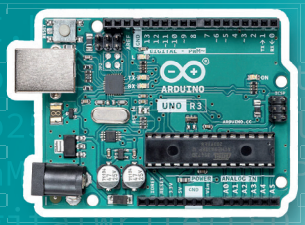
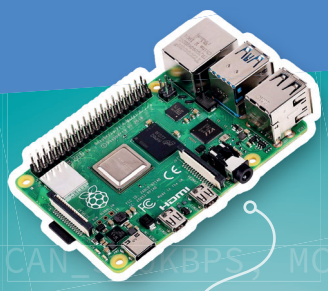
elektor books

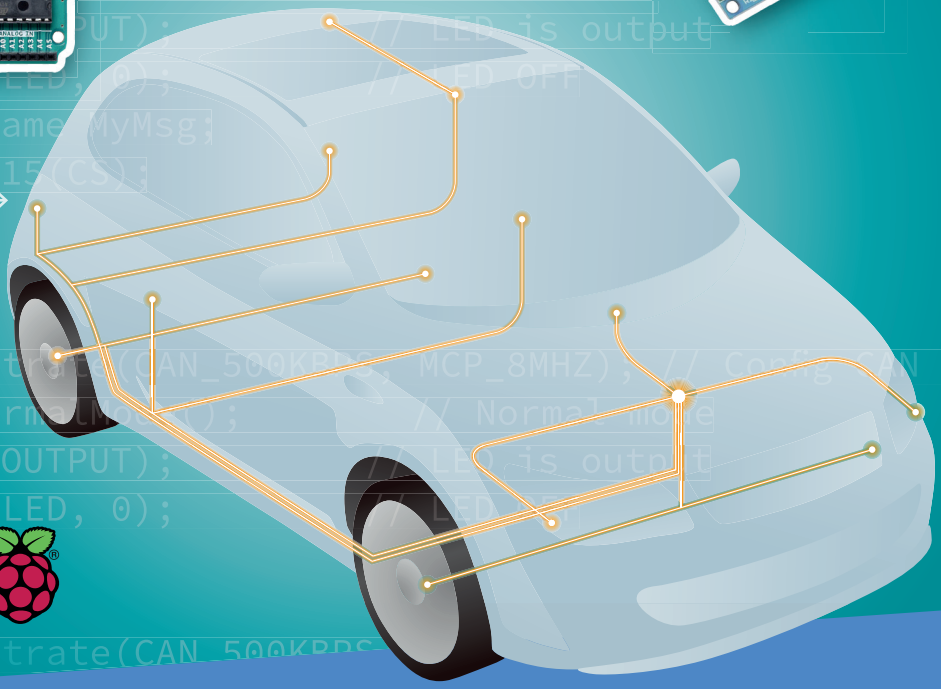
The CAN Bus Companion

Projects with Arduino Uno & Raspberry Pi

With Examples for the **MCP2515 CAN Bus Interface Module**







Dr Dogan Ibrahim
Ahmet Ibrahim BSc., MSc.

elektor
design > share > earn

The CAN Bus Companion

Projects with Arduino Uno & Raspberry Pi

Dr Dogan Ibrahim
&
Ahmet Ibrahim, BSc., MSc.

elektor
design > share > earn

This is an Elektor Publication. Elektor is the media brand of
Elektor International Media B.V.
PO Box 11, NL-6114-ZG Susteren, The Netherlands
Phone: +31 46 4389444

Declaration

The authors and publisher have used their best efforts in ensuring the correctness of the information contained in this book. They do not assume, or hereby disclaim, any liability to any party for any loss or damage caused by errors or omissions in this book, whether such errors or omissions result from negligence, accident or any other cause.
All the programs given in the book are Copyright of the Author and Elektor International Media. These programs may only be used for educational purposes. Written permission from the Author or Elektor must be obtained before any of these programs can be used for commercial purposes.

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN 9783895765414 Print
ISBN 9783895765421 eBook

Editor: Jan Buiting, MA
Prepress Production: D-Vision, Julian van den Berg

Elektor is part of EIM, the world's leading source of essential technical information and electronics products for pro engineers, electronics designers, and the companies seeking to engage them. Each day, our international team develops and delivers high-quality content - via a variety of media channels (including magazines, video, digital media, and social media) in several languages - relating to electronics design and DIY electronics. **www.elektormagazine.com**

Contents

Preface

The Controller Area Network (acronym: CAN) structure was originally developed for use in passenger cars. Today, sophisticated CAN controller chips are available from over 20 manufacturers, and CAN is finding applications in other fields such as medical, aerospace, process control, automation, and so on. With the establishment of CAN within the Automation (CiA) association back in 1992, manufacturers and users rallied to exchange ideas and develop the CAN standards and specifications.

This book is about the use of the Arduino Uno development board and Raspberry Pi 4 with CAN bus interface modules for the design of practical CAN bus based projects. Examples of popular hardware and software development kits are described concisely. Using these kits simplifies the embedded design cycle considerably and considerably eases developing, debugging, and testing a project based on, or relying on, the CAN bus. Projects are given using both the 2- and 3-node CAN bus variants.

This book is written for students, practicing engineers, enthusiasts, and for everyone else who may need to learn more about the CAN bus and its applications. The book assumes that the reader has some knowledge of basic electronics. Knowledge of the C programming language and Python and familiarity with the Arduino Uno and Raspberry Pi will be an advantage, especially if the reader intends to develop microcontroller-based projects using the CAN bus.

The book should be a useful source and reference for anyone interested in finding an answer to one or more of the following questions:

- What bus systems are available for the automotive industry?
- What are the principles of the CAN bus?
- What types of frames (or data packets) are available in a CAN bus system?
- How can errors be detected in a CAN bus system, and how reliable is a CAN bus system?
- What types of CAN bus controllers exist?
- What are the operational principles of the MCP2515 CAN bus controller?
- How do I create a CAN bus project using Arduino Uno?
- How do I create 2-node and 3-node Arduino CAN bus projects?
- How do I analyze the data on the CAN bus?
- How do I create a CAN bus project using Raspberry Pi?
- How do I create 2-node and 3-node Raspberry Pi CAN bus projects?

We hope that you will find the book helpful and enjoyable and will be able to create your next CAN bus project using Arduino Uno and/or Raspberry Pi with CAN bus interface modules.

Dogan Ibrahim & Ahmet Ibrahim
London, 2022

Chapter 1 • Automotive Bus Systems

1.1 Overview

Today's vehicles are complex machines incorporating mechanical and electronic parts. The number of electronic components used in vehicles has rapidly increased in recent years. As a result of the increase in safety, comfort, and performance requirements, we see many more electronic components being added to modern vehicles. Henceforth, there has been an increasing demand to connect these electronic components in such a way that they can communicate with each other reliably, safely, and in real time.

In the past, electronic units used to be connected in a complex way, with hundreds of wires running to distinct parts of a vehicle. Consequently, it was difficult to trace an electronic fault. There was no coordination between various parts of the electronics, as each electronic part was controlled independently of the others. Maintenance and repair of the vehicle electronics were extremely difficult, as in many cases it was not easy to locate and change a faulty component.

Passenger safety has also become one of the most important considerations in today's vehicles. In the past decade, safety equipment has evolved and moved from physical to electronic assisted safety, starting with braking and type technology, through collision protection and airbags, and most recently, on to safety-related driver assistance systems. The latest modern vehicles are smart and intelligent machines and are equipped with many sensors that can evaluate the surroundings and display useful and safety-related information to drivers. These sensors form intelligent local networks together with actuators, displays, and fast digital processors, such as high-speed purpose-built microcontrollers.

Figure 1.1 shows a traditional old-style vehicle electronic system with sensors and actuators interconnected to each other in a complex manner. One of the major problems in this type of design is maintenance. The wiring was so complex that it was nearly impossible to trace and rectify faults.

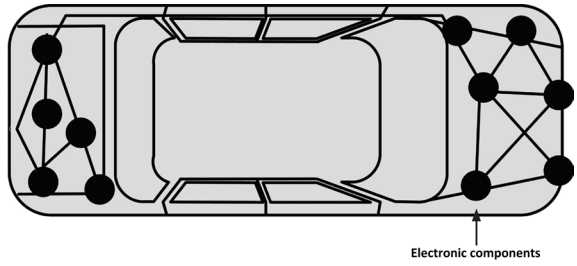


Figure 1.1: Traditional old-style vehicle electronics.

As the complexity of vehicle electronics has increased, it has become difficult for manufacturers to design safe and reliable electronic systems based on old traditional methods. Current requirements cannot be obtained using a simple electronic control unit. The solution is to interconnect the various electronic modules with a high-performance network. This is why it has become necessary to design a network-based electronic system where electronic

modules can easily be attached to a network and then controlled from a central, intelligent unit (e.g., Engine Control Unit). The result of this is an "intelligent" car where many sensors and actuators are used to sense the environment and perform multiple functions. An example is the automatic turning on of headlights when it becomes dark or when the car goes through a tunnel. Another example is the automatic operation of windscreen wipers when the rain begins, and so on.

One of the advantages of a network-based vehicle electronic system is that it is relatively easy to trace and detect a faulty module. In addition, the wiring is a lot simpler and easier to maintain. For example, by communicating with the central intelligent controller unit, one can tell whether or not the overall electronics system is healthy, and if not, the faulty modules can easily be detected. A networked system also allows the various modules on the bus to communicate with each other and exchange information if required. For example, the intelligence controller unit can receive the engine temperature value from the temperature sensor module. This temperature can then be displayed on an electronic dashboard. Should the temperature be too high, appropriate messages can be sent to the responsible parts of the engine and corrective measures can easily be taken. Figure 1.2 shows a modern vehicle where an electronic network system is used to interconnect and control the electronic modules.

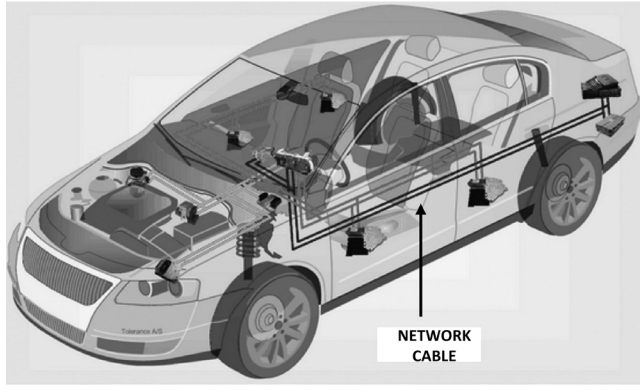


Figure 1.2: Modern vehicle with an electronic network.

This chapter provides an overview of the most important vehicle network systems currently used in vehicles. Moreover, it provides a table to compare the advantages and disadvantages of each system.

1.2 Vehicle network systems

Vehicle network systems (or networks) were classified in 1994 by the Society of Automotive Engineers (SAE). According to this classification, bus systems were classified based on their bandwidth (i.e., data rate) and functions of the network system. The classification divides bus networks into four: Class A, Class B, Class C, and Class D.

Class A networks are low-speed, low-cost networks with data rates of less than 10 kbps. These systems are mainly used in the body of the car.

Class B networks operate between 10 and 125 kbps and are used for information exchange, e.g., instrument cluster, vehicle speed, and so on.

Class C networks operate between 125 and 1 Mbps and are used in a wide range of applications, such as engine control.

Class D networks operate above 1 Mbps, and they are mainly used in telematics applications.

There are many automotive network systems (or bus systems, since the vehicle networks are in the form of bus wires), some developed by vehicle manufacturers on their own, and some developed jointly with semiconductor manufacturers. Popular vehicle network systems include:

- CAN bus
- LIN bus
- FlexRay
- MOST
- Byteflight
- DSI bus
- Intellibus
- SAE J1850
- BST bus
- NML bus
- And others…

In this section, we shall be looking at the brief details of the most commonly used automotive network systems listed below, and concentrate on the CAN bus, which is the topic of this book.

- LIN bus
- FlexRay
- MOST
- Byteflight
- Intellibus
- CAN bus

1.2.1 LIN bus

The Local Interconnect or "LIN" bus is a low-cost bus operating at 20 kbps that has been developed to create a standard for low-cost, low-end communication in automotive electronics. LIN fits in the low end of the automotive network systems, thus making it cost-effective. This bus is mainly used for non-important body/comfort functions. LIN is a single-wire, single-master/multiple-slave type bus system where the vehicle chassis is used as the return path. In a typical application, the master broadcasts a message with a message header, asking for data, and the slave that has the correct message header sends the requested data including a checksum for error checking. Typical LIN bus appli-

cations include the control of small motors for wipers, sunroof control, heating control, rain sensor control, steering wheel, seats, doors, etc. where high bandwidth is not required. The LIN bus is used in applications where the implementation of CAN bus would be too expensive. The initial LIN specification was defined by a consortium consisting of BMW, Audi, Volvo, VW, Motorola, Volcano, and DaimlerChrysler. The LIN bus is based on the Serial Communications Interface (UART) with 8-bit data.

1.2.2 FlexRay

As the demand for improved safety, increased performance, and enhanced comfort rises, then the need for more complex and more sophisticated electronic vehicle network systems arises. FlexRay was initially developed by BMW and Daimler Chrysler in 1999 as a fast, efficient, and error-free automotive bus system. FlexRay is suited to real-time, high-speed applications as it supports a bandwidth of up to 10 Mbps. Both electrical and optical transmission mediums can be used. FlexRay supports single and dual-channel configurations, with the dual channel offering enhanced fault-tolerance and increased bandwidth. Considered to be the next step beyond LIN and CAN, it is mainly used in safety-critical applications and in real-time high-speed engine control. Some application areas of FlexRay are: engine control, ABS, transmission control, break control, suspension control, etc.

1.2.3 MOST

The Media Oriented Systems Transport (MOST) bus is mainly used in automotive telemetric and multimedia applications, such as audio control, video, navigation, communication, and so on. Modern vehicle systems include multimedia devices such as DVD/CD players, TVs, navigation systems, graphics displays, and mobile computing which require very high data transfer rates. This is achieved using the MOST network. The original MOST network was developed by BMW and DaimlerChrysler in 1998.

MOST supports a maximum of 64 devices on the bus with very high bandwidth. Typical data rates are 28.8 Mbps synchronous, and 14.4 Mbps asynchronous. Up to 50 Mbps and even 150 Mbps are currently under development. An optical medium is used for data transmission that is free of any electromagnetic radiation or interference.

1.2.4 Byteflight

Byteflight was developed by BMW and offers 10 Mbps bandwidth. Byteflight is mainly used in safety-related networks, such as automotive and avionic systems (e.g., in vehicle airbags, body electronics, and so on). For it to be safe, the data protocol must be fault-tolerant and deterministic. Data control mechanisms before Byteflight had been either event-controlled or time-controlled. Event-controlled (e.g., CAN) only transmits data when data is ready or when a data request arrives. Time-controlled data protocols grant time to each node in accordance with a pre-defined sequence. The number of messages to be transmitted cannot generally be changed during operation, since the number of allocated time slots is fixed. The Byteflight protocol combines both event-controlled and time-controlled protocols and guarantees deterministic latencies for a specific number of high-priority messages, and flexible use of bandwidth for lower-priority messages.

The first use of the Byteflight in mass production was by BMW in their Series-7 cars and within a networked passive safety application. Byteflight supports various network protocols in a mixed bus environment.

1.2.5 Intellibus

Intellibus is a high-speed multi-drop communications bus offering up to 15 Mbps bandwidth. It was initially conceived by Boeing to reduce the wiring complexity associated with distributed systems in aerospace applications. It is a low-cost bus that allows a large number of sensors to be connected, and it is used mainly in engine control, transmission, and other parts requiring high speed. A typical Intellibus network consists of a Network Interface Controller (NIC) and 1 to 255 Intellibus Interface Modules (IBIM) that can be connected to sensors. The NIC can be installed on a PC or some other electronic device. Normally, the NIC is downloaded with software to sample data from various IBIMs in a scheduled manner, at specific intervals. In a complex network, two or more NIC cards can be used to increase node capacity. Special dedicated software is available to program the NIC and IBIMs. The maximum bus length is 30 m (at 12.5 Mbps and with 64 nodes). The Intellibus is used in automotive electronics, process control, automation, avionics, medical fields, and several other fields.

1.2.6 CAN bus

The Controller Area Network (CAN) bus is the main subject of this book. In this section, we examine the basic properties of this bus. Detailed descriptions of the CAN bus and CAN bus-related projects are provided in the remaining chapters of the book.

CAN is a serial, two-wire multi-master bus developed by Robert Bosch GmbH in the 1980s. It is one of the most widely used automotive electronic system buses today. The physical layer of CAN consists of a pair of twisted cables. CAN provides dependable, robust, and fast communication up to 1 Mbps (with a 40 m bus length). CAN 2.0A is the original CAN version consisting of the fields:

- Start of Frame bit
- 18-bits header (having 11-bit message identifier)
- 0-8 bytes data
- 15-bit Cyclic Redundancy Check (CRC)
- 3-bit acknowledgement slot
- 7-bit of End of Frame

The CAN bus is based on the CSMA/CR (Carrier Sense Multiple Access/Collision Resolution) "mechanism" to prevent frame collisions during transmissions on the bus. Each CAN node monitors the bus and when the node detects that the bus is idle, it may start transmitting data. If other nodes on the bus attempt to send data at the same time, arbitration will take place and the node with the highest priority (i.e., lowest message identifier) will win the arbitration and send its own data. The CAN bus has a simple error detection-and-recovery mechanism. Receiving nodes check the integrity of the messages by looking at the CRC fields. If an error is detected, the other nodes on the bus are informed through error flag messages. Figure 1.3 shows a typical CAN bus implementation with two nodes, A and B. CAN is a Class A/B type network.

Summary of CAN bus features:

- Multi-master bus where each node can be a master
- Multicast reception
- Speeds up to 1 Mbits/s
- Highly dependable

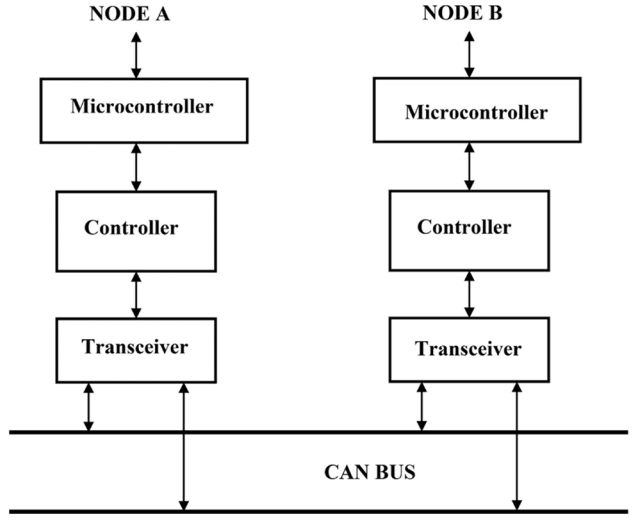


Figure 1.3: CAN bus with two nodes.

1.2.7 Others

Other automotive bus standards that should be mentioned in this chapter are the SAE J1850 (or simply J1850) which was developed in 1994, and the MI bus (Motorola Interconnect). J1850 was widely used in cars such as GM, Chrysler, and Ford, and this bus is used for diagnostics and data-sharing applications. There are two versions of this standard: PWM (Pulse Width Modulation) with 41.6 Kbps using a two-wire differential physical layer, and VPW (Variable Pulse Width) with 10.4 Kbps using a single-wire physical layer. The two standards are incompatible with each other. The J1850 protocol is frame-based and uses CSMA/CR arbitration where a frame consists of a Start of Frame, a header byte, data bytes, one byte CRC, and End of Data symbol (a 200-μs Low pulse). Most OBD (On Board Diagnostic) tools support the J1850 protocol for diagnostic purposes. The J1850 standard is old and is being phased out. The J1850 standard is a Class B type network.

The MI bus is a single-wire bus with one master and many slaves. This is a low-cost and low-data rate bus mainly used to drive seats, mirrors, etc. The master is the controller which sends addresses and data to all slaves on the bus. Slaves with matching addresses respond to the request.

1.3 Comparison of automotive bus systems

Currently, one of the most commonly used bus systems is the CAN bus, which can be used at speeds up to 1 Mbps.