

The sequence of the jump frequency will be settle by the host.
The method is the host to be connected to the device, The sequence data S(8bit) and HostID(16bit) will bring through scan the RSSI of RF Module in different channel, and the data will be transmitted to the device. Then the both side will calculate to the next number (F) by same agument(S).

The way of bring sequence is as follows:

N: the numbers of useful channel

S: number of seed (8bit)=S7-S0

P: replacement control (21bit)=P20-P0 ,Each 3 bit become one figure,
as follow :

$$Q_6 = P_{20}P_{19}P_{18} \quad Q_5 = P_{17}P_{16}P_{15} \quad Q_4 = P_{14}P_{13}P_{12}$$

$$Q_3 = P_{11}P_{10}P_9 \quad Q_2 = P_8P_7P_6 \quad Q_1 = P_5P_4P_3$$

$$Q_0 = P_2P_1P_0$$

$Q_i \neq Q_j$ ($0 \leq i < 6$, $i < j \leq 6$) to be request!

F: the figure of the sequence ($0 \leq F < N$)

Supurse the S to be S(k) in K cycle. It means the channel of the (K+1) cycle will be F(K+1) channel.

$$1. S(k+1) = S(k) + 1 = S(k+1)_7 \sim S(k+1)_0$$

2. $S(k+1)$'s Bit6~Bit0 will conversion to $R(k+1)=r(k+1)_7 \sim r(k+1)_0$

$r(k+1)_i = S(k+1)_{0i} \ (0 \leq i < 6)$, $r(k+1)_7 = s(k+1)_7$ to be request!

3. $F(k+1)=R(k+1) \bmod N$

So we got to know from above procedure, S data is come from the device ,then transmitter the data to the device. And the above replace control data P is calculate from 16bit HostID of host by the Host and Device both side.

Please reference to the following C program for to calculate method.

```
void FindFHSeed(unsigned char HostIDH, unsigned char HostIDL)
{
//-----
//      The following code does the same job, but in C language format
    unsigned short b = 5040;
    unsigned short m;
    char i,j,k,l;
    m = (unsigned short)(HostIDH) * 0x0100 + (unsigned short)(HostIDL);
    for (i=7; i>1; i--)
    {
        m %= b;
        b /= (unsigned short)i;
        q[i-1] = (unsigned char)(m / b);
    }
    q[0] = 0;
    for (i=6; i>0; i--)
    {
        for (j=i-1; j>=0; j--)
```

```

    {
        if (q[j] >= q[i]) q[j]++;
        do
        {
            l = 0;
            for (k=i; k<7; k++)
            {
                if (q[j] == q[k])
                {
                    q[j]++;
                    l = 1;
                }
            }
        }while (l!=0);
    }
}

//-----

}

void NextFrequencyHop() //Calculate the frequency channel of the next hop
{
    // "seed" and "q[0]- q[6]" are known parameters.
    seed++;          //Result is in r18
    r11 = 0x01;
    if (seed & 0x80) r18 = 0x80;
    else r18 = 0x00;
    for (r10=0; r10<7; r10++)
    {
        if (r11 & seed)
        {
            r12 = q[r10];
            r18 |= errorPattern[r12];
        }
    }
}

```

```
    }  
    r11 <<= 1;  
  }  
  while (r18 >= TOTAL_CHANNEL_NO) r18 -= TOTAL_CHANNEL_NO;  
}
```