FlexCAN Bit Timing Calculation

by: Petr Stancik
TIC

The FlexCAN module supports a variety of means to setup bit timing parameters that are required by the CAN protocol.

This document gives a basic insight into bit timings relationship and provide easy step-by-step guide to calculate bit timing parameters for desired baudrate.

Table Of Contents

1 Introduction ...........................................2
2 Step-by-step guide .................................5
3 Example ................................................6
1 Introduction

The clock dependency can be described based on the following figure.
Starting from a data frame transferred on the bus. A desired baudrate tells how many bits are sent on the network per second. The Nominal Bit Rate (baudrate) is uniform throughout the network and is given by:

\[
\text{baudrate} = \frac{1}{t_{\text{NBT}}}
\]  

(1)

The \( t_{\text{NBT}} \) represents a period of the Nominal Bit Time (NBT), i.e. single bit transferred on the bus. The NBT is separated into four non-overlapping segments, SYNC_SEG, PROP_SEG, PHASE_SEG1 and PHASE_SEG2. Each of these segments is an integer multiple of Time Quantum \( t_Q \).

\[
t_{\text{NBT}} = t_Q + \text{PROP}_\text{SEG} * t_Q + \text{PHASE}_\text{SEG1} * t_Q + \text{PHASE}_\text{SEG2} * t_Q = \text{NBT} * t_Q
\]

(2)

The below table summarizes the possible values each segment can be set to.

<table>
<thead>
<tr>
<th>Segment</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNC_SEG</td>
<td>( t_{\text{SYNC}_\text{SEG}} = 1 \ t_Q )</td>
</tr>
<tr>
<td>PROP_SEG</td>
<td>( t_{\text{PROP}_\text{SEG}} = 1, 2 \ldots 8 \ t_Q )</td>
</tr>
<tr>
<td>PHASE_SEG1</td>
<td>( t_{\text{PHASE}_\text{SEG1}} = 1, 2 \ldots 8 \ t_Q )</td>
</tr>
<tr>
<td>PHASE_SEG2</td>
<td>( t_{\text{PHASE}<em>\text{SEG2}} = \max(\text{IPT}, t</em>{\text{PHASE}_\text{SEG1}}) )</td>
</tr>
</tbody>
</table>

Note: The function \( \max(\,\, , ) \) returns the larger of the two arguments.

The duration of the segment PHASE_SEG1 may be between 1 and 8 Time Quanta if one sample per bit is selected and may be between 2 and 8 Time Quanta if three samples per bit are selected. If three samples per bit are chosen, the most frequently sampled value is taken as the bit value.

The duration of segment PHASE_SEG2 must be equal to PHASE_SEG1, unless PHASESEG1 is less than the Information Processing Time (IPT), in which case PHASE_SEG2 must be equal to the Information Processing Time. The Information Processing Time is equal to 2 Time Quanta.

From the table, it would appear that the minimum number of Time Quanta per bit (NBT) is 5. However, many CAN controllers require a minimum of 8 Time Quanta per bit. The maximum number of Time Quanta per bit is 25.
A Time Quantum is the atomic unit of time handled by the CAN engine and it is given by the Serial Clock (Sclock) period as

\[ t_Q = \frac{1}{\text{Sclock}} = \frac{1}{[\text{CPI\_clock} \div (\text{PRESDIV} + 1)]}. \] (3)

The CPI\_clock is the CAN Protocol Interface clock, can be either the peripheral clock (driven by the FMPLL) or the crystal oscillator clock. The PRESDIV controls a Prescaler that generates the Serial Clock (Sclock).

![Diagram]

The clock source selection is done by the CLK\_SRC bit of the CTRL register. Moreover this register contains other fields used to control bit timing parameters as was mentioned above: PRESDIV, PROPSEG, PSEG1, PSEG2 and RJW. The SMP specifies if one or three samples per bit is used.

![Register Diagram]

These parameters have to be properly calculated/selected to meet the CAN specification.

The following chapter provides a step-by-step guide for determining the optimum bit timing parameters which satisfy the requirements for proper bit sampling. Besides the desired baudrate and selected CPI clock a physical interface parameters and bus length is also needed to determine propagation segment duration.
2 Step-by-step guide

STEP 1: Determine minimum permissible time for the PROP_SEG segment.
Obtain the maximum propagation delay of the physical interface for both the transmitter and the receiver from the manufacturers data sheet. Calculate the propagation delay of the bus by multiplying the maximum length of the bus by the signal propagation delay of the bus cable. Use these values to calculate $t_{PROP\_SEG}$ using equation

$$t_{PROP\_SEG} = 2(t_{BUS} + t_{TX} + t_{RX})$$

(4)

STEP 2: Choose CPI_clock, the Prescaler and NBT.
Determine Prescaler and a Nominal Bit Time (NBT). Basically you can combine equations 1, 2 and 3 into

$$NBT \times \text{Prescaler} = \frac{\text{CPI\_clock}}{\text{baud rate}}.$$  

(5)

As already mentioned NBT may be integer number in range 8 to 25 and Prescaler in range 1 to 256. Find those combinations of NBT * Prescaler which is equal to CPI_clock /baud rate. There may be a lot of possible combinations. If none can be found choose different CPI_clock.

STEP 3: Calculate PROP_SEG duration.
The number of time quanta required for the PROP_SEG segment are calculated based on chosen Prescaler value

$$\text{PROP\_SEG} = \text{ROUND\_UP}\left(\frac{t_{PROP\_SEG}}{t_{Q}}\right) = \text{ROUND\_UP}\left(t_{PROP\_SEG} \times \frac{\text{CPI\_clock}}{\text{Prescaler}}\right)$$

(6)

where the function ROUND_UP( ) returns the argument rounded up to the next integer value.

STEP 4: Determine PHASE_SEG1, PHASE_SEG2
From the number of Time Quanta per bit obtained in Step 2, subtract the PROP_SEG value calculated in Step 3 and subtract 1 $t_Q$ for SYNC_SEG.
If the remaining number is an odd number greater than 3 then add one to the PROP_SEG value and recalculate.
If the remaining number is equal to 3 then PHASE_SEG1 = 1 and PHASE_SEG2 = 2 and only one sample per bit may be chosen. Otherwise divide the remaining number by two and assign the result to PHASE_SEG1 and PHASE_SEG2.
**STEP 5: Determine RJW**

RJW is chosen as the smaller of 4 and PHASE_SEG1

**STEP 6: Calculate required oscillator tolerance**

There are two clock tolerance requirements which must be satisfied.

\[
(2 \cdot \Delta f) \cdot 10 \cdot t_{NBT} < t_{RJW} \tag{8}
\]

\[
(2 \cdot \Delta f) \cdot (13 \cdot t_{NBT} - t_{PHASE\_SEG2}) < \text{MIN}(t_{PHASE\_SEG1}, t_{PHASE\_SEG2}) \tag{9}
\]

where the function MIN( , ) returns the smaller of the two arguments.

As the required oscillator tolerance select the smaller of the calculated values.

### 3 Example

Calculate the bit segments for the following system constraints:

- Bit rate = 500k bit per second
- Bus length = 10m
- Bus propagation delay = 5 x 10^-9 s/m
- Physical Interface (PCA82C250) transmitter plus receiver propagation delay = 150ns at 85°C
- CPI_clock = 40 MHz

**STEP 1:**

Physical delay of bus = 10m x 5 ns/m = 50ns

\[t_{PROP\_SEG} = 2(50 + 150) = 400\text{ ns} \]

**STEP 2:**

\[NBT \cdot \text{Prescaler} = \text{CPI\_clock} / \text{baud rate} = 40 \text{ MHz} / 500\text{kbps} = 80\]

There can be 4 possibilities:

- NBT = 20, Prescaler = 4
- NBT = 16, Prescaler = 5
- NBT = 10, Prescaler = 8
- NBT = 8, Prescaler = 10

Lets choose the first one

**STEP 3:**

\[PROP\_SEG = \text{ROUND\_UP}(400\text{ns} \times 40\text{MHz} / 4) = 4\]
STEP 4:
NBT – 1 – PROP_SEG = 20 – 1 – 4 = 15
As the result is odd increase PROP_SEG and recalculate
20 – 1 – 5 = 14
Divide the number by two and assign the result to PHASE_SEG1 and PHASE_SEG2.
PHASE_SEG1 = 7 and PHASE_SEG2 = 7

STEP 5:
RJW is chosen as the smaller of 4 and PHASE_SEG1, so RJW = 4

STEP 6:
( 2 * Δf) * 10 * t_{NBT} < t_{RJW}
( 2 * Δf) * 10 * 20 < 4 => Δf < 0.01 = 1%

( 2 * Δf) * (13 * t_{NBT} – t_{PHASE_SEG2}) < MIN(t_{PHASE_SEG1}, t_{PHASE_SEG2})
( 2 * Δf) * (13 * 20 – 7) < MIN(7, 7) => Δf < 0.0138 = 1.38%

required oscillator tolerance is 1%

In summary:
Prescaler = 4
Nominal Bit Time = 20
PROP_SEG = 5
PHASE_SEG1 = 7
PHASE_SEG2 = 7
RJW = 4
Oscillator tolerance = 1%

Similarly repeat steps 3-6 for next NBT, Prescaler values given in step 2.
Calculated bit timing parameters are given in following table.

<table>
<thead>
<tr>
<th>Prescaler</th>
<th>Sclk [MHz]</th>
<th>t_q [ns]</th>
<th>NBT [#t_q]</th>
<th>PROPSEG</th>
<th>PSEG1 [#t_q]</th>
<th>PSEG2 [#t_q]</th>
<th>RJW [# t_q]</th>
<th>Δf [%]</th>
<th>sample point [%]</th>
<th>CANx_CR [hex]</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>10</td>
<td>100</td>
<td>20</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>4</td>
<td></td>
<td>1</td>
<td>65.00</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>125</td>
<td>16</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>1.231527</td>
<td>68.75</td>
<td>0x04E40004</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>200</td>
<td>10</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1.181102</td>
<td>70.00</td>
<td>0x07920002</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>250</td>
<td>8</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0.980392</td>
<td>75.00</td>
<td>0x09490002</td>
</tr>
</tbody>
</table>