

Alarme connectée

Généré par Doxygen 1.9.6

1 Index hiérarchique	1
1.1 Hiérarchie des classes	1
2 Index des classes	3
2.1 Liste des classes	3
3 Index des fichiers	5
3.1 Liste des fichiers	5
4 Documentation des classes	7
4.1 Référence de la classe Bouton	7
4.1.1 Description détaillée	8
4.1.2 Documentation des constructeurs et destructeur	8
4.1.2.1 Bouton()	9
4.1.3 Documentation des fonctions membres	9
4.1.3.1 Get_pin()	9
4.1.3.2 Get_state_sensor()	9
4.1.3.3 init()	9
4.2 Référence de la classe Buzzer	10
4.2.1 Description détaillée	12
4.2.2 Documentation des constructeurs et destructeur	12
4.2.2.1 Buzzer()	12
4.2.3 Documentation des fonctions membres	12
4.2.3.1 Buzz()	12
4.2.3.2 Buzz_off()	12
4.2.3.3 Get_pin()	12
4.2.3.4 init()	13
4.3 Référence de la classe Display	13
4.3.1 Description détaillée	15
4.3.2 Documentation des constructeurs et destructeur	15
4.3.2.1 Display()	15
4.3.2.2 ~Display()	15
4.3.3 Documentation des fonctions membres	15
4.3.3.1 afficher_message()	15
4.3.3.2 clear()	16
4.3.3.3 init()	16
4.4 Référence de la classe Mail	16
4.4.1 Description détaillée	17
4.4.2 Documentation des constructeurs et destructeur	17
4.4.2.1 Mail()	17
4.4.2.2 ~Mail()	17
4.4.3 Documentation des fonctions membres	17
4.4.3.1 sendMail()	17

4.5 Référence de la classe Motion	18
4.5.1 Description détaillée	20
4.5.2 Documentation des constructeurs et destructeur	20
4.5.2.1 Motion()	20
4.5.3 Documentation des fonctions membres	20
4.5.3.1 Get_pin()	20
4.5.3.2 Get_state_sensor()	20
4.5.3.3 init()	21
4.5.3.4 isDetect()	21
4.5.3.5 operator!=(=)	21
4.5.3.6 operator==(=)	21
4.6 Référence de la classe Sensor	22
4.6.1 Description détaillée	23
4.6.2 Documentation des fonctions membres	23
4.6.2.1 Get_pin()	23
4.6.2.2 Get_state_sensor()	23
4.6.2.3 init()	23
4.7 Référence de la classe Wifi	24
4.7.1 Description détaillée	24
4.7.2 Documentation des constructeurs et destructeur	25
4.7.2.1 Wifi()	25
4.7.2.2 ~Wifi()	25
4.7.3 Documentation des fonctions membres	25
4.7.3.1 afficheStatus()	25
4.7.3.2 changePassword()	25
4.7.3.3 changeSsid()	26
4.7.3.4 connect()	26
4.7.3.5 disconnect()	26
4.7.3.6 Get_SSID()	26
4.7.3.7 isConnected()	27
5 Documentation des fichiers	29
5.1 Référence du fichier main/Bouton.cpp	29
5.2 Référence du fichier main/Bouton.hpp	29
5.3 Bouton.hpp	30
5.4 Référence du fichier main/Buzzer.cpp	31
5.5 Référence du fichier main/Buzzer.hpp	31
5.6 Buzzer.hpp	32
5.7 Référence du fichier main/Display.cpp	33
5.8 Référence du fichier main/Display.hpp	33
5.9 Display.hpp	34
5.10 Référence du fichier main/Mail.cpp	35

5.11 Référence du fichier main/Mail.hpp	35
5.11.1 Documentation des macros	36
5.11.1.1 DEFAULT_MAIL_PORT	36
5.11.1.2 DEFAULT_MAIL_RECIPIENT	36
5.11.1.3 DEFAULT_MAIL_SENDER_ADRESS	37
5.11.1.4 DEFAULT_MAIL_SENDER_PASSWORD	37
5.11.1.5 DEFAULT_MAIL_SERVER	37
5.11.1.6 DEFAULT_MAIL_SUBJECT	37
5.11.1.7 MAX_MAIL_ADRESS LENGHT	37
5.11.1.8 MAX_MAIL_PASSWORD LENGHT	37
5.11.1.9 MAX_MAIL_PORT LENGHT	37
5.11.1.10 MAX_MAIL_SERVER LENGHT	37
5.12 Mail.hpp	38
5.13 Référence du fichier main/Motion.cpp	38
5.14 Référence du fichier main/Motion.hpp	39
5.15 Motion.hpp	39
5.16 Référence du fichier main/Sensor.cpp	40
5.17 Référence du fichier main/Sensor.hpp	41
5.18 Sensor.hpp	41
5.19 Référence du fichier main/Wifi.cpp	42
5.20 Référence du fichier main/Wifi.hpp	42
5.20.1 Documentation des macros	43
5.20.1.1 MAX_SSID LENGHT	43
5.20.1.2 MAX_WPA2_PWD LENGHT	43
5.21 Wifi.hpp	44
Index	45

Chapitre 1

Index hiérarchique

1.1 Hiérarchie des classes

Cette liste d'héritage est classée approximativement par ordre alphabétique :

Mail	16
Sensor	22
Bouton	7
Buzzer	10
Display	13
Motion	18
Wifi	24

Chapitre 2

Index des classes

2.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

Bouton	Classe du bouton	7
Buzzer	Classe du buzzer	10
Display	Classe de l'écran	13
Mail	Classe du mail	16
Motion	Classe du capteur de mouvement	18
Sensor	Classe du capteur	22
Wifi	Classe du wifi	24

Chapitre 3

Index des fichiers

3.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

main/Bouton.cpp	29
main/Bouton.hpp	29
main/Buzzer.cpp	31
main/Buzzer.hpp	31
main/Display.cpp	33
main/Display.hpp	33
main/Mail.cpp	35
main/Mail.hpp	35
main/Motion.cpp	38
main/Motion.hpp	39
main/Sensor.cpp	40
main/Sensor.hpp	41
main/Wifi.cpp	42
main/Wifi.hpp	42

Chapitre 4

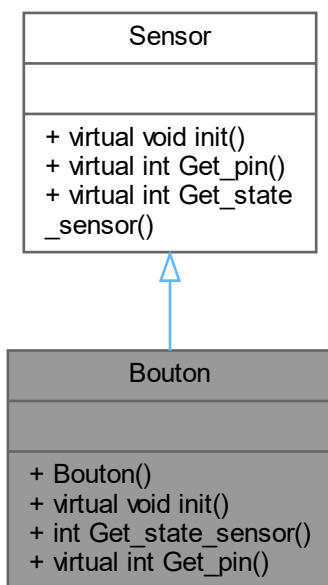
Documentation des classes

4.1 Référence de la classe Bouton

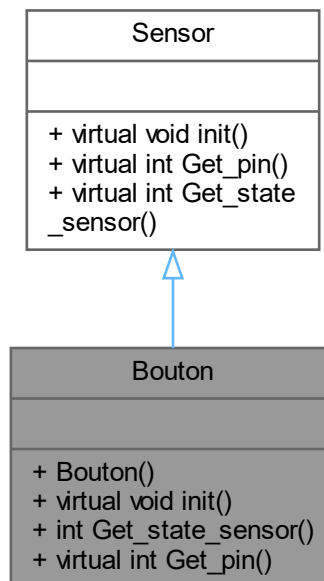
Classe du bouton.

```
#include <Bouton.hpp>
```

Graphe d'héritage de Bouton:



Graphe de collaboration de Bouton:



Fonctions membres publiques

- [Bouton](#) ()
Constructeur par défaut de la classe [Bouton](#).
- virtual void [init](#) ()
Initialisation du bouton, appelé par le constructeur.
- int [Get_state_sensor](#) ()
Récupération de l'état du bouton.
- virtual int [Get_pin](#) ()
Récupération du pin du bouton.

- virtual void [init](#) ()
Initialisation du capteur.
- virtual int [Get_pin](#) ()
Récupération du pin du capteur.
- virtual int [Get_state_sensor](#) ()
Récupération de l'état du capteur.

4.1.1 Description détaillée

Classe du bouton.

4.1.2 Documentation des constructeurs et destructeur

4.1.2.1 Bouton()

```
Bouton::Bouton ( )
```

Constructeur par défaut de la classe [Bouton](#).

4.1.3 Documentation des fonctions membres

4.1.3.1 Get_pin()

```
int Bouton::Get_pin ( ) [virtual]
```

Récupération du pin du bouton.

Renvoie

Le pin du bouton.

Réimplémentée à partir de [Sensor](#).

4.1.3.2 Get_state_sensor()

```
int Bouton::Get_state_sensor ( ) [virtual]
```

Récupération de l'état du bouton.

Renvoie

L'état du bouton.

Réimplémentée à partir de [Sensor](#).

4.1.3.3 init()

```
void Bouton::init ( ) [virtual]
```

Initialisation du bouton, appelé par le constructeur.

Réimplémentée à partir de [Sensor](#).

La documentation de cette classe a été générée à partir des fichiers suivants :

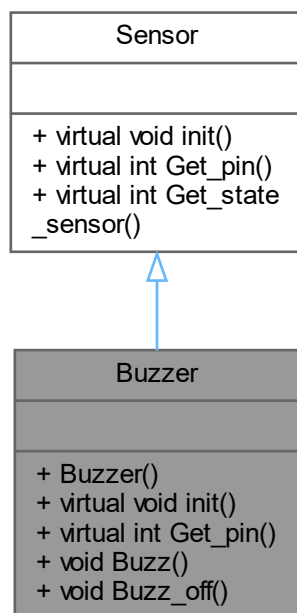
- [main/Bouton.hpp](#)
- [main/Bouton.cpp](#)

4.2 Référence de la classe Buzzer

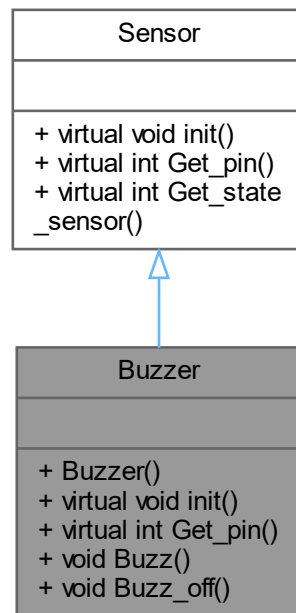
Classe du buzzer.

```
#include <Buzzer.hpp>
```

Graphe d'héritage de Buzzer:



Graphe de collaboration de Buzzer:



Fonctions membres publiques

- `Buzzer ()`
*Constructeur par défaut de la classe **Buzzer**.*
- `virtual void init ()`
Initialisation du buzzer, appelé par le constructeur.
- `virtual int Get_pin ()`
Récupération du pin du buzzer.
- `void Buzz ()`
Fait sonner le buzzer.
- `void Buzz_off ()`

Fonctions membres publiques hérités de **Sensor**

- `virtual void init ()`
Initialisation du capteur.
- `virtual int Get_pin ()`
Récupération du pin du capteur.
- `virtual int Get_state_sensor ()`
Récupération de l'état du capteur.

4.2.1 Description détaillée

Classe du buzzer.

Le buzzer est un capteur qui permet de faire du bruit.

Avertissement

Le buzzer hérite de la classe [Sensor](#).

4.2.2 Documentation des constructeurs et destructeur

4.2.2.1 Buzzer()

```
Buzzer::Buzzer ( )
```

Constructeur par défaut de la classe [Buzzer](#).

4.2.3 Documentation des fonctions membres

4.2.3.1 Buzz()

```
void Buzzer::Buzz ( )
```

Fait sonner le buzzer.

4.2.3.2 Buzz_off()

```
void Buzzer::Buzz_off ( )
```

4.2.3.3 Get_pin()

```
int Buzzer::Get_pin ( ) [virtual]
```

Récupération du pin du buzzer.

Renvoie

Le pin du buzzer.

Réimplémentée à partir de [Sensor](#).

4.2.3.4 init()

```
void Buzzer::init ( ) [virtual]
```

Initialisation du buzzer, appelé par le constructeur.

Réimplémentée à partir de [Sensor](#).

La documentation de cette classe a été générée à partir des fichiers suivants :

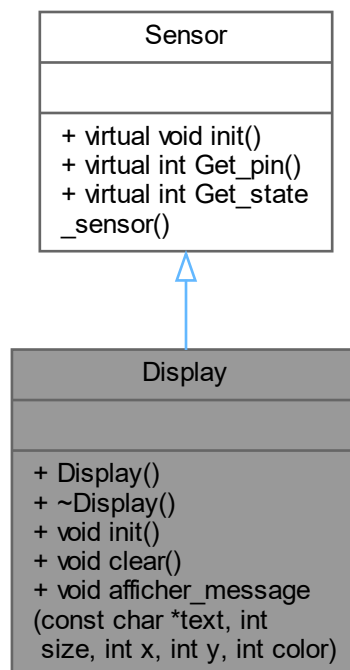
- [main/Buzzer.hpp](#)
- [main/Buzzer.cpp](#)

4.3 Référence de la classe Display

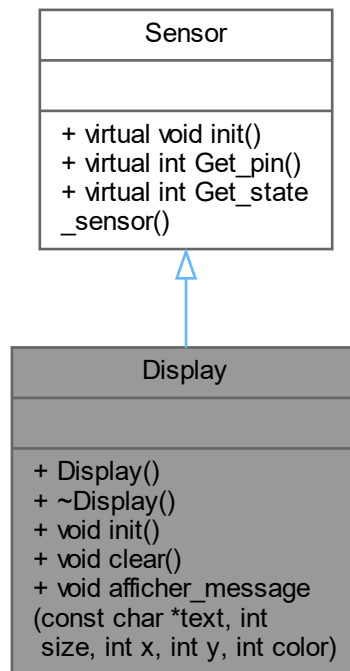
Classe de l'écran.

```
#include <Display.hpp>
```

Graphe d'héritage de Display:



Graphe de collaboration de Display:



Fonctions membres publiques

- `Display ()`
Constructeur par défaut de la classe `Display`.
- `~Display ()`
Destructeur par défaut de la classe `Display`.
- `void init ()`
Initialisation de l'écran, appelé par le constructeur.
- `void clear ()`
Vide l'écran.
- `void afficher_message (const char *text, int size, int x, int y, int color)`
Affiche un message sur l'écran.

Fonctions membres publiques hérités de `Sensor`

- `virtual void init ()`
Initialisation du capteur.
- `virtual int Get_pin ()`
Récupération du pin du capteur.
- `virtual int Get_state_sensor ()`
Récupération de l'état du capteur.

4.3.1 Description détaillée

Classe de l'écran.

L'écran permet d'afficher des messages.

Avertissement

L'écran hérite de la classe [Sensor](#).

4.3.2 Documentation des constructeurs et destructeur

4.3.2.1 Display()

```
Display::Display ( )
```

Constructeur par défaut de la classe [Display](#).

4.3.2.2 ~Display()

```
Display::~~Display ( )
```

Destructeur par défaut de la classe [Display](#).

4.3.3 Documentation des fonctions membres

4.3.3.1 afficher_message()

```
void Display::afficher_message (
    const char * text,
    int size,
    int x,
    int y,
    int color = WHITE )
```

Affiche un message sur l'écran.

4.3.3.2 clear()

```
void Display::clear ( )
```

Vide l'écran.

4.3.3.3 init()

```
void Display::init ( ) [virtual]
```

Initialisation de l'écran, appelé par le constructeur.

Réimplémentée à partir de [Sensor](#).

La documentation de cette classe a été générée à partir des fichiers suivants :

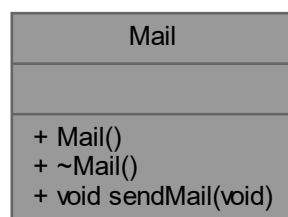
- [main/Display.hpp](#)
- [main/Display.cpp](#)

4.4 Référence de la classe Mail

Classe du mail.

```
#include <Mail.hpp>
```

Graphe de collaboration de Mail:



Fonctions membres publiques

- [Mail](#) ()
Constructeur par défaut de la classe [Mail](#).
- [~Mail](#) ()
Destructeur par défaut de la classe [Mail](#).
- void [sendMail](#) (void)
Permet d'envoyer un mail.

4.4.1 Description détaillée

Classe du mail.

4.4.2 Documentation des constructeurs et destructeur

4.4.2.1 Mail()

```
Mail::Mail ( )
```

Constructeur par défaut de la classe [Mail](#).

Construct a new [Mail:: Mail](#) object.

4.4.2.2 ~Mail()

```
Mail::~~Mail ( )
```

Destructeur par défaut de la classe [Mail](#).

Destroy the [Mail:: Mail](#) object.

4.4.3 Documentation des fonctions membres

4.4.3.1 sendMail()

```
void Mail::sendMail (
    void )
```

Permet d'envoyer un mail.

Send a mail.

La documentation de cette classe a été générée à partir des fichiers suivants :

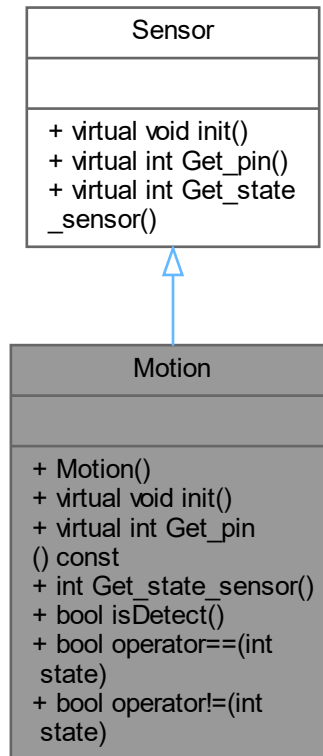
- [main/Mail.hpp](#)
- [main/Mail.cpp](#)

4.5 Référence de la classe Motion

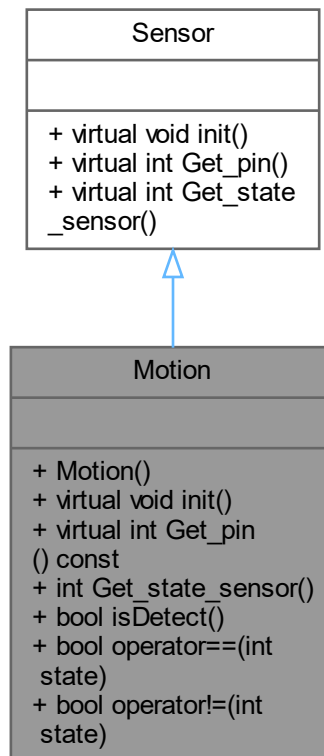
Classe du capteur de mouvement.

```
#include <Motion.hpp>
```

Graphe d'héritage de Motion:



Graphe de collaboration de Motion:



Fonctions membres publiques

- [Motion](#) ()
Constructeur par défaut de la classe [Motion](#).
- virtual void [init](#) ()
Initialisation du capteur de mouvement, appelé par le constructeur.
- virtual int [Get_pin](#) () const
Récupération du pin du capteur de mouvement.
- int [Get_state_sensor](#) ()
Récupération de l'état du capteur de mouvement.
- bool [isDetect](#) ()
Detection d'un mouvement.
- bool [operator==](#) (int state)
Surcharge de l'opérateur ==.
- bool [operator!=](#) (int state)
Surcharge de l'opérateur !=.

Fonctions membres publiques hérités de [Sensor](#)

- virtual void [init](#) ()
Initialisation du capteur.
- virtual int [Get_pin](#) ()
Récupération du pin du capteur.
- virtual int [Get_state_sensor](#) ()
Récupération de l'état du capteur.

4.5.1 Description détaillée

Classe du capteur de mouvement.

Le capteur de mouvement est un capteur qui permet de détecter un mouvement.

Avertissement

Le capteur de mouvement hérite de la classe [Sensor](#).

4.5.2 Documentation des constructeurs et destructeur

4.5.2.1 Motion()

```
Motion::Motion ( )
```

Constructeur par défaut de la classe [Motion](#).

4.5.3 Documentation des fonctions membres

4.5.3.1 Get_pin()

```
int Motion::Get_pin ( ) const [virtual]
```

Récupération du pin du capteur de mouvement.

Renvoie

Le pin du capteur de mouvement.

4.5.3.2 Get_state_sensor()

```
int Motion::Get_state_sensor ( ) [virtual]
```

Récupération de l'état du capteur de mouvement.

Renvoie

L'état actuel du capteur de mouvement.

Ancien état du capteur de mouvement.

Réimplémentée à partir de [Sensor](#).

4.5.3.3 init()

```
void Motion::init ( ) [virtual]
```

Initialisation du capteur de mouvement, appelé par le constructeur.

Réimplémentée à partir de [Sensor](#).

4.5.3.4 isDetect()

```
bool Motion::isDetect ( )
```

Detection d'un mouvement.

Renvoie

Vrai si un mouvement est détecté

Faux si aucun mouvement n'est détecté

4.5.3.5 operator!=()

```
bool Motion::operator!= (
    int state )
```

Surcharge de l'opérateur !=.

4.5.3.6 operator==()

```
bool Motion::operator== (
    int state )
```

Surcharge de l'opérateur ==.

La documentation de cette classe a été générée à partir des fichiers suivants :

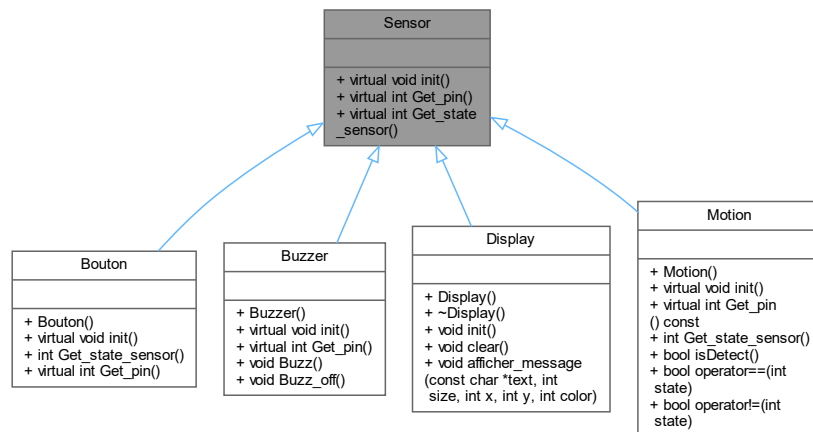
- [main/Motion.hpp](#)
- [main/Motion.cpp](#)

4.6 Référence de la classe Sensor

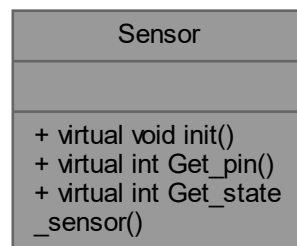
Classe du capteur.

```
#include <Sensor.hpp>
```

Graphe d'héritage de Sensor:



Graphe de collaboration de Sensor:



Fonctions membres publiques

- virtual void `init()`
Initialisation du capteur.
- virtual int `Get_pin()`
Récupération du pin du capteur.
- virtual int `Get_state_sensor()`
Récupération de l'état du capteur.

4.6.1 Description détaillée

Classe du capteur.

4.6.2 Documentation des fonctions membres

4.6.2.1 Get_pin()

```
int Sensor::Get_pin ( ) [virtual]
```

Récupération du pin du capteur.

Renvoie

Le pin du capteur.

Réimplémentée dans [Bouton](#), et [Buzzer](#).

4.6.2.2 Get_state_sensor()

```
int Sensor::Get_state_sensor ( ) [virtual]
```

Récupération de l'état du capteur.

Renvoie

L'état du capteur.

Réimplémentée dans [Bouton](#), et [Motion](#).

4.6.2.3 init()

```
void Sensor::init ( ) [virtual]
```

Initialisation du capteur.

Réimplémentée dans [Bouton](#), [Buzzer](#), [Display](#), et [Motion](#).

La documentation de cette classe a été générée à partir des fichiers suivants :

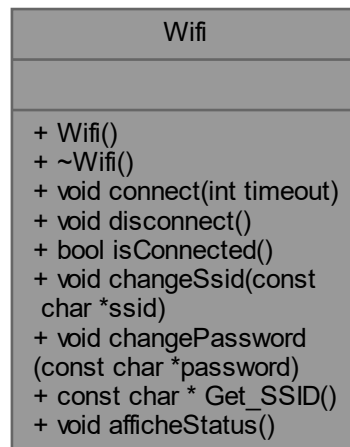
- [main/Sensor.hpp](#)
- [main/Sensor.cpp](#)

4.7 Référence de la classe Wifi

Classe du wifi.

```
#include <Wifi.hpp>
```

Graphe de collaboration de Wifi:



Fonctions membres publiques

- [Wifi](#) ()
Constructeur par défaut de la classe [Wifi](#).
- [~Wifi](#) ()
Destructeur par défaut de la classe [Wifi](#).
- void [connect](#) (int timeout)
Connexion au wifi.
- void [disconnect](#) ()
Déconnexion du wifi.
- bool [isConnected](#) ()
verifie si le wifi est connecté
- void [changeSsid](#) (const char *ssid)
Changer le SSID.
- void [changePassword](#) (const char *password)
Changer le mot de passe.
- const char * [Get_SSID](#) ()
Retourne le SSID.
- void [afficheStatus](#) ()
Retourne l'état du wifi.

4.7.1 Description détaillée

Classe du wifi.

4.7.2 Documentation des constructeurs et destructeur

4.7.2.1 Wifi()

```
Wifi::Wifi ( )
```

Constructeur par défaut de la classe [Wifi](#).

4.7.2.2 ~Wifi()

```
Wifi::~~Wifi ( )
```

Destructeur par défaut de la classe [Wifi](#).

4.7.3 Documentation des fonctions membres

4.7.3.1 afficheStatus()

```
void Wifi::afficheStatus ( )
```

Retourne l'état du wifi.

Renvoie

vrai if connected
faux if not connected

4.7.3.2 changePassword()

```
void Wifi::changePassword (
    const char * password )
```

Changer le mot de passe.

Paramètres

<i>password</i>	
-----------------	--

4.7.3.3 changeSsid()

```
void Wifi::changeSsid (
    const char * ssid )
```

Changer le SSID.

Paramètres

<i>ssid</i>	
-------------	--

4.7.3.4 connect()

```
void Wifi::connect (
    int timeout )
```

Connexion au wifi.

Paramètres

<i>timeout</i>	Durée maximale de tentative de connexion
----------------	--

4.7.3.5 disconnect()

```
void Wifi::disconnect ( )
```

Déconnexion du wifi.

4.7.3.6 Get_SSID()

```
const char * Wifi::Get_SSID ( )
```

Retourne le SSID.

Renvoie

const char*

4.7.3.7 isConnected()

```
bool Wifi::isConnected ( )
```

verifie si le wifi est connecté

Renvoie

true if connected

false if not connected

La documentation de cette classe a été générée à partir des fichiers suivants :

- [main/Wifi.hpp](#)
- [main/Wifi.cpp](#)

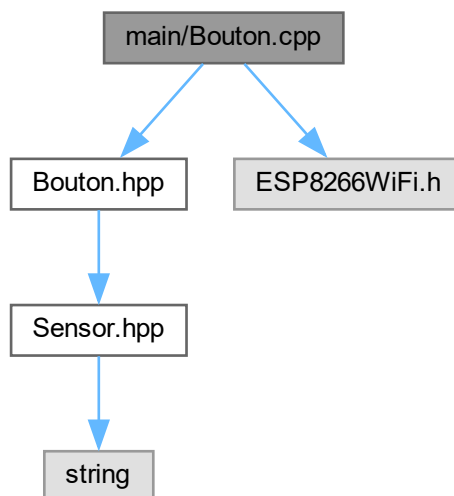
Chapitre 5

Documentation des fichiers

5.1 Référence du fichier main/Bouton.cpp

```
#include "Bouton.hpp"  
#include <ESP8266WiFi.h>
```

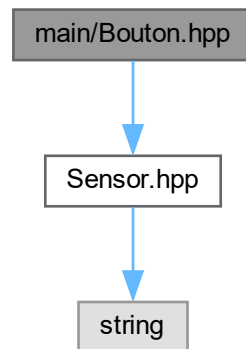
Graphe des dépendances par inclusion de Bouton.cpp:



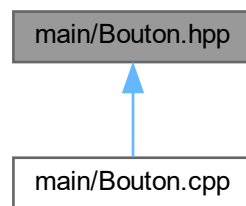
5.2 Référence du fichier main/Bouton.hpp

```
#include "Sensor.hpp"
```

Graphe des dépendances par inclusion de Bouton.hpp:



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Classes

- class [Bouton](#)
Classe du bouton.

5.3 Bouton.hpp

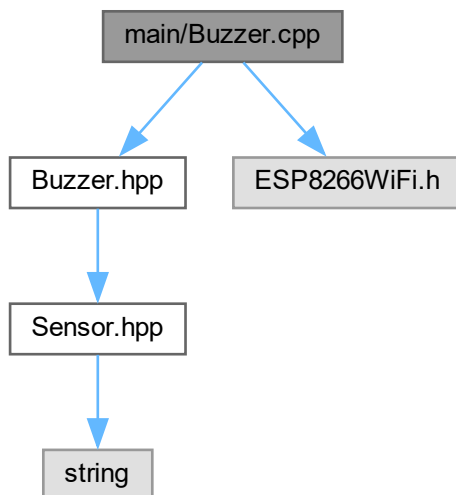
[Aller à la documentation de ce fichier.](#)

```
00001 #ifndef DEF_BOUTON
00002 #define DEF_BOUTON
00003 #include "Sensor.hpp"
00007 class Bouton : public Sensor
00008 {
00009     private:
00010         int state_bouton;
00011         int pinBouton;
00012     public:
00016         Bouton();
00017         //~Bouton();
00021         virtual void init();
00026         int Get_state_sensor();
00031         virtual int Get_pin();
00032 };
00033 #endif
```

5.4 Référence du fichier main/Buzzer.cpp

```
#include "Buzzer.hpp"  
#include <ESP8266WiFi.h>
```

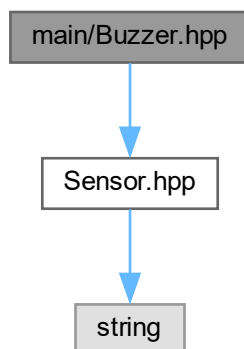
Graphe des dépendances par inclusion de Buzzer.cpp:



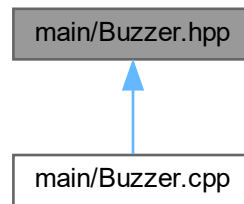
5.5 Référence du fichier main/Buzzer.hpp

```
#include "Sensor.hpp"
```

Graphe des dépendances par inclusion de Buzzer.hpp:



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Classes

— class `Buzzer`
Classe du buzzer.

5.6 Buzzer.hpp

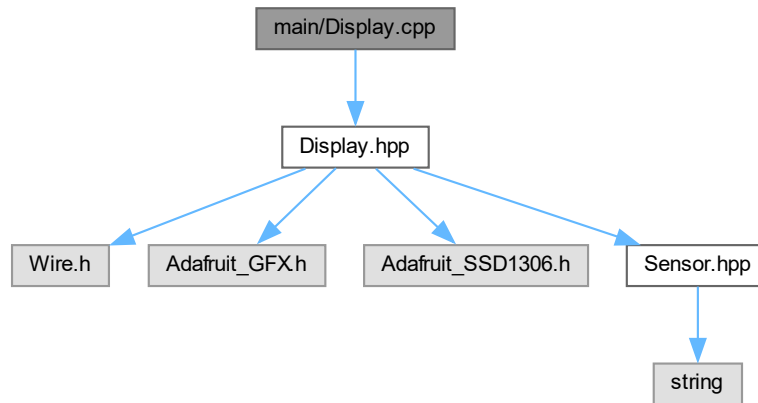
[Aller à la documentation de ce fichier.](#)

```
00001 #ifndef DEF_BUZZER
00002 #define DEF_BUZZER
00003 #include "Sensor.hpp"
00009 class Buzzer :public Sensor
00010 {
00011     private:
00012         int pinBuzzer;
00013     public:
00017         Buzzer();
00018         // ~Buzzer();
00022         virtual void init();
00027         virtual int Get_pin();
00031         void Buzz();
00032         void Buzz_off();
00033 };
00034 #endif
```

5.7 Référence du fichier main/Display.cpp

```
#include "Display.hpp"
```

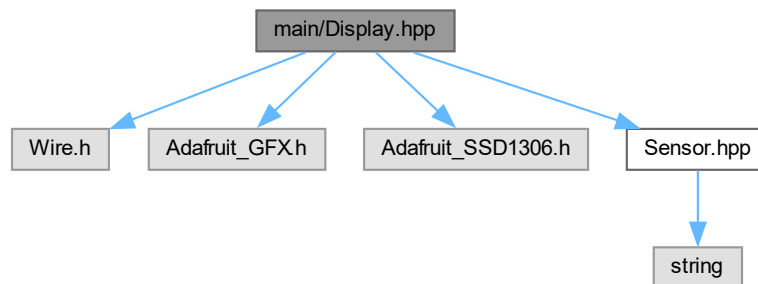
Graphe des dépendances par inclusion de Display.cpp:



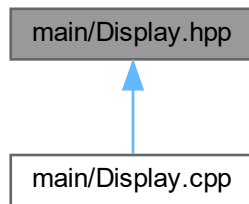
5.8 Référence du fichier main/Display.hpp

```
#include <Wire.h>  
#include <Adafruit_GFX.h>  
#include <Adafruit_SSD1306.h>  
#include "Sensor.hpp"
```

Graphe des dépendances par inclusion de Display.hpp:



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Classes

— class `Display`
Classe de l'écran.

5.9 Display.hpp

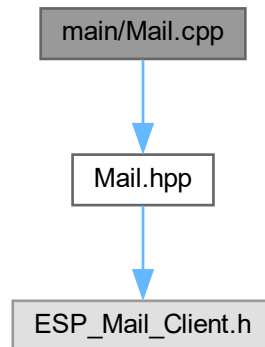
[Aller à la documentation de ce fichier.](#)

```
00001 #ifndef DEF_DISPLAY
00002 #define DEF_DISPLAY
00003 #include <Wire.h>
00004 #include <Adafruit_GFX.h>
00005 #include <Adafruit_SSD1306.h>
00006 #include "Sensor.hpp"
00012 class Display : public Sensor
00013 {
00014     private:
00015         int pinDisplay;
00016         Adafruit_SSD1306* display;
00017         const int SCREEN_WIDTH = 128;
00018         const int SCREEN_HEIGHT = 64;
00019         const int OLED_RESET = -1;
00020     public:
00024         Display();
00028         ~Display();
00032         void init();
00036         void clear();
00040         void afficher_message(const char* text, int size, int x, int y, int color);
00041         // void get_display();
00042 };
00043 #endif
```


5.10 Référence du fichier main/Mail.cpp

```
#include "Mail.hpp"
```

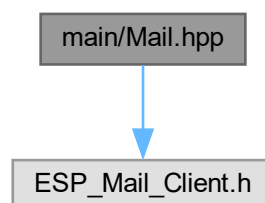
Graphe des dépendances par inclusion de Mail.cpp:



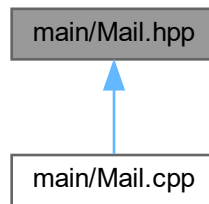
5.11 Référence du fichier main/Mail.hpp

```
#include <ESP_Mail_Client.h>
```

Graphe des dépendances par inclusion de Mail.hpp:



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Classes

- class [Mail](#)
Classe du mail.

Macros

- #define [MAX_MAIL_ADRESS_LENGTH](#) 65
- #define [MAX_MAIL_PASSWORD_LENGTH](#) 65
- #define [MAX_MAIL_SERVER_LENGTH](#) 65
- #define [MAX_MAIL_PORT_LENGTH](#) 65
- #define [DEFAULT_MAIL_SENDER_ADDRESS](#) "Alarme@caramail.fr"
- #define [DEFAULT_MAIL_SENDER_PASSWORD](#) "v5pGdtRVxrS6gEU"
- #define [DEFAULT_MAIL_SERVER](#) "mail.gmx.com"
- #define [DEFAULT_MAIL_PORT](#) 587
- #define [DEFAULT_MAIL_SUBJECT](#) "Alarme déclenchée à l'instant !"
- #define [DEFAULT_MAIL_RECIPIENT](#) "veychenn@etud.insa-toulouse.fr"

5.11.1 Documentation des macros

5.11.1.1 DEFAULT_MAIL_PORT

```
#define DEFAULT_MAIL_PORT 587
```

5.11.1.2 DEFAULT_MAIL_RECIPIENT

```
#define DEFAULT_MAIL_RECIPIENT "veychenn@etud.insa-toulouse.fr"
```

5.11.1.3 DEFAULT_MAIL_SENDER_ADRESS

```
#define DEFAULT_MAIL_SENDER_ADRESS "Alarme@caramail.fr"
```

5.11.1.4 DEFAULT_MAIL_SENDER_PASSWORD

```
#define DEFAULT_MAIL_SENDER_PASSWORD "v5pGdtRVxrS6gEU"
```

5.11.1.5 DEFAULT_MAIL_SERVER

```
#define DEFAULT_MAIL_SERVER "mail.gmx.com"
```

5.11.1.6 DEFAULT_MAIL_SUBJECT

```
#define DEFAULT_MAIL_SUBJECT "Alarme déclenchée à l'instant ! "
```

5.11.1.7 MAX_MAIL_ADRESS_LENGTH

```
#define MAX_MAIL_ADRESS_LENGTH 65
```

5.11.1.8 MAX_MAIL_PASSWORD_LENGTH

```
#define MAX_MAIL_PASSWORD_LENGTH 65
```

5.11.1.9 MAX_MAIL_PORT_LENGTH

```
#define MAX_MAIL_PORT_LENGTH 65
```

5.11.1.10 MAX_MAIL_SERVER_LENGTH

```
#define MAX_MAIL_SERVER_LENGTH 65
```

5.12 Mail.hpp

[Aller à la documentation de ce fichier.](#)

```

00001 #include <ESP_Mail_Client.h>
00002 #ifndef DEF_MAIL
00003 #define DEF_MAIL
00004 #define MAX_MAIL_ADRESS_LENGTH 65
00005 #define MAX_MAIL_PASSWORD_LENGTH 65
00006 #define MAX_MAIL_SERVER_LENGTH 65
00007 #define MAX_MAIL_PORT_LENGTH 65
00008 // #define DEFAULT_MESSAGE "Alarme déclenchée ! Jerem est en danger !"
00009 #define DEFAULT_MAIL_SENDER_ADRESS "Alarme@caramail.fr"
00010 #define DEFAULT_MAIL_SENDER_PASSWORD "v5pGdtRVxrS6gEU"
00011 #define DEFAULT_MAIL_SERVER "mail.gmx.com"
00012 #define DEFAULT_MAIL_PORT 587
00013 #define DEFAULT_MAIL_SUBJECT "Alarme déclenchée à l'instant !"
00014 #define DEFAULT_MAIL_RECIPIENT "veychenn@etud.insa-toulouse.fr"
00018 class Mail
00019 {
00020 public:
00024     Mail();
00028     ~Mail();
00032     void sendMail(void);
00033 private:
00035     SMTPSession smtp;
00036     ESP_Mail_Session session;
00037     ESP_Mail_Client mailClient;
00038     ESP_Mail_Message message;
00039 };
00040 #endif

```

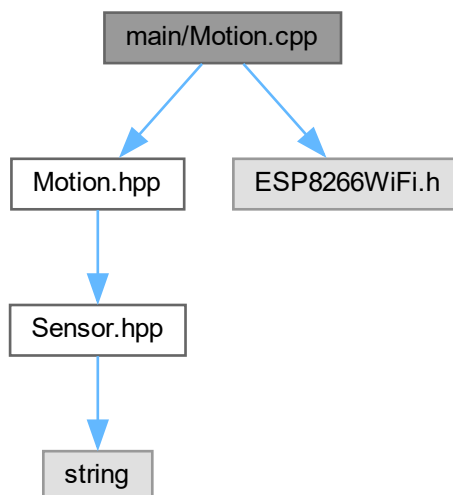
5.13 Référence du fichier main/Motion.cpp

```

#include "Motion.hpp"
#include <ESP8266WiFi.h>

```

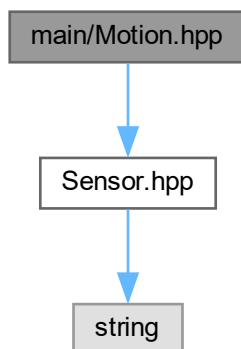
Graphe des dépendances par inclusion de Motion.cpp:



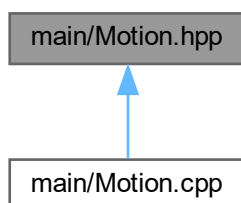
5.14 Référence du fichier main/Motion.hpp

```
#include "Sensor.hpp"
```

Graphe des dépendances par inclusion de Motion.hpp:



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Classes

— class [Motion](#)

Classe du capteur de mouvement.

5.15 Motion.hpp

[Aller à la documentation de ce fichier.](#)

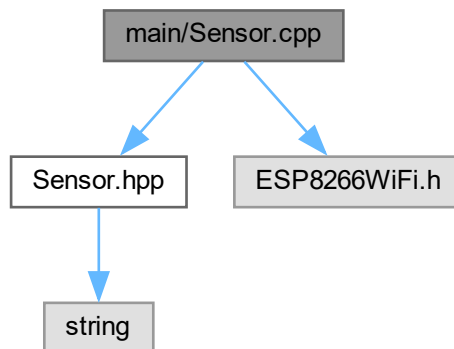
```
00001 #ifndef DEF_MOTION
00002 #define DEF_MOTION
00003 #include "Sensor.hpp"
00009 class Motion : public Sensor
00010 {
```

```
00011 private:
00012     int pinMotion;
00013     int current_state;
00014     int old_state;
00015
00016 public:
00020     Motion();
00021     // ~Motion();
00025     virtual void init();
00030     virtual int Get_pin() const;
00036     int Get_state_sensor();
00042     bool isDetect();
00047     bool operator==(int state);
00052     bool operator!=(int state);
00053 };
00054
00055 #endif
```

5.16 Référence du fichier main/Sensor.cpp

```
#include "Sensor.hpp"
#include <ESP8266WiFi.h>
```

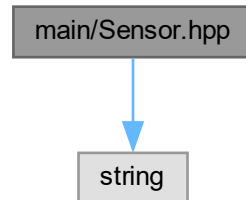
Graphe des dépendances par inclusion de Sensor.cpp:



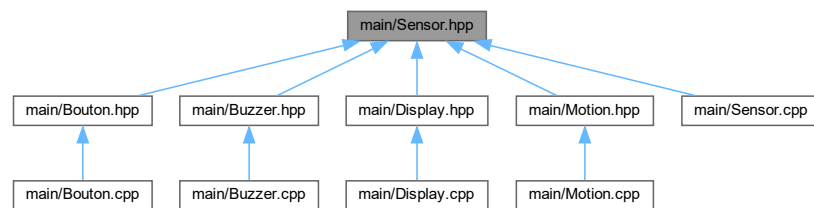
5.17 Référence du fichier main/Sensor.hpp

```
#include <string>
```

Graphe des dépendances par inclusion de Sensor.hpp:



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Classes

— class `Sensor`
Classe du capteur.

5.18 Sensor.hpp

[Aller à la documentation de ce fichier.](#)

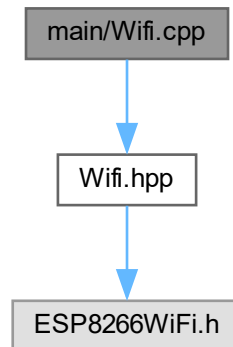
```

00001 #ifndef DEF_SENSOR
00002 #define DEF_SENSOR
00003
00004 #include <string>
00008 class Sensor
00009 {
00010     private:
00011         int pinSensor;
00012     public:
00013         // Sensor();
00014         // ~Sensor();
00018         virtual void init();
00023         virtual int Get_pin();
00028         virtual int Get_state_sensor();
00029 };
00030 #endif
  
```

5.19 Référence du fichier main/Wifi.cpp

```
#include "Wifi.hpp"
```

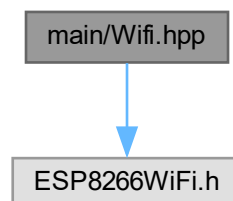
Graphe des dépendances par inclusion de Wifi.cpp:



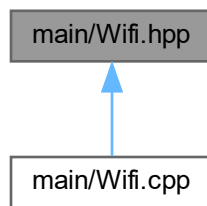
5.20 Référence du fichier main/Wifi.hpp

```
#include <ESP8266WiFi.h>
```

Graphe des dépendances par inclusion de Wifi.hpp:



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Classes

- class [Wifi](#)
Classe du wifi.

Macros

- #define [MAX_SSID LENGHT](#) 33
- #define [MAX_WPA2_PWD LENGHT](#) 65

5.20.1 Documentation des macros

5.20.1.1 MAX_SSID_LENGHT

```
#define MAX_SSID_LENGHT 33
```

5.20.1.2 MAX_WPA2_PWD_LENGHT

```
#define MAX_WPA2_PWD_LENGHT 65
```

5.21 Wifi.hpp

[Aller à la documentation de ce fichier.](#)

```
00001 #include <ESP8266WiFi.h>
00002 #define MAX_SSID_LENGTH 33
00003 #define MAX_WPA2_PWD_LENGTH 65
00004 #ifndef WIFI_HPP
00005 #define WIFI_HPP
00010 class Wifi
00011 {
00012 public:
00017     Wifi();
00022     ~Wifi();
00027     void connect(int timeout);
00032     void disconnect();
00038     bool isConnected();
00044     void changeSsid(const char *ssid);
00050     void changePassword(const char *password);
00056     const char* Get_SSID();
00063     void afficheStatus();
00064
00065 private:
00066     const char *ssid = new char[MAX_SSID_LENGTH];
00067     const char *password = new char[MAX_WPA2_PWD_LENGTH];
00068 };
00069 #endif
```

Index

- ~Display
 - Display, [15](#)
- ~Mail
 - Mail, [17](#)
- ~Wifi
 - Wifi, [25](#)
- afficher_message
 - Display, [15](#)
- afficheStatus
 - Wifi, [25](#)
- Bouton, [7](#)
 - Bouton, [8](#)
 - Get_pin, [9](#)
 - Get_state_sensor, [9](#)
 - init, [9](#)
- Buzz
 - Buzzer, [12](#)
- Buzz_off
 - Buzzer, [12](#)
- Buzzer, [10](#)
 - Buzz, [12](#)
 - Buzz_off, [12](#)
 - Buzzer, [12](#)
 - Get_pin, [12](#)
 - init, [12](#)
- changePassword
 - Wifi, [25](#)
- changeSsid
 - Wifi, [26](#)
- clear
 - Display, [15](#)
- connect
 - Wifi, [26](#)
- DEFAULT_MAIL_PORT
 - Mail.hpp, [36](#)
- DEFAULT_MAIL_RECIPIENT
 - Mail.hpp, [36](#)
- DEFAULT_MAIL_SENDER_ADRESS
 - Mail.hpp, [36](#)
- DEFAULT_MAIL_SENDER_PASSWORD
 - Mail.hpp, [37](#)
- DEFAULT_MAIL_SERVER
 - Mail.hpp, [37](#)
- DEFAULT_MAIL_SUBJECT
 - Mail.hpp, [37](#)
- disconnect
 - Wifi, [26](#)
- Display, [13](#)
 - ~Display, [15](#)
 - afficher_message, [15](#)
 - clear, [15](#)
 - Display, [15](#)
 - init, [16](#)
- Get_pin
 - Bouton, [9](#)
 - Buzzer, [12](#)
 - Motion, [20](#)
 - Sensor, [23](#)
- Get_SSID
 - Wifi, [26](#)
- Get_state_sensor
 - Bouton, [9](#)
 - Motion, [20](#)
 - Sensor, [23](#)
- init
 - Bouton, [9](#)
 - Buzzer, [12](#)
 - Display, [16](#)
 - Motion, [20](#)
 - Sensor, [23](#)
- isConnected
 - Wifi, [26](#)
- isDetect
 - Motion, [21](#)
- Mail, [16](#)
 - ~Mail, [17](#)
 - Mail, [17](#)
 - sendMail, [17](#)
- Mail.hpp
 - DEFAULT_MAIL_PORT, [36](#)
 - DEFAULT_MAIL_RECIPIENT, [36](#)
 - DEFAULT_MAIL_SENDER_ADRESS, [36](#)
 - DEFAULT_MAIL_SENDER_PASSWORD, [37](#)
 - DEFAULT_MAIL_SERVER, [37](#)
 - DEFAULT_MAIL_SUBJECT, [37](#)
 - MAX_MAIL_ADRESS_LENGTH, [37](#)
 - MAX_MAIL_PASSWORD_LENGTH, [37](#)
 - MAX_MAIL_PORT_LENGTH, [37](#)
 - MAX_MAIL_SERVER_LENGTH, [37](#)
- main/Bouton.cpp, [29](#)
- main/Bouton.hpp, [29](#), [30](#)
- main/Buzzer.cpp, [31](#)
- main/Buzzer.hpp, [31](#), [32](#)

- main/Display.cpp, [33](#)
- main/Display.hpp, [33](#), [34](#)
- main/Mail.cpp, [35](#)
- main/Mail.hpp, [35](#), [38](#)
- main/Motion.cpp, [38](#)
- main/Motion.hpp, [39](#)
- main/Sensor.cpp, [40](#)
- main/Sensor.hpp, [41](#)
- main/Wifi.cpp, [42](#)
- main/Wifi.hpp, [42](#), [44](#)
- MAX_MAIL_ADRESS LENGHT
 - Mail.hpp, [37](#)
- MAX_MAIL_PASSWORD LENGHT
 - Mail.hpp, [37](#)
- MAX_MAIL_PORT LENGHT
 - Mail.hpp, [37](#)
- MAX_MAIL_SERVER LENGHT
 - Mail.hpp, [37](#)
- MAX_SSID LENGHT
 - Wifi.hpp, [43](#)
- MAX_WPA2_PWD LENGHT
 - Wifi.hpp, [43](#)
- Motion, [18](#)
 - Get_pin, [20](#)
 - Get_state_sensor, [20](#)
 - init, [20](#)
 - isDetect, [21](#)
 - Motion, [20](#)
 - operator!=, [21](#)
 - operator==, [21](#)
- operator!=
 - Motion, [21](#)
- operator==
 - Motion, [21](#)
- sendMail
 - Mail, [17](#)
- Sensor, [22](#)
 - Get_pin, [23](#)
 - Get_state_sensor, [23](#)
 - init, [23](#)
- Wifi, [24](#)
 - ~Wifi, [25](#)
 - afficheStatus, [25](#)
 - changePassword, [25](#)
 - changeSsid, [26](#)
 - connect, [26](#)
 - disconnect, [26](#)
 - Get_SSID, [26](#)
 - isConnected, [26](#)
 - Wifi, [25](#)
- Wifi.hpp
 - MAX_SSID LENGHT, [43](#)
 - MAX_WPA2_PWD LENGHT, [43](#)