|  |  |  |
| --- | --- | --- |
| AN60594 | | |
| ＜＜★PSoC® 3 and PSoC 5LP: Low-Frequency FSK Modulation and Demodulation＞＞PSoC® 3およびPSoC 5LP: 低周波数FSK変調および復調   |  | | --- | | ＜＜★Author: Todd Dust＞＞著者：Todd Dust | | ＜＜★Associated Project: Yes＞＞関連プロジェクト：はい | | ＜＜★Associated Part Family: CY8C3xxx,CY8C5xxx LP＞＞関連部品ファミリ：CY8C3xxx、CY8C5xxx LP | | ＜＜★Software Version: PSoC® Creator™4.0＞＞ソフトウェアバージョン: PSoC® Creator™4.0 | | ＜＜★Related Application Notes: [AN52705](http://www.cypress.com/?rID=37793), [AN2336](http://www.cypress.com/?rID=2735), [AN2108](http://www.cypress.com/?rID=2780), [AN58827](http://www.cypress.com/?rID=40247), [AN76458＞＞](http://www.cypress.com/?rID=64581)関連アプリケーションノート：[AN52705](http://www.cypress.com/?rID=37793)、[AN2336](http://www.cypress.com/?rID=2735)、[AN2108](http://www.cypress.com/?rID=2780)、[AN58827](http://www.cypress.com/?rID=40247)、[AN76458](http://www.cypress.com/?rID=64581) | | | |
| ＜＜★［01］AN60594 describes how to create a frequency shift keying (FSK) modulator and demodulator with PSoC® 3 or PSoC 5LP, with digital 0s represented by 2100 Hz and digital 1s represented by 1300 Hz. The method described in this application note uses zero CPU—everything is done in PSoC hardware.＞＞AN60594は、PSoC® 3またはPSoC 5LPで周波数シフトキーイング（FSK）変調器と復調器を作成する方法を説明します。デジタル0は2100 Hzで表され、デジタル1は1300 Hzで表されます。このアプリケーションノートで説明する方法では、CPUを使用しません。すべてがPSoCハードウェアで行われます。＜＜★This application note covers only the physical layer implementation of an FSK transmitter and receiver; higher-level encoding techniques and physical modem connections are not discussed.＞＞このアプリケーションノートでは、FSKトランスミッターとレシーバーの物理層の実装のみを扱います。より高レベルのエンコーディング技術と物理モデム接続については説明しません。 |

目次

[はじめに 1](#_Toc396928705)

[＜＜★FSK Modulator＞＞FSK変調器 2](#_Toc396928706)

[＜＜★Transmit Results＞＞結果を送信する 3](#_Toc396928707)

[＜＜★Generating Different Frequencies＞＞異なる周波数を生成する 4](#_Toc396928708)

[＜＜★FSK Demodulator＞＞FSK復調器 6](#_Toc396928709)

[＜＜★Band-pass Filter＞＞バンドパスフィルター 6](#_Toc396928710)

[＜＜★［0］Zero-Crossing Detector＞＞ゼロ交差検出器 6](#_Toc396928711)

[＜＜★Correlator＞＞相関器 6](#_Toc396928712)

[LPF 7](#_Toc396928713)

[＜＜★Digitize＞＞デジタル化 7](#_Toc396928714)

[＜＜★Receiving Results＞＞結果を受け取る 8](#_Toc396928715)

[＜＜★Design Modifications＞＞設計変更 9](#_Toc396928716)

[＜＜★Attached Projects＞＞付属プロジェクト 9](#_Toc396928717)

[まとめ 9](#_Toc396928718)

[著者について 9](#_Toc396928719)

[付録A 10](#_Toc396928720)

[Worldwide Sales and Design Support 13](#_Toc396928721)

# はじめに

＜＜★In some applications, it is necessary to transmit digital data through analog signals.＞＞一部のアプリケーションでは、アナログ信号を介してデジタルデータを送信する必要があります。＜＜★This is often due to bandwidth limitations in the transmission medium.＞＞これは多くの場合、伝送媒体の帯域幅の制限が原因です。＜＜★There are numerous methods of transmitting digital data through analog signals; this application note specifically focuses on frequency shift keying (FSK) and how to implement an FSK modulator and demodulator in a PSoC 3 or PSoC 5LP device.＞＞アナログ信号を介してデジタルデータを送信する方法は多数あります。このアプリケーションノートでは、特に周波数シフトキーイング（FSK）と、PSoC 3またはPSoC 5LPデバイスにFSK変調器と復調器を実装する方法に焦点を当てています。

＜＜★［1］An FSK modulator converts digital data into two distinct frequencies; one frequency represents a digital 1 and the other represents a digital 0 (see Figure 1).＞＞FSK変調器は、デジタルデータを2つの異なる周波数に変換します。1つの周波数はデジタル1を表し、もう1つの周波数はデジタル0を表します（Figure 1を参照）。

Figure 1. ＜＜★Example FSK Waveform＞＞FSK波形の例



＜＜★The FSK demodulator receives the transmitted analog signal and demodulates it to determine the correct sequence of 1s and 0s.＞＞FSK復調器は、送信されたアナログ信号を受信して​​復調し、1と0の正しいシーケンスを決定します。

＜＜★This application note demonstrates how to configure a PSoC 3 or PSoC 5LP device to implement an FSK modulator and demodulator. The example project included with this application note uses 2100 Hz to represent digital 0 and 1300 Hz to represent digital 1.＞＞このアプリケーションノートは、FSK変調器と復調器を実装するためにPSoC 3またはPSoC 5LPデバイスを構成する方法を示します。このアプリケーションノートに含まれるサンプルプロジェクトでは、2100 Hzを使用してデジタル0を表し、1300 Hzを使用してデジタル1を表します。

＜＜★The advantage of PSoC with an FSK modulator and demodulator is that no CPU intervention is needed.＞＞FSK変調器と復調器を備えたPSoCの利点は、CPUの介入が必要ないことです。＜＜★［1］All of the demodulation and modulation is done in the integrated analog and digital blocks of PSoC 3 and PSoC 5LP.＞＞すべての復調と変調は、PSoC 3およびPSoC 5LPの統合されたアナログおよびデジタルブロックで行われます。＜＜★The CPU is left free to do other important system tasks.＞＞CPUは、他の重要なシステムタスクを実行するために解放されたままです。

1. ＜＜★［1］The included example is not a complete modem; it does not include the phone line or other interfaces; it also does not include the data processing code.＞＞含まれている例は完全なモデムではありません。電話回線やその他のインターフェースは含まれません。また、データ処理コードも含まれていません。

＜＜★If you are looking for information on Power Line Communication (PLC) and application of FSK, see the application note, [AN76458](http://www.cypress.com/?rID=64581).＞＞電力線通信（PLC）およびFSKのアプリケーションに関する情報をお探しの場合は、アプリケーションノート[AN76458](http://www.cypress.com/?rID=64581)を参照してください。

# ＜＜★FSK Modulator＞＞FSK変調器

＜＜★An FSK modulator converts digital data from a digital source such as a universal asynchronous receiver transmitter (UART) into the appropriate analog output signals.＞＞FSK変調器は、ユニバーサル非同期レシーバートランスミッター（UART）などのデジタルソースからのデジタルデータを適切なアナログ出力信号に変換します。＜＜★For this discussion, the analog output signal is a sine wave of either 1300 Hz or 2100 Hz, depending on the digital input.＞＞この説明では、アナログ出力信号は、デジタル入力に応じて、1300 Hzまたは2100 Hzの正弦波です。

＜＜★PSoC 3 and PSoC 5LP devices include voltage digital-to-analog converters (VDACs).＞＞PSoC 3およびPSoC 5LPデバイスには、電圧デジタルアナログコンバーター（VDAC）が含まれています。＜＜★The output voltage of the VDAC is controlled by an 8-bit digital word.＞＞VDACの出力電圧は、8ビットのデジタルワードによって制御されます。＜＜★This word can be updated either by the CPU or by the direct memory access (DMA).＞＞このワードは、CPUまたはダイレクトメモリアクセス（DMA）によって更新できます。＜＜★DMA is capable of reading an 8-bit word from RAM and writing that word to the VDAC.＞＞DMAは、RAMから8ビットワードを読み取り、そのワードをVDACに書き込むことができます。＜＜★The DMA is capable of automatically incrementing through an array in memory and writing to the VDAC.＞＞DMAは、メモリ内のアレイを介して自動的にインクリメントし、VDACに書き込むことができます。＜＜★This allows you to create a sine table in memory, and use the DMA to write values from the sine table to the VDAC.＞＞これにより、メモリにサインテーブルを作成し、DMAを使用してサインテーブルからVDACに値を書き込むことができます。

＜＜★The included example creates sine wave with PSoC  3 or PSoC 5LP using a VDAC with the DMA periodically writing values to it from a sine table in RAM. See Figure 2. The VDAC by itself has high output impedance; therefore, an opamp is used to buffer the VDAC voltage so it can drive a load.＞＞含まれている例では、VDACを使用してPSoC 3またはPSoC 5LPで正弦波を作成し、DMAがRAMの正弦テーブルから定期的に値を書き込みます。Figure 2を参照してください。VDAC自体は高い出力インピーダンスを持っています。したがって、オペアンプを使用してVDAC電圧をバッファし、負荷を駆動できます。

＜＜★The DMA can be configured to transfer individual bytes from the sine table in memory to the VDAC, based on a hardware trigger (DRQ).＞＞DMAは、ハードウェアトリガー（DRQ）に基づいて、メモリ内のサインテーブルからVDACに個々のバイトを転送するように構成できます。＜＜★The sine table is 64 elements wide, so the trigger must occur 64 times faster than the target frequencies.＞＞サインテーブルは64要素幅なので、トリガーはターゲット周波数より64倍速く発生する必要があります。＜＜★To generate sine waves of 1300 Hz and 2100 Hz, the DMA DRQ is triggered at 83.2 kHz or 134.4 kHz, respectively.＞＞1300 Hzおよび2100 Hzの正弦波を生成するために、DMA DRQはそれぞれ83.2 kHzまたは134.4 kHzでトリガーされます。

＜＜★This example uses two separate clocks to trigger the DMA transfer.＞＞この例では、2つの別個のクロックを使用してDMA転送をトリガーします。＜＜★A digital mux controls the input into the DRQ, as Figure 2 shows.＞＞Figure 2に示すように、デジタルマルチプレクサがDRQへの入力を制御します。＜＜★The mux is controlled by the digital source, in this case a UART.＞＞マルチプレクサはデジタルソース（この場合はUART）によって制御されます。＜＜★［1］This can be changed to any other digital component.＞＞これは、他のデジタルコンポーネントに変更できます。＜＜★For more information on DMA, see the application note, [AN52705](http://www.cypress.com/?rID=37793).＞＞DMAの詳細については、アプリケーションノート[AN52705](http://www.cypress.com/?rID=37793)を参照してください。

Figure 2. ＜＜★FSK TX PSoC Creator™ Schematic＞＞FSK TX PSoC Creator™回路図



＜＜★The associated project is configured to create the FSK waveforms based on the output of a UART.＞＞関連するプロジェクトは、UARTの出力に基づいてFSK波形を作成するように構成されています。＜＜★This program can be loaded into a CY8CKIT-001.＞＞このプログラムは、CY8CKIT-001にロードできます。＜＜★The main loop can be updated to control the output of the UART to send the desired data.＞＞メインループを更新して、UARTの出力を制御し、目的のデータを送信できます。＜＜★The FSK output can be observed on P0.1 and the UART output can be observed on P1.7. The project is configured to continuously send the string “Hello World”.＞＞FSK出力はP0.1で観察でき、UART出力はP1.7で観察できます。プロジェクトは、文字列「Hello World」を継続的に送信するように構成されています。

## ＜＜★Transmit Results＞＞結果を送信する

＜＜★Now that we know how to create an FSK modulator with PSoC 3 or PSoC 5 LP, it is time to look at the quality of the FSK waveforms.＞＞PSoC 3またはPSoC 5 LPでFSK変調器を作成する方法がわかったので、次に、FSK波形の品質を確認します。＜＜★Presented below are screen captures of the FSK outputs and their FFTs.＞＞以下は、FSK出力とそのFFTの画面キャプチャです。

Figure 3. ＜＜★2100 Hz Output＞＞2100 Hz出力

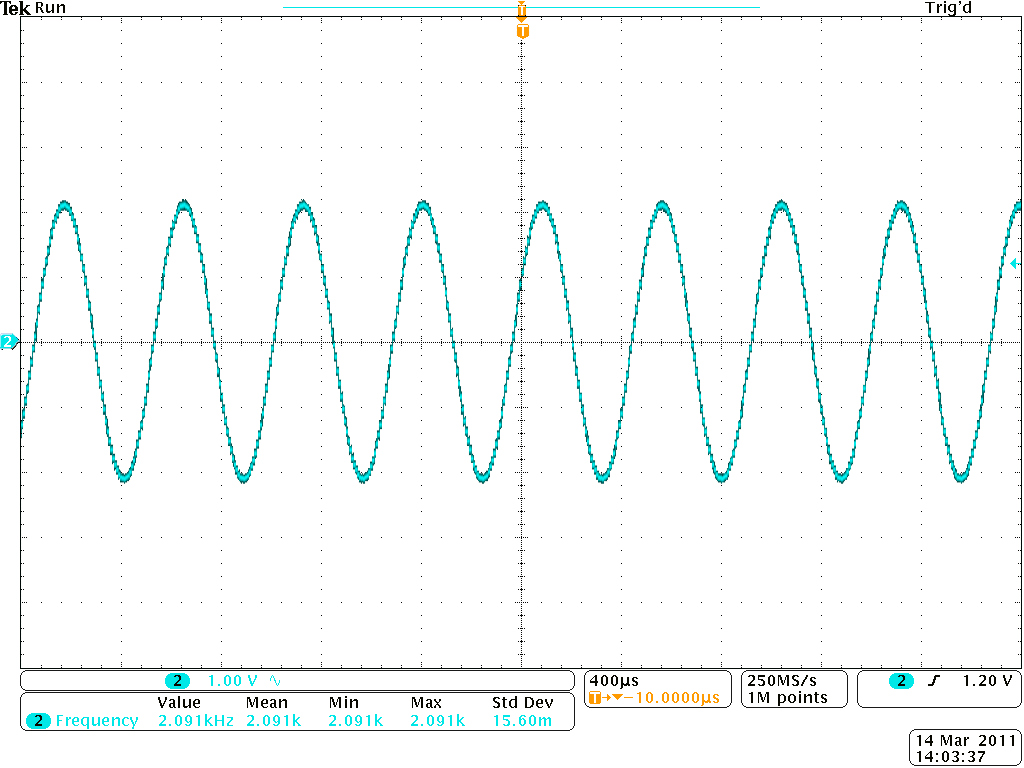
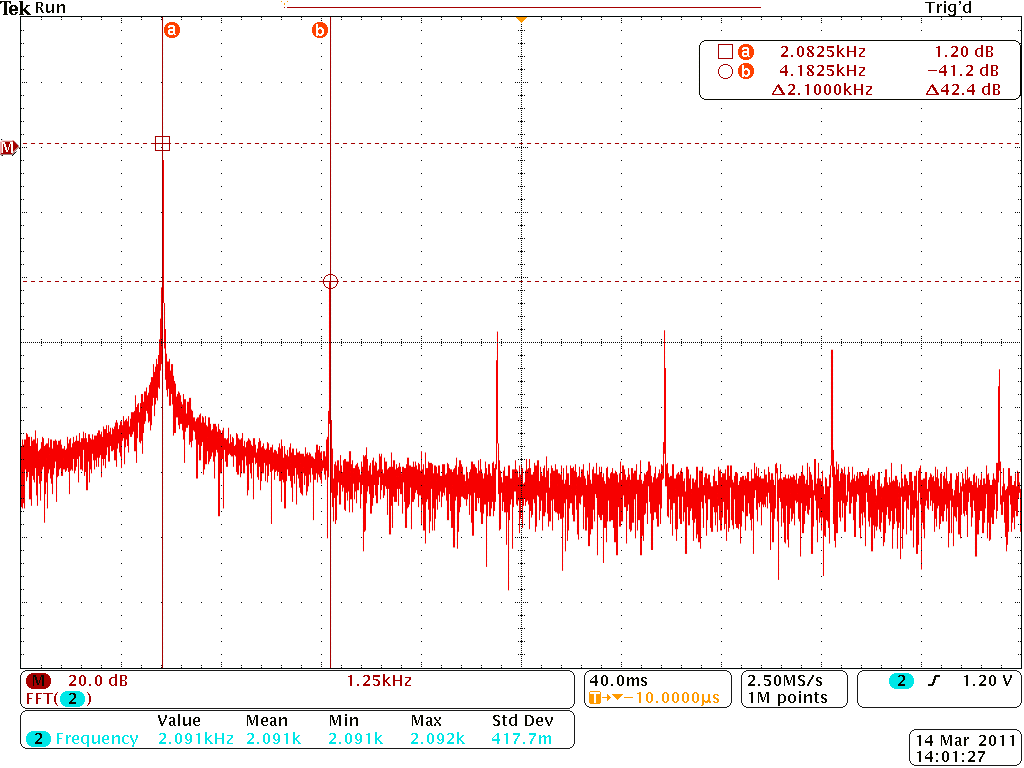
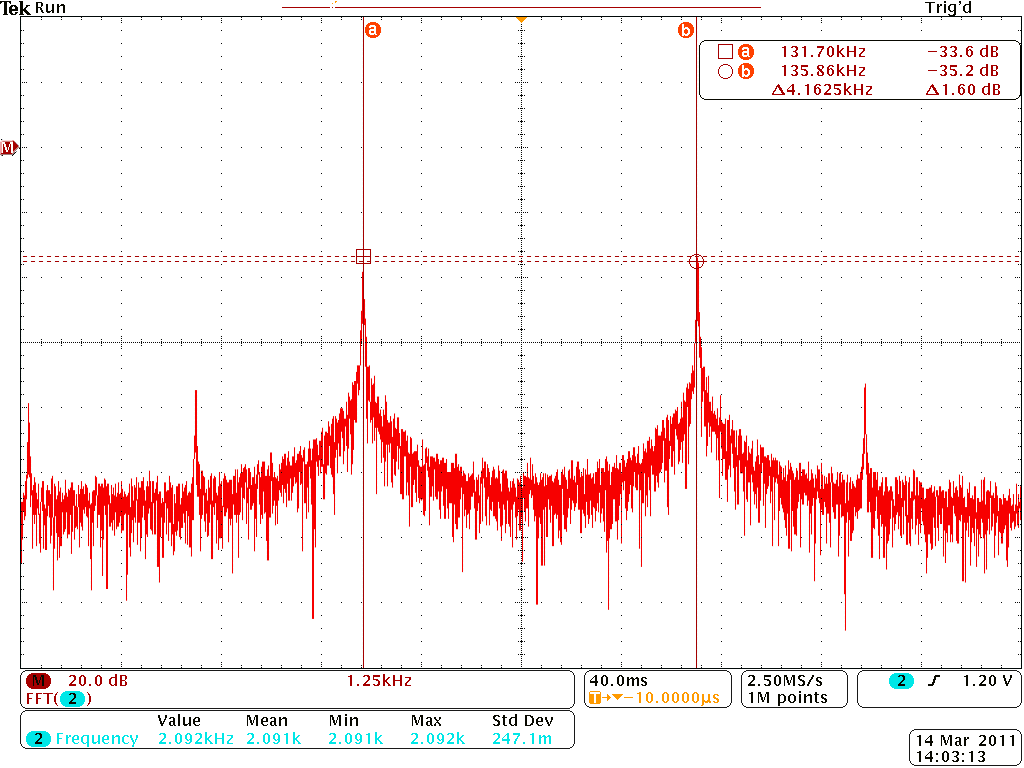


Figure 4. 2100 Hz FFT



＜＜★［2］The cursor window in the upper right of Figure 4 shows the fundamental frequency at 1.2 dB and the second harmonic at –41.2 dB, which is approximately 42 dB down.＞＞Figure 4の右上にあるカーソルウィンドウは、1.2 dBの基本周波数と–41.2 dBの2次高調波を示しています。これは約42 dB低くなっています。

Figure 5. ＜＜★2100 Hz Primary Alias＞＞2100 Hzのプライマリエイリアス



＜＜★The cursor window in Figure 5 shows that the sample alias noise is located at approximately 131.7 kHz and 135.86 kHz, and has a magnitude of –33.6 dB and –35.2 dB, respectively.＞＞Figure 5のカーソルウィンドウは、サンプルエイリアスノイズが約131.7 kHzと135.86 kHzにあり、振幅がそれぞれ–33.6 dBと–35.2 dBであることを示しています。

Figure 6. ＜＜★1300 Hz Output＞＞1300 Hz出力

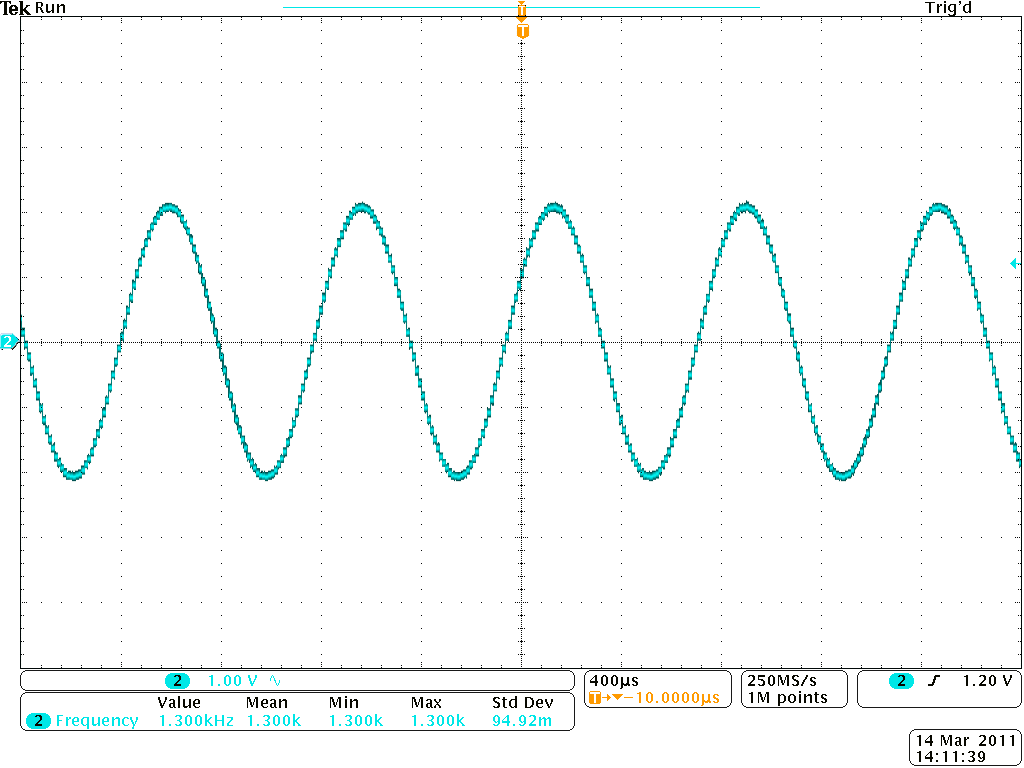
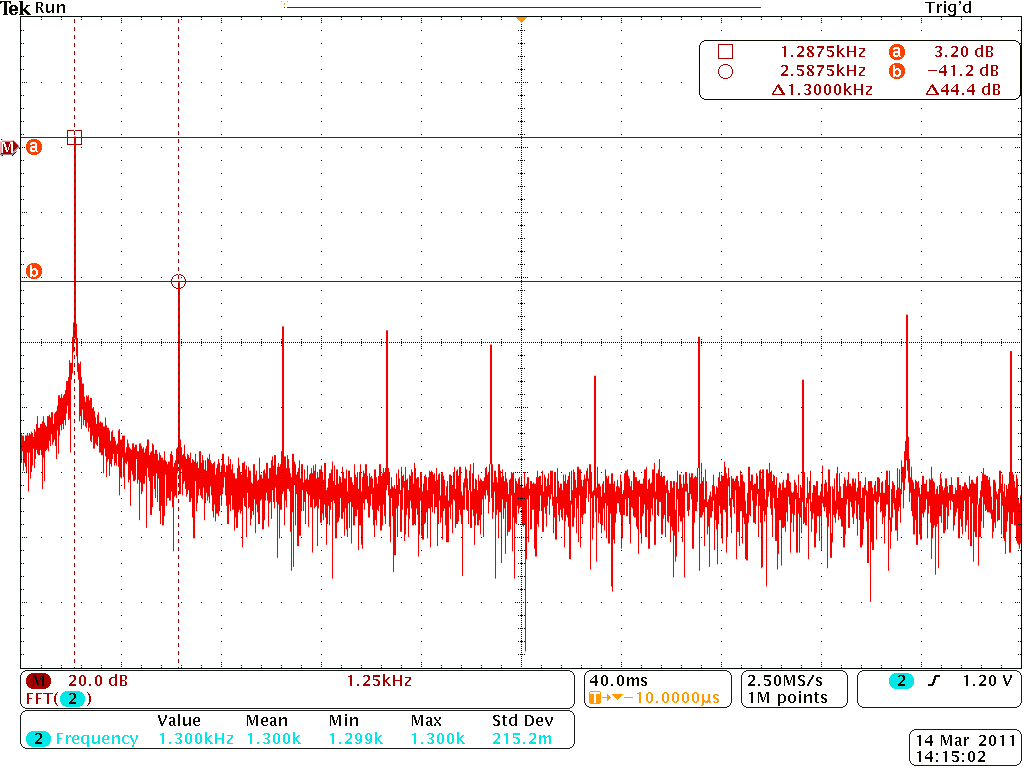
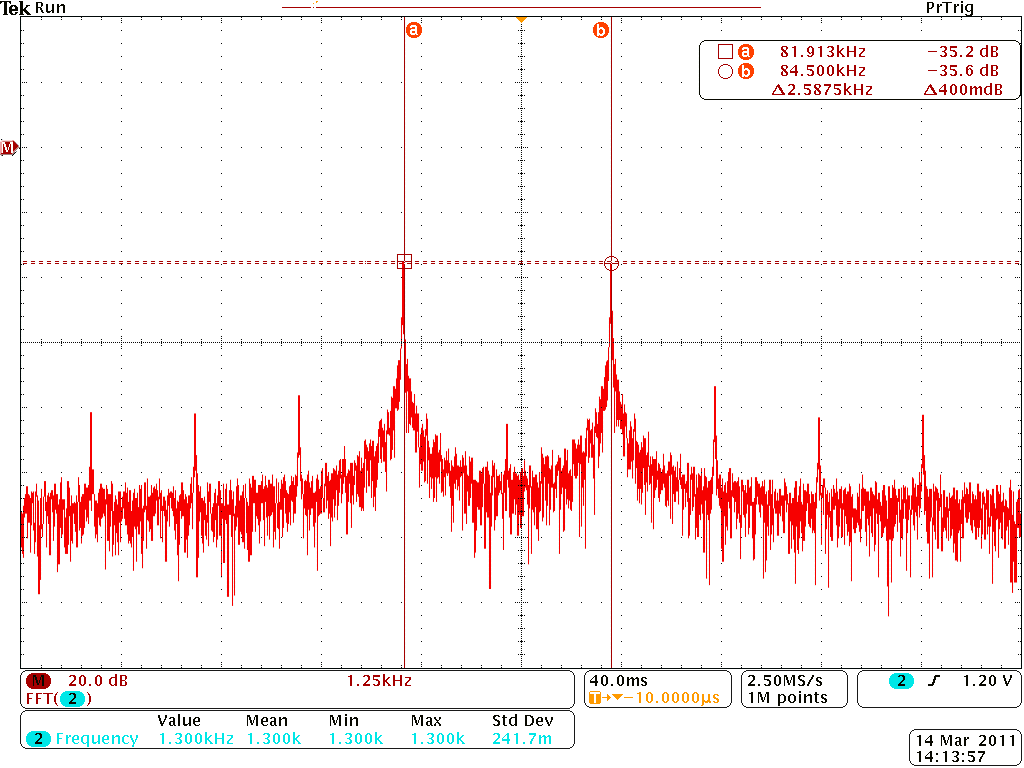


Figure 7. 1300 Hz FFT



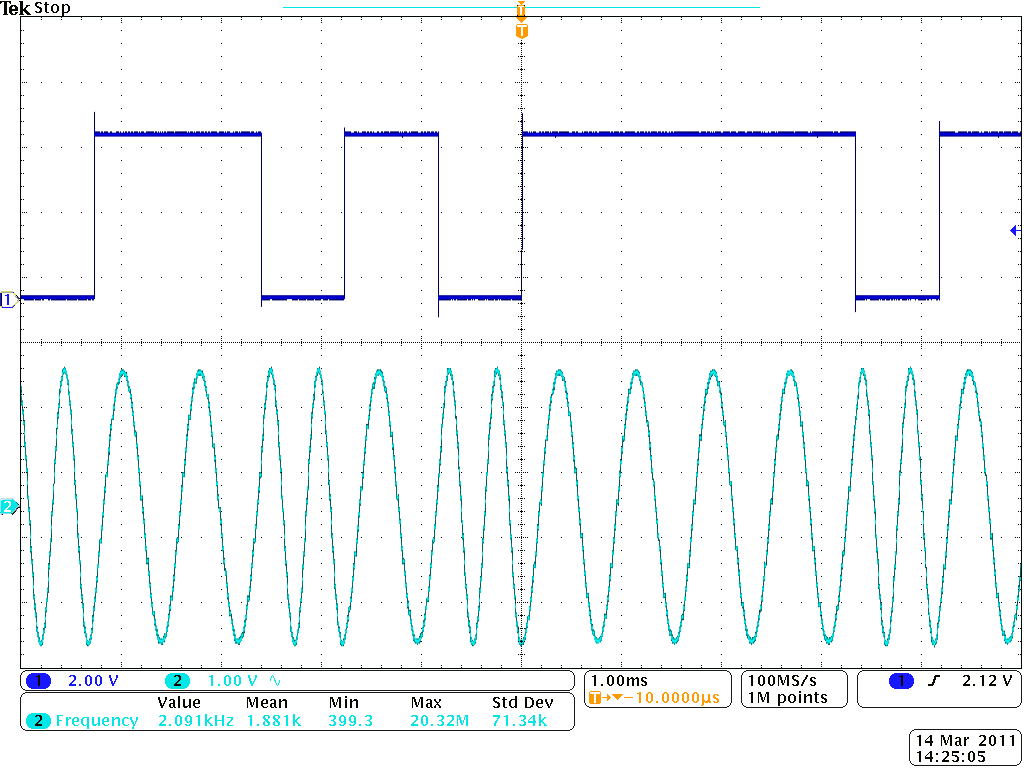
＜＜★［2］The cursor window in Figure 7 shows the fundamental frequency at 3.2 dB and the second harmonic at –41.2 dB, which is approximately 44 dB down.＞＞Figure 7のカーソルウィンドウは、3.2 dBの基本周波数と–41.2 dBの2次高調波を示しています。これは約44 dB低くなっています。

Figure 8. ＜＜★1300 Hz Primary Aliases＞＞1300 Hzプライマリエイリアス



＜＜★The cursor window in Figure 8 shows that the sample alias noise is located at approximately 81.9 kHz and 84.9 kHz, and has a magnitude of –35.2 dB and –35.6 dB, respectively.＞＞Figure 8のカーソルウィンドウは、サンプルエイリアスノイズが約81.9 kHzと84.9 kHzにあり、振幅がそれぞれ–35.2 dBと–35.6 dBであることを示しています。

Figure 9. ＜＜★Combined Output＞＞結合出力

  
＜＜★The FFTs show that the output waveforms are clean.＞＞FFTは、出力波形がクリーンであることを示しています。＜＜★［2］The second harmonic and higher are at least 40 dB below the fundamental.＞＞2次およびそれ以上の高調波は、基本波より少なくとも40 dB低くなります。＜＜★The combined waveform is also clean with no phase errors at bit transitions.＞＞結合された波形もきれいで、ビット遷移時に位相エラーがありません。＜＜★With PSoC 3 and PSoC 5LP devices, it is easy to create an FSK modulator that requires no CPU interaction.＞＞PSoC 3およびPSoC 5LPデバイスを使用すると、CPUの相互作用を必要としないFSK変調器を簡単に作成できます。

## ＜＜★Generating Different Frequencies＞＞異なる周波数を生成する

＜＜★For your application, 1300 Hz and 2100 Hz may not be the frequencies that you need.＞＞アプリケーションによっては、1300 Hzと2100 Hzが必要な周波数ではない場合があります。＜＜★This section describes how you can achieve different frequencies.＞＞このセクションでは、さまざまな周波数を実現する方法について説明します。

＜＜★The FSK output frequency can be calculated by dividing the sample clock frequency by the size of the sine table, as shown in Equation 1.＞＞式1に示すように、FSK出力周波数は、サンプルクロック周波数を正弦波テーブルのサイズで割ることによって計算できます。

 式1

＜＜★N is the number of elements in the sine table. The sample clock is the clock used to drive the DRQ of the DMA.＞＞Nはサインテーブルの要素数です。サンプルクロックは、DMAのDRQを駆動するために使用されるクロックです。

＜＜★To generate a digital 0 in the previous example, a 134.4-kHz sample clock was used with a 64-element sine table.＞＞前の例でデジタル0を生成するために、64要素の正弦表で134.4 kHzのサンプルクロックが使用されました。＜＜★Using Equation 1, the output frequency can be calculated as:＞＞式1を使用すると、出力周波数は次のように計算できます。



＜＜★For a digital 1, an 83.2-kHz sample clock was used with the same 64-element sine table.＞＞デジタル1では、83.2 kHzのサンプルクロックが、同じ64要素の正弦波テーブルで使用されました。＜＜★Again, using Equation 1, the output frequency can be calculated:＞＞ここでも、式1を使用して、出力周波数を計算できます。



＜＜★To increase the output frequency, the sample clock can be increased, or the number of sine table elements can be decreased.＞＞出力周波数を上げるには、サンプルクロックを増やすか、サインテーブルエレメントの数を減らします。

＜＜★To decrease the output frequency, the sample clock can be decreased, or the number of sine table elements can be increased.＞＞出力周波数を下げるには、サンプルクロックを減らすか、サインテーブルエレメントの数を増やします。

＜＜★The question that often arises at this point is how fast can I get my FSK output with this method? This method only works with low frequencies; it will not work for MHz signals.＞＞この時点でよくある質問は、この方法でFSK出力をどれだけ速く取得できるかということです。この方法は低周波数でのみ機能します。MHz信号では機能しません。＜＜★The following paragraphs discuss how to determine the maximum output frequency.＞＞次の段落では、最大出力周波数を決定する方法について説明します。

＜＜★A VDAC configured in the 0 V to 4.048 V range has a maximum sample rate of 250 ksps; using a 64-bit sine table, the maximum output frequency that can be achieved is 3.9 kHz.＞＞0 V〜4.048 Vの範囲で構成されたVDACの最大サンプルレートは250 kspsです。64ビットの正弦波テーブルを使用すると、達成可能な最大出力周波数は3.9 kHzです。＜＜★A VDAC configured in the 0 V to 1.024 V range has a maximum sample rate of 1000 ksps; using a 64-bit sine table, the maximum output frequency that can be achieved is 15.625 kHz.＞＞0 V〜1.024 Vの範囲で構成されたVDACの最大サンプルレートは1000 kspsです。64ビットの正弦表を使用すると、達成できる最大出力周波数は15.625 kHzです。

＜＜★If an IDAC is used instead of a VDAC, the maximum sample rate is 8000 ksps.＞＞VDACの代わりにIDACを使用する場合、最大サンプルレートは8000 kspsです。＜＜★This allows for a maximum output frequency of 125 kHz with a 64-element sine table.＞＞これにより、64要素の正弦波テーブルで125 kHzの最大出力周波数が可能になります。

＜＜★If an IDAC is used, the current needs to be routed to an external resistor to convert the current to a voltage. See Figure 10.＞＞IDACを使用する場合、電流を外部抵抗にルーティングして、電流を電圧に変換する必要があります。Figure 10を参照してください。

Figure . ＜＜★Converting an IDAC to a VDAC＞＞IDACをVDACに変換する



＜＜★Table 1 shows the maximum output frequencies for different sized sine tables.＞＞Table 1は、さまざまなサイズの正弦波テーブルの最大出力周波数を示しています。

Table 1. ＜＜★Sine Table Size versus Maximum Output Frequency＞＞サインテーブルサイズと最大出力周波数

|  |  |  |  |
| --- | --- | --- | --- |
| ＜＜★Sine Table Elements＞＞サインテーブル要素 | IDAC | 1-V VDAC | 4-V VDAC |
| 256 | 31.25 kHz | 3.9 kHz | 976 Hz |
| 128 | 62.5 kHz | 7.8 kHz | 1.95 kHz |
| 64 | 125 kHz | 15.6 kHz | 3.9 kHz |
| 32 | 250 kHz | 31.25 kHz | 7.8 kHz |
| 16 | 500 kHz | 62.5 kHz | 15.6 kHz |

＜＜★Decreasing the number of elements in the sine table has the disadvantage of increasing the harmonic and sampling noise.＞＞サインテーブルの要素数を減らすと、高調波ノイズとサンプリングノイズが増えるという欠点があります。＜＜★When designing an FSK modulator, the tradeoff between speed and noise needs to be considered.＞＞FSK変調器を設計するときは、速度とノイズのトレードオフを考慮する必要があります。＜＜★The FFTs shown previously give a good approximation of the noise present in this system for a 64-element sine table.＞＞前に示したFFTは、64要素の正弦表に対してこのシステムに存在するノイズの適切な近似を提供します。

# ＜＜★FSK Demodulator＞＞FSK復調器

＜＜★The FSK demodulator takes the analog input signal and demodulates it into the appropriate sequence of digital 1s and 0s.＞＞FSK復調器は、アナログ入力信号を受け取り、デジタル1と0の適切なシーケンスに復調します。＜＜★The block diagram for the FSK demodulator is shown in Figure 11.＞＞FSK復調器のブロック図をFigure 11に示します。

Figure 11. ＜＜★Demodulator Block Diagram＞＞復調器のブロック図



## ＜＜★Band-pass Filter＞＞バンドパスフィルター

＜＜★The band-pass filter (BPF) is designed to allow 1300 Hz and 2100 Hz signals to pass through with little attenuation (around 1 dB), and everything else to be attenuated.＞＞バンドパスフィルター（BPF）は、1300 Hzと2100 Hzの信号をほとんど減衰せず（約1 dB）に通過させ、それ以外はすべて減衰させるように設計されています。＜＜★The purpose of this filter is to remove any out-of-band noise, thus making the demodulation processes easier.＞＞このフィルターの目的は、帯域外ノイズを除去して、復調プロセスを容易にすることです。

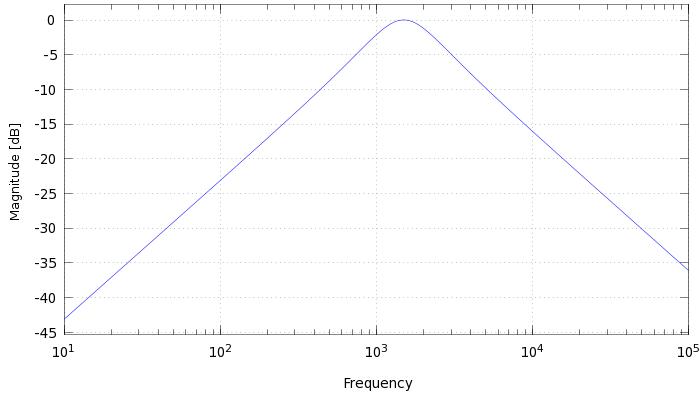
＜＜★This filter is designed using standard filter design techniques. It has a center frequency close to 1500 Hz. The bandwidth is set to 1570 Hz to achieve less than 1 dB of attenuation at the modulating frequencies.＞＞このフィルターは、標準のフィルター設計手法を使用して設計されています。中心周波数は1500 Hzに近い。帯域幅は1570 Hzに設定され、変調周波数で1 dB未満の減衰を実現します。＜＜★This filter is designed for a gain of 1 V/V.＞＞このフィルターは、1 V / Vのゲイン用に設計されています。＜＜★［1］The circuit for the BPF is shown in Figure 12; derivation of component values is a standard design problem, see [Appendix A](#_Appendix_A:) for more detail on how the filter was designed or if you need to design a filter for different frequencies.＞＞BPFの回路をFigure 12に示します。コンポーネント値の導出は標準の設計問題です。フィルターの設計方法の詳細、または異なる周波数用のフィルターを設計する必要があるかどうかについては、[付録A](#_Appendix_A:)を参照してください。

＜＜★In this project, Vdda/2 is provided internally by the PSoC.＞＞このプロジェクトでは、Vdda / 2はPSoCによって内部的に提供されます。＜＜★It is buffered through another opamp and routed to all the locations that need a Vdda/2 connection.＞＞別のオペアンプを介してバッファリングされ、Vdda / 2接続を必要とするすべての場所にルーティングされます。＜＜★Ensure you have a strong Vdda/2 source for the project to work correctly.＞＞プロジェクトが正しく機能するための強力なVdda / 2ソースがあることを確認してください。

Figure 12. ＜＜★BPF Circuit＞＞BPF回路



Figure 13. ＜＜★BPF Bode Plot＞＞BPFボード線図



＜＜★This filter is meant to be an example to the designer.＞＞このフィルターは、設計者への例を示すためのものです。＜＜★The filter should be modified appropriately for the target design.＞＞フィルターは、ターゲットデザインに合わせて適切に変更する必要があります。＜＜★Signaling frequencies, signal amplitude, and out-of-band noise all need to be considered when designing this front-end BPF.＞＞このフロントエンドBPFを設計するときは、信号周波数、信号振幅、および帯域外ノイズをすべて考慮する必要があります。＜＜★The topology in Figure 12 can be modified for increased gain, as well as different center frequencies and bandwidths.＞＞Figure 12のトポロジーは、ゲインを増加させるだけでなく、さまざまな中心周波数と帯域幅に合わせて変更できます。＜＜★If more out-of-band attenuation is needed, a higher-order filter must be constructed out of multiple PSoC opamps.＞＞より多くの帯域外減衰が必要な場合は、複数のPSoCオペアンプから高次のフィルターを構築する必要があります。

## ＜＜★［0］Zero-Crossing Detector＞＞ゼロ交差検出器

＜＜★After the waveforms are filtered, it is important to digitize them for use in the correlator.＞＞波形がフィルタリングされた後、相関器で使用するために波形をデジタル化することが重要です。＜＜★［01］This is done by constructing a simple zero-crossing detector.＞＞これは、単純なゼロ交差検出器を構築することによって行われます。＜＜★This detector can be created with a comparator set to trip at VDD/2.＞＞この検出器は、コンパレータをVDD / 2で作動するように設定して作成できます。

Figure 14. ＜＜★［0］Zero-Cross Detector＞＞ゼロクロス検出器



## ＜＜★Correlator＞＞相関器

＜＜★The theory for the correlator is explained in application note [AN2336](http://www.cypress.com/?rID=2735).＞＞コリレータの理論は、アプリケーションノート[AN2336](http://www.cypress.com/?rID=2735)で説明されています。＜＜★［0］The basic function of the correlator is to delay the digital signal out of the zero-crossing detector, and then XOR that signal with the non-delayed signal.＞＞相関器の基本的な機能は、デジタル信号をゼロ交差検出器から遅延させ、その信号を遅延されていない信号とXORすることです。＜＜★［1］The result is that one frequency ends up with a high duty cycle, while the other frequency has a low duty cycle at the output of the XOR gate.＞＞その結果、一方の周波数は高いデューティサイクルで終了し、もう一方の周波数はXORゲートの出力で低いデューティサイクルになります。＜＜★Figure 15 and Figure 16 are scope captures at the output of the XOR gate for the two different frequencies.＞＞Figure 15とFigure 16は、2つの異なる周波数のXORゲートの出力でのスコープキャプチャです。

Figure 15. ＜＜★Correlator Output at 2100 Hz＞＞2100 Hzでの相関器出力

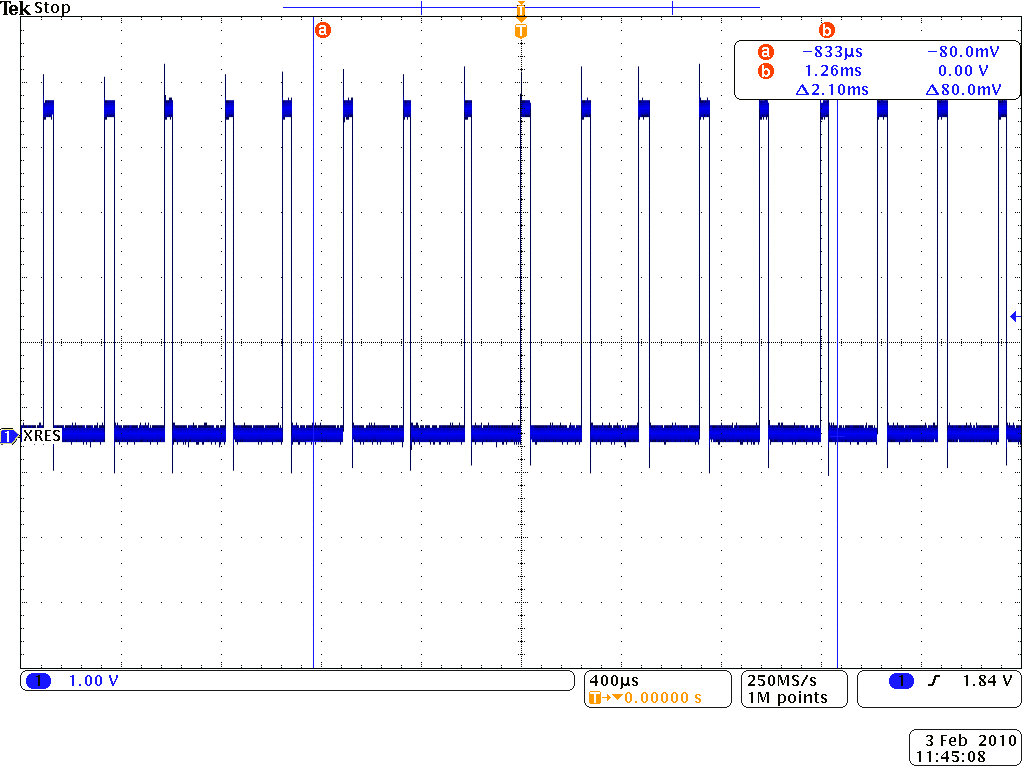
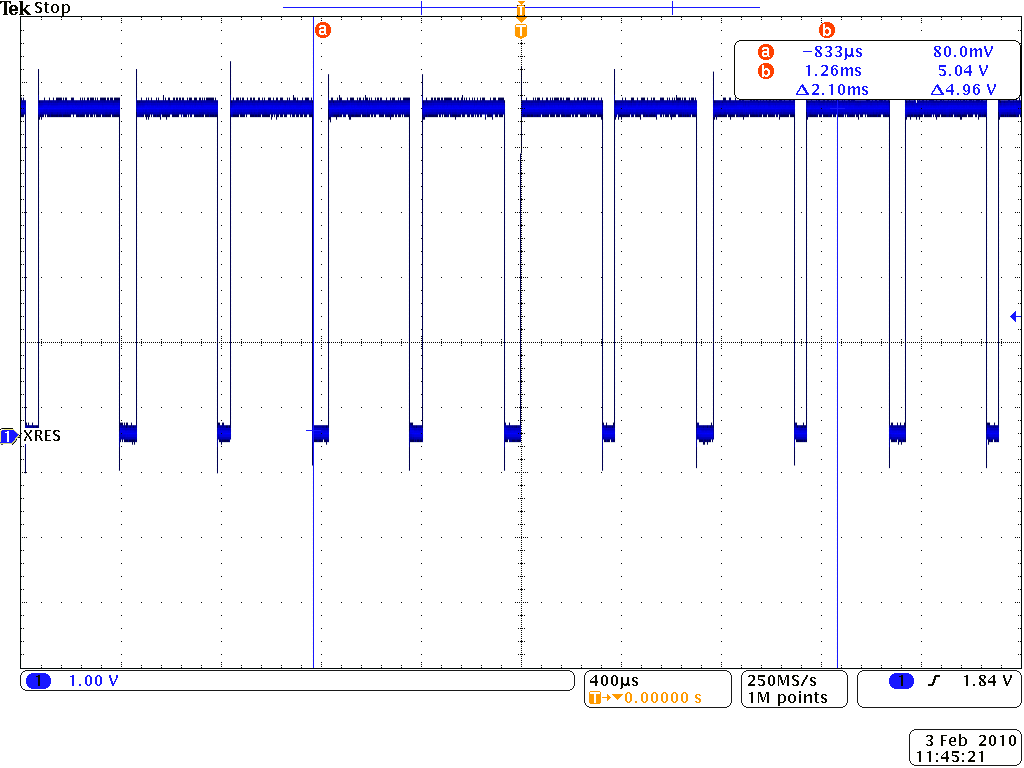


Figure 16. ＜＜★Correlator Output at 1300 Hz＞＞1300 Hzでの相関器出力



＜＜★Figure 15 and Figure 16 show that the output of the correlator has a high duty cycle at 1300 Hz, while the output has a low duty cycle at 2100 Hz.＞＞Figure 15およびFigure 16は、相関器の出力のデューティサイクルが1300 Hzで高く、出力のデューティサイクルが2100 Hzで低いことを示しています。

＜＜★From AN2336, it is calculated that a delay of approximately 450 μs is needed.＞＞[AN2336](http://www.cypress.com/?rID=2735)から、約450μsの遅延が必要であると計算されています。＜＜★This delay is constructed out of a shift register; see Figure 17.＞＞この遅延は、シフトレジスタから構成されます。Figure 17を参照してください。＜＜★The shift register is configured for 24 bits and the input clock is 53.5 kHz or 18.7 μs; 18.7 μs multiplied by 24 is approximately 448 μs.＞＞シフトレジスタは24ビット用に構成され、入力クロックは53.5 kHzまたは18.7μsです。18.7μsに24を掛けると、約448μsになります。

Figure 17. ＜＜★PSoC 3 Correlator＞＞PSoC 3コリレータ



＜＜★If signal frequencies other than 2100 Hz and 1300 Hz are used, the delay must be adjusted accordingly.＞＞2100 Hzおよび1300 Hz以外の信号周波数を使用する場合は、それに応じて遅延を調整する必要があります。＜＜★The delay can be calculated by using Equation 2.＞＞遅延は、式2を使用して計算できます。＜＜★For a detailed derivation of Equation 2, see [AN2336](http://www.cypress.com/?rID=2735).＞＞式2の詳細な導出については、[AN2336](http://www.cypress.com/?rID=2735)を参照してください。

 式2

＜＜★fL is the lower frequency, fH is the high frequency, and d is the delay in seconds.＞＞fLは低周波数、fHは高周波数、dは遅延（秒単位）です。＜＜★You can enter this equation into Microsoft Excel to find the maximum difference.＞＞この式をMicrosoft Excelに入力して、最大の差を見つけることができます。

＜＜★Typically, the delay with the largest possible difference simplifies the filtering and detection process.＞＞通常、可能な限り最大の差がある遅延は、フィルタリングと検出のプロセスを簡素化します。

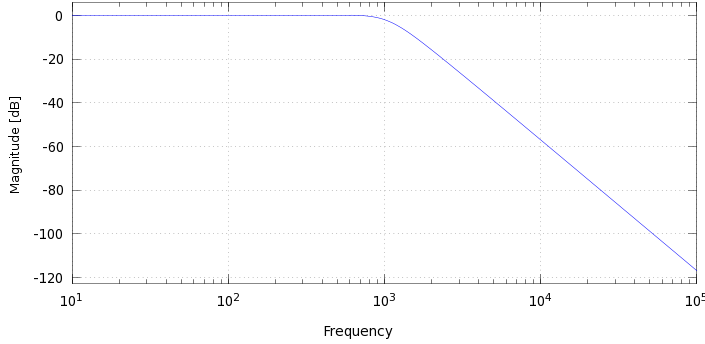
## LPF

＜＜★The best way to remove the glitches in Figure 15 and Figure 16 is to low-pass filter the data out of the correlator.＞＞Figure 15とFigure 16でグリッチを除去する最良の方法は、相関器からデータをローパスフィルター処理することです。＜＜★For this design, a three-pole filter with a cut-off at 1.1 kHz is designed.＞＞この設計では、1.1 kHzにカットオフを持つ3極フィルターが設計されています。＜＜★The LPF is designed so that the rise time is sufficiently fast for the signaling constraints.＞＞LPFは、信号の制約に対して立ち上がり時間が十分に速くなるように設計されています。＜＜★For example, at 1200 baud, the bit time is approximately 800 μs.＞＞たとえば、1200ボーでは、ビット時間は約800μsです。＜＜★Thus, the rise time of the LPF must be faster to avoid errors.＞＞したがって、LPFの立ち上がり時間は、エラーを回避するために速くなければなりません。＜＜★This filter has a rise time of approximately 450 μs.＞＞このフィルターの立ち上がり時間は約450μsです。

Figure 18. ＜＜★Low-Pass Filter＞＞ローパスフィルタ



Figure 20. ＜＜★LPF Bode Plot＞＞LPFボード線図



＜＜★［1］The LPF is constructed out of external components and a PSoC PGA configured in unity gain mode.＞＞LPFは、外部コンポーネントとユニティゲインモードで構成されたPSoC PGAで構成されています。＜＜★The PGA is used to save one of the valuable opamps for other uses in your application.＞＞PGAは、貴重なオペアンプの1つをアプリケーションの他の用途のために保存するために使用されます。

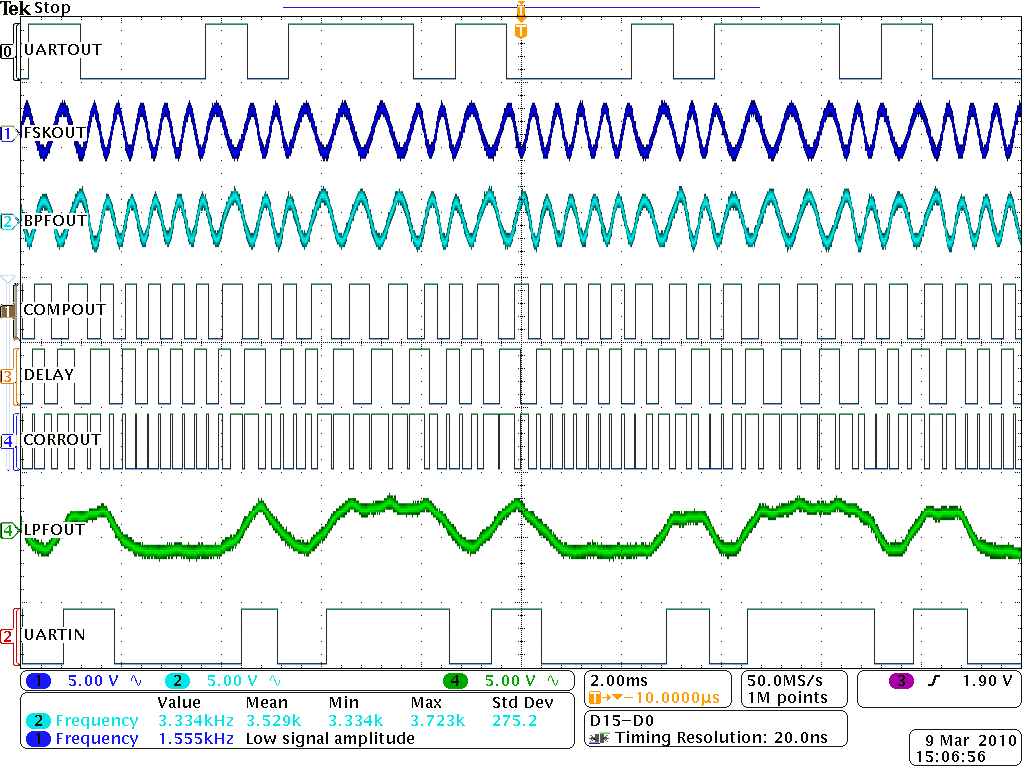
＜＜★The LPF must be designed so that it sufficiently removes all glitches, but with a rise time sufficiently shorter than the bit time, to avoid bit errors.＞＞LPFは、すべてのグリッチを十分に除去するように設計する必要がありますが、立ち上がり時間はビット時間よりも十分に短く、ビットエラーを回避します。

## ＜＜★Digitize＞＞デジタル化

＜＜★The output of the LPF is run through a comparator to digitize the signal.＞＞LPFの出力は、信号をデジタル化するためにコンパレータを介して実行されます。＜＜★This digitized signal can then be fed into a UART or other digital circuitry.＞＞このデジタル化された信号は、UARTまたは他のデジタル回路に供給されます。＜＜★In some designs, it may be necessary to add hysteresis to this comparator.＞＞設計によっては、このコンパレータにヒステリシスを追加する必要がある場合があります。＜＜★This becomes necessary in systems with faster baud rates and higher signal noise.＞＞これは、ボーレートが高速で信号ノイズが大きいシステムで必要になります。＜＜★The PSoC comparator has a built-in 10 mV of hysteresis.＞＞PSoCコンパレータには、10 mVのヒステリシスが組み込まれています。＜＜★AN2108 describes how to achieve wider hysteresis＞＞[AN2108](http://www.cypress.com/?rID=2780)は、より広いヒステリシスを実現する方法を説明しています

# ＜＜★Receiving Results＞＞結果を受け取る

Figure 21. ＜＜★Putting It All Together＞＞すべてを一緒に入れて



＜＜★Figure 20 shows the signals received:＞＞Figure 20は、受信した信号を示しています。

* ＜＜★The top signal (UARTOUT) is the output of the TX UART.＞＞一番上の信号（UARTOUT）は、TX UARTの出力です。
* ＜＜★［2］The second signal (FSKOUT) is the analog output from the FSK modulator described earlier in this application note.＞＞2番目の信号（FSKOUT）は、このアプリケーションノートで前述したFSK変調器からのアナログ出力です。
* ＜＜★［3］The third signal (BPFOUT) is the output of the BPF; 2100 Hz has slightly more attenuation because of resistor tolerance issues.＞＞3番目の信号（BPFOUT）はBPFの出力です。2100 Hzは、抵抗の許容誤差の問題のため、わずかに減衰が大きくなります。
* ＜＜★［0］The fourth signal (COMPOUT) is the output of the first zero-crossing detector circuit.＞＞4番目の信号（COMPOUT）は、最初のゼロ交差検出回路の出力です。
* ＜＜★［5］The fifth signal (DELAY) is the output of the shift register, the delayed signal.＞＞5番目の信号（DELAY）は、遅延信号であるシフトレジスタの出力です。
* ＜＜★The sixth signal (CORROUT) is the XOR of the COMPOUT and DELAY signals; notice that with one input frequency the correlator output has a high duty cycle, while at the other frequency it has a low duty cycle.＞＞6番目の信号（CORROUT）は、COMPOUT信号とDELAY信号のXORです。1つの入力周波数では相関器の出力のデューティサイクルが高く、他の周波数ではデューティサイクルが低いことに注意してください。
* ＜＜★The seventh output (LPFOUT) is that of the LPF.＞＞7番目の出力（LPFOUT）は、LPFの出力です。
* ＜＜★The eighth signal (UARTIN) is the final output that goes to the receive UART.＞＞8番目の信号（UARTIN）は、受信UARTに送られる最後の出力です。＜＜★Notice that the eighth signal is essentially a delayed version of the first signal (UARTOUT).＞＞8番目の信号は基本的に最初の信号（UARTOUT）の遅延バージョンであることに注意してください。

# ＜＜★Design Modifications＞＞設計変更

＜＜★A bare essential FSK modulator and demodulator design has been shown.＞＞必要不可欠なFSK変調器と復調器の設計が示されています。＜＜★There are numerous opportunities for design refinements, including increasing gain, improving filter selectivity, and adapting to other operating frequencies.＞＞ゲインの増加、フィルターの選択性の向上、他の動作周波数への適応など、設計を改善する多くの機会があります。＜＜★The design principles presented are easily extended to these other applications.＞＞提示された設計原則は、これらの他のアプリケーションに簡単に拡張できます。

# ＜＜★Attached Projects＞＞付属プロジェクト

＜＜★There are two projects attached to this application note: FSK\_RX and FSK\_TX.＞＞このアプリケーションノートには、FSK\_RXとFSK\_TXの2つのプロジェクトが添付されています。＜＜★They are located in the same workspace.＞＞それらは同じワークスペースにあります。＜＜★The FSK\_TX project transmits the string “Hello World”.＞＞FSK\_TXプロジェクトは文字列「Hello World」を送信します。＜＜★［1］The FSK\_RX project receives that string and displays it on the LCD of one of the Cypress standard development boards (CY8CKIT-001, 030, or 050).＞＞FSK\_RXプロジェクトはその文字列を受け取り、サイプレス標準開発ボード（CY8CKIT-001、030、または050）のいずれかのLCDに表示します。＜＜★The FSK output of FSK\_TX is found on P0.1.＞＞FSK\_TXのFSK出力はP0.1にあります。

＜＜★The resistors and capacitors for the BPF and LPF must be added to the development board for the FSK\_RX project.＞＞BSKおよびLPFの抵抗とコンデンサは、FSK\_RXプロジェクトの開発ボードに追加する必要があります。＜＜★Follow the schematics shown in this document or in the project.＞＞このドキュメントまたはプロジェクトに示されている回路図に従ってください。

＜＜★Attach the FSK output on FSK\_TX board (P0.1) to the input of the BPF on the FSK\_RX board.＞＞FSK\_TXボード（P0.1）のFSK出力をFSK\_RXボードのBPFの入力に接続します。＜＜★In addition, attach the grounds of the two development boards together.＞＞さらに、2つの開発ボードのグラウンドを一緒に取り付けます。

＜＜★The projects attached to this application note are designed for both PSoC 3 and PSoC 5LP.＞＞このアプリケーションノートに添付されているプロジェクトは、PSoC 3とPSoC 5LPの両方向けに設計されています。＜＜★The default device is PSoC 3 (CY8C3866AXI-040).＞＞デフォルトのデバイスはPSoC 3（CY8C3866AXI-040）です。＜＜★To change to another device, including PSoC 5LP devices, go to **Project** > **Device Selector** and select the device.＞＞PSoC 5LPデバイスを含む別のデバイスに変更するには、**Project** > **Device Selector**に移動して、デバイスを選択します。＜＜★For further information, see the PSoC Creator Help article, Device Selector.＞＞詳細については、PSoC Creatorのヘルプ記事、デバイスセレクターを参照してください。

# まとめ

＜＜★This application note explains how to create an FSK modulator and demodulator in PSoC 3 and PSoC 5LP without any CPU interaction.＞＞このアプリケーションノートでは、CPUの相互作用なしでPSoC 3およびPSoC 5LPでFSK変調器および復調器を作成する方法について説明します。＜＜★Attached to this document are example projects that implement what is described.＞＞このドキュメントに添付されているのは、説明されている内容を実装するサンプルプロジェクトです。＜＜★［1］The appropriate external components must be provided for the projects to work.＞＞プロジェクトを機能させるには、適切な外部コンポーネントを提供する必要があります。

＜＜★With a proper understanding of the material presented in this application note, you can implement FSK in your design.＞＞このアプリケーションノートに記載されている内容を適切に理解すると、設計にFSKを実装できます。

＜＜★The advantage of using a PSoC device is that the CPU is not used, so it can be used to accomplish other tasks for your application.＞＞PSoCデバイスを使用する利点は、CPUが使用されないため、アプリケーションの他のタスクを実行するために使用できることです。＜＜★Several hardware resources are also available to be used for your application.＞＞アプリケーションで使用するために、いくつかのハードウェアリソースも利用できます。

# 著者について

|  |  |
| --- | --- |
| **名前:** | Todd Dust |
| **役職:** | ＜＜★Applications Engineer Staff＞＞アプリケーションエンジニアスタッフ |
| **経歴:** | BSEE, Seattle Pacific University |
| **連絡先:** | [tdu@cypress.com](mailto:tdu@cypress.com) |

# 付録A

＜＜★In the main body of the application note, we did some hand waving when it came to deriving the requirements for the band-pass filter.＞＞アプリケーションノートの本文では、バンドパスフィルターの要件を導出する際に手を振っていました。＜＜★As designing the BPF is a more advanced topic and not the focus of the application note, it is discussed in this Appendix.＞＞BPFの設計はより高度なトピックであり、アプリケーションノートの焦点では​​ないため、この付録で説明します。

＜＜★Remember that the signaling frequencies are 1300 Hz and 2100 Hz.＞＞信号周波数は1300 Hzと2100 Hzであることに注意してください。＜＜★The goal of this filter is to have very little attenuation at these frequencies, ~−1 dB if possible, and as much attenuation on any out-of-band signals.＞＞このフィルターの目的は、これらの周波数での減衰を非常に少なくし、可能であれば〜1 dBにし、帯域外信号でできるだけ減衰させることです。＜＜★The filter is designed following these steps.＞＞フィルターは、次の手順に従って設計されています。

＜＜★The first step is to find the corner frequency of a low-pass filter (LPF) with a –1 dB point at 800 Hz; this number will give the bandwidth of the BPF.＞＞最初のステップは、800 Hzで–1 dBポイントのローパスフィルター（LPF）のコーナー周波数を見つけることです。この数は、BPFの帯域幅を提供します。＜＜★800 Hz is the difference between 2100 Hz and 1300 Hz.＞＞800 Hzは2100 Hzと1300 Hzの差です。

＜＜★［1］Consider the transfer function of a first order LPF.＞＞1次LPFの伝達関数を考えます。



＜＜★A gain of –1 dB is required, so insert that into the equation.＞＞-1 dBのゲインが必要なので、それを方程式に挿入します。



＜＜★Next, convert –1 dB into units we can work with.＞＞次に、1 dBを使用できる単位に変換します。



＜＜★Now, we can simplify this equation to determine.＞＞これで、この方程式を簡略化して決定できます。

＜＜★After all the algebra and re-arranging, the equation looks as follows:＞＞すべての代数と再配置を終えると、方程式は次のようになります。



＜＜★Enter 800 for＞＞に800を入力してください









＜＜★The bandwidth of the filter is 1572.2 Hz.＞＞フィルターの帯域幅は1572.2 Hzです。＜＜★The next step is to determine the center frequency of the filter.＞＞次のステップは、フィルターの中心周波数を決定することです。＜＜★To do this, compute the following equation:＞＞これを行うには、次の方程式を計算します。



ここに

＜＜★-is the center frequency＞＞-は中心周波数です

＜＜★-is the upper frequency＞＞-は上限周波数です

＜＜★-is the lower frequency.＞＞-は低い周波数です。

＜＜★We already calculated BW to be 1572.2 Hz; we need to now determine the upper and lower frequencies of the band.＞＞BWはすでに1572.2 Hzと計算されています。次に、帯域の上限周波数と下限周波数を決定する必要があります。＜＜★The upper and lower frequencies are the –3 dB points of the filter.＞＞上下の周波数は、フィルターの–3 dBポイントです。＜＜★To find these frequencies, subtract 800 Hz (the difference between 2100 Hz and 1300 Hz) from the BW of 1572.2 Hz; this results in 772.2 Hz.＞＞これらの周波数を見つけるには、1572.2 Hzの帯域幅から800 Hz（2100 Hzと1300 Hzの差）を差し引きます。これにより772.2 Hzになります。＜＜★Divide this by two and we get 386.1 Hz to add to either side of the filter.＞＞これを2で除算すると、386.1 Hzになり、フィルターの両側に追加されます。



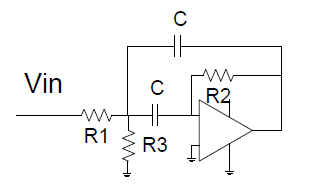




＜＜★Now we know the bandwidth of the filter and the center frequency.＞＞これで、フィルターの帯域幅と中心周波数がわかりました。＜＜★The Q of the filter can be calculated with the following equation:＞＞フィルターのQは、次の式で計算できます。



＜＜★A multi-feedback (MFB) BPF is used here; the generic implementation of a MFB BPF looks as follows:＞＞ここではマルチフィードバック（MFB）BPFが使用されています。MFB BPFの一般的な実装は次のようになります。



＜＜★Following are the equations for the resistor values:＞＞抵抗値の式は次のとおりです。









＜＜★Now, we need to choose a value for C. For this project, the value selected is 100 nf.＞＞ここで、Cの値を選択する必要があります。このプロジェクトでは、選択した値は100 nfです。＜＜★C needs to be chosen such that the values for the resistors are realistic and are not too big or too small.＞＞Cは、抵抗の値が現実的で、大きすぎたり小さすぎたりしないように選択する必要があります。＜＜★Too big will mean more noise, too small will mean more power.＞＞大きすぎるとノイズが多くなり、小さすぎるとパワーが大きくなります。

＜＜★Using 100 nf and all of the other information, the resistors were calculated to be the following values:＞＞100 nfおよびその他すべての情報を使用して、抵抗は次の値になるように計算されました。







＜＜★If you need to use different frequencies for your design, then follow the steps in this section; however, change the frequency values.＞＞デザインに異なる周波数を使用する必要がある場合は、このセクションの手順に従ってください。ただし、頻度の値は変更してください。

改訂履歴

**＜＜★Document Title:** **AN60594 - PSoC® 3 and PSoC 5LP: Low-Frequency FSK Modulation and Demodulation＞＞ドキュメントタイトル：AN60594-PSoC® 3およびPSoC 5LP：低周波数FSK変調および復調**

**＜＜★Document Number: 001-60594＞＞文書番号：001-60594**

|  |  |  |  |
| --- | --- | --- | --- |
| 版 | ECN | 発行日 | 変更内容 |
| \*\* |  |  |  |

セールス, ソリューションおよび法律情報

**ワールドワイドな販売と設計サポート**

サイプレスは、事業所、ソリューションセンター、メーカー代理店、および販売代理店の世界的なネットワークを保持しています。お客様の最寄りのオフィスについては、[サイプレスのロケーションページ](http://www.cypress.com/?id=1062)をご覧ください。

**製品**

Arm® Cortex® Microcontrollers [cypress.com/arm](http://www.cypress.com/products/32-bit-arm-cortex-mcus)

車載用 [cypress.com/automotive](http://www.cypress.com/applications/automotive-solutions)

クロック&バッファ [cypress.com/clocks](http://www.cypress.com/products/clocks-buffers)

インターフェース [cypress.com/interface](http://www.cypress.com/products/interface)

IoT (モノのインターネット) [cypress.com/iot](http://www.cypress.com/internet-things-iot)

メモリ [cypress.com/memory](http://www.cypress.com/products/memory-products)

マイクロコントローラ [cypress.com/mcu](http://www.cypress.com/mcu)

PSoC [cypress.com/psoc](http://www.cypress.com/psoc/)

電源用IC [cypress.com/pmic](http://www.cypress.com/products/power-management)

タッチセンシング [cypress.com/touch](http://www.cypress.com/products/touch-sensing)

USBコントローラー [cypress.com/usb](http://www.cypress.com/products/usb-controllers)

ワイヤレス [cypress.com/wireless](http://www.cypress.com/products/wireless-connectivity)

**PSoC®ソリューション**

[PSoC 1](http://www.cypress.com/products/psoc-1) | [PSoC 3](http://www.cypress.com/products/psoc-3) | [PSoC 4](http://www.cypress.com/products/psoc-4) | [PSoC 5LP](http://www.cypress.com/products/32-bit-arm-cortex-m3-psoc-5lp) | [PSoC 6 MCU](http://cypress.com/psoc6)

**サイプレス開発者コミュニティ**

[コミュニティ](https://community.cypress.com/welcome) | [サンプルコード](http://www.cypress.com/cypressgithub) | [Projects](http://www.cypress.com/projects) | [ビデオ](http://www.cypress.com/video-library) | [ブログ](http://www.cypress.com/blog) | [トレーニング](http://www.cypress.com/training) | [Components](http://www.cypress.com/cdc/community-components)

**テクニカルサポート**

[cypress.com/support](http://www.cypress.com/support)

|  |  |
| --- | --- |
|  | Cypress Semiconductor  An Infineon Technologies Company  198 Champion Court  San Jose, CA 95134-1709 |

© Cypress Semiconductor Corporation, 2010-2020. 本書面は, Cypress Semiconductor Corporation及びSpansion LLCを含むその子会社 (以下「Cypress」という。) に帰属する財産である。本書面 (本書面に含まれ又は言及されているあらゆるソフトウェア若しくはファームウェア (以下「本ソフトウェア」という。) を含む) は, アメリカ合衆国及び世界のその他の国における知的財産法令及び条約に基づきCypressが所有する。Cypressはこれらの法令及び条約に基づく全ての権利を留保し, 本段落で特に記載されているものを除き, その特許権, 著作権, 商標権又はその他の知的財産権のライセンスを一切許諾しない。本ソフトウェアにライセンス契約書が伴っておらず, かつCypressとの間で別途本ソフトウェアの使用方法を定める書面による合意がない場合, Cypressは, (1) 本ソフトウェアの著作権に基づき, (a) ソースコード形式で提供されている本ソフトウェアについて, Cypressハードウェア製品と共に用いるためにのみ, かつ組織内部でのみ, 本ソフトウェアの修正及び複製を行うこと, 並びに (b) Cypressのハードウェア製品ユニットに用いるためにのみ, (直接又は再販売者及び販売代理店を介して間接のいずれかで) 本ソフトウェアをバイナリーコード形式で外部エンドユーザーに配布すること, 並びに (2) 本ソフトウェア (Cypressにより提供され, 修正がなされていないもの) が抵触するCypressの特許権のクレームに基づき, Cypressハードウェア製品と共に用いるためにのみ, 本ソフトウェアの作成, 利用, 配布及び輸入を行うことについての非独占的で譲渡不能な一身専属的ライセンス (サブライセンスの権利を除く) を付与する。本ソフトウェアのその他の使用, 複製, 修正, 変換又はコンパイルを禁止する。

適用される法律により許される範囲内で, Cypressは, 本書面又はいかなる本ソフトウェア若しくはこれに伴うハードウェアに関しても, 明示又は黙示をとわず, いかなる保証 (商品性及び特定の目的への適合性の黙示の保証を含むがこれらに限られない) も行わない。いかなるコンピューティングデバイスも絶対に安全ということはない。従って, Cypressのハードウェアまたはソフトウェア製品に講じられたセキュリティ対策にもかかわらず, Cypressは, Cypress製品への権限のないアクセスまたは使用といったセキュリティ違反から生じる一切の責任を負わない。加えて, 本書面に記載された製品には, エラッタと呼ばれる設計上の欠陥またはエラーが含まれている可能性があり, 公表された仕様とは異なる動作をする場合がある。適用される法律により許される範囲内で, Cypressは, 別途通知することなく, 本書面を変更する権利を留保する。Cypressは, 本書面に記載のある, いかなる製品若しくは回路の適用又は使用から生じる一切の責任を負わない。本書面で提供されたあらゆる情報 (あらゆるサンプルデザイン情報又はプログラムコードを含む) は, 参照目的のためのみに提供されたものである。この情報で構成するあらゆるアプリケーション及びその結果としてのあらゆる製品の機能性及び安全性を適切に設計, プログラム, かつテストすることは, 本書面のユーザーの責任において行われるものとする。Cypress製品は, 兵器, 兵器システム, 原子力施設, 生命維持装置若しくは生命維持システム, 蘇生用の設備及び外科的移植を含むその他の医療機器若しくは医療システム, 汚染管理若しくは有害物質管理の運用のために設計され若しくは意図されたシステムの重要な構成部分としての使用, 又は装置若しくはシステムの不具合が人身傷害, 死亡若しくは物的損害を生じさせるようなその他の使用 (以下「本目的外使用」という。) のためには設計, 意図又は承認されていない。重要な構成部分とは, それの不具合が装置若しくはシステムの不具合を生じさせるか又はその安全性若しくは実効性に影響すると合理的に予想できるような装置若しくはシステムのあらゆる構成部分をいう。Cypress製品のあらゆる本目的外使用から生じ, 若しくは本目的外使用に関連するいかなる請求, 損害又はその他の責任についても, Cypressはその全部又は一部をとわず一切の責任を負わず, かつCypressはそれら一切から本書により免除される。CypressはCypress製品の本目的外使用から生じ又は本目的外使用に関連するあらゆる請求, 費用, 損害及びその他の責任 (人身傷害又は死亡に基づく請求を含む) から免責補償される。

Cypress, Cypressのロゴ, Spansion, Spansionのロゴ及びこれらの組み合わせ, WICED, PSoC, CapsSense, EZ-USB, F-RAM, 及びTraveoは, 米国及びその他の国におけるCypressの商標又は登録商標である。Cypressのより完全な商標のリストは, cypress.comを参照すること。その他の名称及びブランドは, それぞれの権利者の財産として権利主張がなされている可能性がある。