International Journal of Control and Automation
Vol.9, No.2 (2016), pp.9-18
http://dx.doi.org/10.14257/ijca.2016.9.2.02

Development of the Logging System for LKAS Electronic Control Unit Algorithm Verification

Hyoung-Keun Park[1*]

[1*] *Dept. of Electronic Engineering, Namseoul University,*
91 Daehak-ro Seonghwan-eup Sebuk-gu Cheonan-si, Chungnam-do, South Korea
phk315@nsu.ac.kr

Abstract

Recently, car makers and the governmental agencies make a greater investment in and efforts to develope the application technologies for "Advanced Safety Vehicle" than ever, it is very necessary to verify each of algorithms applied to the core ECUs as existing cars have rapidly evolved into sophisticated assembles of computer devices equiped with tens of ECUs while geared with high-tech IT technologies. Therefore, this study developed a verification logging system for the electronic control algorithm of a lane keeping assistance system camera. The developed system can not only acquire the lane information of a road through the camera, but also analyze such information and adjust a vehicle automatically through controlling the control board.

Keywords*: Advanced safety vehicle, ECU algorithm verification, Data logging, LKAS, Automotive camera*

1. Introduction

As smart cars have recently become more necessary and commercializing them is more demanded in automobile market, active safety measures such as integration control of chassis and non-chassis, active accident prevention and avoidance, lane departure and obstacle warning, autonomous driving, and safety support for passengers and pedestrians are equally asked to be more robust than ever. In addition, efforts are actively being made to establish the concept of vehicle-traffic informationization and commercialization to solidify the convenience of a vehicle through connection with Intelligent Transportation System, which is vehicle information system.

Although the car makers and the governmental agencies make a greater investment in and efforts to develope the application technologies for "Advanced Safety Vehicle" than ever, it is very necessary to verify each of algorithms applied to the core ECUs as existing cars have rapidly evolved into sophisticated assembles of computer devices equiped with tens of ECUs while geared with high-tech IT technologies.

Particularly, logging system is necessary to record and retain working conditions and to record various kinds of information to analyze drivers' driving habits and system functions. In addition, it can understand the error status and console operation status by examining working logs such as error logs (error information) of hardware as well as general logs such as manipulation text, command text, and report text notified to center operator to record and retain driving conditions.

Therefore, this study developed a verification logging system for the electronic control algorithm of a lane keeping assistance system camera. The developed system can not only acquire the lane information of a road through the camera, but also analyze such information and adjust a vehicle automatically through controlling the control board. It is believed that the developed system can thoroughly verify ECU control algorithm, which is a core element directly connected to passenger's safety, and enhance the safety of passengers.

ISSN: 2005-4297 IJCA

2. Verification Logging System for ECU Algorithm

MCU, which is the essential part of a control panel that governs an intelligent vehicle, is designed to control electronic devices through image information obtained from a camera as well as errors in its algorithm itself. Therefore, there exists a possibility that distorted image due to a problem in a camera and optical lens themselves lead to false recognition of road information, which can cause a serious situation. In this case, the system can end up in a drastic result, rather than securing safety. When an algorithm works with incorrect information, it can cause error. Therefore, the algorithm applied inside of MCU should be designed to detect the defects of a camera and optical lens by itself and runs the electronic devices of a car correctly.

Therefore, it is imperative to run objective and precise tests and verification on the various parameters of electronic device control algorithm that works with camera image-based information, and carry out modification and correction. Control board algorithm verification system for LKAS (lane keeping assistance system) camera of a smart car that was developed in this study consists of synchronization controller unit, MCU and communication unit as seen in Figure 1.

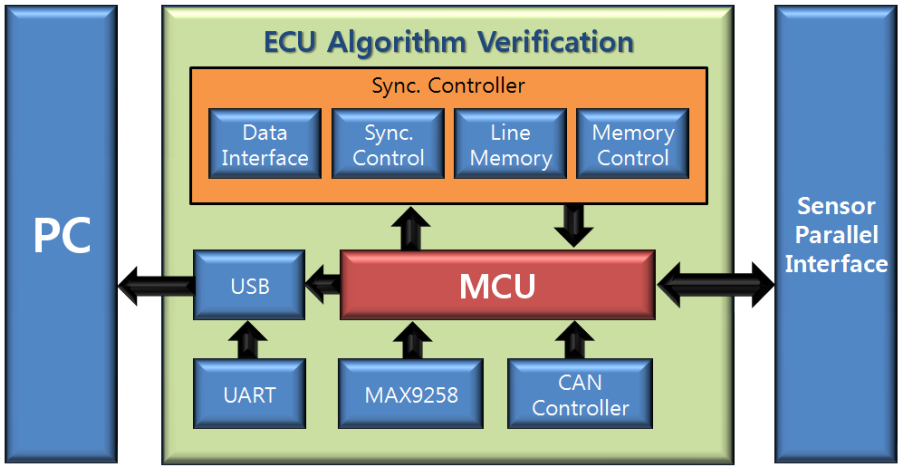


Figure 1. Logging System for Algorithm Verification in the LKAS Camera

In Figure 1, synchronization controller consists of data interface, synchronization control, lne memory and memory control unt while communication unit consists of MAX9258, CAN (Controller Area Network), and UART (Universal Asyncronous Receiver/Transmitter). This study defined the specifications of signal activity pixel coming in the system, the form of CMOS image sensor output data, UART transmission data and others and then saving format of image (video) output from the system and image display method as requested by a user.

2.1. CAN Data Bus and Operational Principle

CAN data bus is manly used for data transmission between ECUs of vehicle safety system and convenience system and to control information/communication system and entertainment system. CAN transmits data through 2 twisted or concealed data wires. In addition, CAN works according to multi-master principle that multiple ECUs perform ECU master in master-slave system.

2.1.1. Can Data Bus System

CAN data bus system consists of 2 nodes, CAN-low wire, CAN-high wire, and at least 2 terminal resistors. CAN bus node has the same internal structure as LIN-bus node. High-quality controller and transceiver are used for CAN bus node to speed up data transmission and distribute a different level of voltage to us from in LIN-data bus. Magnetic field, which is created in the two CAN-wires whenever switched, is offset because the voltage run in opposite directions to each other. Therefore, the two wires are electronically neutral for external environment, which means that they do not cause any external interference. In other words, they secure resistance to interference. A terminal resistor connects the circuit of CAN-high wire and CAN-low wire. It is done so to prevent CAN-bus wire from making reflection. It is usually installed in a node. However, terminal resistance-free CAN-bus wire can cause functional error in CAN class C system. Therefore, terminal resistor should be checked first of all when error occurs. Terminal resistance can be tested at the contact point of CAN wire by a resistance measuring apparatus in CAN class C system.

2.1.2. Operational Principle of CAN

Each node (ECU) can transmit a message to bus wire according to multi-master principle except when the latter is in transmission of information through bus wire. When it is necessary multiple ECUs transmit message at the same time, the most important message is transmitted through arbitration. Arbitration is a process that monitors, manages and arbitrates multiple processes or users' competing requests for one resource. When multiple ECUs want to transmit message at the same time, it controls access to data bus wire. The importance (priority) of messages is defined by ID. Lower ID has high priority.

2.2. Outline and Specification of Hardware

2.2.1 Outline of Hardware

The verification logging system for the electronic control algorithm of a lane keeping assistance system camera, which was developed by this study, is used for actual vehicle test or a racing car. It saves the data obtained from various sensors while driving in direct circuit of a computer memory chip and outputs when a car is in halt or reads then in connection with a telemeter system while driving. The hardware of the developed system continues transmitting to PC or a notebook in real time the image data of an automotive mega image sensor through USB 3.0 interface while reading CAN data simultaneously coming to two CAN ports, and adds them to the image data. Table 1 shows the specifications of the hardware.

Table 1. Hardware Specifications

Item	Description
Input Image Sensor Interface	FPD_LinkⅢ, format : YUV422, size : 1280x800, FrameRate = 30fps, 8/10bit
Input CAN Interface	CAN V2.0B, 2Port Support
PC Upload Interface	USB3.0 Interface
Onboard Buffer Memory	512Mbit x 2 Buffer
Parallel Debugging Interface	13 x 2 Connector
Configuration Sensor Registers	Programmable USB Packet Support
Power Supply	Multi Power Supply (DC5V/3A, USB power)
Dimensions	105 X 105 X 33(mm)

Figure 2 shows the verification logging system for the electronic control algorithm of a lane keeping assistance system camera, which was developed by this study, and the functions of each block are as follows.

1) Main Controller
2) FPD_LinkⅢ/CAN Interface
3) CAN Interface,
4) FPD_LinkⅢ De-serializer
5) CAN Controller
6) USB 3.0 Controller
7) Multi Power Switch
8) Parallel Interface

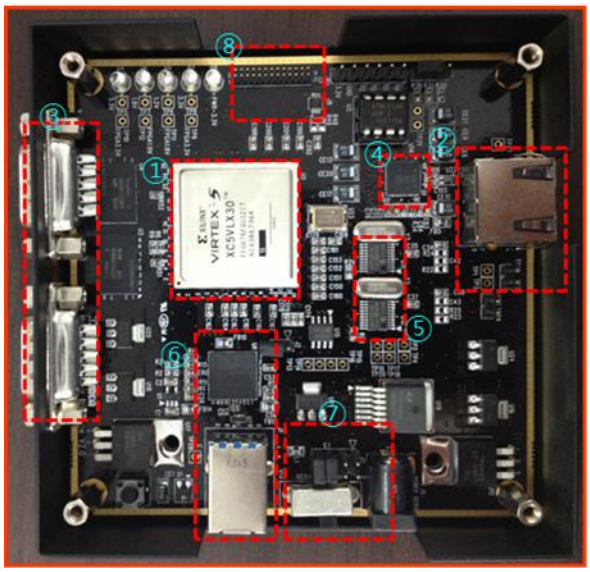


Figure 2. Developed Logging System

2.2.2. Detailed Design and Dependence

1. LVDS to FPGA

The interrelation of LVDS de-serializer and FPGA main controller is that video data coming in LVDS (low volume dissemination system) are converted into parallel data (Vsync, Hsync, Data0~7, Pclk) through de-serializer chip (DS90UB914) and enters into the input pin of FPGA.

2. LVDS De-serializer Signal

Of input/output pins, all the pins except LVDS pin are controlled by 1.8V IO power. Figure 3 shows the block diagram of de-serializer chip (DS90UB914) used in this study.

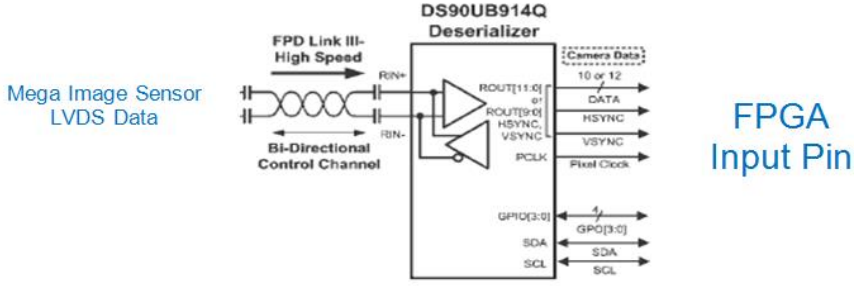


Figure 3. Block Diagram of De-Serializer Chip (DS90UB914)

3. CAN to FPGA

As for the motions of CAN controller and FPGA main controller, CAN data coming through the ports are saved in the internal buffer through CAN controller chip (MCP2515) and input in FPGA main controller in SPI communication mode. Since the internal structure of MCP2515 is made in dual buffer, it converts CAN data in real time and continually sends them to FPGA.

4. AN Controller Signal

Both input and output pins of CAN controller are controlled by 3.3V I/O Power. Figure 4 shows the connection of CAN controller pins.

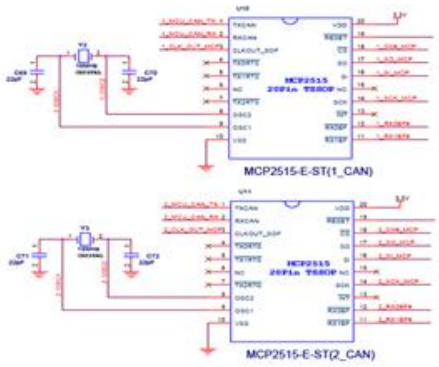


Figure 4. CAN Controller Pin Connection

5. FPD-LinkⅢLVDS 와 CAN connector

The chip, which is used as FPD-LinkⅢ LVDS and CAN connector, is MOLEX/ 04462000002/RJ-45 as shown in Figure 5. LVDS input pins are RIN0+/RIN0- and dual port CAN input pins are 1_CAN_H, 1_CAN_L, 2_CAN_H, 2_CAN_L.

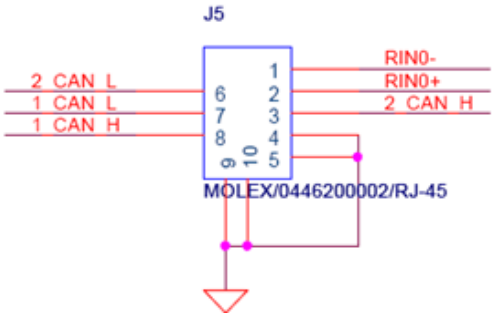


Figure 5. FPD-Link ⅢLVDS 와 CAN connector

6. FPGA to USB

The relation of USB controller and FPGA is that video data coming in through LVDS input pins and dual CAN data from CAN input pins are combined into one video frame by FPGA and those data are continuously transmitted to USB 3.0 interface of PC or a notebook through USB Controller (USB3014) chip in real time.

7. The Format of Data Transmitted from FPGA to USB Controller

Figure 6 shows the formats of data transmitted from FPGA to USB controller.

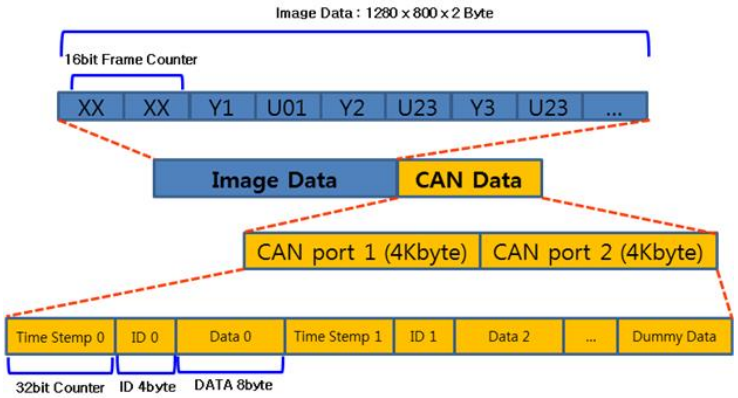


Figure 6. Transmit Data Format of FPGA to USB Controller

2.2.3. Main Controller (FPGA)

The top module is SanDaCon.v module, consisting of as in Figure 7. It plays a role of connecting each module.

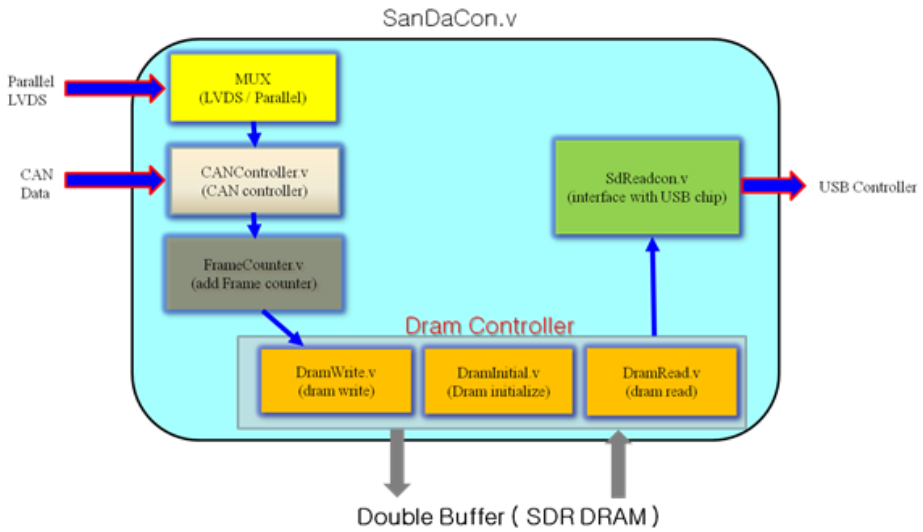


Figure 7. Structure of Main Controller (FPGA) Top Module

In the top module, input data format has two paths of 'parallel' and 'LVDS de-serializer' and comprises uniformed data (Vsync, Hsync, Pclk, Data). CAN data that are converted to SPI are added to image data through CANController.v module. FrameCounter.v module adds count data to the beginning part of image data coming in real time and delivers them to Dram controller part. Dram controller consists of DramWrite.v module that inputs entered image data in Dram; DramRead.v module that outputs the data of Dram; and DrameInitial.v that initializes Dram. Image data that output from Dram are sent to CY3014 chip, which is USB controller, through SdReadCon.v module.

After receiving interrupt signal from two CAN controllers (MCP2515), main controller receives CAN data by through SPI interface and saves them in the internal buffer, synchronizes them to incoming image data frame, merges CAN data to image data and sends them out. At this point, each of CAN data has the format as shown in Figure 8.

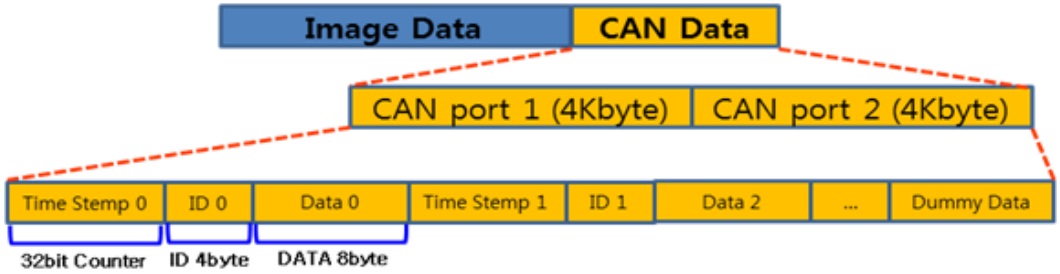


Figure 8. CAN Data Format

Increasing their addresses of image data (Vsync, Hsync, Data, Pclk) in order, DramWrite module records them in SDR Dram. Ras-Cas latency is 3 and uses Bust 8 mode to record them in Dram. It is a module that increases data address stored in Dram according to Read pin High and reads and transmits them. When the data are transmitted, it helps valid data synchronize at the next phase by maintaining valid signals at high state. They are synchronized with input Enb signals, keeping ReadOut pin high; ReadOut signals are connected to read signals of DataRead.v module; and the data are read in Dram and transmitted to USB controller part. at this point, the data are put for each FiFo according to FifoAd[1:0] signal.

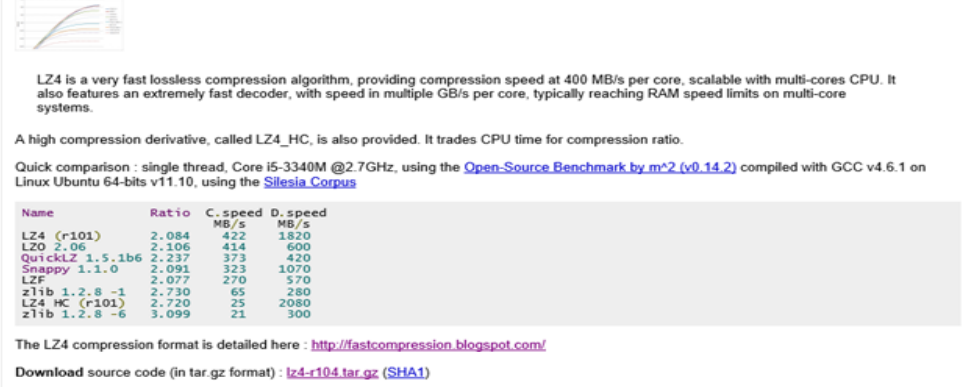
3. Performance Test

3.1. Performance Measuring Environment

- Conversion of files into save folder into 8bit and lossless compression (LZ4 algorithm)
- Maintaining synchronization with existing program while moving files

3.2. Detailed Performance Test

When compressing word type, changing the result scenes of feasibility test, this study measured that the condition with lens has two times higher capacity of word type than that with no lens. Even it turned out 4 times higher than byte type. It can be explained from the fact that word type has more empty space than byte type, so the former has higher compression efficiency than the latter. In case that word type is compressed while logging, it took 50% more than otherwise (no logging), which disabling the option of increasing compression efficiency. Therefore, when save file in existing word type is converted into that in byte type and compressed, it can increase 40% ~ 45% more of existing storage capacity.



LZ4 is a very fast lossless compression algorithm, providing compression speed at 400 MB/s per core, scalable with multi-cores CPU. It also features an extremely fast decoder, with speed in multiple GB/s per core, typically reaching RAM speed limits on multi-core systems.

A high compression derivative, called LZ4_HC, is also provided. It trades CPU time for compression ratio.

Quick comparison : single thread, Core i5-3340M @2.7GHz, using the Open-Source Benchmark by m^2 (v0.14.2) compiled with GCC v4.6.1 on Linux Ubuntu 64-bits v11.10, using the Silesia Corpus

Name	Ratio	C.speed MB/s	D.speed MB/s
LZ4 (r101)	2.084	422	1820
LZO 2.06	2.106	414	600
QuickLZ 1.5.1b6	2.237	373	420
Snappy 1.1.0	2.091	323	1070
LZF	2.077	270	570
zlib 1.2.8 -1	2.730	65	280
LZ4 HC (r101)	2.720	25	2080
zlib 1.2.8 -6	3.099	21	300

The LZ4 compression format is detailed here : http://fastcompression.blogspot.com/

Download source code (in tar.gz format) : lz4-r104.tar.gz (SHA1)

Figure 9. LZ4 Compression Test

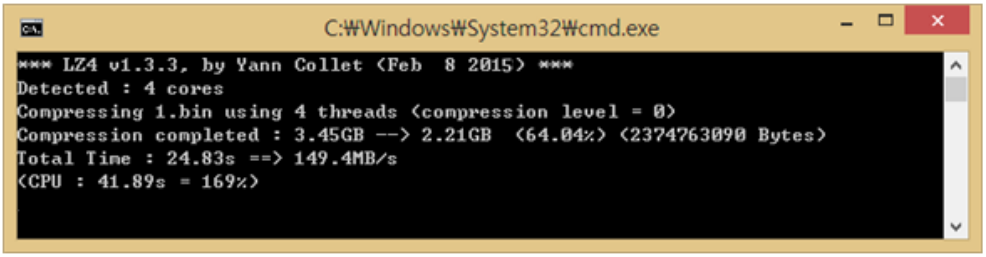


Figure 10. Compression Speed Test (Word image_30 [sec])

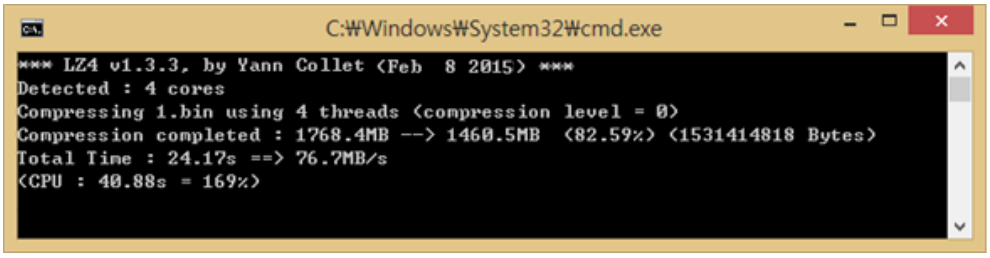


Figure 11. Compression Speed Test (BYTE image_30 [sec])

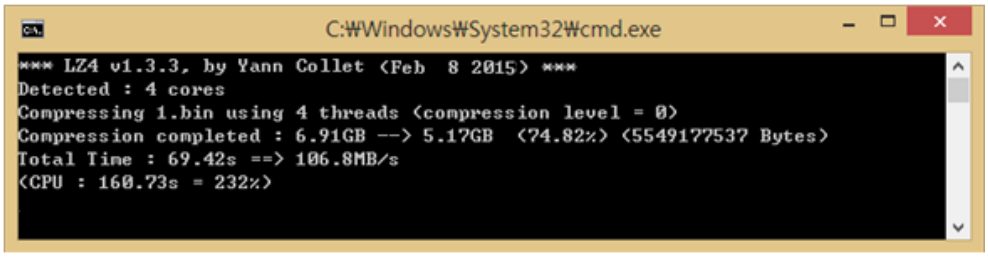


Figure 12. Compression Speed Test (BYTE image_60 [sec])

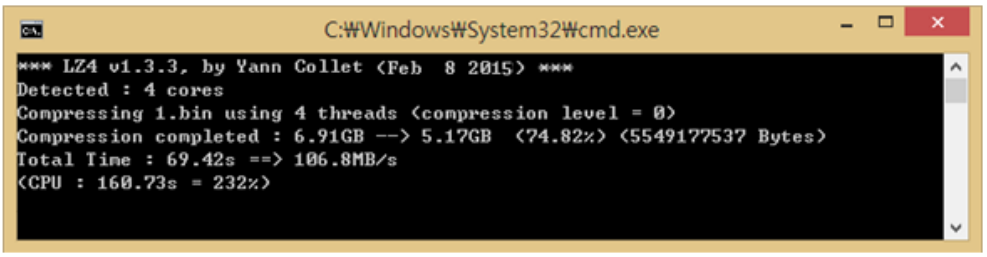


Figure 13. Compression Speed Test (BYTE image_120 [sec])

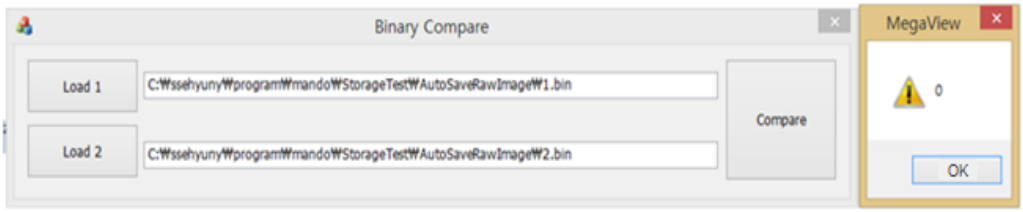


Figure 14. Binary Compare Result

3.3. Performance Analysis

This study developed not only synchronized file transfer program, but also made it possible to convert files in save folder into byte format, compress and shift them to a removable storage, so that logging program and synchronization can be maintained when moving data. In addition, this study developed decompression S/W that alarms save capacity, sets up and aligns save folder target. As a result, it made it possible to upgrade speed 30% more and secure stability through upgrading FPGA codes and USB3.0 Firmware for respective purpose.

4. Conclusion

The verification logging system for the electronic control algorithm of a lane keeping assistance system camera, which was developed by this study, can be applied to ASV. As a result, it can be possible to record various kinds of information while a car is in motion to record and retain working conditions and analyze drivers' driving habits and system functions. In addition, it can understand error status and console operation status by examining working logs such as error logs (error information) of hardware as well as general logs such as manipulation text, command text, and report text notified to center operator to record and retain driving conditions.

Therefore, it is believed that the system developed in this study can lead the part industry related to future intelligent vehicle and be also applied to other industries such as industrial unmanned vehicle and leisure vehicle as well as thoroughly verify ECU control algorithm, which is a core element directly connected to passenger's safety and enhance its safety.

Table 2. Performance Analysis

Performance Specification	Values
1. CAN data Logging	≧ 120EA/sec
2. Frame Drop	Real Time Logging(No Drop)
3. Data Synchronize Delay	≦ Delay 16ms
4. Activity Pixel	752(H) × 480(V)
5. CMOS Image Sensor Output Data	10Bit RAW Data
6. UART Trans. Data	8Byte × 250 (Max)
7. CAN Communication	Dual Control
8. Frame Save Format	(7752(H)X480(V)X10Bit RAW Data)+Data(1024Byte)
9. Image Display	Gray Image(7752(H)× 480(V)×8bit×3)
10. Save File Format	10bit Image RAW Data

Acknowledgments

Funding for this paper was provided by Namseoul University.

References

[1] J. Hwang, K. Huh, H. Na, H. Jung, H. Kang and P. Yoon, “Evaluation of Lane Keeping Assistance Controllers in HIL Simulations”, Proceedings of the 17th World Congress The International Federation of Automatic Control, Seoul, Korea, **(2008)**.

[2] C. Connolly, “A review of data logging systems, software and applications”, Sensor Review, vol. 30, no. 2, **(2010)**, pp.192-196.

[3] O.R. Omotayo, "Data logging with microcomputers", Information and Software Technology, vol. 26, no. 10, **(2014)**, pp.35-37.
[4] F.A. Murzin, N.V. Poplevina and D.F. Semich, "Algorithms and software for detecting oil reservoirs from nuclear logging data", Optoelectronics, Instrumentation and Data Processing, vol. 47, no.4, **(2011)**, pp. 395-405.
[5] H.M. Firoozabadi, K. Rahimzade, P. Pourafshary and M. Edalat, "Analysis of Production Logging Data to Develop a Model to Predict Pressure Drop in Perforated Gas Condensate Wells", Petroleum Science and Technology, vol. 29, no. 16, **(2011)**, pp.1722-1732.
[6] M. Kim, D.W. Lee, K. Kim and J.H. Kim, "Hierarchical structured data logging system for effective lifelog management in ubiquitous environment", Multimedia Tools and Applications, vol. 74, no. 10, **(2013)**, pp. 3561-3577.
[7] M. Maisch, B. Bertsche and R. Hettich, "An approach to online reliability evaluation and prediction of mechanical transmission components", vol. 3, no. 2, **(2006)**, pp.207-214.
[8] P. Nipesh and T. Jochen, "Automatic spatio-temporal analysis of construction site equipment operations using GPS data", Automation in Construction, vol. 29, **(2012)**, pp.107-122.
[9] K.I. Hwang, "Data aggregation method guaranteeing minimum traffic in multi-hop automatic meter reading networks", J-KICS, vol. 36, no. 7, **(2011)**, pp. 848-857.
[10] N. Birla and A. Swarup, "Performance of Preview Control based on Evolutionary Algorithms", International Journal of Advanced Science and Technology, vol. 38, **(2012)**, pp.37-52.
[11] J. Cao, L. Yin and H. Zhao, "Design and Development of Embedded Multimedia Terminal", Distributed Computing and Applications to Business Engineering and Science (DCABES), Ninth International Symposium, **(2010)**, pp. 289-292.

Author

Hyoung-Keun Park, He received the M. S. and Ph. D. degrees in electronic engineering from Wonkwang University, Iksan, Korea in 1995 and 2000, respectively. He is currently a professor department of the electronic engineering at Namseoul University, Chungnam, Korea, in 2005. His research interests are in embedded system, applied ubiquitous sensor network and LKAS.

