usbadc10

Generated by Doxygen 1.8.13

Contents

1 Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

2 File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

3 Data Structure Documentation

3.1 userimpl_data_t Struct Reference

Data Fields

- void * **payload**
- usbadc10_logging_callback_t **cb**

3.1.1 Field Documentation

3.1.1.1 cb

```
usbadc10_logging_callback_t cb
```

3.1.1.2 payload

```
void* payload
```

The documentation for this struct was generated from the following file:

- logging.cpp

4 File Documentation

4.1 usbadc10.h File Reference

usbadc10 API

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

Data Structures

- 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

Macros

- #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)"

Logging level

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

Typedefs

- 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)

Functions

- 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 Detailed Description

usbadc10 API

4.1.2 Data Structure Documentation

4.1.2.1 struct usbadc10_get_identity_information_t

Data Fields

uint16_t	BootloaderBugfix	Bootloader revision number.
uint8_t	BootloaderMajor	Bootloader major version number.
uint8_t	BootloaderMinor	Bootloader minor version number.
uint8_t	ControllerName[16]	User controller name. This name can be set by user via additional command.
uint16_t	FirmwareBugfix	Firmware revision number.
uint8_t	FirmwareMajor	Firmware major version number.
uint8_t	FirmwareMinor	Firmware minor version number.
uint16_t	HardwareBugfix	Revision number of the hardware version.
uint8_t	HardwareMajor	Major number of the hardware version.
uint8_t	HardwareMinor	Minor number of the hardware version.
uint8_t	Manufacturer[16]	Manufacturer name. The name is set by the manufacturer.
uint8_t	ProductName[16]	Product name. The name is set by the manufacturer.
uint8_t	Reserved[8]	Software should not rely on the value of this field. To provide compatibility with future products the value of this field shouldn't be modified.
uint32_t	SerialNumber	Device serial number.

4.1.2.2 struct usbadc10_debug_read_t

Data Fields

uint8_t	DebugData[128]	Arbitrary debug data.
uint8_t	Reserved[8]	

4.1.2.3 struct usbadc10_debug_write_t

Data Fields

uint8_t	DebugData[128]	Arbitrary debug data.
uint8_t	Reserved[8]	

4.1.2.4 struct usbadc10_get_conversion_raw_t

Data Fields

uint16_t	data[10]	Array with conversion results from 10 ADC channels with 12-bit precision. 0 corresponds to minimal voltage (GND), 4095 corresponds to ADC reference voltage (3.3 V).

4.1.2.5 struct usbadc10_get_conversion_t

Data Fields

uint16_t	data[10]	Array with conversion results from 10 channels. Units: 100 uV. For example returned value 123 corresponds to the voltage of 12.3 mV.

4.1.2.6 struct usbadc10_calibration_settings_t

Data Fields

uint8_t	Reserved[4]	

4.1.3 Macro Definition Documentation

4.1.3.1 LOGLEVEL_DEBUG

```
#define LOGLEVEL_DEBUG 0x04
```

Logging level - debug

4.1.3.2 LOGLEVEL_ERROR

```
#define LOGLEVEL_ERROR 0x01
```

Logging level - error

4.1.3.3 LOGLEVEL_INFO

```
#define LOGLEVEL_INFO 0x03
```

Logging level - info

4.1.3.4 LOGLEVEL_WARNING

```
#define LOGLEVEL_WARNING 0x02
```

Logging level - warning

4.1.4 Typedef Documentation

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)
```

Logging callback prototype.

Parameters

loglevel	- A logging level.
message	- A message.

4.1.5 Function Documentation

4.1.5.1 usbadc10_close_device()

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

Close specified device.

Parameters

handle_ptr	- An identifier of device.

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 )
```

Read data from firmware for debug purpose. Its use depends on context, firmware version and previous history. Should be used by manufacturer only.

Parameters

in	*handle*	- Device ID, obtained by usbadc10_open_device() function.
out	*output*	- Device out data.

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 )
```

Write data to firmware for debug purpose. Should be used by manufacturer only.

Parameters

in	*handle*	- Device ID, obtained by usbadc10_open_device() function.
in	*input*	- Device in data.

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 )
```

This command writes the calibration coefficients. Can be used only by the manufacturer.

Parameters

in	*handle*	- Device ID, obtained by usbadc10_open_device() function.
out	*output*	- Device out data.

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 )
```

Gets the last conversion result from all channels.

Parameters

in	*handle*	- Device ID, obtained by usbadc10_open_device() function.
out	*output*	- Device out data.

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 )
```

Gets the last conversion result from all channels in ADC codes.

Parameters

in	*handle*	- Device ID, obtained by usbadc10_open_device() function.
out	*output*	- Device out data.

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 )
```

Return device identity information such as firmware version and serial number. It is useful to find your device in a list of available devices. It can be called from the firmware and bootloader.

Parameters

in	*handle*	- Device ID, obtained by usbadc10_open_device() function.
out	*output*	- Device out data.

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 )
```

Load profile from device.

Parameters

in	*handle*	- Device id.
out	*buffer*	- Pointer to output char* buffer. Memory for char* pointer must be allocated.
out	*allocate*	- Function for memory allocate.

4.1.5.9 usbadc10_libversion()

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

Get library version.

Parameters

out	*lib_version*	- Library version.

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 *  )
```

Simple callback for logging to stderr in narrow (single byte) chars.

Parameters

loglevel	- A logging level.
message	- A message.

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 *  )
```

Simple callback for logging to stderr in wide chars.

Parameters

loglevel	- A logging level.
message	- A message.

4.1.5.12 usbadc10_open_device()

```
USBADC10_URPC_API_EXPORT device_t USBADC10_URPC_CALLING_CONVENTION usbadc10_open_device (
```

```
            const char * uri )
```

Open a device by name *name* and return identifier of the device which can be used in calls.

Parameters

in	*name*	- A device name. Device name has form "com:port" or "xi-net://host/serial" or "udp://host:port". In case of USB-COM port the "port" is the OS device uri. For example "com:\\.\COM3" in Windows or "com:///dev/ttyACM34" in Linux/Mac. In case of network device the "host" is an IPv4 address or fully qualified domain uri (FQDN), "serial" is the device serial number in hexadecimal system. For example "xi-net://192.168.0.1/00001234" or "xi-net://hostname.com/89ABCDEF". In case of ethernet udp-com adapter the "host" is an IPv4 address, "port" is network port For example: "udp://192.168.0.2:1024" Note: only one program may use COM-device in same time. If errors occur when opening device, you need to make sure that the COM port is in the system and device is not currently used by other programs.

4.1.5.13 usbadc10_read_settings()

```
USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_read_settings (
            device_t handle )
```

Read all settings from controller's flash memory to controller's RAM, replacing previous data in controller's RAM.

Parameters

in	*handle*	- Device ID, obtained by usbadc10_open_device() function.

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 )
```

This command reboots the controller to the bootloader. After receiving this command, the firmware sets a flag (for bootloader), sends reply and restarts the controller.

Parameters

in	*handle*	- Device ID, obtained by usbadc10_open_device() function.

4.1.5.15 usbadc10_reset()

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

Resets controller equivalently to the power switch reset. Shouldn't be used in normal practice.

Parameters

in	*handle*	- Device ID, obtained by usbadc10_open_device() function.

4.1.5.16 usbadc10_save_settings()

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

Save all settings from controller's RAM to controller's flash memory, replacing previous data in controller's flash memory.

Parameters

in	*handle*	- Device ID, obtained by usbadc10_open_device() function.

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 )
```

This command writes the calibration coefficients. Can be used only by the manufacturer.

Parameters

in	*handle*	- Device ID, obtained by usbadc10_open_device() function.
in	*input*	- Device in data.

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 )
```

Sets a logging callback. Passing NULL disables logging.

Parameters

logging_callback	a callback for log messages

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 )
```

Save profile to device

Parameters

in	*handle*	- Device id.
in	*buffer*	- Input char* buffer.

4.1.5.20 usbadc10_update_firmware()

```
USBADC10_URPC_API_EXPORT result_t USBADC10_URPC_CALLING_CONVENTION usbadc10_update_firmware (
            device_t handle )
```

This command is deprecated. It is saved in the protocol for backwards compatibility. The new devices use the command rbld.

Parameters

in	*handle*	- Device ID, obtained by usbadc10_open_device() function.

Index