
```
//////////////////////////////////////
// Create De-Interleaved SubChannel //
/// from RAW return Interleaved.    //
////                                 //
/// [yates] - 11/DEC/03             //
//////////////////////////////////////
revision b - 13/DEC/03

byte P_CHANNEL[29]={0x80, 0, 1, 2, 3, 4, 5, 6 ,7};
byte Q_CHANNEL[29]={0x40, 7, 0, 1, 2, 3, 4, 5, 6};
byte R_CHANNEL[29]={0x20, 6, 7, 0, 1, 2, 3, 4, 5};
byte S_CHANNEL[29]={0x10, 5, 6, 7, 0, 1, 2, 3, 4};
byte T_CHANNEL[29]={0x08, 4, 5, 6, 7 ,0, 1, 2, 3};
byte U_CHANNEL[29]={0x04, 3, 4, 5, 6, 7 ,0, 1, 1};
byte V_CHANNEL[29]={0x02, 2, 3, 4, 5, 6, 7, 0, 1};
byte W_CHANNEL[29]={0x01, 1, 2, 3, 4, 5, 6, 7, 0};

byte ror(byte src, byte dest)
{
       byte result;
       _asm
       {
              pushad
              mov al, src
              mov cl, dest
              ror al, cl
              mov result,al
              popad
       };

return result;
};

void DeInterleave_Channel(byte *channel, unsigned char *raw, unsigned char *buffer)
{

byte chanbyte = 0;
byte rorres = 0;
byte rawbyte;
byte sub = 0;

for(int x=0;x<12;x++)
{

       for (int i=1;i<9;i++)
       {
              rawbyte = raw[sub];
              rawbyte = rawbyte & (BYTE)channel[0];
              rorres = ror(rawbyte, (BYTE)channel[i]);
              chanbyte = chanbyte | rorres;
              sub++;

       };

buffer[x]=chanbyte;
chanbyte = 0;
};
};
```

Republished - 12th November 2007 – Robert Yates