Fluidic Control System Development for Fluorescence Imaging Photometer

Ergün Kara

Follow this and additional works at: https://scholarcommons.sc.edu/etd

Part of the Chemistry Commons

Recommended Citation

This Open Access Thesis is brought to you by Scholar Commons. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact dillarda@mailbox.sc.edu.
FLUIDIC CONTROL SYSTEM DEVELOPMENT FOR FLUORESCENCE IMAGING PHOTOMETER

by

Ergün Kara

Bachelor of Science
Middle East Technical University, 2013

Submitted in Partial Fulfillment of the Requirements
For the Degree of Master of Science in
Chemistry
College of Arts and Sciences
University of South Carolina
2019

Accepted by:
Michael L. Myrick, Director of Thesis
John L. Ferry, Reader
Cheryl L. Addy, Vice Provost and Dean of the Graduate School
ABSTRACT

Phytoplankton are the primary autotrophic organisms in the marine food web and they produce nearly half of the global oxygen as a photosynthetic by-product [1]. Through photosynthesis, phytoplankton annually fix between 30 and 50 billion metric tons of atmospheric carbon into the marine systems, which approximately accounts for 40% of total carbon fixation [2]. Furthermore, the phytoplankton population trends are important indicators of the environmental perturbations [3]. Therefore, detection and characterization of the phytoplankton species is an important goal of the science. Several spectrometric, cytometric and microscopic techniques are already employed for characterization and quantification of the phytoplankton communities [4-6]. However, all these existing techniques have disadvantages, such as being not suitable for the on-board research vessel applications or being significantly slower than the common fluorometric and spectrometric methods [5,6]. A fluorescence imaging photometer (FIP) system to collect the fluorescence images of the single cells on a flowing stream was previously developed in the research group of Dr. Michael L. Myrick at University of South Carolina [7], and it was employed for characterization of the phytoplankton species based upon the interspecific differences in their fluorescence emission intensities at some specific wavelength regions. Nevertheless, it is seen that the data collected by this FIP system contain significant level of uncertainty.
This uncertainty is attributed to the instability of the instrument and the variance in the pigment profiles of the cells of the same species [7]. It was not possible to take multiple fluorescence images of the flowing cells for improving the signal-to-noise ratio with this FIP system. Therefore, a fluidic control unit for locating the cells within the instrument and trapping them for multiple measurements is developed in this study. The previous 2D flow cell is also replaced with a quartz capillary tube in order to reduce the number of the optical components in the system, which is expected to result in an improved signal-to-noise ratio.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF ABBREVIATIONS</td>
<td>vii</td>
</tr>
<tr>
<td>CHAPTER 1: FLUIDIC CONTROL SYSTEM FOR FLUORESCENCE IMAGING PHOTOMETER (FIP)</td>
<td>1</td>
</tr>
<tr>
<td>1.1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.2 THE PREVIOUS FIP SYSTEM</td>
<td>2</td>
</tr>
<tr>
<td>1.3 THEORETICAL CALCULATIONS FOR SELECTING THE OPTICAL BANDPASS FILTERS</td>
<td>5</td>
</tr>
<tr>
<td>1.4 ANALYSIS OF THE DATA COLLECTED BY THE PREVIOUS FIP SYSTEM</td>
<td>8</td>
</tr>
<tr>
<td>1.5 THE RESULTANT SYSTEM AND DISCUSSIONS</td>
<td>10</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>23</td>
</tr>
<tr>
<td>APPENDIX A: ARDUINO IDE CODE FOR CIRCUIT 1</td>
<td>25</td>
</tr>
<tr>
<td>APPENDIX B: ARDUINO IDE CODE FOR CIRCUIT 2</td>
<td>29</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 1.1 The Fluorescence Imaging Photometer (FIP).........................................................3
Figure 1.2 The filter wheel of FIP ............................................................................................4
Figure 1.3 A fluorescence image example..................................................................................4
Figure 1.4 The clustering analysis for the FIP data of 9 calibration species .........................6
Figure 1.5 Schematic representation of fluidic control system.................................................14
Figure 1.6 A photograph of the mechanical part of the fluidic control system .................15
Figure 1.7 Circuit 1 for the computerized fluidic control of the trapped cells ..................16
Figure 1.8 A photograph of Circuit 1......................................................................................17
Figure 1.9 Circuit 2 for the control of the pinch valves..........................................................18
Figure 1.10 A photograph of the combined fluidic control system.......................................19
Figure 1.11 A bead trapped at the edge of the flow cell.........................................................20
Figure 1.12 The trapped bead shown in Figure 1.11 is moved downward in the flow cell by the fluidic control system..................................................................................21
Figure 1.13 The trapped bead shown in Figure 1.12 is moved further downward in the flow cell by the fluidic control system..................................................................................22
LIST OF ABBREVIATIONS

Eq. .............................................................................................................................. Equation

FIP ............................................................................................................................ Fluorescence Imaging Photometer

FOM ............................................................................................................................. Figure of Merit
CHAPTER 1
FLUIDIC CONTROL SYSTEM FOR FLUORESCENCE IMAGING PHOTOMETER (FIP)

1.1 INTRODUCTION

American Chemical Society states that “The Earth’s climate is changing due to elevated concentrations of greenhouse gases and particulate matter in the atmosphere, as a result of human activities”, referring to the global warming [8]. Therefore, number of research studies conducted for understanding the earth’s climate system and developing high-tech solutions to fix the atmospheric carbon has significantly increased in the past decades. The phytoplankton play a key role in the carbon cycle being responsible for some 40% of total global carbon fixation [1]. Promoting the phytoplankton biomass by increasing the nutritional availability is a suggested solution for the increased atmospheric carbon levels. However, some phytoplankton species can produce toxins and grow out of control, resulting in harmful algal blooms [3]. Thus, monitoring the population dynamics of them and characterization of the phytoplankton species has long been desired by the scientists.

There are several methods applied for characterization of the phytoplankton communities and estimating the biomass of them. Direct counting with light microscope is a highly accurate method, but it requires specialized workforce. Remote sensing by satellite imaging is commonly used for biomass estimation and monitoring the population dynamics of phytoplankton [9,10]. However, the discriminative power of the satellite-
based remote sensing techniques is limited, therefore, they are not useful for taxonomic characterization of the phytoplankton communities. Flow cytometry is commonly applied for counting the phytoplankton cells, and it is regarded as an accurate and fast method for that [5,11], but the flow cytometry is not successful at classifying species. All phytoplankton species are known to contain chlorophyll A, and therefore, spectroscopic and fluorometric chlorophyll A sensors has been widely employed to detect and quantify them in the sample waters [12]. Nevertheless, chlorophyll A concentration is not sufficient to discriminate between phytoplankton species alone. Phytoplankton species also contain several accessory pigments that work in conjunction with chlorophyll A and transfer the absorbed photonic energy to the photosystem chlorophyll structures. The relative concentrations of accessory pigments, such as phycocyanin and phycoerythrin, are mainly taxa-specific and that can be utilized to determine the taxonomic classes of the species. In other words, the phytoplankton species can be characterized based upon the pigment compositions of them. Several HPLC-based methods have already been successfully employed to measure the relative concentrations of different pigments in the phytoplankton culture samples and the characterization of the species is performed based upon their pigment profiles [13]. HPLC-based methods, yet, require solvent extraction of the pigments from the phytoplankton samples. Therefore, they are considered to be tedious and slow. An accurate, fast and portable system for phytoplankton characterization is still demanded by the scientific instrument markets. The instrument is expected to be energy efficient and suitable for the research vessel missions.

1.2 THE PREVIOUS FIP SYSTEM

A fluorescence imaging photometer (FIP) for automated taxonomic classification of the phytoplankton was already developed in our research group, Dr. Michael L.
Myrick’s group at University of South Carolina [7,14]. This FIP system utilizes interspecific differences in the fluorescence excitation spectra of the phytoplankton species for the characterization of them. The phytoplankton sample solution is injected through a 2D flow cell, and the fluorescence images of the individual cells are taken by a CCD array camera. The polychromatic incident light is modulated by a set of specifically selected optical bandpass filters, which are placed on a filter wheel in front of the light source. Figure 1.1 shows a simplified scheme of the FIP system, and Figure 1.2 shows the filter wheel.

Figure 1.1. The Fluorescence Imaging Photometer (FIP)
The overall fluorescence intensity under each filter is monitored by the intensities of the corresponding streaks in the resultant fluorescence images. By modulating the angular speed of the filter wheel rotation and the flow rate of the sample solution, the

Figure 1.2. The filter wheel.

Figure 1.3. Fluorescence image example.
fluorescence images of single phytoplankton cells can be collected with satisfactory resolution in this system. An example fluorescence image can be seen in Figure 1.3.

Phytoplankton species are characterized by the ratio of integrated streak intensities of different filters, acquired from the fluorescence images by a MATLAB® script [7]. In the example fluorescence image, Figure 1.3, there are 2 binary filter sets and a reference neutral density filter employed. Therefore, a total of 5 streaks are seen. Let’s call the filters in the first binary filter set as A and B. The following ratio is calculated using the fluorescence streak intensity by filter A ($I_A$) and the fluorescence intensity by filter B ($I_B$):

$$m_1 = \frac{I_A - I_B}{I_A + I_B} \quad \text{(Eq.1)}$$

‘$m_1$’ is called ‘Filter Set 1 Ratio’ in our notation. Likewise, we can calculate an ‘$m_2$’ using the fluorescence intensities for the remaining two streaks that are generated by the filters of second binary set, arbitrarily named as C and D. In order to perform the taxonomic classification of the phytoplankton species, $m_1$ and $m_2$ values are calculated for each analyzed cell and a clustering analysis is applied by visualizing the distribution of these ratios in a Cartesian plane. Cells of the same species are expected to cluster together. An example clustering analysis for 9 species can be seen in Figure 1.4.

1.3 THEORETICAL CALCULATIONS FOR SELECTING THE OPTICAL BANDPASS FILTERS

The optical bandpass filters that are mounted on the filter wheel of the FIP system are selected to maximize the inter-taxa differences in the integrated fluorescence intensities of the phytoplankton species. A theoretical method for selecting the optical
Figure 1.4: The clustering analysis for the FIP data of 9 calibration species [7]. These species are *D. tertiolecta, E. huxleyi, A. carterae, P. tricornutum, T. pseudonana, G. theta, R. salina, P. sulcata* and *C. ovata*. 
bandpass filters among the commercially available candidates was developed and a MATLAB® script was written for this method in our research group [7]. A calibration set of phytoplankton species are randomly selected among all available phytoplankton cultures for choosing the filters. The fluorescence excitation spectra of these calibration set species are measured and saved as a data matrix in MATLAB®. Similarly, the spectral profile of the excitation source, the transmission profiles of the optical bandpass filters and the transmission profiles of all optical elements in the FIP system are saved as data matrices in MATLAB®. 201 commercially available bandpass filters in the desired wavelength range (400-630 nm) are listed, and a list of all possible filter pairs is generated into a MATLAB® array in the reference study [7]. All binary pairs of species that are being tested are listed in another MATLAB® array. For each filter, consecutive matrix multiplications and spectral integrations are applied to estimate the fluorescence responses (I) of species theoretically. In other words, the integrated fluorescence intensity of each species under the filter of interest is estimated by multiplying the irradiance profile matrix of the excitation source by the transmission profile matrices of all optical elements in the FIP system, then multiplying the resultant matrix by the fluorescence spectra matrix of the species. This is followed by the summation of the intensity values in all rows (channels), which is the integration part of the MATLAB® algorithm [7]. Figure of Merit (FOM) values that quantify the separability of the binary phytoplankton pairs by the filter pairs of interest are calculated for all available combinations. For each pair of species, the filter sets of interest generate two ‘m’ values, ‘m₁’ and ‘m₂’ respectively, calculated by Eq. 1.
These $m_1$ and $m_2$ ratios, along with the estimated variance of these species-specific ratios ($V_{m_1}$ and $V_{m_2}$), are used to calculate a FOM:

$$FOM = \frac{(m_1 - m_2)^2}{V_{m_1} + V_{m_2}}$$  \hspace{1cm} (Eq.2)$$

By the history of single-cell measurements in our experience, it is seen that the variation in the ratio ‘$m$’ is independent of the ratio itself [7]. Our research group approximated the standard deviation of ‘$m$’ to be 0.05 (corresponding to a variance of 0.0025 in Eq. 2) for all species. All binary pairs with a FOM greater than 4 are accepted to be separable by the filter pair of interest. Therefore, the filter pairs with the highest discriminative power are selected by this method.

1.4 ANALYSIS OF THE DATA COLLECTED BY THE PREVIOUS FIP SYSTEM

A correlation between the number of the filter sets necessary for the characterization of the phytoplankton species and the number of physical factors that affect the fluorescence spectra of them is expected. In fact, the concentration of each pigment available in these species is expected to be a physical factor affecting the fluorescence excitation spectrum. Therefore, effective rank of the fluorescence excitation spectra matrix of the species is also investigated by Principal Component Analysis (PCA). The fluorescence excitation spectra of the known pigments are expected to be the Principal Components (PCs) of the combined fluorescence excitation spectra matrix of the species.

The dataset is expected to be high in rank considering the variety of pigments that shape the fluorescence spectra. Rekully et al. [7] demonstrate a list of methods for testing the effective rank and apply all these methods on the fluorescence excitation spectra dataset. By the Broken Stick method, the effective rank is calculated to be 2, for
95% of the total variance. By Kaiser-Guttman method, in which the PCs with a score above the average score are accepted to be significant, the effective rank of the data matrix is calculated as 3. Regarding these test results, the effective rank of the dataset is concluded to be around 2-3 [7]. That was confirmed by the experimental findings since the number of filter sets that are sufficient to discriminate all available species is found to be 2. These results indicate a possible covariance between different pigment concentrations in the phytoplankton cells, since the effective rank is significantly smaller than the number of the known pigments in these species. The tilted shape of the clusters also hints a covariance in the relative concentrations of some certain pigments in the analyzed phytoplankton species. A significant standard deviation in the filter set ratios is seen in Figure 1.4. A variance in the relative concentrations of the pigments in the cells of the same species is a possible explanation for this low reproducibility [15].

Another potential source of this standard deviation might be the noise associated with the camera detector. A National Instruments PIXIS 1024 CCD camera is employed in this FIP system. Pixel anomalies, dark noise and readout noise are common in CCD cameras [16]. In addition, the beam splitter is supposed to contribute to the low reproducibility due to its surface irregularities. The extent of the noise due to the variance in the pigmentation profiles of the cells of the same species could easily be determined by taking multiple pictures of the phytoplankton cells and reducing the population size of the calibration set. However, the FIP system, used in the reference study of our research group [7], is not able to trap the cells and take multiple fluorescence pictures of them. Our research group decided on developing a fluidic control system in order to locate and trap the analyzed cells in the system. Data averaging is planned to be performed by such
a fluidic control system. Furthermore, the CCD camera detector is planned to be decommissioned and replaced by a photomultiplier tube (PMT) detector due to the difficulties with synchronizing the CCD arrays with the time resolved systems and the potential pixel anomalies in CCD detectors. A significant improvement in the signal-to-noise ratio is expected from these modifications. Also, the 2D flow cell is replaced with a quartz tube and that eliminates the need for the beam splitter, a source of uncertainty in the measurements due to its surface irregularities.

1.5 THE RESULTANT SYSTEM AND DISCUSSIONS

In order for trapping the phytoplankton cells within the FIP system, taking multiple pictures of them and eventually disposing the cells out of the flow cell, a fluidic control system, as shown in Figure 1.5 and Figure 1.6, is built in this study. The fluidic system is desired to be fast, computer-controlled and continuous. The phytoplankton solution is introduced into this fluidic controller through the flexible NResearch TBGM100 silicone tubing with an internal diameter of 1/32 ” and a transparent flow cell is built by Polymicro TSPC300794/150 quartz tubing with an internal diameter of 300 µm. The maximum operational pressure of TBGM100 silicone tubing is around 10 psi, and the desired flow rate of the sample solution is around 1.5 mL/min. The flow cell, being narrow in diameter, is expected to be a high-pressure region. The pressure gradient at the flow cell should not exceed the working pressure of the main silicone tubing, and the length of the flow cell should be 5-10 cm for a proper monitoring. Therefore, the quartz tube for the flow cell is selected among the commercially available candidates with different internal diameters under these constraints. Hagen–Poiseuille Equation is used for estimating the fluidic pressure gradient at the flow cell:
\[ \Delta P = \frac{8 \cdot \mu \cdot L \cdot Q}{\pi \cdot R^4} \]  
(Eq.3)

\(\Delta P\) stands for the pressure difference between the two ends of the flow cell, \(\mu\) is the approximate viscosity of the phytoplankton culture solution, \(L\) is the length of the flow cell, and \(R\) stands for the radius of the quartz tube in Eq.3. For a 10 cm tube with a diameter of 200 \(\mu\)m, a seawater flow with a rate of 1.5 mL/minute creates a pressure gradient around 9 psi according to Eq.3. In order to guarantee a flow cell pressure below 10 psi, the diameter of the flow cell should be larger than 200 \(\mu\)m and Polymicro TSPC300794/150 is seen to be a satisfactory tube for building the flow cell.

The fluidic control system includes two NResearch 161P021-11 normally open pinch valves placed on each side of the flow cell to trap the cells as soon as they are detected in the flow cell. Two gear blanks, mounted on a metal shaft, are available in the fluidic control unit for squeezing the flexible silicon tubing and adjusting the location of the analyzed cell by displacing the fluid in the system. The rotational motion is provided by a HS-625MG servo motor, which is connected to the shaft that bears the gear blanks.

A special circuit, Circuit 2 in Figure 1.9, is built with an Arduino Uno R3 microcontroller, generic push buttons and a potentiometer. Circuit 2 is employed to computerize the control of the pinch valves as shown in Figure 1.9 and Figure 1.10. A delay time between the actuations of the NResearch 161P021-11 normally open pinch valves can be set by the potentiometer in this circuit. As shown in Figure 1.7 and Figure 1.8, another circuit, Circuit 1, is built to control the Hitec HS-625MG servo motor, using an Arduino Uno R3 microcontroller and three push buttons: LEFT, RIGHT and RESET. The LEFT button rotates the shaft in clockwise direction and therefore pushes the trapped cell upwards in the flow cell, whereas, the RIGHT button turns the shaft in
counterclockwise direction and pushes the trapped cell downwards in the flow cell. The RESET button brings the shaft to its original angular position at any time it is pressed.

The fluidic control system is manually tested and seen to be successful. An aqueous solution of Polysciences Fluoresbrite fluorescent beads with a diameter of 25 μm is prepared for the test. The solution is taken into a 45 mL plastic syringe and the syringe is placed onto New Era NE-300 pump. The main silicone tubing is attached to the syringe and the liquid is introduced into the fluidic control system by NE-300 pump. The flow cell is monitored by a Point Grey CMLNN2 USB camera mounted on an Meifi EMZ-8TR light microscope for the visual detection of the fluorescent beads in the flow cell in this test. The button in Circuit 2, shown in Figure 1.9, is pressed and the NResearch 161P021-11 normally open pinch valves are actuated as soon as the fluorescent bead is visually detected in the flow cell, trapping the bead immediately. The liquid flow is simultaneously diverted into a waste container by actuating the NResearch 225P091-11 two-tube pinch valve upon pushing the button of Circuit 2. Due to the hydraulic pressure, the trapped bead continues to move downwards in the flow cell after pressing the push button and closing the valves. The trapped bead can move out of vision or it can stop at locations that are inconvenient for the optical analysis. Figure 1.11 shows the location of cell after actuation of NResearch 161P021-11 normally open pinch valves in the test. By the LEFT and RIGHT buttons of the Circuit 1, the trapped cell can be brought to the center of the flow cell, as shown in Figure 1.12 and Figure 1.13.

The fluidic control unit described in this study is designated to be integrated into the FIP system developed in the research group of Dr. Michael Myrick at University of South Carolina [7,14], in the future. Although the fluidic control system is only tested
with 25 μm Fluoresbrite fluorescent bead solution, it is expected to locate the phytoplankton cells successfully. There are plenty of phytoplankton species with a cell diameter around 25 μm, such as Cryptomonas ovata [17], and Ditylum brightwellii [18]. Therefore, testing the system with 25 μm Fluoresbrite bead solution was a realistic and fast method to prove the operational capability of it.

Upon the integration of this fluidic control unit, the FIP instrument will be capable of taking several measurements on the captured cells. Therefore, the sources of variation in the filter set ratio values of the phytoplankton will be analyzed properly. The instrumental noise will be separated from the effect of the cell-to-cell variation in the pigment profile of the phytoplankton. Also, a significant noise reduction can be achieved by the signal averaging after the integration of this fluidic control unit to the FIP system. That is particularly important for the calibration calculations that determine the locations of the phytoplankton clusters on the Cartesian plane of filter set ratios (Figure 1.4). The fluidic control unit also renders the beam splitter unnecessary, reducing the number of optical elements in the system. The detector will directly be placed perpendicular to the direction of the incident light after the integration of this fluidic control unit to the FIP system. Please note the beam splitter is a potential source of noise due to its surface irregularities, and the elimination of it is expected to improve the signal-to-noise ratio.
Figure 1.5 The fluidic control system. 1) Hitec 32625S HS-625MG Hi-Speed Metal Gear 2BB Servo Motor. 2) NResearch 161P021-11 pinch valves. 3) Pic Design BP2-12 gear blanks, specifically machined for squeezing the silicon tube to locate the trapped cell by displacing the fluid. 4) The transparent flow cell, built with Polymicro TSPC300794/150 quartz tubing, heat shrink tubes and plastic ties. 5) New Era NE-300 Just Infusio Syringe Pump. 6) NResearch 225P091-11 two-tube pinch valve, employed to divert the liquid flow to a sink when the NResearch 161P021-11 pinch valves, parts 2, are closed. 7) The waste container.
Figure 1.6 A photograph of the mechanical part of the fluidic control system that was shown in Figure 1.5.
Figure 1.7: Circuit 1 for the computerized fluidic control of the location of the trapped cell in the system. I) Arduino Uno R3 Microcontroller  II) JANSANE 16x2 1602 LCD Arduino Display Screen Blue with I2C Module Interface Adapter  III) HiLetgo L293D Motor Driver Shield  IV) Hitec 32625S HS-625MG Hi-Speed Metal Gear 2BB Servo. Servo Motor is also shown in Figure 1.5 as Part 1.
Figure 1.8: A photograph of Circuit 1 described in Figure 1.7.
Figure 1.9. Circuit 2 for the control of the pinch valves. A) JANSANE 16x2 1602 LCD Arduino Display Screen Blue + IIC I2C Module Interface Adapter. B) Arduino Uno R3 Microcontroller C) NResearch 161D1X250 Cool Drive Single Valve Driver. D) NResearch 225D1X250 Cool Drive Single Valve Driver. E) GW Instek GPS-3303 DC Power Supply. Parts 2 and 6 are described in Figure 1.5
Figure 1.10. The combined system. The flow cell is placed on Meifi EMZ-8TR light microscope and the pictures in Figure 1.11, Figure 1.12 and Figure 1.13 are taken by the Point Grey CMLNN2 USB camera.
Figure 1.11. The fluorescent bead is trapped at the edge of the flow cell.
Figure 1.12. The florescent bead is moved downward by pushing the RIGHT button of Circuit 1.
Figure 1.13: The florescent bead is moved further down by keeping the RIGHT button of Circuit 1 pushed.
REFERENCES


8) Global Climate Change


APPENDIX A:

ARDUINO IDE CODE FOR CIRCUIT 1

// Controlling a servo with two Push buttons with Arduino. When the script starts, the angular position of the servo is set to 90 degrees. When LEFT push button is pressed, the servo starts rotating in the counterclockwise direction, until it reaches the maximum angle set (100 degrees). When RIGHT push button is pressed, the servo starts moving in the clockwise direction until it reaches the minimum angle set (80 degrees). For increasing the resolution of the system, microseconds are used instead of the actual degrees. In other words, the command 'servo.writeMicroseconds()' is employed instead of 'servo.write()' command. The following mathematical relation exists between the microseconds and degrees: “Degrees = (Microseconds - 1000)* 0.18”. At any instance if the button is released, servo stops. The circuit also includes a third push button, RESET, for setting the angular position of the servo to 90 degrees at any instance it is pressed.

#include <Servo.h> // include the servo library
#include <Wire.h> // include the wire library
#include <LiquidCrystal_I2C.h> // that is a special library for LCD monitoring

LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); // Setting the pins for the I2C interface adapter of the JANSANE 16x2 1602 LCD Arduino Display Screen

Servo myservo; // create servo object to control a servo
int angle =1500; // initial angle for servo
int angleold; // a generic parameter to trigger the LCD display update
int angleStep = 1; // the angle step is set to 1 microsecond
int LEFT = 12; // setting the pin for the LEFT button
int RIGHT = 2; // setting the pin for the RIGHT button
int RESET = 5; // setting the pin for the RESET button
#define LEFT 12   // pin 12 is connected to LEFT button
#define RIGHT 2  // pin 2 is connected to RIGHT button

void setup() {
  lcd.begin(16, 2); // setup LCD
  lcd.clear();  // clear up the LCD monitor
  Serial.begin(9600);  // setup serial
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
  pinMode(LEFT, INPUT_PULLUP); // assign pin 12 as input for LEFT button
  pinMode(RIGHT, INPUT_PULLUP); // assign pin 2 as input for RIGHT button
  pinMode(RESET, INPUT_PULLUP); // assign pin 5 as input for RESET button
  myservo.write(angle); // send servo to the middle at 90 degrees
  lcd.print("angle: "); // type the string "angle" on the LCD script
  lcd.print(((angle - 1000) * 0.18)); // type the angular position of the servo on the LCD screen, after converting it to degrees.
  attachInterrupt(digitalPinToInterrupt(RIGHT), ISRfunction , CHANGE); // Start the loop 'ISRfunction' upon any change in the RIGHT button signal
  attachInterrupt(digitalPinToInterrupt(LEFT), ISRfunction , CHANGE); // Start the loop 'ISRfunction' upon any change in the LEFT button signal
}

void ISRfunction () {
angle = angle; // upon any interrupt in the RIGHT or LEFT button, the angular position of the servo is updated.

void loop() { angleold = angle; // the generic parameter 'angleold' is set to be equal to the angular position of the servo

while(digitalRead(RIGHT) == LOW) {

//When the RIGHT button is pressed, the angular value is decreased by 'angleStep (1 microsecond)'. However, the maximum angle is 100 degrees (1557 microseconds) and the minimum value is 80 degrees (1444 microseconds). If the angular position is beyond these limits, the rotation is automatically brought to a halt.

if (angle > 1444 && angle <= 1557) {
    angle = angle - angleStep;

    if(angle < 1445) {
        angle = 1445;
    }else{
        myservo.writeMicroseconds(angle); // move the servo to desired angle
    }

    lcd.clear(); // clear up the LCD monitor
    lcd.print("angle:"); // type the string "angle" onn the LCD script
    lcd.print((angle-1000)*0.18); // type the angular position of the servo on the LCD screen, after converting it to degrees
    delay(150); } // waits for the servo to get there

// When the LEFT button is pressed, the angular value is increased by 'angleStep (1 microsecond)'. However, the maximum angle is 100 degrees (1557 microseconds) and the minimum value is 80 degrees (1444 microseconds). If the angular position is beyond these limits, the rotation is automatically brought to a halt.
while(digitalRead(LEFT) == LOW)
{
    if (angle >= 1444 && angle <= 1557) { angle = angle + angleStep;
if(angle >1556) {
    angle =1556; }else{
    myservo.writeMicroseconds(angle) };// move the servo to desired angle
    delay(150); // waits for the servo to get there
    lcd.clear();
    lcd.print("angle:");
    lcd.print((angle-1000)*0.18); }
if ( angle == angleold) { }
else { lcd.clear(); // clear up the LCD monitor
    lcd.print("angle:");
    lcd.print((angle-1000)*0.18); }
if ( digitalRead(RESET) == LOW ) {
    myservo.writeMicroseconds(angle); // move the servo to desired angle
    lcd.clear(); // clear up the LCD monitor
    lcd.print("angle:"); // type the string "angle" on the LCD script
    lcd.print((angle-1000)*0.18) });// type the angular position of the servo on the LCD
    screen, after converting it to degrees.
APPENDIX B:

ARDUINO IDE CODE FOR CIRCUIT 2

// The script to control the actuation of NResearch 161P021-11 pinch valves with a push button. The delay time between the actuations of the valves is set by a potentiometer. The delay time ranges from -500 ms to + 500 ms. The first valve (pin 7) is actuated after the second one (pin 10) for the negative values, and the second valve is actuated after the first one for the positive values. The valves are connected to NResearch Cool Drive Single Valve Drivers, and these drivers are connected to the Elegoo Uno r3 Microcontroller at pins 7,10 and 12. The delay time is also shown on the LCD display.

// First the pins used are declared:

int pbuttonPin = 2; // connect output to push button
int relayPin1 = 10; // Connected to second NResearch 161D1X250 Cool Drive Single Valve Driver
int relayPin2 = 7; // Connected to second NResearch 161D1X250 Cool Drive Single Valve Driver
int relayPin3 = 12; // Connected to second NResearch 225D1X250 Cool Drive Single Valve Driver
int potPin = A0; // Declare potPin to be analog pin A0
int readValue; // Use this variable to read Potentiometer
int writeValue; // Use this variable for setting the delay time between the actuations of the valves
int writeValueOld; // A generic variable created to check if the potentiometer value is altered significantly. That is for updating the LCD display value without monitoring the fluctuations in the potentiometer value.

int val = 0; // push value from pin 2
int pushed = 0; // push status

#include <Wire.h> // include the wire library
#include <LiquidCrystal_I2C.h> // that is a special library for LCD monitoring

LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); // Setting the pins for the I2C interface adapter of the JANSANE 16x2 1602 LCD Arduino Display Screen

int sensitivity; // the minimum alteration in the potentiometer value for updating the LCD display is declared.

void setup() {

  pinMode(pbuttonPin, INPUT_PULLUP); // assign pin 2 as input for the push button
  pinMode(potPin, INPUT_PULLUP); // set potPin to be an input
  pinMode(relayPin1, OUTPUT); // assign pin 10 as output
  pinMode(relayPin2, OUTPUT); // assign pin 7 as output
  pinMode(relayPin3, OUTPUT); // assign pin 12 as output

  digitalWrite(relayPin1, HIGH); // keep the load OFF at the beginning. If you wanted to be ON, change the HIGH to LOW
  digitalWrite(relayPin2, HIGH); // keep the load OFF at the beginning. If you wanted to be ON, change the HIGH to LOW
  digitalWrite(relayPin3, HIGH); // keep the load OFF at the beginning. If you wanted to be ON, change the HIGH to LOW
}
attachInterrupt(digitalPinToInterrupt(pbuttonPin), function , CHANGE); // Start the loop 'function' upon any change in the push button signal

readValue = analogRead(potPin);  //Read the voltage on the Potentiometer 1

writeValue = (1000./1023.) * readValue; //Calculate Write Value for LED

sensitivity = 2.;// the minimum alteration in the potentiometer value for updating the LCD display is declared.

lcd.begin(16,2);// setup LCD

lcd.clear(); // clear up the LCD monitor

if (writeValue<500) {// if writeValue<500, it is a negative delay time

clear(); // clear up the LCD monitor

c.print("delay:");// type the string "delay" on the LCD script

c.print(-abs(writeValue-500)) ;// subtract 500 from the write value and print it on LCD display

} else { lcd.clear(); // clear up the LCD monitor

c.print("delay:");// type the string "delay" on the LCD script

c.print(abs(writeValue-500))); // subtract 500 from the write value and print it on LCD display

void function () {// This function will be executed upon any change in the Push Button signal. Update the Write Value upon any change in the push button signal:

val = digitalRead(pbuttonPin);// read the push button value

readValue = analogRead(potPin); //Read the voltage on the Potentiometer 1

writeValue = (1000./1023.) * readValue; //Calculate Write Value for LED

writeValueOld = (1000./1023.) * readValue;
if(val == LOW){
pushed = 1-pushed;}

if ( writeValue < 460 || writeValue > 540 ) { // If the Write Value is around 500, which
means an LCD display number around 0, actuate all valves simultaneously upon any
trigger. Otherwise, actuate the first two valves upon a trigger. The third valve will be
actuated later in the loop in the latter case. The "third vale" is selected based upon the
Write Value.
} else {
if(pushed == HIGH){
digitalWrite(relayPin1, LOW);
digitalWrite(relayPin2, LOW);
digitalWrite(relayPin3, LOW); }
else { digitalWrite(relayPin1, HIGH);
digitalWrite(relayPin2, HIGH);
digitalWrite(relayPin3, HIGH); }}}

void loop () { // at least two of the valves were actuated in the interrupt function, before
this main loop
readValue = analogRead(potPin); //Read the voltage on the Potentiometer 1
writeValue = (1000./1023.) * readValue; //Calculate Write Value for LED
if ( writeValue < 460 || writeValue > 540 ) { // if the Write Value is not around 500,
actuate the third valve after the delay time:
if(pushed == HIGH){
if (writeValue<500){
digitalWrite(relayPin1, LOW);
digitalWrite(relayPin3, LOW); } 
else { digitalWrite(relayPin2, LOW);
digitalWrite(relayPin3, LOW); }
else{ if (writeValue<500) {
digitalWrite(relayPin1, HIGH);
}
else{
digitalWrite(relayPin2, HIGH); 
}};
if(pushed == HIGH){
if (writeValue<500){
delay(500-writeValue);
digitalWrite(relayPin2, LOW});
else { delay(abs(writeValue-500));
digitalWrite(relayPin1, LOW})
else{ if (writeValue<500){
delay(500-writeValue);
digitalWrite(relayPin2, HIGH});
else{ delay(abs(writeValue-500));
digitalWrite(relayPin1, HIGH});
if (pushed == LOW) { digitalWrite(relayPin3, HIGH) } 

if(abs(writeValueOld-writeValue)>sensitivity){/ If Write Value is smaller than 500, that is a negative delay time. The valve connected to pin 10 is actuated before the valve connected to pin 7 for the negative delay times. If Write Value is larger than 500, that is a
positive delay time and pin 7 is actuated before pin 10 in this case. LCD display value is calculated by subtracting 500 from the Write Value.

```cpp
if (writeValue<500) {
    lcd.clear();// clear up the LCD monitor
    lcd.print("delay:");// type the string "delay" on the LCD monitor
    lcd.print(-abs(writeValue-500))// subtract 500 from the write value, multiply it with -1
    and print it on LCD display
else {
    LCD monitor lcd.print(abs(writeValue-500)))// subtract 500 from the write value, and
    print it on LCD display, and the loop closes.
```