Application Report
SWRA566–June 2017

CC13x0 Low Data Rate Operation

Marius Moe, Siri Johnsrud, and Sverre Hellan

ABSTRACT

This application report describes a CC13x0 SimpleLink™ Sub-1 GHz Ultra-Low-Power Wireless Microcontroller RF Core bit repetition patch. The patch allows lower data rates than what is possible with the normal genfsk patch. The patch also increases the maximum deviation for a given data-rate setting, and improves the frequency offset tolerance for low data rates.

Contents

List of Figures

List of Tables

1 Overview

This application report describes a CC13x0 RF Core bit repetition patch (brepeat patch). The purpose of this patch is to allow for lower data rates and (or) larger deviation and frequency offset in RX than what is possible with the normal genfsk patch. This is done by repeating the data bits over the air two, three, or four times and then remove this repetition before the data is written to the data entry by the radio CPU. The patch also allows lower data rates in TX. The normal genfsk patch limits the data rate to 1 ksps.

In the CC13x0 device, there is a limit to the lowest data rate allowed for an RX filter bandwidth (rxBW) as well as a limit on the maximum deviation for a data rate (DR) setting.

1.1 Impact on Data Rate

Table 1 lists the min/max data rates for some selected rxBW with and without the patch at 868 MHz and using divider 5.

Table 1. Minimum and Maximum Data Rates vs rxBW @868 MHz, Divider 5

	genfsk Patch		brepeat Patch (Repetition 4×)	
rxBW [kHz]	Minimum DR [kbps]	Maximum DR [kbps]	Minimum DR [kbps]	Maximum DR [kbps]
39	1.8	29	0.5	7.4
49	1.8	29	0.5	7.4
78	3.6	58	0.9	14.7
98	3.6	58	0.9	14.7
118	3.6	58	0.9	14.7
155	7.3	117	1.8	29.4
236	7.3	117	1.8	29.4
310	14.7	235	3.7	58.9
471	14.7	235	3.7	58.9
622	29.4	470	7.4	117.7

1.2 Impact on Minimum and Maximum Deviation

The maximum supported deviation in RX is limited by the sampling rate of the data symbols. The CC13x0 device can have 4×, 8×, or 16× oversampling that is controlled by the PDIF decimation setting (*pdifdecim*) (see Section 2.6). The maximum deviation is given by Equation 1.

$$\text{Max deviation [Hz]} = \text{Symbol Rate [sps]} \times 2^{\text{pdifdecim}} \quad (1)$$

Pdifdecim is an internally calculated parameter controlling the oversampling of the data; this parameter can have the values 0, 1, and 2 corresponding to 4×, 8×, and 16× oversampling, respectively. In general, for low data rates relative to the minimum and maximum for a given rxBW, *pdifdecim* will be 2. For high data rates, *pdifdecim* will be 0.

If the deviation plus frequency offset is larger than symbol_rate × $2^{\text{pdifdecim}}$, the symbols will experience clipping in the receiver. It is possible to demodulate the signal even with some clipping, but performance will start to degrade. The speed of degradation varies from case to case.

The bit repetition patch increases the oversampling of the symbols. Assume the maximum deviation plus frequency offset is 5 kHz for a given setting using the genfsk patch. If the bit repetition patch is used with a repetition factor of 4×, the maximum deviation can be up to 4 × 5 kHz = 20 kHz.

NOTE: The channel filter sampling rate sets an upper limit on the symbol oversampling and the following rule applies:

Channel Filter Sampling Rate ≥ Symbol Oversampling Rate = Bit Repetition Factor × DR × $2^{(2 + \text{pdifdecim})}$

Use the Excel spreadsheet calculator explained in Section 1.4 to calculate channel filter sampling rate and symbol oversampling rate.

1.3 Impact on Frequency Offset Tolerance

Section 1.2 has an example where the maximum deviation is increased from 5 kHz to 20 kHz by using 4× bit repetition. This maximum limit assumes zero frequency offset. Room for both the deviation and the frequency offset within the limits calculated in Section 1.2 is required.

The bit repetition patch will increase the frequency offset tolerance in many cases. A higher frequency offset value can be represented without clipping due to the larger sampling rates that can be achieved. If the deviation is still 5 kHz, as in the previous example, the frequency offset is increased by ±15 kHz when using 4× bit repetition.

1.4 Calculator

An Excel spreadsheet can be used to check the validity of the PHY settings [1]. Figure 1 shows a picture of the spreadsheet. Input is entered in the yellow fields. The channel filter sample rate versus symbol oversampling rate, rate configuration, and deviation configuration all must be green to support the entered PHY settings.

Repetition factor = 1 corresponds to genfsk (that is, no bit repetition). The brepeat patch supports repetition factors 2, 3, and 4.

Divider	5	
RF frequency	868	MHz
DataRate:	4800	bps
Deviation:	25400	Hz
RXBW	110	kHz
Repetition factor:	3	(1 for GENFSK mode)
OUTPUT:		
bdec1+bdec2	5	
actual RxBW	98	kHz
Channel filter sample rate	235460	Hz
min_rate possible	1226	Hz
max_rate possible	19622	Hz
max_pdif possible	2	
Symbol oversampling rate with max pdif	230400	Hz
max_deviation for given data rate	57600	
Channel filter sample rate > Symbol oversampling rate	valid	
Rate configuration	valid	
Deviation configuration	valid	

Figure 1. Excel Spreadsheet Calculator

1.5 Example

Example test case from the CC1101 device: 4.8 kbaud, ±25.4-kHz deviation, 100-kHz rxBW.

For the CC13x0 device, the closest rxBW setting is 98 kHz @868 MHz. This setting will give a channel filter sampling rate of 235 ksamples/s.

With the standard genfsk patch *pdifdecim* = 2 because it is close to the lowest data rate supported for this rxBW. The oversampling is thus 16× the data symbols. Using Equation 1, the maximum deviation becomes 4.8 kHz × 2^2 = 19.2 kHz. This test case will therefore experience clipping in the standard genfsk mode.

With the bit repetition patch, the bits can be repeated three times, and the programmed symbol rate will then be 3 × 4.8 ksps = 14.4 ksps. Setting *pdifdecim* = 2 yields a symbol oversampling of 16× (and data oversampling of 3 × 16 = 48x). This gives a symbol oversampling rate of 14.4 ksps × 16 = 230.4 kHz, which is below the channel filter sample rate of 235 ksamples/s. The maximum deviation supported will then be 14.4 kHz × 2^2 = 57.6 kHz (see Figure 1). Consequently, with bit repetition we are able to receive this format without clipping and also have room for frequency offset.

In this particular example, the bit repetition factor cannot be 4 because this gives a symbol oversampling rate greater than the channel filter sampling rate.

2 Bit Repetition Mode Configuration

The look and feel of this mode is just like the standard genfsk mode, but there are a few exceptions. This section describes how to configure the modem for repeated bit operation.

2.1 Repetition Factor

The values for the number of bit repetitions are set at register address 0x5178. The supported values are 2, 3, 4 which gives 2, 3, 4 times bit repetition, respectively.

Example: Setting the bit repetition factor to 2.

HW_REG_OVERRIDE(0x5178, 0x0002); // Repetition factor = 2.

2.2 Data Rate

If Y-times repetition is programmed, the data rate also must be multiplied by Y. The deviation setting should not be changed.

Example: If the user wants to send a 10-kbps signal with deviation ±20 kHz and 4× bit repetition, the resulting data rate programmed must be 40 kbps, but the deviation must still be ±20 kHz.

2.3 Sync Word Programming

The sync word must be programmed manually when using the bit repetition patch.

2.3.1 Enable Manually Entered Sync Word

The user must prevent the RF Core from overwriting the programmed sync word; this is done by using the following override.

Example: (uint32_t)0x000C8103 // Bit3 in 0x000C prevents automatic programming of sync word.

2.3.2 Sync Word Length

The sync word length must be set by a hardware register override as well as through the proprietary radio setup command (CMD_PROP_RADIO_DIV_SETUP).

The sync word length is programmed at register address 0x5050, bits 15:8. The sync word length is to be set to the length before bit repetition (values 15 to 31 are supported in the patch, giving lengths of 16 to 32 bits).

Examples:

HW_REG_OVERRIDE(0x5050,0x**0F**7F) // 2-byte sync word.

HW_REG_OVERRIDE(0x5050,0x**1F**77) // 4-byte sync word.

In CMD_PROP_RADIO_DIV_SETUP, the sync word length is set through the nSwBits field.

2.3.3 Sync Word

This section provides two examples of how the CC13x0 device can be configured to be backward compatible with designs using the CC1101 device.

The sync word must be programmed at register addresses 0x510C, 0x5110, 0x5114, and 0x5118 (SYNC0, SYNC1, SYNC2, and SYNC3, respectively). The repetition must be done manually.

The patch supports a sync word length of 32 to 64 symbols after repetition. Some valid combinations follow:

- 16-bit sync word × 2 = 32 symbols
- 16-bit sync word × 4 = 64 symbols
- 32-bit sync word × 2 = 64 symbols
- 16-bit sync word × 3 = 48 symbols

The sync word is transmitted over the air in the following order:

1. SYNC2 sent LSB to MSB
2. SYNC3 sent LSB to MSB
3. SYNC0 sent LSB to MSB
4. SYNC1 sent LSB to MSB

Example 1:

4-BYTE SYNC WORD. 2× REPEAT							
0x1	0x9	0x3	0xD	0x01	0x9	0x3	0xD
1 1 0 0 0 0 0 0 1 1 0 0 0 0 1 1 1 1 1 1 0 0 0 0 1 1 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 1 1 0 0 0 0 1 1 1 1 1 1 0 0 0 0 1 1 0 0 1 1 1 1							
C0	C3	F0	CF	C0	C3	F0	CF
SYNC1 = 0xC0C3		SYNC0 = 0xF0CF		SYNC3 = 0xC0C3		SYNC2 = 0xF0CF	

Figure 2. Four-Byte Sync Word 0xD391D391 (Used by CC1101), 2× Bit Repetition

HW_REG_OVERRIDE(0x510C, 0xF0CF); // Part of sync word. SYNC0

HW_REG_OVERRIDE(0x5110, 0xC0C3); // Part of sync word. SYNC1

HW_REG_OVERRIDE(0x5114, 0xF0CF); // Part of sync word. SYNC2

HW_REG_OVERRIDE(0x5118, 0xC0C3); // Part of sync word. SYNC3

Example 2:

4-BYTE SYNC WORD. 2× REPEAT							
0x5	0x4	0x7	0x6	0x01	0x9	0x3	0xD
1 1 0 0 1 1 0 0 0 0 0 0 1 1 0 0 1 1 1 1 1 1 0 0 0 0 1 1 1 1 0 0 1 1 0 0 0 0 0 0 0 1 1 0 0 0 0 1 1 1 1 1 1 0 0 0 0 1 1 0 0 1 1 1 1							
CC	0C	FC	3C	C0	C3	F0	CF
SYNC1 = 0xCC0C		SYNC0 = 0xFC3C		SYNC3 = 0xC0C3		SYNC2 = 0xF0CF	

Figure 3. Four-Byte Sync Word 0xD3916745, 2× Bit Repetition

```
HW_REG_OVERRIDE(0x510C, 0xFC3C); // Part of sync word. SYNC0
HW_REG_OVERRIDE(0x5110, 0xCC0C); // Part of sync word. SYNC1
HW_REG_OVERRIDE(0x5114, 0xF0CF); // Part of sync word. SYNC2
HW_REG_OVERRIDE(0x5118, 0xC0C3); // Part of sync word. SYNC3
```

Example 3:

2-BYTE SYNC WORD 4× REPEAT							
0x1		0x9		0x3		0xD	
1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1							
F0	0	F0	0F	FF	0	F0	FF
SYNC1 = 0xF000		SYNC0 = 0xF00F		SYNC3 = 0xFF00		SYNC2 = 0xF0FF	

Figure 4. Two-Byte Sync Word 0xD391 (Used by CC1101), 4× Bit Repetition

```
HW_REG_OVERRIDE(0x510C, 0xF00F); // Part of sync word. SYNC0
HW_REG_OVERRIDE(0x5110, 0xF000); // Part of sync word. SYNC1
HW_REG_OVERRIDE(0x5114, 0xF0FF); // Part of sync word. SYNC2
HW_REG_OVERRIDE(0x5118, 0xFF00); // Part of sync word. SYNC3
```

2.4 Programming Preamble

The preamble pattern is programmed at register address 0x5058. The preamble must be repeated manually. Bits 3:0 at register address 0x5050 register controls the length of the preamble pattern, and bits 7:4 at register address 0x5050 controls how many times to repeat this pattern.

Example: Repetition factor = 2 and 0x55 as preamble

HW_REG_OVERRIDE(0x5050, 0x1F**77**); // Repeat an 8-bit pattern eight times.

HW_REG_OVERRIDE(0x5058, 0x00CC); // Preamble pattern to send

Example: Repetition factor = 4 and 0x55 preamble

HW_REG_OVERRIDE(0x5050, 0x0F**7F**); // Repeat a 16-bit pattern eight times.

HW_REG_OVERRIDE(0x5058, 0xF0F0); // Preamble pattern to send

2.5 Other Overrides

Sync word search uses a correlator. When using the brepeat patch the correlator threshold might have to be increased compared to the threshold used with the genfsk patch. For a 4-byte sync word with the genfsk patch, the threshold is 0x27. The threshold is programmed at register address 0x5104, bits 7:0.

Example: HW_REG_OVERRIDE(0x5104, 0x0030) // Adjust this value to optimize false sync / sensitivity.

The frequency offset tracking can perform worse in the presence of long strings of 1s or 0s during preamble and sync word. So if the number of repetitions is 3 or 4 the frequency offset tracking algorithm should be slowed down. This is achieved using the override that follows.

Example: HW_REG_OVERRIDE(0x50EC, 0x0009); // Set slower frequency offset tracking when repetition factor is 3 or 4.

2.6 PDIFDECIM

TI recommends using the maximum possible *pdifdecim* setting from the Excel calculator because a lower *pdifdecim* setting reduces the combined deviation and frequency offset supported. *pdifdecim* is set in the API through the symbolRate field in the CMD_PROP_RADIO_DIV_SETUP.

Add the following to the CMD_PROP_RADIO_DIV_SETUP:

.symbolRate.decimMode = 0x1, 3, or 5 // PDIF decimation (pdifdecim) set manually to 0, 1, or 2, respectively

2.7 Examples

This section provides two examples on how the CC13x0 device can be configured to be backward compatible with designs using the CC1101 device. Use the rfPacketRX and rfPacketTX examples available when downloading the SimpleLink™ CC13x0 Software Development Kit [3] as a starting point.

2.7.1 CC1101 Mode: 2.4 kBaud, ±5.2-kHz Deviation

- 49 kHz rxBW
- 4-byte sync word; 0xD391D391
- Repetition factor = 2
- Use the 10-kbps settings from SmartRF Studio [2] as a starting point.
- Change data rate to 4.8 kbps, deviation to 5.2 kHz, and rxBW to 49 kHz.
- Include the MCE patch, which is part of the SimpleLink™ CC13x0 Software Development Kit [3].

Three of the 10-kbps, 2-GFSK overrides from SmartRF™ Studio [2] must be changed (see Table 2).

Table 2. Register Overrides to be Changed from 10-kbps, 2-GFSK Override List

Override	Comment
MCE_RFE_OVERRIDE(1, 0, 0, 1, 0, 0)	Run the MCE and RFE patches.
HW_REG_OVERRIDE(0x6088,0xC118)	RFC_RFE.SPARE1 Enabling PA ramping 0xC1, AGC reference level [dB] 0x18
HW_REG_OVERRIDE(0x608C,0xC113)	RFC_RFE.SPARE2, PA ramping settings
HW_REG_OVERRIDE(0x5104,0x3030)	Increase sync threshold.

Two of the 10-kbps, 2-GFSK overrides from SmartRF Studio must be removed (see Table 3).

Table 3. Register Overrides to be Removed from 10-kbps, 2-GFSK Override List

Override	Comment
HW_REG_OVERRIDE(0x52AC,0x3C05)	Remove from SmartRF Studio code export.
HW_REG_OVERRIDE(0x50EC,0x0005)	Remove from SmartRF Studio code export.

Table 4 shows the overrides needed in addition to the recommended 10-kbps, 2-GFSK overrides from SmartRF Studio [2].

Table 4. Additional Register Overrides Required for the Patch

Override	Comment
(uint32_t)0x000C8103	No sync word and preamble configuration from RF Core
HW_REG_OVERRIDE(0x5050,0x1F77)	32b sync word, repeat 8 symbol preamble eight times
HW_REG_OVERRIDE(0x5058,0x00CC)	Preamble pattern
HW_REG_OVERRIDE(0x510C,0xF0CF)	Part of sync word (sync0)
HW_REG_OVERRIDE(0x5110,0xC0C3)	Part of sync word (sync1)
HW_REG_OVERRIDE(0x5114,0xF0CF)	Part of sync word (sync2)
HW_REG_OVERRIDE(0x5118,0xC0C3)	Part of sync word (sync3)
HW_REG_OVERRIDE(0x5178,0x0002)	Repetition factor

2.7.2 CC1101 Mode: 1.2 kBaud, ±5.2-kHz Deviation

- 49 kHz rxBW
- 2-byte sync word; 0xD391
- Repetition factor = 4
- Use the 10-kbps settings from SmartRF Studio [2] as a starting point.
- Change data rate to 4.8 kbps, deviation to 5.2 kHz, and rxBW to 49 kHz.
- Include the MCE patch, which is part of the SimpleLink™ CC13x0 Software Development Kit [3].

Three of the 10-kbps, 2-GFSK overrides from SmartRF Studio [2] must be changed (see Table 5).

Table 5. Register Overrides to be Changed from 10-kbps, 2-GFSK Override List

Override	Comment
MCE_RFE_OVERRIDE(1, 0, 0, 1, 0, 0)	Run the MCE and RFE patches.
HW_REG_OVERRIDE(0x6088,0xE118)	RFC_RFE.SPARE1 Enabling PA ramping 0xC1, AGC reference level [dB] 0x18
HW_REG_OVERRIDE(0x608C,0xC113)	RFC_RFE.SPARE2, PA ramping settings
HW_REG_OVERRIDE(0x50EC,0x0009)	Set slower frequency offset tracking when repetition factor is 4.

Two of the 10-kbps, 2-GFSK overrides from SmartRF Studio must be removed (see Table 6).

Table 6. Register Overrides to be Removed from 10-kbps, 2-GFSK Override List

Override	Comment
HW_REG_OVERRIDE(0x5104,0x7F30)	Remove from SmartRF Studio code export.
HW_REG_OVERRIDE(0x52AC,0x3C05)	Remove from SmartRF Studio code export.

Table 7 shows the overrides needed in addition to the recommended 10-kbps, 2-GFSK overrides from SmartRF Studio [2].

Table 7. Additional Register Overrides Required for the Patch

Override	Comment
(uint32_t)0x000C8103	No sync word and preamble configuration from Cortex-M0
HW_REG_OVERRIDE(0x5050,0x0F7F)	16b sync word, repeat 16 symbol preamble 8 times.
HW_REG_OVERRIDE(0x5058,0xF0F0)	Preamble pattern
HW_REG_OVERRIDE(0x510C,0xF00F)	Part of sync word (sync0)
HW_REG_OVERRIDE(0x5110,0xF000)	Part of sync word (sync1)
HW_REG_OVERRIDE(0x5114,0xF0FF)	Part of sync word (sync2)
HW_REG_OVERRIDE(0x5118,0xFF00)	Part of sync word (sync3)
HW_REG_OVERRIDE(0x5178,0x0004)	Repetition factor

If the CC1101 sync word is 4 bytes (that is, 0xD391D391) the CC13x0 can be set up to do 3× bit repetition and use only the 21 least significant bits of the CC1101 sync word when programming the CC13x0 sync word in RX.

3 References

References for this document follow:

1. Link to Excel spreadsheet
2. SmartRF™ Studio 7
3. SimpleLink™ CC13x0 Software Development Kit

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources 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.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

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