International Journal of ASRO
Volume 11, Number 02, pp. 166-176
Apr 2020

DESIGN OF PRESSURE MEASUREMENT IN THE EXERCISE SMART MINE USING A PRESSURE SENSOR

Endin Tri Hartanto, Edy Widodo, Ayip Rivai Prabowo, Sulaiman

Indonesian Naval Technology College, STTAL
Bumimoro-Moro Krembangan, Surabaya, 60178, Indonesia
sulaimanizoel@gmail.com

ABSTRACT

The Sea Mines are explosive devices placed in the waters to destroy ships or submarines. The sea mines were placed in the water depths and waited until it was triggered to be blown up by an approaching enemy vessel. The waves in the leave can be measured using the air pressure sensor placed below the water surface, the change in the sensor up and down will result in changes in air pressure. Changes in the resulting pressure of the vessel will result in water waves, resulting from changes in the water wave height will result in a change in water pressure gauge sensors in the air. The MPX5700 air pressure Sensor, as a source of pressure measurement with analog voltage output is comparable to the large value of air pressure. Design the air pressure measuring instrument system using the Arduino microcontroller as the unit of the voltage conversion process to the pressure variable in the KPA. On testing authors use applications that are created using Visual Studio 2012 to plotting the pressure graphs and large voltage output sensors. Results obtained using the MPX5700 sensor with an analog output can result in a linear pressure output with a large voltage output, from the test result obtained the conversion value 0.2 V as the value of 0 Kpa and the value of 4.7 V as the maximum value pressure measurement of 700 Kpa.

***Keywords**: mines, MPX5700, Arduino microcontrollers*

1. INTRODUCTION

Sea mines are explosive devices placed in the waters to destroy ships or submarines. The marine mines are placed in the water depths and wait until it is triggered to be blown up by the approaching enemy vessels. Marine mines work if there is target enemy vessels approaching mine site, This will enable or trigger mines to explode. Marine mines are one of the most ancient but effective weapons because marine mines are difficult to find. Marine mines can be run offensively by putting close to the target of defensive even by placing them on safe locations so as not to harm the personnel or friends them selves.

Forms of sea mines of various kinds, early in the appearance of drums, in the era of World War tubular or spherical. In order not to be easily known to enemies, sea mines are usually camouflaged by garbage, in the form of marine debris to avoid suspicious. But in the present era, the placement of sea mines is no longer floating in the water that can still be seen the naked eye but already under the surface of water.

The air in the atmosphere has pressure. Air pressure exists due to heavy air particles attracted in the Earth's gravitational style. If this particle's tensile style is lost then air particles will fly into the sky and the earth has no atmosphere. Without the atmosphere there will be no life on Earth. Expert air pressure is an energy that works to drive air masses in each unit of a particular area. In principle, air pressure equals pressure in liquids. Mountain air pressure will differ from the air pressure at the beach. This is caused at the summit of the mountain the smaller amount of air particles resulting in a particle's gravity style is also small, so the pressure on the air was even smaller.

According to the experts, air pressure is the energy that works to push the air masses in each unit of the particular area. Measured using a barometer. Millibar Air pressure Unit (mb). The line connecting the same place of air pressure is referred to as Isobar. The irrigation vessel will produce water waves, water wave changes emerge due to changes in water pressure due to change of vessel movement.

The resulting waves can be measured using the air pressure sensor that is placed below the water surface, Changes in the sensor up and down will result in changes in air pressure. Changes in vessel-generated pressure will result in water waves, Result of changes in water wave height will result in changes in pressure received by air pressure gauge sensors in the water.

For this reason, then in this research the author will make a design to build the instrument of air pressure gauge In water aimed at triggering a mine to explode. Use of Arduino microcontroller on air pressure data processing, will be applied to this research. Air pressure Sensor MPX5700, As a source of pressure measurement with analog voltage output comparable to large air pressure values. In realizing the air pressure measuring instrument system, the author tries to lift it into a research theme with the research title of Design of Pressure Measurement in Exercise Smart Mine using a Pressure Sensor.

2 RESEARCH METHODS

2.1 Design Research

Research design designing the pressure measurement in Exercise Smart Mine using Pressure sensors is a design research that consists of steps to achieve the final goal of the design and manufacture of measuring instruments. Air pressure. This research uses a method of designing builds.

Here is the design definition, according to Pressman: Design means a series of procedures to translate system analysis into the programming language example to explain in detail the system components implemented.

While waking/building systems are activities to create new systems or replace or repair existing systems either in whole or in part. Thus, the building design is the activity of translating the results into the form of software that then creates or corrects the existing system. Designing this research using Waterfall model design. Waterfall design models are workmanship of a system that is done sequentially or linearly. This design has 4 steps: Analysis, Design, Code, and Test. (Pressman: 2010, 39)

2.2 Research procedure

To implement the research well, the preparation of steps to find the data source as well as conducting research on the set up pressure measurement tool on Exercise Smart Mine using Pressure sensors that include instruments Research, primary data, secondary data, and validity testing and reliability of data.

a. Block diagrams

Designing will be easier in the realization of equipment with hardware diagram blocks. In designing this is expected to clarify what hardware systems and devices used in this research.

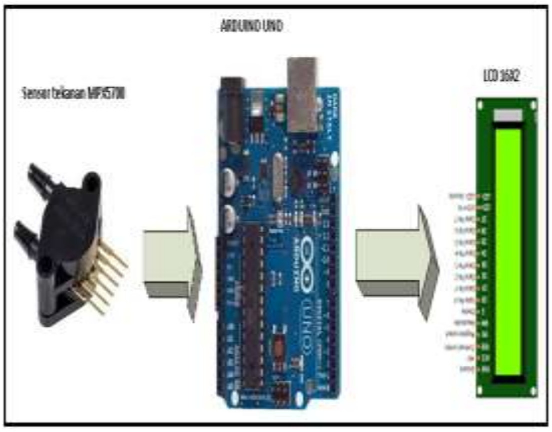


Figure 2.1 Block Diagram System

In planning the instrument of writers using the MPX5700 air pressure sensor that is easy to obtain in the market at an affordable price. The MPX5700 series Piezoresistive transducer is a sophisticated monolithic silicon pressure sensor designed for a wide range of applications, specifically to be integrated using microcontroller or microprocessor by via A/D. MPX5700 Channel gives the analog output signal comparable to the measured pressure. For an electrical connection with the Arduino Uno microcontroller can be seen in the Figure 2.2.

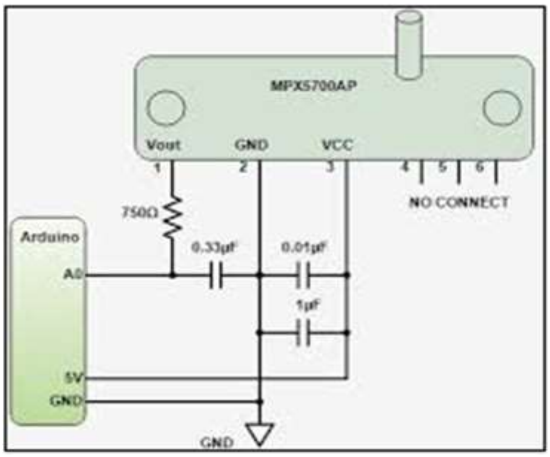


Figure 2.2 Air Pressure Measuring Instrument Schematic

For physical connection with the Arduino Uno microcontroller can be seen in Figure 2.3

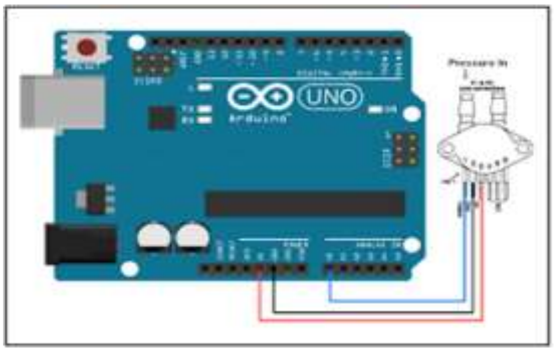


Figure 2.3 Air Pressure Measuring Instrument Series

To display the results of measurements on the LCD display can be seen in the overall measuring instrument scheme in Figure 2.4. The pressure Sensor is connected to the Pin A0 microcontroller analog input and the measurement results are displayed on the LCD display on the ports D8, D9, D10, D11 and D12.

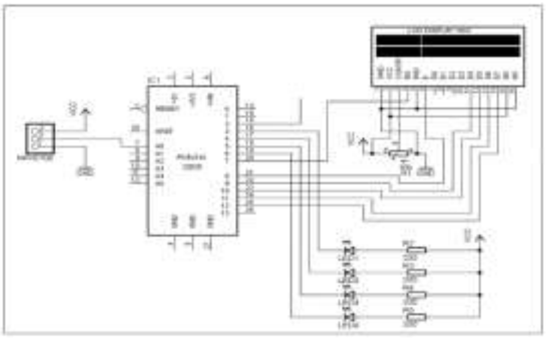


Figure 2.4 Images LCD Display and Air Pressure Measuring Instruments

To illustrate the instrument's measuring system can be seen on the system diagram block in general as shown in the following image:

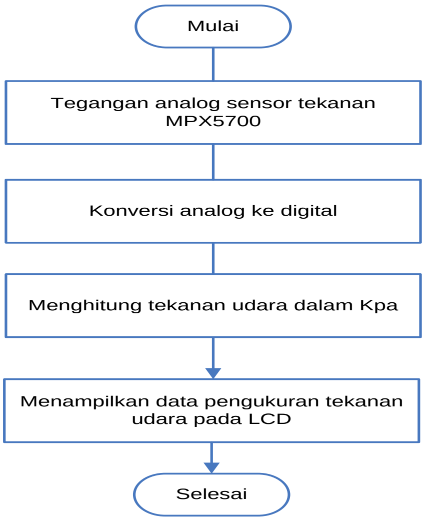

Mulai
Tegangan analog sensor tekanan MPX5700
Konversi analog ke digital
Menghitung tekanan udara dalam Kpa
Menampilkan data pengukuran tekanan udara pada LCD
Selesai

Figure 2.5 Air Pressure Measuring Instrument Flow Diagram

b. Diagram Alir Penelitian

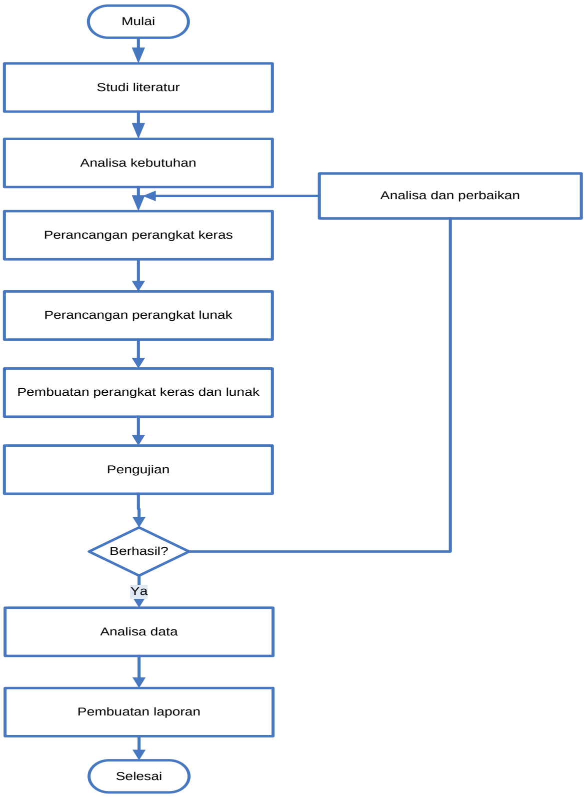


Figure 2.6 Flowchart Research Stages

2.3 Data Collection

In the research on designing the air pressure measurement in the Arduino based Exercise Smart Mine The author conducts the study by collecting data. The method of research data collection design build air pressure measurement in an Arduino-based Exercise Smart Mine that the author will do, namely:

a. Observation, this way the authors do by conducting research directly in the field.

b. Interviews, this way by doing a question and answer to the competent party regarding the communication of data wirelessly.

c. Documentation, this way is done by collecting data from books, records, and research results in related agencies regarding air pressure and measurements.

3.4 Data processing

Data processing is a process by which authors have taken some information collected in the data collection process. The Data obtained will be conducted and become reference to the design and system design analysis process.

2.5 Operational Definitions

The operational definition of this design plan explains and defines the variables used in this design. The following operational definitions in the design of the:

a. mBar : Unit of air pressure measure in Milli Bar

b. Vdc : large measure of the potential difference in direct current voltage or dc.

c. Meters: Height, depth or length unit of measure.

3. DESIGNING, IMPLEMENTING AND TESTING THE SYSTEM

3.1 Air Pressure Measurement System Design

In the design analysis of the system that will be made the author will explain how the system works, so that the design can be made according to the purpose. In the draft air pressure measurement, The author uses a MPX5700 air pressure sensor which is a sensor or transducer piezoresistive pressure sensor monolithic silicon and can be connected Microcontroller device through analog input channels. From the MPX5700 pressure sensor datasheet is obtained that the analog output is proportional to the pressure value received on the sensor. Figure 3.1 The following is a MPX5700 sensor diagram block.

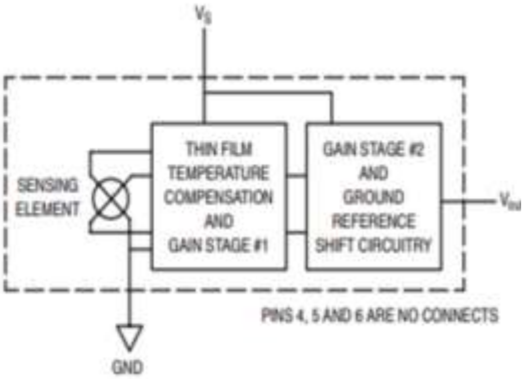

V_S
SENSING ELEMENT
THIN FILM TEMPERATURE COMPENSATION AND GAIN STAGE #1
GAIN STAGE #2 AND GROUND REFERENCE SHIFT CIRCUITRY
V_{out}
PINS 4, 5 AND 6 ARE NO CONNECTS
GND

Figure 3.1 Sensor Block Diagram MPX5700

The operating characteristics of the MPX5700 pressure sensor, the reliability test and internal qualifications are based on air use as a pressure medium. Figure 3.2 shows sensor output signal relative to pressure input.

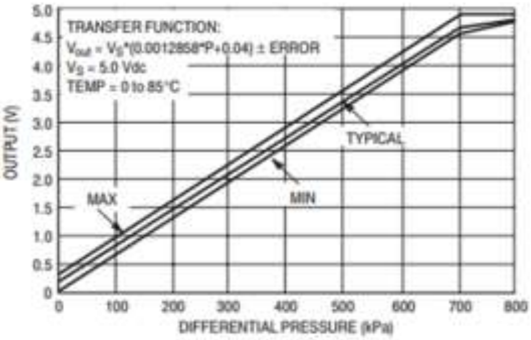


Figure 3.2 Sensor Output Graph Received Pressure

The output curve is at the minimum and maximum limit operating at 0 ° to 85 ° C, using a series of decouplings that show on Figure 3.3 The output will be saturated or saturation outside the specified pressure range. Figure 3.3 shows the recommended decoupling set for connecting the output from the sensor to A microcontroller's A/D input.

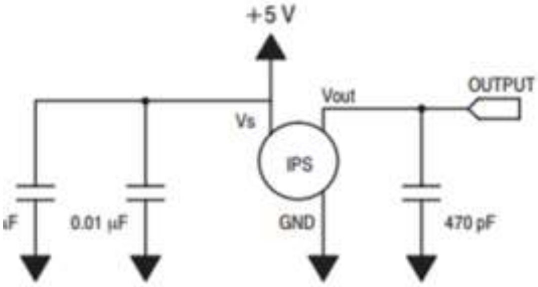


Figure 3.3 Decoupling Circuit on Output Sensor

The general design of the system for integrated pressure measurements on the Arduino microcontroller, can be seen in Figure 3.4 below.

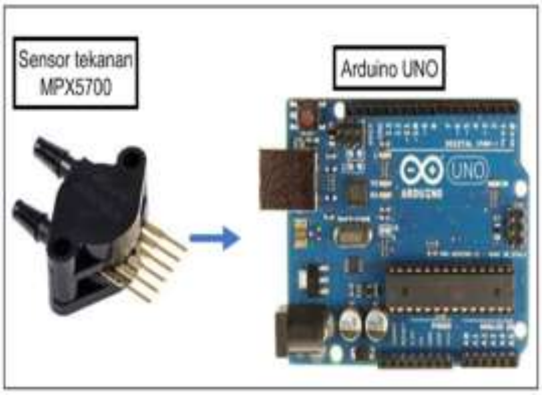


Figure 3.4 System General Design

Detail of the wiring scheme explaining the output link of MPX5700 sensor with analog input of Arduino Uno can be seen in Figure 3.5 below.

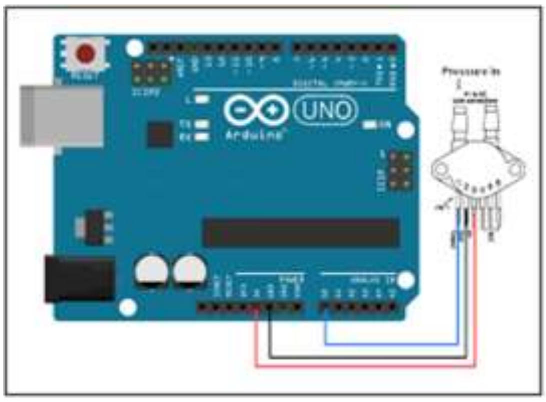


Figure 3.5 Wiring Connection Sensors with Arduino Uno

System interface planning as an application that can display the voltage and pressure data of MPX5700 sensors, is a design aimed at obtaining a system design User friendly or easy to use. The interface is designed using the Visual Studio 2012 Program editor.

The application-designed features are a feature to display data in the form of a number of digits or variable voltages and pressures, as well as voltage and pressure data in plots on the chart with a maximum data retrieval of 1000 data. After reaching the maximum of 1000 data then the graph will be reset for the repetition of the data. This is aimed at the graph does not occur in the data that results in data not being informative. Interface planning can be demonstrated in the design drawings between the 3.6 picture systems.

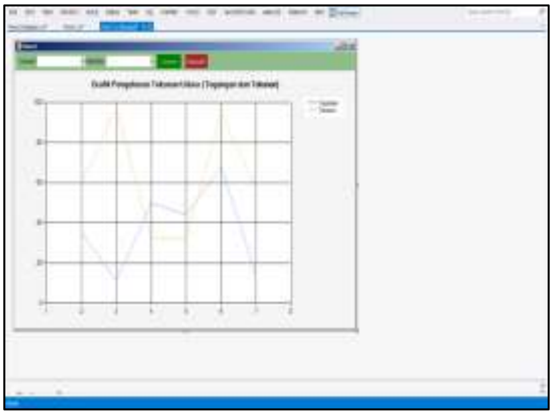


Figure 3.6 App Interface Design

To be able to communicate with the sensor pressure MPX5700 device then used USB data communication channels. In system design, authors use the SerialPort program libraries that have been provided by the Visual Studio 2012 program. In order to be able to communicate with the device via serial communication It is necessary to specify the parameters of the communication paramter corresponding to the sensor device parameters. These parameters are:

a. Baudrate

It is a synonym for symbols per second or pulse per second. It is a unit of velocity symbol, also known as the modulation rate, i.e. the number of changes Different symbols (signal earnings) made in the transmission medium per second.

b. Stopbits

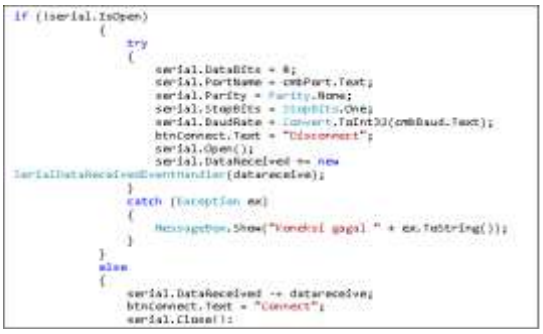
Is a synchronization, Stop bits can be 2 or 3 bits in size. As the name suggests, these bits will end the packet data. Stop bit size 1 or 2-bit.

c. Parity

Parity bits are useful for transfer data that is affected by noise. However, the use of parity bits can slow down the communicating speed. Parity bit usage also requires synchronization between transmitter and receiver. Otherwise, the likelihood of error in data interpretation is enormous.

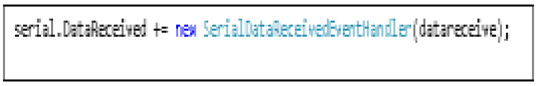
d. Databits

It is the number of bits in one data, generally to communicate with an Arduino Uno data bits microcontroller of 8 bits. Since the Arduino is a microcontrolller of 8 bits. In the program code to set the parameters baudrate, data bits, parity, should and stop bits of communication can be demonstrated on the following code snippet program. On the program code snippet is the process the user performs the button presses that will enable serial communication on the application.



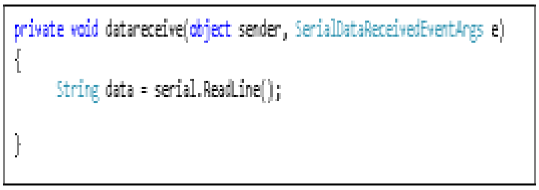
```
if (!serial.IsOpen)
        {
            try
            {
                serial.DataBits = 8;
                serial.PortName = cmbPort.Text;
                serial.Parity = Parity.None;
                serial.StopBits = StopBits.One;
                serial.BaudRate = Convert.ToInt32(cmbBaud.Text);
                btnConnect.Text = "Disconnect";
                serial.Open();
                serial.DataReceived += new
SerialDataReceivedEventHandler(datareceive);
            }
            catch (Exception ex)
            {
                MessageBox.Show("Koneksi gagal " + ex.ToString());
            }
        }
        else
        {
            serial.DataReceived -= datareceive;
            btnConnect.Text = "Connect";
            serial.Close();
```

Once the serial communication is set up the parameters, then the next design adds an event handler or system that can handle if there is a change in the incident on the serial communication. An event handler designed is an Event Handler to handle if any data is received through a serial communication. Here's the program code to handle event handlers.



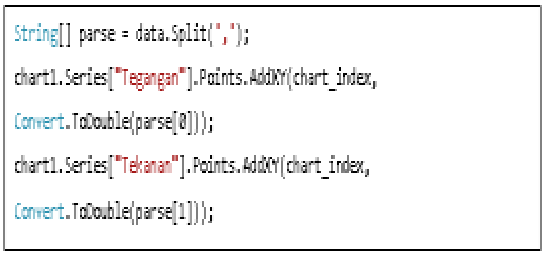
```
serial.DataReceived += new SerialDataReceivedEventHandler(datareceive);
```

The program code event handler to handle the data received on a serial communication will call the Datareceive function. This function functions to read the data received on a serial communication. Here's the tightening program code to read data through a serial communication.



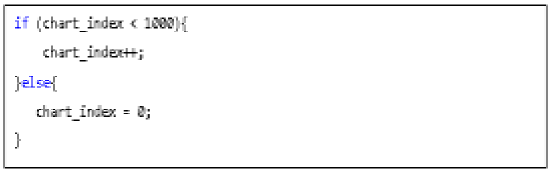
```
private void datareceive(object sender, SerialDataReceivedEventArgs e)
{
    String data = serial.ReadLine();

}
```

In the data receive data function in the form of string variables is a data reading result of serial data communication, each sentence of data series ends with new line (\r\n) as indicator of one sentence of data. In function data receive added function to sort out the received Data, data is sorted according to data intent and purpose. The first data is data voltage and the second data is the pressure data. Here's the program code design to parse the data.



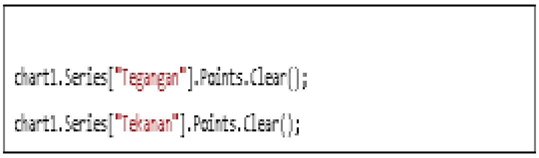
```
String[] parse = data.Split(',');
chart1.Series["Tegangan"].Points.AddXY(chart_index,
Convert.ToDouble(parse[0]));
chart1.Series["Tekanan"].Points.AddXY(chart_index,
Convert.ToDouble(parse[1]));
```

The data sorting results will be displayed to the graph with the chart name Chart1. The Chart1 chart is set by making two series of voltage and pressure series. As described earlier then every 1000 data is received then the graph will be reset and clear. The following program code shows the maximum data restriction of 1000.



```
if (chart_index < 1000){
    chart_index++;
}else{
   chart_index = 0;
}
```

The program code to perform a reset and clear chart can be shown in the following program code.



```
chart1.Series["Tegangan"].Points.Clear();
chart1.Series["Tekanan"].Points.Clear();
```

Once all the program code is designed and written in the Visual Studio 2012 editor, it can be generated a class diagram describing the existing class relationship diagram on the system. Draw a system diagram class designed to be seen in figure 3.7 below.

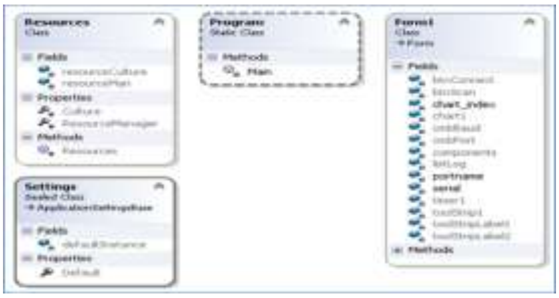


Figure 3.7 Class Diagram Application

3.2 Implementation

The implementation phase of air pressure measurement system on the draft that has been made is as follows.

a. Analog signal readings through the A/D pin of the Arduino microcontroller to be able to know the output of the MPX5700 sensor analog signal, it is necessary to do analog signal readings and to perform the process of converting the signal to digital. It can be done on Arduino Uno device by reading analog values on analog inputs Arduino in the form of digital data with a resolution of 10 bits. The Arduino analog input has a maximum input of VCC or 5 V voltages, with a resolution of 10 bits then a 5 V voltage readout value comparable to the digital value of 1023.

b. Performs a large conversion process of analog signals to the pressure size in the KPA. The conversion process can be done referring and based on datasheet MPX5700, large sensor output MPX5700 is comparable to the large pressure produced then sensor output is a linear output variable.

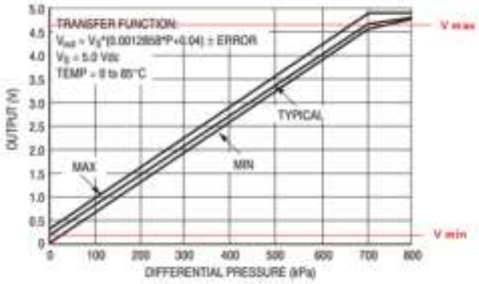
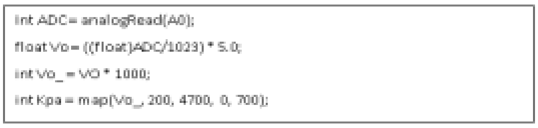


Figure 3.8 Analog Voltage Relations Chart with Air Pressure

Based on the chart of analog voltage relations with air pressure, can be created a formula on the Arduino program code to get the air pressure value:



```
int ADC= analogRead(A0);
float Vo= ((float)ADC/1023) * 5.0;
int Vo_ = VO * 1000;
int Kpa = map(Vo_, 200, 4700, 0, 700);
```

3.3 Analysis and Testing

3.3.1 Air pressure Measurement Testing

To be able to test and perform the air Tekana measurement, the required devices need to be connected as in Figure 3.9 the following wiring diagram:

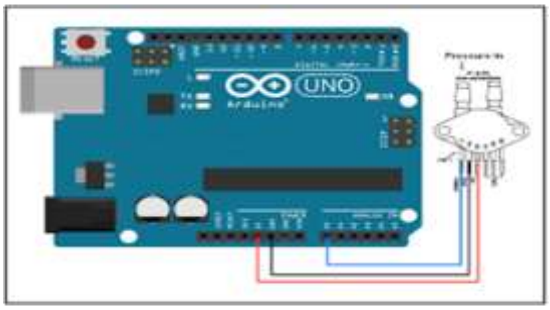
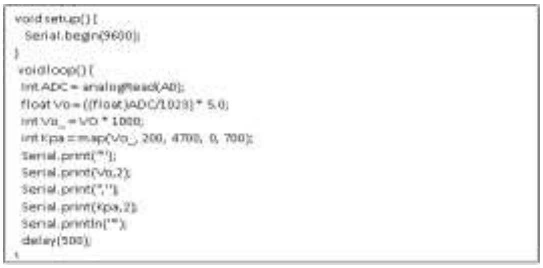


Figure 3.9 Wiring Diagram Device

The next step is to upload a program that has been created to the Arduino Uno device. The air pressure sensor measurement test program code can be seen in the following program code.



```
void setup() {
  Serial.begin(9600);
}
 void loop() {
  int ADC = analogRead(A0);
  float Vo= ((float)ADC/1023) * 5.0;
  int Vo_ = VO * 1000;
  int Kpa = map(Vo_, 200, 4700, 0, 700);
  Serial.print("");
  Serial.print(Vo,2);
  Serial.print(",");
  Serial.print(Kpa,2);
  Serial.println("");
  delay(500);
}
```

Integrated and connected laptop devices via USB channels for testing.

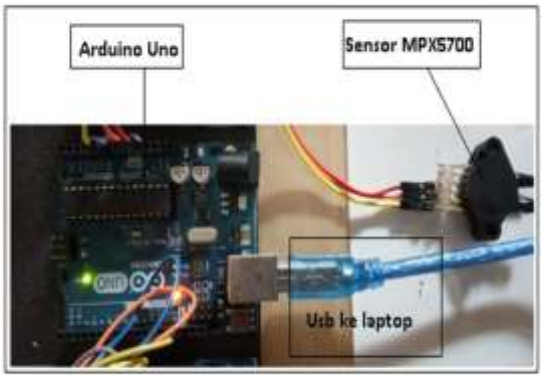
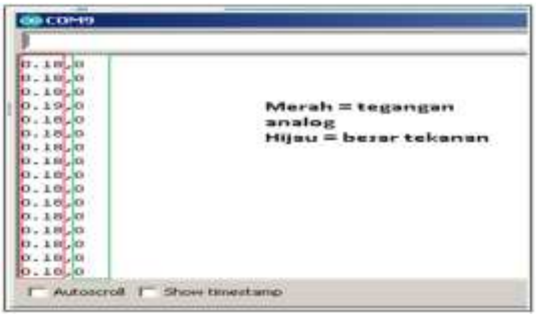


Figure 3.10 Test Device Image

In the program code created by the authors plan a test using the serial monitor to see the results of the sensor readings in the voltage and pressure magnitude of the KPA.

The following Image 3.10 is the result of the sensor readings displayed in the serial monitor.



```
0.18,0
0.18,0
0.18,0
0.19,0
0.18,0
0.18,0
0.18,0
0.18,0
0.18,0
0.18,0
0.18,0
0.18,0
0.18,0
0.18,0
0.18,0
0.18,0
```

Merah = tegangan analog
Hijau = besar tekanan

Image 3.11 Testing I Air Pressure Measurement Through Serial Monitor

To obtain the measurement results in the form of graph authors perform measurements using a monitoring program created using Visual Studio 2012. Here is a picture of air pressure measurement test chart. At this test the author has not given air pressure on the sensors mathematically that the pressure will approach the value 0.

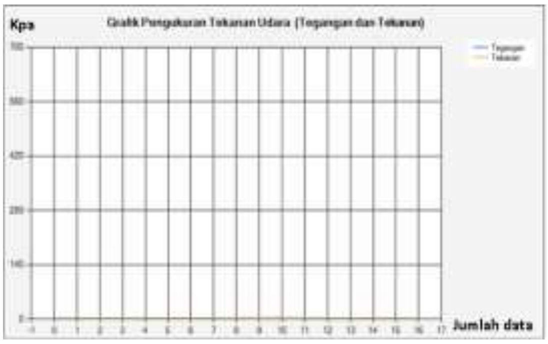
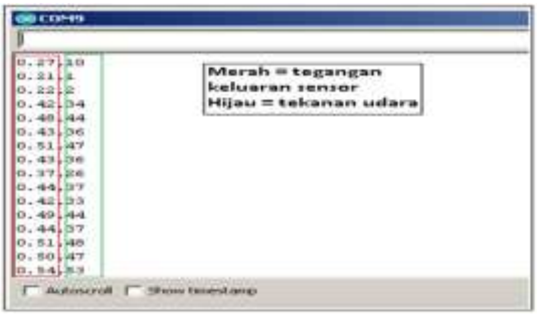


Figure 3.12 Test Graph Image I

In testing to two authors put pressure on the sensor by way of blown, it is only done the author to prove whether the pressure sensor care work properly.



```
0.27,10
0.21,1
0.22,2
0.42,34
0.48,44
0.43,36
0.51,47
0.43,36
0.37,26
0.44,37
0.42,33
0.49,44
0.44,37
0.51,48
0.50,47
0.54,53
```

Merah = tegangan keluaran sensor
Hijau = tekanan udara

Figure 3.13 Testing II Air Pressure Measurement Through Serial Monitor

To test the measurement of air pressure in the form of graphs, authors perform measurements using Visual Studio programs to obtain a measurement chart of voltage and pressure sensors.

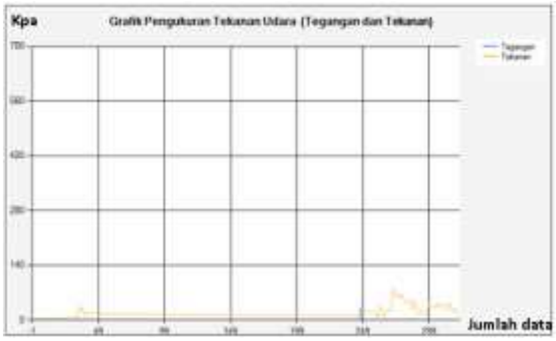


Figure 3.14 Test Chart Image II

Subsequent testing authors test the sensors by giving pressure on the sensor by 3 kpa, on the test chart displayed on the application program's interface graph indicated by the voltage and pressure graphs shown in Figure 3.15 Following.

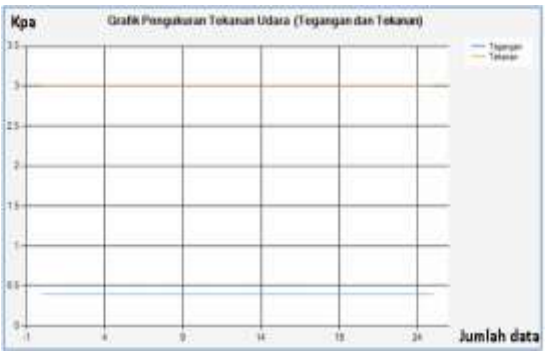


Figure 3.15 Testing Graph 3 Kpa Images

Subsequent testing of the authors testing the sensors by giving pressure on the sensor by 5 Kpa, on the test chart shown on the application program's interface graph indicated by the voltage and pressure graphs shown in Figure 3.16 Following.

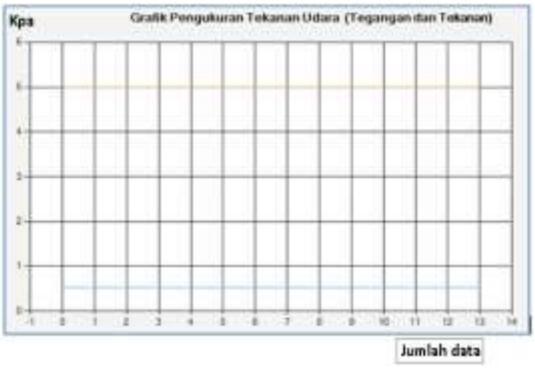


Figure 3.16 Test Chart 5 KPA Image

P Next, the author tests the sensors by giving pressure on the sensor by 6 Kpa, on the test chart shown on the application program's interface graph indicated by the voltage and pressure graphs shown in Image 3.17 following.

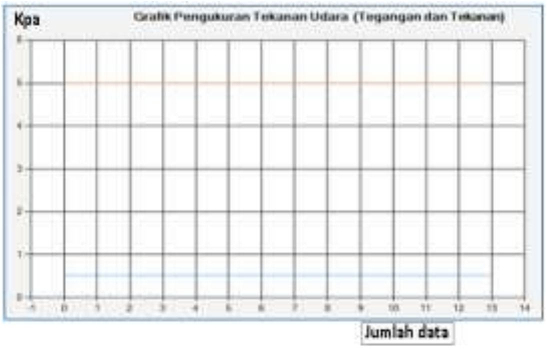


Figure 3.17 Test Graph Image 6 Kpa

Subsequent testing of the authors testing the sensors by giving pressure on the sensor by 7 Kpa, on the test chart displayed on the application program's interface graph indicated by the voltage and pressure graphs shown in Figure 3.18 Following.

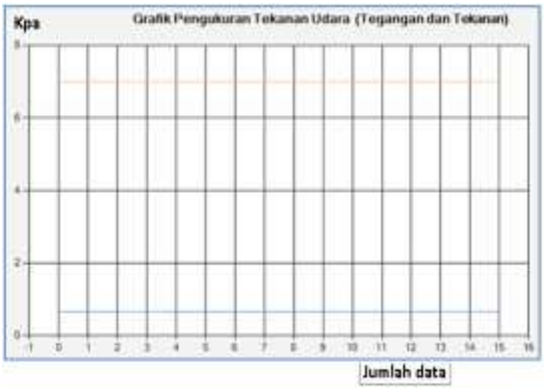


Figure 4.18 Test Charts 7 Kpa Image

Subsequent testing of the authors testing the sensors by giving pressure on the sensor by 8 Kpa, on the test chart displayed on the application program's interface graph indicated by the voltage and pressure graphs shown in Figure 3.19 Following.

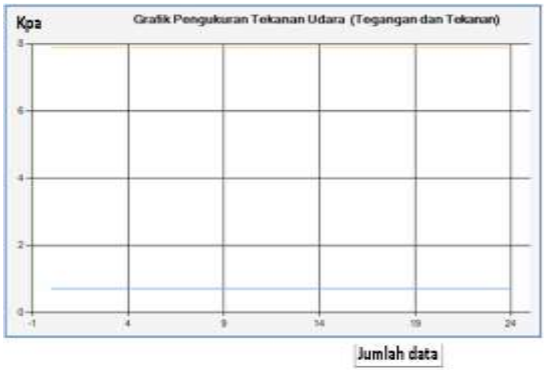


Figure 3.19 Test Chart 8 KPA Image

Subsequent testing of the authors testing the sensors by giving pressure on the sensor by 10 Kpa, on the test chart displayed on the application program's interface graph indicated by the voltage and pressure graphs shown in Figure 3.20 Following.

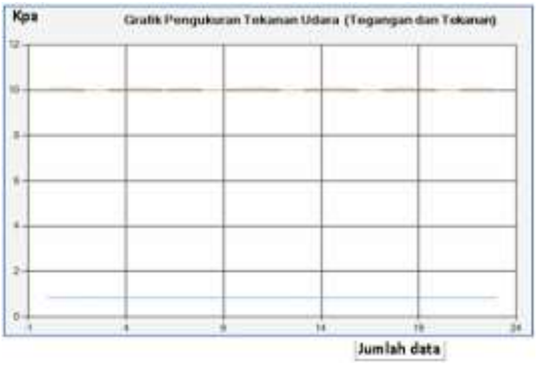


Figure 3.20 Image of 10 KPA Test Chart

Further testing the authors test the sensors by giving pressure on the sensor by 20 Kpa, on the test chart shown on the application program's interface graph indicated by the voltage and pressure graphs shown in Figure 3.21 Following.

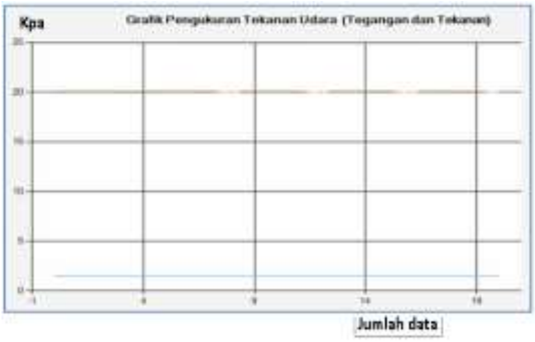
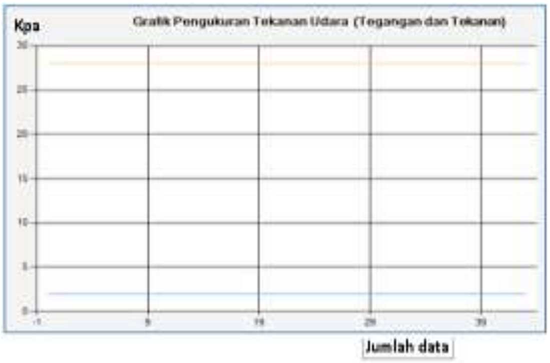


Figure 3.21 Test Graph Image 20 Kpa

Subsequent testing of the authors testing the sensors by giving pressure on the sensor at 27 Kpa, on the test chart shown on the application program's interface graph indicated by the voltage and pressure graphs shown in Figure 3.22 Following.



3.22 Images Test graph Image 27 Kpa

Subsequent testing of the authors testing the sensors by giving pressure on the sensor at 33 Kpa, on the test chart shown on the application program's interface graph indicated by the voltage and pressure graphs shown in Figure 3.23 Following.

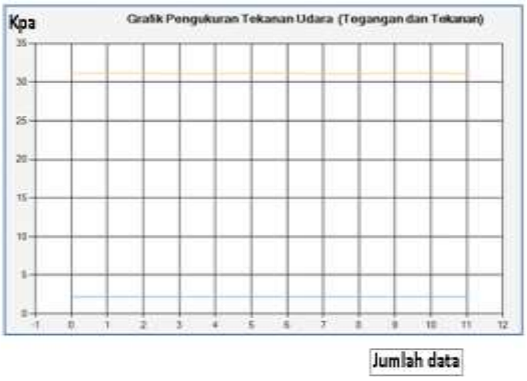


Figure 3.23 Test Graph Image 33 Kpa

From the tests that have been done can be created a table explaining the calibration between the output voltage sensor with a large pressure reading table 3.1. To illustrate the voltage calibration chart and the MPX5700 sensor pressure can be seen in Figure 3.24 below.

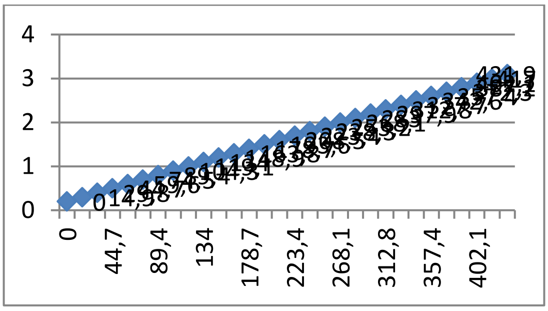


Figure 3.24 Voltage Calibration Chart with Sensor Pressure

The graphs resulting from the voltage and pressure relationship demonstrate the voltage and pressure interference linearity of the MPX5700 sensor. The chart shows a straight line of comparison so that it can be proven good level of interference linearity sensor MPX5700.

The equation used to find the pressure value based on the large output voltage, can be seen in the following formula:

$$kpa = \left(\frac{Vout-0.2}{5}\right)x700$$

Where:

Vout = Sensor output voltage, KPA = pressure conversion result

The following table of voltage relations with pressure.

Table 3.1 voltage calibration table with pressure

No	Voltage (Vdc)	Pressure (Kpa)
1	0.2	0.0
2	0.3	14.9

3	0.4	29.8
4	0.5	44.7
5	0.6	59.6
6	0.7	74.5
7	0.8	89.4
8	0.9	104.3
9	1	119.1
10	1.1	134.0
11	1.2	148.9
12	1.3	163.8
13	1.4	178.7
14	1.5	193.6
15	1.6	208.5
16	1.7	223.4
17	1.8	238.3
18	1.9	253.2
19	2	268.1
20	2.1	283.0
21	2.2	297.9
22	2.3	312.8
23	2.4	327.7
24	2.5	342.6
25	2.6	357.4
26	2.7	372.3
27	2.8	387.2
28	2.9	402.1
29	3	417.0
30	3.1	431.9

Based on the results of the tests that have been done analysis related to the conversion of sensor output voltage to the value of pressure received. In tests that have been performed can be obtained linear pressure values against the output voltage, this can be seen in the graph of the relationship between the voltage and pressure indicated by a linear relationship line. Pressure at 0 Kpa starts with a voltage of 0.2 Vdc and the maximum pressure is terminated with a large voltage of 4.7 Vdc.

A discussion of the air pressure measuring instrument system that has been made and tested, the author can make an analysis and discuss it on this sub-chapter. The measurement of air pressure on the mine device is a step or a way of detecting the presence of vessels passing through the sea water. Changes in water pressure around the mines due to passing vessels, resulting in changes in air pressure on the mine devices. The specific identification Parameter is not addressed in the creation of this tool, but refers to changes in air pressure that occur can provide a data that can be used as a guideline in the detection process. The results of the test show a linear relationship of air pressure output and sensor voltage output, there is a central point shifting pressure output at a voltage of 0.2 V. Meeting Value of 0 kpa point obtained at a value of 0.2 V, value 0.2 V is Negative pressure values. Negativ pressure is obtained when the MPX5700 sensor performs a negative air pressure measurement as well. Can be simplified by depiction, pressure with air push will produce positive variables while the pressure is attractive which results in a comparison of air volume with a small volume of space will produce a variable The negative.

Based on the testing and creation of the tool that has been done can be explained the work process of the device to get the output of air pressure variable in the KPA unit, the MPX5700 sensor is a state of the monolithic silicone air pressure sensor Art, which is a single-element tranducer combining sophisticated micromachining techniques, thin film metalization, and bipolar processing to provide an analog output signal comparable to that given pressure. The sensor output is connected to the Arduino Uno device with an analog interface on the Arduino Uno input analog pin. Since the MPX5700 sensor output results in an analog of 0.2 V to 4.7 V, it is necessary to create a formula that can produce the data output 0 – 700 Kpa. To generate output in the form of system graphs then on the tools made, Added an application using C# programming language to be able to interfacing with Arduino Uno device using serial communication interface. Data received through serial communications will be in the plot on the interface graph to obtain a graph of air pressure and large analog voltage output sensor MPX5700. On the resulting chart can be obtained a large value of the voltage comparison with a large air pressure to prove that the air pressure sensor MPX5700 as a pressure sensor that has a linear output.

4 CONCLUSIONS

Conclusions derived from the results of the test can be mentioned, namely:

a. The use of MPX5700 sensors with analog output can result in a linear pressure output with a large voltage output, from the test result obtained a conversion value of 0.2 V as the value of 0 Kpa and a value of 4.7 V as the maximum value of pressure measurement of 700 Kpa.

b. The conversion process can be done referring and based on the MPX5700 sheet data, the large MPX5700 sensor output is comparable to the large pressure produced then the sensor output is a linear output variable.

a. Develop a more sophisticated pressure monitoring system, higher accuracy and maximum limitation of measurements higher than the MPX5700 sensor.

b. Develop a system to obtain pressure variables in PSI and bar units.

ACKNOWLEDGEMENT

The authors greatly acknowledge the support from STTAL Surabaya Indonesia for providing the necessary resources to carry out this research work. The authors are also grateful to the anonymous reviewers and journal editorial board for their many insightful comments, which have significantly improved this article.

REFERENCES

Andrianto, Heri dan Aan Darmawan. Arduino Fast Learning and Programming. Bandung: Bandung Informatic, 2016.

Arduino, 2016, Arduino Uno Board, https://www.arduino.cc/en/Main/ArduinoBoardUno, June 11th 2019

I. Dyah and I. K. A. Puspitasari, "Portable calibrator," Muhammadiyah University of Yogjakarta, 2016.

Kadir, Abdul. A Practical Guide Learning Microcontroller Applications and Their Programming Using Arduino. Yogyakarta : C.V Andi Muis, 2013.

O. Adhitya, Digital Pressure Meter Sphygmomanometer Equipped Hega-20G Sensor Based On ATMega8 Microcontroller. Muhammadiyah University of Yogjakarta, 2017.

Ono, S., "MPX 5700 + Atmega 8", Mecanic Electronic Programming and Android, http://mekatronika-corner.blogspot.com June 19th 2019.

Urner, D., 2013, "Arduino – MPX", GitHub, Inc Vol.3, http://github.com, June 2019.

Sugiyanto, Totok. Rancang Bangun Identifikasi Personil GPS Life Jaket MenggunakanArduino Guna deteksi Dini Posisi Personil Jatuh di Laut Utuk Mendukung Tugas di Medan Operasi, STTAL Surabaya, 2018.