**SM**art

**T**emperature

**A**cquisition

**S**ystem

for

**08** channels

# <u>SMTAS08</u>

# Contents.

Smartec © 2002

## 1. Introduction

This document describes an 8 channel temperature measurement system Smart Temperature Acquisition System or SMTAS08. It is based on the use of the Smart-temperature sensors SMT160 of Smartec. The SMT160 is a three-terminal integrated temperature sensor with a duty cycle output. Two terminals are used for the power supply of 5 volts and the third terminal carries the output signal. The output signal of the sensor is a duty-cycle-modulated square-wave signal (see Figure 1).
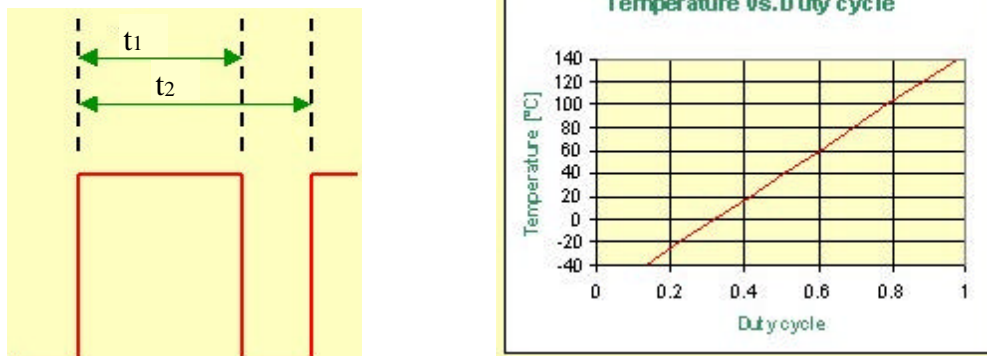


*Figure 1 (a) Output signal of the SMT, (b) Relation between the duty-cycle and the temperature.*

After measuring both $t_1$ and $t_2$, the temperature   in $^0$C is calculated by equation:

$$\boldsymbol{q} = \frac{t_1}{0.0047t_2} - 68.09$$

The temperature sensors are sold separately from the SMTAS08 board, because the SMT 160-30 temperature sensor is available in different encapsulations (TO18,TO92, TO220, etc), each with their specific properties. One important issue is their accuracy. The TO18 version yields the most accurate sensor and has an accuracy of 0.7 $^0$C. The complete specifications of the temperature sensor range is presented in the datasheet, which should be consulted in conjunction with this document.

The SMTAS08 is equipped with a microcontroller of the type Microchip PIC16F876. An RS232 interface chip offers external serial communication with the microcontroller. In turn each of the sensors is powered by a corresponding processor output. An analog multiplexer (74HC4051), which is controlled by the microcontroller, selects one of the eight sensor outputs by connecting it to the microcontroller input port. The input is then sampled to measure the values of *t1* and *t2* for this sensor. This process is repeated for each of the sensors, after which the 8 values of *t1* and *t2* are sent to the RS232 port.
A block diagram and a photograph of the system are shown in Figure 2 and Figure 3, respectively.
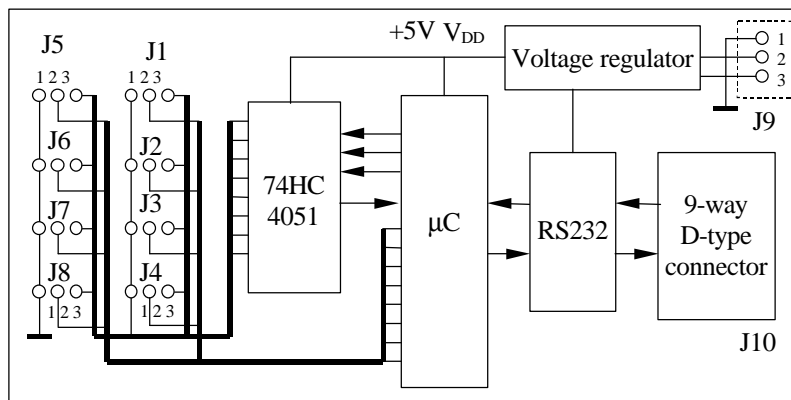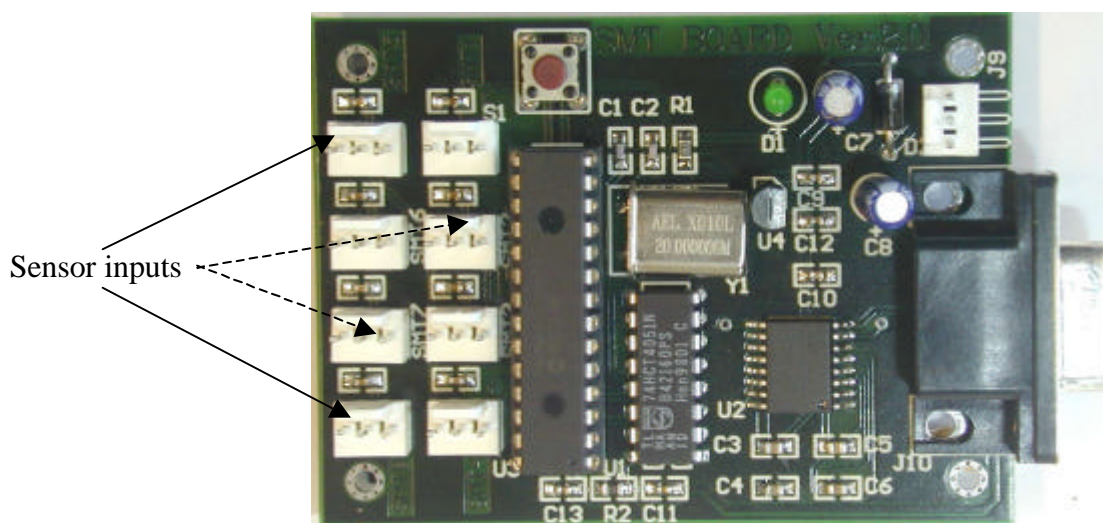
3

*Figure 2 Functional block diagram.*



*Figure 3 A photograph of the SMTAS08 system.*

## 2. Quick start and functional check

This document assumes that the SMTAS08 board has to be connected to a personal computer or a labtop. Any other device capable of handling RS232 data could do the job just as well however.

In order to get the board running, a few things have to be prepared. Please find the hardware checklist below:

- The SMTAS08 board itself
- One or more SMT 160-30 sensors
- A D9 RS232 connecting cable (straight)
- A power supply delivering between 7 and 18 Volts (min 20 mA)
- A PC or laptop running a terminal program for instance Windows Hyper terminal (57600,8,non,1,none)

After connecting the parts together temperatures can be measured in case one or more temperature sensors are connected to the SMTAS08 board. Make sure the com port parameters, as mentioned above, and the portnumber itself are chosen correctly. Type "m" and the datastream should start flowing. More information about this will be given later but this is the fastest way to check whether the SMTAS board is functioning correctly.

4

Please refer to section 4 for the pin layout of the board and to the datasheet of the SMT160-30 for the sensor pin connections. In case no sensors are connected to the SMTAS08 board, in the terminal program all data will appear as "FFFF".

## 3. Inside the SMTAS08 system

**General**

The SMTAS08 system can be used to measure up to 8 Smartec temperature sensors. The microcontroller measures the output signal of the selected sensor, provides the 5 V power supply for the selected sensor and communicates with the outside digital world. Figure 4 shows the flowchart of the program in the microcontroller.

For speed of measurement of the duty-cycle of the sensors, the program of the PIC processor is written in Assembler. The processor is mounted on a socket so the user is free to write his own program if desired.
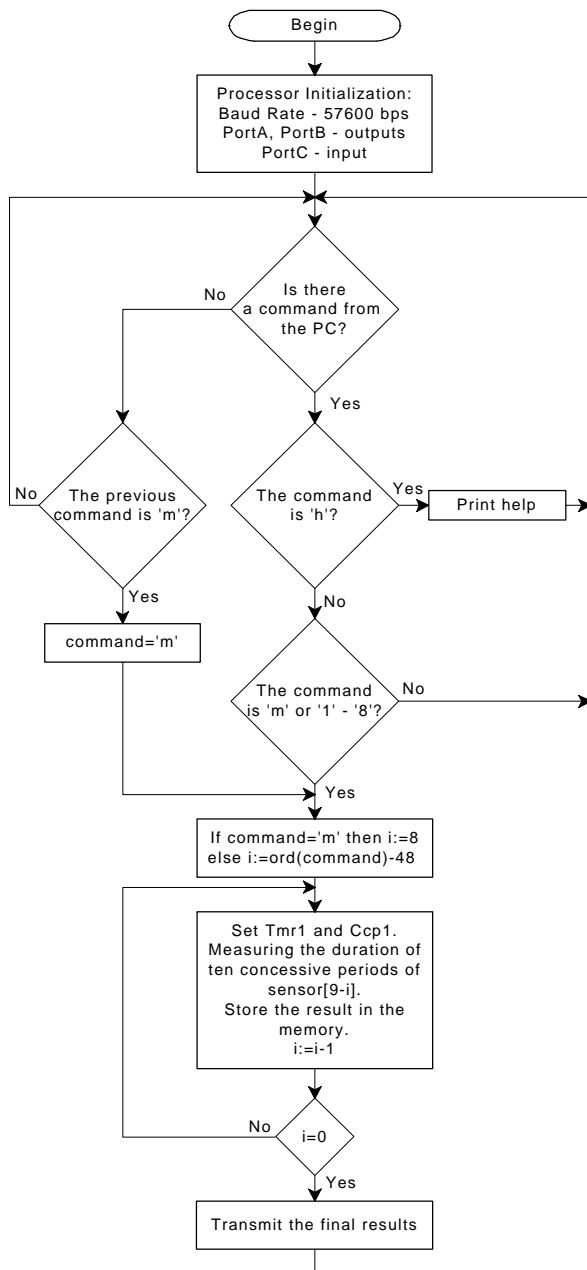
```
                    ┌─────────────┐
                    │    Begin    │
                    └──────┬──────┘
                           ▼
              ┌──────────────────────────┐
              │ Processor Initialization: │
              │   Baud Rate - 57600 bps   │
              │   PortA, PortB - outputs  │
              │      PortC - input        │
              └──────────────────────────┘
```

Is there a command from the PC?  — No / Yes

The previous command is 'm'? — No / Yes

command='m'

The command is 'h'? — Yes → Print help / No

The command is 'm' or '1' - '8'? — No / Yes

If command='m' then i:=8 else i:=ord(command)-48

Set Tmr1 and Ccp1. Measuring the duration of ten concessive periods of sensor[9-i]. Store the result in the memory. i:=i-1

i=0 — No / Yes

Transmit the final results

*Figure 4 Flowchart of the program in the microcontroller.*

## Self-heating of temperature sensor

Each sensor needs some power, however little, which leads to an error in the temperature measurement. By switching off the power of the sensors that are not being sampled and by making sure the measurement duration of each sensor is as short as possible the effect of selfheating is very small. Even smaller then the resolution of the sensor; so it can be neglected.
Each of the eight sensors is only switched on for a period of maximum 10.5 ms. In figure 5 the timing diagram of the subsequent sensors is depicted. With all sensors mounted, each sensor is being powered about 10% of time and when fewer sensors are mounted this percentage will increase towards 35% (one sensor is mounted).
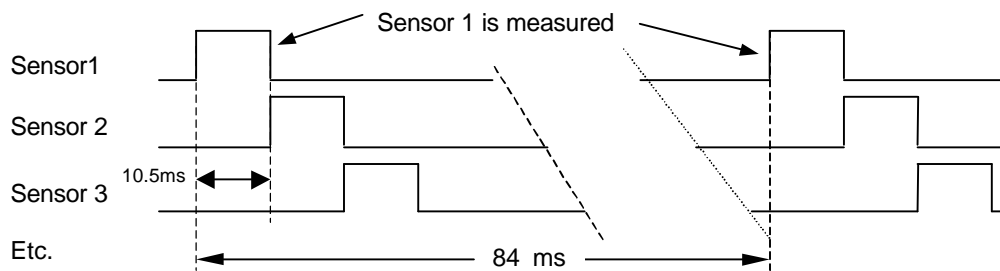
6

*Figure 5 Timing diagram of subsequent sensors. (times given in timing-diagram are typical)*

**Measurement accuracy**

The resolution (i.e. the accuray with which the sensor output duty cycle is determined), depends on the number of samples taken. With the choosen sampling speed (5 MHz) and duration (about 10.5 ms) a resolution of  0.015 °C is obtained and at the same time the self heating effect is minimized. The absolute accuracy of the all over measurement system still depends on the sensors (0,7 °C at best for the TO18).

## 3. Circuit diagram and connections to the board

### General
In figure 6 the circuit diagram of the SMTAS08 is given. The on-board voltage regulator provides the internal power supply voltage of 5 V. The SMTAS08 board requires a DC supply voltage in the range of 7 V to 18 V. The on-board LED will light in case the supply voltage is connected.
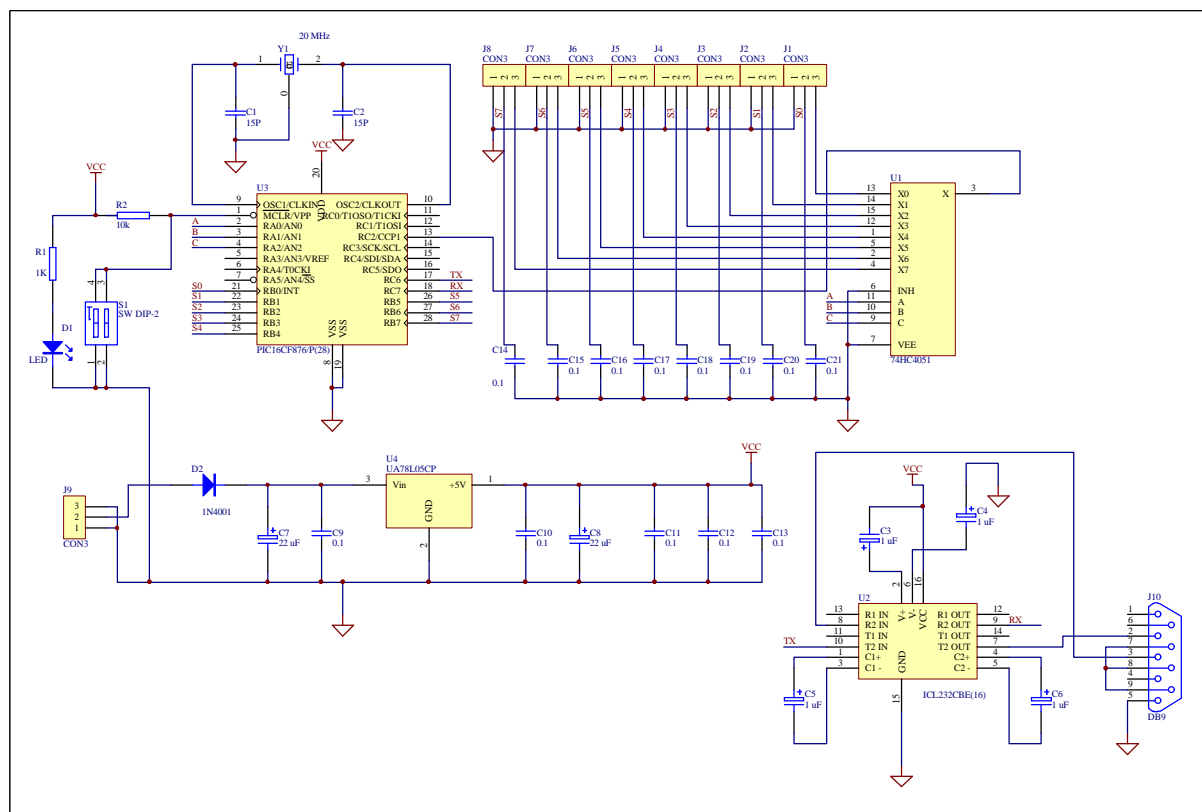If needed a reset switch can be mounted (SW DIP-2). This may be convenient when developing software by yourself.



*Figure 6 Circuit diagram of the SMTAS08 system.*

Smartec © 2002

**Connector layout**

Connection to the SMTAS08 system board is implemented with 10 connectors:
J1 ~ J8   3-pin header to connect 8 sensors.
J9         3-pin header for power supply.
J10       9-way D-type connector for communication with a PC (standard serial cable, straight)

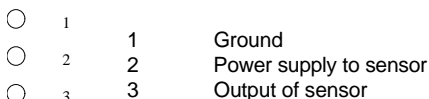Figures 7, 8 and 9 show the pin connection of connectors.

| | |
|---|---|
| 1 | Ground |
| 2 | Power supply to sensor |
| 3 | Output of sensor |

*Figure 7. The pin configuration of J1 ~ J8 (pitch 0.1 ").*

| | |
|---|---|
| 1 | Ground |
| 2 | External power supply of the board 7 – 15 volt DC |
| 3 | Ground |
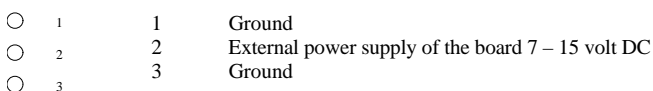
*Figure 8. The pin configuration J9 (pitch 0.1").*

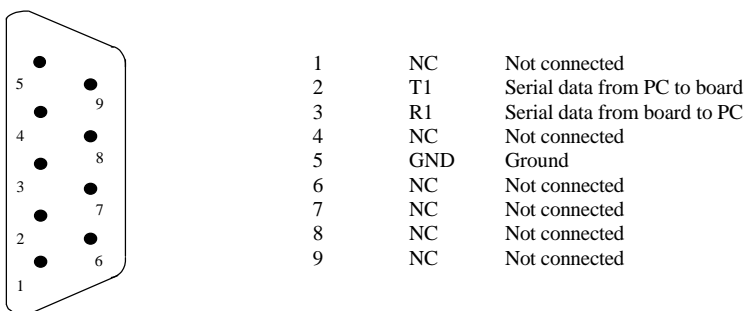| | | |
|---|---|---|
| 1 | NC | Not connected |
| 2 | T1 | Serial data from PC to board |
| 3 | R1 | Serial data from board to PC |
| 4 | NC | Not connected |
| 5 | GND | Ground |
| 6 | NC | Not connected |
| 7 | NC | Not connected |
| 8 | NC | Not connected |
| 9 | NC | Not connected |

*Figure 9. The pin configuration of  J10 (9 pole SUB-D female).*


# 5. SMTAS8 software

There are many ways to display results from the system. As an example, we will discuss the program **Terminal** (or **Hyper terminal**) under Windows, and the program **LABVIEW**.  Finally, an example is given in **Visual Basic** how to handle the incoming data of the serial port.

**Hyper terminal**

Configuration

The program **Terminal** (or **Hyper terminal**) under Windows can easily display the measurement results via the serial port of the PC. The configuration of **Terminal** (or **Hyper terminal**) is: baudrate 57600 bps, 8 data bit, 1 stop bit and No parity error check.
Once the communication between the microcontroller and the PC is actuated, by typing "h" to get on-line help. The following will be displayed:

Smartec © 2002

S M T A S 8   T E M P E R A T U R E
M E A S U R E M E N T   S Y S T E M
Version :


m -> Infinite measurement cycle
1 -> First sensor, single measurement
2 -> First two sensors, single measurement
3 -> First three sensors, single measurement
4 -> First four sensors, single measurement
5 -> First five sensors, single measurement
6 -> First six sensors, single measurement
7 -> First seven sensors, single measurement
8 -> First eight sensors, single measurement
s -> Stop


Please Make a Choice :



As soon as you press "m" or a digit between "1" and "8", the board will start sending data. The data format is Hexadecimal. For each sensor, both $t_1$ and $t_2$ are represented by four hexadecimal Ascii digits, separated by "  " (space) and followed by "/n" or  "CRLF" at the end. In case that there is no sensor connected to a selected position, or when a selected sensor does not transmit a period-modulated output signal, this will be detected by the microcontroller, which will send "FFFF"  for both $t1$ and $t2$.

Below you find a typical example of a terminal output. As you can see, sensor 8 is not connected.

```
1B36 3F92 1AEF  3ED2 1A6B  3D95 1AFA 3EC3  1BFB 413A 1B08 3EF9 19FB 3C7C  FFFF FFFF
1B34 3F89 1AF1   3ED3 1A65  3D8F 1AFB 3ECD 1BFA 4138 1B13 3F14  19FA 3C7E  FFFF FFFF
1B34 3F8E 1AED 3ECD 1A68  3D91 1AF6  3EC2 1BF7  4130 1B15 3F15  19F6  3C7A FFFF FFFF
```

Sensor 1    Sensor 2    Sensor 3    Sensor 4    Sensor 5    Sensor 6    Sensor 7    Sensor 8


*Figure 10. Typical example of a Terminal program output (Sensor 8 not connected)*.


To obtain the corresponding temperature, the values for $t_1$ and $t_2$ have to substituted in formula (1). A terminal program cannot do this calculation, but gives you a convenient way of displaying the board data. Included with the board is also a Labview program, which will do the calculations as well and display the actual temperatures.


**Executable LabView**

Contribution disk

The SMTAS08 development kit contains a contribution disk with the software to run the system with Labview in a PC. There are three files, named: SMTAS08.arj, ARJ.exe and SERPDRV, respectively, on the disk. The file SERPDRV is a serial-port-driver file. The file SMTAS08.arj is a freshened file of SMTAS08.exe, and ARJ.exe is a file manager.
To use the software, the user must have an IBM-compatible PC with Windows 95, or later, installed. The three files on the contribution disk have to be copied to a subdirectory, for instance, SMTAS08, in the hard disk of your PC. After that, the file SMTAS08.arj should be extracted by running ARJ.exe under DOS or Windows. To perform this task the command ***arj e SMATAS08.arj*** can be used.

When working under Windows and using the FILE MANAGER you can simply double click SMTAS08.EXE to start the program. When the program starts up, a front panel window appears. There are one output mode selection and for control items: ***NumSensor, Serial port No. Offset correction*** and ***Average Index***. Further, there are a few display items: ***Real-time value***, ***Std Deviation*** and ***Average value***.

Output mode selection

Using the output mode selection, ***OUTPUT MODE***, the measured temperatures can be displayed in Celcius, Fahrenheit or Kelvin.

Control items

The item ***Serial port No***.. allows the user to select the desired serial port of the PC.
The item ***NumSensor*** is used to select the number of sensor to be measured (maximum is 8).
The item ***Offset Correction*** is used to correct the tolerance of the sensors. For instance, to use 8 sensors to measure a temperature, there is a deviation from sensor to sensor due to the fabrication tolerance. Then, the average value of these 8 sensors at a certain same temperature is used to correct this deviation.
The item ***Average Index*** is used to set the number of measurements to be averaged. A large number of measurements results in lower noise, but also increases the measurement time.

Display items

The item ***Real-time value*** enables the numerical display of the measurement result. Meanwhile, a graphic chart displays the real-time value by a red line.

The item ***Average Value*** averaging over a variable number of measurements indicated by the ***Average Index***.

The item ***Std Deviation*** indicates the standard deviation of the system for a variable number (indicated by ***Average Index***) of measurements.

To Start Measurement

Double click the button ⇒ at the top-left corner of the front-panel window to start the measurement. After a few seconds the results are displayed at the monitor.

**Visual Basic.**

To be implemented

# 6.  Ordering information

SMTAS08                 Temperature Acquisition System for 8 Smartec temperature sensors.

SMT1603018             Smartec temperature sensor in TO18 encapsulation (metal can)
SMT1603092             Smartec temperature sensor in TO92 encapsulation (commercial)
SMT16030220           Smartec temperature sensor in TO220 encapsulation
SMT16030SO            Smartec temperature sensor in SOIC8
SMT16030HE            Smartec temperature sensor as small hybrid (2.5. x 8 mm)