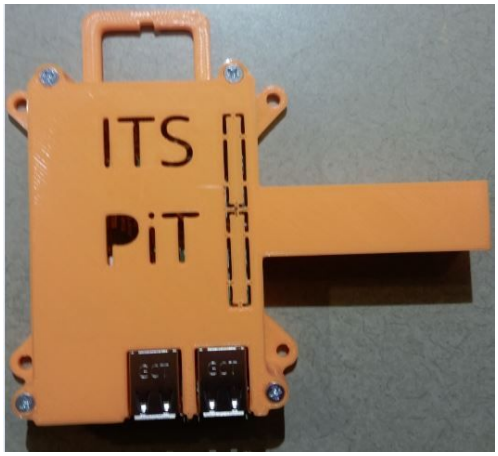
PiT Single v1.0
Assembly and Configuration Guide

Documentation v1.1

Overview:

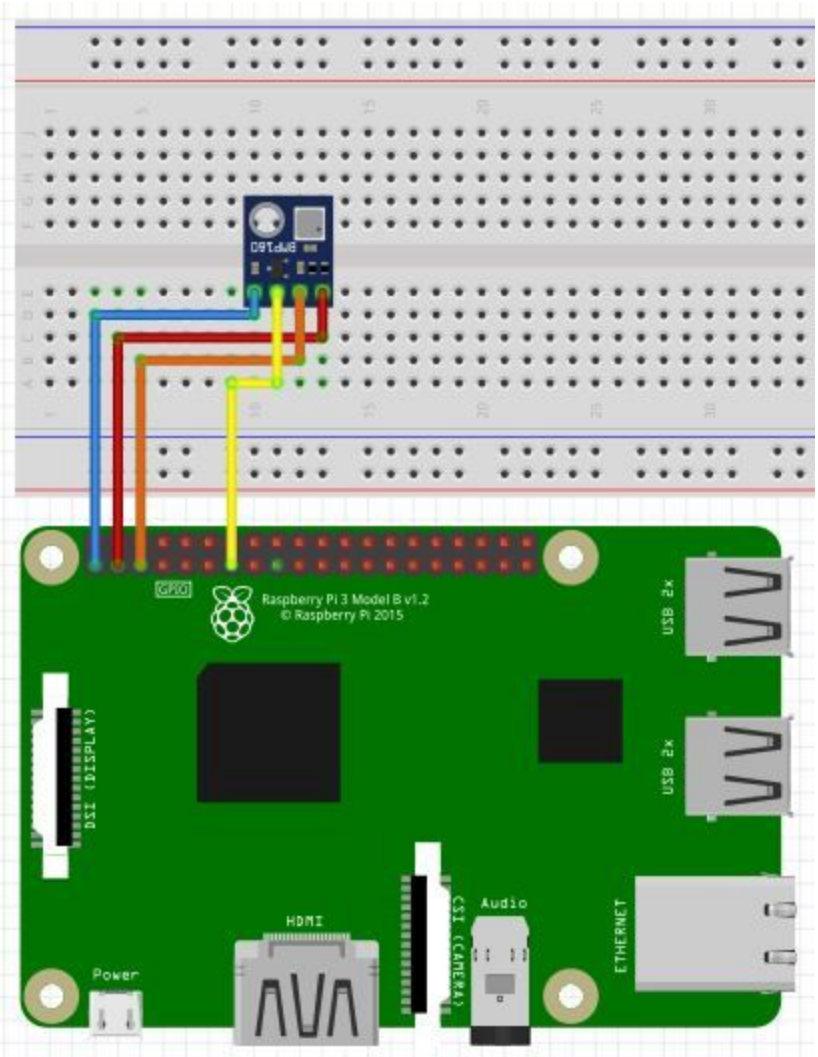
This guide will walk you through the steps needed to assemble and configure your very own PiT unit. PiT stands for ‘Pi Temperature Sensor’, but can be expanded to do more than just measure temperature. At its core, a PiT is just a Raspberry Pi with sensors attached to it that can be polled using SNMP.

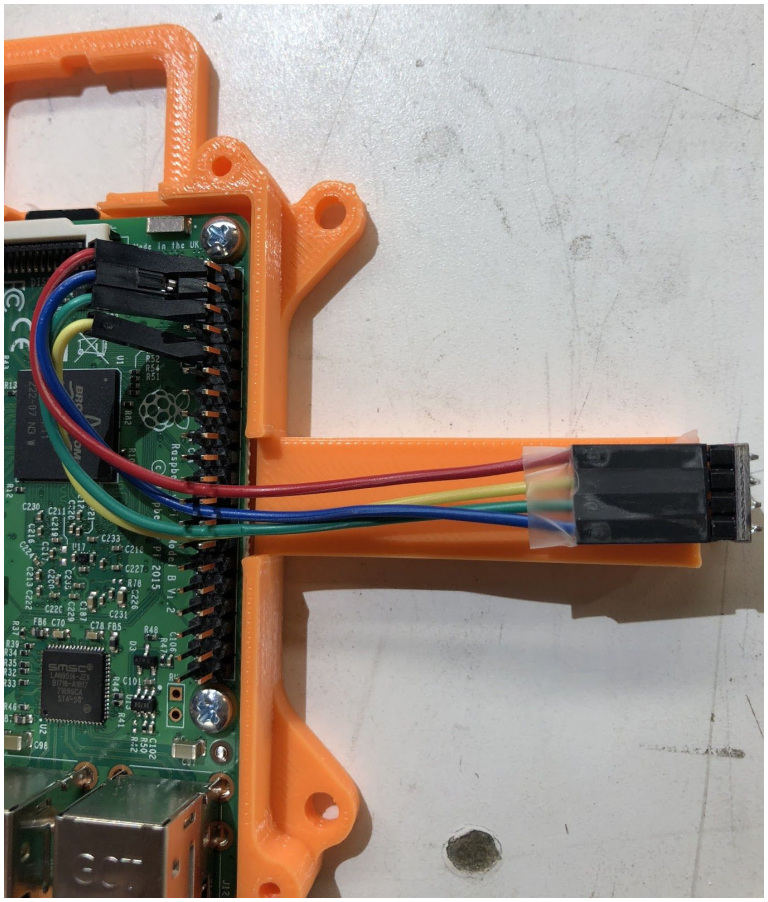


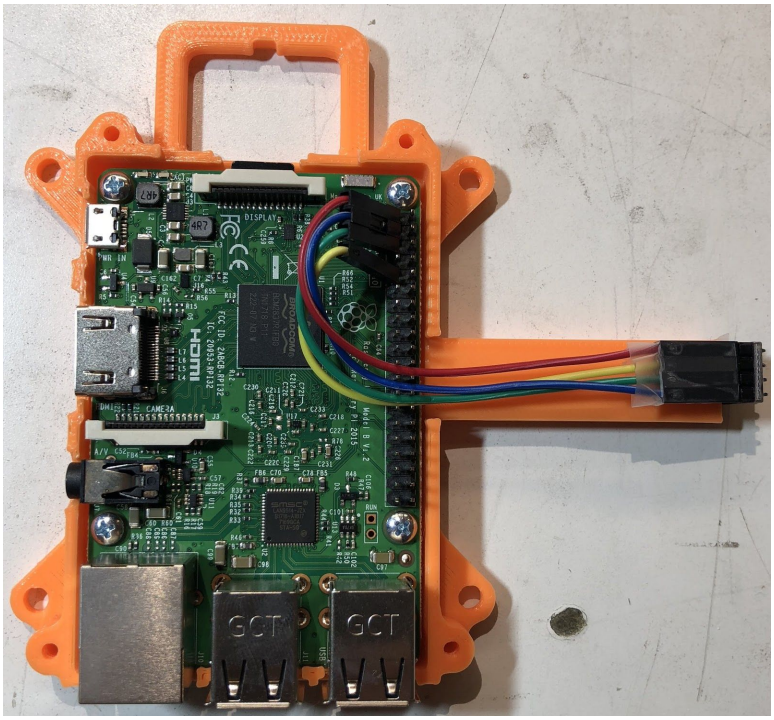
What You’ll Need:

- PiT Case File
- Raspberry Pi 3
- Rasbian OS Image
- 2.5 x 8 mm screw to attach sensor to case.

Adafruit Sensor Wiring Guide







Chinese Sensor Wiring Guide
Configuring the PiT

= Executable commands

= Additional Info

Updating and Upgrading

*****Some SD cards get corrupted on an Update/Upgrade depending on manufacturer and/or speed**

After assembling and powering on your PiT unit, updating and upgrading the Rasbian OS running on the Pi is recommended.

```
sudo apt-get update && apt-get dist-upgrade
```

Enabling I2C

I2C allows the sensors connected to the PiT unit to talk back to the Raspberry Pi. This will need to be enabled on the PiT unit. Connect to the PiT console (SSH/Monitor+Keyboard) and enable I2C on the Raspberry Pi.

```
sudo raspi-config
Interfacing Options > I2C > Enable
```

Installing Packages

There are a few packages that need to be installed for your PiT Unit to operate properly.

```
sudo apt-get --yes install i2c-tools python-smbus snmpd snmp sshpass
```

Sensor Detection

Test to see if the temperature sensor is being detected by I2C. The Chinese Temperature Sensors will show as a 76 while the Adafruit Temperature Sensors will show a 77.

```
i2cdetect -y 1
```

I2C Access

Allow the snmp user access to I2C.

```
sudo adduser snmp i2c
```

PiT Directory and Files

Adafruit_BME280.py	Initializes sensor and provides code for pulling values from it. Referenced by other scripts.
cpu-temp.sh	Pulls system CPU Temperature
snmp-baro-c.sh | snmp-baeo-c.py	Pulls barometric pressure
snmp-humd-c.sh | snmp-humd-c.py **snmp-humd-a.sh | snmp-humd-a.py**	Pulls humidity
snmp-temp-c.sh | snmp-temp-c.py **snmp-temp-a.sh | snmp-temp-a.py**	Pulls temperature
tempOffset.txt	Allows user to set a temperature offset to “calibrate” sensors

C = Chinese Sensor | A = Adafruit Sensor

Now we need to setup the PiT Directory (/usr/local/bin/PiT), this is where almost all PiT related activities take place on the backend. We also change ownership of /bin/ and /bin/PiT/ to the pi user.

```
sudo mkdir /usr/local/bin/PiT

sudo chown pi /usr/local/bin
sudo chown pi /usr/local/bin/PiT
```

Now that we have the PiT directory created with the proper ownership, we need to unpack the PiT TAR file into the newly created PiT Directory. This will place all the necessary backend scripts onto your PiT Unit.

PiT TAR File *****This is out-of-date!**

```
sudo tar -zxvf [Insert PiT TAR File Name] -C /usr/local/bin/PiT
```

We now need to make the system see the scripts we just moved into the PiT Directory as executables.

```
sudo chmod u+x /usr/local/bin/PiT/*.sh
sudo chmod u+x /usr/local/bin/PiT/*.py
```

Next, we need to install the necessary pre-compiled Python scripts.

Python Scripts

```
sudo tar -zxvf 2017-06-20-egg.tar.gz -C /usr/local/lib/python2.7/dist-packages
```

Testing PiT Scripts

After setting up the PiT Directory and installing the appropriate components, we should test the PiT Scripts directly to ensure they work before we point SNMPD to them. They should all return a decimal value if operating appropriately.

Chinese Sensor:

```
sudo python /usr/local/bin/PiT/snmp-temp-c.py 0x76
sudo python /usr/local/bin/PiT/snmp-humd-c.py 0x76
sudo python /usr/local/bin/PiT/snmp-baro-c.py 0x76
```

Adafruit Sensor:

```
sudo python /usr/local/bin/PiT/snmp-temp-a.py 0x77
sudo python /usr/local/bin/PiT/snmp-humd-a.py 0x77
```

Configure SNMPD for PiT Scripts

We now need to configure SNMPD to call our PiT Scripts when a SNMPD call is made to a particular OID. Open up the SNMPD configuration file.

```
sudo nano /etc/snmp/snmpd.conf
```

The following line needs to be commented out:

```
agentAddress udp:127.0.0.1:161 (Add a # in front of this line)
```

Add this line following the one you just commented out:

```
agentAddress udp:161
```

Find the following line:

```
#rocommunity public localhost
```

Add this line below it:

```
rocommunity public
```

Find the following line:

```
#  "Pass-through" MIB extension command
```

Add this line below it:

```
pass .1.3.6.1.2.1.25.1.20 /bin/sh /usr/local/bin/cpu-temp.sh
```

Add these lines as well **DEPENDING ON WHAT SENSOR YOU'RE USING!**:

(Adafruit Sensor)

```
pass .1.3.6.1.2.1.25.1.8 /bin/sh /usr/local/bin/PiT/snmp-temp-a.sh
pass .1.3.6.1.2.1.25.1.9 /bin/sh /usr/local/bin/PiT/snmp-humd-a.sh
```

(Chinese Sensor)

```
pass .1.3.6.1.2.1.25.1.8 /bin/sh /usr/local/bin/PiT/snmp-temp-c.sh
pass .1.3.6.1.2.1.25.1.9 /bin/sh /usr/local/bin/PiT/snmp-humd-c.sh
pass .1.3.6.1.2.1.25.1.10 /bin/sh /usr/local/bin/PiT/snmp-baro-c.sh
```

With all the appropriate changes made to the SNMPD config, we need to restart the snmpd daemon.

```
sudo service snmpd restart
```

Testing SNMPD

Now we need to test SNMP to ensure it is using the PiT Scripts and returning values appropriately.

```
snmpget -v2c -cpublic localhost .1.3.6.1.2.1.25.1.8
snmpget -v2c -cpublic localhost .1.3.6.1.2.1.25.1.9
snmpget -v2c -cpublic localhost .1.3.6.1.2.1.25.1.10
```

Remote PiT Upgrades

Overview

A number of bash scripts and additional files have been created to simplify the upgrading of multiple PiT Units when changes are made or additional scripts are added. As with everything else, these scripts live in the /usr/local/bin/PiT/ directory.

IPs.txt	Text file used to store the IPs of all PiT Units to be upgraded. Each IP should be on its own line. Referenced by the pre-remoteUpgrade.sh and remoteUpgrade.sh scripts.
pre-remoteUpgrade.sh	Used to obtain the SSH keys of all the hosts in the IPs.txt file. User will need to manually type “Yes” for each host. The remoteUpgrade.sh script will fail if you do not have all the necessary SSH keys already installed on the PiT that is pushing the upgrade.
remoteUpgrade.sh	1.) Creates TAR of /PiT/ directory along with a copy of the snmpd.conf 2.) Pushes TAR to remote PiTs, recreates /PiT/ directory, and overwrites snmpd.conf with the updated version Does not affect tempOffset.txt on remote PiT Units.