RESEARCH ARTICLE

Performance of thermoelectric cooling system with smart graphical user interface for solidifying liquid sample

Hanin Athirah Harun*, Roslinda Zainal

Department of Physics, Faculty of Science, Universiti Teknologi Malaysia, 81310 Johor Bahru, Johor, Malaysia

* Corresponding author: hathirah2 @live.utm.my

Abstract

A Thermoelectric Heat Pump (THP) controller system with smart Graphical User Interface (GUI) was introduced to solidify liquid samples for laser induced breakdown spectroscopy (LIBS) analysis. This paper describes the smart GUI and THP controller system based on the Arduino platform. The THP controller system was built to provide a user-friendly smart GUI for controlling the Peltier Thermoelectric Cooler (TEC) temperature and monitoring the sample temperature acquired from the temperature sensor. Instructions on the construction of the smart GUI and THP controller circuit were explained in this paper. The experimental results on solidifying distilled water and maintaining its freezing phase are presented in this work that demonstrated the excellent performance of the developed system.

Keywords: Thermoelectric heat pump, Peltier device, laser induced breakdown spectroscopy, liquid sample, Arduino

INTRODUCTION

Laser induced breakdown spectroscopy (LIBS) technique is one of the most favorable analytical technique due to its minimum sample preparation and rapid analysis [1,2]. Some recent applications of LIBS are materials science [3], coal analysis for power plants application [4], arts and artifacts cultural heritage [5], agriculture [6], biological samples [7], and others. Nowadays, more attention has been given on applying LIBS technique on liquid samples. Unlike solids LIBS measurements, direct LIBS analysis on the bulk or the surface of liquid samples could cause some difficulties such as splashing, surface ripples, and shorter plasma duration [8]. To solve these drawbacks, some approaches have been suggested such as horizontal [9] and vertical liquid jet system for laminar flow [10,11], liquid to aerosol conversion [12,13], liquid sample in droplet form [14-16], and liquid to solid matrix layer conversion [17,18]. However, some of these methods involve complex experimental configuration, impractical for real time on site measurements, and unsuitable for a limited or hazardous sample.

Liquid to solid conversion by freezing is one of the simplest sample preparation methods in reducing splashing. Once splashing is eliminated, the laser repetition rate can be increased, resulting in emission enhancement and improved LIBS measurement [19,20]. This method also involves less complicated laser or fiber coupling arrangement, unnecessary liquid optical transparency, and practical to be handled [21]. The most important precaution when dealing with frozen sample is controlling the sample temperature to ensure accurate LIBS measurement. This is because, the sample temperature could also affect the ablation rate and plasma intensity [22]. However, since most of the LIBS experiment involving liquid to solid conversion by freezing usually use liquid nitrogen, the sample temperature is quite difficult to control [19, 20,23].

In response, we propose the Thermoelectric Heat Pump (THP) controller system with an open source smart Graphical User Interface (GUI) to facilitate a more effective liquid to solid sample conversion method. In our previous work [24], we demonstrated the feasibility of using Thermolectric Cooler (TEC) to freeze the samples prior to the LIBS analysis. Our results showed that frozen samples provided improvement on the spectral signal quality when compared to the liquid samples due to increased ablation rate and better laser-sample interaction.

The purpose of the present work is to describe the development, calibration, and performance of the THP controller system equipped with smart GUI. The term "smart" GUI was reflected on the goal of this study which is to monitor and control the THP controller system simultaneously. The features included in the THP controller system with the user-friendly smart GUI are as listed below:

- Affordable and easy-to-assemble THP controller circuit;
- Designed and constructed using open source software namely Arduino and NetBeans 8.1 Integrated Development Environment (IDE);
- Simple serial communication procedure;
- Able to freeze the sample within a short period of time with the aid of Peltier Thermoelectric Cooler (TEC);
- Temperature measurement accuracy similar to standard mercury thermometer;
- Real-time temperature measurement output display and plot; and
- Data storage in *.xlsx file format.

For reasons explained in the next section, we chose to use Java with Arduino to develop the system, along with NetBeans IDE as the development platform of the smart GUI.

THE SYSTEM

Arduino UNO microcontroller

Arduino is an open source microcontroller, based on the Atmega328P microprocessor. This well-known electronic prototype is established to create control devices for projects that are inexpensive to obtain and easy to use [25]. The Arduino platform is widely used in many applications including in ultrasonic radar system [26], agriculture [25], construction and building [27,28], and portable fluorimeter [29]. The vast applications of this microcontroller is owing to its desirable characteristics: affordable; easily accessible open source hardware platform; small board size (68.6 mm × 53.3 mm) which makes it suitable as a portable device; built-in high speed USB interface for data transfer; considerable I/O peripherals for control purpose; and free available IDE [30]. These criteria prove its reliability and making it a viable choice to be utilized in project developments.

The Arduino IDE supported a simplified C/C++ programming language and is capable of running on all major operating systems [27]. A conventional Arduino code consists of two main functions, the setup and loop, that are compiled and linked with a program *main ( )* into an executable cyclic executive program. The *setup ( )* function indicates the beginning of the program with the intention to set the environment in which the program will run. The Arduino program also comes with a *loop ( )* function where the function will be called repeatedly until the microcontroller board powers off [30].

However, the IDE has some disadvantages such as lack of selective code folding for hide and display feature, error information, auto completion in source code editor, and analyzing complex code structures. Therefore, NetBeans IDE 8.1 which is a software development platform written in Java was chosen as an additional programming platform to develop the smart GUI. Although Arduino is a bit too constrained to run Java directly, communicating and controlling an Arduino from Java running on another computer would be easier with the help of serial port communication software. Within the Java language, there are also several libraries intended to promote communication with the serial port of the computer to establish communication with Arduino [30].

Peltier Thermoelectric Cooler (TEC)

Peltier thermoelectric heat pump is a popular thermoelectric energy conversion module that transfers heat energy from one side of the device to the other side. Peltier effect occurs when two different semiconductor materials (n-type and p-type) are joined together with electric current passing through its junction. Therefore, the junction could either be cooled or heated [29]. The advantages of using this device are long lifetime, environmental friendly, satisfactory performance, high efficiency, no maintenance or complex water distribution pipes requirements, noiseless, and light-weight. Due to the aforementioned advantages, thermoelectric devices have been applied as heaters, coolers, or power generators [31].

Fig. 1 shows a Peltier device with voltage being applied at both free ends of the semiconductors. Direct current (DC) flows across the materials' junction creating temperature difference along its path. The heat lost is transferred to the heat sink. Adjusting the power supplied to the Peltier circuit would change the temperature gradient and thus the heat loss of the device, which is the main focus of this study. Pulse width modulation (PWM) modification is implied to control the amount of power delivered to the load (Peltier TEC device).
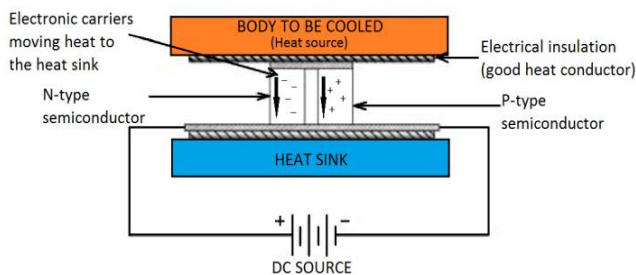

Fig. 1 Peltier device.

Pulse width modulation (PWM)

Pulse width modulation (PWM) is a technique applied for controlling analog circuits with digital outputs of the microprocessor (Arduino). PWM was employed in this study to control the power of the Peltier device by adjusting the duty cycle percentage. The percentage of the duty cycle specifically indicates the percentage of time a digital signal is on over an interval or period of time, where the period is the inverse of the frequency of the waveform. Changing the pulse width would also change the percentage of the duty cycle and therefore averaging the output power signal [32]. Six of Arduino 14 I/O pins can be implemented for PWM purpose (8 bits decimal integer with 0-255 range) using *analogWrite* whereas all the pins can be performed with *digitalWrite* method [25]. The *analogWrite* method was chosen because of the simpler calculation needed to get the preferable duty cycle [33].

The *analogWrite* method is divided into three parts, including: (1) variables initialization; (2) void setup: *pinMode* function specifies pins or variables as input or output; and (3) void loop: codes location that run in an infinite loop. By using the *analogWrite (a, b)* function, the *analogWrite* method could generate square waves by allocating *a* and *b* as the pin and the duty cycle value, respectively. Equation (1) and equation (2) explain the relation between duty cycle, bits, and voltage [33].

$$Duty\ cycle\ x\ 255 = analogWrite\ value \qquad (1)$$

$$analogWrite\ value = \frac{(255\ x\ Analogue\ voltage)}{(maximum\ voltage)} \qquad (2)$$

The *analogWrite value* in the equations above is equivalent to *b* value which is the 8-bit decimal integer. The current of the Peltier element, I_{avg} can be calculated using equation (3)

$$I_{avg} = (I_p\ x\ t_p)/T \qquad (3)$$

From equation (3), I_p is the operating current, t_p is the pulse width and T is the period (256.7 μs).

METHODOLOGY: HARDWARE AND SOFTWARE DEVELOPMENT

Fig. 2 shows the basic layout of a Thermoelectric Heat Pump (THP) controller system in schematic form. This system included the Arduino UNO microcontroller attached to a computer using a universal serial bus (USB) cable, a power supply unit (PSU) connected to a metal oxide semiconductor field effect transistor (MOSFET), a thermoelectric cooler (TEC) with attached heat sink and fans (also connected with PSU), and temperature sensor. The construction of the whole system was divided into two sections: (1) Software: development of smart Graphical User Interface (GUI) and (2) Hardware: development of Peltier Thermoelectric Cooler (TEC) controller circuit.
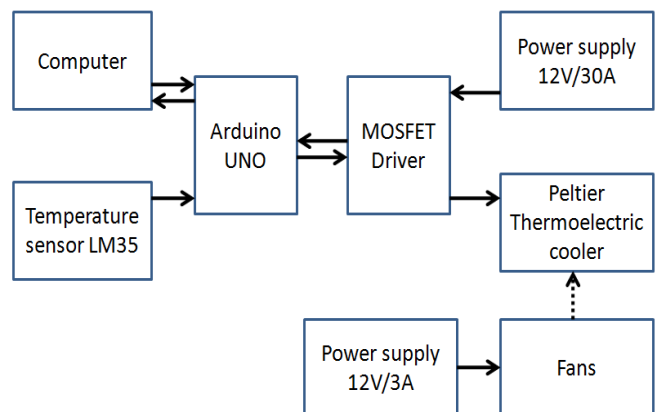

Fig. 2 Basic layout of a THP controller system.

Development of smart GUI

The development of the smart GUI is divided into several parts, namely serial communication between Arduino and NetBeans, PWM modification, output temperature display acquired from LM35, and data storage, as shown in Fig. 3.
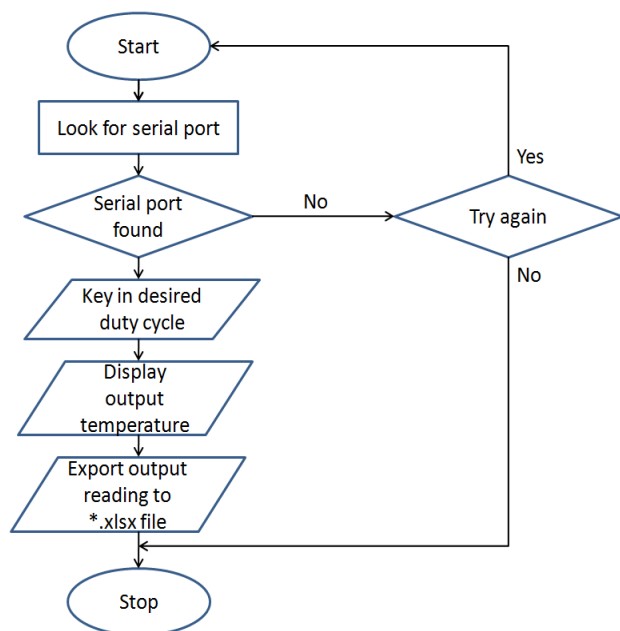

Fig. 3 Flow chart of the smart GUI of THP controller system.

a) Serial communication

In order to provide a user-friendly serial communication procedure, this GUI is equipped with three easy steps to promote the communication of the computer and the Arduino via Java application. They are: (1) find the serial port, (2) Connect/Disconnect the Arduino to computer, and (3) Start/Stop the operation.

First, the smart GUI was designed with a dialog box with selections of available COM port. To do so, an open source library called RXTX is required by Java to send and receive information through the corresponding serial port. The *RXTXcomm.jar* was added as a library under the NetBeans Project for interfacing the Arduino board with NetBeans IDE 8.1 through a serial connection. A refresh button was also added as an option to re-select the port in case the user mistakenly selects a wrong port.

After successfully finding the COM port where the Arduino is connected, a "Connect/Disconnect" was added to confirm the start of the Arduino-Computer communication. "Start" and "Stop" buttons were also available to start and stop (or pause) the whole THP controller system to give way to saving the data collected.

b) PWM modification

The temperature generated by the THP was controlled by the PWM. The smart GUI is equipped with a slider that allows the user to vary the duty cycle easily. The Arduino programming command used to facilitate this modification is as shown in command line 1. This command line is situated in the *loop ( )* function so that it runs repeatedly.

Command line 1 – PWM modification

```
if (Serial.available()>0){        //duty cycle setting section
    input=Serial.read()-48;
    delay(10);
    c=Serial.read()-48;
    delay(10);
    d=Serial.read()-48;
    delay(10);
    u=Serial.read()-48;
    delay(10);
    p=(100*c)+(10*d)+u;
    if (input==1){
```

```
analogWrite(10, p);//MOSFET connected to pin 10
    }
```

The *Serial.read ( )* method displays the integer in ASCII format. In ASCII, the numbers 0 to 9 correspond to the values from 48 (0) to 57 (9). It is necessary to minus 48 after the *Serial.read ( )* function to remove 48 after the ASCII number and display the digits from 0 to 9. Furthermore, c, d, and u correspond to three place values (hundreds, tens, and ones) of the duty cycle. A delay of 10 ms after each *Serial.read ( )* is important to provide a sequence when transmitting the data.

The MOSFET is connected to pin 10, which provides a frequency of 500 Hz to the PWM. However, it is more suitable to use higher PWM frequency for THP application. Therefore, the PWM frequency of pin 10 output was changed to ~3895 Hz by adding the line (*TCCR1B = TCCR1B & 0b11111000|0x02;*) under the *setup ( )* function in the Arduino source code.

c) Output temperature display and plot

LM35 is a temperature sensor device that provides an output voltage that is linearly proportional to Centigrade temperature. This low-cost and low-impedance device is suitable for remote analysis applications due to its small size. For the smart GUI application, both the real time display and plot of the temperature reading output can be viewed by the user.

To acquire the data from this sensor, a LM35 library for Arduino is needed. The *#include <lm35.h>* is implied in the Arduino source code to include the library header file in the sketch. It is also necessary to include a parameter specifying the Arduino pin connected to the sensor as *lm35 temps(0),* where the number zero between the bracket corresponds to the pin A0. A more detailed LM35 source line as shown in Command line 2 below is situated in the *loop ( )* function.

Command line 2 – LM35 output reading

```
if (Serial.available()>0){
    message =Serial.read();
    if (message=='1'){
    Start=true;
            }
    else
            {
    Start=false;
            }
        }

    if (Start==true){
    temps.MeasureTemp();
    T=temps.TempInCelcius;
    Serial.println(T);
    delay(30000);
            }
```

To plot the real-time temperature sensor data, the JFreeChart library was used. This library was implemented by adding the *jfreechart-1.0.14.jar* and the *jcomm-1.0.17.jar* files to the NetBeans Project's classpath.

The XY line chart was chosen to represent the dataset acquired where the method provided by the *ChartFactory* class is *createXYLineChart(title, categoryAxisLabel, valueAxisLabel, dataset, orientation, legend, tooltips, urls)* as shown below:

JFreeChart Graf=ChartFactory.createXYLineChart("Temperature vs. Time", "Time (second)", "Temperature (Celcius)", Koleksi, PlotOrientation.VERTICAL, true, true, false);

d) Data storage (*.xlsx file)

The smart GUI was programmed to save the data acquired in *.xlsx file format using the POI library. The basic class used to read and write this file format is HSSFWorkbook (Horrible SpreadSheet Format) that act as an Excel Workbook. The excerpt from the NetBeans source code is as shown in command line 3.

Command line 3 – Creating and writing data in *.xlsx file
HSSFWorkbook book = new HSSFWorkbook();
HSSFSheet sheet = book.createSheet();
HSSFRow row = sheet.createRow(0);
HSSFCell cell = row.createCell(0);
cell.setCellValue("Data From Temperature Sensor LM35");

row = sheet.createRow(1);   //Create a new row in current sheet
cell = row.createCell(0);   //Create cell for HOUR in current row
cell.setCellValue("HOUR"); //Set value to new value
cell = row.createCell(1);   //Create cell for TEMP in current row
cell.setCellValue("TEMPERATURE");   //Set value to new value

By pressing the "Export to Excel" button on the interface, a modal dialog is displayed and the user can freely select where to save the *.xlsx file. This option is implemented by executing the JFileChooser application program interface (API). The constructor chosen for this work was *JFileChooser ( )* which is an empty constructor that points to user's default directory. All data stored in *.xlsx file can later be opened and plotted in spreadsheet programs (Microsoft Excel) once retrieved.

The final form of the smart GUI with the function of each feature is shown in Fig. 4 with its user guide explained in the PWM modification section. Table 1 shows the summary of all the open source libraries used along with the NetBeans and Arduino IDE.
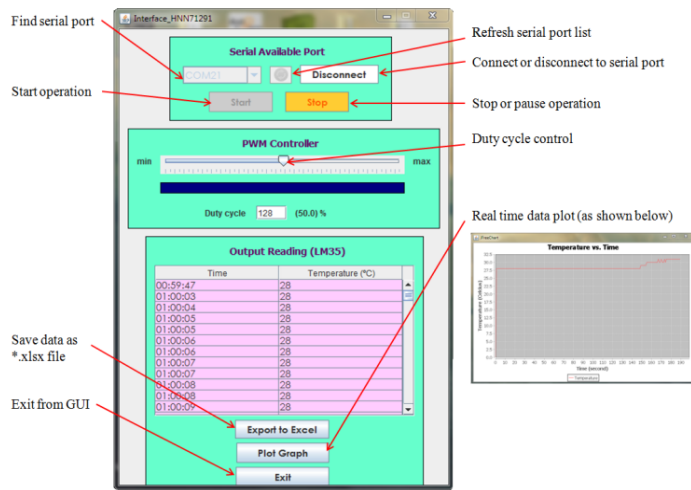

Fig. 4 Design of smart GUI for THP controller system.

Table 1 Summary of smart GUI dependencies: all open source.

Library	Version	Source
RXTX	2.2	users.frii.com/jarvi/rxtx/
LM35	1.0.1	arduinolibraries.info/libraries/lm35
JFreeChart	1.0.14	jfree.org/jfreechart/download.html
JCommon	1.0.17	jfree.org/jcommon/
Apache POI	3.9	poi.apache.org/

Development of THP controller circuit

The THP circuit consisted of a TEC (Peltier module TEC1-12715), PSU S-360-12, and MOSFET4 UO4 which was constructed as shown in Fig. 5. The power supply with DC output of 0-12 V and 30 A was chosen due to its compatibility with Peltier device (V_{max} 15.4 V and I_{max} 15 A) and MOSFET4 UO4 (max. output power 100 V/30 A). The surface area of the Peltier TEC plate (40 x 40 x 3.3 mm) used in this work is sufficient to cover the bottom surface of the sample holder made up of aluminum foil, enabling rapid heat transfer between the Peltier and sample. Heat dissipation of the Peltier TEC was managed using an aluminum heat sink with two mounted cooling fans (supplied with PSU 12 V/3 A) and heat sink compound used as an interface between the Peltier device and the heat sink.
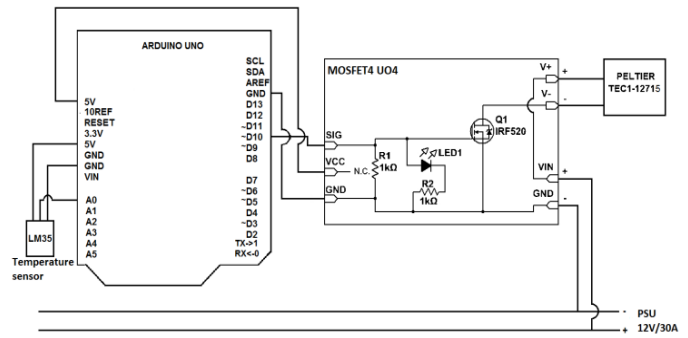

Fig. 5 Schematic diagram of THP and temperature sensor circuit.

The four channels MOSFET4 UO4 worked as a switch due to its excellent role in being a switching device to power high-powered devices such as TEC with PWM. The MOSFET's positive (+) and negative (-) terminals were connected to the 12 V power, while its channels (CH+ and CH -) were connected to the Peltier TEC's positive and negative wires, respectively. The connection to the Arduino microcontroller to turn the channels on and off is through the 3-pin connectors, namely ground (-), voltage (+), and signal (S). Therefore, the MOSFET's ground, voltage, and signal pins were connected to Arduino's GND, 5V, and ~D10 pins. The MOSFET was also equipped with a heat sink to ensure that it does not heat up.

To enable the communication between the LM35 temperature sensor and the microcontroller board, three sensor pins were connected to the 5V, A0, and GND of the Arduino board. Through these pins, the microcontroller board manages the temperature sensor and retrieves the temperature readings.

CALIBRATION

The performance of the smart GUI (PWM modification and temperature sensor) was tested before the whole system can be validated for future experimental application in solidifying various types of samples.

PWM modification

To ensure the validity of the smart GUI in changing the duty cycle and the output power signal, a calibration process was done by changing the duty cycle slider on the GUI with an increment of 10 from 10-100%. An oscilloscope was used to visualize the electrical signal output of 5V supplied by Arduino UNO that varies with time. Fig. 6 shows the example of the displayed signal with the duty cycle of 60%. From Fig. 6, the pulse width (t_p), period (T), amplitude, average output voltage, and others were also acquired. The input duty cycle demonstrates comparable results with the output value as shown by the oscilloscope, indicating that the smart GUI successfully controls the duty cycle accordingly
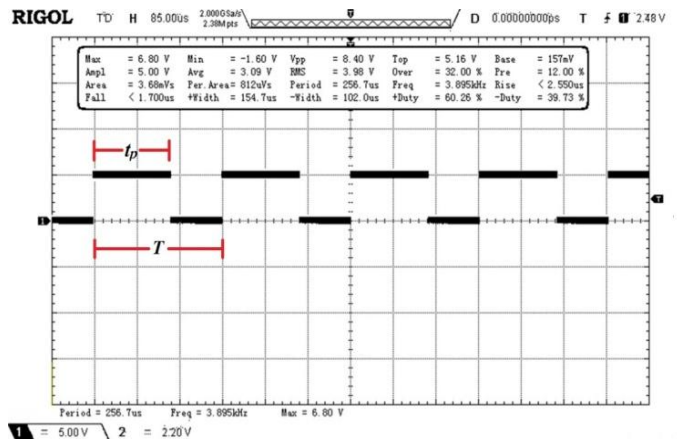

Fig. 6 Duty cycle shown by oscilloscope during calibration.

Temperature sensor

To check the accuracy of the temperature reading displayed by the GUI, temperature readings of a glass of iced water left at room temperature of 29 ˚C were acquired by LM35 and a standard mercury thermometer. Fig. 7 shows the comparison plot of both temperature readings from LM35 and the thermometer. LM35 managed to record almost the same reading as the thermometer with only small differences. The R-squared values on the graph are closer to 1.0 proving the better the fit of the regression line. The linear fit gradient acquired from LM35 reading of the melting ice described a temperature rise of 0.261 ˚C per minute.
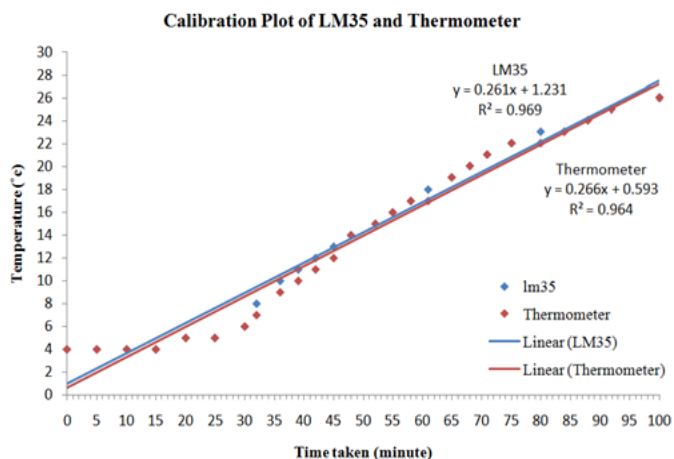

Fig. 7 Calibration plot of LM35 compared with standard mercury thermometer.

RESULTS AND DISCUSSIONS

Further experiment was carried to test the performance of this system in solidifying sample. 1.5 ml distilled water with fix initial temperature of 23 ˚C was placed inside an aluminum sample holder on top of the Peltier device for freezing purpose.

The experiment was done by supplying the circuit with PSU S-360-12. The COM port where the Arduino was connected to the computer was selected from the dialog box under the Serial Available Port section of the GUI (see Fig. 4). The "Connect/Disconnect" button followed by the "Start" button was pressed to confirm the communication between Arduino and the computer. The PWM controller slider was used to vary the PWM (increment of 10 from 10%-100%). Next, the temperature reading measurements of the sample as collected by LM35 sensor was displayed by the GUI and later save as *.xlsx file. Fig. 8 shows the solidified sample maintained in its freezing phase.
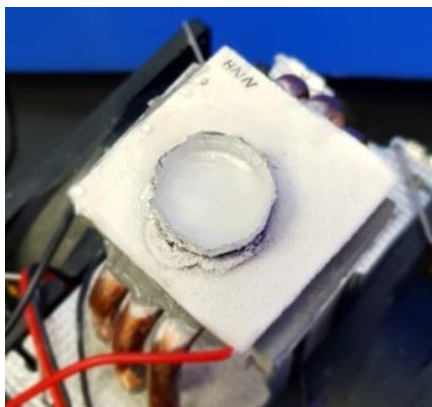

Fig. 8 Peltier device with sample in solidified phase.

From the calibration, we acquired several parameters, namely the lowest temperature of the solidified samples, operating voltage, power and current supplied by PSU, time taken for the samples to solidify to the lowest temperature, and time duration of the sample to stay at lowest temperature. The current of the Peltier element (PWM output current), I_{avg} can be calculated using Equation (3). Table 2 shows the data tabulated from the measurements acquired.

Table 2 Temperature, time taken, t_p, operating voltage, current and power and I_{avg} of each duty cycle.

Duty cycle (%)	Operating voltage (V)	Operating current (A)	Operating power (W)	t_p (µs)	I_{avg} (A)	Lowest temperature (˚C)	Time taken to acquire stable lowest temperature (minute)	Time duration to remain at the lowest temperature (minute)
10	12	15	18	25.67	1.5	18	16.5	3
20	12	15	36	51.34	3.0	16	8	7
30	12	15	54	77.01	4.5	9	10	3
40	12	15	72	102.68	6.0	7	9	2
50	12	15	90	128.35	7.5	0	28	6
60	12	15	108	154.02	9.0	0	26.5	17.5
70	12	15	126	179.69	10.5	0	19.5	20[1]
80	12	15	144	205.36	12.0	0	18.5	20[1]
90	12	15	162	231.03	13.5	0	7	20[1]
100	12	15	180	256.7	15	0	4	20[1]

Sample temperature stayed at 0 ˚C and did not increase until the power supply was switched off.

The following sections explain the relationships between the duty cycle and the parameters as mentioned in Table 2, including: duty cycle and Peltier TEC's operating power (Section 4.2.1); cooling plot of different duty cycle (Section 4.2.2); duty cycle and time taken to acquire lowest sample temperature (Section 4.2.3); and duty cycle and time duration in maintaining temperature stability (Section 4.2.4).

Duty cycle and Peltier TEC's operating power

Fig. 9 shows the relationship between the operating power required by the Peltier TEC and the duty cycle. The results depict that the operation power varies linearly with duty cycle. As the duty cycle increase, the system is increasingly capable of lowering the sample temperature, thereby raising the power consumption of the TEC to keep on transferring heat from one side of the Peltier device to the other for cooling purpose. The gradient from the linear fit demonstrated a power increase of 1.8 W each time the duty cycle is raised by 1 %.
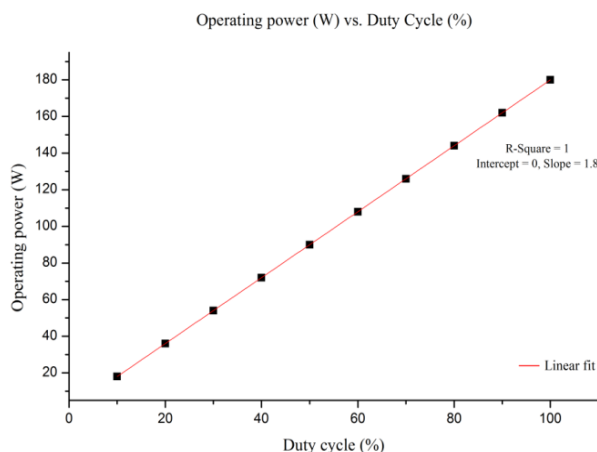

Fig. 9 Peltier TEC's operating power vs. duty cycle.

Cooling plot for different duty cycles

Since the THP controller system was capable of solidifying the sample only when the duty cycles value was between 50–100 % (see Table 2), further discussion is focused on these specific duty cycles. The cooling curves of the sample under data acquisition time limitation of 630 seconds are plotted to show clearer relationship between the parameters, as shown in Fig. 10. The sample's temperature drop drastically starting from the beginning until the 150th second into data acquisition. The temperature then continued to decrease, and started to stabilize from 325 seconds onward. The fit curves show that if the duty cycle is higher, the sample coagulation progress is better as it successfully reaches low temperature even faster.
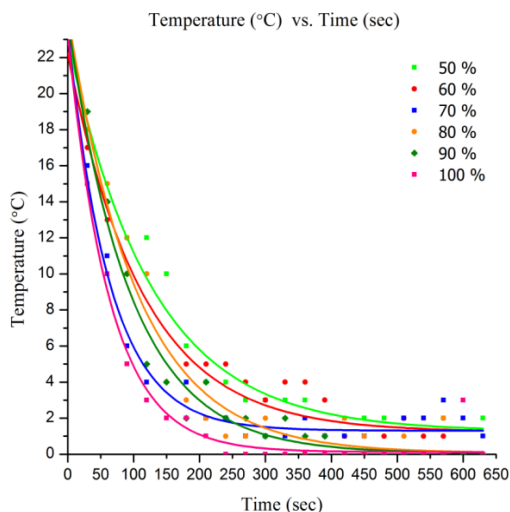

Fig. 10 Cooling plot of the sample under various duty cycles.

Duty cycle and freezing time

Fig. 11 summarizes the relation between the duty cycle and the time required by the sample to freeze. It is observed that with higher duty cycle, lower time is required for the sample to freeze. For instance, it took less than 10 minutes to freeze the sample when the duty cycle was in the range of 90 % and above. The gradient from the linear fit also demonstrated a decrement of 0.51286 minute each time the duty cycle is raised by 1 %. It is implied that higher duty cycle is a better selection because of its ability to freeze samples in a shorter time.
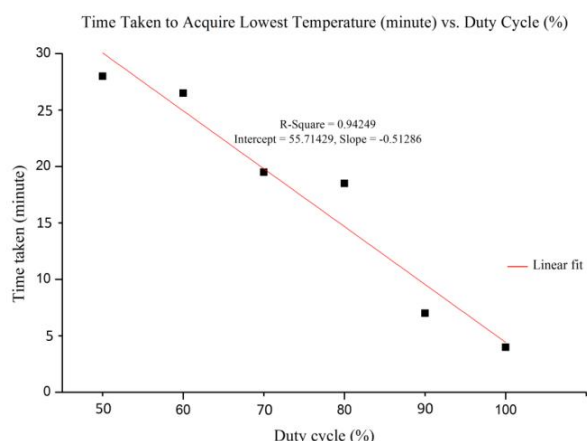

Fig. 11 Relation between duty cycle and time taken to acquire the lowest sample temperature.

Duty cycle and temperature stability

The graph of the time duration to maintain temperature stability versus duty cycle is as shown in Fig. 12. It shows that higher duty cycle presented better THP controller system performance in solidifying the sample. Lower duty cycle (below 70 %) failed to maintain the sample's lowest temperature due to insufficient power provided to the Peltier

TEC. For the duty cycle of 70 % and above, the sample remained at the same low temperature until the end of the 20 minutes experiment. Hence, higher duty cycle is once again proven to be more preferable due to its ability to maintain the temperature stability.
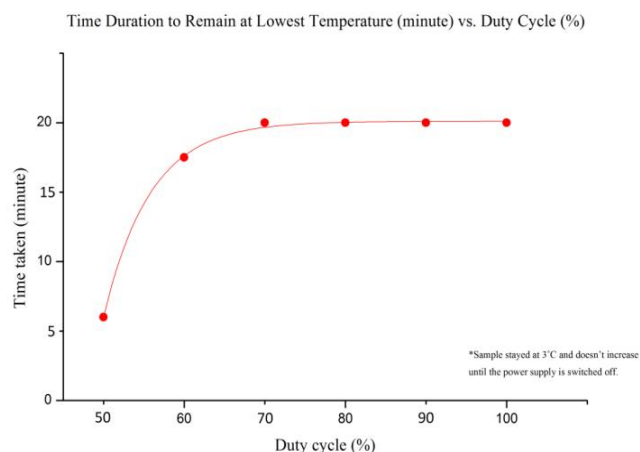

Fig. 12 Relation between duty cycle and time duration of the sample to remain at lowest temperature.

CONCLUSION

In this paper, a low cost, user-friendly, easy-to-assemble, and open source THP controller system for solidifying LIBS sample is presented. This system is capable of replacing other sample preparation method of solidifying sample such as the conventional refrigerator or liquid nitrogen method. The present study shows that a high duty cycle will increase the current of the Peltier TEC (I_{avg}). This will lower the minimum reachable temperature and the time taken to reach the minimum temperature.

Duty cycle in the range of 70–100 % seems to be ideal for freezing whereas 100 % duty cycle provide the fastest freezing time. However, lower duty cycle could also be chosen by changing the power consumption and time taken for certain preferable sample to freeze. In the future, it is also possible to apply the system to any viscous liquids with varied viscosity as their analysis may be required in LIBS applications.

REFERENCES

[1] Musazzi, S., & Perini, U. (2014). *Laser-induced breakdown spectroscopy: Theory and applications* (Vol. 1). Berlin: Springer-Verlag.

[2] Winefordner, J. D., Gornushkin, I. B., Correll, T., Gibb, E., Smith, B. W., & Omenetto, N. (2004). Comparing several atomic spectrometric methods to the super stars: Special emphasis on laser induced breakdown spectrometry, LIBS, a future super star. *Journal of Analytical Atomic Spectrometry*, *19*(9), 1061-1083.

[3] Papai, R., Sato, R. H., Nunes, L. C., Krug, F. J., & Gaubeur, I. (2017). Melted paraffin wax as an innovative liquid and solid extractant for elemental analysis by laser-induced breakdown spectroscopy. *Analytical Chemistry*, *89*(5), 2807-2815.

[4] Zhao, Y., Zhang, L., Zhao, S. X., Li, Y. F., Gong, Y., Dong, L., Ma, W. G., Yin, W. B., Yao, S. C., Lu, J. D., Xiao, L. T., & Jia, S. T. (2016). Review of methodological and experimental LIBS techniques for coal analysis and their application in power plants in China. *Frontiers of Physics*, *11*(6), 114211.

[5] Syvilay, D., Wilkie-Chancellier, N., Trichereau, B., Texier, A., Martinez, L., Serfaty, S., & Detalle, V. (2015). Evaluation of the standard normal variate method for laser-induced breakdown spectroscopy data treatment applied to the discrimination of painting layers. *Spectrochimica Acta Part B-Atomic Spectroscopy*, *114*, 38-45.

[6] Peng, J. Y., Liu, F., Zhou, F., Song, K. L., Zhang, C., Ye, L. H., & He, Y. (2016). Challenging applications for multi-element analysis by laser-induced breakdown spectroscopy in agriculture: A review. *Trac-Trends in Analytical Chemistry*, *85*, 260-272.

[7] Harun, H. A., Zainal, R., & Daud, Y. M. (2017). Analysing human nails composition by using laser induced breakdown spectroscopy. *Sains Malaysiana*, *46*(1), 75-82.

[8] Jantzi, S. C., Motto-Ros, V., Trichard, F., Markushin, Y., Melikechi, N., & De Giacomo, A. (2016). Sample treatment and preparation for laser-induced breakdown spectroscopy. S*pectrochimica Acta Part B: Atomic Spectroscopy*, *115*, 52-63.

[9] St-Onge, L., Kwong, E., Sabsabi, M., & Vadas, E. B. (2004). Rapid analysis of liquid formulations containing sodium chloride using laser-induced breakdown spectroscopy. *Journal of pharmaceutical and Biomedical Analysis*, *36(*2), 277-284.

[10] Lee, D.-H., Han, S.-C., Kim, T.-H., & Yun, J.-I. (2011). Highly sensitive analysis of boron and lithium in aqueous solution using dual-pulse laser-induced breakdown spectroscopy. *Analytical Chemistry*, *83*(24), 9456-9461.

[11] Rai, N. K., & Rai, A. (2008). LIBS—an efficient approach for the determination of Cr in industrial wastewater. *Journal of Hazardous Materials*, *150*(3), 835-838.

[12] Aras, N., Yeşiller, S. Ü., Ateş, D. A., & Yalçın, Ş. (2012). Ultrasonic nebulization-sample introduction system for quantitative analysis of liquid samples by laser-induced breakdown spectroscopy. *Spectrochimica Acta Part B: Atomic Spectroscopy*, *74-75*, 87-94.

[13] Zhong, S.-L., Lu, Y., Kong, W.-J., Cheng, K., & Zheng, R. (2016). Quantitative analysis of lead in aqueous solutions by ultrasonic nebulizer assisted laser induced breakdown spectroscopy. *Frontiers of Physics*, *11*, 1-9.

[14] Cahoon, E. M., & Almirall, J. R. (2012). Quantitative analysis of liquids from aerosols and microdrops using laser induced breakdown spectroscopy. *Analytical chemistry*, *84*(5), 2239-2244.

[15] Godwal, Y., Kaigala, G., Hoang, V., Lui, S.-L., Backhouse, C., Tsui, Y., & Fedosejevs, R. (2008). Elemental analysis using micro laser-induced breakdown spectroscopy (μLIBS) in a microfluidic platform. *Optics Express*, *16*(17), 12435-12445.

[16] Groh, S., Diwakar, P., Garcia, C., Murtazin, A., Hahn, D., & Niemax, K. (2010). 100% efficient sub-nanoliter sample introduction in laser-induced breakdown spectroscopy and inductively coupled plasma spectrometry: implications for ultralow sample volumes. *Analytical Chemistry*, *82*(6), 2568-2573.

[17] Motto-Ros, V. (2015). Characteristics of indirect laser-induced plasma from a thin film of oil on a metallic substrate. *Frontiers of Physics*, *10*(2), 231-239.

[18] Xiu, J., Bai, X., Negre, E., Motto-Ros, V., & Yu, J. (2013). Indirect laser-induced breakdown of transparent thin gel layer for sensitive trace element detection. *Applied Physics Letters*, *102*(24), 244101.

[19] Cáceres, J., López, J. T., Telle, H., & Ureña, A. G. (2001). Quantitative analysis of trace metal ions in ice using laser-induced breakdown spectroscopy. *Spectrochimica Acta Part B: Atomic Spectroscopy*, *56*(6), 831-838.

[20] Sobral, H., Sanginés, R., & Trujillo-Vázquez, A. (2012). Detection of trace elements in ice and water by laser-induced breakdown spectroscopy. *Spectrochimica Acta Part B: Atomic Spectroscopy*, *78*, 62-66.

[21] Jantzi, S. C., & Almirall, J. R. (2011). Characterization and forensic analysis of soil samples using laser-induced breakdown spectroscopy (LIBS). *Analytical and Bioanalytical Chemistry*, *400*(10), 3341-3351.

[22] Lazic, V., Colao, F., Fantoni, R., Spizzichino, V., & Jovićević, S. (2007). Underwater sediment analyses by laser induced breakdown spectroscopy and calibration procedure for fluctuating plasma parameters. *Spectrochimica Acta Part B: Atomic Spectroscopy*, *62*(1), 30-39.

[23] El-Hussein, A., Kassem, A., Ismail, H., & Harith, M. (2010). Exploiting LIBS as a spectrochemical analytical technique in diagnosis of some types of human malignancies. *Talanta*, *82*(2), 495-501.

[24] Harun, H. A., & Zainal, R. (2018). Improvement of laser induced breakdown spectroscopy signal for sodium chloride solution. *Malaysian Journal of Fundamental and Applied Sciences*, *14*, 429-433.

[25] Wishkerman, A., & Wishkerman, E. (2017). Application note: A novel low-cost open-source LED system for microalgae cultivation. *Computers and Electronics in Agriculture*, *132*, 56-62.

[26] Tedeschi, A., Calcaterra, S., & Benedetto, F. (2017). Ultrasonic Radar System (URAS): Arduino and virtual reality for a light-free mapping of indoor environments. *Ieee Sensors Journal*, *17*(14), 4595-4604.

[27] Ali, A. S., Zanzinger, Z., Debose, D., & Stephens, B. (2016). Open Source Building Science Sensors (OSBSS): A low-cost Arduino-based platform for long-term indoor environmental data collection. *Building and Environment*, *100*, 114-126.

[28] Barroca, N., Borges, L. M., Velez, F. J., Monteiro, F., Gorski, M., & Castro-Gomes, J. (2013). Wireless sensor networks for temperature and humidity monitoring within concrete structures. *Construction and Building Materials*, *40*, 1156-1166.

[29] Hossain, M. A., Canning, J., Yu, Z. K., Ast, S., Rutledge, P. J., Wong, J. K. H., Jamalipour, A., & Crossley, M. J. (2017). Time-resolved and temperature tuneable measurements of fluorescent intensity using a smartphone fluorimeter. *Analyst*, *142*(11), 1953-1961.

[30] Purdum, J. J., & Levy, B. (2012). *Beginning C for Arduino*. New York: Springer.

[31] Nemati, A., Nami, H., Yari, M., Ranjbar, F., & Kolvir, H. R. (2016). Development of an exergoeconomic model for analysis and multi-objective optimization of a thermoelectric heat pump. *Energy Conversion and Management*, *130*, 1-13.

[32] Holmes, D. G., & Lipo, T. A. (2003). *Pulse width modulation for power converters: principles and practice* (Vol. 18). Hoboken, NJ: John Wiley & Sons.

[33] Recktenwald, G. (2011). Basic pulse width modulation. Retrieved from https://docplayer.net/21283522-Basic-pulse-width-modulation.html