JURNAL INFOTEL
Informatics - Telecommunication - Electronics
Website Jurnal : http://ejournal.st3telkom.ac.id/index.php/infotel
ISSN : 2085-3688; e-ISSN : 2460-0997

A Custom Firmware and Lightweight Battery System Design for Portable RFID Reader

Hans Kasan[1], Trio Adiono[2,*], Suksmandhira Harimurti[3], Farkhad Ihsan Hariadi[4]
[1,2,4] School of Electrical Engineering and Informatics Institut Teknologi Bandung
[1,2,3,4] University Center of Excellence on Microelectronics Institut Teknologi Bandung
[1,2,4] 2nd Floor, Achmad Bakrie Building, Jl. Ganesha 10, Bandung 40132, Indonesia
[1,2,3,4] 4th Floor, PAU Building, Jl. Tamansari No. 126, Bandung 40132, Indonesia
Corresponding email:*tadiono@stei.itb.ac.id

Received 21 May 2018, Revised 27 May 2018, Accepted 29 May 2018

Abstract — Small to large-scale companies mostly have warehouses to store their inventories, and to manage them a warehouse management system is required. A low cost, yet powerful solution is using a portable RFID reader. In RFID portable reader system, there are three components which are the most essential, i.e. host and its firmware, RF module, and battery.. In this paper, we propose a custom firmware design, which is compatible with different RFID reader chips or development boards. The custom firmware is designed to work by triggering the execution of Electronic Product Code (EPC) Generation 2 protocol standard command on the reader chip. Hence, the firmware can fully utilize the reader chip's command. Furthermore, a lightweight battery system is also designed. Targeting for a high mobility use, a very lightweight Li-Pro battery, weighing of only 0.1 kg, is used for the battery system. It is also able to work at long operating hour up to 4 hours.

Keywords – RFID, firmware, Li-Pro battery, warehouse management system

I. INTRODUCTION

Companies with warehouses require a management system to manage their inventories [1]-[2]. With a warehouse management system, information about all items and their corresponding location can be obtained with ease and most importantly all of those items can be fully tracked, so that the company can have a full control of all items.

Warehouse management system has always been expensive, having an abundance of features-not all features are necessarily needed, and mostly uses a sophisticated automation system [3]-[5]. This advanced management system is only appropriate for large-scale companies with a massive items transaction. However, it is very not cost effective for small and medium-scale companies. This poses a problem in Indonesia, where the economy is mostly dominated by small to medium-scale companies (99.9%) [6]. This means that most industries in Indonesia cannot afford to buy an advanced automated warehouse management system. To overcome this issue, a much cheaper warehouse management system should be made.

A portable, handheld RFID reader can overcome this issue. It is very convenience for the user to use it, yet also powerful in locating and identifying the items in the warehouse. Fig. 1 depicts the example of portable RFID reader. In this system, RFID reader module and smartphone are a separate system. It means that both of them can be modified or change independently depends on the user need.

(a) (b)

Fig. 1. Portable RFID Reader. a) Side view, b) Top view with the smartphone is attached

Figure 2 shows the system diagram of portable RFID reader system. The reader itself has three main components: RF module, host, and battery. In portable RFID reader system, a microcontroller acts as the host. Since RF module is widely available in the market, the main challenge of designing a portable RFID reader is the host and battery.

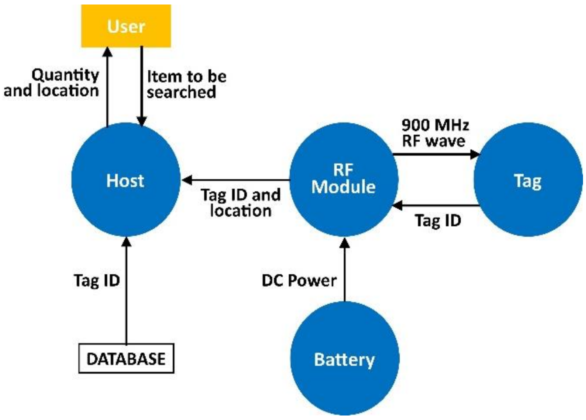

Fig.2. System diagram of portable RFID reader

Many works have been presented in designing the firmware of the host. However, most of them use a bundle of kit with a built-in microcontroller [7]-[9]. The bundle with built-in commercial microcontroller certainly offers a wide range of integration and functional capabilities. But, its firmware may not compatible with other microcontrollers. Hence, it lowers the flexibility of the system configuration. Moreover, the built-in microcontroller typically has a higher production cost since it is designed for multipurpose applications [10]-[11]. Therefore, in this work, we propose a custom firmware that can be applied in any microcontroller and most importantly it can be run on any development board. Even though designing the custom firmware is more complex than using the built-in firmware from a commercial microcontroller, it offers a high flexibility for a cross platform implementation as well as lowering the production cost.

Furthermore, battery is also an essential part of portable RFID reader system. It sets the working hours of the device. For portable RFID reader application, the battery system should perform a long working hour, yet lightweight. For a continuous scanning, a minimum 2 working hours should be fulfilled [12]. Several approaches have been explored in obtaining a long working hour as well as the lightness of a RFID reader system. L. Kumari *et al* [13] presented an ultra-lightweight battery using a paper battery for RFID application. Another approach is designing an efficient reading system by rigorous calculation of reading load and rate, as presented in [14]. Another straightforward approach is using a bigger battery capacity to achieve a long working hour, as presented in [15]. Here, in this work, we present a lightweight battery system design, which is able to work at long working hour, using a low cost and low complexity battery system. In our approach, to lower the complexity of the system, we use a commercially available battery. To ensure the lightness and meet the long working hour requirement, we systematically calculate the system load, so that an optimum number and capacity of battery cell can be selected for the system.

II. RESEARCH METHOD

A. Firmware Design

The firmware design comprises of several programming layers, as shown in Fig.3. These layers are correlated to the communication sequence that the firmware has to follow when it communicates with a RFID reader chip.

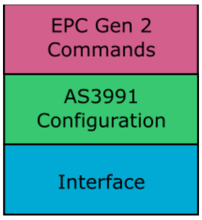

Fig.3. Firmware programming layers

Interface layer consists of functions and procedures to execute communication with reader chip via a parallel protocol. In this protocol, the microcontroller always acts as the master, controlling the clock that is fed to the reader chip which acts as the slave.

AS3991 Configuration layer is used to execute self-test procedures, configure operating frequency and PLL (phase-locked loop) locking. Self-test procedure consists of I/O and EN pins checks. PLL locking is required to alter the operating frequency of the internal oscillator.

EPC Gen 2 Commands layer is used to execute commands which comply with the EPC Gen 2 Commands standard. The commands are used to interact with RFID tags.

The layers are inter-dependent to each other, as shown in Fig.4.

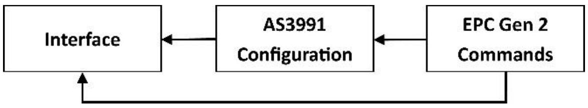

Fig. 4. Inter-layer dependency

In the Interface layer, parallel communication protocol is used as the communication mode between firmware and AS3991 reader chip. The advantage of using parallel communication is that it supports both single and continuous communication modes. In the single communication mode, the master will read and write to a single register, whereas in the parallel communication mode, the master will read and write to a register with the address sent previously, and then continue reading and/or writing to its adjacent registers. Consequently, there are several communication modes in parallel communication protocol, as shown in Table 1.

Table 1. Modes of communication in the parallel communication protocol

Mode	Description
Single Read	reading of 1 byte data from 1 register
Single Write	writing of 1 byte data to 1 register
Single Command	writing of 1 byte command to be run by the chip
Continuous Read	reading of more than 1 byte data from a register with adjacent address
Continuous Write	writing of more than 1 byte data to a register with adjacent address

All of the communication modes in parallel protocol are realized based on writeread() procedure. The dependency of functions and procedures related to the communication modes in parallel protocol is shown in Fig.5.

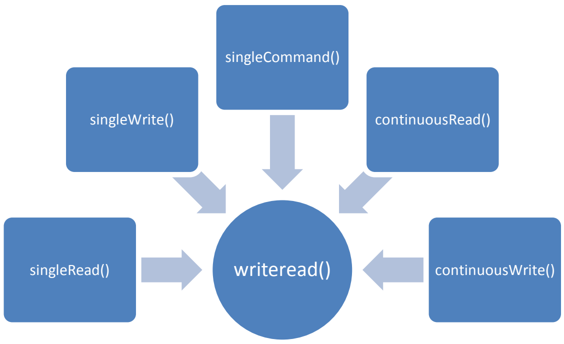

Fig.5. Dependency diagram of all functions and procedures in parallel protocol

To use a function and procedure, it has to be called in writeread() procedure and an input parameter has also to be defined. There are several input parameters in writeread() procedure, as follows,

a) **wbuf byte*, an array of bytes to be written to the chip
b) wr_en byte, a number of bytes to be written to the chip
c) **rbuf byte*, a buffer array of bytes used to store bytes sent by the chip
d) *rd_en byte*, a number of bytes to be read from the chip
e) *stop_mode byte*, writing a stop sequence state to the chip. 0 means single stop sequence, 1 means continous stop sequence, and 2 means stop sequence will not be written
f) *doStart byte*, wiriting a start sequence state to the chip. 1 means start sequence will be written to the chip, while 0 means start sequence will not be written.

The flowchart of writeread() procedure is illustrated in Fig.6.

The AS3991 Configuration layer is run by executing the initialization_chip(frequency) procedure, which works according to the flowchart shown in Fig. 7.

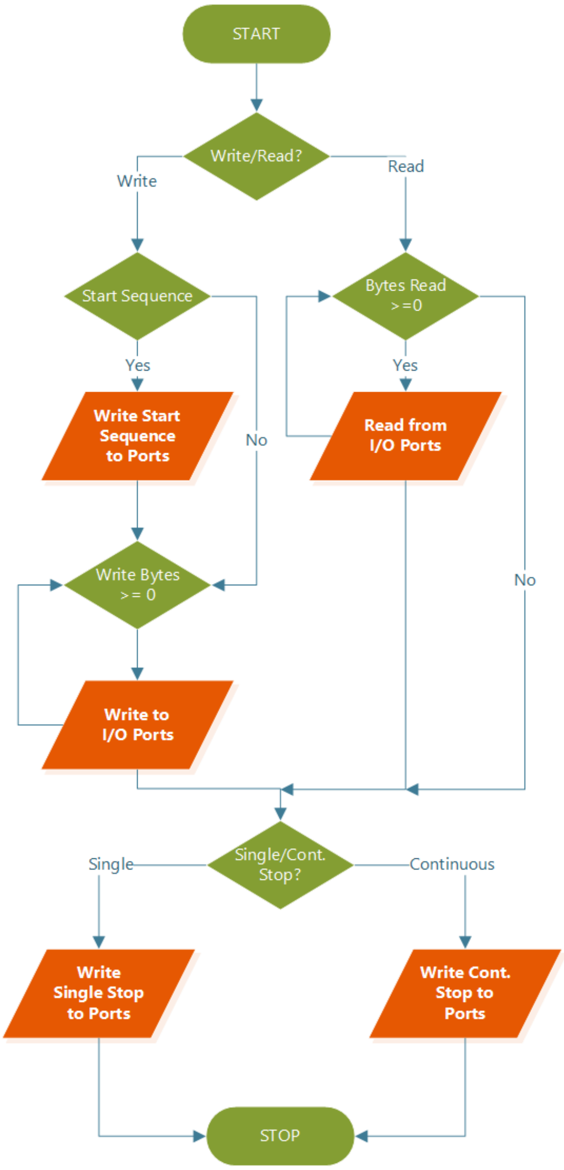

Fig. 6. writeread() procedure flowchart

The selectLinkFrequency(int a) procedure has a function to change the link frequency, which associates with the frequency of the backscatter signal from the tag. The link frequency is regulated using the register RX Options. The bit configuration of RX Options is shown in Table 2.

Table 2. Configuration bit of RX Options register

Bit	Sinyal	Function
7	Rx_LF3	Link frequency selection
6	Rx_LF2	
5	Rx_LF1	
4	Rx_LF0	
3	-	Don’t care
2	-	Don’t care
1	TRext	RX preamble length
0	-	Don’t care

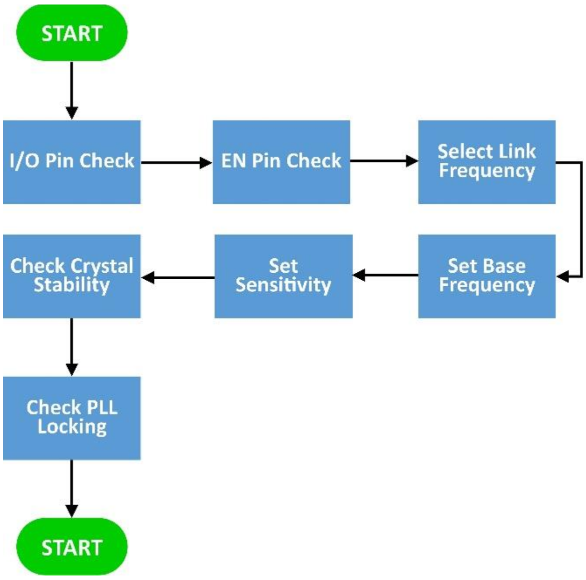

Fig. 7. Initialization_chip() flowchart

The integer a in selectLinkFrequency(int a) procedure has a function to set the RX bits (bit 7, 6, 5, and 4). For example, if a is set to 2, all four RX bits will be bitwise OR with a. If a is set to 0, the default link frequency from the chip, 160 kHz, will be used.

The setBaseFrequency() procedure has a function to regulate the output frequency of PLL, so called as frequency hopping. The base frequency of the reader chip is defined in (1) [16].

$$Frequency = Freq_{ref} \times (32A + 33B) \quad (1)$$

Firstly, the procedure will read the current working frequency of the chip. Then, when the user input a new base frequency value, the function will find the corresponding A and B value. The values will then be stored in PLLMAIN register. The bit configuration in PLLMAIN register is shown in Table 3.

Table 3. Configuration bit of PLLMAIN register

Bit	Signal	Function
23	ai2x	Increasing the internal PA bias by two folds
22:20	RefFreq<2:0>	PLL reference frequency Frekuensi referensi PLL
19:10	B value	PLL divider value
9:0	A value	

The EPC Gen 2 Commands layer is used to search for tags in surroundings. It is run by executing the gen2SearchForTags() function, which works according to the flowchart shown in Fig.8.

The function supports tag searching with and without a mask. If the tag searching uses a mask, a select command will be sent, so that only the tag the correct EPC which will respond. If the mask is not used, all the tags within the antenna range will respond.

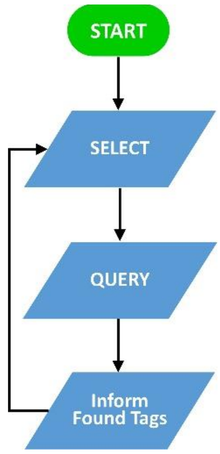

Fig. 8. gen2SearchForTags() flowchart

B. Battery System Design

The portable RFID reader should be as light as possible, as it will be carried around when locating the items. Therefore, Li-Po (Lithium Polymer) battery is used. To choose the appropriate Li-Pro battery for portable RFID reader application, the system current consumption is calculated beforehand.

The system current consumption is approximated as the total of current consumed by all active components, such as AS3991 reader chip, power amplifier, and microcontroller at the time when the system is transmitting or receiving data from/to tag. Table 4 shows the calculation of the system current consumption.

Table 4. Current consumption calculation of the system

Load	Current (mA)
AS3991 reader chip	310
Power amplifier	610
Microcontroller	20
Dissipated current	10
Total current consumption	950

In our design, the reader is required to transmit and/or receive data continuously for a minimum 2 hours long. To have a longer battery lifetime in real practice, the battery should not be completely drained, therefore in our design only 90% of the total capacity will be used. The required battery capacity is then calculated using (2) [17].

$$C_{req} = \frac{I_{total} \times t}{C_{usage}} \quad (2)$$

where

C_{req} = required battery capacity (mAh)
I_{total} = total current consumption (mA)
t = required usage duration (hours)
C_{usage} = battery capacity usage limit (%)

By inputting all the values, the required battery capacity is 2110 mAh.

The Li-Pro batteries which have the closest capacity are the 2000 mAh and 2200 mAh Li-Po battery. For our design, we choose to use the 2 cells Turnigy 2200 mAh 3S 25C Li-Pro battery. The corresponding battery is shown in Fig.9.

Fig.9. Two Cells 2200 mAh 20C Li-Po Battery

The 2 cells Li-Po battery produces 7.4-8.4 V DC, which exceeds the recommended input voltage of the reader chip, power amplifier, and microcontroller-require only 5 V DC. To regulate the voltage, an additional LM7805 regulator is used. This regulator is used with the following considerations,

a) it operates at 7–25 V DC input voltage
b) it can produce an output current up to 1.5 A

III. RESULT

A. *Firmware Test*

The firmware is tested to check its functionality within all the programming layers. Therefore, in this firmware test, there three aspects that will be tested, which are

a) I/O and EN pins self-test, to test the Interface layer
b) PLL locking test, to test the AS3991 Configuration layer
c) tag searching test, to test the EPC Gen 2 Commands layer

The test configuration diagram is shown in Fig.10. A Linksprite RF development board is used as the testbench board, while Arduino running on the firmware acts as the DUT (Device Under Test).

The onboard microcontroller on the development board is put on reset, so that it will not interfere with the signals from/to the Arduino.

The response from the AS3992 reader chip is read by the Arduino and shown by the serial monitor provided by the Arduino IDE (Integrated Development Environment).

The I/O pin self-test and PLL locking test are successful, as shown in Fig.11.

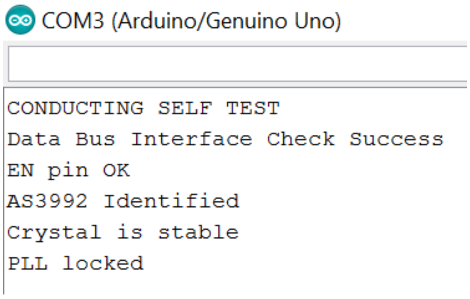

Fig. 11. I/O pin self-test and PLL locking test result

If one or more pins are disconnected, the firmware will display the corresponding error message, as shown in Fig.12.

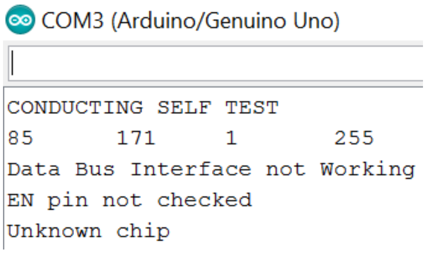

Fig. 12. Error messages when I/O pin self-test is failed

The tag searching test result is also successful, as shown in Fig.13. The firmware is successfully able to send tag searching command to reader chip.

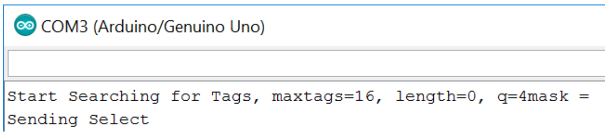

Fig. 13. Tag searching test result

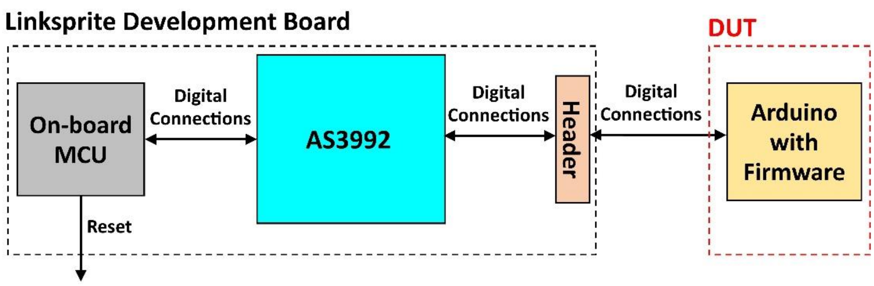

Fig. 10. Configuration diagram of firmware testing

The firmware test result summary is provided in Table 5. The test result proofed that the custom firmware design is successfully able to communicate well with the RFID reader chip and instruct the tag searching command.

Table 5. Firmware test result

Programming Layer	Test Result
Interface layer	I/O and EN pin are successfully connected Disconnected pin can also be detected
AS3991 Configuration layer	PLL is successfully locked
EPC Gen 2 Commands layer	Tag searching command is successfully instructed

B. Battery Test

The configuration diagram for the battery test is shown in Fig. 14. A Skyetek Nova Development Board is used for this measurement, instead of Linksprite RF Develepment Board, to also show our firmware compatibility with other RFID development boards.

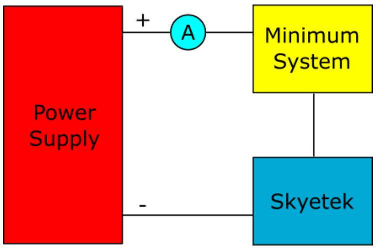

Fig. 14. Configuration diagram of battery testiug

The battery life/operating time is calculated by the following steps:

a) instructing the reader chip to transmit and receive data continuously
b) measuring the current consumed by the system and module when the module is transmitting and receiving data
c) measuring the battery life/operating time using (3) [17]

$$t = \frac{C}{I} \quad (3)$$

where
t = battery life/operating time (hours)
C = battery capacity (mAh)
I = total measured current (mA)

From the measurement result, provided in Table 6, the total current consumption is 434 mA. Hence, it means that the 90% capacity of a 2200 mAh battery is able to power up the whole system for up to 4 hours, 33 minutes, and 43 seconds, which surpasses the required working hour.

Table 6. Battery test result

Parameter	Value
Battery capacity	2200 mAh
Real usage of battery capacity	90%
Measured current consumption	434 mA
Battery life/operating time	**4 hours 33 minutes 43 seconds**

IV. DISCUSSION

The testing result reflects two important aspects in our design. Firstly, the firmware is proven to be compatible with different development boards. Here, in this work, we tested the firmware on Linksprite development board with AS3992 RFID reader chip and Skyetek Nova development board with ARM Cortex chip. Each board has different chip. The firmware is however able to successfully run on both chips and executing the instructed procedures. This means that the system designer can choose any chip or development board they would like to use without any worry of compatibility issues.

Secondly, by considering the lightness of the battery, we are able to design a lightweight battery system with using the Turnigy 2200 mAh Li-Pro battery, weighted only 0.1 kg, for a continuous operating hour up to 4 hours. This operating hour passes already the required minimum continuous operating hour of 2 hours. The lightness of the battery is essential since the portable RFID reader will be carried all the way by the user during the operation. Weighing of only 0.1 kg, the battery will negligibly have an impact of weight to the total weight of the portable RFID reader.

V. CONCLUSION

A custom firmware which is able to work on any development board with different chips and a lightweight battery system that is able to support a long working hour has been demonstrated. The custom firmware is able to communicate via parallel protocol with the reader chip and configure the reader chip internal PLL. This is shown by the successful test results on the development board. It is also proven that our custom firmware is able to be run on different chips, giving a high flexibility to system designer to implement the firmware. Moreover, by using the Turnigy 2200 mAh Li-Pro battery, a lightweight battery system, which has a long continuous operating hour, up to 4 hours long, is able to be realized. Hence, the portable RFID reader itself is expected to be light enough to be carried comfortably by the user.

REFERENCES

[1] J. C. Chen, C. H. Cheng, P. B. Huang. (2013, July). Supply chain management with lean production and RFID application: A case study. *Expert Systems with Applications*. [Online]. *40(9)*, pp. 3389-3397. Available: https://doi.org/10.1016/j.eswa.2012.12.047

[2] R. Y. Zhong, Q. Y. Dai, T. Qu, G. J. Hu, G. Q. Huang. (2013, Apr.). RFID-enabled real-time manufacturing execution system for mass-customization production. *Robotics and Computer-Integrated Manufacturing*.

[Online]. *29(2)*, pp. 283-292. Available: https://doi.org/10.1016/j.rcim.2012.08.001

[3] G. Q. Huang, M. Z. Q. Chen, J. Pan (2015, Apr.). Robotics in ecommerce logistics. *HKIE Transactions*. [Online]. *22(2)*, pp. 68-77. Available: https://doi.org/10.1080/1023697X.2015.1043960

[4] M. A. K. Bahrin, M. F. Othman, N. H. N. Azli, M. F. Talib (2016, March). Industry 4.0: A review on insdustrial automation and robotic. *Jurnal Teknologi*. [Online]. *78(6-13)*, pp. 137-143. Available: https://doi.org/10.11113/jt.v78.9285

[5] J. Irizarry, E. P. Karan, F. Jalaei (2013, May). Integrating BIM and GIS to improve the visual monitoring of construction supply chain management. *Automation in Construction*. [Online]. *31*, pp. 241-254. Available: https://doi.org/10.1016/j.autcon.2012.12.005

[6] (2015, Sep.). Profil Bisnis Usaha Mikro, Kecil dan Menengah (UMKM). LPPI-Bank Indonesia. [Online]. Available: https://www.bi.go.id/id/umkm/penelitian/nasional/kajian/Documents/Profil%20Bisnis%20UMKM.pdf

[7] R. Colella, D. D. Donno, L. Tarricone, L. Catarinucci (2014, June). Unconventional UHF RFID tags with sensing and computing capabilities. *Journal of Communications Software and Systems*. [Online] *10(2)*, pp. 83-89. Available: https://doi.org/10.24138/jcomss.v10i2.128

[8] D. D. Donno, *et al* (2014, June). Integrating passive UHF RFID tags with WSN nodes: Challenges and opportunities. *Journal of Communications Software and Systems*. [Online] *10(2)*, pp. 99-106. Available: http://dx.doi.org/10.24138/jcomss.v10i2.130

[9] J. D. L. Morenas, A. García, J. Blanco (2014, Dec.). Prototype traceability system for the dairy industry. *Computers and Electronics in Agriculture*. [Online] *101*, pp. 34-41. Available: http://dx.doi.org/10.1016/j.compag.2013.12.011

[10] M. H. U. Abdeen, U. S. Khan, J. Iqbal (2016, June). A multipurpose vehicle tracking system based on ARM CORTEX-M3 STM32, HMC5883L, MPU-6050, GSM and GPS. *Journal of Traffic and Logistics Engineering*. [Online] *4(1)*, pp. 7-12. Available: http://dx.doi.org/10.18178/jtle.4.1.7-12

[11] A. R. Zubair, O. K. Atusiuba (2016, Dec.). Microcontroller based signal processing for multi-source energy flow control and voltage regulation. *International Journal of Engineering and Technology*. [Online] *6(12)*. Available: http://iet-journals.org/archive/2016/december_vol_6_no_12/6876751481214876_abstract.php

[12] (2014, Jan.). EmbiVentory UHF handheld RFID reader. Embisphere. [Online]. Available: http://www.embisphere.com/en/rfid-products/embiventory-power

[13] L. Kumari, K. Narsaiah, M. K. Grewal, R. K. Anurag (2015, June). Application of RFID in agri-food sector. *Trends in Food Science & Technology*. [Online] *43(2)*, pp. 144-161. Available: https://doi.org/10.1016/j.tifs.2015.02.005

[14] X. H. Lin, *et al* (2015, Oct.). Balancing time and energy efficiencies with identification reliability constraint for portable reader in mobile RFID systems. *Computer Networks*. [Online] *92*. pp. 55–71. Available: http://dx.doi.org/10.1016/j.comnet.2015.09.017

[15] L. Lu (2014, Sep.). Research on how to design an android-based RFID reader. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. [Online] *12(9)*. pp. 6431-6436. Available: http://dx.doi.org/10.11591/telkomnika.v12i9.3945

[16] *AS3990/AS3991 UHF RFID Single Chip Reader EPC Class1 Gen2 Compatible Data Sheet*, Revision 3.81, Austria Microsystems AG, Austria, 2009.

[17] K. Korhonen, “Predicting mobile device battery life,” M.S. thesis, Dept. Commun. and Networking, Aalto University, Espoo, Finland, 2011.