

/dev/random

All-around Linux / tech blog



Temperature measuring using Linux and Raspberry Pi

3.58 (71.58%) 19
votes

The story

I have a small server room, with some rack servers, NAS's, VMware ESXi's and other fun stuff, but I had one problems.. in the summer it got really hot, and I had to open the window to let fresh air in, and the old out to lower the temperature again, but I never knew when it was too hot, so i had to drive by the server room once or twice a day to manually feel if it was too hot..

This was a problem, because sometimes I was not home, or sometimes I forgot, or just had no time to drive by to check the temp.

Having a temperature sensor in the room, from a freezer or something like that didn't help me much, since I still had to drive by to check it manually.

So I had to make something I was able to get the room temperature from, over the internet/lan, and that's when I decided to make my network thermometer.

I looked for weeks on the internet for a good guide about this, but none of them worked. so I decided to write my own to help others with the same problem out.

This post will be about

How to make your own network thermometer, using open source software and a cheap USB temperature sensor!

What you will need

- A raspberry Pi, or other small Linux computer that uses a small amount of power (this can be a laptop or normal computer too)
- network cable to connect the Raspberry, laptop or computer to the network, the link is a 5 meter which

is enough in most cases, there are other length on the same page if you need more.

- USB temperature sensor, I used a “[TEMPer v1.4](#)“

If you are planning on using a Raspberry Pi for the project, you will need these additional items:

- [Powered USB hub](#)
- [Power supply](#) for the RPI (Raspberry PI)
- [MicroUSB cable](#) from charger to the Raspberry PI
- [SD card](#) 4+GB, and min class4 – That fits in the RPI
- [Case](#) for the raspberry (Optional, but highly recommended)

How to put it all together

I will be making it using a Raspberry PI in this guide, but it should work with a normal computer or laptop too.

I use Raspbian on it, which is Debian Linux optimized for speed on the Raspberry Pi.

Step 1 – Install the OS on your raspberry

To install Raspian on your raspberry you have to do the following. (If you already have a running Raspberry, then you can skip to step 2):

- Download the zip file containing the dd image from [Raspberrypi.org](#) (A little under 500MB)
- Extract the zip file to your hard drive, giving you the dd image “2013-07-26-wheezy-raspbian.img”
- Write this image to the target SD card.

Linux

1. Replacing sdX with the location of the SD card, run: dd bs=1M if=/path/to/2013-07-26-wheezy-raspbian.img of=/dev/sdX

Windows

1. Download and install [Win32DiskImager](#)
2. Select the 2013-07-26-wheezy-raspbian.img image file, select your SD card drive letter, and click Write

- Eject the card from your computer, insert into the Raspberry Pi, and power it on.
- If your keyboard, mouse, or other USB device doesn't appear to be working properly, try using it through a POWERED USB hub. The Raspberry Pi's USB ports are limited to 140mA. This limitation has been fixed in newer boards; however, you may still run into power issues.
- The default username is ‘pi’ with a password ‘raspberry’

Once you started up the device and is logged in, you will notice that there are no desktop. It's all console based. This entire guide will be done in the console only, and you should already know how to write commands and navigate around in the console to make this a lot more easy.

Step 2 – Install dependencies

When you have booted your Raspberry up, and it's running you can change the password and other stuff you like, but I am going to skip this part for now and do it later. You can find guides to all the things like changing

the keyboard layout, changing the timezone or language if you just google it. Maybe I will make small guides for each and every one, but for now I will skip it since it does not matter in this guide or device.

Now you need to install some dependencies that is needed to make the temperature reading software run. Since i use Raspbian in this guide, the program used to install software from the repositories is called “apt-get”. Run the following command to install all the needed software (they are space separated):

```
apt-get install libusb-dev libusb-1.0-0-dev
```

And press yes to everything it asks.

When done, you have to set some system variables, running the two following commands:

```
export LIBUSB_LIBDIR=/usr/lib/arm-linux-gnueabihf  
export LIBUSB_INCDIR=/usr/include/libusb-1.0
```

Step 3 – compile the program and get the temperature

You now have to download, compile and try out the software you have to use for reading the temperature of your TEMPer device. follow the step below.

1. Download the zip from here: [temperv14](#)

This can be done using the following command on your Raspberry:

```
wget http://dev-random.net/wp-content/uploads/2016/01/temperv14.zip
```

2. Extract the zip file on your Raspberry, to do this you first have to install the Unzip program on your Raspberry using the following command

```
apt-get install unzip
```

When installed, you can unzip the archive using the following command

```
unzip temperv14.zip
```

3. Change directory to the new unzipped folder using the command

```
cd temperv14/
```

4. Now compile the program using the following command:

```
make
```

It should output something like what you see in the picture below

```
[ss@laptop temperv14]$ make  
cc -DUNIT_TEST -o temperv14 temperv14.c -lusb  
[ss@laptop temperv14]$
```

5. Test the newly compiled program using this command

```
./temperv14
```

If you have your TEMPer device connected, it should output a timestamp and a temperature, if not it will output an error that the device was not found.

6. Move the newly compiled program to a special directory so that can be accessed from everywhere in your system using the command

```
mv temperv14 /usr/bin/
```

Now if you type “temperv14” anywhere in your system, it will run the program and output the temperature!

If you want to output only the temperature in Celsius (handy for implementing this in programs or monitoring systems like Nagios), you can run the command

```
temperv14 -c
```

Or for the temperature in Fahrenheit only

```
temperv14 -f
```

Step 4 - Manually adjust the thermometer

Roland (In comments) made a version that accepts arguments for subtraction and addition, so this step is no longer needed.

Package has been reuploaded with new source

Some TEMPer devices show the temperature a little off, like mine for example, it show 6 degrees Celsius too much, every single time!! This was an issue for me since it was not the correct temperature I got from the device.

I was just about to send it back, but then I decided I did not want to wait the 2-3 weeks for a new to arrive, so I just made the program subtract 6 degrees every time before outputting the temperature.

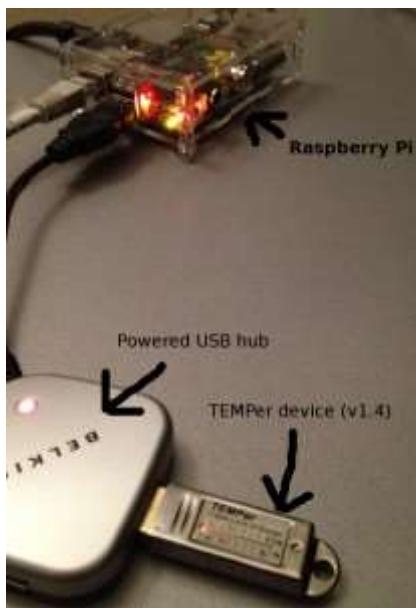
I made a static variable in the code of the program that you can change. so you should do the following

1. Find a manual temperature sensor, like from a fridge, and put it in the same room, right besides the TEMPer device, leave it there for an hour until it shows the correct temperature.
2. Run the command, and note how many degrees the output is at.
3. Compare it to the degrees on the manual thermometer.
4. If it's more, then do the following
 1. Edit the file temperv14.c with Nano or Vi
 2. Change line number 80 from "static int subtract = 0" to the number of degrees in CELSIUS the TEMPer device is showing too much.
 3. Save it, and repeat step 2 starting from substep number 4 (Where you run "make")

Done

Done, that's it.. you now are able to monitor your temperature remote using SSH, or by implementing this to Nagios or other Monitoring software just by running the command "temperv14" on the device.

You should know, that I had problems with my TEMPer device, when I did not attach it to a powered USB hub. when attached directly to the Raspberry it got hot, and showed way too much when running the command. so I recommend attaching it to a powered USB hub that is attached to the Raspberry's USB ports if this happens to you as well, it fixed the heating issue for me.



Share this:

This entry was posted in Debian, Guide, hardware, Linux, Nagios, Raspberry Pi, Windows and tagged Debian, guide, Hardware, how-to, Linux, network, Raspberry Pi, sensor, Temperature, thermometer, Windows on August 19, 2013 [<http://dev-random.net/temperature-measuring-using-linux-and-raspberry-pi/>] .

47 thoughts on “Temperature measuring using Linux and Raspberry Pi”

gigi

October 23, 2013 at 14:26

Thank you!!!

Steffan Post author

October 23, 2013 at 21:27

You are very welcome 😊

Feel free to subscribe to email notifications when i make new posts!

Norman

June 3, 2014 at 21:55

You should be able to use a USB extension cable instead of a powered hub. Much less expensive.

Steffan Post author

June 4, 2014 at 07:28

I don't think you have read the entire page..

i use a powered USB hub because the Raspberry can't deliver enough power to the TEMPer device, it somehow gets really hot if you plug it directly into the raspberry. so a USB extension cable here would just make things worse.

If you are using a normal computer, with lots of power in the usb port however, then yes. you can get the TEMPer device further away from the computer itself place it somewhere you can't put the computer using an USB extension cable.

Norman

June 4, 2014 at 20:17

I did read the entire page. Consider editing it so this point is clear.

From the article, it can be construed that the problem was heat from the Pi causes temperature readings to be high due to proximity. Now it seems you're saying the main problem is the Pi overheats due to power drawn by the device. For the former case an extension cable would be a solution. For the latter, a powered hub is required.

By the way, how much power does the TEMPer pull? That information is listed in the device's USB descriptors. You can get it using the "lsusb -v" command.

Steffan Post author

June 5, 2014 at 12:27

Nope. I think you have totally misunderstood what i mean.

The TEMPPer device itself get's hot if connected to the pi directly. The pi keeps its temperature as it should. I tried unplugging the TEMPPer device multiple times but every time i plugged it into the pi directly it (the TEMPPer device) turned hot.

As soon as i plugged it into a powered USB hub, the TEMPPer device did not get hot.

I'm sure if you plug it into a normal computer, and not a pi. It won't get hot and the usb hub will not be needed.

Shawn

June 5, 2014 at 02:15

Perhaps, when the TEMPPer is connected directly to the RaspPi, it gets hot because the TEMPPer is acting as heat-sink for the RaspPi. If so, a long USB extension would solve the problem.

What is the amperage rating of your power supply? If you're using a 1.0 or 1.5 amp power supply, the RaspPi should easily be able to power the TEMPPer since 500 mA is the maximum for any bus powered device.

John Halford

November 26, 2014 at 09:11

For some time I have had a TEMPPer1 unit for some time to play with on my windows notebook.

I works well there on their latest software I downloaded.

I have one with a sensor on the end of a lead plugged into a powered USB hub.

Getting my Raspberry Pi B up and running I thought I would give it a go there

I am running Raspian from the latest version of NOOBS. I am also running SSH from my Notebook as I am able to cut and paste from the web page into the SSH window (I discovered right click pastes the instruction).

Following the instructions at <http://dev-random.net/temperature-measuring-using-linux-and-raspberry-pi/> Some instructions seemed not to have permission so I prefixed them with sudo if they failed at first but I tried to muddle through.

I now get

1.

pi@raspberrypi ~/temperv14 \$ temperv14

Could not set configuration 1

or 2.

```
pi@raspberrypi ~/temperv14 $ temperv14
```

Couldn't find the USB device, Exiting

[I am mildly confused as the hub has on/off switches that do not seem to always work the same way, 1 with the switch seemingly off and 2 for the switch seemingly on]

It may be because I have the early version of the hardware – any suggestions

Steffan Post author

November 26, 2014 at 09:21

You need to run all commands as root (or with sudo in front) including the "./temperv14" because you need more privileges than the "pi" user has to read from the usb device.

There might be some permission tweaks you could make to allow the pi user to read from the device, but i never needed it since i just run it with sudo (in /etc/sudoers you can specify a command that pi can run with sudo without having to type the password)

John Halford

November 26, 2014 at 09:27

Further reading and at <http://rpimc.com/discussion/4/ssh-to-raspbmc-default-username-and-password/p1> on switching to logging on as root and it works.

George

November 29, 2013 at 13:13

Hey!

Thanks a lot!

I am using Debian Squeeze on the RSPI but weren't able to get the program to work.

```
#~/temperv14 $ ./temperv14
```

Could not set configuration 1

I also checked using

```
# lsusb
```

and it seems like the pi found my TEMPPer (Bus 001 Device 005: ID 0c45:7401 Microdia).

I'm quite new to Linux and Raspberry Pi, so I might have overseen something.

Thanks for helping!

Steffan Post author

November 29, 2013 at 16:48

Hello, sorry for the late reply, was at work.

If you get the configuration error that you pasted in your comment, it is because you don't have enough rights to read/write to the Temper device.

Run the program as root, and everything should work perfectly.

In Debian, you can do this by typing:

```
sudo ./temperv14
```

and then your password.

George

November 30, 2013 at 02:31

Thanks a lot! It worked great!

crg

February 17, 2014 at 04:18

Worked perfectly on raspbian.

To fix the permissions problem (so everyone can read temps off the USB) you need to:

```
$ lsusb | grep -i Microdia  
Bus 001 Device 010: ID 0c45:7401 Microdia  
$ chmod a+r /dev/bus/usb/001/010
```

David

May 31, 2014 at 05:01

Hello all,

I am running into a similar problem. I noticed that after I run the program 15 times, it throws the same "Could not set configuration 1" error because the device number increments by one. I have to fix the permissions each time. Any thoughts/suggestions?

I am trying to write a script to poll the device for temp regularly, so this is an annoying problem. If it helps, I am prototyping in Ubuntu for an eventual pi device.

Thanks for the handy program and instructions.

Steffan Post author

June 4, 2014 at 07:32

Are you running your script as root ?

I do, and have no problems with "configuration 1" error, I have only seen those appear if you do not have enough permissions

Tom Stephenson

March 14, 2014 at 21:00

Thanks so much for your excellent instruction. I bought one of these TEMPer devices to see if I could use it at work for thermal monitoring in a room full of hardware. I have 60 chassis running in a unattended room with 4 servers controlling hardware testing in each of the chassis. I wish to monitor thermals in each of the chassis. I think I can do that very well to detect failing fans etc in any of the chassis.

The instructions you provided were professional quality. Very well done.

Thanks,
Tom Stephenson

Lauri
April 23, 2014 at 16:54

Thanks a LOT for great instructions. Got this working on my debian server in no time ☺

Can't wait to graph with RRD!

Pingback: [Temperature measuring using Linux and Raspberry...](#)

Holger Ernst
July 7, 2014 at 14:57

```
#!/bin/bash

# alarm temp (Centigrade, no decimal)
threshold=25

# email addresses to send alerts to:
alert="reporting@someone.com another@someone.com"
mailheader="Temperature alert on `hostname`!"

# logfile
log="/var/log/temp.log"

# Here we go:

# Get the full reading
# On my test system sometimes i get an error:
# "Resource temporarily unavailable"
# So i check and try again
OK="no"
```

```

while [ ! "$OK" = "yes" ]
do
result=`./temperv14` && OK="yes"
done
echo $result >> $log

# Just the temperature reading
OK="no"
while [ ! "$OK" = "yes" ]
do
temp=`./temperv14 -c` && OK="yes"
done

# change temp to type integer
i=`echo "($temp)/1" | bc`

# Test and send emails
if [ $i -gt $threshold ]; then
for dest in $alert
do
# Make sure your mail agent is working propperly.
echo $result | mail -s "$mailheader" $dest
done
fi

```

chris

August 11, 2014 at 12:04

First of all: thank you for coding 😊

I ran this on Debian Wheezy 64bit with kernel 3.10 and get _one_ measurement right and any subsequent ones fail (at least until the device is re-plugged):

```

USB interrupt read: Resource temporarily unavailable
Fatal error> USB read failed

```

any idea?

Steffan Post author

August 11, 2014 at 12:28

I have never seen that error before..

It might be your Temper device that's broken.

I run mine of a raspberry pi with Rasbian using kernel 3.10.25

If I was your, I would try on another install of Debian, maybe an older version just because you can.

anon

May 19, 2015 at 05:50

I had the same issue.

So, I used the following code to reset the device:

<http://askubuntu.com/questions/645/how-do-you-reset-a-usb-device-from-the-command-line>

```
./usbreset /dev/bus/usb/001/068 ; ./temperv14
```

Resetting USB device /dev/bus/usb/001/068

Reset successful

2015/05/19 00:47:39 Temperature 78.98F 26.10C

Chris R.

June 3, 2015 at 13:14

This may or may not be helpful to anyone, but I came up with a script to use the two scripts (because scripts like scripts) for those of you having issues with

```
USB interrupt read: Resource temporarily unavailable
```

```
Fatal error> USB read failed
```

This script assumes that you have the temperv14 and the usbreset binary (from the above comment and link) in /usr/bin.

```
#!/bin/bash
```

```
DEVICE=$(echo /dev/bus/usb/`lsusb | grep -i Microdia | sed -e 's/\//g' -e 's/\//g' -e 's/\:\.*$//' | awk '{\$1 = sprintf("%03d", \$1); print}'` | tr " " "/")

/usr/bin/temperv14 #add whatever options you want such as -c

/usr/bin/usbresest $DEVICE
```

This script will parse the output from lsusb into the format that the usbresest binary wants (/dev/bus/usb/XXX/XXX). I have tested this on a server running Debian Jessie and it works just fine. I wouldn't suggest it for use in a script that is called often (more than once every minute) because of the constant resetting of the usb device.

To the author: It would seem that for some reason the usb_release_interface call is not working on full-version Debian (read: those of us using this with a full Debian install as opposed to Raspbian) systems for some (myself included). I think it comes from the fact that the function usb_detach_kernel_driver_np has no usb_attach_kernel_driver_np, unless you get a patched version of libusb that has it. That might be the issue.

anon

December 29, 2015 at 20:59

Within about 15 mins of starting to look at my Temper "Caps Lock to Excel" thermometer (lsusb: ID 0c45:7401 Microdia) I had things going great, and that included the time looking for a USB extension cable.

Dunno what was on the CD that came with the Temper, but I'll take source code I can compile any day! Thanks for the work.

I have had the error too, but only once and so far it hasn't returned:

```
$ ./temperv14
USB interrupt read: Numerical result out of range
Fatal error> USB read failed
```

The USB reset code hasn't been needed. This all so far has been on my Slackware64 laptop. But I plan to run the gadget attached to a PPC Debian Jessie box, a first gen Mac Mini, so who knows if this error will crop up again?

This error happened whilst I am trying to push the temperature about to see how the Temper aligns with another thermometer, and so maybe if temperatures are moving fast then it causes the error? I am currently chilling the Temper and the other thermometer's probe with non-direct contact with icy water.

I was going to ask about having to run the program as root, but the udev rules should deal with it once installed. I think the file 99-tempsensor.rules is corrupt in the zip file though.

```
$ file 99-tempsensor.rules  
99-tempsensor.rules: data
```

From a copy of pcsensor-0.0.1.tgz the udev file is plain text, as I expected, and contains:

```
SUBSYSTEMS=="usb", ACTION=="add", ATTRS{idVendor}=="0c45", ATTRS{idProduct}=="7401", MODE="666"
```

Not actually used it yet, but it looks good.

Steffan Post author

January 6, 2016 at 10:59

Thank you for pointing out that the 99-tempsensor.rules file was corrupted. I have replaced it and reuploaded the .zip file with a fixed one containing the line you pasted in your comment.

Keith
August 16, 2014 at 03:29

Followed your instructions and it worked. I seem to get some errors from the program every once in a while, so I am tracking that down.

George
September 15, 2014 at 02:35

When I connect the USB device it always works perfectly the first time. However, the second time I run the script I get:

```
$ sudo ./temperv14 -c  
Fatal error> USB read failed
```

Do you know what the problem might be?

Lauri

October 16, 2014 at 19:50

I'm getting weird values when temperature is below 0 celsius.

Something like this:

2014/10/16 20:48:13 Temperature 491.00F 255.00C

Any idea why it behaves like this?

Ed Hunsinger

December 17, 2014 at 05:19

I'm also seeing super high values:

2014/12/16 19:15:13 Temperature 492.79F 256.00C

Obviously it's not that hot or else I'd be dead.

Ed

December 17, 2014 at 06:27

Unplug and replugged it in and works fine now.

Manuele

November 9, 2014 at 18:49

[15127.934352] usb 1-1.3: Product: TEMPPerV1.4

[15127.934363] usb 1-1.3: Manufacturer: RDing

```
[15127.947134] input: RDing TEMPPerV1.4 as /devices/platform/bcm2708_usb/usb1/1-1/1-1.3/1-1.3:1.0/0003:0C45:7401.0003/input/input1  
[15127.948890] hid-generic 0003:0C45:7401.0003: input,hidraw0: USB HID v1.10 Key  
board [RDing TEMPPerV1.4] on usb-bcm2708_usb-1.3/input0  
[15127.963058] hid-generic 0003:0C45:7401.0004: hiddev0,hidraw1: USB HID v1.10 D  
evice [RDing TEMPPerV1.4] on usb-bcm2708_usb-1.3/input1
```

```
xbian@xbian ~/temperv14 $ make  
cc -DUNIT_TEST -o temperv14 temperv14.c -lusb  
make: cc: Command not found  
make: *** [temperv14] Error 127
```

Any ideas??

Manuele

November 9, 2014 at 19:25

apt-get install gcc and make worked like a charm!

Andy Ormsby

January 7, 2015 at 14:37

Worked perfectly, just as described. I went from powering up the Raspberry Pi to having this working in about 5 minutes. Thanks!

Mattijs Vantorre

February 9, 2015 at 22:05

Hey

It works perfect. I want to get the result of the command 'temperv14 -c' in a variable through a script.
Somebody how can help me.

Thanks

Steffan Post author

March 3, 2015 at 10:30

Bash? VAR=`temperv14 -c` or VAR=\$(temperv14 -c)
\$VAR will then contain the returned temperature

php? exec("temperv15 -c", \$output, \$returnvar);
\$output will then contain the returned temperature

Derek

April 1, 2015 at 18:44

Hi, just curious as to why you didn't plug the USB extender/hub into your server and serve the temperature from one of those machines instead of adding a Pi to the mix.

Steffan Post author

April 2, 2015 at 10:20

Because I am one of those who likes to have a separate server for each service. Example, a webserserver is for serving webpages, not hosting the sql too, that's what the sql server does.
and because a raspberry is easy to move around. Also my other servers are running ESXi or freenas, which is a bitch to install stuff on, don't really want to bother with it (hard to do an upgrade in the future if i do system changes on those)

Dick Quaif

April 13, 2015 at 22:53

Very nice code. Thanks.

I solved the rights problem by having temperv14 run as root and allow anyone to run it. That seemed the simplest solution.

```
sudo chown root temperv14  
sudo chmod +s temperv14
```

On the Raspberry PI 2, you can set the max USB current up to 1.2 amps by adding max_usb_current=1 to the Config.txt file in /boot. (Provided your power supply will handle it).

It have no trouble with the TEMPPer I have when connected to the PI 2.

I suspect it is a voltage problem you are having and not a current problem. The older Raspberry PI B I have, has a rather low VBus. I boosted the power supply a bit to 5.2 Volts to compensate. That makes other things work better although I have not tried the TEMPPer on it.

Keep up the good work and thanks.

Jussi

April 28, 2015 at 20:46

Thanks for tips, it was very easy to get it up and running! My sensor was accurate so adjusting was not necessary.

However, I was able to find one bug. I'm not sure if it's temperv14 program or sensor itself, but below-zero temperatures are reported as values starting from 256 C. E.g. 0 C is 256 C, -1 C is 255 C etc. I know most of you don't care about this, but I purchased this sensor to make sure I don't run my Raspberry with batteries in temperatures below -20 C as this is considered dangerous (my raspbi is located in a car and we are used to quite cold winters).

Fortunately it is easy to calculate correct figures as I don't expect over 150 C temperatures, but I just wanted to let you know. ☺

Darren

October 31, 2015 at 12:28

Hi

If I have 2 more sensor.

How to get their temperature ?

Richard Goodwin

November 13, 2015 at 10:04

Hi, thanks for the code – works great on a Pi.

I changed the following line:

static int substract = 0;

to:

static float substract = 1.38;

So that I could adjust the difference between what the probe was measuring and the reading from a digital thermometer with more accuracy (my diff was 1.38C)

Richard

Richard Lippmann

January 15, 2016 at 17:26

Great, I hate to buy network-temperatursensors for 150 Euros. I compiled a lot of binaries like temperv14_minus1, temperv14_minus2, etc. I would love to see an extra parameter to subtract from measured temperature like this:

./temperv14 -subtract-degrees 7

Roland

March 8, 2016 at 17:06

I modified the program to read more than one sensor – if anyone is interested.

I also added support for a -s argument (subtract degrees):

./temperv14 -h

pcsensor version 0.0.1

Aviable options:

-h help
-v verbose
-s[n] subtract 'n' degrees Celsius
-c output only in Celsius
-f output only in Fahrenheit
-m output for mrtg integration
-d[n] show only device 'n'

Zach

March 24, 2016 at 10:26

Fantastic...thanks so much! I'm a noob, but this works flawlessly. Now I just need to figure out how to make it log values and occasionally upload them to some sort of graphing service. Anyone recommend a guide?

Zach

March 25, 2016 at 09:02

I just noticed a bug here with below zero temperatures. This site explains how to fix ->

<http://lee.smallbone.com/2014/10/using-temperv1f-usb-temperature-probe-on-raspberry-pi/>

Bernhard Fischer-Wasel

May 14, 2016 at 20:46

I am running Raspi P3 and XMBC...

I like to integrate Temper1 USB Sensor...

when initiating "make" I get the following error:

```
root@raspbmc:~/temperv14# make
cc -DUNIT_TEST -o temperv14 temperv14.c -lusb
make: cc: Command not found
make: *** [temperv14] Error 127
```

I had done apt-get install gcc already...

Any idea ?

TIA
