temperature: 20.7
humidity: 50.0
crc_ok: True
temperature: 20.7
humidity: 50.3
```

To exit the loop, press <Ctrl><c> or wait until it completes.

If you study the simple *example.py* script, you will see how easy it is to add the sensor data to your project. In many cases, this is all you need. Included in the *Sopwith* download is a Python script that does more than *example.py*. Copy the script to your Pi and run it with the command:
$*python aosong_am2315.py.*

You should see the output shown in Figure-11.

Figure-11



```
pi@raspberrypi:~/Desktop $ python aosong_am2315.py
Humidity = 47.7%
Temperature = 20.7C
Temperature = 69.3F
Dewpoint = 9C 48F
pi@raspberrypi:~/Desktop $ 
```

Notice this script provides a Fahrenheit temperature and calculates the dew point. Again, study the code and tweak it to fit your needs. If you are new to Python, both these scripts are simple enough to show you how easy Python is to learn.

Summary

The implementation of the AM2315 temperature sensor is much easier than it was in 2014. With this new code, Python3 is not longer required, nor is the obsolete *quick2wire* library. The new *Raspian Switch* OS runs on every Pi ever built. This means you can connect your AM2315 to any Pi you want.

Contact me if you have any issues getting you sensor working. Bug reports and enhancement requests are also welcome. Send them to sopwith@ismellsmoke.net.

Sopwith - November 12, 2017