

wxModbus

Создано системой Doxygen 1.8.18

1 Приложение	1
2 MainFrame	3
3 Иерархический список классов	5
3.1 Иерархия классов	5
4 Алфавитный указатель классов	7
4.1 Классы	7
5 Список файлов	9
5.1 Файлы	9
6 Классы	11
6.1 Структура <code>_modbus</code>	11
6.2 Структура <code>_modbus_backend</code>	11
6.3 Структура <code>_modbus_mapping_t</code>	12
6.4 Структура <code>_modbus_rtu</code>	12
6.5 Структура <code>_modbus_tcp</code>	13
6.6 Структура <code>_modbus_tcp_pi</code>	13
6.7 Структура <code>_sft</code>	13
6.8 Класс <code>Adder</code>	13
6.9 Класс <code>Dial</code>	14
6.10 Класс <code>Interruptable</code>	14
6.11 Шаблон класса <code>InterruptableClasses< x ></code>	15
6.12 Класс <code>InterruptManager</code>	15
6.13 Класс <code>mainFrame</code>	15
6.13.1 Подробное описание	16
6.13.2 Конструктор(ы)	16
6.13.2.1 <code>mainFrame()</code>	16
6.13.3 Методы	16
6.13.3.1 <code>onButExit()</code>	16
6.14 Класс <code>MyApp</code>	17
6.14.1 Подробное описание	17
6.14.2 Методы	17
6.14.2.1 <code>OnInit()</code>	17
6.15 Класс <code>Subber</code>	18
7 Файлы	19
7.1 Файл <code>C:/tut/wx/inc/mainframe.hpp</code>	19
7.1.1 Подробное описание	19
Предметный указатель	21

Глава 1

Приложение

Состоит из:

- Цикл главного приложения [MyApp](#)

Глава 2

MainFrame

Глава 3

Иерархический список классов

3.1 Иерархия классов

Иерархия классов.

_modbus	11
_modbus_backend	11
_modbus_mapping_t	12
_modbus_rtu	12
_modbus_tcp	13
_modbus_tcp_pi	13
_sft	13
Interruptable	14
Adder	13
Subber	18
InterruptableClasses< x >	15
InterruptableClasses< 1 >	15
InterruptableClasses< 2 >	15
InterruptManager	15
wxApp	
MyApp	17
wxDialog	
Dial	14
wxFrame	
mainFrame	15

Глава 4

Алфавитный указатель классов

4.1 Классы

Классы с их кратким описанием.

<code>_modbus</code>	11
<code>_modbus_backend</code>	11
<code>_modbus_mapping_t</code>	12
<code>_modbus_rtu</code>	12
<code>_modbus_tcp</code>	13
<code>_modbus_tcp_pi</code>	13
<code>_sft</code>	13
<code>Adder</code>	13
<code>Dial</code>	14
<code>Interruptable</code>	14
<code>InterruptableClasses< x ></code>	15
<code>InterruptManager</code>	15
<code>mainFrame</code>	
Класс главного окна	15
<code>MyApp</code>	
класс приложения ждущего события в бесконечном цикле	17
<code>Subber</code>	18

Глава 5

Список файлов

5.1 Файлы

Полный список документированных файлов.

C:/tut/wx/inc/classes.hpp	??
C:/tut/wx/inc/config.h	??
C:/tut/wx/inc/dialog.hpp	??
C:/tut/wx/inc/mainframe.hpp файл	19
C:/tut/wx/inc/modbus-private.h	??
C:/tut/wx/inc/modbus-rtu-private.h	??
C:/tut/wx/inc/modbus-rtu.h	??
C:/tut/wx/inc/modbus-tcp-private.h	??
C:/tut/wx/inc/modbus-tcp.h	??
C:/tut/wx/inc/modbus-version.h	??
C:/tut/wx/inc/modbus.h	??

Глава 6

Классы

6.1 Структура `_modbus`

Открытые атрибуты

- `int slave`
- `int s`
- `int debug`
- `int error_recovery`
- `struct timeval response_timeout`
- `struct timeval byte_timeout`
- `struct timeval indication_timeout`
- `const modbus_backend_t * backend`
- `void * backend_data`

Объявления и описания членов структуры находятся в файле:

- `C:/tut/wx/inc/modbus-private.h`

6.2 Структура `_modbus_backend`

Открытые атрибуты

- `unsigned int backend_type`
- `unsigned int header_length`
- `unsigned int checksum_length`
- `unsigned int max_adu_length`
- `int(* set_slave)(modbus_t *ctx, int slave)`
- `int(* build_request_basis)(modbus_t *ctx, int function, int addr, int nb, uint8_t *req)`
- `int(* build_response_basis)(sft_t *sft, uint8_t *rsp)`
- `int(* prepare_response_tid)(const uint8_t *req, int *req_length)`
- `int(* send_msg_pre)(uint8_t *req, int req_length)`
- `ssize_t(* send)(modbus_t *ctx, const uint8_t *req, int req_length)`
- `int(* receive)(modbus_t *ctx, uint8_t *req)`
- `ssize_t(* recv)(modbus_t *ctx, uint8_t *rsp, int rsp_length)`

- `int(* check_integrity)(modbus_t *ctx, uint8_t *msg, const int msg_length)`
- `int(* pre_check_confirmation)(modbus_t *ctx, const uint8_t *req, const uint8_t *rsp, int rsp←_length)`
- `int(* connect)(modbus_t *ctx)`
- `void(* close)(modbus_t *ctx)`
- `int(* flush)(modbus_t *ctx)`
- `int(* select)(modbus_t *ctx, fd_set *rset, struct timeval *tv, int msg_length)`
- `void(* free)(modbus_t *ctx)`

Объявления и описания членов структуры находятся в файле:

- `C:/tut/wx/inc/modbus-private.h`

6.3 Структура `_modbus_mapping_t`

Открытые атрибуты

- `int nb_bits`
- `int start_bits`
- `int nb_input_bits`
- `int start_input_bits`
- `int nb_input_registers`
- `int start_input_registers`
- `int nb_registers`
- `int start_registers`
- `uint8_t * tab_bits`
- `uint8_t * tab_input_bits`
- `uint16_t * tab_input_registers`
- `uint16_t * tab_registers`

Объявления и описания членов структуры находятся в файле:

- `C:/tut/wx/inc/modbus.h`

6.4 Структура `_modbus_rtu`

Открытые атрибуты

- `char * device`
- `int baud`
- `uint8_t data_bit`
- `uint8_t stop_bit`
- `char parity`
- `struct termios old_tios`
- `int confirmation_to_ignore`

Объявления и описания членов структуры находятся в файле:

- `C:/tut/wx/inc/modbus-rtu-private.h`

6.5 Структура `_modbus_tcp`

Открытые атрибуты

- `uint16_t t_id`
- `int port`
- `char ip [16]`

Объявления и описания членов структуры находятся в файле:

- `C:/tut/wx/inc/modbus-tcp-private.h`

6.6 Структура `_modbus_tcp_pi`

Открытые атрибуты

- `uint16_t t_id`
- `int port`
- `char node [1025]`
- `char service [32]`

Объявления и описания членов структуры находятся в файле:

- `C:/tut/wx/inc/modbus-tcp-private.h`

6.7 Структура `_sft`

Открытые атрибуты

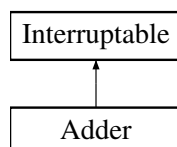
- `int slave`
- `int function`
- `int t_id`

Объявления и описания членов структуры находятся в файле:

- `C:/tut/wx/inc/modbus-private.h`

6.8 Класс `Adder`

Граф наследования: `Adder`:



Открытые члены

- `void run () override`
- `void add ()`

Открытые атрибуты

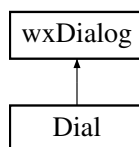
- `int * x =nullptr`

Объявления и описания членов класса находятся в файле:

- `C:/tut/wx/inc/classes.hpp`

6.9 Класс Dial

Граф наследования:Dial:



Открытые члены

- `Dial (const wxString &title)`
- `void onBut1Click (wxCommandEvent &event)`
- `void onBut2Click (wxCommandEvent &event)`

Статические открытые данные

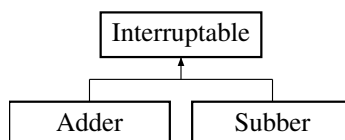
- `static constexpr uint8_t TIMER_ID =1`
- `static constexpr uint8_t TIMER_ID1 =2`

Объявления и описания членов классов находятся в файлах:

- `C:/tut/wx/inc/dialog.hpp`
- `C:/tut/wx/src/dialog.cpp`

6.10 Класс Interruptable

Граф наследования:Interruptable:



Открытые члены

- virtual void run ()=0

Объявления и описания членов класса находятся в файле:

- C:/tut/wx/inc/classes.hpp

6.11 Шаблон класса InterruptableClasses< x >

Открытые статические члены

- static void setVector ()
- static void addClass (Interruptable *classs)
- static void InterruptHandler ()

Объявления и описания членов класса находятся в файле:

- C:/tut/wx/inc/classes.hpp

6.12 Класс InterruptManager

Открытые статические члены

- static void AddHandler (void(*Fptr)(), int x)
- static void call (int x)

Статические открытые данные

- static void(* IsrVi [10])() = {nullptr}

Объявления и описания членов класса находятся в файле:

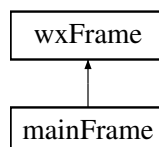
- C:/tut/wx/inc/classes.hpp

6.13 Класс mainFrame

Класс главного окна

```
#include <mainframe.hpp>
```

Граф наследования: mainFrame:



Открытые члены

- `mainFrame` (`const wxString &title`)
Конструктор главного окна
- `mainFrame` (`const mainFrame &`)=`delete`
запрет конструктора копирования
- `mainFrame & operator=` (`const mainFrame &`)=`delete`
запрет конструктора перемещения
- `void onButExit` (`wxCommandEvent &event`)

Открытые атрибуты

- `Dial * d` {`nullptr`}
Указатель на диалоговое окно
- `uint16_t holdRegs` [10] = {0}

6.13.1 Подробное описание

Класс главного окна

Дата

24.06.2020

Предупреждения

alpha version

6.13.2 Конструктор(ы)

6.13.2.1 mainFrame()

```
mainFrame::mainFrame (
    const wxString & title )
```

Конструктор главного окна

```
ctx = modbus_new_rtu("\\\\.\\COM11", 9600, 'N', 8, 1); //none parity 8-bits 1-stop bits
```

[Str]

[Str]

6.13.3 Методы

6.13.3.1 onButExit()

```
void mainFrame::onButExit (
    wxCommandEvent & event )
```

Метод срабатывает по событию нажатия на кнопку.

Аргументы

in	event	
----	-------	--

Возвращает

{void}

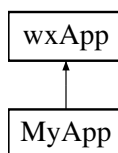
Объявления и описания членов классов находятся в файлах:

- C:/tut/wx/inc/[mainframe.hpp](#)
- C:/tut/wx/src/mainframe.cpp

6.14 Класс MyApp

класс приложения ждущего события в бесконечном цикле

Граф наследования: MyApp:



Открытые члены

- virtual bool [OnInit](#) ()

6.14.1 Подробное описание

класс приложения ждущего события в бесконечном цикле

- Главное окно [MainFrame](#)

6.14.2 Методы

6.14.2.1 OnInit()

```
virtual bool MyApp::OnInit ( ) [inline], [virtual]
```

Создание главного окна

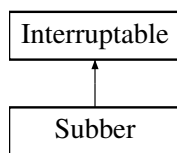
Вызов главного окна

Объявления и описания членов класса находятся в файле:

- C:/tut/wx/src/main.cpp

6.15 Класс Subber

Граф наследования:Subber:



Открытые члены

- void run () override
- void sub ()

Открытые атрибуты

- int * x =nullptr

Объявления и описания членов класса находятся в файле:

- C:/tut/wx/inc/classes.hpp

Глава 7

Файлы

7.1 Файл C:/tut/wx/inc/mainframe.hpp

файл

```
#include "dialog.hpp"  
#include "modbus.h"  
#include "modbus-rtu.h"  
#include "modbus-private.h"  
#include "modbus-rtu-private.h"
```

Классы

- class `mainFrame`
Класс главного окна

7.1.1 Подробное описание

файл

Предметный указатель

- `_modbus`, [11](#)
- `_modbus_backend`, [11](#)
- `_modbus_mapping_t`, [12](#)
- `_modbus_rtu`, [12](#)
- `_modbus_tcp`, [13](#)
- `_modbus_tcp_pi`, [13](#)
- `_sft`, [13](#)

Adder, [13](#)

C:/tut/wx/inc/mainframe.hpp, [19](#)

Dial, [14](#)

Interruptable, [14](#)

InterruptableClasses< x >, [15](#)

InterruptManager, [15](#)

mainFrame, [15](#)

- mainFrame, [16](#)

- onButExit, [16](#)

MyApp, [17](#)

- OnInit, [17](#)

onButExit

- mainFrame, [16](#)

OnInit

- MyApp, [17](#)

Subber, [18](#)