

Stress Controller Device C++ Code

Generated by Doxygen 1.8.15

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 CppTimer Class Reference	7
4.1.1 Constructor & Destructor Documentation	7
4.1.1.1 CppTimer()	7
4.1.1.2 ~CppTimer()	7
4.1.2 Member Function Documentation	8
4.1.2.1 start()	8
4.1.2.2 timerEvent()	8
4.2 Fir1 Class Reference	8
4.2.1 Detailed Description	8
4.2.2 Constructor & Destructor Documentation	9
4.2.2.1 Fir1() [1/4]	9
4.2.2.2 Fir1() [2/4]	10
4.2.2.3 Fir1() [3/4]	10
4.2.2.4 Fir1() [4/4]	10
4.2.2.5 ~Fir1()	11
4.2.3 Member Function Documentation	11
4.2.3.1 filter()	11
4.2.3.2 getLearningRate()	11
4.2.3.3 getTapInputPower()	11
4.2.3.4 getTaps()	11
4.2.3.5 lms_update()	11
4.2.3.6 reset()	12
4.2.3.7 setLearningRate()	12
4.2.3.8 zeroCoeff()	12
4.3 ObtainData Class Reference	12
4.4 VEML6030rpi Class Reference	13
4.4.1 Constructor & Destructor Documentation	13
4.4.1.1 VEML6030rpi()	13
4.4.1.2 ~VEML6030rpi()	14
4.4.2 Member Function Documentation	14
4.4.2.1 AutoSet()	14
4.4.2.2 Gain2Bits()	14
4.4.2.3 getALS()	14

4.4.2.4 getALS_INT()	14
4.4.2.5 getWhite()	14
4.4.2.6 init()	15
4.4.2.7 IntTime2Bits()	15
4.4.2.8 PowerOn()	15
4.4.2.9 powerSaving()	15
4.4.2.10 setALS()	15
4.4.2.11 setALS_WH()	15
4.4.2.12 setALS_WL()	15
4.4.2.13 SetGain()	16
4.4.2.14 SetIntTime()	16
4.4.2.15 SetResolution()	16
4.4.2.16 Shutdown()	16
4.4.3 Member Data Documentation	16
4.4.3.1 als	16
4.4.3.2 lux	16
4.4.3.3 resolution	17
4.4.3.4 white	17
4.4.3.5 whitelux	17
5 File Documentation	19
5.1 C:/Users/Mufasa/Desktop/123/Final_V3/CppTimer.h File Reference	19
5.1.1 Macro Definition Documentation	19
5.1.1.1 CLOCKID	19
5.1.1.2 SIG	20
5.2 C:/Users/Mufasa/Desktop/123/Final_V3/Fir1.cpp File Reference	20
5.3 C:/Users/Mufasa/Desktop/123/Final_V3/Fir1.h File Reference	20
5.4 C:/Users/Mufasa/Desktop/123/Final_V3/StressController.cpp File Reference	20
5.4.1 Macro Definition Documentation	22
5.4.1.1 PIN	22
5.4.2 Function Documentation	22
5.4.2.1 alarm()	22
5.4.2.2 init()	23
5.4.2.3 main()	23
5.4.2.4 startalarm()	23
5.4.3 Variable Documentation	23
5.4.3.1 cnt	23
5.4.3.2 config	23
5.4.3.3 configfname	23
5.4.3.4 configstat	23
5.4.3.5 configstep	24
5.4.3.6 configtxt	24

5.4.3.7 ctr	24
5.4.3.8 duserrest	24
5.4.3.9 duserrun	24
5.4.3.10 duserstress	24
5.4.3.11 eventlogfname	24
5.4.3.12 eventlogtxt	24
5.4.3.13 exitfname	25
5.4.3.14 exittxt	25
5.4.3.15 fir	25
5.4.3.16 ipulse	25
5.4.3.17 iuserrest	25
5.4.3.18 iuserrun	25
5.4.3.19 iuserstress	25
5.4.3.20 mpulse	25
5.4.3.21 myexit	26
5.4.3.22 pbpm	26
5.4.3.23 pd	26
5.4.3.24 pulse	26
5.4.3.25 restctr	26
5.4.3.26 restT	26
5.4.3.27 runctr	26
5.4.3.28 runT	26
5.4.3.29 scale	27
5.4.3.30 session	27
5.4.3.31 sessionbpm	27
5.4.3.32 sessionfname	27
5.4.3.33 sessiontxt	27
5.4.3.34 statpulse	27
5.4.3.35 status	27
5.4.3.36 statusfname	28
5.4.3.37 statustxt	28
5.4.3.38 tbpm	28
5.4.3.39 tempbpm	28
5.4.3.40 userfound	28
5.4.3.41 username	28
5.4.3.42 usernamefname	28
5.4.3.43 usernametxt	28
5.4.3.44 usersfname	29
5.4.3.45 userstxt	29
5.4.3.46 veml6030	29
5.4.3.47 whitefname	29
5.4.3.48 whitetxt	29

5.5 C:/Users/Mufasa/Desktop/123/Final_V3/VEML6030rpi.cpp File Reference	29
5.5.1 Macro Definition Documentation	30
5.5.1.1 _ADR	30
5.5.1.2 ALS_CMD	30
5.5.1.3 ALS_CONF	30
5.5.1.4 ALS_INT	30
5.5.1.5 ALS_WH	31
5.5.1.6 ALS_WL	31
5.5.1.7 GAIN_1	31
5.5.1.8 GAIN_1_4	31
5.5.1.9 GAIN_1_8	31
5.5.1.10 GAIN_2	31
5.5.1.11 IT100	31
5.5.1.12 IT200	31
5.5.1.13 IT25	32
5.5.1.14 IT400	32
5.5.1.15 IT50	32
5.5.1.16 IT800	32
5.5.1.17 PWR_SVG	32
5.5.1.18 WHITE_CMD	32
5.6 C:/Users/Mufasa/Desktop/123/Final_V3/VEML6030rpi.h File Reference	32
Index	33

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

CppTimer	7
ObtainData	12
Fir1	8
VEML6030rpi	13

Chapter 2

Class Index

2.1 Class List

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

CppTimer	7
Fir1	8
ObtainData	12
VEML6030rpi	13

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

C:/Users/Mufasa/Desktop/123/Final_V3/CppTimer.h	19
C:/Users/Mufasa/Desktop/123/Final_V3/Fir1.cpp	20
C:/Users/Mufasa/Desktop/123/Final_V3/Fir1.h	20
C:/Users/Mufasa/Desktop/123/Final_V3/StressController.cpp	20
C:/Users/Mufasa/Desktop/123/Final_V3/VEML6030rpi.cpp	29
C:/Users/Mufasa/Desktop/123/Final_V3/VEML6030rpi.h	32

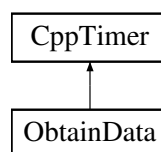
Chapter 4

Class Documentation

4.1 CppTimer Class Reference

```
#include <CppTimer.h>
```

Inheritance diagram for CppTimer:



Public Member Functions

- [CppTimer](#) ()
- virtual [~CppTimer](#) ()
- void [start](#) (long nanosecs)
- virtual void [timerEvent](#) ()=0

4.1.1 Constructor & Destructor Documentation

4.1.1.1 CppTimer()

```
CppTimer::CppTimer ( ) [inline]
```

4.1.1.2 ~CppTimer()

```
virtual CppTimer::~~CppTimer ( ) [inline], [virtual]
```

4.1.2 Member Function Documentation

4.1.2.1 start()

```
void CppTimer::start (
    long nanosecs ) [inline]
```

4.1.2.2 timerEvent()

```
virtual void CppTimer::timerEvent ( ) [pure virtual]
```

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

- C:/Users/Mufasa/Desktop/123/Final_V3/CppTimer.h

4.2 Fir1 Class Reference

```
#include <Fir1.h>
```

Public Member Functions

- `template<unsigned nTaps>`
`Fir1` (const double(&_coefficients)[nTaps])
- `Fir1` (double *_coefficients, unsigned number_of_taps)
- `Fir1` (const char *_coeffFile, unsigned number_of_taps=0)
- `Fir1` (unsigned number_of_taps)
- `~Fir1` ()
- double `filter` (double input)
- void `lms_update` (double error)
- void `setLearningRate` (double _mu)
- double `getLearningRate` ()
- void `reset` ()
- void `zeroCoeff` ()
- unsigned `getTaps` ()
- double `getTapInputPower` ()

4.2.1 Detailed Description

Finite impulse response filter. The precision is double. It takes as an input a file with coefficients or an double array.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 Fir1() [1/4]

```
template<unsigned nTaps>
Fir1::Fir1 (
    const double(&) _coefficients[nTaps] ) [inline]
```

Coefficients as a const double array. Because the array is const the number of taps is identical to the length of the array.

Parameters

<i>_coefficients</i>	A const double array with the impulse response.
----------------------	---

4.2.2.2 Fir1() [2/4]

```
Fir1::Fir1 (
    double * coefficients,
    unsigned number_of_taps )
```

Coefficients as a (non-constant-) double array where the length needs to be specified.

Parameters

<i>coefficients</i>	Coefficients as double array.
<i>number_of_taps</i>	Number of taps (needs to match the number of coefficients)

4.2.2.3 Fir1() [3/4]

```
Fir1::Fir1 (
    const char * coeffFile,
    unsigned number_of_taps = 0 )
```

Coefficients as a text file (for example from Python) The number of taps is automatically detected when the taps are kept zero.

Parameters

<i>coeffFile</i>	Path to textfile where every line contains one coefficient
<i>number_of_taps</i>	Number of taps (0 = autodetect)

4.2.2.4 Fir1() [4/4]

```
Fir1::Fir1 (
    unsigned number_of_taps )
```

Initiates all coefficients and the buffer to zero

This is useful for adaptive filters where we start with zero valued coefficients.

4.2.2.5 ~Fir1()

```
Fir1::~~Fir1 ( )
```

Releases the coefficients and buffer.

4.2.3 Member Function Documentation

4.2.3.1 filter()

```
double Fir1::filter (
    double input )
```

The actual filter function operation: it receives one sample and returns one sample.

Parameters

<i>input</i>	The input sample.
--------------	-------------------

4.2.3.2 getLearningRate()

```
double Fir1::getLearningRate ( ) [inline]
```

Getting the learning rate for the adaptive filter.

4.2.3.3 getTapInputPower()

```
double Fir1::getTapInputPower ( )
```

Returns the power of the of the buffer content:

$\text{sum_k } \text{buffer}[k]^2$ which is needed to implement a normalised LMS algorithm.

4.2.3.4 getTaps()

```
unsigned Fir1::getTaps ( ) [inline]
```

Returns the number of taps.

4.2.3.5 lms_update()

```
void Fir1::lms_update (
    double error )
```

LMS adaptive filter weight update:

Every filter coefficient is updated with: $w_k(n+1) = w_k(n) + \text{learning_rate} * \text{buffer_k}(n) * \text{error}(n)$

Parameters

<i>error</i>	Is the term error(n), the error which adjusts the FIR coefficients.
--------------	---

4.2.3.6 reset()

```
void Fir1::reset ( )
```

Resets the buffer (but not the coefficients)

4.2.3.7 setLearningRate()

```
void Fir1::setLearningRate (
    double _mu ) [inline]
```

Setting the learning rate for the adaptive filter.

Parameters

<i>_mu</i>	The learning rate (i.e. rate of the change by the error signal)
------------	---

4.2.3.8 zeroCoeff()

```
void Fir1::zeroCoeff ( )
```

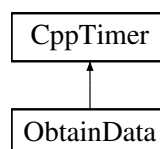
Sets all coefficients to zero

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

- C:/Users/Mufasa/Desktop/123/Final_V3/Fir1.h
- C:/Users/Mufasa/Desktop/123/Final_V3/Fir1.cpp

4.3 ObtainData Class Reference

Inheritance diagram for ObtainData:



Additional Inherited Members

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

- [C:/Users/Mufasa/Desktop/123/Final_V3/StressController.cpp](#)

4.4 VEML6030rpi Class Reference

```
#include <VEML6030rpi.h>
```

Public Member Functions

- [VEML6030rpi](#) ()
- virtual [~VEML6030rpi](#) ()
- void [init](#) (uint8_t _ADR)
- void [setALS](#) (uint16_t cmd)
- void [setALS_WH](#) (uint16_t wh)
- void [setALS_WL](#) (uint16_t wl)
- uint8_t [SetIntTime](#) (uint16_t TimeBits)
- uint8_t [SetGain](#) (uint16_t GainVal)
- uint16_t [getALS](#) (void)
- uint16_t [getWhite](#) (void)
- uint16_t [getALS_INT](#) (void)
- unsigned int [IntTime2Bits](#) (unsigned int Time)
- uint16_t [Gain2Bits](#) (float GainVal)
- void [powerSaving](#) (uint16_t ps)
- uint8_t [Shutdown](#) (void)
- uint8_t [PowerOn](#) (void)
- void [SetResolution](#) (float Gain, float IntTime)
- void [AutoSet](#) (void)

Public Attributes

- uint16_t [als](#)
- uint16_t [white](#)
- float [resolution](#)
- float [lux](#)
- float [whitelux](#)

4.4.1 Constructor & Destructor Documentation

4.4.1.1 VEML6030rpi()

```
VEML6030rpi::VEML6030rpi ( )
```

4.4.1.2 ~VEML6030rpi()

```
VEML6030rpi::~~VEML6030rpi ( ) [virtual]
```

4.4.2 Member Function Documentation

4.4.2.1 AutoSet()

```
void VEML6030rpi::AutoSet (
    void )
```

4.4.2.2 Gain2Bits()

```
uint16_t VEML6030rpi::Gain2Bits (
    float GainVal )
```

4.4.2.3 getALS()

```
uint16_t VEML6030rpi::getALS (
    void )
```

4.4.2.4 getALS_INT()

```
uint16_t VEML6030rpi::getALS_INT (
    void )
```

4.4.2.5 getWhite()

```
uint16_t VEML6030rpi::getWhite (
    void )
```

4.4.2.6 init()

```
void VEML6030rpi::init (
    uint8_t _ADR )
```

4.4.2.7 IntTime2Bits()

```
unsigned int VEML6030rpi::IntTime2Bits (
    unsigned int Time )
```

4.4.2.8 PowerOn()

```
uint8_t VEML6030rpi::PowerOn (
    void )
```

4.4.2.9 powerSaving()

```
void VEML6030rpi::powerSaving (
    uint16_t ps )
```

4.4.2.10 setALS()

```
void VEML6030rpi::setALS (
    uint16_t cmd )
```

4.4.2.11 setALS_WH()

```
void VEML6030rpi::setALS_WH (
    uint16_t wh )
```

4.4.2.12 setALS_WL()

```
void VEML6030rpi::setALS_WL (
    uint16_t wl )
```

4.4.2.13 SetGain()

```
uint8_t VEML6030rpi::SetGain (
    uint16_t GainVal )
```

4.4.2.14 SetIntTime()

```
uint8_t VEML6030rpi::SetIntTime (
    uint16_t TimeBits )
```

4.4.2.15 SetResolution()

```
void VEML6030rpi::SetResolution (
    float Gain,
    float IntTime )
```

4.4.2.16 Shutdown()

```
uint8_t VEML6030rpi::Shutdown (
    void )
```

4.4.3 Member Data Documentation

4.4.3.1 als

```
uint16_t VEML6030rpi::als
```

4.4.3.2 lux

```
float VEML6030rpi::lux
```

4.4.3.3 resolution

```
float VEML6030rpi::resolution
```

4.4.3.4 white

```
uint16_t VEML6030rpi::white
```

4.4.3.5 whitelux

```
float VEML6030rpi::whitelux
```

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

- C:/Users/Mufasa/Desktop/123/Final_V3/[VEML6030rpi.h](#)
- C:/Users/Mufasa/Desktop/123/Final_V3/[VEML6030rpi.cpp](#)

Chapter 5

File Documentation

5.1 C:/Users/Mufasa/Desktop/123/Final_V3/CppTimer.h File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <signal.h>
#include <time.h>
```

Classes

- class [CppTimer](#)

Macros

- #define [CLOCKID](#) CLOCK_MONOTONIC
- #define [SIG](#) SIGRTMIN

5.1.1 Macro Definition Documentation

5.1.1.1 CLOCKID

```
#define CLOCKID CLOCK_MONOTONIC
```

GNU GENERAL PUBLIC LICENSE Version 3, 29 June 2007

(C) 2018, Bernd Porr mail@bernporr.me.uk

This is inspired by the timer_create man page.

5.1.1.2 SIG

```
#define SIG SIGRTMIN
```

5.2 C:/Users/Mufasa/Desktop/123/Final_V3/Fir1.cpp File Reference

```
#include "Fir1.h"  
#include <string.h>  
#include <stdlib.h>  
#include <assert.h>  
#include <stdexcept>
```

5.3 C:/Users/Mufasa/Desktop/123/Final_V3/Fir1.h File Reference

```
#include <stdio.h>
```

Classes

- class [Fir1](#)

5.4 C:/Users/Mufasa/Desktop/123/Final_V3/StressController.cpp File Reference

```
#include "VEML6030rpi.h"  
#include <wiringPiI2C.h>  
#include "Fir1.h"  
#include "CppTimer.h"  
#include <fstream>  
#include <array>  
#include <time.h>  
#include <string>  
#include <unistd.h>  
#include <bits/stdc++.h>  
#include <iostream>  
#include <stdio.h>  
#include <errno.h>  
#include <wiringPi.h>  
#include <softTone.h>
```

Classes

- class [ObtainData](#)

Macros

- `#define PIN 7`

Functions

- void `startalarm` ()
- void `alarm` ()
- void `init` ()
- int `main` (void)

Variables

- int `ipulse` = 0
- int `statpulse` = 0
- int `pd` = 0
- int `tempbpm` = 80
- int `iuserrest`
- int `iuserrun`
- int `iuserstress`
- int `tbpm`
- int `pbpm`
- int `configstep` = 1
- int `cnt` = 1
- int `scale` [14] = {659, 659, 0, 659, 0, 523, 659, 0, 784, 0, 0, 0, 392, 0}
- int `configstat` = 1
- double `restctr` = 0
- double `runctr` = 0
- double `runT` = 0
- double `restT` = 0
- double `ctr` = 0
- double `pulse` [300] = {0}
- double `mpulse` = 0
- double `duserrest`
- double `duserrun`
- double `duserstress`
- `VEML6030rpi veml6030`
- string `status`
- string `username`
- string `myexit`
- string `session`
- string `config`
- int `sessionbpm` [20] = {0, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000}
- `Fir1 fir` ("coeffnoise.dat", 801)
- bool `userfound` = false
- ifstream `statustxt`
- string `statusfname` = "/var/www/html/Project/status.txt"
- ifstream `usernameetxt`
- string `usernamefname` = "/var/www/html/Project/username.txt"
- fstream `userstxt`
- string `usersfname` = "/var/www/html/Project/userdata.txt"
- ifstream `exittxt`

- string `exitfname` = "/var/www/html/Project/exit.txt"
- ifstream `sessiontxt`
- string `sessionfname` = "/var/www/html/Project/session.txt"
- ifstream `configtxt`
- string `configfname` = "/var/www/html/Project/config.txt"
- ofstream `whitetxt`
- string `whitefname` = "/var/www/html/Project/white.txt"
- ofstream `eventlogtxt`
- string `eventlogfname` = "/var/www/html/Project/eventlog.txt"

5.4.1 Macro Definition Documentation

5.4.1.1 PIN

```
#define PIN 7
```

"Monitoring of stress level" University of Glasgow Supervised by: Bernd Porr By Ahmed Elmogamer, Daewon Jung and Gabriel Galeote Checa

Official project location: https://github.com/GGChe/Stress_Controller_Device

Open Readme for details about the project

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

5.4.2 Function Documentation

5.4.2.1 alarm()

```
void alarm ( )
```

5.4.2.2 init()

```
void init ( )
```

5.4.2.3 main()

```
int main (
    void )
```

Firstly, we need to check whether status is 0, 1 or 2. Then, if status is equals to 0, we are in the main webpage. if status is equals to 1, we are logged but the user already exists to we don't need to calibrate if status is equals to 2, we need to signup and calibrate.

5.4.2.4 startalarm()

```
void startalarm ( )
```

5.4.3 Variable Documentation

5.4.3.1 cnt

```
int cnt = 1
```

5.4.3.2 config

```
string config
```

5.4.3.3 configfname

```
string configfname = "/var/www/html/Project/config.txt"
```

5.4.3.4 configstat

```
int configstat = 1
```

5.4.3.5 configstep

```
int configstep = 1
```

5.4.3.6 configtxt

```
ifstream configtxt
```

5.4.3.7 ctr

```
double ctr = 0
```

5.4.3.8 duserrest

```
double duserrest
```

5.4.3.9 duserrun

```
double duserrun
```

5.4.3.10 duserstress

```
double duserstress
```

5.4.3.11 eventlogfname

```
string eventlogfname = "/var/www/html/Project/eventlog.txt"
```

5.4.3.12 eventlogtxt

```
ofstream eventlogtxt
```

5.4.3.13 exitfname

```
string exitfname = "/var/www/html/Project/exit.txt"
```

5.4.3.14 exittxt

```
ifstream exittxt
```

5.4.3.15 fir

```
Fir1 fir("coeffnoise.dat", 801)
```

5.4.3.16 ipulse

```
int ipulse = 0
```

5.4.3.17 iuserrest

```
int iuserrest
```

5.4.3.18 iuserrun

```
int iuserrun
```

5.4.3.19 iuserstress

```
int iuserstress
```

5.4.3.20 mpulse

```
double mpulse = 0
```

5.4.3.21 myexit

```
string myexit
```

5.4.3.22 pbpm

```
int pbpm
```

5.4.3.23 pd

```
int pd = 0
```

5.4.3.24 pulse

```
double pulse[300] = {0}
```

5.4.3.25 restctr

```
double restctr = 0
```

5.4.3.26 restT

```
double restT = 0
```

5.4.3.27 runctr

```
double runctr = 0
```

5.4.3.28 runT

```
double runT = 0
```


5.4.3.29 scale

```
int scale[14] = {659, 659, 0, 659, 0, 523, 659, 0, 784, 0, 0, 0, 392, 0}
```

5.4.3.30 session

```
string session
```

5.4.3.31 sessionbpm

```
int sessionbpm[20] = {0, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000}
```

5.4.3.32 sessionfname

```
string sessionfname = "/var/www/html/Project/session.txt"
```

5.4.3.33 sessiontxt

```
ifstream sessiontxt
```

5.4.3.34 statpulse

```
int statpulse = 0
```

5.4.3.35 status

```
string status
```

5.4.3.36 statusfname

```
string statusfname = "/var/www/html/Project/status.txt"
```

5.4.3.37 statustxt

```
ifstream statustxt
```

5.4.3.38 tbpm

```
int tbpm
```

5.4.3.39 tempbpm

```
int tempbpm = 80
```

5.4.3.40 userfound

```
bool userfound = false
```

5.4.3.41 username

```
string username
```

5.4.3.42 usernamefname

```
string usernamefname = "/var/www/html/Project/username.txt"
```

5.4.3.43 usernametxt

```
ifstream usernametxt
```

5.4.3.44 usersfname

```
string usersfname = "/var/www/html/Project/userdata.txt"
```

5.4.3.45 userstxt

```
fstream userstxt
```

5.4.3.46 veml6030

```
VEML6030rpi veml6030
```

5.4.3.47 whitefname

```
string whitefname = "/var/www/html/Project/white.txt"
```

5.4.3.48 whitetxt

```
ofstream whitetxt
```

5.5 C:/Users/Mufasa/Desktop/123/Final_V3/VEML6030rpi.cpp File Reference

```
#include "VEML6030rpi.h"  
#include <wiringPiI2C.h>  
#include <fstream>  
#include <math.h>  
#include <iostream>
```

Macros

- `#define _ADR 0x48`
- `#define ALS_CONF 0x00`
- `#define ALS_WH 0x01`
- `#define ALS_WL 0x02`
- `#define PWR_SVG 0x03`
- `#define ALS_CMD 0x04`
- `#define WHITE_CMD 0x05`
- `#define ALS_INT 0x06`
- `#define IT25 0x300`
- `#define IT50 0x200`
- `#define IT100 0x000`
- `#define IT200 0x040`
- `#define IT400 0x080`
- `#define IT800 0x0C0`
- `#define GAIN_1 0x0000`
- `#define GAIN_2 0x0800`
- `#define GAIN_1_8 0x1000`
- `#define GAIN_1_4 0x1800`

5.5.1 Macro Definition Documentation

5.5.1.1 _ADR

```
#define _ADR 0x48
```

5.5.1.2 ALS_CMD

```
#define ALS_CMD 0x04
```

5.5.1.3 ALS_CONF

```
#define ALS_CONF 0x00
```

5.5.1.4 ALS_INT

```
#define ALS_INT 0x06
```

5.5.1.5 ALS_WH

```
#define ALS_WH 0x01
```

5.5.1.6 ALS_WL

```
#define ALS_WL 0x02
```

5.5.1.7 GAIN_1

```
#define GAIN_1 0x0000
```

5.5.1.8 GAIN_1_4

```
#define GAIN_1_4 0x1800
```

5.5.1.9 GAIN_1_8

```
#define GAIN_1_8 0x1000
```

5.5.1.10 GAIN_2

```
#define GAIN_2 0x0800
```

5.5.1.11 IT100

```
#define IT100 0x000
```

5.5.1.12 IT200

```
#define IT200 0x040
```

5.5.1.13 IT25

```
#define IT25 0x300
```

5.5.1.14 IT400

```
#define IT400 0x080
```

5.5.1.15 IT50

```
#define IT50 0x200
```

5.5.1.16 IT800

```
#define IT800 0x0C0
```

5.5.1.17 PWR_SVG

```
#define PWR_SVG 0x03
```

5.5.1.18 WHITE_CMD

```
#define WHITE_CMD 0x05
```

5.6 C:/Users/Mufasa/Desktop/123/Final_V3/VEML6030rpi.h File Reference

```
#include <unistd.h>
#include <fcntl.h>
#include <string>
#include <fstream>
#include <sys/ioctl.h>
#include <linux/i2c-dev.h>
#include <stdio.h>
#include <iostream>
```

Classes

- class [VEML6030rpi](#)

Index

`_ADR`
 VEML6030rpi.cpp, 30
`~CppTimer`
 CppTimer, 7
`~Fir1`
 Fir1, 10
`~VEML6030rpi`
 VEML6030rpi, 13

`alarm`
 StressController.cpp, 22
`als`
 VEML6030rpi, 16
`ALS_CMD`
 VEML6030rpi.cpp, 30
`ALS_CONF`
 VEML6030rpi.cpp, 30
`ALS_INT`
 VEML6030rpi.cpp, 30
`ALS_WH`
 VEML6030rpi.cpp, 30
`ALS_WL`
 VEML6030rpi.cpp, 31
`AutoSet`
 VEML6030rpi, 14

C:/Users/Mufasa/Desktop/123/Final_V3/CppTimer.h, 19
C:/Users/Mufasa/Desktop/123/Final_V3/Fir1.cpp, 20
C:/Users/Mufasa/Desktop/123/Final_V3/Fir1.h, 20
C:/Users/Mufasa/Desktop/123/Final_V3/StressController.cpp, 20
C:/Users/Mufasa/Desktop/123/Final_V3/VEML6030rpi.cpp, 29
C:/Users/Mufasa/Desktop/123/Final_V3/VEML6030rpi.h, 32
`CLOCKID`
 CppTimer.h, 19
`cnt`
 StressController.cpp, 23
`config`
 StressController.cpp, 23
`configname`
 StressController.cpp, 23
`configstat`
 StressController.cpp, 23
`configstep`
 StressController.cpp, 23
`configtxt`
 StressController.cpp, 24
`CppTimer`, 7
 ~CppTimer, 7
 CppTimer, 7
 start, 8
 timerEvent, 8
`CppTimer.h`
 CLOCKID, 19
 SIG, 19
`ctr`
 StressController.cpp, 24

`duserrest`
 StressController.cpp, 24
`duserun`
 StressController.cpp, 24
`duserstress`
 StressController.cpp, 24

`eventlogfname`
 StressController.cpp, 24
`eventlogtxt`
 StressController.cpp, 24
`exitfname`
 StressController.cpp, 24
`exittxt`
 StressController.cpp, 25

`filter`
 Fir1, 11
 fir
 StressController.cpp, 25
 Fir1, 8
 ~Fir1, 10
 filter, 11
 Fir1, 9, 10
 getLearningRate, 11
 getTapInputPower, 11
 getTaps, 11
 lms_update, 11
 reset, 12
 setLearningRate, 12
 zeroCoeff, 12

`Gain2Bits`
 VEML6030rpi, 14
`GAIN_1`
 VEML6030rpi.cpp, 31
`GAIN_1_4`
 VEML6030rpi.cpp, 31
`GAIN_1_8`
 VEML6030rpi.cpp, 31

- GAIN_2
 - VEML6030rpi.cpp, 31
- getALS
 - VEML6030rpi, 14
- getALS_INT
 - VEML6030rpi, 14
- getLearningRate
 - Fir1, 11
- getTapInputPower
 - Fir1, 11
- getTaps
 - Fir1, 11
- getWhite
 - VEML6030rpi, 14
- init
 - StressController.cpp, 22
 - VEML6030rpi, 14
- IntTime2Bits
 - VEML6030rpi, 15
- ipulse
 - StressController.cpp, 25
- IT100
 - VEML6030rpi.cpp, 31
- IT200
 - VEML6030rpi.cpp, 31
- IT25
 - VEML6030rpi.cpp, 31
- IT400
 - VEML6030rpi.cpp, 32
- IT50
 - VEML6030rpi.cpp, 32
- IT800
 - VEML6030rpi.cpp, 32
- iuserrest
 - StressController.cpp, 25
- iuserun
 - StressController.cpp, 25
- iuserstress
 - StressController.cpp, 25
- lms_update
 - Fir1, 11
- lux
 - VEML6030rpi, 16
- main
 - StressController.cpp, 23
- mpulse
 - StressController.cpp, 25
- myexit
 - StressController.cpp, 25
- ObtainData, 12
- pbpm
 - StressController.cpp, 26
- pd
 - StressController.cpp, 26
- PIN
 - StressController.cpp, 22
- PowerOn
 - VEML6030rpi, 15
- powerSaving
 - VEML6030rpi, 15
- pulse
 - StressController.cpp, 26
- PWR_SVG
 - VEML6030rpi.cpp, 32
- reset
 - Fir1, 12
- resolution
 - VEML6030rpi, 16
- restctr
 - StressController.cpp, 26
- restT
 - StressController.cpp, 26
- runctr
 - StressController.cpp, 26
- runT
 - StressController.cpp, 26
- scale
 - StressController.cpp, 26
- session
 - StressController.cpp, 27
- sessionbpm
 - StressController.cpp, 27
- sessionfname
 - StressController.cpp, 27
- sessiontxt
 - StressController.cpp, 27
- setALS
 - VEML6030rpi, 15
- setALS_WH
 - VEML6030rpi, 15
- setALS_WL
 - VEML6030rpi, 15
- SetGain
 - VEML6030rpi, 15
- SetIntTime
 - VEML6030rpi, 16
- setLearningRate
 - Fir1, 12
- SetResolution
 - VEML6030rpi, 16
- Shutdown
 - VEML6030rpi, 16
- SIG
 - CppTimer.h, 19
- start
 - CppTimer, 8
- startalarm
 - StressController.cpp, 23
- statpulse
 - StressController.cpp, 27
- status

- StressController.cpp, 27
- statusfname
 - StressController.cpp, 27
- statustxt
 - StressController.cpp, 28
- StressController.cpp
 - alarm, 22
 - cnt, 23
 - config, 23
 - configfname, 23
 - configstat, 23
 - configstep, 23
 - configtxt, 24
 - ctr, 24
 - duserrest, 24
 - duserrun, 24
 - duserstress, 24
 - eventlogfname, 24
 - eventlogtxt, 24
 - exitfname, 24
 - exittxt, 25
 - fir, 25
 - init, 22
 - ipulse, 25
 - iuserrest, 25
 - iuserrun, 25
 - iuserstress, 25
 - main, 23
 - mpulse, 25
 - myexit, 25
 - pbpm, 26
 - pd, 26
 - PIN, 22
 - pulse, 26
 - restctr, 26
 - restT, 26
 - runctr, 26
 - runT, 26
 - scale, 26
 - session, 27
 - sessionbpm, 27
 - sessionfname, 27
 - sessiontxt, 27
 - startalarm, 23
 - statpulse, 27
 - status, 27
 - statusfname, 27
 - statustxt, 28
 - tbpm, 28
 - tempbpm, 28
 - userfound, 28
 - username, 28
 - usernamefname, 28
 - usernameetxt, 28
 - usersfname, 28
 - userstxt, 29
 - veml6030, 29
 - whitefname, 29
 - whitetxt, 29
- tbpm
 - StressController.cpp, 28
- tempbpm
 - StressController.cpp, 28
- timerEvent
 - CppTimer, 8
- userfound
 - StressController.cpp, 28
- username
 - StressController.cpp, 28
- usernamefname
 - StressController.cpp, 28
- usernameetxt
 - StressController.cpp, 28
- usersfname
 - StressController.cpp, 28
- userstxt
 - StressController.cpp, 29
- veml6030
 - StressController.cpp, 29
- VEML6030rpi, 13
 - ~VEML6030rpi, 13
 - als, 16
 - AutoSet, 14
 - Gain2Bits, 14
 - getALS, 14
 - getALS_INT, 14
 - getWhite, 14
 - init, 14
 - IntTime2Bits, 15
 - lux, 16
 - PowerOn, 15
 - powerSaving, 15
 - resolution, 16
 - setALS, 15
 - setALS_WH, 15
 - setALS_WL, 15
 - SetGain, 15
 - SetIntTime, 16
 - SetResolution, 16
 - Shutdown, 16
 - VEML6030rpi, 13
 - white, 17
 - whitelux, 17
- VEML6030rpi.cpp
 - _ADR, 30
 - ALS_CMD, 30
 - ALS_CONF, 30
 - ALS_INT, 30
 - ALS_WH, 30
 - ALS_WL, 31
 - GAIN_1, 31
 - GAIN_1_4, 31
 - GAIN_1_8, 31
 - GAIN_2, 31

- IT100, [31](#)
- IT200, [31](#)
- IT25, [31](#)
- IT400, [32](#)
- IT50, [32](#)
- IT800, [32](#)
- PWR_SVG, [32](#)
- WHITE_CMD, [32](#)

- white
 - VEML6030rpi, [17](#)
- WHITE_CMD
 - VEML6030rpi.cpp, [32](#)
- whitefname
 - StressController.cpp, [29](#)
- whitelux
 - VEML6030rpi, [17](#)
- whitetxt
 - StressController.cpp, [29](#)

- zeroCoeff
 - Fir1, [12](#)