lab_4 UART and SPI

Generated by Doxygen 1.8.13

Contents

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

Chapter 2

File Documentation

2.1 src/configuration_bits.c File Reference

2.2 src/main.c File Reference

```
#include "scrolling_text.h"
#include "OLED/OLED.h"
#include <string.h>
```

Include dependency graph for main.c:

2.3 src/scrolling_text.c File Reference

initialization and system functions, inter

```
#include "scrolling_text.h"
#include <string.h>
#include <xc.h>
#include <sys/attribs.h>
#include "uart.h"
#include "OLED/OLED.h"
```

Include dependency graph for scrolling_text.c:

Functions

- void DelayMs (int t)
- void init_gpio (void)

  GPIO initialization, Initialization for buttons and LEDs Disabling analog mode and setting pins directions.
- void init_app (void)

  App initialization, Calling initialization functions for UART, GPIO and OLED Basic OLED configuration.
- void shift_str (uint8_t ∗str)

  Circular shift of array to the left.
- void __ISR (_CHANGE_NOTICE_A_VECTOR, IPL2SOFT)

  Buttons interrupt handler, Changes program state when button is pressed.

Variables

- uint32_t delay_g = 200

2.3.1 Detailed Description

initialization and system functions, inter

2.3.2 Function Documentation

2.3.2.1 __ISR()

```
void __ISR (
             _CHANGE_NOTICE_A_VECTOR ,
             IPL2SOFT  )
```

Buttons interrupt handler, Changes program state when button is pressed.

2.3.2.2 DelayMs()

```
void DelayMs (
             int t )
```

2.3.2.3 init_app()

```
void init_app (
             void  )
```

App initialization, Calling initialization functions for UART, GPIO and OLED Basic OLED configuration.

2.3.2.4 init_gpio()

```
void init_gpio (
             void  )
```

GPIO initialization, Initialization for buttons and LEDs Disabling analog mode and setting pins directions.

2.3.2.5 shift_str()

```
void shift_str (
             uint8_t * str )
```

Circular shift of array to the left.

Parameters

in	*array*	

2.3.3 Variable Documentation

2.3.3.1 delay_g

```
uint32_t delay_g = 200
```

2.4 src/scrolling_text.h File Reference

contains definitions, macroses and function prototypes

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

Include dependency graph for scrolling_text.h: This graph shows which files directly or indirectly include this file:

Macros

- #define LED_1 LATGbits.LATG6
- #define LED_2 LATDbits.LATD4
- #define LED_3 LATBbits.LATB11
- #define LED_4 LATGbits.LATG15
- #define BTN_1 PORTAbits.RA5
- #define BTN_2 PORTAbits.RA4
- #define BTN_4_SCHLD PORTAbits.RA2
- #define BTN_3_SCHLD PORTFbits.RF1
- #define BTN_2_SCHLD PORTDbits.RD5
- #define BTN_1_SCHLD PORTAbits.RA3

Typedefs

- typedef enum states STATES

Enumerations

- enum states { RESET = 0, START, PAUSE }

Functions

- void DelayMs (int t)
- void delay (volatile uint32_t val)
- void init_app (void)

  App initialization, Calling initialization functions for UART, GPIO and OLED Basic OLED configuration.
- void shift_str (uint8_t *str)

  Circular shift of array to the left.

Variables

- volatile uint32_t cur_state_g
- uint8_t msg_g [120] = "test msg "
- uint32_t delay_g

2.4.1 Detailed Description

contains definitions, macroses and function prototypes

2.4.2 Macro Definition Documentation

2.4.2.1 BTN_1

```
#define BTN_1 PORTAbits.RA5
```

2.4.2.2 BTN_1_SCHLD

```
#define BTN_1_SCHLD PORTAbits.RA3
```

2.4.2.3 BTN_2

```
#define BTN_2 PORTAbits.RA4
```

2.4.2.4 BTN_2_SCHLD

```
#define BTN_2_SCHLD PORTDbits.RD5
```

2.4.2.5 BTN_3_SCHLD

```
#define BTN_3_SCHLD PORTFbits.RF1
```

2.4.2.6 BTN_4_SCHLD

```
#define BTN_4_SCHLD PORTAbits.RA2
```

2.4.2.7 LED_1

```
#define LED_1 LATGbits.LATG6
```

2.4.2.8 LED_2

```
#define LED_2 LATDbits.LATD4
```

2.4.2.9 LED_3

```
#define LED_3 LATBbits.LATB11
```

2.4.2.10 LED_4

```
#define LED_4 LATGbits.LATG15
```

2.4.3 Typedef Documentation

2.4.3.1 STATES

```
typedef enum states STATES
```

2.4.4 Enumeration Type Documentation

2.4.4.1 states

```
enum states
```

Enumerator

RESET	
START	
PAUSE	

2.4.5 Function Documentation

2.4.5.1 delay()

```
void delay (
            volatile uint32_t val )
```

2.4.5.2 DelayMs()

```
void DelayMs (
            int t )
```

2.4.5.3 init_app()

```
void init_app (
            void  )
```

App initialization, Calling initialization functions for UART, GPIO and OLED Basic OLED configuration.

2.4.5.4 shift_str()

```
void shift_str (
            uint8_t * str )
```

Circular shift of array to the left.

Parameters

in	*array*	

2.4.6 Variable Documentation

2.4.6.1 cur_state_g

```
volatile uint32_t cur_state_g
```

2.4.6.2 delay_g

```
uint32_t delay_g
```

2.4.6.3 msg_g

```
uint8_t msg_g[120] = "test msg "
```

2.5 src/uart.c File Reference

```
#include "uart.h"
```
Include dependency graph for uart.c:

Functions

- void UART4_init (void)

  UART initialization.
- char UART4_getc (void)

  Reading char from UART.
- void UART4_putc (char c)

  Sending char by UART.
- void UART4_puts (char ∗s)

  Sending string by UART.

2.5.1 Function Documentation

2.5.1.1 UART4_getc()

```
char UART4_getc (
            void  )
```

Reading char from UART.

It waits until character received and returns it

Returns

U4RXREG readed character

2.5.1.2 UART4_init()

```
void UART4_init (
            void  )
```

UART initialization.

Setting pins for UART usage and UART speed

2.5.1.3 UART4_putc()

```
void UART4_putc (
            char c )
```

Sending char by UART.

It waits until transmit buffer not full and transmits char

Parameters

in	*c*	char to send

2.5.1.4 UART4_puts()

```
void UART4_puts (
            char * s )
```

Sending string by UART.

It sends chars from string while it's not end symbol

Parameters

in	*s*	pointer for string to send

2.6 src/uart.h File Reference

`#include <xc.h>`
Include dependency graph for uart.h: This graph shows which files directly or indirectly include this file:

Functions

- void UART4_init (void)
  UART initialization.
- char UART4_getc (void)
  Reading char from UART.
- void UART4_putc (char c)
  Sending char by UART.
- void UART4_puts (char *s)
  Sending string by UART.

2.6.1 Function Documentation

2.6.1.1 UART4_getc()

```
char UART4_getc (
            void  )
```

Reading char from UART.

It waits until character received and returns it

Returns

U4RXREG readed character

2.6.1.2 UART4_init()

```
void UART4_init (
            void  )
```

UART initialization.

Setting pins for UART usage and UART speed

2.6.1.3 UART4_putc()

```
void UART4_putc (
            char c )
```

Sending char by UART.

It waits until transmit buffer not full and transmits char

Parameters

in	*c*	char to send

2.6.1.4 UART4_puts()

```
void UART4_puts (
            char * s )
```

Sending string by UART.

It sends chars from string while it's not end symbol

Parameters

in	*s*	pointer for string to send

Index