Smart Lighting With RGB LED

4

Generated by Doxygen 1.8.13

Contents

Chapter 1

README

Device IoT, which allows you to control the color and brightness of the RGB LED. To manage the LED, the state machine was implemented in which the further development of the project is envisaged. So, the LED is controlled by commands like:

- "c135" - sets the red LED to half power;
- "b55" - sets the brightness of white to a maximum;
- "f" - turns on the mode of smooth color change of the LED.

All management is done through an application in which all these commands are already implemented in a convenient graphical interface. You can also control the LED using the UART interface, which uses bluetooth.

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

File Documentation

3.1 configuration_bits.c File Reference

3.2 main.c File Reference

```
#include <stdint.h>
#include <stdbool.h>
#include "user.h"
```

Functions

- int32_t main (void)

  Initialize I/O and Peripherals for application Setup functionality and port direction.

3.2.1 Function Documentation

3.2.1.1 main()

```
int32_t main (
            void  )
```

Initialize I/O and Peripherals for application Setup functionality and port direction.

Parameters

out	*none*	
in	*none*	

Returns

none

3.3 README.md File Reference

3.4 UART.c File Reference

```
#include "UART.h"
```

Functions

- void uart4_init (void)
- char uart4_getc (void)
- void uart4_putc (char c)
- void uart4_puts (char *s)
- void uart4_test (void)

3.4.1 Function Documentation

3.4.1.1 uart4_getc()

```
char uart4_getc (
             void  )
```

Function prototype:

void uart4_getc(void);

Description:

read char symboll from UART

Parameters:

none

Returns:

none

3.4.1.2 uart4_init()

```
void uart4_init (
             void   )
```

Function prototype:

void uart4_init(void);

Description:

initialize UART

Parameters:

none

Returns:

none

3.4.1.3 uart4_putc()

```
void uart4_putc (
             char c )
```

Function prototype:

void uart4_putc(char c);

Description:

put char symboll to UART

Parameters:

char symbol 'c'

Returns:

none

3.4.1.4 uart4_puts()

```
void uart4_puts (
             char * s )
```

Function prototype:

void uart4_puts(char *s);

Description:

put char array to UART

Parameters:

char array 's'

Returns:

none

3.4.1.5 uart4_test()

```
void uart4_test (
            void )
```

Function prototype:

void uart4_test(void);

Description:

UART test and beginning program

Parameters:

none

Returns:

none

3.5 UART.h File Reference

```
#include <xc.h>
```

Functions

- void uart4_init (void)
- char uart4_getc (void)
- void uart4_putc (char c)
- void uart4_puts (char ∗s)
- void uart4_test (void)

3.5.1 Function Documentation

3.5.1.1 uart4_getc()

```
char uart4_getc (
            void )
```

Function prototype:

void uart4_getc(void);

Description:

read char symboll from UART

Parameters:

none

Returns:

none

3.5.1.2 uart4_init()

```
void uart4_init (
             void  )
```

Function prototype:

void uart4_init(void);

Description:

initialize UART

Parameters:

none

Returns:

none

3.5.1.3 uart4_putc()

```
void uart4_putc (
             char c )
```

Function prototype:

void uart4_putc(char c);

Description:

put char symboll to UART

Parameters:

char symbol 'c'

Returns:

none

3.5.1.4 uart4_puts()

```
void uart4_puts (
             char * s )
```

Function prototype:

void uart4_puts(char *s);

Description:

put char array to UART

Parameters:

char array 's'

Returns:

none

3.5.1.5 uart4_test()

```
void uart4_test (
            void  )
```

Function prototype:

void uart4_test(void);

Description:

UART test and beginning program

Parameters:

none

Returns:

none

3.6 user.c File Reference

```
#include <stdint.h>
#include <stdbool.h>
#include <string.h>
#include "user.h"
#include <sys/attribs.h>
#include "UART.h"
```

Functions

- void init_gpio (void)

  Initialize input output Setup functionality and port direction.
- void InitTimer2AndOC5And4And8 (void)

  Initialize Timer2 Initialize Timer2. Configure OC4,OC5,OC8 control register. Configure PWM for RGB.
- void init_app (void)
- void rgb (int red, int green, int blue)
- void start_program ()

  all functional of programm
- void fade ()
- void delay (uint32_t n)
- void brightness (int bright)

3.6.1 Function Documentation

3.6.1.1 brightness()

```
void brightness (
            int bright )
```

func delay

Parameters

out	*none*	
in	*brigh(PWM)*	

Returns

none

3.6.1.2 delay()

```
void delay (
            uint32_t n )
```

func delay

Parameters

out	*none*	
in	*n*	- time in mills

Returns

none

3.6.1.3 fade()

```
void fade ( )
```

fade mode

Parameters

out	*none*	
in	*none*	

Returns

none

3.6.1.4 init_app()

```
void init_app (
            void  )
```

begins program

Parameters

out	*none*	
in	*none*	

Returns

none

3.6.1.5 init_gpio()

```
void init_gpio (
            void  )
```

Initialize input output Setup functionality and port direction.

Parameters

out	*none*	
in	*none*	

Returns

none

3.6.1.6 InitTimer2AndOC5And4And8()

```
void InitTimer2AndOC5And4And8 (
            void  )
```

Initialize Timer2 Initialize Timer2. Configure OC4,OC5,OC8 control register. Configure PWM for RGB.

Parameters

out	*none*	
in	*none*	

Returns

none

3.6.1.7 rgb()

```
void rgb (
            int red,
            int green,
            int blue )
```

func for control rgb led(set color)

Parameters

out	*none*	
in	*red,green,blue*	(PwM)

Returns

none

3.6.1.8 start_program()

```
void start_program ( )
```

all functional of programm

Functional: command a - turn on RGB led. command f - fade mode command c - set color command b - set brightness

Parameters

out	*none*	
in	*none*	

Returns

none

3.7 user.h File Reference

```
#include <stdint.h>
```

Macros

- #define LD1_PORT_BIT LATGbits.LATG6
- #define LD2_PORT_BIT LATDbits.LATD4
- #define LD3_PORT_BIT LATBbits.LATB11
- #define LD4_PORT_BIT LATGbits.LATG15
- #define BTN1_PORT_BIT PORTAbits.RA5
- #define BTN2_PORT_BIT PORTAbits.RA4
- #define PWM_FREQ_HZ (1000)
- #define PWM_PERIOD_COUNTS (100000000/(256∗PWM_FREQ_HZ))
- #define MAX_ADC_VALUE (4095)

Functions

- void init_app (void)

3.7.1 Macro Definition Documentation

3.7.1.1 BTN1_PORT_BIT

```
#define BTN1_PORT_BIT PORTAbits.RA5
```

3.7.1.2 BTN2_PORT_BIT

```
#define BTN2_PORT_BIT PORTAbits.RA4
```

3.7.1.3 LD1_PORT_BIT

```
#define LD1_PORT_BIT LATGbits.LATG6
```

3.7.1.4 LD2_PORT_BIT

```
#define LD2_PORT_BIT LATDbits.LATD4
```

3.7.1.5 LD3_PORT_BIT

```
#define LD3_PORT_BIT LATBbits.LATB11
```

3.7.1.6 LD4_PORT_BIT

```
#define LD4_PORT_BIT LATGbits.LATG15
```

3.7.1.7 MAX_ADC_VALUE

```
#define MAX_ADC_VALUE (4095)
```

3.7.1.8 PWM_FREQ_HZ

```
#define PWM_FREQ_HZ (1000)
```

3.7.1.9 PWM_PERIOD_COUNTS

```
#define PWM_PERIOD_COUNTS (100000000/(256*PWM_FREQ_HZ))
```

3.7.2 Function Documentation

3.7.2.1 init_app()

```
void init_app (
             void  )
```

begins program

Parameters

out	*none*	
in	*none*	

Returns

none

Index