# FBO文件格式说明

1. FBO文件结构说明：
2. 文件头长度为256字节，结构体为FBOFILEINF
3. 测试数据需要跳过头部的256字节开始读取，读取的每个数据包的长度=（（FBODATA结构体）-1000+（单体数\*2））；
4. 使用到的结构体：

//日期时间

typedef struct \_FBOdatetime

{

unsigned char year; // 年（实际年份-2000）

unsigned char month;

unsigned char day;

unsigned char hour;

unsigned char minute;

unsigned char second;

}Date\_Time;

//时间

typedef struct \_FBOworktime

{

unsigned char hour;

unsigned char minute;

unsigned char second;

}Test\_Time;

//数据头标记，连续的四个相同的值，FC表示充电，FD表示放电

typedef struct \_FBOdatatypetag

{

unsigned char TypeTag0;

unsigned char TypeTag1;

unsigned char TypeTag2;

unsigned char TypeTag3;

}DATA\_TYPE;

//

typedef struct \_FBOdatatag

{

unsigned short CRC16; // 校验码

Test\_Time m\_TestTime; // 测试时长

unsigned char BattGroup; //电池组数

unsigned short BattSum; // 电池节数

unsigned short OnlineVol; // 在线电压

unsigned short SumVoltage; // 组端电压

unsigned short SumCurrent; // 电流

unsigned short SubCurrent[4]; // 支路电流

unsigned short AllCap; // 测试容量

unsigned short SubCap[4]; // 支路测试容量

unsigned short SingleVol[500]; // 单体电压

}FBO\_DATA;

// 测试数据结构

typedef struct FBORT232Data\_tag

{

DATA\_TYPE DataType;

FBO\_DATA FBOData;

}FBODATA;

typedef struct \_FBOdatainf1

{

Test\_Time TestTimeLong; //测试时长

unsigned char StopType; //结束方式

unsigned char BlockSum; //保存数据的总块数

unsigned char StandBy; //保留备用

unsigned short SMaxIndex[4]; //最高单体索引

unsigned short SMinIndex[4]; //最低单体索引

unsigned short SMaxVol[4]; //最高单体

unsigned short SMinVol[4]; //最低单体

unsigned short TestCap; //测试容量

}Data\_Stop\_inf;

typedef struct \_FBOdatainf

{

Date\_Time TestStartTime; //放电开始的时间

unsigned char Device; //仪表类型

unsigned char DataVersion; //数据版本

unsigned char DataType; //数据类型;0xFD表示放电,0xFC表示充电

unsigned char HourRate; //小时率

unsigned char SaveInterval; //采集间隔

unsigned char MonomerVol; //单体电压类型

unsigned short STDCap; //标称容量

unsigned short TestCur; //测试电流

unsigned short MVLLimit; //单体下限

unsigned short SumVLLimit; //组端下限

unsigned short BattSum; //单体数量

unsigned short BattGroup; //电池组数

unsigned short MVLLimitCount;//单体下限个数

}Data\_Start\_Inf;

//文件头结构

typedef struct \_FBOfileinf

{

Data\_Start\_Inf TestStartInf;

Data\_Stop\_inf TestStopInf;

}FBOFILEINF;

1. CRC算法

const unsigned short CRC16Table[256] =

{

0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50A5, 0x60C6, 0x70E7,

0x8108, 0x9129, 0xA14A, 0xB16B, 0xC18C, 0xD1AD, 0xE1CE, 0xF1EF,

0x1231, 0x0210, 0x3273, 0x2252, 0x52B5, 0x4294, 0x72F7, 0x62D6,

0x9339, 0x8318, 0xB37B, 0xA35A, 0xD3BD, 0xC39C, 0xF3FF, 0xE3DE,

0x2462, 0x3443, 0x0420, 0x1401, 0x64E6, 0x74C7, 0x44A4, 0x5485,

0xA56A, 0xB54B, 0x8528, 0x9509, 0xE5EE, 0xF5CF, 0xC5AC, 0xD58D,

0x3653, 0x2672, 0x1611, 0x0630, 0x76D7, 0x66F6, 0x5695, 0x46B4,

0xB75B, 0xA77A, 0x9719, 0x8738, 0xF7DF, 0xE7FE, 0xD79D, 0xC7BC,

0x48C4, 0x58E5, 0x6886, 0x78A7, 0x0840, 0x1861, 0x2802, 0x3823,

0xC9CC, 0xD9ED, 0xE98E, 0xF9AF, 0x8948, 0x9969, 0xA90A, 0xB92B,

0x5AF5, 0x4AD4, 0x7AB7, 0x6A96, 0x1A71, 0x0A50, 0x3A33, 0x2A12,

0xDBFD, 0xCBDC, 0xFBBF, 0xEB9E, 0x9B79, 0x8B58, 0xBB3B, 0xAB1A,

0x6CA6, 0x7C87, 0x4CE4, 0x5CC5, 0x2C22, 0x3C03, 0x0C60, 0x1C41,

0xEDAE, 0xFD8F, 0xCDEC, 0xDDCD, 0xAD2A, 0xBD0B, 0x8D68, 0x9D49,

0x7E97, 0x6EB6, 0x5ED5, 0x4EF4, 0x3E13, 0x2E32, 0x1E51, 0x0E70,

0xFF9F, 0xEFBE, 0xDFDD, 0xCFFC, 0xBF1B, 0xAF3A, 0x9F59, 0x8F78,

0x9188, 0x81A9, 0xB1CA, 0xA1EB, 0xD10C, 0xC12D, 0xF14E, 0xE16F,

0x1080, 0x00A1, 0x30C2, 0x20E3, 0x5004, 0x4025, 0x7046, 0x6067,

0x83B9, 0x9398, 0xA3FB, 0xB3DA, 0xC33D, 0xD31C, 0xE37F, 0xF35E,

0x02B1, 0x1290, 0x22F3, 0x32D2, 0x4235, 0x5214, 0x6277, 0x7256,

0xB5EA, 0xA5CB, 0x95A8, 0x8589, 0xF56E, 0xE54F, 0xD52C, 0xC50D,

0x34E2, 0x24C3, 0x14A0, 0x0481, 0x7466, 0x6447, 0x5424, 0x4405,

0xA7DB, 0xB7FA, 0x8799, 0x97B8, 0xE75F, 0xF77E, 0xC71D, 0xD73C,

0x26D3, 0x36F2, 0x0691, 0x16B0, 0x6657, 0x7676, 0x4615, 0x5634,

0xD94C, 0xC96D, 0xF90E, 0xE92F, 0x99C8, 0x89E9, 0xB98A, 0xA9AB,

0x5844, 0x4865, 0x7806, 0x6827, 0x18C0, 0x08E1, 0x3882, 0x28A3,

0xCB7D, 0xDB5C, 0xEB3F, 0xFB1E, 0x8BF9, 0x9BD8, 0xABBB, 0xBB9A,

0x4A75, 0x5A54, 0x6A37, 0x7A16, 0x0AF1, 0x1AD0, 0x2AB3, 0x3A92,

0xFD2E, 0xED0F, 0xDD6C, 0xCD4D, 0xBDAA, 0xAD8B, 0x9DE8, 0x8DC9,

0x7C26, 0x6C07, 0x5C64, 0x4C45, 0x3CA2, 0x2C83, 0x1CE0, 0x0CC1,

0xEF1F, 0xFF3E, 0xCF5D, 0xDF7C, 0xAF9B, 0xBFBA, 0x8FD9, 0x9FF8,

0x6E17, 0x7E36, 0x4E55, 0x5E74, 0x2E93, 0x3EB2, 0x0ED1, 0x1EF0

};

//---------------------------------------------------------------------------

unsigned short \_\_fastcall CalCRC16(const void \*data, const int count)

{

const unsigned char \*pdata = (const unsigned char \*)data;

unsigned short crc = 0x00;

unsigned short i, tmp1;

for (i=0; i<count; i++)

{

tmp1 = crc;

crc = (unsigned short)CRC16Table[(tmp1>>8) ^ (\*pdata++)];

crc = crc ^ (tmp1<<8);

}

return(crc);

}