

# Design and Development of Meteorological Data Logger

Yin-Yeo Ng, Soo-Hong Park\*, *Member, KIMICS*

**Abstract**— In this paper, an effort has been made to design and develop a meteorological data logger for meteorological purpose. This data logger is proposed to be included various sensor interface that used in weather sensors. Besides, numbers of meteorological process libraries are added into this data logger to make it able to perform as unattended weather monitoring system. Data output of this data logger are also design to support multiple protocol that commonly used in data logger, and several communication devices that commonly used in the market. Each data that logged will be logged together with date and time and able to retrieve via serial port using hyper terminal. It is also configurable via serial port.

**Index Terms**—Data Logger, Meteorological Processes, Analog-to-Digital Converter

## I. INTRODUCTION

A data logger system is application that involves acquiring data, performing some analysis, storing it and then recalling it at later point for some type of analysis and presentation [1]. There are varieties of data loggers that perform task for various applications such as weather station data recording, offshore buoys for environmental condition recording and road traffic monitoring.

The primary objective in this research is to design and develop a data logger which serves as a standalone environmental data acquisition and logging for meteorological purpose. In this paper, general system description is presented first, along with the overview of the electronics and firmware design. Sensor interfaces is then introduced and described. Also, details of data processing and data output specification is addressed. Finally, conclusion of this paper is presented.

Manuscript received October 13, 2010; revised November 1, 2010; accepted November 10, 2010.

YinYeo Ng is a student who is studying under the Department of Mechatronics, Dongseo University, Busan, 617-716, Korea (Email: yeoyeo85@yahoo.com)

SooHong Park\* (Corresponding author) is Professor in the Dept. of Mechatronics Engineering of the Dongseo University, Jurye-dong, Sasang-gu, Pusan, 617-716, Korea (Tel: +82-11-849-1765, Email: shpark@dongseo.ac.kr)

## II. SYSTEM DESCRIPTION

### A. Functional Overview

This Meteorological Data Logger (MDL) is a 32-bit data acquisition system designed to collect, process, store, and transmit data from multiple sensors. Overall, the features of MDL can be divided into three main groups which are describes as follows:

- a. Collect data from sensors:  
Capture signal from various interfaces (analog, digital or serial interface), and perform some basic preprocessing on data captured.
- b. Process the collected data:  
Perform some processing if necessary. This logger provides some environmental related processing libraries that allow user to process the collected data in this logger.
- c. Store and/or transmit the processed and collected data:  
The final output of this logger can be configured as log to memory or transmit via specific interface. Users are allowed to choose either one of both of them.

### B. Electronics and Firmware Design

#### 1) Overview

Basic hardware interfaces and function can easily understand via illustration as Fig. 1. MDL contains two types of analog input which are 12-bit high speed ADC and 16-bit high resolution low speed ADC. Both of them serve the same purpose that convert analog signal to digital data. Details will be discussed in section II.B.2. Besides, digital interfaces are included in MDL. It supports fundamental digital input/output and also frequency/pulse counter operation. Due to there are many sensors come with serial interface, MDL was designed to consist of several channels of serial port. Apart from reading serial sensor, serial port is also serving as output port of MDL. Final output data will be output via serial port. In addition, it can be configured to communicate with some modem which using serial port as communication interface. Hence, transmitting data via communication modem is possible with proper setting of MDL. As a result of saving power, regulated switched power was added in MDL design. User can choose to on/off the sensor's power during appropriate time to achieve power saving purpose.

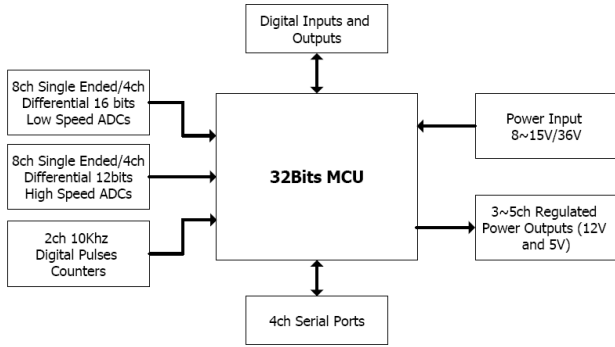


Fig. 1. MDL's Functional Block Diagram

## 2) Sensor Interfaces

There are two Analog-to-Digital Converter (ADC) installed in MDL which enable MDL to convert analog signal from sensor to digital value. They are 12-bit ADC and 16-bit ADC. 12-bit ADC is strictly operates with unipolar (single-ended) mode, where 16-bit ADC supports bipolar (differential) measurements. It is user responsibility to determine which ADC is suitable for their purpose. Table I illustrates the specification of both ADCs.

TABLE I  
MDL'S ADC SPECIFICATION

	12-bit ADC	18-bit ADC
Resolution/Step	12bits/4096	15bits/32768 + 1 sign bit
Analog Input	Unipolar	Unipolar/Bipolar
Voltage Range	0 to 5V DC	-5V to +5V DC
Per-bit Accuracy	1.22mV DC	0.1526mV DC

In addition, both ADCs support ratio metric and non-ratio metric measurements. Fig. 2 shows a non-ratiometric measurement technique. This configuration is suitable for applications requiring measurements against an absolute reference, or a single ADC serves as several unrelated analog inputs. Non ratiometric is not applicable during some circumstances, especially to a resistive sensor where sensor's output is proportional to sensor's excitation voltage input. Change of excitation voltage will cause change of sensor's output that can be seen by ADC. Ratiometric measurement technique was designed to measure resistive based sensor without a complicated circuits as illustrated in Fig. 3. This design does not use an absolute voltage as voltage reference but derived from excitation voltage. Thus any change from excitation will cause proportional change of ADC's reference voltage and sensor's output. For this reason, it is suitable for resistive sensor (e.g. thermistor).

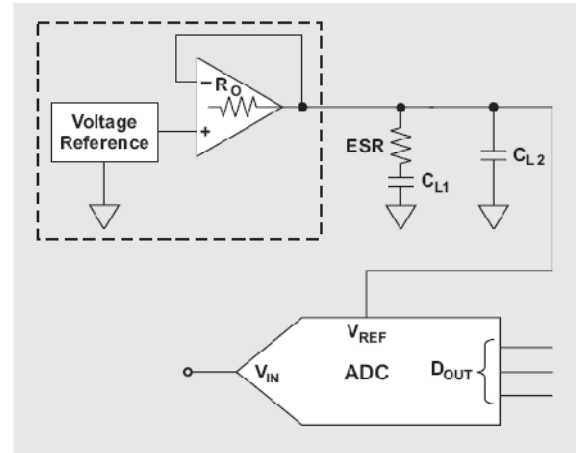


Fig. 2. Configuration of a Non Ratiometric Measurement System [3]

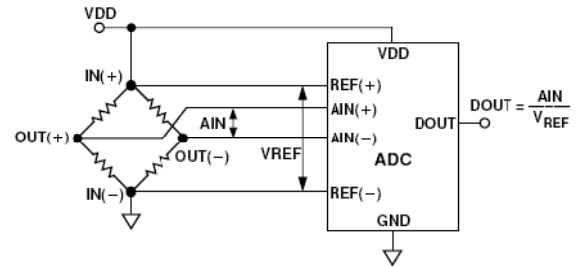


Fig. 3. Configuration of a Ratiometric Measurement System [2]

For ratiometric measurement, MDL support multiple excitation voltages which are 1.25V, 2.5V and 5.0V DC. It is user's responsibility to determine which excitation voltage to be used so that the excitation current through the sensor must be large enough so that the smallest temperature change to be measured results in a voltage change that exceeds the system noise, offset, drift of the system, at the same time [2], it will not cause self-heating problem if using resistive sensors.

It is common for the voltages to vary until a steady state is reached during an analog sensor is powered up. The amount of time for the voltages to reach steady state depends upon the electronics components within the circuit. Taking a measurement before steady state might give an inaccurate measurement, hence it is necessary to wait sensor to wait for several milliseconds for an analog sensors to be fully settled after it is powered [4]. MDL provides delay mechanism that allow user to configure ADC interface to operate in such way. User can setup a delay within 0 milliseconds to 100 milliseconds with step of 10 milliseconds. With all this features, most of analog sensors available for environmental used can be work together with MDL.

Apart from analog interface, MDL does contain digital sensor interface that provide a way to measure digital signal. It can be configure to be a simple on/off state, digital frequency measurement, digital period measurement, edge capture and quadrature encoder input. Table II presents various operation modes and corresponding specification of each mode of digital sensor interface.

TABLE II  
DIGITAL INTERFACE'S OPERATION MODE AND ITS SPECIFICS

Operation Mode	Specifics
Digital Frequency Measurement	Measuring frequency of digital signal. Shortest measurable interval is 0.25us = 4MHz.
Digital Period Measurement	Measuring high period of input pulses. Shortest measurable interval is 0.25us.
Edge Capture	Capture rising/falling edge of pulses. A 16-bits accumulator support up to 65536 edges can e counted. If more than 65536 pulses counted, accumulator will reset to 0 and start counting again.
Quadrature Encoder Input	Specifically for differential encoder. An 16-bits pulse accumulator support 32767 pulses counted in single direction.
On/Off State	Test is input on or off only.

Aside from above, serial sensor interface is also supported by MDL. Generally serial sensors can be classified into two general categories:

- Polled: Serial sensor transmit data only after receiving specific instruction to do so.
- Asynchronous: Serial sensor transmits data on its own. Usually with a specific interval time.

MDL supports both polled and asynchronous serial sensors. User need to set proper instruction to send before receiving data from polled sensor while data from asynchronous sensor will be received automatically by MDL and specified amount of time is allotted for MDL to wait to receive from serial sensor.

Serial can also be classified according to their message format such as NMEA or SDI-12 standard. Even though, manufacturers are not limit to use standard among these two standards only, they are free to choose its own message format. Many choose their own proprietary format as sensor output [4]. MDL is designed to complaint with various serial sensor formats as follow:

- NMEA 0183: This is a interface protocol created by National Marine Electronics Association. It defines electrical signal requirements, data transmission protocol and time, and specific sentence formats for a 4800 baud serial data bus. The data is in printable ASCII format [5].
- SDI-12: It is acronym stands for Serial Data Interface at 1200 baud. It is a standard to interface

battery powered data recorder with micro-processor based sensor designed for Environmental Data Acquisition (EDA) [6].

- Proprietary: This represent any message format which is not belongs to any known message protocol. User need to specify message format for MDL using formatted string in C printf function.

Besides reading from serial sensors, serial interfaces are also serving as communication interface for MDL's access, data collection/output, and communication purpose. CDMA modem is one of the communication devices which able to connect directly to MDL as data transmission devices. Usually, a CDMA modem is able to support both SMS and TCP/IP service, data can be sent via both SMS and TCP/IP. Since there is no standard interfacing method for CDMA modem, thus, not all CDMA modems are supported by MDL. RCU890 is one of the modem can be used with MDL. Fig. 4 demonstrates example of CDMA modem's (Model: RCU890) TCP/IP operation. Different CDMA modem will have different operation and commands [7].

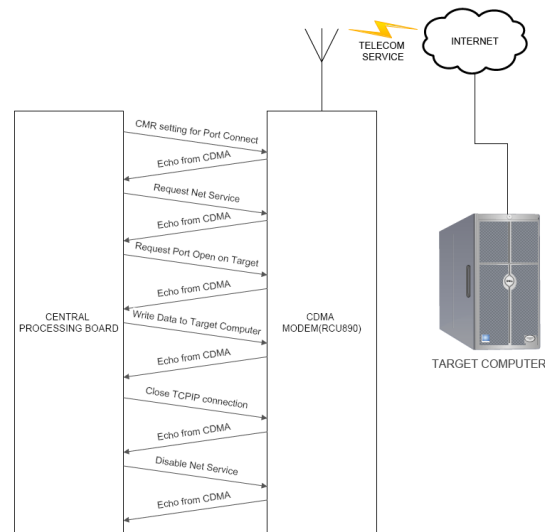


Fig. 4. TCP/IP Operation of CDMA Modem

### 3) Data Processes

As environmental data logger, MDL provides a large library of processes which allow user to process the collected data in various ways. All processes are running throughout the sample interval times rather than being calculated at the end of the sample duration, hence output of processes remains valid throughout the sample interval times. Generally, processes can be categorized into few categories which are:

- General Processes: Processes that perform simple averaging, rate of change, etc.
- Environmental Processes: Processes that calculates environment data such as vector average wind speed and direction, dew point, significant wave height, etc.

- Exception Processes: Processes that perform analysis on input data to determine if exception occur. Example of exception is value out of range, bad sensor value, setting error flags, etc.
- Arithmetic Processes: Processes that perform arithmetic operations such as addition, subtraction, multiplication, division Boolean operations, etc.

Following list some commonly used meteorological processes that included in MDL.

- Averaging Process: Calculating mean and standard deviation of a set of data.

$$x_{avg} = \frac{1}{n} \sum_{i=1}^n x_i \quad (1)$$

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - x_{avg})^2} \quad (2)$$

- Wind Vector Average Process: Vector average for wind speed and direction, and approximate standard deviation of wind direction using Yamartino equation [8].

$$x_i = s_i \cos \theta_i \quad (3)$$

$$y_i = s_i \sin \theta_i \quad (4)$$

$$\bar{x}_i = \frac{1}{n} \sum_{i=1}^n x_i \quad (5)$$

$$\bar{y}_i = \frac{1}{n} \sum_{i=1}^n y_i \quad (6)$$

$$s_v = \sqrt{(\bar{x}_i^2 + \bar{y}_i^2)} \quad (7)$$

$$\theta_v = \tan^{-1}(\bar{y}_i / \bar{x}_i) \quad (8)$$

$$\sigma_\theta = \sin^{-1} \left[ \varepsilon \left( 1 + \frac{2}{\sqrt{3}} \varepsilon^3 \right) \right] \quad (9)$$

Where:

$$\varepsilon = \sqrt{1 - \left[ \left( \frac{1}{n} \sum_{i=1}^n \cos \theta_i \right)^2 + \left( \frac{1}{n} \sum_{i=1}^n \sin \theta_i \right)^2 \right]} \quad (10)$$

- U-V vector to Magnitude-Direction Process:

This process takes the square root of the squares of the two input values to produce the vector magnitude (M), and takes the arc-tangent (V/U) to produce the vector direction (D).

$$\begin{aligned} M &= \sqrt{U^2 + V^2} \\ D &= \tan^{-1} \left( \frac{V}{U} \right) \end{aligned} \quad (11)$$

#### 4) Data Output

MDL is designed to have flexible output. User is allowed to define data fields individually to the data output. Data fields can be output of a sensor, output of a process, date or time and literal string that defined by user. Besides, in combination with time function, data fields can be conditionally added to a data output message. Data that output via serial port will be printed as ASCII text format.

Furthermore, to optimize memory space and determine data to be seen or hidden, data fields can be set to operate among one of these three modes:

- Data fields that transmit via output port only.
- Data fields that to be logged into memory only.
- Data fields that to be transmitted and logged at the same time.

As an example, total of 20 data fields are possible to logged or transmitted in MDL. It can be configured so data fields #1, #2, #3, #10, #15 and #18 are logged into memory where data fields #1, #2, ..., #20 are transmitted via serial port. When data retrieved from memory, only data fields #1, #2, #3, #10, #15 and #18 are shown. To optimize MDL's memory space, data that logged into memory are saved as binary format and will be printed as ASCII text format when there is data retrieval command.

### III. RESULT

To verify the calculation method, an experiment had been performed. This experiment used MDL to captured temperature reading from HMP45A temperature sensor. At the same time, another temperature data logger unit (PTU303) was used to compare both data. Besides, wave height sensor was simulated using function generator.

The following is the setup and configuration of testing platform:

MDL:

- HMP45A on 24bit ADC interface. Sampling rate 1Hz.

- Output from function generator on 12bit ADC interface. Sampling rate 1 Hz
- Average process of 200 data on temperature reading.
- Significant wave height and period calculation on 1024 wave data.

#### PTU303 Data Logger:

- Attached with a temperature probe. Data output from serial interface with sampling rate 1Hz.

#### Function Generator

- Simulate wave sensor, output a wave form with the following equation:

$$y = \sin(x) + 0.45 \sin(2x + 60) \quad (12)$$

To verify the final outcome, all captured and calculated data were sent to PC via serial port for further analysis. Microsoft Excel Spreadsheet and Matlab program are used in analysis the data. Finally comparison was made between MDL, PC and PTU303 to verify the accuracy and precision of MDL.

Fig. 5 shows the temperature reading of MDL and PTU303 versus length of sampled data. Fig. 6 shows the simulated input data for wave height sensor. Table III illustrates the final calculated average and standard deviation of temperature data using Microsoft Excel and MDL and Table IV compares wave height and wave period which calculated by MDL, Matlab and Wave Analysis for Fatigue and Oceanography (WAFO) Toolbox for Matlab.

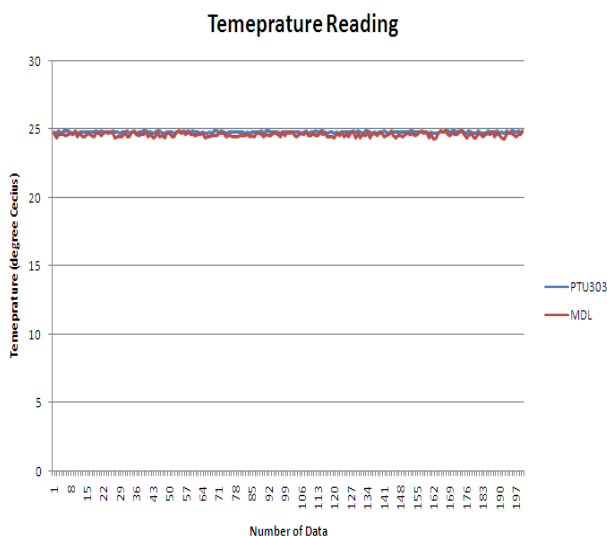


Fig. 5. Temperature versus Length of data

TABLE III  
COMPARISON BETWEEN EXCEL AND MDL

Parameter	Result (Excel)	Result (MDL)
Average over 200 data	24.61	24.61
Standard deviation over 200 data	0.122	0.122

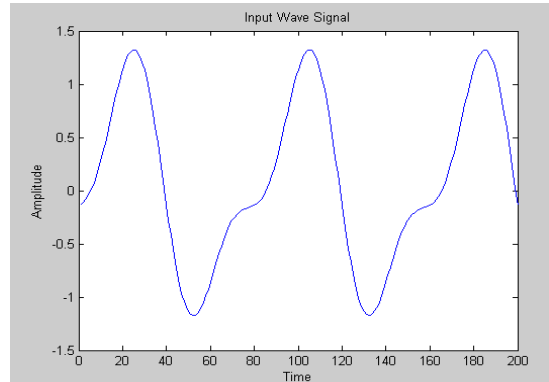


Fig. 6. Input Waveform for Wave Height and Period Simulation

TABLE IV  
COMPARISON BETWEEN MATLAB AND MDL

Parameter	Wave Height (m)	Wave Period (s)
Matlab (WAFO)	3.1093	80.7678
Matlab (Custom Algorithm)	3.1016	81.3711
MDL (Custom Algorithm)	3.1016	81.4226

## IV. DISCUSSION AND CONCLUSION

From the result, MDL perform well in standard arithmetic calculation such as addition, subtraction, multiplication and division. This can be seen from the outcome from MDL which perform average and standard deviation process is almost identical with the outcome from Microsoft Excel.

For wave height calculation, there is slightly different from MDL and Matlab's result. The main reason is the algorithm which used in WAFO toolbox and MDL is different. As the result, the performance still within satisfies range due the result is the same if the same algorithm is performing using Matlab.

There are still some problems exist in MDL. The main problem is power consumption. Since MDL still in developing progress, the total power consumption still cannot be determined. In the other hand, meteorological processes that supported by MDL are not sufficient to function as a standalone meteorological data logger. More meteorological processes will be added to MDL in the future.

## ACKNOWLEDGEMENT

This work is resulted from the project of Shindong Company & Marine Craft (2009-2011) financially supported by the DongNam Economic Area ([www.dlio.leading.or.kr](http://www.dlio.leading.or.kr)).

## REFERENCES

- [1] Developing Data Logger Applications with LabVIEW. [Online]. Available: <http://zone.ni.com/devzone/cda/tut/p/id/3263>
- [2] Albert O' Grady. (August-September 2000). Transducer/Sensor Excitation and Measurement Techniques [Online]. Available: <http://www.analog.com/library/analogDialogue/archives/34-05/sensor/>
- [3] Bonnie Baker, and Miro Oljaca. (June 2009). How the voltage reference affects ADC performance, Part 1. [Online]. Available: <http://focus.ti.com/lit/an/slyt331/slyt331.pdf>
- [4] Micheal J. Hart, ZENO-3200 User Manual Version 2.02:Coastal Environmental Systems, Inc., pp.155, 2003.(Book Style)
- [5] NMEA 0183 Standard V4.00. [Online]. Available: [http://www.nmea.org/content/nmea\\_standards/nmea\\_083\\_v\\_400.asp](http://www.nmea.org/content/nmea_standards/nmea_083_v_400.asp)
- [6] SDI-12 Support Group. [Online]. Available: <http://www.sdi-12.org/>
- [7] Lee Chee-Cheong, and Park Soo-Hong, "Different Approaches to Design a Meteorological Buoy for Weather Monitoring Purpose," unpublished.
- [8] R. J. Yamartino, "A comparison of several Single-Pass Estimators of the Standard Deviation of Wind Speed," *Journal of Climate and Applied Meteorology.*, vol. 23, pp.1362-1366, 14 May 1984. (Transaction Journals style)
- [9] Walter J. Saucier, *Principles of Meteorological Analysis*: Dover Publications, May 1989. (Book style)



**Ng Yin Yeo** is a Malaysian citizen who born on 13/7/1985. He graduated as Bachelor Degree in BEng (Hons) Electronic Engineering in year 2008 at Multimedia University (<http://www.mmu.edu.my>), Malaysia. Also he obtained his master Degree in Mechatronics Engineering in year 2007 at Dongseo University, South Korea.



**Soohong Park** (M'2006) is a Senior Professor who works under Mechatronics Department at Dongseo University. He obtained his Bachelor Degree at year 1986, master degree at year 1989 and Ph.D. degree at year 1993. During 1995-1996, he was visit professor at the Beijing Aero & Astrometry University, China. In year 2002-2003, he also was visit professor at the Oregon University, U.S.A.. His main research topics are control and unmanned vehicles and robot.