

# Code Composer Programming

## Objective

We want to write a code which sends a complete wave cycle whenever there is one in the data.

### Code#1:

```
// adapted from Rulph chassing book
// NUCES FAST ISB
#include "dsk6713_aic23.h" // support file for codec,DSK
Uint32 fs = DSK6713_AIC23_FREQ_8KHZ; // set sampling rate
volatile short loop = 0; // table index
short gain = 1; // gain factor
short sine_table[8]={707,1000,707,0,-707,-1000,-707,0}; // sine values
short dataindex=0;
short data[6]={1,0,1,0,1,0};
interrupt void c_int11() // interrupt service routine
{
    if(data[dataindex]==1)
    {
        output_sample(sine_table[loop]*gain); // output for on-time sec
        if (loop < 7)
            ++loop; // check for end of table
        else
        {
            loop=0;
            if(dataindex<5)
                dataindex++;
            else
                dataindex=0;
        }; // reinit loop index
        return;
    }

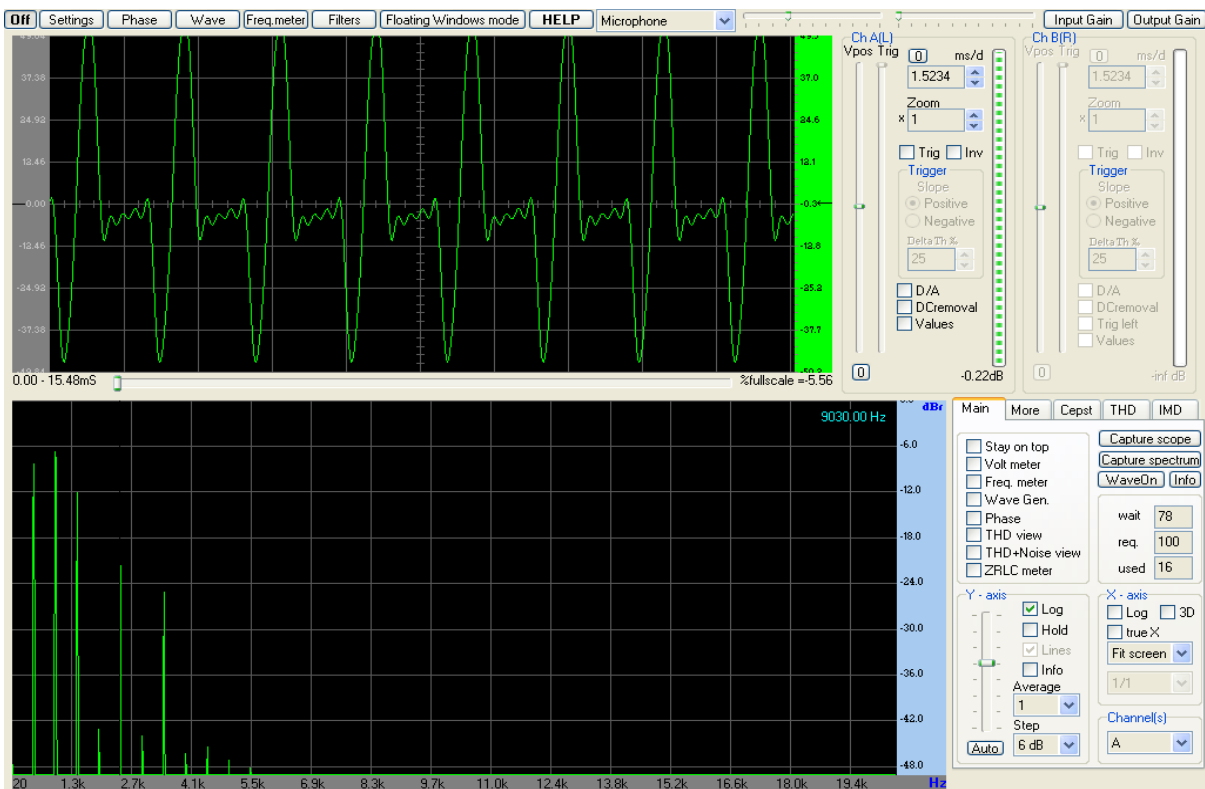
    if(data[dataindex]==0)
    {
        output_sample(0); // output for on-time sec
        if (loop < 7)
            ++loop; // check for end of table
```

```

else
{
    loop=0;
    if(dataindex<5)
        dataindex++;
    else
        dataindex=0;
}; // reinit loop index
return;
}
}
void main()
{
    comm_intr();
    while(1); // infinite loop
}

```

### Snapshot



### Code#2:

```

// adapted from Rulph chassing book
// NUCES FAST ISB ( Munzir Ahmad)

```

```
#include "dsk6713_aic23.h"          // support file for codec,DSK

Uint32 fs = DSK6713_AIC23_FREQ_8KHZ; // set sampling rate

volatile short loop = 0;          // table index

short gain = 1;                   // gain factor

short sine_table[8]={0,707,1000,707,0,-707,-1000,-707}; // sine values
short sine_table2[8]={0,-707,-1000,-707,0,707,1000,707};
short data[6] = { 1, 0, 1, 0, 1, 0 };
short dataindex=0;

interrupt void c_int11()          // interrupt service routine
{
    if(data[dataindex]==1)
    {
        output_sample(sine_table[loop]*gain); // output for on-time sec

        if (loop < 7)
            ++loop;                          // check for end of table
        else
        {
            loop=0;
            if(dataindex<5)
                dataindex++;
            else dataindex=0;
        };
        // reinit loop index
        return;
    }
    if(data[dataindex]==0)
    {
        output_sample(sine_table2[loop]*gain); // output for on-time sec
        if (loop < 7)
            ++loop;                          // check for end of table
        else {
            loop=0;
            if(dataindex<5)
                dataindex++;
        }
    }
}
```

```

else
    dataindex=0;

}; // reinit loop index

return;

}

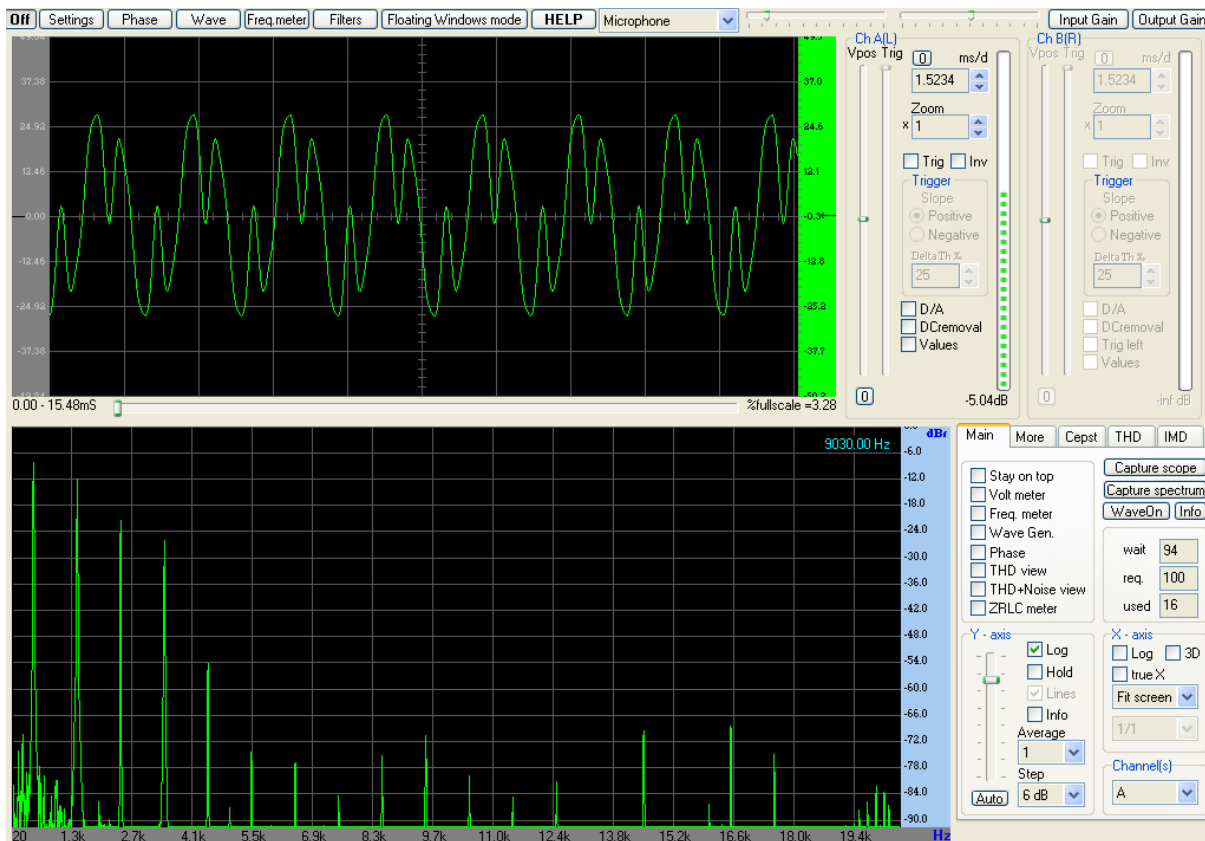
void main()
{
comm_intr();

while(1); // infinite loop

}

```

## Snapshot



## Code#3:

```

// adapted from Rulph chassing book
// NUCES FAST ISB (Munzir Ahmad)

#include "dsk6713_aic23.h" // support file for codec, DSK

uint32 fs = DSK6713_AIC23_FREQ_8KHZ; // set sampling rate

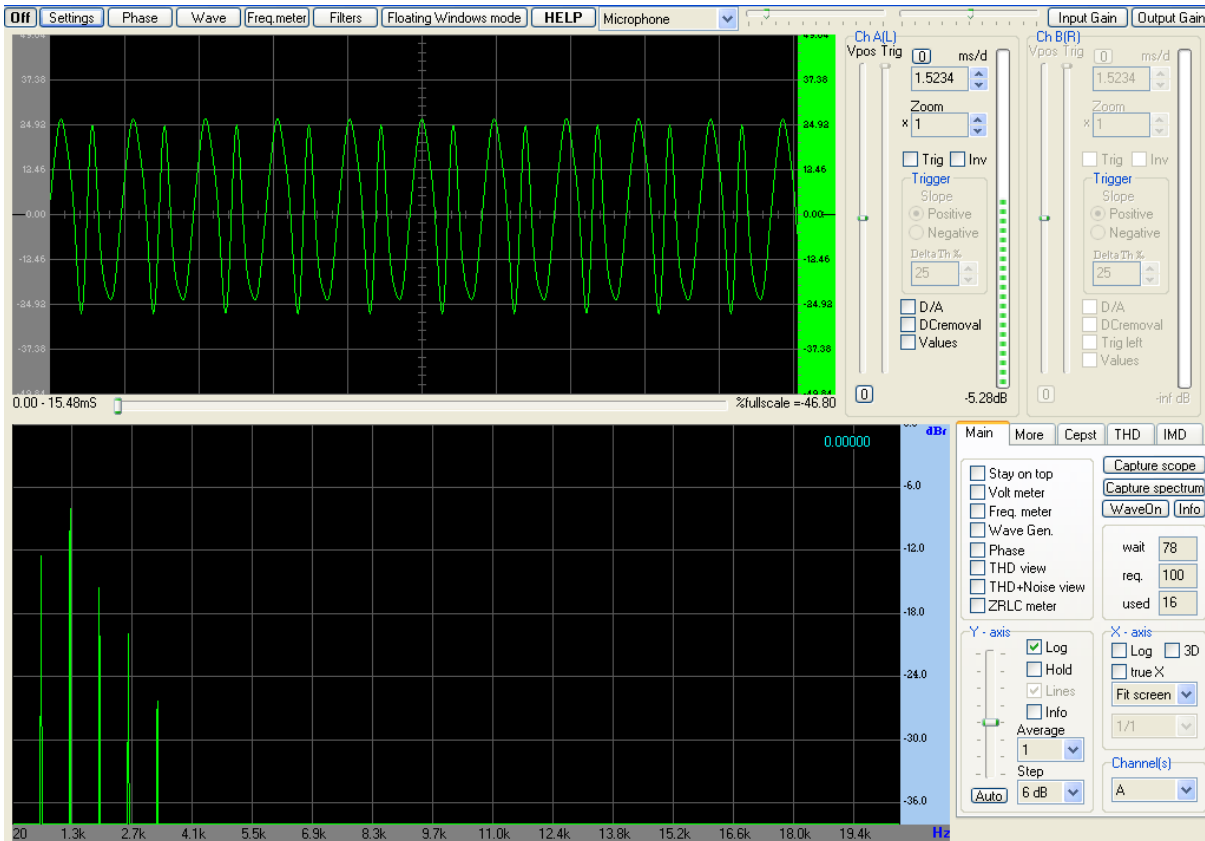
volatile short loop = 0; // table index

```

```
short gain = 1; // gain factor
short sine_table[8]={0,707,1000,707,0,-707,-1000,-707}; // sine values
short sine_table2[4]={0,1000,0,-1000};
short data[6] = { 1, 0, 1, 0, 1, 0 };
short dataindex=0;
interrupt void c_int11() // interrupt service routine
{
    if(data[dataindex]==1)
    {
        output_sample(sine_table2[loop]*gain); // output for on-time sec
        if (loop < 3)
            ++loop; // check for end of table
        else
        {
            loop=0;
            if(dataindex<5)
                dataindex++;
            else
                dataindex=0;
        }; // reinit loop index
    }
    return;
}
if(data[dataindex]==0)
{
    output_sample(sine_table[loop]*gain); // output for on-time sec
    if (loop < 7)
        ++loop; // check for end of table
    else
    {
        loop=0;
        if(dataindex<5)
            dataindex++;
        else
            dataindex=0;
    }; // reinit loop index
}
return;
```

```
    }}  
void main()  
{  
    comm_intr();  
    while(1);           // infinite loop  
}
```

### Snapshot:



### Result:

By using code composer we have written the desired code and the output can be clearly seen in the Visual Analyzer (As given by figures above).