

# Aqua Control

1.0

Generated by Doxygen 1.8.17



<b>1 Todo List</b>	<b>1</b>
<b>2 Bug List</b>	<b>3</b>
<b>3 Class Index</b>	<b>5</b>
3.1 Class List . . . . .	5
<b>4 File Index</b>	<b>7</b>
4.1 File List . . . . .	7
<b>5 Class Documentation</b>	<b>9</b>
5.1 httpUpdater Class Reference . . . . .	9
5.1.1 Detailed Description . . . . .	9
5.2 RGB Struct Reference . . . . .	9
5.2.1 Detailed Description . . . . .	10
5.2.2 Member Data Documentation . . . . .	10
5.2.2.1 b . . . . .	10
5.2.2.2 g . . . . .	10
5.2.2.3 r . . . . .	10
5.3 serverA Class Reference . . . . .	10
5.3.1 Detailed Description . . . . .	10
<b>6 File Documentation</b>	<b>13</b>
6.1 main.cpp File Reference . . . . .	13
6.1.1 Detailed Description . . . . .	14
6.1.2 Function Documentation . . . . .	15
6.1.2.1 handleAction() . . . . .	15
6.1.2.2 handleCss() . . . . .	15
6.1.2.3 handleJs() . . . . .	16
6.1.2.4 handleNotFound() . . . . .	16
6.1.2.5 handleRoot() . . . . .	16
6.1.2.6 led() . . . . .	18
6.1.2.7 loop() . . . . .	18
6.1.2.8 readFile() . . . . .	18
6.1.2.9 serverA() . . . . .	19
6.1.2.10 setup() . . . . .	19
6.1.2.11 wait() . . . . .	19
6.1.3 Variable Documentation . . . . .	20
6.1.3.1 color . . . . .	20
6.1.3.2 hand . . . . .	20
6.1.3.3 host . . . . .	20
6.1.3.4 httpUpdater . . . . .	20
6.1.3.5 page . . . . .	21
6.1.3.6 pass . . . . .	21

6.1.3.7 PASSWD . . . . .	21
6.1.3.8 pinB . . . . .	21
6.1.3.9 pinG . . . . .	21
6.1.3.10 pinR . . . . .	22
6.1.3.11 previusTime . . . . .	22
6.1.3.12 pump . . . . .	22
6.1.3.13 setTime . . . . .	22
6.1.3.14 SSID . . . . .	22
6.1.3.15 sunLight . . . . .	23
6.1.3.16 update_path . . . . .	23
6.1.3.17 uploadFile . . . . .	23
6.1.3.18 user . . . . .	23

<b>Index</b>	<b>25</b>
--------------	-----------

# Chapter 1

## Todo List

File [main.cpp](#)

Implementation List

- Turn on Led
- OnOff Pump



## Chapter 2

# Bug List

File [main.cpp](#)

Bugs list





## Chapter 3

# Class Index

### 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">httpUpdater</a>	Update class . . . . .	9
<a href="#">RGB</a>	Struct for Led . . . . .	9
<a href="#">serverA</a>	WebServer Class . . . . .	10



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

<a href="#">main.cpp</a>	Controls pool pump and <a href="#">RGB</a> led . . . . .	<a href="#">13</a>
--------------------------	--	--------------------



## Chapter 5

# Class Documentation

### 5.1 httpUpdater Class Reference

Update class.

#### 5.1.1 Detailed Description

Update class.

provide update for firmware on the http request

See also

other classes

- [serverA](#)
- [RGB](#)

### 5.2 RGB Struct Reference

Struct for Led.

#### Public Attributes

- byte [r](#)  
*RECEIVE BYTE COLOR RED.*
- byte [g](#)  
*RECEIVE BYTE COLOR GREEN.*
- byte [b](#)

### 5.2.1 Detailed Description

Struct for Led.

Generates struct for rgb color object

See also

other classes

- [serverA](#)
- [httpUpdater](#)

Definition at line 93 of file main.cpp.

### 5.2.2 Member Data Documentation

#### 5.2.2.1 b

`byte RGB::b`

RECEIVE BYTE COLOR BLUE

Definition at line 100 of file main.cpp.

#### 5.2.2.2 g

`byte RGB::g`

RECEIVE BYTE COLOR GREEN.

Definition at line 98 of file main.cpp.

#### 5.2.2.3 r

`byte RGB::r`

RECEIVE BYTE COLOR RED.

Definition at line 96 of file main.cpp.

## 5.3 serverA Class Reference

WebServer Class.

### 5.3.1 Detailed Description

WebServer Class.

This create a object for handle request response of client

## Parameters

80	- Web port
----	------------

## See also

other classes

- [httpUpdater](#)
- [RGB](#)





## Chapter 6

# File Documentation

### 6.1 main.cpp File Reference

Controls pool pump and [RGB](#) led.

```
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <FS.h>
#include <LittleFS.h>
#include <ESP8266mDNS.h>
#include <ESP8266HTTPUpdateServer.h>
```

#### Classes

- struct [RGB](#)  
*Struct for Led.*

#### Functions

- ESP8266WebServer [serverA](#) (80)
- void [handleRoot](#) ()  
*manipulates index.htm file*
- void [handleCss](#) ()  
*Set stylesheet of page.*
- void [handleJs](#) ()  
*set script of page*
- void [handleAction](#) ()  
*function to do action requested by client*
- void [handleNotFound](#) ()  
*handle error page*
- void [led](#) ([RGB](#) colorname)  
*Set color led.*
- bool [wait](#) ()  
*Wait for set time.*
- String [readFile](#) (String path)  
*Read File.*
- void [setup](#) ()  
*Setting source.*
- void [loop](#) ()

## Variables

- `const char * user = "admin"`
- `const char * pass = "senha123"`
- `const char * host = "aqua"`
- `const char * update_path = "/firmware"`
- `const char * SSID = "YOURSSID"`  
*Ssid Network.*
- `const char * PASSWD = "YOURPASSWORD"`  
*Password for Network.*
- `bool hand = true`
- `unsigned long setTime = 1000`
- `unsigned long previusTime = 0`
- File `uploadFile`
- `int pinR = 12`
- `int pinG = 13`
- `int pinB = 14`
- `int pump = 5`  
*define pin of pump*
- `int sunLight = 4`
- `String page = ""`  
*define ldr sensor*
- `ESP8266HTTPUpdateServer httpUpdater`  
*INSTANCE FOR HTTP UPDATE.*
- `RGB color = {0,0,0}`

### 6.1.1 Detailed Description

Controls pool pump and RGB led.

#### Author

Cristiano Rocha

#### Warning

This source is only first implementation

#### Todo Implementation List

- Turn on Led
- OnOff Pump

#### Bug Bugs list

#### Copyright

RC Automação all copyrights reserved

## 6.1.2 Function Documentation

### 6.1.2.1 `handleAction()`

```
void handleAction ( )
```

function to do action requested by client

handles customer request

```
    turn on led, turn on pump and all actions
```

#### Parameters

-	void
---	------

#### Returns

void

On Off Pump

Definition at line 351 of file main.cpp.

### 6.1.2.2 `handleCss()`

```
void handleCss ( )
```

Set stylesheet of page.

set style page

```
    set style on page with file.css
```

#### Parameters

-	void parameters
---	-----------------

#### Returns

void

Definition at line 330 of file main.cpp.

### 6.1.2.3 handleJs()

```
void handleJs ( )
```

set script of page

Javascript source.

```
set script on the page whit file.js
```

#### Parameters

-	void parameters
---	-----------------

#### Returns

void

Definition at line 337 of file main.cpp.

### 6.1.2.4 handleNotFound()

```
void handleNotFound ( )
```

handle error page

handle error 404

```
response for request error
```

#### Parameters

-	void
---	------

#### Returns

void

Definition at line 344 of file main.cpp.

### 6.1.2.5 handleRoot()

```
void handleRoot ( )
```

manipulates index.htm file

send response a first requisition

**Parameters**

-	void parameters
---	-----------------

**Returns**

void

Definition at line 323 of file main.cpp.

**6.1.2.6 led()**

```
void led (
    RGB colorname )
```

Set color led.

turn on led in the desired color

receive color for turn on led

**Parameters**

<i>colorname</i>	- struct type {r,g,b}
------------------	-----------------------

**Returns**

void

Definition at line 300 of file main.cpp.

**6.1.2.7 loop()**

```
void loop ( )
```

Definition at line 258 of file main.cpp.

**6.1.2.8 readFile()**

```
String readFile (
    String path )
```

Read File.

Read file SPIFFS.

Open and read all files in the flash system

**6.1.2.8.1 Variables** **String** *content* — //return variable  
**File** *file* — //add file data

#### Parameters

<i>path</i>	- File name
-------------	-------------

#### Returns

String read from file

Definition at line 309 of file main.cpp.

### 6.1.2.9 serverA()

```
ESP8266WebServer serverA (  
    80 )
```

### 6.1.2.10 setup()

```
setup ( )
```

Setting source.

```
put all configuration setup of device
```

Definition at line 206 of file main.cpp.

### 6.1.2.11 wait()

```
bool wait ( )
```

Wait for set time.

generates delay without locking the code

```
set delay with millis() function
```

#### Parameters

-	void
---	------

**Returns**

Boolean type

Definition at line 287 of file main.cpp.

### 6.1.3 Variable Documentation

#### 6.1.3.1 color

```
RGB color = {0,0,0}
```

object [RGB](#) color

Definition at line 103 of file main.cpp.

#### 6.1.3.2 hand

```
bool hand = true
```

handle manual or automatic action

Definition at line 43 of file main.cpp.

#### 6.1.3.3 host

```
const char* host = "aqua"
```

hostname for local access

Definition at line 39 of file main.cpp.

#### 6.1.3.4 httpUpdater

```
ESP8266HTTPUpdateServer httpUpdater
```

INSTANCE FOR HTTP UPDATE.

Definition at line 68 of file main.cpp.



#### 6.1.3.5 page

```
String page = ""
```

define ldr sensor

Store page HTML

Definition at line 52 of file main.cpp.

#### 6.1.3.6 pass

```
const char* pass = "senha123"
```

paasword for access

Definition at line 38 of file main.cpp.

#### 6.1.3.7 PASSWD

```
const char* PASSWD = "YOURPASSWORD"
```

Password for Network.

Definition at line 42 of file main.cpp.

#### 6.1.3.8 pinB

```
int pinB = 14
```

Set pin blue as number 14

Definition at line 49 of file main.cpp.

#### 6.1.3.9 pinG

```
int pinG = 13
```

Set pin green as number 13

Definition at line 48 of file main.cpp.

#### 6.1.3.10 pinR

```
int pinR = 12
```

Set pin red as number 12

Definition at line 47 of file main.cpp.

#### 6.1.3.11 previusTime

```
unsigned long previusTime = 0
```

receive previus time for trigger delay

Definition at line 45 of file main.cpp.

#### 6.1.3.12 pump

```
int pump = 5
```

define pin of pump

Definition at line 50 of file main.cpp.

#### 6.1.3.13 setTime

```
unsigned long setTime = 1000
```

set time delay

Definition at line 44 of file main.cpp.

#### 6.1.3.14 SSID

```
const char* SSID = "YOURSSID"
```

Ssid Network.

Definition at line 41 of file main.cpp.

#### 6.1.3.15 sunLight

```
int sunLight = 4
```

Definition at line 51 of file main.cpp.

#### 6.1.3.16 update\_path

```
const char* update_path = "/firmware"
```

Path for update

Definition at line 40 of file main.cpp.

#### 6.1.3.17 uploadFile

```
File uploadFile
```

Store data for SPIFFS readed

Definition at line 46 of file main.cpp.

#### 6.1.3.18 user

```
const char* user = "admin"
```

username for login

Definition at line 37 of file main.cpp.



# Index

- b
  - RGB, [10](#)
- color
  - main.cpp, [20](#)
- g
  - RGB, [10](#)
- hand
  - main.cpp, [20](#)
- handleAction
  - main.cpp, [15](#)
- handleCss
  - main.cpp, [15](#)
- handleJs
  - main.cpp, [15](#)
- handleNotFound
  - main.cpp, [16](#)
- handleRoot
  - main.cpp, [16](#)
- host
  - main.cpp, [20](#)
- httpUpdater, [9](#)
  - main.cpp, [20](#)
- led
  - main.cpp, [18](#)
- loop
  - main.cpp, [18](#)
- main.cpp, [13](#)
  - color, [20](#)
  - hand, [20](#)
  - handleAction, [15](#)
  - handleCss, [15](#)
  - handleJs, [15](#)
  - handleNotFound, [16](#)
  - handleRoot, [16](#)
  - host, [20](#)
  - httpUpdater, [20](#)
  - led, [18](#)
  - loop, [18](#)
  - page, [20](#)
  - pass, [21](#)
  - PASSWD, [21](#)
  - pinB, [21](#)
  - pinG, [21](#)
  - pinR, [21](#)
  - previusTime, [22](#)
  - pump, [22](#)
  - readFile, [18](#)
  - serverA, [19](#)
  - setTime, [22](#)
  - setup, [19](#)
  - SSID, [22](#)
  - sunLight, [22](#)
  - update\_path, [23](#)
  - uploadFile, [23](#)
  - user, [23](#)
  - wait, [19](#)
- page
  - main.cpp, [20](#)
- pass
  - main.cpp, [21](#)
- PASSWD
  - main.cpp, [21](#)
- pinB
  - main.cpp, [21](#)
- pinG
  - main.cpp, [21](#)
- pinR
  - main.cpp, [21](#)
- previusTime
  - main.cpp, [22](#)
- pump
  - main.cpp, [22](#)
- r
  - RGB, [10](#)
- readFile
  - main.cpp, [18](#)
- RGB, [9](#)
  - b, [10](#)
  - g, [10](#)
  - r, [10](#)
- serverA, [10](#)
  - main.cpp, [19](#)
- setTime
  - main.cpp, [22](#)
- setup
  - main.cpp, [19](#)
- SSID
  - main.cpp, [22](#)
- sunLight
  - main.cpp, [22](#)
- update\_path
  - main.cpp, [23](#)

uploadFile  
    main.cpp, [23](#)  
user  
    main.cpp, [23](#)  
  
wait  
    main.cpp, [19](#)