Temperature sensor for WRT54G 2.x, WRT54GS, WL-500gx with DS1820

Introduction

This is a tutorial for attaching Dallas DS1820, DS18B20, DS18S20, 1-wire thermometer to Linksys WRT54G/S. The motivation was due to frustration of adapter stacking found on internet. Even some authoritative open source such as OWFS suggest DS2480B line driver, 3.3V -> 5V MAX883 level shifter, voltage regulator and so on.

It can be shown that for simple temperature measurement with WRT54G only sensor and transistor with three resistors is sufficient for the job. No PCB, just soldering on SIL socket. With testing it was found out that DS1820 cannot work with parasite power on 3.3V. Additional wire is needed to power DS1820 sensors during temperature conversion.

WRT54G/S serial interface

For opening the case and how to void warranty please see Linksys WRT54G/WRT54GS Dual Serial Port Mod. Here is excerpt of the basic info:

The pin-out of JP1 on the Linksys PCB is as follows:

```
Pin  1: 3.3V         Pin  2: 3.3V
Pin  3: Tx (ttyS1)   Pin  4: Tx (ttyS0)
Pin  5: Rx (ttyS1)   Pin  6: Rx (ttyS0)
Pin  7: NC           Pin  8: NC
Pin  9: GND          Pin 10: GND
```

We will concentrate on second port *ttsS1* as first is reserved for serial console.

Measured BCM4712 serial port DC characteristics

As no data was available for Broadcom BCM4712 processor (didn't bother to search for it), the following measurements were obtained:

- TX source current with 4k7 resistor shows now voltage drop at 0.7mA. This means that driver is quite strong. We will not go into Max ratings. Sink current was not measured as this assumed that it is at least equally strong.
- RX sink current measured with 100k resistor shows drop to 2.47V. This means 25 µA source current for weak pull-up. This is equivalent to at least 35k resistor. Pull-up is quite strong. To achieve less than 0.8V at input the input protection resistor R2 should be less than 32k ohm.

Testing serial interface /dev/tts/1

Initial testing should be done without any 1-wire device connected. Open one telnet console and enter port capture command like this:

```
cat < /dev/tts/1
```

On the second telnet console enter the following command:

```
echo 0123456789 > /dev/tts/1
```

You should see the same echo on first console. If this is not the case then you are doing something wrong! If you see garbage like ðõõþàà then try lower baud rate with the command

```
stty -F /dev/tts/1 9600
```

.

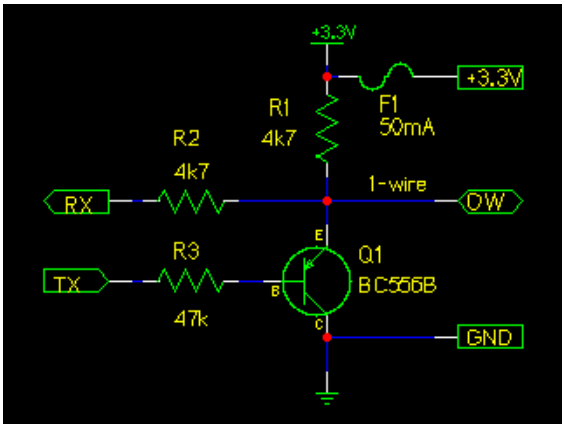
If all is OK, then you should receive what was send also at 115200 baud. As last resort you should short circuit RX and TX pins and verify with the above commands, that serial port is working. If it works at 9600 and not at 115200 then try to change pull-up resistor or change transistor.

For continuous watching with oscilloscope i suggest

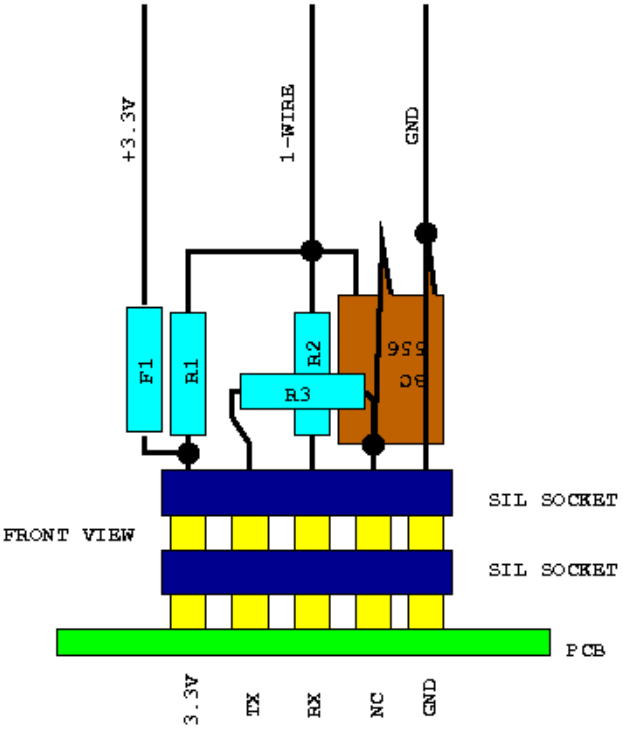
```
while 1; do echo 0123456789 > /dev/tts/1; done
```

With `Ctrl-C` for end.

Suggested circuit for driving DS1820 thermometers

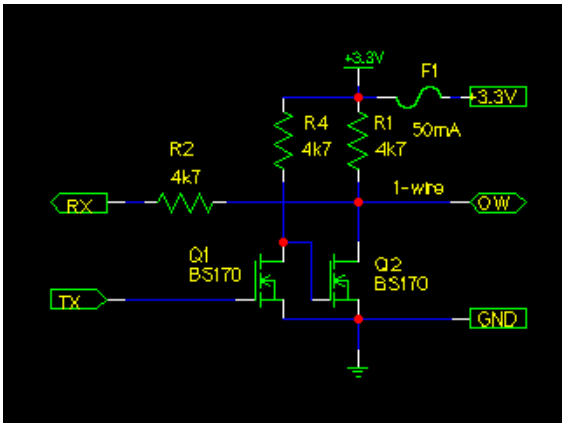


This is 1-wire driver with minimal component count. Widely available components are used. One general purpose PNP transistor, there resistors and one DIL socket as connector and component holder. NO PCB is required. Resistors R2 and R3 are inserted for current limiting protection of the processor. Fuse F1 is to prevent possible short circuit. The circuit should work also without R2 and R3 giving only R1 and Q1. Two closeup photos are available: small resistors and wiring through case .



Despite the simplicity of the above 1-wire driver, one can run into troubles. The basic problem of this driver is that the LOW level is intrinsically near the specs of DS1820 and also processor RX pin. In other words; due to V_EB diode, the low output level will be approx 0.7V which is not so good.

Better circuit shown below should meet most 1-wire signal requirements (except transient). It has one transistor more and NMOS technology! NMOS circuit is recommended as I've found out that with single PNP LOW level is likely to fail although mine works OK.



When selecting NMOS transistor try to find one with low Gate-Source Threshold Voltage V_GS. For example 2N7002. Despite that I've had success with general purpose BS170 with V_GS=(min 1.0, typ 2, max 3.0) V. For this application integrated solution such as Fairchild NC7WZ07 is not the best solution for home-alone practitioner as suggested by Application Note 214: Using a UART to Implement a 1-Wire Bus Master

When running 1-wire bus on a long run (10m or more), add protection shotky diodes as suggested in various maxim-ic app notes.

Compiling digitemp for WRT54G

Download appropriate Linksys GPL software with the same codebase version numbers as your firmware in WRT54G/S. Move compiler tools into `/opt/brcm` directory. Digitemp Makefile should be modified only with cross compiler and include directories, no locking eg:

```
CC      = mipsel-uclibc-gcc
CFLAGS  = -I/tmp/WRT54G_3_01_3_0922/release/src/linux/linux -I./src -I./userial -O2 # -g -Wall
LOCK = no
```

In case of shared files mismatch add `-static` to LIBS:

```
LIBS = -lm -static
```

Fetching compiled binary

From a working ftp server example:

```
ftpget -u leon -p password 192.168.1.22 digitemp /tmp/digitemp-3.3.2/digitemp_DS9097
```

Alternatively one can use secure copy `scp` if firmare allows **sshd** connections to WRT. Use `chmod +x digitemp` for setting executable bit.

Reading temperature

```
~ # ./digitemp -s /dev/tts/1 -i
DigiTemp v3.3.2 Copyright 1996-2004 by Brian C. Lane
GNU Public License v2.0 - http://www.brianlane.com
Turning off all DS2409 Couplers
.
Searching the 1-Wire LAN
1074490900000036 : DS1820/DS18S20/DS1920 Temperature Sensor
ROM #0 : 1074490900000036
Wrote .digitemprc
```

Reading sensor:

```
~ # ./digitemp -a -q
Jan 01 00:08:23 Sensor 0 C: 18.23 F: 64.81
```

Logging temperature in a file for a whole day

```
~ # ./digitemp -l /tmp/temperature.log -d 60 -q -n 1440 -a &
```

For checking current temperatures one can issue

```
tail -f /tmp/temperature.log
```

Setting .digitemprc

One can change log format for better space consumption. For running with 20 bytes per line/minute this means 200k per week. Free RAM on GS is typically 18MB so this suffices for two years if running without reboot.

```
TTY /dev/tts/1
READ_TIME 1000
LOG_TYPE 1
LOG_FORMAT "%b %d %H:%M %.2C"
CNT_FORMAT "%b %d %H:%M:%S Sensor %s #%n %C"
HUM_FORMAT "%b %d %H:%M:%S Sensor %s C: %.2C F: %.2F H: %h%%"
SENSORS 1
ROM 0 0x28 0x54 0x6B 0x51 0x00 0x00 0x00 0xDA
```

Another way which is also suitable for `rc_startup` script in NVRAM is to fetch the binaries with WGET from your server via HTTP or FTP. For setting `rc_startup` variable use the following commands:

```
  nvram  set rc_startup='#!/bin/sh
  mkdir /tmp/www
  wget http://your-server-ip/wrt54ow/digitemp_DS9097 -O /tmp/root/digitemp
  wget http://your-server-ip/wrt54ow/temp.asp -O /www/user/temp.asp
  cd /tmp/root
  ./digitemp -s /dev/tts/1
  ./digitemp -l /tmp/temperature.log -d 60 -q -n 0 -a &'
```

Do not forget `nvram commit` for permanent store.

Setting date

This example can be helpful if NTP client is not running on machine. `date -s 022822302005`

Web interface

Minimalist HTML page which can be help you bring temperature to desktop every minute is created by the following shell script (dtrun.sh):

```
mkdir /tmp/www
OUT=/www/user/temp.asp
while true ; do
echo "<html><head><META HTTP-EQUIV=Refresh CONTENT=61></head><body>" > $OUT
./digitemp -o "%H:%M %0.2C&deg;C" -a -q > $OUT
echo "</body></html>" > $OUT
sleep 60
done
```

To run this script in background use `sh dtrun.sh &`. ASP page can be opened with `http://192.168.1.20/user/temp.asp`. The page is auto reloaded every minute. I popup window should be opened you can use the following HTML code (eg. `/www/user/index.asp`) which will help open popup window. Most browsers block this type of window now-days. User should grace opening popups from this site.

```
<body onLoad="window.open('temp.asp', null, 'width=150,height=50,status=no,nocation=no,menubar=no');">
<h1>User page</h1>
```

Other uses of the measurement data are bound only by user needs. Graphing, etc. Probably the simplest way to post temperature sampling is with wget GET method:

```
./digitemp -a -q -o"%4.1C" -n 0 -d 300 | while true; do
        read temp;
        wget -q "http://www.myserver.net/temperatura.php?$temp" -O /dev/null
done
```

Then the server with temperatura.php receives and stores reading into javascript file `temperatura.js` which can be included into html:

```
<?php
if ($argc == 1)
```

```
{
        echo "Temperature upload  succesfull\n";
        $prejeto = date("G:i");
        $f=fopen("temperatura.js","w");
        fputs($f, "var prejeto = '" . $prejeto . "';\n");
        fputs($f, "var temperatura = " . $argv[0] . ";\n");
        fclose($f);
}
else
{
   echo "Invalid UPLOAD request\n";
}
?>
```

It should be noted that digitemp does not have line buffering for `stdout` and that source should be patched with `setlinebuf(stdout);` for pipelining to work as espected!

Download

Here is digitemp binary precompiled for running on Alchemy-6.0-RC6a v3.01.3.8sv or similar firmware. Static build is also provided for different firmware version! As you noticed so far, all examples are described when running above firmware. Here is ZIP file with binaries and scripts. For DD-WRT v23 i use this digitemp_DS9097-dd binary.

Conclusion

Another application which came to my mind is to have two thermometers. One for external temperature and another for WiFi chipset temperature.

One may wonder how to include digitemp/www software in firmware. For firmwares with open source code such as HyperWRT, OpenWRT or eWRT this is not a problem. Unfortunately Sveasoft tries to overcome GPL licence with limiting/re-licencing source code distribution. This means that you cannot build your own firmware until the source, if ever, will be realeased.

As with all projects no Monte-Carlo analysis were performed. So no warranty except **void** is given. Although any suggestions are welcome.

Update

Now digitemp also runs on Asus WL-500gx and nslu2 architecture. Both USB and serial versions can be used. See Optware Digitemp package description for details.

Last modified: Fri Feb 10 09:14:53 CET 2006