

The Problem Statement for Q1

You must answer the following two questions to the best of your ability.

If you are not sure of the answer to a question, try to answer it anyways. It is still better to answer something (and show you tried) than nothing.

If something seems unclear or ambiguous, make some reasonable assumptions and state them in your answer. Feel free to use any documentation you want.

Question 1:

Your task will be to parse a "Sensequake .bin" data file. "Sensequake .bin" data files have the following format:

- The file starts with a text (ASCII) header followed by binary data.
- The text header is separated from the data by two newlines (more precisely: "\r\n\r\n").
- The text header contains multiple lines, the last of which (before the "\r\n\r\n") contains a list of axes separated by spaces (e.g. " Vx Vy Ax Az " means that there are 4 axes: Vx, Vy, Ax and Az).
- The data is packed binary containing multiple "frames". Each "frame" contains as many "samples" as there are axes in the aforementioned list. There is no separation (like spaces) between frames or samples.
- Each "sample" is made of 3 bytes. To decode it, you must treat these 3 bytes as the 3 most significant bytes of a 4 bytes little-endian signed integer where the least significant byte is set to 0.

In file "quesion1.cpp", implement a function which reads a "Sensequake .bin" file, prints its header, and then prints the decoded data as text with a space between samples and a newline between frames.

The code should be written in C++20 (or an earlier C/C++ standard) and, when compiled, should run on a PC.

The only library you can use is the C++ (or C) standard library.

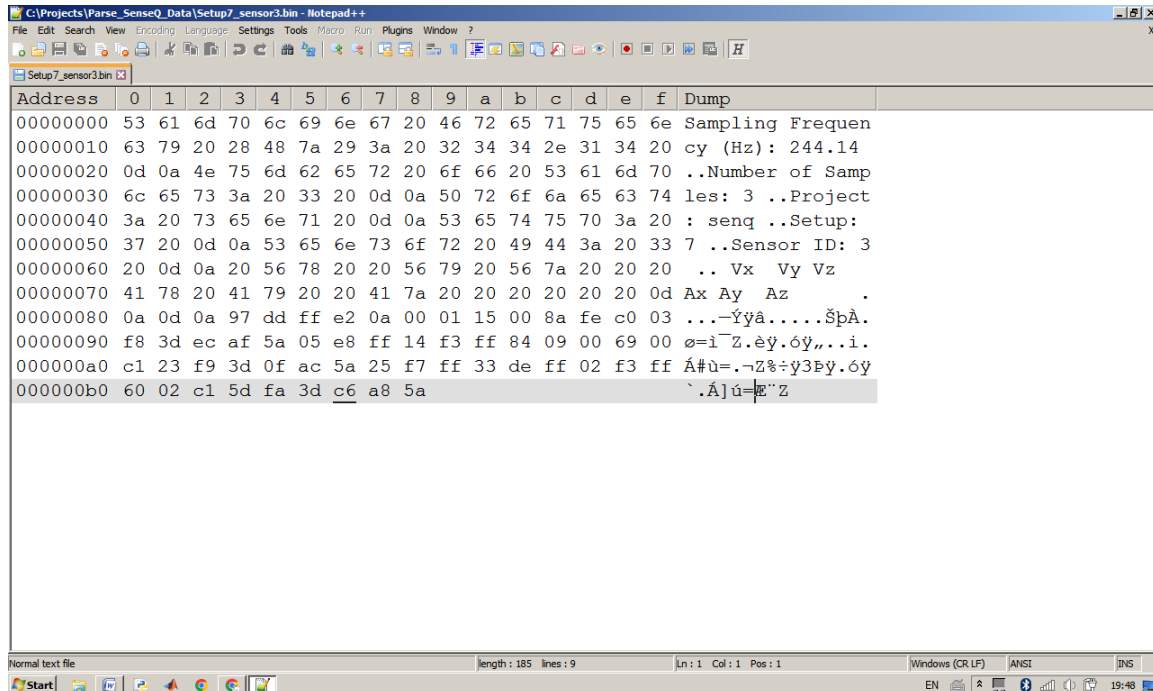
A "Setup7_sensor3.bin" file is provided as an example of a "Sensequake .bin" file for testing.

Developing the Solution for Q1

Step 1:

Analysis of the given binary file (Setup7_sensor3.bin)

Illustration of opening it in a Hex Editor Application (for e.g. Notepad++)



Observations

The total no. of bytes are 185 (0x00 to 0xb8) i.e. the data is available between an matrix of 12 rows and 16 columns < 192 bytes.

Counting the no. of bytes for the below listed headers :

1. The header named **sampling Frequency (Hz): 244.14** ranges between 0-31 bytes including 4 space characters at (8, 18, 24 and 31 each) and is followed by 0a and 0d (32 and 33 as end of line/ new line characters).
2. The header named **Number of Samples : 3** ranges between 34-54 bytes including 4space characters at (40, 43, 52, and 54 each) and is followed by 0a and 0d (55 and 56 as end of line/ new line characters).
3. The header named **Project: senq** ranges between 57-70 bytes including 2 space characters at (65 and 70 each) and is followed by 0a and 0d (71 and 72 as end of line/ new line characters).
4. The header named **Setup: 7** ranges between 73-81 bytes including 2 space characters at (79 and 81each) and is followed by 0a and 0d (82 and 83 as new line/end of line characters).
5. The header named **Sensor ID: 3** ranges between 84-96 bytes including 3 space characters at (90, 94, and 96 each) and is followed by 0a and 0d (97 and 98 as new line/end of line characters)
6. The header named **Vx Vy Vz Ax Ay Az** (axes variables) ranges between 99-120 bytes including 10 space characters at (99, 102, 103, 106, 109,110,111, 114, 117, and 118 each).
7. The header named **Vx Vy Vz Ax Ay Az** (axes variables) ends with 6 space characters ranging between (121-126) and 4 newline/end line characters ranging between (127-130).

8. The remaining bytes ranging between 131 to 184 make up of 54 bytes for and I can assume if it is for the samples.

Step 2:

As per given information:

- There are 6 axes **Vx Vy Vz Ax Ay Az**
- Each sample is made up of 3 bytes
- Therefore there should be a total of 18 bytes per frame
- Therefore, there should be a total of 3 frames of axes samples to occupy the last 54 bytes.

Step 3:

Now to solve assignment, I plan to use the fstream class in C++

I start with creating a fstream file object, and apply various functions such as seekp, tellp, read, with appropriate parameters from the above known's

Later I shall develop a function to decode the 3 sample bytes per each data frame (convert into integer value, the 3 consecutive bytes by adding a 0 LSB as the 4th byte) as mentioned in the problem statement.

Finally, I should develop my code so that it can display a final desired output.

Results:

Using above methods I was able to visualise the data from the file as bellow

Then I was able to separate the 3 byte samples and covert (decode) them into 4 bytes by adding 0x0 on the LSB of the little-endian signed integer

```

C:\Users\Shree\Desktop\Jobb\Sensequake\embedded_applicant_test_v2\Decoded_Samples.bin - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Samples.bin Decoded_Samples.bin
Address 0 1 2 3 4 5 6 7 8 9 a b c d e f Dump
00000000 00 dd ff e2 00 0a 00 01 00 15 00 8a 00 fe c0 03 .Ýÿâ.....Š.pÀ.
00000010 00 f8 3d ec 00 af 5a 05 00 e8 ff 14 00 f3 ff 84 .ø=i.¬Z..èÿ..óÿ,,
00000020 00 09 00 69 00 00 c1 23 00 f9 3d 0f 00 ac 5a 25 ...i..Á#.ù=..¬Z%
00000030 00 f7 ff 33 00 de ff 02 00 f3 ff 60 00 02 c1 5d .÷ÿ3.Ëÿ..óÿ`..Á]
00000040 00 fa 3d c6 00 a8 5a c6 .ú=Æ.¬ZÆ

```

The final solution that I was able to bring to the assignment questions was as shown in figure below

```

C:\Users\Shree\Desktop\Jobb\Sensequake\embedded_applicant_test_v2\Solution\question1.exe
The .bin file that you are searching is found and read.

The total number of bytes read in the file were 185

Header 1 (approx. 32 Bytes)---> Sampling Frequency (Hz): 244.14
Header 2 (approx. 32 Bytes)---> Number of Samples: 3
Header 3 (approx. 18 Bytes)---> Project: senq
Header 4 (approx.8 Bytes)---> Setup: 7
Header 5 (approx.13 Bytes)---> Sensor ID: 3
Header 6 (approx. 23 Bytes)---> Ux Uy Uz Ax Ay Az

The size of Frame1 is 18 Bytes. The binary data is follows:  ù! ô
08 è ¼°=ý»Z
The size of Frame2 is 18 Bytes. The binary data is as follows:  b 7% ä i ¼#''=*
%Z
The size of Frame3 is 18 Bytes The binary data is as follows:  , 3i 0% `0¹]-=ã¿Z

The six variable axes & its samples are illustrated in a table shown below:

Ux      Uy      Uz      Ax      Ay      Az
14548962 655361 1376394 16695299 16268780 11491845
<----- Newline between frames ----->
15269652 15990660 589929 49443 16334095 11295269
<----- Newline between frames ----->
16252723 14614274 15990624 180573 16399814 11033286

There are a total 18 samples found as shown in 3 rows and 6 columns as above., I
he six in first row are from the frame 1. The six in second row are from the fra
me 2. The six in third row are from the frame 3. Each sample in each frame row b
elongs to the six axes respectively. Therefore there are 3 samples per axes as s
hown in columns. The decoded binary file is also generated and saved in the fold
er. It can be viewed using HEX a viewer application such as Notepad++.

Process returned 0 (0x0)  execution time : 0.063 s
Press any key to continue.

```

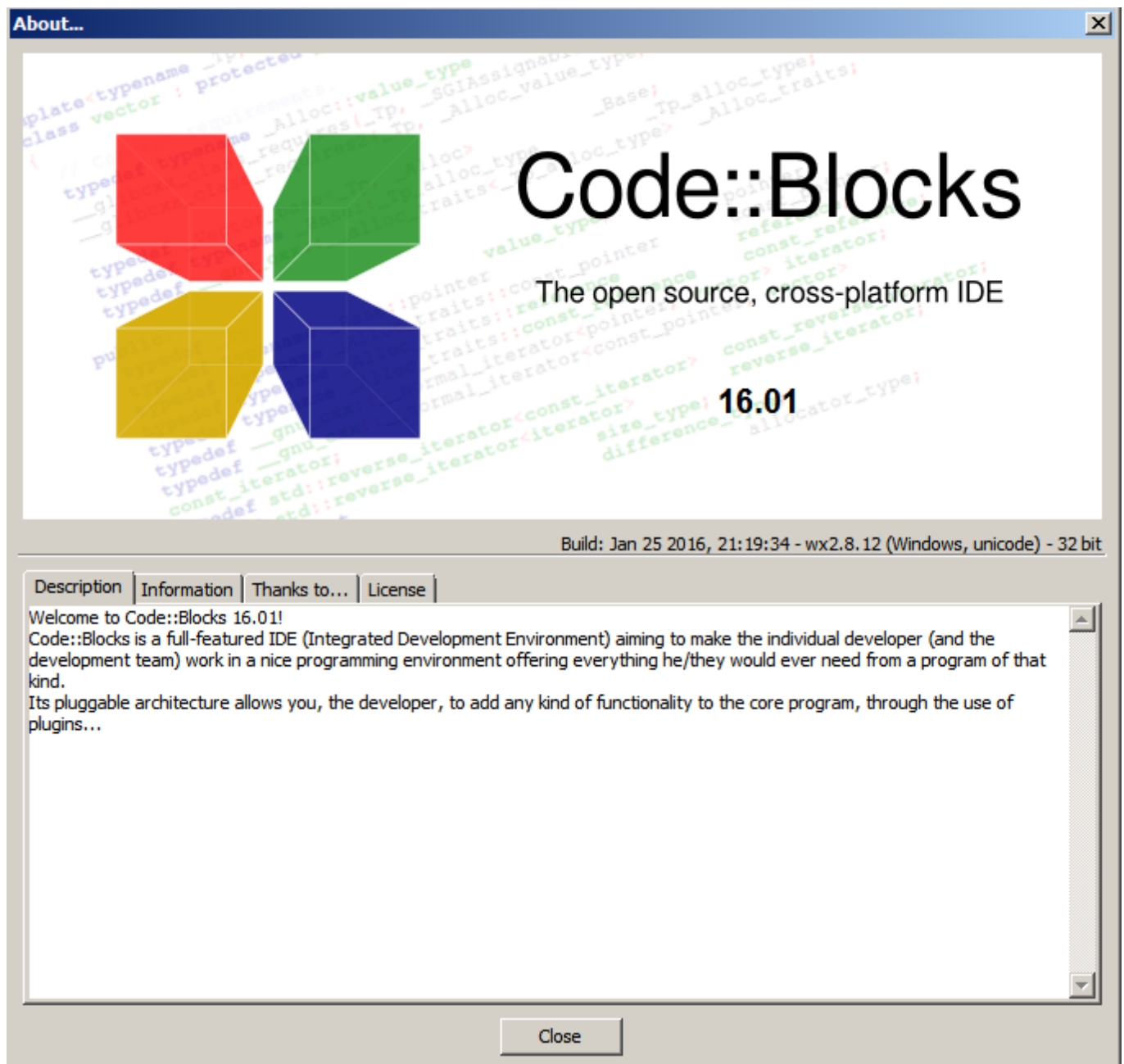
Challenges:

I haven't practised C/C++ programming for several years therefore,

- My solution was based on the assumptions that I made as discussed in step 1
- I am not able to implement solution in the form of a function
- I tried my best to format the display output of data as nicely as possible
- I did not parse data based on '\r\n' i.e. 0xA and 0xD

Appendix:

The IDE that I used for this assignment was:



The Code:

```
// Any header you need here

// Any extra functions, declarations, etc. here

// This function reads a "Sensequake .bin" file located at [filepath], prints its header, and then prints the
// decoded data as text with a space between samples and a newline between frames.
//void print_sensequake_bin(const char* filepath)
//{
//    // TODO: implement this
//}

// This code was written by Nandkshor Motiram Dhawale, On May 22nd 2022.
// This code is written to solve the assignment question no. 1
// The problem statement was given by the CTO member of https://www.sensequake.com/
// This code reads a "Setup7_sensor3.bin" file located at
// [CC:\Projects\Solution_embedded_applicant_test_v2],
// prints its header, and then prints the decoded data as text with a
// space between samples and a newline between frames.

#include <iostream>
#include <fstream>
#include <stdio.h>

using namespace std;
//
//void print_sensequake_bin(const char* filepath)
//{
//    // TODO: implement this
//}

int buffToInteger(char * buffer) // This function is used to convert the decoded 4 bytes into an integer
{
    int a = int((unsigned char)(buffer[0]) << 24 |
                (unsigned char)(buffer[1]) << 16 |
                (unsigned char)(buffer[2]) << 8 |
                (unsigned char)(buffer[3]));
}

int main() {
    char buffer[185]; // to read and store data from the .bin file
    char frame1 [18]; //to store the frame1 data and to display it on the screen
    char frame2 [18]; //to store the frame2 data and to display it on the screen
    char frame3 [18]; //to store the frame3 data and to display it on the screen

    fstream myFile("Setup7_sensor3.bin", ios::binary | ios::in |ios::out);
    if(!myFile.is_open()) {
        cout << " error while opening the file" <<endl;
    } else {
```

```

myFile.read ((char *) &buffer,186);
if (!myFile) {
    cout<<endl<<"The .bin file that you are searching is found and read." <<endl<<endl;
    cout<<endl<<"The total number of bytes read in the file were " << myFile.gcount() <<endl<<endl;
    //cout<<" " << buffer <<endl;

cout<<"Header 1 (approx. 32 Bytes)---> ";
for(int i=0; i < 31; i++)
{
    cout<<buffer[i];    // extracting the first header and displaying the data on the screen
}

cout<<endl<<"Header 2 (approx. 32 Bytes)---> ";
for(int i=34; i < 56; i++)
{
    cout<<buffer[i];    // extracting the second header and displaying the data on the screen
}

cout<<endl<<"Header 3 (approx. 18 Bytes)---> ";
for(int i=57; i < 72; i++)
{
    cout<<buffer[i];    // extracting the third header and displaying the data on the screen
}

cout<<endl<<"Header 4 (approx.8 Bytes)---> ";
for(int i=73; i < 81; i++)
{
    cout<<buffer[i];    // extracting the fourth header and displaying the data on the screen
}

cout<<endl<<"Header 5 (approx.13 Bytes)---> ";
for(int i=84; i < 97; i++)
{
    cout<<buffer[i];    // extracting the fifth header and displaying the data on the screen
}

cout<<endl<<"Header 6 (approx. 23 Bytes)---> ";
for(int i=99; i < 121; i++)
{
    cout<<buffer[i];    // extracting the sixth header and displaying the data on the screen
}

cout<<endl<<endl<<"The size of Frame1 is 18 Bytes. ";
cout<<"The binary data is follows: ";
for(int i=131; i < 149; i++)
{
    frame1[i-131]=buffer[i];
    cout<<frame1[i-131];    // extracting the sample data from first frame and displaying the data
on the screen
}

```

```

cout<<endl<<"The size of Frame2 is 18 Bytes. ";
cout<<"The binary data is as follows: ";
    for(int i=150; i < 167; i++)
    {
        frame2[i-150]=buffer[i];
        cout<<frame2[i-150];           // extracting the sample data from second frame and displaying the
data on the screen
    }
cout<<endl<<"The size of Frame3 is 18 Bytes ";
cout<<"The binary data is as follows: ";
    for(int i=168; i < 185; i++)
    {
        frame3[i-168]=buffer[i];
        cout<<frame3[i-168];           // extracting the sample data from third frame and displaying the data
on the screen
    }
}
}

myFile.clear();
myFile.close();

int iposition;
int oposition;

ifstream infile;
infile.open("Setup7_sensor3.bin", ios::binary | ios::in); // open file again to extract the sample bytes
and to decode
ofstream ofile ("Decoded_Samples.bin",ios::binary | ios::out); // write the decoded sample bytes into
a file

    cout<<endl<<"The six variable axes & its samples are illustrated in a table shown below: "<<endl; //
information
    cout<<endl<<endl<<"Vx"<<"\t"    "<<"Vy"<<"\t"    "<<"Vz"<<"\t"    "<<"Ax"<<"\t"    "<<"Ay"<<"\t"
"<<"Az"<<endl<<endl;
    int k=131;
    int l=0;
    for (int j=0; j < 70; j=j+4)
    {
        infile.seekg(k, ios::beg); //assign the seek cursor pointer to the fist before one of the starting sample
        iposition = infile.tellg(); //
        //cout<<"The first byte position of sample was:"<<iposition<<" ";
        char A[4];                //create a char array to store the 4bytes
        infile.read ((char*)&A, sizeof(A)); //read the consecutive four bytes from the input file
        A[0]=0x0;                 // make the LSB byte to zero
        k=k+3;                    //increment the seek curs0r pointer to read the next 3 bytes

        ofile.seekp(j, ios::beg);
        oposition = ofile.tellp();
        //cout<<"Current Position of Opointer is:"<<oposition<<endl;
        ofile.write((char*)&A, sizeof(A)); // write the four decoded bytes to the output file

```



```

int b=buffToInteger((char*)&A); //call function to convert the decoded bytes into integer value

    cout<<b<<" ";           // display the samples as per the axes variables
    if ((l == 5)||(l==11))
    {
        cout<<endl<<"<----- Newline between frames ----->"<<endl; // add newline between frames
    }
    l=l+1;
}
cout<<endl<<endl<<"There are a total 18 samples found as shown in 3 rows and 6 columns as above.";
// Information
cout<<" The six in first row are from the frame 1.";           // Information
cout<<" The six in second row are from the frame 2.";           // Information
cout<<" The six in third row are from the frame 3.";           // Information
cout<<" Each sample in each frame row belongs to the six axes respectively.";           //
Information
cout<<" Therefore there are 3 samples per axes as shown in columns.";           // Information
cout<<" The decoded binary file is also generated and saved in the folder.";           // Information
cout<<" It can be viewed using HEX a viewer application such as Notepad++."<<endl;           //
Information

infile.close(); // close input file
ofile.close(); // close output file
}

```