

ET-IOT HAT

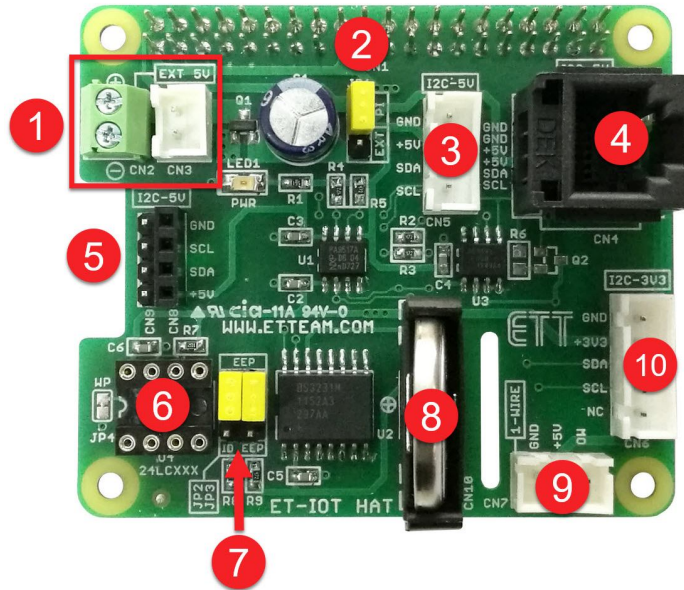


Board ET-IOT HAT is particularly designed for using together with Board Raspberry PI by using I2C BUS Interface; so, it can communicate to Board INPUT, OUTPUT or Board SENSORS in the format of I2C BUS Interface or 1-WIRE Interface effectively.

SPECIFICATIONS OF BOARD ET-IOT HAT

- Can be used with Board Raspberry PI by using I2C BUS Interface
- Connecting part of Circuit EEPROM (**OPTION**) can be used either as ID EEPROM or general EEPROM as required
- Part of Circuit RTC (DS3231) is used as Time Base of Board Raspberry, provided with BATTERY BACKUP
- Connecting position of I2C BUS can be connected to both 3.3V and 5V external devices
- Circuit I2C to 1-Wire (DS2482-100) can be connected to 1-Wire Device
- Connector External POWER SUPPLY 5VDC can provide power for the connected board; it can use power either from Board Raspberry PI or from External Board by setting JUMPER
- PCB Size: 6.5 x 5.6 cm.

COMPONENTS OF BOARD ET-IOT HAT



- **No.1** Connector External POWER SUPPLY 5VDC provides power for the connected board when the energy source of Board Raspberry PI hasn't got enough POWER SUPPLY to provide for the circuit.
- **No.2** JUMPER (JP1) chooses 5V energy source between from Board Raspberry PI (PI) or from External Board (EXT).
- **No.3** CONNECTOR I2C BUS WAFER 4PIN (CN5) can be used with 5V System.
- **No.4** CONNECTOR I2C BUS RJ11 6PIN (CN4) can be used with 5V System.
- **No.5** CONNECTOR I2C BUS HEADER 4PIN can be used with 5V System.
- **No.6** Connecting Area of Circuit EEPROM (**OPTION**) can be used as either ID EEPROM or general EEPROM as required.
- **No.7** JUMPER (JP2, JP3) chooses EEPROM Interface either to be ID EEPROM (ID EEP Position) or general EEPROM (EEP Position)
- **No.8** BATTERY BACKUP (CR2032) is used for CIRCUIT RTC (DS3231).
- **No.9** CONNECTOR 1-WIRE WAFER 3PIN (CN7) can be used with 5V System.
- **No.10** CONNECTOR I2C BUS WAFER 5PIN (CN6) can be used with 3.3V System.

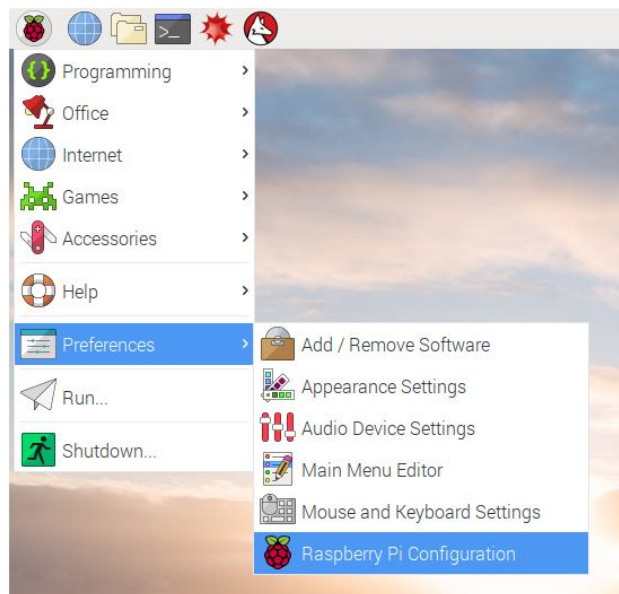
HOW TO CONNECT BOARD ET-IOT HAT TO BOARD RASPBERRY PI



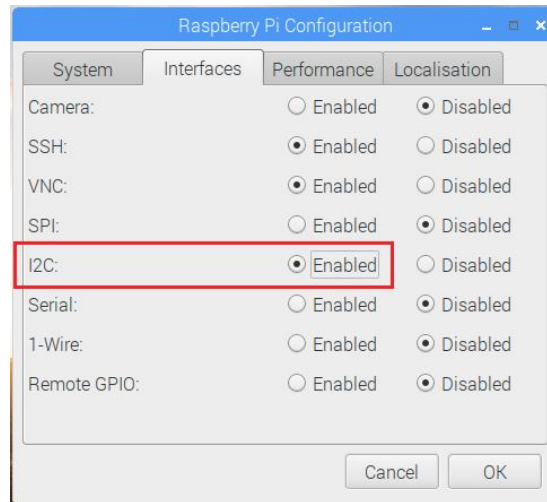
HOW TO SETUP VALUES OF BOARD RASPBERRY PI FOR USING WITH BOARD ET-IOT HAT

Board ET-IOT HAT uses I2C Bus Interface; so, it requires enabling the I2C BUS Interface on Board Raspberry PI first, please follow the instructions below. The example below shows how to connect Board ET-IOT HAT to Board Raspberry PI3 Mode B and it runs by the Operating System of **RASPBIAN STRETCH WITH DESKTOP** Version: **April 2018** Release date: **2018-04-18**.

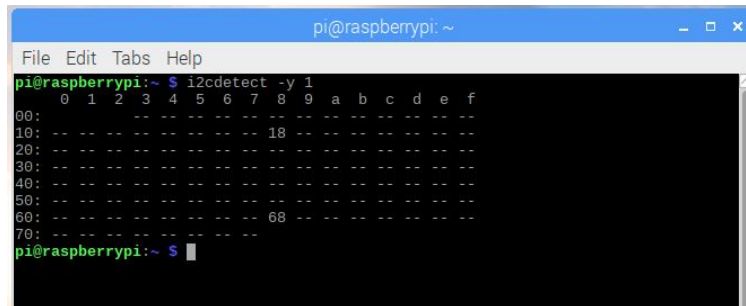
1. Connect Board ET-IOT HAT together with Board Raspberry PI and then provide POWER SUPPLY into Board Raspberry PI completely.
2. Choose Menu **Preferences --- Raspberry Pi Configuration**



3. Choose Tab **Interface**, choose **I2C** as **Enable** and then click **OK** as shown in the picture. Next, please reboot Board Raspberry Pi.



4. Next. Open **Program Terminal** and type the Command **i2cdetect -y 1** as shown in the picture below

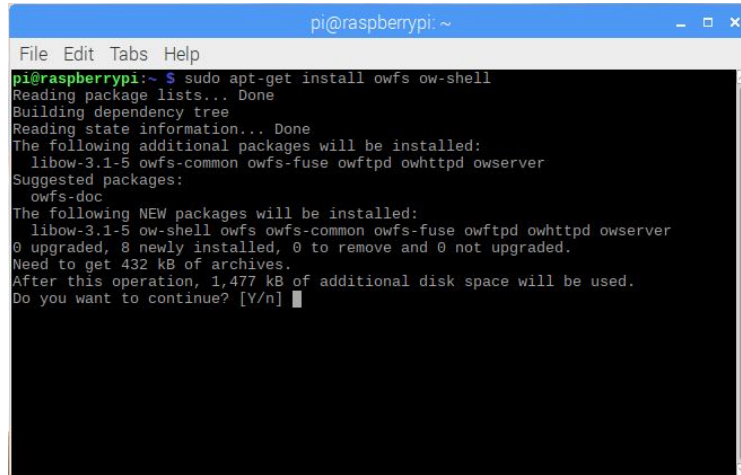


Then, the Program starts scanning and looking at the actually connected I2C devices on Board ET-IOT HAT; **DS2482-100 Address 18** and **DS3231 Address 68**. If the program shows these values, it means that Board ET-IOT HAT is ready to run.

HOW TO USE PROBE DS18B20 HAT TOGETHER WITH BOARD ET-IOT HAT

PROBE DS18B20 HAT uses Temperature Sensor "DS18B20" that is 1-Wire Interface; so, it requires connecting PROBE DS18B20 HAT at CONNECTOR 1-Wire WAFER 3PIN (CN7). This example shows how to write program to communicate to 1-Wire device by using **OWFS 1-Wire File System**, please follows the instructions below;

1. Open **Program Terminal** and type the Command **sudo apt-get install owfs ow-shell** to install **owfs** and **ow-shell**.

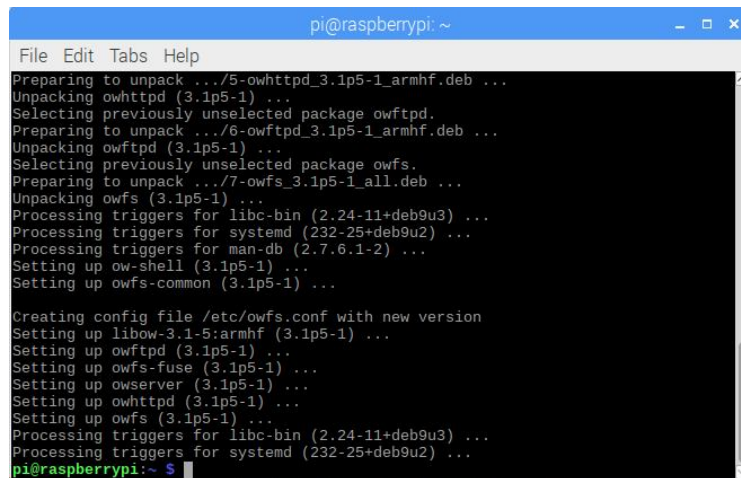


```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo apt-get install owfs ow-shell
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libow-3.1-5 owfs-common owfs-fuse owftpd owhttpd owserver
Suggested packages:
  owfs-doc
The following NEW packages will be installed:
  libow-3.1-5 ow-shell owfs owfs-common owfs-fuse owftpd owhttpd owserver
0 upgraded, 8 newly installed, 0 to remove and 0 not upgraded.
Need to get 432 kB of archives.
After this operation, 1,477 kB of additional disk space will be used.
Do you want to continue? [Y/n]

```

2. Type **Y** and press **ENTER** to confirm; and please wait until the installation process is complete.



```

pi@raspberrypi: ~
File Edit Tabs Help
Preparing to unpack .../5-owhttpd_3.1p5-1_armhf.deb ...
Unpacking owhttpd (3.1p5-1) ...
Selecting previously unselected package owftpd.
Preparing to unpack .../6-owftpd_3.1p5-1_armhf.deb ...
Unpacking owftpd (3.1p5-1) ...
Selecting previously unselected package owfs.
Preparing to unpack .../7-owfs_3.1p5-1_all.deb ...
Unpacking owfs (3.1p5-1) ...
Processing triggers for libc-bin (2.24-11+deb9u3) ...
Processing triggers for systemd (232-25+deb9u2) ...
Processing triggers for man-db (2.7.6.1-2) ...
Setting up ow-shell (3.1p5-1) ...
Setting up owfs-common (3.1p5-1) ...

Creating config file /etc/owfs.conf with new version
Setting up libow-3.1-5:armhf (3.1p5-1) ...
Setting up owftpd (3.1p5-1) ...
Setting up owfs-fuse (3.1p5-1) ...
Setting up owserver (3.1p5-1) ...
Setting up owhttpd (3.1p5-1) ...
Setting up owfs (3.1p5-1) ...
Processing triggers for libc-bin (2.24-11+deb9u3) ...
Processing triggers for systemd (232-25+deb9u2) ...
pi@raspberrypi:~ $

```

3. Create new Folder name **1wire** in **/mnt** by using the Command **sudo mkdir /mnt/1wire**
4. Open File **owfs.conf** to edit value by using the Command **sudo nano /etc/owfs.conf**; next, type the number sign (#) in front of the message
server: FAKE = DS18S20,DS2405 as shown in the picture below;

```

pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.7.4 File: /etc/owfs.conf Modified
# Sample configuration file for the OWFS suite for Debian GNU/Linux.
#
#
# This is the main OWFS configuration file. You should read the
# owfs.conf(5) manual page in order to understand the options listed
# here.
##### SOURCES #####
#
# With this setup, any client (but owserver) uses owserver on the
# local machine...
! server: server = localhost:4304
#
# ...and owserver uses the real hardware, by default fake devices
# This part must be changed on real installation
#server: FAKE = DS18S20_DS2405
#
# USB device: DS9490
#server: usb = all
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^N Replace ^U Uncut Text ^T To Spell ^_ Go To Line
    
```

5. The following messages must be typed below the file.

```

device = /dev/i2c-1
mountpoint = /mnt/1wire
Celsius
allow_other
error_print = 0
error_level = 0
    
```

Next, please press **CTRL+X** and **y** to save the new values.

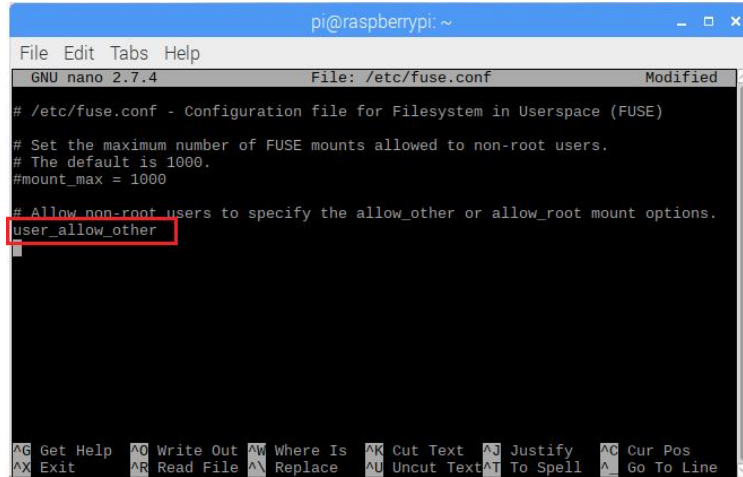
```

pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.7.4 File: /etc/owfs.conf Modified
ftp: port = 2120
##### OWSERVER #####
server: port = localhost:4304
device = /dev/i2c-1
mountpoint = /mnt/1wire
Celsius
allow_other
error_print = 0
error_level = 0
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^N Replace ^U Uncut Text ^T To Spell ^_ Go To Line
    
```

6. Open File **fuse.conf** to edit by using the Command **sudo nano /etc/fuse.conf**; next, remove the number sign (#) in front of the message

user_allow_other

And then press **CTRL+X** and **y** to save the new value.



```

pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.7.4 File: /etc/fuse.conf Modified
# /etc/fuse.conf - Configuration file for Filesystem in Userspace (FUSE)
# Set the maximum number of FUSE mounts allowed to non-root users.
# The default is 1000.
#mount_max = 1000
# Allow non-root users to specify the allow_other or allow_root mount options.
user_allow_other
AG Get Help AO Write Out AW Where Is AK Cut Text AJ Justify AC Cur Pos
AX Exit AR Read File AL Replace AU Uncut Text AT To Spell AA Go To Line

```

7. Reboot Board Raspberry PI. After rebooted completely, please type the Command **sudo owfs** in Program Terminal to stimulate **owfs** to run.

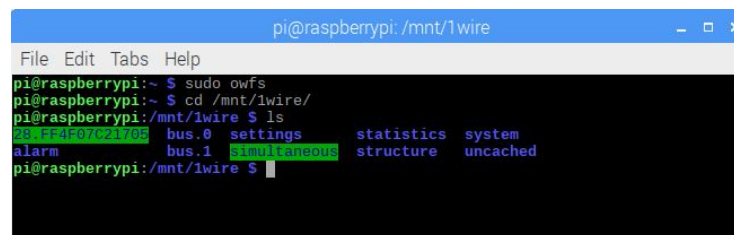


```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo owfs
pi@raspberrypi:~ $

```

8. Type the Command **cd /mnt/1wire/** to enter the Folder name **1wire**; next, type the Command **ls** to see more details as shown in the picture below;



```

pi@raspberrypi: /mnt/1wire
File Edit Tabs Help
pi@raspberrypi:~ $ sudo owfs
pi@raspberrypi:~ $ cd /mnt/1wire/
pi@raspberrypi:/mnt/1wire $ ls
28_FF4F07C21705 bus.0 settings statistics system
alarm bus.1 simultaneous structure uncached
pi@raspberrypi:/mnt/1wire $

```

Then, user can see the Folder name **28.FF4F07C21705** that is the identity code of DS18B20 and each identity code is different.

9. Type the Command **cd 28.FF4F07C21705** to enter the Folder name **28.FF4F07C21705**; next , type the Command **ls** to see more details as shows in the picture below;

```

pi@raspberrypi: /mnt/1wire/28.FF4F07C21705
File Edit Tabs Help
pi@raspberrypi:~ $ sudo owfs
pi@raspberrypi:~ $ cd /mnt/1wire/
pi@raspberrypi:/mnt/1wire $ ls
28.FF4F07C21705 bus.0 settings statistics system
alarm bus.1 simultaneous structure uncached
pi@raspberrypi:/mnt/1wire $ cd 28.FF4F07C21705
pi@raspberrypi:/mnt/1wire/28.FF4F07C21705 $ ls
address family locator r_locator temperature11 templo
alias fasttemp power scratchpad temperature12 tempres
crc8 id r_address temperature temperature9 type
sprata latesttemp r_id temperature10 temphigh
pi@raspberrypi:/mnt/1wire/28.FF4F07C21705 $
    
```

Then, user can see folders that include details of DS18B20 and it is read-out.

10. Test the operation by using the Command **cat temperature** to read temperature value; it shows the temperature value of 26.375 degrees Celsius as shown in the picture below.

```

pi@raspberrypi: /mnt/1wire/28.FF4F07C21705
File Edit Tabs Help
pi@raspberrypi:~ $ sudo owfs
pi@raspberrypi:~ $ cd /mnt/1wire/
pi@raspberrypi:/mnt/1wire $ ls
28.FF4F07C21705 bus.0 settings statistics system
alarm bus.1 simultaneous structure uncached
pi@raspberrypi:/mnt/1wire $ cd 28.FF4F07C21705
pi@raspberrypi:/mnt/1wire/28.FF4F07C21705 $ ls
address family locator r_locator temperature11 templo
alias fasttemp power scratchpad temperature12 tempres
crc8 id r_address temperature temperature9 type
sprata latesttemp r_id temperature10 temphigh
pi@raspberrypi:/mnt/1wire/28.FF4F07C21705 $ cat temperature
26.375pi@raspberrypi:/mnt/1wire/28.FF4F07C21705 $
    
```

11. Referred to this example, it always types the Command **sudo owfs** every time user opens Board Raspberry PI to stimulate **owfs** to run. Please use the Command **sudo crontab -e** and **owfs** auto-runs every time user opens the Board as shown in the picture below.

```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo crontab -e
no crontab for root - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/ed
 2. /bin/nano <---- easiest
 3. /usr/bin/vim.tiny

Choose 1-3 [2]:
    
```


12. Choose **No.2** and press **ENTER**. Next, type the message **@reboot sudo -u root owfs** below the file and **owfs** always auto-runs after rebooted completely. Then, press **CTRL+X** and **y** to save the new value.

```

pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.7.4 File: /tmp/crontab.FLersR/crontab Modified
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow command
@reboot sudo -u root owfs
AG Get Help AC Write Out AW Where Is AK Cut Text AJ Justify AC Cur Pos
AX Exit AR Read File AN Replace AU Uncut Text AT To Spell A Go To Line
    
```

13. Test the operation by writing program by Python Language (DS18B20.py) to read and show the value on Program Terminal. Some part of program must be edited that is identity code of DS18B20 because each value is different; so, user needs to edit the value according to user's requirement. Please refer to the example, the identity code is **28.FF4F07C21705**

```

DS18B20.py - /home/pi - Geany
File Edit Search View Document Project Build Tools Help
Symbols
Variables
  t [21]
  t_raw [18]
  temp [17]
Imports
  time [14]
6 #
7 # Board MCU : RASPBERRY PI 3 MODEL B
8 # Editor-IDE : Geany 1.29
9 # Target Board Interface : ET-IOT HAT + PROBE DS18B20 HAT (1-WIRE)
10 # Thanks http://raspberrypi.tomasgreno.cz/
11 #
12 #*****
13
14 import time
15
16 while True:
17     temp = open("/mnt/1wire/28.FF4F07C21705/temperature", "r")
18     t_raw = temp.read()
19     temp.close()
20
21     t = float(t_raw)
22
23     print ("Temperature : %.2f C" %t) # Print Temp to screen
24
25     time.sleep(0.1)
26
08:35:19: This is Geany 1.29.
08:35:19: New file "untitled" opened.
08:35:29: File untitled closed.
08:35:29: File /home/pi/DS18B20.py opened(1).
08:38:20: File /home/pi/DS18B20.py saved.
line: 25 / 26 col: 19 sel: 0 INS TAB mode: LF encoding: UTF-8 filetype: Python scope: unknown
    
```

14. Finally, save the Code Program and run the program; in this case, user may go to Program Terminal and use the Command `python3 DS18B20.py` and the program starts running as shown in the picture below;

