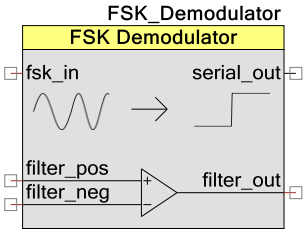
FSK Demodulator (FSK_Demodulator)

1.0

Features



Demodulates a frequency shift keying (FSK) signal at one of two carrier frequencies (110 kHz or 132 kHz)

Selectable filter to attenuate out-of-band signals

Selectable receive gain for improved input sensitivity (as low as 45 dBμVrms)

Receive signal strength indicator (RSSI)

Requires a PSoC 5LP device with a SAR ADC and DFB (CY8C56xx or CY8C58xx)

Compatible with the FSK Modulator and PLC Network Protocol components, as well as the PSoC® 1 powerline communication solution

General Description

The FSK Demodulator component converts an FSK signal into a baseband serial signal, which can be received by a UART. An FSK signal represents each bit of information as one of two frequencies (for example, 133 kHz represents logic ‘0’ and 130 kHz represents logic ‘1’).

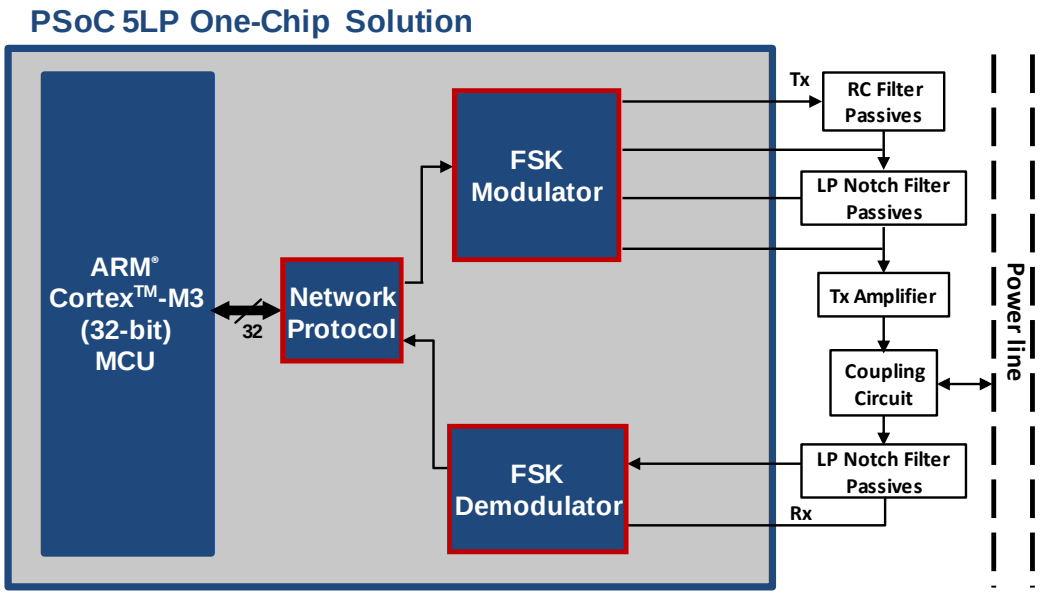
The FSK Demodulator component also filters out any out-of-band noise so that it can more reliably demodulate the FSK signal. The filter is a low-pass notch filter, which uses the PSoC 5LP’s integrated opamp and some external passive components, which are described below in the Schematic Macro Information section.

When to Use a FSK Demodulator

This component is most commonly used for powerline communication (PLC). The packet of data is created by the PLC Network Protocol component, which has a UART to convert each byte of data into a serial signal. The FSK Modulator then converts the serial signal into an FSK signal, which can be coupled to the powerline by a coupling circuit (described in application note AN76458 “PSoC 5LP High Voltage (120 – 240 VAC) Powerline Communication”). The receiver on the other end of the line uses the FSK Demodulator component to convert the FSK signal back into a serial signal for the Network Protocol.

PRELIMINARY <remove this label when the component is production ready>

Figure 1. Powerline Communication System Diagram



Input/Output Connections

This section describes the various input and output connections for the FSK Demodulator. An asterisk (*) in the list of I/Os states that the I/O may be hidden on the symbol under the conditions listed in the description of that I/O.

fsk_in – Analog

The fsk_in terminal is the input for the FSK signal.

serial_output –Output

The serial_output terminal is the output of the demodulated serial baseband signal, which can be received by a serial communication component (for example, a UART).

filter_pos – Analog *

The filter_pos terminal is the positive input of the internal opamp that filters the FSK signal. It is present when the FSK Demodulator Receive_Filter parameter is Enabled in the Configuration window.

filter_neg– Analog *

The filter_neg terminal is the negative input of the internal opamp that filters the FSK signal. It is present when the FSK Demodulator Receive_Filter parameter is Enabled in the Configuration window.

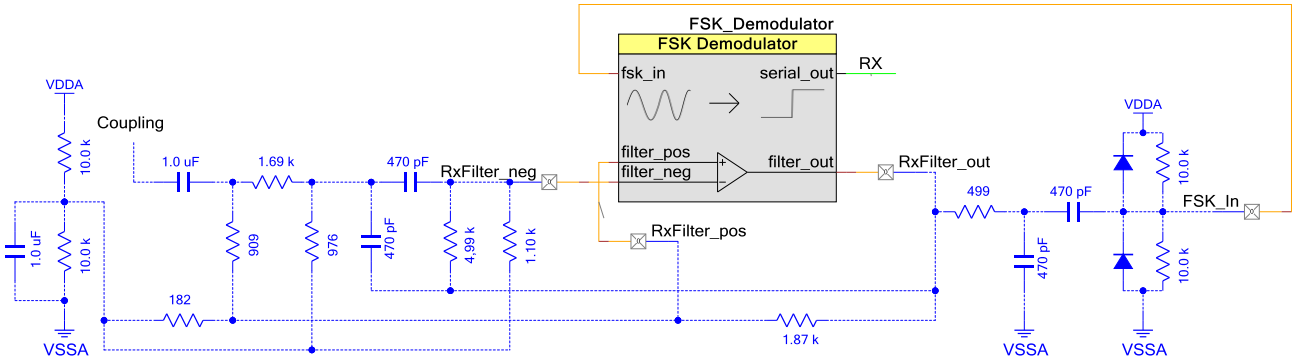
fsk_aout – Analog *

The fsk_aout terminal is the output of the internal opamp that filters the FSK signal. It is present when the FSK Demodulator Receive_Filter parameter is Enabled in the Configuration window.

Schematic Macro Information

The default FSK Demodulator in the Cy_Ref Catalog is a schematic macro using a FSK Demodulator component with the default settings. It connects to analog pins for the opamp connections and for the FSK input). The input to the circuit (labeled “Coupling”) is from the coupling circuit, which is described in the application note AN76458 “PSoC 5LP High Voltage (120 – 240 VAC) Powerline Communication”.

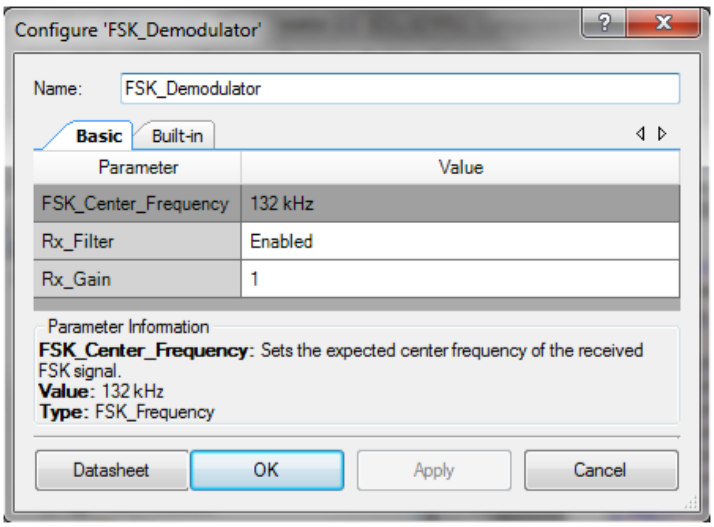
Figure 2. FSK Demodulator Macro



Component Parameters

Drag a FSK Demodulator onto your design and double click it to open the Configure dialog. Figure 3 shows the Configure dialog.

Figure 3. Configure FSK Demodulator Dialog



The FSK Demodulator provides the following parameters.

Basic Options

FSK_Center_Frequency

This parameter sets the expected center frequency of the received FSK signal. When set to 132 kHz, the FSK Demodulator will demodulate a 133.333 kHz input to a logic ‘0’ and demodulate a 130.434 kHz input to a logic ‘1’. When set to 110 kHz, the FSK Demodulator will demodulate a 111.111 kHz input to a logic ‘0’ and demodulate a 109.091 kHz input to a logic ‘1’. The default value is 132 kHz.

Receive_Filter

This parameter enables the opamp that (along with external passives) filters out out-of-band noise (not in the FSK frequency band). This increases the signal-noise-ratio and improves the receiver’s ability to correctly demodulate the FSK signal in the presence of noise. When set to Disabled, the opamp will be off, which saves power. The default value is Enabled.

Receive_Gain

This parameter sets the gain of the built-in PGA, which amplifies the received signal. A higher value will provide better input sensitivity (can receive a smaller signal). A lower value will provide

less sensitivity to noise (can receive a signal with higher noise). When set to None, the PGA will be bypassed, which saves power. The default is 1.

Clock Selection

The master clock for the project must be 48 MHz to have the correct ADC sample rate and to execute all of the Digital Filter Block (DFB) instructions in time before the next sample. In order to accurately demodulate the FSK signal, the PLL must be sourced by an external 24 MHz crystal with a maximum stability/tolerance of +/-50 ppm.

Resources

The FSK Demodulator uses the following device resources. The number of resources depends on the value of the Receive_Filter and Receive_Gain parameters.

Configuration		Digital Blocks	Analog Blocks								
Receive_Filter	Receive_Gain	Control Cells	SAR ADC	SC Fixed Blocks	Opamps	DFB	DMA Channels	Digital Clocks	Analog Clocks	Interrupts	Pins
Enabled	1 - 32	1	1	1	1	1	2	1	1	1	4
Enabled	None	1	1	0	1	1	2	1	1	1	4
Disabled	1 - 32	1	1	1	0	1	2	1	1	1	1
Disabled	None	1	1	0	0	1	2	1	1	1	1

API Memory Usage

The component memory usage varies significantly, depending on the number of APIs used and component configuration. The following table provides the memory usage for all APIs available in the given component configuration.

The measurements were performed with the associated compiler configured in Release mode and with optimization set for Size. For a specific design, the map file generated by the compiler can be analyzed to determine the memory usage.

Configuration		API Memory (Bytes)	
Receive_Filter	Receive_Gain	Flash	RAM
Enabled	1 - 32	3800	32
Enabled	None	3448	32
Disabled	1 - 32	3608	32
Disabled	None	3280	32

Application Programming Interface

Application Programming Interface (API) routines allow you to configure the component using software. The following table lists and describes the interface to each function. The subsequent sections cover each function in more detail.

By default, PSoC Creator assigns the instance name "FSK_Demodulator_1" to the first instance of a component in a given design. If using the Network Protocol component, then you must name this component “FSK_Demodulator”. If not using the Network Protocol component, you can rename it to any unique value that follows the syntactic rules for identifiers. The instance name becomes the prefix of every global function name, variable, and constant symbol. For readability, the instance name used in the following table is "FSK_Demodulator."

Functions

Function	Description
FSK_Demodulator_Start()	Starts the blocks to demodulate an FSK signal
FSK_Demodulator_Stop()	Stops the blocks
FSK_Demodulator_Init()	Initializes the default configuration provided by the customizer
FSK_Demodulator_SetRxGain()	Sets the amount of gain that is applied to the received FSK signal
FSK_Demodulator_SetFskFrequency()	Sets the expected FSK center frequency
FSK_Demodulator_ReadFskPeak()	Returns the amplitude (in dB) of the largest received signal in the FSK frequency band
FSK_Demodulator_ResetFskPeak()	Clears the amplitude of the largest received signal in the FSK frequency band

Global Variables

Function	Description
FSK_Demodulator_initVar (static)	The initVar variable indicates the initial configuration of this component. This variable is prepended with the component name. The variable is initialized to zero and set to 1 the first time FSK_Demodulator_Start() is called. This allows for component initialization without reinitialization in all subsequent calls to the FSK_Demodulator_Start() routine.

void FSK_Demodulator_Start(void)

Description: This is the preferred method to begin component operation. This function turns on the demodulator with the center frequency and amplitude from the settings provided during the configuration or the current values after FSK_Demodulator_Stop() has been called. It also turns on the opamp for the filter. On the first call of this function, it sets the FSK_Demodulator_initVar variable and calls the FSK_Demodulator_Init() function.

Parameters: None

Return Value: None

Side Effects: None

void FSK_Demodulator_Stop(void)

Description: This function turns off the demodulator and turns off the opamp for the filter.

Parameters: None

Return Value: None

Side Effects: None

void FSK_Demodulator_Init(void)

Description: Initializes or restores the component according to the customizer Configure dialog settings. It is not necessary to call FSK_Demodulator_Init() because the FSK_Demodulator _Start() API calls this function and is the preferred method to begin component operation.

Parameters: None

Return Value: None

Side Effects: None

void FSK_Demodulator_SetRxGain(uint8 rxGain)

Description: Sets the gain of the receiver's programmable gain amplifier. This is only available when the Receive_Gain parameter is set to a value other than None.

Parameters: uint8 rxGain: one of the following gain values

Options	Description
FSK_Demodulator_GAIN_01	Gain = 1
FSK_Demodulator_GAIN_02	Gain = 2
FSK_Demodulator_GAIN_04	Gain = 4
FSK_Demodulator_GAIN_08	Gain = 8
FSK_Demodulator_GAIN_16	Gain = 16
FSK_Demodulator_GAIN_24	Gain = 24
FSK_Demodulator_GAIN_32	Gain = 32

Return Value: None

Side Effects: The global variable FSK_Demodulator_rxGainDelta is set according to the gain so that the measured peak amplitude (FSK_Demodulator_ReadFskPeak) is compensated for the gain.

void FSK_Demodulator_SetFrequency(uint8 centerFrequency)

Description: Sets the expected center frequency of the received FSK signal.

Parameters: uint8 centerFrequency: one of the following center frequencies of the FSK signal

Options	Description
FSK_Demodulator_110K	The expected center frequency is 110 kHz. The FSK Demodulator will demodulate a 111.111 kHz signal to Logic '0' and will demodulate a 109.091 kHz signal to Logic '1'.
FSK_Demodulator_132K	The expected center frequency is 132 kHz. The FSK Demodulator will demodulate a 133.333 kHz signal to Logic '0' and will demodulate a 130.434 kHz signal to Logic '1'.

Return Value: None

Side Effects: None

uint8 FSK_Demodulator_ReadFskPeak(void)

Description: Returns the largest amplitude (in dB) that the FSK Demodulator has received since the last time FSK_Demodulator_ResetFskPeak was called or the device was reset.

Parameters: None

Return Value: uint8: the largest amplitude (in dB) that the FSK Demodulator has received

Side Effects: None

void FSK_Demodulator_ResetFskPeak(void)

Description: Resets the largest amplitude that the FSK Demodulator has received.

Parameters: None

Return Value: None

Side Effects: None

Sample Firmware Source Code

The code example in the application note AN76458 “PSoC 5LP High Voltage (120 – 240 VAC) Powerline Communication” contains the FSK Demodulator, along with the FSK Modulator and Network Protocol components. This forms a complete powerline communication (PLC) system and demonstrates how to read an analog voltage with an ADC, transmit that value over the powerline, then receive the value and display it on an LCD.

Functional Description

The FSK Demodulator component converts a FSK signal into a baseband serial signal (for example, a UART data stream). A FSK signal represents each bit of information as one of two frequencies. **Error! Reference source not found.** provides an example where a 100 kHz sine wave is demodulated to logic ‘1’ and a 200 kHz sine wave is demodulated to logic ‘0’.

Figure 4. FSK Demodulation Example

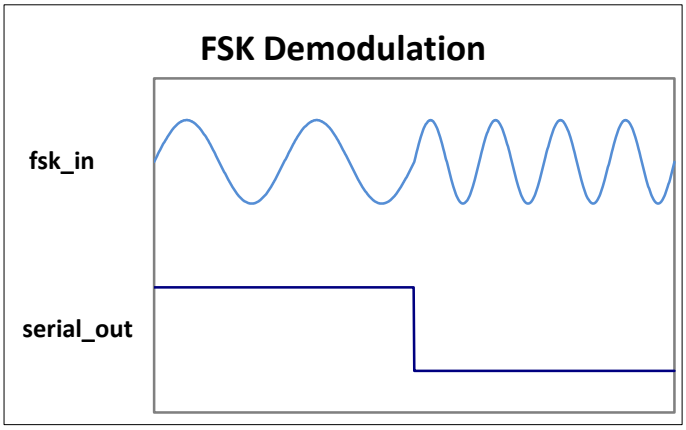
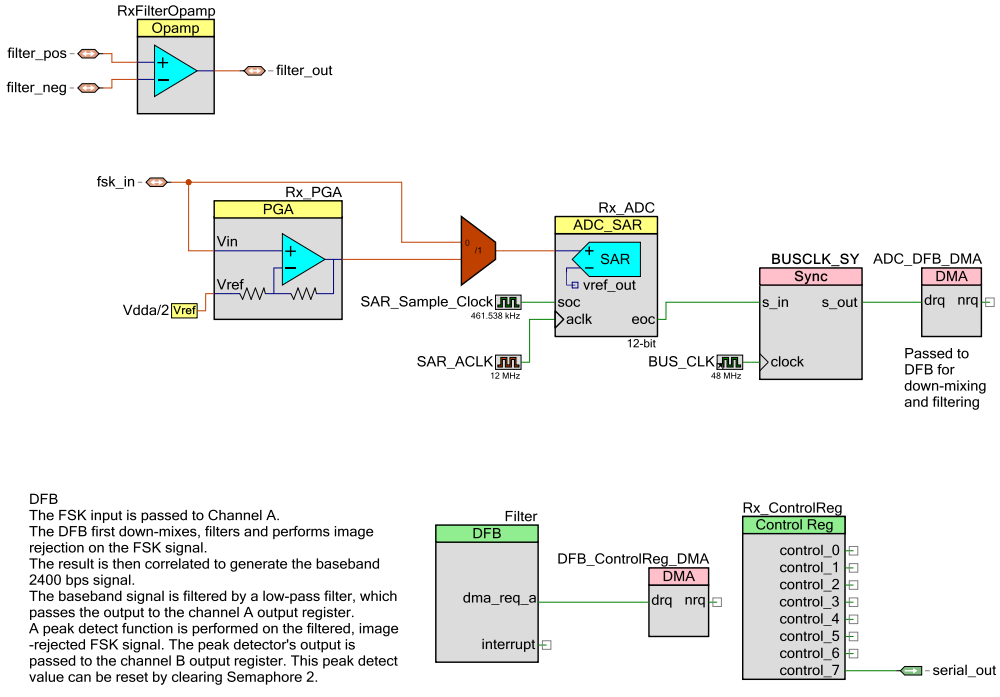


Figure 5 shows the FSK Demodulator schematics. The FSK Demodulator consists of the following parts (PSoC Creator components in parentheses):

- External Low-pass Notch Filter (opamp) - Optional
- Amplifier (PGA) - Optional
- Sampler (ADC)
- Down-mixer with image rejection, Band-pass Filter, Correlator, and Low-Pass Filter (DFB, Control Register)

Figure 5. FSK Demodulator Internal Schematics



PRELIMINARY <remove this label when the component is production ready>

Input Low-pass Notch Filter - Optional

The low-pass notch filter attenuates out-of-band high frequency noise. It uses an internal opamp along with external resistors and capacitors. More information on this circuit is in the Schematic Macro Information section and in the application note AN76458 “PSoC 5LP High Voltage (120 – 240 VAC) Powerline Communication”. It can be disabled if the powerline has low noise such that the signal-noise-ratio (SNR) is >10 dB across all noise frequencies above 135 kHz. It is disabled by setting the Receive_Filter parameter to Disabled.

Amplifier - Optional

To boost small signals, the receiver uses a programmable gain amplifier (PGA). A higher gain improves input sensitivity, but it can also cause noise to saturate the input signal, which degrades the noise immunity. For systems that do not attenuate the signal significantly (smallest amplitude is always > 50 dBμV), this can be disabled by setting the Receive_Gain parameter to None.

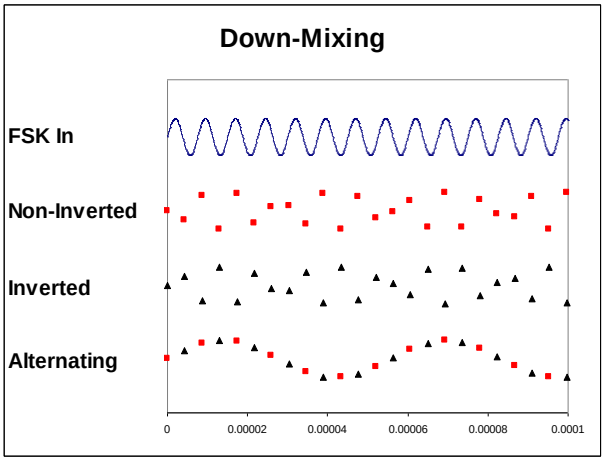
Sampler

A 12-bit SAR ADC samples the received signal. The sample rate determines the resulting down-mixed frequency. The next section describes how to select the sample rate.

Down-mixer with Image rejection and Band-pass Filter

This block shifts the FSK frequencies down to a lower frequency, which is easier to correlate back into the serial bit stream. The DFB Assembler component performs this down-mixing, which is triggered at the sample rate of the ADC and runs at the frequency of the device’s master clock (48 MHz in the code example). To down mix the signal, the ADC samples the signal at slightly less than two times the FSK frequency and the DFB Assembler inverts every other sample. This is shown in Figure 9.

Figure 6. Down-Mixing Timing Diagram



The Code tab of the DFB Assembler component contains the code for this inversion. The desired mixed down frequency range is 10 to 20 kHz. Based on simulations, these frequencies are low enough so that filtering and phase compensation are easy, but not so low that correlation is difficult.

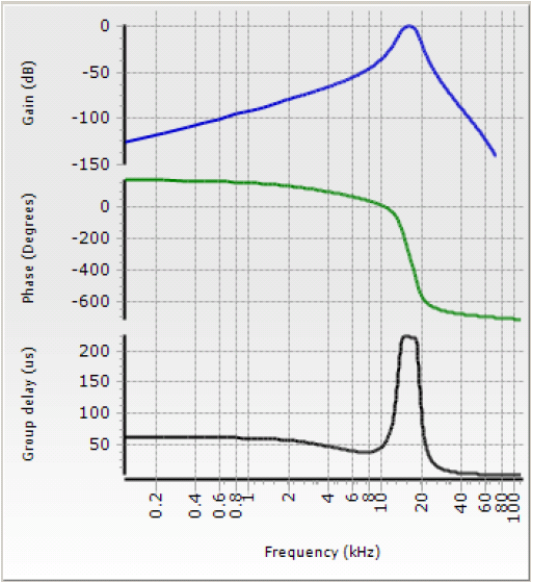
When the FSK frequencies are 130.434 kHz and 133.333 kHz, a sampling rate of 230.769 ksps yields mixed down frequencies of:

15.050 kHz: (130.434 – (230.769 / 2))

17.948 kHz: (133.333 – (230.769 / 2))

In addition to down-mixing, the DFB filters the signal to further attenuate any out-of-band noise. The filter is a wideband, constant delay biquad filter, which is documented in Blinchikoff & Zverev[1]. Figure 7 shows the response of the filter. The passband is in the range of 15 to 18 kHz, which is the band of the down-mixed FSK signals.

Figure 7. Band-pass Biquad Filter (I-phase)



One common problem with down mixing is that the mixer will mix the image of the FSK signal down to the same frequency as it will mix the FSK signal down to. The image frequency is the same distance apart from the mixer's local frequency as the FSK signal, but is on the other side. When the 130.434 kHz signal is mixed by the (230.769 kHz / 2) sampling frequency down to 15.050 kHz, the image would be:

$$fImage = 130.434 - (2 \times 15.050\ kHz) = 100.334\ kHz$$

When this signal is down-mixed, it will have the same frequency (15.050 kHz) as the intended FSK signal. This would make the system more susceptible to noise in this frequency (i.e. poorer signal-noise ratio). To prevent this, the receiver uses an image reject mixer. The image reject mixer samples two versions of the signal: an I phase and a Q phase (-90 degrees). Then, it shifts the Q phase signal by an additional -90 degrees. The I phase and Q phase samples use the same filter as in Figure 7, except the filter for the Q phase has an additional -90 degree phase shift. Since there is one ADC sample delay between the I sample and Q sample, the phase shift also includes this sampling time in addition to the 90 degrees.

The result is a signal that is exactly out of phase with the I phase. These two signals are subtracted from each other, which produces a signal with twice the amplitude of the original. This is shown in Figure 8. When the image frequency is the input, the Q signal lags (+90 degrees) the I signal. When the -90 degree shift is added, the resulting signal is exactly in phase with the I signal. The subtraction of these two signals is near zero (Figure 9).

Figure 8. Down Mixing of the Intended Frequency

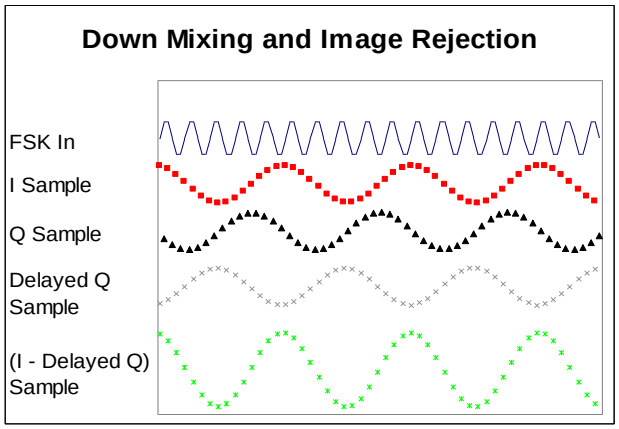
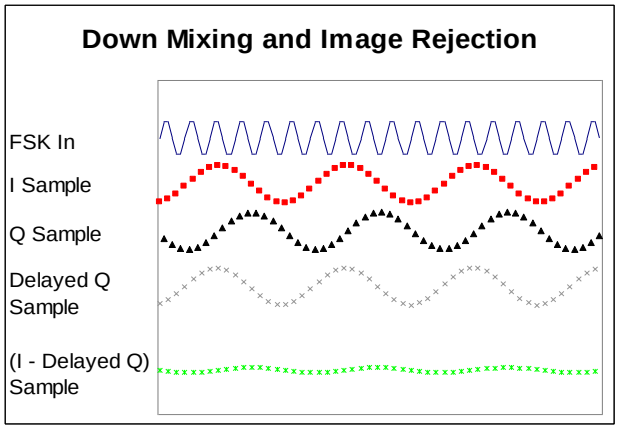


Figure 9. Down Mixing of the Image Frequency



To generate two phases, the receiver samples the input twice as fast (461.538 ksps to sample the 130-133 kHz waveform). The receiver samples each cycle of the FSK waveform approximately four times:

- I sample (non-inverted)
- Q sample (non-inverted)
- I sample (inverted)
- Q sample (inverted)

Since it takes one I sample and one Q sample to form the result, the output sample rate (230.769 ksps) is half of the input rate.

Correlator

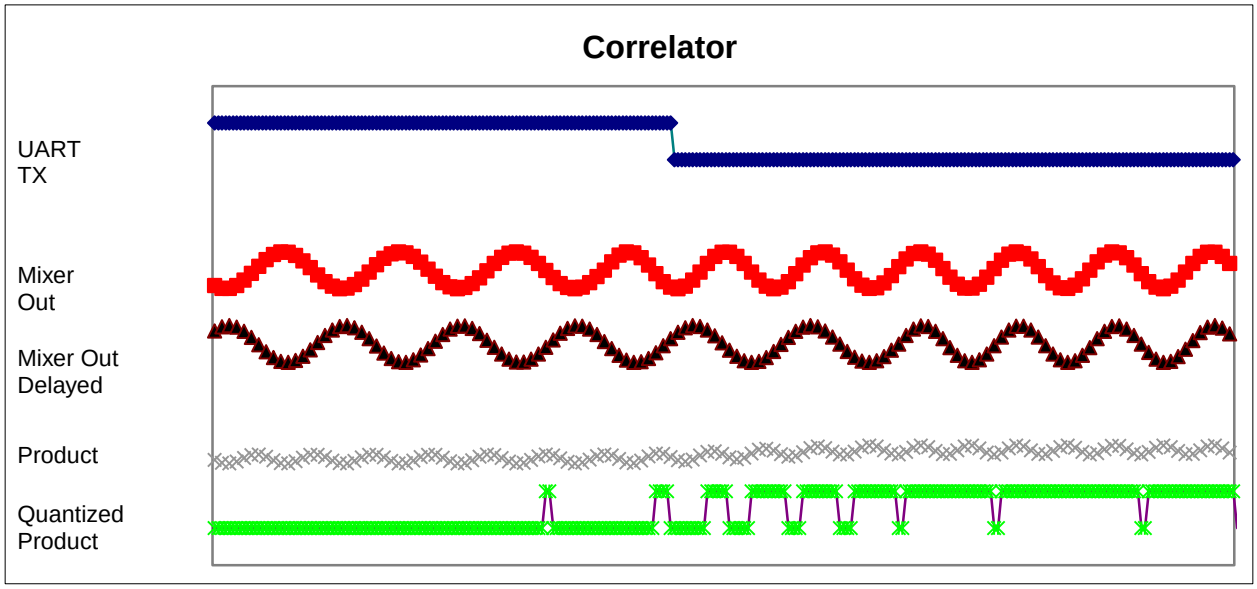
The correlator converts the down-mixed FSK signal back into the serial bit stream. Figure 10 is a timing diagram. First, the correlator uses only the most significant bit of the down-mixed signal from the DFB, which quantizes the signal to one bit. Then, the correlator delays this quantized signal by a fixed amount. These two signals (original and delayed) are multiplied, which generates the correlated signal. The delay is chosen such that the delayed version of the mixed down FSK '0' frequency is exactly in phase with the original, and the FSK '1' version is exactly out of phase. When the signals are in phase, the output is '0'. When they are out of phase, the output is '1'.

In the timing diagram, the correlator output signal looks similar to the original UART TX signal, except that it is inverted and there are occasionally some short pulses. These pulses are due to the quantization error of the sampled signal and the integer delay steps. The low-pass filter removes these pulses. The result is subtracted from 0x800000 so that it has the correct polarity.

The DFB performs correlation by:

- Storing each down-mixed sample in a circular buffer (the DFB's RAMA)
- Performing a multiplication (using the DFB's MAC) of the most recent down-mixed sample with the nth most recent sample that is stored in the DFB. The number n determines the amount of the correlator's delay (n * sample period). The multiplication acts like an XOR because the MAC is signed (positive * positive = positive, positive * negative = negative, negative * negative = positive).

Figure 10. Correlator Timing Diagram



Low-pass Filter

The DFB filters out the unwanted pulses in the correlator's output . The filter in the DFB is a low-pass Chebychev biquad filter with a 7 kHz cutoff. The DFB executes this filter for each correlator sample.

The cutoff frequency is chosen to pass all frequencies up to the fifth harmonic of the serial data. Since the data rate is 2.4 kbps, the equivalent clock frequency (0101 pattern) is 1.2 kHz and the fifth harmonic is 6 kHz.

At this point, the receiver has demodulated the FSK signal. The MSb (polarity) of the DFB is passed to a control register via DMA. The output of the control register drives the serial_out terminal.

DFB Coefficients

The values in Table 1 are the Digital Filter Block coefficients for the receive input filter. The first four groups of numbers in each column are for the Blinchikoff and Zverev[1] filter and the last group of numbers in each column is for the (90 degree minus one sample) phase shift. The filter coefficients are the same for both phases.

These numbers are converted into the DFB code by multiplying by 2^22 and converting to hexadecimal. The negative numbers are subtracted from 2^24.

Table 1. DFB Coefficients of Blinchikoff & Zverev Biquad Filter with -90 Degree Phase Shift

I Phase	Q Phase
0.046806574	0.046806574
-0.093613386	-0.093613386
0.046806574	0.046806574
-1.78692174	-1.78692174
0.926942348	0.926942348
0.017025948	0.017025948
0.034053564	0.034053564
0.017025948	0.017025948
-1.702949047	-1.702949047
0.903810263	0.903810263
0.010339975	0.010339975
0.020679951	0.020679951
0.010339975	0.010339975
-1.735077381	-1.735077381
0.903748274	0.903748274

Table 1. DFB Coefficients of Blinchikoff & Zverev Biquad Filter with -90 Degree Phase Shift

I Phase	Q Phase
0.074727535	0.074727535
0.149455309	0.149455309
0.074727535	0.074727535
-1.689572096	-1.689572096
0.927065849	0.927065849
-0.84722662	-0.456362963
1	1
0	0
-0.84722662	-0.456362963
0	0

The filter coefficients in Table 2 are for the low-pass Chebychev biquad filter with a 7 kHz cutoff (for a 115.384 ksps sample rate).

Table 2. DFB Coefficients of Chebychev Low Pass Filter

Low Pass Filter
0.002629995
0.005259752
0.002629995
-1.757896662
0.812820673
0.016414881
0.032829523
0.016414881
-1.765179396
0.872582436
0.010176897
0.020353556
0.010176897
-1.761868

Table 2. DFB Coefficients of Chebychev Low Pass Filter

Low Pass Filter
0.780933142
0.146869421
0.29373908
0.146869421
-1.805193663
0.953766346

Performance

The performance of the FSK Demodulator with the recommended coupling circuit is provided in the application note AN76458 “PSoC 5LP High Voltage (120 – 240 VAC) Powerline Communication”.

References

1. “Filtering in the Time and Frequency Domains”, Herman J. Blinchikoff & Anatol I. Zverev, SciTech Publishing, 2001

Component Changes

This section lists the major changes in the component from the previous version.

Version	Description of Changes	Reason for Changes / Impact
1.0	First release.	First release.

© Cypress Semiconductor Corporation, 2015. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control, or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

PSoC® is a registered trademark, and PSoC Creator™ and Programmable System-on-Chip™ are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

Any Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.



PRELIMINARY <remove this label when the component is production ready>