The University of Jordan
Department of EE - Microcontrollers Lab
MidTerm - Lab Report

Mohammad A. Mardeni - 0207316
Prof. Dr. Jamal Rahal

[Two Channel Custom Signal Generator with AM Modulation mode using ESP32]

Introduction

Custom signal generators are devices that can output any signal given a set of data or generating function. These generators are particularly useful in software-defined radio (SDR) applications, enabling the generation of various coding schemes for CDMA, orthogonal frequency division multiplexing (OFDM), and other encoding techniques such as PSK, ASK, and QAM. By using custom signal generators, SDR systems can flexibly adapt to different communication standards and protocols, making them highly versatile.

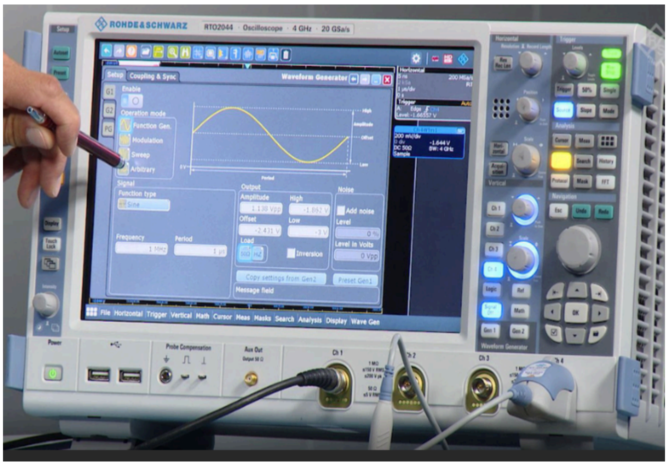


Figure 1. Arbitrary Waveform Generator

One significant advantage of custom signal generators is their ability to create security codes, which are essential for secure communications. These generators can produce complex, pseudo-random sequences that are difficult to predict, enhancing the security of transmitted data. This capability is crucial in military communications, secure wireless networks, and other applications where data integrity and confidentiality are paramount.

In addition to communication systems, custom signal generators find applications in various fields such as biomedical engineering, where they can be used to simulate physiological signals like ECG and EEG for testing and calibration of medical devices. In the automotive industry, they are used to generate test signals for evaluating the performance of sensors and electronic control units (ECUs).

Furthermore, custom signal generators are valuable tools in research and development. They enable scientists and engineers to experiment with new modulation techniques and communication protocols, accelerating innovation and the development of next-generation technologies. For instance, in the realm of Internet of Things (IoT), custom signal generators can help design and test low-power communication protocols tailored for various IoT devices and applications.

The University of Jordan
Department of EE - Microcontrollers Lab
MidTerm - Lab Report

Mohammad A. Mardeni - 0207316
Prof. Dr. Jamal Rahal

[Two Channel Custom Signal Generator with AM Modulation mode using ESP32]

Overall, the flexibility and adaptability of custom signal generators make them indispensable in both commercial and research settings. They not only facilitate the development and testing of advanced communication systems but also contribute to the security and reliability of various technological applications.

Accept Waveform Data from user and display it

Data can be entered using the serial interface by entering the waveform data and the metadata needed for proper operation of the AM modulation mode and raw data plotting mode. Metadata can be raw data sample period, raw data number of samples, carrier frequency of the modulation, modulation technique. The data of the two channels can also be included in the JSON format as channel0 and channel1 keys. The ESP 32 can then store the data in its EEPROM and show them at demand to the pins 25 and 26 using the internal DAC module included.

The proposed JSON format for my signal generator:

```
{ ‘Fc’:float_in_MegaHertz,
  ‘SamplePeriod’:float_in_MilliSeconds,
  ‘ch0’:[16 element list of floats normalized],
  ‘ch1’:[16 element list of floats normalized] }
```

To Receive, process and store the incoming JSON data from the serial i used the following code block:

1. check if the serial cache has data received in the loop part of the code.
   while(Serial.available()){ … }
2. At received JSON analyze the JSON and store relevant settings (i.e carrier frequency and sample period) then store wave data.

```
Text_set = Serial.readString();
JSONVar parsedText = JSON.parse(Text_set);

//Interpret JSON DATA into signal generator data
```

Mohammad A. Mardeni - 0207316
Prof. Dr. Jamal Rahal

[Two Channel Custom Signal Generator with AM Modulation mode using ESP32]

```
Freq = (float)(double)parsedText["Freq"];
sample_period = (float)(double)parsedText["sample_period"];
for (int i = 0 ; i < 16 ; i++){
  channel0[i] = (float)(double)parsedText["ch0"][i];
  channel1[i] = (float)(double)parsedText["ch1"][i];
}
```

3. Then print data to the PC serial monitor for validation

```
Serial.println("Freq ( kHz ): " + (String)Freq);
Serial.println("Sample Period ( millisec ): " + (String)sample_period);
Serial.print("Channel0: ");
for (int i=0 ; i< 16 ; i++){
      Serial.print(channel0[i]);
      Serial.print(" ");
 }
Serial.println();
Serial.print("Channel1: ");
for (int i=0 ; i< 16 ; i++){
      Serial.print(channel1[i]);
      Serial.print(" ");
}
 Serial.println();
```

4. Store the data in the EEPROM; using EEPROM.put() will allow for float storing

```
EEPROM.put(ADDR_FRE_EEPROM, Freq);
EEPROM.put(ADDR_SP_EEPROM, sample_period);
for (int i = 0; i < FLOAT_ARRAY_SIZE; i++) {
           EEPROM.put(ADDR_CH0_EEPROM + (i * sizeof(float)), channel0[i]);
}
 for (int i = 0; i < FLOAT_ARRAY_SIZE; i++) {
           EEPROM.put(ADDR_CH1_EEPROM + (i * sizeof(float)), channel1[i]);
}
 EEPROM.commit();
```

The University of Jordan
Department of EE - Microcontrollers Lab
MidTerm - Lab Report

Mohammad A. Mardeni - 0207316
Prof. Dr. Jamal Rahal

[Two Channel Custom Signal Generator with AM Modulation mode using ESP32]

---

5. Send Serial data back to Validate Stored data.

```
float freq;
float SP;
 EEPROM.get(ADDR_FRE_EEPROM,freq);
 EEPROM.get(ADDR_SP_EEPROM,SP);
 Serial.println("##################Validation of memory storage##################");
 Serial.println("Freq_Stored ( kHz ): "+ (String)freq);
 Serial.println("Sample Period Stored ( MilliSec. ): "+ (String)SP);
 Serial.print("Stored Channel0  Data: ");
 for (int i = 0; i < FLOAT_ARRAY_SIZE; i++) {
     float value;
     EEPROM.get(ADDR_CH0_EEPROM + (i * sizeof(float)), value);
     Serial.print(value);
     Serial.print(" ");
  }
 Serial.println();
 Serial.print("Stored Channel1  Data: ");
 for (int i = 0; i < FLOAT_ARRAY_SIZE; i++) {
     float value;
     EEPROM.get(ADDR_CH1_EEPROM + (i * sizeof(float)), value);
     Serial.print(value);
     Serial.print(" ");
 }
 Serial.println();
```

6. At pin 5 LOW the data would be displayed as it is by simply using the ESP32 DAC to change pin 25 and 26 (ch0 and ch1) voltage level every sample period time. The code below can perform this task easily.

```
for (int i=0; i < FLOAT_ARRAY_SIZE; i++){

       dacWrite(ch1_pin, 255/2 *channel1[i]);
       dacWrite(ch0_pin, 255/2 *channel0[i]);
       delayMicroseconds(uint32_t(sample_period*1000));

     }
```

The University of Jordan
Department of EE - Microcontrollers Lab
MidTerm - Lab Report

Mohammad A. Mardeni - 0207316

Prof. Dr. Jamal Rahal

[Two Channel Custom Signal Generator with AM Modulation mode using ESP32]

Using The AM modulation mode

AM modulation mode can be activated by pulling up the pin 5 to high voltage. The ESP would then display an AM modulated signal at both pins 25 and 26 (channel 0 and channel 1).

To modulate a signal we mean that we are changing its property with respect to another signal, or to represent the information of a signal in the characteristic of another signal like amplitude, phase, or frequency. AM modulation will modify the amplitude of a cosine function (using sinusoids makes smaller bandwidths and easier propagation in space). Multiplying the sinusoidal with a signal will modify its amplitude continuously as equation 1 shows below.

$\phi(t) = m(t) \times cos(\omega_C t + \theta)$; $\phi(t)$: **Modulated signal,** $m(t)$: **message,** $cos(\omega_C t + \theta)$: **Carrier … (Equ1)**

To modulate a signal with a sinusoid will mean a shift in the message frequency. To ensure no aliasing the carrier frequency should be at least twice the message frequency as shown in figure 2 below.

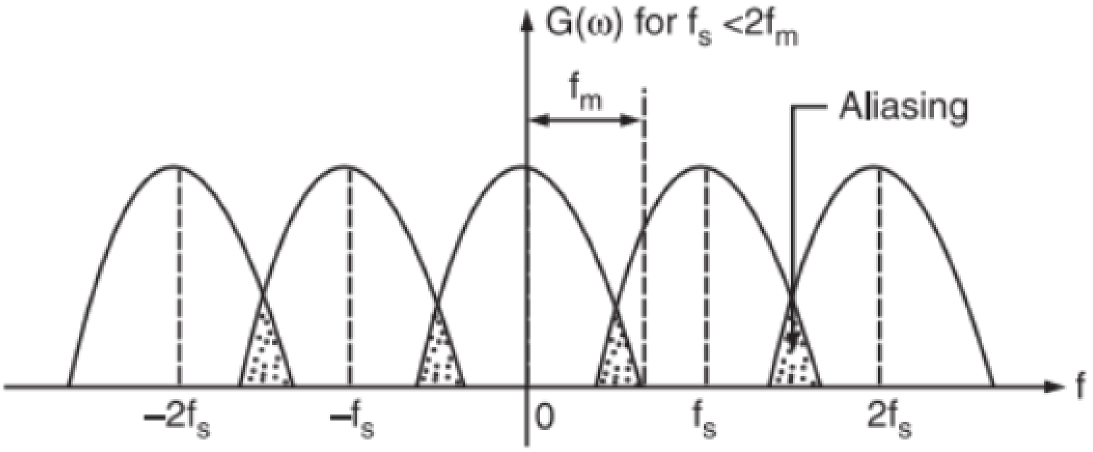


Figure 2. Effect of aliasing at low carrier frequency at fc<2fm

Mohammad A. Mardeni - 0207316
Prof. Dr. Jamal Rahal

Due to the above where the amplitude changes with respect to data values, the message would be in the envelope of the carrier which under modulation can simply be extracted by envelope detector (Rectifier and smoothing circuit). under modulation is where no phase reversal of the carrier occur. Figure 3 shows how the message can be in the carrier envelope.

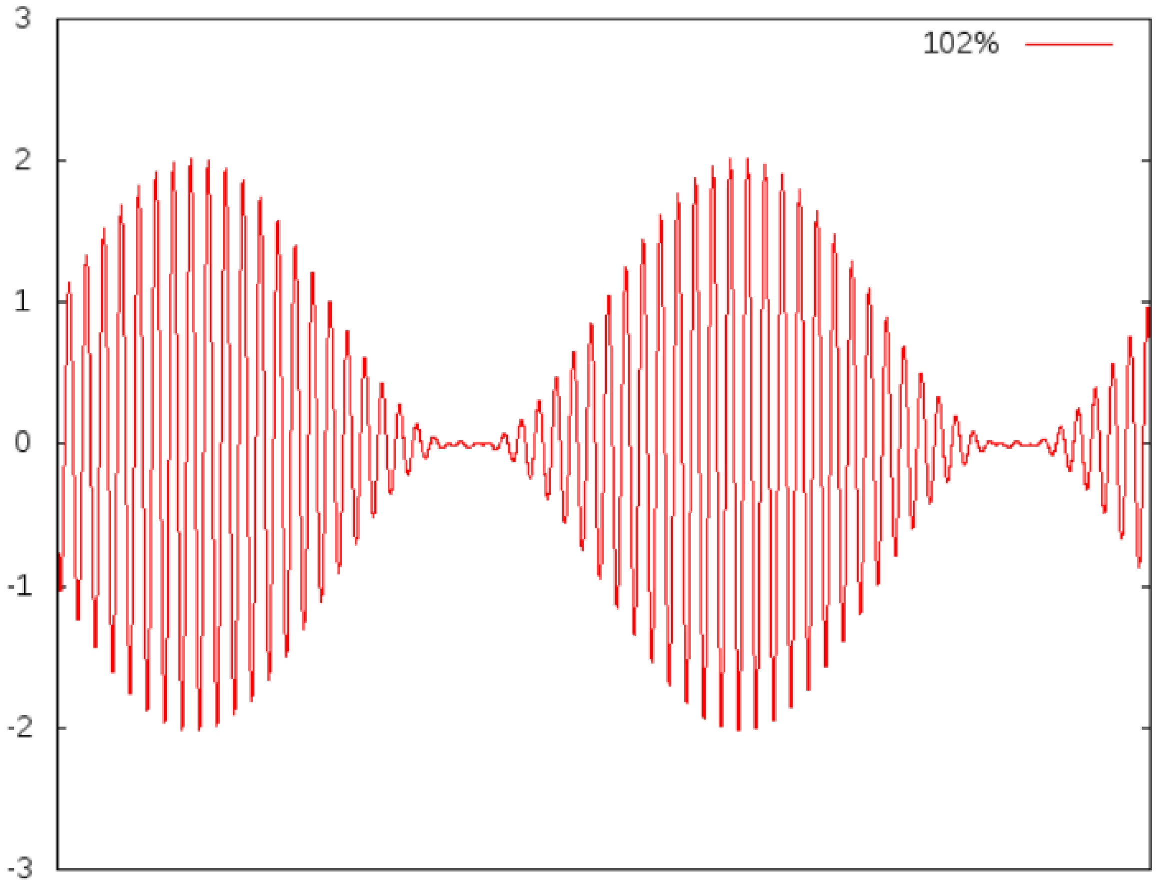


Figure 3. A tone signal modulated in a carrier. The envelope clearly shows the message.

In discrete time given by the discrete data provided by the JSON and the nature of the digitized clocked hardware to have DSB-LC we can follow the equation below:

$\phi(n) = (m(n) + 1) \times cos(2\pi \times F_C \times n \times CarrierSample_{period})$; **n: sample number … (Equ 2)**

The University of Jordan
Department of EE - Microcontrollers Lab
MidTerm - Lab Report

Mohammad A. Mardeni - 0207316
Prof. Dr. Jamal Rahal

[Two Channel Custom Signal Generator with AM Modulation mode using ESP32]

The following represents a code to generate AM - DSB - LC using ESP32 accounting for 0 - 3.3 volt output range of the ESP32. For each data sample the ESP32 would plot sample_period / carrier sample period where carrier sample period at least 1/(2*Carrier frequency).

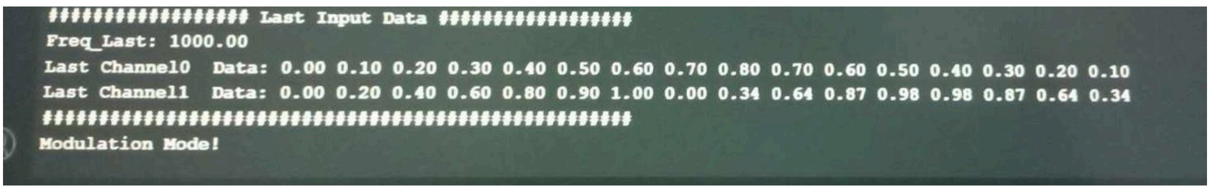
```
for (int i = 0; i < FLOAT_ARRAY_SIZE; i++) {

    for (int j = 0 ; j < 5 ; j++){
      double angle = 2.0 * M_PI * Freq * sample_period_Carr * j ;
      double signal0 =  (channel0[i]+1) * (cos(angle)) * 0.5;
      double signal1 =  (channel1[i]+1) * (cos(angle)) * 0.5;
      dacWrite(ch1_pin, 255/2 * ( signal1 + 1 ));
      dacWrite(ch0_pin, 255/2 * ( signal0 + 1 ));
      delayMicroseconds(sample_period_Carr * 1000);
    }
```

Results

1. First I entered the following test JSON using the serial monitor with no line ending.

{"Freq":100,
"sample_period":0.02,
"ch0":[0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.7,0.6,0.5,0.4,0.3,0.2,0.1],
"ch1":[0.0,0.2,0.4,0.5,0.6,0.8,0.9,1.0,0.0,0.17,0.32,0.433,0.4924,0.433,0.32,0.17]}

2. The ESP32 returned the following text to my serial monitor.



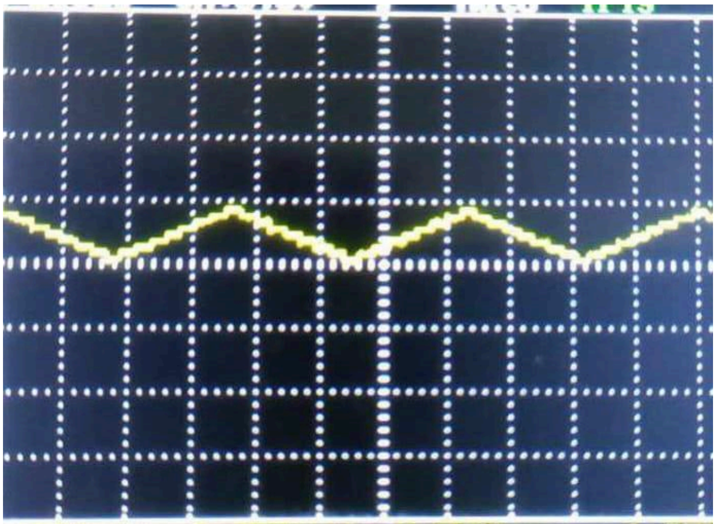
The University of Jordan
Department of EE - Microcontrollers Lab
MidTerm - Lab Report

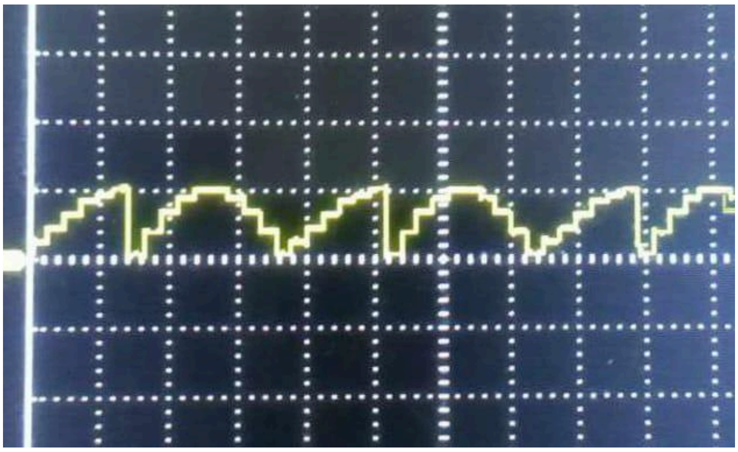
Mohammad A. Mardeni - 0207316
Prof. Dr. Jamal Rahal

[Two Channel Custom Signal Generator with AM Modulation mode using ESP32]

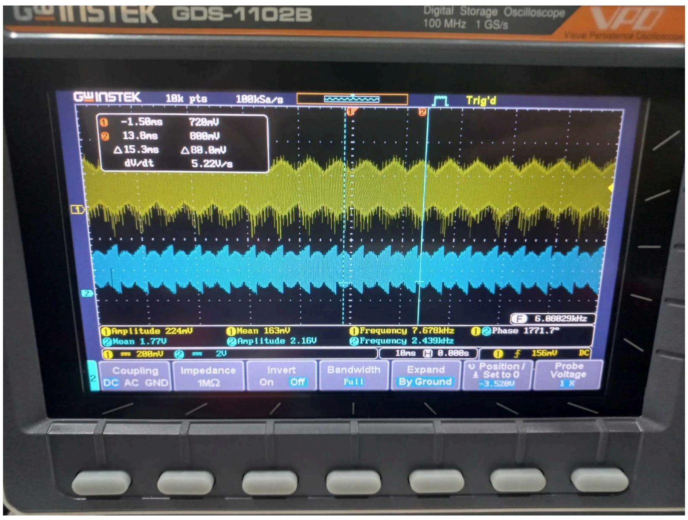
---

3. At pin 5 Low the raw data will be displayed as follows. Right picture is from pin 26 and left picture is from pin 25.





4. At pin 5 active, AM modulated signal will be outputted as in follows:



The University of Jordan
Department of EE - Microcontrollers Lab
MidTerm - Lab Report

Mohammad A. Mardeni - 0207316
Prof. Dr. Jamal Rahal

[Two Channel Custom Signal Generator with AM Modulation mode using ESP32]

Conclusion

Furthermore, the report has detailed the design and operation of a custom signal generator using an ESP 32 microcontroller, which is capable of accepting waveform data and metadata via a serial interface, storing this information in its EEPROM, and displaying the generated signals through its internal DAC module. The generator supports both raw data plotting mode and AM modulation mode, with the latter modifying the amplitude of a cosine function based on the input message.

The results section of the report demonstrated the successful operation of the custom signal generator, displaying both the raw data and AM modulated signals for a given test JSON input. The ability to generate and manipulate custom signals in this manner has significant implications for various fields, including biomedical engineering, automotive engineering, and research and development of advanced communication technologies.

In summary, custom signal generators are powerful tools for creating and testing various signal types, with numerous applications in commercial and research settings. The implementation of such a generator using an ESP 32 microcontroller, as detailed in this report, provides a flexible and adaptable solution for SDR systems and other applications requiring custom signal generation.