This file has been cleaned of potential threats.

If you confirm that the file is coming from a trusted source, you can send the following SHA-256 hash value to your admin for the original file.

e75bf9b1268d37dda65416b3f8f65b589235fafd182fd17b5aa6e90ad7b68fac

To view the reconstructed contents, please SCROLL DOWN to next page.
The Development of Automatic Weather Station Data Logger by Microcontroller Netduino

Fahdil Ahady Hazain¹* and Benfano Soewito²

¹Research and Development of Metindo Instrument, Jakarta, Indonesia
²Faculty Member of Magister of Information Technology, Bina Nusantara University, Jakarta, Indonesia

hazain.ahady@live.com, benfano@gmail.com

Abstract

The main purpose of the research was to develop an Automatic Weather Station (AWS) data logger using microcontroller Netduino. It will be useful to each sector which affected by weather. This research implement a telemetry system on Netduino logger in purpose to place it on remote area application, beside the study was developed a switching channel method to expand the number of I/O analog pin on Netduino. A correction factor resulted by data analysis during calibration and validation using linear regression method was “0.8594x + 4.7486” with $R^2 = 0.9367$. This research conclude that processing analog signal on Netduino has an alteration between its real value even not significantly, some treatment has implemented to reduce the error of data output, with using good power resource and moving averaging on each data reads (FAH).

Keywords: Microcontroller, Weather Station, Data Logger, Switching Channel Method

1. Introduction

1.1. Background

Indonesia is an archipelago between the Indian and the Pacific Oceans, and is located in the equator. These conditions affect to the climate and weather in Indonesia. Changes of weather and climate directly influence the condition of agriculture, ground, water and air transportation scheduling. Therefore people collect the weather data, studied and analyze them to obtain the necessary predictions. For this purpose, people do observing weather on the surface condition and the high level of the atmosphere (by using remote sensing) continuously.

Continuous weather information can be obtained from the weather station by observing surface condition [11]. Surface observations at weather stations have been done with the conventional technique in years, but it has many drawbacks considering it has been doing by humans on a regular basis. Weakness in these conventional techniques include incomplete data availability is often caused by the observer itself and the problem in technique of sending information to the data center. It encourages people to make an automatic weather station (AWS), so the risk of data loss can be minimized [11].

Indonesia has installed the AWS in many regions, and each year it continues to increase the number of installed AWS throughout Indonesia. The installed AWS in Indonesia are mostly from overseas manufacturers. This causes the cost to be high because the related shipping and installation. Moreover AWS that have been installed and operated will have the technical obstacles that need maintenance. Maintenance became issues because it is costly to invite the engineers from overseas manufacturers. In the
other hand, the knowledge of local engineer to maintain the AWS is limited [9]. Those could be the reasons AWS deployment process surrounding Indonesia becomes an issue.

Another obstacle faced after implementing the AWS is the diversity of applied communication in data transmissions from AWS to the local server in each region. This diversity is due to the differences AWS manufacturers which using different protocol in data transmissions such as FTP, SMS, XML, etc. and also the diversity of network infrastructure available in each remote station. Furthermore, sending data from local server to central database have done manually. Data in central database was inputted manually based on information that was received from local server by email. In this case, there must be a system that can integrate any communication techniques into the data center.

1.2. Objective

General purpose of installing the AWS system is to increase reliability of surface observation. Increasing reliability can be done by (1) increasing the density of an existing AWS by installing AWS in remote area, (2) Replacing the instruments to sophisticated technology and modern with digital measurement technique, and (3) Taking measurement, analyzing and reporting in short period and continuously, (4) Developing an integrated telemetry database system using existing network (GSM).

The specific purpose of this study is to provide an alternative solution for government and the agribusiness companies to be able to obtain AWS for a reasonable cost and fully utilize by local human resources. Surely thus AWS can be installed in more places (because of the cheaper cost). The more AWS installed in sites, the observation results of naturally weather data obtained will be more reliable, and certainly can be used to promote agriculture, plantation, forestry, and transportation. With an integrated telemetry database system, it will ease to forecast the weather information to public in real-time and accurately [12].

1.3. Problem Statement

In the development of an instrument data logger, there are some problems to solve. Those problem includes (1) how to covert an analog and digital signal into the Netduino, (2) how to records data into a storage media on Netduino, (3) how to communicates between the Netduino and data center using GSM network, (4) Are the analog and digital I/O channels on the Netduino sufficient to accomplish the needs of the research, and is there a technique to expands those number of channels.

2. Methodology

2.1 Research Framework

The ideal weather stations should be one system that is reliable to perform weather observations 24 hours a day. In this research, the steps that needed to be done in order to have a good performance of the AWS showed in Figure 1.
2.2 Software Architecture

This research will use microcontroller Netduino plus with .Net Micro Framework v.4.2. as the brain of the AWS. The microcontroller will be the center of multiple processes, including data acquisition, processing, storing and transmissions. Along with integrated telemetry database system, the data will be stored at datacenter. Main software of the logger was built in OOP (object oriented programming) concept. OOP was used to reduce the complexity of an application development process by divide it into smaller part, and formed in manageable class. Each class represents an idea, whether real or concrete [3] and [10]. Figure 2 shows the complete architecture of the main software in classes and function.

---

Figure 1. Workflow diagram of the Research

Figure 2. The Class Diagram of the AWS Data Logger
2.3 System Design

![Network Diagram](image)

**Figure 3. Applied Integrated Telemetry System**

This research was to develop an AWS which applied the telemetry system. Each station was designed to send the measured data to data center periodically. This telemetry system was relied on the existing GSM network, so in the other word the AWS can be implemented only on GSM covered area. The data was sent through SMS service, this will ease the development process but costly to applied continuously. Above shows the network topology, which will applied on the AWS system.

2.4 Data Records Method

Above picture was described the method of data collection, each data raw was recorded in every 10 minutes into storage media and send to server. Those data raw was an average value from several data samples. The averaging of data samples are using a moving average method, this was used to reduce data bouncing which caused by the electrical noise during analog to digital signal conversion.

2.5 Calibration and Validation

Calibration and validation is done by measuring the AWS weather data at a certain period and comparing it with digital voltmeter. Measurement data will be compared and analyzed using linear regression method to get the value or calibration factor. Then validation is performed by same way, but the data was taken at different times, the more variation in the measurement of different time, the more valid the data is generated. In this study, the calibration is performed only on the data logger, not including sensor calibration. Sensor calibration is done by comparing the potential difference recorded by the data logger and the actual potential. This can indicate a shift in the ratio of the value of the potential difference or the accuracy of the data.

![Activity Diagram](image)

**Figure 4. Data Logging Process in the Activity Diagram**
3. Research Result and Discussion

3.1 Programming

Main software of the data logger was built in C# language, by applying an OOP concept the software divided into some classes. These classes includes analog to digital conversion (ADC) class, real time clock (RTC) class, storage class, LCD display class, and GSM modem class.

3.1.1 Analog to Digital Converter Class

ADC on Netduino microcontroller is 10 bit type and 3.3 v maximum inputs (Secret Labs LLC, 2013). Therefore measured value on one cycles is 1024, below is the conversion equation.

\[ mV = \text{input} \times \frac{\text{max Analog Ref} \times 1000}{\text{max digital Value}} \]

Note:
- \( mV \): Output Voltage on millivolt unit
- Input: measured value on Netduino analog channel
- Max Analog Reference: maximum voltage (3.3v).
- Max Digital Value: 1023.

Measuring analog value using ADC on Netduino has some noise caused by the power supply; electromagnetics and static electricity surround it [1] and [2]. These factors causing the unstable result during inputs measurement, besides it’s also makes the accuracy became worse. On solving this problem, the moving average technique was applied to reducing the data noise. The moving average process takes 4 data sampling. This technique was done when data still on the integer value, below is the bit shifting method which applied to averaging the inputs.

\[ ADC_{average} = \sum_{i=0}^{n} \text{input} \gg \sqrt{n} \]

*\( n = 2,4,8,16\ldots \text{ Etc.} \)

Note:
- \( n \): Numbers of sampling (in this case, 4)
- \( \gg \): bit shifting Operator.

3.1.2 Main Looping

The Looping on this logger has 2 tasks to be done, includes data collecting and recording. There 3 sensors were used in this study, collecting data from those sensor were done by switching channel method. This method was developed to anticipate growth of the number of sensor in the future. Therefore, it’s an essential for an analog channel to observe more than 1 input. Below is the pseudo code of the switching channel method.

```c
if (_second % 15 == 0){
    if (_timeToSwitch){
        switcher.switch();
        if (switcher.channel == 0 || switcher.channel == 4) { switcher.channel = 0;
            _getDistinguisher = true;
            _timeToSwitch = false; }
        _timeToSwitch = true; }
    switch (switcher.channel) {
```
case 0:
    A0.Read();
    _currentReadValue = "Ch 0: " + A0.valueToString;
    break;

case 1:
    A0.Read();
    TempContainer.Collect(A0.value);
    _currentReadValue = "T: " + A0.valueToString;
    break;

case 2:
    A0.Read();
    HumidityContainer.Collect(A0.value);
    _currentReadValue = "RH: " + A0.valueToString;
    break;

case 3:
    A0.Read();
    AirPressContainer.Collect(A0.value);
    _currentReadValue = "P: " + A0.valueToString;
    break;

default:break; }

Observing a sensor was done in 15 seconds on each channel. This 15 seconds was considered by some factors include (1) the diversity of response time for each sensor to observes, (2) needs of some period to data sampling during observation, (3) period within processing signal of the electronic board was needed. Acquiring sequence in this technique require one channel to distinguish between first channel and the others. In this case, this channel was valued 2900mv as the 0 array index of the channels.

Later after collecting data, next task to do is record it into the data storage and send it to data center. A time stamp was used on this step, therefore it’s an essential to make sure the RTC (real time clock) has the appropriate and accurate time orientation. Each file was contained data for single day only, it’s to reduce load of file indexing during data records with append filing technique. Below is the pseudo code to record and send data.

```csharp
Logger.ContentFile = _Hour.ToString("HH:mm") + data;
if (!Logger.Write())
{
    myLcd.Write("Logging Failed");
    else {Logger.Reporting("Logging Failed");}
}
try{
    if (modem.Ready)
    {
        myLcd.Clear(); myLcd.SetCursorPosition(0, 1);
        myLcd.Write("sending data...");
        modem.SendSMS(Logger.NoDest, Logger.ContentFile);
        while (!modem.SmsSent)
        { Thread.Sleep(1000);}  
    }
    else{
        myLcd.Write("Data Send Error");
        Logger.Reporting("Data Send Error");
        Thread.Sleep(1000);  
    }
}
}
catch (Exception ex){ Logger.Reporting(ex.Message);}  

3.2 Electronics Design

Electronics used in this logger includes the microcontroller, GSM module and signal processing board. Microcontroller and GSM module shield are easy to use components, it makes they don’t need to be change at all. However to help this two board works as need, there is one electronics which purpose to ease the input and output signal, called signal processing board. The core of the signal processing board was includes LCD multiplexer
(IC 74HC595) and I/O expander (using IC4017 and 4066 in switching channel method). There above is the schematic diagram of signal processing board built in this research.

3.3 Data Analysis

Data analysis in this study was done by applied some tasks include noise reduction, data calibration and validation.

3.3.1 Noise Reduction

One of the treatment on data analysis in this study was done by setting up the analog input with a constant voltage, in this case the constant voltage is 0 mv (by put the ground into the analog channel). This treatment expresses some noise on data record shown by Figure 6.

Figure 5. The Electronic Schematic Diagram of Signal Processing Board
This unstable data reads caused by an inappropriate power supply applied (regular 5v Ac to DC adaptor). By applying an appropriate power supply and moving average method can reduce the noise. The analog input was steady on 0mv after the treatment. However, this study doesn’t observe the reason why the regular adaptor causing the data noise.

Common weather data doesn’t change significantly in short period, in example an air temperature is naturally impossible to change 1°C on a minute. Therefore, by applying moving average method on data series will not reduce its reliability, instead helping data reading.

Noise on data at figure above has multiplicity 3.3 mV caused by its data resolution, explained by calculation below:

\[
\text{Analog signal resolution} = \frac{\text{max Voltage Reference}}{\text{max bit value}}
\]

\[
= \frac{3300 \text{ mv}}{1023} = 3.23 \text{mV per count}
\]

The calculation above explain that smallest alteration unit given by ADC on Netduino is 3.23mV per count, however applying moving average method makes data reads has smaller resolution.

3.3.2 Calibration

Data calibration was done by comparing data read on Netduino and data shown on the voltmeter. This step was done to obtain the correction factor by applying linier regression method on data output. Below is the graph of the analog signal calibration.
Data series on figure above has 60 samples of data, it’s explain that correction factor for the analog signal is “0.88x + 4.031”. This equation can be used on the programming code to revise the data reads. The $R^2$ above describe that the equation can explain 86.44% of the data series entirely; it’s also used to prove the correlation level between data reads on the Netduino and data shown on the voltmeter. In the other words, the alteration value occurred not significantly and tolerated.

3.3.3 Validation

Analog signal data validation was done by the same step as data calibration process, but organized at differs time. Below is the graph of data validation.

In this validation process, the result of linier regression has a little different from result of calibration. Though it was differs, but if referred to ratio of differential gradient from both measurement which was valued 2.3% (1- (0.8594/0.88), then it can concluded that the differentiation occurs on both calibration and validation is near equal and tolerated.

3.4 Future Potential Problems and its Solutions.

The AWS data logger in this research can be developed with potential alternatives to do on the next research. This potential will grow as needs of the users.
3.4.1 I/O Digital

The solution resulted in this research was to solve the limitation of analog I/O on Netduino, however if the needs of usable digital I/O is rising then expansion of the digital I/O channel is necessary on the next research. In some literature, there’s information about this expansion technique. The MCP23S17 is 16 bit I/O expander with serial interface. With using a single chip of this component can expands the digital I/O channel into 16 nodes. Another benefit of the component is, it can be configured to operate individually or together [5]. The component was easy to apply; the Netduino forum has released the driver to operate this component with Netduino. However there must be an experiment to discover its benefit and its limitation to be applied on an AWS data logger.

3.4.2 Telemetry Technique Development

Telemetry technique by using SMS is popular since a long period. However the main problem in this technique is about the cost and the time delay which relatively slow [7]. Today there is a communication technique developed by IBM. This technology was applying a Machine to Machine (M2M) communication concept. This technology was popular within software developers as MQTT (Message Queuing Telemetry Transport). One of today popular chat engine “Facebook” was using this telecommunication technique [4].

By applying this technology into the logger, it is assumed to reduce the cost and to shorten the delay time. The focus to learn this technique is how to make the MQTT work in the existing GSM network. It’s supposed to lessen the research and implementation cost.

3.4.3 Lack of Power Resource as the Rise of Number of Sensors

Escalation of the number of sensors may occur to fulfill the needs of micrometeorology observation. Therefore, it is very possible for the logger will lack of power. It will risky when the power less of charge during cloudy weather and can’t stand in the night. To prevent these problems there must be a research about power consume of the sensors. Possible solution can be applied to handle the load is by interchanging within the sensors to turn on and off during observation. In that case, a research about sensor characteristic is necessary to discover the diversity of response time for each sensor during the observation.

4. Conclusions

Analog signal conversion can be done easily, however it’s must concern that the data reads has a noise when inappropriate power supply is applied. Therefore, it has to moving averaging for each data reads and there must a correction factor to regulate the data to become near to its real value.

In this study, the correction factor was resulted by calibration and validation if the data. It can be done by comparing the data output on Netduino and voltmeter, and applying linier regression later on it. The correction factor resulted was “0.8594x + 4.7486” with $R^2 = 0.9367$. The $R^2$ value means to the correlation level between the data records on Netduino and the voltmeter is 93.67%. The correction factor resulted during validation has an alteration about 2.3% from the calibration; it’s mean that both the correction factors are near equal and tolerated.

The data was recorded into storage in WMO standard period for weather data, in every 10 minutes. In data recording process there must be a class to handle the naming file and folder to prevent error during the recording process. To reduce the load during file indexing at the recording process, it’s better for the file and the folder to split up based on
days and months. Besides it is necessary to back up the data in several periods for the same reason.

The weakness of the Netduino microcontroller was the limitation of numbers of the analog and digital I/O channels. However, this study has resolve the problem by developed a switching channel method. The method was developed by using the component of 4017 and 4066; this method can expand the analog I/O to reach 4 channels for each pin on the Netduino.

Telemetry system was applied on the remote station, and then it will ease the user to obtain the weather data. This telemetry system was relied on the existing GSM services, the short message service. By using existing GSM network, it assumed to reduce the research and implementation cost.

The budget of this AWS data logger development has 7 to 5 times cheaper than the price of the AWS data logger from overseas manufactures. It means that the implementation of the AWS data logger will have lower cost in the future. It is one of the purposes in this research.

References