AN1523

Sine Wave Generator Using Numerically Controlled Oscillator Module

Author: *Vinaya Skanda*
Microchip Technology Inc.

INTRODUCTION

A key requirement in most applications is the ability to generate and control waveforms at various frequencies. Most common demands for such sources are industrial test setups for providing frequency stimulus, communication equipment with low-noise requirements, or medical testing devices.

The Direct Digital Synthesis (DDS) technique is gaining wide popularity and acceptance from the industrial community to achieve programmable analog outputs with accuracy and high resolution. The traditional Pulse-Width Modulation (PWM), which is commonly referred to as the poor man's Digital-to-Analog Converter (DAC) was previously being used for this purpose. The PWM method has the limitation of generating arbitrary waveforms in low-frequency ranges, which is overcome using the DDS technique.

This application note focuses on the use of the Numerically Controlled Oscillator (NCO) module for designing a Sine Wave Generator. The NCO module uses the DDS technique for generating waveforms, and is available on various PIC16F family and PIC10F320/322 family of MCUs. For more information on other Core Independent Peripherals refer to www.microchip.com/CIP.

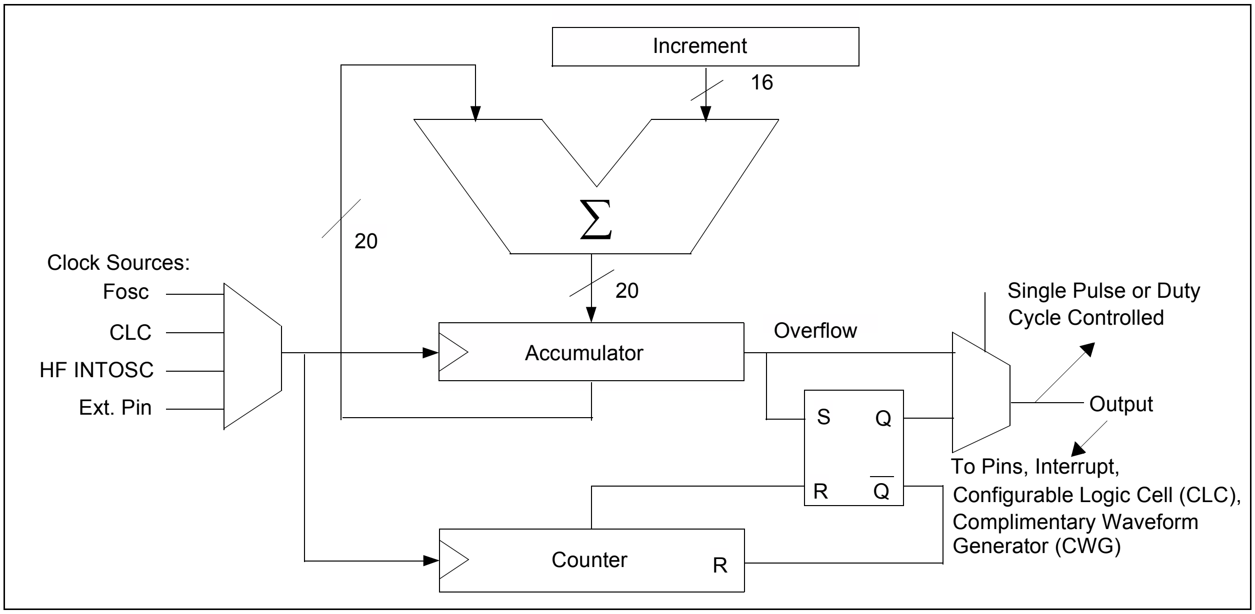
DIRECT DIGITAL SYNTHESIS (DDS) AND NCO

Direct Digital Synthesis is a technique of generating an analog waveform, generally of sinusoidal wave shape from a time varying signal in its digital form and a DAC.

The NCO module operates on the principle of DDS by repeatedly adding a fixed value to an accumulator. The accumulator is 20 bits in length and additions occur at the input clock rate, which can be a maximum of about 16 MHz. The accumulator will overflow with a carry bit set periodically, and this will produce a transition in the output of the NCO module.

The NCO module can operate in two modes: fixed duty cycle PWM and frequency controlled Pulse mode. With such an arrangement, the response will be very linear across a wide range of frequencies, ranging from 0 kHz up to 500 kHz using a clock of 16 MHz. The frequency resolution that can be obtained is precise and is in steps of 15 Hz across this entire frequency range. The linear frequency control and the increased frequency resolution are the key distinguishing factors when compared to the traditional PWM-based frequency control. Figure 1 illustrates the internal block diagram of the NCO module.

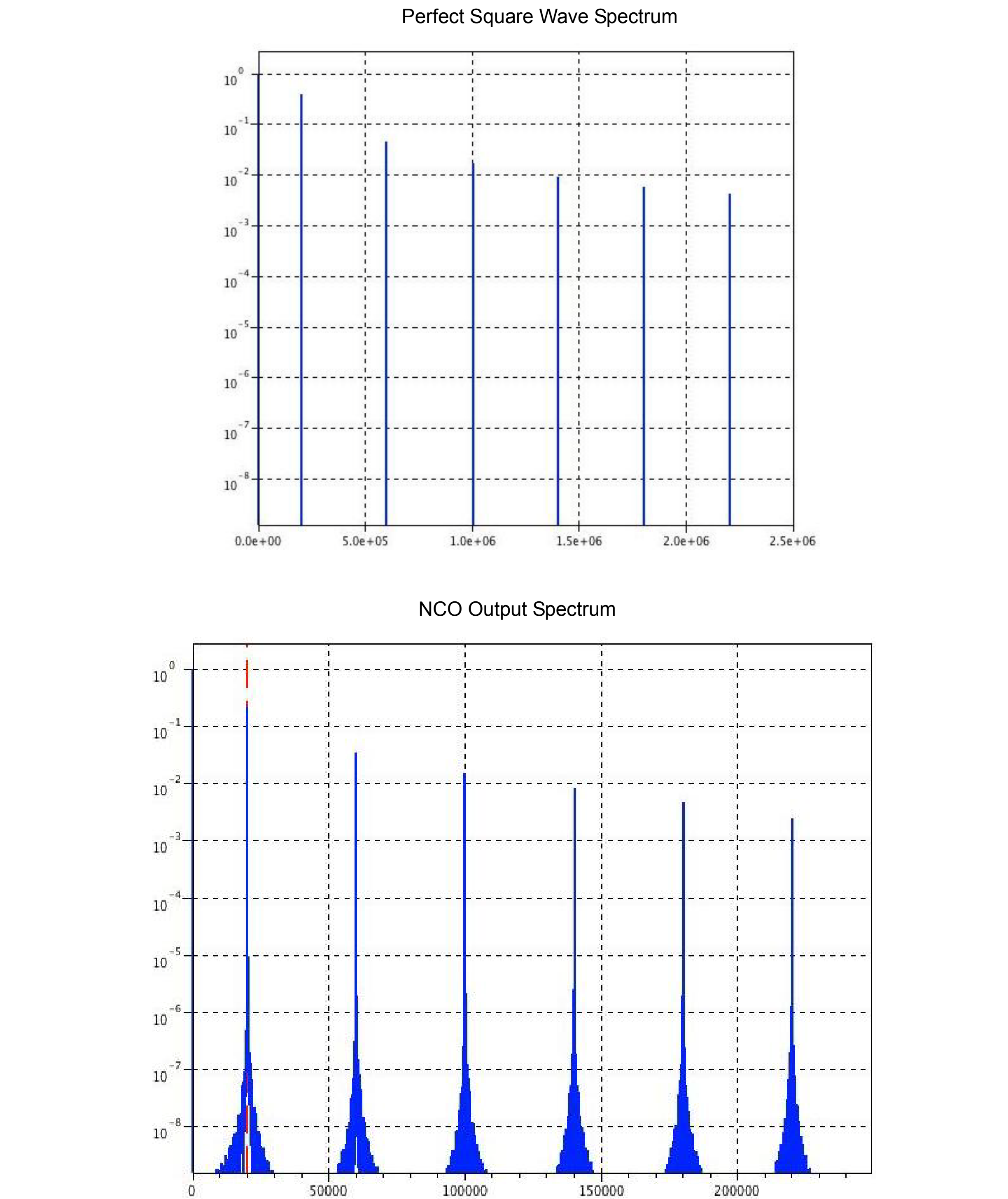
FIGURE 1: INTERNAL BLOCK DIAGRAM OF NCO MODULE



The NCO module generates precisely controllable output frequencies using the DDS technique. The DDS technique essentially provides a clock with carefully controlled jitter on it. Therefore, it is necessary that the signal be aggregated on the frequency domain.

Figure 2 illustrates the typical output spectra when generating 50% duty cycle square wave using the NCO module. The sideband noise generated by the jitter is insignificant in comparison to the fundamental frequency. When plotted on a logarithmic scale, the NCO output compares to that of a perfect square wave.

FIGURE 2: PERFECT SQUARE WAVE SPECTRUM COMPARED TO NCO OUTPUT SPECTRUM



NCO Output and PWM Output Comparison

This section provides the comparison between the NCO module and the traditional PWM module. When using a PWM module to generate a pulse train with variable frequency, use Equation 1 to calculate the PWM frequency.

EQUATION 1: PWM FREQUENCY CALCULATION

$$F_{PWM} = \frac{F_{OSC}}{4(PR2 + 1)}$$

Where,

F_{PWM} = Desired frequency of PWM

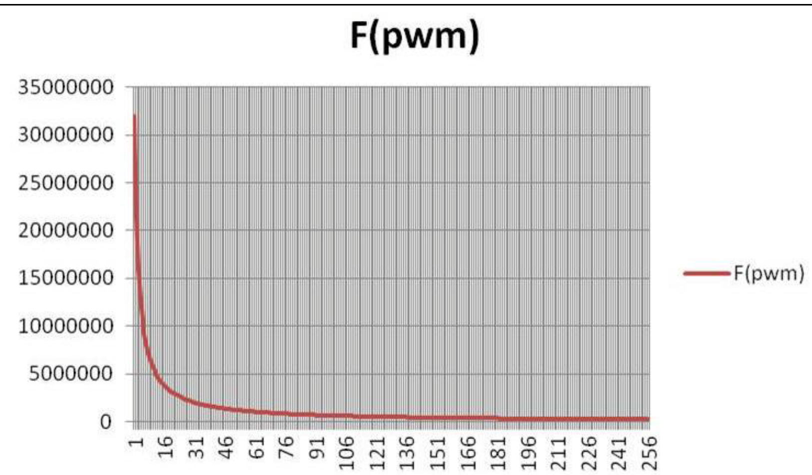
F_{OSC} = Oscillator clock frequency

$PR2$ = Period register to be loaded

For an 8-bit PR2 register, the value can vary from 0 to 255. With the oscillator clock frequency being fixed, the value in the PR2 register determines the frequency of the PWM output. Since the PR2 register value forms the denominator in Equation 1, any change in the value of $PR2$ will not yield a linear variation of F_{PWM}, although the incremental change in the denominator or $PR2$ is linear.

Figure 3 illustrates the variation of PWM frequency with respect to a corresponding change in the $PR2$ value.

FIGURE 3: FREQUENCY vs PR VALUE IN PWM MODULE



The relation between the frequency of the NCO output and the incremental register is provided in Equation 2. From Equation 2, note that F_{NCO} is directly proportional to the increment value, and the accumulator overflow value is always fixed to 2^{20} = 1048576. Therefore, any change in the increment value will yield a very linear variation in the output frequency of the NCO (i.e, F_{NCO}).

EQUATION 2: PWM FREQUENCY CALCULATION USING NCO MODULE

$$F_{NCO} = \left(\frac{F_{OSC}}{Accumulator}\right)(IncrementValue)$$

Where,

F_{NCO} = Frequency of the output of NCO module

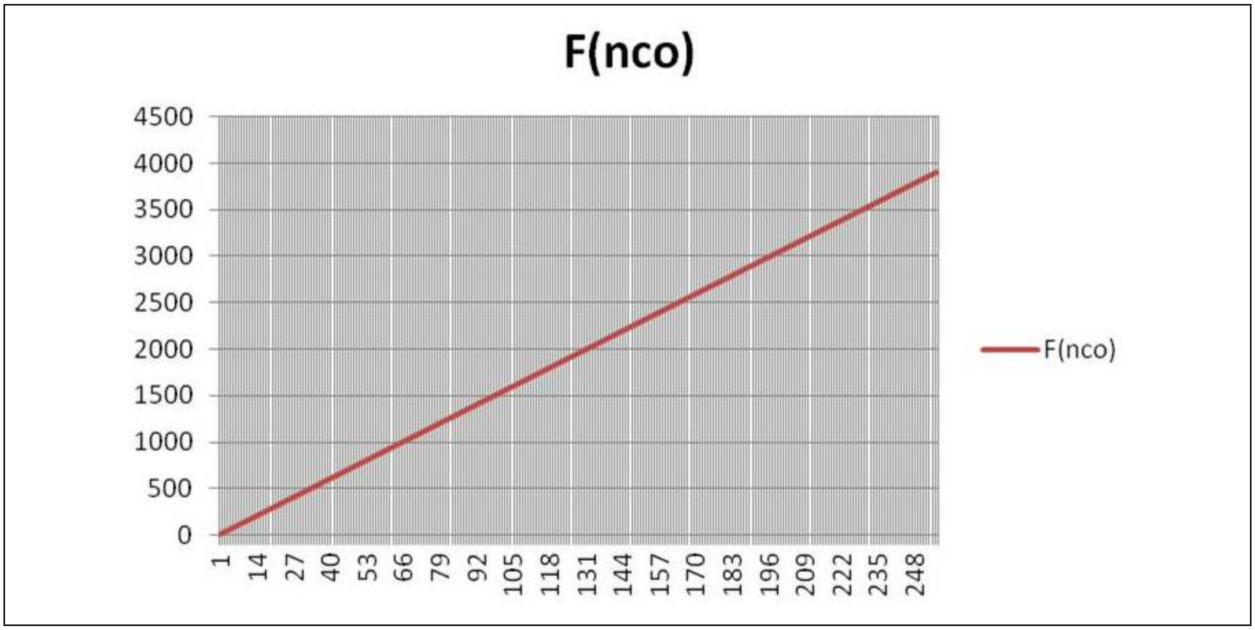
F_{OSC} = Oscillator clock frequency (about 16 MHz)

$Accumulator$ = 20 bit summing register that overflows to create an output transition

$Increment\ Value$ = Value loaded to change F_{NCO}

This concept illustrates the variation of F_{NCO} with a corresponding change in the increment value, see Figure 4.

FIGURE 4: FREQUENCY VERSUS INCREMENT VALUE IN NCO MODULE



Therefore, a better frequency resolution over a wide frequency range can be obtained using the NCO for waveform generation, when compared to the conventional PWM-based approach.

PRINCIPLE OF SINE WAVE GENERATION USING NCO MODULE

The output of the NCO module will be a square wave at the configured frequency. A square wave has many frequency components with the main frequency being the center frequency, as per the NCO configuration. A square wave could be generated by adding a series of pure tones (sine waves) with appropriate amplitude and phase as per the Fourier transforms.

Fourier theorem assumes that the user add sine waves of infinite duration. Therefore, a square wave is essentially composed of *Fundamental frequency-1/3 of third harmonic tone+1/5 of fifth harmonic tone-1/7 of seventh harmonic tone*, and so on (see Figure 5). The square wave output from the NCO can be passed through a Band Pass Filter with a high Q factor to generate a sine wave at the desired frequency.

FIGURE 5: FREQUENCY COMPONENTS IN SQUARE WAVE

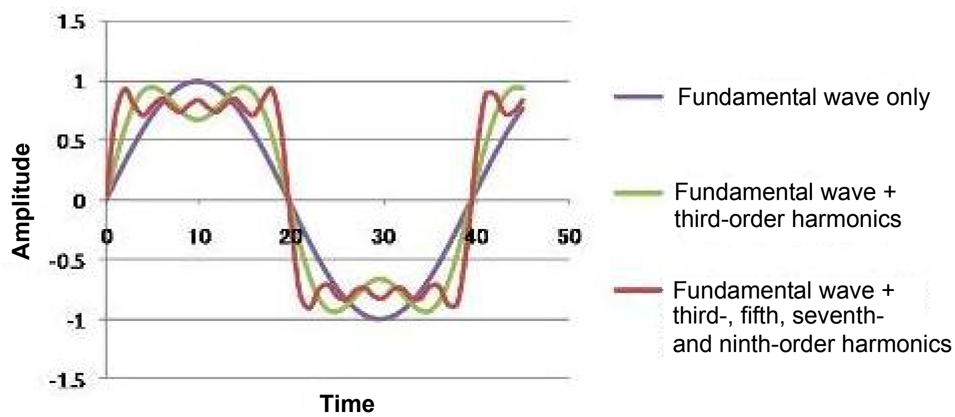
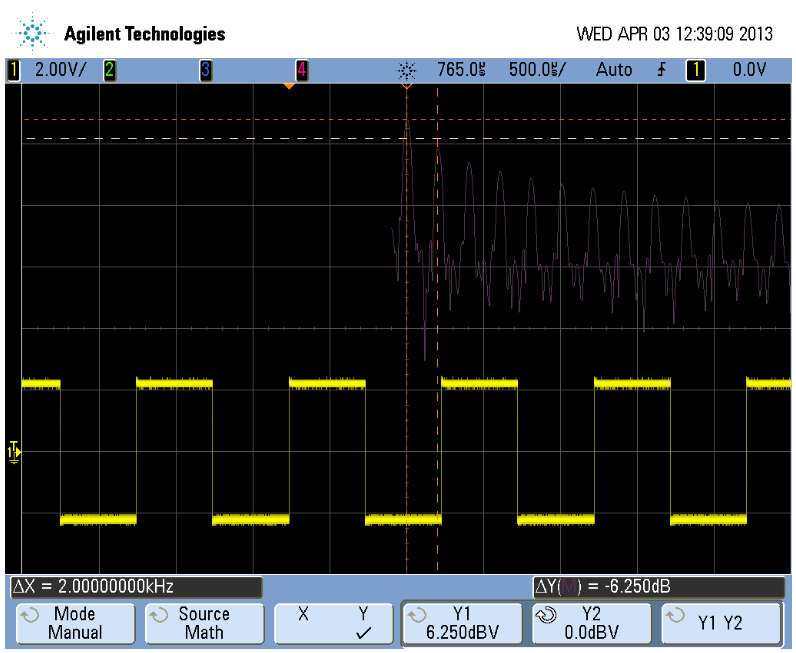


Figure 6 illustrates the frequency spectrum of a symmetric square wave observed using an oscilloscope.

FIGURE 6: FREQUENCY SPECTRUM OF A SQUARE WAVE



Applications of Sine Wave Generator

There are a wide variety of applications which necessitate the use of a sine wave. Some of the applications are as follows:

- Calibration of sound equipment or speakers
- Detection of frequency components in a signal
- Generate test tones for radio audio level alignment
- Radio tuning circuitry
- Reference tone generation to tune and adjust musical instruments
- Acoustic equalization and testing
- Creation of harmonics for generating multiple sound frequencies
- Sound card quality control
- White noise generator
- Hearing test equipment

CASE STUDY: INTRUDER DETECTION SYSTEM USING DISCRETE FOURIER TRANSFORM (DFT) BY CORRELATION

This section describes the application of a sine wave generator in computing correlation for a DFT-based intruder detection system.

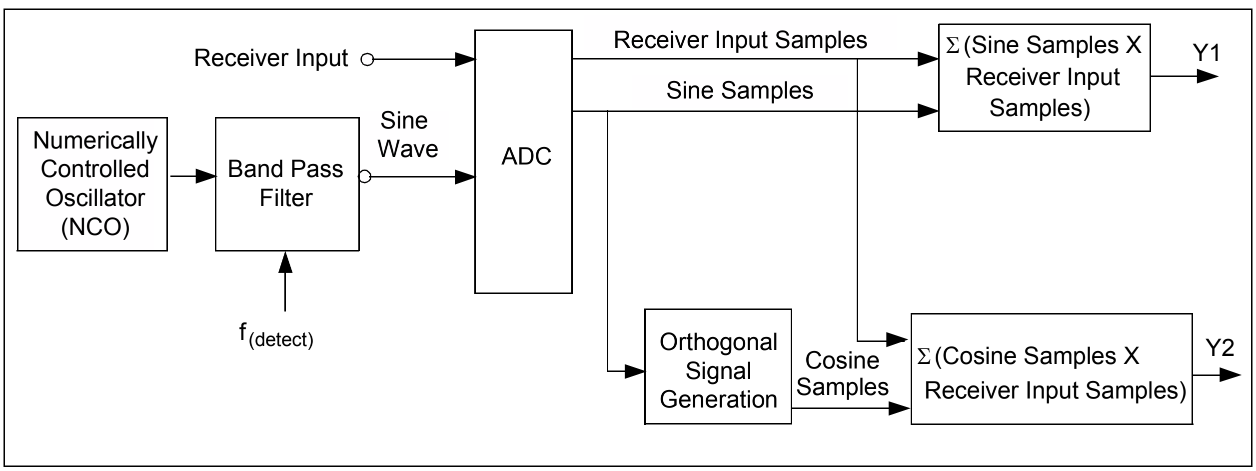
An intruder detection system basically consists of a transmitter which emits a signal at a specific frequency whenever an intruder is found inside a room or an enclosed space.

In most intruder-based systems, the Infrared (IR) signals are used. The receiver receives the signal and detects if any specific frequency component exists. In this detection process, usually a DFT is performed on the received signal and is checked for the presence of the frequency component of interest. When the DFT is implemented using the Correlation method, the sine and cosine waves are required at the frequency of interest.

The sine and cosine waves used in the DFT are called as DFT basis functions. The output of the DFT is a set of numbers that represent amplitudes. The DFT basis functions are a set of sine and cosine waves with unity amplitude. In the frequency domain, if each of the amplitudes is assigned to the sine or cosine waves, the outcome will be a set of sine and cosine waves that can be added to form the time domain signal.

Figure 7 illustrates a typical block diagram of the intruder detection system. An NCO module is configured to produce a square wave of the desired frequency to be detected by the receiver. The output of the NCO is passed through a Band Pass Filter with a suitable frequency band to allow only the frequencies of interest around the center frequency, which is the frequency to be detected. The Band Pass Filter must have a high Q factor to get a better and sharper cutoff around the corner frequencies. Therefore, the output of the Band Pass Filter will be a sine wave at the fundamental frequency.

FIGURE 7: BLOCK DIAGRAM OF INTRUDER DETECTION SYSTEM



The ADC inside the PIC® MCU has two inputs: the sine wave output of the Band Pass Filter, and the signal sent from the TX and received at the RX. This signal must be correlated to determine if the frequency of interest (f_{detect}) exists. After the digital samples from the ADC are obtained, the sine samples are then passed through an orthogonal signal generation process to produce a cosine output. This orthogonal signal generation is computed in the firmware and some of the transforms or filters are described below:

- Hilbert Transforms: This transform is used to produce output signals which are 90° out of phase with respect to the input signal (i.e, orthogonal to each other). Therefore, if a sine wave is applied at the input, the result will be a cosine wave at the output with no attenuation.
- Low Pass Filter: If a low pass filter of first order is designed such that the user operates it in the stop band (i.e, beyond the cutoff frequency), and an input sine wave is provided to this filter, then the resulting output will be an attenuated signal of the same wave shape as the input, but shifted in phase by 90°. This signal can then be amplified in the firmware by multiplying with an appropriate gain to get the output amplitude to be same as the input amplitude. Therefore, the resulting waveform will be similar to the input waveform in shape (sine wave), frequency and amplitude, but shifted in phase by 90° (cosine wave).
- Shifting the Sampled Array: If the sine wave samples (elements in the array) are shifted appropriately such that the output samples are shifted by 90°, the resultant waveform will be a cosine wave.

Because the sine wave and cosine waveforms have been generated and are available in digital form, the correlation is performed on the input signal by performing the following computations:

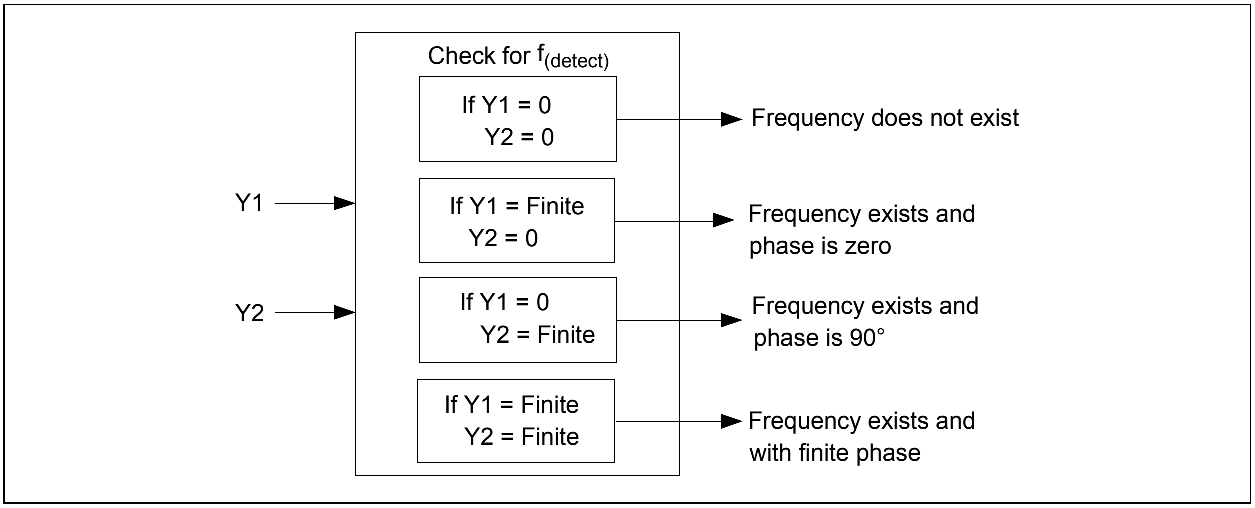
- Summation of the product of the individual sine wave samples and the input signal samples
- Summation of the product of the individual cosine wave samples and the input signal samples

Once the output (Y1 and Y2) of the two summations are available, check if the frequencies exist by interpreting the following results:

- If the frequency to be detected does not exist in the received signal, then the sum of the sine bins and cosine bins will be zero. Otherwise, there will be a finite value.
- If the sum of the sine bin is finite and the cosine bin is zero, then the signal at the detection frequency exists and the phase shift is zero.
- If the sum of the sine bin is zero and the cosine bin is finite, then the signal at the detection frequency exists and has a phase shift of 90°.
- If the sum in the sine bin and the cosine bin both have finite values, then the signal at the detection frequency exists and there is finite phase shift also.

The interpreted results are illustrated in Figure 8.

FIGURE 8: FREQUENCY DETECTION PRINCIPLE IN INTRUDER DETECTION SYSTEM



Sine Wave Generation Using NCO Module

As discussed in Case Study: Intruder Detection System Using Discrete Fourier Transform (DFT) by Correlation, a sine wave of desired frequency can be generated using the NCO module. Example 1 shows a code snippet for generating a square wave at 1 kHz. The square wave generated by this method is passed through a fourth order Sallen Key filter.

EXAMPLE 1: CODE SNIPPET FOR SINE WAVE GENERATION USING NCO MODULE

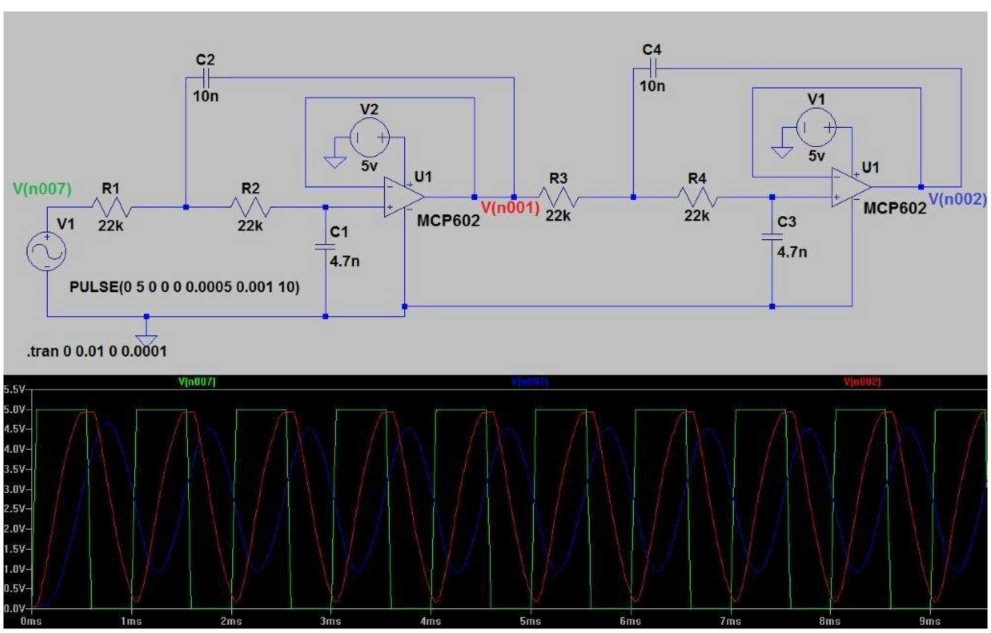
```
    #pragma config FOSC = INTOSC
// Fosc configuration
    OSCCON = 0x78;// Fosc = 16 MHz with internal oscillator
    __delay_us(100);

    TRISC = 0x00;          // Port C as digital output port
    ANSELC = 0x00;         // Port C as digital output port

    APFCON = 0x00;
    NCO1CON = 0xC0;        // NCO enable, NCO output enable, fixed frequency
    NCO1CLK = 0x01;        // NCO clock = FOSC = 16 MHz
    NCO1INCH = 0x00;       // NCO increment register high byte
    NCO1INCL = 0x84;       // NCO increment register high byte
// NCO output toggled at frequency = 2 kHz to get the square wave of 1 kHz
```

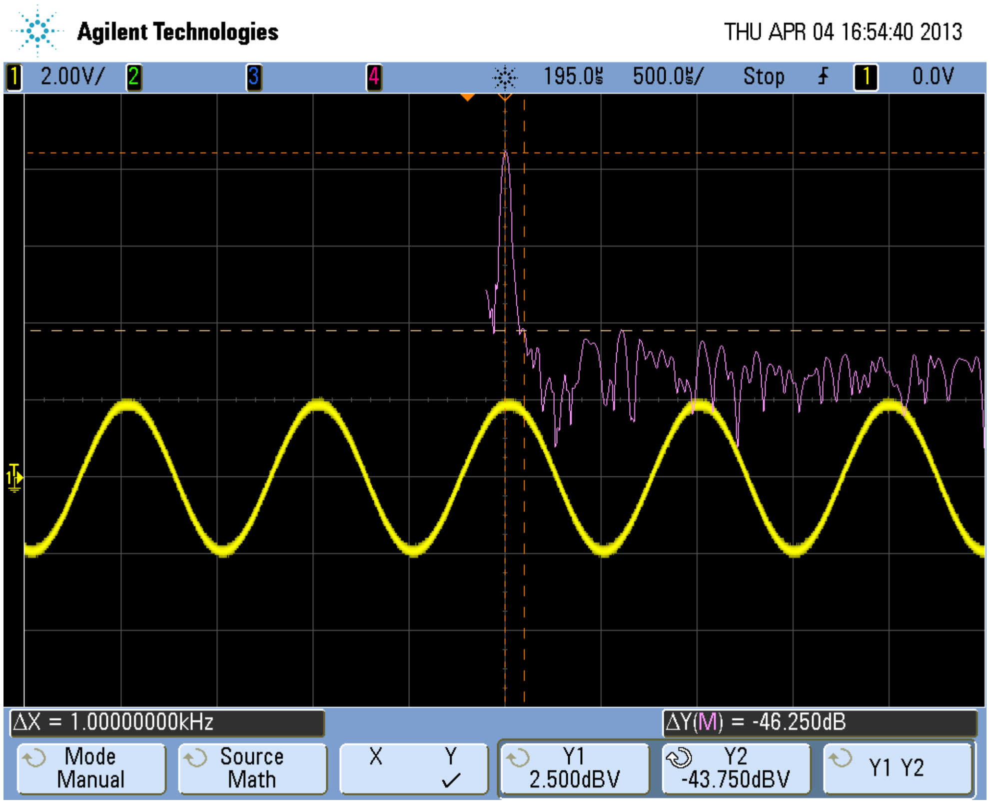
The simulation of the Sallen Key filter and the resulting waveforms are illustrated in Figure 9.

FIGURE 9: SIMULATION OF A SALLEN KEY FILTER



The output sine wave along with its resulting frequency spectrum is captured using an oscilloscope, see Figure 10. The fundamental frequency at 1 kHz is the most dominant while the other harmonic frequencies at 2 kHz, 3 kHz, 4 kHz, and so on are negligible or are very small.

FIGURE 10: FREQUENCY SPECTRUM OF A SINE WAVE GENERATED USING NCO MODULE



Sine Wave Generation Using the Look-Up Table Method

This section describes the look-up table method for sine wave generation. This is one of the most fundamental and popular methods of sine wave generation. In this method, the values used to approximate a sine wave are stored in memory.

There are three subsets in the look-up table method:

- The first method involves the synthesis of sine waves with frequencies which are multiples of the fundamental frequency for which the table elements are calculated.
- The second method involves the synthesis of sine waves with frequencies which are fractional multiples of the fundamental frequency for which the table elements are calculated. In this method, the frequencies are not integer multiples of the fundamental table frequency, and have substantially high Total Harmonic Distortion (THD).
- In the third method, the synthesis can be done for sine waves of non-integer multiples and also maintain a low THD by using interpolation.

When the look-up table-based implementations are done, the entire energy of the generated sine waves will not only be at the fundamental frequency, but a small amount of the energy will also be spread out at frequencies other than the fundamental frequency. These frequencies can be both harmonic and subharmonic frequencies. The presence of these frequencies will create a certain amount of distortion in the resultant waveform.

The harmonic distortions in the resulting waveform can be attributed to two factors: quantization and sampling errors. The sine table elements are stored in data memory and have definite word length such as 8 bits, 10 bits, 16 bits, and so on. Therefore, the values of these elements cannot be exactly represented and might result in quantization errors which are related to the word length. When dealing with frequencies which are non-integer multiples of the fundamental table frequency, the sample values between the two table entries must be estimated. These calculations would introduce sampling errors. Because these estimations inherently use the table values for calculation purpose, the resulting values will have quantization errors embedded in them, and the sampling errors will always be more than the quantization errors.

In order to reduce the quantization and sampling errors, a combination of the look-up table method along with interpolation must be used. This will reduce the distortions significantly. By using interpolation, the sine values between the values of table elements can be represented more precisely. For ease of implementation, Linear Interpolation method is used mostly. In this method, the values between any two table entries are assumed to lie on a straight line.

Example 1 shows a code snippet for generating a sine wave at 1 kHz using the look-up table method.

EXAMPLE 1: CODE SNIPPET FOR SINE WAVE GENERATION USING LOOK-UP TABLE METHOD

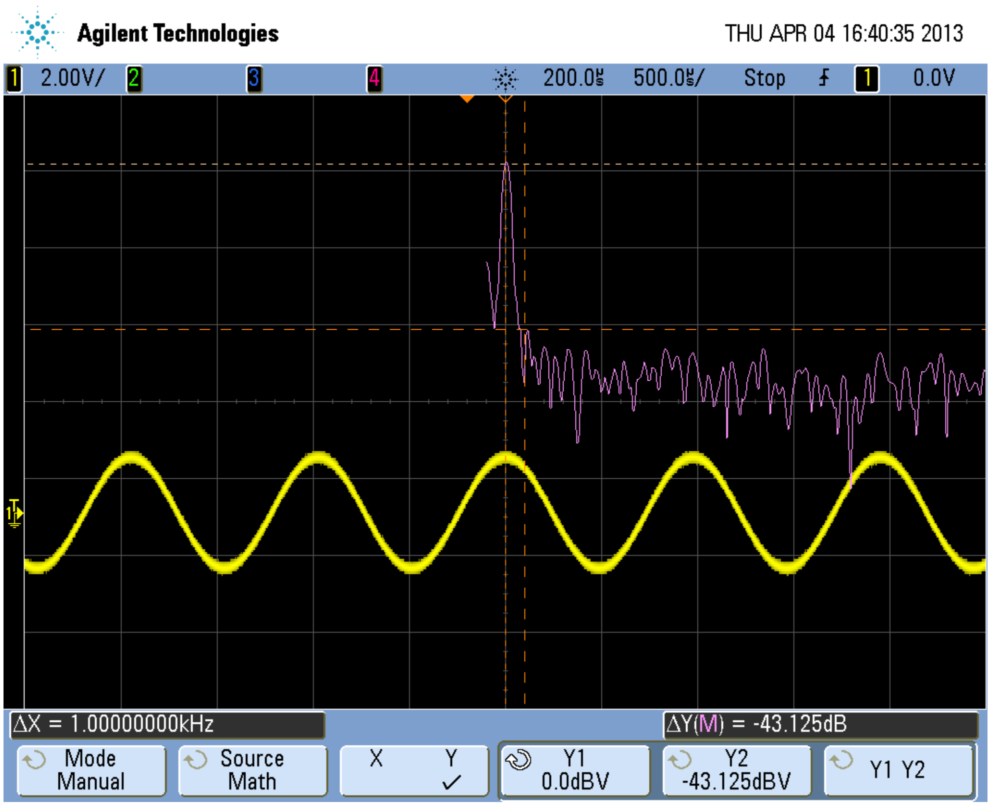
```
    #pragma config FOSC = INTOSC
    unsigned char gDutycount =0;
    const char SINETABLE[40]=
    {
    50,55,60,65,70,75,80,85,90,95,
    100,95,90,85,80,75,70,65,60,55,
    50,45,40,35,30,25,20,15,10,5,
    0,5,10,15,20,25,30,35,40,45
    };
// FOSC configuration
    OSCCON = 0x78;      // Fosc = 16 MHz with internal oscillator
    __delay_us(100);
// Timer2 configuration for PWM
   PR2 = 99;            // PWM period register for 40 kHz
   T2CON = 0x04;        // Timer2 on
// PWM 1 configuration
   PWM1CON = 0xC0;             // PWM1 on, PWM 1 output enable
   PWM1DCH = 50;               // PWM duty initialized to 50%
   PWM1DCL = 0;

   PIE1bits.TMR2IE =1;         // Timer2 interrupt enable
   INTCON =0xC0;               // Global interrupt enable, peripheral interrupt enable

   TRISC = 0x00;               // Port C as digital output port
   ANSELC = 0x00;              // Port C as digital output port
   void interrupt Timer2_ISR(void)
   {
      if (TMR2IF)
      {
          ++gDutycount;        // Increment the counter variable by 1
          if(gDutycount == 39)
          {
                 gDutycount = 0;
          }
         PWM1DCH = SINETABLE[gDutycount];     // Load the duty cycle register
         according to the sine table
         TMR2IF = 0;
      }
   }
```

The duty cycle of the PWM is varied and is passed through a Sallen Key filter. The resulting sine wave and the frequency spectrum is illustrated in Figure 11. The resulting harmonics are slightly more than the NCO-based methods, as provided in Sine Wave Generation Using NCO Module.

FIGURE 11: FREQUENCY SPECTRUM OF A SINE WAVE GENERATED USING LOOK-UP TABLE



Sine Wave Generation by Implementing Trigonometric Expressions

Sine wave and cosine wave are represented as shown in Equation 3.

EQUATION 3: SINE WAVE AND COSINE WAVE REPRESENTATION

$$V(t) = Sin(2\pi Ft)$$
$$V(t) = Cos(2\pi Ft)$$

Where,

$V(t)$ = Instantaneous value

t = Time instant

F = Signal frequency

2π = Used for converting to radians

The trigonometric value of the radian angle is the instantaneous value of signal. The only difference between sine and cosine wave is the phase difference of 90°. Therefore, the instantaneous values of one wave can be obtained by phase shifting that of another wave by 90°. These waves are generated in the analog domain through oscillators. The basic wave equation, when converted to digital signal, can be represented as shown in Equation 4.

EQUATION 4: DIGITAL REPRESENTATION OF BASIC WAVE EQUATION

$$X(n) = Sin(2\pi FnT)$$
$$X(n) = Cos(2\pi FnT)$$

Where,

n = Instantaneous digital sample time which is an integer (i.e, 0, 1, 2, 3, 4.......n)

T = Period/time between two samples of the wave

F = Signal frequency

$$X(n) = Sin(2\pi FnT) = Sin\left(\frac{2\pi Fn}{F_s}\right) = Sin(2\pi fn)$$

Also, $X(n) = Cos(2\pi fn)$

Where,

$f = F/Fs$ = Digital frequency from range -1/2 to 1/2

$Fs = 1/T$ or $T = 1/Fs$ (Fs = Sampling frequency)

In a digital computer, Equation 4 can be evaluated to generate samples of a specific frequency signal. The sine wave can be constructed using the Interpolation method or DAC. The digital frequency (f) can be calculated using F/Fs.

For example, if signal frequency = 1000 Hz, sampling frequency = 360000 Hz, then the value of f is shown in Equation 5.

EQUATION 5: DIGITAL FREQUENCY

$$f = \frac{F}{Fs} = \frac{1000}{360000}$$
$$= 0.00277777777777777777777777777778\ Hz$$

Substituting the value of f in Equation 4 and incrementing the value of n from 0 to ∞, the sample values for cosine wave can be calculated. However, every calculation involves the evaluation of cosine of radian angle, which is the evaluation of cosine infinite series. The problem with this approach is that it consumes a higher number of CPU cycles. To overcome this problem, trigonometric analysis can be used to reduce the evaluation of infinite series to few floating point calculations. The standard trigonometric expressions are provided in Equation 6 and Equation 7.

EQUATION 6: STANDARD TRIGONOMETRIC EXPRESSION

$$cos(x+y) = \cos x \cos y - \sin x \sin y$$

Where,

$sinx$ = Sine of signal frequency present sample

$siny$ = Sine of sampling frequency

$cos\ x$ = Cosine of signal frequency present sample

$cos\ y$ = Cosine of sampling frequency

$cos(x+y)$ = Cosine of signal frequency next sample

EQUATION 7: STANDARD TRIGONOMETRIC EXPRESSION

$$cos(x-y) = \cos x \cos y + \sin x \sin y$$

Where,

$cos(x-y)$ = Cosine of signal frequency previous sample

Equation 8 is derived by adding Equation 6 and Equation 7.

EQUATION 8: SUMMATION OF STANDARD TRIGONOMETRIC EXPRESSION

$$cos(x+y) + cos(x-y) = 2\cos x \cos y$$

Equation 8 is rearranged as shown in Equation 9.

EQUATION 9: REARRANGEMENT OF TRIGONOMETRIC EXPRESSION

$$cos(x+y) = 2\cos x \cos y - cos(x-y)$$

Consider Equation 10 for the angular representation of a sinusoidal waveform. Equation 10 can be used to calculate the next cosine value of cosine wave, if the previous and current samples of the signal and the cosine of sampling frequency is also known.

EQUATION 10: ANGULAR REPRESENTATION

$$x = 2\pi FnTs = \frac{2\pi nF}{Fs}$$

Where,

F = Signal frequency in Hz

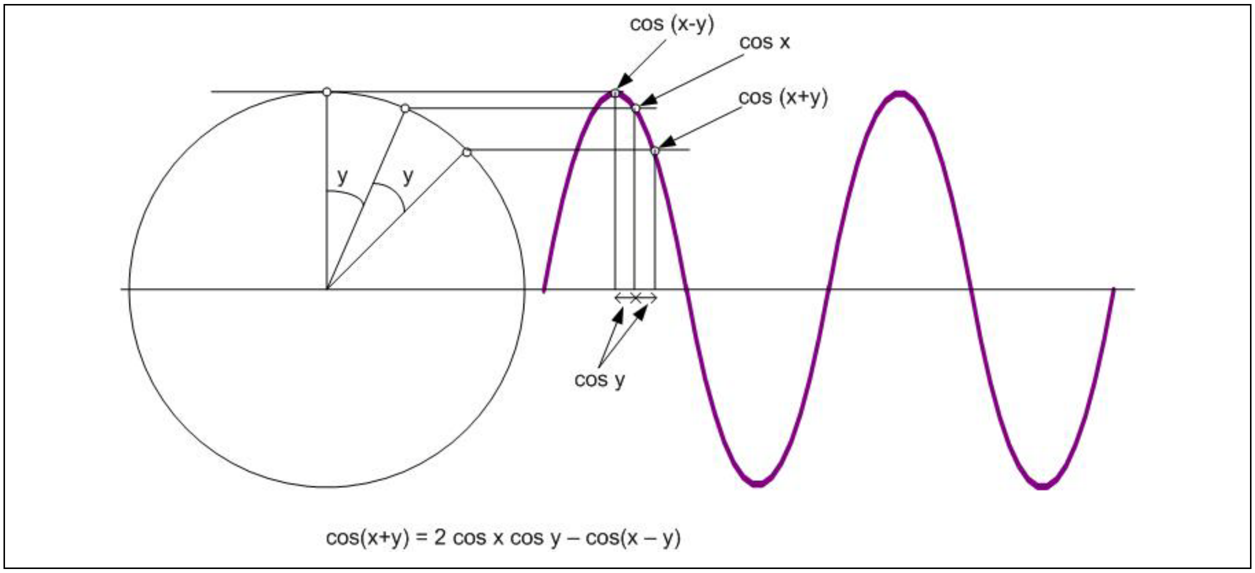
Fs = Sampling frequency in Hz

$f = F/Fs$ = Digital frequency from -0.5 to 0.5

y = Angular distance between the two points on the circle in radians (sampling rate in terms of angle in radians)

Figure 12 illustrates the trigonometric expressions provided in Equation 6 through Equation 10.

FIGURE 12: ANGULAR REPRESENTATION OF SINUSOIDAL WAVEFORM



Equation 10 can be rewritten as shown in Equation 11.

EQUATION 11: SAMPLE-BASED REPRESENTATION OF COSINE WAVE

$$y(n) = 2 \times \cos y \times y(n-1) - y(n-2)$$

Where,

$y(n)$ = nth sample being calculated

$y(n-1)$ = n-1th sample

$y(n-2)$ = n-2th sample

The first two samples of the cosine wave can be calculated by evaluating $cos(2\pi f0)$ and $cos(2\pi f1)$ directly.

Note 1: $cos2\pi f1$ is the cosine of the first sample which represents the angular distance between the 0th and first sample. Hence, it represents the angular distance covered during one sampling period on the signal.

2: Consider a circle of radius (r). Dividing the circle into Fs/F equal parts, the angle between two successive radius lines is the sampling rate. Hence, the cosine of that angle is the sampling frequency, which is in $2\pi fn$ radian, for n = 1. This is also the second sample of the signal where, n = 1 and $y(n-1) = cos\ y = cos\ x$.

The 8-bit MCUs do not contain floating point processor to perform the cosine infinite series evaluation, but the basic floating point operations such as addition, subtraction, multiplication and division can be emulated in software.

The recursive Equation 10 can be used to calculate the frequency signal samples for any sampling frequency. Also, the sine and cosine waves are periodic in nature and hence, only a single cycle needs to be calculated. The sample values repeat themselves after one cycle, hence no further calculations are needed. This will reduce the number of CPU cycles. A circular buffer can be used for easier implementation.

To convert the sample values into sine wave, at least 10 samples per cycle are required to reconstruct the signal shape. Hence, it is recommended to choose a sampling frequency that is ten times higher than the highest signal frequency. If the maximum signal frequency is 3000 Hz, the sampling rate should be 30000 Hz to keep the signal shape intact while reconstructing the sine wave.

The calculated sample values are in floating point format which need to be converted to integer format before sending them as input to the DAC, to create the analog signal. The resolution of the signal should be as high as possible (i.e, a signal with 10-bit sample values contains more fidelity in shape in reconstruction compared to a signal with 8-bit sample values).

Consider γ as the bit resolution for DAC. Multiplying the floating point sample values with $(2\gamma - 2(\gamma\text{-}1))$ and adding $2(\gamma\text{-}1)$ to all the sample values will convert from (-1, 1) range to (0, 2γ), and rounding of the result will make an integer array of signal samples for one cycle as shown in Equation 12.

EQUATION 12: INTEGER ARRAY OF SIGNAL SAMPLES FOR ONE CYCLE

$$\textit{sample\_value (integer value)} =$$
$$\textit{sample\_value(floating point value)} \times (2\gamma - 2(\gamma\text{-}1)) + 2(\gamma\text{-}1)$$

In 8-bit MCUs, the DAC with high resolution is rare, but there are Capture Compare (CCP) modules with PWM mode which can be used to convert the digital values to analog signal with a minimum resolution of 10 bits.

Initially, the PWM should be configured at 50% duty cycle. The PWM frequency should be the sampling frequency of the signal. For example, for a signal of 1000 Hz frequency sampled at 30000 Hz, the PWM frequency should be 30000 Hz at 50% duty cycle since every cycle or period of PWM signal should contain only one sample value of signal. If the two PWM periods contain the same sample value, filtering of the PWM will generate a stepped sine wave signal. The shape will hence be distorted, and the signal frequency will decrease by half. Therefore, the PWM frequency should be the sampling frequency with every period containing only one sample value of the signal. For this, the timer2 for PWM generation in PIC MCU can be used to generate the interrupt. In the interrupt handler, the CCP duty cycle value can be updated for every period. The output of PWM is then filtered to generate the sine wave signal using the double pole RC filter.

> **Note:** We can generate higher multiples of the calculated frequency (F) up to a maximum frequency of $Fs/10$. For example, consider a 1000 Hz computed signal sampled at a rate of 30000 Hz. To generate a 2000 Hz frequency from the 1000 Hz samples (harmonic or multiple of 1000 Hz), the samples should be selected such that they are alternate samples from the 1000 Hz signal at even or odd sample positions.
>
> If $x(n)$ = Samples of 1000 Hz signal at 30000 Hz rate, then $y(n)$ = Samples of 2000 Hz signal at 30000 Hz rate = $x(2n)$ or $x(2n\text{-}1)$.

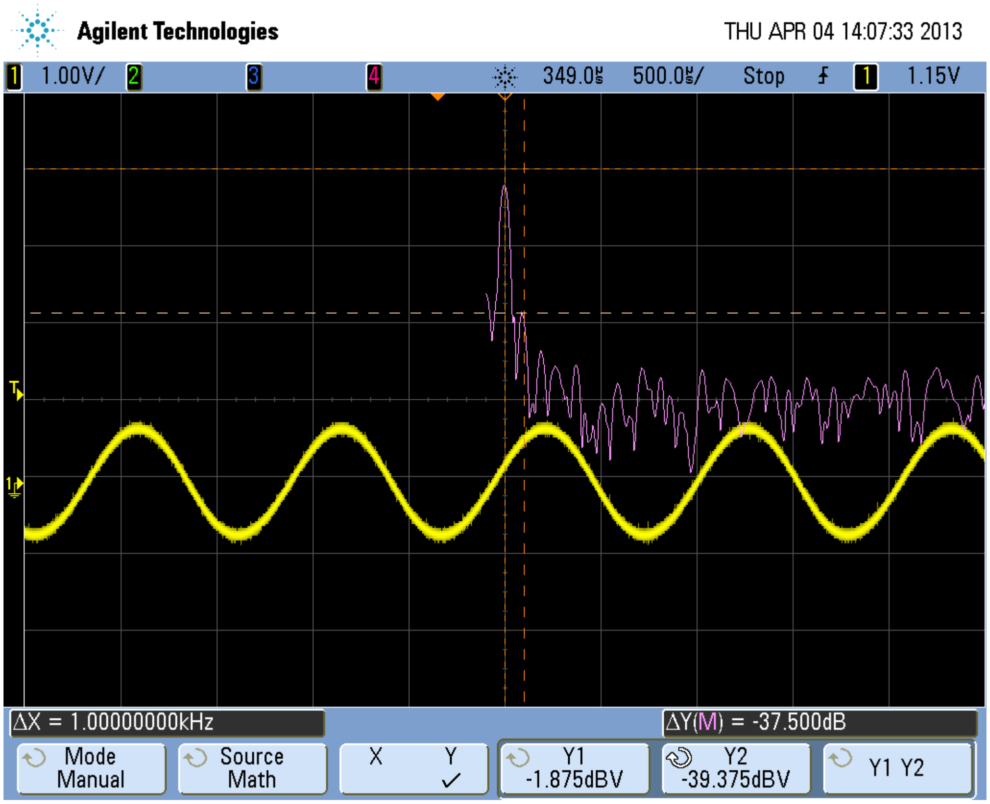
Example 2 shows a code snippet for generating a sine wave using the trigonometric method.

EXAMPLE 2: CODE SNIPPET FOR SINE WAVE GENERATION USING TRIGONOMETRIC COMPUTATION

```
Program to generate cosine/sine wave using trigonometric equation cos(a+b) = 2 cos a cos b - cos(a-b)
// where a is the current sample, b is the sampling frequency, a-b is the previous sample and a+b is the next sample
// to be calculated
    int single_cycle_array[40], samples; //array-integer array to store sample values
    of one sine/cosine cycle, samples-no of samples in one cycle=Fs/F
    void main()
    {
        float y, y_1, y_2=1.0, sampling_freq_angle;
// y_2 is previous sample, y-1 is current sample and y is the next sample
// sampling_freq_angle is the cosine of sampling frequency b
        int signal_freq=1000, i; //i-counter
        OSCCON = 0b11111100;
        OSCTUNE = 0b11000000;//31.25kHz, PLL enabled, factory calibrated frequency
        while(1)
        {
            samples = (int)(31250/signal_freq);
// The first and second sample value is hard coded, since the evaluation of infinite series in non-feasible on 8-bit MCU
        y_1 = sampling_freq_angle = 0.979855052384246865714793409500002;
// cos(2Π(signal_freq/31250)) and sampling frequency = PWM frequency = 31250 Hz
// This one time calculation will also save CPU cycles
        sampling_freq_angle *= 2.0;
// The sample values are rounded into range of 2^10, i.e, 0 to 1024 values
// Higher resolution PWM are advised to use for better and symmetric sine wave
// reconstruction
        single_cycle_array[0] = (int)(y_2*510+510);
        single_cycle_array[1] = (int)(y_1*510+510);
// The loop to finish calculation of remaining samples
        for(i=2;i<samples;i++)
        {
            y = y_1*sampling_freq_angle - y_2;
            single_cycle_array[i] = (int)(y*510+510);
            y_2 = y_1;
            y_1 = y;
        }
      }
    }
// Interrupt for PWM duty cycle update
```

The harmonics are higher compared to the sine wave generation using NCO or look-up table methods. The PWM output produced using this method is passed through a Sallen Key filter and the output is observed using an oscilloscope. This method yields a sine wave at 1 kHz with frequency spectrum, as illustrated in Figure 13.

FIGURE 13: FREQUENCY SPECTRUM OF A SINE WAVE GENERATED USING TRIGONOMETRIC COMPUTATION



CONCLUSION

Many applications can be designed using a sine wave generator. This application note has taken few applications and dealt with one case study of intruder detection in particular.

However, there are plenty of applications of a sine wave, because it forms the basic function for most of the electrical and electronic systems. Using the Numerically Controlled Oscillator (NCO) module to generate a sine wave at any desired frequency and its advantages over the conventional Pulse-Width Modulation (PWM) approach have also been covered. The use of the NCO is not limited to the generation of a sine wave. By using a proper filter with an appropriate cutoff frequency, any desired wave shape can be rendered to the resultant output.

For more information on other Core Independent Peripherals refer to www.microchip.com/CIP.

NOTES:

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

Printed on recycled paper.

ISBN: 9781620772140

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hangzhou
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7828
Fax: 886-7-330-9305

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

11/29/12