**Spi 通讯协议**

**简述**

本文的描述 WiFi 和单片机之间进行串口通讯，注意通讯是继续四线 SPI，通讯速度支持大于 1MB/s .

spi 配置：

* WiFi 作为 spi master
* Clock 速度为 8MHZ
* MSB

**注意： 协议中所有大于 2Byte 数据高位在前，低位在后。**

**协议结构**

|  |  |  |  |
| --- | --- | --- | --- |
| 字节序 | 项 | 长度（byte） | 说明 |
| 0 | mask | 1 | 协议头，目前固定为 0x51 |
| 1 | cmd | 1 | 命令 |
| 2~3 | len | 2 | data 部分的长度，高位在前低位在后 |
| 4~n+4 | data | n | 数据 |
| n + 5 | crc8 | 1 | 包括 mask + cmd + len + data 的crc8 校验，算法C code见附录 |

|  |
| --- |
| C++ typedef struct SPIPROTOCOL\_T{  uint8\_t mask;  uint8\_t cmd;  uint16\_t len;  uint8\_t data[];  uint8\_t crc8; }SpiProtocol\_T; |

**错误码(BT端添加)**

|  |  |
| --- | --- |
| 0 | 成功 |
| 1 | 操作失败 |
| 2 | 写入sd卡失败 |
| 3 | 传输出错 |

**命令**

**0x10 写入歌曲名称**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字节序 | 项 | 长度（byte） | 值 | 说明 |
| 0 | mask | 1 |  | 协议头，目前固定为 0x51 |
| 1 | cmd | 1 | 0x10 | 命令 |
| 2~3 | len | 2 | 20 + n | data 部分的长度，高位在前低位在后 |
| 4~7 | totoLen | 4 |  | 歌曲总的字节长度 |
| 8 ~ 23 | md5 | 16 |  | 该文件的 md5 |
| 24~ 25+n | path | n | string | 字符串，保存在文件中的路径 |
| 26 + n | crc8 | 1 |  | 包括 mask + cmd + len + data 的crc8 校验，算法C code见附录 |

Data 字段结构：

|  |
| --- |
| C++ typedef struct DATA\_T{  uint32\_t totoLen;  uint8\_t md5[16];  unsigned char path[]; }Data\_T; |

**0x11 写入名称回应**

* Bluetooth 回应成功代表可以开始写入

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字节序 | 项 | 长度（byte） | 值 | 说明 |
| 0 | mask | 1 |  | 协议头，目前固定为 0x51 |
| 1 | cmd | 1 | 0x11 | 命令 |
| 2~3 | len | 2 | 16 | data 部分的长度，高位在前低位在后 |
| 4 | ack | 1 | 0 | 0->成功，可以开始写入；否则为失败 |
| 5 | crc8 | 1 |  | 包括 mask + cmd + len + data 的crc8 校验，算法C code见附录 |

**0x12 写入歌曲**

* 速度为首，没有ack，传输完成后通过 0x20确认写入是否成功.
* 一首歌过大，采用分包传输.
* 暂定单次传输最大不超过 1024Bytes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字节序 | 项 | 长度（byte） | 值 | 说明 |
| 0 | mask | 1 |  | 协议头，目前固定为 0x51 |
| 1 | cmd | 1 | 0x12 | 命令 |
| 2~3 | len | 2 | 2 + n | data 部分的长度，高位在前低位在后 |
| 4 | total | 1 |  | 总包数 |
| 5 | index | 1 |  | 报数 |
| 6~ 6+n | rawData | n |  | 原始的音乐数据 |
| 7 + n | crc8 | 1 |  | 包括 mask + cmd + len + data 的crc8 校验，算法C code见附录 |

Data 字段结构：

|  |
| --- |
| C++ typedef struct MUSCIDATA\_T{  uint8\_t total;  uint8\_t index;  uint16\_t rawData[]; }MusicData\_T; |

**0x13 获取歌曲写入结果**

* WiFi 发出该请求会等待回应，若超过 3s 无 0x21 回应认为写入歌曲失败。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字节序 | 项 | 长度（byte） | 值 | 说明 |
| 0 | mask | 1 |  | 协议头，目前固定为 0x51 |
| 1 | cmd | 1 | 0x13 | 命令 |
| 2~3 | len | 2 | 16 | data 部分的长度，高位在前低位在后 |
| 4~19 | md5 | 16 |  | 该文件的 md5，用于校验文件 |
| 20 | crc8 | 1 |  | 包括 mask + cmd + len + data 的crc8 校验，算法C code见附录 |

**0x14 BT 回应文件写入结果**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字节序 | 项 | 长度（byte） | 值 | 说明 |
| 0 | mask | 1 |  | 协议头，目前固定为 0x51 |
| 1 | cmd | 1 | 0x14 | 命令 |
| 2~3 | len | 2 | 22 + nlen+plen | data 部分的长度，高位在前低位在后 |
| 4~7 | id | 4 |  | 文件唯一id，由保存音乐的蓝牙端产生，后续接口通过该 id 操作文件. |
| 8 ~ 23 | md5 | 16 |  | 该文件的 md5 |
| 24 | nameLen | 1 |  | 文件名称长度 |
| 25~25 +nlen | name | nlen | string | 文件名称 |
| 26 + nlen | pathLen | 1 |  | 文件保存路径长度 |
| 27+nlen~ 27+plen+nlen | path | plen | string | 字符串，保存在文件中的路径 |
| 28+plen + nlen | crc8 | 1 |  | 包括 mask + cmd + len + data 的crc8 校验，算法C code见附录 |

Data 字段结构：

|  |
| --- |
| C++ typedef struct MUSCIDATA\_T{  uint32\_t id;  uint8\_t md5[16];  unsigned char path[]; }MusicData\_T; |

**0x30 删除歌曲**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字节序 | 项 | 长度（byte） | 值 | 说明 |
| 0 | mask | 1 |  | 协议头，目前固定为 0x51 |
| 1 | cmd | 1 | 0x30 | 命令 |
| 2~3 | len | 2 | 16 | data 部分的长度，高位在前低位在后 |
| 4~7 | id | 4 |  | 要删除歌曲的id |
| 8 | crc8 | 1 |  | 包括 mask + cmd + len + data 的crc8 校验，算法C code见附录 |

**0x31 BT 回应删除结果**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字节序 | 项 | 长度（byte） | 值 | 说明 |
| 0 | mask | 1 |  | 协议头，目前固定为 0x51 |
| 1 | cmd | 1 | 0x31 | 命令 |
| 2~3 | len | 2 | 16 | data 部分的长度，高位在前低位在后 |
| 4 | ret | 1 | 0 | 删除结果： 0：成功，其他值为失败 |
| 5~8 | id | 4 |  | 已删除歌曲的id |
| 9 | crc8 | 1 |  | 包括 mask + cmd + len + data 的crc8 校验，算法C code见附录 |

**0x40 获取文件列表(TODO)**

**0x41 获取文件列表(TODO)**

**0x50 播放控制**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字节序 | 项 | 长度（byte） | 值 | 说明 |
| 0 | mask | 1 |  | 协议头，目前固定为 0x51 |
| 1 | cmd | 1 | 0x50 | 命令 |
| 2~3 | len | 2 | 1 | data 部分的长度，高位在前低位在后 |
| 4 | control | 1 |  | 0->开始播放  1->暂停播放  2->下一曲  3->上一曲  4->音量增加  5->音量减少 |
| 5 | crc8 | 1 |  | 包括 mask + cmd + len + data 的crc8 校验，算法C code见附录 |

**0x51 BT回应播放控制**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字节序 | 项 | 长度（byte） | 值 | 说明 |
| 0 | mask | 1 |  | 协议头，目前固定为 0x51 |
| 1 | cmd | 1 | 0x51 | 命令 |
| 2~3 | len | 2 | 20 + n | data 部分的长度，高位在前低位在后 |
| 4 | ret | 1 | 0 | 控制结果： 0->成功，否则失败 |
| 5~8 | id | 4 | 9527 | 文件唯一id，由保存音乐的蓝牙端产生，后续接口通过该 id 操作文件. |
| 9~ 9+n | name | n | string | 字符串，当前播放的文件名称 |
| 10 + n | crc8 | 1 |  | 包括 mask + cmd + len + data 的crc8 校验，算法C code见附录 |

**0x60 设置 EQ模式**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字节序 | 项 | 长度（byte） | 值 | 说明 |
| 0 | mask | 1 |  | 协议头，目前固定为 0x51 |
| 1 | cmd | 1 | 0x60 | 命令 |
| 2~3 | len | 2 | 1 | data 部分的长度，高位在前低位在后 |
| 4 | EQ | 1 | 1 | 可能的值：1/2/3 |
| 5 | crc8 | 1 |  | 包括 mask + cmd + len + data 的crc8 校验，算法C code见附录 |

**0x61 BT回应 EQ 设置**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字节序 | 项 | 长度（byte） | 值 | 说明 |
| 0 | mask | 1 |  | 协议头，目前固定为 0x51 |
| 1 | cmd | 1 | 0x61 | 命令 |
| 2~3 | len | 2 | 1 | data 部分的长度，高位在前低位在后 |
| 4 | ret | 1 | 0 | 设置结果： 0->成功，否则失败 |
| 5 | EQ | 1 | 1 | 当前 EQ 值 |
| 6 | crc8 | 1 |  | 包括 mask + cmd + len + data 的crc8 校验，算法C code见附录 |

**0x70 切换音源**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字节序 | 项 | 长度（byte） | 值 | 说明 |
| 0 | mask | 1 |  | 协议头，目前固定为 0x51 |
| 1 | cmd | 1 | 0x70 | 命令 |
| 2~3 | len | 2 | 2 | data 部分的长度，高位在前低位在后 |
| 4~5 | data | 2 | TF | String 字符串：  TF -> 音源为TF  BT->音源为BT |
| 6 | crc8 | 1 |  | 包括 mask + cmd + len + data 的crc8 校验，算法C code见附录 |

**0x71 BT回应切换音源**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字节序 | 项 | 长度（byte） | 值 | 说明 |
| 0 | mask | 1 |  | 协议头，目前固定为 0x51 |
| 1 | cmd | 1 | 0x71 | 命令 |
| 2~3 | len | 2 | 3 | data 部分的长度，高位在前低位在后 |
| 4 | ret | 1 | 0 | 设置结果： 0->成功，否则失败 |
| 5~6 | source | 2 | TF | 字符串，表明当前音源 |
| 7 | crc8 | 1 |  | 包括 mask + cmd + len + data 的crc8 校验，算法C code见附录 |

**0x80 TF卡状态查询**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字节序 | 项 | 长度（byte） | 值 | 说明 |
| 0 | mask | 1 |  | 协议头，目前固定为 0x51 |
| 1 | cmd | 1 | 0x80 | 命令 |
| 2~3 | len | 2 | 0 | data 部分的长度，高位在前低位在后 |
| 5 | crc8 | 1 |  | 包括 mask + cmd + len + data 的crc8 校验，算法C code见附录 |

**0x81 BT回应当前 TF 卡状态(需要BT确认)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字节序 | 项 | 长度（byte） | 值 | 说明 |
| 0 | mask | 1 |  | 协议头，目前固定为 0x51 |
| 1 | cmd | 1 | 0x81 | 命令 |
| 2~3 | len | 2 | 13 | data 部分的长度，高位在前低位在后 |
| 4 | status | 1 | 1 | TF 卡状态：  0->无卡  1->有卡  2->卡异常 |
| 4~8 | total | 4 | 0xffff | 卡总容量，单位为MB |
| 9~12 | free | 4 | 123 | 可用容量，单位为MB |
| 13~16 | code | 4 | 9527 | 状态码 |
| 17 | crc8 | 1 |  | 包括 mask + cmd + len + data 的crc8 校验，算法C code见附录 |

**附录**

**crc8 c code**

|  |
| --- |
| C++ unsigned char CRC8Table[]={  0, 94, 188, 226, 97, 63, 221, 131, 194, 156, 126, 32, 163, 253, 31, 65,  157, 195, 33, 127, 252, 162, 64, 30, 95, 1, 227, 189, 62, 96, 130, 220,  35, 125, 159, 193, 66, 28, 254, 160, 225, 191, 93, 3, 128, 222, 60, 98,  190, 224, 2, 92, 223, 129, 99, 61, 124, 34, 192, 158, 29, 67, 161, 255,  70, 24, 250, 164, 39, 121, 155, 197, 132, 218, 56, 102, 229, 187, 89, 7,  219, 133, 103, 57, 186, 228, 6, 88, 25, 71, 165, 251, 120, 38, 196, 154,  101, 59, 217, 135, 4, 90, 184, 230, 167, 249, 27, 69, 198, 152, 122, 36,  248, 166, 68, 26, 153, 199, 37, 123, 58, 100, 134, 216, 91, 5, 231, 185,  140, 210, 48, 110, 237, 179, 81, 15, 78, 16, 242, 172, 47, 113, 147, 205,  17, 79, 173, 243, 112, 46, 204, 146, 211, 141, 111, 49, 178, 236, 14, 80,  175, 241, 19, 77, 206, 144, 114, 44, 109, 51, 209, 143, 12, 82, 176, 238,  50, 108, 142, 208, 83, 13, 239, 177, 240, 174, 76, 18, 145, 207, 45, 115,  202, 148, 118, 40, 171, 245, 23, 73, 8, 86, 180, 234, 105, 55, 213, 139,  87, 9, 235, 181, 54, 104, 138, 212, 149, 203, 41, 119, 244, 170, 72, 22,  233, 183, 85, 11, 136, 214, 52, 106, 43, 117, 151, 201, 74, 20, 246, 168,  116, 42, 200, 150, 21, 75, 169, 247, 182, 232, 10, 84, 215, 137, 107, 53 };  unsigned char CRC8\_Table(unsigned char \*pucFrame, char usLen) {  unsigned char crc8 = 0;  while(usLen--)  crc8 = CRC8Table[crc8^\*(pucFrame++)];  return(crc8); } |