TSB Software Reference Manual

Temperature Sensors Board Commands Description

L. Castellani

I.N.F.N. sez. PADOVA

10 December 2009

Document Version 1.4

Firmware Version 2.5

Introduction

The TSB emulate the mini-crate protocol communications and implement only the command concerning the temperature sensors. The board can read up to 100 sensors
At power on the board transmit the following data:

```
char 0xDA
char b1w_err1;          The DS2482_800 initialization error.
char b1w_err2;          The DS2482_100 initialization error
char nsensor;           The number of found sensors.
int  id;                The board identifier.
char HVersion;          High version of the firmware.
char LVersion;          Low version of the firmware.
int LaserCurrent;       The laser current.
```

The commands:

Watchdog Reset, code 0xF8:
This command force CPU to enter in a infinite loop to force a watchdog reset. CPU will be reset after sent return data.

Return data:

```
char 0xFC;
char 0xF8;
```

HD Watchdog Reset, code 0xF7:
This command force CPU to enter in a infinite loop to force a watchdog reset. CPU will be reset after sent return data.

Return data:

```
char 0xFC;
char 0xF7;
```

Read Com Error, code 0xF0:
Read and clear the first error on communications ports.

Return data:

```
char 0xF0;
char port;              primary port=1. secondary port=2
char Parity:1;
char Framing:1;
char Break:1;
char Noise:1;
char Overrun:1;
char Sync:1;
char Crc:1;
char LoseData:1;
char Size:1;
char TimeOut:1;
char Unexpected:1;
char unused:4;
```

Set Current, code 0xD0:
Set the laser current.
Arguments:

int current;	DAC value to set current. I(A)=DAC*5/(1024*25.5ohm). Maximum value 600=117mA.

Return data:
char 0xFC;
char 0xD0;

Set CFG, code 0xDD:
Set the DS2482-800 configuration register.
Arguments:

char cfg;	bit0 Active Pull-up, bit1 Presence Pulse Masking, bit2 Strong Pull-up, bit3 1-Wire Speed;

Return data:

char 0xD2;	
char err1;	The DS2482_800 initialization error.
char err2;	The DS2482_100 initialization error.
char cfg;	Configuration value stored in eeprom.

Set ID, code 0xDC:
Set the board identifier.
Arguments:

int id;	The bard identifier;

Return data:
char 0xFC;
char 0xDC;

Status, code 0xEA:
Return some info on the firmware and sensors.

Return data:

char 0xDB;	
char nsensor;	The number of found sensor.
int id;	The board identifier.
char HVersion;	High version of the firmware.
char LVersion;	Low version of the firmware.
int Current;	Laser Current;
int OptOfs;	Offset optical signal;
int OptAmp;	Amplitude optical signal
int CalOfs;	Calibration offset;
int AmpliOfs;	Offset amplified signal;
int AmpliAmp;	Amplitude amplified signal.
int Noise;	Noise on the amplified signal.
char Mode;	Mode to measure amplitude.
char Filter;	Filter for the amplitude measure.

I2C Commands, code 0x64:
Execute I2C commands sequence on I2C bus . WARNING: The sequence must terminate with a I2C stop.
Arguments:

int ctrl_data[size]; — The bit8 is the start flag, bit9 is stop flags, bit10 is read flags, bit11 is ackn flag, bit12 is abort on error, the bits from 0 to 7 are the write data and will be ignored if read flag is 1.
Size can be from 1 to 25.
If start flag is 1 will be executed I2C start sequence before the read or write data on I2C bus.
If stop flag is 1 will be executed I2C stop sequence after the read or write data on I2C bus.
If read flag is 0 ackn flag is ignored and will be executed a write command with the data bit(7..0).
If read flag is 1 will be executed a I2C read command with I2C ACKN bit set to the state of ackn flags.

Return data:

char 0x65;
char err_data[size] ; — The bits from 0 to 7 are the read data if read flag was 1, or wrote data if read flags was 0. The bits from 8 to 15 are the I2C bus diagnostic 0=OK, 1=WRITE error (the I2C ACKN bit was 1) , -1 =SDA short circuit, -2=SCL short circuit. The size is the same of the sent data if you have not enabled abort on error.

Temperature, code 0x3C:
Read temperature.
Arguments:

char bank; — bank from 0 to 4;

bank	Description
0	Read sensor on bus 1wire 0, 1, 2
1	Read sensor on bus 1wire 3, 4, 5
2	Read sensor on bus 1wire 6
3	Read sensor on bus 1wire 7
4	Read sensor on bus 1wire 8

Return data:

char 0x3D;
float temp[20]; — The temperatures.
char code[20][8]; — The sensor ID;

Read sensors ID, code 0x8F:
Read temperature sensors identified code . Identified code have 64 bit format see DS18S20 data sheets.
Arguments:

char bank; — bank from 0 to 4;

bank	Description
0	Read sensor ID on bus 1wire 0, 1, 2
1	Read sensor ID on bus 1wire 3, 4, 5
2	Read sensor ID on bus 1wire 6
3	Read sensor ID on bus 1wire 7
4	Read sensor ID on bus 1wire 8

Return data:

char 0x90;
char code[20][8]; — Identified code of the sensors

Find Sensors, code 0xA6:
Find all sensor connected on to nine 1wire bus.

Return data:
char 0xA7;
char nsensor; The number of sensors found.

DAC, code 0xD1:
Set the DAC value.
Arguments:
char ch; The channel number from 0 to 3;
int value; The dac value in little_endian format
Return data:
char 0xFC;
char 0xD1;

ADC, code 0xD3:
Get the ADC value.
Arguments:
char ch; The physical PIC channel number from 0 to 8; channel 0 and 1 are connected to optical link, channel 2, 3, 4 are not connected and channel 5, 6, 7, 8 are connected to J5 connector,
Return data:
char 0xD4;
int value; The ADC value in little_endian format

Retransmit , code 0xB5:
This command can be used to request the last transmission data.

Read Histogram, code 0xB0:
Read the amplified optical signal histogram, the bin of histogram is 3 bits.
The histogram is generated every second, and it is not update if you read only the first bank.

Arguments:
char bank; The bank, from 0 to 1;
Return data:
char 0xB1;
int data[64]; The histogram data, in little_endian format.

Set Measure Mode and Filter Noise, code 0xC7:
Set the mode to measure the amplified optical signal and the filter noise.
The measure is used to set the threshold of the signal discriminator of the primary communication port.
Arguments:
char mode; mode=0 use the min and max acquired value to calculate the amplitude signal ;
mode=1 use the histogram to calculate amplitude signal.
char filter; if mode=0 filter is the level in DAC counts over which the signal amplitude is calculated;
If mode=1 filter is the minimum distance between two bigger peaks, the distance(DAC counts)=filer*8. Only if exists two peaks with the specified characteristic the amplitude is calculated.
Return data:
char 0xFC;
char 0xC7;

Special codes:

The special code can be added to the head of all commands. A command can have more than a special code added to the head, but of different value.

Host, code 0xEC:
This code followed by one byte argument can be used from a server program to mark the commands send by client. The return data from the CCB have to the head the same code and argument. The argument can assume any value.

Split, code 0xB6:
This code followed by one byte argument can be used to split large return data from mini-crate.
When mini-crate receive a command with this code at the head, following by the argument that define the size to split data, it if the return data exceed the specified size data, transmit only amount of the size data specified. The data are marked at the head by code 0xB6 following by one byte that define the data offset and one byte that define the data size remaining to transmit. To receive remaining data mini-crate must receive command code 0xB7 describe below.

Send Remaining, code 0xB7:
This is a standard command used to read remaining return split data.
Arguments:

```
        char offset;
```

Return data:

```
        char 0xB6;
        char offset;            data offset.
        char remaining;         reaming data to transmit.
        char data[size];        size is the value specified with the split code.
```