PSoC® 1 - Simplified FSK Detection

Author: Dennis Seguine

Associated Project: Yes
Associated Part Family: CY8C27xxx, CY8C28xxx, CY8C29xxx
Software Version: PSoC® Designer™ 5.4 CP1
Related Application Notes: AN67391, AN60594
To get the latest version of this application note, or the associated project file, please visit http://www.cypress.com/go/AN2336.

AN2336 presents the standard correlator method for frequency shift keying (FSK) demodulation using an efficient almost all-hardware implementation in PSoC® 1. This signal processing technique is readily applicable to caller ID and several modem standards. The associated project makes extensive use of switched-capacitor block features unique to PSoC 1 and that is not available in PSoC 3 and PSoC 5. For PSoC 3 and PSoC 5, see AN60594.

Contents

Introduction

Frequency shift keying (FSK) is the modulation cornerstone of a number of digital data transmission systems. The transmit signal is fairly easy to generate, see AN67391 or the Appendix A section. Receive processing is tolerant of a wide range of signals and relatively immune to a large class of interfering signals. This Application Note outlines a simple, almost all-hardware method for detecting FSK signals using the analog signal processing capabilities of the PSoC 1 device. It describes circuits and signal processing techniques for demodulation. It is not a complete modem; it does not include the phone line or other interfaces; it also does not include the data processing code. A separate project is included and outlined in the Appendix A to simplify testing of the demodulator.

Operating Frequencies

The FSK signal is represented by:

$$v(t) = V_P \sin(2\pi(f_L + data(f_H - f_L))t) \qquad \text{Equation 1}$$

f_L and f_H are the respective low and high frequency modulation frequencies and the data value is logical 0 or 1. (Alternatively, the low frequency can represent a digital 1 and the high frequency a digital 0, as determined by whichever standard is used). Modulation frequencies come in standard pairs for various applications. The example used in this Application Note has operating frequencies of 1200 Hz and 2200 Hz modulated at a 1200 baud rate, and typical operating frequencies for Bell 202 modems, caller ID systems, and HART modems.

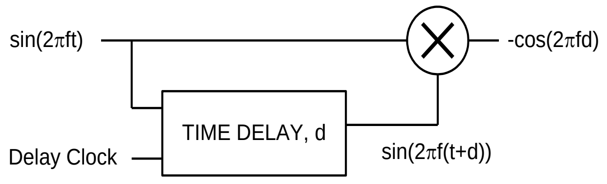
Detection Basics

There are several basic ways to detect FSK signals. First basic way will be filtering. You can setup the band pass filters at 1.2 kHz and 2.2 kHz, rectify and low-pass-filter the results, compare the output levels, and declare a bit "winner." In hardware this takes a bunch of parts, in software a fair amount of code and MIPS (we call that "computationally intensive"). Even then, it is level sensitive and not particularly immune to "twist," the telecom guy's name for imbalanced signals, where the low frequency level is typically 1-2 dB larger than the higher frequency.

The next standard technique is period measurement. Run the input signal through a zero-crossing detector, and then measure the period. This works well when the baud rate is slow relative to the lowest FSK frequency; unfortunately, this is not true in the Bell 202 case. Further, it takes a clock high enough to get a decent timing resolution and is not particularly immune to the noise.

The example here uses a mix of a simple analog and digital hardware. The core element in the FSK detector is the correlator, consisting of a delay line and a multiplier, shown in Figure 1.

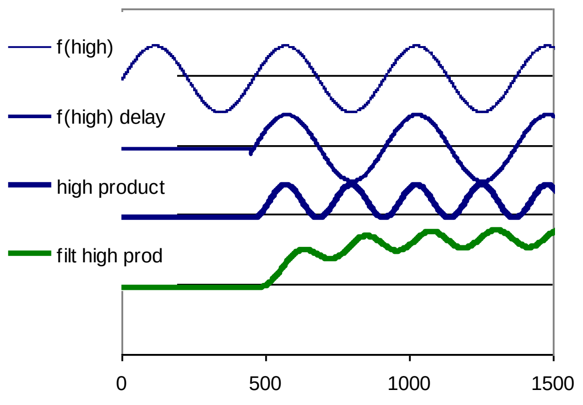
Figure 1. Correlator Block Diagram



The delay line delivers a time-delayed replica of the source signal. The multiplier multiplies the input signal by this delayed replica.

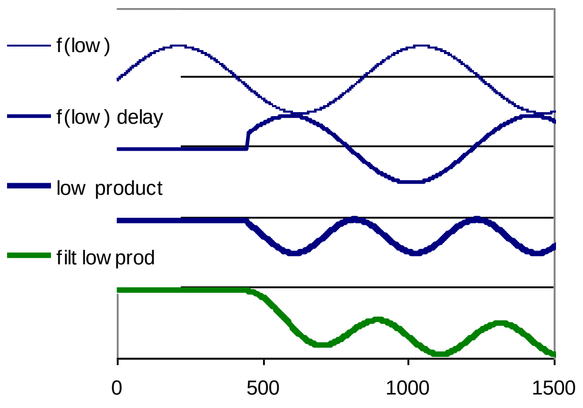
When the time delay is equal to a single cycle (or integer number of cycles), the product of the input and the delayed replica is positive, as shown in Figure 2. The product includes terms at DC and at twice the input frequency. The 2x term is filtered out leaving a positive DC value.

Figure 2. Correlator Waveform, Integer Delay



When the time delay is equal to a half cycle (or integer number of cycles plus a half), the product of the input and the delayed replica is negative, as shown in Figure 3. As in the integer cycle case, the 2x term is filtered out, this time leaving a negative DC value.

Figure 3. Correlator Waveform, Half Cycle Delay



Finding the Delay

The mathematics of the multiplier is simple:

$$m(t) = \sin(2\pi ft)\sin(2\pi f(t+d)) \qquad \text{Equation 2}$$

We shall start with the simplifying assumption that the levels are "unit" sized and fixed. Recalling trigonometric identities from high school and applying a little algebra, we find the multiplier output:

$$m(t) = \frac{1}{2}\begin{pmatrix} \cos(2\pi f(-d)) \\ -\cos(2\pi f(2t+d)) \end{pmatrix} \qquad \text{Equation 3}$$

The DC value (on the first line) represents the data, the 2π*2f term (double the input frequency, on the second line) is eliminated with a low-pass filter (LPF).

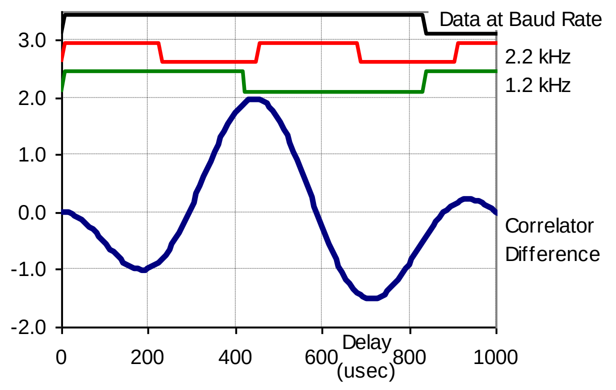
The product waveforms for the two frequencies are distinctly different. The correlator's multiplier output is a DC level that is a clear function of the operating frequencies and the delay. Eliminating the scaling for simplicity, the difference between the high and the low frequency delay products is:

$$Diff = \cos(2\pi f_L d) - \cos(2\pi f_H d) \qquad \text{Equation 4}$$

The design intent is to find a delay that will result in one multiplier output value for the low frequency and a very different value for the high frequency. That is, to find d so as to maximize *Diff*.

An example for $f_L = 1200$ and $f_H = 2200$ is shown in Figure 4. Creative use of a spreadsheet or MATLAB® shows the correlator output as a function of the delay. The lower (blue) trace is the difference between the correlator output at low frequency and the correlator output at high frequency. The Baud rate, low frequency, and high frequency are shown as square waves for timing reference.

Figure 4. Detector Difference vs. Correlator Delay



Data is detected when the DC output is sufficiently distinctly positive to declare a logic 1, or sufficiently distinctly negative to declare a logic 0. The DC output is compared to a threshold; a larger difference between the positive and negative values will result in more reliable data.

So, what constitutes a “large enough?” value of *Diff*? Since we defined these as being unit-sized waveforms, the maximum and minimum signal values after the multiplication operation are 1/2 and -1/2 times the correlator input signal level. We need to set a threshold between these values to determine the level detection. We could set the threshold at zero, thus, any positive value represents one frequency and any negative value represents the other. In this case, the difference must be greater than 1/2. While a difference of 1/4 could be enough, there is nothing to assure that both values are not positive, say one at 0.15 and the other at 0.4. In this case, setting up a comparator to find the appropriate non-zero level is less straightforward than a zero-crossing detector.

As a practical matter, a delay that yields the LARGEST possible difference simplifies the filtering and detection process. This is done by optimizing Equation 4 for the maximum value of *Diff*. For the example chosen, the first peak that meets the minimum separation requirement occurs at a delay of 446 µsec with a difference value (*Diff*) close to the maximum possible of 2.0. For the available clocks, the closest delay value is 448 µsec. A delay of 720 µsec will also work. Detection with the minimum delay results in the highest detection bandwidth or baud rate and allows the most time for low pass filtering to remove the 2*carrier frequency term from the multiplier and reduce the noise.

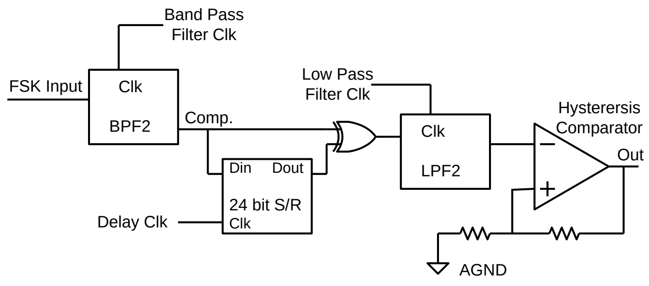
Correlator Implementation

The elements of the correlator are easily implemented in the PSoC device. Once out-of-band noise has been rejected, the detection process is a digital problem. The sine wave output of the filter is converted to a square wave with a comparator. Conveniently, this comparator is built into the band pass filter (BPF) user module (UM) in the PSoC; it can directly feed the digital blocks.

The time delay function is a simple shift register, with the length and clock set for the delay required, and the sample rate high enough to faithfully represent the waveform.

There are two ways to implement the multiplier. One way is to use an analog modulator. Mathematically equivalent, and even easier to implement, an Exclusive-OR (XOR) gate acts as a digital multiplier. The output of the XOR is passed to the same filter that would be used on the modulator version. The completed block diagram is shown in Figure 5.

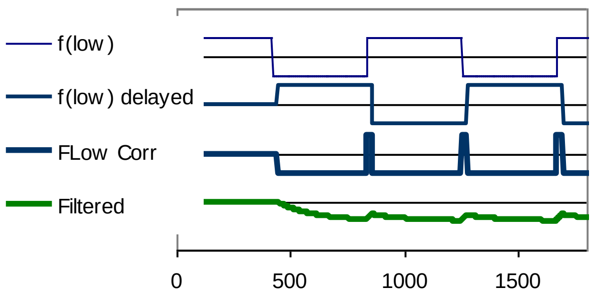
Figure 5. PSoC FSK Detector



The correlator waveforms and output are shown for the high frequency in Figure 6.

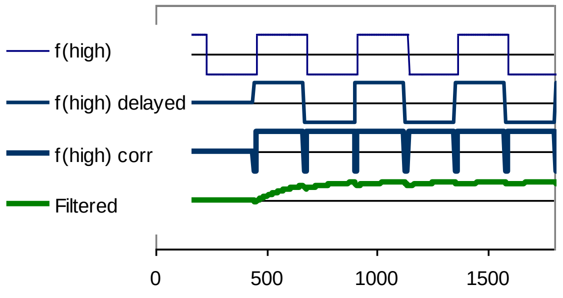
The multiplier output waveform appears as a digital signal with a low duty cycle at f_L and a high duty cycle at f_H. This is filtered to recover the DC level, clearly different for the two modulating frequencies.

Figure 6. Correlator Waveforms, Half Cycle Delay



For the delay selected (448 usec), the delayed waveform at the low frequency is one half cycle behind the input. Multiplied in the XOR, the output of the correlator is a pulse train, mostly positive and easily filtered with the LPF2 to a negative DC value. With the same delay applied to the high frequency, shown in Figure 7, the input and the delayed signal are separated by a full cycle. Multiplied in the XOR, the output of the correlator is mostly negative pulse train. The same low-pass filter works equally well at either frequency.

Figure 7. Correlator Waveform, Integer Cycle Delay



Now it is just a simple matter of routing the blocks, designing the filters, and finding a workable set of clock frequencies to fit both the filter and the delay line requirements.

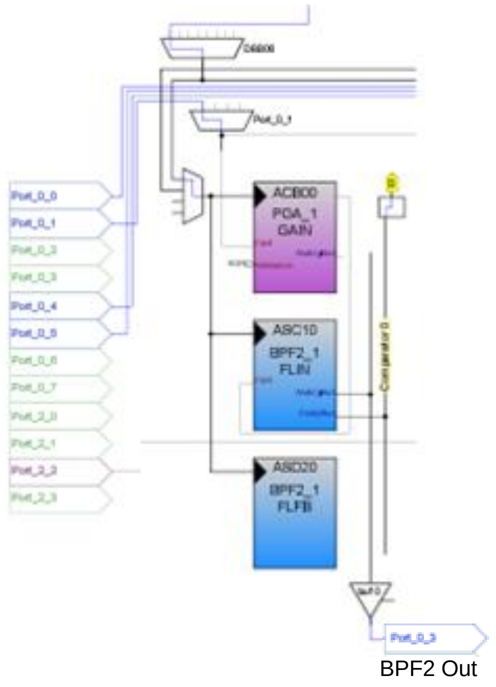
PSoC Implementation

The PSoC Designer™ project is built in a CY8C27x43, using relatively little of the block, routing and code resources.

Analog Front End

The analog front end consists of an amplifier and a band pass filter, shown in the block layout, see Figure 8.

Figure 8. Analog Input



The input to the programmable gain amplifier (PGA) UM must be externally biased at a fixed reference voltage, typically Vdd/2. The PGA enables the use of additional gain for a wider input dynamic range. Over-driving the PGA output (by using too much gain) will bring up the out-of-band noise but will not adversely affect the FSK detection process. The band pass is a single-pole pair filter implemented in a switched capacitor filter BPF2 UM. The filter is initially designed for 0 dB gain (V/V = 1.0) and centered at the geometric mean of the two modulating frequencies. The bandwidth is set somewhat wider than the frequency difference. Significantly increasing filter bandwidth reduces the noise rejection. Decreasing filter bandwidth slows the detection time and reduces the signal level. The filter has a balanced loss of about 1.5 dB at both modulating frequencies. More complex filters for better out-of-band noise rejection can be easily implemented using the BPF4 UM, if desired; consuming an additional two switched capacitor analog blocks.

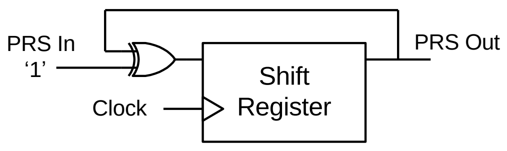
The comparator output of the filter is used directly by the correlator; there is no need to bring out the analog filter output. The analog filter output, however, does provide a useful point to monitor system performance and is brought out on P0[3].

The band pass filter's comparator bus output for column 0 is routed to the Input 1 of a digital buffer (DigBuf) User Module; the output of the DigBuf is routed to an output row and the adjacent digital block. This enables use of the output row logic lookup table (LUT) to implement the XOR.

Correlator - Shift Register

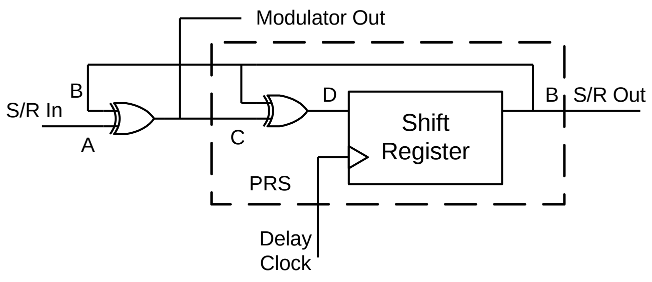
There is no shift register present in the CY8C27xxx and CY8C29xxx, so we creatively connect blocks, the LUT and routing resources to make one. The shift register is a modification of the pseudo random sequence (PRS) generator user module. This user module has selectable feedback taps that are XORed back to the input, normally used to generate a maximum length digital sequence. For this application, the polynomial routes the single tap at the end of the shift register chain back to the PRS input. The PRS is hard-wired to invert the output fed back to the input using an XOR, as shown in Figure 9.

Figure 9. Basic PRS



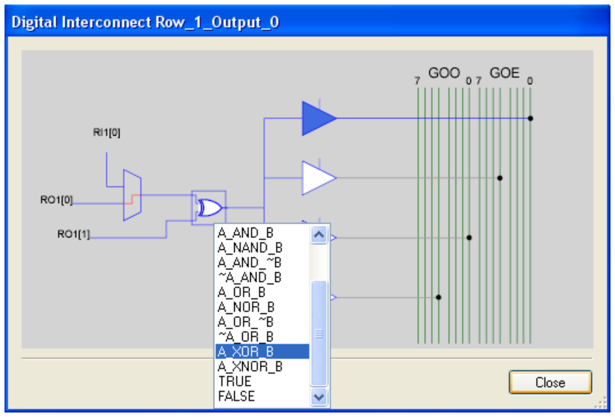
XORing the S/R input with the PRS output yields a shift register by un-inverting the input to the PRS's internal S/R. The input connection is modified by adding another Exclusive-OR as shown in Figure 10, with the PSoC Designer routing shown in Figure 11.

Figure 10. Shift Register Implementation with PRS



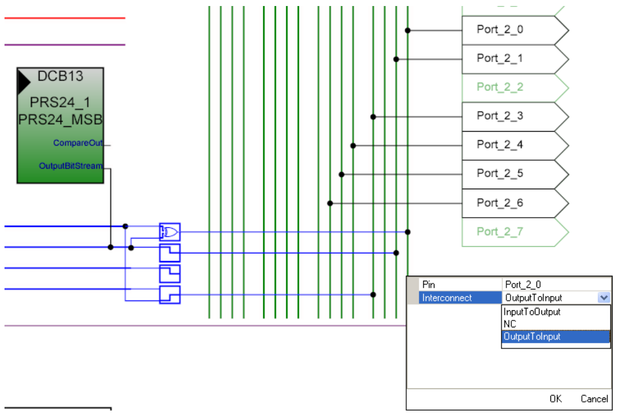
The added XOR is accomplished in the output row LUT.

Figure 11. XOR Selection in LUT



The output of the LUT is routed back to the PRS input using global routing resources to get to the input row. This is done by selecting the Global Output (in this case GlobalOutEven0, or GOE0) and accessing Output to Input in the pull-down window as shown in Figure 12.

Figure 12. Output to Input Global Routing



The input row is selected as the data input of the PRS through software, as the PRS data input is normally wired high for the pseudo-random feedback function.

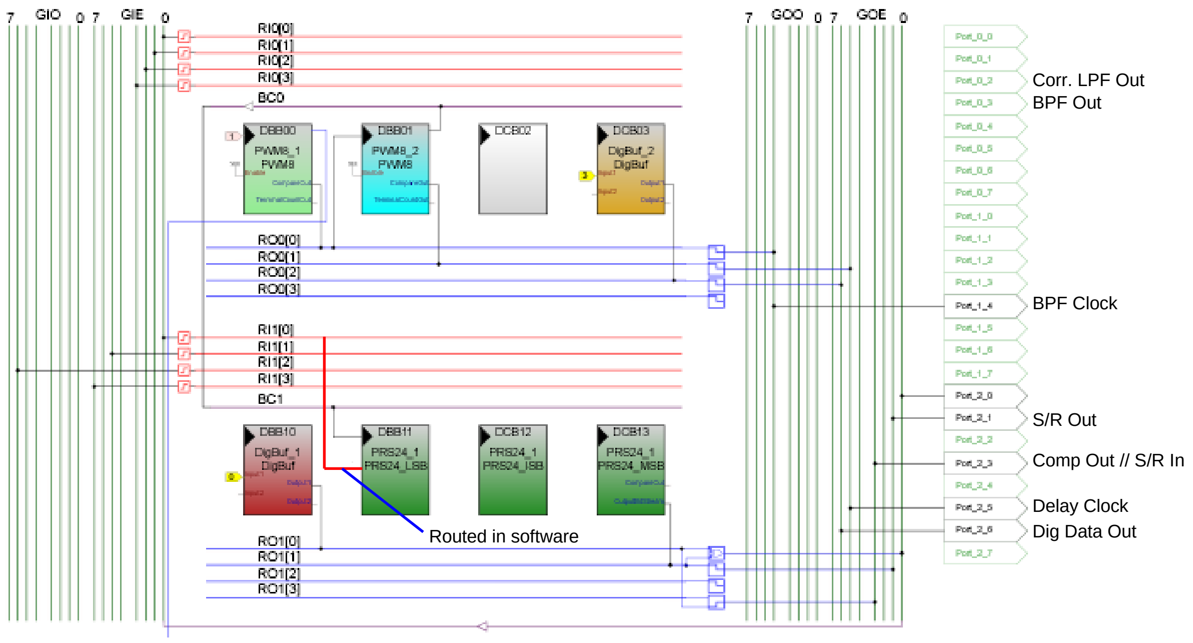
Digital Resource Implementation

The shift register clock is derived from the 24 MHz system clock. With the delay set to 448 µsec (selected from data of Equation 4), and a 24-bit shift register, we calculate a delay clock of f_{CLK} = 1/(448 µs/24) = 53.57 kHz, divided down by a PWM and the VC1 clock. The length of the shift register is set by selecting the proper tap on the 24-bit PRS using the WritePolynomial API provided as part of the user module. In this case, the maximum length is set with dwPoly = 00800000h.

CY8C28xxx has a shift register user module which eliminates the need for routing connections in software.

The complete digital block routing for the CY8C27xxx and CY8C29xxx solution is shown in Figure 13.

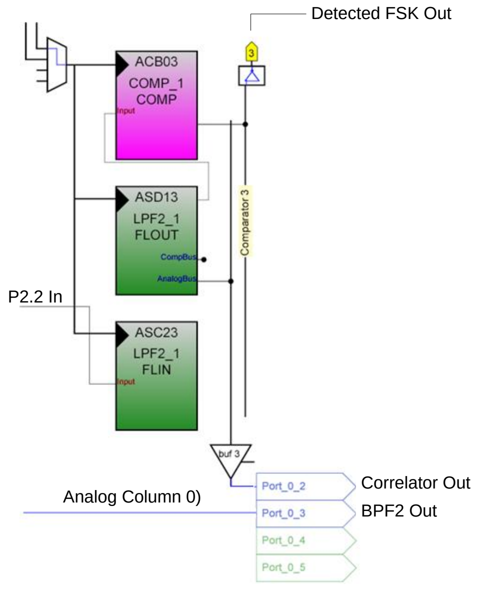
Figure 13. Digital Block Structure



Data Recovery

The modulator (XOR) output is a logic signal; it swings from rail-to-rail. The signal swing is a function of the degree of match to the required correlator delay and the gain of the LPF. The output of the modulator directly feeds the input of the LPF, as shown in Figure 14.

Figure 14. Low-Pass Filter and Data Recovery



The filter is a two-pole Butterworth at 760 Hz with a rise time of approximately 250 μsec.

Added to the correlator delay, the system's response time to a step change in frequency is approximately 620 μsec. In order to eliminate false logic triggers caused by the noise, a hysteresis comparator is used. In a typical application, the comparator output is routed to a UART. In CY8C27xxx and CY8C29xxx, the comparator output is an internal-only signal. In the included project, the comparator output is routed to a digital buffer to provide access on a digital output pin. In CY8C28xxx, the comparator output can drive the Global Output lines, and then connect to a pin, obviating the need for the digital buffer.

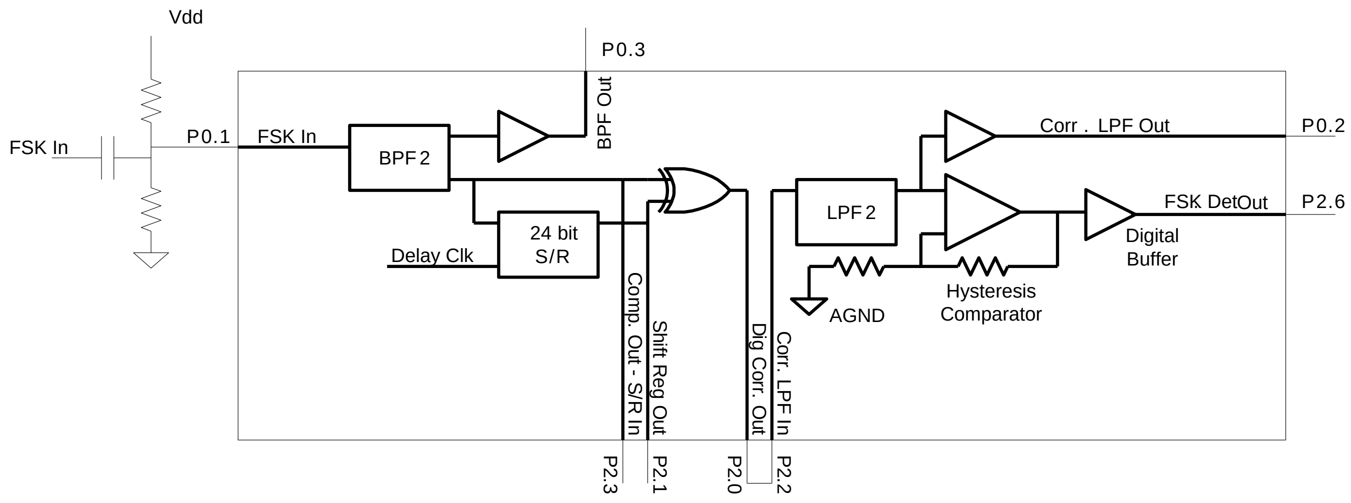
The software configures some of the routing selections, sets power levels and starts all of the user modules.

Design Demonstration

A separate project was generated using the CY8C29466 device to provide continuous FSK data for testing the detector, in lieu of using an arbitrary waveform generator. The project provides a 1200-baud signal, two bits high, one bit low, modulating the filtered output at 1.2 kHz and 2.2 kHz. Methods for building the FSK signal generator are covered in the Application Note AN67391.

A complete schematic is for the FSK demodulator is shown in Figure 15. The only external components required are for bypass and input biasing.

Figure 15. PSoC FSK Detector Schematic



The digital source (modulating waveform), FSK output, intermediate waveforms, and digital output are shown in Figure 16.

Figure 16. FSK Waveforms

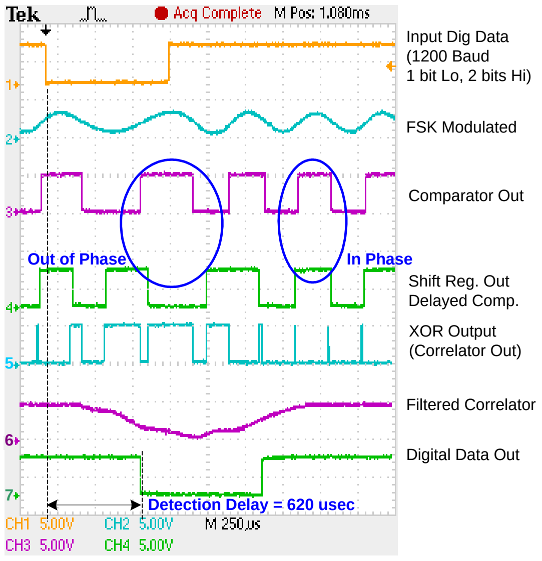
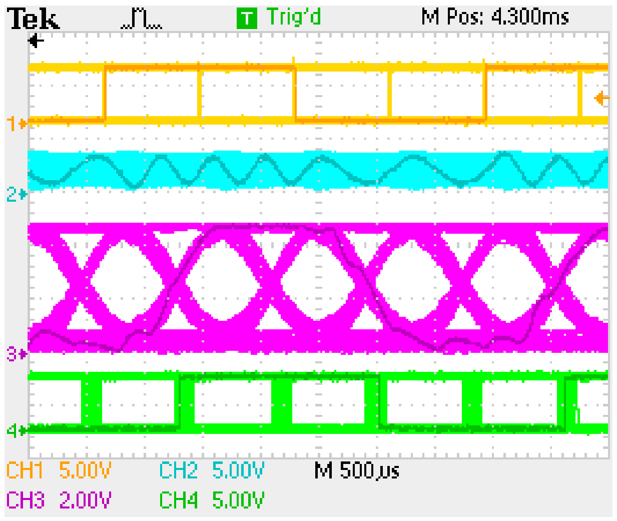


Figure 17. FSK Eye Diagram



Note Waveform Reference

1. Digital modulation at 1200-baud signal, two bits high, one bit low. (From test source)
2. 1200 and 220 Hz FSK signals, easily seen because of the wide spread in frequency. (From test source)
3. Output of filter comparator.
4. Correlator delay line output shifted 448 µsec from the comparator output.
5. XOR output. The duty cycle of the waveform is aligned up with the phase relationship of the comparator and delayed signals.
6. Filtered correlator signal. This recovers the DC average output of the XOR.
7. Detection comparator output. With just a little bit of hysteresis in the comparator, has been found to be very stable and noise-free over a wide range of input levels.

The detection delay time is approximately equal to the RMS sum of several delays, including input BPF response, correlator delay, and correlator low-pass filter response. The total detection time is about 620 µsec, less than the bit time of 833 µsec for the 1200-baud signal.

The standard data presentation for demodulation is the eye diagram, shown in Figure 17. Pseudo-random data (Trace 1) is fed to the test modulator then into the FSK detector. The output of the demodulator is measured for some extended period. The quality of the demodulation is measured by the “openness” of the “eye.” This technique clearly shows robust performance.

Design Opportunities

A bare essential FSK detector design has been shown. There are numerous opportunities for design refinements, including increasing gain, improving filter selectivity, adapting to other operating frequencies to meet other modem standards, and completing the connection of the recovered serial digital data through a UART RX to the CPU's bus. The design techniques are easily extended to these other applications.

When connection to a phone line, a direct (or data) access arrangement (DAA) is used. The DAA circuit usually includes a transformer with specific regulatory requirements for coupling to the line. Design of this coupling circuit is not part of this application note.

Design Alternatives

This design implemented is nearly pure hardware, but the signal processing techniques are equally realizable in software. So, here is the algorithm (a description, not the code):

1. Digitize. Do not believe Nyquist, setup sample rate for filter at a frequency equal to an integer multiple of both FSK signals. In this case, 13.2 ksps works. (11x on 1.2 kHz, 6x on 2.2 kHz). Higher frequency is better still. Use enough bits to make the filter work over your expected input signal range.
2. Implement BPF 1 dB flatness from 1.2 kHz to 2.2 kHz.
3. Build a comparator on the filter output. You have already got one. All you need now is the polarity (think, comparator = 1 bit ADC).

4. Build a delay line with a FIFO, length = 448 µsec*sample frequency (= 6 in this case). Adjusting the filter and delay line sample rates so that they are synchronous can provide a more accurate delay and cleaner correlator “waveforms.”
5. Multiply comparator and delay signal single-bit values using logical single bit XOR.
6. Low pass filter the product (the XOR) using an IIR or FIR filter.
7. Set thresholds, make a hysteresis comparator
8. Direct comparator output to UART or caller ID as required.

Summary

This project demonstrates bare essential FSK signal detection. It is immediately adaptable to caller ID and Bell 202 modem (1200-baud, half-duplex) applications. It uses less than half of the analog capability of the CY8C27x43, most (but not all) of the digital capacity of the device, only 14 lines of source code in C, no run-time manipulations and no interrupts. It uses less than half of the digital block count in CY8C29xxx. This leaves a considerable amount of analog and digital signal processing capability and almost all of the code space for the user to implement a system in which the FSK demodulation (and transmission) is a small part of the system.

The author has tested this technique successfully at 1200 baud in Bell 202 applications, v.23 modems, 2400 baud PSoC power line modems at carrier frequencies of 110 kHz and 132.45 kHz plus numerous other frequency ranges.

FSK detection using the correlator is THE standard way to go. It is easy to implement in hardware (Cypress PSoC and ZERO external parts) or in code.

About the Author

Name:	Dennis Seguine.
Title:	Applications Engrg MTS

Appendix A

Testing the Demodulator

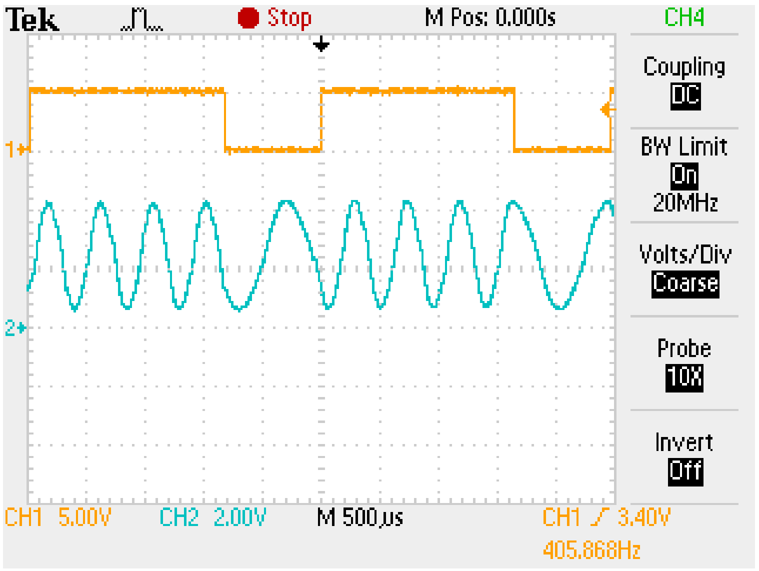
The demodulator is tested by driving an FSK modulated signal to the input. This can be derived from the user's system, an arbitrary waveform generator (example Agilent 33210A or equivalent) or by using an FSK generator built from a separate PSoC chip.

The PSoC test source is easily constructed from a CY8C29466 using project AN2336_FSK_TestGen.soc. It can be cloned to any CY8C27, 28 or 29xxx. The FSK output of the test generator is automatically biased at Vdd/2; this eliminates the need to provide biasing components at the input of the demodulator under test.

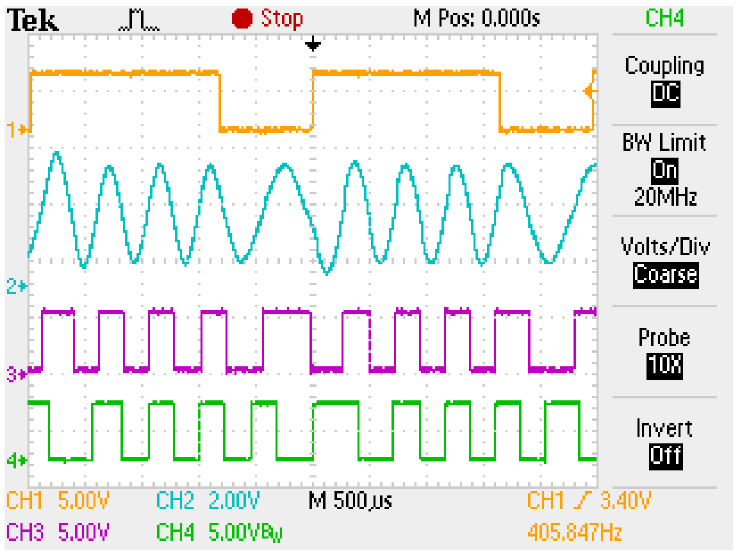
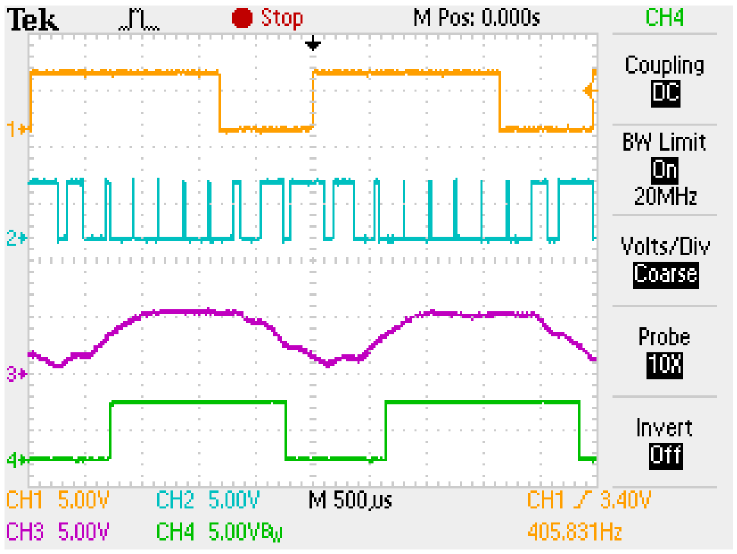
The test source has two PWMs operating at a multiple of the output frequency. The PWMs are controlled by simulated FSK data driving the PWM enable inputs. The simulated FSK data is provided by a PWM16 which outputs a pulse train at the Baud rate. When one PWM is enabled the other is disabled and static. The outputs are XORed to make a clock for the final output PWM and to provide a clock for the output filter. The PWM16's period and pulse width can be modified to change the FSK data pattern. It can be replaced with a psuedo-random binary sequence generator user module (UM) to provide data for “eye” testing, as in Figure 17. The user has considerable flexibility in modifying this simple project to generate the necessary data patterns.

Operation of the demodulator and test project is straight-forward. Download the FSK project into the target CY8C27443 chip. Download AN2336_FSK_TestGen.soc into a CY8C29466, provide power connections and connect P0.3 of the test generator output to demodulator input on P0.1 of the demodulator PSoC.

Verification of performance is a simple matter of stepping through the waveforms. The test generator comes first.

1. P1.4 (test gen), Pulse train 400 Hz, 67% duty cycle simulates 1200 Baud, 1 bit low, 2 bits high. 2. P0.3 (test gen), Sine wave 1.2 kHz when P1.4 is low, Sine wave 2.2 kHz when P1.4 is high.	

Next, using the FSK data output from the test generator chip as an oscilloscope trigger, the demodulator waveforms are examined.

1. P1.4 (test gen) 2. P0.3 (UUT), Band pass filtered copy of input on P0.1 3. P2.3 (UUT), Zero-crossing replica of filter output on P0.3, 1.2 and 2.2 kHz sq wave 4. P2.1 (UUT), 448 µsec delayed copy of output on P2.3	 Tek Stop M Pos: 0.000s CH4 Coupling DC BW Limit On 20MHz Volts/Div Coarse Probe 10X Invert Off CH1 5.00V CH2 2.00V M 500µs CH1 ⟋ 3.40V CH3 5.00V CH4 5.00V 405.847Hz
1. P1.4 (test gen) 2. P2.0 (and P2.2) (UUT), Correlator output XOR of P2.1 and P2.3 Mostly low when input pulse train is high. Mostly high when input pulse train is low. Translation from low duty cycle to high duty cycle is delayed ~500 µsec from input pulse train. 3. P0.2 (UUT), Low pass filtered correlator output, follows polarity of P2.0 output with logic level slew ~ 600 µsec 4. P2.6 (UUT), Hyst. comp.of P0.2 Logic level follows input pulse train (P1.4 test gen) with delay of ~ 600 µsec	 Tek Stop M Pos: 0.000s CH4 Coupling DC BW Limit On 20MHz Volts/Div Coarse Probe 10X Invert Off CH1 5.00V CH2 5.00V M 500µs CH1 ⟋ 3.40V CH3 5.00V CH4 5.00V 405.831Hz

Document History

Document Title: PSoC® 1 - Simplified FSK Detection - AN2336

Document Number: 001-15224

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	1058621	SEG	07/29/2007	New application note.
*A	3060389	YJI	10/14/2010	No technical updates. Completing Sunset Review.
*B	3184719	SEG	02/03/2011	Updated Detection Basics: Updated Figure “Correlator Waveforms, Optimized Delay” and description. Updated PSoC Implementation: Removed Figure “Multiplier and Output”. Updated Design Demonstration: Updated Figure 15 and description. Updated Design Opportunities: Updated description. Updated Summary: Updated description.
*C	3190744	SEG	03/08/2011	Updated attached Example Project.
*D	3428146	SEG	11/03/2011	Updated Abstract: Updated description. Updated Introduction: Updated description. Updated Detection Basics: Removed Figure “Correlator Waveforms, Optimized Delay”. Added Figure 2, Figure 3. Updated Correlator Implementation: Removed Figure “Modulator Schematic” and “Digital Correlator Waveforms”. Added Figure 6, Figure 7 and updated description. Updated PSoC Implementation: Removed Figure “Multiplier and Output”. Added Figure 11, Figure 12 and updated description. Updated Digital Resource Implementation: Updated Figure 13 and updated description. Updated Design Demonstration: Updated description. Added Appendix A. Updated to new template.
*E	4580688	DIMA	11/26/2014	Updated Software Version as “PSoC® Designer™ 5.4 CP1” in page 1. Updated attached code example to PSoC Designer 5.4 CP1. Updated to new template. Completing Sunset Review.
*F	5713034	AESATP12	04/26/2017	Updated logo and copyright.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709