In-Vehicle Automotive Network Gateway
Electronic Control Unit for Low Price Vehicle

T. S. Shamin Dudu\textsuperscript{1}, S. G. Shivaprasad Yadav\textsuperscript{2}, M. Arun Kumar\textsuperscript{2}, Nisha C. Rani\textsuperscript{3}
\textsuperscript{1} - M. Sc. [Engg.] Student, \textsuperscript{2} - Senior Lecturer, Real Time Embedded Systems Centre, M.S. Ramaiah School of Advanced Studies, Bangalore 560 054
\textsuperscript{3} - Senior Lecturer, Department of Electrical and Electronics Engineering, The Oxford College of Engineering, Bangalore 560 068

Abstract
The rising numbers of sensors, actuators and electronic controls increased the complexity of automotive networks. Moreover, multiple network systems have evolved to meet the different requirements coming from automotive applications. A gateway Electronic Control Unit (ECU) is a central network interconnecting system to link various field buses in a vehicle. In this paper, a gateway ECU is proposed to interconnect Controller Area Network (CAN) and Local Interconnect Network (LIN) field buses for Low Price Vehicles (LPVs).

A gateway ECU is necessary for addressing the communication and network challenges in today’s vehicles. A study is conducted to know about existing commercial gateway ECU, and derive the specification for a gateway ECU suitable for LPVs. The proposed gateway ECU is designed based on this specification and implemented using PIC microcontroller and line transceivers for interconnecting LIN and CAN buses. The designed gateway ECU has been successfully validated using two other nodes – one node with LIN and another with CAN networks.

The proposed gateway ECU has optimal functionality and is cost effective solution for LPV segments. The trends in automotive networks and electronics show that the LPV needs to have only CAN and LIN networks in next one to two decades. In short, the proposed gateway ECU has optimal balance between functionality and cost, and is best suitable for LPVs.

Key Words: Electronic Control Unit, Gateway, In Vehicle Network, Controller Area Network, Local Interconnect Network, Low Price Vehicle

Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTOSAR</td>
<td>Automotive Open System Architecture</td>
</tr>
<tr>
<td>AVC-LAN</td>
<td>Audio Visual Communication - Local Area Network</td>
</tr>
<tr>
<td>BEAN</td>
<td>Body Electronics Area Network</td>
</tr>
<tr>
<td>CAN</td>
<td>Controller Area Network</td>
</tr>
<tr>
<td>ECU</td>
<td>Electronic Control Unit</td>
</tr>
<tr>
<td>FPGA</td>
<td>Field Programmable Gate Array</td>
</tr>
<tr>
<td>LIN</td>
<td>Local Interconnect Network</td>
</tr>
<tr>
<td>LPV</td>
<td>Low Price Vehicle</td>
</tr>
<tr>
<td>Mbps</td>
<td>Mega bits per second</td>
</tr>
<tr>
<td>MOST</td>
<td>Media-Oriented System Transport</td>
</tr>
<tr>
<td>OSI</td>
<td>Open Systems Interconnection</td>
</tr>
<tr>
<td>PDU</td>
<td>Protocol Data Unit</td>
</tr>
<tr>
<td>PSoC</td>
<td>Programmable System on-Chip</td>
</tr>
<tr>
<td>RAM</td>
<td>Random Access Memory</td>
</tr>
<tr>
<td>TP</td>
<td>Transport Protocol</td>
</tr>
<tr>
<td>TTP</td>
<td>Time Triggered Protocol</td>
</tr>
<tr>
<td>kbps</td>
<td>kilo bits per second</td>
</tr>
<tr>
<td>uC</td>
<td>Microcontroller</td>
</tr>
</tbody>
</table>

1. INTRODUCTION
Since the 1970s, an exponential increase in the number of electronic systems has been observed that have gradually replaced those that are purely mechanical or hydraulic as discussed in [1]. In early days, each new function was implemented as a stand-alone ECU, which is a sub-system composed of a microcontroller and a set of sensors and actuators. The evolution of automotive electronics since 1960 was tremendous starting from simple ignition control to X-by-wire technology by 2010 [2]. As the electronics increased, the need for functions to be distributed over several ECUs and the need for information exchanges among them have been evolved.

The different performance requirements throughout a vehicle, as well as competition among the companies in automotive industry, have led to the design of a large number of communication networks. A gateway is required to manage all these in vehicle networks and messages effectively. Network gateway is a device or a piece of software in a computer that forwards and routes data packets along networks. A possible gateway concept in a LPV is depicted in Fig. 1. This paper focuses to design, implement and validate gateway ECU to interconnect LIN and CAN networks of LPV.

Fig. 1 Possible Gateway concept in LPV

2. TRENDS IN GATEWAY ECU
Field-bus is serial bus allowing message exchange between nodes connected to bus. Today most ECUs exchange information with each other using field-buses.
The major requirements in automotive communications are fault tolerance, determinism, bandwidth, flexibility and security [3] depending on the applications. For example, determinism and bandwidth are major requirements for powertrain application, whereas bandwidth and flexibility are major requirements for multimedia applications.

Today several field bus technologies are used by different automotive system vendors. The most common field bus technologies are LIN, CAN and MOST. The upcoming technologies are FlexRay™, TTP, ZigBee etc. LIN is single wire communication which uses master-slave concept, can transfer data up to 8 bytes in a frame at maximum data rate of 19.6 kbps. But CAN uses multi-master concept with maximum data rate of 1 Mbps, which has a differential bus in physical medium. FlexRay™ has advantage of data rate of up to 10Mbps with maximum of 255 bytes per frame [4].

Due to higher complexity, the entire electronics in vehicle are classified as logical domains such as powertrain domain, chassis domain, telematics domain etc. [5]. The range of domains in today vehicles is between 2 and 4. In a low or mid class vehicle, typically two domains - body and powertrain - can be found. Up to four domains can be found in luxury cars. Besides the body and powertrain, separate domains for telematics and chassis are also realized. Gateways are required to connect different vehicle domains. The central gateway, multi-gateway or backbone topologies are three different concepts for the organization of domains. In small or mid sized cars, two vehicle domains protocols (body, chassis) and central gateway, multi-gateway architecture are possible. For current luxury cars, three domains (body, powertrain) with central gateway, multi-gateway or backbone topologies are possible. For future luxury cars, four or more vehicle domains with backbone architecture could be required.

The Gateway converts a data frame from one protocol form to another protocol form. In order to meet the challenging demand, the gateway functionality has been distributed between hardware and software. Dedicated gateways must be developed with the objective of reducing the demands on CPU. This concept consists of a standard CPU core, an internal RAM or Flash and several communication controllers.

One such FlexRay-CAN gateway embedded system can be designed with MPC5554 uC, MFR4200 FlexRay controller, CAN transceiver and FlexRay transceiver [6].

Standardization is necessary in order to ensure the adaptability to various platforms or systems, flexibility to integrate the module in any existing system or addition of new features in the module and reusability of the module. According to AUTOSAR [7], the entire gateway functionality can be divided into three units – PDU gateway, TP gateway and Signal gateway. PDU gateway routes entire data packets from one network to another and to route transport protocol data. A TP gateway is required to transfer the extensive diagnostic data of an ECU from one network to another via the Transport Protocol. If only individual signals are needed on the other network, the signal gateway disassembles the received PDU into individual signals, and then assembles them into one or more PDUs.

The gateway shall provide an efficient exchange of data and on the other hand, safety-critical communication (e.g., steer-by-wire, brake-by-wire) must not be influenced by non-safety-critical traffic [8]. Latency and performance are important parameters of a gateway. In contrast to the commercially available gateways, the automotive gateway platform which is currently developed by the UAS Technikum, Wien [8] can be configured and expanded to a high degree concerning both hardware and software by using a PSoC architecture. The proposed P160 gateway expansion board includes Xilinx Spartan-3E FPGA (XC3S1600E) and related circuitry. One of the possible LIN-CAN-FlexRay gateway is discussed in detail [9], which uses MPC5516 uC and another one is discussed in detail [10], which uses MC9S12DP256 uC.

Gateway ECU has been considered as a necessary component in a high end vehicle since few years. Toyota has introduced a gateway ECU in their previous models since 2004 to interconnect communication systems such as CAN, BEAN (Body Electronics Area Network) and AVC - LAN [11]. Another example of gateway is Central Electronic Module [3] in Volvo XC90 models, which interconnects low speed CAN (125 kbps), high speed CAN (500 kbps or 1 Mbps) and LIN. In order to understand the complexity and types of messages to be exchanged in a vehicle, CAN messages in Mitsubishi motors’ GRANDIS vehicle is considered [12]. Here, around 40 different messages are exchanged between 8 ECUs at various intervals required by the system.

LIN, CAN, and FlexRay protocols have different message transmission speeds, communication paradigms and message frames. As a result, an algorithm to minimize the differences between the protocols is required. The differences in message transmission speed cannot be changed because the speeds are set by the protocols. An algorithm has been proposed for the gateway embedded system [13] which focuses on interpreting the message frames of each protocol and minimizing the differences in communication paradigms.

The trends in automotive communication are shown in Fig. 2. From the requirements for LPVs and focus on cost effective solutions, it can be concluded that at least for next one to two decades, only LIN and CAN networks needed to meet requirements of LPVs. Hence, the proposed gateway is designed with CAN and LIN networks only. The trends in LIN protocol have been discussed in detail in [14], for sensor networking application in vehicle.

![Fig. 2 Trends in automotive communication](image-url)
3. DESIGN OF GATEWAY ECU FOR LPV

3.1 Specification

The specification of a typical gateway ECU for LPVs is listed down in Table 1.

Table 1 Specification of gateway ECU for LPVs

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values in LPVs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Typical Number of ECUs</td>
<td>~10…20</td>
</tr>
<tr>
<td>Types of Protocols</td>
<td>CAN, LIN</td>
</tr>
<tr>
<td>Typical Number of signals</td>
<td>~200…400</td>
</tr>
<tr>
<td>Required maximum bus speed</td>
<td>CAN : 500kbaud, LIN : 9.6kbaud</td>
</tr>
<tr>
<td>Need of Operating system</td>
<td>Simple scheduler</td>
</tr>
<tr>
<td>Number of CAN, LIN buses</td>
<td>Single</td>
</tr>
<tr>
<td>Interface to Diagnostic tester</td>
<td>Possible</td>
</tr>
<tr>
<td>Typical Flash memory size</td>
<td>32..64kB</td>
</tr>
<tr>
<td>Typical CPU clock speed</td>
<td>8…20MHz</td>
</tr>
<tr>
<td>Message transmission time slots</td>
<td>&lt;=3 (10ms, 100ms, 1000ms)</td>
</tr>
</tbody>
</table>

3.2 Major Requirements

Functional requirements: The following are the major functional requirements of the proposed gateway ECU. The gateway ECU shall
- Use battery voltage input to derive its power
- Have its own power supply unit to derive 5V supply for uC and other digital components and 12 V supply for powering line drivers
- Have a optimal scheduler to schedule CAN transmission and LIN communication messages
- Support one CAN 2.0 and one LIN 2.0 bus interface
- Have following CAN protocol requirements:
  - 11 - bit identifier length
  - 3 transmission time slots for transmission (10 ms, 100 ms, 1000 ms)
  - Baud rate up to 500 kbps
- Have following LIN protocol requirements:
  - 6-bit identifier
  - 2 transmission time slots (100 ms, 1000 ms)
  - Baud rate up to 9.6 kbps
- Keep the overview of all CAN messages.
- Be the master for LIN communication.
- Be able to configure CAN message parameters-identifier, message length and data pointer.
- Be able to configure LIN message parameters - identifier, communication mode – transmit/ receive and message data.
- Compute the checksum within the respective protocol handlers.
- Compute parity for identifiers of LIN within the respective protocol handler.
- Be able to configure the messages to be transferred from CAN to LIN or LIN to CAN easily

Non-functional requirements: Non-functional requirements are the requirements which are related to the behavior of a system in its operational environment. They are as listed down below:
- Environment: System shall be able to operate in the temperature range of - 40 to 125 deg
- Reliability: The system shall be consistent and repeatable for its entire functionality
- Power: Power consumption of the system shall be optimal and support power saving modes like sleep, idle to save power consumption
- Flexibility: The system shall be flexible to add new functionalities. The system shall have at least 16kbytes of flash memory.
- Maintainability: The system shall be able to service and upgrade very easily.
- Safety: The system shall not behave in an unsafe mode during any operations.
- Cost: Cost of entire components used shall not exceed $10.

3.3 Design and Implementation

The high level design of proposed gateway is shown in Fig. 3. Major components used in the design are microcontroller, CAN and LIN transceivers. The microcontroller chosen for gateway implementation is PIC 18F4580 [19] from Microchip. The CAN transceiver, MCP2551 [20] is from Microchip and LIN transceiver, TPIC1021 [18] is from Texas Instruments.

3.4 Justification of the Hardware Design

The entire design and components used in the proposed gateway ECU can meet the cost and temperature requirements.
- The hardware components used in the design meets automotive standards and requirements
- PIC 18F4580 uC has support for both CAN (using Enhanced CAN module) and LIN (using Enhanced USART module) protocols
- PIC 18F4580 uC conforms to CAN 2.0 and supports LIN2.0 specifications
- PIC 18F4580 uC meets the memory requirements of the system
- TPIC1021 conforms to LIN physical layer specification revision 2.0
- MCP2551 meets CAN physical layer specification 2.0
• The hardware components used can be upgradeable for incorporating more functionality without higher cost impact.

3.5 Software Design

The overall software architecture of the proposed gateway ECU is shown in Fig. 4. The entire system is divided into small functional components in order to have higher maintainability and scalability. The subcomponents are:

• GatMain.c: The main module of gateway to control the entire functionality of the system
• Can.c: The CAN transmission and reception routines defined in this module
• LINMast.c: Master transmission and reception functions for LIN are defined in this module
• ProtConv.c: LIN to CAN conversion and CAN to LIN conversion routines are handled in this module
• Disp.c: The LCD display control and routine to refresh the display are defined in this module
• OSCPU.c: The scheduler for gateway is implemented in this module to generate 10ms time base for the scheduler

The software design of GatMain module for the proposed gateway ECU is shown in Fig. 5 and LINMast module is shown in Fig. 6.

---

Fig. 4 Software architecture of gateway ECU

Fig. 5 Software design of GatMain module

Fig. 6 Software design of LINMast module
The software design of CAN module is shown in Fig. 7.

![Fig. 7 Software design of can module](Image)

The process of configuring gateway functionality is depicted in Fig. 8. The LIN to CAN and CAN to LIN protocol conversion algorithm (ProtConv module) which is being performed at runtime is shown in Fig. 9.

![Fig. 8 Gateway functional work flow](Image)

### 3.6 Database Design for the Messages

- CAN message buffer is defined below as an array of the data structure.

typedef struct {
  unsigned int CANID; // Identifier
  unsigned char CAN_MSGLEN; // Length
  unsigned char * CAN_MsgPtr; // Data Ptr
}CANMsgHdr;

- LIN message buffer is defined as array of the below data structure.

typedef struct {
  unsigned char LIN_FrameStat; // Frame status
  unsigned char LIN_ComMode; // Com mode:
  (RECEIVE or TRANSMIT)
  unsigned char LIN_Identifier; // Identifier
  unsigned char LIN_MsgPtr[9]; // data
}LINMsgHdr;

![Fig. 9 Software design of Prot-Conv module](Image)

### 3.7 Hardware Design

The hardware implementation of the gateway ECU for LPV is shown in Fig. 11.

### 4. VALIDATION AND RESULTS

The demo concept for validating the gateway ECU is shown in Fig. 10.

![Fig. 10 Demo concept for gateway validation](Image)
The photograph of the prototyped gateway ECU with demo setup is shown in Fig. 12.

- Checking CAN reception by the gateway ECU node. Test conditions are: 1) Connect the CAN bus of CAN node to gateway node 2) Send one CAN message from CAN node to gateway
- Checking CAN transmission by the gateway ECU node. Test conditions are: 1) Connect the CAN bus of CAN node to gateway node 2) Send one CAN message from gateway to CAN node
- Checking the CAN transmission at various rate 10 ms, 100 ms and 1000 ms and validate the received message. The test conditions are: 1) Connect the CAN bus of CAN node to gateway node 2) Send one CAN message from CAN node to gateway 3) Repeat the message transmission in time slots 10 ms, 100 ms and 1000 ms
- Check the CAN communication at higher bus load. The test conditions are: 1) Connect the CAN bus of CAN node to gateway node 2) Connect CANalyzer to CAN bus 3) Send one CAN message from CANalyzer continuously in every 10 ms
- Checking LIN transmission by master gateway node. The test conditions are: 1) Connect the LIN bus of LIN node to gateway node 2) Send one LIN message from gateway to LIN node
- Checking LIN reception by gateway master node. The test conditions are: 1) Connect the LIN bus of LIN node to gateway node 2) Send one LIN message from LIN node to gateway
- Checking the LIN transmission at various rate 100 ms and 1000 ms and validate the received message 1) Connect the LIN bus of LIN node to gateway node 2) Send one LIN message from LIN node to gateway 3) Repeat the message transmission in time slots 100 ms and 1000 ms
Flash Memory consumption

- Used: 22.74%
- Free: 77.26%

Memory usage - Module wise

- OS CPU: 1%
- PROT Conv: 15%
- CAN: 20%
- LIN Mast: 22%
- Library: 6%
- GatMain: 18%
- disp: 14%

RAM Memory usage

- Used: 34%
- Free: 66%

RAM Usage - Module wise

- OS CPU: 2%
- LIN Mast: 8%
- GatMain: 18%
- disp: 11%
- can: 14%
- stack: 48%
- Library: 7%

Cost is the major parameter considered by the LPV manufacturers for any ECU to be considered for the vehicle. Hence, there shall be an optimal balance of the functionality and cost for the gateway ECU. The cost estimation and justification of the proposed gateway ECU is summarized in Table 2. From this, it is evident that the proposed gateway is at least 60% cheaper than the current commercially available gateway ECU.

Table 2 Cost justification of the proposed gateway ECU for LPVs

<table>
<thead>
<tr>
<th>Components</th>
<th>High end (approximate) $14.3</th>
<th>Proposed (approximate) $5.72</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microcontroller AL</td>
<td>PIC18F4580 $4.38</td>
<td></td>
</tr>
<tr>
<td>LIN Transceiver</td>
<td>TPIC1021 $0.55</td>
<td></td>
</tr>
<tr>
<td>CAN Transceiver</td>
<td>MCP2551 $0.79</td>
<td></td>
</tr>
<tr>
<td>Total (approximate)</td>
<td>$14.3</td>
<td>$5.72</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Tools</th>
<th>High end (approximate) $2493.00</th>
<th>Proposed (approximate) $938.98</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compiler suite CodeWarrior Professional (IDE+Compiler) from $1995</td>
<td>MPLAB IDE: Free</td>
<td></td>
</tr>
<tr>
<td>Programmer M68CYCLONEP ROE ~$498</td>
<td>PICSTART PLUS Programme: $ 223.99</td>
<td></td>
</tr>
<tr>
<td>Debugger MPLAB ICD 3 : $219.99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total (approximate)</td>
<td>$2493.00</td>
<td>$938.98</td>
</tr>
</tbody>
</table>


Development effort ~12-18 man months | ~6-12 man months

The following are some of the benefits of a gateway ECU in an automotive networking system.

- Easy configuration and good overview of messages (Messages are centrally coordinated via single database)
- Higher flexibility of message transfer (Message on one bus type can be made available to other bus without influencing the existing setup)
- Reduces cost of the complete system (By reducing types of interface nodes between ECUs)
- Highly configurable system (Addition/deletion of nodes easily possible by just changing configuration only in the gateway ECU)
- Easily upgradeable to new technology (New bus or protocol can be incorporated with only adapting gateway node and not influencing any other existing nodes)
5. LIMITATIONS OF PROPOSED GATEWAY ECU

Some of the limitations of the proposed gateway ECU are listed down in Table 3.

Table 3 Limitation of the proposed gateway ECU

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Proposed gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Performance</td>
<td>Optimal performance with upto 10 MIPS and 20 MHz operation only</td>
</tr>
<tr>
<td>Message database</td>
<td>Upto 3.2 kbytes of RAM, 256 bytes of stack and 256 bytes for normal application, 1 Message structure need ~14 bytes (~200 messages only can be managed)</td>
</tr>
<tr>
<td>Number of CAN and LIN buses</td>
<td>Only one CAN and LIN buses possible</td>
</tr>
<tr>
<td>Coprocessor</td>
<td>No coprocessor available to share CPU load</td>
</tr>
</tbody>
</table>

The current implementation of CAN and LIN protocol has been validated against the OSI reference model and inferred that the current implementation does not have network and session layer.

6. CONCLUSION

The conclusions at the end of this study are listed down here:

• From literature review, it was concluded that the existing gateway ECUs does not provide cost effective solution for LPVs.
• It was also concluded that gateway ECU with CAN and LIN networks are sufficient to meet the requirements for LPVs.
• From the cost calculation, it was concluded that the proposed gateway ECU is very much suitable and affordable for LPVs.
• As the proposed gateway ECU have sufficient free memory (~70%), it is still possible to enhance the proposed gateway ECU with more features and customize according to the customer needs, without changing the hardware and cost.

REFERENCES


1) accessed on 21-Jan-2009