Precision Diode Temperature Sensing using ADM1034

September 27, 2007

Contents

1 Introduction **1**
1.1 Advantages of Diode temperature sensors 2
Disadvantages 2

2 Demo Board Hardware **3**

3 Initialisation **3**

4 Temperature Format **3**
4.1 Reading with the I2C2PC Adaptor 3
4.2 Converting the Hex Data to Temperatures 4

5 Matlab Example **4**

6 Results **5**
6.1 Room Temperature Measurements 5
6.2 Interchanging Sensors 7
6.3 Noise 7
6.4 Self Heating and Internal Sensor 8
6.5 Summary of Results 8

7 Drawings **8**
7.1 Circuit Diagrams 9
7.2 PCB Drawings 10

1 Introduction

The Analog Devices `www.analog.com` ADM1034 temperature sensor and fan controller IC is an easy way to make an I2C based temperature measurement system using external diode sensors[1].

- 2 channels of external sensing using diodes + 1 internal sensor channel
- -64 - 191°C range
- 0.03C resolution.
- Internal averaging to improve noise. (0.1°C rms measured)
- 3 current level measurement technique to eliminate series resistance errors
- Internal comparators allow direct over/under temperture indication without any software.
- Simple circuit, small package.
- Specified at 3.3V, operates to 5.5V
- 4 hardware I2C Addresses: Up to 24 external sensor channels can be read directly from a single I2C2PC adaptor. (Use PCS9545 4 way switches to read 96)

Analog Devices (and others) make related devices. The ADT7466 is a similar device with an almost compatible footprint, but somewhat lower specs.

1.1 Advantages of Diode temperature sensors

We have compared different temperature sensing techniques elsewhere. The external diode temperature sensor has

- Most readily available sensor: any common bipolar transistor is suitable, available anywhere, any time.
- Huge choice of packages: From less than 1mm to packages with screw holes to bolt on.
- Lowest possible sensor cost: Sensors are so cheap they are disposable, and can be freely glued, soldered, cast in place
- Interchangablity is reasonable: No calibration needed for many jobs
- Reasonable temperature range: -64 - 191 °C
- Only a passive device on the sensor cable, no active, powered, digital devices: robust and simple to make.
- Negligible self heating of sensor (<10uW sensor heat) (compared to internal sensor IC's)

Disadvantages

- Capacitance limits cable length from IC to Diode to ~20m to 40m max using Cat5 cable.

[1] When we say diode sensors, we normally use a bipolar transistor with base and collector connected.

2 Demo Board Hardware

3 Initialisation

At power on the sensor will be automatically updating its temperature registers 8 times a second, with averaging enabled for noise reduction. The only setup required is to setup the block read size for the number of temperature values you want to read.

```
S A0 00 06 P //setup block read size (6 bytes) for temperature blocks.
SA0C0R06P //read the 3 temp sensor values
```

If you want to enable hardware threshold levels these must be initialised. There are two outputs that can be used for comparison: THRESH and ALERT/INT. THRESH can only be used to signal over temperature. In our demo board there is an LED on this. The ALERT output can be programmed with separate upper and lower limits. While normally used as an interrupt out to the bus, it can also just be used as a comparator output for an LED or alarm.

The chip also has a calibration offset register which can be set during initialisation.

The file InitADMx4.dat initialises 4 chips on a common I2C with addresses 0xA0, 0xA2, 0xA4, 0xA6

4 Temperature Format

The data is stored in 2 bytes, and is stored LSB first. It must be read LSB *then* MSB as there is a data lock mechanism to prevent updates between byte reads.

Data is most easily read in blocks. The ADM1034 has an non-intuitive block read mechanism.

- First you must write the number of bytes that will be read, into the block length register (reg 0x00). This only needs to be done when it changes (i.e. in the initialisation above)
- To do a block read, you must address the register with bit 7 set (so to block read reg 0x40 you address 0xC0)
- When you do a block read, the first byte returned will be the block length value. So you must read 1 byte more than the data length, and ignore this byte when processing the data
- Write the register address with bit 7 set, then do a repeated start and read the block of data.

4.1 Reading with the I2C2PC Adaptor

This assumes that you are familiar with the I2C2PC's simple I2C commands.

So to read 3 temperature value (6 bytes) from the device at I2C bus address 0xA0, you must read 7 bytes

```
S A0 C0 R 07 P .
```

If you only want to read the external diode sensors then just read 4 bytes, from register 0x42 (0xC2 with bit 7 set)

```
S A0 C2 R 05 P .
```

To read data from two chips (addr=0xA0,0xA2), with a comma delimiter between data.

```
S A0 C0 R 07 P , S A2 C0 R 07 P .
```

The read commands for 4 sensors are in Read3TempsADMx4.dat. Corresponding files for 1,2,3 sensors are also given.

4.2 Converting the Hex Data to Temperatures

The datasheet formula for calculating temperature is a little misleading. This following formula works.

Temperature = (MSB-64) + ((LSB and 248) / 256)

$$Temperature = (MSB - 64) + \left(\frac{LSB\&248}{256}\right)$$

Again, note that the LSB byte is first, and MSB second.

5 Matlab Example

This example will use our Realterm program to control the I2C2PC adaptor, and collect the data into a text file. The text file will be analysed and plotted with Matlab.

- Realterm dumps an initialisation file to configure the I2C2PC adaptor, and to configure the ADM1034. It does this once at the start only.
- Realterm opens a capture file.
- Realterm will add a timestamp to the start of each line, before the data. (Matlab time format chosen)
- Realterm goes into a loop. Every second it sends a file with commands to read the temperatures.
- The capture file can be read and processed while this goes on. But beware of using Microsoft programs (ie excel), that try to lock the files they access.

6 Results

Four ADM1034's were logged simultaeneously at room temperature, VCC=3.3V.

All were obtained from the same supplier. #1&2 were from adjacent postions on the tape, #3,4 came several weeks later, and were supplied loose. They likely came from the same reel.

All were mounted on separate pcbs, with 100uF of bypass capacitance. BC337 NPN transistors were used as sensors. These were taken randomly from the same order . The external sensors were connected with 0.5m of UTP cable. They were bundled together, with the reference thermistor, in foam. The pcbs (and therefore the internal sensors) were left open to air.

A thermometrics EC95, 10k, 0.2° accuracy thermistor was used as a reference.

The ADM1034's were running in default, ie continuous operation, averaging, reading all 3 channels in round-robin. The data was collected every second.

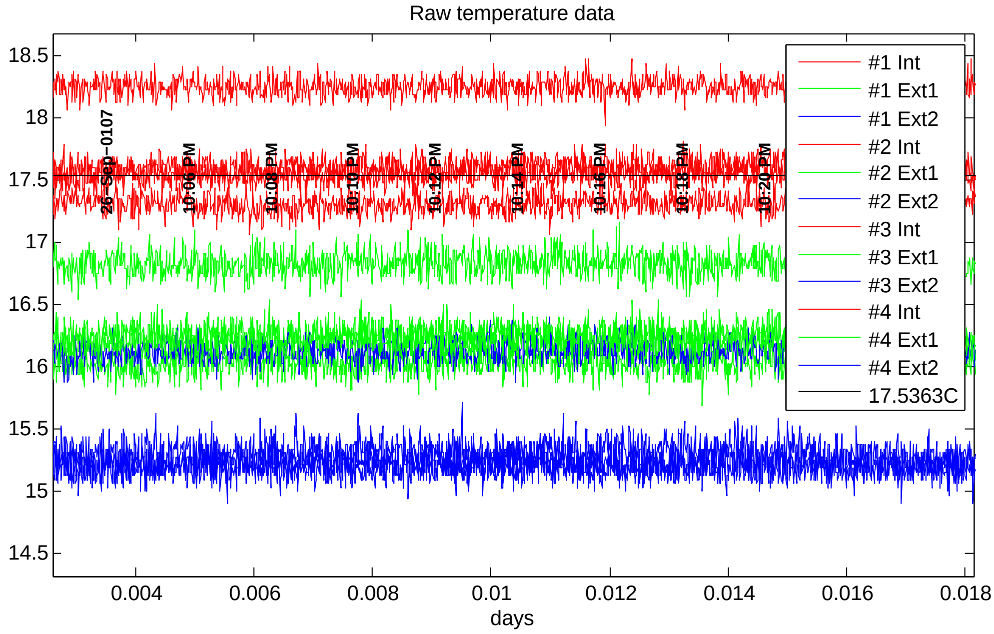
6.1 Room Temperature Measurements

Temperature during the measurement was around 17.5° by thermistor.

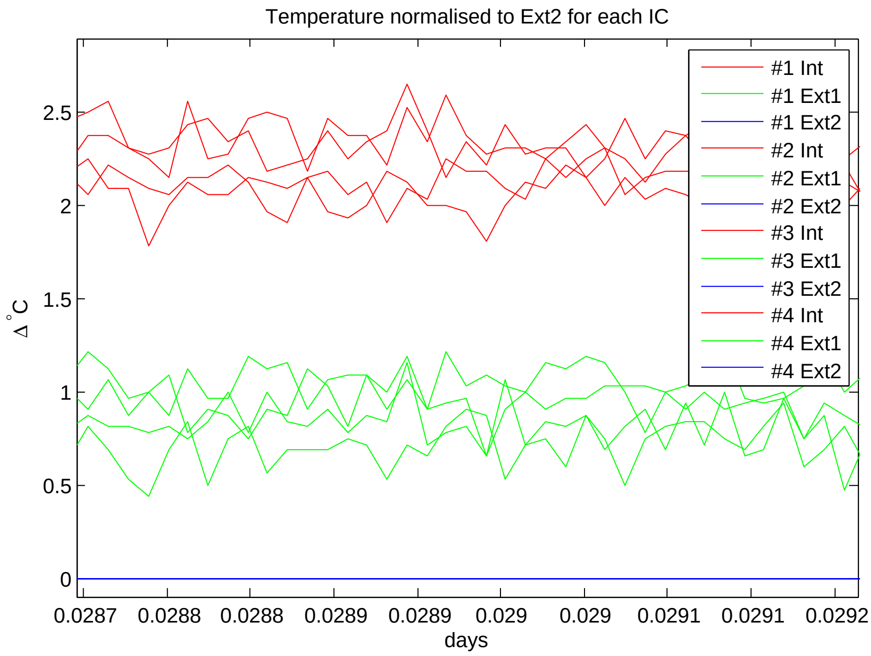
Immediately striking is the grouping of the corresponding channels on the different chips. This is especially so, since it appears that one of the chips is calibrated noticeably different to the other three.

Sensors were selectively heated, and disconnected to verify that the graph is displaying the expected sensors, and both chips.

Some of the difference between internal and external is expected due to self heating. However no systematic difference between the two external diodes is expected.



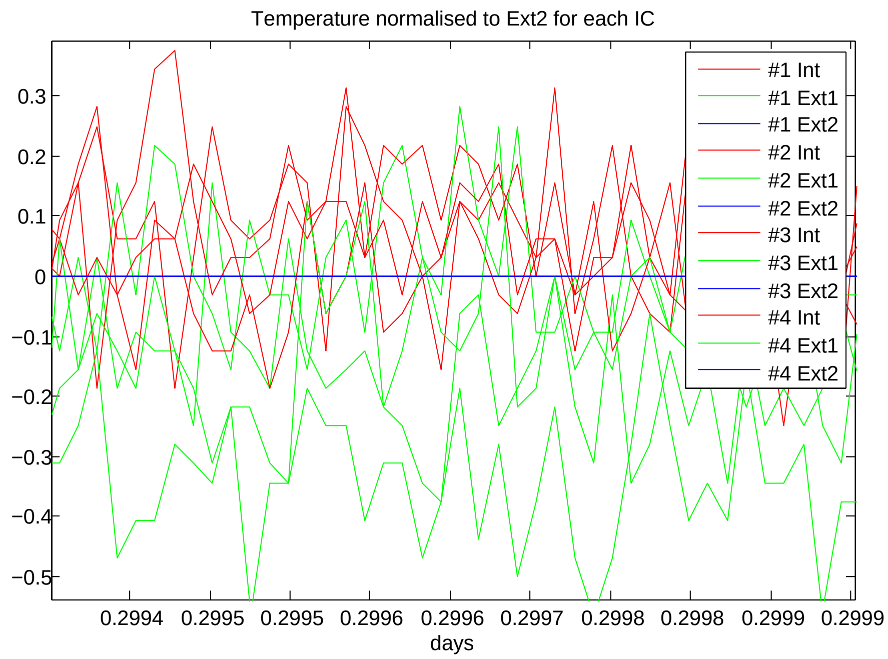
Below, we have normalised for each chips calibration error, by subtracting the Ext2 value for each IC. (So the blue Ext2 line is always 0). Here we see what appears to be a suspiciously systemic 1°C difference between the channels. Note that the Int sensor (red) is expected to be 0.35°C high due to self heating6.4



So if we normalise with both the calibration error and the postulated °C difference between channels we get the graph below.

The conclusion is that

$$ActualTemp = INT + 0C^{\circ} = EXT1 + 1^{\circ}C = EXT2 + 2^{\circ}C$$



6.2 Interchanging Sensors

The external sensors were swapped between IC's. ie swap(cyan,green); swap(magenta,blue). The effect was <0.2°.

<complete>

6.3 Noise

The *rms* noise at room temperature was calculated for the 3 channels of two sensors.

Noise	IC	Internal	External D1	External D2
°C rms	1	0.0686	0.1004	0.1088
°C rms	2	0.0675	0.0893	0.0958

6.4 Self Heating and Internal Sensor

The internal sensor will have systematically higher temperatures due to the dissipation within the IC. This is dependent on the thermal resistance of the pcb design. Under continuous operation we can make some estimates of the range we expect.

Max: At VCC 5V with rated current drain of 3mA = 15mW, and a TjA=150C/W, that dT=2.25°C.

Min: At VCC 3V with rated current drain of 3mA = 9mW, and a TjC=39C/W, that ΔT=0.35°C. Note that this error can be changed by altering the sample rate and averaging settings.

So far the experiments don't allow an estimate of the actual self heating error.

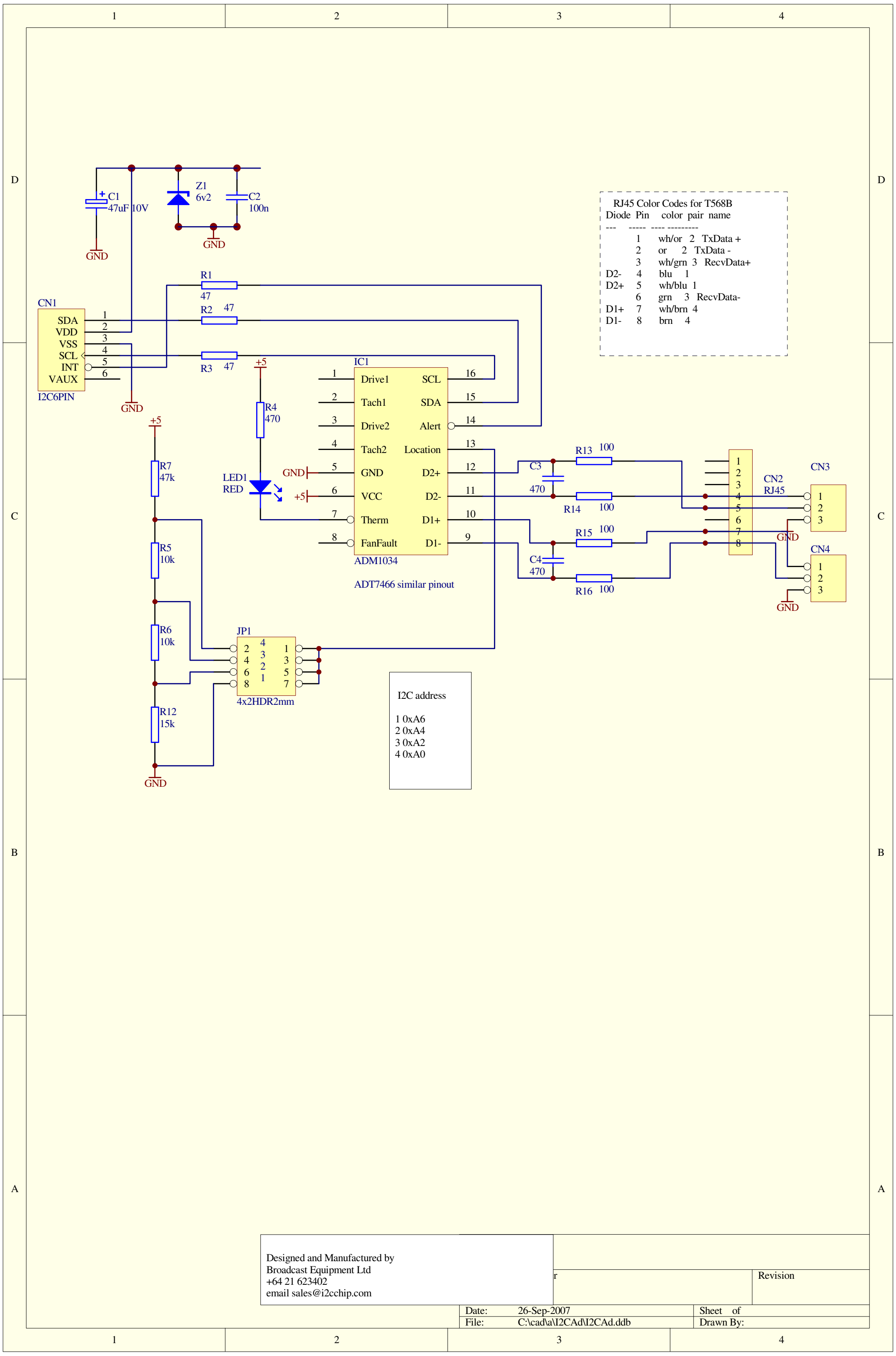
6.5 Summary of Results

Based on room temperature testing of 4 samples, likely from the same production lot, there appears to be a systematic error of 1° between sucessive channels in the same chip.

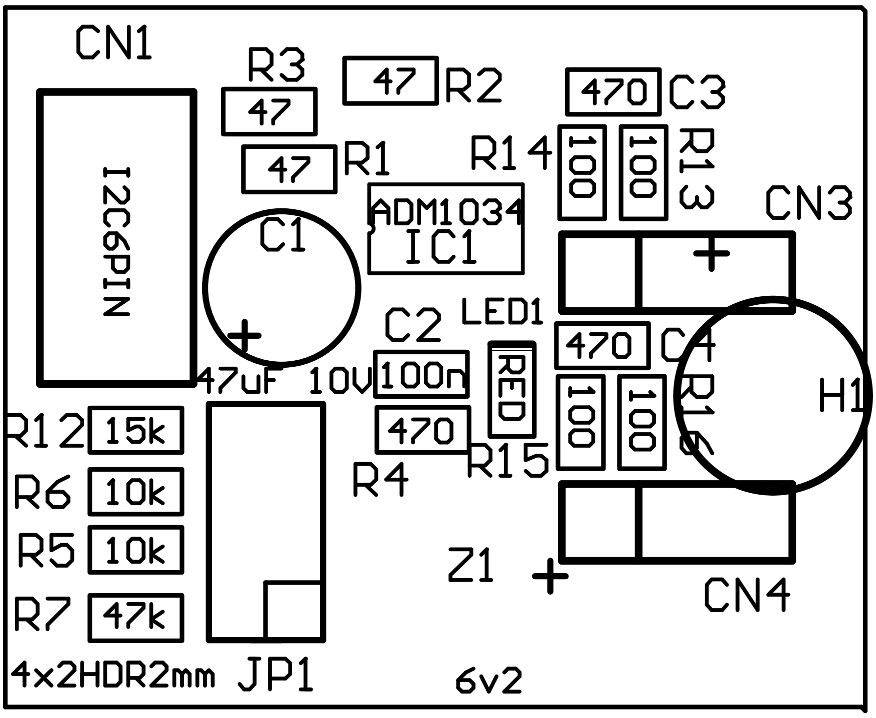
If this postulated error is removed, the correlation between sensors appears excellent. Noise performance is also good for integrated sensors

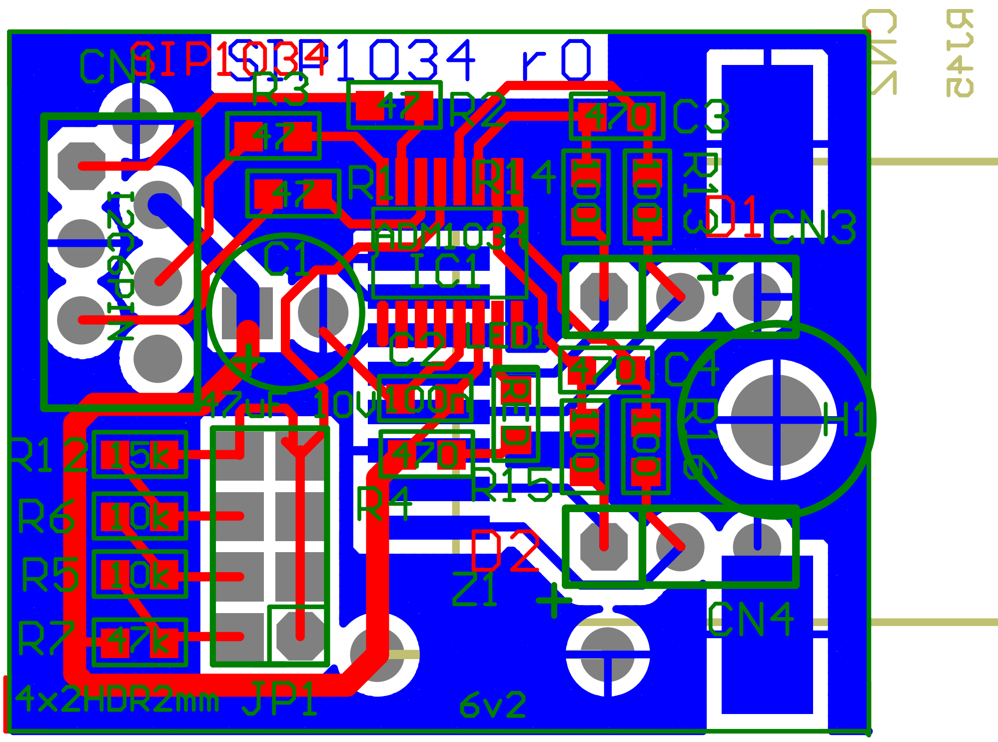
7 Drawings

7.1 Circuit Diagrams



7.2 PCB Drawings




SIP1034 r0
CN1
R3
R2
R1
R14
R13
C3
D1
CN3
ADM1034
IC1
C1
C2
LED1
C4
H1
47uF 10V
100n
R12
15k
R6
10k
R5
10k
R7
47k
R4
R15
R16
D2
Z1
CN4
4x2HDR2mm JP1
6v2