LabVIEW Based Low-cost Digital Temperature Measurement & Control System

Ravinder Kumar Banyal* and B. Ravindra
Indian Institute of Astrophysics, Bangalore INDIA

Shibu K. Mathew
Udaipur Solar Observatory, Badi Road, Udaipur INDIA

(Dated: June 20, 2012)

This paper describes the design and construction of an automated, low-cost temperature measurement and control system. Temperature sensing is done with a semiconductor based digital thermometer and thermostat DS1621 from *Maxim*. Digital input/output pins of the PC printer port are utilized to implement a LabVIEW controlled 2-wire serial interface for the sensor. Maximum resolution of the sensor is 0.125°C and the temperature range is -55°C to +125°C. The interface circuit was built to regulate the temperature of an oven enclosing optical filters meant for solar studies. An easy to implement, inexpensive and open source approach outlined in this work is useful to develop customized temperature control modules to meet specific experimental needs in science and engineering labs.

I. INTRODUCTION

The teaching and research laboratories are equipped with a host of computer aided test and measurement instruments. Based on the type of interface, these instruments can be classified into several categories. Most notable among these are serial (RS-232), GPIB (IEEE-488), USB, ethernet, firewire, plug-in boards and modular instruments. Besides these examples, there are application oriented, on-chip sensors that are beginning to find niche in vast areas of sensing applications in physics, engineering and medical fields[1–3]. The significant advantage of integrated circuit (IC) sensors over the conventional sensors is their miniaturized size, reduced cost, multi-functionality and simple digital interface for device control and data transfer.

In this paper, our focus will be on the IC temperature sensors, the usage of which is limited to $-55°$C to $+150°$C temperature range. These IC sensors are largely used in many industrial and engineering applications, embedded systems and consumer electronics but remains relatively less utilized in teaching and research laboratories despite their unique advantages in thermometry. They can represent temperature in digital format easily read by microprocessor or other control hardware. Further, the on-chip functionality built to perform logical operations, analog-to-digital conversion, temperature comparison etc are extremely useful features for building low-cost, fast deployable, calibration free, dynamic thermal control systems.

This demonstration note has two-fold objective. Our first goal is to introduce a simple, low-cost, open source interface circuit to build a digital temperature measurement and control (TMC) system. The I/O pins of PC's printer port are utilized to create a 2-wire serial interface to communicate with the temperature sensor. The second objective is to demonstrate the usage of National Instruments' LabVIEW software to control the digital I/O lines of the IC sensor via printer port for performing necessary tasks related to temperature measurements, data collection and thermostat operation. A software approach in this case can effectively compensate the need for a microcontroller hardware[4]. Therefore, students and researchers conversant with LabVIEW can easily exploit the versatility of IC sensors in diverse range of experiments in science and engineering applications. In the current work the usage of IC thermal sensors is demonstrated in a practical application that requires temperature stabilized optical filters for narrow band imaging of the Sun's atmosphere. We have designed and built a temperature controlled oven using a digital thermometer ICs (Model: DS1621) supplied by *Maxim* (formally *Dallas Semiconductor*). The approach outlined in this paper is also amenable to similar sensors from other manufacturers.

The material presented is organized into VI Sections. The basis for semiconductor thermometry and the functional details of the DS1621 temperature sensor IC are discussed in Section II. A typical electronic hardware circuit that provides easy communication interface between 2-wire device and computer's parallel port is presented in the Section III. The LabVIEW based software approach to control DS1621 sensor is explained in Section IV. The design of temperature stabilized oven to house the interference filters is described in Section V. Finally, conclusions are presented in Section VI.

II. TEMPERATURE SENSORS

The common temperature measurements are usually carried out with thermocouples, resistance temperature detector (RTD), thermistors and semiconductors based sensors[5]. The choice of a particular sensor depends on the temperature range, linearity, precision and desired accuracy level in the specific experiment[5, 6]. For in-

* banyal@iiap.res.in

stance, thermocouples are capable of making measurements over a wide temperature range $-260°\mathrm{C}$ to $+1600°\mathrm{C}$. The RTDs have low drift errors and excellent repeatability over $-200°\mathrm{C}$ to$+850°\mathrm{C}$. Thermistors are preferred for their high sensitivity to detect minute changes in temperature within $-100°\mathrm{C}$ to$+150°\mathrm{C}$ range. Except Si-semiconductors, the output signal from other sensors is mostly nonlinear and require lookup table to obtain correct temperature readout. Additionally, the analog temperature sensors also need extra signal conditioning, calibration, A/D conversion, logic comparator and a voltage or a current amplifier to make their output useful in practical applications. The discussion in this paper is confined to IC-based digital sensors for simple laboratory requirements where temperature resolution is ordinarily limited to about $0.5 - 1°\mathrm{C}$ and a large number of experiments can be performed within their measurement range.

Silicon is the most commonly used semiconductor material in thermometry. It operates on the principle of temperature dependent behavior of voltage-current relationship at semiconductor junctions forming diode or a transistor[7–9]. The bandgap circuit produces a voltage that varies linearly with temperature. By integrating on-chip electronic sub-systems, the device functionality of an IC based temperature sensors have been further enhanced to suit the specific requirements in TMC applications. Temperature ICs are now commercially available from several reputed companies such as NXP Semiconductors, Maxim, National Semiconductors, Microchip and Analog Devices. As a readily available test sample, we chose DS1621 IC from *Maxim* for our application.

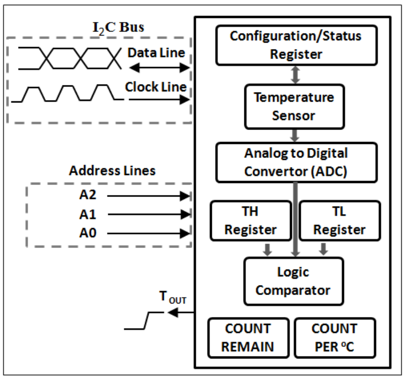

FIG. 1. Functional diagram of DS1621 temperature sensor.

A. DS1621 Temperature Sensor

The DS1621 is an 8-pin programmable IC from *Maxim* with onboard thermal-diode that is capable of measuring ambient temperature from -55°C to +125°C. The functional components of the DS1621 hardware are outlined in the block diagram shown in Fig. 1. The DS1621 also has a pre-defined set of hexadecimal commands to perform various tasks. A detailed description of DS1621 features and its command structure is provided in the product data sheet[10]. The data and command transfer between DS1621 and computer take place serially over an I_2C-bus comprising two lines, namely, a serial data line (SDL) and a clock line (SCL). The address pins A0, A1 and A2 uniquely identify the device on I^2C bus. The bit-0 and bit-1 of the configuration/status register determine the mode of operation and polarity of the thermostat T_{out} pin, respectively. An in-built analog-to-digital convertor (ADC) is used for converting the sensor output to binary format. Temperature can be read either as a single-byte (1°C resolution) or a two-byte (0.5°C resolution) data. The low and high temperature limits for thermostat operation are stored in T_H and T_L registers. The in-built comparator monitors the deviations of T_H and T_L away from the measured temperature. The T_{out} pin becomes active and turns the heater ON when the temperature equals or drops below T_L. It remains active until the temperature exceeds T_H. Same interrupt signal from T_{out} can be used to issue an alarm or shutdown an equipment to prevent a catastrophic failure. A readout from *counter* and *slope* registers is used to obtain higher temperature resolution (0.125°C)[11].

III. INTERFACE HARDWARE

Besides microcontroller, virtually any computer port (e.g. serial, parallel or USB) with digital I/O lines can be utilized to control an I^2C device. Some electronics hardware is, however, necessary to convert the port specific voltage levels to those used for data communication over the 2-wire bus. One of the simplest ways to control an I^2C device is to use PC's parallel ports which is also known as LPT or printer port. Fig. 2(a) shows one such interface circuit built using off-the-shelf electronic components[4, 12]. It uses a standard 25-pins printer port which is associated with three 8-bit registers, namely, Data Register, Control register and Status register[13]. Each of these register port is uniquely mapped to I/O address space of the PC. The Data register is usually mapped to the Base address of the port, e.g., 0x378 (or it could be 0x278 or 0xBD00). The Status register and Control register would then be at Base+1 and Base+2, respectively. The DB-25 pin assignment to the corresponding registers is shown in Fig. 2(b). In next section, we will demonstrate the usage of *Inport* and *Outport* VI functions available in base package of LabVIEW to access LPT port directly and see how to alter the ON/OFF condition of individual pins to control the DS1621 sensor.

The interface circuit, shown in Fig. 2(a), uses a 74HC05 hex-inverter to converts printer-port (TTL) signals to I^2C standard. The open drain outputs of inverters are pulled high using external pull-up resistors (10 kΩ each) connected to the +5 VDC power supply. A summary of the interface circuit is given in the following:

- The serial data (SDA) from computer to DS1621 is transmitted from Pin-2 which is D0 bit of the

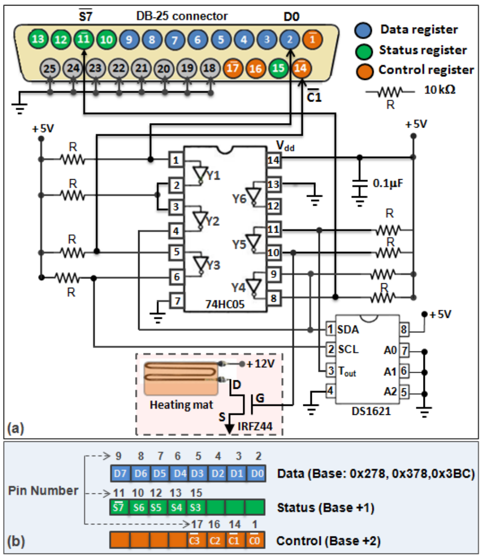

FIG. 2. A computerized TMC system with DS1621 IC. (a) Hardware circuit to convert printer port voltage signals to I_2C standard and (b) physical mapping of the printer port pins to the register ports.

Data register. For the correct logic, the Pin-2 signal goes through two inverting stages (Y1 & Y2) before feeding the SDA line of the DS1621.

- Pin-11 which is $\overline{S7}$ bit (note the inverted logic) of the Status register is configured to read the serial data (SDA) from DS1621 via Y4 inverter. Before reading data from DS1621, the software should release the SDA line by forcing the pin-2 (D0-bit) to high.
- The clock (SCL) signal is generated at pin-14 which is $\overline{C1}$ bit (note the inverted logic again) of the Control register. It is routed through Y3 inverter to generate the clock (SCL) signal for DS1621.
- Inputs of unused inverter (Y6) is grounded to prevent its output from floating.
- The configured device address, e.g., 000 is obtained by connecting the address select pins A2, A1 and A0 to the ground. For configuring multiple devices, pins A2, A1 and A0 should be accordingly connected to ground and +5VDC power supply.
- T_{out} pin of DS1621 (routed through Y4) drives the gate terminal of the power MOSFET to switch the current ON/OFF through heating element.

IV. SOFTWARE APPROACH

A. I^2C Bus Protocols

The I^2C protocols, developed by Philips Semiconductor in early 1980s, specify a common standard for signal timings, bit format, addressing scheme, synchronization and hand-shaking mechanism to foster a reliable data exchange. For a detailed overview and understanding, readers should consult I^2C bus specification document[14]. Here, we provide some salient features and general considerations that must be borne in mind while writing the LabVIEW program to control an I_2C device.

- The master (in this case LabVIEW program) initiates a transaction on the I^2C bus by issuing a START condition. The slave (DS1621) is responsible for monitoring the bus and identify itself if the address matches.
- Both master and slave can transmit (Tx) and receive (Rx) data.
- Each byte placed on the SDA line must be 8-bit long. A byte transfer has to be followed by an Acknowledge bit from the receiver. Data is transferred with the Most Significant Bit (MSB) first.
- Data is placed on SDA lines when the clock is low.
- The data on SDA line is valid for READ/WRITE only during the high period of the clock pulse.
- The master terminates the communication with slave by sending a STOP condition.

B. LabVIEW Program

Among a variety of general purpose programming platforms, National Instruments LabVIEW has gained a wider acceptance across a spectrum of users from industry and academia[15, 16]. In addition to its intuitive and easy to learn graphical programming environment, LabVIEW is supported by a powerful and rich collection of pre-written library functions and programming tools meant to accomplish a variety of tasks related to user controlled applications for equipment interface, laboratory measurements, data visualization and analysis.

The control software for DS1621 was written in LabVIEW Version 8.2 running on Windows XP platform. The LabVIEW does not have high level functionality to communicate with I^2C devices. However, its *In Port* and *Out Port* functions can directly read or write to three registers associated with the printer port. Both these VIs are located in *Functions > Advanced > Port I/O* palette. The task specific hexa-decimal commands for the DS1621 sensor can be written to it using *Out Port*. The internal logic circuitry of the I^2C device is responsible for interpreting the commands.

A collection of I^2C LabVIEW routines, shown in Fig. 3, were written to create a quick and easy graphical interface for DS1621. These Sub-VIs are used in our *Main.vi* program that carries out temperature measurements, thermostat and data logging operations. Temperature data from two DS1621 sensors can be displayed in

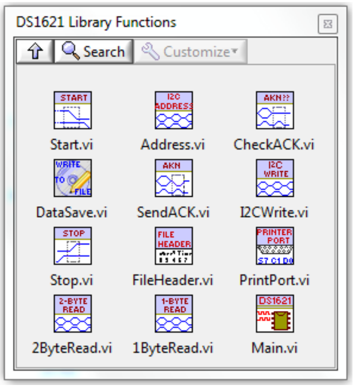

FIG. 3. A set of LabVIEW VIs written to control DS1621.

real-time on the LabVIEW front panel. The maximum update rate is about 1Hz as each temperature conversion inside DS1621 takes about one second to complete. A time stamped temperature data collected from both channels is stored in a text file using *DataSave.vi*. Without going into details of each sub-VI, we only outline a general approach to elucidate the basic concepts[1].

Prior to any communication with a slave device, the master (the LabVIEW) has to take control of the bus by initiating a START condition. The timing diagram in Fig. 4(a) indicates the occurrence of START condition forced by master pulling the SDA line low while the SCL was high. Following the START condition, the master sends an unique address byte comprising 7-bit I^{2}C device address and a $R/\overline{W}$-bit. For DS1621, the address has a fixed and a programmable part. The fixed part consists of 4-bit control code (1001) and the programmable part has three device select bits A2, A1, A0, allowing up to 8 similar sensors to share the I^{2}C bus. The LSB ($R/\overline{W}$)-bit indicates the direction of data flow, with low (0) implying WRITE to slave and high (1) indicating READ from slave. After reading one-byte of data, the receiver (could be master or slave) sends an acknowledgment (AKN) signal to transmitter by pulling SDA line low during the additional clock period, indicating a successful receipt of the data. After this master can generate a STOP condition (by pulling the SDA high while SCL is high) or issue a repeated START to initiate new transfer.

A generic LabVIEW code outlined in Fig. 4(b) uses a *For* loop to write 1-byte hex-command to DS1621. The hex-code is first converted into 8-bit array using LabVIEW's 'Number To Boolean Array' VI and reversed to transmit the bits in correct order, i.e., the MSB first. A 3-frame sequence structure built inside the *For* loop writes individual bits, one at a time, to D0 pin. In the 1st frame, SCL line which is driven by $\overline{C1}$-pin of the Control register (Base+2) is first pulled low by writing 0 to the input terminal of the *Out port.vi*. This is to ensure that clock is held low before data bit is placed on SDA line. In the 2nd frame, the first bit (MSB) of the binary code is written to D0-pin of the Data register. Note that port address (e.g., 0x378) here is wired to 'address' terminal and the binary digit is wired to 'write value' terminal of the *Out port.vi*. The data bit placed on SDA line is read by I^{2}C device when SCL is driven to high state in 3rd frame. The high SCL implies valid data condition during which the DS1621 reads the SDA line. This completes the transfer of first (MSB) bit to the DS1621. The *For* loop then increments to write the remaining bits in the same manner. Since software is used to controls the high and low states of the port pins , there is no real-time constraints on system clock. The minimum hold-time requirement for the high/low period of SCL and SDA lines are easily met by introducing a finite but small delay ($\approx$ 1ms) between the frames. The function *In port.vi*, likewise, can be programmed to read the content of the 8-bit Status register port as implemented in *1ByteRead.vi* and *2ByteRead.vi*. The $\overline{S7}$-bit of the Status register is extracted using LabVIEW's 'Index Array' function from each data byte read by iterating *For* loop. An 8-bit data array collected outside the *For* loop is then inverted using 'Reverse 1D Array' function to obtain the correct value. A typical usages of Sub-VI routines to read 2-byte temperature data from DS1621 is illustrated in Fig. 4(c).

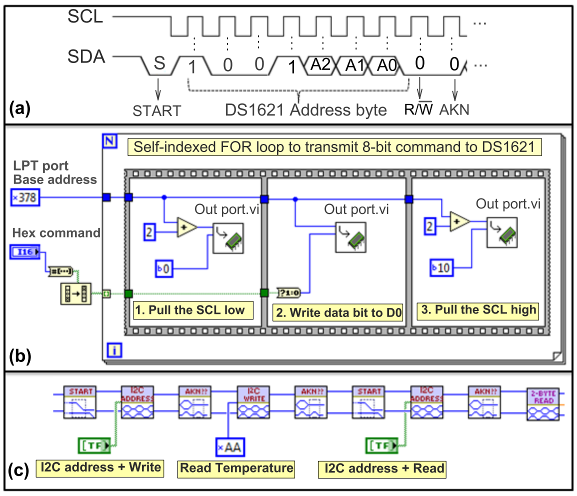

FIG. 4. (a) Timing diagram for the address-byte transfer to DS1621. (b) A LabVIEW code to write 1-byte command to DS1621 via printer port. (c) A programming sequence used in *Main.vi* to read 2-byte temperature data from DS1621.

V. TEMPERATURE CONTROL EXAMPLE

In our astronomy application, we required a temperature controlled oven to house narrow-band optical filters meant to take images of the sun during the daytime. These filters operates on the principle of thin film interference caused by multi-layer coating. The coating thickness and hence the pass-band wavelength of the filter is

[1] The LabVIEW routines are available from authors upon request.

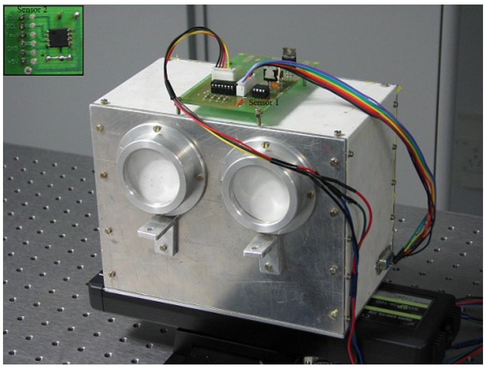

FIG. 5. Laboratory picture of the dual-port filter box with PCB temperature controller. The inset image is of the sensor 2 which is placed inside the filter box.

highly sensitive to ambient temperature changes. For a stable response, the filter has to be mounted inside a thermally insulated box whose temperature is then maintained constant at a preset level which is about $3-5°$C above the ambient. An oven was designed to hold two circular filters of 5cm diameter each. A two-channel digital TMC system was built following the electronic circuit given in Fig. 2. A laboratory picture of the filter-box oven and fabricated printed circuit board (PCB) is shown in Fig. 5. One of the DS1621 ICs (sensor 1) was mounted on the PCB board to measure the ambient temperature while the other (sensor 2) was placed inside the filter-box assembly. Resistive heating was used to elevate the oven temperature to the desired set-point. Two self-adhesive heating mats (RS components) of 10W each were connected in parallel and glued to opposite walls of the filter-box. The sensor electronics (PCB) was powered by a +5 VDC source and the heater was driven by a +24 VDC power supply. A simple ON/OFF control of the heater was found sufficed to achieve the required level of temperature stability ($\approx 0.5°$C) inside the oven box. The current switching circuit was built with IRFZ44 power MOSFET[17]. The $\mathrm{T_{OUT}}$ pin of sensor 2 was used to drive the gate terminal of the MOSFET to switch the current ON and OFF through the heater. All electronics components used in the circuit were purchased from local market costing around INR 2000.

Device operation and temperature stability of the oven were tested in the lab. A standard Pt100 thermistor, placed alongside with sensor 2 inside the oven, was used to test and verify the device performance independently. Fig. 6 shows the record of temperature data acquired for 70 minutes. The lower red-curve shows the ambient temperature inside the lab while the upper black and blue-curves indicate temperature measurements by Pt100 and DS1621 inside the filter assembly. The wide fluctuations in the lab temperature were purposely introduced by intermittent usage of air conditioning system during the experiment. The temperature inside the oven has evolved in two distinct phases. During the initial phase the temperature rise is evidently linear until the set temperature limit is crossed. The temperature readings between Pt100 and DS1621 during this phase could differ as much as 1°C. This difference is due to a relatively slower response time of the onboard thermal diode of DS1621 which is embedded inside the IC casing. The internal processing of DS1621, in addition, takes about 1 second to complete the digitization of the temperature data. Any temperature variations within that time will be averaged out. On the contrary, Pt100 is faster in responding to temperature changes as clearly seen in the rising temperature curves. After about 10 minutes, the oven temperature gradually stabilizes and nearly reaches the constant temperature phase. Temperature measurements by Pt100 and DS1621 during this phase are almost identical. The constant temperature phase is modulated by slow oscillations ($\approx \pm 0.06°$C) about the temperature set-point. These oscillation results from the thermal inertia of the system caused by a) the finite separation between the heat source and the detector and b) the poor thermal conductivity of the air mass inside the oven. It also implies the existence of a small but non-zero temperature gradients across the oven volume. In the present case, this error is not be significant to affect the the overall performance of the filters.

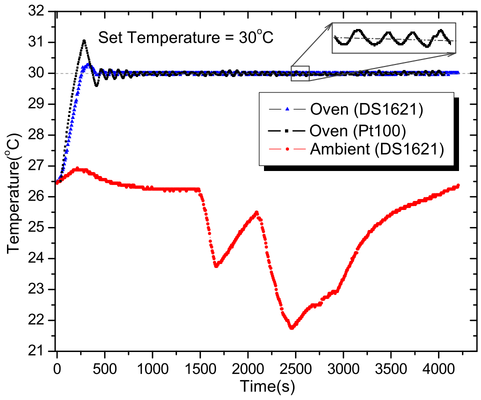

FIG. 6. Measurements of laboratory temperature (red-curve) and regulated temperature inside the filter-box oven. The inset image shows ($\pm 0.06°$C) oscillations about the temperature set-point.

One obvious drawback of simple ON/OFF controller is that it injects constant power into the system during the ON state and zero power during the OFF state. Such approach along with thermal inertia of the system prevents the ON/OFF controller from operating exactly at the temperature set point as indicated by the inset image in the Fig. 6. For a better temperature stability, the heater power should be proportional to the temperature difference between the set point and the ambient. With little alteration, the current design can be modified to build a more sophisticated PID controller[18].

VI. CONCLUSIONS

Teaching instrument interfacing methods to design and control experiments is now a standard laboratory practice in many universities and colleges. Students ability to quickly interface standard off-the-shelf instruments while also keeping pace with emerging technologies is important to meet the challenging needs of industry and the academic research. An automated temperature measurement and control system is a vital component of many industrial and research requirements. While conventional methods are still useful and relevant, the IC-based temperature sensing is relatively a new and robust approach. In this paper, we have discussed IC-based thermal sensors, suitable to meet modest needs of a many laboratory level requirements. A programmable sensor IC DS1621 obtained from *Maxim* was chosen as a test sample. An open source hardware circuit was built to implement 2-wire data communication using computer's printer port. A LabVIEW software emulating the behavior of a microprocessor was written to support I^{2}C digital read/write operation necessary to facilitate temperature measurements with DS1621. A simple but instructive ON/OFF temperature controlled oven was built to enclose a narrow-band optical filters meant for solar studies. The real-time performance and temperature stability of the device was examined in laboratory experiments. A comparison with Pt100 thermistor was performed to establish the efficacy of DS1621 in TMC applications.

[1] Freed, M., Krüger, M., Spanos, C. J., and Poolla, K., 2001, "Autonomous On-Wafer Sensors for Process Modeling, Diagnosis, and Control," IEEE Trans. Semicond. Manuf.,**14**, pp. 255–264.

[2] Yonzon, C. R., Stuart, D. A., Zhang, X., McFarland, A. D., Haynes C. L., and Van Duyne, R. P., 2005, "Towards advanced chemical and biological nanosensorsAn overview," Talanta, **67**, pp. 438-448.

[3] Tsai, N., and Sue, C. Y., 2007, "Review of MEMS-based drug delivery and dosing systems," Sensors and Actuators A: Physical, **134**, pp. 555-564.

[4] Lunca, E., Salceanu A., and Cretu, M., "Implementing the I2C Communication Protocol in LabVIEW," accessed online from www.imeko.org/publications/tc4-2007/IMEKO-TC4-2007-112.pdf

[5] Childs, P. R. N., Greenwood, J. R., and Long, C. A., 2000, "Review of Temperature Measurements," Rev. Sci. Instrum., **71**, pp. 2959-2978.

[6] Moore, J. H., Davis, C. C., Coplan, M. A., and Greer, S., 2009, *Building Scientific Appratus*, Cambridge University Press, 4th Edition, Chap. 8.

[7] Widlar, R. J., 1971, "New Developments in IC Voltage Regulators," IEEE J. Solid-St. Circ. **6**, pp. 2-7.

[8] Brokaw, P., 1974, "A simple three-terminal IC bandgap reference," IEEE J. Solid-St. Circ. **9**, pp. 388-393.

[9] Precker, J. W., and de Silva, M. A. , 2002, "Experimental estimation of the band gap in silicon and germanium from the temperature voltage curve of diode thermometers," Am. J. Phys. **70**, pp. 1150–1153.

[10] Available online from http://datasheets.maxim-ic.com/en/ds/DS1621.pdf

[11] *High Resolution Temperature Measurement With Dallas Direct-to-Digital Temperature Sensors*, Application Note AN105, accessed online from http://glotov.pp.ru/filebase/cpu/APP105.PDF.

[12] *How to Use a PCs Parallel Port to Communicate with 2-Wire Devices*, May 2004, Application Note 3230, Available at http://www.maxim-ic.com/app-notes/index.mvp/id/3230

[13] Axelson, J., 1997, *Parallel Port Complete: Programming, Interfacing, & Using the PC's Parallel Printer Port*, Lakeview Research.

[14] Available online from http://www.nxp.com/documents/user_man

[15] Moriarty, P. J., Gallagher, B. L., Mellor, C. J., and Baines, R. R., 2003, "Graphical computing in the undergraduate laboratory: Teaching and interfacing with LabVIEW", Am. J. Phys., **71**, pp. 1062–1074.

[16] Elliott, C., Vijayakumar, V., Zink, W. and Hansen, R., 2007, "National Instruments LabVIEW: A Programming Environment for Laboratory Automation and Measurement", Journal of Laboratory Automation,**12**, pp. 17–24.

[17] Available online from http://www.vishay.com/docs/91291/91291.

[18] Ibrahim, D., 2003, "Teaching digital control using a low-cost microcontroller-based temperature control kit," Int. J. Elect. Eng. Educ. **40**, pp. 175–187.