Technical Note

TN_156

What is CAN?

Version 1.0

Issue Date: 2015-03-23

The purpose of this technical note is to explain the basics of CAN (Controller Area Network) which are a feature of FTDI's MCUs.

Future Technology Devices International Limited (FTDI)
Unit 1, 2 Seaward Place, Glasgow G41 1HH, United Kingdom
Tel.: +44 (0) 141 429 2777 Fax: + 44 (0) 141 429 2758
Web Site: http://ftdichip.com

Table of Contents

1 Introduction

CAN (Controller Area Network), sometimes referred to as CAN bus, is a communications protocol which has been around for years as a vehicle bus standard designed to allow MCUs to communicate with each other without the need for a host. Its use has since expanded and more industries and applications use it for MCU communication. See Typical Applications.

CAN communication is very robust with a significant amount of error checking in place. See Error Checking for more information.

The purpose of this technical note is to explain the basics of a CAN bus network and the key features.

2 CAN Bus

CAN is a multi-master broadcast serial bus standard for connecting MCUs together.

Each node, or connection to the bus, is able to send and receive messages, but not simultaneously. A message consists primarily of an ID (identifier), which represents the priority of the message, and up to eight data bytes. It is transmitted serially onto the bus in a non-return-to-zero (NRZ) signal pattern which is received by all nodes.

FTDI MCUs offer the CAN Specification 2.0 (parts A & B) variant, as defined by BOSCH Gmbh, which is backward compatible with v1.1b

Bit rates of up to 1 Mbit/second are possible at network lengths below 40m, with a maximum of 30 nodes in this configuration Note that speed and bus lengths are contributing factors to the maximum number of nodes.

An external physical layer device, also known as a PHY, is required to perform the physical signalling such as bit encoding and decoding (NRZ) as well as bit-timing and synchronization.

2.1 Signals

The following signals are used to create a CAN interface.

The MCU signals are connected to the PHY which performs the CAN encoding and decoding.

Typical Interconnect shows the basic hardware concept.

2.1.1 MCU Signals

2.1.1.1 Rx

The Receive (Rx) signal of CAN is the signal receiving the data. This will be an input to the MCU and is connected to an external physical layer device.

2.1.1.2 Tx

The Transmit (Tx) signal of CAN is the signal transmitting the data. This will be an output of the MCU and is connected to an external physical layer device.

2.1.2 PHY Signals

The CAN network is based on a half-duplex differential signal. There are two logical states: dominant and recessive. The figure below shows the general concept.

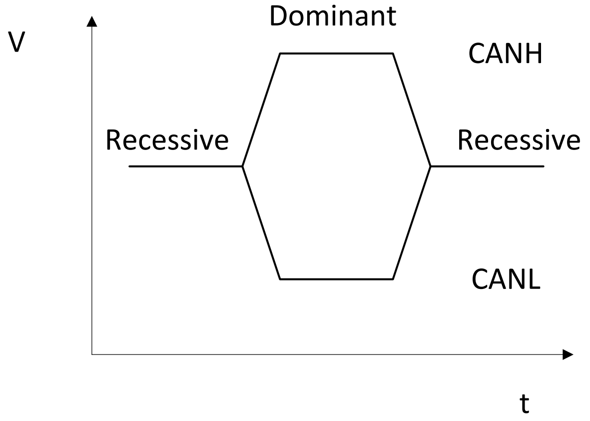

Figure 2.1 Differential Signal

2.1.2.1 CANH

This is the HIGH-level CAN bus line which is a differential signal.

2.1.2.2 CANL

This is the LOW-level CAN bus line which is a differential signal.

2.2 Typical interconnect

The figure below shows a typical MCU to PHY connection.

Note that the transmit and receive signals are not switched like some other communications protocol connections.

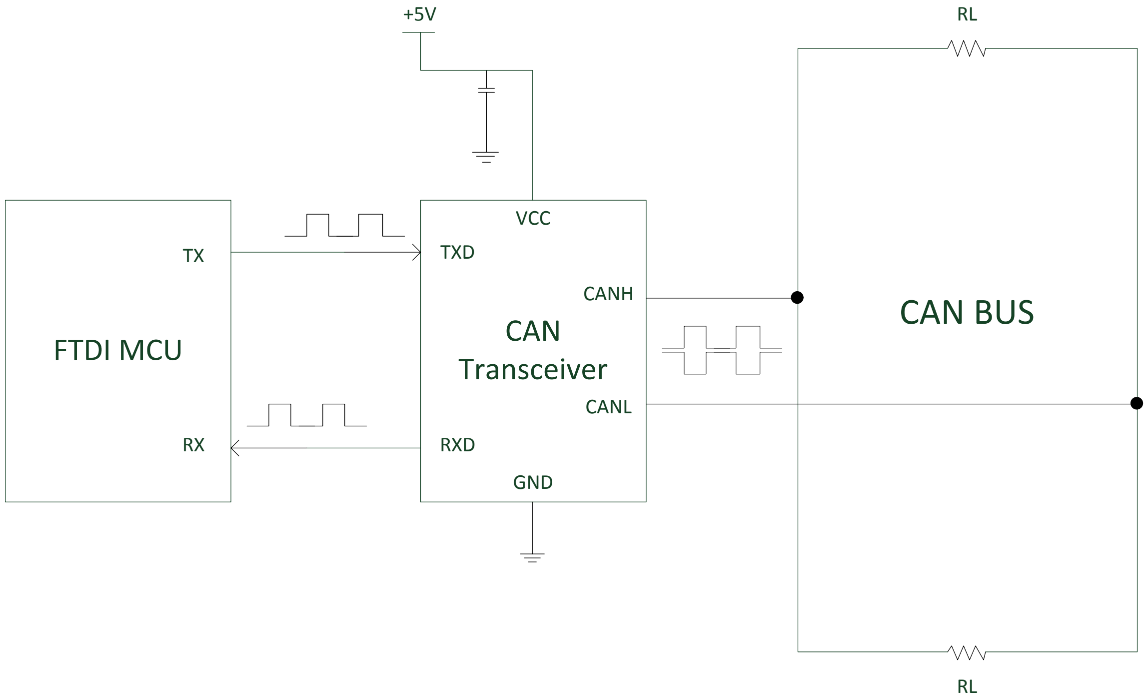

Figure 2.2 Typical Interconnect

2.3 Topology

Each node connected to the CAN bus is able to send and receive messages.

In order for the node to read the bus level correctly, the CAN bus must be terminated with termination resistors at both ends to avoid electrical signal reflections. Typically RL is 120R.

Twisted pair wires in a shielded cable should be used to minimise emissions and reduce interference.

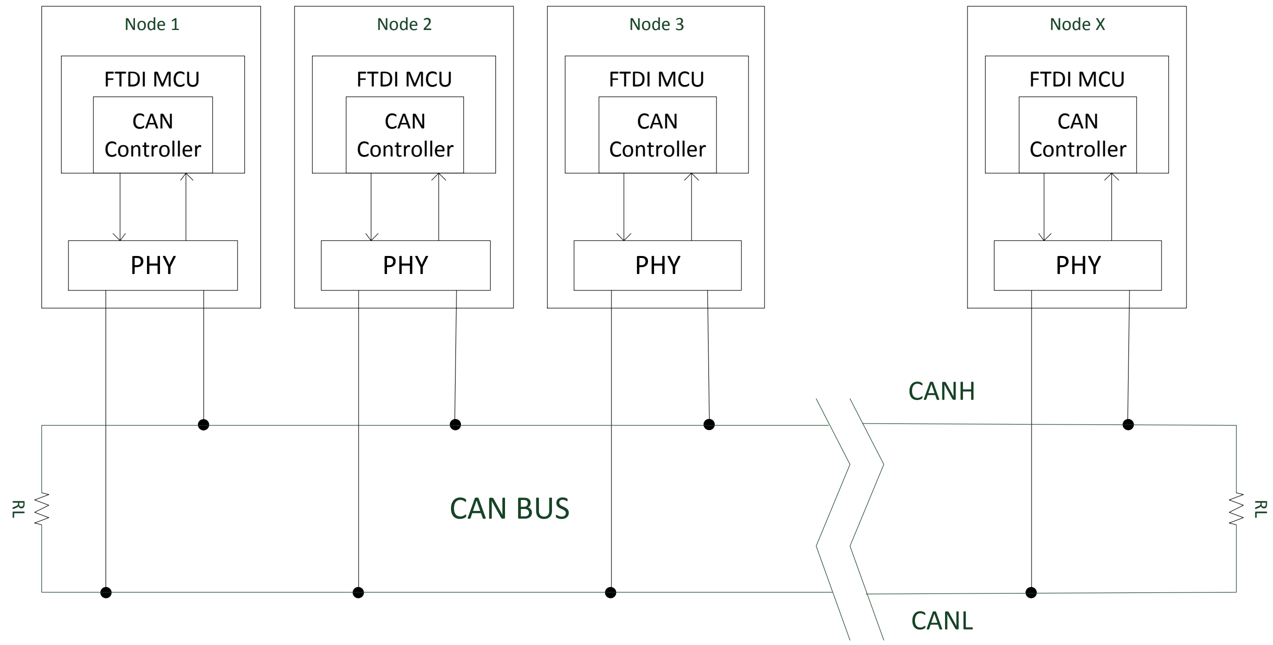

Figure 2.3 Topology

2.4 Message Format

There are two types of message (also known as 'frame') formats: standard and extended. The only difference between the two is the length of the identifier. Standard is 11 bits and extended is 29 bits in length. An IDE bit in the frame controls this selection, dominant for standard and recessive for extended.

CAN has four frame types which are discussed in this section.

2.4.1 Data Frame

This is a frame containing data for transmission.

2.4.1.1 Standard Frame

A standard frame structure looks like the following, with field descriptions in the table below.

S O F	Identifier	R T R	I D E	r0	DLC	Data (0-8 Bytes)	CRC	A C K	EOF	IFS

Field name	Length (bits)	Description
Start of Frame (SOF)	1	Denotes the start of frame transmission.
Identifier	11	An identifier for the data which also represents the message priority.
Remote Transmission Request (RTR)	1	Dominant. See Remote Frame subsection.
Identifier Extension (IDE)	1	Sets standard (0) or extended format (1).
Reserved bit (r0)	1	Reserved bit.
Data Length Code (DLC)	4	Number of bytes of data (0–8 bytes).
Data	0–64 (0-8 bytes)	Data to be transmitted.
CRC + delimiter	16	Cyclic Redundancy Check.
ACK slot + delimiter	2	Transmitter sends recessive and receiver asserts dominant on successful reception.
End Of Frame (EOF)	7	Marks the end of a CAN frame.
Interframe Space (IFS)	7	Contains the time required by the controller to move a correctly received frame to its proper position in a message buffer area.

Table 2.1 Standard Frame

2.4.1.2 Extended Frame

An extended frame structure looks like the following, with additional field descriptions in the table below.

S O F	Identifier	S R R	I D E	Identifier 2	R T R	r1	r0	DLC	Data (0-8 Bytes)	CRC	A C K	EOF	IFS

Field name	Length (bits)	Description
Identifier 2	18	Additional 18 bits
Substitute Remote Request (SRR)	1	Replaces RTR in the standard message frame.
Reserved bit (r1)	1	Reserved bit.

Table 2.2 Extended Frame

2.4.2 Remote Frame

A destination node can request data from the source by sending a Remote Frame request.

RTR is set to 1 and there is no data field.

2.4.3 Error Frame

The error frame consists of error flags (6–12 bits) followed by the error delimiter (8 recessive bits).

2.4.4 Overload Frame

An overload frame is similar to the error frame with regard to the format, and it is transmitted by a node that becomes too busy.

2.4.5 Bus Arbitration

The CAN protocol uses a modified version of the Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) technique used on Ethernet. Should two messages determine that they are both trying to send at the same time then instead of both backing off and re-trying later as is done with Ethernet, in the CAN scheme, the nodes detect which message has the highest priority and only the lower priority message gets delayed. This means that a high priority message is sure of getting through. The node transmitting the lower priority message automatically attempts to re-transmit 6 bit clocks after the end of the dominant message.

2.5 Error Checking

A key feature of CAN is error checking which incorporates five methods of error checking: three at the message level and two at the bit level. If a message fails any one of these error detection methods, it is not accepted and an error frame is generated from the receiving node. This forces the transmitting node to resend the message until it is received correctly. However, if a faulty node hangs up a bus by continuously repeating an error, its transmit capability is removed by its controller after an error limit is reached.

2.5.1 Message level

Error checking at the message level is enforced by the CRC and the ACK slots in the data frame shown previously.

The 16-bit CRC contains a checksum of the preceding data for error detection.

The ACK bit is used by the transmitter to send a recessive bit and receiver asserts dominant on successful reception.

Fields in the message which must always be recessive bits are checked. If a dominant bit is detected, an error is generated.

2.5.2 Bit level

At bit level each bit transmitted is monitored by the transmitter of the message. If a data bit is written onto the bus and its opposite is read, an error is generated.

CAN also inserts a bit of opposite polarity after five consecutive bits of the same polarity due to NRZ coding. If the next bit is not a complement, an error is generated.

3 Comparison

The table below gives a quick guide to the differences between some key MCU to MCU communication modules.

	CAN	**UART**	**I²C**	**SPI**
Typical Speed	1Mbps	3-12Mbps	3.4Mbps	2-15Mbps
Minimum Number of MCU Signals	2	2	2	4
External PHY Required?	Yes	No	No, but pull-ups required	No
Communication Style	Half-duplex	Full-duplex	Half-duplex	Full-duplex
FTDI MCU Availability	Yes	Yes	Yes	Yes

Table 3.1 Communication Comparison

4 Typical Applications

The main application, and reason for the CAN bus design, is automotive. CAN bus allows various subsystems such as the main Engine Control Unit (ECU), airbags, Anti-lock Braking System (ABS), power steering, etc to be connected together.

However, now there are multiple applications and manufacturers who use CAN as an embedded network due to its reliability and feasibility. Some examples include:

- Industrial and automation (eg Packaging and printing)
- Avionics (eg flight state sensors and navigation systems)
- Aerospace (eg in-flight data analysis to aircraft engine control systems such as fuel systems and pumps)
- Railway (eg trams, undergrounds, light railways, and long-distance trains)
- Medical (eg lights, tables, cameras, X-ray machines and even beds)
- Lifts and escalators
- Laboratory Equipment
- Sports cameras
- Automatic doors
- Telescopes
- Coffee machines

The limits are endless for CAN bus applications due to the low cost of CAN controllers and processors!

5 What FTDI offer

At the time of writing, FTDI offer an FT900 - 32 Bit RISC MCU. With a parallel camera input, 10/100 base Ethernet interface, CAN bus, and USB2.0 Hi-Speed peripheral and host ports, the FT900 offers excellent interconnect capabilities and computational power.

There are 2 x CAN 2.0 controllers on chip.

The FT900, FT901, FT905, FT907 devices all have CAN bus controllers on chip. Other variants of FT900 do not have CAN bus controllers.

6 Contact Information

Head Office – Glasgow, UK

Future Technology Devices International Limited
Unit 1, 2 Seaward Place, Centurion Business Park
Glasgow G41 1HH
United Kingdom
Tel: +44 (0) 141 429 2777
Fax: +44 (0) 141 429 2758

E-mail (Sales) sales1@ftdichip.com
E-mail (Support) support1@ftdichip.com
E-mail (General Enquiries) admin1@ftdichip.com

Branch Office – Tigard, Oregon, USA

Future Technology Devices International Limited (USA)
7130 SW Fir Loop
Tigard, OR 97223-8160
USA
Tel: +1 (503) 547 0988
Fax: +1 (503) 547 0987

E-Mail (Sales) us.sales@ftdichip.com
E-Mail (Support) us.support@ftdichip.com
E-Mail (General Enquiries) us.admin@ftdichip.com

Branch Office – Taipei, Taiwan

Future Technology Devices International Limited (Taiwan)
2F, No. 516, Sec. 1, NeiHu Road
Taipei 114
Taiwan , R.O.C.
Tel: +886 (0) 2 8791 3570
Fax: +886 (0) 2 8791 3576

E-mail (Sales) tw.sales1@ftdichip.com
E-mail (Support) tw.support1@ftdichip.com
E-mail (General Enquiries) tw.admin1@ftdichip.com

Branch Office – Shanghai, China

Future Technology Devices International Limited (China)
Room 1103, No. 666 West Huaihai Road,
Shanghai, 200052
China
Tel: +86 21 62351596
Fax: +86 21 62351595

E-mail (Sales) cn.sales@ftdichip.com
E-mail (Support) cn.support@ftdichip.com
E-mail (General Enquiries) cn.admin@ftdichip.com

Web Site

http://ftdichip.com

Appendix A – References

Document References

FTDI MCUs

Bosch CAN Literature

Acronyms and Abbreviations

Terms	Description
CAN	Controller Area Network
MCU	Microcontroller Unit
RISC	Reduced Instruction Set Computer
USB	Universal Serial Bus
PHY	Physical Layer Device
NRZ	Non Return to Zero
Rx	Receive
Tx	Transmit
CRC	Cyclic Redundancy Check
ACK	Acknowledgement
Mbps	Mega bits per second
UART	Universal Asynchronous Receiver/Transmitter
I^2C	Inter-Integrated Circuit
SPI	Serial Peripheral Interface
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance

Appendix B – List of Tables & Figures

List of Tables

List of Figures

Appendix C – Revision History

Document Title: TN_156 What is CAN?

Document Reference No.: FT_001079

Clearance No.: FTDI# 441

Product Page: http://www.ftdichip.com/FTProducts.htm

Document Feedback: Send Feedback

Revision	Changes	Date
0.0	Initial draft.	2014-06-27
1.0	Release using new template.	2015-03-23