Configure Controller Area Network (CAN) Bit Timing to Optimize Performance

By Dr. Conal Watterson

Introduction

Controller area network (CAN) offers robust communication between multiple network locations, supporting a variety of data rates and distances. Featuring data-link layer arbitration, synchronization, and error handling, CAN is widely used in industrial, instrumentation, and automotive applications. Standardized under ISO 11898, with distributed multimaster differential signaling and built-in fault handling, various protocols such as DeviceNet and CANopen define implementations for physical and data-link layers. This article describes how to optimize settings for a given application, considering hardware restrictions such as controller architecture, clocks, transceivers, and logic-interface isolation. It addresses network configuration—both data rate and cable length—explaining when reconfiguration of CAN nodes is necessary and how nodes can be optimally configured from the beginning.

Isolating the Logic Interface

For harsh industrial and automotive environments, system robustness can be further enhanced by isolating the logic interface to the CAN transceiver, allowing large potential differences between ground nodes and providing immunity to high-voltage transients. An isolated CAN node can be formed by integrating a CAN transceiver with digital isolators. The ADM3052, ADM3053, and ADM3054 isolated CAN transceivers provide various options for powering the interface. For DeviceNet networks, the isolated side can be powered from the bus, so the ADM3052 integrates a linear regulator to provide a 5-V supply from the 24-V bus supply. The ADM3053, shown in Figure 1, integrates an isoPower dc-to-dc convertor to power the transceiver and bus side of the digital isolator. Systems that already have an isolated dc-to-dc convertor to provide power across the barrier can use the ADM3054, which integrates only the digital isolator and CAN transceiver.



Figure 1. Isolated CAN node with ADM3053 transceiver powered by *iso*Power dc-to-dc.

Effect of Propagation Delay

Implementing a CAN node requires an isolated or nonisolated CAN transceiver and a CAN controller or processor with the appropriate protocol stack. Standalone CAN controllers can be used, even without a standard protocol stack, but microprocessors used in CAN applications may already include the CAN controller. In either case, the CAN controller must be configured to reconcile the data rate and timing on the bus with the hardware oscillator used for the controller.

As cable length increases, the high frequency content of the signal is attenuated, so data rates are limited for long distances. The bus is a multimaster, so all nodes can attempt transmission at the same time, and arbitration depends on the physical layer signaling. Propagation delay, which also increases with cable length, can interfere with the synchronization and arbitration between nodes.

The differential signal on a CAN bus can be one of two states: dominant (Logic 0, a differential voltage between signal lines CANH and CANL) or recessive (Logic 1, no differential voltage, all CAN transceiver outputs are high impedance). If two nodes attempt to transmit at the same time, a dominant bit transmission will overwrite a concurrent recessive bit transmission, so all nodes must monitor the bus state while transmitting and cease transmission if overwriting occurs when they transmit a recessive bit. Thus, the node transmitting the dominant bit wins arbitration, as shown in Figure 2.



Figure 2. Arbitration logic between two CAN nodes.

CAN 2.0b, which defines the implementation of the data-link layer, specifies the structure of CAN frames used for transmission. An arbitration field, which includes a message ID, starts the message. A lower message ID (more initial zeroes) will have higher priority, so the node is more likely to win arbitration when transmitting a message. Although CAN nodes synchronize to the bus transmissions, two nodes that transmit together will not be exactly concurrent due to the propagation delay between them. For arbitration to work, the propagation delay must not be too large, or the faster node may sample the bus before detecting the bit state transmitted by the slower node. The worst-case propagation delay is twice the delay between the two furthest nodes. If nodes A and B in Figure 3 are the furthest apart on the bus, the critical parameter is the round trip time of T_{PropBA} .



Figure 3. Arbitration between two CAN nodes in practice, with propagation delay.

The total propagation delay comprises the round trip time through the cable, two CAN controllers I/O, and two CAN transceivers. The CAN controller I/O is not a major contributor, and can often be ignored, but a thorough assessment will allow for it. Loop time comprises the propagation delay from TxD to CANH/CANL and back to RxD. The cable propagation delay depends on the cable and distance, with 5 ns/m being a typical value.

At lower data rates, the allowed bit time is longer, so the propagation delay (and thus the cable distance) can also be longer. At the maximum standard CAN data rate of 1 Mbps, the allowed propagation delay is more limited, though ISO 11898-2 specifies a bus length of 40 meters for operation at 1 Mbps.

Impact of Isolation

With isolation, an extra element must be considered in the round trip propagation delay calculation. Digital isolators reduce the propagation delay compared to optocouplers, but even the fastest isolated CAN transceivers will be comparable to the slower nonisolated transceivers. If the total allowable propagation delay remains the same, the maximum cable length will be shorter in isolated systems, but it may be possible to reconfigure the CAN controller to increase the total allowed propagation delay.

Compensating for Propagation Delay

To compensate for propagation delays added by longer buses or isolation, specific parameters related to timing and synchronization must be set up for the CAN controller. When configuring the controller, instead of simply selecting a data rate, set the variables that dictate the bit time used by the controller. The baud rate prescaler (BRP) for the oscillator or internal clock sets the time quantum (TQ), and the bit time is a multiple of TQ. The hardware selection of the oscillator, and the software configuration of the BRP and number of TQ per bit time set the data rate.

The controller's bit time is broken into three or four segments as shown in Figure 3. The total number of TQ per bit time includes one for synchronization, plus programmed numbers for propagation delay (PROP), Phase Segment 1 (PS1), and Phase Segment 2 (PS2). Sometimes PROP and PS1 are combined. Configuration adjusts the sample point to allow for propagation delay and resynchronization.

Setting the sample point later in the bit time allows for longer propagation delays, but like the overall data rate, the sample point is dependent on other timing variables, each of which has its own restrictions. For example, the internal clock/oscillator may be fixed and only integer BRP and TQ numbers can be used. Thus, the ideal data rate required for a certain cable length may be impossible to achieve, so either the cable must be shortened or the data rate must be reduced.

Resynchronization results in PS1 being lengthened or PS2 being shortened by the number of TQ specified by the synchronization jump width (SJW), so PS2 can never be shorter than SJW. The number of TQ required for SJW depends on the clock tolerance of CAN controllers, with crystal oscillators typically allowing the minimum TQ for SJW and PS2.

Configuring a CAN Controller

To achieve a robust network with reliable timing and synchronization between nodes, the system must be able to tolerate the propagation delay with the chosen data rate and CANcontroller clock. If it this is not possible, the options are to reduce the data rate, shorten the bus, or use a different CANcontroller-clock rate. A three-step configuration process follows.

Step 1: Check Clock and Prescaler-Match Data Rate

First check the possible configurations given the desired data rate and the CAN-controller clock. The TQ interval must be calculated based on the clock and various BRP values, and only the combinations where the TQ interval divides into the bit time by an integer number are possible. Depending on the system design stage, it may be possible to consider other CANcontroller-clock rates as well. Table 1 shows example calculations given a 1-Mbps maximum data rate using a Microchip® MCP2515 standalone CAN controller and an ADSP-BF548 Blackfin processor with built-in CAN controller. The MCP2515 f_{OSC} depends on the external-hardware oscillator used, while the ADSP-BF548 f_{SCLK} is determined by the hardware CLKIN and internal PLL settings (CLKIN multiplier for VCO, VCO divide-down for SCLK). Only certain combinations of CANcontroller clock and BRP (an integer number of TQ) will allow a 1-Mbps data rate, as shown in bold. This limits the settings for the bit timing, as only certain options are available once a bus data rate has been chosen.

Table 1. Number of TQ for 1-Mbps Operation Given f and BRP

MCP2515: #TQ for 1-Mbps Operation

$f_{\rm osc}$	BRP = 1	BRP = 2	BRP = 3	BRP = 4			
40	20	10	6.667	5			
38	19	9.5	6.333	4.75			
30	15	7.5	5	3.75			
20	10	5	3.333	2.5			
10	5	2.5	1.667	1.25			

ADSP-BF548: #TQ for 1-Mbps Operation

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~						
$f_{\rm SCLK}$	BRP = 5	BRP = 6	BRP = 7	BRP = 8		
133	26.6	22.167	19	16.625		
100	20	16.667	14.286	12.5		
50	10	8.3333	7.143	6.25		
40	8	6.667	5.714	5		

Step 2: Determine Bit Segment Configuration

The next step is to determine the number of TQ required for each bit segment. The most difficult case is supporting the maximum propagation delay at a 1-Mbps data rate, with a 40-m cable and isolated nodes, for example. Ideally, the bit time segments should be configured so that the sample point is as late as possible in the bit. For each integer total number of TQ in Table 1, one TQ must be allowed for the SYNC segment, and the PS2 (or TSEG2) segment must be large to accommodate the CAN-controller-information processing time (2 TQ for MCP2515, <1 TQ for ADSP-BF548 as long as BRP > 4). Also, for the MCP2515, PROP and PS1 can be 8 TQ maximum each; for the ADSP-BF548, TSEG1 (PROP + PS1) can be 16 TQ maximum.

Figure 4 and Figure 5 show the possible total TQ configurations for MCP2515 and ADSP-BF548 respectively, with the latest sample point possible for the valid combinations of clock and BRP for 1-Mbps operation. The optimum total TQ for the MCP2515 is 19, requiring a hardware oscillator of 38 MHz and BRP of 1. For the ADSP-BF548, all the configurations except for a total of 5 TQ are at least 85% sample point, but the optimum setting is 10 TQ, requiring $f_{SCLK} = 50$ MHz with BRP = 5.





ADSP-BF548: TQ FOR EACH SEGMENT VS. PERCENTAGE OF BIT TIME 20 TQ 1 TC 16 TQ зтQ **FOTAL NUMBER OF TQ** 19 TQ 1 TQ 16 TQ 2 TQ 10 TQ 1 TQ 8 TQ 1 TQ 6 TO 8 TQ 1 TQ 1 TQ 5 TQ 1 TQ 3 TQ 1 TQ 0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100% PERCENTAGE OF BIT TIME SYNC SYNC SYNC SYNC

Figure 5. Possible total TQ configurations for ADSP-BF548, 1-Mbps maximum propagation delay.

Step 3: Match Transceiver/Isolation Delay and Bus Length to Configuration

Having achieved the optimum sample point for the CAN controller, the last step is to compare the allowable propagation delay with the CAN transceiver/isolation and bus length used. Assuming the ADSP-BF548 optimum configuration of 10 TQ (f_{SCLK} = 50 MHz, BRP = 5), the maximum propagation delay possible is 900 ns. For the ADM3053 isolated CAN transceiver with integrated isolated power, the data-sheet maximum loop delay (TxD off to receiver inactive) is 250 ns. This must be doubled (500 ns) to include both transmit and receive delays at two nodes at the furthest ends of the bus.

Assuming a 5-ns/m cable propagation delay, a bus length of 40 meters (maximum for 1 Mbps per ISO 11898) could be accommodated with the ADSP-BF548 having a total of 10 TQ in bit time with only 1 TQ for the TSEG2 bit segment. In practice, slightly earlier sample points may suffice, as even an extreme transceiver propagation delay on one node would likely result in a simple retransmission (handled automatically by the CAN controller at the data-link layer), but the small delay between the CAN-controller I/O and the CAN transceiver makes it advisable to configure the sample point to the latest point possible.

Conclusion

Isolation adds robustness, but it will also add a propagation delay, in both transmit and receive directions. This delay must be doubled to account for two nodes in arbitration. If the allowed propagation delay in the system is fixed, then the cable length or data rate may be decreased if isolation is added. An alternative is to reconfigure CAN controllers to allow the maximum possible propagation delay to ensure that the desired data rate and bus length are possible, even with isolated nodes.

References

Controller Area Network (CAN) transceivers

Digital Isolators

Isolated Controller Area Network (CAN)-Video

Marais, Hein. How to Calculate the Controller Parameters for an Isolated CAN (Controller Area Network) Network to Run at 1 Mbps

O'Brien, Maurice. "Designing Robust, Isolated I²C/PMBus Data Interfaces for Industrial, Telecommunications, and Medical Applications". *Analog Dialogue*, Volume 48, Number 3, 2014.

Watterson, Conal. AN-1123 Applications Note. Controller Area Network (CAN) Implementation Guide. Analog Devices, Inc., 2012.



Dr. Conal Watterson [conal.watterson@analog.com] is an applications engineer for transceivers in the Interface and Isolation Technology Group (ITG) at Analog Devices in Limerick, Ireland. He received a B. Eng. degree in Computer Engineering from the University of Limerick (UL) in 2003 and an M. Eng. degree in 2005 following research into fieldbus diagnostics at the Automation Research Centre (ARC) in UL. He received his Ph.D. from UL in 2010 following research on embedded software monitoring and reliability at the Circuits and Systems Research Centre (CSRC) and the Centre for Telecommunications Value-Chain Research (CTVR).



Analog Dialogue 48-09, September 2014

Dr. Conal Watterson