

Article

Conformance Testing of SGSF-064-1 Using CANoe

Intaek Kim ^{1,*}, Ahmed Al-Hilo ¹, Hyuk Soo Jang ² and Jong-Geol Yoo ³

¹ Department of Communications and Information Engineering, Myongji University, Yongin 17058, Korea; E-Mail: ahmedsmail_91@yahoo.com

² Department of Computer Engineering, Myongji University, Yongin 17058, Korea; E-Mail: jang-h@mju.ac.kr

³ Korea Testing Laboratory, S.M.A.R.T. Infrastructure Center, Jinju 52852, Korea; E-Mail: yoojg@ktl.re.kr

* Author to whom correspondence should be addressed; E-Mail: kit@mju.ac.kr; Tel.: +82-10-4206-0879.

Academic Editor: Minh Shin

Received: 11 September 2015 / Accepted: 4 November 2015 / Published: 11 November 2015

Abstract: In this paper, the authors describe a conformance testing system for SGSF-064-1, the communication protocol between electric vehicles and conductive DC (direct current) chargers in Korea. Since the SGSF-064-1 is based on CAN (controller area network), the testing system was developed by CANoe. The DC charger known as EVSE (electric vehicle supply equipment) is the system being tested and the developed system implemented in PC (personal computer). The developed system performs as a tester to ensure that the DC chargers from various manufactures can conform to the communication protocol in SGSF-064-1. The testing system contains four testing modes which also consist of several test cases.

Keywords: conformance test; SGSF-064-1; electric vehicle; EVSE; CAN bus; CANoe

1. Introduction

Electric vehicles (EVs) were once popular in the late 19th century and early 20th century before cars with internal combustion engines became available. A century later, EVs have begun gaining attention once again and are regarded as the means of transportation for the near future for their environmentally friendly features. Most present automobile manufacturers have begun making a

considerable effort in developing various types of EVs including electric cars, plug-in hybrids, and hybrid cars. However, the rapid deployment of EVs is hindered by the limitation of batteries and a shortage of charging stations.

In current technology, DC charging is faster than AC (alternating current) in delivering power to EVs. It usually takes less than 30 min to charge up to 80% of the battery's capacity, and it is therefore the preferred way of charging on the street. Unlike conventional fuel cars, electric charging for EVs is more complicated when it comes to ensuring safe and fast charging. At the moment, several standards dictating the charging methods are available worldwide [1].

Both electric vehicles (EVs) and electric vehicle supply equipment (EVSE), also known as EV chargers, should be manufactured according to the industrial or organizational standard such as the International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), and Smart Grid Standardization Forum (SGSF). In order to check if they meet the standard, a procedure for testing is necessary. Conformance testing, also known as compliance testing, is a type of methodology utilized in engineering to make sure that a product, process, or even computer program meets a defined set of standards.

This paper developed conformance testing for the SGSF-064-1, which is known as the Korean standard for the communication interface of EVs and EVSE for fast DC charging. The testing system was developed with CANoe, the most popular software developed by Vector for testing or developing CAN-based communications protocols. In this paper, conformance testing was applied to EVSE and it was therefore assumed to be a system under test (SUT). The EV side was implemented into a PC and it checks if the EVSE is faithfully following the procedure defined in the SGSF-064-1. Four test modes including one normal mode and three abnormal modes have been proposed for this purpose. They can be applied separately by navigating the menu in CANoe's main window. The software implemented for conformance testing with CANoe can be used commercially to assess the conformity of EVSE made according to SGSF-064-1.

The implementing methods of conformance testing for the SGSF-064-1 have been evolved by the authors [2,3]. The objective was to realize the system for conformance testing for the standard but the methods have been renewed. In the first paper [2], Testing and Test Control Notation Version 3 (TTCN-3) were employed in a single PC to simulate and test procedures in SGSF-064-1. The second paper [3] extends the contribution of the first one by using two PCs; one simulates EVSE and the other works as a tester. A real CAN bus was also utilized thanks to two serial-to-CAN converters, and thus, two PCs were communicated with by CAN protocol. In this paper, the previous work is advanced by replacing TTCN-3 with CANoe, which is a comprehensive tool for testing and analyzing the CAN bus. In addition, the paper includes some results of four testing modes on a real EVSE.

We organize the remainder of this paper as follows. Section 2 describes the definition and the properties of conformance testing. A brief explanation on CAN is given in Section 3. Testing software used for developing the proposed system, CANoe, is the topic of Section 4. In addition, it includes CAN Access Programming Language (CAPL). Section 5 briefly explains the Korean standard. Section 6 describes the structure of the test system. Finally, Section 7 presents the conclusions of this paper.

2. Conformance Testing

Conformance testing is an inspection method that functions as a way to check and affirm that a product faithfully meets the requirements of a specific standard. Namely, it is a way to ascertain that a predicted standard is implemented in the product. In other words, it aims to achieve the main goal of communication standards which ensure that all manufacturers follow applicable requirements [4]. This approach of testing is applied in different areas including electronic and electrical engineering, software engineering, and other areas as well [5].

Besides conformance testing, there are many types of testing such as performance testing, robustness testing, department testing, functions, and interoperability. In general, testing does not check if a product complies with standard specifications. It can only detect the lack of conformity between the two “views” of the system [6].

According to the European Telecommunications Standards Institute (ETSI), the regular form of conformance testing contains two sides, system under test (SUT) and test system [7]. The test systems run test procedures which include some test cases. A test case or a test scenario specifies the behaviors of a standard in adequate detail, so the test system can assess a product’s behavior and determine whether it precisely follows a standard or not. At the end of each test case, the test case generates a verdict of pass/fail. A conformance testing program usually includes a standard or specification, test tools, procedures for testing and organization to-do testing, and issues certificates of validation [8].

In SGSF-064-1, there are some cases where conformance testing should be done to ensure that EVSE meets Korean standard specifications. The SGSF-064-1 has some procedures that EVSE should follow. In this paper we use test modes which consist of several test cases and then the test system checks each test case individually to confirm that EVSE complies with the requisites of SGSF-064-1.

3. CAN Communication

Controller area network (CAN) is a serial communication bus primarily used for vehicle component communication defined by the ISO [9]. CAN is a multi-master bus that allows microcontrollers and devices to communicate with each other in applications without a host computer or central processing unit (CPU). It was originally developed by the German BOSCH Corporation in the early 1980s for the automotive industry to replace the complex wiring harness with a two-wire bus.

The CAN bus is a broadcast-based protocol where every individual node listens to the bus and it is a job for the CAN physical layer to filter messages according to their identifiers [10]. Each CAN node contains a CAN transceiver and microcontroller with a CAN controller. The CAN transceiver receives the data stream and converts it to a form that the CAN controller can use. On the other hand, the CAN transceiver sends the data coming from the CAN controller after converting it to the CAN bus level. The CAN bus has two transmission lines: CAN-Low and CAN-High [11].

The latest version of CAN, CAN 2.0, was announced in 1991 with two parts: CAN 2.0A and CAN 2.0B. The former has an 11-bit identifier while the latter uses a 29-bit identifier. CAN 2.0 has bitrates up to 1 Mbit/s with a message frame having data field up to eight bytes as shown in Figure 1 [12].

Arbitration field						Control field			Data field	CRC field	ACK		EOF	
SOF	11 bit Identifier	SRR	IDE	18 bit Identifier	RTR	r1	r0	DLC	Data (0-8 Bytes)	15 bits	Delimiter	Slot	Delimiter	EOF

Figure 1. Data frame of CAN 2.0 B.

SGSF-064-1 uses the CAN bus for communication. When an EV wants to get charged from an EVSE, it has to establish CAN communication first. With the CAN bus, the EV and EVSE can exchange their information for the charging process; for example, at the charging period, the EV keeps sending the state of charge (SOC) to let the EVSE know what the current percentage of charge is in the EV battery. The communication in SGSF-064-1 is bidirectional, so that each side has its own identifiers. The EVSE sends messages with identifier = 0x631 for data and identifier = 0x630 for environment information, while the EV uses identifiers 0x639 and 0x638, respectively, for the same purpose. Like the CAN bus, the data field of the Korean standard includes one message and this message is divided into signals. The length of each signal varies according to its message [13].

4. CANoe Software

CANoe is an extensive development tool for the CAN bus made by Vector Informatik GmbH for simulating, testing, and analyzing electronic control unit (ECU) networks and individual ECUs. This software is used mainly by automotive and ECU industries. It provides a wide range of functions and tools for the development process such as configuring and starting-up ECU networks as well as testing and diagnostics. CANoe supports several bus systems including CAN, FlexRay, and Ethernet [14].

At the beginning of the development process for an ECU network or ECU, CANoe is used to create simulation models to imitate the behavior of the ECUs. In the second phase of the development process, CANoe is connected to the real ECU that we want to test. At last, CANoe is connected to the real ECUs including the target [15]. CANoe uses CAPL (CAN Access Programming Language) for simulation and evaluation.

As mentioned above, with CANoe it is possible to simulate an EV. When we want to test an EVSE with CANoe, we first have to apply the same actions that come with an EV. For this purpose, CANoe simulates CAN data sent from an EV to an EVSE as is done in real EVs. Moreover, CANoe can handle electric signals by using the serial RS232 bus. With the RS232 bus, the system controls an electric load and the whole system, as explained in Section 6.

4.1. Tools Supporting Extended Multiplexing

Section 3 demonstrates that CANoe must deal with messages and their signals. For this purpose, CAN uses database container (DBC) files to register messages according to their identifiers and signals depending on their start bit address and length. CANoe provides a visual tool called the CANdb++ editor to create and write in a DBC file [16].

4.2. CAN Access Programming Language

CAN Access Programming Language (CAPL) is a development and testing programming language based on the C language for developing programs in CANoe and CANalyzer. CAPL extends the capabilities of CANoe and CANalyzer. CANoe and CANalyzer have some basic tools to do simple simulations and testing, but with CAPL, simulation and testing are greatly extended for CAN communication.

The testing in CANoe consists of test modules, and each module contains several test cases. Test modules can be implemented in XML (EXtensible Markup Language) and/or CAPL. The main difference between XML and CAPL is the amount of execution time. With CAPL it is possible to execute the same test more than once [17].

For the proposed system of this paper, CAPL is utilized for writing test cases. The testing system handles the EV messages with CAPL and it sends messages from CANoe to EV as well. As CAPL opens a wide range of capabilities, it is used to observe signals in messages. Also, with CAPL, the system can determine how much time is taken by each test case. At the end of a test case, it gives a final verdict for each one separately by calculating the spent time and the received values.

5. SGSF-064-1

The Smart Grid Standard Forum (SGSF) in Korea has developed a standard for communication protocols between EVs and conductive DC chargers which is called SGSF-064-1. This standard is based on CAN and it adopted the basic idea of the CHAdEMO (CHARge de MOve) which is a quick charging method proposed by The Tokyo Electric Power Company and other five Japanese automobile manufacturers. It was updated in June of 2014 to elaborate the process to avoid the charging error that the previous standard SGS-03-001 had. The standard document consists of seven sections: scope, normative references, definitions, system components, interface circuit and requirements, charging processes, and fault diagnosis.

- Scope shows that this standard defines the communication protocol between EV and EVSE with IEC 61851 and IEC 61851-23.
- Normative references reveal other standards which are indispensable for the application of these documents. They include ISO 11898-1/2, SGS 03-003, and IEC 61851-1/23/24.
- Key terminologies used in the document are listed in the section of definitions.
- System components detail the physical connection and equipment necessary for EV charging in DC mode. Among them are circuit breaker, current limiter, electric ground, circuit for coupler locking, *etc.*
- Interface circuit and requirements describes the pilot signal circuit, wake up signal circuit, and protocol for charging the controller. The latter part further details features provided during DC charging, the circuit for CAN communication, and OSI (Open Systems Interconnection) layers for DC charging.
- The charging process can be classified into two cases according to how it is terminated: normal charging and abnormal charging. In the case of abnormal charging, the sequence leads to stopping charging before it is completed for several reasons. These faults can occur in either the

EV or EVSE and they also can happen prior to or during charging, whereas normal charging takes place when it occurs as expected. This will be elaborated on later in this section.

- Fault diagnosis is a major addition to the previous standard. This part includes 10 types of charger faults such as insulation fault, abnormal charge, wake up signal fault, pilot signal fault, and others. For each fault, the location of fault detection is identified whether it occurs in a vehicle or in a charger and the corresponding warning statement is also suggested.

The objective of the conformance testing of SGSF-064-1 is to check whether EVs and EVSEs comply with the charging sequences described in the standard. Therefore, this document details the charging process and it is further divided into three stages: pre-charging process, charging in-process, and terminating the post-charging process.

Terminating the charging sequence can be either normal or abnormal. The normal sequence occurs when the EV battery is fully charged—to be specific, it is about 80% charged. Another case is when the user pushes the stop button in the charger for some reason to end charging. Abnormal terminating charging sequences occur in several cases. When there is a fault either at the EV or EVSE, then the charging is terminated by following the EV fault or charger fault sequence. Sometimes abnormal terminating charging sequences can be initiated before charging by the request of the EV or EVSE. The charging connector cannot be unlocked due to its higher voltage condition, and it also leads to the abnormal terminating charging sequences.

The flow diagram of a normal charging sequence is shown in Figure 2. On the top, the charger and vehicle indicate EVSE and EV, respectively. Each small box represents the action of each entity and solid arrows represent wired communication while broken arrows represent CAN communication. Time T shown on the left or right next to a vertical line with an arrow at both ends indicates the time limit for the task. If it is not finished in that period, then it is regarded as an error.

In Figure 2, there are three stages in SGSF-064-1. First, before charging occurs, the EV and EVSE need to communicate so that they are ready for charging. Physical contact and CAN communication are established in this stage. Initial physical contacts via wired signal are the Pilot and Wake-up signals. The Pilot signal gives a reference for the communication and Wake-up is to alert the EV for the physical connection. Second, during charging, the EV and EVSE exchange basic messages on their status for smooth power transfer. Lastly, once the EV is full of charge, it sends notice to the EVSE and they start terminating the charging process. At that point, CAN communication is disabled, the connector is unlocked, and the wake-up signal is also disconnected.

6. Testing System

This section describes the structure of the testing system in Figure 3. The testing system consists of a PC and electric load, and both of them together represent the EV in a real-world charging process. The EVSE, the EV charger which is the system under test (SUT), is connected to the testing system. The EVSE has two pins for power delivery to the electric load, two for CAN communication via the CAN bus, and two for the Pilot and Wake-up signals, respectively. Especially CW-DIO32 [18], an I/O board, is utilized for communication of the Pilot and Wake-up signals between the EVSE and the PC as shown in Figure 2.

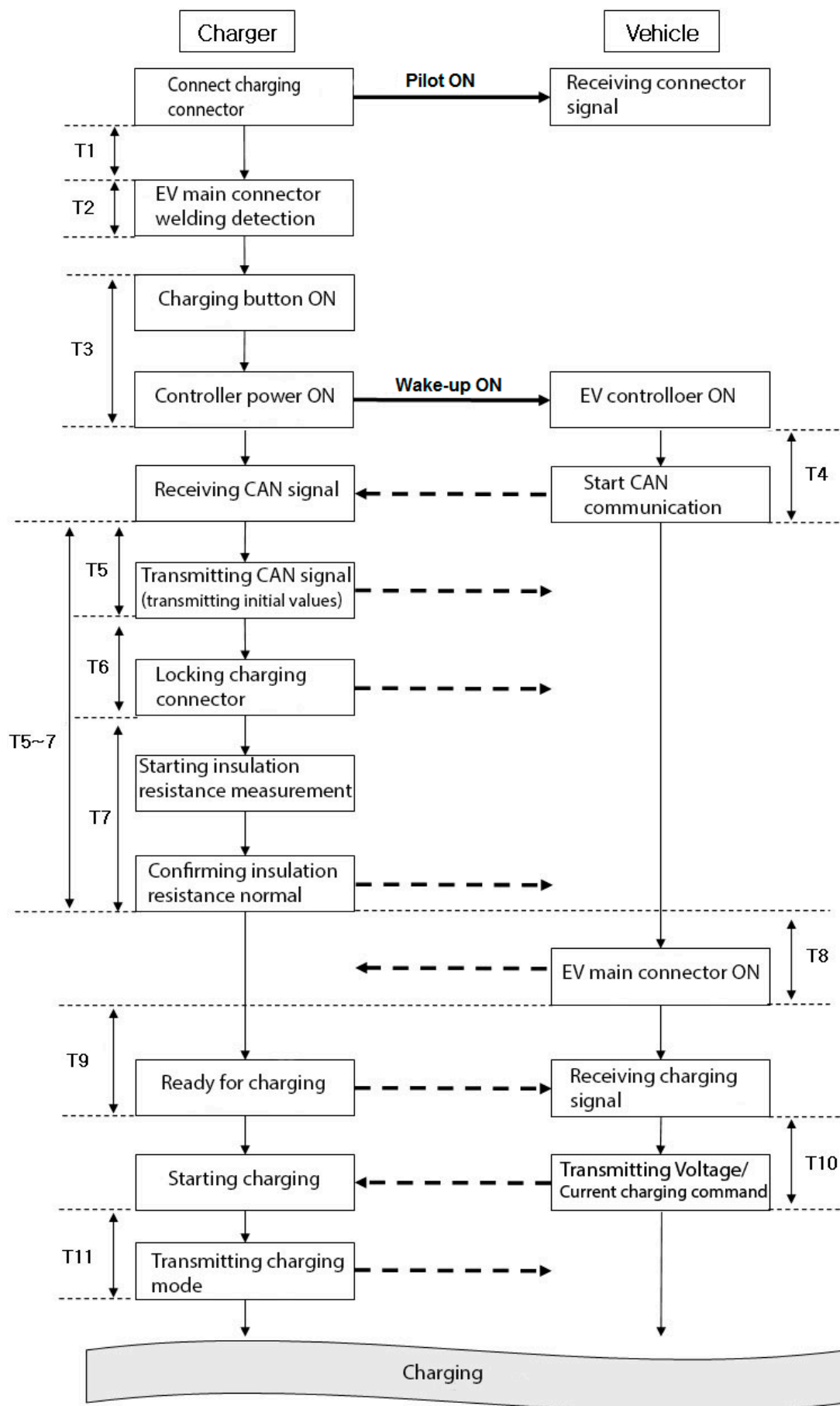


Figure 2. Cont.

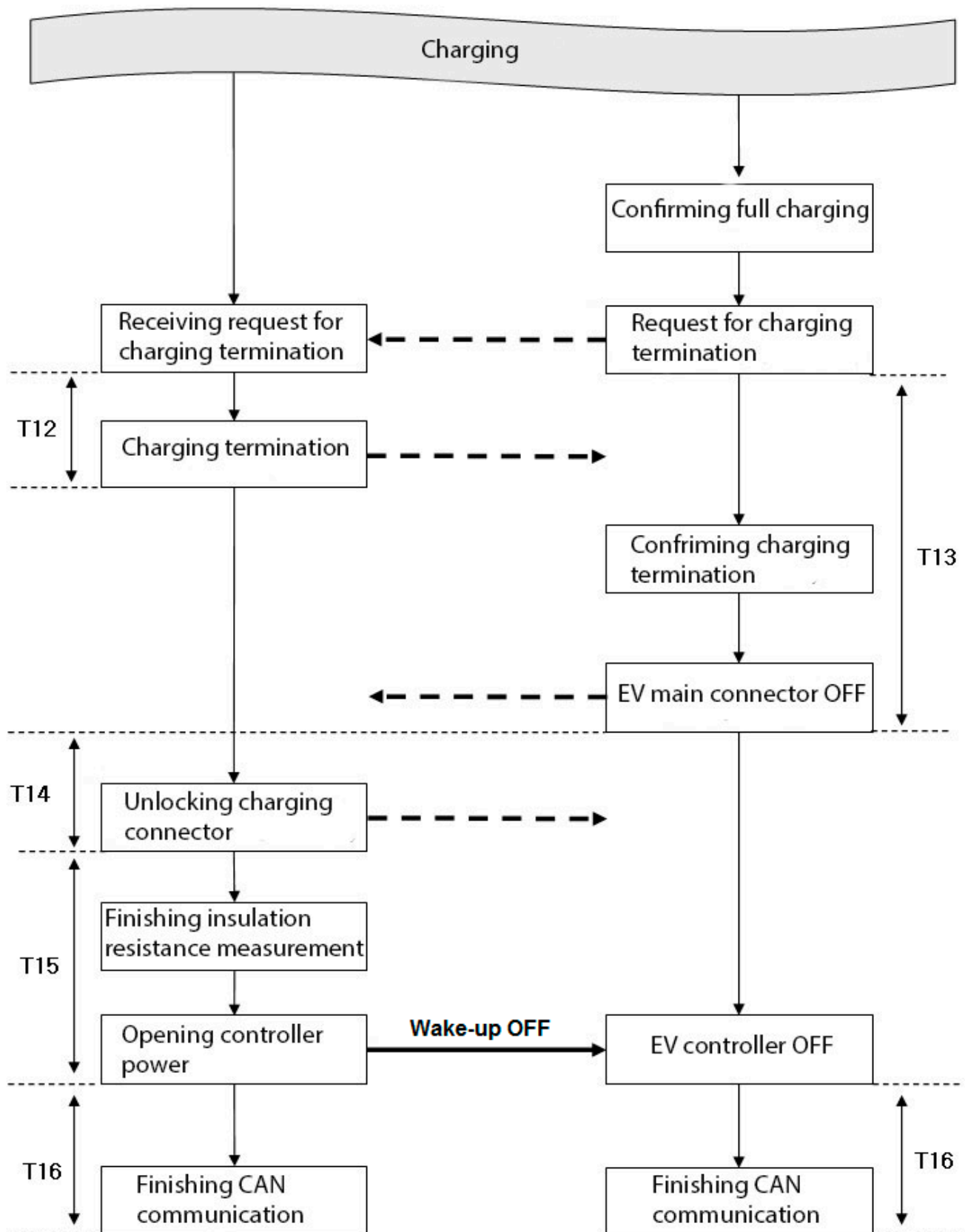


Figure 2. Flow diagram of a normal charging sequence.

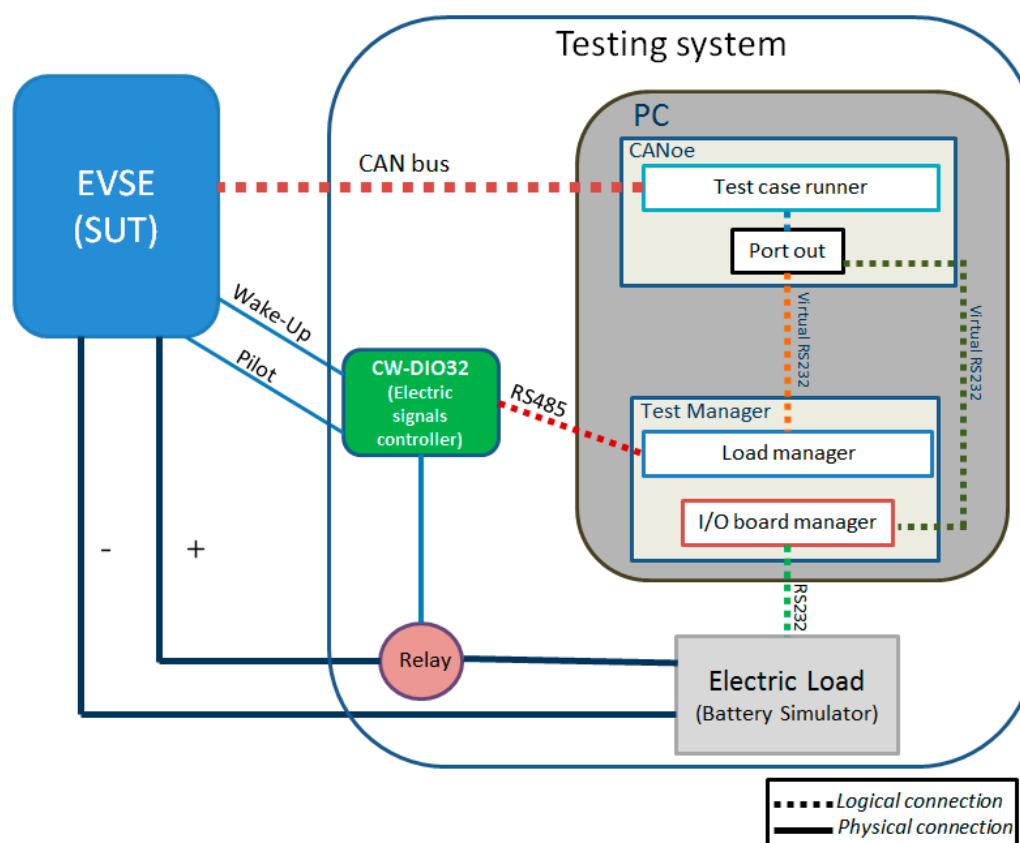


Figure 3. Physical and logical connection between components in the testing system.

CW-DIO32 uses RS-485 and Modbus protocol in connecting with other devices. It works with 12–24 VDC and has 16 input and 16 output points with four commons. In addition, the relay coil, which enables power transfer from the EVSE to the load, is connected to the output port of the board. The PC runs as a tester, monitoring and administering the whole testing procedure. It also manages CW-DIO32 and the electric load. Inside the PC, it contains two applications: CANoe and test manager.

CANoe is the main part of the testing system; it is the only part that has the privilege of giving orders and processing the received data. In contrast, the rest of the equipment is subjected to the CANoe instructions. CANoe runs a program written in CAPL. This program has two parts: the test case runner and port out. The test case runner performs a set of test cases while the port out is responsible for connecting CANoe with the RS232 bus whenever it is required. CANoe is connected directly to the EVSE through the CAN bus and to the test manager in the PC via the virtual RS232 bus.

The test manager is an application written in C#. It handles the electric load and the CW-DIO32 board. Therefore, it contains the load manager and the I/O board manager as well. The load manager provides the load with an interface to configure it and to switch the charging modes of the electric load between constant volt (CV) and constant current (CC). It also monitors the input voltage of the electric load. On the other hand, the I/O board manager is used to control and monitor the CW-DIO32 manager. It enables us to observe the Pilot and Wake-up signals, and the status of the relay coil. During the testing, real components have been used. The EVSE is manufactured by PNE solution, Co, Ltd., Gyeonggi-do, and the DC electronic load by Chroma Systems Solutions. The testing system has been developed and tested in the KTL (Korean Testing Laboratory) as shown in Figure 4.



Figure 4. The testing system components.

The system secures the electric power connection using a relay. By default, the relay is set to off unless it gets a command from CANoe to turn on. As shown in Figure 3, CANoe has a virtual RS232 to communicate with the test manager which, in turn, controls the relay through CW-DIO32 via the RS485 bus. In cases where there is an error, such as getting a fault message or the test device CAN controller is forced into a bus off state, CANoe switches the relay off to save the system from being at risk.

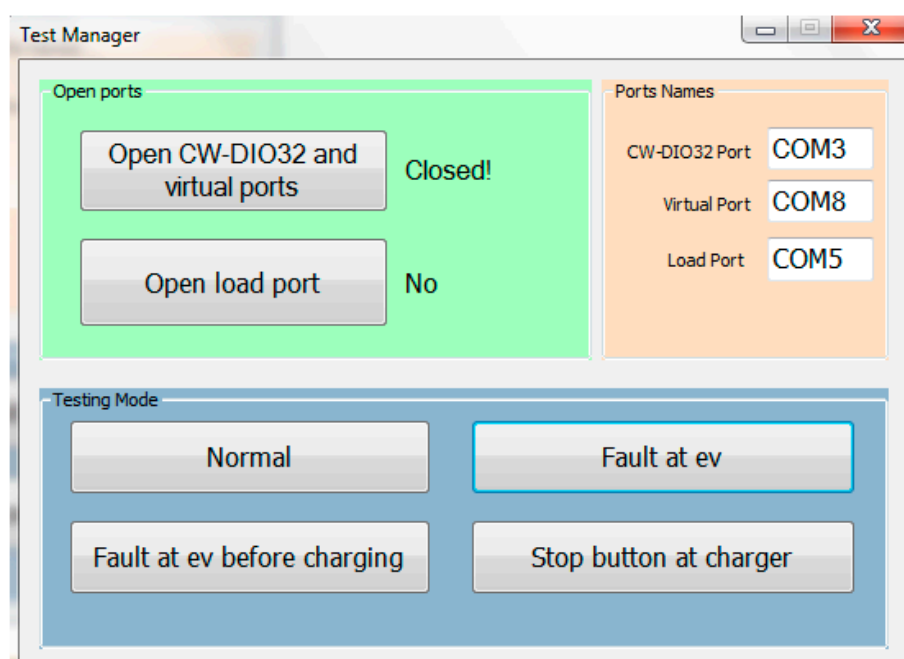
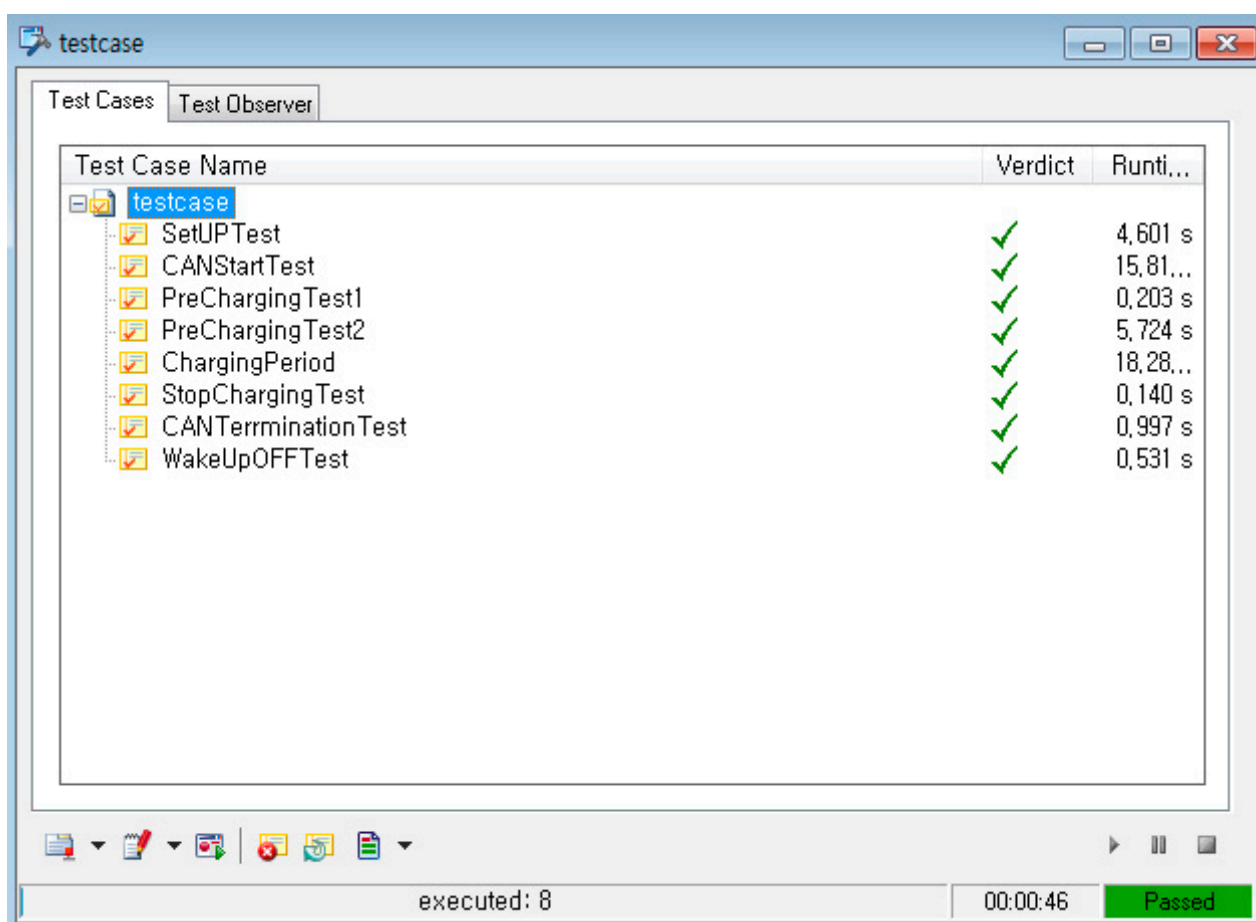


Figure 5. Main window of the test manager.

Conformance testing of EVSE can be performed in the test manager by choosing a desired testing mode. When the testing start button is pressed, CANoe begins running the testing process. The set of test cases is extracted from the SGSF-064-1 and written in CAPL. There are four testing modes or scenarios as depicted in Figure 5. The following are the testing modes.

6.1. Normal Charging

Normal charging is reserved for testing whether the EVSE can properly respond to the EV during the entire period. This testing procedure goes through eight small tests called test cases. Since it is for the conformance testing of the EVSE, it is assumed that the EV works ideally. The whole test ends when CAN communication is disconnected without any interruption or failure from either party. Figure 6 shows the results after applying the normal charging mode; the first column, Test Case Name, lists the test cases' names that come under the test, while the second column, Verdict, shows the final result of each test case. The last column, Runtime, displays the time consumed by each test case.



Test Case Name	Verdict	Runtime
testcase		
SetUPTest	✓	4,601 s
CANStartTest	✓	15,81...
PreChargingTest1	✓	0,203 s
PreChargingTest2	✓	5,724 s
ChargingPeriod	✓	18,28...
StopChargingTest	✓	0,140 s
CANTerminationTest	✓	0,997 s
WakeUpOFFTest	✓	0,531 s

executed: 8 00:00:46 Passed

Figure 6. Test runner in CANoe after successfully testing the normal charging mode.

Figure 7 shows the graphic window in CANoe after successfully testing the normal charging mode. This window shows five signals coming from the EV during testing: CF_Qc_ConLock for connector status, CF_Qc_PowEnaStat for enabling power status, CR_Qc_ChargerMod for specifying the charging mode, CR_Qc_OutVolt for the output voltage of the EVSE multiplied by 10, and CF_BMS_AbnorChg for BMS (battery management system) faults.

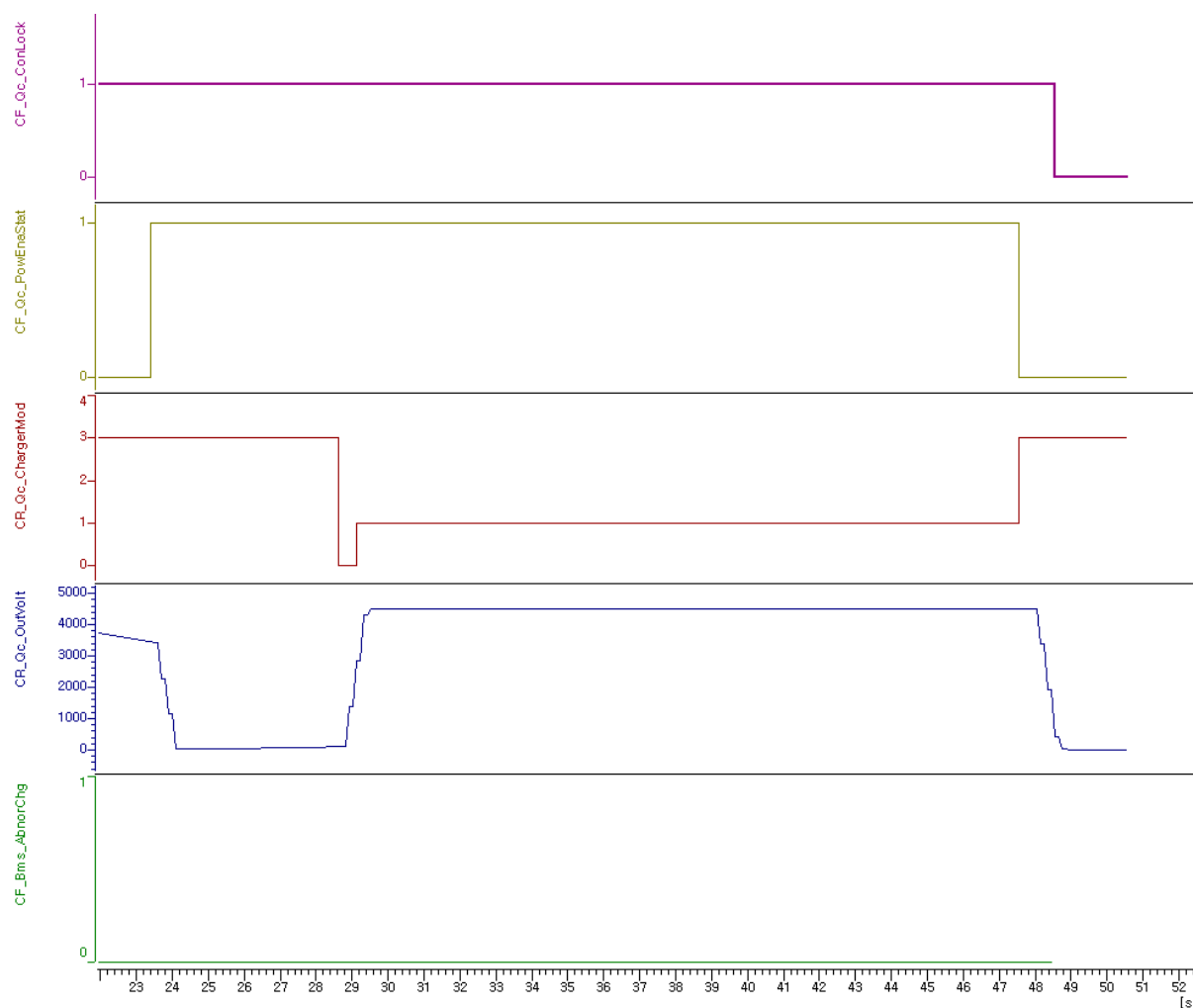


Figure 7. Graphic window in CANoe after successfully testing the normal charging mode.

A part of the final report generated by the system is shown in Figure 8. This report gives several details about the testing process such as the beginning and the ending time of each test case. The entire events are also shown in this report; in PreChargingTest2 we can see when the test system gets into CC and CV mode. It is a very important and crucial thing to know exactly at which step there is a problem. Also, in the ChargingPeriod test case the system lists all the abnormal out voltage values coming from the EVSE. This is done by reading the input voltage from the electric load and comparing it with the output voltage coming from the EVSE through the CAN bus.

6.2. Fault at EV

It is plausible to assume that there might be an error in the EV during the charging period. If it occurs, then the charging process should end immediately and safely. To test this capacity of the EVSE, the fault at car option is reserved for the test manager. To implement this procedure, the testing system simulates the terminating charging request from the EV side and the EVSE is monitored to see whether it responds properly to the request so that both the EV and EVSE are disconnected physically and electrically.

4 PreChargingTest2: Passed

Test case begin: 2015-01-13 16:41:37 (logging timestamp 39.329111)
 Test case end: 2015-01-13 16:41:42 (logging timestamp 44.942227)

Main Part of Test Case

Timestamp	Test Step	Description	Result
44.541412	Resume reason	Resumed on signal 'Signal:CAN1/PowerTrain/MSG631/CR_Qc_ChargerMod' Elapsed time=5212.3ms (max=10000ms)	-
44.541412	1	Get Mode - CC	pass
44.942227	Resume reason	Resumed on signal 'Signal:CAN1/PowerTrain/MSG631/CR_Qc_ChargerMod' Elapsed time=400.815ms (max=10000ms)	-
44.942227	1	Get CV mode	pass

5 ChargingPeriod: Passed

Test case begin: 2015-01-13 16:41:42 (logging timestamp 44.942227)
 Test case end: 2015-01-13 16:41:53 (logging timestamp 55.942227)

Main Part of Test Case

Timestamp	Test Step	Description	Result
45.942227	Resume reason	Elapsed time=1000ms (max=1000ms)	-
45.942227	1		pass
55.942227	Resume reason	Elapsed time=10000ms (max=10000ms)	-
55.942227	1		pass

6 StopChargingTest: Passed

Test case begin: 2015-01-13 16:41:53 (logging timestamp 55.942227)
 Test case end: 2015-01-13 16:41:54 (logging timestamp 56.064674)

Main Part of Test Case

Timestamp	Test Step	Description	Result
56.064674	Resume reason	Resumed on signal 'Signal:CAN1/PowerTrain/MSG631/CF_Qc_ChgFinished' Elapsed time=122.447ms (max=10000ms)	-
56.064674	1	Charging stopped	pass

Figure 8. Part of the final report after successfully testing the normal charging mode.

6.3. Fault at EV before Charging

Car fault can occur before the charging starts. When there is something wrong with the EV, the CAN communication needs to be disconnected. The testing system generates a request to the EVSE before the beginning of the charging period and this fault should interrupt the charging process.

6.4. Pressing the Stop Button

When there is a need to stop the charging, the user should be able to terminate the charging process by pressing the stop button on the EVSE during the charging period. This testing mode checks whether the EVSE ensures the interruption of the charging period immediately and safely. When the stop button is on, terminate charging request is sent by the EVSE to the EV.

Figure 9 shows the procedure of pressing the stop button on the EVSE. It consists of seven test cases as shown in Figure 10 and the last three test cases are found in Figure 9. Each test case contains one or more transmissions of messages so that they, as a whole, accomplish a specific mission.

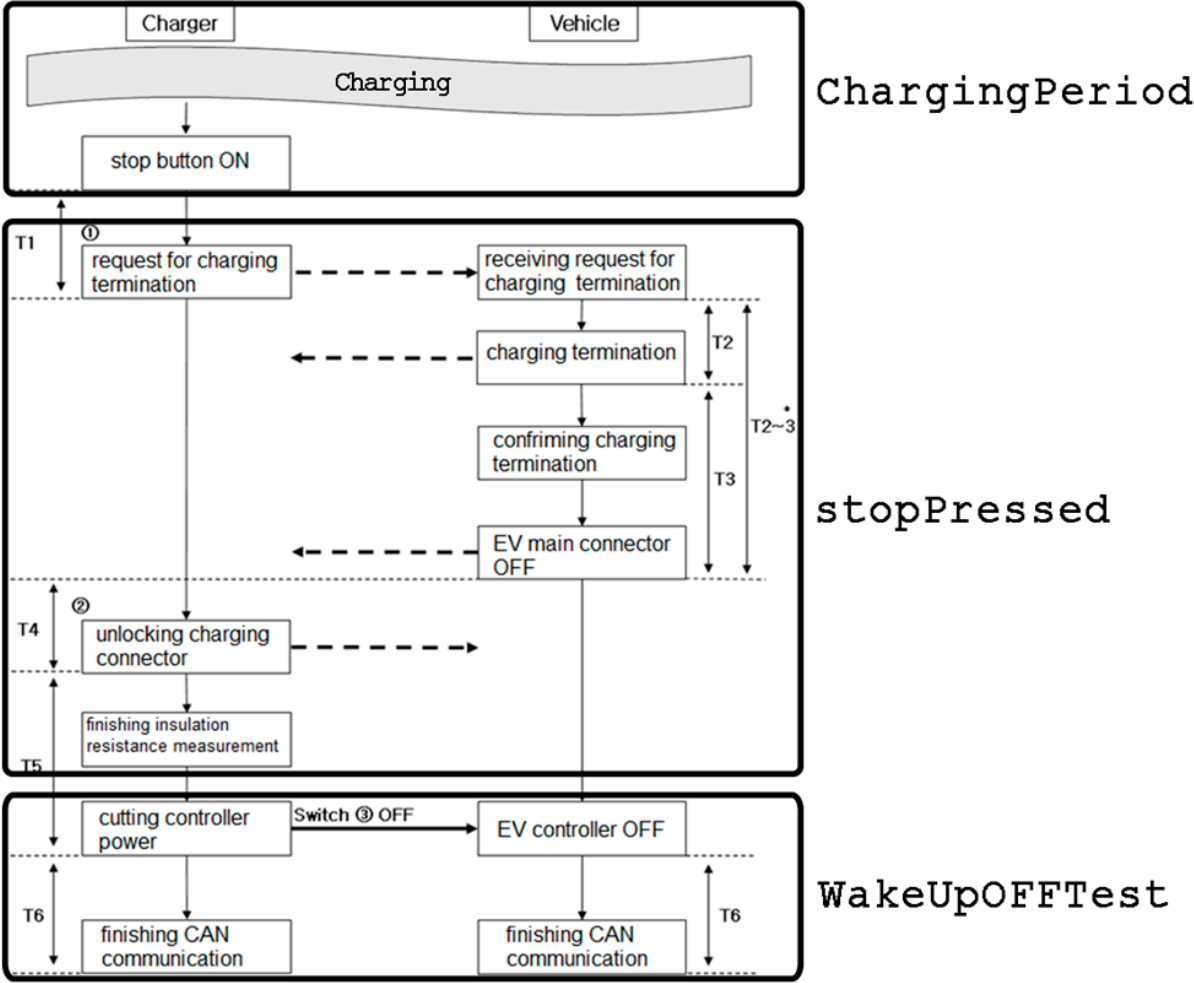


Figure 9. Pressing the stop button test mode.

Test Cases		Test Observer	
Test Case Name		Verdict	Runti...
testcase			
SetUPTest		✓	4,555 s
CANStartTest		✓	18,22...
PreChargingTest1		✓	0,202 s
PreChargingTest2		✓	5,616 s
ChargingPeriod		✓	4,211 s
stopPressed		✓	3,026 s
WakeUpOFFTest		✓	0,592 s

Figure 10. Test runner in CANoe after successfully testing the press stop button mode.

The test of pressing the stop button can be done in the middle of charging. Doing so requires waiting until the charging period begins. In other words, the “ChargingPeriod” is interrupted by pressing the stop button as shown in Figure 10. The ChargingPeriod normally ends when the EV is charged up to 80% of the SOC (state of charge). However, the event “stopPressed” causes the EVSE to send a request message to the EV to terminate the charge. The stopPressed is followed by the WakeUpOFFtest, which releases a wired connection between the EV and EVSE.

The testing results can be observed in the test runner in CANoe during and after the testing process is complete. The test runner shows the final verdict and the running time of individual test cases as shown in Figure 10. At the end of the test, a full report is generated by CANoe to give an overview of the EVSE status and it shows the outcomes of the test cases in detail.

7. Conclusions and Future Research

This paper implemented a conformance testing system for SGSF-064-1 based on CANoe software. The testing system can monitor the communication of messages between EVs and EVSEs, electrical signals, and the level of voltage or current associated with a proper transfer of power. It also included four testing modes described in the standard to check the behavior of the EVSE, which is the system under test. Through four testing modes of the charger, the developed system has been proven to be an effective and efficient tool for testing the EVSE. In terms of further research, it is imperative to expand the testing system in two directions. First, the system should cover every possible scenario of abnormal termination, including edge cases, which could occur in the EVSE by extensive testing and validation. In this study, only three cases were considered. Secondly, it is also highly recommended to build a conformance testing system for EVs in contrast to the one utilized for EVSEs in this paper.

Acknowledgments

This work was supported by the Power Generation & Electricity Delivery Core Technology Program of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) granted financial resource from the Ministry of Trade, Industry & Energy, Republic of Korea (No. 20131020400780).

Author Contributions

Intaek Kim contributed to the research idea and Ahmed Al-Hilo implemented the idea. Hyuk Soo Jang supervised the research and Jong-Geol Yoo provided the facility for the experiment with comments. All authors contributed to the writing of the paper.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Falvo, M.C.; Sbordone, D.; Bayram, I.S.; Devetsikiotis, M. EV Charging stations and modes: International standards. In Proceedings of the International Symposium on Power Electronics, Electrical Drives, Automation and Motion, Ischia, Italy, 18–20 June 2014; pp.1134–1139.

2. Kim, I.; Awan, T.W.; Abed, A.M.; Jang, H.S. Conformance testing for SGSF-064-1. *Appl. Mech. Mater.* **2014**, *598*, 704–708.
3. Awan, T.W.; Abed, A.M.; Kim, I.; Jang, H.S.; Shin, M. CAN based conformance testing using TTCN-3. *Int. J. Comput. Commun. Eng.* **2014**, *3*, 417–423.
4. Part 10: Conformance test. In *Communication Networks and Systems in Power Utility Automation*, 2nd ed.; IEC 61850; International Electro Technical Commission: Geneva, Switzerland, 2011.
5. Hagwood, C.; Rosenthal, L. Reliability of conformance tests. *IEEE Trans. Reliab.* **2001**, *50*, 204–208.
6. Constant, C.; Jeron, T.; Marchand, H.; Rusu, V. *IEEE Trans. Softw. Eng.* **2007**, *33*, 558–574.
7. Methods for Testing and Specification (MTS); Protocol and profile conformance testing specifications; Standardization methodology. Available online: http://www.etsi.org/deliver/etsi_i_ets/300400_300499/300406/01_60/ets_300406e01p.pdf (accessed on 8 October 2015).
8. Conformance Testing. Available online: <http://www.nist.gov/itl/ssd/is/conformancetesting.cfm> (accessed on 26 June 2015).
9. Renjun, L.; Chu, L.; Feng, L. A design for automotive CAN bus monitoring system. In Proceedings of the Vehicle Power and Propulsion Conference VPPC '08. IEEE, Harbin, China, 3–5 September 2008; pp. 1–5.
10. Qiangsheng, Y. Research and application of CAN and LIN bus in automobile Network System. In Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering, Chengdu, China, 20–22 August 2010; pp. 150–154.
11. *Road Vehicles—Controller Area Network (CAN)—Part 1: Data Link Layer and Physical Signaling; ISO Standard 11898-1, Ed. 12*; International Organization for Standardization: Geneva, Switzerland, 2003.
12. Robert, B. *CAN Specification*, 2nd ed.; Robert Bosch GmbH: Stuttgart, Germany, 1991; pp. 1–72.
13. *Communication Protocol between Electric Vehicle and Conductive DC Charger*; Smart Grid Standardization Forum: Seoul, South Korea, 2014.
14. Jianmin, D.; Hualin D.; Yongchuan Y. Co-simulation of SBW and FlexRay bus based on CANoe_MATLAB. In Proceedings of IEEE International Conference on Automation and Logistics, Hong Kong, China; Macau, China, 16–20 August 2010; pp. 202–206.
15. *User Manual CANoe, Version 7.5*; Vector Informatik GmbH: Stuttgart, Germany, 2010.
16. Heiko, G. *Extended Signal Multiplexing in DBC Databases, Version 1.1*; Vector CANtech, Inc.: Stuttgart, Germany, 2010.
17. *CANoe Test Feature Set Tutorial, Version 2.0*; Vector CANtech, Inc.: Stuttgart, Germany, 2009.
18. *CW-DIO32 User's Manual*; Comfile Technology: Sterling, VA, USA, 2012; Available online: http://www.comfile.co.kr/download/etc/cwdio32_manual.pdf (accessed on 27 October 2015).