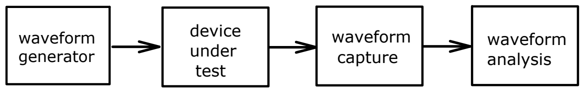
ELEX 3525 : Data Communications
2020 Fall Term

Generating Test Signals

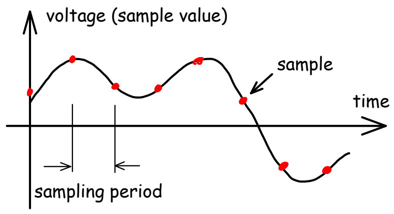
Introduction

One way to test a communication system is to apply a test waveform as the input and analyze the output:

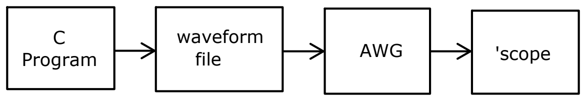


This approach is useful when you need to test a device or system for which there is no specialized test equipment. In this lab you will use an Arbitrary Waveform Generator (AWG) to generate a test waveform.

As the name implies, an AWG is able to generate an arbitrary waveform because it allows you to specify the output waveform as a sequence of sample values. These sample values are numbers whose value is proportional to the voltage of the waveform at equally-spaced intervals of time:



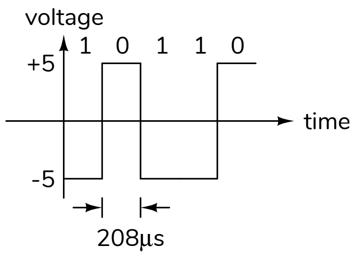
In this lab you will compute the sample values using Octave (Matlab) and write them to a file. Then you will transfer the sample values to the AWG using a Python utility. You will then view the waveform on the oscilloscope and check it using the 'scope's RS-232 decoding feature.



Analog Waveform Specification

The analog waveform output by the AWG should consist of a voltage that can be either +5 V or -5 V and is held steady for intervals of 208 μs. Each interval corresponds to one bit of data. For a '1' bit the voltage should have should have a voltage of -5 V and for a '0' it should have a voltage of +5 V.

Here's an example of the waveform corresponding to the sequence of bits 1, 0, 1, 1, 0:

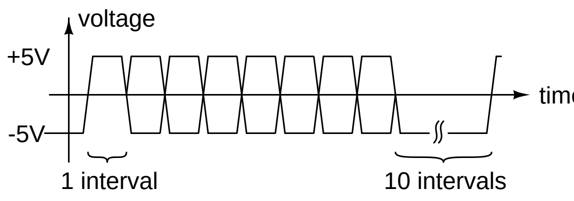


Your waveform must encode a 7-bit binary number that is the sum of all the digits of your student ID. For example, if your ID number is A00123456 the number you would output is 1+2+3+4+5+6 = 21 (hex 0x15, binary 0010101). This number may not match that of others doing the lab.

The bits of the number should be output in *least significant bit first* (lsb-first) bit order. For the binary value 0010101, the output will begin with 101···, *not* 001···.

Note that both the bit order and the signal polarity are probably the opposite of what you were expecting.

In addition, one positive voltage interval (equivalent to a '0' bit) should be added before the data bits and ten negative voltage intervals (equivalent to ten '1' bits) should be added after the data bits:



The AWG will continuously repeat your waveform and these additional "start" and "stop" bits will make it possible to identify the start and end of your waveform.

Generating the Waveform File

We will use Octave to create a CSV file with the required sample values. The commands required are listed below. By studying the comments you should be able to change the values to produce the required waveform. Lines starting with % are comments. Variables in Matlab are two-dimensional matrices of complex numbers. This includes column vectors (an $n \times 1$ matrix), row vectors (a $1 \times n$ matrix), and scalars (1×1). Most lines below are function calls that assign values to variables.

Note that the waveform used in the examples below are *not* the required waveform. You will need to change various parameters and your output will look different.

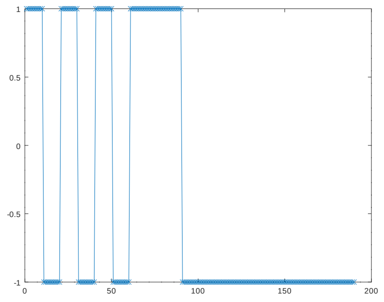
```
% load the communications package to access the
% de2bi() (decimal to binary) function

pkg load communications

% the value to be converted to a waveform

n=21

% convert each character to a row vector of 8 bits
% (lsb-first by default)

b=de2bi(n,8)

% add one start before and ten stop bits after

b=[0,b,ones(1,10)]

% replicate each row 10 times:

s=repmat(b,10,1)

% covert 0 to +1 and 1 to -1

v=1-2*s

% write the samples, in one column, to a CSV file

csvwrite('samples.csv',v(:))
```

You can enter these commands in the Octave GUI. The file browser lets you navigate to the folder where you will store your CSV file, the Workspace window shows your variables and the command window allows you to type in commands, such as the ones above, and view the results.

Procedure

Use Octave (or Matlab if you prefer) to create a CSV file with the required sample values. The sample levels should be ±1 (not ±5, see below). You should replicate each sample value 10 times to get an approximation to pulses.

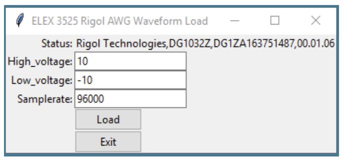
Use the **`plot(v(:),'x-')`** command to generate a plot of your waveform. An example is:



Check the waveform at this point to make sure it's correct. Use File > Save to save it to a format that you can include in your report (e.g. **`.png`**).

As in the previous lab, the instructor should already have connected the AWG channel 1 output to the 'scope channel 1 input so you can do this lab remotely. Run the "Ultra Scope" utility as in the previous lab to view the AWG output on the 'scope.

Run the **`awgload.py`** script:


ELEX 3525 Rigol AWG Waveform Load

Status: Rigol Technologies,DG1032Z,DG1ZA163751487,00.01.06

High_voltage: 10

Low_voltage: -10

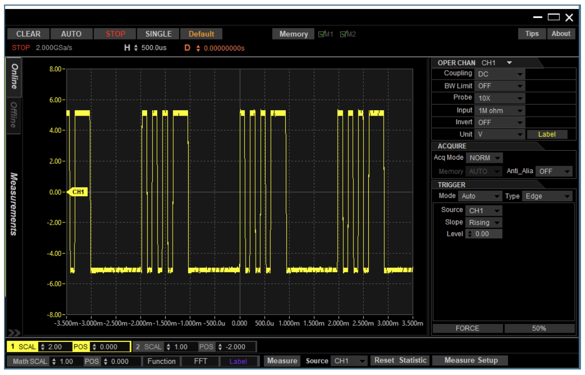
Samplerate: 96000

Load

Exit

and enter the required high and low voltage levels (±5 V). Values in the CSV file between -1 and +1 will be linearly scaled to the range of voltages between the low and high voltage levels respectively. The sample rate should be set so that each bit (not each sample) has a duration of 208 μs. Click on the Load button and choose the CSV file to load into the AWG. You should now see the waveform on the 'scope.

Adjust the 'scope vertical and horizontal scales so that you can see two or more periods of the waveform. Remember to change the probe scaling to 10X. Stop the 'scope. The display should look similar to:



STOP 2.000GSa/s H 500.0us D 0.00000000s

OPER CHAN CH1

Coupling DC

BW Limit OFF

Probe 10X

Input 1M ohm

Invert OFF

Unit V

Acq Mode NORM

Memory AUTO Anti_Alia OFF

Mode Auto Type Edge

Source CH1

Slope Rising

Level 0.00

1 SCAL 2.00 POS 0.000

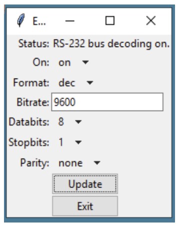
2 SCAL 1.00 POS -2.000

Math SCAL 1.00 POS 0.000

Source CH1

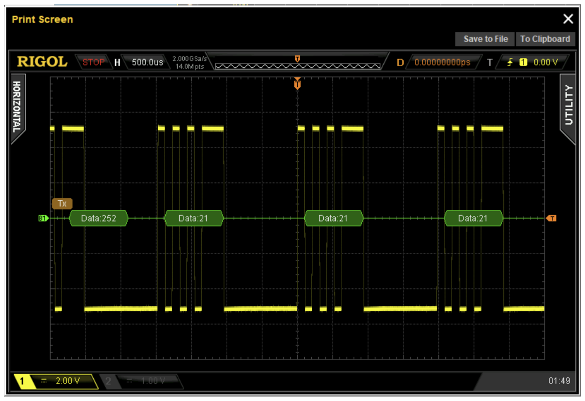
The 'scope can decode serial data waveforms in several formats but unfortunately the Ultra Scope software does not support configuration or display of this bus decoding.

Run the `scopebusdecode.py` script:


Status: RS-232 bus decoding on.
On: on
Format: dec
Bitrate: 9600
Databits: 8
Stopbits: 1
Parity: none

and configure the bus decoding for your waveform (7 data bits, 1 stop bit, no parity bits). The bit rate should be set according to the specifications above. Use a decimal base display for this lab. Press Update to configure the 'scope to decode the RS-232 signal.

Right-click on the screen and select Print Instrument Screen to capture what is shown on the 'scopes actual display. It will look similar to:



Verify that the decoded value matches the desired number. Take a screen capture of this display for your report.

Lab Report

Submit a report that includes:

- The cover page information as described in the course information handout.
- The calculation of the sum of the digits in your student number.
- A table showing the conversion of this number to bits and voltage levels. Show the bit number (0, 1, 2, ...), the bit value (0 or 1) and the signal voltage (+5 or -5).
- The computation of the sample period (time between samples) based on the bit period and the number of samples per bit specified above. Compute the corresponding sample rate that you will need to configure into the AWG.
- Screen captures showing the Octave waveform, and the 'scope's display with the decoded RS-232 value.

Convert your report to a PDF format file and submit it via the appropriate course web site Assignment folder.