

# Temperature measurement on the Leighton Telescope: Surface Memo No. 2

7-Jan-2003 (Some typos corrected on 7-May-2003)

By: RACHamberlin, cham@submm.caltech.edu  
Caltech Submillimeter Observatory, Hilo, HI 96720

Location of memo source files:  
ulu//home/cham/ha5/ha5\_memo\_6may03.tex

## 1 Introduction

The CSO telescope surface figure may be adversely effected by temperature gradients. The nature, magnitude, and severity of the induced surface error depends on the type of gradient: e.g. front to back, radial, etc [1]. On a structure like the Leighton antenna, gradients of more than a few tenths of a degree have the potential to adversely affect the surface, and gradients of a degree or larger probably have a serious effect and more research is required into their cause and mitigation.

To characterize any possible gradients, obviously the temperature of the telescope structure should be measured accurately at many places simultaneously. To achieve this objective we have experimented with a set of eight DS18B20 temperature measurement probes in a one-wire network and found very encouraging results: they were very accurate, inexpensive, and it was easy to automate computer logging results from them. In principle networks of these sensors using a one-wire bus are very expandable. Under optimum conditions up to 200 DS18B20 probes on a bus 2,000 feet long can be supported by one bus controller.

Initial results from our small network of DS18B20 sensors has already produced some interesting results as we shall see later in this memo. The first part of the memo will describe the DS18B20 calibration and implementation. The second part will present some preliminary results produced from logging a few nights of data.

## 2 Hardware

The DS18B20 temperature measurement probe is based on a bandgap device (i.e. diode thermometry). Note that this detection method is unlike the

original DS1820 which obtained temperature indications by relying on a low temperature coefficient oscillator to gate counts from a high temperature coefficient oscillator. In controlled lab tests we compared the response from the original DS1820 device to the DS18S20 and the DS18B20. In the three DS1820 devices we tested, we occasionally observed quantized jumps in the indicated temperature. The jumps were about 0.3 degrees C in magnitude. No such jumps were observed temperature indications from the DS18S20 and DS18B20 probes. Like the DS18B20, the DS18S20 relies on bandgap temperature measurement.

The DS18B20 is packaged in a transistor-like TO-92 package with three leads: ground, data, and optional power. In the mode we are using the devices, power is derived from the data line and the optional power lead is grounded. All devices share a common ground and data line and each device is addressed by a unique 64 bit serial number. More details can be found on the data sheet: see <http://pdfserv.maxim-ic.com/arpdf/DS18B20.pdf>.

The one-wire bus controller is a PointSix HA5: see <http://www.pointsix.com/cgi-bin/PointSix.cgi?IHosts&HA5>. The HA5 bus controller we use has an RS232 interface. We have successfully interfaced the HA5 to several PC style computers running the QNX and Linux operating systems. For examples of Linux compatible interfacing code see Appendix 1.

The HA5 derives power for itself and the one-wire bus from the computer com port. In practice we found that PC com ports putting out power lower than about 6 volts caused the HA5 and devices on its bus to behave in a marginal manner. So if marginal behavior is observed, external power for the HA5 or and/or the one-wire bus may be indicated. In the current implementation we are acquiring data using one of the puuoo.submm.caltech.edu com ports and no marginal behavior is observed. The HA5 is situated on a short RS232 cable near the puuoo com port and seven DS18B20's share one twisted pair of a CAT-5 ethernet cable. (The eighth sensor was kept in the Hilo lab as a secondary calibration source.) The estimated CAT-5 cable length is 200 feet. Devices were dropped off the main cable with leads no longer than about 6 inches. The drop leads were made by soldering short lengths of wirewrap wire to the DS18B20 device leads. No effort was made to mitigate cable reflections, although it is noted that some network topologies require such measures: e.g. see

[http://www.maxim-ic.com/appnotes.cfm/appnote\\_number/857](http://www.maxim-ic.com/appnotes.cfm/appnote_number/857) and  
<http://www.pointsix.com/support/documents/app1820.pdf>.

The HA5 is also available with an RS485 interface. The RS485 standard

supports up to 26 devices on one serial line and RS232 to RS485 converters are easily available: e.g. see <http://www.telebyteusa.com/interfaceconverter.htm>. Thus, in principle one RS232 port on any computer could be used to interface up to 26 HA5's with each HA5 controlling up to 200 DS18B20's: that is, 5200 temperature probes can be read through one RS232 port.

The approximate cost of an HA5 is \$100, each DS18B20 costs about \$2.50, and an RS232 to RS485 converter costs about \$200. Because of its low cost, the DS18B20 technology looks very attractive for applications where many channels need to be logged, but is it accurate enough?

To establish the temperature accuracy of our DS18B20's in the Hilo laboratory we made comparisons with a NIST traceable thermometer described as follows.

## 2.1 Temperature calibration

A set of eight DS18B20's were electrically connected to a one-wire bus on a small bread board. The bread board was placed in several plastic bags and weighted. The weighted set of bags with the DS18B20's were placed in the reservoir of a Neslab temperature controlled bath (Neslab, Model RTE-110). The top of the set of bags was sealed as well as possible consistent with the one-wire bus cable needing to pass through. Care was taken to prevent the top seals of the bags from being immersed in bath solution. That is, the DS18B20's were kept dry during these measurements. The bath solution was approximately 50% water and 50% automotive anti-freeze.

The temperature of the bath solution was independently monitored with a NIST traceable calibrated mercury thermometer (Make: Miller & Weber; Type: P63C; Range: -8/32°C; Calibration date: 7/14/99 ). The thermometer minor gradations were in 0.10° and a correction table to 0.01°C accuracy was provided with the instrument.

Comparisons of the DS18B20's and the NIST thermometer were made at seven different temperatures between -7°C and 30°C. Figure 1 shows the calibration curve for one of the DS18B20 devices. The data in this curve were fit to a straight line. Over this narrow range of temperatures the response of all tested DS18B20's were quite linear. For the device represented in Figure 1, the zero intercept of the fit was 0.056°C and the slope was 0.987. At 0°C, the differences of all the DS18B20 devices from the NIST traceable with thermometer were within 0.2°C or less. Table 1 shows the results of calibrating all eight DS18B20 devices. The first column is the unique 64 bit

Table 1: Results of calibrating the DS18B20 devices. The first column is the unique 64 bit device serial number, the second column is the zero intercept of the linear fit to the NIST traceable thermometer, and the third column is the slope.

ds18b20_s/n	intercept	slope
1F00000030E38128	0.06217670441	0.9846894741
B9000000313E6728	0.1135530472	0.9848024249
3300000030D25928	0.09945774078	0.9876177311
3D00000031493E28	0.05446529388	0.9872584343
7700000030D64928	0.2024679184	0.9804323912
3E00000030DD3128	0.06291103363	0.9887344837
4D00000030CFFF28	0.05568313599	0.9874327779
4D00000031222328	0.03308677673	0.9863148928

device serial number, the second column is the zero intercept of the linear fit to the NIST traceable thermometer, and the third column is the slope.

Calibration curves for the individual devices were then used in all data presented here to correct the device response to match the NIST thermometer. With these individual calibration data applied to the DS18B20 we estimate that the absolute accuracy being achieved by each device was within  $\pm 0.01^\circ\text{C}$ , i.e. limited by our ability to read the NIST mercury thermometer.

As described elsewhere, accuracies as great as  $0.05^\circ\text{C}$  can be achieved over an extended temperature range ( $-35$  to  $85^\circ\text{C}$ ). Because of the extended range, fitting DS18B20 calibration data to a second order curve is required. See <http://pdfserv.maxim-ic.com/arpdf/AppNotes/app208.pdf>. Over the relatively narrow calibration range used here,  $-7^\circ\text{C}$  to  $30^\circ\text{C}$ , inspection of Figure 1 shows that fitting to a linear calibration curve is sufficient.

## 2.2 Comparison with the antenna computer thermometer

The antenna computer is interfaced to an electronic thermometer (Make: Weather Measure, Weathertronics, Division of Qaulimetrics; Model: 5123-E; s/n 102) which is hanging in free air near the shutter, on the third floor on the left hand side as seen from the control room. In principle, the antenna

computer thermometer measures the air temperature at that location. However, since circulating air is not forced over the sensor, it more likely measures the temperature of its own enclosure which is an aluminum tube about 15cm long and 2cm in diameter. The antenna thermometer enclosure temperature can be expected to be at or near the free air temperature.

To check the calibration of the antenna computer thermometer, one of the calibrated DS18B20's (s/n 4D00000031222328) was taped to its aluminum enclosure. Thermal contact was made with a dab of conductive grease and the DS18B20 device was wrapped tightly to the antenna computer thermometer aluminum enclosure with silicon tape. Figure 2 shows a time series with data from both thermometers plotted. The data from the calibrated DS18B20 is shown in red and the data from the antenna computer thermometer is shown in black. As can be seen, the two curves are almost coincident. These data verify that the antenna computer thermometer was making very accurate measurements of its enclosure temperature.

### 3 Measurements on the Leighton Telescope

As described above, the DS18B20 technology has the potential to allow logging of massive amounts of accurate temperature data from different places on the Leighton telescope structure. One challenge will be to process in this large amount of data way which provides insight into how telescope thermal behavior affects its optical properties. For example, in real-time it is expected that some effects caused by thermal distortions could be compensated for by secondary mirror positioning (D. Woody, OVRO, private communication). In addition, efficient implementation of antenna surface control system (CSO staff, Surface Memo No.1, October, 2002) requires accurate measurement of the difference in temperature between the standoff and the telescope backing structure around the standoff. The DS18B20 technology being explored in this memo could provide one method for accurately establishing what this difference is in real-time.

To explore the utility of the measurement method, a network of six DS18B20's were distributed on truss elements on the telescope backing structure. Figure 3 shows one of the DS18B20s installed on the horizontal truss element between nodes 67 and 68. The seventh sensor was left mounted on the antenna computer thermometer enclosure to provide ongoing overlap with it measurement. The eighth sensor was not installed and instead it was

returned to the Hilo lab to be available as a secondary calibration source.

### 3.1 A possible ZA dependent gradient in the “up” direction

Figure 4 shows a time series for a few of the logged variables. In this rather messy looking plot are shown Zenith Angle ( $ZA$ , black lines), antenna computer temperature sensor ( $T_{ant}$ , also black lines), and three of the DS18B20’s labelled T6 (in red), T5 (blue), and T4, (green). Figure 5 indicates the location of these three sensors in the backup structure. Sensor T6 was between nodes 67 and 68, T5 between nodes 34 and 33, and T4 between nodes 4 and 3. Placed in this way, sensor T6 provides a temperature indication at the “top” of the backup structure, T5 in the middle, and T4 at the bottom.

Inspection of Figure 4 immediately shows some intriguing behavior. At night, the temperature at sensor T6 was apparently strongly anti-correlated with Zenith Angle. At night, during cool temperatures, a large zenith angle position was almost every time correlated with a dip in temperature at T6.

To see this behavior in more detail, a single night was expanded into one plot, see Figure 6. To see the relative dips in temperature at T6 with even more clarity the data from sensor T6 was differenced with data from sensor T5 located at the center of the structure. That difference is presented as the red line in Figure 7. Also shown on the figure are the Zenith Angle (in black) and  $T_{ant}$  (also in black).

The apparent inverse dependence between Zenith Angle and temperature at T6 is clear to see in these time series. To quantify the inverse dependence over the one night shown in Figure 7, the difference temperature (T6 - T5) was plotted versus Zenith Angle, see Figure 8. Over this particular night,  $\langle T_{ant} \rangle = 2.2^\circ\text{C}$ .

To see if the correlation had any significant dependence on ambient temperature, data from the first night in Figure 4 was also plotted, see Figure 9. On this first night  $\langle T_{ant} \rangle = -0.18^\circ\text{C}$ . Despite the  $\approx 2.4^\circ\text{C}$  temperature difference between the first and third nights, the (T6-T5) versus  $ZA$  dependence in the two figures are fairly similar. The similarity between the two nights suggests that the (T6-T5) difference is dominated more by the zenith angle than by ambient temperature. This inferred behavior is consistent with the observed fact that the CSO optical performance is evidently

repeatable over different nights - at least to the levels measured so far.<sup>1</sup>

The observed dependence between (T6-T5) versus  $ZA$  may represent a zenith angle dependent temperature gradient. How could this come about? Perhaps at larger zenith angles the backup structure in proximity of T6 is subjected to a significantly different thermal environment than T5 which is at the middle of the structure just over the “hex plate”. This difference could come about because there are many thermal sources in the dome and perhaps even the ground acts a thermal sink absorbing heat during the day and radiating it into objects near the surface at night. In contrast, at large zenith angles the sensor at T6 was approximately 10 meters above the ground and much more exposed to the unperturbed free air temperature and cold sky.

### 3.2 Is there a left/right gradient?

In the previous subsection we have seen that there may be evidence of a zenith angle dependent gradient in the “up” direction toward sensor T6. Is there a left/right gradient? To explore that questions we examined the time series from sensors T0, T5, and T2 which were mounted in the backup structure as indicated in Figure 10.

Figure 11 shows a timeseries of  $ZA$ ,  $T_{ant}$ , and sensors T0, T5, and T2. The time series of T0 (green), T5 (blue), and T2 (red), are almost coincident during the three nighttime periods displayed. From this data we provisionally conclude that there is no left/right gradient greater than about  $0.2^{\circ}\text{C}$ . We note that during these nights  $T_{ant}$  (black line) was often cooler than T0, T5, and T2, sometimes by  $1^{\circ}\text{C}$  or more.

## 4 Conclusion

This memo shows that an accurate temperature measurement network based on DS18B20 sensors may be quite practical. A network of such sensors distributed around the antenna could be provide real-time information which could be used to improve the secondary mirror positioning and also to augment the calibration and real-time control of the newly installed surface

---

<sup>1</sup>New measurements available from the SharcII camera (C. Dowel, private communication, Jan 2003) may indicate a very slight temperature trend in the best secondary focus position.

correction system.

Preliminary measurements using a small network of DS18B20 sensors distributed on the Leighton Telescope backing structure show that there may be a zenith angle dependent temperature gradient toward the “top” of the backup structure.

## A HA5 control code

For the measurements presented here, the HA5 one-wire bus controller was interfaced to a linux PC com port.

To run the HA5, with the original DS1820 devices, three programs were developed by former CSO staff member Lynn Holt. The first program, was “ha5probe.c” which verifies the presence of an HA5 bus controller on the computer com port. The second program was “ha5serial.c” which commands the HA5 to interrogate the DS1820 device and get its unique 64 bit serial number. Note that this software interrogation is the only way to obtain the device serial number - there are no unique external marking on the device package. To provide unique device identification there must be only one DS18x20 device on the one-wire bus when running this program.

The third program was “ha5rtmp.c” was designed to read the temperature from eight different DS1820’s. The unique serial number of each DS1820 expected to be on the one-wire bus was hardcoded into the program.

Original versions of all three programs can be found in `ulu//home/lholt/devel`.

For our purposes, reading the DS18B20, it was necessary to significantly modify the original “ha5rtmp.c” because of differences in device timing. See, <http://www.pointsix.com/support/documents/app017.pdf>. The new program for reading DS18B20’s is written as a subroutine and it is called “get\_ha5\_blockmode.c”.

To log the DS18B20’s and the some concurrent telescope antenna computer variables such as time, zenith angle, and antenna computer temperature “get\_ha5\_blockmode.c” is called by “log\_ant\_ha5.c”. Besides reading the HA5, “log\_ant\_ha5.c” also has tcp/ip client code which attaches to a special server provided on the antenna computer. To continually log data, “log\_ant\_ha5” was called by a shell script “loop.sh”. Viable copies of “get\_ha5\_blockmode.c”, “log\_ant\_ha5.c”, and “loop.sh” live in `puuoo//home/cham/ha5`.

Printouts of some of the needed programs are provided here for convenience.



## A.1 ha5probe.c

```
/* read the HA5 N= no response P=OK
*/

#include <stdio.h>
#include <sys/time.h>
#include <unistd.h>
#include <sys/types.h>
#include <linux/termios.h>
#include <sys/file.h>
#include <errno.h>

/* Globals */
int comport;

void gett() {
    int num,i;
    char devdata[8];
    /* HA 5 probe Command*/
    unsigned char mesg[]={ 'a', 'R', 0x0D };
    write(comport,mesg,sizeof(mesg));
    usleep (2000);
    /* get Serial data */
    for (i=0;i<10000;++i) {
        if((num = read(comport,devdata,8)) == -1) {
            if(errno != EAGAIN){
                printf("Communications Error\n");
                return ;
            }
        }
        if(num > 0) break;
        if(i==9999) {
            printf("HA5 not Responding\n");
            return ;
        }
    }
    printf(" HA5: %s \n", devdata);
    return;
}

void main() {
    struct termio svbuf;
    /* serial port setup */
    comport = open("/dev/ttyS0",O_RDWR|O_NDELAY);
    ioctl (comport,TCGETA, &svbuf);
```

```

    svbuf.c_iflag = 0;
    svbuf.c_oflag = 0;
    svbuf.c_lflag = 0;
    svbuf.c_cflag = B115200 | CS8 | CLOCAL | CREAD;
    ioctl (comport,TCSETA, &svbuf);
    gett();
    close (comport);
}

```

Note that the baud rate defined as B115200 in the main() routine should be changed to B1200 to be compatible with the current HA5 configuration.

## A.2 ha5serial.c

```

/* read the silicon serial number of a single DS1820. The device must be the
only one connected to the HA5.
*/

```

```

#include <stdio.h>
#include <sys/time.h>
#include <unistd.h>
#include <sys/types.h>
#include <linux/termios.h>
#include <sys/file.h>
#include <errno.h>

/* Globals */
int comport;

void gett() {
    int num,i;
    char devdata[255];
    /* HA 5 probe Command*/
    unsigned char mesg[]={ 'a', 'S', ',', ',', 'F', 'F', 0x0D };
    write(comport,mesg,sizeof(mesg));
    usleep (2000);
    /* get Serial data */
    for (i=0;i<1000;++i) {
        if((num = read(comport,devdata,255)) == -1) {
            if(errno != EAGAIN){
                printf("Communications Error\n");
                return ;
            }
        }
    }
}

```

```

    }
}
if(num > 0) break;
if(i==999) {
    printf("HA5 not Responding\n");
    return ;
}
}
printf(" HA5> %s \n", devdata);
return;
}

void main() {
    struct termio svbuf;
    /* serial port setup */
    comport = open("/dev/ttyS0",O_RDWR|O_NDELAY);
    ioctl (comport,TCGETA, &svbuf);
    svbuf.c_iflag = 0;
    svbuf.c_oflag = 0;
    svbuf.c_lflag = 0;
    svbuf.c_cflag = B115200 | CS8 | CLOCAL | CREAD;
    ioctl (comport,TCSETA, &svbuf);
    gett();
    close (comport);
}

```

Note that the baud rate defined as B115200 in the main() routine should be changed to B1200 to be compatible with the current HA5 configuration.

### A.3 get\_ha5\_blockmode.c

```

/* get_ha5_blockmode.
 * programmer: lholt with mods by rac.
 * dates: 1998-2002
 * blockmode acquisition added by rac nov02.
 * os's: works under linux and probably qnx
 */

/* read a group of DS1820 devices using the HA5.
The HA5 is set to 1200 baud, address "a", no check sum.
Use ha5serial to obtain the device serial numbers, then plug them in here.
*/

#include "ha5.h"

```

```

int fd,comport;
unsigned char devdata[256];

void get_ha5_blockmode(float out_ha5[8])
{
    /* HA 5 get temperature Command*/
    unsigned char ds_rdtmp[]={ 'a','V',0x0D};

    /* DS18x20's supersede ds1820's. The temperature conversion times
       are extended compared to the ds1820's. The PointSix HA5 'V' command
       is based on the shorter ds1820 conversion time and thus does not
       work with the DS18x20's. To overcome that problem, the ds18x20's
       have to be addressed in "block" mode. See Point Six Application Note
       http://www.pointsix.com/support/documents/app017.pdf -rac, Dec02 */

    /* HA 5 start temperature conversion command*/ unsigned char ds_convrt[]={ 'a','J',
    ',','0','1','4','4',0x0D}; /* HA 5 read scratch pad buffer command*/
    unsigned char ds_read[]={ 'a','J','0','A','B','E','F','F','F','F','F','F','F','F',
    'F','F','F','F','F','F','F','F','F','F',0x0D};

    /* the silicon serial numbers of the devices. use ha5serial to get them. */
    /* nibbles 16, and 17 are family codes. 10h = ds1820 or ds18s20.
        28h = ds18b20. */
    unsigned char* ds_sel[]={ "aA1F00000030E38128\x0D", /* ds18B20 */
                              "aAB9000000313E6728\x0D", /* ds18B20 */
                              "aA3300000030D25928\x0D", /* ds18b20 */
                              "aA3D00000031493E28\x0D", /* ds18b20 */
                              "aA7700000030D64928\x0D", /* ds18b20 */
                              "aA3E00000030DD3128\x0D",
                              "aA4D00000030CFFF28\x0D",
                              "aA4D00000031222328\x0D"};

    struct termio svbuf;
    int devno, i,j;
    short tread;
    float tempc, count_remain,count_perdeg;
    int conval[25], itest;
    char familycode[3];
    char ds1820code[3], ds18b20code[3];

    strcpy(ds1820code,"10\0");
    strcpy(ds18b20code,"28\0");
    /* serial port setup */

```

```

printf ( " Reading HA5...\n" );
fflush(NULL);
comport = open("/dev/ttyS0",O_RDWR|O_NDELAY);
ioctl (comport,TCGETA, &svbuf);
svbuf.c_iflag = 0;
svbuf.c_oflag = 0;
svbuf.c_lflag = 0;
svbuf.c_cflag = B1200 | CS8 | CLOCAL | CREAD;
ioctl (comport,TCSETA, &svbuf);
for(devno=0;devno <=7;++devno) {
    printf ( " %d ", devno );
    fflush(NULL);
    talkDS(ds_sel[devno]);
    sleep(1);
/*    printf(" DS1820: %d ", devno);*/
/*    talkDS(ds_rdtmp); */

    /* use block commands to make compatible with ds18x20's */
    talkDS(ds_convrt);
    /* type out hex string from ds18x20 scratch pad. echo of command */
    for (i=0; i<2; i++){
        /* printf("%c",devdata[i]); */
    }
/*    printf(" < command echo\n"); */

    sleep(1);
    talkDS(ds_read);

    /* type out hex string from ds18x20 scratch pad. command byte + 9byte string
*/
    for (i=0; i<20; i++){
        printf("%c",devdata[i]);
    }
    printf(" < scratch pad result\n");

/* convert from Hex string sent in ascii (???*&~%) */
    for (i=0;i< 16;++i) {
        if ( (devdata[i+2] > 47) & (devdata[i+2] < 58 ) ){
            conval[i] = devdata[i+2] -48;
        }

        else if ( ( devdata[i+2] > 64 ) & (devdata[i+2] < 71 ) ) {
            conval[i] = devdata[i+2]-55; /* A to F converted to int */
        }
        else

```

```

        {
/*      printf(" ***unexpected characters from HA5: exiting in get_ha5() \n");
*/
        fprintf(fp1_data_out," ***unexpected characters from HA5 \n");
        break;
    }
    /* printf( " i = %d, conval[i] = %d, devdata[i] = % d \n", i, conval[i], d
evdata[i] );*/
    }
    /* printf(" family code %s \n \n", ds_sel[devno]+16 ); */
    strcpy(familycode, ds_sel[devno]+16);
    familycode[2]=0;
/*      printf( " family code string = %s \n", familycode);
      printf(" ds1820 test string = %s \n", ds1820code); */
/* if (familycode[0] == 49){ */
    itest = strcmp(familycode,ds1820code);
    /* printf(" itest = %d \n", itest); */
    if ( strcmp(familycode, ds1820code ) ==0 ) {
        tread = (conval[0] *0x10 + conval[1])
            +((conval[2] *0x10 + conval[3]) *0x100 );
        count_remain = (conval[12] *0x10) + conval[13];
        count_perdeg = (conval[14] *0x10) + conval[15];
        tempc =(tread /2.0 ) - 0.25 + ((count_perdeg - count_remain)/ count_perdeg
);
        out_ha5[devno] = tempc;
        printf(" count_remain = %f \n", count_remain);
        printf(" count_perdeg = %f \n", count_perdeg);

        printf(" %+-2.4f degrees C \n\n", tempc);
    }
    itest=strcmp(familycode,ds18b20code);
    if( itest==0){
        /*      printf(" this device is an ds18b20\n"); */
        tread = (conval[0] *0x10 + conval[1])
            +((conval[2] *0x10 + conval[3]) *0x100 );
        tempc =(tread *.0625);
        out_ha5[devno] = tempc;
        /* printf(" %+-2.4f degrees C \n\n",tempc); */

    }
    /* printf( "\n");*/ }
    close (comport);
}

```

```

int talkDS(char * mesg) {
    int num,i;
    /* clean up the data array */
    for(i=0;i<=255;++i) {
        devdata[i]= 0;
    }
    write(comport,mesg,strlen(mesg));
    usleep (1000000); /*the 1-wire devices are very slow to respond */
    /* get Serial data */
    for (i=0;i<10000;++i) {
        if((num = read(comport,devdata,255)) == -1) {
            if(errno != EAGAIN){
                printf("Communications Error in get_ha5() \n");
                fprintf(fp1_data_out," ***Communications Error in HA5");
                break;
            }
        }
        if(num > 0) {
            /* printf( " num = %d", num);
            for (i=0; i<= num; ++i) {
                printf(" i = %d devdata = %d \n", i, devdata[i]);
            } */
            break;
        }
        if(i==9999) {
            printf("HA5 not Responding\n");
            fprintf(fp1_data_out, " ***HA5 not resonding\n");
            break;
        }
    }
}

```

## B Error in the previous memo

In the previous memo in this series, “A method for implementing the CSO surface correction hardware: Memo No.1”, 18-Oct-2002, we note that there is an error on page 3, section 3.1, line 7. There it is stated that “ $D_k$  is in the direction parallel to the optical axis”, whereas it should state that “ $D_k$  is in the direction normal to the surface”.

## References

- [1] J. W. Lamb. Thermal considerations for mmA antennas. Millimeter Array memo no. 83, NRAO, 1992.



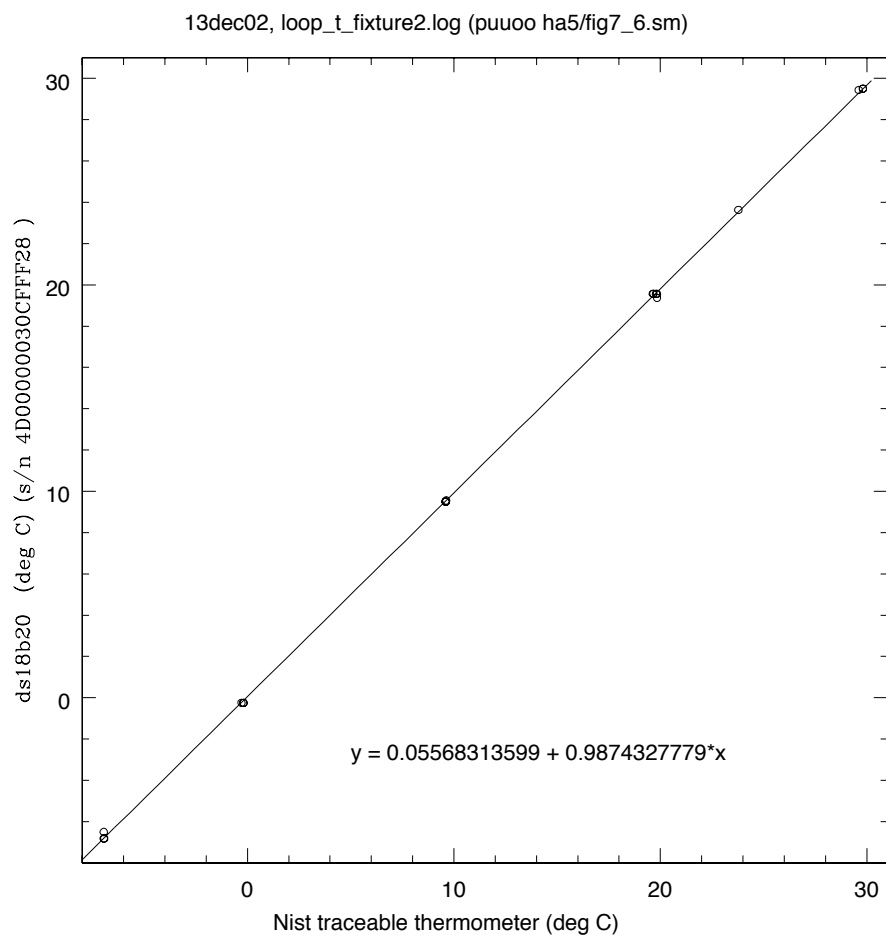


Figure 1: A typical DS18B20 calibration curve. The slope and intercept are given in the body of the plot. The individual device serial number is given in the y-axis label. The top label on the plot just indicates some specific details about where this plot creation file is located, its creation date, etc.

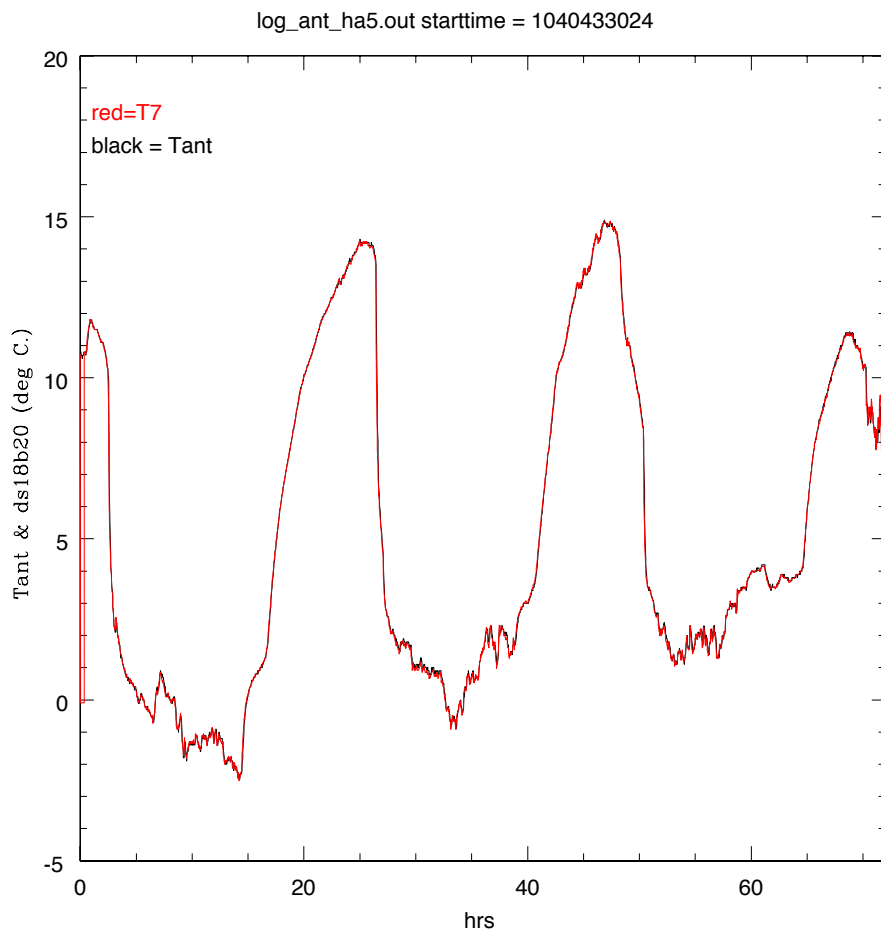


Figure 2: The antenna computer thermometer (Tant in black) and one of the DS18B20's (in red) are compared over a 72 hour period. As can be seen, their indications are almost identical. The plot top label indicates the data source file and the plot start time in unix “ctime”.

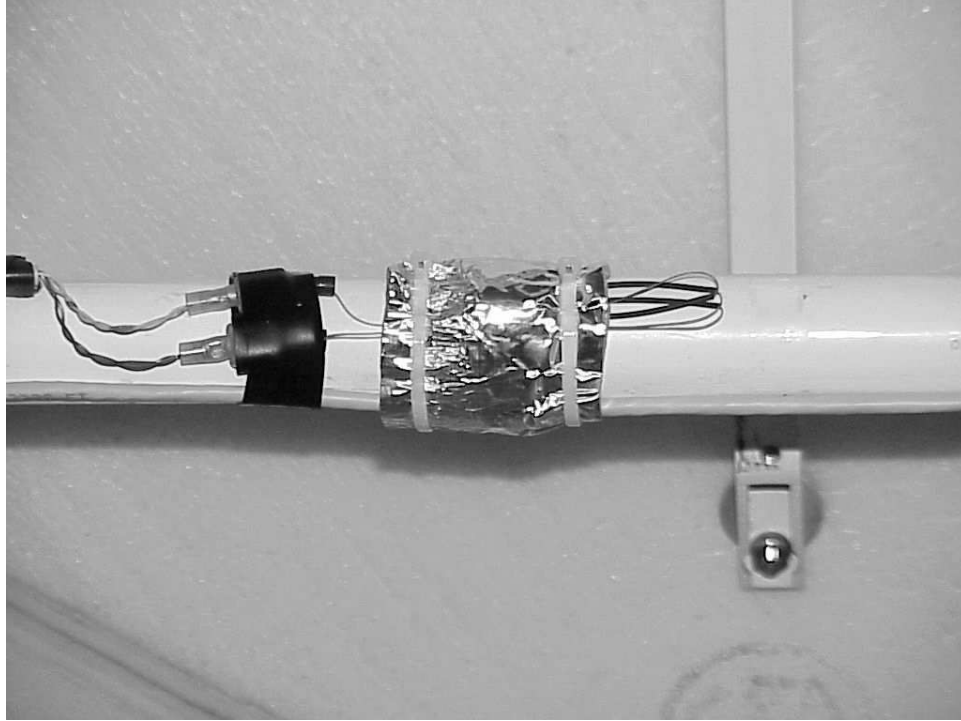


Figure 3: An installed DS18B20. The device is contacted to the truss element with a dab of conductive grease and then wrapped tightly with silicon tape. To provide additional isolation from ambient conditions, an insulated, reflective mylar layer is wrapped around the device location. Also visible are the wire wrap leads which make the short drop from the CAT-5 one-wire. For the installation presented here, simple automotive-type press-fit connectors were used for connection to the device drop.

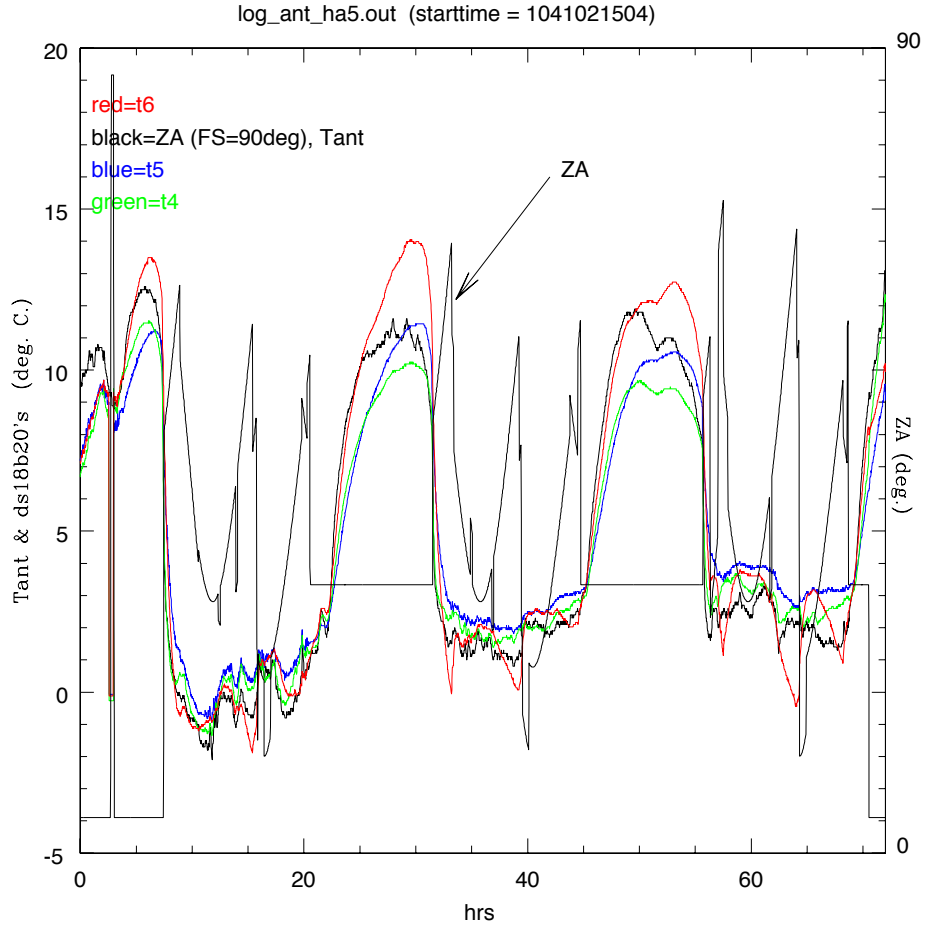


Figure 4: A 72 hour time series of the Zenith Angle,  $T_{ant}$ , and response from three DS18B20 sensors (T6, T5, T4) distributed in an up/down direction in the telescope backing structure.

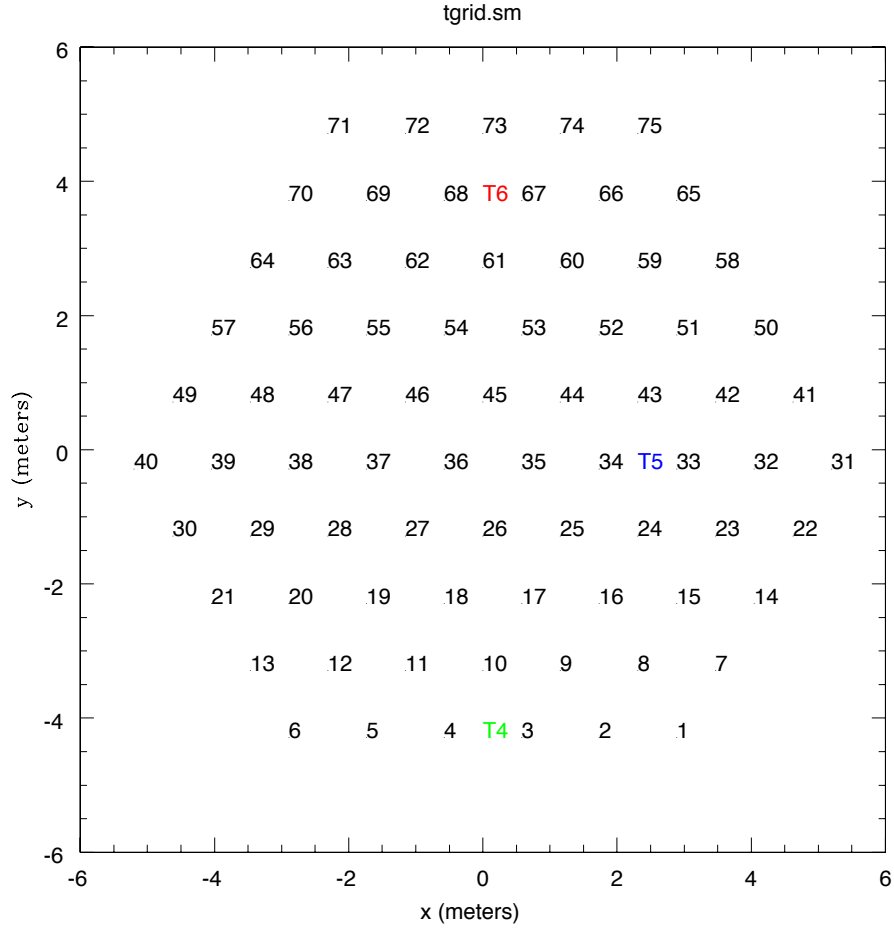


Figure 5: The distribution of the T6, T5, & T4 DS18B20 sensors in the backing structure. The node numbers are shown in black. Most nodes on the circumference of the backing structure (nodes 76-99) are not shown.

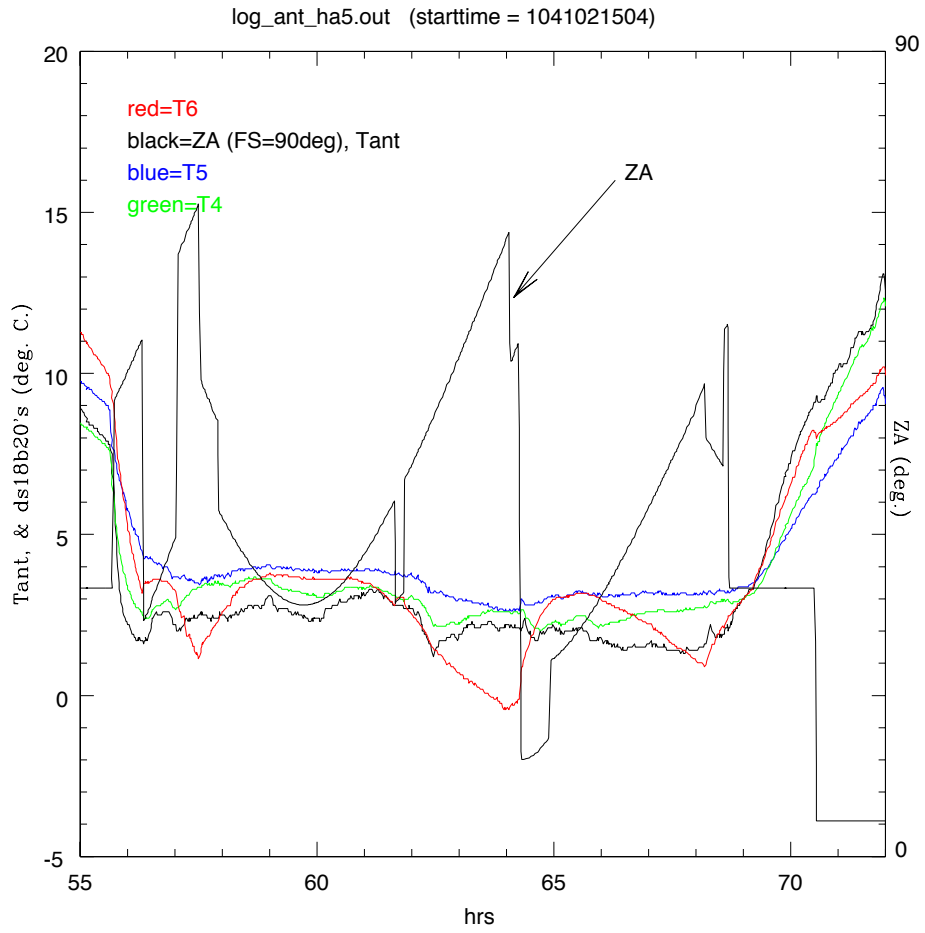


Figure 6: The time series of Figure 4 is expanded to show a one night time series in more detail.

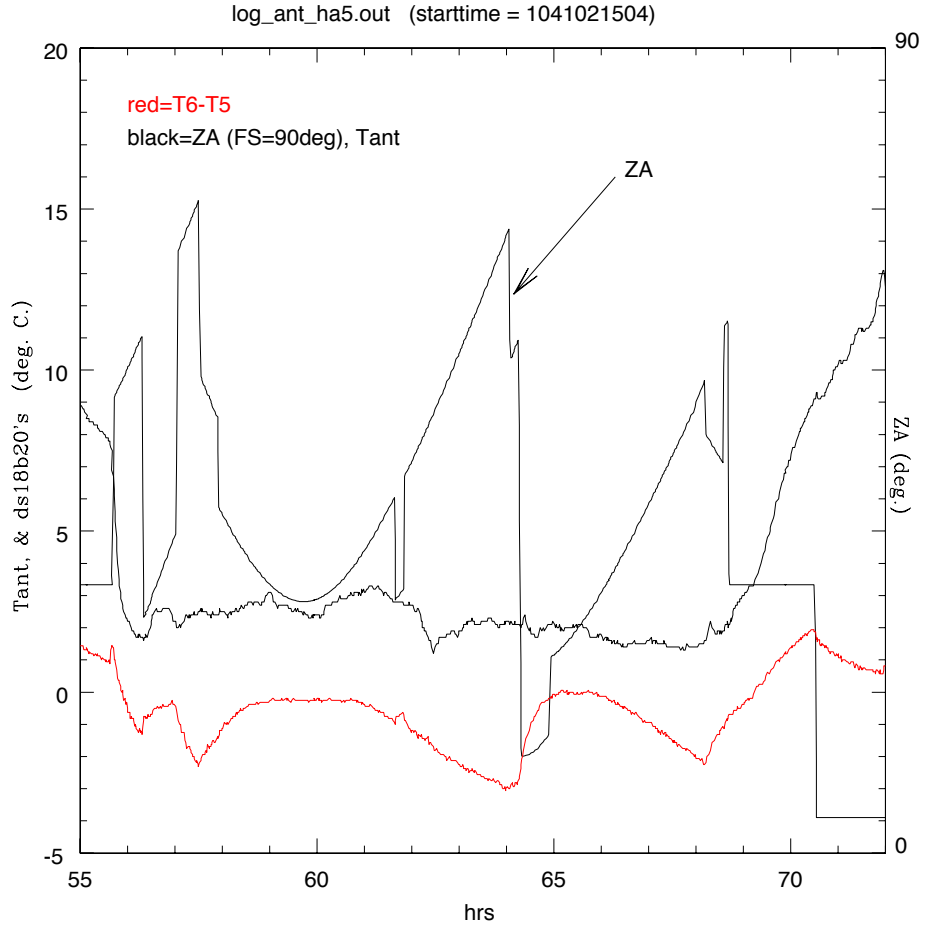


Figure 7: The time series of the difference between sensors T6 and T5 is shown as a redline. Also shown are the Zenith Angle and  $T_{ant}$  time series. The data is from Figure 6.

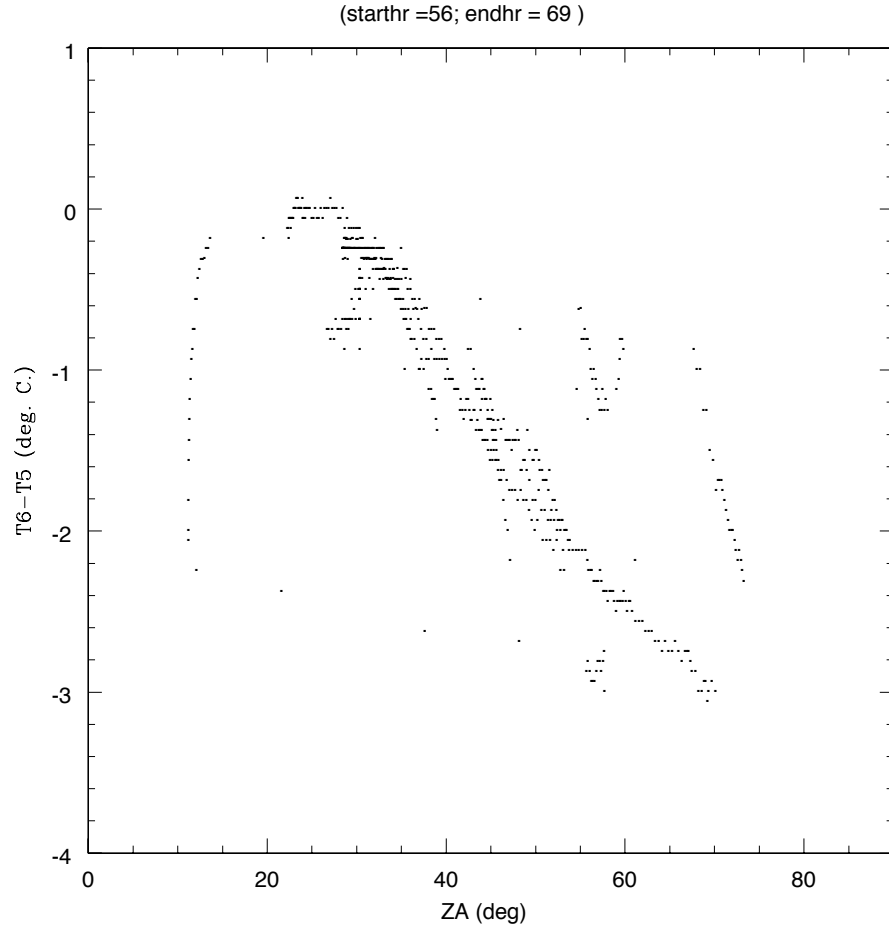


Figure 8: Nighttime ( $T_6-T_5$ ) and Zenith angle data from Figure 7 are plotted against each other to explicitly show their inverse correlation. During this night  $\langle T_{ant} \rangle = 2.2^\circ\text{C}$ .



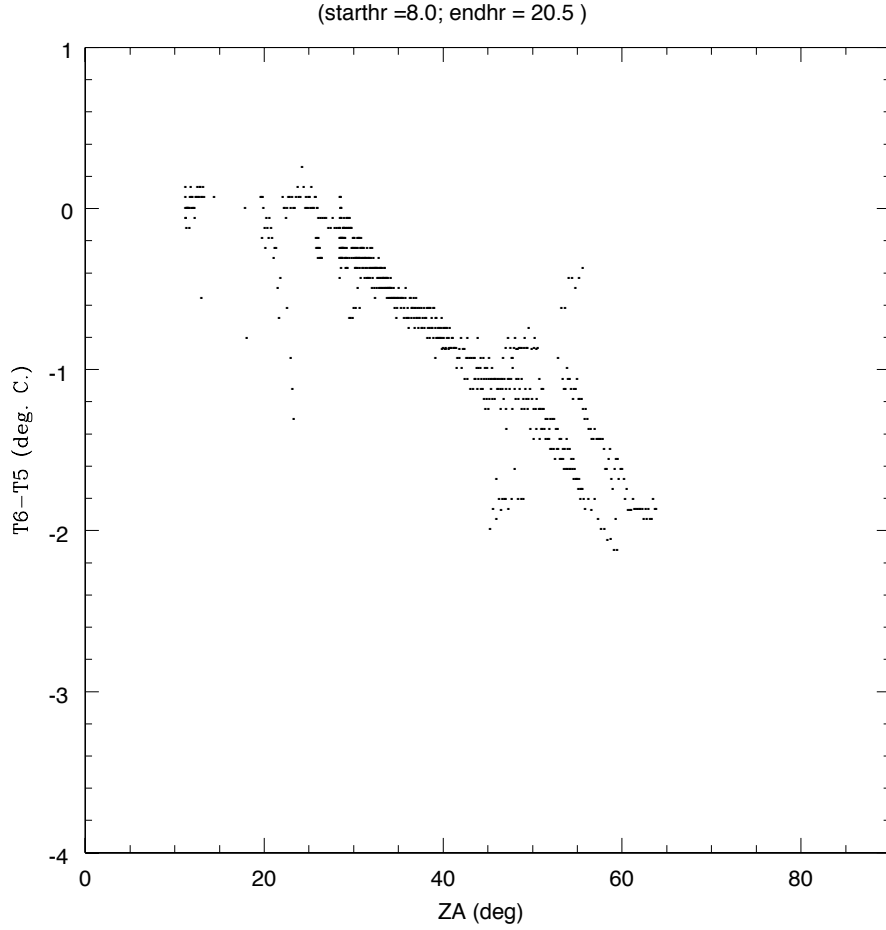


Figure 9:  $(T_6-T_5)$  and Zenith angle data from the first night in Figure 4 are plotted against each other to explicitly show their inverse correlation. During this night  $\langle T_{ant} \rangle = -0.18^\circ\text{C}$ . We note that comparison of this plot and the previous plot show a similar  $(T_6-T_5)$  vs.  $ZA$  dependence despite the difference in  $\langle T_{ant} \rangle$  between the two nights.

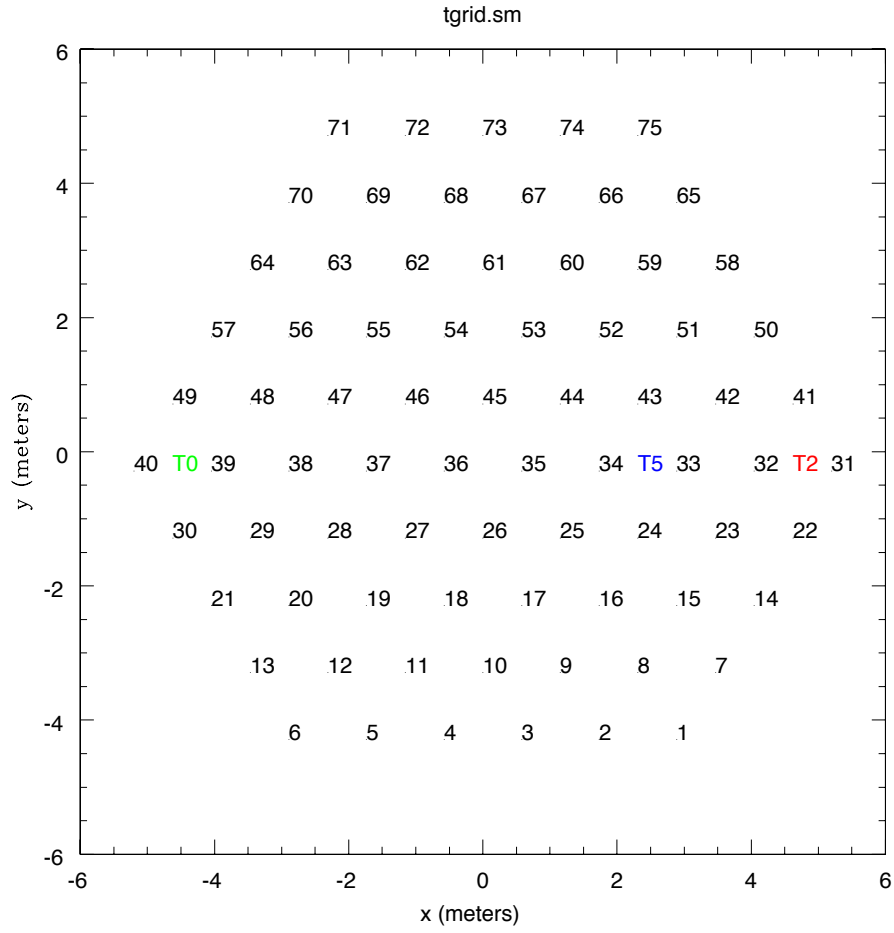


Figure 10: The distribution of the T0, T5, & T2 DS18B20 sensors in the backing structure. The node numbers are shown in black. Most nodes on the circumference of the backing structure (nodes 76-99) are not shown.

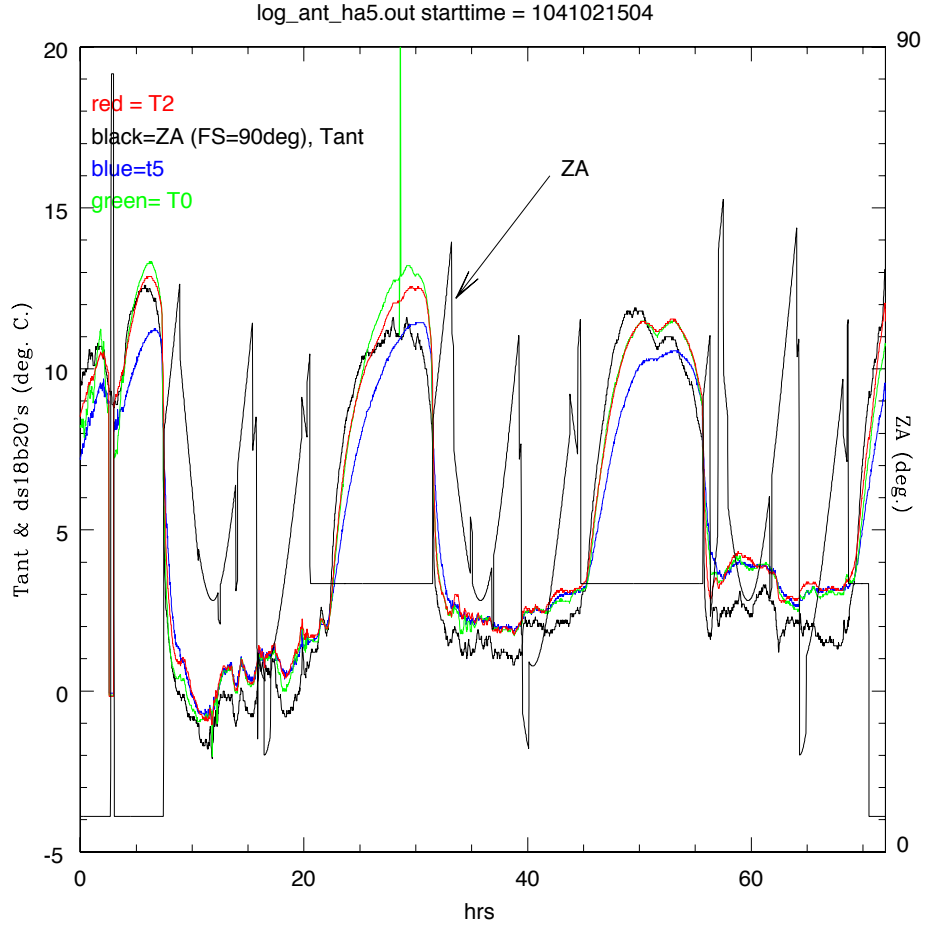


Figure 11: A 72 hour time series of the Zenith Angle,  $T_{ant}$ , and response from three DS18B20 sensors (T0, T5, T2) distributed in a left/right direction in the telescope backing structure. Inspection of this time series implies relatively little, if any, left/right thermal gradient during nighttime.