ООО «Мегапиксел»

Библиотека для создания приложений по определению автомобильных номеров, определению номеров железнодорожных вагонов, систем видеонаблюдения и систем охраны с использованием детекции движения MegaLib V1.2

Руководство программиста

Москва 2013 год

Содержание

1.Назначение

Библиотека MegaLib V1 является развитием ранее выпущенных библиотек ООО «Мегапиксел» CARFLOW5, TRNFLOW5, MOTION5 и объединяет их в единую систему с более широкими возможностями. Единый интерфейс захвата изображений и возможность комбинирования различных функций обработки изображений в одной библиотеки позволяют разработчикам прикладных программных средств создавать разнообразные системы безопасности, которые сочетали бы в себе следующие основные компоненты:

- многоканальное видеонаблюдение;
- многоканальный контроль аудио данных;
- охранные системы на базе детекторов движения и детекторов оставленных и унесенных предметов;
- системы распознавания государственных номерных знаков автомобилей;
- системы распознавания номеров железнодорожных вагонов.

По сравнению с ранее разработанными библиотеками MegaLib обладает следующими отличительными свойствами:

- возможность работы с разнородными устройствами ввода одновременно;
- работа с устройствами с прогрессивной разверткой;
- работа с USB и IP устройствами захвата изображений;
- произвольные размеры обрабатываемых изображений;
- многозонная система выделения номеров, как с широкими, так и с квадратными пикселями изображения;
- одновременная работа, как функций распознавания, так и детекторов движения;
- работа со стандартными и специализированными устройствами захвата аудио данных;
- возможность построения многозадачных приложений с параллельной обработкой данных.

Библиотека работает под управлением операционных систем WINDOWS2000, WINDOWSXP, WINDOWS7, WINDOWS8 и обеспечивает возможность построения многоканальных приложений с использование как устройств захвата изображений производимых ООО «Мегапиксел» (МЕГАФРЕЙМ-4, МЕГАФРЕЙМ-16, МЕГАФРЕЙМ-4x4, МЕГАФРЕЙМ-X), так и устройств, которые используют DirectShow драйверы (TV тюнеры, USB камеры, IP устройства ввода). Система позволяет одновременно активировать различные типы вводных устройств одновременно, что позволяет оптимально компоновать конечную систему с точки зрения количества используемых каналов. Библиотека может комплектоваться ключом защиты GUARDANT и SenseLockSL, которые дают возможность использовать функции библиотеки с изображениями, полученными с источников пользователя.

Для более эффективного использования в библиотеке применены команды SSE2. Поддержка таких команд определяется автоматически и значительно

ускоряет процесс обработки. Поэтому желательно использовать компьютеры на базе Pentium IV, Pentium Core, Pentium I7 или XEON.

В состав библиотеки входят следующие основные функции:

- функции управления различными устройствами ввода изображений;
- функции управления различными устройствами ввода аудио данных;
- функции управления дополнительными внешними устройствами;
- функции определения автомобильных номеров;
- функции детекции движения и определения оставленных и унесенных предметов;
- функции определения номеров железнодорожных вагонов.

Активация различных функций библиотеки осуществляется автоматически в зависимости от установленных устройств захвата изображений и модулей, осуществляющих аппаратную защиту системы от несанкционированного использования.

Библиотека построена с использованием компилятора BORLAND C++.

2.Список файлов библиотеки

Рабочие файлы библиотеки MegaLib1

MEGALIB1.DLL - основной файл процедур;
MegaLibV12.PDF - этот файл;
MEGALIB1.LIB - экспортная библиотека для компиляции программ;
MEGALIB1.H - файл описания для языка C++;
MEGALIB1.INI - файл инициализации внутренних переменных;
MEGALIB1.DEF - список вызова внешних функций.
MPCARREC.MPL - файлы необходимые для работы основных
MPCARREC.TBL процедур;
MPCNUM.TBL
MPFNDCAR.TBL
MPSYMBOLSC.TBL
MPZONE.TBL
MPTRNREC.MPL
NVV.DLL
MPTRAINMNG.EXE
MPTASKMNG.EXE
MPCHANNELMNG.EXE
AXISCONFIG.EXE
COUNTRY.CFG - конфигурационный файл типов номеров различных стран.

Библиотеки управления устройствами ввода/вывода

DEVICE\MegaFrameX_RT.DLL
DEVICE\MegaFrameX_2SW.DLL
DEVICE\MegaFrameX_4SW.DLL
DEVICE\MegaFrame4x4.DLL
DEVICE\MegaFrame4.DLL
DEVICE\MegaFrame4_RT.DLL
DEVICE\Guardant.DLL
DEVICE\SenseLockEL.DLL
DEVICE\WDM_DeviceN.DLL
DEVICE\FileEmulatorN.DLL
DEVICE\PointGrayCameras.DLL
\PGRFlyCapture.DLL
DEVICE\MegaFrameE.DLL
\DVEX7115.DLL
DEVICE\MegaFrameE_RT.DLL
\DVEX7115.DLL
DEVICE\AUDIO_DeviceN.DLL
DEVICE\DLPInOut.DLL
\DLPInOutSW.EXE
\ftd2xx.DLL
DEVICE\SoftInOut.DLL
\SoftInOutSW.EXE
DEVICE\Conexant_CX258xx_0.DLL

DEVICE\Conexant_CX258xx_1.DLL
DEVICE\USB3104.DLL
DEVICE\MFSaa713x.DLL
DEVICE\MFSaa713x_RT.DLL
DEVICE\MPTW6816.DLL
DEVICE\MPTW6816 _RT.DLL
DEVICE\MPTW6869.DLL
\TW68XX.DLL
DEVICE\AXISWebCameraN.DLL
\AXISInterface.EXE
\PICN1120.DLL
\PICN20.DLL

Подробное описание управляющих библиотек представлено в п.9 описания.

3. Функции инициализации библиотеки

- **Инициализация библиотеки MegaLib**

UINT MPCreate(VOID);

Первая функция использования библиотеки. Инициализирует ресурсы системы, необходимые для работы, и активирует требуемые функции. Процедура всегда возвращает значение 0.

- **Инициализация части ресурсов библиотеки MegaLib**

UINT MPCreateEx(UINT In, UINT Md, UINT Cf, UINT Tf);

Функция выполняет действия предыдущей функции, но инициализирует только часть ресурсов системы, необходимых для работы. Она активирует требуемое количество каналов в пределах разрешенного количества. Функция предназначена для построения многозадачных приложений в многопроцессорных (многоядерных) системах, в которых каналы обрабатываются параллельно.

Параметры:

In
Количество требуемых каналов ввода изображений.

Md
Количество требуемых каналов детекторов движения.

Cf
Количество требуемых каналов определения автомобильных номеров.

Tf
Количество требуемых каналов определения железнодорожных номеров.

Процедура всегда возвращает значение 0.

- **Отключение библиотеки MegaLib**

VOID MPRelease(VOID);

Последняя функция использования библиотеки. Освобождает ранее распределенные ресурсы системы, и выгружает функции библиотеки.

- **Проверка ресурсов библиотеки MegaLib**

UINT MPResource(UINT mode);

Функция возвращает значение количества доступных каналов после выполнения функций MPCreate и MPCreateEx. Для определения максимального количества каналов доступных для пользователя данная функция может выполняться до вызова функции MPCreate.

Параметры:

mode
Тип запрашиваемого ресурса.
MSENABLED 0x0 - количество каналов ввода.
CFENABLED 0x1 - количество каналов определения автомобильных номеров.
MDENABLED 0x2 - количество каналов детекторов движения.
TFENABLED 0x4 - количество каналов определения железнодорожных номеров.

4. Функции обслуживания виртуальных устройств захвата изображений

Библиотека обеспечивает захват изображений и аудио данных от различных источников, которые могут быть следующих типов: устройства ввода поставляемые ООО «Мегапиксел», универсальные устройства ввода, поставляемые с DirectShow драйверами (TV тюнеры, USB устройства, IP камеры) и файлы, содержащие последовательность ранее записанных кадров. Обслуживание конкретных устройств обеспечивают управляющие библиотеки, расположенные в каталоге **<INSTALLDIR>\\DEVICE**. Система позволяет работать одновременно с несколькими разнотипными устройствами захвата изображения, которые последовательно добавляются в список устройств, увеличивая соответственно число каналов системы. Каждое устройство содержит элементы защиты или имеет соответствующее ему защитное устройство. Каждый элемент защиты имеет информацию о возможности использования той или иной части библиотеки.

Существуют два уровня обслуживания устройств захвата: обслуживание с использованием идентификатора устройства (в данном случае работа осуществляется непосредственно с каждым отдельно взятым устройством) и на уровне отдельных каналов (система сама определяет требуемое устройство, которое соответствует данному каналу).

4.1. Функции обслуживания виртуальных устройств захвата изображений с использованием идентификатора устройства

При вызове данной группы функций, в качестве параметров определяющих канал обработки используется идентификатор устройства и номер канала в пределах данного устройства.

- **Прочитать версию библиотеки**

UINT IDGetVersion(VOID);

Возвращает номер текущей версии функций устройств ввода/вывода.
0x100 – Версия библиотеки MegaLib V 1.0
0x101 – Версия библиотеки MegaLib V 1.1
0x102 – Версия библиотеки MegaLib V 1.2

- **Инициализация устройства ввода**

HDEVICE IDAdd(char *DeviceName,UINT Mode);

Функция добавляет в систему одну из управляющих библиотек захвата изображений, создает под это устройство каналы обработки и возвращает идентификатор устройства. Если в системе установлены несколько разнотипных устройств захвата, то можно произвести инициализацию всех устройств и создать под них соответствующие каналы обработки.

Параметры:

DeviceName
Полный путь к одной из управляющих библиотек.
Например: C:\\Program File\\MegaLib\\DEVICE\MegaFrame4.DLL
Если данный параметр NULL, инициализируются все каналы детекторов движения и определения номеров, доступных в системе.

Mode
0 – режим ввода PAL;
1 – режим ввода NTSC.

Результат:
При успешной инициализации функция возвращает идентификатор устройства. Он должен быть больше или равен 16-и. Если результат меньше 16-и, то он имеет следующие значения ошибки:

Определение	Величина	Значение
ERRORDEVICESLIMIT	0	Превышение количества возможных устройств.
ERRORDEVICESECURITY	1	Проверка защиты показала наличие

		использования несанкционированного устройства.
ERRORDEVICELOAD	2	Ошибка загрузки DLL или одной из функций устройства.
ERRORDEVICEOPEN	3	Ошибка открытия управляющей библиотеки.
ERRORDEVICEUSE	4	Устройство не предназначено для работы с данной библиотекой.
ERRORDEVICECODE	5	Библиотека создана сторонним лицом или была попытка взлома.

- **Окончание работы с виртуальным устройством**

VOID IDClose (HDEVICE Hdv);

Функция завершает работу с виртуальным устройством, освобождает ранее распределенные ресурсы и отключает соответствующие библиотеки. Одновременно происходит уменьшение количества доступных каналов на величину каналов разрешенных для данного устройства. После осуществления отключения устройства следует произвести повторную инициализацию всех оставшихся каналов.

Параметры:

Hdv
Идентификатор устройства, возвращенный функцией **IDAdd.**

```
HDEVICE Hdv;

MPCreate();
Hdv = IDAdd("C:\\Program File\\MegaLib\\DEVICE\\MegaFrame4.DLL",0);
if(Hdv>=16) {

      Использование устройства

      IDClose(Hdv);
      }
MPRelease();
```

- **Окончание работы со всеми виртуальными устройствами**

VOID IDCloseAll (VOID);

Функция завершает работу со всеми виртуальными устройствами, освобождает ранее распределенные ресурсы и отключает все управляющие библиотеки.

```
HDEVICE Hdv[2];

MPCreate();
Hdv[0] = IDAdd("C:\\Program File\\MegaLib\\DEVICE\\MegaFrame4_RT.DLL",0);
Hdv[1] = IDAdd("C:\\Program File\\MegaLib\\DEVICE\\MegaFrameX_RT.DLL",0);

//Использование устройств

IDCloseAll();
MPRelease();
```

- **Проверка количества активных устройств захвата изображений**

UINT IDGetNumberDevices(void);

Результат:

Функция возвращает количество устройств активизированных в системе функцией **IDAdd**.

```
HDEVICE Hdv[2];
UINT NumberDevices;

MPCreate();
Hdv[0] = IDAdd("C:\\Program File\\MegaLib\\DEVICE\\MegaFrame4.DLL",0);
Hdv[1] = IDAdd("C:\\Program File\\MegaLib\\DEVICE\\MegaFrameX.DLL",0);

NumberDevices  = IDGetNumberDevices(); // NumberDevices=2;

//Использование устройств

IDCloseAll();
MPRelease();
```

- **Проверка количества каналов доступных системе.**

UINT IDGetMaxChannels(void);

Результат:

Функция возвращает общее количество доступных каналов системы.

```
HDEVICE Hdv[2];
UINT MaxChannels;

MPCreate();
Hdv[0]=IDAdd("C:\\Program File\\MegaLib\\DEVICE\\MegaFrame4_RT.DLL",0); //4
Hdv[1]=IDAdd("C:\\Program File\\MegaLib\\DEVICE\\MegaFrameX_RT.DLL",0); //8
```

```
MaxChannels = IDGetMaxChannels(); // MaxChannels = 12

//Использование устройств

IDCloseAll();
MPRelease();
```

- **Получение свойств устройства захвата изображения**

VOID IDGetDeviceProperty(HDEVICE Hdv,TDEVICEPROPERTY *Prop);

После инициализации устройства данная функция возвращает свойства устройства ввода. Кроме того, при помощи данной функции можно определить возможность использования различных элементов библиотеки для данного устройства.

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Prop
Указатель на структуру, которая содержит свойства устройства ввода.

```
typedef struct TDeviceProperty {
        UINT NumberInLines;
        UINT NumberOutLines;
        UINT NumberChannels;
        UINT MaxHorFrameSize;
        UINT MaxVerFrameSize;
        UINT DeviceFlags;
        UINT NumberAudioChannels;
      } TDEVICEPROPERTY;
```

NumberInLines
Число доступных аппаратных линий ввода цифровых сигналов для подключения датчиков.

NumberOutLines
Число доступных аппаратных линий вывода цифровых сигналов для подключения внешних устройств.

NumberChannels
Число доступных каналов ввода изображений для данного устройства.

MaxHorFrameSize
Максимальный размер изображения по горизонтали для данного устройства.

MaxVerFrameSize
Максимальный размер изображения по вертикали для данного устройства.

DeviceFlags
Флаги, определяющие параметры вводимого кадра для данного устройства.

Определение	Величина	Значение
TVSTANDARD	0x00000001	Устройство вводит изображение в TV стандарте (два полукадра).
PROGRESIVE	0x00000002	Устройство вводит изображение в прогрессивной развертке (один кадр полного размера).
SQUAREPIXELS	0x00000004	Ввод осуществляется в режиме квадратных пикселей.
WIDEPIXELS	0x00000008	Ввод осуществляется в режиме широких пикселей (один полукадр в TV стандарте).
ADJUSTENABLED	0x00000010	Устройство позволяет изменять параметры ввода (яркость, контрастность, цветность).
PRIORITYENABLED	0x00000020	Устройство позволяет использовать приоритет ввода по отдельным каналам.
RGBPIXELS	0x00000040	Ввод осуществляется в формате RGB24.
UYVYPIXELS	0x00000080	Ввод осуществляется в формате UYVY.
GRAYPIXELS	0x00000100	Ввод осуществляется в формате Y (256 градаций яркости)
YUY2PIXELS	0x00000200	Ввод осуществляется в формате YUYV.
DEVCFENABLED	0x00010000	Разрешено использование функций определения автомобильных номеров.
DEVMDENABLED	0x00020000	Разрешено использование функций детекции движения.
DEVTFENABLED	0x00040000	Разрешено использование функций определения номеров железнодорожных вагонов.

NumberAudioChannels
Число доступных каналов ввода аудио данных для данного устройства.

- **Изменение свойств устройства ввода**

VOID IDSetDeviceProperty(HDEVICE Hdv,TDEVICEPROPERTY *Prop);

После инициализации устройства данная функция позволяет переустановить размеры вводимого изображения. Некоторые устройства не позволяют изменять размеры кадра и для проверки необходимо снова выполнить функцию IDGetDeviceProperty для данного устройства с целью проверки правильности выполнения функции.

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Prop
Указатель на структуру, которая содержит свойства устройства ввода.

```
typedef struct TDeviceProperty {
        UINT NumberInLines;
        UINT NumberOutLines;
        UINT NumberChannels;
        UINT MaxHorFrameSize;
        UINT MaxVerFrameSize;
        UINT DeviceFlags;
        UINT Reserved;
      }  TDEVICEPROPERTY;
```

NumberInLines
NumberOutLines
NumberChannels
DeviceFlags
Reserved
Данные значения не могут быть изменены.

MaxHorFrameSize
Устанавливаемый размер изображения по горизонтали для данного устройства.

MaxVerFrameSize
Устанавливаемый размер изображения по вертикали для данного устройства.

- **Получение свойств отдельного канала устройства ввода**

VOID IDGetChannelProperty(
HDEVICE Hdv,
UINT Channel,
TCHANNELPROPERTY *CHProp);

После инициализации устройства данная функция возвращает свойства отдельных каналов устройства ввода.

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Channel
Номер канала, для которого необходимо произвести чтение свойств. Номер канала определяется только для данного устройства ввода и не соответствует номеру общего числа каналов при использовании нескольких устройств. При использовании одного устройства номера каналов устройства совпадают с общим номером канала системы.

CHProp
Указатель на структуру, которая содержит свойства канала устройства ввода.

```
typedef struct TChannelProperty {
        UINT HorFrameSize;
        UINT VerFrameSize;
        UINT PixelMode;
        UINT ColorCode;
        UINT ChannelFlags;
        UINT Reserved;
      } TCHANNELPROPERTY;
```

HorFrameSize
Размер в пикселях вводимого кадра по горизонтали.

VerFrameSize
Размер в пикселях вводимого кадра по вертикали.

PixelMode
Свойства кадра.

Определение	Величина	Значение
WIDE_PIXELS	0x00000001	Каждый кадр (или полукадр при

		установленном флаге DOUBLEFIELD_PIXELS) содержит в ширину удвоенные размеры.
SQUARE_PIXELS	0x00000002	Каждый кадр (или полукадр при установленном флаге DOUBLEFIELD_PIXELS) содержит в ширину квадратные пиксели.
DOUBLEFIELD_PIXELS	0x00000004	Кадр содержит два полукадра, которые следуют один за другим.

ColorCode
Цветовая кодировка каждого пикселя. Может принимать одно из следующих значений:
MAKEFOURCC('G','R','A','Y'); - полутоновое изображение (1 байт на пиксель).
MAKEFOURCC('U','Y','V', 'Y'); - цветное изображение в формате UYVY (2 байта на пиксель).
MAKEFOURCC('Y','U','Y','2'); - цветное изображение в формате YUY2 (2 байта на пиксель).
MAKEFOURCC('R','G','B',' '); - цветное изображение в формате RGB24 (3 байта на пиксель).

Где: MAKEFOURCC(ch0, ch1, ch2, ch3) \
((DWORD)(BYTE)(ch0) | ((DWORD)(BYTE)(ch1) << 8) | \
((DWORD)(BYTE)(ch2) << 16) | ((DWORD)(BYTE)(ch3) << 24))

ChannelFlags
Возможность работы с функциями библиотеки.

Определение	Величина	Значение
CFENABLED	0x00000001	Разрешено работать с функциями определения автомобильных номеров.
MDENABLED	0x00000002	Разрешено работать с функциями детекции движения.
TFENABLED	0x00000004	Разрешено работать с функциями определения номеров железнодорожных вагонов.

Reserved
Зарезервировано для дальнейшего развития системы.

```
HDEVICE Hdv[2];
UINT NumberDevices;
UINT MaxChannels=0;
TCHANNELPROPERTY CProp;
TDEVICEPROPERTY DProp;
char *SaveImageT[MAXCHANNELS];
void *SaveImage[MAXCHANNELS];

MPCreate();
Hdv[0] = IDAdd("C:\\Program File\\MegaLib\\DEVICE\\MegaFrame4.DLL",0); //4
```

```
Hdv[1] = IDAdd("C:\\Program File\\MegaLib\\DEVICE\\MegaFrameX.DLL",0); //8

NumberDevices  = IDGetNumberDevices(); // NumberDevices=2;

for(int dv=0;dv<NumberDevices;dv++){
     IDGetDeviceProperty(Hdv[dv],&DProp);

     for(int i=0;i<DProp.NumberChannels;i++){
          IDGetChannelProperty(Hdv[dv],i,&CProp);
          UINT cl;
          if(CProp.ColorCode ==  MAKEFOURCC('G','R','A','Y')) cl=1;
          if(CProp.ColorCode ==  MAKEFOURCC('R','G','B',' ')) cl=3;
          if(CProp.ColorCode ==  MAKEFOURCC('U','Y','V','Y')) cl=2;
          if(CProp.ColorCode ==  MAKEFOURCC('Y','U','Y','2')) cl=2;
          SaveImageT[MaxChannels] =
               new char[CProp.HorFrameSize * CProp.VerFrameSize*cl+256];
          SaveImage[MaxChannels]=
               (PVOID)((UINT)(SaveImageT[MaxChannels]+0x100)&0xffffff00);
          MaxChannels++;
          }
     }

//Использование устройств

MaxChannels = IDGetMaxChannels(); // MaxChannels = 12
for(int i=0;i<MaxChannels;i++) delete SaveImageT[i];

IDCloseAll();
MPRelease();
```

- **Получить кадры изображения**

VOID IDGetPicture(HDEVICE Hdv, UINT *ImgTegg, VOID **Images);

Вызов данной функции должен производиться непрерывно и по возможности с максимальной интенсивностью. Рекомендуется использование потока с высоким приоритетом. Функция может не вызываться в случае отключения всех каналов.

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

ImgTegg
Параметр инициализируется пользователем и заполняется функцией. Указатель на массив **ImgTegg[NumberChannels]**, где **NumberChannels** количество каналов поддерживаемых данным устройством, который определен в структуре **TDEVICEPROPERTY**. Каждый элемент массива может принимать следующие значения:

Определение	Величина	Значение
NOPICTURE	0x00000000	Готовности кадра нет
PICTUREREADY	0x00000001	Готовность кадра есть

INPUTUNCONNECTED	0x00000002	Произошло отключение канала от видеосигнала

Images
Указатель на массив указателей изображений каналов **Images[NumberChannels]**, где **NumberChannels** количество каналов поддерживаемых данным устройством. Размер каждого буфера должен соответствовать запрошенному изображению, который определяется при чтении свойств каналов**.** Каждый буфер должен быть выровнен до 256 байт для правильной работы SSE2 инструкций**.**

```
void GetImagesFromDevice()
{
UINT MaxChannels;
UINT *ImgTegg;
UINT NumberDevices;

MaxChannels = IDGetMaxChannels();
ImgTegg = new UINT [MaxChannels];

NumberDevices  = IDGetNumberDevices();

for(int dv=0;dv<NumberDevices;dv++)
     IDGetPicture(IDGetDevices(dv),
                    &ImgTegg[IDGetFirstChannel(dv)],
                      &SaveImage[IDGetFirstChannel(dv)]);

for(int i=0;i<MaxChannels;i++)
     switch(ImgTegg[i]){
        case NOPICTURE : break;
        case PICTUREREADY : LibRun(&i);/*обработка буфера*/ break;
        case INPUTUNCONNECTED : /*обработка отключения канала*/ break;
     }

delete ImgTegg;
}
```

- **Установить параметры канала**

VOID IDSetParams (
HDEVICE Hdv,
UINT Channel,
TINPUTPARAMS *Params);

Данная функция производит установку параметров ввода: яркость, контрастность, цветность, коррекцию, режим цвета и пространственное разрешение ввода для каждого канала отдельного устройства. Функция применяется с ограниченным числом типов устройств и для более эффективного использования устройств захвата изображений следует использовать функции **IDSetCaptureProperty, IDGetCaptureProperty**.

Параметры

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Channel
Номер устанавливаемого канала. Номер канала определяется только для данного устройства ввода и не соответствует номеру общего числа каналов при использовании нескольких устройств. При использовании одного устройства номера каналов устройства совпадают с общим номером канала системы.

Params
Указатель на структуру параметров канала

typedef struct TInputParam {
UINT ColorMode; - 0 – черно/белый; 1 - цветной
UINT InputMode;
UINT Brightness; - яркость 0 - 255
UINT Contrast; - контрастность 0-255
UINT ChromaV; - яркость V 0-255
UINT ChromaU; - яркость U 0-255
UINT Hue; - баланс 0-255
} TINPUTPARAMS;

InputMode
Параметр определяет тип вводимого изображения:
0 – один полукадр низкого разрешения с учетом полей
1 – один полукадр высокого разрешения с учетом полей
2 – два полукадра низкого разрешения с учетом полей
3 – два полукадра высокого разрешения с учетом полей
4 – один полукадр низкого разрешения без учета полей
5 – один полукадр высокого разрешения без учета полей
6 – два полукадра низкого разрешения без учета полей
7 – два полукадра высокого разрешения без учета полей

Примечание:

Параметр **InputMode** при использовании библиотеки с обработкой двух полукадров следует установить в режим 7. При работе с одним полукадром данный параметр устанавливается в значение 1 для 25 кадров/с или 5 для 50 кадров/с.

Функция **IDSetParams** должна быть выполнена раньше функции **IDGetChannelProperty**, т.к. параметр **InputMode** меняет значения размеров вводимого кадра.

- **Включить/выключить ввод по каналам**

VOID IDInput (HDEVICE Hdv, UINT *UsesChannels);

Данная функция производит включение или выключение (разрешение и запрет) ввода по соответствующим каналам ввода изображения.

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

UsesChannels
Указатель на массив **UsesChannels[NumberChannels]**, где **NumberChannels** количество каналов поддерживаемых данным устройством. Каждый элемент массива может принимать следующие значения:
0 – канал выключен
1 – наивысший приоритет канала
2 – высокий приоритет канала
3 – нормальный приоритет канала
4 – низкий приоритет канала
5 – самый низкий приоритет канала

Если флаг PRIORITYENABLED возвращенный функцией IDGetDeviceProperty не установлен, для включения устройства следует установить нормальный приоритет ввода.

```
void StartAllChannel()
{
UINT MaxChannels;
UINT *UsesChannels;
UINT NumberDevices;

MaxChannels = IDGetMaxChannels();
UsesChannels = new UINT [MaxChannels];
for(int i=0;i<MaxChannels;i++) UsesChannels[i]=3;

NumberDevices  = IDGetNumberDevices();

for(int dv=0;dv<NumberDevices;dv++)
     IDInput(IDGetDevices(dv),
               &UsesChannels[IDGetFirstChannel(IDGetDevices(dv))]);

delete UsesChannels;
}
//*****************************************************************
void StopAllChannel()
{
UINT MaxChannels;
UINT *UsesChannels;
UINT NumberDevices;

MaxChannels = IDGetMaxChannels();
UsesChannels = new UINT [MaxChannels];
for(int i=0;i<MaxChannels;i++) UsesChannels[i]=0;

NumberDevices  = IDGetNumberDevices();

for(int dv=0;dv<NumberDevices;dv++)
     IDInput(IDGetDevices(dv),
               &UsesChannels[IDGetFirstChannel(IDGetDevices(dv))]);
```

```
delete UsesChannels;
}
```

- **Получить символьное описание устройства ввода.**

PCHAR IDGetName (HDEVICE Hdv);

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Результат:

Указатель на строку символов с именем устройства.

- **Получить символьное описание кода устройства ввода.**

PCHAR IDGetSN(HDEVICE Hdv);

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Результат:

Указатель на строку символов с описанием сериального номера устройства ввода.

- **Получить идентификатор устройства по номеру устройства**

HDEVICE IDGetDevices(UINT Number);

Параметры:

Number
Номер устройства ввода от нуля до значения определяемого по функции IDGetNumberDevices-1.

Результат:

Идентификатор устройства, возвращенный функцией **IDAdd** для конкретного устройства.

- **Получить номер первого канала для данного устройства**

UINT IDGetFirstChannel(HDEVICE Hdv);

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Результат:

Номер первого канала для данного устройства.

- **Получение свойств устройства ввода канала**

VOID IDGetCaptureProperty (
HDEVICE Hdv,
UINT Channel,
MPCaptureProperty Prop,
PROPERTYDATA *chProp);

Данная функция производит считывание свойств для каждого канала отдельного устройства.

Параметры

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Channel
Номер канала, из которого считываются свойства. Номер канала определяется только для данного устройства ввода и не соответствует номеру общего числа каналов при использовании нескольких устройств. При использовании одного устройства номера каналов устройства совпадают с общим номером канала системы.

Prop
Определяет свойство, которое необходимо считать. В следующей таблице приведен список параметров, которые могут быть прочитаны из устройства ввода изображений:

Свойство	Значение свойства
CAPTURE_BRIGHTNESS	Яркость
CAPTURE_EXPOSURE	Выдержка
CAPTURE_SHARPNESS	Четкость
CAPTURE_WHITE_BALANCE	Баланс белого
CAPTURE_HUE	Оттенок
CAPTURE_SATURATION	Цветность

CAPTURE_GAMMA	Гамма
CAPTURE_IRIS	Диафрагма
CAPTURE_FOCUS	Фокус
CAPTURE_ZOOM	Увеличение/уменьшение
CAPTURE_PAN	Смещение по горизонтали
CAPTURE_TILT	Наклон
CAPTURE_ROLL	Поворот
CAPTURE_SHUTTER	Затвор
CAPTURE_GAIN	Усиление
CAPTURE_BACKLIGHT	Компенсация освещенности
CAPTURE_TRIGGER_DELAY	Задержка триггера экспозиции
CAPTURE_FRAME_RATE	Режимы передачи кадров/сек
CAPTURE_FRAME_RATE_SLIDER	Плавное изменение передачи кадров/сек
CAPTURE_CONTRAST	Контрастность
CAPTURE_COLOR_MODE	Режим кодировки цветности изображения
CAPTURE_CHROMAU	Усиление цветовой компоненты U
CAPTURE_CHROMAV	Усиление цветовой компоненты V
CAPTURE_CUSTOM_FRAME	Задание размера кадра
CAPTURE_FRAME_MODE	Режим ввода кадра
CAPTURE_SETUP_DIALOG	Установочные диалоги производителя
CAPTURE_NAME	Имя канала
CAPTURE_SERIAL_NUMBER	Серийный номер устройства
CAPTURE_CROSSBAR	Время переключения между кадрами
CAPTURE_STANDARD	Видеостандарт
CAPTURE_FIELD_MODE	Режим ввода ТВ кадра

chProp

Указатель на структуру **PROPERTYDATA**, в которую будут помещены свойства канала.

```
typedef struct TPropertyData {
        bool Present;              // Поддержка свойства
        int  Value;                // Текущее значение
        int  DefaultValue;         // Значение по умолчанию
        int  MinValue;             // Минимальное значение свойства
        int  MaxValue;             // Максимальное значение свойства
        int  SteppingDelta;        // Шаг изменения значения
        bool AutoEnabled;          // Автоизменение возможно
        bool Auto;                 // Значение автоизменения
        bool OnOffEnabled;         // Включение/выключение возможно
        bool OnOff;                // Значение включения/выключения
        bool OnePushEnabled;       // Импульсное включение возможно
        bool OnePush;              // Значение состояния импульса
        void *Data;                // Указатель на дополнительные данные
      } PROPERTYDATA;
```

Результат:

0 – функция не выполнена, свойство не поддерживается устройством,
1 – функция выполнена успешно,
2 – интерфейс не поддерживается данным устройством.

Примечание:

Значение параметра структуры **PROPERTYDATA Present** определяет возможность использования свойства данным устройством. Если данный параметр имеет значение false, то свойство не используется.

При считывании свойства CAPTURE_COLOR_MODE параметр **Data** содержит указатель на массив **unsigned int ColorMode[32]**, в который будет помещены значения возможных цветовых режимов работы устройства:
RGBPIXELS 0x00000040 – Цветное RGB24 изображение (3 байта на пиксель)
UYVYPIXELS 0x00000080 – Цветное изображение с кодировкой U-Y-V-Y
GRAYPIXELS 0x00000100 – Полутоновое изображение
YUY2PIXELS 0x00000200 – Цветное изображение с кодировкой Y–U–Y-V
Y411PIXELS 0x00000400 – Цветное изображение с кодировкой Y411
Параметр **MaxValue** определяет количество возможных режимов.

При считывании свойства CAPTURE_FRAME_RATE параметр **Data** содержит указатель на массив **unsigned int FrameRate[32]**, в который будет помещены возможные значения периодов передачи кадров в микросекундах. Например, значение 40000 определяет скорость передачи 25 кадров в секунду.
Параметр **MaxValue** определяет количество возможных значений скоростей.

При считывании свойства CAPTURE_NAME параметр **Data** содержит указатель на строку **char Name[512]**, в которую будет помещено имя канала (устройства).

При считывании свойства CAPTURE_FRAME_MODE параметр **Data** содержит указатель на массив **unsigned int VideMode[32][4]**, в который будет помещены возможные значения режимов ввода.
Параметр **MaxValue** определяет количество возможных значений режимов ввода.
VideMode[n][0] – содержит значение цветового режима:
RGBPIXELS 0x00000040 – Цветное RGB24 изображение (3 байта на пиксель)
UYVYPIXELS 0x00000080 – Цветное изображение с кодировкой U-Y-V-Y
GRAYPIXELS 0x00000100 – Полутоновое изображение
YUY2PIXELS 0x00000200 – Цветное изображение с кодировкой Y–U–Y-V
Y411PIXELS 0x00000400 – Цветное изображение с кодировкой Y411
VideMode[n][1] – размер в пикселях по горизонтали.
VideMode[n][2] – размер в пикселях по вертикали.
VideMode[n][3] – параметр пикселя в изображении:
SQUARE_PIXELS – квадратные пиксели;
WIDE_PIXELS – широкие пиксели.

При считывании свойства CAPTURE_SERIAL_NUMBER параметр **Data** содержит указатель на массив **unsigned int SerNumber[32]**, в который будет помещено значение серийного номера устройства.
Параметр **MaxValue** определяет количество слов, определяющих серийный номер канала.

При считывании свойства CAPTURE_FIELD_MODE параметр **Value** содержит слово, биты которого означают следующее:
Бит 0 – 0 – ширина изображения равна половине ширины строки (кадратные пиксели);
1 – ширина изображения равна ширине строки (широкие пиксели).
Бит 1 - 0 – ввод одного полукадра;
1- ввод двух полукадров, один за другим.
Бит 2 - 0 – синхронизация по полукадрам включена;
1- синхронизация по полукадрам выключена.
Бит 3 - 0 – чрезстрочный режим выключен;
1 – чрезстрочный режим включен.
Данное свойство поддерживается только с ТВ устройствами ввода.

При считывании свойства CAPTURE_STANDARD параметр **Value** содержит следующие значения:
TVSTANDARD_NONE 0
TVSTANDARD_PAL 1
TVSTANDARD_NTSC 2
TVSTANDARD_SECAM 3
Если свойство равно TVSTANDARD_NONE это означает, что устройство с прогрессивной разверткой или подключено цифровое устройство. При этом свойство CAPTURE_FIELD_MODE недоступно.

- **Установить свойство устройства ввода канала**

VOID IDSetCaptureProperty (
HDEVICE Hdv,
UINT Channel,
MPCaptureProperty Prop,
PROPERTYDATA chProp);

Данная функция производит установку параметров ввода для каждого канала отдельного устройства.

Результат:

0 – функция не выполнена, свойство не поддерживается устройством,
1 – функция выполнена успешно,
2 – интерфейс не поддерживается данным устройством.

Параметры

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Channel
Номер устанавливаемого канала. Номер канала определяется только для данного устройства ввода и не соответствует номеру общего числа каналов при использовании нескольких устройств. При использовании одного устройства номера каналов устройства совпадают с общим номером канала системы.

Prop
Определяет свойство, которое необходимо установить. В следующей таблице приведен список параметров, которые могут быть установлены на устройстве ввода изображений:

Свойство	Значение свойства
CAPTURE_BRIGHTNESS	Яркость
CAPTURE_EXPOSURE	Выдержка
CAPTURE_SHARPNESS	Четкость
CAPTURE_WHITE_BALANCE	Баланс белого
CAPTURE_HUE	Оттенок
CAPTURE_SATURATION	Цветность
CAPTURE_GAMMA	Гамма
CAPTURE_IRIS	Диафрагма
CAPTURE_FOCUS	Фокус
CAPTURE_ZOOM	Увеличение/уменьшение
CAPTURE_PAN	Смещение по горизонтали
CAPTURE_TILT	Наклон
CAPTURE_ROLL	Поворот
CAPTURE_SHUTTER	Затвор
CAPTURE_GAIN	Усиление
CAPTURE_BACKLIGHT	Компенсация освещенности
CAPTURE_TRIGGER_DELAY	Задержка триггера экспозиции
CAPTURE_FRAME_RATE	Режимы передачи кадров/сек
CAPTURE_FRAME_RATE_SLIDER	Плавное изменение передачи кадров/сек
CAPTURE_CONTRAST	Контрастность
CAPTURE_COLOR_MODE	Режим кодировки цветности изображения
CAPTURE_CHROMAU	Усиление цветовой компоненты U
CAPTURE_CHROMAV	Усиление цветовой компоненты V
CAPTURE_CUSTOM_FRAME	Задание размера кадра
CAPTURE_FRAME_MODE	Режим ввода кадра
CAPTURE_SETUP_DIALOG	Установочные диалоги производителя
CAPTURE_NAME	Имя канала
CAPTURE_SERIAL_NUMBER	Серийный номер устройства
CAPTURE_CROSSBAR	Время переключения между кадрами
CAPTURE_STANDARD	Видеостандарт
CAPTURE_FIELD_MODE	Режим ввода ТВ кадра

chProp

Структура **PROPERTYDATA**, которая содержит пареметры необходимые для установки соответствующего свойства. Структура заполняется по команде **IDGetCaptureProperty.**

```
typedef struct TPropertyData {
    bool Present;           // Поддержка свойства
    int  Value;             // Текущее значение
    int  DefaultValue;      // Значение по умолчанию
    int  MinValue;          // Минимальное значение свойства
    int  MaxValue;          // Максимальное значение свойства
    int  SteppingDelta;     // Шаг изменения значения
    bool AutoEnabled;       // Автоизменение возможно
    bool Auto;              // Значение автоизменения
    bool OnOffEnabled;      // Включение/выключение возможно
    bool OnOff;             // Значение включения/выключения
    bool OnePushEnabled;    // Импульсное включение возможно
    bool OnePush;           // Значение состояния импульса
    void *Data;             // Указатель на дополнительные данные
  } PROPERTYDATA;
```

Для установки соответствующего свойства необходимо определить следующие параметры структуры: **Value, Auto, OnOff, OnePush.**

Примечание:

Параметр может быть установлен если значение параметра структуры **Present** имеет значение true и параметр **Auto** имеет значение false.

При установке свойства CAPTURE_COLOR_MODE параметр **Value** должен содержать значение режима т.е. одно из следующих значений:

RGBPIXELS	0x00000040 – Цветное RGB24 изображение (3 байта на пиксель)
UYVYPIXELS	0x00000080 – Цветное изображение с кодировкой U-Y-V-Y
GRAYPIXELS	0x00000100 – Полутоновое изображение
YUY2PIXELS	0x00000200 – Цветное изображение с кодировкой Y–U–Y-V

При установке свойства CAPTURE_FRAME_RATE и CAPTURE_FRAME_MODE параметр **Value** должен содержать индекс значения полученного при вызове функции **IDGetCaptureProperty** в параметре **Data**.

Свойства CAPTURE_SERIAL_NUMBER и CAPTURE_NAME не могут быть изменены.

При установке свойства CAPTURE_FIELD_MODE параметр **Value** содержит слово, биты которого означают следующее:
Бит 0 – 0 – ширина изображения равна половине ширины строки (кадратные пиксели);
1 – ширина изображения равна ширине строки (широкие пиксели).
Бит 1 - 0 – ввод одного полукадра;
1- ввод двух полукадров, один за другим.
Бит 2 - 0 – синхронизация по полукадрам включена;
1- синхронизация по полукадрам выключена.
Бит 3 - 0 – чрезстрочный режим выключен;
1 – чрезстрочный режим включен.
Данное свойство поддерживается только с ТВ устройствами ввода.

При установке свойства CAPTURE_STANDARD параметр **Value** содержит следующие значения:
TVSTANDARD_PAL 1
TVSTANDARD_NTSC 2
TVSTANDARD_SECAM 3
Свойство доступно только для ТВ устройств.

- **Определение числа возможных каналов обработки**

UINT IDGetMaxEnabledChannels(UINT type);

Данная функция производит определение числа каналов максимально доступных для осуществления обработки различными алгоритмами.

Параметры

type
Идентификатор типа обработки:

Тип	Величина	Значение
CFENABLED	0x00000001	Разрешено работать с функциями определения автомобильных номеров.
MDENABLED	0x00000002	Разрешено работать с функциями детекции движения.
TFENABLED	0x00000004	Разрешено работать с функциями определения номеров железнодорожных вагонов.

Результат:

Максимальное количество каналов обработки для данного типа.

4.2. Функции обслуживания виртуальных устройств захвата изображений с использованием номера канала

При вызове данной группы функций в качестве идентификатора используется только номер канала в полном списке каналов всех активированных устройств. Определение устройства соответствующего каналу определяется автоматически. Максимальное число каналов определяется функцией **IDGetMaxChannels.**

- **Получение свойств отдельного канала устройства ввода**

VOID CHGetProperty(UINT Channel, TCHANNELPROPERTY CHProp);

После инициализации устройства данная функция возвращает свойства отдельных каналов устройства ввода.

Параметры:

Channel
Номер канала, для которого необходимо произвести чтение свойств. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

CHProp
Указатель на структуру, которая содержит свойства канала устройства ввода.

```
typedef struct TChannelProperty {
        UINT HorFrameSize;
        UINT VerFrameSize;
        UINT PixelMode;
        UINT ColorCode;
        UINT ChannelFlags;
        UINT Reserved;
      } TCHANNELPROPERTY;
```

HorFrameSize
Размер в пикселях вводимого кадра по горизонтали.

VerFrameSize
Размер в пикселях вводимого кадра по вертикали.

PixelMode
Свойства кадра.

Определение	Величина	Значение
WIDE_PIXELS	0x00000002	Каждый кадр (или полукадр при установленном флаге DOUBLEFIELD_PIXELS) содержит в ширину удвоенные размеры.

SQUARE_PIXELS	0x00000002	Каждый кадр (или полукадр при установленном флаге DOUBLEFIELD_PIXELS) содержит в ширину квадратные пиксели.
DOUBLEFIELD_PIXELS	0x00000004	Кадр содержит два полукадра, которые следуют один за другим.

ColorCode
Цветовая кодировка каждого пикселя. Может принимать одно из следующих значений:

MAKEFOURCC('G','R','A','Y'); - полутоновое изображение (1 байт на пиксель).
MAKEFOURCC('U','Y','V','Y'); - цветное изображение в формате UYVY (2 байта на пиксель).
MAKEFOURCC('Y','U','Y','2'); - цветное изображение в формате YUY2 (2 байта на пиксель).
MAKEFOURCC ('R','G','B',' '); - цветное изображение в формате RGB24 (3 байта на пиксель).

Где: MAKEFOURCC(ch0, ch1, ch2, ch3) \
((DWORD)(BYTE)(ch0) | ((DWORD)(BYTE)(ch1) << 8) | \
((DWORD)(BYTE)(ch2) << 16) | ((DWORD)(BYTE)(ch3) << 24))

ChannelFlags
Возможность работы с функциями библиотеки.

Определение	Величина	Значение
CFENABLED	0x00000001	Разрешено работать с функциями определения автомобильных номеров.
MDENABLED	0x00000002	Разрешено работать с функциями детекции движения.
TFENABLED	0x00000004	Разрешено работать с функциями определения номеров железнодорожных вагонов.
NSENABLED	0x00000000	Разрешено работать только с функциями ввода изображений.

Reserved
Зарезервировано для дальнейшего развития системы.

```
HDEVICE Hdv[2];
UINT MaxChannels;
TCHANNELPROPERTY CProp;
char *SaveImageT[MAXCHANNELS];
void *SaveImage[MAXCHANNELS];

MPCreate();
Hdv[0] = IDAdd("C:\\Program File\\MegaLib\\DEVICE\\MegaFrame4.DLL",0); //4
Hdv[1] = IDAdd("C:\\Program File\\MegaLib\\DEVICE\\MegaFrameX.DLL",0); //8
```

```
MaxChannels = IDGetMaxChannels(); // MaxChannels = 12

for(int i=0;i<MaxChannels;i++){
     CHGetProperty(i,&CProp);
     UINT cl;
     if(CProp.ColorCode ==  MAKEFOURCC('G','R','A','Y')) cl=1;
     if(CProp.ColorCode ==  MAKEFOURCC('R','G','B',' ')) cl=3;
     if(CProp.ColorCode ==  MAKEFOURCC('U','Y','V','Y')) cl=2;
     if(CProp.ColorCode ==  MAKEFOURCC('Y','U','Y','2')) cl=2;
     SaveImageT[i]= new char[CProp.HorFrameSize*CProp.VerFrameSize*cl+256];
     SaveImage[i]=(PVOID)((UINT)(SaveImageT[i]+0x100)&0xffffff00);
     MaxChannels++;
     }


IDCloseAll();
MPRelease();
```

- **Получить кадры изображения**

VOID CHGetPicture(UINT *ImgTegg, VOID **Images);

Вызов данной функции должен производиться непрерывно и по возможности с максимальной интенсивностью. Рекомендуется использование потока с высоким приоритетом. Функция может не вызываться в случае отключения всех каналов.

Параметры:

ImgTegg
Параметр инициализируется пользователем и заполняется функцией. Указатель на массив **ImgTegg[MaxNumberChannels]**, где **MaxNumberChannels** количество каналов поддерживаемых всеми устройствами. Максимальное число каналов определяется функцией **IDGetMaxChannels.** Каждый элемент массива может принимать следующие значения:

Определение	Величина	Значение
NOPICTURE	0x00000000	Готовности кадра нет
PICTUREREADY	0x00000001	Готовность кадра есть
INPUTUNCONNECTED	0x00000002	Произошло отключение канала от видеосигнала

Images
Параметр инициализируется пользователем и заполняется функцией. Указатель на массив указателей изображений каналов **Images[MaxNumberChannels]**, где **MaxNumberChannels** количество каналов поддерживаемых всеми устройствами. Максимальное число каналов определяется функцией **IDGetMaxChannels.** Размер каждого буфера должен соответствовать запрошенному изображению, который определяется при чтении свойств каналов**.** Каждый буфер должен быть выровнен до 256 байт для правильной работы SSE2 инструкций**.**

```
void GetImagesFromDevice()
{
UINT MaxChannels;
UINT *ImgTegg;

MaxChannels = IDGetMaxChannels();
ImgTegg = new UINT [MaxChannels];

CHGetPicture(ImgTegg,SaveImage);

for(int i=0;i<MaxChannels;i++)
     switch(ImgTegg[i]){
        case NOPICTURE : break;
        case PICTUREREADY : LibRun(&i);/*обработка буфера*/ break;
        case INPUTUNCONNECTED : /*обработка отключения канала*/ break;
     }

delete ImgTegg;
}
```

- **Установить параметры канала.**

VOID CHSetParams (UINT Channel, TINPUTPARAMS *Params);

Данная функция производит установку параметров ввода: яркость, контрастность, цветность, коррекцию, режим цвета и разрешение ввода для всех каналов системы. Номер канала определяется в пределах от нуля до максимального числа каналов. Функция применяется с ограниченным числом типов устройств и для более эффективного использования устройств захвата изображений следует использовать функции **CHSetCaptureProperty, CHGetCaptureProperty**.

Параметры

Channel
Номер канала, для которого необходимо произвести установку параметров. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

Params
Указатель на структуру параметров канала

typedef struct TInputParam {
UINT ColorMode; - 0 – черно/белый; 1 - цветной
UINT InputMode;
UINT Brightness; - яркость 0 - 255
UINT Contrast; - контрастность 0-255
UINT ChromaV; - яркость V 0-255
UINT ChromaU; - яркость U 0-255
UINT Hue; - баланс 0-255
} TINPUTPARAMS;

InputMode
Параметр, определяющий тип вводимого изображения:
0 – один полукадр низкого разрешения с учетом полей
1 – один полукадр высокого разрешения с учетом полей
2 – два полукадра низкого разрешения с учетом полей
3 – два полукадра высокого разрешения с учетом полей
4 – один полукадр низкого разрешения без учета полей
5 – один полукадр высокого разрешения без учета полей
6 – два полукадра низкого разрешения без учета полей
7 – два полукадра высокого разрешения без учета полей

Примечание:

Параметр **InputMode** при использовании библиотеки с обработкой двух полукадров следует установить в режим 7. При работе с одним полукадром данный параметр устанавливается в значение 1 для 25 кадров/с или 5 для 50 кадров/с.

Функция **CHSetParams** должна быть выполнена раньше функции **CHGetChannelProperty**, т.к. параметр **InputMode** меняет значения размеров вводимого кадра.

- **Включить/выключить ввод по каналам**

VOID CHInput (UINT *UsesChannels);

Данная функция производит включение или выключение (разрешение и запрет) ввода по соответствующим каналам ввода изображения.

Параметры:

UsesChannels
Указатель на массив **UsesChannels[MaxChannels]**, где **MaxChannels** количество каналов поддерживаемых всеми устройствами. Максимальное число каналов определяется функцией **IDGetMaxChannels.**
Каждый элемент массива может принимать следующие значения:
0 – канал выключен
1 – наивысший приоритет канала
2 – высокий приоритет канала
3 – нормальный приоритет канала
4 – низкий приоритет канала
5 – самый низкий приоритет канала

Если флаг PRIORITYENABLED возвращенный функцией IDGetDeviceProperty не установлен, для включения устройства следует установить нормальный приоритет ввода.

```
void StartAllChannel()
{
UINT MaxChannels;
UINT *UsesChannels;

MaxChannels = IDGetMaxChannels();
UsesChannels = new UINT [MaxChannels];
for(int i=0;i<MaxChannels;i++) UsesChannels[i]=3;

CHInput(UsesChannels);

delete UsesChannels;
}
//*****************************************************************
void StopAllChannel()
{
UINT MaxChannels;
UINT *UsesChannels;

MaxChannels = IDGetMaxChannels();
UsesChannels = new UINT [MaxChannels];
for(int i=0;i<MaxChannels;i++) UsesChannels[i]=0;

CHInput(UsesChannels);

delete UsesChannels;
}
```

- **Включить ввод по конкретному каналу**

VOID CHStart (UINT Channel, UINT Priority);

Данная функция производит включение ввода по конкретному каналу системы.

Параметры:

Channel
Номер канала, для которого необходимо произвести старт ввода. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

Priority
Значение приоритета по данному каналу:
0 – канал выключен
1 – наивысший приоритет канала
2 – высокий приоритет канала
3 – нормальный приоритет канала
4 – низкий приоритет канала
5 – самый низкий приоритет канала

Если флаг PRIORITYENABLED возвращенный функцией IDGetDeviceProperty не установлен, для включения устройства следует установить нормальный приоритет ввода.

- **Выключить ввод по конкретному каналу**

VOID CHStop (UINT Channel);

Данная функция производит выключение ввода по конкретному каналу системы.

Параметры:

Channel
Номер канала, для которого необходимо произвести останов ввода. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

- **Получить идентификатор устройства по номеру канала**

HDEVICE CHGetDevices(UINT Channel);

Параметры:

Channel
Номер канала, для которого необходимо определить идентификатор устройства. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

Результат:

Идентификатор устройства, возвращенный функцией **IDAdd** для конкретного канала.

- **Получение свойств устройства ввода по номеру канала**

VOID CHGetCaptureProperty (
UINT Channel,
MPCaptureProperty Prop,
PROPERTYDATA *chProp);

Данная функция производит считывание свойств устройства ввода по номеру канала.

Параметры

Channel
Номер канала, из которого считываются свойства. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

Prop

Определяет свойство, которое необходимо считать. В следующей таблице приведен список параметров, которые могут быть прочитаны из устройства ввода изображений:

Свойство	Значение свойства
CAPTURE_BRIGHTNESS	Яркость
CAPTURE_EXPOSURE	Выдержка
CAPTURE_SHARPNESS	Четкость
CAPTURE_WHITE_BALANCE	Баланс белого
CAPTURE_HUE	Оттенок
CAPTURE_SATURATION	Цветность
CAPTURE_GAMMA	Гамма
CAPTURE_IRIS	Диафрагма
CAPTURE_FOCUS	Фокус
CAPTURE_ZOOM	Увеличение/уменьшение
CAPTURE_PAN	Смещение по горизонтали
CAPTURE_TILT	Наклон
CAPTURE_ROLL	Поворот
CAPTURE_SHUTTER	Затвор
CAPTURE_GAIN	Усиление
CAPTURE_BACKLIGHT	Компенсация освещенности
CAPTURE_TRIGGER_DELAY	Задержка триггера экспозиции
CAPTURE_FRAME_RATE	Режимы передачи кадров/сек
CAPTURE_FRAME_RATE_SLIDER	Плавное изменение передачи кадров/сек
CAPTURE_CONTRAST	Контрастность
CAPTURE_COLOR_MODE	Режим кодировки цветности изображения
CAPTURE_CHROMAU	Усиление цветовой компоненты U
CAPTURE_CHROMAV	Усиление цветовой компоненты V
CAPTURE_CUSTOM_FRAME	Задание размера кадра
CAPTURE_FRAME_MODE	Режим ввода кадра
CAPTURE_SETUP_DIALOG	Установочные диалоги производителя
CAPTURE_NAME	Имя канала
CAPTURE_SERIAL_NUMBER	Серийный номер устройства
CAPTURE_CROSSBAR	Время переключения между кадрами
CAPTURE_STANDARD	Видеостандарт
CAPTURE_FIELD_MODE	Режим ввода ТВ кадра

chProp

Указатель на структуру **PROPERTYDATA**, в которую будут помещены свойства канала.

```
typedef struct TPropertyData {
        bool Present;           // Поддержка свойства
        int  Value;             // Текущее значение
```

```
    int  DefaultValue;          // Значение по умолчанию
    int  MinValue;              // Минимальное значение свойства
    int  MaxValue;              // Максимальное значение свойства
    int  SteppingDelta;         // Шаг изменения значения
    bool AutoEnabled;           // Автоизменение возможно
    bool Auto;                  // Значение автоизменения
    bool OnOffEnabled;          // Включение/выключение возможно
    bool OnOff;                 // Значение включения/выключения
    bool OnePushEnabled;        // Импульсное включение возможно
    bool OnePush;               // Значение состояния импульса
    void *Data;                 // Указатель на дополнительные данные
  } PROPERTYDATA;
```

Результат:

0 – функция не выполнена, свойство не поддерживается устройством,
1 – функция выполнена успешно,
2 – интерфейс не поддерживается данным устройством.

Примечание:

Значение параметра структуры **PROPERTYDATA Present** определяет возможность использования свойства данным устройством. Если данный параметр имеет значение false, то свойство не используется.

При считывании свойства CAPTURE_COLOR_MODE параметр **Data** содержит указатель на массив **unsigned int ColorMode[32]**, в который будет помещены значения возможных цветовых режимов работы устройства:
RGBPIXELS 0x00000040 – Цветное RGB24 изображение (3 байта на пиксель)
UYVYPIXELS 0x00000080 – Цветное изображение с кодировкой U-Y-V-Y
GRAYPIXELS 0x00000100 – Полутоновое изображение
YUY2PIXELS 0x00000200 – Цветное изображение с кодировкой Y–U–Y-V
Параметр **MaxValue** определяет количество возможных режимов.

При считывании свойства CAPTURE_FRAME_RATE параметр **Data** содержит указатель на массив **unsigned int FrameRate[32]**, в который будет помещены возможные значения периодов передачи кадров в микросекундах. Например, значение 40000 определяет скорость передачи 25 кадров в секунду.
Параметр **MaxValue** определяет количество возможных значений скоростей.

При считывании свойства CAPTURE_NAME параметр **Data** содержит указатель на строку **char Name[512]**, в которую будет помещено имя канала (устройства).

При считывании свойства CAPTURE_FRAME_MODE параметр **Data** содержит указатель на массив **unsigned int VideMode[32][4]**, в который будет помещены возможные значения режимов ввода.
Параметр **MaxValue** определяет количество возможных значений режимов ввода.
VideMode[n][0] – содержит значение цветового режима:
RGBPIXELS 0x00000040 – Цветное RGB24 изображение (3 байта на пиксель)

UYVYPIXELS 0x00000080 – Цветное изображение с кодировкой U-Y-V-Y
GRAYPIXELS 0x00000100 – Полутоновое изображение
YUY2PIXELS 0x00000200 – Цветное изображение с кодировкой Y–U–Y-V
VideMode[n][1] – размер в пикселях по горизонтали.
VideMode[n][2] – размер в пикселях по вертикали.
VideMode[n][3] – параметр пикселя в изображении:
SQUARE_PIXELS – квадратные пиксели;
WIDE_PIXELS – широкие пиксели.

При считывании свойства CAPTURE_SERIAL_NUMBER параметр **Data** содержит указатель на массив **unsigned int SerNumber[32]**, в который будет помещено значение серийного номера устройства.
Параметр **MaxValue** определяет количество слов, определяющих серийный номер устройства.

При считывании свойства CAPTURE_FIELD_MODE параметр **Value** содержит слово, биты которого означают следующее:
Бит 0 – 0 – ширина изображения равна половине ширины строки (кадратные пиксели);
1 – ширина изображения равна ширине строки (широкие пиксели).
Бит 1 - 0 – ввод одного полукадра;
1- ввод двух полукадров, один за другим.
Бит 2 - 0 – синхронизация по полукадрам включена;
1- синхронизация по полукадрам выключена.
Бит 3 - 0 – чрезстрочный режим выключен;
1 – чрезстрочный режим включен.
Данное свойство поддерживается только с ТВ устройствами ввода.

При считывании свойства CAPTURE_STANDARD параметр **Value** содержит следующие значения:
TVSTANDARD_NONE 0
TVSTANDARD_PAL 1
TVSTANDARD_NTSC 2
TVSTANDARD_SECAM 3
Если свойство равно TVSTANDARD_NONE это означает, что устройство с прогрессивной разверткой или подключено цифровое устройство. При этом свойство CAPTURE_FIELD_MODE недоступно.

- **Установить свойство устройства ввода по номеру канала**

VOID CHSetCaptureProperty (
UINT Channel,
MPCaptureProperty Prop,
PROPERTYDATA chProp);

Данная функция производит установку параметров ввода для каждого канала по его номеру среди всех открытых устройств.

Параметры

Channel
Номер канала, из которого считываются свойства. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

Prop
Определяет свойство, которое необходимо установить. В следующей таблице приведен список параметров, которые могут быть установлены на устройстве ввода изображений:

Свойство	Значение свойства
CAPTURE_BRIGHTNESS	Яркость
CAPTURE_EXPOSURE	Выдержка
CAPTURE_SHARPNESS	Четкость
CAPTURE_WHITE_BALANCE	Баланс белого
CAPTURE_HUE	Оттенок
CAPTURE_SATURATION	Цветность
CAPTURE_GAMMA	Гамма
CAPTURE_IRIS	Диафрагма
CAPTURE_FOCUS	Фокус
CAPTURE_ZOOM	Увеличение/уменьшение
CAPTURE_PAN	Смещение по горизонтали
CAPTURE_TILT	Наклон
CAPTURE_ROLL	Поворот
CAPTURE_SHUTTER	Затвор
CAPTURE_GAIN	Усиление
CAPTURE_BACKLIGHT	Компенсация освещенности
CAPTURE_TRIGGER_DELAY	Задержка триггера экспозиции
CAPTURE_FRAME_RATE	Режимы передачи кадров/сек
CAPTURE_FRAME_RATE_SLIDER	Плавное изменение передачи кадров/сек
CAPTURE_CONTRAST	Контрастность
CAPTURE_COLOR_MODE	Режим кодировки цветности изображения
CAPTURE_CHROMAU	Усиление цветовой компоненты U
CAPTURE_CHROMAV	Усиление цветовой компоненты V
CAPTURE_CUSTOM_FRAME	Задание размера кадра
CAPTURE_FRAME_MODE	Режим ввода кадра
CAPTURE_SETUP_DIALOG	Установочные диалоги производителя
CAPTURE_NAME	Имя канала
CAPTURE_SERIAL_NUMBER	Серийный номер устройства
CAPTURE_CROSSBAR	Время переключения между кадрами
CAPTURE_STANDARD	Видеостандарт
CAPTURE_FIELD_MODE	Режим ввода ТВ кадра

chProp

Структура **PROPERTYDATA**, которая содержит пареметры необходимые для установки соответствующего свойства. Структура заполняется по команде **IDGetCaptureProperty.**

```
typedef struct TPropertyData {
    bool Present;           // Поддержка свойства
    int  Value;             // Текущее значение
    int  DefaultValue;      // Значение по умолчанию
    int  MinValue;          // Минимальное значение свойства
    int  MaxValue;          // Максимальное значение свойства
    int  SteppingDelta;     // Шаг изменения значения
    bool AutoEnabled;       // Автоизменение возможно
    bool Auto;              // Значение автоизменения
    bool OnOffEnabled;      // Включение/выключение возможно
    bool OnOff;             // Значение включения/выключения
    bool OnePushEnabled;    // Импульсное включение возможно
    bool OnePush;           // Значение состояния импульса
    void *Data;             // Указатель на дополнительные данные
  } PROPERTYDATA;
```

Для установки соответствующего свойства необходимо определить следующие параметры структуры: **Value, Auto, OnOff, OnePush.**

Результат:

0 – функция не выполнена, свойство не поддерживается устройством,
1 – функция выполнена успешно,
2 – интерфейс не поддерживается данным устройством.

Примечание:

Параметр может быть установлен если значение параметра структуры **Present** имеет значение true и параметр **Auto** имеет значение false.
При установке свойства CAPTURE_COLOR_MODE параметр **Value** должен содержать значение режима т.е. одно из следующих значений:

RGBPIXELS	0x00000040 – Цветное RGB24 изображение (3 байта на пиксель)
UYVYPIXELS	0x00000080 – Цветное изображение с кодировкой U-Y-V-Y
GRAYPIXELS	0x00000100 – Полутоновое изображение
YUY2PIXELS	0x00000200 – Цветное изображение с кодировкой Y–U–Y-V

При установке свойства CAPTURE_FRAME_RATE и CAPTURE_FRAME_MODE параметр **Value** должен содержать индекс значения полученного при вызове функции **IDGetCaptureProperty** в параметре **Data**.

Свойства CAPTURE_SERIAL_NUMBER и CAPTURE_NAME не могут быть изменены.

При установке свойства CAPTURE_FIELD_MODE параметр **Value** содержит слово, биты которого означают следующее:

Бит 0 – 0 – ширина изображения равна половине ширины строки (кадратные пиксели);
1 – ширина изображения равна ширине строки (широкие пиксели).

Бит 1 - 0 – ввод одного полукадра;
1- ввод двух полукадров, один за другим.

Бит 2 - 0 – синхронизация по полукадрам включена;
1- синхронизация по полукадрам выключена.

Бит 3 - 0 – чрезстрочный режим выключен;
1 – чрезстрочный режим включен.

Данное свойство поддерживается только с ТВ устройствами ввода.

При установке свойства CAPTURE_STANDARD параметр **Value** содержит следующие значения:

TVSTANDARD_PAL 1
TVSTANDARD_NTSC 2
TVSTANDARD_SECAM 3

Свойство доступно только для ТВ устройств.

- **Определение идентификатора устройства по номеру канала**

HDEVICE CHGetVideoDevice(UINT VideoChannel);

Параметры:

VideoChannel
Номер канала, для которого необходимо определить идентификатор устройства. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

Результат:

Идентификатор устройства, возвращенный функцией **IDAdd** для конкретного канала.

4.3. Функции обслуживания виртуальных устройств захвата аудио данных с использованием идентификатора устройства

- **Определение общего числа аудио каналов в системе.**

UINT IDGetMaxAudioChannels (void);

Результат:

Общее число аудио каналов определенных в системе.

- **Получение свойств аудио канала устройства ввода**

VOID IDGetChannelAudioProperty(
HDEVICE Hdv,
UINT AudioChannel,
TAUDIOPROPERTY *CHAProp);

После инициализации устройства данная функция возвращает свойства отдельных каналов устройства ввода.

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetAudioDevice**.

AudioChannel
Номер канала, для которого необходимо произвести чтение свойств. Номер канала определяется только для данного устройства ввода и не соответствует номеру общего числа каналов при использовании нескольких устройств. При использовании одного устройства номера каналов устройства совпадают с общим номером канала системы.

CHAProp
Указатель на структуру, которая содержит свойства канала аудио ввода.

```
typedef struct TAudioProperty {
    WORD  wFormatTag;
    WORD  nChannels;
    DWORD nSamplesPerSec;
    DWORD nAvgBytesPerSec;
    WORD  nBlockAlign;
    WORD  wBitsPerSample;
    WORD  cbSize;
    } TAUDIOPROPERTY;
```

Структура полностью совпадает со структурой WAVEFORMATEX, которая используетя в DirectX.

wFormatTag

Формат аудио инормации. Данная переменная всегда равна 1.

nChannels

Количество аудио каналов. Для всех режимов данная переменная равна 1. В случае использования устройств, которые вводят стерео данные, библиотека вводит отдельно левый и правый каналы как два канала в моно режиме.

nSamplesPerSec

Количество цифровых отсчетов в секунду (Герц).

nAvgBytesPerSec

Количество байт в секунду.

nBlockAlign

Выравнивание данных в блоке.

wBitsPerSample

Количество бит в отчете.

cbSize

Данная переменная не используется и должна быть равна 0.

- **Получить аудио данные устройства**

VOID IDGetAudioBuffer(HDEVICE Hdv, UINT *AudioTegg, VOID **AudioBuffers);

Вызов данной функции должен производиться непрерывно и по возможности с максимальной интенсивностью. Рекомендуется использование потока с высоким приоритетом. Функция может не вызываться в случае отключения всех каналов.

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetAudioDevice**.

AudioTegg

Параметр инициализируется пользователем и заполняется функцией. Указатель на массив **AudioTegg[NumberAudioChannels]**, где **NumberAudioChannels** количество каналов поддерживаемых данным устройством, который определен в структуре **TDEVICEPROPERTY**. Каждый элемент массива может принимать следующие значения:

Определение	Величина	Значение
DATA_NOT_READY	0x00000000	Готовности блока нет
DATA_READY	0x00000001	Готовность блока есть

AudioBuffers
Указатель на массив указателей буферов аудио каналов **AudioBuffers[NumberAudioChannels]**, где **NumberAudioChannels** количество каналов поддерживаемых данным устройством. Размер каждого буфера должен составлять 4096 байт**.**

- **Включить/выключить ввод аудио данных по каналам**

VOID IDInputAudio(HDEVICE Hdv, UINT *UsesAudioChannels);

Данная функция производит включение или выключение (разрешение и запрет) ввода по соответствующим каналам ввода аудио данных.

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetAudioDevice**.

UsesAudioChannels
Указатель на массив **UsesAudioChannels[NumberAudioChannels]**, где **NumberAudioChannels** количество аудио каналов поддерживаемых данным устройством. Каждый элемент массива может принимать следующие значения:
0 – канал выключен
1 – канал включен

- **Получение свойств устройства аудио ввода канала**

VOID IDGetAudioProperty (
HDEVICE Hdv,
UINT AudioChannel,
MPAudioProperty Prop,
PROPERTYDATA *chProp);

Данная функция производит считывание свойств для каждого аудио канала отдельного устройства.

Параметры

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetAudioDevice**.

AudioChannel
Номер канала, из которого считываются свойства. Номер канала определяется только для данного устройства ввода и не соответствует номеру общего числа каналов при использовании нескольких устройств. При использовании одного устройства номера каналов устройства совпадают с общим номером канала системы.

Prop
Определяет свойство, которое необходимо считать. В следующей таблице приведен список параметров, которые могут быть прочитаны из устройства ввода аудио данных:

Свойство	Значение свойства
AUDIOCAPTURE_ENABLED	Разрешение/запрещение работы канала
AUDIOCAPTURE_MONO	Включение/выключение режима МОНО
AUDIOCAPTURE_RECORDLEVEL	Уровень записи
AUDIOCAPTURE_PAN	Баланс левого и правого каналов
AUDIOCAPTURE_LOUDNESS	Включение/выключение звука
AUDIOCAPTURE_TREBLE	Уровень высоких частот
AUDIOCAPTURE_BASS	Уровень низких частот
AUDIOCAPTURE_CAPTUREMODE	Режим ввода
AUDIOCAPTURE_NAME	Имя канала
AUDIOCAPTURE_SERIAL_NUMBER	Серийный номер канала

chProp
Указатель на структуру **PROPERTYDATA**, в которую будут помещены свойства канала.

```
typedef struct TPropertyData {
    bool Present;          // Поддержка свойства
    int  Value;            // Текущее значение
    int  DefaultValue;     // Значение по умолчанию
    int  MinValue;         // Минимальное значение свойства
    int  MaxValue;         // Максимальное значение свойства
    int  SteppingDelta;    // Шаг изменения значения
    bool AutoEnabled;      // Автоизменение возможно
    bool Auto;             // Значение автоизменения
    bool OnOffEnabled;     // Включение/выключение возможно
    bool OnOff;            // Значение включения/выключения
    bool OnePushEnabled;   // Импульсное включение возможно
    bool OnePush;          // Значение состояния импульса
```

```
    void *Data;                    // Указатель на дополнительные данные
} PROPERTYDATA;
```

Результат:

0 – функция не выполнена, свойство не поддерживается устройством,
1 – функция выполнена успешно,
2 – интерфейс не поддерживается данным устройством.

Примечание:

Значение параметра структуры **PROPERTYDATA Present** определяет возможность использования свойства данным устройством. Если данный параметр имеет значение **false**, то свойство не используется.

При считывании свойства AUDIOCAPTURE_NAME параметр **Data** содержит указатель на строку **char Name[512]**, в которую будет помещено имя каналатва).

При считывании свойства AUDIOCAPTURE_CAPTUREMODE параметр **Data** содержит указатель на массив **unsigned int AudioMode[32][4]**, в который будет помещены возможные значения режимов ввода.
Параметр **MaxValue** определяет количество возможных значений режимов ввода.
AudioMode[n][0] – количество отчетов в секунду (Герц);
AudioMode[n][1] – количество каналов.
AudioMode[n][2] – выравнивание данных.
AudioMode[n][3] – количество бит на отчет.

При считывании свойства AUDIOCAPTURE_SERIAL_NUMBER параметр **Data** содержит указатель на массив **unsigned int SerNumber[32]**, в который будет помещено значение серийного номера канала устройства.
Параметр **MaxValue** определяет количество слов, определяющих серийный номер устройства.

- **Установка свойств устройства аудио ввода канала**

VOID IDSetAudioProperty (
HDEVICE Hdv,
UINT AudioChannel,
MPAudioProperty Prop,
PROPERTYDATA chProp);

Данная функция производит установку свойств для каждого аудио канала отдельного устройства.

Параметры

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetAudioDevice**.

AudioChannel
Номер канала, для которого устанавливается свойства. Номер канала определяется только для данного устройства ввода и не соответствует номеру общего числа каналов при использовании нескольких устройств. При использовании одного устройства номера каналов устройства совпадают с общим номером канала системы.

Prop
Определяет свойство, которое необходимо установить. В следующей таблице приведен список параметров, которые могут быть установлены для каждого канала аудио ввода:

Свойство	Значение свойства
AUDIOCAPTURE_ENABLED	Разрешение/запрещение работы канала
AUDIOCAPTURE_MONO	Включение/выключение режима MOHO
AUDIOCAPTURE_RECORDLEVEL	Уровень записи
AUDIOCAPTURE_PAN	Баланс левого и правого каналов
AUDIOCAPTURE_LOUDNESS	Включение/выключение звука
AUDIOCAPTURE_TREBLE	Уровень высоких частот
AUDIOCAPTURE_BASS	Уровень низких частот
AUDIOCAPTURE_CAPTUREMODE	Режим ввода
AUDIOCAPTURE_NAME	Имя канала
AUDIOCAPTURE_SERIAL_NUMBER	Серийный номер канала

chProp
Структуру **PROPERTYDATA**, в которую размещаютсяут помещены свойства канала.

```
typedef struct TPropertyData {
        bool Present;              // Поддержка свойства
        int  Value;                // Текущее значение
        int  DefaultValue;         // Значение по умолчанию
        int  MinValue;             // Минимальное значение свойства
        int  MaxValue;             // Максимальное значение свойства
        int  SteppingDelta;        // Шаг изменения значения
        bool AutoEnabled;          // Автоизменение возможно
        bool Auto;                 // Значение автоизменения
        bool OnOffEnabled;         // Включение/выключение возможно
        bool OnOff;                // Значение включения/выключения
        bool OnePushEnabled;       // Импульсное включение возможно
        bool OnePush;              // Значение состояния импульса
        void *Data;                // Указатель на дополнительные данные
      } PROPERTYDATA;
```

Результат:

0 – функция не выполнена, свойство не поддерживается устройством,
1 – функция выполнена успешно,
2 – интерфейс не поддерживается данным устройством.

Для установки соответствующего свойства необходимо определить следующие параметры структуры: **Value.**

Примечание:

Параметр может быть установлен если значение параметра структуры **Present** имеет значение true и параметр **Auto** имеет значение false.

4.4. Функции обслуживания виртуальных устройств захвата аудио данных с использованием номера аудио канала

- **Получение свойств канала устройства аудио ввода**

VOID CHGetChannelAudioProperty(
UINT AudioChannel,
TAUDIOPROPERTY *CHAProp);

После инициализации устройства данная функция возвращает свойства отдельных каналов устройства ввода.

Параметры:

AudioChannel
Номер канала, для которого необходимо произвести чтение свойств. Номер канала соответствует номеру канала для всех устройств, активированных в системе: от 0 до MaxAudioChannels-1. Величина MaxAudioChannels получена по функции IDGetMaxAudioChannels.

CHAProp
Указатель на структуру, которая содержит свойства канала аудио ввода.

```
typedef struct TAudioProperty {
    WORD  wFormatTag;
    WORD  nChannels;
    DWORD nSamplesPerSec;
    DWORD nAvgBytesPerSec;
    WORD  nBlockAlign;
    WORD  wBitsPerSample;
    WORD  cbSize;
    } TAUDIOPROPERTY;
```

Структура полностью совпадает со структурой WAVEFORMATEX,
которая используетя в DirectX.

wFormatTag

Формат аудио инормации. Данная переменная всегда равна 1.

nChannels

Количество аудио каналов. Для всех режимов данная переменная равна 1. В случае использования устройств, которые вводят стерео данные, библиотека вводит отдельно левый и правый каналы как два канала в моно режиме.

nSamplesPerSec

Количество цифровых отсчетов в секунду (Герц).

nAvgBytesPerSec

Количество байт в секунду.

nBlockAlign

Выравнивание данных в блоке.

wBitsPerSample

Количество бит в отчете.

cbSize

Данная переменная не используется и должна быть равна 0.

- **Получить кадры изображения**

VOID CHGetAudioBuffer(UINT *AudioTegg, VOID **AudioBuffers);

Вызов данной функции должен производиться непрерывно и по возможности с максимальной интенсивностью. Рекомендуется использование потока с высоким приоритетом. Функция может не вызываться в случае отключения всех каналов.

Параметры:

AudioTegg
Параметр инициализируется пользователем и заполняется функцией. Указатель на массив **AudioTegg[MaxAudioChannels]**, где **MaxAudioChannels** величина полученная по функции IDGetMaxAudioChannels.

Определение	Величина	Значение
DATA_NOT_READY	0x00000000	Готовности блока нет
DATA_READY	0x00000001	Готовность блока есть

AudioBuffers
Указатель на массив указателей буферов аудио каналов **AudioBuffers[MaxAudioChannels]**, где **MaxAudioChannels** количество каналов поддерживаемых всеми устройствами инициализированных в системе. Размер каждого буфера должен составлять 4096 байт**.**

- **Включить/выключить ввод аудио данных по каналам**

VOID CHInputAudio(UINT *UsesAudioChannels);

Данная функция производит включение или выключение (разрешение и запрет) ввода по соответствующим каналам ввода аудио данных.

Параметры:

UsesAudioChannels
Указатель на массив **UsesAudioChannels[MaxAudioChannels]**, где **MaxAudioChannels** количество аудио каналов поддерживаемых системой. Величина MaxAudioChannels получена по функции IDGetMaxAudioChannels. Каждый элемент массива принимает следующие значения.
0 – канал выключен
1 – канал включен

- **Включить ввод по конкретному аудио каналу**

VOID CHStartAudio (UINT AudioChannel);

Данная функция производит включение ввода по конкретному аудио каналу системы.

Параметры:

AudioChannel
Номер канала, для которого необходимо произвести старт ввода. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

- **Выключить ввод по конкретному аудио каналу**

VOID CHStopAudio (UINT AudioChannel);

Данная функция производит выключение ввода по конкретному каналу системы.

Параметры:

AudioChannel
Номер канала, для которого необходимо произвести останов ввода. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

- **Получить идентификатор устройства по номеру канала**

HDEVICE CHGetAudioDevice(UINT AudioChannel);

Параметры:

AudioChannel
Номер канала, для которого необходимо определить идентификатор устройства. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

Результат:

Идентификатор устройства, возвращенный функцией **IDAdd** для конкретного канала.

- **Получение свойств устройства аудио ввода канала**

VOID CHGetAudioProperty (
HDEVICE Hdv,
UINT AudioChannel,
MPAudioProperty Prop,
PROPERTYDATA *chProp);

Данная функция производит считывание свойств для каждого аудио канала отдельного устройства.

Параметры

AudioChannel
Номер канала, для которого необходимо произвести чтение свойств. Номер канала соответствует номеру канала для всех устройств, активированных в системе: от 0 до MaxAudioChannels-1. Величина MaxAudioChannels получена по функции IDGetMaxAudioChannels.

Prop
Определяет свойство, которое необходимо считать. В следующей таблице приведен список параметров, которые могут быть прочитаны из устройства ввода аудио данных:

Свойство	Значение свойства
AUDIOCAPTURE_ENABLED	Разрешение/запрещение работы канала
AUDIOCAPTURE_MONO	Включение/выключение режима MOHO
AUDIOCAPTURE_RECORDLEVEL	Уровень записи
AUDIOCAPTURE_PAN	Баланс левого и правого каналов
AUDIOCAPTURE_LOUDNESS	Включение/выключение звука
AUDIOCAPTURE_TREBLE	Уровень высоких частот
AUDIOCAPTURE_BASS	Уровень низких частот
AUDIOCAPTURE_CAPTUREMODE	Режим ввода
AUDIOCAPTURE_NAME	Имя канала
AUDIOCAPTURE_SERIAL_NUMBER	Серийный номер канала

chProp
Указатель на структуру **PROPERTYDATA**, в которую будут помещены свойства канала.

```
typedef struct TPropertyData {
        bool Present;              // Поддержка свойства
        int  Value;                // Текущее значение
        int  DefaultValue;         // Значение по умолчанию
        int  MinValue;             // Минимальное значение свойства
        int  MaxValue;             // Максимальное значение свойства
        int  SteppingDelta;        // Шаг изменения значения
        bool AutoEnabled;          // Автоизменение возможно
        bool Auto;                 // Значение автоизменения
        bool OnOffEnabled;         // Включение/выключение возможно
        bool OnOff;                // Значение включения/выключения
        bool OnePushEnabled;       // Импульсное включение возможно
        bool OnePush;              // Значение состояния импульса
        void *Data;                // Указатель на дополнительные данные
      } PROPERTYDATA;
```

Результат:

0 – функция не выполнена, свойство не поддерживается устройством,
1 – функция выполнена успешно,
2 – интерфейс не поддерживается данным устройством.

Примечание:

Значение параметра структуры **PROPERTYDATA Present** определяет возможность использования свойства данным устройством. Если данный параметр имеет значение **false**, то свойство не используется.

При считывании свойства AUDIOCAPTURE_NAME параметр **Data** содержит указатель на строку **char Name[512]**, в которую будет помещено имя каналатва).

При считывании свойства AUDIOCAPTURE_CAPTUREMODE параметр **Data** содержит указатель на массив **unsigned int AudioMode[32][4]**, в который будет помещены возможные значения режимов ввода.
Параметр **MaxValue** определяет количество возможных значений режимов ввода.
AudioMode[n][0] – количество отчетов в секунду (Герц);
AudioMode[n][1] – количество каналов.
AudioMode[n][2] – выравнивание данных.
AudioMode[n][3] – количество бит на отчет.

При считывании свойства AUDIOCAPTURE_SERIAL_NUMBER параметр **Data** содержит указатель на массив **unsigned int SerNumber[32]**, в который будет помещено значение серийного номера устройства.
Параметр **MaxValue** определяет количество слов, определяющих серийный номер устройства.

- **Установка свойств устройства аудио ввода канала**

VOID CHSetAudioProperty (
UINT AudioChannel,
MPAudioProperty Prop,
PROPERTYDATA chProp);

Данная функция производит установку свойств для каждого аудио канала отдельного устройства.

Параметры

AudioChannel
Номер канала, для которого необходимо произвести чтение свойств. Номер канала соответствует номеру канала для всех устройств, активированных в системе: от 0 до MaxAudioChannels-1. Величина MaxAudioChannels получена по функции IDGetMaxAudioChannels.

Prop
Определяет свойство, которое необходимо установить. В следующей таблице приведен список параметров, которые могут быть установлены для каждого канала аудио ввода:

Свойство	Значение свойства
AUDIOCAPTURE_ENABLED	Разрешение/запрещение работы канала
AUDIOCAPTURE_MONO	Включение/выключение режима МОНО
AUDIOCAPTURE_RECORDLEVEL	Уровень записи
AUDIOCAPTURE_PAN	Баланс левого и правого каналов
AUDIOCAPTURE_LOUDNESS	Включение/выключение звука
AUDIOCAPTURE_TREBLE	Уровень высоких частот
AUDIOCAPTURE_BASS	Уровень низких частот
AUDIOCAPTURE_CAPTUREMODE	Режим ввода
AUDIOCAPTURE_NAME	Имя канала
AUDIOCAPTURE_SERIAL_NUMBER	Серийный номер канала

chProp
Структуру **PROPERTYDATA**, в которую размещаютсяут помещены свойства канала.

```
typedef struct TPropertyData {
    bool Present;          // Поддержка свойства
    int  Value;            // Текущее значение
    int  DefaultValue;     // Значение по умолчанию
    int  MinValue;         // Минимальное значение свойства
    int  MaxValue;         // Максимальное значение свойства
    int  SteppingDelta;    // Шаг изменения значения
```

```
    bool AutoEnabled;        // Автоизменение возможно
    bool Auto;               // Значение автоизменения
    bool OnOffEnabled;       // Включение/выключение возможно
    bool OnOff;              // Значение включения/выключения
    bool OnePushEnabled;     // Импульсное включение возможно
    bool OnePush;            // Значение состояния импульса
    void *Data;              // Указатель на дополнительные данные
  } PROPERTYDATA;
```

Результат:

0 – функция не выполнена, свойство не поддерживается устройством,
1 – функция выполнена успешно,
2 – интерфейс не поддерживается данным устройством.

Для установки соответствующего свойства необходимо определить следующие параметры структуры: **Value.**

Примечание:

Параметр может быть установлен если значение параметра структуры **Present** имеет значение true и параметр **Auto** имеет значение false.

- **Получить имени входной линии по номеру**

char * CHGetAudioName(UINT AudioChannel);

Параметры:

AudioChannel
Номер канала, для которого необходимо произвести чтение имени канала. Номер канала соответствует номеру канала для всех устройств, активированных в системе: от 0 до MaxAudioChannels-1. Величина MaxAudioChannels получена по функции IDGetMaxAudioChannels.

Результат:
Указатель на символьное имя аудио канала.

4.5. Функций обслуживания устройств ввода/вывода цифровых сигналов

- **Определение общего числа линий входных цифровых сигналов.**

UINT IDGetMaxInLines (void);

Результат
Общее число линий входных цифровых сигналов.

- **Определение общего числа линий выходных цифровых сигналов.**

UINT IDGetMaxOutLines (void);

Результат
Общее число линий выходных цифровых сигналов.

- **Ввод состояния линии цифрового сигнала.**

UINT IDGetInLine(HDEVICE Hdv,UINT Line);

Функция обеспечивает чтение состояния соответствующей линии ввода цифрового сигнала.

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Line
Определяет номер линии ввода устройства ввода в пределах **NumberInLines** структуры TDEVICEPROPERTY.

Результат

0- на линии состояние логического 0
1- на линии состояние логической 1.

- **Вывод сигнала на линию вывода цифровой информации.**

VOID IDSetOutLine(HDEVICE Hdv,UINT Line,UINT value);

Функция обеспечивает вывод сигнала на соответствующую линию вывода устройства ввода.

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Line
Определяет номер линии вывода в пределах **NumberInLines** структуры DEVICEPROPERTY.

Value
0 – запись логического 0
1 – запись логической 1

- **Получить идентификатор устройства по номеру выходной линии**

HDEVICE CHGetOutLineDevice(UINT LineNumber);

Параметры:

LineNumber
Номер выходной линии от нуля до значения определяемого по функции IDGetMaxOutLines-1.

Результат:

Идентификатор устройства, возвращенный функцией **IDAdd** для конкретного устройства.

- **Получить имени выходной линии по номеру**

char * CHGetOutLineName(UINT LineNumber);

Параметры:

LineNumber
Номер выходной линии от нуля до значения определяемого по функции IDGetMaxOutLines-1.

Результат:
Указатель на символьное имя выходной линии.

- **Получить идентификатор устройства по номеру входной линии**

HDEVICE CHGetInLineDevice(UINT LineNumber);

Параметры:

LineNumber
Номер входной линии от нуля до значения определяемого по функции IDGetMaxInLines-1.

Результат:

Идентификатор устройства, возвращенный функцией **IDAdd** для конкретного устройства.

- **Получить имени входной линии по номеру**

char * CHGetInLineName(UINT LineNumber);

Параметры:

LineNumber
Номер выходной линии от нуля до значения определяемого по функции IDGetMaxInLines-1.

Результат:
Указатель на символьное имя входной линии.

- **Ввод состояния линии цифрового сигнала**

UINT CHGetInLine(UINT Line);

Функция обеспечивает чтение состояния соответствующей линии ввода цифрового сигнала.

Параметры:

Line
Определяет номер линии ввода в пределах от 0 до **MaxNumberInLines-1.** Число **MaxNumberInLines** определяется как сумма всех значений NumberInLines структуры TDEVICEPROPERTY для всех устройств ввода активированных в системе или по функции **IDGetMaxInLines**.

Результат

0 - на линии состояние логического 0
1 - на линии состояние логической 1.

- **Вывод сигнала на линию вывода цифровой информации**

VOID CHSetOutLine(UINT Line, UINT value);

Функция обеспечивает вывод сигнала на соответствующую линию вывода устройства ввода.

Параметры:

Line
Определяет номер линии вывода устройства ввода в пределах от 0 до **MaxNumberOutLines-1.** Число **MaxNumberOutLines** определяется как сумма всех значений NumberOutLines структуры TDEVICEPROPERTY для всех устройств ввода активированных в системе или по функции **IDGetMaxInLines**.

Value
0 – запись логического 0
1 – запись логической 1

5. Функции преобразования форматов изображений

- **Функции преобразования формата YUY2 в цветное RGB изображений.**

VOID YUY2toRGB(PVOID inb, PVOID ob, UINT SizeX, UINT SizeY);

Вспомогательная функция формирования цветных RGB изображений для вывода на экран монитора. Данная функция формирует цветное RGB изображение из исходного цветоразностного изображения.

Параметры:

inb
Входной буфер изображения **Images** полученный функцией IDGetPicture или CHGetPicture.

0:0 (Y – U – Y – V – Y – U – Y …..) Входное цветное изображение SizeX*2 SizeY байт.

ob
Выходной буфер для обработки размером SizeX*3*SizeY. Буфер содержит цветное изображение в формате RGB.

0:0 (R – G – B – R – G - B …..) Цветное изображение SizeX*3 на SizeY байт.

SizeX
Размер по горизонтали, который соответствует размеру полутонового изображения.

SizeY
Размер по вертикали в пикселях.

При использовании обработки с двумя полукадрами размер по вертикали следует увеличить соответственно в 2 раза.

- **Функции преобразования формата Y411 в цветное RGB изображений.**

VOID Y411toRGB(PVOID inb, PVOID ob, UINT SizeX, UINT SizeY);

Вспомогательная функция формирования цветных RGB изображений для вывода на экран монитора. Данная функция формирует цветное RGB изображение из исходного цветоразностного изображения.

Параметры:

inb
Входной буфер изображения **Images** полученный функцией IDGetPicture или CHGetPicture.

0:0 (U – Y– Y – V – Y – Y …..) Входное цветное изображение SizeX /8*12* SizeY байт.

ob
Выходной буфер для обработки размером SizeX*3*SizeY. Буфер содержит цветное изображение в формате RGB.

0:0 (R – G – B – R – G - B …..) Цветное изображение SizeX*3 на SizeY байт.

SizeX
Размер по горизонтали, который соответствует размеру полутонового изображения.

SizeY
Размер по вертикали в пикселях.

При использовании обработки с двумя полукадрами размер по вертикали следует увеличить соответственно в 2 раза.

- **Функции преобразования формата UYVY в цветное RGB изображений.**

VOID UYVYtoRGB(PVOID inb, PVOID ob, UINT SizeX, UINT SizeY);

Вспомогательная функция формирования цветных RGB изображений для вывода на экран монитора. Данная функция формирует цветное RGB изображение из исходного цветоразностного изображения.

Параметры:

inb
Входной буфер изображения **Images** полученный функцией IDGetPicture или CHGetPicture.

0:0 (U – Y – V – Y – U – Y – V …..) Входное цветное изображение SizeX*2 SizeY байт.

ob
Выходной буфер для обработки размером SizeX*3*SizeY. Буфер содержит цветное изображение в формате RGB.

0:0 (R – G – B – R – G - B …..) Цветное изображение SizeX*3 на SizeY байт.

SizeX
Размер по горизонтали, который соответствует размеру полутонового изображения.

SizeY
Размер по вертикали в пикселях.

При использовании обработки с двумя полукадрами размер по вертикали следует увеличить соответственно в 2 раза.

- **Функция преобразования формата UYVY в яркостное изображение**

VOID UYVYtoGRAY(PVOID inb, PVOID ob, UINT SizeX, UINT SizeY);

Вспомогательная функция формирования яркостного изображения из цветоразностного изображения.

Параметры:

inb
Входной буфер изображения **Images** полученный функцией IDGetPicture или CHGetPicture.

0:0 (U – Y – V – Y – U – Y – V …..) Цветоразностное изображение SizeX*2 на SizeY

ob
Выходной буфер для обработки размером SizeX*SizeY. Буфер содержит полутоновое изображение.

0:0 (Y – Y – Y …..) Полутоновое изображение SizeX на SizeY

SizeX
Размер по горизонтали, который соответствует размеру полутонового изображения.

SizeY
Размер по вертикали в пикселях.

При использовании обработки с двумя полукадрами размер по вертикали следует увеличить соответственно в 2 раза.

- **Функция преобразования формата Y411 в яркостное изображение**

VOID Y411toGRAY(PVOID inb, PVOID ob, UINT SizeX, UINT SizeY);

Вспомогательная функция формирования яркостного изображения из цветоразностного изображения.

Параметры:

inb
Входной буфер изображения **Images** полученный функцией IDGetPicture или CHGetPicture.

0:0 (U – Y– Y – V – Y– Y …..) Цветоразностное изображение SizeX*2 на SizeY

ob
Выходной буфер для обработки размером SizeX*SizeY. Буфер содержит полутоновое изображение.

0:0 (Y – Y – Y …..) Полутоновое изображение SizeX на SizeY

SizeX
Размер по горизонтали, который соответствует размеру полутонового изображения.

SizeY
Размер по вертикали в пикселях.

При использовании обработки с двумя полукадрами размер по вертикали следует увеличить соответственно в 2 раза.

- **Функция преобразования формата YUVY в яркостное изображение**

VOID YUY2toGRAY(PVOID inb, PVOID ob, UINT SizeX, UINT SizeY);

Вспомогательная функция формирования яркостного изображения из цветоразностного изображения.

Параметры:

inb
Входной буфер изображения **Images** полученный функцией IDGetPicture или CHGetPicture.

0:0 (Y – U – Y – V – Y – U – Y - V …..) Цветоразностное изображение SizeX*2 на SizeY

ob
Выходной буфер для обработки размером SizeX*SizeY. Буфер содержит полутоновое изображение.

0:0 (Y – Y – Y …..) Полутоновое изображение SizeX на SizeY

SizeX
Размер по горизонтали, который соответствует размеру полутонового изображения.

SizeY
Размер по вертикали в пикселях.

При использовании обработки с двумя полукадрами размер по вертикали следует увеличить соответственно в 2 раза.

- **Функция преобразования формата RGB в яркостное изображение**

VOID RGBtoGRAY(PVOID inb, PVOID ob, UINT SizeX, UINT SizeY);

Вспомогательная функция формирования яркостного изображения из цветоразностного изображения.

Параметры:

inb
Входной буфер изображения **Images** полученный функцией IDGetPicture или CHGetPicture.

0:0 (R – G – B – R – G - B …..) Цветное изображение SizeX*3 на SizeY байт.

ob
Выходной буфер для обработки размером SizeX*SizeY. Буфер содержит полутоновое изображение.

0:0 (Y – Y – Y …..) Полутоновое изображение SizeX на SizeY

SizeX
Размер по горизонтали, который соответствует размеру полутонового изображения.

SizeY
Размер по вертикали в пикселях.

При использовании обработки с двумя полукадрами размер по вертикали следует увеличить соответственно в 2 раза.

- **Функция преобразования формата яркостное изображение в UYVY**

VOID GRAYtoUYVY (PVOID inb, PVOID ob, UINT SizeX, UINT SizeY);

Вспомогательная функция формирования изображений для вывода на экран. Данная функция формирует необходимое цветоразностное изображение из исходного полутонового изображения.

Параметры:

inb
Входной буфер изображения **Images** полученный функцией IDGetPicture или CHGetPicture.

0:0 (Y – Y – Y …..) Первый полукадр цветного изображения SizeX на SizeY

ob
Выходной буфер для вывода размером SizeX*2 на SizeY. Буфер содержит цветоразностное изображение с двойным разрешением по горизонтали, которое необходимо для работы процедур вывода изображений.

0:0 (U – Y – V – Y – U – Y – V …..) Первый полукадр цветного изображения SizeX*2 на SizeY

SizeX
Размер по горизонтали, который соответствует размеру полутонового изображения.

SizeY
Размер по вертикали в пикселях.

При использовании обработки с двумя полукадрами размер по вертикали следует увеличить соответственно в 2 раза.

- **Функция преобразования формата Y411 в UYVY**

VOID Y411toUYVY (PVOID inb, PVOID ob, UINT SizeX, UINT SizeY);

Вспомогательная функция формирования изображений для вывода на экран. Данная функция формирует необходимое цветоразностное изображение из исходного полутонового изображения.

Параметры:

inb
Входной буфер изображения **Images** полученный функцией IDGetPicture или CHGetPicture.

0:0 (U - Y – Y – V – Y - Y …..) Первый полукадр цветного изображения SizeX*8/12 на SizeY

ob
Выходной буфер для вывода размером SizeX*2 на SizeY. Буфер содержит цветоразностное изображение с двойным разрешением по горизонтали, которое необходимо для работы процедур вывода изображений.

0:0 (U – Y – V – Y – U – Y – V …..) Первый полукадр цветного изображения SizeX*2 на SizeY

SizeX
Размер по горизонтали, который соответствует размеру полутонового изображения.

SizeY
Размер по вертикали в пикселях.

При использовании обработки с двумя полукадрами размер по вертикали следует увеличить соответственно в 2 раза.

- **Функция преобразования формата яркостное изображение в YUY2**

VOID GRAYtoYUY2 (PVOID inb, PVOID ob, UINT SizeX, UINT SizeY);

Вспомогательная функция формирования изображений для вывода на экран. Данная функция формирует необходимое цветоразностное изображение из исходного полутонового изображения.

Параметры:

inb
Входной буфер изображения **Images** полученный функцией IDGetPicture или CHGetPicture.

0:0 (Y – Y – Y …..) Первый полукадр цветного изображения SizeX на SizeY

ob

Выходной буфер для вывода размером SizeX*2 на SizeY. Буфер содержит цветоразностное изображение с двойным разрешением по горизонтали, которое необходимо для работы процедур вывода изображений.

0:0 (Y –U – Y – V – Y – U – Y - V …..) Первый полукадр цветного изображения SizeX*2 на SizeY

SizeX
Размер по горизонтали, который соответствует размеру полутонового изображения.

SizeY
Размер по вертикали в пикселях.

При использовании обработки с двумя полукадрами размер по вертикали следует увеличить соответственно в 2 раза.

- **Функция преобразования формата Y411 в YUY2**

VOID Y411toYUY2 (PVOID inb, PVOID ob, UINT SizeX, UINT SizeY);

Вспомогательная функция формирования изображений для вывода на экран. Данная функция формирует необходимое цветоразностное изображение из исходного полутонового изображения.

Параметры:

inb
Входной буфер изображения **Images** полученный функцией IDGetPicture или CHGetPicture.

0:0 (U - Y – Y – V – Y - Y …..) Первый полукадр цветного изображения SizeX/8*12 на SizeY

ob
Выходной буфер для вывода размером SizeX*2 на SizeY. Буфер содержит цветоразностное изображение с двойным разрешением по горизонтали, которое необходимо для работы процедур вывода изображений.

0:0 (Y –U – Y – V – Y – U – Y - V …..) Первый полукадр цветного изображения SizeX*2 на SizeY

SizeX
Размер по горизонтали, который соответствует размеру полутонового изображения.

SizeY
Размер по вертикали в пикселях.

При использовании обработки с двумя полукадрами размер по вертикали следует увеличить соответственно в 2 раза.

- **Функция редуцирования яркостного изображения по горизонтали**

VOID GRAYHREDUCE (PVOID inb, PVOID ob, UINT SizeX, UINT SizeY);

Вспомогательная функция, которая позволяет редуцировать в два раза по горизонтали полутоновое изображение. Данная функция предназначена для формирования изображения с квадратными пикселями из изображения с широкими пикселями.

Параметры:

inb
Входной буфер изображения **Images** полученный функцией IDGetPicture, CHGetPicture или UYVYtoGRAY.

ob
Выходной буфер для обработки размером SizeX/2*SizeY.

SizeX
Размер изображения в пикселях по горизонтали.

SizeY
Размер изображения в пикселях по вертикали.

- **Функция редуцирования цветного RGB изображения по горизонтали**

VOID RGBHREDUCE (PVOID inb, PVOID ob, UINT SizeX, UINT SizeY);

Вспомогательная функция, которая позволяет редуцировать в два раза по горизонтали цветное RGB изображение. Данная функция предназначена для формирования изображения с квадратными пикселями из изображения с широкими пикселями.

Параметры:

inb
Входной буфер, в который помещается полутоновое изображение размером SizeX*3 * SizeY.

ob
Выходной буфер, в который будет помещено полутоновое изображение размером SizeX*3/2 * SizeY.

SizeX

Размер изображения в пикселях по горизонтали.

SizeY
Размер изображения в пикселях по вертикали.

- **Функция редуцирования цветоразностного изображения по горизонтали**

VOID UYVYHREDUCE (PVOID inb, PVOID ob, UINT SizeX, UINT SizeY);

Вспомогательная функция, которая позволяет редуцировать в два раза по горизонтали цветоразностное UYVY изображение. Данная функция предназначена для формирования изображения с квадратными пикселями из изображения с широкими пикселями.

Параметры:

inb
Входной буфер, в который помещается полутоновое изображение размером SizeX*2 * SizeY.

ob
Выходной буфер, в который будет помещено полутоновое изображение размером SizeX*2/2 * SizeY.

SizeX
Размер изображения в пикселях по горизонтали.

SizeY
Размер изображения в пикселях по вертикали.

- **Функция редуцирования цветоразностного изображения по горизонтали**

VOID YUY2HREDUCE (PVOID inb, PVOID ob, UINT SizeX, UINT SizeY);

Вспомогательная функция, которая позволяет редуцировать в два раза по горизонтали цветоразностное YUYV (YUY2) изображение. Данная функция предназначена для формирования изображения с квадратными пикселями из изображения с широкими пикселями.

Параметры:

inb
Входной буфер, в который помещается полутоновое изображение размером SizeX*2 * SizeY.

ob
Выходной буфер, в который будет помещено полутоновое изображение размером SizeX*2/2 * SizeY.

SizeX
Размер изображения в пикселях по горизонтали.

SizeY
Размер изображения в пикселях по вертикали.

- **Функция преобразования размеров изображений по горизонтали**

VOID IMAGERESIZE(UINT SSizeX,UINT SSizeY,PVOID inb,UINT DSizeX,UINT DSizeY,PVOID obuf,UINT colorcode);

Вспомогательная функция, которая позволяет преобразовать размеры вводимого изображения с 704 пикселей до 768 пикселей для формирования квадратных пикселей и устранения искажений изображения для распознавания номеров.

SSizeX
Размер исходного изображения в пикселях по горизонтали (704).

SSizeY
Размер исходного изображения в пикселях по вертикали.

inb
Буфер входного изображения.

DSizeX
Размер выходного изображения в пикселях по горизонтали (768 или 384).

DSizeY
Размер выходного изображения в пикселях по вертикали (должен быть равен SSizeX).

obuf
Буфер выходного изображения.

colorcode
Цветовое разрешение изображения. Может принимать одно из следующих значений:
MAKEFOURCC('G','R','A','Y');
MAKEFOURCC('U','Y','V','Y');
MAKEFOURCC('Y','U','Y','2');

Пример:
IMAGERESIZE(704,288,inbuffer,768,288,outbuffer, MAKEFOURCC('U','Y','V','Y'));

Преобразует изображение 704*288 в 768*288

IMAGERESIZE(704,288,inbuffer,384,288,outbuffer, MAKEFOURCC('U','Y','V','Y'));
Преобразует изображение 704*288 в 384*288

6. Функции определения автомобильных номеров

Компания «Мегапиксел», первая в СНГ и одна из первых в мире, приступила в 1995 году и впоследствии успешно завершила работы по созданию передовой технологии - автоматическое распознавание автомобильных номеров. Изначально, задача считывания российских номеров оказалась гораздо сложнее, чем в других странах. Это объясняется обилием типов номеров, использованием символов различных размеров, а также более сложными природными условиями, что выражается в слабой контрастности обрабатываемых изображений.

Данная группа функций библиотеки предназначены для создания приложений по автоматическому распознаванию автомобильных номеров, как в условиях скоростных магистралей, так и в стояночных приложениях. Высокая устойчивость алгоритмов к естественным и искусственным источникам помех позволяет успешно использовать конечные приложения в условиях российского зимнего периода. Библиотека реализована с использованием нейроподобных алгоритмов цифровой обработки изображений.

Распознаются все типы российских номеров (с 3-х-значным кодом региона включительно), а так же номерные пластины многих стран мира. Возможна адаптация к новым типам номерных знаков.

Обеспечивается распознавание номеров различных государств:

- номерные знаки России;
- номерные знаки стран СНГ;
- номерные знаки Германии;
- номерные знаки Великобритании;
- номерные знаки Испании;
- номерные знаки Италии;
- номерные знаки Тайваня;
- номерные знаки Сингапура.
- номерные знаки Голландии.
- номерные знаки Бразилии.
- номерные знаки Греции.

Имеется возможность дополнения номеров других типов и государств.

Функции библиотеки способны обрабатывать изображения произвольного размера следующих типов:

- полутоновое изображение (1 байт на пиксель);
- цветоразностное изображение (4 байта на 2 пикселя в формате U-Y-V-Y);
- цветное изображение (3 байта на пиксель в формате RGB).

По типу изображения могут быть:

- с квадратными пикселями;
- с широкими пикселями (разрешение по горизонтали в два раза шире).

Поиск номера осуществляется во всем диапозоне от 6 до 50 пикселей по вертикали.
В ходе обработки функции формируют следующую информацию:

- информацию о распознанных номерах, формирует наиболее качественное результирующее изображение зоны номера, формирует наилучшее изображение транспортного средства;
- информацию о коррекции номера, формирует бинарное и полутоновое изображение текущей зоны номера, передает текущие параметры номера;
- информацию о всех найденных зонах, формирует бинарное и полутоновое изображение каждой зоны;
- информацию о присутствии транспортного средства;
- вычисление скорости движения автомобиля по двум соседним полукадрам с точностью до 2%.

Для проверки возможности использования функций определения автомобильных номеров, необходимо выполнить функцию **IDGetDeviceProperty** и проверить бит DEVCFENABLED в параметре **DeviceFlags**.

Как и с функциями виртуальных устройств существуют два уровня обработки: обслуживание с использованием идентификатора устройства (в данном случае работа осуществляется непосредственно с каждым отдельно взятым устройством) и на уровне отдельных каналов (библиотека сама определяет требуемое устройство, которое соответствует данному каналу).

6.1. Функции определения автомобильных номеров с использованием идентификатора устройства

При вызове данной группы функций в качестве параметров определяющих канал обработки используется идентификатор устройства и номер канала в пределах данного устройства.

- **Прочитать версию библиотеки определения автомобильных номеров**

UINT CFGetVersion(VOID);

Возвращает номер текущей версии функций определения автомобильных номеров. 0x6XX
0x600 – Версия библиотеки MegaLib V 1.0
0x601 – Версия библиотеки MegaLib V 1.1, MegaLib V 1.1.

- **Инициализация процедур определителя номерного знака**

UINT CFOpen (UINT Country, char *LibPath);

Параметры:

Country
Код страны, номерные знаки которой должны быть распознаны:

номерные знаки России	0
номерные знаки стран СНГ	1
номерные знаки Германии	2
номерные знаки Великобритании	3
номерные знаки Испании	4
номерные знаки Италии	5
номерные знаки Тайваня	6
номерные знаки Сингапура	7
номерные знаки Голландии	8
номерные знаки Бразилии	9
номерные знаки Греции	10

Кроме перечисленных выше стран можно формировать собственные описания номеров для их определения в других странах. Описания и номера стран располагаются в специальном файле **COUNTRY.CFG**. Описание данного файла приводится в соответствующем документе («Формат файла COUNTRY.CFG»).

LibPath
Указатель на путь к библиотеке. Полный путь к каталогу, в котором располагается файл **MEGALIB1.dll**

Функция возвращает код необходимый для завершения работы процедур распознавания. Код 0 – процедура выполнена успешно.

- **Окончание работы процедуры определения номерного знака**

VOID CFClose (UINT ErrorCode);

Параметры:

ErrorCode
Параметр, возвращаемый функцией **CFOpen.**

```
// Исполнительный файл находится в каталоге c:\megalib1

UINT Code = CFOpen(0, "C:\\megalib1");

//Использование функций

CFClose ( Code );
```

- **Установка параметров обрабатываемого кадра**

VOID CFSetChannelProperty (
HDEVICE Hdv,
UINT Channel,
UINT SizeX,
UINT SizeY,
UINT PixelMode);

Параметры устанавливаемого кадра должны соответствовать параметрам вводимого изображения для данного канала. Размеры и параметр ширины пикселей соответствует параметрам черно-белого изображения, которое будет подано для обработки.

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Channel
Номер канала, по которому производится установка параметров. Номер канала определяется только для данного устройства ввода и не соответствует номеру общего числа каналов при использовании нескольких устройств. При использовании одного устройства номера каналов устройства совпадают с общим номером канала системы.

SizeX
Размер обрабатываемого кадра по горизонтали.

SizeY

Размер обрабатываемого кадра по вертикали.

PixelMode
Описание пикселей в обрабатываемом кадре. Параметр может содержать комбинацию следующих флагов:

Название флага	Описание
SQUARE_PIXELS	квадратные пиксели.
WIDE_PIXELS	широкие пиксели, изображение по горизонтали имеет двойное разрешение.
GRAY_PIXELS	каждый пиксель 8 бит (серое изображение).
RGB_PIXELS	каждый пиксель 24 бита (цветное RGB изображение).
YUY2_PIXELS	каждый пиксель 16 бит (цветоразностное изображение Y-U-Y-V)
UYVY_PIXELS	каждый пиксель 16 бит (цветоразностное изображение U-Y-V-Y)

Если параметр PixelMode содержит флаг SQUARE_PIXELS, то изображение имеет квадратные пиксели, как изображено на рисунке.

```
//Размер изображения 384*288

CFSetChannelProperty(Hdv,channel,384,288,SQUARE_PIXELS|GRAY_PIXELS);
```

Если параметр PixelMode содержит флаг WIDE_PIXELS, то изображение имеет широкие пиксели (изображение в ширину имеет двойной размер), как изображено на рисунке.

.

```
//Размер изображения 768*288

CFSetChannelProperty(Hdv,channel,768,288,WIDE_PIXELS|GRAY_PIXELS);
```

- **Установка параметров обрабатываемого кадра для всех каналов**

VOID CFSetProperty (
HDEVICE Hdv,
UINT SizeX,
UINT SizeY,
UINT PixelMode);

Параметры устанавливаемого кадра должны соответствовать параметрам вводимого изображения для данного канала. Размеры и параметр ширины пикселей соответствует параметрам черно-белого изображения, которое будет подано для обработки. Данная функция осуществляет установку параметров для всех активированных каналов.

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

SizeX
Размер обрабатываемого кадра по горизонтали.

SizeY
Размер обрабатываемого кадра по вертикали.

PixelMode
Описание пикселей в обрабатываемом кадре. Параметр может содержать комбинацию следующих флагов:

Название флага	Описание
SQUARE_PIXELS	квадратные пиксели.
WIDE_PIXELS	широкие пиксели, изображение по горизонтали имеет двойное разрешение.
GRAY_PIXELS	каждый пиксель 8 бит (серое изображение).
RGB_PIXELS	каждый пиксель 24 бита (цветное RGB изображение).
UYVY_PIXELS	каждый пиксель 16 бит (цветоразностное изображение U-Y-V-Y)
YUY2_PIXELS	каждый пиксель 16 бит (цветоразностное изображение Y-U-Y-V)

```
//Размер изображения 384*288 (с квадратными пикселями)

CFSetProperty(Hdv,384,288,SQUARE_PIXELS|GRAY_PIXELS);

//Размер изображения 768*288 (то же изображение с широкими пикселями)

CFSetProperty(Hdv,768,288,WIDE_PIXELS|GRAY_PIXELS);
```

- **Установка исходных параметров канала**

VOID CFRestart (HDEVICE Hdv,UINT Channel, NUMINITPARAMS *ChParams);

Функция устанавливает исходные значения параметров канала для определения номерного знака.

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Channel
Номер устанавливаемого канала. Номер канала определяется только для данного устройства ввода и не соответствует номеру общего числа каналов при использовании нескольких устройств. При использовании одного устройства номера каналов устройства совпадают с общим номером канала системы.

ChParams
Указатель на структуру параметров канала.

```
typedef struct TNumInitParam {
        UINT CarThreshold;
        UINT ZoneThreshold;
        UINT ZoneMode;
        RECT Zone_visio;
        UINT Thhory;
        UINT Thhorx;
        UINT Thbin;
        UINT Min_Pipe_Size;
        UINT Max_Pipe_Size;
      } NUMINITPARAMS;
```

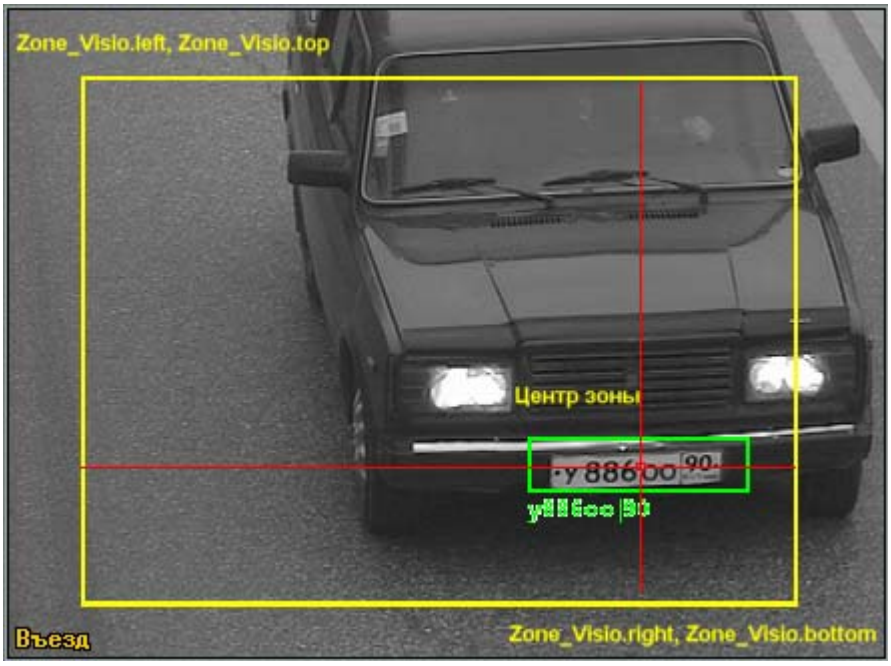
CarThreshold=168
Чувствительность системы к появлению центра автомобиля в поле зрения. Данная величина изменяется в пределах от 100 до 255 и подбирается опытным путем при установке системы.

ZoneThreshold=168 (в версии 0x601 данный параметр не используется)
Чувствительность системы к поиску зоны номерного знака. Данная величина изменяется в пределах от 100 до 255 и подбирается опытным путем при установке системы.
ZoneMode
0 – обычный режим 96*24 используется для совместимости с ранними версиями;
Сочитание следующих флагов позволяет производить многозонный поиск номеров:
ZONE_MODE_96x24 – разрешение обработки 96*24 , что соответствует высоте символа номера 12 строк;
ZONE_MODE_128x32 – разрешение обработки 128*32 , что соответствует высоте символа номера 16 строк;
ZONE_MODE_160x40 – разрешение обработки 160*40 , что соответствует высоте символа номера 20 строк **(присутствует в версии 0x601)**;
ZONE_MODE_192x48 – разрешение обработки 192*48 , что соответствует высоте символа номера 24 строк;
ZONE_MODE_256x64 – разрешение обработки 256*64 , что соответствует высоте символа номера 32 строк **(присутствует в версии 0x601)**;
ZONE_MODE_320x80 – разрешение обработки 320*80 , что соответствует высоте символа номера 40 строк **(присутствует в версии 0x601)**;
ZONE_MODE_384x96 – разрешение обработки 384*96 , что соответствует высоте символа номера 48 строк **(присутствует в версии 0x601)**;

ZONE_MODE_48x48 – разрешение обработки 48*48 , что соответствует высоте символа номера 12 строк для двухстрочных номеров **(присутствует в версии 0x601)**;
ZONE_MODE_64x64 – разрешение обработки 64*64 , что соответствует высоте символа номера 16 строк для двухстрочных номеров **(присутствует в версии 0x601)**;
ZONE_MODE_80x80 – разрешение обработки 80*80 , что соответствует высоте символа номера 20 строк для двухстрочных номеров **(присутствует в версии 0x601)**;
ZONE_MODE_96x96 – разрешение обработки 96*96 , что соответствует высоте символа номера 24 строк для двухстрочных номеров **(присутствует в версии 0x601)**;
ZONE_MODE_128x128 – разрешение обработки 128*128 , что соответствует высоте символа номера 32 строк для двухстрочных номеров **(присутствует в версии 0x601)**;
ZONE_MODE_160x160 – разрешение обработки 160*160 , что соответствует высоте символа номера 40 строк для двухстрочных номеров **(присутствует в версии 0x601)**;
ZONE_MODE_192x192 – разрешение обработки 192*192 , что соответствует высоте символа номера 48 строк для двухстрочных номеров **(присутствует в версии 0x601)**.

Zone_visio
Определяет границы изображения, в которых производится поиск центра зоны номерного знака. Размер по горизонтали соответствует размеру квадратных пикселей. Например, если реальный размер изображения по горизонтали равен

768 и флаг WIDE_PIXELS установлен, то границы поиска зоны определяются в пределах от 0 до 383.



Thhory (по умолчанию 208)
Параметр определяет степень очистки бинарного изображения сверху и снизу от символов номерного знака. **(в версии 0x601 данный параметр не используется)**

Thhorx (по умолчанию 202)
Параметр определяет степень очистки бинарного изображения слева и справа от символов номерного знака. **(в версии 0x601 данный параметр не используется)**

Thbin=136
Параметр определяет степень очистки бинарного изображения от помех. **(в версии 0x601 данный параметр не используется)**

Min Pipe Size=0
Минимально допустимая ширина номера в процентах по отношению к ширине зоны номерного знака. **(в версии 0x601 данный параметр не используется)**
Max Pipe Size=100
Максимально допустимая ширина номера в процентах по отношению к ширине зоны номерного знака. **(в версии 0x601 данный параметр не используется)**

```
void StartAllChannel()
{
UINT MaxChannels;
UINT *UsesChannels;
TCHANNELPROPERTY prop;
```

```
MaxChannels = IDGetMaxChannels();
UsesChannels = new UINT [MaxChannels];
for(int i=0;i<MaxChannels;i++) {
     UsesChannels[i]=3;
     CHSetParams(i,&INPUTPARAMS);
     CFRestart(CHGetDevice(i),
                  i-IDGetFirstChannel(CHGetDevices(i)),
                      & NUMINITPARAMS);
     CHGetProperty(i,&prop);
     CFSetChannelProperty(CHGetDevice(i),
                           i-IDGetFirstChannel(CHGetDevices(i)),
                              prop.HorFrameSize,
                                 prop.VerFrameSize,
                                    prop.PixelMode&SQUARE_PIXELS);
     }

CHInput(UsesChannels);
CFError = CFOpen(0,"c:\\ProgramFiles\\MegaLib");

delete UsesChannels;
}
//*****************************************************************
void StopAllChannel()
{
UINT MaxChannels;
UINT *UsesChannels;

MaxChannels = IDGetMaxChannels();
UsesChannels = new UINT [MaxChannels];
for(int i=0;i<MaxChannels;i++) UsesChannels[i]=0;

CFClose(CFError);
CHInput(UsesChannels);

delete UsesChannels;
}
```

- **Установка параметров распознавания**

VOID CFSetRecognitionParams(HDEVICE Hdv,UINT Channel, NUMINITRECPARAMS *Params);

Функция устанавливает исходные параметры распознавания для определения номерного знака.

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Channel
Номер устанавливаемого канала. Номер канала определяется только для данного устройства ввода и не соответствует номеру общего числа каналов при

использовании нескольких устройств. При использовании одного устройства номера каналов устройства совпадают с общим номером канала системы.

Params
Указатель на структуру параметров распознавания.

```
typedef struct TNumInitRecognitionParam {
        UINT SymbolRecFlag;
        UINT ZoneRecFlag;
        UINT ErrorCorrectionTime;
        UINT WeightThreshold;
        UINT Reserved1;
        UINT Reserved2;
        UINT Reserved3;
      } NUMINITRECPARAMS;
```

SymbolRecFlag
Флаги, отвечающие за посимвольную структуризацию номерного знака.

SYM_ENABLED_CLEARNUMBER (0x00000001)
Флаг разрешает удаление объектов из номера, которые не соответствуют признакам алфавитно-цифрового символа. Таким образом, происходит очистка номера от загрязнений при бинаризации. В случае низкого качества бинарного изображения, при котором символы могут соединяться друг с другом данный флаг следует выключить.

SYM_ENABLED_CORRECTION (0x00000002)
Флаг разрешает дополнительную корректировку символов после проведения распознавания в случае отсутствия гипотезы. Осуществляется проведение анализа последовательности отдельно определенных символов и построения вероятностной окончательной гипотезы.

SYM_ENABLED_DELETEZONES (0x00000004)
Флаг разрешает исключение зон, в которых невозможно определить отдельные буквенно-цифровые символы. Данные зоны считаются не номерными и исключаются из обработки.

ZoneRecFlag
Флаги, влияющие на исключения номеров, не отвечающих определенным критериям.

ZONE_ENABLED_CHECK_SIZE (0x00000001)
При работе системы в поле зрения камеры попадают надписи или структуры, которые не являются номерами. Для отсечения подобных надписей в программе устанавливаются предельные значения высоты символов, которые считаются недопустимыми. Для высоты символа в 10 пикселей границы составляют от 6 до 15, для символов в 15 пикселей – от 10 до 20 и для символов в 20 пикселей – от 15 до 25 и т.д.. Если данный флаг установлен, номера не удовлетворяющие критерию высоты символа исключаются из обработки.

ZONE_ENABLED_CHECK_WEIGHT (0x00000002)
В процессе определения номера формируется параметр вероятности распознавания. Данный параметр может принимать значение от 0 до 100. . Если данный флаг установлен, данные с вероятностью меньше, либо равно величине **WeightTreshold** блокируются системой и не передаются на выход.

ZONE_ENABLED_CHECK_GIPOTIZE (0x00000004)
В процессе определения номера формируется гипотеза, которая соответствует типу номерного знака. При невозможности определения типа формируется нулевая гипотеза. Если флаг установлен, такие номера не передаются на выход.
Установка данного флаг позволяет исключить появления «ложных» номеров, но и не позволяет осуществить попытку построения вероятностной гипотезы.

ZONE_ENABLED_CHECK_LOST (0x00000008)
В процессе определения номера возможно пропадание зоны номера на короткий промежуток времени, что вызывает выработку признака окончания распознавания с последующим возобновлением прослеживания номера. Данный флаг предотвращает повторное определение одного и того же номера. Распознанный номер хранится в списке распознанных номеров около 2.5 секунд и блокирует повторное распознавание.

ErrorCorrectionTime
Данный параметр определяет количество коррекций номерного знака при удачно определенной зоне и неудачно распознанного номера. Число определяет количество неудачных циклов распознавания, после которых срабатывает признак окончания распознавания.

WeightTreshold
Минимальное значение вероятности, при которой номер рассматривается как распознанный. Действует при установленном флаге **ZONE_ENABLED_CHECK_WEIGHT.**

- **Выполнение процедуры определения номерного знака**

UINT CFRun (HDEVICE Hdv, UINT Channel, void *Imgbuf, UINT Flag);

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Channel
Номер устанавливаемого канала обработки. Номер канала определяется только для данного устройства ввода и не соответствует номеру общего числа каналов

при использовании нескольких устройств. При использовании одного устройства номера каналов устройства совпадают с общим номером канала системы.

Imgbuf
Буфер, в котором помещено полутоновое изображение для обработки. Буфер представляет собой либо одно поле, либо два соседних полукадра расположенных один за другим размером HorFrameSize* VerFrameSize пикселя в зависимости от величины параметра **Flag**. Данные расположены обычным образом: первый пиксель в левом верхнем углу.

Flag
Признак обработки одного полукадра или 2 полукадров при вызове процедуры:
RUN_FRAME 1 - обработка двух полукадров
RUN_FIELD 0 - обработка одного полукадров
RUN_ZONEONLY - обработка канала без выполнения процедур распознавания номерного знака. Результат доступен по флагам RESUL_FINDZONE, RESULT_CARPOSITION.

Результат:
Функция возвращает следующие значения:

NOT_FOUND (0) - не найдено ни одной зоны
NUMBER_CORRECTED (1) - имеются найденные зоны с номерами
ONLY_ZONE (2) - найдены только зоны без распознанных знаков
FATAL_ERROR (3) - ошибка инициализации процедур распознавания
ERROR_KEY (4) - ошибка обработки ключа

Если установлен флаг **DOUBLE_FRAME (0x80)**, то произведена обработка двух соседних полукадров изображения

Примечание:

Данная функция вызывается при готовности изображения, полученного с вводного устройства.

```
int LibRun(int *ch)
{
int RunChannel;
TCHANNELPROPERTY prop;
HDEVICE Hdv;
UINT ChInDev;

RunChannel = *ch;

CHGetProperty(RunChannel,&prop);

Hdv= CHGetDevice(RunChannel);
ChInDev = RunChannel-IDGetFirstChannel(CHGetDevices(RunChannel));

//Если в функции CFSetProperty PixelMode флаг = UYVY_PIXELS
//следующая функция не выполняется
if(prop.ColorCode==MAKEFOURCC('U','Y','V','Y'))
        YVYUtoGRAY(SaveImage[RunChannel],SaveImage[RunChannel],
                                prop.HorFrameSize,prop.VerFrameSize);
```

```
if((prop.PixelMode&DOUBLEFIELD_PIXELS)!=0)
       retv = CFRun(Hdv,ChInDev,SaveImage[RunChannel],RUN_FRAME);
  else retv = CFRun(Hdv,ChInDev,SaveImage[RunChannel],RUN_FIELD);

return 0;
```

- **Процедура определение готовности обработанных данных**

UINT CFQuery (HDEVICE Hdv, UINT Channel, UINT QueryFlag);

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Channel
Номер канала, по которому производится проверка готовности данных. Номер канала определяется только для данного устройства ввода и не соответствует номеру общего числа каналов при использовании нескольких устройств. При использовании одного устройства номера каналов устройства совпадают с общим номером канала системы.

QueryFlag
Тип данных необходимых для тестирования.

Определение	Величина	Значение
RESULT_CORRECTION	0x00000001	Проверка количества зон, в которых проводится корректировка номерного знака.
RESULT_LOST	0x00000002	Проверка окончания определение номера. При данном флаге все данные готовы, зона номера вышла за границы определения номера. Если флаг **Flag** в процедуре **CFRun** был установлен в RUN_FIELD, то окончание определения зоны номера произошло в первом полукадре обрабатываемого изображения. В противном случае данный флаг говорит о том, что результат был получен во втором полукадре, а в первом данная зона имела флаг **RESULT_CORRECTION**.
RESULT_PRELOST	0x00000004	Проверка окончания определение номера. При данном флаге все данные готовы, зона номера вышла за границы определения номера. Данная проверка осуществляется в

		случае если флаг **Flag** в процедуре **CFRun** был установлен в RUN_FRAME Это говорит о том, что окончание определения зоны номера произошло в первом полукадре обрабатываемого изображения.
RESULT_CARPOSITION	0x00000008	Проверка определения позиции автомобиля в кадре.
RESULT_FINDZONE	0x00000010	Проверка всех зон определенных в кадре.

Результат

Функция возвращает количество готовых данных, которые отвечают требуемому значению **QueryFlag**. Данные затем могут быть получены функцией **CFGetResult.**

```
flag = CFQuery(Hdv,ChInDev,RESULT_CORRECTION);
for(int i=0;i<flag;i++){
        //Обработка зон корректировки
        }
```

- **Инициализация процедур определителя номерного знака**

UINT CFGetResult (
HDEVICE Hdv,
UINT Channel,
UINT QueryFlag,
UINT NumberQuery,
PLATERESULT *Result);

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Channel
Номер канала, по которому производится проверка готовности данных. Номер канала определяется только для данного устройства ввода и не соответствует номеру общего числа каналов при использовании нескольких устройств. При использовании одного устройства номера каналов устройства совпадают с общим номером канала системы.

QueryFlag

Тип данных необходимых для тестирования.

Название флага	Описание
RESULT_LOST	Получение результата по готовым зонам.
RESULT_PRELOST	Получение результата по готовым зонам.
RESULT_CORRECTION	Получение данных по зонам, в которых в данный момент проводиться корректировка номерного знака.
RESULT_CARPOSITION	Получение данных о позиции маркеров признака наличия автомобиля.
RESULT_FINDZONE	Получение данных по всем зонам найденным на изображении, включая зоны, в которых определение номера не произошло.

NumberQuery
Номер получаемых данных. Номер находится в пределах от 0 до величины полученной процедурой **CFQuery**.

Result
Указатель на структуру результата:

```
typedef struct TZoneResult {
        UINT        Flag;
        UINT        Status;
        UINT        Resolution;
        RECT        Position;
        CHAR        PlateNumber[16];
        CHAR        PlateGipotiz[16];
        UINT        Heght;
        INT         Angle;
        UINT        Weight;
        UINT        Width;
        UINT        Color;
        UINT        PeekValue;
        UINT        Direction;
        UINT        Speed;
        PVOID       Image_Gray;
        PVOID       Image_Bin;
        PVOID       Image_Lap;
        PVOID       Image_Mask;
        PVOID       Image_Pipe;
        } PLATERESULT;
```

Flag
Соответствует параметру QueryFlag:

RESULT_LOST – структура содержит результирующие данные.
RESULT_CORRECTION – структура содержит данные о коррекции номера и прослеживания зоны номера.
RESULT_PRELOST – структура содержит результирующие данные, которые были определены в первом полукадре при обработке обоих полукадров.
RESULT_FINDZONE – структура содержит данные о всех зонах найденных на изображении.
RESULT_CARPOSUTION – структура содержит данные о положении середины автомобиля в текущем кадре.

Status
При Flag = RESULT_CORRECTION
Определяет наличие распознавания в данной зоне при установленном флаге RESULT_CORRECTION.
0 – зона прослежена, но нет распознавания;
1- зона прослежена, распознавание есть.

Resolution
При Flag = RESULT_LOST, RESULT_PRELOST, RESULT_CORRECTION, RESULT_FINDZONE.
Размер зоны номерного знака, который размещен в буфере Image_Gray:
ZONE_RESOLUTION_96x24 0
ZONE_RESOLUTION_128x32 1
ZONE_RESOLUTION_160x40 3 **(присутствует в версии 0x601)**
ZONE_RESOLUTION_192x48 2
ZONE_RESOLUTION_256x64 4 **(присутствует в версии 0x601)**
ZONE_RESOLUTION_320x80 5 **(присутствует в версии 0x601)**
ZONE_RESOLUTION_384x96 6 **(присутствует в версии 0x601)**

ZONE_RESOLUTION_48x48 10 **(присутствует в версии 0x601)**
ZONE_RESOLUTION_64x64 11 **(присутствует в версии 0x601)**
ZONE_RESOLUTION_80x80 12 **(присутствует в версии 0x601)**
ZONE_RESOLUTION_96x96 13 **(присутствует в версии 0x601)**
ZONE_RESOLUTION_128x128 14 **(присутствует в версии 0x601)**
ZONE_RESOLUTION_160x160 15 **(присутствует в версии 0x601)**
ZONE_RESOLUTION_192x192 16 **(присутствует в версии 0x601)**

Position
При Flag = RESULT_CORRECTION, RESULT_FINDZONE.
Позиция и размер зоны номера на экране для отображения. Позиция определяется в пределах, которые определены функцией **SetCarFlowSize**. Определен прямоугольник, в котором помещается номер.
Position.left, Position.top – положение верхнего левого угла зоны;
Position.right, Position.bottom - положение нижнего правого угла зоны.

При Flag = RESULT_CARPOSITION
Position.left, Position.top – положение маркера автомобиля.

При Flag = RESULT_LOST, RESULT_PRELOST.

Позиция и размер зоны, в которой было наиболее уверенное распознавание номерного знака. Позиция определяется в пределах, которые определены функцией **SetCarFlowSize**. Определен прямоугольник, в котором помещается номер.
Position.left, Position.top – положение верхнего левого угла лучшей зоны;
Position.right, Position.bottom - положение нижнего правого угла лучшей зоны.

PlateNumber
При Flag = RESULT_LOST, RESULT_PRELOST, RESULT_CORRECTION.
Указатель на строку символов номерного знака.

PlateGipotiz
При Flag = RESULT_LOST, RESULT_PRELOST, RESULT_CORRECTION.
Указатель на строку символов гипотезы номерного знака.

Heght
При Flag = RESULT_CORRECTION.
Высота символов номерного знака.

Angle
При Flag = RESULT_CORRECTION.
Угол наклона номерного знака умноженный на 100.

Weight
При Flag = RESULT_CORRECTION.
Вес распознавания в процентах.

Width
При Flag = RESULT_CORRECTION.
Ширина номерного знака в процентах к общей ширине зоны.

Color
При Flag = RESULT_LOST, RESULT_PRELOST, RESULT_CORRECTION.
Цвет номерного знака (‘W’,’B’,’Y’,’R’,’S’ – белый, черный, желтый, красный,синий).

PeekValue
При Flag = RESULT_CORRECTION.
Максимум захвата зоны номерного знака. Данный параметр удобно использовать для установки порогов прослеживания зоны.

Direction
При Flag = RESULT_LOST, RESULT_PRELOST, RESULT_CORRECTION.
Направление движения транспортного средства: 0 – снизу вверх, 1 – сверху в низ.

Speed
При Flag = RESULT_LOST, RESULT_PRELOST, RESULT_CORRECTION.
Скорость движения автомобиля в условных единицах. Если значение 0xFFFFFFFF, то скорость не определена.

Image_Gray
При Flag = RESULT_CORRECTION.
Указатель на полутоновое изображение номерного знака. Изображение номерного знака имеет всегда квадратные пиксели.

При Flag = RESULT_LOST, RESULT_PRELOST.
Указатель на полутоновое изображение номерного знака, которое содержит изображение зоны полученной при наиболее уверенном распознавании номерного знака. Изображение номерного знака имеет всегда квадратные пиксели.

Image_Bin
При Flag = RESULT_CORRECTION.
Указатель на бинарное изображение номерного знака размером 384*96*1. Биты упакованы в байты в соответствии с кодировкой в WINDOWS.

Image_Lap
При Flag = RESULT_CORRECTION.
Указатели на бинарные изображения необходимые для уточнения распознавания (Лапласовское, маскирующее и очищающее изображения) размером 384*96*1 каждое. Биты упакованы в байты в соответствии с кодировкой в WINDOWS.

При Flag = RESULT_LOST, RESULT_PRELOST.
Указатель на кадр, который соответствует наиболее лучшему входному изображению. Данное изображение может быть использовано при занесении в базу данных. Размер изображения и параметры пикселей соответствует входному изображению.

Image_Mask
При Flag = RESULT_CORRECTION.
Указатели на бинарные изображения необходимые для уточнения распознавания (Лапласовское, маскирующее и очищающее изображения) размером 384*96*1 каждое. Биты упакованы в байты в соответствии с кодировкой в WINDOWS.

Image_Pipe
При Flag = RESULT_CORRECTION.
Указатели на бинарные изображения необходимые для уточнения распознавания (Лапласовское, маскирующее и очищающее изображения) размером 384*96*1 каждое. Биты упакованы в байты в соответствии с кодировкой в WINDOWS.

```
int LibRun(int *ch)
{
UINT flag;
int RunChannel;
PLATERESULT Result;
TCHANNELPROPERTY prop;
HDEVICE Hdv;
UINT ChInDev;
```

```
RunChannel = *ch;

CHGetProperty(RunChannel,&prop);

Hdv= CHGetDevice(RunChannel);
ChInDev = RunChannel -IDGetFirstChannel(CHGetDevices(RunChannel));

//Если в функции CFSetProperty PixelMode флаг = UYVY_PIXELS
//следующая функция не выполняется
if(prop.ColorCode==MAKEFOUTCC('U','Y','V','Y'))
        YVYUtoGRAY(SaveImage[RunChannel],SaveImage[RunChannel],
                                prop.HorFrameSize,prop.VerFrameSize);

if((prop.PixelMode&DOUBLEFIELD_PIXELS)!=0)
       retv = CFRun(Hdv,ChInDev,SaveImage[RunChannel],RUN_FRAME);
  else retv = CFRun(Hdv,ChInDev,SaveImage[RunChannel],RUN_FIELD);

flag = CFQuery(Hdv,ChInDev,RESULT_CARPOSITION);
for(int i=0;i<flag;i++){
        CFGetResult(Hdv,ChInDev,RESULT_CARPOSITION,i,&Result);
        //Прорисовка положения автомобиля
        }

flag = CFQuery(Hdv,ChInDev, RESULT_FINDZONE);
for(int i=0;i<flag;i++){
        CFGetResult(Hdv,ChInDev,RESULT_CARPOSITION,i,&Result);
        //Прорисовка всех найденных зон
        }

flag = CFQuery(Hdv,ChInDev,RESULT_CORRECTION);
for(int i=0;i<flag;i++){
        CFGetResult(Hdv,ChInDev,RESULT_CORRECTION,i,&Result);
        //Обработка зон корректировки
        }

flag = CFQuery(Hdv,ChInDev,RESULT_PRELOST);
if((retv&DOUBLE_FRAME)!=DOUBLE_FRAME) flag=0;
for(int i=0;i<flag;i++){
        CFGetResult(Hdv,ChInDev,RESULT_PRELOST,i,&Result);
        //Обработка результата окончания определения номера
        //в первом полукадре
        }

flag = CFQuery(Hdv,ChInDev,RESULT_LOST);
for(int i=0;i<flag;i++){
        CFGetResult(Hdv,ChInDev,RESULT_LOST,i,&Result);
        //Обработка результата окончания определения номера
        }

return 1;
}
```

6.2. Функции определения автомобильных номеров с использованием номеров каналов

При вызове данной группы функций в качестве идентификатора используется только номер канала в полном списке каналов всех активированных устройств. Определение устройства соответствующего каналу определяется автоматически. Максимальное число каналов определяется функцией **IDGetMaxChannels.**

- **Установка параметров обрабатываемого кадра**

```
VOID  CFCHSetChannelProperty (
        UINT Channel,
        UINT SizeX,
        UINT SizeY,
        UINT PixelMode);
```

Параметры устанавливаемого кадра должны соответствовать параметрам вводимого изображения для данного канала. Размеры и параметр ширины пикселей соответствует параметрам черно-белого изображения, которое будет подано для обработки. Данная функция осуществляет установку параметров для всех активированных каналов.

Параметры:

Channel
Номер канала, по которому производится установка параметров изображения. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

SizeX
Размер обрабатываемого кадра по горизонтали.

SizeY
Размер обрабатываемого кадра по вертикали.

PixelMode
Описание пикселей в обрабатываемом кадре. Параметр может содержать комбинацию следующих флагов:

Название флага	Описание
SQUARE_PIXELS	квадратные пиксели.
WIDE_PIXELS	широкие пиксели, изображение по горизонтали имеет двойное разрешение.
GRAY_PIXELS	каждый пиксель 8 бит (серое изображение).
RGB_PIXELS	каждый пиксель 24 бита (цветное RGB изображение).

UYVY_PIXELS	каждый пиксель 16 бит (цветоразностное изображение U-Y-V-Y)
YUY2_PIXELS	каждый пиксель 16 бит (цветоразностное изображение Y-U-Y-V)

Если параметр PixelMode содержит флаг SQUARE_PIXELS, то изображение имеет квадратные пиксели, как изображено на рисунке.

```
//Размер изображения 384*288

CFCHSetProperty(Channel,384,288,SQUARE_PIXELS|GRAY_PIXELS);
```

Если параметр PixelMode содержит флаг WIDE_PIXELS, то изображение имеет широкие пиксели (изображение в ширину имеет двойной размер), как изображено на рисунке.

.

```
//Размер изображения 768*288

CFCHSetProperty(Channel,768,288,WIDE_PIXELS|GRAY_PIXELS);
```

- **Установка исходных параметров канала.**

VOID CFCHRestart (UINT Channel, CARINITPARAMS *ChParams);

Функция устанавливает исходные значения параметров канала для определения номерного знака.

Параметры:

Channel
Номер устанавливаемого канала. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

ChParams
Указатель на структуру параметров канала.

```
typedef struct TNumInitParam {
        UINT CarThreshold;
        UINT ZoneThreshold;
        UINT ZoneMode;
        RECT Zone_visio;
        UINT Thhory;
        UINT Thhorx;
        UINT Thbin;
        UINT Min_Pipe_Size;
        UINT Max_Pipe_Size;
      } NUMINITPARAMS;
```

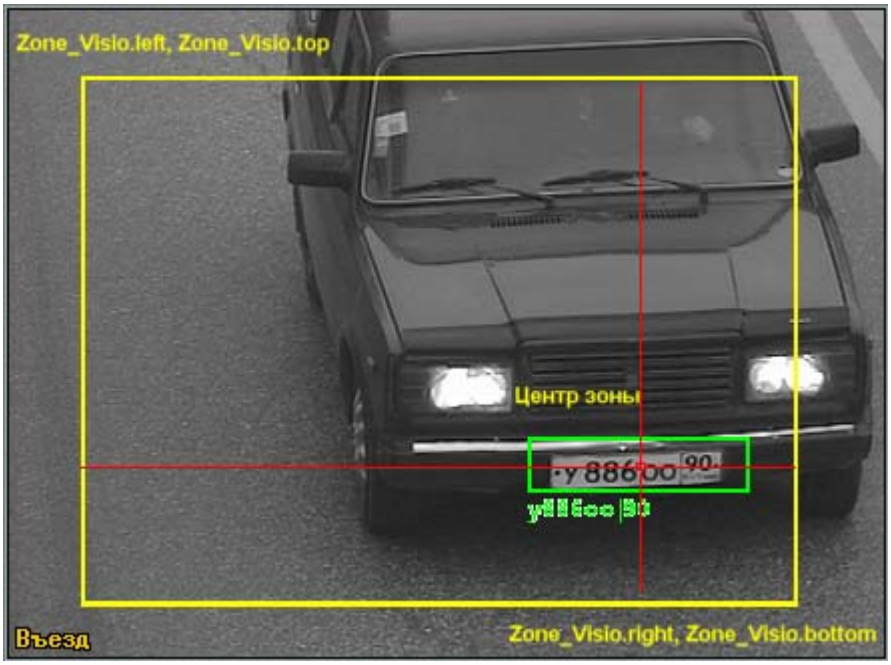
CarThreshold=168
Чувствительность системы к появлению центра автомобиля в поле зрения. Данная величина изменяется в пределах от 100 до 255 и подбирается опытным путем при установке системы.

ZoneThreshold=168 (в версии 0x601 данный параметр не используется)
Чувствительность системы к поиску зоны номерного знака. Данная величина изменяется в пределах от 100 до 255 и подбирается опытным путем при установке системы.
ZoneMode
0 – обычный режим 96*24 используется для совместимости с ранними версиями;
Сочитание следующих флагов позволяет производить многозонный поиск номеров:
ZONE_MODE_96x24 – разрешение обработки 96*24 , что соответствует высоте символа номера 12 строк;
ZONE_MODE_128x32 – разрешение обработки 128*32 , что соответствует высоте символа номера 16 строк;
ZONE_MODE_192x48 – разрешение обработки 192*48 , что соответствует высоте символа номера 24 строк;

ZONE_MODE_160x40 – разрешение обработки 160*40 , что соответствует высоте символа номера 20 строк **(присутствует в версии 0x601)**;
ZONE_MODE_256x64 – разрешение обработки 256*64 , что соответствует высоте символа номера 32 строк **(присутствует в версии 0x601)**;
ZONE_MODE_320x80 – разрешение обработки 320*80 , что соответствует высоте символа номера 40 строк **(присутствует в версии 0x601)**;
ZONE_MODE_384x96 – разрешение обработки 384*96 , что соответствует высоте символа номера 48 строк **(присутствует в версии 0x601)**;

ZONE_MODE_48x48 – разрешение обработки 48*48 , что соответствует высоте символа номера 12 строк для двухстрочных номеров **(присутствует в версии 0x601)**;
ZONE_MODE_64x64 – разрешение обработки 64*64 , что соответствует высоте символа номера 16 строк для двухстрочных номеров **(присутствует в версии 0x601)**;
ZONE_MODE_80x80 – разрешение обработки 80*80 , что соответствует высоте символа номера 20 строк для двухстрочных номеров **(присутствует в версии 0x601)**;
ZONE_MODE_96x96 – разрешение обработки 96*96 , что соответствует высоте символа номера 24 строк для двухстрочных номеров **(присутствует в версии 0x601)**;
ZONE_MODE_128x128 – разрешение обработки 128*128 , что соответствует высоте символа номера 32 строк для двухстрочных номеров **(присутствует в версии 0x601)**;
ZONE_MODE_160x160 – разрешение обработки 160*160 , что соответствует высоте символа номера 40 строк для двухстрочных номеров **(присутствует в версии 0x601)**;
ZONE_MODE_192x192 – разрешение обработки 192*192 , что соответствует высоте символа номера 48 строк для двухстрочных номеров **(присутствует в версии 0x601)**.

Zone_visio
Определяет границы изображения, в которых производится поиск центра зоны номерного знака. Размер по горизонтали соответствует размеру квадратных пикселей. Например, если реальный размер изображения по горизонтали равен 768 и флаг WIDE_PIXELS установлен, то границы поиска зоны определяются в пределах от 0 до 383.



Thhory (по умолчанию 208)
Параметр определяет степень очистки бинарного изображения сверху и снизу от символов номерного знака. **(в версии 0x601 данный параметр не используется)**

Thhorx (по умолчанию 202)
Параметр определяет степень очистки бинарного изображения слева и справа от символов номерного знака. **(в версии 0x601 данный параметр не используется)**

Thbin=136
Параметр определяет степень очистки бинарного изображения от помех. **(в версии 0x601 данный параметр не используется)**

Min Pipe Size=0
Минимально допустимая ширина номера в процентах по отношению к ширине зоны номерного знака. **(в версии 0x601 данный параметр не используется)**
Max Pipe Size=100
Максимально допустимая ширина номера в процентах по отношению к ширине зоны номерного знака. **(в версии 0x601 данный параметр не используется)**

```
void StartAllChannel()
{
UINT MaxChannels;
UINT *UsesChannels;
TCHANNELPROPERTY prop;

MaxChannels = IDGetMaxChannels();
UsesChannels = new UINT [MaxChannels];
for(int i=0;i<MaxChannels;i++) {
     UsesChannels[i]=3;
```

```
     CHSetParams(i,&INPUTPARAMS);
     CFCHRestart(i,&NUMINITPARAMS);
     CHGetProperty(i,&prop);
     CFCHSetChannelProperty(i,prop.HorFrameSize,
                                 prop.VerFrameSize,
                                    prop.PixelMode&SQUARE_PIXELS);
     }

CHInput(UsesChannels);
CFError = CFOpen(0,"c:\\ProgramFiles\\MegaLib");

delete UsesChannels;
}
//*****************************************************************
void StopAllChannel()
{
UINT MaxChannels;
UINT *UsesChannels;

MaxChannels = IDGetMaxChannels();
UsesChannels = new UINT [MaxChannels];
for(int i=0;i<MaxChannels;i++) UsesChannels[i]=0;

CFClose(CFError);
CHInput(UsesChannels);

delete UsesChannels;
}
```

- **Установка параметров распознавания канала**

VOID CFCHSetRecognitionParams(UINT Channel, NUMINITRECPARAMS *Params);

Функция устанавливает исходные параметры распознавания для определения номерного знака.

Параметры:

Channel
Номер устанавливаемого канала. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

Params
Указатель на структуру параметров распознавания.

```
typedef struct TNumInitRecognitionParam {
        UINT SymbolRecFlag;
        UINT ZoneRecFlag;
        UINT ErrorCorrectionTime;
        UINT WeightThreshold;
        UINT Reserved1;
        UINT Reserved2;
```

```
  UINT Reserved3;
} NUMINITRECPARAMS;
```

SymbolRecFlag
Флаги, отвечающие за посимвольную структуризацию номерного знака.

SYM_ENABLED_CLEARNUMBER (0x00000001)
Флаг разрешает удаление объектов из номера, которые не соответствуют признакам алфавитно-цифрового символа. Таким образом, происходит очистка номера от загрязнений при бинаризации. В случае низкого качества бинарного изображения, при котором символы могут соединяться друг с другом данный флаг следует выключить.

SYM_ENABLED_CORRECTION (0x00000002)
Флаг разрешает дополнительную корректировку символов после проведения распознавания в случае отсутствия гипотезы. Осуществляется проведение анализа последовательности отдельно определенных символов и построения вероятностной окончательной гипотезы.

SYM_ENABLED_DELETEZONES (0x00000004)
Флаг разрешает исключение зон, в которых невозможно определить отдельные буквенно-цифровые символы. Данные зоны считаются не номерными и исключаются из обработки.

ZoneRecFlag
Флаги, влияющие на исключения номеров, не отвечающих определенным критериям.

ZONE_ENABLED_CHECK_SIZE (0x00000001)
При работе системы в поле зрения камеры попадают надписи или структуры, которые не являются номерами. Для отсечения подобных надписей в программе устанавливаются предельные значения высоты символов, которые считаются недопустимыми. Для высоты символа в 10 пикселей границы составляют от 6 до 15, для символов в 15 пикселей – от 10 до 20 и для символов в 20 пикселей – от 15 до 25 и т.д.. Если данный флаг установлен, номера не удовлетворяющие критерию высоты символа исключаются из обработки.

ZONE_ENABLED_CHECK_WEIGHT (0x00000002)
В процессе определения номера формируется параметр вероятности распознавания. Данный параметр может принимать значение от 0 до 100. . Если данный флаг установлен, данные с вероятностью меньше, либо равно величине **WeightTreshold** блокируются системой и не передаются на выход.

ZONE_ENABLED_CHECK_GIPOTIZE (0x00000004)
В процессе определения номера формируется гипотеза, которая соответствует типу номерного знака. При невозможности определения типа формируется нулевая гипотеза. Если флаг установлен, такие номера не передаются на выход.

Установка данного флаг позволяет исключить появления «ложных» номеров, но и не позволяет осуществить попытку построения вероятностной гипотезы.

ZONE_ENABLED_CHECK_LOST (0x00000008)
В процессе определения номера возможно пропадание зоны номера на короткий промежуток времени, что вызывает выработку признака окончания распознавания с последующим возобновлением прослеживания номера. Данный флаг предотвращает повторное определение одного и того же номера. Распознанный номер хранится в списке распознанных номеров около 2.5 секунд и блокирует повторное распознавание.

ErrorCorrectionTime
Данный параметр определяет количество коррекций номерного знака при удачно определенной зоне и неудачно распознанного номера. Число определяет количество неудачных циклов распознавания, после которых срабатывает признак окончания распознавания.

WeightTreshold
Минимальное значение вероятности, при которой номер рассматривается как распознанный. Действует при установленном флаге **ZONE_ENABLED_CHECK_WEIGHT.**

- **Выполнение процедуры определения номерного знака**

UINT CFCHRun (UINT Channel, void *Imgbuf, UINT Flag);

Параметры:

Channel
Номер канала обработки. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

Imgbuf
Буфер, в котором помещено полутоновое изображение для обработки. Буфер представляет собой либо одно поле, либо два соседних полукадра расположенных один за другим размером HorFrameSize* VerFrameSize пикселя в зависимости от величины параметра **Flag**. Данные расположены обычным образом: первый пиксель в левом верхнем углу.

Flag
Признак обработки одного полукадра или 2 полукадров при вызове процедуры:
RUN_FRAME 1 - обработка двух полукадров
RUN_FIELD 0 - обработка одного полукадров
RUN_ZONEONLY - обработка канала без выполнения процедур распознавания номерного знака. Результат доступен по флагам RESUL_FINDZONE, RESULT_CARPOSITION.

Результат:
Функция возвращает следующие значения:

NOT_FOUND (0)	- не найдено ни одной зоны
NUMBER_CORRECTED (1)	- имеются найденные зоны с номерами
ONLY_ZONE (2)	- найдены только зоны без распознанных знаков
FATAL_ERROR (3)	- ошибка инициализации процедур распознавания
ERROR_KEY (4)	- ошибка обработки ключа

Если установлен флаг **DOUBLE_FRAME (0x80)**, то произведена обработка двух соседних полукадров изображения

Примечание:

Данная функция вызывается при готовности изображения, полученного с вводного устройства.

```
int LibRun(int *ch)
{
int RunChannel;
TCHANNELPROPERTY prop;

RunChannel = *ch;

CHGetProperty(RunChannel,&prop);

//Если в функции CFSetProperty PixelMode флаг = UYVY_PIXELS
//следующая функция не выполняется
if(prop.ColorCode==MAKEFOURCC('U','Y','V','Y'))
        YVYUtoGRAY(SaveImage[RunChannel],
                        SaveImage[RunChannel],
                              prop.HorFrameSize,prop.VerFrameSize);

if((prop.PixelMode&DOUBLEFIELD_PIXELS)!=0)
        retv = CFCHRun(RunChannel,SaveImage[RunChannel],RUN_FRAME);
   else retv = CFCHRun(RunChannel,SaveImage[RunChannel],RUN_FIELD);

return 0;
```

- **Процедура определение готовности обработанных данных**

UINT CFCHQuery(UINT Channel, UINT QueryFlag);

Параметры:

Channel
Номер канала, по которому производится проверка готовности данных. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

QueryFlag
Тип данных необходимых для тестирования.

Определение	Величина	Значение
RESULT_CORRECTION	0x00000001	Проверка количества зон, в которых проводится корректировка номерного знака.
RESULT_LOST	0x00000002	Проверка окончания определение номера. При данном флаге все данные готовы, зона номера вышла за границы определения номера. Если флаг **Flag** в процедуре **CFRun** был установлен в RUN_FIELD, то окончание определения зоны номера произошло в первом полукадре обрабатываемого изображения. В противном случае данный флаг говорит о том, что результат был получен во втором полукадре, а в первом данная зона имела флаг **RESULT_CORRECTION**.
RESULT_PRELOST	0x00000004	Проверка окончания определение номера. При данном флаге все данные готовы, зона номера вышла за границы определения номера. Данная проверка осуществляется в случае если флаг **Flag** в процедуре **CFRun** был установлен в RUN_FRAME Это говорит о том, что окончание определения зоны номера произошло в первом полукадре обрабатываемого изображения.
RESULT_CARPOSITION	0x00000008	Проверка определения позиции автомобиля в кадре.
RESULT_FINDZONE	0x00000010	Проверка всех зон определенных в кадре.

Результат

Функция возвращает количество готовых данных, которые отвечают требуемому значению **QueryFlag**. Данные затем могут быть получены функцией **CFGetResult.**

```
UINT flag = CFCHQuery(RunChannel,RESULT_CORRECTION);
for(int i=0;i<flag;i++){
        //Обработка зон корректировки
        }
```

- **Инициализация процедур определителя номерного знака**

UINT CFCHGetResult (
UINT Channel,
UINT QueryFlag,
UINT NumberQuery,
PLATERESULT *Result);

Параметры:

Channel
Номер канала, по которому производится проверка готовности данных.

QueryFlag
Тип данных необходимых для тестирования.

Название флага	Описание
RESULT_LOST	Получение результата по готовым зонам.
RESULT_PRELOST	Получение результата по готовым зонам.
RESULT_CORRECTION	Получение данных по зонам, в которых в данный момент проводиться корректировка номерного знака.
RESULT_CARPOSITION	Получение данных о позиции маркеров признака наличия автомобиля.
RESULT_FINDZONE	Получение данных по всем зонам найденным на изображении, включая зоны, в которых определение номера не произошло.

NumberQuery
Номер получаемых данных. Номер находится в пределах от 0 до величины полученной процедурой **CFQuery**.

Result
Указатель на структуру результата:

```
typedef struct TZoneResult {
        UINT        Flag;
        UINT        Status;
        UINT        Resolution;
        RECT        Position;
        CHAR        PlateNumber[16];
        CHAR        PlateGipotiz[16];
        UINT        Heght;
        INT         Angle;
        UINT        Weight;
        UINT        Width;
        UINT        Color;
        UINT        PeekValue;
        UINT        Direction;
        UINT        Speed;
        PVOID       Image_Gray;
        PVOID       Image_Bin;
        PVOID       Image_Lap;
        PVOID       Image_Mask;
        PVOID       Image_Pipe;
        } PLATERESULT;
```

Flag
Соответствует параметру QueryFlag:

RESULT_LOST – структура содержит результирующие данные.
RESULT_CORRECTION – структура содержит данные о коррекции номера и прослеживания зоны номера.
RESULT_PRELOST – структура содержит результирующие данные, которые были определены в первом полукадре при обработке обоих полукадров.
RESULT_FINDZONE – структура содержит данные о всех зонах найденных на изображении.
RESULT_CARPOSUTION – структура содержит данные о положении середины автомобиля в текущем кадре.

Status
При Flag = RESULT_CORRECTION
Определяет наличие распознавания в данной зоне при установленном флаге RESULT_CORRECTION.
0 – зона прослежена, но нет распознавания;
1- зона прослежена, распознавание есть.

Resolution
При Flag = RESULT_LOST, RESULT_PRELOST, RESULT_CORRECTION, RESULT_FINDZONE.
Размер зоны номерного знака, который размещен в буфере Image_Gray:
ZONE_RESOLUTION_96x24 0
ZONE_RESOLUTION_128x32 1
ZONE_RESOLUTION_160x40 3 **(присутствует в версии 0x601)**
ZONE_RESOLUTION_192x48 2
ZONE_RESOLUTION_256x64 4 **(присутствует в версии 0x601)**
ZONE_RESOLUTION_320x80 5 **(присутствует в версии 0x601)**
ZONE_RESOLUTION_384x96 6 **(присутствует в версии 0x601)**

ZONE_RESOLUTION_48x48 10 **(присутствует в версии 0x601)**
ZONE_RESOLUTION_64x64 11 **(присутствует в версии 0x601)**
ZONE_RESOLUTION_80x80 12 **(присутствует в версии 0x601)**
ZONE_RESOLUTION_96x96 13 **(присутствует в версии 0x601)**
ZONE_RESOLUTION_128x128 14 **(присутствует в версии 0x601)**
ZONE_RESOLUTION_160x160 15 **(присутствует в версии 0x601)**
ZONE_RESOLUTION_192x192 16 **(присутствует в версии 0x601)**

Position
При Flag = RESULT_CORRECTION, RESULT_FINDZONE.
Позиция и размер зоны номера на экране для отображения. Позиция определяется в пределах, которые определены функцией **SetCarFlowSize**. Определен прямоугольник, в котором помещается номер.
Position.left, Position.top – положение верхнего левого угла зоны;
Position.right, Position.bottom - положение нижнего правого угла зоны.

При Flag = RESULT_CARPOSITION
Position.left, Position.top – положение маркера автомобиля.

При Flag = RESULT_LOST, RESULT_PRELOST.
Позиция и размер зоны, в которой было наиболее уверенное распознавание номерного знака. Позиция определяется в пределах, которые определены функцией **SetCarFlowSize**. Определен прямоугольник, в котором помещается номер.
Position.left, Position.top – положение верхнего левого угла лучшей зоны;
Position.right, Position.bottom - положение нижнего правого угла лучшей зоны.

PlateNumber
При Flag = RESULT_LOST, RESULT_PRELOST, RESULT_CORRECTION.
Указатель на строку символов номерного знака.

PlateGipotiz
При Flag = RESULT_LOST, RESULT_PRELOST, RESULT_CORRECTION.
Указатель на строку символов гипотезы номерного знака.

Heght
При Flag = RESULT_CORRECTION.
Высота символов номерного знака.

Angle
При Flag = RESULT_CORRECTION.
Угол наклона номерного знака умноженный на 100.

Weight
При Flag = RESULT_CORRECTION.
Вес распознавания в процентах.

Width
При Flag = RESULT_CORRECTION.
Ширина номерного знака в процентах к общей ширине зоны.

Color
При Flag = RESULT_LOST, RESULT_PRELOST, RESULT_CORRECTION.
Цвет номерного знака (‘W’,’B’,’Y’,’R’,’S’ – белый, черный, желтый, красный,синий).

PeekValue
При Flag = RESULT_CORRECTION.
Максимум захвата зоны номерного знака. Данный параметр удобно использовать для установки порогов прослеживания зоны.

Direction
При Flag = RESULT_LOST, RESULT_PRELOST, RESULT_CORRECTION.
Направление движения транспортного средства: 0 – снизу вверх, 1 – сверху в низ.

Speed
При Flag = RESULT_LOST, RESULT_PRELOST, RESULT_CORRECTION.
Скорость движения автомобиля в условных единицах. Если значение 0xFFFFFFFF, то скорость не определена.

Image_Gray
При Flag = RESULT_CORRECTION.
Указатель на полутоновое изображение номерного знака. Изображение номерного знака имеет всегда квадратные пиксели.

При Flag = RESULT_LOST, RESULT_PRELOST.
Указатель на полутоновое изображение номерного знака, которое содержит изображение зоны полученной при наиболее уверенном распознавании номерного знака. Изображение номерного знака имеет всегда квадратные пиксели.

Image_Bin
При Flag = RESULT_CORRECTION.
Указатель на бинарное изображение номерного знака размером 384*96*1. Биты упакованы в байты в соответствии с кодировкой в WINDOWS.

Image_Lap
При Flag = RESULT_CORRECTION.

Указатели на бинарные изображения необходимые для уточнения распознавания (Лапласовское, маскирующее и очищающее изображения) размером 384*96*1 каждое. Биты упакованы в байты в соответствии с кодировкой в WINDOWS.

При Flag = RESULT_LOST, RESULT_PRELOST.
Указатель на кадр, который соответствует наиболее лучшему входному изображению. Данное изображение может быть использовано при занесении в базу данных. Размер изображения и параметры пикселей соответствует входному изображению.

Image_Mask
При Flag = RESULT_CORRECTION.
Указатели на бинарные изображения необходимые для уточнения распознавания (Лапласовское, маскирующее и очищающее изображения) размером 384*96*1 каждое. Биты упакованы в байты в соответствии с кодировкой в WINDOWS.

Image_Pipe
При Flag = RESULT_CORRECTION.
Указатели на бинарные изображения необходимые для уточнения распознавания (Лапласовское, маскирующее и очищающее изображения) размером 384*96*1 каждое. Биты упакованы в байты в соответствии с кодировкой в WINDOWS.

```
int LibRun(int *ch)
{
UINT flag;
int RunChannel;
PLATERESULT Result;
TCHANNELPROPERTY prop;

RunChannel = *ch;

CHGetProperty(RunChannel,&prop);

//Если в функции CFSetProperty PixelMode флаг = UYVY_PIXELS
//следующая функция не выполняется
if(prop.ColorCode==MAKEFOURCC('U','Y','V','Y'))
        YVYUtoGRAY(SaveImage[RunCounter],
                        SaveImage[RunCounter],
                              prop.HorFrameSize,prop.VerFrameSize);

if((prop.PixelMode&DOUBLEFIELD_PIXELS)!=0)
        retv = CFCHRun(RunChannel,SaveImage[RunCounter],RUN_FRAME);
   else retv = CFCHRun(RunChannel,SaveImage[RunCounter],RUN_FIELD);

flag = CFCHQuery(RunChannel,RESULT_CARPOSITION);
for(int i=0;i<flag;i++){
        CFCHGetResult(RunChannel,RESULT_CARPOSITION,i,&Result);
        //Прорисовка положения автомобиля
        }

flag = CFCHQuery(RunChannel,RESULT_FINDZONE);
for(int i=0;i<flag;i++){
        CFCHGetResult(RunChannel,RESULT_CARPOSITION,i,&Result);
        //Прорисовка положения найденных зон
```

```
        }

flag = CFCHQuery(RunChannel,RESULT_CORRECTION);
for(int i=0;i<flag;i++){
        CFCHGetResult(RunChannel,RESULT_CORRECTION,i,&Result);
        //Обработка зон корректировки
        }

flag = CFCHQuery(RunChannel,RESULT_PRELOST);
if((retv&DOUBLE_FRAME)!=DOUBLE_FRAME) flag=0;
for(int i=0;i<flag;i++){
        CFCHGetResult(RunChannel,RESULT_PRELOST,i,&Result);
        //Обработка результата окончания определения номера
        //в первом полукадре
        }

flag = CFCHQuery(RunChannel,RESULT_LOST);
for(int i=0;i<flag;i++){
        CFCHGetResult(RunChannel,RESULT_LOST,i,&Result);
        //Обработка результата окончания определения номера
        }

return 1;
}
```

7. Функции детекции движения

Из нейрофизиологии известно, что зрительная система человека многоканальная, а именно: изображения параллельно проходят через ряд полосовых пространственных фильтров с разным разрешением. В функциях библиотеки реализовано 3 фильтра – точный, средний и грубый. Такая фильтрация позволяет подавить помехи типа: фотонного шума, шума видеосенсора и предающего тракта. Кроме того, игнорируются глобальные изменения яркости сцены. Пространственная фильтрация достаточно затратная вычислительная процедура. Что толку от видеодетектора, который лишь по одному каналу полностью загружает процессор. Однако специалистам компании удалось настолько оптимизировать вычисления, за счёт SIMD команд процессора (MMX, SSE) и структуры видеоданных (промежуточные вычисления никогда не выходят за пределы одного байта на пиксель), что система может одновременно обрабатывать 16 каналов в режиме real-time – по 25 fps. А фактически – параллельно работают 48 видеодетекторов.

Для подавления так называемых высокоскоростных помех компания МегаПиксел разработала фирменную технологию **SlowSpeedFilter**. Алгоритм высокоэффективен для борьбы с такими природными явлениями как: дождь, снег, листва, мошкара, пролетающие птицы и другими быстрыми объектами, имеющими высокие угловые скорости перемещения.

Каждое устройство содержит элементы защиты или имеет соответствующее ему защитное устройство. Каждый элемент защиты имеет информацию о возможности использования той или иной части библиотеки. Для проверки возможности использования функций детекции движения, необходимо выполнить функцию **IDGetDeviceProperty** и проверить бит DEVMDENABLED в параметре **DeviceFlags**.

Как и с функциями виртуальных устройств существуют два уровня обработки: обслуживание с использованием идентификатора устройства (в данном случае работа осуществляется непосредственно с каждым отдельно взятым устройством) и на уровне отдельных каналов (библиотека сама определяет требуемое устройство, которое соответствует данному каналу).

7.1. Функции детекции движения с использованием идентификатора устройства

При вызове данной группы функций в качестве параметров определяющих канал обработки используется идентификатор устройства и номер канала в пределах данного устройства.

- **Прочитать версию библиотеки детекции движения**

UINT MDGetVersion(VOID);

Возвращает номер текущей версии функций детекции движений. 0x6XX
0x600 – Версия библиотеки MegaLib V 1.0
0x601 – Версия библиотеки MegaLib V 1.1

- **Установка исходных параметров канала.**

VOID MDSetParam (
HDEVICE Hdv,
UINT Channel,
MOTIONPARAMS *ChParams);

Функция определяет исходные значения параметров канала виртуального устройства. Функция копирует данные из внешнего буфера во внутреннюю структуру параметров.

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Channel
Номер канала, по которому производится установка начальных параметров. Номер канала определяется только для данного устройства ввода и не соответствует номеру общего числа каналов при использовании нескольких устройств. При использовании одного устройства номера каналов устройства совпадают с общим номером канала системы.

ChParams
Указатель на структуру параметров канала.

```
typedef struct TMotion {
    UINT MotionThresholdLow;
    UINT MotionThresholdMedium;
    UINT MotionThresholdHigh;
    UINT MotionNoiseReductionLow;
    UINT MotionNoiseReductionMedium;
```

```
UINT MotionNoiseReductionHigh;
UINT AdaptTime;
UINT SSDetectTime;
UINT SSMotionb;
UINT ThzoneLowDown[MAXZONES];
UINT ThzoneLowUp[MAXZONES];
UINT ThzoneMediumDown[MAXZONES];
UINT ThzoneMediumUp[MAXZONES];
UINT ThzoneHighDown[MAXZONES];
UINT ThzoneHighUp[MAXZONES];
UINT ThzoneSDDDown[MAXZONES];
UINT ThzoneSDDUp[MAXZONES];
char *ZoneMask;
UINT Quality;
UINT DetectFlag;
UINT SDDResolution;
UINT SDDThreshold;
UINT SDDDetectTime;
UINT SDDk1;
UINT SDDk2;
UINT SDDk3;
UINT SDDk4;
UINT SDDNoiseReduction;
UINT SDDMotionb1;
UINT SDDMotionb2;
UINT SDDUseZone[MAXZONES];
TINPUTPARAMS InputParam;
} TMOTIONPARAMS;
```

Значения параметров структуры

MotionThresholdLow
Порог для зон детектора движения низкого разрешения. Этот параметр определяет чувствительность канала и имеет значения в пределах от 100 до 255 (для внутренних помещений эта величина равна 115).

MotionThresholdMedium
Порог для зон детектора движения среднего разрешения. Этот параметр определяет чувствительность канала и имеет значения в пределах от 100 до 255 (для внутренних помещений эта величина равна 115).

MotionThresholdHigh
Порог для зон детектора движения высокого разрешения. Этот параметр определяет чувствительность канала и имеет значения в пределах от 100 до 255 (для внутренних помещений эта величина равна 115).

MotionNoiseReductionLow
Параметр шумоподавления уменьшает влияние одиночных нарушений на выработку тревоги для зон детектора движения низкого разрешения. Данный

параметр изменяется в пределах от 0 до 6 и определяет семь уровней шумоподавления.

MotionNoiseReductionMedium
Параметр шумоподавления уменьшает влияние одиночных нарушений на выработку тревоги для зон детектора движения среднего разрешения. Данный параметр изменяется в пределах от 0 до 6 и определяет семь уровней шумоподавления.

MotionNoiseReductionHigh
Параметр шумоподавления уменьшает влияние одиночных нарушений на выработку тревоги для зон детектора движения высокого разрешения. Данный параметр изменяется в пределах от 0 до 6 и определяет семь уровней шумоподавления.

AdaptTime
Время адаптации. Этот параметр определяет время адаптации канала к условиям изменения сцены в мсек. Диапазон изменения параметра определен в пределах от 0 до 250 сек (0-250000). Величина равная 0 запрещает использование данного параметра.

SSDetectTime
Фильтрация низких скоростей. Этот параметр определяет время, в течение которого объект может находиться в поле зрения до наступления тревоги. Данный параметр может изменяться от 0 до 10000 миллисекунд. Величина равная 0 запрещает использование данного параметра.

SSMotionb
Корректирующие параметры фильтра низких скоростей (3). Подбирается при настройке.

ThzoneLowDown
Порог срабатывания по каждой из 32 зон контроля в режиме низкого разрешения. Данная величина определяет нижний порог срабатывания и в сочетании с параметром ThzoneLowUp определяет приделы допустимых пороговых значений. Величина равная 0 запрещает использование режима низкого разрешения для каждой зоны.

ThzoneLowUp
Верхний порог срабатывания по каждой из 32 зон контроля в режиме низкого разрешения. В сочетании с параметром ThzoneLowDown определяет приделы допустимых пороговых значений.

ThzoneMediumDown
Порог срабатывания по каждой из 32 зон контроля в режиме среднего разрешения. Данная величина определяет нижний порог срабатывания и в сочетании с параметром ThzoneMediumUp определяет приделы допустимых пороговых значений. Величина равная 0 запрещает использование режима среднего разрешения для каждой зоны.

ThzoneMediumUp
Верхний порог срабатывания по каждой из 32 зон контроля в режиме среднего разрешения. В сочетании с параметром ThzoneMediumDown определяет приделы допустимых пороговых значений.

ThzoneHighDown
Порог срабатывания по каждой из 32 зон контроля в режиме высокого разрешения. Данная величина определяет нижний порог срабатывания и в сочетании с параметром ThzoneHighUp определяет приделы допустимых пороговых значений. Величина равная 0 запрещает использование режима высокого разрешения для каждой зоны.

ThzoneHighUp
Верхний порог срабатывания по каждой из 32 зон контроля в режиме высокого разрешения. В сочетании с параметром ThzoneHighDown определяет приделы допустимых пороговых значений.

ThzoneSDDDown
Порог срабатывания по каждой из 32 зон контроля в режиме детектора замедленных движений. Данная величина определяет нижний порог срабатывания и в сочетании с параметром ThzoneSDDUp определяет приделы допустимых пороговых значений. Величина равная 0 запрещает использование режима детектора замедленных движений для каждой зоны.

ThzoneSDDUp
Верхний порог срабатывания по каждой из 32 зон контроля в режиме детектора замедленных движений. В сочетании с параметром ThzoneSDDDown определяет приделы допустимых пороговых значений.

ZoneMask
Указатель на буфер, в котором будут сохранены зоны нарушения. Данный буфер распределяется библиотекой при инициализации виртуального устройства ввода и освобождается автоматически после закрытия устройства.
Буфер сочетает в себе информацию от различных разрешений и формирует общий буфер размером **sizex**/4 на **sizey**/4 (в зависимости от размеров устанавливаемых функцией MDSetProperty).
Для определения типа нарушения введены следующие значения бит:
XX1XXXX1 – сработал детектора движения низкого разрешения
X1XXXXX1 – сработал детектора движения среднего разрешения
1XXXXXX1 – сработал детектора движения высокого разрешения
XXXXXX1X - зоны с повышенным интересом при работе алгоритма SDD
XXXXX1XX – появление нового объекта при работе алгоритма SDD
XXXX1XXX – исчезновение старого объекта при работе алгоритма SDD
XXX1XXXX – рамка старта при работе алгоритма SDD

Quality
Величина качества сохраняемого JPEG изображения.

DetectFlag

Параметр не оказывает никакого влияния и введен для использования в будущем.

SDDResolution
Алгоритм работы канала в SDD режиме:
0 – средний алгоритм работы канала,
1 - грубый алгоритм работы канала,
2 - точный алгоритм работы канала,

SDDThreshold
Порог SDD детектора. Этот параметр определяет чувствительность канала и имеет значения в пределах от 100 до 255 (для внутренних помещений эта величина равна 110).

SDDDetectTime
Время срабатывания SDD детектора в миллисекундах.

SDDk1 ((0.50-2.00)*100) (значение 100)
SDDk2 ((0-1.0)*100) (значение 50)
SDDk3 ((1.0-20.0)*100) (значение 900)
SDDk4 ((1.0-20.0)*100) (значение 100)
SDDMotionb1 (0-3)(значение 2)
SDDMotionb2 (0-3)(значение 2)
Корректирующие параметры SDD детектора. Подбирается при настройке.

SDDNoiseReduction
Параметр шумоподавления уменьшает влияние одиночных нарушений на выработку тревоги. Данный параметр изменяется в пределах от 0 до 6 и определяет семь уровней шумоподавления.

SDDUseZone
Параметр использования SDD детектора:
0 – детектор не используется
1 – детектор используется для детекции вновь появившихся объектов
2 – детектор используется для детекции исчезнувших объектов
3 – детектор используется для детекции появ./исчез. объектов

InputParam
Структура, используемая в функции **SetInputDeviceParams**

- **Инициализация канала.**

VOID MDRestart(HDEVICE Hdv, UINT Channel);

Функция инициализирует канал обработки.

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Channel
Номер канала, по которому производится инициализация. Номер канала определяется только для данного устройства ввода и не соответствует номеру общего числа каналов при использовании нескольких устройств. При использовании одного устройства номера каналов устройства совпадают с общим номером канала системы.

- **Установка параметров обрабатываемого кадра**

VOID MDSetChannelProperty (
HDEVICE Hdv,
UINT Channel,
UINT sizex,
UINT sizey,
UINT PixelMode);

Параметры устанавливаемого кадра должны соответствовать параметрам вводимого изображения для данного канала. Размеры и параметр ширины пикселей соответствует параметрам черно-белого изображения, которое будет подано для обработки.

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Channel
Номер канала, по которому производится установка параметров. Номер канала определяется только для данного устройства ввода и не соответствует номеру общего числа каналов при использовании нескольких устройств. При использовании одного устройства номера каналов устройства совпадают с общим номером канала системы.

sizex
Размер обрабатываемого кадра по горизонтали.

sizey
Размер обрабатываемого кадра по вертикали.

PixelMode
Параметры пикселей в обрабатываемом кадре.
Если PixelMode = SQUARE_PIXELS, то изображение содержит квадратные пиксели.
Если PixelMode = 0, то изображение содержит широкие пиксели (изображение в ширину имеет двойной размер).

```
void StartAllChannel()
```

```
{
UINT MaxChannels;
UINT *UsesChannels;
TCHANNELPROPERTY prop;

MaxChannels = IDGetMaxChannels();
UsesChannels = new UINT [MaxChannels];
for(int i=0;i<MaxChannels;i++) {
     UsesChannels[i]=3;
     CHSetParams(i,&INPUTPARAMS);
     MDSetParam(CHGetDevice(i),
                  i-IDGetFirstChannel(CHGetDevices(i)),&MOTIONPARAMS);
     MDRestart(CHGetDevice(i),i-IDGetFirstChannel(CHGetDevices(i));
     CHGetProperty(i,&prop);
     if((prop.PixelMode&SQUARE_PIXELS)!=0)
              MDSetChannelProperty(CHGetDevice(i),
                  i-IDGetFirstChannel(CHGetDevices(i));
                     prop.HorFrameSize,
                        prop.VerFrameSize,prop.PixelMode&SQUARE_PIXELS);
     if((prop.PixelMode&WIDE_PIXELS)!=0) {
              MDSetChannelProperty(CHGetDevice(i),
                  i-IDGetFirstChannel(CHGetDevices(i));
                     prop.HorFrameSize/2,
                        prop.VerFrameSize,prop.PixelMode&SQUARE_PIXELS);
     MDLoadMask(CHGetDevice(i),(char *)MaskBuf[i]);
     }

CHInput(UsesChannels);

delete UsesChannels;
}
```

- **Установка параметров обрабатываемого кадра для всех каналов**

VOID MDSetProperty (
HDEVICE Hdv,
UINT sizex,
UINT sizey,
UINT PixelMode);

Параметры устанавливаемого кадра должны соответствовать параметрам вводимого изображения для данного канала. Размеры и параметр ширины пикселей соответствует параметрам черно-белого изображения, которое будет подано для обработки. Данная функция осуществляет установку параметров для всех активированных каналов.

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

sizex
Размер обрабатываемого кадра по горизонтали.

sizey

Размер обрабатываемого кадра по вертикали.

PixelMode
Параметры пикселей в обрабатываемом кадре.
Если PixelMode = SQUARE_PIXELS, то изображение содержит квадратные пиксели.
Если PixelMode = 0, то изображение содержит широкие пиксели (изображение в ширину имеет двойной размер).

- **Обрабатывающая функция канала.**

```
UINT MDRun(
        HDEVICE Hdv,
        UINT Channel,
        MOTIONRESULT *MotResult,
        void *imgbuf,
        UINT Flags,
        char *mask);
```

Функция производит обработку данных и формирование результатов обработки.

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Channel
Номер канала, по которому производится обработка. Номер канала определяется только для данного устройства ввода и не соответствует номеру общего числа каналов при использовании нескольких устройств. При использовании одного устройства номера каналов устройства совпадают с общим номером канала системы.

MotResult
Указатель на структуру результата обработки.

```
typedef struct TMotionResult {
        UINT resultLow;
        UINT resultMedium;
        UINT resultHigh;
        UINT resultSDD;
        } TMOTIONRESULT;
```

resultLow
Результат работы грубого алгоритма детектора движения.

resultMedium
Результат работы среднего алгоритма детектора движения.

resultHigh
Результат работы точного алгоритма детектора движения.

resultSDD
Результат работы SDD детектора.

Бит взведенный в каждой из вышеприведенных переменных соответствует зоне детекции.

imgbuf
Исходное изображение полученное функцией CreateInputDeviceImages (obuf2) . Ч/б изображение размером SizeX * SizeY первый байт в верхнем левом углу.

Flags
Флаги необходимые для работы детектора движения. Параметр должен быть сочетанием следующих флагов
DETECT_LOW 0x40 грубый алгоритм используется
DETECT_MEDIUM 0x80 средний алгоритм используется
DETECT_HIGH 0x100 точный алгоритм используется

mask
Указатель на буфер приемник маски нарушений. В данный буфер копируется содержимое буфера **ZoneMask**

```
int MotionDetectorRun(int *ch)
{
UINT flag,k;
UINT RunChannel;
TCHANNELPROPERTY prop;
char *MaskImage;
HDEVICE Hdv;
UINT ChInDev;
MOTIONRESULT MotResult;
char *ImageBuffer;

RunChannel=*ch;

CHGetProperty(RunCounter,&prop);

Hdv= CHGetDevice(RunChannel);
ChInDev = RunChannel -IDGetFirstChannel(CHGetDevices(RunChannel));

if((prop.PixelMode&WIDE_PIXELS)!=0) {
          ImageBuffer = new char [prop.HorFrameSize/2*prop.VerFrameSize];
          if(prop.ColorCode==MAKEFOURCC('U','Y','V','Y'))
                 YVYUtoGRAY(SaveImage[RunCounter],SaveImage[RunCounter],
                              prop.HorFrameSize,prop.VerFrameSize);
          GRAYHREDUCE(SaveImage[RunCounter],ImageBuffer,
                       prop.HorFrameSize,prop.VerFrameSize);
          }
    else  {
          ImageBuffer = new char [prop.HorFrameSize*prop.VerFrameSize];
          if(prop.ColorCode==MAKEFOURCC('U','Y','V','Y'))
                 YVYUtoGRAY(SaveImage[RunCounter],
                        ImageBuffer,prop.HorFrameSize,prop.VerFrameSize);
          if(prop.ColorCode== MAKEFOURCC('G','R','A','Y'))
```

```
                memcpy(ImageBuffer,SaveImage[RunCounter],
                      prop.HorFrameSize*prop.VerFrameSize);
           }

flag=0;

for(UINT i=0;i<MAXZONES;i++) {
        if(MotParam.ThzoneLowDown[i])    flag|=DETECT_LOW;
        if(MotParam.ThzoneMediumDown[i]) flag|=DETECT_MEDIUM;
        if(MotParam.ThzoneHighDown[i])   flag|=DETECT_HIGH;
        }

if((prop.PixelMode&WIDE_PIXELS)!=0)
          MaskImage = new char [prop.HorFrameSize/2/4*prop.VerFrameSize/4];
    else  MaskImage = new char [prop.HorFrameSize/4*prop.VerFrameSize/4];

MDRun(Hdv,ChInDev,&MotResult,ImageBuffer,flag,MaskImage);

flag = MotResult.resultLow|
          MotResult.resultHigh|
              MotResult.resultMedium|
                  MotResult.resultSDD;

if(flag) {/*Обработка тревожной ситуации*/}

delete MaskImage;
delete ImageBuffer;

return 0;
}
```

- **Загрузка масок канала.**

VOID MDLoadMask(HDEVICE Hdv,UINT Channel,VOID *Msk);

Функция загружает заранее установленные маски из внешнего буфера.

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Channel
Номер канала, по которому производится установка маски. Номер канала определяется только для данного устройства ввода и не соответствует номеру общего числа каналов при использовании нескольких устройств. При использовании одного устройства номера каналов устройства совпадают с общим номером канала системы.

Msk
Указатель на комбинированный буфер, который содержит маску канала и 32 зонные маски. Структура буфера следующая:

```
struct Mask{
        char ChannelMask[sizeX/4 * sizeY/4 / 8];
```

char ChannelZoneMask[32][sizeX/8 * sizeY/8 / 8];
}

Где: **sizex** и **sizey** параметры, которые устанавливаются функцией DSetProperty.

Маски упакованы по 8 бит – младший бит первый

7.2. Функции детекции движения с использованием номеров каналов

При вызове данной группы функций в качестве идентификатора используется только номер канала в полном списке каналов всех активированных устройств. Определение устройства соответствующего каналу определяется автоматически. Максимальное число каналов определяется функцией **IDGetMaxChannels.**

- **Установка параметров обрабатываемого кадра**

```
VOID  MDCHSetChannelProperty (
        UINT Channel,
        UINT sizex,
        UINT sizey,
        UINT PixelMode);
```

Параметры устанавливаемого кадра должны соответствовать параметрам вводимого изображения для данного канала. Размеры и параметр ширины пикселей соответствует параметрам черно-белого изображения, которое будет подано для обработки.

Параметры:

Channel
Номер канала, по которому производится установка параметров. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

sizex
Размер обрабатываемого кадра по горизонтали.

sizey
Размер обрабатываемого кадра по вертикали.

PixelMode
Параметры пикселей в обрабатываемом кадре.
Если PixelMode = SQUARE_PIXELS, то изображение содержит квадратные пиксели.
Если PixelMode = 0, то изображение содержит широкие пиксели (изображение в ширину имеет двойной размер).

- **Установка исходных параметров канала.**

```
VOID MDCHSetParam (
        UINT Channel,
        MOTIONPARAMS *ChParams);
```

Функция определяет исходные значения параметров канала виртуального устройства. Функция копирует данные из внешнего буфера во внутреннюю структуру параметров.

Параметры:

Channel
Номер канала, по которому производится установка начальных параметров. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

ChParams
Указатель на структуру параметров канала.

```
typedef struct TMotion {
        UINT MotionThresholdLow;
        UINT MotionThresholdMedium;
        UINT MotionThresholdHigh;
        UINT MotionNoiseReductionLow;
        UINT MotionNoiseReductionMedium;
        UINT MotionNoiseReductionHigh;
        UINT AdaptTime;
        UINT SSDetectTime;
        UINT SSMotionb;
        UINT ThzoneLowDown[MAXZONES];
        UINT ThzoneLowUp[MAXZONES];
        UINT ThzoneMediumDown[MAXZONES];
        UINT ThzoneMediumUp[MAXZONES];
        UINT ThzoneHighDown[MAXZONES];
        UINT ThzoneHighUp[MAXZONES];
        UINT ThzoneSDDDown[MAXZONES];
        UINT ThzoneSDDUp[MAXZONES];
        char *ZoneMask;
        UINT Quality;
        UINT DetectFlag;
        UINT SDDResolution;
        UINT SDDThreshold;
        UINT SDDDetectTime;
        UINT SDDk1;
        UINT SDDk2;
        UINT SDDk3;
        UINT SDDk4;
        UINT SDDNoiseReduction;
        UINT SDDMotionb1;
        UINT SDDMotionb2;
        UINT SDDUseZone[MAXZONES];
        TINPUTPARAMS InputParam;
        } TMOTIONPARAMS;
```

Значения параметров структуры

MotionThresholdLow
Порог для зон детектора движения низкого разрешения. Этот параметр определяет чувствительность канала и имеет значения в пределах от 100 до 255 (для внутренних помещений эта величина равна 115).

MotionThresholdMedium
Порог для зон детектора движения среднего разрешения. Этот параметр определяет чувствительность канала и имеет значения в пределах от 100 до 255 (для внутренних помещений эта величина равна 115).

MotionThresholdHigh
Порог для зон детектора движения высокого разрешения. Этот параметр определяет чувствительность канала и имеет значения в пределах от 100 до 255 (для внутренних помещений эта величина равна 115).

MotionNoiseReductionLow
Параметр шумоподавления уменьшает влияние одиночных нарушений на выработку тревоги для зон детектора движения низкого разрешения. Данный параметр изменяется в пределах от 0 до 6 и определяет семь уровней шумоподавления.

MotionNoiseReductionMedium
Параметр шумоподавления уменьшает влияние одиночных нарушений на выработку тревоги для зон детектора движения среднего разрешения. Данный параметр изменяется в пределах от 0 до 6 и определяет семь уровней шумоподавления.

MotionNoiseReductionHigh
Параметр шумоподавления уменьшает влияние одиночных нарушений на выработку тревоги для зон детектора движения высокого разрешения. Данный параметр изменяется в пределах от 0 до 6 и определяет семь уровней шумоподавления.

AdaptTime
Время адаптации. Этот параметр определяет время адаптации канала к условиям изменения сцены в мсек. Диапазон изменения параметра определен в пределах от 0 до 250 сек (0-250000). Величина равная 0 запрещает использование данного параметра.

SSDetectTime
Фильтрация низких скоростей. Этот параметр определяет время, в течение которого объект может находиться в поле зрения до наступления тревоги. Данный параметр может изменяться от 0 до 10000 миллисекунд. Величина равная 0 запрещает использование данного параметра.

SSMotionb

Корректирующие параметры фильтра низких скоростей (3). Подбирается при настройке.

ThzoneLowDown
Порог срабатывания по каждой из 32 зон контроля в режиме низкого разрешения. Данная величина определяет нижний порог срабатывания и в сочетании с параметром ThzoneLowUp определяет приделы допустимых пороговых значений. Величина равная 0 запрещает использование режима низкого разрешения для каждой зоны.

ThzoneLowUp
Верхний порог срабатывания по каждой из 32 зон контроля в режиме низкого разрешения. В сочетании с параметром ThzoneLowDown определяет приделы допустимых пороговых значений.

ThzoneMediumDown
Порог срабатывания по каждой из 32 зон контроля в режиме среднего разрешения. Данная величина определяет нижний порог срабатывания и в сочетании с параметром ThzoneMediumUp определяет приделы допустимых пороговых значений. Величина равная 0 запрещает использование режима среднего разрешения для каждой зоны.

ThzoneMediumUp
Верхний порог срабатывания по каждой из 32 зон контроля в режиме среднего разрешения. В сочетании с параметром ThzoneMediumDown определяет приделы допустимых пороговых значений.

ThzoneHighDown
Порог срабатывания по каждой из 32 зон контроля в режиме высокого разрешения. Данная величина определяет нижний порог срабатывания и в сочетании с параметром ThzoneHighUp определяет приделы допустимых пороговых значений. Величина равная 0 запрещает использование режима высокого разрешения для каждой зоны.

ThzoneHighUp
Верхний порог срабатывания по каждой из 32 зон контроля в режиме высокого разрешения. В сочетании с параметром ThzoneHighDown определяет приделы допустимых пороговых значений.

ThzoneSDDDown
Порог срабатывания по каждой из 32 зон контроля в режиме детектора замедленных движений. Данная величина определяет нижний порог срабатывания и в сочетании с параметром ThzoneSDDUp определяет приделы допустимых пороговых значений. Величина равная 0 запрещает использование режима детектора замедленных движений для каждой зоны.

ThzoneSDDUp
Верхний порог срабатывания по каждой из 32 зон контроля в режиме детектора замедленных движений. В сочетании с параметром ThzoneSDDDown определяет приделы допустимых пороговых значений.

ZoneMask
Указатель на буфер, в котором будут сохранены зоны нарушения. Данный буфер распределяется библиотекой при инициализации виртуального устройства ввода и освобождается автоматически после закрытия устройства.
Буфер сочетает в себе информацию от различных разрешений и формирует общий буфер размером **sizex**/4 на **sizey**/4 (в зависимости от размеров устанавливаемых функцией MDSetProperty).
Для определения типа нарушения введены следующие значения бит:
XX1XXXX1 – сработал детектора движения низкого разрешения
X1XXXXX1 – сработал детектора движения среднего разрешения
1XXXXXX1 – сработал детектора движения высокого разрешения
XXXXXX1X - зоны с повышенным интересом при работе алгоритма SDD
XXXXX1XX – появление нового объекта при работе алгоритма SDD
XXXX1XXX – исчезновение старого объекта при работе алгоритма SDD
XXX1XXXX – рамка старта при работе алгоритма SDD

Quality
Величина качества сохраняемого JPEG изображения.

DetectFlag
Параметр не оказывает никакого влияния и введен для использования в будущем.

SDDResolution
Алгоритм работы канала в SDD режиме:
0 – средний алгоритм работы канала,
1 - грубый алгоритм работы канала,
2 - точный алгоритм работы канала,

SDDThreshold
Порог SDD детектора. Этот параметр определяет чувствительность канала и имеет значения в пределах от 100 до 255 (для внутренних помещений эта величина равна 110).

SDDDetectTime
Время срабатывания SDD детектора в миллисекундах.

SDDk1 ((0.50-2.00)*100) (значение 100)
SDDk2 ((0-1.0)*100) (значение 50)
SDDk3 ((1.0-20.0)*100) (значение 900)
SDDk4 ((1.0-20.0)*100) (значение 100)
SDDMotionb1 (0-3)(значение 2)
SDDMotionb2 (0-3)(значение 2)
Корректирующие параметры SDD детектора. Подбирается при настройке.

SDDNoiseReduction
Параметр шумоподавления уменьшает влияние одиночных нарушений на выработку тревоги. Данный параметр изменяется в пределах от 0 до 6 и определяет семь уровней шумоподавления.

SDDUseZone
Параметр использования SDD детектора:
0 – детектор не используется
1 – детектор используется для детекции вновь появившихся объектов
2 – детектор используется для детекции исчезнувших объектов
3 – детектор используется для детекции появ./исчез. объектов

InputParam
Структура, используемая в функции **SetInputDeviceParams**

- **Инициализация канала.**

VOID MDCHRestart(UINT Channel);

Функция инициализирует канал обработки.

Параметры:

Channel
Номер канала, по которому производится инициализация. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

```
void StartAllChannel()
{
UINT MaxChannels;
UINT *UsesChannels;
TCHANNELPROPERTY prop;

MaxChannels = IDGetMaxChannels();
UsesChannels = new UINT [MaxChannels];
for(int i=0;i<MaxChannels;i++) {
     UsesChannels[i]=3;
     CHSetParams(i,&INPUTPARAMS);
     MDCHSetParam(i,&MOTIONPARAMS);
     MDCHRestart(i);
     CHGetProperty(i,&prop);
     if((prop.PixelMode&SQUARE_PIXELS)!=0)
              MDCHSetChannelProperty(i,
                     prop.HorFrameSize,
                        prop.VerFrameSize,prop.PixelMode&SQUARE_PIXELS);
     if((prop.PixelMode&WIDE_PIXELS)!=0) {
              MDCHSetChannelProperty(i,
                     prop.HorFrameSize/2,
                        prop.VerFrameSize,prop.PixelMode&SQUARE_PIXELS);
     MDCHLoadMask(i,(char *)MaskBuf[i]);
     }

CHInput(UsesChannels);

delete UsesChannels;
}
```

- **Обрабатывающая функция канала.**

```
UINT MDCHRun(
        UINT Channel,
        MOTIONRESULT *MotResult,
        void *imgbuf,
        UINT Flags,
        char *mask);
```

Функция производит обработку данных и формирование результатов обработки.

Параметры:

Channel
Номер канала, по которому производится обработка. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

MotResult
Указатель на структуру результата обработки.

```
typedef struct TMotionResult {
        UINT resultLow;
        UINT resultMedium;
        UINT resultHigh;
        UINT resultSDD;
        } TMOTIONRESULT;
```

resultLow
Результат работы грубого алгоритма детектора движения.

resultMedium
Результат работы среднего алгоритма детектора движения.

resultHigh
Результат работы точного алгоритма детектора движения.

resultSDD
Результат работы SDD детектора.

Бит взведенный в каждой из вышеприведенных переменных соответствует зоне детекции.

imgbuf
Исходное изображение полученное функцией CreateInputDeviceImages (obuf2) .
Ч/б изображение размером SizeX * SizeY первый байт в верхнем левом углу.

Flags

Флаги необходимые для работы детектора движения. Параметр должен быть сочетанием следующих флагов
DETECT_LOW 0x40 грубый алгоритм используется
DETECT_MEDIUM 0x80 средний алгоритм используется
DETECT_HIGH 0x100 точный алгоритм используется

mask
Указатель на буфер приемник маски нарушений. В данный буфер копируется содержимое буфера **ZoneMask**

```
int MotionDetectorRun(int *ch)
{
UINT flag,k;
UINT RunChannel;
TCHANNELPROPERTY prop;
char *MaskImage;
MOTIONRESULT MotResult;
char *ImageBuffer;

RunChannel=*ch;

CHGetProperty(RunCounter,&prop);

if((prop.PixelMode&WIDE_PIXELS)!=0) {
          ImageBuffer = new char [prop.HorFrameSize/2*prop.VerFrameSize];
          if(prop.ColorCode==MAKEFOURCC('U','Y','V','Y'))
                YVYUtoGRAY(SaveImage[RunCounter],SaveImage[RunCounter],
                           prop.HorFrameSize,prop.VerFrameSize);
          GRAYHREDUCE(SaveImage[RunCounter],ImageBuffer,
                      prop.HorFrameSize,prop.VerFrameSize);
          }
    else  {
          ImageBuffer = new char [prop.HorFrameSize*prop.VerFrameSize];
          if(prop.ColorCode== MAKEFOURCC('U','Y','V','Y'))
                YVYUtoGRAY(SaveImage[RunCounter],
                      ImageBuffer,prop.HorFrameSize,prop.VerFrameSize);
          if(prop.ColorCode== MAKEFOURCC('G','R','A','Y'))
                memcpy(ImageBuffer,SaveImage[RunCounter],
                      prop.HorFrameSize*prop.VerFrameSize);
           }

flag=0;

for(UINT i=0;i<MAXZONES;i++) {
        if(MotParam.ThzoneLowDown[i])    flag|=DETECT_LOW;
        if(MotParam.ThzoneMediumDown[i]) flag|=DETECT_MEDIUM;
        if(MotParam.ThzoneHighDown[i])   flag|=DETECT_HIGH;
        }

if((prop.PixelMode&WIDE_PIXELS)!=0)
          MaskImage = new char [prop.HorFrameSize/2/4*prop.VerFrameSize/4];
    else  MaskImage = new char [prop.HorFrameSize/4*prop.VerFrameSize/4];

MDCHRun(RunChannel,&MotResult,ImageBuffer,flag,MaskImage);

flag = MotResult.resultLow|
          MotResult.resultHigh|
              MotResult.resultMedium|
                  MotResult.resultSDD;

if(flag) {/*Обработка тревожной ситуации*/}
```

```
delete MaskImage;
delete ImageBuffer;

return 0;
}
```

- **Загрузка масок канала.**

VOID MDCHLoadMask(UINT Channel,VOID *Msk);

Функция загружает заранее установленные маски из внешнего буфера.

Параметры:

Channel
Номер канала, по которому производится установка маски. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

Msk
Указатель на комбинированный буфер, который содержит маску канала и 32 зонные маски. Структура буфера следующая:

```
struct Mask{
        char ChannelMask[sizeX/4 * sizeY/4 / 8];
        char ChannelZoneMask[32][sizeX/8 * sizeY/8 / 8];
        }
```

Где: **sizex** и **sizey** параметры, которые устанавливаются функцией DSetProperty.

Маски упакованы по 8 бит – младший бит первый

8. Функции определения номеров железнодорожных вагонов

Функции библиотеки предназначены для создания приложений по автоматическому распознаванию номеров железнодорожных вагонов. Высокая устойчивость алгоритмов к естественным и искусственным источникам помех позволяет успешно использовать конечные приложения в условиях российского зимнего периода. Библиотека реализована с использованием нейроподобных алгоритмов цифровой обработки изображений

Для проверки возможности использования функций детекции движения, необходимо выполнить функцию **IDGetDeviceProperty** и проверить бит DEVTFENABLED в параметре **DeviceFlags**.

Функции библиотеки способны обрабатывать изображения произвольного размера следующих типов:

- полутоновое изображение (1 байт на пиксель);
- цветоразностное изображение (4 байта на 2 пикселя в формате U-Y-V-Y);
- цветное изображение (3 байта на пиксель в формате RGB).

По типу изображения могут быть:

- с квадратными пикселями;
- с широкими пикселями (разрешение по горизонтали в два раза шире).

Поиск номера осуществляется во всем диапозоне от 6 до 50 пикселей по вертикали.

В ходе обработки функции формируют следующую информацию:

- информацию о распознанных номерах, формирует наиболее качественное результирующее изображение зоны номера, формирует наилучшее изображение транспортного средства;
- информацию о коррекции номера, формирует бинарное и полутоновое изображение текущей зоны номера, передает текущие параметры номера;
- информацию о всех найденных зонах, формирует бинарное и полутоновое изображение каждой зоны;
- информацию о присутствии вагона.

Как и с функциями виртуальных устройств существуют два уровня обработки: обслуживание с использованием идентификатора устройства (в данном случае работа осуществляется непосредственно с каждым отдельно взятым устройством) и на уровне отдельных каналов (библиотека сама определяет требуемое устройство, которое соответствует данному каналу).

8.1. Функции определения номеров железнодорожных выгонов с использованием идентификатора устройства

При вызове данной группы функций в качестве параметров определяющих канал обработки используется идентификатор устройства и номер канала в пределах данного устройства.

- **Прочитать версию библиотеки определения номеров вагонов**

UINT TFGetVersion(VOID);

Возвращает номер текущей версии функций определения номеров железнодорожных вагонов. 0x6XX
0x600 – Версия библиотеки MegaLib V 1.0
0x601 – Версия библиотеки MegaLib V 1.1, MegaLib V 1.2

- **Инициализация процедур определителя номерного знака**

UINT TFOpen (char *LibPath);

LibPath
Указатель на путь к библиотеке. Полный путь к каталогу, в котором располагается файл **MEGALIB1.dll**

Функция возвращает код необходимый для завершения работы процедур распознавания. Код 0 – возвращается в случае удачной инициализации.

- **Окончание работы процедуры определения номерного знака**

VOID TFClose (UINT ErrorCode);

Параметры:

ErrorCode
Параметр, возвращаемый функцией **TFOpen.**

```
// Исполнительный файл находится в каталоге c:\megalib1

UINT Code = TFOpen("C:\\megalib1");

//Использование функций

TFClose ( Code );
```

- **Установка параметров обрабатываемого кадра**

VOID TFSetChannelProperty (
HDEVICE Hdv,
UINT Channel,
UINT sizex,
UINT sizey,
UINT PixelMode);

Параметры устанавливаемого кадра должны соответствовать параметрам вводимого изображения для данного канала. Размеры и параметр ширины пикселей соответствует параметрам черно-белого изображения, которое будет подано для обработки.

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Channel
Номер канала, по которому производится установка параметров. Номер канала определяется только для данного устройства ввода и не соответствует номеру общего числа каналов при использовании нескольких устройств. При использовании одного устройства номера каналов устройства совпадают с общим номером канала системы.

sizex
Размер обрабатываемого кадра по горизонтали.

sizey
Размер обрабатываемого кадра по вертикали.

PixelMode
Описание пикселей в обрабатываемом кадре. Параметр может содержать комбинацию следующих флагов:

Название флага	Описание
SQUARE_PIXELS	квадратные пиксели.
WIDE_PIXELS	широкие пиксели, изображение по горизонтали имеет двойное разрешение.
GRAY_PIXELS	каждый пиксель 8 бит (серое изображение).
RGB_PIXELS	каждый пиксель 24 бита (цветное RGB изображение).
YUY2_PIXELS	каждый пиксель 16 бит (цветоразностное изображение Y-U-Y-V)
UYVY_PIXELS	каждый пиксель 16 бит (цветоразностное изображение U-Y-V-Y)

Если параметр PixelMode содержит флаг SQUARE_PIXELS, то изображение имеет квадратные пиксели, как изображено на рисунке.

```
//Размер изображения 384*288

TFSetProperty(Hdv,384,288,SQUARE_PIXELS|GRAY_PIXELS);
```

Если параметр PixelMode содержит флаг WIDE_PIXELS, то изображение имеет широкие пиксели (изображение в ширину имеет двойной размер), как изображено на рисунке.

.

```
//Размер изображения 768*288

TFSetProperty(Hdv,768,288,WIDE_PIXELS|GRAY_PIXELS);
```

- **Установка параметров обрабатываемого кадра для всех каналов**

VOID TFSetProperty (
HDEVICE Hdv,
UINT sizex,
UINT sizey,
UINT PixelMode);

Параметры устанавливаемого кадра должны соответствовать параметрам вводимого изображения для данного канала. Размеры и параметр ширины пикселей соответствует параметрам черно-белого изображения, которое будет подано для обработки. Данная функция осуществляет установку параметров для всех активированных каналов.

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

sizex
Размер обрабатываемого кадра по горизонтали.

sizey
Размер обрабатываемого кадра по вертикали.

PixelMode
Описание пикселей в обрабатываемом кадре. Параметр может содержать комбинацию следующих флагов:

Название флага	Описание
SQUARE_PIXELS	квадратные пиксели.
WIDE_PIXELS	широкие пиксели, изображение по горизонтали имеет двойное разрешение.
GRAY_PIXELS	каждый пиксель 8 бит (серое изображение).
RGB_PIXELS	каждый пиксель 24 бита (цветное RGB изображение).
YUY2_PIXELS	каждый пиксель 16 бит (цветоразностное изображение Y-U-Y-V)
UYVY_PIXELS	каждый пиксель 16 бит (цветоразностное изображение U-Y-V-Y)

- **Установка исходных параметров канала.**

VOID TFRestart (HDEVICE Hdv,UINT Channel, NUMINITPARAMS *ChParams);

Функция устанавливает исходные значения параметров канала для определения номерного знака.

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Channel
Номер устанавливаемого канала обработки в пределах данного устройства. Номер канала определяется только для данного устройства ввода и не соответствует номеру общего числа каналов при использовании нескольких устройств. При использовании одного устройства номера каналов устройства совпадают с общим номером канала системы.

ChParams
Указатель на структуру параметров канала.

```
typedef struct TNumInitParam {
        UINT CarThreshold;
        UINT ZoneThreshold;
        UINT ZoneMode;
        RECT Zone_visio;
        UINT Thhory;
        UINT Thhorx;
        UINT Thbin;
        UINT Min_Pipe_Size;
        UINT Max_Pipe_Size;
      } NUMINITPARAMS;
```

CarThreshold=168
В системе определения номеров железнодорожных вагонов данный параметр не используется.

ZoneThreshold=168 (в версии 0x601 данный параметр не используется)
Чувствительность системы к поиску зоны номерного знака. Данная величина изменяется в пределах от 100 до 255 и подбирается опытным путем при установке системы.
ZoneMode
0 – обычный режим 96*24 используется для совместимости с ранними версиями;
Сочитание следующих флагов позволяет производить многозонный поиск номеров:
ZONE_MODE_96x24 – разрешение обработки 96*24 , что соответствует высоте символа номера 12 строк;
ZONE_MODE_128x32 – разрешение обработки 128*32 , что соответствует высоте символа номера 16 строк;
ZONE_MODE_160x40 – разрешение обработки 160*40 , что соответствует высоте символа номера 20 строк **(присутствует в версии 0x601)**;
ZONE_MODE_192x48 – разрешение обработки 192*48 , что соответствует высоте символа номера 24 строк;
ZONE_MODE_256x64 – разрешение обработки 256*64 , что соответствует высоте символа номера 32 строк **(присутствует в версии 0x601)**;
ZONE_MODE_320x80 – разрешение обработки 320*80 , что соответствует высоте символа номера 40 строк **(присутствует в версии 0x601)**;
ZONE_MODE_384x96 – разрешение обработки 384*96 , что соответствует высоте символа номера 48 строк **(присутствует в версии 0x601)**;

Zone_visio
Определяет границы изображения, в которых производится поиск зоны номерного знака. Поиск будет осуществлен в пределах :
(Zone_visio.left < центр зоны по x < Zone_visio.left) & (Zone_visio.top < центр зоны по y < Zone_visio.bottom), где значения меняются в зависимости от содержимого структуры TCHANNELPROPERTY в пределах:

если: PixelMode = SQUARE_PIXELS.
0 - HorFrameSize-1 0 - VerFrameSize-1;
если: PixelMode = WIDE_PIXELS.
0 – HorFrameSize/2-1 0 - VerFrameSize-1;
если: PixelMode = SQUARE_PIXELS| DOUBLEFIELD_PIXELS.
0 - HorFrameSize-1 0 – VerFrameSize/2-1;
если: PixelMode = WIDE_PIXELS| DOUBLEFIELD_PIXELS.
0 – HorFrameSize/2-1 0 – VerFrameSize/2-1;

Thhory (по умолчанию 208)
Параметр определяет степень очистки бинарного изображения сверху и снизу от символов номерного знака. **(в версии 0x601 данный параметр не используется)**

Thhorx (по умолчанию 202)
Параметр определяет степень очистки бинарного изображения слева и справа от символов номерного знака. **(в версии 0x601 данный параметр не используется)**

Thbin=136
Параметр определяет степень очистки бинарного изображения от помех. **(в версии 0x601 данный параметр не используется)**

Min Pipe Size=0
Минимально допустимая ширина номера в процентах по отношению к ширине зоны номерного знака. **(в версии 0x601 данный параметр не используется)**

Max Pipe Size=100
Максимально допустимая ширина номера в процентах по отношению к ширине зоны номерного знака.
(в версии 0x601 данный параметр не используется)

```
void StartAllChannel()
{
UINT MaxChannels;
UINT *UsesChannels;
TCHANNELPROPERTY prop;

MaxChannels = IDGetMaxChannels();
UsesChannels = new UINT [MaxChannels];
for(int i=0;i<MaxChannels;i++) {
     UsesChannels[i]=3;
     CHSetParams(i,&INPUTPARAMS);
     TFRestart(CHGetDevice(i),
                  i-IDGetFirstChannel(CHGetDevices(i)),
                      &NUMINITPARAMS);
     CHGetProperty(i,&prop);
     TFSetChannelProperty(CHGetDevice(i),
                           i-IDGetFirstChannel(CHGetDevices(i)),
                              prop.HorFrameSize,
                                 prop.VerFrameSize,
                                    prop.PixelMode&SQUARE_PIXELS);
     }

CHInput(UsesChannels);
```

```
CFError = TFOpen(0,"c:\\ProgramFiles\\MegaLib");

delete UsesChannels;
}
//*********************************************************************
void StopAllChannel()
{
UINT MaxChannels;
UINT *UsesChannels;

MaxChannels = IDGetMaxChannels();
UsesChannels = new UINT [MaxChannels];
for(int i=0;i<MaxChannels;i++) UsesChannels[i]=0;

TFClose(CFError);
CHInput(UsesChannels);

delete UsesChannels;
}
```

- **Установка параметров распознавания**

VOID TFSetRecognitionParams(HDEVICE Hdv,UINT Channel, NUMINITRECPARAMS *Params);

Функция устанавливает исходные параметры распознавания для определения номерного знака.

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Channel
Номер устанавливаемого канала. Номер канала определяется только для данного устройства ввода и не соответствует номеру общего числа каналов при использовании нескольких устройств. При использовании одного устройства номера каналов устройства совпадают с общим номером канала системы.

Params
Указатель на структуру параметров распознавания.

```
typedef struct TNumInitRecognitionParam {
        UINT SymbolRecFlag;
        UINT ZoneRecFlag;
        UINT ErrorCorrectionTime;
        UINT WeightThreshold;
        UINT Reserved1;
        UINT Reserved2;
        UINT Reserved3;
      } NUMINITRECPARAMS;
```

SymbolRecFlag
Данные флаги в распознавании номеров железнодорожных вагонов не применяются.

ZoneRecFlag
Флаги, влияющие на исключения номеров, не отвечающих определенным критериям.

ZONE_ENABLED_CHECK_SIZE (**0x00000001)**
При работе системы в поле зрения камеры попадают надписи или структуры, которые не являются номерами. Для отсечения подобных надписей в программе устанавливаются предельные значения высоты символов, которые считаются недопустимыми. Для высоты символа в 10 пикселей границы составляют от 6 до 15, для символов в 15 пикселей – от 10 до 20 и для символов в 20 пикселей – от 15 до 25 и т.д.. Если данный флаг установлен, номера не удовлетворяющие критерию высоты символа исключаются из обработки.

ZONE_ENABLED_CHECK_WEIGHT (0x00000002)
В процессе определения номера формируется параметр вероятности распознавания. Данный параметр может принимать значение от 0 до 100. . Если данный флаг установлен, данные с вероятностью меньше, либо равно величине **WeightTreshold** блокируются системой и не передаются на выход.

ZONE_ENABLED_CHECK_GIPOTIZE (0x00000004)
В процессе определения номера формируется гипотеза, которая соответствует типу номерного знака. При невозможности определения типа формируется нулевая гипотеза. Если флаг установлен, такие номера не передаются на выход.
Установка данного флаг позволяет исключить появления «ложных» номеров, но и не позволяет осуществить попытку построения вероятностной гипотезы.

ErrorCorrectionTime
Данный параметр не используется.

WeightTreshold
Минимальное значение вероятности, при которой номер рассматривается как распознанный. Действует при установленном флаге **ZONE_ENABLED_CHECK_WEIGHT.**

- **Выполнение процедуры определения номерного знака**

UINT TFRun (HDEVICE Hdv, UINT Channel, void *Imgbuf, UINT Flag);

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Channel
Номер канала обработки. Номер канала определяется только для данного устройства ввода и не соответствует номеру общего числа каналов при использовании нескольких устройств. При использовании одного устройства номера каналов устройства совпадают с общим номером канала системы.

Imgbuf
Буфер, в котором помещено полутоновое изображение для обработки. Буфер представляет собой либо одно поле, либо два соседних полукадра расположенных один за другим размером HorFrameSize* VerFrameSize пикселя в зависимости от величины параметра **Flag**. Данные расположены обычным образом: первый пиксель в левом верхнем углу.

Flag
Признак обработки одного полукадра или 2 полукадров при вызове процедуры:
RUN_FRAME 1 - обработка двух полукадров
RUN_FIELD 0 - обработка одного полукадров
RUN_ZONEONLY - обработка канала без выполнения процедур распознавания номерного знака. Результат доступен по флагам RESUL_FINDZONE, RESULT_CARPOSITION.

Результат:
Функция возвращает следующие значения:

NOT_FOUND (0) - не найдено ни одной зоны
NUMBER_CORRECTED (1) - имеются найденные зоны с номерами
ONLY_ZONE (2) - найдены только зоны без распознанных знаков
FATAL_ERROR (3) - ошибка инициализации процедур распознавания
ERROR_KEY (4) - ошибка обработки ключа

Если установлен флаг **DOUBLE_FRAME (0x80)**, то произведена обработка двух соседних полукадров изображения

Примечание:

Данная функция вызывается при готовности изображения, полученного с вводного устройства.

```
int LibRun(int *ch)
{
int RunChannel;
TCHANNELPROPERTY prop;
HDEVICE Hdv;
UINT ChInDev;

RunChannel = *ch;

CHGetProperty(RunChannel,&prop);
```

```
Hdv= CHGetDevice(RunChannel);
ChInDev = RunChannel-IDGetFirstChannel(CHGetDevices(RunChannel));

//Если в функции CFSetProperty PixelMode флаг = UYVY_PIXELS
//следующая функция не выполняется
if(prop.ColorCode== MAKEFOURCC('U','Y','V','Y'))
        YVYUtoGRAY(SaveImage[RunChannel],SaveImage[RunChannel],
                                prop.HorFrameSize,prop.VerFrameSize);

if((prop.PixelMode&DOUBLEFIELD_PIXELS)!=0)
       retv = TFRun(Hdv,ChInDev,SaveImage[RunChannel],RUN_FRAME);
  else retv = TFRun(Hdv,ChInDev,SaveImage[RunChannel],RUN_FIELD);

return 0;
```

- **Процедура определение готовности обработанных данных**

UINT TFQuery (HDEVICE Hdv, UINT Channel, UINT QueryFlag);

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Channel
Номер канала, по которому производится проверка готовности данных. Номер канала определяется только для данного устройства ввода и не соответствует номеру общего числа каналов при использовании нескольких устройств. При использовании одного устройства номера каналов устройства совпадают с общим номером канала системы.

QueryFlag
Тип данных необходимых для тестирования.

Определение	Величина	Значение
RESULT_CORRECTION	0x00000001	Проверка количества зон, в которых проводится корректировка номерного знака.
RESULT_LOST	0x00000002	Проверка окончания определение номера. При данном флаге все данные готовы, зона номера вышла за границы определения номера. Если флаг **Flag** в процедуре **CFRun** был установлен в RUN_FIELD, то окончание определения зоны номера произошло в первом полукадре обрабатываемого изображения. В противном случае данный флаг говорит о том, что результат был получен во втором полукадре, а в

		первом данная зона имела флаг **RESULT_CORRECTION**.
RESULT_PRELOST	0x00000004	Проверка окончания определение номера. При данном флаге все данные готовы, зона номера вышла за границы определения номера. Данная проверка осуществляется в случае если флаг **Flag** в процедуре **CFRun** был установлен в RUN_FRAME Это говорит о том, что окончание определения зоны номера произошло в первом полукадре обрабатываемого изображения.
RESULT_CARPOSITION	0x00000008	Проверка определения позиции автомобиля в кадре.
RESULT_FINDZONE	0x00000010	Проверка всех зон определенных в кадре.

Результат

Функция возвращает количество готовых данных, которые отвечают требуемому значению **QueryFlag**. Данные затем могут быть получены функцией **CFGetResult.**

```
flag = TFQuery(Hdv,ChInDev,RESULT_CORRECTION);
for(int i=0;i<flag;i++){
        //Обработка зон корректировки
        }
```

- **Инициализация процедур определителя номерного знака**

UINT TFGetResult (
HDEVICE Hdv,
UINT Channel,
UINT QueryFlag,
UINT NumberQuery,
PLATERESULT *Result);

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Channel

Номер канала, по которому производится проверка готовности данных. Номер канала определяется только для данного устройства ввода и не соответствует номеру общего числа каналов при использовании нескольких устройств. При использовании одного устройства номера каналов устройства совпадают с общим номером канала системы.

QueryFlag
Тип данных необходимых для чтения.

Название флага	Описание
RESULT_LOST	Получение результата по готовым зонам.
RESULT_PRELOST	Получение результата по готовым зонам.
RESULT_CORRECTION	Получение данных по зонам, в которых в данный момент проводиться корректировка номерного знака.
RESULT_CARPOSITION	Получение данных о позиции маркеров признака наличия вагона.
RESULT_FINDZONE	Получение данных по всем зонам найденным на изображении, включая зоны, в которых определение номера не произошло.

NumberQuery
Номер получаемых данных. Номер находится в пределах от 0 до величины полученной процедурой **TFQuery** .

Result
Указатель на структуру результата:

```
typedef struct TZoneResult {
        UINT        Flag;
        UINT        Status;
        UINT        Resolution;
        RECT        Position;
        PCHAR       PlateNumber;
        PCHAR       PlateGipotiz;
        UINT        Heght;
        INT         Angle;
        UINT        Weight;
        UINT        Width;
        UINT        Color;
        UINT        PeekValue;
        UINT        Direction;
        UINT        Speed;
        PVOID       Image_Gray;
        PVOID       Image_Bin;
        PVOID       Image_Lap;
```

PVOID Image_Mask;
PVOID Image_Pipe;
} PLATERESULT;

Flag
Соответствует параметру QueryFlag:

RESULT_LOST – структура содержит результирующие данные.
RESULT_CORRECTION – структура содержит данные о коррекции номера и прослеживания зоны номера.
RESULT_PRELOST – структура содержит результирующие данные, которые были определены в первом полукадре при обработке обоих полукадров.
RESULT_FINDZONE – структура содержит данные о всех зонах найденных на изображении.
RESULT_CARPOSUTION – структура содержит данные о положении вагона в текущем кадре.

Status
При Flag = RESULT_CORRECTION
Определяет наличие распознавания в данной зоне при установленном флаге RESULT_CORRECTION.
0 – зона прослежена, но нет распознавания;
1- зона прослежена, распознавание есть.

Resolution
При Flag = RESULT_LOST, RESULT_PRELOST, RESULT_CORRECTION, RESULT_FINDZONE.
Размер зоны номерного знака, который размещен в буфере Image_Gray:
ZONE_RESOLUTION_96x24 0
ZONE_RESOLUTION_128x32 1
ZONE_RESOLUTION_160x40 3 **(присутствует в версии 0x601)**
ZONE_RESOLUTION_192x48 2
ZONE_RESOLUTION_256x64 4 **(присутствует в версии 0x601)**
ZONE_RESOLUTION_320x80 5 **(присутствует в версии 0x601)**
ZONE_RESOLUTION_384x96 6 **(присутствует в версии 0x601)**

Position
При Flag = RESULT_CORRECTION, RESULT_FINDZONE.
Позиция и размер зоны номера на экране для отображения. Позиция определяется в пределах, которые определены функцией **TFSetProperty**. Определен прямоугольник, в котором помещается номер.
Position.left, Position.top – положение верхнего левого угла зоны;
Position.right, Position.bottom - положение нижнего правого угла зоны.

При Flag = RESULT_CARPOSITION
Position.left, Position.top – положение маркера автомобиля.

При Flag = RESULT_LOST, RESULT_PRELOST.

Позиция и размер зоны, в которой было наиболее уверенное распознавание номерного знака. Позиция определяется в пределах, которые определены функцией **TFSetProperty**. Определен прямоугольник, в котором помещается номер.
Position.left, Position.top – положение верхнего левого угла лучшей зоны;
Position.right, Position.bottom - положение нижнего правого угла лучшей зоны.

PlateNumber
При Flag = RESULT_LOST, RESULT_PRELOST, RESULT_CORRECTION.
Указатель на строку символов номерного знака.

PlateGipotiz
При Flag = RESULT_LOST, RESULT_PRELOST, RESULT_CORRECTION.
Указатель на строку символов гипотезы номерного знака.

Heght
При Flag = RESULT_CORRECTION.
Высота символов номерного знака.

Angle
При Flag = RESULT_CORRECTION.
Угол наклона номерного знака умноженный на 100.

Weight
При Flag = RESULT_CORRECTION.
Вес распознавания в процентах.

Width
При Flag = RESULT_CORRECTION.
Ширина номерного знака в процентах к общей ширине зоны.

Color
Не исползуется.

PeekValue
При Flag = RESULT_CORRECTION.
Максимум захвата зоны номерного знака. Данный параметр удобно использовать для установки порогов прослеживания зоны.

Direction
При Flag = RESULT_LOST, RESULT_PRELOST, RESULT_CORRECTION.
Направление движения транспортного средства: 0 – снизу вверх, 1 – сверху в низ.

Speed
При Flag = RESULT_LOST, RESULT_PRELOST, RESULT_CORRECTION.
Скорость движения автомобиля в условных единицах. Если значение 0xFFFFFFFF, то скорость не определена.

Image_Gray

При Flag = RESULT_CORRECTION.
Указатель на полутоновое изображение номерного знака. Изображение номерного знака имеет всегда квадратные пиксели.

При Flag = RESULT_LOST, RESULT_PRELOST.
Указатель на полутоновое изображение номерного знака, которое содержит изображение зоны полученной при наиболее уверенном распознавании номерного знака. Изображение номерного знака имеет всегда квадратные пиксели.

Image_Bin
При Flag = RESULT_CORRECTION.
Указатель на бинарное изображение номерного знака размером 384*96*1. Биты упакованы в байты в соответствии с кодировкой в WINDOWS.

Image_Lap
При Flag = RESULT_CORRECTION.
Указатели на бинарные изображения необходимые для уточнения распознавания (Лапласовское, маскирующее и очищающее изображения) размером 384*96*1 каждое. Биты упакованы в байты в соответствии с кодировкой в WINDOWS.

При Flag = RESULT_LOST, RESULT_PRELOST.
Указатель на кадр, который соответствует наиболее лучшему входному изображению. Данное изображение может быть использовано при занесении в базу данных. Размер изображения и параметры пикселей соответствует входному изображению.

Image_Mask
При Flag = RESULT_CORRECTION.
Указатели на бинарные изображения необходимые для уточнения распознавания (Лапласовское, маскирующее и очищающее изображения) размером 384*96*1 каждое. Биты упакованы в байты в соответствии с кодировкой в WINDOWS.

Image_Pipe
При Flag = RESULT_CORRECTION.
Указатели на бинарные изображения необходимые для уточнения распознавания (Лапласовское, маскирующее и очищающее изображения) размером 384*96*1 каждое. Биты упакованы в байты в соответствии с кодировкой в WINDOWS.

```
int LibRun(int *ch)
{
UINT flag;
int RunChannel;
PLATERESULT Result;
TCHANNELPROPERTY prop;
HDEVICE Hdv;
UINT ChInDev;

RunChannel = *ch;
```

```
CHGetProperty(RunChannel,&prop);

Hdv= CHGetDevice(RunChannel);
ChInDev = RunChannel -IDGetFirstChannel(CHGetDevices(RunChannel));

//Если в функции CFSetProperty PixelMode флаг = UYVY_PIXELS
//следующая функция не выполняется
if(prop.ColorCode==MAKEFOURCC('U','Y','V','Y'))
        YVYUtoGRAY(SaveImage[RunChannel],SaveImage[RunChannel],
                                prop.HorFrameSize,prop.VerFrameSize);

if((prop.PixelMode&DOUBLEFIELD_PIXELS)!=0)
       retv = TFRun(Hdv,ChInDev,SaveImage[RunChannel],RUN_FRAME);
  else retv = TFRun(Hdv,ChInDev,SaveImage[RunChannel],RUN_FIELD);

flag = TFQuery(Hdv,ChInDev,RESULT_CARPOSITION);
for(int i=0;i<flag;i++){
        TFGetResult(Hdv,ChInDev,RESULT_CARPOSITION,i,&Result);
        //Прорисовка положения автомобиля
        }

flag = TFQuery(Hdv,ChInDev, RESULT_FINDZONE);
for(int i=0;i<flag;i++){
        TFGetResult(Hdv,ChInDev,RESULT_CARPOSITION,i,&Result);
        //Прорисовка всех найденных зон
        }

flag = TFQuery(Hdv,ChInDev,RESULT_CORRECTION);
for(int i=0;i<flag;i++){
        TFGetResult(Hdv,ChInDev,RESULT_CORRECTION,i,&Result);
        //Обработка зон корректировки
        }

flag = TFQuery(Hdv,ChInDev,RESULT_PRELOST);
if((retv&DOUBLE_FRAME)!=DOUBLE_FRAME) flag=0;
for(int i=0;i<flag;i++){
        TFGetResult(Hdv,ChInDev,RESULT_PRELOST,i,&Result);
        //Обработка результата окончания определения номера
        //в первом полукадре
        }

flag = TFQuery(Hdv,ChInDev,RESULT_LOST);
for(int i=0;i<flag;i++){
        TFGetResult(Hdv,ChInDev,RESULT_LOST,i,&Result);
        //Обработка результата окончания определения номера
        }

return 1;
}
```

8.2. Функции определения номеров железнодорожных выгонов с использованием номеров каналов

При вызове данной группы функций в качестве идентификатора используется только номер канала в полном списке каналов всех активированных устройств. Определение устройства соответствующего каналу определяется автоматически. Максимальное число каналов определяется функцией **IDGetMaxChannels.**

- **Установка параметров обрабатываемого кадра**

```
VOID  TFCHSetChannelProperty (
        UINT Channel,
        UINT sizex,
        UINT sizey,
        UINT PixelMode);
```

Параметры устанавливаемого кадра должны соответствовать параметрам вводимого изображения для данного канала. Размеры и параметр ширины пикселей соответствует параметрам черно-белого изображения, которое будет подано для обработки. Данная функция осуществляет установку параметров для всех активированных каналов.

Параметры:

Channel
Номер канала, по которому производится установка параметров обработки. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

sizex
Размер обрабатываемого кадра по горизонтали.

sizey
Размер обрабатываемого кадра по вертикали.

PixelMode
Описание пикселей в обрабатываемом кадре. Параметр может содержать комбинацию следующих флагов:

Название флага	Описание
SQUARE_PIXELS	квадратные пиксели.
WIDE_PIXELS	широкие пиксели, изображение по горизонтали имеет двойное разрешение.
GRAY_PIXELS	каждый пиксель 8 бит (серое изображение).
RGB_PIXELS	каждый пиксель 24 бита (цветное RGB изображение).
YUY2_PIXELS	каждый пиксель 16 бит (цветоразностное

	изображение Y-U-Y-V)
UYVY_PIXELS	каждый пиксель 16 бит (цветоразностное изображение U-Y-V-Y)

- **Установка исходных параметров канала.**

VOID TFCHRestart (UINT Channel, NUMINITPARAMS *ChParams);

Функция устанавливает исходные значения параметров канала для определения номерного знака.

Параметры:

Channel
Номер инициализируемого канала. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

ChParams
Указатель на структуру параметров канала.

```
typedef struct TNumInitParam {
        UINT CarThreshold;
        UINT ZoneThreshold;
        UINT ZoneMode;
        RECT Zone_visio;
        UINT Thhory;
        UINT Thhorx;
        UINT Thbin;
        UINT Min_Pipe_Size;
        UINT Max_Pipe_Size;
      } NUMINITPARAMS;
```

CarThreshold=168
В системе определения номеров железнодорожных вагонов данный параметр не используется.

ZoneThreshold=168 (в версии 0x601 данный параметр не используется)
Чувствительность системы к поиску зоны номерного знака. Данная величина изменяется в пределах от 100 до 255 и подбирается опытным путем при установке системы.
ZoneMode
0 – обычный режим 96*24 используется для совместимости с ранними версиями;
Сочитание следующих флагов позволяет производить многозонный поиск номеров:
ZONE_MODE_96x24 – разрешение обработки 96*24 , что соответствует высоте символа номера 12 строк;
ZONE_MODE_128x32 – разрешение обработки 128*32 , что соответствует высоте символа номера 16 строк;

ZONE_MODE_160x40 – разрешение обработки 160*40 , что соответствует высоте символа номера 20 строк **(присутствует в версии 0x601)**;
ZONE_MODE_192x48 – разрешение обработки 192*48 , что соответствует высоте символа номера 24 строк;
ZONE_MODE_256x64 – разрешение обработки 256*64 , что соответствует высоте символа номера 32 строк **(присутствует в версии 0x601)**;
ZONE_MODE_320x80 – разрешение обработки 320*80 , что соответствует высоте символа номера 40 строк **(присутствует в версии 0x601)**;
ZONE_MODE_384x96 – разрешение обработки 384*96 , что соответствует высоте символа номера 48 строк **(присутствует в версии 0x601)**;

Zone_visio
Определяет границы изображения, в которых производится поиск зоны номерного знака. Поиск будет осуществлен в пределах :
(Zone_visio.left < центр зоны по x < Zone_visio.left) & (Zone_visio.top < центр зоны по y < Zone_visio.bottom), где значения меняются в зависимости от содержимого структуры TCHANNELPROPERTY в пределах:

если: PixelMode = SQUARE_PIXELS.
0 - HorFrameSize-1 0 - VerFrameSize-1;
если: PixelMode = WIDE_PIXELS.
0 – HorFrameSize/2-1 0 - VerFrameSize-1;
если: PixelMode = SQUARE_PIXELS| DOUBLEFIELD_PIXELS.
0 - HorFrameSize-1 0 – VerFrameSize/2-1;
если: PixelMode = WIDE_PIXELS| DOUBLEFIELD_PIXELS.
0 – HorFrameSize/2-1 0 – VerFrameSize/2-1;

Thhory (по умолчанию 208)
Параметр определяет степень очистки бинарного изображения сверху и снизу от символов номерного знака. **(в версии 0x601 данный параметр не используется)**

Thhorx (по умолчанию 202)
Параметр определяет степень очистки бинарного изображения слева и справа от символов номерного знака. **(в версии 0x601 данный параметр не используется)**

Thbin=136
Параметр определяет степень очистки бинарного изображения от помех. **(в версии 0x601 данный параметр не используется)**

Min Pipe Size=0
Минимально допустимая ширина номера в процентах по отношению к ширине зоны номерного знака. **(в версии 0x601 данный параметр не используется)**

Max Pipe Size=100
Максимально допустимая ширина номера в процентах по отношению к ширине зоны номерного знака. **(в версии 0x601 данный параметр не используется)**

```
void StartAllChannel()
```

```
{
UINT MaxChannels;
UINT *UsesChannels;
TCHANNELPROPERTY prop;

MaxChannels = IDGetMaxChannels();
UsesChannels = new UINT [MaxChannels];
for(int i=0;i<MaxChannels;i++) {
     UsesChannels[i]=3;
     CHSetParams(i,&INPUTPARAMS);
     TFCHRestart(i,&NUMINITPARAMS);
     CHGetProperty(i,&prop);
     TFCHSetChannelProperty(i,prop.HorFrameSize,
                                prop.VerFrameSize,
                                   prop.PixelMode&SQUARE_PIXELS);
     }

CHInput(UsesChannels);
CFError = TFOpen(0,"c:\\ProgramFiles\\MegaLib");

delete UsesChannels;
}
//*****************************************************************
void StopAllChannel()
{
UINT MaxChannels;
UINT *UsesChannels;

MaxChannels = IDGetMaxChannels();
UsesChannels = new UINT [MaxChannels];
for(int i=0;i<MaxChannels;i++) UsesChannels[i]=0;

TFClose(CFError);
CHInput(UsesChannels);

delete UsesChannels;
}
```

- **Установка параметров распознавания канала**

VOID TFSetRecognitionParams(HDEVICE Hdv,UINT Channel, NUMINITRECPARAMS *Params);

Функция устанавливает исходные параметры распознавания для определения номерного знака.

Параметры:

Hdv
Идентификатор устройства, возвращаемый функцией **IDAdd** или **IDGetDevices**.

Channel
Номер инициализируемого канала. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

Params
Указатель на структуру параметров распознавания.

```
typedef struct TNumInitRecognitionParam {
        UINT SymbolRecFlag;
        UINT ZoneRecFlag;
        UINT ErrorCorrectionTime;
        UINT WeightThreshold;
        UINT Reserved1;
        UINT Reserved2;
        UINT Reserved3;
      } NUMINITRECPARAMS;
```

SymbolRecFlag
Данные флаги в распознавании номеров железнодорожных вагонов не применяются.

ZoneRecFlag
Флаги, влияющие на исключения номеров, не отвечающих определенным критериям.

ZONE_ENABLED_CHECK_SIZE (**0x00000001)**
При работе системы в поле зрения камеры попадают надписи или структуры, которые не являются номерами. Для отсечения подобных надписей в программе устанавливаются предельные значения высоты символов, которые считаются недопустимыми. Для высоты символа в 10 пикселей границы составляют от 6 до 15, для символов в 15 пикселей – от 10 до 20 и для символов в 20 пикселей – от 15 до 25 и т.д.. Если данный флаг установлен, номера не удовлетворяющие критерию высоты символа исключаются из обработки.

ZONE_ENABLED_CHECK_WEIGHT (0x00000002)
В процессе определения номера формируется параметр вероятности распознавания. Данный параметр может принимать значение от 0 до 100. . Если данный флаг установлен, данные с вероятностью меньше, либо равно величине **WeightTreshold** блокируются системой и не передаются на выход.

ZONE_ENABLED_CHECK_GIPOTIZE (0x00000004)
В процессе определения номера формируется гипотеза, которая соответствует типу номерного знака. При невозможности определения типа формируется нулевая гипотеза. Если флаг установлен, такие номера не передаются на выход.
Установка данного флаг позволяет исключить появления «ложных» номеров, но и не позволяет осуществить попытку построения вероятностной гипотезы.

ErrorCorrectionTime
Данный параметр не используется.
WeightTreshold

Минимальное значение вероятности, при которой номер рассматривается как распознанный. Действует при установленном флаге **ZONE_ENABLED_CHECK_WEIGHT.**

- **Выполнение процедуры определения номерного знака**

UINT TFCHRun (UINT Channel, void *Imgbuf, UINT Flag);

Параметры:

Channel
Номер канала обработки. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

Imgbuf
Буфер, в котором помещено полутоновое изображение для обработки. Буфер представляет собой либо одно поле, либо два соседних полукадра расположенных один за другим размером HorFrameSize* VerFrameSize пикселя в зависимости от величины параметра **Flag**. Данные расположены обычным образом: первый пиксель в левом верхнем углу.

Flag
Признак обработки одного полукадра или 2 полукадров при вызове процедуры:
RUN_FRAME 1 - обработка двух полукадров
RUN_FIELD 0 - обработка одного полукадров
RUN_ZONEONLY - обработка канала без выполнения процедур распознавания номерного знака. Результат доступен по флагам RESUL_FINDZONE, RESULT_CARPOSITION.

Результат:
Функция возвращает следующие значения:

NOT_FOUND (0) - не найдено ни одной зоны
NUMBER_CORRECTED (1) - имеются найденные зоны с номерами
ONLY_ZONE (2) - найдены только зоны без распознанных знаков
FATAL_ERROR (3) - ошибка инициализации процедур распознавания
ERROR_KEY (4) - ошибка обработки ключа

Если установлен флаг **DOUBLE_FRAME (0x80)**, то произведена обработка двух соседних полукадров изображения

Примечание:

Данная функция вызывается при готовности изображения, полученного с вводного устройства.

```
int LibRun(int *ch)
{
int RunChannel;
TCHANNELPROPERTY prop;

RunChannel = *ch;

CHGetProperty(RunChannel,&prop);


if((prop.PixelMode&DOUBLEFIELD_PIXELS)!=0)
        retv = TFCHRun(RunChannel,SaveImage[RunChannel],RUN_FRAME);
   else retv = TFCHRun(RunChannel,SaveImage[RunChannel],RUN_FIELD);

return 0;
```

- **Процедура определение готовности обработанных данных**

UINT TFCHQuery(UINT Channel, UINT QueryFlag);

Параметры:

Channel
Номер канала, по которому производится проверка готовности данных. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

QueryFlag
Тип данных необходимых для тестирования.

Определение	Величина	Значение
RESULT_CORRECTION	0x00000001	Проверка количества зон, в которых проводится корректировка номерного знака.
RESULT_LOST	0x00000002	Проверка окончания определение номера. При данном флаге все данные готовы, зона номера вышла за границы определения номера. Если флаг **Flag** в процедуре **CFRun** был установлен в RUN_FIELD, то окончание определения зоны номера произошло в первом полукадре обрабатываемого изображения. В противном случае данный флаг говорит о том, что результат был получен во втором полукадре, а в первом данная зона имела флаг **RESULT_CORRECTION**.
RESULT_PRELOST	0x00000004	Проверка окончания определение номера. При данном флаге все

		данные готовы, зона номера вышла за границы определения номера. Данная проверка осуществляется в случае если флаг **Flag** в процедуре **CFRun** был установлен в RUN_FRAME Это говорит о том, что окончание определения зоны номера произошло в первом полукадре обрабатываемого изображения.
RESULT_CARPOSITION	0x00000008	Проверка определения позиции автомобиля в кадре.
RESULT_FINDZONE	0x00000010	Проверка всех зон определенных в кадре.

Результат

Функция возвращает количество готовых данных, которые отвечают требуемому значению **QueryFlag**. Данные затем могут быть получены функцией **CFGetResult.**

```
UINT flag = TFCHQuery(RunChannel,RESULT_CORRECTION);
for(int i=0;i<flag;i++){
        //Обработка зон корректировки
        }
```

- **Инициализация процедур определителя номерного знака**

UINT CFCHGetResult (
UINT Channel,
UINT QueryFlag,
UINT NumberQuery,
PLATERESULT *Result);

Параметры:

Channel
Номер канала, по которому производится проверка готовности данных. Номер канала соответствует номеру канала для всех устройств, активированных в системе.

QueryFlag
Тип данных необходимых для чтения.

Название флага	Описание
RESULT_LOST	Получение результата по готовым

	зонам.
RESULT_PRELOST	Получение результата по готовым зонам.
RESULT_CORRECTION	Получение данных по зонам, в которых в данный момент проводиться корректировка номерного знака.
RESULT_CARPOSITION	Получение данных о позиции маркеров признака наличия вагона.
RESULT_FINDZONE	Получение данных по всем зонам найденным на изображении, включая зоны, в которых определение номера не произошло.

NumberQuery
Номер получаемых данных. Номер находится в пределах от 0 до величины полученной процедурой **TFQuery** .

Result
Указатель на структуру результата:

```
typedef struct TZoneResult {
        UINT        Flag;
        UINT        Status;
        UINT        Resolution;
        RECT        Position;
        PCHAR       PlateNumber;
        PCHAR       PlateGipotiz;
        UINT        Heght;
        INT         Angle;
        UINT        Weight;
        UINT        Width;
        UINT        Color;
        UINT        PeekValue;
        UINT        Direction;
        UINT        Speed;
        PVOID       Image_Gray;
        PVOID       Image_Bin;
        PVOID       Image_Lap;
        PVOID       Image_Mask;
        PVOID       Image_Pipe;
        } PLATERESULT;
```

Flag
Соответствует параметру QueryFlag:

RESULT_LOST – структура содержит результирующие данные.
RESULT_CORRECTION – структура содержит данные о коррекции номера и прослеживания зоны номера.

RESULT_PRELOST – структура содержит результирующие данные, которые были определены в первом полукадре при обработке обоих полукадров.
RESULT_FINDZONE – структура содержит данные о всех зонах найденных на изображении.
RESULT_CARPOSUTION – структура содержит данные о положении вагона в текущем кадре.

Status
При Flag = RESULT_CORRECTION
Определяет наличие распознавания в данной зоне при установленном флаге RESULT_CORRECTION.
0 – зона прослежена, но нет распознавания;
1- зона прослежена, распознавание есть.

Resolution
При Flag = RESULT_LOST, RESULT_PRELOST, RESULT_CORRECTION, RESULT_FINDZONE.
Размер зоны номерного знака, который размещен в буфере Image_Gray:
ZONE_RESOLUTION_96x24 0
ZONE_RESOLUTION_128x32 1
ZONE_RESOLUTION_160x40 3 **(присутствует в версии 0x601)**
ZONE_RESOLUTION_192x48 2
ZONE_RESOLUTION_256x64 4 **(присутствует в версии 0x601)**
ZONE_RESOLUTION_320x80 5 **(присутствует в версии 0x601)**
ZONE_RESOLUTION_384x96 6 **(присутствует в версии 0x601)**

Position
При Flag = RESULT_CORRECTION, RESULT_FINDZONE.
Позиция и размер зоны номера на экране для отображения. Позиция определяется в пределах, которые определены функцией **TFSetProperty**. Определен прямоугольник, в котором помещается номер.
Position.left, Position.top – положение верхнего левого угла зоны;
Position.right, Position.bottom - положение нижнего правого угла зоны.

При Flag = RESULT_CARPOSITION
Position.left, Position.top – положение маркера автомобиля.

При Flag = RESULT_LOST, RESULT_PRELOST.
Позиция и размер зоны, в которой было наиболее уверенное распознавание номерного знака. Позиция определяется в пределах, которые определены функцией **TFSetProperty**. Определен прямоугольник, в котором помещается номер.
Position.left, Position.top – положение верхнего левого угла лучшей зоны;
Position.right, Position.bottom - положение нижнего правого угла лучшей зоны.

PlateNumber
При Flag = RESULT_LOST, RESULT_PRELOST, RESULT_CORRECTION.
Указатель на строку символов номерного знака.

PlateGipotiz
При Flag = RESULT_LOST, RESULT_PRELOST, RESULT_CORRECTION.
Указатель на строку символов гипотезы номерного знака.

Heght
При Flag = RESULT_CORRECTION.
Высота символов номерного знака.

Angle
При Flag = RESULT_CORRECTION.
Угол наклона номерного знака умноженный на 100.

Weight
При Flag = RESULT_CORRECTION.
Вес распознавания в процентах.

Width
При Flag = RESULT_CORRECTION.
Ширина номерного знака в процентах к общей ширине зоны.

Color
Не исползуется.

PeekValue
При Flag = RESULT_CORRECTION.
Максимум захвата зоны номерного знака. Данный параметр удобно использовать для установки порогов прослеживания зоны.

Direction
При Flag = RESULT_LOST, RESULT_PRELOST, RESULT_CORRECTION.
Направление движения транспортного средства: 0 – снизу вверх, 1 – сверху в низ.

Speed
При Flag = RESULT_LOST, RESULT_PRELOST, RESULT_CORRECTION.
Скорость движения автомобиля в условных единицах. Если значение 0xFFFFFFFF, то скорость не определена.

Image_Gray
При Flag = RESULT_CORRECTION.
Указатель на полутоновое изображение номерного знака. Изображение номерного знака имеет всегда квадратные пиксели.

При Flag = RESULT_LOST, RESULT_PRELOST.
Указатель на полутоновое изображение номерного знака, которое содержит изображение зоны полученной при наиболее уверенном распознавании номерного знака. Изображение номерного знака имеет всегда квадратные пиксели.

Image_Bin

При Flag = RESULT_CORRECTION.
Указатель на бинарное изображение номерного знака размером 384*96*1. Биты упакованы в байты в соответствии с кодировкой в WINDOWS.

Image_Lap
При Flag = RESULT_CORRECTION.
Указатели на бинарные изображения необходимые для уточнения распознавания (Лапласовское, маскирующее и очищающее изображения) размером 384*96*1 каждое. Биты упакованы в байты в соответствии с кодировкой в WINDOWS.

При Flag = RESULT_LOST, RESULT_PRELOST.
Указатель на кадр, который соответствует наиболее лучшему входному изображению. Данное изображение может быть использовано при занесении в базу данных. Размер изображения и параметры пикселей соответствует входному изображению.

Image_Mask
При Flag = RESULT_CORRECTION.
Указатели на бинарные изображения необходимые для уточнения распознавания (Лапласовское, маскирующее и очищающее изображения) размером 384*96*1 каждое. Биты упакованы в байты в соответствии с кодировкой в WINDOWS.

Image_Pipe
При Flag = RESULT_CORRECTION.
Указатели на бинарные изображения необходимые для уточнения распознавания (Лапласовское, маскирующее и очищающее изображения) размером 384*96*1 каждое. Биты упакованы в байты в соответствии с кодировкой в WINDOWS.

```
int LibRun(int *ch)
{
UINT flag;
int RunChannel;
PLATERESULT Result;
TCHANNELPROPERTY prop;

RunChannel = *ch;

CHGetProperty(RunChannel,&prop);


if((prop.PixelMode&DOUBLEFIELD_PIXELS)!=0)
        retv = TFCHRun(RunChannel,SaveImage[RunCounter],RUN_FRAME);
   else retv = TFCHRun(RunChannel,SaveImage[RunCounter],RUN_FIELD);

flag = TFCHQuery(RunChannel,RESULT_CARPOSITION);
for(int i=0;i<flag;i++){
        TFCHGetResult(RunChannel,RESULT_CARPOSITION,i,&Result);
        //Прорисовка положения автомобиля
        }

flag = TFCHQuery(RunChannel,RESULT_FINDZONE);
for(int i=0;i<flag;i++){
```

```
        TFCHGetResult(RunChannel,RESULT_CARPOSITION,i,&Result);
        //Прорисовка положения найденных зон
        }

flag = TFCHQuery(RunChannel,RESULT_CORRECTION);
for(int i=0;i<flag;i++){
        TFCHGetResult(RunChannel,RESULT_CORRECTION,i,&Result);
        //Обработка зон корректировки
        }

flag = TFCHQuery(RunChannel,RESULT_PRELOST);
if((retv&DOUBLE_FRAME)!=DOUBLE_FRAME) flag=0;
for(int i=0;i<flag;i++){
        TFCHGetResult(RunChannel,RESULT_PRELOST,i,&Result);
        //Обработка результата окончания определения номера
        //в первом полукадре
        }

flag = TFCHQuery(RunChannel,RESULT_LOST);
for(int i=0;i<flag;i++){
        TFCHGetResult(RunChannel,RESULT_LOST,i,&Result);
        //Обработка результата окончания определения номера
        }

return 1;
}
```

9. Управляющие библиотеки устройств захвата изображений

Управляющие библиотеки располагаются в подкаталоге **DEVICE*.*** и обеспечивают обслуживание внешних устройств и устройств защиты.

9.1 Устройства, не требующие ключа защиты

MegaFrame4x4.DLL
Обеспечивает ввод изображений при помощи устройства «Мегафрейм-4x4» по 16 каналам в режиме переключения (по 40 мс на канал). Данное устройство снято с производства и в настоящее время недоступно.

MegaFrame4.DLL
Обеспечивает ввод изображений с устройств реализованных на Bt878 в режиме переключения каналов (4 * количество установленных чипов). В качестве таких устройств используются Megaframe4 (1 чип), Megaframe16 (4 чипа).

MegaFrame4_RT.DLL
Обеспечивает ввод изображений с устройств реализованных на Bt878 в режиме реального времени (количество каналов ровно количеству установленных чипов). В качестве таких устройств используются Megaframe4 (1 чип) и Megaframe16 (4 чипа).

MegaFrameX_RT.DLL
Обеспечивает ввод изображений при помощи устройства «Мегафрейм-X» по 8 (16 каналам при использовании двух устройств) в режиме реального времени.

MegaFrameX_2SW.DLL
Обеспечивает ввод изображений при помощи устройства «Мегафрейм-X» по 16 (32 каналам при использовании двух устройств) в режиме быстрого переключения, при котором переключение осуществляется по двум каналам с независимыми входными усилителями.

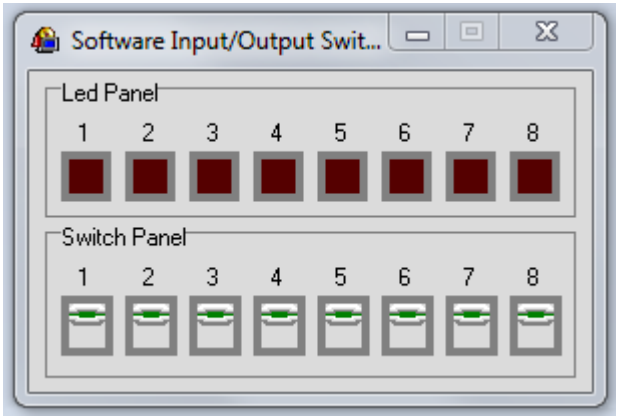
MegaFrameX_4SW.DLL
Обеспечивает ввод изображений при помощи одного устройства «Мегафрейм-X» по 32 в режиме переключения.

AUDIO_Device0.DLL
AUDIO_Device1.DLL
AUDIO_Device2.DLL
AUDIO_Device3.DLL
AUDIO_Device4.DLL
AUDIO_Device5.DLL
AUDIO_Device6.DLL
AUDIO_Device7.DLL
Устройства, которые позволяют вводить аудио данные при помощи устройств, которые работают с DirectShow драйверами (например, «лин. вход (High Defenition Audio Device)»). Система позволяет работать с любым количеством такими устройств. Для увеличения числа доступных устройств необходимо скопировать

любую библиотеку AUDIO_DeviceN.dll в файл AUDIO_DeviceM.dll. Например: AUDIO_Device7.dll в файл AUDIO_Device8.dll, расширив таким образом число аудио устройств до 9. Как правило аудио устройство может захватывать до двух каналов аудио данных.

DLPInOut.DLL
Обеспечивает ввод/вывод цифровых сигналов, подключенных к USB порту. В качестве таких устройств применяются следующие адаптеры:
Устройство DLP-IO8 предназначено для ввода/вывода управляющих цифровых сигналов по 8 линиям. Устройство подключается к USB гнезду компьютера.
Устройство DLP-IOR4 предназначено для вывода управляющих сигналов на исполнительные устройства по 4 линиям. Устройство подключается к USB гнезду компьютера. Позволяет переключать сигналы напряжением до 220 Вольт постоянного и 250 Вольт переменного напряжения с максимальным током 2 А.

SoftInOut.DLL
Обеспечивает ввод/вывод цифровых сигналов имулированных программными средствами непосредственно на исполняемом компьютере. В случае если активизируется данное устройство на эране появляется следующие окно:



При помощи данного приложения можно осуществлять вывод сигналов на Led Panel и ввод со Switch Panel. Данное устройство может использоваться для отладки технологического оборудования или для осуществления контроля над приложениями.

9.2 Устройства, требующие ключа защиты GUARDANT или SenseLockEL

MegaFrame4.DLL
Обеспечивает ввод изображений с устройств реализованных на Bt878 в режиме переключения каналов (4 * количество установленных чипов). В качестве таких устройств используются SDVR1604 (4 чипа), HW104 (1 чип).

MegaFrame4_RT.DLL
Обеспечивает ввод изображений с устройств реализованных на Bt878 в режиме реального времени (количество каналов ровно количеству установленных чипов). В качестве таких устройств используются SDVR1604 (4 чипа), HW104 (1 чип).

MegaFrameE.DLL
Обеспечивает ввод изображений с устройств реализованных на Phillips SAA7114 в режиме переключения каналов (4 * количество установленных чипов). В качестве таких устройств используются MegaframeE (4 чипа). Устройство имеет интерфейс PCI-E.

MegaFrameE_RT.DLL
Обеспечивает ввод изображений с устройств реализованных на Phillips SAA7114 в режиме реального времени (количество каналов ровно количеству установленных чипов). В качестве таких устройств используются MegaframeE (4 чипа). Устройство имеет интерфейс PCI-E.

MFSaa713x.DLL
Обеспечивает ввод изображений с устройств реализованных на микросхемах Phillips SAA713X в режиме переключения каналов (4 * количество установленных чипов).

MFSaa713x_RT.DLL
Обеспечивает ввод изображений с устройств реализованных на микросхемах Phillips SAA713X в режиме одного канала (количество каналов ровно количеству установленных чипов). В качестве таких устройств используются SDVR7008 (8 чипов).

MPTW6816.DLL
Обеспечивает ввод изображений с устройств реализованных на микросхемах Techwell TW6816 в режиме переключения каналов. В качестве таких устройств используется HW-G800X (TW6816 - 8 * 2 установленных чипов).

MPTW6816 _RT.DLL
Обеспечивает ввод изображений с устройств реализованных на микросхемах Techwell TW6816 в режиме реального времени (количество каналов ровно количеству DMA каналов). В качестве таких устройств используется HW-G800X (TW6816 - 4 * количество установленных чипов 2).

MPTW6869 _RT.DLL
Обеспечивает ввод изображений с устройств реализованных на микросхемах Techwell TW6869 в режиме реального времени.

FileEmulator0.DLL
FileEmulator1.DLL
FileEmulator2.DLL
FileEmulator3.DLL
Устройства, которые позволяют вводить изображения из заранее записанных файлов. Каждое такое устройство содержит 1 канал и их можно использовать до 4 устройств. Для работы с данными библиотеками в системе должно присутствовать одно из устройств ООО «Мегапиксел», или ключ GUARDANT. Для активизирования канала необходимо в каталоге, в котором располагается библиотека расположить файл под темже именем с расширением *.INI в котором занесен путь к файлу *.DVR. Например:
В каталоге C:\MEGALIB1\DEVICE\FileEmulator0.INI;

В данном файле помещена запись:
D:\\DVRFILE\\TEST.DVR
В каталоге D:\DVRFILE расположен файл TEST.DVR.

Формат файла *.DVR следующий:
В заголовке располагается структура

```
struct TChannelProperty {
        UINT HorFrameSize;
        UINT VerFrameSize;
        UINT PixelMode;
        UINT ColorCode;
        UINT ChannelFlags;
        UINT Reserved;
      };
```

HorFrameSize
Размер в пикселях вводимого кадра по горизонтали.

VerFrameSize
Размер в пикселях вводимого кадра по вертикали.

PixelMode
Свойства кадра.

Определение	Величина	Значение
WIDE_PIXELS	0x00000001	Каждый полукадр кадр содержит в ширину удвоенные размеры.
SQUARE_PIXELS	0x00000002	Каждый полукадр кадр содержит в ширину квадратные пиксели.
DOUBLEFIELD_PIXELS	0x00000004	Кадр содержит два полукадра, которые следуют один за другим.

ColorCode
Цветовая кодировка каждого пикселя. Может принимать одно из следующих значений:
MAKEFOURCC('G','R','A','Y'); - полутоновое изображение (1 байт на пиксель).
MAKEFOURCC('U','Y','V','Y'); - цветное изображение в формате UYVY (2 байта на пиксель).
MAKEFOURCC('Y','U','Y','2'); - цветное изображение в формате YUY2 (2 байта на пиксель).
MAKEFOURCC('R','G','B',' '); - цветное изображение в формате RGB24 (3 байта на пиксель).

Где:
```
MAKEFOURCC(ch0, ch1, ch2, ch3)                          \
        ((DWORD)(BYTE)(ch0) | ((DWORD)(BYTE)(ch1) << 8) |   \
        ((DWORD)(BYTE)(ch2) << 16) | ((DWORD)(BYTE)(ch3) << 24 ))
```

ChannelFlags
Не используется.

Далее располагаются кадры изображений, которые соответствуют заголовку один за другим.

WDM_Device0.DLL
WDM_Device1.DLL
WDM_Device2.DLL
WDM_Device3.DLL
WDM_Device4.DLL
WDM_Device5.DLL
WDM_Device6.DLL
WDM_Device7.DLL
Устройства, которые позволяют вводить изображения при помощи устройств, которые работают с DirectShow драйверами. Система позволяет работать с 8 такими устройствами. Для работы с данными библиотеками в системе должно присутствовать одно из устройств ООО «Мегапиксел», или ключ GUARDANT или ключ SenseLockEL. В качестве таких устройств могут использоваться TV-тюнеры типа: XCapture USB, ANALOG STICK U100, AVERT MEDIA и т.д.. Как правило подобные устройства могут вводить до двух каналов аудио данных.

Guardant.DLL
Обеспечивает возможность работы библиотеки без использования вводного устройства под защитой ключем GUARDANT. В данном случае используется произвольное устройство захвата изображения, и пользователь берет на себя все действия связанные с вводом и преобразованием изображения.

SenseLockEL.DLL
Обеспечивает возможность работы библиотеки без использования вводного устройства под защитой ключем SenseLock. В данном случае используется произвольное устройство захвата изображения, и пользователь берет на себя все действия связанные с вводом и преобразованием изображения.

Conexant_CX258xx_0.DLL
Conexant_CX258xx_1.DLL
Обеспечивает ввод изображений с устройств реализованных на микросхемах Connexant CX25820, CX25821, CX25858 в режиме реального времени. В качестве таких устройств используются VMX-200-4 на CX25858 (4 канала видео и 4 канала аудио), VMX-200-8 на CX25858 (8 каналов видео и 8 каналов аудио). Устройства на CX25820 вводят 4 канала видео. Устройства на CX25821 вводят 8 канала видео. Все устройства имеют интерфейс PCI-E. В системе возможно использование до 2 таких устройств.

PointGrayCameras.DLL
Обеспечивает ввод изображений с камер высокого разрешения, разработанных канадской фирмой Point Gray. Данная библиотека осуществляет управление камерой «Калибри».

USB3104.DLL

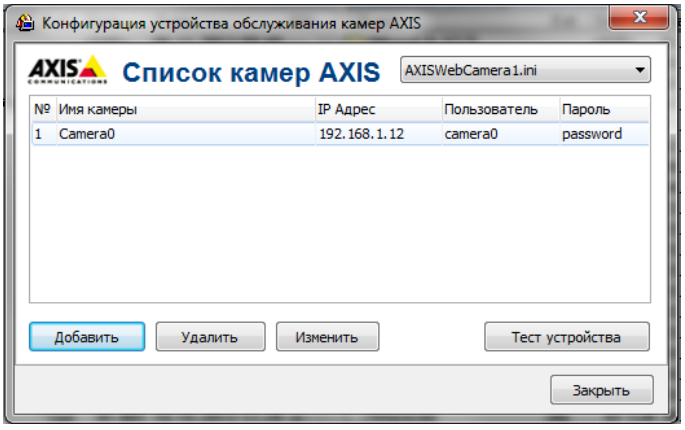
Обеспечивает ввод изображений с устройства USB BOX. Данное устройство позволяет вводить изображения с 4 камер в режиме переключения каналов.

AXISWebCameraN.DLL
Обеспечивает ввод изображений с IP камер высокого разрешения, разработанных фирмой AXIS, которые создают поток данных в формате MJPG. Система позволяет работать с любым количеством такими устройств. Для увеличения числа доступных устройств необходимо скопировать любую библиотеку AXISWebCameraN.dll в файл AXISWebCameraM.dll. Например: AXISWebCamera0.dll в файл AXISWebCamera4.dll, расширив таким образом число аудио устройств до 5. Данные о параметрах камеры храняться в соответствующих файлах инициализации с именами: AXISWebCameraN.ini

[Camera0]
Name=Без имени
IPAddress=192.168.1.10
UserName=source0
Password=password0
…..
[CameraN]
Name=Без имени
IPAddress=192.168.1.11
UserName=sourceN
Password=password

Каждая управляющая библиотека может обслуживать несколько IP камер.

Для удобства пользователя в поставке системы имеется приложение AXISConfig.exe.



Данное приложение позволяет сформировать файл инициализации для каждого устройства AXISWebCameraN и протестировать правильность установленных параметров.

10. Описание файла MEGALIB1.INI

В состав библиотеки может входить файл MegaLib1.INI, который предназначен для инициализации исходных параметров распознавания. Данные параметры загружаются при выполнении функций **CFOpen** и **TFOpen** и устанавливаются для всех каналов обработки. После инициализации распознавания данные параметры могут быть изменены для каждого канала при помощи следующих функций:

CFSetRecognitionParams
CFCHSetRecognitionParams
TFSetRecognitionParams
TFCHSetRecognitionParams

Для распознавания автомобильных номеров:
[CarPlate]
DisableSize=0
При работе системы в поле зрения камеры попадают надписи или структуры, которые не являются номерами. Для отсечения подобных надписей в программе устанавливаются предельные значения высоты символов, которые считаются недопустимыми. Для высоты символа в 10 пикселей границы составляют от 6 до 15, для символов в 15 пикселей – от 10 до 20 и для символов в 20 пикселей – от 15 до 25. Если данный параметр устанавливается в 1, данные ограничения запрещаются.

DisableWeght=0
В процессе определения номера формируется параметр вероятности распознавания. Данный параметр может принимать значение от 0 до 100. . Если данный флаг установлен в 0, данные с вероятностью меньше, либо равно величине **WeightTreshold** блокируются системой и не передаются на выход.

DisableGipo=0
В процессе определения номера формируется гипотеза, которая соответствует типу номерного знака. При невозможности определения типа данные блокируются системой и не передаются на выход. Если данный параметр устанавливается в 1, данные ограничения запрещаются.

EnableCheckLost=0
В процессе определения номера возможно пропадание зоны номера на короткий промежуток времени, что вызывает выработку признака окончания распознавания. Данный параметр предотвращает повторное определение одного и того же номера. Распознанный номер хранится в списке распознанных номеров около 2.5 секунд и блокирует повторное распознавание. Если данный параметр устанавливается в 1, выполнение данных действий разрешено.

ErrorRecTime=N
Данный параметр определяет количество коррекций номерного знака при удачно определенной зоне и неудачно распознанного номера. Число N определяет количество неудачных циклов распознавания, после которых срабатывает признак окончания распознавания.

WeightThreshold=85
В процессе определения номера вагонов формируется параметр вероятности распознавания. Данный параметр может принимать значение 0-100, при котором распознавание происходит если вероятность выше заданной величины.

Для распознавания железнодорожных номеров:
[TrnPlate]
DisableSize=0
При работе системы в поле зрения камеры попадают надписи или структуры, которые не являются номерами. Для отсечения подобных надписей в программе устанавливаются предельные значения высоты символов, которые считаются недопустимыми. Для высоты символа в 10 пикселей границы составляют от 6 до 15, для символов в 15 пикселей – от 10 до 20 и для символов в 20 пикселей – от 15 до 25. Если данный параметр устанавливается в 1, данные ограничения запрещаются.

DisableWeght=0
В процессе определения номера формируется параметр вероятности распознавания. Данный параметр может принимать значение от 0 до 100. . Если данный флаг установлен в 0, данные с вероятностью меньше, либо равно величине **WeightTreshold** блокируются системой и не передаются на выход.

DisableGipo=0
В процессе определения номера формируется гипотеза, которая соответствует типу номерного знака. При невозможности определения типа данные блокируются системой и не передаются на выход. Если данный параметр устанавливается в 1, данные ограничения запрещаются.

WeightThreshold=85
В процессе определения номера вагонов формируется параметр вероятности распознавания. Данный параметр может принимать значение 0-100, при котором распознавание происходит если вероятность выше заданной величины.

11. Что нового в библиотеке MegaLibV1

06.11.2013

1. Ведены новые функции инициализации библиотеки и управления ресурсами в многозадачных приложениях:
MPCreateEx
MPResource

2. Добавлено новое устройство при вызове функции IDAdd – NULL устройство для инициализации всех каналов доступных библиотеке после вызова процедур инициализайии.

3. Добавлены новые управляющие библиотеки для обслуживания IP самер фирмы AXIS:
AXISWebCameraN.dll

4. Для распознавания номеров автомобилей введен новый код региона 7XX.

26.02.2012

1. Ведены новые функции управления устройствами ввода аудио информации:
IDGetMaxAudioChannels
IDGetChannelAudioProperty
IDGetAudioBuffer
IDInputAudio
IDGetAudioProperty
IDSetAudioProperty
CHGetChannelAudioProperty
CHGetAudioBuffer
CHInputAudio
CHStartAudio
CHStopAudio
CHGetAudioDevice
CHGetAudioProperty
CHSetAudioProperty
CHGetAudioName

2. Ведены новые функции управления устройствами ввода/вывода цифровых сигналов:
IDGetMaxInLines
IDGetMaxOutLines
IDGetInLine
IDSetOutLine
CHGetOutLineDevice
CHGetOutLineName
CHGetInLineDevice
CHGetInLineName

3. Ведены новые функции установки параметров распознавания для исключения использования файла MEGALIB1.INI:

CFSetRecognitionParams
CFCHSetRecognitionParams
TFSetRecognitionParams
TFCHSetRecognitionParams

05.04.2010

1. Ведены новые функции управления свойствами устройств ввода, которые обеспечивают максимальное использование их возможностей:
IDGetCaptureProperty
IDSetCaptureProperty
CHGetCaptureProperty
CHSetCaptureProperty

2. Добавлена возможность обработки изображений в формате YUY2 для экономии времени на конвертации.

3. Дополнено количество процедур преобразования форматов изображений:
YUY2toRGB
UYVYtoRGB
UYVYtoGRAY
YUY2toGRAY
RGBtoGRAY
GRAYtoUYVY
GRAYtoYUY2
GRAYHREDUCE
UYVYHREDUCE
YUY2HREDUCE
RGBHREDUCE

4. В качестве элемента защиты добавлено более быстродействующее и компактное устройство защиты **SenseLockEL.**

5. Система адаптирована к устройствам ввода высокого разрешения.

01.09.2009

1. В функциях установки исходных значений параметров канала для определения номерных знаков автомобилей и номеров вагонов CFRestart, TFRestart, CFCHRestart, TFCHRestart в структуре **NUMINITPARAMS** имеются следующие изменения:

- Параметры **ZoneThreshold, Thhory, Thhorx, Thbin, Min Pipe Size, Max Pipe Size** не используются и вычесляются автоматически;
- В параметре **ZoneMode** введены дополнительные флаги определения размера зоны
  ZONE_MODE_96x24
  ZONE_MODE_128x32
  ZONE_MODE_160x40
  ZONE_MODE_192x48
  ZONE_MODE_256x64

ZONE_MODE_320x80
ZONE_MODE_384x96
ZONE_MODE_48x48 (только для определения автомобильных номеров)
ZONE_MODE_64x64 (только для определения автомобильных номеров)
ZONE_MODE_80x80 (только для определения автомобильных номеров)
ZONE_MODE_96x96 (только для определения автомобильных номеров)
ZONE_MODE_128x128 (только для определения автомобильных номеров)
ZONE_MODE_160x160 (только для определения автомобильных номеров)
ZONE_MODE_192x192 (только для определения автомобильных номеров)

2. В функциях установки исходных значений параметров канала для определения номерных знаков автомобилей и номеров вагонов **CFGetResult**, **TFGetResult**, **TFCHGetResult**, **CFCHGetResult** в структуре **PLATERESULT** параметр **Resolution** возвращает следующие значения:
ZONE_RESOLUTION_96x24 0
ZONE_RESOLUTION_128x32 1
ZONE_RESOLUTION_160x40 3
ZONE_RESOLUTION_192x48 2
ZONE_RESOLUTION_256x64 4
ZONE_RESOLUTION_320x80 5
ZONE_RESOLUTION_384x96 6
ZONE_RESOLUTION_48x48 10 (только для определения автомобильных номеров)
ZONE_RESOLUTION_64x64 11 (только для определения автомобильных номеров)
ZONE_RESOLUTION_80x80 12 (только для определения автомобильных номеров)
ZONE_RESOLUTION_96x96 13 (только для определения автомобильных номеров)
ZONE_RESOLUTION_128x128 14 (только для определения автомобильных номеров)
ZONE_RESOLUTION_160x160 15 (только для определения автомобильных номеров)
ZONE_RESOLUTION_192x192 16 (только для определения автомобильных номеров)

3. Удалены из INI файла секции, определяющие КРР значения Российских номерных знаков.

4. Применены новые методы выделения и распознавания символов, что позволило увеличить процент правильного определения номерных знаков.