FSK Generation Using the Scenix SX Microcontroller

Chris Fogelklou, Scenix Semiconductor, Inc.

Introduction

This document describes the use of a Scenix SX microcontroller to perform FSK, or frequency-shift keying modulation. FSK is an early form of modem communication techniques. The source code this document describes is called simple_fsk_gen_1_03.src.

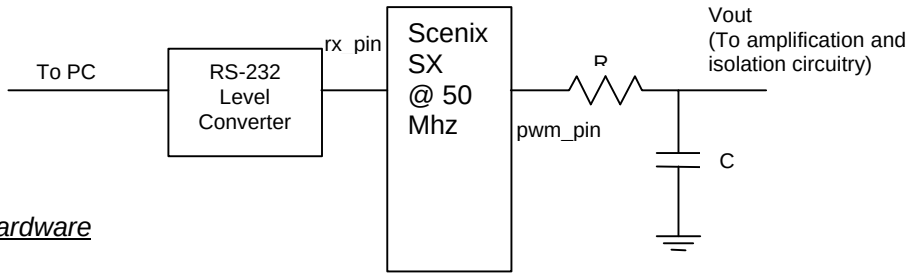
In the past, such telephony functions as FSK (frequency-shift keying) generation and detection, DTMF (dual-tone, multi-frequency) dialing generation and detection, and Caller ID could not be implemented with an 8-bit embedded MCU because performance levels were not high enough to support them. As a result, either a custom MCU had to be designed or a 16- or 32-bit device used. Now, the 8-bit Scenix Semiconductor SX Series MCUs, which have performance reaching 100 MIPS (million instructions per second) and a deterministic interrupt architecture, overcome this roadblock by providing the ability to perform these functions in software.

Unlike other MCUs that add functions in the form of additional silicon, the SX Series uses its industry-leading performance to execute functions as software modules, or Virtual Peripherals. These are loaded into a high-speed (10 ns access time) on-chip flash/EEPROM program memory and executed as required. In addition, a set of on-chip hardware peripherals is available to perform operations that cannot readily be done in software, such as timers, comparators, and oscillators.

One of the oldest protocols for modem communications is called FSK, or Frequency Shift Keying. With frequency shift keying, the modem converts a bitstream (1's or 0's) into a frequency which can be easily transmitted over telephone lines. Using the Scenix FSK solution, digital data received from the PC through the RS-232 interface is converted to analog signals for transmission over the telephone network using pulse-width modulation (PWM) techniques. To minimize I/O pin usage, a single PWM output pin is used. To ensure the smooth frequency shifts that are required by the FSK specification, all of the frequency shifts are phase-coherent. The FSK specification described in this document uses 1300Hz to represent a '1' and 2100Hz to represent a '0'. The maximum data rate for this type of modulation is 1200 baud.

To minimize code space and required processing power, this FSK modulation technique uses an artificial sine wave generator to simulate the characteristics of a real sin wave, without the use of a large lookup table. The sin wave generator utilizes the properties of gravity to create a near-perfect sin wave at the desired frequency.

Chris Fogelklou, Scenix Semiconductor, Inc.



The Hardware

The hardware necessary to demonstrate the generation of FSK is extremely simple. A block diagram is shown below.

Depending on the maximum frequency you wish to obtain, you should adjust the component values for R and C to choose the resolution of the PWM. Ideally, you should calculate the maximum SIN frequency output you will use and choose the cutoff to be at this frequency. For instance, for a maximum output frequency of 2.1kHz, calculate R and C:

First, choose a value for R.

R=1000 ohms

Now, calculate C:

C = 1/(2 * pi * Cutoff Frequency * R)

Therefore:

C = 1/(2 * 3.14 * 2100Hz * 1000 ohms)

And

C = 0.076uF

The software is designed to run on the Scenix DTMF demo boards. To build and test the demo yourself, these are the pins to which your hardware should be connected:

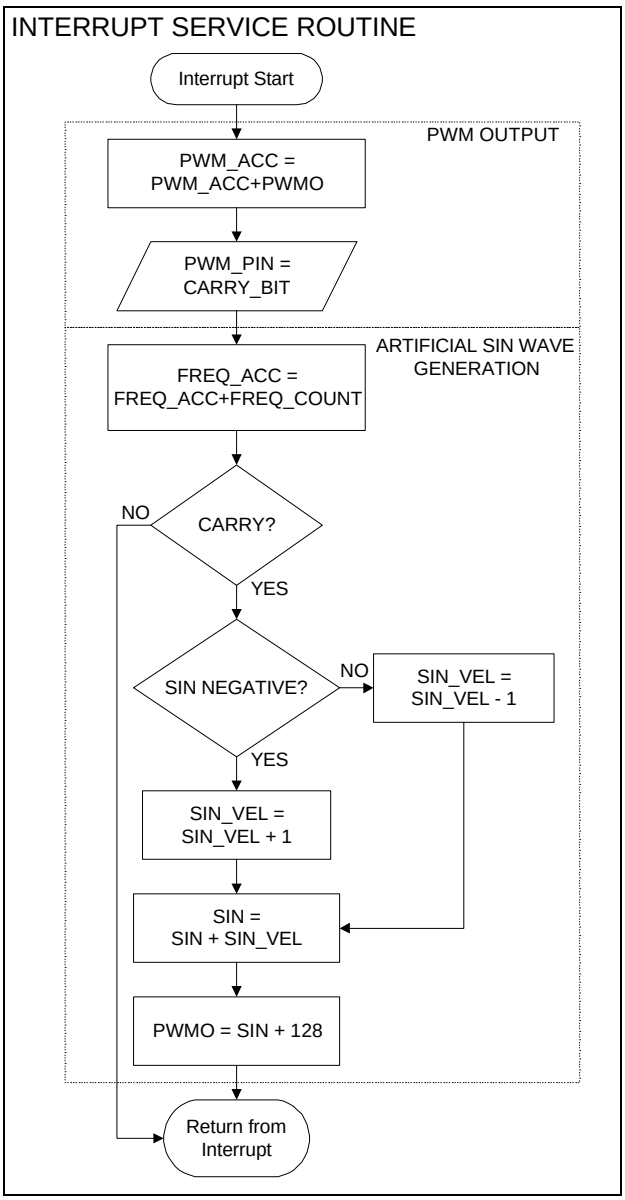
```
PWM_pin     equ     ra.0    ; Pin used for PWM output
rx_pin      equ     ra.1    ; Pin used for rs-232 receive
tx_pin      equ     ra.2    ; Pin used for rs-232 transmit
led_pin     equ     rb.0    ; For visual indication of transmitted data
```

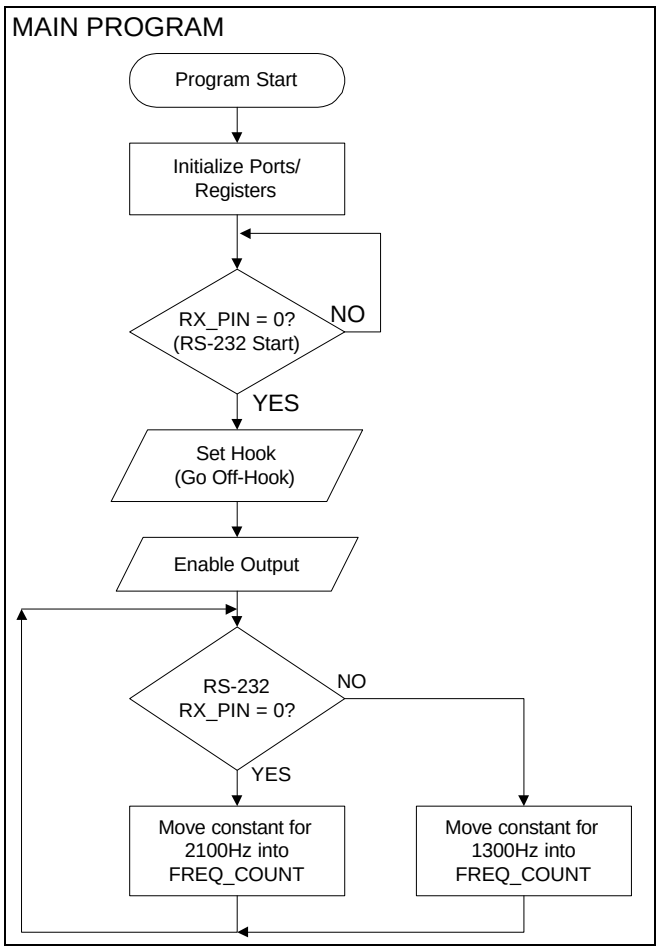
For more information, there is documentation in the source code on how to use the source code, and how to generate other modulation frequencies.

Chris Fogelklou, Scenix Semiconductor, Inc.

The Software

The software consists of an interrupt service routine and a mainline routine. The two run independently, with the mainline routine loading the frequencies for the interrupt service routine to output, depending on the state of the RS-232 input pin.





This software has been tested using the Scenix FSK receive modem software, which has been tested with a production BELL202 modem and with CallerID signals from the telephone company.

This is an extremely simple method to perform FSK modulation. To make the system more powerful, an RS-232 UART could be added so the incoming RS-232 data can be processed (Required to implement an AT-command set). To create a full modem solution, this software module must be combined with the FSK receive module and the DTMF transmit (Dialing) module.

For more information on frequency generation, FSK reception, or DTMF generation, consult the documentation at www.scenix.com

Chris Fogelklou, Scenix Semiconductor, Inc.

Documentation from the source code:

```
;       Designed for use with Scenix Modem Rev 1.2
;
;       Filename:       simple_fsk_gen_1_03.src
;
;       Authors:        Chris Fogelklou
;                       Applications Engineer
;                       Scenix Semiconductor, Inc.
;
;       Revision:       1.03
;
;       Part:           SX28AC datecode 9929AA/ SX52BD datecode AB9919AA
;       Freq:           50Mhz
;
;       Compiled using Parallax SX-Key software v1.07 and SASM 1.40
;
;      Date Written:   December 9, 1998
;       Last Revised:  November 22, 1999
;
;       Program Description
;       This program demonstrates a simple FSK modulation scheme.  It contains
;       no UART, so the baud rate of the FSK modulation is set by the external
;       Terminal's UART.  It simply converts high data to a low frequency
;       and low data to a high frequency.  The analog voltage is generated by
;       a 1-pin PDM D/A, and the sine wave values are generated by an artificial
;       sine wave generator.  This program could be improved by adding a UART to
;       the program, and sending the data out after it has been received.  Using
;       this method, the data can be manipulated.
;
;       The FSK modulation constants for other FSK standards besides bell202 are
;       also supported, but most require a baud rate generator and flow control.
;
;       BELL202:          Supported with other BELL202 Modems.
;       V.23 Answer Mode:  Supported using a 1200bps connection to the modem board,
;                         but won't be autodetected because it must output 2100Hz
;                         answer tone for 3 seconds after picking up.
;       V.23 Originate Mode: Can't be supported because requires a 75bps baud rate
;                         generator.
;       BELL103 Originate Mode: Supported using a 300bps connection to the board.
;                         must use another device to dial out.
;       BELL103 Answer Mode: Supported using a 300bps connection to the modem board,
;                         but can't be auto-detected for the same reasons as the
;                         V.23 Answer mode.
;       Uncomment the desired modulation type.
;
BELL202
;V23_ORIGINATE_MODE
;V23_ANSWER_MODE
;BELL103_ORIGINATE_MODE
;BELL103_ANSWER_MODE
;
;       Revision History:
;       1.01 & 1.02:  Documentation Updates.
;       1.03: Updated for new SX template, and Modem Boards, Rev 1.2
;
;       Pins Used:
;              PDM_pin        equ     ra.0    ; D/A output pin, connect to filter
;                                             ; circuitry
;              rx_pin         equ     ra.1    ; RS-232 reception pin
;              tx_pin         equ     ra.2    ; RS-232 transmission pin, received
;                                             ; characters
;                                             ; are echoed on this pin.
;              led_pin        equ     rb.0    ; LED pin... Flashes when characters are
;                                             ; received
;              hook           equ     rb.4    ; drive hook low to go off-hook
;              cts            equ     rb.7    ; indicates to the PC that the SX
;                                             ; is ready to receive data.
```