



Torstein Ermesjo

## ABSTRACT

This application report provides a guideline on how to find optimal settings for the CC13xx family if the settings in SmartRF™ Studio software cannot be used unmodified.

---

## Table of Contents

<b>1 Introduction</b> .....	2
<b>2 Recommended Test Setup</b> .....	2
<b>3 Overrides</b> .....	2
<b>4 Find the Needed RX BW</b> .....	3
<b>5 How to Set the Deviation</b> .....	3
<b>6 Theoretical Sensitivity</b> .....	4
<b>7 Determine AGC_REF Level</b> .....	4
<b>8 Determine Anti-Aliasing BW</b> .....	5
<b>9 Determine PA Ramp Settings</b> .....	5
<b>10 Intermediate Frequency (IF)</b> .....	6
<b>11 LNA Ib Offset</b> .....	6
<b>12 Sensitivity and Selection of Sync Word</b> .....	7
<b>13 Narrowband</b> .....	7
13.1 Frequency Offset Tolerance.....	7
13.2 Low Datarate.....	8
13.3 Phase Noise.....	9
<b>14 RSSI Offset</b> .....	10

## List of Tables

Table 9-1. PA Ramp Times for CC13x0.....	6
--	---

## Trademarks

SmartRF™ is a trademark of Texas Instruments.  
All trademarks are the property of their respective owners.

## 1 Introduction

Texas Instruments provides a number of recommended settings in SmartRF Studio for CC13xx and CC26xx. If a different data rate, deviation or frequency band than covered by those listed in SmartRF Studio is needed, it is required to adjust a few settings to get the best possible performance. This may also be the case if changing the RX bandwidth for an existing setting. This application report provides a guideline on how settings for a wanted data rate can be found. The steps covered in this application report may not cover all possible combinations.

The guidelines listed in this application report are only valid when the following cpe patch is used:

CC13x0: rf\_patch\_cpe\_genfsk

CC13x2: rf\_patch\_cpe\_prop

## 2 Recommended Test Setup

To fully verify a setting, extensive testing is required.

A minimum to run all the measurements suggested in this document is:

- Signal generator that is able to send packets
- Spectrum analyzer
- Software that is able to control the instruments for automated measurements. This may be Labview or Matlab.

All testing should be done conducted. Strong unwanted transmitters could impact the results and use of a shielded test setup should be considered if a base station or similar impacts the results.

A VNA or similar should be used to measure the cable loss if exact results are needed. For sensitivity measurements, it is recommended to adjust for the frequency offset for the DUT measured. First, set the DUT to send a CW. Measure the frequency offset using the spectrum analyzer and adjust the frequency set on the signal generator to account for the offset.

Measurements should be done on a sufficient number of devices to give statistical significance. Ideally measurements should also be done for min and max temperature the device will be used in.

## 3 Overrides

The parameters the modem uses are set in a series of registers. Some registers change meaning dependent on which patch is used and the registers are placed in different domains. Overrides are therefore used to set or change the registers the modem is using. In this document, the following commands are typically used to set the wanted override:

- HW\_REG\_OVERRIDE
- ADI\_HALFREG\_OVERRIDE
- ADI\_2HALFREG\_OVERRIDE
- (uint32\_t)<32 bit>

Overrides are typically placed in the override list. This is a uint32\_t array named pOverrides\_X in the examples from Texas Instruments where X is a descriptive name of the override list. An override list is always terminated by (uint32\_t)0xFFFFFFFF. Some of the overrides listed through this document already exist in the override list. In these cases, the override should just be changed to the new value. If the override needed does not exist in the current override list it has to be added to the list. The recommended is to add the new override at the end of the list but before the termination word.

In SmartRF Studio, overrides can be edited using the Override Editor.

## 4 Find the Needed RX BW

The transmitted signal will have a certain signal bandwidth (SBW), which depends on the symbol rate and modulation format. This bandwidth can be approximated by Carson's rule given that the modulation format is FSK:

$$SBW = 2 * f_m + 2 * f_{dev} \quad (1)$$

Where

- $f_m$  is the highest modulating frequency.  $2 * f_m =$  symbol rate
- $f_{dev}$  is the frequency deviation.  $2 * f_{dev} =$  frequency separation

Since GFSK uses a shaping to reduce the side lobes the GFSK signal bandwidth will be less than estimated by Carson's rule. The exact signal bandwidth can be measured with a spectrum analyzer.

On the receiver side there is an RX channel filter, which is centered on the down-converted received RF frequency, such that, the intermediate frequency (IF). The RX filter has a programmable bandwidth (RX BW). The transmitted signal bandwidth has to be less than RX BW, but the frequency error of the transmitter and receiver must also be taken into account.

If there is an error in the transmitter carrier frequency and the receiver LO frequency, there will also be an error in the IF frequency. For simplicity assume the frequency error in the transmitter and receiver is equal (same type of crystal). If the receiver has an error of  $-X$  ppm and the transmitter has an error of  $+X$  ppm the IF frequency will have an error of  $+2 * X$  ppm. Conversely, if the receiver has an error of  $+X$  ppm and the transmitter an error of  $-X$  ppm the IF frequency will have an error of  $-2 * X$  ppm.

RX BW has to be larger than the maximum SBW plus the maximum frequency error due to crystal inaccuracies. Worst case scenario will be for the crystal errors on TX and RX side to be of opposite signs.

$$RX\ BW > SBW + 4 * XTAL_{ppm} * f_{RF} \quad (2)$$

where

- $XTAL_{ppm}$  is the total accuracy of the crystal including initial tolerance, temperature drift, loading, and ageing
- $f_{RF}$  is the RF operating frequency

---

### Note

There are both a lower and an upper limit on the symbol rate for a given RX BW. The wanted combination of symbol rate and RX BW should be entered in SmartRF Studio. The tool will give an error message if the combination is not possible.

---

## 5 How to Set the Deviation

Theoretically, there is an optimum separation/symbol rate setting if you simultaneously minimize the receiver filter bandwidth. Every halving of receiver filter bandwidth improves sensitivity by 3 dB whereas sensitivity vs separation/datarate decreases with about 1.5-2.5 dB per halving down to a certain limit where the loss increases very fast. In our experience a modulation index ( $h = 2 * \text{deviation} / \text{symbol rate} = \text{separation} / \text{symbol rate}$ ) = 0.5 - 1 is a good design compromise.

For 4-GFSK the modulation index for inner symbols should be  $\geq 1/3$ . Lower modulation index will degrade sensitivity significantly.

## 6 Theoretical Sensitivity

Before starting to measure RF performance, it is recommended to calculate the theoretical sensitivity to be able to know if the measured sensitivity is as expected. To get close to the theoretical value, use BER = 1%. It is possible to calculate the relation between PER and BER using [Equation 3](#).

$$\text{PER} = 1 - (1 - \text{BER})^N \quad (3)$$

where, N is the total packet length in number of bits including payload and CRC. For sensitivity measurements it is recommended to compare the number of received packets without error vs. the number of transmitted packets.

The theoretical sensitivity is:

$$-174 \text{ dBm/Hz} + \text{NF} + 10 \cdot \log_{10}(\text{RX BW}) + \text{SNR} \quad (4)$$

where,

- -174 dBm: Thermal noise in a 1 Hz bandwidth
- NF: Noise figure. Noise of all stages referred to the input in a 1 Hz bandwidth
- RX BW: Receiver filter bandwidth
- SNR: Signal-to-noise ratio required by demodulator
- Sum of the first three terms is the total integrated noise in the system and is called the noise floor.
- NF can be set to 7 dB in the calculations. This is set mainly by the LNA.
- SNR is dependent of modulation format and modulation index. For 2-(G)FSK with  $h = 1$  and BER = 1% assume SNR = 7 dB

## 7 Determine AGC\_REF Level

To get the wanted dynamic range in the receive chain Automatic Gain Control (AGC) is used. The gain in the receive chain is adjusted according to the input level. The gain always is set to max gain when going into RX and the gain is then adjusted according to the input level. AGC\_REF sets for which input signal level the AGC sets the gain to something else than maximum.

Too high AGC\_REF will degrade the blocking performance and too low will give increased residual BER.

For CC13x0 the AGC\_REF level is set by the following override:

**HW\_REG\_OVERRIDE(0x6088,0xYYXX)**, where the last byte (0xXX) is wanted AGC\_REF. YY sets PA ramping, covered in [Chapter 9](#).

For CC13x2 the AGC\_REF level is set by the following override:

**HW\_REG\_OVERRIDE(0x609C,0x00XX)**, where the last byte is wanted AGC\_REF.

The following steps should be done to set the AGC\_REF level:

- Measure the sensitivity level with max AGC\_REF level
- Do a sweep from -120 dBm to 0 dBm with 1 dB steps and read out the agc gain for each step.

The gain could be read out with a command:

```
Rfc_CMD_READ_RFREG t cmdreadRXgain = {
    .commandNo = 0x0601,
    .address = (0x40046094 & 0xFFFF),
    .value = 0,
};
```

The .address field is 0x40046080 for CC13x0 and 0x40049094 for CC13x2.

Set the chip in RX forever using:

```
/* Enter RX mode and stay forever in RX */
RF_EventMask terminationReason = RF_runCmd(rfHandle, (RF_Op*)&RF_cmdRxTest,
                                             RF_PrioritNormal, 0,
                                             RF_EventRxEntryDone);
```

A clock event or similar can be used to read out the gain at given intervals using the following:

```
RF_runImmediateCmd(rfHandle, (uint32_t*)&cmdreadRXgain);
```

The AGC\_REF level should be set 7- 8 dB higher than the sensitivity limit. To verify the new AGC\_REF level, a PER vs level measurement has to be done. The resulting curve should not have residual PER above the sensitivity limit. In other words, the PER should be 0% for all input levels ~3 dB or higher than the sensitivity limit.

## 8 Determine Anti-Aliasing BW

The anti-aliasing BW (AABW) should be as low as possible to give best possible blocking. The AABW is set by the following override:

**ADI\_HALFREG\_OVERRIDE(0,61,0xMask,0xValue)**. Mask is a bit mask. This is typically set to 0xF. "Value" is 4 bits where a lower value gives larger bandwidth.

Find the anti-aliasing bandwidth that gives a marginal decrease in sensitivity and select a setting value that is two settings below this. Meaning if tests using 0x09 starts to give a reduced sensitivity, 0x07 should be used.

## 9 Determine PA Ramp Settings

PA ramping is used to reduce splatter in the frequency domain when the PA starts and stops sending a packet. The target is that the PA as a maximum uses a time equal to 1 – 2 bits to do ramping. In addition relevant regulatory transient TX requirements should be considered.

The ramp up and down timing are set differently for CC13x0 and CC13x2.

The ramp up time should be measured using a spectrum analyzer using zero span and a positive trigger to capture the start of a packet. The sweep time should be adjusted to reflect the data rate to simplify the measurement.

The waiting time before ramping down should be measured with a spectrum analyzer using zero span and set the trigger to trig on negative flank. If the waiting time is too short the ramping down will be cut-off.

For CC13x0:

The ramping is controlled through two registers:

**HW\_REG\_OVERRIDE(0x6088,0xXXYY)**

**HW\_REG\_OVERRIDE(0x608C,0xXXYY)**

Where XX is the value that is possible to adjust. [Table 9-1](#) shows the estimated ramp time as a function of settings.

**Table 9-1. PA Ramp Times for CC13x0**

0x6088 Override Value	0x608C Override Value	Ramp Time [ $\mu$ s]
0x1F	0x3F	4
0x08	0x10	6
0x04	0x08	10
0x02	0x04	17
0x01	0x02	30
0x41	0x82	45
0x61	0x81	81
0xE1	0xC1	304

The actual ramp time should be measured.

For CC13x2:

The PA ramping up time is set by the following override:

**ADI\_2HALFREG\_OVERRIDE(0,16,0x8,0xL,17,0x1,0xM)**, where L is least significant bit (8 or 0 to set as 1 or 0 respectively as L is bit[3] in the nibble) and M is most significant bit (1 or 0).

This means that it is a total of four possible PA ramp times.

The following override set the wait time before turning ramping off:

**HW\_REG\_OVERRIDE(0x6028,0x00XX)**, where 0xXX is the wait time.

## 10 Intermediate Frequency (IF)

Most PHYs uses the default IF frequency. The default frequency is dependent on the RX BW and is given in the *Receiver Bandwidth Settings* table in the device-specific Technical Reference Manual (TRM). In some cases it could be a benefit to use a different IF than the default. The IF frequency is set through `CMD_PROP_RADIO_DIV_SETUP.intFreq`. A lower IF can potentially improve blocking as a lower anti-aliasing BW can be used.

The `.intFreq` can be calculated as `dec2hex(IF frequency*4096/1e6)`.

Good design practice:  $RXIF - \text{deviation} + \text{guard band} > DC$ . Guard band is needed to keep the "lower end" of the RX BW sufficiently above DC. There is a DC filter that removes  $1/f$  noise, and the guard band is needed so that the DC filter does not remove any of the signal.

## 11 LNA Ib Offset

It is possible to improve sensitivity by up to 0.5 dB by increase the current in the LNA and thereby the total current consumption.

The LNA current can be set through the following override:

```
// Set LNA IB offset to 0xL
(uint32_t)0x000L8883,
```

where the maximum value for L is 0xF.

The RSSI offset may change when changing the LNA gain.

## 12 Sensitivity and Selection of Sync Word

The sensitivity can be impacted by the selection of the sync word and the sync threshold.

First, the sync word should have good auto correlation properties. The auto correlation could be visualized in Matlab or similar using `plot(xcorr(2*vector-1))` where vector is the sync word in the following format: Vector = [1,0,0,1,1,0,.....]. Ideally the plot should have only one peak. If the plot has more than one peak this shows that it is a risk for getting a false sync. The probability for finding a false sync increases with the amplitude of the peak. Preamble is a good example of a sync word with poor auto correlation since the plot has many peaks where some are fairly high. A false sync is when the radio gives a sync found when the wanted sync word is not fully received which cause the received bits to be shifted which will give a CRC error. For short sync words, it is often recommended to include the preamble as part of the sync word.

Typically, the sync search consists of two parts. First, a correlator compares the incoming bit stream with the programmed sync. The sync threshold sets how equal the incoming steam has to be with the wanted sync word. Too strict threshold and the sensitivity will be limited by the threshold. Too loose and false sync will occur. In the second part the incoming stream is compared bit by bit to the programmed word. The second part is not done for all phys.

The sync threshold can be set as

CC13x0: **HW\_REG\_OVERRIDE(0x5104,0xYYZZ)**

CC13x2: **HW\_REG\_OVERRIDE(0x5114,0xYYZZ)**

where YY and ZZ is the threshold for the two correlators. For simplicity the two thresholds can be set equal. Higher value is stricter. The default sync threshold is given by:

$$\text{Sync Theshold} = (30 * nSwBits) / 32 + 9 \quad (5)$$

## 13 Narrowband

For narrowband applications the following parameters have to be considered:

- Frequency offset tolerance
- Low datarate
- Phase noise

Here narrowband is defined as applications that use ~25 kHz RX BW or lower. CC13x0 is not a narrowband chip and the CC13x2 should be selected for these applications but settings for CC13x0 are included since increasing the frequency offset tolerance could be useful even in systems with a wider RX BW.

### 13.1 Frequency Offset Tolerance

According to [Equation 2](#),  $XTAL_{ppm}$  has to be set low if the wanted signal bandwidth is close to the wanted RX BW to be able to receive. In some cases it is not possible or wanted to decrease  $XTAL_{ppm}$  to the extent that the equation dictates. An alternative is to use "Feedback to NCO<sup>1</sup>" (FB2NCO), which increases the effective RX BW without increasing the noise BW which determines the sensitivity limit. In other words, it is possible to increase the frequency error tolerance without degrading sensitivity.

For the CC13x0 family, FB2NCO is controlled with:

**HW\_REG\_OVERRIDE(0x52AC, <value>)**

For the CC13x2 family, FB2NCO is controlled with:

**HW\_REG\_OVERRIDE(0x5320, <value>)**

<sup>1</sup> NCO: Numerically Controlled Oscillator

Where <value> is a 32 bit register but only the 16 LSB bits are used.

- Enable FB2NCO frequency offset compensation (FOC) mode by setting bit[13]=1
- Disable strict sync word check by setting bit[12]=1. Since the strict sync search is turned off it could be required to increase the sync threshold. How to do this is documented in <SDK install path>/docs/proprietary-rf/proprietary-rf-users-guide/proprietary-rf/packet-format.html#sync-word-qualifier.
- Configure FOC\_GAIN[11:8]. Higher value will give larger frequency "correction" steps and might cause a longer settling time.
- Configure FOC\_LIMIT[7:0] for offset regulation
  - The improved BW is given as FOC\_LIMIT\*NCO resolution. Resolution is RF frequency\*LO\_DIV/2/12/24/1024/2^bde1
  - Run PER vs level vs offset and check if there is an error floor (aka residual PER) and if the frequency tolerance is acceptable.
  - If an error floor is present the adjustment loop could be made slower by adjusting the IIRGAIN. The IIRGAIN is available through bit[3:1] of <value>: **HW\_REG\_OVERRIDE(0x50EC, <value>)** for CC13x0 and **HW\_REG\_OVERRIDE(0x0x50FC, <value>)** for CC13x2. The most optimal value has to be found through trial and error.

For FB2NCO to work properly the NCO resolution should be less than half the deviation.

$RXBW = chfi_{val} / (2^{(bde1 + bde2)} * 288) * LODIV * RF \text{ frequency} * 1000/2$  where chfi can have one of the following four values: 0.5, 0.33, 0.416, 0.29. chfi[1:0] selects which chfi\_val to use according to the table below:

chfi[1:0]	chfi_val
0	0.5
1	0.33
2	0.416
3	0.29

The following can be done to improve NCO resolution:

Change the values for bde1 and bde2:

- Set bde1 as high as possible. The max value for bde1 is 4.
- Add symbolRate.decimMode = 7 as part of the CMD\_PROP\_RADIO\_DIV\_SETUP settings.
- FOC\_LIMIT has to be adjusted if bde1 is increased since the "additional" BW is given as FOC\_LIMIT \* NCO resolution.
- Check AGC gain after changing bde2 and bde1.
- When intFreq = 0x8000 the IF is  $10^6/bde1$ . The performance with the new IF should be verified.

The rxbw setting given as part of the CMD\_PROP\_RADIO\_DIV\_SETUP is build up as follows:

$Rxbw = 0xbde2[7:5], bde1[4:2], chfi[1:0]$

### 13.2 Low Datarate

The default deviation resolution is 250 Hz. If the wanted deviation is not a factor of 250 Hz the resolution can be changed. Setting **modulation.deviationStepSz = 2** as part of the CMD\_PROP\_RADIO\_DIV\_SETUP settings sets the deviation resolution to 15.625 Hz.

Due to how the modem is designed not all datarates are valid for a given RXBW setting. SmartRF Studio can be used to find if the planned combination of datarate, deviation and RX BW is possible. If one or more values are outside the valid range the tool will give an error message and the valid range.

These limits can be modified by using bit repetition. This is covered in detail in this app note: [CC13x0 Low Data Rate Operation](#)

For CC13x2 bit repetition doesn't require a separate patch. To turn the bit repetition on, set .formatConf.fecMode = 0x2 under CMD\_PROP\_RADIO\_DIV\_SETUP. The repetition factor is set through the following override:

## HW\_REG\_OVERRIDE(0x5324,0x000X)

Where x is the wanted factor.

### 13.3 Phase Noise

For narrow band systems it could be required to modify the phase noise shaping by adjusting the loop bandwidth. Higher loop bandwidth gives better close-in phase noise. For some systems it could be required to have different loop bandwidth in RX and TX since getting a good result for TX parameters (ACP, close in spurs) may require a different loop BW than getting good close in selectivity in RX.

The overrides below are valid for CC13x2. If a loop bandwidth that are not listed below is needed, please make a request.

#### Loop bandwidth: 20 kHz

```
//Synth: Set loop bandwidth after lock to 20 kHz (K2)
(uint32_t)0x0A480583,
//Synth: Set loop bandwidth after lock to 20 kHz (K2)
(uint32_t)0x000005A3,
//Synth: Set loop bandwidth after lock to 20 kHz (K3, LSB)
(uint32_t)0x7AB80603,
//Synth: Set loop bandwidth after lock to 20 kHz (K3, MSB)
(uint32_t)0x00000623,
//Synth: Set FREF = 8 MHz
(uint32_t)0x000684A3,
```

#### Loop bandwidth: 40 kHz

```
//Synth: Set loop bandwidth after lock to 40 kHz (K2)
(uint32_t)0x29200583,
//Synth: Set loop bandwidth after lock to 40 kHz (K2)
(uint32_t)0x000005A3,
//Synth: Set loop bandwidth after lock to 40 kHz (K3, LSB)
(uint32_t)0xF5700603,
//Synth: Set loop bandwidth after lock to 40 kHz (K3, MSB)
(uint32_t)0x00000623,
//Synth: Set FREF = 4 MHz
(uint32_t)0x000C84A3,
```

#### Loop bandwidth: 60 kHz

```
//Synth: Set loop bandwidth after lock to 60 kHz (K2)
(uint32_t)0x5C870583,
//Synth: Set loop bandwidth after lock to 60 kHz (K2)
(uint32_t)0x000005A3,
//Synth: Set loop bandwidth after lock to 40 kHz (K3, LSB)
(uint32_t)0x70280603,
//Synth: Set loop bandwidth after lock to 40 kHz (K3, MSB)
(uint32_t)0x00010623,
//Synth: Set FREF = 4 MHz
(uint32_t)0x000C84A3,
```

#### Loop bandwidth: 80 kHz

```
//Synth: Set loop bandwidth after lock to 80 kHz (K2)
(uint32_t)0xA47E0583,
//Synth: Set loop bandwidth after lock to 80 kHz (K2)
(uint32_t)0x000005A3,
//Synth: Set loop bandwidth after lock to 80 kHz (K3, LSB)
(uint32_t)0xEAE00603,
//Synth: Set loop bandwidth after lock to 80 kHz (K3, MSB)
(uint32_t)0x00010623,
//Synth: Set FREF = 8 MHz
(uint32_t)0x000684A3,
```

**Loop bandwidth: 150 kHz**

```
//Synth: Set loop bandwidth after lock to 150 kHz (K2)
(uint32_t)0x424C0583,
//Synth: Set loop bandwidth after lock to 150 kHz (K2)
(uint32_t)0x000205A3,
//Synth: Set loop bandwidth after lock to 150 kHz (K3, LSB)
(uint32_t)0x98630603,
//Synth: Set loop bandwidth after lock to 150 kHz (K3, MSB)
(uint32_t)0x00030623,
//Synth: Set FREF = 8 MHz
(uint32_t)0x000684A3,
```

**Loop bandwidth: 200 kHz**

```
//Synth: Set loop bandwidth after lock to 200 kHz (K2)
(uint32_t)0x04150583,
//Synth: Set loop bandwidth after lock to 200 kHz (K2)
(uint32_t)0x000405A3,
//Synth: Set loop bandwidth after lock to 200 kHz (K3, LSB)
(uint32_t)0xCB2F0603,
//Synth: Set loop bandwidth after lock to 200 kHz (K3, MSB)
(uint32_t)0x00040623,
//Synth: Set FREF = 8 MHz
(uint32_t)0x000684A3,
```

**Other overrides:**

In some cases better results can be achieved using FREF dithering:

```
//Synth: Set FREF dither = 9.6 MHz
(uint32_t)0x000584B3,
```

Using an IIR filter will help reducing the phase noise outside the loop filter BW but will increase the power consumption some. If the far out phase noise is too high with the settings given above the overrides below could be tested.

```
//Two word override starting from LoopCoeff
HW32_ARRAY_OVERRIDE(0x4028,2),
// IIR EN, 2nd order, IIR_FILT_BW=1
(uint32_t)0x38000000,
//Set RFC_FSCA:PLLCTL0.IIR_CLK_DIV to 1
(uint32_t)0x01608402,
```

## 14 RSSI Offset

The RSSI value calculated by the chip has to be adjusted by an offset to give out a number that represents the input power in dBm.

To find the RSSI offset: Set the radio in RX. Apply a CW where the power in to the chip is stepped from -130 dBm to 10 dBm in 1 dB step. For each step, read the RSSI value using the RF\_getRssi() function. It is recommended to do the measurements on several boards since the RSSI results have some chip to chip variation.

The RSSI offset is set by the following override:

**(uint32\_t)0x00XX88A3**

where XX is the offset in two's complement.

The RSSI offset is measured and given as an override for the settings given in SmartRFStudio. Using a different RF frontend than for the board the settings are measured for could change the RSSI offset. A typical example is if a Front End Module (FEM) is added since the LNA in the FEM adds gain.

## IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale ([www.ti.com/legal/termsofsale.html](http://www.ti.com/legal/termsofsale.html)) or other applicable terms available either on [ti.com](http://ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2020, Texas Instruments Incorporated