

# Universal Tester for Electronic Control Units in Automotives

Jayashree Hegde,  
Aeronautical Development Establishment,  
Bangalore, India

Rajeshwari Hegde  
BMS College of Engineering,  
Bangalore, India

## ABSTRACT

Electronics plays a major role in Modern Automobiles. Several Electronic Control Units (ECU) are employed to improve the performance of the Vehicle resulting in Lower Emissions, Reduced Fuel consumption, Increased safety, Improved drivability and last but not least better driving comfort. In general an ECU is a micro-controller based embedded system with the hardware and software suitably designed for the application. ECU interfaces with various sensors and actuators for real-time data acquisition and control. Electronic Control Unit (ECU) has ability to carry out self-diagnostics, according to a pre-defined logic and store the result in a non-volatile (EEPROM) fault memory. ECU software provides the possibility to perform diagnostics and read the result of self-diagnostics at the service station without dismantling the electrical and electronic systems. This is done by using a "Tester" connected to a serial line provided by the ECU. We have developed a Tester which reads the information from the ECU and stores it in the error memory of the ECU. In this paper, we explain a PC based simulator of a real life "Tester" which is refereed as Universal Tester Simulator (UTS) in this article. This is to ensure ECU development is not hampered due to non availability of an actual tester.

## General Terms

Automotive Electronics

**Keywords:** ECU, CAN, K-Line, tester.

## 1. INTRODUCTION

Increased Use of Electronic Control Units (ECUs) boosts Demand for ECU Testers. Electronic content in present-day automobiles is constantly increasing, creating the need for ECUs that process the information obtained from different sensors in a vehicle and pass it on to actuators. This, in turn, is expected to lead to a corresponding rise in demand for ECU test equipment, which is becoming increasingly important due to the highly critical nature of various vehicle functions such as engine control, anti-lock brake system (ABS), and air bag deployment[1].

Following are the important building blocks of an Electronic Control Unit (ECU):

- Micro-controller with ROM, RAM.
- Software embedded in the EPROM.
- ADCs and Digital ports (interfaces for Analog and Digital inputs and outputs).
- Interfaces for communication with other devices (CAN, K-Line etc).
- Special control ICs.

- EEPROM (for end-of-line settings and storage of faults detected by self-diagnostics).
- Discrete components (like Resistors and Capacitors etc.).

A mechanism to allow reading of Information from the ECU in order to localize a problem of a complex system is of high importance in the workshop. This communication goes on a special line called as K-Line [2]. Using this, the service personnel can read out the error memory of the ECU (which has the encoded information of all the failures registered in the system during operation). International Standards have been established to define common requirements for diagnostic systems. The standard is based on the Open Systems Interconnection (O.S.I.) Basic Reference Model in accordance with ISO standard 7498.

Most of the ECUs support a standard serial communication protocol called as Keyword Protocol 2000(KWP2000). ISO 14229 standards specify the common requirements for implementation of diagnostic services. KWP200 protocol has specifications for the physical layer, data link layer and application layer as follows.

ISO 14230-1 Keyword Protocol 2000:Physical Layer Specification

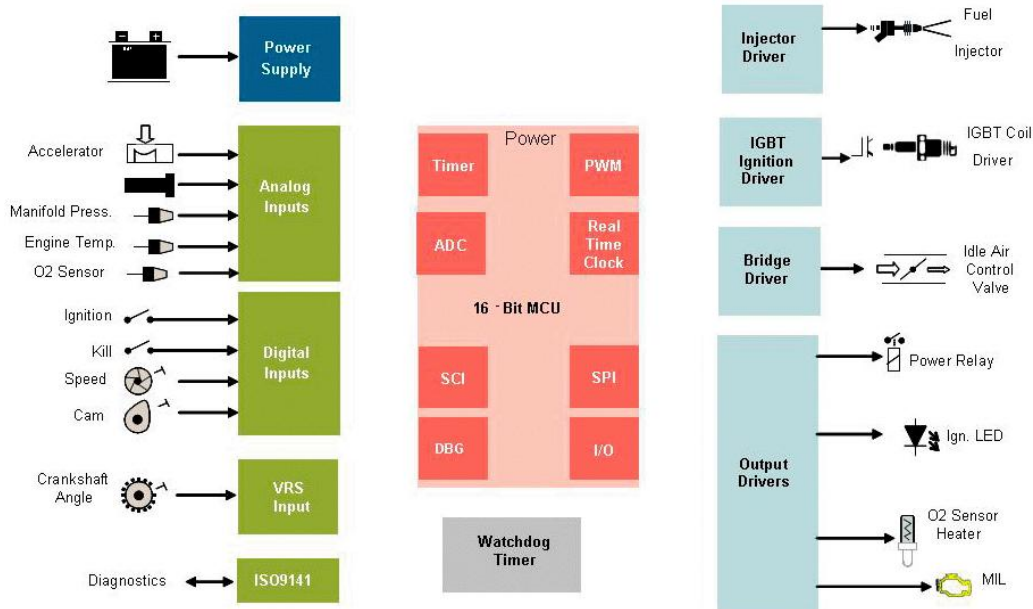
ISO 14230-2 Keyword Protocol 2000:Data Link Layer Specification

ISO 14230-3 Keyword Protocol 2000:Application Layer Specification [3].

In spite of the above mentioned standardization efforts, due to historical reasons considerable difference exists in the actual protocol used by the various Vehicle manufacturers. Difference is mainly found in upper layers (Data Link and Application Layers) though all of them follow the ISO specifications for the physical layer. Vehicle manufacturers develop their own "Testers" (Normally a dedicated hand held hardware device) to support the protocol variant used by them. Whenever ECU software is developed to suit the requirements of individual Vehicle manufacturer, the same is to be tested using a suitable tool. The paper is organized as follows. Section 2 describes ECU and Interfaces. Section 3 looks at the need for a tester. Section 4 explains the design methodology. Section 5 gives the implementation details. The paper is concluded in section 6.

## 2. ECU AND INTERFACES

The "Universal Tester Simulator (UTS)" is to be used for testing of the ECU (Electronic Control Unit) Software. In this section a brief introduction to ECU and related components is provided. Figure1 gives an overview of the Major components of a typical ECU used for Fuel Injection Control. Diagram shows major functional blocks of the ECU along with External Interfaces.

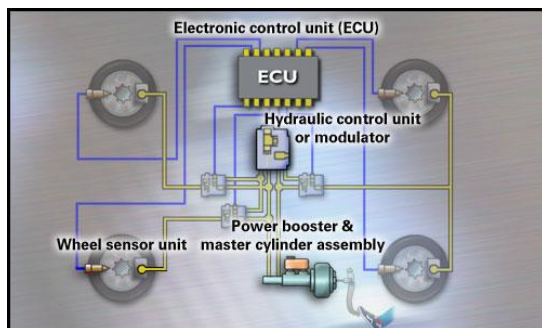


**Fig1: Schematic Diagram of Engine Control Unit**

As can be seen in the “Schematic Diagram” above, the ECU communicates with the external world through a Diagnostic interface. One of the standard features of such interface is the support for serial diagnostic line providing interface to the diagnostic “Tester”. The serial line used for this purpose is known as ISO K-Line and is specified by ISO 14230-1 standards [4].

### 2.1 ECU in a Vehicle

Fig. 2 shows a closed ECU typically used in a Vehicle and also external components, which interface with ECU. Here the ECU is completely sealed in water/air tight container and any diagnostics of the system is possible only through the K-Line interface.



**Fig2: Electronic Control Unit**

### 3. NEED FOR A TESTER

In a vehicle service station, self-diagnostic and reporting capability give the repair technician access to state of health information for various vehicle sub-systems. The amount of

diagnostic information available via diagnostic Tester has varied widely since the introduction in the early 1980s of on-board vehicle computers or ECUs. Early instances of self diagnostic would simply illuminate a malfunction indicator

light, or MIL, if a problem was detected but would not provide any information as to the nature of the problem. Modern self diagnostic implementations use a standardized fast digital communications port to provide real-time data in addition to a standardized series of diagnostic trouble codes, or DTCs, which allow one to rapidly identify and remedy malfunctions within the vehicle.

#### A. Main functionalities of a diagnostic Tester

- Reading stored ECU fault codes, giving a fast “health check” on the car’s electronic systems.
- Actuator tests allow the operator to energise various actuators in the car as required to assist with fault finding.
- Clearing stored fault codes after the fault has been rectified.
- Reset the idle speed of some models.
- Activates diagnosis of the sub components of the ECUs.

End customers (Vehicle Manufacturers) normally develop a dedicated hardware tool for K-Line interface in the form of a hand-held Tester [5]. In this paper, we explain the design of a PC based system to simulate this hardware device as closely as possible for Laboratory tests of ECU software. Mechanical Engine tester is used at the service station for the diagnostics of faults in vehicle.

### 4. DESIGN METHODOLOGY

Design Methodology followed is similar to standard structured design. The total system is sub-divided into three main components as listed below

- PC based software with a well-designed GUI (Graphical User Interface) for configuration of the communication and processing the results of the communication.
- Embedded software on the configurable tester for co-ordination of communication with PC including detection and correction of communication errors and relaying back the results to the PC.
- Embedded software on the configurable tester for execution of final commands and communication with

the target ECU. This component also stores the result of communication in a ring buffer.

#### 4.1 Key Design Objectives

- To design a system that can simulate a “Tester” (Used by Vehicle manufacturer in the Service stations) in all respects.
- The system should support KWP2000 and any other serial (single line) communication protocol on K-Line.
- The system should consist of PC-based software interfacing with a configurable “Tester”.
- The PC communicates with the “Tester” for configuring the communication protocol and the communication parameters.
- The “Tester” communicates with the ECU and for all practical purposes behaves similar to the Tester used by end customer (Vehicle manufacturer).
- To summarize, the aim of the design is to develop a system where any customer defined protocol can be supported without changing the embedded software of the configurable tester. Entire configuration of the protocol should be possible through set of configuration files and Menu options on the PC based software.

#### 4.2 Design Considerations

The following are the key considerations addressed in the design.

##### 4.2.1 Functionality

Tool (PC Based front End + Configurable Universal Tester H/W) should be able to simulate customer tester in all the following aspects

- Initialisation sequence. (An ECU needs to be initialised in-order to start a “Diagnostic Session” with it. The

method used for initialisation differs significantly in different protocols. The System should be able to simulate any such Initialisation sequence. Fast / 5 Baud Initialisation for KWP2000 OR any customer defined initialization sequence).

- Timings for both transmission and reception (both Inter-byte and inter-frame timings).
- Frame structure with respect to header bytes and Checksum etc.,
- Support all possible Baud rate within an allowed range.

##### 4.2.2 Reliability

Tools should be able to simulate the customer Tester as close as possible. Test of the ECU software with this tool should ensure adherence to all the protocol parameters such as baud rate, Inter byte and Inter Block timings.

##### 4.2.3 Expandability

- Command set should be easily expandable.
- With minor modification in PC based software, it should be possible to provide Tailored Menu interfaces for simulating Testers from different vehicle manufacturers.

##### 4.2.4 Ease of Use

It should be possible to configure the tester so that it can satisfy varied protocols through simple menu options in the front-end and/or a set of commands in configuration file(s).

#### 4.3 Design Overview

Working of this tool can be schematically represented in fig3.

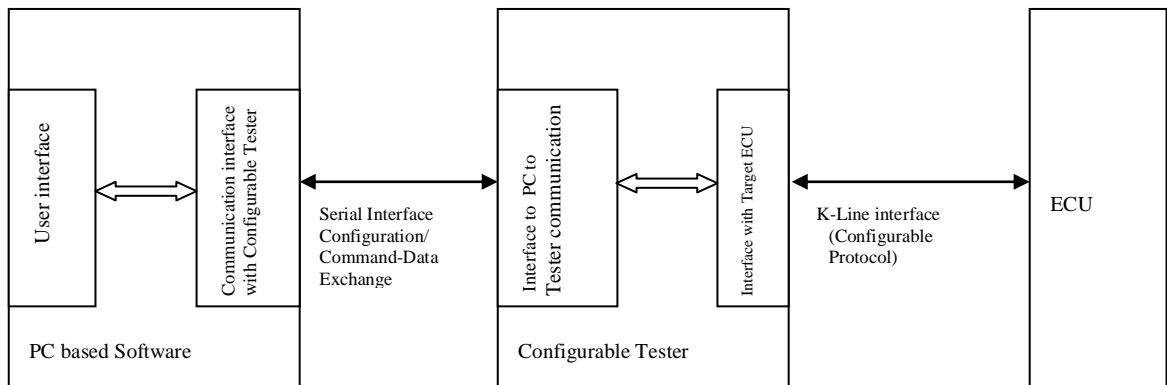


Fig3:Block schematic of the tester

Software is designed both for the “PC based front-end” as well as for “Universal-Tester”. Scope of the work includes development of PC based software and design and simulation of Universal Tester software. Front-end has to be developed to work on Windows based system and provides user interfaces through Menu-options and a set of dialog boxes. The software is designed to be able to read and take appropriate actions based on the commands stored in configuration file(s). Selection of configuration file(s) should be possible through Menu-Options.

#### 4.3.1 The basic features on the PC side

- PC-based user-friendly front-end for basic operation of the system
- Specifying the configuration of the communication protocol through user-friendly front-end and transferring the same to the “Tester”.
- Sending commands to the “Tester” to perform various services (including configuration).
- Receiving data from the “Tester” and displaying/storing the results of communication.

#### 4.3.2 The software features embedded on the tester

- Communication with the PC to receive the configuration of the communication protocol and store it in memory.
- Execution of commands sent by the PC.
- Communication with the ECU over the “K” line. Configuration of the communication is decided by commands received from PC based front-end.
- Transfer of data received from ECU to the PC on request.

The “Tester” is a microcontroller based system with embedded software to fulfill features described in section 1. Communication of the tester with the ECU is completely configurable and supports various timing parameters including zero inter-byte time. Scope of the work includes design of the complete system, development of PC based software also defining the communication between the PC and the “Tester”. The same is well defined and implemented as an API to allow future expansion of commands. The command set defined is flexible enough to handle different types of initialization, complete range of timing parameters and various header formats.

## 5. IMPLEMENTATION

PC based front end has been implemented using Microsoft VC++ as a Win 32 Application.

This portion of software performs following operations.

- Provides user interface through a set of Menu Options.
- Handles communication interface to Configurable Universal Tester.

### 5.1 Implementation of Tester software

A simulation software was developed to test the proper working of “PC based front-end”. Simulation was done using a second PC with the serial port connection between the two PCs. (with an RS232 compatible interface). After verifying the result of simulation, Communication interface was developed on the Tester. The communication between the PC based front end and Tester was verified for correctness of timings and data.

## 6. CONCLUSION

In this paper, Universal tester for testing the ECU has been presented. ECU stores several information like Sensor Failure, and other abnormalities in a non-volatile memory. In Service stations it is possible to read this information using the Diagnostic Tester. The protocol used for the communication between the Diagnostic Tester and the ECU differs significantly across the Vehicle manufacturers.

Whenever an ECU software is modified to support the customer specific protocol over K-Line interface the same can be tested using “UTS”. This avoids the delay in development due to non-availability of the actual Tester. The tool will also help to Test the ECU software more completely with respect to every aspect of the specification like Various initialization methods and timing parameters.

## 7. REFERENCES

- [1] Jochen Schaffnit, Stefan Sinsel, Rolf Isermann, “Hardware-in-the-loop simulation for the investigation of truck diesel injection systems”, Proceedings of the American Control Conference, Philadelphia, 1998, 498-502.
- [2] Global Automotive ECU Functional Test Equipment Market Research, Trends, Analysis.htm.
- [3] International Organisation for Standards, available at [www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=55591](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=55591)
- [4] freescale, available at <http://www.freescale.com/files/BRAUTOSOL.pdf>
- [5] CDX Online eTextbook, available at <http://www.cdxetextbook.com/toolsEquip/toolsEquip.html>
- [4] ABS Troubleshooting for Trucks, Trailers, and Buses, available at <http://www.abstroubleshooting.com/abs-diagnostic-tools.html>
- [6] Minsheng Wang. National Instruments LabVIEW Basic I. Beijing: Publishing House of Electronics Industry, 2002.
- [7] David J. Kruglinski, Scot Wingo, George Shepherd, Programming Microsoft Visual C++ 6.0. Beijing:Beijing Hope Electronic Press, 1999.
- [9] CAN in Automation, available at [www.can-cia.org](http://www.can-cia.org).