

Extension of Drive Recorder Function of In-vehicle ECU and Record Information Collection Technique

Hirofumi OHBUCHI*

ABSTRACT

Two of the drive recorder functions of the engine control unit were extended. First, a function was added that can use the vehicle diagnostic tool to freely generate a conditional expression that will record snapshot data as well as to enable modification of the conditional expression. Several patterns of conditional expressions were prepared in advance so that they could be combined to generate the conditional expressions that assure the capability to record the desired phenomena. Next, the system was configured to allow the snapshot data to be transmitted to a server external to the vehicle. The wireless network functionality of the Display Audio unit is used for transmission to the server, and the snapshot data is transmitted from the engine control unit to the Display Audio unit by divided transmission to enable the large volume of data to be handled for transmission between vehicle units, which are intended to handle small volumes of data. This was also done using simplex transmission in order to limit bus load increase on the transmission line. Measures were taken against data loss, as well, by providing for retransmission of the snapshot data even if the transmission is interrupted due to interruption of the power supply or other such reason.

1. Introduction

The electrification of vehicle devices has been progressing and the software used to control those devices has been growing ever more complex. The increased complexity of engine control, in particular, has been conspicuous, and there are various engine behavior phenomena that users may have occasion to consult dealers about. As a result, a growing necessity exists for detailed analysis of engine behavior.

An engine control unit (ECU) is equipped with control software and on-board diagnostics (OBD)⁽¹⁾ functionality for self-diagnosis of malfunctions in sensors and actuators that are used to control the engine. The OBD is able to identify a malfunctioning area by detecting changes in a sensor's output values when there is a malfunction in the area. In addition to identifying malfunctioning areas, it is also important to identify the causes of malfunction. To identify the causes of malfunction, it is necessary to analyze engine behavior up to the point when the malfunction occurs. For this it would be effective to perform analysis using a drive recorder function with the capability for recording engine behavior in time series. Drive recorder functions are generally configured either to mount a dedicated recorder

tool on the vehicle⁽²⁾ or as software mounted internally in the ECU. Configuration with a dedicated recorder enables long-term data recording that is not dependent on the memory capacity of the ECU. However, the number of dedicated recorders is limited, which means that the number of vehicles that can be analyzed will be limited as well. There is also the possibility that mounting a dedicated recorder will have a psychologically oppressive effect on a user. When the functionality is configured in software mounted internally in the ECU, however, a dedicated recorder is not needed, there is no limit to the number of vehicles on which the functionality can be applied, and there will be less of a feeling of psychological oppression for the user. The constraints imposed by ECU memory capacity, however, will not allow lengthy recording. In light of the above, and considering the factors of portability and psychological burden on users, it would be desirable to have the drive recorder function take the latter configuration.

2. Conceptual Approach to Development

At present, the ECU has a drive recorder function called on-board snapshot (OBS) that is built-in as software. The OBS can take a malfunction or other such event as its

* Automobile R&D Center

trigger to extract time-series data about engine behavior (snapshot data) that has been collected over a certain period of time preceding the trigger and retain that snapshot in the ECU (Fig. 1). However, the present configuration faces issues such as the following. The first is that the trigger condition that is the starting point for retaining snapshot data in the ECU has to be written into the software in advance. As a result, the present configuration cannot be used except for analysis of malfunctions that were already known and recognized at the time the software was developed, and its applications are therefore limited. The second is that implementation of this functionality requires the buffer that records engine behavior to have relatively large memory capacity. Increasing the memory capacity contributes to increased ECU cost, and this means that new trigger conditions cannot readily be defined and further snapshot data added every time there is an increase in the phenomena for analysis. Given these circumstances, it would be desirable for the OBS to be capable of handling analysis of a variety of phenomena without increasing the memory capacity.

Another factor is that in the present configuration, snapshot data can only be read using a diagnostic tool. Consequently, carrying out troubleshooting using snapshot data necessarily requires the user to go to a dealer. If replacement parts are needed, users must return to the dealer, so there is a time burden on the user. It would be a great advantage for users if snapshot data could be read before users go to the dealer, so that the diagnosis could be made in advance and the repair completed with one visit to the dealer.

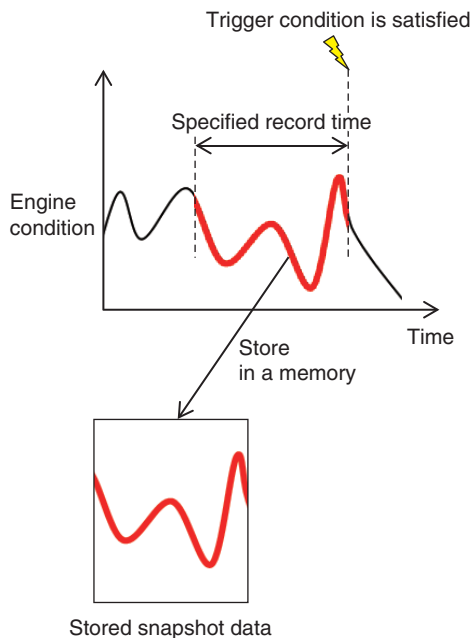


Fig. 1 Conceptual image of drive recorder function

3. Development Goals

The development goals are summarized below.

- (1) To create an OBS that enables later modification of a trigger condition expression by means of a diagnostic tool
- (2) To devise a configuration that enables the snapshot data retained in the ECU to be read by a data server external to the vehicle using a wireless network (Fig. 2)

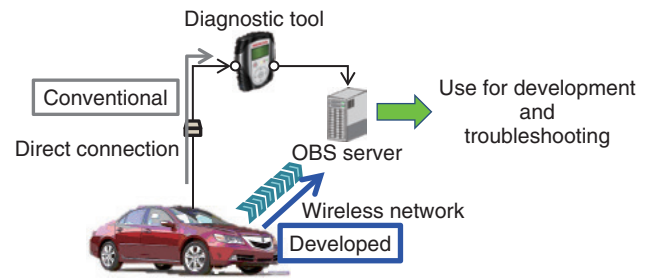


Fig. 2 How snapshot data is collected

4. OBS Configuration that Enables Modification of Trigger Condition Expressions

In order to acquire useful snapshot data by means of OBS, it is important to be able to create detailed trigger condition expressions. In a conventional OBS, the trigger condition expression is written into the software in advance in order to enable definition of a conditional expression that is optimal for grasping the phenomenon. Even when defining a trigger condition expression from a diagnostic tool, a configuration that enables defining a trigger condition expression in detail equivalent to the existing OBS is needed. The configuration proposed in this paper enables modification of the trigger condition expression by means of the diagnostic tool, and this therefore involves having the component elements of a number of conditional expressions built into the software in advance. Then one of those component elements can be selected and populated with variables, threshold value, equal sign, unequal sign, and so on, from a diagnostic tool. This generates the desired trigger condition expression. Combining a number of these component elements can create an even more complex conditional expression. An outline of the generation of trigger condition expressions using a diagnostic tool is shown in Fig. 3. The next section will explain the component elements of those trigger condition expressions.

4.1. Configuration of a Trigger Condition Expression

Figure 4 shows the overall trigger condition configuration. Trigger condition expressions are configured from conditional expressions 1 to 6, and the occurrence

of phenomena satisfying each conditional expression determines the timing for snapshot data to be held in the ECU. Conditional expressions 1 to 6 are configured variously from the first and second conditional expressions, and one each of Expression Type 1 to Type 4 can be selected for the first and second conditional expressions. Expression Type 1 to Expression Type 4 are operation

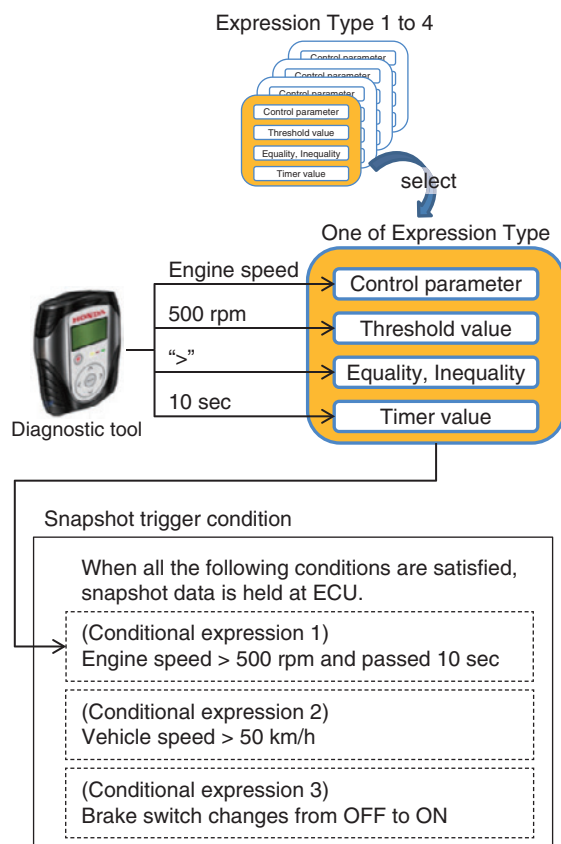


Fig. 3 How to set up trigger condition using diagnostic tool

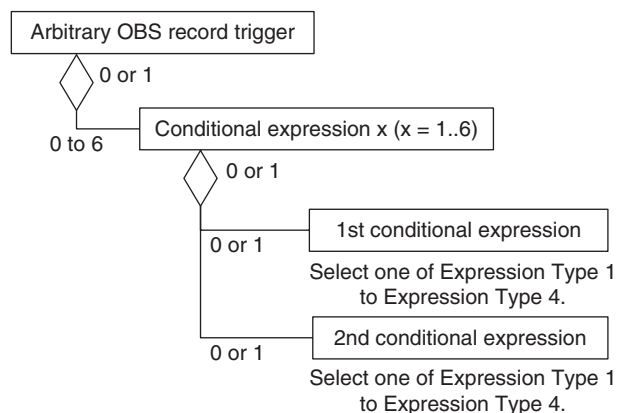


Fig. 4 Entire configuration of arbitrarily selectable OBS record trigger condition

Table 1 Expression type setting

How to judge	Conditional expression	Example of setting
Expression Type 1		
Parameter and threshold	Parameter □ Threshold	Current engine speed > 0 rpm
Expression Type 2		
Each of parameters	Parameter1 □ Parameter2	Current engine torque > Target engine torque
Absolute	Parameter1 - Parameter2 □ Threshold	Current engine speed - Target engine speed ≥ 100 rpm
Expression Type 3		
Parameter edge	Parameter edge (Up edge or down edge)	Brake switch changes from OFF to ON
Expression Type 4		
Many parameters by logical decision	Many parameters logical judgment (Judged by logical decision)	Satisfy any of the conditions below (logical decision = "or") Perform transmission control, air conditioning control, or meter control

* □ is equality or inequality

expressions that use equal signs, unequal signs, or logical operators. Execution of the first and second conditional expressions is then determined in accordance with the selected expression type. The number of conditional expressions and the degree of detail of the configurable items were decided with reference to known phenomena.

4.1.1. Definition of expression types

The items that can be defined as Expression Type 1 to Expression Type 4 are shown in Table 1. The defined content of each expression type is shown below.

- (1) Expression Type 1: Makes conditional judgment according to the relationship between the control parameter and the threshold value. The relationship between the control parameter and the threshold value is judged by an equal sign or unequal sign. For example, this type will be defined when an engine speed greater than 0 rpm is the trigger condition.
- (2) Expression Type 2: Makes conditional judgment according to the relationship between control parameters. The conditional judgment can be made by two methods. The first judges the relationship between control parameters by an equal sign or unequal sign. For example, this type will be defined when an increase in engine torque beyond a target engine torque is the trigger condition. The second makes the conditional judgment by the absolute value of the difference between control parameters and by the threshold value. The relationship between the absolute value and the threshold value is judged by an equal sign or unequal sign. For example, this type will be defined when the difference between the engine speed and the target engine speed reaching 100 rpm or more is the trigger condition.
- (3) Expression Type 3: Makes conditional judgment by when the control parameter passes above or below the threshold value. The control parameter can be defined as passing down from above the threshold value to a level below it or as passing up from below the threshold value

Table 2 Relation setting of expressions

How to judge	Condition	Example of setting
Relation of 1st conditional expression and 2nd conditional expression		
Logical decision	And	Engine speed > 0 rpm or Vehicle speed > 0 km/h (Example of setting that condition is "or")
	Or	
	Xor	
Decision order	None	Brake switch changes from OFF to ON under the condition of "Accelerator pedal > 0 %" (Example of setting that condition is "1st conditional expression to 2nd")
	1st conditional expression to 2nd	
	2nd conditional expression to 1st	
Relation of conditional expression x (x = 1 to 6)		
Logical decision	And	Satisfy all conditions of the below (logical decision = "AND") Conditional expression 1 and Conditional expression 2 and ... and Conditional expression 6
	Or	
Decision timer	Decision delay timer of OBS trigger	When 10 sec pass after "logical decision," OBS trigger condition is satisfied.

to a level above it; either can be selected. For example, this type will be defined when the brake switch changing from OFF to ON is the trigger condition.

- (4) Expression Type 4: Makes conditional judgment by the logical sum or logical product of multiple control parameters. For example, this type will be defined when the trigger condition occurs during the implementation of transmission control, the implementation of air conditioning control, or the implementation of meter control.

In addition to triggering by the time of occurrence of a phenomenon, triggering by the number of times a phenomenon occurs has also been enabled for each expression type. This is to enable the prevention of mistaken recognition of a trigger condition as well as to allow evaluation by the number of times or the frequency of occurrence of a phenomenon.

4.1.2. Definition of relationships between conditional expressions

The items that can be defined as relationships between the various conditional expressions are shown in Table 2. For conditional expressions 1 to 6, logical decisions between conditional expressions are enabled. The relationship between the first and second conditional expressions allows for logical decisions between conditional expressions as well as for definition of the order in which conditional expressions are satisfied. This allows for the capture of phenomena that are limited, such as those that occur under certain conditions. The configuration also allows the timing for retention of snapshot data to be delayed by a defined time from the point when the trigger condition is satisfied. This enables retention of snapshot data that shows the behavior of the control parameter after the trigger condition

has been satisfied, and also allows additional freedom in defining conditions.

4.2. Results from Confirmation of Operation

Figure 5 shows the results for confirmation of operation when snapshot data is acquired through a trigger condition defined using a diagnostic tool. As a configuration example, a conditional expression was defined so that the trigger condition occurs when the vehicle is moving at 30 km/h or faster and when the accelerator pedal is released and then the brake pedal is pressed. As a result, the snapshot data was successfully acquired by means of the defined trigger condition.

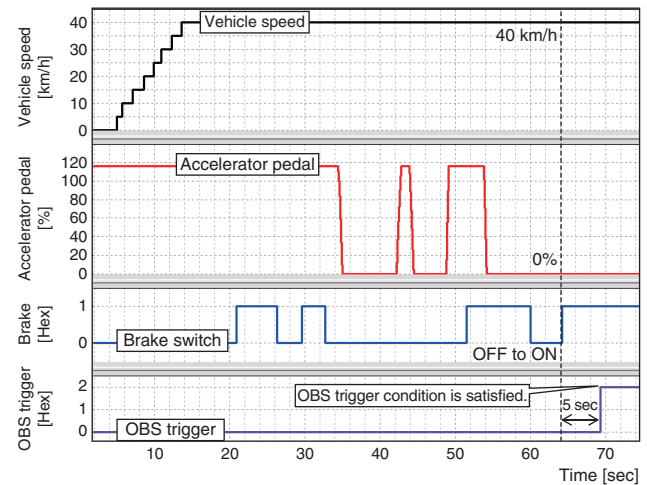


Fig. 5 Results of OBS trigger set up by diagnostic tool

5. Snapshot Data Collection Technique Using a Wireless Network

In order to transmit snapshot data to a data server that is external to the vehicle, wireless network functionality is needed. Already-existing functions of the display audio (DA) unit are used for that wireless network functionality. The ECU and DA unit are connected by wire over the controller area network (CAN). Figure 6 summarizes the configuration of the ECU and peripheral equipment. The CAN is connected with a number of units other than the ECU and DA unit, so it is unable to transmit large amounts of data like the snapshot data, which amounts to several kilobytes, all at one time. Therefore the system is configured to send the snapshot data by divided transmission and put it back together in the DA unit. Furthermore, transmission from the ECU to the DA unit is handled by simplex transmission in order to limit the bus load increase on CAN communication lines due to the added snapshot data transmission. The next section will explain the technique for transmitting snapshot data from the ECU to the DA unit.

5.1. Divided Transmission Technique

Communication from the ECU to the DA unit is conducted on the CAN. CAN communications are capable of handling eight data bytes per frame.

Snapshot data sent by divided transmission will then be reconstructed on the DA unit end, and this involves the possibility that noise or other such extraneous signals will be applied to the communication line, resulting in the loss of transmitted data. This is countered by dividing the transmission into header frames and data frames and guaranteeing the transmitted data.

The header frames are used to transmit the data size, checksum, and chunk number of the snapshot data due to be transmitted. The data size and checksum are used to guarantee the integrity of the snapshot data when it is reconstituted. Chunk numbers enable identification of multiple items of snapshot data on the DA unit end. Chunk numbers are allocated to items of snapshot data in order of their retention in the ECU. In the DA unit, the snapshot data and chunk numbers are strung together for management, enabling recognition of additions to or deletions from snapshot data sent by the ECU. In this way, the DA unit will add to, retain, or delete snapshot data being held in the unit as requested by the ECU. Figure 7 shows an example of snapshot data and chunk numbers strung together. This

example shows a new chunk number being allocated for the snapshot data being held in the ECU when the OBS trigger condition has occurred four times. The OBS shown in this example also updates its snapshot data every time the trigger condition occurs, and the snapshot data held in the ECU up to that point is erased. Consequently, the chunk number that had been allocated to the old snapshot data is erased each time the OBS trigger condition occurs.

The data frames are used to transmit the divided snapshot data in six-byte units. When this is done, continuous numbers are assigned as frame counters to guarantee the continuity of the snapshot data. Figure 8 shows the situation when the trigger condition occurs while snapshot data is being held as Chunk 0, so the transmission data is changed when new snapshot data is added and held.

The technique proposed in this paper uses simplex transmission to send snapshot data from the ECU to the DA unit, so the ECU cannot know how much of the snapshot data has been received by the DA unit. Therefore the system was configured to repeat the transmission of snapshot data.

5.2. Guarantee of Data by Rewind Transmission

The ECU and the DA unit use different timing for start-up and shutdown. The ECU has a power source retention function so that the unit can remain operational by ECU

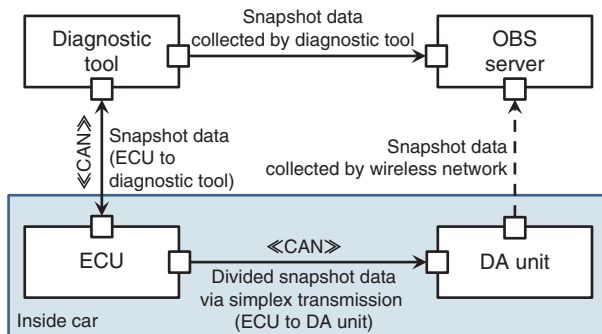


Fig. 6 Entire configuration of system

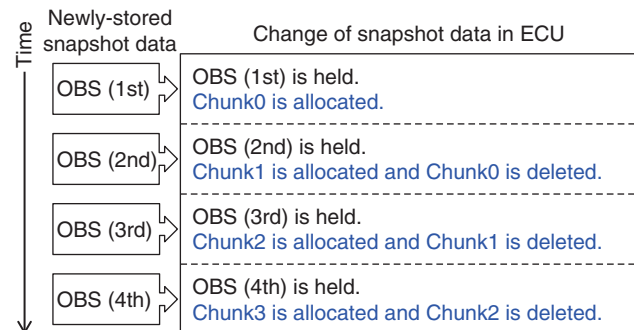


Fig. 7 How to allocate Chunk No. at snapshot

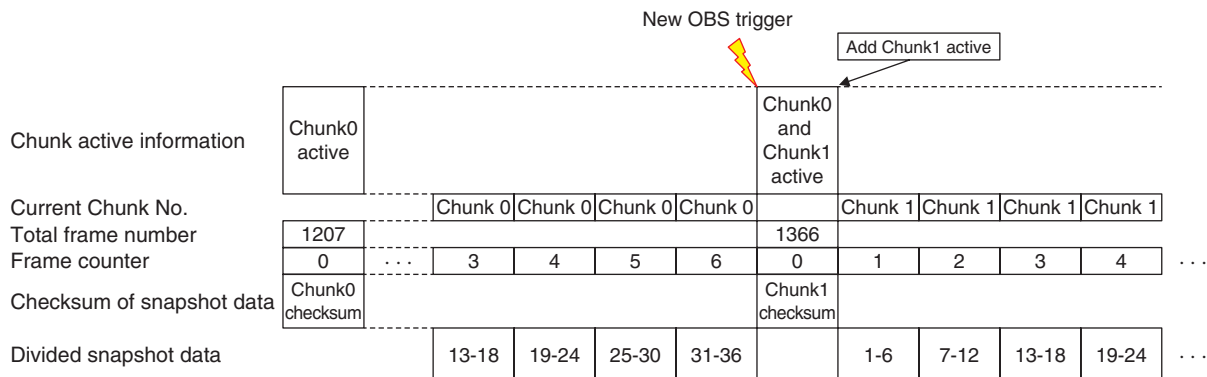


Fig. 8 CAN data management in divided transmission

request even after the ignition has been switched OFF. The DA unit, on the other hand, shuts down immediately after the ignition is switched OFF. The same also applies while the engine is being cranked during start-up, when the DA unit alone shuts down. In this case, the DA unit is unable to receive any data even if the ECU has transmitted snapshot data, so data omission occurs. Therefore, as indicated above, in cases when the DA unit may be shut down because the power supply is cut off due to ignition switch operation, data omission is prevented by causing the frame counter to be rewound by a prescribed number of frames when the DA unit is restarted, and that snapshot data is retransmitted. Figure 9 shows an example of the rewind transmission operation. In the example operation, a DA unit reset is detected while the fifth frame is being transmitted. This example shows the operation of rewind transmission after the restart of the DA unit has been completed, causing retransmission from the fourth frame.

5.3. Results from Confirmation of Operation

Figure 10 shows the results from confirmation of operation when snapshot data held in the ECU is sent to the DA unit by divided transmission. The monitoring results from the CAN communication line were compared with the snapshot data read directly from the ECU by a diagnostic tool. The two are identical, providing confirmation that

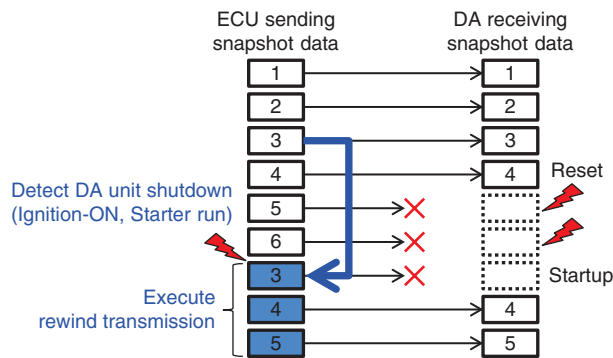


Fig. 9 Rewind transmission

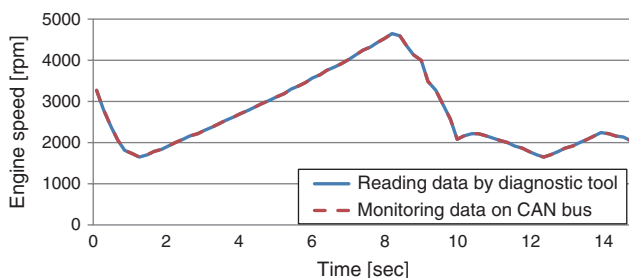


Fig. 10 Result of comparison of data being read by diagnostic tool and being monitored on CAN bus

divided transmission of snapshot data held in the ECU is taking place successfully. Figure 11 shows the results from confirmation of operation when the power supply is cut off due to ignition switch operation. Here the transmission data guarantee by means of rewind transmission was confirmed. It was confirmed that when the ECU detected an ignition operation, the frame counter was rewound from 125 back to 108 and the snapshot data was retransmitted. There is a rewind limit value of 108 on the frame counter during rewind transmission, and this limit is imposed to prevent the frame counter from being rewound more than necessary by successive ignition operations.

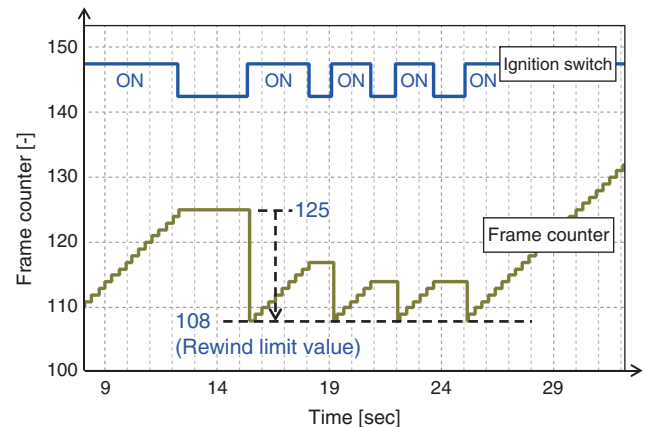


Fig. 11 Result of rewind transmission

6. Conclusion

The following functions were created to extend the drive recorder functions of the in-vehicle ECU.

- (1) An OBS function was created to enable later changes to be made to trigger condition expressions using a diagnostic tool. In order to enable the generation of conditional expressions capable of recording the desired phenomenon, conditional judgment of absolute values, conditional judgment by parameter edge, and other such expressions in addition to conditional judgment of a control parameter and threshold value, were made usable as trigger condition expressions.
- (2) A system was configured with the capability for transferring snapshot data to an external server by means of the wireless network functionality of the DA unit. Transmission from the ECU to the DA unit was achieved using simplex CAN communication. The system was made capable of handling large volumes of data by sending the snapshot data using divided transmission. Measures were taken to perform rewind transmission so that the transmission of snapshot data would be completed even if the DA unit were shut down by ignition switch or other such operation.

References

- (1) <http://www.arb.ca.gov/msprog/obdprog/obdprog.htm>
- (2) Imai, R., Iikuma, H., Kawamura, T., Saito, T.:
Diagnostic Tool “SUBARU Select Monitor III”, Journal
of Society of Automotive Engineers of Japan, Vol. 61,
No. 2, p. 93-97 (2007) (in Japanese)

■ Author ■



Hirofumi OHBUCHI