

Computation of CAN Bit Timing Parameters Simplified

Meenanath Taralkar,
OTIS ISRC PVT LTD, Pune, India

CAN bit timing and synchronization play an important role in ensuring performance of CAN network. A misplaced sampling point can lead to one of the transmitters going error passive. Computation of right timing parameters require detailed knowledge of CAN bit synchronization and also may lead to complexities owing to number of input parameters & a variety of possible solutions. The engineer should choose the optimum Sampling Point and Time Quanta so as to ensure robust CAN implementation.

Purpose of this paper is to present a simplified method, such as a ready reckoner, to compute Time Quanta using a two-step computation method. During the first step, Pre-scalar values for each Time Quanta are calculated using CAN Frequency and Baud-rate. This provides a set of permissible Time Quanta's. During the next step, Sampling Point and Oscillator Tolerance are calculated for each permissible Time Quanta. This yields output variables such as Time Segment 1 & 2 and Synchronization Jump Width which are subsequently used to configure the CAN controller. In addition, this paper attempts to analyze trend of Oscillator Tolerance and Sampling Point for various values of Time Quanta.

1. Introduction

CAN bit time includes different segments such as - Sync_Seg, Prop_Seg, Phase_Seg1 and Phase_Seg2.

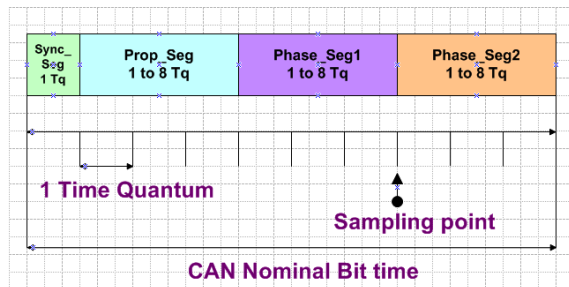


Figure 1: CAN Bit time segments

- Sync_Seg - Used to synchronize all the nodes on the bus.
- Prop_Seg - Used to compensate the physical delay on the network.
- Phase_Seg1 and Phase_Seg2 – Used to compensate for edge phase errors.
- Sampling point - Sampling point is the point of time at which the bus

level is read and interpreted as the value at that respective time.

- Time Quantum (Tq) - The Time Quantum is a fixed unit of time derived from the oscillator period. The Time Quantum is equal to the period of the CAN system clock, which is derived from the system clock or oscillator by dividing the system clock or oscillator by the programmable pre-scalar, called Baud Rate Pre-scalar (BRP).^[1]
- Nominal Bit Time (NBT): This is the sum of all the CAN bit time segments.

Sync_Seg duration is fixed to 1 Tq. The duration of other segments is programmable, and ranges from 1 to 8 Tq. Minimum length of Phase_Seg2 is 2 Tq because it must not be shorter than information processing time (IPT). The duration of IPT is ≤ 2 Tq. Table 1 lists the minimum and maximum duration of CAN bit time segments, and indicates that the

minimum number of Time Quanta per bit 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.^[5]

Segment	Minimum Duration	Maximum Duration
Sync_Seg	1 Tq	1 Tq
Prop_Seg	1 Tq	8 Tq
Phase_Seg1	1 Tq	8 Tq
Phase_Seg2	2 Tq	8 Tq
Total	5Tq	25Tq

Table 1: Bit time segment duration

Along with the above mentioned bit time segments, there is another parameter which needs to be configured is Re-synchronization Jump Width (RJW) also called as Synchronization Jump Width (SJW). This parameter is used as a maximum limit of bit time adjustment in the case of re-synchronization when there is an edge phase error detected by the receiver CAN nodes. RJW value ranges from 1 to MIN(4, Phase_Seg1).

To achieve a specified bit rate, values of all the bit time segments and RJW need to be determined properly. A given bit rate may be met by different bit time configurations.

E.g. With 48MHz CAN clock frequency, 125K bit rate can be achieved with NBT = 8, 12, 16 and 24 Tq, with 0% deviation in the bit rate. From these four choices an appropriate value of NBT needs to be determined along with the bit time segments and the RJW value. While determining these values, the following characteristics have to be considered.

- Propagation delay
- Oscillator tolerance

This paper describes these characteristics and a procedure to generate a ready reckoner which lists out all combinations of NBT and all permissible settings of bit time segments and RJW for each value of NBT. This paper also lists the steps to be followed to choose appropriate value of NBT and bit

timing parameters to get the desired bit rate as per the propagation delay and oscillator tolerance requirements. The goal is to equip the engineer with a simplified approach and tool for quick & easy determination of CAN bit time parameters used to program CAN controllers.

2. Propagation delay

Propagation delay includes physical delay times within the network - signal propagation time on the bus and the internal delay time of the CAN nodes.^[2] The Prop_Seg in the CAN bit time is used to compensate for the physical delay on the network. CAN protocol's non-destructive arbitration and in-frame acknowledgement features allows transmission from multiple nodes at the same time. In the case of non-destructive arbitration, more than one node can start transmission during the arbitration field. Each node transmitting the arbitration field, samples the data from the bus to determine whether it has won the arbitration or lost the arbitration. If a node loses the arbitration, then it has to accept the arbitration field. When any node samples any bit, the value sampled must be compliant to CAN standards. In the case of the acknowledgement field, the transmitting node transmits a recessive bit and expects to receive a dominant bit.^[6]

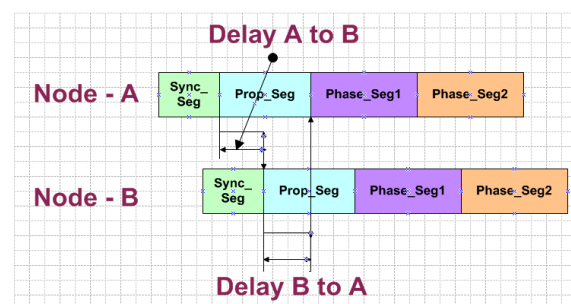


Figure 2: Two way propagation delay

Figure 2 shows the propagation delay between two nodes; the bit transmitted by Node A reaches at node B after Delay A to B and the bit transmitted by Node B reaches at node A after Delay B to A. The bit

transmitted by node B must be received at node A before end of Prop_Seg of node A, to be sampled correctly by node A. Maximum propagation delay consists of Tx node output delay plus bus line delay plus Rx node input delay.^[2]

$$\text{Maximum propagation delay} \geq 2 * [\text{MAX}(\text{Tx node output delay} + \text{bus line delay} + \text{Rx node input delay})] \quad (1)$$

Note: The reason for the MAX operator is to consider the delay of all transceivers and the physical cable delay between all transceivers.

For a CAN bit time, all the Time Quanta's before the sampling point can be considered for propagation delay. However, because of oscillator tolerances, which are described in next section, the propagation time will be reduced by the maximum oscillator tolerance. If the oscillator tolerance is 0%, then all the Time Quanta's before the sampling point can be considered for propagation delay.

3. Synchronization

All nodes in the system must be synchronized while receiving a transmission. The beginning of each received bit is expected to occur in the Sync_Seg. However, because of differences in oscillator frequencies and changes in propagation delay due to the cable length between the transmitter and receiver, the bit might get received either before or after the Sync_Seg. This is called phase error.

Phase error is managed by two types of synchronizations – hard synchronization and re-synchronization. Hard synchronization occurs at every Start-Of-Frame when recessive to dominant bit edge is detected after the intermission or idle period. Re-synchronization occurs during the whole frame whenever a recessive to dominant bit edge is detected outside of the Sync_Seg. Re-synchronization is achieved by lengthening Phase_Seg1 or shortening the

Phase_Seg2 by each CAN controller during message reception.^[5]

4. Oscillator tolerance

The Phase_Seg1, Phase_Seg2 and the RJW are used to compensate for the oscillator tolerance. In a CAN based system each node has its own clock source – typically a quartz oscillator. Oscillators are expected to generate a specific output frequency, however practically small variations can be observed in the frequency due to manufacturing variations, changes in temperature or voltage and by changes due to time. Additionally it may not be possible to achieve the exact bit rate based on the given oscillator frequency for a given node. This frequency deviation from the desired frequency will result in sampling point drift. If the sampling point drifts out of the bit time, then it will result in fault error detection. These are local errors and will degrade performance of the CAN network.

To overcome this issue, RJW was introduced to lengthen Phase_Seg1 or shorten Phase_Seg2, which shifts the sampling point location in the bit time to compensate for differences of clock frequency. RJW sets the maximum limit for lengthening Phase_Seg1 or shortening Phase_Seg2. In other words RJW sets the limit of maximum allowed oscillator tolerance. The value of RJW should be chosen based on maximum differences between oscillators. The smaller the variation in oscillator frequency, the smaller value can be assigned to RJW. The value of RJW should be same at all nodes in the network.^[6]

The tolerance range for an oscillator's frequency f_{osc} around the nominal frequency f_{nom} depends on Phase_Seg1 and Phase_Seg2, RJW and the bit time (NBT). The maximum allowed tolerance df of f_{osc} with $[(1 - df) * f_{nom} \leq f_{osc} \leq (1 + df) * f_{nom}]$ is defined by two conditions (both shall be met):^[2]

$$I. df \leq \text{MIN}(\text{Phase_Seg1}, \text{Phase_Seg2}) / [2 * (13 * \text{bit time} - \text{Phase_Seg2})] \quad (2)$$

$$\text{II. } df \leq \text{RJW} / (20 * \text{bit time}) \quad (3)$$

The maximum difference between frequencies of two oscillators must be $\leq 2 * df * f_{\text{nom}}$. During non-error conditions, the maximum length between two edges is 10 bits including 5 consecutive dominant bits and 5 consecutive recessive bits. If the 6th bit is erroneously read as dominant then node will send an error flag with bits 7 to 12 as dominant. To identify whether this is local or global error, next bit i.e. 13th bit needs to be evaluated correctly. If this bit is recessive, it indicates global error and if dominant then local error. This identification is required to increase the error count.^[6] So considering erroneous condition the oscillator tolerance is calculated as follows –

$$(2 * df) * (13 * \text{Bit Time} - \text{Phase_Seg2}) \leq \text{MIN}(\text{Phase_Seg1}, \text{Phase_Seg2}) \quad (4)$$

And considering non-error conditions the oscillator tolerance is calculated as follows –

$$(2 * df) * 10 * \text{bit time} \leq \text{RJW} \quad (5)$$

These two equations (4) and (5) results in equation (2) and (3) mentioned above.

5. Ready reckoner

Considering all these requirements, here we attempt to create a ready reckoner using Microsoft Excel. For all values of NBT, this ready reckoner lists out all the permissible combinations of bit time segments and RJW along with sampling point and maximum allowed oscillator tolerance for each combination. Also the maximum propagation delay is calculated in microseconds for the desired bit rate for all the permissible combinations. This ready reckoner makes it easy to choose proper values of NBT along with bit time segments and RJW. The ready reckoner follows the following procedure. User has to follow the procedure starting from section 7, to use this ready reckoner.

Ready Reckoner Step1:

Calculated all valid combinations of bit time segments and RJW for all values of NBT, considering all CAN protocol requirements. For this example we are using 125K bit rate.

- 1.1. For a given value of NBT, starting from $\text{NBT} = 25 T_q$, list all values of Tseg1 starting from 16 to 2 without applying any CAN protocol requirements.
- 1.2. For each value of Tseg1, list all values of RJW starting from 1 to 4. Each value of Tseg1 is repeated 4 times with different value of RJW.
- 1.3. For each combination of Tseg1 and RJW, calculate Prop_Seg and Phase_Seg1 values. $\text{Prop_Seg} = \text{if}(\text{Tseg1} - \text{RJW} > 8, 8, \text{Tseg1} - \text{RJW})$. $\text{Phase_Seg1} = \text{Tseg1} - \text{Prop_Seg}$. These values are calculated to program CAN controllers which have separate fields to program Prop_Seg and Phase_Seg1.
- 1.4. Calculate the maximum length of Prop_Seg and Phase_Seg1 which will be used by CAN controllers. $\text{Prop_Seg_Len} = \text{Sync_Seg} + \text{Tseg1} - \text{RJW}$. $\text{Phase_Seg1_Len} = \text{RJW}$. Many CAN controllers consider Phase_Seg1 = RJW and remaining Time quanta's from Tseg1 are considered for propagation delay. $t_{\text{PROP}} = t(\text{Tseg1} - \text{RJW})$.
- 1.5. Calculate $\text{Tseg2} = T_q - \text{Tseg1} - \text{Sync_Seg}$.
- 1.6. Calculate the sampling point (SP) for all the combinations. $\%SP = ((\text{Sync_Seg} + \text{Tseg1}) / \text{NBT}) * 100$.
- 1.7. Calculate maximum allowed oscillator tolerance for all the combinations using equations (2) and (3) mentioned in Oscillator tolerance section. To meet both the conditions, minimum oscillator tolerance value is chosen out of both the calculated values.

1.8. Calculate maximum physical delay allowed using Prop_Seg_Len value for 125K bit rate. Delay in usec = $(1/(125000 * NBT)) * Prop_Seg_Len * 1000000$. For other bit rates maximum delay can be calculated by changing bit rate in the above equation.

1.9. Apply the following CAN protocol requirements for all the combinations -

- $1 \leq Phase_Seg1 \leq 8$ (For 1 Sample point)
- $2 \leq Phase_Seg1 \leq 8$ (For 3 Sample point). Note that this ready reckoner considers only 1 sampling point.
- $1 \leq Phase_Seg2 \leq 8$
- $1 \leq Prop_Seg_Len \leq 16$
- $RJW = \text{MIN}(4, Phase_Seg1, Phase_Seg2)$
- $NBT > Phase_Seg1 + Phase_Seg2$
- $NBT > 2 * RJW$

Filter out all the combinations which do not comply with these requirements. This leaves all the permissible combinations for the given value of NBT.

1.10. Repeat same procedure for all values of NBT from 24 to 8.

1.11. Finally all the permissible combinations of CAN bit time segments and RJW along with sampling point, maximum oscillator tolerance and maximum delay for 125K bit rate are available for all values of NBT.

1.12. Apply data filter for all the data available for all the values of NBT.

After following above steps 1.1 to 1.12 we get results as shown in Table 2. Results are shown for NBT = 25 and 24 and not for other values of NBT.

NBT	Sync	Tseg1	Tseg2	RJW	Prop	Pseg1	SP	Osc	Delay
Tq	Seg				Seg		%	Tol	usec
25	1	16	8	1	8	8	68	0.15	5.12
25	1	16	8	2	8	8	68	0.31	4.8
25	1	16	8	3	8	8	68	0.47	4.48
25	1	16	8	4	8	8	68	0.63	4.16
24	1	16	7	1	8	8	70.83	0.16	5.33
24	1	16	7	2	8	8	70.83	0.32	5
24	1	16	7	3	8	8	70.83	0.49	4.66
24	1	16	7	4	8	8	70.83	0.65	4.33
24	1	15	8	1	8	7	66.66	0.16	5
24	1	15	8	2	8	7	66.66	0.32	4.66
24	1	15	8	3	8	7	66.66	0.49	4.33
24	1	15	8	4	8	7	66.66	0.65	4

Table 2: Bit timing parameter combinations

Ready Reckoner Step2:

2.1. Calculate BRP values for all values of NBT from 8 to 25, for given bit rate and CAN clock frequency.

$$BRP = \text{CAN Clock Freq} / (\text{NBT} * \text{bit rate})$$

2.2. Round-off BRP values to nearest integer values because fractional part cannot be programmed into CAN configuration registers.

$$BRP(\text{Rounded}) = \text{INT}(BRP + 0.5)$$

2.3. Calculate actual bit rate with rounded BRP values.

$$\text{Actual bit rate} = \text{CAN Clock Freq} / (BRP(\text{Rounded}) * NBT)$$

2.4. Calculate deviation in the actual bit rate from the requested bit rate.

$$\% \text{Bit rate Deviation} = ((\text{Actual bit rate} * 100) / \text{Requested bit rate}) - 100$$

Table 3 shows BRP values and bit rate deviation, calculated for 125K bit rate with 48MHz CAN clock frequency. Here it can be observed that, specified bit rate is achievable with NBT = 8, 12, 16 and 24 Tq, without any deviation in the desired bit rate. These values are highlighted in PINK. For

all other NBT values there is some deviation in the bit rate. Considering maximum allowable oscillator tolerance +/-1.58%, some more values of Tq are available which can be used to get the desired bit rate. All of these values are highlighted in YELLOW.

Baud Rate-->		125K	
		125000	
CAN Clock Freq	NBT Tq	BRP (Rounded)	Bit Rate Deviation %
48000000	8	48	0
48000000	9	43	-0.775193798
48000000	10	38	1.052631579
48000000	11	35	-0.25974026
48000000	12	32	0
48000000	13	30	-1.538461538
48000000	14	27	1.587301587
48000000	15	26	-1.538461538
48000000	16	24	0
48000000	17	23	-1.79028133
48000000	18	21	1.587301587
48000000	19	20	1.052631579
48000000	20	19	1.052631579
48000000	21	18	1.587301587
48000000	22	17	2.673796791
48000000	23	17	-1.79028133
48000000	24	16	0
48000000	25	15	2.4

Table 3: BRP Calculation

6. Trend Charts

Trend charts for all the permissible combinations achieved in Ready Reckoner Step1, are plotted against the sampling point and oscillator tolerance and shown in Figures 3 and 4.

As the sampling point decreases, oscillator tolerance increases and vice versa. Highest sampling point = 89.47% can be achieved with NBT = 19, Tseg1 = 16, Tseg2 = 2 and RJW = 1 Tq, where Tseg1 is at highest and Tseg2 is at lowest value of those respective segments. Highest oscillator tolerance = 1.58% can be achieved with NBT = 10, Tseg1 = 4, Tseg2 = 4 and RJW = 4 Tq.

Here also it can be observed that, sampling point and hence the propagation delay time decreases above NBT = 19 Tq. This is because maximum value of TSeg1 is limited to 16 Tq. When total number of Tq in NBT are increased above 19, TSeg1 value cannot be increased beyond 16, and Tseg2 value needs to be increased. On the other

side, increase in total number of Tq in NBT increases CAN clock frequency which reduces the propagation delay time with Tseg1 = 16 Tq.

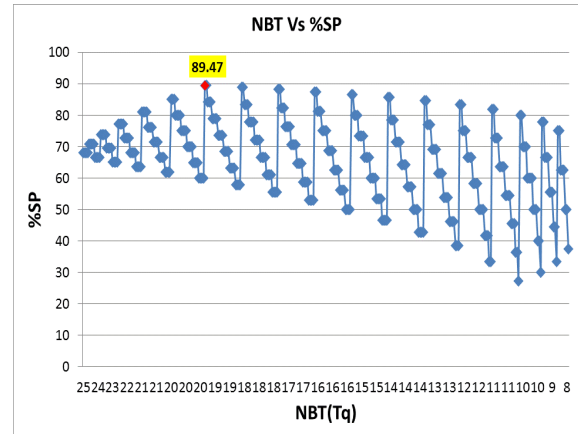


Figure 3: NBT (Tq) Vs % Sampling Point

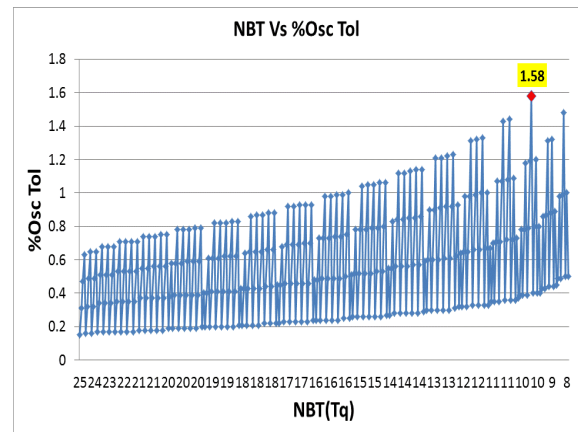


Figure 4: NBT (Tq) Vs % Oscillator Tolerance

Although NBT = 19 Tq gives highest sampling point, most of times it is not possible to use that value in the CAN bit rate settings. This is because, with standard oscillators it is very difficult to achieve BRP value with 0% deviation in the CAN clock frequency for NBT = 19 Tq. In Table 3 we can observe that for NBT = 19 Tq, the deviation in CAN clock frequency is 1.05%. So most of the documents on internet suggests to choose sampling point = 87.5%, with NBT=16, Tseg1=13, Tseg2=2 and RJW=1 or 2 Tq.

7. Bit time parameter selection using ready reckoner

This includes selection of appropriate NBT value along with bit time segment and RJW values using above ready reckoner. Perform the following steps.

Step1:

Calculate the propagation delay from the cable specifications and transceivers datasheets.

Propagation delay $\geq 2 * [\text{MAX}(\text{Tx node output delay} + \text{bus line delay} + \text{Rx node input delay})]$

Typical CAN cable propagation delay is 5ns/m. However it should be confirmed with the cable specifications that being used.

Step2:

Identify the maximum oscillator tolerance among all the nodes in the network. Oscillator tolerance is specified in PPM (Parts Per Million). 1000ppm oscillator tolerance = +/-0.1%.

Step3:

In BRP calculation excel sheet in the ready reckoner, set the CAN clock frequency and get the BRP values and bit rate deviation.

Step4:

From BRP calculation excel sheet note the NBT values with clock deviation = 0%. If no option available with 0% clock deviation then follow Step5, else go to Step6.

Step5:

From BRP calculation excel sheet note NBT values with \leq +/-1.58% bit rate deviation. If no option available, then exit (No Solution).

Step6:

From the CAN bit time parameter selection Excel sheet:

6.1. Filter out all the Time Quanta values except the values from Step4 or Step5.

6.2. To be on safer side, give 5 to 10% extra margin for propagation delay. Filter out all the propagation delay values $<$ propagation delay calculated in Step1 + extra margin (5 to 10%).

6.3. Filter out all the combinations having oscillator tolerance less than tolerance identified in Step2.

Step7:

At this point we have the combination/s of values which satisfy the given requirements. If more than one combination is available then pick the combination that gives flexibility in a desired attribute. E.g. increasing the cable length will increase the propagation delay.

For Example:

- Required bit rate = 125K
- CAN clock frequency = 48MHz
- Line/cable delay = 5 ns/m.
- Cable length = 500m
- For PCA82C250 transceiver, transmit and receive propagation delay is 155 ns at 125 degree C in High Speed Mode.
- Maximum Oscillator tolerance = 0.1%.

Step1:

Required propagation delay = $2 * ((500 * 5) + 155) = 5.31\mu\text{s}$

Step2:

Maximum oscillator tolerance = 0.1%.

Step3:

Calculate BRP values with 48MHz CAN clock frequency for 125K bit rate. BRP values are calculated in Table 3.

Step4:

NBT = 8, 12, 16 and 24 Tq with clock deviation = 0%. Skip Step5.

Step6:

From the CAN bit time parameter selection Excel sheet in the ready reckoner:

- 6.1. Filter out all NBT values except NBT = 8, 12, 16 and 24 Tq.
- 6.2. Propagation delay with 10% margin = $5.31+0.531 = 5.841\text{usec}$. Filter out all NBT values with delay $< 5.841\text{usec}$.
- 6.3. Filter out all NBT values with oscillator tolerance $< 0.1\%$.

Step7:

We remain with the following NBT values along with bit time segments and RJW value combinations as shown in Table 4. There are four combinations available which satisfies the given requirements. The best option is the first as it allows more propagation delay and still satisfies the oscillator tolerance requirement.

NBT Tq	Sync Seg	Tseg1	Tseg2	RJW	Prop Seg	Pseg1	SP %	Osc Tol	Delay usec
16	1	13	2	1	8	5	87.5	0.24	6.5
16	1	13	2	2	8	5	87.5	0.48	6
16	1	12	3	1	8	4	81.25	0.24	6
12	1	9	2	1	8	1	83.33	0.32	6

Table 4: CAN bit time parameter selection

8. Conclusion and future works

In absence of such ready reckoner, it is very confusing for an engineer to choose a proper combination of bit time segment and RJW among multiple available options. The ready reckoner showing the trend with impact on propagation delay and oscillator tolerance for different bit timing parameters helps engineers to analyze the requirements and choose one of the best combinations. The present implementation is done in excel. Future work includes a web based interface and a PC based GUI.

9. Acknowledgements

I wish to acknowledge the following colleagues from OTIS Elevator who provided review and guidance:

1. Arthur Grondine, Otis Engineering /Farmington, CT, USA
2. Dang Nguyen, Otis Engineering /Farmington, CT, USA
3. Pravin Jain, Otis Engineering /Pune, India

Meenanath Taralkar
 OTIS ISRC Pvt. Ltd.
 Pune - 411006, INDIA
 Tel: +91-20-66837371
 Fax: +91-20-66837385
 meenanath.taralkar@otis.com
 www.otisisrc.com

References

- [1] "CAN Specification", Version 2.0, Parts A and B, Robert Bosch GmbH, 1991
- [2] Florian Hartwich, Armin Bassemir, "The Configuration of the CAN Bit Timing", 6th International CAN Conference 2nd to 4th November, Turin (Italy)
- [3] Lawrenz, Wolfhard, "CAN System Engineering From Theory to Practical Applications", Springer, 1997
- [4] Freescale BCANPSV2.0/D
- [5] Stuart Robb, East Kilbride – Freescale Semiconductors - AN1798 - CAN Bit Timing Requirements
- [6] Lars-BernoFredriksson KVASER – Controller Area Network(CAN) discussion forum
- [7] "APPLICATION NOTE PCA82C250 / 251 CAN Transceiver AN96116" – Philips semiconductors