

SARKER, M.M.K., PARK, D., HAM, W., TUMENJARGAL, E. and LEE, J. 2012. Embedded workbench application of GPS sensor for agricultural tractor. In *Arabnia, H.R., Deligiannidis, L. and Solo, A.M.G. (eds) Proceedings of the 2012 Embedded systems and applications international conference (ESA'12), 25-28 July 2012, Las Vegas, USA*. Athens, GA: CSREA Press. Hosted by WorldComp proceedings [online], pages 40-45. Available from: <http://worldcomp-proceedings.com/proc/proc2012/esa/contents.pdf>

Embedded workbench application of GPS sensor for agricultural tractor.

SARKER, M.M.K., PARK, D., HAM, W., TUMENJARGAL, E. and LEE, J.

2012

© 2012 CSREA Press.

Embedded Workbench Application of GPS Sensor for Agricultural Tractor

Md. Mostafa Kamal Sarker¹, DongSun Park², Woonchul Ham³, Enkhbaatar Tumenjargal³ and JaeHwan Lee³

^{1&3} Division of Electronic Engineering, Chonbuk National University, Jeonju-si, Jeonbuk, Republic of Korea

² IT Convergence Research Center, Chonbuk National University, Jeonju-si, Jeonbuk, Republic of Korea

Abstract - This paper presents a design of an embedded workbench application of Global Positioning System (GPS) for agricultural tractor. Electronic Control Unit (ECU) is Global Positioning System (GPS) sensor using IAR (IAR Embedded Workbench) and an open source library which follows the most important characteristics of International Organization for Standardization (ISO) 11783 communication protocol in the serial communication network of agricultural vehicles. These applications are written in C/C++ programming methods. We explain some test connection configuration between working Personal Computer (PC) and test board for studying the application program and GPS sensor working status. This research work mainly describes the system architecture and programming methodology of an application program which follows some standards for agricultural machinery.

Keywords: Electronic control unit, isobus, controller area network, open source library, embedded workbench

1 Introduction

Since the past few years, manufacturers of agricultural machineries have increasingly turned to electronics to provide products with enhanced functionality, productivity, and performance to clients. Electronic content in agricultural equipment has increased. A natural outcome of adding electronic components to agricultural equipment has been realization of the advantages of allowing the components to communicate. A GPS sensor on a tractor, for example, may communicate with a virtual terminal [1] (receiving the CAN message continuously and send it to Virtual terminal through CAN-bus). Developing the electronic control systems, a lot of ECUs interconnected inside agriculture tractor [2]. Such as ECU Data Source, ECU Display, ECU GPS Sensor, ECU Tractor Bridge, etc. All ECUs connected with CAN-bus (Controller Area Network or CAN-bus is an ISO standard computer network protocol and bus standard, designed for microcontrollers and devices to communicate with each other without a host computer) and exchanging data between control units take place on a uniform platform. This platform is called a protocol. The CAN bus acts as a so-called data highway.

This research illustrates the design of an application program for agricultural tractor GPS sensor. It also gives some idea about tractor software design. The principle idea of this application is developing software for tractor ECUs. On the other hand, open source library provides the main resources for this research work with following some standards. Using C/C++ programming methods for the application program and the software environment is embedded workbench.

In our application design, we chose our test board is STM32F107 ARM 32-bit Cortex-M3 board for ECU hardware of GPS sensor [3]. We also use RealSYS CANPro USB device for analyzing CAN messages received by GPS sensor and AMTEL mini JLINK is an optimizing C/C++ compiler for ARM Cortex-M3 microcontroller. We select the embedded workbench "IAR Embedded Workbench" and the open source programming library "ISOAgLib" [4] for developing our application program. This paper is organized as follows: In section 2, 3, 4 and 5, we have described an overview of standards, test environment, embedded workbench applications, workbench results and discussion, respectively. Finally, Conclusions are presented in section 6.

2 An overview of standards

2.1 ISO 11783 communication protocol

The ISO 11783 is a new standard for electronic communications protocol for tractors and machinery in agriculture and forestry. This ISO 11783 standard is sometimes called as ISOBUS [5]. The network has messages defined to allow communications between any of the components, like communication between the Task Controller and the GPS ECU. Navigational messages are defined and allow positional information to be received by the Task Controller. The task controller can then deliver the prescription to an implement as needed based on position measured by an onboard GPS system. It consists of several parts: general standard for mobile data communication, physical layer, data link layer, network layer, network management, virtual terminal, implement messages applications layer, power train messages, tractor ECU, task controller and management information system data

interchange, mobile data element dictionary, diagnostic and file server. The structure of electronic data communication according to ISO 11783 is based on the Open system interconnect (OSI) model layers, however, the higher functional layers sometimes defined differently. Figure 1 schematically illustrates the layer stricter ISO 11783 standard.

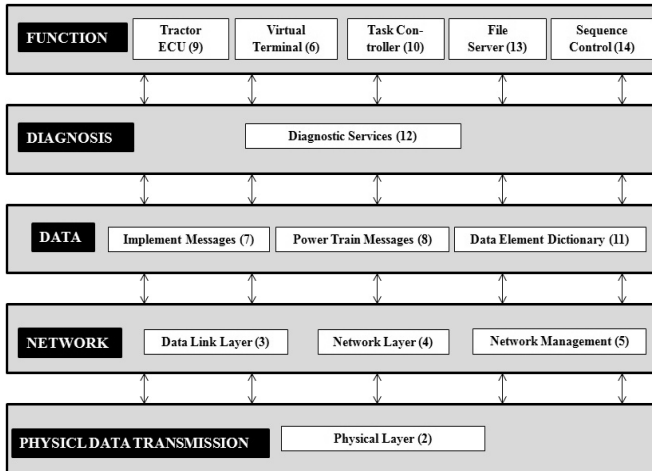


Figure 1. Diagram of the ISO 11783 standard parts (own illustration)

The purpose of ISO 11783 is to provide an open, interconnected system for on-board electronic systems. It is intended to enable electronic control units (ECUs) to communicate with each other, providing a standardized system. The tractor ECU shall have at least one node for connection to the implement bus.

2.2 CAN networks

ISO 11783 standardizes a multiplex wiring system as described above, based on the Controller Area Network (CAN) protocol developed by Bosch (Bosch, 1991)[6]. This protocol uses a prioritized arbitration process to allow messages access to the bus. When two messages are sent at the same time, their identifiers are imposed bit-serially onto the bus. The bus must be designed to allow dominant bits to overwhelm recessive bits when both are applied simultaneously by different ECUs on the bus. No conflict occurs as long as the ECUs are sending the same bits, but when one sends a recessive bit while the other sends a dominant bit, the bus state is dominant. The ECU sending the recessive bit must sense the bus is at a dominant state when the bit was sent and must cease transmitting the message at that time and retry the next time the bus becomes idle. This strategy allows more dominant identifiers, those with a lower value, to have a higher priority on the bus. To allow this feature to work properly, CAN synchronizes messages at the beginning of each transmission to assure bits are aligned. The result is that ISO 11783 provides a communication system where ECUs share a communications link, and messages at any point in time are allowed access to the bus based on their priority.

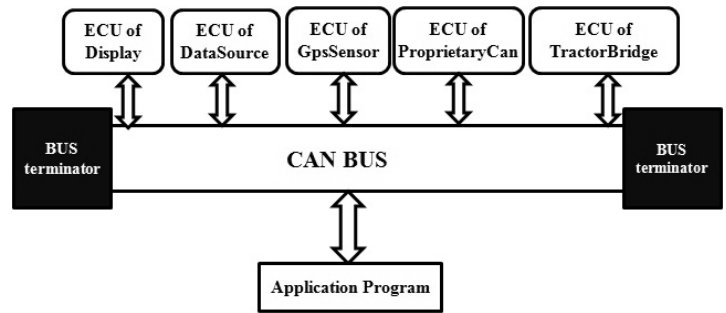


Figure 2. Structure of CAN-bus

2.3 CAN Message Structure

The implementation of the CAN message for tractors and machinery for agriculture is based on CAN Version 2.0B [7]. This describes a 29-bit identifier and a data rate of 250 kbit/s.

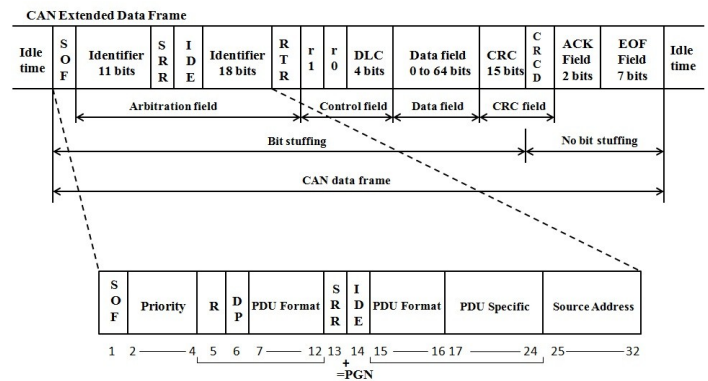


Figure 3. Message frame format of CAN Data (CAN 2.0 B Extended Frame Format)

The composition of the 29-bit identifier is shown in Figure 3. The Start Of Frame (SOF) bit 1, the Substitute Remote Request (SRR) bit 13 and the ID (identifier) Identifier Extension (IDE) bit 14 is not considered for the identifier length.

2.4 Navigation system messages

The set of navigational messages defined in ISO 11783-7 [8] is provided by the installation of a global positioning system (GPS) or differential global positioning system (DGPS) receiver on the tractor. A special classification, “N”, shall be appended to the class number when the tractor is able to provide navigational information on the implement bus. For example, a class 3 tractor implement interfaces is able to support navigational messages can be classified as class 3N, and supports the following parameters: navigational system high output position; navigational system position data; navigational pseudorange noise statistics. The navigation location parameters specified in IEC 61162-3 (NMEA 2000[9]).The configuration of a tractor–implement connection and the offset to and from the tractor implement reference

points, are used in the navigational parameters and in the implement configuration of process data messages.

3 Test environment

The task-controller applications layer, which defines the requirements and services needed for communicating between the task controller and electronic control units [10]. Task controller is used to issue instructions to different equipment to complete some task and management computer interface is used for data exchange between task controller and external management computer. Communication is realized between different equipment in the bus network by way of the sending of messages, and its typical application is as follows: task controller in real time receives information of navigation and location generated by GPS, the ECU of the engine provides its current torque curve for transmission gearbox, and so on. The ECU of the tractor functions as a filter for message transport between the tractor bus and the equipment bus, which can avoid the event that the communication task of one bus is so heavy that the other bus is overloaded.

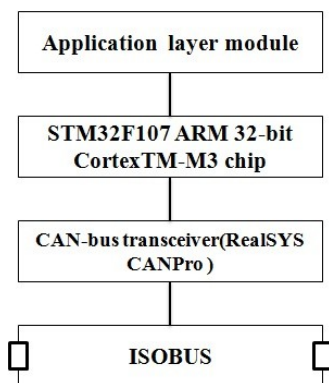


Figure 4. Network structure of test GPS sensor

Figure 4 show the network structure of test GPS sensor based on STM32F107 ARM Cortex-M3 board. The main board STM32F107 adopts the ARM 32-bit Cortex-M3 SCM (Single Chip Microcomputer) produced by STMicroelectronics company of French-Italy. It is a totally integrated mixed-signal system-on-chip, which integrates in one chip almost all the analog peripherals, digital peripherals and other functional components that are necessary to form a data sampling or control system of a SCM. BOTSH CAN controller is compatible with CAN technical specification 2.0A and 2.0B is integrated in STM32F107 and also 2.4 inch TFT LCD Panel (320*240) with touch screen. It is composed of CAN kernel with 256KB Flash and 64KB RAM internal memory, message processing unit and register. CAN controller has 32 message destinations which can be used to send or transmit data. Received data, message destinations and identification code are storage in Message RAM.

All the protocol functions (such as data transmission and receipt of filter) are performed by CAN controller. Through

the special register in the main control chip, CAN controller can be configured to visit received data and transmitted data. In this way, CAN communication can be realized by use of less bandwidth of CPU. STM32F107 can perform all the functions of the data link layer and application layer of ISOBUS protocol. Figure 5 shows test connection configuration between working PC and test board for checking the ECU of GPS sensor working status.

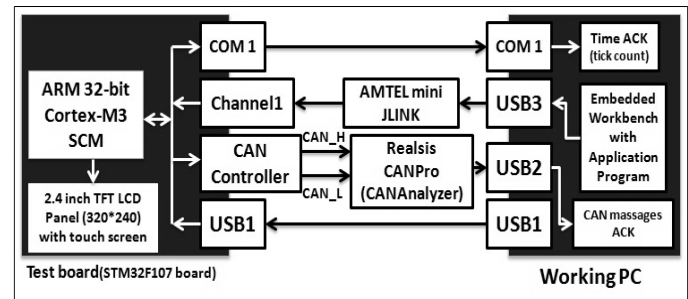


Figure 5. Connection configuration between working PC and test board

In this figure, test board COM1 port (i.e.name of serial port hardware for input and output) is linked with PC COM1 port for sending time acknowledgement (ACK) of GPS messages (CAN message) to PC. In this relationship scheme, we use AMTEL mini JLINK (USB driven JTAG interface for ARM cores including mini USB cable) is an optimizing C/C++ compiler (i.e. download and debug the application program) for ARM Cortex-M3 microcontroller and attach between test board Channel1 and PC USB3 port (Universal Serial Bus). This connection is main platform of our application program development. Because of this connection download application program from PC to the microcontroller for debugging and make sure the ECU (test board) becomes a GPS sensor. CAN controller has some ports but for our test purpose we use only two ports for CAN_L and CAN_H [11] and connect with CAN analyzer. Here, we also make a connection between test PC and CAN analyzer (i.e. Real SYS CAN Pro USB device for hardware) by USB2 port through USB cable. After establishing all the connections, we can verify the GPS message status by CAN Pro Analyzer v1.0 software in PC.USB1 port is for test board power supply through the USB cable and LCD display is only showing some information about GPS manufacturer.

4 Embedded workbench application

4.1 System architecture

For the application program of GPS sensor, we use an open source programming library named ISOAgLib. The IsoAgLib is a C++ library in development of ISO 11783 standard applications in an Object Oriented way to serve as a software layer between application specific program and communication protocol details. The author of IsoAgLib library, Dipl. - Inform. Achim Spangler, licensed with exceptions under the terms of the GNU General Public

License (GPL). By providing simple function calls for jobs like starting a measuring program for a process data value on a remote ECU, the main program has not to deal with single CAN message formatting. This way communication problem between ECU's which use this library should be prevented. The IsoAgLib has a modular design pursuant to the various functional components of the standard ISO 11783. The library has this design to make sure the minimum use of IsoAgLib in program memory of Implement ECU. The IsoAgLib demonstrates the layered architecture to be easily familiar with new hardware platforms. Most of the software can be used without alteration on all platforms. The layered architecture is described by the diagram in Figure 6.

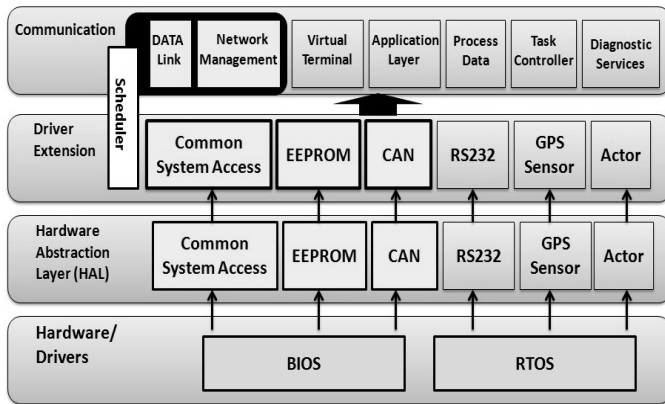


Figure 6. System architecture of embedded workbench applications

The IsoAgLib was developed to be suitable with different systems, and these systems can be an element of processor, memory, Human Machine Interface (HMI) and interface with the CAN bus. Therefore, the IsoAgLib is divided into two sections: the library itself and HAL. The HAL is responsible for communicating with the operating system (OS) or BIOS device that is running the application, as can be seen in Figure 7. We implement CAN-bus in real-time operating system. The application program initialized CAN controller and accessing CAN-bus.

4.2 Programming methodology

For executing our GPS application program, we should build some configuration of development board (STM32F107) into the IAR embedded C/C++ programming interface. We created all configurations by using ARM C/C++ [12] and "ISOAgLib" libraries and our self what we needed.

Firstly, initializing all peripherals of our test board (STM32F107) [code (main.cpp): void Init_All_Periph (void) {RCC_Configuration (); InitDis (); GPIO_configuration (); NVIC_configuration (); ;}. Here, RCC (Reset and Clock Control) configuration [RCC_Configuration()] is creating system clock configuration for all peripherals, initializing display [InitDis()] is LCD display configuration, GPIO (General-Purpose function of Input and Outputs) configuration [GPIO_Configuration()] is creating structure of

every pin (i.e. CAN pin: RX,TX) for our development board and setting their mode, NVIC(Nested vectored interrupt controller) configuration [NVIC_Configuration()] is enables low latency interrupt processing and efficient processing of late arriving interrupts. The bxCAN (Basic Extended CAN) [13] module handles the transmission and the reception of CAN messages fully autonomously. Standard identifiers (11-bit) and extended identifiers (29-bit) are fully supported by hardware. Secondly, SysTick timer (STK) configuration [SysTick_Conf ()] is setup SysTick Timer for interrupts. CAN interrupt [CAN_Interrupt ()] is interrupt mode for CAN. The processor has a 24-bit system timer SysTick which counts down from the reload value to zero, reloads (wraps to) the value in the load register on the next clock edge, then counts down on subsequent clocks. The bxCAN interrupts has four interrupt vectors dedicated. Each interrupt source can be independently enabled or disabled by means of the CAN Interrupt Enable Register CAN_IER). Thirdly, Universal synchronous and asynchronous receiver transmitter [USART1_Configuration ()] configuration is the configuration of the CAN bit timing. According to the CAN specification [14], the bit time is divided into four segments (see Figure 7). The synchronization segment, the propagation time segment, the phase buffer segment 1, and the phase buffer segment 2. Each segment consists of a specific, programmable number of time quanta (see Table 1). The length of the time quantum (t_q), which is the basic time unit of the bit time, is defined by the CAN controller's system clock f_{sys} and the Baud Rate Prescaler (BRP) : $t_q = BRP / f_{sys}$. Typical system clocks are: $f_{sys} = f_{osc}$ or $f_{sys} = f_{osc}/2$.

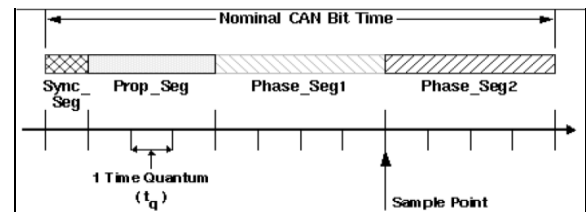


Figure 7. CAN bit timing

Parameter	Range	Remark
BRP	[1 .. 32]	defines the length of the time quantum t_q
Sync_Seg	$1 t_q$	fixed length, synchronization of bus input to system clock
Prop_Seg	$[1 .. 8] t_q$	compensates for the physical delay times
Phase_Seg1	$[1 .. 8] t_q$	may be lengthened temporarily by synchronization
Phase_Seg2	$[1 .. 8] t_q$	may be shortened temporarily by synchronization
SJW	$[1 .. 4] t_q$	may not be longer than either Phase Buffer Segment

This table describes the minimum programmable ranges required by the CAN protocol

Table 1: CAN Bit Timing Parameter

Finally, make a loop for frequently CAN message received by our GPS sensor within a fixed time period. In additionally, we also create our device driver and startup (STM32 driver and startup) configuration. After complete all steps, we can build and execute our application program completely. Figure 8 shows the application program of GPS sensor in IAR Embedded Workbench.

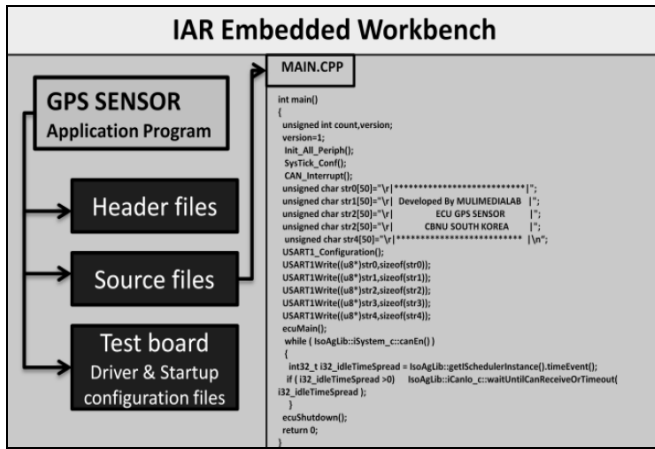


Figure 8. Application program of GPS sensor in IAR Embedded Workbench

5 Workbench Results and Discussion

The main task of this work is developing the test board as an ECU for agricultural tractor GPS sensor. With following the programming methodology, we can build our application program. In IAR Embedded Workbench, the program should be downloaded to ARM Cortex-M3 microcontroller by AMTEL mini JLINK for debugging. When debugging is completed then run the program. After finishing all, the test board is performing as an ECU of GPS sensor. Now, the CAN messages are frequently received by the test GPS sensor. We can easily analysis those messages with standards by CANPro Analyzer v1.0. Figure 9 shows the CANPro Analyzer window define CAN message received by our test GPS sensor.

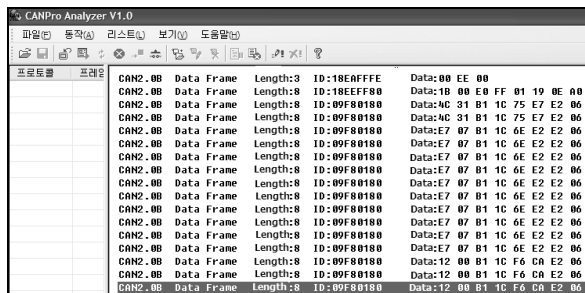


Figure 9. CAN message received by GPS sensor

The output window of CAN Pro Analyzer, we get the first message data frame in 3 bytes data length and, ID (Identifier Bit) is 18-EA-FF-FE16(hexadecimal) means that this data have (5bit-8bit-8bit-8bit)2 CAN ID and first message define by request for address claimed or request PG is 00EE0016 means that first data PGN is 6092810[15]. Second message data frame is 8 bytes data length; ID is same as first message data frame. Second message define by address claimed and data is NAME which has some fields. We can explain all messages that classify by hexadecimal numbers with the help of ISO 11783 standards. Figure 10 explain only two messages with some standards. So our result shows that CAN messages follow the standards perfectly without error.

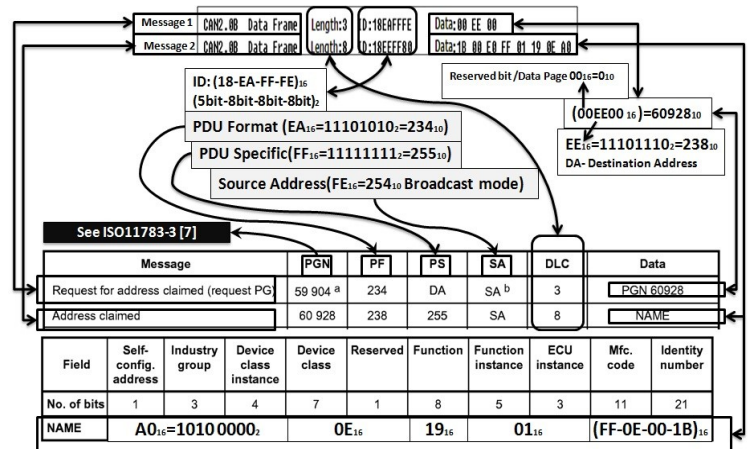


Figure 10. Analysis CAN messages(First and second)

Now we can clarify all messages with standards which are received by our GPS sensor.

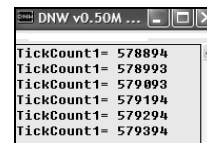


Figure 11. Time difference between CAN messages

Figure 11 shows the time difference between two CAN messages received by GPS sensor is 100milisec. This mean the events of CAN interrupt and System timer is working perfectly (i.e. when events are changed it takes 1milisec). So we can get CAN messages continuously with standard time. Therefore, we have no error in our application program and our developed GPS show's great performance.

6 Conclusions

Recently, a great amount of development has happened in the field of agriculture by using information technology over the world. Most important part has developed by German, European and some of American researchers. Now in Asia, Korea has been started developing their own agricultural field by using recent information technology and for this purpose our research team initially doing some important research work on this sector, like developed application program for agricultural tractor electronic control units (ECUs) and virtual terminal, etc. All application procedures are followed by ISO 11783 and some other standards. For the development of our application program for agricultural tractor GPS sensor, we use an open source library with object oriented way. In our research result, we found that our GPS sensor can receive CAN messages frequently with expected time. So it works perfectly without any fault. In our future work, we are going to compose application program for every ECU of an agricultural tractor (ECU Data Source, ECU Display, ECU GPS Sensor, ECU Tractor Bridge, etc.) and developed the virtual terminal.

7 Acknowledgment

This work was supported in part by Brain Korea-21 and the Center for IT Convergence Agricultural Machinery (ITAM) grant (NO. R09-6)* funded by the Ministry of Knowledge Economy, Republic of Korea.

8 References

- [1] ISO11783-6: Tractors and machinery for agriculture and forestry -Serial control and communications data network-Virtual Terminal. 2002.
- [2] ISO11783-9: Tractors and machinery for agriculture and forestry -Serial control and communications data network-Tractor ECU. 2002.
- [3] The STMicroelectronics: [Online Available: <http://www.st.com/>], 1998-2012.
- [4] IsoAgLib: "Development of ISO 11783 applications in an Object Oriented way", [Online Available: <http://isoaglib.org/>], 2009.
- [5] Peter Felimeth, "CAN-based tractor- agricultural implement communication ISO 11783," CAN Newsletter, 2003, 9.
- [6] Bosch, Robert, GmbH. CAN Specification, Version 2.0., Germany, 1991.
- [7] ISO11783-3: Tractors and machinery for agriculture and forestry -Serial control and communications data network-Data link layer. 1998.
- [8] ISO11783-7: Tractors and machinery for agriculture and forestry -Serial control and communications data network-Implement messages application layer, 2002.
- [9] NMEA 2000: The National Marine Electronic Association's NMEA 2000® Standard for Serial Data Networking of Marine Electronic Devices has been approved by the International Electrotechnical Commission (IEC), 2008
- [10] ISO11783-10: Tractors and machinery for agriculture and forestry -Serial control and communications data network- Task controller and management information system data interchange. 2009.
- [11] ISO11783-2: Tractors and machinery for agriculture and forestry -Serial control and communications data network-Physical Layer. 2002.
- [12] PM0056: STM32F107 Cortex-M3 programming manual, STMicroelectronics, 1998-2012.
- [13] RM0008: STM32F107 Reference manual for advanced ARM-based 32-bit MCUs, STMicroelectronics, 1998-2012.
- [14] Florian Hartwich, Armin Bassemir, Robert Bosch GmbH "The Configuration of the CAN Bit Timing," 6th International CAN Conference, 1999; Turin (Italy).
- [15] ISO11783-5: Tractors and machinery for agriculture and forestry -Serial control and communications data network-Network management. 2002.