

极目 W718LC/LS 视频图像压缩模块编程指南

V1.6 2008-03-26

| | |
|---------------------------------------|----|
| I 通讯接口 | 2 |
| 1.1 异步串行主机通讯接口 | 2 |
| II 主机指令 | 3 |
| 2.1 控制指令格式 | 3 |
| 2.2 < Encoder_Setup > 指令DATA域解释 | 4 |
| 2.3 < SetComBaud > 指令DATA域解释 | 5 |
| 2.4 < ReadOnePic > 指令DATA域解释 | 5 |
| 2.5 指令字节流示例: | 5 |
| III 输出码流格式 | 7 |
| 3.1 极目W718LC的CAMEC编码P帧累积误差修正 | 7 |
| 3.2 编码数据帧及其子数据帧 | 7 |
| IV 解码 | 9 |
| 4.1 极目W718LC输出码流的解码 | 9 |
| 4.2 极目W718LS输出码流的解码 | 10 |
| V 编程要点 | 11 |

I 通讯接口

主机在极目 W718L 模块复位后以 9600bps 速率向其发出串口速率设置指令<SetComBaud>，将其串口速率设置为预先设计好的工作速率，主机再将自己的串口速率改为与极目 W718L 模块相一致就可以开始正常通讯了。

1.1 异步串行主机通讯接口

极目 W718 压缩模块的异步串行主机通讯接口的速率范围是 9600-230400bps，格式是 8-N-1，TTL 电平，支持全双工模式，收发使用相同的波特率。该异步通讯接口共有三根信号线：RXD/TXD/CTS，RXD 数据为输入信号，TXD 为数据输出信号，CTS 为流量控制输入信号。

极目 W718 压缩模块的数据输出即码流输出采用 CTS/RTS 硬件流量控制，极目 W718 压缩模块的 CTS 输入信号端接收主机的 RTS 输出控制信号。当极目 W718 压缩模块在 CTS 上检测到主机输出的允许发送信号（低电平）时就将压缩编码码流输出给主机；当在 CTS 上检测到主机输出的暂停发送信号（高电平）时就暂停压缩编码码流的输出。这个 CTS/RTS 硬件流量控制机制非常类似于交通中的红绿灯机制，对于视频传输特别是无线视频传输非常有用，主机可以根据信道拥塞误码以及通讯缓冲区占用率等情况来随机决定压缩模块的数据输出流量，在保证通讯可靠性的前提下最大限度地利用通讯信道的带宽。

主机向极目 W718 压缩模块发送数据无流量控制机制。

当极目 W718 压缩模块复位后异步通讯口的速率被置为 9600bps，主机通过向极目 W718 的异步通讯口写入专门指令来改变端口速率。可以选择的端口波特率有：9600、19200、38400、57600、115200、230400。

II 主机指令

控制主机可通过以下 5 条指令来控制极目 W718LC/LS 视频压缩编码板：

- < VEnc_Start > : 启动一个连续的 JPEG 压缩编码序列
- < VEnc_Start1F > : 启动单帧 JPEG 压缩编码
- < VEnc_StartCAMEC > : 启动一个以 I 帧开始的 CAMEC 压缩编码帧序列 (W718LC 特有)
- < VEnc_Stop > : 停止编码，在当前帧结束后关闭码流输出
- < VEnc_Setup > : 设定编码器参数
- < SetComBaud > : 设定极目 W718 异步串行通讯口的波特率
- < GetROMVersion > : 读出极目 W718 模块的固件版本信息

以下 5 条指令为 JPEG 抓拍保存指令，为 W718LS（固件 V1.00 以上）所特有：

- < GetPicCount > : 读取抓拍 JPEG 图片计数器
- < ClrPicCount > : JPEG 抓拍计数器清零
- < ShootOnePic > : 抓拍并缓存一张 JPEG 图片（抓拍后不向串口输出数据）
- < ReadOnePic > : 读取一张 JPEG 图片
- < GetFreeMem > : 读取可用存储容量

2.1 控制指令格式

| | | | |
|------------|--------|------------------------------|------------|
| HEADER | (4 字节) | : 0xaa, 0x55, 0xaa, 0x55 | 前缀码 |
| CODE | (1 字节) | : = 0x01/0x0e/0x0f/0x20/0x30 | 指令码 |
| | | 0x01 = < VEnc_Start > | |
| | | 0x02 = < VEnc_Start1F > | |
| | | 0x03 = < VEnc_StartCAMEC > | |
| | | 0x0e = < VEnc_Stop > | |
| | | 0x0f = < VEnc_Setup > | |
| | | 0x20 = < SetComBaud > | |
| | | 0x30 = < GetROMVersion > | |
| | | 0x10 = < GetPicCount > | |
| | | 0x11 = < ClrPicCount > | |
| | | 0x12 = < ShootOnePic > | |
| | | 0x13 = < ReadOnePic > | |
| | | 0x14 = < GetFreeMem > | |
| CODE_CHK | (1 字节) | : = 0xff - CODE | 指令码取反校验 |
| HANDLE | (1 字节) | : = 0x00 | 保留/未用 |
| HANDLE_CHK | (1 字节) | : = 0xff | 保留/未用 |
| LEN | (1 字节) | : = 0-255 | 数据域长度 |
| | | < VEnc_Start > | : LEN = 0 |
| | | < VEnc_Start1F > | : LEN = 0 |
| | | < VEnc_StartCAMEC > | : LEN = 0 |
| | | < VEnc_Stop > | : LEN = 0 |
| | | < VEnc_Setup > | : LEN = 16 |
| | | < SetComBaud > | : LEN = 4 |
| | | < GetROMVersion > | : LEN = 0 |
| | | < GetPicCount > | : LEN = 0 |
| | | < ClrPicCount > | : LEN = 0 |
| | | < ShootOnePic > | : LEN = 0 |
| | | < ReadOnePic > | : LEN = 1 |
| | | < GetFreeMem > | : LEN = 0 |

| | | | |
|----------|------------|----------------|-----------|
| LEN_CHK | (1 字节) | : = 0xff - LEN | 数据域长度取反校验 |
| DATA | (1-255 字节) | : | 数据域 |
| DATA_CHK | (1 字节) | : | 数据域校验和 |

备注:

- 1) 当 LEN=0, 指令字节序列中不含 DATA 和 DATA_CHK 域。
- 2) DATA_CHK 等于将 DATA 域所有字节累加得到的值的最低字节,
即: DATA_CHK = SUM(DATA_1, DATA_2, ..., DATA_n) AND 0xff

2.2 < Encoder_Setup > 指令 DATA 域解释

| | | |
|------------|--------------------|--|
| CHANNEL | (1 字节) : = 0 - 3 | 视频输入通道选择 0..3 = 通道 1..4 |
| COLOR | (1 字节) : = 0, 1, 2 | 0 = 黑白编码 (CAMEC-I 或者 JPEG) 1 = 彩色编码 (CAMEC-I 或者 JPEG) 2 = 彩色编码 (CAMEC-II) |
| QUALITY | (1 字节) : = 0 - 255 | 压缩质量选择 0 - 99, 0=最差, 99=最优 |
| ERROR | (1 字节) : = 0 - 255 | 压缩编码误差修正 W718LC: 0 = 无误差修正 1..254 = 在输出 1..254 个 P 帧后 自动插入一个 I 帧 255 = P 帧帧内自动修正 W718LS: 未用 |
| RESERVED | (2 字节) : = 0 | 保留/未用 |
| W | (1 字节) : = 8 - 128 | W x 8 = 编码图像宽度, 以像素为单位 |
| H | (1 字节) : = 8 - 128 | H x 8 = 编码图像高度, 以像素为单位 |
| BRIGHT | (1 字节) : = 0 - 99 | 亮度 |
| CONTRAST | (1 字节) : = 0 - 99 | 对比度 |
| SATURATION | (1 字节) : = 0 - 99 | 色饱和度 |
| RESERVED | (5 字节) : = 0 | 保留/未用 |

备注:

- 1) W718LC 模块支持两种连续视频压缩编码格式: CAMEC-I 和 CAMEC-II。

| | CAMEC-I | CAMEC-II |
|-------|---------|----------|
| 压缩模式 | 连续帧序列 | 连续帧序列 |
| 压缩倍数 | 中等 | 高 |
| 图像质量 | 高 | 中等 |
| 最高分辨率 | 704x576 | 352x288 |
| 彩色编码 | 支持 | 支持 |
| 黑白编码 | 支持 | 不支持 |

使用<VEnc_StartCAMEC>指令启动压缩编码器时, 之前通过<VEnc_Setup>指令设定的 Color 参数值决定使用何种编码方式:

- Color = 0 : CAMEC-I 黑白编码
- Color = 1 : CAMEC-I 彩色编码
- Color = 2 : CAMEC-II 彩色编码

2) W, H 值决定编码图像的大小, 可有以下选择:

| W718LC/LS | W | H | 图像格式 |
|-----------|----|----|---------|
| | 88 | 72 | 704x576 |
| | 80 | 60 | 640x480 |
| | 44 | 36 | 352x288 |
| | 40 | 30 | 320x240 |
| | 22 | 18 | 176x144 |

* W718LC/LS 的 W/H 值可在 64-704/64-576 之间任意取 16 的倍数。

3) 上电后的缺省设置值:

| | | | |
|-----------|------------|-----|----------------|
| W718LC/LS | CHANNEL | 0 | |
| | COLOR | 1 | (CAMEC-I 编码器) |
| | QUALITY | 70 | |
| | ERROR | 200 | (只对 W718LC 有效) |
| | W | 40 | |
| | H | 30 | |
| | BRIGHT | 55 | |
| | CONTRAST | 55 | |
| | SATURATION | 80 | |

2.3 < SetComBaud > 指令 DATA 域解释

| | | |
|------|--------------------------|--|
| BAUD | (4 字节) : = 9600 - 230400 | 异步通讯串口波特率 高位字节在前, 例如 230400bps, 16 进制数值为 0x00038400, 则指令数据域的 4 个字节依次为: 0x00, 0x03, 0x84, 0x00 |
|------|--------------------------|--|

2.4 < ReadOnePic > 指令 DATA 域解释

| | | |
|----------|---------------------|--|
| PicIndex | (2 字节) : = 0 - 1000 | JPEG 图片索引 高位字节在前, 例如 PicIndex= 300, 16 进制数值为 0x012c, 则指令数据域的 2 个字节依次为: 0x01, 0x2c |
|----------|---------------------|--|

2.5 指令字节流示例:

< VEnc_Start > : (0xaa, 0x55, 0xaa, 0x55,
0x01, 0xfe, 0x00, 0xff, 0x00, 0xff
)

< VEnc_Start1F > : (0xaa, 0x55, 0xaa, 0x55,
0x02, 0xfd, 0x00, 0xff, 0x00, 0xff
)

```
< VEnc_Stop > : ( 0xaa, 0x55, 0xaa, 0x55,
                  0x0e, 0xf1, 0x00, 0xff, 0x00, 0xff
                )

< VEnc_Setup > : CHANNEL = 0, COLOR = 2 (CAMEC-II) , QUALITY = 70, ERROR = 255,
                  W = 40, H = 30,
                  BRIGHT=50, CONTRAST=50, SATURATION = 80
                  ( 0xaa, 0x55, 0xaa, 0x55,
                    0x0f, 0xf0, 0x00, 0xff, 0x10, 0xef,
                    0x00, 0x02, 0x46, 0xff,
                    0x00, 0x00,
                    0x28, 0x1e,
                    0x32, 0x32, 0x50,
                    0x00, 0x00, 0x00, 0x00, 0x00,
                    0x41
                  )

< SetComBaud > : BAUD = 230400
                  (
                    0xaa, 0x55, 0xaa, 0x55,
                    0x20, 0xdf, 0x00, 0xff, 0x04, 0xfb,
                    0x00, 0x03, 0x84, 0x00,
                    0x87
                  )

< GetROMVersion > : (
                    0xaa, 0x55, 0xaa, 0x55,
                    0x30, 0xcf, 0x00, 0xff, 0x00, 0xff
                  )

< GetPicCount > : (
                    0xaa, 0x55, 0xaa, 0x55,
                    0x10, 0xef, 0x00, 0xff, 0x00, 0xff
                  )

< ClrPicCount > : (
                    0xaa, 0x55, 0xaa, 0x55,
                    0x11, 0xee, 0x00, 0xff, 0x00, 0xff
                  )

< ShootOnePic > : (
                    0xaa, 0x55, 0xaa, 0x55,
                    0x12, 0xed, 0x00, 0xff, 0x00, 0xff
                  )

< ReadOnePic > : PicIndex = 300
                  (
                    0xaa, 0x55, 0xaa, 0x55,
                    0x13, 0xec, 0x00, 0xff, 0x02, 0xfd,
                    0x01, 0x2c, 0x2d
                  )

< GetFreeMem > : (
                    0xaa, 0x55, 0xaa, 0x55,
                    0x14, 0xeb, 0x00, 0xff, 0x00, 0xff
                  )
```

III 输出码流格式

极目 W718LC 中采用的 CAMEC 编码算法是一种视频序列压缩算法,输出的视频压缩编码帧序列一般形式: IPP...PPIPP...PPIPP...。 “I”指帧内压缩编码帧,“P”指帧间预测编码帧。

3.1 极目 W718LC 的 CAMEC 编码 P 帧累积误差修正

一般来说 P 帧序列会导致误差的传递和累积,为减弱这种误差传递累积效应,可根据需要通过对 VEnc_Setup 指令中的 ERROR 参数设定不同数值来选择多种修正方式。极目 W718LC 可提供以下三种误差修正方式:

- 1) 指令修正: ERROR = 0, 用户在收到一定数量的 P 帧后可通过发送<VEnc_StartCAMEC> 指令强行插入一个 I 帧来达到误差修正的目的。
- 2) 自动 I 帧修正: ERROR = 1..254, 极目 W718LC 在输出与 ERROR 值相同的 P 帧后自动插入一个 I 帧来达到误差修正的目的。
- 3) 自动 P 帧修正: ERROR = 255, 极目 W718LC 在 P 帧中自动插入修正数据来达到误差修正的目的。

极目 W718LC/LS 中采用 JPEG 帧内压缩算法,不需要进行累积误差修正。

3.2 编码数据帧及其子数据帧

极目 W718LC/LS 输出的每个 CAMEC I/P 编码帧或 JPEG 编码帧在输出时被分割并重新包装成多个长度不超过 255 字节的子数据帧构成,每个编码帧最多可被分割重包装成不超过 65535 个子数据帧,编码帧被分割的各部分编码数据被分别依次封装在各子数据帧的数据域(DATA)中。另外一些特殊类型的子数据帧被用来向主机报告状态信息。

[子数据帧格式]

| | | | |
|--------|------------|--------------------------|--|
| HEADER | (4 字节) | : 0x55, 0xaa, 0x55, 0xaa | 前缀码 |
| TYPE | (1 字节) | : | 数据帧类型标识 |
| | | | 0x80 = 视频编码子数据帧 |
| | | | 0x90 = 视频编码帧结束标记帧 |
| | | | 0x91 = 视频编码帧意外或强行终止标记帧 |
| | | | 0xa0 = 状态帧: 无视频信号输入 |
| | | | 0xb0 = 状态帧: 编码器故障 |
| | | | 0xc0 = 编码器固件版本信息 |
| | | | 0x10 = JPEG 抓拍计数器数值 |
| | | | 0x14 = 剩余的可用抓拍存储器空间(字节数) |
| INDEX | (2 字节) | : = 0 - 65535 | 子数据帧索引 高位字节在前, 如 1000 = 0x03, 0xe8 |
| LEN | (1 字节) | : | 数据域长度 |
| | | | TYPE = 0x80/0xc0, LEN= 0 - 255 |
| | | | TYPE = 0x90/0x91/0xa0/0xb0, LEN = 0 |
| | | | TYPE = 0x10, LEN = 2 |
| | | | TYPE = 0x14, LEN = 4 |
| DATA | (0-255 字节) | : | 数据域 |
| CHK | (1 字节) | : | 校验和。包括 TYPE/COUNT/LEN /DATA |

注:

- 1) 一个视频编码帧的在输出时被分割并重新包装成多个编码帧子数据帧和一个结束标记子数据帧构成,一般形式为:(编码帧子数据帧 1)(编码帧子数据帧 2)...(编码帧子数据帧 n)(编码帧结束标记帧)。
- 2) “INDEX”域是构成一个视频编码帧的各子数据帧的顺序计数。例如:一个视频编码帧由

- 20 个编码子数据帧构成，则依次输出的各编码子数据帧的 INDEX 域值为 0-19，结束标记子数据帧的 INDEX 域值为 20。程序员可利用此域值来检查子数据是否缺失，或在多径传输信道上进行排序恢复。
- 3) 状态帧只用来传递编码板的状态信息，有可能插入在任意两编码子数据帧之间。
 - 4) 编码器固件版本信息帧的 DATA 域包含一个 0x00 结尾的字符串，该字符串描述了模块的固件版本信息。
 - 5) 如果 LEN = 0, 则数据帧中不包括 DATA 域。
 - 6) 对于 TYPE=0x10/0x14 的字数据帧，DATA 域的字节顺序是高位 (MSB) 字节在前。

依照上述关于子数据帧的描述可知：一个完整的编码帧由各个子数据帧的数据域 (DATA) 部分按子数据帧索引顺序重新拼接而成，程序员可从极目 W718L 的输出数据流中恢复出各个视频帧的完整的编码帧，然后就可以解码还原出原视频帧图像。一个完整的视频编码帧遵循以下格式：

表 3.2

| | | | |
|-----------|----------|---|-------|
| CODE_TYPE | (1 字节) | : | 编码帧类型 |
| CODE_INFO | (1-6 字节) | : | 编码帧信息 |
| CODE | (N 字节) | : | 编码数据 |

注：在 JPEG 解码显示时只需提供 [CODE] 部分的数据。请参见“IV. 解码”一节中相关的说明。

[编码帧类型 CODE_TYPE] : 1 字节

- bit_7 : 最高位代表彩色或者黑白。0 代表黑白编码，1 代表彩色编码
 bit_6..0 : 编码器类型。0x41 代表 JPEG 压缩编码，0x40 代表 CAMEC-II 编码，如果是其他数值代表 CAMEC-I 编码。

[编码帧信息 CODE_INFO] : 1-6 字节

- JPEG 编码 : 长度=1 字节。
 内容=视频通道，0..3 代表视频输入通道 1..4

- CAMEC 编码 : 长度=6 字节
 [字节 1] = 保留
 [字节 2] = 视频通道，0..3 代表视频输入通道 1..4
 [字节 3] = X1 图像左边界座标
 [字节 4] = Y1 图像上边界座标
 [字节 5] = X2 图像右边界座标
 [字节 6] = Y2 图像下边界座标

注：被压缩视频的分辨率可以这样计算 $W=(X2-X1+1) \times 8$, $H=(Y2-Y1+1) \times 8$

IV 解码

4.1 极目 W718LC 输出码流的解码

极目 W718LC 输出压缩编码帧只能在 PC+Windows 环境下，通过调用极目 W718LC 软件开发包的解码库 (DLL) 中相应函数来完成。

解码库包含在随半模块提供的开发包中，解码库是名为<M7Codec.Dll>的动态链接库，一般需要拷贝到\Windows\System 或\Windows\System32 目录下。开发包中还提供了 VC 的示范代码和 VC 的导入库<M7Codec.Lib>和<M7Codec.h>。<M7Codec.Dll>、<M7Codec.Lib>、<M7Codec.h>这三个文件都包括在开发包的“解码库”子目录下。

下面列出可调用的解码库函数：

□ LONG WINAPI M7VID_DecodeRGB(unsigned char *CodBuf, unsigned char *ResBuf)

输入： unsigned char *CodBuf - 编码数据指针
unsigned char *ResBuf - 解码图像数据指针

返回值： 0 = 成功，非零 = 失败

输出： ResBuf - 返回解码图像数据

解释： 将 CodBuf 指向的编码数据解压缩还原成原始图像并拷贝至 ResBuf 指向的缓冲区。解码图像是 24bit RGB 格式，该格式的像素排列与 Bitmap 像素排列方式一致：(B0, B1, B2 - R, G, B)。解码后的图像帧的坐标系原点在图像矩形的左上角。CodBuf 中的数据包括 [\(表 3.2\)](#) 中的 [CODE_TYPE]、[CODE_INFO]、[CODE] 三个部分。

□ LONG WINAPI M7VID_CheckCodeFrame(unsigned char *CodBuf)

输入： unsigned char *CodBuf - 编码数据指针

返回值： 0 = I 帧，1=P 帧，小于零=失败

输出： 无

解释： 检查 CodBuf 指向的编码数据，如果正确返回 I、P 帧检测结果。CodBuf 中的数据包括 [\(表 3.2\)](#) 中的 [CODE_TYPE]、[CODE_INFO]、[CODE] 三个部分。**只适用于 CAMEC-I 编码帧。**

□ LONG WINAPI M7VID_CheckCodeFrameA(unsigned char *CodBuf, long CodLen)

输入： unsigned char *CodBuf - 编码数据指针
long CodLen - 编码数据帧长度（字节）

返回值： 0 = I 帧，1=P 帧，小于零=失败

输出： 无

解释： 检查 CodBuf 指向的编码数据，如果正确返回 I、P 帧检测结果。CodBuf 中的数据包括 [\(表 3.2\)](#) 中的 [CODE_TYPE]、[CODE_INFO]、[CODE] 三个部分。**只适用于 CAMEC-II 编码帧，解码前必须调用此函数检查 CAMEC-II 编码帧。如何判断编码帧类型请参见 [\(3.2\)](#)**

```

□ LONG WINAPI M7VID_DisplayImage( HWND hDispWnd,
                                   LONG X, LONG Y, LONG W, LONG H,
                                   LONG ImgClrBits,
                                   void *lpImgBuffer);

```

输入: HWND hDisplayWindow : 窗口句柄
 LONG X, Y : 显示图象原点（左上角）在窗口中的坐标。
 LONG W, H : 图象宽度和高度, ImgH 应为负值以适应 Windows 要求
 LONG ImgClrBits : 图象颜色比特数, 8=黑白图像, 24=RGB 彩色图像
 void *lpImgBuffer : 图象数据指针

返回值: 无

解释: 在指定窗口中显示图象

```

□ LONG WINAPI M7VID_DisplayImageZoom( HWND hDispWnd,
                                       LONG X, LONG Y,
                                       LONG DispW, LONG DispH,
                                       LONG W, LONG H,
                                       LONG ImgClrBits,
                                       void *lpImgBuffer);

```

输入: HWND hDisplayWindow : 窗口句柄
 LONG X, Y : 显示图象原点（左上角）在窗口中的坐标。
 LONG DispW, DispH : 显示图象宽度和高度
 LONG W, H : 图象宽度和高度, ImgH 应为负值以适应 Windows 要求
 LONG ImgClrBits : 图象颜色比特数, 8=黑白图像, 24=RGB 彩色图像。
 void *lpImgBuffer : 图象数据指针

返回值: 无

解释: 在指定窗口中显示图像, 图像大小适应窗口大小。

```

□ LONG WINAPI M7VID_SaveImgToBmpFile( CHAR *BmpFileName,
                                       void * ImgBuf,
                                       LONG Img_Width, LONG Img_Height,
                                       LONG Img_ClrBits);

```

输入: CHAR *BmpFileName : 文件名指针(null-terminated string)
 void *ImgBuf : 图像数据指针
 LONG Img_Width, Img_Height : 图像宽度和高度
 LONG Img_ClrBits : 图像颜色比特数, 8=黑白图像, 24=RGB 彩色图像。

返回值: 无

解释: 将图像帧按 BITMAP 格式文件存盘。

□ 如何计算 CAMEC 视频解码图像的分辨率:

视频的分辨率可以这样计算 $W=(X2-X1+1) \times 8$, $H=(Y2-Y1+1) \times 8$, 单位是像素。可参照 [\(3.2\)](#) 中关于 CODE_INFO 数据域的解释。

4.2 极目 W718LS 输出码流的解码

由于极目 W718LS 采用标准 JPEG 编码, 因此不需要特别提供解码函数。输出码流的 [CODE] 部分 ([表 3.2](#)) 符合 JFIF 标准, 只要将该部分保存成文件并命名成 “*.jpg” 形式, 就得到了一个标准的 JPEG 文件, 可以被许多软件和控件所使用。

由于 JPEG 格式的被广泛采用, 许多编程环境都提供了多种 JPEG 图像解码和显示的方式, 例如开发包 “W718LS-EVB” 目录的 W718LS 示范代码中利用 Windows 的 OLE 接口在 VC 中直接显示 JPEG 图像。VB 和 Delphi 中的图像控件都提供了对 JPEG 格式文件的支持。

V 编程要点

1) 如何进行连续帧压缩编码?

首先发送<Encoder_Setup>设置视频编码参数，然后发送<Encoder_Start>或<Encoder_StartCAMEC>指令，极目 W718L 就自动地开始进行连续压缩编码，极目 W718LC 编码帧序列一般形式为：IPP... PPIPP... PPIPP...，极目 W718LS 则连续输出 JPEG 编码。需要停止编码时发送<Encoder_Stop>指令即可。**建议设置编码参数时设置 Color=2，以选择 CAMEC-II 压缩编码以获得较高的压缩效率。**

2) 如何在极目 W718LC 上进行单帧压缩编码?

有两种方法：

i). 首先执行<Encoder_Setup>设置视频编码参数，然后发送一个<Encoder_Start>或<Encoder_StartCAMEC>指令启动压缩编码，在收到一个编码帧数据帧后发送一个<Encoder_Stop>指令，极目 W718LC/LS 将会立即回应一个类型为<0x91>的数据帧并在输出完当前的编码帧后自动停止编码输出。

ii). 首先执行<Encoder_Setup>设置视频编码参数，然后发送一个<Encoder_Start1F>指令启动单帧 JPEG 压缩编码。极目 W718LC/LS 会在输出完 JPEG 编码数据后自动停止。

3) 如何确定收到 W718LC 输出的编码帧是 I 帧还是 P 帧?

可以用 M7VID_CheckCodeFrame 函数来检查。

4) 如何控制编码输出流量?

如果采用并行通讯口，极目 W718L 的编码输出流量取决于主机的读出速率。

如果采用串行接口，应该首先按实际系统的构成和需求通过<SetComBaud>指令设定极目 W718 的串口的波特率，然后根据信道拥塞程度通过动态地控制 W718 的 CTS 端来达到动态控制编码流量的目的。可参见 [\(1.1 异步串行主机通讯接口\)](#)。

5) 如何将编码帧解压缩? 解码后的图像是什么格式的?

编码帧解压缩请参见 [“IV. 解码”](#) 一节。

极目 W718LC 的 CAMEC 编码数据解码后的图像是 24bit RGB 格式，该格式的像素排列与 Bitmap 像素排列方式一致：(B0, B1, B2 - R, G, B)。解码后的图像帧的坐标系原点在图像矩形的左上角。

6) 可以在非 Windows 环境下将编码帧解压缩吗?

CAMEC 编码目前还只能在 PC+Windows 环境下进行解码操作，在其他环境下只能进行传输和存储等应用操作。

JPEG 编码格式由于已经被广泛应用的，因此可以在任何支持 JPEG 格式的操作系统环境下解压缩显示。

7) 如何判断视频输入端是否有视频信号输入?

发送<VEnc_Start> 指令，如果收到编码帧而未接收到“无视频信号状态帧”，表示有有效的视频信号输入；如果接收到“无视频信号状态帧”（类型为 0xa0 的数据帧），表示无有效视频信号输入。如果在视频编码过程中突然失去有效视频输入信号，极目 W718L 将在结束当前编码帧的传送后，发送一个“无视频信号状态帧”，然后停止编码发送，进入待机状态。建议：每次在收到接收到“无视频信号状态帧”或其他错误状态信息帧后，延迟 1 秒钟后重新发送<VEnc_Start>指令，这样反复操作，可实现视频输入恢复后自动恢复视频编码输出。

8) 如何先进行连续多张图片抓拍? (728LS/JPEG)

在很多情况下，W718LS 受串口速度和用户处理器速度的双重的限制，不能做到快速连续抓拍多张 JPEG 图片。W718LS V1.00 以后的固件新增了图片抓拍的功能，基本工作过程是：W718LS 接到用户抓拍指令后立即抓拍一张图片并压缩成 JPEG 文件，压缩结果并不立即从串口输出给用户处理器而是先保存在 W718LS 的内存中。等抓拍到足够数量的 JPEG 图片后，用户程序可按需要以较慢速度读出。这样即使有串口速度和处理器速度的限制，也可以在短时间内快速抓拍大量图片。

极目 W718LS 用于保存抓拍图片的存储器容量为 2MB，掉电不保存。

下面简单示范一个外信号触发抓拍并读出过程：

```
{  
  
    < VEnc_Setup >      // 设置抓拍图片的分辨率、压缩质量等参数  
    < ClrPicCount >    // 清零抓拍计数器  
lp1:  
    if (外触发 == FALSE) goto lp1;  
    < ShootOnePic >  
    if (抓拍数量 < 预定数量){  
        抓拍数量 ++  
        goto lp1;  
    }  
  
    for (i=0;i<抓拍数量;i++)  
    {  
        < ReadOnePic (i)>  
    }  
}
```

在抓拍一张后 W718LS 会输出一个 TYPE=0x10 的数据子帧，通知用户处理器当前地 JPEG 抓拍计数器的数值。如果抓拍内存不够或读出索引超届，W718LS 会输出一个 TYPE=0xb0 的子数据帧通知用户处理器。