Waveform Generator For Communication Lab

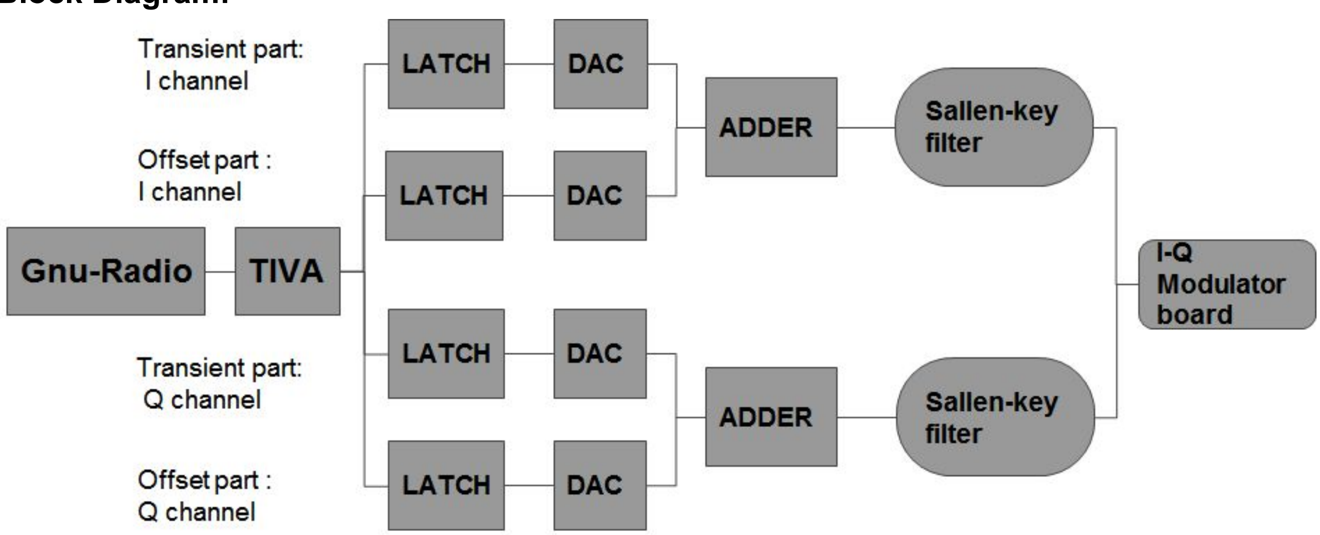
Faculty - Prof. Shalabh Gupta
TA-Wilson
Surya Iyer-130070004
Ayush Shringi-130070014
Raunak Suryavanshi-130070024

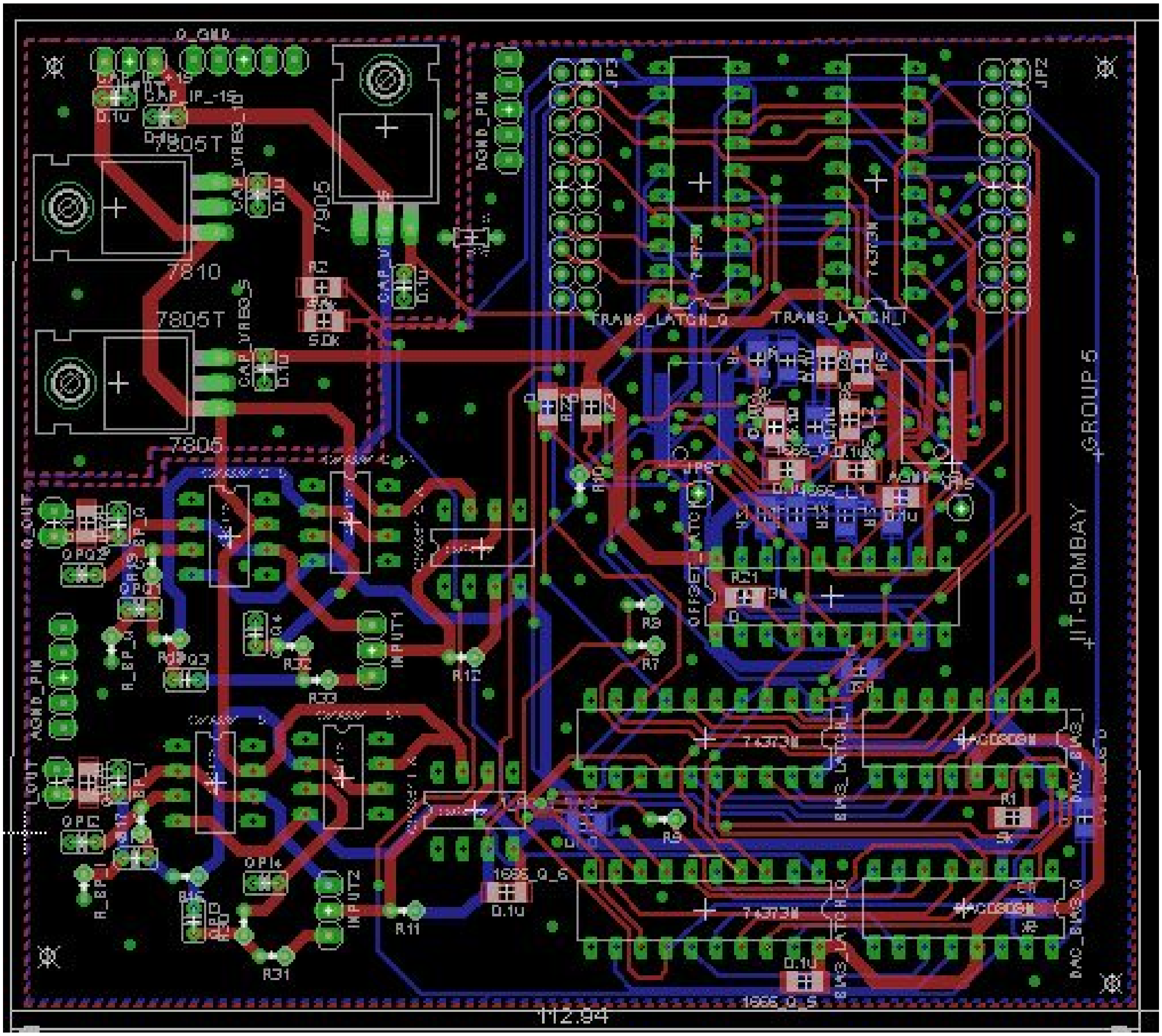
Abstract:
Arbitrary waveform generators are very costly. But we may not need all the functionalities that this complex AFG offers. To generate waveforms within a given voltage and frequency range, we will build very low cost and portable waveform generators.

Aim:
We wish to generate the waveforms of frequency and voltage range that are used as inputs to I-Q modulator boards.

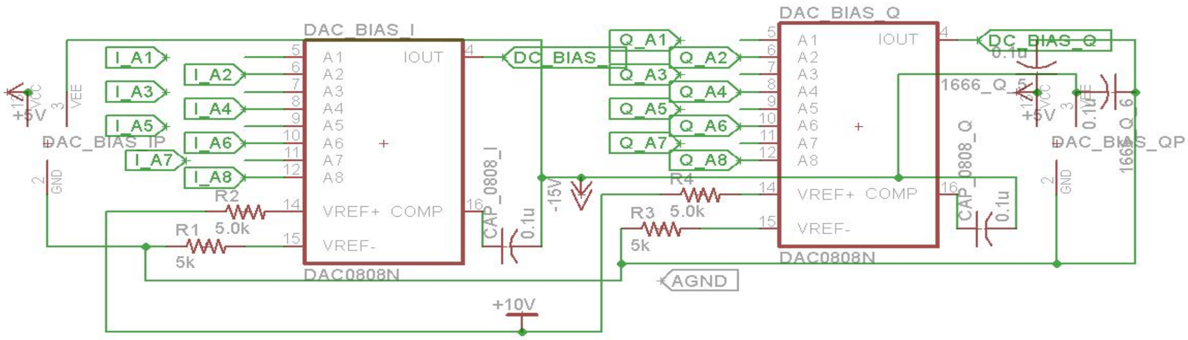
Introduction:
We will be using GNU-Radio, an open source software, to generate and sample the waveforms which are then sent to TIVA for storage of samples. Thereafter Tiva sends these samples to the hardware circuitry.

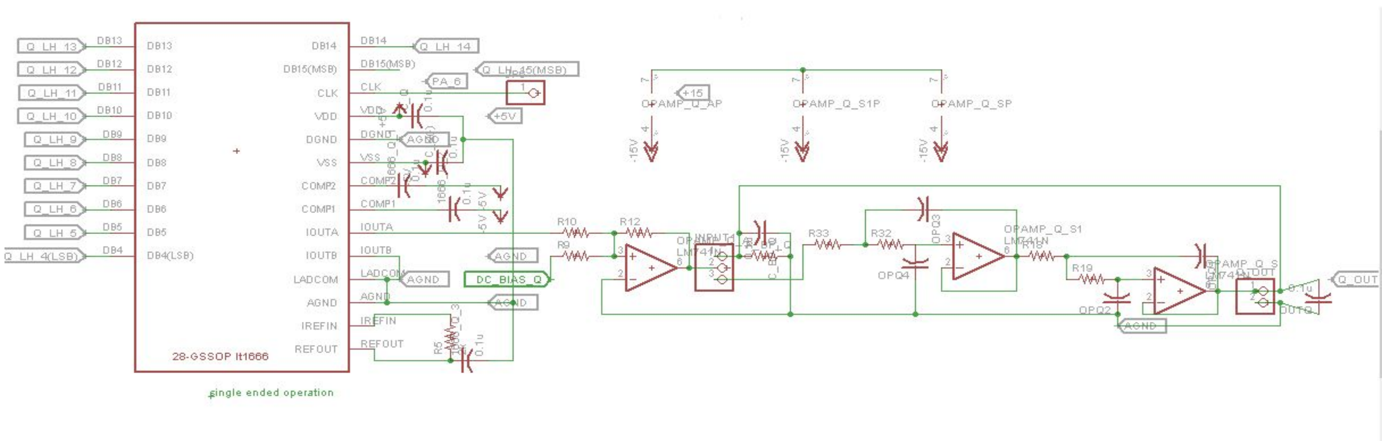
Block Diagram:





7805T
7810
7805T
7805
7905
TRANS_LATCH_Q
TRANS_LATCH_I
GROUP 5
IIT-BOMBAY
JP3
JP2
INPUT1
INPUT2
74373N
CD809N
BIAS_LATCH_I
BIAS_LATCH_Q
DAC_BIAS_I
DAC_BIAS_Q
OFFSET_LATCH
112.94





Block diagram description:

GNU radio:

GNU Radio is a free software development toolkit that provides signal processing blocks to implement signal processing systems.

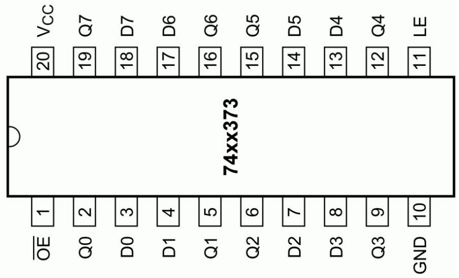
We are using in-built signal source block which generates samples of a standard wave(sine, triangle ,etc) and its output is given to a custom block made by us which transmits it to the TIVA board.

TIVA:

Tiva-c Launchpads are inexpensive, single-board microcontroller with a high frequency crystal (80MHz). Tiva’s job is to receive and store the samples transmitted from GNU radio and then continuously send these samples to hardware circuitry.

Latch :

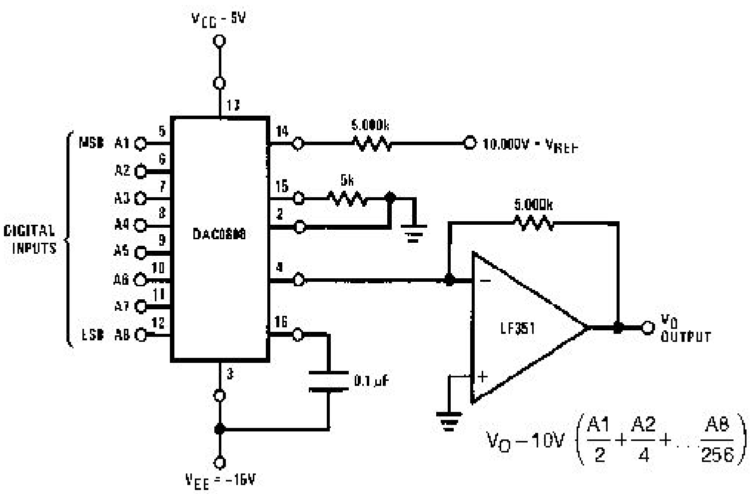
74373 is used for latching. It is 8 bit latch.We have used four latches because TIVA has limited number of output ports.



DAC :

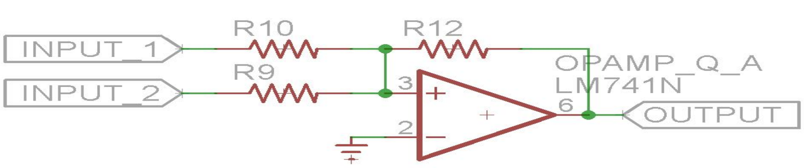
Two types of DAC's are used:

a. DAC0808 : It is a 8 bit dac. We are using it for getting analog offset value.
b. LTC-1666: It is a 12 bit dac. It is a very fast dac. We will use its lower 8 bits for transient waveform. Its upper 4 bits will be used to provide a constant voltage.

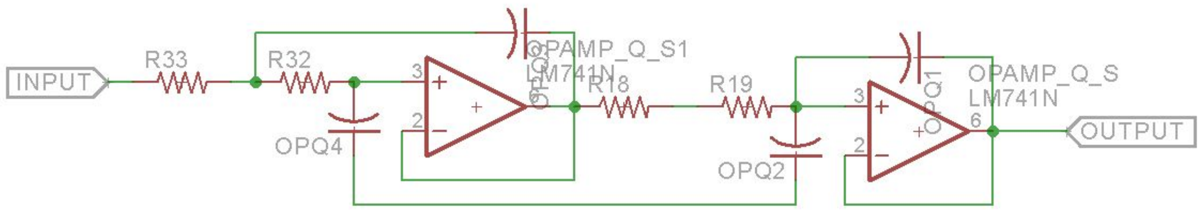


ADDER:

A simple adder is made using opamp(OP-741) to sum the offset and the transient analog values.



Sallen-Key filter:



Its a 4th order low pass filter that we are using to remove glitches from our final waveform.
Poles are at 34kHz and 3 poles at 72 kHz

Waveform :

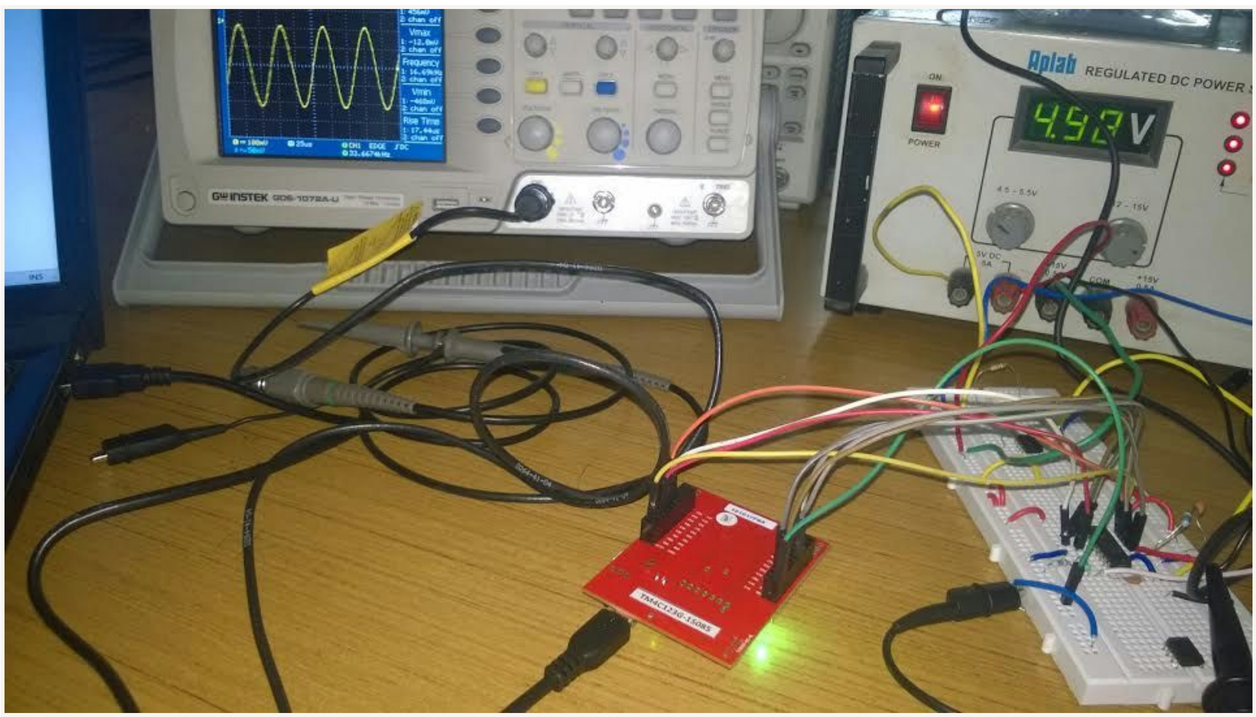


Figure1: Sine wave (16kHz)

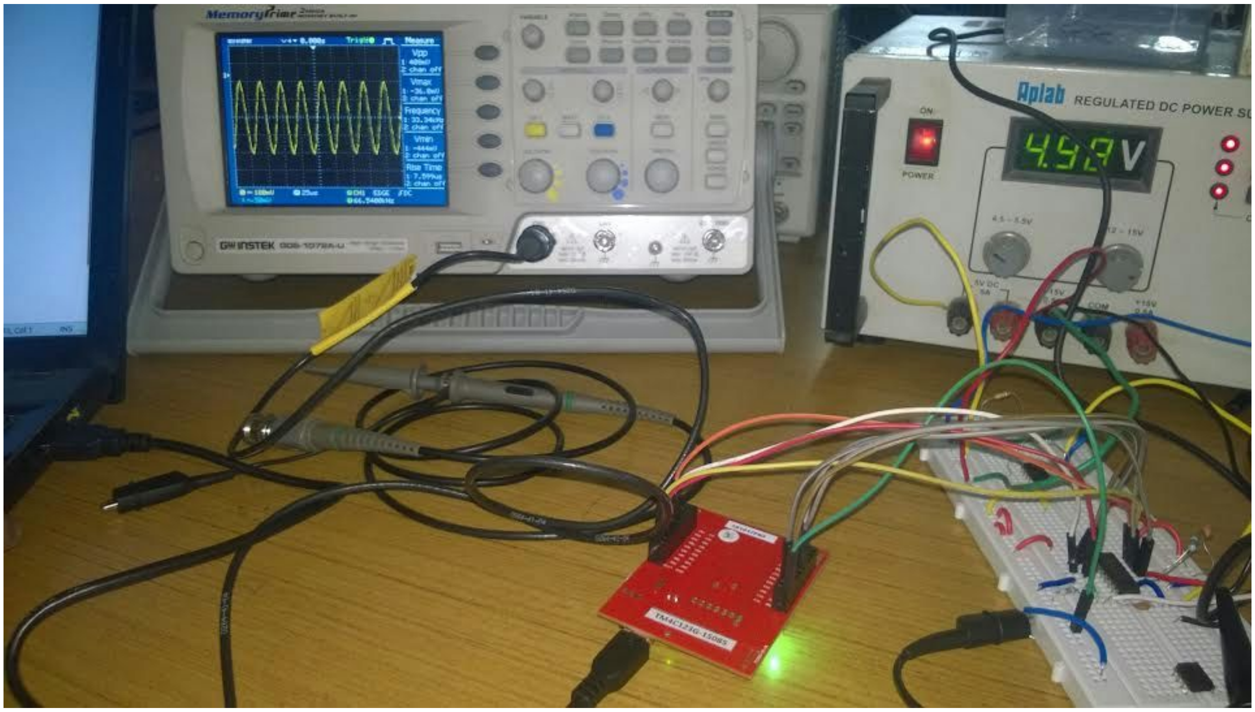


Figure2: Sine wave (32kHz)

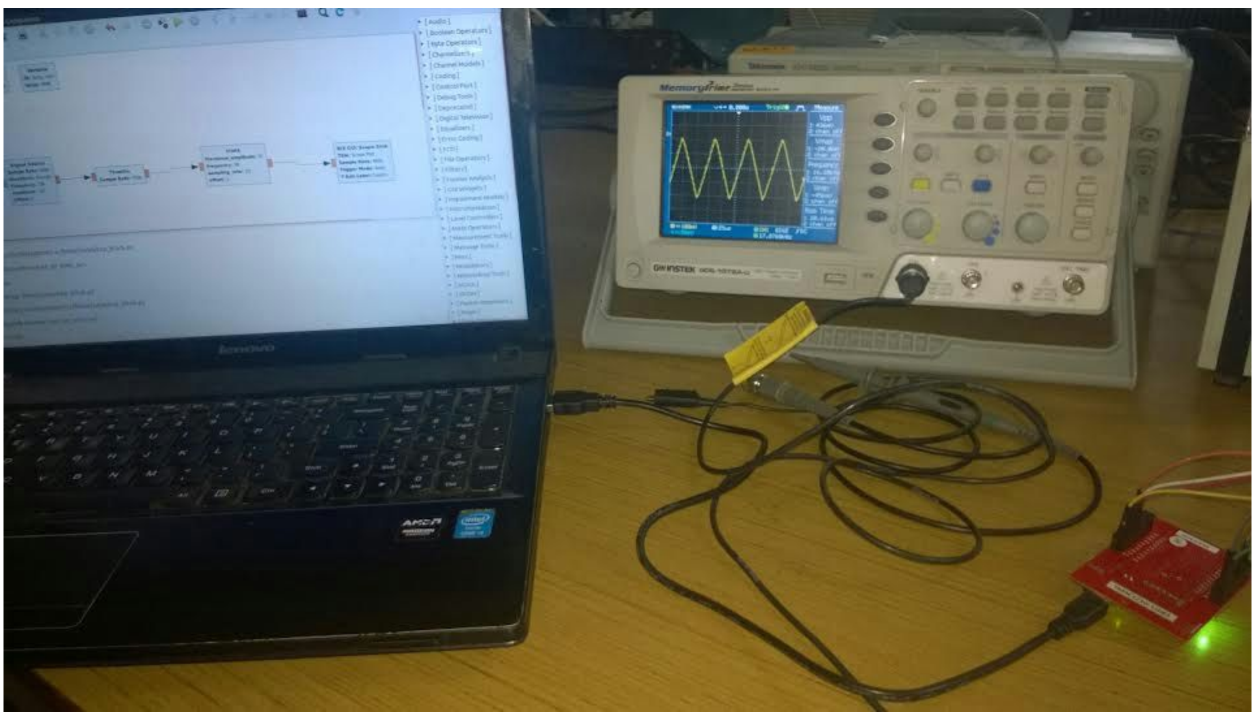


Figure3: Triangular wave

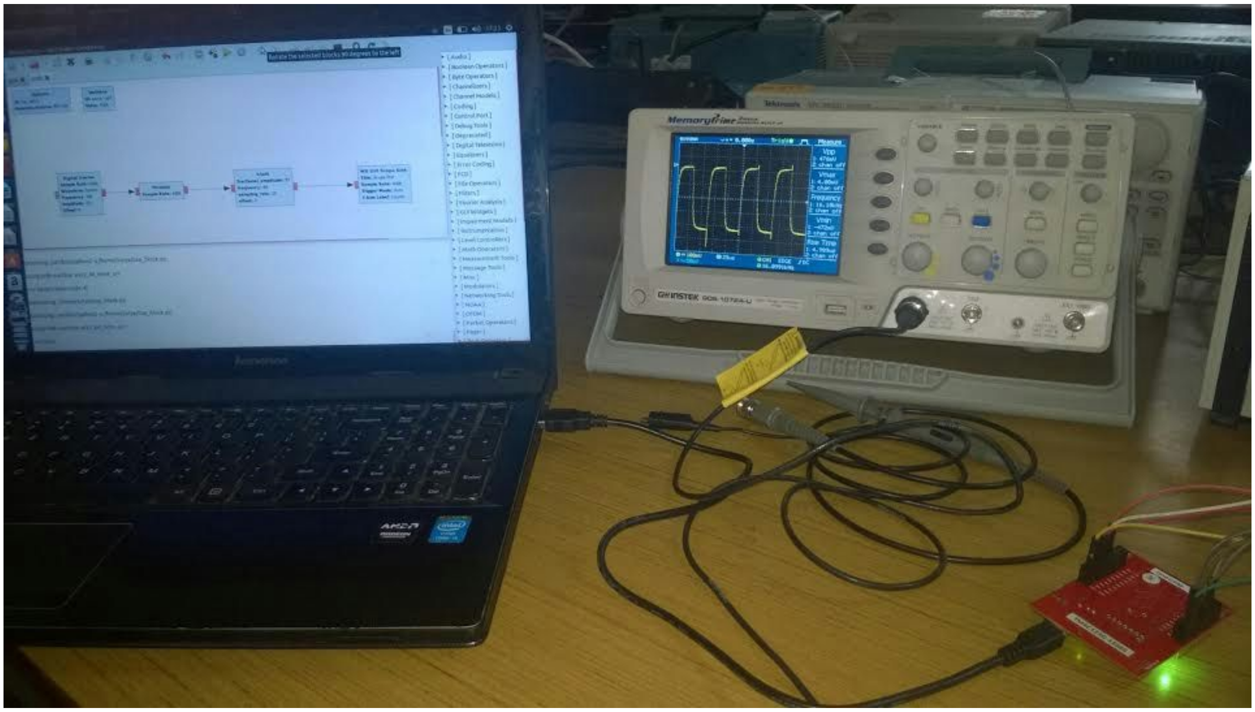


Figure4:Square wave

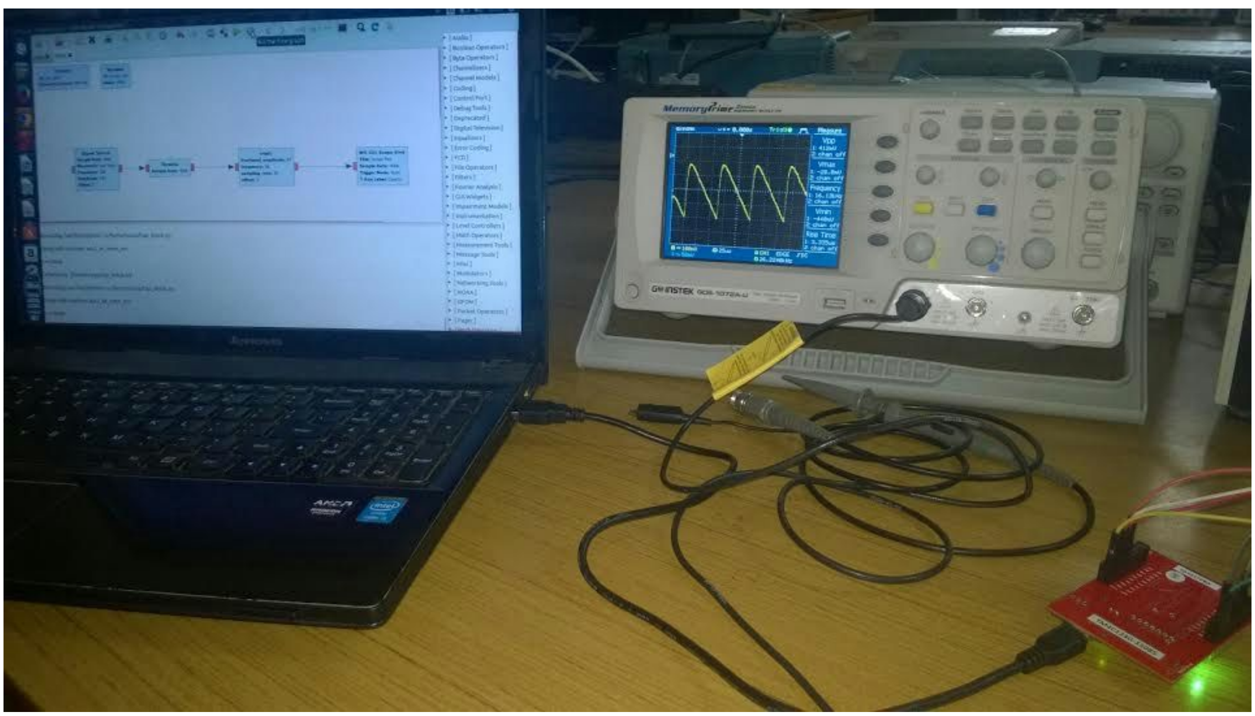


Figure5: Sawtooth wave

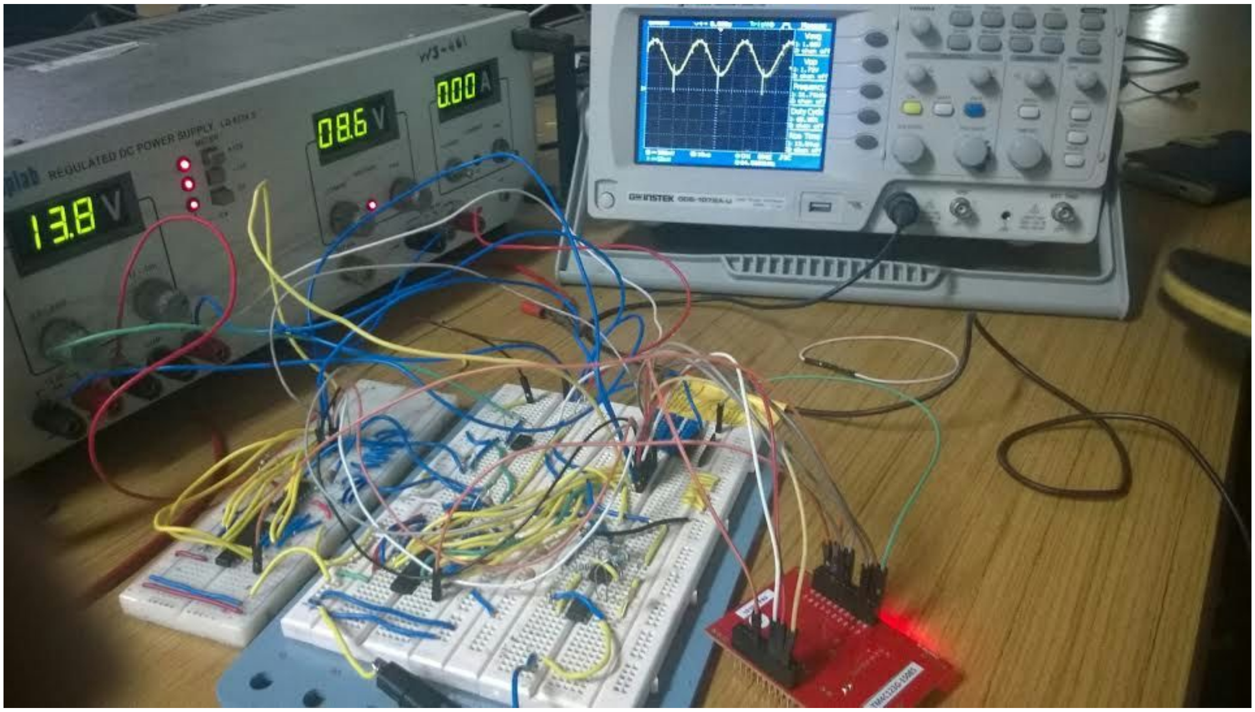


Figure6: Testing to hardware circuitry

Results/Achievements:

1. We are able to take samples from a standard signal source and communicate them to TIVA successfully through our custom block.

2. As of now we have successfully sent offset and transient samples via two latches to two demo DACs (0808) and then obtained the output by adding them and passing it through a 4-pole sallen-key filter.

3. We have hence obtained standard waveforms such as sine, square, triangle and sawtooth.

Problems Faced:

1. While transmitting from GNU-radio we were getting 8 samples per wave when sampling rate was 10 times the frequency when ideally we should get 10 samples per period .

2. We were getting spikes in the waveform. So we used 4th order sallen-key filter to filter out the high frequency glitches.

3. We were unable to use Port-C of Tiva. Port-C has 4 JTAG pins and 4GPIO's. We initialized only the GPIO's but still when we uploaded this code on Tiva, it got damaged and couldn't be further programmed.

Conclusions:

We are able to generate different kinds of waveform within a voltage and frequency range. The final testing with the PCB is still remaining as we haven't yet received our PCB.

Future Work Suggestions:

Our design can be improved by generating higher order frequencies. For this we have to study Tiva in more depth, for finding if such a possibility exists in Tiva. Or else we can switch to some other microcontroller that has higher frequency crystal.

The other thing is that we can make our design real-time i.e. Tiva will keep on transmitting data to hardware in parallel to GNUradio sending samples to Tiva. This can be implemented using a buffer. The amount of useful data stored in the buffer will signal Tiva and GNU radio to adjust their transmission rate.

References:

1. http://www.ti.com/tool/ek-tm4c123gxl
2. https://gnuradio.org/redmine/projects/gnuradio/wiki/OutOfTreeModules
3. http://www.ti.com/product/DAC0808
4. http://ecee.colorado.edu/~mcclurel/sn74ls373rev5.pdf
5. http://www.ti.com/lit/ds/symlink/lm741.pdf
6. http://www.alldatasheet.com/view.jsp?Searchword=LT1666
7. http://sim.okawa-denshi.jp/en/OPseikiLowkeisan.htm