

SDK. Руководство пользователя

© 2020 ООО "Элмин Ср" www.elmins-sr.com

Содержание

Часть I SDK, Руководство пользователя	4
1 Введение.....	4
2 Необходимые файлы.....	4
3 Список функций.....	7
Функции инициализации	7
Функции проверки состояния	7
Функции установки	7
Функции получения статуса	7
Функции управления сканированием	7
Функции управления полосами ПЗС-матрицы	7
CCD_Init	7
CCD_GetSerialNum	8
CCD_GetSerialNumber	8
CCD_GetID	8
CCD_HitTest	8
CCD_SetExtendParameters	9
CCD_GetExtendParameters	9
CCD_SetParameter	9
CCD_GetParameter	10
CCD_SetParameters	11
CCD_GetParameters	12
CCD_GetMeasureStatus	12
CCD_InitMeasuring	12
CCD_InitMeasuringData	13
CCD_DoneMeasuringData	13
CCD_StartWaitMeasuring	14
CCD_StartMeasuring	14
CCD_GetData	14
CCD_CameraReset	15
CCD_ClearStrips	15
CCD_AddStrip	15
CCD_DeleteStrip	15
4 Описание структур.....	16
TCCDUSBParams	16
TCCDUSBExtendParams	17
5 Получение информации о камере.....	18
6 Синхронизация	18
7 Управление параметрами АПЦ	19
8 Режим быстрого сканирования.....	21
9 Режим сканирования DEVICEMODESTREAM.....	21
10 Пример 1.....	24
11 Пример 2.....	24
12 Пример 3.....	24

13	Пример 4.....	25
----	---------------	----

1 SDK, Руководство пользователя

1.1 Введение

DLL CCDUSBDCOM01.DLL содержит все функции необходимые для управлением контроллером системы регистрации фирмы Элминс-СР на основе CCD датчика. Функции работают с Windows 98/ME/2000/XP/7/8/10, Linux Ubuntu. Все функции должны быть вызваны с директивой компилятора stdcall.

В папке x86 находятся DLL и примеры для 32-битной версии операционной системы Windows.

В папке x64 находятся DLL и примеры для 64-битной версии операционной системы Windows.

В папке Linux находятся So-файл и примеры для 64-битной версии операционной системы Linux.

1.2 Необходимые файлы

Для работы нужны следующие файлы:

CCDUSBDCOM01.dll основная DLL.

CyUSB2Ormins.dll дополнительная DLL. Должна располагаться в той же директории.

StrHexToInt.dll дополнительная DLL. Должна располагаться в той же директории (для версии x86).

ep_pair.hex дополнительный файл.. Должен располагаться в той же директории.

CCDUSB021_I.pas файл содержит определения типов, констант и прототипы функций на Delphi.

UsbCDD.h, UsbCDD.cpp файлы содержат определения типов, классов, констант и прототипы функций на MS VC++

*.cnf все файлы с расширением cnf из комплекта поставки.

Камеры на основе светочувствительных линеек

Камеры на основе светочувствительных линеек могут работать с разными наборами драйверов.

Системы регистрации с интерфейсом USB 1.1

Для работы камеры необходимо установить соответствующие драйверы.

Для Windows 98, Me:

ezusb.sys	Драйвер для USB прибора. Этот файл должен быть помещен в системную директорию \WINDOWS\SYSTEM32\DRIVERS.
SLLOADER.sys	Драйвер для USB прибора. Этот файл должен быть помещен в системную директорию \WINDOWS\SYSTEM32\DRIVERS.
Ezusbw2k.inf	Системный файл. Этот файл должен быть помещен в системную директорию \WINDOWS\INF.

Для Windows 2000\XP:

Для работы камеры необходимо установить драйверы.

ezusb.sys	Драйвер для USB прибора.
SLLOADER.sys	Драйвер для USB прибора.
Ezusbw2k.inf	Системный файл.

Для Windows 7/8/10 x86 / x64

Для работы камеры необходимо установить драйверы.

cyusb.inf	Системный файл.
cyusb.sys	Драйвер для USB прибора.
cyusb.cat	Каталог безопасности.

Для Windows 7 x86 из папки SDK\Drivers\wlh\x86.

Для Windows 7 x64 из папки SDK\Drivers\wlh\x64.

Системы регистрации с интерфейсом USB 2.0

Для работы камеры необходимо установить соответствующие драйверы.

Для Windows 2000\XP:

ezusb.sys	Драйвер для USB прибора.
SLLOADER.sys	Драйвер для USB прибора.
Ezusbw2k.inf	Системный файл.

Для Windows 7/8/10 x86 / x64

cyusb.inf	Системный файл.
cyusb.sys	Драйвер для USB прибора.
cyusn.cat	Каталог безопасности.

Для Windows 7/8/10 x86 из папки SDK\Drivers\wlh\x86.

Для Windows 7/8/10 x64 из папки SDK\Drivers\wlh\x64.

Камеры на основе ПЗС-матрицы:

Системы регистрации с интерфейсом USB 2.0

Для работы камеры необходимо установить драйверы.

Для Windows 2000\XP:

ezusb.sys Драйвер для USB прибора.

SLLOADER.sys Драйвер для USB прибора.

Ezusbw2k.inf Системный файл.

Операционные системы Linux

1. Нужно установить библиотеку LibUSB с официального сайта "<http://libusb.org>". Для этого в консоли ОС выполните команду:

```
sudo apt-get install libusb-1.0-0-dev
```

2. Копировать файлы 88-cyusb.rules и 89-elminssr.rules из папки SDK/Linux/rules в системную папку /etc/udev/rules.d . Нужны права администратора.

3. Что бы скомпилировать пример Project1 нужно установить Free pascal и IDE Lazarus.

Устанавливаем Free Pascal командой в консоли:

```
sudo apt-get install fpc
```

4. Устанавливаем IDE Lazarus:

```
sudo apt-get install Lazarus
```

Для работы нужны следующие файлы:

libCCDUSBDCOM01.so основной файл библиотеки.

ep_pair.hex дополнительный файл.. Должен располагаться в той же папке.

*.cnf все файлы с расширением cnf из комплекта поставки.

В папке Linux можно найти файлы примера Project1.

1.3 Список функций

Следующие функции реализованы в DLL CCDUSBDCOM01.dll:

1.3.1 Функции инициализации

[CCD_Init](#)
[CCD_GetSerialNum](#)
[CCD_GetSerialNumber](#)
[CCD_GetID](#)
[CCD_CameraReset](#)

1.3.2 Функции проверки состояния

[CCD_HitTest](#)

1.3.3 Функции установки

[CCD_SetExtendParameters](#)
[CCD_GetExtendParameters](#)
[CCD_SetParameter](#)
[CCD_GetParameter](#)
[CCD_SetParameters](#)
[CCD_GetParameters](#)

1.3.4 Функции получения статуса

[CCD_GetMeasureStatus](#)

1.3.5 Функции управления сканированием

[CCD_InitMeasuring](#)
[CCD_InitMeasuringData](#)
[CCD_StartWaitMeasuring](#)
[CCD_StartMeasuring](#)
[CCD_GetData](#)
[CCD_CameraReset](#)
[CCD_DoneMeasuring](#)

1.3.6 Функции управления полосами ПЗС-матрицы

[CCD_ClearStrips](#)
[CCD_AddStrip](#)
[CCD_DeleteStrip](#)

1.3.7 CCD_Init

function CCD_Init (ahAppWnd : HWND; Prm : PChar; Var ID : Integer) : boolean; stdcall;

Эта функция должна быть вызвана до любых других. Она вызывается только один раз.
Функция производит поиск подключенных приборов и устанавливает начальные параметры.

ahAppWnd это дескриптор приложения (HWND), вызывающего функции DLL. Если есть проблемы с получением HWND, то hAppWnd может быть 0. Ptm и ID не используются. Если функция выполнила операцию правильно, то она вернет значение TRUE.

1.3.8 CCD_GetSerialNum

function CCD_GetSerialNum (ID : Integer; Var sernum : PChar) : boolean; stdcall;

Функция CCD_GetSerialNum возвращает уникальный серийный номер ПЗС-камеры. Если к компьютеру подключено несколько приборов, то ID используется как идентификатор ПЗС-прибора. ID может принимать значение 0,1,2. Если используется один прибор, то ID = 0. Если функция выполнила операцию правильно, то она вернет значение TRUE.

1.3.9 CCD_GetSerialNumber

function CCD_GetSerialNumber (ID : Integer) : PChar; stdcall;

Функция CCD_GetSerialNumber возвращает уникальный серийный номер ПЗС-камеры. Если к компьютеру подключено несколько приборов, то ID используется как идентификатор ПЗС-прибора. ID может принимать значение 0,1,2. Если используется один прибор, то ID = 0. Если функция выполнила операцию правильно, то она вернет указатель на строку серийного номера.

1.3.10 CCD_GetID

function CCD_GetID (sernum : PChar; Var ID : Integer) : boolean; stdcall;

Функция CCD_GetID позволяет получить ID для камеры с известным серийным номером. sernum указатель на строку символов, содержащих серийный номер камеры ID которого мы ищем. Если функция вернула значение TRUE, то камера с указанным серийным номером найдена и в переменной ID ее идентификатор. В противном случае, камера с серийным номером sernum не найдена.

1.3.11 CCD_HitTest

function CCD_HitTest (ID : Integer) : boolean; stdcall;

Функция CCD_HitTest используется для быстрого тестирования ПЗС-камеры. Если к компьютеру подключено несколько приборов, то ID используется как идентификатор ПЗС-прибора. ID может принимать значение 0,1,2. Если используется один прибор, то ID = 0. Если функция выполнила операцию правильно и камера готова к работе в данный момент, то она вернет значение TRUE.

1.3.12 CCD_SetExtendParameters

```
function CCD_SetExtendParameters(ID : Integer; var Prms : TCCDUSBExtendParams) :  
boolean; stdcall;
```

Функция CCD_SetExtendParameters устанавливает параметры регистрации.

Параметр Prms это структура типа [TCCDUSBExtendParams](#).

Можно устанавливать только следующие параметры.

- время экспозиции
- число регистраций
- [способ синхронизации](#)
- режим прибора
- полосы
- время предварительного обжига
- повышенная чувствительность
- время затвора
- частота АЦП

Все остальные параметры устанавливаются автоматически.

Разные типы камер имеют разные возможности. Вы должны запрашивать PRM_DEVICEPROPERTY, используя функцию [CCD_GetParameter](#) или поле dwProperty структуры [TCCDUSBExtendParams](#), используя функцию [CCD_GetExtendParameters](#). См. раздел [«Получение информации о ПЗС-приборе»](#).

См. [CCD_SetParameter](#), [CCD_SetParameters](#), [TCCDUSBExtendParams](#), [Синхронизация](#).

Если к компьютеру подключено несколько приборов, то ID используется как идентификатор ПЗС-прибора. ID может принимать значение 0,1,2.

Если используется один прибор, то ID = 0.

Если функция выполнила операцию правильно, то она вернет значение TRUE.

1.3.13 CCD_GetExtendParameters

```
function CCD_GetExtendParameters(ID : Integer; var Prms : TCCDUSBExtendParams) :  
boolean; stdcall;
```

Функция CCD_GetExtendParameters позволяет получить параметры ПЗС-камеры.

CCD_GetExtendParameters заполняет поля переменной Prms : [TCCDUSBExtendParams](#).

Смотрите также [CCD_SetExtendParameters](#), [CCD_GetParameter](#) и [CCD_GetParameters](#).

Если к компьютеру подключено несколько приборов, то ID используется как идентификатор ПЗС-прибора. ID может принимать значение 0,1,2.

Если используется один прибор, то ID = 0.

Если функция выполнила операцию правильно, то она вернет значение TRUE.

1.3.14 CCD_SetParameter

```
function CCD_SetParameter(ID : Integer; dwPrmID : DWORD; Prm : single) : boolean;  
stdcall;
```

Функция CCD_SetParameter устанавливает параметры регистрации.

Можно устанавливать только следующие параметры.

- время экспозиции
- число регистраций
- [способ синхронизации](#)
- режим прибора
- полосы
- время предварительного обжига
- повышенная чувствительность
- время затвора
- частота АЦП
- усилением АЦП
- смещением нулевого уровня АЦП

Все остальные параметры устанавливаются автоматически.

dwPrmID : DWORD – идентификатор параметра. См. [CCD_GetParameter](#).

Prm : single – устанавливаемое значение параметра.

Разные типы камер имеют разные возможности. Вы должны запрашивать PRM_DEVICEPROPERTY, используя функцию [CCD_GetParameter](#) или поле dwProperty структуры [TCCDUSBExtendParams](#), используя функцию [CCD_GetExtendParameters](#). См. раздел [«Получение информации о ПЗС-приборе»](#).

См. [CCD_SetExtendParameters](#), [TCCDUSBExtendParams](#), [CCD_SetParameters](#).

Если к компьютеру подключено несколько приборов, то ID используется как идентификатор ПЗС-прибора. ID может принимать значение 0,1,2.

Если используется один прибор, то ID = 0.

Если функция выполнила операцию правильно, то она вернет значение TRUE.

1.3.15 CCD_GetParameter

function CCD_GetParameter(ID : Integer; dwPrmID : DWORD; var Prm : single) : boolean; stdcall;

Функция CCD_GetParameter позволяет получить параметр ПЗС-камеры.

dwPrmID : DWORD – идентификатор параметра. Может принимать следующие значения:

PRM_DIGIT = 1 — Число бит АЦП.

PRM_PIXELRATE = 2 — Частота работы АЦП. Частота выходного регистра ПЗС-датчика (КГц).

PRM_NUMPIXELS = 3 — Число пикселей ПЗС-датчика.

PRM_READOUTS = 4 — Число регистраций.

PRM_EXPTIME = 5 — Время экспозиции. Время одной регистрации в миллисекундах.

PRM_SYNCHR = 6 — Способ [синхронизации](#).

PRM_NUMPIXELSH = 7 — Число пикселей по горизонтали (число колонок ПЗС-матрицы).

PRM_NUMPIXELSV = 8 — Число пикселей по вертикали (число рядов ПЗС-матрицы).

PRM_SUMMING = 9 — Суммирующий режим. Не используется.

PRM_DEVICEMODE = 10 — Режим камеры. DEVICEMODEA1 – режим матрицы №1.

DEVICEMODEA2 – режим матрицы №2. DEVICEMODES - Спектроскопический режим.

Матрица работает как одна ПЗС-линейка.

DEVICEMODESTREAM - камера работает в режиме потока. DEVICEMODEFASTSCAN -

камера работает в режиме быстрого сканирования.

PRM_STRIPCOUNT = 11 — Число выделенных полос для спектроскопического режима ПЗС-матрицы.

PRM_SENSIT = 14 — Управление чувствительностью. 0 — обычная чувствительность. 1 — повышенная чувствительность.

PRM_DEVICEPROPERTY = 15 — Свойства прибора. См. [Получение информации о ПЗС-камере](#).

PRM_PREBURNING = 16 — Время предварительного обжига в секундах. Действительно только для режима синхронизации SYNCHR_CONTR_NEG, но не для всех камер. См. [Получение информации о ПЗС-камере](#).

PRM_SHUTERTIME = 17 — Время срабатывания электронного затвора в миллисекундах.

PRM_KAmpRed = \$1022; — Усиление АЦП. Первый канал

PRM_OffsetRed = \$1034; — Смещение нулевого уровня АЦП. Первый канал

PRM_KAmpGreen = \$1023; — Усиление АЦП. Второй канал

PRM_OffsetGreen = \$1035; — Смещение нулевого уровня АЦП. Второй канал

Prm : single – возвращаемое значение параметра.

Разные типы камер имеют разные возможности. Вы должны запрашивать PRM_DEVICEPROPERTY, используя функцию [CCD_GetParameter](#) или поле dwProperty структуры [TCCDUSBExtendParams](#), используя функцию [CCD_GetExtendParameters](#). См. раздел [Получение информации о ПЗС-камере](#).

См. [CCD_SetExtendParameters](#), [CCD_SetParameters](#).

Если к компьютеру подключено несколько приборов, то ID используется как идентификатор ПЗС-прибора. ID может принимать значение 0,1,2.

Если используется один прибор, то ID = 0.

Если функция выполнила операцию правильно, то она вернет значение TRUE.

1.3.16 CCD_SetParameters

function CCD_SetParameters(ID : Integer; var Prms : TCCDUSBParams) : boolean;
stdcall;

Функция CCD_SetParameters устанавливает параметры регистрации.

Параметр Prms это структура типа [TCCDUSBParams](#).

Можно устанавливать только следующие параметры.

- время экспозиции
- число регистраций
- [способ синхронизации](#)

Все остальные параметры устанавливаются автоматически.

См. [CCD_SetParameter](#), [CCD_SetExtendParameters](#), [Синхронизация](#).

Если к компьютеру подключено несколько приборов, то ID используется как идентификатор ПЗС-прибора. ID может принимать значение 0,1,2.

Если используется один прибор, то ID = 0.

Если функция выполнила операцию правильно, то она вернет значение TRUE.

1.3.17 CCD_GetParameters

```
function CCD_GetParameters ( ID : Integer; var Prms : TCCDUSBParams) : boolean;  
stdcall;
```

Функция CCD_GetParameters позволяет получить параметры ПЗС-камеры.

CCD_GetParameters заполняет поля переменной Prms : [TCCDUSBParams](#).

Смотрите также [CCD_SetParameters](#), [CCD_GetParameter](#) и [CCD_GetExtendParameters](#).

Если к компьютеру подключено несколько приборов, то ID используется как идентификатор ПЗС-прибора. ID может принимать значение 0,1,2.

Если используется один прибор, то ID = 0.

Если функция выполнила операцию правильно, то она вернет значение TRUE.

1.3.18 CCD_GetMeasureStatus

```
function CCD_GetMeasureStatus ( ID : Integer; var adwStatus : DWORD) : boolean;  
stdcall;
```

Функция CCD_GetMeasureStatus используется для проверки состояния камеры в процессе регистрации. Эта функция вызывается совместно с [CCD_StartMeasuring](#).

dwStatus – может принимать следующие значения:

STATUS_WAIT_DATA – камера в процессе регистрации.

STATUS_WAIT_TRIG – камера в режиме ожидания внешнего синхроимпульса.

STATUS_DATA_READY – предыдущая регистрация окончена. Можно забирать данные и начинать новую регистрацию.

Если к компьютеру подключено несколько приборов, то ID используется как идентификатор ПЗС-прибора. ID может принимать значение 0,1,2.

Если используется один прибор, то ID = 0.

Если функция выполнила операцию правильно, то она вернет значение TRUE.

1.3.19 CCD_InitMeasuring

```
function CCD_InitMeasuring(ID : Integer) : boolean; stdcall;
```

Функция CCD_InitMeasuring должна вызываться до начала регистрации, в случае если были изменены параметры регистрации. Можно использовать только для систем регистрации на основе ПЗС-линеек. Для новых разработок и систем регистрации на основе ПЗС-матриц следует вызвать функцию [CCD_InitMeasuringData](#).

Если к компьютеру подключено несколько приборов, то ID используется как идентификатор ПЗС-прибора. ID может принимать значение 0,1,2.

Если используется один прибор, то ID = 0.

Если функция выполнила операцию правильно, то она вернет значение TRUE.

1.3.20 CCD_InitMeasuringData

function CCD_InitMeasuringData(ID : Integer; apData : PDWORDArr) : boolean; stdcall;

Функция CCD_InitMeasuringData должна вызываться до начала регистрации, в случае если были изменены параметры регистрации.

apData – указатель на массив DWORD. Длина выделенной памяти должна быть эквивалентна числу пикселей ПЗС-датчика и числу установленных регистраций ($4 * N_{\text{pixels}} * N_{\text{frames}}$).

Если к компьютеру подключено несколько приборов, то ID используется как идентификатор ПЗС-прибора. ID может принимать значение 0, 1, 2.

Если используется один прибор, то ID = 0.

Если функция выполнила операцию правильно, то она вернет значение TRUE.

См. [Пример 1](#)

1.3.21 CCD_DoneMeasuringData

function CCD_DoneMeasuring(ID : Integer) : boolean; stdcall;

Функция CCD_DoneMeasuring Может вызываться в конце серии регистраций, если вы не планируете начинать измерения опять. Имеет существенное значение для камер использующих USB ISO передачу данных.

Ниже перечислены названия этих камер:

U2C-14P4
U2C-14P22
U2C-14P24
U2C-14S415
U2C-16E421
U2C-16H7108
U2C-16H7158
U2C-16H7306
U2C-16H7307
U2C-16H7308
U2C-16H7316
U2C-16H7317
U2C-16H7318
U2C-10M403

Для этих камер функция останавливает ISO-поток в USB драйвере, который использует ресурсы компьютера.

Для остальных камер эта функция не имеет значения.

CCD_DoneMeasuring автоматически вызывается при выходе из DLL и останавливает поток.

Если к компьютеру подключено несколько приборов, то ID используется как идентификатор ПЗС-прибора. ID может принимать значение 0, 1, 2.

Если используется один прибор, то ID = 0.

Если функция выполнила операцию правильно, то она вернет значение TRUE.

1.3.22 CCD_StartWaitMeasuring

function CCD_StartWaitMeasuring (ID : Integer) : boolean; stdcall;

Функция CCD_StartWaitMeasuring инициирует регистрацию. Функция не вернет управление до тех пор, пока регистрация не окончится. Для того, что бы прервать регистрацию нужно из другого потока вызвать функцию [CCD_CameraReset](#).

См. также [CCD_InitMeasuringData](#), [CCD_StartMeasuring](#).

Если к компьютеру подключено несколько приборов, то ID используется как идентификатор ПЗС-прибора. ID может принимать значение 0,1,2.

Если используется один прибор, то ID = 0.

Если функция выполнила операцию правильно, то она вернет значение TRUE.

[Пример.](#)

1.3.23 CCD_StartMeasuring

function CCD_StartMeasuring (aID : Integer) : boolean; stdcall;

Функция CCD_StartMeasuring инициирует регистрацию. Функция возвращает управление сразу же, до окончания регистрации. Для проверки статуса вызывайте функцию [CCD_GetMeasureStatus](#). Для того, что бы прервать регистрацию нужно функцию [CCD_CameraReset](#).

См. также [CCD_InitMeasuringData](#), [CCD_StartWaitMeasuring](#)

Если к компьютеру подключено несколько приборов, то ID используется как идентификатор ПЗС-прибора. ID может принимать значение 0,1,2.

Если используется один прибор, то ID = 0.

Если функция выполнила операцию правильно, то она вернет значение TRUE.

[Пример.](#)

1.3.24 CCD_GetData

function CCD_GetData (ID : Integer; pData : PDWORDArr) : boolean; stdcall;

Функция CCD_GetData используется для получения данных (результатов регистрации). pData – указатель на массив DWORD. Длина выделенной памяти должна быть эквивалентна числу пикселей ПЗС-датчика и числу установленных регистраций (4*Npixels*Nframes). Функция применима только для камер на основе ПЗС-линейки. Используется совместно с функцией [CCD_InitMeasuring](#). В новых разработках и для всех камер на основе ПЗС-матрицы следует использовать функцию [CCD_InitMeasuringData](#).

Если к компьютеру подключено несколько приборов, то ID используется как идентификатор ПЗС-прибора. ID может принимать значение 0,1,2.

Если используется один прибор, то ID = 0.

Если функция выполнила операцию правильно, то она вернет значение TRUE.

1.3.25 CCD_CameraReset

function CCD_CameraReset (ID : Integer) : boolean; stdcall;

CCD_CameraReset прерывает текущую регистрацию, инициализирует внутренние переменный DLL, систему регистрации, прерывает передачу данных и инициализирует канал передачи.

Вызывайте функцию CCD_CameraReset только если произошел сбой или пользователю необходимо прервать текущую регистрацию. Нет необходимости вызывать эту функцию каждый раз перед началом регистрации или после ее окончания.

Если к компьютеру подключено несколько приборов, то ID используется как идентификатор ПЗС-прибора. ID может принимать значение 0,1,2.

Если используется один прибор, то ID = 0.

Если функция выполнила операцию правильно, то она вернет значение TRUE.

1.3.26 CCD_ClearStrips

function CCD_ClearStrips (ID : Integer) : boolean; stdcall;

Эта функция используется для управления параметрами спектроскопического режима ПЗС-матрицы. Т

CCD_ClearStrips очищает список полос.

Если к компьютеру подключено несколько приборов, то ID используется как идентификатор ПЗС-прибора. ID может принимать значение 0,1,2.

Если используется один прибор, то ID = 0.

Если функция выполнила операцию правильно, то она вернет значение TRUE.

См. также [CCD_AddStrip](#), [CCD_DeleteStrip](#), [CCD_GetExtendParameters](#), [CCD_SetExtendParameters](#), [Получение информации о ПЗС-камере](#).

1.3.27 CCD_AddStrip

function CCD_AddStrip (ID : Integer; arcStrip : TRect) : boolean; stdcall;

Эта функция используется для управления параметрами спектроскопического режима ПЗС-матрицы.

CCD_AddStrip добавляет одну полосу в список полос. Параметры новой полосы определяются в arcStrip.

Если к компьютеру подключено несколько приборов, то ID используется как идентификатор ПЗС-прибора. ID может принимать значение 0,1,2.

Если используется один прибор, то ID = 0.

Если функция выполнила операцию правильно, то она вернет значение TRUE.

См. также [CCD_ClearStrips](#), [CCD_DeleteStrip](#), [CCD_GetExtendParameters](#), [CCD_SetExtendParameters](#), [Получение информации о ПЗС-камере](#).

1.3.28 CCD_DeleteStrip

function CCD_DeleteStrip (ID : Integer; Index : Integer) : boolean; stdcall;

Эта функция используется для управления параметрами спектроскопического режима ПЗС-

матрицы.

CCD_DeleteStrip удаляет полосу с индексом Index из списка полос.

Если к компьютеру подключено несколько приборов, то ID используется как идентификатор ПЗС-прибора. ID может принимать значение 0,1,2.

Если используется один прибор, то ID = 0.

Если функция выполнила операцию правильно, то она вернет значение TRUE.

См. также [CCD_ClearStrips](#), [CCD_AddStrip](#), [CCD_GetExtendParameters](#), [CCD_SetExtendParameters](#), [Получение информации о ПЗС-камере](#).

1.4 Описание структур

1.4.1 TCCDUSBParams

Функции [CCD_SetParameters](#) и [CCD_GetParameters](#) используют как параметр переменную типа TCCDUSBParams. Ниже описаны значения этой структуры.

```
TCCDUSBParams =
record
  dwDigitCapacity : DWORD;           //The digit capacity of CCD-camera
  nPixelRate      : integer;         //The pixel rate kHz
  nNumPixels      : integer;         //The number of pixels

  nNumReadOuts    : integer;         //The number of readouts
  nExposureTime   : integer;         //The exposure time
  dwSynchr        : DWORD;           //The synchronization mode
end;
```

dwDigitCapacity	: DWORD;	Число бит АЦП системы регистрации
nPixelRate	: integer;	Частота АЦП. Частота оцифровки пикселей
nNumPixels	: integer;	Число пикселей датчика
nNumReadOuts	: integer;	Число накоплений (регистраций)
nExposureTime	: integer;	Время накопления (экспозиции)
dwSynchr	: integer;	Способ синхронизации

1.4.2 TCCDUSBExtendParams

Функции [CCD_SetExtendParameters](#) и [CCD_GetExtendParameters](#) используют как параметр переменную типа TCCDUSBExtendParams.

```
TCCDUSBExtendParams =
record
    dwDigitCapacity : DWORD;          // Число бит АЦП.
    nPixelRate : integer;             // Частота работы АЦП. Частота выходного регистра ПЗС-
    датчика. (КГц).
    nNumPixelsH      : integer;        // Число пикселей по горизонтали (число колонок
    ПЗС-матрицы).
    nNumPixelsV      : integer;        // Число пикселей по вертикали (число рядов ПЗС-
    матрицы).
    Reserve1 : DWORD;                 // не используется
    Reserve2 : DWORD;                 // не используется

    nNumReadOuts      : integer;       // Число регистраций.

    sPreBurning        : single;        // Время предварительного обжига в секундах.
                                         // Действительно только для режима синхронизации
    SYNCHR_CONTR_NEG, но не для всех камер. См. Получение информации о ПЗС-камере.
    sExposureTime      : single;        // Время экспозиции. Время одной регистрации в
    миллисекундах.
    sTime2             : single;;       // не используется

    dwSynchr           : DWORD;         // Способ синхронизации.

    bSummingMode        : boolean;      // не используется

    dwDeviceMode        : DWORD;         // Режим камеры. DEVICEMODEA1 – режим матрицы №1.
                                         // DEVICEMODEA2 – режим матрицы №2.
                                         // DEVICEMODES – Спектроскопический режим.
                                         // Матрица работает как одна ПЗС-линейка.
                                         // DEVICEMODESTREAM – камера работает в режиме
    потока.
                                         // DEVICEMODEFASTSCAN – камера работает в режиме
    быстрого
                                         // сканирования.
    nStripCount         : integer;       // Число выделенных полос для спектроскопического
    режима
                                         // ПЗС-матрицы.
    rcStrips            : array [0..MAXSTRIPS-1] of TRect; // Массив полос для
    спектроскопического режима.
                                         //MAXSTRIPS=10;

    Reserve11 : integer;               // не используется

    dwSensitivity       : DWORD;         // Управление чувствительностью. 0 – обычная
    чувствительность.
                                         // 1 – повышенная чувствительность.

    dwProperty : DWORD;               // Свойства прибора. См. Получение информации о ПЗС-
    камере.
    sShutterTime        : Single;        // Время срабатывания электронного затвора в
    миллисекундах.

    Reserve6 : DWORD;                 // не используется
    Reserve7 : DWORD;                 // не используется
    Reserve8 : DWORD;                 // не используется
```

```

Reserve9   : DWORD;      // не используется
Reserve10  : DWORD;      // не используется
end;
```

См. файл CCDUSB021_I.pas , [CCD_GetExtendParameters](#), [CCD_SetExtendParameters](#), [Синхронизация](#), [Получение информации о ПЗС-камере](#).

1.5 Получение информации о камере

Фирма Elmins SP производит ряд камер на основе светочувствительных линеек и матриц. Назначение и, соответственно, возможности у разных типов камер различные. Поэтому, прежде чем начать работу с камерой полезно прочитать ее свойства. Камера должна быть подсоединена к компьютеру, а источник питания должен быть включен.

Свойства прибора кодируются в константе. Её можно получить функциями [CCD_GetParameter](#) ([PRM_DEVICEPROPERTY](#)) или [CCD_GetExtendParameters](#) ([TCCDUSBExtendParams.dwProperty](#)). Единица в соответствующем разряде указывает на наличие свойства.

DP_SYNCHR_CONTR	= \$00000001; // способ синхронизации SYNCHR_CONTR разрешен.
DP_SYNCHR_CONTR_FRS	= \$00000002; // способ синхронизации SYNCHR_CONTR_FRS разрешен.
DP_SYNCHR_CONTR_NEG	= \$00000004; // способ синхронизации SYNCHR_CONTR_NEG разрешен.
DP_SYNCHR_EXT	= \$00000008; // способ синхронизации SYNCHR_EXT разрешен.
DP_SYNCHR_EXT_FRS	= \$00000010; // способ синхронизации SYNCHR_EXT_FRS разрешен.
DP_SENSIT	= \$00000020; // Датчик (первый датчик в случае камеры с двумя линейками) имеет режим повышенной чувствительности.
DP_MODEA2	= \$00000040; // Дополнительный матричный режим камеры.
DP_MODES1	= \$00000080; // Спектроскопический режим ПЗС-матрицы №1
DP_MODES2	= \$00000100; // Спектроскопический режим ПЗС-матрицы №2
DP_PREBURNING	= \$00000200; // Возможность устанавливать время предварительного обжига. Используется только для режима синхронизации SYNCHR_CONTR_NEG.
DP_SHUTER	= \$00000400; // Наличие функции электронного затвора.
DP_CLOCKCONTROL	= \$00000800; // Возможность управлять частотой работы АЦП (аналого-цифрового преобразователя) через поле nPixelRate.
DP_FASTSCAN	= \$00001000; // поддерживает ли конкретная камера режим сканирования Быстрый режим сканирования
DP_STREAMSCAN	= \$00002000; // поддерживает ли конкретная камера режим сканирования DEVICEMODESTREAM
DP_SENSIT2	= \$00004000; // Управление чувствительностью для второй линейки камеры.

Режим синхронизации SYNCHR_NONE разрешен для всех типов камер.

1.6 Синхронизация

Для управления способом синхронизации используются следующие константы:

SYNCHR_NONE = \$1; // Камера работает без синхронизации. Регистрация спектров осуществляется периодически в соответствии с установленными параметрами регистрации
 SYNCHR_CONTR = \$20; // Регистрация осуществляется периодически в соответствии с установленными параметрами. В начале первого накопления формируется положительный синхроимпульс.
 SYNCHR_CONTR_FRS = \$4; // Импульс синхронизации формируется в начале каждого накопления.
 SYNCHR_CONTR_NEG = \$8; // Один синхроимпульс генерируется на все время регистрации. Импульс отрицательной полярности.
 SYNCHR_EXT = \$10; // Камера будет ожидать один синхроимпульс. Первое накопления начнется по возрастающему фронту внешнего синхроимпульса. Все остальные накопления произойдут так быстро, как это возможно. В пределе – без пропусков.
 SYNCHR_EXT_FRS = \$2; // Каждое накопление начнется по возрастающему фронту внешнего синхроимпульса. Камера будет ожидать столько синхроимпульсов, сколько установлено накоплений.

Параметры генерируемого синхроимпульса:

- полярность положительная
- амплитуда, В > 3
- длительность импульса по полувысоте (для режимов SYNCHR_CONTR и SYNCHR_CONTR_FRS), мксек не более 10

Требования к параметрам внешнего синхроимпульса:

- полярность положительная
- амплитуда, В 2.5 – 20
- длительность импульса по полувысоте, мксек 5 – 20
- длительность переднего фронта, мксек не более 1

См. [CCD_SetExtendParameters](#), [CCD_SetParameter](#), [CCD_SetParameters](#)

1.7 Управление параметрами АЦ

В светочувствительные камеры, оснащенные АЦП AD9826, заложена возможность управлять некоторыми параметрами аналого-цифрового преобразователя (АЦП). Признак АЦП AD9826 можно прочитать в snf-файле камеры:

AD9826=1

Управлять можно параметрами: частота, коэффициентом усиления и смещением нулевого уровня.

Следует обратить внимание, что камера с двумя светочувствительными датчиками, например HAMAMATSU S13496X2, использует два канала АЦП. По одному на каждый датчик. Поэтому управлять нужно параметрами (усиление и смещение) для каждого из них.

Можно или нет управлять частотой работы АЦП, можно узнать прочитав свойства камеры. См.

[Получение информации о ПЗС-камере](#). DP_CLOCKCONTROL должно быть 1.

DP_CLOCKCONTROL = \$00000800; // Возможность управлять частотой работы АЦП через поле nPixelRate.

Не все камеры имеют возможность управлять частотой АЦП. Частоты не могут быть произвольным. Функция [CCD_GetParameter](#) вернет допустимое установленное значение.

Для установки используется функция DLL [CCD_SetParameter](#) с параметрами:

PRM_PIXELRATE	= 2	— Частота работы АЦП. Частота выходного регистра ПЗС-датчика (КГц).
PRM_KAmpRed	= \$1022;	— Усиление АЦП. Первый канал
PRM_OffsetRed	= \$1034;	— Смещение нулевого уровня АЦП. Первый канал
PRM_KAmpGreen	= \$1023;	— Усиление АЦП. Второй канал
PRM_OffsetGreen	= \$1035;	— Смещение нулевого уровня АЦП. Второй канал

Коэффициент усиления можно выставлять от 1 до 6.

В функцию [CCD_SetParameter](#) передается расчетное число:

$CCD_AD9826_KRed2000 := 63.0 - (6.0 / Value - 1.0) * 63.0 / 5.0;$

Где Value - Коэффициент усиления АЦП от 1 до 6. Таким образом расчетное число должно быть целочисленным в диапазоне от 0 до 63.

Смещение нулевого уровня выставляется непосредственно в битах АЦП. Может быть в диапазоне от -255 до 255.

Для второго канала расчет аналогичный.

Пример:

```

if CCD_SetParameter(ID, PRM_KAmpRed, CCD_AD9826_KRed2000) then
  if CCD_SetParameter(ID, PRM_OffsetRed, CCD_AD9826_OffsetRed2000) then
    if CCD_SetParameter(ID, PRM_KAmpGreen, CCD_AD9826_KGreen2000) then
      if CCD_SetParameter(ID, PRM_OffsetGreen, CCD_AD9826_OffsetGreen2000) then
        Result := true;

```

Управлять АЦП (усиление и смещение) можно и через snf-файл камеры. В файле должны быть строки:

```

KRed2000=1
KGreen2000=1
KBlue2000=1
OffsetRed2000=100
OffsetGreen2000=100
OffsetBlue2000=0

```

KRed2000 — расчетное число для коэффициента усиления АЦП. Первый канал.
 KGreen2000 — расчетное число для коэффициента усиления АЦП. Второй канал.
 KBlue2000 — расчетное число для коэффициента усиления АЦП. Третий канал. Не используется.

OffsetRed2000=100 — Смещение нулевого уровня АЦП. Первый канал
OffsetGreen2000=100 — Смещение нулевого уровня АЦП. Второй канал
OffsetBlue2000=0 — Смещение нулевого уровня АЦП. Третий канал. Не используется.

Частотой АЦП можно управлять только через функцию [CCD_SetParameter](#).

1.8 Режим быстрого сканирования

Реализован для камер:

U2C-14S415,

U2C-10M403

Для вышеуказанных камер, режим DEVICEMODEFASTSCAN позволяет добиться максимальной скорости регистрации для практически любого числа кадров.

За один вызов функции CCD_StartMeasuring регистрируется только один кадр. Функция сразу же возвращает управление. В это время драйвер регистрирует один кадр. Указатель на массив передается в функцию CCD_GetData.

Определить, поддерживает ли конкретная камера режим сканирования DEVICEMODEFASTSCAN, можно запросив свойства камеры, как на примере ниже.

```
if CCD_GetExtendParameters ( ID , Prms ) then
begin
  if (Prms.dwProperty and DP_FASTSCAN) <> 0 then
  begin
    // Можем работать в режиме DP_FASTSCAN
  end;
end;
```

[Пример.](#)

1.9 Режим сканирования DEVICEMODESTREAM

Реализован для камер:

U2C-14S5

В этом режиме камера работает непрерывно, пока не будет вызвана функция [CCD_CameraReset](#). Используются два попеременно работающих буфера. Размер каждого буфера устанавливается в поле [nNumReadOuts](#). Память под них выделяется и уничтожается внутри DLL. Размер буферов ограничен:

Камера	U2C-14S5
Максимальный размер буферов	6 кадров

Определить, поддерживает ли конкретная камера режим сканирования DEVICEMODESTREAM, можно запросив свойства камеры, как на примере ниже.

```

if CCD_GetExtendParameters( ID , Prms ) then
begin
  if (Prms. dwProperty and DP\_STREAMSCAN) <> 0 then
  begin
    // Можем работать в режиме DEVICEMODESTREAM
  end;
end;

```

[Простейший пример регистрации 10000 кадров.](#)

```

//Переводим камеру в режим DEVICEMODESTREAM
if not CCD_SetParameter(ID,PRM_DEVICEMODE, DEVICEMODESTREAM) then Exit;

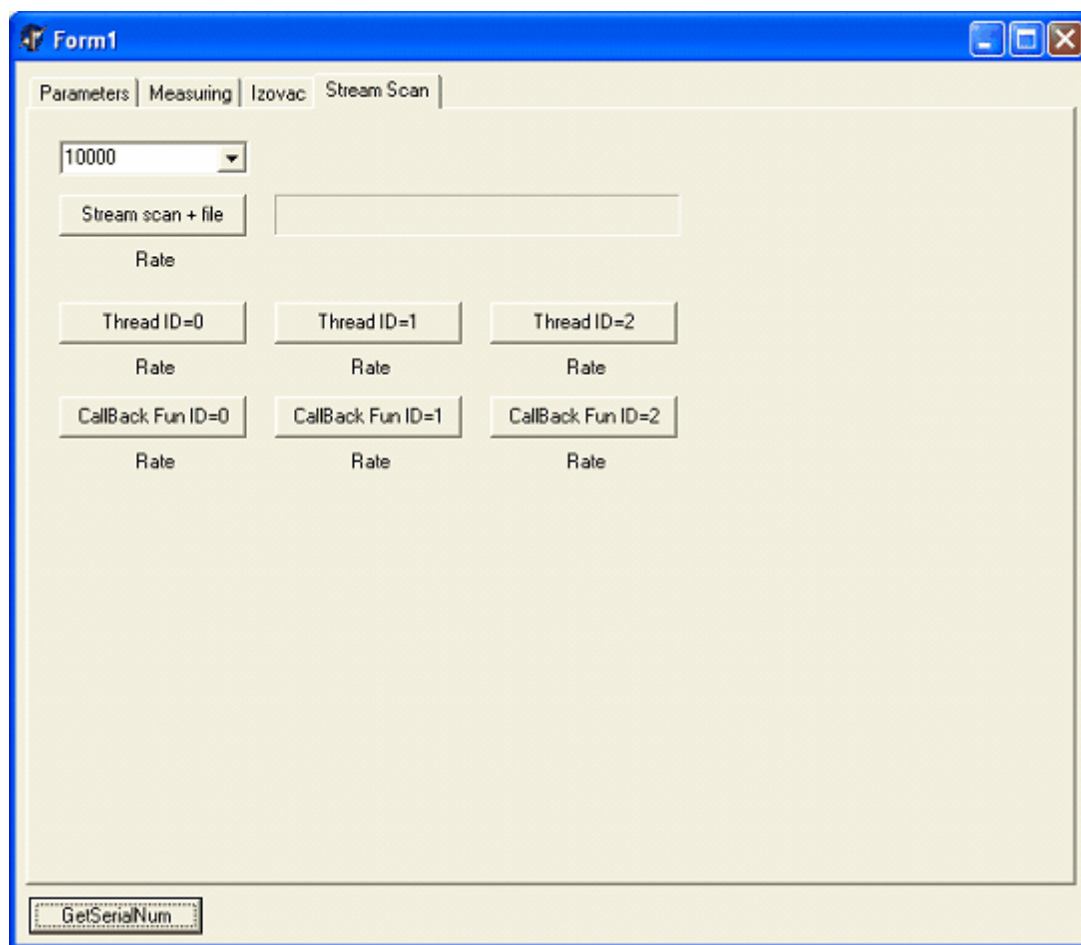
if not CCD_GetParameter(ID,PRM_NUMPIXELSH, Ncol) then Exit;
if not CCD_GetParameter(ID,PRM_NUMPIXELSV, Nrow) then Exit;
if not CCD_GetParameter(ID,PRM_READOUTS, NF) then Exit;

N := 10000;
N := Ceil(N/Round(Nf))*Round(Nf);
Block := Round(Np*Nf)*SizeOf(DWORD);
Blockdiv4 := Block div SizeOf(DWORD);;
NBlock := Round (N / NF);

// Выделяем память для кадров. pData указатель на массив DWORD-ов!!!
GetMem(pData, Round(Np* SizeOf(DWORD)*N));
try
  if CCD_HitTest(ID) then
  begin
    if CCD_InitMeasuring(ID) then
    begin
      if CCD_StartMeasuring (ID) then
      repeat // Повторяем только одну функцию CCD_GetData и постоянно переносим
        // указатель на новое место.
        Res := CCD_GetData(ID,Addr(pData[*Blockdiv4]));
        if Res then
        Inc(f);
      until (F>=NBlock );
    finally
      CCD_CameraReset(ID); // Останавливаем камеру!!!
    end;
    ... // некоторые операции с кадрами
  finally
    FreeMem(pData); // Освобождаем память.
  end;
end;// if CCD_HitTest(ID) then

```

В программном примере SDK можно видеть специальную вкладку, демонстрирующую работу камеры в режиме DEVICEMODESTREAM.



Соответствующие функции снабжены комментариями в исходном коде программы.

Продемонстрированы следующие приемы работы:

1. Работа камеры в режиме DEVICEMODESTREAM, данные записываются в файл. Для скорости используется file-mapping object. Работает только одна камера. Кнопка «Stream Scan + file» в примере.
2. Работа камеры в режиме DEVICEMODESTREAM, данные записываются в файл. Для скорости используется file-mapping object. Независимо работают до 3-х камер. Для каждой камеры запускается собственная нить (Thread). Кнопки «Thread ID=0», «Thread ID=1» и «Thread ID=2» в примере.
3. Работа камеры в режиме DEVICEMODESTREAM, данные записываются в файл. Для скорости используется file-mapping object. Независимо работают до 3-х камер. Используются callback функции. Кнопки «CallBack Fun ID=0», «CallBack Fun ID=1» и «CallBack Fun ID=2» в примере.

В SDK включен дополнительный пример StreamScanDemo.

Он демонстрирует работу одной камеры в режиме DEVICEMODESTREAM и влияние прорисовки части кадра на скорость регистрации кадров.

Для того, что бы получить максимальную скорость регистрации (в случае камеры U2C-14S5 1000 кадров в секунду) необходимо выполнение некоторых требований.

1. Производительность компьютера должна обеспечивать быстрый ввод данных по USB 2.0

(High Speed) интерфейсу. Компьютер с конфигурацией Intel Celeron 1200 МГц, 256 Мбайт оперативной памяти обеспечивает такую скорость.

2. Наличие достаточного объема оперативной памяти для размещения всех кадров. Скорость сканирования может падать при работе с динамическим выделением блока памяти. Если недостаточно свободной оперативной памяти операционная система эмулирует память на жестком диске, что приводит к снижению скорости. При использовании file-mapping object максимальная скорость будет достигнута, если свободной памяти достаточно для размещения всего блока данных. Важно помнить, что операционная система и загруженные программы занимают оперативную память.

3. Нужно освободить процессорное время для ввода данных по USB 2.0 (High Speed) интерфейсу. Загруженные, хотя и не используемые, программы или процессы «занимают» время процессора и его производительности может не хватить для реализации максимальной скорости регистрации. Отображение кадра или обработка данных, допустимая между вызовами функции CCD_GetData, не должны очень длительными.

4. К компьютеру не должны быть подсоединены другие USB 2.0 (High speed) устройства.

1.10 Пример 1

Пример 1.

```
if CCD_HitTest(ID) then
  if CCD_InitMeasuringData(ID,pData) then
    if CCD_StartWaitMeasuring (ID) then
      begin
        for i:=0 to (Round(Np*NF)-1) do
          ...;
        end;
```

1.11 Пример 2

Пример2.

```
if CCD_HitTest(ID) then
  if CCD_InitMeasuringData(ID,pData) then
    if CCD_StartMeasuring (ID) then
      begin
        repeat
          if not CCD_GetMeasureStatus (ID,status)
            then Exit;
        until status = STATUS_DATA_READY;
        for i:=0 to (Round(Np*NF)-1) do
          ...;
        end;
```

1.12 Пример 3

Пример для режима сканирования [DEVICEMODESTREAM](#).

```
//Переводим камеру в режим DEVICEMODESTREAM
if not CCD_SetParameter(ID,PRM_DEVICEMODE, DEVICEMODESTREAM) then Exit;

if not CCD_GetParameter(ID,PRM_NUMPIXELSH, Ncol) then Exit;
if not CCD_GetParameter(ID,PRM_NUMPIXELSV, Nrow) then Exit;
```



```

if not CCD\_GetParameter(ID,PRM_READOUTS, NF) then Exit;

N := 10000;
N := Ceil(N/Round(Nf))*Round(Nf);
Block := Round(Np*Nf)*SizeOf(DWORD);
Blockdiv4 := Block div SizeOf(DWORD);
NBlock := Round (N / NF);

// Выделяем память для кадров. pData указатель на массив DWORD-ов!!!
GetMem(pData, Round(Np* SizeOf(DWORD)*N));
try
  if CCD\_HitTest(ID) then
  begin
    if CCD\_InitMeasuring(ID) then
    begin
      if CCD\_StartMeasuring (ID) then
        repeat // Повторяем только одну функцию CCD_GetData и постоянно переносим
              // указатель на новое место.
              Res := CCD\_GetData(ID,Addr(pData[f*Blockdiv4]));
              if Res then
                Inc(f);
            until (F>=NBlock );
        finally
          CCD\_CameraReset(ID); // Останавливаем камеру!!!
        end;
        ... // некоторые операции с кадрами
        finally
          FreeMem(pData); // Освобождаем память.
        end;
      end;
    end;
  end;
end;

```

1.13 Пример 4

Пример для режима сканирования режим [DEVICEMODEFASTSCAN](#).

```

Const N = 50;
Var pData : array [0..N-1] of PDWORDArr;
    Np, Nrow, Ncol : single;
    f, i : Integer;
    StTime, CurTime : Cardinal;
begin
  CCD_GetParameter(ID,PRM_NUMPIXELSH, Ncol);
  CCD_GetParameter(ID,PRM_NUMPIXELSV, Nrow);
  Np := Ncol * Nrow;
  for i := 0 to N-1 do
    GetMem(pData[i], Round(Np*SizeOf(DWORD)));
    CCD_SetParameter(ID,PRM_DEVICEMODE, DEVICEMODEFASTSCAN);

  try
    if CCD_HitTest(ID) then
    begin
      if CCD_InitMeasuring(ID) then
      StTime := GetTickCount;
      if CCD_StartMeasuring (ID) then
        for f := 0 to (N-1) do
        begin
          if CCD_GetData(ID,pData[f]) then
          begin
            if f<>(N-1) then

```

```
        if CCD_StartMeasuring (ID) then
        begin
            Series1.Clear;
            for i:=0 to (Round(1000)-1) do
                Series1.AddXY(i,pData[f][i],'',clTeeColor);
            Application.ProcessMessages;
        end else
            Exit;
        end
        else Exit;
    end;
    CurTime := GetTickCount;
    lbRate.Caption := FormatFloat ('0.###', N/((CurTime-StTime)/1000))+ ' frame/sec';
    CCD_CameraReset(ID);
    Series1.Clear;
    for i:=0 to (Round(Ncol)-1) do
        Series1.AddXY(i,pData[(N-1)][i],'',clTeeColor);
    Application.ProcessMessages;
    end;
finally
    for i := 0 to N-1 do
        FreeMem(pData[i]);
    end; // try ... finally
end;
```