

**Preliminary Application Note**

**LIN Slave Communications  
with 78K0/Kx2 and Fx2**

**Auto Baud Rate Capabilities**

---

## Legal Notes

- **The information contained in this document is being issued in advance of the production cycle for the product. The parameters for the product may change before final production or NEC Electronics Corporation, at its own discretion, may withdraw the product prior to its production.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special", and "Specific". The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics products before using it in a particular application.
  - "Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots.
  - "Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support).
  - "Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

---

The quality grade of NEC Electronics products is "Standard" unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact an NEC Electronics sales representative in advance to determine NEC Electronics' willingness to support a given application.

(Note)

(1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.

(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

---

## Regional Information

Some information contained in this document may vary from country to country. Before using any NEC product in your application, please contact the NEC office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

### NEC Electronics Corporation

1753, Shimonumabe, Nakahara-ku,  
Kawasaki, Kanagawa 211-8668, Japan  
Tel: 044 4355111  
<http://www.necel.com/>

#### [America]

**NEC Electronics America, Inc.**  
2880 Scott Blvd.  
Santa Clara, CA 95050-2554,  
U.S.A.  
Tel: 408 5886000  
<http://www.am.necel.com/>

#### [Europe]

**NEC Electronics (Europe) GmbH**  
Arcadiastrasse 10  
40472 Düsseldorf, Germany  
Tel: 0211 65030  
<http://www.eu.necel.com/>

#### United Kingdom Branch

Cygnus House, Sunrise Parkway  
Linford Wood, Milton Keynes  
MK14 6NP, U.K.  
Tel: 01908 691133

#### Succursale Française

9, rue Paul Dautier, B.P. 52  
78142 Velizy-Villacoublay Cédex  
France  
Tel: 01 30675800

#### Sucursal en España

Juan Esplandiú, 15  
28007 Madrid, Spain  
Tel: 091 5042787

#### Tyskland Filial

Täby Centrum  
Entrance S (7th floor)  
18322 Täby, Sweden  
Tel: 08 6387200

#### Filiale Italiana

Via Fabio Filzi, 25/A  
20124 Milano, Italy  
Tel: 02 667541

#### Branch The Netherlands

Steijgerweg 6  
5616 HS Eindhoven,  
The Netherlands  
Tel: 040 2654010

#### [Asia & Oceania]

**NEC Electronics (China) Co., Ltd**  
7th Floor, Quantum Plaza, No. 27  
ZhiChunLu Haidian District,  
Beijing 100083, P.R.China  
Tel: 010 82351155  
<http://www.cn.necel.com/>

#### NEC Electronics Shanghai Ltd.

Room 2511-2512, Bank of China  
Tower,  
200 Yincheng Road Central,  
Pudong New Area,  
Shanghai 200120, P.R. China  
Tel: 021 58885400  
<http://www.cn.necel.com/>

#### NEC Electronics Hong Kong Ltd.

12/F., Cityplaza 4,  
12 Taikoo Wan Road, Hong Kong  
Tel: 2886 9318  
<http://www.hk.necel.com/>

#### NEC Electronics Taiwan Ltd.

7F, No. 363 Fu Shing North Road  
Taipei, Taiwan, R.O.C.  
Tel: 02 27192377

#### NEC Electronics Singapore Pte. Ltd.

238A Thomson Road,  
#12-08 Novena Square,  
Singapore 307684  
Tel: 6253 8311  
<http://www.sg.necel.com/>

#### NEC Electronics Korea Ltd.

11F., Samik Lavied'or Bldg., 720-2,  
Yeoksam-Dong, Kangnam-Ku, Seoul,  
135-080, Korea Tel: 02-558-3737  
<http://www.kr.necel.com/>

## Table of Contents

<b>Chapter 1</b>	<b>Introduction</b>	6
<b>Chapter 2</b>	<b>Bit Rate Generator Precision</b>	7
<b>Chapter 3</b>	<b>Bit Rate Detection Capability</b>	8
<b>Chapter 4</b>	<b>Bit Rate Adjustment Capabilities</b>	10
<b>Chapter 5</b>	<b>Tests</b>	11
<b>Chapter 6</b>	<b>Results</b>	12
<b>Chapter 7</b>	<b>Conclusion</b>	14
<b>Chapter 8</b>	<b>Attachments</b>	18

# Chapter 1 Introduction

According to the LIN specification 2.1, LIN slave nodes must be able to detect the baud rate of the master node and to adjust their baud rate with a maximum error of 1.5%. This document shows that such a feature can be achieved on 78K0/Kx2 and Fx2 devices with minimum software effort even when the internal oscillator (8MHz +/-5%) is used. The first part studies the precision of the baud rate generator and the precision of the timer which is used to measure the master's baud rate. The second part studies how to deduce the settings of the UART from the measured baud rate. Finally, some simulation results are shown to validate the approach. In addition to the simulation software, a real implementation of the synchronisation mechanism is provided as attached files.

78K0/Kx2 and Fx2 devices are fitted with the UART6 macro. This document discusses the capabilities of this macro only. The UART0 of 78K0/Kx2 devices is not considered here. The reader should read the LIN protocol specification, UART6 and Timer TM00 chapters of a 78K0/Kx2 datasheet before reading this document (If you have a 78K0/Fx2 datasheet, you will find different signal names and interrupt vectors but the principles are the same).

The LIN protocol support speed up to 20kbps. Since the master node must have a maximum deviation of 0.5%, the maximum bit rate considered in this document is 20100 bit per second (bps). The minimum speed considered in this document is 2000bps. Since the UART6 outputs binary symbols, baud rate and bit rate are equal, the term bit rate will be used throughout this document.

## Chapter 2 Bit Rate Generator Precision

Worst case: Fprs=8MHz-5%, target bit rate is 20100bps.

In those conditions, the maximum difference between two selectable bit rates is 105.8 bps (0.53% of 20100bps):

CKSR6=0	
BRGC6 value	actual bit rate
189	20105.8
190	20000.0
191	19895.3

This maximum difference happens at 20100bps and decrease faster than the bit rate (0.4% when target bit rate is 16kbps).

Fprs=20MHz-1%, target bit rate is 20100bps

In those conditions, the maximum difference between two selectable bit rates is 81.5 bps (0.4% of 20100bps):

CKSR6=1	
BRGC6 value	actual bit rate
246	20122
247	20040.5
248	19959.7

Conclusion: UART6 baud rate generator is able to achieve a target bit rate with 0.26% accuracy with the internal 8MHz oscillator as source clock.

$$(accuracy = \frac{maximum\_step}{2} * \frac{100}{target\_bit\_rate})$$

## Chapter 3 Bit Rate Detection Capability

The Rxd6 pin can be internally connected to the T1000 input of TM00 in order to measure the bit rate used by the master. Depending on the measurement strategy, the time to measure ranges from 1Tbit up to 8Tbit.

In the typical case where TM00 use the internal 8MHz oscillator as clock source, durations up to 8192us can be measured with 0.125us accuracy (all figures +/-5%).

If the bit rate is in the range 2 to 20kbp, the duration of the first 8bit of the sync field can be done with 0.125us resolution (8 \* Tbit @ 2kbps is 4000us).

In the case of 1Tbit measurement (worst case), the precision is 026%:

$$\text{min\_time\_to\_measure} = 1 \text{ Tbit} @ 20.1 \text{ kbps} = \frac{1}{20100} = 49 \text{ us}$$

$$\text{precision} = \frac{(\text{resolution} + \text{maximum\_resolution\_deviation}) * 100}{\text{min\_time\_to\_measure}} = \frac{(0.125 + 5\%) * 100}{49} = 0.26$$

In the case of 8Tbit measurement, the minimum time to measure is 398us and the precision is 0.03%.

This measurement can be achieved with the following procedure:

- Detect the sync break field
- Start the TM00 timer in the free running mode and enable INTTM010 interrupt on falling edges.
- During the first pass in INTTM010, store CR010 value in the StartTime variable (must be an unsigned 16bit variable).
- During the eighth pass in the INTTM010 interrupt, stop the TM00 timer.
- Compute the duration of the 8 first bit: Duration=CR010-StartTime (The result will be valid even if a timer overflow occurs as long as the duration last less than 8192us).

The measured duration must be divided by 8 to get the duration of 1Tbit:

From a precision point of view, the worst case would happen with the maximum bit rate 20100bps and the minimum Fprs frequency 8MHz-5%: In this conditions, the duration of 8Tbit are measured as 3024 ticks.

As this value must be divided by 8, we need to consider the range from 3024 to 3024+7 ticks:

Duration in ticks	floating point division	int division	error	rounded int division	error
3024	378	378	0	378	0
3025	378.13	378	0.03	378	0.03
3026	378.25	378	0.07	378	0.07
3027	378.38	378	0.1	378	0.1

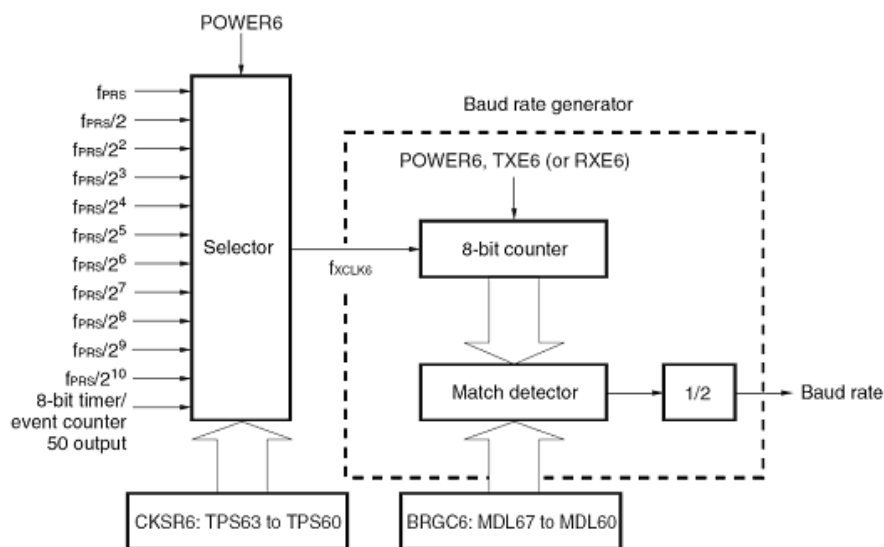
Duration in ticks	floating point division	int division	error	rounded int division	error
3028	378.5	378	0.13	379	0.13
3029	378.63	378	0.17	379	0.1
3030	378.75	378	0.2	379	0.07
3031	378.88	378	0.23	379	0.03

The maximum error of this operation is 0.23% in case of simple shifting and can be reduced to 0.13% with rounding. The overall errors are respectively 0.26% and 0.16%.

In conclusion, 1 Tbit measurement is quicker and has the same precision than the 8 Tbit measurements without rounding.

## Chapter 4 Bit Rate Adjustment Capabilities

Several methods can be used to compute UART6 setting from the duration in ticks of 1Tbit. All methods considered here share the same principle, directly deduced from UART6 architecture:



- Remarks**
- CKSR6 select roughly the range of the bit rate and BRGC6 is used to select precisely the desired bit rate.
  - The bigger Tbit is, the bigger CKSR6 needs to be.  
The bigger Tbit is, the bigger BRGC6 needs to be.
  - To generate a constant bit rate, if CKSR6 is incremented, BRGC6 must be divided by 2.

$$\text{Equation: } 1\text{Tbit} = \frac{2 * \text{BRGC6} * (2^{\text{CKSR6}})}{\text{Fprs}}$$

We have one non linear equation and two unknown variables so mathematics will not help us a lot (Exact values of 1Tbit and Fprs are also unknown separately but 1Tbit\*Fprs, or a multiple of it is measured by the timer TM0). Anyway we can still rewrite this equation as follows:

$$\text{CKSR6} = \log\left(\frac{1\text{Tbit} * \text{Fprs}}{2 * \text{BRGC6}}, 2\right)$$

$$\text{BRGC6} = \frac{1\text{Tbit} * \text{Fprs}}{2^{\text{CKSR6}+1}}$$

Since there are only 11 possible values for CKSR6 we can try each one starting from 0, compute BRGC6 using the equation and stop the process when the result fit in 8bits.

# Chapter 5 Tests

As it is rather difficult to mathematically determine the maximum error done by this algorithm, a software simulation has been used. Six methods have been simulated:

## **1TBit**

- Measure  $1TBit * Fprs$  and store it in a 16bit variable “ticks”.
- Shift to the left ticks until it is below 256 and increment CKSR6 for each shift.
- If CKSR6 is zero, save the lsb of ticks in a variable “last\_bit” and shift left ticks, else decrement CKSR6.
- Set BRGC6 with ticks+last\_bit.

This method is implemented in the attached demo project.

## **8TBit**

Same as “1TBit” but the measured time is  $8 * 1TBit * Fprs$ . Ticks is divided by 8 before entering in the shifting loop.

## **1TBitBRGC6Rounding**

Same as “1TBit” but BRGC6 is set with ticks + the value of the last bit which has been shifted out. If the last shift is viewed as a “divide by 2” operation, this procedure rounds the result of this division.

## **8TBitBRGC6Rounding**

Combination of “1TBitBRGC6Rounding” and “8Tbit”.

## **8TBitRoundAll**

Same as “8TBitBRGC6Rounding” but the division by 8 of “ticks” is split into a division by 4 and then a division by 2 with rounding.

## **Best**

This method doesn't use the principle explained before. It uses brute force computation to find the optimal setting of the UART. This method is for reference only because it can't be implemented on a real device since it needs to know precisely Fprs.

The sources of the simulation software are provided in the UART6LinSync.cpp file. This software is a console application which can be build on Win32 and Linux using the GCC compiler.

# Chapter 6 Results

The test conditions cover the full range of Fprs deviation and the full range of LIN bit rates:

MinFprs (Hz)	MaxFprs (Hz)	MinMasterBitrate (bit per second)	MaxMasterBitrate (bit per second)
7520000	8480000	1840.75	20100

The simulation step for the master bit rate is 1. This step is small enough to covers all possible values in the measurement timer (TM00).

The next page summarizes the results of each method:

It indicates the most critical master bit rate, the computed UART6 settings, the resulting bit rate and its deviation from the master bit rate in %.

<b>Best@MaxFprs</b>			
		MasterBitrate	
		8249	
BRGC6	CKSR6	ActualBitrate	Error
129	2	8217.05	0.387268

<b>Best@MinFprs</b>			
		MasterBitrate	
		14631	
BRGC6	CKSR6	ActualBitrate	Error
128	1	14687.5	0.386166

<b>1TBitBRGC6Rounding@MaxFprs</b>			
		MasterBitrate	
		16563	
BRGC6	CKSR6	ActualBitrate	Error
255	0	16627.5	0.389126

<b>1TBitBRGC6Rounding@MinFprs</b>			
		MasterBitrate	
		3672	
BRGC6	CKSR6	ActualBitrate	Error
255	2	3686.27	0.388739

<b>8TBitBRGC6Rounding@MaxFprs</b>			
		MasterBitrate	
		16563	
BRGC6	CKSR6	ActualBitrate	Error
255	0	16627.5	0.389126

<b>8TBitBRGC6Rounding@MinFprs</b>			
		MasterBitrate	
		3672	
BRGC6	CKSR6	ActualBitrate	Error
255	2	3686.27	0.388739

<b>8TBitRoundAll@MaxFprs</b>			
		MasterBitrate	
		16514	
BRGC6	CKSR6	ActualBitrate	Error
129	1	16434.1	-0.48378

<b>8TBitRoundAll@MinFprs</b>			
		MasterBitrate	
		14644	
BRGC6	CKSR6	ActualBitrate	Error
129	1	14573.6	0.480447

<b>8TBit@MaxFprs</b>			
		MasterBitrate	
		2039	
BRGC6	CKSR6	ActualBitrate	Error
129	4	2054.26	0.748581

<b>8TBit@MinFprs</b>			
		MasterBitrate	
		3645	
BRGC6	CKSR6	ActualBitrate	Error
128	3	3671.88	0.737311

<b>1TBit@MaxFprs</b>			
		MasterBitrate	
		16435	
BRGC6	CKSR6	ActualBitrate	Error
128	1	16562.5	0.775783

<b>1TBit@MinFprs</b>			
		MasterBitrate	
		7287	
BRGC6	CKSR6	ActualBitrate	Error
128	2	7343.75	0.778784

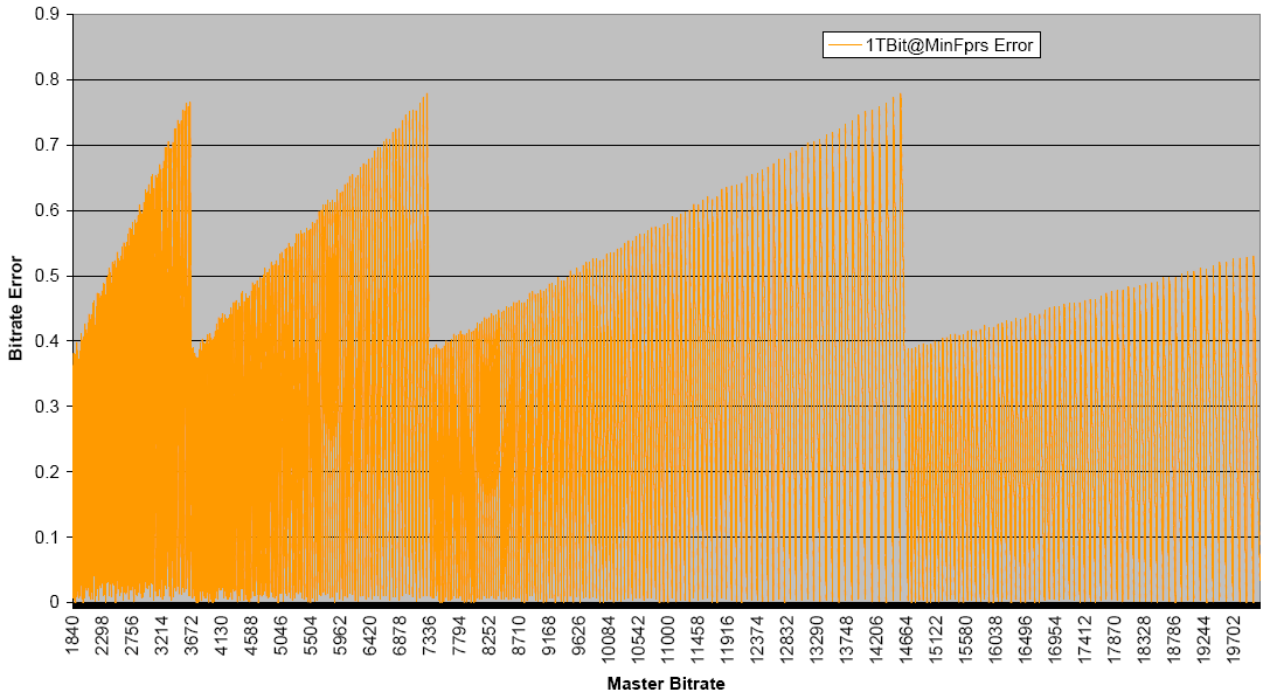
## Chapter 7 Conclusion

The UART6 macro is able to fulfill LIN protocol requirements even when the internal high speed oscillator of 78K0/Kx2 or Fx2 devices is used as clock source.

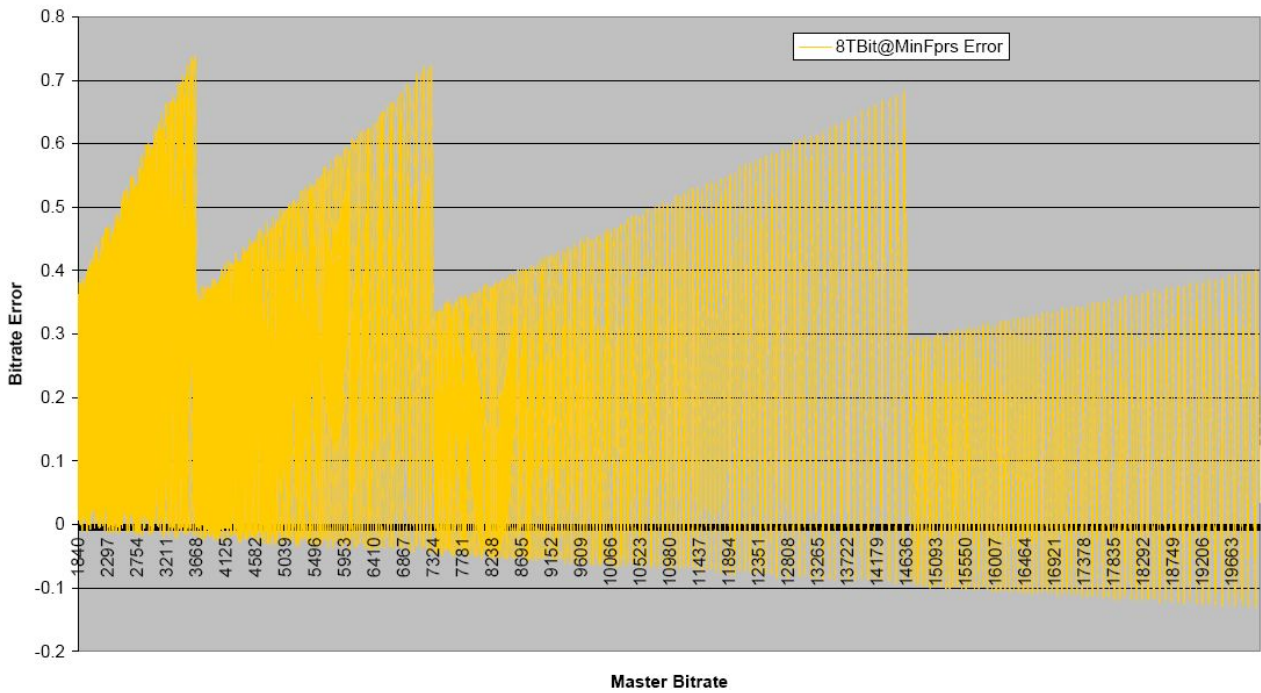
The simulation results show:

- All methods fulfill LIN protocol requirements with comfortable margins.
- As expected the best method is “Best”. Then it is interesting to see that the best “real” methods are “1TBitBRGC6Rounding” and “8TBitBRGC6Rounding”.
- As expected, the worst cases are always with the minimum Fprs.
- The “1TBit” method achieves always a positive error.
- The LIN protocol recommends to measure 8Tbit and to round the result, this is what “8TbitRoundAll” does. However, the results show that other method have a better precision.
- All methods have 4 critical regions of bit rates around 1840, 3697, 7394, 14787 bit per second.

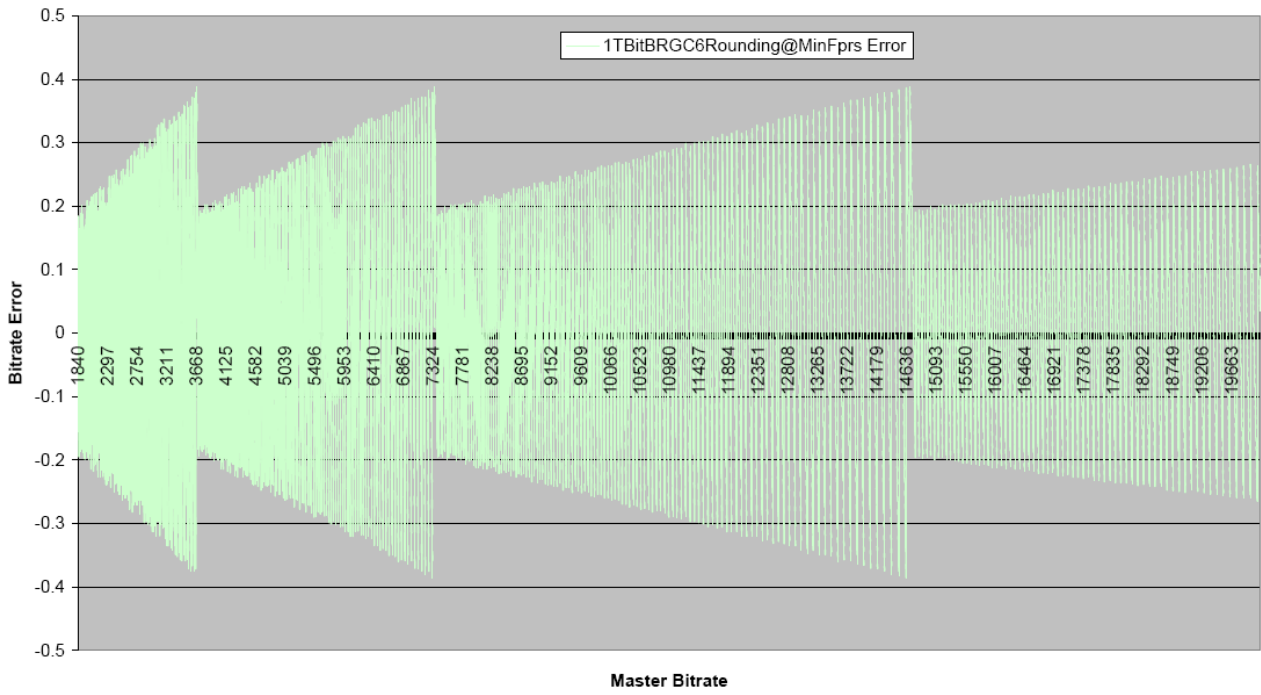
1TBit@MinFprs Error



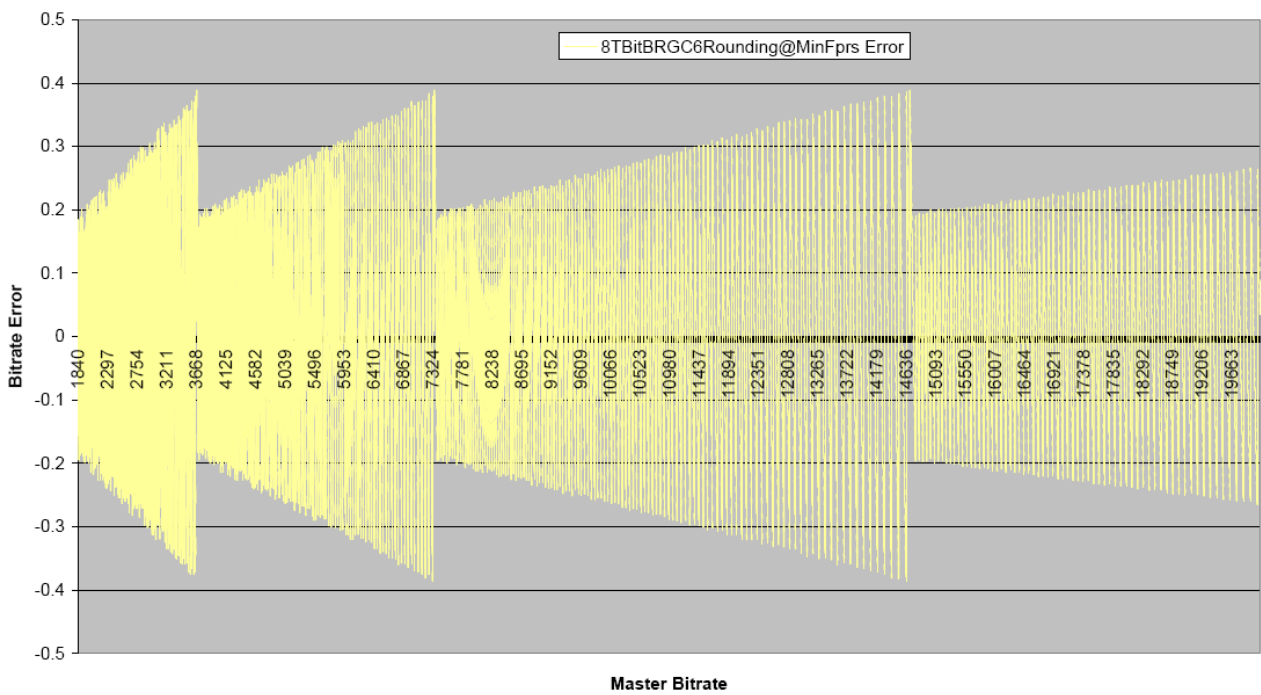
8TBit@MinFprs Error

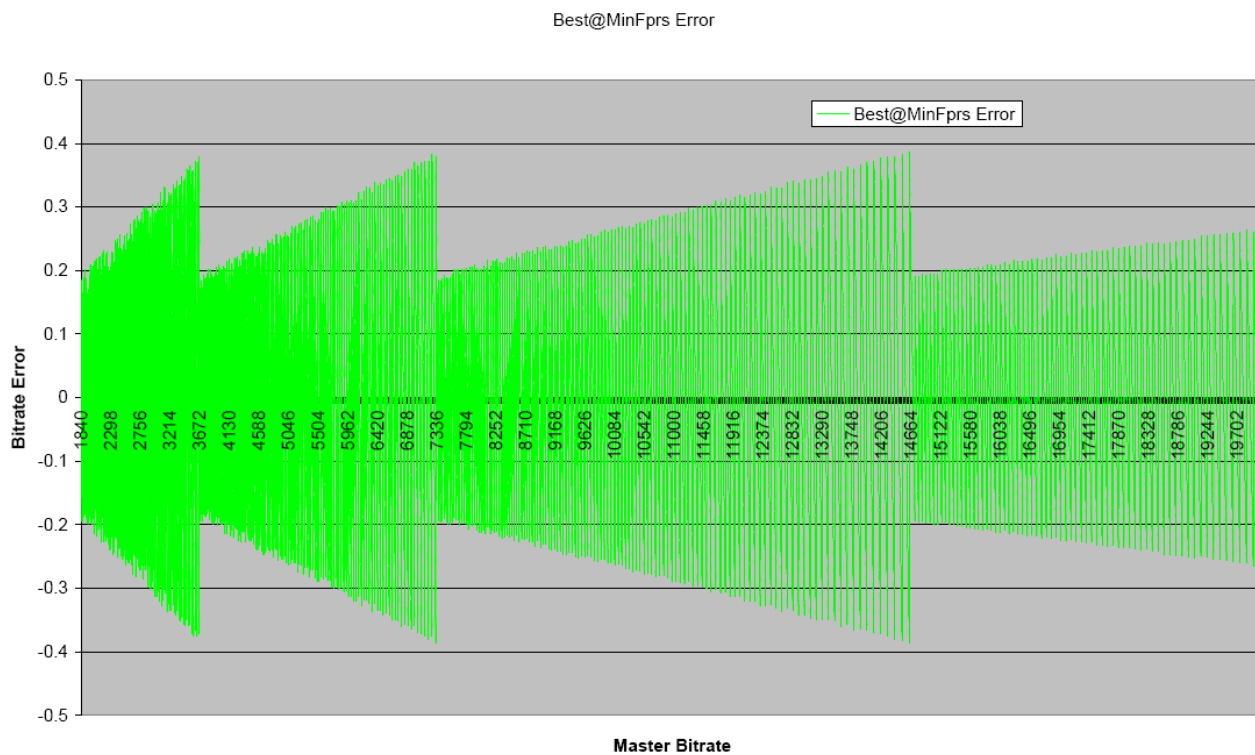
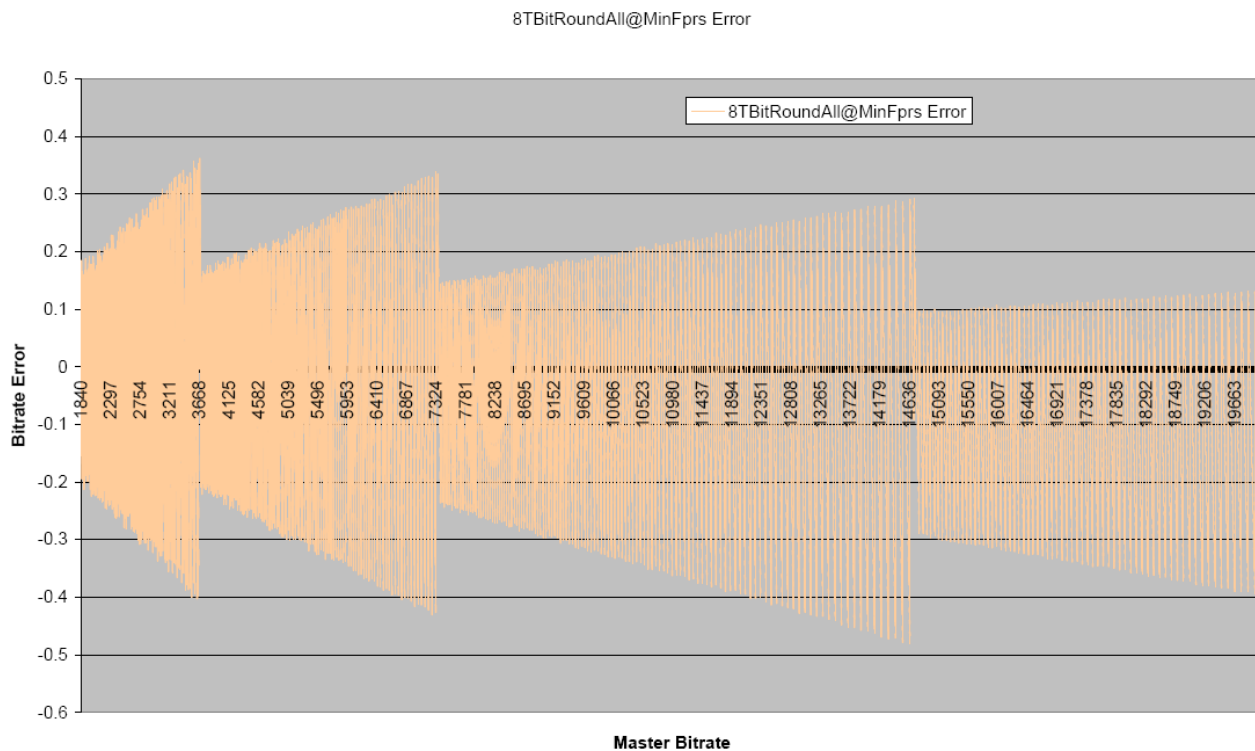


1TBitBRGC6Rounding@MinFprs Error



8TBitBRGC6Rounding@MinFprs Error





## Chapter 8 Attachments

This document contains two file attachments, *Demo* and *Simulation*. Use the *Attachments* tab to see them.

Due to PDF-internal security restrictions you cannot open them directly. Please perform the following actions:

- Mark the file(s) and save them to a local drive
- Rename the file(s) to "Demo.zip" resp. "Simulation.zip"
- Unzip the file(s)

The demo project runs on the 78K0/Fx2 CANit! demo kit. As the TK78 debugger driver needs to know which COM port is associated with the demo kit, please follow the procedure described in chapter 12 "IAR sample session" of the CANit documentation (document reference U18596EE).

