Journal of Electronics, Electromedical Engineering, and Medical Informatics (JEEEMI)
Multidisciplinary : Rapid Review : Open Access Journal **eISSN: 2656-8632**

RESEARCH ARTICLE **OPEN ACCESS**

Manuscript received January 18, 2022; revised February 03, 2022; accepted February 20, 2022; date of publication April 29, 2022
Digital Object Identifier (DOI): https://doi.org/10.35882/jeeemi.v4i2.6

Comparison of Air Pressure Control Between Discrete and PID Control Applied in the Calibration Process in Blood Pressure Meter

Harisha Avin Nurcahyana[1], Tri Bowo Indrato[1], Triana Rahmawati[1], and Wahyu Caesarendra[2]
[1] Department of Electrical Engineering Poltekkes Ministry of Health, Pucang Jajar Timur Num. 10, Surabaya, 60245, Indonesia
[2] Faculty of Integrated Technologies, Universiti Brunei Darussalam, Brunei Darussalam

Corresponding author: Harisha Avin Nurcahyana (email: harishaavin06@gmail.com)

ABSTRACT In performing the calibration of the sphygmomanometer, the officer needs to first reset the installation and pump the bulb slowly until it reaches the set point in accordance with the calibration settings where this does not provide convenience to the calibration officer. So the author wants to do research on making additional devices to support DPM calibration instruments that have been commercialized to speed up the pump process in Sphygmomanometer calibration. The purpose of this research is to make an Automatic Pump module with PID control to analyze the stability of the pressure achievement in accordance with the set point when using the smoothing program or not. This study used set points of 50, 100, 150, 200, and 250 mmHg. Data retrieval was carried out within 260 seconds at each set point at the Campus of the Department of Electrical Engineering Poltekkes Kemenkes Surabaya. The results of this study indicate that the tool testing using the smoothing program experienced small oscillations compared to the program without smoothing. The data obtained are at setting 50 the average overshoot is 54 and the average undershoot is 49; at setting 100 the average overshoot is 109 and the average undershoot is 99; at setting 150 the average overshoot is 156 and the average undershoot is 149; at setting 200 the average overshoot is 206 and the average undershoot is 196; at setting 250 the average overshoot is 253 and the average undershoot is 247. The importance of this device was made in order to make it easier and faster for the calibration officer to calibrate the Sphygmomanometer.

INDEX TERMS Sphygmomanometer; Calibration; PID; Pump, DPM

I. INTRODUCTION

Sphygmomanometer consists of two words, namely Sphygmo (Greek) which means knock and manometer which means pressure gauge[1]. Measurement of blood pressure is an important component of the general examination of each patient and The sphygmomanometer is one of the most commonly used diagnostic medical devices [2]. Sphygmomanometer is a medical device used by medical personnel to determine the level of blood pressure in patients. Sphygmomanometer is a tool that is often used by health workers to perform examinations on patients [3]. Sphygmomanometers are divided into two types: conventional and digital [4]. The high intensity of the use of the Sphygmomanometer causes a decrease in the quality of the tool, so that the accuracy of the Sphygmomanometer decreases. The medical devices used are of doubtful accuracy and have the potential to threaten the safety of patients as recipients of health services that use them (Susana et al., 2020). To find out the condition of a tool, it is necessary to check the tool, especially the calibration process of the tool on the Sphygmomanometer. Blood pressure is the pressure exerted by circulating blood on the walls of the arteries [5]. Calibration is a technical activity consisting of determination, determination of one or more properties and characteristics of a product, process or service in accordance with a specific procedure that has been determined and Calibration is an observation to determine the correctness of the designation of measuring instruments and or measuring materials [6]. Calibration tests performed on medical devices are useful to determine the condition of the equipment and measure the feasibility of medical devices. The calibration test was carried out on the Sphygmomanometer using a Digital Pressure Meter (DPM) to determine the amount of pressure generated by the Sphygmomanometer. Digital pressure meter (DPM) is a calibration tool used to determine pressure in mmHg. DPM changes the data from the pressure sensor into the form of numbers that are displayed on the display (Wibisono, 2017). Calibration is a calibration activity to determine the correct value of the designation of measuring instruments and/or measuring materials (PERMENKES, 2015). To calibrate the Sphygmomanometer, the officer needs to set up the installation in advance and pump the bulb slowly and stop at every point of air pressure that has been set in the Sphygmomanometer calibration [7].

However, in calibrating there is a process where the officer must pump and ensure the pressure in accordance with the Sphygmomanometer calibration rules where it does not provide convenience to the calibration officer. Based on the literature that has been read has never been done making a device to support the calibration process to be more optimal. So the authors wanted to do research on making enhancements to support commercialized DPM calibration instruments to speed up the pump process in Sphygmomanometer calibration. The importance of this device is made in order to facilitate and speed up calibration officers in calibrating the Sphygmomanometer. It is expected that officers will not have to bother to set the pressure on the Sphygmomanometer to be equal to the set point in the Sphygmomanometer calibration regulations [8].

Chung-Hsien et al, used the PID control system for cylinder motor control on the sphygmomanometer. According to experimental results on healthy subjects, AIM reduced 34.21% and ADM reduced 15.78% measurement time when compared to commercial BP monitors. In the future, system control parameters are optimized by taking into account characteristic models, such as single-acting cylinder and cuff systems. Thus, the planning and efficiency of the control system can be further improved. In addition, the announcement of the IEC EN 1060-4 standard will be the next work plan [9]. Ihan et al, used the curve fitting method with 2 different approaches, namely regression analysis and interpolation. In this study, it was revealed that the biggest weakness of the automatic blood pressure measuring device that uses the oscillator method is that this tool measures the DBP value with lower accuracy than the SBP value. This is because the detection of the point where the cuff pressure oscillations begin to disappear is more difficult than the point where the oscillations begin to appear [5]. Debralee Nelson et al, our study found that the automatic arm and hand unit and aneroid sphygmomanometer used in the study did not meet the standards set by BHS and AAMI. So there is evidence that more research is needed before mercury measurements are officially replaced with other ones [10]. However, Hamied et al stated that digital and aneroid sphygmomanometer can be used to utilize mercury sphygmomanometer in a community environment in Bandung [5]. Parati giantranco explained that his research on most of this is based on the oscillometric method for measuring blood pressure. the rate of cuff deflation may differ in different devices, thereby introducing another possible source of error in automated blood pressure measurements. All of these problems can cause significant differences in the blood pressure values provided by automatic oscillometric devices manufactured by different manufacturers. When calibrated with a mercury sphygmomanometer, a mean difference of 3 mmHg was considered acceptable; however, 58% of aneroid sphygmomanometers have been shown to have errors greater than 4 mmHg, with about one third of these errors being higher than 7 mmHg [10]. Kim Won Ki et al, described in their research that the proposed double bladder-based cuff system improves the accuracy of oscillometric blood pressure measurements. The developed system reduces the range of errors about systolic and about diastolic compared to sphygmomanometers developed commercially before 2006[11]. Fita FR et al, made DPM using 2 sensors to measure in mmhg and Kpa units. In this study, the results obtained an average error measurement of 7.3 mmHg for sphygmomanometer measurements, and for suction pumps less than 1.5 Kpa. The weakness of the module that is made is that the power on the battery runs out quickly [12]. A de Greeff et al, investigated the accuracy of calibration of non-invasive blood pressure measuring devices in hospitals. Of the 22 different device models have been identified. 12 of these were automatic tensimeters and only four were validated evidence according to recognized protocols (BHS, AAMI or European Hypertension Society International Protocol). From this incident, it is important to periodically calibrate so as not to read [6], [13]. Digital Pressure Meter (DPM) has a concept such as a digital blood pressure meter that uses an integrated pressure sensor, analog signal conditioning circuit, and a microcontroller [14]. In 2016 Novella Lasdrei Anna L. carried out a project entitled Digital Pressure Gauge equipped with a PC-Based Thermohydrometer which aims to facilitate users and conduct research on the calibration process, while also reducing the possibility of errors in data entry. this tool already has a fairly small error value but this tool is still less efficient to be used as a digital pressure gauge [15]. In 2019 Ketut Dyah Kusumadewi conducted a research entitled Two Mode DPM Equipped with a Thermohydrometer and Positive Pressure. This tool has been equipped with negative pressure measurement and unit selection at positive pressure [3]. In addition, there are also patents devoted to the level of accuracy of the Sphygmomanometer [16]. In 2001 T. Knight et al, designed the study i.e. Sphygmomanometer used in general practice: neglected quality aspect Inpatient, to audit the condition of mercury and aneroid sphygmomanometer used in general practice in relation to accuracy [17]. in 2016 Bhaskar Shahbabu et al, conducted a study comparing which one is more accurate in measuring blood pressure? Digital or aneroid sphygmomanometer, this study was conducted to compare the accuracy of aneroid readings and a referenced digital sphygmomanometer to a mercury sphygmomanometer and determine an agreement on the classification of hypertension between mercury and non-mercury devices [18]. in 2014 B.

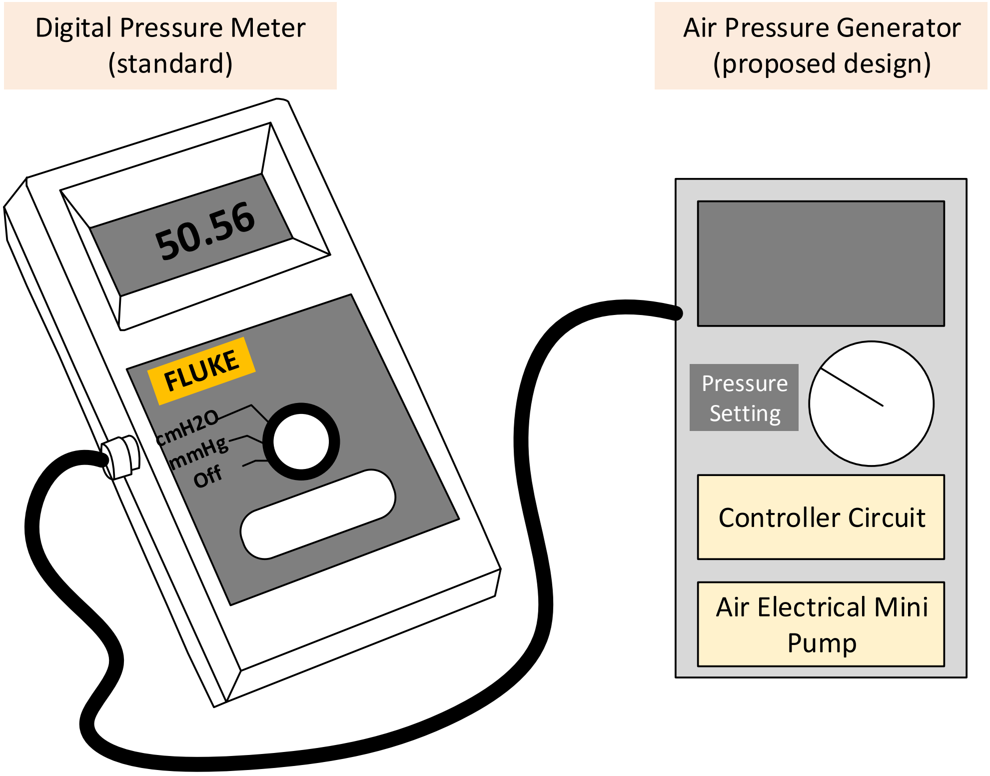


FIGURE 1. Precision air pressure generator to calibrate the digital pressure meter. The digital pressure meter is used to measure the conventional or digital sphygmomanometer. Furthermore, the air pressure generator is used to calibrate the digital pressure meter.

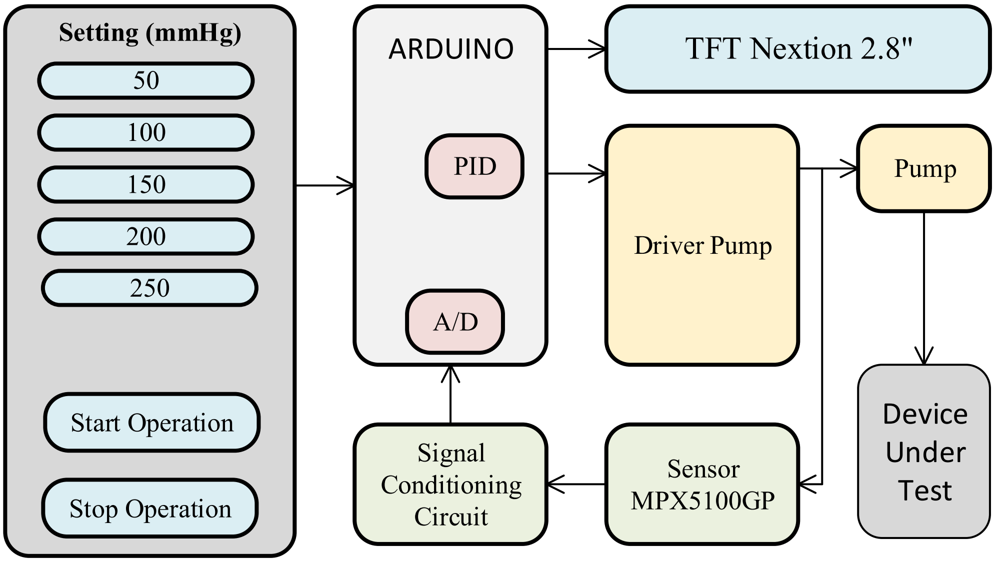


FIGURE 2. The proposed design of precision air pressure generator to calibrate the digital pressure meter.

Mustapha et al. make research on PC-Based Blood Pressure Meters for E-health Applications in this study discusses the development of PC-based blood pressure meters for e-health applications that may be suitable for doctors or doctors used in hospitals, clinics and other medical institutions [19]. Houde Dai et al, conducted a study on Wireless Interface Continuous Blood Pressure Monitor, in this study advanced an outpatient digital wrist BP monitor that can be used for long-term monitoring and wireless data acquisition [20]. Zhang Jin-ling at all, design electronic blood pressure monitoring system based on cellular telemedicine system, in which the study blood pressure monitoring tool has: design and presented for blood cuffs non-invasive pressure monitoring, and blood pressure algorithm is oslometric method [21]. Md. Manirul Islami et al, designed a study that is Development of a noninvasive continuous blood pressure measurement and monitoring system, which uses invasive principles for measuring blood pressure, which is difficult for patients. So in this study we have designed a non-invasive continuous blood pressure measurement and monitoring system [22].

The purpose of this research is to make an Automatic Pump module with PID control to analyze the stability of the pressure achievement in accordance with the set point when using the smoothing program or not.

II. MATERIALS AND METHOD

A. SETTING TEST

This study used a PID system. The data collection is done in ±260 seconds at each set point. In data collection the prototype is juxtaposed with DPM and the data is recorded using Delphi.

1) MATERIALS AND TOOLS

The study used the MPX5100GP sensor as a pressure sensor to detect the pressure exerted by the pump motor. MPX 5100 sensor is a pressure sensor with temperature compensation, signal conditioning and has been calibrated [10]. Microcontrollers used are Arduino UNO to process data and send data to LCD TFT Nextion LCD. Arduino Uno microcontroller is used to process analog data into digital data that is displayed on the TFT LCD [23]. The TFT device presented by Adafruit is one of the most inexpensive and easy to find devices for purchase, both national and international, easily accessible nationally and at a very affordable price, this device is the most common for use with projects for Arduino, or for the internet of things [24]. TFT Nextion LCD serves as a display to display the measurement, calibration process [25]. and Delphi for data recording.

2) EXPERIMENT

In this study after the design was completed, mpx5100GP pressure sensor response testing was conducted to detect the pressure value given by the pump. Tool testing is done with smoothing programs and without smoothing programs at each set point. The set points used are 50, 100, 150, 200, and 250 mmHg. Then the pump motor will pump the cuff until it reaches the test point and maintain the pump pressure at the set point specified by the PID control method. The graph and the pressure values provided will be displayed on the TFT and Delphi LCD displays to store the data values. Data obtained from the Automatic Pump module will be analyzed with standard deviation and error values.

B. BLOCK DIAGRAM

Changes in pressure on the cuff will make the MPX5100GP sensor produce a voltage change in the output pin (FIGURE 1). The signal conditioning circuit serves to condition the signal so that the voltage changes leading to the microcontroller range from 0-5Vdc. In this system the value on the sensor must show a value of 0, so it needs to be conditioned through a series of summing to determine the value of zero. The output of the signal conditioning circuit is converted to an ADC value and processed into a tension pressure value with mmHg units through a program planted in Arduino Uno. PID control is implanted into the microcontroller to set the pump to accelerate the pressure according to the set point. The pressure (mmhg) settings button is pressed according to will and forwarded by pressing the start button then the system will run. The system is marked with the pump on according to the PID system control program that has been planted. The pressure on the cuff is displayed on the TFT display (FIGURE 2).

C. FLOW CHART

Program system diagrams are built on flowchart images as in figure 2. When the system is turned on, the user must select a set point and then press the start button to turn on the pump to start applying pressure to the cuff. The pressure sensor will detect the pressure so that an error value appears as the input of the PID control system so that the tool will maintain the pressure in accordance with the set point. Then, if the stop button is pressed then the pump motor will stop applying pressure to the cuff (FIGURE 3).

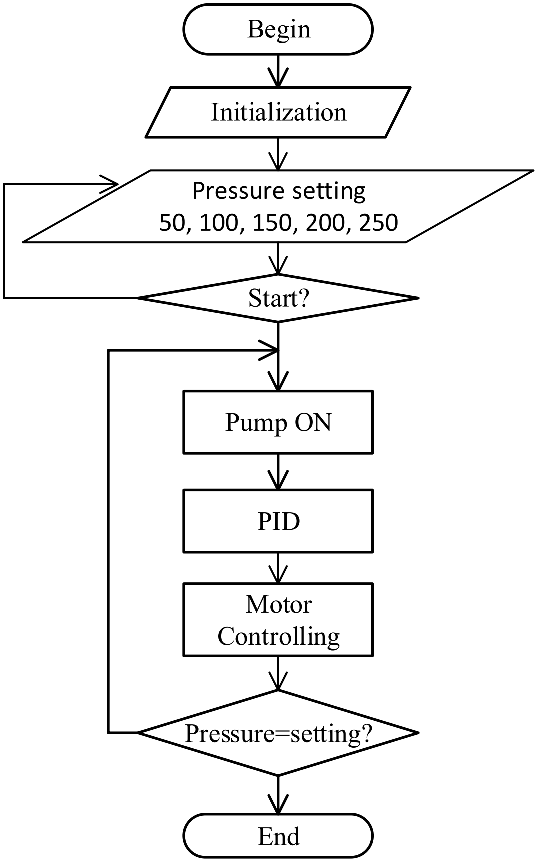


FIGURE 3. Flow Chart Of Blood Pressure Meter

D. PID CONTROLLER DIAGRAM

PID control is a control with a closed loop system. The block diagram above explains that the selected pressure setting will be compared with the sensor readings. The result of the comparison between the settings and the sensor readings is an error value. The error value will be calculated by the Kp, Ki, Kd control and the results of these calculations are added up so that the value comes out and is converted into a PWM pulse (FIGURE 4). The PWM value will be the input to control whether the plan works or not, which plan we use is the pump. When the pump is on, the MPX5100GP pressure sensor used in this study reads the resulting pressure changes so that it changes the error value and ultimately changes the PWM value. This step is repeated until the controller reaches its goal of reaching the set point and maintaining it at that set point.

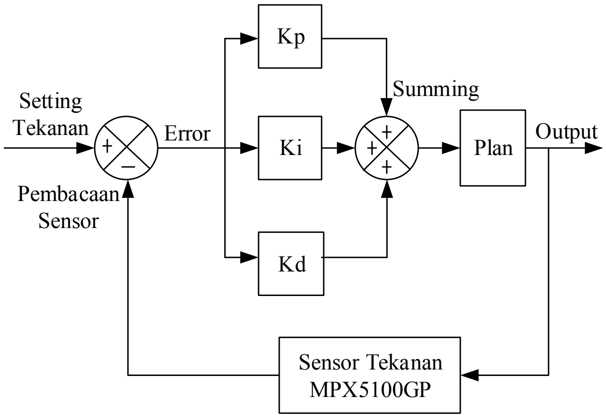


FIGURE 4. **PID CONTROLLER DIAGRAM**

E. SCHEMATIC

1) POWER SUPPLY CIRCUIT

The 6v DC voltage lowering supply circuit is assembled to lower the voltage of 12v DC from the Adapter / Battery to match the specifications of the next circuit voltage, made 2 supply circuits functioned supply A specifically to the motor voltage and supply B specifically to the next frame.

2) SENSOR MPX5100GP AND BUFFER

The voltage released by the sensor output then passes through the buffer where the buffer circuit serves as a buffer, so that the resulting current is equal to the input current.

3) PHASE REVERSAL NETWORK

Phase-reversing circuit, functioning to reverse the voltage of 6v DC to -6v DC specifically for the supplier of IC OP-AMP circuit.

4) AMPLIFIER INVERTING CIRCUIT

Sensor output that has passed buffer will be at Zero-point using summing frame with the dissing obtained from the inverting circuit. After zero-point, continue to the amplifier inverting circuit. The amplifier inverting circuit here serves to reverse the phase on the summing output and strengthen the output until it is condensed to 4.5v.

5) DRIVER NETWORK

The driver circuit has 2 inputs in the form of pin inputs from the PWM microcontroller output and a 6v DC voltage input for motor-specific voltages.

6) MICROCONTROLLER CIRCUIT

The output of the sensor and the signal conditioning circuit will be forwarded to the microcontroller on the Analog A1 pin to be read and converted into ADC data so that it can then be entered into the pressure conference formula, the result from this will be compared to the pressure setting variable to be achieved where the result of the error value obtained will enter into the PID process to regulate the size of the PWM value to be issued from the PWM pin D3.

III. RESULT

In this study, the pressure on the system has been calibrated with DPM, where the conformity process is standardized and performed directly with the pump in an active state, for the appearance of the following tool form:

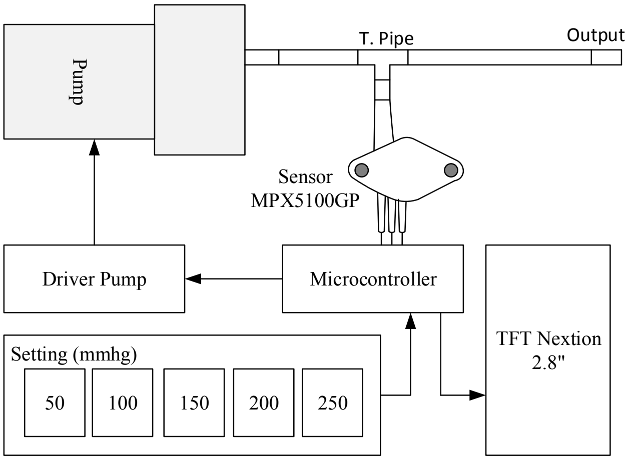


FIGURE 5. **PID sphygmomanometer control pump design**

A. DESIGN

The design photos are shown in Figure 3 to Figure 8. It consists of a power supply circuit as a voltage source, the voltage generated by the MPX5100GP output sensor then passes through a buffer where the buffer circuit functions as a buffer, so that the resulting current is the same as the input current. The output sensor that has passed the buffer will be at the zero-point using the sum of the frames with the dissing obtained from the inverting circuit. After the zero point, proceed to the inverting amplifier circuit. The amplifier inverting circuit here works to phase the summing output and amplify the output. The driver circuit has 2 inputs in the form of an input pin of the PWM microcontroller output and a 6v DC voltage input. The output from the sensor and the signal conditioning circuit will be used for the microcontroller on Analog pin A1 to be read and converted into ADC data so that it can then be entered into the conference formula. Then the error value obtained will be entered into the PID process to set the PWM size. the value to be output from the PWM pin D3.

B. PID AND NON-PID CONTROL TEST RESULTS

This measurement is done gradually. The initial condition of the point 0 mmhg rises periodically with a multiple of 50 mmhg according to the set point that has been determined up to 250 mmhg. Then from the highest point of 250 mmhg the pressure is lowered gradually to the point of 50 mmhg. This measurement proves the control response to set point changes is quite good.

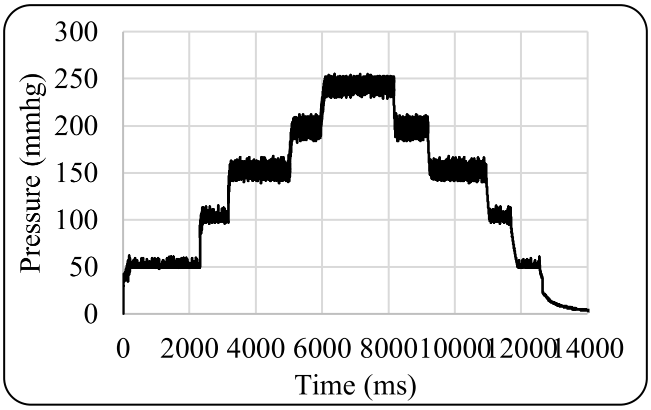


FIGURE 6. PID Data Collection step by step

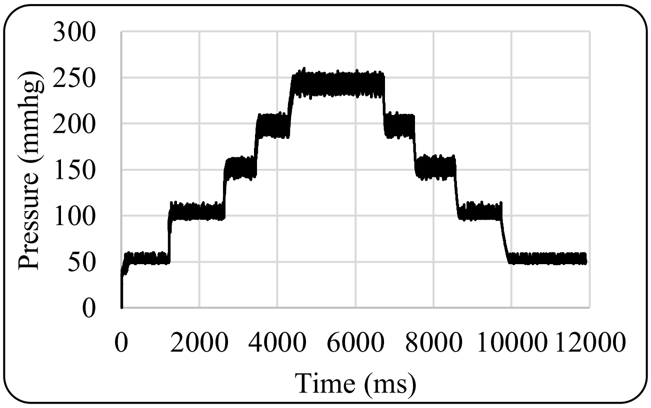


FIGURE 7. Non-PID Data Collection step by step

FIGURE 6 shows the result of the data colection stages at settings 50, 100, 150, 200, and 250 using the PID control. Furhtermore, FIGURE 7 shows result of data collection in stages at settings 50, 100, 150, 200, and 250 using Non-PID control. From FIGURE 6 and FIGURE 7 can be seen the pressure change response can be executed properly in the absence of extreme oscillations that make the air pressure control system uncontrolled. Measurements are flow up and down, this is tested in practice will do the same thing. So, the author has to test the tool system with the same thing. Here is a graph of the response of PID control and on-off control response.

From data collection, it can be seen the difference between rise time, overshoot and undershoot (TABLE 1). Here is the difference data (FIGURE 8 and FIGURE 9):

TABLE 1
Table of Differences in Rise Time, Overshoot, and Undershoot on PID and On-off Controls

		PID			Non-PID		
		RT	OS	US	RT	OS	US
		(s)	(mmhg)		(s)	(mmhg)	
Set point (mmHg)	50	2,9 s	54	49	2,4 s	53	49
	100	4,1 s	109	98	5,6 s	105	99
	150	5,6 s	155	146	5,8 s	158	147
	200	7,7 s	203	189	7,6 s	204	191
	250	10 s	252	234	9,4 s	251	235
Average		6,06	154,6	143,2	6,16	154,2	144,2

Note: RT indicates rise time (seconds), OS shows overshoot (mmHg), and US indicates undershoot (mmHg).

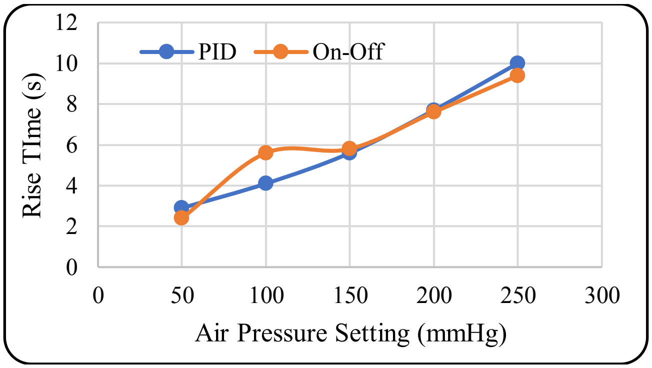


FIGURE 8. Rise time between the PID and on-off control

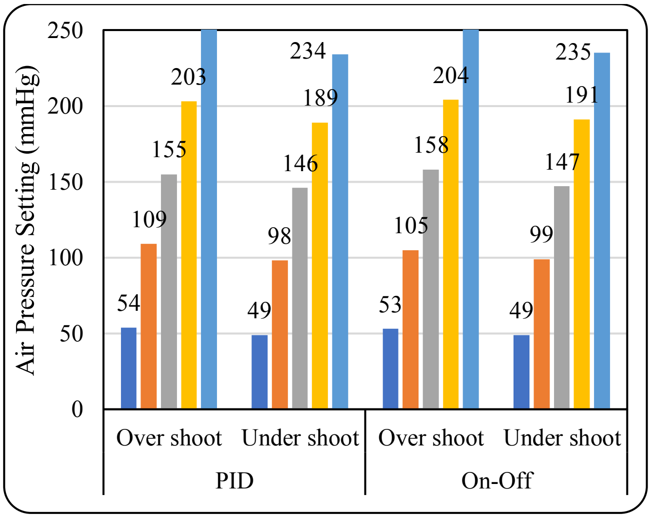


FIGURE 9. The comparison of overshoot and undershoot between the PID and on-off control.

From the data collection experiment above, it is also found that the difference in the results of voltage and current when using PID and Non-PID controls. The following are the differences: the PID Control has an average rise time of 6.06s, an average over shoot of 154.6, and an average under shoot of 143.2 while the Non-PID Control has an average The average rise time is 6.16 s, the over shoot average is 154.2, and the under shoot average is 144.2. It can be seen from these results that the rise time and under shoot when using PID Control is superior to when using Non-PID Control, which has a difference in rise time of 0.10 s and under shoot difference of 1. superior to PID Control the difference is 0.4.

TABLE 2
Table of Voltage and Current Differences in PID Control

PID					
mmHg	50	100	150	200	250
VDC	11,86	11,87	11,84	11,7	11,73
Ampere	0,32	0,36	0,54	0,69	0,77

From TABLE 2 on the PID control, the current flowing when setting 50 is 0.32 Ampere and has a voltage of 11.86 VDC. At setting 100 the current flowing is 0.36 Ampere and has a voltage of 11.87 VDC. At setting 150 the current that flows at this setting is 0.54 Ampere and has a voltage of 11.84 VDC. At setting 200 the current flowing at this setting is 0.69 Ampere and has a voltage of 11.70 VDC. While in the setting of 250 the current that flows at this setting is 0.77 amperes and has a voltage of 11.73 VDC.

TABLE 3
Table Of Voltage And Current Differences In Pid Control

Non-PID					
mmHg	50	100	150	200	250
VDC	11,9	11,89	11,87	11,72	11,67
Ampere	0,38	0,52	0,64	0,73	0,79

From TABLE 3 on the Non-PID control, the current flowing when setting 50 is 0.38 Ampere and has a voltage of 11.9 VDC. At setting 100 the current flowing at this setting is 0.52 Ampere and has a voltage of 11.89 VDC. At 150 settings, the current flowing at the time of this adjustment is 0.64 Ampere and has a voltage of 11.87 VDC. At setting 200 the current flowing at the time of this setting is 0.73 Ampere. And has a voltage of 11.72 VDC. While in setting 250 the current flowing at this setting is 0.79 Ampere and has a voltage of 11.67 VDC.

TABLE 4
Voltage comparison on PID control and On-Off control

	mmHg					Average
	50	100	150	200	250	
PID (volt)	11,86	11,87	11,84	11,7	11,73	11,8
On-Off (volt)	11,9	11,89	11,87	11,72	11,67	11,81

The results of the calculation of the average required voltage in the system created using the PID control reduce the average power consumption of 0.01 volts which is superior to the On-Off control (TABLE 4 and TABLE 5).

TABLE 5
Current comparison in PID control and On-Off control

	mmHg					Average
	50	100	150	200	250	
PID (Ampere)	0,32	0,36	0,54	0,69	0,77	0,536
On-Off (Ampere)	0,38	0,52	0,64	0,73	0,79	0,612

The results of the calculation of the average current consumption required in the system created using the PID control, the current consumption required is 0.536 amperes, which is superior to using the On-Off control with a current consumption of 0.0612 amperes.

VI. DISCUSSION

This research design using PID controls has been fully tested in this study. Testing with a setting of 250mmHg by setting the value of Kp (multiple of 10) to determine the value of the PID constant After tuning the controller with the second Ziegler-Nichols method, obtained the result of temperature oscillation at the setting point that is considered the most stable occurs when the value K = 50 which is then known as Kcr. After getting the result next determine the PCR by calculating the average of the peak period to the peak. Which is then obtained a period of 0.3 seconds. From the results of Kcr and Pcr get a constant PID value result that is Kp = 1000 Ki = 142,87 Kd = 660. Based on sphygmo-manometer output measurements using PID control, the output generated when the calibration measurement of the othermother calibration shows a small oscillation value in the results data that uses smoothing.

Results obtained on PID, the data obtained is at the setting of 50 the average overshoot is 54 and the average undershoot is 49; at setting 100 the overshoot average is 109 and the undershoot average is 98; at setting 150 the overshoot average is 155 and the undershoot average is 146; at the setting of 200 the overshoot average is 203 and the undershoot average is 189; At 250 the overshoot average is 252 and the undershoot average is 234. While the result of non-PID is the data obtained is at the setting of 50 the average overshoot is 53 and the average undershoot is 49; at setting 100 the overshoot average is 105 and the undershoot average is 99; at setting 150 the overshoot average is 158 and the undershoot average is 147; at the setting of 200 the overshoot average is 204 and the undershoot average is 191; At 250 the overshoot average is 252 and the undershoot average is 235.

VII. CONCLUSION

This research was conducted to test and prove the stability of PID control, it can be concluded that the use of PID control in this system using Ziegler Nichols 1 obtained results that with KP: 1000, Ki: 142. and Kd: 660. The output of the setting is obtained by smaller oscillations. Ziegler Nichols 2 is also used in this system but the results cannot be optimal with Ziegler one, this is because the pump has a built-in loss when the pump is dead. Oscillation becomes more optimal when the addition of smoothing to the sensor reading. Systems that are made have a lack of design from hardware can be made small by using SMD components, so the tool box is smaller. In the research, it is hoped that it can be developed by reducing the tool so that it is easy to carry everywhere, adding a stopper to make it more precise and to help medical personnel

REFERENCES

[1] V. A. Tsyrlin, M. G. Pliss, and N. V. Kuzmenko, "The history of blood pressure measurement: from Hales to our days," *"Arterial'naya Gipertenz. ("Arterial Hypertens.*, vol. 22, no. 2, pp. 144–152, 2016, doi: 10.18705/1607-419x-2016-22-2-144-152.

[2] M. Ashworth, K. Gordon, G. Baker, and A. Deshmukh, "Sphygmomanometer calibration: A survey of one inner-city primary care group," *J. Hum. Hypertens.*, vol. 15, no. 4, pp. 259–262, 2001, doi: 10.1038/sj.jhh.1001172.

[3] F. F. Rooswita, T. Rahmawati, and S. Syaifudin, "A Two-Mode Digital Pressure Meter Equipped With An Automatic Leak Test Using MPX5050gp And MPXv4115vc6u Sensors," *J. Electron. Electromed. Eng. Med. Informatics*, vol. 2, no. 2, pp. 80–85, 2020, doi: 10.35882/jeeemi.v2i2.7.

[4] A. Mujadin and P. W. Kusuma, "Design a noninvasive digital blood pressure meter using high sensitivity pressure gauge MPX5050GP," *2017 Int. Symp. Electron. Smart Devices, ISESD 2017*, vol. 2018-Janua, pp. 236–241, 2017, doi: 10.1109/ISESD.2017.8253339.

[5] S. K. Singha, K. Sarkar, and M. Ahmad, "An Easy Approach to Develop a Digital Blood Pressure Meter," *2nd Int. Conf. Electr. Comput. Commun. Eng. ECCE 2019*, pp. 1–5, 2019, doi: 10.1109/ECACE.2019.8679246.

[6] A. Pudji and M. R. Makruf, "Design of the Digital Pressure Meter with Thermohygrometer," *J. Inf. Eng. Appl.*, vol. 7, no. 9, pp. 35–39, 2017.

[7] S. Ghareeb *et al.*, "Results of a project to calibrate mercury sphygmomanometer blood pressure-measuring devices in Egypt," *J. Hum. Hypertens.*, vol. 35, no. 10, pp. 921–926, 2021, doi: 10.1038/s41371-020-00424-0.

[8] M. R. Z. Fajar, S. Hadiyoso, and A. Rizal, "An interfacing digital blood pressure meter with arduino-GSM module for real-time monitoring," *ICCREC 2017 - 2017 Int. Conf. Control. Electron. Renew. Energy, Commun. Proc.*, vol. 2017-Janua, pp. 98–102, 2017, doi: 10.1109/ICCEREC.2017.8226669.

[9] K. Negishi *et al.*, "Importance of calibration method in central blood pressure for cardiac structural abnormalities," *Am. J. Hypertens.*, vol. 29, no. 9, pp. 1070–1076, 2016, doi: 10.1093/ajh/hpw039.

[10] A. Cholid, H. G. Ariswati, and S. Syaifudin, "Digital Pressure Meter Equipped with Temperature and Humidity," *Indones. J. Electron. Electromed. Eng. Med. informatics*, vol. 2, no. 1, pp. 1–5, 2020, doi: 10.35882/ijeeemi.v2i1.1.

[11] C. F. Babbs, "Oscillometric measurement of systolic and diastolic blood pressures validated in a physiologic mathematical model," *Biomed. Eng. Online*, vol. 11, pp. 1–22, 2012, doi: 10.1186/1475-925X-11-56.

[12] J. H. M. Won Ki Kim, Ki Young Shin, "4 The Development of a Cuff for the Accuracy Enhancement of Sphygmomanometer.pdf." Jurnal of the Korean Society of Precision Engineering, 2005.

[13] A. de Greeff, I. Lorde, A. Wilton, P. Seed, A. J. Coleman, and A. H. Shennan, "Calibration accuracy of hospital-based non-invasive blood pressure measuring devices," *J. Hum. Hypertens.*, vol. 24, no. 1, pp. 58–63, 2010, doi: 10.1038/jhh.2009.29.

[14] P. Martin, "An inexpensive blood pressure heart rate computer for laboratory or teaching use," *J. Appl. Physiol.*, vol. 40, no. 2, pp. 253–255, 1976, doi: 10.1152/jappl.1976.40.2.253.

[15] S. N. W. Erdyansyah, T. Hamzah, and D. Titisari, "A Digital Pressure Meter Equipped with Pressure Leak Detection," *J. Teknokes*, vol. 14, no. 2, pp. 55–61, 2021, doi: 10.35882/teknokes.v14i2.2.

[16] G. Parati, A. Faini, and P. Castiglioni, "Accuracy of blood pressure measurement: Sphygmomanometer calibration and beyond," *J. Hypertens.*, vol. 24, no. 10, pp. 1915–1918, 2006, doi: 10.1097/01.hjh.0000244935.19299.f5.

[17] T. Knight *et al.*, "Sphygmomanometers in use in general practice: An overlooked aspect of quality in patient care," *J. Hum. Hypertens.*, vol. 15, no. 10, pp. 681–684, 2001, doi: 10.1038/sj.jhh.1001251.

[18] B. Shahbabu, A. Dasgupta, K. Sarkar, and S. K. Sahoo, "Which is more accurate in measuring the blood pressure? A digital or an aneroid sphygmomanometer," *J. Clin. Diagnostic Res.*, vol. 10, no. 3, pp. LC11–LC14, 2016, doi: 10.7860/JCDR/2016/14351.7458.

[19] B. Mustapha, M. Z. Hassan, Z. Mohamad, I. Kamaruddin, and R. Yahya, "PC Based Blood Pressure Meter for E-health Application," *Proc. - 2014 4th Int. Conf. Artif. Intell. with Appl. Eng. Technol. ICAIET 2014*, pp. 319–324, 2014, doi: 10.1109/ICAIET.2014.59.

[20] H. Dai, L. T. D'Angelo, and T. Lueth, "Continuous blood pressure monitor with wireless interface," *2010 IEEE Int. Conf. Robot. Biomimetics, ROBIO 2010*, pp. 36–40, 2010, doi: 10.1109/ROBIO.2010.5723299.

[21] X. Y. Meng, J. L. Zhang, Y. Liu, J. B. Wen, and S. L. Huang, "Design of electronic blood pressure monitoring system based on double filter," *2015 2nd Int. Symp. Futur. Inf. Commun. Technol. Ubiquitous Heal. Ubi-HealthTech 2015*, pp. 56–60, 2015, doi: 10.1109/Ubi-HealthTech.2015.7203358.

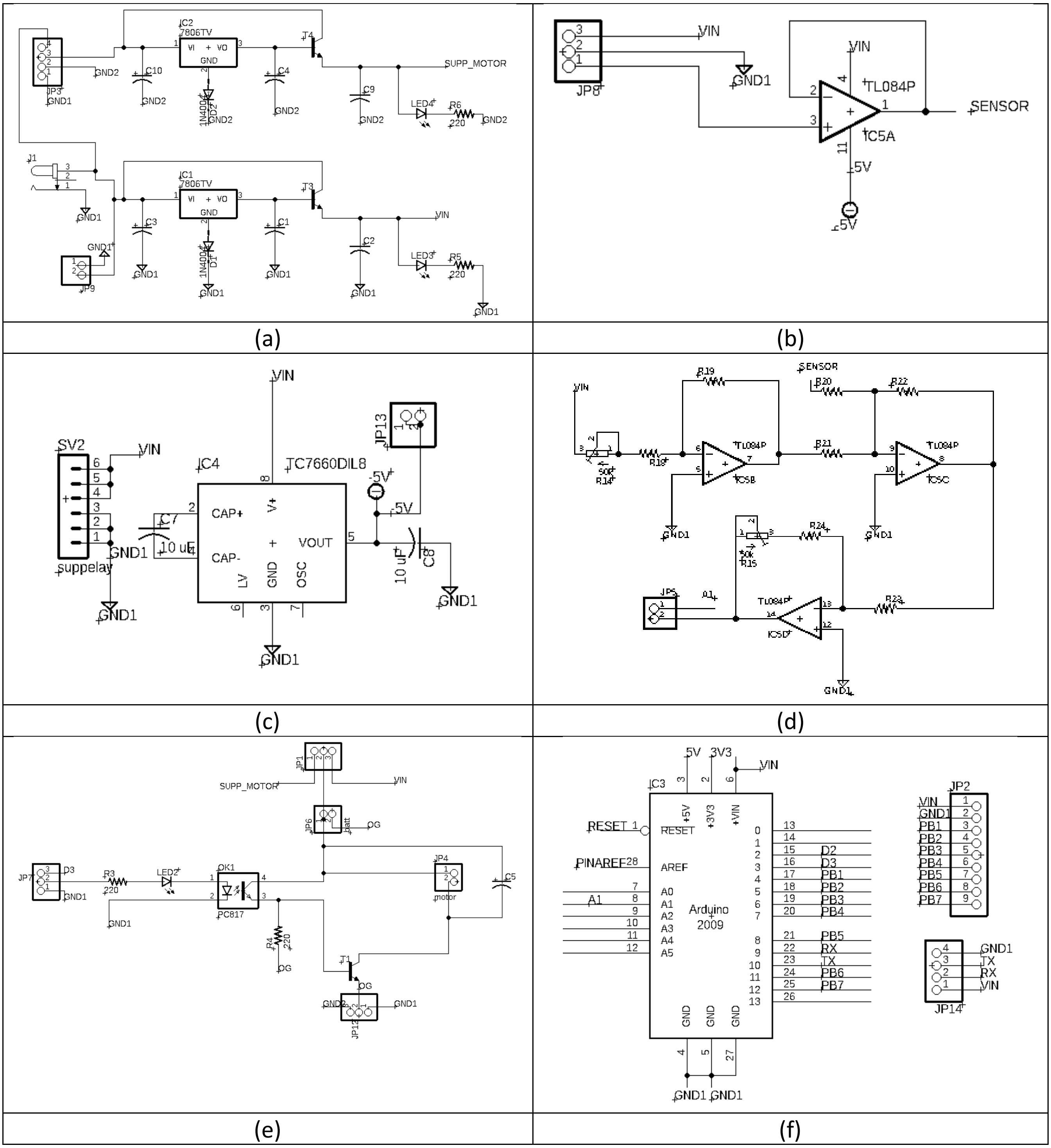
[22] M. M. Islam, F. H. M. Rafi, A. F. Mitul, M. Ahmad, M. A. Rashid, and M. F. B. A. Malek, "Development of a noninvasive continuous blood pressure measurement and monitoring system," *2012 Int. Conf. Informatics, Electron. Vision, ICIEV 2012*, pp. 1085–1090, 2012, doi: 10.1109/ICIEV.2012.6317436.

[23] Y. W. Kusumaningtyas, T. B. Indrato, M. P. A. T.P, and B. Utomo, "Digital Sphygmomanometer Based on Arduino Using TFT LCD Display," *Indones. J. Electron. Electromed. Eng. Med. informatics*, vol. 1, no. 1, pp. 34–38, 2019, doi: 10.35882/ijeeemi.v1i1.6.

[24] A. C. Bento, "IoT of Nextion X TFT ILI9341: Experimental Results and Comparative Survey," *Int. Res. J. Eng. IT Sci. Res.*, vol. 4, no. 2, pp. 14–23, 2018, doi: 10.21744/irjeis.v4n2.52.

[25] B. Utomo, I. D. G. Hariwisana, and S. Misra, "Design a Low-Cost Digital Pressure Meter Equipped with Temperature and Humidity Parameters," *Indones. J. Electron. Electromed. Eng. Med. informatics*, vol. 3, no. 2, pp. 59–64, 2021, doi: 10.35882/ijeeemi.v3i2.4.

APPENDIX



(a)

(b)

(c)

(d)

(e)

(f)

APPENDIX

1) ARDUINO PROGRAM INITIALIZATION

Initialization for the program using the serial software library to initialize the pins that will be connected to the TX and RX of the TFT LCD. Conversion program from ADC to cmH2O, as well as a smoothing program to make it more stable.

Program Listing 1. library to initialize the pins that will be connected to the TX and RX of the TFT LCD.

```
#include <SoftwareSerial.h>

SOFTWARESERIAL NEXSERIAL(9,10); //RX, TX
ARDUINO
```

The following listing is an initialization of a smoothing program that averages the last few output values on a reading. In this program, it can be seen in num Readings which is the number of data to be averaged which is 5, which means 5 values of readings to be averaged.

Program Listing 2. an initialization of a smoothing program that averages the last few output values on a reading

```
const int numReadings = 5;
int readings[numReadings];      // the readings from the analog
input
int readIndex = 0;              // the index of the current reading
int total = 0;                  // the running total
int average = 0;                // the average
```

From the following listing, the function is to initialize the sensor circuit output to be converted to a pressure unit.

Program Listing 3. initialize the sensor circuit output to be converted to a pressure unit.

```
float dis, vin_dis, milimeter_hg_dis, mmhg, VarSett=0;
```

From the following listing, it is the initialization of the PID process where there is a mention of the error difference, the result of the PID process, and the respective multiplier values of KP, KI, and KD.

Program Listing 4. the respective multiplier values of KP, KI, and KD.

```
float error,errorx;
float p,i,d,pid=0;
float kp = 50;
float ki = 0;
float kd = 55;
```

From the listing above, it is the initialization of the PWM output pins, variables on the control system, variables on the system On-Off the pump, and variables on the Nextion page display.

Program Listing 5. the PWM output pins, variables on the control system, variables on the system On-Off the pump, and variables on the Nextion page display.

```
int pwmout = 3;
int control;
int kon = 0;
int CurrentPage = 0;
```

2) VSET BUTTON PROGRAM

From the following listing, it is a void for setting the pressure setting variable from the button logic program, which means that when the button is pressed, the amount of that setting is currently selected.

Program Listing 6. VSET Button Program

```
 void vset(){
 if (digitalRead(4)==0){VarSett=50;delay(100);}
 if (digitalRead(5)==0){VarSett=100;delay(100);}
 if (digitalRead(6)==0){VarSett=150;delay(100);}
 if (digitalRead(7)==0){VarSett=200;delay(100);}
 if (digitalRead(8)==0){VarSett=250;delay(100);}

}
```

3) PROGRAM ON-OFF BUTTON

From the following listing, it is void for the logic of the On-Off pump button and the logic of the back to menu button on the Nextion page. When the on button is pressed, the program will activate the pump. When the off button is pressed, the program will turn off the pump. When the on and off buttons are pressed together, the program will turn off the pump and return the Next-tion page to the initial menu.

Program Listing 7. Program On-Off Button

```
void tombol(){
 if (digitalRead(11)==0){kon=1;delay(100);}
 if (digitalRead(12)==0){kon=0;delay(100);}
 if (digitalRead(11)==0&&digitalRead(12)==0){
  kon=0;
  CurrentPage = 0;
 HMISerial.print("page ");
 HMISerial.print(CurrentPage);
 HMISerial.write(0xff);
 HMISerial.write(0xff);
 HMISerial.write(0xff);
  delay(1000);
 }}
HMISerial.write(0xff);
 HMISerial.write(0xff);
 HMISerial.write(0xff);
  delay(1000);
 }
}
```

4) LISTING ARDUINO'S INITIAL PROGRAM

The void setup () function will only run once every time the Arduino is started or when it is restarted. The baud rate used by Arduino is 38400. Meanwhile for the TFT LCD it also uses a baud rate of 9600. So, it is necessary to change the Baud rate Nextion to 38400. It can also be seen on pin A1 Arduino as input from Arduino which will be processed and pins D4, D5, D6, D7, D8, D11, D12 Arduino as input from Arduino as button processor.

Program Listing 8. Listing Arduino's Initial Program

```
void setup() {
Serial.begin(38400);
for (int thisReading = 0; thisReading < numReadings;
thisReading++) {
   readings[thisReading] = 0;
  }
HMISerial.begin(9600);
  delay(1);
  HMISerial.print("baud=38400");
  HMISerial.write(0xff);
  HMISerial.write(0xff);
  HMISerial.write(0xff);
  HMISerial.end();
  HMISerial.begin(38400);

pinMode(pwmout,OUTPUT);
pinMode(4,INPUT);
pinMode(5,INPUT);
pinMode(6,INPUT);
pinMode(7,INPUT);
pinMode(8,INPUT);
pinMode(11,INPUT);
pinMode(12,INPUT);
pinMode(A1,INPUT);
}

 HMISerial.write(0xff);

 HMISerial.write(0xff);

 HMISerial.end();
 HMISerial.begin(38400);

pinMode(pwmout,OUTPUT);
pinMode(4,INPUT);
pinMode(5,INPUT);
pinMode(6,INPUT);
pinMode(7,INPUT);
pinMode(8,INPUT);
pinMode(11,INPUT);
pinMode(12,INPUT);
pinMode(A1,INPUT);

    }
```

5) PID PROGRAM

From the listing below, it is a void to activate and issue a PWM voltage value that will be sent to the driver.

Program Listing 9. PID Program

```
void PID(){
 analogWrite(pwmout,pid);
}
```

6) INITIAL PROGRAM

From the following listing, it is a void for the pump mode activation process in the form of PID mode and Non-PID mode, when entering PID mode, the display on Nextion shows that there is a color change in the background of the PID writing which will then enter the PID page. Likewise for the Non-PID.

Program Listing 10. Initial Program

```
void awal(){
 if (digitalRead(11)==0){
 HMISerial.print("t1.bco=");
 HMISerial.print(65535);
 HMISerial.write(0xff);
 HMISerial.write(0xff);
 HMISerial.write(0xff);
 delay(500);
  CurrentPage = 1;
  HMISerial.print("page ");
 HMISerial.print(CurrentPage);
 HMISerial.write(0xff);
HMISerial.write(0xff);
 HMISerial.write(0xff);
  delay(500);
 }
 if (digitalRead(12)==0){
 HMISerial.print("t2.bco=");
 HMISerial.print(65535);
 HMISerial.write(0xff);
 HMISerial.write(0xff);
 HMISerial.write(0xff);
 delay(500);
  CurrentPage = 2;
  HMISerial.print("page ");
 HMISerial.print(CurrentPage);
 HMISerial.write(0xff);
 HMISerial.write(0xff);
 HMISerial.write(0xff);
  delay(500);
 }
}
HMISerial.write(0xff);
 delay(500);
  CurrentPage = 2;
  HMISerial.print("page ");
 HMISerial.print(CurrentPage);
 HMISerial.write(0xff);
 HMISerial.write(0xff);
 HMISerial.write(0xff)
  delay(500);
 }
}
```

7) TEST PROGRAM WITH PID

From the following listing, it is a void to activate Pump mode with PID which will read the pressure setting value, then read the pump button command, enter the conversion process from ADC data into pressure units. Followed by a smoothing process taken from every 5 data to get an average result, then the data read has been smoothed. Then enter the PID system where the error value, the difference between the sensor and the setting value, will be multiplied by the PID system. The VarSett and Pressure values will be sent via serial transmission to the computer to be read by the comport on Delphi. Not only sending serial to the computer, also sending serial to Nextion with a different serial pin. The data sent is in the form of PWM percentages, pressure readings, VarSett values, and pressure values for the converted graph to match the size of the graph in Nextion.

Program Listing 11. Test Program with PID

```
void testpid(){
 vset();
 tombol();
dis=analogRead(A1);
  vin_dis=(dis*4.5/920);
  milimeter_hg_dis=(vin_dis*300/4.9);

  total = total - readings[readIndex];
  readings[readIndex] = milimeter_hg_dis;
  total = total + readings[readIndex];
  readIndex = readIndex + 1;
  if (readIndex >= numReadings) {
    readIndex = 0;  }
  average = total / numReadings;
  mmhg=average;

    error = VarSett - mmhg;
  p = error * kp;
  i = ki * (error + errorx) ;
  d = kd * (error - errorx) ;
  pid = p + i + d;
  if(pid < 1){pid = 0;}
  if(pid > 255){pid = 255;}
  errorx = error;
  //Serial.println(pwmout);
  control = map(pid,0,255,0,100);

    //if(count==1){ //Sensor MPX
  dis=analogRead(A1);
  vin_dis=(dis*4.5/920);
  milimeter_hg_dis=(vin_dis*300/4.9);

  total = total - readings[readIndex];
  // read from the sensor:
  readings[readIndex] = milimeter_hg_dis;
  // add the reading to the total:
  total = total + readings[readIndex];
  // advance to the next position in the array:
  readIndex = readIndex + 1;

  // if we're at the end of the array...
  if (readIndex >= numReadings) {
    // ...wrap around to the beginning:
    readIndex = 0;  }

  // calculate the average:
  average = total / numReadings;

//milimeter_hg_dis=map(milimeter_hg_dis,8,204,0,204);
  mmhg=average;///adj;

  //kilo_pascal_dis=(vin_dis/0.045)-(4.45);
  //milimeter_hg_dis=kilo_pascal_dis*7.5006;
    int grav=map((mmhg+25),0,275,0,165);

  //HMI

  Serial.print('a');
  Serial.print(grav);
  Serial.print('b');
  Serial.print('c');
  Serial.print(VarSett);
  Serial.print('d');
  Serial.print('e');
  Serial.print(mmhg);
  Serial.print('f');

  //Serial.print('b');
  HMISerial.print("add 3,0,"+String(grav));
  HMISerial.write(0xff);
  HMISerial.write(0xff);
  HMISerial.write(0xff);

  HMISerial.print("t0.txt=\"");
  HMISerial.print(mmhg, 1);
  HMISerial.print("\"");
  HMISerial.write(0xff);
  HMISerial.write(0xff);
  HMISerial.write(0xff);

  HMISerial.print("t2.txt=\"");
  HMISerial.print(VarSett,0);
  HMISerial.print(" mmHg\"");
  HMISerial.write(0xff);
  HMISerial.write(0xff);
  HMISerial.write(0xff);

  HMISerial.print("j0.val=");
  HMISerial.print(control);
  HMISerial.write(0xff);
  HMISerial.write(0xff);
  HMISerial.write(0xff);

  if (kon == 1){
    //analogWrite(pwmout,254);
    PID();
    digitalWrite(keran,1);
  }
```

```
  if (kon == 0){
    analogWrite(pwmout,0);
    pid=0;
    digitalWrite(keran,0);
  }
}
```

8) TEST PROGRAM WITH NON-PID

From the following listing, it is a void to activate Pump mode without PID which will read the pressure setting value, then read the pump button command, entering the conversion process from ADC data into pressure units. Followed by a smoothing process taken from every 5 data to get an average result, then the data read has been smoothed. Then enter the pwm control system where if the sensor value is smaller than VarSett then PWM will be active 100%, and if the sensor value is greater than VarSett then PWM will turn off. The VarSett and Pressure values will be sent via serial transmission to the computer to be read by the port on Delphi. Not only sending serial to the computer, also sending serial to Nextion with a different serial pin. The data sent is in the form of PWM percentage, pressure reading, VarSett value, and pressure value for the converted graph to match the size of the graph in Nextion.

Program Listing 12. Test Program With Non-Pid

```
void test(){
  vset();
  tombol();
    dis=analogRead(A1);
  vin_dis=(dis*4.5/920);
  milimeter_hg_dis=(vin_dis*300/4.9);
  total = total - readings[readIndex];
  // read from the sensor:
  readings[readIndex] = milimeter_hg_dis;
  // add the reading to the total:
  total = total + readings[readIndex];
  // advance to the next position in the array:
  readIndex = readIndex + 1;
  // if we're at the end of the array...
  if (readIndex >= numReadings) {
    // ...wrap around to the beginning:
    readIndex = 0;
  }
  // calculate the average:
  average = total / numReadings;
  mmhg=average;

if(mmhg<VarSett){pid=255;}
if(mmhg>=VarSett){pid=0;}
  control = map(pid,0,255,0,100);

  int grav=map((mmhg+25),0,275,0,165);

  Serial.print('a');
  Serial.print(mmhg);
  Serial.print('b');
  Serial.print('c');
  Serial.print(VarSett);
  Serial.print('d');
  Serial.print('e');
  Serial.print(mmhg);
  Serial.print('f');
    //Serial.print('b');
  HMISerial.print("add 3,0,"+String(grav));
  HMISerial.write(0xff);
  HMISerial.write(0xff);
  HMISerial.write(0xff);
   HMISerial.print("t0.txt=\"");
  HMISerial.print(mmhg, 1);
  HMISerial.print("\"");
  HMISerial.write(0xff);
  HMISerial.write(0xff);
  HMISerial.write(0xff);
  HMISerial.print("t2.txt=\"");
  HMISerial.print(VarSett,0);
  HMISerial.print(" mmHg\"");
  HMISerial.write(0xff);
  HMISerial.write(0xff);
  HMISerial.write(0xff);
  HMISerial.print("j0.val=");
  HMISerial.print(control);
  HMISerial.write(0xff);
  HMISerial.write(0xff);
  HMISerial.write(0xff);

  if (kon == 1){
   //analogWrite(pwmout,254);
    PID();
    digitalWrite(keran,1);
  }
  if (kon == 0){
 analogWrite(pwmout,0);
    pid=0;
    digitalWrite(keran,0);
  }
}
```

9) RECURRING PROGRAMS

Here we will read the page which will call the initial void for the mode selection, if the Nextion page is on page 0 it will display the initial menu on Nextion. If the Nextion page is on page 1, it will display the Non-PID menu and call the void Non-PID test program. If the Nextion page is on page 2, it will display the PID menu and call the void PID test program.

Program Listing 13. Recurring Programs

```
void loop() {
   // put your main code here, to run repeatedly:
  if(CurrentPage == 0){  // If the display is on page 0, do:
  HMISerial.print("t0.txt=\"");
  HMISerial.print("Pilih Mode\"");
  HMISerial.write(0xff);
  HMISerial.write(0xff);
  HMISerial.write(0xff);
  HMISerial.print("t2.txt=\"");
  HMISerial.print("PID\"");
  HMISerial.write(0xff);
  HMISerial.write(0xff);
  HMISerial.write(0xff);
    awal();
  }
   if(CurrentPage == 1){test();}
if(CurrentPage == 2){testpid();}
}
```