Arbitrary Waveform Generator

Nicholas Antoniades

2020

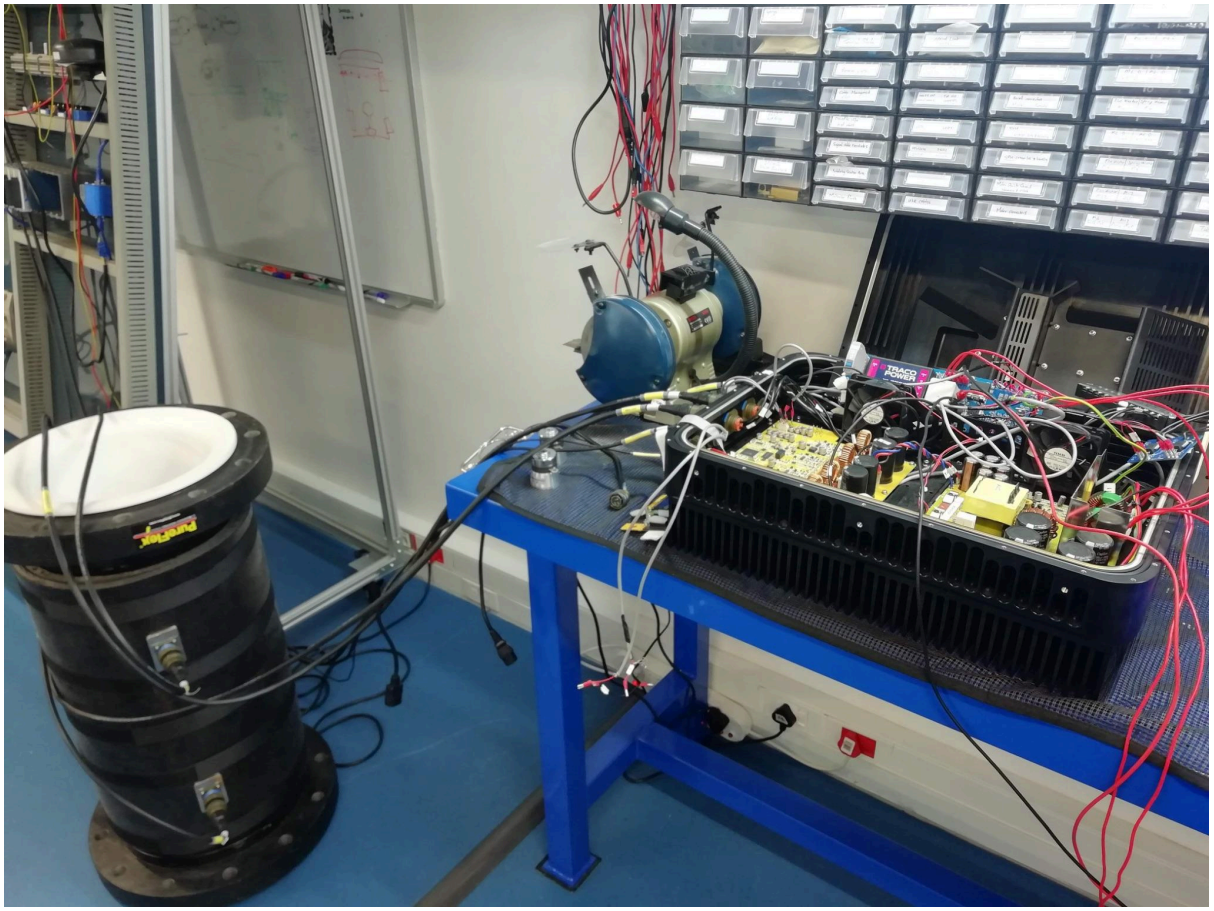
Table of Contents

Project Overview

This project focuses on developing an Arbitrary Waveform Generator (AWG) that allows the control of input signal frequency and amplitude through digital logic, interfacing with a CRIO (Compact Reconfigurable Input/Output) system. The system is designed to output a signal within the range of -2V to 2V peak-to-peak and supports a frequency range of 0-20kHz. The AWG is programmable via TTL or other logic interfaces.

The Arbitrary Waveform Generator was successfully deployed in an industrial sugar processing plant to combat scale formation in pipework. The system was integrated with a high-power amplifier to drive electromagnetic coils wrapped around critical pipe sections, implementing an innovative non-chemical scale prevention solution.


LIFT
HERE
STRAP
ONLY



Requirements

System Requirements:

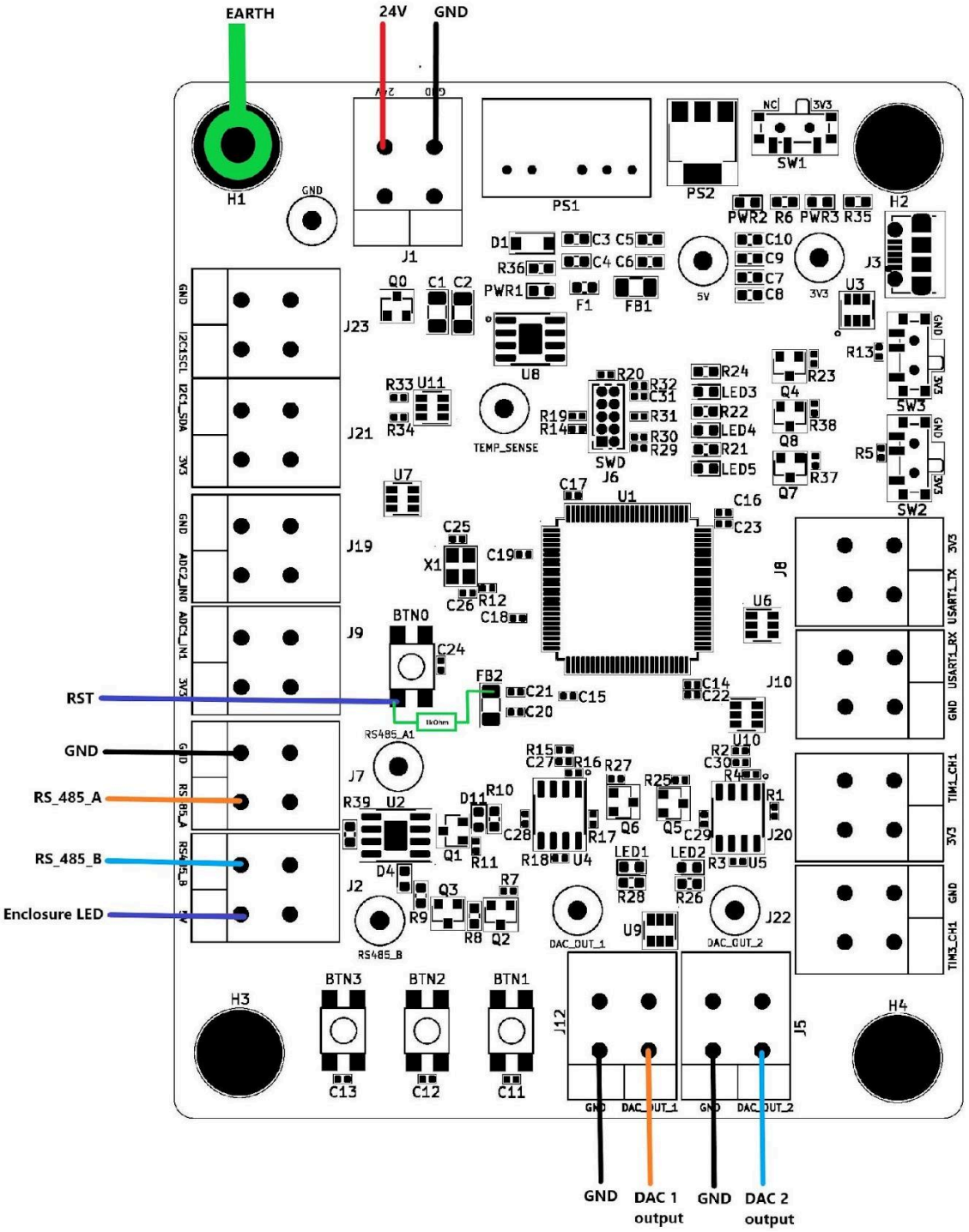
1. An output achieving at least a -2V to 2V peak to peak output signal.
2. 0-20kHz frequency range.
3. Programmable via TTL or other logic.

PCB Requirements:

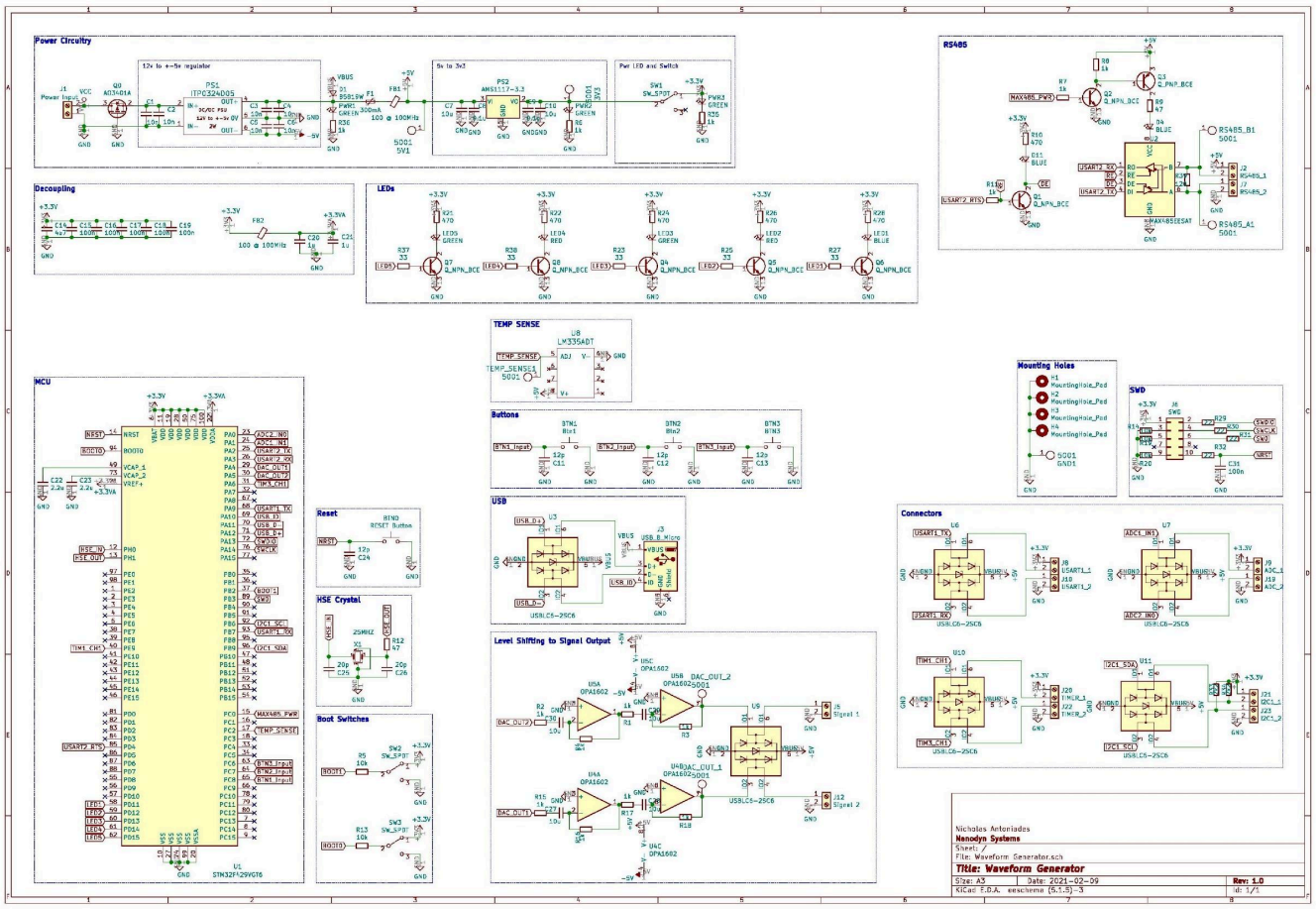
- USB DFU
- Power swich
- Level shifting circuitry
- DIP switch for boot mode settings
- DIP switch between power section and rest of circuit
- Power supply for +-5V
- Fuses
- Reset buttons
- LEDs
- ESD protection
- Reverse polarity protection
- SW with trace for debugging
- Temp sensor
- MAX485for RS485

Design

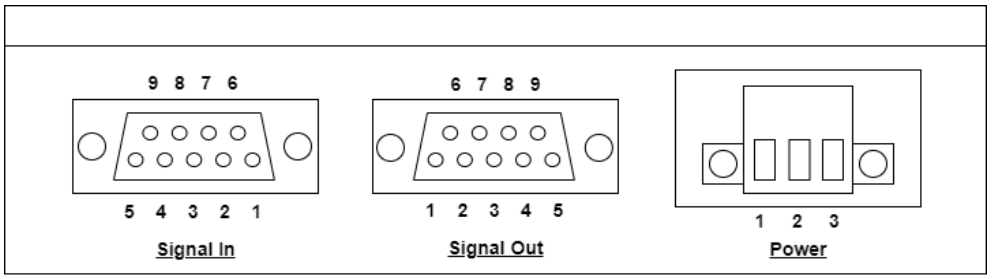
PCB overview



Schematics



AWG Enclosure wiring diagram



1 - RS485_A
2 - RS485_B
6 - GND
7 - RST
9 - GND

1 - Signal 1
2 - GND
3 - Signal 2
4 - GND

1 - 24v
2 - GND

PCB Configuration

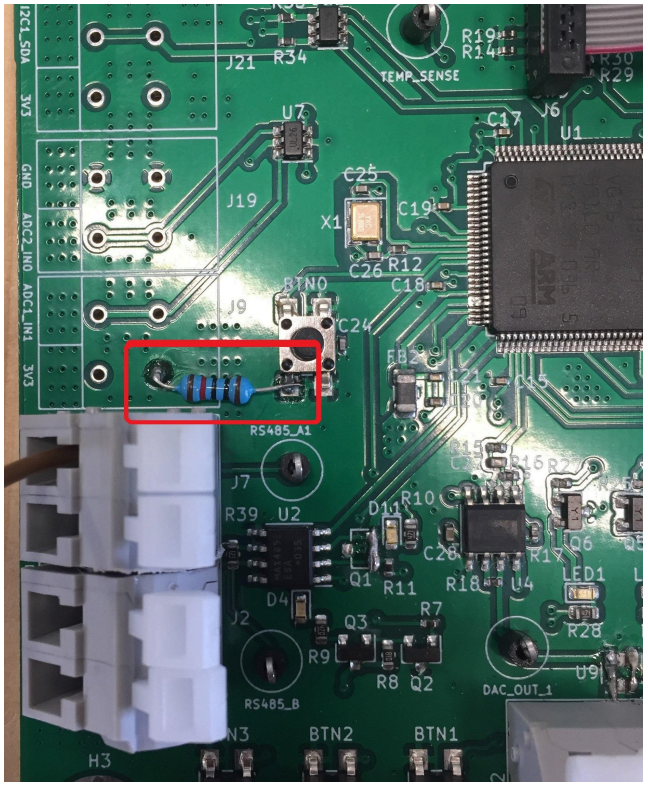
Steps for setting up the PCB:

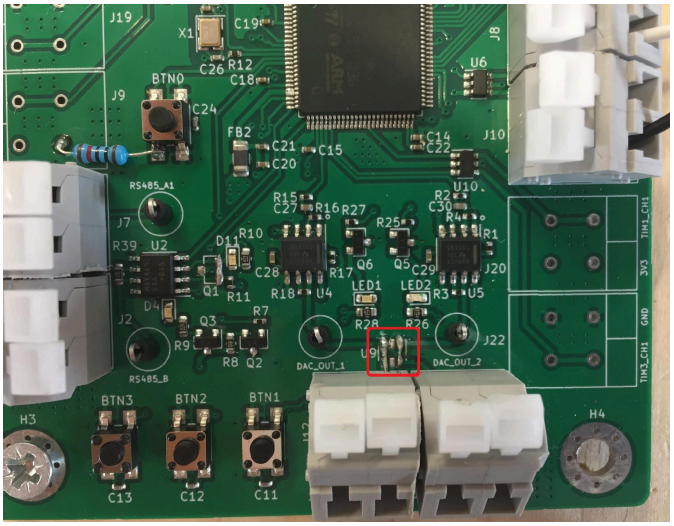
1. Add Fuse F1
2. Solder pull up resistor to NRST
3. Remove ESD protection U9. Bridge tracks as indicated
4. 10kOhm resistor was added as a pull-up resistor to bring the NRST pin to 3.3v
5. Removing LED D4 and bridging the pads as the voltage drop over the LED is such that the voltage over the MAX485 chip is too low.
6. Removing transistor Q1 and bridging its pads 1 and 2, then removing LED D11. This is so that DE and RE can be set at the same value. The original logic was so that they would be opposite values.

The 10kOhm pull-up resistor added can be seen in the figure below.

The ESD protection after the Op-Amp circuits were removed as the diode to the ground was clamping the circuit. Seen in the image below.

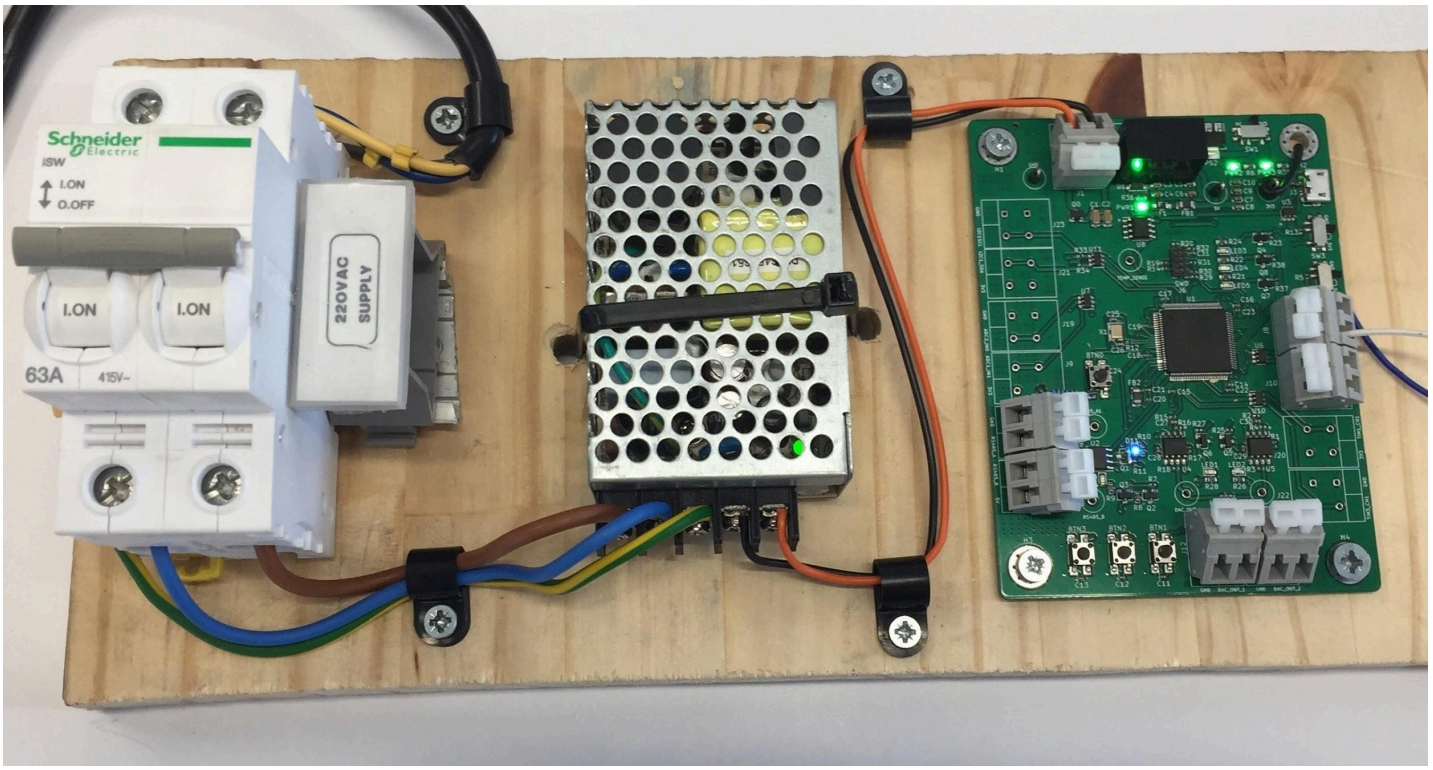
The inverting logic gate (The Npn diode) that was used to set the – pin and – pin which sets the MAX485 into transmit or receive mode was removed. This logic is not necessary as the pin is inverted already so both pins receive the same logic to switch between transmission and reception mode.





Power Supply Validation:

A 24V supply was connected to the 9V - 36V Input. +-5v and 3.3v were measured at the correct locations and STM MCU started up correctly and began to run the code.

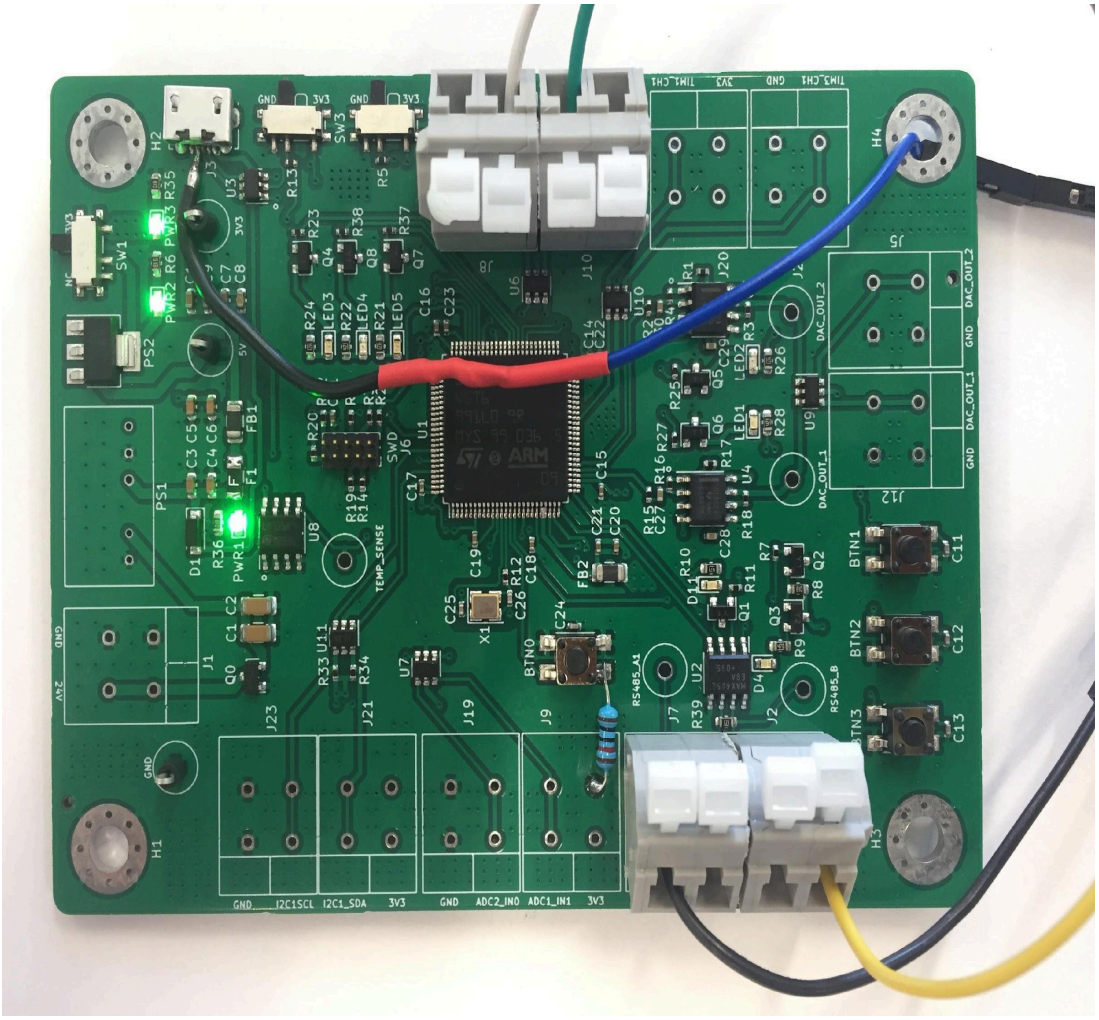


System Programming

UART Programming of the AWG

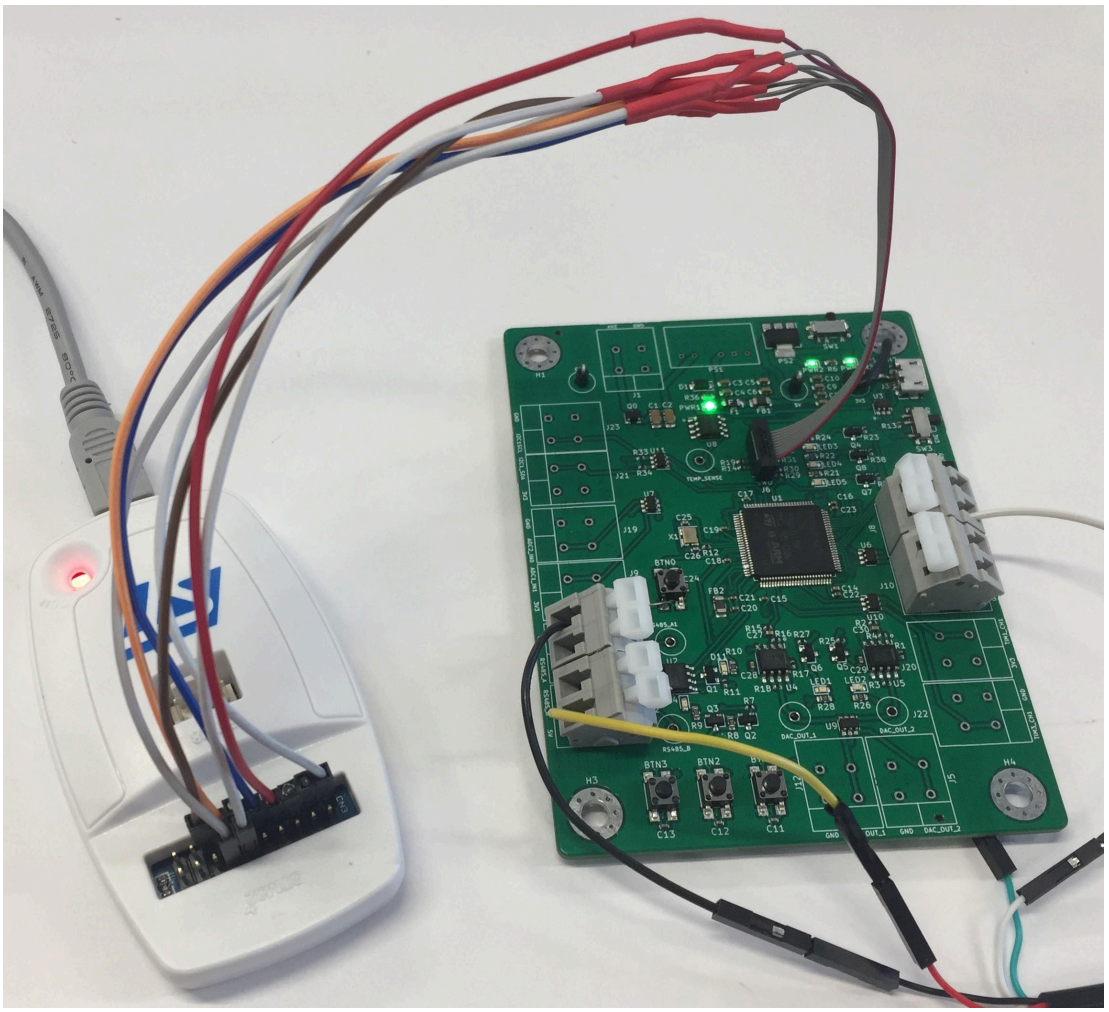
The AWG was first programmed via the USB to serial interface

1. An external wire was soldered to a USB ID pin to access PA10
2. Boot0 switch(sw3) was set to 3v3 and the device was reset to enter bootloading mode.
3. Start STM32CubeProgrammer and select UART programming
4. Select erasing & programming.
5. Select the .elf file to be uploaded, this can be found in the Debug folder of the build files from the STM32CubeIDE.



ST-Link V2 Programming of the AWG

1. Connect debugger and AWG PCB using the connector as seen in the figure below.
   - The footprint for the debugger on the AWG PCB is incorrect and inverted therefore the connector had to be manually wired to the correct orientation.
   - Add an extra wire from the ground of the debugger to the ground of the AWG PCB. Not seen in this image
2. Start STM32CubeProgrammer and select ST-Link programming
3. Select erasing & programming.
4. Select the .elf file to be uploaded, this can be found in the Debug folder of the build files from the STM32CubeIDE.



System Test

The max output frequency is dependent on Ns, the number of samples that form one full period of the sine wave. For lower output frequencies a higher Ns can be used to increase the resolution of the signal. The following table indicates the appropriate ns value for a desired output frequency range. These are based on the STM32F429VGT6 used on the AWG.

Based on the calculation:

PSC = - 1

There should be a function in the code that takes in the desired frequency and selects the appropriate Ns.

Other requirements for AWG:

1. The output signal must be scaled by o.5.
2. The output signal must be offset by 500.
3. DAC resolution = 4096
4. Fclock = 90 MHz
5. Period = 1.
6. PSC > 50

Setting up the RS485

1. Work on the correct response for uart
2. Set the MAX485 chip on

Waveform Generation Calculations:

PSC = - 1, Fsine = - 1, Fsine = = 13.636 kHz

Notes:

- When using op-amps it is important to consider the frequency limitations caused by the limited slew rate and bandwidth.
- Managing to get a working sine wave in the range of 0Hz and +- 50kHz.

Research

Waveform Generation Calculations:

PSC = - 1, Fsine = - 1, Fsine = = 13.636 kHz

Notes:

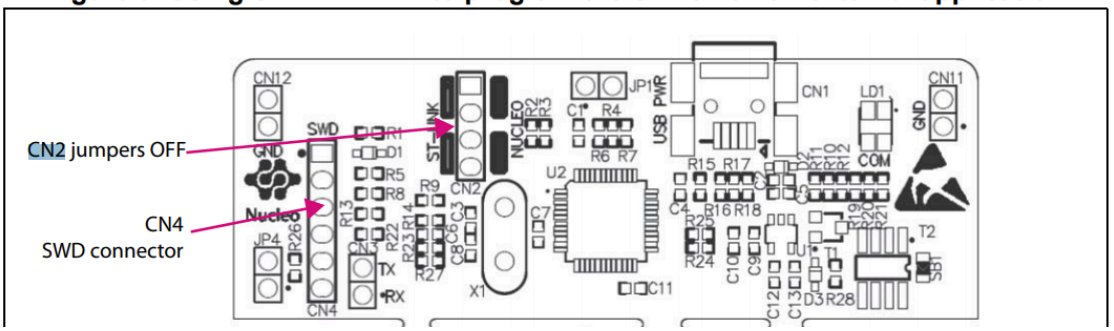
- When using op-amps it is important to consider the frequency limitations caused by the limited slew rate and bandwidth.
- Managing to get a working sine wave in the range of 0Hz and +- 50kHz.

STM32 Nucleo 144 SWD Pinout

Using this pinout proved difficult and I was not able to programme the development board this way.

Pin	CN4	Designation
1	VDD_TARGET	VDD from application
2	SWCLK	SWD clock
3	GND	ground
4	SWDIO	SWD data input/output
5	NRST	RESET of target STM32
6	SWO	Reserved

Figure 9. Using ST-LINK/V2-1 to program the STM32 on an external application



List of commands from CRIO to STM:

1. On/off command.
2. Change - the frequency of DAC channel 1.
3. Change - the frequency of DAC channel 2.
4. Change - the amplitude of DAC channel 1.
5. Change - the amplitude of DAC channel 2.
6. Request - Voltage and Current measurement of channel 1 output.
7. Request - Voltage and Current measurement of channel 2 output.
8. Request - Temperature sensor 1 and 2 output.
9. Acknowledge message received.
10. Bad message received.
11. Request current system state.

List of messages from STM to CRIO:

1. Return - the frequency of DAC channel 1.
2. Return - the frequency of DAC channel 2.
3. Return - the amplitude of DAC channel 1.
4. Return - the amplitude of DAC channel 2.
5. Return - Voltage and Current measurement of channel 1 output.
6. Return - Voltage and Current measurement of channel 2 output.
7. Return - Temperature sensors 1 and 2 output.
8. Acknowledge message received.
9. Bad message received.
10. Low power mode.
1.

Software changes

Problem

The max485 chip is physically connected to UART2 Tx and Rx pins. UART2 uses the same DMA channels as the DAC 1 and 2 outputs. So using the DMA method for receiving data on the UART2 RX line won't work.

Solution : - Using the UART interrupt (IT) method to handle the received messages. - Using the UART polling method for sending messages