



苏州德亚车牌识别摄像机 动态库用户手册

(机密等级：1)

机密等级：

- 0 (公开)
- 1 (可提供客户和集成商)
- 2 (仅限于公司内部)
- 3 (仅限于公司内部指定人员)

苏州德亚交通技术有限公司

文件号：SU14h

版本：V1.8

日期：2017/04/10

制定：张振中

修改记录：

2014/10/31: 初版

2014/12/17: 1.1 版

- 1) 修改第一节“说明”里关于软触发的解释。配合识别机软件 1.2.6 版后增加双识别区功能。

2014/12/17: 1.2 版

- 1) 增加 2 个接口：
LPNR_EnableLiveFrame: 使能，禁能实时图像发送。
LPNR_TakeSnapFrame: 让识别机抓拍并且发送一帧最新图像但不做分析处理。

2015/01/15: 1.3 版

- 1) 增加 2 个接口：
LPNR_Lock: 锁住实时图像和识别结果数据图片。
LPNR_Unlock: 解锁实时图像和识别结果数据图

2015/01/15: 1.4 版

- 1) 增加 1 个接口：
LPNR_SyncTime: 同步计算机和识别机的时间。

2015/05/20: 1.5 版

- 1) 接口标准改为 CDECL:

2015/08/21: 1.6 版

- 1) 识别机 1.3.x.y 版本后已经取消双识别区功能，修改说明配合新版本识别机。

2016/11/08: 1.7 版

- 1) 增加以下接口函数
LPNR_SetCallBack 设置回调函数
LPNR_GetCameraLabel 获取识别机名称
LPNR_ResetHeartBeat 重置心跳帧时间
LPNR_GetExtDI 获取 DI 点状态（只有某些机型有额外的 DI 点）
LPNR_GetHeadImageSize 获取车头图大小
LPNR_GetHeadImage 获取车头图
LPNR_GetQuadImageSize 获取 1/4 解像度抓拍图大小
LPNR_GetQuadImage 获取 1/4 解像度抓拍图
LPNR_GetMachineIP 获取识别机 IP
LPNR_GetPlateAttribute 获取车牌/车辆额外属性讯息

2017/04/10: 1.8 版

- 1) 增加以下接口函数
LPNR_GetModelAndSensor 获取识别机机型和感光器件型号代码
LPNR_GetVersion 获取识别机程序版本号和识别算法版本号
LPNR_GetCapability 获取识别机硬件、内核、条件编译等功能字
LPNR_EnableLog 使能/禁能日志功能
LPNR_SetLogPath 设置日志路径
- 2) 所有接口改为动态库标准的 STDCALL

目 录

1	说明	1
2	动态库接口	4
3	函数接口说明	7
3.1	LPNR_Init	7
3.2	LPNR_SetWndMsg	7
3.3	LPNR_SetCallBack	8
3.4	LPNR_Terminate	8
3.5	LPNR_GetPlateNumber	8
3.6	LPNR_GetPlateColorImage	9
3.7	LPNR_GetPlateColorImageSize	9
3.8	LPNR_GetPlateBinaryImage	9
3.9	LPNR_GetPlateBinaryImageSize	10
3.10	LPNR_GetCapturedImage	10
3.11	LPNR_GetCapturedImageSize	11
3.12	LPNR_GetLiveFrame	11
3.13	LPNR_GetLiveFrameSize	12
3.14	LPNR_GetQuadImage	12
3.15	LPNR_GetQuadImageSize	12
3.16	LPNR_GetHeadImage	12
3.17	LPNR_GetHeadImageSize	13
3.18	LPNR_IsOnline	13
3.19	LPNR_ReleaseData	13
3.20	LPNR_SoftTrigger	14
3.21	LPNR_IsIdle	14
3.22	LPNR_GetTiming	14
3.23	LPNR_EnableLiveFrame	15
3.24	LPNR_TakeSnapFrame	15
3.25	LPNR_Lock	16
3.26	LPNR_Unlock	17
3.27	LPNR_GetPlateAttribute	17
3.28	LPNR_GetMachineIP	18
3.29	LPNR_GetExtDI	18
3.30	LPNR_GetCameraLabel	19
3.31	LPNR_SyncTime	19
3.32	LPNR_ResetHeartBeat	19
3.33	LPNR_GetEventText	20
3.34	LPNR_GetModelAndSensor	20

3.35	LPNR_GetVersion	20
3.36	LPNR_GetCapability	21
3.37	LPNR_SetLogPath.....	21
3.38	LPNR_EnableLog.....	22

表格目录

表格 1：函数列表.....4

1 说明

苏州德亚公司车牌识别使用一体机设备，具备低功耗、可靠性高、识别准确的特点。尤其适合使用于高速公路和停车场出入口。这些应用都是将识别摄像机安装在路测，和车牌有很大的视角（俯仰和摇摆角度）。识别机的算法会自动检测车牌角度进行矫正，提高识别结果。一般车牌都可以在 0.5 秒内识别完成。

识别机可以允许多上位机连接。安装时需使用我司的工具程序 LPNRTTool 调整焦距、角度、识别区域 (ROI)、识别参数、ISP 参数。这些调整好后就不需要更改。这些维护使用的功能并没有提供在动态库里面，动态库只提供日常检测车牌过程中上位机程序需要用到的功能。这样区分使用和维护可以让动态库简化，同时不容易因为动态库使用错误造成识别机设置错误的问题。

动态库启动后就自动和识别机进行 TCP 连接，内部会有工作线程，不停接收识别机的数据。如果有通信断线情况，工作线程会自动重连。

对于事件，动态库以 Windows 的消息机制发送给上层程序。事件只有以下 10 个：

- 识别机连线 (编号 1)
- 识别机离线 (编号 2)
- 识别处理开始 (编号 3)
- 识别处理结束，数据已接收 (编号 4)
- 收到一帧实时视频帧 (编号 5)
- 有车辆进入虚拟线圈¹ (编号 6)

¹ 虚拟线圈进入、离开事件暂时不支持。

- 车辆离开虚拟线圈 (编号 7)
- 识别机DI点有变化² (编号 8)
- 收到使用 LPNR_TakeSnapFrame 触发的抓拍图片(编号 9)
- 收到识别机版本讯息 (编号 10)

应用程序需在初始化识别机动态库后就立刻设置消息接收的窗口句柄以及消息编号。识别机动态库设计是可以让应用程序同时连接多个识别机,因此初始化函数会返回一个识别机句柄。后续针对某个识别机的操作都需要以句柄为第一个参数。对于消息通知,应用程序收到的消息参数 WPARAM 是识别机句柄(由哪台识别机发出),LPARAM 是消息编号。也就是说,上位机程序可以使用同一个消息处理函数来处理所有识别机的消息。

对于识别结果,应用程序在收到消息编号 4 (识别结束)时,才调用动态库接口去获取识别结果。识别结果有:

- 车牌号码(颜色+汉字+数字/英文),例如:"蓝苏 E12345"。如果没有识别出来,车牌号码字符串内容为" 整牌拒识"。前面颜色部分是两个空白字节。
- 抓拍图片:这是一个 jpeg 图,将收到的整个 buffer 内容写到一个文件里面就是一个完整的 jpeg 图档。
- 车牌彩色图:这是车牌区域的小图,为 BMP 格式。将收到的 buffer 内容完整写入文件里面就是一个 BMP 图档。
- 车牌二值图:这同样是车牌区域的图片,但是只有黑\白两色。一样是 BMP 格式。
- 抓拍图片 1/4 解像度图³:1/4 解像度的抓拍图。
- 抓拍图片车头图⁴:车头图是以车牌为中心,截取D1

² 只有具备扩展 DI 的识别机机型才会有这个事件。

³ 必须使用识别机配置工具使能 1/4 解像度图输出选项。

⁴ 必须使用识别机配置工具使能车头图输出选项。

(720x576)大小的图片，这样可以减小图片大小还能保持原始解像度的车牌清晰度。

- 车辆和车牌额外信息：可以提取可信度、车速、车身颜色等额外信息做为参考。

识别结果获取完后，应用程序调用释放识别结果的接口，将结果释放掉，回收内存。当然，不释放的话下个识别结果收到后一样会自动释放，再储存新的结果。

动态库提供“软触发”功能。当线圈因为跟车没有触发的时候，应用程序可以调用接口触发一次抓拍识别。当然，这时候车辆的车牌位置很可能会跑出识别区，识别区设置时候必须考虑到能覆盖车辆停下后补抓拍的车牌位置。

动态库另外会一直保留一帧最近的实时图像，这个图像的帧率大约是每秒5~10帧（依照jpeg压缩质量设置还有网络速度会有增减）。应用程序可以收到消息编号5后去获取以显示识别机实时图像。如果不需要实时视频，最好禁止实时视频以节省识别机工作量和网络的通信量。应用程序可以在收到消息编号1时就禁止实时视频（调用 [*LPNR_EnableLiveFrame*](#)）。

动态库所有接口都是 **STDCALL**，回调函数则是 C call。一般动态库的使用程序应该是使用消息机制获取消息通知，不建议使用回调函数。

2 动态库接口

下表列出所有动态库的函数以及功能说明。详细调用参数和返回结果见各个函数的说明。

以下函数宣告中，int 都是表示 32 位（win32 版本）或是 64 位（x64 版本）的整数型。

表格 1：函数列表

函数名	功能	备注
<i>LPNR_Init</i>	启动工作线程，连接设备。	必须第一个调用
<i>LPNR_SetWndMsg</i>	设置接收消息的窗口句柄以及使用的消息编号。	在 LPNR_Init 后立刻调用，否则识别机连接上后的第一个 ONLINE 的消息可能会漏失掉。
<i>LPNR_SetCallBack</i>	设置回调函数	在 LPNR_Init 后立刻调用。避免漏失消息。另外，使用回调函数就不应该在使用消息通知机制。
<i>LPNR_Terminate</i>	析构函数	
<i>LPNR_GetPlateNumber</i>	获取车牌识别结果字符串	
<i>LPNR_GetPlateColorImage</i>	获取车牌彩色图	
<i>LPNR_GetPlateColorImageSize</i>	获取车牌彩色图字节数	
<i>LPNR_GetPlateBinaryImage</i>	获取车牌二值化图	
<i>LPNR_GetPlateBinaryImageSize</i>	获取车牌二值化图字节数	
<i>LPNR_GetCapturedImage</i>	获取抓拍图片	
<i>LPNR_GetCapturedImageSize</i>	获取抓拍图片字节数	
<i>LPNR_GetLiveFrame</i>	获取最近的实时图像帧	
<i>LPNR_GetLiveFrameSize</i>	获取最近的实时图像帧字节数	
<i>LPNR_GetQuadImage</i>	获取 1/4 解像度图	
<i>LPNR_GetQuadImageSize</i>	获取 1/4 解像度图字节数	
<i>LPNR_GetHeadImage</i>	获取车头图图	
<i>LPNR_GetHeadImageSize</i>	获取车头图图字节数	
<i>LPNR_IsOnline</i>	询问识别机当前连线状态	
<i>LPNR_ReleaseData</i>	释放当前保存的识别结	

函数名	功能	备注
	果。	
<i>LPNR_SoftTrigger</i>	进行一次软触发，识别处理机制完全和线圈触发一样。	
<i>LPNR_IsIdle</i>	获知识别机现在是否闲置。返回 TRUE 表示闲置状态，没有进行识别。返回 FALSE 表示正在进行识别处理。	
<i>LPNR_GetTiming</i>	获取识别处理时间（过程时间和总处理时间）	
<i>LPNR_EnableLiveFrame</i>	使能或是禁能实时图像的传送。当网络比较拥挤的时候，禁能实时图像的传送可以减少大量数据。	
<i>LPNR_TakeSnapFrame</i>	触发识别机一次实时图像的获取发送到客户端。可以选择是否要打补光灯（当然前提是识别机的设置里需要使能补光功能）。	这个功能只获取图片，识别机不会进行车牌分析。这是和 LPNR_SoftTrigger 的区别。
<i>LPNR_Lock</i>	锁住实时图像帧和抓拍图像分析结果等数据。锁住后，动态库工作线程即无法更新，直到解锁为止。	获取实时图像帧数据时，最好先锁定，获取完后再解锁。避免图像数据刚好正被更新。至于其他分析结果数据，可以不用加锁，毕竟下次触发分析完才会有新的数据。此时上位机早就获取完前一组数据。
<i>LPNR_Unlock</i>	解锁数据。	
<i>LPNR_GetPlateAttribute</i>	获取车牌/车身额外信息	
<i>LPNR_GetMachineIP</i>	获取识别机 IP	这是设计来给用户的回调函数或是消息处理函数获取当前识别机 IP 用的。这样用户程序不需要刻意维护一个句柄和 IP 的关系表。
<i>LPNR_GetExtDI</i>	获取识别机 DI 点状态。	对于有额外 DI 点的识别机才有效。可以用来获取前后线圈状态，栏杆机反馈等讯号。
<i>LPNR_GetCameraLabel</i>	获取识别机名称	识别机可以用工具程序设置一个名称。这个名称可以叠加在抓拍图片上。此函数可以让上位机获取识别机的名称。

函数名	功能	备注
<i>LPNR_SyncTime</i>	同步时间	
<i>LPNR_ResetHeartBeat</i>	重置心跳帧时间	在连线状态下，当上位机系统时间被重新设置后，需要调用这个接口，重新计算心跳帧超时时间。
<i>LPNR_GetMachineIP</i>	获取识别机 IP	
<i>LPNR_GetExtDI</i>	获取识别机扩展 DI 的值。	只有某些机型有扩展 DI
<i>LPNR_GetCameraLabel</i>	获取识别机标签字符串（使用工具软件设置的一个字符串,可以理解为给识别机取的名字)	
<i>LPNR_GetEventText</i>	将动态库发送的事件编号以可读文字表述。	
<i>LPNR_GetModelAndSensor</i>	获取识别机的型号和感光器件代号	
<i>LPNR_GetVersion</i>	获取识别机的软件版本号	
<i>LPNR_GetCapability</i>	获取识别机的功能描述字	
<i>LPNR_EnableLog</i>	动态库日志控制-使能或是禁能	
<i>LPNR_SetLogPath</i>	动态库日志保存路径设置	

3 函数接口说明

3.1 LPNR_Init

HANDLE _stdcall LPNR_Init(const char *strIP)

功能： 连接设备

参数：

◆ strIP: 设备的 IP 字符串。

返回： 识别机的句柄，NULL 表示初始化失败。

说明： 本函数除非是 IP 地址错误，或是 Windows 系统出错无法创建工作线程，否则一定会返回成功。但是返回成功不表示已经连接到设备，正常情况下，如果网络是通的而且识别机有上电，工作线程会立刻建立连接并发送消息给应用程序。

3.2 LPNR_SetWndMsg

BOOL _stdcall LPNR_SetWinMsg(HANDLE h, HWND hwnd, int msgno)

功能： 使能事件消息回送使用者的程序。

参数：

h: LPNR_Init 返回的句柄（以下函数都相同）。

hwnd: 车道程序要接收消息的窗口指针。如果给 NULL 表示停止接收消息。

msgno: 发送的消息编号，也就是动态库会调用
PostMessage (hwnd, msgno, h, event_no)
发送消息。事件编号见第一节的说明。

返回： TRUE 成功，FALSE 失败。

3.3 LPNR_SetCallback

BOOL _stdcall LPNR_SetCallback(HANDLE h, LPNR_callback mycbfx)

功能： 设置用户的回调函数。

返回： TRUE 成功，FALSE 失败

说明： 失败的唯一可能是句柄错误。回调函数是使用 C 语音的 Calling Sequence，传入两个参数，第一个是识别机句柄，第 2 个事件编号。函数没有返回值。以 C 语音写的话，格式是：

void my_callback(HANDLE h, int nEventId);

特别注意，回调函数是使用动态库的工作线程调用的，因此不能占用太多处理时间，应该尽快返回。需要长时间处理的工作，应该另外启动线程处理。建议使用 Windows 消息通知机制，不要使用回调函数。

3.4 LPNR_Terminate

BOOL _stdcall LPNR_Terminate(HANDLE h)

功能： 结束工作线程，关闭连接，析构识别机对象。

返回： TRUE 成功，FALSE 失败

说明： 失败的唯一可能是句柄错误。

3.5 LPNR_GetPlateNumber

BOOL _stdcall LPNR_GetPlateNumber(HANDLE h, char *buf)

功能： 获取车牌颜色和号码字符串。

参数：

h: 识别机句柄。

buf: 接收车牌颜色和车牌号的字符串地址。车牌颜色占最前面 2 个字节，接着是车牌汉字及后面 6 位英文数字。例如：蓝苏 E12345。如果无识别出车牌，字符串内容为" 整牌拒识"。最前面 2 个字节是空白。字符最后会加一个结尾的 0。所以 buf 长度最少要有 11 个字节。

返回: TRUE 成功，FALSE 失败。

3.6 LPNR_GetPlateColorImage

*int _stdcall LPNR_GetPlateColorImage(HANDLE h, char *buf)*

功能: 获取车牌彩色图

参数:

h: 识别机句柄。

buf: 接收图片的内存块地址。

返回: 图片长度（字节数）。

注意: 应用程序需要保证接收的内存区大小足够，否则会造成程序退出。.

3.7 LPNR_GetPlateColorImageSize

int _stdcall LPNR_GetPlateColorImageSize(HANDLE h)

功能: 获取车牌彩色图字节数

参数:

h: 识别机句柄

返回: 图片长度（字节数）。

3.8 LPNR_GetPlateBinaryImage

*int _stdcall LPNR_GetPlateBinaryImage(HANDLE h, char *buf)*

功能： 获取车牌二值图

参数：

h: 识别机句柄。

buf: 接收图片的内存块地址。

返回： 图片长度（字节数）。

注意： 应用程序需要保证接收的内存区大小足够，否则会造成程序退出。.

3.9 LPNR_GetPlateBinaryImageSize

int _stdcall LPNR_GetPlateBinaryImageSize(HANDLE h)

功能： 获取车牌二值图字节数

参数：

h: 识别机句柄

返回： 图片长度（字节数）。

3.10 LPNR_GetCapturedImage

*int _stdcall LPNR_GetCapturedImage(HANDLE h, char *buf)*

功能： 获取抓拍图片

参数：

h: 识别机句柄。

buf: 接收图片的内存块地址。

返回： 图片长度（字节数）。

注意： 应用程序需要保证接收的内存区大小足够，否则会造成程序退出。.

3.11 LPNR_GetCapturedImageSize

int _stdcall LPNR_GetCapturedImageSize(HANDLE h)

功能： 获取抓拍图片字节数

参数：

h: 识别机句柄

返回： 图片长度（字节数）。

3.12 LPNR_GetLiveFrame

*int _stdcall LPNR_GetLiveFrame(HANDLE h, char *buf)*

功能： 获取实时图片

参数：

h: 识别机句柄。

buf: 接收图片的内存块地址。

返回： 图片长度（字节数）。

注意： 应用程序需要保证接收的内存区大小足够，否则会造成程序退出。最好准备一个固定的大 buffer。每次动态配置可能会出现一个问题，当获取当前实时图像帧长度后再动态分配好去接收，此时有可能动态库又接收了下一帧(如果没有加锁的话)。如果新的帧比较大，就会出问题。安全而有效率的做法是根据 LPNRTool 观察的实时图像帧显示的帧大小(会一直跳变)，准备一个足够大的 buffer。300KB 是一个安全的大小。另外，获取实时帧时，应该加锁，避免工作线程开始接收下一帧时，破坏当前帧内容。当然，如果用户是在回调函数内做这个动作，那不需要加锁。

3.13 LPNR_GetLiveFrameSize

int _stdcall LPNR_GetLiveFrameSize(HANDLE h)

功能： 获取实时图像帧字节数

参数：

h: 识别机句柄

返回： 图片长度（字节数）。

3.14 LPNR_GetQuadImage

*int _stdcall LPNR_GetQuadImage(HANDLE h, char *buf)*

功能： 获取 1/4 解像度的抓拍图片

参数：

h: 识别机句柄。

buf: 接收图片的内存块地址。

返回： 图片长度（字节数）。

3.15 LPNR_GetQuadImageSize

int _stdcall LPNR_GetQuadImageSize(HANDLE h)

功能： 获取 1/4 解像度抓拍图片字节数

参数：

h: 识别机句柄

返回值： 图片字节数（0 表示没有这个图片）或是句柄错误。

3.16 LPNR_GetHeadImage

*int _stdcall LPNR_GetHeadImage(HANDLE h, char *buf)*

功能： 获取抓拍图片内车头部分的截取图（720x576）

参数：

h: 识别机句柄。

buf: 接收图片的内存块地址。

返回： 图片长度（字节数）。

3.17 LPNR_GetHeadImageSize

int _stdcall LPNR_GetHeadImageSize(HANDLE h)

功能：获取车头图图片字节数

参数：

h：识别机句柄

返回值：图片字节数（0 表示没有这个图片）或是句柄错误。

3.18 LPNR_IsOnline

BOOL _stdcall LPNR_IsOnline(HANDLE h)

功能： 询问抓拍机是否连线中

参数：

h: 识别机句柄

返回： TRUE 为连线，FALSE 为离线（句柄非法也是返回 FALSE）。

3.19 LPNR_ReleaseData

BOOL _stdcall LPNR_ReleaseData(HANDLE h)

功能： 释放识别结果的储存数据，回收内存。

参数:

h: 识别机句柄

返回: TRUE 为成功, FALSE 为失败 (只有 h 非法才可能返回失败)。

3.20 LPNR_SoftTrigger

BOOL _stdcall LPNR_SoftTrigger(HANDLE h)

功能: 触发一次抓拍识别。

参数:

h: 识别机句柄

返回: TRUE 为成功, FALSE 为失败 (只有可能是 h 非法或是离线)。

3.21 LPNR_IsIdle

BOOL _stdcall LPNR_IsIdle(HANDLE h)

功能: 获知识别机当前是否闲置(没有进行识别处理中)。

参数:

h: 识别机句柄

返回: TRUE 为闲置, FALSE 为正在识别 (如果离线或是句柄非法也会返回 FALSE)

3.22 LPNR_GetTiming

BOOL _stdcall LPNR_GetTiming(HANDLE h, int *elapsed, int *processed)

功能: 获取当前识别结果的总过程时间和总处理时间。

参数:

h: 识别机句柄

elapsed: 32bit 整数型变量的指针，接收总过程时间。

processed: 32bit 整数型变量的指针，接收总处理时间。

返回: TRUE 为成功, FALSE 为失败 (离线或是句柄非法)

3.23 LPNR_EnableLiveFrame

BOOL_stdcall LPNR_EnableLiveFrame(HANDLE h, BOOL bEnable)

功能: 使能或是禁能实时图像帧的传送。

参数:

h: 识别机句柄

bEnable: TRUE 使能, FALSE 禁能。

返回值: TRUE 为成功, FALSE 为失败 (离线或是句柄非法)

注: 识别机在客户端连接后, 默认是发送实时视频的。因此, 如果应用程序不需要实时视频的话, 可以在消息编号 1 的消息处理函数里面调用本函数禁止实时视频。

3.24 LPNR_TakeSnapFrame

BOOL_stdcall LPNR_TakeSnapFrame(HANDLE h, BOOL bFlashLight)

功能: 命令识别机获取一帧图像发送到客户端。

参数:

h: 识别机句柄

bFlashLight: 抓拍时是否补光 (如果 TRUE, 必须是识别机的参数设置里有配置补光功能, 否则无效)。

返回值: TRUE 为成功, FALSE 为失败 (离线、句柄非法、超时没有收到抓拍帧)

注：函数返回成功时候，动态库已经收到刚抓拍的帧。程序可以调用 LPNR_GetCapturedImageSize 获知帧大小，调用 LPNR_GetCapturedImage 获取帧图像（JPEG）。

3.25 LPNR_Lock

BOOL _stdcall LPNR_Lock(HANDLE h)

功能：对数据加锁

参数：

h：识别机句柄

返回值：TRUE：成功，FALSE：失败

说明：为了避免 racing condition 出现，访问数据前先加锁，访问完解锁是比较好的习惯。尤其是实时视频，因为它在不停的在更新。以下的例子有可能会造成灾难的结果：

```
int size = LPNR_GetLiveFrameSize(h);  
void *frame = malloc(size); // 动态分配一个当前实时帧大小的内存  
int realsize = LPNR_GetLiveFrame(h, frame);  
...  
free(frame); // 释放内存
```

如果在 LPNR_GetLiveFrameSize 和 LPNR_GetLiveFrame 之间，动态库收到一帧新的实时帧，且大于前一帧 (realsize > size)，那结果就是内存越位，程序退出。因此需要在 LPNR_GetLiveFrameSize 之前先做加锁，在 LPNR_GetLiveFrame 之后立刻解锁。动态库内部在更新数据时也会加锁，更新完解锁。这样可以彼此安全的访问数据，相安无事。当然，更好的做法是分配一个足够大的 buffer 一直重复使用。即使如此也需要加锁，否则有可能获取的图像内

容刚好被动态库线程更新中。

3.26 LPNR_Unlock

BOOL _stdcall LPNR_Unlock(HANDLE h)

功能： 对数据解锁

参数：

h: 识别机句柄

返回值： TRUE: 成功, FALSE: 失败

3.27 LPNR_GetPlateAttribute

BOOL cdecl LPNR_GetPlateAttribute(HANDLE h, BYTE *attr)

功能： 获取当前车牌和车辆的附属信息。

参数：

h: 识别机句柄

attr: 获取信息的 BYTE 数组指针。attr 长度为 8 个字节, 每个字节的内容为:

0: 可信度 (0~100)

1: 车头或是车尾的车牌 (1 为车头, 0xff 为车尾, 0 表示未知)

2: 车身颜色码

3: 车速 (如果是超速触发抓拍的话才有这个值, 单位是 km/hour)

4: 触发源 (0 为图片返送分析、1 为线圈触发、2 为视频触发、3 为定时触发、4 为虚拟线圈触发、5 为超速触发)

5~7: 保留。

返回值: RUE: 成功, FALSE: 失败

3.28 LPNR_GetMachineIP

BOOL _stdcall LPNR_GetMachineIP(HANDLE h, char *chIP)

功能: 获取识别机的 IP 字符串。

参数:

h: 识别机句柄

chIP: 用户用来接收 IP 字符串的 buffer 指针。

返回值: TRUE 成功, FALSE 失败。

3.29 LPNR_GetExtDI

BOOL _stdcall LPNR_GetExtDI(HANDLE h, int *diLastnThis)

功能: 获取识别机当前的 DI 变化读值。

参数:

h: 识别机句柄

diLastnThis: 用户用来保存 DI 变化的整数型变量指针。

返回值: TRUE 成功, FALSE 失败。

说明: 用户收到消息 (或是事件) 编号 8 的消息 (事件) 时, 调用此函数获取 DI 读值。返回成功后, diLastnThis 所指整数变量的内容为:

bit0~bit15: 当前 DI 点读值, 一个 bit 代表一个 DI 点 (识别机只有最多 4 个 DI 点), bit16~bit31 为本次变化前的 DI 读值, 一个 bit 代表一个 DI 点。 例如: diLastnThis 的内容为 0x00030001

表示识别机的第 2 个 DI 点由 1 变为 0。

3.30 LPNR_GetCameraLabel

LPCTSTR _stdcall LPNR_GetCameraLabel(HANDLE h)

功能： 获取识别机的名称,此名称由配置工具程序设置,最长 20 字节。

参数：

h: 识别机句柄

返回值： 识别机名称的字符串。

3.31 LPNR_SyncTime

BOOL _stdcall LPNR_SyncTime(HANDLE h)

功能： 设置识别机的系统时间(根据当前计算机系统时间)。

参数：

h: 识别机句柄

返回值： TRUE: 成功, FALSE: 失败

注： 动态库每次连接识别机后就自动发送对时帧,因此这个对时帧可以在连线后设置计时器,每隔一段时间(例如一小时)调用一次即可。

3.32 LPNR_ResetHeartBeat

BOOL _stdcall LPNR_ResetHeartBeat(HANDLE h)

功能： 重置动态库计算心跳帧超时的时间。

参数：

h: 识别机句柄

返回值： TRUE: 成功, FALSE: 失败

3.33 LPNR_GetEventText

const char * STDCALL LPNR_GetEventText(int event)

功能： 将识别机的事件编号使用以可读字符串表述返回

参数：

event: 识别机发送的事件编号

返回值： 事件编号的解释字符串

3.34 LPNR_GetModelAndSensor

BOOL STDCALL LPNR_GetModelAndSensor(HANDLE h, int *modelId, int *sensorId)

功能： 获取识别机的型号和感光器件型号 ID

参数：

h: 识别机句柄

modelId: 保存识别机型号的变量地址

sensorId: 保存感光器型号的变量地址

返回值： TRUE 成功，FALSE 失败

3.35 LPNR_GetVersion

BOOL STDCALL LPNR_GetVersion(HANDLE h, DWORD *ver, int *algver)

功能： 获取识别机的软件版本号和识别算法版本

参数：

h: 识别机句柄

ver: 保存识别机版本号的地址。此版本号是以 BCD 编码，
例如 ver 内容为 0x01020304 代表版本号为 1.2.3.4

algver: 保存算法库版本号变量地址，算法库版本为 binary

保存，高 16 位为主板本号，低 16 位为次版本号。

返回值： TRUE 成功，FALSE 失败

3.36 LPNR_GetCapability

BOOL STDCALL LPNR_GetCappability(HANDLE h, DWORD *cap)

功能： 获取识别机的功能字(capability word)

参数：

h: 识别机句柄

cap: 保存识别机功能字的变量地址

返回值： TRUE 成功，FALSE 失败

注释： 功能字是用来标示识别机一些系统相关以及软件条件编译使能禁能的一些讯息。例如是否支持 IPv6, 是否有插 SD 卡。这是我司内部使用的 API, 客户如有需要详细说明可以另行提供。

3.37 LPNR_SetLogPath

BOOL STDCALL LPNR_SetLogPath(HANDLE h, const char *path)

功能： 设置动态库日志保存的路径

参数：

h 设备句柄。

Path 日志保存路径

返回值： TRUE 成功，FALSE 失败。

说明： 设置设备日志的保存路径。h 是要设置的日志路径的设备句柄。如果 h 是 NULL, 这个路径对所有后续创建的设备都生效，对已经创建并且使用中的设备日志文件无效。必须等该设备当前日志文件大小达到上限, 要开启下一个日志时才生效。

如果h不是NULL,指定设备的当前日志会关闭,当有新的日志行要写入时,重新在指定的位置创建新的日志文件。

Path 参数是要保存日志的文件夹路径。路径最后一个分支节点可以不存在,动态库会创建出来。但是分支所在的父节点必须存在。例如 Path 指定为 : "T:\Runwell\ALBDLL\20170410", 则 "T:\Runwell\ALBDLL\"这个父节点必须是存在的,20170410 这个分支节点可以不存在,由动态库创建出来。

3.38 LPNR_EnableLog

BOOL STDCALL LPNR_EnableLog(HANDLE h, BOOL bEnable)

功能: 使能/禁能日志功能

参数:

h	设备句柄。
bEnable	使能(TRUE)或是禁能(FALSE)

返回值: TRUE 成功, FALSE 失败

说明: 使能/禁能针对该设备的日志,参数h为设备句柄,必须是LPNR_Init返回的句柄。bEnable参数控制使能(TRUE)或是禁能(FALSE)。

日志在设备创建后,默认是使能日志功能的。在调用LPNR_Init时,如果日志功能是使能的状态,函数就会开启日志文件。日志文件在LPNR_Terminate函数中关闭。应用程序可以先调用LPNR_EnableLog(NULL,FALSE)关闭日志,这样在LPNR_Init函数中就不会开启日志。

应用程序也可以在设备创建后随时改变日志的使能状态,禁能时,日志文件被关闭。使能后,下

次打印日志时再度开启文件。

日志文件的保存路径默认是应用程序盘符下的 \rwlog。可以使用 LPNR_SetLogPath 函数改变这个保存路径。日志文件名为 LPNRDLL-<IP>.log，它的最大上限为 4MB。当超过上限后，会滚动备份到带数字后缀的文件，例如 LPNRDLL-<IP>-01.log。后缀数字越大的文件越老，一共保存 30 个滚动备份。