Alarme connectée

Generated by Doxygen 1.9.6

1 Hierarchical Index 1

1.1 Class Hierarchy . . . 1

2 Class Index 3

2.1 Class List . . . 3

3 File Index 5

3.1 File List . . . 5

4 Class Documentation 7

4.1 Bouton Class Reference . . . 7

4.1.1 Constructor & Destructor Documentation . . . 8

4.1.1.1 Bouton() . . . 8

4.1.2 Member Function Documentation . . . 8

4.1.2.1 Get_pin() . . . 8

4.1.2.2 Get_state_sensor() . . . 9

4.1.2.3 init() . . . 9

4.2 Buzzer Class Reference . . . 9

4.2.1 Constructor & Destructor Documentation . . . 10

4.2.1.1 Buzzer() . . . 10

4.2.2 Member Function Documentation . . . 10

4.2.2.1 Buzz() . . . 10

4.2.2.2 Get_pin() . . . 11

4.2.2.3 init() . . . 11

4.3 Mail Class Reference . . . 11

4.3.1 Constructor & Destructor Documentation . . . 11

4.3.1.1 Mail() . . . 11

4.3.1.2 ~Mail() . . . 12

4.3.2 Member Function Documentation . . . 12

4.3.2.1 sendMail() . . . 12

4.4 Motion Class Reference . . . 12

4.4.1 Constructor & Destructor Documentation . . . 13

4.4.1.1 Motion() . . . 13

4.4.2 Member Function Documentation . . . 13

4.4.2.1 Get_pin() . . . 13

4.4.2.2 Get_state_sensor() . . . 14

4.4.2.3 init() . . . 14

4.4.2.4 isDetect() . . . 14

4.5 Sensor Class Reference . . . 15

4.5.1 Member Function Documentation . . . 15

4.5.1.1 Get_pin() . . . 15

4.5.1.2 Get_state_sensor() . . . 16

4.5.1.3 init() . . . 16

4.6 Wifi Class Reference 16
4.6.1 Detailed Description 17
4.6.2 Constructor & Destructor Documentation 17
4.6.2.1 Wifi() 17
4.6.2.2 ∼Wifi() 17
4.6.3 Member Function Documentation 17
4.6.3.1 afficheStatus() 17
4.6.3.2 changePassword() 17
4.6.3.3 changeSsid() 18
4.6.3.4 connect() 18
4.6.3.5 disconnect() 18
4.6.3.6 Get_SSID() 18
4.6.3.7 isConnected() 19

5 File Documentation **21**

5.1 Bouton.hpp 21
5.2 Buzzer.hpp 21
5.3 Mail.hpp 21
5.4 Motion.hpp 22
5.5 Sensor.hpp 22
5.6 Wifi.hpp 22

Index **25**

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Mail 11
Sensor 15
  Bouton 7
  Buzzer 9
  Motion 12
Wifi 16

Chapter 2

Class Index

2.1 Class List

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

Bouton 7
Buzzer 9
Mail 11
Motion 12
Sensor 15
Wifi
    Class to manage the wifi 16

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

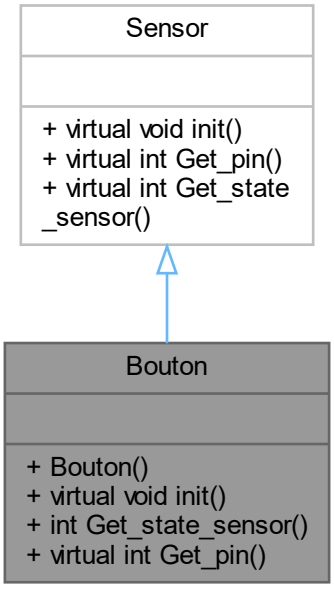
main/Bouton.hpp 21
main/Buzzer.hpp 21
main/Mail.hpp 21
main/Motion.hpp 22
main/Sensor.hpp 22
main/Wifi.hpp 22

Chapter 4

Class Documentation

4.1 Bouton Class Reference

Inheritance diagram for Bouton:



Public Member Functions

- Bouton ()

  Construct a new Bouton:: Bouton object.
- virtual void init ()

  Destroy the Bouton:: Bouton object.

- int Get_state_sensor ()

  Get the state sensor object.
- virtual int Get_pin ()

  Get the pin object.

- virtual void init ()

  Construct a new Sensor:: Sensor object.
- virtual int Get_pin ()

  Get the Sensor pin number.
- virtual int Get_state_sensor ()

  Get the state of the sensor.

4.1.1 Constructor & Destructor Documentation

4.1.1.1 Bouton()

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

Construct a new Bouton:: Bouton object.

4.1.2 Member Function Documentation

4.1.2.1 Get_pin()

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

Get the pin object.

Returns

Button pin number

Reimplemented from Sensor.

4.1.2.2 Get_state_sensor()

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

Get the state sensor object.

Returns

1 if button is pressed

0 if button is not pressed

Reimplemented from Sensor.

4.1.2.3 init()

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

Destroy the Bouton:: Bouton object.

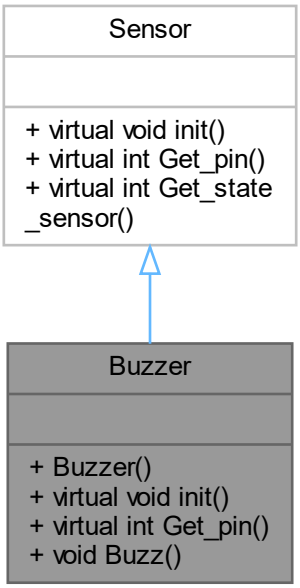
Reimplemented from Sensor.

The documentation for this class was generated from the following files:

- main/Bouton.hpp
- main/Bouton.cpp

4.2 Buzzer Class Reference

Inheritance diagram for Buzzer:



Public Member Functions

- Buzzer ()

  Construct a new Buzzer:: Buzzer object.
- virtual void init ()

  init the Buzzer
- virtual int Get_pin ()

  Get the pin object.
- void Buzz ()

  Emit a buzz during 800ms.

Public Member Functions inherited from Sensor

- virtual void init ()

  Construct a new Sensor:: Sensor object.
- virtual int Get_pin ()

  Get the Sensor pin number.
- virtual int Get_state_sensor ()

  Get the state of the sensor.

4.2.1 Constructor & Destructor Documentation

4.2.1.1 Buzzer()

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

Construct a new Buzzer:: Buzzer object.

4.2.2 Member Function Documentation

4.2.2.1 Buzz()

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

Emit a buzz during 800ms.

4.2.2.2 Get_pin()

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

Get the pin object.

Returns

Buzzer pin number

Reimplemented from Sensor.

4.2.2.3 init()

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

init the Buzzer

Reimplemented from Sensor.

The documentation for this class was generated from the following files:

- main/Buzzer.hpp
- main/Buzzer.cpp

4.3 Mail Class Reference

Public Member Functions

- Mail ()
  Construct a new Mail:: Mail object.
- ~Mail ()
  Destroy the Mail:: Mail object.
- void sendMail (void)
  Send a mail.

4.3.1 Constructor & Destructor Documentation

4.3.1.1 Mail()

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

Construct a new Mail:: Mail object.

4.3.1.2 ~Mail()

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

Destroy the Mail:: Mail object.

4.3.2 Member Function Documentation

4.3.2.1 sendMail()

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

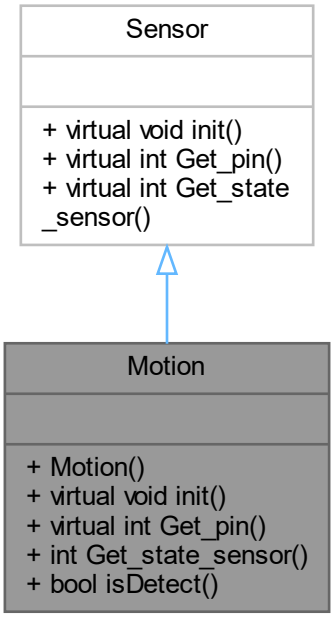
Send a mail.

The documentation for this class was generated from the following files:

- main/Mail.hpp
- main/Mail.cpp

4.4 Motion Class Reference

Inheritance diagram for Motion:



Public Member Functions

- Motion ()

  Construct a new Motion:: Motion object.
- virtual void init ()

  init the Motion sensor, only called by the constructor
- virtual int Get_pin ()

  Get the Motion pin number.
- int Get_state_sensor ()

  Get the state sensor.
- bool isDetect ()

  Check if motion is detected.

- virtual void init ()

  Construct a new Sensor:: Sensor object.
- virtual int Get_pin ()

  Get the Sensor pin number.
- virtual int Get_state_sensor ()

  Get the state of the sensor.

4.4.1 Constructor & Destructor Documentation

4.4.1.1 Motion()

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

Construct a new Motion:: Motion object.

4.4.2 Member Function Documentation

4.4.2.1 Get_pin()

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

Get the Motion pin number.

Returns

Motion pin number

Reimplemented from Sensor.

4.4.2.2 Get_state_sensor()

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

Get the state sensor.

Returns

1 if motion is detected

0 if motion is not detected

Reimplemented from Sensor.

4.4.2.3 init()

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

init the Motion sensor, only called by the constructor

Reimplemented from Sensor.

4.4.2.4 isDetect()

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

Check if motion is detected.

Returns

true if motion is detected

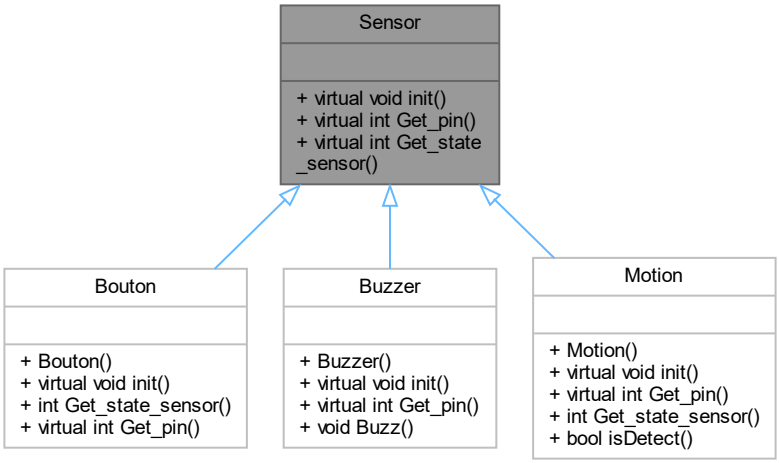
false if motion is not detected

The documentation for this class was generated from the following files:

- main/Motion.hpp
- main/Motion.cpp

4.5 Sensor Class Reference

Inheritance diagram for Sensor:



Public Member Functions

- virtual void init ()

  Construct a new Sensor:: Sensor object.
- virtual int Get_pin ()

  Get the Sensor pin number.
- virtual int Get_state_sensor ()

  Get the state of the sensor.

4.5.1 Member Function Documentation

4.5.1.1 Get_pin()

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

Get the Sensor pin number.

Returns

Sensor pin number

Reimplemented in Bouton, Buzzer, and Motion.

4.5.1.2 Get_state_sensor()

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

Get the state of the sensor.

Reimplemented in Bouton, and Motion.

4.5.1.3 init()

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

Construct a new Sensor:: Sensor object.

Reimplemented in Bouton, Buzzer, and Motion.

The documentation for this class was generated from the following files:

- main/Sensor.hpp
- main/Sensor.cpp

4.6 Wifi Class Reference

Class to manage the wifi.

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

Public Member Functions

- Wifi ()
  Construct a new Wifi object.
- ~Wifi ()
  Destroy the Wifi object.
- void connect (int timeout)
  Connect to the wifi.
- void disconnect ()
  Disconnect from the wifi.
- bool isConnected ()
  Check if the wifi is connected.
- void changeSsid (const char *ssid)
  Change the ssid.
- void changePassword (const char *password)
  Change the password.
- const char * Get_SSID ()
  Get the SSID.
- void afficheStatus ()
  Get the status of the wifi.

4.6.1 Detailed Description

Class to manage the wifi.

4.6.2 Constructor & Destructor Documentation

4.6.2.1 Wifi()

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

Construct a new Wifi object.

4.6.2.2 ~Wifi()

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

Destroy the Wifi object.

4.6.3 Member Function Documentation

4.6.3.1 afficheStatus()

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

Get the status of the wifi.

Returns

true if connected

false if not connected

4.6.3.2 changePassword()

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

Change the password.

Parameters

password	

4.6.3.3 changeSsid()

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

Change the ssid.

Parameters

ssid	

4.6.3.4 connect()

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

Connect to the wifi.

Parameters

timeout	

4.6.3.5 disconnect()

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

Disconnect from the wifi.

4.6.3.6 Get_SSID()

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

Get the SSID.

Returns

const char*

4.6.3.7 isConnected()

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

Check if the wifi is connected.

Returns

true if connected

false if not connected

The documentation for this class was generated from the following files:

- main/Wifi.hpp
- main/Wifi.cpp

Chapter 5

File Documentation

5.1 Bouton.hpp

```
00001 #ifndef DEF_BOUTON
00002 #define DEF_BOUTON
00003 #include "Sensor.hpp"
00004 class Bouton : public Sensor
00005 {
00006     private:
00007     int state_bouton;
00008     int pinBouton;
00009     public:
00010     Bouton();
00011     //~Bouton();
00012     virtual void init();
00013     int Get_state_sensor();
00014     virtual int Get_pin();
00015 };
00016 #endif
```

5.2 Buzzer.hpp

```
00001 #ifndef DEF_BUZZER
00002 #define DEF_BUZZER
00003 #include "Sensor.hpp"
00004
00005 class Buzzer :public  Sensor
00006 {
00007     private:
00008     int pinBuzzer;
00009     public:
00010     Buzzer();
00011    // ~Buzzer();
00012     virtual void init();
00013     virtual int Get_pin();
00014     void Buzz();
00015 };
00016 #endif
```

5.3 Mail.hpp

```
00001 #include <ESP_Mail_Client.h>
00002 #ifndef DEF_MAIL
00003 #define DEF_MAIL
00004 #define MAX_MAIL_ADRESS_LENGHT 65
00005 #define MAX_MAIL_PASSWORD_LENGHT 65
00006 #define MAX_MAIL_SERVER_LENGHT 65
00007 #define MAX_MAIL_PORT_LENGHT 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"
00015 class Mail
00016 {
00017 public:
00018     Mail();
00019     ~Mail();
00020     void sendMail(void);
00021
00022 private:
00023     SMTPSession smtp;
00024     ESP_Mail_Session session;
00025     ESP_Mail_Client mailClient;
00026     ESP_Mail_Message message;
00027     };
00028 #endif
```

5.4 Motion.hpp

```
00001 #ifndef DEF_MOTION
00002 #define DEF_MOTION
00003 #include "Sensor.hpp"
00004
00005 class Motion: public Sensor
00006 {
00007     private:
00008     int pinMotion;
00009     int current_state;
00010     int old_state;
00011     public:
00012     Motion();
00013    // ~Motion();
00014     virtual void init();
00015     virtual int Get_pin();
00016     int Get_state_sensor();
00017     bool isDetect();
00018 };
00019
00020 #endif
```

5.5 Sensor.hpp

```
00001 #ifndef DEF_SENSOR
00002 #define DEF_SENSOR
00003
00004 #include <string>
00005 class Sensor
00006 {
00007     private:
00008     int pinSensor;
00009     public:
00010     // Sensor();
00011     // ~Sensor();
00012     virtual void init();
00013     virtual int Get_pin();
00014     virtual int Get_state_sensor();
00015 };
00016 #endif
```

5.6 Wifi.hpp

```
00001 #include <ESP8266WiFi.h>
00002 #define MAX_SSID_LENGHT 33
00003 #define MAX_WPA2_PWD_LENGHT 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:
00070     const char *ssid = new char[MAX_SSID_LENGHT];
00075     const char *password = new char[MAX_WPA2_PWD_LENGHT];
00076 };
00077 #endif
```

Index

~Mail
    Mail, 11
~Wifi
    Wifi, 17

afficheStatus
    Wifi, 17

Bouton, 7
    Bouton, 8
    Get_pin, 8
    Get_state_sensor, 8
    init, 9
Buzz
    Buzzer, 10
Buzzer, 9
    Buzz, 10
    Buzzer, 10
    Get_pin, 10
    init, 11

changePassword
    Wifi, 17
changeSsid
    Wifi, 18
connect
    Wifi, 18

disconnect
    Wifi, 18

Get_pin
    Bouton, 8
    Buzzer, 10
    Motion, 13
    Sensor, 15
Get_SSID
    Wifi, 18
Get_state_sensor
    Bouton, 8
    Motion, 13
    Sensor, 15

init
    Bouton, 9
    Buzzer, 11
    Motion, 14
    Sensor, 16
isConnected
    Wifi, 18
isDetect
    Motion, 14

Mail, 11
    ~Mail, 11
    Mail, 11
    sendMail, 12
main/Bouton.hpp, 21
main/Buzzer.hpp, 21
main/Mail.hpp, 21
main/Motion.hpp, 22
main/Sensor.hpp, 22
main/Wifi.hpp, 22
Motion, 12
    Get_pin, 13
    Get_state_sensor, 13
    init, 14
    isDetect, 14
    Motion, 13

sendMail
    Mail, 12
Sensor, 15
    Get_pin, 15
    Get_state_sensor, 15
    init, 16

Wifi, 16
    ~Wifi, 17
    afficheStatus, 17
    changePassword, 17
    changeSsid, 18
    connect, 18
    disconnect, 18
    Get_SSID, 18
    isConnected, 18
    Wifi, 17