

libasa

0.3.3

Создано системой Doxygen 1.9.1

1	Список файлов	1
1.1	Файлы	1
2	Файлы	1
2.1	Файл C:/projects/meridian/md_asa_lib/include/asa.h	1
2.1.1	Подробное описание	2
2.1.2	Структуры данных	2
2.1.3	Макросы	5
2.1.4	Функции	9
2.2	Файл C:/projects/meridian/md_asa_lib/include/config.h	14
2.2.1	Подробное описание	15
	Предметный указатель	17

1 Список файлов

1.1 Файлы

Полный список документированных файлов.

C:/projects/meridian/md_asa_lib/include/asa.h	1
C:/projects/meridian/md_asa_lib/include/config.h	14

2 Файлы

2.1 Файл C:/projects/meridian/md_asa_lib/include/asa.h

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

Структуры данных

- struct `version_t`
- struct `server_t`
- struct `asa_settings_t`
- struct `iv_curve_t`
- struct `asa_buttons_t`
- struct `asa_temperature_t`
- struct `asa_coefficients_t`

Макросы

- `#define EXPORT`
- `#define MAX_NUM_POINTS 1000`
- `#define NUM_ALL_COMBINATIONS 380`
- `#define COMPONENT_MODEL_TYPE_NONE 0`
- `#define COMPONENT_MODEL_TYPE_RESISTOR 1`
- `#define COMPONENT_MODEL_TYPE_CAPACITOR 2`
- `#define MODE_AUTO 0`
- `#define MODE_MANUAL 1`
- `#define API_VERSION_MAJOR 0`
- `#define API_VERSION_MINOR 1`
- `#define API_VERSION_BUGFIX 6`
- `#define status_t int32_t`
- `#define message_t char`
- `#define ASA_OK 0`
- `#define ASA_IN_PROGRESS 1`
- `#define SERVER_RESPONSE_ERROR -1`
- `#define INTERNAL_SERVER_ERROR -2`
- `#define ASA_VALUE_ERROR -3`
- `#define ASA_TYPE_ERROR -4`
- `#define ASA_FORMAT_ERROR -5`
- `#define ASA_CONNECTION_ERROR -6`
- `#define FAST_CLOSE_CALIBRATE 0`
- `#define FAST_OPEN_CALIBRATE 1`
- `#define FAST_CLOSE_CALIBRATE_AND_SAVE 2`
- `#define FAST_OPEN_CALIBRATE_AND_SAVE 3`
- `#define FULL_CLOSE_CALIBRATE_AND_SAVE 4`
- `#define FULL_OPEN_CALIBRATE_AND_SAVE 5`

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, uint32_t size)`
- `EXPORT status_t TriggerMeasurement (server_t *Server)`
- `EXPORT status_t Calibrate (server_t *Server, uint8_t Type)`
- `EXPORT status_t GetStatusButtons (server_t *Server, asa_buttons_t *Buttons)`
- `EXPORT status_t GetTempProbes (server_t *Server, asa_temperature_t *Temperature)`
- `EXPORT uint32_t GetNumberPointsForSinglePeriod (asa_settings_t *Settings)`
- `EXPORT status_t GetLastOperationResult (server_t *Server)`
- `EXPORT void LoadCoefficientTable (char *FileName, asa_coefficients_t *C)`
- `EXPORT void SaveCoefficientTable (char *FileName, asa_coefficients_t *C)`
- `EXPORT status_t SetCoefficients (server_t *Server, asa_coefficients_t *C)`
- `EXPORT status_t GetCoefficients (server_t *Server, asa_coefficients_t *C)`

2.1.1 Подробное описание

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

2.1.2 Структуры данных

2.1.2.1 `struct version_t` Структура для хранения трёхцифровых номеров версий

См. определение в файле `asa.h` строка 67

Поля структур

uint8_t	Major	Основной номер версии
uint8_t	Minor	Второстепенный номер версии
uint8_t	Bugfix	Номер правок

2.1.2.2 struct server_t Структура для хранения ip и порта

См. определение в файле asa.h строка 77

Поля структур

const char *	Host	IP адрес или URL сервера. Данные хранятся в виде строки. Пример: "127.0.0.1"
const char *	Port	Порт сервера. Пример: "8888"

2.1.2.3 struct asa_settings_t Структура для хранения настроек сигнатурного анализатора.

См. определение в файле asa.h строка 86

Поля структур

double	SamplingRateHz	Частота дискретизации в Гц. Определяется устройством и зависит от текущей частоты пробного сигнала. При получении настроек возвращается актуальное для данной частоты пробного сигнала значение частоты дискретизации. При установке настроек данное поле никак не влияет на параметры работы устройства.
double	ProbeSignalFrequencyHz	Частота пробного сигнала в Гц.
uint32_t	NumberPoints	Количество точек, образующих кривую (одно измерение).
uint32_t	NumberChargePoints	Значение данного поля не должно использоваться в прикладном ПО. Добавлено для совместимости с другими устройствами.
double	VoltageAmplV	Амплитуда пробного сигнала по напряжению в В. Важно! Данный параметр вместе с параметром MaxCurrentMA должен принимать значения строго в соответствии со спецификацией.
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.
uint32_t	MeasureFlags	Значение данного поля не должно использоваться в прикладном ПО. Добавлено для совместимости с другими устройствами.
uint32_t	DebugModelType	Модель измеряемого элемента. Используется для отладки совместно с виртуальным измерителем. В нормальном режиме работы смысла не имеет. Принимает одно из значений: COMPONENT_MODEL_TYPE_NONE, COMPONENT_MODEL_TYPE_RESISTOR или COMPONENT_MODEL_TYPE_CAPACITOR.

Поля структур

double	DebugModelNominal	Номинал измеряемого компонента. Для резистора в Омах. Для конденсатора в Фарадах. Используется для отладки совместно с виртуальным измерителем. В нормальном режиме работы смысла не имеет.
uint32_t	TriggerMode	Режим запуска измерений. Принимает одно из значений: COMPONENT_TRIGGER_AUTO (непрерывный автоматический запуск измерений), COMPONENT_TRIGGER_MANUAL (запуск измерений по запросу клиента или по нажатию аппаратной кнопки запуска измерений на щупе).

2.1.2.4 struct iv_curve_t Структура для передачи измеренной сигнатуры

См. определение в файле asa.h строка 103

Поля структур

double	Voltages[MAX_NUM_POINTS]	Массив напряжений (координат сигнатуры по напряжению). Единицы измерения: Вольты. Количество актуальных значений может быть меньше MAX_NUM_POINTS и определяется текущими настройками измерителя (поле NumberPoints структуры asa_settings_t). Актуальные значения располагаются начиная с нулевого элемент массива.
double	Currents[MAX_NUM_POINTS]	Массив токов (координат сигнатуры по току). Единицы измерения: мА. Количество актуальных значений может быть меньше MAX_NUM_POINTS и определяется текущими настройками измерителя (поле NumberPoints структуры asa_settings_t). Актуальные значения располагаются начиная с нулевого элемент массива.

2.1.2.5 struct asa_buttons_t Структура для передачи состояния кнопок щупов

См. определение в файле asa.h строка 112

Поля структур

uint32_t	GrayProbeButton	Состояние кнопки щупа-генератора. 1 - если кнопка нажата, 0 - если нет.
uint32_t	BlueProbeButton	Состояние кнопки щупа-измерителя. 1 - если кнопка нажата, 0 - если нет.

2.1.2.6 struct asa_temperature_t Структура для температуры и индикаторов превышения температуры щупов

См. определение в файле asa.h строка 121

Поля структур

uint32_t	GrayTemperature	Температура кнопки щупа-генератора. Значение в градусах.
uint32_t	BlueTemperature	Температура кнопки щупа-измерителя. Значение в градусах.
uint32_t	GrayOverheat	Индикатор превышения температуры кнопки щупа-измерителя допустимого порога. 1 - температура превышена, 0 - нет.
uint32_t	BlueOverheat	Индикатор превышения температуры кнопки щупа-измерителя допустимого порога. 1 - температура превышена, 0 - нет.

2.1.2.7 struct asa_coefficients_t Таблица соответствия коэффициентов калибровки и значений резисторов настройкам устройства (частота пробного сигнала, амплитуда напряжения, максимальный ток).

См. определение в файле asa.h строка 132

Поля структур

double	ProbeFrequencyArray[NUM_ALL_COMBINATIONS]	Массив частот пробного сигнала .
double	VoltageAmplArray[NUM_ALL_COMBINATIONS]	Массив амплитуд пробного сигнала по напряжению в В.
double	MaxCurrentArray[NUM_ALL_COMBINATIONS]	Массив максимальных токов в мА.
uint32_t	LimitResistorArray[NUM_ALL_COMBINATIONS]	Массив токоограничительных резисторов в Омах.
uint32_t	SenseResistorArray[NUM_ALL_COMBINATIONS]	Массив токоизмерительных резисторов в Омах.
uint32_t	CurCalibrationArray[NUM_ALL_COMBINATIONS]	Массив коэффициентов калибровки по осям для тока.
uint32_t	VoltCalibrationArray[NUM_ALL_COMBINATIONS]	Массив коэффициентов калибровки по осям для напряжения.
uint32_t	CurCalibrationKZArray[NUM_ALL_COMBINATIONS]	Массив коэффициентов калибровки наклона для тока.
uint32_t	VoltCalibrationKZArray[NUM_ALL_COMBINATIONS]	Массив коэффициентов калибровки наклона для напряжения.

2.1.3 Макросы

2.1.3.1 MAX_NUM_POINTS #define MAX_NUM_POINTS 1000

Максимально возможное количество точек кривой.

См. определение в файле asa.h строка 19

2.1.3.2 NUM_ALL_COMBINATIONS `#define NUM_ALL_COMBINATIONS 380`

Количество возможных комбинаций настроек (частота пробного сигнала, амплитуда напряжения и амплитуда тока).

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

2.1.3.3 COMPONENT_MODEL_TYPE_NONE `#define COMPONENT_MODEL_TYPE_NONE 0`

Неопределённый тип.

См. определение в файле `asa.h` строка 22

2.1.3.4 COMPONENT_MODEL_TYPE_RESISTOR `#define COMPONENT_MODEL_TYPE_RESISTOR 1`

Резистор.

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

2.1.3.5 COMPONENT_MODEL_TYPE_CAPACITOR `#define COMPONENT_MODEL_TYPE_CAPACITOR 2`

Конденсатор.

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

2.1.3.6 MODE_AUTO `#define MODE_AUTO 0`

Автоматический запуск измерений. Устройство непрерывно проводит измерения и сохраняет результат последнего измерения.

См. определение в файле `asa.h` строка 26

2.1.3.7 MODE_MANUAL `#define MODE_MANUAL 1`

Ручной запуск измерений. Измерения производятся только в случае получения запроса от клиента (вызов функции `TriggerMeasurement()`) либо в случае нажатия аппаратной кнопки на щупе.

См. определение в файле `asa.h` строка 27

2.1.3.8 API_VERSION_MAJOR #define API_VERSION_MAJOR 0

Основной номер версии протокола

См. определение в файле asa.h строка 29

2.1.3.9 API_VERSION_MINOR #define API_VERSION_MINOR 1

Второстепенный номер версии протокола

См. определение в файле asa.h строка 30

2.1.3.10 API_VERSION_BUGFIX #define API_VERSION_BUGFIX 6

Номер правок протокола

См. определение в файле asa.h строка 31

2.1.3.11 ASA_OK #define ASA_OK 0

Коды ошибок Всё работает штатно

См. определение в файле asa.h строка 40

2.1.3.12 ASA_IN_PROGRESS #define ASA_IN_PROGRESS 1

Операция выполняется

См. определение в файле asa.h строка 41

2.1.3.13 SERVER_RESPONSE_ERROR #define SERVER_RESPONSE_ERROR -1

Некорректный ответ сервера

См. определение в файле asa.h строка 42

2.1.3.14 INTERNAL_SERVER_ERROR #define INTERNAL_SERVER_ERROR -2

В ходе выполнения функции на сервере возникла ошибка

См. определение в файле asa.h строка 43

2.1.3.15 ASA_VALUE_ERROR #define ASA_VALUE_ERROR -3

Сервером было получено значение вне допустимого диапазона

См. определение в файле asa.h строка 44

2.1.3.16 ASA_TYPE_ERROR #define ASA_TYPE_ERROR -4

Сервером было получено значение с неправильным типом

См. определение в файле asa.h строка 45

2.1.3.17 ASA_FORMAT_ERROR #define ASA_FORMAT_ERROR -5

Сервером было получено значение с неправильным форматом или не получено вовсе

См. определение в файле asa.h строка 46

2.1.3.18 ASA_CONNECTION_ERROR #define ASA_CONNECTION_ERROR -6

Серверу не удалось подключиться к устройству

См. определение в файле asa.h строка 47

2.1.3.19 FAST_CLOSE_CALIBRATE #define FAST_CLOSE_CALIBRATE 0

Типы калибровки Запуск калибровки для заданного набора параметров с замкнутыми щупами (калибровка наклона и калибровка по напряжению)

См. определение в файле asa.h строка 52

2.1.3.20 FAST_OPEN_CALIBRATE #define FAST_OPEN_CALIBRATE 1

Запуск калибровки для заданного набора параметров с разомкнутыми щупами (калибровка по току)

См. определение в файле asa.h строка 53

2.1.3.21 FAST_CLOSE_CALIBRATE_AND_SAVE #define FAST_CLOSE_CALIBRATE_AND_SAVE 2

Запуск калибровки для заданного набора параметров с замкнутыми щупами (калибровка наклона и калибровка по напряжению) с сохранением результата в энергонезависимую память устройства

См. определение в файле asa.h строка 54

2.1.3.22 FAST_OPEN_CALIBRATE_AND_SAVE #define FAST_OPEN_CALIBRATE_AND_SAVE 3

Запуск калибровки для заданного набора параметров с разомкнутыми щупами (калибровка по току) с сохранением результата в энергонезависимую память устройства

См. определение в файле asa.h строка 55

2.1.3.23 FULL_CLOSE_CALIBRATE_AND_SAVE #define FULL_CLOSE_CALIBRATE_AND_SAVE 4

Запуск калибровки для всех режимов с замкнутыми щупами (калибровка наклона и калибровка по напряжению) с сохранением результатов в энергонезависимую память устройства

См. определение в файле asa.h строка 56

2.1.3.24 FULL_OPEN_CALIBRATE_AND_SAVE #define FULL_OPEN_CALIBRATE_AND_SAVE 5

Запуск калибровки для всех режимов с разомкнутыми щупами (калибровка по току) с сохранением результатов в энергонезависимую память устройства

См. определение в файле asa.h строка 57

2.1.4 Функции

2.1.4.1 GetLibraryVersion() EXPORT version_t GetLibraryVersion (void)

Функция для получения версии используемой библиотеки.

Аргументы

out	Version	Версия библиотеки.
-----	---------	--------------------

2.1.4.2 `GetAPIVersion()` `EXPORT version_t GetAPIVersion (`
`void)`

Функция для получения версии протокола.

Аргументы

out	Version	Версия протокола.
-----	---------	-------------------

2.1.4.3 `SetSettings()` `EXPORT status_t SetSettings (`
`server_t * Server,`
`asa_settings_t * Settings)`

Функция для установки настроек сигнатурного анализатора. Установленные настройки будут применены ко всем последующим измерениям до следующего вызова функции установки настроек либо до аппаратной перезагрузки устройства. Настройки измерителя могут быть независимо изменены другими клиентскими приложениями. Для получения актуальных настроек измерителя используйте функцию [GetSettings\(\)](#).

Аргументы

in	Server	Указатель на структуру с параметрами подключения сервера.
in	Settings	Указатель на структуру с настройками, которые требуется установить.
out	Status	Код статуса завершения. Если функция завершилась успешно, должен быть возвращён статус ASA_OK.

2.1.4.4 `GetSettings()` `EXPORT status_t GetSettings (`
`server_t * Server,`
`asa_settings_t * Settings)`

Функция для получения настроек сигнатурного анализатора. Настройки могут быть изменены с помощью явного вызова функции [SetSettings\(\)](#), перезагрузки измерителя, а также из других клиентских приложений.

Аргументы

in	Server	Указатель на структуру с параметрами подключения сервера.
in	Settings	Указатель на структуру, в которую будут записаны актуальные настройки.
out	Status	Код статуса завершения. Если функция завершилась успешно, должен быть возвращён статус ASA_OK.

2.1.4.5 `GetIVCurve()` `EXPORT status_t GetIVCurve (`
`server_t * Server,`
`iv_curve_t * IVC,`
`uint32_t size)`

Функция для получения измеренной сигнатуры. Возвращает результат последнего измерения. Для запуска измерений в ручном режиме используется функция [TriggerMeasurement\(\)](#) либо аппаратная кнопка запуска измерений на щупе. Режим измерений определяется полем `TriggerMode` в структуре [asa_settings_t](#).

Аргументы

in	Server	Указатель на структуру с параметрами подключения сервера.
in	IVC	Указатель на структуру, в которую будет записан результат измерения.
in	size	Количество точек сигнатуры. Зависит от текущих настроек измерителя. Определяется значением поля <code>NumberPoints</code> в структуре asa_settings_t .
out	Status	Код статуса завершения. Если функция завершилась успешно, должен быть возвращён статус <code>ASA_OK</code> .

2.1.4.6 `TriggerMeasurement()` `EXPORT status_t TriggerMeasurement (`
`server_t * Server)`

Функция осуществляет запуск единичного измерения. Актуальна только для ручного режима запуска измерений (определяется параметром `TriggerMode` структуры [asa_settings_t](#)). Результат измерения можно получить с помощью функции [GetIVCurve\(\)](#). Вызов данной функции эквивалентен нажатию аппаратной кнопки запуска измерений на щупе.

Аргументы

in	Server	Указатель на структуру с параметрами подключения сервера.
out	Status	Код статуса завершения. Если функция завершилась успешно, должен быть возвращён статус <code>ASA_OK</code> .

2.1.4.7 `Calibrate()` `EXPORT status_t Calibrate (`
`server_t * Server,`
`uint8_t Type)`

Функция запускает калибровку в зависимости от выбранного типа. Актуальна только для реального режима запуска измерений. Результат измерения можно получить с помощью функции [GetIVCurve\(\)](#).

Аргументы

in	Server	Указатель на структуру с параметрами подключения сервера.
----	--------	---

Аргументы

in	Type	Тип калибровки
out	Status	Код статуса завершения. Если функция завершилась успешно, должен быть возвращён статус ASA_OK.

2.1.4.8 `GetStatusButtons()` `EXPORT status_t GetStatusButtons (`
`server_t * Server,`
`asa_buttons_t * Buttons)`

Функция возвращает статус кнопок щупов.

Аргументы

in	Server	Указатель на структуру с параметрами подключения сервера.
in	Buttons	содержит статусы кнопок.
out	Status	Код статуса завершения. Если функция завершилась успешно, должен быть возвращён статус ASA_OK.

2.1.4.9 `GetTempProbes()` `EXPORT status_t GetTempProbes (`
`server_t * Server,`
`asa_temperature_t * Temperature)`

Функция возвращает температуру щупов.

Аргументы

in	Server	Указатель на структуру с параметрами подключения сервера.
in	Temperature	- структура для хранения величин температуры и индикаторов превышения температуры щупов.
out	Status	Код статуса завершения. Если функция завершилась успешно, должен быть возвращён статус ASA_OK.

2.1.4.10 `GetNumberPointsForSinglePeriod()` `EXPORT uint32_t GetNumberPointsForSinglePeriod (`
`asa_settings_t * Settings)`

Функция возвращает количество точек в одном периоде. Количество точек зависит от частоты и поэтому нужно вызывать эту функцию каждый раз при её изменении.

Аргументы

in	Settings	Указатель на структуру с параметрами настроек. Настройки обязательно должны быть получены с устройства, чтобы все поля в них соответствовали параметрам, которые в данный момент установлены на устройстве.
out	OnePeriodPoints	количество точек в одном периоде

2.1.4.11 `GetLastOperationResult()` EXPORT status_t GetLastOperationResult (
[server_t](#) * Server)

Функция позволяет узнать результат последней операции. В случае, если операция еще выполняется, функция возвращает ASA_IN_PROGRESS.

Аргументы

in	Settings	Указатель на структуру с параметрами настроек. Настройки обязательно должны быть получены с устройства, чтобы все поля в них соответствовали параметрам, которые в данный момент установлены на устройстве.
out	OperationStatus	статус выполнения последней операции.

2.1.4.12 `LoadCoefficientTable()` EXPORT void LoadCoefficientTable (
char * FileName,
[asa_coefficients_t](#) * C)

Функция загружает таблицу соответствия частот, напряжения, токов, резисторов и калибровочных коэффициентов в буфер.

Аргументы

in	FileName	Имя файла, из которого нужно загрузить таблицу.
in	C	Структура, хранящая в себе таблицу соответствия частот, напряжения, токов, резисторов и калибровочных коэффициентов.

2.1.4.13 `SaveCoefficientTable()` EXPORT void SaveCoefficientTable (
char * FileName,
[asa_coefficients_t](#) * C)

Функция сохраняет таблицу соответствия частот, напряжения, токов, резисторов и калибровочных коэффициентов в файл.

Аргументы

in	FileName	Имя файла, в который нужно сохранить таблицу.
in	C	Структура, хранящая в себе таблицу соответствия частот, напряжения, токов, резисторов и калибровочных коэффициентов.

```
2.1.4.14 SetCoefficients() EXPORT status_t SetCoefficients (
    server_t * Server,
    asa_coefficients_t * C )
```

Функция отправляет таблицу соответствия частот, напряжения, токов, резисторов и калибровочных коэффициентов серверу для последующей установки значений на устройство.

Аргументы

in	Server	Указатель на структуру с параметрами подключения сервера.
in	C	Структура, хранящая в себе таблицу соответствия частот, напряжения, токов, резисторов и калибровочных коэффициентов.
out	Status	Код статуса завершения. Если функция завершилась успешно, должен быть возвращён статус ASA_OK.

```
2.1.4.15 GetCoefficients() EXPORT status_t GetCoefficients (
    server_t * Server,
    asa_coefficients_t * C )
```

Функция отправляет запрос на получение таблицы соответствия частот, напряжения, токов, резисторов и калибровочных коэффициентов с сервера.

Аргументы

in	Server	Указатель на структуру с параметрами подключения сервера.
in	C	Структура, хранящая в себе таблицу соответствия частот, напряжения, токов, резисторов и калибровочных коэффициентов.
out	Status	Код статуса завершения. Если функция завершилась успешно, должен быть возвращён статус ASA_OK.

2.2 Файл C:/projects/meridian/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_WCSNCMP 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___STRTOULL 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

2.2.1 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

- API_VERSION_BUGFIX
 - asa.h, 7
- API_VERSION_MAJOR
 - asa.h, 6
- API_VERSION_MINOR
 - asa.h, 7
- asa.h
 - API_VERSION_BUGFIX, 7
 - API_VERSION_MAJOR, 6
 - API_VERSION_MINOR, 7
 - ASA_CONNECTION_ERROR, 8
 - ASA_FORMAT_ERROR, 8
 - ASA_IN_PROGRESS, 7
 - ASA_OK, 7
 - ASA_TYPE_ERROR, 8
 - ASA_VALUE_ERROR, 8
 - Calibrate, 11
 - COMPONENT_MODEL_TYPE_CAPACITOR, 6
 - COMPONENT_MODEL_TYPE_NONE, 6
 - COMPONENT_MODEL_TYPE_RESISTOR, 6
 - FAST_CLOSE_CALIBRATE, 8
 - FAST_CLOSE_CALIBRATE_AND_SAVE, 9
 - FAST_OPEN_CALIBRATE, 8
 - FAST_OPEN_CALIBRATE_AND_SAVE, 9
 - FULL_CLOSE_CALIBRATE_AND_SAVE, 9
 - FULL_OPEN_CALIBRATE_AND_SAVE, 9
 - GetAPIVersion, 10
 - GetCoefficients, 14
 - GetIVCurve, 11
 - GetLastOperationResult, 13
 - GetLibraryVersion, 9
 - GetNumberPointsForSinglePeriod, 12
 - GetSettings, 10
 - GetStatusButtons, 12
 - GetTempProbes, 12
 - INTERNAL_SERVER_ERROR, 7
 - LoadCoefficientTable, 13
 - MAX_NUM_POINTS, 5
 - MODE_AUTO, 6
 - MODE_MANUAL, 6
 - NUM_ALL_COMBINATIONS, 5
 - SaveCoefficientTable, 13
 - SERVER_RESPONSE_ERROR, 7
 - SetCoefficients, 14
 - SetSettings, 10
 - TriggerMeasurement, 11
- asa_buttons_t, 4
- asa_coefficients_t, 5
- ASA_CONNECTION_ERROR
 - asa.h, 8
- ASA_FORMAT_ERROR
 - asa.h, 8
- ASA_IN_PROGRESS
 - asa.h, 7
- ASA_OK
 - asa.h, 7
- asa_settings_t, 3
- asa_temperature_t, 4
- ASA_TYPE_ERROR
 - asa.h, 8
- ASA_VALUE_ERROR
 - asa.h, 8
- C:/projects/meridian/md_asa_lib/include/asa.h, 1
- C:/projects/meridian/md_asa_lib/include/config.h, 14
- Calibrate
 - asa.h, 11
- COMPONENT_MODEL_TYPE_CAPACITOR
 - asa.h, 6
- COMPONENT_MODEL_TYPE_NONE
 - asa.h, 6
- COMPONENT_MODEL_TYPE_RESISTOR
 - asa.h, 6
- FAST_CLOSE_CALIBRATE
 - asa.h, 8
- FAST_CLOSE_CALIBRATE_AND_SAVE
 - asa.h, 9
- FAST_OPEN_CALIBRATE
 - asa.h, 8
- FAST_OPEN_CALIBRATE_AND_SAVE
 - asa.h, 9
- FULL_CLOSE_CALIBRATE_AND_SAVE
 - asa.h, 9
- FULL_OPEN_CALIBRATE_AND_SAVE
 - asa.h, 9
- GetAPIVersion
 - asa.h, 10
- GetCoefficients
 - asa.h, 14
- GetIVCurve
 - asa.h, 11
- GetLastOperationResult
 - asa.h, 13
- GetLibraryVersion
 - asa.h, 9
- GetNumberPointsForSinglePeriod
 - asa.h, 12
- GetSettings
 - asa.h, 10
- GetStatusButtons
 - asa.h, 12

GetTempProbes
asa.h, 12

INTERNAL_SERVER_ERROR
asa.h, 7
iv_curve_t, 4

LoadCoefficientTable
asa.h, 13

MAX_NUM_POINTS
asa.h, 5

MODE_AUTO
asa.h, 6

MODE_MANUAL
asa.h, 6

NUM_ALL_COMBINATIONS
asa.h, 5

SaveCoefficientTable
asa.h, 13

SERVER_RESPONSE_ERROR
asa.h, 7
server_t, 3

SetCoefficients
asa.h, 14

SetSettings
asa.h, 10

TriggerMeasurement
asa.h, 11

version_t, 2