Laboratory agitator controlled by Arduino

Tanawat Musika

A Report Submitted in Partial Fulfillment of the Requirements
for the Degree of Bachelor of Engineering (Petrochemical Engineering)
Department of Chemical Engineering, Faculty of Engineering,
King Mongkut's Institute of Technology Ladkrabang
Academic Year 2018

การควบคุมมอเตอร์สำหรับเครื่องปั่นกวนโดยใช้อาร์ดูโน่

ธนวัฒน์ มูสิกะ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมปิโตรเคมี

ภาควิชาวิศวกรรมเคมี คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

Title	Laboratory agitator controlled by Arduino
By	Mr. Tanawat Musika
Field of Study	Petrochemical Engineering
Advisor	Asst. Prof. Dr. Santi Wattananusorn

Accepted by the Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang in Partial Fulfillment of the Requirements for the Degree of Bachelor of Engineering (Petrochemical Engineering).

Thesis Committee

_______________________________Chairman
(Asst. Prof. Dr. Santi Wattananusorn)

_______________________________Committee
(Asst. Prof. Dr. Surat Areerat)

_______________________________Committee
(Asst. Prof. Dr. Yaneeporn Patcharavorachot)

Title	Laboratory agitator controlled by Arduino
By	Mr. Tanawat Musika
Advisor	Asst. Prof. Dr. Santi Wattananusorn
Field of Study	Petrochemical Engineering

Abstract

The objective of this research is to produce laboratory agitator. There are many laboratory agitators for sale but its cost is really high. This project will produce laboratory agitator to decrease the cost of the mixing process by using Arduino to control motor. Arduino is microcontroller with low price and easy to use. It can control by writing command code in Arduino IDE program. The direction, speed and working time of motor can be input from the numeric keypad to drive the motor. During rotation of motor, the LCD screen will show the status of the motor. After the construction of laboratory agitator is complete, it is found that the production costs decrease by approximately 80%.

Keywords: agitator, mixing, Arduino.

เรื่อง การควบคุมมอเตอร์สำหรับเครื่องปั่นกวนโดยใช้อาร์ดูโน่

โดย นายธนวัฒน์ มูสิกะ

อาจารย์ที่ปรึกษา ผศ.ดร.สันติ วัฒนานุสรณ์

สาขาวิชา วิศวกรรมปิโตรเคมี

บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์เพื่อผลิตเครื่องปั่นกวนสำหรับห้องแลป เนื่องจากเดิมมีเครื่องปั่นกวนสำหรับห้องแลปหลายแบบวางขายทั่วไป แต่มีราคาที่ค่อนข้างสูง จึงมีการคิดค้นวิธีที่จะสร้างเครื่องปั่นกวนสำหรับห้องแลป โดยคำนึงถึงการลดต้นทุนในการผลิตเป็นสำคัญ ดังนั้นจึงได้ใช้อาร์ดูโน่ ซึ่งเป็นไมโครคอนโทรลเลอร์ที่มีราคาต่ำ และง่ายต่อการใช้งาน มาควบคุมมอเตอร์ที่ใช้ปั่นกวน โดยการเขียนคำสั่งการทำงานผ่านโปรแกรม Arduino IDE ให้รับค่าทิศทางการหมุน ความเร็ว และเวลาที่ทำงานจากแป้นตัวเลข แล้วสั่งให้มอเตอร์หมุน โดยมีหน้าจอแสดงสถานะการทำงานของมอเตอร์ ณ ขณะนั้น หลังจากที่สามารถผลิตเครื่องปั่นกวนสำหรับห้องแลปได้แล้วนั้น พบว่าใช้ต้นทุนในการผลิตลดลง 80% โดยประมาณ

คำสำคัญ: เครื่องปั่นกวน, การผสมสาร, อาร์ดูโน่

Acknowledgements

I would like to express the deepest appreciation to my advisor Asst. Prof. Dr. Santi Wattananusorn, who has the attitude and the substance of a genius: he continually and convincingly conveyed a spirit of adventure in regard to research and scholarship, and an excitement in regard to teaching. Without his guidance persistent help, this dissertation would not have been possible.

I would like to thank my committee member, Asst. Prof. Dr. Surat Areerat and Asst. Prof. Dr. Yaneeporn Patcharavorachot, for the suggestions related to my research. Without their suggestion, my research might be not correctly developed.

I would like to thank department of chemical engineering, King Mongkut's Institute of Technology Ladkrabang and authorities, Mr. Pisan Ponpo, Mrs. Pimchai Poochanakit, and Mr. Aekkarat Bumrungthaichaichan, for supporting the equipment and workspace. My research might be not successful without them.

Finally, I would like to thank faculty of teachers for the knowledge that has been teaching in the bachelor degree of Petrochemical Engineering.

Tanawat Musika

Table of Contents

Table of Contents (cont.)

List of Figures

List of Tables

CHAPTER I
INTRODUCTION

1.1 Background

The mixing process developed from 1950s with the work of Uhl and Gray (1966) and Nagata (1975) [1]. Mixing process is widely used throughout the chemical industry in processes related to physical and chemical changes. Mixing process is not just used in the chemical industry. Other industries such as medical foods also have mixing process. As a result, the cost of the mixing process increased dramatically [2].

Agitators are machines used in mixing process [3] and other process (blending, heat transfer, etc.), depended on process function. Material selection will be considered for corrosion resistance to product properties [4]. Over the past 30 years, designing multiple engineering design principles and designing mixing equipment for the purpose of the desired process has become possible [1].

Today, the technology has progressed. Microcontrollers It is a small control device. Act like a small computer, what do we want it to do, we write the program we want to stuff it into microcontroller [5] Microcontrollers are widely used today. It is easy to control electronic and electrical devices.

The purpose of this work is to produce a small-scale agitator. The author considers that the microcontroller can be used combined with engineering knowledge in design and create an agitator machine to have equal ability or near the blender used in the current. The cost of agitator is lower. This reduces the cost of mixing process.

1.2 Objectives

1.2.1 To study of working system of Arduino.

1.2.2 To produce a small-scale agitator for laboratory controlled by Arduino.

1.3 Scopes of work

1.3.1 Use Arduino to control a small-scale agitator.

1.4 Expected Outputs

1.4.1 Reduce the cost of agitator.

CHAPTER II
LITERATURE REVIEW

2.1 Mixing

2.1.1 Mixing process

Mixing is the reduction of inhomogeneity in order to achieve a desired process result. The inhomogeneity can be one of concentration, phase, or temperature. Secondary effects, such as mass transfer, reaction, and product properties are usually the critical objectives [2].

In the engineering industry, the mixing process is the operation of the departments concerned in managing different physical systems with the intention of being more homogeneous. Familiar examples include pumping in the pool to dissolve the water temperature and stimulate the pancakes to eliminate the lumps. Mixing is done for heat transfer and / or mass transfer between parts or single or multiple stages [7]. Modern industrial processes often involve the formation of mixtures. Some chemical reactors have mixers. With the right equipment can mix solids, liquids or gases into solids, liquids or gases. Bio-diesel fermenters may require the mixing of microorganisms, gases and liquids for maximum yield. Organic nitrites require concentrated nitric acid (liquid) and sulfuric acid to mix with the hydrophobic organic phase. Tablet production is solid [8].

2.1.2 Mixing classification

The type of work and equipment used during mixing depends on the state of the materials being blended. (Semi-solid or solid liquid) and the mixing capacity of the material being processed. In this context, the act of mixing may be synonymous with agitation or process massage [8].

2.1.3 Mixing mechanisms

There are two types of mixing process: convective mixing and intensive mixing [9]. In the case of convective mixing, they are transported from one location to another. This mixer will lead to fewer order conditions within the mixer. Mixing components are spread throughout the mixture. Over time, the mix becomes a random order. After the blending time, the maximum probability is reached. This type of mix is often used for flowing and coarse materials [7].

2.1.4 Laboratory mixing

In the laboratory, mix with a magnetic stirrer or hand. Sometimes mixing in laboratory containers is better and faster than possible in the magnetic barrel industry. This is acceptable in small sizes because of the smaller vessels and the resulting ingredients. Due to the small size and (in general) the low viscosity of the liquid, it is possible to use the configuration for almost all ingredients. The cylinder can be used for suspension of solids seen in iodine. deagglomeration It is beneficial for media preparation, microbial growth from powders, and liquid and liquid mixtures. Another set of laboratories is a mixer at the bottom of the vessel, instead of being suspended near the center. In addition, the containers used in the mixing laboratory are often different from those used in the mixed industry. For example, an Erlenmeyer bottle or a Florence bottle may be used in addition to a cylindrical cylinder [7].

2.1.5 Industrial mixing equipment

On an industrial scale, effective system integration can be difficult. Many engineering efforts go into designing and improving the mixing process. Industry mixes are made in batches (dynamic mixers), inline or with the help of a static mixer. Electric motors that run at 1800 or 1500 rpm are usually faster than those set. Gearbox used to reduce speed and increase torque. Some programs require a multi-shaft mixer that mixes well [10].

In addition to traditional batch processing, some combinations are still available. At least one ingredient and at least one liquid mixture can be accurately and uniformly inspected in the machine and continuously mixed in the machine. For a number of reasons, some of them are easy to clean, reduce power consumption, reduce footprint, control the versatility, and much more. Continuous mixers such as dual core processors can also support high viscosity [7].

2.2 Agitator

The agitation is caused by different mass movements (liquid - solid phase) with the propeller formed by a mechanical catalyst to rotate the propeller. This group consists of various substances and the aim of the work is to blend or to improve the efficiency of the reaction by good contact between the reaction products. Or the group is already mixed, and the purpose of stirring is to increase heat transfer or to keep the particles in suspension to avoid any deposit.

Agitation have 4 types mainly used, they are Anchor, Turbine, Propeller and Gas induction. [3] The option of stirring depends on the distance to be mixed. (One or more stages): liquid only liquid and solid, liquid and gaseous or liquid with solid and gas Depending on the type of phase and viscosity of the mixer group can be named mixer, kneader, starch mixer and so on. Liquid catalyst can be placed on the top of the tank in vertical or horizontal position. sleep (Side of the tank) or less [4]. The stirrer is located at the bottom of the tank.

The agitator consists of a propulsion device. The navigation system of the shaft (Bearings), shaft and rotor. If the working under high conditions, high pressure or high temperature, the impulse mixer must be fitted with a sealing system to keep the inner side of the tank tight when the cross shaft [3].

2.3 Microcontroller (MCU)

Microcontrollers are small computers that are integrated into a single chip. Central processing unit (CPU) circuit, input / output, memory, RAM and flash, timer counters, and so on. Microcontrollers are electronic devices that control electrical or electronic devices. The microcontroller is like a small computer. Wanting to do it, we write the program we want to stuff it into. Microcontrollers are involved to cope with the desire to control the system that we want to have the smallest. But it's not just small. It can also be used to execute commands automatically. With the programming language, according to the expertise of the product designer. And inventors. The inevitable need to rely on electronic circuits. Be involved in the control. But the electronic circuit that brings the serial to the ability that we need. It was so big. It seems to conflict with the needs of consumers. And product design principles. Microcontrollers that are used today are many, such as PLC, AVR Arduino, etc. Arduino is the most attractive because it has a low price. Easy to use. But the performance is high [5].

2.4 Arduino

Arduino is Microcontroller have control Commit electronics to work as we want. Arduino highlights in terms of ease of use for newbies. because the command style is designed to be uncomplicated and its Open Source, is there any information disclosure hardware and software, easy to modify development of the whole board or program and can be really used [6].

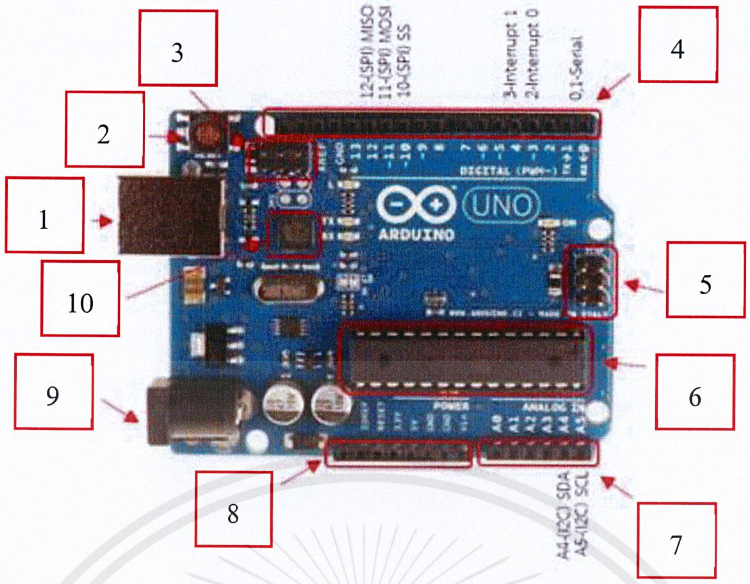

Figure 1. Layout & Pin out Arduino Board (Model: Arduino UNO R3)

1.USB Port: Used to connect with Computer to download the program microcontroller and power supply to the board.

2.Restet Button: button Reset Press to allow. MCU restart

3.ICSP Port: It is a port that uses the program.

4. Digital Input/output Port

5.ICSP Port: It is a port that uses the program bootloader

6. Microcontroller: Atmega328 are Microcontroller on board Arduino

7. Analog Input/output Port

8. Power Port: Boards for power supply to external circuits. Includes power pins + 3.3 V, + 5V, GND, Vin

9. Power Jack: Get power from Adapter by the voltage between 7-12 V

10. MCU 's the Atmega16U2 MCU acts as The USB to Serial Atmega328 will interface with Computer through Atmega16U2

Arduino use the programming style on the computer Via the program Arduino IDE It is free software for writing code. Code writing is like C, which is the basic language of programming. But it will be slightly different. When writing the code, then continue with the computer through the channel. USB, then select the board and the connection port. Upload Arduino will start working on the code we wrote.

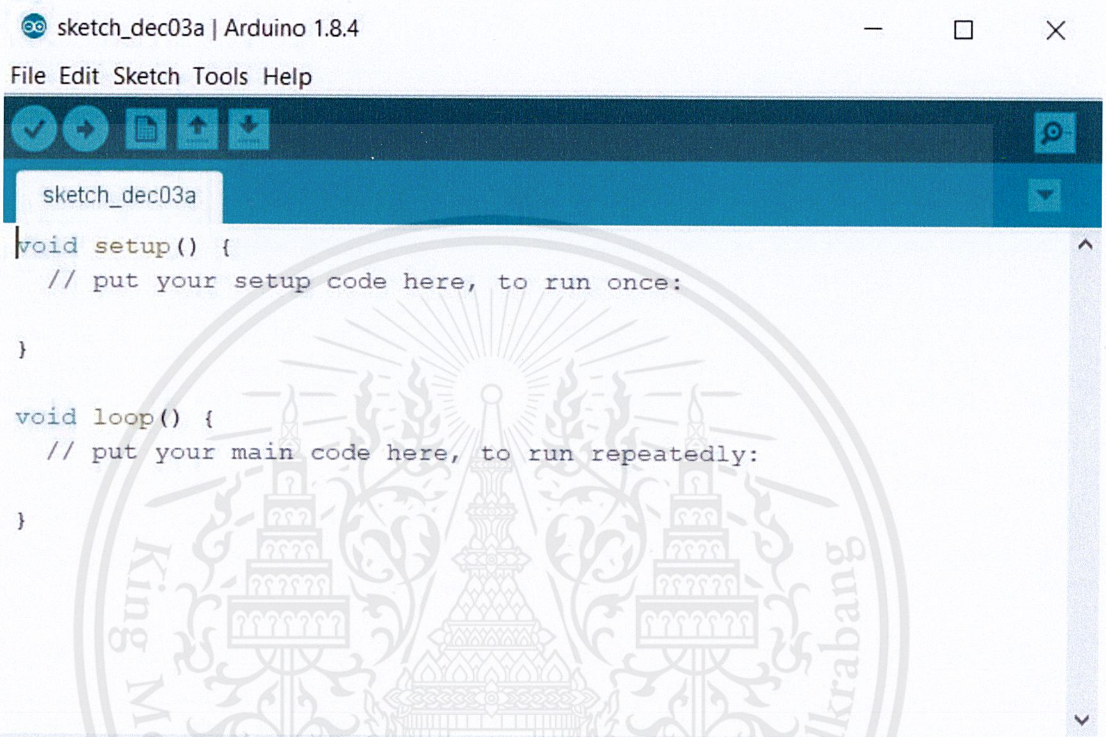

Figure 2. Arduino IDE software windows

What makes Arduino so interesting?

-Arduino has been popular in the MCU industry for many years, with both electronic and novice electronics, so we can find a guide to how to solve it. Easy on the Internet.

-Arduino is ready to use immediately because the Arduino board is equipped with basic equipment.

-Arduino can be programmed with C / C ++ syntax, which is easy for anyone with some programming backgrounds. But for those who have never programmed before. I can start studying. And finding books is not difficult. There are also many libraries to choose from. Make programming easier and faster.

-Arduino is too inexpensive for those who want to get started.

-Uploading programs written on your computer to the Arduino is as simple as using a USB cable to connect the Arduino board to your computer. Then upload with Arduino IDE [6].

2.5 Pulley system

A pulley system consists of two pulleys (usually of different diameters) and a belt loop to link the pulleys. In the figure above, the belt is marked with a red color.

One of the two pulleys is called the driver pulley - it means that transmitting power is applied to it, causing it to rotate. The other pulley is called the driven pulley. It rotates because of the force transmitted through the belt.

There are two main parameters associated with each of the pulleys. The first one is the diameter (twice the radius), and the second one is its angular velocity measured in revolutions per minute.

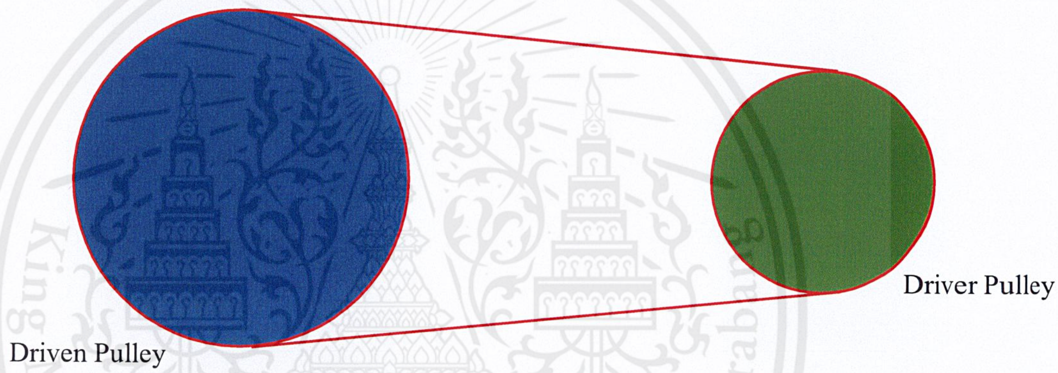

Figure 3. Pulley system

This pulley calculator analyzes a system of two pulleys joined by a conveyor belt (also called a belt drive). You can use it to calculate the pulley RPM (revolutions per minute), but also its diameter, and some properties of the whole system (such as the pulley speed, belt tension or torque). You can use this tool right away or continue reading to learn more about the logic behind the pulley formulas.

CHAPTER III
EXPERIMENTAL

3.1 Study of working system of Arduino

Arduino is used to communicate and control other electrical devices. With programming to control the electrical signal. In order for the electronics to work in accordance with the conditions set. Arduino operates in a loop, it will run loop in the command repeatedly until the stop command.

3.2 Study of working of agitator

Agitator the prototype of the study is MTOPS Model BL 620D which can be stirred at a speed of 2,000 RPM maximum. Use electrical power 60W $230V_{AC}$ 50Hz

Figure 4. prototype agitator

3.3 Equipment

3.3.1 Arduino Uno R3

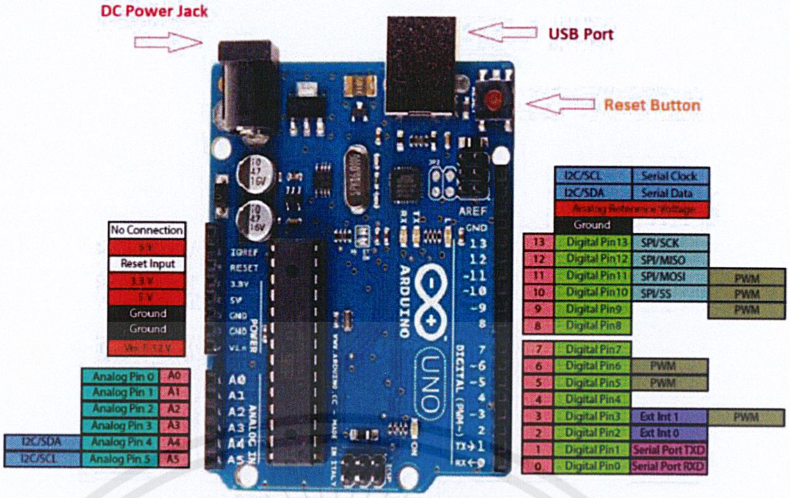

Figure 5. Arduino Uno R3 & Pin layout

The reason for choosing Arduino Uno R3 is because it is a widely used model. Suitable for use with the device that will be created and has a sufficient number of input/output ports for use. The use uses a digital port to control the motor, receive values from the keypad and receive process variable of speed from the Electric Motor Speed Sensor. Use an analog port to control the LCD screen and speakers. And use the 5V and GND port to supply power to the other devices.

3.3.2 XD-3420 24V Motor & Pulley system

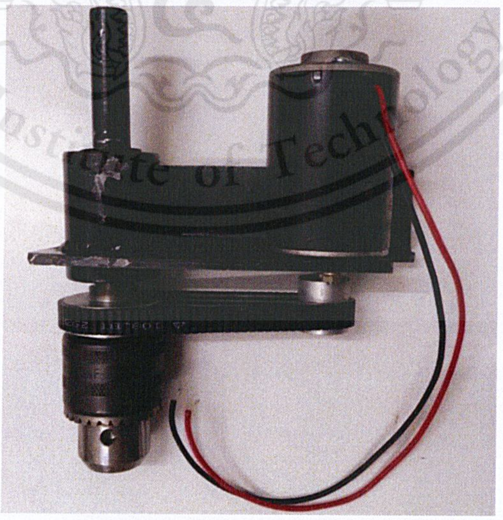

Figure 6. XD-3420 24V Motor & Pulley system

XD-3420 is a 24-volt 48-watt motor connected to a drill chuck with a pulley system and has a strong mounting bracket.

3.3.3 Module L298N

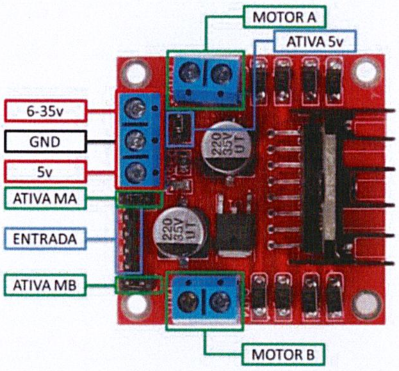

Figure 7. Module L298N & Pin layout

Module L298N is a device used to drive a motor with a maximum voltage of 24 volts and a maximum power of 48 watts with H-Bridge built-in for the direction control of motor, this make it easy to use and do not have to add a H-Bridge circuit to control motor direction. The usage will receive the value from Arduino through the ENTRADA port to order motor direction and ATIVA MA to order the motor speed. Then use the MOTOR A port to control the motor, using a 5V and GND port to receive the power from Arduino and use a 6-35V and GND port to receive the power from power supply

3.3.4 2004 LCD With IC2

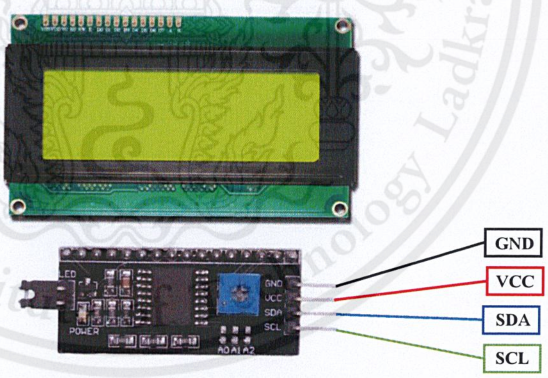

Figure 8. 2004LCD, IC2 & Pin layout

2004LCD is an LCD screen that can display 4 rows of 20 characters per row and is used in conjunction with IC2, which will help reduce the number of ports used to connect to only 4 ports. The usage will connect the SDA and SCL ports to the Arduino to receive the display orders and use the VCC and GND port to receive the power from Arduino.

3.3.5 4x4 matrix keypad

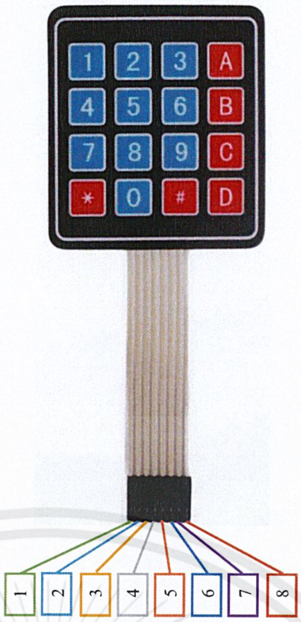

Figure 9. 4x4 matrix keypad & Pin layout

4x4 matrix keypad is an Arcade Push Button 16 button sorting together into a matrix, used to enter values into the Arduino. The usage will connect port 1-8 to the Arduino to enter data into the Arduino.

3.3.6 FC-33

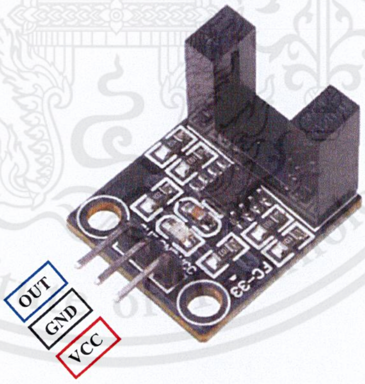

Figure 10. FC-33 & Pin layout

FC-33 is a light sensor used to count the number of rounds of agitator. Work by releasing light from one side and having a light detector on the other side. If the propeller passes through, the receiver will count once when the light disappeared, counting 20 times, equal to 1 rotation cycle. The usage will connect OUT port to the Arduino to return the speed and use the VCC and GND port to receive the power from Arduino.

3.3.7 Power supply

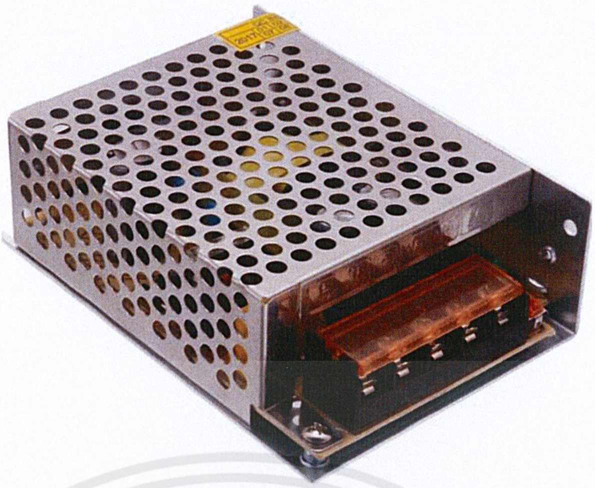

Figure 11. Power supply

24 volt power supply for drive motor and 12 volt for Arduino

3.4 Arduino working design

The design of the agitator works as close to the agitator as studied. Then write the algorithm as shown in Figure. so that it will be easier to write the code in the next step.

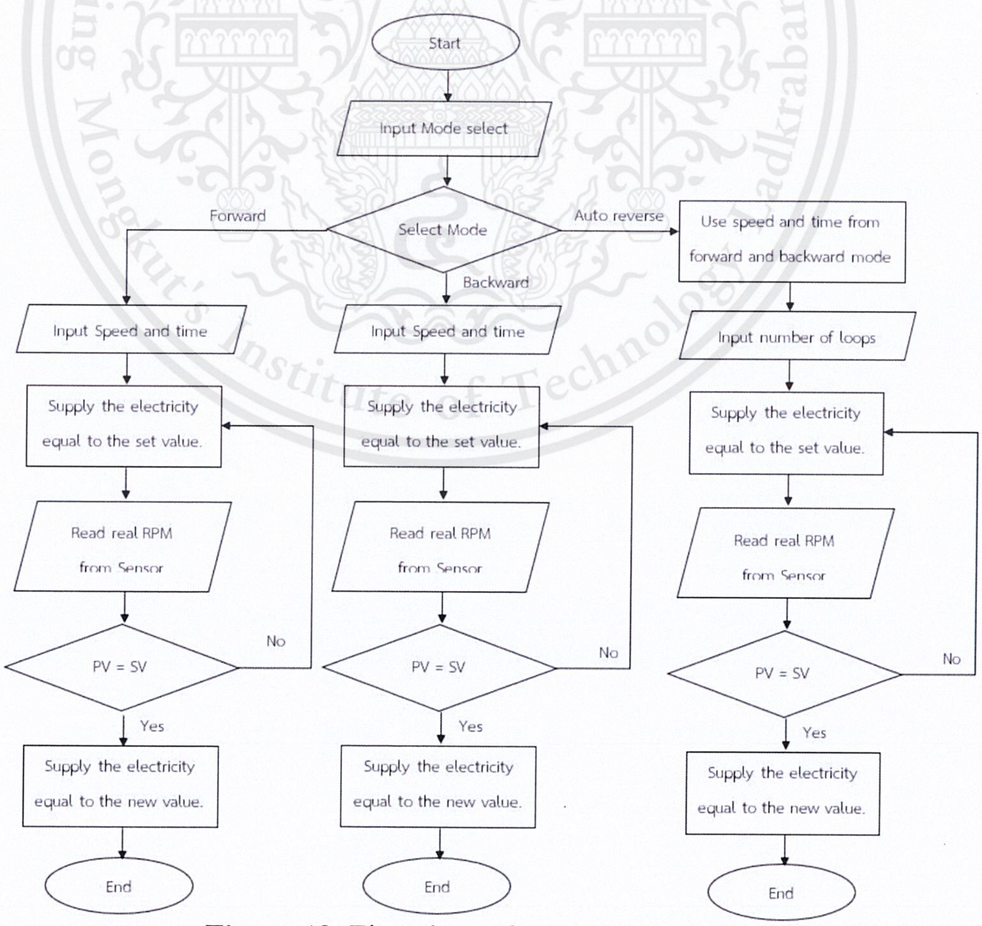

Figure 12. Flowchart of working of agitator

3.5 Write code according to the working designed and test run

Write code in the Arduino IDE program to work as a drawing algorithm. If use agitator, stir the mixer in a high viscosity liquid. This may cause the rotational speed of the motor to drop while supplying the same current. The speed readings are required to get the actual speed and then the power supply. The motor can be increased and rotate according to the number of cycles set.

3.6 Machine manufacture

3.6.1 Control system

Assembling the agitator using Arduino, L298N Module and FC33 to control the motor, keypad was used to enter values into the Arduino and display via 2004 LCD. Connect various devices together according to the port shown in the table 1.

Table 1: Porting of various devices

Module Pin	**Arduino Pin**
LCD	
VCC	5V
GND	GND
SDA	A4
SCL	A5
L298N	
ENA	3
IN1	8
IN2	9
12V	24 V Power supply
GND	GND power supply & Arduino
5V	5V
OUT 1	Motor +
OUT 2	Motor -
Sensor Signal	2

Table 1: Porting of various devices (continuous)

Module Pin	Arduino Pin
Keypad	
1	4
2	5
3	6
4	7
5	10
6	11
7	12
8	13
Buzzer +	A0
Buzzer -	GND

Use Arduino and L298N Module to command the motor to spin. Connect the motor and rotor handle with belt and install a small rotor on top of rotor handle. Install a speed sensor around small rotor to measure the speed and feedback speed to Arduino to supply the electricity to the desired speed.

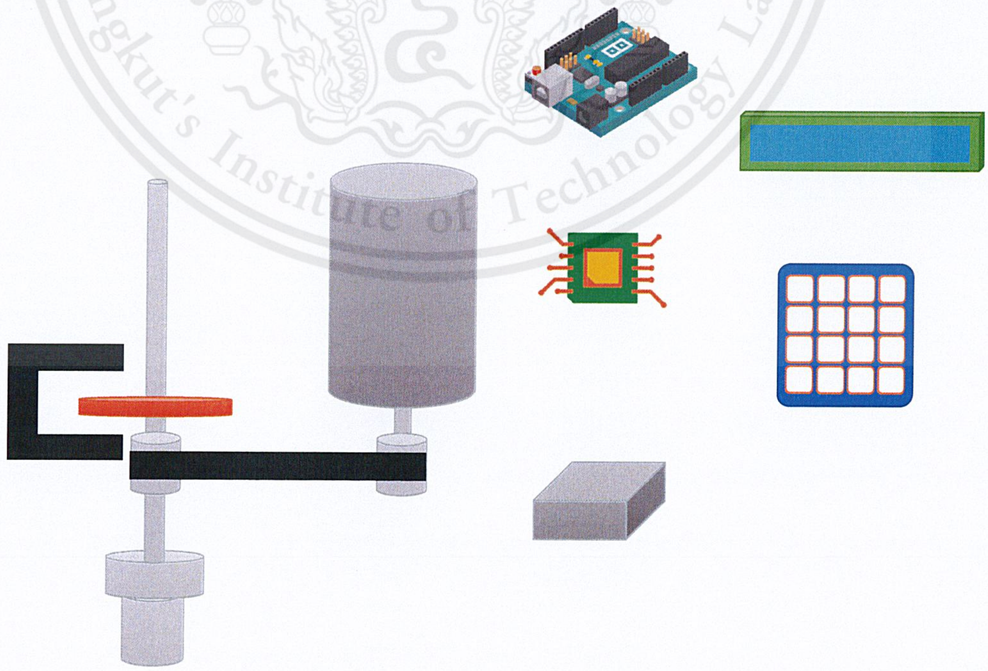

Figure 13. Control system

3.6.2 Pulley system design

(1) Diameter and RPM of each pulley

For a pulley system like this, the product of pulley diameter (d) and RPM (n) is the same for both driver and driven pulley. It means that

$$d_1 * n_1 = d_2 * n_2$$

This formula can use to find any of these four values: driver pulley diameter (d_1), it's angular velocity (n_1), the driven pulley diameter (d_2) or it's angular velocity (n_2).

When the driven pulley diameter is obtained, weld the motor and shaft into the steel frame used as a clamp. Then then made a pulley that are the diameter that is needed on both sides, install the pulley on the motor and shaft with a belt and turning the screw to insert the drill chuck.

Figure 14. Pulley system

(2) Drive torque

The last value that can be found with this pulley calculator is the drive torque (torque of the driver pulley). Use the following equation:

$$T = \frac{P}{\frac{2\pi n_1}{60}}$$

where the angular velocity n_1 is expressed in revolutions per minute.

3.6.3 Object manufacture

Connect the motor to the pulley system to catch the rotor to use in the mixing tank or the reaction tank. Then it is integrated into the main unit. Then make a bracket to place the agitator.

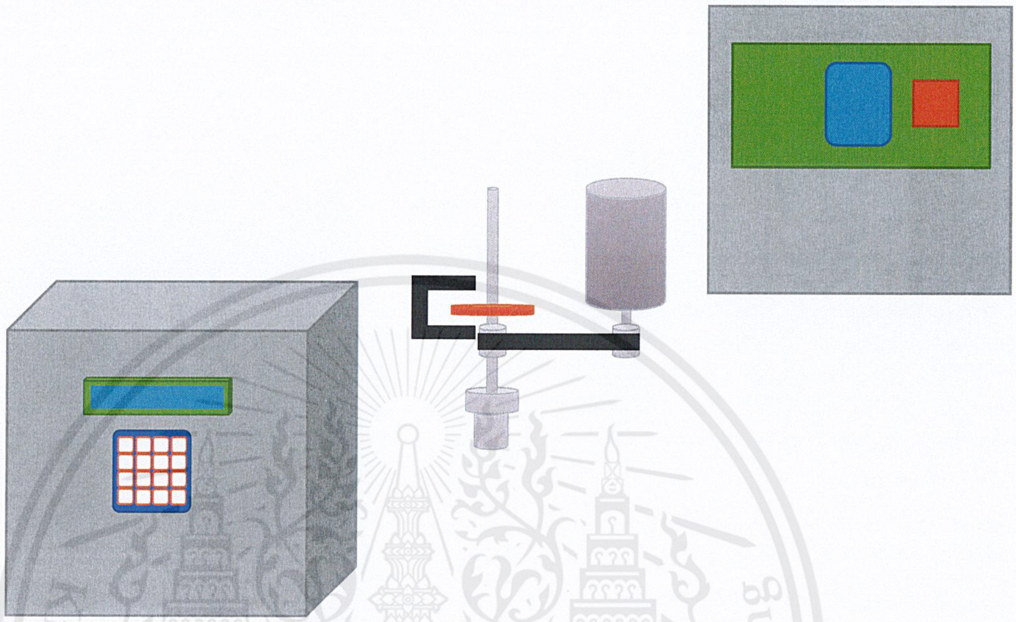

Figure 15. Object manufacture

CHAPTER IV

RESULTS AND DISCUSSION

4.1 Usability

4.1.1 Calibration step

Upload calibrate code into agitator to find the maximum speed of agitator that can be used. After done uploading, the agitator will automatically run for 60 seconds and then stop counting and will show calibration factor on LCD screen.

When get the calibration factor, put the value in the variable "m_max_speed" in working code. The program will take this value to calculate the maximum speed that agitator can be used, then upload the working code to the agitator.

4.1.2 Working step

(1) Select mode

When the agitator is turned on, it will enter the operating mode selection window, there are three modes: Forward, Backward and Auto reverse. Press the numerical of mode on keypad to choose the require mode.

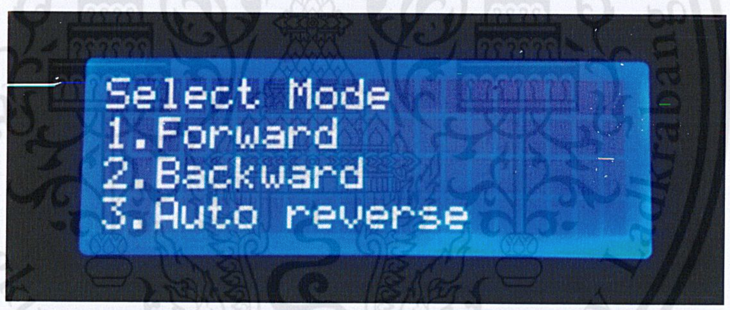

Figure 16. Operating mode selection window

(2) Forward or Backward mode

(2.1) If Forward or Backward mode was selected, the agitator will enter the setting window. Which has 3 values that need to be entered: speed (RPM), time in minutes (Min) and time in seconds (Sec).

Figure 17. Setting window of Forward and Backward mode

(2.2) In setting window, can select the value that need to enter by pressing " * " the cursor will move the position during 3 values that must be entered. Press the numeric on keypad to set the desired value, then press "A" to enter the value into that variable. Repeating until all 3 values are entered.

(a)

(b)

Figure 18. (a) setting window with some value entered
(b) setting window with all value entered

(2.3) When entering all 3 values, press "B" to start working and the agitator will enter the working window. While working, the agitator will show the direction on an LCD screen and showing the both of speed (Process Variable and Setting Variable). In that part of the timer, it will countdown the time as set along with telling the time set in brackets. When counting down to zero, the agitator will stop working automatically.

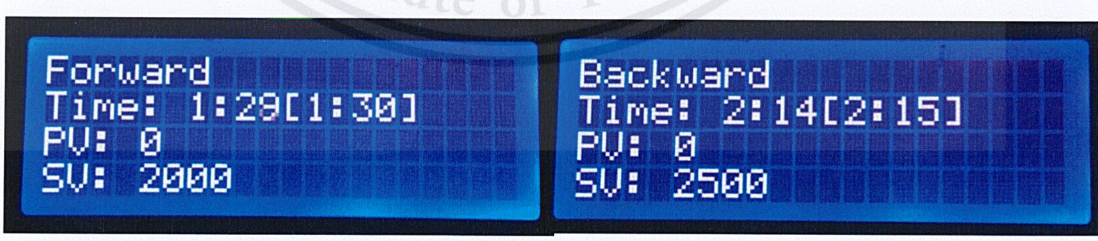

Figure 19. Working window of Forward and Backward mode

(3) Auto reverse mode

(3.1) If Auto reverse mode is selected, the agitator will be used the speed and time from Forward and Backward mode and display the speed and time of both directions on LCD screen. Therefore, before use this mode user have to enter the speed and time in mode Forward and Backward mode first.

The value that must be entered in this mode is the number of loops which is shown at the top of the screen. Can increase the number of loops by pressing "A" and reducing the number of loops by pressing "D". The minimum number of loops is 1 loop.

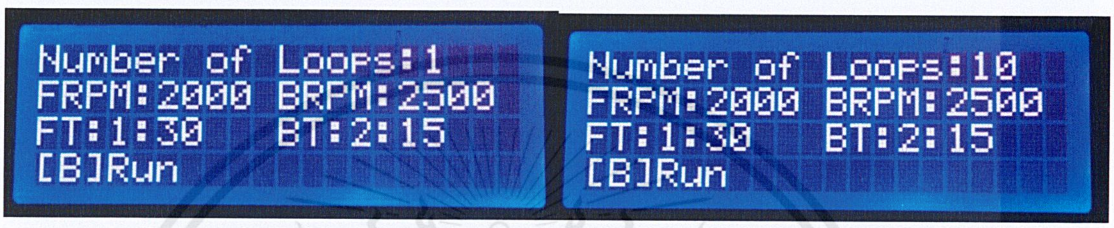

Figure 20. Setting window of Auto reverse mode

(3.2) When entered number of loops, press "B" to start working. While working, the agitator will display Setting Variable of both direction on both side of LCD screen. The Process Variable is shown below of LCD screen. In that part of the timer, it will countdown the time as set in each direction. When complete, the agitator will automatically reverse to other direction. After fully rotated in both directions (1 loop) The number of loops will increase until its equal to the number of loops that was set, the agitator will stop working automatically.

Figure 21. Working window of Auto reverse mode

(4) Additional functions

(4.1) "#" : Can press the "#" button as a back in setting window and working window

(4.2) "*" : While working, can press the "*" button to pause working. The agitator will pause working and countdown. Press the "*" button again if need to continue working, the agitator will continue working and continue countdown time

(4.3) "C" : While working, can press the "C" button to stop working. The agitator will stop working and countdown. If stop, the agitator can't continue working, must go back to step (2.3) or (3.2) to re-set working.

(4.4) If starting working but the agitator must overload until the motor does not rotate more than 10 seconds, the agitator will stop working automatically to prevent burning of motor.

4.2 Test working

4.2.1 Speed

The maximum speed of agitator can be calculated from the maximum speed of the motor and the Pulley system from the equation

$$d_1 \cdot n_1 = d_2 \cdot n_2$$

$$n_{2,max} = \frac{d_1 \cdot n_{1,max}}{d_2}$$

$$n_{2,max} = \frac{(2.1\ cm) \cdot (6{,}000\ RPM)}{3.1\ cm}$$

$$n_{2,max} = 4{,}064\ RPM$$

The real maximum speed of agitator can be calculated from the calibration step. When running the calibration step for 60 seconds will receive calibration factor which can calculated the real maximum speed of agitator from

$$RPM_{max} = \frac{Calibration\ factor}{Numbers\ of\ propeller}$$

$$RPM_{max} = \frac{27{,}576}{8}$$

$$RPM_{max} = 3{,}447\ RPM$$

Which makes finding the efficiency of the agitator from

$$\eta = \frac{RPM_{max}}{n_{2,max}} \cdot 100$$

$$\eta = \frac{3{,}447}{4{,}064} \cdot 100$$

$$\eta = 0.8482 = 84.82\ \%$$

4.2.2 torque

The torque of the agitator is equal to the torque of the motor but after passing through the Pulley system, agitator efficiency was decrease. Therefore, the torque value of the agitator is

$$\tau_r = \tau \cdot \eta$$

$$\tau_r = 0.20 \cdot 0.8482$$

$$\tau_r = 0.1696\ Nm$$

4.3 Assessment

Assessment is an important part of this project because it will be a conclusion that can reduce the cost in the mixing process or the processes that have to use agitator. Which will evaluate the price from all equipment used in the production of agitator. The price of all equipment that use was show in table 2.

Table 2: Price of all equipment

Equipment	**Price (THB)**
Arduino Uno R3	160
Module L298N	115
FC-33	50
20x4 LCD With IC2	215
4x4 matrix keypad	35
Active Buzzer	7
XD-3420 24V Motor & Pulley system	2000
24V 3A Power supply	222
12V 2A Adapter	67
etc.	629
Summation	**3500**

From the table 2 shows that the price of equipment used to build the machine is very inexpensive. Can create a machine for only 3500 baht or 86% lower than buy the prototype agitator

4.4 Compare performance

When able to produce a new agitator, it was found that the performance of the newly created agitator with the prototype agitator was not very different. By showing the comparison according to the table

Table 3: Compare performance

Specification	Prototype	Project
Stirring capacity (H_2O)	20L	20L
Motor (output)	Brushless DC motor 60W	DC motor 48W
Speed range	80~2,000 rpm	200~3,000 rpm
Torque	0.28 Nm	0.17 Nm
Speed controller	Feedback control	Feedback control
Speed display	Digital	Digital
Rotations	1. Forward 2. Backward 3. Auto reverse	1. Forward 2. Backward 3. Auto reverse
Timer	99 hour 59 min	166 hour 40 min
Electrical supply	220 V 50/60 Hz	220 V 50/60 Hz
Price	25,000 THB	3,500 THB

From the table, it can be seen that in the part of the work can be done similarly, there are 3 modes, stir in the same maximum volume tank, Feedback Control and using the power supply equally. The new agitator has torque less than prototype agitator, or can be said that can use to stir liquid with viscosity lower than prototype agitator but can achieve a speed of up to about 2000 RPM, can set a longer time stop and most importantly, the costs decreased by approximately 80%.

CHAPTER V
CONCLUSION

5.1 Conclusion

This project will produce laboratory agitator to decrease the cost of the mixing process or other process that use agitator by using Arduino to control motor. This agitator can work on 3 modes, stir in 20L maximum volume tank, Feedback Control detect speed with maximum speed 4000 RPM, maximum timer 166 hours 40 minutes and maximum torque 0.19684 Nm, which is considered to work similarly to the prototype. The difference is the equipment used (motor and controller) by using devices that are lower in price, but provide similar performance. After production of laboratory agitator is complete, it is found that the production costs decrease by approximately 80%.

5.2 Suggestion

- The higher powerful motor will be used in order to generate higher maximum torque.
- The use of motor will be easier for user by developing a new code.

Reference

[1] Environmental Impact Assessment.; Domingo G. Orea, Teresa G. Villarino.: Madrid, 2013.

[2] Handbook of Industrial Mixing: Science and Practice.; Edward L. Paul, Victor A. Atiemo-Obeng, Suzanne M. Kresta.: 2004.

[3] Industrial agitator. Wikipedia. https://en.wikipedia.org/wiki/Industrial_agitator (accessed April 9, 2019).

[4] Agitator for chemical process. Hexon Engineering co., ltd. http://www.hexonengineering.com/agitator.php

[5] Microcontroller. Manufacture Overhaul Rapid and Optimal Co., Ltd. http://www.moro.co.th/ไมโครคอนโทรลเลอร์(accessed July 24, 2018).

[6] Arduino Basic [EP0]: Arduino คืออะไร? ทำอะไรได้? มีกี่แบบ? Arduinospro. https://www.arduinospro.com/article/7/arduino-basic-ep0-arduino-คืออะไร-ทำอะไรได้-มีกี่แบบ (accessed Oct 21, 2017).

[7] Mixing (process engineering). Wikipedia. https://en.wikipedia.org/wiki/Mixing_(process_engineering)#cite_note-ullmann-1 (accessed April 21, 2019).

[8] Ullmann, Fritz. Ullmann's Chemical Engineering and Plant Design, Volumes 1–2.; John Wiley & Sons: 2005. http://app.knovel.com/hotlink/toc/id:kpUCEPDV02/ullmanns-chemical-engineering

[9] Mixing design. PowderProcess. https://powderprocess.net/Mixing/Comparison_Continuous_Batch_Mixing.html (accessed Sep 28, 2017).

[10] Multi-Shaft Mixer. Hockmeyer. http://www.hockmeyer.com/products/high-viscosity-mixers/dual-triple-shaft-mixer-detail.html (accessed June 23, 2017).

APPENDIX

Appendix A: Calibration command code

```
#include <Wire.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C  lcd(0x27,2,1,0,4,5,6,7);
#include <Keypad.h>

byte sensorInterrupt = 0;  // 0 = digital pin 2
byte sensorPin  = 2;
float calibrationFactor = 20;
volatile byte pulseCount;
float flowRate;
unsigned int flowMilliLitres;
unsigned long totalMilliLitres;
unsigned long oldTime;
int rpm;
int readp;

int L_EN1=8;
int R_EN1=9;
int pwm_pin = 3;
int m_speed=255;

const int buttonPin[2]={0,2};
int buttonState[2];
int lastButtonState[2];

unsigned long previousMillis = 0;
static const long interval = 500;

const byte ROWS = 4; //four rows
const byte COLS = 4; //three columns
```

```
char keys[ROWS][COLS] =
{
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {4, 5, 6, 7}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {10, 11, 12, 13}; //connect to the column pinouts of the keypad
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
char key;
String command;

int t;
void setup()
{
  Serial.begin(9600);
  lcd.begin (20,4);
  lcd.setBacklightPin(3,POSITIVE);
  lcd.setBacklight(HIGH);
  pinMode(2,INPUT);
  pinMode(L_EN1,OUTPUT);
  pinMode(R_EN1,OUTPUT);
  pinMode(pwm_pin,OUTPUT);
  attachInterrupt(sensorInterrupt, pulseCounter, FALLING);
}

void loop()
{
  key = keypad.getKey();
  if (key)
  {
    Serial.println(key);
  }
```

```
unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= interval)
{
  previousMillis = currentMillis;

  lcd.setCursor(0, 0);
  lcd.print(" RPM:");
  lcd.print(rpm*60);
  lcd.print("  ");
  lcd.setCursor(0, 1);
  lcd.print("PWM:");
  lcd.print(m_speed);
  lcd.print("  ");

  lcd.setCursor(0, 2);
  lcd.print(" Count:");
  lcd.print(readp);
  lcd.print("  ");
  lcd.setCursor(0, 3);
  lcd.print("Time:");
  lcd.print(t);
  lcd.print("  ");

  if(m_speed<=254)
  {
    m_speed+=1;
  }
}

comp("up");
if(t<=59)
{
  read_flow();
}
```

```
}

void comp(String cmd)
{
  if(cmd=="stop")
  {
    analogWrite(pwm_pin,0);
    digitalWrite(L_EN1, LOW);
    digitalWrite(R_EN1, LOW);
    // Serial.println("Motor 2 Stop");
  }
  else if(cmd=="up")
  {
    analogWrite(pwm_pin,m_speed);
    digitalWrite(L_EN1, HIGH);
    digitalWrite(R_EN1, LOW);
    //Serial.println("Motor 2 Forward");
  }
  else if(cmd=="dn")
  {
    analogWrite(pwm_pin,m_speed);
    digitalWrite(L_EN1, LOW);
    digitalWrite(R_EN1, HIGH);
    //Serial.println("Motor 2 Reverse");
  }
}

void read_flow()
{
  if((millis() - oldTime) > 1000)
  {
    detachInterrupt(sensorInterrupt);
    flowRate = ((1000.0 / (millis() - oldTime)) * pulseCount) / calibrationFactor;
    oldTime = millis();
```

```
        flowMilliLitres = (flowRate / 60) * 1000;
        totalMilliLitres += flowMilliLitres;
        unsigned int frac;
        rpm=flowRate;
        t+=1;
        pulseCount = 0;
        attachInterrupt(sensorInterrupt, pulseCounter, FALLING);
    }
}

void pulseCounter()
{
    if(t<=59)
    {
        readp++;
    }
    pulseCount++;
}
```

Appendix B: Working command code

```
#include <Wire.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C  lcd(0x27,2,1,0,4,5,6,7); // 0x27 or 0x3F
#include <Keypad.h>

byte sensorInterrupt = 0;  // 0 = digital pin 2
byte sensorPin       = 2;
float calibrationFactor = 1; //Number of blade
volatile byte pulseCount;
unsigned long oldTime;
int rpm;

int L_EN1=8;
int R_EN1=9;
int pwm_pin = 3;
int m_speed;

int rpm_set[4];
int sec_set[4];
int sec_count[4];
int min_set[4];
int min_count[4];
int menu;
int sub_menu=1;
int f_active;
int b_active;
int err_count;

int m_max_speed=7000; //From calibrate code
int mix_loop=1;
int loop_count;
```

```
const int buttonPin[2]={0,2};
int buttonState[2];
int lastButtonState[2];

unsigned long previousMillis = 0;
static const long interval = 1000;

const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] =
{
   {'1','2','3','A'},
   {'4','5','6','B'},
   {'7','8','9','C'},
   {'*','0','#','D'}
};
byte rowPins[ROWS] = {4, 5, 6, 7};
byte colPins[COLS] = {10, 11, 12, 13};
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
char key;
String command;

void setup()
{
   Serial.begin(9600);
   lcd.begin (20,4);
   lcd.setBacklightPin(3,POSITIVE);
   lcd.setBacklight(HIGH);
   pinMode(2,INPUT);
   pinMode(L_EN1,OUTPUT);
   pinMode(R_EN1,OUTPUT);
   pinMode(pwm_pin,OUTPUT);
   attachInterrupt(sensorInterrupt, pulseCounter, FALLING);
```

```
}

void loop()
{
  key = keypad.getKey();
  if (key)
  {
    tone(A0,2000,20);
    Serial.println(key);
  }
  if(key=='1' || key=='2' || key=='3' || key=='4' || key=='5' || key=='6' || key=='7' ||
key=='8' || key=='9' || key=='0')
  {
    command += key;
    if(command.length()>4)
    {
      command="";
      lcd.clear();
    }
  }
  if(key=='1' && menu==0)
  {
    menu=1;
    command="";
    lcd.clear();
  }
  if(key=='2' && menu==0)
  {
    menu=2;
    command="";
    lcd.clear();
  }
  if(key=='3' && menu==0)
  {
```

```
    menu=3;
    command="";
    lcd.clear();
  }
  if(key=='*' && menu==1)
  {
    sub_menu+=1;
    if(sub_menu>3)
    {
      sub_menu=1;
    }
   lcd.clear();
  }
  if(key=='*' && menu==2)
  {
    sub_menu+=1;
    if(sub_menu>3)
    {
      sub_menu=1;
    }
   lcd.clear();
  }
  if(key=='A' && menu==1 && sub_menu==1)
  {
    rpm_set[1]=command.toInt();
    if(rpm_set[1]>=7000)
    {
      rpm_set[1]=7000;
    }
    command="";
    lcd.clear();
  }
  if(key=='A' && menu==1 && sub_menu==2)
  {
```

```
    min_set[1]=command.toInt();
    command="";
    lcd.clear();
  }
  if(key=='A' && menu==1 && sub_menu==3)
  {
    sec_set[1]=command.toInt();
    if(sec_set[1]>=60)
    {
      sec_set[1]=60;
    }
    command="";
    lcd.clear();
  }
  if(key=='A' && menu==2 && sub_menu==1)
  {
    rpm_set[2]=command.toInt();
    if(rpm_set[2]>=7000)
    {
      rpm_set[2]=7000;
    }
    command="";
    lcd.clear();
  }
  if(key=='A' && menu==2 && sub_menu==2)
  {
    min_set[2]=command.toInt();
    command="";
    lcd.clear();
  }
  if(key=='A' && menu==2 && sub_menu==3)
  {
    sec_set[2]=command.toInt();
    if(sec_set[2]>=60)
```

```
    {
      sec_set[2]=60;
    }
    command="";
    lcd.clear();
  }
  if(key=='A' && menu==3)
  {
    mix_loop+=1;
    command="";
    lcd.clear();
  }
  if(key=='D' && menu==3)
  {
    if(mix_loop>=2)
    {
      mix_loop-=1;
    }
    command="";
    lcd.clear();
  }
  if(key=='B' && menu==1)
  {
    m_speed=map(rpm_set[1],0,m_max_speed,0,255);
    Serial.print("PWM1:");
    Serial.println(m_speed);
    sec_count[1]=sec_set[1];
    min_count[1]=min_set[1];
    com("up");
    menu=4;
    f_active=1;
    command="";
    lcd.clear();
  }
```

```
if(key=='*' && menu==4)
{
   if(f_active==1)
   {
      f_active=0;
      com("stop");
   }
   else
   {
      f_active=1;
      com("up");
   }
   command="";
   lcd.clear();
}
if(key=='C' && menu==4)
{
   f_active=0;
   com("stop");
   menu=1;
   sub_menu=1;
   command="";
   lcd.clear();
}
if(key=='B' && menu==2)
{
   m_speed=map(rpm_set[2],0,m_max_speed,0,255);
   Serial.print("PWM2:");
   Serial.println(m_speed);
   sec_count[2]=sec_set[2];
   min_count[2]=min_set[2];
   com("dn");
   menu=5;
   f_active=2;
```

```
    command="";
    lcd.clear();
  }
  if(key=='*' && menu==5)
  {
    if(f_active==2)
    {
      f_active=0;
      com("stop");
    }
    else
    {
      f_active=2;
      com("dn");
    }
    command="";
    lcd.clear();
  }
  if(key=='C' && menu==5)
  {
    f_active=0;
    com("stop");
    menu=2;
    sub_menu=1;
    command="";
    lcd.clear();
  }
  if(key=='B' && menu==3)
  {
    m_speed=map(rpm_set[1],0,m_max_speed,0,255);
    Serial.print("PWM2:");
    Serial.println(m_speed);
    sec_count[1]=sec_set[1];
    min_count[1]=min_set[1];
```

```
    sec_count[2]=sec_set[2];
    min_count[2]=min_set[2];
    loop_count=0;
    com("up");
    menu=6;
    f_active=1;
    loop_count=1;
    command="";
    lcd.clear();
}
if(key=='C' && menu==6)
{
    f_active=0;
    com("stop");
    menu=3;
    command="";
    lcd.clear();
}
if(key=='D')
{
    command="";
    lcd.clear();
}
if(key=='#' && menu<=3)
{
    menu=0;
    sub_menu=1;
    command="";
    lcd.clear();
}
analogWrite(pwm_pin,m_speed);
unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= interval)
{
```

```
previousMillis = currentMillis;
if(f_active!=0 && (rpm*60)<100)
{
   err_count+=1;
}
else
{
   err_count=0;
}
if(err_count==5)
{
   com("stop");
   tone(A0,2000,100);
   sec_count[1]=0;
   sec_count[2]=0;
   min_count[1]=0;
   min_count[2]=0;
   f_active=0;
}
if(f_active==1 && menu==4)
{
   if(sec_count[1]>=1 && min_count[1]>=0)
   {
      sec_count[1]-=1;
   }
   if(sec_count[1]==0 && min_count[1]>=1)
   {
     min_count[1]-=1;
     sec_count[1]=60;
   }
   if(sec_count[1]==0 && min_count[1]==0 &&menu==4)
   {
      f_active=0;
      com("stop");
```

```
    }
  }
  if(f_active==2)
  {
    if(sec_count[2]>=1 && min_count[2]>=0)
    {
      sec_count[2]-=1;
    }
    if(sec_count[2]==0 && min_count[2]>=1)
    {
      min_count[2]-=1;
      sec_count[2]=60;
    }
    if(sec_count[2]==0 && min_count[2]==0)
    {
      f_active=0;
      com("stop");
      if(loop_count<mix_loop)
      {
        m_speed=map(rpm_set[1],0,m_max_speed,0,255);
        Serial.print("PWM2:");
        Serial.println(m_speed);
        sec_count[1]=sec_set[1];
        min_count[1]=min_set[1];
        sec_count[2]=sec_set[2];
        min_count[2]=min_set[2];
        com("up");
        menu=6;
        f_active=1;
        loop_count+=1;
        command="";
        lcd.clear();
      }
    }
```

```
    }
    if(f_active==1 && menu==6)
    {
       if(sec_count[1]>=1 && min_count[1]>=0)
       {
          sec_count[1]-=1;
       }
       if(sec_count[1]>=0 && min_count[1]>=1)
       {
          min_count[1]-=1;
          sec_count[1]=60;
       }
       if(sec_count[1]==0 && min_count[1]==0)
       {
          f_active=0;
          com("stop");
          tone(A0,2000,100);
         m_speed=map(rpm_set[2],0,m_max_speed,0,255);
         Serial.print("PWM2:");
          Serial.println(m_speed);
          sec_count[2]=sec_set[2];
          min_count[2]=min_set[2];
          com("dn");
          menu=6;
          f_active=2;
       }
    }
 }
 lcd.setCursor(10, 0);
 lcd.print(command);

 if(menu==0)
 {
    lcd.setCursor(0, 0);
```

```
    lcd.print("Select Mode");
    lcd.setCursor(0, 1);
    lcd.print("1.Forward");
    lcd.setCursor(0, 2);
    lcd.print("2.Backward");
    lcd.setCursor(0, 3);
    lcd.print("3.Auto reverse");
}
else if(menu==1)
{
    lcd.setCursor(0, sub_menu);
    lcd.print("->");
    lcd.setCursor(0, 0);
    lcd.print("Forward");
    lcd.setCursor(2, 1);
    lcd.print("RPM:");
    lcd.print(rpm_set[1]);
    lcd.setCursor(2, 2);
    lcd.print("Min:");
    lcd.print(min_set[1]);
    lcd.setCursor(2, 3);
    lcd.print("Sec:");
    lcd.print(sec_set[1]);
    lcd.setCursor(12, 3);
    lcd.print("[B]Run");
}
else if(menu==2)
{
    lcd.setCursor(0, sub_menu);
    lcd.print("->");
    lcd.setCursor(0, 0);
    lcd.print("Backward");
    lcd.setCursor(2, 1);
    lcd.print("RPM:");
```

```
    lcd.print(rpm_set[2]);
    lcd.setCursor(2, 2);
    lcd.print("Min:");
    lcd.print(min_set[2]);
    lcd.setCursor(2, 3);
    lcd.print("Sec:");
    lcd.print(sec_set[2]);
    lcd.setCursor(12, 3);
    lcd.print("[B]Run");
}
else if(menu==3)
{
    lcd.setCursor(0, 0);
    lcd.print("Number of Loops:");
    lcd.print(mix_loop);
    lcd.setCursor(0, 1);
    lcd.print("FRPM:");
    lcd.print(rpm_set[1]);
    lcd.setCursor(10, 1);
    lcd.print("BRPM:");
    lcd.print(rpm_set[2]);
    lcd.setCursor(0, 2);
    lcd.print("FT:");
    lcd.print(min_set[1]);
    lcd.print(":");
    lcd.print(sec_set[1]);
    lcd.setCursor(10, 2);
    lcd.print("BT:");
    lcd.print(min_set[2]);
    lcd.print(":");
    lcd.print(sec_set[2]);
    lcd.setCursor(0, 3);
    lcd.print("[B]Run");
}
```

```
else if(menu==4)
{
   lcd.setCursor(0, 0);
   lcd.print("Forward");
   lcd.setCursor(0, 1);
   lcd.print("Time: ");
   lcd.print(min_count[1]);
   lcd.print(":");
   lcd.print(sec_count[1]);
   lcd.print("[");
   lcd.print(min_set[1]);
   lcd.print(":");
   lcd.print(sec_set[1]);
   lcd.print("]");
   lcd.print("  ");
   lcd.setCursor(0, 2);
   lcd.print("PV: ");
   lcd.print(rpm*60);
   lcd.print("   ");
   lcd.setCursor(0, 3);
   lcd.print("SV: ");
   lcd.print(rpm_set[1]);
}
else if(menu==5)
{
   lcd.setCursor(0, 0);
   lcd.print("Backward");
   lcd.setCursor(0, 1);
   lcd.print("Time: ");
   lcd.print(min_count[2]);
   lcd.print(":");
   lcd.print(sec_count[2]);
   lcd.print("[");
   lcd.print(min_set[2]);
```

```
        lcd.print(":");
        lcd.print(sec_set[2]);
        lcd.print("]");
        lcd.print("  ");
        lcd.setCursor(0, 2);
        lcd.print("PV: ");
        lcd.print(rpm*60);
        lcd.print("   ");
        lcd.setCursor(0, 3);
        lcd.print("SV: ");
        lcd.print(rpm_set[2]);
    }
    else if(menu==6)
    {
        lcd.setCursor(0, 0);
        lcd.print("Loops:");
        lcd.print(loop_count);
        lcd.print("/");
        lcd.print(mix_loop);
        lcd.setCursor(0, 1);
        lcd.print("FRPM:");
        lcd.print(rpm_set[1]);
        lcd.setCursor(10, 1);
        lcd.print("BRPM:");
        lcd.print(rpm_set[2]);
        lcd.setCursor(0, 2);
        lcd.print("FT:");
        lcd.print(min_count[1]);
        lcd.print(":");
        lcd.print(sec_count[1]);
        //lcd.print(" ");
        lcd.setCursor(10, 2);
        lcd.print("BT:");
        lcd.print(min_count[2]);
```

```
        lcd.print(":");
        lcd.print(sec_count[2]);
        //lcd.print(" ");
        lcd.setCursor(0, 3);
        lcd.print("PV: ");
        lcd.print(rpm*60);
        lcd.print("    ");
    }
    read_flow();
}

void com(String cmd)
{
    if(cmd=="stop")
    {
        analogWrite(pwm_pin,0);
        digitalWrite(L_EN1, LOW);
        digitalWrite(R_EN1, LOW);
        Serial.println("Motor Stop");
    }
    else if(cmd=="up")
    {
        analogWrite(pwm_pin,m_speed);
        digitalWrite(L_EN1, HIGH);
        digitalWrite(R_EN1, LOW);
        Serial.println("Motor Forward");
    }
    else if(cmd=="dn")
    {
        analogWrite(pwm_pin,m_speed);
        digitalWrite(L_EN1, LOW);
        digitalWrite(R_EN1, HIGH);
        Serial.println("Motor Reverse");
    }
```

```
}

void read_flow()
{
  if((millis() - oldTime) > 1000)
  {
    detachInterrupt(sensorInterrupt);
    rpm = ((1000.0 / (millis() - oldTime)) * pulseCount) / calibrationFactor;
    oldTime = millis();
    pulseCount = 0;
    attachInterrupt(sensorInterrupt, pulseCounter, FALLING);
  }
}

void pulseCounter()
{
  pulseCount++;
}
```

BIBLIOGRAHPY

Name: Tanawat Musika

Date of Birth: 11/08/1996

Address: 152 Moo 5, Kampangzao Sub-district, Muang District, Nakhon Sri Thammarat Province, Thailand 80280

E-mail: 58010516@kmitl.ac.th, pond16-tr@hotmail.com

Academic Background: Bachelor degree of Petrochemical Engineering

Working Experience: Internship with Khanom Gas Separation Plant, PTT Public Company Limited