Function Generator using OpAmp

This project showcases DIY Function generator with satisfactory range and accuracy

Ved RUDANI, 64

Vijay PANCHAL, 65

Semester 2 Project

Mentor: **Mr. D. B. PATEL**

Head of Department: **Dr. P. N. GAJJAR**

Gujarat University

April 14, 2023

Abstract

Function generator are useful tools in academia and industries. Mostly they are avalaible in market. In this project we are trying to understand and study simple frequency generators with use of OpAmp. We usec generic OpAmp Ic LM741, which is single package and easy to understand with benefit of extensive acedemic experince.

Acknowledgement

We would like to thank to the our Head of Department - Dr. P. N. Gajjar sir for their faith in us and supporting us in everyway. Special thanks to our respected mentor Dr. D. B. Patel sir for their support, encouragement and supervision in every step of this project. We also thank to all our respected professors for their support to complete this project successfully.
We would also like to thank scientists and authors on whom work we build our work.
We are also grateful of our classmates for their help and support for this project work. We heartly appreciate their contribution and thank them too.

Table Of Contents

1 Introduction

Function generator is circuit which generates periodic function with predictable frequencies with respect to time. Here, we will study only mono frequency generator but it can also generate superposed functions. Signals from Function generator comes in many forms but mostly it is either sinusoidal or square wave. We will generate sinusoidal, square and triangle wave as output.

We used basic circuits with few modification as our need. With use of IC LM741 we used OpAmp in our circuit.

For Sinusoidal wave we used Wein Bridge circuit, which is easy to understand and impliment. Also, wein bridge circuit is quite less noice compare to it's compitition RC phase shift Oscillator, which have more component than Wein bridge and more complicated to understand. For Square wave we used standard astable multivibrator cicuit, with little modification. Lastly, Triangle wave can be made from just attaching Integrator to our square wave output with some regulation.

Now, each circuit (this wave form generator) has different block, basically we divided whole circuit in there block. Main work for us is to combine all of this. We wandered across CMOS families, BJTs but finally we sattled into physical swith which is coupled for power transmission and also for output change.[2]

2 Blocks

As told in introduction each circuit is in their blocks. First block for sine wave which is nothing but wien bridge circuit, second is sqaure which is astable multivibrator, third for triangular wave which integrator attached to second block (square wave block).

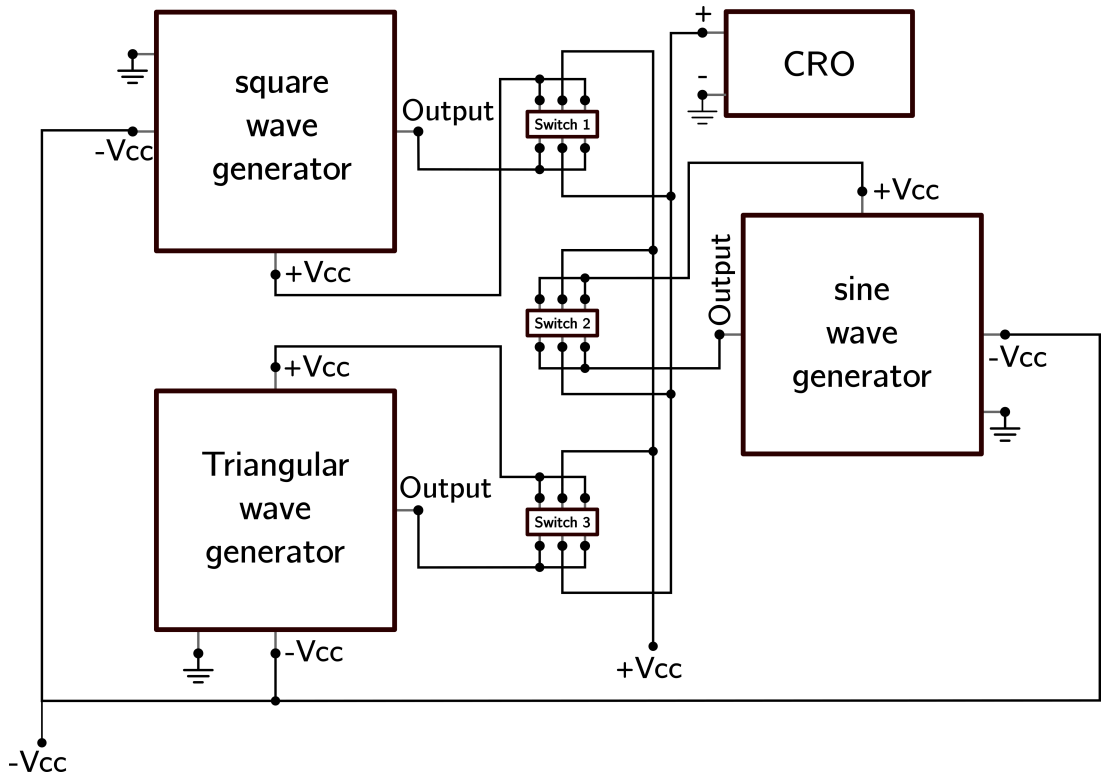

Figure 1: Block diagram of our function generator

2.1 Block 1: Sine wave generator

In first block, we have basic circuit of wein bridge. You can see in figure 1. In center we have OpAmp (IC LM741). This is amplifier with RC component attached with input and output. Here, at one end there is RC parallel component and at other end series RC component.

Here, frequency is given by,

$$f = \frac{1}{2\pi RC} \tag{1}$$

For sustaing oscilation gain must be 3 and for non inverting amplifier gain,

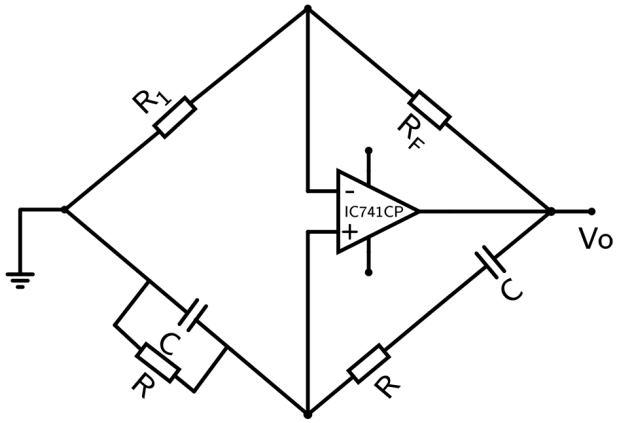

Figure 2: Wein bridge circuit

$$A = 1 + \frac{R_F}{R_1} = 3 \tag{2}$$

So, we get relation $R_F = R_1$

Here, you can see our block circuit, at the end we attached two zener diode for regulation to the output. As you can see OpAmp in IC LM741 package. Power supply given from 4 and 7 to 12V and -12V. We chose $R_1 = 12k\Omega$. By relation of R_1 and R_F, we got $R_F = 24k\Omega$.

For frequency range we used Potential with max range of $100k\Omega$. So, lowest and maximum frequency whould be (with constant capacitance at $50nF$),

$$f_{min} = \frac{1}{2\pi \times 100k \times 10n} \approx 159hz$$

$$f_{max} = \frac{1}{2\pi \times 100 \times 10n} \approx 159khz$$

So, frequency range would be $159hz$ to $159khz$

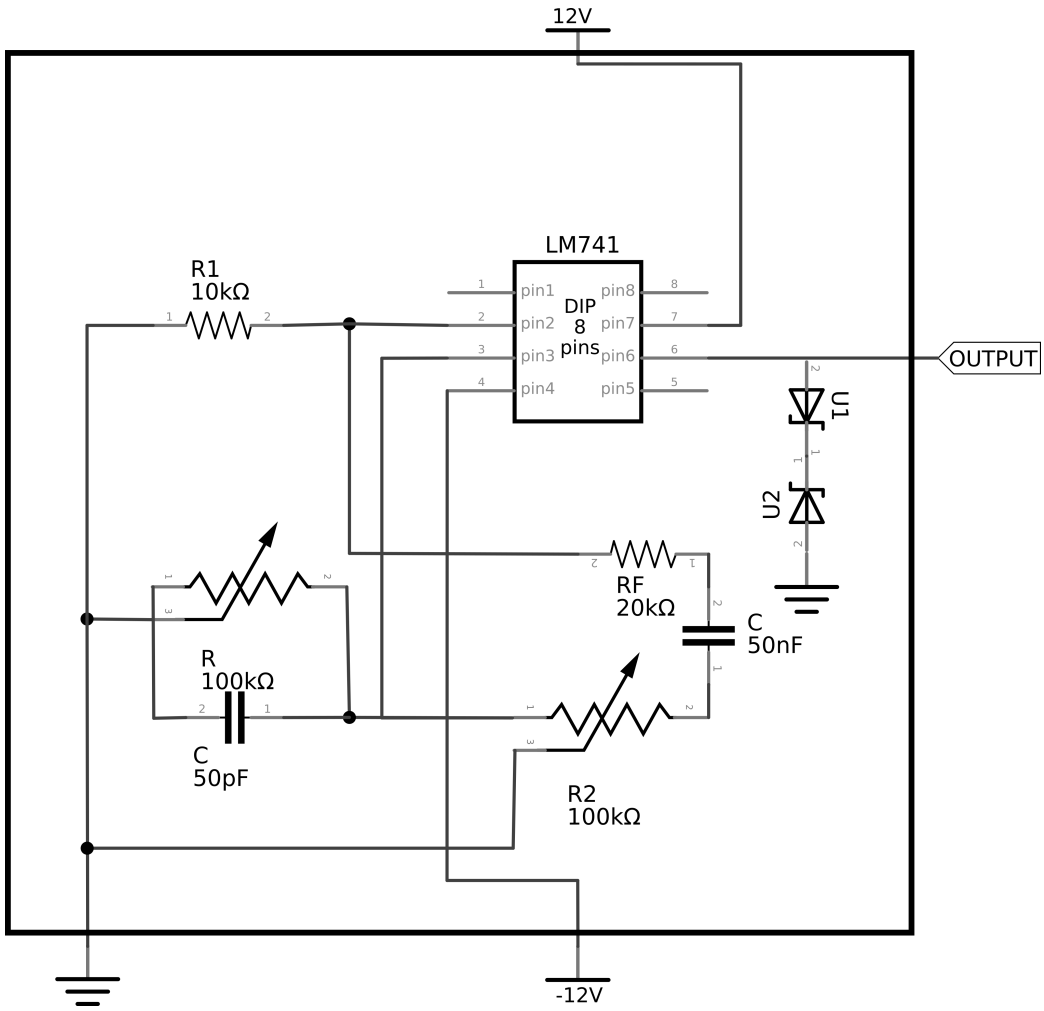

Figure 3: Our block 1, which consinst of IC741CP

2.2 Tuning sine wave generator

The output of sine wave generator can be little noisy. This will be reasonable as we will see it's working. Any sine wave generator will work as frequency extractor from DC or any AC levels. Since, Row signals have superposed waves in nearly all the spectrum, one have to rely on different filters and component which can attenuate desire frequency and theoretically minimize every other frequency.

In wien bridge, this is principle is mostly exploited. We have two RC components, one in series make low pass filter and secondly there is parallel component which work as high pass filter. In **fig-**

ure there is highlighted areas of both filters. Here, it is quite straight forward see that low pass will block higher frequency and high pass will vice versa. So, if we set both filter such that combination will give us some band (quite narrow band in fact). Center of this frequencies will cut off frequency of both filters. When wien bridge balances than this band of frequency will be resonated and give final output.

2.2.1 Series RC components in Wien bridge

Series RC component which works as low pass filter have this type of phenomenan, total V_{in} and V_{out} will be proportional to the to total reactance. With voltage divider low,

$$\frac{V_o}{V_i} = \frac{X_c}{R + X_c}$$

Where, X_c is reactance of capacitor valued as $\frac{-j}{wC}$. So,

$$\frac{V_o}{V_i} = \left(\frac{1}{1 + w^2R^2C^2}\right)^{\frac{1}{2}}$$

If we take w_0 as breakpoint or curoff point for our RC component than $w_0 = \frac{1}{RC}$. Here RC is time constant.

Graph of low pass shown in figure 8a. Where we can see frequency equals to $w = w_0$ at some point. Also, notice that even though we have cutoff frequency at w_0, there is enough frequencies around w_0. Basically filters always have some noise which does not filtered. Here, if you use higher order filter than this slope of voltage to frequency would be slightly higher. With sufficiently high order filter you can make abrupt change in frequency domain, but this comes with it's consequences. With higher order filters other noises dominates since we will have too much components. We will

use second order filter here, which is quite balance in accuracy and component noise.

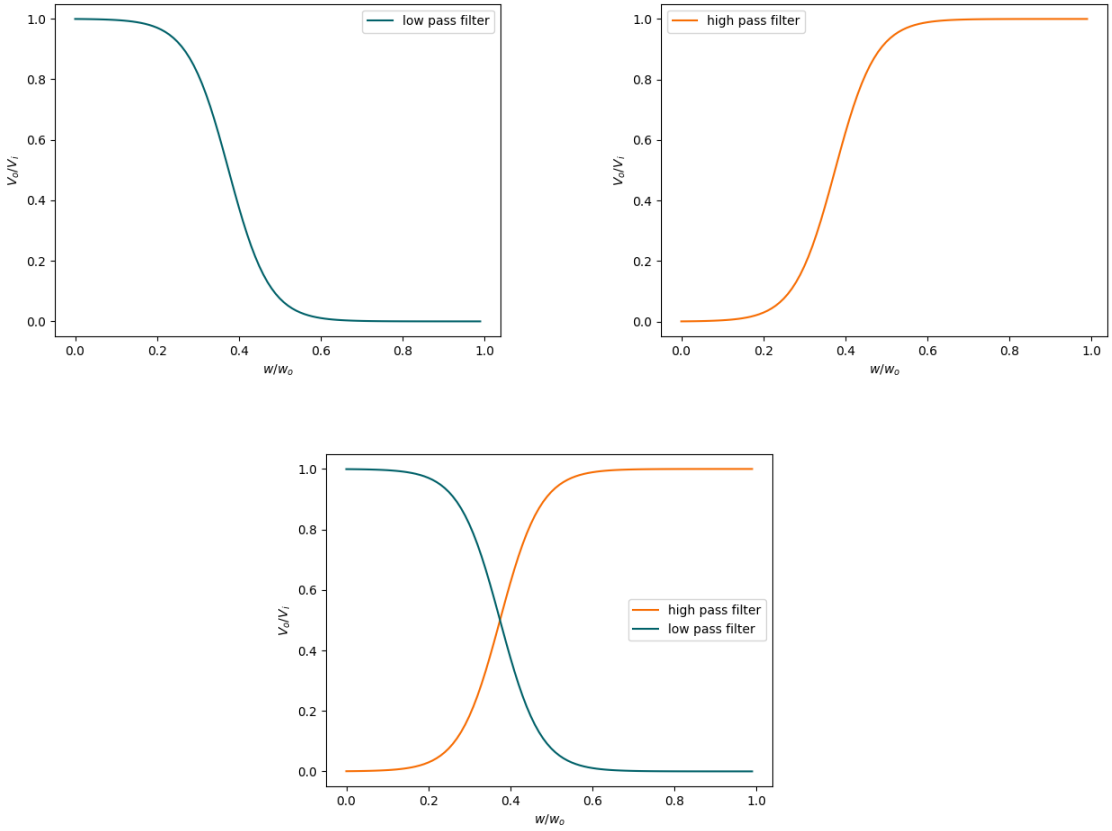

Figure 4: here, we have a) low pass filter, b) high pass filter and c) combination of high and low pass fiter

2.2.2 Parallel RC components in Wien bridge

Similarly to that of series RC components, we can define high pass filter as parallel RC component. In parallel circuit when frequency increases reactance decreases and total reactance decreases. So, consequently higher frequency pass and lower frequency will not. Reactance of high pass filter would be following,

$$\frac{V_o}{V_i} = \frac{R}{R + X_c}$$

Again, X_c is capacitance reactance and valued at $-frac-jwC$

$$\frac{V_o}{V_i} = \left(\frac{R^2}{R^2 + \frac{1}{w^2 C^2}} \right)^{\frac{1}{2}}$$

This relationship is shown in figure 8b. With cutoff frequency at w_0. As we can see here also noise of unwanted frequency range are here.

2.2.3 Total signal and Error terms

In wien bridge we have both the low pass and high pass filters. So, total response of that shown in figure 8c. Here, we have gain frequencies in range between cutoff frequency. Since, this range amplify in non inverting amplifier and feedback. This frequency will resonant and becomes our output signal. From now on, we will say w_0 as resonant frequency. Final output in our theoretical studies will be this resonant frequency. Practically this frequency is observed with error frequencies.

Error terms in here will be in following cases. *1) since we have band, we get many frequency output from the band, which is quite distorted in itself.* and *2) here working of filters are note up to expectation and we have noise from whole spectrum of frequency.* This is quite headache, unfortunately we have both the cases in our experiment.

2.2.4 Fourier analysis of Output signal

We can minimize this errors by using Fourier analysis of output signal. As one can say that DC level is made of superposed infinite number of waves with different wavelengths,

$$DC_{level} = \sum_{n}^{\infty}(a_n \cos(w_n t) + b_n \sin(w_n t))$$

Here, a_n and b_n are coefficients of Fourier series. What wein bridge does is extract desire frequency from DC level.

In our experiment we got distorted sine wave which means their is higher frequencies in effect. Also after some values of Potentiometer, there is just square signal. Another distortion occur was from lower frequencies manly $50Hz$ and around $300Hz$, which are making signal less stable and sometimes dominates resonant frequency.

For higher frequency, we got idea to put low pass filter around value of resonant frequency that would bring signal to more on resonant frequency. This is can be seen in block diagram of sine wave from figure 3 and figure below 5. This should give us better results ad we intended.

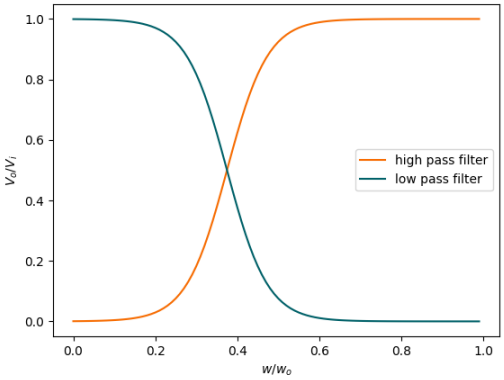

Figure 5: low pass filter at the output of our signal

For lower frequency, we have high pass filter, which eliminate those lower frequencies and stabilize our signal. This can be shown from block diagram figure 3 and figure 6.

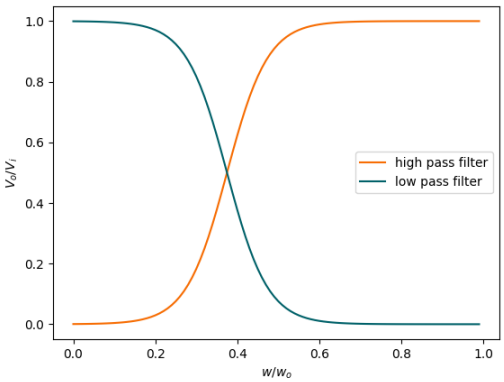

Figure 6: high pass filter at the output of our signal

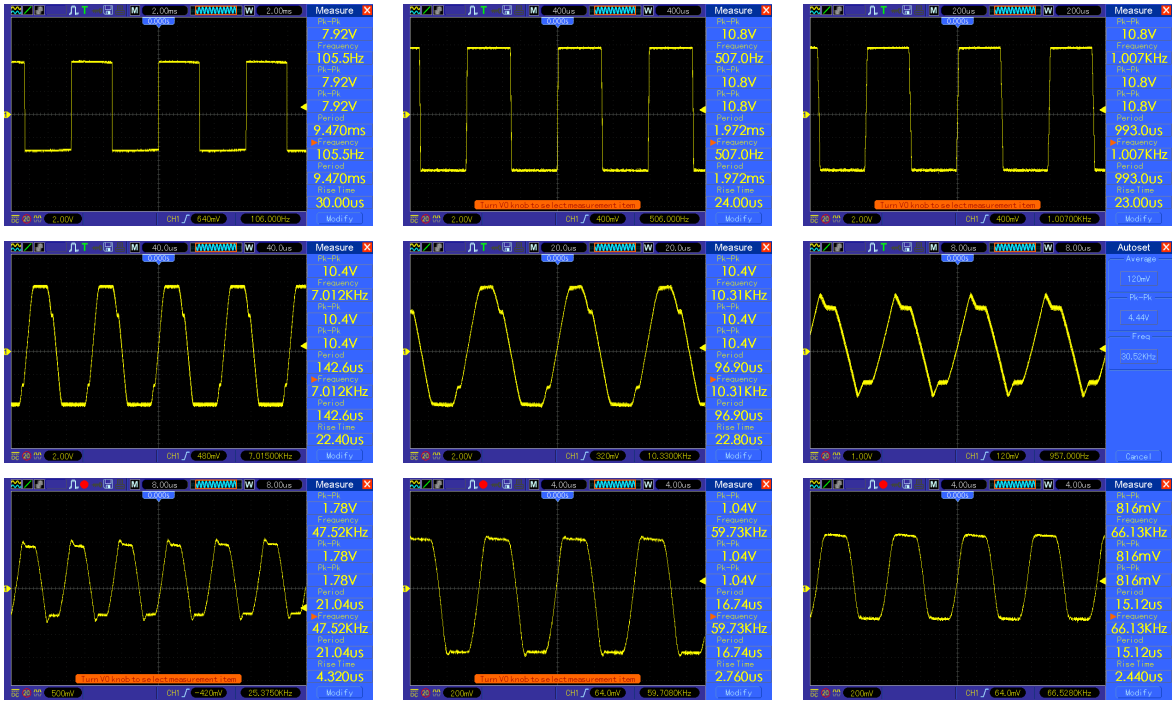

Figure 7: You can see all square wave outputs left to right respectevely 100Hz, 500Hz, 1kHz, 7kHz, 10kHz, 30kHz, 47kHz, 60kHz and 66kHz

2.2.5 Output of sine wave after tuning

The output which we expected from our upper analysis at different frequency is shown below in figure **??**. The frequency range of

sine wave output is given below in table. You should know that this $V_{out}p-p$ is after applying all the filters and tuning. Original output is quite large in peak to peak voltage around 5 times big.

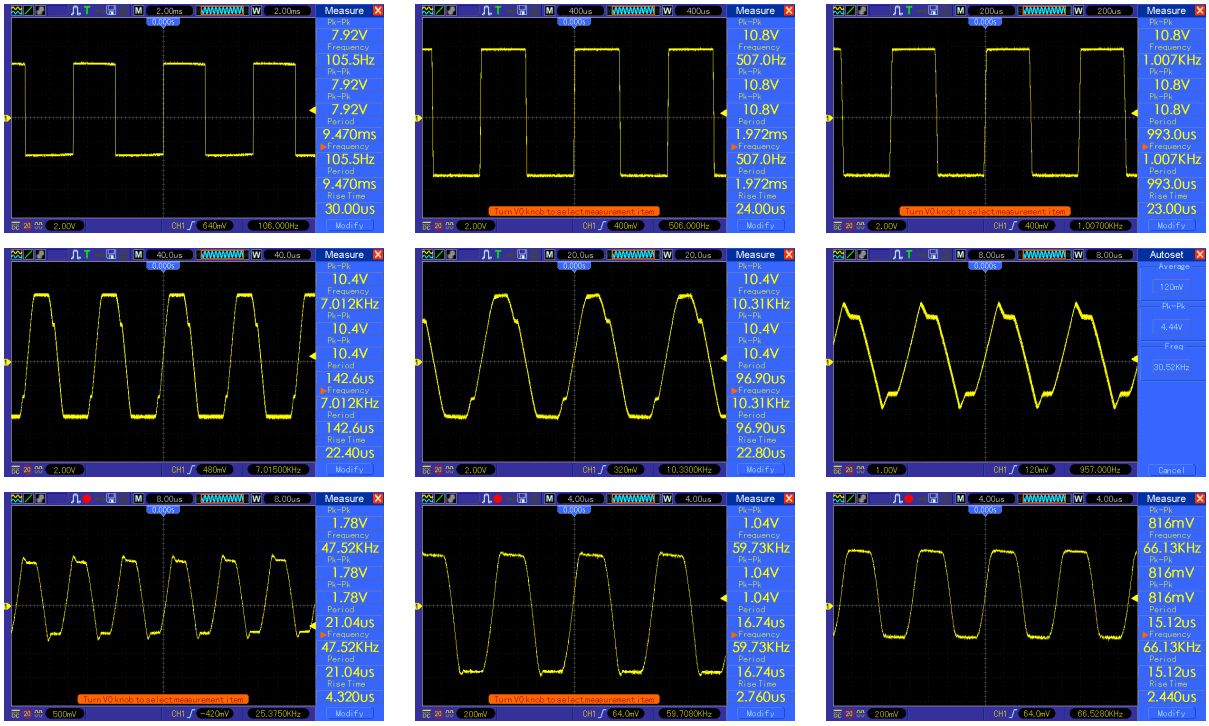

Figure 8: here, we have a) low pass filter, b) high pass filter and c) combination of high and low pass fiter

2.3 Block 2: Square wave generator

As square wave generator we have basic astable multivibrator. This circuit works on scenario where output will have to stable state and it will swing between them, hence the name. When circuit is $+V_{sat}$, we will have high signal output and when circuit is $-V_{sat}$, we will have low signal output. So, we will have square wave as desired. The circuit for astable multivibrator is shown below.

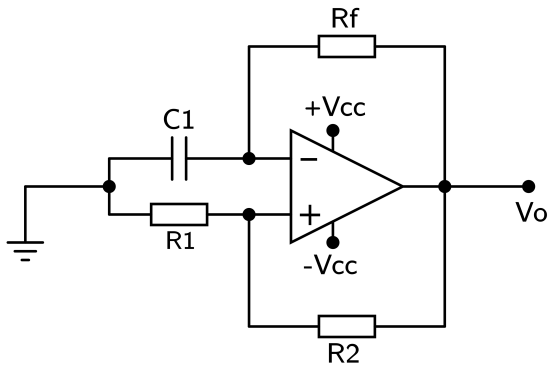

Figure 9: astable multivibrator circuit

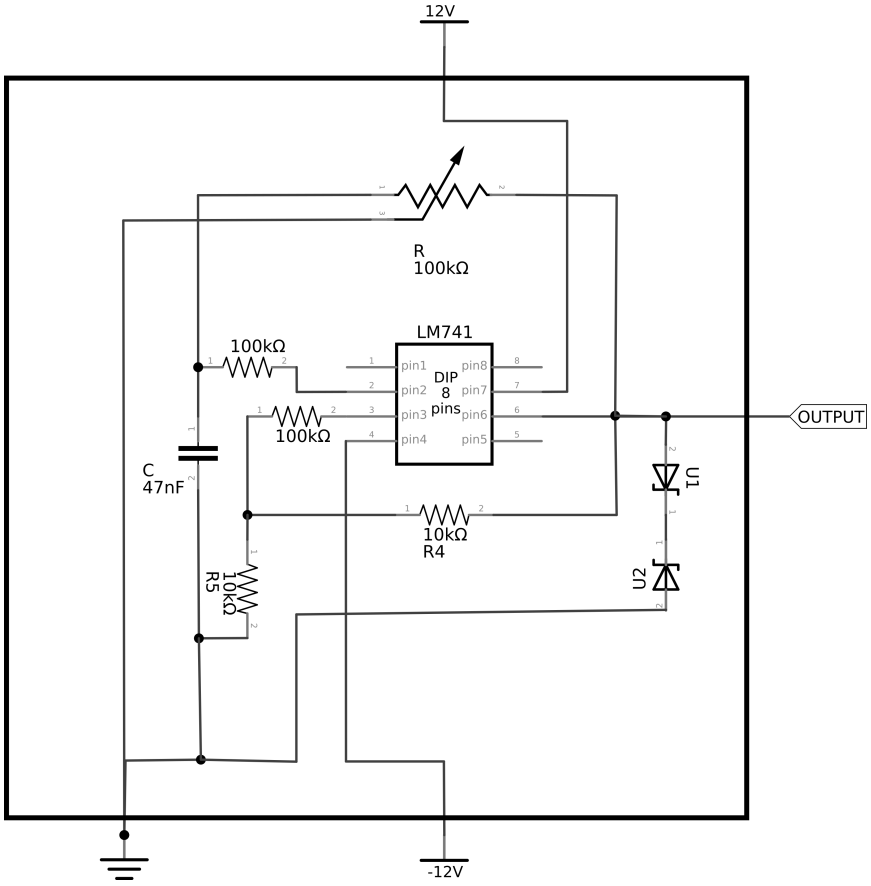

Figure 10: second block: square wave generator

Here, frequency would be,

$$f = \frac{1}{2RCln(\frac{2R_1+R_2}{R_2})} \tag{3}$$

If, we take $R_2 = 1.16R_1$ then,

$$f = \frac{1}{2RC} \tag{4}$$

Here, we took $R_1 = 10k\Omega$ and $R_2 = 11.6k\Omega$ such that $\frac{R_2}{R_1} = 1.16$. Also, you can see that we employed $100k\Omega$ in input terminals for accurate and reliable signal.

Frequency range would be of (for constant capacitance at $50nF$),

$$f_{min} = \frac{1}{2 \times 100k \times 50n} \approx 100hz$$

$$f_{max} = \frac{1}{2 \times 100 \times 50n} \approx 100khz$$

2.4 Block 3: Triangluar Wave generator

We basically extend block 2 with integrator circuit. Which would give triangular wave as intended. Here, this integrator circuit differs from basic circuit that $100k\Omega$ as feedback resister is joined. Which would give better stability and accurate output. Circuit diagram is shown below,

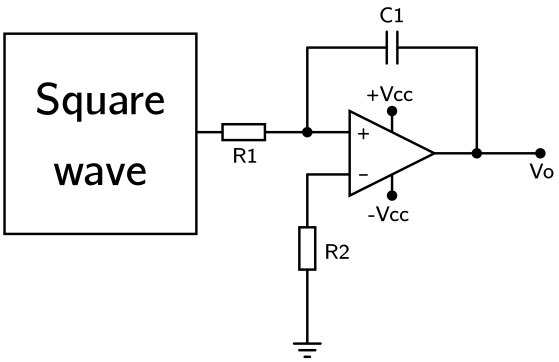

Figure 11: integrator circuit with square wave as input

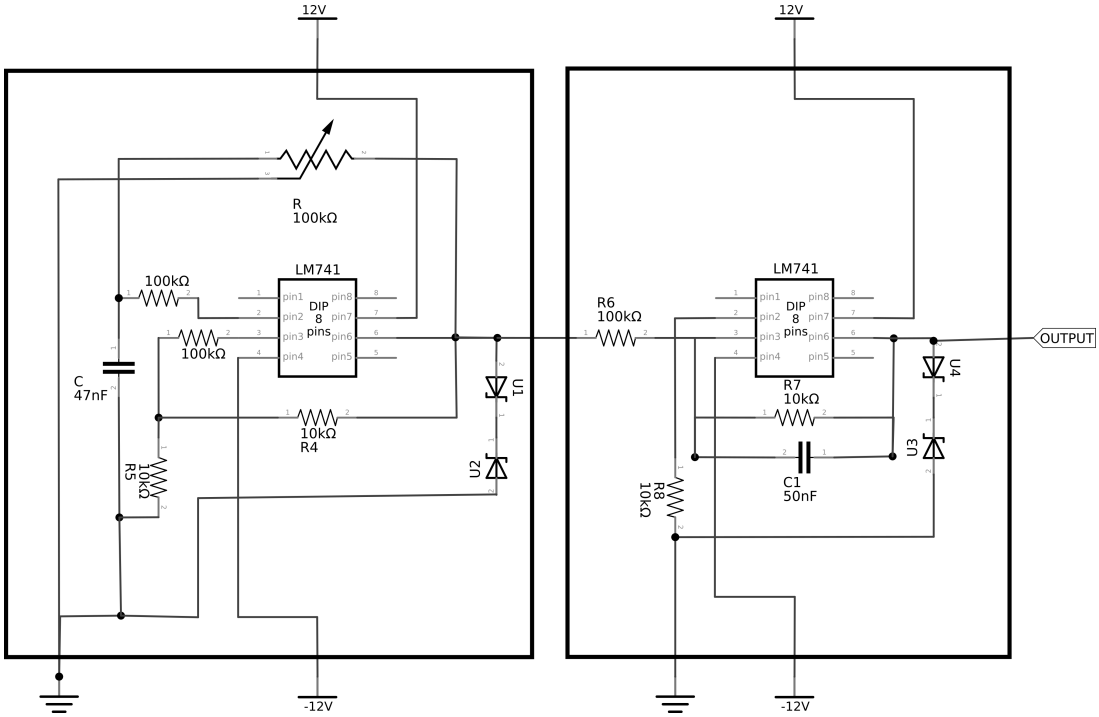

Figure 12: block 3: triangluar wave generator

Here, R_4 have to be $10R_3$. Frequency is give by same relation as block 2.

3 connection and switching

For connection of all this block we have used push pull button with two poles. One for power controlling and other for output controlling. Basic diagram of this switch is drawn in figure below.

When switch is **ON** (means pushed) it will connect 1 terminals with common and complete the circuit. When switch is **OFF** (pulled condition), the circuit will open and we will not get connection.

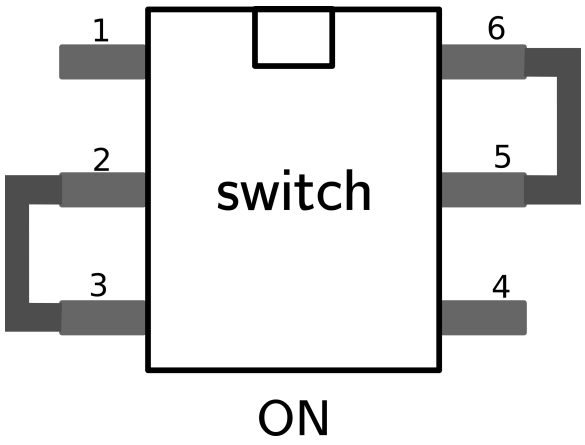

Figure 13: switch on state for two pole push button switch

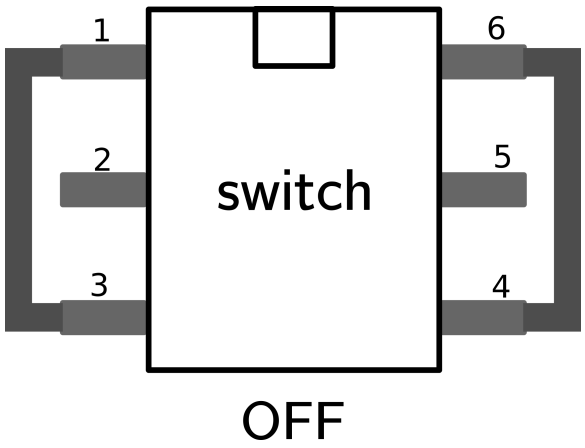

Figure 14: switch off state for two pole push button switch

The +Vcc in common (upper common) is completely independent of Output terminal common (lower common). Which means

switch can completely operate two tasks, which is when on it power the block and take output and give to CRO. You can see this is on block diagram in figure 1 .

4 Output

The output of whole circuit which is single output after connecting switches, will goes to CRO (cathode ray oscilloscope), which will measure gain and show signal form.

Theoritically, it shoult give exact signal but errors from ICs, connections, components are reasonable. We are tried our best to minimize it with simulation in Pspice simulations. But real life and simulations are distanced things. We are expected to see some divergence.

5 Pspice simulations

We did Pspice simulation In `https://www.falstad.com/circuit/` [1] by Paul Falsted. Here are simlations result from different blocks. This outputs are for Potentiometer valued at $3.3k\Omega$. We gain peek to peek voltage value at $2.8917V$ for sine wave and $2.11V$ and 2.2 in square wave and triangular wave respectively. This figures are from matplotlib [4][3], since we could not get from falsted. We got accurate p-p voltages.

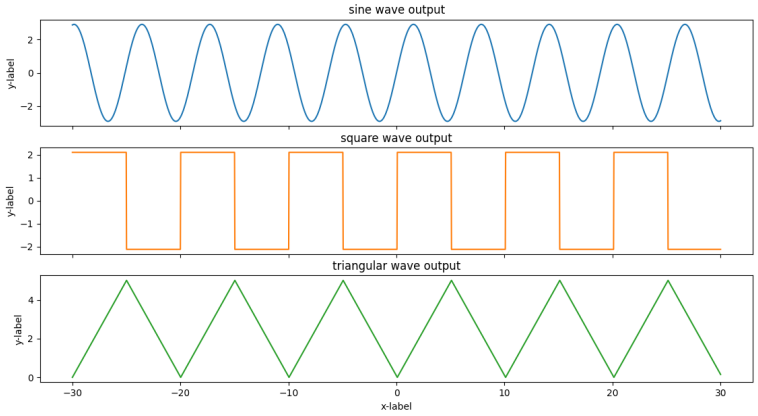

Figure 15: Outputs

References

[1] Paul Falsted. Online circuit simulator. `https://www.falstad.com/circuit/`. Accessed: 2023-03-30.

[2] Ramakant A Gayakwad. Op-amps and linear integrated circuit. 2012.

[3] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[4] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.