**送片机与上位机通信协议**

1. 通信数据类型
   1. 通信电气接口：RS232
   2. 通信波特率：57600
   3. 数据格式：8N1 （16进制数据）
2. 协议格式：

|  |  |  |  |
| --- | --- | --- | --- |
| 包格式 | 长度 | 值 | 说明 |
| 包头 | 2字节 | 0xEB90 |  |
| 包长度 | 1字节 | 0x00-0xff | 从DIR到CRC |
| 数据方向（DIR） | 1字节 | 0x00-0xff | 见附录一 |
| 命令（CMD） | 1字节 | 0x00-0xff | 见附录二 |
| 数据域 | N字节 |  | 见附录三 |
| CRC校验 | 2字节 | 0x0000-0xffff | CRC16/IBM从包长度到数据域。见附录四 |

协议约定在进行多字节数据传输时低字节数据在前，高字节数据在后。

1. 通信协议
   1. 上位机发送命令格式

|  |  |  |
| --- | --- | --- |
| 字节序号 | 发送内容 | 备注 |
| 0,1 | 包头 | 0xEB90 |
| 2 | 包长度 |  |
| 3 | 数据方向 | DIR = 0x00 |
| 4 | 命令 | CMD |
| (N) | 数据内容 |  |
| 4+N+1, 4+N+2 | 校验码 | CRC16/IBM检验，低位在前 |

* 1. 送片机应答数据

|  |  |  |
| --- | --- | --- |
| 字节序号 | 发送内容 | 备注 |
| 0,1 | 包头 | 0xEB90 |
| 2 | 包长度 |  |
| 3 | 数据方向 | DIR = 0x01 |
| 4 | 命令 | CMD |
| (N) | 数据内容 |  |
| 4+N+1, 4+N+2 | 校验码 | CRC16/IBM检验，低位在前 |

1. 附录一
   1. 数据方向：DIR = 0x00，数据方向从上位机到送片机
   2. 数据方向：DIR = 0x01，数据方向从送片机到上位机
2. 附录二

|  |  |
| --- | --- |
| 代码（CMD） | 内容 |
| 0x00 | 无操作 |
| 0x01 | 硬件初始化操作（含校准） |
| 0x02 | 机械复位操作/暂停（归0） |
| 0x03 | 获取片盒的状态 |
| 0x04 | 取片操作 |
| 0x05 | 还片操作 |
| 0x06 | 获取当前片盒 |
| 0x07 | 切换片盒 |
| 0x08 | 状态信息上报（包含错误信息，以及开机状态信息） |
| 其他 | 扩展 |

1. 附录三，数据域说明
   1. 上位机到送片机
      1. 通用常规命令

下表代表上位机请求“初始化操作”命令

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 包头 | | LEN | DIR | CMD | CRC校验 | |
| 0x90 | 0xEB | 0x04 | 0x00 | 0x01 | CRC1 | CRC2 |

* + 1. 在CMD为“还片操作”需要填充，长度为一字节，Value=需要选取的

下表代表上位机请求取2号载玻片

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 包头 | | LEN | DIR | CMD | 数据域 | CRC校验 | |
| 0x90 | 0xEB | 0x05 | 0x00 | 0x05 | 0x00-0xFF | CRC1 | CRC2 |

备注：数据域填充“0”换到上次取片槽位，其他数据则还到指定位置。

* + 1. 在CMD为“取片操作”需要填充，长度为一字节，Value=需要选取的片。

下表代表上位机请求取2号载玻片

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 包头 | | LEN | DIR | CMD | 数据域 | CRC校验 | |
| 0x90 | 0xEB | 0x05 | 0x00 | 0x04 | 0x02 | CRC1 | CRC2 |

* 1. 送片机到上位机。所有上位机请求的命令，送片机均返回2次。第一次为命令已收到返回，第二次为执行结果返回。
     1. 返回指令（CMD）执行状态，长度为两字节，数据域1为状态Value=0x00（成功），Value=0x01（失败），Value=0x02（命令已收到）。数据域2为取还片等指令失败原因。

下表代表送片机返回“获取片盒的状态”命令已收到

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 包头 | | LEN | DIR | CMD | 数据域1 | 数据域2 | CRC校验 | |
| 0x90 | 0xEB | 0x06 | 0x01 | 0x03 | 0x02 | 0x00-0xFF | CRC1 | CRC2 |

下表代表送片机返回“还片成功”应答

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 包头 | | LEN | DIR | CMD | 数据域1 | 数据域2 | CRC校验 | |
| 0x90 | 0xEB | 0x06 | 0x01 | 0x05 | 0x00 | 0x00-0xFF | CRC1 | CRC2 |

备注：数据域二为错误具体原因。

|  |  |  |
| --- | --- | --- |
| 位号 | 说明 | 备注 |
| BIT0 | 正在取片过程 |  |
| BIT1 | 正在还片过程 |  |
| BIT2 | 载玻台有玻片 |  |
|  |  |  |
|  |  |  |

* + 1. 当CMD为“获取片盒”。

下表代表返回载玻片，数据域1，2同上（6.2.1），数据域3为片盒状态。

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 包头 | | LEN | DIR | CMD | 数据域1 | 数据域2 | 数据域3 | CRC校验 | |
| 0x90 | 0xEB | 0x07 | 0x01 | 0x03 | 0x00 | 0x00-0xFF | 0x00-0xFF | CRC1 | CRC2 |

* + 1. 当CMD为“获取片盒状态”命令时，数据域1，2同上（6.2.1），在应答状态后加16字节玻片状态，当Bit = 0，该位置无玻片，Bit = 1，该位置有玻片。

下表代表返回载玻片

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 包头 | | LEN | DIR | CMD | 数据域1 | 数据域2 | 数据域3（16B） | CRC校验 | |
| 0x90 | 0xEB | 0x16 | 0x01 | 0x03 | 0x00 | 0x00-0xFF | 载玻片状态 | CRC1 | CRC2 |

* 1. 状态信息上报
     1. 状态信息上报格式：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 包头 | | LEN | DIR | CMD | 数据域 | CRC校验 | |
| 0x90 | 0xEB | 0x05 | 0x01 | 0x08 | 0x00-0xFF | CRC1 | CRC2 |

数据域具体内容

|  |  |  |
| --- | --- | --- |
| 位号 | 说明 | 备注 |
| BIT0 | 夹头上有玻片 | 等待INIT指令 |
| BIT1 | 载玻台有玻片 | 等待INIT指令 |
| BIT2 | X轴过流 | 等待断电重启 |
| BIT3 | Y轴过流 | 等待断电重启 |
| BIT4 | Z轴过流 | 等待断电重启 |
| BIT5 | 其他错误 | 待定 |
|  |  |  |

说明：若上电后夹头没有玻片以及载玻台没有玻片时，系统自动归0并校准。当夹头有玻片或者载玻台有玻片时，以1S的频率发送状态信息，直到收到初始化指令。

1. 附录四

// CRC校验公式: x^16 + x^15 + x^2 + 1。（CRC16/IBM）

const uint16 crc16\_table[256] = {

0x0000, 0xC0C1, 0xC181, 0x0140, 0xC301, 0x03C0, 0x0280, 0xC241,

0xC601, 0x06C0, 0x0780, 0xC741, 0x0500, 0xC5C1, 0xC481, 0x0440,

0xCC01, 0x0CC0, 0x0D80, 0xCD41, 0x0F00, 0xCFC1, 0xCE81, 0x0E40,

0x0A00, 0xCAC1, 0xCB81, 0x0B40, 0xC901, 0x09C0, 0x0880, 0xC841,

0xD801, 0x18C0, 0x1980, 0xD941, 0x1B00, 0xDBC1, 0xDA81, 0x1A40,

0x1E00, 0xDEC1, 0xDF81, 0x1F40, 0xDD01, 0x1DC0, 0x1C80, 0xDC41,

0x1400, 0xD4C1, 0xD581, 0x1540, 0xD701, 0x17C0, 0x1680, 0xD641,

0xD201, 0x12C0, 0x1380, 0xD341, 0x1100, 0xD1C1, 0xD081, 0x1040,

0xF001, 0x30C0, 0x3180, 0xF141, 0x3300, 0xF3C1, 0xF281, 0x3240,

0x3600, 0xF6C1, 0xF781, 0x3740, 0xF501, 0x35C0, 0x3480, 0xF441,

0x3C00, 0xFCC1, 0xFD81, 0x3D40, 0xFF01, 0x3FC0, 0x3E80, 0xFE41,

0xFA01, 0x3AC0, 0x3B80, 0xFB41, 0x3900, 0xF9C1, 0xF881, 0x3840,

0x2800, 0xE8C1, 0xE981, 0x2940, 0xEB01, 0x2BC0, 0x2A80, 0xEA41,

0xEE01, 0x2EC0, 0x2F80, 0xEF41, 0x2D00, 0xEDC1, 0xEC81, 0x2C40,

0xE401, 0x24C0, 0x2580, 0xE541, 0x2700, 0xE7C1, 0xE681, 0x2640,

0x2200, 0xE2C1, 0xE381, 0x2340, 0xE101, 0x21C0, 0x2080, 0xE041,

0xA001, 0x60C0, 0x6180, 0xA141, 0x6300, 0xA3C1, 0xA281, 0x6240,

0x6600, 0xA6C1, 0xA781, 0x6740, 0xA501, 0x65C0, 0x6480, 0xA441,

0x6C00, 0xACC1, 0xAD81, 0x6D40, 0xAF01, 0x6FC0, 0x6E80, 0xAE41,

0xAA01, 0x6AC0, 0x6B80, 0xAB41, 0x6900, 0xA9C1, 0xA881, 0x6840,

0x7800, 0xB8C1, 0xB981, 0x7940, 0xBB01, 0x7BC0, 0x7A80, 0xBA41,

0xBE01, 0x7EC0, 0x7F80, 0xBF41, 0x7D00, 0xBDC1, 0xBC81, 0x7C40,

0xB401, 0x74C0, 0x7580, 0xB541, 0x7700, 0xB7C1, 0xB681, 0x7640,

0x7200, 0xB2C1, 0xB381, 0x7340, 0xB101, 0x71C0, 0x7080, 0xB041,

0x5000, 0x90C1, 0x9181, 0x5140, 0x9301, 0x53C0, 0x5280, 0x9241,

0x9601, 0x56C0, 0x5780, 0x9741, 0x5500, 0x95C1, 0x9481, 0x5440,

0x9C01, 0x5CC0, 0x5D80, 0x9D41, 0x5F00, 0x9FC1, 0x9E81, 0x5E40,

0x5A00, 0x9AC1, 0x9B81, 0x5B40, 0x9901, 0x59C0, 0x5880, 0x9841,

0x8801, 0x48C0, 0x4980, 0x8941, 0x4B00, 0x8BC1, 0x8A81, 0x4A40,

0x4E00, 0x8EC1, 0x8F81, 0x4F40, 0x8D01, 0x4DC0, 0x4C80, 0x8C41,

0x4400, 0x84C1, 0x8581, 0x4540, 0x8701, 0x47C0, 0x4680, 0x8641,

0x8201, 0x42C0, 0x4380, 0x8341, 0x4100, 0x81C1, 0x8081, 0x4040

};

uint16 CRC16(uint8\_t \*pBuf, int nLen)

{

uint16 CRC = 0;

int i;

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

{

CRC ^= (\*pBuf++) & 0x00FF;

CRC = crc16\_table[CRC & 0x00FF] ^ (CRC >> 8);

}

return CRC;

}

// crc在协议中低位在前，如下图所示

// \*pBuf++ = CRC & 0x00FF;

// \*pBuf++ = (CRC & 0xFF00) >> 8;

1. 附录五（外部设备控制建议）
2. 数据发出后等待返回,一般 2min后认为超时
3. 多次执行失败之后，可控制供电复位。
4. 该配套电路板采用单任务模式，即当前正在执行某一 CMD 情况下，不会响应其他 CMD，只有完成该 CMD 才可以执行下一条 CMD。

示例

1. 取一号片：90 eb 05 00 04 01 c3 cc

下位机启动流程:



上位机启动流程:

