

Arduino Applied in Temperature Controller Design for Photonic Devices ---Based on Engineering Students' Creativity Development Program

Yuanyuan Chen¹ and Xinran Zhao²

Abstract. In an optical communication system, providing a stable and constant temperature environment for the optical chip is the key for a less loss communicating performance.

In this paper, a new design of temperature controller for photonics is set up based on the combination of the Thermoelectric Coolers (TEC) and Arduino. Through the process of the whole design, as we rely on Arduino to collect the temperature data and make the corresponding response according to the varying temperature. Thus, the design of the coding also plays a significant role in this project.

One great merit of using the coding to play a role as a controller and data collector, the characteristic and the function of one specific code do not change when the environment change. It can offer a stable constant temperature environment within $\pm 0.05^\circ\text{C}$. The temperature controller for photonics offers a reliable temperature stabilizing approach for the optical chip. At last, it also offers a simple as well as useful structure of temperature controller which can be a useful reference to other fields besides the optical communication area.

1. Temperature controlling demand for the optical devices

As the developing trend of the optical communication is moving towards the miniaturization and integration and the increasing widely use of the erbium-doped fiber as well as the RAMAN amplifier, it is required that the pump source is able to work under some specific wavelength such as 980nm and 1480 which are the typical widely used wavelength. However, the pump laser is usually in continuous or pulsed operation mode, thus only a small proportion of the energy provided by the laser input will be converted to the laser output whilst the rest of the energy is converted to the heat loss. The output of the laser is significantly influenced by the heat which is generated by the working materials. This is because the increase in working materials' temperature will lead to the fluorescence spectrum broadening and the decrease of the quantum efficiency. In addition to that, the threshold current of the laser will be increased according to the equation $I(t) = I_0 + I_t e^{\frac{T}{T_0}}$ and the efficiency will be decreased on the other hand. The working substance not only absorbs the light radiation from the pump but also leads to the uneven inner temperature distribution which results in the heat pressure, pressure birefringence and the thermal lensing effects. In order to have a last output both efficiently and stable, the temperature of the laser need to be strictly controlled. One of the traditional methods is to use a water-cooling approach which is not only bulky, uneven cooling feature but also cannot strictly control the laser's temperature.

2. Temperature reading

Through the process of the whole design, as we rely on Arduino to collect the temperature data and make the corresponding response according to the varying temperature. Thus, the design of the coding also plays a significant role in this project.

2.1 Methodology of getting the temperature data from a thermistor

As the primary factor that directly affect the action of the controller, the reading method of the thermistor is as important as the sensor selection. To read the temperature from the thermistor, we need to get the resistance so that the Steinhart-Hart equation can be used to do the resistance-temperature conversion. However, the Arduino does not have a built-in resistance meter so that we need to come up with an indirect way to get the resistance. Further looking into the feature of Arduino, the function of the voltage reader which is known as the analog-digital converter can be taken advantage of so that the resistance can be converted into a voltage.

To do the conversion, a voltage divider is a straightforward and efficient way by connecting the thermistor with another resistor. And in this way, simply by measuring the voltage in the middle, when the resistance of the thermistor varying according to the temperature, the voltage changes respectively. As the 10K Ω thermistor is chosen in the Sensor Selection Part, the

fixed resistor is also $10K\Omega$, and the thermistor is called R_t , the output voltage V_0 is

$$V_0 = \frac{R_t}{R_t + 10K} \times V_{cc} \quad (1)$$

where V_{cc} stands for the power supply voltage (3.3 V or 5 V).

When to connecting the voltage divider to the Arduino, measurement of the voltage into an Arduino ADC, what we get is

$$ADC \text{ value} = \frac{V_i \times 1023}{V_{cc}} \quad (2)$$

make $V_0 = V_i$,

$$ADC \text{ value} = \frac{R_t}{R_t + 10K} \times V_{cc} \times 1023 \div V_{cc} \quad (3)$$

then it can be noticed that the V_{cc} can be cancelled out, thus

$$DC \text{ value} = \frac{R_t}{R_t + 10K} \times 1023 \quad (4)$$

And finally, we can get the equation which can convert the ADC value into the resistance of the thermistor,

$$R = \frac{10}{\frac{1023}{ADC} - 1} \quad (5)$$

By converting the equation into the Arduino Coding, it is found that one of the problems that occur during the temperature reading is that, in a relatively stable environment, the temperature measured is unstable which varies around . Taking a further looking for the reason that may cause this problem, we found that the Arduino itself is quite ‘noisy’ which makes the ADC value unstable even in a stable condition. The results are then improved by applying two improvements to the code. One takes the 3.3V voltage pin as an external analog reference and the second development is use an array to read several data and calculate an average value to reduce the errors. By connecting the 3.3V to AREF and use it as the V_{cc} voltage, the noise can be significantly reduced sin the 5V power supply which comes from the PC is always more noise as it does a lot of functions and actions.

The following step is to convert the resistor to the temperature which is exactly what we need to get. This step can be done only change the equation

$$\frac{1}{T} = A + B \ln(R_t) + C [\ln(R_t)]^2 \quad (6)$$

into the Arduino Coding language.

2.2 The Real-time Represent of the Temperature

2.1.1 Liquid Crystal Display (LCD)

Besides to the consideration of the accuracy of the thermistor reading, another consideration is that the real-time presentation of the current temperature which is measured by the two thermistors. Especially, as mentioned in the following chapters, the SD card based data collecting function is used and it became quite hard to get the up-to-time temperature without put out the SD card from the

controller. Thus one screen is needed so give a simple, straightforward approach for the users to get accessed to the temperature.

One common way is using the LCD screen which is every cheap and quite easy to access.

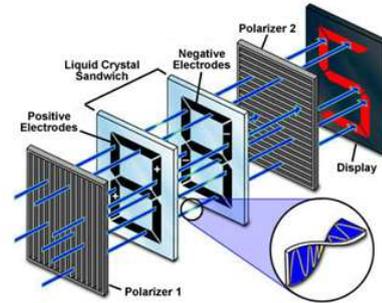


Figure 1. Demonstration of a seven-segment liquid crystal display

To get this type of LCD screen works properly, the drive circuit must be used to drive each segment digital code, so if we want to show the number, it depends on the driving mode of the digital control approach. The main driving method can be concluded as following:

The static drive, which is also known as the DC drive, refers to that each digital code for each segment is driven by a microcontroller I/O port or, use the two-decimal code such as BCD decoder to drive. This driving method has the advantages such as simple programming, high displaying brightness. However, the drawbacks of this kind of driving method is that it occupies too many ports. For instance, when 5 digital static display is needed, the port number we need is $5 \times 8 = 40$ ports which is a very large number of ports considering that a 89S51 microcontroller only has I/O ports available in total. And in the actually application, the decoding drive must be applied which increases the complexity of the hardware circuit.

However, when deciding which screen to choose, only 5 pins are available on the Arduino and in this case, another option is more suitable.

2.1.2 Organic Light-Emitting Diode

Organic Light-Emitting Diode, with the characteristics of thin, flexible style, outstanding portability and low power consumption, has become a new trend for the nowadays and future display technology.

The main idea of the OLED is by adding a bias voltage, the electrons move through the whole transfer layer and the electron transport layer and enter in an organic substance. In this substance, the recombination occurs and entered in the condition of exciton. The energy is released soon after that and the electron return to the ground state. In the energy released, due to the characteristics of the luminescent material selection and electron spin of a ‘photon excitation’, only 25% of the energy is transferred into light and the rest of it is a phosphorescent or in the form of heat. Depends on the different band gap of the chosen materials, this 25% of the energy can be released into different colors which is shown in the OLEDs.

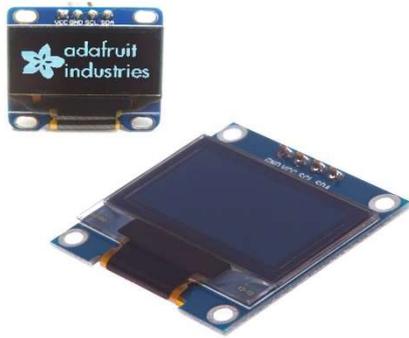


Figure 2. A display example of the OLED screen

Besides of the characteristic of the OLED that it occupies less Pins on the Arduino, another advantage is that the OLED can represent more information in a very small area as 27.0mm × 27.0mm × 4.1mm which is used in the temperature controller.

The OLED which is finally chosen is the 0.96" I2C IIC 128X64 OLED White. It has a size of 27(mm) × 27(mm) × 4.1(mm) and a very high resolution of 128 × 64 . In addition to that, the suitable working temperature of this OLED is also desirable which -30 to 70. Thus the OLED is chosen to be the screen of the temperature controller to represent the room temperature, measured Temperature and other information such as the current power or current time.

The corresponding code is generated and figure of display of the screen is shown as following.



Figure 3. A display example of the OLED screen applied on the thesis

3 Data logger function

Above all the circuit design and the function coding, one thing which is also of much importance is the data logging method. Without it, the performance cannot be vividly presented to us, the engineers and thus we cannot get to know what and where the problems and neither can we find where the improvement can be made.

Through the whole design process, two main methods of data logging are used considering the condition that the controller can be used in both of the conditions with or without an access of PC. The main design idea of these two approaches can be described as following.

3.1 PC Based Logging Approach

First, one is under the circumstance of the controller is used with an access of a PC. In this condition, what is done is the data collected by the Arduino is print on the serial monitor directly use the function of Serial. Print () and use the CSV coding to allow the Excel collect the data from the serial port. To give a better represent of this method, the picture is made as followed to show the data flow direction.

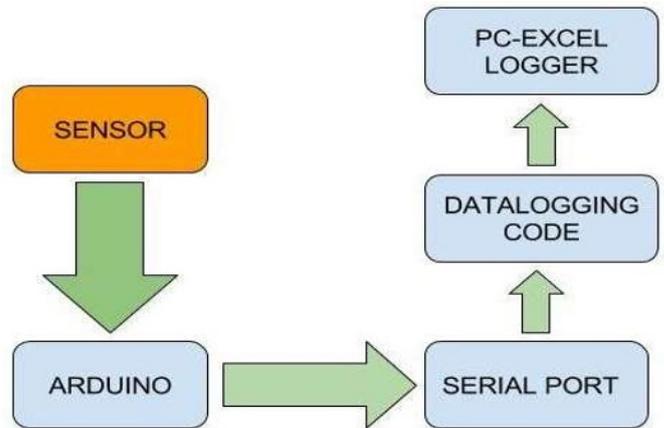


Figure 4. Data Logging Method based on PC.

The data flow, labeled in green, collected by the sensor and passed by Arduino to the PC based logger. At the first few versions of the logger design, one function of the real-time data logging for the maximum, minimum and the average temperature of every single round (around 1000) is added. However, as the number of the temperature data growing from a few hundred which is roughly equids to several minutes to several thousands, the problem raised is that, the total code assigning into the Arduino is running out of range and the function cannot be performed in a good stability.

After a detailed debugging, the reason that causes the out of-range size of the controlling method is that 1. The simple counting method occupies a lot of spaces when using the Max, Min, Average functions as the array used to save these data increased to a huge size, and also the stability of the PC based data logger is significantly affected when one single sheet of the excel contains tens of thousands of data. One straight example is that the excel may crush down. And thus, one of the improvements is the method that we count.

4 Data logging approach based on sd card

The second try on the data logger design is using the SD card to save the received data under the circumstance of the computer is not or hardly accessible, especially considering the controller should be more portable. Following the idea in the 3.2.1 which describe in details how we change the resistance of the thermistor into the temperature data that we want, what we need to do is just wrap the data into different files and when the computer is accessible, the data analyzing software can be used to give a further analyze.

With the help of the head file SD library and the Data Logger Shield which is already a widely used product with a reasonable price shown as following.



Figure 5. SD shield of Arduino

And this function has already been added into the project and in the analysis part. Both of these two data logging method is used and compared and the future customer can pick one method from them depends on what they need.

With multiple test throughout the whole project process, the SD card data logging method, the maximum sampling rate is 35Hz which is very close to the maximum sampling rate that the code can reach without any data logging function, which is 47Hz. However, when it comes to the data logging method based on the Excel, the maximum sampling rate is just 2 Hz, any higher sampling rate will cause a disorder in the data logging, such as the randomly blank cell which is not be desired.

This is mainly because the time delay during the Excel based data logger reading the data from the serial monitor. As the pointer is used at the first stage and the CSV is used in the second stage, and also taking the operation on the PC into the consideration, the chance of an error happens is incredibly increased leaving a very unstable condition of a data logger.

Thus, when choosing which way to use for a data logger, the PC base data logger is more suitable when a real time monitoring and plotting is needed and when it comes to a demand of a long-time data logging and more portability, SD card based data logger is a better choice for the customers.

Acknowledgement

This research is financially supported by the Fundamental Research Funds for University of Science and Technology Liaoning (Grant NO. XJGJF201831).

References

1. D. Pollock, Ed., The Theory and Properties of Thermocouple Elements, STP492-EB, ASTM International, West Conshohocken, PA, 1971, page 15
<http://dx.doi.org.ezproxy1.library.usyd.edu.au/10.1520/STP492-EB>
2. Qingjun Cai¹, Ya-Chi Chen, Chialun Tsai and Jeffrey F DeNatale, Development of a platinum resistance thermometer on the silicon substrate for phase change studies, 4 July 2012
3. A High-Quality Platinum Resistance Thermometer to 661°C, Mingjian Zhao, Xumo Li and Deming Chen
4. Thermistor Basics, May, 2013 Page 1; Application Note AN-TC11 Rev. A; Wavelength Electronics
5. Thermoelectric thermal regulation systems; Robert DeVilbiss; May 1996; published on Electronics Cooling
6. I. O'Connor, F. Gaffiot, "Advanced Research in on-Chip Optical Interconnects". Lower Power Electronics and Design, 2004.
7. Chai Yeh, Applied Photonics. Academic Press. Inc. 1992
8. Yang, J.; Liu, Y.; Rauch, C.B.; Stevens, R.L.; Liu, R.H.; Lenigk, R.; Grodzinski, P. High sensitivity PCR assay in plastic micro reactors. Lab Chip 2002, 2, 179–187.
9. <http://www.buyincoins.com/item/45381.html#.V01H7kaHqZY> ; New 0.96" I2C IIC Serial 128X64 OLED LCD LED Display Module for Arduino
10. <https://radhesh.wordpress.com/2008/05/11/pid-controller-simplified/> ; PID Controller Simplified