

libasa

0.1.2

Создано системой Doxygen 1.8.16

1 Алфавитный указатель структур данных	1
1.1 Структуры данных	1
2 Список файлов	1
2.1 Файлы	1
3 Структуры данных	2
3.1 Структура <code>asa_settings_t</code>	2
3.1.1 Подробное описание	2
3.1.2 Поля	2
3.2 Структура <code>iv_curve_t</code>	4
3.2.1 Подробное описание	4
3.2.2 Поля	4
3.3 Структура <code>server_t</code>	5
3.3.1 Подробное описание	5
3.3.2 Поля	5
3.4 Структура <code>version_t</code>	6
3.4.1 Подробное описание	6
4 Файлы	6
4.1 Файл <code>C:/Users/New user_2/Desktop/c_lib_with_score/md_asa_lib/include/asa.h</code>	6
4.1.1 Подробное описание	7
4.1.2 Макросы	7
4.1.3 Функции	8
4.2 Файл <code>C:/Users/New user_2/Desktop/c_lib_with_score/md_asa_lib/include/config.h</code> . .	10
4.2.1 Подробное описание	11
Предметный указатель	13

1 Алфавитный указатель структур данных

1.1 Структуры данных

Структуры данных с их кратким описанием.

<code>asa_settings_t</code>	2
<code>iv_curve_t</code>	4
<code>server_t</code>	5
<code>version_t</code>	6

2 Список файлов

2.1 Файлы

Полный список документированных файлов.

C:/Users/New user_2/Desktop/c_lib_with_score/md_asa_lib/include/asa.h	6
C:/Users/New user_2/Desktop/c_lib_with_score/md_asa_lib/include/config.h	10

3 Структуры данных

3.1 Структура asa_settings_t

```
#include <asa.h>
```

Поля данных

- double [DiscFreqHz](#)
- double [ProbeSignalFrequencyHz](#)
- uint32_t [NumberPoints](#)
- uint32_t [NumberChargePoints](#)
- double [VoltageAmplV](#)
- double [MaxCurrentMA](#)
- uint32_t [MeasureFlags](#)
- uint32_t [DebugModelType](#)
- double [DebugModelNominal](#)
- uint32_t [TriggerMode](#)

3.1.1 Подробное описание

Структура для хранения настроек сигнатурного анализатора

См. определение в файле asa.h строка 62

3.1.2 Поля

3.1.2.1 DebugModelNominal double DebugModelNominal

Номинал измеряемого компонента (используется для отладки в виртуальном режиме). Для резистора в Омах. Для конденсатора в Фарадах.

См. определение в файле asa.h строка 72

3.1.2.2 DebugModelType uint32_t DebugModelType

Модель измеряемого элемента (используется для отладки в виртуальном режиме). Принимает одно из значений: COMPONENT_MODEL_TYPE_NONE, COMPONENT_MODEL_TYPE_RESISTOR или COMPONENT_MODEL_TYPE_CAPACITOR.

См. определение в файле asa.h строка 71

3.1.2.3 DiscFreqHz double DiscFreqHz

Частота дискретизации в Гц

См. определение в файле asa.h строка 64

3.1.2.4 MaxCurrentMA double MaxCurrentMA

Максимальный ток в мА (определяет выбор токоизмерительного резистора). Важно! Данный параметр вместе с параметром VoltageAmplV должен принимать значения строго в соответствии со спецификацией.

См. определение в файле asa.h строка 69

3.1.2.5 MeasureFlags uint32_t MeasureFlags

Номер используемого флага. (Не используется!)

См. определение в файле asa.h строка 70

3.1.2.6 NumberChargePoints uint32_t NumberChargePoints

Количество точек, образующих кривую (все измерения). (Не используется!)

См. определение в файле asa.h строка 67

3.1.2.7 NumberPoints uint32_t NumberPoints

Количество точек, образующих кривую (одно измерение)

См. определение в файле asa.h строка 66

3.1.2.8 ProbeSignalFrequencyHz double ProbeSignalFrequencyHz

Частота пробного сигнала в Гц

См. определение в файле asa.h строка 65

3.1.2.9 TriggerMode uint32_t TriggerMode

Режим сохранения кривой. Принимает одно из значений : COMPONENT_TRIGGER_AUTO- кривая обновляется при любом изменении параметров, COMPONENT_TRIGGER_MANUAL- кривая обновляется только при нажатии кнопки "Запуск измерений"

См. определение в файле asa.h строка 73

3.1.2.10 VoltageAmplV double VoltageAmplV

Амплитуда пробного сигнала по напряжению в В. Важно! Данный параметр вместе с параметром MaxCurrentMA должен принимать значения строго в соответствии со спецификацией.

См. определение в файле asa.h строка 68

Объявления и описания членов структуры находятся в файле:

- C:/Users/New user_2/Desktop/c_lib_with_score/md_asa_lib/include/[asa.h](#)

3.2 Структура iv_curve_t

```
#include <asa.h>
```

Поля данных

- double [Voltages](#) [[MAX_NUM_POINTS](#)]
- double [Currents](#) [[MAX_NUM_POINTS](#)]

3.2.1 Подробное описание

Структура для передачи измеренной сигнатуры

См. определение в файле asa.h строка 79

3.2.2 Поля

3.2.2.1 Currents double Currents[[MAX_NUM_POINTS](#)]

Массив точек по току в мА.

См. определение в файле asa.h строка 82

3.2.2.2 Voltages double Voltages[MAX_NUM_POINTS]

Массив точек по напряжению в В.

См. определение в файле asa.h строка 81

Объявления и описания членов структуры находятся в файле:

- C:/Users/New user_2/Desktop/c_lib_with_score/md_asa_lib/include/asa.h

3.3 Структура server_t

```
#include <asa.h>
```

Поля данных

- const char * Host
- const char * Port

3.3.1 Подробное описание

Структура для хранения ip и порта

См. определение в файле asa.h строка 53

3.3.2 Поля

3.3.2.1 Host const char* Host

IP адрес или URL сервера. Пример: "127.0.0.1"

См. определение в файле asa.h строка 55

3.3.2.2 Port const char* Port

Порт сервера. Пример: "8888"

См. определение в файле asa.h строка 56

Объявления и описания членов структуры находятся в файле:

- C:/Users/New user_2/Desktop/c_lib_with_score/md_asa_lib/include/asa.h

3.4 Структура version_t

```
#include <asa.h>
```

Поля данных

- int Major
- int Minor
- int Bugfix

3.4.1 Подробное описание

Структура для хранения трёхцифрных номеров версий

См. определение в файле asa.h строка 43

Объявления и описания членов структуры находятся в файле:

- C:/Users/New user_2/Desktop/c_lib_with_score/md_asa_lib/include/[asa.h](#)

4 Файлы

4.1 Файл C:/Users/New user_2/Desktop/c_lib_with_score/md_asa_lib/include/asa.h

```
#include "stdint.h"
```

Структуры данных

- struct [version_t](#)
- struct [server_t](#)
- struct [asa_settings_t](#)
- struct [iv_curve_t](#)

Макросы

- #define EXPORT
- #define [MAX_NUM_POINTS](#) 1000
- #define [COMPONENT_MODEL_TYPE_NONE](#) 0
- #define [COMPONENT_MODEL_TYPE_RESISTOR](#) 1
- #define [COMPONENT_MODEL_TYPE_CAPACITOR](#) 2
- #define [COMPONENT_TRIGGER_AUTO](#) 0
- #define [COMPONENT_TRIGGER_MANUAL](#) 1
- #define API_VERSION_MAJOR 0
- #define API_VERSION_MINOR 1
- #define API_VERSION_BUGFIX 1
- #define status_t int
- #define message_t char
- #define [ASA_OK](#) 0
- #define [SERVER_RESPONSE_ERROR](#) -1

Функции

- EXPORT `version_t` `GetLibraryVersion` (void)
- EXPORT `version_t` `GetAPIVersion` (void)
- EXPORT `status_t` `SetSettings` (`server_t` *Server, `asa_settings_t` *Settings)
- EXPORT `status_t` `GetSettings` (`server_t` *Server, `asa_settings_t` *Settings)
- EXPORT `status_t` `GetIVCurve` (`server_t` *Server, `iv_curve_t` *IVC, `size_t` size)
- EXPORT `status_t` `TriggerMeasurement` (`server_t` *Server)
- EXPORT void `SetMinVC` (double min_v, double min_c)
- EXPORT double `CompareIvc` (`iv_curve_t` *a, `iv_curve_t` *b, `size_t` size)

4.1.1 Подробное описание

Библиотека для управления аналоговым сигнатурным анализатором. Позволяет подключиться к серверно-аппаратной части, управлять настройками оборудования и получать сигнатуры.

4.1.2 Макросы

4.1.2.1 ASA_OK `#define ASA_OK 0`

Всё работает штатно

См. определение в файле `asa.h` строка 34

4.1.2.2 COMPONENT_MODEL_TYPE_CAPACITOR `#define COMPONENT_MODEL_TYPE_CAPACITOR 2`

Конденсатор

См. определение в файле `asa.h` строка 21

4.1.2.3 COMPONENT_MODEL_TYPE_NONE `#define COMPONENT_MODEL_TYPE_NONE 0`

Неопределённый тип

См. определение в файле `asa.h` строка 19

4.1.2.4 COMPONENT_MODEL_TYPE_RESISTOR `#define COMPONENT_MODEL_TYPE_RESISTOR 1`

Резистор

См. определение в файле `asa.h` строка 20

4.1.2.5 COMPONENT_TRIGGER_AUTO `#define COMPONENT_TRIGGER_AUTO 0`

Автоматический

См. определение в файле `asa.h` строка 23

4.1.2.6 COMPONENT_TRIGGER_MANUAL `#define COMPONENT_TRIGGER_MANUAL 1`

Ручной

См. определение в файле `asa.h` строка 24

4.1.2.7 MAX_NUM_POINTS `#define MAX_NUM_POINTS 1000`

Максимально возможное количество точек кривой

См. определение в файле `asa.h` строка 17

4.1.2.8 SERVER_RESPONSE_ERROR `#define SERVER_RESPONSE_ERROR -1`

Некорректный ответ сервера

См. определение в файле `asa.h` строка 35

4.1.3 Функции

4.1.3.1 CompareIvc() `EXPORT double CompareIvc (` `iv_curve_t * a,` `iv_curve_t * b,` `size_t size)`

Функция для сравнения двух кривых. Возвращает степень визуальной схожести кривых

Аргументы

in	size	Количество точек кривой, обычно равно параметру <code>NumberPoints</code>
----	------	---

4.1.3.2 GetAPIVersion() `EXPORT version_t GetAPIVersion (` `void)`

Функция для получения API - версии

4.1.3.3 GetIVCurve() EXPORT status_t GetIVCurve (
 server_t * Server,
 iv_curve_t * IVC,
 size_t size)

Функция для получения измеренной сигнатуры. Возвращает результат последнего измерения в Manual Mode (TriggerMode=COMPONENT_TRIGGER_MANUAL), в Auto (TriggerMode=COMPONENT_TRIGGER_AUTO) делает измерение и возвращает результат

Аргументы

in	size	Желаемое количество точек кривой, обычно равно параметру NumberPoints
----	------	---

4.1.3.4 GetLibraryVersion() EXPORT version_t GetLibraryVersion (
 void)

Функция для получения версии используемой библиотеки

4.1.3.5 GetSettings() EXPORT status_t GetSettings (
 server_t * Server,
 asa_settings_t * Settings)

Функция для получения настроек сигнатурного анализатора

4.1.3.6 SetMinVC() EXPORT void SetMinVC (
 double min_v,
 double min_c)

Функция для получения измеренной сигнатуры.

Аргументы

in	min_v	Минимальное значение напряжения, которое задает диапазон шума в кривых для вычисления их схожести
in	min_c	Минимальный значение тока, которое задает диапазон шума в кривых для вычисления их схожести

4.1.3.7 SetSettings() EXPORT status_t SetSettings (
 server_t * Server,
 asa_settings_t * Settings)

Функция для установки настроек сигнатурного анализатора

4.1.3.8 TriggerMeasurement() EXPORT status_t TriggerMeasurement (
server_t * Server)

Функция делает одно измерение и сохраняет его в буффер

4.2 Файл C:/Users/New
user_2/Desktop/c_lib_with_score/md_asa_lib/include/config.h

Макросы

- #define HAVE_WCHAR_H 1
- #define HAVE_SYS_FILIO_H 0
- #define HAVE_SYS_IOCTL_H 1
- #define HAVE_SYS_SELECT_H 1
- #define HAVE_WCSNCP 1
- #define HAVE_SETGROUPS 1
- #define HAVE_ASPRINTF 1
- #define HAVE_SETENV 1
- #define HAVE_STRTOLL 1
- #define HAVE_STRTOULL 1
- #define HAVE_STRTOQ 1
- #define HAVE_STRTOUQ 1
- #define HAVE___STRTOLL 0
- #define HAVE___STRTOULL 0
- #define HAVE___STRTOUI64 0
- #define HAVE_PSELECT 1
- #define HAVE_GETTIMEOFDAY 1
- #define HAVE_LOCALTIME_R 1
- #define HAVE_GMTIME_R 1
- #define HAVE_STRCASECMP 1
- #define HAVE_STRICMP 0
- #define HAVE___STRICMP 0
- #define VA_LIST_IS_ARRAY 1
- #define HAVE_LIBWWW_SSL 0
- #define ATTR_UNUSED __attribute__((__unused__))
- #define DIRECTORY_SEPARATOR "/"
- #define HAVE_UNICODE_WCHAR HAVE_WCHAR_H
- #define __inline__
- #define MSVCRT 0
- #define HAVE_TIMEVAL 1
- #define HAVE_TIMESPEC 1
- #define HAVE_WINDOWS_THREAD 0
- #define HAVE_PTHREAD 1
- #define XMLRPC_SNPRINTF snprintf
- #define XMLRPC_VSNPRINTF vsnprintf
- #define HAVE_REGEX 1
- #define XMLRPC_SOCKETPAIR socketpair
- #define XMLRPC_CLOSESOCKET close
- #define XMLRPC_STRTOLL strtoll
- #define XMLRPC_STRTOULL strtoull
- #define XMLRPC_S_IWUSR S_IWUSR
- #define XMLRPC_S_IRUSR S_IRUSR
- #define XMLRPC_CHDIR chdir
- #define XMLRPC_GETPID getpid
- #define HAVE_ABYSS_OPENSSL 0
- #define XMLRPC_FINITE finite

4.2.1 Подробное описание

xmlrpc_config.h is generated from xmlrpc_config.h.in by 'configure'.

The purpose of this file is to define stuff particular to the build environment being used to build Xmlrpc-c. Xmlrpc-c source files can `#include` this file and have build-environment-independent source code.

A major goal of this file is to reduce conditional compilation in the other source files as much as possible. Even more, we want to avoid having to generate source code particular to a build environment except in this file.

This file is NOT meant to be used by any code outside of the Xmlrpc-c source tree. There is a similar file that gets installed as `<xmlrpc-c/config.h>` that performs the same function for Xmlrpc-c interface header files that get compiled as part of a user's program.

This file just uses plain AC_SUBST substitution, the same as config.mk. Wherever you see @XXX@, that gets replaced by the value of 'configure' variable XXX.

Logical macros are 0 or 1 instead of the more traditional defined and undefined. That's so we can distinguish when compiling code between "false" and some problem with the code.

Предметный указатель

asa.h
 ASA_OK, 7
 CompareIvc, 8
 COMPONENT_MODEL_TYPE_CAPACITOR, 7
 COMPONENT_MODEL_TYPE_NONE, 7
 COMPONENT_MODEL_TYPE_RESISTOR, 7
 COMPONENT_TRIGGER_AUTO, 7
 COMPONENT_TRIGGER_MANUAL, 8
 GetAPIVersion, 8
 GetIVCurve, 9
 GetLibraryVersion, 9
 GetSettings, 9
 MAX_NUM_POINTS, 8
 SERVER_RESPONSE_ERROR, 8
 SetMinVC, 9
 SetSettings, 9
 TriggerMeasurement, 9
 ASA_OK
 asa.h, 7
 asa_settings_t, 2
 DebugModelNominal, 2
 DebugModelType, 2
 DiscFreqHz, 2
 MaxCurrentMA, 3
 MeasureFlags, 3
 NumberChargePoints, 3
 NumberPoints, 3
 ProbeSignalFrequencyHz, 3
 TriggerMode, 3
 VoltageAmplV, 4
 C:/Users/New user_2/Desktop/c_lib_with_score/md_asa_lib/include/asa.h, 6
 C:/Users/New user_2/Desktop/c_lib_with_score/md_asa_lib/include/asa.h, 10
 CompareIvc
 asa.h, 8
 COMPONENT_MODEL_TYPE_CAPACITOR
 asa.h, 7
 COMPONENT_MODEL_TYPE_NONE
 asa.h, 7
 COMPONENT_MODEL_TYPE_RESISTOR
 asa.h, 7
 COMPONENT_TRIGGER_AUTO
 asa.h, 7
 COMPONENT_TRIGGER_MANUAL
 asa.h, 8
 Currents
 iv_curve_t, 4
 DebugModelNominal
 asa_settings_t, 2
 DebugModelType
 asa_settings_t, 2
 DiscFreqHz
 asa_settings_t, 2
 GetAPIVersion
 asa.h, 8
 GetIVCurve
 asa.h, 9
 GetLibraryVersion
 asa.h, 9
 GetSettings
 asa.h, 9
 Host
 server_t, 5
 iv_curve_t, 4
 Currents, 4
 Voltages, 4
 MAX_NUM_POINTS
 asa.h, 8
 MaxCurrentMA
 asa_settings_t, 3
 MeasureFlags
 asa_settings_t, 3
 NumberChargePoints
 asa_settings_t, 3
 NumberPoints
 asa_settings_t, 3
 Port
 server_t, 5
 ProbeSignalFrequencyHz
 asa_settings_t, 3
 SERVER_RESPONSE_ERROR
 asa.h, 8
 server_t, 5
 Host, 5
 Port, 5
 SetMinVC
 asa.h, 9
 SetSettings
 asa.h, 9
 TriggerMeasurement
 asa.h, 9
 TriggerMode
 asa_settings_t, 3
 version_t, 6
 VoltageAmplV
 asa_settings_t, 4
 Voltages
 iv_curve_t, 4