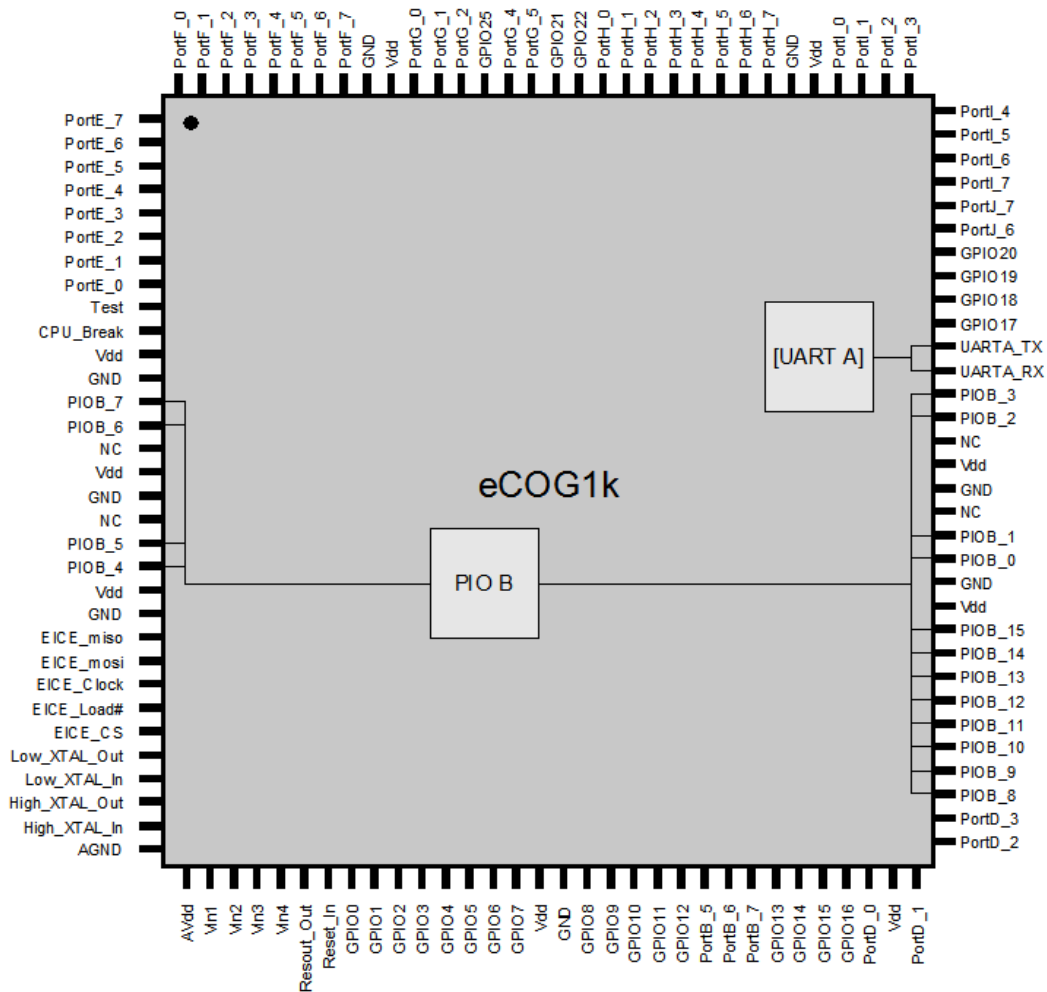


AN044 – Generating DTMF and FSK Signals

Version 1.1

This application note describes a simple for generating Dual Tone Multiple Frequency (DTMF) and Frequency Shift Keying (FSK) signals using the eCOG1k microcontroller.



Confidential and Proprietary Information

©Cyan Technology Ltd, 2008

This document contains confidential and proprietary information of Cyan Technology Ltd and is protected by copyright laws. Its receipt or possession does not convey any rights to reproduce, manufacture, use or sell anything based on information contained within this document.

Cyan Technology™, the Cyan Technology logo and Max-eICE™ are trademarks of Cyan Holdings Ltd. CyanIDE® and eCOG® are registered trademarks of Cyan Holdings Ltd. Cyan Technology Ltd recognises other brand and product names as trademarks or registered trademarks of their respective holders.

Any product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given by Cyan Technology Ltd in good faith. However, all warranties implied or expressed, including but not limited to implied warranties of merchantability, or fitness for purpose, are excluded.

This document is intended only to assist the reader in the use of the product. Cyan Technology Ltd shall not be liable for any loss or damage arising from the use of any information in this guide, any error or omission in such information, or any incorrect use of the product.

This product is not designed or intended to be used for on-line control of aircraft, aircraft navigation or communications systems or in air traffic control applications or in the design, construction, operation or maintenance of any nuclear facility, or for any medical use related to either life support equipment or any other life-critical application. Cyan Technology Ltd specifically disclaims any express or implied warranty of fitness for any or all of such uses. Ask your sales representative for details.



Revision History

Version	Date	Notes
V1.0	10/10/2006	First release.
V1.1	23/10/2006	Added references to application notes on other analogue output methods.

Contents

1	Introduction	5
2	Glossary	5
3	Overview.....	6
4	Hardware	7
5	Software	8
5.1	Generating Sinewaves	8
5.2	Output Signals.....	8
5.2.1	<i>Using the Hardware Timer</i>	9
5.2.2	<i>Percentage Error</i>	9
5.2.3	<i>DTMF Output</i>	10
5.2.4	<i>FSK Output</i>	10
5.2.5	<i>Tone Duration</i>	10
6	Example Application	11

1 Introduction

This application note describes a simple for generating Dual Tone Multiple Frequency (DTMF) and Frequency Shift Keying (FSK) signals using the eCOG1k microcontroller.

This example uses the parallel output port PIOB with external passive components to implement a simple and low-cost digital-to-analogue converter. The application software demonstrates the transmission of different keys and characters, as used on a telephone keypad, in DTMF and FSK formats.

The example code is available as a project for CyanIDE development and runs on the eCOG1k development board.

2 Glossary

A table of abbreviations used in this document.

ADC	Analogue to Digital Converter
DAC	Digital to Analogue Converter
DTMF	Dual Tone Multiple Frequency
eCOG1	Cyan Technology target micro controller
FSK	Frequency Shift Keying
PIO	Parallel Input/Output
UART	Universal Asynchronous Receiver/Transmitter

3 Overview

Dual-tone multiple-frequency (DTMF) signalling is used in telephony applications for sending information, usually key presses, over the telephone line to the exchange or call switching centre. The information must be encoded into signals in the voice-frequency band, as higher frequencies cannot be transmitted through a traditional telephone system.

A typical telephone keypad has 12 to 16 keys, arranged in a 4x3 or 4x4 array. DTMF encoding assigns two frequencies to each key; one frequency indicates the row and the other frequency indicates the column within the keypad matrix. In this way, a 4x4 keypad requires eight frequencies to encode all 16 keys.

In DTMF encoding, the eight different frequencies include four high band frequencies for the columns and four low band frequencies for the rows in the keypad. When a key is pressed, the low band frequency for the row and the high band frequency for the column are transmitted simultaneously.

DTMF keypad frequencies				
	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

Figure 1. DTMF telephone keypad frequencies

Frequency-shift keying (FSK) is a simpler system for encoding binary data. It is a form of frequency modulation in which the modulating signal selects one of two preset output frequencies. FSK signalling is used in Caller ID (CID) systems. Frequencies commonly used in FSK signalling are 1200Hz for '0' and 2200Hz for '1', or 1300Hz for '0' and 2100Hz for '1'.

4 Hardware

Analogue output signals can be generated in various ways, with different costs, performance (accuracy, resolution, frequency range) and software requirements. Examples include:

- Digital to analogue converter (DAC).
- PWM output with external low-pass filter.
- Parallel output port with R-2R ladder.

Application note [AN031](#) provides an example of how to drive an external DAC with the SPI port. Application notes [AN023](#) and [AN033](#) use a PWM timer for analogue output of audio signals. The example described here uses a passive R-2R ladder.

Digital to analogue converters are rarely available on low-cost microcontrollers. They can provide high performance with minimal software overhead, but if the DAC function is not available on-chip, a solution using an external DAC device can increase the system cost.

Digital PWM outputs can generate analogue signals with the addition of an external low pass filter. The quality of the output signal depends on the PWM carrier frequency and the design of the output filter. Active filters using operational amplifiers can provide high performance filtering in a small space, but again can increase the system cost. Simple passive filters are cheap but less effective and often have high output impedance.

A parallel port can be used as a simple, low-resolution digital to analogue converter with the addition of an R-2R ladder circuit. This requires only low-cost passive components, and forms the basis of the example described here. This circuit also includes a low pass filter.

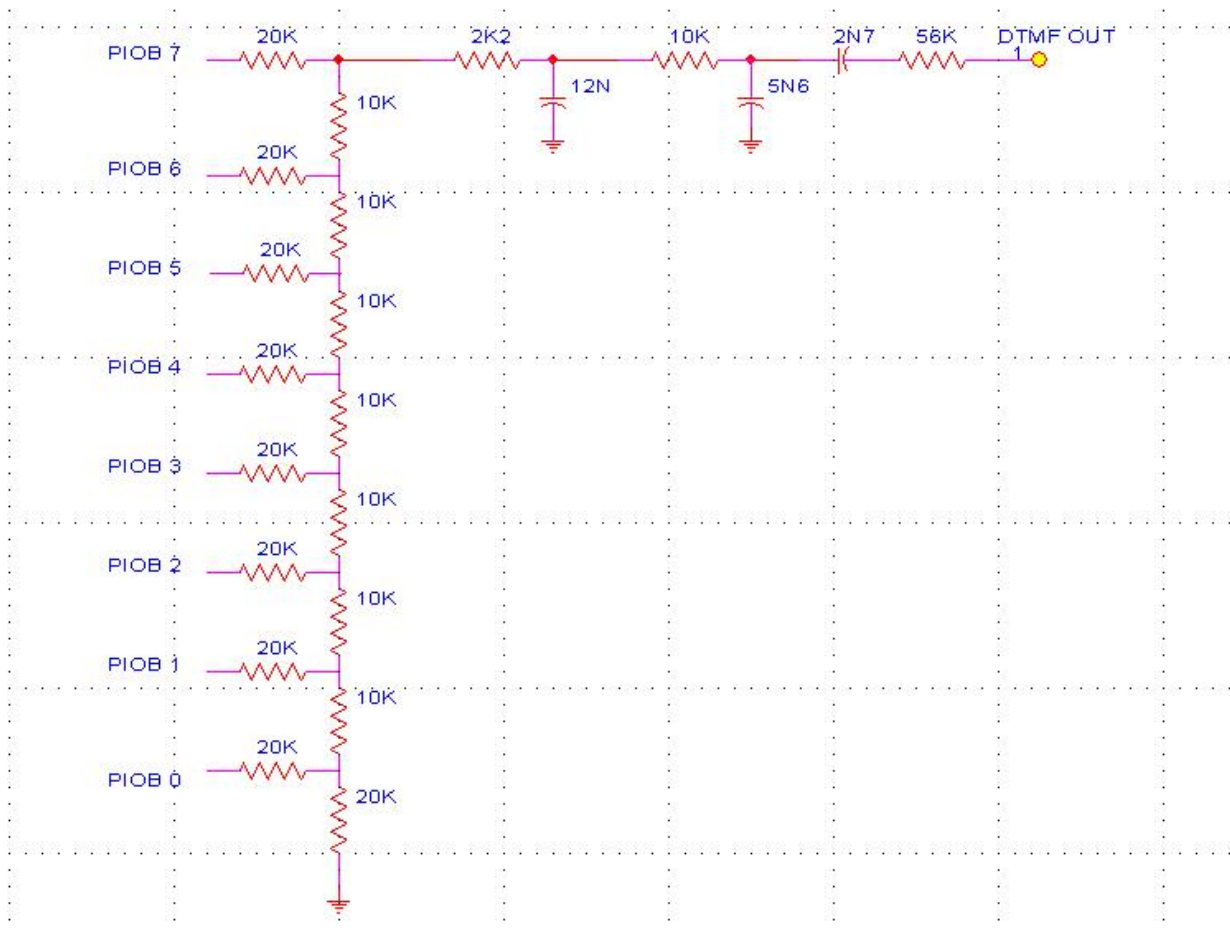


Figure 2. R-2R ladder circuit

5 Software

5.1 Generating Sinewaves

Both DTMF and FSK schemes require the generation of sine wave signals. These may be generated in several ways; for most microcontrollers there is a trade-off between the data space required, the amount of software required, and the execution time.

This example application uses different wave lookup tables to output sine values with different frequency efficiently. Wave tables could be pre-calculated and stored in flash memory. All required power level will be pre-calculated and stored in the flash memory which helps implement easier and use less processing power.

DTMF requires that the high band frequencies have a power level 2dB above the low band frequencies in order to compensate for the increased power loss at higher frequencies in the transmission medium and any low pass filtering. Therefore, an adjustment to the power level is added to keep the high band frequency 2dB above the low band frequency at the output.

FSK requires that the different frequencies have the same power level. Therefore, an adjustment to the output signal power level is added only to compensate for any losses in the low pass filter.

Freq (Hz)	Twist (dB) Required for DTMF	Adjust (dB) Compensation for low pass filter	Total added power (dB)
697	0	0	0
770	0	0.4	0.4
852	0	1	1
941	0	1.4	1.4
1209	2	3	5
1336	2	4	6
1477	2	4.6	6.6
1633	2	5.5	7.5

Table 1. Power level compensation table

5.2 Output Signals

DTMF and FSK both require multiple frequencies and different frequency ranges. A simple way to control the output frequency is to use a hardware timer to generate interrupts at a fixed update rate, and then read from different frequency wave table to produce corresponding frequency.

The values from tables are output as unsigned quantities. A hardware timer is used to control the output updates. On each timer interrupt, the next value of is read from the required frequency' lookup table and output through the PIO port.

Frequency	Number of cycles		Number of points										
1477	1		13										
Sample	1	2	3	4	5	6	7	8	9	10	11	12	13
Output	0	9	33	68	104	135	152	152	135	104	68	33	9

Table 2. Example of 1477Hz lookup table

5.2.1 Using the Hardware Timer

In this example, one of the on-chip general purpose timers is configured to generate interrupts at a rate of 19200Hz. On each interrupt, the processor updates its index into the wave table, and reads the value from the required frequency table at the new index.

For example, according to Table 2:

At time 0, the first sample with a value of 0 is output via the PIO. After 1/19200 second, the processor updates the index into the table, reads the next value (=9), and it is output via the PIO. After another 1/19200 second, the next value (=33) is output. This produces a reasonable approximation to a sine wave at 1477Hz on the output pin, after the low pass filters.

5.2.2 Percentage Error

Using a fixed update time interval, it is not possible to generate all the required frequencies exactly with a small lookup table, because an exact number of samples does not always represent one single sine wave period. Increasing the size of the table such that the total number of samples is close to a whole number of periods can reduce the error.

The following table lists the actual frequencies achieved using this example, and the errors between these and the ideal frequencies.

DTMF frequency (Hz)	Number of sine wave periods	Number of data points	Number of data points used in the code	Actual frequency (Hz)	% Error
697	2	55.1	55	698.2	0.17
770	1	24.9	25	768.0	-0.26
852	2	45.1	45	853.3	0.16
941	2	40.8	41	936.6	-0.47
1209	1	15.9	16	1200.0	-0.74
1366	3	42.2	42	1371.4	0.40
1477	1	13.0	13	1476.9	-0.005
1633	4	47.0	47	1634.0	0.06

Table 3. DTMF frequencies and errors

DTMF frequency (Hz)	Number of sine wave periods	Number of data points	Number of data points used in the code	Actual frequency (Hz)	% Error
1200	1	16.0	16	1200.0	0.0
1300	4	59.1	59	1301.7	0.13
2100	5	45.7	46	2087.0	-0.62
2200	4	34.9	35	2194.3	-0.26

Table 4. FSK frequencies and errors

5.2.3 DTMF Output

For DTMF output, both the high and low frequencies for a pressed key are output simultaneously. The level of the high band frequency is required to be 2dB above that of the low band frequency; this is to compensate for losses in transmission. In the precalculated wave table, 2dB is already added into the high band frequency. The output value is calculated simply by reading the values from the two tables for the high and low frequencies and adding both values together.

5.2.4 FSK Output

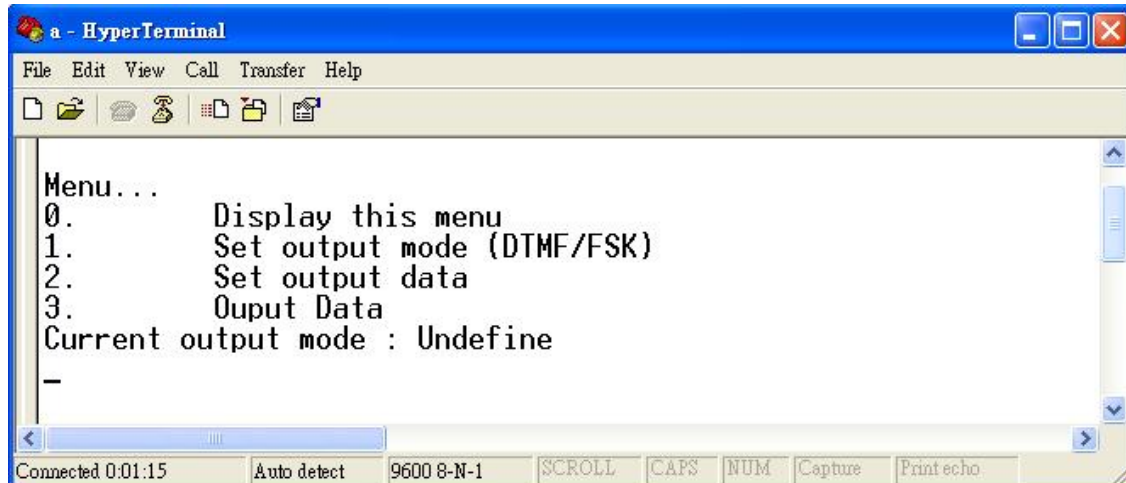
For FSK output, only one of the two possible frequencies is generated at one time. The signal level is the same for both frequencies, and therefore no scaling is required. The output values from the lookup table can be used directly and written to the parallel output port PIOB.

5.2.5 Tone Duration

In both DTMF and FSK schemes, the duration of the output tones is controlled by using the same hardware timer to count tick periods, for both tones and gaps between tones. This example uses a tone and gap duration of 1152 ticks = 60ms ($1152 / 19200 = 0.06\text{s}$). It is defined by the symbol `DTMF_OFF_TIME` in the include file `dtmf_fsk.h`.

6 Example Application

The example project for this application note demonstrates using one of the PIO ports with an external R-2R ladder circuit to generate simple DTMF and FSK signals. The example is controlled via a serial port at 9600 baud; the port settings are 8 data bits, 1 stop bit and no handshaking. The picture below shows the output from the serial port with a simple menu selection.



```
a - HyperTerminal
File Edit View Call Transfer Help
Menu...
0.      Display this menu
1.      Set output mode (DTMF/FSK)
2.      Set output data
3.      Ouput Data
Current output mode : Undefined
-
```

Connected 0:01:15 Auto detect 9600 8-N-1 SCROLL CAPS NUM Capture Print echo

Set the output mode to either DTMF or FSK. Then enter some numbers to be transmitted. The application generates the required tones to send the numbers as DTMF or FSK signals.



```
a - HyperTerminal
File Edit View Call Transfer Help
Menu...
0.      Display this menu
1.      Set output mode (DTMF/FSK)
2.      Set output data
3.      Ouput Data
Current output mode :Transmitting:123
DONE!!!
-
```

Connected 0:08:46 Auto detect 9600 8-N-1 SCROLL CAPS NUM Capture Print echo