Optimizing CAN bit configuration on robustness

This article describes an approach on how to get your CAN system robust according to the knowledge provided.

In most serial communications, for example USB and Ethernet, bit-configuration is fixed or performed automatically. This simplifies the use of the communication standard, but also puts constraints on the cable-length, cable-quality, oscillator tolerance and the signal transceiver in use. CAN was designed to be robust, even for a cost-optimized physical layer;

1. Low-cost oscillators -> low tolerance oscillators.
2. Low-cost twisted pair or even a single wire -> large impedance variations.
3. Any cable length -> signal loss, causing lower signal to noise ratio and delays.
4. Low-cost cable driver, like a transistor -> long delays and variations in the driver delay.
5. Real-time performances known and guaranteed -> delays will limit the bit-rate.
6. Functional in a tough electrical environment -> large variations in amplitude and edge location.
7. Functional across a wide temperature range -> large variations in impedance and delays.
8. Functional in a rough chemical environment -> large impedance variations.

All the constraints listed above will cause variations and demands on the edges and the amplitude in each bit. However, by configuring the CAN bit, it is possible to protect the sample point to ensure that the communication is as robust as possible for a specific environment.

If the CAN bit-rate and the CAN bit configuration are predefined in some way, it results in constraints in all the parameters listed above, resulting in a less robust communication. Some of the parameters such as cable length, oscillator tolerance, and delays can be found in the oscillator and CAN driver data sheets and specifications. Other parameters such as EMI (electromagnetic interference) and changes due to temperature and ageing must be predicted. It is almost impossible to test all possible variations that could exist in the environment over the lifetime of the system. The best solution is to find a sample point most protected from expected variation in the future. Normal sampling of the bit is done at the center of the bit to ensure that the sample point is as far away from the edges at both ends of the bit to be sampled (Figure 1)

The sampling is synchronized to the first edge received (A1) and the sampling of the bit is relative to this edge at B1. If the transmitter of the frame has an oscillator that is faster than the receiver, as shown in Figure 1, the edge shifts a little to the left for every bit (time) that passes, compared to the location of the edge at A1, C1, D1, and G1.

After several bits, the edge reaches the sample point and the sampling takes place in the wrong bit. To solve this problem there is a resynchronization mechanism within the communication. When the edge is detected to be outside of the sync-segment (D1), the receiver adjusts the bit-length by removing as many time quanta as necessary to put the edge in the sync-segment. In this case, one TQ (time quantum) is removed in the propagation segment at (E1) to get a better sample location at (F1) and the edge is now located in the sync-segment G1. From this it is obvious that the phase-segments around the location of the sample point must be large enough to ensure that oscillator difference phase shift does not reach the sample point. It should also be obvious that SJW, defining the maximum number of TQ (time quanta) that can be removed, is large enough to allow necessary adjustment. On the other hand, SJW should be as small as possible to ensure that other random phase shifts do not result in over-compensation.

If the transmitter of the frame has an oscillator that is slower than the receiver, the edge shifts a little to the right for every bit that passes. If the signal is exposed to noise, for example under an EMC-test, the noise also introduces a phase shift in time that is closely related to the

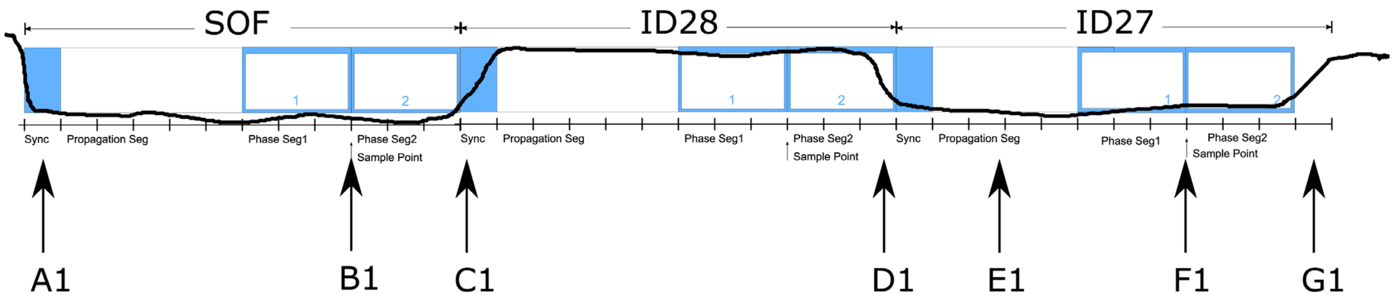

Figure 1: Alteration of phase shift due to oscillator tolerance by adjusting the number of time quanta (Source: Kvaser)

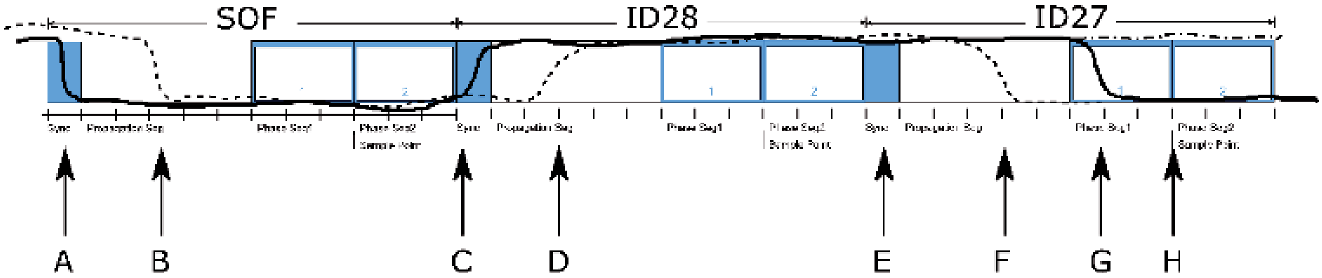

Figure 2: A 'worst case' phase shift due to arbitration (Source: Kvaser)

radiated energy. A bit short in time (high bit-rate) is more sensitive than a long bit because this edge variation will take a greater part of the bit short in time.

Protecting sample point from phase shift

In CAN, the signal is either electrically forced to a dominant (0) or passively restored to recessive (1) by the ending resistor. This typically provides a well-defined edge when the bit level changes from recessive to dominant. The edge from dominant to recessive is restored by discharging the stored charges connected to the bus-line in the ending resistors. If the CAN network is long, with many installed units, this discharge could be considerable, resulting in a delay in the switch from a dominant to recessive state. The resulting phase shift (delay) of the dominant to recessive edge must be considered when protecting the sample point. The problem with oscillator tolerance is the same for any bit-rate but the phase shift due to the passive switch from dominant to recessive has a fixed value in time and the effect of this part increases when the bit-length becomes smaller at a higher bit-rate.

CAN is a multi-master communication which allows any unit to start sending as soon as the communication media is idle. If more than one transmitter starts a CAN frame, the collision is solved by arbitration. Arbitration demands that all units sample each CAN bit, one by one, and evaluate each bit, one at a time. At the start of the CAN frame, in the arbitration section, the sender with a lower priority level is excluded from transmitting the complete CAN frame. Figure 2 shows why this arbitration could result in a large phase shift. To simplify the description in this example, only two units are involved in the arbitration. The two units are located at the ends of the CAN cable, with the longest possible signal delay. In CAN, any unit can start sending when there is no communication i.e. the CAN network is idle. The first unit starts to send the first bit (SOF-bit) at A and this edge propagates down CAN and reaches the other end at B.

▷

Up to this point, the CAN network is considered idle by the other unit, so this unit starts a CAN frame at B. This first bit in a CAN frame is always dominant and, in this case, the first arbitration bit is recessive. The first unit switches recessive at C and the second unit does this with the delay at D. The ID27-bit in the first unit is recessive and keeps the recessive level at E. The second unit has a dominant ID27-bit and makes the CAN network dominant at F. This dominant edge propagates back to the first unit and makes the CAN network dominant at the first unit at G. The propagation segment should be large enough to keep the phase shift due to propagation delay within the propagation segment. In the example above, the second unit has an oscillator that is slower than the first unit and this phase shift adds to the phase shift caused be the propagation delay, causing the edge to reach some distance into phase segment 1. This is not a big problem because the resynchronization mechanism described in Figure 1 adds in TQ in the propagation segment and moves the sample point at H further back in time.

Achieving the correct bit-setting

There are some basic rules to follow to simplify CAN bit configuration. Those rules can by divided into five different areas.

1. The oscillators and their parameters used in different units connected to the CAN network.
   a. High tolerance oscillators simplify the configuration and increase system robustness.
   b. An identical oscillator frequency in all units simplifies the configuration and makes the system more robust.
2. The CAN drivers, which have different delays and capability to shape the signal.
   a. CAN drivers with low delay permit more bit-time to be used for cable delay or to protect the sample point against phase errors.
   b. CAN drivers with low variations in the delay reduce the risks created by temperature and ageing.
   c. CAN drivers with good signal forming reduce phase errors.
   d. CAN drivers with good signal shaping reduce EM-noise from the CAN.
3. EMC filters, which introduce a capacitive and inductive load that reshape the signal.
   a. If possible, do not use EMC filters between the CAN driver and the CAN network.
   b. If EMC-filters are used, select devices with low capacitance and inductance, to minimize the amount of energy stored.
4. The length of the drop-lines, from the main bus-line to the units.
   a. Each cable segment holds capacitance and inductance, which affects the signal. To minimize the effect, make the drop-lines as short as possible.
5. The length of the CAN network.
   a. Long cables cause long signal delays, which limits the arbitration bit-rate.
   b. Long cables cause a voltage drop, due to resistive loss.
   c. Long cables can pick-up more EM-noise.
   d. Long cables can transfer more EM-radiation.

How to select oscillators

1. If using oscillators with better than 100 parts per million stability, then all accumulating errors can be ignored.
2. Use the same oscillator frequency in all modules to ensure that the same settings are applied to all modules. This ensures that no phase errors are introduced due to the bit-configuration.
3. If using different oscillator frequency in different units, ensure that the relationship fulfills the following equation: osc_x/Mx == osc_y/My, where Mx and My are integers. (for example if osc_x = 12 Mhz and osc_y is 25 MHz, the smallest possible integer is Mx=12 and My=25
   a. Divide the higher frequency to get the same time quanta length as in the unit with the lowest oscillator frequency.
   b. This makes it possible to have the same configuration of the CAN bit in all units, which removes any possible phase error.
4. If any combination of units do not comply with the equation, osc_x/Mx == osc_y/My, where Mx and My are integers, it causes an accumulating phase error.
   a. If this is the case, there are very few configurations of the CAN bit that work and it could even be impossible for the higher bit-rates.
   b. The above problem is reduced if small time quanta are used, allowing small bit adjustments to be made.
   c. The phase error introduced by this condition is in addition to the oscillator tolerance error, making it hard to use low precision oscillators
   d. This demands a carefully defined CAN bit to secure good protection of the sample point.

How to select CAN drivers

1. To minimize the effects from the CAN driver, select a CAN driver made for handling high bit-rates.
   a. Note that high bit-rate typically results in higher EM emission, because the slew rate must be higher to form the small bits.
2. If EM-emission is a problem, use a CAN driver that support adjustable slew-rate and use as little slew-rate as possible. Also, use as low a bit-rate as possible.

How to select drop lines

A drop line connected to the main bus line is like an imperfection in the cable. The introduction of a drop line is like adding a circuit with capacitive and inductive loads to the ▷

CAN network, which affects the shape of the signal. The imperfection increases in proportion to the length of the drop line.

1. Use as short a drop-line as possible to avoid this problem.
2. High bit-rate signals with a high slew rate are more impacted than a signal with a lower slew rate, which can be used for the lower bit rate.
3. High frequency signals are more sensitive toimperfections.
4. The frequency where the drop-line causes a problem is related to the length of the drop-line.
   a. freq > 30/(drop_line[meter]), is largely affected by the drop-line.
   b. The digital signal has energy at the following frequencies; bit_rate + 3*bit_rate + 5*bit_rate ...
      I. For example 1 Mbit/s has energies at 1 MHz, 3 MHz, and 5 MHz.
      II. At 1 Mbit/s, drop-lines over 3 m will be detectable.
      III. At 1 Mbit/s, a total drop-length above 30 m will have a major impact.
      IV. The problem is less acute if drop-lines are equally distributed along the main CAN network.
   c. The distortion accumulates and increases with the number of drop-lines.

How to design the CAN cable

1. Use the same impedance in all cable segments.
   a. Any change in the impedance along the cable causes signal reflections at the change of the impedance.
   b. The twist is not directly necessary to secure the impedance. The impedance is defined by the cross-section of the wires in combination with the dielectric parameters of the wire isolation. The twist is a method to secure a continuous mechanical cross section.
   c. Variations in impedance shorter than the wavelength have a minor impact.
      I. For example 1 Mbit/s has energies at 1 MHz, 3 MHz, and 5 MHz.

Five parameters that affect CAN bit configuration

1. SJW, Synch Jump Width: This value defines how much of the CAN bit is reserved for handling accumulating phase-errors. If good oscillators are used in combination with common oscillator frequency in all units, this value is zero and can be ignored. The minimum value for the SJW parameter is 1 time quanta.
2. PHASE_SEGMENT2: This is the space between the sample point and the end of the CAN bit. This part must be large enough to fit the SJW plus the phase shift caused by non-accumulating phase errors (noise). If SJW is small it is possible to handle large variations from non-accumulating phase errors (noise).
3. PHASE_SEGMENT1: The phase shift, accumulating, and non-accumulating errors can come from both sides of the sample point. To cover this, phase segment 1 must be of the same size as phase segment 2.
4. PROPAGATION _SEGMENT: This part must be large enough to fit twice the longest delay in time between any unit connected to the CAN system.
5. SYNCH SEGMENT: This part is the precision in the edge detection. The CAN status is sampled every time quanta and the bit-edge can be anywhere between two samples of the CAN network status. If the time quanta is short in time, it is possible to have a large number of TQ in a bit, making the synch_ segment a relatively small portion of the bit.

Using data from installed nodes

By collecting data about the installed node, it is possible to calculate the values on those parameters. Those parameters are given by the CAN layout and the parameters in the ECUs. The minimum length of the CAN bit is defined by the sum of those parameters.

BIT_LENGT_MIN = SYNCH_SEGMENT + PROPAGATION_ SEGMENT + PHASE_SEGMENT1 + PHASE_SEGMENT2.

In most cases, the used bit-length is longer that the BIT_LENGTH_MIN and in that case it is possible to divide this slack into the different parts in the CAN bit. To ensure the CAN system’s robustness against variations in the cable length and delays in the circuits, this slack should be assigned to the PROPAGATION_SEGMENT. To ensure the CAN system’s robustness against phase noise and variations in clock tolerance, this slack should be divided between PHASE_SEGMENT1 and PHASE_SEGMENT2. If it is not known which part is the weakest, the best solution is probably to divide the slack between PROPAGATION_SEGMENT, PHASE_SEGMENT1, and PHASE_SEGMENT2.

It should be noted that an unnecessary large SJW makes the CAN system more sensitive to noise, because the resynchronization mechanism can´t know if the phase shift is due to accumulated phase shift, that should be compensated, or non-accumulated phase errors that could result in an unnecessarily large phase compensation.

With this approach you get the most robust CAN system according to the knowledge provided. The other approach is to predefine some basic setting for the bit-timing. If this setting is used, there is one system with maximum robustness. By deviating from this optimal system, a system becomes weaker compared to an optimal CAN system. CAN is very robust so it is possible to have larger deviations before the CAN system encounters any real problems. ◀

Author

Kent Lennartsson
Kvaser
info@kvaser.com
www.kvaser.com