

Low-cost Thermoluminescence Measurement Using Photodiode Sensing

Author: M. Mbongo

Supervisor: R.O. Ocaya

A dissertation submitted to Faculty of Natural and Agricultural sciences,
Department of Physics, in fulfillment for the Degree of Magister Scientiae.

University of the Free State, May 2015

DECLARATION

I hereby declare that the work contained in this dissertation is entirely my own and where necessary, credit is given to materials and sources that have been referred to.

- *in memory of my mother* -

Acknowledgements

The author wish to thank the following:

- The National Research Foundation of South Africa DST/NRF Innovation for the Masters study scholarship,
- Dr. R.O.Ocaya for his supervision and guidance through the research,
- The Department of Physics, University of the Free State (QwaQwa campus) staff and students for their support.

Abstract

Many branches of scientific and industrial research require precise instrument(s) for control and measurement. Such instruments tend to be prohibitively expensive. In the current global economic climate the funding to procure research equipment is fast dwindling. One current interest that our institution has is the synthesis and thermoluminescence (TL) characterization of phosphors, polymers and nano-materials. TL measurement requires precise control and measurement of sample temperature as a function of output intensity.

In the present research, we describe the design and construction of a low-cost TL instrument that allows automatic control of various steps of the experiment while logging instantaneous intensity output. This work started with two fundamental considerations. Firstly, whether a low-cost thermoluminescence equipment is feasible. Secondly and more importantly, whether a photodiode can form the intensity sensing apparatus. We answer these questions affirmatively by first putting together a course of research and assimilating the necessary tools needed. Using the the resulting demonstrable TL instrument, we demonstrate the versatility for temperature sequencing, range and heating control of the sample over the temperature range of 23 to 600 ± 0.5 °C. A comparable instrument in the institution operates at a maximum ceiling of 300 °C. Additional refinements to the prototype instrument enable the sample temperature to be held constant at any temperature within this range with the aid of software tuned Proportional-Integral-Derivative (PID) control. The intensity measurements are made using a temperature-compensated, large area photo-diode operated in photovoltaic mode and covering a wavelength range 400 to 1100 nm. The interfaces of the instrument that made the instrument easy to use were developed con-currently. For instance the Universal Serial Bus (USB) handler, the Visual BASIC.NET control program that also logs the temperature and intensity data, and the PIC18F2520 micro-controller firmware code that was written in the C-language. Several other tools, listed in the body of the dissertation were also used. Finally, we present various results of temperature control and measurement and a demonstration measurement on a ceramic sample.

Table of Contents

Declaration	ii
Acknowledgements	iv
Abstract	v
Table of Contents	vi
List of Figures	ix
List of Tables	xi
1 An overview of thermoluminescence measurement	1
1.1 Introduction	1
1.2 Identification of the research problem	3
1.3 Methodology	4
1.4 Structure of the report	4
1.5 Intended applications of the instrument	5
2 The theory of thermoluminescence	6
2.1 Background	6
2.2 The origin of TL in materials	7
2.3 Mathematical descriptions of TL	9
2.4 Kinetic equations	10
2.4.1 General-order kinetics	10
2.4.2 First-order kinetics (slow retrapping)	12
2.4.3 Second-order kinetics (fast retrapping)	13
2.5 Analysis of glow peaks	15
2.5.1 The initial rise method	15
2.5.2 The total glow peak method	16
2.5.3 Peak shape method	17
2.5.4 Heating rate method	19
2.5.5 The isothermal decay method	20

3	System design for photodiode sensing	22
3.1	Block diagram and system description	22
3.2	The sample holder and heating element	23
3.3	Design of the heater power supply	24
3.4	The temperature sensor circuit	26
3.4.1	Development of the sample temperature sensor	26
3.4.2	A model of the heating arrangement	27
3.4.3	Simulations and actual heating test	30
3.5	The P-I-N photodiode photometer	33
3.5.1	Development of the photometer circuit	35
3.5.2	Performance of the photometer	37
3.6	Powering the TL system using USB	38
3.6.1	Power converters	38
3.6.2	Development of the LVPS	44
3.6.3	Simulation and performance of the LVPS	45
4	High-resolution PID temperature controller	48
4.1	Introduction	48
4.2	System identification	49
4.2.1	The heater and feedback arrangement	49
4.2.2	The software PID temperature controller	50
4.2.3	Realization of a discrete PID controller	52
5	Results of sample measurements on the TL system	56
5.1	Introduction	56
5.2	Photograph of the prototype TL instrument	56
5.2.1	Unit step responses of the TL instrument	59
5.2.2	Closed-loop unit step response	59
5.2.3	Demonstration of linear heating	60
5.2.4	Demonstration of intensity measurement	61
6	Conclusions	65
	References	68
A	Sample thermometer development	73
A.1	Schematic diagram of first prototype	73
A.2	Sample temperature sensor evaluation PCB	74
A.3	Final TL system schematics	75
A.3.1	Connectors	75
A.3.2	Microcontroller and USB interface board	76
A.3.3	Photometer and temperature sensors	77
A.3.4	LVPS power inverter and $\pm 5V$ regulators	78

B	The PIC firmware	79
C	User interface program	83
C.1	Visual BASIC.NET control program	84

List of Figures

2.1	Energy transitions in a TL process.	8
2.2	Comparison of first ($b = 1$), second ($b = 2$) and intermediate order ($b = 1.3$ and $b = 1.6$).	14
2.3	Initial rise part of the glow curve, $I_C = 15\% I_M$	15
2.4	Parameters that characterize a single peak of thermoluminescence. 18	
3.1	Block diagram of sample holder/heating element.	22
3.2	Block diagram of sample holder/heating element.	24
3.3	Block diagram of temperature conditioning sensor.	29
3.4	Schematic simulation model.	30
3.5	Simulation model results.	31
3.6	The temperature measurement circuit based on a K-type thermocouple. 32	
3.7	Results of temperature conditioning sensor circuit(early results) 33	
3.8	Results of temperature conditioning sensor circuit(latest results) 34	
3.9	Photodetector block diagram	35
3.10	PV mode operation of the photodiode for ultra-low light level sensing.	36
3.11	Photodiode spectral response	39
3.12	Simplified circuit diagram of a boost converter.	40
3.13	Phase one, storing energy in the transformer.	42
3.14	Phase two, dumping the energy from the transformer into the buffer capacitor.	43
3.15	Phase three, energy dump completed discharge of drain-source capacitor.	44
3.16	Schematic diagram of USB to rail power inverter.	46
3.17	Predicted USB to rail power inverter results.	47
4.1	Diagram of the PWM heater driver.	51
4.2	Parameters of interest in Ziegler-Nichols open loop tuning.	52
4.3	Block diagram of the implemented temperature controller. The closed-loop function can be written as $H(s)=C(s)G(s)$	53

4.4	MATLAB/Simulink model of the closed-loop system that gives good transient performance. The input shown is a linear ramp with input duty cycle starting from 0 to 100%.	53
4.5	Parallel implementation of the PID TL-system heater controller.	54
5.1	Labeled photograph of the TL instrument prototype.	58
5.2	Experimental and simulated open-loop unit step response based on Eq. 4.2.3.	59
5.3	Heating behavior under PID control.	60
5.4	Baseline measurement (a) with ceramic sample (b) showing fitting function.	63
5.5	Plot of the baseline corrected measurements on the ceramic sample.	64

List of Tables

2.1	Appropriate values of c_α and b_α for first order	19
2.2	Appropriate values of c_α and b_α for second order	20
3.1	Determination of number of turns for transformer	25

Chapter 1

An overview of thermoluminescence measurement

1.1 Introduction

Thermoluminescence (TL) is the emission of light from some minerals and certain other crystalline materials. The light energy is derived from electron displacements within the crystal lattice of such substance caused by previous exposure to high-energy radiation. Electrons in some solids can exist in two energy states, a lower energy state called the valence band and a higher energy state called the conduction band. The difference in energy between the two bands is called the band gap. Electrons in the conduction band or in the band gap have more energy than the valence band electrons. Normally in a solid, no electrons exist in energy states contained in the band gap, this is a “forbidden region”. In some materials, defects in the material exist or impurities are added that can trap electrons in the band gap and hold them there. These trapped electrons represent stored energy for the time that the electrons are held. This energy is given up, usually emitted as light photons when the material is heated up, as the electron returns to the valence band. Thermoluminescence measurement requires three things: a heater, which raises the temperature of the material, a meter – to record the

temperature of the material as it is heated and photodetector - to measure light output from the material. Most TL systems use a photomultiplier tube (PMT) as the light detector. For the heater different heating methods are used which fall under two categories:- contact and non-contact heating systems. In systems that use heating by contact the TL sample holder is heated by passing an electrical current or by a hot finger moved by some lift mechanism. A hot finger heating system was used in versions of the TL instrument [22, 40]. In some existing non-contact heating systems, the samples are heated by hot nitrogen gas, laser beam, or a light pulse from halogen lamp. Laser heating methods, for instance were used by Braunlich [7]. Contemporary researchers have developed different TL systems. For instance P. Neelamegam et al.[41] developed a system that permits the recording of glow curve data (intensity vs temperature) and enables computer processing of such data to evaluate glow peak temperature and activation energy using the 6502 micro-controller. P. Molina et al[40] developed a fully digital system that allows arbitrary heating profiles, including a logarithmic heating scheme. R. Bhatnagar et al[43] developed a system that has automatic control of temperature sequence, the range and the rate of cooling and heating of the sample with additional light-emitting diodes (LEDs) for excitation of sample with a minimum of operator interference. Lyamayev [25] developed a low-cost system that has a wide range of linear heating and cooling rates, precise temperature regulation, simplicity of construction and low cost. Neelamegam and Rajendran[36] developed a system that controls linear heating using PIC16F877 micro-controller [33, 36]. The system of J.W. Quilty et al[22] is interesting because it uses PT100 resistors as heating and sensing elements, with hardware PID temperature control. In the present research we report on the construction of a low-cost TL system based on the PIC18F2520 for both solid and powdered samples. The work relies on an original, low-noise and high sensitivity photodiode sensor and conditioning circuitry that, to the best of our current knowledge, has not been reported before. Also, rather than rely heavily on hardware

temperature control, the designed system implements a firmware (PIC software) based PID control. The sample temperature is set using a resistive heating element that is directly driven by a MOSFET with a 10 kHz pulse width modulated (PWM) drive issued from the control firmware as well. The temperature level is sensed using conventionally, K-type thermocouple and used, through the analog-to-digital port of the processor to provide error feedback to the PID controller. A few unique refinements extend the range of the temperature control and measurement resolution stated above. For example, a rewind step-down toroidal mains transformer outputting +180 V/0.6 A drives the heater, enabling a wider temperature range for 0 to 100% duty cycle range. The TL instrument is interfaced as a full-speed device to a Windows.net computer on the Universal Serial Bus 2 (USB 2.0) port from which it also derives the operating power for the low voltage interfaces and operational amplifiers. The user commands are initiated on a program written in Microsoft.NET frameworks 1.0 - 4.0 [12]. Response data is sent to the personal computer for further processing through the same port and stored in a file format that enables direct import into analysis environments like Microcal Origin 6 and MS Excel.

1.2 Identification of the research problem

The goal of creating a working model of a TL instrument with its various control systems forms the main aim of this research. The primary consideration is a final product that is low-cost, has relative ease of development and prototyping using the available resources within the faculty, ease of deployment into the target environment, reliable and easy to use for a non-technical researcher. In particular, the contribution to the body of research knowledge is the investigation of the use of a low-cost, large detecting area photodiode as the light sensing element. This is the first time such a study has been undertaken. The initial

results are encouraging, but at the same time there are clear indications arising from the study of considerable scope for future work. Inherent in achieving these objectives is the presentation of the current state of the knowledge in the instrumentation of thermoluminescence.

1.3 Methodology

To achieve the foregoing goals the approach will be mainly as follows. Firstly, an extensive literature survey will be done to highlight the state of TL instruments under the various competing requirements. Secondly, the tools required for the successful conduction of the research will be assimilated. These include mostly the software simulation packages such as MATLAB, LTSpice and Diptrace prototyping program, MPLAB and mikroC PIC development. Additional tools are hardware-based such as a low-cost PIC programmer capable of writing development firmware code to the PIC processor. These tools will then be used to create a model of the intended TL instrument. Thirdly, the outputs of these tools will then be used to create a prototype that will generate results using real samples. Finally, the results will be obtained and the performance merits of the designed system will be evaluated and commented upon.

1.4 Structure of the report

The report is organized for a connectedness and a flow that speeds up the delivery of the essential concepts and ideas of the research. Chapter one will define the concept of thermoluminescence, from historical perspectives to early forms of TL measurement, to recent developments and trends in TL instrumentation. Chapter two will present the equations for the energy dynamics or kinetics of TL processes. It will show how the all important trap

and other parameters can be deduced from empirical data, such as a good TL instrument will automatically record and store. Ordinarily the sensed data is stored on a convenient medium on the controlling computer. Chapter three presents the design of the various components of the TL system for the research and outlines their integration into a demonstrable instrument. Chapter four reports on the various specific aspects of the design that require particular elucidation, such as the firmware-based PID controller, the temperature compensated photo-detector and the USB computer interfaces. Chapter five presents the results of typical runs of the completed instrument as well as the discussions of the results. Chapter six presents the concluding remarks, comments and suggestions for future directions of the instrument, or similar low-cost instruments. An appendix section is included with useful supplementary material.

1.5 Intended applications of the instrument

TL measurement is one of the most important characterization operations in many areas of material science research. For instance, in the Department of Physics at QwaQwa campus of the University of the Free State where the research was done, there is much interest in TL measurements on phosphors that are doped with rare earth metals. The demand on the existing instrument indicates this importance and led to the question of alternative instrumentation. TL measurements are also expected in the life sciences, where plant and animal based materials and products may need to have their TL emissions characterized. Other areas that are expected to find the designed system useful include, but are not limited to geology, forensic sciences, and even undergraduate laboratories to demonstrate TL measurement concepts in general.

Chapter 2

The theory of thermoluminescence

2.1 Background

The phenomenon of thermoluminescence (TL) of minerals was practically observed and scientifically recorded in 1663 by Sir Robert Boyle[32], upon warming a diamond in contact with his body in the dark it shone. Elsholtz [32] observed a similar effect in 1776 from the mineral fluorspar. In 1883 Henri Becquerel monitored a similar effect while measuring the infrared spectra of uranium. It was concluded that not only diamonds but a large number of minerals emit light energy upon warming. These observations led to the definition of TL. From 1895 the studies of thermally stimulated emission of light were carried out and contributed to discovery of ionization radiation. Examples of minerals that have thermoluminescent properties are quartz, feldspar, calcite, clays, limestone and flint. These phenomena are today used in solid-state research, nuclear safety, medical dosimetry, geologic age determination and for archaeological dating. In the dating method, mechanisms which are responsible for resetting the dating clock thereby linking the intensity of the emitted light energy to a time scale are established [32]. The First law of TL was established [26]. This law states that the TL of minerals is roughly

proportional to the irradiation dose to which they had been exposed [26]. Another important observation for making TL a useful tool for dating was that, if the same mineral is re-heated, no more light is emitted, and it is only after the application of a new radiation dose that light may again be emitted [52]. Thermoluminescence is today very important, with applications in many fields of study such as material sciences, archaeological sciences, health physics, space sciences, geology aiding physics, earth sciences, biology and biochemistry and quality control in industry, to name just some of the mainstream areas of study [32].

2.2 The origin of TL in materials

TL originates from the temperature induced release of energy, stored in the lattice structure of the crystal following long-term internal and external exposure to nuclear radiation (high energy particles). This emission of light is from an insulator or a semiconductor material following the previous absorption of energy from ionizing radiation which occurs at low temperature as it gets heated. It must not be confused with incandescence, the ordinary light you get when you heat something up white/red hot. The temperatures involved are far lower, the maximum temperatures involved being around 400 °C, while most of emitted light is obtained at much lower temperatures. From the description above it is apparent that there are three ingredients that are necessary for the production of TL. Firstly, the material must be an insulator or a semiconductor, i.e. there must be a band gap. Metals therefore do not exhibit luminescent properties. Secondly, the material must have at some time absorbed energy during exposure to ionizing radiation and has to

happen at lower temperatures. The radiation sources can be alpha (α), beta (β) or gamma (γ) particles, χ -rays, electron beams and neutrons particles or ultraviolet (UV) light. Exposure of materials to any of these creates electron-hole pairs which eventually get trapped in localized trap-states existing inside the band gap of the material, and most of the charge carriers remain trapped afterwards. Thirdly, light emission is triggered by heating of the material [32, 44]. Heating the material gives trapped electrons/holes enough thermal energy to escape from the traps to the conduction band (or the valence band). From here they may get re-trapped again or may recombine with trapped holes/electrons. The site where recombination takes place is called recombination centre. If this site is radiative, then the centre is called a luminescence centre.

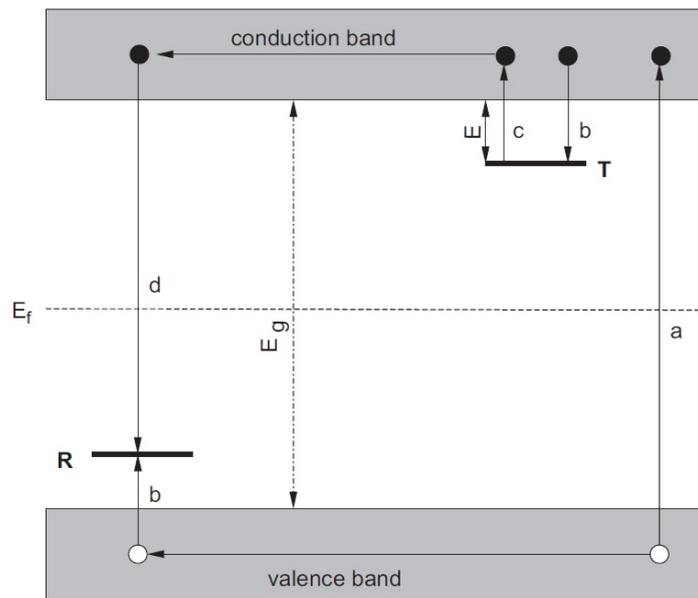


Figure 2.1: Energy transitions in a TL process.

Figure 2.1 shows the energy band model showing the electronic transitions in a TL material according to a simple two-level model:

- a. Generation of electrons and holes,

- b. Electron and hole trapping,
- c. Electron release due to thermal stimulation, and
- d. Recombination.

Solid circles are electrons, open circles are holes. Level **T** is a electron trap, level **R** is a recombination centre, E_f is Fermi level, E_g is the energy band gap. To heat up the material one can use any of these heating systems, for instance: planchet heating, hot anvil heating, hot gas heating, RF (radio frequency) heating, IR heating or laser heating [21].

2.3 Mathematical descriptions of TL

There are several forms of the equations that describe the thermoluminescence process and the shape of the resultant glow-peak. These correspond to different sets of underlying assumptions. In order to quantify the response of a given material to TL, it is instructive to establish the kinetics of the material. Most TL materials obey one of two kinetics: first-order or second-order kinetics. General-order kinetics have been developed that reduce to either first-order or second-order under specific assumptions. The theory of first order kinetics was first described by Randall and Wilkins [45]. It is based on the assumption that there is no re-trapping of electrons after they are released from traps, i.e. if an electron is liberated from a trap it always goes straight to a luminescence centre. Three additional simplifying assumptions are also made:

- a. Only the trapping and release of electrons is considered. The treatment of holes would be exactly similar.
- b. only one kind of trap and one kind of recombination centre are involved.

c. The temperature increases at a constant rate during readout.

Since the electrons in the traps have a Maxwellian distribution of thermal energies, the probability per unit time of an electron escaping from a trap of depth E below the conduction band is given by

$$p = s \exp \left\{ -\frac{E}{kT} \right\}, \quad (2.3.1)$$

where k is Boltzmann's constant, T is the absolute temperature in Kelvin, and s is a constant, although it may vary slowly with temperature, according to Randall and Wilkins [45]. Following Mott and Gurney [45], s is interpreted by regarding the trap as a potential box; then s is the product of the frequency with which the electron strikes the sides of the box, and the reflection coefficient. s has units Hz and is called the frequency factor. Its value is expected to be somewhat less than the vibrational frequency of the crystal, typically $10^{-12} Hz$.

2.4 Kinetic equations

As mentioned in the preceding section, TL materials conform to either first-order or second-order kinetics. General-order kinetics have been developed that reduce to either first-order or second-order under specific assumptions.

2.4.1 General-order kinetics

According to May and Partridge [29], the thermoluminescence intensity I_{TL} for the general order kinetics as

$$I_{TL} = -\frac{dn}{dt} = s'n^b \exp \left(-\frac{E}{kT} \right) \quad (2.4.1)$$

where k is Boltzmann constant ($eV.K^{-1}$), T is temperature of the sample (K), E is activation energy or the trap depth (eV), s' is a constant quantity called the pre-exponential factor, b is the order of kinetics, which may have any value from 1 to about 2 but can exceed this range and n is concentration of trapped carriers (m^{-3}). Hence the rate per second of electrons recombining with holes is dn/dt . Equation(2.4.1) is not related to any particular model described by energy level scheme. For example, when two electrons are contained in a single trap $b = 1.5$. By integrating Equation(2.4.1) using constant heating rate $\beta = dT/dt$ leads to the equation describing the thermoluminescence for general-order kinetics

$$I(T) = s'' n_0 \left(-\frac{E}{kT} \right) \left[1 + \frac{s(b-1)}{\beta} \int_{T_0}^T \exp \left(-\frac{E}{kT'} \right) dT' \right]^{-\frac{b}{b-1}} \quad (2.4.2)$$

where $s'' = s'n_0^{b-1}$ expressed in Hz and n_0 is the initial concentration of trap carriers. TL peaks generated by Equation(2.4.2) for $b = 1.3$ and $b = 1.6$ are compared with first-order and second-order TL peaks in Figure 2.2.

By considering the logarithm derivative of Equation(2.4.2)

$$d(\ln I)/dT = (1/I)(dI/dT). \quad (2.4.3)$$

Under the condition $dI/dT = 0$ where $T = T_M$, this gives general-order kinetics equation:

$$\frac{2kT_M^2 bs}{\beta E} \exp \left(-\frac{E}{kT_M} \right) = 1 + \frac{s(b-1)}{\beta} \int_{T_0}^{T_M} \exp \left(-\frac{E}{kT'} \right) dT'. \quad (2.4.4)$$

In Equation(2.4.4) the temperature of the peak maximum T_M depends on the initial concentration of charge carriers n_0 since $s = s''n_0^{b-1}$.

2.4.2 First-order kinetics (slow retrapping)

Substituting $b = 1$ and $s' = s$ where s is called frequency factor (s^{-1}), Equation(2.4.1) transforms to the equation describing first-order kinetics. It relies on the energy level model of Randall and Wilkins (RW)[45, 9]

$$I = -\frac{dn}{dt} = -p n = n s \exp\left(\frac{-E}{kT}\right) \quad (2.4.5)$$

The RW model is based on the assumption that there is a strong tendency for recombination and that electrons that are released thermally from the traps and excited into the conduction band recombine quickly with trapped holes i.e. no re-trapping of electrons occurs after they are released from traps. By integrating Equation(2.4.5) with respect to T gives the equation describing thermoluminescence with first-order kinetics

$$I(T) = n_0 s \exp\left(\frac{-E}{kT}\right) \exp\left[\int \left(\frac{s}{\beta}\right) \exp(-E/kT) dT\right] \quad (2.4.6)$$

It can be found from Equation(2.4.6) that the peak height varies with n_0 but peak position stays fixed. The peak is asymmetric, wider at lower temperature side than on the high temperature side. As activation energy increases the peak shifts to high temperatures along with decrease in height and increase in width. The peak shifts to high temperatures and size of the peak increases as heating rate increases, and are said to be the properties of first-order kinetics equation.

To find the condition at maximum temperature, the same method as in general-order kinetics was used leading to

$$\frac{\beta E}{kT_M^2} = 2s \exp(-E/kT_M) \quad (2.4.7)$$

In Equation(2.4.7) n_0 does not appear which shows that peak does not depend on n_0 for first-order order kinetics but on the heating rate. It can be shown

that as β increases, peak also increases.

2.4.3 Second-order kinetics (fast retrapping)

Second-order kinetics are obtained by substituting $b = 2$ and $s' = s/N$ under the assumptions made by Garlick and Gibson (GG) model [17]

$$I = -\frac{dn}{dt} = \left(\frac{n^2}{N}\right) s \exp\left(\frac{-E}{kT}\right). \quad (2.4.8)$$

The GG model assumes that re-trapping dominates. This means that electrons that are released thermally from the traps and excited into the conduction band have a higher probability to be re-trapped by an electron trap than recombination with a hole at a luminescence centre. By integrating Equation(2.4.8) with respect to T leads to gives the equation describing thermoluminescence with second-order kinetics:

$$I(T) = \frac{n_0^2 s \exp\left(\frac{-E}{kT}\right)}{\left[1 + \left(n_0 \frac{s'}{\beta}\right) \int \exp\left(\frac{-E}{kT}\right) dT^{-2}\right]} \quad (2.4.9)$$

It can be found from Equation(2.4.9) that the peak grows nearly proportional to n_0 and shifts to lower temperatures as n_0 increases. The peak height decreases and shifts to high temperatures as activation energy E increases. Peak height increases and its position shifts to high temperatures as heating rate increases, and this are said to be the properties of second-order kinetics equation.

Also the second-order kinetics for maximum temperature can be found in a manner similar to first order kinetics

$$\left(\frac{n_0 s'}{\beta}\right) \int \exp\left(\frac{-E}{kT}\right) dT + 1 = (2kT_M^2 n_0 s' \beta E) \exp\left(\frac{-E}{kT_M}\right) \quad (2.4.10)$$

Form Equation(2.4.10) it can be seen that peak depends on n_0 , as n_0 increases, so too does s'' which in turn causes the TL peak to shift to lower temperatures.

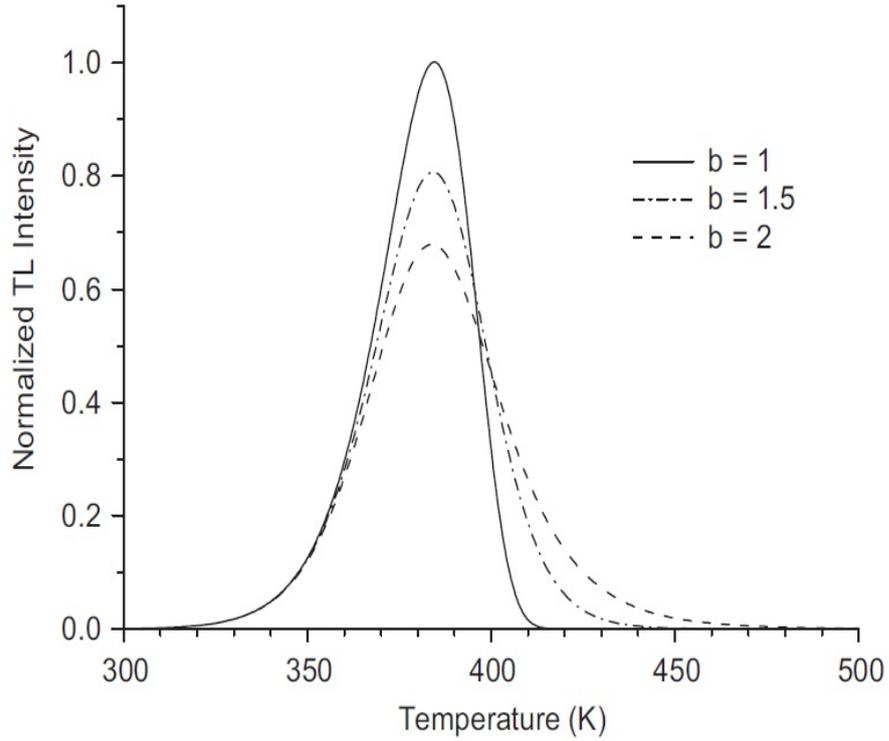


Figure 2.2: Comparison of first ($b = 1$), second ($b = 2$) and intermediate order ($b = 1.3$ and $b = 1.6$).

By looking at the equations for maximum temperature conditions for three kinetic orders, it can be concluded that for any non-first-order kinetics the temperature of the peak maximum is dependent of the initial concentration of charge carriers n_0 .

The characteristics of three kinetic equations can be drawn from Figure 2.2 which shows the comparison between three kinetic orders. It can be seen that the general-order retains some of the character of first- and second-order. It can be seen that the TL peak maximum is inversely proportional to the order of kinetics b .

2.5 Analysis of glow peaks

There are many methods available for the analysis of glow peaks to obtain values of the trapping parameters E and S , and the kinetics order for the release of electrons. These methods have been reviewed in detail by Chen and Winer [8], Nicholas and Woods [37], and Shalgaonkar and Narlikar [50]. The methods of analysis used in this study are presented briefly below.

2.5.1 The initial rise method

The initial rise method is the simplest and most generally applicable method for evaluating the activation energy E of single glow peak. Its analysis applies to the low-temperature tail of the glow peak ($T < T_C$ and $I < I_C$) as shown in Figure 2.3.

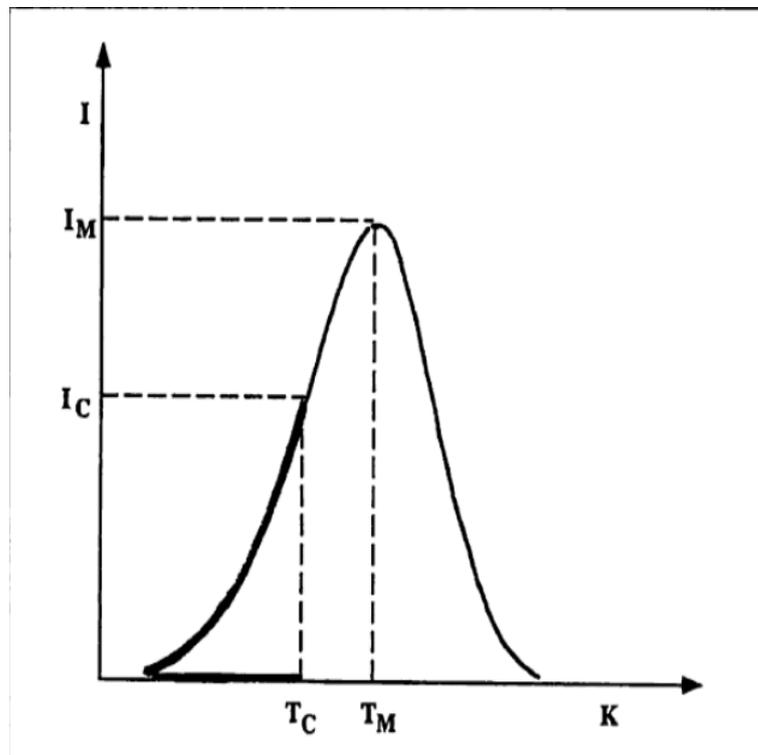


Figure 2.3: Initial rise part of the glow curve, $I_C = 15\% I_M$.

In this region the number of trapped electrons n changes with temperature only by small amount and can be regarded as constant [17, 44]. Equation (2.4.5) may then be written as

$$I = \exp\left(\frac{-E}{kT}\right) \quad (2.5.1)$$

A plot of $\ln I$ versus $1/T$ should yield a straight line of slope $(-E/K)$ from which activation energy E is readily found. This analysis is independent of the kinetics of the process but has the disadvantage that the frequency factor s cannot be determined.

2.5.2 The total glow peak method

The shape of an isolated glow peak may be analyzed to obtain the values of E , s and the kinetics order b . Equation (2.4.5) may be integrated to give

$$\int_t^{t_\infty} I dt = n = \frac{I}{s} \exp\left(\frac{-E}{kT}\right) \quad (2.5.2)$$

which may be written as

$$\frac{I}{\int_t^{t_\infty} I dt} = \frac{\beta I}{\int_T^{T_\infty} I dT} = s \exp\left(\frac{-E}{kT}\right). \quad (2.5.3)$$

Plotting $\ln(I/\int_T^{T_\infty} I dT)$ versus $1/T$ over the whole of the isolated glow peak will give a straight line of slope $-E/k$ and intercept $\ln(s/\beta)$ for a first-order kinetics peak, providing that a linear heating rate is used. If the glow peak is plotted as intensity I versus time t then the plot of $\ln(\int_t^{t_\infty} I dt)$ versus $1/T$ yields a slope of $-E/k$ and intercept $\ln s$ and is independent of heating rate. The integral $\int_T^{T_\infty} I dT$ is proportional to the number of trapped electrons n , and is measured as an area from the glow peak. This method of analysis has advantages over other methods since it uses data for the whole of the

glow peak and not just specific regions or points. The initial rise method is essentially incorporated in the low-temperature region of the total glow peak plot where the number of trapped electrons may be regarded as constant. In general order kinetics where

$$dn/dt = -n^b s' \exp\left(\frac{-E}{kT}\right), \quad (2.5.4)$$

the order of the kinetics b may be determined by plotting $\ln(I/n^b)$ versus $1/T$. A straight line will result when the correct value for the order of kinetics has been chosen [3].

2.5.3 Peak shape method

In peak shape method the kinetic parameters are evaluated using small numbers of points extracted from the glow curve. Figure 2.4 shows points which are considered for characterizing a single peak where I_M is intensity at peak maximum, $1/2 I_M$ is intensity at half peak maximum, T_M is peak temperature at maximum, T_1, T_2 temperature on either side of T_M corresponding the half-maximum intensity and derived parameters $\tau = T_M - T_1$ is the half-width at low temperature side of the peak, $\delta = T_2 - T_M$ is the half-width at high temperature side of the peak and $\omega = T_2 - T_1$ is total half-width. The very simple and although not accurate method which only depends on T_M was develop by Urbach [5] where

$$E = T_M/500. \quad (2.5.5)$$

Grosswiener [24] develop his method based on first-order TL peaks and found

$$E = 1.41kT_M T_1/\tau \quad (2.5.6)$$

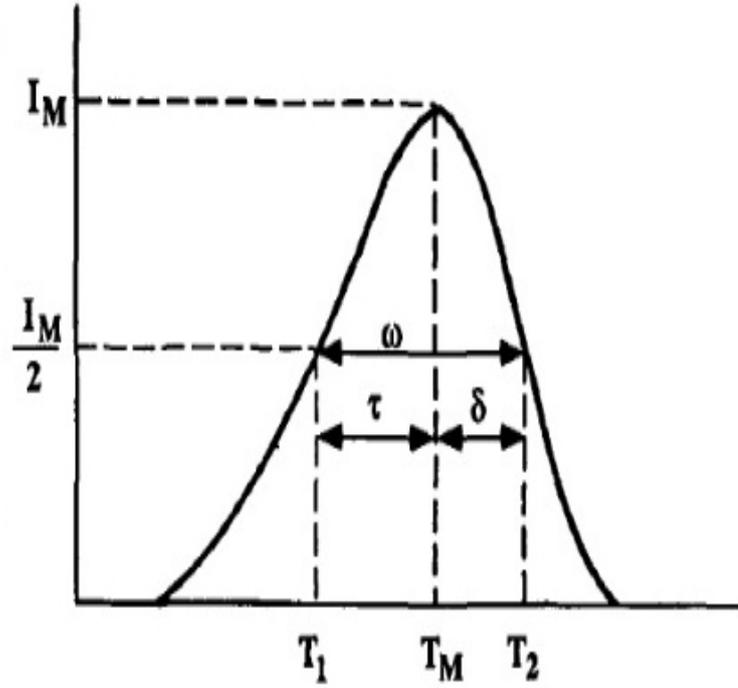


Figure 2.4: Parameters that characterize a single peak of thermoluminescence.

where τ had coefficient of 1.51 which was later changed to 1.41 by Dussel and Bube [16]. Halperin and Braner [3] developed the formula for first-order kinetics

$$E = 1.72(kT_M^2\tau)(1 - 1.58\Delta) \quad (2.5.7)$$

where $\Delta = 2kT_M/E$. It was found that s is independent of T in this case. Chen [44] gave a better version of Halperin and Braner method namely

$$E = 1.52(kT_M^2\tau - 1.58(2kT_M)). \quad (2.5.8)$$

He summarized this and similar methods by a single Equation(2.5.9) and a set of parameters to be used with t , d or w and first- or second-order kinetics.

$$E_\alpha = c_\alpha(kT_M^2/\alpha) - b_\alpha(2kT_M) \quad (2.5.9)$$

where α represents τ , δ or ω and the appropriate values of c_α and b_α are as shown in Table 2.1 and 2.2, a is given by the variation of the pre-exponential factor (s) as T^a , which has to be determined before Equation(2.5.9) can be used.

2.5.4 Heating rate method

2.5.4.1 Various heating rates

This analysis is based on the shift of the peak maximum to higher temperatures with increased heating rate, as predicted by the RW model [45]. The condition for a maximum is

$$\frac{\beta E}{kT_M^2} = 2s \exp\left(\frac{-E}{kT_M}\right) \quad (2.5.10)$$

Hoogenstraten [19] suggested measuring the peak maximum temperature T_M as a function of heating rate β . Plotting $(\ln T_M^2/\beta)$ versus $(1/T_M)$ yields a straight line of slope (E/k) and intercept $(\ln E/sk)$ for a first-order kinetics peak. Non-linear heating rates may be used in this analysis provided that the heating rate at the peak maximum is determined [39]. The method has also been extended to general-order kinetics [8].

2.5.4.2 Two different heating rates

Booth-Bohun [14, 46] developed a method using two different heating rates but it is applied to non-first-order TL peaks and it is based on the variation

Table 2.1: Appropriate values of c_α and b_α for first order

First order	τ	δ	ω
c_α	1.51	0.976	2.52
b_α	$1.58 + a/2$	$a/2$	$1 + a/2$

Table 2.2: Appropriate values of c_α and b_α for second order

Second order	τ	δ	ω
c_α	1.81	1.71	3.54
b_α	$2 + a/2$	$a/2$	$1 + a/2$

of I_M with b which is much more faster than the variation of T_M with b and it is represented by

$$\ln I_M = \frac{-E}{kT_M^2} - \ln sn_0 - \frac{b}{b-1} \ln \left[1 + (b-1) \frac{sE E_2(u_m)}{k u_m \beta} \right] \quad (2.5.11)$$

where $u_m = E/kT_M$ and $E_2(u_m)$ is the second exponential integral. The factor in square brackets is very close to unity, hence using two linear heating rates β_1 and β_2 one obtains

$$\ln I_{m1} \cong \frac{-E}{kT_M^2} - \ln sn_0 \quad (2.5.12)$$

$$\ln I_{m2} \cong \frac{-E}{kT_M^2} - \ln sn_0 \quad (2.5.13)$$

which gives

$$E = \left[\frac{kT_{m1}}{T_{m2}} T_{m1} - T_{m2} \right] \ln \frac{I_{m1}}{I_{m2}} \quad (2.5.14)$$

To finding frequency factor s the following relation is used

$$s = \frac{E\beta_1}{kT_{m1}^2} \left[\frac{\beta_1}{\beta_2} \left(\frac{T_{m2}}{T_{m1}} \right)^2 \right]^{T_{m2}/(T_{m1}-T_{m2})} \quad (2.5.15)$$

2.5.5 The isothermal decay method

The isothermal decay method of analysis uses data for the decay of luminescence when the sample is held at constant temperature. It is carried out in the temperature range where thermoluminescence is normally exhibited. The

decay in luminescence intensity is predicted by solving Equation(2.4.5) for the constant-temperature condition giving

$$I = Csn_0 \exp\left(\frac{-E}{kT}\right) \exp\left[-st \exp\left(\frac{-E}{kT}\right)\right] \quad (2.5.16)$$

Plotting $\ln I$ versus t gives a straight line of slope $m = s \exp(-E/kT)$. Performing isothermal decay measurements at different temperatures and plotting $\ln m$ versus $(1/T)$ yields a straight line of slope $(-E/k)$ and intercept $(\ln s)$ for a first-order kinetics peak. The method may be applied to overlapping peaks by holding at temperatures suitable for the decay of the lowest temperature peak. This gives fast- and slow-decaying components which may be separated graphically.

The effect of thermal quenching maybe observed while performing a series of TL measurements with different heating rates. Typically, with increasing heating rate, the maximum of a TL glow peak shifts to higher temperatures. At a higher temperature, the luminescence is quenched more intensely so that the whole area under TL peak decreases. The thermal quenching efficiency versus temperature, $\eta(T)$, is given by the following equation [42]

$$\eta(T) = \frac{1}{1 + C \cdot \exp\frac{-W}{kT}} \quad (2.5.17)$$

where C and W are quenching parameters [35].

Chapter 3

System design for photodiode sensing

3.1 Block diagram and system description

The designed TL system, Figure 3.1, consists of aluminium sample holder, a type K-thermocouple (chromel-alumel) for temperature feedback, a temperature-compensated, large area photodiode, several conditioning amplifiers, a PIC18F2520 micro-controller, the USB 2.0 interface and the multi-rail power supplies. The following sections describe the various aspects of the overall block diagram in more detail.

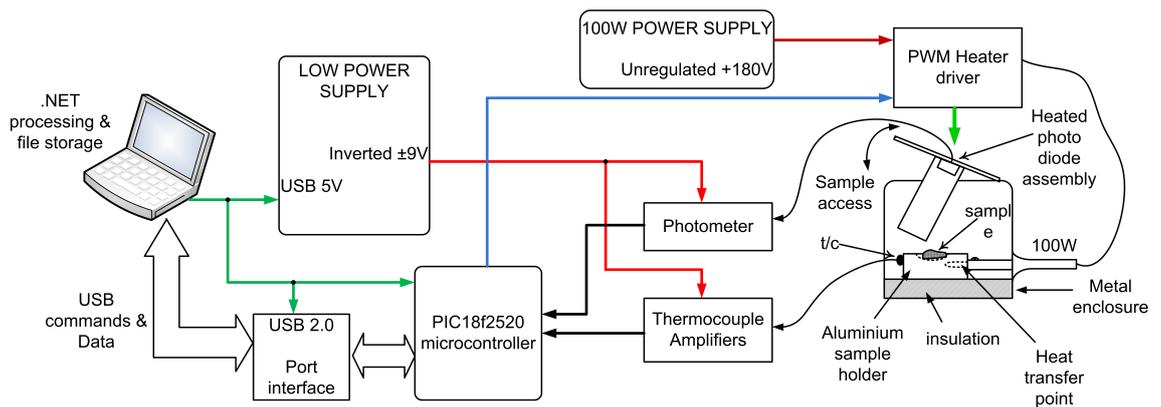


Figure 3.1: Block diagram of sample holder/heating element.

3.2 The sample holder and heating element

The most critical part of a TL system is the heater [32]. Several methods exist by which the temperature of the sample can be raised in a controlled manner and the most common of these implies passing of current through a planchet or a coil (i.e resistive heating). The use of a planchet is probably the most popular [6, 15, 40, 54]. The planchet itself is usually a thin metal strip (e.g. tantalum or nickel-chrome), with typical dimensions of about $0.025 \text{ cm} \times (1-2) \text{ cm} \times (4-5) \text{ cm}$ [32]. The advantage of the planchet system over other arrangements is the low thermal mass of the heater and fast thermal response. A disadvantage is that the planchet can often warp at high temperatures, sometimes resulting in permanent distortion. An alternative to the planchet arrangement is a heater block, usually of copper [23] which itself is heated by a resistance. The heater block design is especially useful where the block can be maintained at low temperatures. These heating methods can not achieve fast heating and cooling rates and this is due to their large thermal inertia [32]. Quilty et al [22] realized that both planchet and block heater present design challenges in obtaining uniform temperature distribution across the sample and the temperature sensor is not in contact with the sample.

Figure 3.2 a block diagram of sample holder/heating element, resistive heater and thermocouple. The sample holder is an aluminium block of $2.5 \times 1.7 \times 0.5 \text{ cm}^3$ to which the K-type thermocouple was been affixed using a pressing metal plate and screws. A convenient 100 W resistive heating source was implemented simply using a commercial pen-type soldering iron with a tip long and narrow enough to be inserted tightly into a compatible hole drilled into the sample holder.

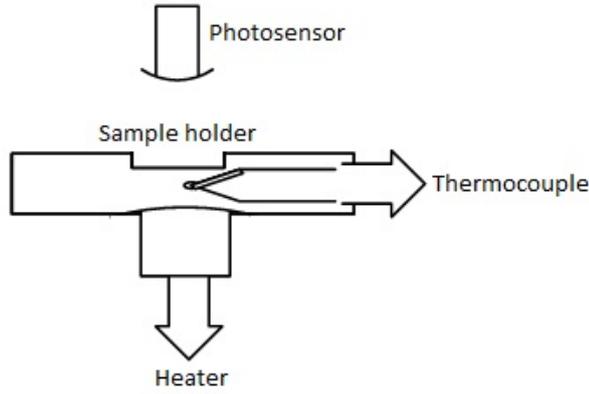


Figure 3.2: Block diagram of sample holder/heating element.

3.3 Design of the heater power supply

In order to supply the heater/heat source with required power, a homemade secondary rewind transformer was built to drive a 100 W soldering iron with a resistance of 554Ω , V_{rms} (root mean square voltage) can be calculated from peak voltage V_{pk} :

$$V_{pk} = V_{rms} \sqrt{2} \quad (3.3.1)$$

but since

$$I_L = \frac{V_{pk}}{R_L} \quad (3.3.2)$$

I_L is load current and R_L is resistance of the load. The power can be represented in terms of V_{pk} and R_L by

$$P_L = I_L V_{pk} = \frac{V_{pk}^2}{R_L} \quad (3.3.3)$$

rearranging Equation(3.3.3) we get

$$V_{pk} = \sqrt{P_L R_L} \quad (3.3.4)$$

For values of $P_L = 100W$ and $R_L = 554\Omega$, V_{pk} was found to be 235.4 V and V_{rms} was 166 V for a sine wave. In order to drive a 100 W soldering Iron, the transformer that will produce V_{rms} of 166 V is needed. To make transformer with such output, the calculations tabulated in Table 3.1 were made to help in finding the number of turns of primary coil.

Table 3.1: Determination of number of turns for transformer

V_p	V_s	n_p	N
240.3	17.51	90	1235
240.3	15.75	80	1220
240.3	9.82	49	1199

Using data in Table 3.1, the average of $N = 1220$ turns ratio was calculated.

The number of turns in the secondary winding can be calculated

$$N = \frac{n_p}{n_s} = 1220$$

In an ideal transformer, the induced voltage in the secondary winding (V_s) is in proportion to the primary voltage (V_p), and is given by the ratio of the number of turns in the secondary (n_s) to the number of turns in the primary (n_p) as follows from Faraday's law

$$\frac{V_s}{V_p} = \frac{n_s}{n_p} \quad (3.3.5)$$

rearranging Equation (3.3.5) we get

$$n_s = \frac{V_s}{V_p} n_p$$

substituting known variables we get the number of turns in secondary windings

as

$$n_s = \frac{166}{240.3} \times 1220 = 845 \text{ turns}$$

The diameter of the wire for the secondary coil of transformer was found to be 0.57 mm. The transformer should have 1220 turns in the primary core and 845 turns on the secondary core. Test result showed that it was able to producing 157 V unloaded and 146 V loaded. At 100 % duty cycle the heater temperature rose to 320 °C.

3.4 The temperature sensor circuit

Temperature measurements are essential in TL systems. The most commonly used temperature sensors are compensated integrated circuits like the LM35, thermocouples, resistance temperature detectors (RTD), thermistors and silicon based sensors. Commercial TL systems use thermocouples for temperature measurement circuit. Some systems use two thermocouples. One is used for the feedback purposes of the temperature controller and the other for the sample temperature measurements [27, 28]. Recently, Quilty et. al [22] built a system that uses platinum wire (PT100) resistors as both the heating and the sensing element in order to obtain uniform temperature distribution across both sample and temperature sensor. In this project a K-type thermocouple was used for sample temperature sensing. The room-temperature compensation sensor was based on the LM35 integrated circuit.

3.4.1 Development of the sample temperature sensor

The temperature conditioning sensor circuit consists of low-cost elements, namely a Chromel-Alumel (K-type) thermocouple, the LM35 temperature sensor and LM324N operational amplifiers. The K-type thermocouple was used for its high thermopower and good resistance to oxidation. It can operate

in the temperature range of -269 to $+1260$ °C. At room temperature its overall sensitivity (Seebeck coefficient) is $41 \mu\text{V}/^\circ\text{C}$. The output of the thermocouple is not sufficient to drive the analog to digital converter (ADC) directly and must therefore be conditioned using operational amplifier circuits prior to the ADC input. The four independent, high gain, internally frequency compensated operational amplifiers found in the LM324 were used for this purpose. However, the low-cost LM324 has an appreciable input offset voltage of about 2mV [49] and additional circuitry based on resistor networks were added to eliminate the offset. The LM35 precision temperature sensor was used to measure the reference room temperature. It has a higher output than the thermocouple and in this design was anticipated to work over the temperature range of 0 to 100 °C. At room temperature its voltage varies at $10 \text{mV}/^\circ\text{C}$, it stays accurate to within ± 0.75 °C over its temperature range [48]. The conditioning circuit of the LM35 is much simpler because of its higher output level and output impedance. A knowledge of the mathematical models of all the components of the heating arrangement was necessary to design a highly controllable and reliable sample heater. The following section describes the overall mathematical model of the sample heater arrangement.

3.4.2 A model of the heating arrangement

The subsequent equations describe modeling and design of temperature conditioning sensor. The output voltage of thermocouple is given by

$$V_{TC} = k_1(T_H - T_R) \quad (3.4.1)$$

where $k_1 = 41 \mu\text{V}/^\circ\text{C}$ is the temperature coefficient of the thermocouple, T_H is the measured temperature and T_R is the reference room temperature. If the output of the thermocouple is to be boosted to a suitable level by an

operational amplifier of gain A_1 , then the output of such amplifier can be written:

$$\begin{aligned}
 V_1 &= A_1 V_{TC} \\
 &= k_1 A_1 (T_H - T_R) \\
 &= k_1 A_1 T_H - k_1 A_1 T_R.
 \end{aligned} \tag{3.4.2}$$

Similarly, the output voltage of a buffered LM35 output T_R is given by:

$$V_2 = k_2 T_R \tag{3.4.3}$$

where $k_2 = 10 \text{ mV}/^\circ\text{C}$ is the temperature coefficient of the LM35. In order to get actual measured temperature, the reference temperature needs to be subtracted from thermocouple output i.e. Equation (3.4.2) subtract from Equation (3.4.3). This operation can be performed using an subtractor operational amplifier having a transfer function of the form:

$$\begin{aligned}
 V_3 &= A_2 (V_2 - V_1) \\
 &= A_2 [k_2 T_R - (k_1 A_1 T_H - k_1 A_1 T_R)] \\
 &= A_2 [k_2 T_R - k_1 A_1 T_H + k_1 A_1 T_R]
 \end{aligned} \tag{3.4.4}$$

where A_2 is the gain of the subtractor. To eliminate the effect of room temperature on the results during TL measurements let

$$k_2 T_R = -k_1 A_1 T_R.$$

Then, since

$$k_1, k_2 \neq 0,$$

we have

$$k_2 = -k_1 A_1.$$

Both k_1 and k_2 are both positive with $k_2 > k_1$ so that by rearranging the last result we get:

$$A_1 = -\frac{k_2}{k_1}. \quad (3.4.5)$$

For the K-type Chromel-Alumel thermocouple, A_1 was found to be -243.9. This implies the further use of an inverting amplifier with a gain of 243.9, so that $-k_1 A_1 = k_2$. The overall transfer function of the thermocouple is then

$$\begin{aligned} V_3 &= A_2(-k_1 A_1 T_H) \\ &= k_2 A_2 T_H \end{aligned} \quad (3.4.6)$$

From Equation (2.4.7) it can be concluded that change in room temperature will have no effect on the sample temperature measurement. To calculate the value of A_2 , we let for example $V_3 = 5$ V at the highest TL measurement with maximum temperature of 500 °C, then substitution in Equation (3.4.6) gave unity gain for A_2 .

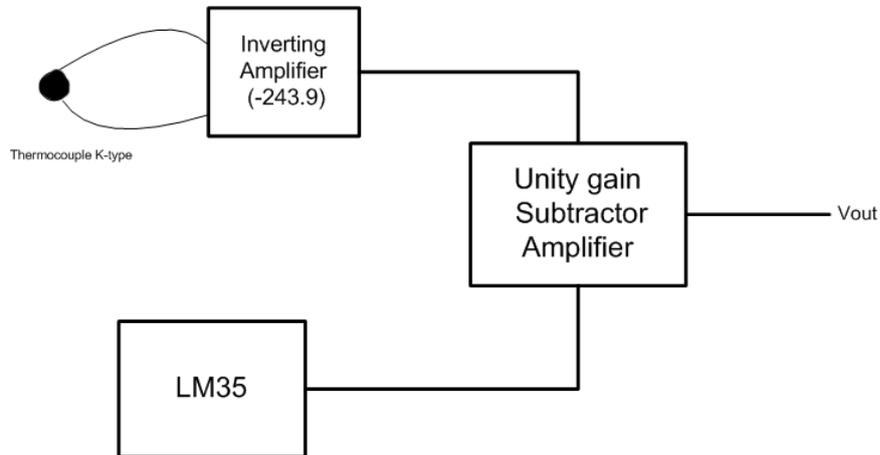


Figure 3.3: Block diagram of temperature conditioning sensor.

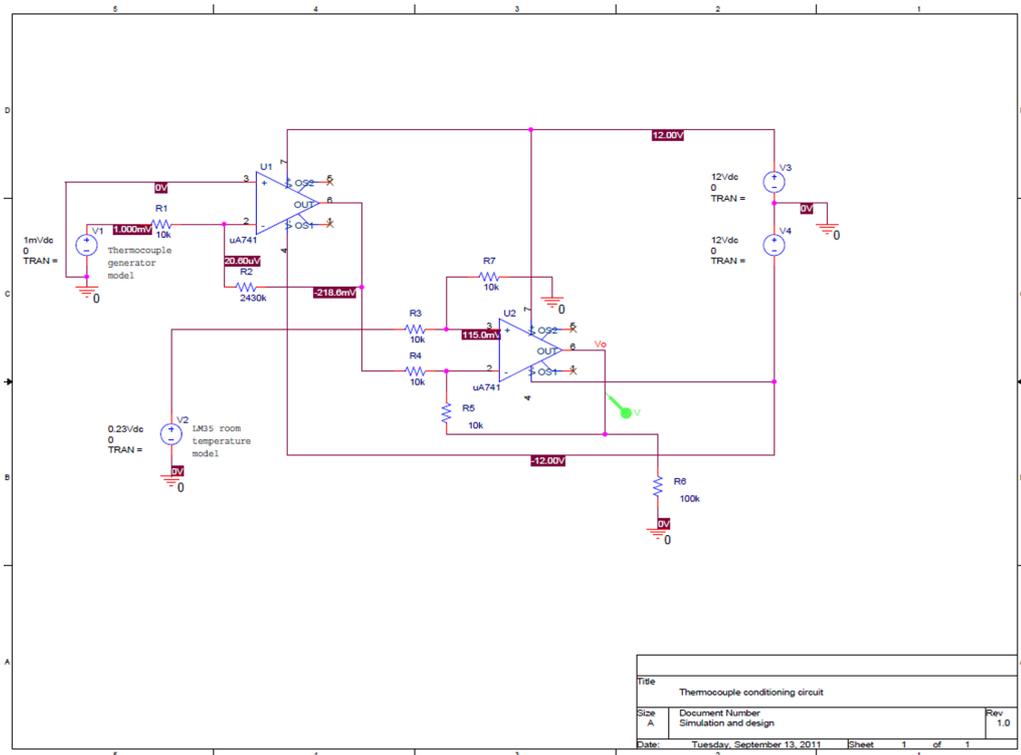


Figure 3.4: Schematic simulation model.

3.4.3 Simulations and actual heating test

3.4.3.1 Simulation

The thermocouple circuit was first simulated, built and then verified using an actual heating test. The circuit was built around the low-cost LM324N quad operational amplifier. External offset trimming and room-temperature compensation using the LM35CZ device were used to improve temperature measurement accuracy. The sample holder is an aluminium block of $2.5 \times 1.7 \times 0.5$ cm³ to which the K-type thermocouple was affixed using a pressing metal plate and screws. A 100 W resistive heating source was implemented simply using a commercial pen-type soldering iron with a tip long and narrow enough to be inserted tightly into a compatible hole drilled into the sample holder. For the simulation OrCAD/PSpice software was used. Simulations allow the evaluation of the functionality of the concept and safely gives insight

into possible future performance issues. Simulation has the advantage that parameters can be readily adjusted as much as is necessary at no cost. Figure 3.4 shows the schematic diagram as it appears in OrCAD/PSpice. In Figure 3.4, the thermocouple model is given by V_1 . Its value was swept over the voltage range of 1 to 24 mV in steps of 0.1 mV to emulate rising sample temperature. The room temperature model as measured by LM35 is V_2 . Its value was preset to fixed voltages of 0.15 V, 0.23 V and 0.30 V to simulate temperatures of 15, 23 and 30 °C respectively. The simulated results are

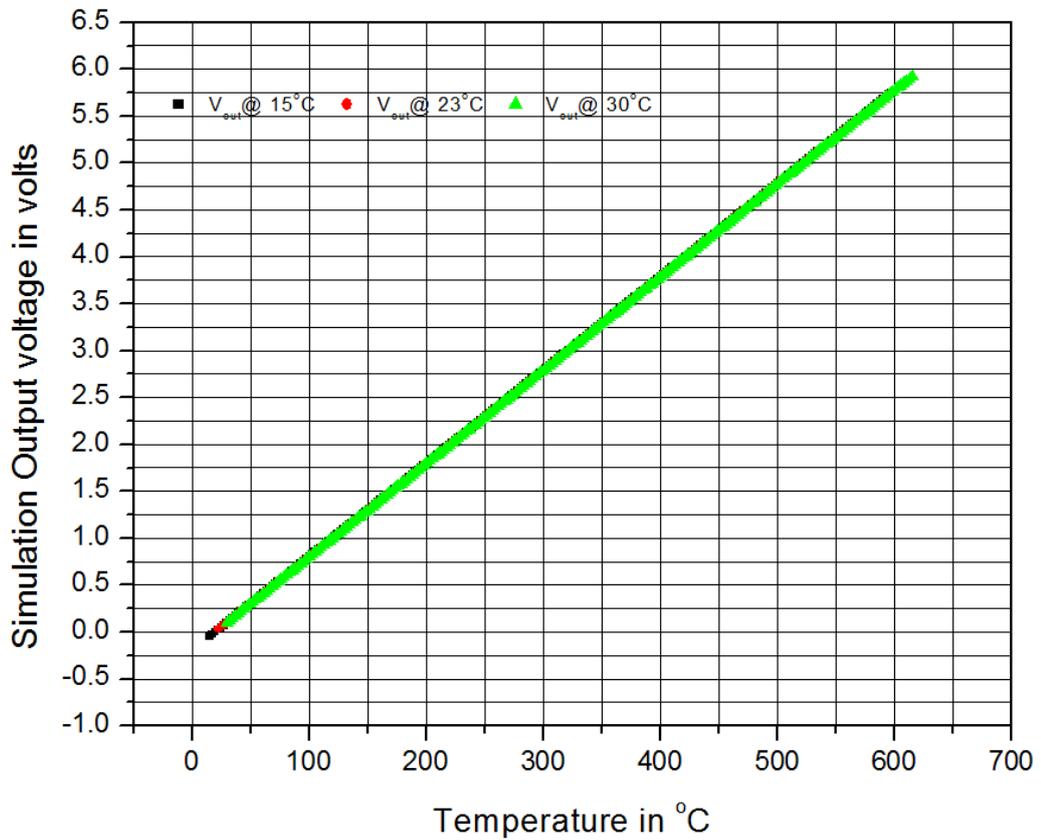


Figure 3.5: Simulation model results.

shown graphically as output voltage versus the input thermoelectric voltage. Using the Equation (3.4.1), inputs voltage were converted to temperature. Figure 3.5 shows the graph of measured temperature versus simulated output

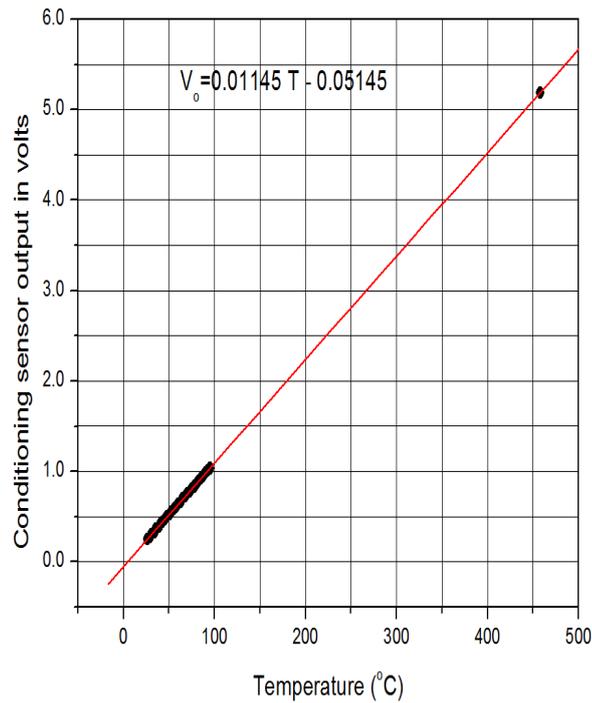


Figure 3.7: Results of temperature conditioning sensor circuit(early results)

was then connected and output of the LM35 conditioning circuit to check the room temperature reading. For one particular test it produced an output of 270 mV which corresponded to room temperature of 27 °C. The thermocouple was placed in a beaker filled with boiling water and temperature measurements were recorded, two tests were run and the results are shown in Figure 3.7 and 3.8

3.5 The P-I-N photodiode photometer

The quality of TL data depends on the instrumentation used, particularly the operating characteristics of the photodetector system. Sensitivity, efficient conversion, fast response, low noise, sufficient area, high reliability and low-cost are the basic requirements for the photodetector. In the literature cited

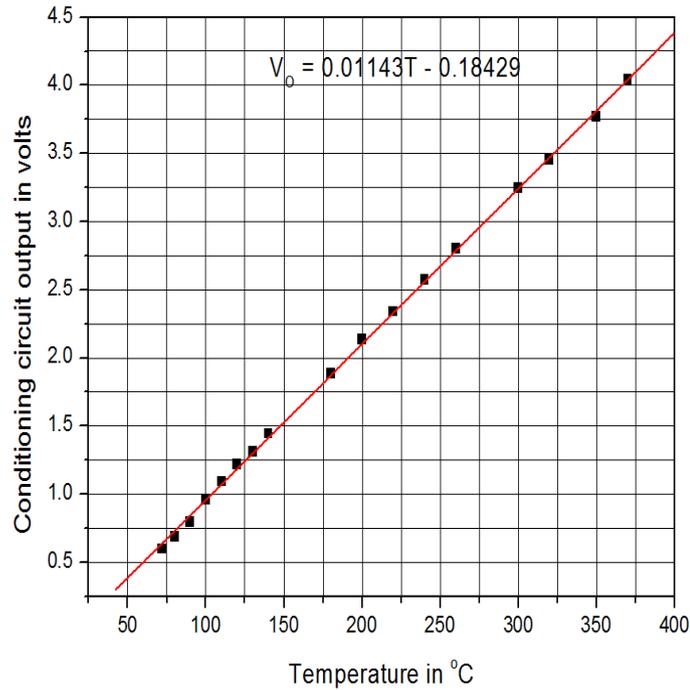


Figure 3.8: Results of temperature conditioning sensor circuit(latest results)

before the development of the reported TL systems has relied on the classical, generally hard to find thermionic emission photo-multiplier tube (PM tube). This sensor is also notoriously sensitive to temperature variations. It exhibits noise figures that can only be kept low by careful cooling to realize the stated sensitivity. The cooling required is often at cryogenic temperatures. However, they are the industry standard and although they do not meet all the requirements simultaneously, PM tubes are versatile enough to provide ultra-fast response and extremely high sensitivity. The material of photocathode in PM tubes determines its the spectral response.

The point of departure of the present research from the literature is the hypothesis that a carefully configured silicon P-I-N photodiode may be used in place of the PM tube. If this is the case then, being cheap, easy to use and condition, the P-I-N photodiode may hold considerable promise. The quantum

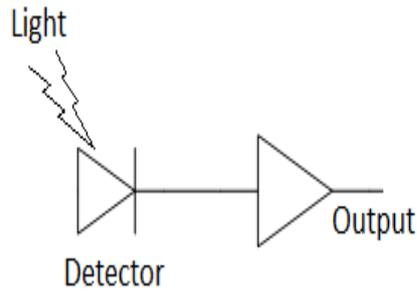


Figure 3.9: Photodetector block diagram

efficiency (QE) of P-I-N diodes ranges from 65 to 90% unlike PM tubes made of bialkali materials such as $K_2-Cs-Sb$, whose QE has a ceiling of about 27%. Photodiodes have very small volume and are insensitive to magnetic fields while most of PM tubes are affected by the presence of magnetic field deflecting electrons from their normal trajectories and causing a loss of gain. The spectral response of the P-I-N diodes available for this research range from 400 to 1100 nm, with a spectral responsivity of 0.6 A/W [1, 2, 47, 18]

3.5.1 Development of the photometer circuit

The photometer circuit, Figure 3.10, employs two OPA111BM operational amplifiers suitable where very low noise (7 nV/pHz), low input offset ($\pm 50 \mu V$), low drift ($0.5 \mu V/.C$) and low bias current ($\pm 0.5 pA$) amplifiers are necessary [10]. This is mainly because extremely high gain is required to sense very low intensity counts. The intensity sensor was proposed to be used with the BPW21R photodiode from Vishay [2] operated in the photovoltaic (PV) mode. Generally, photodiodes operated in the photovoltaic mode have much higher intensity sensitivity than in the photoconductive mode [11]. Photovoltaic

mode means no voltage across the diode and no dark current, this leads to linear output and low noise. The lack of dark current removes error, low noise makes smaller measurements possible and linear output makes calculations easier. Solid state photodiodes also have sensitivity to temperature. The

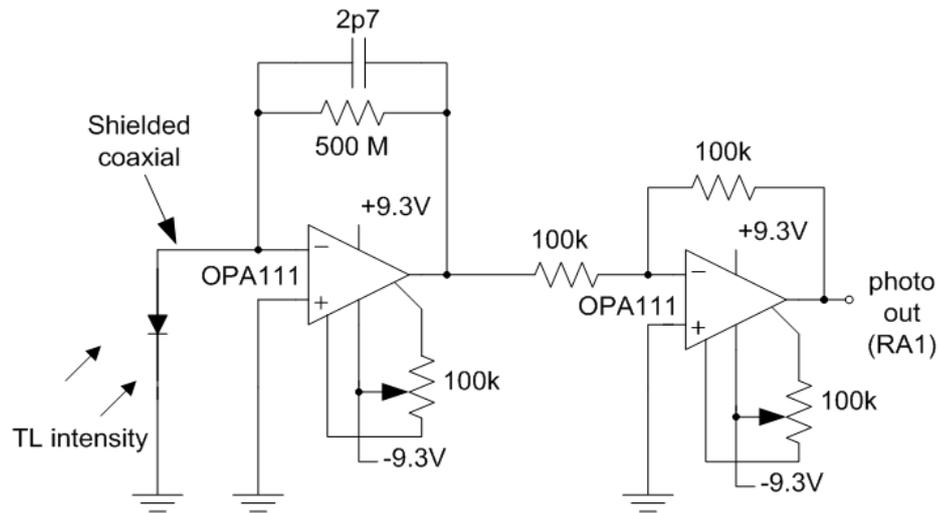


Figure 3.10: PV mode operation of the photodiode for ultra-low light level sensing.

approach used here to assure repeatability is to preheat the diode before running baseline or sample characterization. This was done using a slightly overdriven TIP41C transistor bolted onto the photodiode array to maintain the detector temperature at around 30 °C. Pre-cooling the sensor using a low-cost Peltier effect device from a portable car-cooler to about 0 °C can also be done. The operational amplifier conditioning circuits produce outputs between 0 and 5 V to exploit the maximum resolution of the 10-bit analog to digital converter (ADC) in the PIC18F2520. The photovoltaic mode of operation (unbiased) is preferred when a photodiode is used in low frequency applications (up to 350 kHz) as well as ultra low light level applications like in our case. Figure 3.10 shows the schematic diagram of PV mode operation of photodiode for ultra low light level. The 500 MΩ resistor achieves negative

feedback (i.e. R_F) and its value was chosen for the very small currents that are expected to flow near darkness. It provides the necessary high gain in the current to voltage (I-V) converter circuit of the first operational amplifier. The 2.7 pF capacitor is a feedback capacitor (C_F) that is included to stabilize the feedback loop. Without the capacitor the extremely high gain of the I-V converter would lead to spurious operation due to the increased likelihood of noise pickup and spontaneous oscillation. The second operational amplifier is an inverting amplifier that corrects the negative output of the I-V converter to positive for a 0 to 5V input of the ADC. The total voltage output of the photometer is given by:

$$V_{OUT} = I_P R_F, \quad (3.5.1)$$

where R_F is a resistance feedback resistor and I_P is photocurrent.

3.5.2 Performance of the photometer

The performance of photo-sensor can be characterized by the responsivity. Figure 3.11 shows the graph of responsivity versus wavelength. The responsivity R is related to quantum efficiency η by

$$R = I_p/P_o = \eta q/h\nu \quad (3.5.2)$$

where I_p is photo-current and P_o is optical power. Quantum efficiency is given by

$$\text{QE} = 1240R_\lambda/\lambda \quad (3.5.3)$$

where R_λ is responsivity and λ is wavelength. Manche [27] reported that light emission of thermoluminescent materials is in the visible light wavelength range 390 to 720 nm. Figure 3.11 shows the spectral responsivity of the BPW21 at room temperature. In the graph it can be seen that the photodiode

responsivity in the visible range is 0.06 to 0.48 A/W. Since the photodiode is mounted on an aluminium block that acts as an isothermal unit, changes in temperature due to the heater as well as ambient are not expected to affect the spectral response. An assumption that is made here is that the wavelengths emitted from the sample and detected by the photodiode will be within a small range in the infrared part of the spectrum. Therefore, the effect on the spectral response is assumed to be minimal, and a linearization function in the infrared part of the spectrum can be found, being generally decreasing with increasing spectral response. This assumption is reasonable since the sample and sample-holder do not glow red hot, that is, they remain mostly in a narrow infrared region. Furthermore, the linearization itself is undertaken in the control software itself based on the room temperature and the photometer output - both of which are acquired by the ADC.

3.6 Powering the TL system using USB

3.6.1 Power converters

3.6.1.1 The boost converter

The TL instrument derives its low voltage power supplies (LVPS) from the USB port. The high-voltage 180 V supply that drives the PID controlled heater is derived from a separate mains-based transformer. The latter is described in a different section. The LVPS of the TL instrument supplies power to the low-voltage sections of the TL instrument, in particular the PIC controller (+5 V), the analogue amplifiers ($\pm 12V$) which comprise the temperature sensors, the intensity signal conditioning circuit and various voltage references.

The LVPS scheme is based on a small flyback transformer that is driven

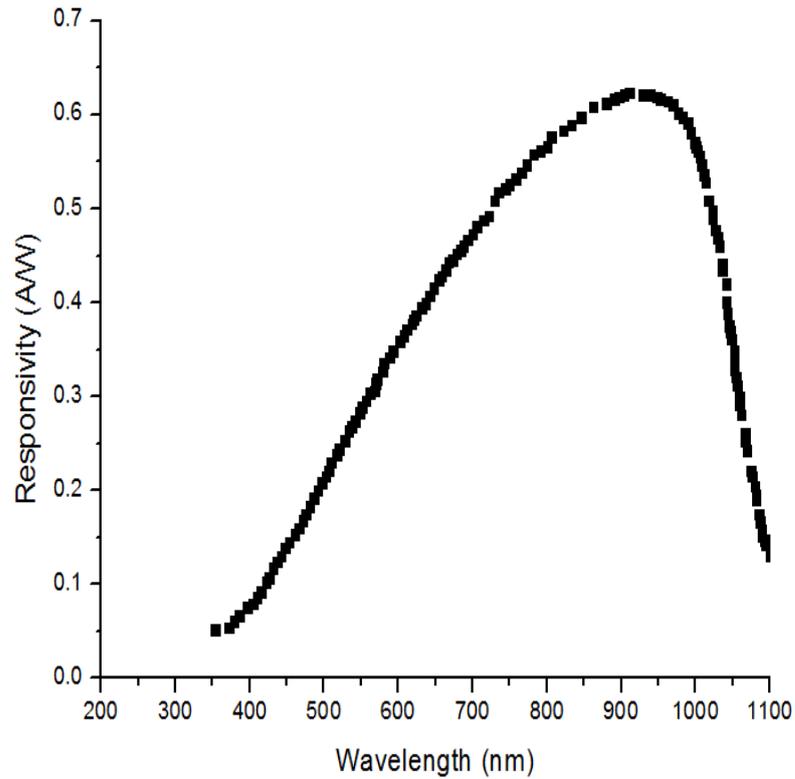


Figure 3.11: Photodiode spectral response

at 300 kHz and a common pulse-width modulated astable oscillator built around the commonly available 555 timer device. The high frequency of the astable enables the flyback transformer to be kept small while delivering relatively high power output with good regulation. Additionally, the high frequency of operation allows easier output filtering for low ripple. In order to understand the operation of the LVPS consider the circuit of the standard “boost” converter, so called because its output voltage is generally higher than its input.

The boost converter tends to be the simplest of all switched mode converters. It consists of a single energy storing inductor and no transformers. The inductor is actively driven by a fast acting switch that in practice takes the form of a low capacitance MOSFET transistor. The general scheme of the

boost converter is shown below. At $t=0$ the switch closes as shown in Figure

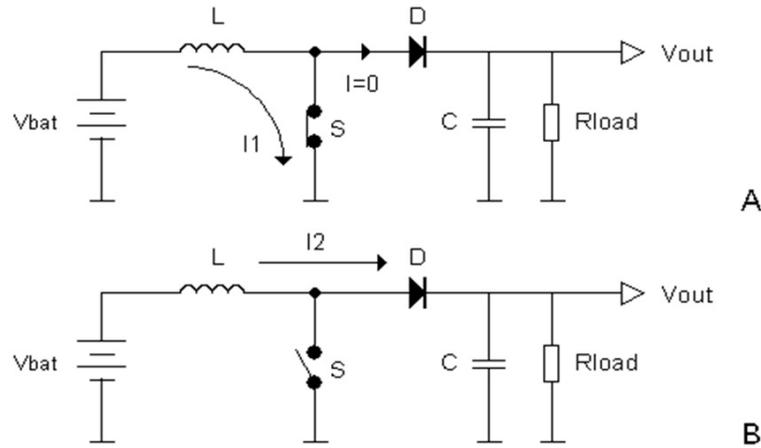


Figure 3.12: Simplified circuit diagram of a boost converter.

3.12(A). As a result the current through the inductor will start to increase linearly according to

$$I = \frac{V_{bat}L}{t}.$$

When the switch is opened, Figure 3.12(B), the current at that moment will have reached a value denoted by I_{pk} . A basic property of an inductor is that it tends to maintain the current flowing through its windings constant according to Lenz's law. With the switch now open, the circuit conditions must then change to satisfy Lenz's law. That is, the inductor forward biases diode D so that the energy built-up in the inductor is dumped into the buffer capacitor C in order to maintain the current. It finds that the capacitor was charged to V_{out} . For the diode to be forward biased the inductor must generate an e.m.f. $(V_{out}-V_{in})$. The current now quickly drops according to

$$I = I_{pk} - \frac{V_{out}}{L}t.$$

This means that it will take a fraction equal to (V_{out}/V_{in}) of the time it took to reach I_{pk} when the switch was closed, to drop again from I_{pk} to 0 now the

switch is open. The cycle then repeats at the operation rate of the switch, f_s . The flyback configuration is usually preferred over the boost converter for the following reason. In a practical boost converter circuit the power switch (MOSFET transistor) will have to handle both a high current when the switch is closed and a high blocking voltage when the switch is open. This places a difficult requirement on the switch specifications, namely a high blocking voltage coupled with low-series resistance, R_{on} . Unfortunately, a transistor that has a high breakdown voltage also tend to have higher R_{on} . Since high series currents and efficiencies are typically demanded of switch mode converters, there tends to be unavoidable energy losses from the switch in the form of joule heating (temperature rise). This lowers operating efficiency. In order to calculate the output voltage of the boost converter in terms of the operating duty-cycle, frequency, load current and input voltage, consider the steady state of operation. That is, under steady state one assumes that the energy stored in the inductor's magnetic field is fully converted into electrical energy in the load. The equations below also apply to the flyback converter [13]. The electrical power dissipated by the load per second is:

$$P_{load} = \frac{V_{out}^2}{R_{load}}. \quad (3.6.1)$$

If T is the period of the switch and D is the switch duty-cycle (fraction of T that the switch is closed), then the maximum current in the inductor is:

$$I_{pk} = \frac{V_{bat}}{L} D T. \quad (3.6.2)$$

The energy per package delivered by the inductor is:

$$P_L = \frac{1}{2} L I_{pk}^2 = \frac{1}{2} \frac{V_{bat}^2}{L} D^2 T^2. \quad (3.6.3)$$

In one second the number of such packages delivered equals $f_s=1/T$. Therefore the total amount of energy delivered per second is:

$$P_{L,tot} = \frac{1}{2} \frac{V_{bat}^2}{L} D^2 T. \quad (3.6.4)$$

In steady-state the amount of energy delivered should equal the amount of energy used, i.e. $P_{load}=P_{L,tot}$. This leads to

$$V_{out} = V_{bat} D \sqrt{\frac{R_{load} T}{2L}}. \quad (3.6.5)$$

3.6.1.2 The flyback converter

The general scheme of the flyback converter is shown in Figure 3.13 where the switch is replaced typically by a MOSFET in practice. Assume that at $t=0$

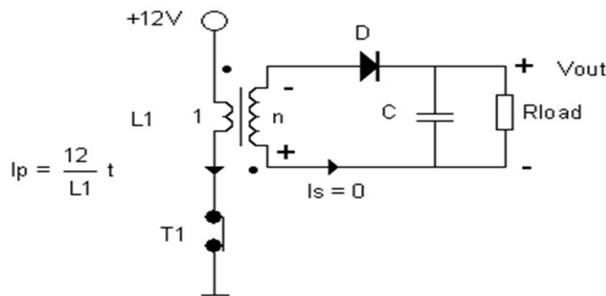


Figure 3.13: Phase one, storing energy in the transformer.

the buffer capacitor is charged to the nominal output voltage V_{out} and that the current through the primary windings of the transformer is zero. At $t=0$ the switch closes and a current starts to flow through the primary winding. This will induce a voltage over the secondary winding with a polarity as indicated. Since the diode is reverse-biased no secondary current can flow. The secondary winding is essentially open-circuit. In other words at the primary side of the

transformer we “just see an inductor”. As a result the primary current will start to increase linearly according to $I=(V_{bat}/L_1)t$. During the time the switch is closed the voltage induced over the secondary windings will be nV_{bat} . This means that the diode must block a minimal reverse voltage of $(nV_{bat}+V_{out})$. At a certain moment, shown in Figure 3.14, the switch is opened. Let the current in the primary winding at the moment just before the switch was opened be I_{pk} . The magnetic energy stored in the inductor at that is then $0.5I_{pk}^2L_1$.

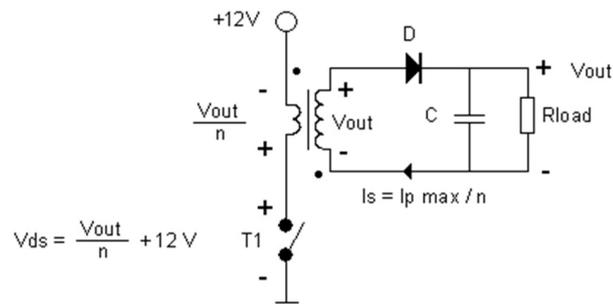


Figure 3.14: Phase two, dumping the energy from the transformer into the buffer capacitor.

Due to the flux linkage between the primary winding and the secondary winding, with the primary circuit open the inductor induces a voltage at the secondary side high enough ($> V_{out}$) to forward bias the diode. The initial value of the current will be $I_2=I_{pk}/n$. During the time that the diode is forward biased, the voltage over the secondary winding will equal $V_{out}+0.7 \text{ V}$. This can also be seen as a transformation of the primary side voltage down to V_{out}/n . The switch therefore has to block a voltage of effectively $V_{bat}+(V_{out}/n)$ when it is open. This is the main advantage that the flyback converter has over the boost converter of comparable input and output voltages, namely the reduced voltage it must handle when it is opened. In the flyback converter the voltage during the off phase is transformed down to a value determined by the ratio of transformer winding turns. This means that a MOSFET with a

much lower R_{on} (i.e. lower breakdown voltage) can be used. Additionally, in the boost converter the diode must carry both the high on current and a high reverse voltage. In the flyback converter the diode at the secondary side only has to block a high voltage while the current is low (I_{pk}/n). This makes it possible to select a diode with smaller capacitances and hence higher switching speed. The consequence is reduced energy losses and an increased efficiency.

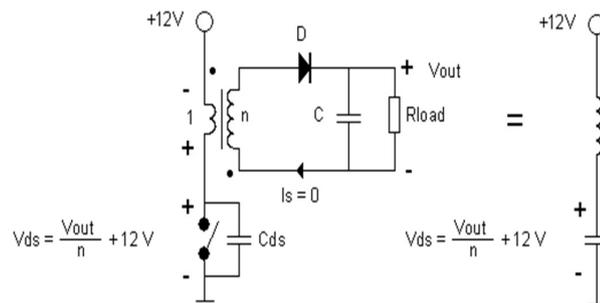


Figure 3.15: Phase three, energy dump completed discharge of drain-source capacitor.

3.6.2 Development of the LVPS

The inverter consists of a single 555 timer, an assembled transformer, the IRF640 MOSFET, BC547 transistor, schottky diodes and capacitors. The NE555 monolithic timing circuit is a highly stable controller capable of producing accurate time delays or oscillation. In the time delay mode of operation, the time is precisely controlled by one external resistor and capacitor. For astable operation as an oscillator, the free running frequency and the duty cycle are both accurately controlled with two external resistors and one capacitor. The circuit may be triggered and reset on falling waveforms, and the output structure can source or sink up to 200 mA. In this application it is used in the astable mode. IRF640 is a power MOSFET with $R_{DS,(on)} = 0.150$

Ω , extremely high dV/dt capability, very low intrinsic capacitances and gate charge minimized. It can be used for high current switching, un-interruptible power supply (UPS), DC/DC converters for telecommunication, industrial, and lighting equipment. In this project, IRF640 was used for switching high current. BC547 is NPN general purpose transistor with a low current (maximum. 100 mA) and low voltage (maximum. 65 V).

3.6.3 Simulation and performance of the LVPS

3.6.3.1 Simulation

The USB has maximum DC voltage of 5V, and from the simulation results it can be seen that the circuit was able to produce the required minimum of $\pm 6V$, alongside the 5V of the USB port itself. It generates adequate power to supply the low-voltage sections of the TL instrument, the PIC controller (+5 V), the analogue amplifiers ($\pm 12V$) which comprise the temperature sensors, the intensity signal conditioning circuit and various voltage references. Figure 3.16 shows the schematic diagram of simulation using LT-Spice and its results are shown in the Figure 3.17.

3.6.3.2 Actual performance of the LVPS

The prototype of the LVPS was built and adjusted to output $\pm 12V$. In the final printed circuit board (PCB) version of the TL instrument a further refinement was made by adding a 78L05 and 79L05 regulator to produce $\pm 5V$ for the operational amplifiers. The performance of the LVPS was found to be stable and reliable. The next chapter describes the design of the proportional-integral-derivative (PID) temperature controller. It was thought to present this design in a separate chapter because of the level of detail it required. Its

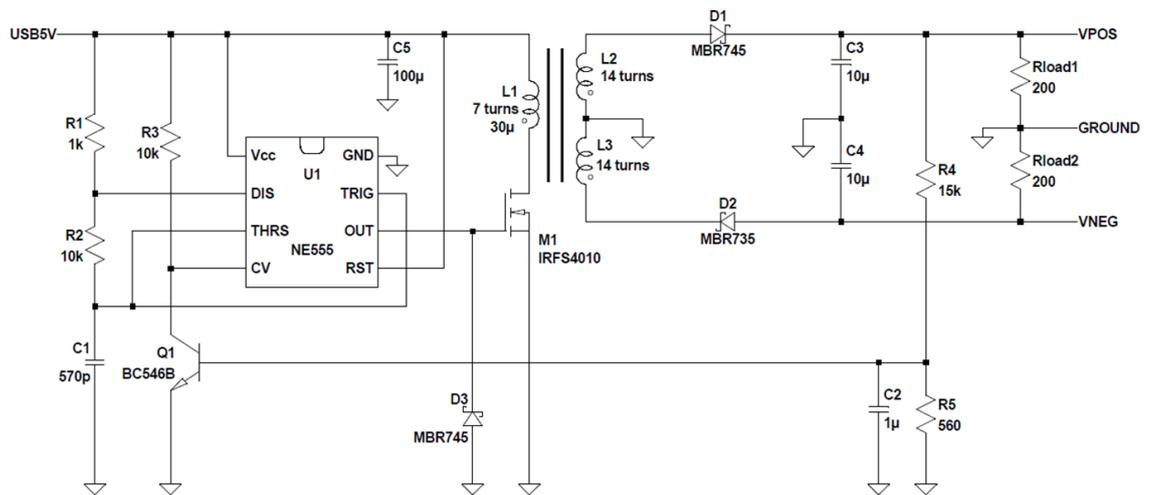


Figure 3.16: Schematic diagram of USB to rail power inverter.

design is also unique in that with the exception of the heater driver itself, much of the control algorithm of the PID controller is implemented in the firmware of the micro-controller. The feedback control algorithm is based on digital principles using the z-transform.

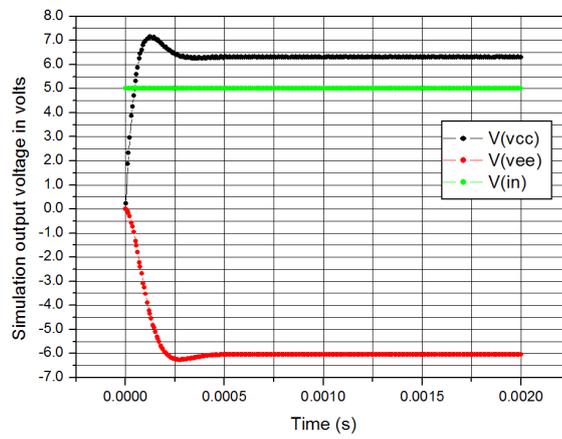


Figure 3.17: Predicted USB to rail power inverter results.

Chapter 4

High-resolution PID temperature controller

4.1 Introduction

To assist their practical studies of TL several independent researchers have offered alternative solutions to commercial instrumentation with the main impetus being low cost, simplicity of usage or suitability for their specific purposes [22, 25, ?, 41, 40, 43]. While TL is simple in principle, in practice there are many technical instrumentation challenges that must be identified and solved. The control of temperature over wide ranges (though typically below 400°C) with good measurement resolution and accuracy for the small dimensioned samples can be difficult, more so if the samples are in powder form. Other issues stem from nonlinearities, output drift, detector correlation errors and so on. The alternatives encountered in the literature have diverse attributes, for example arbitrary, wide range heating, profiled excitation such as logarithmic heating, “automatic” control of heating, and even open-loop “bang-bang” servos. The heating apparatus themselves are just as varied as are the intensity sensing apparatus also. Several of the foregoing designs have used third-party hardware proportional-integral (PI) or proportional-integral-derivative (PID) control of the heating element, notwithstanding this

detraction from true low-cost, our point of contention in this report is that our own experiences with the open-loop, small heater “plant” have shown that a reported temperature regulation technique often can not have the claimed accuracy, even at the lowest heating rates [30, 31]. For example, in [41], the stated temperature accuracy is erroneously the bit resolution of the analog to digital converter (ADC) itself in open-loop. The design of a temperature controller using a digital PID algorithm implemented as part of the firmware of the PIC18F2520 controller is reported. The output element is a 100 watt resistive soldering iron driven directly by a 2SK3115 power MOSFET using pulse-width modulation (PWM) at 10 kHz, with temperature feedback provided by a K-type thermocouple referenced to ambient temperature. The heater power is derived from a locally built toroidal mains transformer rectified to supply direct current of 0.6 amperes at 180 V. Varying the duty cycle in the algorithm from 0 to nearly 100 % allows a wide range of temperature for the small heater. We report on the considerations of the design, ranging from the identification of the open-loop heater plant to optimization simulations and then to the coding of the digital algorithm. Finally, we present experimental data obtained from actual performance of the heating arrangement for a typical heating run.

4.2 System identification

4.2.1 The heater and feedback arrangement

A full description of the overall TL system can be found in [30, 31, 38]. For the purposes of designing the temperature controller and a digital PID algorithm, only a small part of the overall system will suffice. That is, the aluminium sample holder, a type K-thermocouple (Chromel-Alumel) for temperature

feedback, and the ambient-compensated temperature conditioning circuit. The K-type thermocouple has an effective Seebeck coefficient of about $41 \mu\text{V}$ per degree change in temperature. A system is considered to be open-loop if a control signal sent out simply hopes to achieve a desired action, known as the set point, without any sense of what the instantaneous value of the action actually is. In the present context it can be looked at as the heater simply being turned on in some manner without temperature feedback. It is important to know the various time parameters of the open-loop plant associated with certain types of control input. For example, the times associated with the attainment of a new steady state when a control input is suddenly changed from one value to another. Loosely speaking, these responses convey a sense of the “sluggishness” of the plant and ultimately how best to ply its input.

4.2.2 The software PID temperature controller

In PID temperature controller the control signal to the heating element, $u(t)$ is derived from the feedback from past and present temperature [20]. The controller first determines the error difference signal $e(t)$ that indicates the difference between the target and the current temperature. It then generates a control signal that is a sum of three quantities, one proportional to $e(t)$, another dependent on the time accumulated (integral) error and another dependent on how fast the error is changing with time i.e. the error derivative. Mathematically,

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt}, \quad (4.2.1)$$

where K_p is the proportional gain, T_i and T_d are the integral and derivative time constants respectively. Transforming Equation(4.2.1) and defining a

common forward gain K_p in Laplace transform space gives the PID controller:

$$C(s) = K_p \left[1 + \frac{1}{T_I} + T_D s \right] \quad (4.2.2)$$

Figure 4.1 shows the PWM actuator used. The gate of the MOSFET was driven at a software generated 10 kHz, forming the input of the controlled plant. Therefore, an input is subsequently any fixed amplitude (i.e. 5 V), 10 kHz gate input distinguishable from other inputs by its duty-cycle only. A true “unit step” input was then a duty-cycle change from 0 to 100 %. In practice a smaller step range can also be used to characterize the plant. To implement Equation (4.2.1), the time constants T_i and T_d were determined using the Ziegler and Nichols tuning method (ZNM) [53, 4] on the open-loop plant. Converting the system to z-transform space allowed digital implementation on the PIC18F2520 micro-controller.

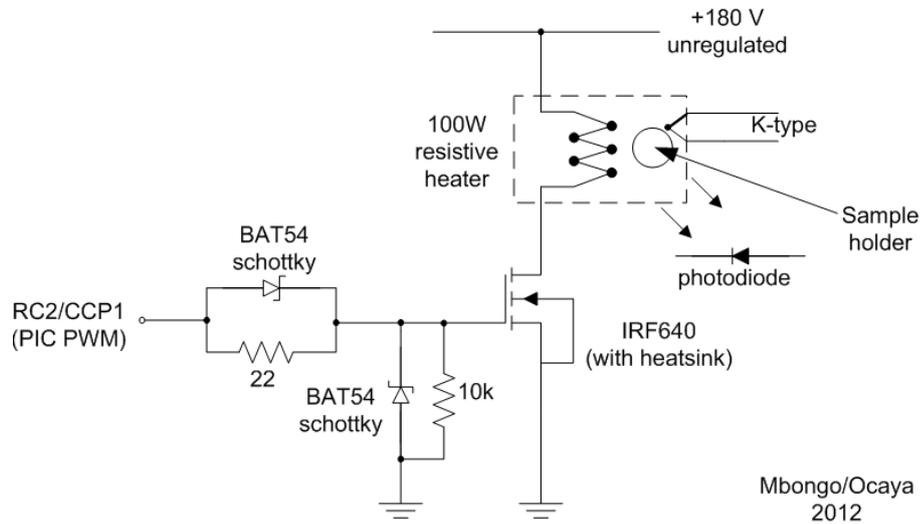


Figure 4.1: Diagram of the PWM heater driver.

4.2.3 Realization of a discrete PID controller

The Ziegler-Nichols(ZNM) method can be used in both open- and closed-loop configurations to ascertain the parameters of a plant thereby enabling a reasonable controller to be defined. In the present work the response to a step input on the open-loop plant was found to be sufficient. According to ZNM [53] the open-loop unit step response, $G(s)$, can be approximated by the Laplace transform

$$G(s) = \frac{K e^{-sT_d}}{sT_1 + 1} \quad (4.2.3)$$

with the parameters K (d.c. forward gain), T_d and T_1 as in Figure 4.2.

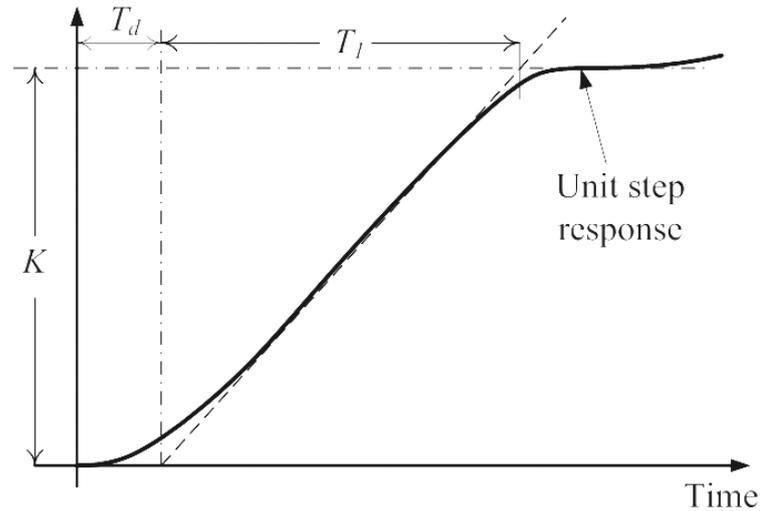


Figure 4.2: Parameters of interest in Ziegler-Nichols open loop tuning.

By direct experimentation the parameters for the heater plant (sample holder and heating arrangement) were estimated graphically to be $T_d=4.0\text{s}$, $K=4.4$ and $T_1=184\text{s}$. The ZNM method suggests that the parameters of Equation(4.2.2) for a starting PID controller for the plant response in Equation (4.2.3) are:

$$K_p = \frac{1.2T_1}{KT_d} \approx 1.882, \quad T_I = 2T_d = 150\text{s} \quad \text{and} \quad T_D = \frac{1}{2}T_d \approx 6.67\text{s}. \quad (4.2.4)$$

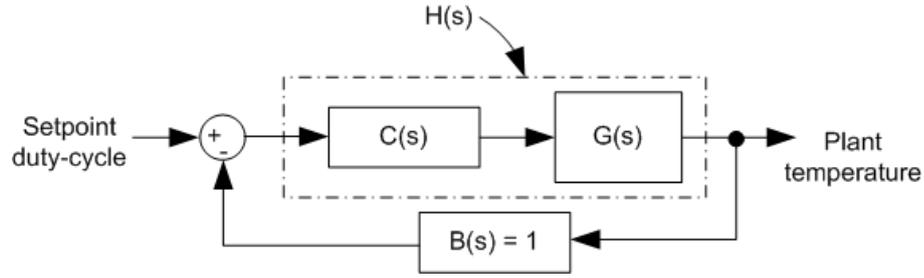


Figure 4.3: Block diagram of the implemented temperature controller. The closed-loop function can be written as $H(s)=C(s)G(s)$.

Substitution of the parameters in Equation(4.2.4) into Equation (4.2.2) gives a PID controller $C(s)$ that has good transient response.

$$C(s) = 12.55 \left[\frac{s^2 + 0.15s + 0.001}{s} \right]. \quad (4.2.5)$$

The MATLAB/Simulink model, Figure 4.4, was obtained from $H(s)=C(s)G(s)$. The simulated performance of this function against the actual experimental data is shown with the results Chapter 5, in Figure 5.3(a). The transport delay block models the delay T_d in Equation(4.2.3). To implement the controller

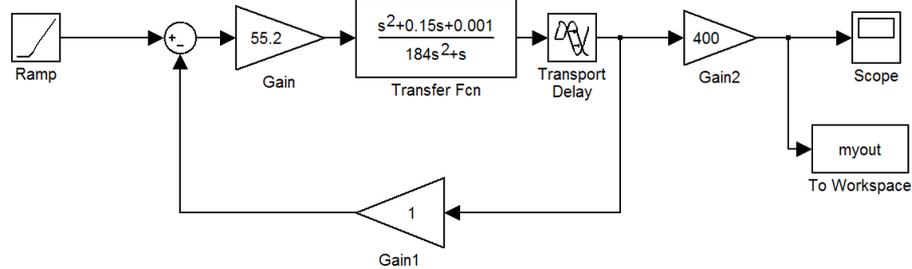


Figure 4.4: MATLAB/Simulink model of the closed-loop system that gives good transient performance. The input shown is a linear ramp with input duty cycle starting from 0 to 100%.

using a digital PID algorithm running on the PIC18F2520 firmware, it was necessary to convert the controller to the sampled time domain. The techniques of the z-transform were readily applied [51], though a full description of the technique is beyond the present scope. The PID controller in Equation (4.2.2)

has the “velocity” z -transform form in sampled time $t \in 0, T, 2T, \dots$ given by

$$C(z) = K_p \left[1 + \frac{T}{T_I(1-z^{-1})} + T_D \frac{(1-z^{-1})}{T} \right] = a + \frac{b}{1-z^{-1}} + c(1-z^{-1}), \quad (4.2.6)$$

where $a=K_p$, $b=K_p T/T_I$ and $c=K_p T_D/T$. The parameter T is the sampling interval of the converter. It is roughly equal to the firmware looping or polling time. The result is the parallel PID implementation in Figure 4.5 which can readily be coded into the micro-controller firmware (Appendix B). The

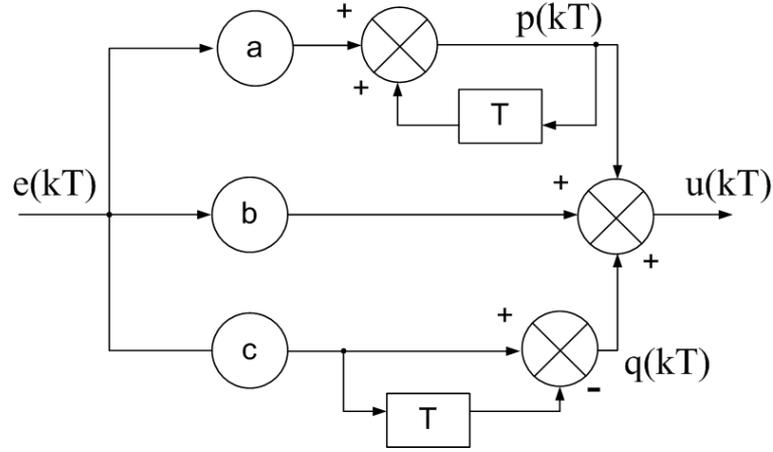


Figure 4.5: Parallel implementation of the PID TL-system heater controller.

controller output $u(kT)$ is the PWM MOSFET gate drive. The sampled PID equations, Equation(4.2.7), were implemented on the PIC device in mikroC code [34].

$$\begin{aligned} p(kT) &= b e(kT) + p(kT - T) \\ q(kT) &= c e(kT) - c e(kT - T) \\ u(kT) &= p e(kT) + a e(kT) + q(kT) \end{aligned} \quad (4.2.7)$$

The firmware implementation of the digital PID algorithm can be seen in the listing in Appendix B.

In this chapter the sample temperature feedback and heater circuitry have been outlined. We have shown that the time responses of the arrangement must be properly quantified for accurate temperature measurement. From

this information a good PID controller can then be derived and implemented digitally in the micro-controller firmware, with a response typified by Figure 5.3(a). Linear heating was possible by ramping the control input from the software through the output range. Heating rate is known to affect observed results [15], but simulations show that rates exceeding 20 °C/min are possible with the designed controller. Figure 5.3(b) shows the simulated output of the closed-loop plant in response to ramping duty-cycle input. The innovative use of the digital PID algorithm allows ease of tuning the controller by allowing minor adjusting the parameters in the firmware. Finally, the present approach to low-cost TL instrument design is significant in many respects and ultimately allows higher accuracy and finesse in glow peak separation on the temperature axis.

Chapter 5

Results of sample measurements on the TL system

5.1 Introduction

The goal of creating a TL-instrument using common components has been largely met. This chapter presents the results of tests on the temperature controller as well as a simple actual run of the prototype TL instrument on a common sample compound. We show that the physically small plant has a transfer function that needs to be known precisely if controlled heating of samples is to occur, as in Figure 5.2. Precise characterizations of the actual heating arrangement are rarely reported in the comparable TL instruments in the literature.

5.2 Photograph of the prototype TL instrument

The foregoing elements of the TL system were integrated into a single design. A double-sided PCB was then designed and constructed on fibre-glass with both through-hole and surface-mount components. A photograph of the basic TL system design is shown in Figure 5.1. More details of its construction such

as schematics can be found in Appendix A.

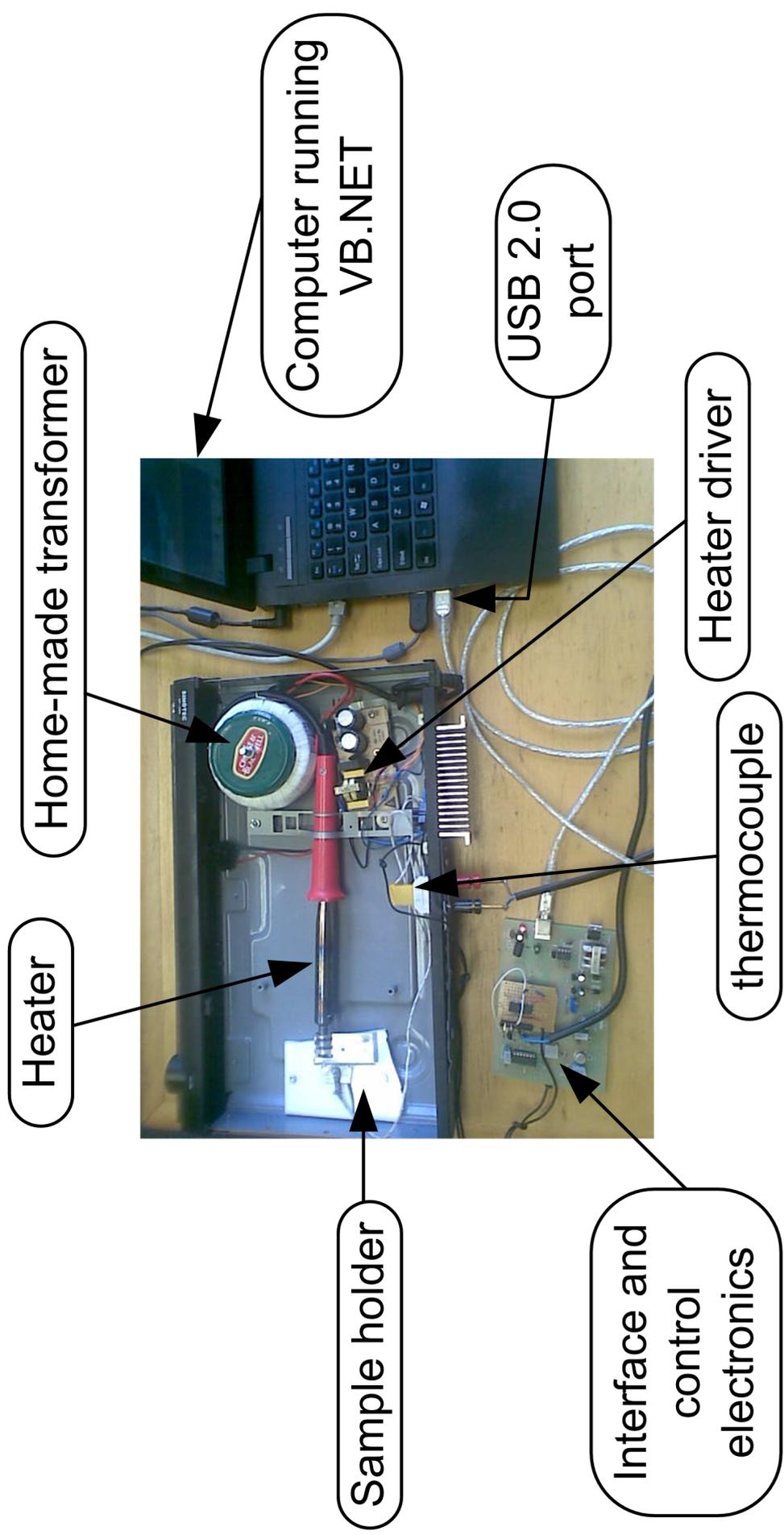


Figure 5.1: Labeled photograph of the TL instrument prototype.

5.2.1 Unit step responses of the TL instrument

Figure 5.2 shows the actual and open-loop performance of the sample holder when subjected to the 100 W heater. The heater was turned fully on and the temperature was monitored until steady state was reached. Using the data from the actual heating, the mathematical plant model of the heater was then obtained, as shown in Equation (4.2.4) in Chapter 4.

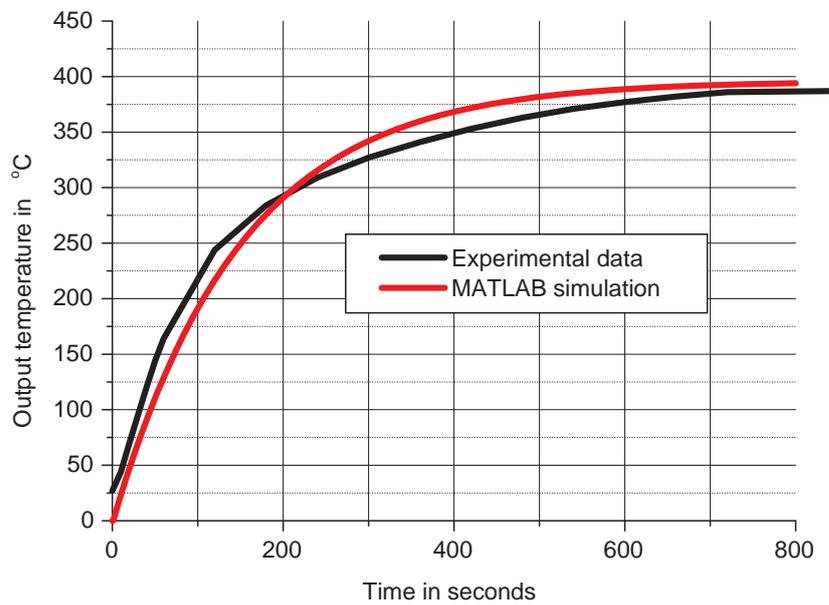


Figure 5.2: Experimental and simulated open-loop unit step response based on Eq. 4.2.3.

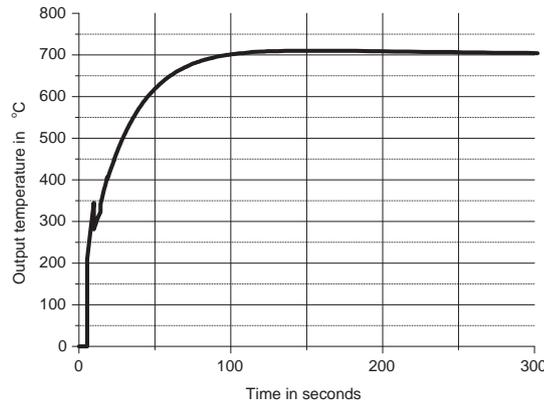
5.2.2 Closed-loop unit step response

To evaluate the closed-loop step response an instruction was issued from the Microsoft Windows control program to set a temperature of 700 °C. Figure 5.3(a) shows the closed-loop unit step from data logged on the host computer. The graph shows that the target temperature is reached within 100 s of the

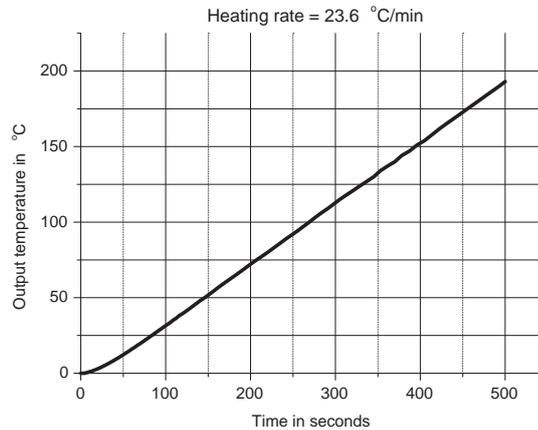
command, the peak results from overshooting of heater's temperature.

5.2.3 Demonstration of linear heating

Linear heating is one of several possible heating profiles that the designed TL instrument is capable of. In future implementations of the control software,



(a) Closed-loop unit step response under scaling showing good transient response.



(b) A graph showing a linear heating with a ramped duty-cycle input of 0.06 per min with Pearson's coefficient of linearity $R^2=0.999$.

Figure 5.3: Heating behavior under PID control.

arbitrary heating regimens can be coded. Figure 5.3(b) shows the response to a linear ramp command issued from the host GUI program for a heating rate of 23.5 °C per minute.

5.2.4 Demonstration of intensity measurement

The measurement of intensity was done in two ways. First, the TL instrument was run without a sample. The ADC values obtained this way are the baseline, shown in Figure 5.4(a)-(b). Second, a ceramic sample composed of an old piece of clay pottery found nearby and broken further to fit the sample holder was heated using the linear heating profile. The result of this process is shown in Figure 5.4(a). The presence of the baseline readings in the measured intensity sample meant a correction scheme had to be devised. The trend in the baseline data was determined using Microsoft Excel to be approximated by a number of functions, such as an order-four polynomial, and also a simpler natural logarithm function. For simplicity, the latter function was decided upon. The motivation for this choice is that if the correction using the baseline function were to be computed in real-time (as the ADC data is being acquired) then a simpler function would be handled faster. Additionally, it would not affect the timings of both the PID algorithm and the USB communications that also need to be implemented in real-time. The best baseline intensity versus temperature fitting function was found to be

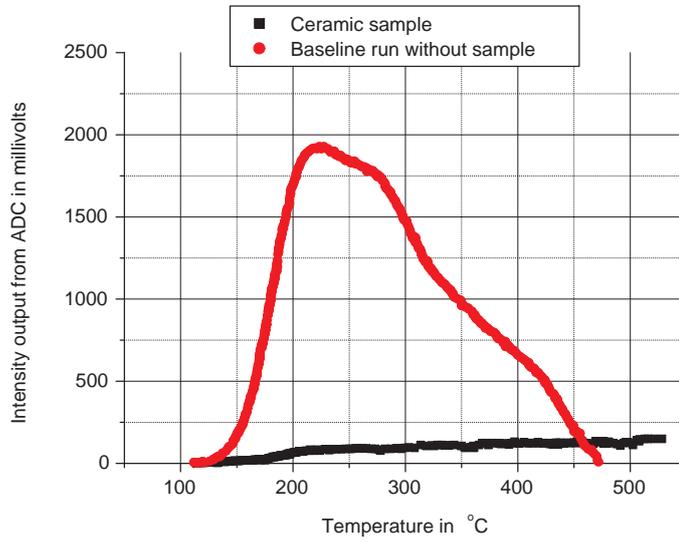
$$I_{BL} = 96.03 \ln(T) - 456.8 \quad (5.2.1)$$

I_{BL} is in mV when T is in °C. Therefore, for each measured sample intensity I_{ADC} at measured sample temperature T_S , the baseline corrected intensity I_C

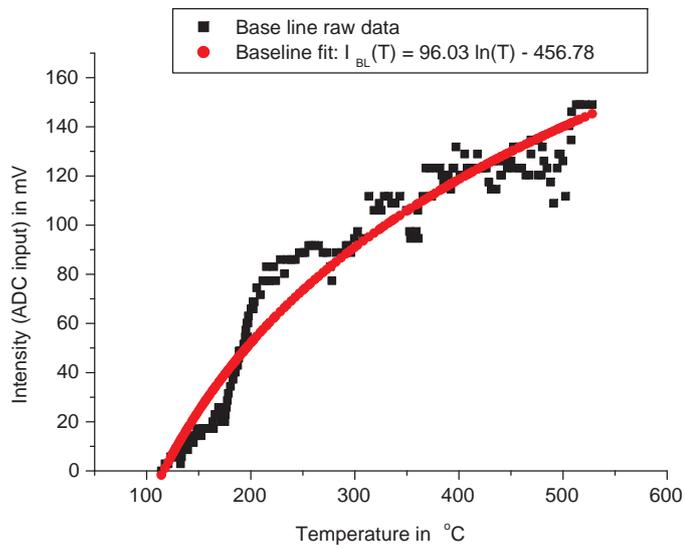
for the sample is given by

$$I_C = I_{\text{ADC}} - I_{\text{BL}}. \quad (5.2.2)$$

Figure 5.5 shows the corrected plot for the ceramic sample. The intensity of a TL instrument is usually expressed in arbitrary units, (a.u.). In this particular implementation each a.u. corresponds to 1 mV.



(a) TL test with and without a ceramic sample, showing the baseline.



(b) Baseline plot showing logarithmic fitting function.

Figure 5.4: Baseline measurement (a) with ceramic sample (b) showing fitting function.

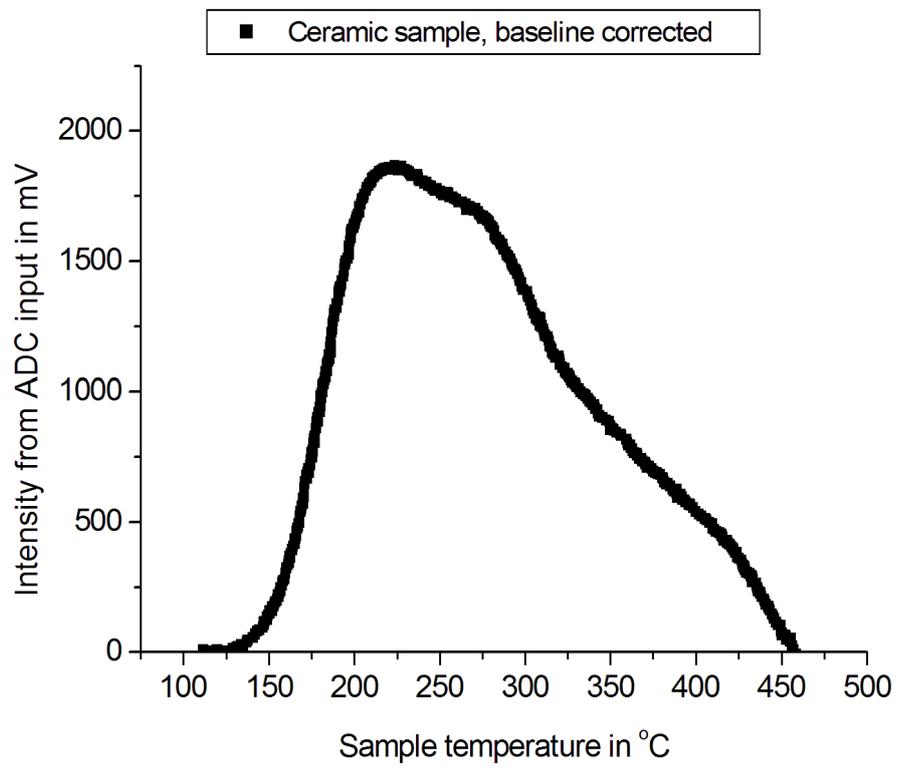


Figure 5.5: Plot of the baseline corrected measurements on the ceramic sample.

Chapter 6

Conclusions

The study began with the fundamental question of whether a low-cost TL system could readily be built. There were two main motivating factors behind this question. First, the apparent ease with which several researchers have reportedly devised low-cost alternatives for TL measurement. Second, the limited time and accessibility of the existing and expensive TL instrument in the department. A variation in the theme in the quest for a low-cost instrument, one which also puts this particular research into a unique bracket, is the possibility of using a P-I-N photodiode as the intensity sensing element instead of the customary cooled photo-multiplier tube. A survey of the literature has shown that this has not been attempted before. On the outset the task of creating a working TL instrument appears trivial, but as the research quickly revealed, it can be formidable. The competing requirements of the functional parts of a TL system need to be carefully considered, with particular attention being given to the temperature sensing and making it as independent of ambient temperature as possible, the detection of the low-light levels emitted by TL processes in the sample, the conditioning circuits and the data acquisition of all the parameters of measurement. These issues gave rise to significant challenges and hurdles that had to be overcome. For instance, many of the reported TL instruments have used open-loop control for the temperature.

Our preliminary tests showed that the repeatability of setting the temperature is generally poor and the heating regimes are not necessarily linear, contrary to the some of the claims in the literature. This motivated the development of the PID temperature controller with variable degrees of tuning and the possibility of linear sample heating. The temperature range of the designed controller was vastly wider than other controllers, by almost 100 K. Other significant challenges involved the creation of an isothermal block for the photodiode to eliminate the effect of temperature during a sample characterization run. The versatility of a scientific instrument rests somewhat on its user friendliness. The creation of a convenient and friendly graphical user interface (GUI) using USB, which has also not been reported for such low-cost TL instruments, was taken to be a necessary task. Writing a friendly GUI that inherently provides logging of sample temperature and intensity data in a convenient format on the controlling modern computer system posed significant challenges that had to be solved. To solve these challenges, new software and hardware tools had to be assimilated and mastered before and during the research undertaking. Although not formally presented in this dissertation, many aspects of these tools or their outputs can be found in the Appendices. The most significant of the software tools were MATLAB, Simulink, OrCAD/LTSpice, Visual Basic programming for the .NET Microsoft frameworks, MPLAB Assembler, mikro-elektronika's mikroC programming environment for the PIC micro-controller, and ICProg that loaded the firmware into the PIC device. The firmware, listed in Appendix B, handles both the data acquisition and USB communications between the TL instrument and the controlling host computer. To create the printed circuit board the DIPTrace program had to be acquired and mastered. The hardware tools comprised a locally made PIC programmer, a printed-circuit board facility and a solder station. The schematics presented in the

appendices were created in this environment. An attestation to the proficiency gained in the usage of these tools is the culmination of the working prototype of the TL instrument, that was demonstrated for one sample in Chapter 5.

Finally, the main theme of the work presented above pertains to the evaluation and quantification of the performance of the P-i-N diode sensor in the role of intensity sensing for the emissions in a thermoluminescence process. The approach taken in attempting to answer the fundamental question has been difficult but the rewards have been in the lessons learnt and the resulting demonstrable prototype of a TL system. However, there is still much scope for future work. Further quantification studies of the thermal noise of such a sensor, the effect of lowering the temperature of the sensor even further by means other than by Peltier effect devices using liquid nitrogen, for instance, opens up some avenues for future research. The scope for future work with respect to temperature control remains to write algorithms for arbitrary heating regimes that can be initiated from the Microsoft Windows graphical user interface of the control program. The gradual appearance of solid state photomultipliers offers further potential to improve the intensity detection apparatus significantly, without driving costs too much higher. Aspects of the work presented herein have been presented at conference gatherings [30, 31, 38].

References

- [1] Hamamatsu Photonics 1998, *Photomultiplier Tubes: construction and operating characteristics and connections to external circuits*, (last accessed - August 2011).
- [2] Vishay Telefunken 1999, *BPW21R Silicon PN photodiode*, (last accessed - November 2012).
- [3] A. Ben-Zvi A. Halperin, A.A. Braner and N. Kristianpoller, *Phys. Rev.* **117** (1960), 416.
- [4] K.J. Åström and T. Hägglund, *PID controllers-theory, design, and tuning.*, 1 ed., 2nd edition, ISA, 1995.
- [5] F. Urbach *Wiener Ber*, **139** (1930), 363.
- [6] V.J. Bortolot, *Products Description Model 1100 Automated TL system: Daybreak Nuclear and Medical Systems*, USA (2000).
- [7] P. Braunlich, *Radiat Prot. Dosimetry* **34** (1990), 345–351.
- [8] R. Chen and S.A.A. Winer, *J. Appl. Phys.* **41** (1970), 5227.
- [9] E.M. Yoshimura C.M. Suntai, R.N. Kulkarnii and E. Okuno, *Radiat Prot. Dosimetry* **65** (1996), 21–24.
- [10] Burr-Brown Corporation, *Designing photodiode application circuits with OPA128*, (last accessed - November 2012).

- [11] ———, *OPA111: Low noise precision FET operational amplifier*, (last accessed - November 2012).
- [12] Microsoft Corporation, *Microsoft corporation, .net downloads, developer resources & case studies*, (last accessed - November 2011).
- [13] R. Dekker, *Flyback converters for dummies*, (last accessed- February 2012).
- [14] C. Furetta and P. Weng, *Operational thermoluminescence dosimetry*, World Scientific publishing, Singapore, 2003.
- [15] J Papadopoulos G. Kitis, I Spiropulu and S. Charalambous, N tcl. Instr tm. Methods Phys.Res. **73** (1993), 367–372.
- [16] Dussel G.A. and Bube R.H., **155** (1967), 764.
- [17] G.F.J. Garlick and A.F Gibson, **60** (1948), 574–590.
- [18] G. Hall, *Semiconductorsensors.pdf*, (last accessed - November 2012).
- [19] W. Hoogenstraten, Philips Res. Rep. **13** (1958), 515.
- [20] D. Ibrahim, *Microcontroller based temperature monitoring and control*, Newnes, 2002.
- [21] H.W. Julius, Radiat Prot. Dosimetry **17** (1986), 267–273.
- [22] G.A. Appleby J.W. Quilty, J. Robinson and A. Edgar, Rev. Sci. Instrum. **78** (2007), 083905.
- [23] J.C. Kelly and A. Smith, Joint Europn Study Group on Phys, Chem. Techn. Appl. to Archaeology **PACT 2** (1978), 66.
- [24] Grosswiener L.I., **24** (1953), 1306.
- [25] V.I. Lyamayev, Meas. Sci. Technol. **17** (2006), N75.

- [26] B. Poty M. Menager and E. Roth, *Nuclear methods of dating*, Springer, 1989.
- [27] E.P. Manche, Rev. Sci. Instrum. **49** (1978).
- [28] ———, J. Chem. Edu **56** (1979), A303.
- [29] C.E. May and J.A. Partridge, **40** (1964), 1401–1409.
- [30] M. Mbongo and R.O. Ocaya, *Design of a high-resolution PID temperature controller for use in a low-cost thermoluminescence system*, The 56th SAIP, Pretoria, South Africa. 12-16 July **56** (2012).
- [31] ———, *A low-cost thermoluminescence system for use with .net computing environments*, The 56th SAIP Annual conference, Pretoria, South Africa. 12-16 July **56** (2012).
- [32] S.W.S. McKeever, *Thermoluminescence of Solids*, Cambridge University Press, 1988.
- [33] Microchip, *Pic18f2420/2520/4420/4520 enhanced flash microcontrollers*, (last accessed - May 2012).
- [34] mikroElektronika, *mikroc pro for pic microcontrollers, mikroc pro v3.2*, (last accessed August 2011).
- [35] L.N Agersnap M.S Akselrod and S.W.S McKeever, **84** (1998), 3364–3373.
- [36] P. Neelamegam and A. Rajendran, J. Inst. **1** (2006), T05001.
- [37] K.H. Nicholas and J. Woods, Brit. J. Appl. Phys **19** (1964), 783.
- [38] R.O. Ocaya and M. Mbongo, *A small-plant PID temperature controller for thermoluminescence measurement.*, IEEE AFRICON 2013, Mauritius. **8-13th September** (2013).
- [39] K. Osada, J. Phys. Soc. Japan **15** (1960), 145.

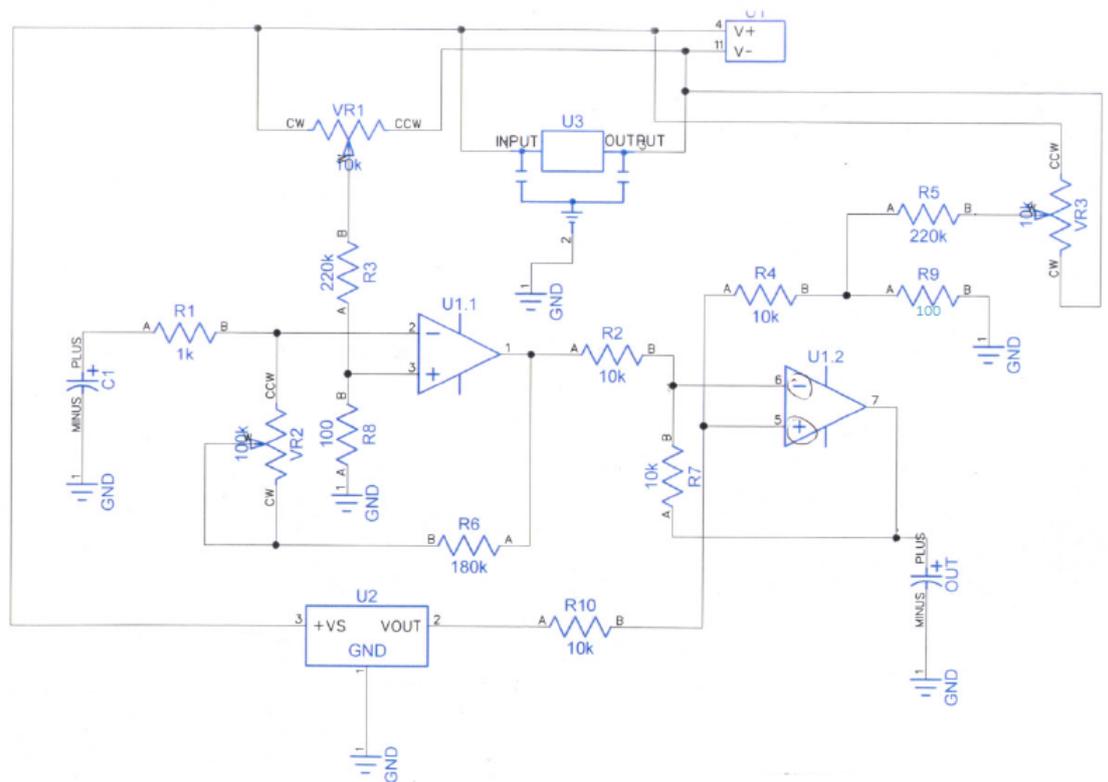
- [40] E. Caselli-M. Lester P. Molina, M. Santiago and F. Spano, *Meas. Sci. Technol* **13** (2002), N16.
- [41] K. Padmanabhan P. Neelamegam and S. Selvasekarapandiang, *Meas. Sci. Technol.* **3** (1992), 581–587.
- [42] S.A Petrov and I.K Bailiff, **27** (1997), 185.
- [43] H.S. Vora-V.K. Dubey K.K. Sarangpani N.D. Shirke R. Bhatnagar, P. Saxena and S.K. Bhattacharjee, *Meas. Sci. Technol.* **13** (2002), 2017.
- [44] S.W.S. McKeever R Chen, *Theory of Thermoluminescence and Related Phenomena*, World Scientific, Singapore, 1997 (29-35).
- [45] J.T. Randall and M.H.F. Wilkins, **184** (1945), 366–389.
- [46] T.S.S. Sing R.K. Gartia, S. Ingotombi and P.S. Mazumdar, *On the determination of the activation energy of a thermoluminescence peak by the two-heating-rates method*.
- [47] Scribd, *Chapter 6 - photodetectors*, (last accessed - November 2012).
- [48] A.S. Sedra, *Microelectronic circuits*, Oxford University Press.
- [49] National Semiconductor, *Lm124/lm224/lm324/lm2902 low power quad operational amplifiers*, (ast accessed - June 2012).
- [50] C.S. Shalgaonkar and A.V. Narlikar, *J. Mater. Sci* (1972) **7** **7** (1972), 1465–71.
- [51] K.A. Stroud, *Advanced engineering mathematics*, 4 ed., Industrial Press, NY, 2011.
- [52] H.J. van Es, *Thermoluminescence dating of sediments using mineral zircon*, (last accessed - May 2012).
- [53] J.G. Ziegler and N.B. Nichols, *Optimum settings for automatic controllers*, (Trans. of the ASME. **64** (1942).

[54] D.W. Zimmerman, *Archaeometry* **13** (1971), 29–52.

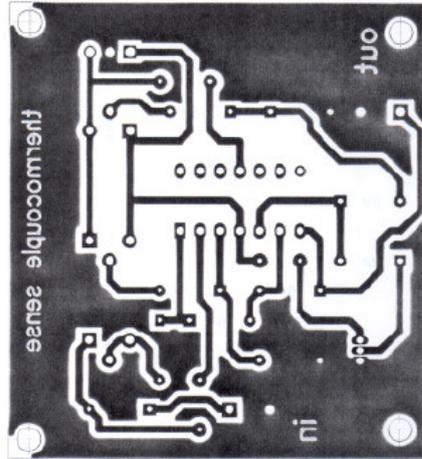
Appendix A

Sample thermometer development

A.1 Schematic diagram of first prototype

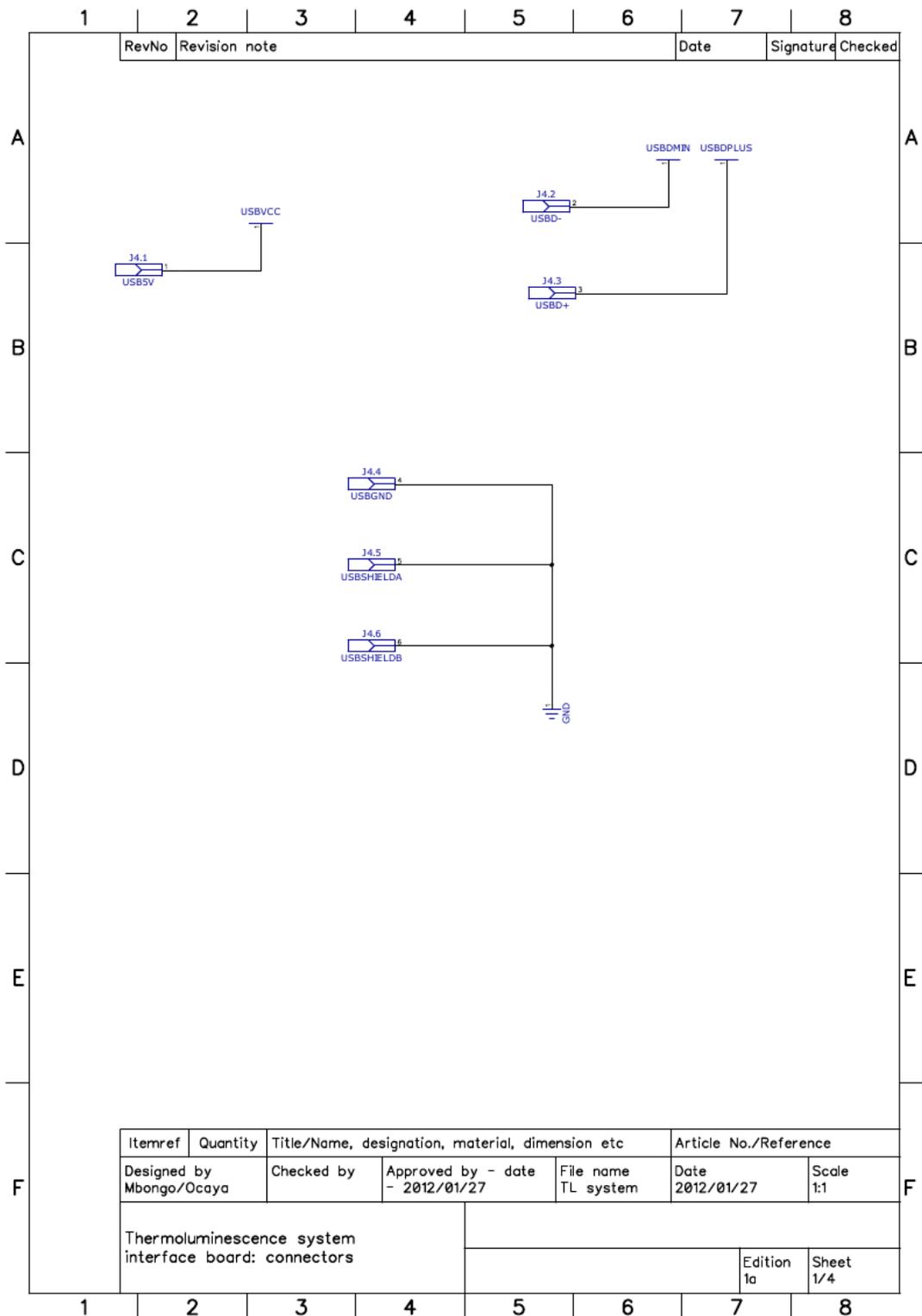


A.2 Sample temperature sensor evaluation PCB

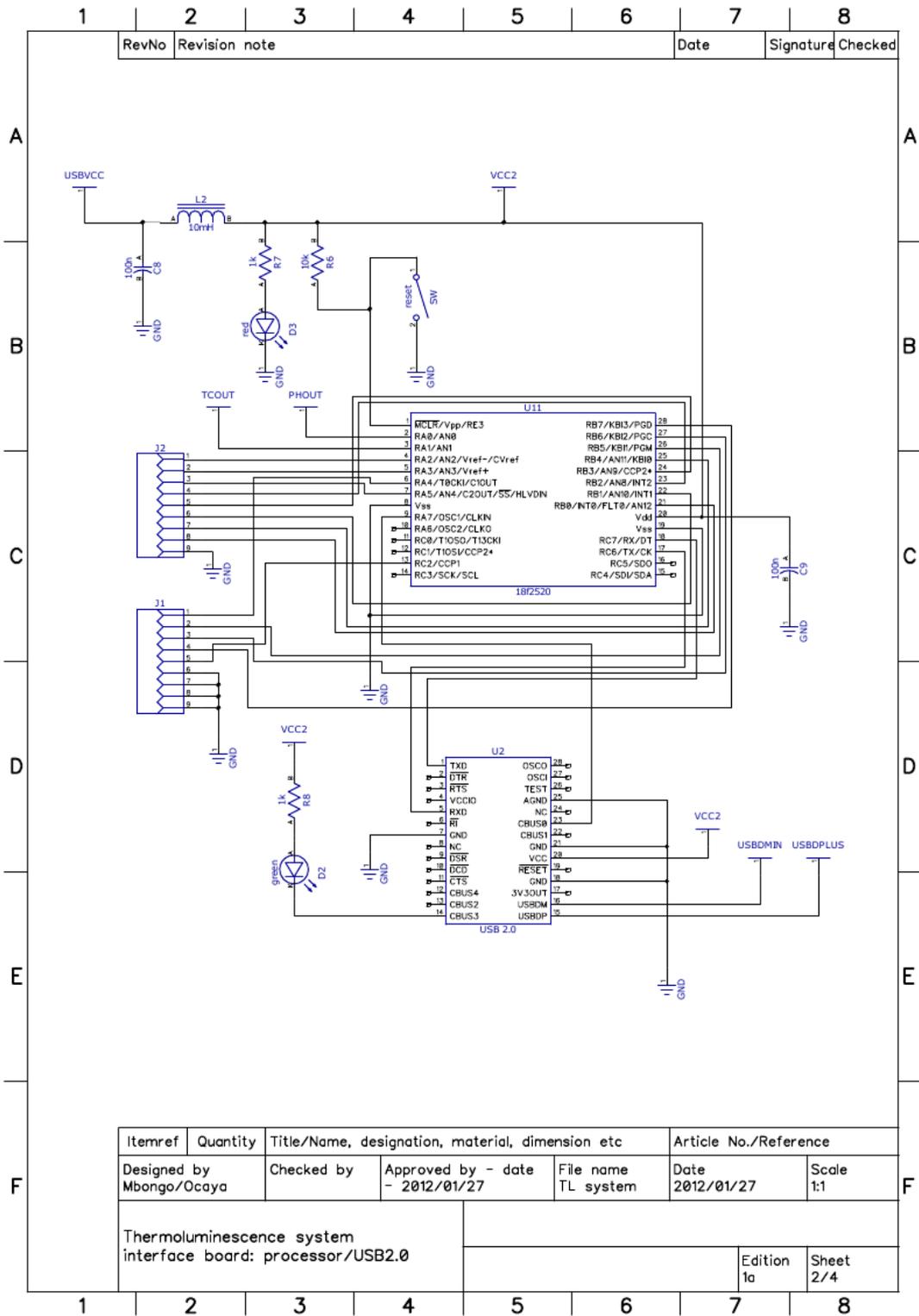


A.3 Final TL system schematics

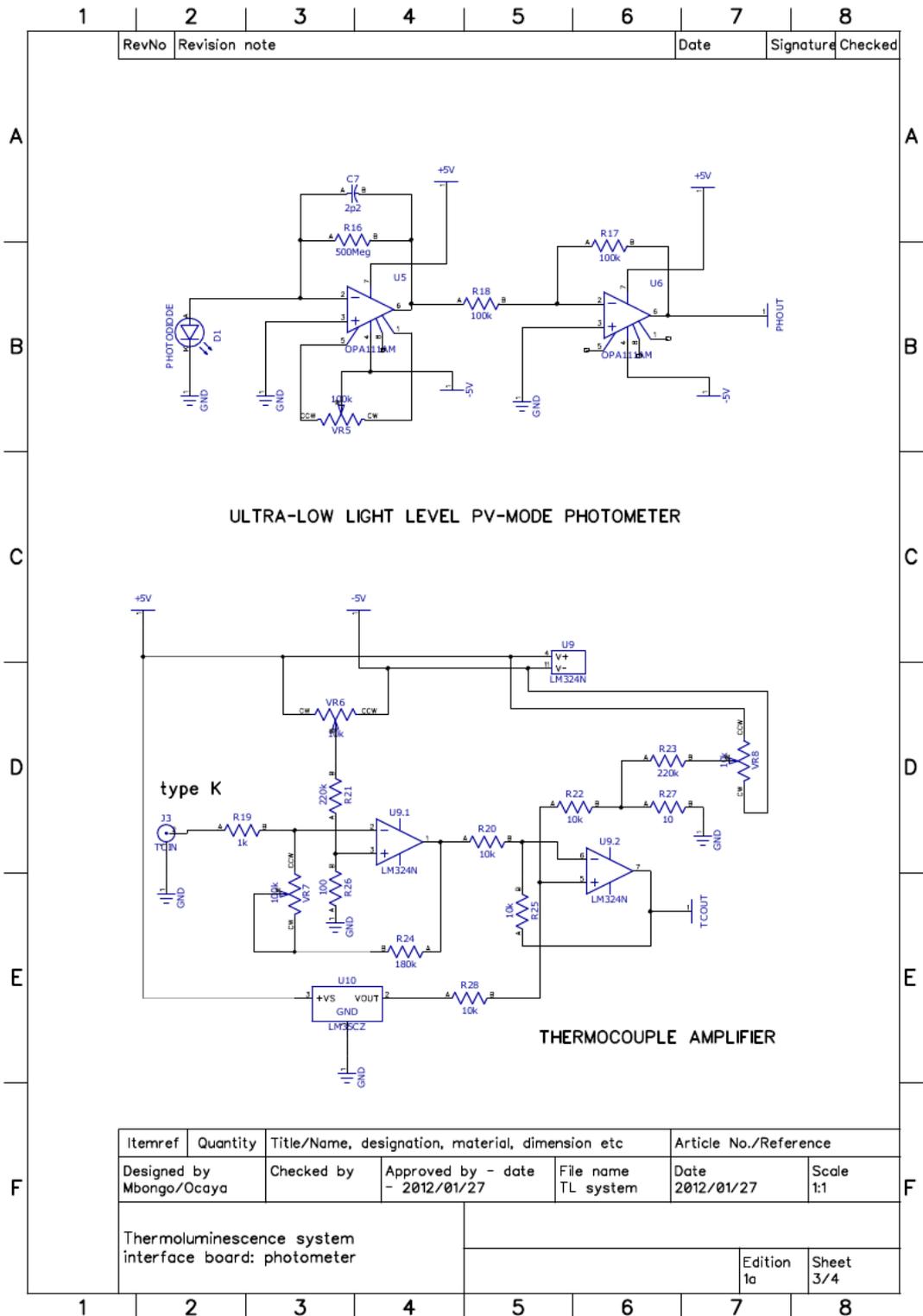
A.3.1 Connectors



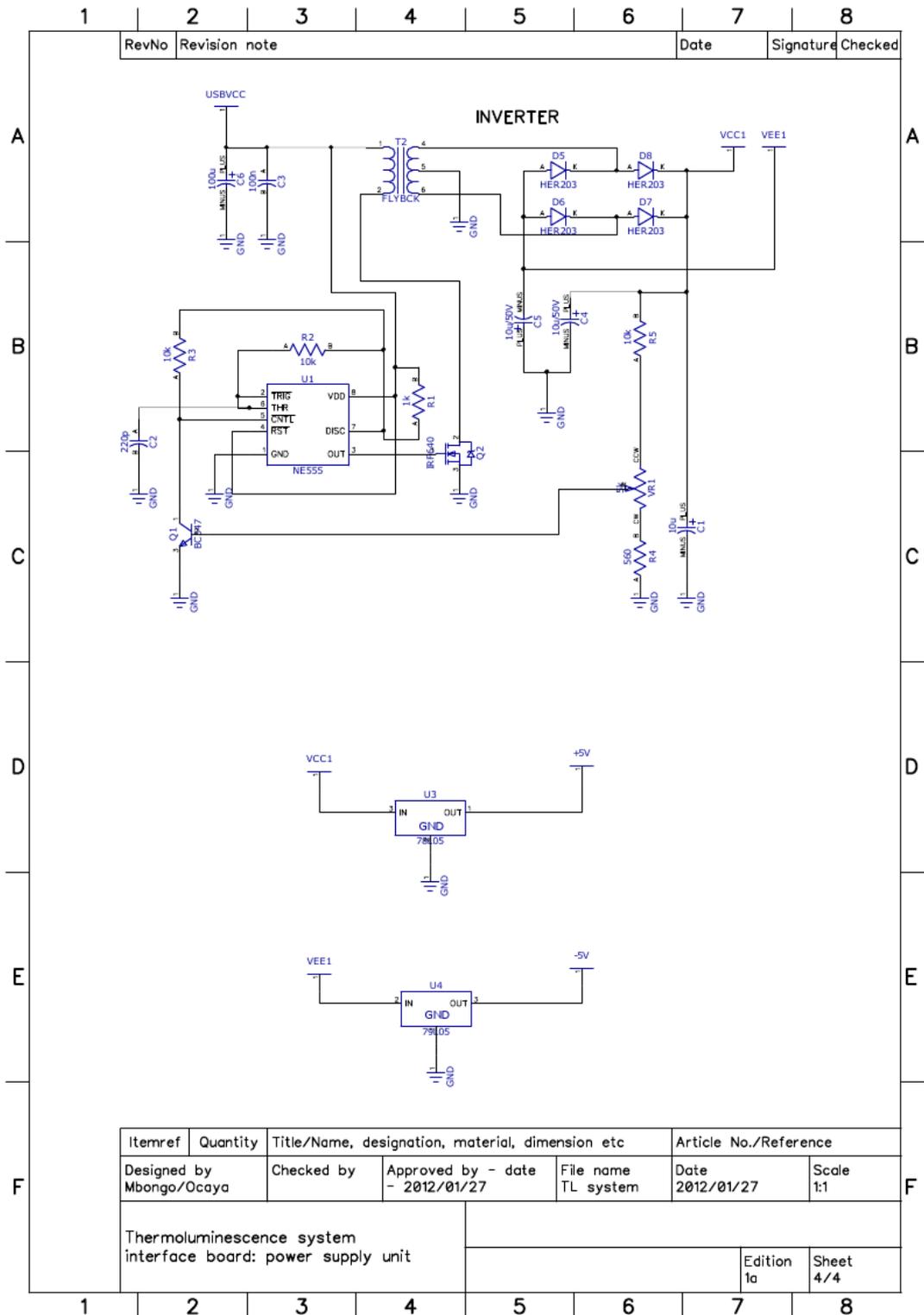
A.3.2 Microcontroller and USB interface board



A.3.3 Photometer and temperature sensors



A.3.4 LVPS power inverter and $\pm 5V$ regulators



Appendix B

The PIC firmware

```
// Note that all strings from the UART to the PC must be
// terminated with ;
// as my programming dictates in the VB.NET control
// program running on the PC
//
#include <built_in.h>
// char USB_DeviceID[]="TL hardware detected;";

// the sbit type must be defined as a global variable
sbit ADCon at ADCON0.B0; sbit ADCgo at ADCON0.B1;

unsigned int i, control, ADCread;
unsigned long ADCreadLong;

// will hold long version of ADCread char ADCvalStr[6],
*ADCvalLtrimStr; char uart_rd;
// only char type is unsigned by default

float paramA, paramB, paramC, set_point; float rkt, ekt, pkt, qkt,
ykt, ukt; float maxDuty, minDuty, pkt_1, ekt_1; float cLT;
// closed-loop settling time

minDuty = 0; maxDuty = 255;
pkt_1 = 0; ekt_1 = 0; paramA = 0.0923;
paramB = 0.923; paramC = 0.231; cLT = 10000;
// system settling time in ms

unsigned int myADC_Read(unsigned char channel) {
// char tmpCh;
// unsigned int ADCresult;
// ADC steps recommended in DS39637A page 251
// Step 1 - configure ADC module and select the channel

ADCON0 = channel << 2;
```

```

// get bit 0 into bit 2 position
// ADC module is still OFF after this
// ADCON1 = 0x0A;
// already set in main(), refs=VSS,VDD, 5 analog inputs
// step 2 - select acquisition time and ADC clock source
ADCON2 = 0xA1;
// right justified, 8TAD acquisition time, Fosc/8 ADC clock
ADCon = 1;
// turn the ADC on. DS39637A-page 255 says this and the start ADC

// must NOT be executed within the same instruction
delay_us(6);
// start the conversion
ADCgo = 1;
// CPU waits a time TACQ to start conversion
while (ADCgo);
// in DS39637A, next conversion must
// start after at least 2TAD
return ((ADRESH<<8) + ADRESL);
}

void main() {
    TRISA = 0xFF;           // PORTA is input
    TRISB = 0x7F;         // PORTB is input
    TRISC = 0;             // designate PORTC pins as output
    PORTC = 0;             // PORTC RC2 is PWM output
    ADCON1 = 0x0A;         // configure only RA4-RA0 pins as analog inputs
                           // other otherwise analog RB0,1,4 pins digital
    UART1_Init(2400);      // Initialize UART module at 2400 bps
    Delay_ms(100);         // Wait for UART module to stabilize
    ADC_Init();            // initialize ADC to use RC clock

    PWM1_Init(5000);       // Initialize PWM1 module at 5KHz
    DutyLo = 13;           // low duty cycle is 5%
    DutyHi = 230;          // high duty cycle is 90%

    PWM1_Start();         // start PWM1
    PWM1_Set_Duty(DutyLo);

/* Notes relating to the ADC converter

1: The RC source has a typical 4 us <= TAD <= 6 us time 2: For
device frequencies above 1 MHz, the device must be in
Sleep for the entire conversion or the A/D
accuracy may be out of specification. This
system uses 6.000 MHz USB controller derived

```

clock. Therefore, putting the PIC into sleep mode is necessary for accurate conversions. I have noticed that conversions done out of sleep are not accurate - producing significant error in expected value.

Reference: Microchip Technology Inc., Preliminary DS39637A-page 253, 2004

*/

```
while (1) { // Endless loop
  if (UART1_Data_Ready()) // if data is received
  {
    uart_rd = UART1_Read(); // then read the received data,
    if (uart_rd=='Q') // respond to a PC query
      UART1_Write_Text(USB_DeviceID); // and send data via

    if (uart_rd=='A')
    {
      ADCRead = myADC_Read(0);
      WordToStr(ADCRead, ADCvalStr);
      ADCvalLtrimStr = ltrim(ADCvalStr);
      UART1_Write_Text(ADCvalLtrimStr);
      UART1_Write_Text(";");
    }
    if (uart_rd=='B')
    {
      ADCRead = myADC_Read(1);
      WordToStr(ADCRead, ADCvalStr);
      ADCvalLtrimStr = ltrim(ADCvalStr);
      UART1_Write_Text(ADCvalLtrimStr);
      UART1_Write_Text(";");
    }
    if (uart_rd=='C')
    {
      ADCRead = myADC_Read(2);
      WordToStr(ADCRead, ADCvalStr);
      ADCvalLtrimStr = ltrim(ADCvalStr);
      UART1_Write_Text(ADCvalLtrimStr);
      UART1_Write_Text(";");
    }
    if (uart_rd=='D')
    {
      ADCRead = myADC_Read(3);
      WordToStr(ADCRead, ADCvalStr);
      ADCvalLtrimStr = ltrim(ADCvalStr);
      UART1_Write_Text(ADCvalLtrimStr);
    }
  }
}
```

```

    UART1_Write_Text(";");
}
if (uart_rd=='E')
{
    ADCRead = myADC_Read(4);
    WordToStr(ADCRead, ADCvalStr);
    ADCvalLtrimStr = ltrim(ADCvalStr);
    UART1_Write_Text(ADCvalLtrimStr);
    UART1_Write_Text(";");
}
} // if(character received) ends here

set_point = 120.0;           // temperature setpoint
PWM1_Init(5000);           // Initialize PWM1 module at 5KHz
PWM1_Start();

// Read thermocouple sensor output in degrees, connected to CH1
ykt = 0.48828125*myADC_Read(1); // multiplier=100*5/1024
rkt = set_point;           // in degrees C

ekt = rkt-ykt;             // Calculate error
pkt = paramB*ekt+pkt_1;    // Calculate I term
qkt = paramC*(ekt-ekt_1); // Calculate D term
ukt = pkt+paramA*ekt+qkt;  // Calculate PID output

if (ukt > maxDuty) {
    pkt = pkt_1;
    ukt = maxDuty;
}
else if (ukt < minDuty) {
    pkt = pkt_1;
    ukt = minDuty;
}
PWM1_Set_Duty(ukt); // send control to heater driver
pkt_1 = pkt;        // save variables
ekt_1 = ekt;
Delay_ms(cLT);     // closed-loop settling time
}
}

```



```

End While
offset3 = Val(tmpString(3)) / 1000
cnt = cnt + 1 ' skip the intermediate line feed

While offsetData(cnt) <> vbLf
    tmpString(4) = tmpString(4) + offsetData(cnt)
    cnt = cnt + 1
End While
offset4 = Val(tmpString(4)) / 1000
cnt = cnt + 1 ' skip the intermediate line feed

While offsetData(cnt) <> vbLf
    tmpString(5) = tmpString(5) + offsetData(cnt)
    cnt = cnt + 1
End While
offset5 = Val(tmpString(5)) / 1000
cnt = cnt + 1 ' skip the intermediate line feed

While offsetData(cnt) <> vbLf
    tmpString(6) = tmpString(6) + offsetData(cnt)
    cnt = cnt + 1
End While
offset6 = Val(tmpString(6)) / 1000
cnt = cnt + 1 ' skip the intermediate line feed

While offsetData(cnt) <> vbLf
    tmpString(7) = tmpString(7) + offsetData(cnt)
    cnt = cnt + 1
End While
offset7 = Val(tmpString(7)) / 1000
cnt = cnt + 1 ' skip the intermediate line feed

While offsetData(cnt) <> vbLf
    tmpString(8) = tmpString(8) + offsetData(cnt)
    cnt = cnt + 1
End While
offset8 = Val(tmpString(8)) / 1000
cnt = cnt + 1 ' skip the intermediate line feed

While offsetData(cnt) <> vbLf
    tmpString(9) = tmpString(9) + offsetData(cnt)
    cnt = cnt + 1
End While

offset9 = Val(tmpString(9)) / 1000

MyFile.Close()
MyStream.Close()
Exit Sub ' exit if a previous settings file already exists, else create a
default one
Else
    offset0 = 0
    offset1 = 0
    offset2 = 0
    offset3 = 0
    offset4 = 0
    offset5 = 0
    offset6 = 0
    offset7 = 0
    offset8 = 0
    offset9 = 0
End If
End Sub

Private Sub OpenToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles OpenFileSolar.Click
    System.Diagnostics.Process.Start("notepad.exe", DataFileName)
End Sub

Private Sub SaveToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    SaveFileDialog.ShowDialog()
    statusSolar.Text = SaveFileDialog.FileName

```

```

End Sub

Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    dlgSetCOMport.comboCOMlist.Items.Clear() ' start with a clear COM list to allow
updating of ports
    Dim ports As String() = serialPort.GetPortNames()
    Dim port As String
    For Each port In ports
        dlgSetCOMport.comboCOMlist.Items.Add(port) 'populate the COM port list in the
combo list
    Next
    dlgSetCOMport.ShowDialog()
    DataFileName = DataFileName + "\solar.txt" ' default filename
    SaveFileDialog.Filter = "Comma delimited Microsoft Excel file (*.txt)|*.txt"
    fileString = ""
    toolFileName.Text = DataFileName + " (Default)"
    ContentsToolStripMenuItem.Enabled = True
    TimeSample = 1000 ' store in a global variable
    timerSolar.Interval = TimeSample
    rtBox.Text = "The ADC is stopped or USB port is not set."
    serialPort.ReadTimeout = 1000 ' if no response in 1000 ms, report timeout and
then exit
    serialPort.WriteTimeout = 500
    createOffsetFile() ' check for offset file, create a default one if none
End Sub

Function IsMidnight() As Integer
' Test the current time for midnight
' If it is midnight, return 1 else return 0
    Dim midnightVal As Integer
    Dim hrVal, minVal, secVal As Integer
    hrVal = DateAndTime.Now.Hour
    minVal = DateAndTime.Now.Minute
    secVal = DateAndTime.Now.Second

    midnightVal = 0
    If hrVal = 0 Then
        If minVal = 0 Then
            If secVal <= 5 Then ' allow up to five seconds for reset
                midnightVal = 1
            End If
        End If
    End If

    Return midnightVal
End Function

Private Sub ToolStripMenuItem3_Click(ByVal sender As System.Object, _
ByVal e As System.EventArgs) Handles _
    menuRunADC.Click, timerSolar.Tick, btnADCcontrol.Click
    Dim COMstring As String
    Dim ADCvalue As Integer
    Dim fileString As String
    Dim midnightValreset As Integer

    If PortSet = True Then
        timerSolar.Interval = 1000 '1000 ms is the default
        timerSolar.Enabled = True
        timerSolar.Start()
        Me.rtBox.Text = "Running ADC..."
        menuRunADC.Enabled = False
        menuStopADC.Enabled = True
        btnADCcontrol.Enabled = False
        btnADCstop.Enabled = True

        Dim sb As New StringBuilder() ' use the VB.net version 9 StringBuilder class
        ' see also http://msdn.microsoft.com/en-us/library/6kalwd3w(VS.90).aspx
        ' and http://www.dotnetperls.com/stringbuilder-vbnet

        fileString = Format(DateAndTime.Now) + " "
    End If

```

```

' In version 4 the millisecond part of the filestring has been removed
' see also http://www.techrepublic.com/article/convert-datetime-values-into-
strings-in-vbnet/6106789
midnightValreset = IsMidnight() ' check to see if it is 1 (=midnight)

sb.Append(fileString)

' Get CH0
Me.serialPort.Write("B")
Try
    COMstring = Me.serialPort.ReadTo(";")
    ADCvalue = Val(COMstring) ' convert it to integer type

    lblCh0.Text = Str(ADCvalue) ' write it to CH0 display label
    ' above is important since the format of incoming
    ' COMstring from SIV3 module has leading spaces

    lblCh0volt.Text = Format((ADCvalue * 5 / 1023) - offset0, "0.000")
    lblCh0mV.Text = Format((ADCvalue * 5 / 1023) - offset0) * 9.95, "0.0")
    lblCh0integral.Text = Format(Val(lblCh0integral.Text) + Val(lblCh0mV.Text)
, "#.0")

    If midnightValreset = 1 Then
        lblCh0integral.Text = Format(0, "#.0") ' reset the integrator a
midnight
    End If

    If chkBox.Checked = True Then ' build up the string to write to file
        sb.Append(lblCh0integral.Text + " ")
    Else
        sb.Append(lblCh0mV.Text + " ")
    End If

Catch ex As Exception
    timerSolar.Stop()
    Dim msgResult As DialogResult
    msgResult = MsgBox(ex.Message + " Continue?", MsgBoxStyle.YesNo, "USB
communications error:")
    If msgResult = Windows.Forms.DialogResult.No Then
        Application.Exit()
    End If
End Try

' Get CH1 -----
Me.serialPort.Write("C")
Try
    COMstring = Me.serialPort.ReadTo(";")
    ADCvalue = Val(COMstring) ' convert it to integer type

    lblCh1.Text = Str(ADCvalue) ' write it to CH0 display label
    ' above is important since the format of incoming
    ' COMstring from SIV3 module has leading spaces
    lblCh1volt.Text = Format((ADCvalue * 5 / 1023) - offset1, "0.000")
    lblCh1mV.Text = Format((ADCvalue * 5 / 1023) - offset0) * 9.804, "0.0")
    lblCh1integral.Text = Format(Val(lblCh1integral.Text) + Val(lblCh1mV.Text)
, "#.0")

    If midnightValreset = 1 Then
        lblCh1integral.Text = Format(0, "#.0") ' reset the integrator a
midnight
    End If

    If chkBox.Checked = True Then ' build up the string to write to file
        sb.Append(lblCh1integral.Text + " ")
    Else
        sb.Append(lblCh1mV.Text + " ")
    End If

Catch ex As Exception
    timerSolar.Stop()
    Dim msgResult As DialogResult
    msgResult = MsgBox(ex.Message + " Continue?", MsgBoxStyle.YesNo, "USB

```

```

communications error:")
    If msgResult = Windows.Forms.DialogResult.No Then
        Application.Exit()
    End If
End Try

' end CH get-----

' Get CH2 -----
Me.serialPort.Write("D")
Try
    COMstring = Me.serialPort.ReadTo(";")
    ADCvalue = Val(COMstring) ' convert it to integer type

    lblCh2.Text = Str(ADCvalue) ' write it to CH0 display label
    ' above is important since the format of incoming
    ' COMstring from SIV3 module has leading spaces
    lblCh2volt.Text = Format((ADCvalue * 5 / 1023) - offset2, "0.000")
    lblCh2mV.Text = Format((ADCvalue * 5 / 1023) - offset0) * 9.872, "0.0")
    lblCh2integral.Text = Format(Val(lblCh2integral.Text) + Val(lblCh2mV.Text)
, "#.0")

    If midnightValreset = 1 Then
        lblCh2integral.Text = Format(0, "#.0") ' reset the integrator a
midnight
    End If

    If chkBox.Checked = True Then ' build up the string to write to file
        sb.Append(lblCh2integral.Text + " ")
    Else
        sb.Append(lblCh2mV.Text + " ")
    End If

Catch ex As Exception
    timerSolar.Stop()
    Dim msgResult As DialogResult
    msgResult = MsgBox(ex.Message + " Continue?", MsgBoxStyle.YesNo, "USB
communications error:")
    If msgResult = Windows.Forms.DialogResult.No Then
        Application.Exit()
    End If
End Try

' end CH get-----

' Get CH3 -----
Me.serialPort.Write("E")
Try
    COMstring = Me.serialPort.ReadTo(";")
    ADCvalue = Val(COMstring) ' convert it to integer type

    lblCh3.Text = Str(ADCvalue) ' write it to CH0 display label
    ' above is important since the format of incoming
    ' COMstring from SIV3 module has leading spaces
    lblCh3volt.Text = Format((ADCvalue * 5 / 1023) - offset3, "0.000")
    lblCh3mV.Text = Format((ADCvalue * 5 / 1023) - offset0) * 9.872, "0.0")
    lblCh3integral.Text = Format(Val(lblCh3integral.Text) + Val(lblCh3mV.Text)
, "#.0")

    If midnightValreset = 1 Then
        lblCh3integral.Text = Format(0, "#.0") ' reset the integrator a
midnight
    End If

    If chkBox.Checked = True Then ' build up the string to write to file
        sb.Append(lblCh3integral.Text + " ")
    Else
        sb.Append(lblCh3mV.Text + " ")
    End If

Catch ex As Exception
    timerSolar.Stop()
    Dim msgResult As DialogResult
    msgResult = MsgBox(ex.Message + " Continue?", MsgBoxStyle.YesNo, "USB

```

```

communications error:")
    If msgResult = Windows.Forms.DialogResult.No Then
        Application.Exit()
    End If
End Try

' end CH get-----

' Get CH4 -----
Me.serialPort.Write("F")
Try
    COMstring = Me.serialPort.ReadTo(";")
    ADCvalue = Val(COMstring) ' convert it to integer type

    lblCh4.Text = Str(ADCvalue) ' write it to CH0 display label
    ' above is important since the format of incoming
    ' COMstring from SIV3 module has leading spaces
    lblCh4volt.Text = Format((ADCvalue * 5 / 1023) - offset4, "0.000")
    lblCh4mV.Text = Format((ADCvalue * 5 / 1023) - offset0) * 10.0, "0.0")
    lblCh4integral.Text = Format(Val(lblCh4integral.Text) + Val(lblCh4mV.Text)
, "#.0")

    If midnightValreset = 1 Then
        lblCh4integral.Text = Format(0, "#.0") ' reset the integrator a
midnight
    End If

    If chkBox.Checked = True Then ' build up the string to write to file
        sb.Append(lblCh4integral.Text + " ")
    Else
        sb.Append(lblCh4mV.Text + " ")
    End If

Catch ex As Exception
    timerSolar.Stop()
    Dim msgResult As DialogResult
    msgResult = MsgBox(ex.Message + " Continue?", MsgBoxStyle.YesNo, "USB
communications error:")
    If msgResult = Windows.Forms.DialogResult.No Then
        Application.Exit()
    End If
End Try

' end CH get-----

' Get CH5 -----
Me.serialPort.Write("G")
Try
    COMstring = Me.serialPort.ReadTo(";")
    ADCvalue = Val(COMstring) ' convert it to integer type

    lblCh5.Text = Str(ADCvalue) ' write it to CH0 display label
    ' above is important since the format of incoming
    ' COMstring from SIV3 module has leading spaces
    lblCh5volt.Text = Format((ADCvalue * 5 / 1023) - offset5, "0.000")
    lblCh5mV.Text = Format((ADCvalue * 5 / 1023) - offset0) * 10.07, "0.0")
    lblCh5integral.Text = Format(Val(lblCh5integral.Text) + Val(lblCh5mV.Text)
, "#.0")

    If midnightValreset = 1 Then
        lblCh5integral.Text = Format(0, "#.0") ' reset the integrator a
midnight
    End If

    If chkBox.Checked = True Then ' build up the string to write to file
        sb.Append(lblCh5integral.Text + " ")
    Else
        sb.Append(lblCh5mV.Text + " ")
    End If

Catch ex As Exception
    timerSolar.Stop()
    Dim msgResult As DialogResult
    msgResult = MsgBox(ex.Message + " Continue?", MsgBoxStyle.YesNo, "USB

```

```

communications error:")
    If msgResult = Windows.Forms.DialogResult.No Then
        Application.Exit()
    End If
End Try

' end CH get-----

' Get CH6 -----
Me.serialPort.Write("H")
Try
    COMstring = Me.serialPort.ReadTo(";")
    ADCvalue = Val(COMstring) ' convert it to integer type

    lblCh6.Text = Str(ADCvalue) ' write it to CH0 display label
    ' above is important since the format of incoming
    ' COMstring from SIV3 module has leading spaces
    lblCh6volt.Text = Format((ADCvalue * 5 / 1023) - offset6, "0.000")
    lblCh6mV.Text = Format((ADCvalue * 5 / 1023) - offset0) * 10.02, "0.0")
    lblCh6integral.Text = Format(Val(lblCh6integral.Text) + Val(lblCh6mV.Text)
, "#.0")

    If midnightValreset = 1 Then
        lblCh6integral.Text = Format(0, "#.0") ' reset the integrator a
midnight
    End If

    If chkBox.Checked = True Then ' build up the string to write to file
        sb.Append(lblCh6integral.Text + " ")
    Else
        sb.Append(lblCh6mV.Text + " ")
    End If

Catch ex As Exception
    timerSolar.Stop()
    Dim msgResult As DialogResult
    msgResult = MsgBox(ex.Message + " Continue?", MsgBoxStyle.YesNo, "USB
communications error:")
    If msgResult = Windows.Forms.DialogResult.No Then
        Application.Exit()
    End If
End Try

' end CH get-----

' Get CH7 -----
Me.serialPort.Write("I")
Try
    COMstring = Me.serialPort.ReadTo(";")
    ADCvalue = Val(COMstring) ' convert it to integer type

    lblCh7.Text = Str(ADCvalue) ' write it to CH0 display label
    ' above is important since the format of incoming
    ' COMstring from SIV3 module has leading spaces
    lblCh7volt.Text = Format((ADCvalue * 5 / 1023) - offset7, "0.000")
    lblCh7mV.Text = Format((ADCvalue * 5 / 1023) - offset0) * 10.0, "0.0")
    lblCh7integral.Text = Format(Val(lblCh7integral.Text) + Val(lblCh7mV.Text)
, "#.0")

    If midnightValreset = 1 Then
        lblCh7integral.Text = Format(0, "#.0") ' reset the integrator a
midnight
    End If

    If chkBox.Checked = True Then ' build up the string to write to file
        sb.Append(lblCh7integral.Text + " ")
    Else
        sb.Append(lblCh7mV.Text + " ")
    End If

Catch ex As Exception
    timerSolar.Stop()
    Dim msgResult As DialogResult
    msgResult = MsgBox(ex.Message + " Continue?", MsgBoxStyle.YesNo, "USB

```

```

communications error:")
    If msgResult = Windows.Forms.DialogResult.No Then
        Application.Exit()
    End If
End Try

' end CH get-----

' Get CH8 -----
Me.serialPort.Write("J")
Try
    COMstring = Me.serialPort.ReadTo(";")
    ADCvalue = Val(COMstring) ' convert it to integer type

    lblCh8.Text = Str(ADCvalue) ' write it to CH0 display label
    ' above is important since the format of incoming
    ' COMstring from SIV3 module has leading spaces
    lblCh8volt.Text = Format((ADCvalue * 5 / 1023) - offset8, "0.000")
    lblCh8mV.Text = Format((ADCvalue * 5 / 1023) - offset0) * 9.901, "0.0")
    lblCh8integral.Text = Format(Val(lblCh8integral.Text) + Val(lblCh8mV.Text)
, "#.0")

    If midnightValreset = 1 Then
        lblCh8integral.Text = Format(0, "#.0") ' reset the integrator a
midnight
    End If

    If chkBox.Checked = True Then ' build up the string to write to file
        sb.Append(lblCh8integral.Text + " ")
    Else
        sb.Append(lblCh8mV.Text + " ")
    End If

Catch ex As Exception
    timerSolar.Stop()
    Dim msgResult As DialogResult
    msgResult = MsgBox(ex.Message + " Continue?", MsgBoxStyle.YesNo, "USB
communications error:")
    If msgResult = Windows.Forms.DialogResult.No Then
        Application.Exit()
    End If
End Try

' end CH get-----

' Get CH9 -----
Me.serialPort.Write("K")
Try
    COMstring = Me.serialPort.ReadTo(";")
    ADCvalue = Val(COMstring) ' convert it to integer type

    lblCh9.Text = Str(ADCvalue) ' write it to CH0 display label
    ' above is important since the format of incoming
    ' COMstring from SIV3 module has leading spaces
    lblCh9volt.Text = Format((ADCvalue * 5 / 1023) - offset9, "0.000")
    lblCh9mV.Text = Format((ADCvalue * 5 / 1023) - offset0) * 10.07, "0.0")
    lblCh9integral.Text = Format(Val(lblCh9integral.Text) + Val(lblCh9mV.Text)
, "#.0")

    If midnightValreset = 1 Then
        lblCh9integral.Text = Format(0, "#.0") ' reset the integrator a
midnight
    End If

    If chkBox.Checked = True Then ' build up the string to write to file
        sb.Append(lblCh9integral.Text + " ")
    Else
        sb.Append(lblCh9mV.Text + " ")
    End If

    fileString = sb.ToString
    Me.rtBox.Text = ("Written to file: " + fileString)

```

```

C:\Users\Ocaya\Documents\Visual Studio 2008\Projects\Solar\Solar v1 revision\Main.vb 9
    Using outfile As New StreamWriter(DataFileName, True) ' flag=True for
append
    outfile.WriteLine(fileString)
End Using ' this gets rid of the resource outfile, i.e. works
like close()

    Catch ex As Exception
        timerSolar.Stop()
        Dim msgResult As DialogResult
        msgResult = MsgBox(ex.Message + " Continue?", MsgBoxStyle.YesNo, "USB
communications error:")
        If msgResult = Windows.Forms.DialogResult.No Then
            Application.Exit()
        End If
    End Try

    ' end CH get-----

    Else : MsgBox("Please specify the USB port first", MsgBoxStyle.Information, "Error
specifying USB port")
        rtBox.Text = "The ADC cannot be run until a USB port is specified"
    End If
End Sub

Private Sub menuStopADC_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles menuStopADC.Click, btnADCstop.Click
    Dim enableADC As MsgBoxResult
    enableADC = MsgBox("The Solar Integrator is running. Are you sure you want to stop
it?", MsgBoxStyle.YesNo, "S13 program interruption:")
    If enableADC = MsgBoxResult.Yes Then
        timerSolar.Stop()
        menuStopADC.Enabled = False
        menuRunADC.Enabled = True
        btnADCcontrol.Text = "Start ADC"
        rtBox.Text = "Solar Integrator is stopped."
        btnADCstop.Enabled = False
        btnADCcontrol.Enabled = True
    End If
End Sub

Private Sub StartADCToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles menuRunADC.Click

End Sub

Private Sub ExitToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    Application.Exit()
End Sub

Private Sub AssignPORTToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles AssignPORTToolStripMenuItem.Click
    dlgSetCOMport.ShowDialog()
    If lblPortName.Text = "Not set" Then
        rtBox.Text = "The ADC cannot be run until a USB port is specified"
        lblPortName.ForeColor = Color.Red
    Else
        rtBox.Text = "USB port has been set. The ADC may now be started."
        lblPortName.ForeColor = Color.Green
    End If
End Sub

Private Sub QueryADCToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles QueryADCToolStripMenuItem.Click
    Dim COMstring As String
    Dim cnt As Integer = 0
    If PortSet = True Then

        Try
            Me.serialPort.Write("Q")
            Me.rtBox.Text = "Looking for S13 module..."

            COMstring = Me.serialPort.ReadTo(";")

```

```

        Me.rtBox.Text = Me.rtBox.Text + COMstring

        Catch ex As Exception
            MsgBox(ex.Message + " USB hardware not found or not responding")
            Application.Exit()
        End Try
    Else : MsgBox("Please specify the USB port first", MsgBoxStyle.Information, "Port ✘
specification error:")
    End If
End Sub

Private Sub AboutToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As ✘
System.EventArgs) Handles AboutToolStripMenuItem.Click
    AboutSIV.ShowDialog()
End Sub

Private Sub DataFilenameToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e ✘
As System.EventArgs)

End Sub

Private Sub ToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As System. ✘
EventArgs) Handles newDatafile.Click
    SaveFileDialog.ShowDialog()
    DataFileName = SaveFileDialog.FileName
    toolFilename.Text = DataFileName
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) ✘
Handles Button1.Click, ExitToolStripMenuItem.Click
    Dim ExitResult As DialogResult
    ExitResult = MsgBox("Are you sure you want to exit?", MsgBoxStyle.YesNo, "Program ✘
termination requested:")
    If ExitResult = Windows.Forms.DialogResult.Yes Then Application.Exit()
End Sub

Private Sub SampleIntervalToolStripMenuItem_Click(ByVal sender As System.Object, ByVal ✘
e As System.EventArgs)

End Sub

Private Sub ToolStripMenuItem1_Click_1(ByVal sender As System.Object, ByVal e As ✘
System.EventArgs)
    Dim result As DialogResult = PrintDialog.ShowDialog()
    ' If the result is OK then print the document.
    If (result = DialogResult.OK) Then
        PrintDocument.DocumentName = DataFileName
        PrintDocument.Print()
    End If
End Sub

Private Sub OffsetAdjustToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e ✘
As System.EventArgs) Handles OffsetAdjustToolStripMenuItem.Click
    offsetADJ.Show()
End Sub

Private Sub TestFileReadToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e ✘
As System.EventArgs) Handles TestFileReadToolStripMenuItem.Click
    Dim offsetData, tmpString(10) As String
    Dim oFileName = "solar.ini"

    oFileName = Application.StartupPath() + oFileName

    If File.Exists(oFileName) = False Then
        MsgBox("Channel offsets file does not yet exist.", vbOKCancel, "Attempting to ✘
read offsets from file")
        Exit Sub
    Else

        Dim MyFile As FileStream = New FileStream(oFileName, FileMode.OpenOrCreate, ✘
FileAccess.Read)
        Dim MyStream As StreamReader = New StreamReader(MyFile)
        offsetData = MyStream.ReadToEnd()

```

```
Dim cnt As Integer = 0
tmpString(0) = "" ' start with empty strings
tmpString(1) = ""
tmpString(2) = ""
tmpString(3) = ""
tmpString(4) = ""
tmpString(5) = ""
tmpString(6) = ""
tmpString(7) = ""
tmpString(8) = ""
tmpString(9) = ""

While offsetData(cnt) <> vbLf
    tmpString(0) = tmpString(0) + offsetData(cnt)
    cnt = cnt + 1
End While

cnt = cnt + 1 ' skip the intermediate line feed

While offsetData(cnt) <> vbLf
    tmpString(1) = tmpString(1) + offsetData(cnt)
    cnt = cnt + 1
End While

cnt = cnt + 1 ' skip the intermediate line feed

While offsetData(cnt) <> vbLf
    tmpString(2) = tmpString(2) + offsetData(cnt)
    cnt = cnt + 1
End While

cnt = cnt + 1 ' skip the intermediate line feed

While offsetData(cnt) <> vbLf
    tmpString(3) = tmpString(3) + offsetData(cnt)
    cnt = cnt + 1
End While

cnt = cnt + 1 ' skip the intermediate line feed

While offsetData(cnt) <> vbLf
    tmpString(4) = tmpString(4) + offsetData(cnt)
    cnt = cnt + 1
End While

cnt = cnt + 1 ' skip the intermediate line feed

While offsetData(cnt) <> vbLf
    tmpString(5) = tmpString(5) + offsetData(cnt)
    cnt = cnt + 1
End While

cnt = cnt + 1 ' skip the intermediate line feed

While offsetData(cnt) <> vbLf
    tmpString(6) = tmpString(6) + offsetData(cnt)
    cnt = cnt + 1
End While

cnt = cnt + 1 ' skip the intermediate line feed

While offsetData(cnt) <> vbLf
    tmpString(7) = tmpString(7) + offsetData(cnt)
    cnt = cnt + 1
End While

cnt = cnt + 1 ' skip the intermediate line feed

While offsetData(cnt) <> vbLf
    tmpString(8) = tmpString(8) + offsetData(cnt)
    cnt = cnt + 1
End While
```

```
    cnt = cnt + 1 ' skip the intermediate line feed

    While offsetData(cnt) <> vbLf
        tmpString(9) = tmpString(9) + offsetData(cnt)
        cnt = cnt + 1
    End While

    MsgBox("Ch0: " + tmpString(0) + "Ch1: " + tmpString(1) + "Ch2: " + tmpString
(2) + "Ch3: " + tmpString(3) +
        "Ch4: " + tmpString(4) + "Ch5: " + tmpString(5) + "Ch6: " +
tmpString(6) + "Ch7: " + tmpString(7) +
        "Ch8: " + tmpString(8) + "Ch9: " + tmpString(9), MsgBoxStyle.OkOnly
, "Channel offsets in millivolts")

    MyFile.Close()
    MyStream.Close()
End If
End Sub

Private Sub ContentsToolStripMenuItem_Click(ByVal sender As System.Object, _
ByVal e As System.EventArgs) _
Handles ContentsToolStripMenuItem.Click
    System.Windows.Forms.Help.ShowHelp(Me, "solar.chm", HelpNavigator.TableOfContents)
End Sub

End Class
```

C:\Users\Ocaya\Documents\Visual Studio 2008...Solar\Solar v1 revision\setPort.Designer.vb.1

```
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Partial Class dlgSetCOMport
    Inherits System.Windows.Forms.Form
    Public currentCOMport, oldCOMport As String

    'Form overrides dispose to clean up the component list.
    <System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
        Finally
            MyBase.Dispose(disposing)
        End Try
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    <System.Diagnostics.DebuggerStepThrough()> _
    Private Sub InitializeComponent()
        Me.comboCOMlist = New System.Windows.Forms.ComboBox
        Me.btnAcceptPort = New System.Windows.Forms.Button
        Me.Button2 = New System.Windows.Forms.Button
        Me.SuspendLayout()
        '
        'comboCOMlist
        '
        Me.comboCOMlist.FormattingEnabled = True
        Me.comboCOMlist.Location = New System.Drawing.Point(12, 32)
        Me.comboCOMlist.Name = "comboCOMlist"
        Me.comboCOMlist.Size = New System.Drawing.Size(121, 21)
        Me.comboCOMlist.TabIndex = 0
        Me.comboCOMlist.Text = "Not set"
        '
        'btnAcceptPort
        '
        Me.btnAcceptPort.Location = New System.Drawing.Point(34, 207)
        Me.btnAcceptPort.Name = "btnAcceptPort"
        Me.btnAcceptPort.Size = New System.Drawing.Size(83, 26)
        Me.btnAcceptPort.TabIndex = 1
        Me.btnAcceptPort.Text = "Accept"
        Me.btnAcceptPort.UseVisualStyleBackColor = True
        '
        'Button2
        '
        Me.Button2.Location = New System.Drawing.Point(128, 209)
        Me.Button2.Name = "Button2"
        Me.Button2.Size = New System.Drawing.Size(74, 23)
        Me.Button2.TabIndex = 2
        Me.Button2.Text = "Cancel"
        Me.Button2.UseVisualStyleBackColor = True
        '
        'dlgSetCOMport
        '
        Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)
        Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
        Me.ClientSize = New System.Drawing.Size(284, 262)
        Me.Controls.Add(Me.Button2)
        Me.Controls.Add(Me.btnAcceptPort)
        Me.Controls.Add(Me.comboCOMlist)
        Me.Name = "dlgSetCOMport"
        Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterParent
        Me.Text = "Assign COM port to SIV3"
        Me.ResumeLayout(False)
    End Sub

    Friend WithEvents comboCOMlist As System.Windows.Forms.ComboBox
    Friend WithEvents btnAcceptPort As System.Windows.Forms.Button
    Friend WithEvents Button2 As System.Windows.Forms.Button
```

C:\Users\Ocaya\Documents\Visual Studio 2008...\Solar\Solar v1 revision\setPort.Designer.vb.2

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnAcceptPort.Click
    oldCOMport = currentCOMport 'save the old port for closing if different from
current
    currentCOMport = Me.comboCOMlist.Text() ' get the newly specified port

    If currentCOMport = "Not set" Then
        MsgBox("A COM port must be specified to use the converter", MsgBoxStyle.
Exclamation, "Port specification error:")
        main.PortSet = False
        main.lblPortName.ForeColor = Color.Red
        main.menuStopADC.Enabled = False
        Me.Close()
    ElseIf main.serialPort.IsOpen And currentCOMport = oldCOMport Then
        MsgBox("The COM port is already open", MsgBoxStyle.Exclamation, "Port
specification error:")
        main.PortSet = True
        main.lblPortName.ForeColor = Color.Green
    Else
        main.serialPort.Close() 'close the old port
        main.serialPort.PortName = currentCOMport
        main.QueryADCToolStripMenuItem.Enabled = True
        main.lblPortName.Text = currentCOMport
        main.lblPortName.ForeColor = Color.Green
        main.serialPort.BaudRate = 9600
        main.serialPort.DataBits = 8
        main.serialPort.Handshake = IO.Ports.Handshake.None ' don't use flow control
        main.serialPort.Parity = IO.Ports.Parity.None
        main.serialPort.StopBits = IO.Ports.StopBits.One
        Try
            main.serialPort.Open()
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
        main.PortSet = True
    End If

    Me.Close()
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
    Me.DialogResult = System.Windows.Forms.DialogResult.Cancel
    main.menuStopADC.Enabled = False
    main.QueryADCToolStripMenuItem.Enabled = False
    Me.Close()
End Sub
End Class
```