# Lampete

Generated by Doxygen 1.10.0

# Chapter 1

# README

This project is for a dimmable lamp build inside a trumpete (there the name is coming from (Trompete means trumpete 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:

# 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.

```
//Definitions for Lampete

#ifndef definitions_H
#define definitions_H


#define FIRSTPRESS 0
#define LONGPRESSWAIT 1
#define LONGPRESSACTION 2

#define WAITTIMEDIM 1000    // wait time for starting the dimming action

#define PWMRESOLUTION   8                                   //in bits
#define DIMRATE         (0.078125*(pow(2, PWMRESOLUTION)))  //per second
#define DIMLOWERLIMIT   (0.0390625*pow(2, PWMRESOLUTION))  //lowest dimming limit
#define DIMHIGHLIMIT    (pow(2, PWMRESOLUTION)-1)           //Highest pwm duty cycle
#define FIRSTDIMSTEP    (0.03125*(pow(2, PWMRESOLUTION)))   //Dimmming step when only pressing the button once


#define BUTTONUPDATETIME    15          // in ms
#define DUTYSAVETIME        30000       // in s
#endif
//EOF
```

## 3.3 variables.h File Reference

```
#include <cstdint>
```

### Variables

- bool onOff

  *Current state of lamp.*
- uint8_t b_onOff_State

  *Current state of state machine for on-off buttton.*
- uint8_t b_decrease_State

  *Current state of state machine for decrease buttton.*
- uint8_t b_increase_State

  *Current state of state machine for increase buttton.*
- 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 buttton.

#### 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 buttton.

#### 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 buttton.

#### 3.3.1.7 onOff

```
bool onOff  [extern]
```

Current state of lamp.

## 3.4 variables.h

Go to the documentation of this file.

```
//Variables for Lampete

#ifndef variables_H
#define variables_H

#include <cstdint>

extern bool onOff;
extern uint8_t b_onOff_State;
extern uint8_t b_decrease_State;
extern uint8_t b_increase_State;
extern uint32_t b_decrease_Time;
extern uint32_t b_increase_Time;
extern int16_t analogDuty;

#endif
//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**

| | |
|---|---|
| *dir* | 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**

| | |
|---|---|
| *duty* | 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**

| | |
|---|---|
| *tempAnalogDuty* | 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**

| | |
|---|---|
| *dir* | 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**

| | |
|---|---|
| *duty* | 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**

| | |
|---|---|
| *tempAnalogDuty* | 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.

```

#ifndef dimming_H
#define dimming_H

#include <cstdint>

void switchOnOff();

void lumIncrease();

void lumDecrease();

void changeDimming(int8_t dir);

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

int16_t coerceDimming(int16_t tempAnalogDuty);


#endif
//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 buttton.*
- uint8_t b_decrease_State = FIRSTPRESS

  *Current state of state machine for decrease buttton.*
- uint8_t b_increase_State = FIRSTPRESS

  *Current state of state machine for increase buttton.*
- 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 buttton.

### 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 buttton.

### 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 buttton.

### 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