VOL. 10, NO. 15, AUGUST 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences


www.arpnjournals.com

MULTIPLEXING OF UNFORMATTED ASYNCHRONOUS SERIAL DATA TO A SINGLE NMEA 0183 PROTOCOL COMPLIANCE DATA-STREAM FOR AUTONOMOUS SURFACE VEHICLES DATA LOGGING

M. H. Mat Idris, M. I. Sahalan, M. A. Abdullah and Z. Z. Abidin
Autonomous Agent Research Group (AARG), Department of Mechatronics Engineering.
Kulliyyah of Engineering, International Islamic University of Malaysia, Malaysia
E-Mail: zzulkifli@iium.edu.my

ABSTRACT

Development of autonomous surface vehicle (ASV) for hydrography survey requires varieties of transducers to be setup. However, several of the commercial of the shelf (COTS) transducers are using a different format of data transmission. Therefore, there is a need for built-in communication protocols for seamless integration to your monitoring platform and data logging. This paper focuses on the practical implementation of the technique to convert unformatted asynchronous serial data from transducers into NMEA 0183 protocol. NMEA 0183 protocol is being selected as the common protocol since it is used by many marine vehicles. The technique lies on extracting the non-formatted asynchronous serial data and generates a custom made NMEA sentence that complies with the format. By implementing such technique, the data will not only synchronize in a single format but also stamp with the global positioning system (GPS) data which is crucial for mapping.

Keywords: autonomous surface vehicle; unmanned surface vehicle; hydrography; NMEA 0183; Asynchronous serial data; Arduino; AtlasScientific.

INTRODUCTION

The motive of this development is a part and parcel of multi-agent autonomous surface vehicle (ASV), AquaDrone for hydrography applications. Autonomous surface vehicle (ASV) can be defined as a marine surface vehicle without any pilot on board that able to manoeuvre autonomously by an intelligent system. Another common term related to ASV is unmanned surface vehicle (USV). Respect to its purpose, hydrography is a study of the diverse measurement of water qualities which includes bathymetry, pH, temperature, salinity etc. Many other research institutes are developing ASVs for similar purposes. [1-5] ROAZ had been developed in Portugal for risk assessment for shallow water environments and water-land interface zones [1]. SESAMO was deployed at Lake Zurich for monitoring algal blooms [4]. Hence, Universiti Teknologi Malaysia (UTM) had developed as jet-ski USV for sea patrol and environmental monitoring [5]. These mentioned ASV required the fundamental instruments for navigation (e.g. global positioning system (GPS), inertial measurement unit (IMU), and magnetic compass) including hydrography transducers (e.g. sonar sensor, Hydrolab, YSI Sonde, AtlasScientific etc.). Therefore, multiple types of transducers need to integrate to satisfy a hydrography survey operation and later required for data logging.

The major issue highlighted in this paper is the incompatibility of different data format from multiple navigation and sensing devices. Whenever data are not altogether in a single format they are not synchronize together which cause difficulty for plotting and analysis. This certainly occurs whenever it includes non-marine devices, such as laboratory instruments to be used in the ASV functionality. In marine practices, the sensing and navigation electronic data transmit via a certain protocol. The most common marine protocol is National Marine Electronics Association (NMEA) which will be discussed on section 2.2. The reason of protocol is for ease data integration such as GPS coordinates, heading, and speed which are essential for marine navigation as well as synchronize data logging. Since laboratory equipment (e.g. water quality measurement sensors) is mounted to AquaDrone for bathymetry operation, there is a need for seamless integration of data collection. There are other protocols available such as Sea Talk, FastNet, NavNet, SDML and etc. which develop by marine electronics equipment makers. However, this paper focuses on one protocol which is NMEA 0183.

In the development of AquaDrone, the problem incurs when there are a differences of data format between the equipment; non-formatted asynchronous serial data and NMEA 0183 data. To resolve this issue, either format has to be converted to another. NMEA 0183 is selected in this case as the convention since NMEA 0183 protocol are widely used by marine transportation and equipment. By synchronizing the format, this will ease the process of monitoring the oncoming data on a single terminal and data logging. The GPS coordinates stamp of the data in NMEA 0183 highly aids in allocating the position which the parameter was picked up. This resolves the problem of the untraceable long list of recorded water parameter. Arduino Uno open source microcontroller [6] is used as the platform for programming the conversion algorithm. The objective of the paper is to discuss the algorithm of converting non-formatted asynchronous serial data into NMEA 0183 using Arduino platform.

BACKGROUND

Asynchronous Serial

Asynchronous serial data communication is a serial communications protocol that a start signal is sent prior to each byte, character or code word and a stop signal is sent after each code word [7]. This character differentiates asynchronous with synchronous as the latter does not have a start and stop bits but it transmission is synchronized with the clock speed. The start signal serves to prepare the receiving mechanism for the reception and registration of a symbol and the stop signal serves to bring the receiving mechanism to rest in preparation for the reception of the next symbol. A sample of asynchronous serial data diagram is shown in Figure-1.

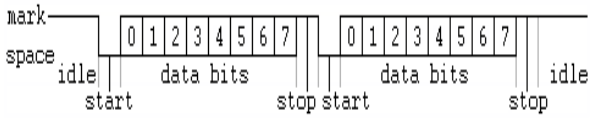

Figure-1. Asynchronous serial data bit diagram.

A character that relates to the reading of the asynchronous serial data is <CR>, carriage return. This character is looked up in the ASCII table as decimal value 13. Carriage return is derived from the classical term in typewriter machine which now refer to reset a device's position to the beginning of a line of text [8].This character appears after every ending from each transducer that indicates is the end of its parameter.

NMEA Format

NMEA 0183 is a combined electrical and data specification for communication between marine electronics such as GPS receivers, anemometer, autopilot, sonars, gyrocompass, and many other instruments. It has been defined by and is controlled by the National Marine Electronics Association (in which the abbreviation derived from).

The NMEA 0183 standard uses a simple ASCII, serial communications protocol that defines how data are transmitted in a "sentence" from one "talker" to multiple "listeners" at a time. Through the use of intermediate expanders, a talker can have a unidirectional conversation with a nearly unlimited number of listeners, and using multiplexers, multiple sensors can talk to a single computer port.

```
$GPAAM,A,A,0.10,N,WPTNME*32
(a)
$PGRMZ,2062,f,3*2D
(b)
```

Figure-2. (a) NMEA 0183 defined name (b) Propriety NMEA 0183 name.

The format of a NMEA 0183 sentence is shown in Figure-2. Every NMEA 0183 sentence begins with the dollar sign '**$**' followed by codes of capitalized letters. There are two kinds of rules for the letters; NMEA 0183 standard sentences (Figure-2(a)) which used by all NMEA 0183 devices and propriety sentences (Figure-2(b)) which set by hardware manufacturer for the own defined sentences. For standard, the first two letters representing the talker ID and next three letters indicates the parameter carries along with it. (e.g. $GPAAM implies global positioning system; arrival alarm information). While the propriety sentence must begin with the letter '**P**' followed by three letters that identify the manufacturer controlling that sentence. The coded letters is then continued by the measured parameter from the transducer until the asterisk '*'. Each parameter is separated by a comma '**,**'. Further details of the measurement unit of the parameter should refer to the manual of NMEA 0183 protocol. At the end of the sentence is the checksum of the line. The checksum is the bytes summation of XOR operations from the coded letter until the character before the asterisk symbol. The concept will be explained in section 3.

The latest generation of NMEA is NMEA 2000 that has the specialty of multiple talkers and multiple listeners. Although NMEA 0183 is not multiple talkers by itself but it can be upgraded so by using a multiplexer among the multiple NMEA 0183 devices. The concept of NMEA 2000 is the Controller Area Network ("CAN bus"). The CAN bus is a backbone shape like which all NMEA 2000 components are plugged, both power and devices. The architecture of CAN does provide a simpler connection for a sail boat however for a small size drone the CAN bus takes much length that may not fit for drone. Conventional wiring fitting in NMEA 0183 works better in limited length and space compare to NMEA 2000. Thus, the main issue to be highlighted in this paper is not the architecture of the system but how does different format of serial data can be merged into one common protocol for ease of data stamping and logging.

ALGORITHM

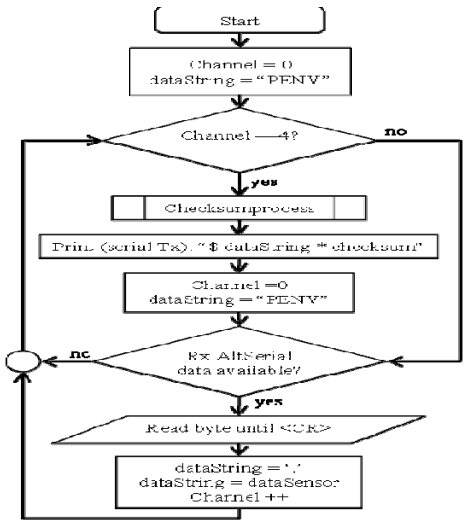

Figure-3. Asynchronous data conversion into NMEA 0183 flow chart.

This algorithm is implemented on Arduino Uno microcontroller and programmed through its IDE (integrated development environment). The flow chart of the conversion is shown in Figure-3. It is related to the designated sensors used by AquaDrone ASV. The flow starts with the initial setup for the channel selector (Channel = 0) and the defined propriety name in a string variable (dataString) for NMEA 0183 device and parameter field identification. The flow requires the program to check on the channel number using **if** flow control statement prior collecting data to ensure that once the channel had reached its maximum number of channel (Channel = 4) then it will transmit the NMEA 0183 sentence. Other users may define the maximum value according to the amount of transducers used for conversion.

Yet for the first cycle this **if the** condition is skipped and proceeds to the serial read. From the AltSoftSerial library [9], which required by Arduino UNO for additional serial communication port, it reads the asynchronous serial data whenever data is available to a temporary sensor string (dataSensor) until it finds <**CR**>. It adds comma delimited '**,**' to dataString before including the transducer dataSensor reading to separate between NMEA 0183 propriety name and other transducer parameters. Then dataString includes the value from dataSensor and increment Channel value. Since the transducers' serial port connector uses 2 digital pin 6 and 7 for channel control (shown in Figure-5), Channel is used as the expression in switch-case flow control statement to manipulate the channel control (e.g. 00,01,10,11). The amount of channel control pins depend on the design of multiplexer used; the more input channel of the multiplexer requires more channel control pins.

The loop returns to the first **if** statement. If Channel has not reached the maximum value, the previous flow continues. Once Channel = 4, it generates the checksum of the sentence as shown in Figure-4. The checksum function generates summation XOR bit operations from every byte in dataString and produces the hex value. It uses '^' binary operation in **for** loop flow control statement, starts by checksum=0 with the first byte in dataString and continues until it reaches the last byte, dataString. length().

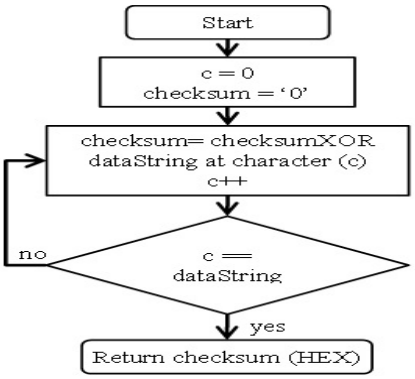

Figure-4. Checksum operation flow chart.

Finally the program prints out the NMEA 0183 sentence format by printing in sequent '**$**', dataString, '*****', and checksum in HEX. Channel is resets to 0 and dataString="PENV".

SETUP

Figure-5 shows the schematic setup of the Arduino to AtlasScientific serial port connector and RS232 port which connects to NMEA 0183 combiner. Apart from the power supply connection (Vin and GND), pin 8 and pin 9 are utilized as serial communication to AtlasScientific serial port connector's Tx and Rx respectively. By adding AltSoftSerial.h library and instantiate an object from the library, pin 8 and 9 are automatically convert into serial communication port. Pin 0 and pin 1 of Arduino Uno is also a serial communications port and connected to female RS232 data transmit (pin 3) and data received (pin 2) respectively. The serial port connector requires 2 output direction control signals (S0 and S1) which connected to pin 6 and pin 7 of Arduino.

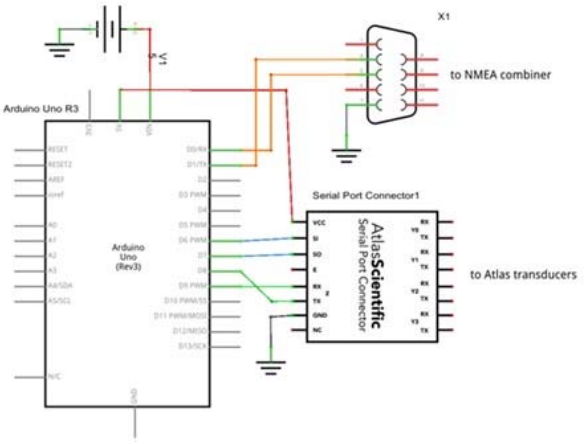

Figure-5. Arduino to AtlasScientific serial port connector and RS232 output.

RESULTS

In developing AquaDrone hydrography survey ASV, four AtlasScientific water probes (e.g. pH, conductivity, dissolve oxygen, and oxidation-reduction potential) non-formatted asynchronous serial data were fused into NMEA 0183 protocol by linking the transducer as in figure 6. Apart from the mentioned transducers, GPS, electronic compass, and depth sounder with NMEA 0183 format are mounted to the ASV for navigation and sensing.

AquaDrone had been tested together with the sensor fusion box (Figure-7(a)) at Taman Layang-layang Lake (Figure-7 (b) and at Maryam lake, IIUM. The total speed response of the data logging is 1Hz according to the frequency of the GPS and the rest of the NMEA 0183 devices even though the AltasScientific probe was operating at 4800 baud rate. Figure-8 exhibits a sample reading of the serial data controller and the reading after fusing with the other NMEA 0183 data. According to the rule of NMEA, this sentence is categorized as non-

standard NMEA 0183 message a.k.a. propriety sentence. Therefore the code "PENV" is used as to refer to "propriety environment".

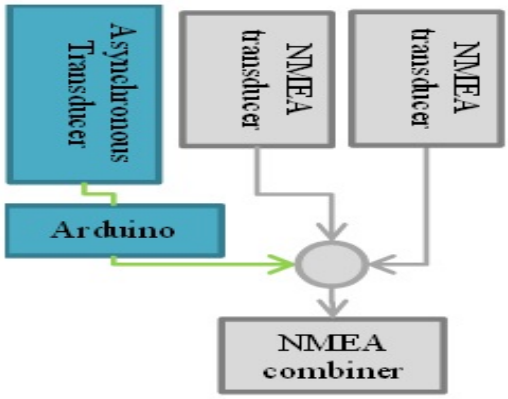

Figure-6. Architecture of transducers data logging.

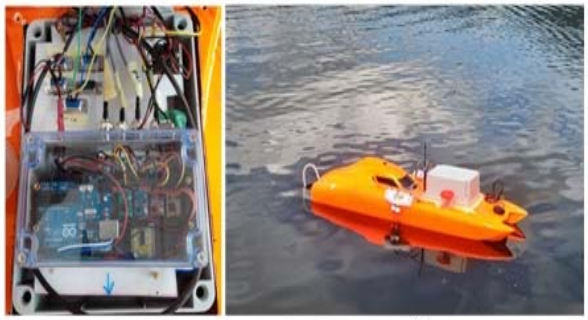

Figure-7. (a) Data fusion box (b) AquaDrone in operation at Taman Layang-layang lake side.

```
//Channel 0 @ pH
7.00
//Channel 1 @ Dissolve Oxygen
8.19
//Channel 2 @ Oxidation Reduction Potential
272.28
//Channel 3 @ Conductivity
12.36,6,0.00,1.00
```

```
$GPGSV,3,3,11,30,30,175,35,07,05,153,00,09,13,119,00*4B
$PENV,7.00,8.19,272.28,12.36,6,0.00,1.000*51
$SDMTW,023.1,C*34
```

Figure-8. Result of conversion of 4 AtlasScientific asynchronous serial data transducers into a NMEA 0183 sentence.

A test on the data transmission was performed to evaluate the performance of the conversion. 1641 lines of NMEA 0183 sentences were collected within 27 minutes. The delayed time between the input transmission from Atlas sensor and the output of the NMEA 0183 sentence were measured.

It is found that the delay time between the input and output resembled a decline saw tooth waveform as shown Figure-9(a). Seemingly the waveform pattern in the scatter plot diagram has a gap between 1500 and 1000 millisecond. However, the data was then rebuilt on a line plot as in Figure-9(b) showed that the waveform start at the average peak of 1040 millisecond and average trough of 525 milliseconds. The high values of delays (1500-1900) were sudden ripples that occurred while the graph was declining. A single waveform of the saw tooth is in the average of 3.17 minutes which is 191 NMEA lines per saw tooth. Figure-10 shows how the ideal sawtooth waveform for this system.

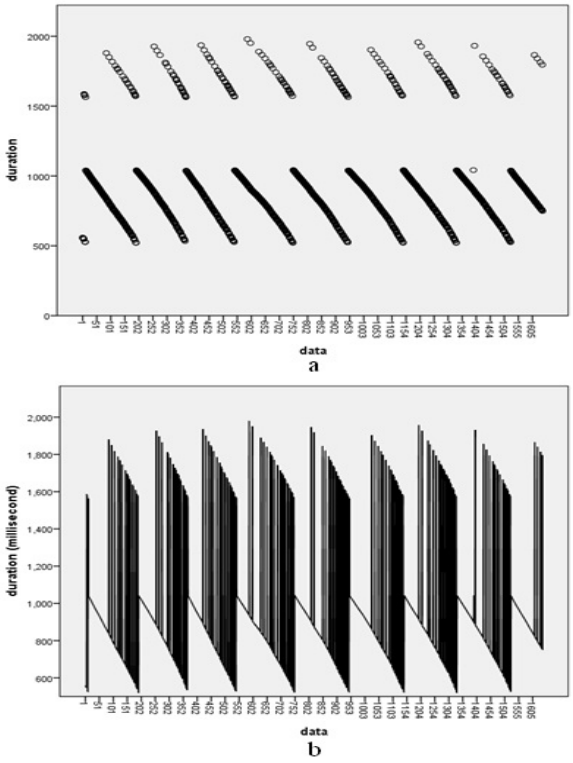

Figure-9. (a) Input to output delay scatter plot (b) input to output line plot.

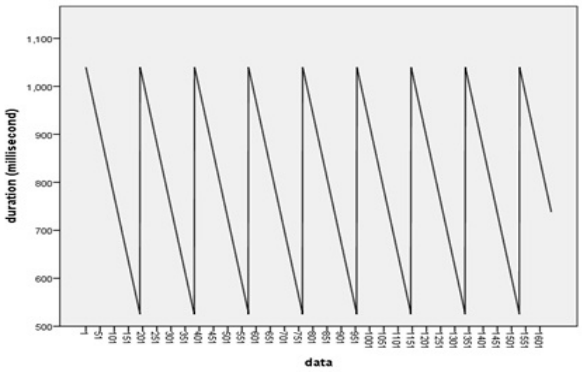

Figure-10. Ideal input-output delay waveform.

This high ripples and the decline slope, however, do not correlate to the error data in NMEA 0183. Both at

high and low duration of delay produced valid readings of the sensor. From the total data transmission, only 1.5% data was corrupted. The error occurrence was random which does not conclude with the ripple and slope pattern. Figure-11 shows the histogram of error in the data. The reason for such input to output delay pattern may due to the microcontroller executing the stack of instructions and loop back the cycle. The influence of the Atlas sensors and its combiner could as well be the reason of the pattern. Overall the average duration time between the input and output is 924.9 millisecond and mode of 1038 millisecond.

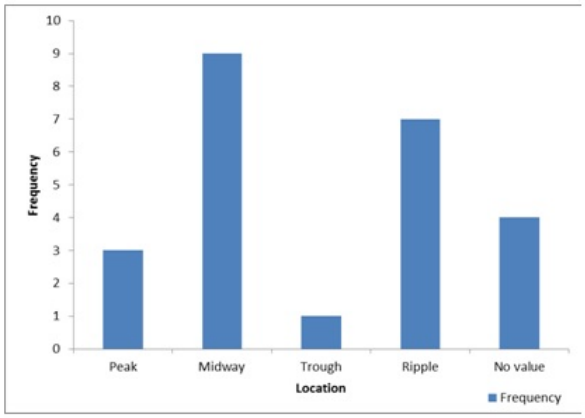

Figure-11. Error data histogram by location in waveform.

CONCLUSIONS AND FUTURE WORK

This technique provides an example of a solution to communicate non-asynchronous serial data with NMEA 0183. The algorithm proves the functionality of multiplexing non-formatted asynchronous serial data to NMEA 0183 compliant based on the program which was done in Arduino platform. This has solved the un-standardized data issue with ease for data logging. It is worth mentioning that the amount sensors involved does not limit the program capability with just a slight adjustment of the channel value and adding more channel control pin. Further work on the scope can be expanded by implementing in other protocol such as NMEA 2000 protocol for unformatted asynchronous serial data or into other formats.

ACKNOWLEDGEMENT

The authors give the acknowledgment to Ministry of Education Malaysia for providing funding to pursue this research.

REFERENCES

[1] H. Ferreira, C. Almeida, A. Martins, J. Almeida, N. Dias and A. Dias *et al.* 2009. "Autonomous bathymetry for risk assessment with ROAZ robotic surface vehicle," in OCEANS 2009 - EUROPE, pp. 1-6.

[2] M. Caccia, R. Bono, G. Bruzzone, G. Bruzzone, E. Spirandelli, G. Veruggio *et al.* 2005. "Sampling sea surfaces with SESAMO: an autonomous craft for the study of sea-air interactions," Robotics & Automation Magazine, IEEE, Vol. 12, pp. 95-105.

[3] M. Dunbabin, A. Grinham, and J. Udy. 2009. "An autonomous surface vehicle for water quality monitoring," in Australasian Conference on Robotics and Automation, pp. 1-6.

[4] G. Hitz, F. Pomerleau, M.-E. Garneau, C. Pradalier, T. Posch and J. Pernthaler *et al.* 2012. "Autonomous inland water monitoring: Design and application of a surface vessel," Robotics & Automation Magazine, IEEE, Vol. 19, pp. 62-72.

[5] O. Yaakob, Z. Mohamed, M. Hanafiah, D. Suprayogi, M. A. Ghani and F. Adnan *et al.* 2012. "Development of unmanned surface vehicle (USV) for sea patrol and environmental monitoring," in Proceedings of the 8th International Conference on Marine Technology, Kuala Terengganu.

[6] (2014). Arduino. Available: http://arduino.cc/

[7] "Desciption, Typebar Page Printer (Model 15)," Teletype Corporation, Chicago1931.

[8] E. S. Roberts. 1995. The Art and Science of C: Addison-Wesley.

[9] PaulStoffregen. 2014. AltSoftSerial Library. Available: http://www.pjrc.com/teensy/td_libs_AltSoftSerial.html