



IMPLEMENTATION OF INFORMATION DISPLAY DEVICE FOR ESTIMATION OF BUS ARRIVAL TIME

Fauzi Maulana, Misbahuddin, Riri Fitri Sari, Ruki Harwahu, Anak Agung Putri Ratna, and Ellen S.W. Tangkudung

Department of Electrical Engineering Universitas Indonesia, Kampus Baru UI, Depok, Indonesia

E-Mail: fauzi.maulana@ui.ac.id

ABSTRACT

The purpose of the research is to implement a device that functions as a real-time display of the estimated arrival time of the yellow bus in University of Indonesia. The device uses an 8-bit Atmel AVR microcontroller based on Arduino platform as its main controller, and utilizes GPRS-based connection to transmit the data to the server. The system fetches the data of the calculated estimation arrival time of the Yellow Bus for each bus stop. The information will be updated periodically by the server. The testing results show that the success rate of the data-fetching was 99.6% for the duration of 23:53 until 02:46, and 99.7% for the duration of 07:24 until 09:43. The average usage of GPRS quota was 10.3 MB for a period of 10 days. Finally, the device's additional function for maintenance and debugging by utilizing SMS service has been proved to work properly.

Keywords: microcontroller, GPRS, server, estimated bus arrival time, arduino, display device

1. INTRODUCTION

In the most urban area, traffic congestion is a serious problem due to the increase of traffic volume. This problem causes many impacts to socio-economy, and environment such as people got stress on the road, increasing of the traveling cost, and increasing of CO₂ emissions produced during congested periods. The traffic congestion is the main source of CO₂ emissions in the transport industry [1]. In addition, Jourmard *et al.* [2] report that vehicles traveling at a low speeds produce CO₂ emissions higher than vehicles traveling at high speed in urban area. One of the ways to reduce the traffic congestion is the increase of the convenience of usage of the public transportation, so that the travelers will transfer to the public transportation. Therefore, University of Indonesia is as one of the biggest universities in Indonesia that highly cares against the environmental sustainability by applying a policy, i.e. the use public transportation to reduce the air pollution. Available public transportation in the University of Indonesia called the Yellow Bus operate from 07:00 am until 22:00 pm.

One of the public transportation facilities that should be available is the providing of the estimation of arrival time both an online accessed system and an information display deploying at each bus stop. Some researchers have developed the both of systems. Swati *et al.* [3] designed a system to display the real-time location of the city buses that are deployed at each bus stop, can be accessed by web based and mobile application. Their system just able to display the real-time location of the buses, but it cannot display the real-time arrival time of the buses. Therefore, the passengers have to estimate by themselves the bus arrival time. Misbahuddin *et al.* [4] have proposed a model to predict the bus arrival time at each bus stop, but the system has not had an information display deploying at each bus stop.

To complete the drawback system of the public transportation of University of Indonesia, this paper proposes to implement an information display device of

estimated arrival time of the Yellow Bus deployed at each bus stop in University of Indonesia.

2. LITERATURE REVIEW

The information display device consists of two basic components, i.e. Atmel AVR microcontroller and General Packet Radio Service (GPRS). To support the both components so that they can well function, they need the instructions constructed by the AT commands. All components will be discussed in this section.

a) Arduino as electronic prototyping platform

Arduino is a platform that is used in designing and developing microcontroller-based electronic device's prototype. Arduino was initially designed at Ivrea Interaction Design Institute in Italia [5] as a tool to ease up the design process of a prototype, of which the target is students who lack of educational background in electronics and programming.

There are several benefits of using Arduino over other microcontroller platforms: inexpensive cost, cross-platform, easy to use, open-source and extensible software, and open-source and extensible hardware. [6]

i. ATMEL AVR microcontroller

Atmel AVR microcontroller is a Reduced Instruction-Set Computers (RISC) microcontroller family of 8-bit, 16-bit, and 32-bit developed by Atmel Microchip [7]. It is the first microcontroller family using the storage for the executable binary in the internal flash memory. it differs from other microcontrollers using the One-Time Programmable Read-Only Memory (OTP-ROM), the Electrically Programmable Read-Only Memory (EPROM), or the Electrically Erasable Programmable Read-Only Memory (EEPROM) as the storage.

Alf-Egil Bogen and Vegard Wollan are the first developer that designs the AVR architecture. The development was highly related with the compiler for the Ingenjörfirman Anders Rundgre (IAR) system a Swedish



company engaged in the field of software engineering for embedded system [8], so that the resulting instruction set can be compiled more efficient by high-level programming language compiler.

There are several features of the AVR microcontroller: Bi-directional I/O port with internal configurable pull-up resistor; Calibrated internal oscillator with configurable frequency; Internal flash memory which could be programmed using the SPM (Store Program Memory) and LPM (Load Program Memory) instructions.; Bootloader section in the internal flash memory to allow programming without the needs of external programmer chip; On-chip debugging which supports Joint Test Action Group (JTAG) and debug WIRE protocol; Internal SRAM; Timers with 8-bit and 16-bit resolution; Serial interface: Two-Wire Interface (TWI), Universal Synchronous/Asynchronous Receiver Transmitter (USART), Serial Peripheral Interface (SPI), and Universal Serial Interface (USI).

ii. Arduino hardware

The Arduino board has various Atmel AVR microcontroller types in 8-bit, 16-bit, or 32-bit. It is also completed components that enable programming and integration into other circuits. It has standard connectors that are an important aspect to connect the CPU board against the interchangeable add-on modules called as shields. Some shields can be connected directly with the Arduino board on various pins, yet most shields can be individually addressable through an I2C serial bus. Thus, many shields allow to be stacked and used in parallel.

Some mega AVR series of chips such as the ATmega8, ATmega168, ATmega328, ATmega1280, and ATmega2560 have been used by the official Arduinos. Moreover, the developers allow to produce the clone of Arduino, yet they should follow the license and terms of the Creative Common Attribution-Share Alike 3.0 [9], and publish their design under the same licence.

iii. Arduino software

The Integrated Development Environment (IDE) of Arduino based on a cross-platform application is written in Java that is developed from IDE of the processing programming language and the writing projects [10]. The design of the IDE is easy to be used by artists and other newbie programmers. It is come by a code editor with features like the enterprise IDE such as brace matching, automatic indentation, and syntax highlighting. Furthermore, it can compile and upload the programs to the board with a once click. "Sketch" is the familiar term for the source code or program in Arduino.

Arduino program is very easy to be developed because it is written in standard C or C++, and is come with a program library known as "Wiring" derived from the original wiring project. This enriches the operations to develop many common input/output systems.

b) General packet radio service (GPRS)

One of the GSM based packet-switched technologies is the GPRS that enables to connect to other

GSM devices based on Internet Protocol (IP). It supports wide range of commercial and enterprise application. The data that will be transferred is divided in packets and is that transmitted via the radio and core network.

GPRS technology is the first milestone to be intermediate between the second-generation GSM technology and the third-generation WCDMA/UMTS technology. The GPRS technology can serve the data transmission with data rates up to a maximum of 175 Kbps, so that it allows to facility web browsing, yet its data rate was too slow for real data application.

GPRS technology has some benefits for users and network operators similar on the standard GSM system. The offered significant benefits are speed, packet-switched operations, always-on connectivity, more applications, and CAPEX and OPEX. [11]

c) AT commands

Instructions that are usually used to control a mode are AT commands. Every command line is always begun with "AT" or "at", where AT is the abbreviation of "Attention". The prefix AT is a mark for modem to start a command line but it is not an instruction in the AT commands. For example, an instruction AT+ CMGS is an actual AT command for modem.

AT commands for a GSM/GPRS modem or mobile phone that can used are: [11] Obtain basic information of the GSM/GPRS model or mobile phone; Obtain basic information of the subscriber; Obtain the current status of the GSM/GPRS model or mobile phone; Create a data connection or voice connection for the remote modem; Send and receive fax; and Send, read, write, or delete SMS messages.

There are two types of AT commands [11]: **AT commands** are commands for dial (D), answer (A), hook control (H), return to online data state (O). These commands do not start with prefix "+"; and **Extended AT commands** are commands for send SMS message (+CMGS), list SMS messages (+CMGL), and read SMS messages (+CMGR). These commands start with prefix "+" that is used for the GSM modem.

3. SYSTEM DESIGN

a) System requirement

This section discusses the requirements of the electronic display device which displays the estimation time of arrival of the yellow bus. This section will generate some requirements from the basic concepts to state the application of the system. Besides the basic concepts, there are discussions with lecturers in computer engineering at University of Indonesia.

The requirements of the device are:

- The system is able to automatically connect via wireless with the server that does the statistic calculation.
- The system is able to fetch the calculation result of the estimation time of arrival of the yellow bus from the server.
- The system is able to fetch the data periodically.



- The system is able to show the information during the operational time of the yellow bus.
- The system is able to track the local time for maintenance and debugging purposes.
- The system is able to regulate power usage and distribute it properly to all components.
- The system has vivid and unambiguous display.
- The system is able to show the information of the red route and blue route bus.
- The system has appropriate and suitable dimension to be properly placed at the bus stop.
- The system is securely mounted to the bus stop.
- System administrator is able to remotely do the maintenance of the system.
- System administrator is able to debug the system efficiently.

After some considerations regarding the basic functionality, prototyping time, prototyping cost, and implementation cost, the required hardware components include DFRduino Uno R3, IComSat v1.1, 4-inch common anode 7-segment, mainboard, power supply, and case. The software components consist of device's firmware, Arduino IDE 1.6.5, Altium Designer 2009, and Autodesk AutoCAD 2007.

b) Hardware components

This section discusses the hardware specification used to implement the system.

i. DFRDUINO UNO R3

DFRduino Uno R3 is a clone of the Arduino Uno R3 produced by DFRobot [12]. DFRduino Uno R3 is fully compatible with Arduino Uno R3 and the Arduino IDE. This board uses an ATmega16U2 programmed as a USB-to-Serial converter, and still uses a DIP target microcontroller so that it can be detached for bootloader reprogramming.

The specifications of the DFRduino Uno R3 are: [10] Microcontroller: ATmega328 (DIP package); Operating voltage is 5V; Input voltage recommended is 7 ~ 12V; Limit of input voltage is 6 ~ 20V; Digital I/O pins is 14 (of which 6 provide PWM output); DC current per I/O pin is 40 mA; Flash memory is 32 KB of which 2 KB used by bootloader; SRAM is 2 KB; Clock speed is 16 MHz; Size: 75 x 54 x 15 mm; and RoHS compliance



Figure-1. DFRduino UnoR3 [10].

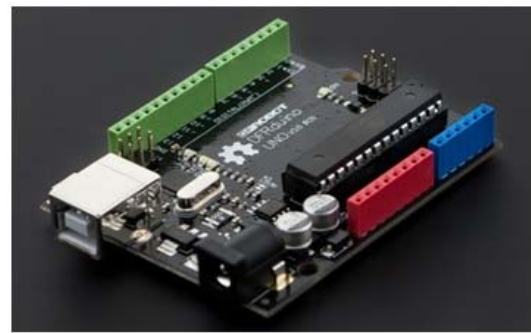


Figure-2. IComSat v1.1 [12].

ii. ICOMSAT V1.1

IComSat v1.1 is a GSM/GPRS module based on quad-band SIM900 which is produced by Itead Studio. The module is fully compatible with the Arduino platform, and is controlled using AT commands (GSM 07.05, 07.07, and SIMCOM enhanced AT commands).

The features of the IComSat v1.1 are: [12] Quad-band is 850/900/1800/1900 MHz; GPRS multi-slot class; GPRS mobile station class B; SMS; Header for all pins of the SIM900 module; Real-Time Clock support with super capacitor battery.

The specifications of the IComSat v1.1 are: [12] PCB dimension is 77.2 x 66 x 1.6 mm; Power indicator, status LED; network status LED; Power is 9 ~ 20V; UART interface; and RoHS compliance.

iii. 4-INCH COMMON ANODE 7-SEGMENT

The display output used in the device is 4 pieces of red, 4-inch common anode 7-segment. 2 pieces are used to display the information of the red route; the other two are used for the blue route. The choice is made with consideration to requirement no. 7 and 8.

iv. Mainboard design

The function of mainboard is to connect all the components: DFRduino Uno R3 + IComSat v1.1, 7-segment driver circuit, DC barrel jack for power supply, voltage regulator circuit, and the 7-segments itself.

Components in the mainboard consist, but not including the DFRduino Uno R3 + IComSat v1.1, are:

- 7-segments driver which consists of four TIP42C PNP-type silicon epitaxial transistors, two LN2804 8-channel darlington sink, drivers, and four axial resistors.
- Voltage regulator circuit consists of one simple-switcher step-down voltage regulator, two stabilizing capacitors, one inductor, one 1N5822 schottky barrier rectifier diode, and 3 voltage-dividing axial resistors
- Some connectors consist of two 8-pin male headers and two 6-pin male headers for the DFRduino + IComSat, and 4 8-pin male headers for the four 7-segments

Figure-3 shows the schematic design of the mainboard. Blue block is the 7-segment driver circuit, consists of 2 8-channel darlington sing drivers, 4 axial resistors, and 4 silicon epitaxial transistors. Output pins of



the left darlington driver are all connected in parallel to all segments A-G of the 4 7-segments. There are only 4 output pins of the right darlington driver used, each of which is connected to the base of each epitaxial transistor.

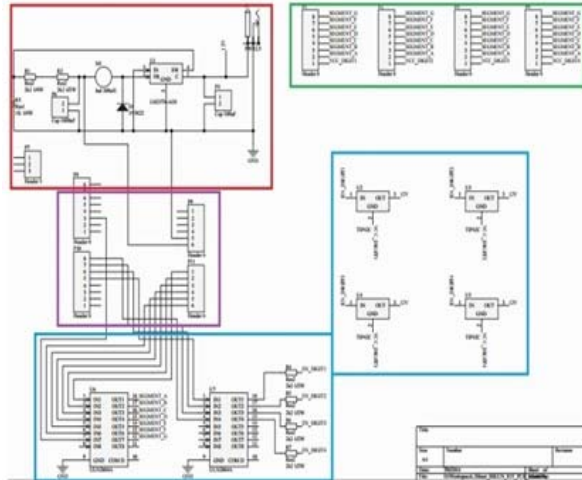


Figure-3. Schematic design.

Schematic of the mainboard is designed using Altium Designer application by picking and placing all the required components from the application's libraries. After the schematic design is done, a new file with *PcbDoc* extension is then made manually, which is used to design the PCB based on the schematic made earlier.

v. Power supply

The device uses a 9V 1A DC adapter. The chosen adapter is based on the calculation of the peak current consumption of the four 7-segments, current consumed by the DFRduino, and the IComSat to fetch the data using GPRS. The consumed current total can be calculated by the following equation.

$$I_{\text{tot}} = (I_p \times ns) + I_o \quad (1)$$

where I_p is peak current per segment, ns is the number of used segment, and I_o is current consumption by the GPRS and DFRduino.

The prototyping phase found out that the peak current consumption by the DFRduino and the IComSat to fetch the data is 500 mA, thus:

$$I_{\text{tot}} = (70 \text{ mA} \times 7) + 500 \text{ mA} = 990 \text{ mA}$$

vi. Case design

Figure-4 shows the front and back side design of the case, respectively, with consideration to the whole dimension of the device, production cost, assembly time, and installation time.

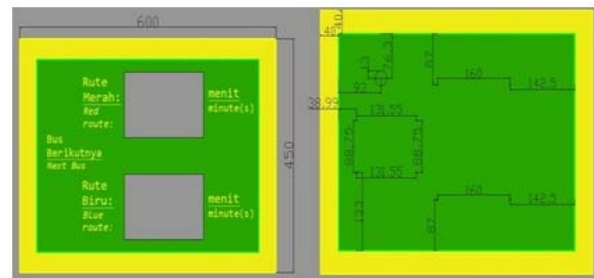


Figure-4. Front side and back side of case design.

The case has green background color with yellow border. The font color is yellow, representing the alma mater color of the University of Indonesia, while the green background represents the Go Green movement at the University of Indonesia. The case uses acrylic for front and back side, and aluminum for left, right, upper, and bottom side. The design is made using Autodesk AutoCAD 2007.

c) Software components

This section discusses the software components of the device. Referring to the requirement no. 8, the conventional number of I/O pins used to control 4 7-segments is 32 pins: 4 pins are used to control the supply voltage for each digit, and 28 pins are used to control segment A to G of each digit. This number of pins is not available, as the DFRduino Uno R3 has only 20 pins available (14 digital I/O + 6 analog inputs).

To meet the needs, the scanning method using the round-robin technique is used to display each digit one at a time with a very fast rate, so that human visual will interpret that the four 7-segments are on at the same time. With the scanning method, the number of I/O pins needed to control the four 7-segments is 11 pins: 4 pins are used to control the supply voltage of one 7-segment at a time and 7 pins are used to control the segment A to G of the four 7-segments in parallel. Figure-5 shows the flowchart of the scanning method.

To be able to fetch the data from the server, the device should first create a GPRS connection on the network provided by the GSM operator, initialize the HTTP service access, input the server's address, and execute the HTTP-GET method.

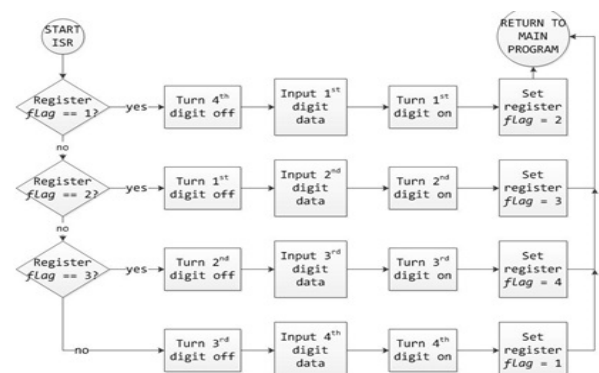


Figure-5. Scanning method flowchart.



The sequence consists of Set the GPRS connection by entering APN name and connection type to be made, activate the GPRS connection, Initialize the HTTP service access, set the GPRS bearer to be used to access the HTTP service, Input the server's address, execute the HTTP-GET function, and read the response from the server.

The Sequence no. 1-3 above will be executed once since those are initialization, while sequence no. 4-7 will be executed once every 4 seconds, with consideration on the traffic access to the server and the frequent change of the yellow bus position. Figure-6 shows the flowchart of the main program of the device.

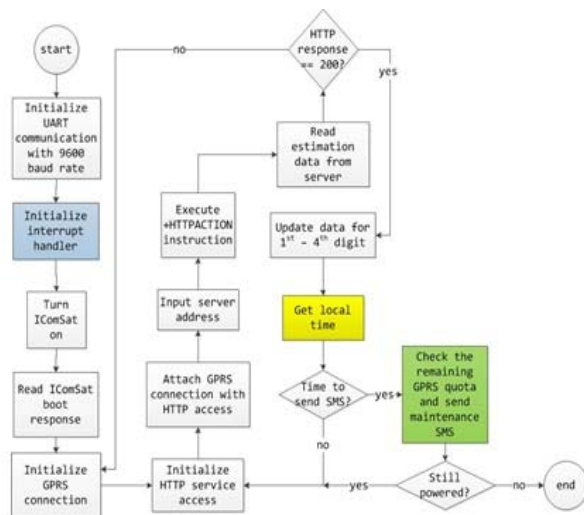


Figure-6. System flowchart.

4. IMPLEMENTATION

This section discusses the implementation result of the device, the physical design, mainboard implementation, interconnection of the components, and the difficulty faced in the implementation.

a) Physical design

Figure-7 shows the mainboard implementation which is used as a central interconnection for DFRduino + IComSat, power supply, and 4 7-segments. Components used on the mainboard are:

- 7-segments driver circuit consist of Red – 4 PNP-type epitaxial silicon transistors, Blue – 2 8-channel darlington sink drivers, and Yellow – 4 axial resistors
- Power supply circuit consist of Green – 1 simple-switcher step-down voltage regulator, Orange – 2 stabilizing capacitors, Purple – 1 inductor, Pink – 1 schottky barrier rectifier diode, and Cyan – 3 axial resistors.
- Connectors consists of Black – 2 8-pin male, headers and 2 6-pin male headers, and White – 4 8-pin male headers.

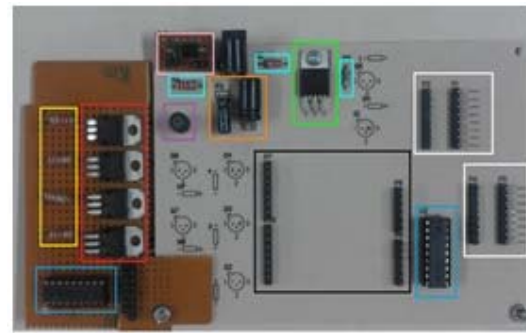


Figure-7. Mainboard implementation.

All components are well connected to the mainboard as showed in Figure-9. Interconnections between connectors on the mainboard and the pins on the 7-segments use AWG 18 jumper cables. Interconnections between the 7-segments and the back side of the case use a combination of 2 cm spacers and 1 cm spacer. The voltage regulator circuit is able to distribute power to all components. The firmware written is capable of controlling all hardware components, fetching the data from the server periodically, and displaying it on the four 7-segments using the scanning method.

Figure-10 shows the implementation of the device at the bus stop of the Faculty of Engineering, considering the path of the pedestrian, user's visibility, and protection from rainfall and direct sun light.



Figure-8. Components interconnection.



Figure-9. Implementation at the bus stop.



b) Solution for 7-segment control

The principle of the scanning method used for displaying the data is to turn on one digit at a time with a very fast rate. Figure-11 shows the algorithm of the implemented scanning method.

When the algorithm is executed, the Interrupt service Routine (ISR) in Figure-9 will be executed once every $1/390,000 \text{ Hz} = 2.56 \mu\text{s}$, so that the scanning method should not be interrupted too long because it would cause a blinking effect on the 7-segments, but the software-controlled UART communication between the DFRduino and the IComSat contains a cli instruction (shown in Figure-13), which means the communication between the DFRduino and the IComSat will still interrupt the scanning method.

The solution is to increase the baud rate of the software-controlled UART from the default value of 1200 to the maximum allowed baud rate of 9600.

```

start ISR
if(flag == 1) {
    turn 4th off
    input data for 1st
    digit
    turn 1st digit on
    flag = 2
} else if(flag == 2) {
    Turn 1st digit off
    Input data for 2nd
    data
    Turn 2nd digit on
    flag = 3
} else if(flag == 3) {
    Turn 2nd digit off
    Input data for 3rd
    digit
    Turn 3rd digit on
    flag = 4
} else {
    Turn 3rd digit off
    Input data for 4th
    digit
    Turn 4th digit on
    flag = 1
}

```

Figure-10. Scanning method algorithm.

```

start
SoftwareSerialWrite
if(_tx_delay == 0)
    SetWriteError;
    Return;
If(inv)
    b = ~b
clear Interrupt;
if(inv)
    *reg |= reg_mask;
else
    *reg &= inv_mask;
tunedDelay(delay);
for(i = 0; i > 0; --i)
    if(b & 1)
        *reg |= reg_mask;
    else
        *reg &= inv_mask;
    tunedDelay(delay);
    b >>= 1;
if(inv)
    *reg &= inv_mask;
else
    *reg |= reg_mask;
SREG = oldSREG;
tunedDelay(_tx_delay);
return;

```

Figure-11. The software serial write algorithm.

5. TESTING

The performance of the system is evaluated based on the success rate of fetching the data from the server, validity test of the estimation time of arrival of the bus, and the GPRS quota usage measurement.

FETCHING SUCCESS RATE

One of the determiner factors of estimated time accuracy level which will display on the device is the time stability of data fetching from the server to the device. In our work, there are 1000 fetching divided into 10 times of measurement. Figure-13 shows the maximum, minimum, and average of the fetching time in 100 fetching. The average fetching time shows a relative constant graph, but there is fluctuation in maximum time in seventh fetching time. This shows an acceptable stability time of data fetching.

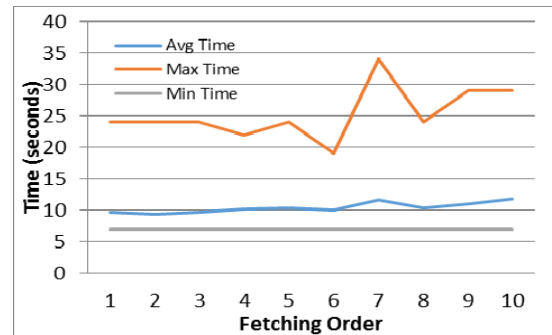


Figure-12. Fetching time.

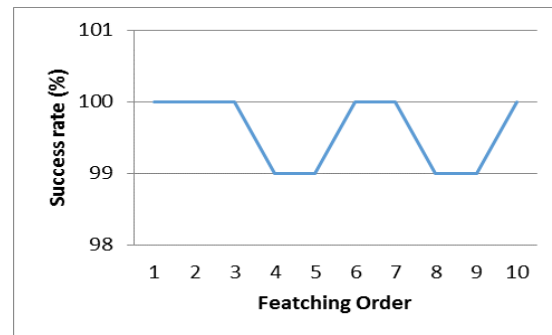


Figure-13. Success rate.

Figure-14 shows the consistency of the success rate of the 1000 fetching, with the average success rate of 99.6% and average fetching time of 10.40 seconds.

Figure-15 and 16 shows the test result graph of the second 1000 fetching from 07:24 to 09:43. The average success rate is 99.7% with average fetching time of 8.31 seconds.

a) Validity test

The test is done to find error toward the estimation time of arrival caused by the transmission delay of the system. The method used is to find the ratio of the displayed time when the bus arrives at a bus stop and the real traveling time of the bus to get to that bus stop.



$$\text{Error} = \frac{t_d - t_r}{t_r} \times 100\% \quad (2)$$

where t_d is the displayed remaining time and t_r is the real travel time.

Figure-15 shows graph plot based on 50 test results. It shows large error value and large error fluctuation, this is mostly because of 3 factors: (1) the difference unit used in the two measurements, the remaining time displayed is measured in minute, while the real traveling time is measured in second, so a conversion is done for each real traveling time from second to minute, (2) transmission delays which possibly occurred from the bus to the server and from the server to the display device, and (3) the big range of real traveling time which can be caused by the behaviour of each bus driver in driving the bus.

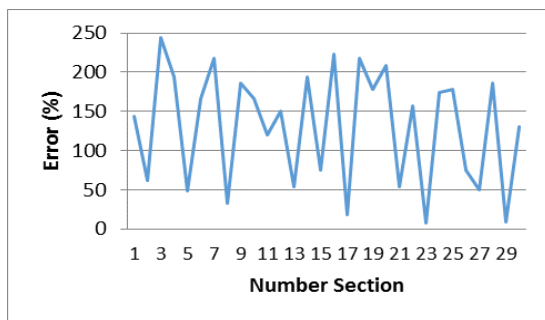


Figure-14. The error in each section.

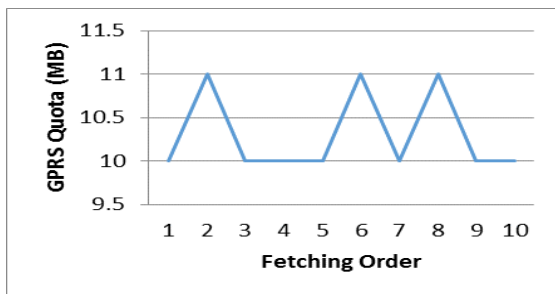


Figure-15. GPRS Quota usage test graph.

b) GPRS quota usage measurement

The purpose of this measurement is to get the estimation of GPRS quota usage for a period of 10 days. The measurement is done by running the device for a duration covering the operational time of the yellow bus, which is from 06:00 until 21:00 a day for 10 days.

Figure-16 shows that the usage of the GPRS quota is stable, with the value of 10 to 11 MB, and with average value of 10.3 MB. The result of the GPRS quota usage measurement is obtained each day using USSD code to get the difference between the starting quota and the remaining quota, after the device has been run from 06:00 until 21:00.

6. CONCLUSIONS

Arduino is a sufficient platform to ease the development of microcontroller-based electronic devices with libraries, IDE, and hardware platform supports. Coupled with IComSat v1.1, Arduino is effective for small-scope HTTP applications using GPRS, and is able to be a basic platform for Internet of Things devices. The scanning method used to display the data is efficiently saving the needs of pins.

The result of fetching success rate test showed a stable average fetching time with a very high success rate of 99.6% for the duration of 23:53 to 02:46, and 99.7% for the duration of 07:24 to 09:43. The validity test showed a significant fluctuation of error caused by the different time unit, which furthermore causes an exponential calculation error. The GPRS quota usage test showed a stable GPRS quota usage with the average usage of 10.3 MB for a period of 10 days.

REFERENCES

- [1] C. S. Li and M. C. Chen, "A data mining based approach for travel time prediction in freeway with non-recurrent congestion," *Neurocomputing*, vol. 133, pp. 74–83, 2014.
- [2] R. Joumard, P. Jost, J. Hickman, and D. Hassel, "Hot passenger car emissions modelling as a function of instantaneous speed and acceleration," *Sci. Total Environ.*, vol. 169, no. 1, pp. 167–174, 1995.
- [3] C. Mrs.Swati, S. Mugade, S. Sinha, M. Misal, and B. Pooja, "Implementation of Real Time Bus Monitoring and Passenger Information System," *Int. J. Sci. Res. Publ.*, vol. 3, no. 5, pp. 1–5, 2013.
- [4] Misbahuddin and R. F. Sari, "Bayesian mixture model for prediction of bus arrival time," *Int. J. Technol.*, vol. 6, no. 6, pp. 961–970, 2015.
- [5] M. Schmidt, *Arduino A Quick-Start Guide*. Pragmatic, 2011.
- [6] S. Monk, *30 Projetos com Arduino*. New York Chicago San Francisco Lisbon London Madrid Mexico City Milan New Delhi San Juan Seoul Singapore Sydney Toronto: Bookman, 2014.
- [7] S. Barrett and M. Thornton, *Embedded Systems Design with the Atmel AVR Microcontroller Part I*. Morgan & Claypool Publishers, 2009.
- [8] S. Barrett, *Embedded Systems Design with the Atmel AVR Microcontroller: Part II (Synthesis Lectures on Digital Circuits and Systems)*. Morgan & Claypool Publishers, 2009.
- [9] M. Schwartz and S. Buttigieg, *Arduino Android Blueprints*. Birmingham - Mumbai: Packt Publishing Ltd., 2014.



- [10] E. R. Melgar and C. C. Diez, Arduino and Kinect Projects: Design, Build, Blow Their Minds. Apress, 2012.
- [11] B. Terence, GPRS 100 Success Secrets. Emereo Publishing, 2008.
- [12] Itead, "IComSat GSM / GPRS SIM900 Module Expansion Board Shield With Antenna for Arduino Mega," 2015. [Online]. Available: <https://www.itead.cc/icomsat.html>. [Accessed: 15-Nov-2015].