

Obloq

制作者 Doxygen 1.8.13

Contents

1	类索引	1
1.1	类列表	1
2	文件索引	3
2.1	文件列表	3
3	类说明	5
3.1	Obloq类 参考	5
3.1.1	构造及析构函数说明	6
3.1.1.1	Obloq() [1/2]	6
3.1.1.2	Obloq() [2/2]	6
3.1.1.3	~Obloq()	6
3.1.2	成员函数说明	7
3.1.2.1	connect() [1/2]	7
3.1.2.2	connect() [2/2]	7
3.1.2.3	disconnect()	8
3.1.2.4	getWifiState()	8
3.1.2.5	isSerialReady()	8
3.1.2.6	ping()	9
3.1.2.7	publish()	9
3.1.2.8	receiveData()	10
3.1.2.9	reconnect()	10
3.1.2.10	sendMsg()	10
3.1.2.11	setHandleJson()	11

3.1.2.12	setHandleRaw()	11
3.1.2.13	subscribe()	11
3.1.2.14	update()	12
3.1.3	类成员变量说明	12
3.1.3.1	_handleJson	12
3.1.3.2	_handleRaw	12
3.1.3.3	_hardSerial	12
3.1.3.4	_interval	12
3.1.3.5	_isSerialReady	13
3.1.3.6	_mac	13
3.1.3.7	_port	13
3.1.3.8	_pwd	13
3.1.3.9	_softSerial	13
3.1.3.10	_ssid	13
3.1.3.11	_time	13
3.1.3.12	_url	13
3.1.3.13	_wifiState	13
4	文件说明	15
4.1	C:/WorkFile/GitFile/GitHub/OBLOQ/Obloq/Obloq.cpp 文件参考	15
4.2	C:/WorkFile/GitFile/GitHub/OBLOQ/Obloq/Obloq.h 文件参考	15
4.2.1	宏定义说明	15
4.2.1.1	INTERVAL	16
4.2.2	类型定义说明	16
4.2.2.1	HandleJson	16
4.2.2.2	HandleRaw	16
	索引	17

Chapter 1

类索引

1.1 类列表

这里列出了所有类、结构、联合以及接口定义等，并附带简要说明：

Obloq	5
-----------------------------	---

Chapter 2

文件索引

2.1 文件列表

这里列出了所有文件，并附带简要说明：

C:/WorkFile/GitFile/GitHub/OBLOQ/Obloq/Obloq.cpp	15
C:/WorkFile/GitFile/GitHub/OBLOQ/Obloq/Obloq.h	15

Chapter 3

类说明

3.1 Obloq类 参考

```
#include <Obloq.h>
```

Public 成员函数

- **Obloq** (HardwareSerial &hardSerial, String ssid, String pwd="", String mac="")
- **Obloq** (SoftwareSerial &softSerial, String ssid, String pwd="", String mac="")
- **~Obloq** ()
- void **setHandleJson** (**HandleJson**)
设置处理*Obloq*返回数据的回调函数
- void **setHandleRaw** (**HandleRaw**)
设置回显原始数据的回调函数
- bool **isSerialReady** ()
监测串口是否准备好
- int **getWifiState** ()
获取*wifi*连接状态
- void **ping** ()
发送串口敲门指令
- void **update** ()
*Obloq*循环监测和处理串口数据
- void **connect** (const String &client_id, const String &iot_id, const String &iot_pwd)
连接*MQTT*, 默认: *url* = "iot.dfrobot.com.cn", *port* = "1883"
- void **connect** (const String &client_id, const String &iot_id, const String &iot_pwd, const String &url, const String &port)
连接*MQTT*
- void **reconnect** ()
重新连接*MQTT*
- void **subscribe** (const String &topic)
注册设备
- void **publish** (const String &topic, const String &message)
向特定设备发送消息数据
- void **disconnect** ()
断开*MQTT*

Private 成员函数

- void `receiveData` (String)
处理接收的消息数据
- bool `sendMsg` (const String &)
发送消息数据

Private 属性

- HardwareSerial * `_hardSerial` = NULL
- SoftwareSerial * `_softSerial` = NULL
- HandleJson `_handleJson` = NULL
- HandleRaw `_handleRaw` = NULL
- int `_wifiState` = -1
- int `_interval` = INTERVAL
- bool `_isSerialReady` = false
- unsigned long `_time` = 0
- String `_ssid` = ""
- String `_pwd` = ""
- String `_mac` = ""
- String `_url` = "iot.dfrobot.com.cn"
- String `_port` = "1883"

3.1.1 构造及析构函数说明

3.1.1.1 Obloq() [1/2]

```
Obloq::Obloq (
    HardwareSerial & hardSerial,
    String ssid,
    String pwd = "",
    String mac = "" )
```

3.1.1.2 Obloq() [2/2]

```
Obloq::Obloq (
    SoftwareSerial & softSerial,
    String ssid,
    String pwd = "",
    String mac = "" )
```

3.1.1.3 ~Obloq()

```
Obloq::~~Obloq ( )
```

3.1.2 成员函数说明

3.1.2.1 connect() [1/2]

```
void Obloq::connect (
    const String & client_id,
    const String & iot_id,
    const String & iot_pwd )
```

连接MQTT，默认：url = "iot.dfrobot.com.cn"，port= "1883"

参数

<i>client_id</i>	物联网client_id，填写唯一的英文标识符即可
<i>iot_id</i>	物联网iot_id，从物联网网址上直接复制
<i>iot_pwd</i>	物联网iot_pwd，从物联网网址上直接复制

返回

无

3.1.2.2 connect() [2/2]

```
void Obloq::connect (
    const String & client_id,
    const String & iot_id,
    const String & iot_pwd,
    const String & url,
    const String & port )
```

连接MQTT

参数

<i>client_id</i>	物联网client_id，填写唯一的英文标识符即可
<i>iot_id</i>	物联网iot_id，从物联网网址上直接复制
<i>iot_pwd</i>	物联网iot_pwd，从物联网网址上直接复制
<i>url</i>	连接的物联网url
<i>port</i>	连接的物联网端口

返回

无

3.1.2.3 disconnect()

```
void Obloq::disconnect ( )
```

断开MQTT

参数

无	
---	--

返回

无

3.1.2.4 getWifiState()

```
int Obloq::getWifiState ( )
```

获取wifi连接状态

参数

无	
---	--

返回

- 1: wifi断开连接
- 1: wifi连接中
- 2: wifi连接成功, 已获取Obloq的IP
- 3: wifi处于未连接状态

3.1.2.5 isSerialReady()

```
bool Obloq::isSerialReady ( )
```

监测串口是否准备好

参数

无	
---	--

返回

true:表示串口准备好, **false:**表示串口木有准备好

3.1.2.6 ping()

```
void Obloq::ping ( )
```

发送串口敲门指令

参数

无	
---	--

返回

无

3.1.2.7 publish()

```
void Obloq::publish (
    const String & topic,
    const String & message )
```

向特定设备发送消息数据

参数

<i>topic</i> :设备ID	
--------------------	--

返回

message: 发送的消息内容

3.1.2.8 receiveData()

```
void Obloq::receiveData (
    String data ) [private]
```

处理接收的消息数据

参数

<i>data</i> :接收的消息数据	
----------------------	--

返回

无

3.1.2.9 reconnect()

```
void Obloq::reconnect ( )
```

重新连接MQTT

参数

无	
---	--

返回

无

3.1.2.10 sendMsg()

```
bool Obloq::sendMsg (
    const String & msg ) [private]
```

发送消息数据

参数

<i>msg</i> :发送的消息内容	
---------------------	--

返回

true:表示发送成功，**false**:表示发送失败

3.1.2.11 setHandleJson()

```
void Obloq::setHandleJson (
    HandleJson handle )
```

设置处理Obloq返回数据的回掉函数

参数

<i>handle</i> :回掉函数	
---------------------	--

返回

无

3.1.2.12 setHandleRaw()

```
void Obloq::setHandleRaw (
    HandleRaw handle )
```

设置回显原始数据的回掉函数

参数

<i>handle</i> :回掉函数	
---------------------	--

返回

无

3.1.2.13 subscribe()

```
void Obloq::subscribe (
    const String & topic )
```

注册设备

参数

<i>topic</i> :设备ID	
--------------------	--

返回

无

3.1.2.14 update()

```
void Obloq::update ( )
```

Obloq循环监测和处理串口数据

参数

无	
---	--

返回

无

3.1.3 类成员变量说明

3.1.3.1 _handleJson

```
HandleJson Obloq::_handleJson = NULL [private]
```

3.1.3.2 _handleRaw

```
HandleRaw Obloq::_handleRaw = NULL [private]
```

3.1.3.3 _hardSerial

```
HardwareSerial* Obloq::_hardSerial = NULL [private]
```

3.1.3.4 _interval

```
int Obloq::_interval = INTERVAL [private]
```


3.1.3.5 _isSerialReady

```
bool Obloq::_isSerialReady = false [private]
```

3.1.3.6 _mac

```
String Obloq::_mac = "" [private]
```

3.1.3.7 _port

```
String Obloq::_port = "1883" [private]
```

3.1.3.8 _pwd

```
String Obloq::_pwd = "" [private]
```

3.1.3.9 _softSerial

```
SoftwareSerial* Obloq::_softSerial = NULL [private]
```

3.1.3.10 _ssid

```
String Obloq::_ssid = "" [private]
```

3.1.3.11 _time

```
unsigned long Obloq::_time = 0 [private]
```

3.1.3.12 _url

```
String Obloq::_url = "iot.dfrobot.com.cn" [private]
```

3.1.3.13 _wifiState

```
int Obloq::_wifiState = -1 [private]
```

该类的文档由以下文件生成:

- [C:/WorkFile/GitFile/GitHub/OBLOQ/Obloq/Obloq.h](#)
- [C:/WorkFile/GitFile/GitHub/OBLOQ/Obloq/Obloq.cpp](#)

Chapter 4

文件说明

4.1 C:/WorkFile/GitFile/GitHub/OBLOQ/Obloq/Obloq.cpp 文件参考

```
#include <Arduino.h>
#include <SoftwareSerial.h>
#include <HardwareSerial.h>
#include <ArduinoJson.h>
#include "Obloq.h"
```

Obloq.cpp 的引用(Include)关系图:

4.2 C:/WorkFile/GitFile/GitHub/OBLOQ/Obloq/Obloq.h 文件参考

此图展示该文件直接或间接的被哪些文件引用了:

类

- class [Obloq](#)

宏定义

- `#define` [INTERVAL](#) 1000

类型定义

- `typedef void`(* [HandleJson](#)) (JsonObject &data)
- `typedef void`(* [HandleRaw](#)) (const String &data)

4.2.1 宏定义说明

4.2.1.1 INTERVAL

```
#define INTERVAL 1000
```

4.2.2 类型定义说明

4.2.2.1 HandleJson

```
typedef void(* HandleJson) (JsonObject &data)
```

4.2.2.2 HandleRaw

```
typedef void(* HandleRaw) (const String &data)
```

Index

- [_handleJson](#)
 - [Obloq, 12](#)
 - [_handleRaw](#)
 - [Obloq, 12](#)
 - [_hardSerial](#)
 - [Obloq, 12](#)
 - [_interval](#)
 - [Obloq, 12](#)
 - [_isSerialReady](#)
 - [Obloq, 12](#)
 - [_mac](#)
 - [Obloq, 13](#)
 - [_port](#)
 - [Obloq, 13](#)
 - [_pwd](#)
 - [Obloq, 13](#)
 - [_softSerial](#)
 - [Obloq, 13](#)
 - [_ssid](#)
 - [Obloq, 13](#)
 - [_time](#)
 - [Obloq, 13](#)
 - [_url](#)
 - [Obloq, 13](#)
 - [_wifiState](#)
 - [Obloq, 13](#)
- [~Obloq](#)
 - [Obloq, 6](#)
- [C:/WorkFile/GitFile/GitHub/OBLOQ/Obloq/Obloq.cpp,](#)
 - [15](#)
- [C:/WorkFile/GitFile/GitHub/OBLOQ/Obloq/Obloq.h, 15](#)
- [connect](#)
 - [Obloq, 7](#)
- [disconnect](#)
 - [Obloq, 8](#)
- [getWifiState](#)
 - [Obloq, 8](#)
- [HandleJson](#)
 - [Obloq.h, 16](#)
- [HandleRaw](#)
 - [Obloq.h, 16](#)
- [INTERVAL](#)
 - [Obloq.h, 15](#)
- [isSerialReady](#)
 - [Obloq, 8](#)
- [Obloq, 5](#)
 - [_handleJson, 12](#)
 - [_handleRaw, 12](#)
 - [_hardSerial, 12](#)
 - [_interval, 12](#)
 - [_isSerialReady, 12](#)
 - [_mac, 13](#)
 - [_port, 13](#)
 - [_pwd, 13](#)
 - [_softSerial, 13](#)
 - [_ssid, 13](#)
 - [_time, 13](#)
 - [_url, 13](#)
 - [_wifiState, 13](#)
 - [~Obloq, 6](#)
 - [connect, 7](#)
 - [disconnect, 8](#)
 - [getWifiState, 8](#)
 - [isSerialReady, 8](#)
 - [Obloq, 6](#)
 - [ping, 9](#)
 - [publish, 9](#)
 - [receiveData, 9](#)
 - [reconnect, 10](#)
 - [sendMsg, 10](#)
 - [setHandleJson, 11](#)
 - [setHandleRaw, 11](#)
 - [subscribe, 11](#)
 - [update, 12](#)
- [Obloq.h](#)
 - [HandleJson, 16](#)
 - [HandleRaw, 16](#)
 - [INTERVAL, 15](#)
- [ping](#)
 - [Obloq, 9](#)
- [publish](#)
 - [Obloq, 9](#)
- [receiveData](#)
 - [Obloq, 9](#)
- [reconnect](#)
 - [Obloq, 10](#)
- [sendMsg](#)
 - [Obloq, 10](#)
- [setHandleJson](#)
 - [Obloq, 11](#)
- [setHandleRaw](#)
 - [Obloq, 11](#)

subscribe

Obloq, [11](#)

update

Obloq, [12](#)