usbadc10

Создано системой Doxygen 1.8.13

Содержание

1 Алфавитный указатель структур данных

1.1 Структуры данных

Структуры данных с их кратким описанием.

2 Список файлов

2.1 Файлы

Полный список документированных файлов.

3 Структуры данных

3.1 Структура userimpl_data_t

Поля данных

- void * payload
- usbadc10_logging_callback_t cb

3.1.1 Поля

3.1.1.1 cb

usbadc10_logging_callback_t cb

3.1.1.2 payload

void* payload

Объявления и описания членов структуры находятся в файле:

- logging.cpp

4 Файлы

4.1 Файл usbadc10.h

usbadc10 API

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

Структуры данных

- struct usbadc10_get_identity_information_t
- struct usbadc10_debug_read_t
- struct usbadc10_debug_write_t
- struct usbadc10_get_conversion_raw_t
- struct usbadc10_get_conversion_t
- struct usbadc10_calibration_settings_t

Макросы

- #define USBADC10_BUILDER_VERSION_MAJOR 0
- #define USBADC10_BUILDER_VERSION_MINOR 10
- #define USBADC10_BUILDER_VERSION_BUGFIX 27
- #define USBADC10_BUILDER_VERSION_SUFFIX ""
- #define USBADC10_BUILDER_VERSION "0.10.27"
- #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_nodevice (-4)
- #define result_timeout (-5)
- #define STR_result_ok_0 "result_ok 0"
- #define STR_device_undefined_1 "device_undefined (-1)"
- #define STR_result_error_1 "result_error (-1)"
- #define STR_result_not_implemented_2 "result_not_implemented (-2)"
- #define STR_result_value_error_3 "result_value_error (-3)"
- #define STR_result_nodevice_4 "result_nodevice (-4)"
- #define STR_result_timeout_5 "result_timeout (-5)"

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 *)
- USBADC10_URPC_API_EXPORT void USBADC10_URPC_CALLING_CONVENTION usbadc10_logging_callback_stderr_narrow (int loglevel, const wchar_t *message, void *)
- 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_save_settings (device_t handle)
- USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_read_settings (device_t handle)

- 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_reboot_to_bootloader (device_t handle)
- USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_debug_read (device_t handle, usbadc10_debug_read_t *output)
- USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_debug_write (device_t handle, usbadc10_debug_write_t *input)
- USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_reset (device_t handle)
- USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_update_firmware (device_t handle)
- 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_get_calibration_settings (device_t handle, usbadc10_calibration_settings_t *output)
- USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_set_calibration_settings (device_t handle, usbadc10_calibration_settings_t *input)
- 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)

4.1.1 Подробное описание

usbadc10 API

4.1.2 Структуры данных

4.1.2.1 struct usbadc10_get_identity_information_t

Поля структур

uint16_t	BootloaderBugfix	Номер ревизии загрузчика.
uint8_t	BootloaderMajor	Мажорный номер версии загрузчика.
uint8_t	BootloaderMinor	Минорный номер версии загрузчика.
uint8_t	ControllerName[16]	Пользовательское имя контроллера. Может быть установлено пользователем с помощью отдельной команды.
uint16_t	FirmwareBugfix	Номер ревизии прошивки.
uint8_t	FirmwareMajor	Мажорный номер версии прошивки.
uint8_t	FirmwareMinor	Минорный номер версии прошивки.
uint16_t	HardwareBugfix	Номер ревизии платы.
uint8_t	HardwareMajor	Основной номер версии железа.
uint8_t	HardwareMinor	Второстепенный номер версии железа.
uint8_t	Manufacturer[16]	Имя производителя. Устанавливается производителем.
uint8_t	ProductName[16]	Название продукта. Устанавливается производителем.

Поля структур

uint8_t	Reserved[8]	Значение данного поля не должно использоваться в прикладном ПО. Для обеспечения совместимости с другими устройствами не изменяйте значение этого поля.
uint32_t	SerialNumber	Серийный номер изделия.

4.1.2.2 struct usbadc10_debug_read_t

Поля структур

uint8_t	DebugData[128]	Отладочные данные.
uint8_t	Reserved[8]	

4.1.2.3 struct usbadc10_debug_write_t

Поля структур

uint8_t	DebugData[128]	Отладочные данные.
uint8_t	Reserved[8]	

4.1.2.4 struct usbadc10_get_conversion_raw_t

Поля структур

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

4.1.2.5 struct usbadc10_get_conversion_t

Поля структур

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

4.1.2.6 struct usbadc10_calibration_settings_t

Поля структур

uint8_t	Reserved[4]	

4.1.3 Макросы

4.1.3.1 LOGLEVEL_DEBUG

#define LOGLEVEL_DEBUG 0x04

Уровень логирования - отладка

4.1.3.2 LOGLEVEL_ERROR

#define LOGLEVEL_ERROR 0x01

Уровень логирования - ошибка

4.1.3.3 LOGLEVEL_INFO

#define LOGLEVEL_INFO 0x03

Уровень логирования - информация

4.1.3.4 LOGLEVEL_WARNING

#define LOGLEVEL_WARNING 0x02

Уровень логирования - предупреждение

4.1.4 Типы

4.1.4.1 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.

4.1.5 Функции

4.1.5.1 usbadc10_close_device()

USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_close_device (
device_t * handle_ptr)

Закрывает устройство.

Аргументы

handle_ptr	- Идентификатор устройства.

4.1.5.2 usbadc10_debug_read()

```
USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_debug_read
(
        device_t handle,
        usbadc10_debug_read_t * output )
```

Чтение данных из прошивки для отладки и поиска неисправностей. Получаемые данные зависят от версии прошивки, истории и контекста использования. Должна использоваться только производителем.

Аргументы

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

4.1.5.3 usbadc10_debug_write()

```
USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_debug_write
(
        device_t handle,
        usbadc10_debug_write_t * input )
```

Запись данных в прошивку для отладки и поиска неисправностей. Используется только производителем.

Аргументы

in	handle	- Идентификатор устройства, полученный от usbadc10_open_device().
in	input	- Данные, отправляемые устройству.

4.1.5.4 usbadc10_get_calibration_settings()

```
USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_get_↩
calibration_settings (
        device_t handle,
        usbadc10_calibration_settings_t * output )
```

Записывает коэффициенты калибровки. Используется только производителем.

Аргументы

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

4.1.5.5 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 usbadc10_open_device().
out	output	- Данные, получаемые с устройства.

4.1.5.6 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 usbadc10_open_device().
out	output	- Данные, получаемые с устройства.

4.1.5.7 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 usbadc10_open_device().
out	output	- Данные, получаемые с устройства.

4.1.5.8 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 char* должна быть выделена.
out	allocate	- Функция для выделения памяти.

4.1.5.9 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.

4.1.5.10 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 * )
```

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

Аргументы

loglevel	- Уровень логирования.
message	- Сообщение.

4.1.5.11 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 * )
```

Простая функция логирования на stderr в широких символах.

Аргументы

loglevel	- Уровень логирования.
message	- Сообщение.

4.1.5.12 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 или udp://host:port. Для COM устройства "port" это имя устройства в ОС. Например "com:\\.\COM3" (Windows) или "com:///dev/tty/ttyACM34" (Linux/Mac). Для сетевого (XiNet) 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". Для ethernet переходника com-udp "host" это IPv4 адрес переходника, "port" это порт переходника. Например "udp://192.168.0.2:1024" 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

4.1.5.13 usbadc10_read_settings()

```
USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_read_↩
settings (
        device_t handle )
```

Чтение всех настроек контроллера из flash памяти в оперативную, заменяя текущие настройки.

Аргументы

in	handle	- Идентификатор устройства, полученный от usbadc10_open_device().

4.1.5.14 usbadc10_reboot_to_bootloader()

```
USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_reboot_to_bootloader (
        device_t handle )
```

Команда служит для перезагрузки контроллера в загрузчик. Получив такую команду, прошивка платы устанавливает флаг (для загрузчика), отправляет ответ и перезагружает контроллер.

Аргументы

in	handle	- Идентификатор устройства, полученный от usbadc10_open_device().

4.1.5.15 usbadc10_reset()

```
USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_reset (
        device_t handle )
```

Команда для перезагрузки контроллера эквивалентная перезагрузке по питанию. В нормальной практике использоваться не должна.

Аргументы

in	handle	- Идентификатор устройства, полученный от usbadc10_open_device().

4.1.5.16 usbadc10_save_settings()

```
USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_save_settings (
        device_t handle )
```

При получении команды контроллер выполняет операцию сохранения текущих настроек во встроенную энергонезависимую память контроллера.

Аргументы

in	handle	- Идентификатор устройства, полученный от usbadc10_open_device().

4.1.5.17 usbadc10_set_calibration_settings()

```
USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_set_↩
calibration_settings (
        device_t handle,
        usbadc10_calibration_settings_t * input )
```

Записывает коэффициенты калибровки. Используется только производителем.

Аргументы

in	handle	- Идентификатор устройства, полученный от usbadc10_open_device().
in	input	- Данные, отправляемые устройству.

4.1.5.18 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

4.1.5.19 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.

4.1.5.20 usbadc10_update_firmware()

```
USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_update_↩
firmware (
        device_t handle )
```

Эта команда считается устаревшей. Сохраняется в протоколе для обратной совместимости. В новых устройствах следует использовать команду rbld.

Аргументы

in	handle	- Идентификатор устройства, полученный от usbadc10_open_device().

Предметный указатель