lab_3 PWM and ADC

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/ADC.c File Reference

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

Include dependency graph for ADC.c:

2.2 src/ADC.h File Reference

This graph shows which files directly or indirectly include this file:

Functions

- void initWiFIREadc (void)
- int convertWiFIREadc (uint8_t channelNumber)
- int ReadPotentiometerWithADC (void)

2.2.1 Function Documentation

2.2.1.1 convertWiFIREadc()

```
int convertWiFIREadc (
            uint8_t channelNumber )
```

2.2.1.2 initWiFIREadc()

```
void initWiFIREadc (
            void  )
```

2.2.1.3 ReadPotentiometerWithADC()

```
int ReadPotentiometerWithADC (
            void  )
```

2.3 src/configuration_bits.c File Reference

2.4 src/main.c File Reference

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

Functions

- void main (void)

2.4.1 Function Documentation

2.4.1.1 main()

```
void main (
            void  )
```

2.5 src/morse_code.c File Reference

contains descroption for initialization and system functions

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

Functions

- void morse_code_encoding (const uint8_t msg[ ], uint8_t ∗res)

  Converting text to morse code.
- void init_timer (void)

  OC8 timer initialization to work in pwm mode.
- void init_gpio (void)

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

  App initialization, Calling all initialization functions.
- void delay (volatile uint32_t val)
- void blink_leds (uint32_t mode)

  Setting LEDs on and off,.
- void display_msg (void)

  Displaying converted msgs on LEDs,.
- void __ISR (_CHANGE_NOTICE_A_VECTOR, IPL2SOFT)

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

2.5.1 Detailed Description

contains descroption for initialization and system functions

2.5.2 Function Documentation

2.5.2.1 __ISR()

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

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

2.5.2.2 blink_leds()

```
void blink_leds (
             uint32_t mode )
```

Setting LEDs on and off,.

Parameters

in	*mode*	values: 1 - blink with one led, 3 - 2 leds

2.5.2.3 delay()

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

2.5.2.4 display_msg()

```
void display_msg (
            void  )
```

Displaying converted msgs on LEDs,.

2.5.2.5 init_app()

```
void init_app (
            void  )
```

App initialization, Calling all initialization functions.

2.5.2.6 init_gpio()

```
void init_gpio (
            void  )
```

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

2.5.2.7 init_timer()

```
void init_timer (
            void  )
```

OC8 timer initialization to work in pwm mode.

2.5.2.8 morse_code_encoding()

```
void morse_code_encoding (
            const uint8_t msg[],
            uint8_t * res )
```

Converting text to morse code.

Parameters

in	*msg*	text message to be converted to morse code
in	*res*	array for storing converted msg

2.6 src/morse_code.h File Reference

contains definitions, macroses and function prototypes

```
#include <stdint.h>
#include <string.h>
#include <sys/attribs.h>
#include "ADC.h"
```

Include dependency graph for morse_code.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 BLINK_DELAY 1400
- #define DEBOUNCE_DELAY 10
- #define PWM_FREQ_HZ (1000)
- #define PWM_PERIOD_COUNTS (100000000/(256∗PWM_FREQ_HZ))
- #define MAX_ADC_VALUE (4095)
- #define VR1_AN_CHAN_NUM (8)

Typedefs

- typedef enum states STATES

Enumerations

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

Functions

- void delay (volatile uint32_t val)
- void init_app (void)

  App initialization, Calling all initialization functions.
- void blink_leds (uint32_t mode)

  Setting LEDs on and off,.
- void display_msg (void)

  Displaying converted msgs on LEDs,.
- void morse_code_ecoding (const uint8_t msg[ ], uint8_t ∗res)

Variables

- const uint8_t MSG [ ] = "test msg"
- uint8_t encoded_msg_g [100]
- volatile uint32_t cur_state_g
- volatile uint32_t cur_delay_g

2.6.1 Detailed Description

contains definitions, macroses and function prototypes

2.6.2 Macro Definition Documentation

2.6.2.1 BLINK_DELAY

```
#define BLINK_DELAY 1400
```

2.6.2.2 BTN_1

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

2.6.2.3 BTN_2

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

2.6.2.4 DEBOUNCE_DELAY

```
#define DEBOUNCE_DELAY 10
```

2.6.2.5 LED_1

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

2.6.2.6 LED_2

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

2.6.2.7 LED_3

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

2.6.2.8 LED_4

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

2.6.2.9 MAX_ADC_VALUE

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

2.6.2.10 PWM_FREQ_HZ

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

2.6.2.11 PWM_PERIOD_COUNTS

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

2.6.2.12 VR1_AN_CHAN_NUM

```
#define VR1_AN_CHAN_NUM (8)
```

2.6.3 Typedef Documentation

2.6.3.1 STATES

```
typedef enum states STATES
```

2.6.4 Enumeration Type Documentation

2.6.4.1 states

```
enum states
```

Enumerator

RESET	
START	
PAUSE	

2.6.5 Function Documentation

2.6.5.1 blink_leds()

```
void blink_leds (
            uint32_t mode )
```

Setting LEDs on and off,.

Parameters

in	*mode*	values: 1 - blink with one led, 3 - 2 leds

2.6.5.2 delay()

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

2.6.5.3 display_msg()

```
void display_msg (
            void  )
```

Displaying converted msgs on LEDs,.

2.6.5.4 init_app()

```
void init_app (
            void  )
```

App initialization, Calling all initialization functions.

2.6.5.5 morse_code_ecoding()

```
void morse_code_ecoding (
            const uint8_t msg[],
            uint8_t * res )
```

2.6.6 Variable Documentation

2.6.6.1 cur_delay_g

```
volatile uint32_t cur_delay_g
```

2.6.6.2 cur_state_g

```
volatile uint32_t cur_state_g
```

2.6.6.3 encoded_msg_g

```
uint8_t encoded_msg_g[100]
```

2.6.6.4 MSG

```
const uint8_t MSG[] = "test msg"
```

Index