www.sensorsportal.com

Sensors & Transducers

ISSN 1726-5479
© 2008 by IFSA
http://www.sensorsportal.com

Multi Channels PWM Controller for Thermoelectric Cooler Using a Programmable Logic Device and Lab-Windows CVI

Eli FLAXER
AFEKA - Tel-Aviv Academic College of Engineering
218 Bney Efrayim Rd. 69107 Tel-Aviv, Israel
Tel.: +972-3-6449344
E-mail: flaxer@afeka.ac.il

Received: 16 July 2008 /Accepted: 15 September 2008 /Published: 30 September 2008

Abstract: We present a complete design of a multi channels PID controller for Thermoelectric Cooler (TEC) using a pulse width modulation (PWM) technique implemented by a dedicated programmable logic device (PLD) programmed by VHDL. The PID control loop is implemented by software written by National Instrument Lab-Windows CVI. Due to the fact that the implementation is by a VHDL and PLD the design is modular, as a result, the circuit is very compact in size and very low cost as compared to any commercial product. In addition, since the control loop is implemented by software running on a personal computer (PC) using a C language, it is easy to adjust the controller to various environmental conditions and for a width range of sensors like: a thermo couple (TC), thermistor, resistance temperature detectors (RTD) etc. We demonstrate the performance of this circuit as a controller for a small incubator using thermistor as the temperature sensor. *Copyright © 2008 IFSA.*

Keywords: Thermoelectric cooler, PID controller, Programmable logic

1. Introduction

Temperature control is needed for a wide range of applications. For some, a precision of ±1 °C is satisfactory, but accurate applications require precision of ±0.1°C and in some cases even better. For example, the performances of electro-optics devices depend strongly on temperature; hence, these devices require high precision temperature control [1]. Analogously, in bio-tech systems a constant regulation with a precision of a 0.1°C is satisfactory. Moreover, in several systems it is not sufficient to regulate the temperature to a constant value, but instead, an optimization procedure is required. In systems like these, feedback control mechanisms vary the target temperature to maximize the output signal from the device under control.

Thermoelectric Coolers (TEC) [2, 3, 4] are solid state heat pumps that operate according to the Peltier effect [5] - a heating or cooling effect when an electric current passes through two conductors. A voltage applied to the free ends of two dissimilar materials creates a temperature difference. With this temperature difference, Peltier cooling will cause heat to move from one end to the other. The heat pumping capacity of a cooler is proportional to the current and the number of pairs of n- and p- type elements.

In this work we present a complete design of a multi channels PID controller for TEC, implemented by a dedicated programmable logic device (PLD) that is programmed by very high-speed integrated circuit hardware description language (VHDL). The PID control loop is implemented by software written by National Instrument [6] Lab-Windows CVI. We introduce, in detail, the design of a 4 channels controller and then explain how to expand it to 32 channels. The design is modular due to the fact that the implementation is by a VHDL and PLD. As a result, the circuit is very compact in size and at very low cost.

2. Design Considerations

The controller, presented here, was developed to operate with a variety TECs, at a wide temperature range, with variety sensors like: thermo couple (TC), thermistor, and resistance temperature detectors (RTD). The controller is connected to a personal computer (PC) via the standard parallel port - to set the outputs power and monitoring the actual temperature, while the PID control loop is implemented by software on the PC. To achieve a compact design with high efficiency, we used a high frequency PWM technique with a small coil filter. High frequency PWM technique combines the advantages of high efficiency energy transfer and a very low ripple at the output. As mentioned above, the TEC heat pump capacity is proportional to the current flow through the device, this means that by inverting the current direction, one converts the cooler to a heater and vice versa. To improve the controller performance, we drive the TEC with a bi-directional current, switched by full-bridge solid-state devices.

3. Electronic Circuit

Fig. 1 is the circuit diagram of the 4 channels controller. The major device in the circuit is U_1 - the digital control unit, which is comprised of one Complex Programmable Logic Device (CPLD) timed by a 25 MHz clock – U_3. The CPLD was programmed on board through JTAG (J_2), and implements the following:

1. Communicating with a PC in Enhance Parallel Port (EPP) mode thru J_1.
2. Controlling the data acquisition unit - U_2.
3. Implementing the multi channels 256 levels PWM controller.

The PWM outputs of the CPLD are connected to the full bridge power devices – U_4 and U_5. The bridge - L298 - converts the TTL logic level to power pulses of up to 36 V and up to 2.5 A. The outputs of the bridge are filtered by low pass filters (L_3-L_{10} and C_9-C_{16}) in order to achieve a DC level on the TECs that are connected to JP_5 - JP_8.

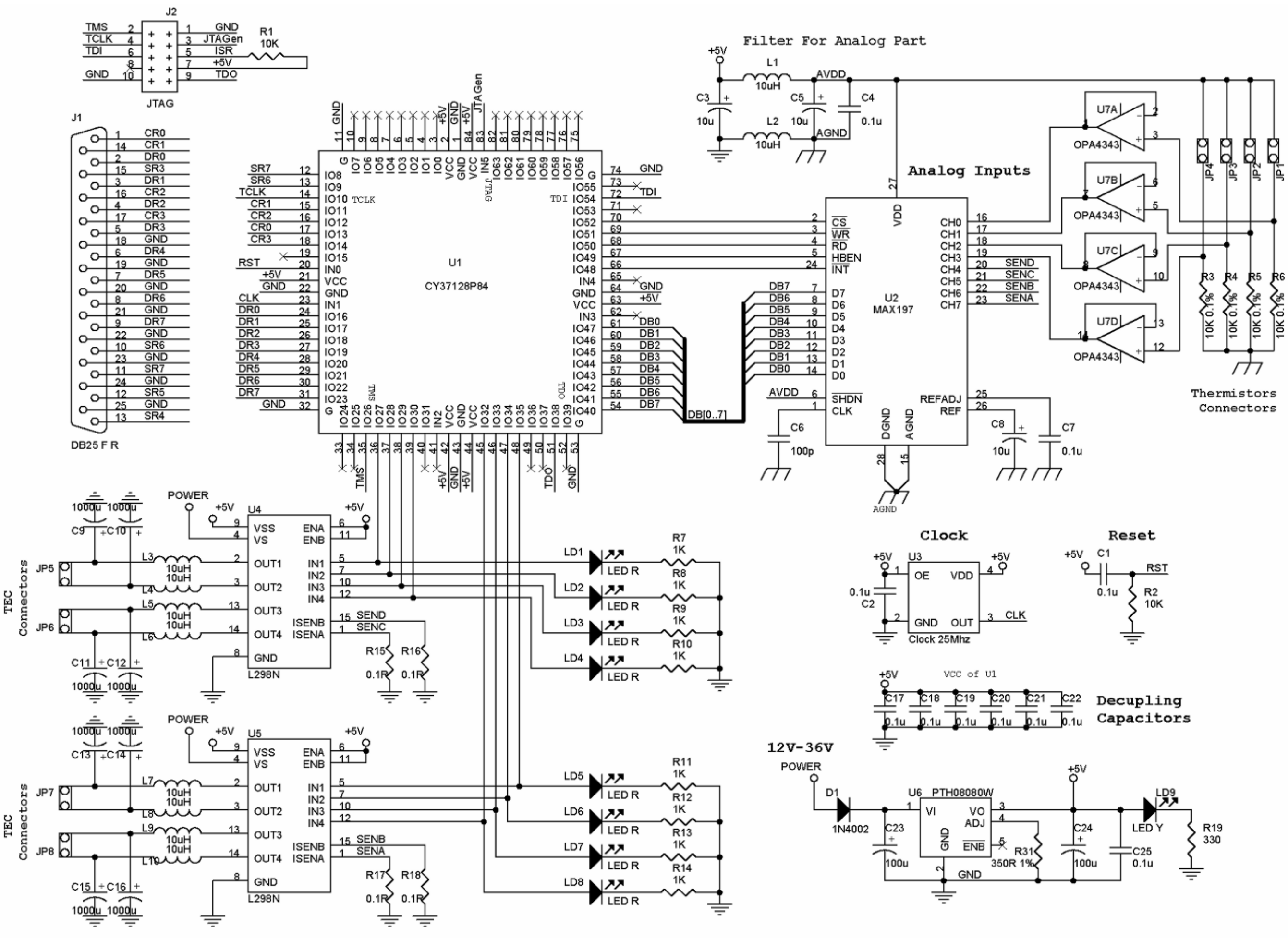


Fig. 1. The controller circuit diagram.

The temperatures of the regulated objects, attached to the TECs are sampled by Negative Temperature Coefficient (NTC) thermistors [7] that are connected to JP_1-JP_4. The room temperature resistance of the sampling thermistor is 10 kΩ while its $B_{25/100}$-Constant is 3988 K which leads to a resistance of 32.65 KΩ at 0 °C and 0.68 KΩ at 100 °C. Each thermistor is serially connected to a 10 KΩ 0.1% reference resistor (R_3-R_6) and both form a voltage divider whose configuration depends on the temperature of the object. The output voltage of this divider V_M - runs between 0 V-5 V ruling by the expression $5V\bullet\{R_{25}/(R_{25}+R_K)\}$. Where R_K - the thermistor resistance at temperature T_K – is defined by $R_K = R_{25}\bullet\exp\{B\bullet(1/T_K–1/T_{25})\}$. If we defined:

$$\alpha \equiv R_K/R_{25} = \exp\{B\bullet(1/T_K-1/T_{25})\} \qquad (1)$$

and

$$\beta \equiv R_{25}/(R_{25}+R_K) \qquad (2)$$

It is easy to see that $\alpha = 1/\beta-1$, thus the transformation between the temperature and the measuring voltage V_M is:

$$1/T_K = 1/T_{25} + 1/B\bullet\ln(1/\beta-1), \qquad (3)$$

where $\beta = V_M / 5V$.

The data acquisition unit – U_2 is made of a single Analog to Digital Converter (DAC) that operates with an internal reference and clock. The MAX197 [8] is a multi range, 12 bit data acquisition system, which requires only a single +5 V supply for operation. This converter provides 8 analog input channels that are independently software programmable for a variety of ranges: ±10 V, ±5 V, 0 V to +10 V, or 0 V to +5 V, while the input is protected against an over voltage of ±16.5 V. This converter supports an 8 + 4 bits parallel interface to a microprocessor, and an internal reference of 4.096 V. Four channels of this unit are connected to thermistors to monitor the temperature, while the other channels monitor the TEC's currents. Since the converter inputs current is several hundreds micro ampere, we buffered the inputs by U_7 - single-supply, rail-to-rail, CMOS operational amplifiers.

U_6 is a DC to DC converter that accepts 12 V-18 V in its input and converts it to 5 V on the output. In order to work with voltage above 18 V, one must replace this converter by others that work with 36 V.

4. Chipset Design

Digital designs are increasingly based on programmable logic devices. These devices are very flexible, inexpensive, and replace many basic logic parts with a single high-density device. The designs become very compact, small in size, and ultra-fast. There are several methods to design, synthesize, and simulate the content of the programmable logic: Hardware Description Language (HDL) as Verilog and VHDL, Schematic Editors, and more. This design is implemented by VHDL using Synplify 8.5 [9] to synthesize and Active-HDL 6.3 [10] to simulate the CPLD.

Our device - CY37128P84 from Cypress Semiconductor [11] - includes all the logic of the circuit and functions as a chipset of the system. Since the parallel port does not include an address bus, we must read or write any data in two cycles, first writing the address and afterwards reading or writing the relevant data. Moreover, due to the fact that the parallel port data bus consists of only 8 bits, while the converters resolution is of 12 bits, each analog data should map on 2 addresses in the chipset; low byte and high byte. By using this architecture each register in the chipset has its own local address:

0 – ADCL : read-low byte of ADC.
1 – ADCH : read-high byte of ADC.
2 – ADCS : write-start conversion / read-end of conversion.
3 – PWM1 : write-PWM1 duty cycle.
4 – PWM2 : write-PWM2 duty cycle.
5 – PWM3 : write-PWM3 duty cycle.
6 – PWM4 : write-PWM4 duty cycle.
7 – ENABLE : write-enable to the 8 PWM outputs.

The internal architecture of the CPLD is based on the state machine that is shown in Fig. 2.

At IDLE state, the machine is waiting for an Address Strobe (nA) or a Data Strobe (nD). If nA is detected, the machine starts an address write cycles, or, if nD is detected, the machine starts a data read/write cycle. At an ASTB state, the relevant address is saved in the internal register. At an STB state, the machine sets the proper chip selection to the peripheral, while the nWrite signal determines the cycle mode - write data or read data. At WRn and RD1 states, the machine transfers the specified data between the peripherals and the register, as can be seen in Fig. 2.

The implementation of the four bipolar PWM channels in VHDL is very simple. Fig. 3 shows a VHDL list in one process that implements all the PWM channels. The clock (clk) is 25 MHz while the counter (cnt) is 8 bits, thus, the modulation frequency is about 100 kHz (25 MHz / 2^8).

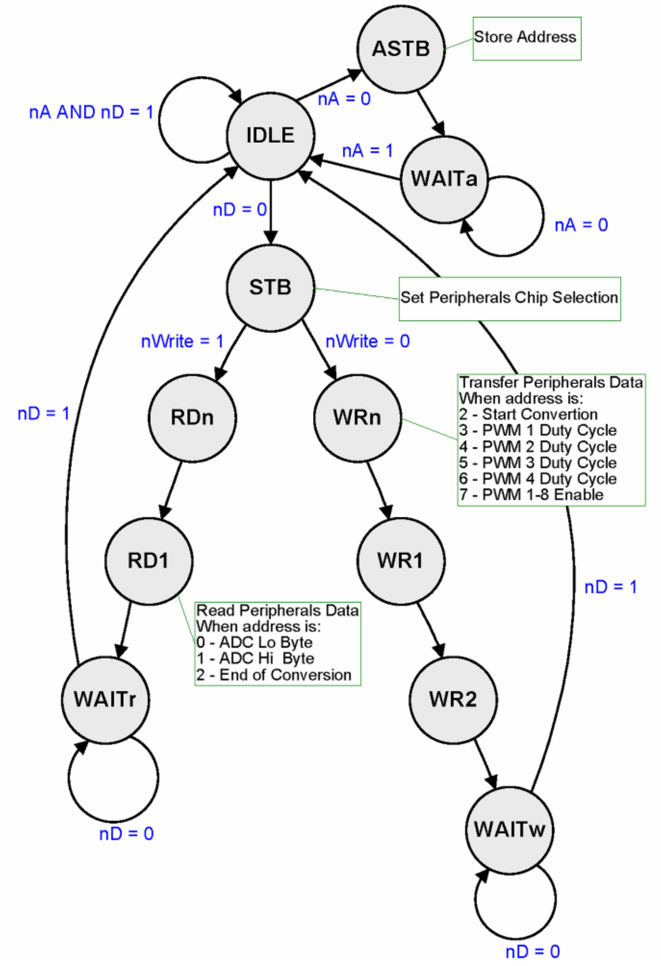
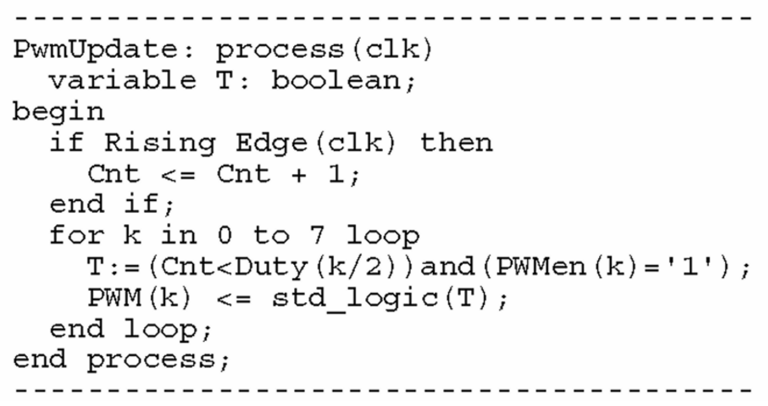


Fig. 2. The chipset state machines diagram.

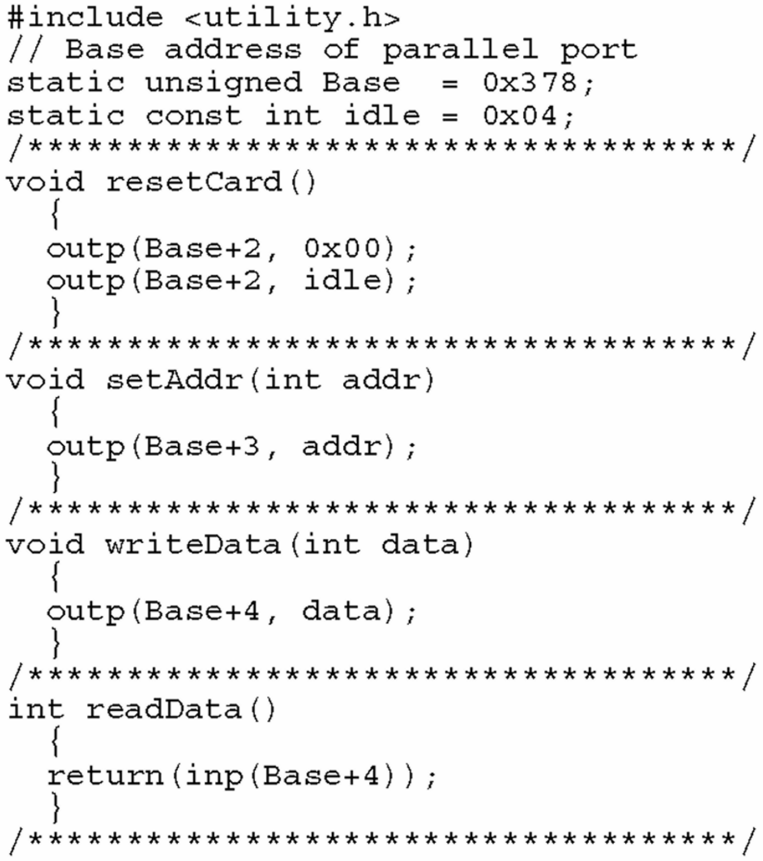


```
----------------------------------------
PwmUpdate: process(clk)
  variable T: boolean;
begin
  if Rising Edge(clk) then
    Cnt <= Cnt + 1;
  end if;
  for k in 0 to 7 loop
    T:=(Cnt<Duty(k/2))and(PWMen(k)='1');
    PWM(k) <= std_logic(T);
  end loop;
end process;
----------------------------------------
```

Fig. 3. A VHDL list in one process that implements the all PWM channels.

5. Drivers and Software

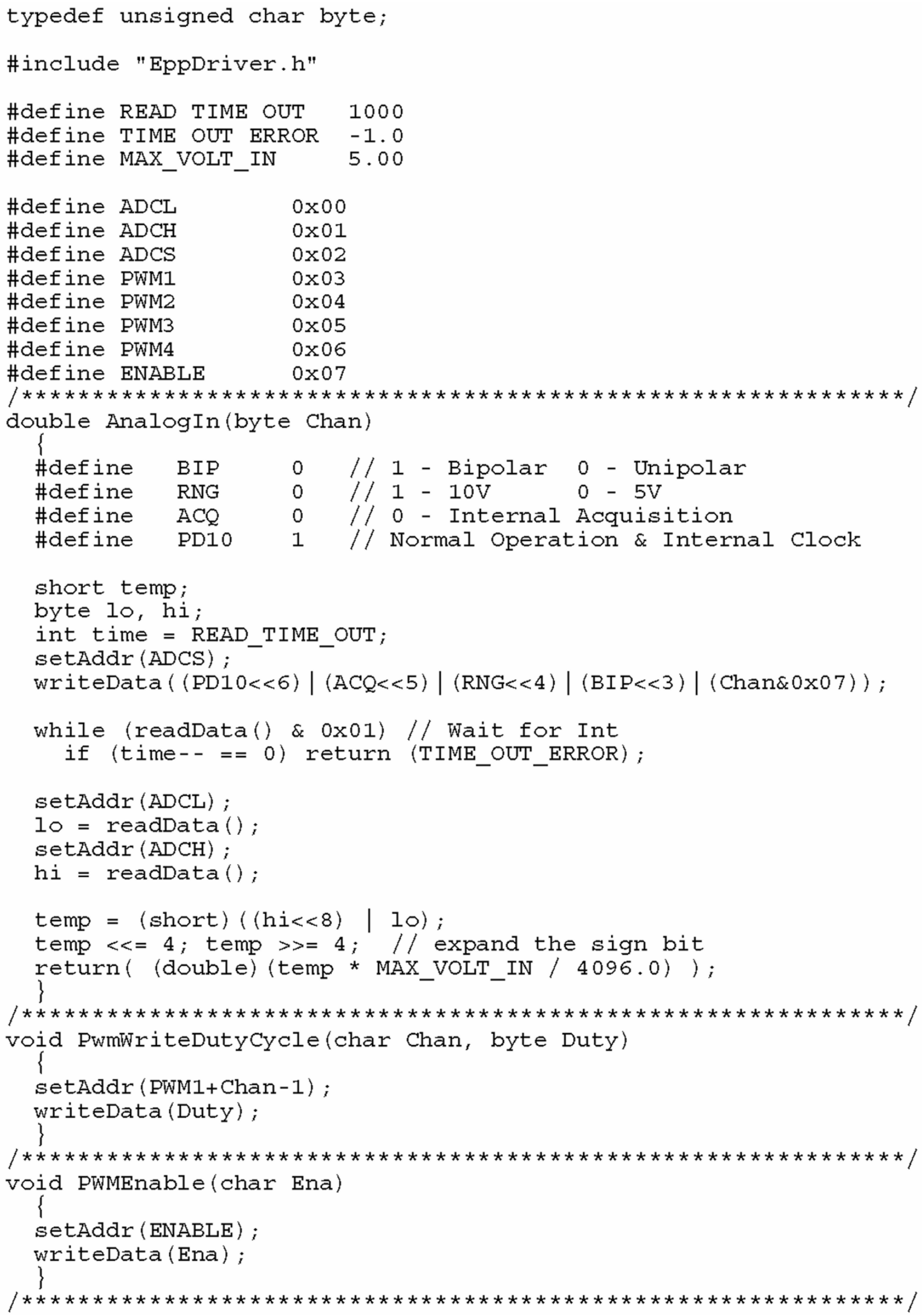
The communication between the PC and the chipset is achieved by a simple protocol. From the programmer’s point of view, one needs three low-level functions (drivers): setAddress, writeData, and readData to establish communication between the PC and the chipset. Above this low-level layer, the user needs a data acquisition library to translate the digital data to a real value (duty-cycle, temperature, etc.). These functions can be written in an assembly, C or any other high level language. We used C language to write both the low-level drivers and the data acquisition library. Fig. 4 and Fig. 5 show the low-level drivers that communicate with the parallel port in EPP mode and the data acquisition library.



```
#include <utility.h>
// Base address of parallel port
static unsigned Base  = 0x378;
static const int idle = 0x04;
/************************************/
void resetCard()
  {
  outp(Base+2, 0x00);
  outp(Base+2, idle);
  }
/************************************/
void setAddr(int addr)
  {
  outp(Base+3, addr);
  }
/************************************/
void writeData(int data)
  {
  outp(Base+4, data);
  }
/************************************/
int readData()
  {
  return(inp(Base+4));
  }
/************************************/
```

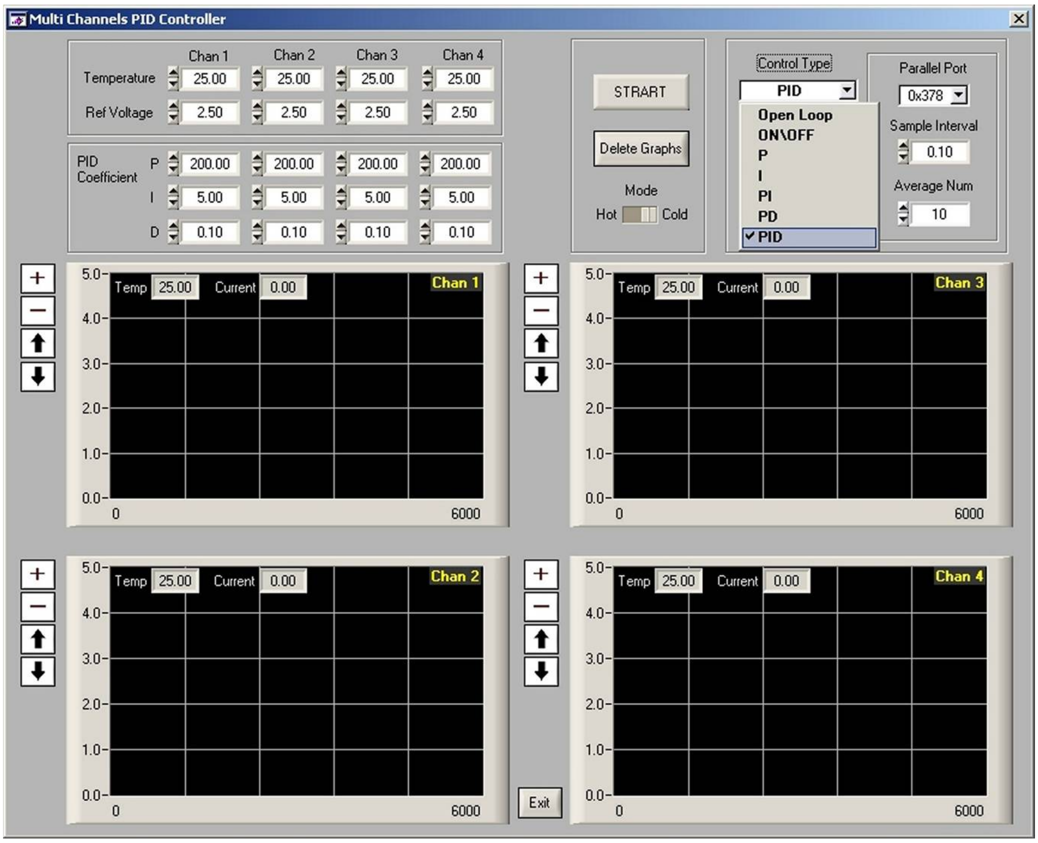
Fig. 4. The low-level drivers to communicate with the parallel port in EPP mode.

For the PC to control the system, we built a Graphical User Interface (GUI) using a National Instrument LabWindows CVI, as shown in Fig. 6. In this panel one sees a four strip chart - one for each control channel, targets temperature, PID coefficients, control type, TECs current, and sampled interval. The control type can choose from a list that includes: Open-Loop, On-Off, Proportional, Integral, Differential, as well as any of their combinations. As mentioned above, the PID loop is implemented by software, this means that the compensation expression is very flexible and in principal can even be a non-linear expression. The type of compensation and coefficient values should be determined by the user, as this will depend on the thermal response of the object and the TEC properties [12, 13, 14].



```c
typedef unsigned char byte;

#include "EppDriver.h"

#define READ TIME OUT   1000
#define TIME OUT ERROR  -1.0
#define MAX_VOLT_IN     5.00

#define ADCL        0x00
#define ADCH        0x01
#define ADCS        0x02
#define PWM1        0x03
#define PWM2        0x04
#define PWM3        0x05
#define PWM4        0x06
#define ENABLE      0x07
/**************************************************************/
double AnalogIn(byte Chan)
  {
  #define   BIP     0   // 1 - Bipolar  0 - Unipolar
  #define   RNG     0   // 1 - 10V      0 - 5V
  #define   ACQ     0   // 0 - Internal Acquisition
  #define   PD10    1   // Normal Operation & Internal Clock

  short temp;
  byte lo, hi;
  int time = READ_TIME_OUT;
  setAddr(ADCS);
  writeData((PD10<<6)|(ACQ<<5)|(RNG<<4)|(BIP<<3)|(Chan&0x07));

  while (readData() & 0x01) // Wait for Int
    if (time-- == 0) return (TIME_OUT_ERROR);

  setAddr(ADCL);
  lo = readData();
  setAddr(ADCH);
  hi = readData();

  temp = (short)((hi<<8) | lo);
  temp <<= 4; temp >>= 4;  // expand the sign bit
  return( (double)(temp * MAX_VOLT_IN / 4096.0) );
  }
/**************************************************************/
void PwmWriteDutyCycle(char Chan, byte Duty)
  {
  setAddr(PWM1+Chan-1);
  writeData(Duty);
  }
/**************************************************************/
void PWMEnable(char Ena)
  {
  setAddr(ENABLE);
  writeData(Ena);
  }
/**************************************************************/
```

Fig. 5. The C language data acquisition library.



Multi Channels PID Controller

		Chan 1	Chan 2	Chan 3	Chan 4
Temperature		25.00	25.00	25.00	25.00
Ref Voltage		2.50	2.50	2.50	2.50
PID Coefficient	P	200.00	200.00	200.00	200.00
	I	5.00	5.00	5.00	5.00
	D	0.10	0.10	0.10	0.10

Control Type: PID

Parallel Port: 0x378

Sample Interval: 0.10

Average Num: 10

Mode: Hot Cold

Chan 1: Temp 25.00 Current 0.00

Chan 3: Temp 25.00 Current 0.00

Chan 2: Temp 25.00 Current 0.00

Chan 4: Temp 25.00 Current 0.00

Fig. 6. The Graphical User Interface of the control program consists: four strip chart - one for each control channel, targets temperature, PID coefficients, control type, TEC current, and sampled interval. The control type can choose from a list that includes: Open-Loop, On-Off, Proportional, Integral, Differential, as well as any of these combinations.

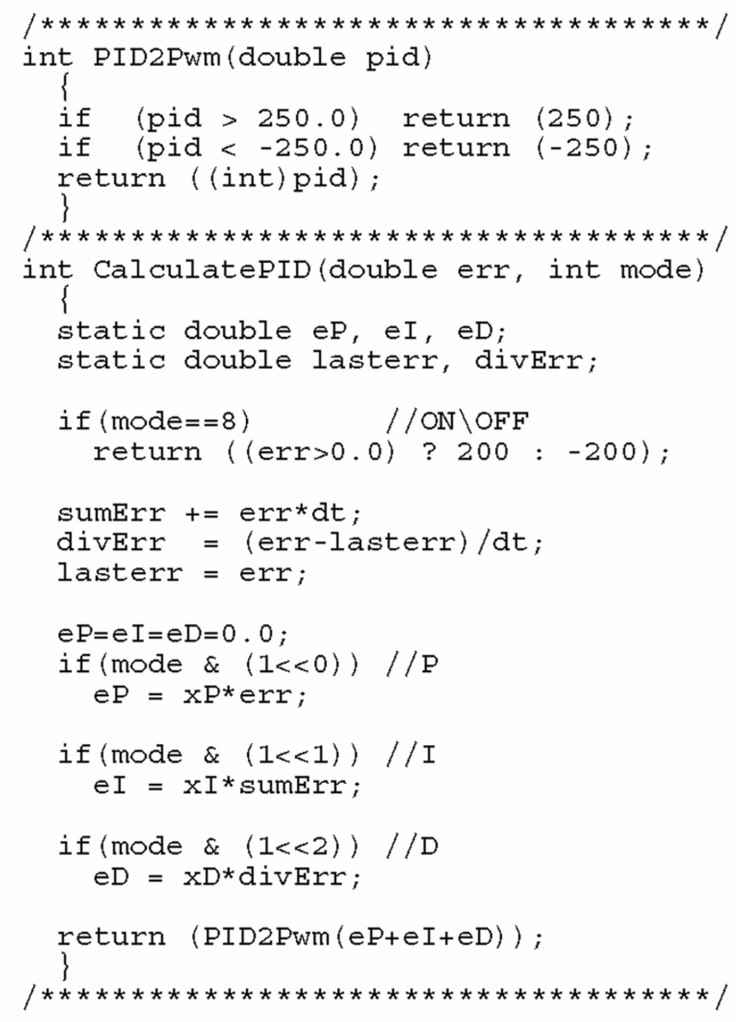
Fig. 7 is the programming list of the PID compensation loop that receives the error and calculates the compensation value in duty cycle units (-250 to +250). In all cases we send the absolute value of this value to the PWM controller, while the sign determines the direction of the current in the bridge.

The analog acquisition time of the controller is about 20 µs – much faster than the feedback-sampled interval, thus, there is no difficulty in simultaneously controlling all 4 channels. Moreover, since the loop sampling time is very short, we can average the sampled data 10 times for each point (to reduce the noise) and still then freeing the personal computer resources for other applications.

6. Expanding to 32 Channels

As one can understand, the design described above consists of 4 channels. Since we implemented the design by VHDL and PLD, it is therefore very simple to expand this controller to include more channels – using stronger CPLD. In order to expand the controller to 32 channels one is required to adjust the following:

- Replace the programmable logic device - U_1 by a stronger one – CY37512.
- Duplicate the entire analog and power units 8 times, using common data bus and separate chip selects.
- Expand the address register in the chipset to 64 addresses instead of 8 addresses.
- Change the loop index – K – in Fig. 3 from 7 to 63.
- Expand the commands in states WRn and RD1 of the state machine for 64 addresses.



```
/************************************/
int PID2Pwm(double pid)
  {
  if  (pid > 250.0)  return (250);
  if  (pid < -250.0) return (-250);
  return ((int)pid);
  }
/************************************/
int CalculatePID(double err, int mode)
  {
  static double eP, eI, eD;
  static double lasterr, divErr;

  if(mode==8)        //ON\OFF
    return ((err>0.0) ? 200 : -200);

  sumErr += err*dt;
  divErr  = (err-lasterr)/dt;
  lasterr = err;

  eP=eI=eD=0.0;
  if(mode & (1<<0)) //P
    eP = xP*err;

  if(mode & (1<<1)) //I
    eI = xI*sumErr;

  if(mode & (1<<2)) //D
    eD = xD*divErr;

  return (PID2Pwm(eP+eI+eD));
  }
/************************************/
```

Fig. 7. The programming list of the PID compensation loop that gets the error and calculates the compensation value in duty cycle units (-250 to +250).

As a result, the overall circuit is very compact in size (20 cm^2/channel) and very low in cost (10$ per channel) as compared to any commercial product.

7. Results

The controller, described here, can be used by any system that basically requires 1 to 32 channels of temperature control. We used the controller to regulate the temperature of water, including bacteria, in 2 inches Petri dish. We used a 1 inch 15V/2A TEC [15] as a heater/cooler element; when a 25x25x0.4 mm piece of copper with a heat capacity of ~ 1 J/C° is attached to the cold side of a TEC while its hot side is attached to the heat sink. The Petri dish is clamped to the copper piece when the temperature of this copper is sampled by the thermistor [7].

Fig. 8 shows a typical temperature time response produced by the TEC, at room temperature of 25 °C, when applying a 37 °C as the target temperature and using PID control. The feedback sampled interval was 0.1 sec and the data acquisition averaging is 10. The figure and data analysis show that the accuracy is better than 0.1°C while the stabilization time is about 2 minutes.

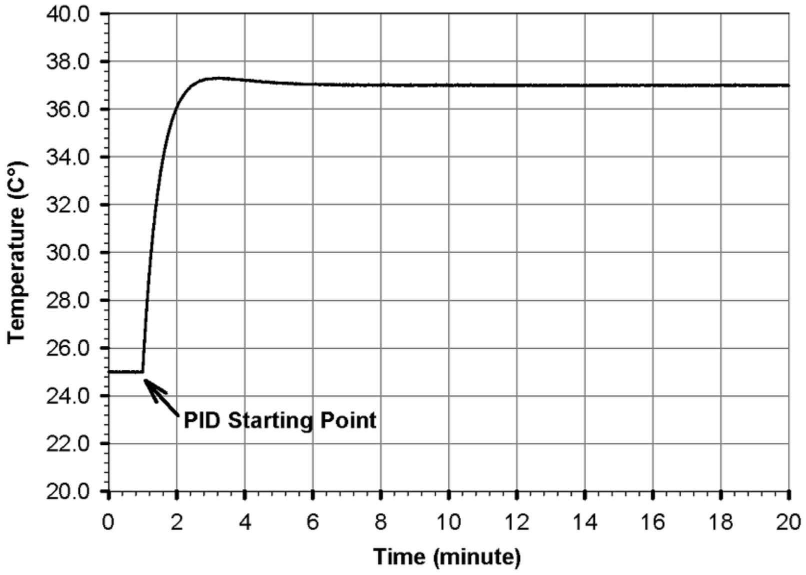


Fig. 8. A typical temperature time response produced by a TEC connected to the controller, at room temperature of 25 °C, when setting 37 °C as the target temperature. This figure shows that the stabilization time is about 2 minutes while the accuracy is around 0.1 °C.

8. Conclusion

In conclusion, we have demonstrated the complete design of a multi channels (4-32) PID controller for TEC that compatible to the several kinds of temperature sensors which can be used for systems that involve several channels of temperature control governed by a personal computer. Since we implemented the digital design by VHDL and PLD, it was therefore very simple to expand this controller to 32 channels, at the same time keeping the circuit very compact in size and at a very low cost compared to any commercial product.

Acknowledgment

I wish to thank Boris Fisherman for his help in building the setup of this experiment.

References

[1]. E. Flaxer, *Rev. Sci. Instrum.,* 74, 2003, p. 3862.
[2]. D. M. Rowe, CRC Handbook of Thermoelectrics, ISBN 0849301467, *CRC Press*, 1995.
[3]. J. Sharp, H. J. Goldsmid, George Nolas, Julian Goldsmid, Thermoelectrics: Basic Principles and New Materials Developments , ISBN 354041245X, *Springer Verlag*, 2001.
[4]. Marlow Industries, Inc, http://www.marlow.com
[5]. Mark W. Zemansky, Richard H. Dittman, Heat and Thermodynamics, ISBN 0-07-072808-9, *McGraw-Hill*, NY, 1981, pp. 431-438.
[6]. National Instrument: http://www.ni.com
[7]. Thermistors model - Epcos B57703M0103G040.
[8]. Maxim: http://www.maxim-ic.com
[9]. Synplicity: http://www.synplicity.com
[10].Aldec: http://www.aldec.com
[11].Cypress Semiconductor: http://www.cypress.com
[12].Richard C. Dorf, Robert H. Bishop, Modern Control System, ISBN 0-201-84559-8, *Addison Wesley*, NY, 1995, pp. 662-670.

[13].John J. D'Azzo, Constantine H. Houpis, Linear Control System, ISBN 0-07-113295-3, *McGraw-Hill*, NY, 1995, pp. 352-366.

[14].John Van De Vegte, Feedback Control System, ISBN 0-13-191503-7, *Prentice_Hall*, NJ, 1994, pp. 246-257.

[15].TEC model - Melcor CP 0.8-127-06.

Guide for Contributors

Aims and Scope

Sensors & Transducers Journal (ISSN 1726-5479) provides an advanced forum for the science and technology of physical, chemical sensors and biosensors. It publishes state-of-the-art reviews, regular research and application specific papers, short notes, letters to Editor and sensors related books reviews as well as academic, practical and commercial information of interest to its readership. Because it is an open access, peer review international journal, papers rapidly published in *Sensors & Transducers Journal* will receive a very high publicity. The journal is published monthly as twelve issues per annual by International Frequency Association (IFSA). In additional, some special sponsored and conference issues published annually.

Topics Covered

Contributions are invited on all aspects of research, development and application of the science and technology of sensors, transducers and sensor instrumentations. Topics include, but are not restricted to:

- Physical, chemical and biosensors;
- Digital, frequency, period, duty-cycle, time interval, PWM, pulse number output sensors and transducers;
- Theory, principles, effects, design, standardization and modeling;
- Smart sensors and systems;
- Sensor instrumentation;
- Virtual instruments;
- Sensors interfaces, buses and networks;
- Signal processing;
- Frequency (period, duty-cycle)-to-digital converters, ADC;
- Technologies and materials;
- Nanosensors;
- Microsystems;
- Applications.

Submission of papers

Articles should be written in English. Authors are invited to submit by e-mail editor@sensorsportal.com 6-14 pages article (including abstract, illustrations (color or grayscale), photos and references) in both: MS Word (doc) and Acrobat (pdf) formats. Detailed preparation instructions, paper example and template of manuscript are available from the journal's webpage: http://www.sensorsportal.com/HTML/DIGEST/Submition.htm Authors must follow the instructions strictly when submitting their manuscripts.

Advertising Information

Advertising orders and enquires may be sent to sales@sensorsportal.com Please download also our media kit:
http://www.sensorsportal.com/DOWNLOADS/Media_Kit_2008.pdf

Smart Sensors Systems Design

A five-day advanced engineering course

10-14 November 2008, Barcelona, Spain

SMARTSES

Marie Curie Chair (EXC) project

General Information

This course is suitable for engineers who design different digital and intelligent sensors, data acquisition, and measurement systems. It is also useful for researchers, graduate and post graduate students. Course will be taught in English.

Course Description

An advanced engineering course describes modern developments and trends in the field of smart sensor systems and digital sensors design.

After a general overview of data acquisition methods, modern smart, digital and quasi-digital sensors, smart systems details are discussed. A systematic approach towards the design of low-cost high-performance smart sensors systems with self-adaptation and self-identification possibilities is presented.

Contact Person

Susana Escriche
Fundació UPC. Edifici Vèrtex
Plaça Eusebi Güell, 6, 08034 Barcelona
Tel.: +34 93 401 08 94
E-mail: susana.escriche@fundacio.upc.edu

Course Instructor

Prof. Sergey Y. Yurish,
Centre de Disseny d'Equips Industrials (CDEI),
Universitat Politecnica de Catalunya (UPC-Barcelona)
Tel.: + 34 93 401 74 37, fax: + 34 93 401 19 89
E-mail: syurish@sensorsportal.com

Online Registration:

http://www.sensorsportal.com/HTML/SSSD_Course_2008.htm

Deadline for Registration:

31 October, 2008

Fundació UPC   MARIE CURIE ACTIONS   IFSA

www.sensorsportal.com