

Cellcorder Upgrade Protocol

Structure of upgrade file

The upgrade file is composed of many data lines. The structure of every line is shown below.

| START | BYTE COUNT N | FIRST DATA ADDRESS | NO USE | FIRST DATA | ... | N th DATA | END |
|-------|--------------|--------------------|---------|------------|-----|-----------|---------|
| : | 2 Bytes | 4 Bytes | 2 Bytes | 2 Bytes | | 2 Bytes | 2 Bytes |

The data in the line is constructed using hexadecimal digits, in the range of 00 to FF hexadecimal. For example, if the BYTE COUNT is 12, that means there are 18 (1x16+2) data bytes in this line.

Structure of frame

The first command sent to firmware is :

| Command | ID | Address(H) | Address(L) | Data | No use | CheckSum |
|---------|----|------------|------------|------|--------|----------|
| 16H | 0 | 0 | 0 | 0 | 0 | xx |

Wait about 5 seconds to make sure firmware is running from old firmware.

Only the data whose address is **greater than 8000H** can be sent to firmware. Firmware can't write data to low address. There are 7 bytes in one frame . The structure of the frame sent to firmware is shown below.

| Command | ID | Address(H) | Address(L) | Data | No use | CheckSum |
|---------|----|------------|------------|------|--------|----------|
| 14H | 0 | xx | xx | xx | 0 | xx |

Since the data (Address, Data) in the frame is a decimal number, the data in the upgrade file should be converted from HEX to DEC and then sent to firmware.

After all the data is sent to firmware, software should send a last frame.

| Command | ID | Address(H) | Address(L) | Data | No use | CheckSum |
|---------|----|------------|------------|------|--------|----------|
| 14H | 0 | 0 | 0 | 0 | 0 | xx |

The following example is written in C language.

```
/*----- DEMO -----*/
if(pf) /* pf point to upgrade file */
{
    send_comm(PROGRAM_CMD,port_id,0,0,0,0); /* send first frame */

    delay(50); /* delay 5 seconds */

    while(fgets(databuff.s,sizeof(databuff.s),pf)!=NULL)
    {

        /* calculate data counts */
        cnt =(convert_hex(databuff.s[1]))*16;
        cnt+= convert_hex(databuff.s[2]);

        /* Calculate first address */
        addr =(convert_hex(databuff.s[3]))*(16*256);
        addr+=(convert_hex(databuff.s[4]))*256;
        addr+=(convert_hex(databuff.s[5]))*16;
        addr+= convert_hex(databuff.s[6]);

        if(addr>=0x8000)
        {
            buff_ptr=9; /* point to data */
            for(y=0;y<cnt;y++)
            {
                data =(convert_hex(databuff.s[buff_ptr++]))*16;
                data+= convert_hex(databuff.s[buff_ptr++]);

                send_comm(WRITE_CMD,port_id,(addr>>8),(addr&0xf
f),data,0);

                if((y%5)==0)delay(1);

                addr++;
            }
        }

        send_comm(WRITE_CMD,port_id,0,0,0,0); /* send last frame */
    }
}
/*----- End of DEMO -----*/
```

Example of upgrade file:

:090000000200402000550216DE4A
:06000B002000500217B4B2
:0600130020004B02171D46
:06001B0020004602175C04
:0600230020003202175D0F
:1000400075250002010075250102010075250202D7
:100050000100752503020100028023028003028053
:070060000B02801302801B5C
:068000000200400216DE42
:03800B000217B4A5
:0380130002171D34
:03801B0002175CED
:0380230002175DE4
:10808000303132333435363738393938373635349C
:10809000333231303031323334353637383939389C
:0880A00037363534333231303C
:1000C00020434F505952494748542031393935203F
:1000D00020414C424552434F52502020202020A6
:1000E0002056455253494F4E20322E30202020209A
:1000F0002030392F32302F393620202020202068
.
.
.