

# **Temperature monitor**

## **Manual**

Brookhaven National Laboratory  
Version 1.0

By Peter Kravtsov (E-mail: [PAKravtsov@lbl.gov](mailto:PAKravtsov@lbl.gov))

This document location: [http://www-rnc.lbl.gov/~pkravt/pdf/tm\\_manual.pdf](http://www-rnc.lbl.gov/~pkravt/pdf/tm_manual.pdf)

August 2003

## Introduction

STAR TPC detector is equipped by 120 temperature sensors placed on the East and West sector mounting wheels and on the FEE and RDO cooling manifolds. Temperature monitor (Fig. 1) is intended for reading out these sensors and provide temperature information for the slow control. These sensors are grouped by location into six groups: Inner Wheel, Outer Wheel, Inner Sector RDO, Outer Sector RDO, Inner Sector FEE, Outer Sector FEE.



Fig. 1. Temperature monitor.

The device is based on the Atmel AT89S8252 (Fig. 2) microcontroller running at 22.1184 MHz clock frequency. It has 8Kb static RAM to keep the temperature values and all data. There is also internal 2Kb EEPROM in the CPU which is used for keeping sensor names and numbers. Analog signals from the National Semiconductors LM34 sensors go through eight 16-channel

multiplexers [MPC506] and instrumentation amplifier [AD711] to 16-bit ADC [ADS7813].

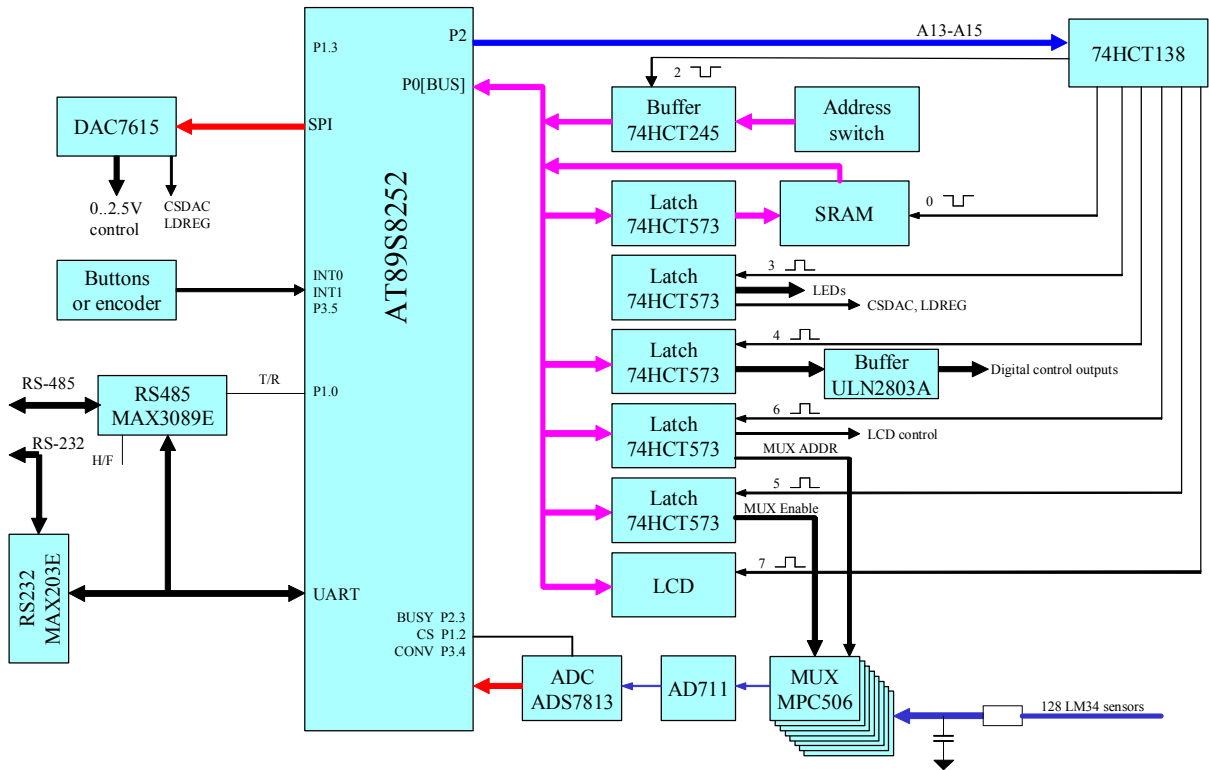


Fig. 2. Temperature monitor function diagram.

All signal lines have RC filters before the multiplexers to avoid long line noise. CPU has internal watch-dog, which attends to controller program faults and resets CPU as fast as in 0.5 s in this case.

Temperature monitor can be connected to PC via standard RS-232 or RS-485 port. Besides, the device is equipped by LCD indicator to display sensors temperature, control buttons, 8+6 open collector digital outputs that can be used in alarms handling procedure. There is a possibility to connect any  $\pm 10V$  sensors to this device. ADC is normally working in 0÷4V range, but this could be changed to 0÷10V or to  $\pm 10V$  range. In spite of single +5V power, multiplexers and amplifiers are powered by  $\pm 15V$  via DC-DC converter [DCP020515D] and can be used for the signals of  $\pm 10V$  range.

Thus input/output features of temperature monitor are the following:

- 128 analog inputs (16-bit, 0÷4V or 0÷10V or ±10V, optional averaging);
- 14 digital outputs (open collector, 500mA maximum current);
- 4 optional analog outputs (12-bit, 0÷2.5V).

There are 26 connectors (9 pin female DSUB marked X1-X26) at the top of temperature monitor to connect all sensors. Each connector has pin 6 connected to +5V for sensors power and pin 9 connected to ground. Pins 1÷5 are used for sensor signals. One additional connector (9 pin male DSUB) connected to 8 digital outputs. Pin 5 of this connector is connected to ground.

## **Controller software**

Controller software is written in C language. It provides reading of 128 sensors with programmed averaging by 1-255 samples, handles communication with the host computer. One of the temperature values is displayed on the LCD. Two buttons are used to scroll the sensors: one selects sensors group (Inner Wheel, Outer Wheel, Inner Sector RDO, Outer Sector RDO, Inner Sector FEE, Outer Sector FEE, Extra 8 sensors) while second one scrolls the sensors of selected group. Sensor names and numbers could be programmed from the host computer and written to the CPU EEPROM.

Optionally controller software also checks all sensors for alarm limits and generates alarm signals.

## **Serial protocol**

Communication protocol consists of two main commands: read memory byte and write memory byte. It could contain also some special commands which are specially described for every device. Protocol is based on 5-byte binary packets exchange. The device sends 5-byte answer for every 5-byte command, if device address field in this command is correct.

Command structure is following:

Byte	Description
1	6-bit device address. Should be equal to device address set on the dip switch inside the device or packet will be ignored otherwise. Two most significant bits are ignored.
2	Bit 7 – read/write bit. Bit 7 = 1 corresponds to write command and bit 7 = 0 – read command. Bit 6 – special command flag. If bit 6 = 1, packet contains special command. Bits 5÷0 – high bits of 14-bits address field.
3	Low byte of 14-bits address field.
4	Data byte.
5	XOR-sum of first 4 bytes. $B5=B1^B2^B3^B4$ . If it is not correct XOR-sum, packet is ignored.

Examples:

1. Host computer wants to read byte at address 0x345 from the device with address 0x02. Device memory contains 0xAA at this address.

```
host request : 0x02 0x03 0x45 0x00 0x44
```

```
device answer : 0x02 0x03 0x45 0xAA 0xEE
```

2. Host computer wants to write byte 0x55 at address 0x1543 to the device with address 0x08 (Note that read/write bit is cleared in the answer).

```
host request : 0x08 0x95 0x43 0x55 0x8B
```

```
device answer : 0x08 0x15 0x43 0x55 0x0B
```

Communication speed could be selected from 9.6, 19.2, 57.6 or 115.2 Kbit/s by dip switch inside the device (see table below). Usually read or write access to the device memory using this protocol takes not more than 2ms for every byte at 115.2 Kbit/s. Memory map is of course different for every device and is specially described as well as additional commands. Byte ordering is little endian, high byte has low address.

Configuration switch description:

Switch position	Description (0=OFF, 1=ON)
1-6	6-bit device address in binary format. Least significant bit is selected by switch position 1. Valid range is 1÷63.
7-8	Communication speed: 00 – 9600 bit/s 01 – 19200 bit/s 10 – 57600 bit/s 11 – 115200 bit/s

Temperature monitor accepts one special command, with byte 2 = 0x41. In response to this command it sends all temperatures as word[128] array and one byte of XOR-sum in the end (257 bytes totally). This command is fastest way to get all temperatures from the device. At 115200 bit/s communication speed it takes about 30ms, while reading by one memory byte takes about 400ms.

## Memory map

Address	Length	Type	Name	Description
0x0000	2	word	WDCCount	Watch-dog resets counter. Should be 0 or small value and not increase. This counter resets to 0 at every “cold” start, i.e. power-on procedure and increases at every watch-dog CPU reset.
0x0007	1	byte	AVGCount	Number of samples to average. No averaging occurs if = 0. Default value is 8.
0x0008	1	byte	ADCchan	ADC channel number. Device will measure only this channel (0÷127) or all 128 channels if ADCchan>127.
0x0009	1	byte	DOUTByte	Digital outputs control byte.
0x000F	1	Byte	ID	Device ID = 0xA1. It is NOT device address.
0x0010	256	Word	ADCval[128]	Sensor values in 2-byte integer form. Temperature should be calculated by the formula: $T [F] = \text{ADCval} / 0xFFFF * 400$
0x04FE	1	Byte	UpdateNums	Control byte for updating sensor numbers in EEPROM. If UpdateNums=1, CPU will write sensor numbers from RAM to internal EEPROM. CPU reads numbers from EEPROM to RAM in startup procedure.

0x04FF	1	Byte	UpdateNames	Control byte for updating sensor names in EEPROM. If UpdateNames=1, CPU will write sensor names from RAM to internal EEPROM, where they will be kept until next power-on procedure. This takes about 1.5 seconds. CPU reads names from EEPROM to RAM in startup procedure.
0x0500	512	Char	Names[128][4]	4-byte sensor names for LCD display.
0x0700	168	Byte	snum[7][24]	Sensor numbers for 7 groups and 24 sensors in each group. Group names are kept in Flash memory and could not be changed.

## PC software

Temperature monitor control software for the host PC was developed for Windows platform. Using this software, one could see all 128 temperature values (Fig. 2) either by sensor names or by connector and pin numbers. It also makes it possible to change sensor names and numbers in the device EEPROM (Fig. 3). Number of samples to average also could be changed by user. Note that this number will reset to 8 in device power-up procedure.

The software reads sensor values using byte memory access or buffer command (this can be changed in the setup window). All sensor values could be saved in history file for future analysis.



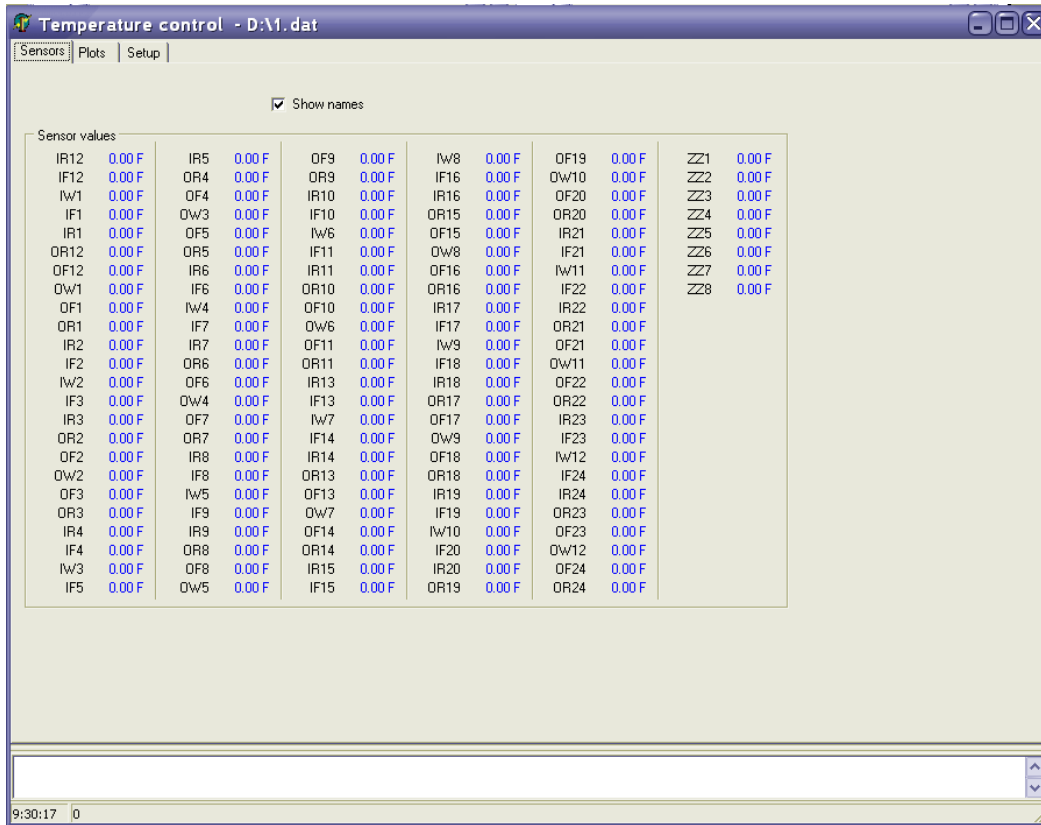


Fig. 2. Sensors display.

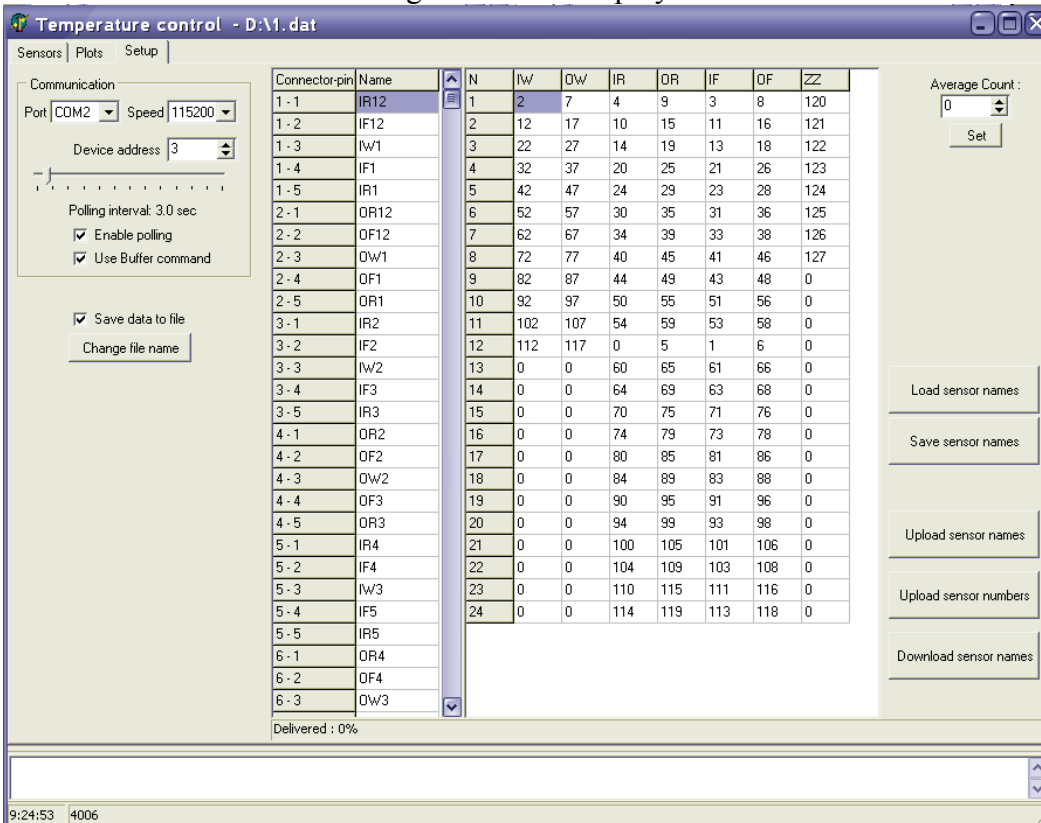


Fig. 3. Configuration window.