MICROCONTROLLER AND SD-CARD BASED MULTICHANNEL DATA LOGGER

DOGAN IBRAHIM, LECTURER AT THE NEAR EAST UNIVERSITY IN CYPRUS, DESCRIBES THE DESIGN OF A MICROCONTROLLER-BASED MULTICHANNEL DATA LOGGER DEVICE WITH SD CARD AND REAL-TIME CLOCK INTERFACE

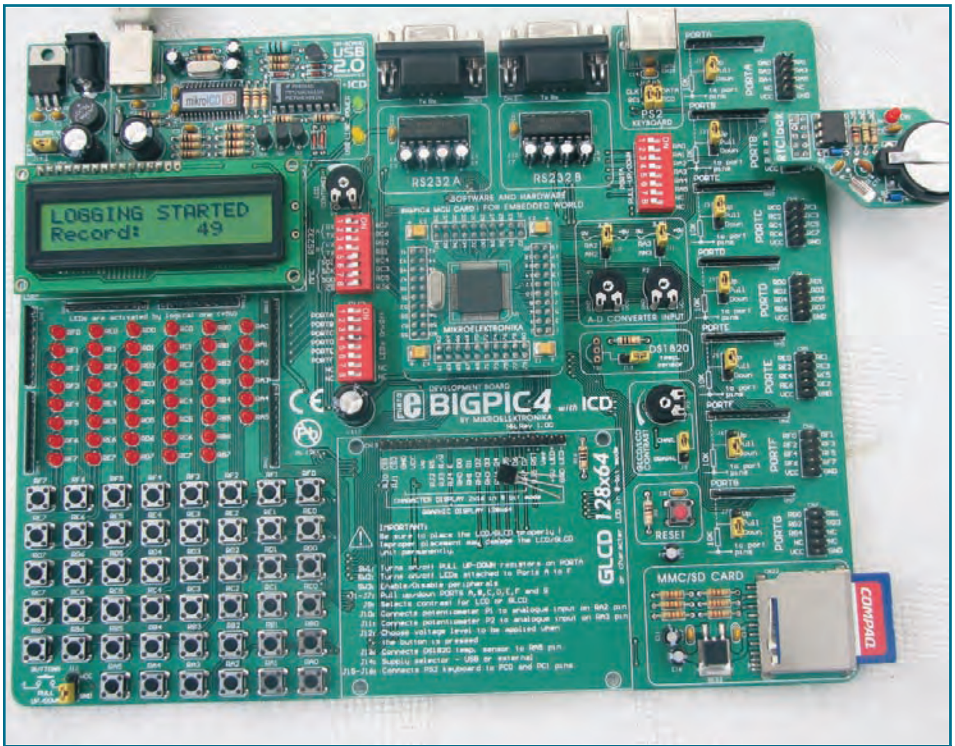

Figure 1: *BIGPIC4 development board with an RTC card*

The term 'data logging' can be defined as the capture and storage of data for use at a later time. Basically, a data logger is an electronic device that captures and records data over time.

Data loggers are nowadays based on the microcontroller technology. They are usually portable, battery-operated devices with internal storage and some incorporating sensors to measure physical quantities such as temperature, pressure, humidity, flow, displacement and so on.

Data loggers can be divided into two basic groups: standalone data loggers and data capturing data loggers.

STANDALONE DATA LOGGERS

This type of data loggers can be used on their own, without requiring other devices for data collection and storage. Standalone data loggers have large amounts of internal non-volatile memories. They may also have real time clock chips. The collected data can be saved in the memory with time stamping.

The data collected in a standalone data logger is usually analysed offline. A standalone data logger is usually configured and then left at the required site to collect data. At the end of the data collection period the device is connected to a PC and the collected data is read and analysed offline by the PC. Some standalone data loggers are dedicated for specific measurements, for example temperature data loggers.

The Thermo Recorder TR-5 Series (**www.tandd.com**) is a typical standalone temperature data logger. This data logger has LCD output, it can record up to 16,000 readings with time intervals from one second to one hour and the battery life is quoted as four years.

One of the disadvantages of standalone data loggers is that the devices should be checked at regular intervals to make sure that the memory is not full, or the battery is not flat. This may sometimes cause problems since the device may be located at a remote location or at a place not easily reachable.

DATA CAPTURING DATA LOGGERS

Data capturing data loggers are used only to capture the data. These devices do not have large internal memories and are normally connected to a PC. The captured data is sent to the PC for storage or for analysis. The data can either be analysed offline or online.

One of the disadvantages of data capturing data loggers is that the devices cannot be used on their own as another device (e.g. a PC) is required to store the captured data. The Pico Technology DrDAQ (**www.drdaq.com**) is a typical data capturing data logger that is connected to a PC to transfer the captured data. The device has built in sensors for light, sound and temperature measurements.

Some data capturing data loggers have wireless capabilities. Usually a transmitter-

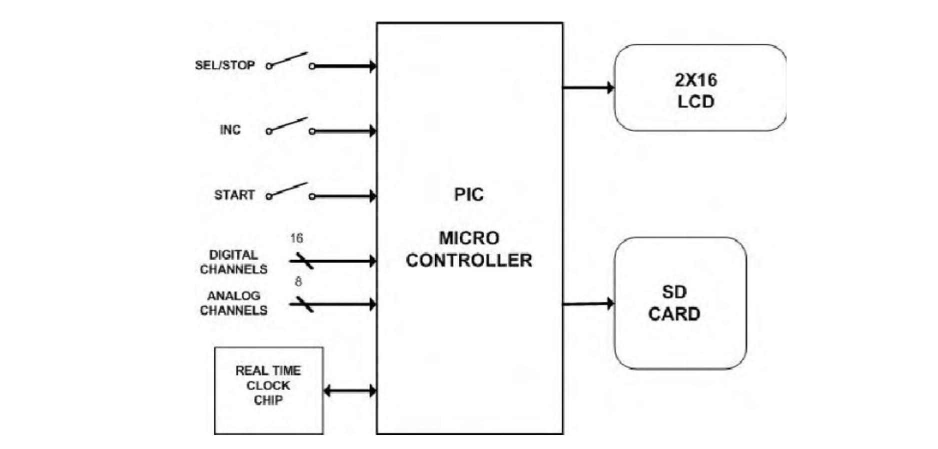

Figure 2: *Block diagram of the data logger*

receiver pair is used: the transmitter captures the data and sends it to the receiving device using wireless communication. The receiving device usually has large internal memory and stores the received data.

SPECIFICATION OF THE DESIGNED DATA LOGGER

The designed data logger has the following basic specifications:

- analogue channels and 16 digital channels
- LCD output
- Real time clock (RTC) chip
- Setting the RTC chip using buttons
- SD card to store the collected data
- Data stored in a file on the SD card
- Data is saved with time stamping
- Data is Excel-compatible
- Selectable logging interval
- Portable

THE CIRCUIT

The data logger was built on the BIGPIC4 development board, manufactured by MikroElektronika **(www.mikroe.com)**. The BIGPIC4 is a full-featured development board for PIC18 series of microcontrollers.The board allows PIC microcontrollers to be interfaced with external circuits and a broad range of peripheral devices, allowing the user to concentrate on software development.

The BIGPIC4 development board has the following specifications:

- External or USB power supply
- PIC18F8520 microcontroller (changeable)
- 36 buttons
- 36 LEDs
- Text-based LCD
- Graphics LCD
- 2 RS232 ports
- PS/2 port
- SD card holder and interface
- In-circuit debugger
- All port pins available on the board

In addition to the basic development board, a number of small external interface cards can be attached to the board, such as real-time clock card, keyboard card, sensor cards and so on. **Figure 1** shows the BIGPIC4 development board with the RTC card attached to PORT B (on the top right-hand side). The SD card holder is placed at the bottom right hand side of the board.

Although the data logger circuit was built and tested using a development board, it is possible to build the circuit on a breadboard or on a PCB using other types of PIC microcontrollers that have the required minimum number of I/O ports, data memory and program memory.

Figure 2 shows the block diagram of the data logger. The device has 24 channels. The captured data is stored on the SD card with time stamping, in a format suitable to be imported into Excel for offline analysis.

Figure 3a shows the circuit diagram of the data logger. This circuit is based on the PIC18F8520 microcontroller, but other PIC18 family members can also be used instead if required.

- Digital Channels: PORT D and PORT E pins are used as the 16 digital input channels.

- Analogue Channels: 5 pins of PORT A (RA0-RA3, and RA5) and 3 pins of PORT F (RF0-RF2) are used as the 8 analogue input channels.

- LCD: A 2x16 column LCD is used in the circuit. The LCD is connected to PORT H pins (RH2-RH7) of the microcontroller.

- RTC: A PCF8583 type RTC chip is used in the circuit. This chip has the I²C bus type interface. The clock (SCL) and data (SDA) pins are connected to PORT B pins RB3 and RB4 respectively via 10K pull-up resistors. These resistors are required for the proper operation of the I²C bus. The RTC chip is connected to a battery so that the date and time information are not lost when power is removed from the circuit. A 32768Hz crystal is used to provide timing pulses to the RTC chip.

- SD Card: The BIGPIG4 development board is equipped with an SD card holder. The SD card is used in the SPI mode and the interface between the microcontroller and the SD card is as follows: RC3 is connected to CLK input, RC4 is connected to DO output, RC5 is connected to DI input and RJ6 is connected to the CS input of the SD card.

The SD card requires 3.3V supply for its operation and this is obtained by using a MC33269DT-3.3 type 3.3V regulator. The input voltages on the inputs of the SD card must not be greater than 3.6V and this is not compatible with the outputs of the microcontroller. Potential divider resistors (2.2K and 3.3K) are used at the outputs of the microcontroller to reduce the SD card input voltages to less than 3.6V.

- Buttons: Three buttons are used to configure and setup the data logger: SEL/STOP is connected to pin RB0, INC is connected to pin RB1 and STOP is connected to pin RB6 of the microcontroller. The buttons are active LOW i.e. a button output is normally at logic HIGH and goes to logic LOW when the button is pressed.

- Power supply: As shown in **Figure 3b**, a 7805 type 5V regulator is used to provide power to the circuit. The circuit can be operated from a 9V battery in portable applications.

- Reset: A reset button is provided so that the microcontroller can be reset externally.

- Clock: A 10MHz crystal and two 22pF capacitors are used to provide timing pulses to the circuit.

OPERATION

Three buttons are used to control operation of the data logger: SEL/STOP, INC and START. The data logger operates in two modes: SETUP mode and LOG mode. Both modes are described below in some detail (see **Figure 4** on how the modes are selected).

- SETUP Mode: The SETUP mode is used to set the logging interval and the date and time. This mode is entered by resetting the microcontroller (or applying power) while holding down the SEL/STOP button. Releasing this button will display the logging INTERVAL and expect the user to select the required interval by pressing the INC button.

After selecting the interval, the user has the choice of either setting the date/time or exiting the SETUP mode. Pressing SEL/STOP twice exits the SETUP mode. Pressing INC enters the date/time setup mode, where the date and time are initially shown as: 01/01/08 12:00:00. The cursor is initially on the 'day' field and pressing the INC button increments this field. Pressing the SEL/STOP button moves the function between the date and time fields.

After setting the seconds field, press SEL/STOP button to exit the SETUP mode. The selected date and time will be updated every second and displayed on the LCD. Now it's all ready to start the data logging (LOG mode) process.

- LOG Mode: The LOG mode is entered by simply resetting the microcontroller, or by applying power to the circuit. This mode is also automatically entered at the end of the SETUP mode. When this mode is selected the current date and time are displayed on the LCD and are updated every second. Pressing the START button starts the data logger to collect data and store on the SD card.

The program checks and data collection does not start if the SD card is not inserted into its holder. During the data collection the current record number is displayed on the LCD as shows in the top left corner of Figure 1.

The data logger is stopped by pressing the SEL/STOP button while in the LOG mode. At this point the SD card can be removed safely from its holder.

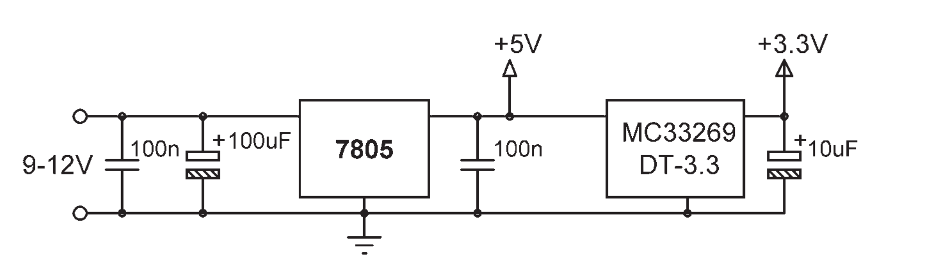

Figure 3b: *Power supply of the circuit*

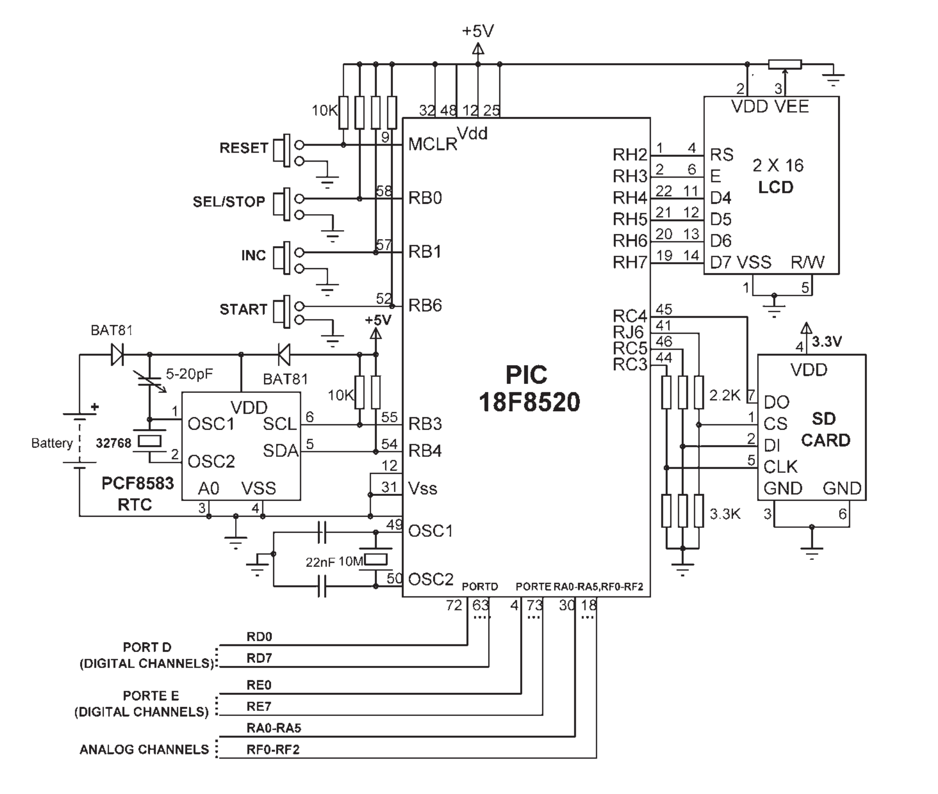

Figure 3a: *Circuit diagram of the data logger*

DATA FORMAT

The collected data is stored in hexadecimal ASCII format with the time stamp, where each record occupies one line. The format of a record is as follows: *dd/mm/yy hh:mm:ss PORTD PORTE A1 A2 A3 A4 A5 A6 A7 A8 <cr><lf>*
where PORTD and PORTE are the digital input data, A1 to A8 are the analogue input data and *cr* and *lf* are the carriage-return and line-feed characters respectively. The digital data is 8-bits wide and is represented by two hexadecimal digits. The analogue data is 10-bits wide and is represented by three hexadecimal characters.

An example record is given below:
12/07/08 10:00:20 FF FE 1FE 1FF 000 000 000 000 000 000 <cr><lf>

In this example, the data was collected on the 12th of July, 2008 at 20 seconds past 10 o'clock. PORT D data was FF (binary: "11111111"), PORT E data was FE (binary: "11111110"). Analogue channel 1 data was 1FE (binary: "0111111110"), analogue channel 2 data was 1FF (binary: "0111111111"), and so on.

The data fields are separated by spaces and are compatible with the Excel spreadsheet. Thus, the collected data can easily be imported into Excel and then analysed statistically. Post-processing can be done on the data and for example graphs can be drawn to show the variation of the data with time.

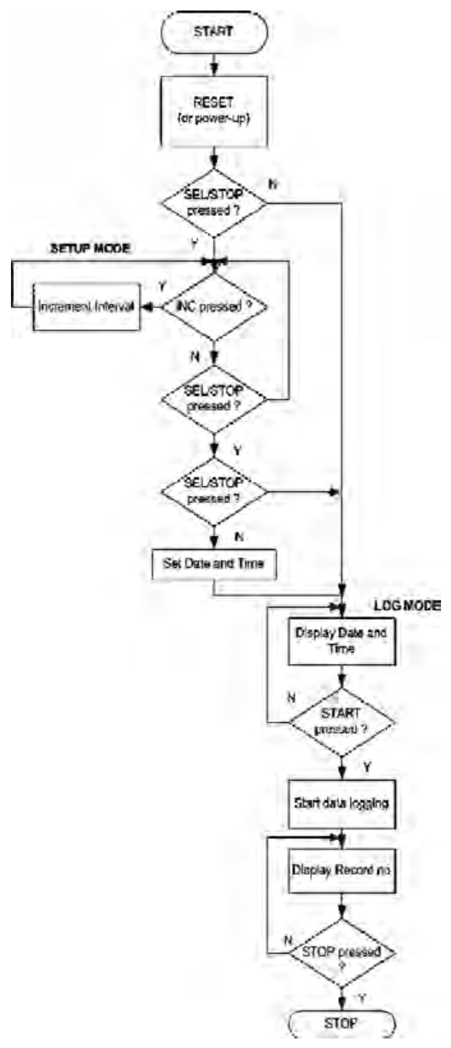

***Figure 4:** Selecting the operation modes*

```
BEGIN
    RESET (or power-up)
    IF SEL/STOP pressed THEN
        Enter SETUP mode
            WHILE SEL/STOP not pressed
                Display Interval
                IF INC is pressed THEN Increment Interval
            WEND
            WAIT UNTIL SEL/STOP OR INC pressed
            IF INC pressed THEN
                Move to DAY field
                WHILE SEL/STOP not pressed
                    IF INC pressed THEN increment DAY
                WEND
                Move to MONTH field
                WHILE SEL/STOP not pressed
                    IF INC pressed THEN increment MONTH
                WEND
                Move to YEAR field
                WHILE SEL/STOP not pressed
                    IF INC pressed THEN increment YEAR
                WEND
                Move to HOUR field
                WHILE SEL/STOP not pressed
                    IF INC pressed THEN increment HOUR
                WEND
                Move to MINUTE field
                WHILE SEL/STOP not pressed
                    IF INC pressed THEN increment MINUTE
                WEND
                Move to SECOND field
                WHILE SEL/STOP not pressed
                    IF INC pressed THEN increment SECOND
                WEND
            ELSE
    ELSE
        Enter LOG Mode
        WHILE START not pressed
            Display Date and Time
        WEND
        DO FOREVER
            Read and store analogue and digital data on SD card
            Display record number
            IF STOP pressed THEN
                STOP
            ENDIF
        ENDDO
    ENDIF
END
```

***Figure 5:** Operation of the software*

THE SOFTWARE

The software of the data logger is based on the mikroC language, developed by mikroElektronika for PIC18 series of microcontrollers. A 2K limited version of the compiler is available free of charge from the developing company. The main reasons for choosing this compiler are that:

- mikroC is a very sophisticated C language compiler;
- It is compatible with the BIGPIC4 development board;
- mikroC supports SD card functions which makes the development of SD card based projects relatively easy;
- mikroC supports software, as well as hardware based I^2C bus functions. The SD card operates in SPI mode and uses PORT C of the microcontroller. RTC chip is based on I^2C bus, which also uses PORT C by default. It was necessary to connect the RTC chip to another port of the microcontroller and use the I^2C bus software functions.

The program has been developed to be modular where procedures and functions are used wherever possible to carry out the required tasks. **Figure 5** shows operation of the software using simple PDL. At the beginning of the program various peripheral devices such as the LCD, SD card, the RTC chip and the I/O ports are initialised. The program then checks whether to enter the SETUP mode or the LOG mode; mikroC function *Button* is used to check the state of the buttons. In SETUP mode the logging interval is set and date/time can be set to the correct values. Function *Set_RTC* is used to implement the SETUP mode. The RTC chip is controlled using the mikroC *Soft_I²C* functions.

Initially, the *Soft_I²C_Start* function is called to start communication on the I²C bus. Then functions *Soft_I²C_Read* and *Soft_I²C_Write* are used to read and write to the RTC chip respectively. The LCD is operated in 4-bit mode where only the high 4-bits of the data bus are used. Data is written to the LCD using mikroC functions *Lcd_Out* and *Lcd_Out_Cp*.

WRITING AND READING

Writing to and reading from the SD card are very easy with the mikroC, as the compiler supports a large number of functions for direct sector based, or FAT-16 based read and write operations.

After initialising the SD card library, function *Mmc_Fat_Assign* is used to specify the filename to be used on the SD card. If a new file is to be opened, then function *Mmc_Rewrite* is called to clear the file and position the file pointer to the beginning of the file. If data is to be appended to the end of the existing file, then *Mmc_Fat_Append* function is called. Data is then written to the specified file on the SD card using function *Mmc_Fat_Write*. The SEL/STOP button is checked continually and when this button is pressed the program stops, displaying a message on the LCD. The SD card then can be taken out safely for offline analysis on a PC.

Function *Read_RTC* reads the date and time from the RTC chip using the soft I²C bus functions and stores the data in global variables *RTCDate* and *RTCTime* respectively. It is important to realise that the RTC chip expects the data to be in BCD format and the data should be converted into this format before writing to the chip. Similarly, the data is read in BCD format and should be converted to the required format before being used.

The function *Display_RTC* displays the RTC data on the LCD with row 1 displaying the date and row 2 displaying

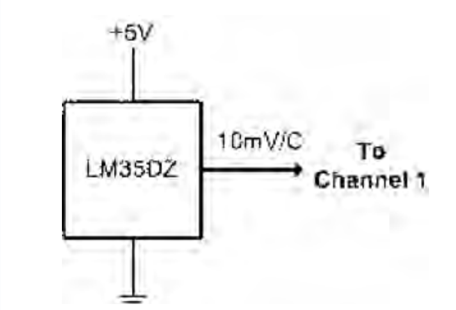

Figure 6: *Connecting the LM35DZ sensor to analogue channel 1*

the time. The function *convert_to_hex* converts a given byte into two hexadecimal digits. This function is used to convert the channel data into hexadecimal format before storing on the SD card.

EXAMPLE DATA LOGGING

An example is given here to illustrate the operation and output file of the data logger. In this example, an LM35DZ type analogue temperature sensor is connected to analogue channel 1 of the data logger (see **Figure 6**). Data is collected every two seconds (Interval = 2) for about 30 seconds.

The data is saved on the SD card in a file called DTLOGGER.TXT. **Figure 7** shows the created file, opened with the WORD program on Windows.

Note that the first two columns show the date and the temperature respectively. Column 5 shows the Channel 1 data (i.e. the temperature) in hexadecimal format. The data is 10-bits wide, having 1024 quantisation levels. Thus, with a reference voltage of +5V, each level corresponds to 5000/1024mV. The LM35DZ sensor output is 10°V/mV. Therefore, the temperature in °C can be calculated as:

$$Temperature = T * 500/1024$$

where, is the value read from Channel 1. For example, the first value read is hexadecimal 03A which has the decimal equivalent of 58. This corresponds to a temperature of 28.3°C.

IMPORTING TO EXCEL

The steps to import the data into Excel and display the change of temperature with time are given below:

- Start the Excel spreadsheet application
- Add Analysis Toolpak: *Tools -> Add-Ins -> Analysis ToolPak -> Ok*
- Import collected data: *File -> Open -> DTLOGGER.TXT -> Text Import Wizard -> Finish*
- You should now have the collected data in Excel. Select all the fields and click the *Text Center* tool to centre the fields in the worksheet.
- Click on Cell E13 and enter the following formula to convert the hexadecimal data in column 5, row 1, into absolute temperature in °C:

= 500*(HEX2DEC(E1))/1024

- Now convert all the entries in column 5 into absolute temperature. Copy cell at E13, then Paste it to cells E14 to E22.
- Format the data so that there are 2

```
06/01/08 09:50:13 FF FF 03A 02F 034 038 064 000 000 000
06/01/08 09:50:15 FF FF 03A 01E 016 01D 04A 000 000 000
06/01/08 09:50:17 FF FF 03A 02E 031 039 06A 000 000 000
06/01/08 09:50:19 FF FF 03B 02A 023 030 061 000 000 000
06/01/08 09:50:21 FF FF 03C 02C 026 032 063 000 000 000
06/01/08 09:50:23 FF FF 03E 030 02E 038 068 000 000 000
06/01/08 09:50:25 FF FF 040 026 020 026 057 000 000 000
06/01/08 09:50:27 FF FF 041 02D 02A 030 060 000 000 000
06/01/08 09:50:29 FF FF 043 02C 029 02D 05D 000 000 000
06/01/08 09:50:31 FF FF 048 02E 02B 032 064 000 000 000
```

Figure 7: *File DTLOGGER.TXT with the collected data*

	A	B	C	D	E	F	G	H	I	J	K	L
1	06.01.2008	09:50:13	FF	FF	03A	02F	34	38	64	0	0	0
2	06.01.2008	09:50:15	FF	FF	03A	01E	16	01D	04A	0	0	0
3	06.01.2008	09:50:17	FF	FF	03A	02E	31	39	06A	0	0	0
4	06.01.2008	09:50:19	FF	FF	03A	02A	23	30	61	0	0	0
5	06.01.2008	09:50:21	FF	FF	03C	02C	26	32	63	0	0	0
6	06.01.2008	09:50:23	FF	FF	03E	30	02E	38	68	0	0	0
7	06.01.2008	09:50:25	FF	FF	40	26	20	26	57	0	0	0
8	06.01.2008	09:50:27	FF	FF	41	02D	02A	30	60	0	0	0
9	06.01.2008	09:50:29	FF	FF	43	02C	29	02D	05D	0	0	0
10	06.01.2008	09:50:31	FF	FF	48	02E	02B	32	64	0	0	0
11												
12												
13					28,32							
14					28,32							
15					28,32							
16					28,32							
17					29,30							
18					30,27							
19					31,25							
20					31,74							
21					32,71							

***Figure 8:** Column 5 stores the temperature in °C*

	A	B	C	D	E	F	G	H	I	J
1	06.01.2008	09:50:13	FF	FF	03A	02F	34	38	64	0
2	06.01.2008	09:50:15	FF	FF	03A	01E	16	01D	04A	0
3	06.01.2008	09:50:17	FF	FF	03A	02E	31	39	06A	0
4	06.01.2008	09:50:19	FF	FF	03A	02A	23	30	61	0
5	06.01.2008	09:50:21	FF	FF	03C	02C	26	32	63	0
6	06.01.2008	09:50:23	FF	FF	03E	30	02E	38	68	0
7	06.01.2008	09:50:25	FF	FF	40	26	20	26	57	0
8	06.01.2008	09:50:27	FF	FF	41	02D	02A	30	60	0
9	06.01.2008	09:50:29	FF	FF	43	02C	29	02D	05D	0
10	06.01.2008	09:50:31	FF	FF	48	02E	02B	32	64	0
11										
12										
13		09:50:13			28,32					
14		09:50:15			28,32					
15		09:50:17			28,32					
16		09:50:19			28,32					
17		09:50:21			29,30					
18		09:50:23			30,27					
19		09:50:25			31,25					
20		09:50:27			31,74					
21		09:50:29			32,71					
22		09:50:31			35,16					

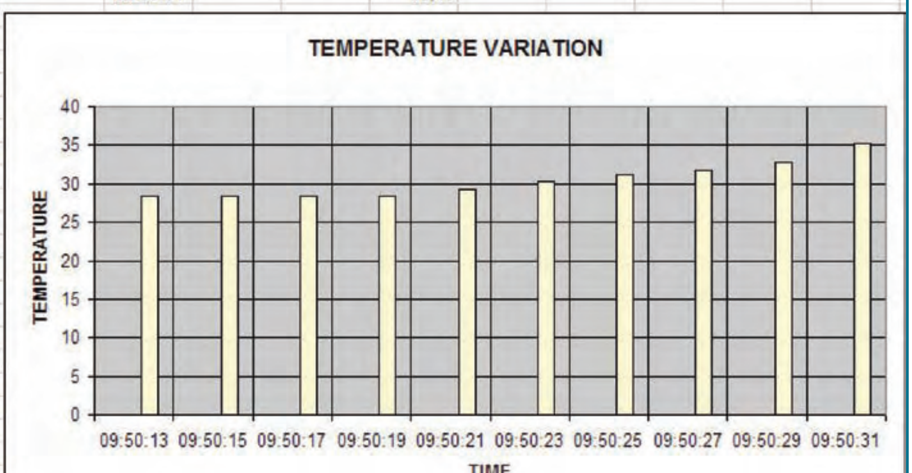

***Figure 9:** Graph of temperature variation*

digits after the decimal point, and select dot for the decimal point. *Format -> Cells -> Number -> Decimal Places = 2 -> Ok.* Click *Use 1000 Separator.*

- Your Excel worksheet should now look as in **Figure 8**.

DRAWING A GRAPH

We can now draw a graph in Excel to show the change of temperature with absolute time:

- Copy and paste the time fields from B1:B10 to B13:B22.
- Select and highlight the time and temperature fields (B13 to E22).
- Click on *Chart Wizard* icon. The default graph type is vertical bar-chart. Click on *Finish* to draw the graph.
- Enter graph title, x-axis, y-axis and gridlines: *Chart -> Chart Options -> Chart title = TEMPERATURE VARIATION. Category (X) axis = TIME. Value (Y) axis = TEMPERATURE. Click Gridlines -> Major gridlines -> Ok*

The graph shown in **Figure 9** will be drawn to show the variation of temperature with time.

FURTHER ENHANCEMENTS

The data logger described here can have many educational, commercial and industrial applications. The design can be improved further by incorporating the following modifications to the hardware and software:

- A communications interface (e.g USB port or RS232 interface) can be added to the hardware so that the collected data can be sent directly to a PC.
- Input signal conditioning circuits can be added to the analogue channels.
- The software can be modified by introducing more than one logging interval so that, for example, one channel of the A/D can be sampled every second, while another channel can be samples every 10 seconds.
- The hardware and software can be modified by introducing triggering so that a channel data is only read after it is triggered by an external or an internal event.
- A graphics LCD display can be added to the data logger so that a selected channel data can be displayed dynamically on the LCD in real time.
- More digital or analogue channels can easily be added to the data logger. ■