

Introduction

In this app note we will use pulse width modulation coupled with an external filter to create an approximation of a sine wave using a SLG46531V device. We will then interface with the GreenPAK using an Arduino Uno and have the GreenPAK play a short song. This functionality will be created by writing to several counters via I2C and triggering a note by toggling a GPIO.

GreenPAK Design

CNT4, CNT6, and DFF3 are the core of this design. CNT4 and CNT6 are both used to clock inverting flip flop DFF3, and CNT6 is exactly one clock cycle longer than CNT4. The result is that the output of DFF3 is a square wave with a varying duty cycle.

In Figure 2 and Figure 3, Yellow is the output of CNT4 and Blue is the output of CNT6, while Pink is the output of DFF3. You can see that the delay between CNT6 pulses is slightly longer than the delay between CNT4 pulses, and the Pink pulses become shorter due to this difference.

In Figure 3, you can see that the zoomed-out effect of the offset counters is a waveform with a period of $CNT4 * CNT6$ that repeats. The output of DFF3 (Pink) oscillates between a 100% duty cycle and a 0% duty cycle.

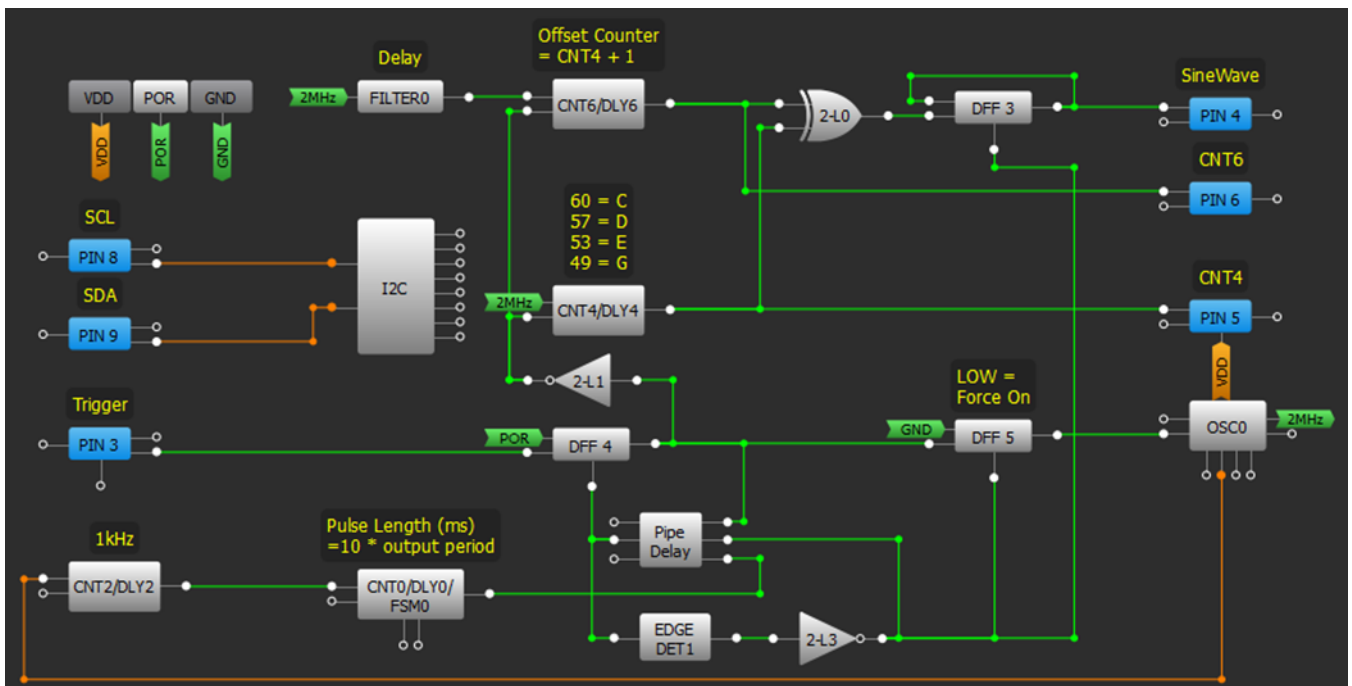


Figure 1. GreenPAK Design

The bottom section of the GreenPAK design file (see Figure 1) creates the Trigger functionality, which will tell the device to begin playing a tone. A rising edge of Pin 3 will latch DFF4 high, which is fed into the Pipe Delay with an inverting output. The Pipe Delay block delays the input for 10 clock periods then resets DFF4.

Meanwhile, the output of DFF5 is used as a latch to power down the oscillator when the triggered period is over. DFF5 is clocked by the output of DFF4, and is reset by the falling edge of the Pipe Delay's output. Figures 4, 5, and 6 include the property settings for the Pipe Delay, DFF4, and DFF5.

CNT2 is used to generate a 1kHz clock signal when the oscillator is powered on, which is used to clock CNT0. CNT0 in turn clocks the Pipe Delay. This means that in order to set the length of time you wish your tone to play, you will need to write the following to CNT0 via I2C:

$$CNT0 = 0.1 * [\text{length of time in ms}] - 1$$

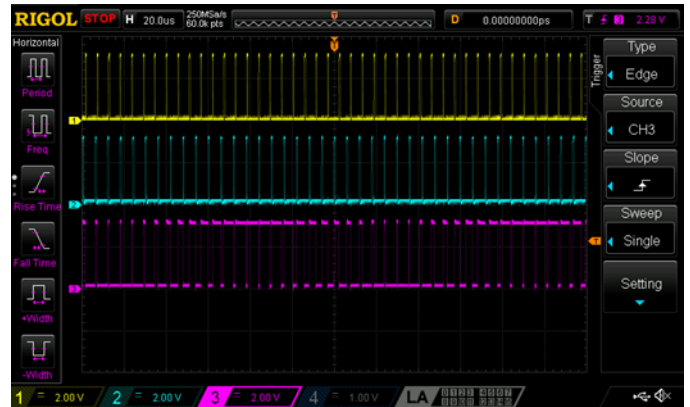


Figure 3. Zoomed-out view

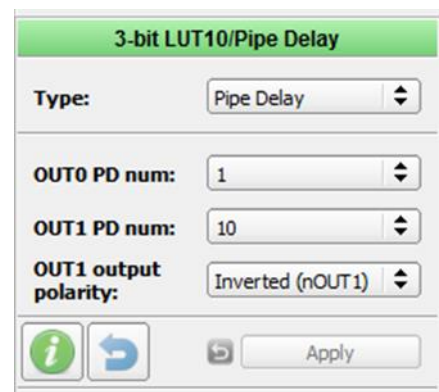


Figure 4. Pipe Delay Properties

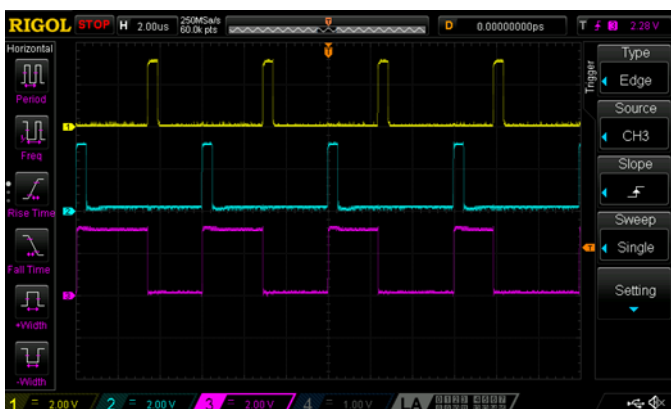


Figure 2. Close-up view

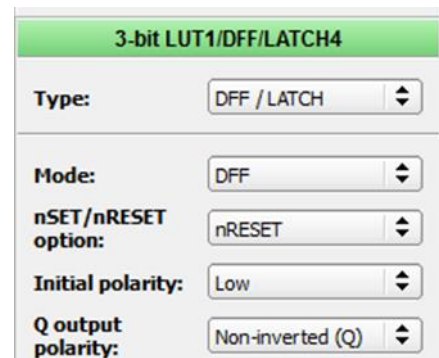


Figure 5. DFF4 Properties

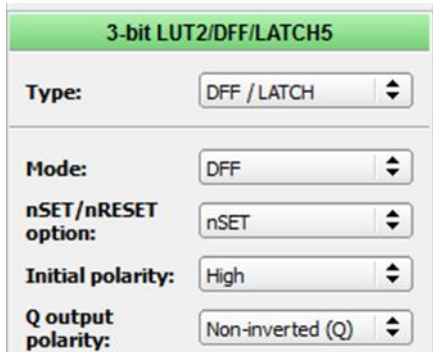


Figure 6. DFF5 Properties

External Circuit

Figure 7 shows the external circuit of the Sine Wave Generator. The trigger pin has an internal 1MΩ pull-down resistor. An old pair of headphones is used for the speaker, which is connected to Pin4 through an RC filter. The RC filter has the effect of smoothing out the square waves shown in Figure 3 and Figure 4, leaving the roughly sinusoidal wave shown in Figure 8. Once again Yellow is the output of CNT4, Blue is the output of CNT6, and Pink is the output of the RC Filter.

Arduino Sketch

This Arduino Uno sketch will play the nursery rhyme "Mary had a little lamb" through the GreenPAK by using the techniques described so far, and it prints out the song lyrics to the Arduino serial monitor in sync with the respective notes. This sketch makes use of the Silego Arduino Library described in **AN-1107**.

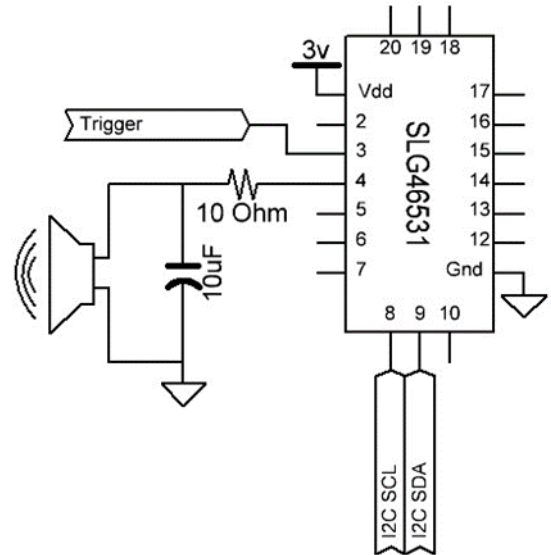


Figure 7. External circuit schematic

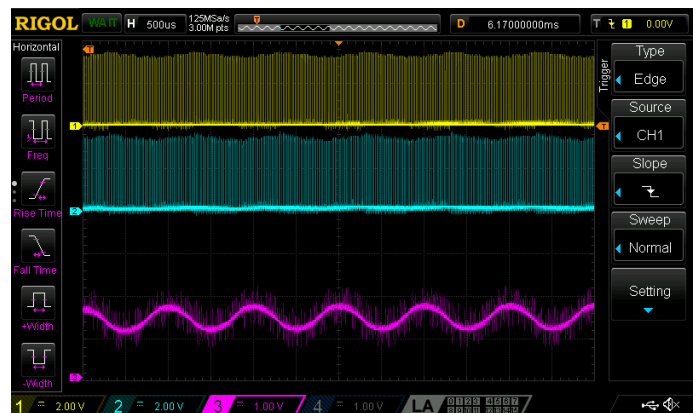


Figure 8. Sine Wave Approximation



```
#include "Silego.h"           // Include Silego header file
#include "macros/SLG46531.h"  // Include macros for SLG46531

#define Trigger 3
#define whole 100             // 100 = 0x0064
#define half 50               // 50 = 0x0032
#define quarter 25           // 25 = 0x0019
#define c 60                  // c = 261Hz
#define d 57                  // d = 294Hz
#define e 53                  // e = 330Hz
#define g 49                  // g = 392Hz

Silego silego(0x00);         // Instantiate silego library

void setup() {
  pinMode(Trigger, OUTPUT);

  Serial.begin(115200);
  Serial.print("GreenPAK: ");
  Serial.println(GreenPAK);
  Serial.println();
}

void loop() {
  delay(3000);

  Serial.print("Ma");        play(e, quarter);
  Serial.print("ry ");       play(d, quarter);
  Serial.print("had ");      play(c, quarter);
  Serial.print("a ");        play(d, quarter);
  Serial.print("lit");       play(e, quarter);
  Serial.print("tle ");      play(e, quarter);
  Serial.print("lamb, ");    play(e, half);

  Serial.print("lit");       play(d, quarter);
  Serial.print("tle ");      play(d, quarter);
  Serial.print("lamb, ");    play(d, half);

  Serial.print("lit");       play(e, quarter);
  Serial.print("tle ");      play(g, quarter);
  Serial.print("lamb! ");    play(g, half);

  Serial.print("Ma");        play(e, quarter);
  Serial.print("ry ");       play(d, quarter);
  Serial.print("had ");      play(c, quarter);
  Serial.print("a ");        play(d, quarter);
  Serial.print("lit");       play(e, quarter);
}
```



```
Serial.print("tle ");    play(e, quarter);
Serial.print("lamb ");   play(e, quarter);
Serial.print("whose ");  play(e, quarter);
Serial.print("fleece "); play(d, quarter);
Serial.print("was ");    play(d, quarter);
Serial.print("white ");  play(e, quarter);
Serial.print("as ");     play(d, quarter);
Serial.println("snow!"); play(c, whole);
}

void play(int pitch, int note) {
    switch (note) {          // Set up CNT0 for length of note
        case 100:
            silego.writeI2C(CNT0_1_DATA, 0x00);
            silego.writeI2C(CNT0_0_DATA, 0x64);
            break;
        case 50:
            silego.writeI2C(CNT0_1_DATA, 0x00);
            silego.writeI2C(CNT0_0_DATA, 0x32);
            break;
        case 25:
            silego.writeI2C(CNT0_1_DATA, 0x00);
            silego.writeI2C(CNT0_0_DATA, 0x19);
            break;
    }
    silego.writeI2C(CNT4_DATA, pitch); // Set up CNT4 for pitch
    silego.writeI2C(CNT6_DATA, pitch + 1); // Set up CNT6 for pitch offset

    digitalWrite(Trigger, HIGH); // Set Trigger high for 1ms
    delay(1);
    digitalWrite(Trigger, LOW);

    switch (note) {
        case quarter:    delay(350);    break; // Delay for note + 100ms
        case half:       delay(600);    break;
        case whole:      delay(1100);   break;
        default:         delay(350);    break;
    }
}
```



Conclusion

In this app note we used a Silego SLG46531V GreenPAK device to create a sine wave generator with just a few external components. We then interfaced with the GreenPAK using an Arduino Uno to play the nursery rhyme "Mary had a little lamb."

This technique could be useful for applications that require generating sine waves of varying frequencies.



About the Author

Name: David Riedell

Background: David attained a BS in Computer Engineering from North Carolina State University. He is currently working with CMICs as an applications engineer at Silego Technology.

Contact: [**appnotes@silego.com**](mailto:appnotes@silego.com)



Document History

Document Title: Sine Wave Generator

Document Number: AN-1111

Revision	Orig. of Change	Submission Date	Description of Change
A	David Riedell	06/14/2016	New application note

Worldwide Sales and Design Support

Silego Technology maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the sales person closest to you, visit us at **Sales Representatives and Distributors**.

About Silego Technology

Silego Technology, Inc. is a fabless semiconductor company headquartered in Santa Clara, California, with operations in Taiwan, and additional design/technology centers in China, Korea and Ukraine.



SILEGO
TECHNOLOGY

Silego Technology Inc.
1515 Wyatt Drive
Santa Clara, CA 95054

Phone: 408-327-8800
Fax: 408-988-3800
Website: www.silego.com