# A CAN oscillator

Lars-Berno Fredriksson
050405 Kvaser

The tolerance of the oscillators in a CAN system is crucial. The relation between oscillators, propagation delays and the maximum bit rate is given by the ISO 11898 standard. The frequency of each ECU, $f_{osc}$, must not deviate from the nominal frequency $f_{nom}$ by more than *df* fulfilling all relations below:

1) $(1-df)*f_{nom} \leq f_{osc} \leq (1+df)*f_{nom}$

2) $df \leq \{(\text{Phase\_Seg1, Phase\_Seg2})_{min} / [2*(13*T_b - \text{Phase\_Seg2})]\}$

3) $df \leq (\text{SJW} / 20*T_b)$

where Tb is the nominal bit time.

A CAN bit can be divided into four parts: Synchronization segment, propagation segment, phase segment 1 and phase segment 2. The relation between the oscillator accuracy and the phase segments is given by 2) above. The lower accuracy, the longer phase segments and thus, for a given bit rate, the shorter propagation segment, i.e., shorter bus length. In other words, the longer distance between nodes, the better oscillators are required. Better oscillators means higher cost and often crystals. In many designs, ceramic oscillators are preferred over crystal ones, as the former are not only cheaper but lighter and more reliable. Equation 2) above gives (with $T_b$ = 25 and Phase_Seg2 = 1) the smallest value of *df* = .154%.

## The best of two worlds

An important fact about *df* is that the requirement is limited to the *relative* accuracy of the oscillators in the system. If we introduce an adjustable ceramic oscillator at each node and tune it to the bit rate of messages on the bus, we can lower the phase segments considerably. A commercially available adjustable oscillator is the DS1085L from Dallas Maxim. Its base frequency can be selected between 33 MHz and 66 MHz in 13 overlapping ranges. It comes in three versions with different ranges, 5.12, 12.80 or 25.60 MHz and it can be tuned in 1024 steps within each range. The Table I below shows two examples on how the 5.12 MHz version would perform and Table II how the 25.6 MHz would perform. The nominal accuracy of the oscillator is .75%. The frequency may drift .75% over the temperature range and another .75 over the voltage range. The integral nonlinearity is within .3% and then, in total, the accuracy is within 1.8 %.

Table I shows the oscillator programmable to a range from 61.4 to 66.6 MHz in steps of 5 kHz. However, the specified tolerances are valid only up to 66 MHz. The frequency is divided by two and the CAN Controller programmed to 250 kbit/s and 16 BTQ at 32 MHz. As discussed above, at a nominal frequency of 64 MHz, the actual frequency could be anywhere within +/- 1.8% if no adjustments are made for actual temperature and voltage. During normal conditions, a re-synchronization edge will appear at least after 10 bit times. One bit quantum would then cover a phase error of +/-.3% after ten bit times and a maximum phase segment of 8 would cover 4.76% . The total range for the oscillator covers 3.1% - -4.2 % for the selected bit timing setting in this example.

Table II shows the oscillator programmable between 41.6 and 66 MHz in steps of 25 kHz. The frequency is divided by two and the CAN Controller programmed to 250 kbit/s and 18 BTQ at 27 MHz. With this setting, the oscillator covers a bit timing range of more than +/- 22%. Still, a 10%

of BTQ is covered by 12 steps and the resolution is good enough to compensate for a one bit time quantum phase adjustment in the hundredth bit of a message. With the same bit timing setting, the bit rate could be changed from 200 to 300 bit/s by adjusting the frequency.

The exercises above with a commercially available adjustable silicon oscillator at less than 3 USD in 1k quantities makes it worthwhile to think about a chip that could adjust itself to the bit rate on a CAN bus. One part would be similar to the discussed oscillator and another part similar to a CAN controller. The example of 250 kbit/s is ad hoc and the 25 MHz steps were chosen as it is offered by the existing component. Yet by choosing 62.5, 125, 250 and 500 as nominal bit rates, almost the whole range from 50 kbit/s to 1 Mbit/s could be covered. To fill the gaps, 87.5, 175, 300 and 600 could be selected with a bit time setting of 20 BTQ and 28 MHz nominal frequency for 350 kbit/s. Table III shows that this would give an overlap and with two CAN Controllers, it would be possible to scan a CAN bus for any bit rate between 50 kbit/s and 1 Mbit/s in only 9 steps requiring only 9 messages on the bus. Most probably it is possible to optimize the oscillator and the nominal bit rate to make it even faster. Once the nominal bit rate is found, the oscillator can be tuned to the existing bit rate good enough to have only two BTQ for the time seg 2. This would maximize the allowable cable length in a system for a given bit rate or maximize the bandwidth for a given network.

## Construction of a CAN oscillator

Figure 1 shows the construction of a CAN oscillator in principle. The variable silicon oscillator 1 could have similar properties as the DS1085L. A careful analysis of the complete concept may show that there is a different set of specifications that would make a more cost efficient design. The oscillator is connected to a simple CAN Controller 2 that is permanently configured into listen only mode, i.e., no ACK bit is produced and it is only connected to the RX line. It has a bit stream processor 3, a counter 4 with capture registers, one or more filters 5 and one or more bit timing registers 6. The filter and bit timing register can be programmed through a serial interface 7 from a micro controller 8. The CAN oscillator is connected to the CANbus by the preferably integrated transceiver TC which is connected to an ordinary CAN Controller CC by the RX and TX lines. The CAN oscillator is connected to the RX line and has a clock output 9 to the CC that also might clock the micro controller. The oscillator is connected to the CAN controller by two output lines 10 and 11 activated when the CAN Controller makes a synch jump. When the falling edge comes too early, output 10 is activated to increase the oscillator frequency and if the edge comes too late, output 11 is engaged to lower the frequency.

## Additional features

A reference frequency input 12 would be good to have for tuning the oscillator. GPS is an obvious reference and some GPS modules provide a 10 kHz output synchronous with the Epoch signal which can be identified in the square wave stream. A small CPU 13 with some RAM and flash, connected to the serial interface would make the oscillator more flexible. It can be optimized to work with the CAN Controller part and some additional features may be offered, e.g., pulse generation and reception for measuring and diagnostic purposes. Then a TX connection 14 is added. For measuring purposes it is essential that SOF and ACK signals can trigger capture registers connected to the counter 4.

## Higher Layer Protocol requirements

The CAN oscillator can be used in very many ways. The simplest way is to have no rules at all. Then an arbitrary node will be the first to transmit and the others will synchronize on the first message. In CAN Kingdom, the first messages are from the King and we have then automatically a kind of frequency master. One or more specific messages can be selected as reference messages. Then the CAN oscillator should synchronize only on messages that fits the filter(s) 5. If the higher layer protocol allows a schedule that guarantees no collisions with the reference messages, the whole message can be used for synchronization. If not, the id field may be excluded to gain

maximum accuracy.

Further investigations

Here only the basic ideas are described. Further investigations should be done for an optimal design. Some areas are suggested below:

1. What would it take to make a CAN oscillator that could auto-baud at any frequency within the CAN spec.?

2. How does the relative frequency accuracy of the nodes depend on nominal tolerances (absolute, temp, power and aging) and busload?

3. What are the requirements on a CAN oscillator that should support a variable bit rate? An example of application could be an engine control system where every node in the system should be synchronized to the position of the engine shaft.

4. What is the relation between the production cost of a chip and the different tolerances and process technology?

Table I
Oscillator frequency range 61.4 - 66.0 (66.6) MHz

| Frequency MHz | Bit rate kbit/s | # BTQ | ns/BTQ | µs/bit | Time deviation ns | Note |
|---|---|---|---|---|---|---|
| 30.7 | 239.8438 | | 260.59 | 4.1693811 | 169.38 | -4.2% |
| | | | | | | |
| 31.9 | 249.2188 | | 250.78 | 4.0125392 | 12.54 | -.3%, -40 steps |
| | | | | | | |
| 31.99 | 249.9219 | | | 4.0012504 | 1.25 | -.03% |
| 31.9925 | 249.9414 | | | 4.0009377 | 0.94 | |
| 31.995 | 249.9609 | | | 4.0006251 | 0.63 | |
| 31.9975 | 249.9805 | | | 4.0003125 | 0.31 | |
| 32 | 250.0000 | 16 | 250.00 | 4 | 0.00 | 78 ppm/step |
| 32.0025 | 250.0195 | | | 3.9996875 | -0.31 | *(min. df* = .154% ) |
| 32.005 | 250.0391 | | | 3.9993751 | -0.62 | |
| 32.0075 | 250.0586 | | | 3.9990627 | -0.94 | |
| 32.01 | 250.0781 | | | 3.9987504 | -1.25 | .03% |
| | | | | | | |
| 32.1 | 250.7813 | | 249.22 | 3.9875389 | -12.46 | .3% ,+40 steps |
| | | | | | | |
| 33 | 257.8125 | | 242.42 | 3.8787879 | -121.21 | 3.1% |
| | | | | | | |
| ( 33.3 ) | 260.1563 | | 240.24 | 3.8438438 | -156.16 | ( 4.1% ) |

| Table II<br>Oscillator frequency range 41.6 - 66.0 (67.2) MHz | | | | | | |
|---|---|---|---|---|---|---|
| Frequency MHz | Bit rate kbit/s | # BTQ | ns/BTQ | µs/bit | Time deviation ns | Note |
| 20.8 | 192.5926 | | 288.46 | 5.1923077 | 1192.31 | -22.9% |
| | | | | | | |
| 25.8 | 238.8889 | | 232.56 | 4.1860465 | 186.05 | -4.4% , -94 steps |
| | | | | | | |
| 26.925 | 249.3056 | | | 4.0111421 | 11.14 | -6 steps |
| | | | | | | |
| 26.975 | 249.7685 | | | 4.0037071 | 3.71 | |
| 26.9875 | 249.8843 | | | 4.0018527 | 1.85 | |
| 27 | 250.0000 | 18 | 222.22 | 4 | 0.00 | 463 ppm/step |
| 27.0125 | 250.1157 | | | 3.998149 | -1.85 | *(min. df* = .154% ) |
| 27.025 | 250.2315 | | | 3.9962997 | -3.70 | |
| | | | | | | |
| 27.075 | 250.6944 | | | 3.9889197 | -11.08 | +6 steps |
| | | | | | | |
| 28.2 | 261.1111 | | 212.77 | 3.8297872 | -170.21 | +4.4%, +94 steps |
| | | | | | | |
| 33 | 305.5556 | | 181.82 | 3.2727273 | -727.27 | 22.2% |

| Table III | |
|---|---|
| 27 MHz 18 BTQ | 28 MHz 20 BTQ |
| 50 | |
| 62.5 | |
| 75 | 70 |
| | 87.5 |
| 100 | 105 |
| 125 | |
| 150 | 140 |
| | 175 |
| 200 | 210 |
| 250 | |
| 300 | 280 |
| | 350 |
| 400 | 420 |
| 500 | |
| 600 | 560 |
| | 700 |
| 800 | 840 |
| 1000 | |

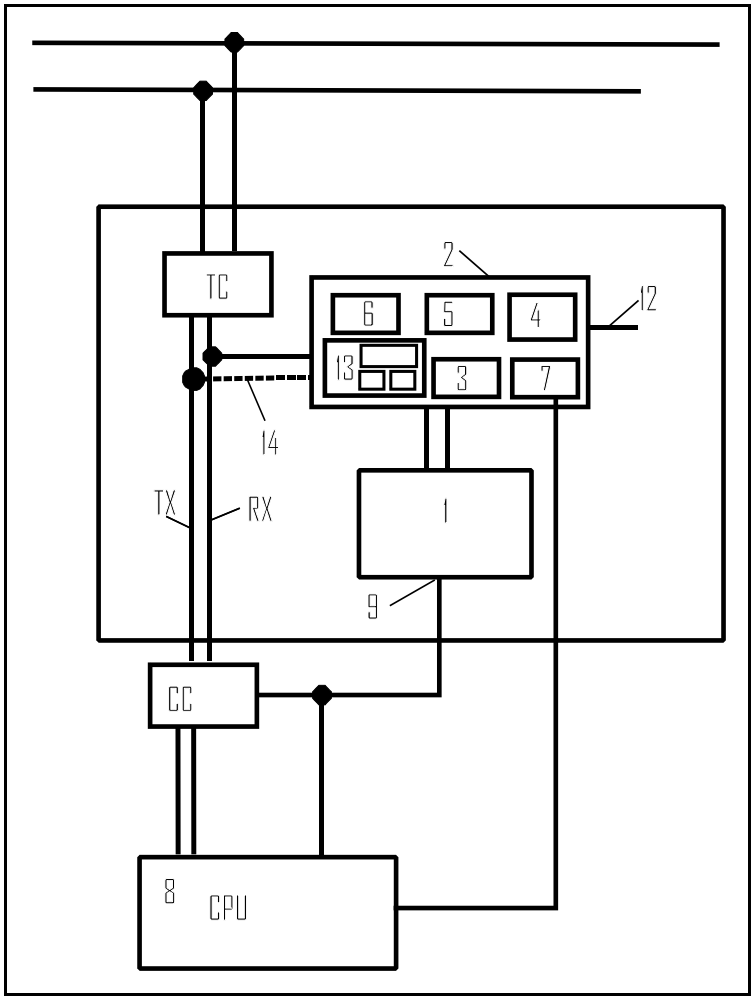


**Figure 1**