My Project

Generated by Doxygen 1.8.14

Contents

1 Namespace Index **1**

1.1 Namespace List 1

2 Hierarchical Index **3**

2.1 Class Hierarchy 3

3 Class Index **5**

3.1 Class List 5

4 Namespace Documentation **7**

4.1 Smart Namespace Reference 7

4.1.1 Detailed Description 7

5 Class Documentation **9**

5.1 Python_Client.CreateButton Class Reference 9

5.1.1 Detailed Description 9

5.1.2 Constructor & Destructor Documentation 9

5.1.2.1 __init__() 10

5.2 Python_Client.CreateLabel Class Reference 11

5.2.1 Detailed Description 11

5.2.2 Constructor & Destructor Documentation 11

5.2.2.1 __init__() 11

5.3 Python_Client.FrameButtons Class Reference 12

5.3.1 Detailed Description 12

5.3.2 Constructor & Destructor Documentation 13

5.3.2.1 __init__() . . . 13
5.3.3 Member Function Documentation . . . 13
5.3.3.1 create_button() . . . 13
5.3.3.2 frame_switch() . . . 13
5.4 Python_Client.sendMail Class Reference . . . 14
5.4.1 Detailed Description . . . 14
5.4.2 Constructor & Destructor Documentation . . . 14
5.4.2.1 __init__() . . . 14
5.4.3 Member Function Documentation . . . 15
5.4.3.1 sendmail() . . . 15
5.5 Python_Client.WifiConnect Class Reference . . . 15
5.5.1 Detailed Description . . . 16
5.5.2 Constructor & Destructor Documentation . . . 16
5.5.2.1 __init__() . . . 16
5.5.3 Member Function Documentation . . . 16
5.5.3.1 close() . . . 17
5.5.3.2 periodic_socket_check() . . . 17
5.5.3.3 send_msg() . . . 17
5.6 Python_Client.WindowFunction Class Reference . . . 17
5.6.1 Detailed Description . . . 19
5.6.2 Constructor & Destructor Documentation . . . 19
5.6.2.1 __init__() . . . 19
5.6.3 Member Function Documentation . . . 20
5.6.3.1 alarm_handler() . . . 20
5.6.3.2 choose_color() . . . 20
5.6.3.3 forget_frames() . . . 20
5.6.3.4 lights_off() . . . 21
5.6.3.5 on_receive() . . . 21
5.6.3.6 setup_alarmframe() . . . 21
5.6.3.7 setup_framebtn() . . . 21
5.6.3.8 setup_frames() . . . 22
5.6.3.9 setup_rgbframe() . . . 22
5.6.3.10 setup_welcomeframe() . . . 23
5.6.3.11 setup_window() . . . 23
5.6.3.12 window_close() . . . 23
5.6.4 Member Data Documentation . . . 23
5.6.4.1 rgb . . . 23

Index **25**

Chapter 1

Namespace Index

1.1 Namespace List

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

Smart
Home-ino This is the dokumentation of the Source Code from the project Smart Home 7

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

object
- Python_Client.CreateButton 9
- Python_Client.CreateLabel 11
- Python_Client.FrameButtons 12
- Python_Client.sendMail 14
- Python_Client.WifiConnect 15
- Python_Client.WindowFunction 17

Chapter 3

Class Index

3.1 Class List

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

Python_Client.CreateButton
: This class creates default buttons 9
Python_Client.CreateLabel
: This class creates labels 11
Python_Client.FrameButtons
: This class handles the frame switch buttons 12
Python_Client.sendMail
: Class to send emails 14
Python_Client.WifiConnect
: Class for the Wifi connection 15
Python_Client.WindowFunction
: The class to setup the main tkinter window 17

Chapter 4

Namespace Documentation

4.1 Smart Namespace Reference

Home-ino This is the dokumentation of the Source Code from the project Smart Home.

4.1.1 Detailed Description

Home-ino This is the dokumentation of the Source Code from the project Smart Home.

Author

Okan Dogtas odogtas@uni-bremen.de

Date

2018-06-20

Copyright

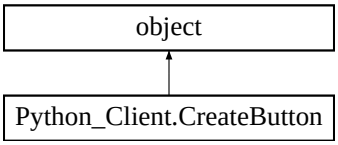
Chapter 5

Class Documentation

5.1 Python_Client.CreateButton Class Reference

This class creates default buttons.

Inheritance diagram for Python_Client.CreateButton:



Public Member Functions

- def __init__ (self, parent, content, size, command)

  The default constructor.

Public Attributes

- **button**

5.1.1 Detailed Description

This class creates default buttons.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 __init__()

```
def Python_Client.CreateButton.__init__ (
              self,
              parent,
              content,
              size,
              command )
```

The default constructor.

in the instructor the button will be created with the given informations

Parameters

parent	the parent frame / window of the button
content	the text, which will be written in the button
size	the font size of the button text
command	the executed command by pressing the button

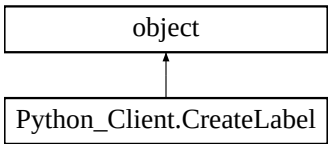
The documentation for this class was generated from the following file:

- Python_Client.py

5.2 Python_Client.CreateLabel Class Reference

This class creates labels.

Inheritance diagram for Python_Client.CreateLabel:



Public Member Functions

- def __init__ (self, parent, content, size, color)
  The default constructor.

Public Attributes

- **label**

5.2.1 Detailed Description

This class creates labels.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 __init__()

```
def Python_Client.CreateLabel.__init__ (
              self,
              parent,
              content,
              size,
              color )
```

The default constructor.

in the instructor the label will be created with the given informations

Parameters

parent	the parent frame / window of the label
content	the content text, which will be written in the label
size	the font size of the label text
color	the color of the label text

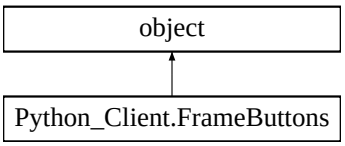
The documentation for this class was generated from the following file:

- Python_Client.py

5.3 Python_Client.FrameButtons Class Reference

This class handles the frame switch buttons.

Inheritance diagram for Python_Client.FrameButtons:



Public Member Functions

- def __init__ (self, frame, name, nr)

  The default constructor.
- def create_button (self)

  creates the button
- def frame_switch (self)

  switches frames acording to the pressed button

Public Attributes

- **frame**
- **name**
- **nr**
- **button**

5.3.1 Detailed Description

This class handles the frame switch buttons.

The class FrameButtons creates and handles the frame switch buttons

5.3.2 Constructor & Destructor Documentation

5.3.2.1 __init__()

```
def Python_Client.FrameButtons.__init__ (
              self,
              frame,
              name,
              nr )
```

The default constructor.

here the informations for the buttons will be saved in the Class Object

Parameters

frame	contains the parent frame / window of the button
name	contains the text/name of the button
nr	contains the number/id of the button (for the frame_switch method)

5.3.3 Member Function Documentation

5.3.3.1 create_button()

```
def Python_Client.FrameButtons.create_button (
              self )
```

creates the button

this method creates the button witch the according colors and command and places them with the .grid method from tkinter

5.3.3.2 frame_switch()

```
def Python_Client.FrameButtons.frame_switch (
              self )
```

switches frames acording to the pressed button

this method is executed, when a frame switch button is pressed, so that the frames switch accordingly

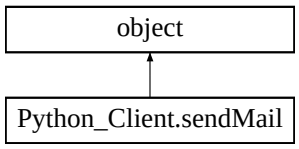
The documentation for this class was generated from the following file:

- Python_Client.py

5.4 Python_Client.sendMail Class Reference

class to send emails

Inheritance diagram for Python_Client.sendMail:



Public Member Functions

- def __init__ (self)
  The default constructor.
- def sendmail (self, receiver, msg)
  method to send an E-Mail
- def close (self)
  method to close the SMTP server connection

Public Attributes

- **server**
- **receiver**
- **msg**

5.4.1 Detailed Description

class to send emails

in this class an email can be send to an defined adress

5.4.2 Constructor & Destructor Documentation

5.4.2.1 __init__()

```
def Python_Client.sendMail.__init__ (
              self )
```

The default constructor.

the Smtp server will be initialized in this constructor

Parameters

server	the SMTP server connection (here gmail)

5.4.3 Member Function Documentation

5.4.3.1 sendmail()

```
def Python_Client.sendMail.sendmail (
              self,
              receiver,
              msg )
```

method to send an E-Mail

here the the e-mail will be send wit a given message and receiver

Parameters

receiver	e-mail adress to which the e-mail will be send
msg	the content of the e-mail

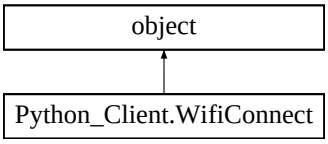
The documentation for this class was generated from the following file:

- Python_Client.py

5.5 Python_Client.WifiConnect Class Reference

a class for the Wifi connection

Inheritance diagram for Python_Client.WifiConnect:



Public Member Functions

- def __init__ (self, window, TCP_IP, TCP_PORT, on_receive)
  the default constructor
- def send_msg (self, message)
  Sends message to the arduino.
- def close (self)
  close the connection
- def periodic_socket_check (self)
  checks for received message

Public Attributes

- **window**
- **on_receive**
- **socket**
- rd_buff

  connect to the arduino
- after_event

  if a message has been received, the message will be send to the on_receive method

5.5.1 Detailed Description

a class for the Wifi connection

This class handles the connection between the arduino and the python program and also contains the methods to send and receive messages through the wirelles connection

5.5.2 Constructor & Destructor Documentation

5.5.2.1 __init__()

```
def Python_Client.WifiConnect.__init__ (
              self,
              window,
              TCP_IP,
              TCP_PORT,
              on_receive )
```

the default constructor

This method handles the connection between the arduino and the python program as a client

Parameters

window	The parent (root) window
TCP_IP	Ip addres of the arduino
TCP_PORT	Port of the arduino
on_receive	function to run when receiving a message from the arduino
window	parent window safed as local variable in the class
on_receive	on_receive function safed as local variable in the class
socket	safe the socket library as socket

5.5.3 Member Function Documentation

5.5.3.1 close()

```
def Python_Client.WifiConnect.close (
              self )
```

close the connection

This method closes the connection between the arduino and the python program

5.5.3.2 periodic_socket_check()

```
def Python_Client.WifiConnect.periodic_socket_check (
              self )
```

checks for received message

This method checks, if a message from the arduino has been send to the python programm in a periodic frequence

Parameters

line	here the received message is saved

5.5.3.3 send_msg()

```
def Python_Client.WifiConnect.send_msg (
              self,
              message )
```

Sends message to the arduino.

This method sends a message to the ESP server (arduino)

Parameters

message	contains the message to be send in bytes

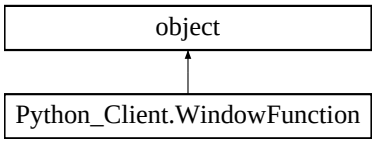
The documentation for this class was generated from the following file:

- Python_Client.py

5.6 Python_Client.WindowFunction Class Reference

The class to setup the main tkinter window.

Inheritance diagram for Python_Client.WindowFunction:



Public Member Functions

- def __init__ (self)

  The default constructor.
- def on_receive (self, line)

  handels the received commands from the Arduino
- def setup_window (self)

  this method creates the window
- def window_close (self)

  handler for Window close
- def run (self)

  runs the tkinter module to show the window
- def setup_framebtn (self)

  sets the frame buttons on window top
- def setup_frames (self)

  this method sets the frames for the main functions of the project
- def forget_frames (self)

  let all frames dissapear (pack_forget)
- def setup_rgbframe (self)

  setup the rgb-frame widgets
- def setup_alarmframe (self)

  setup the alarm-frame widgets
- def setup_welcomeframe (self)

  setup the welcome-frame widgets
- def alarm_handler (self)

  method for the alarm button
- def choose_color (self)

  method for the color choose button
- def lights_off (self)

  method for the lights off button

Public Attributes

- **rgb_red**
- **rgb_green**
- **rgb_blue**
- **arduino**
- **mailer**
- **alarm_status**
- **window**
- **frame**
- **btn1**
- **btn2**
- **btn3**

- **frame_rgb**
- **frame_alarm**
- **frame_welcome**
- **rgb_title**
- **color_btn**
- **rgb_text**
- **rgb_description**
- **rgb_off_btn**
- **alarm_title**
- **alarm_text**
- **alarm_description**
- **alarm_btn**
- **welcome_title**
- **welcome_text**
- **welcome_description**
- rgb

  the colorchooser.askcolor gives the rgb values as an two dimensional array e.g.

5.6.1 Detailed Description

The class to setup the main tkinter window.

in this class the main window will created and the contend setup is executed

5.6.2 Constructor & Destructor Documentation

5.6.2.1 __init__()

```
def Python_Client.WindowFunction.__init__ (
              self )
```

The default constructor.

The arduino and Email server connection will be initialized and all window setup methods will be executed

Parameters

rgb_red	red value for the rgb light
rgb_green	green value for the rgb light
rgb_blue	blue value for the rgb light
arduino	the WifiConnection object
mailer	the sendMail object
host	the IP address for the wifi connection
port	the port for the wifi connection

5.6.3 Member Function Documentation

5.6.3.1 alarm_handler()

```
def Python_Client.WindowFunction.alarm_handler (
              self )
```

method for the alarm button

this method will be executed, when the alarm button is pressed

the method will send messages to the arduino acordingly to the alarmstatus and also change the alarm button configuration. at the start of the message to the arduino there is alarm written, so that the arduino will know for what the message will be

Parameters

alarm_status	status of the alarm (0 = alarm off ; 1 = alarm active ; 2 = alarm triggered)

5.6.3.2 choose_color()

```
def Python_Client.WindowFunction.choose_color (
              self )
```

method for the color choose button

this method will be executed, when the color_btn is pressed

this method asks for a color and saves it in the rgb variables, sends them to the arduino and changes the color of the button to the choosen color. at the start of the message there is rgb written, so that the arduino will know for what the message will be

Parameters

rgb	

5.6.3.3 forget_frames()

```
def Python_Client.WindowFunction.forget_frames (
              self )
```

let all frames dissapear (pack_forget)

here all main frames will be unpacked, so they are not visible and the frame-switch-buttons color will be set to default

this method will be used in the FrameButtons class for the frame_switch method

5.6.3.4 lights_off()

```
def Python_Client.WindowFunction.lights_off (
              self )
```

method for the lights off button

this method will be executed, when the rgb_off_btn is pressed

this method sets the rgb variables to zero and sends these values to the arduino like in choose_color() method

5.6.3.5 on_receive()

```
def Python_Client.WindowFunction.on_receive (
              self,
              line )
```

handels the received commands from the Arduino

receives the the message from the arduino

when the message is "alarm" the rgb light will turn red, the button will be adjusted and the alarm status will turn to 2

Parameters

alarm_status	status of the alarm (0 = alarm off ; 1 = alarm active ; 2 = alarm triggered)

5.6.3.6 setup_alarmframe()

```
def Python_Client.WindowFunction.setup_alarmframe (
              self )
```

setup the alarm-frame widgets

in this method the contend of the alarm_frame will be created and placed

Parameters

alarm_title	the label object for the title of the frame
alarm_text	the description text
alarm_description	the description label object with the description text as content
alarm_btn	the button for the alarm control

5.6.3.7 setup_framebtn()

```
def Python_Client.WindowFunction.setup_framebtn (
```

```
                self )
```

sets the frame buttons on window top

the frame-switch-button frame will created and right after the frame-switch-buttons

Parameters

frame	the parent frame for the window switch buttons
btn1	first frame-switch-button "LIGHT"
btn2	second frame-switch-button "ALARM"
btn3	third frame-switch-button "WELCOME"

5.6.3.8 setup_frames()

```
def Python_Client.WindowFunction.setup_frames (
                self )
```

this method sets the frames for the main functions of the project

the main frames will be created and the on program start opened frame will be packed, so it is visible

Parameters

frame_rgb	
frame_alarm	
frame_welcome	

5.6.3.9 setup_rgbframe()

```
def Python_Client.WindowFunction.setup_rgbframe (
                self )
```

setup the rgb-frame widgets

in this method the contend of the rgb_frame will be created and placed

Parameters

rgb_title	the label object for the title of the frame
color_btn	the button for the color selection
rgb_text	the description text
rgb_description	the description label object with the description text as content
rgb_off_btn	button to set the rgb values to zero (turn rgb led off)

5.6.3.10 setup_welcomeframe()

```
def Python_Client.WindowFunction.setup_welcomeframe (
              self )
```

setup the welcome-frame widgets

in this method the contend of the welcome_frame will be created and placed

Parameters

welcome_title	the label object for the title of the frame
welcome_text	the description text
welcome_description	the description label object with the description text as content

5.6.3.11 setup_window()

```
def Python_Client.WindowFunction.setup_window (
              self )
```

this method creates the window

this method creates the window with the a given title, geometry and an window delete protocol (method window_↩ close)

5.6.3.12 window_close()

```
def Python_Client.WindowFunction.window_close (
              self )
```

handler for Window close

by closing the window first the arduino and SMTP connection will be closed

5.6.4 Member Data Documentation

5.6.4.1 rgb

```
Python_Client.WindowFunction.rgb
```

the colorchooser.askcolor gives the rgb values as an two dimensional array e.g.

[(255, 255, 255),(#FFFFFF)]

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

- Python_Client.py

Index

__init__
    Python_Client::CreateButton, 9
    Python_Client::CreateLabel, 11
    Python_Client::FrameButtons, 13
    Python_Client::WifiConnect, 16
    Python_Client::WindowFunction, 19
    Python_Client::sendMail, 14

alarm_handler
    Python_Client::WindowFunction, 20

choose_color
    Python_Client::WindowFunction, 20
close
    Python_Client::WifiConnect, 16
create_button
    Python_Client::FrameButtons, 13

forget_frames
    Python_Client::WindowFunction, 20
frame_switch
    Python_Client::FrameButtons, 13

lights_off
    Python_Client::WindowFunction, 20

on_receive
    Python_Client::WindowFunction, 21

periodic_socket_check
    Python_Client::WifiConnect, 17
Python_Client.CreateButton, 9
Python_Client.CreateLabel, 11
Python_Client.FrameButtons, 12
Python_Client.sendMail, 14
Python_Client.WifiConnect, 15
Python_Client.WindowFunction, 17
Python_Client::CreateButton
    __init__, 9
Python_Client::CreateLabel
    __init__, 11
Python_Client::FrameButtons
    __init__, 13
    create_button, 13
    frame_switch, 13
Python_Client::WifiConnect
    __init__, 16
    close, 16
    periodic_socket_check, 17
    send_msg, 17
Python_Client::WindowFunction
    __init__, 19
    alarm_handler, 20
    choose_color, 20
    forget_frames, 20
    lights_off, 20
    on_receive, 21
    rgb, 23
    setup_alarmframe, 21
    setup_framebtn, 21
    setup_frames, 22
    setup_rgbframe, 22
    setup_welcomeframe, 23
    setup_window, 23
    window_close, 23
Python_Client::sendMail
    __init__, 14
    sendmail, 15

rgb
    Python_Client::WindowFunction, 23

send_msg
    Python_Client::WifiConnect, 17
sendmail
    Python_Client::sendMail, 15
setup_alarmframe
    Python_Client::WindowFunction, 21
setup_framebtn
    Python_Client::WindowFunction, 21
setup_frames
    Python_Client::WindowFunction, 22
setup_rgbframe
    Python_Client::WindowFunction, 22
setup_welcomeframe
    Python_Client::WindowFunction, 23
setup_window
    Python_Client::WindowFunction, 23
Smart, 7

window_close
    Python_Client::WindowFunction, 23