Engineering Doctorate

Innovation Report

Design of In-Vehicle Networked Control System Architectures through the Use of New Design to Cost and Weight Processes

Christopher Patrick Quigley

This work has been submitted in partial fulfilment of the requirements for the degree of Doctor of Engineering (EngD)

Warwick Manufacturing Group, University of Warwick

30th November 2011

Abstract

Over the last forty years, the use of electronic controls within the automotive industry has grown considerably. In-vehicle network technologies such as the Controller Area Network (CAN) and Local Interconnect Network (LIN) are used to connect Electronic Control Units (ECU) together, mainly to reduce the amount of wiring that would be required if hardwired integration were used.

Modern passenger cars contain many networks, which means that for the architecture designer, there is an almost overwhelming number of choices on how to design/partition the system depending on factors such as cost, weight, availability of ECUs, safety, Electro-Magnetic Compatibility (EMC) etc. Despite the increasing role played by in-vehicle networks in automotive electrical architectures, its design could currently be described as a "black art". Not only is there an almost overwhelming number of choices facing the designer, but there is currently a lack of a quantifiable process to aid decision making and there is a dearth of published literature available.

NetGen is a software tool used to design CAN/J1939, LIN and FlexRay networks. For the product to remain competitive, it is desirable to have novel features over the competition. This report describes a body of work, the aim of which was to research in-vehicle network design processes, and to provide an improvement to such processes. The opportunities of customer projects and availability of customer information resulted in the scope of the research focusing on the adoption of LIN technology and whether the adoption of it could reduce the cost and weight of the target architecture. The research can therefore be seen to address two issues: firstly the general problem of network designers needing to design in-vehicle network based architectures balancing the needs of many design targets such as cost, weight etc, and secondly the commercial motivation to find novel features for the design tool,

NetGen. The outcome of the research described in this report was the development of design processes that can be used for the selection of low cost and weight automotive electrical architectures using coarse information, such as that which would be easily available at the very beginning of a vehicle design programme. The key benefit of this is that a number of candidate networked architectures can be easily assessed for their ability to reduce cost and weight of the electrical architecture.

Acknowledgements

I would like to acknowledge the help from others in the completion of the Engineering Doctorate.

I would like to thank my academic supervisors Peter Jones and Ross McMurran and my industrial mentor Paul Faithfull for their helpful comments and guidance throughout the project.

Finally I would like to thank my partner Libby Hayward for proof reading final drafts of submissions and this report.

Declaration

This report is the original work by the author, parts of which have been published elsewhere, as stated in the text.

Contents

Table of Figures

List of Tables

Abbreviations

ABS	Anti-lock Braking System
API	Application Programming Interface
ASAM	Association for Standardisation of Automation and Measuring Systems
ASIC	Application Specific Integrated Circuit
ASIL	Automotive Safety Integrity Level
AUTOSAR	AUTomotive Open System Architecture
BEAN	Body Electronic Area Network (Toyota proprietary technology used for the control of body electronics)
CAN	Controller Area Network
CD	Compact Disk
COTS	Commercial Off The Shelf
CPU	Central Processing Unit
CSMA-CD-NDBA	Carrier Sense, Multiple Access, Collision Detection, Non-Destructive Bitwise Arbitration
DTC	Diagnostic Trouble Code
DVD	Digital Versatile Disk
EASIS	Electronic Architecture and Systems engineering for Integrated Safety Systems
ECU	Electronic Control Unit
EEPROM	Electrically Erasable Programmable Read Only Memory
EMC	Electromagnetic Compatibility
EPAS	Electrically Power Assisted Steering
EPB	Electric Park Brake

ESP	Electronic Stability Program
EUSART	Enhanced Universal Synchronous Asynchronous Receiver Transmitter
FPGA	Field Programmable Gate Array
HLP	Higher Layer Protocol
HVAC	Heating Ventilation Air Conditioning
ISG	Integrated Starter Generator
IVN	In-Vehicle Network
LAN	Local Area Network
LDF	LIN Description File
LIN	Local Interconnect Network
LNA	LIN Network Architect
MILCAN	Military Controller Area Network Protocol
MISRA	Motor Industry Software Reliability Association
MOST	Media Oriented Systems Transport
NM	Network Management
NMEA2000	National Marine Electronics Association 2000 Protocol
NRE	Non-Recoverable Expenditure
OEM	Original Equipment Manufacturer
PCB	Printed Circuit Board
PDU	Protocol Data Unit
PLC	Power Line Carrier
RAM	Random Access Memory
RMS	Root Mean Squared
ROM	Read Only Memory

RTOS	Real Time Operating System
SENT	Single Edge Nibble Transmission
SIL	Safety Integrity Level
SME	Small Medium Enterprise
SPC	Short PWM Code
TDMA	Time Division Media Access
TTCAN	Time Triggered CAN
TT-Ethernet	Time Triggered Ethernet
TTP	Time Triggered Protocol
UART	Universal Asynchronous Receiver Transmitter
USART	Universal Synchronous Asynchronous Receiver Transmitter

1. INTRODUCTION

Prior to the 1970s there were very few electronic systems implemented in passenger cars. By 1974 there were up to twelve electronic systems or sub-systems in a high end vehicle although they were not microcontroller based. These systems included alternator diodes, voltage regulators, electronic fuel injection and electronic controlled ignition (Jurgen, 1999). Electronic system adoption then began to grow as a result of increasingly stringent emissions legislation world wide (initially from the Californian Air Resources Board) and the emergence of cost effective microprocessor technology. By the end of the 1990s, electronic components and systems accounted for over 20% of the cost of a high end passenger car (Leen et al, 1999). A current high-end passenger car can have over fifty Electronic Control Units (ECUs) of varying complexity contained within its electrical architecture. This figure is expected to grow significantly over the coming years with the introduction of increasingly more complex control systems such as drive-by-wire and multimedia systems giving access to the Internet. Forecasts indicate that by 2010, 24% of the total vehicle costs across all types of vehicle will be due to electronic equipment (Robert Bosch GmbH, 2004).

The large number of ECUs that now exist in modern vehicles has led to the adoption of in-vehicle networks in order to share information between each of the ECUs. There have been many example technologies over the years but currently the de-facto standard is the Controller Area Network (CAN). More recently intelligent sensors and actuators, and low speed digital switching have been integrated with a complementary but lower cost and lower performance technology known as the Local Interconnect Network (LIN). Since 2006 a new higher cost but higher performance network technology known as FlexRay has been implemented in cars from BMW, Audi, Rolls Royce and Bentley. Finally, a technology known as Media Oriented Systems Transport (MOST) is now deployed in many higher-end vehicles for infotainment applications such as camera and navigation systems. MOST is a fast network technology for information transfer and not control and therefore it can be considered as different from the aforementioned network technologies. A typical modern vehicle will now have at least two CAN buses and four to eight LIN buses. A high-end vehicle can have up to six CAN buses and twelve LIN buses and therefore system partitioning can be a significant problem in the design of the electrical architecture.

The designers of a vehicle's electrical architecture must balance the requirements of many different design targets in order to produce the best architecture. These include but are not limited to ensuring that it is the lowest cost to meet the requirement of delivering the maximum value to a customer, achieving the lowest weight possible, which in turn reduces emissions and fuel consumption, ensuring that it is easy to manufacture and assemble and also that it meets legislative requirements in terms of Electro-Magnetic Compatibility (EMC) and safety. Much of this is carried out relying on the judgement and experience of the designers since it is a "black art" rather than using fully quantified design processes.

1.1. The Sponsoring Company – Rapicore

Rapicore was a spin-off company from Potenza Technology founded in 2004 to commercially exploit and develop products relating to design and automatic code generation of communication software stacks for automotive network systems. Potenza Technology itself is part of a group that now owns niche sportscar manufacturer brands Westfield and GTM. Potenza gave access to some information on the design of these vehicles during the doctorate programme which is contained within this report.

Rapicore products included a CAN bus I/O block for Simulink (called RapidTarget) which was targeted towards the Infineon c16x microcontroller family and also integrated into the Prodrive uProteus product. Another key product was an in-vehicle networking design tool called NetGen. Due to a decline in interest in these products, the company was wound down in 2009. During the five year life of the company it consisted of up to five people and participated in the Eureka project SAPECS (Secured Architectures and Protocols for Enhanced Car Safety).

The Rapid Target product is no longer supported as the high cost of maintenance makes this product commercially unviable. The NetGen product, although not financially viable within Rapicore, is now a part of the product portfolio of a company called Warwick Control Technologies, which is a provider of control systems networking technology products for automotive and industrial automation systems. The NetGen intellectual property is currently jointly owned by Potenza Technology and Warwick Control Technologies.

1.2. The Product – NetGen

Background

NetGen is a PC-based network design and automatic code generation tool for LIN, CAN, SAE J1939 (a version of CAN used in truck and off-highway industries) and FlexRay. The NetGen tool is a rule-based LIN network and node designer that can also be used to automatically generate or configure the MISRA C source code stack for network communications. The stack is compiled and included as part of an automotive application that is embedded within an ECU.

NetGen LIN

The first version of NetGen to be released commercially was NetGen LIN. The current versions of NetGen LIN available on the market are for Atmel AVR and Infineon Tricore microcontrollers and have been included in a number of production vehicle programmes. The Infineon Tricore version has been used by a major first tier supplier to generate the communications between a gearbox controller and a gear selector module for a large number of manufacturers. Atmel AVR versions have been used by first tier automotive suppliers in the USA for development of LIN slave devices for $C0_2$ sensing and Heating Ventilation and Air Conditioning (HVAC).

NetGen FlexRay Development

During the SAPECS Project (Secured Architectures and Protocols for Enhanced Car Safety) which was a pan-European Eureka project, the NetGen tool was extended to support the FlexRay protocol. The tool was included as part of the development process for the SAPECS project to design the FlexRay communications of an engine management demonstrator for a first tier European automotive supplier. It was used to automatically generate the C configuration files for an AUTOSAR FlexRay stack targeted to an Atmel AVR32 microcontroller (Laes et al, 2009).

NetGen SAE J1939

The SAE J1939 is a CAN higher layer protocol primarily used in bus, truck and off-highway applications. A J1939/CAN version was developed and extended to support the application programming interface of a commercial J1939 stack as the target. Currently NetGen CAN J1939 is not targeted towards a particular microcontroller, but to CAN-specific software stacks provided by two companies. However for the sake of this discussion, the J1939 software stack can be seen to be in the same scope as a microcontroller.

Summary of NetGen Product Variants

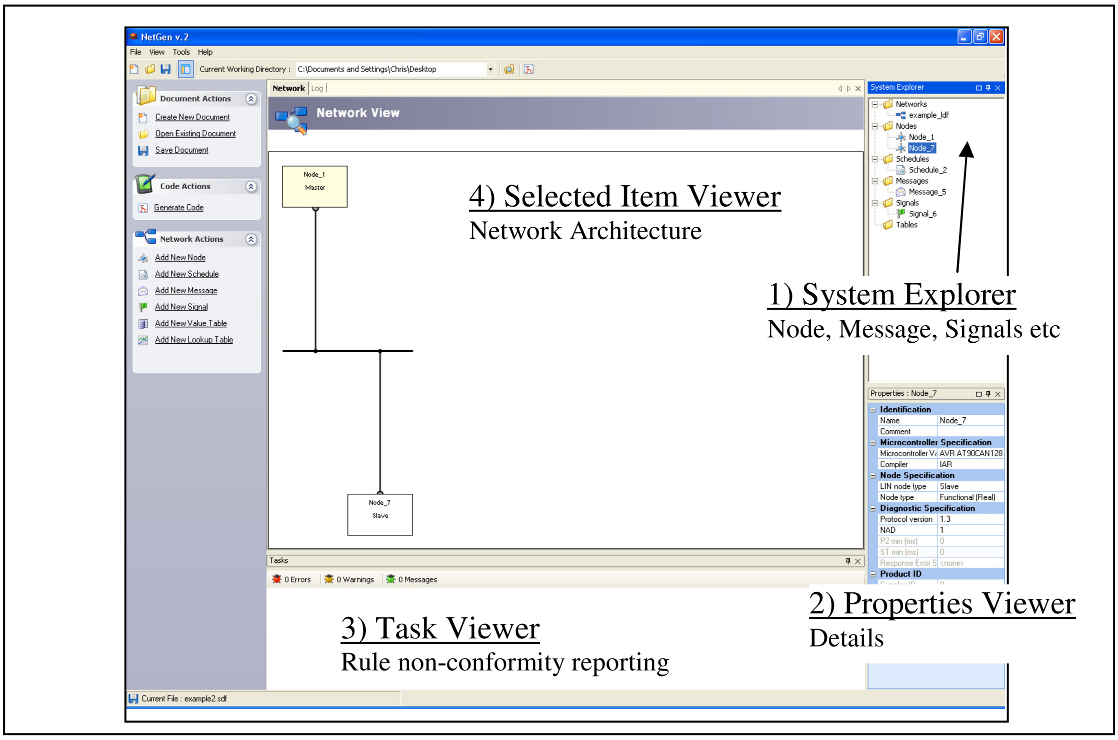
The different versions of the tool that are currently available are summarised in Table 1. It can be seen that the generic naming convention is *NetGen "Network Technology" "Microcontroller Targeted"*. The list of product versions is not exhaustive, these are just the versions that have been developed specifically to meet customer requests or other market needs that have been determined.

Technology	Product offering
LIN	*NetGen LIN Tricore* – automatic code generation tool for LIN targeted at the Tricore microcontroller from Infineon AG and the AVR microcontroller from the Atmel Corporation
CAN	*NetGen CAN LPC* – automatic code generation tool for CAN targeted at the LPC microcontroller from NXP
J1939	*NetGen CAN J1939* – automatic code generation tool for CAN targeted at the J1939 software stacks from two different suppliers
FlexRay	*NetGen FlexRay AVR32* – automatic code generation tool for FlexRay targeted at the AVR32 microcontroller from the Atmel Corporation.

Table 1 : Currently available product versions by network technology

1.3. Distributed System Development Using NetGen

A screen shot of the tool is shown in Figure 1 which shows an example two node network for LIN. The network architecture display shows a network topology. The Node, Message, Signals Tree View shows the number and name of network nodes, messages, signals and also schedules for message transmission. During the development of a network, there are protocol conformity rules that continually check the correctness of the design and provide three priority levels of warnings at the bottom. Figure 2 shows the NetGen view for schedules of message transmission and how they can be designed within the tool.



Identification	
Name	Node_7
Comment	
Microcontroller Specification	
Microcontroller V	AVR AT90CAN128
Compiler	IAR
Node Specification	
LIN node type	Slave
Node type	Functional (Real)
Diagnostic Specification	
Protocol version	1.3
NAD	1
P2 min (ms)	0
ST min (ms)	0
Response Error S	<none>
Product ID	

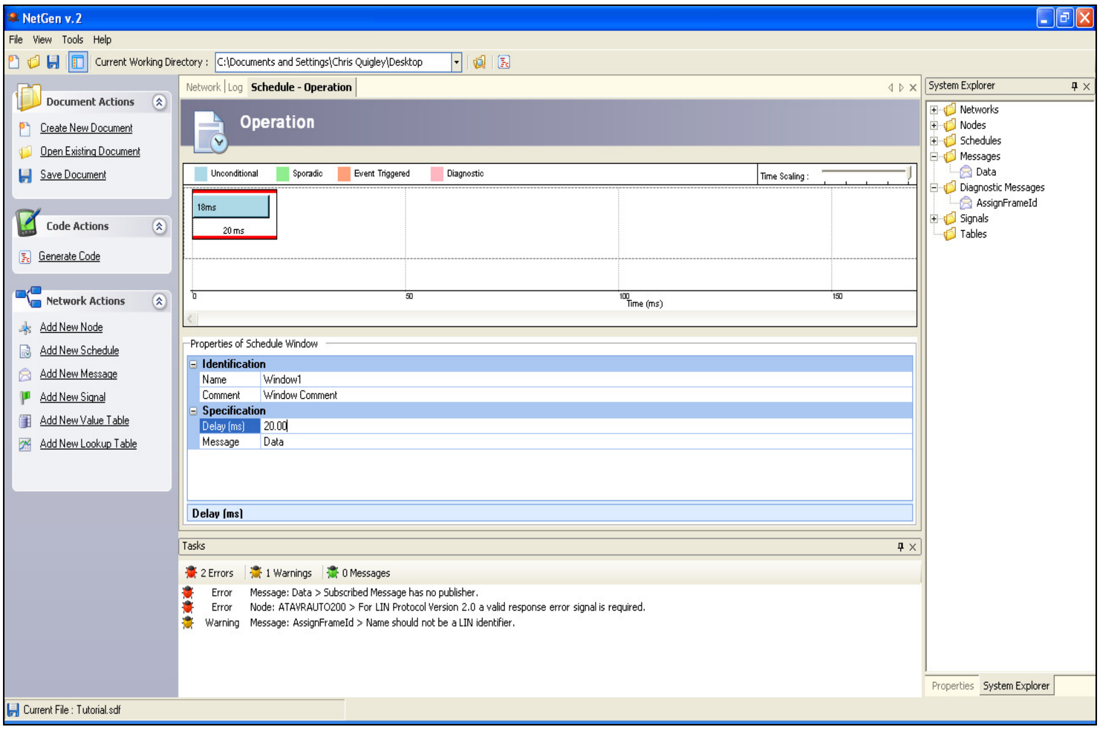
4) Selected Item Viewer
Network Architecture

1) System Explorer
Node, Message, Signals etc

2) Properties Viewer
Details

3) Task Viewer
Rule non-conformity reporting

Figure 1 : Example screen shot of the NetGen LIN tool



Properties of Schedule Window

Identification	
Name	Window1
Comment	Window Comment
Specification	
Delay (ms)	20.00
Message	Data

Delay (ms)

```
Error    Message: Data > Subscribed Message has no publisher.
Error    Node: ATAVRAUTO200 > For LIN Protocol Version 2.0 a valid response error signal is required.
Warning  Message: AssignFrameId > Name should not be a LIN identifier.
```

Figure 2 : Example LIN schedule screen shot of the NetGen LIN tool

Upon opening NetGen and selecting to create a new configuration, the user is presented with a selection of technologies for which they can configure. At present the options include CAN/J1939, LIN and FlexRay. Once the desired network is selected the user is presented with the main configuration environment shown in Figure 1. The interface consists of four main components: the *System Explorer (1)*; the *Properties Viewer (2)*; and the *Task Viewer (3)*; *and* the *Selected Item Viewer (4)*. For the purposes of this discussion a network has already been entered to show the kind of information that is displayed.

The *System Explorer* (1) shows all of the components within the network, including the network itself, its nodes, schedules of communication, messages, signals and lookup tables. This allows developers to view all of the network's constituents at a glance. The configuration parameters for each item can be viewed and edited in the *Properties Viewer* (2) by selecting the item with the mouse. Items can also be copied and deleted through the System Explorer.

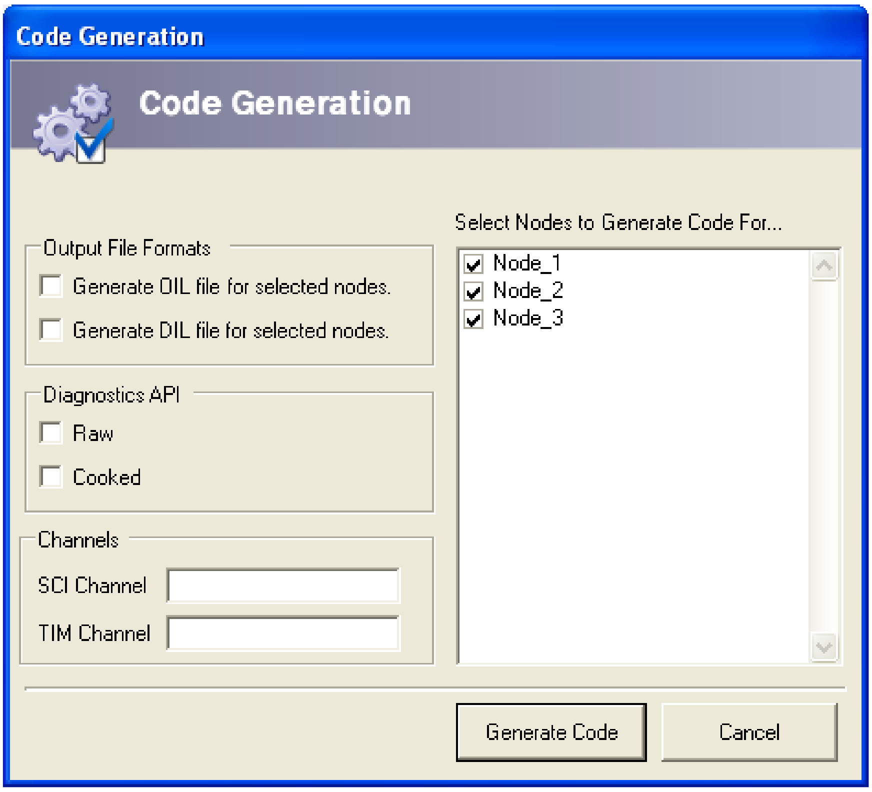
The *Properties Viewer* (2) displays the parameters and settings of the item selected in the *System Explorer*. Through this component the user can modify the item's parameters as required. It also performs checks on the values entered or selected according to rules contained in the schema files. If a parameter is invalid an error or warning message, depending on the nature of the invalidity, is shown in the *Task Viewer* (3). Some errors can also result in a dialog box being displayed, showing the error that has occurred and the action that has been performed e.g. the value is above the permitted bounds and has been reverted back to its original value.

The *Task Viewer* (3) shows any errors, warnings and information concerning the network's current configuration, and offers the user guidance on how to correct the network design. If an error is highlighted in the *Task Viewer*, the user simply has to make the associated correction. Once the item is corrected the error/warning is removed from the viewer. During the NetGen development process the user must work through all errors and warnings that are reported to satisfactorily complete the network design.

The *Selected Item Viewer* (4) displays a number of items depending on the selection in the *System Explorer*. The viewer is used for items that require, or can benefit from, the use of visual artefacts, either for simple display purposes or for more complicated configurations that cannot be done through the *Properties Viewer*. The main items displayed are the *Network* (currently shown in the figure), the *Log* (which displays information on the applications events), and the Schedule, (for modifying the network's schedule which is required for LIN and FlexRay, but not CAN). The Schedule view is the most interactive of the displays, allowing users to set up messages within the schedule.

Throughout development the developer can save (and later reopen) the network configuration to file. The file is called a System Description File and uses the 'SDF' extension and is saved as an XML file.

Once the entire network configuration has been entered correctly and all errors/warnings reported in the *Task Viewer* have been dealt with satisfactorily, the developer can then generate the source code for one or more of the nodes present in the network. The code generation interface is shown in Figure 3. The simple code generation interface allows the user to select the nodes for which the code should be generated. It also allows the configuration of a few other items including the generation of OIL (OSEK Implementation Language) and DIL (Data Input Language) files for each node, diagnostics API, and SCI (Serial Communication Interface)/TIM (Timer) Channel selection (for LIN only). Once the required nodes have been selected the developer clicks on the ‘Generate Code’ button to generate the source code for the selected nodes. These files are saved into a separate directory on the local machine for each node selected.



Code Generation

Output File Formats

☐ Generate OIL file for selected nodes.

☐ Generate DIL file for selected nodes.

Diagnostics API

☐ Raw

☐ Cooked

Channels

SCI Channel

TIM Channel

Select Nodes to Generate Code For...

☑ Node_1

☑ Node_2

☑ Node_3

Figure 3 : The NetGen code generation interface

1.4. NetGen – Competitor Analysis

The competitors for the NetGen tool vary depending upon the network technology that they are targeted towards. The main competitor products to NetGen network design and code generation tool are shown in Table 2.

Feature	NetGen	Vector (DaVinci)	Intrepid (LIN Tool)	Mentor LNA	TTTech TT-Plan	TZM FlexConfig	E-bit
Automatic Scheduling of messages	No	No	No	Yes	Yes	No	Yes
Code Generator	Yes	Yes	No	Yes	Yes	No	Yes
Rule Based Design	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Design Process Workflow	Yes	Yes	None	Yes	Yes	Yes	Yes
Microcontroller selection (e.g. by stack ROM/RAM estimation)	No	No	No	No	No	No	No
Support of architecture design to a target (e.g. Cost/weight)	No	No	No	No	No	No	No
FIBEX support	No	Yes	No	No	Yes	Yes	Yes
Pricing Point	Medium	High	Free	High	High	Medium	High
CAN Support	Yes	Yes	No	Yes	TTCAN	No	No
J1939 Support	Yes	Yes	No	No	No	No	No
LIN Support	Yes	Yes	Yes	Yes	Yes	No	No
FlexRay Support	Yes	Yes	No	No	Yes	Yes	Yes

Table 2: NetGen competitor products as of end of 2009 (Green=NetGen feature coincidence)

The following can be seen from Table 2:

- Only the product from Vector competes across all network technologies (i.e. CAN, J1939, LIN and FlexRay).
- J1939 versions of the tool only have one competitor, which is Vector.
- For FlexRay the market place is very crowded and there is a lot of competition for the FlexRay version of the NetGen.
- The LIN version of the NetGen product has just two serious competitor products (from Vector and Mentor), as two other competitors do not provide the same level of features such as C code generation.
- None of the tools help with microcontroller selection

- None of the tools helps with the selection of the ideal network architecture that can result in reduced cost or weight.

Therefore there are technological gaps in the market for such products in the area of microcontroller selection and architecture selection based on design targets such as cost or weight.

In general NetGen offers the following benefits in addition to what is shown in this table:

1. **Network design and code generator available in same package** – many of NetGen's competitors separate the designer side of the product from the code generator whilst NetGen includes both in the same package.
2. **Easy-to-use user interface** – therefore requiring minimal training.
3. **Customisable rule-base** – editable XML schema offers an easy to customise rule-base.
4. **Competitive pricing** – middle end pricing point makes the product ideal for smaller suppliers and niche /low-volume vehicle manufacturers.
5. **Multiple stack supplier relationships** – not tied to a single stack supplier, i.e. currently supports FlexRay for Atmel AVR32, two different J1939 stacks, HIS Automotive standard for LIN, Atmel AVR microcontrollers.
6. **Integration with the free GNU C compiler –** giving the possibility for the customer to reduce their capital investment cost.
7. **Free CAN stack** – signals API-based stack available with CAN version based on early AUTOSAR standards.

1.5. NetGen – Current Sales and Distribution Strategy

The NetGen product is currently sold world-wide as a part of the Warwick Control in-vehicle network development tool portfolio, primarily via the Warwick Control web-shop, direct sales and distribution partnerships. The product portfolio is shown in Figure 4, and within this portfolio the NetGen product is able to survive commercially as part of a solution that can be supplied to customers.

The product portfolio is a design suite which includes NetGen as a network designer and automatic code generator (i.e. generation of signals API and configuration for the network stack), an automatic J1939 CAN Network Documentation Tool (NDT) for generation of network specifications, Network Comparison Tool (NCT) to help manage the data between different revisions of the vehicle network specification, network analyser and tool for flashing over CAN.

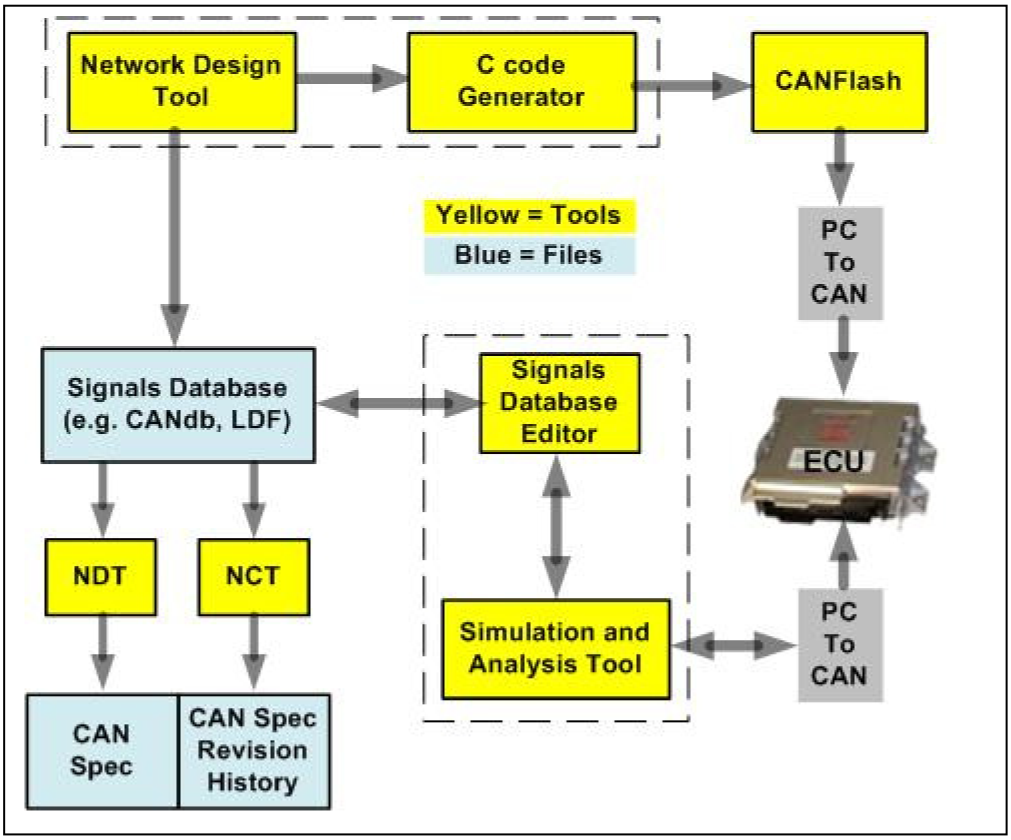


Figure 4 : Warwick Control Toolset for ECU development (McLaughlin et al, 2010)

The tool faces some tough competition from some very large companies. To help improve product exposure, versions of NetGen without any code generation capabilities are given away for free. This is inline with current approaches from competitors. For example, at least two competitors give away a free version of their LIN editor tool. This has the advantage of allowing the customer to try out the product's features before purchase.

FlexRay and CAN versions of the NetGen product are available as custom products, in other words developed specifically as customer projects to their requirements. LIN and J1939 NetGen versions are available as off-the-shelf and also custom products. Custom versions of the tool can be developed for microcontroller manufacturers, first tier tool suppliers, car manufacturers and first tier ECU suppliers.

1.6. NetGen – Requirement for Improvement

For the NetGen product to survive against market threats, the product needs to be taken in new directions. The higher-end of the market (which is represented by tools with a high level of features and commanding higher pricing) is very crowded. This is especially true for FlexRay tools. Therefore NetGen FlexRay has a large amount of competition. There is the potential for NetGen FlexRay to be a lower cost alternative but the disadvantage of being dominant in the low price sectors is that it can be hard to get large revenues in comparison to the competitors in the high-end pricing sectors. The main reason for this is that the expected number of license sales would not be large enough, even if FlexRay experienced a high level of adoption. Currently the adoption of FlexRay is very low due to company budget limitations and therefore the market is small.

NetGen LIN only has serious competition from two competitors and NetGen J1939 only has serious competition from one competitor. It is desirable to increase revenues by further differentiating the product from the competition and move its market positioning to include higher features and therefore be able to command a higher price. Therefore there is a requirement for the development of novel features to help this product differentiation.

The general market for network stacks is in decline due to the continual emergence of open source projects and an increased perception from customers that free stacks can be successfully integrated into their embedded system products.

1.7. Aim of Research

As has previously been stated, the successful design of automotive electrical architectures involves balancing of the requirements of many different factors. Even if the number of factors is limited by the design team, the design of an automotive electrical architecture based on in-vehicle networking technology is a very difficult problem to solve. There is very little published literature providing a formalised process for such a design, probably due to the commercial sensitivity. There is also the problem that the amount of information that is required to make design decisions is potentially overwhelming. There is a lack of process for dealing with this information. Therefore the design of an automotive electrical architecture could be described as a black art. The aim of the research described in this report was to ultimately help improve the understanding of the possible decisions involved in the design of an automotive electrical architecture based on in-vehicle networking technology and to formalise the design processes.

The requirements of one of NetGen's customers led the research to initially focus on designing an automotive electrical architecture to a specified target cost. The requirements of a second customer led to the addition of a second factor and therefore the focus was widened to include designing an electrical architecture to a target weight. The requirements of these two customers led to the two case studies that are described in chapters three and six.

The commercial aim was to provide a commercial advantage for the NetGen tool and explore how this could be achieved. To enable further NetGen product differentiation and a move to the higher-end of the market, research was required to ascertain potential new and innovative features. However there was particular emphasis on how much information could be estimated at the very early design stages so that the information required for design decisions could be reduced.

The scope of the research was limited to LIN technology due to the commercial influence of the NetGen LIN product. LIN was the biggest market for the NetGen tool and therefore potential projects and data had the greatest availability in this area.

To achieve the academic and commercial aims, there were the following research objectives. The first was to research the state of the art literature on in-vehicle networking and the processes used in their design. Secondly to investigate the cost of an automotive door wiring harness which is integrated by hardwiring (in other words not using an in-vehicle networking technology), develop a process for assessing whether a LIN alternative electrical architecture can reduce the cost of the harness and apply the process on the automotive door wiring harness case study. The final research objective was to apply the previously developed cost assessment process on a new case study, a niche sports car manufacturer electrical architecture.

1.8. Flow of Doctorate Submissions

The outcome of the Engineering Doctorate research was that a number of processes were developed through the case studies. These can be expressed as part of one overall process which is summarised in Figure 5. The overall process consisted of four steps. Step 1 is concerned with the selection of the target architecture for a vehicle which is to be analysed for cost or weight improvement. Step 2 is concerned with the proposal of one or more alternative architectures to be compared. Step 3 is concerned with the analysis of the proposed architectures to see if they can achieve lower cost/weight and determine what nodal cost/weights (C_{Node} and W_{Node}) are required to achieve an equal or lower cost/weight target. Step 4 is concerned with ascertaining whether nodal cost and weight targets (C_{Node} and W_{Node}) are realistically achievable by looking at the communications stack ROM/RAM requirements. The development of this process is described in the various submissions that were written throughout the doctorate.

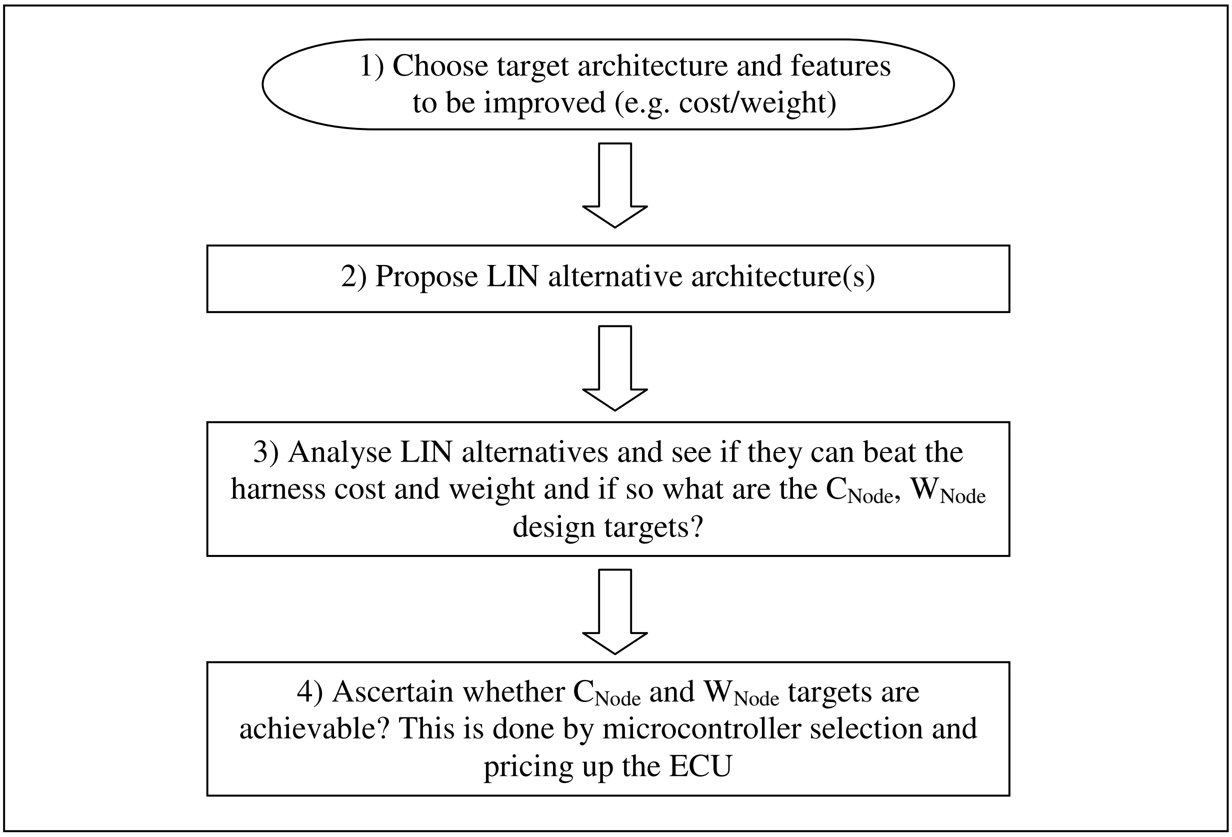

1) Choose target architecture and features to be improved (e.g. cost/weight)
2) Propose LIN alternative architecture(s)
3) Analyse LIN alternatives and see if they can beat the harness cost and weight and if so what are the C_{Node}, W_{Node} design targets?
4) Ascertain whether C_{Node} and W_{Node} targets are achievable? This is done by microcontroller selection and pricing up the ECU

Figure 5 : Top level process developed during the engineering doctorate

There were five submissions that were submitted to the Engineering Doctorate portfolio:

1. Review of Automotive Electronic Control Systems –Trends, Technologies, Processes, and Standards
2. A Comparison of the Cost of Hardwired and LIN Bus Based Car Door Electrical Architectures
3. An Investigation into the Relationship between Microcontroller Monetary Cost and ROM/RAM Capability for Improved Understanding of Automotive Local Interconnect Network Node Cost Issues
4. Modelling of Automotive Microcontroller LIN Communications Stack ROM and RAM Requirements for Improved Cost Estimation

5. A Case Study on the Design to Cost/Low Harness Weight Processes for Local Interconnect Network Based Alternatives to a Niche Sports Car Hardwired Wiring Harness

The suggested reading order is shown in Figure 6. Submission one started the research by looking at the general trends in the adoption of the in-vehicle electronics and in-vehicle networking technologies for the integration of the electronic systems and design process.

The motivation for submission two came from a partnership between a major Japanese wiring harness manufacturer with offices in the UK, and an automotive LIN microcontroller manufacturer with offices in France, both of whom use the NetGen product. The wiring harness manufacturer wanted to see if they could reduce the cost of one of their door system wiring harnesses by the adoption of LIN. Cost data of the bill of materials of the original hardwired driver's door harness was provided by the wiring harness manufacturer. This was analysed and a number of findings came out of this. Firstly it is not very useful to analyse the door harness on its own but best to look at the entire door system. By dividing the entire door system into zones, the harness could be analysed in terms of inter-zone and intra-zone wires (a zone in this case being a vehicle door). The process that was followed is described in chapter four of this report. The study revealed that a very challenging nodal cost target would be needed in order for LIN communications to be added.

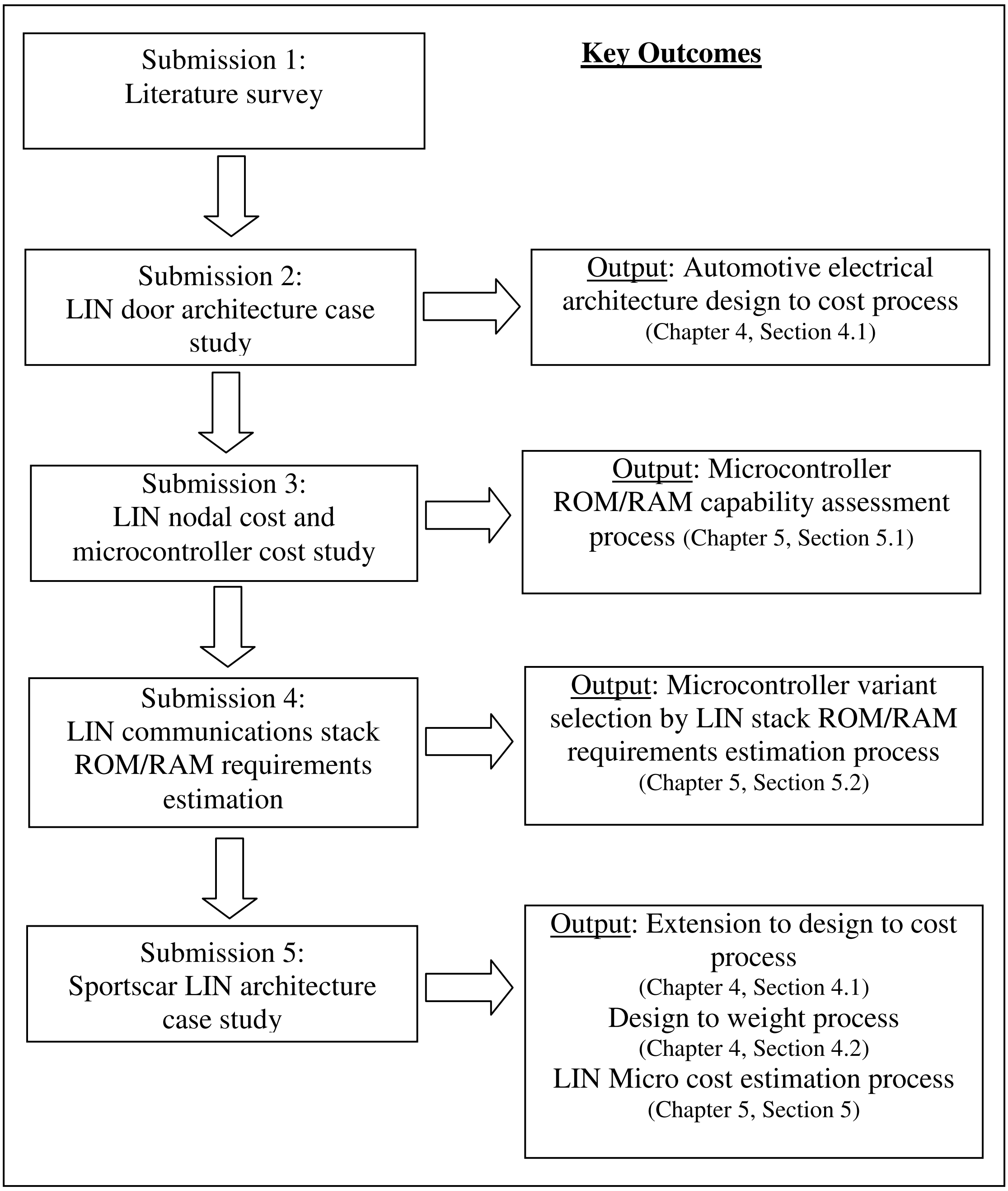

Submission 1:
Literature survey
Key Outcomes
Submission 2:
LIN door architecture case study
Output: Automotive electrical architecture design to cost process
(Chapter 4, Section 4.1)
Submission 3:
LIN nodal cost and microcontroller cost study
Output: Microcontroller ROM/RAM capability assessment process (Chapter 5, Section 5.1)
Submission 4:
LIN communications stack ROM/RAM requirements estimation
Output: Microcontroller variant selection by LIN stack ROM/RAM requirements estimation process
(Chapter 5, Section 5.2)
Submission 5:
Sportscar LIN architecture case study
Output: Extension to design to cost process
(Chapter 4, Section 4.1)
Design to weight process
(Chapter 4, Section 4.2)
LIN Micro cost estimation process
(Chapter 5, Section 5)

Figure 6 : Suggested reading order of the Engineering Doctorate submissions and key outcomes

Submission three was motivated by submission two and investigated LIN node cost issues. In particular the relationship between microcontroller unit price and its memory capabilities was investigated for the Microchip 16F and 18F microcontroller families. This was significant as it had the potential to partially quantify the cost of an embedded software component by characterising its ROM and RAM requirements. It was found that the ROM and RAM capabilities did have an effect on unit price but not clearly enough to be able to determine a generic model for microcontroller price estimation. Further investigation showed that there was a linear relationship between the ROM and RAM capability of the microcontroller and that this relationship varied between families (e.g. PIC 16F, 18F and Atmel AVR). This relationship could be used to determine whether a piece of embedded software such as a LIN communications stack could easily be targeted towards a particular microcontroller.

Submission four continued on the subject of the cost of embedded software relating to microcontroller ROM and RAM. This submission investigated whether it was possible to produce models to estimate the ROM and RAM requirements of a LIN communications stack as a function of the number of network nodes, messages and signals. It was found that linear regression modelling did allow estimations to be made.

Submission five applied provided the opportunity to apply the processes developed in the other submissions to a new case study. The aim was to ascertain whether adoption of LIN in body control could be lower cost and lower weight than the hardwired original in a niche sports car. The design-to-cost process that was used in submission two was adapted to form a design-for-low-architecture weight process. It was found that there was likely to be an additional cost rather than a cost saving. However there was potential to reduce the weight with the adoption of LIN.

1.9. Flow of this Report

Chapter two of this Innovation Report summarises the key points from the literature survey which was outlined in submission one and also makes some updates to the literature based on current developments.

Chapter three summarises the case study that was carried out in submission two, which investigated the design-to-cost process of an automotive door electrical architecture based on LIN.

The case study described in chapter three resulted in the development of a process for the design-to-cost of an electrical architecture that allowed an original hardwired architecture to be compared with one or more LIN candidate architectures to ascertain if they can be of lower cost. This resulted in the determination of a target nodal cost that would be required for the LIN architecture to be of lower cost. The process for this is described in chapter four. A design-to-weight process that allows two architectures to be compared in terms of their weight was developed based on the design-to-cost process. This is important since it helps low weight architectures be developed and was required for the second case study that is described in chapter six. Both the design-to-cost and design-to-weight processes are described in chapter four of this report.

Chapter five outlines the research carried out in submissions three and four that resulted in two processes which are also described in this chapter. The first of these is a process for the assessment of the ROM and RAM capability of a family of microcontrollers so that it can be ascertained how well the family is suited for the design of certain embedded software. The second of these processes is for ROM and RAM requirements estimation of a LIN communications stack so that a microcontroller variant can be selected.

Chapter six describes the case study that was carried out in submission five which investigated the design-to-cost and design-for-low-harness-weight of a sportscar body control electrical architecture. This case study used the design-to-cost/weight processes that are described in chapter four and the microcontroller variant selection process that is described in chapter five.

Chapter seven concludes this Innovation Report and summarises the key innovations from the research carried out. It also outlines recommendations for further research.

2. LITERATURE SURVEY

2.1. Trends in Automotive Electronic Systems Integration

Early microcontroller based automotive control systems had a small number of ECUs with sensors and actuators connected directly to the ECU that used the signal. This often resulted in duplicate sensors being used providing the same information. An example of this method of ECU integration is shown in Figure 7.

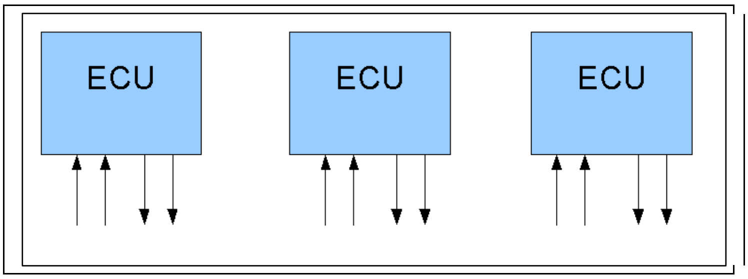


Figure 7: Stand alone ECUs (No Integration)

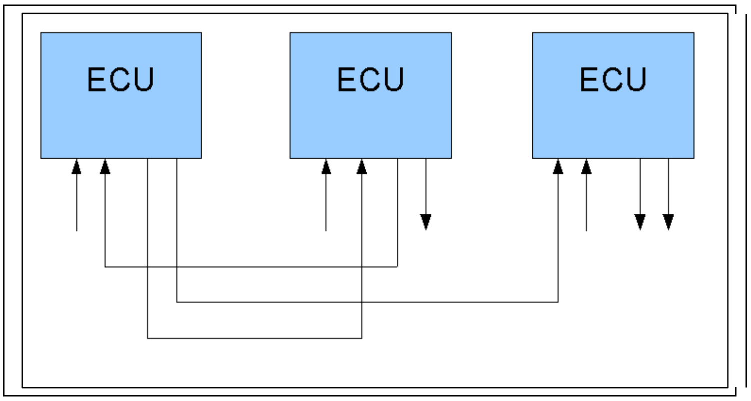


Figure 8: Hard-wired ECU Integration

An improvement to stand alone ECUs was to integrate ECUs with hard-wired signals. Therefore if two ECUs required the same signal, they could share the information from one sensor. This provided the benefit of cost saving by reduction of duplicate sensors. However, as the number of ECUs and sensors grew, this approach to ECU integration became very complicated and the weight of the wiring harness grew significantly. A large number of connectors within the wiring harness were required, which in turn led to reliability problems. Another significant problem of the hardwired integration method is cross-talk between wires caused by induced electrical interference (Khoh, 1993). An example of this method of integration is shown in Figure 8.

The problems associated with hard-wired integration led to the adoption of digital networks. Networks can have ring, star or bus architectures. The single bus architecture was adopted by the automotive industry for integration of ECUs since it was the most economically viable. This provided a number of benefits such as reduced wiring harness weight, reduction of the number of connectors, increased reliability, simplified assembly and ease of upgradeability for new ECUs and automotive platform customer option ECU management (McLaughlin, 1993). An example of this method of integration is shown in Figure 9.

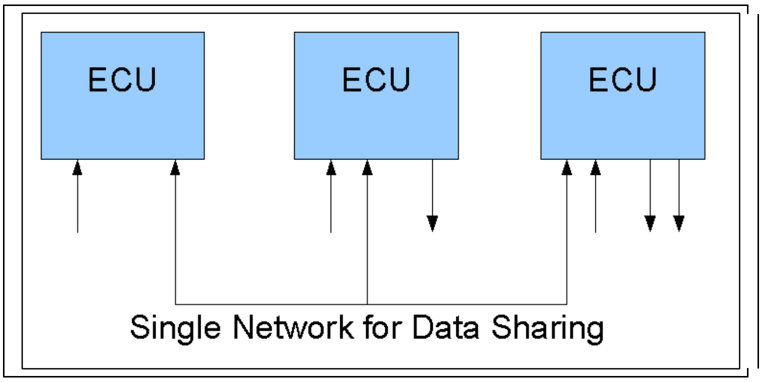


Figure 9: Single Bus Integration

Early data bus systems were based on some kind of message arbitration methodology which resulted in message latency becoming very unpredictable at higher bus loads. Therefore to maintain reasonable message latency, it became appropriate to partition the automotive electrical architecture between hard real time and soft real time functionality. An example of this is shown in Figure 10 in which powertrain (e.g. engine management, gearbox control) or chassis systems (e.g. braking, steering controls) are integrated via the high speed data bus and body control systems (e.g. lighting and door controls) are integrated via the low speed data bus.

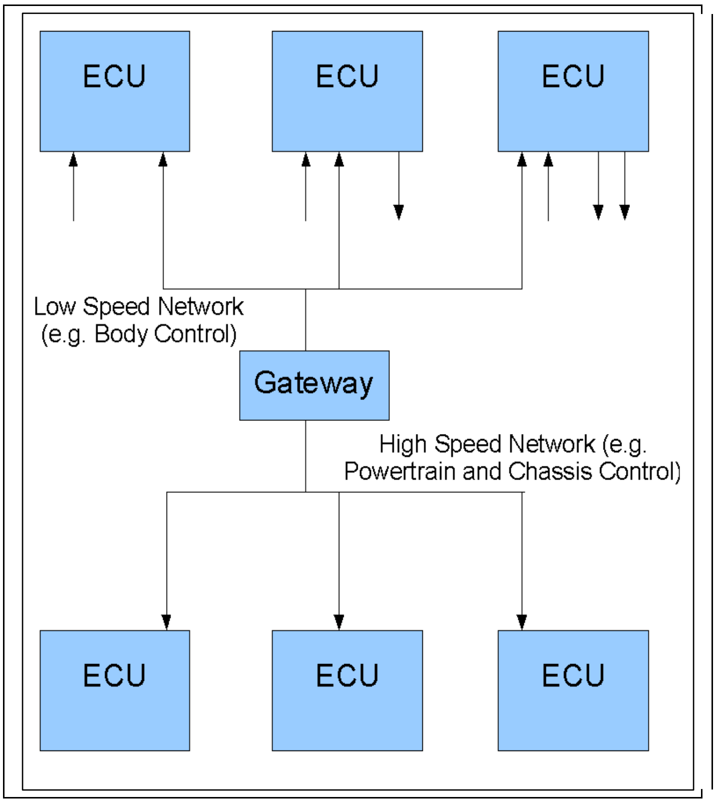


Figure 10: Partitioning of Automotive Electrical Architecture into a Two Bus System

The adoption of electronic control systems continued to grow as a result of the integration ability provided by early in-vehicle network systems. This led to two problems, partitioning and optimal cost. A dual network system had limitations when the number of electronic systems (and therefore ECUs) grew beyond the capabilities of a two-network system. Therefore the number of networks required within the vehicle's electrical architecture increased. However, partitioning the system into similar levels of safety criticality and real-time requirements can lead to a single network technology being sub-optimal. It can be found to be too powerful for some low-end applications (therefore wasting resources and money) or not powerful enough for high end applications. Therefore multiple network partitioned automotive systems have been the state of the art as shown in Figure 11. Infotainment, real-time control, low speed body and sensor/actuator bus systems are shown in Figure 11.

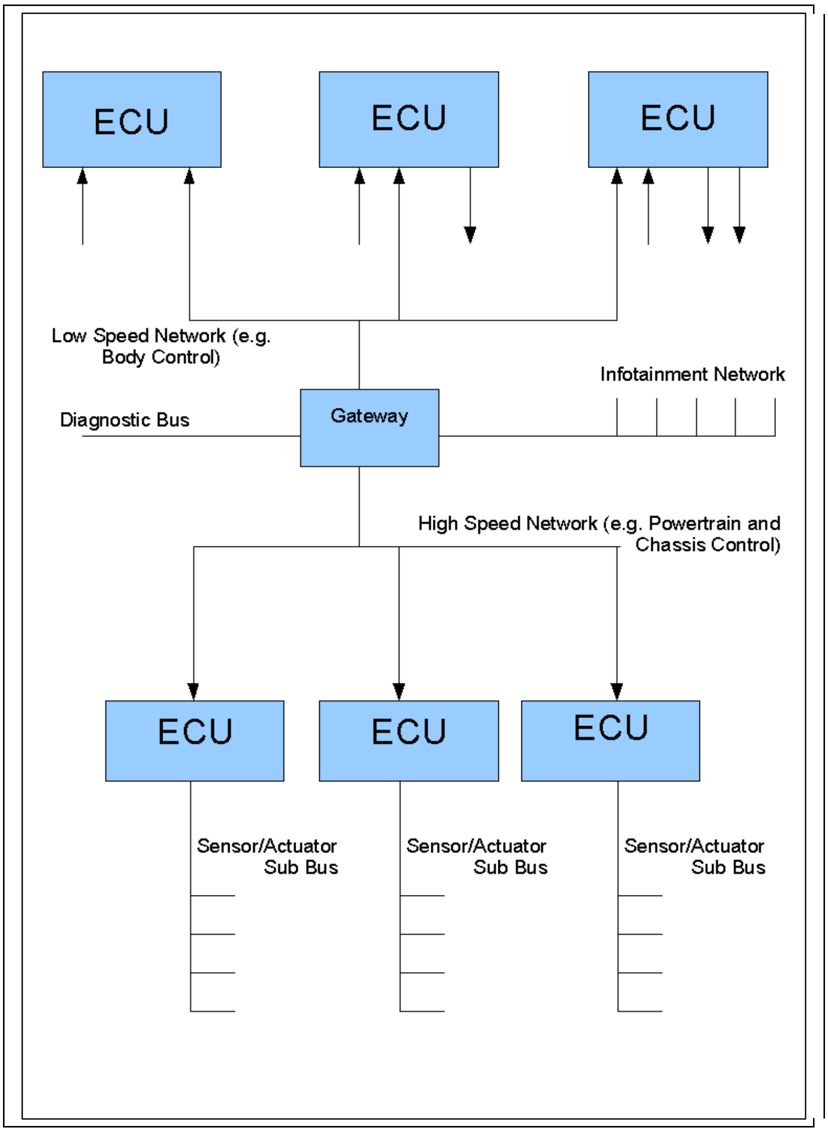


Figure 11: Multiple Network Partitioned Automotive Electrical Architecture (e.g. Real Time Control Networks, Sensor/Actuator Sub-buses, One Infotainment Network, One Diagnostic Bus - ideally partitioned by function but more usually by data sharing requirements)

2.2. Automotive In-Vehicle Networking Technologies for Control

Open standard in-vehicle data networks such as SAE J1850, SAE J1597, SAE J2106 and the Vehicle Area Network (VAN; ISO11519 Part 3) have been successfully applied to many mass production motor vehicles over the last twenty years. BEAN (Body Electronics Area Network) has been adopted as a proprietary protocol used specifically in Toyota. Their successful application has been due to their ability to deliver low system cost, lower weight through reduced wiring loom size, increased reliability, maintainability and sensor data sharing. The fact that such protocols have been open-standard is one of the over-riding causes of their lower cost, since suppliers to the automotive OEMs (both systems and semiconductors) can quickly and economically access the technology and supply different OEMs with similar products. Although these open network protocols existed, suppliers had to adapt to whatever protocol was used by their suppliers. This was costly in terms of tools, expertise and software maintenance.

2.2.1. Controller Area Network Based Technologies

CAN was developed by Robert Bosch GmbH in the 1990s (CAN Specification Version 2.0; 1991). Of the currently available open standard network protocols, CAN has become the most prominent across the world's automotive industry and is the de-facto standard adopted by most automotive manufacturers. The probable reasons for the significant adoption have been its huge support from major semiconductor manufacturers, tool suppliers and automotive OEMs. Now, nearly all automotive OEMs worldwide have products available with CAN or are intending to develop with CAN in the near future. The Controller Area Network is standardised under ISO-11898, which itself is split into four parts. The main features of CAN are summarised as:-

- Event based communication
- Standard CAN (11 bit identifiers) and Extended CAN (29 bit identifiers introduced for the US truck and bus industry in SAE J1939)
- Bus access is Carrier Sense Multiple Access Collision Detection (CSMA/CD) with Non-Destructive Bitwise Arbitration (NDBA), the lowest value identifier has highest priority for bus access
- Retransmission of messages that lose arbitration
- Silicon available from most semiconductor manufacturers

There are at least four physical layers available to suit different applications: CAN High Speed up to 1Mbaud (ISO-11898), Low Speed or Fault Tolerant CAN at up to 125Kbaud (ISO-11519-2), truck and bus protocol up to 250Kbaud (ISO-11992 Part 1), and Single Wire CAN up to 33.3Kbaud.

The CAN protocol specifies the method by which data is passed between communicating devices on a CAN bus. It conforms to the ISO Open System Interconnection (OSI) model (ISO7498), which is a seven-layer description of a telecommunications network standard. The OSI model describes a layered system of communication between two network nodes, whereby in theory each layer can only communicate with the layers directly above and below it in the local node, and only with the equivalent layer in a remote node. In fact, the CAN protocol can be described by the lowest two layers of the OSI model – the Data Link Layer and the Physical Layer (layers 2 and 1 respectively). The Application Layer (layer 7) protocols can be proprietary schemes developed by individual CAN users or one of the emerging standards used within particular industries. In the automotive industry most manufacturers use their own proprietary standard.

CAN was originally designed for the automotive industry, but it has many applications elsewhere. Amongst the first applications were in industrial automation type applications using the CAN higher layer protocols DeviceNet and CANopen. CANopen has been applied in more varied applications than DeviceNet. CANopen has specialist profiles for industries such as lifts, marine controls and railways controls (Pfeiffer et al, 2003). This has now allowed it to reach the aerospace industry since CANopen is used in the Airbus A380 for lift control between three floors of the aircraft and the runway (CAN Newsletter 3/2006_1). DeviceNet is mostly limited to industrial automation applications such as those used in manufacturing lines.

SAE J1939 is the CAN higher layer protocol for the bus, truck and off-highway industries. The passenger car market tends to use proprietary higher layer protocols within vehicle manufacturers with virtually no standardisation. However, this was not possible with the truck industry. Truck cabs are required to connect to a variety of trailers from different manufacturers and therefore standardisation was required. SAE J1939 was specified for this purpose and the result is that certain CAN messages have a specific purpose, e.g. there is a CAN message for engine management information.

CANaerospace is a protocol which was designed for the highly reliable communication of microcomputer-based systems in airborne applications via CAN. The purpose of the protocol is for applications requiring an efficient data flow and easy time-frame synchronisation within redundant systems. The definition is kept widely open to allow implementation of user-defined message types and protocols. It has been applied commercially in applications such as the Airbus A380 and the Eurofighter.

NMEA2000 is a protocol based upon the SAE J1939 standard from the National Marine Electronics Association (http://www.nmea.org) to interconnect various electronic units onboard ships and smaller recreational and commercial vessels. The development of this standard began in 1994 and it was released in 2001. The standard currently operates at 250 Kbit/s.

MILCAN is a specification for a CAN based protocol for military applications and borrows parts from CANopen and SAE J1939. MILCAN has been used commercially by BAe Systems for the communication in an air-transportable armoured combat vehicle Terrier for the British Royal Engineers (CAN Newsletter 3/2006_2). MILCAN is used for operations such as drive-by-wire using a dual redundant bus and also a variety of functions such as power management.

2.2.2. Local Interconnect Network (LIN)

The LIN consortium was set up in 1999 to develop a new low cost bus for intelligent sensor and actuator applications not requiring the sophistication of CAN, referred to as the Local Interconnect Network. Core members include Freescale, BMW, Volvo and DaimlerChrysler. The LIN specification is currently at revision v2.1.

Many body control functions are often simple digital on/off operations, such as activating lights, wipers and windows. These are considered soft requirement real time systems that do not necessarily need the hard real time response that can be provided by CAN. Therefore a lower performance, more economical technology can be used.

The LIN protocol development commenced in 1999 and was introduced in 2000 (Wense; 2000). It provides an open market alternative to the proposed TTP/A protocol (Kopetz; 1995) for low cost sub-bus systems to complement CAN in applications such as vehicle roof (rain sensor, light sensor, light control, sun roof), vehicle doors (mirror, central locking, mirror switch, window lift), engine (sensors, small motors, steering wheel, cruise control switches, wiper, turn signal, radio, climate control) and seat (seat position motors, seat heater, occupancy sensor).

The main features of LIN are that it is UART based and uses a physical layer based on ISO-9141. It is limited to 20Kbaud and allows transfer of up to 8 bytes of information at a time. The application of LIN gives rise to the type of vehicle body control network architecture shown in Figure 12, which shows an example LIN sub-bus for control of a car door electrical sub-system. It can be seen that the main body control CAN network interfaces with the rest of the door system via a LIN-CAN Bus Gateway Unit. A single LIN signal line is used to connect the LIN-CAN Bus Gateway Unit to the Smart Door Lock Unit, Smart Mirror Motor Unit and Smart Window Motor Unit.

A deviation from the LIN consortium's main LIN protocol is SAE J2602, which is a variant of LIN 2.0. SAE J2602 is fixed to 10.4 Kbit/s to bring it in line with the legacy SAE J1850 protocol used by US automotive manufacturers.

The aim of the LIN 2.0/2.1 protocol is to provide the ability to purchase Commercial Off-The-Shelf (COTS) components, such as intelligent sensor and actuators, which are easily and rapidly integrated into an ECU with the use of LIN. Revision 2.0/2.1 provides diagnostic and plug/play features that are not in versions 1.2/1.3.

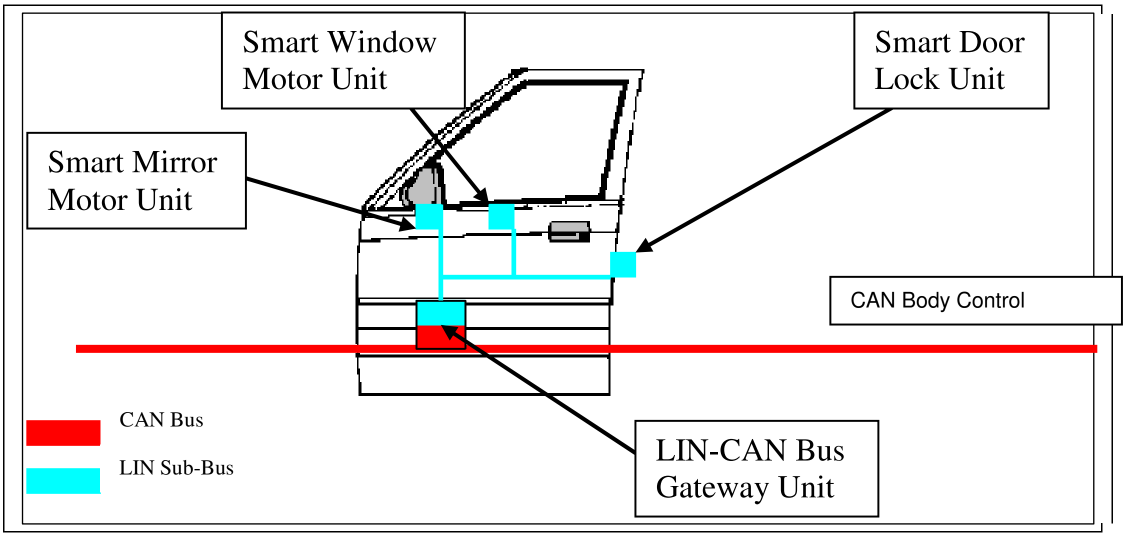


Figure 12 : A typical use of a combination of the LIN sub-bus along with the CAN bus in the localised area of car door control.

2.2.3. Emergence of High Speed Time Triggered Automotive Control Networks

CAN and LIN technologies enjoy widespread adoption in most vehicles that are currently manufactured. However there has been the emergence of a number of network technologies to address the needs of future high speed automotive control networking which tend to be generically referred to as “time triggered”. In general these requirements are to deliver higher bandwidth, determinism and failsafe features. Higher bandwidth provides the ability to transfer more data across the network and therefore a number of lower speed networks such as CAN have the possibility to be replaced by a single higher speed network. Determinism is achieved by rigid scheduling techniques which are generically referred to as “time triggered” in the automotive industry and provide the ability to precisely know when a message is late and synchronise applications over the network. Failsafe features provide support such as backup network channels to provide alternative data routes and bus guardians to prevent babbling idiot failures. Over recent years there have been five main technologies that have emerged as the main contenders for applications beyond the capabilities of CAN, although only one of them currently looks likely to become the de-facto standard. These are Time Triggered CAN (TTCAN), TTP/C, Byteflight and FlexRay. The main protocol features are compared in Table 3.

	Bandwidth	**Scheduling**	**Failsafe Features**
TTCAN	1Mbit/s	TDMA	None
TTP/C	25Mbit/s Unlimited (in theory)	TDMA	Dual channel Bus Guardian
Byteflight	10Mbit/s	Mini-slotting	None
FlexRay	10Mbit/s	TDMA and mini-slotting	Dual channel Bus Guardian

Table 3: Top level comparison of automotive time triggered protocol candidates

The main feature that is common with the emerging network technologies is that their scheduling mechanism is known as “time triggered”. There are two main time triggered scheduling methodologies that are used; Time Division Media Access (TDMA) and mini-slotting.

TDMA is a scheduling method in which a specific window of time (sometimes referred to as a slot) is reserved exclusively for the transmission of one particular message from a particular ECU. The communication schedule is usually represented as a time triggered matrix with a number of columns representing each of the windows and a number of rows representing each cycle of the matrix as shown in Figure 13. In this example, the order of transmission starts at Cycle0, Window0 and moves across each row from left to right. When the end of a cycle is reached, the communication moves down to the next row. When all rows have been executed, the schedule commences again back at Cycle0 and Window0.

Cycle 0	Window0	Window1	Window2	Window3
Cycle 1	Window0	Window1	Window2	Window3
Cycle 2	Window0	Window1	Window2	Window3
Cycle 3	Window0	Window1	Window2	Window3

Figure 13 : Example TDMA time triggered matrix

It should be noted that LIN can be considered in one way as a time triggered network because its schedule of communication is effectively a single row of a TDMA based communication matrix.

Mini-slotting is slightly different procedure in which each message that can be transmitted is exclusively assigned to a mini-slot. A mini-slot is essentially a waiting time that is used to schedule messages. The mini-slot is much shorter in time than would be required to transmit a full message. The highest priority message is given the first mini-slot, the next the second mini-slot and so on. The schedule works by waiting for each mini-slot to elapse and then moving onto the next in turn. During a particular mini-slot can decide to transmit the associated message or not. This way a dynamic schedule is the result by a kind of waiting time based arbitration method.

The fourth part of the CAN standard ISO11898 is an extension to specify TTCAN which was to address the safety critical needs of first generation drive-by-wire systems. The main characteristic of TTCAN is that bus access is controlled via a TDMA like method using a regularly repeating cycle of time called the *Basic Cycle*. The *Basic Cycle* is divided into a fixed number of time windows (i.e. fixed at design time) which can be a mixture of any one of four types: *Reference Message*, *Exclusive Window*, *Arbitration Window* and *Free Window*. The *Reference Message* signifies the start of the *Basic Cycle*. CAN communication is initiated by the *Reference Message*. The *Exclusive Window* is reserved for one particular CAN message only, which can only be sent by one particular node. In an *Arbitration Window* a number of nodes may attempt to transmit a message. Therefore the nodes that may contend for bus access during the *Arbitration Window* may do so by the usual non-destructive bitwise arbitration method of the CAN protocol. With normal CAN systems, nodes losing arbitration will attempt to retransmit the message that lost in the arbitration process. This feature is disabled in TTCAN since a retransmission would upset the remainder of the operation of the *Basic Cycle*. Most modern CAN controllers available commercially have the possibility to enable or disable the retransmission of a message after losing arbitration. This means that TTCAN implementations are possible in future vehicles, especially as private TTCAN buses within specific sub-systems that do not need to be interoperable with ECUs from other suppliers. However there are no known commercial applications of TTCAN that have been published. A *Free Window* is reserved for future expansion of the TTCAN system. A methodology for the implementation of TTCAN that effectively doubled the usable bandwidth of CAN was described in one study (Pope et al; 2005) and is a key benefit from the use of the technology.

TTP/C is a safety driven protocol originally developed at the University of Vienna by the research group of Professor Herman Kopetz and now commercially exploited by company TTTech (Kopetz et al; 1998). It was an early candidate for the new generation of automotive networks to complement or replace CAN in safety-based systems.

It uses a TDMA bus access scheme and the communication controllers currently available support 25 Mbit/s synchronous (using Ethernet-like wiring) and 5 Mbit/s asynchronous (using twisted pair wiring) transmission. Data frames can carry a payload of up to 240 bytes each.

There are currently no automotive production examples of TTP/C, however there have been some control system applications in production in other industries. Honeywell uses TTP for General Electric's F110 engine control system employed on the Lockheed Martin F-16 fighter aircraft. In addition, TTP will be incorporated on the Honeywell F124 digital engine control system to be used by the M-346 fighter trainer aircraft made by Italian aircraft manufacturer Aermacchi. Honeywell's APEX integrated cockpit using TTP has been selected as the standard avionics package for the GROB Ranger G 160 single-engine turboprop business aircraft, the EXTRA EA-500 single-turboprop business aircraft, the IBIS Ae270 single-engine turboprop aircraft, and several other aeroplanes. Nord-Micro has selected TTP as communication protocol for the Airbus A380 cabin pressure control system. Hamilton Sundstrand Corporation has selected a TTP-based data communication platform for use in electric and environmental control systems on the Boeing 787 Dreamliner. Alcatel uses TTP as field bus protocol for their railway signalling system ELEKTRA 2. Commercial production started in June 2002.

Due to the rigidity of the TTP/C protocol and the requirement by the automotive industry for some flexible message arbitration facility, the TTP/C protocol has to date never reached automotive production. Instead the car manufacturer BMW commenced work on the open Byteflight protocol which was finally implemented in the 2001 BMW 7 series (Byteflight Specification; Berwanger J et al; 2000). It runs at 10Mbit/s over fibre optic cables using a mini-slotting message scheduling approach. Byteflight was only implemented on the BMW 7 Series. Subsequent designs at BMW requiring a fast deterministic protocol have been set for the emerging FlexRay protocol.

FlexRay technology has been developed since 1999 as an alternative to TTP/C technology. The main reason, as its name suggests, is the requirement for more flexibility than other candidate technologies. The FlexRay consortium was launched in 2000 to develop the FlexRay specifications and market. It consisted of core members BMW, DaimlerChrysler, Freescale (previously called Motorola), NXP (formally Philips Semiconductors), Robert Bosch and Elecktrobit. The first draft of the FlexRay protocol was introduced in 2001 (Berwanger at al, 2001). The aim of FlexRay is to complement CAN in higher bandwidth and integrity applications and is now at revision 2.1. This particular revision was included in the 2006 model year BMW X5 chassis control system for a five node FlexRay network for adaptive drive (Berwanger et al, 2005). It has also been used in the latest BMW 5- and 7-series vehicles. The new Audi A8 will adopt some FlexRay and one Bentley model that is electrically based on the A8.

FlexRay has established a significant advance on CAN technology by increasing bit rate, providing synchronisation between nodes so that a time triggered bus access methodology is achieved and providing a backup data channel for dual mode redundancy. At the end of 2005, the FlexRay consortium announced the development of “FlexRay II” as a way of reducing the complexity and cost of a FlexRay implementation. This differs to standard FlexRay by only having one channel and only using a single Master for time synchronisation. There is potential that this cost-reduced version could replace CAN on further automotive applications. To date there is no further news of this development. FlexRay is currently only adopted in a few high-end vehicles that were previously stated. The world-wide economic problems of 2007 to 2009 have possibly slowed the adoption of FlexRay with many companies focusing on their near term vehicle projects using CAN and LIN.

2.2.4. Other Emerging Automotive Control Networks

It has been described how modern vehicles now use a blend of CAN, LIN and even FlexRay on some of the higher-end vehicles for applications such as powertrain, chassis and body control. MOST is the de-facto standard for infotainment systems. However there are a number of protocols that are starting to become applied to automotive control in the areas of sensor networks, diagnostics and high integrity networking.

Ethernet is a protocol standardised as IEEE 802.3 that has been used in office, consumer and industrial automation for many years. Now there is a lot of interest in adopting Ethernet in the automotive industry. The first area that needs Ethernet is in the area of garage diagnostics. Modern vehicles have many ECUs that need re-flashing with software updates during the regular service intervals. The current standard diagnostic connection is via the SAE J1962 connector and CAN which has a maximum bit rate of 1 Mbit/s. However each ECU flashed via CAN could take anything from one to several minutes to re-flash. If many ECUs are required to be re-flashed this can add up to a considerable amount time therefore there is a requirement for a faster network to reduce these times. There is now automotive specification Ethernet silicon available for such applications (Jones, 2009).

One of the recent innovations in time triggered technology is that of TT-Ethernet (Kopetz et al, 2005) that is aimed at high speed safety critical networking and has the potential to do the same job as FlexRay in automotive control systems. It has a maximum data rate of 1Gbit/s. It is currently being pushed across all real-time applications.

The SENT protocol (Single Edge Nibble Transmission) is a low cost protocol that is standardised under SAE J2716 aimed at communications between sensors and ECUs. The protocol has been primarily driven by the semiconductor manufacturer Infineon Technologies AG. One of the motivations for the development of SENT is to provide a technology that is cheaper than LIN for single fast point to point communications between digital and analogue sensors and an ECU. Therefore the network traffic is one way from sensor to ECU. The SENT protocol currently tends to be implemented in software. In a host ECU based on the Infineon Tricore microcontroller the protocol requires at least 10 Kbytes ROM and 7 Kbytes RAM. An extension of the SENT protocol is the Short PWM Code (SPC) protocol. SPC is aimed at increasing the performance of the communication link, reducing system costs and also allowing bidirectional communication (Beaurenaut, 2009).

The Safe-by-Wire Plus consortium was formed in February 2004 to create a single and open global standard of an automotive safety bus specifically to be used for occupant safety applications only (e.g. airbag deployment and seat belt restraint). The Safe-by-Wire Plus consortium released version 2.0 of the Automotive Safety Restraints Bus specification (ASRB 2.0) in 2004 and was based on the existing Safe-by-Wire ASRB 1.0 specification, while incorporating some new concepts. The consortium also plans to submit the ASRB 2.0 specification to the appropriate ISO working group for consideration as a global standard. The consortium consists of leading automotive systems and component suppliers, including Analog Devices Inc, Autoliv, Delphi Corporation, Key Safety Systems, Philips, Special Devices, TRW Automotive, Bosch, Siemens VDO Automotive and Continental Temic. Safe-by-

Wire Plus has variable bus speeds of 20, 40, 80 or 160 kbps and has a similar nodal cost comparable to CAN. The application of the Safe-by-Wire protocol is narrow and therefore is not suitable for general network service. As a result, it will never replace general purpose automotive networks such as CAN (Boys, 2004).

Recently wireless networks have begun to be adopted for both inter- and intra-vehicle communications. Inter-vehicle applications are concerned with telematic applications. Intra-vehicle applications are potential competitor technologies for wired or fibre optic based network technologies. Some intra-vehicle wireless applications are for sensor networks that cannot feasibly be wired such as Tyre Pressure Monitoring (TPS) in which wireless sensors are installed in the car wheels and transmit tyre pressure information wirelessly to a central ECU. One of the key benefits of Wireless Automotive Sensor Networks (WASN) is that a wiring harness is highly complex, costly and heavy and the adoption of WASN has the potential to solve the problem. However the big argument against their widespread adoption is whether they can deliver the same level of performance, reliability and safety offered by wired networks (ElBatt et al, 2006).

2.2.5. Multimedia Automotive Networks

One other automotive network domain is in the infotainment area rather than control systems domain. The main requirement of this type of network is the ability to robustly transfer large amounts of information.

Media Oriented Systems Transport (MOST) is a fast ring network which is now the de-facto standard for multimedia applications such as the integration of video, navigation and consumer electronics (e.g. mobile phone) equipments into the vehicle. It tends to be used in higher end vehicles such as BMW, Jaguar, Aston

Martin, Audi etc. The first version of this technology was referred to as MOST25 which provides 23Mbaud bandwidth for data streaming over optical fibre. Later the technology's performance was increased with the introduction of MOST50. MOST50 doubles the bandwidth available from MOST25 and specifies support for both optical and electrical physical layers. MOST50 applications in practice tend to use unshielded twisted pair electrical physical layers. MOST150 was introduced in 2007 and increased upon the functionality and performance of the other versions of MOST by integrating an Ethernet channel.

MOST is the de-facto standard for automotive infotainment systems. However there are alternatives. D2B (Domestic Digital Bus) is a 12 Mbit/s which is run over optic fibres and used in some Mercedes vehicles for the audio systems. IDB-1394 is an international data networking standard for transmitting video, audio and other multimedia data over an in-vehicle network. The IDB-1394 specification is a joint initiative of the 1394 Trade Association and the IDB Forum to develop an in-vehicle network designed for high-speed multimedia applications. Previously known as IDB-M, IDB-1394 is built on the IEEE-1394 technology, which has gained wide acceptance in consumer electronics. The system architecture allows existing IEEE-1394 consumer electronics devices to integrate with embedded automotive grade devices. The system consists of an automotive grade plastic optical fibre network and a Consumer Convenience Port (CCP) interfaces and the ability to attach plug and-play portable devices. Ethernet is another protocol suited to the infotainment type of application with BMW bringing an Ethernet-based video link to market for a park assist camera solution for the model year 2013 X5.

2.2.6. The Future of Automotive Control Networks

In the previous sections it has been established that modern vehicle electrical architectures use a combination of CAN, LIN and FlexRay. Different networks are used as each one has a different advantage and is more appropriate for certain applications. In the next few years the adoption of in-vehicle networking technology is likely to be faster than the previous ten years. Safe-by-Wire for airbag and seatbelt deployment is likely to grow in its use but more as a private network. Many higher end vehicles use the MOST protocol for infotainment systems, however there is the possibility that this could be replaced by Ethernet in the future. At the sensor integration level, the SENT protocol may enjoy some adoption but possibly as a point to point network rather than a bus such as LIN. It is faster than LIN but is more memory hungry in its current implementation and therefore its adoption will probably be limited to sensors needing a faster data transfer compared to LIN. There are a number of other developments concerning in-vehicle networking technology which are outlined in this section.

The Controller Area Network

The CAN protocol is evolving further to enhance its performance. In ISO11898 part 4 the TTCAN protocol was standardised (ISO11898:4-2004) but has seen no reported commercial uptake. One perceived drawback of time triggered protocols is that the message transmission schedule is fixed at design time and this is not flexible enough for some applications. There is a technological gap between CAN and the FlexRay protocols and the migration costs from CAN to FlexRay are significant. For these reasons the owners of the CAN protocol intellectual property, Robert Bosch GmbH, have recently initiated a protocol enhancement referred to as CAN with Flexible

Data-Rate (CAN-FD) (Robert Bosch GmbH, 2011). CAN-FD is based on the CAN protocol as specified in ISO 11898-1 and continues to use the CAN bus arbitration method. However it increases the effective bandwidth of the protocol by switching to a shorter bit time after the end of the arbitration field (with a bit rate of 8Mbit/s) and returns to the longer bit time at the CRC Delimiter, which is importantly before the CAN receivers send their acknowledge bits. In addition to the bit rate switching, the data field will be extended from 8 bytes to 64 bytes. The development of CAN-FD is aimed to be implemented as an additional mode to the standard 11898 CAN for applications such as switching to a fast software download during service or end of line. Since it will be implemented as an additional mode the costs are expected to be in the region of CAN since it will be integrated in the same silicon and use the same physical layer.

The introduction of the CAN-FD technology is expected to happen in one of two ways. In the first of these it is possible that the UDS diagnostic protocol is used to turn off the communications of all non-CAN-FD nodes and therefore remain in standby whilst a CAN-FD node is switched to the CAN-FD mode for fast software download. In the second of these an entire CAN-FD compliant network could be implemented. Although traditional CAN transceivers can be used, it is also possible that CAN-FD type transceivers may be used to provide signals to switch to the higher bit-rate. At the time of writing this protocol is in its infancy and therefore its standardisation process is subject to specification changes. Other activities to be carried out is to find OEMs for first applications, stimulate interest from semiconductor manufacturers to include the additional features in their products, get development tool suppliers to support CAN-FD, initiate ISO standardisation and ensure that full support is included in future releases of AUTOSAR. The net effect of the implementation of CAN-FD is that the effective bit rate is much higher than CAN as defined in ISO11898-1 (e.g. ~3Mbit/s). An example of this is shown in Figure 14.

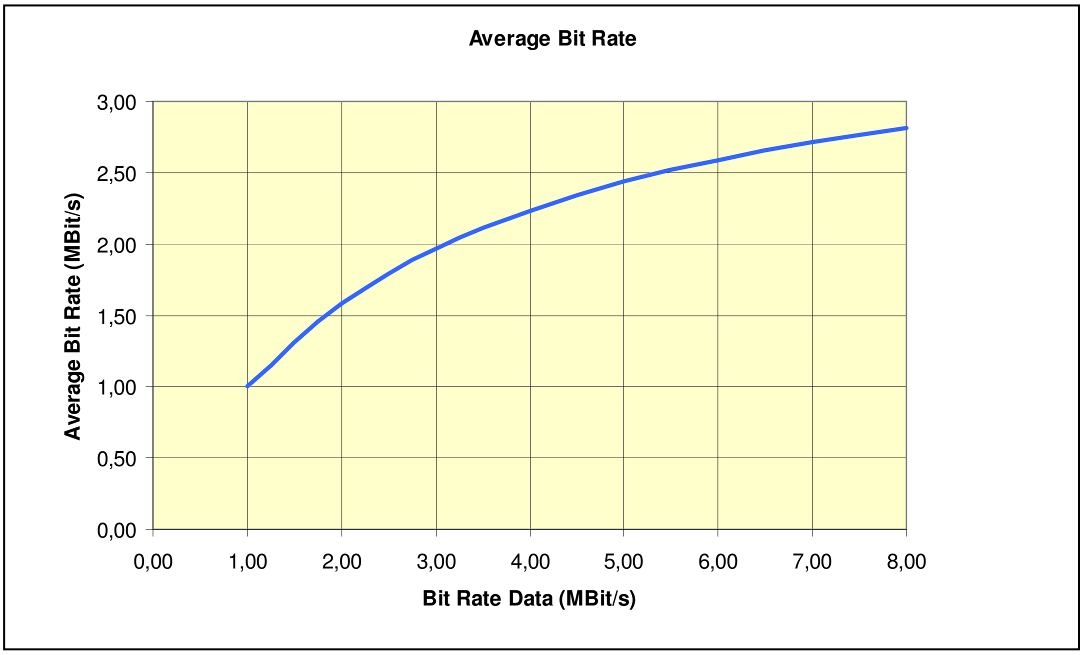


Figure 14 : A plot of bit rate of the data field versus the average bit rate across the whole CAN frame. CAN Identifier 11 Bit, Data Field 8 Byte, Bit Rate Arbitration. 1 MBit/s.

FlexRay

The FlexRay consortium at the time of writing has effectively ended its work with the release of protocol specification package v3.0.1 and the protocol is now undergoing ISO standardisation. The adoption of FlexRay technology has barely started with only a few vehicles from BMW and Audi reaching series production.

FlexRay v3.0.1 has a choice of architectures and synchronisation methods. In terms of the synchronisation methods there is TT-D (this has two or more coldstart nodes resulting in increased fault tolerance), TT-L (this has only one coldstart node which therefore has the benefit of reducing system complexity and reduced start up time) and TT-E (which allows synchronisation with another FlexRay network). In terms of FlexRay architectures the possibilities are from a single linear bus through to a cascaded star network with redundant signal paths and a bus guardian to prevent babbling idiot failure. The trend for the use of FlexRay in high-end automobiles was created by its deployment in series production by BMW and Audi. After the establishment of FlexRay in the small but very demanding sector of premium cars, its wider use in further automobile models is only a matter of time. There is no requirement to implement both channels A and B of the FlexRay protocol. The 2006 model year BMW X5 only implemented channel A for the very first commercial production application of FlexRay. Therefore there is the opportunity that in the future semiconductor providers might provide a "Lighter" version of FlexRay using one channel only reducing the cost of the silicon.

Ethernet in Vehicles

FlexRay is the established bus system for vehicles requiring bandwidth up to 10 MBit/s. Bandwidth requirements will, however, continue to rise. Driver assistance systems that include a camera theoretically require bandwidths of tens of MBit/s. This exceeds the capacity of FlexRay solutions. For higher requirements, Ethernet is a solution as it supports 100 MBit/s data transmission and higher.

By 2015, BMW will start using Ethernet in the area of driver assistance, where video cameras will be connected to a central control unit (Plankensteiner, 2011). In general, the typical applications for Ethernet are those where large amounts of data are processed. An example of this is a passive safety system such as Lane Departure Warning. A future step would be to use Ethernet in an active safety system such as Active Brake Assist.

TT-Ethernet brings together the fast bandwidth of Ethernet with the time triggered principles of FlexRay. Therefore TT-Ethernet could provide the determinism of FlexRay but with bit rates of at least 100Mbit/s. It would be ideal for applications such as camera and vehicle backbone applications. There are currently no publicised plans to adopt TT-Ethernet in the automotive industry however there are several research projects underway with the Austrian company promoting the technology, TTTech GmbH.

Mesh Networking and Smart Grids

Mesh networking is a type of networking where each node must not only capture and disseminate its own data, but also serve as a *relay* for other sensor nodes. Therefore it must collaborate to propagate the data in the network like a router. This approach tends to be used in wireless sensor networks. Zigbee is perhaps the most well known of the wireless mesh sensor networking technologies.

ZigBee is a high level communication protocol using small, low-power digital radios based on the IEEE 802.15.4-2003 standard for Low-Rate Wireless Personal Area Networks (LR-WPANs). Typical commercial applications include wireless light switches with lamps, electrical meters with in-home-displays, consumer electronics equipment via short-range radio needing low rates of data transfer. Recently (SAE 2011) the Society of Automotive Engineers (SAE) in the USA and the ZigBee Alliance have teamed up to make ZigBee Smart Energy the preferred technology for plug-in electric vehicles (PEVs) and enabling essential vehicle-to-grid communication and power capabilities. According to the SAE, ZigBee Smart Energy is the market-leading home area network and advanced metering infrastructure standard for the smart grid. The aim is to use ZigBee Smart Energy to define how PEVs and the grid interact, whether at the consumer's home or at a remote location. Ultimately, the initiative will provide future PEV drivers with the real-time information needed to control their transportation energy use. It will also help them manage their charging costs and receive utility incentives for participating in PEV programs. Adding ZigBee Smart Energy to PEVs will give car manufacturers and utility companies a common language to manage the charging, storage, and use of energy in PEVs. Work between the groups is under way, with completion targeted for next year when ZigBee Smart Energy version 2.0 is scheduled for completion in 2012. SAE joins ZigBee Smart Energy development efforts led by some of the largest utilities, suppliers, and technology companies in the world.

A possible application of this technology is as a Vehicular Ad-Hoc Network (VANET). This is a wireless mesh technology that uses moving cars as nodes in a network to create a mobile network. VANET turns every participating car into a wireless router or node, allowing cars approximately 100 to 300 metres apart to connect and by mesh networking create a network with a wide range.

The use of a common and open standard such as ZigBee for communicating energy management information can be used to optimise the control of hybrid and electric powertrains by estimating journey parameters using techniques introduced in previous studies (Quigley, 2011). Such journey parameters could include journey energy requirement, location of origin and destination, energy that can be recovered using regenerative braking, distance and duration. This information can be used to optimise the hybrid or electric powertrain so that electrical power sources are used in preference to fossil fuel powered sources and also used to provide information to utility companies to allow them to provide energy supplied at locations required, ideally using renewable energy sources.

Figure 15 shows a possible future in-vehicle network architecture comprising of multiple networks and network technologies. In this projection of a future architecture there is much use of current technologies such as CAN, LIN and

FlexRay. However there are many new technologies. Ethernet is used in many areas such as for camera communications, replacing MOST in infotainment and as a backbone technology. CAN could remain strong due to its extension of capabilities and speed with the future development of CAN-FD. Also FlexRay is very complicated when compared to CAN and therefore this may hold back the widespread adoption of FlexRay. SENT may be part of a private network such as sensors to the EMS. Safe-by-Wire may be the de-facto standard for a private airbag and seat restraint system. Zigbee may be used for vehicle to grid communications.

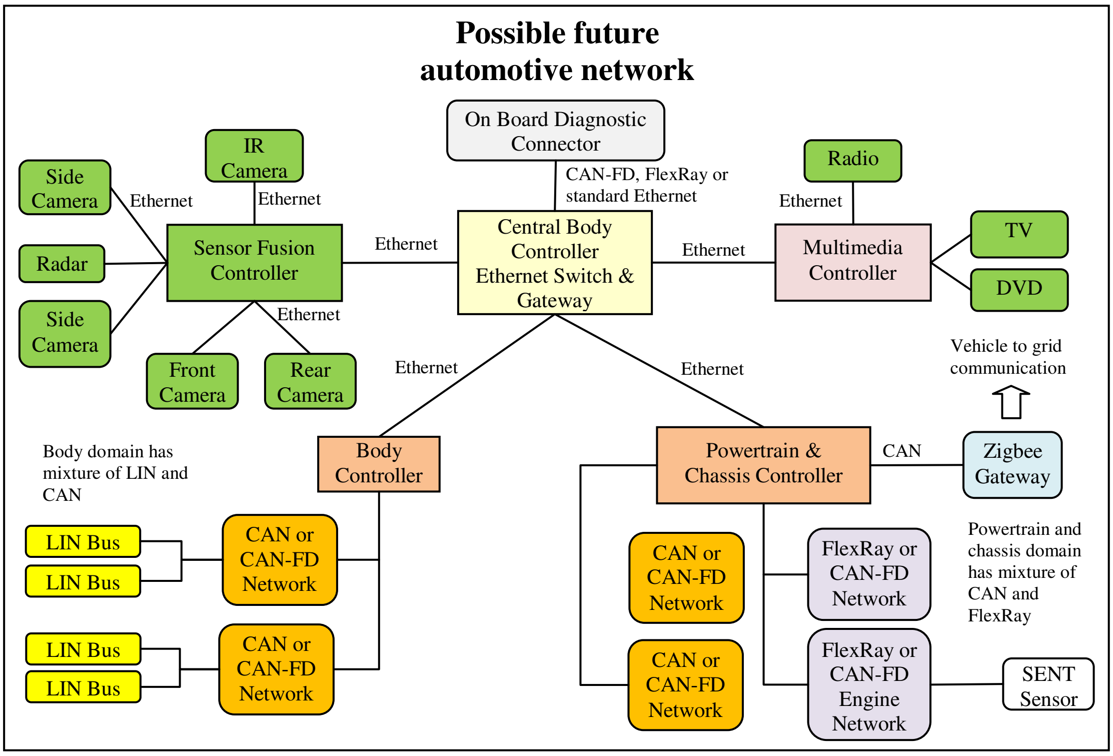

Possible future automotive network
On Board Diagnostic Connector
CAN-FD, FlexRay or standard Ethernet
IR Camera
Ethernet
Side Camera
Ethernet
Radar
Side Camera
Sensor Fusion Controller
Ethernet
Front Camera
Rear Camera
Ethernet
Central Body Controller Ethernet Switch & Gateway
Ethernet
Radio
Ethernet
Multimedia Controller
TV
DVD
Ethernet
Ethernet
Vehicle to grid communication
Body domain has mixture of LIN and CAN
Body Controller
Powertrain & Chassis Controller
CAN
Zigbee Gateway
LIN Bus
LIN Bus
CAN or CAN-FD Network
LIN Bus
LIN Bus
CAN or CAN-FD Network
CAN or CAN-FD Network
CAN or CAN-FD Network
FlexRay or CAN-FD Network
FlexRay or CAN-FD Engine Network
Powertrain and chassis domain has mixture of CAN and FlexRay
SENT Sensor

Figure 15 : Possible in-vehicle network architecture of the future comprising multiple networks and network technologies (Adapted from Plankensteiner, 2011).

Due to the increasing bandwidth requirements for On Board Diagnostics (OBD), particularly for ECU flashing, CAN-FD, FlexRay or Ethernet are the most likely candidates. Wireless connections could be candidates but the integrity of the car would be perceived to be at risk due to hacking. CAN-FD has an advantage that there would not be as much a change as FlexRay or Ethernet. FlexRay is most likely to become the high speed integrity network but it is a difficult technology to deal with and maybe due to this and the implied costs the CAN-FD protocol could challenge the technology in the powertrain and chassis domain.

2.3. Adoption of Control Networks in Other Vehicle Applications

The applications discussed in the previous sections are mainly concerned and driven by the mainstream passenger vehicle industry. However, there are other ground-based vehicle applications that follow the electronic system technology trend of the passenger vehicle industry to some degree.

The truck and bus industry uses all classes of automotive electronic control system. However, this industry is very different from the mainstream automotive industry in that its adoption of electronics control applications occurs over a much longer timeframe. Its applications mainly involve engine control and body control. In the case of the truck industry, the electronic control is involved with the integration of systems between truck and trailer. The challenge for designers is that a truck (or cab) may be connected to numerous different types of trailer during its lifecycle. Typical applications include diesel engine management, body control and distribution of telematic information. Companies producing trucks and buses include Leyland Trucks, Paccar, Alexander Dennis, Volvo and Mercedes Benz. A study by Axelsson et al (Axelsson et al, 2003) compared in-vehicle network integration technologies between the different products under the Volvo brand. It was noted that there is potential for the same in-vehicle networking technologies that are used in Volvo Cars to also be used in Volvo Trucks.

Off-highway is involved with agricultural and construction equipment. Its timeline for the adoption of vehicle electronics is similar to the truck and bus industry and significantly lags the automotive passenger vehicle market. As with the truck and bus sector, a study by Axelsson et al (Axelsson et al, 2003) compared in-vehicle network integration between the different products under the Volvo brand. It was noted that there is potential for the same in-vehicle networking technologies that are used in Volvo Cars to also be used in Volvo Construction, with the exception of infotainment systems.

Motorcycles and scooters have typically had very simple electrical systems. However, the industry does tend to follow the passenger car industry in its electronics mainly because of common suppliers, but at a very delayed pace. Most motorcycles currently only have sophisticated electronic ignition systems; many now have simple engine management systems and ABS is also becoming common. The rarer and more advanced examples implement systems such as airbags for safety, navigation systems, drive-by-wire and even electric / hybrid powertrain control. Audino et al 2007 describe the trends and adoption of electronic systems in modern motorcycles and scooters.

The marine industry does share some technology with the land based vehicles in the engine bay. Some of the suppliers also provide engines to off-highway/bus/truck type applications. Controls in the marine industry are concerned with engine and directional control, but also the sharing of navigation information. Typical systems include radar, engine information, speed transducer.

2.4. Processes Used for Designing the Electrical Architecture of Distributed Automotive Control Systems

The traditional V-model is the design model used in the development of a new vehicle and its components, including the vehicle's electronics. This approach to automotive electronic systems development is based on the Waterfall model (Royce, 1970) and is also the model which is used for the entire vehicle design process. However it differs from the pure Waterfall model since the left side of the "V" is concerned with requirements captures and design, whilst the right side of the "V" is concerned with testing and validation. Each phase of the left side directly maps onto a phase on the right side. The V model often differs between different companies and their requirements.

The process is used for the development of mechanical components and systems, and also electrical systems. There are a number of sub-processes that are used within this. Within submission one those relevant to distributed control system electrical architecture design were discussed. A number of areas were highlighted as areas needing further development. This included how to map time triggered network messages to CAN and the emerging area of Object Oriented Design which is most evident in the data exchange formats provided in the AUTOSAR and ASAM working groups. However the main areas of commercial interest concerning the NetGen product and directly related to network architecture design are discussed further in this chapter.

2.4.1. Design of Network Schedules and Dealing with the Multi-Supplier Problem

The design of in-vehicle networked based systems has evolved over the last twenty years. Early protocols such as SAE J1850, Seriplex, VAN and CAN were developed under a process in which the message transmission from ECUs was designed with little regard to the underlying bus technology characteristics. For example in the case of CAN implementations, the protocol itself was relied upon to schedule the message transmission effectively. For most of the 1990s, this type of process was followed by most vehicle manufacturers for the development of CAN-based systems. It is still followed to some degree by many for CAN based systems even today. However, it created a number of problems during systems integration such as *Non-Deterministic* message delays (especially lower priority ones) and bus loading limitations. It is a well known feature of CSMA-CD (Carrier Sense, Multiple Access, Collision Detection) type networks, such as CAN, that they do not have guaranteed timing properties and only operate acceptably up to about 40% bus loading. An excursion above this bus load tends to lead to great variability in the latency for lower priority messages and such messages can be starved of access to the network. One solution adopted by many vehicle manufacturers is to partition the system over two or more CAN buses, using one for high speed applications such as powertrain or chassis and another for lower speed applications such as body control. This approach has led to some vehicles having up to six CAN buses. This resulted in greater cost through additional wiring, connectors, gateway ECUs and design effort for system partitioning.

Another solution to the bus loading limitation and message latency problems of the *Non-deterministic Approach* was developed in Volvo (Tindell, 1994). This solution could determine the worst case message latency by assigning higher priority (or periodicity) signals to higher priority CAN messages. Lower priority signals (or periodicity signals, such as those that are event triggered) are assigned to lower priority CAN messages. The result of this approach was that the CAN bus was able to run at a higher loading, with the worst case latency of CAN messages known at design time and therefore design tradeoffs were able to be made to ensure acceptable latency. As it maximises the use of the CAN bandwidth it can be a way of reducing the cost of the bill of materials of the CAN system.

This is a *Pseudo-deterministic Approach* since the determinism is defined as a worst case latency and is the process that has been implemented in the Volcano tools (Rajnak and Ramnefors; 2002). The key principles of this CAN design process are based on a publisher-subscriber model (Navet et al; 2005). This methodology improves the procedure of traditional CAN bus development process using a *Non-deterministic Approach*, since it allows the OEM to deal much better with multiple system suppliers. Introduction of new ECUs late in development or as an upgrade is straightforward according to Rajnak and Ramnefors. The only change necessary is for the communication configuration data to be re-flashed with the new network configuration. This therefore helps to reduce the cost of the ECU integration process but does often require high priced design tools.

The *Pseudo-deterministic Approach* for CAN and work by Kopetz on the Time Triggered Protocol (TTP) (Kopetz and Thurner, 1998) has led the way to the improved *Deterministic Approach* to design for time triggered protocols. Time triggered protocols use scheduling methodologies such as the aforementioned TDMA and mini-slotting approaches. Time triggered protocols are deterministic in their nature and their associated design process deals with the system integration issues much better as they allow the designer to easily conceptualise the communications timing and map signals and messages across different buses. They are also good for dealing with the problem of multiple suppliers in the same way as the *Pseudo-deterministic Approach*.

When the systems from the suppliers and the OEMs are integrated together, the final result is the cycle of the communications matrix with no collisions or surprises caused by latency problems. The *Deterministic Approach* used in time triggered protocols result in the potential for high bus utilisation in a similar way as does the *Pseudo-Deterministic Approach* therefore potentially reducing the number of networks required.

One problem that was highlighted in submission one is that all three of these approaches are currently being used within the automotive industry simultaneously. Modern vehicles control systems have architectures which are a combination of CAN, LIN and FlexRay and therefore to seamlessly deal with the mapping of data across these networks is a process that is not currently clear and could be the subject of research. However due to commercial considerations, research in this area was not considered as the highest priority.

2.4.2. Designing an Automotive Electrical Architecture to a Target Cost

Automotive electrical architecture design is a "black-art" that is carried out with consideration of many different factors such as wiring harness cost, weight, reliability, safety, EMC and complexity. One of the challenges of the architecture designer is how to partition the system so that it satisfies the functional design requirements but at the same time pays considerations to each of the design factors. Therefore it is a significant optimisation problem. A larger vehicle manufacturer often can dictate to its suppliers about the design of the systems they provide and therefore the potential for optimisation is great. Smaller vehicle manufacturers tend not to have this relationship with their suppliers and will usually have to take systems off-the-shelf, only with scope for minor tailoring. Therefore the potential for optimisation is much less.

A large amount of product design requires that a component or system be designed to a target. Design for assembly and manufacture is a process, as its name suggests, focuses on ensuring that a component or system can be manufactured and assembled easily. Design to cost is a generic term that describes a process that engineers participate in to ensure that the component or system that they are designing meets a cost target.

The wiring harness is one of the most expensive components in a modern vehicle after the engine. In-vehicle networking technology was originally adopted to reduce weight and cost, and to increase electrical system reliability. Weight and cost were reduced by the removal of duplicate sensors and reduction of the number of wires and connectors. The design to cost process of the electrical architecture is one that is unclear due to the system complexity. This is also something that has been covered very little in published literature, possibly due to the commercially confidential nature of the subject. It is also a process that tends to mean a different thing to different commercial organisations.

One study (McLaughlin, 1993) investigated the advantages of using CAN as an alternative to hardwired ECU integration. Numerous advantages were found, including the potential to reduce the cost of the vehicle electrical architecture. The potential for cost reduction was found by developing a costing model based on interviews with a number of engineers involved with this costing process in a UK based vehicle manufacturer. This model was comprised of two equations and showed that there were potential savings from the adoption of CAN when compared to hard wired integration. The first equation considered the additional costs required to integrate CAN into existing ECUs on the vehicle, based on the bill of materials and labour involved. The second equation was for the cost reduction from the removal of copper signal wires which were originally hardwired but now connected by the CAN bus. The combination of both equations gave the net saving to be gained from the implementation of CAN within the electrical architecture of a typical luxury vehicle manufactured in the early 1990s.

The implementation of FlexRay for chassis control in a production BMW was described by Berwanger et al in 2005. Although the nodal costs of FlexRay are higher, an interesting reason for choosing FlexRay stated by Berwanger et al is that it can reduce system costs when compared to a CAN implementation. This is counter-intuitive when nodal costing is considered. However, it is argued that cost savings can be enjoyed by replacing several CAN buses. This in turn reduces wiring, connectors, gateway ECUs and all of this reduces system partitioning effort. In a presentation at the FlexRay Product Day in December 2005, Berwanger et al stated that reducing the number of CAN sub-buses, cables and redundant sensors meant that, holistically, the implementation of FlexRay is roughly the same as for CAN. They also estimate that integration of the FlexRay controller into the microcontroller and using a lower cost transceiver will save approximately three and one Euros respectively per ECU. However, no real data has been presented to support these arguments and also it is not clear whether this considers the impact of adopting a new and sophisticated technology such as FlexRay.

Of all the published literature, all but one failed to provide the ability to quickly assess candidate architectures on simple amounts of information. The McLaughlin study differed from this and came up with coarse models for assessment of IVN systems (CAN in this case). However these were not validated and failed to assess the effect of the embedded software on the architecture cost.

2.5. Which Network Technology is the Most Cost Effective?

Figure 16 shows a comparison between the relative nodal costs of protocol implementation versus its bandwidth for the main automotive relevant protocols. This figure is based on Figure 1 of LIN specification package Revision 1.2 page 2, but has been updated based on other information already presented in this report to include the current and emerging protocols in the automotive industry in terms of network technology costs. Since CAN is the de-facto automotive standard, CAN is used as the relative cost of unity and is therefore the reference protocol in the figure.

A LIN node is expected to be cheaper to implement than CAN as a result of a number of cost saving features of the protocol such as the use of a single wire for the transmission of data (instead of the twisted pair used for CAN), the ability that LIN slave devices can use cheaper RC oscillators (instead of crystal oscillators that are a necessity for all interoperable CAN devices) and the physical layer is simpler and therefore cheaper in its standard form. The SENT protocol is a unidirectional point to point technology that is slightly faster than LIN (28 KHz) but also reported to be cheaper to implement.

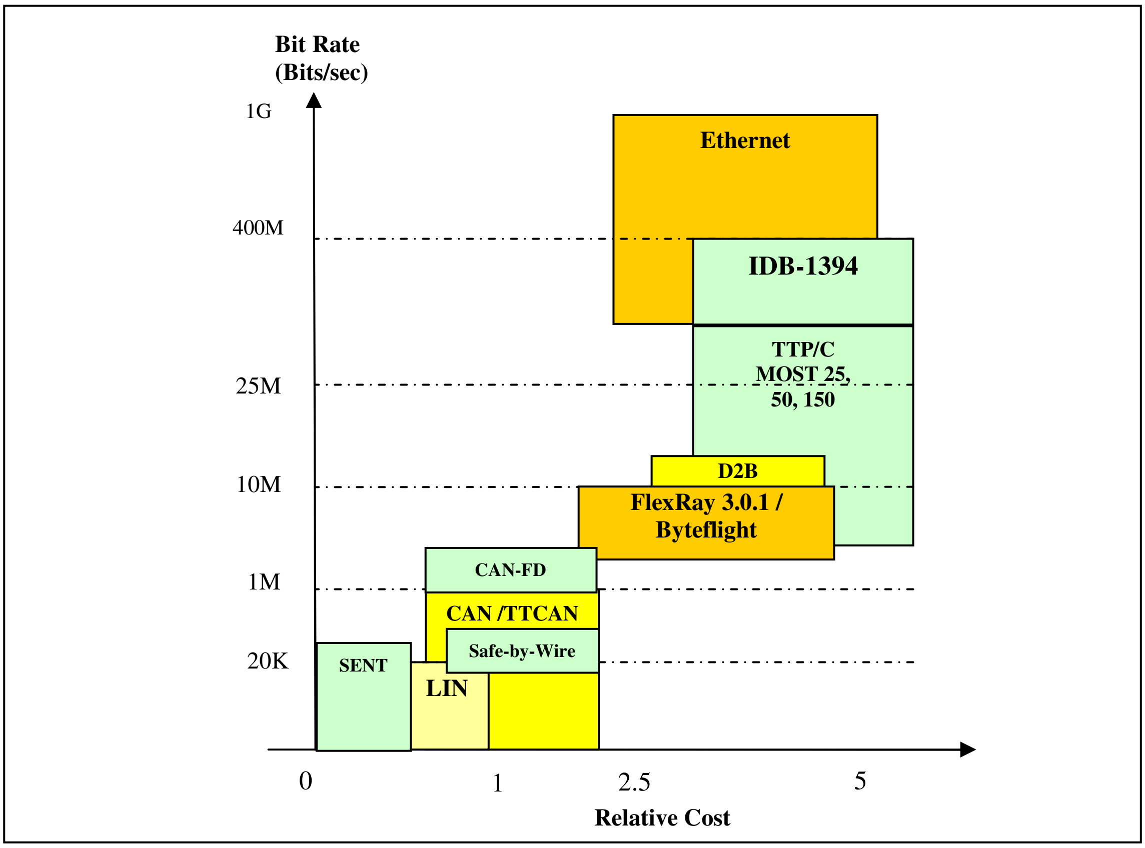


Figure 16: Current and emerging communications technologies versus price versus bandwidth (based on Figure 1 of LIN specification package Revision 1.2 page 2)

TTCAN has been assumed to be similar pricing to CAN since it uses the same silicon. FlexRay in its current revision v2.1, is expected to be significantly more expensive than CAN, mainly due to a greater RAM requirement in the FlexRay controller. There is a spread in the cost domain for FlexRay due to the variety of network architectures that could be used, for example bus and star configurations are possible. Some configurations are lower cost, whilst others are of higher cost. There is contradictory information in the literature concerning FlexRay being a higher cost protocol. In the year 2006, the BMW X5 became the first production car to implement FlexRay within its chassis control system. Although the nodal costs of FlexRay are higher, an interesting reason for choosing FlexRay is that it can replace several CAN buses, thus reducing wiring, connectors, gateways and system partitioning effort (Berwanger et al; 2005). TTP/C has been implemented on faster systems than FlexRay and is slightly more expensive.

It should be noted that Figure 16 does not capture the holistic costs and misses a lot of key information. It does not show data throughput for each network technology which does tend to be but does not necessarily have to be directly related to bit rate. The received wisdom that LIN is the low cost technology, CAN is the medium cost technology and FlexRay is an expensive technology is too much of a simplification. The general trend is that the higher the bandwidth, the higher the cost of nodal implementation. In practice many other factors must be considered such as cost of training employees and buying tools, the architecture employed itself and warranty costs.

3. CAR DOOR SYSTEM LIN BASED HARNESS – CASE STUDY

3.1. Case Study Overview

This case study was reported in submission two and was an exploratory study carried out for a customer who is a wiring harness manufacturer. The report described a study that compared the cost of a hardwired electrical architecture and a LIN bus based alternative. The main objective of the study was to compare the two architectures in terms of the cost of the bill of materials and ascertain if they could be estimated from just signal and node information. Estimation from signal and node information is important in the design of an automotive electrical architecture since this is likely to be the only information available at the very beginning of the design process. During this beginning stage, electrical nodes are partitioned between hardwired and different network technologies. The most mature version of NetGen at the time was that for LIN technology. One of NetGen’s users is a wiring harness manufacturer with offices in the United Kingdom. The wiring harness manufacturer had an interest in understanding how to compare hardwired and LIN bus based car door electrical architectures based on a target cost and therefore this became the subject of the study in submission two.

The study began with a review of published literature on the subject of automotive electrical architecture cost modelling, and design to cost processes. Although some interesting studies were found, generally there was very little literature published in the field probably due to the commercially sensitive nature of this subject.

Wire Cost Modelling

The wiring harness manufacturer provided a dataset for the study that contained the cost of the bill of materials of a driver's door wiring harness from a small passenger car that was manufactured in the United Kingdom. This was analysed to produce wire cost models that could then be applied to the problem of designing a driver's door electrical wiring harness. The dataset contained data for intra-door wires only and also the associated component cost information such as for clips, terminals, grommets and connectors.

A model for intra-door wire cost was developed based on a previous study (McLaughlin, 1993). It was adapted to incorporate the real data from the wiring harness manufacturer. The McLaughlin study simply had a single expression for the typical cost of a wire. However in the study in submission two this was broken down into two parts; the copper wire and wire connector component costs. This gave the following equation that was used for the estimation of typical intra-door wire cost:

$$(\textit{BOM of Wire} + \text{Wire Component Cost Per Wire}) \text{ x LABOUR} \qquad (1)$$

Where

BOM of Wire was taken from the analysis in submission two and had a mean value of 0.042 Euros and a standard deviation of 0.023 Euros

Wire Component Cost Per Wire was shown in submission two to be 0.051 Euros

LABOUR was a cost factor as stated in the McLaughlin study to be 1.6 to 1.8 (McLaughlin, 1993). The wiring harness manufacturer involved in the case study also agreed with this assumption on the factor.

***LABOUR* Multiplier**	**Mean Wire Cost (Euros)**
LABOUR Multiplier of 1.6	(0.042 + 0.051) x 1.6 = 0.14.88
LABOUR Multiplier of 1.8	(0.042 + 0.051) x 1.6 = 0.1674

Table 4: Mean wire cost by using equation (1)

Table 4 shows that depending upon the *LABOUR* multiplier used, the mean cost of a wire varies between 0.1488 to 0.1674 Euros. However, the wiring harness cost data for the driver's door did have a spread in the cost of each wire which is represented by a standard deviation of 0.023 Euros. This is shown in the two box plots for each LABOUR value in Figure 17. No particular shape for the distribution is assumed here. Figure 17 shows that the cost of a wire is between 0.09 and 0.26 Euros.

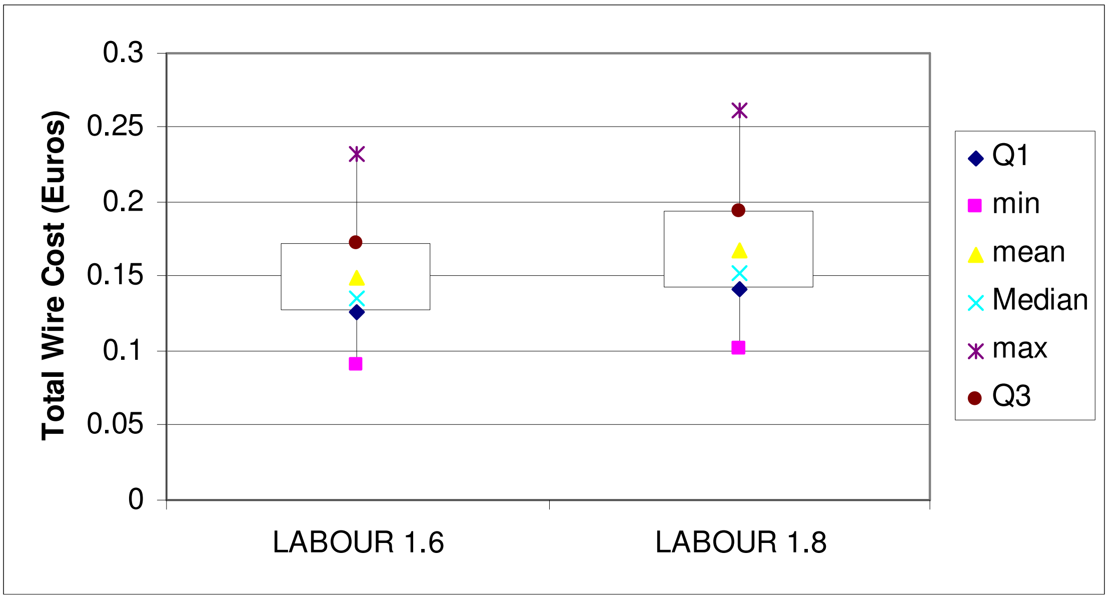


Figure 17: Two possible distributions for the cost of an intra-door wire

Investigation into Inter-Door Wire Cost

The data provided by the wiring harness manufacturer contained the cost of the bill of materials for the wiring harness of a driver's door. Therefore it was possible to analyse this data and produce a cost model for intra-door wires.

However, the driver's door is not a stand alone system and within the door electrical system the interactions between each of the doors should be considered, since these will be the longer signal wires and therefore their removal provides the most potential to benefit from the adoption of LIN. If the entire door electrical system is to be considered then inter-door wire cost models will need to be taken into account. Unfortunately such data was not provided by the wiring harness manufacturer and therefore some further investigation was required.

It was discovered by analysing the wiring harness cost dataset that the mean length of the intra-door wires was slightly over one metre (1032mm) with a standard deviation of 393mm. However due to the lack of data for inter-door wiring, the distribution of wire length and cost is therefore unknown. There are also two types of inter-door signal wire that may exist. According to the wiring harness manufacturer, inter-door wires that are used within a hardwired architecture are likely to be current carrying wires probably of 1.25mm gauge, whilst the inter-door wires added in the LIN bus architecture are likely to be 0.5mm gauge. Therefore there is likely to be variability in the cost of inter-door wires due to the difference in wire gauge and also variation in inter-door wire length between the different doors. Much variation in the length can be caused by routing of the harness through the chassis.

To help ascertain the typical length of an inter-door wire for the door electrical system of a small passenger car, the wiring harness of a mark 4 Vauxhall Corsa VXR was measured. It was measured as a complete harness unit with no harness strip down to measure individual wires. It was found that the inter-door wire length for this vehicle model was in the region of three metres.

Based on a typical length of three metres, it was determined that the bill of materials for a single inter-door network wire (0.50 gauge) is 0.051 (component costs) plus 0.12 (copper wire costs) which equals 0.17 Euros per inter-zone LIN bus wire. As the *LABOUR* multiplier has been stated to be between 1.6 and 1.8, then the cost of an inter-door network wire within a manufactured wiring harness is in the range 0.27 to 0.31 Euros.

It has already been stated that inter-door large current carrying wires will tend to be of a heavier gauge such as 1.25 gauge. Based on a typical length of three metres, it was determined that the bill of materials for a single current carrying inter-door wire (1.25 gauge) is 0.051 (component costs) plus 0.25 (copper wire costs) which equals 0.30 Euros per wire. If the stated *LABOUR* multiplier is between 1.6 and 1.8, then the cost of an inter-door network wire within a manufactured wiring harness is in the range 0.48 to 0.54 Euros.

Therefore it can be seen there is potentially a lot of variation in inter-door wire cost caused by *LABOUR* factor, wire length and wire type.

Application to Case Study

The aim of the case study for the wiring harness manufacturer was to ascertain whether replacing hardwired integration with a LIN equivalent in a small French passenger car driver's door would result in a reduction in the cost of the bill of materials. The target to beat was twelve Euros. The driver's door wiring harness, which was originally a hardwired implementation, contained over fifty wires.

It quickly became clear that there would be negligible savings if only the driver door was considered and therefore in the first part of the analysis, the system functionality of the entire door electrical system was analysed in terms of its features perceived by the customer (e.g. electric windows). These features were then decomposed into the nodes and signals of the electrical architecture. Table 5 shows the functions contained within each of the passenger car doors.

Passenger Door	**Driver's Door**
• Passenger Switches • Electric Window • Electric Mirror • Electric Mirror Heater • Indicator • Central Locking • Puddle Lamp	• Driver Controls for All Windows and Mirrors • Electric Window • Electric Mirror • Electric Mirror Heater • Indicator • Central Locking • Puddle Lamp
Left Rear Door	**Right Rear Door**
• Passenger Switches • Electric Window • Central Locking • Puddle Lamp	• Passenger Switches • Electric Window • Central Locking • Puddle Lamp

Table 5: Body Control Functions by Door

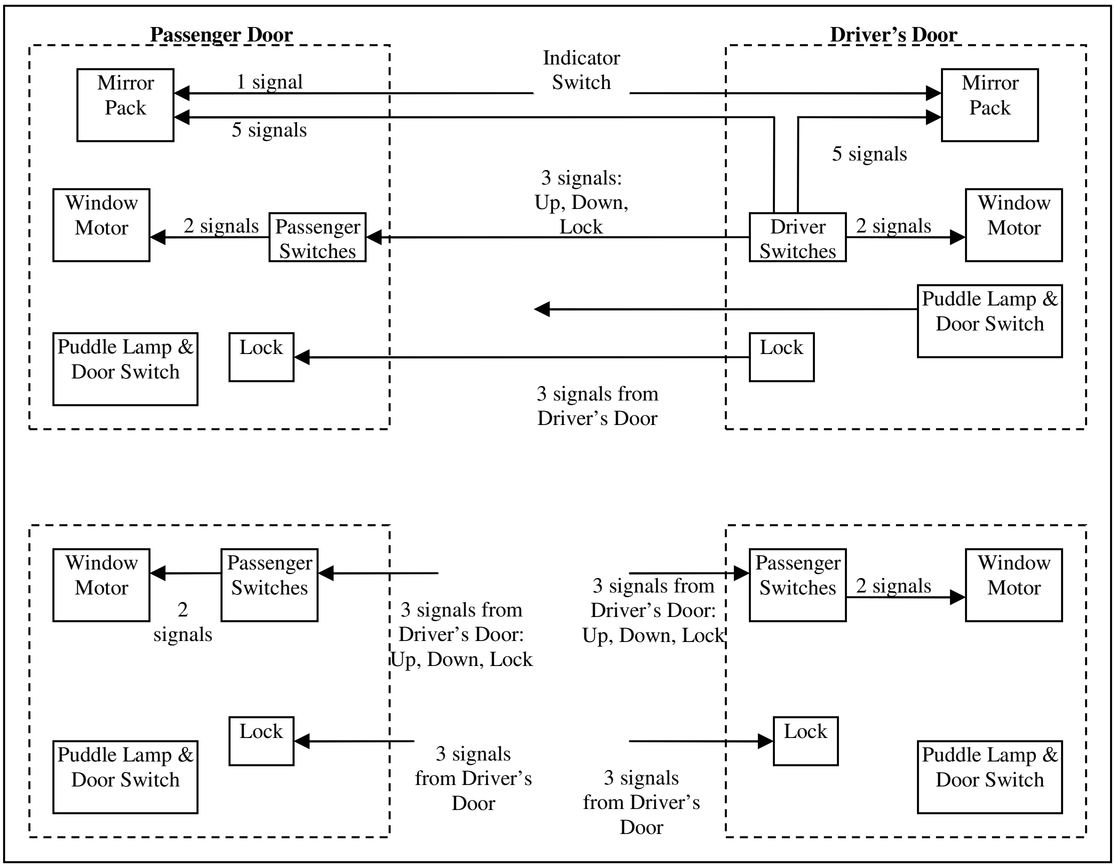

Passenger Door
Driver's Door
Mirror Pack
1 signal
5 signals
Indicator Switch
Mirror Pack
5 signals
Window Motor
2 signals
Passenger Switches
3 signals: Up, Down, Lock
Driver Switches
2 signals
Window Motor
Puddle Lamp & Door Switch
Puddle Lamp & Door Switch
Lock
Lock
3 signals from Driver's Door
Window Motor
Passenger Switches
2 signals
3 signals from Driver's Door: Up, Down, Lock
3 signals from Driver's Door: Up, Down, Lock
Passenger Switches
2 signals
Window Motor
Lock
Puddle Lamp & Door Switch
3 signals from Driver's Door
3 signals from Driver's Door
Lock
Puddle Lamp & Door Switch

Figure 18: Hardwired architecture

Figure 18 shows the topology of the hardwired electrical architecture for all four doors. From Figure 18 it should be noted that puddle lamp switches have been assumed to be turned on/off via Door Switch, the Mirror signals include four motor control and one heater control, there are 23 inter-door signals, there are seven intra-door signals in the driver's door, whilst only two in the other doors.

An alternative to the hardwired architecture based on LIN technology was proposed which replaced hardwired signals with alternatives LIN ones. The topology is shown in Figure 19 but is not necessarily the lowest possible cost LIN topology for a door system with the aforementioned features. However it is a reasonable architecture to look at in this study.

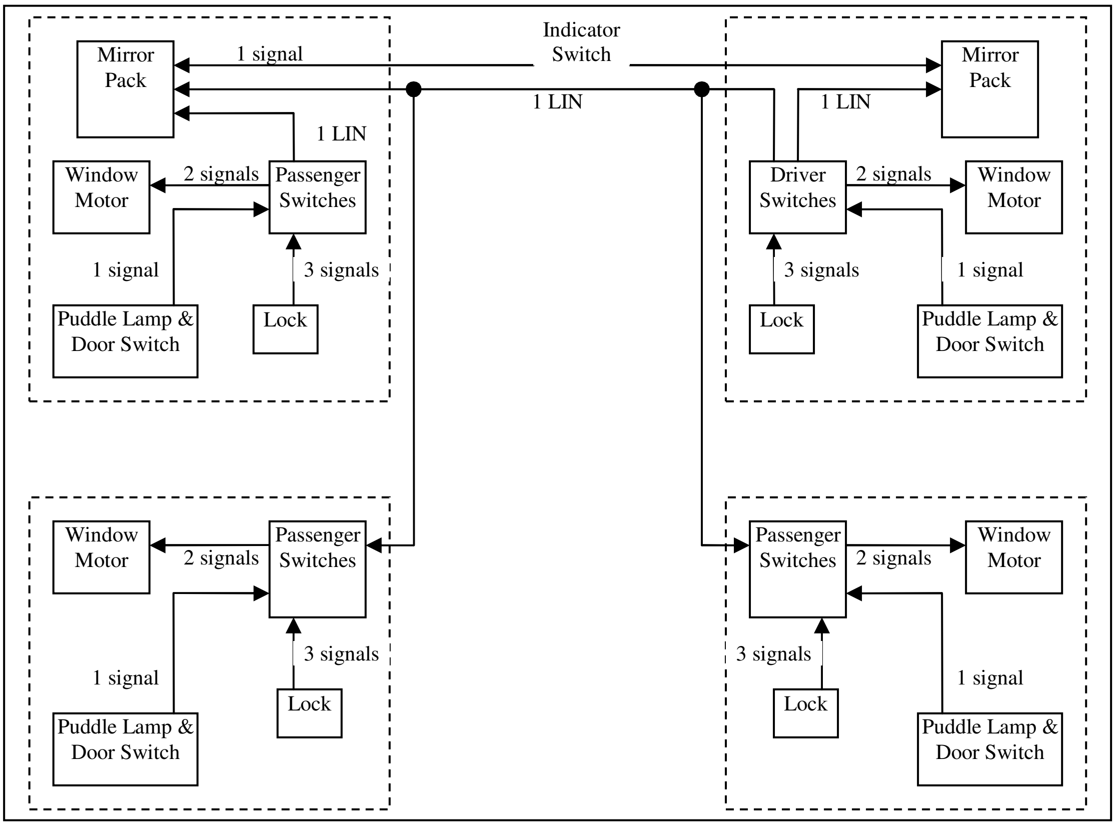


Figure 19: LIN Bus Based Architecture

A signal substitution analysis was used to ascertain any potential savings in terms of the reduction in the number of and types of signal wires (e.g. inter- and intra-door wires). Wires were split into two categories; intra-door wires for wires within a particular door and inter-door wires for wires between doors. The signal substitution analysis is shown in Table 6. From the table it can be seen that twenty inter-door wires have been removed but two intra-door wires have been added. One other thing that can be seen from it is that with the driver's door, five inter-door wires have been replaced with one LIN bus wire. Therefore there is a saving of only four wires within the driver's door which means that there is little benefit in analysing the driver's door on its own. There are more savings when the entire vehicle door systems are analysed, mainly due to the inter-door wire savings.

Although twenty inter-door wires have been removed and two intra-door wires have been added, the two architectures still need to be compared on a cost of bill of materials basis. The result of such a comparison will be dependent upon modelling the cost of both intra-door and inter-door wires and also the cost of adding LIN communication functionality to each of the LIN nodes.

Removed				Replaced With			
Inter-door Wire Type	No.	Intra-door Wire Type	No.	Inter-door Wire Type	No.	Intra-door Wire Type	No.
Mirror Control (Passenger's Door) - 5 current carrying signal wires between Driver's Door and Passenger Door	5			LIN Bus Wire	1		
Window Control (Passenger's Door) – 2 current carrying signal wires between Driver's Door and Passenger Door	3						
Central Locking – 3 signal wires between Driver's Door and Passenger Door	3					3 signal wires from Driver's Central Locking to Driver Switch Node	3
						3 signal wires from Passenger's Central Locking to Passenger Switch Node	3
		Mirror Control (Driver's Door) - 5 current carrying signal wires between Driver's Door and Passenger Door	5			LIN Bus Wire	1
Rear Left Passenger Door Window Control – 2 current carrying signal wires	3			LIN Bus Wire	1		
Rear Left Passenger Door Central Locking – 3 signal wires	3						
Rear Right Passenger Door Window Control – 2 current carrying signal wires	3			LIN Bus Wire	1		
Rear Left Passenger Door Central Locking – 3 signal wires	3						
TOTALS	23		5		3		7
Inter-door wires removed		Intra-door wires removed		Inter-door wires added		Intra-door wires added	
Net Savings (no. wires)		Inter-door		Intra-door			
		20		-2			

Table 6 : The signal substitution between hardwired and LIN architectures.

The signal substitution analysis showed that for a move to the LIN based architecture, the main benefit appears to be from the removal of twenty inter-door wires, whilst two intra-door wires have been added. To ascertain whether the LIN candidate architecture may be of lower cost that the hardwired original, equations that express the cost of each of the architectures must be determined and then set equal to each other. The expression can then be used to ascertain what target LIN nodal cost must be achieved for the LIN candidate architecture to be at least the same cost as the hardwired original. The information from the signal substation analysis was used to set up the following expression for evaluating when both architectures cost the same:

$$23\ C_{\mathrm{Inter}} + 5\ C_{\mathrm{Intra}} = 3\ C_{\mathrm{Inter}} + 7\ C_{\mathrm{Intra}} + 6\ C_{\mathrm{Node}} \tag{2}$$

Where

C_{Inter} is the cost of an inter-door wire

C_{Intra} is the cost of an intra-door wire

C_{Node} is the add-on cost of integrating LIN communications to the node.

Therefore to ascertain the target cost of a node for this particular example, it was required to see the nodal cost when both hardwired and LIN architectures cost the same. This is done by rearranging the above equation for C_{Node} calculation so it becomes:

$$C_{\mathrm{Node}} = (20\ C_{\mathrm{Inter}} - 2\ C_{\mathrm{Intra}})\ /\ 6 \tag{3}$$

Equations (2) and (3) assumed that both inter-door and intra-door wires in both architectures is of the same gauge. This is unlikely as it has already been determined that the inter-door wires in the hardwired architecture are likely to be current carrying (e.g. 1.25 gauge), whilst in the LIN based architecture they will be LIN bus wires (e.g. 0.50 gauge). Therefore the expression for calculating the target nodal cost becomes:

$$C_{Node} = (23\ C_{InterCurrent} - 3\ C_{InterBus} - 2\ C_{Intra}) / 6 \quad (4)$$

Where

$C_{InterCurrent}$ is the cost of an inter-door current carrying wire as found in the hardwired architecture

$C_{InterBus}$ is the cost of an inter-door LIN bus wire

C_{Intra} is the cost of an intra-door wire

C_{Node} is the add-on cost of integrating LIN communications to the node.

This expression can be rearranged for calculating the target nodal cost:

$$C_{Node} = (20\ C_{InterCurrent} - 3\ C_{InterBus} - 2\ C_{Intra}) / 6 \quad (5)$$

Where

$C_{InterCurrent}$ is the cost of an inter-door current carrying wire as found in the hardwired architecture

$C_{InterBus}$ is the cost of an inter-door LIN bus wire

C_{Intra} is the cost of an intra-door wire

C_{Node} is the add-on cost of integrating LIN communications to the node.

Of course one of the aims of this study was to ascertain whether architecture cost can be estimated from just node and signal information. This would therefore not split signal wires into their different types but would simply assume that intra-door and inter-door signals are of the same mean cost. If equation (2) is rearranged under the assumption of all signal wires being of the same cost (therefore the resultant difference between the two architectures was that there were eighteen signal wires removed), the following is obtained:

$$C_{Node} = 3\ C_{Wire} \tag{6}$$

Where

C_{Wire} is the cost of a wire in either architecture

C_{Node} is the add-on cost of integrating LIN communications to the node.

So the result at this stage of the case study was that there were three different equations for calculating the target nodal cost of the LIN based architecture to make it at least the same cost as the hardwired architecture. Each of the equations for the architecture cost comparison has different amounts of information. Increasing the amount of information available ultimately requires further design efforts. It is desirable to be able to make a reliable estimation with the least amount of information as possible. The target nodal cost C_{Node} equations were explored using a sensitivity analysis to help ascertain how each performed.

Sensitivity Analysis

The equations (3) (5) and (6) developed can combined with the wire cost model to calculate a target LIN nodal cost (C_{Node})that must be met for the LIN candidate architecture to be at least the same cost as the hardwired original. Of course this implies that if a cost lower than the target is achieved, then the LIN candidate architecture will be of lower cost. The signal substitution analysis showed that the architecture in this case study was not that sensitive to intra-door wire cost as the number of these wires only changed by two between the hardwired original and the

LIN candidate. Also the wire model investigation showed that the cost of the intra-door wires was very small. However the move from hardwired to LIN architectures resulted in twenty inter-door wires being removed which were also shown to be of much greater length and cost. Therefore the sensitivity to inter-door wire cost, LIN node cost and architecture cost was explored.

Figure 20 shows the sensitivity of the cost of both hardwired and LIN bus based architectures to inter-door wire and LIN node cost based in equation (3). This is shown with intra-door wire cost set to its lower bound of 0.09 Euros. It can be seen that as LIN node cost decreases then so does the cost of the architecture. As the cost of inter-door wire decreases, this will lead to very challenging LIN node cost targets. If for example it were assumed that the effect of intra-door wire cost was negligible and the mean inter-door wire cost from the hardwired architecture is considered (0.54 Euros at 1.8 *LABOUR* factor), then the target nodal cost would be around 1.65 Euros.

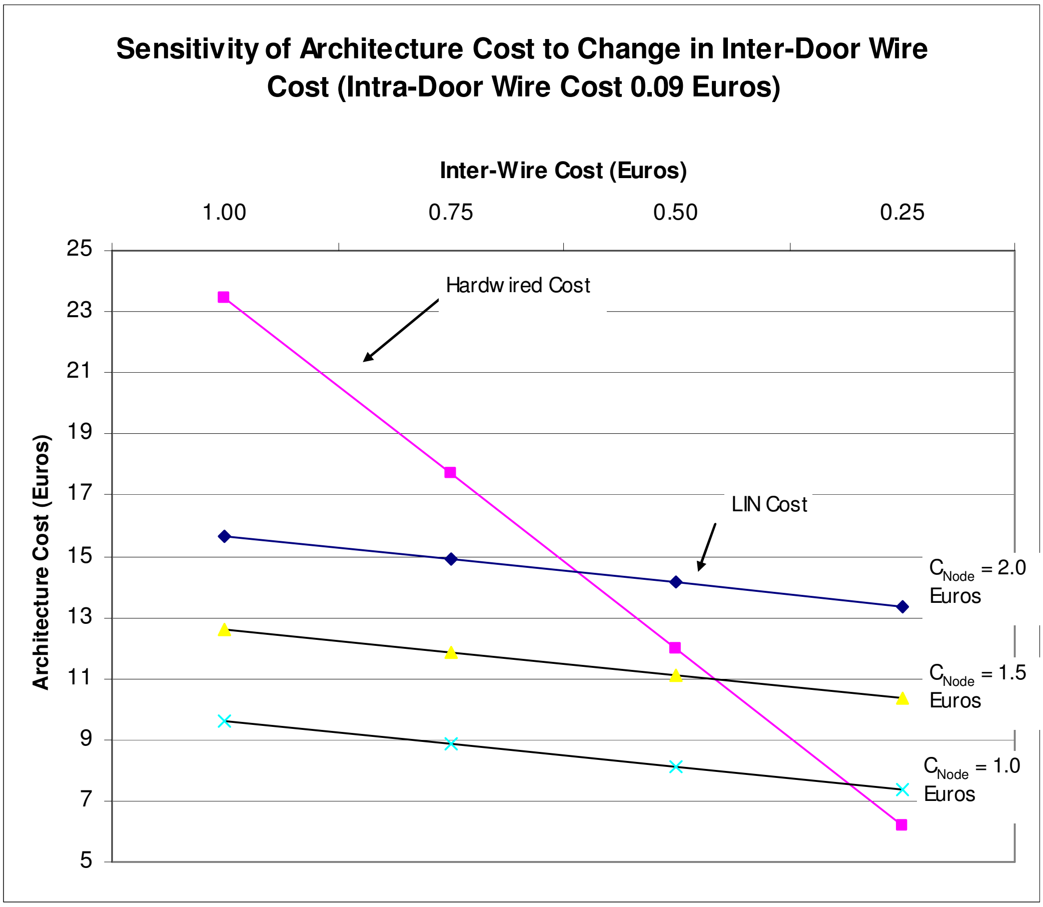


Figure 20: Sensitivity of both hardwired and LIN bus based architecture cost to both inter-door wire and LIN node cost.

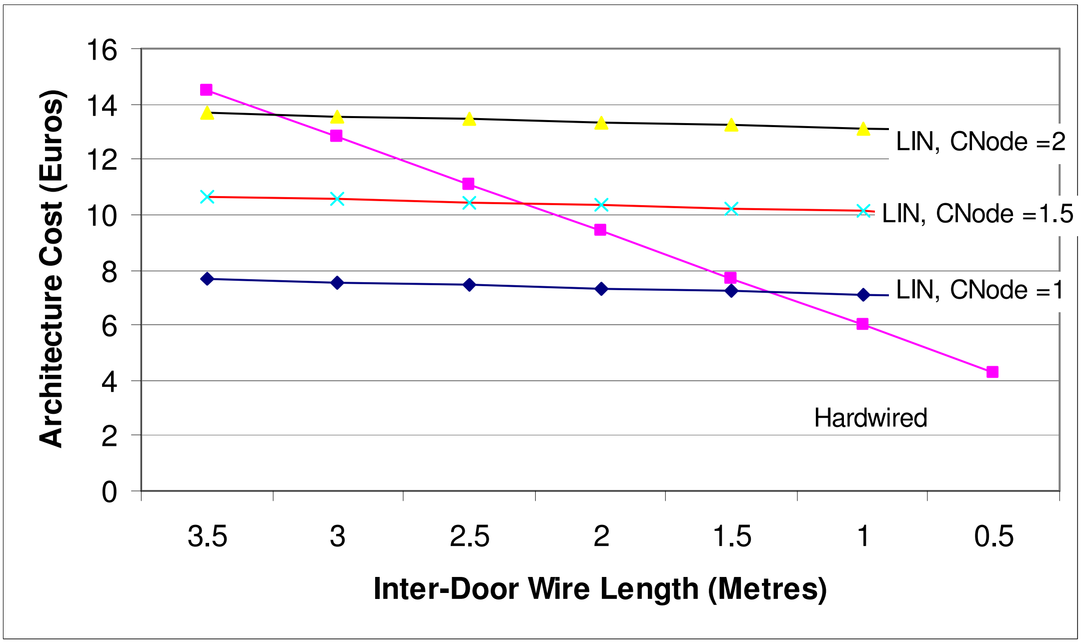


Figure 21: Cost of both hardwired and LIN architectures

For the model described by equation (5), there were two types of inter-door wires (LIN bus and current carrying wires). Therefore for sensitivity analysis, inter-door wire length was used so that both inter-door costs could be combined and displayed on a single graph. Figure 21 shows the sensitivity of the cost of the two architectures to variation in length of the inter-door wires and C_{Node}. The hardwired architecture cost has been calculated using the cost of current carrying wires (1.25 gauge) which are almost twice the cost of the network inter-door wires. The LIN architecture cost has been calculated using both inter-door wire types from equation (5). By examining Figure 21 it can be seen that if a nodal cost of one Euro can be achieved then the LIN architecture cost will be lower for inter-door wire lengths of 1.5 metres. However it was previously established that the inter-door wire length of a typical small passenger car is around three metres and therefore using this measure, a nodal cost of around 1.9 Euros becomes the target. An extension to this model added one power line for each LIN node. This additional cost to the LIN architecture was offset by a reduction in C_{Node} to 1.75 Euros per node.

Figure 22 shows the sensitivity analysis based on the model in equation (6) which only considers the number of nodes and signal wires. It shows the relationship between C_{Wire} versus C_{Node} and therefore can be used to ascertain the target nodal cost of a LIN node based on the mean cost of wires in the architecture.

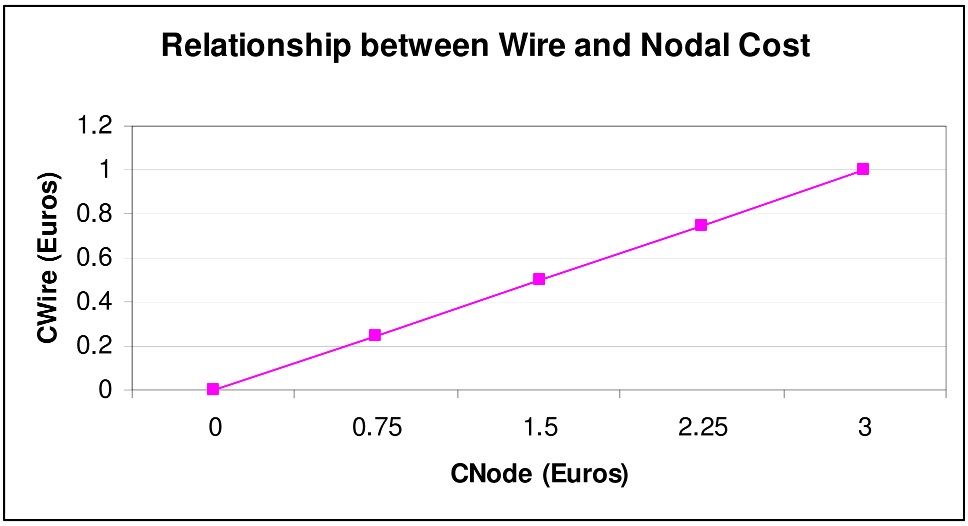


Figure 22: Graph showing linear relationship of C_{Wire} versus C_{Node}.

The estimation of inter-door wire cost was based on a mean length of three metres and resulted in a mean cost of 0.54 Euros. Therefore from these values the maximum target nodal cost that would be estimated from the number of signals alone would be 1.61 Euros.

Analysis of C_{Node} Estimate's Sensitivity to Changes in Level of Information

The case study in submission two developed a number of different cost models that include different amounts of information. The most simplistic model shown in equation (6) considered only the number of signals and nodes, whereas the more sophisticated model in equation (5) with additional extensions included information on wire gauge and power distribution. Therefore a system designer would have the ability to choose the appropriate model dependent upon the level of information available to him at the time. However the results obtained from each of the models did differ slightly. The sensitivity analysis that was carried out, investigated how variations of the level of information changed the estimation of nodal cost with the aim to answer the question of what level of information is sufficient to carry out a comparison of the two architectures. Table 7 compares each of the estimations made (with the assumption that the mean inter-door wire length is three metres). It can be seen that they are of a similar level and vary between 1.61 to 1.9 Euros. At first glance this nodal cost would appear to be a very challenging target to add LIN capability to the electrical architecture nodes.

Ultimately there was no way within the scope of this study of saying which level of information is best but it has been demonstrated that there is a difference in results between each level. In general the more information that is made available to the designer, the better the accuracy of the estimation that is made. Case 4, in which wire types are split into inter-door, intra-door and power distribution wires were also considered, is the highest level of information and will therefore most probably give the most realistic estimation of C_{Node}. The addition of power wires did result in a reduction of C_{Node} (1.9 to 1.75 Euros) because this cost reduction was caused by offsetting the cost of the power wires.

Case Number	**1** based on eqn (6)	**2** based on eqn (3)	**3** based on eqn (5)	**4** based on eqn (5) with extensions
Level of Information	Number of Signals and Nodes Only	Splitting Signal Wires into Inter-door and Intra-door	Splitting Inter-door Signals into Wire Gauge Types	Considering extra wires required for Power Distribution
Target Nodal Cost Indication	~1.61 Euros	~1.65 Euros	~1.9 Euros	~1.75 Euros

Table 7: Variation in target LIN nodal cost to increases in architecture information

The result from this case study, means that an electrical architecture designer could state that the target nodal cost for the addition of LIN communication components is in the range of 1.61 to 1.9 Euros. It appeared to be a very challenging target LIN nodal cost is required to ensure that the cost of bill of materials of the LIN architecture is lower than the hardwired equivalent (i.e. between 1.61 Euros estimated with Case 1 up to 1.9 Euros estimated with Case 3).

The results from the sensitivity analysis were inconclusive as to which level of information is the most appropriate. However, at the beginning of an architecture design the more information that is available, the more accurate the estimations that will be made and therefore the more realistic the results from the comparison. Without having complete electrical architecture designs along with full cost information for both the hardwired architecture and the candidate LIN architecture, it is impossible to absolutely prove which methodology is best. Therefore the comparisons obtained should only be considered for decision support.

Conclusion of Case Study

It was concluded that the cost of the electrical architecture in this particular case study was more dependent upon inter-door wire cost and add-on LIN communications cost. This demonstrated that it is extremely important to study the cost of an entire system (in this case a door electrical system) rather than a single zone of an electrical architecture. The most significant wire count savings from the adoption of LIN based integration were from inter-door wire reduction (i.e. in this case study, the inter-door wires which were replaced with a LIN bus).

Using the number of signal wires with an appropriate wire cost model and the number of nodes is not a sufficient amount of information to make a reasonable comparison of cost of the two architectures. Although the estimation made with only node and signal information in this case study gave a reasonable estimation, the better the understanding of inter-door wiring cost, the more there is the opportunity to use signal and node information to make a reasonable cost comparison. Therefore understanding the types of wire is also important, for example whether a wire is a low current signal wire or a large current carrying signal wire. If this information is provided, then the methodology presented will be a useful tool for the development of a cost model that can be used for decision support. It will therefore provide an insight into wiring harness cost in such a way that was not previously available.

3.2. Case Study – Main Outcomes

This work reported in this submission two and described in this chapter developed the preliminary design to cost process. This can be summarised as a process for comparing a hardwired original electrical architecture with a LIN based alternative and ascertaining a cost target for adding LIN communications capability to electrical architecture nodes. Key steps were:

1. develop an appropriate wire model for the wire type (intra- and inter-zone in this case)
2. ascertain the number of wires and types that are to be added or removed (i.e. by signal substitution analysis)
3. analyse the sensitivity for estimating the target add-on nodal cost to be achieved for the LIN candidate to be the same price or lower than the hardwired original

The methodology of the analysis presented focused on a comparison of the costs of the bill of materials of two different methods of electrical system integration, in this case hardwired and LIN based architectures. It is important to note that it was not a methodology for cost estimation but a methodology for comparing two architectures. Cost estimation itself would require the designers to go a lot further down the design process which could take many man months. The main motivation of the research presented was to establish a methodology with which candidate electrical architectures can be realistically compared on a cost basis at a very early stage of design. Wire cost was investigated and this led to the identification of a target LIN node cost that should be met to help ensure that the LIN architecture will be of lower cost than the hardwired alternative. Historical wire cost data was studied but there was no data available on the LIN nodal costing or some of the other costs that might affect the choice of architecture. Therefore a number of areas of further work were recommended.

The architecture cost comparison methodology presented did not take account of other costs such as non-recoverable software development costs associated with LIN, cross platform portability (which would amortise the costs across a number of platforms), vehicle assembly, warranty, weight savings (which will ultimately reduce costs for both the customer in terms of fuel consumption and the manufacturer in terms of CO_2 based taxation). Making a justification for the use of LIN over a hardwired architecture, based on the cost of bill of materials alone, appeared to be very challenging and as it resulted in very low target nodal costs. Therefore there is motivation for research into the effects of other factors on both LIN and hardwired architectures.

Use of Process

A key point about the use of the process developed is it cannot be used to predict the cost of an alternative architecture based on a particular harness. However it does have two other applications; estimation of target nodal cost or estimation of architecture cost difference.

If the target nodal cost for adding LIN communications (C_{Node}) is not known, the process can be used to estimate the target C_{Node} that must be met for the LIN candidate architecture to be the same cost as the hardwired original. This was the process followed for the mass vehicle wiring harness manufacturer case study that was described in submission two.

It can also be used to estimate whether a move from hardwired integration to LIN would result in a cost reduction or increase. This is the case if C_{Node} is known, e.g. if legacy ECUs are already available and are to be used in a new architecture design. Use of the wire model information and C_{Node} values will give delta values from the architecture equations.

It was recommended that further work should include an investigation of how target nodal cost can be achieved and how it varies with changes in architecture and signal requirements. The aim of this recommendation was to help ascertain whether the target nodal cost values obtained within this submission are realistic and achievable (e.g. 1.61 to 1.9 Euros). It was stated that it would be interesting to understand the cost of FETs, relays, switches and the size of application software and how this affects the target cost of a network node. This recommendation gave motivation to the work in submissions three and four since they helped uncover some of the cost implications of LIN communications and embedded software generally.

Submission three investigated the relationship between microcontroller ROM and RAM capability and microcontroller unit cost. Data from the semiconductor manufacturer Microchip for their PIC16 and PIC18 microcontroller families was investigated. It was found that there was a relationship between cost and ROM/RAM capability with the implication being that an increase in the LIN communications requirements would result in an increase in the LIN communications embedded software ROM and RAM requirements. Therefore an increase in the LIN communications embedded software ROM and RAM requirements would result in increased microcontroller cost. One other output from submission three was that it was discovered that there tends to be a linear relationship between the ROM and RAM capability across a microcontroller family. This therefore provided the basis of a microcontroller capability assessment process for assessing how well it is suited for a particular piece of embedded software (e.g. in this case a LIN communications stack). This is described further in chapter 5.

The relationship between LIN communications requirements and LIN communications software stack ROM/RAM requirements, was explored further in submission four. It was found that it was possible to estimate ROM/RAM requirements of the LIN communications stack from information such as number of nodes, signals, messages etc. using regression models. The process for this is described in chapter 5.

4. PROCESSES FOR ARCHITECTURE SELECTION

This chapter formally describes the processes that were developed in submissions two and five which essentially help an automotive electrical architecture designer select an alternative architecture to a hardwired integrated based on LIN technology. The first of these is a design to cost process which helps the designer by comparing a base and revised architecture and also estimates a target cost for the addition of LIN functionality to architecture nodes. This is described in section 4.1 of this chapter. The second of these is a design to weight process which helps the designer by comparing a base and revised architecture and also estimates a target weight for the addition of LIN functionality to architecture nodes. This is described in section 4.2 of this chapter.

4.1. Architecture Design to Cost Process

A flow chart of the design to cost process is shown in Figure 23. The process is summarised in the following paragraphs that describe each of its eight stages. The aim of this process is to:

1. Ascertain whether a LIN candidate architecture has the possibility of being lower cost than the hardwired original.
2. If it is possible, what is the C_{Node} (nodal cost) target that must be met for the LIN candidate to be the same cost as the hardwired original. Therefore beating this C_{Node} target will result in a lower cost architecture.

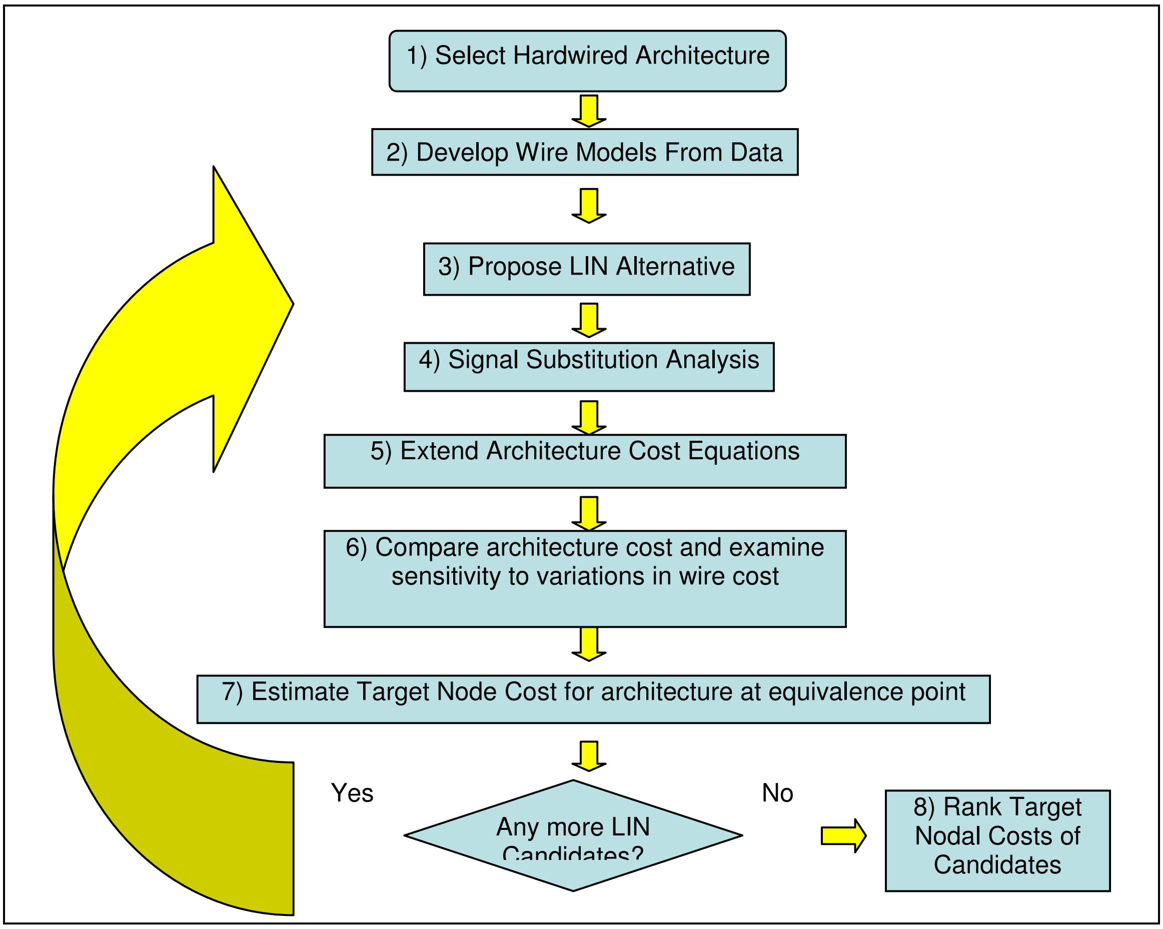

1) Select Hardwired Architecture
2) Develop Wire Models From Data
3) Propose LIN Alternative
4) Signal Substitution Analysis
5) Extend Architecture Cost Equations
6) Compare architecture cost and examine sensitivity to variations in wire cost
7) Estimate Target Node Cost for architecture at equivalence point
Yes
Any more LIN Candidates?
No
8) Rank Target Nodal Costs of Candidates

Figure 23: Process for LIN architecture design to cost

1) Select Hardwired Architecture

The first stage is to select the hardwired architecture or part of the hardwired architecture that is to be replaced with a LIN alternative.

Output from this stage: Hardwired architecture to be replaced.

2) Develop Wire Cost Model

The second stage is to develop a wire cost model from historical data so that the typical cost of a signal wire is available. This signal wire model is then available for later hardwired and LIN architecture comparison. The process used for signal wire model development is shown in Figure 24.

1. Obtain bill of materials for the wiring harness. Ideally this should include names of the signals, signal wire lengths, wire gauge and cost per metre. Also connector and terminal costs were available so that an estimation of these costs per signal wire could be made. If as in submission two, the bill of materials is also not available for the base harness, an alternative from another representative vehicle can be used.
2. Identify zones within the harness: Using a visual clustering method, identify functional zones within the harness or architecture. In submission two, the zones were the doors within the automotive electrical system. In submission five the zones were *Front*, *Dash* and *Rear*. Signal wires between zones are referred to as inter-zone, whilst those inside a zone are referred to as intra-zone.
3. Calculate mean inter-zone and intra-zone signal wire cost. This is achieved by adding together the cost of all of the wires and dividing by the number of wires. This is carried out separately for *Inter-Zone* and *Intra-Zone* wires. Therefore two values are obtained; one for inter-zone wires, the other for intra-zone wires.
4. Calculate mean harness component cost per wire. This is achieved by adding up the cost of the components on a wire by wire basis to get the mean cost per wire. These are components such as connectors and terminals that will vary with a change in the number of signal wires.

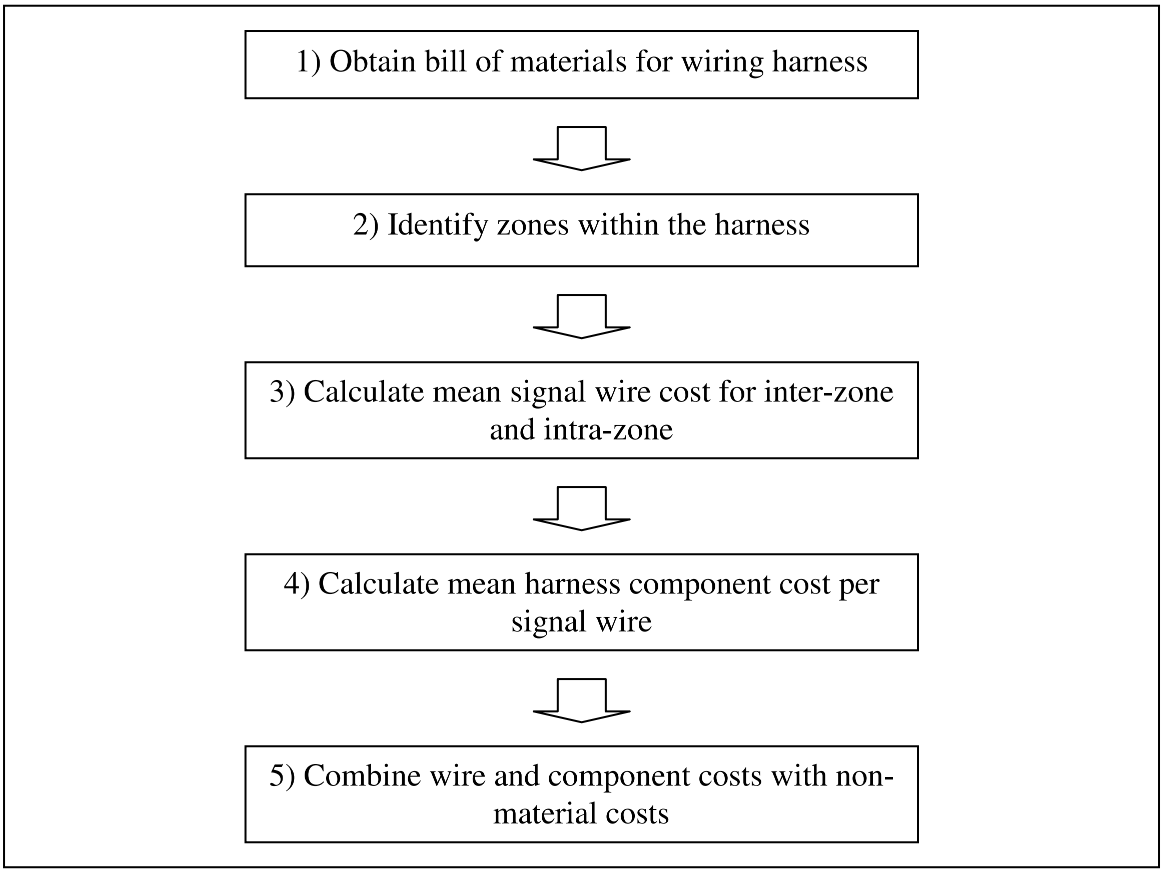

1) Obtain bill of materials for wiring harness
2) Identify zones within the harness
3) Calculate mean signal wire cost for inter-zone and intra-zone
4) Calculate mean harness component cost per signal wire
5) Combine wire and component costs with non-material costs

Figure 24: Process for Signal Wire Model Development

5. Combine Wire and Components Cost with Non-Material Costs. This is achieved by adding together the per-wire components and wire cost. The result is then multiplied by a *Non-Material Cost* factor to give the typical signal wire cost of the system. This is carried out for both inter-zone and intra-zone signal wires. To form a signal wire model a *Non-Material Cost* factor is applied to the combination of the mean signal wire cost and the per-wire components. The *Non-Material Cost* factor accounts for all non-material piece costs associated with the manufacture of the harness. This includes items such as fixed business overheads, variable costs such as manufacturing labour, logistical costs and profit margin. No data is available on the *Non-Material Cost* factor. However, in a study by McLaughlin (McLaughlin, 1993), the costs of adopting CAN over hardwired integration for automotive control was explored. During this study,

interviews with costing engineers at a major UK car manufacturer indicated that the magnitude of such a *Non-Material Cost* factor was in the region of 1.6 to 1.8. This was also confirmed by the wiring harness manufacturer who was the customer of the study in submission two. The *Non-Material Cost* factor, the wire and component costs were combined to form a wire cost model as used in submission two and is restated here:

$$\text{Wire Cost} = (\text{Wire} + \text{Wire Component}) \text{ x } \textit{Non-Material Cost} \tag{7}$$

Output from this stage: Wire cost model for each of the signal wire types.

3) Propose LIN Architectures

During this stage, one or more alternative architectures based on LIN are proposed. This is not a detailed architecture proposal but simply one outlining the signal and node assignment.

Output from this stage: Signal and node based architecture candidates.

4) Signal Substitution Analysis

During signal substitution analysis, each of the LIN candidate architectures is compared with the hardwired one, signal by signal, and the hardwired signals that have been replaced by a LIN equivalent are outlined. It also identifies which wires have been added to act as inputs/outputs to LIN nodes. One output of this analysis shows the savings in terms of number of *Inter-Zone* wires (due to replacement with a LIN equivalent signal) compared with an addition of *Intra-Zone* wires (due to the addition of LIN node input/output signal wires). The other output is a number of coarse architecture equations for the hardwired architecture and each of the LIN candidates.

Output from this stage:

- Identification of the hardwired signals that have been substituted.
- Coarse architecture equations for the hardwired architecture and each of the LIN candidates

5) Extend Architecture Cost Equations

The coarse architecture equations from the signal substitution analysis for hardwired and LIN candidates are extended at this stage to add any other components that are deemed to be necessary and of reasonable cost significance for each signal wire (e.g. relays or low-side drivers for switching lamps).

Output from this stage: Extended architecture cost equations

6) Compare architecture cost and examine sensitivity to variations in wire cost

At this stage the sensitivity of the architecture cost to changes in wire cost (C_{Wire}) and the cost of adding LIN communications functionality to each of the architecture nodes (C_{Node}) are explored. Variations in C_{Wire}, C_{Node} and architecture cost are plotted on a diagram which visually shows how variations in the cost of wire affects the C_{Node} target. Typically the largest source of uncertainty and variation in both inter-zone and intra-zone wire cost comes from the magnitude of *Non-Material Cost* factor. Therefore this is varied to show the variation in wire cost.

Output from this stage: C_{Wire}/ C_{Node} architecture cost sensitivity diagram.

7) Estimate Target Node Cost for architectures at equivalence point

At this stage, the signal wire cost models developed in (2) are applied to the equations developed in (5). Each candidate LIN architecture is set equal to the hardwired architecture and the equation is solved to obtain the target nodal cost. The target nodal cost is the target cost for adding components to each node to make it LIN capable and suitable for the target application.

Output from this stage:

- Estimated C_{Node} target

Stages (3) to (7) are repeated for each of the LIN candidate architectures.

8) Rank Target Nodal Costs of Candidates

This is a new part of the process not previously used in the previous submissions due to fact that there are a number of LIN candidates instead of just the one. During this stage, the target nodal cost, (C_{Node}), for each of the LIN candidate architectures are compared. The architecture costs at the expected mean wire cost values are also compared. The expected mean wire cost values are those around the most likely *Non-Material Cost* factor (e.g. 1.6 to 1.8).

Output from this stage: Ranking of C_{Node} target and architecture cost.

4.2. Architecture Design to Weight Process

A design to harness weight process was developed for the case study in submission five by adapting the architecture design to cost process that was used in submission two. Therefore the architecture design to cost process is shown in Figure 25. The aim of this process is to:

1. Ascertain whether a LIN candidate architecture has the possibility of being a lower weight than the hardwired original.
2. If it is possible, what is the W_{Node} (nodal weight) target that must be met for the LIN candidate to be the same weight as the hardwired original. Therefore beating this W_{Node} target will result in a lower weight architecture.

The stages are:

<u>1) Select Hardwired Architecture</u>

The target harness whose architecture and weight is to be analysed is selected.

Output from this stage: the target architecture for further analysis.

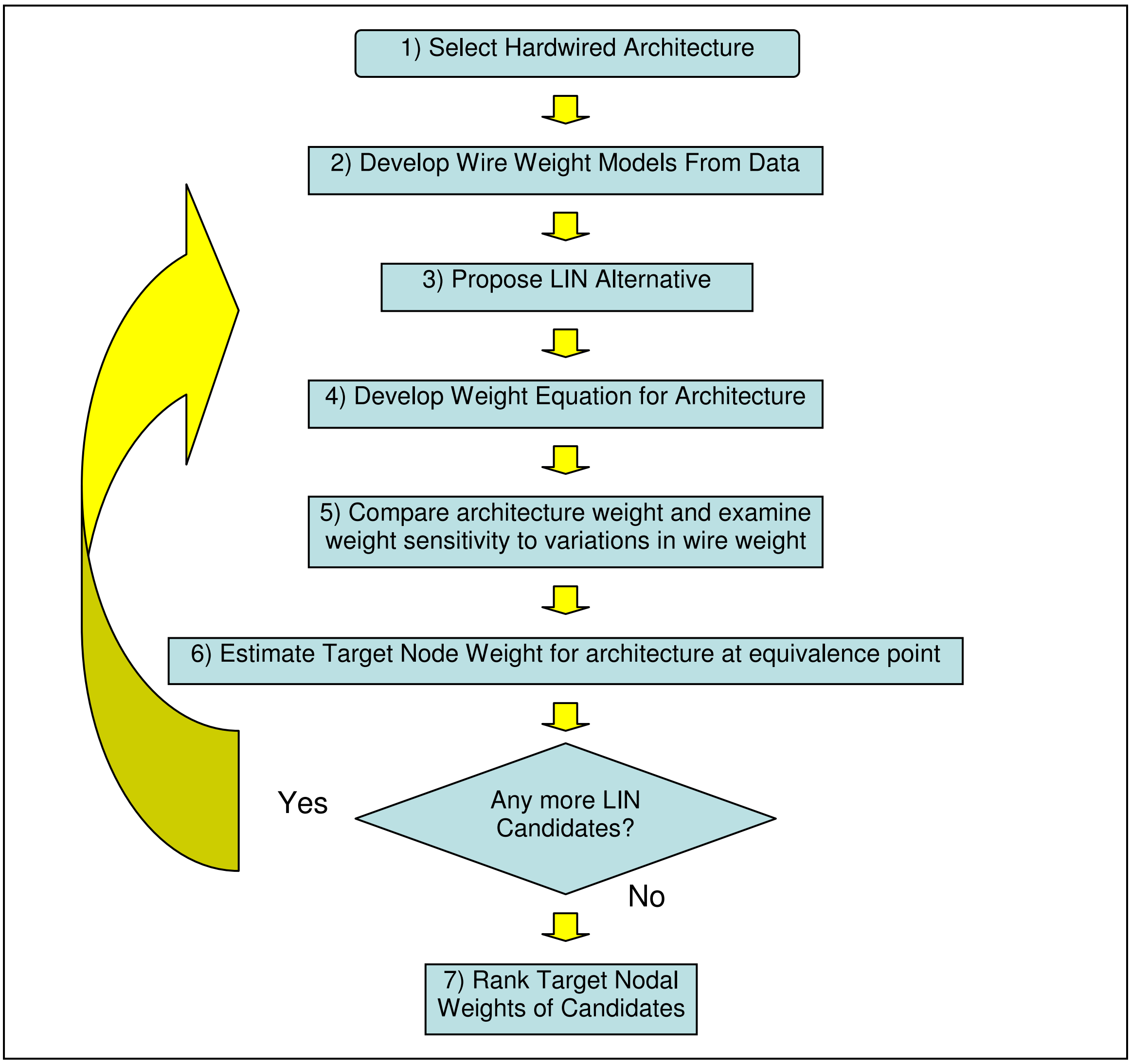

1) Select Hardwired Architecture
2) Develop Wire Weight Models From Data
3) Propose LIN Alternative
4) Develop Weight Equation for Architecture
5) Compare architecture weight and examine weight sensitivity to variations in wire weight
6) Estimate Target Node Weight for architecture at equivalence point
Yes
Any more LIN Candidates?
No
7) Rank Target Nodal Weights of Candidates

Figure 25 : Process for design for harness weight

2) Develop Wire Weight Models From Data

This stage is based on the methodology used for signal wire cost model development. The type of information required is wire length and gauge. With this information and the density of copper, the weight of wires in the architecture can be estimated. The mean wire weight for inter-zone and intra-zone wires is calculated. The wire weight models also require the weight of connectors and wire insulation to be estimated as these will also have a reasonable impact on the harness weight.

A wire weight model is developed based on the weight of the wire, the weight of the insulation and the weight of the connector. The weight of a connector can be obtained from its manufacturer and will typically be a few grammes. The wire insulation will have a proportional impact on the wire weight and is dependent upon the type of material, the wire cross-sectional area and thickness employed. It can vary significantly for different wire types. This results in the following model:

$$\text{Wire Weight Model} = (\textit{Copper Wire Weight} \times \textit{Insulation Factor}) + \textit{Connector Weight} \quad (8)$$

Typical weight for the connectors of a wire is around five grammes. The *Insulation Factor* can have very little impact on the wire weight but in some cases can be as much as 30%.

Output from this stage: Wire weight model for each wire type.

3) Propose LIN Alternative

The LIN alternative architectures for analysis are proposed.

Output from this stage: Candidate LIN architectures.

4) Develop Weight Equation for Architecture

Based on the signal substitution analysis of step 4 of the design to cost process in Figure 23, architecture weight equations are developed that use weight terms instead of cost terms.

Output from this stage: Architecture weight equations.

5) Compare Architecture Weight and Examine Weight Sensitivity to Variations in Wire Weight

W_{Wire} and W_{Node} are used to test the architecture weight sensitivity to changes in wire weight. A major source of this variation is from the unknown level of weight that will come from the wire insulation and therefore this is varied.

Output from this stage: a W_{Wire}/ W_{Node} architecture cost sensitivity diagram comparing the LIN and hardwired architectures.

6) Estimate Target Node Weight for architecture at equivalence point

At this stage the wire weight models developed earlier are applied to the architecture equations. For each LIN candidate, it is set equal to the hardwired architecture weight and the equation is solved to obtain a target nodal weight W_{Node}.

Output from this stage: Estimated W_{Node} target.

7) Rank Target Nodal Weights of Candidates

Each of the candidates are compared and ranked into the order of which is the most likely to provide a lower weight solution.

5. MICROCONTROLLER MEMORY CAPABILITY ASSESSMENT AND VARIANT SELECTION

Chapter 4 described processes for designing a network architecture to a target cost and weight. However other processes also were developed during the course of the research. There were also processes that were concerned with the selection of the most appropriate microcontroller for a LIN application and therefore reducing nodal cost. This chapter describes new processes that were developed during the work described in submissions three, four and five. Firstly the key points from the research that was undertook in submissions three and four that led to the development of the processes are described in sections 5.1 and 5.2 of this chapter. Secondly two processes are described that can be used for selection of microcontrollers for LIN applications, depending upon what type of company is carrying out the project and where they are in the automotive supply chain. The processes will then help in the design of the ECU by the selection of a microcontroller that is most appropriate for the application and for an optimised cost. In the case of LIN, an ECU is an electronic control unit in the traditional sense with PCB, connector and casing or some kind I/O block such as an intelligent sensor or actuator.

The first of these processes (see section 5.3 of this chapter) would be useful to a LIN communications stack supplier if they wish to ascertain how best to design for a particular microcontroller family, or a semiconductor manufacturer who wants to select the amount of memory required in certain microcontroller variants, or finally a vehicle manufacturer or first tier supplier who wants to select stack and microcontroller suppliers for their particular project.

The second of these processes (see section 5.4 of this chapter) is concerned with microcontroller variant selection. It is assumed that the stack and microcontroller suppliers have been selected but not the particular microcontroller variants. The ROM and RAM requirements for the LIN software of each embedded node are estimated from the LIN bus characteristics to assist in microcontroller variant selection. This allows the selection of the most appropriate microcontrollers and provides two key benefits. Firstly but not specifying a microcontroller with surplus ROM or RAM, more competitive quotations to a customer can be made. Secondly, this reduces the risk of having to carry out expensive redesign work for a LIN node which is found not to have enough memory.

Motivation for this study was to understand nodal costing more fully due to the apparently very challenging C_{Node} targets that resulted from the automotive door electrical architecture study in submission two. Therefore there was an interest in finding out whether this challenging target was achievable. Rather that providing an understanding of whether the challenging target was achievable, the work in submissions three, four and five provided processes to help ascertain this.

5.1. ROM and RAM Capability Relationship within a Microcontroller Family

In submission three the relationship between ROM and RAM capability of a microcontroller family and unit pricing was explored. It was concluded that it was clear that both ROM and RAM did affect price but the relationship was not clear enough to be able to generate a generic model for price estimation as a function of ROM and RAM.

Although it was found that it was not possible to model cost as a function of ROM and RAM, one important relationship was found to be inherent in three of the major LIN microcontroller families from two semiconductor manufacturers. This finding was the relationship between the ROM and RAM capability of a microcontroller family which was found to be linear across the family. Figure 26, Figure 27 and Figure 28 are taken from submission three and show scatter plots of ROM versus RAM capability of PIC18, PIC16 and Atmel AVR respectively. The plots include trend lines showing that the general trend is linear with a positive gradient for all three microcontroller families. It should be noted that the PIC18 and AVR are 8-bit microcontrollers and have both low- and high-end variants. The PIC16 family is also 8-bit but only has low-end variants.

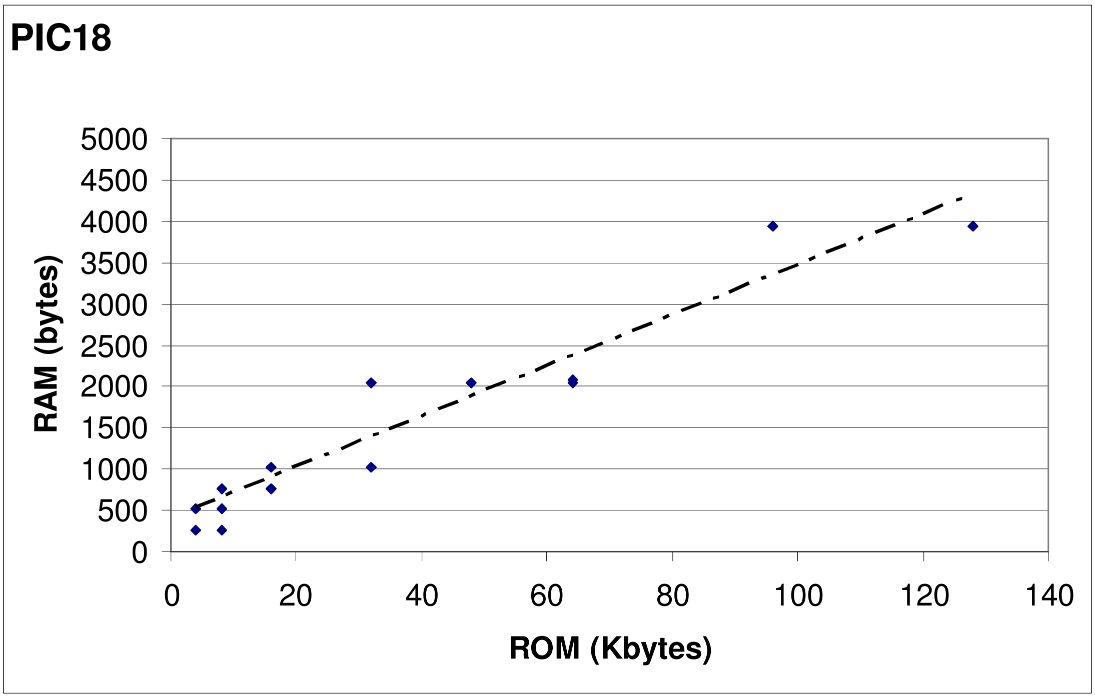


Figure 26: Plot of RAM Size versus ROM Size for the PIC18 Microcontroller

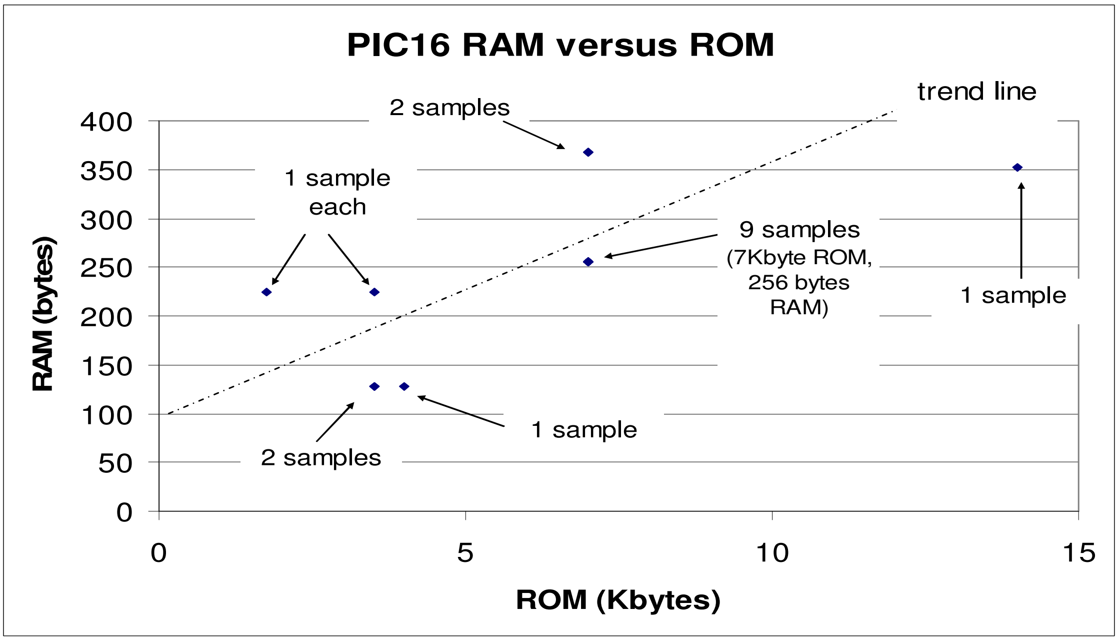


Figure 27: Plot of RAM Size versus ROM Size for the PIC16 Microcontroller

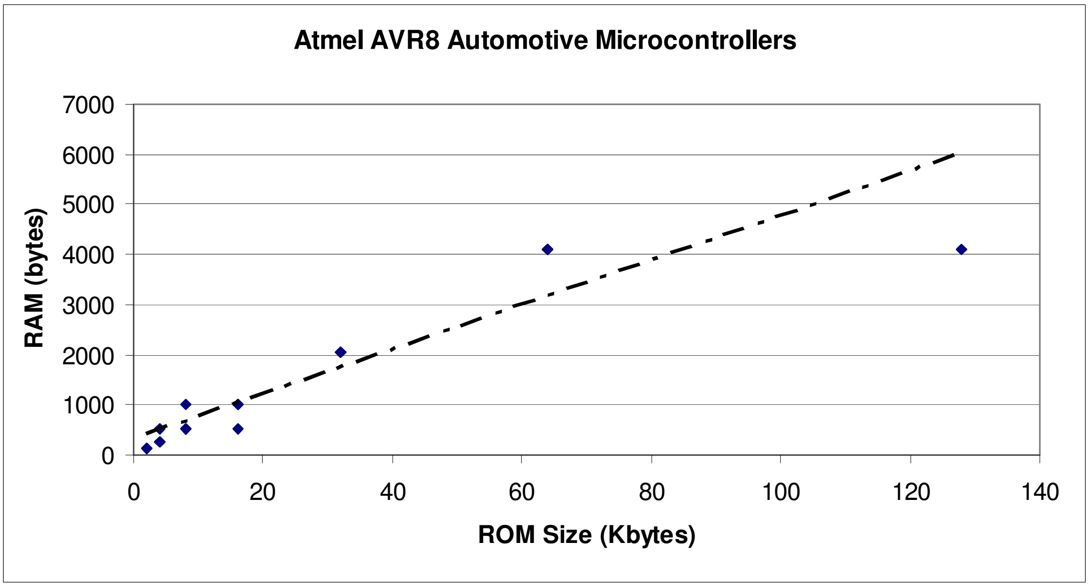


Figure 28: Plot of RAM Size versus ROM Size for the AVR Automotive Microcontrollers

Manufacturer	Family	RAM/ROM Gradient
Microchip	PIC18F	~ 43 bytes per Kbyte
Microchip	PIC16F	~37 bytes per Kbyte
Atmel	AVR	~64 bytes per Kbyte

Table 8 : Comparison of semiconductor manufacturer, microcontroller family and RAM / ROM gradient

Table 8 compares the three microcontroller families and their capability in bytes of RAM per Kbyte of ROM. This is important as it is a measure of the memory capacity of the microcontroller. This in turn shows how well a piece of embedded software can be targeted towards a particular microcontroller family because one way it can be characterised is in terms of its ROM/RAM requirements. In the case of LIN system design, this is particularly important for LIN slave devices for which embedded software has to be squeezed into microcontrollers with smaller memory to reduce cost. Table 8 shows that the Atmel AVR is the most capable in terms of RAM.

To understand the significance of ROM and RAM microcontroller capability on the cost of a LIN node or ECU, it is interesting to consider an example of targeting a LIN stack to real microcontroller variants. The embedded software of a typical LIN node will at minimum be a combination of LIN communications stack and application code. In submission four the memory requirements of the LIN stack exported from NetGen for the Atmel AVR was ascertained for a number of different LIN bus designs. Figure 29 compares the estimated ROM /RAM requirements of the LIN communications stack with five possible Atmel AVR microcontroller variants (called A, B, C, D and E in the figure 29). As the typical LIN node is a combination of LIN communications stack and application code, the ROM/RAM requirements of this combination will be at least that of the LIN communications stack and certainly for some LIN nodes this will be the most significant proportion of memory usage.

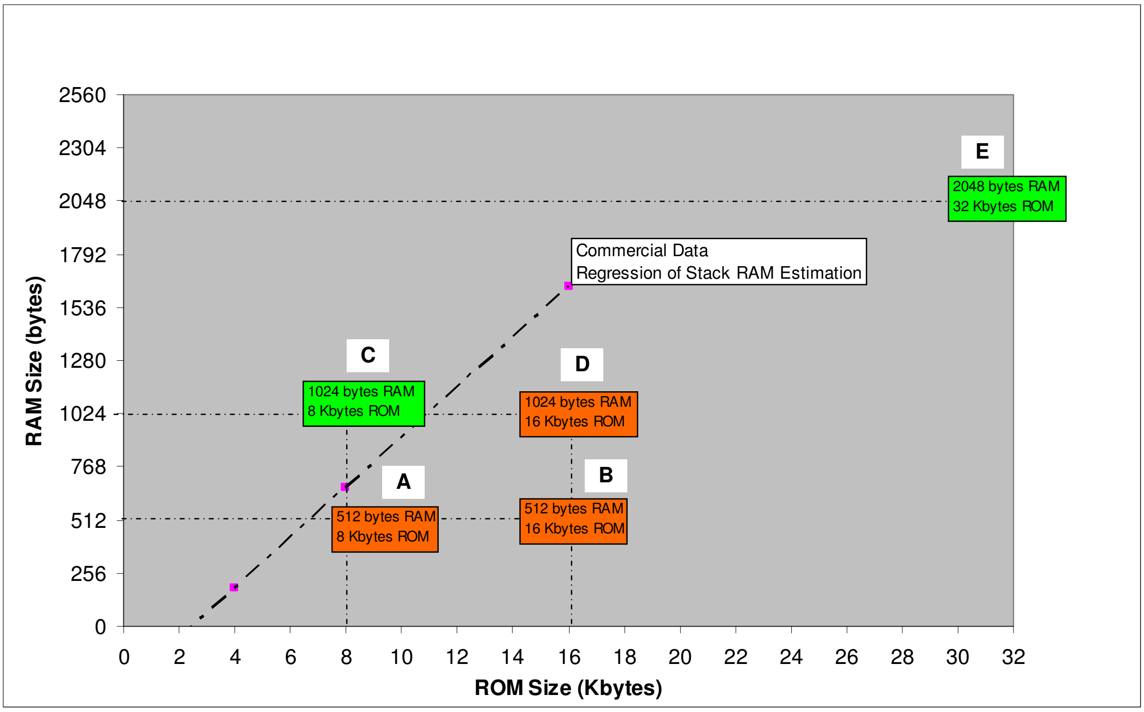


Figure 29: ROM versus RAM relationship of the Atmel AVR microcontroller and LIN communications stack with AVR variants shown. The Atmel AVR microcontroller are the points A, B, C, D and E on the figure.

The commercial LIN examples used in submission four had a stack size of between 5 and 9 Kbytes. For Atmel AVR microcontroller variant A which has 8 Kbytes ROM, it can be seen that the 512 bytes RAM capability would be insufficient for a typical LIN communications stack and application code combination that requires 8 Kbytes of ROM. In this situation, Atmel AVR microcontroller variant C which has more RAM space of a total of 1024 bytes would have to be selected. Figure 29 also indicates that Atmel AVR microcontroller variant B would be very unlikely to have sufficient RAM to support the LIN communications stack. Atmel AVR microcontroller variant D is more likely to be able to support the LIN communications stack with 16 Kbytes ROM and 1024 bytes RAM. However it is quite likely that if a significant amount of ROM and RAM is required (e.g. HVAC network in Appendix C of submission four which required over 8Kbytes of ROM and 700 bytes of RAM), the designer may end up having to choose Atmel AVR microcontroller variant E if the application code itself requires more than 300 bytes of RAM. This would leave nearly 24 Kbytes of ROM space free for the application code.

Understanding the ROM/RAM relationship between a target microcontroller and the intended LIN stack will help the LIN stack designer ascertain how to design for lower cost. In this case it means design for lower RAM usage by making software design decisions to move data to ROM if at all possible. This will reduce nodal cost by allowing selection of a lower cost variant (e.g. variant D instead of variant E in Figure 29), as it has been shown in submission three that microcontrollers with lower memory capacity were of lower cost.

5.2. Estimation of LIN Communications Stack ROM and RAM Requirements

Submission four looked into the feasibility of estimating the ROM and RAM requirements of a LIN communications stack for the Atmel AVR microcontroller using LIN bus characteristics as model inputs. These factors were the number of *Nodes*, *Schedules*, *Messages* and *Signals*. The stack used was the one that could be exported as a C-code from the NetGen LIN tool. An analysis of publicly available LIN communication stack ROM and RAM requirements performance data showed that there is a linear relationship between the LIN bus properties and the memory requirements. However it also showed that this data was not sufficient to be able to produce regression models for the estimation of the ROM and RAM requirements of a LIN communications stack.

A factorial experiment design was used for the collection of a *Model Development* dataset with appropriate coverage of the data space for development of appropriate regression models. In addition to this, commercial examples of LIN bus applications were collected to form a *Model Assessment* dataset so that they could be used to assess the regression models. For both of these datasets, the NetGen tool was used to automatically generate the C language source code library and the commercially available IAR Systems Atmel AVR C compiler was used to compile the code for the stack. The map file generated by the compiler contained the ROM and RAM requirements. A number of regression models were developed for the estimation of ROM and RAM from LIN bus properties such as the number of nodes, signals, messages and schedules. Regression models were also developed for the ROM versus RAM relationships of both the LIN communications stack for the Atmel AVR microcontroller and for the Atmel AVR automotive microcontroller product family itself.

It was shown that regression models could be developed for the estimation of the ROM and RAM requirements of the LIN communications stack using LIN bus characteristics as inputs to the models. The regression models for estimation of ROM/RAM based on all LIN bus factors performed well (e.g. with factors representing the number of *Nodes*, *Schedules*, *Messages* and *Signals*). Figure 30 shows the performance of the five factor ROM model by plotting actual and estimated ROM. The diagonal line shows the position when both agree. It can be seen that the model performed well although it did estimate slightly lower than the actual.

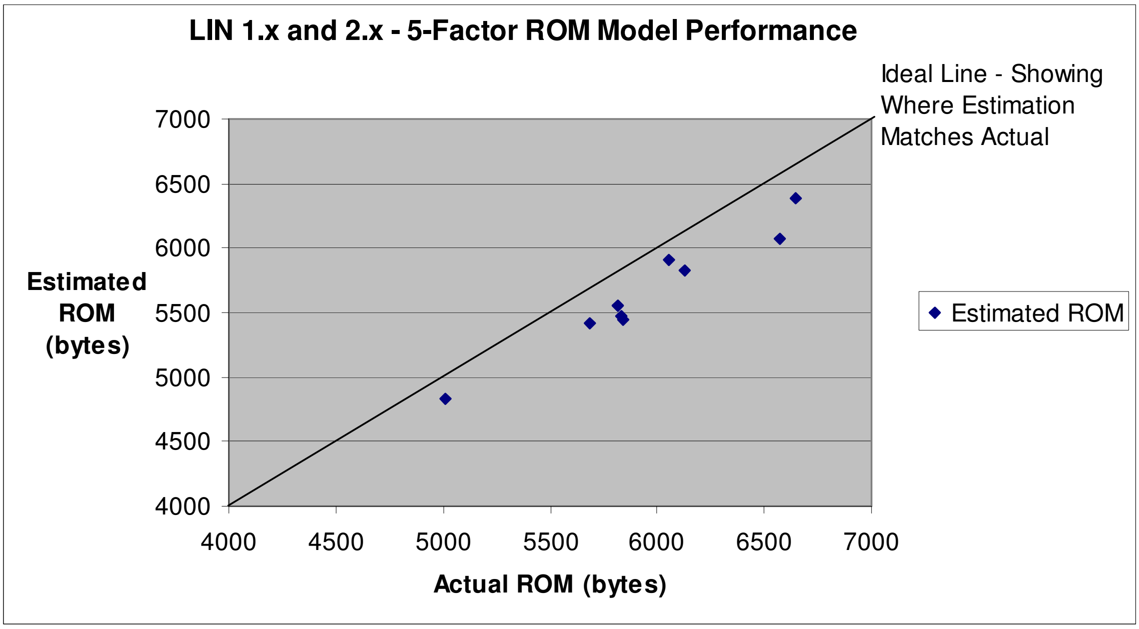


Figure 30: Performance of 5-factor model with *Model Assessment* dataset

The factors *Signals* and *Nodes* are those that are known at the very beginning of the project during which a number of candidate architectures are assessed. Regression models making an estimation based on just these two factors alone did not perform so well. However it was found that a two factor regression model based on *Messages* and *Signals* could be used to make a reasonable estimation of the ROM and RAM requirements. In addition to this, by looking at the typical commercial LIN application examples, the typical number of messages per node could be ascertained and then used to make a two factor *Messages* / *Signals* based estimation of ROM and RAM.

Table 9 and Table 10 compare the performance of each of the models. It can be seen that although the ROM models have a greater model error in terms of bytes, their performance is actually more accurate as a percentage of the mean value of commercial production vehicle examples. The estimation of the LIN communications stack ROM requirements appeared to be the most useful as this has been shown to be possible with an error of just a few percent. Models for the estimation of RAM had a much larger error (i.e. at least 18%) but could still be useful for making a coarse estimation of a LIN communications stack RAM requirements and therefore helping the selection of the most appropriate microcontroller variant. It was also shown that ROM and RAM estimation based on simply the number of nodes and signals is possible without significant loss of accuracy when compared to using all possible regression model factors.

Model	**RMS Model Error**	**Error as % Mean commercial value (bytes)**
5-Factor Model	376.20	6.1%
3-Factors Model (Number of Nodes and Signals information)	581.33	9.4%
2-Factors Model (*Messages* & 8-bit Signals)	329.57	5.3%
2-Factors Model (the factor *Messages* estimated from *Nodes*)	456.09	7.4%

Table 9: Comparison of RMS error of models in context of mean commercial ROM

Model	**RMS Model Error**	**Error as % of Mean commercial value (bytes)**
5-Factor Model	83.00	18.7%
3-Factors Model (Number of Nodes and Signals information)	149.67	33.7%
2-Factors Model (*Messages* & 8-bit Signals)	98.20	22.1%
2-Factors Model (the factor *Messages* estimated from *Nodes*)	94.72	21.3%

Table 10: Selection of the RAM model that overestimates with the lowest error

The models for ROM and RAM estimation that were developed in submission four are only directly relevant for the NetGen Atmel AVR LIN stack used in the analysis, the target AVR microcontroller and compiler used. A change in any one of these will almost certainly change the characteristics of the model required for estimation. However the results may be representative of other 8-bit microcontrollers. The generic process for model development is shown in Figure 31.

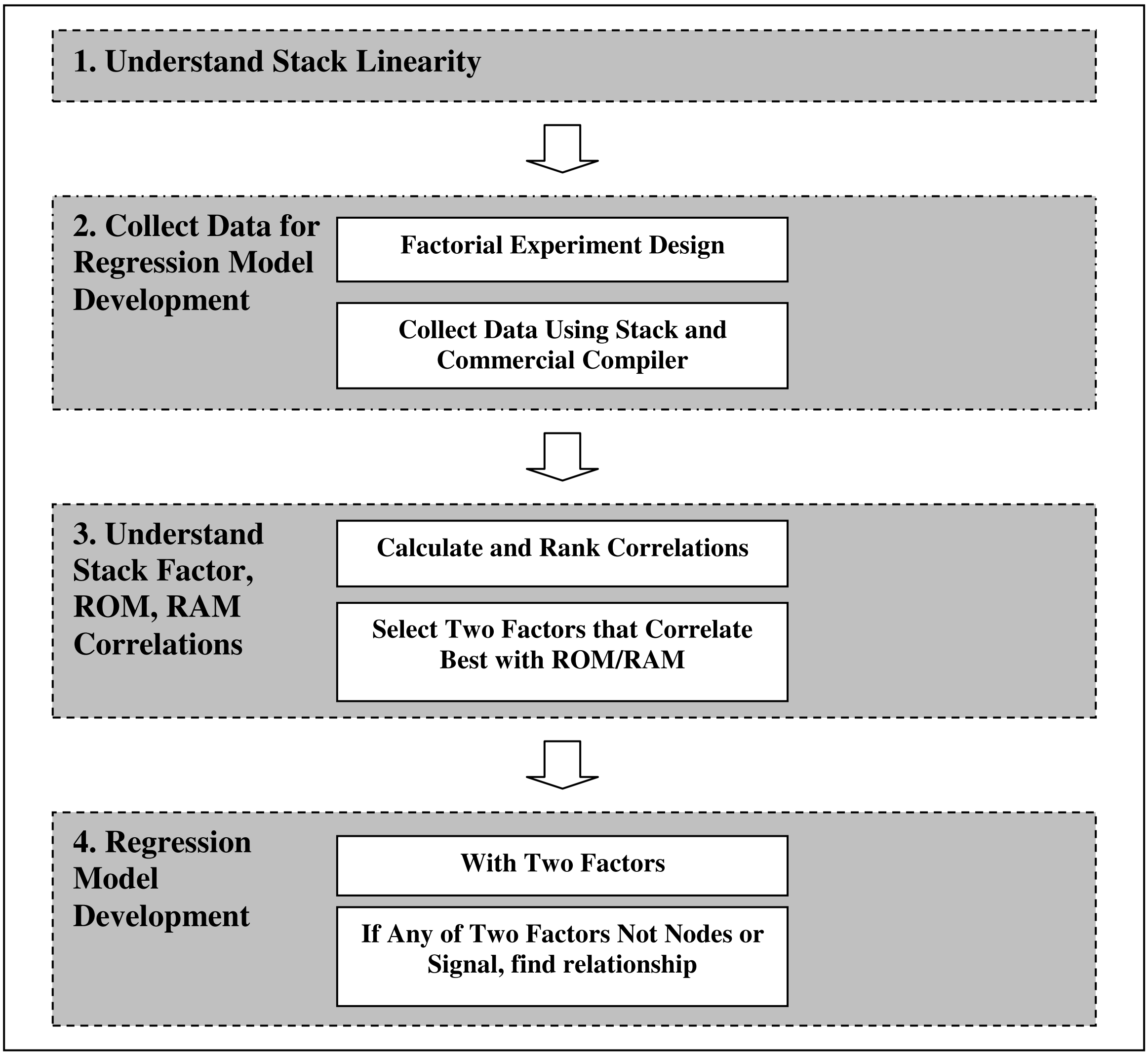

1. Understand Stack Linearity
2. Collect Data for Regression Model Development
Factorial Experiment Design
Collect Data Using Stack and Commercial Compiler
3. Understand Stack Factor, ROM, RAM Correlations
Calculate and Rank Correlations
Select Two Factors that Correlate Best with ROM/RAM
4. Regression Model Development
With Two Factors
If Any of Two Factors Not Nodes or Signal, find relationship

Figure 31: LIN communications stack ROM, RAM modelling process

This process for developing ROM and RAM requirement estimation models for a new LIN communications stack can be broken into the following four generic stages:

1. **Understand Stack Linearity** – The process shown in this report is dependent upon the relationship between LIN communication stack ROM and RAM requirements having a linear relationship between the number of nodes, schedules, messages and signals.

2. **Collect Data for Regression Model Development** – A factorial experiment design was used for the collection of the minimal and appropriate dataset to carry out regression analysis and therefore develop the appropriate models for ROM and RAM estimation.
3. **Understand Stack Factor, ROM and RAM Correlations** – By looking at how each of the LIN communications stack factors correlate with both ROM and RAM, the most appropriate factors can be selected for regression model development. The correlation coefficient can be obtained by carrying out a single variable linear regression or using the statistical features of a computer program such as Excel. The ideal situation is to be able to develop regression models for the estimation of ROM and RAM from only information concerning the number of LIN nodes and signals between the nodes. For the particular LIN communications stack that was the subject of this study, the number of signal and messages correlated the best and therefore the number of messages per node for typical commercial applications was estimated using the *Model Assessment* dataset. Then by knowing the number of nodes, the number of messages could be used as an input into a 2-factor *Messages / 8-bit Signals* ROM estimation regression model.
4. **Regression Model Development** – Carry out regression analysis with the data collected. It is ideal to have a dataset containing real LIN applications with which to be able to assess the regression models and select the best performers.

5.3. Microcontroller Family ROM/RAM Capability Assessment Process

The microcontroller capability assessment process is important to a LIN communications stack supplier if they wish to ascertain how best to design for a particular microcontroller family, or a semiconductor manufacturer who wants to select the amount of memory required in certain microcontroller variants, or finally a vehicle manufacturer or first tier supplier who wants to select stack and microcontroller suppliers for their particular project. If the amount of RAM per Kbyte of ROM is not enough, this may lead to the selection of a microcontroller with a lot of ROM just to ensure that there is enough RAM sufficient for the LIN communications stack. This leads to extra expense by necessitating the selection of a microcontroller variant with sufficient RAM. Therefore the benefit of using the process described in this section is to be able to design a stack and microcontroller together so that they provide the lowest possible nodal cost, which should be a key advantage of using LIN in automotive applications.

To carry out the assessment the designer must obtain details on the relationship between ROM and RAM capability for the microcontroller and ROM and RAM requirements of the embedded LIN communications stack. The suggested metric is number of bytes of RAM per Kbyte of ROM which represents the gradient of the ROM/RAM relationship. The gradient of ROM /RAM for both microcontroller and LIN stack is never going to be 100% accurate but it can give a good estimation of how much leeway a designer will have. Therefore the process provides an additional software metric that can be used along with microcontroller unit cost to help in the selection of stack and microcontroller for a project.

Before the ROM/RAM capability of a microcontroller can be assessed, the LIN communications stack ROM and RAM requirements must be ascertained. The generic process for ascertaining this requirement is shown in Figure 32.

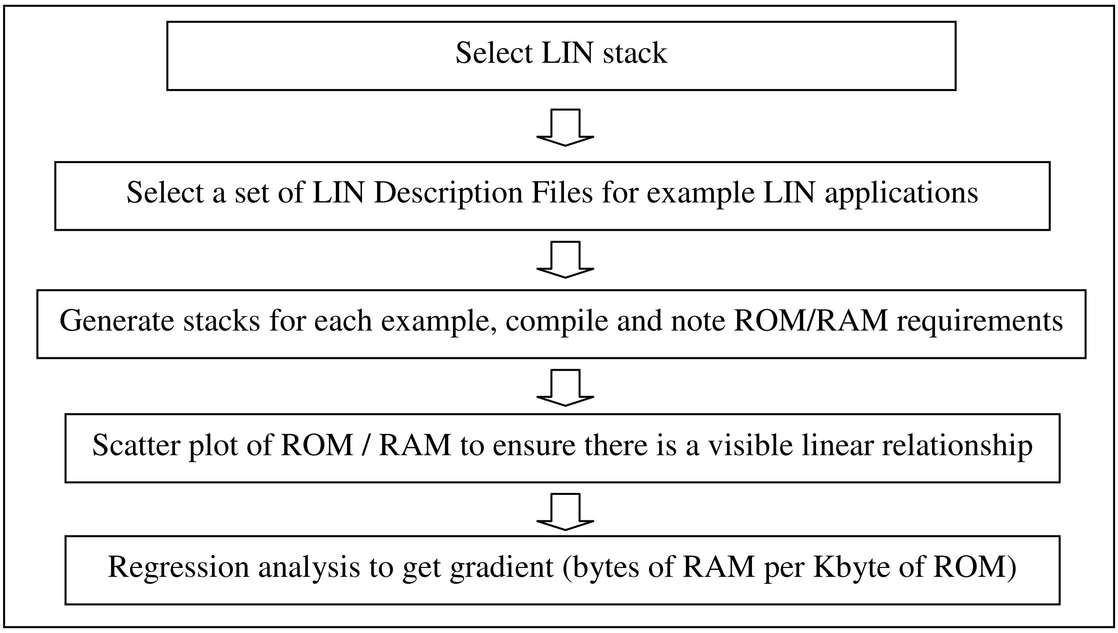

Select LIN stack
Select a set of LIN Description Files for example LIN applications
Generate stacks for each example, compile and note ROM/RAM requirements
Scatter plot of ROM / RAM to ensure there is a visible linear relationship
Regression analysis to get gradient (bytes of RAM per Kbyte of ROM)

Figure 32: Process for ascertaining LIN stack ROM/RAM requirements

Figure 33 shows the generic process for microcontroller memory capability assessment. Its first step is to select a set of microcontroller families to assess. In submission three, there were three families; Microchip PIC16, PIC18 and Atmel AVR. The second step requires the ROM and RAM capability data of each variant be obtained. In the third step the data is plotted and the gradient for each family is obtained by regression analysis. This is represented by the metric number of bytes of RAM per Kbyte of ROM. The fourth step is where the comparison is made between each of the families to understand which microcontroller family is the most capable. In the fifth and final step, a comparison is made with the LIN communications stack ROM and RAM requirements. The result will help the LIN stack designer ascertain whether they need to focus on a reduction of RAM usage by the stack. Alternatively, a microcontroller manufacturer can use this process to ascertain whether they need to provide more RAM in their microcontroller family.

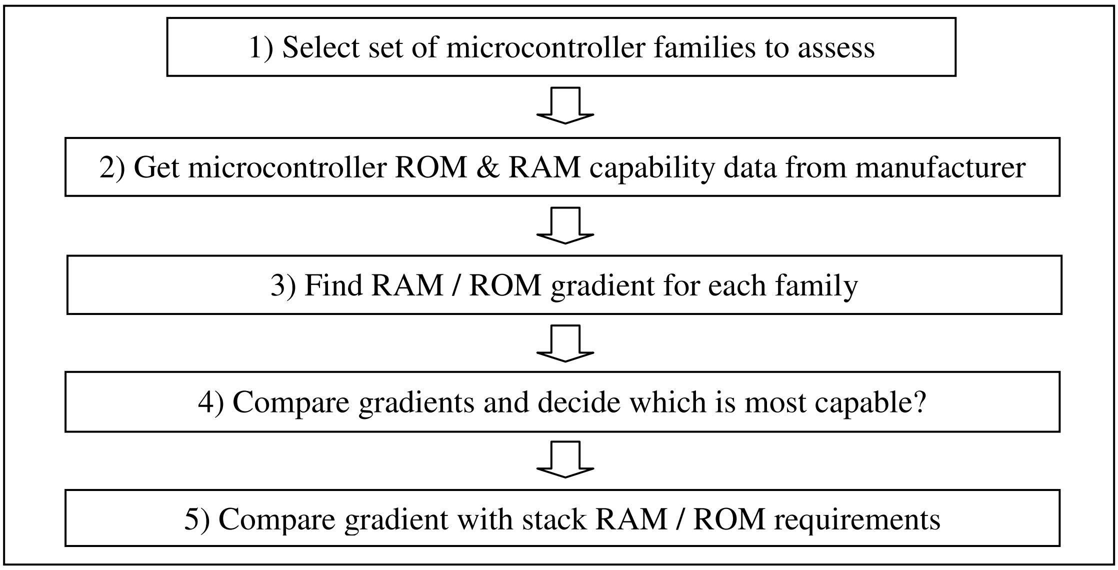

1) Select set of microcontroller families to assess
2) Get microcontroller ROM & RAM capability data from manufacturer
3) Find RAM / ROM gradient for each family
4) Compare gradients and decide which is most capable?
5) Compare gradient with stack RAM / ROM requirements

Figure 33: Process for comparing microcontroller and stack memory capability

This process is demonstrated by the work which was carried out in submission four. In submission four, the ROM and RAM relationship of the Atmel AVR and LIN communications stack was found by using a set of LIN Description Files describing the characteristics of commercial LIN designs. These were used to generate a LIN stack which was then compiled and the ROM/RAM requirements were noted. The ROM / RAM values were viewed in a scatter plot as shown in Figure 34 which has a trend line that represents a requirement of approximately 75 bytes per Kbyte.

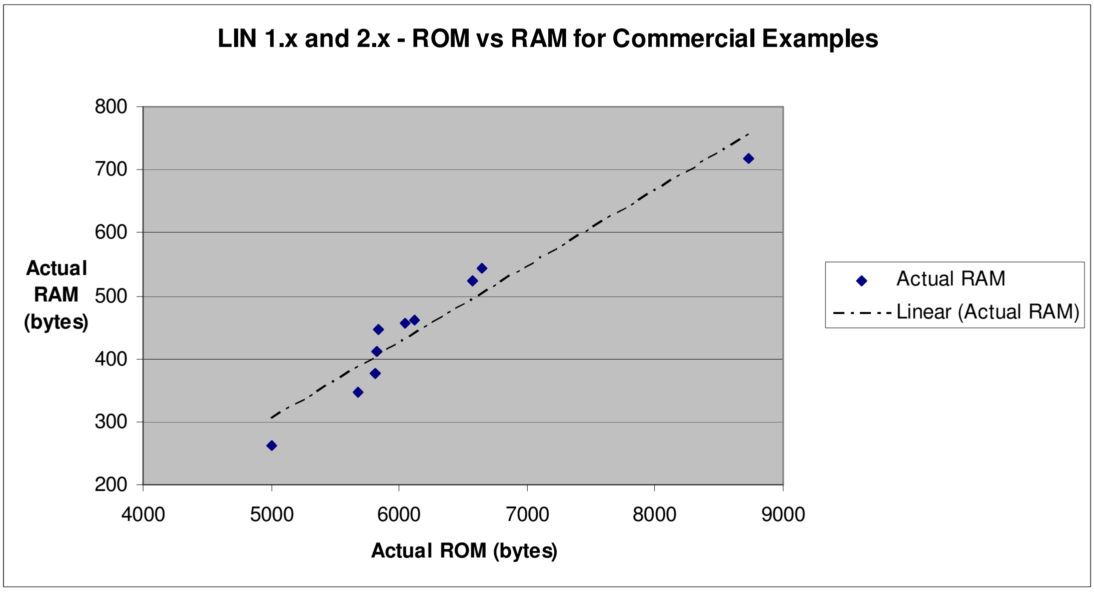


Figure 34: ROM and RAM values of the examples in the *Model Assessment* dataset

Figure 35 shows a comparison between microcontroller ROM/RAM capability of the Atmel AVR family and LIN communications stack requirements using two different datasets of LIN example applications. Both of these datasets were used in submission four for the modelling of LIN stack memory requirements. The first of these datasets is the one used for *Model Development* (or model training), the second was for *Model Assessment* and contains the features of real examples from a number of production vehicles.

In Figure 35 it can be seen that the gradient of the microcontroller ROM and RAM relationship is lower than the two estimations of the LIN communications stack ROM and RAM requirements, therefore indicating that the Atmel AVR microcontroller may not have the optimal ROM and RAM relationship for this particular LIN communications stack. The LIN stack requirement that was made with commercial production vehicle examples is the lowest of the two. This gives a requirement of approximately 75 bytes RAM per Kbyte of ROM which is above the capabilities of all microcontrollers in Table 8.

Figure 35 also shows that for the mean LIN communications stack ROM size (as found from the commercial examples) of 6162 bytes, the Atmel AVR certainly does not have enough RAM (~500 bytes is required). Of course the variants of this microcontroller come in 128, 256, 512 and 1024 byte variants and the real outcome of this analysis is that a variant of 512 would probably be insufficient for a stack of around 500 bytes plus whatever is required for the LIN application software. Therefore a variant with 1024 bytes of RAM would probably be the minimum level of selection for this particular LIN communications stack. It should be noted that 718 bytes is the maximum example RAM size from the *Model Assessment* dataset which contains commercial production vehicle examples.

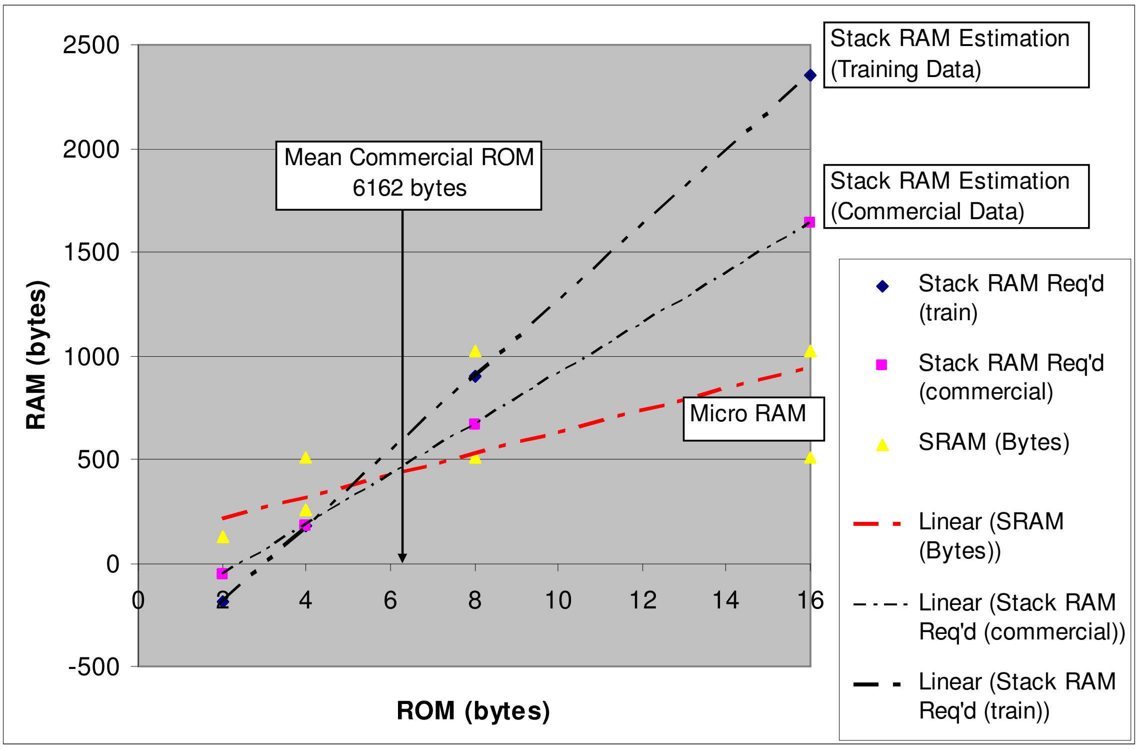


Figure 35: ROM and RAM relationship of the Atmel AVR and LIN stack

5.4. Microcontroller Variant Selection By ROM/RAM Requirement Estimation

The aim of this process is to select a microcontroller variant from a family of microcontrollers and ascertain the impact of pricing on architecture cost. It would be expected that a number of candidate architectures would have to be evaluated and for each of the candidate architectures the ROM and RAM requirements of the LIN communications stack would differ. As the ROM and RAM requirements would differ, then so would the expected pricing between architectures. Therefore the affect of microcontroller pricing on nodal cost between architectures can be ascertained. This process was primarily developed in the work carried out in submissions four and five. The assumption is that to get the most benefit of the economies of scale, the same microcontroller variant will be used for each node within a single LIN candidate architecture and also each ECU will be physically identical. The only difference between each ECU will be the software that is embedded within them.

The process would typically be carried out by the architecture owner such as the sub-system supplier in the case of a LIN system. This would therefore require either the semiconductor manufacturer or LIN communications stack supplier to provide the ROM and RAM estimation models for the architecture owner to use. The process for selecting a microcontroller variant is shown in Figure 36. It can be seen that this process is broken into a number of stages.

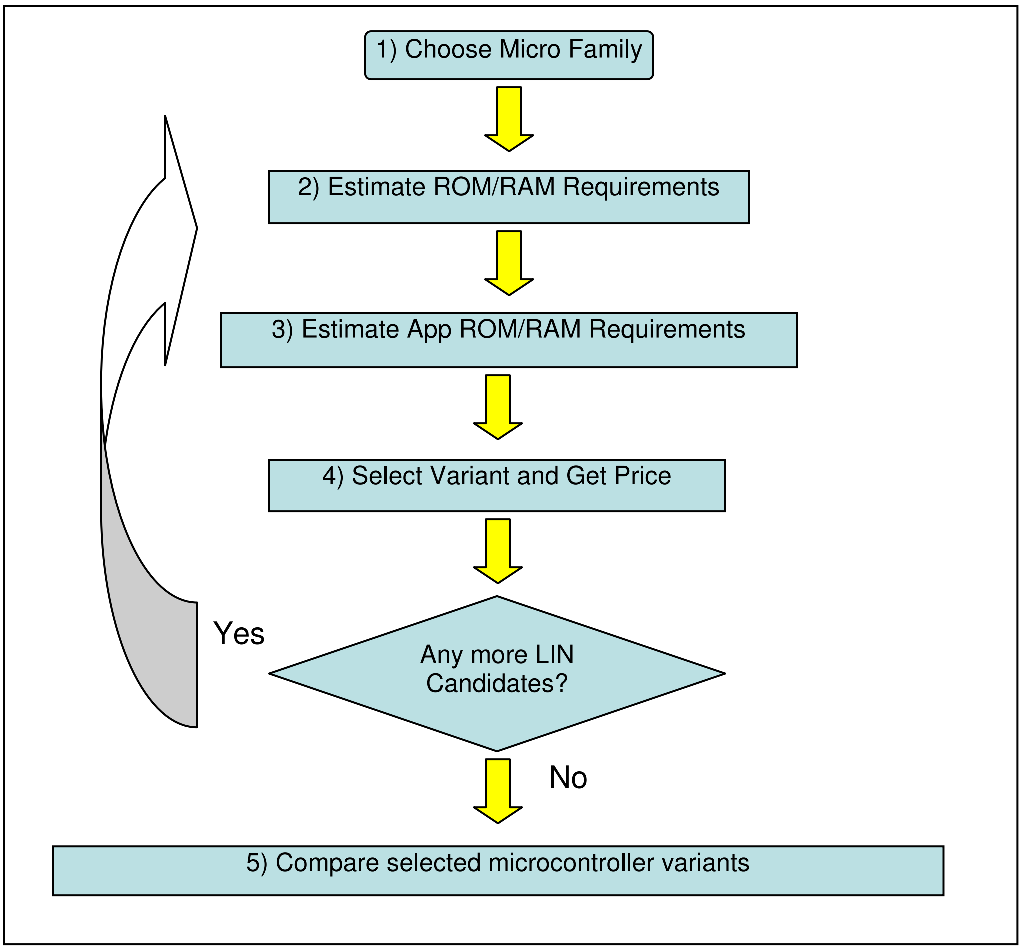

1) Choose Micro Family
2) Estimate ROM/RAM Requirements
3) Estimate App ROM/RAM Requirements
4) Select Variant and Get Price
Yes
Any more LIN Candidates?
No
5) Compare selected microcontroller variants

Figure 36: Process for microcontroller selection and nodal cost estimation

1) Choose Microcontroller Family

The target microcontroller family is chosen which is to be used in the remainder of the study. One of the factors in choice of microcontroller family is the suitability for a LIN communications stack.

Output from this stage: Microcontroller family chosen.

2) Estimate ROM/RAM Requirement

During this stage the ROM and RAM requirements for the LIN communications stack of each node in each of the LIN candidates is estimated from the knowledge of the number of nodes and signals. The 8-bit Atmel AVR models from submission four are used in this stage for 8-bit microcontrollers under the assumption that the results will be equivalent for all 8-bit microcontroller architectures. If a different size CPU microcontroller is the target, then the Atmel AVR models will be less representative.

Output from this stage: The ROM and RAM requirements for each LIN node within each of the candidate architectures.

3) Estimate ROM/RAM Requirement of Application Software

This is a new part of the process in which the ROM and RAM requirements for the application software of each of the nodes was estimated. The application code may contain digital switch processing, Pulse Width Modulation (PWM) and pulse input processing. One possible estimation method is to count each of these features in order to make an estimation. Another method is to look at historical data for equivalent application software and use these values.

Output from this stage: Application code ROM and RAM requirement.

4) Select Variant and Get Price

This is a new part of the process. First the appropriate microcontroller is selected based on its ability to support the I/O and also the ROM and RAM requirements. Quotations are obtained for the microcontroller in different production quantities.

5) Compare Selected Microcontroller Variants

For each of the candidate architectures, the selected microcontroller variant is compared with that of each other candidate architecture and the impact on the cost of the architecture is ascertained.

6. SPORTSCAR BODY CONTROL ELECTRICAL ARCHITECTURE DESIGN – CASE STUDY

6.1. Case Study – Overview

Introduction

Submissions two, three and four describe an investigation for a customer who is a wiring harness manufacturer. The results from these submissions demonstrated to the customer that it was extremely challenging to justify the adoption of LIN over hardwired integration on a cost basis alone. A number of processes were developed in the submissions for architecture design-to-cost, and microcontroller assessment and selection. The design-to-cost process was used to make a comparison of the cost of two architectures and helped estimate a LIN node target cost needed for the new architecture to be lower cost than the original. The process for microcontroller selection was concerned with the estimation of the ROM/RAM requirements of a LIN stack to help ascertain which microcontroller variant to use and also to provide a way of assessing the ROM/RAM capability of a microcontroller family. The results also had an impact on the NetGen LIN communications stack product by highlighting certain deficiencies with the version that was used in the study. The key one was that a redesign was required to reduce its ROM and RAM requirements, which in turn would provide a commercial advantage due to being able to fit into lower memory (and therefore lower cost) microcontroller variants. In submission five the processes were brought together, explored and refined in a new case study, which is described here in this chapter.

The aim of the case study reported in submission five was to investigate whether an alternative architecture based on LIN technology could reduce the cost and weight of a niche sportscar body control wiring harness. The original harness was based on hardwired integration and the study explored whether there was a cost and weight benefit from the adoption of LIN. A second motivation for this report was also to provide a new case study with which to assess the processes developed within this Engineering Doctorate programme.

The case study brought in an additional target of architecture weight in addition to cost. Therefore the design-to-cost process was adapted to provide a design for low harness weight process since low harness weight was a key requirement of the sportscar body control harness design. This process was described in chapter four. The processes that were developed were used to assess six LIN alternative architectures and ascertain which of them would provide a lower cost and lower weight alternative to the hardwired original harness.

The processes were then assessed by adding further detail to the designs such as the actual wire lengths that would be used if the LIN nodes were assumed to be placed at certain locations and also assessing the cost of the bill of materials to manufacture the new LIN nodes. The assessment did not involve an entire harness design to compare with the original estimations. This would require a large amount of time and a project budget for a full electrical architecture design to be carried out. However the level of detail in the assessment of the processes was sufficient to show the key benefit of the processes used and to learn lessons on their best use.

The benefit of the processes was that they gave an assessment of cost and weight saving potential for each of the candidate architectures and this was achieved by the use of coarse information. This means that such an assessment can be carried out with the kind of minimal information that is only available at the start of an electrical architecture design project and therefore not requiring the expense of a full design.

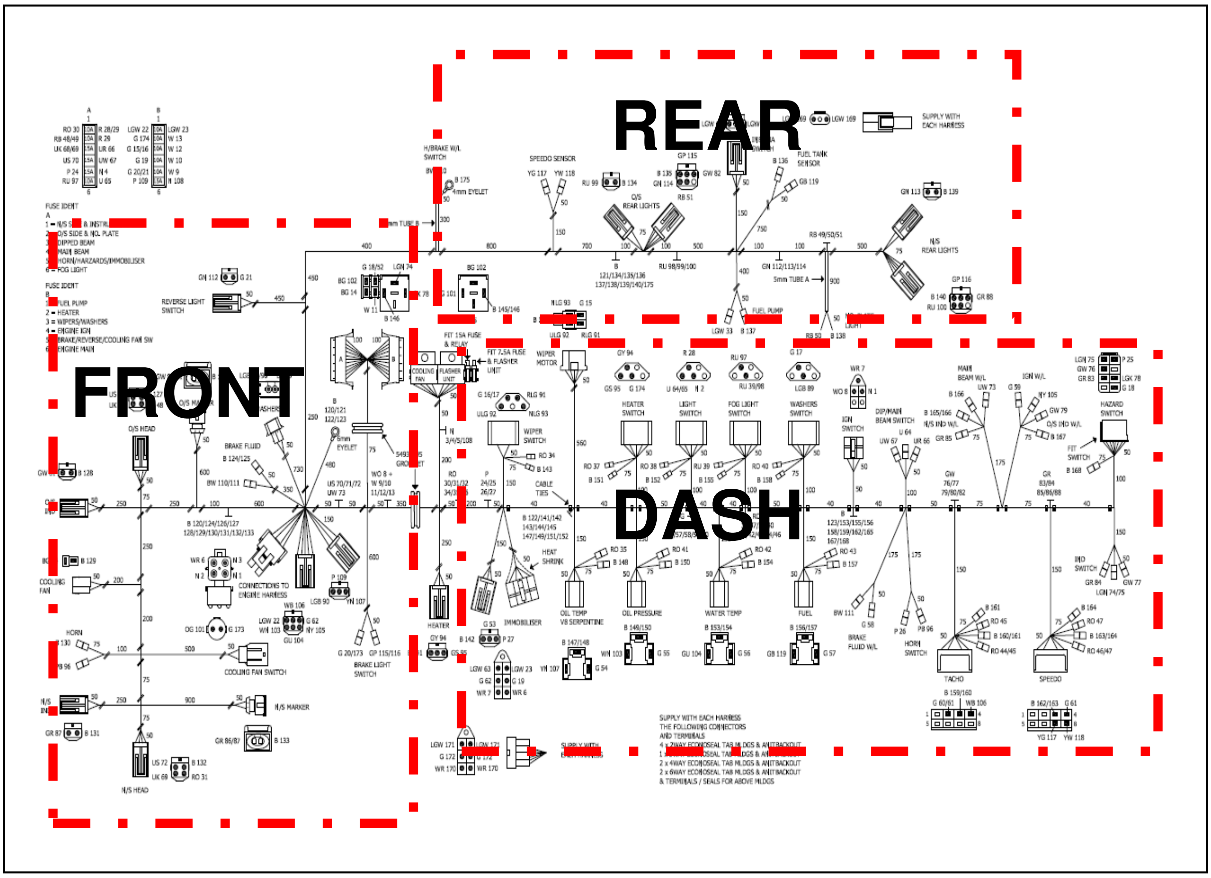


Figure 37: Three zones of the sportscar body control wiring harness (Westfield Sportscars Ltd., 2009).

Analysis of Original Electrical Architecture

A schematic of the original body control harness of the sportscar that was the subject of the study is shown in Figure 37. An analysis of the signal routing from origin (e.g. from a switch) to destination (e.g. to a lamp), showed that there were three main zones in the architecture; *Front*, *Dash* and *Rear*. The schematic contained information on wire length and the gauge was obtained from the wiring harness manufacturer. Wire cost was supplied by the wiring harness manufacturer and used to calculate all of the wire costs. This information was used to develop wire cost and weight models for inter-zone and intra-zone signal wires. For example, for cost the following equation was used:

$$\text{Wire Cost} = (\textit{Wire} + \textit{Wire Component}) \times \textit{Non-Material Cost} \tag{9}$$

Where

Wire is the wire cost

Wire Component is the cost of components such as connectors, pins, terminals etc.

Non-Material Cost factor accounts for all non-material piece costs associated with the manufacture of the harness, and includes items such as fixed business overheads, variable costs such as manufacturing labour, logistical costs and profit margin.

The full processes and equations for cost and weight were described in chapter four. The use of the equations resulted in the range of the values of *inter-zone* and *intra-zone* wire cost and weight. The range of values for cost is shown in Table 11.

Wire Type	**Cost (based on 1.6 *Non-Material Cost* Factor)**	**Cost (based on 1.8 *Non-Material Cost* Factor)**
Typical Inter-Zone Wire Cost	£0.90	£1.01
Typical Intra-Zone Wire Cost	£0.40	£0.45

Table 11: Combined wire and component costs

Proposed LIN Node

Figure 38 shows the general electrical architecture of the LIN nodes that were proposed for this case study. To reduce implementation costs, the Printed Circuit Board (PCB) is potted in resin for this proposal. This has the advantage of eliminating the need for a costly ECU housing which could cost in the region of £10 to £20 for this type of application and production volume. This additional casing cost would significantly reduce the likelihood of the LIN architecture having a cost advantage over the hardwired target. Therefore this is a low cost and low weight solution.

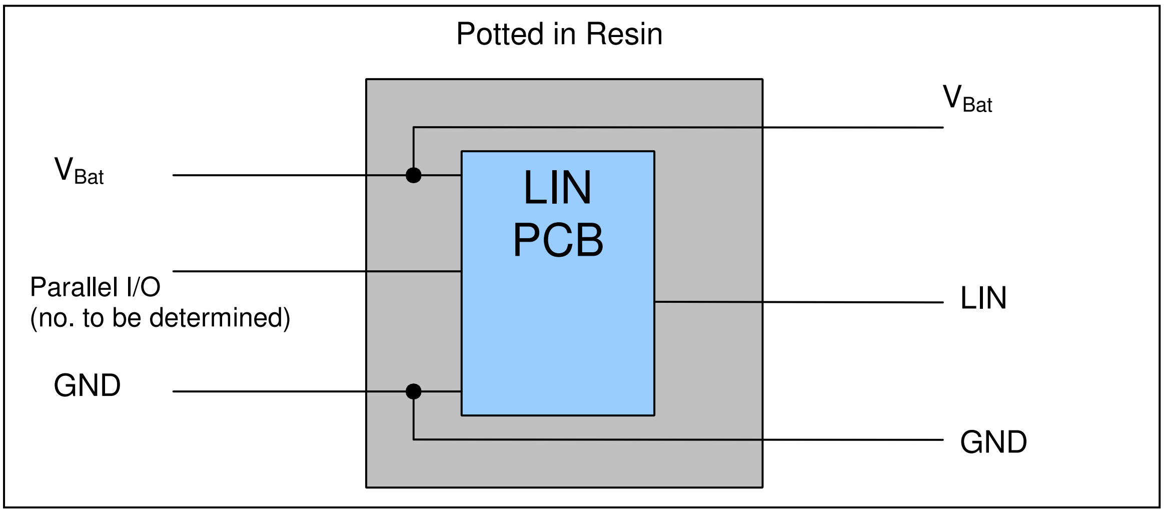


Figure 38 : Generic configuration of LIN node considered in all candidate architectures

Proposed Architectures

Six LIN architecture candidates were proposed as alternatives to the original hardwired architecture and were compared in terms of their estimated cost and weight. Two types of architectures were considered:

1. Ones with inline relays – the benefit of this approach is that nodal cost is kept to a minimum and off-the-shelf automotive relays are included inline.

2. Using nodes with quad low side driver chips – the benefit of this approach is that the number of intra-zone wires is minimised (i.e. does not require extra wires and connectors to go between the relay and the actuator or lamp).

Figure 39 shows an example of the linear bus architecture with one master and two slaves. It has already been stated that there were two types of architecture which differ due to the type of LIN node deployed; inline relay-based and nodes with low side drivers. For each of these architecture types, three different architectures were considered due to the fact that there were three different zones within the target harness as was shown in Figure 37:

- **two nodes** (based on the reasoning that the original harness has most signals between *Front* and *Dash* zones), thus leaving the *Rear* zone signals hardwired.
- **three nodes** (based on the reasoning that the original harness has three zone – *Front*, *Rear* and *Dash*)
- **five nodes** (based on the reasoning that the *Front* and *Rear* zones can be further split into two zones each for offside and nearside).

Therefore there was a total of six candidate LIN architectures to be assessed as lower cost or weight alternatives.

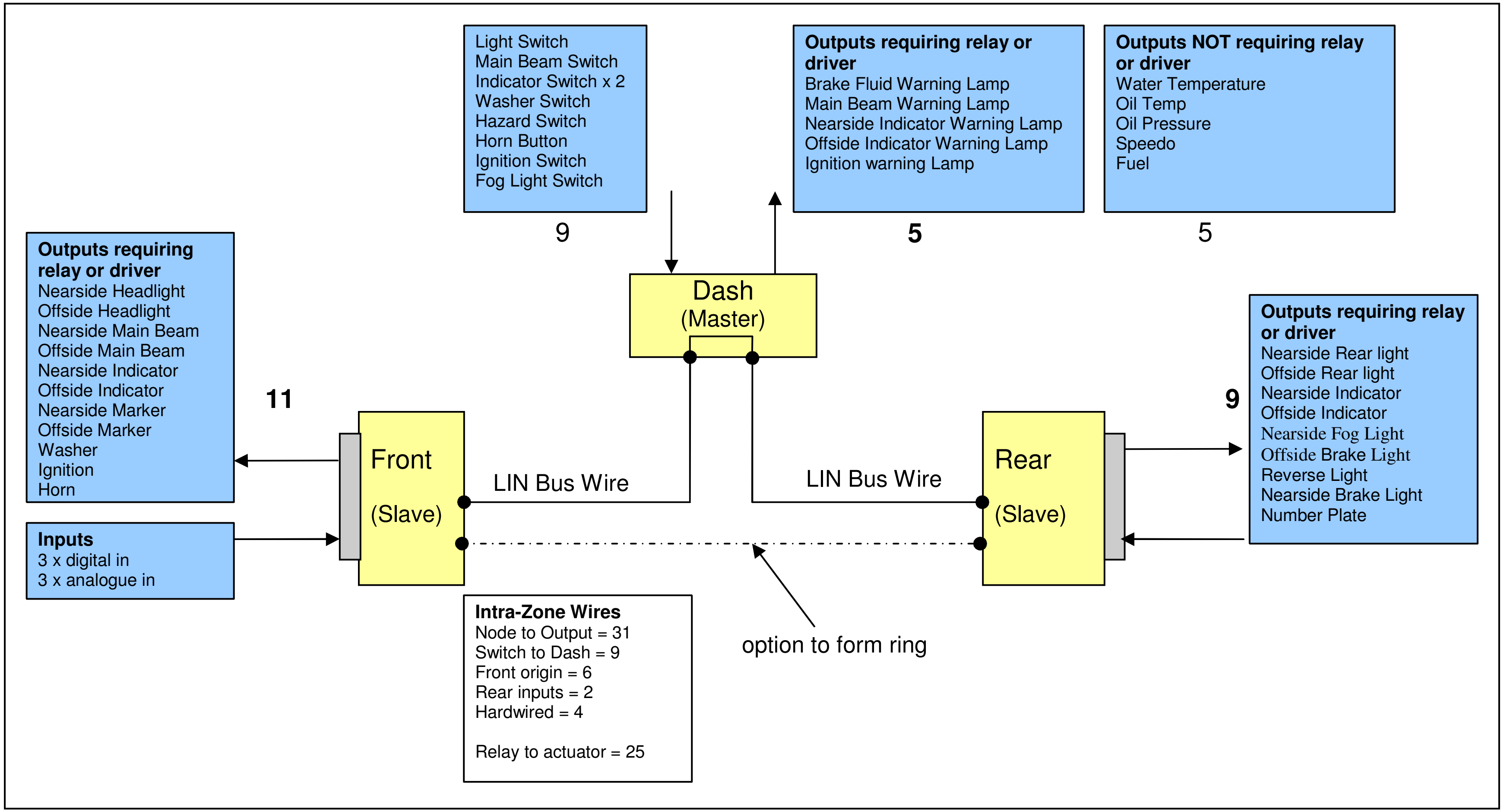


Figure 39 : Example of a LIN linear bus architecture

Assignment of Signals to Nodes

It was established that there were thirty five signals in total within the hardwired architecture. Four of these were to remain as hardwired signals and not be replaced with LIN equivalents. Therefore thirty one signals were replaced by LIN equivalent signals.

- Five of these were intra-zone signals but put onto LIN as they were associated with other signals and therefore adding to LIN was sensible (e.g. indicator offside warning lamp is associated with the indicator offside signal)
- Twenty six were inter-zone signals

Figure 39 shows an example three node LIN architecture with one master, two slaves and the signals assigned to each node.

Signal Substitution Analysis

Signal substitution analysis was carried out to ascertain which architecture results in the removal of the most wires and to compare this with the hardwired original. Table 12 compares these for each of the architectures. It can be seen from the table that for all LIN candidates, the number of inter-zone wires has decreased when compared with the hardwired architecture. However the number of intra-zone wires has increased for all LIN candidates due to inter-zone wires in the hardwired architecture being replaced with sensor to node and node to actuator intra-zone wires. The total number of wires has increased from LIN adoption due to the intra-zone wires but this does not necessarily mean that the LIN architecture candidates are more expensive or heavier. Inter-zone wires have a mean length higher than the mean values of the intra-zone wires and therefore there is more of a cost saving to be enjoyed from the removal of inter-zone wires.

Architecture	**Inter-Zone Signal Wires**	**Intra-Zone Signal Wires**	**LIN Nodes**
Target Hardwired	26	9	0
2 nodes – inline relays	10	60	2
3 nodes – inline relays	2	78	3
5 nodes – inline relays	4	78	5
2 nodes – Lowside Drivers	10	35	2
3 nodes – Lowside Drivers	2	52	3
5 nodes – Lowside Drivers	4	52	5

Table 12: Comparison of the number of wires, wire types and LIN nodes between architectures – forms coarse architecture equations

Table 12 also suggests that the inline relay-based LIN architectures can be removed from any further analysis as they will result in higher costs than low-side driver based ones. The inline relay and low-side driver LIN architectures have the same number of inter-zone wires. However, when intra-zone wires are considered, it can be seen that the low-side driver architectures have a lower number of intra-zone wires. For example, if both two node architectures are compared, the inline relay architecture has 60 intra-zone wires whereas the low-side driver equivalent has only 35 intra-zone wires. This is due to additional wires and connectors required to go from the inline relay to the associated actuator or lamp. Therefore further investigations were only carried out with the low-side driver based architectures.

The signal substitution analysis provides architecture cost equations. The cost of the hardwired architecture is given by the following equation:

$$C_{\text{Architecture}} = x1.C_{\text{InterWire}} + x2.C_{\text{IntraWire}} + C_{\text{Fixed}} \tag{10}$$

Where

$C_{Architecture}$ is the cost of the architecture

$C_{InterWire}$ is the cost of an inter-zone wire

$C_{IntraWire}$ is the cost of an intra-zone wire

C_{Fixed} represents the fixed harness costs

and the coefficients *x1* and *x2* are the number of wires

Extended Architecture Cost Equations

For each of the candidate LIN architectures, cost and weight equations were developed. The equation for the cost of the architecture is given by:

$$C_{Architecture} = x3.C_{InterWire} + x4.C_{IntraWire} + x5.C_{Node} + x6.C_{Driver} + C_{Fixed} \tag{11}$$

Where

C_{Driver} is the cost of the low-side driver or relay

coefficients *x3* and *x4* are the number of wires

coefficients *x5* and *x6* are the number of nodes and drivers respectively

The equations describing the cost of the LIN candidates and the hardwired original architecture were used to explore the sensitivity of the cost to variations in Non-Material Costs and therefore help ascertain whether the LIN candidate architectures could be of lower cost. Figure 38 shows an example sensitivity analysis comparing the cost of hardwired and the two node LIN architecture candidate. The sensitivity to LIN node cost and *Non-Material Costs* is explored. It shows that the hardwired architecture is likely to be of lower cost that the two node LIN candidate architecture.

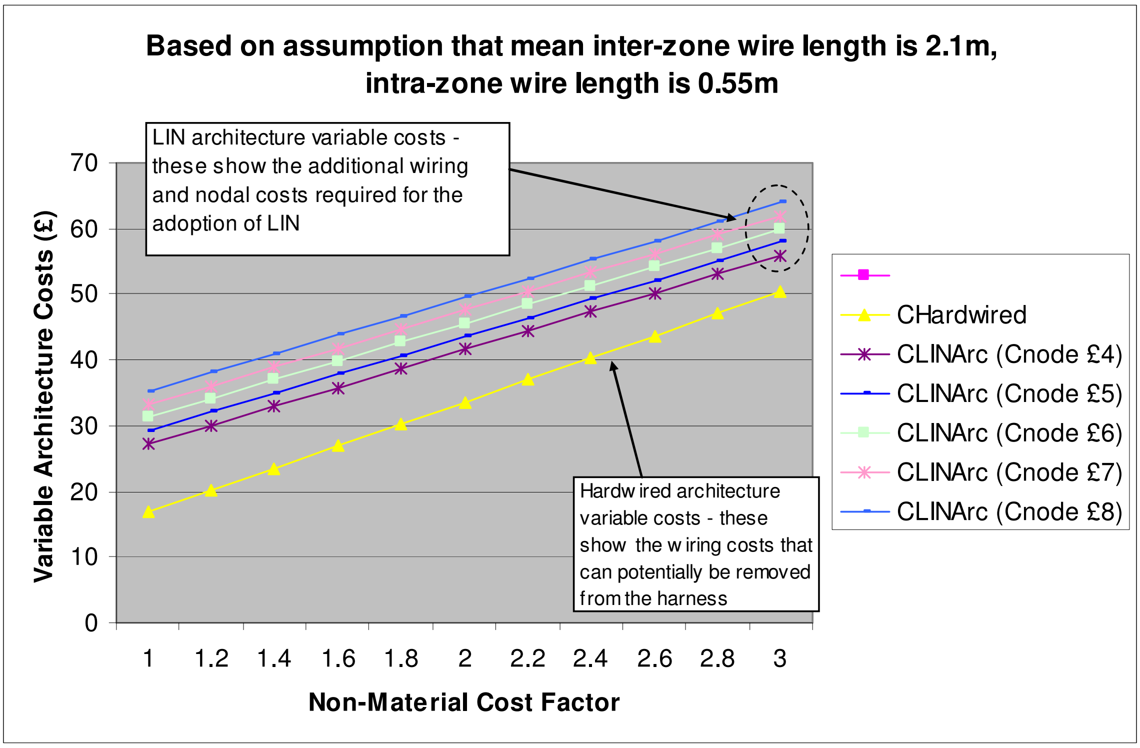


Figure 40: Comparison of Hardwired and Two Node LIN Architecture Cost with Variation of *Non-Material Cost* Factor

The hardwired and LIN architecture equations for each candidate were set equal to each other as shown below so that they could be solved and target nodal cost and weights could be obtained. The purpose of the target nodal cost and weight is that this is the value to be achieved for the LIN candidate to be lower than the hardwired original. Therefore beating these targets will result in the LIN candidate architecture being of lower cost or weight than the hardwired original.

$$x1.\mathrm{C}_{\mathrm{InterWire}} + x2.\mathrm{C}_{\mathrm{IntraWire}} + \mathrm{C}_{\mathrm{Fixed}} = x3.\mathrm{C}_{\mathrm{InterWire}} + x4.\mathrm{C}_{\mathrm{IntraWire}} + x5.\mathrm{C}_{\mathrm{Node}} + x6.\mathrm{C}_{\mathrm{Driver}} + \mathrm{C}_{\mathrm{Fixed}} \tag{12}$$

Table 13 shows the summary of results for cost. All of the LIN candidates were projected to be of higher cost than the hardwired original architecture and therefore there is no cost benefit from the adoption of LIN in this case. It was projected that the two node architecture would result in the lowest add-on nodal cost for the addition of LIN communications.

Architecture	**C_{Node} Target (UK Sterling)**	**Sensitivity Analysis Result**	**Rank**
Two Node	< £0	~£9 add-on cost	1
Three Node	< £0	~£12 add-on cost	2
Five Node	< £0	~£20 add-on cost	3

Table 13: Comparison of LIN candidate architectures on a cost basis

Weight Equations and Sensitivity Analysis

For each of the candidate LIN architectures, the equation for the weight of the architecture is given by:

$$W_{Architecture} = x3.W_{InterWire} + x4.W_{IntraWire} + x5.W_{Node} + x6.W_{Driver} + W_{Fixed} \quad (13)$$

Where

W_{Node} is the cost of the LIN node

W_{Driver} is the cost of the low-side driver or relay

W_{Fixed} is the fixed harness costs

The sensitivity of architecture weight and target W_{Node} to variations in wire weight was explored in the W_{Wire} W_{Node} sensitivity diagram. The wire weight sensitivity is tested by looking at the effect of changes in the *Insulation* factor since this is an unknown and un-quantified factor and can vary, as stated in chapter four. This is an aspect of the wire weight model which has not been fully quantified and is therefore varied across the hardwired and LIN architectures so that they can be compared across all possibilities.

Figure 39 shows the sensitivity of $W_{Architecture}$ and target W_{Node} to wire weight by variation of *Insulation* factor and different nodal weights for the three node low-side driver-based LIN architecture. The plot suggests that the LIN architecture will be of lower weight in all occasions by at least 150 to 350 grammes.

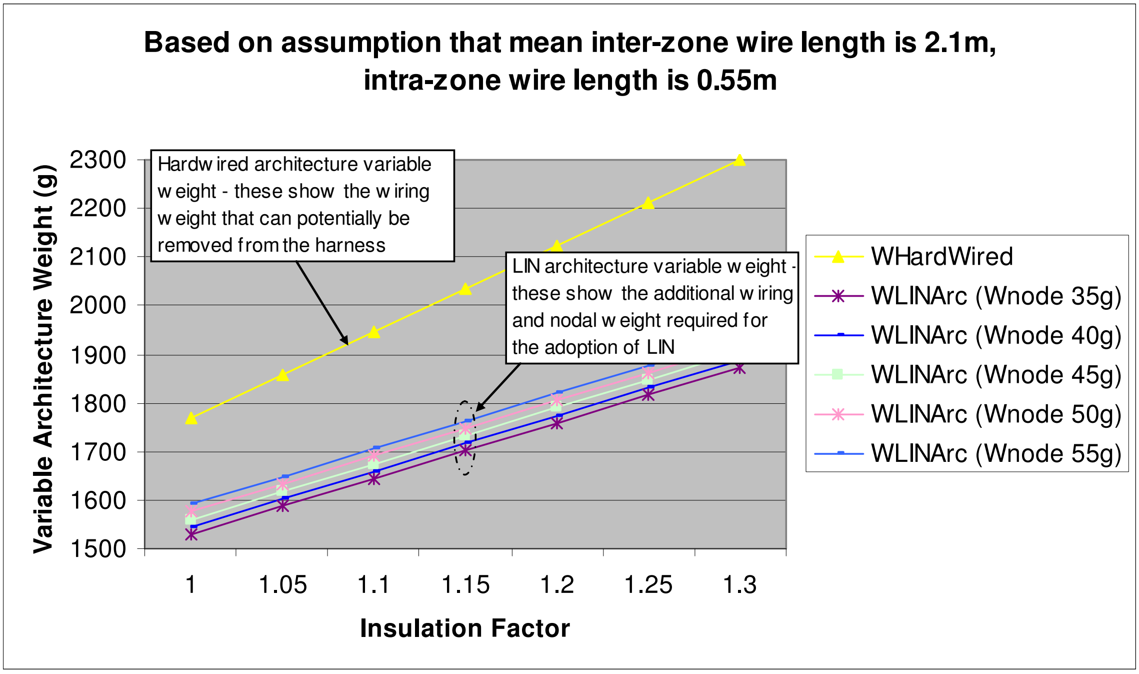


Figure 41: Comparison of Hardwired and Three Node LIN Architecture Weight with Variation of Wire Weight

Table 14 compares the target nodal weight and potential for weight reduction of the three LIN candidates and ranks them. It can be seen that the three node LIN architecture is the candidate that ranks highest as the one that will potentially lead to the greatest weight reduction.

Architecture	**Target W_{Node} (grammes)**	**Potential Weight Reduction @ Max. Nodal Weight (55g) (grammes)**	**Ranking**
Two Node LIN	> 55g	~100g to 250g	2
Three Node LIN	> 55g	~150g to 350g	1
Five Node LIN	< 45g	< 0g to 90g	3

Table 14: Comparison of LIN candidate architectures on a weight basis

Microcontroller Selection for Sportscar LIN Architecture

The number of signals and node information for each of the candidate LIN architectures was used in conjunction with the LIN communications stack ROM/RAM requirement estimation process (as described in chapter five). This was used to estimate the ROM and RAM requirements of the LIN communication stack for each individual node and hence provide enough information for the selection of the most appropriate microcontroller variant. It was found that all nodes could fit into a microcontroller variant with 8Kbytes ROM and 768 such as the PIC 18F6310 and therefore was used in the remainder of the analysis.

Improving Wire Length and Nodal Cost Information Used in Estimations

Next the LIN architecture designs that were proposed were analysed further by adding more detail. This gave a benchmark to compare the results that were obtained from the coarse analysis and therefore validate the processes used.

The assessment was limited in that it did not give the cost and weight values for each proposed architecture as a fully designed harness. This was not feasible in the scope and budget of the study as the harness designs would have taken many man-months to complete, price and weigh up. To keep within the scope, it had been shown that the architectures with inline relays would incur most cost due to the additional inter-zone wires required to go from the relay to the lamp. Therefore only the three architectures based on low-side drivers were analysed further. Improved estimations of wire harness cost and weight were made by first deciding upon the actual location of the LIN nodes within the wiring harness and by then calculating the actual lengths of signal wires to and from the node. Therefore it gave more detail on the differences in wire cost between each of the architectures. Secondly nodal cost and weight were estimated. Cost was estimated by first estimating the LIN communications stack ROM/RAM requirements (as described in chapter five), using this information to select an appropriate microcontroller variant and then pricing up the bill of materials of the proposed node. LIN node weight was obtained by looking at relevant examples that were commercially available at the time. Figure 42 shows the flow chart which was followed to assess the processes as used in submission five.

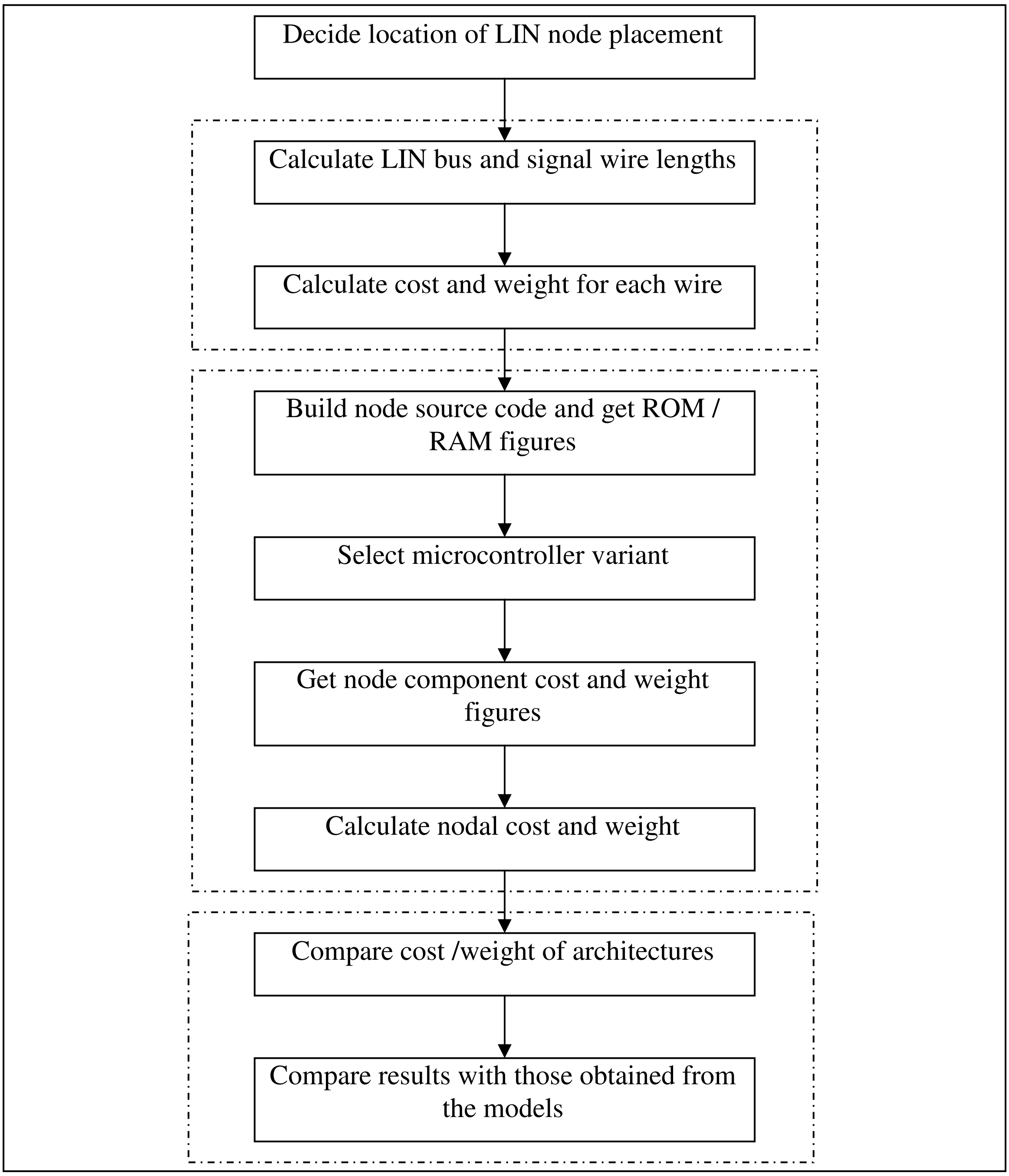

Decide location of LIN node placement
Calculate LIN bus and signal wire lengths
Calculate cost and weight for each wire
Build node source code and get ROM / RAM figures
Select microcontroller variant
Get node component cost and weight figures
Calculate nodal cost and weight
Compare cost /weight of architectures
Compare results with those obtained from the models

Figure 42: Assessment of design processes and models

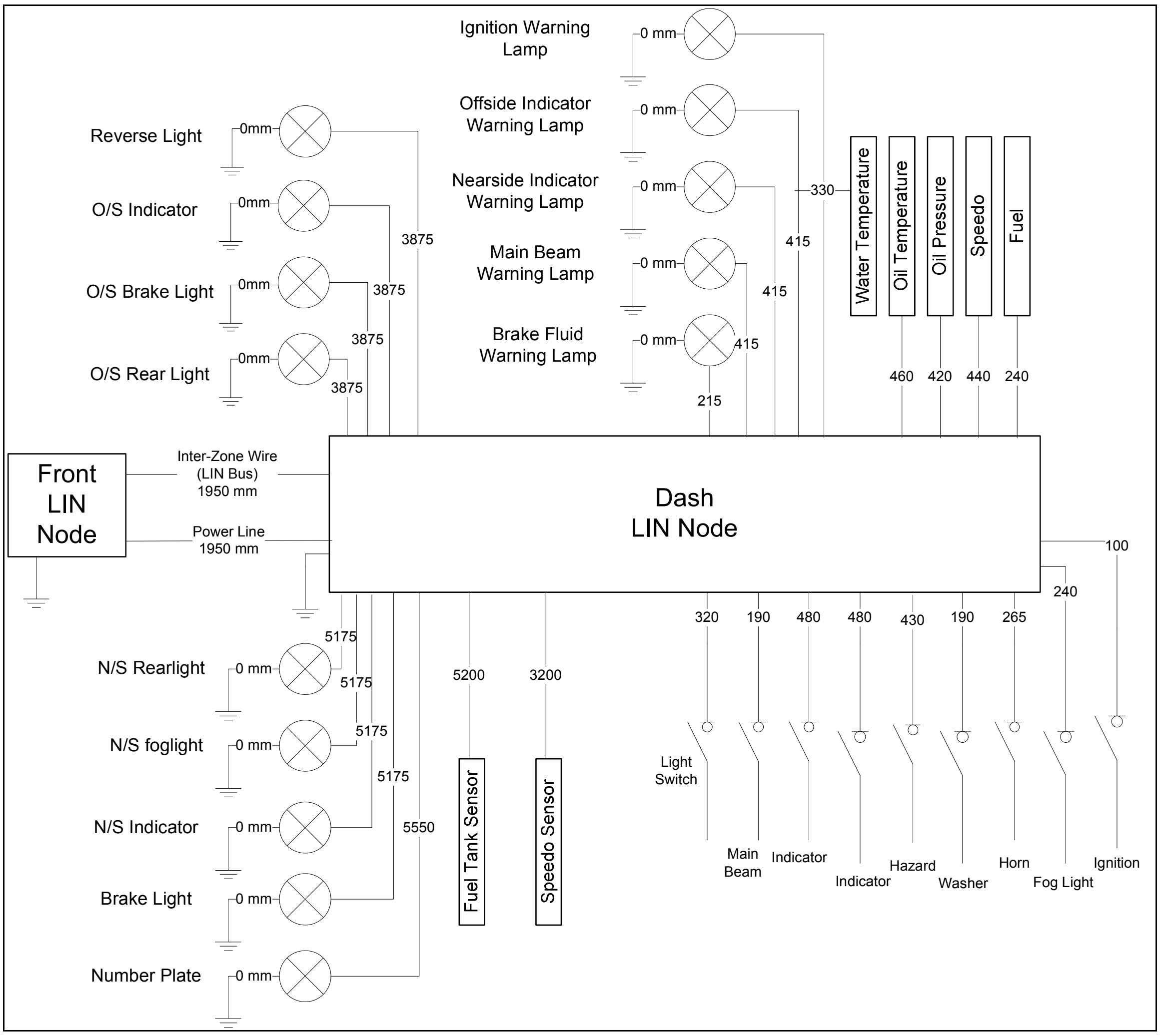


Figure 43: Dash LIN node actual connections two node architecture

Figure 41 shows an example schematic that was developed so that the actual wire costs and weights could be calculated. It can be seen that it contains the actual lengths of the wires in millimetres which were then used in the re-assessment of the cost and weight of the candidate architectures.

The original estimations (that used a mean inter-zone and intra-zone wire length) and the improved estimation (that used actual inter-zone and intra-zone wire length) were compared. The result of this comparison for architecture cost is shown in Table 15. In this table it compares the cost saving per architecture with a variation in Non-Material Cost factor to test the sensitivity. All candidates resulted in an increase in architecture cost. This is the same qualitative projection as was found with the original estimation. The original estimation suggested that the two node candidate resulted in the smallest increase in cost with the three node architecture not far behind. The improved estimation suggested that the three node LIN architecture would be best to select for the least increase in cost. This is because when the *Non-Material Cost* factor was set low and the architecture costs evaluated, there was very little difference in the add-on cost between two or three node architectures (i.e. both are around £25 whilst the five node architecture candidate is around £40). However when the *Non-Material Cost* factor was set high, the three node candidate was projected to only increase cost by around £23 whilst the two node candidate was projected to increase cost by around £37.

	Cost			
	Original Estimation		Improved Estimation	
Nodes	Low *Non-Material Cost* Factor Saving (£)	High *Non-Material Cost* Factor Saving (£)	Low *Non-Material Cost* Factor Saving (£)	High *Non-Material Cost* Factor Saving (£)
2	-£16.38	-£11.46	-£25.25	-£37.58
3	-£23.81	-£18.43	-£25.80	-£23.67
5	-£38.93	-£35.79	-£40.21	-£38.42

Table 15 : Comparison of original and improved cost estimations

Table 16 shows the results of the comparison of the original and improved estimation. It can be seen that the polarity of the estimations agree for the three and five node LIN candidate architectures. The estimations for the two node LIN candidate contradict each other in that the original suggested that a move to a two node LIN architecture would result in a weight saving whereas the improved estimation suggested that there would be an increase in architecture weight. The three node LIN candidate architecture results in the most weight saving.

	Weight			
	Original Estimation		Improved Estimation	
Nodes	**Low Insulation Factor Saving (WNode 35g)**	**High Insulation Factor Saving (WNode 35g)**	**Low Insulation Factor Saving (WNode 35g)**	**High Insulation Factor Saving (WNode 35g)**
2	164	305	-589	-677
3	239	426	206	380
5	36	187	175	364

Table 16 : Comparison of original and improved weight estimations

The results for the two node LIN candidate architecture performed the least well, particularly for weight estimation and therefore it was deemed necessary to find out why there was this contradiction from the estimations. For all of the original estimations, the mean inter-zone wire length used in the model was 2.1 metres and the mean intra-zone wire length was 550mm.

Nodes	**Inter-Zone Wires**		**Intra-Zone Wires**	
	Mean Length (mm)	**% Variation from Original Mean of 2100mm**	**Mean Length (mm)**	**% Variation from Original Mean of 550mm**
2	4173.08	198.7%	656.14	119.3%
3	2925.00	139.3%	713.59	129.7%
5	1768.75	84.2%	600.41	109.2%

Table 17 : Comparison of original model mean and actual mean wire lengths

Table 17 shows the actual mean wire lengths that were used in the improved estimations. It can be seen that for all of them the intra-zone wire lengths are only slightly longer than the 550mm mean length used in the original model. However in the case of inter-zone wires for the two node LIN candidate, the actual mean was around four metres which is around twice the value used. It must be noted that the inter-zone wires is the source of the majority of wire removal in a move from a hardwired architecture to a LIN one. Therefore because the actual length of the inter-zone wires for the two node LIN architecture candidate was nearly twice the value used in the original models, it is not that surprising that the original and improved estimations contradict each other on some occasions. This is a key lesson for the successful use of the processes in future projects.

Overall, it was concluded that the adoption of a LIN-based alternative architecture to that used within the hardwired sportscar body control wiring harness is unlikely to lead to a reduction in the cost of the harness. In fact it is likely to cost at least twelve pounds more per vehicle based on the original estimations and more likely at least twenty five pounds based on the improved estimations. However, it is likely to lead to a reduction in harness weight, e.g. from 3.2 kg reduced by around 0.2 to 0.4 kg. The inclusion of further information in the assessment of the processes showed that the original estimations benefited by using more representative values for the mean inter-zone wire length. It was therefore also concluded that there is likely to be a difference in the mean inter-zone wire length calculated from the bill of materials of the original hardwired architecture and the length of inter-zone wire used in the candidate LIN architectures. Therefore in any future project of this kind it is important to ensure that inter-zone wire length is not blindly calculated from donor harness data and that consideration is given to the realistic lengths of LIN wires that may be deployed in the vehicle.

Although the study indicated that any cost saving from the adoption of LIN was unlikely, it was discussed that there are ways of absorbing the cost in future electrical architecture designs. It was shown that at the target quantities of manufacturing for the sports car manufacturer, each LIN node added around eight pounds to the cost of the architecture. This is a problem because the original chassis harness had no network node and therefore these were completely new items. If in the future other network nodes (maybe based on CAN) were to be added to the vehicle for other applications, there may be the opportunity to add LIN functionality to one or more nodes and absorb some of the add-on cost, thus reducing the cost impact of adding LIN.

6.2. Case Study – Main Outcomes

The study provided a number of outcomes, one of which was that there is now the basis of a preliminary LIN-based sportscar body control electrical architecture design available. This preliminary design has outlined that three nodes are the best for lower weight but for an additional cost. It is possible to make an adaptation of the architecture from this study to offset the additional cost caused by the adoption of LIN. Possible changes include replacing the dash node proposal with an intelligent dash/node/display thus replacing other components such as vehicle and engine speed dials. Therefore this is a case of offsetting the additional cost of one of the nodes by having the LIN functionality in an already existing node or one that brings new functionality.

The architecture design to cost process developed during the work under taken for submission two was applied to the new problem for the niche sports car manufacturer. This provided new challenges such as difference in data on the original harness, assessment of six candidate LIN architectures and an additional design target which was to design for low architecture weight. The design-to-cost process used a factor referred to as *LABOUR* in submission two. This was renamed to help properly describe its purpose to the name of *NON-MATERIAL COSTS*. A design-to-weight process was developed by adapting the design to cost process. The processes were used to ascertain which candidate LIN architecture will provide lowest weight and cost, and the processes were further validated by taking the designs further and using the actual wire lengths that would be used in the LIN architectures to enhance the comparison.

In submission two there were two types of wires in the original architecture that was under analysis: intra-door and inter-door wires. Each of the four doors in the original architecture were effectively a zone containing wires (intra-zone wires) as well as having wires going between each of them (inter-zone wires). In submission five, the concept of zones in the harness or architecture was used again and therefore becoming a main part of the process. In this case the zones were *Front*, *Dash* and *Rear*.

Finally, within the work of submission five that has been described in this chapter has developed the new concept of an intelligent wiring system for niche vehicle manufacturers based on LIN technology.

7. CONCLUSION AND RECOMMENDATIONS FOR FURTHER WORK

7.1. Conclusion

This Innovation Report has provided an overview and analysis of the research carried out during the Engineering Doctorate programme. The aim of the research was to provide an improvement to the design-to-cost processes used in automotive electrical architecture design and selection.

Chapter one described the commercial aims of the sponsoring organisation Rapicore and its key product NetGen and described why there was a commercial and wider industry need for research into the design-to-cost and partitioning of automotive electrical architectures based on in-vehicle networks. In chapter two a review of the relevant literature was carried out. Automotive electronic applications, in-vehicle networking technology, automotive electronic architecture design processes and issues related to in-vehicle networking costing were reviewed.

In chapter three the first of two case studies was described. The aim of this case study was to ascertain whether a LIN alternative to an automotive door system electrical architecture (originally hardwired for electronic integration) could result in a reduction in cost. The key outcome from this case study was a design-to-cost process that can help an automotive electrical architecture designer ascertain whether a particular candidate architecture can reduce cost and also estimate the cost target for the addition of LIN communication components to electrical architecture nodes. Another outcome from the work was that the target nodal cost for the addition of LIN technology to the architecture nodes required an apparently challenging cost target to be met.

In chapter four the design-to-cost process was described. Although research in design to cost was the main focus, the requirements of the second case study meant that an architecture design-to-weight process was also required. An adaption of the design-to-cost process was developed to form a design to weight process. This new process can be used for the design of low weight automotive electrical architectures and was used in the second case study.

In chapter five the work from submissions three and four were described which resulted in the development of two new processes. The relationship between microcontroller ROM/RAM capability and unit cost was explored with the aim to help understand if and how a challenging nodal cost could be met. One key outcome from this investigation was that it was clear that ROM and RAM capability did have a direct impact on the unit cost of a microcontroller but the relationship was not clear enough to be able to produce a model. Another key outcome was that there was a linear relationship between the ROM and RAM capability of the three microcontroller families studied. The relationship between the LIN communications stack features and cost was explored, motivated by a need to understand more about how the design of the LIN communications stack itself can help challenging LIN nodal cost targets be met. The relationship between LIN communication stack ROM/RAM requirements and LIN network features, such as, the number of nodes, schedules, messages and signals was explored. It was found that it was possible to produce linear regression models for the estimation of the LIN communications stack ROM/RAM requirements as a function of the number of nodes, schedules, messages and signals. It was also shown how the capability of a microcontroller could be compared with the ROM/RAM requirements of a LIN communications stack to ascertain their suitability to work together and therefore to also understand how both microcontroller and stack designs could be altered to enjoy a cost reduction. Therefore the two processes that came out of this work were firstly for the assessment of the ROM/RAM capability of a family of microcontrollers and secondly for the selection of a microcontroller variant for LIN applications by estimation of the ROM and RAM requirements of the LIN embedded software.

In chapter six, the opportunity arose to apply the processes to a new case study and to also carry out some validation of the processes. The aim was to ascertain whether there could be a reduction in the cost and weight of the body control harness by the adoption of a LIN-based architecture instead of using the original hardwired integration. It was found that a cost reduction was unlikely but the adoption of a three node LIN architecture had the potential to reduce weight. The processes were further validated by pricing up and weighing up the nodes and determining the actual wire lengths that would be used. The lesson learned was that it is important to use realistic inter-zone wire lengths to ensure that the results obtained are valid.

It is concluded that the research has developed new and innovative processes for automotive electrical architecture design based on in-vehicle networking technology. The key innovation of these processes is that they provide a quantitative methodology requiring coarse information only, the kind which would easily be available at the beginning of a design project. Therefore this means that numerous candidate architectures can be quickly assessed without the need for completing a full architecture and harness design.

The original focus was limited to the design-to-cost process for in-vehicle networked architectures. However this was extended by the requirements of the second case study to include architecture weight as an additional design target. The processes have been demonstrated for LIN bus applications and a key feature has been that these processes use minimal information such as that which would only be available at the beginning of a vehicle programme. However the processes could potentially be applied to any in-vehicle networking technology. The application of the processes and the benefits of their use can be summarised as:

- A design-to-cost process for accessing whether a new in-vehicle networked architecture has the potential of being lower cost than the original that is to be replaced. Another output from this is a cost target of adding the network communications capability to each of the architecture nodes. A key advantage of this process is that it does not require a full architecture design to be priced up. Instead, it uses coarse information of a preliminary design meaning that it is much easier to compare a number of candidate architectures quickly. This was demonstrated in two case studies.
- A design to a target weight process for assessing whether a new in-vehicle networked architecture has potential of being lower weight than the original that is to be replaced. One other output from this is a weight target of adding the network communications capability to each of the architecture nodes. Again, as with the design-to-cost process the benefit is that only coarse information is used making it much easier to compare a number of candidate architectures quickly. This was demonstrated in one case study.
- A process for assessing the memory capabilities of a family of microcontrollers for the family's ability to accommodate an embedded software component such as network communications software. This was demonstrated on a commercially available LIN communications stack design and the outcomes have influenced its design for future commercial applications.

- A process for estimating the memory requirements of LIN embedded software for each node in a proposed LIN system and using this estimation for microcontroller selection and therefore assisting in more accurate pricing projection. It can also be used with the previous process for assessing the memory capability of a family of microcontrollers by comparing with the LIN communications stack requirements. Decisions can then be made on where further design effort is needed for the software or the microcontroller. This was demonstrated on the commercially available LIN communications stack design (as generated by the NetGen tool) and also demonstrated on the sportscar architecture design case study.

7.2. Recommendations for Further Work

7.2.1. Potential Future Exploitation of Research Findings

The project has allowed the development of new processes during commercial research work. These were primarily focused on two main projects whose aims were to explore whether the adoption of LIN would provide a cost benefit over their existing hardwired integration. The second of these projects was for a niche sportscar manufacturer who had a secondary target of reducing the weight of the wiring harness and therefore had an interest in seeing if the adoption of LIN could reduce the weight of the electrical architecture.

The design of the LIN communications stack that is generated from the NetGen tool benefited from the research carried out during the Engineering Doctorate as it was clearly highlighted that its RAM requirements were too high for 8-bit microcontroller applications and has therefore been improved to address this.

However, there are three main areas which will commercially benefit from the research in the future:

1. The NetGen design tool can benefit from small improvements, e.g. ROM/RAM estimation feature added to the current generation of the product. The normal process for ascertaining the memory requirements of a LIN communications stack is to generate the stack, ensure that it compiles without errors and then read the memory map file to get the ROM and RAM amounts used. The problem with this is that firstly it requires the code to compile without errors and secondly it requires a compiler which can cost around two thousand pounds. This is a large project expense just to estimate whether a microcontroller variant is capable in terms of its memory. A new unique selling point of the NetGen tool could be to give an estimation of a stack's memory requirements with details in a LIN Description File (LDF) alone, negating the requirement for a compiler for the customer until they wish to start the project. This means that a customer will be able to evaluate NetGen, the LIN stack and microcontroller with very little upfront investment.
2. A distributed architecture design service could be provided. This service would use the design-to-cost and weight processes developed and could act as a way of gathering further requirements for a future architecture design product, thereby increasing the information available in addition to the two case studies carried out so far.
3. The development of the next generation architecture design tool to replace the currently available NetGen. This could be a design to cost/design to weight tool for comparing and selecting the most appropriate architecture for the design targets. Other design targets could be used in the future such as EMC, assembly,

manufacture etc. Such a tool would also be required to deal with a multi-network vehicle architecture perhaps using different technologies such as CAN, LIN and FlexRay. At the time of writing the securing of funding for this project is being considered.

7.2.2. Embedded Software Memory Requirements Modelling

It was shown that it is possible to model ROM and RAM requirements of a LIN communications stack using linear regression modelling methods. This was possible since the relationships between network characteristics such as nodes, schedules, messages, signals etc. had a linear relationship with ROM and RAM.

Many embedded systems components used in automotive ECUs will have ROM/RAM requirements which are a function of their characteristics. Therefore there is the potential that other such components could be modelled and their ROM/RAM requirements estimated. CAN and FlexRay communications stacks will have ROM/RAM requirements which are a function of their network characteristics. An ECU fault manager embedded software component's memory requirements will be a function of the number of faults. A diagnostic kernel's memory requirements will be a function of the number of diagnostic services. An operating system kernel's memory requirements will be a function of the number of tasks. Research is therefore required to ascertain whether these relationships can be modelled. If this is the case, there will then be the increased possibility of estimating the ROM/RAM requirements of an entire ECU's embedded software with better accuracy.

7.2.3. Adaptation of Processes to Other Network Technologies and Applications

The processes developed during the research in this report have been successfully demonstrated using commercial LIN applications for automotive body control. The LIN applications have been single network architectures and also the typical size of the embedded software is relatively small. It would be interesting to ascertain whether the processes could be adapted for other automotive networked systems such as those based on CAN, FlexRay or any other network technologies that may come along in the future. These differ from the LIN applications that have been covered in this report in that the size of the embedded software is larger, the speed of the networks are faster and they are often exploited in multiple network electrical architectures. Non-automotive applications such as industrial controls may also have design targets such as cost, weight or embedded system memory requirements and therefore it would be interesting to see if the processes could be adapted to this sector too.

7.2.4. Automatic Rule-Based Optimisation of Architecture Design

Processes have been described for the design to cost and weight of automotive electrical architectures. They allow a number of alternative architecture candidates to be assessed and the lowest weight and cost one selected. Currently the process of proposing, evaluating, comparing and selecting a network architecture is manually driven. However it would be desirable if this was semi or fully-automated so that a full optimisation of the architecture can take place.

8. REFERENCES

Albert A and Gerth W (2003); "Evaluation and comparison of the real time performance of CAN and TTCAN", Proceedings of the 9th International CAN Conference, October 14 - 16, 2003, Munich (Germany).

Axelsson J (1999); "Holistic Object-Oriented Modelling of Distributed Automotive Real-Time Control Applications", Second IEEE International Symposium on Object Oriented Real Time Distributed Computing, May 2 - 5, Saint-Malo, France.

Axelsson J, Fröberg J, Hansson H, Norström C, Sandström K, Villing B (2003); "Correlating business needs and network architectures in automotive applications – A comparative case study", 5th IFAC International Conference on Fieldbus Systems and their Applications, 2003, Portugal, July 7th – 8th 2003.

Audino D, Baronti F, Lenzi F, Roncella R, Saletti R, Di Tanna O (2007); "A perspective on in-motorcycle electronic systems", Proceedings of the IET Automotive Electronics Conference, University of Warwick, Coventry, June 2007.

Beaurenaut L, Eggiman C, Gastinger F, Platzdasch H, Rasbornig F, Strasser M (2009); "Single-Edge Nibble Transmission: Challenges and Evolutions", SAE World Congress, April 2009, Detroit, MI, USA. Paper 2009-01-0125.

Berwanger, **J. and C. Ebner, et al. (2001)**; "FlexRay--The Communication System for Advanced Automotive Control Systems", SAE World Congress, Detroit, SAE Press paper 2001001-0676.

Berwanger, **J., Peller M and Griessbach R (2000)**; "A new high performance data bus system for safety related systems", BMW AG, EE-211 Development Safety Systems Electronics, Knorrstrasse 147, 80788 Munich, Germany.

Berwanger J, Schedl A and Peller M (2005); “BMW – First Series Cars with FlexRay in 2006”; Automotive Electronics Systems, Special Edition FlexRay, Page 6 to 8, Hanser Publishing.

Berwanger J, Schedl A, Peller M and Peteratzinger M (2005); “BMW – First Series Cars with FlexRay in 2006”, Proceedings of the FlexRay Product Day, 1st December 2005, Boeblingen, Germany, Hanser Publishing.

Boys, **R (2004)**; “Safe-by-Wire: The Leading Edge in Vehicle Airbag Control”, SAE 2004 World Congress Paper No. 2004-01-0205.

Brandao F and Howell M (2003); “Genetic Learning Automata and Fuzzy Controller Applied to Adaptive Suspension”, SAE 2003 World Congress, Detroit, 2003.

Burke A.F. (1992); “Hybrid/Electric Vehicle Design Options and Evaluations”, Society of Automotive Engineers Paper 920447.

Byteflight Specification 1999, BMW AG (see www.byteflight.com).

CAN Specification, Version 2.0, (1991); Robert Bosch GmbH, Germany.

Electronic Architecture and System Engineering for Integrated Safety Systems, Deliverable D2.1, Requirements Specification for WP2 Hardware Architecture, Version 1.0, EASIS Project Papers (www.easis.org), 2005.

CANaerospace Interface specification for airborne CAN applications V 1.7 (2006), Stock Flight Systems, Schützenweg 8a, 82335 Berg/Farchach, Germany http://www.canaerospace.com/html/projects.html. Accessed on 22/02/2006

CANopen Specification, CAN in Automation, (www.can-cia.org)

CAN Newsletter 3/2006_1, “CANopen Lift in the Airbus A380”, CAN Newsletter 3/2006 page 56, Published by CAN in Automation, Am Weichselgarten 26, DE-91058 Erlangen, Germany (www.can-cia.org).

CAN Newsletter 3/2006_2, “CANopen Lift in the Airbus A380”, CAN Newsletter 3/2006 page 56, Published by CAN in Automation, Am Weichselgarten 26, DE-91058 Erlangen, Germany (www.can-cia.org).

Cox M, Bomya P and Klang J (2003); “Aspects of a Robust In-Vehicle Lead-Acid Battery Monitoring System (inGEN); SAE 2003 World Congress, Detroit, 2003.

DePoyster M, Hoying F and Majeed K (1996); “Rapid Prototyping of Chassis Control Systems”, Proceedings of the 1996 IEEE International Symposium on Computer-Aided Control System Design, Dearborn, MI, September 15-18, 1996.

DeviceNet Specification, Open DeviceNet Vendors Association, (www.odva.org).

Dijkstra, E.W. (1965); "Programming Considered as a Human Activity," Proceedings of the 1965 IFIP Congress (Amsterdam, The Netherlands), North HollandPublishing Company, 1965, pp. 213 – 217.

Dijkstra, E.W. (1968); "Go To Statement Considered Harmful," Communications of the ACM, Vol. 11, No. 3, March 1968, pp. 147 - 148.

Dorey R and Maclay D (1996); “Rapid Prototyping for the Development of Powertrain Control Systems”, Proceedings of the 1996 IEEE International Symposium on Computer-Aided Control System Design, Dearborn, MI, September 15-18, 1996.

ElBatt T, Saraydar C, Ames M, Talty T, (2006); "Potential for intra-vehicle wireless automotive sensor networks", 2006 IEEE Sarnoff Symposium, 27-28 March 2006, Princeton, NJ.

FIBEX - Field Bus Exchange Format - MCD-2 Version 2.0 - Association for Standardisation of Automation and Measuring Systems Dated: 25.01.2005

FlexRay Specification Package v2.1.

Fischer M (2002); "TIRE GUARD – Tyre Pressure Monitoring System", Proceedings of the ERA Conference on Vehicle Electronic Systems, 25-26 June 2002, Coventry, UK.

Graf H.M. (2002); "The Integrated Starter-Generator on 42volt as a Basis for Further Increasing Comfort", Proceedings of the ERA Conference on Vehicle Electronic Systems, 25-26 June 2002, Coventry, UK.

Hamm M. (2002); "Active Safety Performance Generated with Adaptive Lighting Systems"; SAE 2002 World Congress, Detroit 2002.

Hanselmann H (1996); "Hardware-in-the-Loop Simulation Testing and its Integration into a CACSD Toolset", Proceedings of the 1996 IEEE International Symposium on Computer-Aided Control System Design, Dearborn, MI, September 15-18, 1996.

Hayton K. (2002); "Telematics and Electronic Road Tolling", Proceedings of the ERA Conference on Vehicle Electronic Systems, 25-26 June 2002, Coventry, UK.

Haslam J (2006); "Automotive Systems Timeline"; Innovative Manufacturing Research Centre, Internal Report, University of Warwick, United Kingdom.

Hills C (2006); "A Standard Individual: Ingredients and Chips", Embedded Systems Engineering Magazine, October 2006, Page 54.

IEC61508 "Functional safety of electrical/electronic/programmable electronic safety-related systems"

ISO7498-1: 1994 - Information technology -- Open Systems Interconnection

ISO9141 - Road vehicles -- Diagnostic systems -- Requirements for interchange of digital information

ISO11519-1:1994 Road Vehicles – Low Speed Serial Data Communication Part 1: General and Definitions

ISO11519-2:1994 Road Vehicles – Low Speed Serial Data Communication Part 2.

ISO11519-3:1994 Road Vehicles – Low Speed Serial Data Communication Part 3: Vehicle Area Network (VAN).

ISO11898-1:2003 Road Vehicles – Controller Area Network (CAN) Part 1: Data Link Layer and Physical Signalling.

ISO11898-2:2003 Road Vehicles – Controller Area Network (CAN) Part 2: High Speed Medium Access Unit.

ISO11898-3:2003 Road Vehicles – Controller Area Network (CAN) Part 3: Low Speed, Fault Tolerant, Medium Dependent Interface.

ISO11898-4:2004 Road Vehicles – Controller Area Network (CAN) Part 4: Time Triggered Communication.

ISO11992-1:2003 Road Vehicles – Interchange of digital information on electrical connections between towing and towed vehicles – Part 1: Physical and data-link layers.

ISO14230 - Road vehicles -- Diagnostic systems -- Keyword Protocol 2000.

ISO15031: Road vehicles -- Communication between vehicle and external equipment for emissions-related diagnostics.

ISO15504 - Information technology - Process assessment.

ISO15765 - Road vehicles -- Diagnostics on Controller Area Networks (CAN).

Johannessen P and Torin J; (2003); "System Safety Design of the SIRIUS 2001 Drive-by-Wire Car – In Retrospect", PROCEEDINGS of the 21st INTERNATIONAL SYSTEM SAFETY CONFERENCE – 2003

Jones M (2009); "EMI Challenge to Ethernet in the Car", Micrel Inc.

Jurgen R.K.(1999) "Automotive Electronics Handbook", Second Edition, McGraw Hill.

Kim ES (1996); “Nonlinear Indirect Adaptive Control of a Quarter Car Active Suspension”, Proceedings of the 1996 International Conference on Control Applications, Dearborn, MI, September 15-18, 1996.

Khoh S B (1993); “The Design and Build Up of CAN (Controller Area Network)”, MSc Thesis, University of Warwick, UK.

Kopetz H (1995); “TTP/A - A Time-Triggered Protocol of Body Electronics Using Standard UARTs”, Proc. SAE World Congress, Society of Automotive Engineers, SAE Technical Paper 950039. pp.1-9.

Kopetz H, Adernaj A, Gorillinger P, Steinhammer K (2005); “The Time Triggered Ethernet (TTE) Design”, Eighth IEEE International Symposium on Object Oriented Real Time Distributed Computing.

Kopetz H and Thurner T (1998); “TTP – A New Approach to Solving the Interoperability and Problem of Independently Developed ECUs”, In SAE International Congress and Exposition, Detroit, MI, USA.. SAE Technical Paper 981107, Feb. 1998.

Knuth D E (1974), "Structured Programming With GOTO Statements"

Laes E, Corbiere T, Cappiello C, Joubert F, Finney J, Gonnet P and Quigley C (2009); “Next Generation of Fail-Safe Architectures and Components” SAE World Congress, 2009, Detroit, Paper No. 2009-AE-0011.

Leen, G., Heffernan, D., and Dunne, A. (1999) Digital Networks in the Automotive Vehicle. IEE Computing & Control Engineering Journal. Vol. 10, No.6. Dec. pp. 257-266. **LIN Specification Package, Revision 1.2**, 17th November 2000, The LIN Consortium **(www.lin-subbus.org).**

LIN Specification Package, Revision 1.3, 12th December 2002, The LIN Consortium (www.lin-subbus.org).

LIN Specification Package, Revision 2.0, 23rd September 2003, The LIN Consortium (www.lin-subbus.org).

LIN Specification Package, Revision 2.1, The LIN Consortium (www.lin-subbus.org).

Locher J, Bierleutgeb G, Kleinkes M. and Voelker S (2002); "Night Vision – A System Approach to Human Machine Interfaces Improving Perception and Traffic Safety", Proceedings of the ERA Conference on Vehicle Electronic Systems, 25-26 June 2002, Coventry, UK.

Lowenau J and Strobl M (2002); "Advanced Lighting Simulation for the Evaluation of the BMW System Adaptive Light Control (ALC)"; SAE 2002 World Congress, Detroit, 2002.

Lupini, C (2003); "Multiplex Bus Progression 2003"; SAE 2003 World Congress and Exposition, Paper No. 2003-01-0111.

Mauchauer R (2006); "CAN Configuration within AUTOSAR", Proceedings of the 11th International CAN Conference, Stockholm, Sweden, September 26-28, 2006.

McLaughlin R.T. (1993); "In-vehicle Communication Networks", MSc Thesis, University of Warwick, UK.

McLaughlin R, Williams A and Quigley C (2010); "Essential Tools for the CAN/LIN ECU Development Process", Automotive Electronics Show, 17th to 18th June 2010 Shanghai.

McMurran R, McKinney F, Tudor N and Milam W (2006); "Dependable Systems of Systems", SAE 2006 World Congress & Exhibition, April 2006, Detroit, MI, USA,

MISRA-C 2004; “Guidelines for the use of the C language in vehicle based software”, The Motor Industry Research Association, Watling Street, Nuneaton, Warwickshire, United Kingdom.

NAVET N, SONG Y, SIMONOT-LION F, AND WILWERT C (2005); “Trends in Automotive Communication Systems”, PROCEEDINGS OF THE IEEE, VOL. 93, NO. 6, JUNE 2005.

NMEA2000 Specification Package, The National Marine Electronics Association (www.nmea.org).

OSEK/VDX - Operating System Specification 2.2.22, July 5th, 2004. The official home of OSEK/VDX is http://www.osek-vdx.org

PAN-NGUM S (2002); “Alternative Vehicle Electronic Architecture for Individual Wheel Control”, PhD Thesis University of Warwick, United Kingdom.

Plankensteiner, M (2011); “Time-Triggered Ethernet Using the protocol in vehicle networks”, Industry Bus Systems, ATZ Elektronik, 01/2011 Vol. 6.

Poledna S, Gluck M, Tanzer C, Boutin S, Dilger E, Fuehrer T, Ebner C, Fuchs E, Belschner R, Hedenetz B, Holzmann B, Schedl A, Nossal, Pfaffeneder B, Ringler T, Domaratsky Y, Krueger A, Zahir A (2000); „OSEKtime: A Dependable Real-Time Fault Tolerant Operating System and Communication Layer as an Enabling Technology for By-Wire Applications”, SAE 2000 World Congress, Detroit, Michigan, March 6 – 9, 2000.

Pope B, **Quigley** C, **Finney** J and **McLaughlin** R, **(2005)**; “An Automotive Specification of a Time Triggered CAN Implementation: Doubling CAN's Usable Data Throughput”, SAE 2005 Transactions Journal of Passenger Cars – Electronic and Electrical Systems, Society of Automotive Engineers.

Pfeiffer O, Ayre A and Keydel C (2003); "Embedded Networking with CAN and CANopen", Published by RTC Books, 927 Calle Negocio Suite G, San Clemente, CA 92673.

Quigley C and McLaughlin R (2002); "The future of in-vehicle control networking", Proceedings of the ERA Vehicle Electronic Systems Conference, 25-26 June 2002, Coventry, UK

Quigley C and F.H.P. Tan, K.H. Tang, R.T. McLaughlin (2001); "An Investigation into the Future of Automotive In-Vehicle Control Networking Technology", SAE 2001 International Congress and Exposition, Detroit, USA, March 2001.

Quigley C, Jones R P, McMurran R and Faithfull P (2006); "Design Issues of Integrating CAN with Emerging Time Triggered Protocols", Proceedings of the 11th International CAN Conference, 27th to 28th September 2006, Stockholm, Sweden.

Quigley C, Jones R P, McMurran R and Faithfull P (2007); "An Investigation into Cost Modelling for Design of Distributed Automotive Electrical Architectures", IET Automotive Electronics Conference 2007.

Quigley C, Jones R P, McMurran R and Faithfull P (2009); "Comparing the Harness Cost of Hardwired and Networked Integrated Door Systems" SAE World Congress, 2009, Detroit, Paper No. 2009-AE-0010

Quigley C, Jones R P, McMurran R and Faithfull P (2009); "Modelling of Network Communications Stack Software ROM and RAM Requirements" SAE World Congress, 2009, Detroit, Paper No. 2009-AE-0012

Rajnak A and Ramnefors M (2002); "The Volcano Communication Concept", SAE paper no. 2002-21-0056

Robert Bosch (2004); "Automotive Electrics, Automotive Electronics: Systems and Components", 4th Edition.

Robert Bosch GmbH (2011), "CAN with Flexible Data-Rate", White Paper, Version 1.1, Robert Bosch GmbH, Robert Bosch Platz 1, 70839 Gerlingen, Germany, August 2011.

Royce W.W. (1970); "Managing the Development of large Software Systems: Concepts and Techniques, *Proceeding of the IEEE WESTCON*, Los Angeles, California (1970).

Shiao Y C and Moska J (1996); "Model-Based Cylinder-By-Cylinder Air-Fuel Ratio Control For SI Engines using Sliding Observers", Proceedings of the 1996 IEEE International Conference on Control Applications, Dearborn, MI, September 15-18, 1996.

Sobottka C and Singh T (1996); "Optimal Fuzzy Logic Control for an Anti-Lock Braking System", Proceedings of the 1996 IEEE International Conference on Control Applications, Dearborn, MI, September 15-18, 1996.

SAE J1850 Specification Package. The Society of Automotive Engineers, (www.sae.org).

SAE J1939 Specification Package Rev. Oct. 1998, The Society of Automotive Engineers, (www.sae.org).

SAE J1567 Specification Package, The Society of Automotive Engineers, (www.sae.org).

SAE J2108 Specification Package, The Society of Automotive Engineers, (www.sae.org).

SAE J2602 Specification Package, The Society of Automotive Engineers, (www.sae.org).

Szydlowski C (1994); "Tradeoffs Between Stand-Alone and Integrated CAN Peripherals", 1st International CAN Conference, pp3-16 to 3-22.

Tindell K and Burns A (1994); "Guaranteeing message latencies on CAN", Proceedings of the 1st International CAN Conference.

Tshibangu K P (2002); "Vehicle Collision Warning Systems: Parking Aids – Technologies and Limitations", Proceedings of the ERA Conference on Vehicle Electronic Systems, 25-26 June 2002, Coventry, UK.

Ulke W., Schulter W and Driendl D (1999); "Anti-trap Function for Electrical Power Windows", SAE 1999 International Congress and Exposition, Detroit, Michigan, 1st to 4th of March, 1999.

Ward D (2006); "MISRA Standards for Automotive Software", presentation included in the proceedings of the 2nd International Conference on Automotive Electronics, Savoy Place, London, 20-21st March 2006.

Ward P. T. and Mellor S. J.(1985), "Structured Development for Real-Time Systems", Volumes 1, 2 and 3, Yourdon Press, New York, New York, 1985.

Wense, H. **(2000)**; "Introduction to Local Interconnect Network", SAE World Congress, Detroit, March 2000, Document No. 2000-01-0145, SAE Press.

Westfield Sportscars Limited (2009); "Chassis Wiring Harness Schematic", Westfield Sportscars Limited, Unit 1, Gibbons Industrial Park, Dudley Road, Kingswinford, DY6 8XF, United Kingdom.

Yourdon E (1975), Techniques of Program Structure and Design, Prentice-Hall, Englewood Cliffs, New Jersey, 1975.