Micro Processor [EE 337] Mini-Project Report

Tone Generator

Group No 3, Tuesday Batch

Group Members

1. **Mehul Shah- 130020090**
2. **Kalpesh Patil - 130040019**
3. **Krishna Reddy- 130020118**

RA/TA

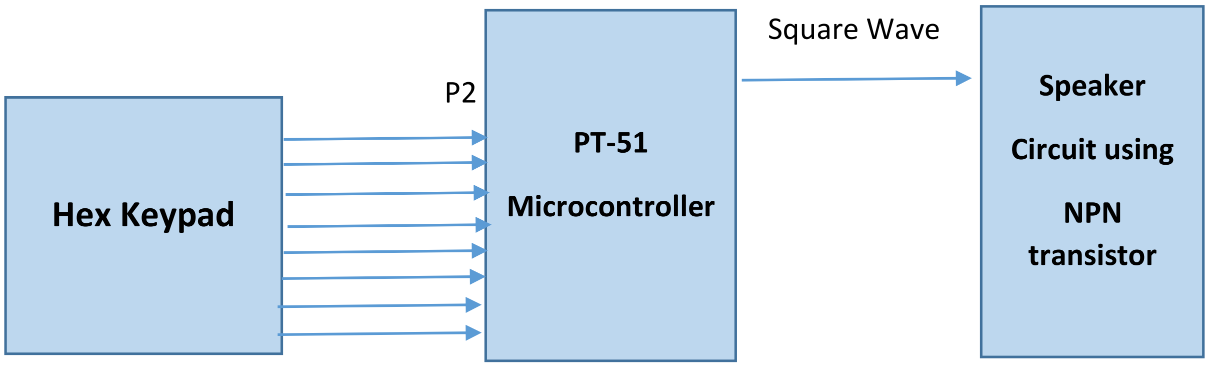
1. **Shubham Joshi**
2. **Ashutosh patil**

Problem Objectives and Deliverables

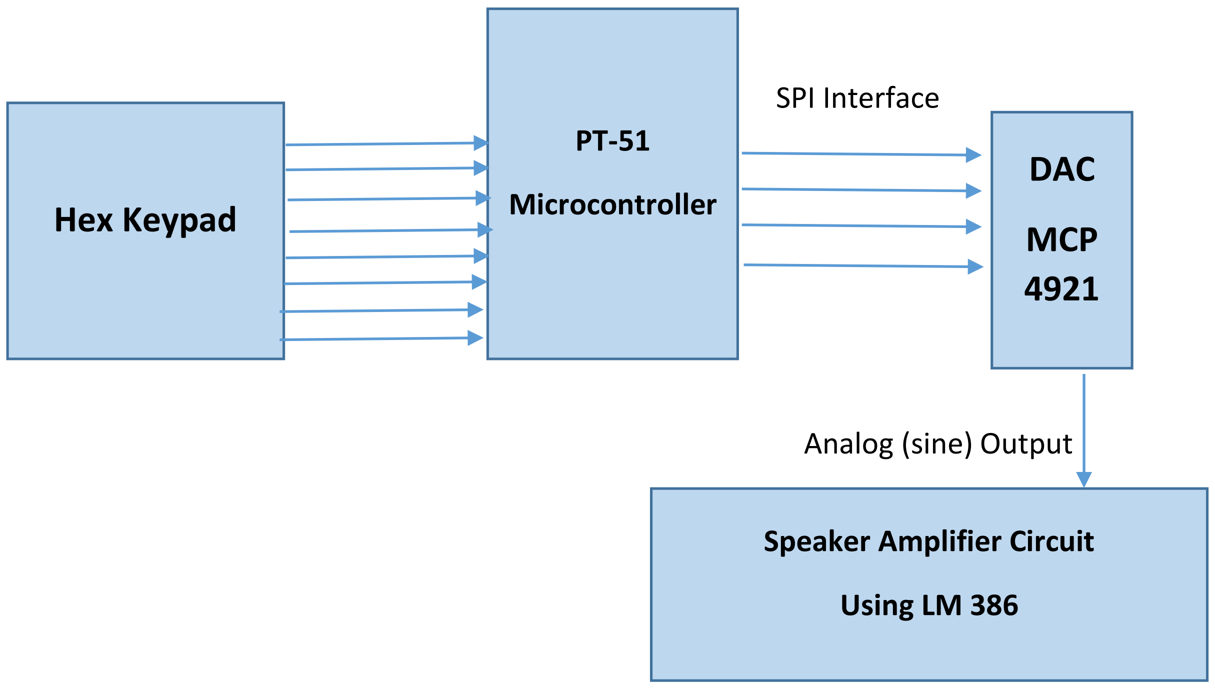
- Generate different sound tones using pt-51 board and play on speaker. Input should be taken from Hex Keypad.
- First 12 keys in first 3 rows of keypad should be used to select which note should be played. On the last row, first 3 keys should be used to select octave.
- You need to write 2 algorithms to generate tones. First one generates notes using a square wave and second one uses a sine value lookup table to generate frequencies.
- Once a tone is generated, it should continue to play till next input is selected. Observe the difference between generated 2 tones.

Block Diagram of Design

Part 1: To generate square wave



Part 2: To generate sine wave



Design Description

Part 1: Square Wave

1. Hex keypad is interfaced with Port 2 of pt-51. Value of frequency to be generated will be taken from Hex keypad (intersection of columns and rows).
2. TH0 and TL0 values corresponding to the required frequency square wave were calculated and stored in lookup table; say I need to output 480 Hz frequency square wave, time period corresponding to that will be 0.00208 sec. TH0,TL0 values will be corresponding to half this value.
3. This square wave is fed to npn transistor to generate sound from the speaker.

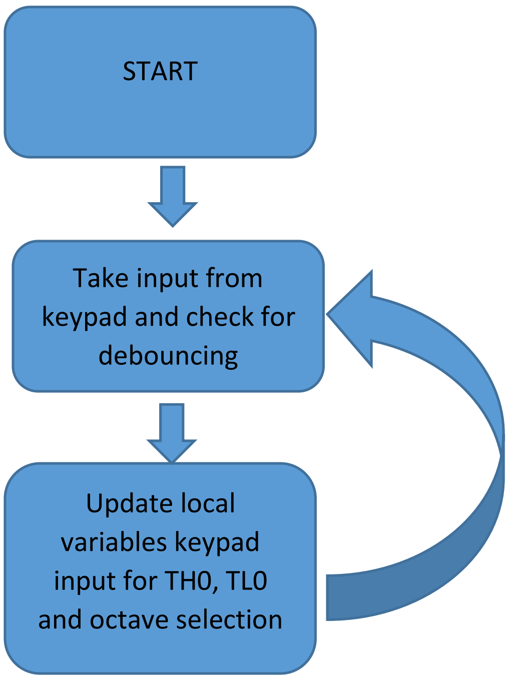
Part 2: Sine wave

1. Value of frequency to be generated is taken from Hex keypad similar to the square wave part.
2. We divide interval of $[0,2\pi]$ in 24 equal parts and store the corresponding sine values in a lookup table. Since input of DAC cannot be negative, we send 1+sinX as the input to DAC (MCP 4921). Input bit stream of 12 bits are formed according to the formula **$2047.5 * (1+ \sin 15i)$; $0 <= i < 24$** (approximated to the nearest integer)

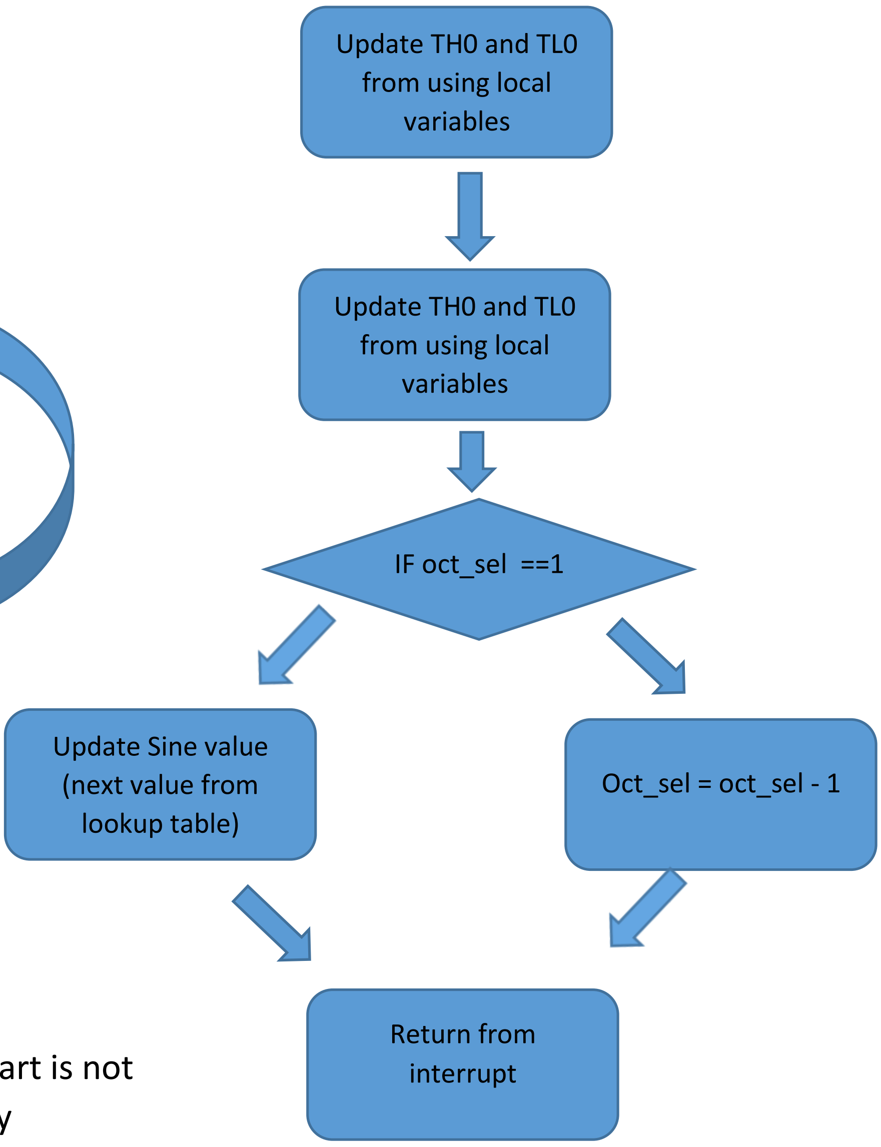
3. DAC requires 2 byte data, higher 4 bits are configuration bits which we set as 0111, and the rest 12 bits is our data.
4. Here for a corresponding frequency, we store TH0,TL0 values corresponding to T/24, as soon as TF0 flag sets, we update to next sine value in interrupt.
5. DAC is used in buffered mode, therefore it will maintain initial value until it is rewritten. Analog sine output generated by DAC is fed to speaker amplifier circuit.

Flow Chart of the Programme (Sine wave generator)

Main flowchart


START
Take input from keypad and check for debouncing
Update local variables keypad input for TH0, TL0 and octave selection

Timer 0 ISR flowchart


Update TH0 and TL0 from using local variables
Update TH0 and TL0 from using local variables
IF oct_sel ==1
Update Sine value (next value from lookup table)
Oct_sel = oct_sel - 1
Return from interrupt

NOTE:

1. Keypad debouncing flowchart is not Shown to avoid redundancy
2. Oct_sel is a variable which selects multiplier Of base time period depending upon the octave Selected from keypad

Challenges faced:

1. DAC (MCP 4921) was not able to provide enough current to get sound of audible intensity to the speaker. Therefore we had to implement an amplifier circuit to get enhanced sound
2. The sine wave generated by DAC was quantized in 24 leves (as shown in picture). Therefore it contained higher harmonics which led to noise in speaker sound. Therefore we had to implement a low pass filter to remove the noise.

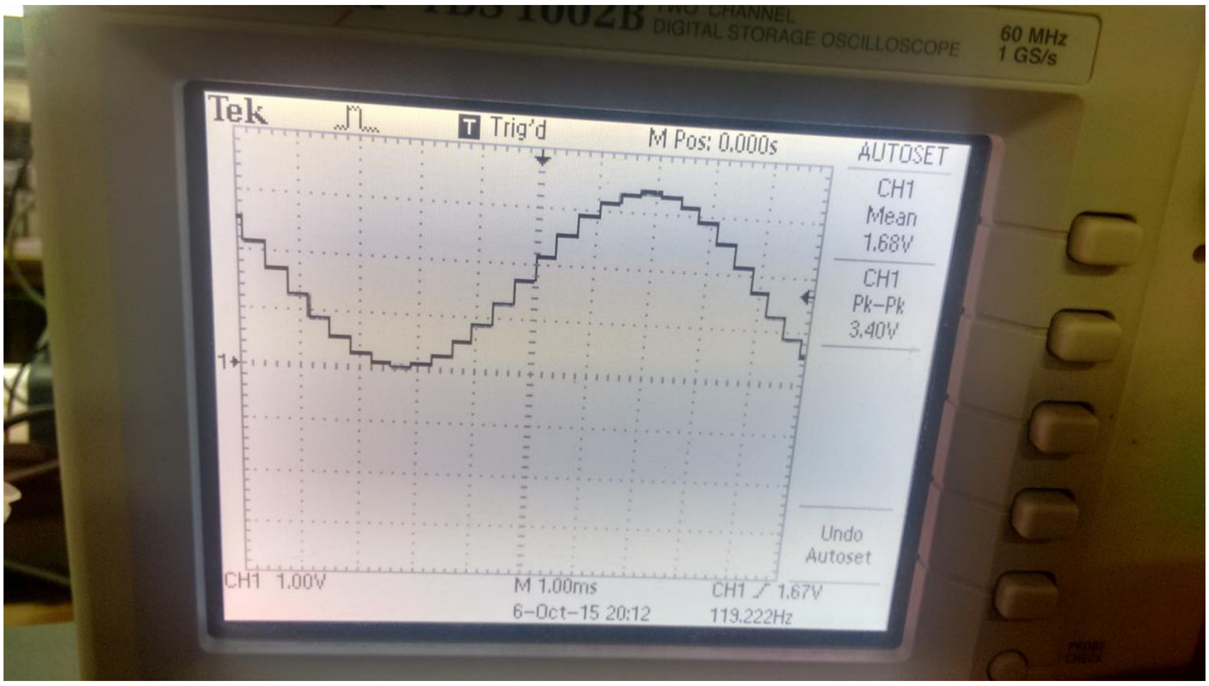
Observations and Conclusions:

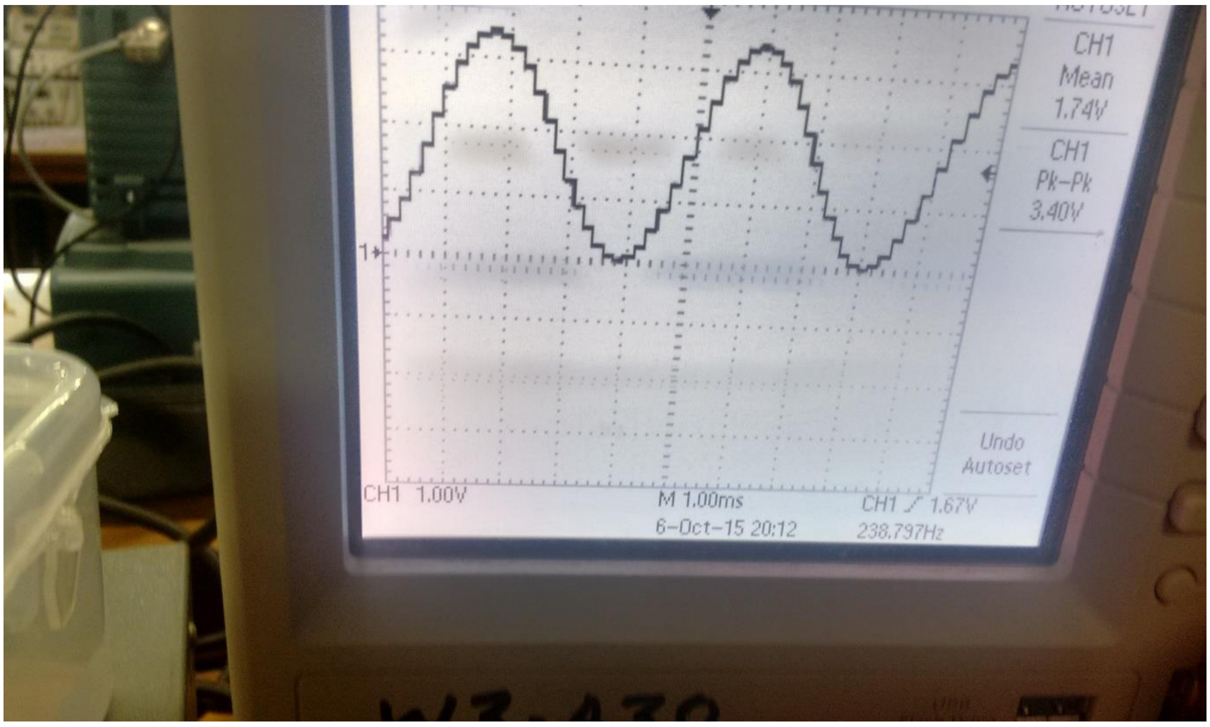
1. Square wave output faces the problem of harmonics, i.e. while the output should only be corresponding to frequency f, output corresponding to 3f,5f,7f… also come. This is the reason for not getting the exact monotonic frequency output.
2. Exactly double and quadruple of frequency comes on the output for octaves, which validates our result (shown in photos attached below).

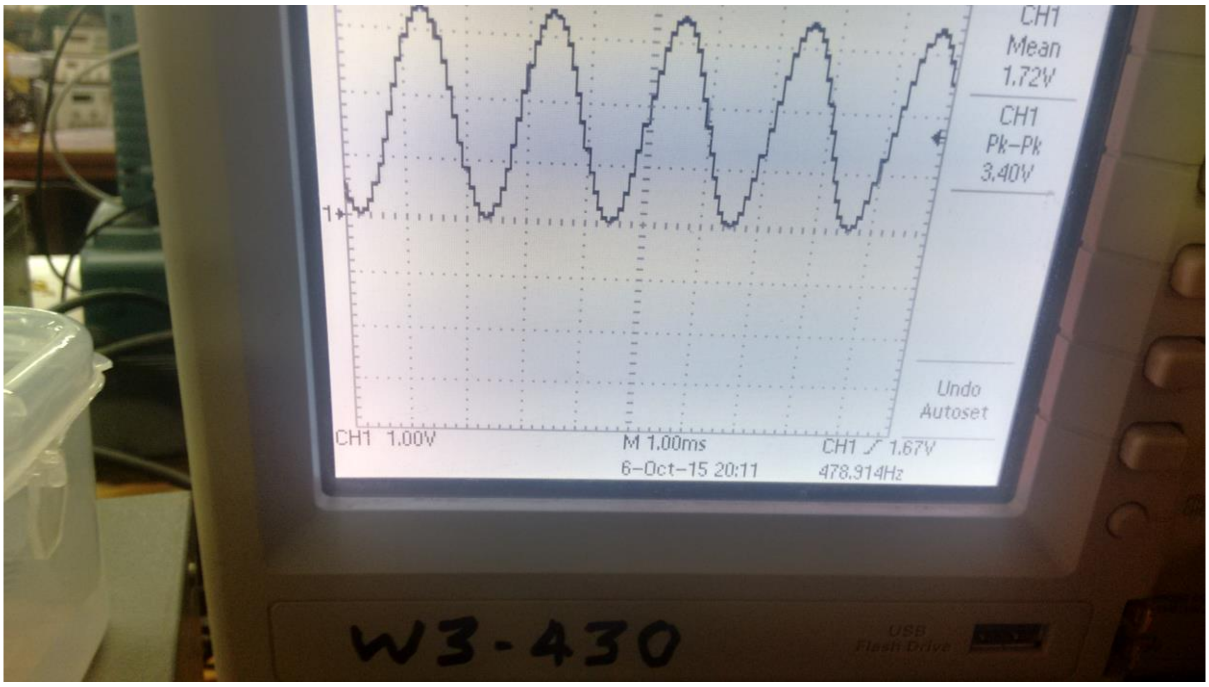
Shortcomings and Future improvements:

We can play any music by hardcoding the notes and their duration in the code which would generate desired music output using our implementation.

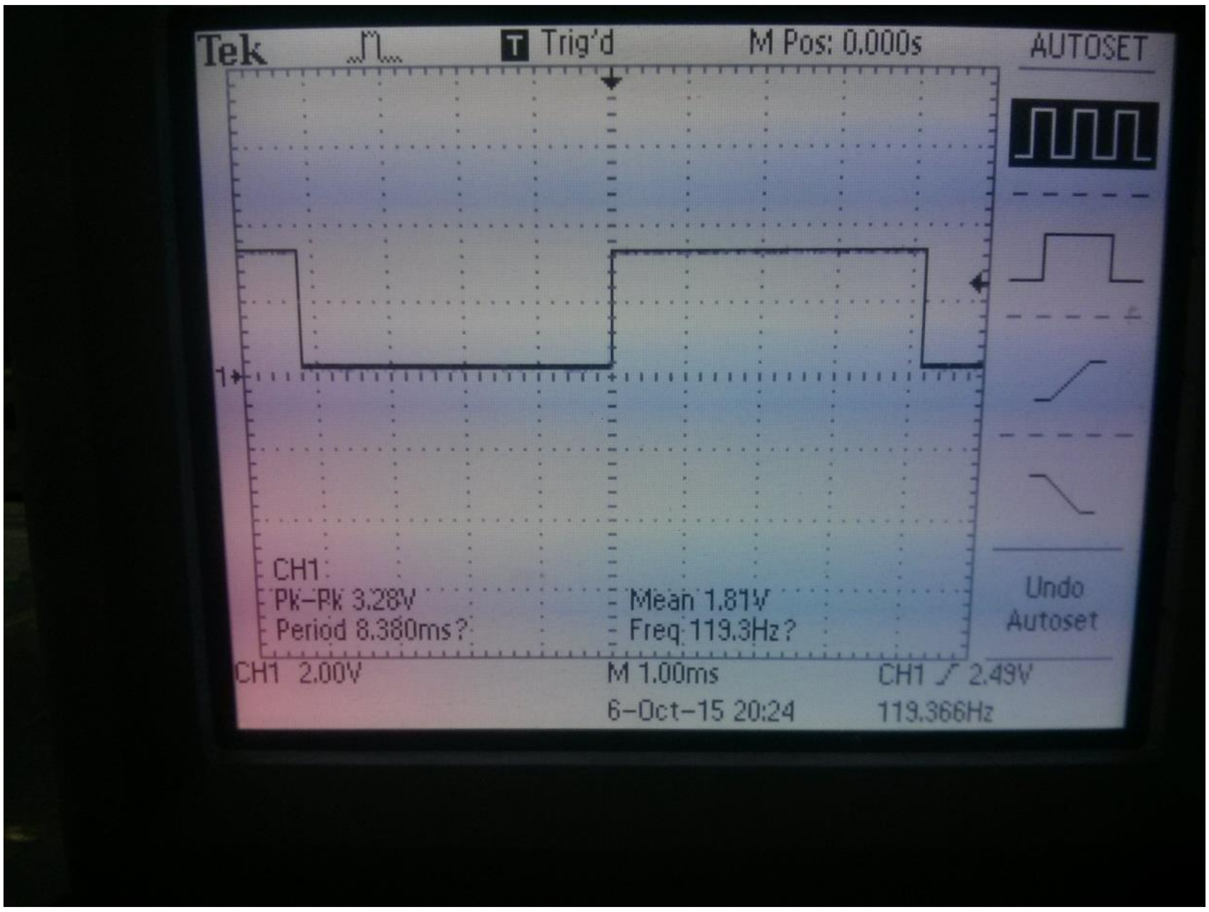
Images of sine wave for different octaves of the same note

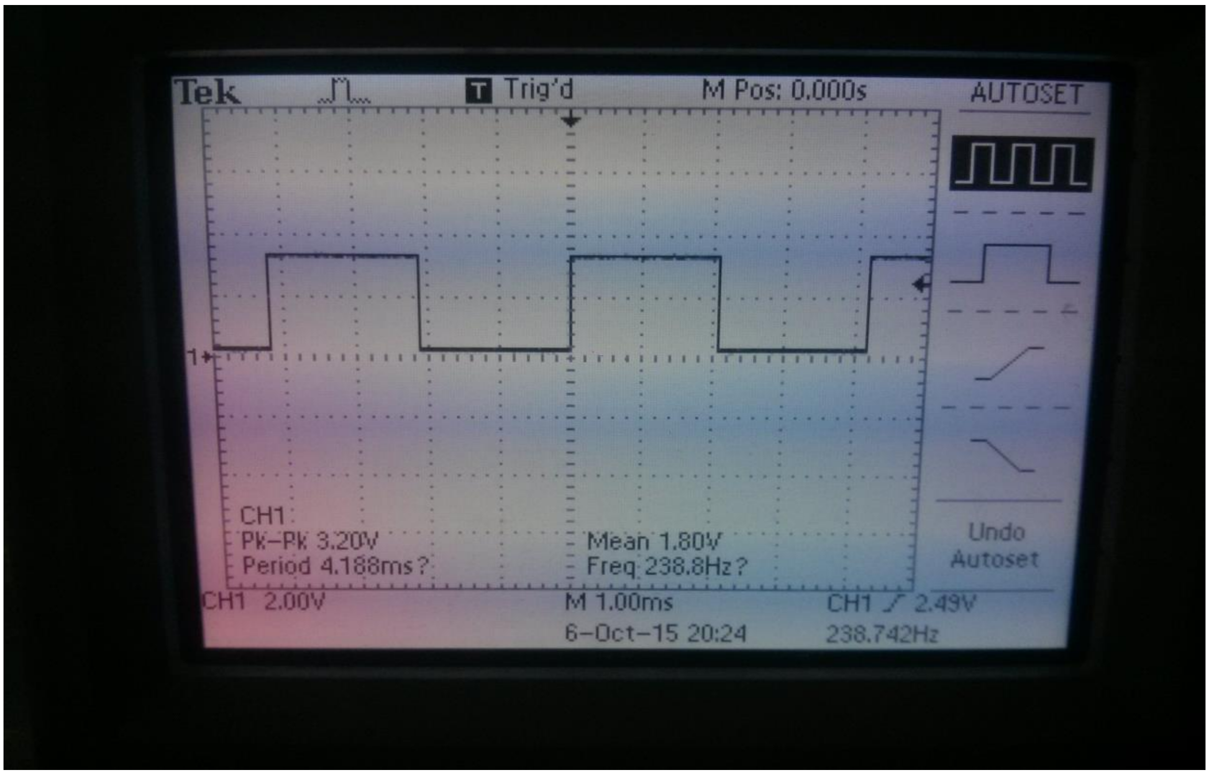


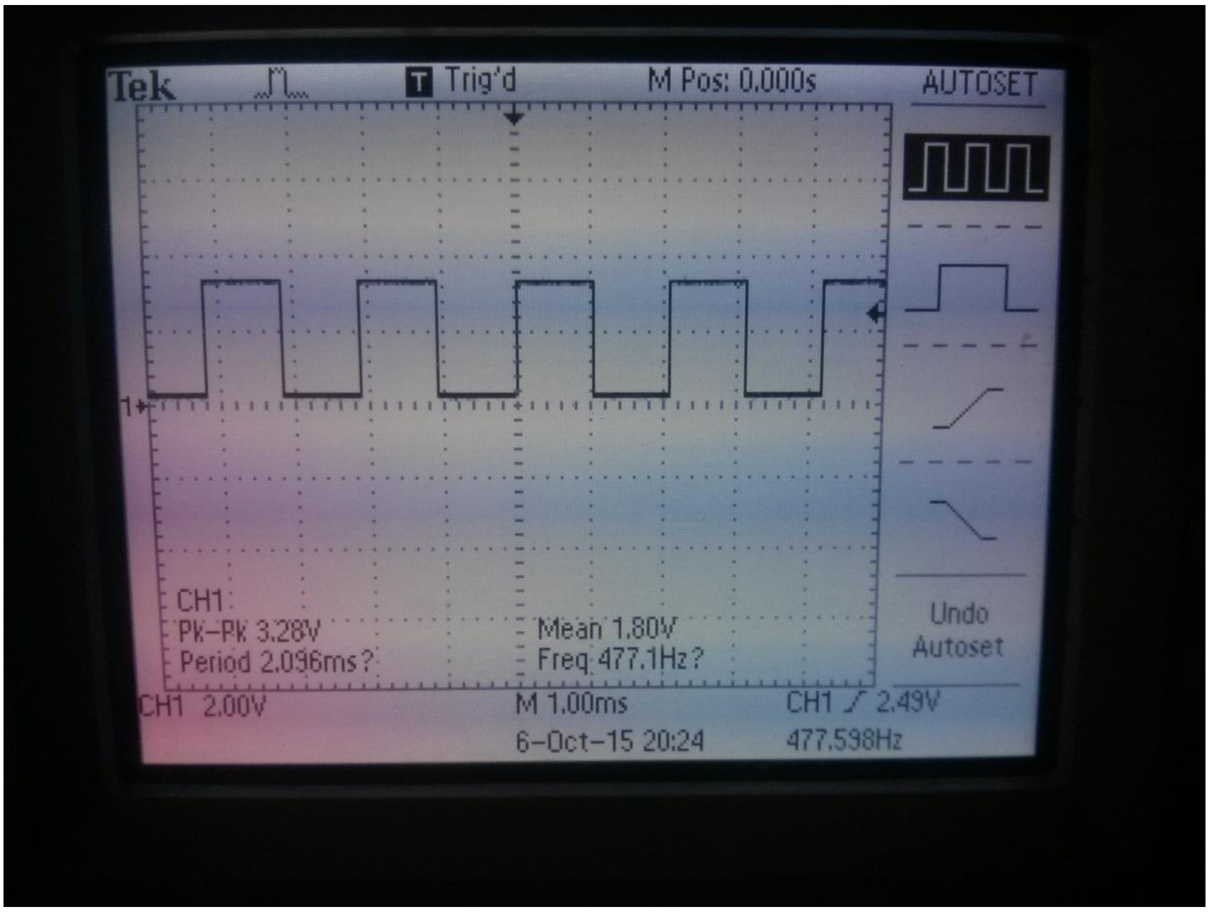

CH1
Mean
1.74V
CH1
Pk-Pk
3.40V
Undo
Autoset
CH1 1.00V
M 1.00ms
CH1 ∕ 1.67V
6-Oct-15 20:12
238.797Hz
W3-430


CH1
Mean
1.72V
CH1
Pk-Pk
3.40V
Undo
Autoset
CH1 1.00V
M 1.00ms
CH1 ∕ 1.67V
6-Oct-15 20:11
478.914Hz
W3-430
USB
Flash Drive

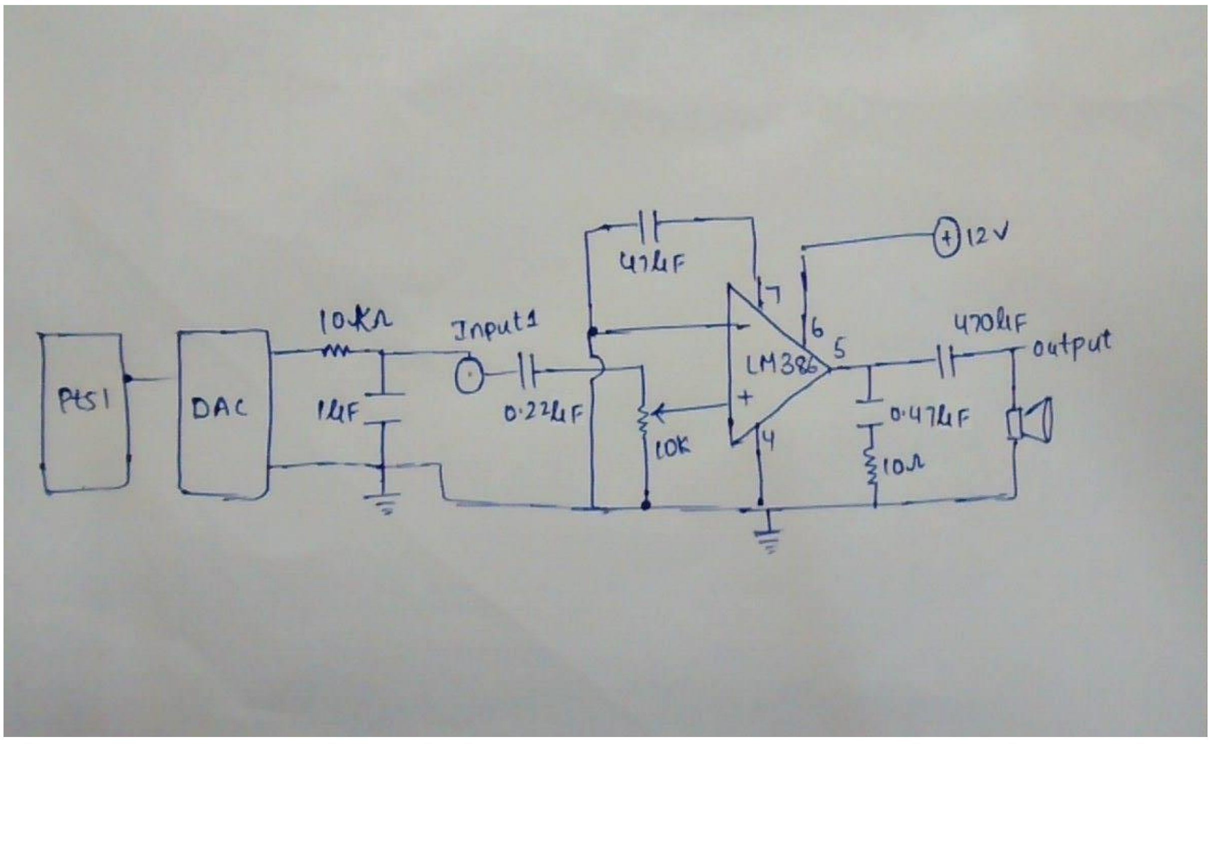
Images of square wave







Amplifier circuit



PtS1
DAC
10kΩ
1µF
Input1
0.22µF
10k
47µF
7
6
5
4
LM386
+
12V
470µF
output
0.47µF
10Ω