

# ***Frequency-Modulated PLL Impact on Controller Area Network (CAN) Communication***

*Peter Steffan and Kevin Lavery*
*TMS470 Microcontroller*

## **ABSTRACT**

Asynchronous communication protocols rely on each node operating at the same frequency. Modulating the microcontroller operating frequency is possible when the modulation frequency is large enough. Unfortunately, the EMC-reducing aspects of clock modulation are most effective when the modulation frequency is small. This application note defines the minimum modulation frequency in relation to the CAN synchronization jump width parameter.

## **Contents**

<b>1</b>	<b>General CAN Considerations Regarding Resynchronization Pulses and the Synchronization Jump Width</b> .....	<b>2</b>
<b>2</b>	<b>Accumulated Error Due to Static Frequency Tolerances</b> .....	<b>3</b>
<b>3</b>	<b>Accumulated Error Due to the FMPLL</b> .....	<b>4</b>
<b>4</b>	<b>Implications of the FMPLL on CAN Communication</b> .....	<b>6</b>
<b>5</b>	<b>Reference</b> .....	<b>13</b>
<b>Appendix A Derivation of Equations</b> .....		<b>14</b>

## **List of Figures**

Figure 1. FMPLL Waveform Illustrating Pertinent Parameters .....	4
Figure 2. Abstracted FMPLL Waveform Illustrating Pertinent Parameters .....	5
Figure 3. CAN Bit Relative to ICLK .....	7
Figure 4. CAN Bit Readjusts to Phase Error (With SJW) .....	8
Figure A-1. FMPLL Parameters and Slope Calculation .....	14

## **List of Tables**

Table 1. Typical Tolerance for Crystal and Ceramic Resonators .....	3
Table 2. Minimum Allowable Modulation Frequency (KHz) .....	6

Trademarks are the property of their respective owners.

## 1 General CAN Considerations Regarding Resynchronization Pulses and the Synchronization Jump Width

As CAN is an asynchronous protocol, all nodes of the CAN bus must operate at the same bit rate. Every bus node assures itself of being synchronized to the bus by detecting a recessive-to-dominant bus edge as a synchronization signal within a predefined segment of the CAN bit. From this resynchronization edge, each node has a fixed time from resynchronization edge to the bit's sample point (determined by the node's CAN bit definition). When the CAN bus has multiple bits of the same polarity in sequence, the next bit's resynchronization is implied (no edge is seen at the CAN node) and the sample point is a fixed time from the implied resynchronization. After 10 or 13 CAN bit times (worst case), the CAN protocol forces a resynchronization edge on the bus, and every CAN node must resynchronize to the edge.

Two CAN nodes drift relative to each other due to frequency and phase offsets between nodes of the CAN bus, and the resynchronization protocol is intended to negate that drift. Every node of the CAN bus must have a minimum time step (synchronization jump width) over which it can resynchronize to account for static frequency offsets and varying phase at the receiver node. When the synchronization edge falls within the synchronization segment of the CAN bus, the node is synchronized. If the synchronization edge falls outside of the synchronization segment, the CAN mode must resynchronize to the CAN bus.

The CAN node is permitted to resynchronize positively or negatively by up to one synchronization jump width ( $t_{SJW}$ ) in order to maintain a constant timing between the data transition and the sample point. That is, if the synchronization edge is more than  $t_{SJW}$  away from the synchronization segment, the CAN node cannot resynchronize. Resynchronization requires that the accumulated error<sup>1</sup> between any two nodes of the CAN bus (due to frequency offset and phase error) must be less than the smaller  $t_{SJW}$ .

$$\text{Accumulated Error}_{\text{Transmitter}} + \text{Accumulated Error}_{\text{Receiver}} < (t_{SJW})_{\text{smaller}}$$

Given that neither the transmitter or receiver should account for more than half of the synchronization jump width, the accumulated error of any node must be less than or equal to  $\frac{1}{2} t_{SJW}$ . Specifically, the critical synchronization jump width is the receiver with the *smallest* synchronization jump width on the CAN bus.<sup>2</sup>

$$\text{Accumulated Error}_{\text{NodeX}} < \frac{t_{SJW}}{2}$$

$t_{SJW}$  is the synchronization jump width of the CAN (receiver) node with the smallest  $t_{SJW}$ .

Due to the protocol definitions, the resynchronization is performed after at least 10 bit periods during normal data frames, or after 13 bit periods in case of bit errors. The accumulated error may therefore not exceed  $\frac{1}{2} t_{SJW}$  at the end of this time period.

<sup>1</sup> Though this footnote deals with frequency "errors" due to the frequency-modulation, other parameters contribute to the accumulated error. The typical accumulated error contributor is oscillator tolerance. Notice that the worst case oscillator tolerance needs to be added.

<sup>2</sup> This restriction is a little over-broad and is used to compute the CAN timings *in general*. If, in a specific system, the receiver with the smallest jump width were known to have minimal accumulated error, the restriction could be loosened using the relation:  $\text{Accumulated Error}_{\text{Tx}} + \text{Accumulated Error}_{\text{Rx}} < (t_{SJW})_{\text{smaller}}$ .

For many CAN systems, the design rules specify a maximum oscillator tolerance in order to ensure that the nodes are able to synchronize; the oscillator tolerance correctly accounts for frequency errors. An FMPLL creates a varying phase rather than a static frequency offset. For FMPLL-based CAN nodes, the oscillator tolerance is not an applicable parameter, therefore an exact analysis of bit time errors in the 10 to 13 bit time range is explained in the following section.

## 2 Accumulated Error Due to Static Frequency Tolerances

The accumulated error of any two nodes can be divided into: Accumulated Error = Accumulated Error<sub>frequency offset</sub> + Accumulated Error<sub>phase</sub>. This application report discusses the phase errors due to modulating frequency, but static errors must also be considered. The easiest way to treat static, or frequency-offset, errors is to compute the worst case frequency offset error and subtract the error from the SJW prior to following the algorithm shown in the remainder of this application report. Although the problem of determining the tolerance of CAN nodes is well understood in CAN applications, a short discussion is provided.

Two nodes operate at CAN frequencies  $f_1$  and  $f_2$ . Time quanta must be chosen such that

$$\frac{(\# \text{ of time quanta})_1}{f_1} = \frac{(\# \text{ of time quanta})_2}{f_2} = 1t_{\text{NBT}}$$

The frequencies differ from their ideal frequencies due to:

- manufacturing tolerance ( $\Delta f/f$ )
- frequency stability over temperature
- frequency stability over aging

Typical tolerances for crystal and ceramic resonators are shown in Table 1.

**Table 1. Typical Tolerance for Ceramic and Crystal Resonators**

	Crystal	Resonator
manufacturing tolerance ( $\Delta f/f$ )	±50 PPM	±5000 PPM
frequency stability over temperature	±50 PPM	±3000 PPM
frequency stability over aging	±5 PPM/Year	±3000 PPM/Year

At the end of 10 years, the CAN bus frequencies differ by

$$\Delta t = \left[ \frac{(\# \text{ of time quanta})_1}{(f_1 + \Delta f_1 + \Delta f(T)_1 + f_{a1}(10\text{yrs}))} - \frac{(\# \text{ of time quanta})_2}{(f_2 - \Delta f_2 - \Delta f(T)_2 - f_{a2}(10\text{yrs}))} \right]$$

$$\Delta t = \left[ \frac{1}{\left(1 + \frac{\Delta f_1}{f_1} + \frac{\Delta f(T)_1}{f_1} + \frac{f_{a1}(10\text{yrs})}{f_1}\right)} - \frac{1}{\left(1 - \frac{\Delta f_2}{f_2} - \frac{\Delta f(T)_2}{f_2} - \frac{f_{a2}(10\text{yrs})}{f_2}\right)} \right] t_{\text{NBT}}$$

The  $\Delta t$  computed over ten CAN bit times should be subtracted from  $1 t_{\text{SJW}}$ .

Consider the worst case impact of the crystal shown in the table above. Over time and temperature, the crystals may differ from their ideal frequency by 150 PPM. In the worst case example, one differs by +150 PPM, while the second crystal varies by -150 PPM. The worst case time difference (over ten bit times) is

$$\Delta t = 0.003 * t_{NBT}$$

In this case, the adjustment due to static frequency offset is 3 ns (for a 1MBit/s CAN). The phase error must be less than  $t_{SJW}(\text{bus minimum}) - t_{\text{frequency offset}}$ .

### 3 Accumulated Error Due to the FMPLL

The FMPLL modulates the clock frequency using a triangular modulation, resulting in a phase error that varies proportional to the square of time. The modulation frequency ( $f_{FM} = 1/T_{MOD}$ ) determines how fast the modulation occurs. The modulation depth ( $\Delta f$ ) determines the maximum PLL offset – that is, the minimum and maximum clock frequencies. Normally, the modulation depth is expressed as a percentage of the nominal frequency –  $\Delta f/f_{nom}$ .

Looking at a possible setting:

$$f_{xtal} = 7.5 \text{ MHz}$$

$$(f_{nom})_{SYSCLK} = 60 \text{ MHz}$$

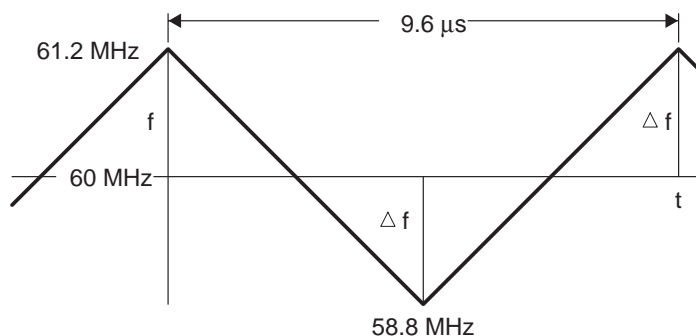
$$f_{FM} = 104 \text{ kHz}$$

$$T_{MOD} = 9.6 \mu\text{s}$$

$$\Delta f/f_{nom} = 2\%$$

$$f_{max} = 61.2 \text{ MHz}$$

$$f_{min} = 58.8 \text{ MHz}$$



**Figure 1. FMPLL Waveform Illustrating Pertinent Parameters**

As shown in the graph, the worst case frequency offset occurs when sampled about the extremes of the waveform. The accumulated error is also greatest when calculated around the extremes of the waveform (e.g. when the SYSCLK frequency is near 58.8 MHz or 61.2 MHz).

Figure 1 can be abstracted to Figure 2.

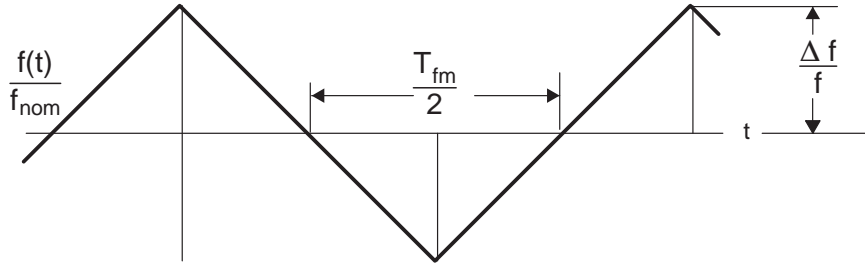


Figure 2. Abstracted FMPLL Waveform Illustrating Pertinent Parameters

The accumulated phase error due to a triangular frequency modulation of the clocks is:

$$\text{Accumulated Error} = \begin{cases} \frac{\Delta f}{f_{\text{nom}}} \left( 13t_{\text{NominalBitTiming}} - 169f_{\text{fm}} t_{\text{NominalBitTiming}}^2 \right) & \text{when } 13t_{\text{NominalBitTiming}} \leq \frac{T_{\text{fm}}}{2} & (1) \\ \frac{\left( \frac{\Delta f}{f_{\text{nom}}} \right)}{4f_{\text{fm}}} & \text{when } 13t_{\text{NominalBitTiming}} \geq \frac{T_{\text{fm}}}{2} & (2) \end{cases}$$

$$\frac{\Delta f}{f_{\text{nom}}} \left( 13t_{\text{NominalBitTiming}} - 169f_{\text{fm}} t_{\text{NominalBitTiming}}^2 \right) < \frac{(t_{\text{SJW}})_{\text{smallest}}}{2} \leq 13t_{\text{NominalBitTiming}} \geq \frac{t_{\text{fm}}}{2}$$

In Equation 1, the modulation frequency is so low that the greatest accumulated error would occur after 13 CAN bits. Since, the CAN protocol requires correct sampling on the 13<sup>th</sup> CAN bit [following a stuff error], the worst case accumulated error is on the 13<sup>th</sup> bit. [For a 500 kBit/s CAN rate, the modulation frequency that meets this requirement is 19 kHz or less.]

In almost all cases, the PLL modulation frequency is large enough that Equation 2 governs the accumulated error.<sup>3</sup> The accumulated error must be smaller than one-half the smallest synchronization jump width.

$$\frac{\left( \frac{\Delta f}{f_{\text{nom}}} \right) \frac{t_{\text{fm}}}{2}}{2} = \frac{\left( \frac{\Delta f}{f_{\text{nom}}} \right)}{4f_{\text{FM}}} < \frac{(t_{\text{SJW}})_{\text{smallest}}}{2}$$

<sup>3</sup> The effect of the oscillator tolerance can be approximated by  $\frac{\left( \frac{\Delta f}{f_{\text{nom}}} \right)}{4f_{\text{FM}}} + \frac{\text{OT}}{2f_{\text{FM}} f_{\text{nom}} t_{\text{NBT}}} < \frac{(t_{\text{SJW}})_{\text{smallest}}}{2}$

## 4 Implications of the FMPLL on CAN Communication

$$\frac{\left(\frac{\Delta f}{f_{nom}}\right)}{4f_{FM}} < \frac{(t_{SJW})_{smallest}}{2}$$

- PLL modulation settings are dependent upon the CAN node with the shortest synchronization jump width time, not necessarily the microcontroller’s settings.
- Increased modulation frequency is beneficial for achieving CAN timings, but not as useful for reducing electromagnetic emissions.

In Table 2, minimum modulation frequencies are computed versus minimum synchronization jump widths and modulation depth.

**Table 2. Minimum Allowable Modulation Frequency (KHz)**

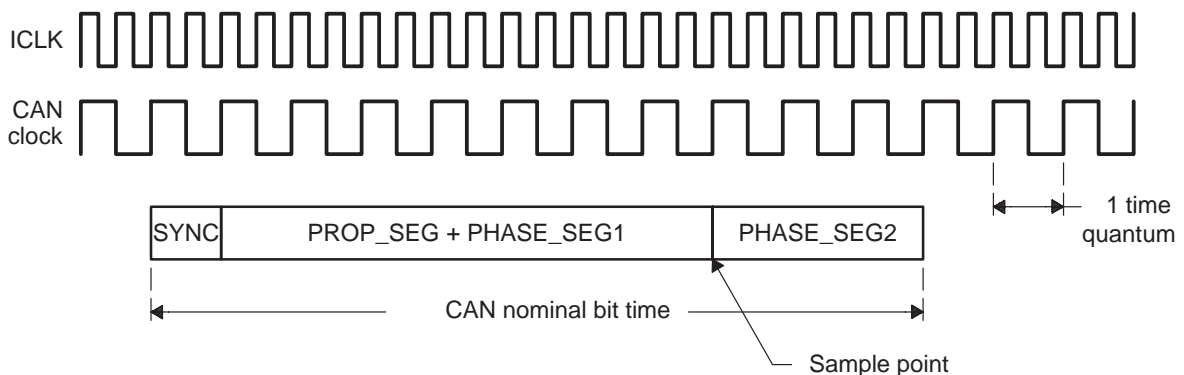
$(t_{SJW})_{smallest}$ (ns)	Modulation Depth			
	4.0%	2.0%	1.0%	0.5%
20.0	1000.0	500.0	250.0	125.0
40.0	500.0	250.0	125.0	62.5
60.0	333.3	166.7	83.3	41.7
80.0	250.0	125.0	62.5	31.3
100.0	200.0	100.0	50.0	25.0
120.0	166.7	83.3	41.7	20.8
140.0	142.9	71.4	35.7	17.9
160.0	125.0	62.5	31.3	15.6
180.0	111.1	55.6	27.8	13.9
200.0	100.0	50.0	25.0	12.5

For a CAN bus on which the smallest  $t_{SJW} = 100$  ns, a 2% modulation depth requires a modulation frequency of 100 kHz or greater.

**Example 1.**

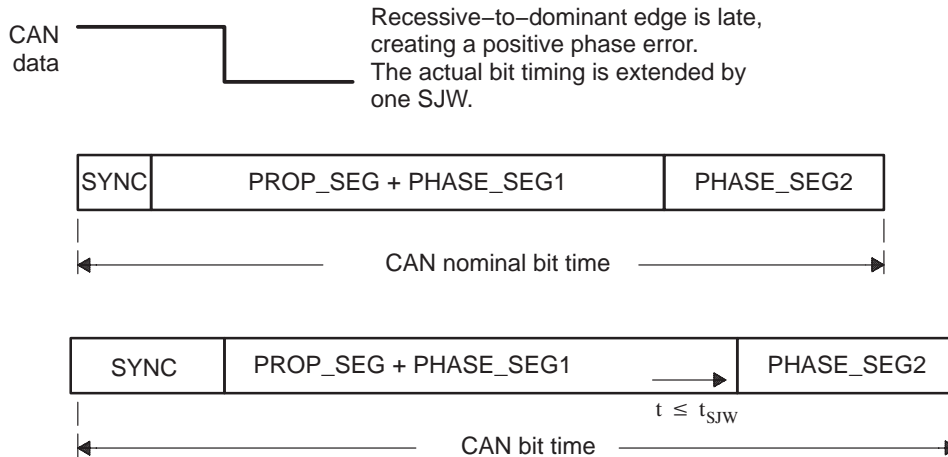
Nominal Frequency (SYSCLK)	60 MHz
Modulation Depth ( $\Delta f/f_{nom}$ )	2% (61.2 MHz to 58.8 MHz)
Modulation Frequency ( $f_{FM}$ )	104 KHz
CAN bitrate	1 MBit/s

- Assume that the microcontroller is known to have the smallest SJW on the CAN bus. (Notice that this is not necessarily true but is useful for the example.)
- The CAN bit is divided into multiple “segments” – Synchronization segment, propagation segment, phase\_segment1, and phase\_segment2. The CAN receiver nodes sample the bit between phase\_segment1 and phase\_segment2. Thus, by adjusting the various lengths of the segments, the user can position the sample point within the CAN bit.
  - The CAN clock (on TMS470, ICLK provides the clock to the CAN) is divided by the BRP (CANBTC.23:16).  $BRP_{CALC} = BRP + 1$ . Each resulting clock tick is one time quantum.
  - Time quanta are summed to create the CAN bit time. On TMS470 microcontrollers:
    - the propagation\_segment and phase\_segment1 are programmed as one block TSEG1(CANBTC.6:3).  $TSEG1_{CALC} = TSEG1 + 1$ . TSEG1\_CALC represents the number of time quanta in the block TSEG1.
      - The propagation segment “is used to compensate for the physical delay times within the network. It is twice the sum of the signal’s propagation time on the bus line, the input comparator delay, and the output driver delay.” *Can Specification* [1]
    - phase\_segment2 is programmed with TSEG2 (CANBTC.2:0).  $TSEG2_{CALC} = TSEG2 + 1$ . TSEG2\_CALC represents the number of time quanta in the block TSEG2.
      - The phase segments (phase\_segment1 and phase\_segment2) are “used to compensate for edge phase errors” – lengthened and shortened by resynchronization. *CAN Specification* [1]
    - synchronization\_segment is always one time quantum.



**Figure 3. CAN Bit Relative to ICLK**

- SJW depends upon how the CAN is programmed, and the SJW timing will determine whether the modulation frequency of the FMPLL is valid. When SJW is programmed to one time quantum, it can resynchronize to phase errors of  $\pm 1$  time quantum; the purpose of the resynchronization is to maintain a constant timing between the data transition and the sampling point. (The example shown below has a one time quantum phase error in the positive direction. If programmed such that  $SJW = 1TQ$ , this phase error represents the maximum shift possible in this configuration.)



**Figure 4. CAN Bit Readjusts to Phase Error (With SJW)**

The following two scenarios, Example 1 (part a) and Example 1 (part b), show how the CAN programming leads to different SJW times and different allowable modulation frequencies.



**Example 1 (part a)**

SYSCLK = 60 MHz

ICLK = 20 MHz

 $BRP_{CALC} = 2$ 

In this configuration, the CAN clock runs at 10 MHz ( $I_{CLK}/BRP_{CALC}$ ); each time quantum is 100 ns.

A 1 MBit/s CAN rate implies that each CAN bit must be composed of ten time quanta. The ten time quanta must be allocated between the synchronization segment [always one time quantum] and the two phase segments,  $TSEG1_{CALC}$  and  $TSEG2_{CALC}$ . (Additional constraints exist on  $TSEG1$  and  $TSEG2$ . For more information, please see the *TMS470R1x Controller Area Network (CAN) Reference Guide* (literature number SPNU197).[2]

 $TSEG1_{CALC} = 6$ 
 $TSEG2_{CALC} = 3$ 

SJW must be programmed such that it is at least one time quantum. Additionally, SJW cannot exceed the minimum of four time quanta or  $TSEG2_{CALC}$ . (In this case,  $TSEG2_{CALC} < 4t_q$ , so  $t_{SJW} \leq TSEG2_{CALC} = 3t_q$ ).

$$\frac{\left(\frac{\Delta f}{f_{nom}}\right)}{4f_{FM}} < \frac{t_{SJW}}{2}$$

$$\frac{\left(\frac{\Delta f}{f_{nom}}\right)}{2t_{SJW}} < f_{FM}$$

$$\frac{\left(\frac{\Delta f}{f_{nom}}\right)}{2t_{SJW}} = \frac{2\%}{2 * t_{SJW}} < f_{FM}$$

SJW		$f_{FM}$ (minimum)
1	100 ns	100 kHz
2	200 ns	50 kHz
3	300 ns	33 kHz
4	Not a valid configuration since $TSEG2_{CALC} = 3$	

Any of these configurations is valid for a 104 kHz modulation frequency.

**Example 2 (part b)**

SYSCLK = 60 MHz

ICLK = 15 MHz

BRP<sub>CALC</sub> = 1

In this configuration, the CAN clock runs at 15 MHz (ICLK/BRP<sub>CALC</sub>); each time quantum is 67 ns.

A 1 MBit/s CAN rate implies that each CAN bit must be composed of 15 time quanta. The fifteen time quanta must be allocated between the synchronization segment [always one time quantum] and the 2 phase segments, TSEG1<sub>CALC</sub> and TSEG2<sub>CALC</sub>. (Additional constraints exist on TSEG1 and TSEG2. For more information, please see the *TMS470R1x Controller Area Network (CAN) Reference Guide* (SPNU197).)

TSEG1<sub>CALC</sub> = 9TSEG2<sub>CALC</sub> = 5

SJW must be programmed such that it is at least one time quantum. Additionally, SJW cannot exceed the minimum of four time quanta or TSEG2<sub>CALC</sub>. (In this case, TSEG2<sub>CALC</sub> > 4t<sub>q</sub>, so t<sub>SJW</sub> ≤ 4t<sub>q</sub>).

$$\frac{\left(\frac{\Delta f}{f_{\text{nom}}}\right)}{4f_{\text{FM}}} < \frac{t_{\text{SJW}}}{2}$$

$$\frac{\left(\frac{\Delta f}{f_{\text{nom}}}\right)}{2t_{\text{SJW}}} < f_{\text{FM}}$$

$$\frac{\left(\frac{\Delta f}{f_{\text{nom}}}\right)}{2t_{\text{SJW}}} = \frac{2\%}{2 * t_{\text{SJW}}} < f_{\text{FM}}$$

SJW		f <sub>FM</sub> (minimum)
1	67 ns	150 kHz
2	133 ns	75 kHz
3	200 ns	50 kHz
4	267 ns	38 kHz

SJW = 1 is **not valid** for a 104 kHz modulation frequency. Other SJW values create a valid range for the 104 kHz modulation frequency.

### Example 3

Many times the CAN tolerances are specified in terms of a maximum allowable *static* offset. It is possible to translate these static timings into FMPLL settings.

(a)  $\pm 0.1\%$  oscillator tolerance for 10 bits

$$t_{SJW} \geq \pm 10(t_{NBT}) \delta$$

Each node can have a tolerance of  $\pm 0.1\%$ . Therefore, if the transmitter and receiver are  $180^\circ$  out of phase, the receiver's synchronization jump width needs to be twice the offset of any node.

$$t_{SJW} \geq 2(10(t_{NBT}) \delta)$$

For a 1 MBit CAN baud rate,

$$t_{SJW} \geq 2(10(1000\text{ns}) 0.1\%)$$

$$t_{SJW} \geq 20\text{ns}$$

Likewise, a 500 kBit CAN requires a  $t_{SJW}$  of at least 40 ns. The allowable oscillator tolerance must always be less than or equal to the CAN bus tolerance.

$$t_{SJW} \text{ (oscillator tolerance)} \leq t_{SJW} \text{ (min CAN bus)}$$

Assuming a strict equality between oscillator tolerance and minimum CAN bus synchronization jump width,

$$\frac{t_{SJW} \text{ (oscillator tolerance)}}{2} = \frac{\Delta f / f_{\text{nom}}}{4f_{\text{FM}(\text{min})}}$$

$$\frac{\Delta f / f_{\text{nom}}}{2 t_{SJW} \text{ (oscillator tolerance)}} = f_{\text{FM}(\text{min})}$$

For 4% modulation depth at 1 MBit CAN baud rate:

$$\frac{0.04}{2 (20 \text{ ns})} = 1000 \text{ kHz} = f_{\text{FM}(\text{min})}$$

CAN Baud Rate	Modulation Depth			
	4.0%	2.0%	1.0%	0.5%
1 MBit	1000	500	250	125
500 kBit	500	250	125	62
250 kBit	250	125	62	31

(b) 0.4 % oscillator tolerance for 10 bits

$$t_{\text{SJW (oscillator tolerance)}} = 2 \left( 10(t_{\text{NBT}}) \delta \right) = 2 \left( 10(t_{\text{NBT}}) \frac{4}{1000} \right)$$

Again, if the CAN bus timings are as tight as the oscillator tolerance, then

$$\frac{\Delta f / f_{\text{nom}}}{4f_{\text{FM(min)}}} = \frac{t_{\text{SJW (oscillator tolerance)}}}{2} = 10(t_{\text{NBT}})\delta$$

$$f_{\text{FM(min)}} = \frac{\Delta f / f_{\text{nom}}}{4 \left[ 10(t_{\text{NBT}}) \delta \right]}$$

$$f_{\text{FM(min)}} = \frac{\Delta f / f_{\text{nom}}}{4 \left[ 10(t_{\text{NBT}}) \frac{4}{1000} \right]}$$

Thus for a 500 kBaud CAN rate and 2% modulation depth the minimum modulation frequency is:

$$f_{\text{FM(min)}} = \frac{2/100}{4 \left[ 10 \left( \frac{1}{500 \text{ kHz}} \right) \frac{4}{1000} \right]}$$

$$f_{\text{FM(min)}} = \frac{2/100}{4 \left[ 10(2\mu\text{s}) \frac{4}{1000} \right]}$$

$$f_{\text{FM(min)}} = \frac{2/100}{4 \left[ 10(2000\text{ns}) \frac{4}{1000} \right]} = 62.5 \text{ kHz}$$

CAN Baud Rate	Modulation Depth			
	4.0%	2.0%	1.0%	0.5%
1 MBit	250	125	62	32
500 kBit	125	62.5	31	16
250 kBit	62.5	31	16	8

(c) configure the FMPLL as 2% modulation depth, 100 kHz modulation frequency

$$\frac{\Delta f}{4f_{FM}} = 10(t_{NBT}) \delta$$

$$\frac{\frac{\Delta f}{f_{nom}}}{4f_{FM} 10(t_{NBT})} = \delta_{(max)}$$

$$\frac{.02}{4(100kHz) 10(t_{NBT})} = \delta_{(max)}$$

$$\frac{5ns}{t_{NBT}} = \delta_{(max)}$$

For a 1 MBaud CAN rate:

$$\frac{5ns}{t_{NBT}} = \delta_{(max)}$$

$$\frac{5ns}{1\mu s} = 0.5\% = \delta_{(max)}$$

$$\delta = 0.5\%$$

For a CAN baud rate of 1 MBit, 0.5% (@ 10 bit) oscillator tolerance is acceptable for modulation depth of 2% and 100 kHz.

For a CAN baud rate of 500 kBit, 0.25% (@ 10 bit) oscillator tolerance is acceptable for modulation depth of 2% and 100 kHz.

For a CAN baud rate of 250 kBit, 0.125% (@ 10 bit) oscillator tolerance is acceptable for modulation depth of 2% and 100 kHz.

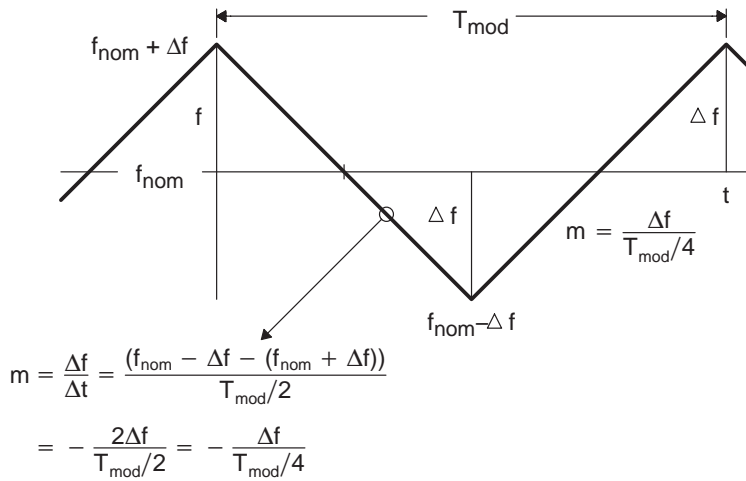
## 5 Reference

1. *CAN Specification*, Robert Bosch GmbH, Stuttgart, 1991.
2. *TMS470R1x Controller Area Network Reference Guide (SPNU197)*, Texas Instruments.

## Appendix A Derivation of Equations

The derivation of the worst case accumulated error is shown (just for completeness).

The triangular-modulated waveform has the following form:



**Figure A-1. FPLL Parameters and Slope Calculation**

The triangular-modulation waveform is piecewise-continuous with constant slope within each segment of the waveform.

$$f_{SYS} = \begin{cases} f_{nom} + \Delta f + \frac{\Delta f}{T_{mod}/4}t & -\frac{T}{2} \leq t \leq 0 \\ f_{nom} + \Delta f - \frac{\Delta f}{T_{mod}/4}t & 0 \leq t \leq \frac{T}{2} \end{cases}$$

$$f_{SYS} = \begin{cases} f_{nom} + \Delta f + 4\Delta f(f_{fm}t) & -T/2 \leq t \leq 0 \\ f_{nom} + \Delta f - 4\Delta f(f_{fm}t) & 0 \leq t \leq T/2 \end{cases}$$

$$f_{SYS} = \begin{cases} f_{nom} \left( 1 + \frac{\Delta f}{f_{nom}} (1 + 4f_{fm}t) \right) & -\frac{T}{2} \leq t \leq 0 \\ f_{nom} \left( 1 + \frac{\Delta f}{f_{nom}} (1 - 4f_{fm}t) \right) & 0 \leq t \leq \frac{T}{2} \end{cases}$$

The equations have reduced to the form  $f_{SYS}=f_{nom}(1+\delta)$  where  $\delta$  is the frequency “error.”

As shown in the graph, the worst case frequency offset occurs when sampled about the extremes of the waveform; that is, the accumulated error is maximal when sampled symmetrically around the point  $[f_{nom} + \Delta f]$  or  $[f_{nom} - \Delta f]$ . Due to the piecewise nature of the waveform, the integral is also handled piecewise.

$$\text{Accumulated Phase Error} = \Delta\varphi = \int_{-\frac{T}{2}}^0 \frac{\Delta f}{f_{nom}} (1 + 4f_{fm}t)dt + \int_0^{\frac{T}{2}} \frac{\Delta f}{f_{nom}} (1 - 4f_{fm}t)dt$$

$$\Delta\varphi = \frac{\Delta f}{f_{\text{nom}}} \int_{-\frac{T}{2}}^0 dt + \frac{\Delta f}{f_{\text{nom}}} \int_{-\frac{T}{2}}^0 4f_{\text{fm}} t dt + \frac{\Delta f}{f_{\text{fm}}} \int_0^{\frac{T}{2}} dt - \frac{\Delta f}{f_{\text{fm}}} \int_0^{\frac{T}{2}} 4f_{\text{fm}} dt$$

$$\Delta\varphi = \frac{\Delta f}{f_{\text{nom}}} \left[ 0 - \left( -\frac{T}{2} \right) \right] + 4f_{\text{fm}} \frac{\Delta f}{f_{\text{nom}}} \left[ 0 - \left( \frac{(-T/2)^2}{2} \right) \right] + \frac{\Delta f}{f_{\text{nom}}} \left[ \frac{T}{2} - 0 \right] - 4f_{\text{fm}} \frac{\Delta f}{f_{\text{fm}}} \left[ \frac{(T/2)^2}{2} - 0 \right]$$

$$\Delta\varphi = \frac{T\Delta f}{2f_{\text{nom}}} - 4f_{\text{fm}} \frac{T^2}{8} \frac{\Delta f}{f_{\text{nom}}} + \frac{T\Delta f}{2f_{\text{nom}}} - 4f_{\text{fm}} \frac{T^2}{8} \frac{\Delta f}{f_{\text{nom}}}$$

$$\Delta\varphi = \frac{T\Delta f}{f_{\text{nom}}} - 4f_{\text{fm}} \frac{2T^2}{8} \frac{\Delta f}{f_{\text{nom}}}$$

$$\Delta\varphi = \frac{\Delta f}{f_{\text{nom}}} (T - f_{\text{fm}} T^2)$$

$$T = N * t_{\text{NominalBitTiming}} \quad 1 \leq N \leq 13,$$

where N is the number of consecutive CAN bits without resynchronization.

As the sampling period increases, the accumulated error tends to get larger. However, after the sampling period becomes greater than ½ the modulation period, the accumulated error decreases. The sampling time is a function of both:

1. the CAN bit timing, and
2. the number of successive bits without resynchronization.

The CAN protocol requires (at least) every 10<sup>th</sup> bit to resynchronize. Additionally, each CAN bit must be capable of resynchronizing (if a resynchronizing pulse were present). A second CAN condition requires that the CAN node correctly sample an error frame. The error frame could occur after the 13<sup>th</sup> CAN bit. That is, for  $N \leq 13$ , the accumulated error must be small enough that resynchronization is possible. Thus, solving for the worst case accumulated error as a function of N allows us to remove a variable from the equation.

The worst case accumulated error as a function of N is a typical maxima problem – solved by finding the 0 of the derivative with respect to N.

$$\Delta\varphi = \frac{\Delta f}{f_{\text{nom}}} (T - f_{\text{fm}} T^2)$$

$$\Delta\varphi = \frac{\Delta f}{f_{\text{nom}}} (N * t_{\text{NominalBitTiming}} - f_{\text{fm}} (N * t_{\text{NominalBitTiming}})^2)$$

$$\frac{d(\Delta\varphi)}{dN} = \frac{\Delta f}{f_{\text{nom}}} (t_{\text{NominalBitTiming}} - 2Nf_{\text{fm}} t_{\text{NominalBitTiming}}^2)$$

$$\frac{d(\Delta\varphi)}{dN} = 0 \quad \text{Find the worst case N}$$

$$\frac{\Delta f}{f_{\text{nom}}} (t_{\text{NominalBitTiming}} - 2Nf_{\text{fm}} t_{\text{NominalBitTiming}}^2) = 0$$

$$N_{\text{worstcase}} = \frac{1}{2f_{\text{fm}} t_{\text{NominalBitTiming}}} = \frac{1}{2} \frac{T_{\text{MOD}}}{t_{\text{NominalBitTiming}}}$$

Given the worst case N, solve the accumulated error:

$$\Delta\varphi = \frac{\Delta f}{f_{\text{nom}}} (N * t_{\text{NominalBitTiming}} - f_{\text{fm}} (N * t_{\text{NominalBitTiming}})^2)$$

$$N_{\text{worstcase}} = \frac{1}{2f_{\text{fm}} t_{\text{NominalBitTiming}}}$$

$$\Delta\varphi = \frac{\Delta f}{f_{\text{nom}}} \left( \frac{1}{2f_{\text{fm}} t_{\text{NominalBitTiming}}} * t_{\text{NominalBitTiming}} - f_{\text{fm}} \left( \frac{1}{2f_{\text{fm}} t_{\text{NominalBitTiming}}} * t_{\text{NominalBitTiming}} \right)^2 \right)$$

$$\Delta\varphi = \frac{\Delta f}{f_{\text{nom}}} \left( \frac{1}{2f_{\text{fm}}} - f_{\text{fm}} \left( \frac{1}{2f_{\text{fm}}} \right)^2 \right)$$

$$\Delta\varphi = \frac{\Delta f}{f_{\text{nom}}} \left( \frac{1}{4f_{\text{fm}}} \right)$$

As a limiting case:

$$\text{if } f_{\text{fm}} \leq \frac{1}{26t_{\text{NominalBitTiming}}}, \quad \text{use } N = 13.4$$

The meaning of this restriction is that the error cannot accumulate beyond the 13<sup>th</sup> CAN bit due to the CAN protocol. Thus, the maximum number of consecutive CAN bits for which resynchronization must be possible is 13.

<sup>4</sup> In the case of an error frame, the 13th frame needs to be correctly sampled. CAN protocol shows this relation (for oscillator tolerance as:

$$(2 \times \delta f)(13t_{\text{NominalBitTiming}} - \text{Phase\_Seg2}) < \min(\text{Phase\_Seg1}, \text{Phase\_Seg2})$$

$$(\delta f)(13t_{\text{NominalBitTiming}} - \text{Phase\_Seg2}) < \frac{\min(\text{Phase\_Seg1}, \text{Phase\_Seg2})}{2}$$

That is, the accumulated error through a little less than 13 nominal bit times must be less than 1/2 the minimum of PS1 and PS2. However, the synchronization jump width is constrained such that it can never be larger than min(PS1, PS2). Thus, a more restrictive condition is:

(note continues on the following page)



4(continued)

$$\delta f \leq \frac{\min(\text{Phase\_seg1}, \text{Phase\_Seg2})}{2 * (13 * \text{bit\_time} - \text{Phase\_Seg2})}$$

$$\delta f \leq \frac{\text{SJW}}{20 * \text{bit\_time}}$$

$$\text{SJW} \leq \min(\text{Phase\_seg1}, \text{Phase\_Seg2})$$

$$\delta f \leq \frac{\text{SJW}}{2 * (13 * \text{bit\_time})} \leq \frac{\min(\text{Phase\_seg1}, \text{Phase\_Seg2})}{2 * (13 * \text{bit\_time})} \leq \frac{\min(\text{Phase\_seg1}, \text{Phase\_Seg2})}{2 * (13 * \text{bit\_time} - \text{Phase\_Seg2})}$$

$$\delta f \leq \frac{\text{SJW}}{26 * \text{bit\_time}} \leq \frac{\text{SJW}}{20 * \text{bit\_time}}$$

Thus, the more restrictive requirement,

$$\delta f \leq \frac{\text{SJW}}{26 * \text{bit\_time}}$$

incorporates both restrictions regarding the oscillator tolerance.

That is,

$$13 * \text{bit\_time} * \delta f \leq \frac{\text{SJW}}{2}$$

The accumulated error through 13 (or less) nominal bit times must be less than ½ SJW.

This derivation is valid for  $\delta f$  which is static. The generalization of this derivation is stated – that the accumulated error through 13 or fewer CAN bits (whether from a static or variable source) must be less than ½ SJW<sub>smallest Rx</sub>.

## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

<b>Products</b>		<b>Applications</b>	
Amplifiers	<a href="http://amplifier.ti.com">amplifier.ti.com</a>	Audio	<a href="http://www.ti.com/audio">www.ti.com/audio</a>
Data Converters	<a href="http://dataconverter.ti.com">dataconverter.ti.com</a>	Automotive	<a href="http://www.ti.com/automotive">www.ti.com/automotive</a>
DSP	<a href="http://dsp.ti.com">dsp.ti.com</a>	Broadband	<a href="http://www.ti.com/broadband">www.ti.com/broadband</a>
Interface	<a href="http://interface.ti.com">interface.ti.com</a>	Digital Control	<a href="http://www.ti.com/digitalcontrol">www.ti.com/digitalcontrol</a>
Logic	<a href="http://logic.ti.com">logic.ti.com</a>	Military	<a href="http://www.ti.com/military">www.ti.com/military</a>
Power Mgmt	<a href="http://power.ti.com">power.ti.com</a>	Optical Networking	<a href="http://www.ti.com/opticalnetwork">www.ti.com/opticalnetwork</a>
Microcontrollers	<a href="http://microcontroller.ti.com">microcontroller.ti.com</a>	Security	<a href="http://www.ti.com/security">www.ti.com/security</a>
		Telephony	<a href="http://www.ti.com/telephony">www.ti.com/telephony</a>
		Video & Imaging	<a href="http://www.ti.com/video">www.ti.com/video</a>
		Wireless	<a href="http://www.ti.com/wireless">www.ti.com/wireless</a>

Mailing Address: Texas Instruments  
Post Office Box 655303 Dallas, Texas 75265