

USBADC10 API

1.0.0

Создано системой Doxygen 1.8.13

Содержание

1	Список файлов	1
1.1	Файлы	1
2	Файлы	1
2.1	Файл <code>usbadc10.h</code>	1
2.1.1	Подробное описание	3
2.1.2	Структуры данных	3
2.1.3	Макросы	3
2.1.4	Типы	4
2.1.5	Функции	4
	Алфавитный указатель	9

1 Список файлов

1.1 Файлы

Полный список документированных файлов.

<code>usbadc10.h</code>	
Usbadc10 API	1

2 Файлы

2.1 Файл `usbadc10.h`

usbadc10 API

```
#include <stdint.h>
#include <wchar.h>
```

Структуры данных

- `struct usbadc10_get_identity_information_t`
- `struct usbadc10_get_conversion_raw_t`
- `struct usbadc10_get_conversion_t`

Макросы

- `#define USBADC10_BUILDER_VERSION_MAJOR 0`
- `#define USBADC10_BUILDER_VERSION_MINOR 10`
- `#define USBADC10_BUILDER_VERSION_BUGFIX 9`
- `#define USBADC10_BUILDER_VERSION_SUFFIX ""`
- `#define USBADC10_BUILDER_VERSION "0.10.9"`
- `#define USBADC10_URPC_API_EXPORT __attribute__((visibility("default")))`
- `#define USBADC10_URPC_CALLING_CONVENTION`
- `#define device_undefined (-1)`
- `#define result_ok 0`
- `#define result_error (-1)`
- `#define result_not_implemented (-2)`
- `#define result_value_error (-3)`
- `#define result_nodvice (-4)`

Уровень логирования

- `#define LOGLEVEL_ERROR 0x01`
- `#define LOGLEVEL_WARNING 0x02`
- `#define LOGLEVEL_INFO 0x03`
- `#define LOGLEVEL_DEBUG 0x04`

Определения типов

- `typedef int device_t`
- `typedef int result_t`
- `typedef void(USBADC10_URPC_CALLING_CONVENTION *usbadc10_logging_callback_t)(int loglevel, const wchar_t *message, void *user_data)`

Функции

- `USBADC10_URPC_API_EXPORT void USBADC10_URPC_CALLING_CONVENTION usbadc10_logging_callback_stderr_wide(int loglevel, const wchar_t *message, void *user_data)`
- `USBADC10_URPC_API_EXPORT void USBADC10_URPC_CALLING_CONVENTION usbadc10_logging_callback_stderr_narrow(int loglevel, const wchar_t *message, void *user_data)`
- `USBADC10_URPC_API_EXPORT void USBADC10_URPC_CALLING_CONVENTION usbadc10_set_logging_callback(usbadc10_logging_callback_t cb, void *data)`
- `USBADC10_URPC_API_EXPORT device_t USBADC10_URPC_CALLING_CONVENTION usbadc10_open_device(const char *uri)`
- `USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_libversion(char *lib_version)`
- `USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_get_identity_information(device_t handle, usbadc10_get_identity_information_t *output)`
- `USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_get_conversion_raw(device_t handle, usbadc10_get_conversion_raw_t *output)`
- `USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_get_conversion(device_t handle, usbadc10_get_conversion_t *output)`
- `USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_close_device(device_t *handle_ptr)`
- `USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_get_profile(device_t handle, char **buffer, void *(*allocate)(size_t))`
- `USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_set_profile(device_t handle, char *buffer)`

2.1.1 Подробное описание

`usbadc10` API

2.1.2 Структуры данных

2.1.2.1 `struct usbadc10_get_identity_information_t`

Поля структур

<code>uint16_t</code>	<code>BootloaderBugfix</code>	Номер ревизии загрузчика.
<code>uint8_t</code>	<code>BootloaderMajor</code>	Мажорный номер версии загрузчика.
<code>uint8_t</code>	<code>BootloaderMinor</code>	Минорный номер версии загрузчика.
<code>uint8_t</code>	<code>ControllerName[16]</code>	Пользовательское имя контроллера. Может быть установлено пользователем с помощью отдельной команды.
<code>uint16_t</code>	<code>FirmwareBugfix</code>	Номер ревизии прошивки.
<code>uint8_t</code>	<code>FirmwareMajor</code>	Мажорный номер версии прошивки.
<code>uint8_t</code>	<code>FirmwareMinor</code>	Минорный номер версии прошивки.
<code>uint16_t</code>	<code>HardwareBugfix</code>	Номер ревизии платы.
<code>uint8_t</code>	<code>HardwareMajor</code>	Основной номер версии железа.
<code>uint8_t</code>	<code>HardwareMinor</code>	Второстепенный номер версии железа.
<code>uint8_t</code>	<code>Manufacturer[16]</code>	Имя производителя. Устанавливается производителем.
<code>uint8_t</code>	<code>ProductName[16]</code>	Название продукта. Устанавливается производителем.
<code>uint8_t</code>	<code>Reserved[8]</code>	Значение данного поля не должно использоваться в прикладном ПО. Для обеспечения совместимости с другими устройствами не изменяйте значение этого поля.
<code>uint32_t</code>	<code>SerialNumber</code>	Серийный номер изделия.

2.1.2.2 `struct usbadc10_get_conversion_raw_t`

Поля структур

<code>uint16_t</code>	<code>data[10]</code>	Массив результатов измерений с 10 каналов 12-битного АЦП. 0 соответствует минимальному напряжению (GND), 4095 соответствует опорному напряжению АЦП (3,3 В).
-----------------------	-----------------------	--

2.1.2.3 `struct usbadc10_get_conversion_t`

Поля структур

<code>uint16_t</code>	<code>data[10]</code>	Массив результатов измерений с 10 каналов. Единицы измерения: 100 мкВ. Например, полученное значение 123 соответствует напряжению 12,3 мВ.
-----------------------	-----------------------	--

2.1.3 Макросы

2.1.3.1 LOGLEVEL_DEBUG

```
#define LOGLEVEL_DEBUG 0x04
```

Уровень логирования - отладка

2.1.3.2 LOGLEVEL_ERROR

```
#define LOGLEVEL_ERROR 0x01
```

Уровень логирования - ошибка

2.1.3.3 LOGLEVEL_INFO

```
#define LOGLEVEL_INFO 0x03
```

Уровень логирования - информация

2.1.3.4 LOGLEVEL_WARNING

```
#define LOGLEVEL_WARNING 0x02
```

Уровень логирования - предупреждение

2.1.4 Типы

2.1.4.1 usbadc10_logging_callback_t

```
typedef void(USBADC10_URPC_CALLING_CONVENTION * usbadc10_logging_callback_t) (int loglevel, const
wchar_t *message, void *user_data)
```

Прототип функции обратного вызова для логирования.

Аргументы

loglevel	- Уровень логирования.
message	- Сообщение.

2.1.5 Функции

2.1.5.1 usbadc10_close_device()

```
USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_close_device
(
    device_t * handle_ptr )
```

Закрывает устройство.

Аргументы

<code>handle_ptr</code>	- Идентификатор устройства.
-------------------------	-----------------------------

2.1.5.2 `usbadc10_get_conversion()`

```
USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_get_↔
conversion (
    device_t handle,
    usbadc10_get_conversion_t * output )
```

Получение результата последнего измерения со всех каналов.

Аргументы

in	handle	- Идентификатор устройства, полученный от <code>usbadc10_open_device()</code> .
out	output	- Данные, получаемые с устройства.

2.1.5.3 `usbadc10_get_conversion_raw()`

```
USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_get_↔
conversion_raw (
    device_t handle,
    usbadc10_get_conversion_raw_t * output )
```

Получение результата последнего измерения со всех каналов в кодах АЦП.

Аргументы

in	handle	- Идентификатор устройства, полученный от <code>usbadc10_open_device()</code> .
out	output	- Данные, получаемые с устройства.

2.1.5.4 `usbadc10_get_identity_information()`

```
USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_get_↔
identity_information (
    device_t handle,
    usbadc10_get_identity_information_t * output )
```

Возвращает идентификационную информацию об устройстве, такую как номера версий прошивки и серийный номер. Эта информация удобна для поиска нужного устройства среди списка доступных. Может быть вызвана как из прошивки, так и из бутлоадера.

Аргументы

in	handle	- Идентификатор устройства, полученный от <code>usbadc10_open_device()</code> .
out	output	- Данные, получаемые с устройства.

2.1.5.5 `usbadc10_get_profile()`

```
USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_get_profile (
    device_t handle,
    char ** buffer,
    void *(*)(size_t) allocate )
```

Загружает профиль с устройства.

Аргументы

in	handle	- Идентификатор устройства.
out	buffer	- Адрес указателя на выходной буфер. Память для указателя на <code>char*</code> должна быть выделена.
out	allocate	- Функция для выделения памяти.

2.1.5.6 `usbadc10_libversion()`

```
USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_libversion (
    char * lib_version )
```

Версия библиотеки.

Аргументы

out	lib_version	- Версия библиотеки.
-----	-------------	----------------------

2.1.5.7 `usbadc10_logging_callback_stderr_narrow()`

```
USBADC10_URPC_API_EXPORT void USBADC10_URPC_CALLING_CONVENTION usbadc10_logging_↔
callback_stderr_narrow (
    int loglevel,
    const wchar_t * message,
    void * user_data )
```

Простая функция логирования на `stderr` в узких (однобайтных) символах.

Аргументы

loglevel	- Уровень логирования.
message	- Сообщение.

2.1.5.8 usbadc10_logging_callback_stderr_wide()

```
USBADC10_URPC_API_EXPORT void USBADC10_URPC_CALLING_CONVENTION usbadc10_logging_↔
callback_stderr_wide (
    int loglevel,
    const wchar_t * message,
    void * user_data )
```

Простая функция логирования на stderr в широких символах.

Аргументы

loglevel	- Уровень логирования.
message	- Сообщение.

2.1.5.9 usbadc10_open_device()

```
USBADC10_URPC_API_EXPORT device_t USBADC10_URPC_CALLING_CONVENTION usbadc10_open_device
(
    const char * uri )
```

Открывает устройство по имени name и возвращает идентификатор устройства.

Аргументы

in	name	- Имя устройства. Имя устройства имеет вид "com:port" или xi-net://host/serial. Для COM устройства "port" это имя устройства в ОС. Например "com:\\.\COM3" (Windows) или "com:///dev/tty/ttyACM34" (Linux/Mac). Для сетевого устройства "host" это IPv4 адрес или полностью определённое имя домена, "serial" это серийный номер устройства в шестнадцатеричной системе. Например "xi-net://192.168.0.1/00001234" или "xi-net://hostname.com/89ABCDEF". Замечание: в один момент времени COM устройство может использоваться только одной программой. Если при открытии устройства возникают ошибки, нужно убедиться, что COM-порт есть в системе и что это устройство в данный момент не используется другими программами
----	------	---

2.1.5.10 usbadc10_set_logging_callback()

```
USBADC10_URPC_API_EXPORT void USBADC10_URPC_CALLING_CONVENTION usbadc10_set_logging_↔
callback (
    usbadc10_logging_callback_t cb,
    void * data )
```

Устанавливает функцию обратного вызова для логирования. Передача NULL в качестве аргумента отключает логирование.

Аргументы

logging_callback	указатель на функцию обратного вызова
------------------	---------------------------------------

2.1.5.11 usbadc10_set_profile()

```
USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_set_profile (  
    device_t handle,  
    char * buffer )
```

Загружает профиль с устройства.

Аргументы

in	handle	- Идентификатор устройства.
in	buffer	- Входной буфер, откуда будет считан профиль.

Предметный указатель

LOGLEVEL_DEBUG

usbadc10.h, 3

LOGLEVEL_ERROR

usbadc10.h, 4

LOGLEVEL_INFO

usbadc10.h, 4

LOGLEVEL_WARNING

usbadc10.h, 4

usbadc10.h, 1

LOGLEVEL_DEBUG, 3

LOGLEVEL_ERROR, 4

LOGLEVEL_INFO, 4

LOGLEVEL_WARNING, 4

usbadc10_close_device, 4

usbadc10_get_conversion, 5

usbadc10_get_conversion_raw, 5

usbadc10_get_identity_information, 5

usbadc10_get_profile, 6

usbadc10_libversion, 6

usbadc10_logging_callback_stderr_narrow,
6

usbadc10_logging_callback_stderr_wide, 7

usbadc10_logging_callback_t, 4

usbadc10_open_device, 7

usbadc10_set_logging_callback, 7

usbadc10_set_profile, 8

usbadc10_close_device

usbadc10.h, 4

usbadc10_get_conversion

usbadc10.h, 5

usbadc10_get_conversion_raw

usbadc10.h, 5

usbadc10_get_conversion_raw_t, 3

usbadc10_get_conversion_t, 3

usbadc10_get_identity_information

usbadc10.h, 5

usbadc10_get_identity_information_t, 3

usbadc10_get_profile

usbadc10.h, 6

usbadc10_libversion

usbadc10.h, 6

usbadc10_logging_callback_stderr_narrow

usbadc10.h, 6

usbadc10_logging_callback_stderr_wide

usbadc10.h, 7

usbadc10_logging_callback_t

usbadc10.h, 4

usbadc10_open_device

usbadc10.h, 7

usbadc10_set_logging_callback

usbadc10.h, 7

usbadc10_set_profile

usbadc10.h, 8