এ. এস. এম. আশরাফুল হক (রিপন)

পিসিতে শব্দ সৃষ্টির প্রোগ্রাম

কমপিউটারের সামনে বসে গেম খেলছে পিংকি। গেমটি কমপিউটারের মাধ্যমে খেলতে হয়। মাউস এবং কিবোর্ডের মাধ্যমে চমৎকার খেলে যাচ্ছে সে। গেমটির নাম হচ্ছে Wolf3D; হিটলারের বাহিনীর বিরুদ্ধে যুদ্ধ-যুদ্ধ খেলা। প্রতিটি পদে নতুন রোমাঞ্চের আবির্ভাব হচ্ছে। রাইফেলের গুলি, ব্রাশফায়ার-এর শব্দে মুখরিত হচ্ছে চারিদিক, বিভিন্ন বোনাস নেবার পরে সুন্দর মিউজিক হচ্ছে, নতুন নতুন লেভেলে নতুন উদ্দীপনার সৃষ্টি হচ্ছে। হঠাৎ গুলির শব্দ শুনলে পিংকি বুঝতে পারে যে, পেছনে বা আশে-পাশে কেউ তাকে গুলি করছে। তাৎক্ষণাৎ সে তার আর্মি কম্যাণ্ডোকে ঘুরিয়ে ঐ আততায়ীকে হত্যা করে যাচ্ছে।

কমপিউটারে ছোট একটা স্পিকার থাকে। এটি সাধারণ অবস্থায় 8Ω/0.5W এর হয়ে থাকে। এর মাধ্যমে সকল প্রকার শব্দ বের হয়। এই স্পিকার অংশটি না থাকলে পিংকির পক্ষে Wolf3D খেলতে যেয়ে আত্মগোপনকারীর আগমনধ্বনি জানা যেতো না। ফলশ্রুতিতে, তার মৃত্যু ছিলো অবধারিত। আমাদের আজকের বিষয় পিংকিকে নিয়ে নয়, স্বাভাবিকভাবেই কমপিউটারে শব্দ নিয়ে আলোচনা করবো আজ।

প্রোগ্রামে সাউণ্ডের ভূমিকা অনেক বেশি। Intiger বসাতে যেয়ে ক্যারেক্টার লিখে ফেললাম ভুলে, কমপিউটার তৎক্ষণাৎ একটা 'বিপ' শব্দ করে জানালো ভুল ইনপুট হয়েছে। বিশেষ ক্ষতিকর কাজের পূর্বে ব্যবহারকারীকে বারবার সাউণ্ডের মাধ্যমে সঠিক সিদ্ধান্তে আসতে বলা হয়। এসব শব্দ তৈরি করা হয় প্রোগ্রামের সুন্দর ব্যবহারের স্বার্থে। শব্দ দ্বারা শ্রুতি-মাধুর্য্য এবং দৃষ্টি আকর্ষণের প্রচেষ্টা শুধু এযুগে নয়, কমপিউটারের প্রথম আবিষ্কারের সময় থেকে চলে এসেছে। ইচ্ছা করলে পুরো কী-বোর্ডটাকে একটা পিয়ানোতে পরিণত করা সম্ভব। কিংবা ইলেকট্রিক ড্রাম বানানো যেতে পারে। এসবই সম্ভব যদি একটি স্পিকার এবং তার সঠিক প্রোগ্রামের পরিবেশ থাকে।

বিভিন্ন ফ্রিকোয়েন্সিতে লাউড স্পিকার কন্ট্রোলার দ্বারা যে bottleneck ডিভাইসকে বোঝায়, তাকে আমরা প্রোগ্রামেবল টাইমার বলতে পারি। টাইমারের শুরুর মানকে নির্দিষ্ট করলে পরে, তা সিস্টেমের ঘড়ি অনুযায়ী শুরুর মানের সাথে সাথে কাউন্ট নাম্বারের বিয়োজন ঘটায়। এই কাউন্ট নাম্বার বিয়োগ হয়ে '0' হলে টাইমার তৎক্ষণাৎ লাউড-স্পিকারে একটি পাল্স প্রদান করে। একই সময়ে নতুন করে আবার শুরুর মান থেকে বিয়োগ হতে থাকবে। অন্যভাবে বলা যেতে পারে, টাইমার তার চাহিদামত ফ্রিকোয়েন্সি অনুযায়ী লাউড স্পিকারকে ইমপাল্স প্রদান করতে সক্ষম, এভাবেই বিভিন্ন ফ্রিকোয়েন্সির শব্দ শোনা সম্ভব।

যদিও বিভিন্ন পিসি সিরিজের মধ্যে কম্প্যাটিবিলিটির প্রশ্ন আসে, তবু এই লাউড স্পিকারের কনট্রোলার দ্বারা শব্দ তৈরি করার ইনটারফেস সবই এক বলা যায়, সাধারণত পিসিতে টাইমারটির চিপ্স নাম্বার হলো 8253, পিসি/এটি তে 8254 নামের অন্য একটি টাইমার ব্যবহৃত হয়। এই 8253 চিপ্সে তিনটি স্বাধীন কাউন্টার আছে।

একটি কাউন্টার দ্বারা লাউড-স্পিকারকে কন্ট্রোল করা হয়।

অন্য দু'টি দ্বারা যথাক্রমে সিস্টেম এর ঘড়ি এর ডাইনামিক ষ্টোরেজকে কন্ট্রোল করা হয়।

টাইমারের ইনপুট-ফ্রিকোয়েন্সিকে কনস্ট্যান্ট বা ধ্রুব হিসেবে 1193180 করা হয়েছে। টাইমারের কাউন্টার দ্বারা কাউন্ট নাম্বার যদি '0' ফ্রিকোয়েন্সিতে পাওয়া যায়, তবে কাউন্টার তার শুরুর মান নির্দিষ্ট করেছে বুঝতে হবে। এই শুরুর মানের সাথে লাউড-স্পিকারে প্রদত্ত পাল্সএর সময়ের একটা সম্পর্ক আছে–

Frequency = 1193180 / starting count value . . . . . (i)

উপরোক্ত সূত্রকে আমরা একটু অদল-বদল করে নিলে বুঝতে আরো সুবিধা হতে পারে–

starting count value = 1193180 / frequency . . (ii)

উদাহরণস্বরূপ 10KHz এর ফ্রিকোয়েন্সিতে 119 কাউন্ট ভ্যালু এবং 100Hz Frequency তে পাবো 11932 পরিমাণ।

টাইমার দ্বারা ইনপুট-আউটপুট ইন্টারফেস এর জন্যে 40H~43H এড্রেসের মধ্যে কাজ করা যায়। নিম্নে টাইমার দ্বারা ষ্টারটিং কাউন্ট ভ্যালু নির্দিষ্ট করা ধাপগুলো লক্ষ্য করা যাক–

(১) 43H এ আউটপুট 0B6H ঃ টাইমার তিন নম্বর কাউন্টার (লাউড স্পিকার এর কাউন্টার এর কন্ট্রোলার) এর ষ্টার্টিং ভ্যালুকে নির্দিষ্ট করবে। এখানে বাইনারী নাম্বারে কাজ হবে, Low 8-bit সামনে, high 8-bit হবে পেছনে।

(২) 42H-এ আউটপুট হবে ষ্টার্টিং কাউন্ট ভ্যালু এর Low 8-bit।

(৩) 42H-এ আউটপুট হবে ষ্টার্টিং কাউন্ট ভ্যালু এর High 8-bit

উপরের তিনটি ধাপ সম্পন্ন করে টাইমার তার ফ্রিকোয়েন্সি নিয়ে ব্যস্ত হয়ে পড়ে। আবার নতুন করে পূর্বের কাউন্ট ভ্যালুকে নির্দিষ্ট করে অথবা পাওয়ার সাপ্লাইকে বন্ধ করে দেয়। টাইমার আবার লাউড-স্পিকার দ্বারা তৈরি শব্দের ফ্রিকোয়েন্সিকে কন্ট্রোল করতে পারে। এক্ষেত্রে দুটো আউটপুট সিগন্যাল পাওয়া যায়; তার 61H এর Low 2-bit এড্রেসে অবস্থান করে। এই 2-bit কে 1-bitএ আনলে টাইমার লাউড স্পিকারে পাল্স প্রদান করে ফ্রিকোয়েন্সি অনুযায়ী শব্দ তৈরি করতে পারে। এভাবে স্পিকারকে ওপেন করা হয়। পাল্স স্পিকারে না পৌছালে closed থাকবে। এখানে উল্লেখ্য যে, 61H এর High-8 bit অন্য ক্ষেত্রে ব্যবহৃত হয়, তাই Low 2-bit বদলানোর সময় ঐ 6-bit কে বদলানো যাবে না।

এবার আমরা বিভিন্ন ল্যাংগুয়েজে সাউন্ড নিয়ে প্রোগ্রাম পদ্ধতির ব্যাপারে আসি। প্রথমে C ল্যাংগুয়েজ নিয়ে কিছু বলা যাক–

(ক) **TURBO C** ঃ টারবো সি তে শব্দ কন্ট্রোলের জন্য যথারীতি বেশ কটি ফাংশন রয়েছে। যেমন : SOUND ( ), NOSOUND ( ), DELAY ( ) ইত্যাদি। শব্দ তৈরি করার জন্যে SOUND ফাংশনকে ব্যবহার করা হয়। যেমন ঃ SOUND (100) ; এভাবে শব্দ তৈরি করা যাবে। Sound ফাংশনের Source কোড এরকমঃ

```
Void  SOUND (unsigned frequency)
{
    union {
            nnsigned divisor;
            unsigned char c[2];
            } tone;
  if (frequency <19) return
  tone. divisor = 1193180L /'frequency;
  outport b (0X43, 0xB6);
  outport b (0x42, tone. c[0]);
  outport b(0x42,tone.c[1]);
  outport b(0x61, inport b(0x61):3);
}
```

উপরের ফাংশনদ্বারা সর্বোচ্চ 65535Hz ফ্রিকোয়েন্সি ব্যবহার করা যেতে পারে। সাধারণ অবস্থায় এই সর্বোচ্চ frequency 12000 Hz হয়ে থাকে। এখন কথা হলো সাউন্ড তৈরি করা হলো, কিন্তু একে না থামালে কমপিউটার সারক্ষণ ঐ শব্দ করে যাবে। ব্যাপারটা খুব সহজ। 61H এর Low 2-bit কে '0' করে দিলেই হলো। টারবো সি তে Nosound নামে এই কাজের জন্য ফাংশন আছে। তার সোর্স কোড দেয়া হলো–

```
void NOSOUND (  void)
{
    outportb ( 0x61, inportb(0x61) & 0xFC);
}
```

শব্দ তৈরি করা হলো, শব্দ বন্ধ করার পদ্ধতিও পেলাম। কিন্তু আমরা জানিনা, কখন প্রথম শব্দের পরে তাকে বন্ধ করতে হবে। একাজে টারবো সি তে Delay নামক ফাংশনকে ব্যবহার করা হয়। তার ফরমেট নিম্নরূপ–

```
void delay ( unsigned milliseconds)
```

Delay-এর মাধ্যমে শব্দ তৈরির পরে NOSOUD ফাংশনকে ব্যবহার করার সময়কে Delay করা হয়। এভাবে শব্দের শ্রুতিমাধুর্য বহুলাংশে বৃদ্ধি পায়। নিচের প্রোগ্রামটি দেখা যাক–

```
/ **********************/
/* NAME : SOUND1.C*/
/* RESULT : LOOP, Frequency based  sound for 1 second */
/**********************/
  # include < stdio.h?
  # include <dos.h>
  main ( )
  {
     unsigned int freq;
     do {
          print ("Enter Frequency:");
          scan f ("%d', & freq);
          if(freq){
                    sound (freq);
                    delay (1000);
                    nosound ( );
```

```
    }
    } while (freq);    /*যদি frequency '0' না হয় */
}
```

এভাবে উপরের প্রোগ্রামটির মাধ্যমে Frequency '0' না লেখা পর্যন্ত তার হরেক ফ্রিকোয়েন্সিতে শব্দ হতে থাকবে। একটা লুপ এর মধ্যে চলবে।

পরবর্তী প্রোগ্রামটিতে তিনটি ব্যাতিক্রমধর্মী শব্দ তৈরি করা হলো। প্রোগ্রামটি কম্পাইল করে EXE ফর্মে তৈরি করে বা সরাসরি TC এর এডিটর থেকে CTRL+F9 দ্বারা রান করে দেখা যেতে পারে। রান করলে শূন্য থেকে তিন (0—3) এর মধ্যে যে কোন একটি সংখ্যা পছন্দ করতে বলা হবে। 0 দ্বারা প্রোগ্রামকে টারমিনেট এবং ১—৩ দ্বারা তিন রকম শব্দ পাওয়া যাবে।

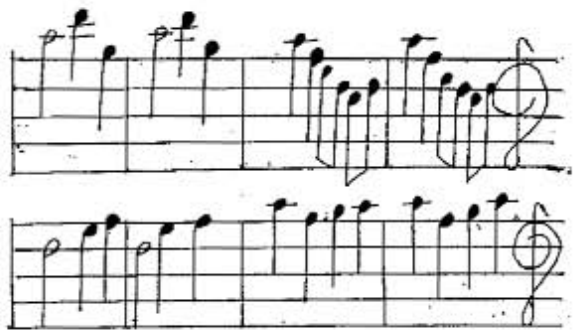
```
/* ********************  */
/* NAME : SOUND2.C       */
/* OUTPUT : 3 KINDS OF MUSIC */
/* ********************  */
# include <stdio.h>
# include <dos.h>
main ( )
{
        int snd;
        int cnt;
        int note;
            while (1)|
                        nosound ( );
                        printf("1-1st Music,
                                2-2nd Music,
                                3-3rd Music,
                                0-EXIT \n");
                        Scanf("%d", & snd);
                        if (snd<0 :: snd>3)
                            |
                            printf ("Input wrong!\n");
                            continue;
                            }
                        if(snd = = 0) break;
                        printf ("Duplicate number :");
                        scanf ("%d", & cnt);
    while(cnt--)|
          switch (snd){
                case 1:
                      for(note=100; note<1000; note+=10){
                                sound (note);
                                delay (4);
                                }
                      for (;note>100; note-=10){
                                sound (note);
                                delay (4);
                                }
                                break;
                case 2 :
                      for(note = 100; note <2000; note+=100)|
                                sound (onte);
                                delay (12);
                                }
                      for (;note >100; note-=10){
                                sound (note)
                                delay (12);
                                }
                                break;
                    case 3:
                      for(note=4000; note>10;
                                note-=10)
                                {sound (note);
                                delay (15);
                                }
                                break;
                            }
                        }
            |
}
```

এবার একটি গানের সুর তোলার চেষ্টা করা যাক। অনেকেই হয়তো 'Two tigers' গানটি শুনে থাকবেন। আমরা তার একটি প্রোগ্রাম করার চেষ্টা করি। গানের সুর তোলার আগে 'সারগামাপাধানি' বা 'CDEFGABC' সম্পর্কে ধারণা থাকতে হবে। নিচের টেবিলটি লক্ষ্য করি—

Phonetic Symbol	c	d	e	f	g	a	b	c'	d'	e'	f'	g'	a'	b'	c'	
IN NUMBER	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	
Frequency	131	147	165	175	196	220	247	262	294	330	344	392	440	494	523	

'Two tigers' এর Node গুলো নিচে দেয়া হলো—

1 = c 4/4

1 2 3 1 56 54 31 | 1 2 3 1 54 54 31 | 345— 251— | 345— 251—

ঐ নোডগুলো ব্যবহার করে এবার তার গানের জন্যে সোর্স কোড তৈরি করা যাক—

```
/* * ***************      */
/* NAME : SOUND3. C       */
/* OUTPUT : SING 'TWO TIGERS' */
/* ***************       */
# include <dosh.h>
main ( )
    {
      unsigned signe [ ] = {  262, 250, 294, 250, 330, 250,
                              262, 250.\
                              262, 250, 294, 250, 330, 250.
                              262, 250.\
                              320, 250, 394, 250, 392, 500,\
                              320, 250, 349, 250, 392, 500,\
                              392, 125, 440, 125, 392, 125,
                              339, 125, 330, 250, 262, 250,\
                              392, 125, 440, 125, 392, 125.
                              339, 125, 330, 250, 262, 250.\
                              294, 250, 196, 250, 262, 500,\
                              294, 250, 196, 250, 262, 500,\
                              0,0};
      unsigned   int *p;
            p=sing;
            while (*p)|
                        sound (*p++);
                        delay (*p++);
                        nosound ( );
                      }
    |
```

এখানে (*P++) এর স্থানে (2*(*p++)); দ্বারা delay ফাংশনকে বদলিয়ে নতুন outpu পেতে পারেন। সাধারণ পিসিতে আমরা 20~18000Hz সীমার মধ্যে শব্দ শুনে থাকি। অনেক সময় সাউন্ড সঠিক না হওয়াতে ঐ রেঞ্জ প্রায় 12000~14000Hz Frequency ধরতে পারে। বহুল ব্যবহৃত সীমা হচ্ছে 100~5000 Hz. টারবো সি তে আমরা inportb( ) এবং outportb( ) ফাংশনদ্বয় ব্যবহার করেছি। এরা Microsoft C (MSC) তে এক নয়। (চলবে)