Technical Note

Journal of Biosystems Engineering

J. Biosyst. Eng. 43(1):72-80. (2018. 3)
https://doi.org/10.5307/JBE.2018.43.1.072

eISSN : 2234-1862
pISSN : 1738-1266

Comparative Analysis of TTAK.KO-06.0288-Part3 and Development of an Open-source Communication Library for Greenhouse Control System

Joon Yong Kim[1], **Sangcheol Kim**[2]*, **Jaesu Lee**[2]

[1]*Research Institute for Agriculture and Life Sciences, Seoul National University, Seoul, Republic of Korea*
[2]*Dept. of Agricultural Engineering, National Institute of Agricultural Sciences, Jeonju, Republic of Korea*

Received: October 19th, 2017; Revised: January 17th, 2018; Accepted: February 16th, 2018

Abstract

Purpose: A modern greenhouse consists of various Information and Communications Technology (ICT) components e.g., sensor nodes, actuator nodes, gateways, controllers, and operating softwarethat communicate with each other. The interoperability between these components is an essential characteristic for any greenhouse control system. A greenhouse control system could not work unless the components communicate via common interfaces. The TTAK.KO-06.0288 is an interface standard consisting of four parts. Notably, TTAK.KO-06.0288-Part3, which describes the interface between a greenhouse operating system (GOS) and a greenhouse control gateway (GCG), is the core standard of TTAK.KO-06.0288. The objectives of this study were to analyze the TTAK.KO-06.0288-Part3 standard, to suggest alternative solutions for identified issues, and to develop a library as a proof of the alternative solutions. **Methods:** The “data field” was analyzed using a comparative analysis method, since it is a data transmission unit of TTAK.KO-06.0288-Part3. It was compared with other parts of TTAK.KO-06.0288 in terms of definition, format, size, and possible values. Although TTAK.KO-06.0288-Part1 and TTAK.KO-06.0288-Part2 do not use a “data field,” they have a similar data structure. That structure was compared with the “data field” of TTAK.KO-06.0288-Part3. **Results:** Twenty-one issues were identified across four categories: inter-standard issues, intra-standard issues, operational issues, and misprint issues. Since some of the issues can raise interoperability problems, 16 alternative solutions were suggested. In order to prove the alternative solutions, an open-source communication library called libtp3 was developed. The library passed 14 unit tests and was adapted to two research. **Conclusions:** Although TTAK.KO-06.0288-Part3 is an interface standard for communication between a GOS and a GCG, it might not communicate between different implementations because of the identified issues in the standard. These issues could be solved by the alternative solutions, which could be used to revise TTAK.KO-06.0288. In addition, a relevant organization should develop a program for compatibility testing and should pursue test products for smart greenhouses.

Keywords: Greenhouse automation, Smart farming, Interoperability, TTAK.KO-06.0288, Communication interface standard, Greenhouse control system

Introduction

A greenhouse is a structure that maintains optimal growing conditions for plants and protects them from major disturbances in the external environment. Various control technologies have been researched to control the environments of greenhouses (Duarte-Galvan et al., 2012). A modern greenhouse control system, based on the best available technology, has been highlighted as one of the “smart farm” technologies in the Rep. Korea.

A greenhouse control system is a type of Farm Management Information System (FMIS). A FMIS is defined as a system for collecting, processing, storing, and disseminating the data needed to carry out the operational functions of the farm (Sorensen et al., 2010).

***Corresponding author:** Sangcheol Kim
Tel: +82-63-238-4031; **Fax:** +82-63-238-4035
E-mail: sckim7777@korea.kr

Bakker et al. (1995) introduced a computerized greenhouse control system. The greenhouse control system consists of three key components: sensors, actuators, and controllers, along with a control algorithm. Sensors measure environmental factors and plant growth; greenhouse operating software in the controller calculates the models and issues control commands; and actuators execute the commands.

Greenhouse control strategies and wireless sensor networks have been important research subjects for greenhouse control systems. Van Straten and van Henten (2010) reviewed more than 110 papers regarding greenhouse control strategies. They divided most greenhouse control methods into three control paradigm categories: feedback control of fast processes, strategies driven by slow-crop processes, and integrated solutions. These control paradigms have been widely applied, from simple environmental control to cost optimization. In the communication area of greenhouse control and operation, wired communication is the common standard because of the harsh environmental conditions, such as high humidity and power supply challenges. With new developments in communication technology, positive results from wireless sensor network research have been reported (Chaudhary et al., 2011; Ibayashi et al., 2016; Park and Park, 2011).

Nowadays, a lack of interoperability is a major obstacle for smart farming because the ICT components of multiple vendors do not operate as an integrated system (Nash et al., 2009; Kruize et al., 2016). The heterogeneous data structures of many existing systems for data acquisition, GIS-based field indexing, different documentation tasks, and precision farming applications result in a variety of data formats and interfaces. Many manual steps are often required to simply convert data from one format to another (Steinberger et al., 2009). Pesonen et al. (2014) also pointed out that interoperability, communication, and smooth data flows are challenging to accomplish in complex functional environments, such as a farm. In order to enhance interoperability in farm environmental monitoring systems, Kim et al. (2013) presented a data middleware to integrate multiple sensor networks. And Kim et al. (2015) suggested a data-exchange platform to facilitate interoperability between FMIS and agricultural information systems.

Nevertheless, interoperability between components of a greenhouse control system is still an important issue. Although there have been many studies about greenhouse control systems, there are only a few studies about the interoperability between components in a greenhouse control system. As greenhouses have spread since the 1990s, demands for greenhouse infrastructure, such as heaters, ventilation fans, and curtain winders has also increased. However, such machines and equipment have been used with different specifications. This disparity has caused problems in the construction and operation of greenhouses, especially when they undergo needed repairs or replacement, often with different brands. Hong et al. (1998) pointed out the necessity for standardization of the machinery and equipment for greenhouse operations and suggested standard specifications for heaters, ventilation fans, and curtain winders. Jeun et al. (1999) extended the range of standardization to include electronic equipment for greenhouses. Their suggestions included where and how to install devices, measuring the range and error of environmental sensors, and measuring the functions of the greenhouse control system. However, their research was insufficient in three points. First, it did not cover all greenhouse systems because these are complex facilities with highly variable components, such as electronic controllers, computers, and sensors. Second, it did not deal with the communication interfaces between components. Third, the results were not published as a standard.

Since 2010, the standardization of greenhouses has received attention again. The RFID/USN association (currently, the Korean IoT association) has tried to propose standards related to plant factories and greenhouses. Various standards, including the composition of components for greenhouses, the interface between components, and operational guidelines for this growth environment, were established as standards of the Telecommunications Technology Association of Korea (TTA) (Hwang et al., 2013). Lately, various standards related to greenhouses were published in an explosion of interest in smart farming. For example, TTAK.KO-10.0843 describes the metadata according to classified types of diagnostic data for crop growth in greenhouses (TTAK.KO-10.0843), and TTAK.KO-10.0934 describes the functional interfaces between the components of a greenhouse (TTAK.KO-10.0934).

In Japan, the Ubiquitous Environment Control System®(UECS), which is a decentralized greenhouse environmental control system, was proposed in 2005

(Hoshi et al., 2008). Communication protocols for exchanging messages between UESCs were developed in 2010 and were revised in 2012 (*UECS Consortium*, 2012). The communication message in the UECS is called the Common Communication Message (CCM), and it is described by Extensible Markup Language (XML). Protocols for CCM have been explored in previous studies (Hoshi et al., 2011; Ibayashi et al., 2014).

In Rep. Korea, TTAK.KO-06.0288 is an interface standard set for a greenhouse control system, and the government agency recommends complying with this standard. The standard is based on TTAK.KO-06.0286, which specifies the application requirements and structures of the greenhouse control system with five components: sensor nodes, actuator nodes, the greenhouse control gateway (GCG), the greenhouse operating system (GOS), and integrated greenhouse management systems (IGMS). The first version of this standard was published in 2012, and it was partially revised in 2015 (TTAK.KO-06.0288- Part1/R1). Four parts of the standard describe protocols of the interfaces between components. There is another interface standard called TTAK.KO-10.0943, which describes the interface between sensor-actuator nodes and the GCG (TTAK.KO-10.0943). It was established based on TTAK.KO-06.0288. The difference is that the TTAK.KO-10.0943 defines a sensor-actuator node, which is an integration of a sensor node and an actuator node and describes communication between the sensor-actuator nodes and the GCG. Similarly, most communication interface standards were initiated from TTAK.KO-06.0288 and some of the other standards, which do not handle communication interfaces but also refer to TTAK.KO-06.0288. Therefore, TTAK.KO-06.0288 is a key standard of communication interface standards in Rep. Korea.

As mentioned before, interoperability between components of a greenhouse control system is important, and TTAK.KO-06.0288 is a key standard to increase and improve that interoperability. However, the standard has rarely been used in the industrial field of greenhouse control systems. In order to encourage usage of this standard, the objectives of this study were to seek out inconsistencies and solutions for TTAK.KO-06.0288 through comparative analysis and to develop an open-source library without any restrictions. The research was conducted in three steps. First, TTAK.06.0288 was analyzed to compare the data fields of TTAK.KO-06.0288-Part3 with the data fields of other parts, in terms of definition, type, size, and values. Second, alternative solutions were suggested when issues arose. And last, an open-source communication library was implemented to show that the proposed alternatives will aid in utilizing the standard.

Materials and Methods

Summary of TTAK.KO-06.0288

The TTAK.KO-06.0288 defines the components of the greenhouse control system and their interface protocols. The components include sensor nodes, actuator nodes, the GCG, the GOS, and the IGMS. Figure 1 shows the relationships between the components and the interface protocols.

A sensor node includes a sensor (or sensors) and a communication module to transfer observations using TTAK.KO-06.0288-Part1 (TTAK.KO-06.0288-Part1/R1). An actuator node operates by receiving a control command via TTAK.KO-06.0288-Part2 and executing it (TTAK.KO-06.0288-Part2/R1). The GCG is a mediator between nodes (sensor nodes, actuator nodes) and the GOS. It receives measurements from sensor nodes and sends them to the GOS using TTAK.KO-06.0288-Part3 (TTAK.KO-06.0288-Part3). The GOS can have environmental models and/or control algorithms. It receives sensor observations and generates control commands using the

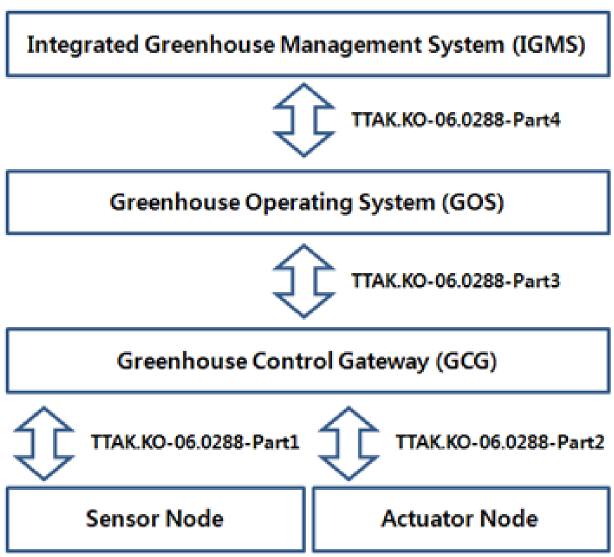

Figure 1. Components of a greenhouse control system and interface standards.

control algorithms and observations. A command is issued to an actuator node via the GCG. Lastly, an IGMS is a monitoring system for multi-site greenhouses, but it does not have any control functions (TTAK.KO-06.0288-Part4).

In summary, a message of TTAK.KO-06.0288-Part1 and a message of TTAK.KO-06.0288-Part2 could be translated to messages of TTAK.KO-06.0288-Part3 by the GCG, and the GOS has full authority over a greenhouse using TTAK.KO-06.0288-Part3. Therefore, the central interface of TTAK.KO-06.0288 is the TTAK.KO-06.0288-Part3.

Details of TTAK.KO-06.0288-Part3

The full name of TTAK.KO-06.0288-Part3 is "Greenhouse control system-Part 3: Interface between the greenhouse control gateway and the greenhouse operating system (TTAK.KO-06.0288-Part3)." A GOS includes the functions of internal and external environmental monitoring, life cycle management of sensor nodes and actuator nodes, fault management, and so on. For this purpose, a GCG gathers internal and external environmental measurements from sensor nodes and sends the appropriate control commands to actuator nodes.

To support the communication between a GOS and a GCG, the TTAK.KO-06.0288-Part3 has eight functions: a connection approval function, a notice function, a profile function, a status information function, an environmental information function, an actuator control function, a fault management function, and a network status check function. These function messages can be transferred by two message transmission models. The TTAK.KO-06.0288-Part3 uses a binary data format consisting of three components: total length, data, and a checksum. The total length component is a four-byte unsigned integer that represents the total bytes of a message. The checksum component is a two-byte word computed by exclusive-or (XOR) of the data component. The data component is a set of data field blocks. The data field has a field code, a field length, and a field data. The field code describes what the data is. For example, 0x1000 refers to a number of sensor nodes. Field length represents the size of the field data. And the field data is the actual value of the data field. Figure 2 shows the message structure of the TTAK.KO-06.0288-Part3.

As shown in Figure 2, a message of TTAK.KO-06.0288-Part3 consists of many data fields, and each data field has its own meaning. Although TTAK.KO-06.0288-Part1 and TTAK.KO-06.0288-Part2 do not use the "data field" per se, they have a similar message structure. The data field is a unit of data to transfer, so it was used as an object of analysis.

Comparative analysis method

A data field of TTAK.KO-06.0288-Part3 has its code, length, and value. They should match with other parts of TTAK.KO-06.0288. The field code represents the meaning of the data in a data field. The definition and description field codes should match others. In the same manner, data type, data size, and the values in a data field should also match. Therefore, the TTAK.KO-06.0288-Part3 and other parts of TTAK.KO-06-0288 were compared in terms of terminology, data type, data size, and the values of the data field.

First, definitions of data fields used in the standards were compared. As TTAK.KO-06.0288 is a set of standards, it should share all details, such as concepts and

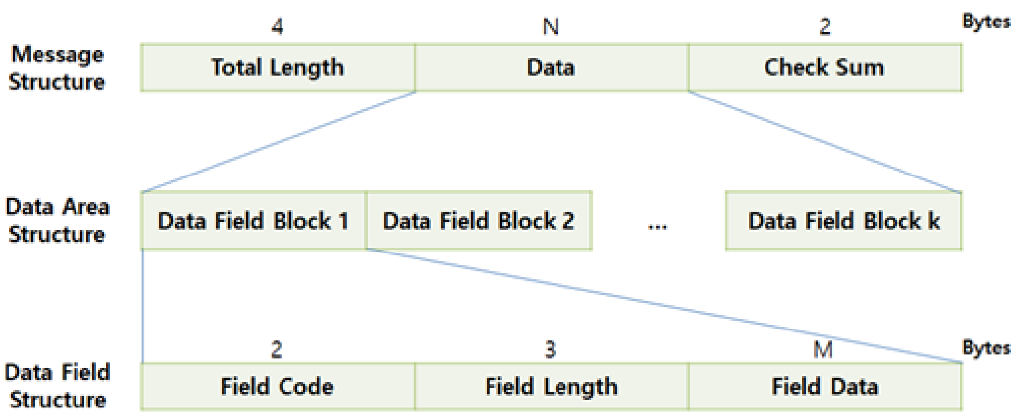

Figure 2. Message structure of TTAK.KO-06.0288-Part3 (TTAK.KO-06.0288-Part3).

descriptions. Each standard of TTAK.KO-06.0288 shares its concept well, but it contains some different descriptions, thus making implementation difficult.

Second, types of data fields were compared. In computer science, a data type is a classification of data. It is obvious that the observations of two temperature sensors should be the same. If two standards use different types of the same data field, it is impossible to translate messages between the standards.

Third, the sizes of data fields were also compared. Even if messages use the same field type, different sizes in the data field may cause data loss. For example, sensor type represents the type of a particular sensor in TTAK.KO-06.0288. If there is a temperature sensor and its sensor type is 257, it is possible to use it in TTAK.KO-06.0288-Part3 because TTAK.KO-06.0288-Part3 uses two bytes for the sensor type. But it is impossible to use the sensor type "257" in TTAK.KO-06.0288-Part1 because this protocol uses only one byte for the sensor type. Thus, there might be information loss because of the size difference.

Lastly, values that a data field can have were compared. There are some data fields that have static values. Actuator type is a good example. There could be lots of actuator types, such as windows, ventilation fans, thermal screens, heaters, heating pumps, and so on. Each actuator type should be mapped with a value, and the values should be shared with other implementations. However, TTAK.KO-06.0288 does not describe actuator types.

Results and Discussion

Analyzed issues

Twenty-one issues were identified as a result of comparative analysis. The issues were classified into four categories. The first category is the inter-standard issue. This issue makes it impossible to translate one interface to another. For example, TTAK.KO-06.0288-Part3 uses two bytes for the sensor type, but TTAK.KO-06.0288-

Table 1. Issue categories and issue list of TTAK.KO-06.0288-Part3

Category	Issues
Inter-standard	1. Part3 needs sensor data type, collection type, collection cycle, upper limit, lower limit, precision, and error range, but Part1 does not support them. 2. Part3 uses two bytes for sensor type, but Part1 uses one byte for it. 3. Part3 needs an actuator data type, but Part2 does not support it. 4. Part3 uses two bytes for actuator type, but Part2 uses one byte for it. 5. Part3 uses one or four bytes for the actuator data value, but Part2 uses two bytes for it; and negative values or floating point values can cause a problem.
Intra-standard	1. Part3 contains conflicting descriptions of whether a message sequence number is binary or a string. 2. The length of actuator data is not clear in Part3. 3. A sensor is identified by Gateway Level SID, not by Sensor ID in Part3. 4. Sensor data type 0x00 is a float type. The float type format is not defined in Part3. 5. There is no description of the actuator node profile request in Part3.
Operational	1. Sensor type is not defined in Part1. Two different implementations cannot communicate. 2. Sensor type is defined for only two sensors, temperature and humidity, in Part3. 3. There is no document named "Greenhouse control data specification," which is a reference for sensor type in Part3. 4. Actuator type is not defined in Part2. 5. Actuator type defines only two actuators – heat pumps and side windows. 6. There is no document named "Greenhouse control data specification," which is a reference to actuator type in Part3. 7. There are only two control methods – on/off and float (analog) – for actuators in Part3. Additionally, the meaning of float is not clear for analog controls. 8. GOS cannot request environmental data from specific sensors in Part3. 9. The maximum number of environmental data responses is 255. This is too small considering the number of sensors in a greenhouse.
Misprint	1. Although the sensor node ID and actuator node ID are each 20 bits in length, they are written to 3 bytes or 1 byte. 2. Part1 contains a table for actuator node. It should be for sensor node.

• For more details, see the analysis document on webpage (https://ezfarm-farmcloud.github.io/libtp3/md_doc_8src_ttap3_analysis.html).

Part1 uses only one byte for it. In this case, the size of the sensor type value should be limited to one byte, and TTAK.KO-06.0288-Part3 should waste one byte. The second category is the intra-standard issue category. If there are more than two descriptions for one item in a document, but they are not the same, it is impossible to know which one is correct. For example, there are two different descriptions of message sequence number in TTAK.KO-06.0288-Part3. Since they do not match, it is impossible to choose one. The third category is the operational issue category. An issue in this category is not a contradiction of the standard itself, but it may cause a problem when an implemented system operates. The last category relates to the misprint issue. Table 1 lists all the issues.

In addition, several comments have been made suggesting that it might helpful to revise the TTAK.KO-06.0288-Part3. First, 20 bits are allocated for node ID. This makes it difficult to handle the ID, even though there is a reference standard for it. If its size were just two or three bytes, it would be easier to handle. Second, seven bytes of structure -“YYYMMDDHHMMSS”- are used to represent date and time in TTAK.KO-06.0288-Part3. If the reason for using this structure is to save space, then epoch time would be better. If saving space is not the reason, then there is no advantage to using date and time compared with other data types, especially considering that other fields for time or date use strings. Finally, there is no reason to use six bytes for a message sequence number. It is too large, and more time is needed to process it on most embedded systems.

Alternative solutions

Some issues make it impossible to implement the standard. Therefore, alternative solutions should be prepared for these situations. Table 2 shows the proposed alternative solutions for the various issues, by category. Fortunately, the misprint issues can be solved by guessing the meaning from the context.

Development of libtp3

In order to prove the alternative solutions, a library called “libtp3,” which is an implementation of TTAK.KO-06.0288-Part3, was designed with five modules: the base module, the utility module, the process module, the gcg module, and the gos module. The base module contains data types, constants, and functions for message structure. The utility module supports the utility functions needed to use the base module easily. The process module handles callback functions that are used when a matched

Table 2. Alternatives for TTAK.KO-06.0288-Part3 issues

Category	Alternative solutions	Issues from Table 1
Inter-standard	1. Keep the message format of each standard, but the value should be ignored.	#1, #3
	2. Keep the message format of each standard, but only one byte of data should have meaning.	#2, #4
	3. Internally, the actuator data is treated as two-byte arrays. A library user can use it with a type cast method.	#5
Intra-standard	1. The message sequence number is treated as a binary value.	#1
	2. Internally, the actuator data is treated as two-byte arrays. A library user can use it with the type cast method.	#2
	3. The Gateway Level SID is used instead of sensor ID.	#3
	4. The 0x00 sensor data type is not used.	#4
	5. The actuator node profile request is implemented similarly to the sensor node profile request in Part3.	#5
Operational	1. Sensor type is ignored because a library user should handle sensor type and sensor data.	#1, #2
	2. The document is ignored.	#3
	3. Actuator type is ignored because a library user should handle actuator type.	#4
	4. Actuator type is ignored because a library user should handle actuator type.	#5
	5. The document is ignored.	#6
	6. Float is not used, and two bytes for actuator data is defined to have working time (seconds) and an argument.	#7
	7. Although enough functions do not exist, it follows the standard.	#8
	8. It ignores data exceeding 255 elements in order to follow the standard.	#9

message arrives. The gcg module and the gos module are specialized modules for the GCG and GOS. There are 12 types of messages in TTAK.KO-06.0288-Part3. The gcg module helps to send four types of messages to the GOS, and the gos module helps to send eight types of messages to the GCG.

The libtp3 library was written in the C programming language. It uses the libuv library, which is a multi-platform support library with a focus on asynchronous I/O. Figure 3 shows a sequence diagram for message processing in libtp3. When a user asks to generate or parse a message, libtp3 generates or parses it using a registered callback function or a default callback function for the message type.

In order to test the library, 14 unit tests were prepared. The unit tests were classified into three categories: base, utility, and connection. The base category tests basic functions, such as handling a message frame, reading data fields, generating a proper response, and so on. The utility category checks extra functions to deal with a data field having multiple data blocks. The environmental data field and status data field could be the examples having multiple data blocks. Since the structures of the data fields are different from other data fields, tests of the utility category are necessary. Finally, the connection category examines connections between the GOS and GCG. In addition, it tests auto generating requests and responses. The 14 unit tests were executed on Linux platforms (Raspbian Jessie on Raspberry PI 3, Lubuntu on Cubieboard2, and Mint Linux on desktop PC) and the Windows platform.

The libtp3 library was published on Github (https://github.com/ezfarm-farmcloud/libtp3) with the BSD license. Since the BSD license is an open-source license, imposing minimal restrictions on the use and redistribution of covered software, it could be used freely. In reality, it was already adopted to develop two applications.

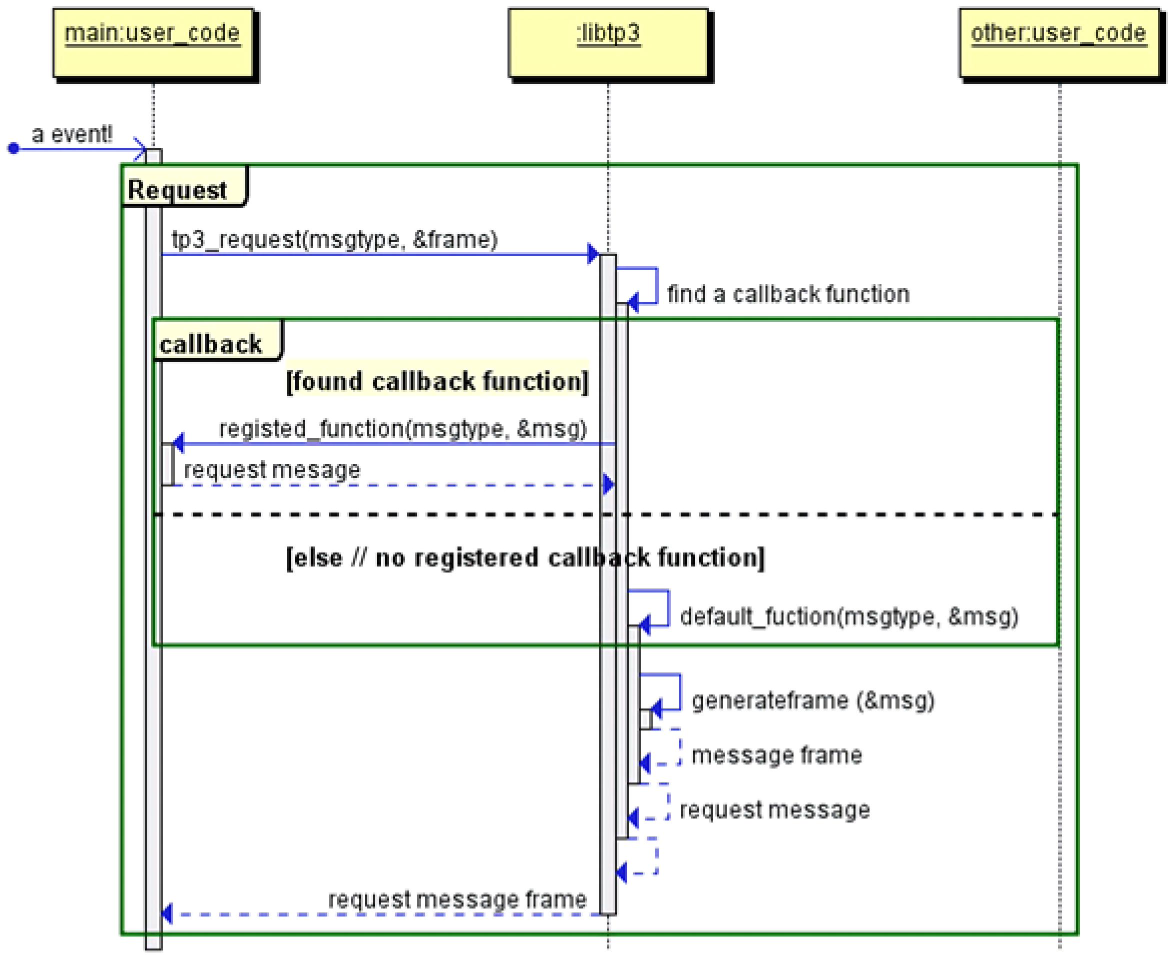

Figure 3. Sequence diagram for a request message processing in libtp3.

The first application was a greenhouse control system engine developed by the Rural Development Administration in Rep. Korea and EZFARM Co., Ltd. The engine, called "cflora," was developed as a common platform for greenhouse control systems. The libtp3 was used to communicate between modules that play the roles of the GCG and GOS. The greenhouse control system engine was also published on Github (https://github.com/ezfarm-farm cloud/cflora). The application was tested in a glass greenhouse at the Protected Horticulture Research Institute, Haman, Gyeongsangnam-do, Republic of Korea.

Second, the libtp3 was also used for communication between the gateway and greenhouse operating software. The gateway and greenhouse operating software were developed by The Convergence Research Center for Smart Farm Solution, Korea Institute of Science and Technology (KIST). The gateway and greenhouse operating software were installed in a Venlo type, multi-greenhouse (experimental area: 16 m x 12.5 m, 200 m^2) located in the Gangleung KIST testing horticulture field. The operating software was used to predict inside temperature and humidity in the greenhouse and to control ventilation (Jung et al., 2017a; Jung et al., 2017b). It has been tested for more than a half-year, and there has been no communication errors related to the library thus far.

Conclusions

In this study, 21 issues in the TTAK.KO-06.0288-Part3 standard were identified, and 16 alternatives were suggested to solve these issues. An open-source library was implemented in order to prove the alternatives. The libtp3 passed 14 unit tests and was used in two research projects without any problems. Although the alternatives worked, TTAK.KO-06.0288 still needs to be revised. In addition, it might be important for a responsible organization to develop a compatibility testing program and to administer the program for government subsidies.

Users can benefit from well-designed standards. If enough vendors follow the standards, a user has a better chance of choosing a suitable product without concern for interoperability. Well-designed standards reduce production costs for manufacturers. In order to share the benefits of standards, continued interest and participation are needed.

Acknowledgements

This work was carried out with the support of the "Cooperative Research Program for Agriculture Science & Technology Development (Project No. PJ01205901)," Rural Development Administration, Republic of Korea. A preliminary version of this paper was presented at the 2015 KSAM Spring Meeting.

References

Bakker, J. C., G. P. A. Bot, H. Challa and N. J. van de Braak. 1995. Greenhouse Climate Control - An integrated approach. Wageningen, The Netherlands: Wageningen Academic Publishers.

Chaudhary, D. D., S. P. Nayse and L. M. Waghmare. 2011. Application of wireless sensor networks for greenhouse parameter control in precision agriculture. International Journal of Wireless & Mobile Networks 3(1):140-149.

Duarte-Galvan, C., I. Torres-Pacheco, R. G. Guevara-Gonzalez, R. J. Romero-Troncoso, L. M. Contreras-Medina, M. A. Rios-Alcaraz and J. R. Millan-Almaraz. 2012. Review. Advantages and disadvantages of control theories applied in greenhouse climate control systems. Spanish Journal of Agricultural Research 10(4): 926-938.

Hong, S.-G., Y.-S. Chang, J.-H. Yun, S.-H. Kim and K.-I. Lee. 1998. Standardization of machinery and equipments for greenhouse. In: *Proceedings of the Korean Society for Bio-Environment Control Conference,* pp. 68-76. (In Korean)

Hoshi, T., Y. Hayashi and K. Shintani. 2008. A communica -tion protocol for collaboration among the measurement and control nodes in a decentralized autonomous environment control system of greenhouses. In: *World conference on agricultural information and IT,* pp. 127-134.

Hoshi, T., R. Ohata, K. Watanabe and R. Osuka. 2011. A gadget-based information management system for environmental measurement and control in greenhouses. In: *SICE Annual Conference,* pp. 2801-2805.

Hwang, J.-H., H.-S. Jeong and H. Yeo. 2013. Trend and prospect of technology convergence of agriculture and food with IT. Information and Communications Magazine 30(10): 53-60.

Ibayashi, H., Y. Kaneda, J. Imahara, N. Oishi, M. Kuroda and H. Mineno. 2016. A reliable wireless control system for tomato hydroponics. Sensors 16(5): 644.

Ibayashi, H., Y. Kaneda, Y. Suzuki and H. Mineno. 2014. Highly reliable wireless environmental control system for home gardening. In: *2014 IEEE 3rd Global Conference on Consumer Electronics,* pp. 725-726.

Jeun, J. G., G. W. Kim and B. G. Oh. 1999. Study on the standardization of environment control system for greenhouse. In: *Conference of the Korean Society for Bio-Environment Control*, pp. 67-70. (In Korean)

Jung, D. -H., H. -J. Kim, S. H. Park and J. Y. Kim. 2017a. Recurrent neural network models for prediction of the inside temperature and humidity in Greenhouse. In: *Proceedings of the KSAM & UMRC 2017 Spring Conference,* pp. 135.

Jung, D. -H., H. -J. Kim, J. Y. Kim and S. H. Park. 2017b. Design optimization of proportional plus derivative band parameters used in greenhouse ventilation by surface response method. In: *Proceedings of the KSAM & ARC 2017 Autumn Conference,* pp. 177.

Kim, J. Y., C. G. Lee, S. H. Baek and J. -Y. Rhee. 2015. Open farm information system data-exchange platform for interaction with agricultural information systems. Agricultural Engineering International: CIGR Journal 17(2): 296-309.

Kim, J., C. Lee, T. -H. Kwon, G. Park and J. -Y. Rhee. 2013. Development of an agricultural data middleware to integrate multiple sensor networks for a farm environment monitoring system. Journal of Biosystems Engineering 38(1): 25-32.

Kruize, J. W., J. Wolfert, H. Scholten, C. N. Verdouw, A. Kassahun and A. J. M. Beulens. 2016. A reference architecture for farm Software ecosystems. Computers and Electronics in Agriculture 125: 12-28.

Nash, E., P. Korduan and R. Bill. 2009. Applications of open geospatial web services in precision agriculture: a review. Precision Agriculture 10(6): 546-560.

Park, D. -H. and J. -W. Park. 2011. Wireless sensor network-based greenhouse environment monitoring and automatic control system for dew condensation prevention. Sensors 11(4): 3640-3651.

Pesonen, L. A., F. K. -W. Teye, A. K. Ronkainen, M. O. Koistinen, J. J. Kaivosoja, P. F. Suomi and R. O. Linkolehto. 2014. Cropinfra - An Internet-based service infrastructure to support crop production in future farms. Biosystems Engineering 120:92-101.

Sorensen, C. G., L. Pesonen, S. Fountas, P. Suomi, D. Bochtis, P. Bildsøe and S. M. Pedersen. 2010. A user-centric approach for information modelling in arable farming. Computers and Electronics in Agriculture 73(1): 44-55.

Steinberger, G., M. Rothmund and H. Auernhammer. 2009. Mobile farm equipment as a data source in an agricultural service architecture. Computers and Electronics in Agriculture 65(2): 238-246.

TTAK.KO-06.0288-Part3, 2012. Greenhouse control system - Part 3: Interface between greenhouse control gateway and greenhouse operating system. Rep. Korea: Telecommunications Technology Association (In Korean, with English abstract).

TTAK.KO-06.0288-Part4, 2013. Greenhouse control system - Part 4: Interface standard between greenhouse operating system and integrated greenhouse management system. Rep. Korea: Telecommunications Technology Association (In Korean, with English abstract).

TTAK.KO-06.0288-Part1/R1, 2015. Greenhouse control system - Part 1: Interface between sensor nodes and greenhouse control gateway. Rep. Korea: Telecommunications Technology Association (In Korean, with English abstract).

TTAK.KO-06.0288-Part2/R1, 2015. Greenhouse control system - Part 2: Interface between actuator nodes and greenhouse control gateway. Rep. Korea: Telecommunications Technology Association (In Korean, with English abstract).

TTAK.KO-10.0843, 2015. Protected horticulture : Metadata for the diagnosis of plant growth. Rep. Korea: Telecommunications Technology Association (In Korean, with English abstract).

TTAK.KO-10.0934, 2016. Interfaces between functional entities for smart greenhouse. Rep. Korea: Telecommunications Technology Association (In Korean, with English abstract).

TTAK.KO- 10.0943, 2016. Communication protocol between greenhouse control gateway and sensor-actuator integrated nodes for smartfarm application. Rep. Korea: Telecommunications Technology Association (In Korean, with English abstract).

UECS Consortium. 2012. Ubiquitous Environmental Control System (UECS), Basic communication protocol book. UFCS-CMM Standardization Committee.

Van Straten, G. and E. J. van Henten. 2010. Optimal greenhouse cultivation control: Survey and perspectives. IFAC Proceedings Volumes 43(26): 18-33.