
Author	Bruce Emaus
Restrictions	Public Document
Abstract	This application note explains the basic concepts of CAN bit timing with emphasis on sending a message for the first time.

Table of Contents

1	Overview	1
2	CAN Bit Timing Requirements	2
2.1	Bit Rate	2
2.2	Bit Rate Accuracy	2
2.3	Sample Point	2
2.4	Sampling Mode	3
2.5	Resynchronization Jump Width	3
3	General CAN Bit Timing Rules	3
3.1	Number of Time Quanta	3
3.2	Timing Accuracy	3
3.3	Sample Mode	3
3.4	Sample Point	3
3.5	SJW	4
3.6	TSEG2	4
4	Implementing Bit Timing	4
4.1	Establishing the Time Quantum	4
4.2	Establish the Time Segment	5
4.3	Setting the Correct Bit Timing Register Values in Silicon	5
5	Solving Implementation Problems	6
5.1	No Transceiver Equals No Transmission	6
5.2	CAN Acknowledgment – One Node Equals Transmission Forever	6
6	Use an Oscilloscope	7
6.1	Connecting to the CAN Bus	7
6.2	Connecting to the TX Pin	8
7	Questions and Answers	8
8	Additional Resources	9
9	Contacts	9

1 Overview

For the beginning CAN developer, setting up the proper bit timing parameters is often frustrating. This industry-wide and quite common experience results from the rather large amount of complexity that surrounds bit timing. Many variables and considerable variation across the CAN Controller implementations have all contributed to the problem. The level of complexity of the mathematics behind the CAN protocol's bit timing is astounding.

Most first time CAN developers select bit timing parameters that work quite well on the bench, but when the distributed product moves into the “real” environment (maximum wiring length, worst case configuration, etc.), the parameters selected earlier are found to be inadequate. This application note should provide some assistance in resolving this problem.

Debugging CAN bit timing will often require some knowledge of the CAN protocol and the Physical Layer selected. Understanding the principles of dominant/recessive logic, bus termination, and the special meaning of CAN acknowledgement are some of the concepts that will help in the process. Taking CAN training classes to learn more about these details is worth considering.

2 CAN Bit Timing Requirements

CAN bit timing usually begins with five essential system requirements:

- Bit rate
- Bit rate accuracy
- Sample point
- Sample Mode
- Re-synchronization Jump Width

2.1 Bit Rate

The CAN communication bit rate is usually pre-determined by the systems engineering activity, or for some protocols (like SAE J1939) it is selected by an industry-sponsored committee.

2.2 Bit Rate Accuracy

The accuracy of the bit rate requires little attention if a crystal-based clock is used. However, the lower cost ceramic resonator may be a possibility for slower speed CAN networks. Most automotive applications use a crystal.

Important: Keep crystals away from heat sources!

2.3 Sample Point

The often-overlooked sample point is an important parameter. Like the bit rate, the value of the sample point is usually selected by an industry standards group or by the systems engineering section of the company creating the distributed product.

Contrary to the 50% sample point of the common serial port or UART, (Universal Asynchronous Receiver Transmitter), the structure of the CAN protocol places the optimal sample point much closer to the end of the bit period. Sample points in the high 80% range are required for industry standards J1939 (250K BPS) and J2284 (500K BPS).

When this important parameter is overlooked, it is common for the developer to arbitrarily pick a value (since some value must be chosen and inserted into the CAN Controller).

Arbitrary values like 50% or 75% all seem to work fine in the lab on the bench top with short wires, but when the module is placed into its real system environment, the effects of longer wires and increased capacitance may cause communication errors that result from an inadequate sample point.

Note: Use sample points of 80% to 90%.

2.4 Sampling Mode

The sample mode is a programmable state set in the CAN Controller that is used to establish the selected form of bit sampling. Typically one of two forms of bit sampling is handled by the CAN silicon. This is done by either a “single sample” or a “2 of 3 majority-rules sample”.

While the selection of the sample mode is usually determined by an industry standards group or by the systems group, some CAN Controllers may only support the “single sample” mode. In general, the “single sample” mode is adequate for most applications. Based on the automotive analysis of high-speed CAN applications “single sample” mode is the only choice.

Note: Use single sample mode.

2.5 Resynchronization Jump Width

The Resynchronization Jump Width (or SJW) is another complex parameter. This somewhat mysterious timing element is again selected by the systems engineering activity or by a committee when a standards activity is involved.

The Resynchronization Jump Width is a variable portion of the CAN bit timing that is programmable in the CAN Controller and is used to automatically compensate for timing variations between nodes. The Resynchronization Jump Width is set by the user's selected parameters that are implemented in the CAN Controller.

Having a correct Resynchronization Jump Width is important. It is preferable if this value is in the governing specification rather than having to determine this value mathematically.

Note: It is recommended that a large Resynchronization Jump Width be used.

3 General CAN Bit Timing Rules

The following general rules are extracted from the detailed analysis of bit timing, which can be found in the application note AN-AND-2-135 “Advanced CAN Bit Timing”.

3.1 Number of Time Quanta

The number of TIME_QUANTA in a bit period must be an integer value. Choose a value for the TIME_QUANTUM as small as practical - the smaller the TIME_QUANTUM the better the resolution in selecting the location of the sample point in the bit period as well as the size of the SJW interval.

Although the allowable range of TIME_QUANTA is typically between 8 and 25 - select a value as close to 25 as possible.

3.2 Timing Accuracy

For the system-level oscillator tolerance for the entire CAN communication system, use crystals to avoid the difficult analysis of bit timing details.

3.3 Sample Mode

Because a sample mode of three reduces the amount of allowable system propagation delay by approximately two tQ, use the single sample mode for high-speed networks.

3.4 Sample Point

As the maximum system propagation delay increases, the size of the TSEG2 interval must decrease, which corresponds to moving the sample point closer to the end of the bit period. A good general rule is to set the sample point above 80%.

3.5 SJW

The maximum number of TIME_QUANTA which can be allocated to the SJW interval is 4, as defined by the CAN protocol specification. The minimum SJW interval must be equal to or greater than the maximum system propagation delay. Set SJW to a large value.

3.6 TSEG2

While the selected TSEG2 interval must be at least 2 TQ or equal the size of the SJW interval, to provide the maximum noise margin against noise events, the largest possible value should be chosen for the TSEG2 interval. Set TSEG2 equal to SJW.

4 Implementing Bit Timing

Using the CAN Controller's clock and clock divider, the Bit Time Interval is implemented as the number of user-designated Time Quanta as shown in Figure 1. There is a complicated side to all this – but this discussion will focus on minimum detail.

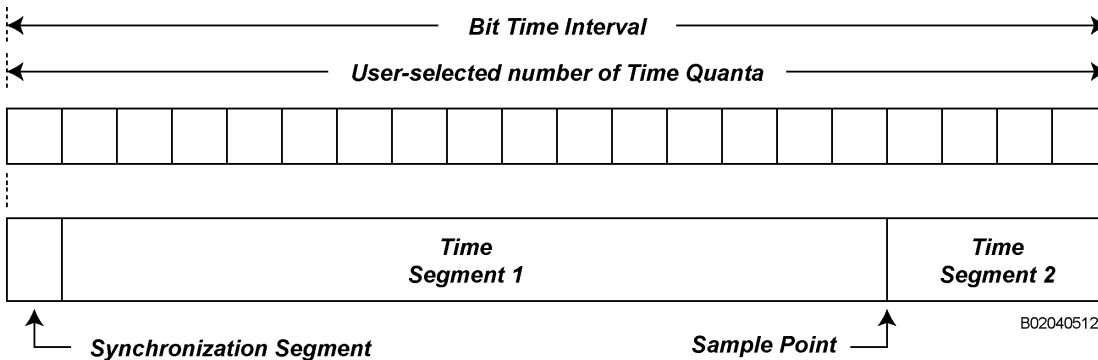


Figure 1 – Implementing CAN Bit Timing Using the Time Quanta

4.1 Establishing the Time Quantum

The typical CAN Controller provides three key register values that establish a major portion of the bit timing.

- CAN clock divider or Bit Rate Prescaler
- Time Segment 1
- Time Segment 2

The CAN clock divider or Bit Rate Prescaler (BRP) is used to divide the microcontroller's main internal clock into a suitable clock rate. This clock establishes the "time quantum" or TQ value.

Important: Always use an exact integer number of TQ's to create the correct bit time interval.

For example, beginning with a 20 MHz clock oscillator and a required CAN bit rate of 500K BPS (Bits Per Second), a divide by 2 value for the CAN clock divider will need 20 TQ increments to establish the 2 microsecond bit time interval.

Oscillator	= 20 MHz
Clock Divider	= 2

TQ = 100 nsec
 Bit Rate = 500K BPS or 2 microseconds
 Number of Required TQs = 2 microseconds / 100 nsec = 20

4.2 Establish the Time Segment

Now set the values of Time Segment 1 and Time Segment 2. This is accomplished by using the following formula –

Number of Required TQs = 1 + Time Segment 1 + Time Segment 2

While it seems relatively easy to set the Time Segment 1 and Time Segment 2 values to fix the bit rate, the selected values also are used to establish the “sample point”. Based on the experiences of the automotive industry, the selected sample point should be near the 80% value.

Next, establish the sample point at 80 percent.

Time Segment 1 = 15
 Time Segment 2 = 4
 Sample Point = $(1 + \text{Time Segment 1}) / (1 + \text{Time Segment 1} + \text{Time Segment 2})$
 = $(1 + 15) / (1 + 15 + 4)$
 = 0.80 = 80%

To complete the bit timing parameters, use “single sample” mode and set a value of 3 for the Resynchronization Jump Width.

Single Sample Mode = yes
 Resynchronization Jump Width = 3

4.3 Setting the Correct Bit Timing Register Values in Silicon

Remember that the silicon registers containing the bit timing parameters are usually implemented as binary values “one less” than the calculated values.

<i>parameter</i>	<i>calculated value</i>	<i>register value</i>
Clock Divider	2	1
Time Segment 1	15	14
Time Segment 2	4	3
Re-synchronization Jump Width	3	2

Table 1 – Difference between Calculated and Register Values

Also, consider getting the register values directly from the application engineer of the semiconductor company. They know their silicon. They should be able to deliver the correct register values that will meet the specific bit timing requirements.

5 Solving Implementation Problems

5.1 No Transceiver Equals No Transmission

Attempting to transmit a CAN message for the first time, even with all the correct bit timing parameters, will fail if the CAN Controller is not connected to a transceiver (or equivalent). Connecting a scope to the transmit pin (TX) of the CAN Controller and initiating a message transmission will yield a single pulse (high-low-high) - the expected message bit pattern is not detected.

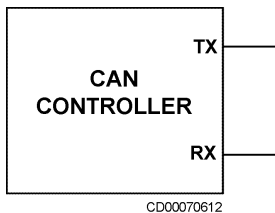


Figure 2 – The Unconnected CAN Controller will not Send a Message

The circuit shown in Figure 2 produces no CAN communication because there is no connection to a Physical Layer. When the CAN Controller starts its transmission by sending out a dominant bit to signal the “Start of Frame” bit, the sampling of the receive pin does not match. Therefore, the CAN Controller abandons the transmission.

Two solutions are available. Connect the CAN Controller to a real transceiver or “force the transmit bit back into the receive pin”.

5.2 CAN Acknowledgment – One Node Equals Transmission Forever

If you connect the CAN Controller to a transceiver chip without connecting the CAN bus to another node (see Figure 3) and send a message, guess what happens - the transmission will continue forever because no other CAN node will acknowledge the message.

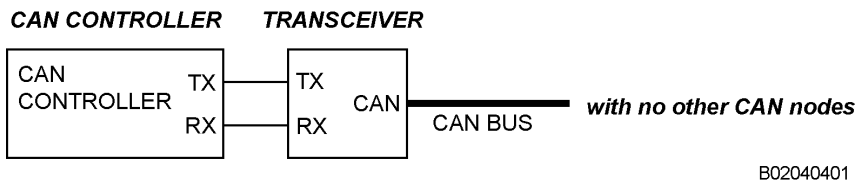


Figure 3 – The “Transmit Forever” Configuration with Transceiver

Why does this transmission continue forever? Since no other receiving node is present to acknowledge the message, the transmitter will automatically try to send the message again. Since the transmitter is waiting for a node to acknowledge its message, this process repeats forever. CAN Acknowledgment occurs near the end of every CAN message and is used to establish a level of message integrity between the message transmitter and ALL message receivers. Because all receivers must participate in the acknowledgment algorithm regardless of whether the message is intended for them or not, an acknowledgment to the transmitter may occur even if the expected receiver is not present on the network. This means that the CAN Acknowledgment does not guarantee that a data transfer has occurred between the transmitter and a designated receiver. It does not confirm that a requested action has been understood or performed. CAN Acknowledgment only confirms that all resident network nodes agree that the CAN message did not violate any Data Link Layer rules.

Even without a transceiver, when the developer “forces the transmit bit back into the receive pin” as shown in Figure 4, the same behavior occurs.

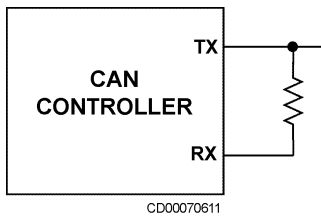


Figure 4 – The “Transmit Forever” Configuration Without Transceiver

For preliminary breadboard experimentation without using a transceiver, a wire or resistive connection of the TX pin to RX pin will satisfy the CAN protocol rule (bit-wise arbitration requirement) and provides a method to play with transmitting a message. This connection provides an opportunity for the receive pin to see the same bit pattern sent out on the transmit pin. Because each transmitted bit is detected at the receive pin as identical, a complete message transmission will occur and because no other node will acknowledge the message, the transmission continues forever.

6 Use an Oscilloscope

Using the oscilloscope can easily determine:

- if a message is actually being output
- the approximate bit rate

Since the CAN message is always preceded by at least an idle time of 11 bit periods in the recessive state, the first dominant bit of the CAN message, called the Start of Frame (SOF), is easy to trigger on with the oscilloscope. Using a transmitted message with an 11-bit identifier of 555 hex as shown in Figure 5 provides a simple recognizable waveform with a recurring one-zero pattern that is easy to detect and trigger on.

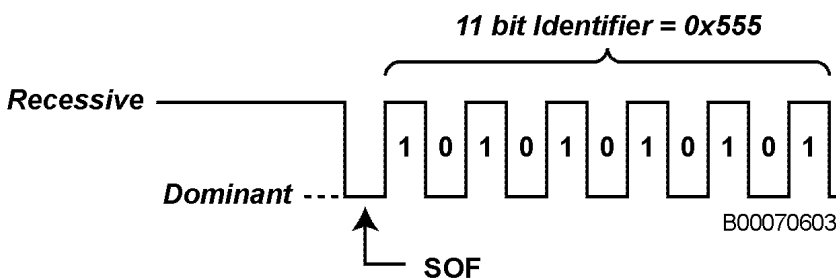


Figure 5 – CAN Waveform Using Identifier 555 Hex

Because of the output driver characteristics of the typical transceiver, oscilloscope triggering on the falling edge (recessive-to-dominant) of the CAN bus signal produces the most stable waveform.

The width of any of these early waveform bits is equal to one bit time interval (BTI). For a 500KBPS bit rate the bit duration is 2 microseconds. Whether correct or incorrect time segment values have been loaded into the CAN Controller timing registers, it is not too difficult to determine by looking at the scope.

6.1 Connecting to the CAN Bus

Connecting an oscilloscope to the CAN bus (as shown in Figure 6) provides an easy way to monitor the adjustment of CAN bit timing.

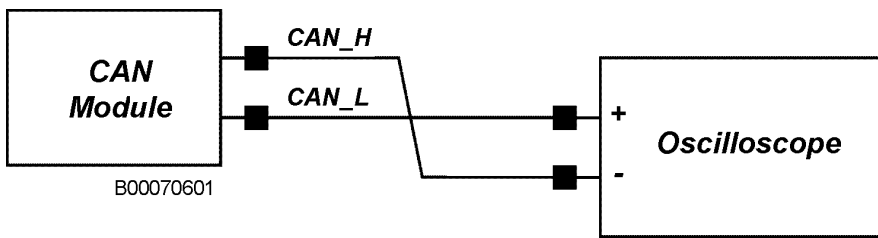


Figure 6 – Basic Oscilloscope Connection

Why a positive connection to the CAN_L rather than the CAN_H signal wire? Because the CAN Controller's transmit pin (TX) uses logic level 0 to indicate the "dominant" bit state, it is convenient to maintain this logic convention when viewing CAN waveforms. The same logic convention can be used in this manner whether examining signals on the bus or at the pins of the CAN Controller. LOW is dominant and HIGH is recessive.

If longer wire connections are used, twisted pair cabling with termination resistors may need to be used to get stable signals as shown in Figure 7.

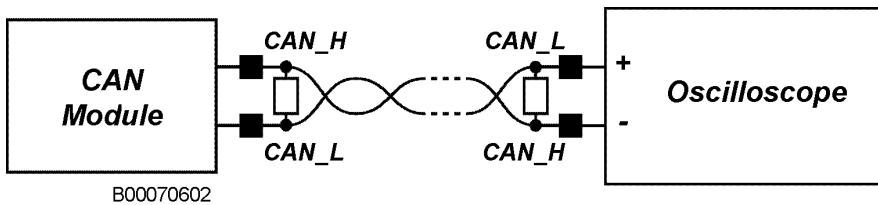


Figure 7 – Oscilloscope Connection Using Twisted Pair Cabling

6.2 Connecting to the TX Pin

Connection to the CAN Controller transmit pin (usually labeled as TX) is also another usable point to examine the CAN transmission. The TX pin operates with the traditional 5 volt logic and makes the signal triggering somewhat easier when compared to the differential signals that are on the CAN bus.

7 Questions and Answers

Q1 – How do the car companies determine bit timing?

A1 – Typically, the car company uses its experts to establish the bit timing parameters for their particular environment.

Wiring length, EMC, and a host of various issues influence bit timing analysis and testing to find suitable solutions. For high speed automotive applications, SAE J2284, an industry-level physical layer specification, establishes CAN bit timing parameter requirements used by DC, Ford, and GM. Each car company requires every electronic module supplier to use the same values.

Q2 – How should a non-automotive company select its bit timing parameters?

A2 – It is perhaps easier to start with an industry-level standard, then determine whether to use it in its entirety or to deviate slightly. For example, the NMEA2000 Physical Layer uses elements from J1939 and also DeviceNet to meet its marine application requirements.

Q3 - How can I learn more about the details of CAN bit timing?

A3 – Consider the following application note –

AN-AND-2-135 "Advanced CAN Bit Timing"

8 Additional Resources

The following material may provide further useful information -

SAE DOCUMENTS

J2284 Recommended Practice

SAE technical paper #970295 "CAN Bit Timing Requirements" by Karl Overberg & Klaus Dietmayer

OTHER RESOURCES

Philips application note AN97046 "Determination of Bit Timing Parameters for SJA 1000 CAN Controller" by Egon Johnk & Klaus Dietmayer

9 Contacts

Germany and all countries not named below:

Vector Informatik GmbH

Ingersheimer Str. 24

70499 Stuttgart

GERMANY

Phone: +49 711-80670-0

Fax: +49 711-80670-111

Email: info@vector-informatik.de

France, Belgium, Luxemburg:

Vector France SAS

168 Boulevard Camélinat

92240 Malakoff

FRANCE

Phone: +33 1 42 31 40 00

Fax: +33 1 42 31 40 09

Email: information@vector-france.fr

Sweden, Denmark, Norway, Finland, Iceland:

VecScan AB

Theres Svenssons Gata 9

41755 Göteborg

SWEDEN

Phone: +46 31 764 76 00

Fax: +46 31 764 76 19

Email: info@vecscan.com

United Kingdom, Ireland:

Vector GB Ltd.

Rhodium

Central Boulevard

Blythe Valley Park

Solihull, Birmingham

West Midlands B90 8AS

UNITED KINGDOM

Phone: +44 121 50681-50

Email: info@vector-gb.co.uk

USA, Canada, Mexico:

Vector CANtech, Inc.

39500 Orchard Hill Pl., Ste 550

Novi, MI 48375

USA

Phone: +1 248 449 9290

Fax: +1 248 449 9704

Email: info@vector-cantech.com

Japan:

Vector Japan Co. Ltd.

Seafort Square Center Bld. 18F

2-3-12, Higashi-shinagawa,

Shinagawa-ku

Tokyo 140-0002

JAPAN

Phone: +81 3 5769 7800

Fax: +81 3 5769 6975

Email: info@vector-japan.co.jp

Korea:

Vector Korea IT Inc.

#1406 Mario Tower

222-12 Guro-dong, Guro-gu

Seoul 152-848

REPUBLIC OF KOREA

Phone: +82 2 2028 0600

Fax: +82 2 2028 0604

Email: info@vector-korea.com