Chapter 5
A Development of Data-Logger for Indoor Environment

Anuj Kumar, I.P. Singh, and S.K. Sud

Abstract This chapter describes a development of data logger for indoor environment. Present work concentrates to environmental parameter (temperature and humidity) and more polluted contaminants (concentration level of CO and CO_2). In this work four channels have been used for data logger and other four channels is open to external sensor module. The data collected will be stored in the EEPROM and output can be taken in note-pad in tabular corresponding to month/date/year using graphical user interface.

1 Introduction

Environment monitoring system is a complete data logging system. It automatically measures and records temperature, humidity and other parameters and provides warnings when readings go out of range [1]. Indoor environment monitoring is required to protect the building occupant's health by providing thermally comfortable and toxicant free environment [2]. We need a system that monitors as well as records the indoor environment data easily. In this chapter, we are discussing about the developed indoor environment monitoring system that monitors and records indoor temperature, humidity, CO and CO_2.

Available technologies for sensing these environmental parameters are developed by Onseat HOBO, Spectrum Technologies (Watch Dog weather conditions), TandD, Telaire (Wireless Monitoring Systems), Testo, Log Tag, Measurement Computing Corporation, Monarch Instruments, MSR Electronics GmbH, P3 International, Quality Thermistor, S3 Crop, Sensaphone, Sansatronics, Lascar, ICP, Graphtech, Extech Instruments, Dickson, Dent Instruments, Davis, ACR System Inc, 3M International,

A. Kumar (✉)
Instrument Design Development Centre, Indian Institute of Technology Delhi, Electronics Lab, New Delhi, India 110016
e-mail: anujkumariitd@gmail.com

S.-I. Ao et al. (eds.), *Machine Learning and Systems Engineering*, Lecture Notes in Electrical Engineering 68,
DOI 10.1007/978-90-481-9419-3_5,

and Acumen [3]. The drawback of these available systems is that, both air quality and thermal comfort can not be measured simultaneously. So there is a need to develop an economical system (prototype data logger) which can help in collecting data to analyze the necessary environmental parameters simultaneously.

A prototype data logger has been developed to monitor the environmental parameters. The developed data logger consist (i) sensors module (ii) LCD (iii) Real Time Clock (RTC) (iv) EEPROM and (v) PC serial communication. This data logger is operated through PC using graphical user interface (GUI) in visual basic.

2 Sensors Module

A sensor is a device that measures a physical quantity and converts it into an equivalent analog or digital signal which can be read by an observer or by an instrument [4]. We have used temperature, relative humidity, CO, and CO_2 sensors in the developed system.

A gas sensor detects particular gas molecules and produces an electrical signal whose magnitude is proportional to the concentration of the gas [5]. Till date, no gas sensor exists that is 100% selective to only a single gas. A good sensor is sensitive to the measured quantity but less sensitive to other quantities. Available gas sensors are based on five basic principles. These can be electrochemical, infrared, catalytic bead, photo ionization and solid-state [6, 7]. We have selected these sensors because they produce a strong signal for the selected variable especially at high gas concentrations with adequate sensitivity. They have a fast response time, high stability, long life, low cost, low dependency on humidity, low power consumption, and compact size [5]. Four sensors along with their signal conditioning circuit are used to sense the desired parameter such as temperature, humidity, CO, and CO_2. Signal conditioning circuit for that sensor needs to be connected externally. In software we can select any of the analog channels. The interface of temperature, humidity, CO, and CO_2 sensors with microcontroller PIC 18F4458 is described as follows.

2.1 Temperature Sensor

National semiconductor's LM 35 IC has been used for sensing the temperature. It is an integrated circuit sensor that can be used to measure temperature with an electrical output proportional to the temperature (in °C). The temperature can be measured more accurately with it than thermistor. The operating and amplification circuit is shown in Fig. 1. The output voltage of IC LM 35 is converted to temperature in °C is given by the following expression [7]

$$Temp.(^{\circ}\mathrm{C}) = (V_{out} \times 100)/7^{\circ}\mathrm{C}$$

The measuring temperature range of the instrument is between 15°C and 70°C.

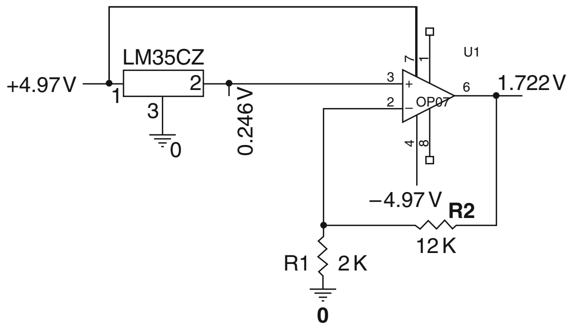

Fig. 1 Operating and amplification circuitof temperature sensor

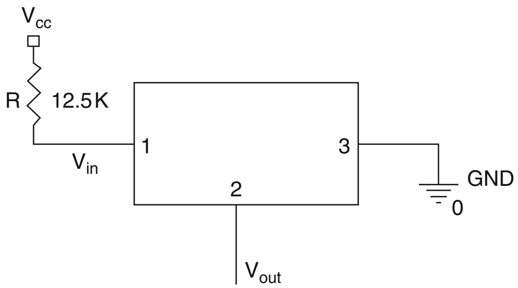

Fig. 2 Operating circuit of humidity sensor

2.2 Humidity Sensor

The sensor circuit develops a linear voltage vs. RH (relative humidity) output, which is ratio metric to the supply voltage. This means when the supply voltage varies, the sensor output voltage follows the same proportion. It can operate over a range of 4–5.8 V supply. At 5 V supply voltage (at room temperature), corresponding to relative humidity variation from 0% to 100% (noncondensing), the output voltage varies from 0.8 to 3.9 V. The humidity sensor functions with a resolution of up to 0.5% of relative humidity (RH), with a typical current draw of only 200 μA, the HIH4000 series is ideally suited for low drain, battery operated systems.

The operating circuit is shown in Fig. 2. The change in the RH of the surroundings causes an equivalent change in the voltage output. The output is an analog voltage proportional to the supply voltage. Consequently, converting it to relative humidity (RH) requires both the supply and the sensor output voltages (At 25°C) and is given by the following expression [7].

$$RH = \left(\left(V_{out}/V_{\sup ply}\right) - 0.16\right)/0.0062$$

The output of the humidity sensor is 2.548 V i.e. the relative humidity is 56% at 25°C.

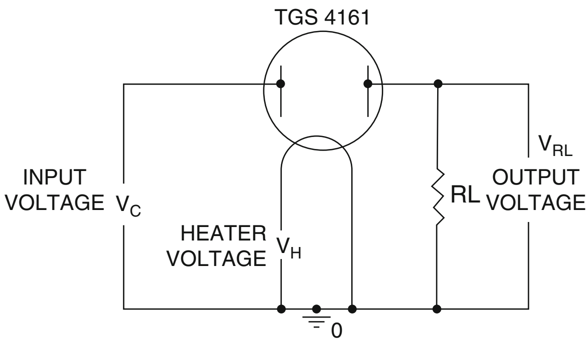

Fig. 3 Circuit for CO and CO_2 sensor [7]

2.3 *CO and CO_2 Sensor*

The operating circuit of CO and CO_2 sensors is shown in Fig. 3. The relationship between output voltage and gas concentration is given by the following expression

$$c = \left[\left(\frac{(V_C R_L / V_{OUT}) - R_L}{R_0} - 1 \right) \frac{1}{K} \right]^2$$

where, V_{OUT} = output voltage; V_C = input voltage, R_0 = electrical resistance of sensor at zero ppm, K = a constant for particular, R_L = sensor load resistance [5, 7].

3 LCD Interface to the Microcontroller

In this work, we are using on-chip analog to digital converter which is on the microcontroller. This analog to digital converter is having the 12 bit resolution with programmable acquisition time. It is sensing the analog signal from the sensor at the variable sampling rate (1 s to 1 h). The sensed value is converted to its digital equivalent. This digital value is displayed on the LCD (liquid crystal display) and is interfaced to the microcontroller [8, 9–12].

4 Real Time Clock Interface to the Microcontroller

The IC DS1307 operates as a slave device on the I^2C bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When V_{CC} falls below 1.25 V_{BAT}, the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out of tolerance system. When V_{CC} falls below V_{BAT}, the device switches into a low-current battery backup mode. Upon power up, the device switches from battery to V_{CC} when V_{CC} is greater than V_{BAT} +0.2 V and recognizes inputs when V_{CC} is greater than 1.25 V_{BAT}. We are using IC DS1307 as real time clock which have features such as real-time clock counts in seconds, minutes, hours, day of month, day of week, month, and year with leap year compensation valid up to 2100, 56 Byte, Nonvolatile (NV) RAM for data storage, I^2C serial interface, programmable square wave output signal, automatic power fail detect and switch circuitry (consumes less than 500 nA in battery backup mode oscillator running), and temperature range −40°C to 85°C [9, 10]. We are using I^2C to interface RTC and EEPROM to the micro-controller. The I^2C bus is the most popular of three serial EEPROM protocols. The I^2C chips include address pins as an easy way to have multiple chips on a single bus while only using two connections to the microcontroller [9].

5 EEPROM Interface to the Microcontroller

The EEPROM will store the digital value which is coming from analog to digital converter. We will require 52.73 MB of EEPROM if we are sampling all analog channels at the rate of 1 sample/s. We are using the EEPROM AT24C256 (ATMEL). This will store the sample data at different instants [10–14].

6 PC Interface Using RS-232 Serial Communication

PIC 18F4458 using MAX-232 is interfaced with PC. IC (MAX-232) used to convert TTL logic level to RS-232 logic level. RS-232 is the serial communication protocol that does not require the clock along with data lines. Two data lines are there one is T_X and another is R_X for serial communication. MAX-432 has two receivers (converts RS-232 logic level to TTL logic) and two drivers. Separate power supply has been provided because minimum power supply needed is 5 V and MAX-232 consumes a lot of current for operation. External capacitors are required for internal voltage pump to convert TTL logic level to RS-232 level. For battery operated application MAX-232 can be used as level converter instead of MAX-232. It is low supply low power consumption logic converter IC for RS-232 [9, 10, 13].

7 Graphical User Interface

The GUI is one of the important parts for this device as it displays the data from microcontroller for data monitoring and analysis. The design template has to be user friendly for best usage. For this chapter, the main objective is to display data received in graphical form. As transducer detects and translate an analog signal, the data will go through a conversion at the ADC. This digital data will be stored in EEPROM chip with the help of Visual Basic 6.0 software. Since the data is using serial RS232 communication, an initialization needs to be done having baud rate, data bits, parity, stop bit, and the COM port at PC. The baud rate is the number of signal changes per second or transition speed between Mark (negative) and Space (positive) which ranges from 110 to 19,200, data bits is the length of data in bit which has one Least Significant Bit and one Most Significant Bit, the parity bit is an optional bit mainly for bit error checking. It can be odd, even, none Mark, and Space. Stop bit is used to frame up the data bits and usually combined with the start bit. These bits are always represented by a negative voltage and can be 1, 1.5 and 2 stop bits, and COM port is the selection of the available COM port at PC. The commonly used setting to establish a serial RS232 communication is 9600 baud rate, none parity, 8 data bits, 1 stop bit, and COM port 1. This can be done by using the GUI monitoring system where it automatically saves the data received in a notepad. The data saved is the date and time at which the data collected and the data value it self. Figures 4 and 5, represents the graphical user interface and logged data in file respectively [12, 13].

8 Schematic of the Data Logger

Figure 6 shows, the full schematic diagram of the data logger for indoor environment. This data logger has four embedded sensor module and other four channels are open to be used for the measurement of other environmental parameters.

9 Software Design of Data Logger

This section includes the discussion on software design for all the modules interfaced with PIC 18F4458. It also explains the functions of software designed for data logger [11].

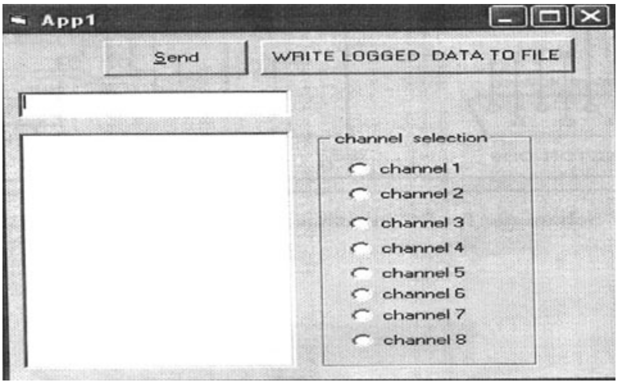

Fig. 4 GUI for the data logger

SET_1 - Notepad

File Edit Format View Help

Date Time	Temperature (°F) (1)	Temperature (°C) (1)	RH (%)	CO (ppm)	CO2 (ppm)
06/01/09 06:00:00.C	89.48	31.93	53	3.2	418
06/01/09 06:10:00.C	89.48	31.93	53.3	3.2	419
06/01/09 06:20:00.C	89.48	31.93	53.5	3.2	418
06/01/09 06:30:00.C	89.48	31.93	53.8	3.2	418
06/01/09 06:40:00.C	89.48	31.93	53.8	3.1	419
06/01/09 06:50:00.C	89.48	31.93	53.8	3.1	420
06/01/09 07:00:00.C	89.48	31.93	53.8	3.1	420
06/01/09 07:10:00.C	89.48	31.93	53.8	3.3	420
06/01/09 07:20:00.C	89.48	31.93	53.8	3.3	420
06/01/09 07:30:00.C	89.48	31.93	54	3.4	421
06/01/09 07:40:00.C	89.48	31.93	53.8	3.4	421
06/01/09 07:50:00.C	89.48	31.93	54.3	3.4	421
06/01/09 08:00:00.C	89.48	31.93	54.3	3.4	421
06/01/09 08:10:00.C	89.48	31.93	54.5	3.2	422
06/01/09 08:20:00.C	89.48	31.93	54.5	3.2	422
06/01/09 08:30:00.C	89.48	31.93	54.5	3.2	422
06/01/09 08:40:00.C	89.48	31.93	54.3	3.4	423
06/01/09 08:50:00.C	89.48	31.93	54.3	3.4	423
06/01/09 09:00:00.C	89.48	31.93	53.8	3.5	423
06/01/09 09:10:00.C	89.48	31.93	54.8	3.5	424
06/01/09 09:20:00.C	88.74	31.52	54.9	3.5	424
06/01/09 09:30:00.C	88.74	31.52	53.1	3.5	425
06/01/09 09:40:00.C	88.74	31.52	52	3.4	425
06/01/09 09:50:00.C	88.74	31.52	51.3	3.4	428
06/01/09 10:00:00.C	88.74	31.52	50.3	3.4	429
06/01/09 10:10:00.C	89.48	31.93	49.4	3.4	428
06/01/09 10:20:00.C	89.48	31.93	48.7	3.2	429
06/01/09 10:30:00.C	89.48	31.93	49	3.2	430
06/01/09 10:40:00.C	89.48	31.93	48.5	3.2	430
06/01/09 10:50:00.C	89.48	31.93	47.9	3.3	430
06/01/09 11:00:00.C	89.48	31.93	47.6	3.4	431
06/01/09 11:10:00.C	89.48	31.93	46.9	3.4	431

Fig. 5 Representations of the logged data in file

9.1 Programming Steps for I^2C Interface

I^2C interface is bi-directional. This is implemented by an "Acknowledge" or "ACK" system allows data to be sent in one direction to one item on the I^2C bus, than, that item will "ACK" to indicate the data received. Normally, the master device controls the clock line, SCL. This line dictates the timing of all transfers on the I^2C bus. Other devices can manipulate this line, but they can only force the line low. This action means that item on the bus cannot deal with more data in to any device.

9.1.1 Writing to an I^2C Chip

The function of writing to the EEPROM is shown here as "Control IN", which represents putting the EEPROM in an "input" mode. Since we are only sending data to the EEPROM (as shown in Fig. 7), we use "Control IN" byte and later used "Control OUT". Next, the EEPROM acknowledges this byte. This is shown by the "A" after the byte. It is put on the next line to indicate that this is transmitted by the EEPROM. The Address Byte contains the address of the location of the EEPROM where we want to write data. Since the address is valid, the data is acknowledged by the EEPROM. Finally, we send the data we want to write. The data is then acknowledged by the EEPROM. When that finishes, we send a stop condition to complete the transfer. Remember the "STOP" is represented as the "T" block on the end. Once the EEPROM gets the "STOP" condition it will begin writing to its memory.

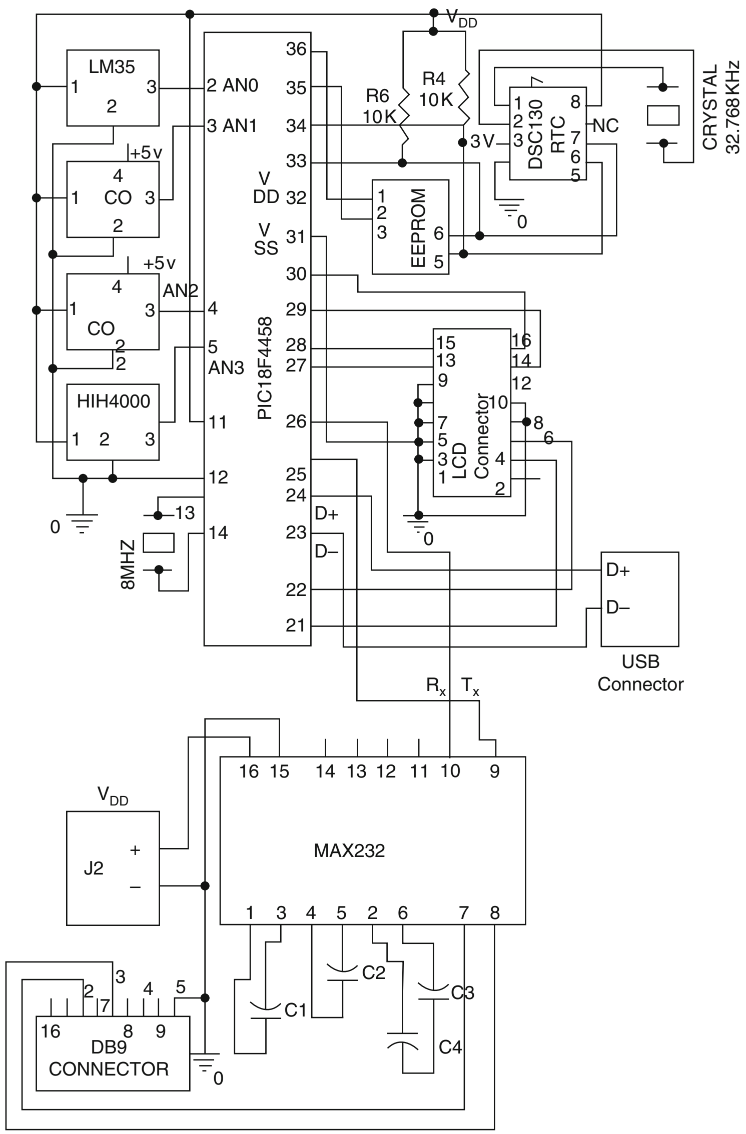

Fig. 6 Full Schematic of the data logger

9.1.2 Reading from an I^{2}C Chip

The transfer will use the “Control IN” byte to load the address into the EEPROM (as shown in Fig. 8). This sends data to the EEPROM which is why we use the control in byte. Once the address is loaded, we want to retrieve the data. So, we send a “Control OUT” byte to indicate to the EEPROM that we want data FROM it.

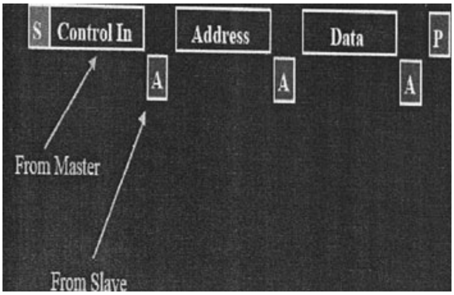

Fig. 7 Writing the data in I^{2}C chip (Controlling Window)

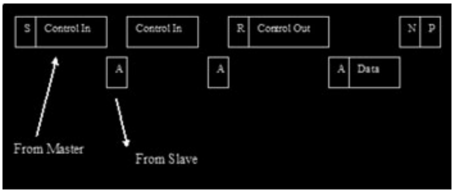

Fig. 8 Reading the data from an I^{2}C chip

The EEPROM will acknowledge this and then send the data we requested. When we are done getting data, we send a "NACK" to tell the EEPROM that we do not want more data. If we were to send an ACK at this point, we could get the next byte of data from the EEPROM. Since we only want to read one byte, we send a "NACK".

9.2 Programming Steps for LCD Interface

Set RS = 0 to send command; Send 0b0010 to data lines three times with a delay of 2 ms; to send a byte on four data lines, send higher nibble first and give a RE pulse of 100 µs at RE; send a set of instruction one after another with a delay of 2 ms between each command to configure various setting as given in instruction set of LCD datasheet [9, 13]; send instruction set again.

Set RS = 1; Send higher nibble at four data lines. Send 100 µs RE pulse; Send lower nibble at data lines. Send RE pulse; Keep track of number of character already displayed on display panel using LCD_count. Go to line 2 or line 1 according to that.

9.3 Programming Steps for Sensor Data Collection

There are four sensor module connected such as temperature, humidity, CO, and CO_2. Data is collected by the ADC inbuilt in PIC. ADC provides 12 bit of data after the conversion is completed.

9.3.1 Temperature Sensor Data Collection

Data collection from the temperature sensor needs following actions to be carried out (a) Selecting the analog channel AN_0, sampling frequency, and alignment of bits for ADRESH and ADRESL, (b) Vref and power on the ADC module by setting $ADCON_0$, $ADCON_1$ and $ADCON_2$ registers, (c) starting analog to digital conversion by setting ADGO bit high (wait till ADIF flag will not indicate the completion of conversion), and (d) copy of results from ADRESH and ADRESL to variables.

9.3.2 Humidity Sensor Data Collection

Select the AN_3 and set other features of ADC as temperature sensor; after completion of conversion copy the result in variable.

9.3.3 CO and CO_2 Sensor Data Collection

Data collection from the CO sensor needs following actions to be carried out (a) Selecting the analog channel AN_1, sampling frequency, and alignment of bits for ADRESH and ADRESL, (b) Vref and power on the ADC module by setting $ADCON_0$, $ADCON_1$ and $ADCON_2$ registers, (c) starting analog to digital conversion by setting ADGO bit high (wait till ADIF flag will not indicate the completion of conversion), and (d) copy of results from ADRESH and ADRESL to variables.

Now repeat the same process to collect the CO_2 data on the channel number AN_2.

10 Results and Discussion

Sensors module, EEPROM, RTC, and LCD have been successfully interfaced to the microcontroller. EEPROM is successfully storing the logged data with time and date tag. The sensors data is being displayed on LCD module. A simple GUI has been designed to store a logged data to a text file, so that it can be analyzed further.

11 Conclusions

We have developed a low cost, 12 bit resolution data logger and successfully measured temperature, humidity, and concentration of CO and CO_2 gases. The GUI designed gives a lucratively look to the functioning of data logger. Initial results of the data logger are encouraging and we are working on to improve the GUI model as well as the accuracy of data logger.

References

1. G.L. Tang, *Lecture notes on Health and the built environment: Indoor air quality* (University of Calgary, Alberta, Canada)
2. J.D. Richard, G.S. Brager, Thermal comfort in naturally ventilated buildings: revisions to ASHRAE standard 55. Energy Buildings **34**, 549–561 (2002)
3. Microdaq (March 3, 2009); http://www.microdaq.com/data-logger/
4. D.D. Lee, D.S. Lee, Environment gas sensors. IEEE Sensors J. **1**(3), 214–215 (2001)
5. N. Kularatna, B.H. Sudantha, An environmental air pollution monitoring system based on the IEEE 1451 standard for low cost requirements. IEEE Sensors J. **8**(4), 415–422 (2008)
6. Sensor industry development and trends, Sensors express (Nov 2002)
7. RS India (April 4, 2009); http://www.rsonlineindia.com
8. R. Luharuka, R.X. Gao, A microcontroller-based data logger for physiological sensing. IEEE Proc. on Instrument and Measurement Technology Conference, Anchorage, AK, USA, 21–23 May, 175–180 (2002)
9. Data acquisition logging circuits (20 Mar 2009); http://www.hobbyprojects.com/A/acquistions_data_circuits.html
10. Data sheet of real time clock DS 1307 (8 May 2008). http://www.maxim-ic.com/products/rtc/real-time-clocks.cfm
11. Microchip (10 Jan 2009). http://www.microchip.com
12. Introduction to data acquisition (10 May 2008). http://zone.ni.com/devzone/concepted.nsf/webmain/
13. G. Mason, A handheld data acquisition system for use in an undergraduate data acquisition course. Data Acquisit. **45**, 338–393 (2002)
14. D. Malone, Build a versatile data logger. Popular Electronics, 35–42, July 1994