B181.dll.

Руководство программиста.

Всего одна функция.
BOOL IOControl(DWORD dwIoControlCode, CDeviceID& DeviceID, void* pParameter);
Возвращает TRUE, если операция выполнена.

DWORD DwIoControlCode это код функции, один из:

IOCTRL_Initialize
IOCTRL_Deinitialize
IOCTRL_ADCConfig
IOCTRL_ADCReadData
IOCTRL_DACConfig
IOCTRL_DACWriteData
IOCTRL_LAConfig
IOCTRL_LAReadData
IOCTRL_LGConfig
IOCTRL_LGWriteData
IOCTRL_CmnConfig
IOCTRL_CmnControl
IOCTRL_CmnReadStatus
IOCTRL_InitClbrInfo

DeviceID это идентификатор устройства, полученный при вызове IOCTRL_Initialize.

pParameter – указатель на структуру данных, специфическую для каждой подфункции.

1. IOCTRL_Initialize

Должна быть вызвана перед началом работы. 3-ий параметр не имеет значения. Инициализирует идентификатор устройсва, который затем будет использоваться другими функциями.

2. IOCTRL_Deinitialize

Должна быть вызвана перед завершением работы. 3-ий параметр не имеет значения.

3. IOCTRL_ADCConfig

Конфигурация АЦП.

```
struct ADCConfigStruct
{
    BOOL    m_bRecorder;
    UCHAR   m_cChannels;
    UCHAR   m_cRunSource;
    UCHAR   m_cClockSource;
    WORD    m_wWorkLen;

}
```

m_bRecorder
1 – АЦП работает в режиме самописца, когда заполнив одну память на глубину WorkLen, начинается заполнение второй памяти.
При этом устанавливается флаг, что соответствующая память заполнена.
Этот режим подразумевает, что АЦП назначены 2 памяти. Останов производится программно;
0 – обычный режим.
m_cChannels
Выбор каналов АЦП, из которых будут производиться выборки, 4 бита. Каждый бит соответствует каналу.
m_cRunSource Источник запуска АЦП (см. RunSources)
m_cClockSource Источник тактирования (см.ClockSources)
m_wWorkLen
Количество выборок за проход. **Внимание!** Под выборкой понимается выборка всех активных каналов. Т.е. если активны все 4 канала, то потребуется памяти WorkLen*4.
Задавая этот параметр, следует иметь в виду, какая память назначена для данного устройства, чтобы не превысить границу. Не рекомендую ставить меньше 4. 16 бит.

4. IOCTRL_ADCReadData

Чтение данных АЦП.

```
ReadADCStruct
{
    ULONG* m_pData;
    UCHAR m_cDataSource;
    WORD  m_wCount;
}
```

m_pData Указатель на массив, куда вычитаются данные.
m_cDataSource Источник данных:
DtSrc_DirectChannels – данные однократного чтения АЦП. m_wCount не имеет значения. Вычитывается всегда 4 данных. Действительных столько, сколько активных каналов выбрано. Чтение следует начинать только после того, как установился флаг ADC_DataReady. Его использование не требует наличия памяти вообще.
DtSrc_Mem0 – память 0 АЦП. Адрес в памяти инкрементируется автоматически после чтения
DtSrc_Mem1 – память 1 АЦП. Адрес в памяти инкрементируется автоматически после чтения
m_wCount Количество отсчетов данных для чтения.

5. IOCTRL_DACConfig

Конфигурация ЦАП.

DACConfigStruct

```
{
    BOOL    m_bSingleCycle;
    BOOL    m_bEndlessGenerator;
    UCHAR   m_cChannels;
    UCHAR   m_cRunSource;
    UCHAR   m_cClockSource;
    WORD    m_wWorkLen;
}
```

m_bSingleCycle

1 – Однократный проход, бит m_bEndlessGen не имеет значения. За время работы, данные один раз будут выбраны из памяти 0, после чего процесс остановится.

0 – циклическая работа, зависит от бита m_bEndlessGen

m_bEndlessGenerator

1 – Режим бесконечного генератора. Как и самописец в АЦП, только наоборот.

0 – Обычный циклический режим, при котором данные по кругу выбираются из памяти 0.

Остальное как у АЦП.

6. IOCTRL_DACWriteData

Запись данных в ЦАП

WriteDACCStruct

```
{
    UCHAR m_cDataDestination;
    ULONG* m_pData;
    WORD   m_wCount;
    UCHAR m_cChannel;
    BOOL  m_bLDAC_;
}
```

m_cDataDestination – куда пишутся данные:

DtSrc_DirectChannels – данные пишутся непосредственно в канал ЦАП. При использовании этого типа записи в поле m_cChannel (2 бита) должен быть заполнены поля m_cChannel, m_bLDAC_. Поле m_wCount значения не имеет.

DtSrc_Mem0 – память 0 ЦАП. Адрес в памяти инкрементируется автоматически после записи

DtSrc_Mem1 – память 1 ЦАП. Адрес в памяти инкрементируется автоматически после записи

m_pData Указатель на данные.

m_wCount – количество отсчетов для записи в память.

m_cChannel – номер канала , в который производится запись.

m_bLDAC_

0 – результат появится на выходе ЦАП сразу после завершения операции

1 – результат появится на выходе выбранного канала после записи данных, в которых этот бит 0. Используется когда необходимо, чтобы все каналы обновились одновременно

7. IOCTRL_LAConfig

Конфигурация анализатора.

LAConfigStruct

```
{
    BOOL    m_bRecorder;
    UCHAR   m_cRunSource;
    UCHAR   m_cClockSource;
    WORD    m_wWorkLen;
}
```

Как и у АЦП, только один канал и всегда выбран.

8. IOCTRL_LAReadData

Чтение данных анализатора.

ReadLAStruct

```
{
    ULONG* m_pData;
    WORD m_wCount;
    UCHAR m_cDataSource;
}
```

Все как обычно, только **m_cDataSource** еще может принимать значение **DtSrc_Trigger**.

Как и при DtSrc_DirectChannel, m_wCount не имеет значения, и вычитывается одно данное.

Это регистр триггера. Обнуляется при запуске. Его использование не требует наличия памяти вообще.

Если за время, пока идет выборка (WorkLen отсчетов или пока не остановят в режиме Recorder) по соответствующему каналу был переход 0–>1 то бит устанавливается в 1.

9. IOCTRL_LGConfig

Конфигурация генератора.

LGConfigStruct

```
{
    BOOL    m_bSingleCycle;
    BOOL    m_bEndlessGenerator;
    UCHAR   m_cRunSource;
    UCHAR   m_cClockSource;
    WORD    m_wWorkLen;
}
```

Как и у ЦАП, только 1 канал и всегда выбран.

10. IOCTRL_LGWriteData

Запись данных в генератор.

```
WriteLGStruct
{
        ULONG* m_pData;
        WORD m_wCount;
        UCHAR m_cDataDestination;
}
```

Как и у ЦАП, только данные, записываемые в каналы непосредственно, имеют ту же структуру, что и в память.

11. IOCTRL_CmnConfig

Конфигурация всей системы.

```
CmnConfigStruct
{
        UCHAR m_cExtMem0Source;
        UCHAR m_cExtMem1Source;
        UCHAR m_cIntMem0Source;
        UCHAR m_cIntMem1Source;
        BOOL m_bLD3_0Direction;
        BOOL m_bLD7_4Direction;
        BOOL m_bLD11_8Direction;
        BOOL m_bLD15_12Direction;
        WORD m_wInternalScale0;
        WORD m_wInternalScale1;
}
```

m_cExtMemXSource
Для каждой памяти назначается прибор, с которым она работает. 1,0 –внешние памяти (см. MemSources).
m_bLDX_YDirection
Направление пинов с Y по X логического порта (см LP_Direction).
m_wInternalScaleX
Временная развертка внутреннего генератораX. Изменение на 1 соответствует изменению временной развертки на 20нс. **Внимание!** Если используется в качестве тактирования для АЦП и ЦАП, то значение не может быть меньше 500, для LG и LA – не меньше 5.

12. IOCTRL_CmnControl

Управление всей системой.

```
CmnControlStruct
{
        BOOL m_bADC_StartSingleRead;
        BOOL m_bADC_SW_Stop;
        BOOL m_bADC_Run;
        BOOL m_bADC_Reset;
        BOOL m_bDAC_MemCntClear;
        BOOL m_bDAC_SW_Stop;
        BOOL m_bDAC_Run;
        BOOL m_bDAC_Reset;
        BOOL m_bLA_SW_Stop;
        BOOL m_bLA_Run;
        BOOL m_bLA_Reset;
        BOOL m_bLG_MemCntClear;
        BOOL m_bLG_SW_Stop;
        BOOL m_bLG_Run;
        BOOL m_bLG_Reset;
}
```

m_bADC_StartSingleRead Запуск одиночной выборки АЦП.
При этом произойдет выборка всех активных каналов.
Данные можно забирать после установки соответствующего бита статусного регистра.
m_bADC_SW_Stop Останов АЦП, работающего в режиме самописца.
m_bADC_Run Запуск АЦП, используется, если выбран в качестве источника запуска АЦП.
m_bADC_Reset Инициализация АЦП. Должна проводится перед каждым запуском.
Если выбран непрограммный источник запуска, то запуск сработает только после инициализации.
m_bDAC_MemCntClear Обнуление счетчика адреса памяти ЦАП. Используется, чтобы проинициализировать память перед чтением данных ЦАП после программного останова.
В остальных случаях инициализация не требуется.

Остальные биты аналогичны уже описанным. Останов ЦАП и LG производится программно.
Только в режиме SingleCycle не требуется останов.

13. IOCTRL_CmnReadStatus

Статусный регистр.

```
CmnStatusStruct
{
    BOOL m_bADC_DataReady;
    BOOL m_bADC_Mem1Full;
    BOOL m_bADC_Mem0Full;
    BOOL m_bDAC_Mem1Empty;
    BOOL m_bDAC_Mem0Empty;
    BOOL m_bLA_Mem1Full;
    BOOL m_bLA_Mem0Full;
    BOOL m_bLG_Mem1Empty;
    BOOL m_bLG_Mem0Empty;
}
```

m_bADC_DataReady Готовность данных АЦП после однократного чтения.

m_bADC_Mem1Full

Память 1 заполнена и начато заполнение памяти 0. Используется только в режиме самописца.
Чтение данных памяти 1 производится, начиная с этого момента.
Внимание! При работе в режиме самописца следить за тем, чтобы успеть вычитать данные памяти пока установлен бит ее заполненности. Т.е. вычитав данные необходимо убедиться, что бит еще не очистился. Иначе результаты измерений недействительны.

m_bADC_Mem0Full

память 0 заполнена и начато заполнение памяти 1,если установлен режим самописца.
Для режима самописца: чтение данных памяти 0 производится, начиная с этого момента.
Внимание! При работе в режиме самописца следить за тем, что бы успеть вычитать данные памяти пока установлен бит ее заполненности. Т.е. вычитав данные необходимо убедиться, что бит еще не очистился. Иначе результаты измерений недействительны.
Для обычного режима: по этому флагу начинается вычитка данных и запуск очередного измерения.
Остальные биты аналогичны.

14. IOCTRL_InitCIbrInfo

Инициализация калибровочных данных.

```
CIbrInfoStruct {
    double m_ADCZeroCode[4];
    double m_ADCCoef[4];
    WORD m_DACOffset;
    WORD m_DACCodePlus10[4];
    WORD m_DACCodeMinus10[4];
}
```

m_ADCZeroCode[ch] Код канала ch АЦП, соответствующий 0 В (double для повышения точности);
m_ADCCoef[ch] Коэффициент преобразования канала ch АЦП (Вольт на код);
m_DACOffset Количество кодов ЦАП по краям диапазона 0-4095, попадающие в зону нелинейной передаточной характеристики (загибы по краям); не используется (может быть не проинициализировано).
m_DACCodePlus10[ch] Код канала ch ЦАП, соответствующий выходному напряжения +10 В;
m_DACCodeMinus10[ch] Код канала ch ЦАП, соответствующий выходному напряжения –10 В;

Для вычисления значения напряжения из кода АЦП *ADCCode* (14-bit, дополнительный код):

```
U = ( ( ( short(ADCCode << 2 ) ) >> 2 ) – m_ADCZeroCode[ch]) * m_ADCCoef[ch]
```

или

```
U = ( ( ( ADCCode ^ 0x2000 ) – 8192 ) – m_ADCZeroCode[ch]) * m_ADCCoef[ch]
```

кому как нравится.

Для вычисления кода ЦАП, соответствующего необходимому напряжению *V*:

```
double Coef = (m_DACCodePlus10[ch] - m_DACCodeMinus10[ch]) / 20.0;
DACCode = max ( 0, min ( 4095, ( short ) floor ( 0.5 + (V + 10) * Coef + m_DACCodeMinus10[ch] ) ) );
```

MemSources. Всего есть 4 памяти. 2 – это внешние памяти каждая 64Кх16, и 2 внутренних, каждая по 512х16. Каждому устройству (АЦП, ЦАП, LA, LG) может ,быть назначено от 0 до 2 памятей. Если прибору не назначена память, то он может использоваться только для однократных чтений/записей в зависимости от прибора. LA может также быть использован как Триггер. Прибору назначается 2 памяти, если он используется как самописец / бесконечный генератор.
Если прибор запущен на очередной проход, то однократные операции запрещены (кроме LA и LG).
Каждой памяти прописывается, с каким прибором она работает. Если прибор использует 1 память, то памяти должен быть сопоставлен в соответствие источник с индексом 0 (ADC0….). **Внимание!!!** Не должно быть 2-х памятей, назначенных на одно и тоже место.
Допустимые значения:

- MemSrc_ADC0
- MemSrc_ADC1
- MemSrc_DAC0
- MemSrc_DAC1
- MemSrc_LA0
- MemSrc_LA1
- MemSrc_LG0
- MemSrc_LG1

RunSources: Для каждого прибора существует 5 источников запуска. 4 из них внешние и берутся с соответствующих контактов разъема логического порта. На пользователе лежит ответственность за то, чтобы при внешнем источнике импульс пришел бы после инициализации прибора.
Принимаемые значения:

- RunSrc_SW
- RunSrc_LD0
- RunSrc_LD2
- RunSrc_LD4
- RunSrc_LD6

ClockSources: Для каждого прибора существует 6 источников запуска. 4 из них внешние и берутся с соответствующих контактов разъема логического порта. **Внимание!** При использовании внутренних генераторов, следует убедиться в том, что данный прибор сможет работать на такой частоте (см. 12. IOCTRL_CmnControl). При использовании внешнего источника пользователь должен обеспечить нужное количество импульсов для тактирования WorkLen выборок.
Принимаемые значения:

- ClockSrc_IntGen0
- ClockSrc_IntGen1
- ClockSrc_LD1
- ClockSrc_LD3
- ClockSrc_LD5
- ClockSrc_LD7

LP_Direction: Чтобы на пине появлялись данные генератора, для четверки, в которую он входи должно быть выбрано направление LDDirection_LG. Если пин используется анализатором, то LDDirection_LA

Работа устройства.

Перед началом работы производится конфигурация: выбираются источники запуска и тактирования для всех используемых приборов, назначаются приборы для каждой памяти, направление логического порта, задаются частоты внутренних генераторов, конфигурируется каждый прибор (режим самописца/обычный, выбираются используемые каналы, записывается WorkLen). Конфигурация общих параметров таких, как, например, внутренние генераторы, должна производиться при остановленных приборах. Конфигурация индивидуальных параметров прибора должна происходить при неработающем приборе.

Цикл измерения для ЦАП и LA в обычном режиме состоит из:
1. Инициализация.
2. Запуск, если выбран программный запуск.
3. Ожидание флага готовности (Mem0Full).
4. Чтение данных памяти.

В режиме самописца:
1. Инициализация.
2. Запуск, если выбран программный запуск.
3. Ожидание флага готовности очередной памяти (MemXFull).
4. Чтение данных памяти.
5. Проверка, что успели вычитать данные до заполнения следующей памяти.
6. Переход на п.3.

И так пока не остановят.

Режим триггера LA: аналогично обычному режиму, только вместо данных из памяти, вычитывается регистр триггера.

Режим однократной выборки АЦП:
1. Запуск одиночной выборки АЦП
2. Ожидание готовности данных (ADC_DataReady).
3. Чтение регистра данных.

Режим однократной выборки LA:
Просто вычитать регистр данных с соответствующим источником данных.

Цикл для АЦП и LG в обычном циклическом режиме состоит из:
1. Запись данных в память.
2. Инициализация.
3. Запуск, если выбран программный запуск.
4. Программный останов, когда понадобится.

Режим однократного прохода:
1. Запись данных в память.
2. Инициализация.
3. Запуск, если выбран программный запуск.

И все.

Режим бесконечного генератора:
1. Запись данных в обе памяти.
2. Инициализация.
3. Запуск, если выбран программный запуск.
4. Ожидание флага MemXEmpty.
5. Запись данных в эту память.
6. Проверка: успели или нет.
7. Переход на пункт 4.

И так до программного останова.

Режим однократной записи в ЦАП:
Просто правильно записать данные в структуру и произвести команду записи.

Режим однократной записи в LG:
Просто записать данные, указав источник.