IEEE754 floating-point Translate

Here we provide an example by C, C# and VB to translate between float and byte format.

◆ C

```
/*****************************************************************************
**      Main Function    main()
*****************************************************************************/
int main (void)
{
        //Example: float=22.11 to byte array = 0x41 0xB0 0xE1 0x48
       unsigned char data[4];
       unsigned char *datap;
       float fdata;
       float *fdatap;

       //===Solution 1: Get float=============================================//
       printf("\n####       Get float from byte array by mapping method       ####\n");
       fdata = 0; //reset fdata
       data[0] = 0x48; data[1] = 0xE1; data[2] = 0xB0; data[3] = 0x41; //init data array
       fdata = *(float *)data;
       printf("float value = %f\n", fdata);
       //===Solution 2: Get float=============================================//
       printf("\n####       Get float from byte array by function call        ####\n");
       fdata = 0; //reset fdata
       data[0] = 0x48; data[1] = 0xE1; data[2] = 0xB0; data[3] = 0x41; //init data array
       if (GetFloatFromByte(&fdata, data[3], data[2], data[1], data[0])==0)
          printf("float value = %f\n", fdata);
       else
          printf("GetFloatFromByte by function failed !!\n");
       //===Solution 1: Get byte==============================================//
       printf("\n####       Get byte array from float by mapping method       ####\n");
       fdata = 22.11; //init fdata
       data[0] = 0; data[1] = 0; data[2] = 0; data[3] = 0; //reset data array
       fdatap = &fdata;
       datap = (unsigned char *)fdatap;
       printf("byte array = %2X%2X%2X%2X\n",datap[3],datap[2],datap[1],datap[0]);
       //===Solution 2: Get byte==============================================//
       printf("\n####       Get byte array from float by function call        ####\n");
       fdata = 22.11; //init fdata
       data[0] = 0; data[1] = 0; data[2] = 0; data[3] = 0; //reset data array
       if (GetByteFromFloat(fdata, &data[3], &data[2], &data[1], &data[0])==0)
          printf("byte array = %2X%2X%2X%2X\n",data[3],data[2],data[1],data[0]);
       else
          printf("GetByteFromFloat by function failed !!\n");
        //=====================================================================//

       return 0;
}


int GetFloatFromByte(float *fdata, unsigned char data0, unsigned char data1, unsigned char data2, unsigned char data3)
{
   //Get IEEE 754 Floating point format from data0, data1, data2, data3
   //fdata : float data
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
    //data0, data1, data2, data3 : 8-bit Hexadecimal data
    //return code ==> 0: OK, -1: Not Available

    int i, s=1;
    unsigned char e=0;
    unsigned long m=0;
    double value, result;

    if (data0 & 0x80)
       s=-1;
    e=(data0 & 0x7F)<<1;
    e = e | ((data1 & 0x80)?1:0);
    m=(data1 & 0x7F)<<16;
    m = m | (data2<<8);
    m = m | data3;

    if ((e == 0) && (m == 0))
    {
       fdata = 0;
       return 0;
    }
    else if ((data0 == 0x7F) && (data1 >= 0x80))//7F800001~7FFFFFFF
    {
       if ((data1 != 0x80) && (data3 != 0x00))
          return -1;
    }
    else if ((data0 == 0xFF) && (data1 >= 0x80))//FF800001~FFFFFFFF
    {
       if ((data1 != 0x80) && (data3 != 0x00))
          return -1;
    }

    result=1;
    for (i=-23;i<0;i++)
    {
       value = (m & 0x00000001)?pow(2,i):0;
       result+=value;
       m=m>>1;
    }
    result*=s;
    result*=pow(2, e-127);
    *fdata=result;

    return 0;
}

int GetByteFromFloat(float fdata, unsigned char *data0, unsigned char *data1, unsigned char *data2, unsigned char
*data3)
{
    //Get data0, data1, data2, data3 from IEEE 754 Floating point format
    //fdata : float data
    //data0, data1, data2, data3 : 8-bit Hexadecimal data
    //return code ==> 0: OK, -1: Not Available

    int i;
    unsigned char s = 0, e = 0;
    unsigned char find = 0, bit_value = 0;
    unsigned long m = 0;
    double value, temp;
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
    if (fdata < 0)
    {
        s = 1;
        fdata *= -1;
    }

    i = -127;
    while ((fdata / pow(2, i)) > 1)
    {
        i++;
        if (i > 128)
            return -1;
    }
    if (i <= -127)
        return -1;

    e = (unsigned char)(i - 1 + 127);
    value = fdata / pow(2, (i - 1));
    value -= 1;

    for (i = -1; i > -24; i--)
    {
        bit_value = 0;
        if (find == 0)
        {
            temp = pow(2, i);
            if (value == temp)
            {
                find = 1;
                bit_value = 1;
            }
            else if (value > temp)
            {
                value -= temp;
                bit_value = 1;
            }
            else
                bit_value = 0;
        }
        m |= bit_value;
        if (i > -23)
            m = m << 1;
    }

    //process data
    *data0 = (unsigned char)((s << 7) + (e >> 1));
    *data1 = (unsigned char)(((e & 0x01) << 7) + ((m & 0xFF0000) >> 16));
    *data2 = (unsigned char)((m & 0xFF00) >> 8);
    *data3 = (unsigned char)(m & 0xFF);

    return 0;
}
```

◆ C#

```
public int GetFloatFromByte(ref double fdata, Byte data0, Byte data1, Byte data2, Byte data3)
{
            //Get IEEE 754 Floating point format from data0, data1, data2, data3
            //fdata : float data
            //data0, data1, data2, data3 : 32-bit Hexadecimal data
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
            //return code ==> 0: OK, -1: Not Available

            byte[] arr = new byte[4];
            arr[0] = data3;
            arr[1] = data2;
            arr[2] = data1;
            arr[3] = data0;
            fdata = BitConverter.ToSingle(arr, 0);

            return 0;
}

public int GetByteFromFloat(double fdata, ref Byte data0, ref Byte data1, ref Byte data2, ref Byte data3)
{
            //Get data0, data1, data2, data3 from IEEE 754 Floating point format
            //fdata : float data
            //data0, data1, data2, data3 : 32-bit Hexadecimal data
            //return code ==> 0: OK, -1: Not Available

            byte[] arr = new byte[4];
            float value = (float)fdata;
            arr = BitConverter.GetBytes(value);
            data0 = arr[3];
            data1 = arr[2];
            data2 = arr[1];
            data3 = arr[0];

            return 0;
}
```

◆ VB

```
Private Function GetFloatFromByte(ByRef fdata As Double, ByVal data0 As Byte, ByVal data1 As Byte, ByVal data2
As Byte, ByVal data3 As Byte) As Integer
        'Get IEEE 754 Floating point format from data0, data1, data2, data3
        'fdata : float data
        'data0, data1, data2, data3 : 32-bit Hexadecimal data
        'return code ==> 0: OK, -1: Not Available

        Dim arr(3) As Byte
        arr(0) = data3
        arr(1) = data2
        arr(2) = data1
        arr(3) = data0
        fdata = BitConverter.ToSingle(arr, 0)

        Return 0
End Function

Private Function GetByteFromFloat(ByVal fdata As Double, ByRef data0 As Byte, ByRef data1 As Byte, ByRef data2
As Byte, ByRef data3 As Byte) As Integer
        'Get data0, data1, data2, data3 from IEEE 754 Floating point format
        'fdata : float data
        'data0, data1, data2, data3 : 32-bit Hexadecimal data
        'return code ==> 0: OK, -1: Not Available

        Dim arr(3) As Byte
        Dim value As Single
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
        value = fdata
        arr = BitConverter.GetBytes(value)
        data0 = arr(3)
        data1 = arr(2)
        data2 = arr(1)
        data3 = arr(0)

        Return 0
End Function
```

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000