

ООО «Спецтелетехника»
www.sptt.ru

Библиотека функций STT_CAM

Руководство программиста

Версия 1.36

Февраль 2008

Оглавление.

1. Назначение и системные требования.....	4
2. Протокол обмена с камерами и структура библиотеки.....	4
3. Описание функций библиотеки.....	4
3.1 Функции инициализации.	6
3.1.1 sdu_open_interface	6
3.1.2 sdu_open_camera	7
3.1.3. sdu_close_interface	8
3.1.4 sdu_get_version	8
3.1.5 sdu_get_serialNumber	8
3.1.6 sdu_get_DeviceCount	8
3.2 Функции управления камерами.....	9
3.2.1. sdu_set_exp	9
3.2.2. sdu_set_master_gain	10
3.2.3. sdu_set_r_ossfet	11
3.2.4. sdu_set_g_ossfet	11
3.2.5. sdu_set_trigmode	12
3.2.6. sdu_set_binning	13
3.2.7. sdu_set_encoding	14
3.2.8. sdu_set_period	15
3.2.9. sdu_set_roi_org	15
3.2.10. sdu_set_roi_size	16
3.2.11. sdu_set_draft	17
3.2.12. sdu_get_status	18
3.2.13. sdu_get_exp	19
3.2.14. sdu_get_min_exp	19
3.2.15. sdu_get_max_exp	19
3.2.16. sdu_get_master_gai	20
3.2.17. sdu_get_max_gain	20
3.2.18. sdu_get_opt_gain	21
3.2.19. sdu_get_r_offset	21
3.2.20. sdu_get_g_offset	22
3.2.21. sdu_get_trigmode	22
3.2.22. sdu_get_period	22

3.2.23. sdu_get_min_period.....	23
3.2.24. sdu_get_binning.....	23
3.2.25. sdu_get_encoding.....	23
3.2.26. sdu_get_roi_org.....	24
3.2.27. sdu_get_roi_size.....	24
3.2.28. sdu_get_draft.....	24
3.2.29. sdu_get_sensor_width.....	25
3.2.30. sdu_get_sensor_height.....	25
3.2.31. sdu_get_color_id.....	25
3.2.32. sdu_get_fifo_count.....	26
3.2.33. sdu_fifo_init.....	26
3.2.34. sdu_cam_start.....	27
3.2.35. sdu_cam_stop.....	28
3.2.36. sdu_dac_use.....	29
3.2.37. sdu_dac_load_array.....	29
3.2.38. sdu_dac_set_arraystart.....	29
3.3 Функции передачи данных.	30
3.3.1. sdu_read_data.....	30
3.3.2. sdu_open_file_rww.....	30
3.3.3. sdu_file_rww_get_first_frame.....	31
3.3.4. sdu_file_rww_get_next_frame.....	31
3.3.5. sdu_file_rww_get_frame.....	32
3.3.6. sdu_file_rww_write_frame.....	32
3.3.7. sdu_close_file_rww.....	32
3.4. Функции обработки изображения.....	33
3.4.1. sdu_demosaic_12to8.....	33
3.4.2. sdu_demosaic_12to16.....	35
3.4.3. sdu_demosaic_8to8.....	37
3.4.4. sdu_demosaic_8to16.....	38
3.4.5. sdu_bw_convert_12to16.....	39
3.4.6. sdu_bw_convert_12to8.....	40
3.4.7. sdu_bw_convert_8to8.....	41
3.4.8. sdu_bw_convert_8to16.....	41
3.4.9. sdu_make_gamma_lut_12to16.....	42
3.4.10. sdu_make_gamma_lut_12to8.....	42
3.4.11. sdu_make_gamma_lut_8to8.....	42
4. История изменений.....	43

1. Назначение и системные требования.

Библиотека функций STT_CAM предназначена для работы с цифровыми камерами семейств SDU и CSDU производства ООО «Спецтелетехника». Библиотека реализована в виде DLL и поддерживает работу с камерами под управлением ОС Windows 2000, Windows XP и Windows Vista. Для работы с камерами необходим также драйвер обмена по USB CyUsb.sys. Функции библиотеки могут вызываться из программ пользователя, написанных на Microsoft Visual C++ версий 6.0 и 7.0.

2. Протокол обмена с камерами и структура библиотеки.

При подключении камеры к компьютеру система автоматически распознает ее тип (при условии, что драйверы уже установлены) и регистрирует ее в соответствии с протоколом Plug and Play. При этом в камеру автоматически загружаются требуемые конфигурационные данные. Передача в камеру управляющих команд происходит следующим образом:

- приложение пользователя вызывает одну из управляющих функций библиотеки
- DLL библиотеки формирует необходимую управляющую последовательность и передает ее в камеру по интерфейсу USB
- камера принимает управляющую посылку и выполняет соответствующую команду
- камера передает в компьютер по USB информацию о выполнении команды и своем состоянии.
- DLL библиотеки принимает ответ камеры и возвращает информацию о результате выполнения команды вызывающему приложению

Функции библиотеки можно разделить на 4 группы:

- функции инициализации интерфейса
- функции управления камерой
- функции передачи данных
- функции обработки изображения

Все функции библиотеки являются едиными для всех типов камер, однако допустимые значения управляющих параметров для разных камер различны. В большинстве случаев эти допустимые значения можно определить, вызывая соответствующие функции запроса параметров (см. п. 3.1.12 - 3.1.31)

За исключением функции **sdu_open_interface**, все функции имеют тип DWORD и возвращают следующие значения, приведенные в таблице:

Возвращаемое значение	Интерпретация
SDU_SUCCESS	Успешное завершение
SDU_TIMEOUT	Таймаут при обмене с камерой
SDU_PARAM_ERR	Недопустимое значение параметра
SDU_PARAM_ADJUSTED	Параметр был скорректирован
SDU_TRIG_ERR	Ошибка запуска/обрыва экспозиции
SDU_CAMERA_NOT_OPENED	Камера не была открыта для доступа
SDU_DATA_ERROR	Ошибка при передаче данных по USB

Описания всех констант и символических имен, используемых при работе с библиотекой, а также прототипы функций библиотеки находятся в файле stt_cam.h

3. Описание функций библиотеки

3.1 Функции инициализации

3.1.1 `CSDUCameraDevice * sdu_open_interface(int *cam_count)`

Функция создает `CSDUCameraDevice` объект и возвращает указатель на него. Почти все остальные функции (за исключением некоторых вспомогательных) получают этот указатель в качестве параметра. Функция также инициализирует USB камеры, проверяет сколько камер подключено к системе и возвращает их количество в `cam_count`. Пример реализации приведен ниже:

Допустим имеется некий класс для работы с камерой `CSomeClass`.

```
// SomeClass.h
class CSDUCameraDevice;

class CSomeClass
{
    CSDUCameraDevice * m_pSDUCameraDevice;
    ....
};

// SomeClass.cpp

// constructor
CSomeClass::CSomeClass(...)
{
    int nCamCount;
    m_pSDUCameraDevice = sdu_open_interface(&nCamCount);
    if (!m_pSDUCameraDevice)
    {
        // handle the error here
    }
}

// destructor
CSomeClass::~~CSomeClass()
{
    sdu_close_interface(m_pSDUCameraDevice);
}
```

Такой подход позволяет создавать несколько объектов класса `CSomeClass`, при этом каждый из них будет работать со своим `CSDUCameraDevice` объектом.

3.1.2 `sdu_open_camera(CSDUCameraDevice * pSDUCameraDevice, DWORD cam_num, char * cam_name)`

Функция открывает соответствующую `cam_num` USB камеру и записывает в массив `cam_name` название камеры. Нумерация камер начинается с 0. Если подключена одна камера и `cam_count=1` то камеру следует открывать, вызывая функцию с параметром `cam_num = 0`. Для второй камеры `cam_num = 1` и т.д.

Под массив `cam_name` в приложении пользователя должно быть выделено не менее 64 байт. Название камеры представляет собой строку символов и состоит из серийного номера камеры (4 символа в коде ASCII) и собственно названия разделенных двумя пробелами. В случае если устройства с номером соответствующим `cam_num` в системе нет, функция возвращает значение `SDU_TIMEOUT`. Серийный номер камеры может служить ее уникальным идентификатором (серийные номера не повторяются).

Функция должна обязательно быть вызвана перед началом работы с соответствующей камерой. Это может быть сделано однократно, но следует учитывать, что при открытии другой камеры (если их несколько в системе) предыдущая автоматически закрывается и становится недоступной.

3.1.3 `sdu_close_interface(CSDUCameraDevice * pSDUCameraDevice)`

Функция закрывает все подключенные к системе камеры и уничтожает объект **CSDUCameraDevice**. Необходимо вызывать при окончании работы (например в деструкторе), а также когда камеру отключают или подключают при работающем приложении пользователя.

Функцию следует вызывать или перед изменением конфигурации системы камер или после изменения, но до начала работы с любой камерой или до последующего изменения конфигурации (например при отключении камеры функция обязательно должна быть вызвана до последующего подключения камеры). Для отслеживания изменения конфигурации системы камер можно использовать системное событие **WN_DEVICECHANGE**.

После вызова этой функции, для продолжения работы, система камер должна быть повторно проинициализирована с помощью функции **sdu_open_interface**.

3.1.4 `sdu_get_version(char * version_string)`

Функция возвращает в **version_string** номер версии библиотеки в коде ASCII.

3.1.5 `sdu_get_serialNumber(CSDUCameraDevice * pSDUCameraDevice, char * pSerialNumber)`

Функция возвращает уникальный серийный номер камеры.

3.1.6 `sdu_get_DeviceCount (CSDUCameraDevice * pSDUCameraDevice, DWORD * pdwDeviceCount)`

Функция возвращает число подсоединенных камер.

3.2 Функции управления камерами

Функции служат для подачи в камеру управляющих команд и получения информации о состоянии камеры.

Для работы с камерой она должна быть открыта с помощью функции **sdu_open_camera**.

3.2.1 **sdu_set_exp(CSDUCameraDevice * pSDUCameraDevice, DWORD *exposition)**

Функция устанавливает величину экспозиции.

Параметр:

exposition - значение экспозиции в микросекундах.

Величина экспозиции может находиться в пределах от **min_exp** до **max_exp**. Величины **min_exp** и **max_exp**, свои для каждого типа камер, возвращаются камерой при вызове функций **sdu_get_min_exp** и **sdu_get_max_exp** соответственно.

Если устанавливаемое значение экспозиции превышает установленное значение периода запуска, то камера автоматически корректирует значение периода.

Если устанавливаемое значение экспозиции меньше **min_exp** либо больше **max_exp** то функция возвратит значение **SDU_PARAM_ERROR**.

В случае если камера не может установить точное требуемое значение экспозиции (такое возможно, если экспозиция превышает 65536 мкс), камера устанавливает экспозицию максимально близкую к требуемой. Погрешность установки не превышает 0.025%. Функция в этом случае возвращает значение **SDU_PARAM_ADJUSTED**. Реально установленное в этом случае значение экспозиции можно узнать с помощью функции **sdu_get_exp**.

3.2.2. `sdu_set_master_gain(CSDUCameraDevice * pSDUCameraDevice, DWORD master_gain)`

Функция устанавливает величину усиления сигнала с выхода ПЗС перед оцифровкой

Параметр:

master_gain – значение усиления в относительных единицах

Для черно-белых камер это единственная функция управления усилением сигнала с выхода ПЗС матрицы. Для цветных камер эта функция управляет усилением для синих (B) пикселей. Обычно сигнал этих пикселей самый слабый, поэтому это усиление максимально. Усиление для красных (R) и зеленых (G) пикселей вычисляется с учетом величин коррекции усиления **r_offset** и **g_offset** (см. ниже).

Величина реального усиления в децибелах зависит от значения **master_gain** следующим образом: изменение **master_gain** на 128 приводит к соответствующему изменению реального усиления на 6 дБ, т.е. в 2 раза.

Минимальное значение устанавливаемого **master_gain** равно 0, но реально устанавливать значение меньше некоторой оптимальной величины не рекомендуется, т.к. это приводит к снижению динамического диапазона камеры по причине неполного использования диапазона входных амплитуд АЦП. Это оптимальное значение обычно свое для каждого типа камеры и еще зависит от величины биннинга. Его значение может быть получено с помощью функции **sdu_get_opt_gain**.

Максимальное значение устанавливаемого **master_gain** может быть получено с помощью функции **sdu_get_max_gain**.

При попытке установить **master_gain > max_gain** функция возвратит значение **SDU_PARAM_ERROR**.

3.2.3. `sdu_set_r_ossfet(CSDUCameraDevice * pSDUCameraDevice, DWORD r_gain_offset)`

Функция устанавливает величину коррекции усиления для красных (R) пикселей.

Параметр:

r_gain_offset – значение величины коррекции в относительных единицах

Значение усиления сигнала с выхода ПЗС матрицы для красных пикселей будет установлено равным

$$\text{master_gain} - \text{r_gain_offset}$$

При попытке установить `r_gain_offset > master_gain` функция возвратит значение **SDU_PARAM_ERROR**.

Функция работает только для цветных камер. При попытке вызова функции для черно-белых камер с параметром отличным от 0 функция возвратит значение **SDU_PARAM_ERROR**.

3.2.4. `sdu_set_g_ossfet(CSDUCameraDevice * pSDUCameraDevice, DWORD g_gain_offset)`

Функция устанавливает величину коррекции усиления для зеленых (G) пикселей.

Параметр:

g_gain_offset – значение величины коррекции в относительных единицах

Значение усиления сигнала с выхода ПЗС матрицы для зеленых пикселей будет установлено равным

$$\text{master_gain} - \text{g_gain_offset}$$

При попытке установить `g_gain_offset > master_gain` функция возвратит значение **SDU_PARAM_ERROR**.

Функция работает только для цветных камер. При попытке вызова функции для черно-белых камер с параметром отличным от 0 функция возвратит значение **SDU_PARAM_ERROR**.

3.2.5. `sdu_set_trigmode(CSDUCameraDevice * pSDUCameraDevice, DWORD trigmode)`

Функция устанавливает режим запуска съемки.

Параметр:

trigmode - режим запуска съемки

допустимые значения:

SDU_TRIGMODE_CONTIN - непрерывная съемка с заданным периодом

SDU_TRIGMODE_EXTERNAL - запуск по внешнему синхроимпульсу

SDU_TRIGMODE_SOFTSTART - программный запуск

SDU_TRIGMODE_BULB - ведомый режим с внешним запуском

SDU_TRIGMODE_EXTERNAL_SINGLE - запуск по внешнему синхроимпульсу
однократный

SDU_TRIGMODE_BULB_SINGLE - ведомый режим с внешним запуском
однократный

Ко всем указанным режимам можно применять флаг (логическое ИЛИ)
SDU_TRIGMODE_DOUBLE_FLAG – при этом камера переходит в режим
“Двойного чтения” – при каждом запуске считывается два кадра(один с
указанной экспозицией, второй – с фиксированной). Если камера
поддерживает данный режим, в статусе состояния будет установлен бит
SDU_STAT_DOUBLE_READ.

Более подробная информация о различных режимах запуска приведена в
описании камер.

3.2.6. `sdu_set_binning(CSDUCameraDevice * pSDUCameraDevice, DWORD binning)`

Функция устанавливает значение коэффициента биннинга (сложения пикселей)

Параметр:

binning – значение коэффициента биннинга

допустимые значения

<code>SDU_BINNING_1x1</code>	-	1x1 (нет биннинга)
<code>SDU_BINNING_2x2</code>	-	2x2
<code>SDU_BINNING_3x3</code>	-	3x3
<code>SDU_BINNING_4x4</code>	-	4x4

Функция работает только для черно-белых камер. При попытке вызова ее для цветных камер с параметром отличным от `SDU_BINNING_1x1` функция возвращает значение `SDU_PARAM_ERROR`.

Коэффициент биннинга 3x3 доступен только для камер с прогрессивной разверткой. При попытке вызова функции для чересстрочных камер с параметром равным `SDU_BINNING_3x3` функция возвращает значение `SDU_PARAM_ERROR`.

При вызове функции также происходит установка параметров зоны считывания (ROI) таким образом, что она соответствует кадру максимального размера:

```
roi_org_h = 0  
roi_org_v = 0  
roi_size_h = ccd_width/(binning+1)  
roi_size_v = ccd_height/(binning +1)
```

3.2.7. `sdu_set_encoding(CSDUCameraDevice * pSDUCameraDevice, DWORD encoding)`

Функция устанавливает формат запаковки данных при передаче камерой оцифрованного изображения.

Параметр:

encoding - формат запаковки данных

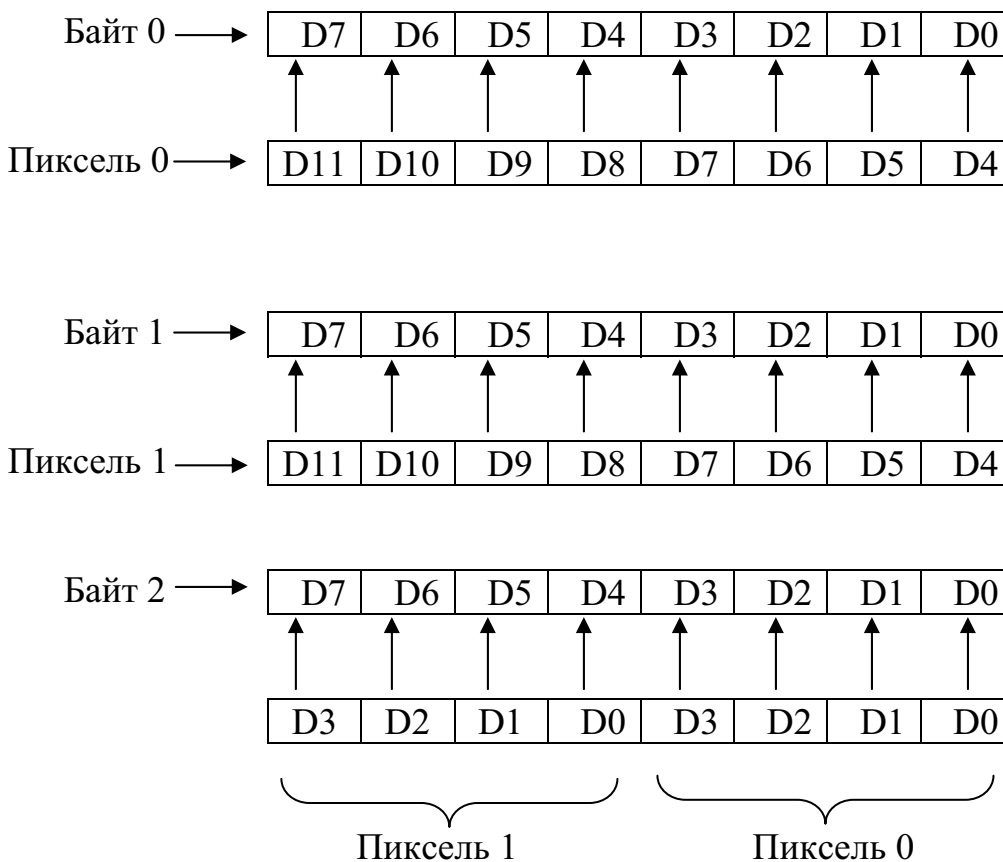
допустимые значения:

SDU_ENCODING_8BPP

SDU_ENCODING_12BPP

При установке формата 8 бит/пиксель (8 bpp) количество байт в одном кадре соответствует количеству пикселей в зоне считывания (ROI), т.е. каждый пиксель представлен одним байтом.

При установке формата 12 bpp количество байт в одном кадре превышает количество пикселей в зоне ROI в 1.5 раза, при этом данные запакованы таким образом, что каждые два пикселя представлены тремя байтами:



3.2.8. `sdu_set_period(CSDUCameraDevice * pSDUCameraDevice, DWORD period)`

Функция устанавливает величину периода запуска съемки при работе камеры в непрерывном режиме.

Параметр:

period - значение периода в микросекундах

Пределы установки периода - от **min_period** до 0x3FFFFFFF, что соответствует максимальному периоду около 18 минут. Величина **min_period** зависит от типа камеры, размера зоны считывания и от биннинга. Её значение автоматически вычисляется контроллером камеры и возвращается при вызове функции **sdu_get_min_period**. При попытке установить значение периода меньше чем **min_period** функция возвратит значение **SDU_PARAM_ERROR**.

Если вследствие изменения приложением пользователя параметров биннинга или зоны считывания (ROI) новое, вычисленное камерой, значение **min_period** окажется больше уже установленного ранее значения периода, то период будет автоматически скорректирован и установлен равным новому значению **min_period**.

3.2.9. `sdu_set_roi_org(CSDUCameraDevice * pSDUCameraDevice, DWORD roi_org_h, DWORD roi_org_v)`

Функция устанавливает положение левого верхнего угла зоны считывания (ROI - region of interest).

Параметры:

roi_org_h - положение начала зоны ROI по горизонтали

roi_org_v - положение начала зоны ROI по вертикали

Параметры **roi_org_h** и **roi_org_v** могут принимать только четные значения. Устанавливаемые значения должны быть такими, чтобы зона считывания не выходила за пределы кадра максимального размера. Контроллер камеры сам следит за этим и игнорирует неправильно заданные значения. Функция в этом случае возвращает значение **SDU_PARAM_ERROR**. Более подробно об этом см. в описании функции **sdu_set_roi_size**.

3.2.10. `sdu_set_roi_size(CSDUCameraDevice *
pSDUCameraDevice, DWORD roi_size_h, DWORD
roi_size_v)`

Функция устанавливает размер зоны считывания (ROI - region of interest).

Параметры:

`roi_size_h` - размер зоны ROI по горизонтали

`roi_size_v` - размер зоны ROI по вертикали

Параметр `roi_size_h` может принимать только значения кратные 4.

Параметр `roi_size_v` может принимать только четные значения.

Устанавливаемые значения должны быть такими, чтобы зона считывания не выходила за пределы кадра максимального размера, т.е. должны выполняться следующие условия:

$$\text{roi_org_h} + \text{roi_size_h} \leq \text{ccd_width} / (\text{binning} + 1)$$
$$\text{roi_org_v} + \text{roi_size_v} \leq \text{ccd_height} / (\text{binning} + 1)$$

Контроллер камеры сам следит за выполнением всех этих условий и игнорирует неправильно заданные значения. Функция в этом случае возвращает значение `SDU_PARAM_ERROR`.

3.2.11. `sdu_set_draft(CSDUCameraDevice * pSDUCameraDevice, DWORD draft)`

Функция служит для переключения камеры в режим съемки с прореживанием изображения. Такой режим поддерживается только камерами с размером изображения больше 2 мегапикселей - SDU-252С, SDU-MT2М, SDU-MT3М. Этот режим удобен для реализации быстрого просмотра изображения в реальном времени.

Параметр:

draft - режим прореживания

Может принимать значения:

SDU_DRAFT_ON - прореживание включено

SDU_DRAFT_OFF - прореживание выключено

При попытке вызова функции с параметром равным **SDU_DRAFT_ON** для камер, не поддерживающих прореживание, будет возвращено значение **SDU_PARAM_ERR**.

При включении прореживания камера устанавливает размер изображения, уменьшенный в N раз (N = 4 либо 6 для разных камер - см. описания камер) Значения **ROI**, **sensor_width** и **sensor_height**, возвращаемые при вызове соответствующих функций запроса параметров, отражают это изменение. Статус режима прореживания отражается соответствующим битом в состоянии камеры **status**.

3.2.12. `sdu_get_status(CSDUCameraDevice * pSDUCameraDevice, DWORD *status)`

Функция служит для получения из камеры слова состояния - статуса.

Параметр:

***status** - указатель на переменную, в которую будет передан статус камеры. Статус разбит на битовые поля, назначение которых приведено в таблице ниже

Bit	Name	Значение
D0	SDU_STAT_RNS	1 - идет непрерывная съемка, 0 - стоп
D1	SDU_STAT_EXP	1 - идет экспозиция
D2	SDU_STAT_BUSY	1 - камера занята (идет экспозиция или запись в FIFO)
D3	SDU_STAT_FIFOEMPTY	1 - в FIFO камеры нет данных
D4	SDU_STAT_FIFOFULL	1 - FIFO камеры переполнено
D5	RESERVED	Резерв (равен 0)
D6	SDU_STAT_DOUBLE_READ	Режим двойного чтения
D7	SDU_STAT_TRIG_ERR	1 - ошибка запуска или обрыва экспозиции
D8	SDU_STAT_TRIGMODE0	Режим запуска экспозиции
D9	SDU_STAT_TRIGMODE1	
D10	SDU_STAT_ENCODING	0 - 8 бит/пиксель , 1 - 12 бит/пиксель
D11	SDU_STAT_OVL_EXP	Режим экспозиции: 0 - SEQ, 1 - OVL (только для чересстрочных камер, для прогрессивных - всегда 0)
D12	SDU_STAT_INTERLACE	Режим считывания: 0 - PROGRESSIVE, 1 - INTL (только для чересстрочных камер, для прогрессивных - всегда 0)
D13	SDU_STAT_DRAFT	Режим прореживания: 0 - выключен, 1 - включен (только для камер, поддерживающих прореживание, для остальных - всегда 0)
D14	SDU_STAT_SLEEP	1 - Sleep , 0 - Work
D15	SDU_STAT_USBSPEED	1 - High Speed (480 Mbps), 0 - Full Speed (12 Mbps)

Все оставшиеся биты переменной **status** (D16-D31) зарезервированы и равны 0.

3.2.13. **sdu_get_exp(CSDUCameraDevice *
pSDUCameraDevice, DWORD *current_exp)**

Функция служит для считывания из камеры установленного значения экспозиции

Параметр:

***current_exp** - указатель на переменную, в которую будет передано текущее значение экспозиции в микросекундах.

3.2.14. **sdu_get_min_exp(CSDUCameraDevice *
pSDUCameraDevice, DWORD *min_exp)**

Функция служит для считывания из камеры минимального допустимого значения экспозиции

Параметр:

***min_exp** - указатель на переменную, в которую будет передано минимальное значение экспозиции в микросекундах.

3.2.15. **sdu_get_max_exp(CSDUCameraDevice *
pSDUCameraDevice, DWORD *max_exp)**

Функция служит для считывания из камеры максимального допустимого значения экспозиции

Параметр:

***max_exp** - указатель на переменную, в которую будет передано максимальное значение экспозиции в микросекундах.

3.2.16. **sdu_get_master_gain(CSDUCameraDevice *
pSDUCameraDevice, DWORD *current_gain)**

Функция служит для считывания из камеры установленного значения усиления

Параметр:

***current_gain** - указатель на переменную, в которую будет передано текущее значение усиления **master_gain**.

3.2.17. **sdu_get_max_gain(CSDUCameraDevice *
pSDUCameraDevice, DWORD *max_gain)**

Функция служит для считывания из камеры максимального допустимого значения усиления

Параметр:

***max_gain** - указатель на переменную, в которую будет передано максимальное значение усиления.

3.2.18. `sdu_get_opt_gain(CSDUCameraDevice * pSDUCameraDevice, DWORD *opt_gain)`

Функция служит для считывания из камеры оптимального (минимального рекомендуемого) значения усиления

Параметр:

***opt_gain** - указатель на переменную, в которую будет передано оптимальное значение усиления.

При установке усиления равным **opt_gain** камера обеспечивает наилучший динамический диапазон при минимуме шумов.

Для цветных камер усиление для красных и зеленых пикселей вычисляется с учетом величины коррекции:

$$\begin{aligned} r_gain &= master_gain - r_gain_offset \\ g_gain &= master_gain - g_gain_offset \end{aligned}$$

Для получения максимального динамического диапазона следует устанавливать значения такими, чтобы вычисленные по приведенным выше формулам значения усиления не были меньше **opt_gain**.

Значение **opt_gain** зависит от типа камеры и от установленного режима биннинга. При изменении биннинга возвращаемое функцией значение **opt_gain** меняется соответственно.

3.2.19. `sdu_get_r_offset(CSDUCameraDevice * pSDUCameraDevice, DWORD *current_r_offset)`

Функция служит для считывания из камеры установленного значения коррекции усиления для красных пикселей

Параметр:

***current_r_offset** - указатель на переменную, в которую будет передано текущее значение коррекции.

3.2.20. **sdu_get_g_offset(CSDUCameraDevice *
pSDUCameraDevice, DWORD *current_g_offset)**

Функция служит для считывания из камеры установленного значения коррекции усиления для зеленых пикселей

Параметр:

***current_g_offset** - указатель на переменную, в которую будет передано текущее значение коррекции.

3.2.21. **sdu_get_trigmode (CSDUCameraDevice *
pSDUCameraDevice DWORD *current_trigmode)**

Функция служит для считывания из камеры установленного значения режима запуска экспозиции.

Параметр:

***current_trigmode** - указатель на переменную, в которую будет передано текущее значение режима запуска. Возвращаемые значения соответствуют устанавливаемым функцией **sdu_set_trigmode**.

Текущее значение режима запуска также отражается в соответствующих битах статуса камеры (см. п. 3.2.12).

3.2.22. **sdu_get_period(CSDUCameraDevice *
pSDUCameraDevice, DWORD *current_period)**

Функция служит для считывания из камеры установленного значения периода запуска

Параметр:

***current_period** - указатель на переменную, в которую будет передано текущее значение периода в микросекундах. Значение периода запуска может автоматически корректироваться камерой, при изменении величины экспозиции, биннинга и зоны считывания. Возвращаемое функцией значение **current_period** отражает эти изменения, т.е. соответствует реально установленному периоду.

3.2.23. **sdu_get_min_period(CSDUCameraDevice *
pSDUCameraDevice, DWORD *min_period)**

Функция служит для считывания из камеры минимального значения периода запуска

Параметр:

***min_period** - указатель на переменную, в которую будет передано текущее значение минимального периода в микросекундах. Значение минимального периода запуска автоматически корректируется камерой, при изменении величины экспозиции, биннинга и зоны считывания. Возвращаемое функцией значение **min_period** отражает эти изменения.

3.2.24. **sdu_get_binning(CSDUCameraDevice *
pSDUCameraDevice, DWORD *current_binning)**

Функция служит для считывания из камеры значения коэффициента биннинга.

Параметр:

***current_binning** - указатель на переменную, в которую будет передано текущее значение коэффициента биннинга. Возвращаемые значения соответствуют устанавливаемым функцией **sdu_set_binning**.

3.2.25. **sdu_get_encoding(CSDUCameraDevice *
pSDUCameraDevice, DWORD *current_encoding)**

Функция служит для считывания из камеры значения режима запаковки данных

Параметр:

***current_encoding** - указатель на переменную, в которую будет передано текущее значение режима запаковки.

3.2.26. **sdu_get_roi_org**(CSDUCameraDevice *
pSDUCameraDevice, DWORD *current_roi_org_h,
DWORD *current_roi_org_v)

Функция считывает из камеры координаты левого верхнего угла зоны считывания ROI.

Параметры:

***current_roi_org_h** -указатель на переменную , в которую будет передано положение начала зоны ROI по горизонтали

***current_roi_org_v** - указатель на переменную , в которую будет передано положение начала зоны ROI по вертикали

3.2.27. **sdu_get_roi_size**(CSDUCameraDevice *
pSDUCameraDevice, DWORD *current_roi_size_h,
DWORD *current_roi_size_v)

Функция считывает из камеры установленные размеры зоны считывания ROI.

Параметры:

***current_roi_size_h** -указатель на переменную , в которую будет передан размер зоны ROI по горизонтали

***current_roi_size_v** - указатель на переменную , в которую будет передан размер зоны ROI по вертикали

3.2.28. **sdu_get_draft**(CSDUCameraDevice *
pSDUCameraDevice, DWORD *real_draft)

Функция считывает из камеры установленное значение режима прореживания изображения. Для камер, не поддерживающих режим прореживания всегда возвращается **SDU_DRAFT_OFF** .

Параметр:

***real_draft** - указатель на переменную , в которую будет считан режим прореживания

Может принимать значения:

SDU_DRAFT_ON - прореживание включено

SDU_DRAFT_OFF - прореживание выключено

Статус режима прореживания также отражается соответствующим битом в состоянии камеры **status**.

3.2.29. **sdu_get_sensor_width(CSDUCameraDevice *
pSDUCameraDevice, DWORD *sensor_width)**

Функция считывает из камеры величину максимально возможного размера кадра по горизонтали, соответствующую количеству пикселей в строке матрицы сенсора.

Параметр:

***sensor_width** - указатель на переменную, в которую будет передано значение максимального размера кадра по горизонтали.

3.2.30. **sdu_get_sensor_height(CSDUCameraDevice *
pSDUCameraDevice, DWORD *sensor_height)**

Функция считывает из камеры величину максимально возможного размера кадра по вертикали, соответствующую количеству пикселей в столбце матрицы сенсора.

Параметр:

***sensor_width** - указатель на переменную, в которую будет передано значение максимального размера кадра по вертикали.

3.2.31. **sdu_get_color_id(CSDUCameraDevice *
pSDUCameraDevice, DWORD *color_id)**

Функция считывает из камеры параметр, определяющий тип мозаичных светофильтров и их расположение.

Параметр:

***color_id** - указатель на переменную, в которую будет передано значение

Для черно-белых камер значение **color_id = SDU_COLOR_ID_BW**

Для цветных камер соответствие последовательности светофильтров и значения **color_id** приведено в таблице:

color_id	SDU_COLOR_ID_RGGB	SDU_COLOR_ID_GRBG
Нечетные строки	R-G-R-G...	G-R-G-R...
Четные строки	G-B-G-B...	B-G-B-G...

3.2.32. `sdu_get_fifo_count(CSDUCameraDevice * pSDUCameraDevice, DWORD *fifo_count)`

Функция служит для считывания из камеры информации о количестве байт в буфере FIFO камеры.

Параметр:

***fifo_count** - указатель на переменную, в которую будет передано количество байт в FIFO.

Функция предназначена для отладочных целей и может неправильно отражать количество байт в FIFO, если она будет вызвана после начала передачи изображения из камеры до его полного завершения (такой случай возможен, например, при передаче данных в несколько приемов).

3.2.33. `sdu_fifo_init(CSDUCameraDevice * pSDUCameraDevice)`

Функция служит для инициализации (очистки) буферного ОЗУ камеры (FIFO). Параметров не имеет. В результате выполнения этой функции счетчик байтов **fifo_count** обнуляется.

Если при работе с камерой возникает переполнение буфера FIFO, то дальнейшее заполнение буфера прекращается и продолжить работу с камерой можно только после вызова функции **sdu_fifo_init**. При этом имеющиеся в FIFO достоверные кадры (все кроме последнего, на котором возникло переполнение) можно считать из камеры до ее вызова.

Желательно также вызывать эту функцию перед началом цикла съемки (вызовом функции **sdu_camera_start**), однако следует учитывать, что после ее вызова имеющиеся в FIFO камеры несчитанные кадры будут потеряны.

3.2.34. `sdu_cam_start(CSDUCameraDevice * pSDUCameraDevice)`

Функция служит для запуска съемки. Параметров не имеет. Работает только в режимах непрерывной съемки и программного запуска. При попытке вызова функции, когда установлены другие режимы съемки (внешний запуск или ведомый режим), команда игнорируется и устанавливается бит **SDU_STAT_TRIG_ERR** в **status**. Функция при этом возвращает значение **SDU_TRIG_ERR**.

Если установлен режим непрерывной съемки, то вызов этой функции разрешает периодический запуск экспозиции с периодом установленным функцией **sdu_set_period**. Сразу после вызова функции камера обрабатывает соответствующий периоду интервал времени, запускает экспозицию, по окончании экспозиции оцифровывает кадр и записывает его во внутренний буфер FIFO. Приложение пользователя должно за время равное периоду успевать считывать из камеры соответствующее размеру изображения количество байт, иначе возникнет переполнение буфера FIFO. Процесс повторяется периодически до подачи команды **sdu_cam_stop** или до переполнения буфера. В последнем случае заполнение буфера прекращается и устанавливается признак **SDU_STAT_FIFOFULL** в **status**.

Если установлен режим программного запуска то вызов этой функции производит однократный запуск экспозиции, после окончания которой камера оцифровывает кадр и записывает его во внутренний буфер FIFO. Изображение может быть считано из буфера приложением пользователя сразу же по окончании цикла (точнее параллельно с процессом записи кадра в FIFO), либо позже. Можно осуществить несколько циклов съемки, после чего считать все данные, но следует учитывать возможность переполнения буфера FIFO.

3.2.35. `sdu_cam_stop` (CSDUCameraDevice * pSDUCameraDevice)

Функция служит для остановки съемки. Параметров не имеет. Работает только в режимах непрерывной съемки и программного запуска. При попытке вызова функции, когда установлены другие режимы съемки (внешний запуск или ведомый режим), команда игнорируется и устанавливается бит **SDU_STAT_TRIG_ERR** в **status**. Функция при этом возвращает значение **SDU_TRIG_ERR**.

Если установлен режим непрерывной съемки, то вызов этой функции запрещает периодический запуск экспозиции. Сразу после вызова функции внутренний таймер камеры останавливается, при этом отработка текущего периода прерывается. Если в этот момент идет экспозиция или оцифровка данных эти процессы завершатся обычным образом (т.е. кадр будет отэкспонирован, оцифрован и записан в буфер FIFO) и камера перейдет в состояние ожидания.

Если установлен режим программного запуска то функция может использоваться для программного обрыва экспозиции. Вызов этой функции в то время когда идет экспозиция (бит **SDU_STAT_EXP** в **status** равен 1) приводит к обрыву экспозиции и запускает процесс оцифровки кадра и его записи в буфер FIFO. Если функция будет вызвана в тот момент, когда экспозиция уже завершилась обычным образом (т.е. камера отработала экспозицию, установленную функцией **sdu_set_exp**), либо экспозиция вообще не была запущена, то устанавливается бит **SDU_STAT_TRIG_ERR** в **status**.

Для использования этой возможности (программного обрыва экспозиции) рекомендуется с помощью функции **sdu_set_exp** установить экспозицию, заведомо превышающую требуемую.

3.2.36. **sdu_dac_use(CSDUCameraDevice *
pSDUCameraDevice, BOOL bUseDac)**

Функция служит для запуска ЦАП. Параметр **bUseDac** служит для включения (TRUE) или выключения (FALSE) ЦАП.

Функция доступна не для всех типов камер.

3.2.37. **sdu_dac_load_array(CSDUCameraDevice *
pSDUCameraDevice, DWORD dwArrayLength, char *
pArray)**

Функция служит для загрузки значений ЦАП. Значения передаются в массиве **pArray**, длина массива **dwArrayLength** не должна превышать 256.

Функция доступна не для всех типов камер.

3.2.38. **sdu_dac_set_arraystart(CSDUCameraDevice *
pSDUCameraDevice, DWORD dwArrayStart, DWORD
dwStepsCount)**

Функция служит для задания начального **dwArrayStart** индекса из массива значений ЦАП и числа используемых значений **dwStepsCount**.

Массив значений ЦАП должен быть предварительно загружен.

Функция доступна не для всех типов камер.

3.3 Функции передачи данных

3.3.1. `sdu_read_data (CSDUCameraDevice * pSDUCameraDevice, UCHAR *bufRead, DWORD length, DWORD *real_length)`

Функция производит считывание данных из буфера FIFO камеры в память компьютера.

Параметры:

***bufRead** - указатель на массив, в который будут считаны данные

length - количество байт которые будут считаны

***real_length** - указатель на переменную, в которой функция возвращает количество реально считанных байт

В случае успешного завершения функция возвращает значение **SDU_SUCCESS**, при этом **length** и **real_length** совпадают.

При попытке прочесть больше данных, чем есть в буфере камеры, функция через 2 секунды возвращает значение **SDU_TIMEOUT**.

При возникновении ошибки в процессе передачи данных из камеры по USB функция возвращает значение **SDU_DATA_ERROR**.

3.3.2. `sdu_open_file_rww(char * szFileName, Cam_RWWStruct * pCam_RWWStruct, BOOL bWriteMode, DWORD * pdwNumberFrames)`

Функция открывает rww (ccd) файл для чтения или для записи.

Параметры:

szFileName - имя файла

pCam_RWWStruct - указатель на Cam_RWWStruct структуру.

Вызывающая программа должна аллокировать эту структуру и, по окончании работы – деаллокировать.

bWriteMode – открытие для записи (TRUE) или для чтения (FALSE)

pdwNumberFrames – при открытии файла для чтения возвращает число кадров в файле. При открытии для записи не используется.

В случае успешного завершения функция возвращает значение **SDU_SUCCESS**, в случае ошибок могут возвращаться значения **SDU_DATA_ERROR** или **SDU_PARAM_ERROR**.

3.3.3. `sdu_file_rww_get_first_frame(Cam_RWWStruct * pCam_RWWStruct, DWORD dwBufferSize, UCHAR * pBuffer)`

Функция читает первый кадр из `rww (ccd)` файла.

Параметры:

`pCam_RWWStruct` – указатель на `Cam_RWWStruct` структуру

`dwBufferSize` – размер буфера памяти

`pBuffer` - буфер памяти для чтения кадра

В случае успешного завершения функция возвращает значение `SDU_SUCCESS`, в случае ошибок могут возвращаться значения `SDU_DATA_ERROR` или `SDU_PARAM_ERROR`.

3.3.4. `sdu_file_rww_get_next_frame(Cam_RWWStruct * pCam_RWWStruct, DWORD dwBufferSize, UCHAR * pBuffer, DWORD * pdwFrameNumber);`

Функция читает следующий кадр из `rww` файла.

Параметры:

`pCam_RWWStruct` – указатель на `Cam_RWWStruct` структуру

`dwBufferSize` – размер буфера памяти

`pBuffer` - буфер памяти для чтения кадра

`*pdwFrameNumber` – если этот параметр ненулевой – в нем возвращается текущий номер кадра.

В случае успешного завершения функция возвращает значение `SDU_SUCCESS`, в случае ошибок могут возвращаться значения `SDU_DATA_ERROR` или `SDU_PARAM_ERROR`.

```
3.3.5. sdu_file_rww_get_frame(Cam_RWWStruct *  
pCam_RWWStruct, DWORD dwBufferSize, UCHAR * pBuffer,  
DWORD dwFrameNumber);
```

Функция читает кадр с заданным номером из gww файла.

Параметры:

pCam_RWWStruct – указатель на **Cam_RWWStruct** структуру

dwBufferSize – размер буфера памяти

pBuffer - буфер памяти для чтения кадра

dwFrameNumber – номер кадра, который нужно прочесть

В случае успешного завершения функция возвращает значение **SDU_SUCCESS**, в случае ошибок могут возвращаться значения **SDU_DATA_ERROR** или **SDU_PARAM_ERROR**.

```
3.3.6. sdu_file_rww_write_frame(Cam_RWWStruct *  
pCam_RWWStruct, DWORD dwFrameSize, UCHAR * pBuffer);
```

Функция записывает кадр в gww (ccd) файл.

Параметры:

pCam_RWWStruct – указатель на **Cam_RWWStruct** структуру

dwFrameSize – размер кадра

pBuffer - буфер памяти, содержащий кадр

В случае успешного завершения функция возвращает значение **SDU_SUCCESS**, в случае ошибок могут возвращаться значения **SDU_DATA_ERROR** или **SDU_PARAM_ERROR**.

```
3.3.7. sdu_close_file_rww(Cam_RWWStruct * pCam_RWWStruct);
```

Функция закрывает gww (ccd) файл.

Параметры:

pCam_RWWStruct – указатель на **Cam_RWWStruct** структуру.

После закрытия файла вызывающая программа может удалить
аллокированную ранее

Cam_RWWStruct структуру.

В случае успешного завершения функция возвращает значение **SDU_SUCCESS**, в случае ошибок может возвращаться значения **SDU_PARAM_ERROR**.

3.4. Функции обработки изображения

3.4.1. `sdu_demosaic_12to8`(

```
    UCHAR *inbuf, UCHAR *outbuf,  
    UCHAR *gamma_r, UCHAR *gamma_g, UCHAR *gamma_b,  
    DWORD h_size, DWORD v_size,  
    DWORD colorseq, DWORD color_id, DWORD method,  
    BOOL bColorCorrection);
```

Функция служит для декодирования принятого из камеры цветного изображения в упакованном формате 12 бит/пиксель (см. формат представления данных в разделе 3.2.8). Результат работы функции - изображение в формате RGB 24 бит/пиксель (8 бит/цвет).

Параметры функции:

***inbuf** - указатель на массив, в котором находится считанное из камеры изображение.

Его размер - **h_size * v_size * 1.5**

***outbuf** - указатель на массив в который записывается результирующее изображение.

Его размер - **h_size * v_size * 3**

***gamma_r**

***gamma_g**

***gamma_b** - указатели на массивы для гамма-коррекции изображения для каждого из цветов.

Их размер - 4096 байт. Для создания этих массивов

можно использовать функцию **sdu_make_gamma_lut_12to8**

Если ***gamma_x=NULL** то, гамма-коррекция для пикселей соответствующего цвета не выполняется.

h_size,

v_size - размер изображения по горизонтали и по вертикали соответственно

colorseq - параметр, определяющий последовательность цветов в группе из трех байт, соответствующих одному пикселю результирующего изображения. Допустимые значения:

SDU_COLORSEQ_BGR - B,G,R

SDU_COLORSEQ_RGB - R,G,B

- color_id** - параметр, определяющий последовательность цветов в исходном изображении, считанном из камеры. Камера возвращает значение этого параметра при вызове функции **sdu_get_color_id**.
- method** - параметр, определяющий метод декодирования цветовой мозаики входного изображения (Bayer pattern) в RGB. Допустимые значения:
SDU_METHOD_FAST – простейшая билинейная Аппроксимация по 2 точкам
SDU_METHOD_BILINEAR - билинейная Аппроксимация по 4 точкам
SDU_METHOD_HQ - более сложный алгоритм, позволяющий получить качественное цветное изображение. Работает медленнее.
SDU_METHOD_GROUPING_PIXELS - алгоритм использующий сочетание методов. Работает еще медленнее.
SDU_METHOD_VARIABLE_GRADIENTS – алгоритм, использующий расчет градиентов.
SDU_METHOD_OPTIMAL – самый медленный и самый качественный алгоритм, работающий по многопроходной схеме.
- bColorCorrection** – параметр, указывающий на необходимость использования цветовой коррекции.

```

3.4.2. sdu_demosaic_12to16(
    UCHAR *inbuf, USHORT *outbuf,
    USHORT *GAMMA_R, USHORT *GAMMA_G,
    USHORT *GAMMA_B,
    DWORD H_SIZE, DWORD V_SIZE,
    DWORD COLORSEQ, DWORD COLOR_ID, DWORD method,
    BOOL bColorCorrection);

```

Функция служит для декодирования принятого из камеры цветного изображения в упакованном формате 12 бит/пиксель (см. формат представления данных в разделе 3.2.10). Результат работы функции - изображение в формате RGB 48 бит/пиксель (16 бит/цвет).

Параметры функции:

***inbuf** - указатель на массив, в котором находится считанное из камеры изображение.

Его размер - **h_size * v_size * 1.5**

***outbuf** - указатель на массив в который записывается результирующее изображение.

Его размер - **h_size * v_size * 3**

Младшие 4 бита каждого 16-битного элемента этого массива равны 0, т.е. значащими являются только 12 старших бит.

***GAMMA_R**

***GAMMA_G**

***GAMMA_B** - указатели на массивы для гамма-коррекции изображения для каждого из цветов.

Их размер - 4096. Для создания этих массивов можно использовать функцию

sdu_make_gamma_lut_12to16

Если ***gamma_x=NULL** то, гамма-коррекция для пикселей соответствующего цвета не выполняется.

h_size,

v_size - размер изображения по горизонтали и по вертикали соответственно

colorseq - параметр, определяющий последовательность цветов в группе из трех байт, соответствующих одному пикселю результирующего изображения. Допустимые значения:

SDU_COLORSEQ_BGR - B,G,R

SDU_COLORSEQ_RGB - R,G,B

- color_id** - параметр, определяющий последовательность цветов в исходном изображении, считанном из камеры. Камера возвращает значение этого параметра при вызове функции **sdu_get_color_id**.
- method** - параметр, определяющий метод декодирования цветовой мозаики входного изображения (Bayer pattern) в RGB. Допустимые значения:
SDU_METHOD_FAST – простейшая билинейная Аппроксимация по 2 точкам
SDU_METHOD_BILINEAR - билинейная Аппроксимация по 4 точкам
SDU_METHOD_HQ - более сложный алгоритм, позволяющий получить качественное цветное изображение. Работает медленнее.
SDU_METHOD_GROUPING_PIXELS-алгоритм использующий сочетание методов. Работает еще медленнее.
SDU_METHOD_VARIABLE_GRADIENTS – алгоритм, использующий расчет градиентов.
SDU_METHOD_OPTIMAL – самый медленный и самый качественный алгоритм, работающий по многопроходной схеме.
- bColorCorrection** – параметр, указывающий на необходимость использования цветовой коррекции.

```

3.4.3. sdu_demosaic_8to8(
    UCHAR *inbuf, UCHAR *outbuf,
    UCHAR *gamma_r, UCHAR *gamma_g, UCHAR *gamma_b,
    DWORD h_size, DWORD v_size,
    DWORD colorseq, DWORD color_id, DWORD method,
    BOOL bColorCorrection);

```

Функция служит для декодирования принятого из камеры цветного изображения в формате 8 бит/пиксель. Результат работы функции - изображение в формате RGB 24 бит/пиксель (8 бит/цвет).

Параметры функции:

***inbuf** - указатель на массив, в котором находится считанное из камеры изображение.
Его размер - **h_size * v_size**

***outbuf** - указатель на массив в который записывается результирующее изображение.
Его размер - **h_size * v_size * 3**

***gamma_r**

***gamma_g**

***gamma_b** - указатели на массивы для гамма-коррекции изображения для каждого из цветов.

Их размер - 4096 байт. Для создания этих массивов можно использовать функцию **sdu_make_gamma_lut_12to8**

Если ***gamma_x=NULL** то, гамма-коррекция для пикселей соответствующего цвета не выполняется.

h_size,

v_size - размер изображения по горизонтали и по вертикали соответственно

colorseq - параметр, определяющий последовательность цветов в группе из трех байт, соответствующих одному пикселю результирующего изображения. Допустимые значения:
SDU_COLORSEQ_BGR - B,G,R
SDU_COLORSEQ_RGB - R,G,B

color_id - параметр, определяющий последовательность цветов в исходном изображении, считанном из камеры. Камера возвращает значение этого параметра при вызове функции **sdu_get_color_id**.

method - параметр, определяющий метод декодирования цветовой мозаики входного изображения (Bayer pattern) в RGB. Допустимые значения:
SDU_METHOD_FAST – простейшая билинейная Аппроксимация по 2 точкам
SDU_METHOD_BILINEAR - билинейная Аппроксимация по 4 точкам
SDU_METHOD_HQ - более сложный алгоритм, позволяющий получить качественное цветное изображение. Работает медленнее.
SDU_METHOD_GROUPING_PIXELS-алгоритм использующий сочетание методов. Работает еще медленнее.
SDU_METHOD_VARIABLE_GRADIENTS – алгоритм, использующий расчет градиентов.
SDU_METHOD_OPTIMAL – самый медленный и самый качественный алгоритм, работающий по многопроходной схеме.

bColorCorrection – параметр, указывающий на необходимость использования цветовой коррекции.

```
3.4.4. sdu_demosaic_8to16(  
    UCHAR *inbuf, USHORT *outbuf,  
    UCHAR *gamma_r, UCHAR *gamma_g, UCHAR *gamma_b,  
    DWORD h_size, DWORD v_size,  
    DWORD colorseq, DWORD color_id, DWORD method,  
    BOOL bColorCorrection);
```

Вспомогательная функция служит для выполнения гамма-коррекции принятого из камеры цветного изображения в формате 8 бит/пиксель. Применяется при записи в 16-битный файл изображения. Внутри вызывает функцию **sdu_demosaic_8to8**.

```
3.4.5. sdu_bw_convert_12to16(  
        UCHAR *inbuf, USHORT *outbuf,  
        USHORT *gamma,  
        DWORD h_size, DWORD v_size)
```

Функция служит для преобразования принятого из камеры черно-белого упакованного изображения в формате 12 бит/пиксель в формат 16 бит/пиксель и выполнения гамма-коррекции изображения.

Параметры функции:

***inbuf** - указатель на массив, в котором находится считанное из камеры изображение.
Его размер - **h_size * v_size * 1.5**

***outbuf** - указатель на массив в который записывается результирующее изображение.
Его размер - **h_size * v_size**
Младшие 4 бита каждого 16-битного элемента этого массива равны 0, т.е. значащими являются только 12 старших бит.

***gamma** - указатель на массив для гамма-коррекции изображения. Его размер - 4096. Для создания этого массива можно использовать функцию **sdu_make_gamma_lut_12to16**
Если ***gamma=NULL** то, гамма-коррекция не выполняется.

h_size,

v_size - размер изображения по горизонтали и по вертикали соответственно

3.4.6. `sdu_bw_convert_12to8` (
`UCHAR *inbuf, UCHAR *outbuf,`
`UCHAR *gamma,`
`DWORD h_size, DWORD v_size)`

Функция служит для преобразования принятого из камеры черно-белого упакованного изображения в формате 12 бит/пиксель в формат 8 бит/пиксель и выполнения гамма-коррекции изображения.

Параметры функции:

***inbuf** - указатель на массив, в котором находится считанное из камеры изображение.
Его размер - **`h_size * v_size * 1.5`**

***outbuf** - указатель на массив в который записывается результирующее изображение.
Его размер - **`h_size * v_size`**

***gamma** - указатель на массив для гамма-коррекции изображения. Его размер - 4096. Для создания этого массива можно использовать функцию **`sdu_make_gamma_lut_12to8`**
Если **`*gamma=NULL`** то, гамма-коррекция не выполняется.

h_size,
v_size - размер изображения по горизонтали и по вертикали соответственно


```
3.4.7.   sdu_bw_convert_8to8 (
          UCHAR *inbuf, UCHAR *outbuf,
          UCHAR *gamma,
          DWORD h_size, DWORD v_size)
```

Функция служит для выполнения гамма-коррекции принятого из камеры черно-белого изображения в формате 8 бит/пиксель.

Параметры функции:

***inbuf** - указатель на массив, в котором находится считанное из камеры изображение.
Его размер - **h_size * v_size**

***outbuf** - указатель на массив в который записывается результирующее изображение.
Его размер - **h_size * v_size**

***gamma** - указатель на массив для гамма-коррекции изображения. Его размер - 256. Для создания этого массива можно использовать функцию **sdu_make_gamma_lut_8to8**
Если ***gamma=NULL** то, гамма-коррекция не выполняется.

h_size,

v_size - размер изображения по горизонтали и по вертикали соответственно

```
3.4.8.   sdu_bw_convert_8to16 (
          UCHAR *inbuf, USHORT *outbuf,
          UCHAR *gamma,
          DWORD h_size, DWORD v_size)
```

Вспомогательная функция служит для выполнения гамма-коррекции принятого из камеры черно-белого изображения в формате 8 бит/пиксель. Применяется при записи в 16-битный файл изображения. Внутри вызывает функцию **sdu_bw_convert_8to8**.

3.4.9. `sdu_make_gamma_lut_12to16(`
`USHORT *gamma, float g)`

Функция служит для формирования массива гамма-коррекции, используемого при преобразовании изображения из формата 12 бит/пиксель в формат 8 бит/пиксель.

***gamma** - указатель на массив для гамма-коррекции изображения. Его размер - 4096.

float g - значение коэффициента гамма-коррекции, для которого строится массив, в пределах от 0 до 1.

3.4.10. `sdu_make_gamma_lut_12to8(`
`UCHAR *gamma, float g)`

Функция служит для формирования массива гамма-коррекции, используемого при преобразовании изображения из формата 12 бит/пиксель в формат 8 бит/пиксель.

***gamma** - указатель на массив для гамма-коррекции изображения. Его размер - 4096 байт.

float g - значение коэффициента гамма-коррекции, для которого строится массив, в пределах от 0 до 1.

3.4.11. `sdu_make_gamma_lut_8to8(`
`UCHAR *gamma, float g)`

Функция служит для формирования массива гамма-коррекции, используемого при обработке изображения в формате 8 бит/пиксель.

***gamma** - указатель на массив для гамма-коррекции изображения. Его размер - 256 байт.

float g - значение коэффициента гамма-коррекции, для которого строится массив, в пределах от 0 до 1.

4. История изменений.

Версия	Дата	Внесенные изменения
1.00	19.11.04	Исходный релиз
1.10	29.11.04	Добавлены методы декодирования цвета SDU_METHOD_GROUPING_PIXELS SDU_METHOD_VARIABLE_GRADIENTS
1.20	27.04.05	Добавлены функции sdu_set_draft и sdu_get_draft
1.21	15.06.05	Добавлена функция sdu_get_serialNumber
1.30	27.06.05	Добавлен параметр CSDUCameraDevice * для большинства функций.
1.31	22.08.05	Добавлен параметр bColorCorrection для DemosaicXXX - функций
1.32	17.10.05	Добавлены функции sdu_demosaic_8to16 и sdu_bw_convert8to16 . Добавлен метод sdu_get_DeviceCount .
1.33	20.02.06	Добавлены флаги SDU_STAT_DOUBLE_READ в статус и SDU_TRIGMODE_DOUBLE_FLAG в методе задания режима запуска.
1.34	21.11.06	Добавлены методы запуска SDU_TRIGMODE_EXTERNAL_SINGLE SDU_TRIGMODE_BULB_SINGLE . Добавлен метод декодирования цвета SDU_METHOD_OPTIMAL .
1.35	5.12.06	Добавлены функции sdu_dac_use sdu_dac_load_array sdu_dac_set_arraystart
1.36	19.12.06	Добавлены функции sdu_open_file_rww sdu_file_rww_get_first_frame sdu_file_rww_get_next_frame sdu_file_rww_get_frame sdu_file_rww_write_frame sdu_close_file_rww