IMPLEMENTATION OF ECU FOR AGRICULTURAL MACHINES BASED ON ISOAGLIB OPEN SOURCE

W.Ham, T.Enkhbaatar, B.Luubaatar and K.Hyeokjae

Division of Electronic Engineering
Chonbuk National University
Jeonju, Jeonbuk, South Korea

ABSTRACT

In this paper work, we consider implementation of electronic control unit (ECU) for agricultural machineries. Software implementation is based on IsoAgLib library developed by OSB&IT Engineering Company. We modify IsoAgLib and upgrade it for our target system. The IsoAgLib is an object oriented C++ library that has the communication services and management systems according to the ISO 11783 standard. This library allows building ISOBUS compatible equipment without the protocols implementation contained in this standard. We focus on developing ECU of GPS sensor, ECU of light control and ECU of sprayer. Hardware implementations of ECUs are developed by using STM32F107 ARM 32-bit Cortex™-M3 CPU and CAN-bus receiver/ transceiver driver chip. Time managing of the system is implemented using time stamp messages between ECUs and virtual terminal.

Keywords: CAN-bus, ECU, Embedded System, ISO11783

INTRODUCTION

This technical article represents for agriculture machinery control system based on ISO11783 standard. The related standard for details of ECUs which is implemented is described in the parts 3, 4, 5, 6 and 10 of ISO11783.

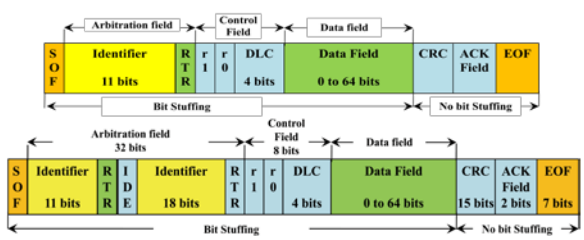
The ISO11783 standard has been jointly developed by tractor and implement manufacturer including AGCO, AGROCOM, DICKEY-Jonh, John Deere, Siemens and Fendt. These manufacturers have also created a specification defining how this standard should be interpreted. This specification is commonly known as ISOBUS. Also, several research teams work on the implementation of ECUs and ISO11783 standard. In domestic, we cooperate with Korea agriculture company in the research and application of ECUs. The purpose of this technical article is to implement hardware design and software design of ECUs based on the ISO11783 standard.

In this paper we introduce implementation of GPS ECU and light control ECU. The design and implementation of reliable devices highly depends on the software process consistency. Since we have a IsoAgLib library as a base library we modified it for our hardware by implementing hardware related functions, buffering sections, connections between network layer and application layer, and timing process such as local and global timing. Though, GPS ECU sends periodically pseudo location value to the CAN bus it processes main functions such as receiving sending and replying to the PGNs from the CAN bus. The light control of ECU control several lights in the tractor and responds with value of status. The remainder of this paper is organized as follows: Section 2 overview of CAN, ISO11783 standard and IsoAgLib library; Section 3 implementation of hardware; Section 4 describes the implementation of software; Section 5 experiment and results; and finally Section 6 discusses future work and conclusions.

BACKGROUNDS AND METHODS

Control Area Network (CAN)

CAN network is that each message is preceded with an identifier that is unique to the transmitting controller and that multiple controllers can communicate over a single two-wire bus. CAN transmitting data in frames containing a header and 0 to 8 bytes of data. There are three separate CAN standards: CAN version 1.0, Version 2.0A (Standard CAN), and version 2.0B (Extended CAN). The main difference in these standards is the length of the identifiers that precede each message. The original specifications (Version 1.0 and 2.0A) specify an 11 bit message identifier. The Version 2.0B Extended Frames contain a 29-bit identifier which allow over 2^{29}-1 message identifiers. The 29-bit identifier is made up of the 11-bit identifier ("Base ID") and the 18-bit Extended Identifier ("ID Extension"). The figures 1 shown in difference of three CAN standards. In our research work presented in this paper based on the CAN2.0B standard.



(a).CAN version 1.0 and version 2.0A (b). CAN version 2.0B

Figure 1. Frame structure of CAN standards

ISO11783 overviews

The ISO11783 standard is sometimes called as ISOBUS. It consists of several parts, which is comprised of fourteen documents shown in Table 1. The ISO11783 as a whole specifies a serial data network for control and communications on forestry or agricultural tractors and mounted, semi-mounted, towed or self-propelled implements. Its propose is to standardize the method and format of transfer of data between sensors, actuators, control elements, tractor or implement and etc.

Table 1. ISO11783 standard documents

Document	Description	Form
ISO 11783-1	General Standard	Complete
ISO 11783-2	Physical Layer	Complete
ISO 11783-3	Data Link Layer	Complete
ISO 11783-4	Network Layer	Complete
ISO 11783-5	Network Management	Complete
ISO 11783-6	Virtual Terminal	Complete
ISO 11783-7	Implement Message Layer	Complete
ISO 11783-8	Power Train Messages	Complete
ISO 11783-9	Tractor ECU	Complete
ISO 11783-10	Task Controller	Complete
ISO 11783-11	Mobile data element dictionary	Complete
ISO 11783-12	Diagnostics services	Complete
ISO 11783-13	File Server	Complete
ISO 11783-14	Sequence Control	Draft

ISOAgLib open source library

In this section we introduce the ISOAgLib open source programming library, which is developed by Munich University and OSB & IT engineering company in Germany. It is suitable for embedded communication software in electronic control units (ECU) such as virtual terminal (VT) or task controller (TC). All functions according to the ISO11783 standards as well as the established machine interfaces are already implemented in the library. The figure 2 describes in the system architecture of ISOAgLib programming library.

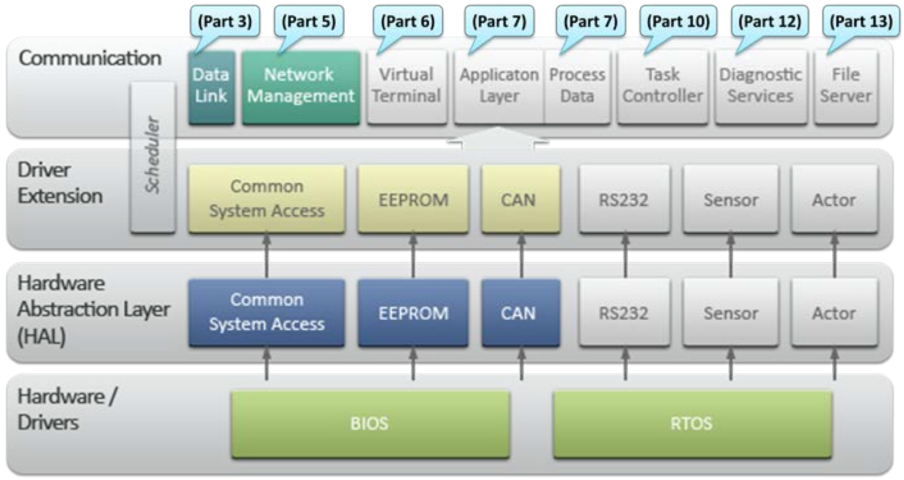


Figure 2. System architecture of ISOAgLib open library.

The IsoAglib library consists of several parts: Communication, Scheduler, Driver Extension, Hardware Abstraction Layer and Hardware/driver. They develop so many versions of IsoAglib programming library, some tutorials and examples for real implement devices. In our case we are studying version 2.2-rc5 of IsoAgLib, why this version has some tutorials and examples. Also those example and tutorial files are compiling no errors, and then we consider that version. Especially we studied CAN server application program, ECU of GPS sensor, ECU of Display, ECU of DataSource, ECU of TractorBridge, and etc. The figure 3 illustrates files structures of ISOAgLib open library. In our case we are focusing on real hardware implementation in embedded Windows CE environment.

IMPLEMENTATION OF CAN-BUS INTERFACE

The hardware of other ECUs implemented by the 32-bit microcontroller STM32f107VC development board. The STM32F microcontroller’s main Central Processing Unit (CPU) architecture is Advanced RISC Machine (ARM) V7 Cortex-M3, its support several useful peripherals. The one of useful peripheral is CAN controller, its supports CAN 2.0A and 2.0B active and passive with data rates up to the maximum 1Mbit/s. The CAN controller also has extensions to support fully deterministic communication defined under the time-triggered CAN (TTCAN) protocol. When enabled, the TTCAN extensions support automatic message retransmission and will place a message timestamp in the last two data bytes of the CAN message packet. The next most important feature of a CAN controller is it’s receives message filtering. Because CAN is a broadcast network, every message transmitted is received by every node on the network. In a CAN network of any reasonable complexity there will be a large number of messages sent over the CAN bus. In such a network the CPU of a CAN node will spend all its runtime responding to CAN messages. To avoid this problem all CAN controllers have some form of message filtering that blocks unwanted messages from reaching the receive buffers. The structure of software implementation for ECU of light control is shown in figure 5.

IMPLEMENTATION OF SOFTWARE

In the firmware level programming CPU follows only one process in the hardware: it means not working with any operating systems. If the hardware system not working with any operating system this time programming code should be write in the firmware level.

The most embedded system developer use standard C programming language, then sometimes limitation in complicated system. ISO11783 standard implementation is not simple project for programmers. The IsoAglib library's source codes written by the object oriented programming language C++, its can easy solve some complicate protocols and etc. Because of programming C++ can support Standard Template Library (STD), it can make templates and namespace for lists, queue, vectors, and etc. And another important thing is development tools for embedded system. The most embedded development tool's compiler and linker proposed for C and assembler codes. The one of famous tools is IAR Embedded Workbench, which was supported by C/C++ and assembler for ARM7/ARM11 core processors. Then we successfully implement the ECU of GPS and ECU of Sprayer codes based on IsoAglib library in the IAR Embedded Workbench environment. IAR Embedded Workbench tool can working with J-Link universal debugger device. The sample hardware debugging process is shown in the figure 3. As an example implementation of the block diagram of light control ECU is shown in figure 4.

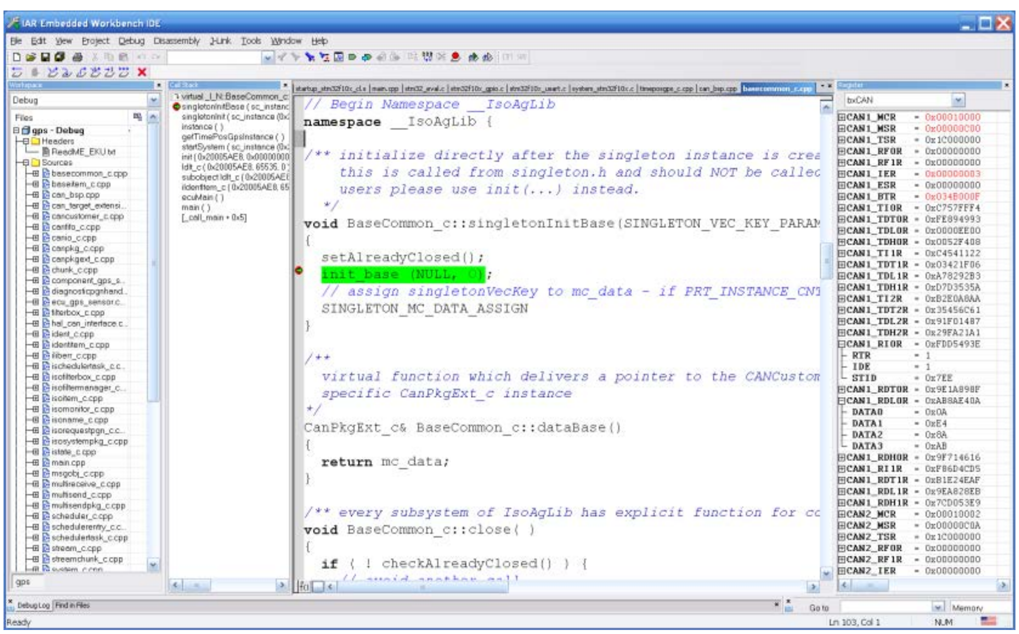


Figure 3. Object oriented C++ codes the ECU of GPS implementation by using the IAR Embedded Workbench.

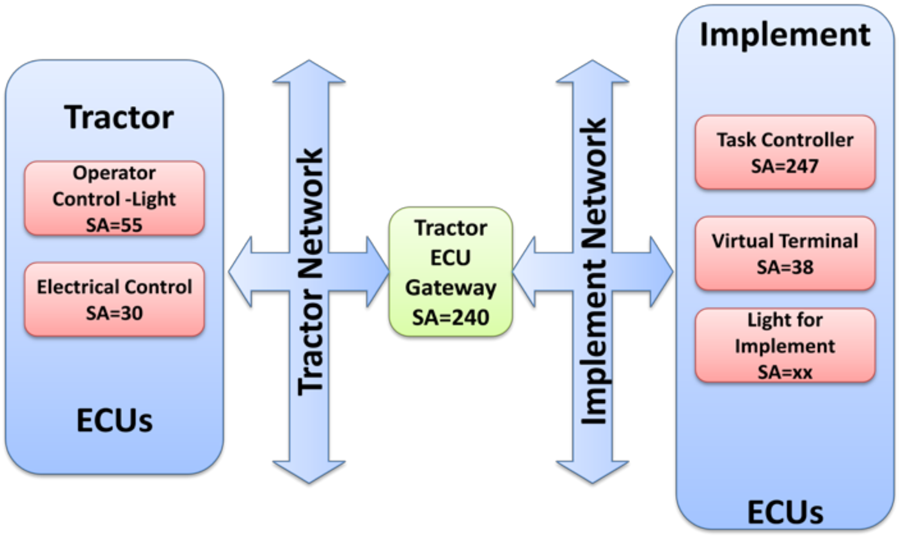


Figure 4. The block diagram of ECU of light control.

EXPERIMENTS AND RESULTS

In order to test the implemented ECUs, experimental environment has set virtually which has virtual ISO CAN bus on the Ethernet networking. CAN server is set in the WinCE6.0 operation system environment on the ARM11 based hardware system to make a connection between hardware implemented real ECUs and software implemented virtual ECUs. The experimental system structure is shown in figure 6. Since IsoAgLib has several virtual ECUs, such as CAN-logolizer, CAN-messenger, WriteISO and ReadISO etc, we use them for implementation of our ECUs. For now, the GPS ECU and Light Control ECU are work quite well and we can see the processing of those ECUs in the virtual experimental environment. Figure 6 shows timing of a CAN diagram. The processing between ECUs and virtual terminal on the ISO Can bus with PGNs, such as a control and status messages, is shown in table 2.

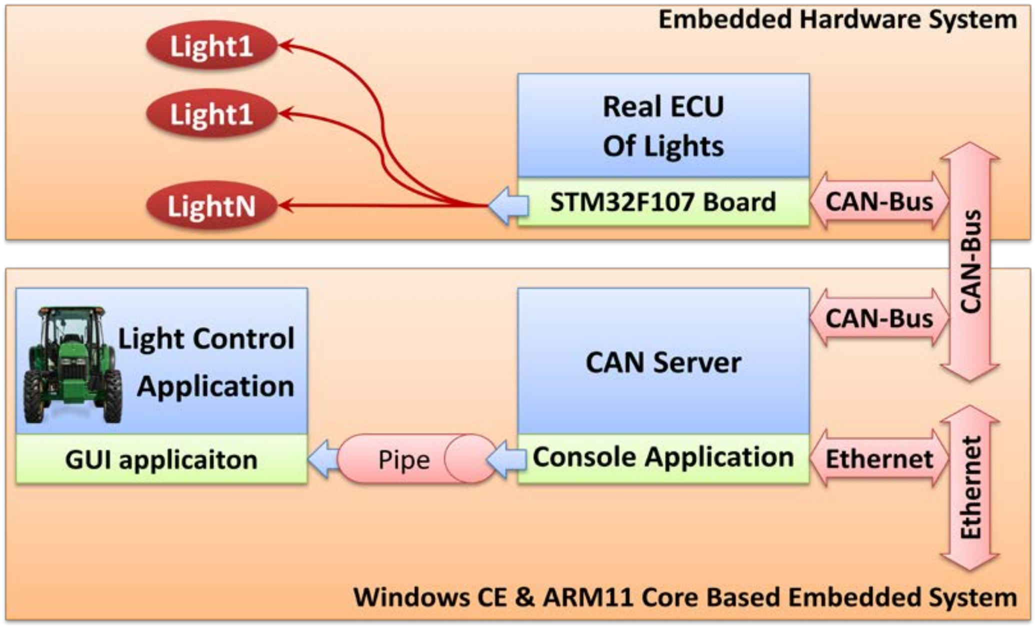


Figure 5. The structure of software implementation for ECU of Light Control.

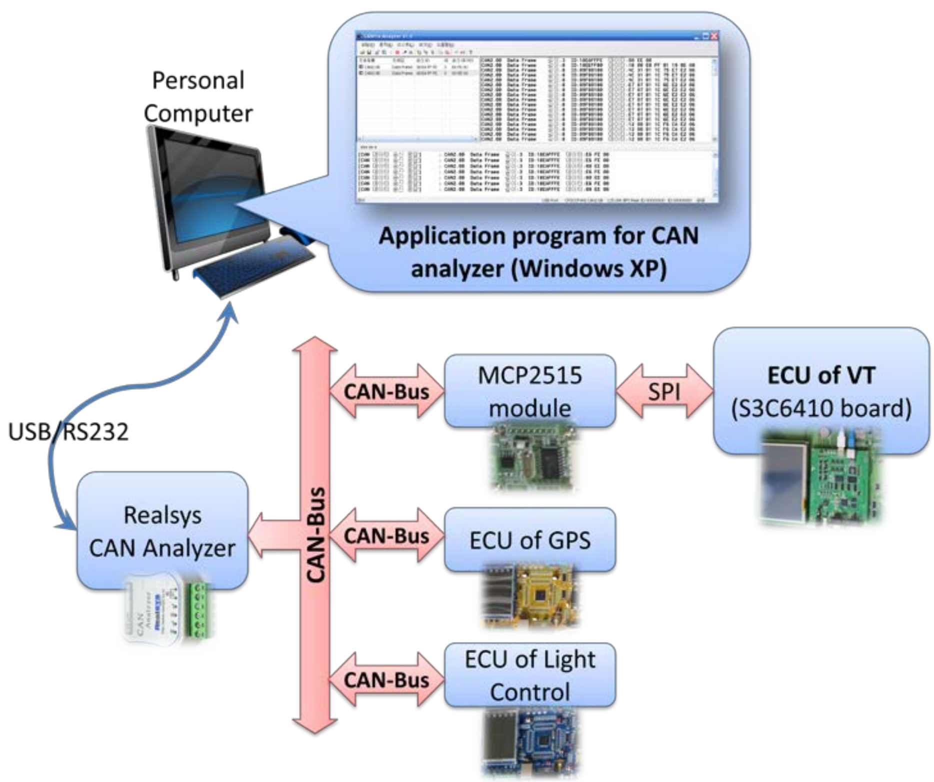


Figure 6. The experimental block diagram of ECU of Light Control and ECU of GPS.

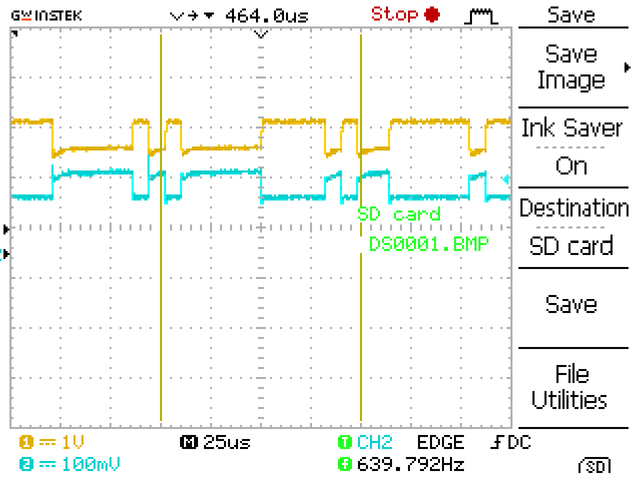


Figure 6. Timing diagram of CAN-Bus.

Table 2. Sample messages of ECU GPS, ECU of Light Control and VT.

Message	**PGN**			**SA**	**DA**	**DLC**	**Data**							
VT Address Claimed	00	EE	00	26	All	8	02	00	A0	E8	00	1D	02	A0
GPS Address Claimed	00	EE	00	1C	All	8	25	B3	FF	E8	00	17	00	A0
Light Control Address Claimed	00	EE	00	80	All	8	2F	B3	FF	E8	00	84	0C	A0
Request	00	EA	00	26	All	3	00	EE	00					
VT Address Claimed	00	EE	00	26	All	8	02	00	A0	E8	00	1D	02	A0
GPS Address Claimed	00	EE	00	1C	All	8	25	B3	FF	E8	00	17	00	A0
Light Control Address Claimed	00	EE	00	80	All	8	2F	B3	FF	E8	00	84	0C	A0
VT Language Code	00	FE	0F	26	--	8	64	65	40	00	00	00	FF	FF
Light Control Command	00	FE	0D	80	--	8	01	FF	FF	FF	FF	FF	FF	FF
Light Control to VT	00	E7	00	80	26	8	C0	FF	00	00	00	00	FF	FF
VT to ECU	00	E6	00	26	All	8	FE	80	64	00	A0	0F	00	FF
GPS Position Data	01	F8	05	1C	26	8	86	00	00	FF	FF	FF	FF	FF
...														

DISCUSSION AND 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 the team of Munch University and OSB&IT Engineering company. 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 our development work is followed by ISO 11783 standards. For developing program of ECU of GPS, light control of ECU we use an IsoAglib open source library. In our research result, we develop own program for ECU of GPS and ECU of light control, there can receive CAN massages frequently with expected time. In our future work, we are going to develop application program for any proposed ECU of an agricultural tractor (for example ECU of Sprayer, ECU of Data Source, ECU Auxilirary Sensor, ECU Tractor Bridge, etc.) and developed our virtual terminal.

ACKNOWLEDGEMENT

This work was supported by the Post BK21 Project of Chonbuk National University and the Center for IT Convergence Agricultural Machinery (ITAM) grant (NO. R09-6)* funded by the Ministry of Knowledge Economy,Republic of Korea.

REFERENCES

Spangler, A. and Wodok, M. 2010. IsoAgLib – Development of ISO 11783 Applications in an Object Oriented way. In: *http://www.isoaglib.com*.

ISO. 1998. Part 3: Data link layer. ISO 11783. *International Standard*, First edition 1998-07-01.

ISO. 2001. Part 5: Network management. ISO 11783. *International Standard*, First edition 2001-05-01.

ISO. 2004. Part 6: Virtual terminal. ISO 11783. *International Standard*, First edition 2004-06-15.

ISO. 2002. Part 9: Tractor ECU. ISO 11783. *International Standard*, First edition 2002-07-01.

Stone, M.L., McKee, K.D., Formwalt, C.W, Benneweis, R.K 1999. ISO 11783: An Electronic Communications Protocol for Agricultural Equipment. *Agricultural Equipment Technology Conference, Louisville, Kentucky*. 7-10 February 1999.