

Embedded Systems Lab

Exp-4

Submitted By:

Pratyush Jainwal

18EE30021

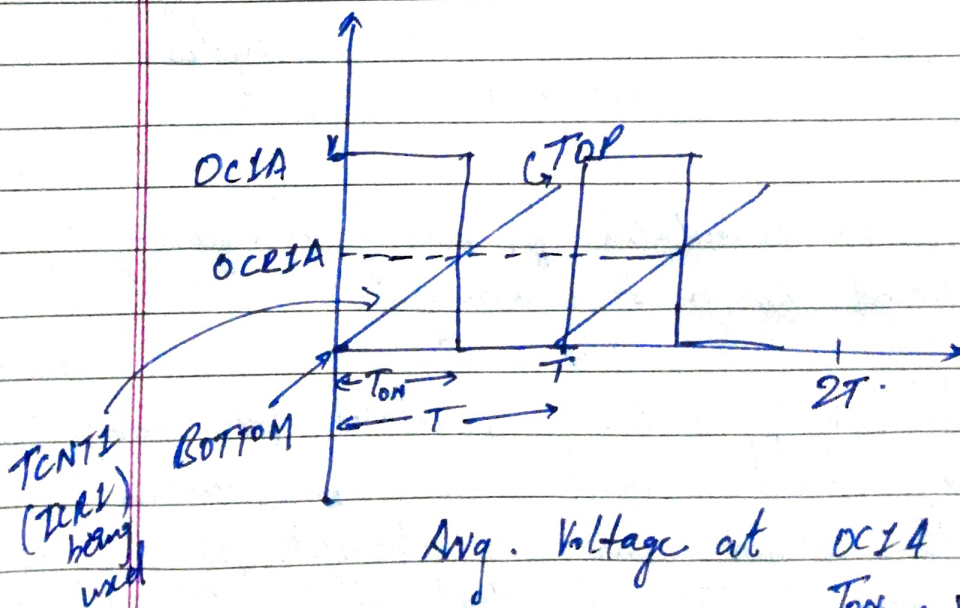
AIM: Generate Square Wave tone using Atmega 328p and its timer and interrupts.

Algorithm Used with Procedures:

FAST PWM mode was used with TOP value calculated for each frequency, using

$$= \frac{F_{cpu}}{\text{Resolution} \times \text{Freq}}$$

Here, concept (algorithm used) :



Avg. Voltage at OC1A

$$= \frac{T_{ON}}{T} \times V_0$$

Here,

when the counter reaches the TOP value, ISR is called where resolution^{count (ticks)} is increased and next value from lookup table is loaded into OCR1A for being compared.

If $idx == resolution$ then, it is reset & number of cycles occurred for current frequency (curr-cycles) is increased.

Here,

$$timep = \frac{1}{res \times freq.}$$

$$freq \text{ or } f_{arr} = \frac{1}{res \times timep.}$$

$$timep = \frac{1}{res \times f_{arr}} \text{ sec}$$

TOP value \rightarrow $timep = \frac{1}{res \times f_{arr}} \times F_{cpu} \text{ cycles}$

Here, timep is calculated for every frequency and stored in a 1-D array.

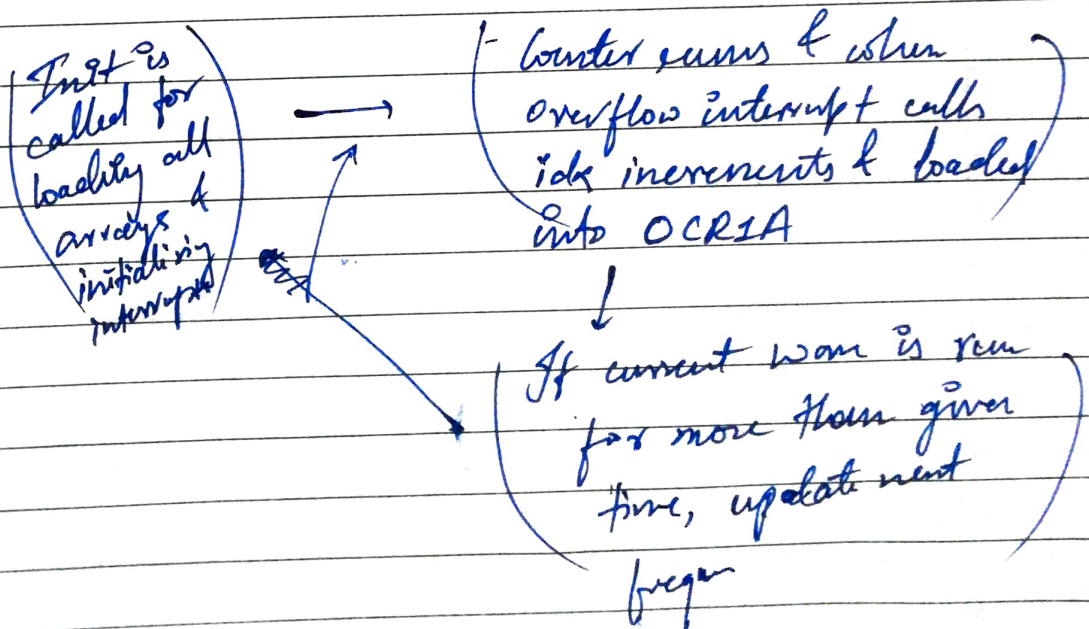
Also, the look-up table is generated for every frequency and stored in a 2-D array. But this is not needed as after all the duty ratio matters in which the ratio of

T_{ON} & T is needed & where it will be independent of time.
=

Now, for the schematic part, an active low pass filter is used for filtering and a speaker is used for tone.

Also, while simulating, make ensure that setting is as same as given in this report.

Algo Diagram:

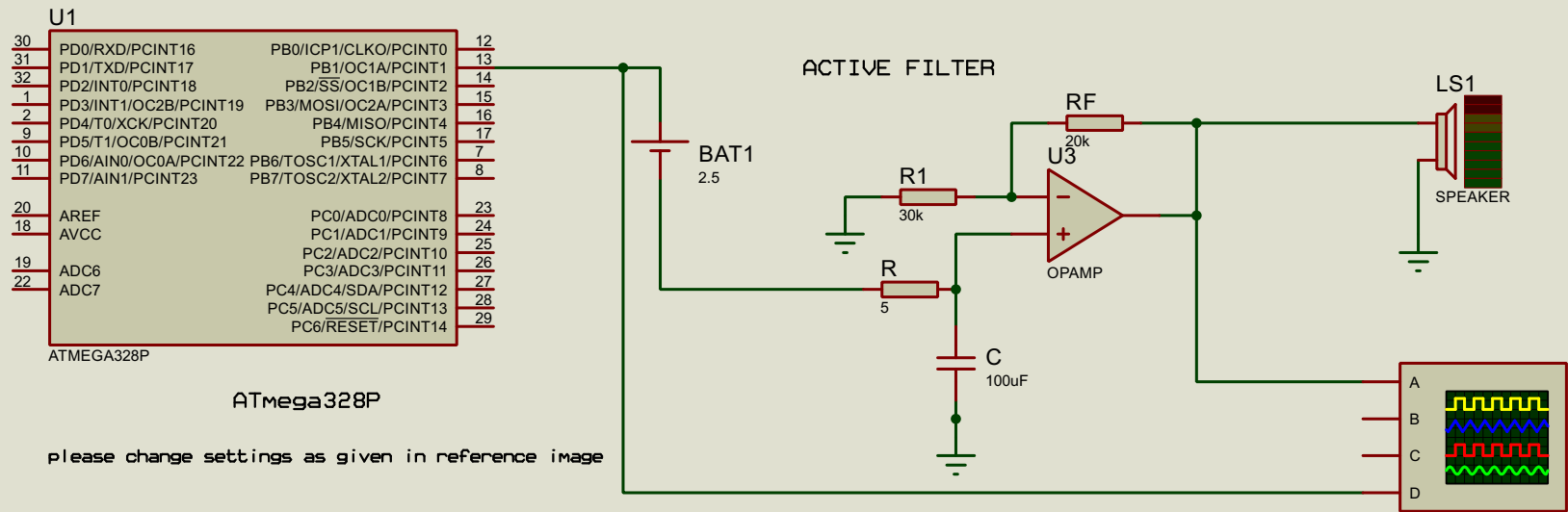


```

1  /*
2   * GccApplication1.cpp
3   *
4   * Created: 19-03-2021 00:19:01
5   * Author : Pratyush Jaiswal
6   */
7
8  #include <avr/io.h>
9  #include <util/atomic.h>
10 #include <math.h>
11 #define res 20
12 #define F_cpu 1000000UL
13 #define tot_freq_count 7
14 int frequencies[] = {240, 270, 300, 320, 360, 400, 450}; //array containing all the frequencies for generating sa re ga ma tune
15 int tot_time = 0.5; //time for which the each freq will run(here all freq are given same time,
16 //can be changed with array for modification
17 int table[tot_freq_count][res]; //2-D array containing all the look table values for every freq with resolution res
18 int timep[tot_freq_count]; //1-D array for containing pre-calculated timep values
19
20 int curr_freq = 0; //counter for maintaining the current frequency index
21
22 unsigned char idx=0; //counter for current index in lookup table
23 int curr_cycles=0; //counter for number of cycles of the current freq
24
25 void init(){
26     for(int j=0;j<tot_freq_count;j++){
27         int f_arr = frequencies[j];
28         timep[j]=F_cpu/(res*f_arr);
29         for(int i=0;i<res;i++){
30             table[j][i]=(0.5+0.5*sin((2*M_PI*i)/res))*timep[j];
31         }
32     }
33
34     TCCR1A = 0b10000010;
35     TCCR1B = 0b00011001;
36     TIMSK1=1;
37     ICR1=timep[0];
38     DDRB = 0b00000010;
39 }
40 int main(void)
41 {
42     /* Replace with your application code */
43     init();
44     sei();
45 }

```

```
46     while (1){
47         if(curr_cycles>=0.5*frequencies[curr_freq]){
48             curr_freq+=1;
49             curr_cycles=0;
50             idx = 0;
51             if(curr_freq==tot_freq_count){
52                 curr_freq=0;
53             }
54             ICR1 = timep[curr_freq];
55         }
56     }
57 }
58
59 ISR(TIMER1_OVF_vect){
60     if(idx==res){
61         idx=0;
62         curr_cycles+=1;
63     }
64
65     OCR1A=table[curr_freq][idx];
66     idx++;
67 }
```



18EE30021

Pratyush Jaiswal



Edit Component



Part Rference:

U1

Hidden: ☐

Part Value:

ATMEGA328P

Hidden: ☐

Element:

New

PCB Package:

QFP80P900X900X120-32



Hide All



Program File:

GccApplication1\GccApplication



Hide All



CLKDIV8 (Divide clock by 8)

(0) Programmed



Hide All



CKOUT (Clock output)

(1) Unprogrammed



Hide All



RSTDISBL (External reset disable)

(1) Unprogrammed



Hide All



WDTON (Watchdog Timer Always On)

(1) Unprogrammed



Hide All



BOOTRST (Select reset vector)

(1) Unprogrammed



Hide All



CKSEL Fuses:

(0010) Int. RC Osc. 8MHz



Hide All



Boot Loader Size:

(00) 1024 words. Starts at 0x1C0



Hide All



SUT Fuses:

(10)



Hide All



Advanced Properties:

Clock Frequency



(Default)

Hide All



Other Properties:

☐ Exclude from Simulation

☐ Attach hierarchy module

☐ Exclude from PCB Layout

☐ Hide common pins

☐ Exclude from Current Variant

☐ Edit all properties as text

OK

Help

Data

Hidden Pins

Edit Firmware

Cancel

Discussion:

- Due to some initial mismatch in Ecp4, the waves were generated with some noise, then for 1MHz, the simulation started to work without any noise.
- Full bits were enabled in the already existing setting of Proteus.
- Also less number of codes were put into ISR-OVF. func to prevent the complexity.