Raspberry Pi Temperature Readout User Manual

Table of Contents

Quick Start Guide....1
Software Install....2
Software Un-Install....2
Setting up NTCs....2
Setting up thermocouples....2
Stacking Multiple Boards....3
Logging....4
Firmware Update....4
Firmware Recovery....4
Manual Operation....6
List of Commands....7
-help....7
-board_info....7
-start....7
-config....7
-read_settings....7
-i2c_find....7
-i2c_change_addr <new_addr>....7
-sw_rev....7
-read_conf....8
-fw_update <file_name>....8
-ntc_calc_beta <C/F> <t1> <r1> <t2> <r2>....8
-init_hw....8
-read_sensor....8
-buzzer....8
Document Revisions....9

Quick Start Guide

1. Turn off the Raspberry Pi and disconnect power.
2. Connect the temperature readout to Raspberry Pi.
3. Connect any NTC sensors or thermocouples
4. Connect power to the raspberry Pi.
5. After the Raspberry Pi boots up install the software. See Software Install section below.
6. Run ***temp_readout.sh -board_info*** to read the board information and verify access to the board
7. Run ***temp_readout.sh -config*** to configure each sensor.
8. Save the configuration file in the previous step.
9. Run ***temp_readout.sh -start*** to start the software.

Software Install

1. Download software from the product page on the website, copy it to any folder on the Raspberry Pi.
2. Unzip the .zip file, for example ***sudo unzip <file_name>***
3. Go into the unzipped install directory, ***cd temperature_readout_linux_1.0/install***
4. Make install.sh executable with ***sudo chmod 777 install.sh***
5. Run ***sudo ./install.sh -install***
6. Run ***temp_readout.sh -help*** to see a full list of commands
7. Run ***temp_readout.sh -board_info*** to read the board information
8. The .zip file and unzipped folder are not used any more, feel free to delete them.

Software Un-Install

1. Run ***sudo /opt/temp_readout/install.sh -uninstall***

Note :

The default software install consists of the files stored inside /opt/temp_readout folder as well as temp_readout.sh inside /usr/local/bin.

Setting up NTCs

1. Connect NTC sensor to one the 32 male headers, note the vertical orientation or the sensor connector.
2. Open the .conf file with any text editor, the .conf files are located in /opt/temp_readout/conf. For example ***sudo nano /opt/temp_readout/conf/temp_readout_1.conf***
3. Enable each connected sensor by setting each NTC_X _EN variable to 1.
4. Set each of the NTC_X _R25 values to the 25 degree C resistance of the NTC. This value can be found in the NTC datasheet or calculated manually for unmarked NTCs.
5. Set each of the NTC_X _BETA values to the beta parameter of this NTC. This value can be found in the NTC datasheet, or calculated manually for unmarked NTCs. The ***-ntc_calc_beta*** command can help calculate the beta constant.
6. Adjust the two alarm values if needed, NTC_X_ALARM_HIGH and NTC_X_ALARM_LOW. If the temperature of that sensor goes outside of that range it will be shown on the screen and recorded in the log file. The buzzer will also beep if it's enabled.
7. Save the .conf file
8. Verify the settings with ***temp_readout.sh -read_settings***
9. Start the software with ***temp_readout.sh -start***

Setting up thermocouples

1. Connect thermocouple wires to the quick connects, gently pressing the spring buttons with something pointy can help with insertion. Unlike NTCs, thermocouples are polarized and need to be

connected a certain way. If one of the wires is red then that is usually the +, connect that to the pin labeled +. If it's not clear which wire is which, then just take a guess because it can be corrected later.

2. Open the .conf file with any text editor, the .conf files are located in /opt/temp_readout/conf. For example ***sudo nano /opt/temp_readout/conf/temp_readout_1.conf***
3. Enable each connected sensor by setting each THRM_X _EN variable to 1.
4. Set each of the THRM_X_TYPE variables to one of {K,J,N,R,S,T,B,E} to match the thermocouple.
5. Adjust the two alarm values if needed, THRM_X_ALARM_HIGH and THRM_X_ALARM_LOW. If the temperature of that sensor goes outside of that range it will be shown on the screen and recorded in the log file. The buzzer will also beep if it's enabled.
6. Save the .conf file
7. Verify the settings with ***temp_readout.sh -read_settings***
8. Start the software with ***temp_readout.sh -start***
9. Verify that the thermocouple is displaying the correct room temperature, press each one between fingers to see that the temperature spikes up a little bit. If the +/- polarity of the wires is incorrect it will be seen here and the wires can then be reversed.

Stacking Multiple Boards

Multiple boards can be stacked one on top of the other to increase the number of available sensors. Repeat the following steps when adding a second board, the key is to make sure that each board has a unique I2C address.

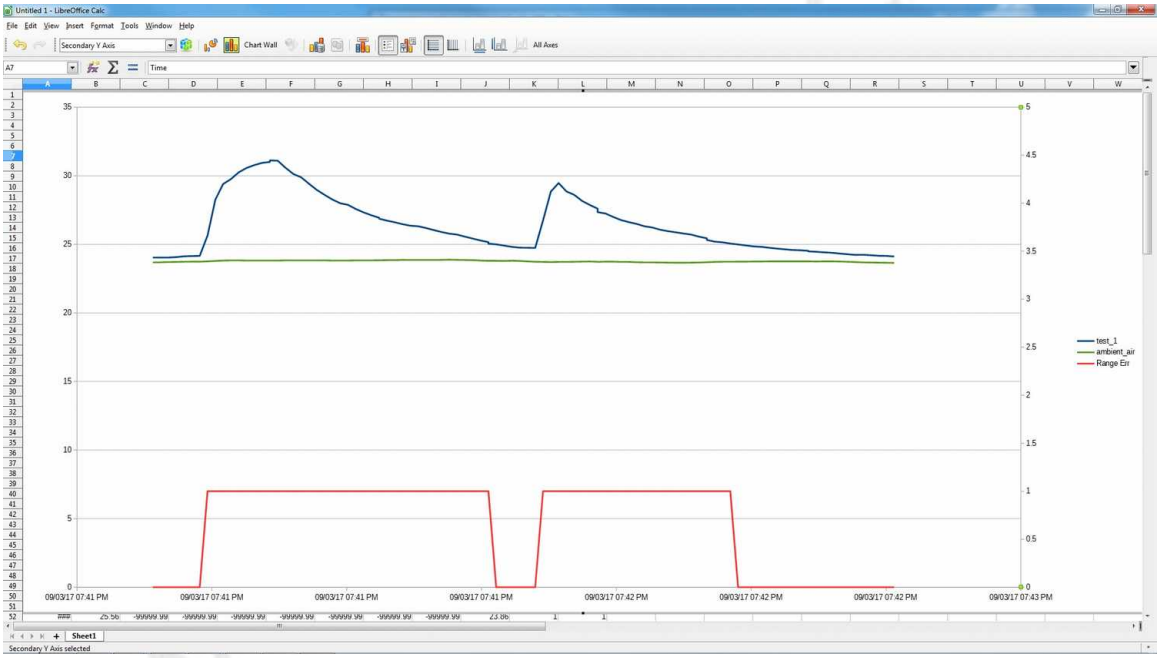
1. Change the I2C address of existing board to anything except 69, which is the default address programmed at the factory and will be the address of the new board soon to be connected. The command for that is ***temp_readout.sh -i2c_change_addr <new_addr>***
2. Turn off Raspberry Pi, call sudo -poweroff, wait 20 seconds then disconnect USB power.
3. Connect the new board board using the stacking hardware kit(taller standoff and stacking connector)
4. Connect USB power, wait for Raspberry Pi to boot up
5. Call ***temp_readout.sh -i2c_find*** and make sure the new board is recognized. It's I2C address should be the default 69.
6. Create a new .conf file for the second board by making a copy of an existing .conf file. The files are stored in ***/opt/temp_readout/conf***. For example
   sudo cd /opt/temp_readout/conf
   sudo cp temp_readout_1.conf temp_readout_2.conf
7. Inside the new .conf file update the I2C_ADDR variable to 69 which will match the new board.
   sudo nano temp_readout_2.conf
8. Setup each of the sensors in the new .conf file, save .conf file.
9. Repeat above steps for each additional board

Note :
When multiple boards are present, global settings such as units and logging to file are taken from the first .conf file. The first in terms of alphabetical order. For example if there are two .conf files, temp_readout_1.conf and temp_readout_2.conf, then the global settings will be taken from temp_readout_1.conf. The name of the file used for the global settings will be displayed during configuration immediately after the -start command is issued.

Some commands will require an additional parameter specifying the .conf file of the particular board. User will be prompted when this is required. For example if updating firmware, you need to specify which board the firmware is to be updated for.

Logging

1. To enable logging, change the SAVE_TO_FILE variable inside the .conf file to 1
2. The sample rate can be adjusted with the FILE_SAVE_DELAY_S variable, the integer value represents how many seconds to wait between samples.
3. MAX_FILE_SIZE_MB variable sets the maximum log file size in mega bytes.
4. The log file name is in the following format ***year_month_day_hour_minute_second_log.txt***.
5. If graphing is required, the contents of the log files can be copy-pasted into a spreadsheet and graphed. Use comma separation when pasting into a spreadsheet. A scatter graph works well for displaying temperatures of multiple sensors. Also moving the Range_err variable to a secondary Y-axis allows for a clean look. Adjust the secondary axis range so that Range_err is clearly visible. In the sample graph below Range_err gets set to 1 when test_1 is above 25C.



Firmware Update

1. Download the new firmware from website
2. unzip it, ***sudo unzip <file_name>***
1. run ***temp_readout.sh -fw_update <file_name>***

Firmware Recovery

If something like a power loss during a firmware update “bricked” the unit, follow the steps below to recover it.

1. Power off the Raspberry Pi and disconnect USB power.
1. Connect the two pins shown in photo below with a wire,paperclip or anything else conductive.
2. Connect USB power, the temperature readout should enter firmware recovery mode which is

indicated by the buzzer beeping quickly.

3. Wait until the Pi boots up
4. Update firmware normally, see “Update Firmware” section.
5. Power off the Raspberry Pi, disconnect USB power, remove the wire from the earlier step.
6. Power up the Raspberry Pi again.

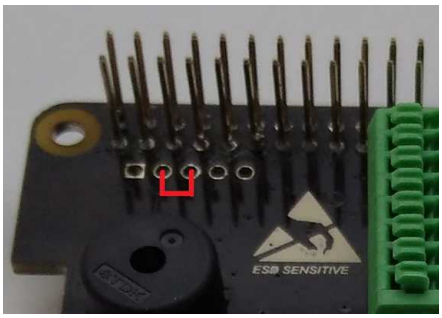


Illustration 1: Short these two pins to enter firmware recovery on power up.

Manual Operation

The -init_hw, -read_sensor, and -buzzer commands can be used for custom applications that need direct assess to sensor data. They allow reading one or more sensors from the command line or script, instead of monitoring them visually through the loop of the -start command.

To operate the hardware in manual mode.

1. Setup all the sensors to be polled in the .conf file, using ***temp_readout.sh -config***
2. Call ***temp_readout.sh -init_hw*** to setup the hardware
3. Poll the sensors with ***temp_readout.sh -read_sensor <sensor_num>.***

Example 1 :
Read NTC 12
temp_readout.sh -read_sensor 12

Example 2 :
Read NTCs 1 and 16 and thermocouple 4. The thermocouples map to numbers 33 to 36.
temp_readout.sh -read_sensor 1 16 36

Example 3:
Read multiple sensors from a script.

==========

```
#!/bin/bash

#Read NTCs 1 and 16 and thermocouple 4 and print the output.

temp=`temp_readout.sh -read_sensor 1 16 36`
parse=( $temp )
echo "NTC 1 = ${parse[0]}"
echo "NTC 16 = ${parse[1]}"
echo "Thermocouple 4 = ${parse[2]}"
```

==========

List of Commands

-help

Result : Prints a list of commands.

Example :
temp_readout.sh -help

-board_info

Result : Prints the board information like PCB revision.

Example :
temp_readout.sh -board info

-start

Result : Starts the software.
temp_readout.sh -start

-config

Result : Used to configure each of the sensors and other parameters.

Example :
temp_readout.sh -config

Note:
A text editor should be used to edit /opt/temp_readout/temp_readout.conf. Inside the file there are instructions on how to edit each parameter.

-read_settings

Result : Prints a summary of all of the settings from the .conf file

Example :
temp_readout.sh -read_settings

-i2c_find

Result : Finds the temperature readout board on the I2C bus, and updates I2C_ADDR variable inside the .conf file.

Example :
temp_readout.sh -i2c_find

-i2c_change_addr <new_addr>

Result : Changes the I2C address, and updates I2C_ADDR variable inside the .conf file.

Example :
temp_readout.sh -i2c_change_addr 25

-sw_rev

Result : Reads the Linux software revision

Example :
emp_readout.sh -sw_rev

-read_conf

Result : Prints the .conf file contents

Example :
temp_readout.sh -read_conf

-fw_update <file_name>

Result : Updates firmware

Example :
temp_readout.sh -fw_update <file.bin>

-ntc_calc_beta <C/F> <t1> <r1> <t2> <r2>

Result : Calculate Beta constant for an unknown NTC.
Units : C or F, representing Celsius and fahrenheit t1 : temperature point 1 r2 : resistance of NTC at temperature point 1 t2 : temperature point 2 r2 : resistance of NTC at temperature point 2

Example :
temp_readout.sh -ntc_calc_beta C 25.1 110966.4 66.7 17903.8

-init_hw

Result :Initializes the hardware prior to using the read_sensor command. See the Manual Operation section for details.

-read_sensor

Result : Returns the reading from one or more sensors, or “err” if there’s an error such as a disconnected sensor. Arguments can be one or more sensor numbers, from 1 to 36. Sensors 1 to 32 correspond to the 32 NTCs, and sensors 33 to 36 correspond to the 4 thermocouples. See the Manual Operation section for details.

To read multiple sensors in one shot, enter multiple sensor numbers in any order, doesn’t have to be incrementing. However the return values will be always be positioned with the ***lowest sensor number first***. For example ***temp_readout.sh -read_sensor 12 1 15*** will return ntc_1 ntc_12 ntc_15.

Example :
temp_readout.sh -read_sensor 1
temp_readout.sh -read_sensor 1 3 4 23 36

-buzzer

This command will beep the buzzer a few times then turn it off. It can be called repeatedly during manual operation while sensor temperature is outside of a required range. See the Manual Operation section for details

Example :
temp_readout.sh -buzzer

Document Revisions

Rev 1.0 :

- Original

Rev 1.1 :

- Added -sw_rev command.
- Removed FAQ section, will keep it on the webpage.

Rev 1.2 :

- Added Manual Operation section
- Added -init_hw command
- Added -read_sensor command
- Added -buzzer command