

Lampete

Generated by Doxygen 1.10.0



<b>1 README</b>	<b>1</b>
<b>2 File Index</b>	<b>3</b>
2.1 File List	3
<b>3 File Documentation</b>	<b>5</b>
3.1 definitions.h File Reference	5
3.1.1 Macro Definition Documentation	5
3.1.1.1 BUTTONUPDATETIME	5
3.1.1.2 DIMHIGHLIMIT	5
3.1.1.3 DIMLOWERLIMIT	5
3.1.1.4 DIMRATE	6
3.1.1.5 DUTYSAVETIME	6
3.1.1.6 FIRSTDIMSTEP	6
3.1.1.7 FIRSTPRESS	6
3.1.1.8 LONGPRESSACTION	6
3.1.1.9 LONGPRESSWAIT	6
3.1.1.10 PWMRESOLUTION	6
3.1.1.11 WAITTIMEDIM	6
3.2 definitions.h	6
3.3 variables.h File Reference	7
3.3.1 Variable Documentation	7
3.3.1.1 analogDuty	7
3.3.1.2 b_decrease_State	7
3.3.1.3 b_decrease_Time	7
3.3.1.4 b_increase_State	7
3.3.1.5 b_increase_Time	8
3.3.1.6 b_onOff_State	8
3.3.1.7 onOff	8
3.4 variables.h	8
3.5 README.md File Reference	8
3.6 dimming.cpp File Reference	8
3.6.1 Function Documentation	9
3.6.1.1 changeDimming()	9
3.6.1.2 changePWMDuty()	9
3.6.1.3 coerceDimming()	10
3.6.1.4 lumDecrease()	10
3.6.1.5 lumIncrease()	10
3.6.1.6 switchOnOff()	10
3.6.2 Variable Documentation	11
3.6.2.1 b_decrease_Time	11
3.6.2.2 b_increase_Time	11
3.7 dimming.h File Reference	11

---

3.7.1 Function Documentation	11
3.7.1.1 changeDimming()	11
3.7.1.2 changePWMDuty()	12
3.7.1.3 coerceDimming()	12
3.7.1.4 lumDecrease()	12
3.7.1.5 lumIncrease()	12
3.7.1.6 switchOnOff()	13
3.8 dimming.h	13
3.9 main.cpp File Reference	13
3.9.1 Function Documentation	14
3.9.1.1 loop()	14
3.9.1.2 saveAnalogDuty()	14
3.9.1.3 setup()	14
3.9.1.4 updateButtons()	14
3.9.2 Variable Documentation	15
3.9.2.1 analogDuty	15
3.9.2.2 b_decrease	15
3.9.2.3 b_decrease_State	15
3.9.2.4 b_increase	15
3.9.2.5 b_increase_State	15
3.9.2.6 b_onOff	15
3.9.2.7 b_onOff_State	15
3.9.2.8 buttonReader	15
3.9.2.9 dutySave	16
3.9.2.10 onOff	16
3.9.2.11 pref	16
3.9.2.12 pwmPin	16
<b>Index</b>	<b>17</b>

# Chapter 1

## README

This project is for a dimmable lamp build inside a trumpet (there the name is coming from (Trompete means trumpet in german)). It is possible to use the circuit and the software for any lamp using 3 buttons and an uC.



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">definitions.h</a>	5
<a href="#">variables.h</a>	7
<a href="#">dimming.cpp</a>	8
<a href="#">dimming.h</a>	11
<a href="#">main.cpp</a>	13





# Chapter 3

## File Documentation

### 3.1 definitions.h File Reference

#### Macros

- `#define FIRSTPRESS 0`
- `#define LONGPRESSWAIT 1`
- `#define LONGPRESSACTION 2`
- `#define WAITTIMEDIM 1000`
- `#define PWMRESOLUTION 8`
- `#define DIMRATE (0.078125*(pow(2, PWMRESOLUTION)))`
- `#define DIMLOWERLIMIT (0.0390625*pow(2, PWMRESOLUTION))`
- `#define DIMHIGHLIMIT (pow(2, PWMRESOLUTION)-1)`
- `#define FIRSTDIMSTEP (0.03125*(pow(2, PWMRESOLUTION)))`
- `#define BUTTONUPDATETIME 15`
- `#define DUTYSAVETIME 30000`

#### 3.1.1 Macro Definition Documentation

##### 3.1.1.1 BUTTONUPDATETIME

```
#define BUTTONUPDATETIME 15
```

##### 3.1.1.2 DIMHIGHLIMIT

```
#define DIMHIGHLIMIT (pow(2, PWMRESOLUTION)-1)
```

##### 3.1.1.3 DIMLOWERLIMIT

```
#define DIMLOWERLIMIT (0.0390625*pow(2, PWMRESOLUTION))
```

### 3.1.1.4 DIMRATE

```
#define DIMRATE (0.078125*(pow(2, PWMRESOLUTION)))
```

### 3.1.1.5 DUTYSAVETIME

```
#define DUTYSAVETIME 30000
```

### 3.1.1.6 FIRSTDIMSTEP

```
#define FIRSTDIMSTEP (0.03125*(pow(2, PWMRESOLUTION)))
```

### 3.1.1.7 FIRSTPRESS

```
#define FIRSTPRESS 0
```

### 3.1.1.8 LONGPRESSACTION

```
#define LONGPRESSACTION 2
```

### 3.1.1.9 LONGPRESSWAIT

```
#define LONGPRESSWAIT 1
```

### 3.1.1.10 PWMRESOLUTION

```
#define PWMRESOLUTION 8
```

### 3.1.1.11 WAITTIMEDIM

```
#define WAITTIMEDIM 1000
```

## 3.2 definitions.h

[Go to the documentation of this file.](#)

```
00001 //Definitions for Lampete
00002
00003 #ifndef definitions_H
00004 #define definitions_H
00005
00006
00007 #define FIRSTPRESS 0
00008 #define LONGPRESSWAIT 1
00009 #define LONGPRESSACTION 2
00010
00011 #define WAITTIMEDIM 1000 // wait time for starting the dimming action
00012
00013 #define PWMRESOLUTION 8 //in bits
00014 #define DIMRATE (0.078125*(pow(2, PWMRESOLUTION))) //per second
00015 #define DIMLOWERLIMIT (0.0390625*pow(2, PWMRESOLUTION)) //lowest dimming limit
00016 #define DIMHIGHLIMIT (pow(2, PWMRESOLUTION)-1) //Highest pwm duty cycle
00017 #define FIRSTDIMSTEP (0.03125*(pow(2, PWMRESOLUTION))) //Dimming step when only pressing the
    button once
00018
00019
00020 #define BUTTONUPDATETIME 15 // in ms
00021 #define DUTYSAVETIME 30000 // in s
00022 #endif
00023 //EOF
```

## 3.3 variables.h File Reference

```
#include <stdint>
```

### Variables

- bool [onOff](#)  
*Current state of lamp.*
- uint8\_t [b\\_onOff\\_State](#)  
*Current state of state machine for on-off button.*
- uint8\_t [b\\_decrease\\_State](#)  
*Current state of state machine for decrease button.*
- uint8\_t [b\\_increase\\_State](#)  
*Current state of state machine for increase button.*
- uint32\_t [b\\_decrease\\_Time](#)  
*variable for state machine behavior. Saves time of last relevant action for decrease luminescence.*
- uint32\_t [b\\_increase\\_Time](#)  
*variable for state machine behavior. Saves time of last relevant action for increase luminescence.*
- int16\_t [analogDuty](#)  
*Current duty cycle of lamp. Only for on-state!*

### 3.3.1 Variable Documentation

#### 3.3.1.1 [analogDuty](#)

```
int16_t analogDuty [extern]
```

Current duty cycle of lamp. Only for on-state!

#### 3.3.1.2 [b\\_decrease\\_State](#)

```
uint8_t b_decrease_State [extern]
```

Current state of state machine for decrease button.

#### 3.3.1.3 [b\\_decrease\\_Time](#)

```
uint32_t b_decrease_Time [extern]
```

variable for state machine behavior. Saves time of last relevant action for decrease luminescence.

#### 3.3.1.4 [b\\_increase\\_State](#)

```
uint8_t b_increase_State [extern]
```

Current state of state machine for increase button.

### 3.3.1.5 b\_increase\_Time

```
uint32_t b_increase_Time [extern]
```

variable for state machine behavior. Saves time of last relevant action for increase luminescence.

### 3.3.1.6 b\_onOff\_State

```
uint8_t b_onOff_State [extern]
```

Current state of state machine for on-off button.

### 3.3.1.7 onOff

```
bool onOff [extern]
```

Current state of lamp.

## 3.4 variables.h

[Go to the documentation of this file.](#)

```
00001 //Variables for Lampete
00002
00003 #ifndef variables_H
00004 #define variables_H
00005
00006 #include <cstdint>
00007
00009 extern bool onOff;
00011 extern uint8_t b_onOff_State;
00013 extern uint8_t b_decrease_State;
00015 extern uint8_t b_increase_State;
00017 extern uint32_t b_decrease_Time;
00019 extern uint32_t b_increase_Time;
00021 extern int16_t analogDuty;
00022
00023 #endif
00024 //EOF
```

## 3.5 README.md File Reference

## 3.6 dimming.cpp File Reference

```
#include <Esp.h>
#include "definitions.h"
#include "variables.h"
#include "dimming.h"
```

## Functions

- void `switchOnOff` ()  
*Switches the lamps on and off.*
- void `lumIncrease` ()  
*Increases the luminescence.*
- void `lumDecrease` ()  
*Decreases the luminescence.*
- int16\_t `changePWMDuty` (uint16\_t duty, boolean coerce)  
*Changes the duty cycle for the PWM signal for the lamp.*
- int16\_t `coerceDimming` (int16\_t tempAnalogDuty)  
*Coerces the duty value in a predefined range.*
- void `changeDimming` (int8\_t dir)  
*State machine for deciding the dimming behavior when a button is pressed.*

## Variables

- uint32\_t `b_decrease_Time` = 0  
*variable for state machine behavior. Saves time of last relevant action for decrease luminescence.*
- uint32\_t `b_increase_Time` = 0  
*variable for state machine behavior. Saves time of last relevant action for increase luminescence.*

## 3.6.1 Function Documentation

### 3.6.1.1 `changeDimming()`

```
void changeDimming (  
    int8_t dir )
```

State machine for deciding the dimming behavior when a button is pressed.

#### Parameters

<i>dir</i>	dimming direction. Positive values increase the luminescence, negative decrease
------------	---

### 3.6.1.2 `changePWMDuty()`

```
int16_t changePWMDuty (  
    uint16_t duty,  
    boolean coerce = true )
```

Changes the duty cycle for the PWM signal for the lamp.

#### Parameters

<i>duty</i>	duty cycle for PWM signal
-------------	---------------------------

**Note**

Coerces the duty cycle in a range.

**3.6.1.3 coerceDimming()**

```
int16_t coerceDimming (
    int16_t tempAnalogDuty )
```

Coerces the duty value in a predefined range.

**Parameters**

<i>tempAnalogDuty</i>	duty cycle to coerce
-----------------------	----------------------

**Returns**

int16\_t coerced duty cycle

**3.6.1.4 lumDecrease()**

```
void lumDecrease ( )
```

Decreases the luminescence.

**Note**

Calls changeDimming with a negative value (-1).

**3.6.1.5 lumIncrease()**

```
void lumIncrease ( )
```

Increases the luminescence.

**Note**

Calls changeDimming with a positive value (+1).

**3.6.1.6 switchOnOff()**

```
void switchOnOff ( )
```

Switches the lamps on and off.

The function switches the duty cycle to 0 (off) or the old duty cycle value (on).

## 3.6.2 Variable Documentation

### 3.6.2.1 b\_decrease\_Time

```
uint32_t b_decrease_Time = 0
```

variable for state machine behavior. Saves time of last relevant action for decrease luminescence.

### 3.6.2.2 b\_increase\_Time

```
uint32_t b_increase_Time = 0
```

variable for state machine behavior. Saves time of last relevant action for increase luminescence.

## 3.7 dimming.h File Reference

```
#include <cstdint>
```

### Functions

- void [switchOnOff](#) ()  
*Switches the lamps on and off.*
- void [lumIncrease](#) ()  
*Increases the luminescence.*
- void [lumDecrease](#) ()  
*Decreases the luminescence.*
- void [changeDimming](#) (int8\_t dir)  
*State machine for deciding the dimming behavior when a button is pressed.*
- int16\_t [changePWMDuty](#) (uint16\_t duty, boolean coerce=true)  
*Changes the duty cycle for the PWM signal for the lamp.*
- int16\_t [coerceDimming](#) (int16\_t tempAnalogDuty)  
*Coerces the duty value in a predefined range.*

## 3.7.1 Function Documentation

### 3.7.1.1 changeDimming()

```
void changeDimming (
    int8_t dir )
```

State machine for deciding the dimming behavior when a button is pressed.

#### Parameters

<i>dir</i>	dimming direction. Positive values increase the luminescence, negative decrease
------------	---

### 3.7.1.2 changePWMDuty()

```
int16_t changePWMDuty (
    uint16_t duty,
    boolean coerce = true )
```

Changes the duty cycle for the PWM signal for the lamp.

#### Parameters

<i>duty</i>	duty cycle for PWM signal
-------------	---------------------------

#### Note

Coerces the duty cycle in a range.

### 3.7.1.3 coerceDimming()

```
int16_t coerceDimming (
    int16_t tempAnalogDuty )
```

Coerces the duty value in a predefined range.

#### Parameters

<i>tempAnalogDuty</i>	duty cycle to coerce
-----------------------	----------------------

#### Returns

int16\_t coerced duty cycle

### 3.7.1.4 lumDecrease()

```
void lumDecrease ( )
```

Decreases the luminescence.

#### Note

Calls changeDimming with a negative value (-1).

### 3.7.1.5 lumIncrease()

```
void lumIncrease ( )
```

Increases the luminescence.

#### Note

Calls changeDimming with a positive value (+1).



### 3.7.1.6 switchOnOff()

```
void switchOnOff ( )
```

Switches the lamps on and off.

The function switches the duty cycle to 0 (off) or the old duty cycle value (on).

## 3.8 dimming.h

[Go to the documentation of this file.](#)

```
00001
00002 #ifndef dimming_H
00003 #define dimming_H
00004
00005 #include <stdint>
00006
00012 void switchOnOff();
00013
00019 void lumIncrease();
00020
00026 void lumDecrease();
00027
00033 void changeDimming(int8_t dir);
00034
00041 int16_t changePWMDuty(uint16_t duty, boolean coerce = true);
00042
00049 int16_t coerceDimming(int16_t tempAnalogDuty);
00050
00051
00052 #endif
00053 //EOF
```

## 3.9 main.cpp File Reference

```
#include <Arduino.h>
#include <Esp.h>
#include <Ticker.h>
#include <ButtonDebounce.h>
#include <Preferences.h>
#include "definitions.h"
#include "variables.h"
#include "..\dimming\dimming.h"
```

### Functions

- void IRAM\_ATTR [updateButtons](#) ()  
*Function to update the button history.*
- void [saveAnalogDuty](#) ()  
*Saves the current duty cycle value.*
- void [setup](#) ()
- void [loop](#) ()

## Variables

- Ticker [buttonReader](#)
- Ticker [dutySave](#)
- Preferences [pref](#)
- ButtonDebounce [b\\_onOff](#) = ButtonDebounce(onOffPin, INPUT, true, false, [\\*switchOnOff](#))
- ButtonDebounce [b\\_decrease](#) = ButtonDebounce(decreasePin, INPUT, true, false, [\\*lumDecrease](#))
- ButtonDebounce [b\\_increase](#) = ButtonDebounce(increasePin, INPUT, true, false, [\\*lumIncrease](#))
- const int [pwmPin](#) = ledPin
- bool [onOff](#) = true
- Current state of lamp.*
- uint8\_t [b\\_onOff\\_State](#) = [FIRSTPRESS](#)
- Current state of state machine for on-off button.*
- uint8\_t [b\\_decrease\\_State](#) = [FIRSTPRESS](#)
- Current state of state machine for decrease button.*
- uint8\_t [b\\_increase\\_State](#) = [FIRSTPRESS](#)
- Current state of state machine for increase button.*
- int16\_t [analogDuty](#) = 125
- Current duty cycle of lamp. Only for on-state!*

## 3.9.1 Function Documentation

### 3.9.1.1 loop()

```
void loop ( )
```

### 3.9.1.2 saveAnalogDuty()

```
void saveAnalogDuty ( )
```

Saves the current duty cycle value.

Will be saved every DUTYSAVETIME s in uC storage.

### 3.9.1.3 setup()

```
void setup ( )
```

### 3.9.1.4 updateButtons()

```
void IRAM_ATTR updateButtons ( )
```

Function to update the button history.

The function calls the update function for all buttons used. Is called periodically every BUTTONUPDATETIME ms.

## 3.9.2 Variable Documentation

### 3.9.2.1 analogDuty

```
int16_t analogDuty = 125
```

Current duty cycle of lamp. Only for on-state!

### 3.9.2.2 b\_decrease

```
ButtonDebounce b_decrease = ButtonDebounce(decreasePin, INPUT, true, false, *lumDecrease)
```

### 3.9.2.3 b\_decrease\_State

```
uint8_t b_decrease_State = FIRSTPRESS
```

Current state of state machine for decrease button.

### 3.9.2.4 b\_increase

```
ButtonDebounce b_increase = ButtonDebounce(increasePin, INPUT, true, false, *lumIncrease)
```

### 3.9.2.5 b\_increase\_State

```
uint8_t b_increase_State = FIRSTPRESS
```

Current state of state machine for increase button.

### 3.9.2.6 b\_onOff

```
ButtonDebounce b_onOff = ButtonDebounce(onOffPin, INPUT, true, false, *switchOnOff)
```

### 3.9.2.7 b\_onOff\_State

```
uint8_t b_onOff_State = FIRSTPRESS
```

Current state of state machine for on-off button.

### 3.9.2.8 buttonReader

```
Ticker buttonReader
```

### 3.9.2.9 dutySave

Ticker dutySave

### 3.9.2.10 onOff

bool onOff = true

Current state of lamp.

### 3.9.2.11 pref

Preferences pref

### 3.9.2.12 pwmPin

const int pwmPin = ledPin

# Index

- analogDuty
  - main.cpp, [15](#)
  - variables.h, [7](#)
- b\_decrease
  - main.cpp, [15](#)
- b\_decrease\_State
  - main.cpp, [15](#)
  - variables.h, [7](#)
- b\_decrease\_Time
  - dimming.cpp, [11](#)
  - variables.h, [7](#)
- b\_increase
  - main.cpp, [15](#)
- b\_increase\_State
  - main.cpp, [15](#)
  - variables.h, [7](#)
- b\_increase\_Time
  - dimming.cpp, [11](#)
  - variables.h, [7](#)
- b\_onOff
  - main.cpp, [15](#)
- b\_onOff\_State
  - main.cpp, [15](#)
  - variables.h, [8](#)
- buttonReader
  - main.cpp, [15](#)
- BUTTONUPDATETIME
  - definitions.h, [5](#)
- changeDimming
  - dimming.cpp, [9](#)
  - dimming.h, [11](#)
- changePWMDuty
  - dimming.cpp, [9](#)
  - dimming.h, [12](#)
- coerceDimming
  - dimming.cpp, [10](#)
  - dimming.h, [12](#)
- definitions.h, [5](#), [6](#)
  - BUTTONUPDATETIME, [5](#)
  - DIMHIGHLIMIT, [5](#)
  - DIMLOWERLIMIT, [5](#)
  - DIMRATE, [5](#)
  - DUTYSAVETIME, [6](#)
  - FIRSTDIMSTEP, [6](#)
  - FIRSTPRESS, [6](#)
  - LONGPRESSACTION, [6](#)
  - LONGPRESSWAIT, [6](#)
- PWMRESOLUTION, [6](#)
- WAITTIMEDIM, [6](#)
- DIMHIGHLIMIT
  - definitions.h, [5](#)
- DIMLOWERLIMIT
  - definitions.h, [5](#)
- dimming.cpp, [8](#)
  - b\_decrease\_Time, [11](#)
  - b\_increase\_Time, [11](#)
  - changeDimming, [9](#)
  - changePWMDuty, [9](#)
  - coerceDimming, [10](#)
  - lumDecrease, [10](#)
  - lumIncrease, [10](#)
  - switchOnOff, [10](#)
- dimming.h, [11](#), [13](#)
  - changeDimming, [11](#)
  - changePWMDuty, [12](#)
  - coerceDimming, [12](#)
  - lumDecrease, [12](#)
  - lumIncrease, [12](#)
  - switchOnOff, [12](#)
- DIMRATE
  - definitions.h, [5](#)
- dutySave
  - main.cpp, [15](#)
- DUTYSAVETIME
  - definitions.h, [6](#)
- FIRSTDIMSTEP
  - definitions.h, [6](#)
- FIRSTPRESS
  - definitions.h, [6](#)
- LONGPRESSACTION
  - definitions.h, [6](#)
- LONGPRESSWAIT
  - definitions.h, [6](#)
- loop
  - main.cpp, [14](#)
- lumDecrease
  - dimming.cpp, [10](#)
  - dimming.h, [12](#)
- lumIncrease
  - dimming.cpp, [10](#)
  - dimming.h, [12](#)
- main.cpp, [13](#)
  - analogDuty, [15](#)
  - b\_decrease, [15](#)

- [b\\_decrease\\_State](#), 15
  - [b\\_increase](#), 15
  - [b\\_increase\\_State](#), 15
  - [b\\_onOff](#), 15
  - [b\\_onOff\\_State](#), 15
  - [buttonReader](#), 15
  - [dutySave](#), 15
  - [loop](#), 14
  - [onOff](#), 16
  - [pref](#), 16
  - [pwmPin](#), 16
  - [saveAnalogDuty](#), 14
  - [setup](#), 14
  - [updateButtons](#), 14
- [onOff](#)
  - [main.cpp](#), 16
  - [variables.h](#), 8
- [pref](#)
  - [main.cpp](#), 16
- [pwmPin](#)
  - [main.cpp](#), 16
- [PWMRESOLUTION](#)
  - [definitions.h](#), 6
- [README](#), 1
- [README.md](#), 8
- [saveAnalogDuty](#)
  - [main.cpp](#), 14
- [setup](#)
  - [main.cpp](#), 14
- [switchOnOff](#)
  - [dimming.cpp](#), 10
  - [dimming.h](#), 12
- [updateButtons](#)
  - [main.cpp](#), 14
- [variables.h](#), 7, 8
  - [analogDuty](#), 7
  - [b\\_decrease\\_State](#), 7
  - [b\\_decrease\\_Time](#), 7
  - [b\\_increase\\_State](#), 7
  - [b\\_increase\\_Time](#), 7
  - [b\\_onOff\\_State](#), 8
  - [onOff](#), 8
- [WAITTIMEDIM](#)
  - [definitions.h](#), 6