GPS LOGGER DEVELOPMENT USING AN ATMEGA 2560 MICROCONTROLLER

DESAIN GPS LOGGER MENGGUNAKAN MIKROKONTROLER ATMEGA 2560

Purwowibowo, Sensus Wijonarko, Tatik Maftukhah
Research Center for Physics – Indonesian Institute of Sciences
E-mail: purw009@lipi.go.id

ABSTRACT
Although the applications of GPS navigation have been widely used, the implementations of GPS logger currently are still scarce. GPS loggers are able to keep locations in form of coordinates and UTC, so that the data might be accessed later on using SD cards equipped in the devices. The purpose of this research was to create a GPS logger using an open source microcontroller system to maintain flexibility in development. A Mega ADK Arduino board and a GPS shield Ublox Lae-6H were used. Furthermore the device was provided with a Thin Film Transistor (TFT) display and a SD module. The received data were displayed on the TFT in coordinates of latitude, longitude and altitude. To get the accuracy level, measurement results of the GPS logger was compared with measurement results from other GPS products in several places. According to the comparison, the average accuracy of the GPS logger was less than 2 meters.

Keywords: GPS logger, microcontroller, accuracy

ABSTRAK
Meskipun aplikasi-aplikasi GPS navigasi sudah banyak digunakan, tetapi penerapan GPS logger saat ini masih jarang. GPS logger ini menyimpan lokasi dalam bentuk koordinat dan UTC, sehingga data dapat diakses kembali melalui SD card yang dipasang di alat tersebut. Tujuan penelitian ini adalah untuk membuat GPS logger dengan menggunakan sistem mikrokontroler yang open source guna menjaga fleksibilitas dalam pembuatan. Sistem ini menggunakan board Arduino Mega ADK dan GPS shield Ublox Lea-6H, yang dilengkapi dengan layar Thin Film Transistor (TFT) dan modul SD card. Data ditayangkan di layar TFT dalam bentuk koordinat lintang utara, bujur timur dan ketinggian. Untuk mengetahui tingkat akurasi, hasil-hasil pengukuran GPS logger tersebut dibandingkan dengan hasil-hasil pengukuran produk GPS lain di beberapa tempat. Berdasarkan perbandingan tersebut diperoleh akurasi rerata dari GPS logger ini kurang dari 2 meter.

Kata kunci: GPS Logger, mikrokontroler, akurasi
1. INTRODUCTION

Many researches at present in sciences such as metrology, geophysics, and aeronautic are using sensors and data loggers. To get accurate results, the data storage is one of the most important processes in the data logger applications (Sidik, 2015). (Li Shen, 2014). (Peter, 2015)

Applications of Global Positioning system (GPS) have been widely used especially for navigation in vehicles by the Indonesian communities, but the GPS that is utilized as a logger is rarely used. The purpose of data logging is to store data that can be later accessed. In this case, the GPS logger will store the location in the form of coordinate and time on a SD card, so that it can be accessed again when needed.

In this project, the GPS logger was made using an Arduino board and a GPS shield, and then is equipped with a TFT screen and an SD card module to display and store data. This GPS was run with an open source software for the sake of easiness to adapt to existing functions.

In this paper, the design and principle of work of the GPS logger prototype that uses a microcontroller system will be explained. Besides, the process of data collection and comparison accuracy with other GPS on the market are also discussed.

2. LITERATURE REVIEW

This section will explain the GPS system, microcontroller that uses an Arduino device, and device that displays and stores data.

2.1. GPS

GPS is a system dedicated to determine the location on the earth surface with the assistance of satellite signal alignments continuously throughout the world regardless of weather conditions. This system has four basic factors, namely latitude, longitude, altitude, and time. This system uses a number of satellites that move around the earth. These satellites send microwave signals to the Earth. From these signals, the GPS reads information provided by the satellite and determines the location, speed, direction, and time on the GPS itself. (Mohammed, 2016). (Sabah, 2018).

GPS information can also be obtained by the Pythagorean formula (Dale DePriest, 2016) as:

\[ P_{rs} + T + E_s = \sqrt{(X - X_s)^2 + (Y - Y_s)^2 + (Z - Z_s)^2} \]

(1)

\[ T = \text{Time error receiver} \]
\[ X, Y, Z = \text{Unknown position} \]
\[ X_s, Y_s, Z_s = \text{Position of satellites known by other satellites} \]
\[ E_s = \text{Total number of modeling errors from GPS} \]
\[ P_{rs} = \text{Received approximate distance between the receiver and satellite} \]
\[ T = \text{Receiver time error} \]

In this study, the GPS Shield Ublox LEA-6H for Arduino Uno (Figure 1) is utilized as a GPS device, that then is connected to the Arduino Mega ADK. The GPS device is a GPS signal receiver has accuracy, anti-jamming technology, power safe mode, backup power, and is able to receive from weak signals (DFRobot, 2017).

The connection between GPS shield and Mega ADK uses an I2C system by connecting the SDA (pin 20) and SCL (pin 21) pins on Mega ADK to the SDA and SCL pins on the GPS respectively.

2.2. Microcontroller

Microcontroller, such as Mega ADK, is a computer system that works in a chip. The microcontroller contains a processor core, memory and input/output port that can be programmed using software. In this research, Mega ADK (Figure 2) is used as a microcontroller device. The Mega ADK is a microcontroller board based on ATmega2560. It has 54 digital input/output pins, a 16 MHz Quartz crystal, USB port, ICSP header and reset button. The board can be turned on by connecting this processor to a computer via a USB port, to a battery, or to AC supply via a DC adapter. The Mega ADK has a much greater capacity than the Uno. Hence, it can contain larger programs. The Mega ADK board is designed able to work using the Uno's shield. The board can be programmed with Software downloaded for free at https://www.arduino.cc/en/Main/Software, and then choose the appropriate board in the tools that available in the software (Arduino, 2017).

Figure 1. GPS Shield Ublox LEA-6H
2.3. Data Logging
Logging is a process of collecting data and storing it in memory for a certain period of time. In this study, SD cards are used as memory to store data sent from GPS via Mega ADK, so that data can be accessed via an SD card connected to a computer.

In this experiment, the SD card breakout (Figure 3) is used as a tool to store data in the SD card. The SD card module uses a serial peripheral interface (SPI) connection to communicate with Mega ADK, through CS pins (53), MOSI (51), MISO (54), SCK (52) found in the board.

2.4. Display
The device used to display results in this project is a Thin Film Transistor /TFT (Figure 4) equipped with a shield for Uno board. In this experiment, the shield is directly connected to the Mega ADK without the need for an additional wiring. The TFT is used as a means to display results because it has a high resolution. Therefore, it is able to display data with smaller sizes than ordinary LCDs.

3. GPS CIRCUIT DESIGN AND CONSTRUCTION
3.1. Materials
The Mega ADK microcontroller is used as a processor that receives position data from satellites via the GPS shield. The position data is displayed on the screen to find out the position directly. The data are also stored in the SD Card memory.

The block diagram of the GPS Logger prototype is presented on Figure 5.
The block diagram in Figure 5 shows the systematic work, so the devices can be arranged in such a way in the Figure 6 and Figure 7.

The flow chart to make the program is presented on Figure 8.

The software development is an open source based programming used to create programs that can be compiled/verified and then can be uploaded to the board which is...
connected through the port. Hence, it can work according to the functions contained in the program (Figure 9).

Figure 9. Arduino Software

The following program requires a library that must be downloaded, because it is separated from the Arduino software, namely SDfat library and UTFT library. To use the separate library, move the library folder to the Arduino libraries folder, so that the library can be called into the program. The UTFTmega library below is a regular UTFT library, where the define function for Arduino Mega has been uncommanded. Therefore it is compatible with Arduino Mega (Figure 10). The file is located in the UTFT, Hardware, AVR, HW_AVR_Defines folder. The library is renamed before being added to the library to make it easier to use and to prevent exchange with the regular UTFT library.

Figure 10. Program to define an Arduino board as a mega Arduino

3.2. Measurement Method

The measurement was carried out in Jakarta, and at Puspiptek Serpong. Measuring instruments used as the accuracy comparison are the 60CSx Garmin GPS, 78s Garmin GPS, and iPhone 5S mobile phones (Figure 11 and 12).

Figure 11. Garmin 60CSx and Garmin 78s
Data gathering was conducted with temperature between 26 °C and 40.1 °C and humidity between 44 % and 87%. This weather conditions did not give any influence to the comparison process of the GPS.

The equipment was turned on a few minutes as a warming up before the data collection. Then, was arranged as in the Figure 12 and placed in the same open location. Data were recorded manually from the left to right every hour for 24 times.

Data collection was carried out to test GPS accuracy in various places in same time using other GPS devices as a comparison.

4. RESULTS AND DISCUSSION

The results of the GPS data are arranged in a table in the form of coordinates and length units for altitude/elevation accompanied by humidity and temperature data for the collection of 72 data. Then, 10 data at Monas and Ancol respectively were taken every 2 minutes, while the data collection at Puspiptek Serpong area were obtained every minute.

Coordinate measurement results have the following units:

- \(\text{Lat} = \text{Latitude (degree}°\text{ minute'}\text{ second")}\)
- \(\text{Lon} = \text{Longitude (degree}°\text{ minute'}\text{ second")}\)
- \(\text{Alt} = \text{Altitude (meter)}\)
- \(\text{Elev} = \text{Elevation (meter)}\)
- \(S = \text{South}\)
- \(E = \text{East}\)
- \(N = \text{North}\)
- \(W = \text{West}\)

Minutes and seconds in this paper are not in a time scale, but they act as a coordinate in the GPS system, namely 1 degree = 60 minutes, 1 minute = 60 second.

From the measurement data, it is seen that the 60csx and Iphone, Garmin display data with degrees, seconds, and seconds while the GPS Ublok Lea-6H and Garmin 78s display only degrees and minutes in decimal form. So to compare the accuracy of the data, they must first be converted to meters and then are subtracted.

It is already known that the circumference of the earth is 40075,017 km (equatorial) and 40007.86 km
(meridional) Bharat (2015). Equatorial earth circumference means the distance of the earth's circumference is measured from the equator and meridional is measured from pole to pole. For calculation of latitude, meridional and longitude are used using equatorial. The following is the calculation of the conversion of coordinates to meters:

Latitude:

$360^\circ = 40007.86 \text{ km}$

$1^\circ = 40007.86/360 = 111.13294 \text{ km}$

$1\text{min} = 111.13294/60 = 1.85221567 \text{ km}$

$1\text{sec} = 1.85221567/60 = 0.03087026117 \text{ km}$

Longitude:

$360^\circ = 40075.17 \text{ km}$

$1^\circ = 40075.17 / 360 = 111.3199167 \text{ km}$

$1\text{min} = 111.3199167/60 = 1.855331945 \text{ km}$

$1\text{sec} = 1.855331945/60 = 0.03092219908 \text{ km}$

From the above calculation results, the coordinate data can be converted into units of length by changing the units of degrees, minutes and seconds into the length unit, and then added. An example of the calculation is presented on Table 1.

### Table 1. Measurement of latitude and longitude at President Soeharto’s Inscription located at Puspiptek Serpong Tangerang

| Time  | Ubik Lava-6H | Lat   | Alt  | Garmin 78s | Lat   | Elev | Garmin 60 CSx | Lat   | Elev | Iphone 5s | Lat   | Elev |
|-------|--------------|-------|------|------------|-------|------|--------------|-------|------|-----------|-------|------|
| 10:10 | 6°21.20987' S| 106°40.036'E | 75.7  | 6°21.209' S | 106°40.035'E | 74.7 | 6°21.12.7' S | 106°40.211'E | 71    | 6°21.12.5 | 106°40.2'E |
| 10:11 | 6°21.20947' S| 106°40.033'E | 77.7  | 6°21.209' S | 106°40.033'E | 75.7 | 6°21.12.6' S | 106°40.211'E | 69    | 6°21.12.5 | 106°40.2'E |
| 10:12 | 6°21.20831' S| 106°40.029'E | 76.4  | 6°21.209' S | 106°40.028'E | 73.7 | 6°21.12.6' S | 106°40.211'E | 69    | 6°21.12.5 | 106°40.2'E |
| 10:13 | 6°21.20868' S| 106°40.027'E | 75.1  | 6°21.209' S | 106°40.027'E | 72.7 | 6°21.12.5' S | 106°40.211'E | 69.5  | 6°21.12.5 | 106°40.2'E |
| 10:14 | 6°21.20785' S| 106°40.026'E | 74.8  | 6°21.209' S | 106°40.026'E | 71.7 | 6°21.12.5' S | 106°40.211'E | 69.5  | 6°21.12.5 | 106°40.2'E |
| 10:15 | 6°21.20734' S| 106°40.025'E | 74.5  | 6°21.209' S | 106°40.025'E | 71.7 | 6°21.12.5' S | 106°40.211'E | 69.5  | 6°21.12.5 | 106°40.2'E |
| 10:16 | 6°21.20688' S| 106°40.023'E | 73.1  | 6°21.209' S | 106°40.023'E | 71    | 6°21.12.5' S | 106°40.211'E | 69.5  | 6°21.12.5 | 106°40.2'E |
| 10:17 | 6°21.20637' S| 106°40.022'E | 72.8  | 6°21.209' S | 106°40.022'E | 70.7 | 6°21.12.5' S | 106°40.211'E | 69.5  | 6°21.12.5 | 106°40.2'E |
| 10:18 | 6°21.20587' S| 106°40.021'E | 72.5  | 6°21.209' S | 106°40.021'E | 70.7 | 6°21.12.5' S | 106°40.211'E | 69.5  | 6°21.12.5 | 106°40.2'E |
| 10:19 | 6°21.20537' S| 106°40.020'E | 72.2  | 6°21.209' S | 106°40.020'E | 70.7 | 6°21.12.5' S | 106°40.211'E | 69.5  | 6°21.12.5 | 106°40.2'E |

Latitude of GPS Arduino was $6°21.20947'$ S. This is equal to

$(6 \times 111.13294) + (21.20947 \times 1.85221567) + 0 = 706.0821527 \text{ km}$

Latitude of GPS Garmin 60 CSx was $6°21'12.6''$ S. This is same with

$(6 \times 111.13294) + (21 \times 1.85221567) + (12.6 \times 0.0308726117) = 706.083164 \text{ km}$

From the results of converting latitude and longitude into unit of length like the above example, graphs can be made to see the accuracy and stability of these tools when compared with the others. Figure 13 is an example of a GPS accuracy comparison chart.
From the entire comparison chart results, it can be seen drastic changes from the data obtained by Iphone. This is caused by their resolution difference. The resolution of Iphone is 1 second for approximately 0.03 km (30 meters), while the Garmin 60csx is 0.1 second (about 3 meters), Garmin 78s is 0.001 minute (around 1.85 meters), and the GPS logger can display resolutions up to 0.00001 minute (around 0.0185 meters).

To measure the highest GPS fluctuation, standard deviation calculations are performed for each measurement table. This is to ensure that the tool can provide constant information if it will be calibrated to a certain standard. The biggest standard deviation from all obtained data for the latitude was 3,60582 meters and longitude was 3,736344.

The data was attained from 72 measurement samples. The data uncertainty was obtained by calculating repeatability and readability, and ignoring other sources of uncertainty (Purwowibowo, 2015). The repeatability and readability are formulated by ISO.

\[ \sigma = \frac{\sum_{i=1}^{n}(y_i - y)^2}{n} \]

\[ \sigma_y = \frac{\sigma}{\sqrt{n}} \]  \[\text{[2]}\]

From the calculation of the standard deviation of 72 sample data, the values of latitude and longitude were 1.434037719 m and 1.535183245 m respectively. The results of the repeatability of latitude and longitude GPS logger were 0.169002966 meters and 0.18092308 meters respectively.

The resolution standard uncertainty (readability) of the GPS logger is processed using the formula:

\[ U_{res} = \frac{R}{2} \]

\[ u_{res} = \frac{U_{res}}{\sqrt{3}} = \frac{R}{\sqrt{12}} \]  \[\text{[3]}\]

R is the resolution in the tool (0.00001 minute), but the resolution that was used only 0.001 minute, because based on the data, 0.00001 minute has too many fluctuations. Readability is 0.534688608 meters for
latitude = 0.001 minute (1.85221567 meters) and latitude = 1.855331945 meters.

The error is attained by subtracting the data between GPS logger and reference. These discrepancies are then averaged (Table 2). It is assumed that the reference tool has correct coordinates.

Error = GPS logger – reference

|       | 78s     | 60csx   | iPhone  |
|-------|---------|---------|---------|
| Lat   | -0.000291467 | 0.001603624 | 0.00053174 |
| lon   | -0.000051537  | 0.001975585  | 0.007129285 |

Table 2. The average errors for latitude and longitude (km)

5. CONCLUSION

Based on the measurements data, the developed GPS logger was able to work in various locations and had the highest standard deviation at 3.60582 meters for latitude and 3.736344 meters for longitude. The resolution value should be displayed up to 0.001 minutes. This GPS had an average error less than 2 meters when was compared to Garmin GPS on the market.

ACKNOWLEDGMENT

The authors would like to thank to Jason Pratama - the student from Swiss German University with ID card 11501004 - for assisting in this research, and to the management of Puspiptek for providing a place and facilities, so that this research activity could be well done.

REFERENCES

Arduino (2017) The open-source Arduino Software (IDE), (https://www.arduino.cc/en/Main/Software, last accessed 4 January 2017)

Bharat Raj Singh and Onkar Singh (2015). Dire Consequences on Little Shifting of the Earth’s Spinning Angle – An Investigation Whether Polar Ice Shrinkage may be the Cause? http://dx.doi.org/10.5772/58708

Dale DePriest (2016). How Your GPS Works (www.gpsinformation.org/dale/theory.htm, last access 27 October 2016)

DFRobot (2017), Arduino GPS Shield ublox LEA-6H (www.dfrobot.com , last access 4 January 2017)

Li Shen & Peter R. Stopher (2014) Review of GPS Travel Survey and GPS Data-Processing Methods, Transport Reviews, 34:3, 316-334, DOI: 10.1080/01441647.2014.903530

Mohammed, Jareer & Moore, Terry & Hill, Chris & Bingley, R.M. & Hansen, D.N. (2016). An assessment of static precise point positioning using GPS only, GLONASS only, and GPS plus GLONASS. Measurement. 88. 10.1016/j.measurement.2016.03.048.

Peter R. Stophera, Li Shenb, Wen Liuc, Asif Ahmed. (2015). The Challenge of Obtaining Ground Truth for GPS Processing. Transportation Research Procedia 11. 206 – 217
Purwowibowo, Jalu Ahmad Prakosa. (2015) Prototipe Frekeunsi Meter Rentang Ukur (10~2000) Hz Terkalibrasi ke Standar Primer. INKOM Jurnal Informatika, Sistem Kendali dan Komputer Vol 9, No.1. http://dx.doi.org/10.14203/j.inkom.412.

Sabah S. Al-Fedaghi, Yousef Atiyah. (2018). Tracking Systems as Thinging Machine: A Case Study of a Service Company, (IJACSA) International Journal of Advanced Computer Science and nd Applications, Vol. 9, No. 10.

Sidik, M.A.B., M.Q.A. Rusli, Z. Adzis, Z. Buntat, Y.Z. Arief, H. Shahroom, Z. Nawawi, M.I. Jambak. (2015). Arduino-Uno Based Mobile Data Logger with GPS feature. Telkomnika Telecommunication Computing Electronics And Control, 13 (1). pp. 250-259. ISSN 1693-6930