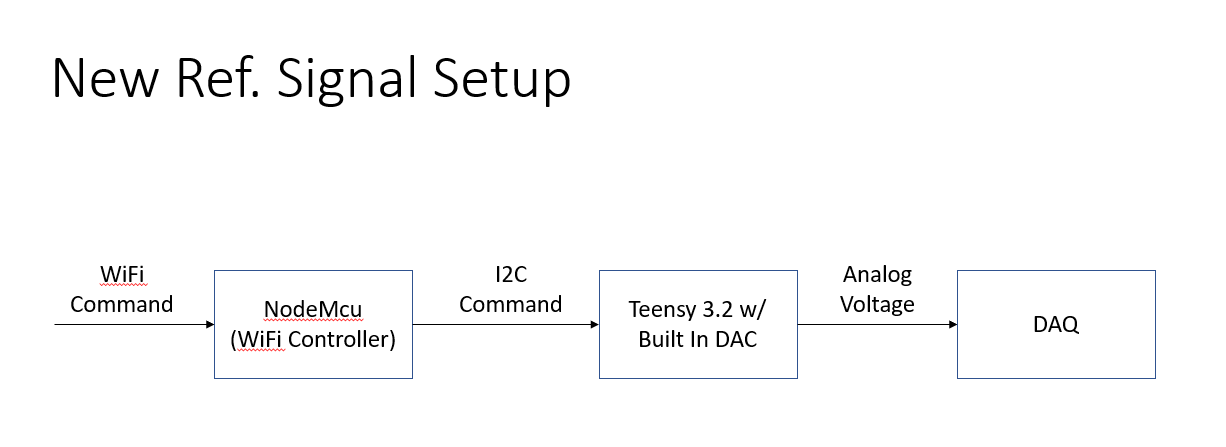
Reference Signal Setup Documentation

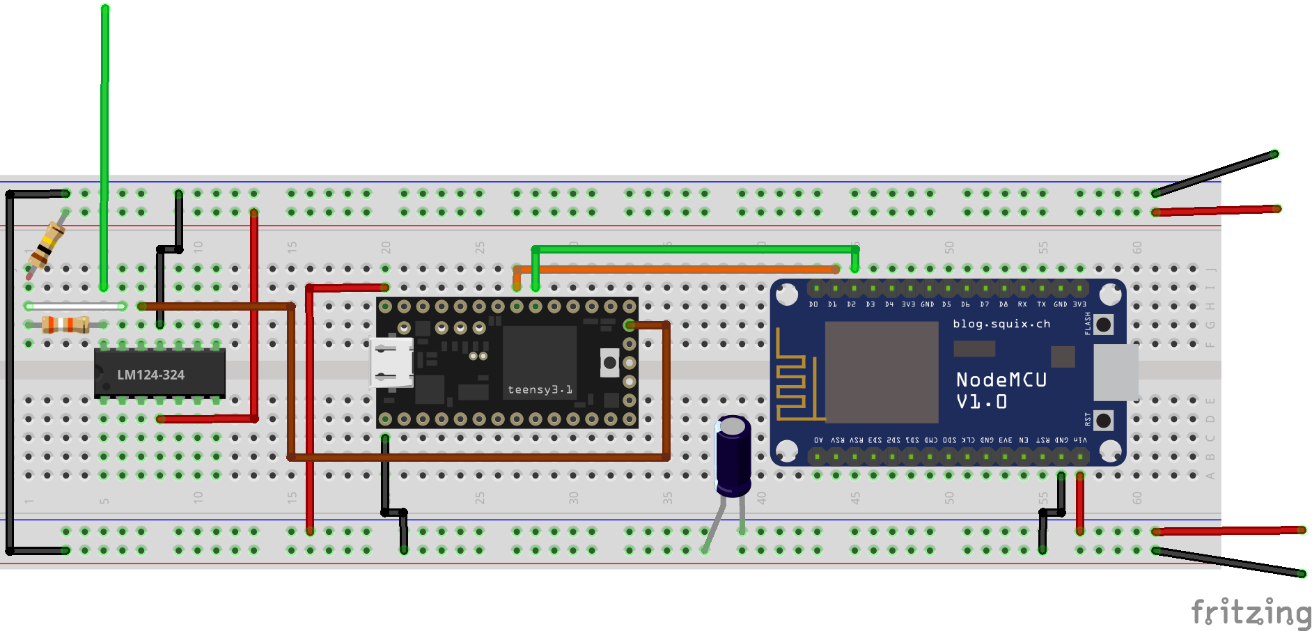
Tanner Bitz

1/31/2019

*System Diagram*



*Breadboard Setup*



+5V Power Source

+15V Power Source

Ref Sig  
 (Connect to DAQ)

*List of Materials*

* 1x NodeMcu
* 1x Teensy 3.2
* 1x LM324 Op Amp
* 1x 470uF capacitor
* 1x 39k Ohm resistor
* 1x 100k Ohm resistor

*Overview:*

The NodeMcu receives commands as http GET requests. It creates a webserver with the static IP 192.168.0.107 on the network:

Network: BetterLateThanNever

Password: PleaseWork

You must be connected to this network to create a http GET request.

If you’re using Python, as I do, you can use the requests library to programmatically make a http GET request to create the wifi signal.

The commands have the following structure:

“http://192.168.0.107/RelayToI2C?Command=[CommandNumber](-hyphenSeperatedArgs)”

[CommandNumber] is mandatory

(-hyphenSeparatedArgs) are optional

Ex:

If I wanted to change the frequency of the reference signal I would use the command

http://192.168.0.107/RelayToI2C?Command=6-0.2

This is because the command to change frequency is 6 and I want to change it freq= 0.2Hz.

The “Commands” section lists a full set of commands.

Once the NodeMcu receives the command it relays the string after “Command=”, so in our example it would relay “6-0.2” over I2C to the Teensy 3.2. The Teensy 3.2 then parses the command string it received and executes the corresponding command.

Note: I didn’t write a lot of error checking code, however I tested each function and it works as intended. If it’s not working, you’re probably not sending the command as expected. This may cause the NodeMcu or Teensy to crash (though I haven’t had this experience). If this is the case, cycle power to the broad.

If the Teensy 3.2 is given a command to output a reference signal, it will do so on pin A14, which is fed into the LM324 op amp to be amplified to ~0-5V. This amplified signal is then connected to the DAQ.

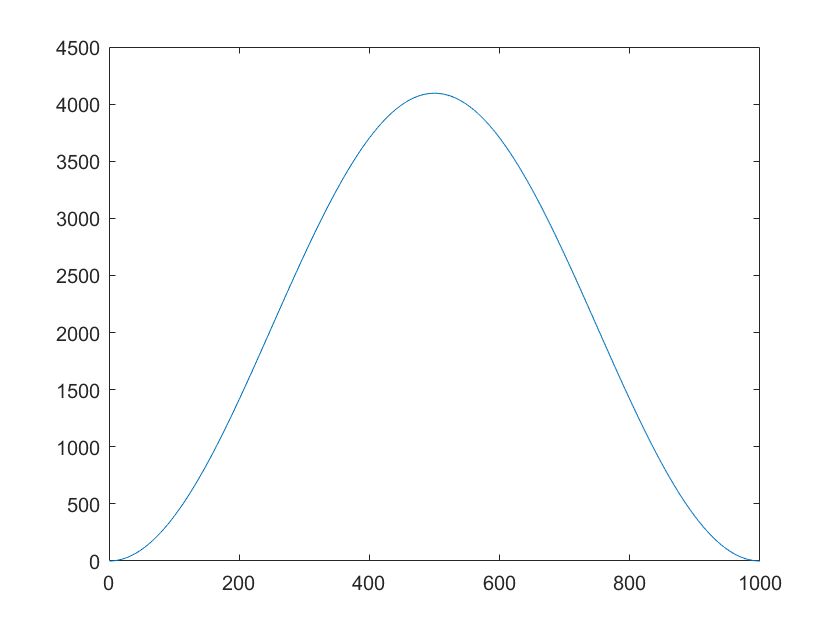
There are built it commands to change the floor and ceiling voltages of the reference signal. The idea was to configure your voltage range so the reference signal would not saturate in the DAQ.

*Commands:*

1. Unidirection Flex

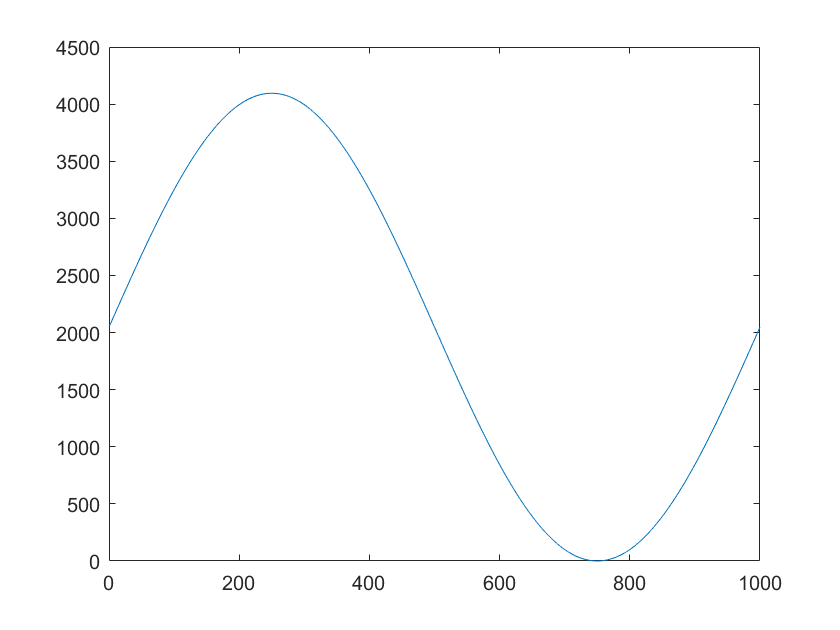
Command: “http://192.168.0.107/RelayToI2C?Command=1”

This command outputs a single sinusoid of the form:



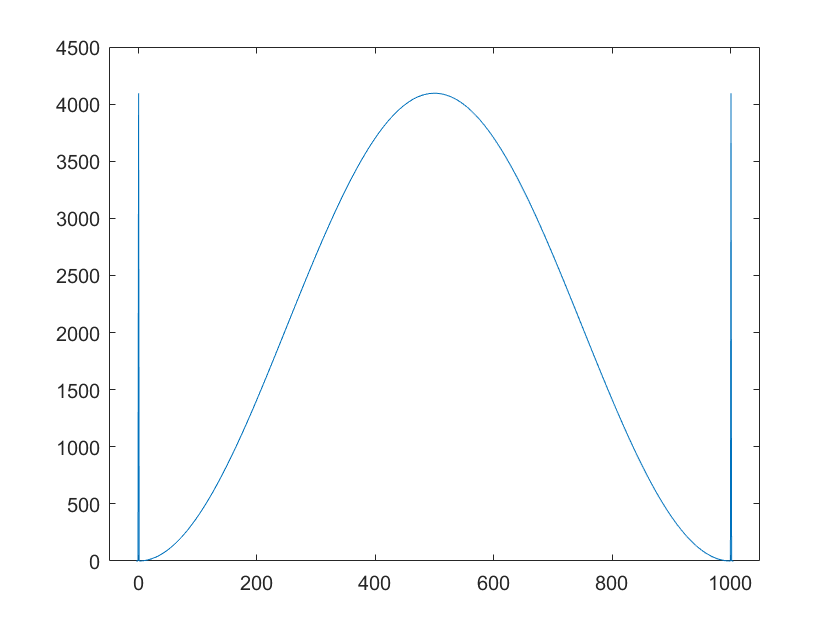
1. Multidirection Flex

Command: “http://192.168.0.107/RelayToI2C?Command=2”

This command outputs a single sinusoid of the form:

1. Unidirection Flex With Bounds

Command: “http://192.168.0.107/RelayToI2C?Command=3”

This command is the same as Command 1, but adds a 5 ms full voltage before and after the sinusoid

1. Step Signal

Command: “http://192.168.0.107/RelayToI2C?Command=4”

This command creates a step signal. By default this step signal last’s 3s, but the duration can be changed

1. Calibration Signal

Command: “http://192.168.0.107/RelayToI2C?Command=5”

This command creates a signal that outputs the floor voltage followed by the ceiling voltage for 0.5s each.

1. Change Freq

Command: “http://192.168.0.107/RelayToI2C?Command=6-[FreqInHz]”

Example: If you would like to change the sinusoid frequencies to 0.20Hz the command would be

“http://192.168.0.107/RelayToI2C?Command=6-0.2”

1. Change Step Duration

Command: “http://192.168.0.107/RelayToI2C?Command=7-[DurationInSeconds]”

Example: If you would like to change the step duration to 5 seconds, the command would be

“http://192.168.0.107/RelayToI2C?Command=7-5”

1. Change Volt Write Floor

Command: “http://192.168.0.107/RelayToI2C?Command=8-[Integer]”

The Teensy has a 12 bit DAC, thus it takes input from 0-4095. By default it uses that whole range but if you would like to change the floor then input a value between 0-4095 for a new floor value.

1. Change Volt Write Floor

Command: “http://192.168.0.107/RelayToI2C?Command=9-[Integer]”

The Teensy has a 12 bit DAC, thus it takes input from 0-4095. By default it uses that whole range but if you would like to change the ceil then input a value between 0-4095 for a new ceiling value.