

AOC 普通皮秒通讯

1) Serial port 设置：

Baud rate：9600

Data：8 bit

Parity：none

Stop：1bit

推荐设置 Timeout 1 秒， 命令发送间隔 500 毫秒

2) 命令格式：

命令读取：

发送 ?CMD\r\n

接收 DATA\r\n

- CMD 可为命令列表里的任意命令， 总是三位 string。请避免输入命令表以外的命令至系统；
- 数据读取成功时 DATA 总是 6 位 string， 返回不足 6 位数前面用 0 填充；
- 如果因为命令错误等原因读取不成功则返回的 DATA 为 ERROR， 5 位 string；
- 如果系统处于忙碌状态， 则返回的 DATA 为 BUSY， 4 位的 string；

例子：

发送?SHT\r\n, 系统返回 000001\r\n, 表示 shutter 处于打开状态。

发送?TEST\r\n, 系统返回 ERROR\r\n, 该命令无效。

发送?PRF\r\n, 系统返回 BUSY\r\n, 系统正处于忙碌状态， 请等待系统脱离该状态再

执行命令设置。

命令设置：

发送 CMD=DATA\r\n

接收 DATA\r\n

- 数据设置成功时 DATA 总是 6 位 string， 返回不足 6 位数前面系统会用 0 填充；
- 如果因为命令错误或设置值超出允许范围等原因设置不成功则返回的 DATA 为 ERROR， 5 位 string；
- 如果系统正处于忙碌状态， 则返回的 DATA 为 BUSY， 4 位 string， 请等待系统脱离忙碌状态再执行命令设置。
- MCH=1 命令是用于检测系统是否处于可更改命令设置的状态， 如果系统处于可更改命令设置状态， 系统会返回 000001；
- 特殊命令 ADC， 此命令为复合开关机命令， 如用此命令开机发送 ADC=1 之后要等待 6 分钟左右系统才返回 000001, 且在这段时间内不可进行其他命令设置。在实际应用中发送 ADC=1 之后请不要接收系统反馈， 等待 6 分钟之后发送 MCH=1 若收到系统返回 000001 再进行其他命令设置操作；

例子：

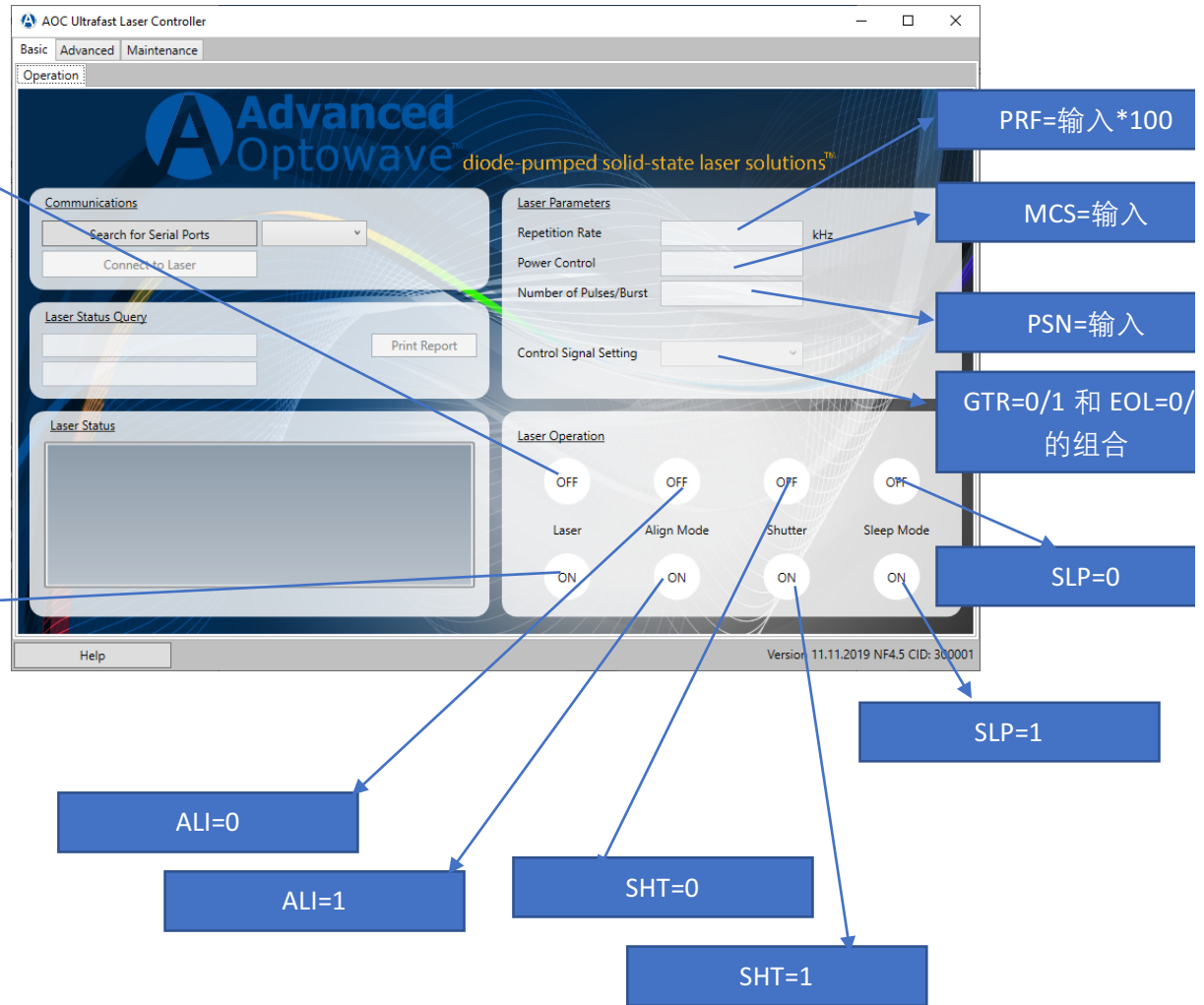
发送 SHT=0\r\n, 如果返回 000000\r\n, 表示系统收到了关闭 Shutter 的命令；

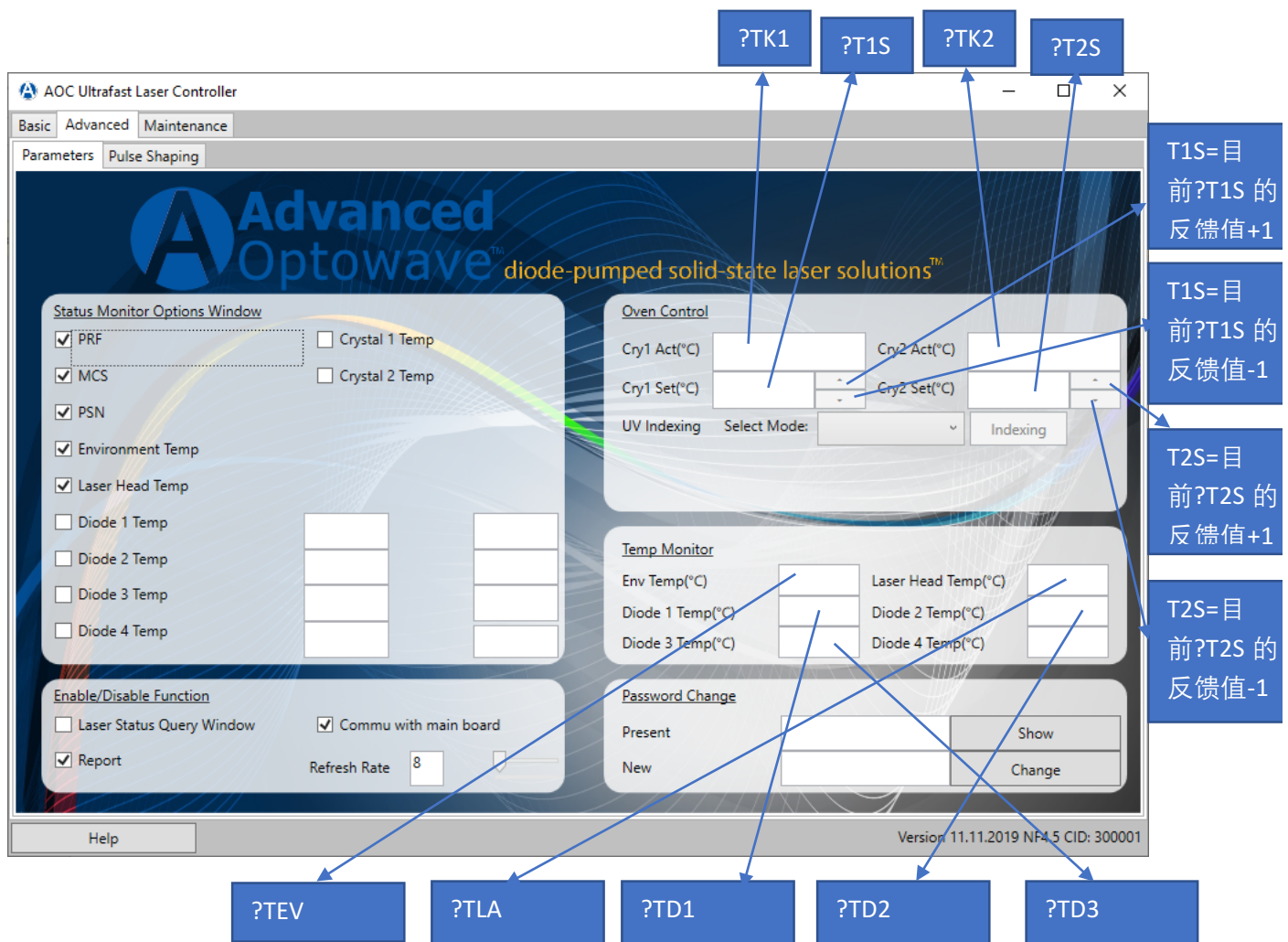
如果返回 ERROR\r\n, 表示系统因为某些原因没有执行设置 ;
如果返回 BUSY\r\n, 表示系统正处于忙碌状态, 用户需要等待系统
脱离忙碌状态再执行命令设置 ;

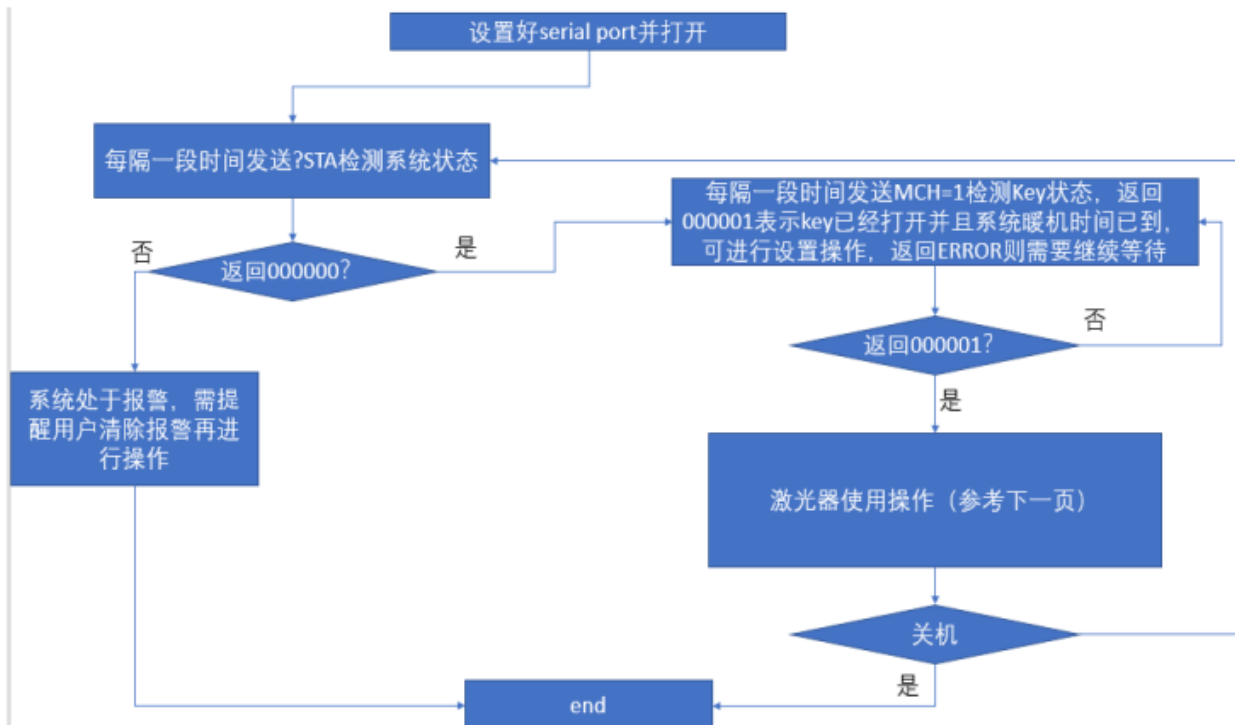
界面操作流程例子 :

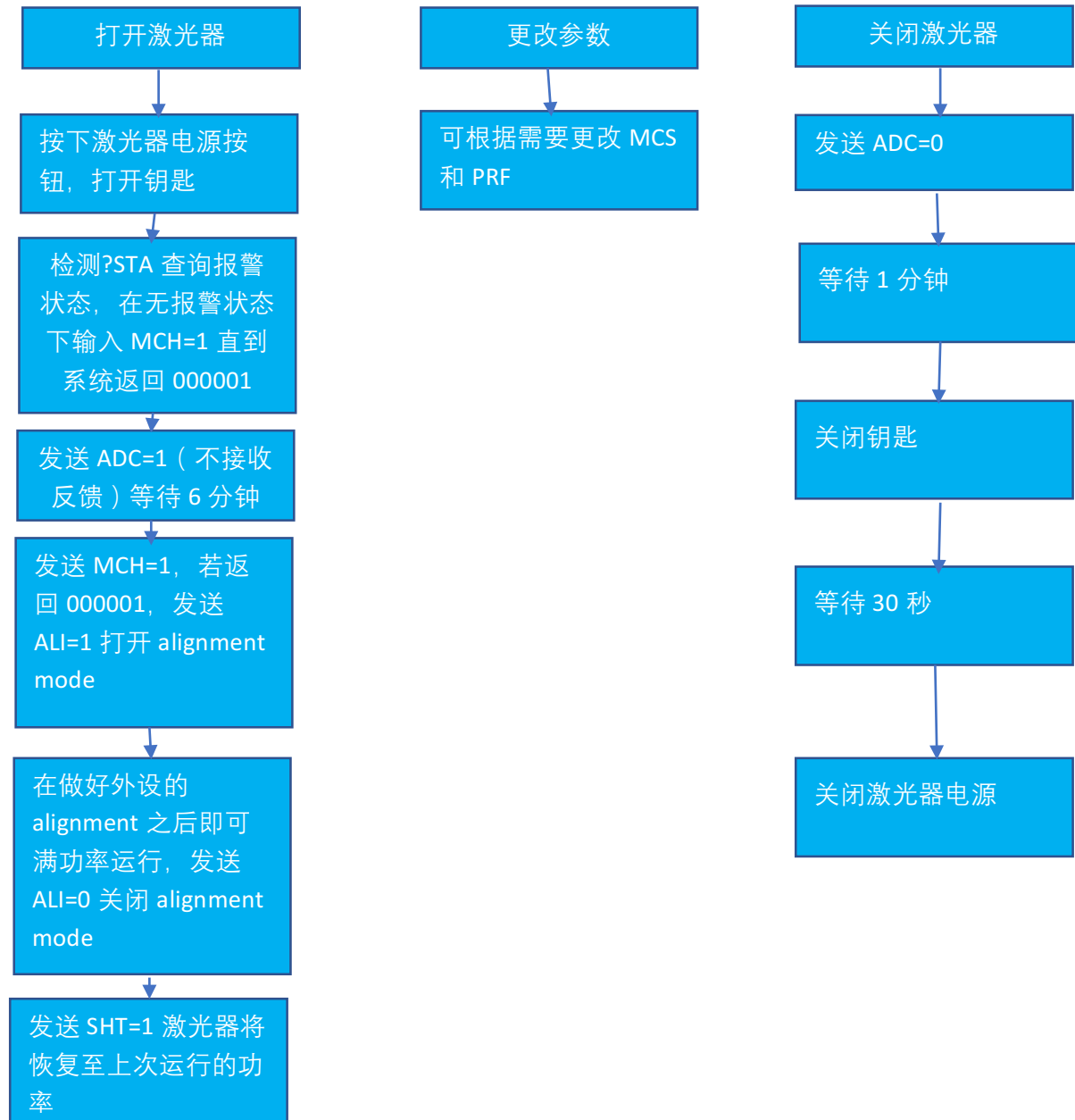
DC=0, 如果设置成功系统会马上反馈 000000, 但此后一分钟激光器处于逐渐关闭的过程, 分钟内不要发送任何更改设置的命令 (不发送含“=”号的命令)

DC=1, 复合命令, 不接收系统反馈, 发送后 6 分钟内不要发送任何更改设置的命令 (不发送含“=”号的命令)









系统报警代号（以最新命令列表信息为准）：

```
"System state: Normal", //0
"System state: ALARM CODE 1 DIODE1_TMP_HIGH",
"System state: ALARM CODE 2 DIODE1_TMP_LOW",
"System state: ALARM CODE 3 LASER_HEAD_TMP_HIGH",
"System state: ALARM CODE 4 LASER_HEAD_TMP_LOW",
"System state: ALARM CODE 5 ENVIRO_TMP_HIGH", //5
"System state: ALARM CODE 6 ENVIRO_TMP_LOW",
"System state: ALARM CODE 7 INTERLOCK_OPEN",
"System state: ALARM CODE 8 FIBER_ERROR",
"System state: ALARM CODE 9 CRYSTAL1_TMP_HIGH",
"System state: ALARM CODE 10 CRYSTAL1_TMP_LOW", //10
"System state: ALARM CODE 11 CRYSTAL2_TMP_HIGH",
"System state: ALARM CODE 12 CRYSTAL2_TMP_LOW",
"System state: ALARM CODE 13 DIODE2_TMP_HIGH",
"System state: ALARM CODE 14 DIODE2_TMP_LOW",
"System state: ALARM CODE 15 DIODE3_TMP_HIGH", //15
"System state: ALARM CODE 16 DIODE3_TMP_LOW",
"System state: ALARM CODE 17 CRYSTAL3_TMP_HIGH",
"System state: ALARM CODE 18 CRYSTAL3_TMP_LOW",
"System state: ALARM CODE 19 CRYSTAL4_TMP_HIGH",
"System state: ALARM CODE 20 CRYSTAL4_TMP_LOW", //20
"System state: ALARM CODE 21 PICKER_COM_ERR",
"System state: ALARM CODE 22 SEED_COM_ERR",
"System state: ALARM CODE 23 MODELOCK_LOST_ERR",
"System state: ALARM CODE 24 EMERGENCY_BTN",
"System state: ALARM CODE 25 SEED_LIFE_WARN", //25
"System state: ALARM CODE 26 PD1_VALUE_ABNORMAL",
"System state: ALARM CODE 27 DATA_LOST",
"System state: ALARM CODE 28 PD2_VALUE_ABNORMAL",
"System state: ALARM CODE 29 MOTION_COM_ERR",
"System state: ALARM CODE 30 OVEN_TEMP_ABNORMAL", //30
"System state: ALARM CODE 31 COMBINATION_CMD__PROC_ERR",
"System state: ALARM CODE 32 SYSTEM_FIRMWARE_MISMATCH",
"System state: ALARM CODE 33 HARMONIC_CRYSTAL_LIFETIME_REACHED"
```

命令发送示例

```
private string ProcCMD(string cmd)
{
    string temp;
    try
    {
        _serialPort.DiscardOutBuffer();
        _serialPort.DiscardInBuffer();
        _serialPort.Write(cmd + "\r\n");
        temp = _serialPort.ReadLine().ToString();
        if (temp.Substring(0, 4) == "BUSY")
        {
            temp = "BUSY";
        }
        else if (temp.Substring(0, 5) == "ERROR")
        {
            temp = "ERROR";
        }
        else
        {
            temp = temp.Substring(0, 6);
        }
    }
    catch
    {
        temp = "PRCERR";
    }
    return temp;
}
```