ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

MSc THESIS

Emrah YİRİK

DESIGN AND IMPLEMENTATION OF AN AUTOMOTIV REMOTE DIAGNOSTIC ELECTRONIC UNIT FOR ROAD VEHICLES

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

ADANA, 2013

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

DESIGN and IMPLEMENTATION OF AN AUTOMOTIVE REMOTE DIAGNOSTIC ELECTRONIC UNIT FOR ROAD VEHICLES

Emrah YİRİK

MSc THESIS

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

We certify that the thesis titled above was reviewed and approved for the award of degree of the Master of Science by the board of jury on 01/06/2013.

…….....	……..……..	……..
Asst. Prof. Dr. Murat AKSOY	Prof. Dr. Mehmet TÜMAY	Assoc. Prof. Dr. Mutlu AVCI
SUPERVISOR	MEMBER	MEMBER

This MSc Thesis is written at the Department of Institute of Natural And Applied Sciences of Çukurova University.
Registration Number:

Prof. Dr. Mustafa GÖK
Director
Institute of Natural and Applied Sciences

Not: The usage of the presented specific declerations, tables, figures, and photographs either in this thesis or in any other reference without citiation is subject to "The law of Arts and Intellectual Products" number of 5846 of Turkish Republic

ABSTRACT

MSc THESIS

DESIGN and IMPLEMENTATION OF AN ELECTRONIC REMOTE DIAGNOSTIC UNIT FOR ROAD VEHICLES

Emrah YİRİK

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

Supervisor : Asst. Prof. Dr. Murat AKSOY
Year: 2013, Pages: 90
Jury : Asst. Prof. Dr. Murat AKSOY
: Prof. Dr. Mehmet TÜMAY
: Assoc. Prof. Dr. Mutlu AVCI

The aim of this project is to design and improve a Centrically Acquisited Data and Fault Management System. The system will provide to collect and store the data and faults on the vehicle, to send these data to a remote centre, to response to vehicle from the remote centre when required and to integrate this system to road vehicles. An electronic card and a firmware which can communicate with the vehicle CAN (Controller Area Network) and read the faults of the vehicle, an electronic card which can get the location of the vehicle through GPS (Global Positioning System), an electronic card and a firmware which can send these data to a remote centre through GPRS is designed. These electronic cards are integrated and adapted to the vehicle.

Keywords: Centrically Controlled Data, Fault Management System, CAN, GPS & GPRS, Remote Fault Code Sending

ÖZ

YÜKSEK LİSANS TEZİ

DESIGN and IMPLEMENTATION of AN ELECTRONIC REMOTE DIAGNOSTIC UNIT FOR ROAD VEHICLES

Emrah YİRİK

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK ANABİLİM DALI

Danışman : Yrd. Doç. Dr. Murat AKSOY
Yıl: 2013, Sayfa: 90
Jüri : Yrd. Doç. Dr. Murat AKSOY
: Prof. Dr. Mehmet TÜMAY
: Doç. Dr. Mutlu AVCI

Bu proje ile araç üzerinde bulunan motor, şanzıman, fren sistemleri ve elektrik aksam gibi donanımlara ait bilgilerin ve bu donanımlarla ilgili oluşabilecek hataların araç üzerinde toplanmasını sağlayacak, toplanan bilgilerin uzakta bulunan bir merkez üzerine iletilebilmesini ve gerektiği durumlarda uzaktan araca müdahale edilmesini sağlayacak bir merkezden veri ve arıza yönetimi sistemi geliştirilip ticari araçlarına entegre edilerek hayata geçirilmesi amaçlanmaktadır.

Bu proje kapsamında araç CAN (Controller Area Network) ile haberleşip bilgi ve arızaları okuyabilen bir elektronik kart ve gömülü yazılım tasarımı çalışması, GPS (Global Positioning System) üzerinden araç lokasyonunu alabilecek bir kart ve yazılım tasarımı çalışması, bu bilgileri GPRS üzerinden uzaktaki bir merkeze iletecek bir kart ve yazılım tasarımı çalışması yapılacaktır. Tasarlanan elektronik kartların entegrasyonu ve araca adaptasyonu çalışılacaktır.

Anahtar Kelimeler: Hata tespiti, *CAN*, GPS & GPRS, hata mesajı gönderimi

ACKNOWLEDGEMENTS

I am heartily thankful to my supervisor, Asst. Prof. Dr. Murat AKSOY, whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the subject.

I would like to send my greatest thanks to TEMSA GLOBAL A.Ş. for their support, advice and time.

My wife Selma and my daughter Pınar have been, always, my pillar, my joy and my guiding light, and I thank them.

CONTENTS **PAGE**

ABSTRACT I
ÖZ II
ACKNOWLEDGEMENTS III
CONTENTS IV
LIST OF FIGURES VI
LIST OF TABLES VIII
1. INTRODUCTION 1
2. LITERATURE REVIEW 5
2.1. Previous Studies about Remote Communication 5
2.2. Previous Studies about Vehicle Data Bus 7
2.3. Previous Studies about ARM and CAN network relationship 14
3. MATERIAL AND METHODS 17
3.1. Vehicle Electronic Control System 17
3.2. CAN Communication Protocol 19
3.2.1. CAN Layer model 19
3.2.2. CAN Node 20
3.3. Sae J1939 Can Bus Standard For Automotive 22
3.3.1. Introduction 22
3.3.2. Message Format and Usage (J1939/21) 22
3.3.3. Transmitting Messages (J1939/21 and J1939/7x) 24
3.3.3.1. TP_BAM messages 25
3.3.3.2. TP_CM messages 25
3.3.4. Receiving Messages (J1939/21 and J1939/7x) 26
3.3.5. ECU Design (J1939/1x, J1939/21, and J1939/7x) 26
3.3.6. Wiring Topology - Physical Layer (J1939/1x) 27
3.3.7. J1939 message format 29
3.3.8. J1939 set of profiles 30
3.4. More about J1939-73 Application Layer (Diagnostics) 34
3.4.1. Diagnostic Trouble Code Definition 34

3.4.2. Active Diagnostic Trouble Codes (Dm1) 36
3.4.3. Malfunction Indicator Lamp 40
3.4.4. Red Stop Lamp 40
3.4.5. Amber Warning Lamp 40
3.4.6. Suspect Parameter Number 41
3.4.7. Failure Mode Identifier 42
3.5. Vehicle Diagnostic System 43
3.5.1. Engine Diagnostic System 43
3.5.2. Brake Control System Diagnostic 49
3.5.3. Suspension Control System (ECAS) Diagnostic 52
3.5.4. Transmission Control System (TCU) Diagnostic 54
3.5.5. Vehicle Diagnostic Message Configuration 58
3.5.6. The ARM Controller 59
3.5.7. How the Diagnostic ECU is connected to CAN bus of the vehicle? 61
3.5.8. How the Diagnostic ECU is connected to GPRS modem? 62
3.5.9. How Diagnostic ECU determines the error codes? 63
3.5.10. How Diagnostic ECU sends the error codes to the remote server? 65
4. RESULTS AND DISCUSSION 67
5. CONCLUSION 75
REFERENCES 77
BIOGRAPHY 81
APPENDIX 82
FMI and DESCRIPTIONS (SAE J1939, 2000) 83

LIST OF FIGURES **PAGE**

Figure 1.1. Topology of the REDI System 3
Figure 2.1. Remote monitoring system topology.(Liu,2011)..... 6
Figure 2.2. Flowchart of the communication software (Wu, 2011)..... 6
Figure 2.3. Structure of the CAN bus LCD diagnostic system (Quanqi, 2011) 7
Figure 2.4. Schematic diagram of the remote diagnosis system in automobiles (Liyuan,2010) 8
Figure 2.5. Architectural diagram of remote monitoring and control system (Li, 2009) 9
Figure 2.6. Fault diagnoses and display system also can be seen in topology (IEEE, 2010) 10
Figure 2.7. Fault diagnosis and display system (IEEE, 2010) 10
Figure 2.8. CAN BUS monitoring and fault diagnosis system (IEEE, 2010)..... 11
Figure 2.9. Running application program of system (IEEE, 2010) 11
Figure 2.10. Engine Control Network (Bin, 2010) 13
Figure 2.11. Hardware structure for the CAN network (Shi, 2012) 14
Figure 2.12. The CAN BUS initialization process algorithms (Shi, 2012)..... 15
Figure 3.1. Block Diagram of the Multiplex System 17
Figure 3.2. Block diagram of the Multiplexer..... 18
Figure 3.3. CAN layer model according to OSI standard 20
Figure 3.4. The Diagram of Typical CAN nodes..... 21
Figure 3.5. The Logical representation of the CAN bus 21
Figure 3.6. A Sample of CANID configurations 28
Figure 3.7. Details of the J1939 CANID..... 30
Figure 3.8. Transport protocol message of the engine 47
Figure 3.9. Real-time representation of Engine DTC information 48
Figure 3.10. ABS Diagnostic message viewed by using CAN Analyzer 51
Figure 3.11. ECAS Diagnostic message viewed by using CAN Analyzer..... 53
Figure 3.12. TCU Diagnostic message viewed by using CAN Analyzer 55
Figure 3.13. Typical CAN connection of the microcontroller..... 61

Figure 3.14. Connection of the Diagnostic ECU on the vehicle 62
Figure 3.15. Schematic of the Diagnostic Message sending to the Remote Center... 62
Figure 3.16. The flowchart of the error detection in the P-CAN 64
Figure 3.17. An error sending procedure via GPRS/3G modem 66
Figure 4.1. Remote Diagnosis Test Bus 69
Figure 4.2. Installation of the test hardware on the test bus 70
Figure 4.3. Display of the Diagnostic Trouble Code in Instrument Cluster of bus . 72
Figure 4.4. Example Error Sending Mail 73
Figure 4.5. Example Error Sending to Mobil Phone 74

LIST OF TABLES **PAGE**

Table 3.1. Structure of a 29-bit identifier 23
Table 3.2. PGN bit configuration example 24
Table 3.4. J1939 Layers 31
Table 3.5. Configuration of the Diagnostic Trouble Code 34
Table 3.6. Example DTC format 36
Table 3.7. DTC Representation in CAN Data Frame for DM1 36
Table 3.8. Specifications of the Diagnostic Messages 38
Table 3.9. Message Format when a request of the DM1 message. 39
Table 3.10. SPN representation of DM1 Message Format 39
Table 3.11. Table of the Malfunction Indicator Lamp 40
Table 3.12. Table of the Red Stop Lamp 40
Table 3.13. Table of the Amber Warning Lamp 41
Table 3.14. Suspect Parameter Number 42
Table 3.15. Failure Mode Identifier 42
Table 3.16. DTC representation of Transmission according to SAE J2012 56
Table 3.17. An example of Transmission Diagnostic Message 57
Table 3.18. An example of Transmission Multiple Diagnostic Message 57
Table 3.19. Vehicle CAN fault code configuration 58
Table 3.20. LPC2000 CAN controller registers 60
Table 4.1. Vehicle diagnostic message 67
Table 4.2. Example vehicle diagnostic trouble code 68
Table 4.3. Engine Coolant Level Sensor Fault Description 71
Table 4.4. Engine Coolant Level Sensor Diagnostic Trouble Code Definirtion 72

1. INTRODUCTION

Development of the electronics in automotive industry is increasing very fast. Engine control unit, transmission control unit, brake control unit and vehicle body controller have already been used in the automotive industry. Nowadays, air conditioner control unit, motor fan control unit, door control unit have been also added to electronic control units. These increases about the automotive electronics makes it difficult maintain, fault analysis and repair of the vehicle. As an example case; if a scenario has been written that a bus driver driving on a highway with over fifty passengers onboard. While driving the vehicle's red stop lamp is activated and starts to blink. Although instrument cluster gives limited information, without a diagnostic tool which is different for every model, understanding the situation and the cause of the error is nearly impossible. At this point a repair call service must be notified, the vehicle must be towed or the service must come to the bus to solve the problem. In any case not all the warnings the vehicle must be stopped or not used. In the case of an active error with a remote diagnostic tool the repair service department can connect to a vehicle and in most cases the problem can be solved without causing customer discomfort and dissatisfaction. The aim of this study is to determine the fault codes such as described in the scenario above. If a personnel of technical service knows active error, the personnel will help to the driver. In this study, following items are considered as important points for remote diagnosis of the vehicles:

- Real time vehicle tracking and determine fault diagnosis, which are possible to happen in the future.
- Warning the driver to save the dangerous faults that are not determined.
- To determine the faults from remote center and to decrease the repair costs.
- Minimizing the customer victimization and to increase the satisfaction.
- Storing a data from the running vehicles to use in future development.
- To make a unique diagnostic tool for electronics.
- To save the data of repairing.

The results of this work are applicable to the other vehicles in automotive industry.

Material and Method: In automotive industry, a data network called "CAN (Controller Area Network)" is used to communicate the electronics. Each electronic control unit, used in the vehicle, is adaptable to the CAN network. ECU receives the signals via CAN, and it transmits the signals via J1939 based CAN data link. For example Engine ECU (Electronic Control Unit) transmits the accelerator pedal position via CAN according to the SAE (Society of Automotive Engineers) J1939 automotive standard. J1939 is an automotive data bus standard package. It contains seven layers like, an application layer and a diagnostic layer. Diagnostic layer defines the failure code information of electronic control units on the CAN bus. Each ECU transmits the diagnostic messages to the CAN network. A diagnostic message defines failure code, parameter number of the part, and type of the fault. In current application, different diagnostic tools are needed to determine the failure codes from different ECUs. A CAN bus node is needed to determine fault codes, which are transmitted by ECU's. In this work, the diagnostic messages of the each different ECU have been investigated. A failure code list has been prepared for engines, transmissions, brake systems, and door system. Also a CAN matrix for the diagnostic messages has been investigated. A CAN bus network communication hardware has been designed to read diagnostic messages by using the literature. This hardware consists of ARM (Advanced RISC Machines) processor, CAN transceiver, GPS (Global Positioning system) module and GPRS (General Packet Radio Service) modem system. A CAN network software; GPS and GPRS network software has been developed in this work. This study is called as the REDI (Remote Diagnostics) project, consists of the following items;

- Real time position, height, velocity information is read via GPS.
- Fault code of the ECU is read from the CAN network.
- Fault code is sent to the remote center via GPRS.
- Some parameters (brake level, temperature, speed, etc.) is sent via GPRS.
- Some functions of the vehicle will be controlled via GPRS remote control.

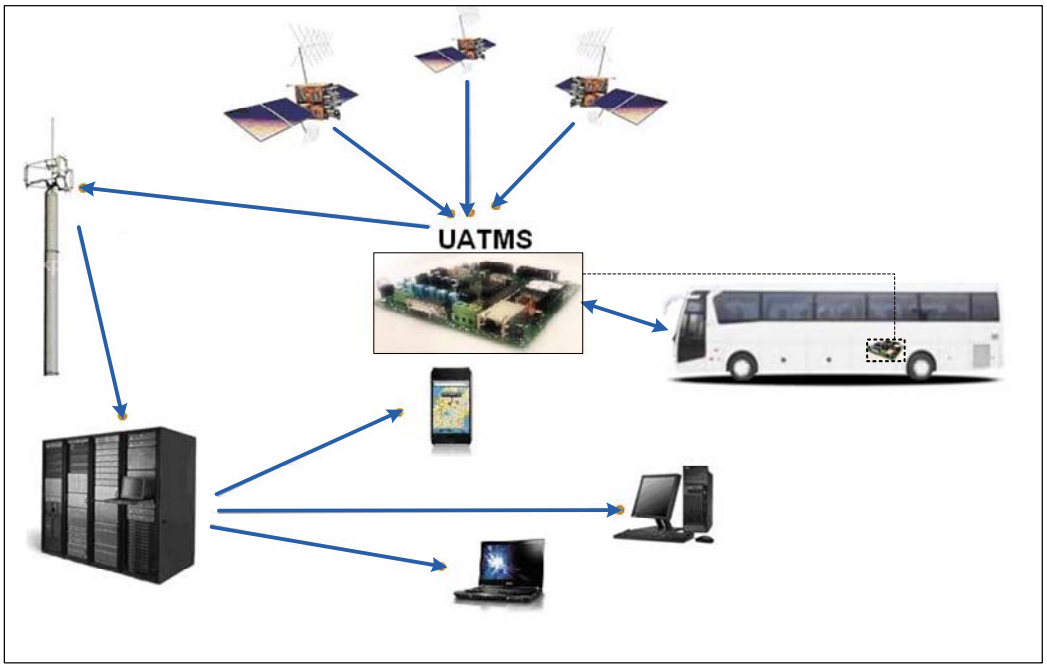


Figure 1.1. Topology of the REDI System

General system structure of the REDI project is shown in Figure 1.1. The developed electronic hardware gets the data from electronic control units like the engine, transmission, and also it is connected to body controller unit. This REDI system has been connected to the vehicle CAN network called PCAN (Powertrain CAN) and KCAN (Body CAN). Development steps of the REDI system are;

- Hardware to receive and to transmit data from the CAN network.
- GPS and GPRS communication hardware design.
- CAN communication software development
- Software design to determine the CAN diagnostic messages.
- Software development for the remote communication

2. LITERATURE REVIEW

The project REDI is a combination of a few specialist subjects. These subjects are; ARM processor, CAN transceiver, vehicle data bus, CAN network, remote communication, GPRS and GPS. In the below, the related previous studies about these subjects has been explained.

2.1. Previous Studies about Remote Communication

Liu, Wu, and Chu, are from the Jilin University (2011), are also worked for this electrical vehicle industry. They have thought that the feedback data and quality reports are very important for growing of the electrical vehicles. They have started with very important part of the electrical vehicles. Battery is still in the growing period for the electrical vehicles. These guys investigated a wireless monitoring system to get the battery fault diagnosis. These systems get the data from the vehicle and sends to the remote center by using GPRS or wireless Internet. Firstly they have decided to the type of communication. The selection table can be seen below in Table 2.1. By looking the below table, GPRS communication is used, because of the low cost equipment and tariff.

Table 2.1. Comparison tables of the network types (Liu,2011)

	CDMA	GPRS	3G
Network rate	expensive	cheap	expensive
Device quantity	few	many	few
Equipment costs	expensive	cheap	expensive
Transmission rate	fast	slow	fast
Network coverage	high	high	low

The remote communication system topology is shown in Figure 2.1.

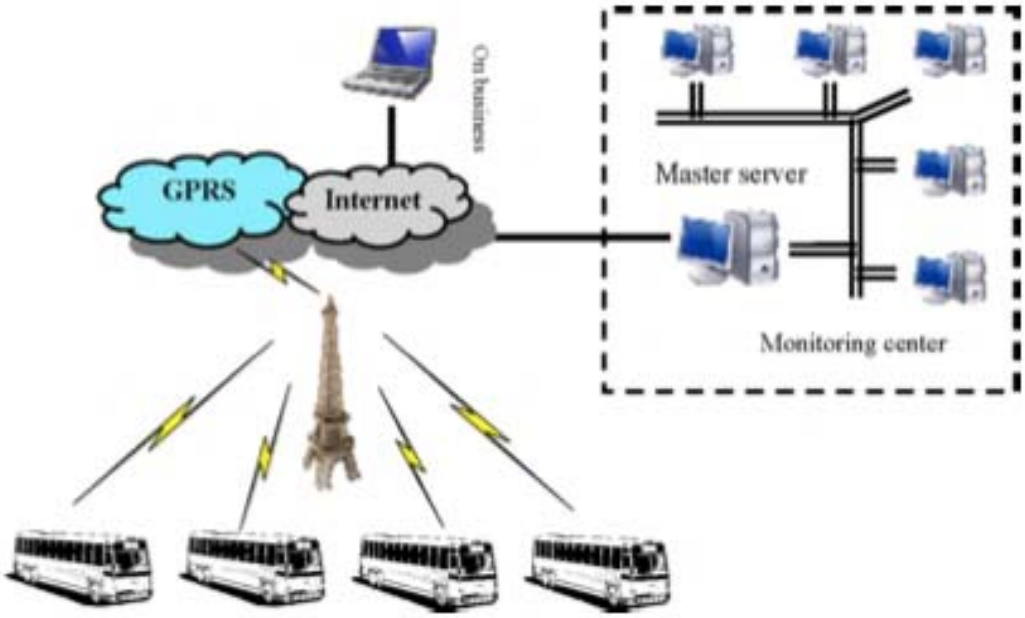


Figure 2.1. Remote monitoring system topology.(Liu,2011)

And the software flowchart is shown in Figure 2.2.

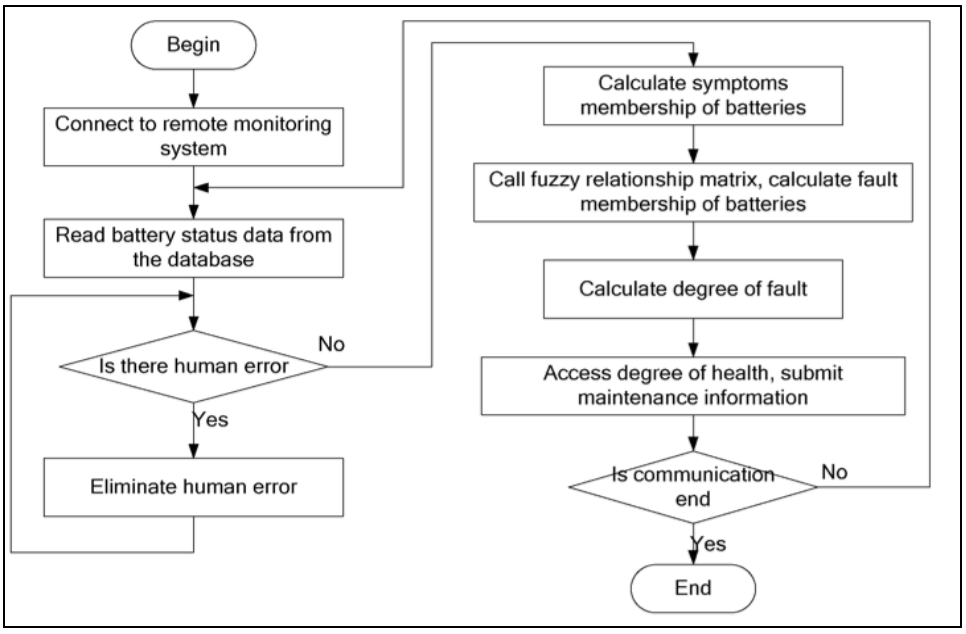


Figure 2.2. Flowchart of the communication software (Wu, 2011)

In the world automotive industry, electrical vehicle is one of the growing subjects. There are still open topics in the electrical vehicles. It is very hard to guarantee the products when the customer bought the vehicle. There will be

feedbacks from the running vehicles and the quality of the products will need to be increased.

2.2. Previous Studies about Vehicle Data Bus

The electronic control system of a vehicle is getting more complex every day. Also the bus network becomes more complex. In the automotive electronic control units, sensors are connected using CAN. CAN protocol also get error messages of the control units and data of the sensors. In general to get these error messages, most skilled technicians or advanced devices must be found in the manufacturers. Most of the error messages may become active when the buses are on road. It is very important to know the error messages status to repair the vehicle.

Quanqi, Jian and Yanyan from Beijing University (2011) proposed a design and method to monitoring system and faults. They have read error messages from the CAN-BUS according to the SAE J1939 standards. They got about 400 faults. A LCD (Liquid Crystal Display) screen added to the system to monitor the faults. The structure of the system is shown in Figure 2.3.

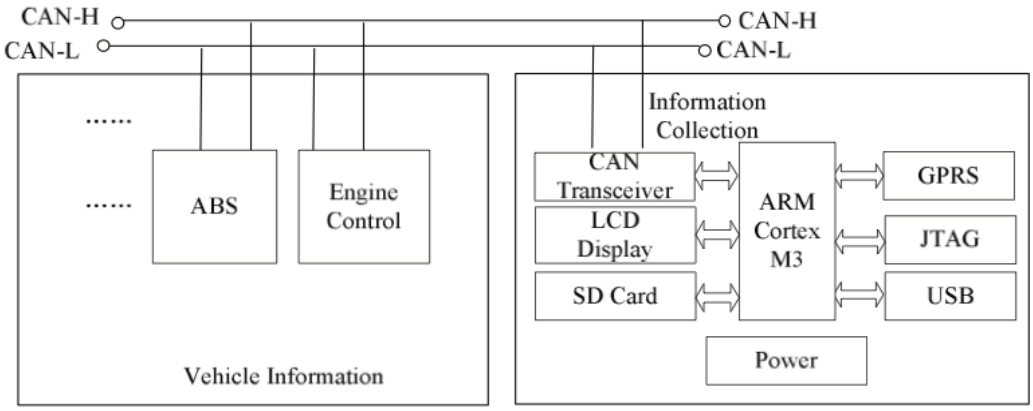


Figure 2.3. Structure of the CAN bus LCD diagnostic system (Quanqi, 2011)

The designed system based on ARM Cortex M3 microcontroller. This controller consists of CAN controller, A/D (Analog to Digital) converters, USB and other modules. This controller reduces the external circuits, reduces circuit costs, by including external circuits.

It is very important for the REDI project to connect the vehicle data bus. This paper of Beijing University consists of 400 fault codes. REDI project must have approximately 1500 fault codes per vehicle. Since of the bus has more complex electronics than cars have. This paper is considered a base study to connect the vehicle CAN-BUS.

Liyuan, Guihe from Jilin University (2010) are also studied remote fault monitoring system for the vehicles. They have got the fault messages and video camera records from the vehicle and the data are sent to remote service using telematics. They chose the 3G networks for the remote service connection. The schematic diagram of the system is shown below Figure 2.4.

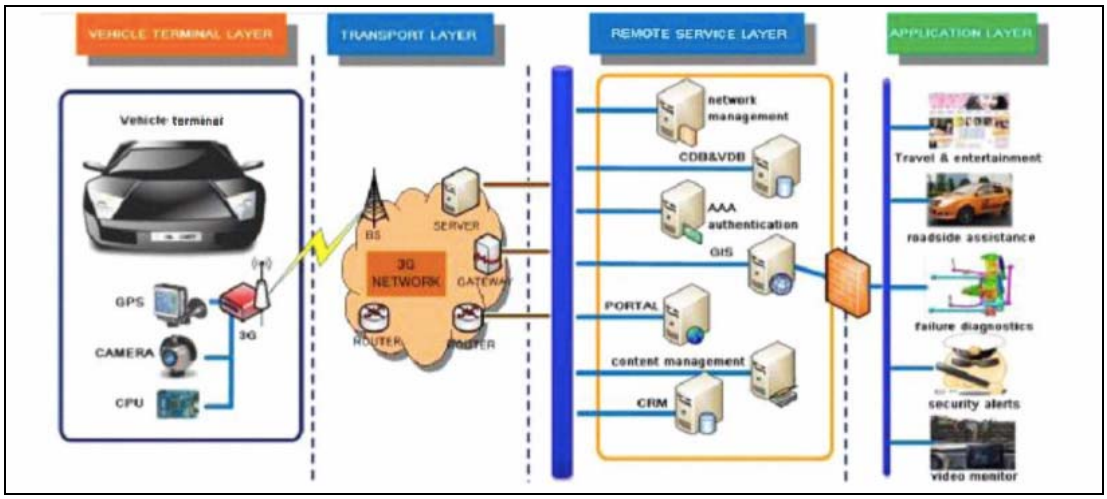


Figure 2.4. Schematic diagram of the remote diagnosis system in automobiles (Liyuan, 2010)

Li and Xu from the Guilin University (2009) are studied for the remote monitoring using embedded systems. They have also considered for the CAN-Bus vehicle information. For example they got the tire pressure, engine temperature from the CAN-BUS and recorded in an embedded database for these vehicle data. Architecture diagram of the monitoring and control system is shown below in Figure 2.5

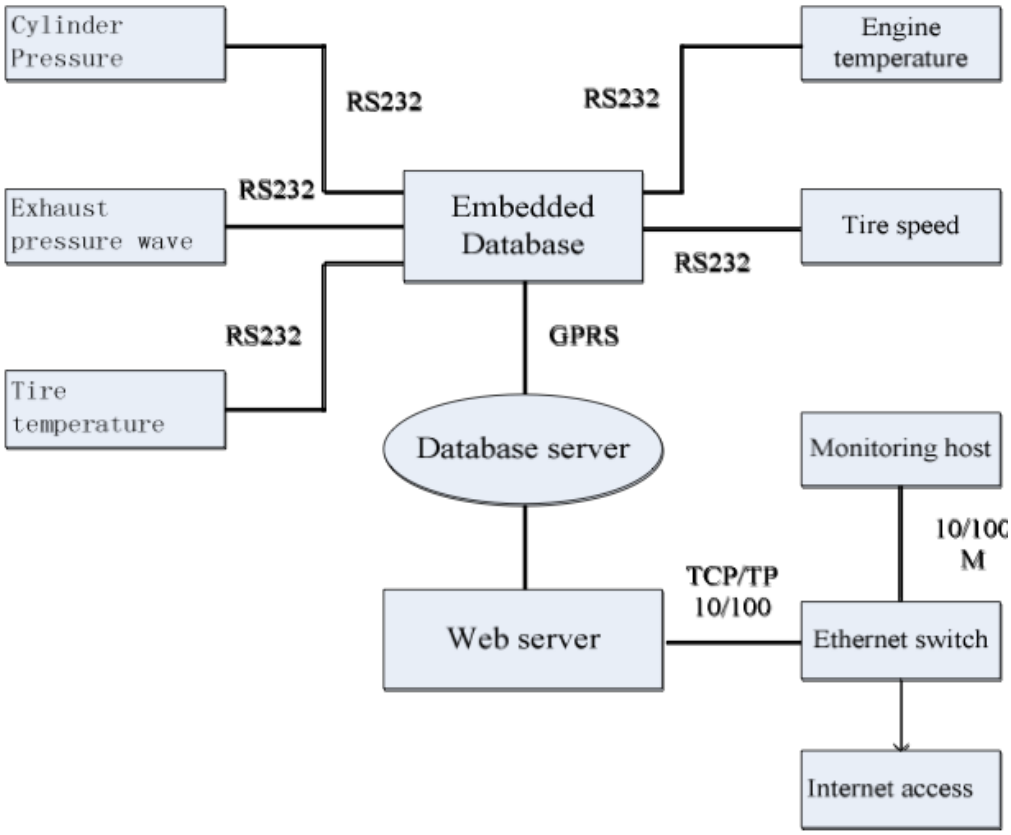


Figure 2.5. Architectural diagram of remote monitoring and control system (Li, 2009)

State Key Laboratory from Jilin University presented a fault diagnosis system for the heavy vehicles. They have implemented a 32-bit embedded processor and Windows CE operating system to monitoring the faults and data. They have investigated SAE J1939 CAN BUS standards to analyze how ECUs send the fault codes. Firstly they have studied how works can. They have designed a fault diagnosis and display system. The general topology of the vehicle data network is given below in Figure 2.6.

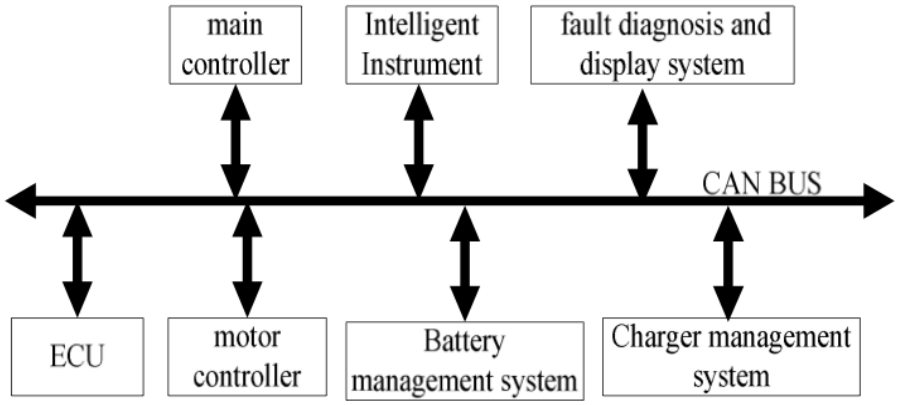


Figure 2.6. Fault diagnoses and display system also can be seen in topology (IEEE, 2010)

A principle diagram of the fault diagnosis and display system is shown below in Figure 2.7. As it seen on Figure 2.7, system gets the CAN BUS data from the vehicle. It writes to SD (Security disk) card or shows the data on the touch screen.

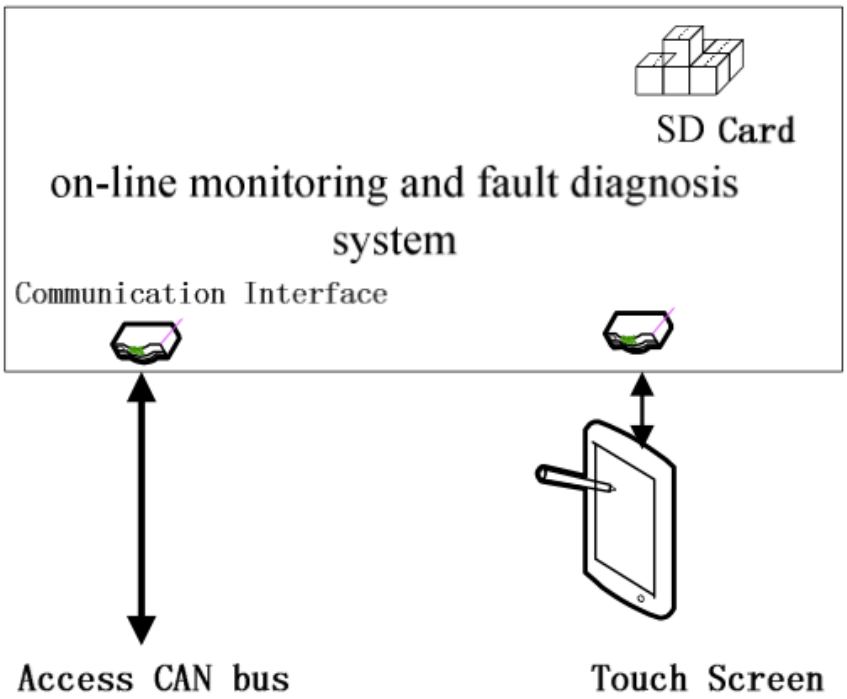


Figure 2.7. Fault diagnosis and display system (IEEE, 2010)

This system has two main studies to achieve the project. Hardware design and software development. The hardware was based on ARM microcontroller. It also had LCD and SD card interfaces. Software design is to work an application program. Below the application photos of the system can be seen in Figure 2.8



Figure 2.8. CAN BUS monitoring and fault diagnosis system (IEEE, 2010)

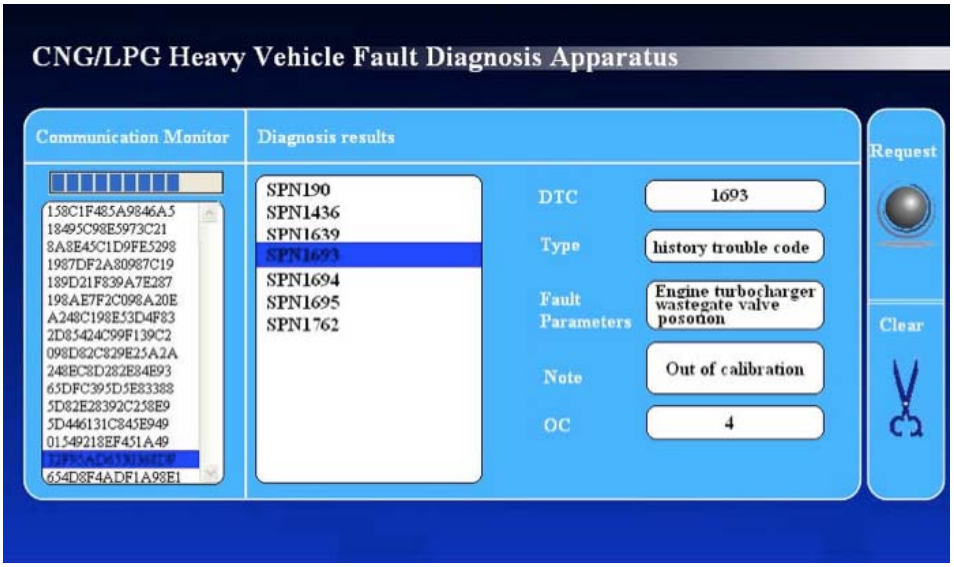


Figure 2.9. Running application program of system (IEEE, 2010)

Sankavaram, Pattipati, Kodali, Azam, Kumar and Pecht (2009) studied about prognostic system. Prognostic means to estimate the remaining life of the vehicle systems and possible faults. They have implemented a system that has prognostic features for the hybrid vehicle battery. Previous studies about the sensor technology, remote communication and network capabilities are very important to monitoring the vehicle's health status. Prognostic is a new trend to fulfill the needs of customers demanding high vehicle availability. This paper presented us the prognostic and condition monitoring steps.

Kai and Fuwu from Wuhan University of Technology (2010) presented a remote vehicle diagnosis system that also performs in the vehicle development and management. This paper presents the problems of the previous diagnosis technologies. This system has two main parts; one of these communication interfaces and the other one is PC diagnostic software that connects the vehicle and remote monitoring center.

Chen and Wei (2010) developed a speech controlled vehicle diagnostic system. Speech controlled interface proposed in this paper to control the vehicle diagnostic. There are predefined commands to monitoring the vehicle speed, engine rpm and etc. They have joined this speech control interface with the 3G wireless network communications to send the vehicle diagnostic data to remote server. The speech control is designed for safely operate the proposed system without removing their eyes from the road.

Chang-hao, Lu and Ju (2011) have designed a monitoring system to monitor the CAN bus. System has screens named as trace, graphic display, write signal etc. These screens used by producers. System can be used for simulate the CAN network. User can send message to the network using write screen and also can see the status of the network using the trace or graphic display window. A CAN card used for writing or reading the messages. In this paper, they have used LabVIEW software to compile a CAN bus monitor system. A NI High-Speed CAN card is used to connect the CAN bus monitoring system to the CAN bus. Trace window shows the data of the messages in the CAN network. Graph window shows the signal in a graph. Write function is used for transmitting the signals.

Bin, Jiansen, Konghui, Haito and Jianwei (2010) have studied about the J1939 in city bus vehicles. They have developed a network that consists of an Eelectronic Control Unit, a Hardware Monitor Interface, an on board computer and electronic control units (Engine, EBS (Electronic Brake System), Transmission, etc). An engine control network is shown in Figure 2.10.

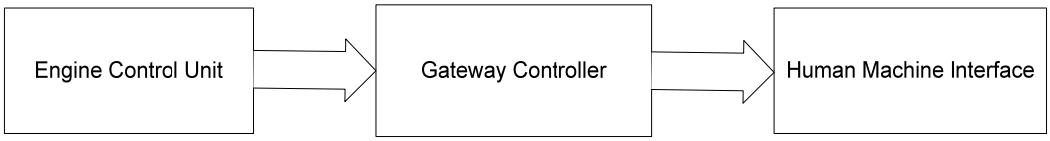


Figure 2.10. Engine Control Network (Bin, 2010)

They have designed two J1939 networks called as Power Net and Body Net. Power Net is a network that is based on SAE J1939 standard. Body Net is the application layer of the vehicle by expanded on SAE J1939. Main aim of their project was to be able to get the diagnostic messages of the Body Net and Power Net. The diagnostic layer of the SAE J1939 has been investigated and diagnostic messages are determined for the engine, transmission, doors, etc.

Prodanov, Valle and Buzas (2009) studied to develop tools and methodologies for design and simulation of CAN. In this paper, they have proposed and have verified a CAN bus transceiver. They tested the accuracy, speed and convergence of the network. Different architectures of the CAN bus transceivers implemented to test the network. The tests were implemented using the VHDL language. Power consumption analysis, corner analysis, fault analysis, and signal integrity, etc. can be implemented.

Çivril (2006) informed that a vehicle tracking system consists of mainly three components: A GPS receiver used to determine the position of the vehicle, a communication unit used to send the positioning data to the tracking center, and a software package used to track the vehicle on maps and to carry out the positional analysis. In this study, GPRS technology is chosen to send the positioning data determined by the GPS receiver. The communication cost of the system is low with this technology. In this study, software was developed in MS Visual C#. NET to display the digital maps.

Scott has published a paper for the importance of the diagnostic, to increase the repair quality and to decrease the costs for the vehicles. This method also helped to improve vehicle design, to improve customer satisfaction. Electronics are very useful systems on the vehicle, but if one of them fails, technical person requires the special diagnostic tools. A mobile diagnostic tool that is configurable to meet the vehicle's requirements is required. The ability to track faults, there may be needed to connect directly into the harness.

2.3. Previous Studies about ARM and CAN network relationship

Shi (2012) implemented a CAN bus network using ARM controller. The paper represents a design for CAN bus network, which is based on NXP LPC2294 ARM microcontroller. The design also uses TJA1050 CAN transceiver. It shows the hardware structure and software algorithm for the CAN bus modules. Hardware structure for the CAN network is shown in Figure 2.11. This paper also shows the CAN controller software algorithm. The initialization process is shown in Figure 2.12

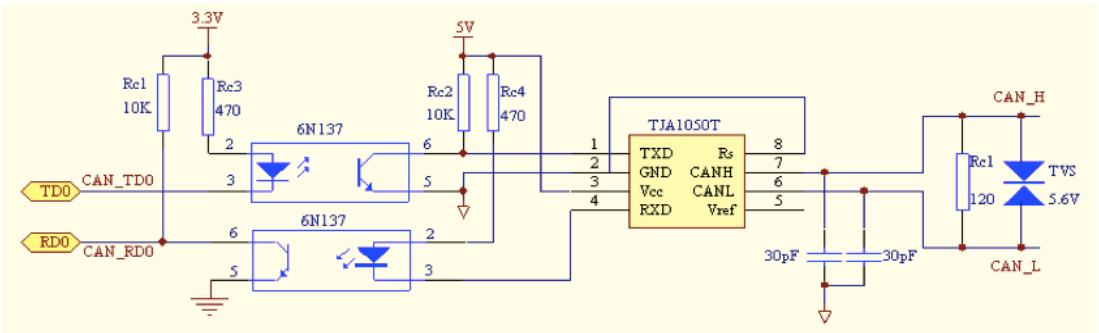


Figure 2.11. Hardware structure for the CAN network (Shi, 2012)

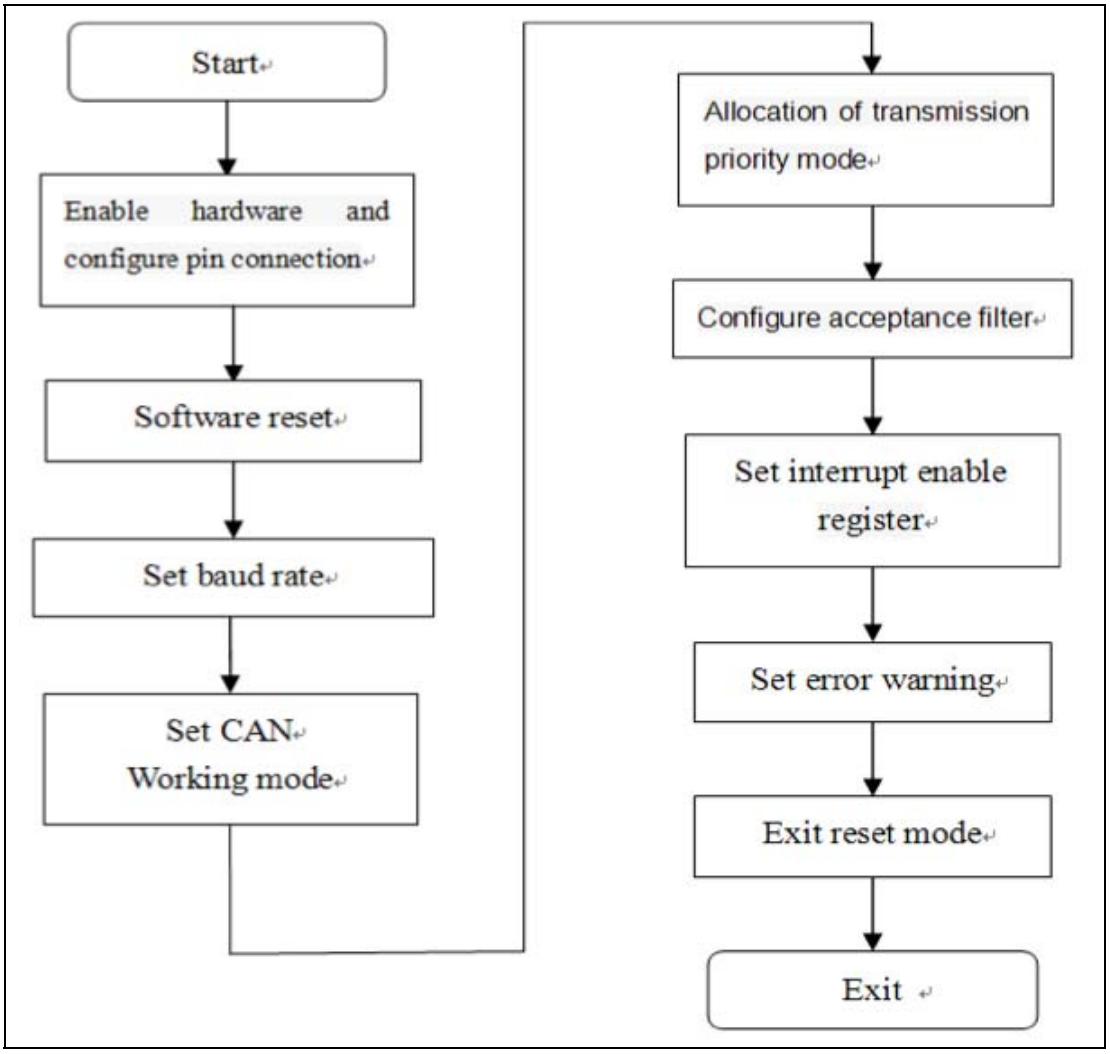


Figure 2.12. The CAN BUS initialization process algorithms (Shi, 2012)

3. MATERIAL AND METHODS

3.1. Vehicle Electronic Control System

Vehicle electronic control system is used to control the all inputs/outputs of the vehicle functions and to warn the user by checking the system faults. The vehicle has a central computer. This computer has multiplexer nodes that have multiple inputs and outputs. It has also CAN bus connectors to connect the powertrain systems and instrumentation systems. It has called as ZR-32 in the buses.

Multiplex control system is a PLC (Programmable Logic Controller) application that consists of one body controller (ZR-32), one or more input output module (CAN Multiplexer). In Figure-1, block diagram of the multiplex system is shown.

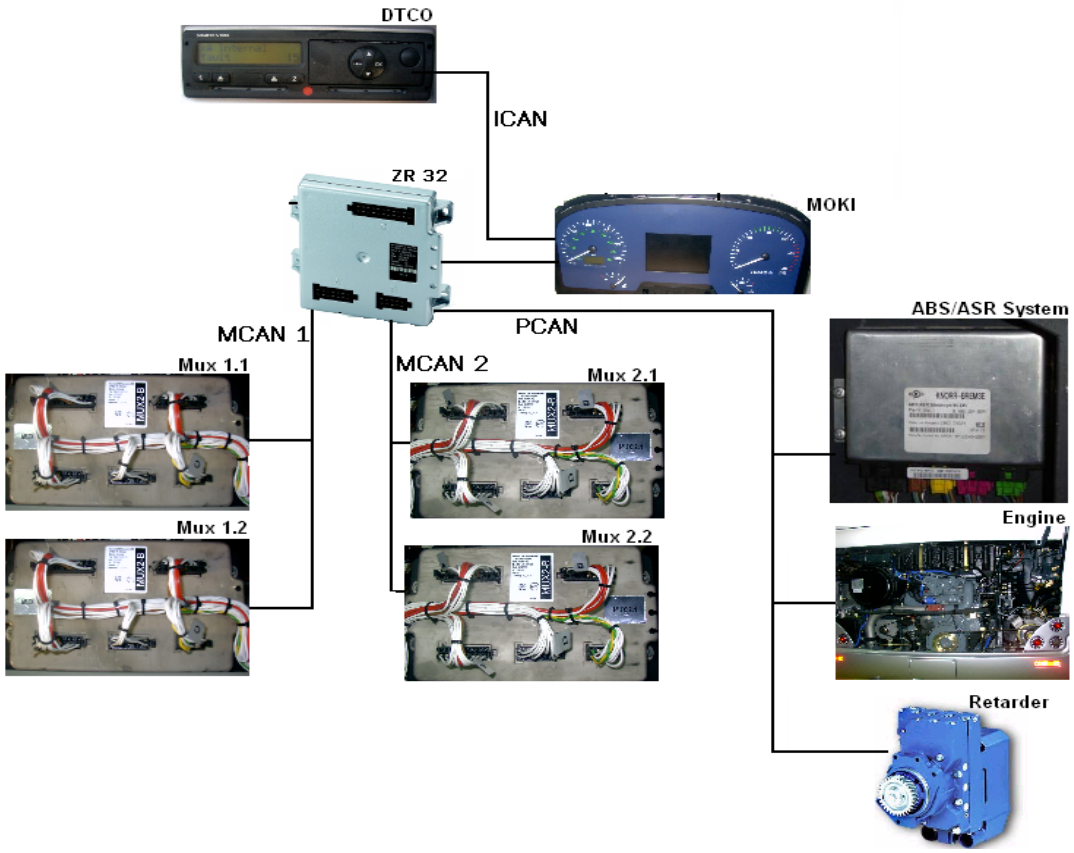


Figure 3.1. Block Diagram of the Multiplex System

PCAN (Powertrain CAN) is a CAN based communication network on the buses. This CAN line makes connection between engine control unit, transmission control unit, and brake systems. It is based on SAE J1939 communication protocol with 250kbyte/s communication speed.

ICAN (Instrumentation CAN) is also a CAN communication between body controller, instrument panel and tachograph. Its communication speed is 250kbyte/s.

MCAN (Multiplexer CAN) is CAN based communication line; its speed is 125kbyte/s, among body controller and I/O multiplexers. Body controller controls a multiplexer. Its basic function is to sense the digital inputs ans outputs on the vehicle. The multiplexer has 24 digital inputs, 6 analog inputs, and 32 digital outputs and a CAN connection input. CAN connection is based on ISO 11898 standard. In figure-2 a block diagram of the multiplexer is shown.

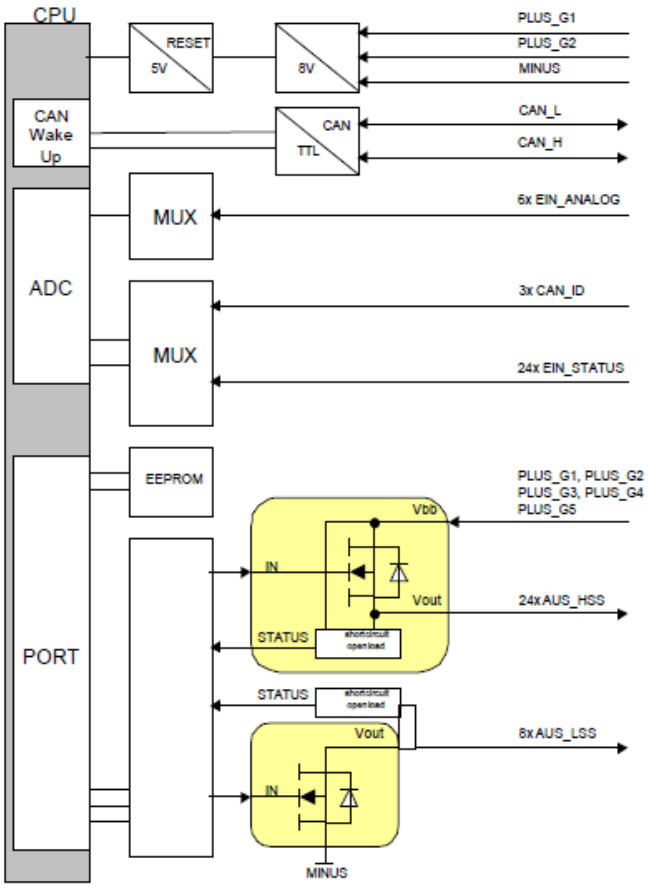


Figure 3.2. Block diagram of the Multiplexer

The basic system of the vehicle electronic system is the body controller. This controller executes an application software called as KIBES-32. This software is a Windows based PLC software. It is developed on IEC 61113-3 visual programming language.

3.2. CAN Communication Protocol

The Controller Area Network (CAN) Protocol was developed by Robert Bosch in1982. Over the last 22 years CAN communication has become a standard for Automotive Networking and has a wide uptake in non-automotive systems where it is required to network together a few embedded nodes. (Hitex, 2007)

CAN communication has many attractive features for the embedded developer. It is a low-cost, easy-to-implement, peer to peer network with powerful error checking and a high transmission rate up to 1 Mbit/sec. Each CAN packet is quite short and may hold a maximum of eight bytes of data. This makes CAN suitable for small-embedded networks, which have to reliably transfer small amounts of critical data between nodes. (Hitex, 2007)

3.2.1. CAN Layer model

OSI model is a reference to how messages are transferred in a network. CAN protocol is also follows this OSI standard. (Hitex, 2007)

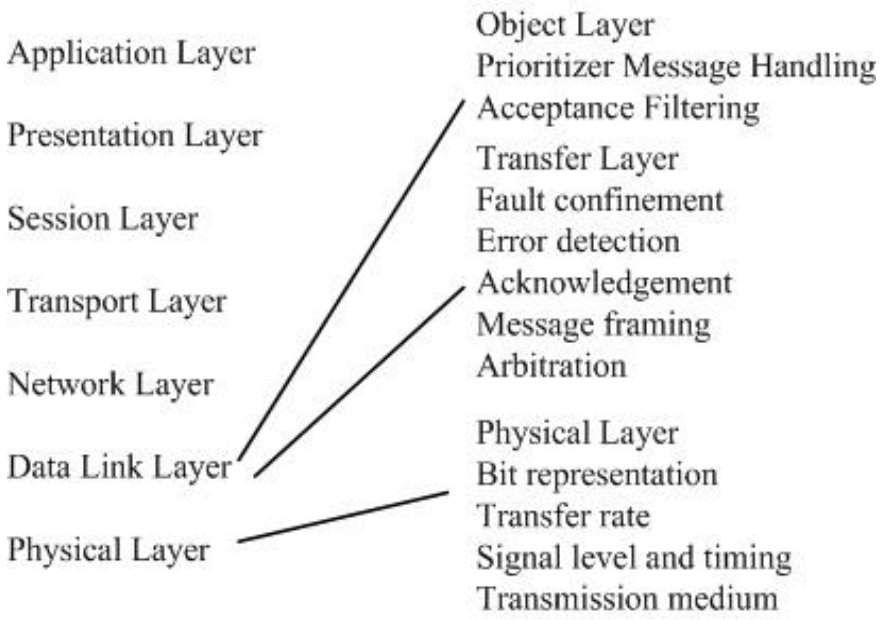


Figure 3.3. CAN layer model according to OSI standard

3.2.2. CAN Node

Below, a general CAN node is shown below in Figure 3.4. Each CAN node consists of a controller and a separate CAN controller. The CAN controller may, as in the case of the microcontroller or it may be a stand alone controller in a separate chip to the microcontroller. The CAN controller is interfaced to the twisted pair by a line driver and the twisted pair is terminated at either end by a 120 Ohm resistor. The most common mistake with a first CAN network is to forget the terminating resistors and then nothing works. (Hitex, 2007)

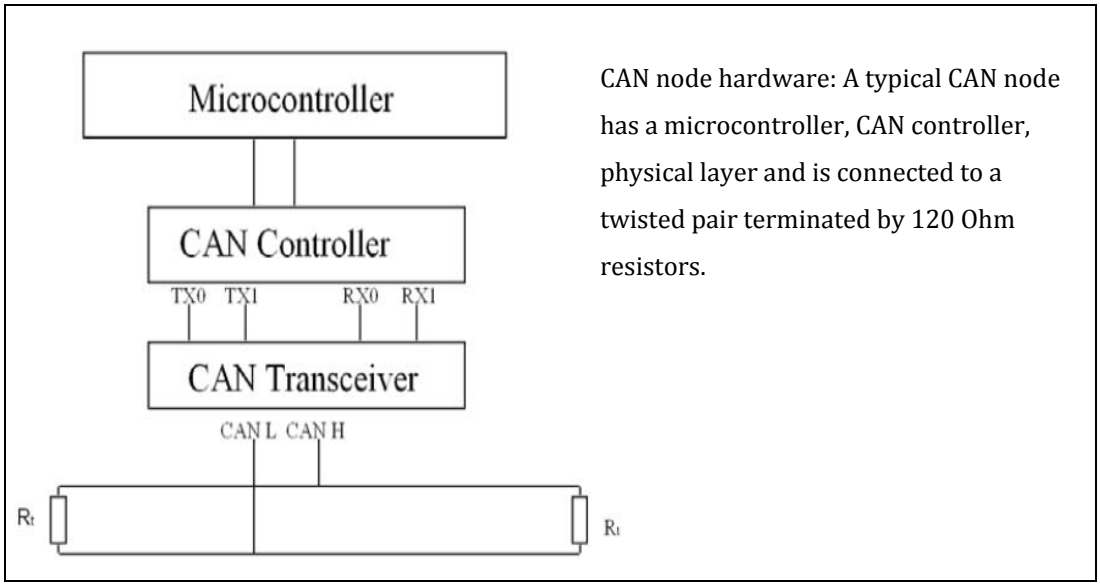


Figure 3.4. The Diagram of Typical CAN nodes

The CAN controller has a physical layer device called as receiver. So, as the node wiring on to the bus it is also listening back at the same time. This is the basis of the message arbitration and for some of the error detection.

The two logic levels are written onto the twisted pair as follows, a logic one is represented by bus idle with both wires held half way between 0 and $V_{CC.}$ A logic Zero is represented by both wires being differentially driven.

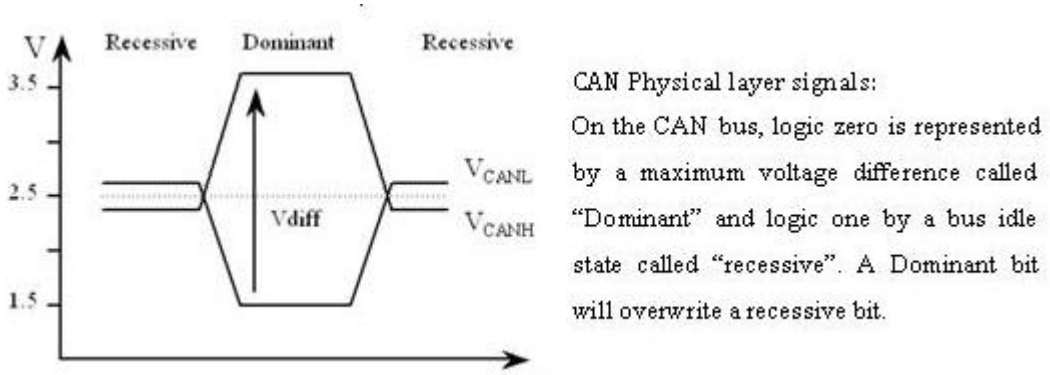


Figure 3.5. The Logical representation of the CAN bus

In "CAN speak" logic one is called a recessive bit and a logic zero is called a dominant bit. In all cases a dominant bit will overwrite a recessive bit. So, if ten nodes write recessive and one writes dominant, then each node will read back a

dominant bit. The CAN bus can achieve bit rates up to a maximum of 1Mbit/sec. typically this can be achieved over about 40 metres of cable. By dropping the bit rate, longer cable runs may be achieved. In practice 1500 metres cable can be used with the standard drivers at 10kbit/sec. (Hitex, 2007)

3.3. Sae J1939 Can Bus Standard For Automotive

3.3.1. Introduction

J1939 is a set of standards defined by SAE. They are used in heavy-duty vehicles (trucks and buses, mobile hydraulics, etc.). In many ways, J1939 is similar to the older J1708 and J1587 standards, but J1939 is built on CAN. The physical layer (J1939/11) describes the electrical interface to the bus. The data link layer (J1939/21) describes the rules for constructing a message, accessing the bus, and detecting transmission errors. The application layer (J1939/71 and J1939/73) defines the specific data contained within each message sent across the network. (SAE, 2000)

- Higher-layer protocol built on CAN
- Used in heavy-duty vehicles
- The speed is nearly always 250 Kbit/s

3.3.2. Message Format and Usage (J1939/21)

Most messages defined by the J1939 standard, are intended to be broadcast. This means that the data is transmitted on the network without a specific destination. This permits any device to use the data without requiring additional request messages. This also allows future software revisions to easily accommodate new devices (address assignments). When a message must be directed to a particular device, a specific destination address can be included within the message identifier.

For example, a request for a specific torque value from the engine instead of a specific torque value from the brake controller.

J1939 uses the 29-bit identifier defined within the CAN 2.0B protocol shown in Figure 1. The identifier is used slightly different in a message with a destination address (”PDU 1”) compared to a message intended for broadcast (”PDU 2”). PDU stands for Protocol Data Unit

The SOF (Start of Frame), SRR (System Readiness Review), and IDE (Identifier Extension) bits are defined by the CAN standard and will be ignored here. The RTR (Remote Transmission Request) bit (remote request bit) is always set to zero in J1939. The 29-bit identifier used in J1939 is structured in the following way.

Table 3.1. Structure of a 29-bit identifier

Priority	Reserved	Data page	PDU format	PDU specific	Source Address
3 bits	1 bit	1 bit	8 bits	8 bits	8 bits

The first three bits of the identifier are used for controlling a message’s priority during the arbitration process. A value of 0 has the highest priority. Higher priority values are typically given to high-speed control messages, for example, the torque control message from the transmission to the engine. Messages containing data that is not time critical, like the vehicle road speed, are given lower priority values.

The next bit of the identifier is reserved for future use and should be set to 0 for transmitted messages.

The next bit in the identifier is the data page selector. This bit expands the number of possible Parameter Groups that can be represented by the identifier.

The PDU format (PF) determines whether the message can be transmitted with a destination address or if the message is always transmitted as a broadcast message. The interpretation of the PDU specific (PS) field changes based on the PF value:

- If the PF is between 0 and 239, the message is addressable (PDU1) and the PS field contains the destination address.
- If the PF is between 240 and 255, the message can only be broadcast (PDU2) and the PS field contains a Group Extension.

The Group extension expands the number of possible broadcast Parameter Groups that can be represented by the identifier.

The term Parameter Group Number (PGN) is used to refer to the value of the Reserve bit, DP, PF, and PS fields combined into a single 18 bit value.
Example: The ID 0xCF004EE can be divided into the following fields in table 2.

Table 3.2. PGN bit configuration example

0x0C				**0xF0**	**0x04**	**0xEE**
000	011	0	0	11110000	00000100	11101110
---	Prio	R	DP	PF	PS	SA

- PGN = the R, DP, PF and PS fields - in this case 0x0F004.
- PF = 0xF0 = 240, i.e. this is a PDU2 (broadcast) message
- PS = 0x04, i.e. the Group Extension = 4

The last 8 bits of the identifier contain the address of the device transmitting the message. The address is the label or "handle" which is assigned to provide a way to uniquely access a given device on the network. For a given network, every address must be unique (254 available). This means that two different devices (ECUs) cannot use the same address. (Anonymous, 2010)

3.3.3. Transmitting Messages (J1939/21 and J1939/7x)

To send a particular data item, a message must be constructed with overhead that describes the data to be sent. Related data items are typically packed together

within a message to reduce overhead. J1939/71 defines some standard PGNs, which describe the parameters to be sent in a message. The J1939/71 document also includes information about message priority and transmission rate. Note that when a device does not have data available for a given parameter, the byte that should have contained this parameter is set to "not available" (0xFF) so that a receiver knows that the data is missing. Messages that need more than eight bytes of data can be sent as multi-packet messages. Multi-packet messages are transmitted by means of the Transport Protocol Functions defined in J1939/21. However there are two ways of transmitting multi-packet messages:

1. Broadcast Announce Message (TP_BAM)
2. Connection Management (TP_CM)

3.3.3.1. TP_BAM messages

TP_BAM messages use a global destination address, which means that all devices on the network will receive these messages. The transmission is started with a Connection Management (CM) message, PGN = 0x00EC00, with a Control byte indicating TP_BAM. The message data follows in Data Transfer (DT) messages, PGN = 0x00EB00.

3.3.3.2. TP_CM messages

TP_CM messages are sent point to point between two devices. The transmission starts with a CM message with a Control byte indicating Request To Send (RTS). The receiving device responds with a CM message with the Control byte indicating Clear To Send (CTS). The transmitting device then sends the portion of the data indicated in the CTS using DT messages. This handshake of CTS then DT messages continues until the entire message is transmitted. The receiver transmitting a CM message with a Control byte indicating End Of Message Acknowledgement (EOM) terminates the connection at the completion of the message. Note that for this

process to work, the CM message contains additional data based on what the control byte is. The RTS includes: number of bytes, number of packets and the PGN whose data will be transported. The CTS includes the number of packets the receiver expects next and the packet number to start with. (Anonymous, 2010)

3.3.4. Receiving Messages (J1939/21 and J1939/7x)

There are various techniques (and chips) available for capturing selected messages off the network. Several general observations can be made, however regarding received messages:

1. If a message is a destination specific request or command, the device must determine if the destination address matches an address claimed by the device. If there is a match, the receiving device must process the message and provide some type of acknowledgment.
2. If a message is a global request, every device, even the originator, must process the request and respond if the data is available.
3. If a message is broadcast, each device must determine if the content is of relevance or not.

3.3.5. ECU Design (J1939/1x, J1939/21, and J1939/7x)

Although every manufacturer will have different performance requirements for the electronic control unit (ECU) contained within their product, several observations should be made regarding the resources needed to support J1939. The current data rate of J1939 is 250 Kbps. A typical message containing 8 data bytes is 128 bits long (excluding bits used for bit stuffing) which in time is approximately 500 microseconds. The shortest message is 64 bits long. This means that a new message could be sent every 250 microsecond. Although not every message is relevant, nor is the bus loading likely to be above 50%, the receiving processor must be able to handle (or buffer) back-to-back messages for short bursts of time. This

will require some RAM (Random Access Memory) space as well as processor time for memory transfers. (Anonymous, 2010)

3.3.6. Wiring Topology - Physical Layer (J1939/1x)

The J1939 network is intended to be a single, linear, shielded twisted pair of wires running around the vehicle to each ECU. A short stub is permitted between the ECU and the "bus". This simplifies routing the main bus wiring by not requiring the main bus to connect directly to each ECU. The linear bus is necessary at a data rate of 250 Kbps in order to minimize electrical signal reflections. The termination resistor at each end of the bus also reduces reflections.

The J1939 network may actually be composed of multiple segments, with an in-line device known as a bridge present between them. These segments do not need to be directly compatible with each other. For instance, the segments may run at different data rates or use a different physical medium. The main function of the bridge is to provide electrical isolation between segments. In the event of a break on the wire between the tractor and trailer, the main J1939 segment on the tractor will continue to function. The bridge can also selectively filter which messages need to be stored and forwarded from one segment to another.

Example of how to interpret a J1939 message

This example is intended to provide the principles of how to interpret a J1939 message. Let’s look at a J1939 message with the following content:

CAN identifier: 0xCF00401

Data Bytes: 0xFF FF 82 DF 1A FF FF FF

What information does the CAN-ID provide?

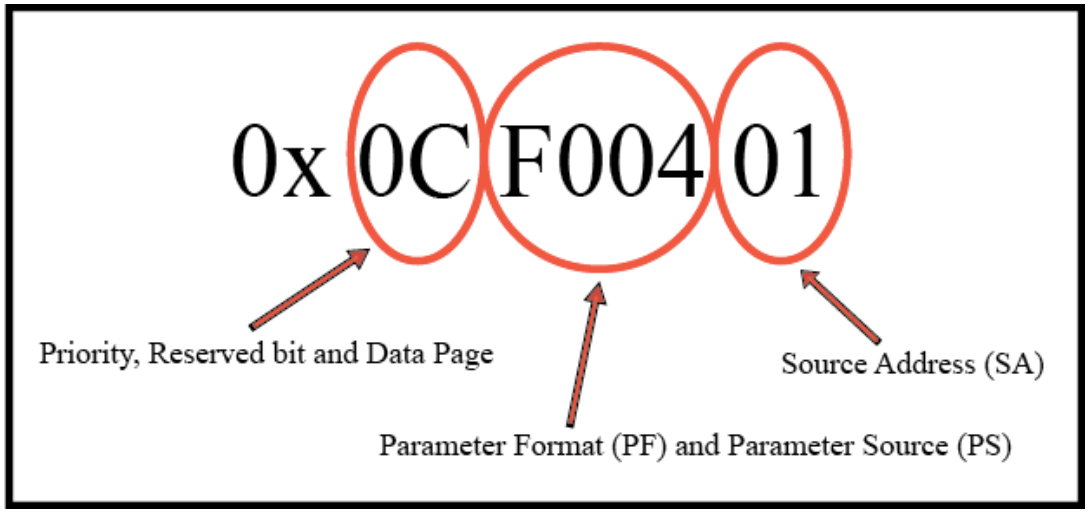


Figure 3.6. A Sample of CANID configurations

First two bytes = 0x0C = 00001100 in binary format. The first 3 bits aren't used since the identifier only consists of 29 bits. The following 3 bits represents the message priority, which in this case is 3. Thereafter follows a reserved bit and then the data page, which are used to determine the complete PGN.

The last byte of the CAN-ID is the Source Address (address of the sending device), which in this case is 1.

The PGN = 0x0F004 which corresponds to the Electric Engine Controller #1 (EEC1) according to the J1939/71 document. This document also describes the parameters and their position within the data bytes. The data field consists of the following bytes in this example:

Table 3.3. Byte positions of the EEC1 message

Data bytes	FF	FF	82	DF	1A	FF	FF	FF
Position	1	2	3	4	5	6	7	8

Data bytes 1, 2, 6, 7 and 8 in this example are not available and are therefore set to 0xFF. No raw parameter value (single byte) could have the value 0xFF.

Data byte 3 is the parameter Actual engine percent torque. The raw value 0x82 is 130 decimal. According to the J1939/71 document the scaling 1% per bit and the offset is -125. Therefore, the actual value for this parameter is 5%.

Data bytes 4 and 5 form the parameter Engine speed. The first byte (4) is the least significant (Intel byte order). The raw value 0x1ADF = 6879 decimal. The scaling is 0.125 rpm per bit and the offset is 0. The actual value for this parameter is therefore just under 859.875 rpm.

3.3.7. J1939 message format

In comparison to CANopen, which is using the 11-bit identifier (CAN-ID) J1939 is using the 29-bit CAN-ID. The CAN-ID in J1939 consists of a parameter group number (PGN) and a source address (see Figure 1). A parameter group (PG) is assembled of various parameters defined in the J1939 series, such as vehicle speed, oil temperature, etc. Thus, a PGN identifies the content of the data field.

The priority field indicates the priority of the message, where '0' is the highest priority and '7' the lowest. The data page field is supposed to be set to '1' in the future, if no more free numbers are left to be assigned to new PGNs. If the value of the field 'PDU format' is between 00_h and $F0_h$, the field 'PDU specific' has to be interpreted as destination address. This is used for peer-to-peer communication between two devices. If the value of the field 'PDU Format' is greater than F0h, 'PDU specific' is interpreted as a so-called 'group extension'. The PGNs are sent according to the producer/consumer model (broadcast). The source address has to be unique for every device in the network and may be achieved by participating the address claiming procedure. (Anonymous, 2010)

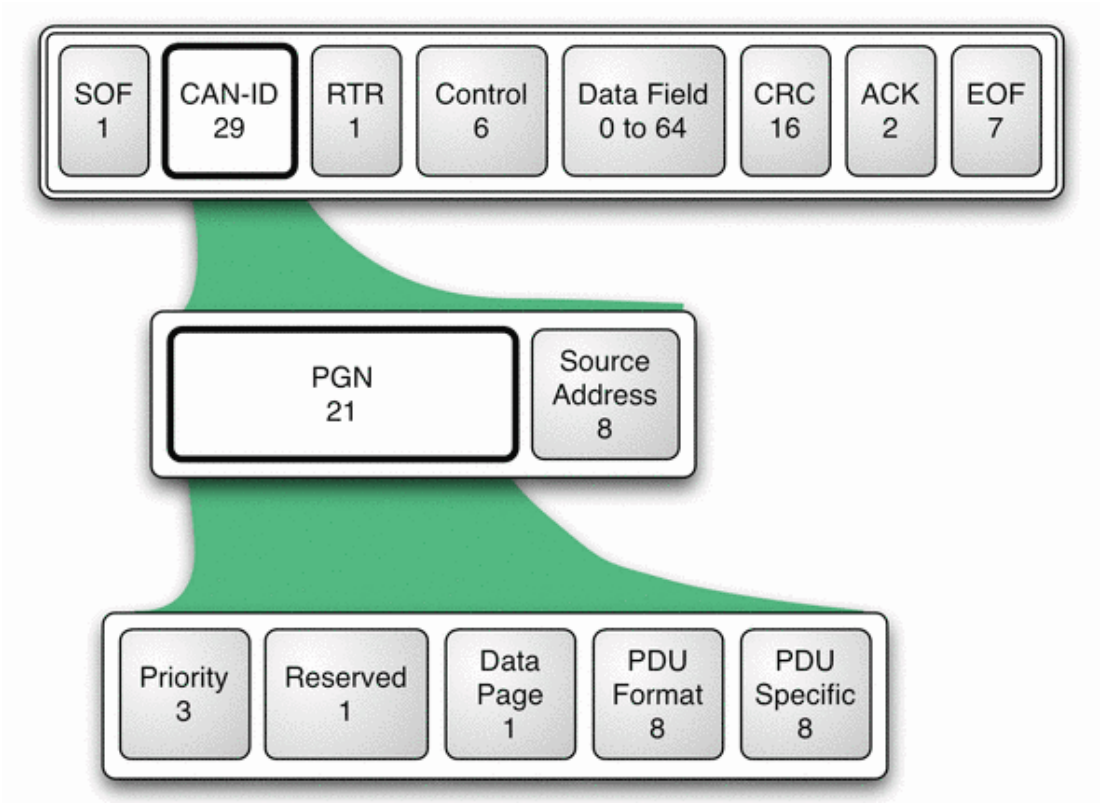


Figure 3.7. Details of the J1939 CANID

3.3.8. J1939 set of profiles

The SAE J1939 set of profiles is based on the Controller Area Network (CAN) data link layer (ISO 11898-1) using the extended frame format (29-bit identifier). Several documents have undergone revision after the initial publication in 1998. The specifications have been added to, parts taken off and clarified. The set of specifications, available from SAE, includes:

Table 3.4. J1939 Layers

• J1939/11	Physical layer (250 kbit/s, twisted shielded pair)
• J1939/13	Off-board diagnostic connector
• J1939/15	Reduced physical layer (250 kbit/s, unshielded twisted pair)
• J1939/21	Data link layer
• J1939/31	Network layer
• J1939/71	Vehicle application layer
• J1939/73	Application layer diagnostics
• J1939/74	(Draft) Application - Configurable messaging
• J1939/75	Application layer - Generator sets and industrial
• J1939/81	Network management

The naming of the layers is not always compliant to the OSI reference model and to recommended terminology. The J1939/21 and J1939/31 define partly an application layer, and the J1939/71 and J1939/73 specify an application profile. (Anonymous, 2010)

SAE J1939/11: This physical layer specification is based on the ISO 11898-2 standard (CAN high-speed physical layer). It defines a single, linear, shielded twisted-pair of wires running around the vehicle linking each of its ECUs together (3-wire connection: CAN_HI, CAN_LO, Shield). The topology is defined as a linear bus with a bus speed of 250 kbit/s and termination resistors to reduce reflections. A J1939 network can be made of multiple bus sections, each one linked with a bridge. The main function of the bridge is to provide electrical isolation between different segments so that electrical failure of one system will not cause such failure to an adjacent system. For example, the failure of the CAN/J1939 system on the trailer should not cause the failure of the truck's tractor main CAN/J1939 control system. The maximum number of ECUs is 30, and the maximum bus length is 40 m. (Anonymous, 2010)

SAE J1939/13: This specification defines a standard diagnostic interface to a diagnostic tool in the driver's cab. (Anonymous, 2010)

SAE J1939/15: As described in the SAE J1939 document, there is a minimum of seven documents required to fully define an SAE J1939 network. This document,

SAE J1939-15, describes a physical layer utilizing Unshielded Twisted Pair (UTP) cable. (Anonymous, 2010)

SAE J1939/21: This document is the heart of the specifications. It describes commonly used messages such as Request, Acknowledgement, and Transport Protocol. The Transport Protocol specifies the breaking up of large amounts of data into multiple CAN-sized frames, along with adequate communication and timing to support effective frame transmission between nodes. Slight modifications have added flexibility to the Transport Protocol, allowing the sender (server) of data to specify the number of CAN frames to be sent at any one time. Previously, this number was greatly determined by the receiver's (client's) limitations in the number of frames it could receive.

The 29-bit identifier comprises the following sub-fields: priority (3 bits), extended data page (EDP, 1 bit), data page (DP, 1 bit), PDU format (PF, 8 bits), PDU specific (PS, 8 bits), and source address (SA, 8 bits). The source address field ensures unique CAN identifiers, so no two nodes can ever transmit the same CAN identifier.

In the beginning, J1939 grouped several parameters (signals) together into a Parameter Group (PG). Each PG was then assigned a Parameter Group Number - its PGN. The PGN identifier contains the EDP, DP PF and PS fields. This structure has since caused some confusion with regard to PDU1-type (destination-specific) messages. Since the PDU-specific (group extension) field becomes the Destination Address in a PDU1 message, the question arose if the PGN changes, which is does not. The PGN is a static number referring to the data being transmitted and should be considered independently of the CAN identifier. (Anonymous, 2010)

SAE J1939/31: This specification describes repeaters, routers, gateways, and bridges. The bridge functionality, how CAN messages from one network to another are transferred, employs a message filter function to reduce the transmission of CAN messages in the individual network segments. (Anonymous, 2010)

SAE J1939/71: In the application profile all parameters as well as assembled messages called parameter groups are specified. Each CAN message is referenced by a unique number, the PGN. The latest release of the J1939/71 document incorporates

several approved additions, and brings the total number of defined messages up to almost 150. New message additions support anti-theft, fuel-specific, turbocharger, ignition, and tire pressure functions, among others. These additions and enhancements include the addition of the "source address of controlling device" parameter to several engines, transmission and brake controller messages. Inclusion of this parameter in a message will allow the receiving device to identify the original source of the message (e.g., a particular device from a bridged network). PGs are added quarterly to the document. (Anonymous, 2010)

SAE J1939/73: Additions to the diagnostics document (J1939/73) involve memory access, start/stop functions, binary data transfer, security, and calibration information. Memory access is provided with security levels. The start/stop message is used during diagnostics performance, to quiet other devices (including nodes providing bridges to other networks). Revisions of the J1939/73 document also provide clarification regarding DTC (diagnostic trouble code) encoding in the data field. This encoding, previously interpreted differently by various manufacturers, was standardized, utilizing the reserved bit as the Conversion Method (CM) bit. (Anonymous, 2010)

SAE J1939/74: This particular document, SAE J1939-74, describes the message structure for a set of messages which enable the user to determine and announce to others on the network, the parameter placement within a particular message from the special set of messages defined within this document. (Anonymous, 2010)

SAE J1939/75: This particular document, SAE J1939-75, describes the parameters and parameter groups that are predominantly associated with monitoring and control generators and driven equipment in electric power generation and industrial applications. Parameters and parameter groups that may be associated with monitoring and control of the power units, such as engines, turbines, etc, that power the generators and driven industrial equipment and which may be applied in on- and off-highway applications, shall be documented in SAE J1939-71. (Anonymous, 2010)

SAE J1939/81: The J1939/81 Draft includes state diagrams for initialization and more clearly defines constraints on the use of addresses. Its main function is to

define address claiming that will ensure unique source addresses on the network. (Anonymous, 2010)

3.4. More about J1939-73 Application Layer (Diagnostics)

J1939-73 identifies the diagnostic connector to be used for the vehicle service tool interface and defines messages to accomplish diagnostic services. California-regulated OBD II requirements are satisfied with a subset of the specified connector and the defined messages. Diagnostic messages (DMs) provide the utility needed when the vehicle is being repaired. Diagnostic messages are also used during vehicle operation by the networked electronic control modules to allow them to report diagnostic information and self-compensate as appropriate, based on information received. Diagnostic messages include services such as periodically broadcasting active diagnostic trouble codes, identifying operator diagnostic lamp status, reading or clearing diagnostic trouble codes, reading or writing control module memory, providing a security function, stopping/ starting message broadcasts, reporting diagnostic readiness, monitoring engine parametric data, etc. (SAE, 2000)

3.4.1. Diagnostic Trouble Code Definition—

A Diagnostic Trouble Code (DTC) is made up of four (4) independent fields, as follows:

Table 3.5. Configuration of the Diagnostic Trouble Code

A.	Suspect Parameter Number	(SPN)	19 bits
B.	Failure Mode Identifier	(FMI)	5 bits
C.	Occurrence Counter	(OC)	7 bits
D.	SPN Conversion Method	(CM)	1 bits

These independent parameters are not used together to form a number. They are merely a set of information that helps in understanding the failure that is being reported.

A diagnostic tool may also want to use the controller source address and the Name to determine which controller is reporting the diagnostic information. This information is not needed to interpret the SPN but may be beneficial to have during the diagnostic process. Diagnostic trouble codes are transmitted as 4 bytes per trouble code. Those 4 bytes are interpreted as defined in section 5.7.1. In an effort to provide continuity between the diagnostics defined in SAE J1587 to that of SAE J1939-73, the fault encoding format remains very similar. When SAE J1587 PID numbers have been mapped one for one as SPNs. (SAE, 2000)
Examples of diagnostic trouble codes (see Table 2):

EXAMPLE 1—this is a SAE J1587 parameter.

SPN=91	Suspect parameter is accelerator pedal position
FMI=3	Failure mode is identified as voltage above normal
OC=5	Occurrence count indicates trouble has occurred 5 times
CM=	02 (1 bit)

EXAMPLE 2—this is not a parameter communicated as an SAE J1587 PID. Therefore, it is assigned a number above 511.

SPN= 656	Suspect parameter is engine injector number 6
FMI= 3	Failure mode is identified as voltage above normal
OC= 2	Occurrence count indicates trouble has occurred 2 times
CM= 02 (1 bit)	

EXAMPLE 3 Diagnostic Trouble Code as transmitted in diagnostic messages

Given: Parameter	“Pre-Filter Oil Pressure,”
Suspect Parameter Number	1208
Failure Mode Identifier	3
Occurrence Count	10
SPN Conversion Method	0

All fields of DTC sent in Intel Format (least significant byte first)

Table 3.6. Example DTC format

	Decimal	Hexadecimal	Binary
SPN	1208	$(4B8)_{16}$	$(000\ 00000100\ 10111000)_2$ (19 bits)
FMI	3	$(3)_{16}$	$(00011)_2$ (5 bits)
OC	10	$(A)_{16}$	$(0001010)_2$ (7 bits)
CM			0_2

Table 3.7. DTC Representation in CAN Data Frame for DM1

DTC																													
Byte 3, LSB of SPN								Byte 4, Second of SPN								Byte 5, 3 MSB bits of SPN, FMI								Byte 6							
SPN																		FMI					C	OC						
8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1

3.4.2. Active Diagnostic Trouble Codes (Dm1)

The information communicated is limited to the currently active diagnostic trouble codes preceded by the diagnostic lamp status. Both are used to notify other components on the network of the diagnostic condition of the transmitting electronic component. The data contains the lamp status and a list of diagnostic codes and occurrence counts for currently active diagnostic trouble codes. These all DTCs including those that are emissions related.

The currently defined lamps (Malfunction Indicator Lamp, Red Stop Lamp, Amber Warning Lamp, and Protect Lamp) are associated with DTCs. If the transmitting electronic component does not have active DTCs, then the lamp status from that component will indicate that the lamps should be off. However, the component controlling the actual lamp illumination must consider the status from all components that use these lamps before changing the state of the lamps. There may be applications that require additional lamp definitions to accomplish their function

(e.g. a lamp that indicates when cruise control is actively controlling would require a separate lamp in another PG). (SAE, 2000)

Transmission Rate: A DM1 message is transmitted whenever a DTC becomes an energetic fault and at a standard update rate of just one occasion per second thenceforth. If a fault has been active for one second or longer, then becomes inactive, a DM1 message shall be transmitted to replicate this action. If a distinct DTC changes state at intervals the one second update amount, a replacement DM1 message is transmitted to replicate this new DTC. To stop a high message rate owing to intermittent faults that have a really high frequency, it's counseled that no over one action per DTC per second be transmitted. therefore a DTC that becomes active / inactive double at intervals a one second interval, like shown in Example Case one, would have one message distinctive the DTC turning into active, and one at future periodic transmission distinctive it being inactive. This message is distributed only if there's an energetic DTC existing or in response to letter of invitation. Note that this Parameter cluster would force victimization the “multipacket Transport” Parameter cluster once over one active DTC exists.

Table 3.8. Specifications of the Diagnostic Messages

Data Length		Variable
Data Page		0
PDU Format		254
PDU Specific		202
Default Priorrity		6
Parameter Group Number		65226 (FECA)hex
Byte 1	Bits 8-7	Malfunction Indicator Lamp Status
	Bits 6-5	Red Stop Lamps Status
	Bits 4-3	Amber Warning Lamp Status
	Bits 2-1	Protect Lamp Status
Byte 2	Bits 8-7	Reserved for SAE
	Bits 6-5	Reserved for SAE
	Bits 4-3	Reserved for SAE
	Bits 2-1	Reserved for SAE
Byte 3	Bits 8-1	SPN, 8 LSB bits of SPN
Byte 4	Bits 8-1	SPN, Second byte of SPN
Byte 5	Bits 8-6	SPN, 3 MSB of SPN
	Bits 5-1	FMI
Byte 6	Bits 8	SPN Conversion Method
	Bits 7-1	Occurrence Count

EXAMPLE 1: The following illustrates the message format for when there is more than one diagnostic trouble code.

Given:

A	=	lamp status
B	=	SPN
C	=	FMI
D	=	CM and OC

Message form will be as follows: a,b,c,d,b,c,d,b,c,d,b,c,d.etc. In this example, the transport protocol of SAE J1939-21 will have to be used to send the information because it requires more than 8 data bytes. Actually any time there is more than one fault the services of the transport protocol will have to be used. (SAE, 2000)

EXAMPLE 2: The following illustrates the message format for when a request of the DM1 is made and there are zero active faults. In order for one of the currently defined lamps (Malfunction Indicator Lamp, Red Stop Lamp, Amber Warning Lamp, and Protect Lamp) to be on, an active DTC must be in existence. The original publication of this recommended practice defined that bytes 6 through 3 should be set to all ones when there are zero faults. This particular implementation is permitted but not preferred. Therefore, this is the Grandfathered Setting. Implementations are preferred to set bytes 6 through 3 to all zeros. (SAE, 2000)
Given:

Table 3.9. Message Format when a request of the DM1 message.

Byte 1	Bits 8-7	00
	Bits 6-5	00
	Bits 4-3	00
	Bits 2-1	00
Byte 2	Bits 8-7	11
	Bits 6-5	11
	Bits 4-3	11
	Bits 2-1	11

Table 3.10. SPN representation of DM1 Message Format

Grandfathered Setting		**Recommended Setting**
Byte 6-3	SPN = 524,287	Indicates not available = 0
	FMI = 31	Indicates not available = 0
	OC = 127	Indicates not available = 0
	CM = 1	Indicates not available = 0
Byte 7	255	255
Byte 8	255	255

3.4.3. Malfunction Indicator Lamp

A lamp used to relay only emissions-related trouble code information. This lamp is only illuminated when there is an emission-related trouble code active.

Table 3.11. Table of the Malfunction Indicator Lamp

00	Lamp Off
00	Lamp On
Type	Status
Suspect Parameter Number	1213

3.4.4. Red Stop Lamp

This lamp is used to relay trouble code information that is of a severe enough condition that it warrants stopping the vehicle.

Table 3. 12 Table of the Red Stop Lamp

00	Lamp Off
00	Lamp On
Type	Status
Suspect Parameter Number	623

3.4.5. Amber Warning Lamp

This lamp is used to relay trouble code information that is reporting a problem with the vehicle system but the vehicle need not be immediately stopped.

Table 3.13. Table of the Amber Warning Lamp

00	Lamp Off
00	Lamp On
Type	Status
Suspect Parameter Number	624

3.4.6. Suspect Parameter Number

This 19-bit number is used to identify the item for which diagnostics are being reported. The SPN is used for multiple purposes, some of those that are specific to diagnostics are:

1. To identify a least repairable subsystem that has failed;
2. To identify subsystems and or assemblies that may not have hard failures but may be exhibiting abnormal operating performance;
3. Identifying a particular event or condition that will be reported; and 4. to report a component and non-standard failure mode.

SPNs are assigned to each individual parameter in a Parameter Group and to items that are relevant to diagnostics but are not a parameter in a Parameter Group. SPNs are independent of the source address for the message. However, the source address may be necessary to determine which controller on the network performed the diagnosis. The first 511 SPNs are reserved and will be assigned the exact same number as the Parameter Identifier (PID) used in SAE J1587. That is, the SPN for an accelerator problem will be reported as SPN 91 which is SAE J1587 PID 91. All other SPNs will be numbered sequentially starting at 512 and incrementing by one for each new assignment. Proprietary Suspect Parameter Numbers have been established to allow the reporting of manufacturer specific diagnostics. The interpretation of the diagnostic trouble codes using proprietary SPNs varies by manufacturer. There are 4096 Suspect Parameter Numbers defined for proprietary

diagnostics. The SPNs for Proprietary Diagnostics cover the range 520192 to 524287. (SAE, 2000)

Table 3.14. Suspect Parameter Number

Data Length:	:	19 bits
Resolution:	:	1 SPN/bit
Data Range:	:	0 to 524,287
Type:	:	Status
Suspect Parameter Number:	:	1214

3.4.7. Failure Mode Identifier

Defines the type of failure detected in the subsystem assigned by an SPN. The fault may be an electrical failure or may be a subsystem failure. Combination of the SPN, FMI, Conversion Method and Occurrence Count Fields gives a diagnostic trouble code. The defined FMIs are shown in Appendix A.

Table 3.15. Failure Mode Identifier

Data Length:	:	5 bits
Resolution:	:	1 FMI/bit
Data Range:	:	0 to 31
Type:	:	Status
Suspect Parameter Number:	:	1215

3.5. Vehicle Diagnostic System

3.5.1. Engine Diagnostic System

Modern engine diagnostics consist of 2 separate parts, the off-board diagnostics and the on-board diagnostics. For both parts are new legal requirements with the introduction of EURO-IV and further. The focus of these legal requirements is on emissions.

On-board diagnostics (OBD): The main function of OBD is to check the emissions during the full life of the engine. If the emissions are above a certain limit then the driver MUST be warned by this system. DAF has put this strategy in both the engine controller (DMCI) and in the after-treatment controller (EAS). These controllers can independently generate a warning lamp to the driver (Malfunction Indicator or MIL). These warnings MUST be shown to the driver to fully comply to the legal requirements and this can only be done by reading the lamp statuses out of Byte #1 of the DM1 CAN-message from the DMCI and the DM1 CAN-message from the EAS.

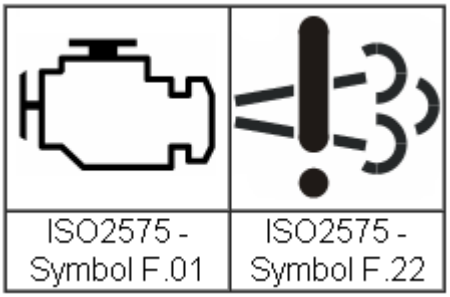
As from 1 October 2009 the new OBD2 regulations came in to force. Generally, this means the following additional regulations with regard to the OBD1+ regulations:

- more extensive Engine system NOx control (addition of NOx sensors)
- more extensive reagent control (some errors are assigned to "non-erasable" fault codes and cannot be erased by any diagnostic tool but will be erased automatically after 400 days or 9600 hours of engine operation)
- measures to discourage tampering of exhaust after treatment systems (40% torque de-rate if emission levels are too high)

According to directive 2005/78/EC the OBD system must incorporate a malfunction indicator readily visible to the vehicle operator and this MI must not be

used for any purpose other than emission related malfunction except to indicate emergency start-up or limp-home routines to the driver. Safety related messages can be given the highest priority. The MI must be visible in all reasonable lighting conditions. When activated, it must display a symbol in conformity with ISO 2575 as a dashboard telltale lamp or a symbol on a dashboard display.

A vehicle must not be equipped with more than one general purpose MI for emission-related problems. The use of red for the MI is prohibited. One of the following symbols must be displayed:



According to directive 2005/78/EC and 2006/51/EC in some occasions the driver MUST be alerted by an additional indication together with the MI. Therefore, a distinct warning mode MUST be activated, e.g. flashing MI or activation of a symbol in addition to MI activation. The additional indications MUST be activated according to the table below:

Source	Active DTC received via DM1 message	Action
3Dh (EAS)	SPN=520229 & FMI=01 or SPN=520495 & FMI=07	Driver must be informed by a distinct warning that the reagent tank is empty.
3Dh (EAS)	SPN=520271 & FMI=18	Driver must be informed by a distinct warning that the reagent tank level is low.
00h (DMCI)	SPN=1787 & FMI=11	Driver must be informed by a distinct warning that the torque limiter is activated.

An engine on the vehicle includes lots of sensors, mechanical parts, valves, plugs, motors. Engine ECU controls the all parts of the engine. Also it includes the control software and parameters of the electrical parts. It is possible to control and to see these parameters and controls via CAN. The informations about electronic parts, is located on engine, like accelerator pedal level sensor, coolant level sensor, camshaft speed sensor, engine speed sensor, engine temperature sensor is broadcasted on the CAN data link. The broadcasting is based on SAE J1939 standards. Engine broadcasts the failure about the electronic components via J1939 such as broadcasted the informations about the components. These failure codes are separated as active and passive failure codes. Engine sends below informations in the diagnostic messages.

SPN	(Suspect parameter number)	: The component that has failure.
FMI	(Failure mode identifier)	: Type of the failure on the component
OC	(Occurrence Count)	: Indicates trouble how many time occured

Above informations are broadcasted according to the J1939-73 standard. An example about the Engine Diagnostic message is shown below.

SPN	= 91	Suspect parameter is accelerator pedal position
FMI	= 3	Failure mode is identified as voltage above normal
OC	= 5	Occurrence count indicates trouble has occurred 5 times

gine sends the diagnostic messages via CANID that is shown below;

18 FE CA 00

By using this message structure, engine can send only four DTC. If there are more than four fault codes, it exceeds the one message structure. This time J1939-21

transport protocol starts up. Engine sends the fault codes according to the transport protocol via below CANID.

18 EB FF 00

18	:	priority of the message
EBFF	:	PGN number of the message
00	:	Source address of the message

An error transmitting on the transport protocol is shown below. The PGN 60415 starts up and the error codes are broadcasted respectively.

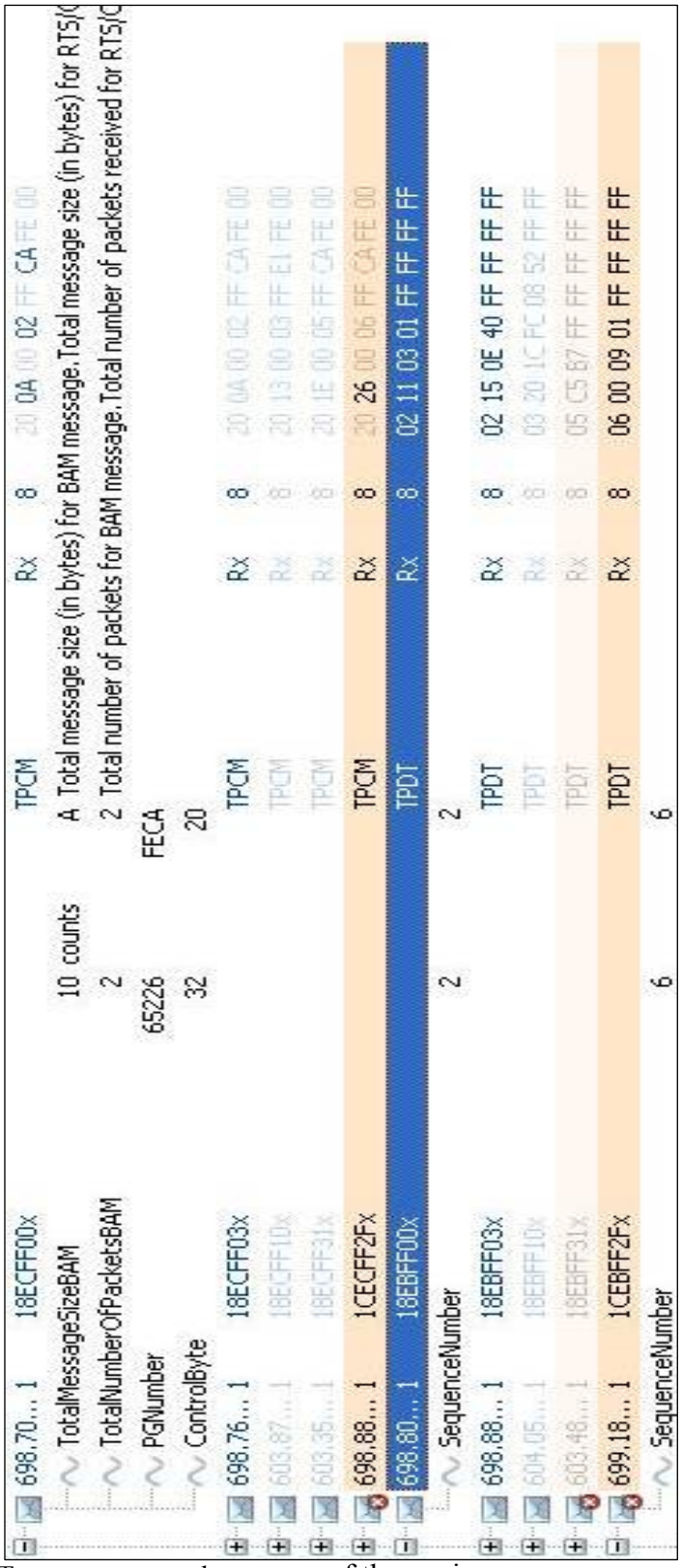


Figure 3.8. Transport protocol message of the engine

Below a CAN data link layer screen shot is shown. These screen shows engine fault codes information like SPN, FMI.

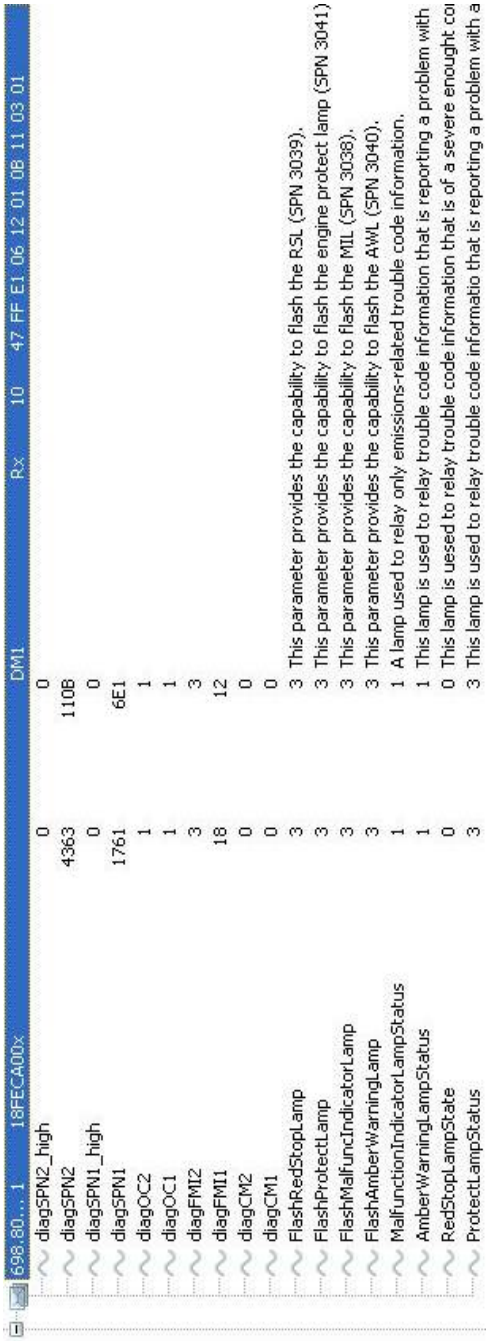


Figure 3.9. Real-time representation of Engine DTC information

3.5.2. Brake Control System Diagnostic

Failures in the brake system are displayed according to their influence on the brake performance on a red and a yellow warning lamp.

- If the red warning lamp is switched on, the brake performance might be reduced.
- If the yellow warning lamp is switched on, the brake performance is not reduced, but control function might be switched off
- some failures are displayed only temporarily, others stay on until the next ignition reset and start check
- if failures (that have occured before switching off the ignition) could not be checked during start check, a memory failure is set and the warning lamp stays on, until the affected function has been checked during operation

KWP2000 diagnosis (faulty component and failure mode) is done via ISO 9141 Kline (Off board). The failure code (SPN) is also transmitted via SAE J 1939-73 (DM1- message)

For failure-lamp display (red, yellow, linings worn-out), then following EBS-information is available (see table below). The vehicle-manufacturer must assure, that the relevant failure-information is displayed to the driver.

Brake control system ECU (ABS or EBS) consists of information about brake system's speed sensors, valves and modulators. For example an ABS system has minimum four speed sensors, four air valves, and one ASR valve. According to the number of the axes, numbers of these ABS components are changed. ABS ECU broadcasts CAN messages that consist of failures about the brake system components. Such as engine system, brake ECU also sends the messages according to J1939 diagnostic layer and transport protocol. Only the difference is the CANID of the messages. The brake system CANIDs are shown below.

18 FE CA 00 18 EB FF 00

An example of brake system diagnostic message is shown below.

SPN=1 Suspect parameter is front left wheel speed sensor

FMI=1 Failure mode is identified as air gap too large, sensor output voltage too low but just exceeds trigger level

OC=5 Occurrence count indicates trouble has occurred 5 times

A CAN message log screen shot is shown below. In the below screen, ABS fault message is shown.

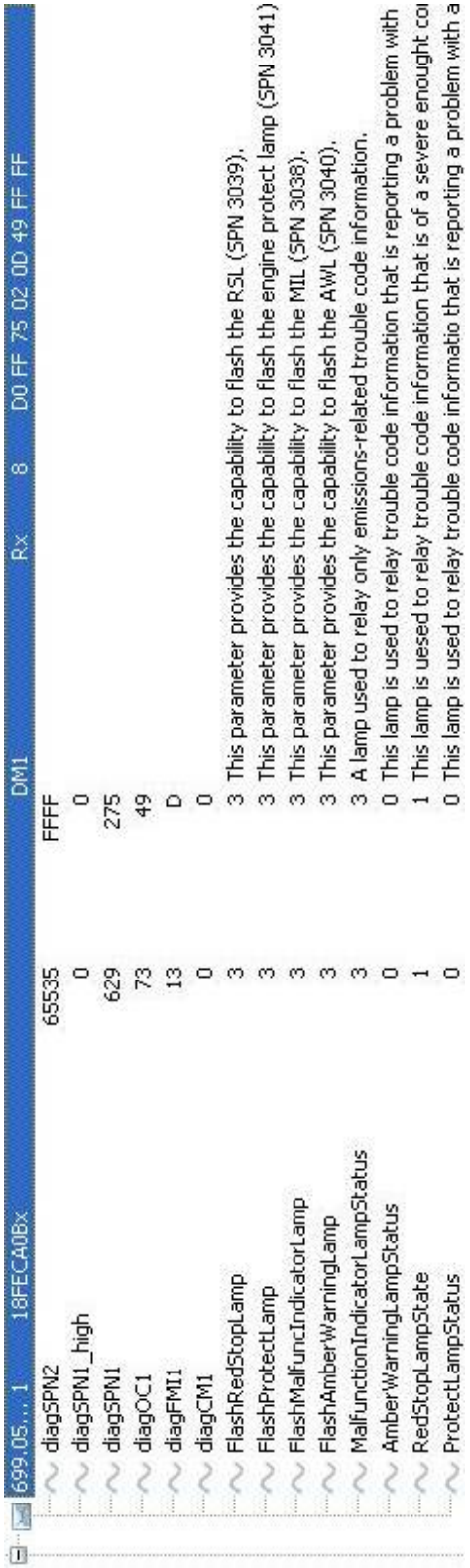


Figure 3.10. ABS Diagnostic message viewed by using CAN Analyzer

3.5.3. Suspension Control System (ECAS) Diagnostic

ECAS is an electronically controlled air suspension system on the road vehicles. In this air suspension system, level sensors, air system valves and an electronic control unit are existed. ECAS ECU is also broadcasts the fault informations about the suspension system components. Such as other system ECAS ECU also sends the messages according to J1939 diagnostic layer and transport protocol. Only the diffrence is the CANID of the messages. The suspension system CANIDs are shown below.

1C FE CA 2F 1C EB FF 2F

An example of ECAS suspension system diagnostic message is shown below.

SPN	=	34	Suspect parameter is common supply for solenoid valves
FMI	=	3	Failure mode is identified as voltage above or shorted to ground
OC	=	3	Occurrence count indicates trouble has occurred 3 times

A CAN message log screen shot is shown below. In the below screen, ECAS fault message is shown.

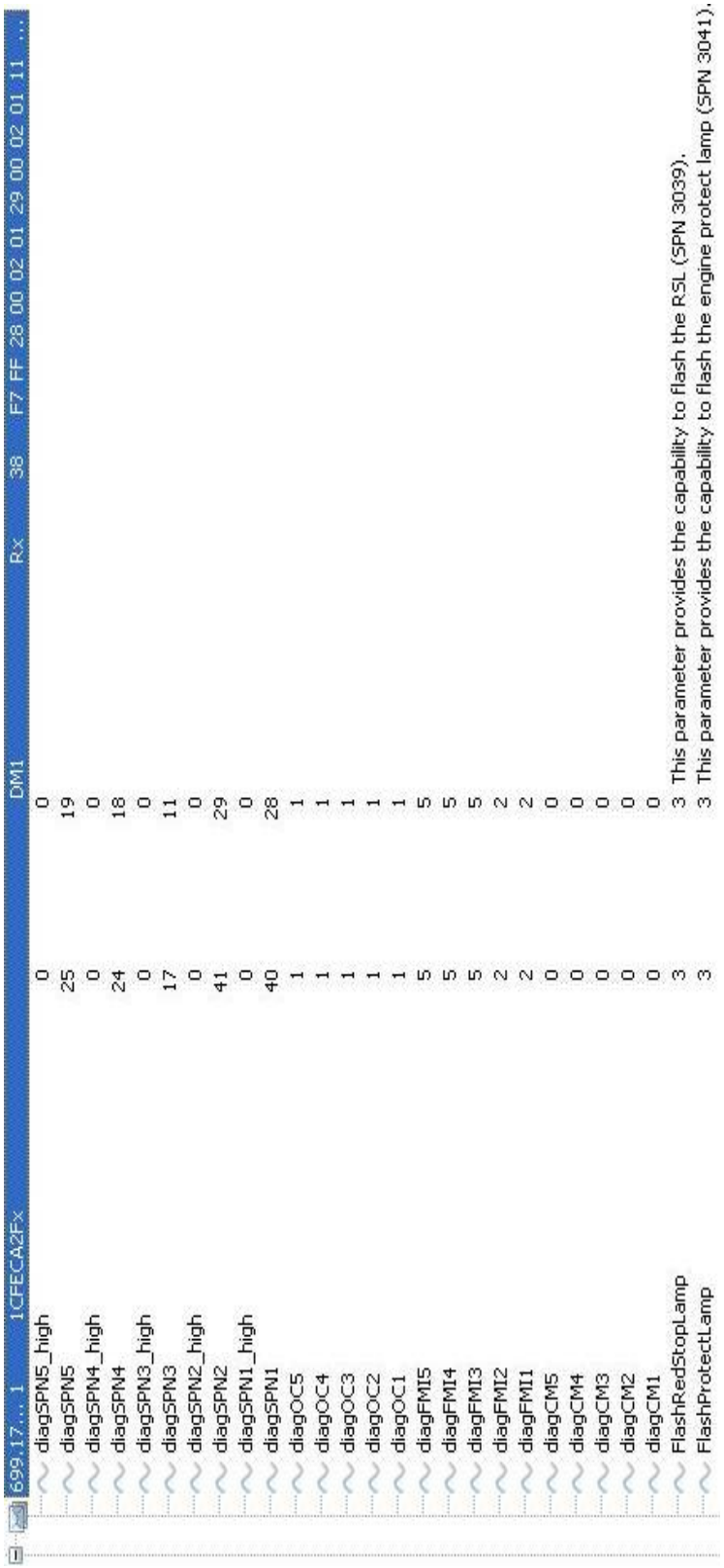


Figure 3.11. ECAS Diagnostic message viewed by using CAN Analyzer

3.5.4. Transmission Control System (TCU) Diagnostic

Transmission systems broadcast the diagnostic messages via CAN by applying J1939 standard. Some differences are shown in transmissions while sending the diagnostic messages. Some of transmission control systems broadcast the diagnostic messages according to J1939-73 and some of them broadcast the messages according to the SAE J2012.

If TCU is based on J1939-73 standard, using CANID defined as 18FECA03 sends the diagnostic messages. If there are more than four faults codes, TCU uses 18EBFF03 as CANID. An example J1939-73 based diagnostic message of TCU is shown below.

SPN =26 Suspect parameter is Lockup_Clutch_Short_GND
FMI =3 Failure mode is identified as voltage above or shorted to ground
OC =3 Occurrence count indicates trouble has occurred 3 times

A CAN message log screen shot is shown below. In the below screen, TCU fault message is shown.

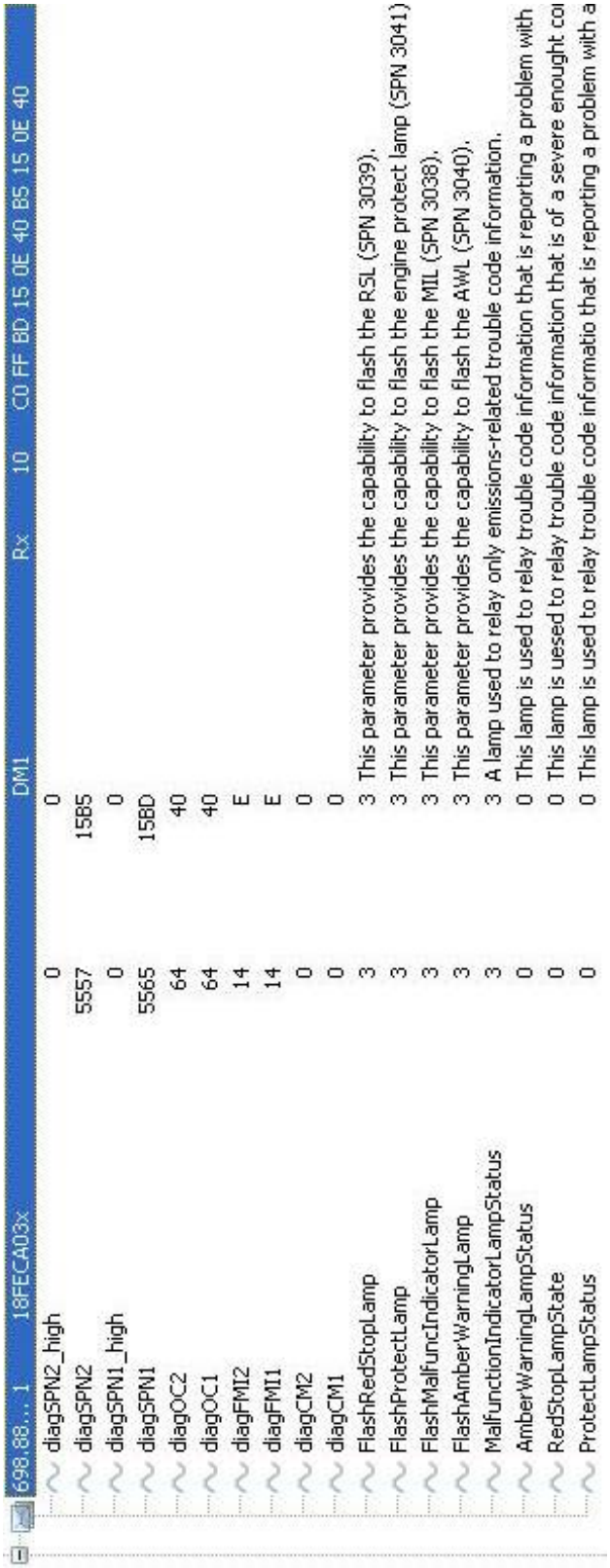


Figure 3.12. TCU Diagnostic message viewed by using CAN Analyzer

If TCU is based on SAE J2012 standard, diagnostic message broadcasting is defined by below steps;

1. Request PGN: TCU broadcast diagnostic messages, if it is requested.
2. TCU broadcast the diagnostic messages according to J1939-21 layer.
3. DTC is defined by detailing the J2012 message.

SAE J2012 message structure is shown below;

Data byte arrangement:
A B11 B21 B31 B41 B51 C1 B1X B2X B3X B4X B5X CX . . .
Where,

Table 3.16. DTC representation of Transmission according to SAE J2012

A	Bit 8-1	Number of J2012 DTCs
$B1_X$	Bit 8-1	1st Character of J2012 DTC
$B2_X$	Bit 8-1	2nd Character of J2012 DTC
$B3_X$	Bit 8-1	3rd Character of J2012 DTC
$B4_X$	Bit 8-1	4th Character of J2012 DTC
$B5_X$	Bit 8-1	5th Character of J2012 DTC
C_X	Bit 8:	J2012 DTC Status
	Bits 7-1:	J2012 DTC Occurrence Count

If PGN 64906 is requested and a supporting device has no active or inactive J2012 DTCs, PGN 64906 shall be sent as a single frame message with the first data byte (Number of J2012 DTCs) set to zero. Any unused bytes in this PGN shall be set to 255. When two or more J2012 DTCs are indicated, PGN 64906 must be sent via Transport Protocol.

EXAMPLE – A device conveying (1) an active P1482 with 9 counts and (2) an inactive U0100 with 4 counts would populate the data bytes as follows:

Table 3.17. An example of Transmission Diagnostic Message

Data Byte	Dec Value	ASCII Value	Data Byte	Dec Value	ASCII Value
1	2	--	8	85	“U”
2	80	”P”	9	48	“0”
3	49	“1”	10	49	“1”
4	52	“4”	11	48	“0”
5	56	“8”	12	48	“0”
6	50	“2”	13	4	--
7	137 –				

Multiple DTCs present

Table 3.18. An example of Transmission Multiple Diagnostic Message

Byte:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Hex:	#02	#50	#30	#32	#31	#38	#7F	#50
Dec:	2	80	48	50	49	56	127	80
ASCII:	“P”	“0”	“2”	“1”	“8”	--	“P”	--

Byte:	**9**	**10**	**11**	**12**	**13**
Hex:	#30	#37	#33	#33	#FF
Dec:	48	55	51	51	255
ASCII:	“7”	“3”	“3”	--	

Byte 1 indicates there are 2 DTCs present, Byte 7, bit 8 indicates the 1st DTC is inactive, and Byte 13, bit 8 indicates the 2nd DTC is active.

P0218 : Transmission Fluid Over temperature

P0733 : Incorrect 3rd Gear Ratio

3.5.5. Vehicle Diagnostic Message Configuration

It is possible to see many failure codes on different ECUs in the vehicle. All these failure codes exist on the vehicle CAN bus data link. It is very important to make meaningful these fault codes for the technicians and end user of vehicles. For these purpose, a fault code configuration has been designed on the vehicle CAN data link. This fault configuration is formed by below steps;

1. Connect to the CAN data link and read the fault codes from the ECUs.
2. Specify a vehicle fault code for the each ECU fault code
3. Broadcast the fault codes to the CAN data lin

Vehicle Diagnostic Message Configuration is shown below.

Table 3.19. Vehicle CAN fault code configuration

SPN	FMI	VEHICLE FAULT CODE	ECU	DESCRIPTION
3363	3	284	ENGINE	Aftertreatment Diesel Exhaust Fluid Tank Heater - Voltage above normal, or shorted to high source.
17	0	3215	TRANSMISSION	DR_MAIN_PRESSURE_UNKNOWN
253	13	4647	BRAKE	ECU does not accept valve system of harness.
22	5	284	ECAS	Pressure sensor front axle right - Current below normal or open circuit

The vehicle CAN fault code is sent to the CAN data link and it is used to show the code on the vehicle instrument panel and for sending to the remote center. The code libraries in the service technicians are found and the technicians have troubleshooting for the vehicle fault.

There was an additional diagnostic ECU requirement to determine the fault codes of the electronics (engine, transmission, etc.), because of there is a limit in the body controller of the vehicle. Body controller can only determine 200 diagnostic trouble codes from the CAN data link of the vehicle. There is possibility to see over than two thousand fault codes. A new electronic control design is needed to get all fault codes from CAN data link. This diagnostic ECU will connect to the two different CAN data link of the vehicle (125kbit/s and 250kbit/s). ARM microcontrollers are very popular to solve this CAN connection problem.

3.5.6. The ARM Controller

On 26 April 1985, the first ARM prototypes arrived at Acorn Computers Limited in Cam California. A few hours later they were running code, and a bottle of Moet & Chan-don was opened in celebration. For the remainder of the 1980s the ARM was quietly developed to underpin Acorn's desktop products which form the basis of educational computing in the UK; over the 1990s, in the care of ARM Limited, the ARM has sprung onto the world stage and has established a market-leading position in high-performance low-power and low-cost embedded applications.

This prominent market position has increased ARM's resources and accelerated the rate at which new ARM-based developments appear.

The highlights of the last decade of ARM development include:

- The introduction of the novel compressed instruction format called 'Thumb' which reduces cost and power dissipation in small systems;
- Significant steps upwards in performance with the ARM9, ARM 10 and 'Strong- ARM' processor families;
- A state-of-the-art software development and debugging environment;
- A very wide range of embedded applications based around ARM processor cores.

Most of the principles of modern SoC and processor design are illustrated somewhere in the ARM family, and ARM has led the way in the introduction of some concepts (such as dynamically decompressing the instruction stream). The inherent simplicity of the basic 3-stage pipeline ARM core makes it a good pedagogical introductory example to real processor design, whereas the debugging of a system based around an ARM core deeply embedded into a complex system chip represents the cutting-edge of technological development today. For example, variants of the LPC2000 are available with up to 4 independent CAN controllers on board the chip. The CAN controllers are one of the more complicated peripherals on the LPC2000. In the LPC2000, each CAN controller has a number of status and control registers plus three transmit buffers and a receive buffer. (Furber, 2000)

Table 3.20. LPC2000 CAN controller registers

CAN Control Registers	Name	Description
Rx Buffer	CAN MOD	Controls operating mode
	CAN CMR	Command register
Tx1 Buffer	CAN GSR	Global status register
	CAN ICR	Interrupt Status
Tx2 Buffer	CAN IER	Interrupt Enable
	CAN BTR	Bit Timing Register
Tx3 Buffer	CAN EWL	Error Warning Limit
	CAN SR	Status Register

Typical CAN controller schematic diagram of ARM microcontroller is shown below.

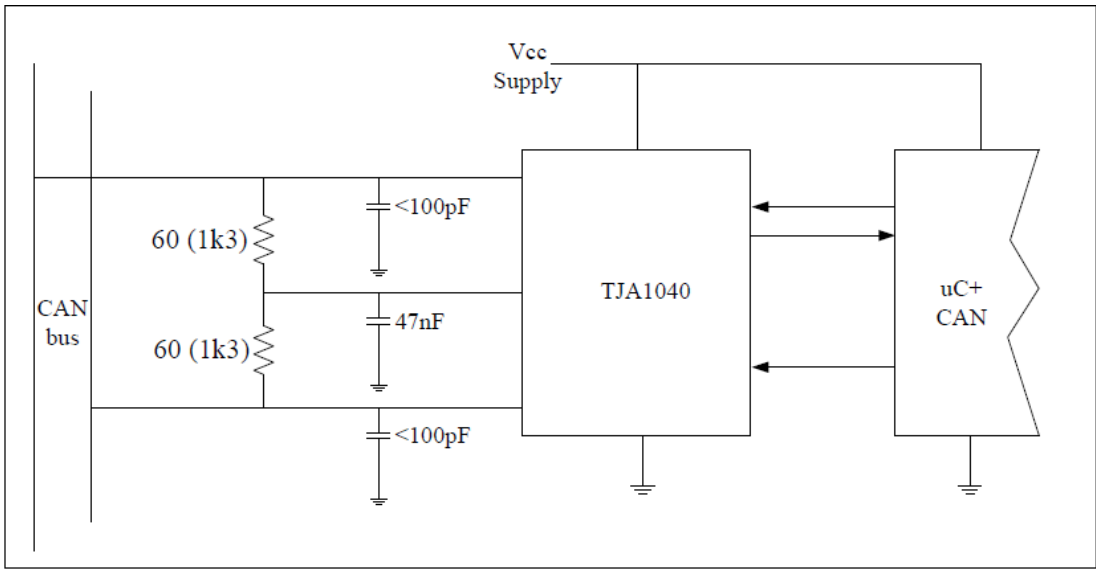


Figure 3.13. Typical CAN connection of the microcontroller

3.5.7. How the Diagnostic ECU is connected to CAN bus of the vehicle?

There is nearly six or seven CAN data line on the vehicle. Diagnostic ECU reads the data from the powertrain components or air conditioner and door systems. To ensure these connections, ECU must be connected to the PCAN and KCAN lines. PCAN and KCAN are separate data buses on the vehicle. Baudrate of the PCAN is 250kbits/sec and KCAN is 125kbits/sec. Because of these two separate CAN lines, the diagnostic ECU must have two CAN ports. The CAN port connection of the ECU is shown in Figure 3.14. In this design there are no need 120 Ohm resistors. Because CAN bus of the vehicle has already 60 Ohm impedance.

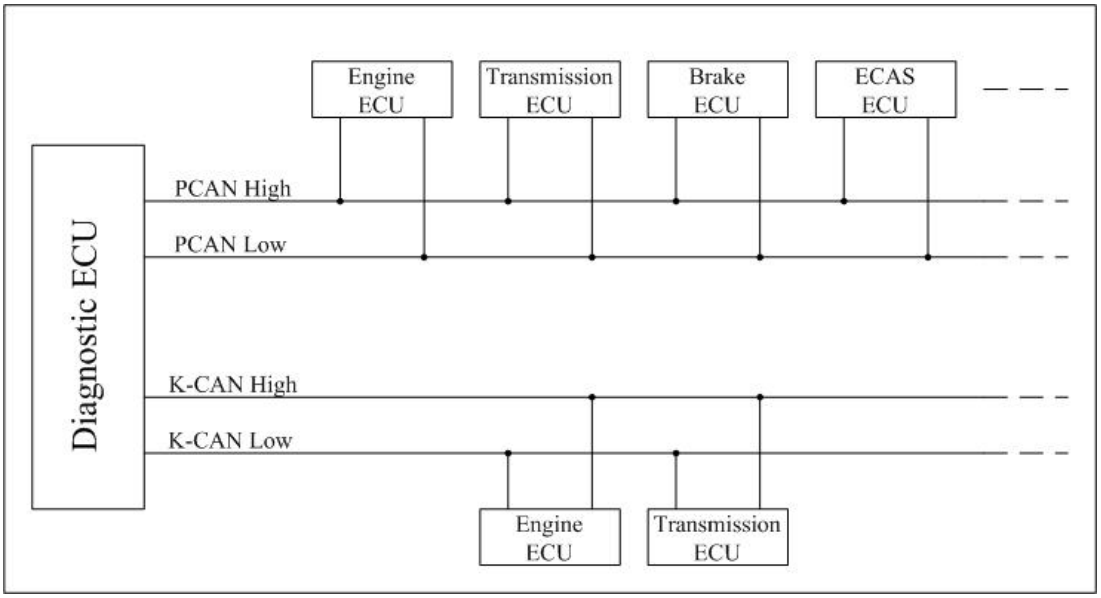


Figure 3.14. Connection of the Diagnostic ECU on the vehicle

3.5.8. How the Diagnostic ECU is connected to GPRS modem?

RS232 serial port communication is used between GPRS modem and Diagnostic ECU to transmit datas. It supports data transmission whose speed is 300 bps. It uses Modem AT command set. It also includes GPS module. Serial port connection diagram is shown in Figure 3.15.

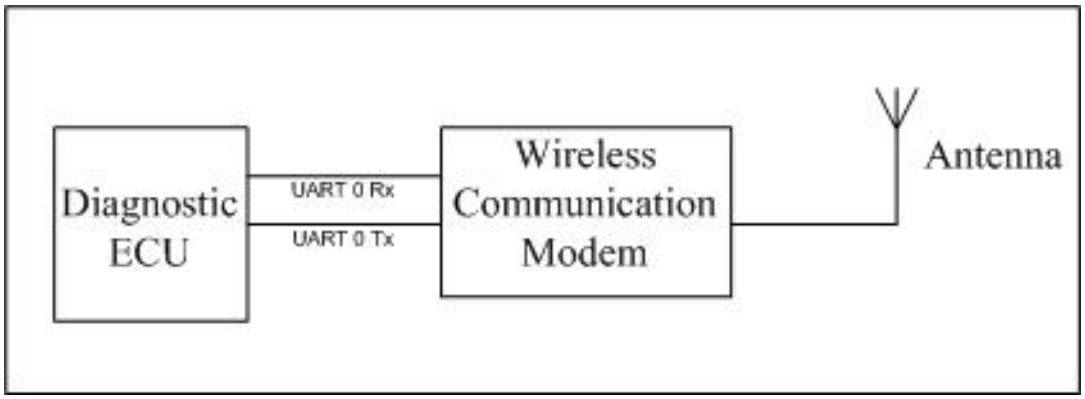


Figure 3.15. Schematic of the Diagnostic Message sending to the Remote Center

3.5.9. How Diagnostic ECU determines the error codes?

Diagnostic ECU is connected to the vehicle P-CAN and K-CAN of the bus. There are over ten ECUs on the vehicle CAN data bus. Diagnostic ECU has two CAN port. One of these ports gets the information from ABS ECU, Engine ECU, Transmission ECU, Body Controller, etc. And the other one gets the information from the Air Conditioner ECU, Door Control ECU, etc. The flowchart of the error detection in the P-CAN data bus is shown below in Figure 3.16.

Implementation software of the Diagnostic ECU has been developed by using Keil uVision software. A CAN message transmitting software can be shown below.

```
*----------------------------------------------------------------------------
        Write a message to CAN peripheral and transmit it.  CAN controller      (1.2)
 *----------------------------------------------------------------------------*/
void CAN_wrMsg (uint32_t ctrl, CAN_msg *msg)  {
  LPC_CAN_TypeDef *pCAN = (ctrl == 1) ? LPC_CAN1: LPC_CAN2;
  uint32_t CANData;
  CANData = (((uint32_t) msg->len) << 16)     & 0x000F0000 |
          (msg->format == EXTENDED_FORMAT ) * 0x80000000 |
          (msg->type   == REMOTE_FRAME)     * 0x40000000;

  if (pCAN->SR & (1<<2))  {                          /* Transmit buffer 1 free */
    pCAN->TFI1  = CANData;                           /* Write frame informations */
    pCAN->TID1 = msg->id;                           /* Write CAN message identifier
*/
    pCAN->TDA1 = *(uint32_t *) &msg->data[0];    /* Write first 4 data bytes */
    pCAN->TDB1 = *(uint32_t *) &msg->data[4];    /* Write second 4 data bytes */
    //pCAN->CMR  = 0x31;                           /* Select Tx1 for Self Tx/Rx */
    pCAN->CMR  = 0x21;                           /* Start transmission without
loop-back */
```

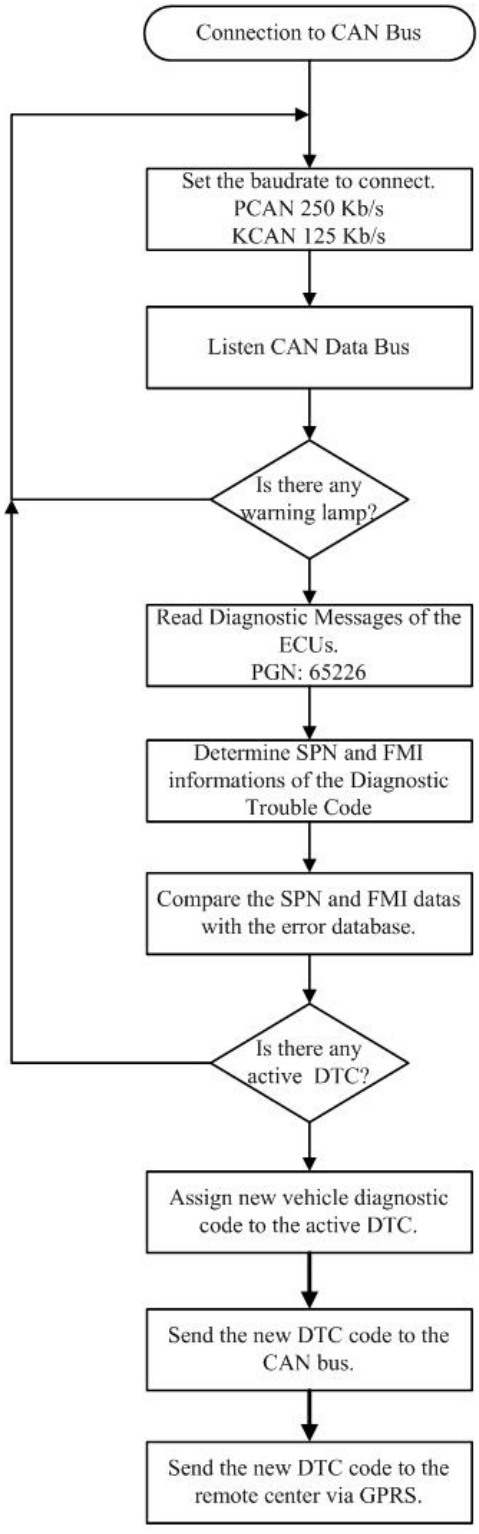


Figure 3.16. The flowchart of the error detection in the P-CAN

3.5.10. how Diagnostic ECU sends the error codes to the remote server?

GPRS is a packet based communication service for mobile devices that allows data to be sent and received accross a mobile phone network. GPRS is a step towards 3G and is often referred to as 2.5G. Benefits of the mobile modem:

- GPRS/3G is packet switched. Higher connection speed is realistic.
- GPRS/3G is an always-active service. There is no need to activate by a trigger.
- GPRS/3G enables full Internet applications.
- GPRS/3G provides lower data sending cost.

An error sending procedure via GPRS/3G modem is shown below in Figure 3.17.

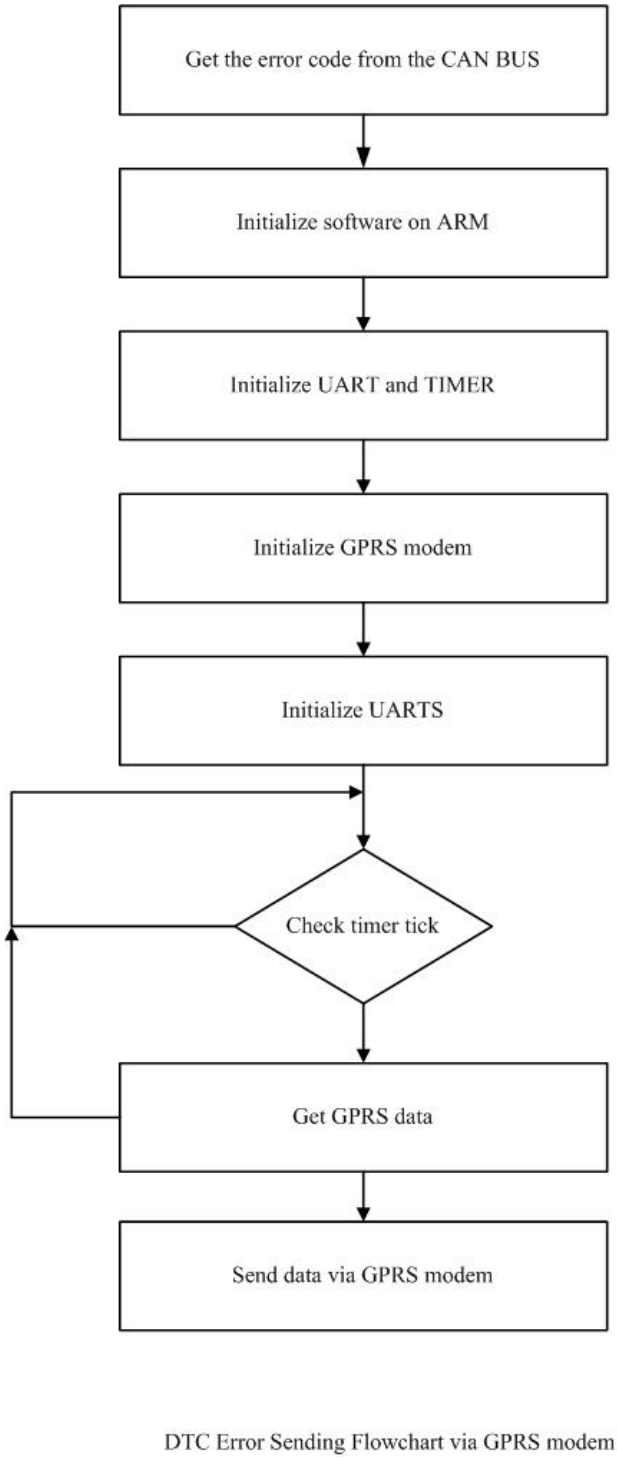


Figure 3.17. An error sending procedure via GPRS/3G modem

4. RESULTS AND DISCUSSION

As a result of this study, a remote vehicle diagnostic code determination project has been achieved successfully. Several ECUs were investigated to get the error codes of the ECUs. And several vehicle CAN bus lines were investigated to check the different vehicle configurations. The REDI was run on microcontroller to get the CAN bus line of the vehicles. Simulation was done by using CAN analyzer software. Nearly ten ECU message matrix has been investigated and a CAN bus diagnostic message map has been prepared. Microcontroller has read the data bus according to the message map and it produced a diagnostic trouble code which is special for the user and this code has been sent via mobile system to the remote help center. The simulated system has been tested on the different vehicles. An overload on CAN bus has been never seemed.

With this study, performance of the ARM microcontrollers on the CAN bus has been tested. Detailed studies have been done related with the ARM architecture. J1939 CAN bus standard has been investigated and a sofware was designed to read/write to the CAN bus of the vehicle. And also microcontroller software has been prepared to determine the failure code of the vehicle and to send the error code to the remote user.

Determined vehicle diagnostic code from the vehicle is shown below in Figure 4.1. According to the configurations of the ECUs, the diagnostic standarts which are KPW2000, SAE J1939-73 and SAE J2012, has been investigated and SPN and FMI information's are learnt. An example of SPN and FMI information's of the vehicle is shown in Figure 4.2.

Table 4.1. Vehicle diagnostic message

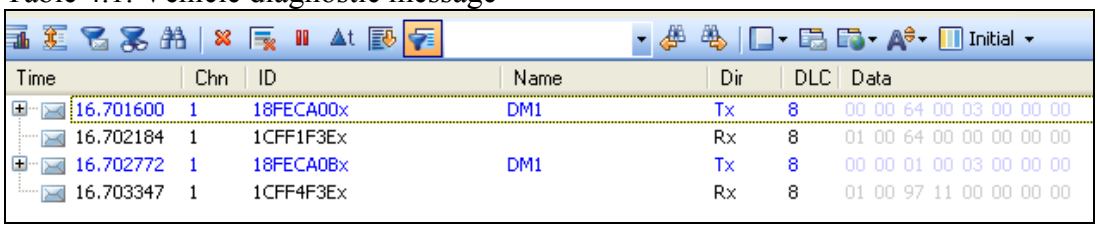


Table 4.2. Example vehicle diagnostic trouble code

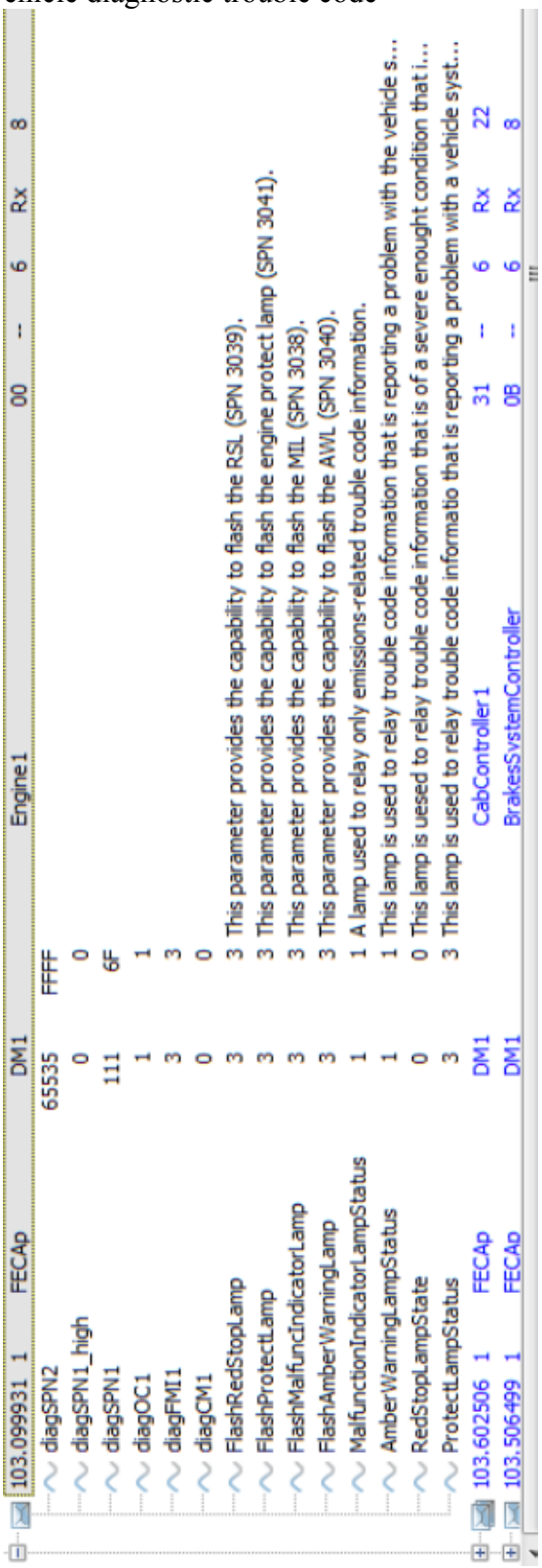


Figure 4.1 shows the test bus, which remote diagnostic system has been tested and it is a fully equipped with J1939 based ECUs.



Figure 4.1. Remote Diagnosis Test Bus

Test hardware of the remote diagnostic system has been installed on the test bus to check system on the real automotive world. Figure 4.2 shows the installation of the test hardware on the bus.

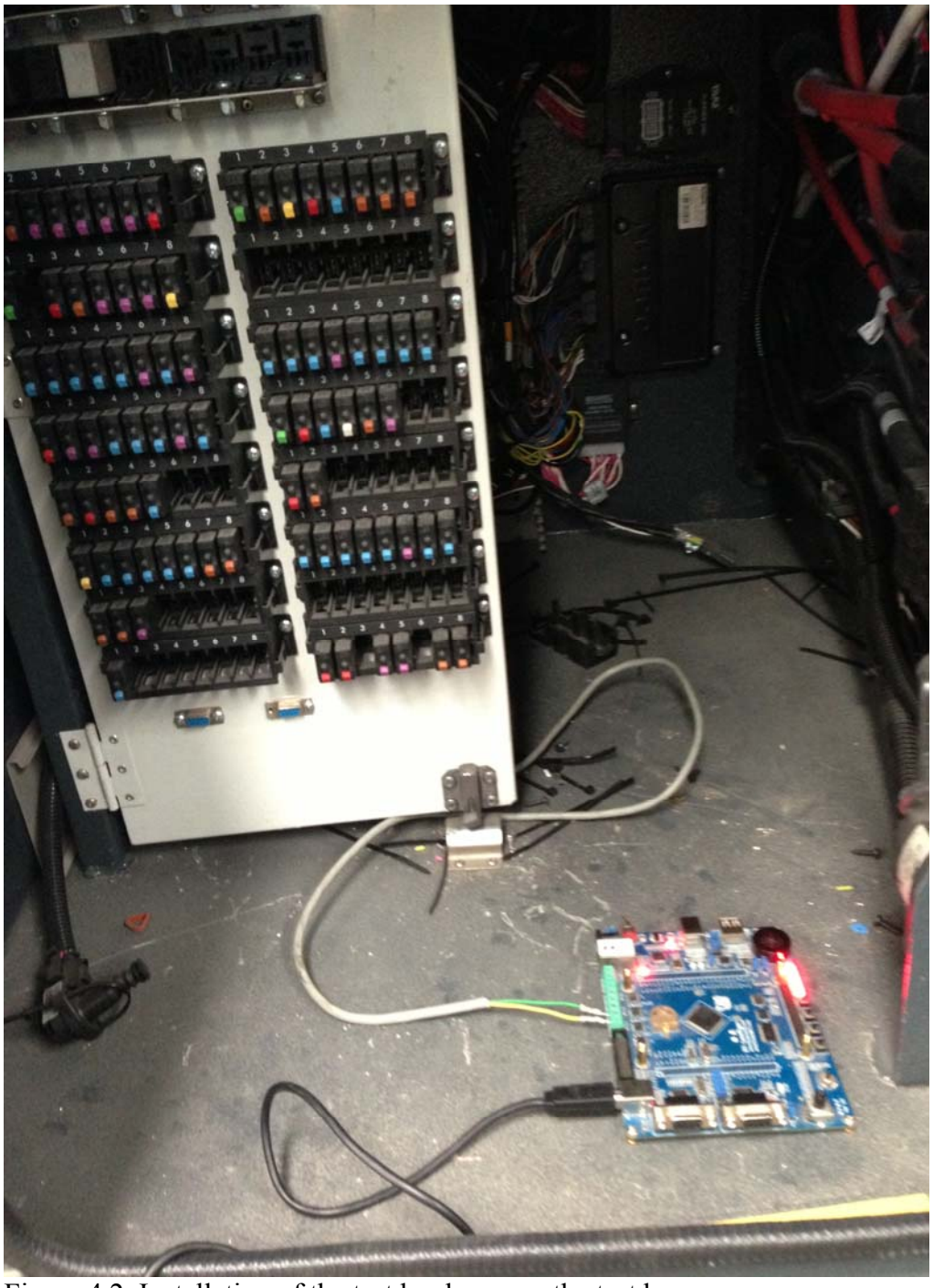


Figure 4.2. Installation of the test hardware on the test bus

A test scenario has been designed to realize the remote diagnostic system. A water level sensor, which ised to measure the coolant water level of the engine, is installed on the engine. If there is a fault on this sensor, engine gives a fault code. If you do not get water level, engine temperature can be increase. It is very hard to find the reason of the failure, if there is no any fault code description. As seen in below table if there is a problem on the coolant level sensor, engine will give below SPN and FMI informations. Description of the fault is also shown below. For example if sensor is unplugged from the level tank, engine will send SPN 111 and FMI 4 information to CAN data bus.

Table 4.3. Engine Coolant Level Sensor Fault Description

SPN	FMI	TYPE	FAULT DESCRIPTON
111	3	CUMMINS	Coolant Level Sensor 1 Circuit -Voltage above normal, or shorted to high source. High signal voltage detected at engine coolant level circuit.
111	4	CUMMINS	Coolant Level Sensor 1 Circuit -Voltage below normal, or shorted to low source. Low signal voltage detected at engine coolant level circuit.
111	18	CUMMINS	Coolant Level - Data Valid But Below Normal Operating Range - Moderately Severe Level. Low engine coolant level detected.
111	1	CUMMINS	Coolant Level - Data valid but below normal operational range - Most Severe Level.
111	17	CUMMINS	Coolant Level - Data Valid But Below Normal Operating Range - Least Severe Level. Low engine coolant level detected.
111	18	CUMMINS	Coolant Level - Data Valid But Below Normal Operating Range - Moderately Severe Level.

If engine has a fault as described above, Diagnostic ECU will get the SPN number 111 and FMI number 4 and it will broadcast the related vehicle diagnostic trouble code as shown in Table 4.4. The DTC code is shown on the instrument cluster of the bus as shown in Figure 4.3.

Table 4.4. Engine Coolant Level Sensor Diagnostic Trouble Code Definirtion

SPN	FMI	DTC	TYPE	DESCRIPTION
111	4	145	CUMMINS	Coolant Level Sensor 1 Circuit -Voltage below normal, or shorted to low source. Low signal voltage detected at engine coolant level circuit.



Figure 4.3. Display of the Diagnostic Trouble Code in Instrument Cluster of bus

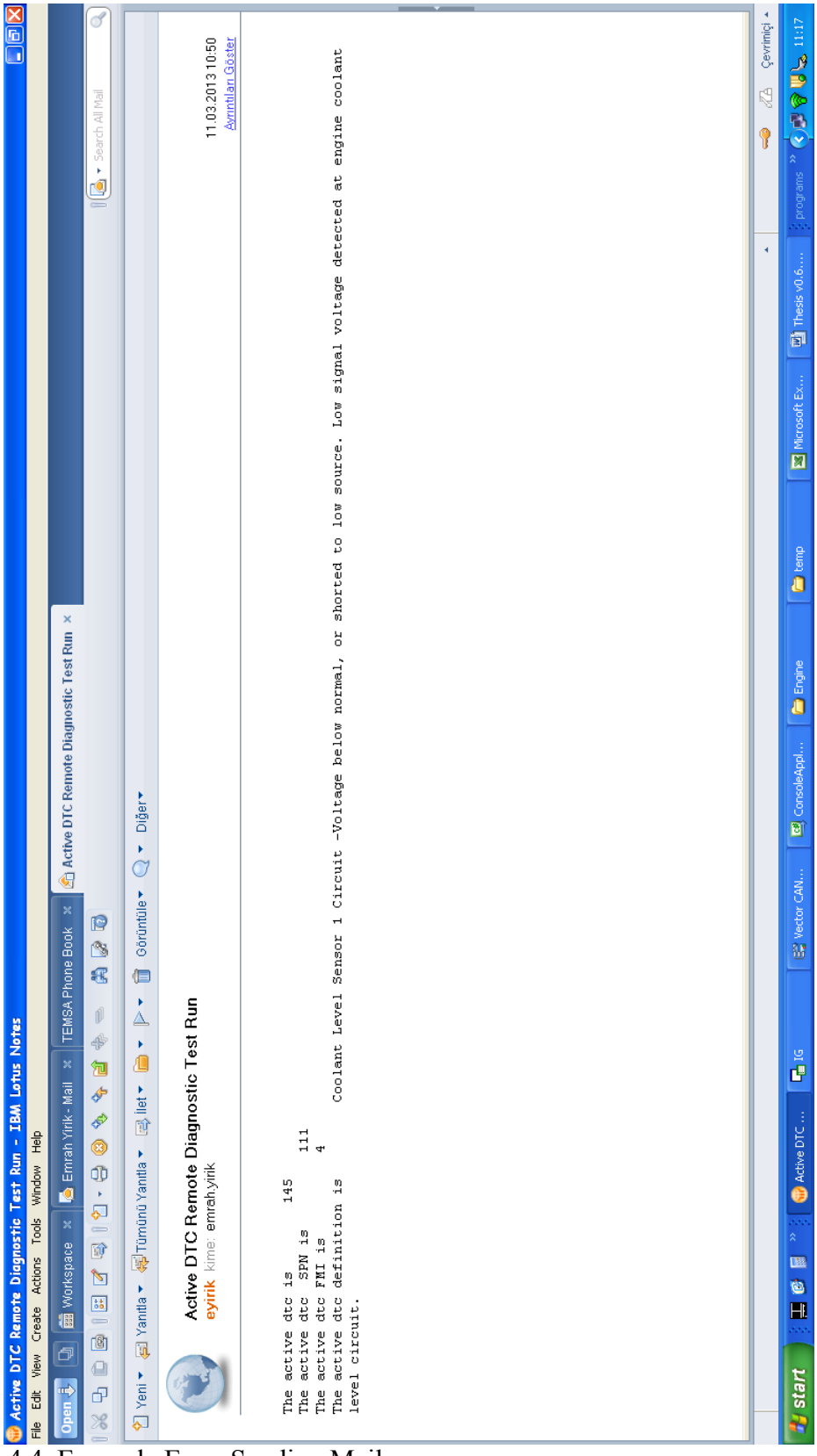


Figure 4.4. Example Error Sending Mail

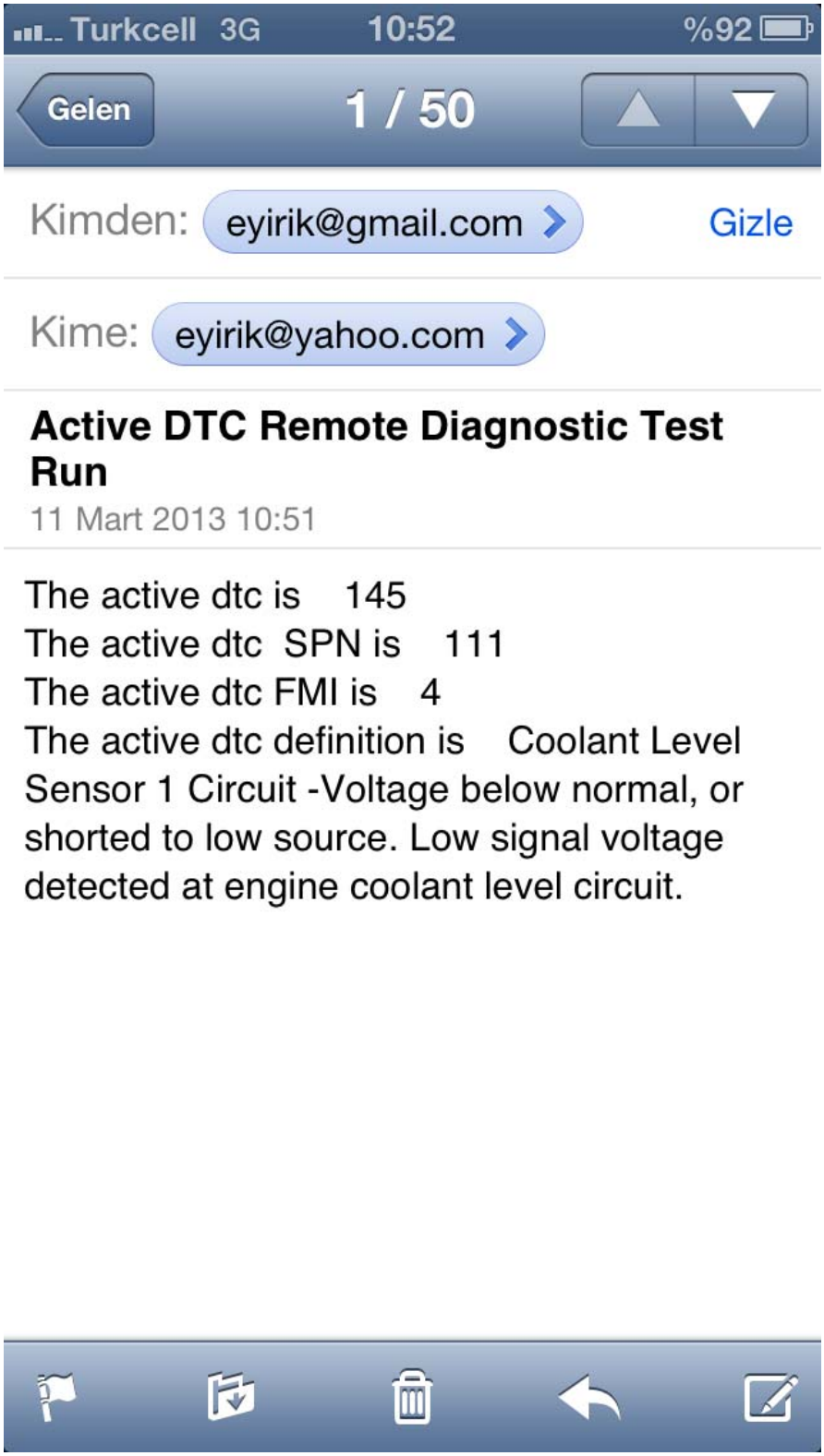


Figure 4.5. Example Error Sending to Mobil Phone

5. CONCLUSION

On this work, the components which is based automotive CAN bus technology, structure of the remote data verification unit were investigated by using present technologies and answered to the question: “how should be determined the failures of the vehicle which is run on the field”. At the end of this study, a sample remote diagnostic system has been composed by gathering the systems.

With this work, hardware and software for data collection from the vehicle CAN bus line, GPRS/3G modem to send the data from vehicle to remote user has been used on vehicle were recorded by using CANanalyzer software on PC. The software was obtained by overcoming some issues and difficulties encountered during improvements studies and solutions to be found. It has seemed that software and hardware can be applicable to the all commercial vehicles which are using most common OEM products. The bus which device is mounted has been tested succesfully and failure codes have been determined simultaneously.

To able to use this sytem on the commercial buses, vehicle configuration must be determined and if configuration is exist on the device, it can be applied to the buses. The system can be used just for the montioring the fault codes on the vehice instrument cluster. To achive the monitoring, GPRS/3G modem is not necessary on the device. So, this modem must be plug in type.

On the next steps of this study, it can be used as service management system for the bus manifacturers and it also can be used to upload new software to the vehicle. And also designed system can be used as data collecting system for the future vehicle development.

REFERENCES

AMOR-SEGAN, M. , 2011; A framework for health monitoring of automotive electrical and electronic control systems, Vehicular Netwoking Conference, 2011, Pages 177-184.

ANONYMOUS, URL: http://www.can-cia.org/index.php?id=19

ANONYMOUS, URL: http://www.kvaser.com/en/about-can.html

ANONYMOUS,URL: http://www.sae.org/standardsdev/groundvehicle/j1939a.htm

BETTA, G., CAPRIGLIONE, D. , AND PIETROSONTO, A. , 2005. A methodology to Test Instrument software ; An applicaiton to the Diagnostic in Automotvie Systems, Instrumentation technology Conference, Ottawa 2005.

BITEUS, JONAS, 2011, Distributed Diagnosis Using a Condensed Representation of Diagnoses With Application to an Automotive Vehicle, Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions, Page(s): 1262-1267

BRADATSCH, C., 2011 ; Towards runtime testing in automotive embedded systems, Industrial Embedded Systems (SIES) Symposium, Pages: 55-58.

CHEN, YU-GUO, 2011, Applications of Bayesian Network in Fault Diagnosis of Braking System, Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2011 International Conference, 26-27 Aug. 2011, Page(s): 234 – 237

CHENG, ANYU, 2011, ECU loader design of in-vehicle CAN network based on ISO15765, Information Science and Technology (ICIST), 2011 International Conference, 26-28 March 2011, Page(s): 1215 – 1217

COSTANTINO, NICO, 2011, Design and Test of an HV-CMOS Intelligent Power Switch With Integrated Protections and Self-Diagnostic for Harsh Automotive Applications, Industrial Electronics, IEEE Transactions on, Page(s): 2715 - 2727

CROLLA, D. , 2009. Automotive Engineering Powertrain, Chassis System, and Vehicle Body.

CROSSMAN, J. , GUO, H. , 2003, Automotive Fault Diagnosis Part: I, IEEE Transactions on Vehicular Technology Vol:52, July 2003.

EBERL, MICHAEL. , 2012, considering diagnosis functionality during automatic system-level design of automotive networks, Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE, **Page(s):** 205 - 213

FUJITSU FR60 MB91460B Series Microelectronics Datasheet

GUPTA, ANIKET, 2011, Model based failure detection of Diesel Particulate Filter, American Control Conference (ACC), 2011, June 29 2011-July 1 2011, Page(s): 1567 - 1572

GUPTA, S. , 2011, Built in self test for fault tolerant real time in vehicle networks through automotive diagnostics, Emerging trends in networks and computer communications Conference, 2011, Page 379-382.

HONGWEI LIU, 2011. Design of remote monitoring and fault diagnosis system for electric vehicle, Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference, 12-14 Aug. 2011, Page(s): 1079 – 1082.

JIANSEN, Y. , KONGHUI, G., HAITAO, D. , AND BIN, X. , 2010, The application of SAE J1939 Protocol in Automobile Smart and Integrated Control System, International Conference on Computer, Mechatronics, Control and Electronic Engineering, 2010.

LEE, MU-YOUL, AUTOMOTIVE NETWORK GATEWAY TO CONTROL ELECTRONIC UNITS THROUGH MOST NETWORK, CONSUMER ELECTRONICS (ICCE), 2010 Digest of Technical Papers International Conference on, Page(s): 309-310

LI, J. , AND XU, Y., 2009, Remote Monitoring Systems based on Embedded Database, Third International Conference on Genetic and Evolutionary Computing, 2009.

LIU, H. , WU, J. , AND CHU, L., 2011. Design of Remote Monitoring and Fault diagnosis System for Electric Vehicle, 2011 International Conference on Electronic & Mechanical Engineering and Information Technology, 2011.

LIYUAN, T. , GUIHE, Q. , AND JINDONG, Z. , 2010. Remote diagnosis system of Vehicle based on Telematics, 2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering, 2010.

MAUDERER, ANDREAS, 2012, Efficient digital design for automotive mixed-signal ASICs using simulink, Design and Diagnostics of Electronic Circuits & Systems (DDECS), 2012 IEEE 15th International Symposium, 18-20 April 2012, Page(s): 372 – 377

MURPHY, Y. , CROSSMAN, J. , CHEN, Z. , AND CARDILLO, J, 2003, Automotive Fault Diagnosis Part II: A Distributed Agent Diagnostic System, IEEE Transactions on Vehicular Technology Vol:52, July 2003.

PARET, D. , 2007, Multiplex Networks for Embedded Systems, John wiley and Sons

PARK, S. , 2012; Development of On-board Diagnosis via CAN for a HVI Technology, Parallel and Distributed Processing with Applicaitons, 2012, Pages 839-840.

POHRONSKA, M. 2011, Embedded System Architecture for Real-Time Rule-Based Reasoning, Engineering of Computer Based Systems (ECBS-EERC), 2011 2nd Eastern European Regional Conference, 5-6 Sept. 2011, Page(s): 85 – 91

PRODANOV, W. , 2009. A Controller Area Network Bus Transceiver Behavioral Model for Network Design and Simulation, IEEE Transactions on Industrial Electronics, Vol 56, No:9, September, 2009.

QUANQI, W. , JIAN, W. ,AND CHU, L. 2011, Design of Vehicle Bus data acquisition and fault diagnosis system, Consumer Electronics, Communications and Networks International Conference, 2011.

RONG, Y. , WANG, H. , ZHANG, L. , QU, M. , 2010, The design of temperature and humidity fuzzy controller based on CAN bus, ICIME The 2nd IEEE International conference, 2010.

SANKAVARAM, C. , PATTIPATI, B. , KODALI, A. , PATTIPATI, K. , AZAM, M., KUMAR, S. , AND PECHT, M. , 2009, Model-based data-driven Prognosis of automotive and Electronics Systems. 5th Annual IEEE Conference on Automation Science and Engineering India, 2009.

SERAFINI, MARCO, 2011, Application-Level Diagnostic and Membership Protocols for Generic Time-Triggered Systems, Dependable and Secure Computing, IEEE Transactions on, Page(s): 177 – 193.

SONG YOU, MARK KRAGE AND LACI JALICS, 2005. Overview of Remote Diagnosis and Maintenance for Automotive Systems, 2005 SAE World Congress Detroit, Michigan April 11-14, 2005

TAHAT, ASHRAF, 2012, Android-based universal vehicle diagnostic and tracking system, Consumer Electronics (ISCE), 2012 IEEE 16th International Symposium, 4-6 June 2012, Page(s): 137 – 143

TREVOR MARTIN, THE INSIDER'S GUIDE TO THE PHILIPS 2005, ARM7-Based Microcontrollers, Published by Hitex (UK) Ltd., 21/04/2005

WEI, S. , JUN, L., YING, G. , DAWEI, Q. , CHENGYANG, Y. , 2010, The design of Embedded Bus Monitoring and Faulşt Diagnosis System Based on Protocol SAE J1939, Power and Energy Conference, 2010.

WOLFHARD LAWRENZ, CAN System Engineering From Theory to Practical Applications, Springer

YAN, M., DELI, W. , 2010, The design of a bus Controller based on SOC in Industrial Control System, International conference on computer Design and Applications, 2010.

YAN, X. , ZHANG, Y. , SHANG, C. ,AND YUAN C. , 2010. Study on remote fault diagnosis system using multi monitoring methods on dredger, PHM'10, 2010.

YUN Y. , ZHIHUA, F. , 2010. Design of a link layer controller for high speed serial bus, ICISE, 2010 2nd International Conference 2010.

ZHANG, XIAODONG, 2010, Fault diagnosis of automotive electric power generation and storage systems, Control Applications (CCA), 2010 IEEE International Conference 8-10 Sept. 2010, Page(s): 719 – 724.

BIOGRAPHY

Emrah YİRİK was born in İstanbul in 1984. He completed his elementary education at Republic Primary School in Ermenek in 1995. He went to high school at Ermenek High School. He graduated from Cukurova University Department of Electrical and Electronics Engineering in 2006. He is working as electrical and electronics engineer in a company in Adana. He is married and father of a daughter. His interest areas are automotive electronics, Controller Area Network and embedded software development.

APPENDIX

FMI and DESCRIPTIONS (SAE J1939, 2000)

FMI=0—DATA VALID BUT ABOVE NORMAL OPERATIONAL RANGE - MOST SEVERE LEVEL

The signal communicating information is within a defined acceptable and valid range, but the real world condition is above what would be considered normal as determined by the predefined most severe level limits for that particular measure of the real world condition (Region e of the signal range definition). Broadcast of data values is continued as normal.

FMI=1—DATA VALID BUT BELOW NORMAL OPERATIONAL RANGE - MOST SEVERE LEVEL

The signal communicating information is within a defined acceptable and valid range, but the real world condition is below what would be considered normal as determined by the predefined least severe level limits for that particular measure of the real world condition (Region d of signal range definition). Broadcast of data values is continued as normal.

FMI=2—DATA ERRATIC, INTERMITTENT OR INCORRECT

Erratic or intermittent data includes all measurements that change at a rate that is not considered possible in the real world condition and must be caused by improper operation of the measuring device or its connection to the module. Broadcast of data value is substituted with the “error indicator” value. Incorrect data includes any data not received and any data that is exclusive of the situations covered by FMIs 3, 4, 5 and 6 as follows in A.1.2.4 through A.1.2.7. Data may also be considered incorrect if it is inconsistent with other information collected or known about the system.

FMI=3—VOLTAGE ABOVE NORMAL, OR SHORTED TO HIGH SOURCE

a. A voltage signal, data or otherwise, is above the predefined limits that bound the range (Region g of the signal range definition). Broadcast of data value is substituted with the “error indicator” value.

b. Any signal external to an electronic control module whose voltage remains at a high level when the ECM commands it to low. Broadcast of data value is substituted with the “error indicator” value.

FMI=4—VOLTAGE BELOW NORMAL, OR SHORTED TO LOW SOURCE

a. A voltage signal, data or otherwise, is below the predefined limits that bound the range (Region f of the signal range definition). Broadcast of data value is substituted with the “error indicator” value.

b. Any signal external to an electronic control module whose voltage remains at a low level when the ECM commands it to high. Broadcast of data value is substituted with the “error indicator” value.

FMI=5—CURRENT BELOW NORMAL OR OPEN CIRCUIT

a. A current signal, data or otherwise, is below the predefined limits that bound the range (Region g of the signal range definition). Broadcast of data value is substituted with the “error indicator” value.

b. Any signal external to an electronic control module whose current remains off when the ECM commands it on. Broadcast of data value is substituted with the “error indicator” value.

FMI=6—CURRENT ABOVE NORMAL OR GROUNDED CIRCUIT

a. A current signal, data or otherwise, is above the predefined limits that bound the range (Region g of the signal range definition). Broadcast of data value is substituted with the “error indicator” value.

b. Any signal external to an electronic control module whose current remains on when the ECM commands it off. Broadcast of data value is substituted with the “error indicator” value.

FMI=7—MECHANICAL SYSTEM NOT RESPONDING OR OUT OF ADJUSTMENT

Any fault that is detected as the result of an improper mechanical adjustment or an improper response or action of a mechanical system that, with a reasonable confidence level, is not caused by an electronic or electrical system failure. This type of fault may or may not be directly associated with the value of general broadcast information.

FMI=8—ABNORMAL FREQUENCY OR PULSE WIDTH OR PERIOD

To be considered in cases of FMI 4 and 5. Any frequency or PWM signal that is outside the predefined limits which bound the signal range for frequency or duty cycle (outside Region b or the signal definition). Also if the signal is an ECM output, any signal whose frequency or duty cycle is not consistent with the signal which is emitted. Broadcast of data value is substituted with the “error indicator” value.

FMI=9—ABNORMAL UPDATE RATE

Any failure that is detected when receipt of data via the data link or as input from a smart actuator or smart sensor is not at the update rate expected or required by the ECM (outside Region c of the signal range definition).

Also any error that causes the ECM not to send information at the rate required by the system. This type of fault may or may not be directly associated with the value of general broadcast information.

FMI=10—ABNORMAL RATE OF CHANGE

Any data, exclusive of the abnormalities covered by FMI 2, that is considered valid but whose data is changing at a rate that is outside the predefined limits that bound the rate of change for a properly functioning system (outside Region c of the signal range definition). Broadcast of data values is continued as normal.

FMI=11—ROOT CAUSE NOT KNOWN

It has been detected that a failure has occurred in a particular subsystem but the exact nature of the fault is not known. Broadcast of data value is substituted with the “error indicator” value.

FMI=12—BAD INTELLIGENT DEVICE OR COMPONENT

Internal diagnostic procedures have determined that the failure is one which requires the replacement of the ECU, used here to mean the packaged unit that includes some microprocessor and its associated components and circuits. It can be assumed that the communications subsystem is not the part that has failed, and that the manufacturer has determined that there is no serviceable component smaller than the ECU involved in the failure. Broadcast of data value is substituted with the “error indicator” value if appropriate, as there may or may not be any broadcast data involved. This error is to include all internal controller trouble codes that can not be caused by connections or systems external to the controller.

FMI=13—OUT OF CALIBRATION

A failure that can be identified to be the result of not being properly calibrated. This may be the case for a subsystem which can identify that the calibration attempting to be used by the controller is out of date. Or it may be the case that the mechanical subsystem is determined to be out of calibration. This failure mode does not relate to the signal range definition as do many of the FMIs.

FMI=14—SPECIAL INSTRUCTIONS

"Special Instructions" is the FMI to be used when the on-board system can isolate the failure to a small number of choices but not to a single point of failure. When this FMI is used, there is a clear necessity for the service technician to take some action to complete the specific diagnosis, and the Manufacturer has provided instructions for the completion of that diagnosis. There are two cases where this will be used: 1. for emission-related diagnostics where the particular failure cannot be separated between a sensor out of range and the case where the actual value is at the

edge of a diagnostic region, and 2. for the older SPN 611 to 615 where the problem is in determining which of two or more circuits (which may interact) is the one that needs repair. SPNs 611 through 615 are defined as “System Diagnostic Codes” and are used to identify failures that cannot be tied to a specific field replaceable component. Specific subsystem fault isolation is the goal of any diagnostic system, but for various reasons this cannot always be accomplished. These SPNs allow the manufacturer some flexibility to communicate non-“specific component” diagnostic information. Since SPNs 611-615 use the standard SPN/FMI format it allows the use of standard diagnostic tools, electronic dashboards, satellite systems and other advanced devices that scan Parameter Groups containing the SPN/FMI formats. Because manufacturer defined codes are not desirable in terms of standardization, the use of these codes should only occur when diagnostic information cannot be communicated as a specific component and failure mode.
Possible reasons for using a System Diagnostic Code include:

1. Cost of specific component fault isolation is not justified, or
2. New concepts in Total Vehicle Diagnostics are being developed, or
3. New diagnostic strategies that are not component specific are being developed.

Due to the fact that SPNs 611-615 are manufacturer defined and are not component specific, FMIs 0-13 and 15-31 have little meaning. Therefore, FMI 14, “Special Instructions”, is usually used. The goal is to refer the service personnel to the manufacturer's troubleshooting manual for more information on the particular diagnostic code. This failure mode does not relate to the signal range definition as do many of the FMIs. This type of fault may or may not be directly associated with the value of general broadcast information.

FMI=15—DATA VALID BUT ABOVE NORMAL OPERATING RANGE - LEAST SEVERE LEVEL

The signal communicating information is within a defined acceptable and valid range, but the real world condition is above what would be considered normal as determined by the predefined least severe level limits for that particular measure of the real world condition (Region i of signal range definition). Broadcast of data values is continued as normal.

FMI=16—DATA VALID BUT ABOVE NORMAL OPERATING RANGE - MODERATELY SEVERE LEVEL

The signal communicating information is within a defined acceptable and valid range, but the real world condition is above what would be considered normal as determined by the predefined moderately severe level limits for that particular measure of the real world condition (Region k of signal range definition). Broadcast of data values is continued as normal.

FMI=17—DATA VALID BUT BELOW NORMAL OPERATING RANGE - LEAST SEVERE LEVEL

The signal communicating information is within a defined acceptable and valid range, but the real world condition is below what would be considered normal as determined by the predefined least severe level limits for that particular measure of the real world condition (Region h of signal range definition). Broadcast of data values is continued as normal.

FMI=18—DATA VALID BUT BELOW NORMAL OPERATING RANGE - MODERATELY SEVERE LEVEL

The signal communicating information is within a defined acceptable and valid range, but the real world condition is below what would be considered normal as determined by the predefined moderately severe level limits for that particular measure of the real world condition (Region j of signal range definition). Broadcast of data values is continued as normal.

FMI=19—RECEIVED NETWORK DATA IN ERROR

Any failure that is detected when the data received via the network is found substituted with the “error indicator” value (i.e. FE16, see J1939-71). This type of failure is associated with received network data. The component used to measure the real world signal is wired directly to the module sourcing the data to the network and not to the module receiving the data via the network. This FMI is applicable to Regions f and g of the signal range definition. This type of fault may or may not be directly associated with the value of general broadcast information.

FMI=20-30—RESERVED FOR SAE ASSIGNMENT

FMI=31—CONDITION EXISTS

Used to indicate that the condition that is identified by the SPN, exists when no more applicable FMI exists or in cases when the reported SPN name spells out the component and a non-standard failure mode. This type of fault may or may not be directly associated with the value of general broadcast information. This FMI will mean "not available" when the associated SPN is also "not available" as when the remainder of a packet is filled with binary ones after all data has been transmitted.