ELECTRIC LOCK

Generated by Doxygen 1.8.16

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

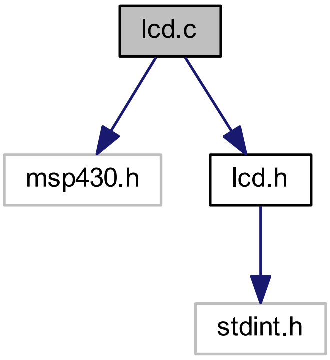
Chapter 2

File Documentation

2.1 lcd.c File Reference

```
#include <msp430.h>
#include <lcd.h>
```

Include dependency graph for lcd.c:



Macros

- #define LCD_PORT P8OUT
- #define LCD_PORT_DIR P8DIR
- #define LCD_EN BIT3
- #define LCD_RS BIT2
- #define MASK_UPPER 0xF0
- #define MASK_FULL 0xFF
- #define MODE_FONT 0x28
- #define CURS_BLINK 0x0C
- #define SHIFTR 0x06
- #define DDRAM 0x80
- #define CLEAR_SCREEN 0x01
- #define ASCII_CONV 0x30

Functions

- void lcd_reset ()

  LCD reset.
- void lcd_command (char cmd)

  LCD command.
- void lcd_begin ()

  Restart the beginning phase.
- void lcd_initialization ()

  LCD initialization.
- void lcd_display (unsigned char dat)

  Displays data.
- void lcd_text (char ∗text, int pos)

  Text display.

2.1.1 Macro Definition Documentation

2.1.1.1 ASCII_CONV

```
#define ASCII_CONV 0x30
```

ASCII code for zero

Definition at line 37 of file lcd.c.

2.1.1.2 CLEAR_SCREEN

```
#define CLEAR_SCREEN 0x01
```

Clear LCD screen

Definition at line 35 of file lcd.c.

2.1.1.3 CURS_BLINK

```
#define CURS_BLINK 0x0C
```

LCD cursor no blink

Definition at line 29 of file lcd.c.

2.1.1.4 DDRAM

```
#define DDRAM 0x80
```

Forcing cursor at the beginning of the first line command

Definition at line 33 of file lcd.c.

2.1.1.5 LCD_EN

```
#define LCD_EN BIT3
```

LCD enable pin

Definition at line 19 of file lcd.c.

2.1.1.6 LCD_PORT

```
#define LCD_PORT P8OUT
```

LCD port output

Definition at line 14 of file lcd.c.

2.1.1.7 LCD_PORT_DIR

```
#define LCD_PORT_DIR P8DIR
```

LCD port direction

Definition at line 16 of file lcd.c.

2.1.1.8 LCD_RS

```
#define LCD_RS BIT2
```

LCD register select pin

Definition at line 21 of file lcd.c.

2.1.1.9 MASK_FULL

```
#define MASK_FULL 0xFF
```

Mask all eight bits

Definition at line 25 of file lcd.c.

2.1.1.10 MASK_UPPER

```
#define MASK_UPPER 0xF0
```

Mask upper four bits

Definition at line 23 of file lcd.c.

2.1.1.11 MODE_FONT

```
#define MODE_FONT 0x28
```

4 bit mode 5x7 font LCD

Definition at line 27 of file lcd.c.

2.1.1.12 SHIFTR

```
#define SHIFTR 0x06
```

Automatic increment no display shift LCD mode command

Definition at line 31 of file lcd.c.

2.1.2 Function Documentation

2.1.2.1 lcd_begin()

```
void lcd_begin ( )
```

Restart the beginning phase.

Sets the display in the "beginning" state where all digits are 0 and cursor is set at the first digit position.

Definition at line 91 of file lcd.c.

References ASCII_CONV, DDRAM, lcd_command(), lcd_display(), and lcd_text().

Referenced by lcd_initialization(), and Timer_A_CCR0_ISR().

2.1.2.2 lcd_command()

```
void lcd_command (
             char cmd )
```

LCD command.

Sends the command to LCD port in the 4-bit mode with sending the upper nibble of the command first.

Parameters

cmd	The hex command to be sent

Definition at line 73 of file lcd.c.

References LCD_EN, LCD_PORT, and MASK_UPPER.

Referenced by ADC12ISR(), lcd_begin(), lcd_initialization(), lcd_text(), Timer_A_CCR0_ISR(), and turn_on_led().

2.1.2.3 lcd_display()

```
void lcd_display (
            unsigned char dat )
```

Displays data.

Displays data on the LCD screen using the 4-bit mode. Digits are limited to number 9. In other words, password input is processed per module 10. Upper nibble of a character is sent first.

Parameters

dat	Character to be displayed

Definition at line 130 of file lcd.c.

References LCD_EN, LCD_PORT, LCD_RS, and MASK_UPPER.

Referenced by ADC12ISR(), lcd_begin(), and lcd_text().

2.1.2.4 lcd_initialization()

```
void lcd_initialization ( )
```

LCD initialization.

Initializes the LCD by setting the display mode, cursor and setting the initial state of the LCD.

Definition at line 110 of file lcd.c.

References CLEAR_SCREEN, CURS_BLINK, lcd_begin(), lcd_command(), lcd_reset(), MODE_FONT, and SHI↩ FTR.

Referenced by main().

2.1.2.5 lcd_reset()

```
void lcd_reset ( )
```

LCD reset.

Resets the LCD with respect to execution time. After executing this function LCD is ready for further use.

Definition at line 46 of file lcd.c.

References LCD_EN, LCD_PORT, LCD_PORT_DIR, and MASK_FULL.

Referenced by lcd_initialization().

2.1.2.6 lcd_text()

```
void lcd_text (
             char * text,
             int pos )
```

Text display.

Displays specific text on the screen starting from the specified position.

Parameters

text	Text to be displayed
pos	Starting position for the text

Definition at line 154 of file lcd.c.

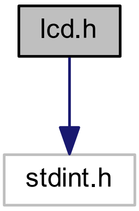
References lcd_command(), and lcd_display().

Referenced by lcd_begin(), and turn_on_led().

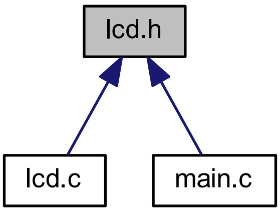
2.2 lcd.h File Reference

```
#include "stdint.h"
```

Include dependency graph for lcd.h:



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



Functions

- void lcd_reset ()

  LCD reset.
- void lcd_command (char cmd)

  LCD command.
- void lcd_begin ()

  Restart the beginning phase.
- void lcd_initialization ()

  LCD initialization.
- void lcd_display (unsigned char dat)

  Displays data.
- void lcd_text (char ∗text, int pos)

  Text display.

2.2.1 Function Documentation

2.2.1.1 lcd_begin()

```
void lcd_begin ( )
```

Restart the beginning phase.

Sets the display to start mode with 4 zeros

Sets the display in the "beginning" state where all digits are 0 and cursor is set at the first digit position.

Definition at line 91 of file lcd.c.

References ASCII_CONV, DDRAM, lcd_command(), lcd_display(), and lcd_text().

Referenced by lcd_initialization(), and Timer_A_CCR0_ISR().

2.2.1.2 lcd_command()

```
void lcd_command (
             char cmd )
```

LCD command.

Sending the command cmd to LCD display

Sends the command to LCD port in the 4-bit mode with sending the upper nibble of the command first.

Parameters

cmd	The hex command to be sent

Definition at line 73 of file lcd.c.

References LCD_EN, LCD_PORT, and MASK_UPPER.

Referenced by ADC12ISR(), lcd_begin(), lcd_initialization(), lcd_text(), Timer_A_CCR0_ISR(), and turn_on_led().

2.2.1.3 lcd_display()

```
void lcd_display (
             unsigned char dat )
```

Displays data.

Printing out the data

Displays data on the LCD screen using the 4-bit mode. Digits are limited to number 9. In other words, password input is processed per module 10. Upper nibble of a character is sent first.

Parameters

dat	Character to be displayed

Definition at line 130 of file lcd.c.

References LCD_EN, LCD_PORT, LCD_RS, and MASK_UPPER.

Referenced by ADC12ISR(), lcd_begin(), and lcd_text().

2.2.1.4 lcd_initialization()

```
void lcd_initialization ( )
```

LCD initialization.

LCD display initialization

Initializes the LCD by setting the display mode, cursor and setting the initial state of the LCD.

Definition at line 110 of file lcd.c.

References CLEAR_SCREEN, CURS_BLINK, lcd_begin(), lcd_command(), lcd_reset(), MODE_FONT, and SHI←
FTR.

Referenced by main().

2.2.1.5 lcd_reset()

```
void lcd_reset ( )
```

LCD reset.

Resets the LCD

Resets the LCD with respect to execution time. After executing this function LCD is ready for further use.

Definition at line 46 of file lcd.c.

References LCD_EN, LCD_PORT, LCD_PORT_DIR, and MASK_FULL.

Referenced by lcd_initialization().

2.2.1.6 lcd_text()

```
void lcd_text (
             char * text,
             int pos )
```

Text display.

Displays specific text starting from pos

Displays specific text on the screen starting from the specified position.

Parameters

text	Text to be displayed
pos	Starting position for the text

Definition at line 154 of file lcd.c.

References lcd_command(), and lcd_display().

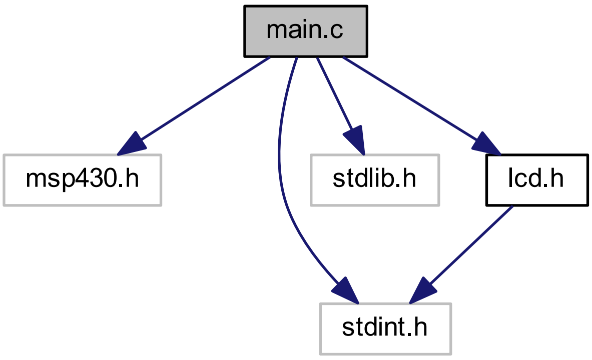
Referenced by lcd_begin(), and turn_on_led().

2.3 main.c File Reference

Electric lock implementation.

```
#include <msp430.h>
#include <stdint.h>
#include <stdlib.h>
#include "lcd.h"
```

Include dependency graph for main.c:



Macros

- #define LED_PORT P4OUT
- #define LED_PORT_DIR P4DIR
- #define BUTTON_PORT_DIR P2DIR
- #define BUTTON_INT_EN P2IE
- #define BUTTON_FLAG P2IFG
- #define BUTTON_EDGE_SEL P2IES
- #define ONE_SECOND (32768-1)
- #define ASCII_CONV 0x30

Functions

- void turn_on_led ()

  Turning on LEDs.
- void check_password ()

  Password check.
- int main (void)

  Main program.
- __interrupt void ADC12ISR (void)

  Updating the current digit using A/D conversion.
- __interrupt void Timer_A_CCR0_ISR (void)

  One second elapsed ISR.

Variables

- volatile uint8_t seconds =0
- volatile uint8_t password [4] = {1,2,3,4}
- volatile uint8_t guess [4] = {0,0,0,0}
- volatile uint8_t data_cnt = 0
- volatile int8_t pressed = 0
- volatile uint8_t correct = 1
- volatile uint8_t conversion_allowed = 1

2.3.1 Detailed Description

Electric lock implementation.

The programs is intended for simulating an electric lock functionality. In order to unlock the lock we need to input the right 4-digit password. The password is predefined in the program, and along with that it cannot be changed. The program starts with the first digit selected on the LCD. Using potentiometer we can change the value of the first digit, and confirm our selection using the first button on the left. The process is similar for remaining three digits. Finally we submit our selection for the fourth digit we get a feedback which indicates if we got the password right. First LED indicates success while the other indicates failure. Both LEDs are on for three seconds.

Date

Dec 2019

Author

Senad Kurtisi

2.3.2 Macro Definition Documentation

2.3.2.1 ASCII_CONV

```
#define ASCII_CONV 0x30
```

ASCII representation for zero

Definition at line 42 of file main.c.

2.3.2.2 BUTTON_EDGE_SEL

```
#define BUTTON_EDGE_SEL P2IES
```

Button edge select

Definition at line 37 of file main.c.

2.3.2.3 BUTTON_FLAG

```
#define BUTTON_FLAG P2IFG
```

Button flag

Definition at line 35 of file main.c.

2.3.2.4 BUTTON_INT_EN

```
#define BUTTON_INT_EN P2IE
```

Button interrupt enable

Definition at line 33 of file main.c.

2.3.2.5 BUTTON_PORT_DIR

```
#define BUTTON_PORT_DIR P2DIR
```

Button port direction

Definition at line 31 of file main.c.

2.3.2.6 LED_PORT

```
#define LED_PORT P4OUT
```

LED port output

Definition at line 26 of file main.c.

2.3.2.7 LED_PORT_DIR

```
#define LED_PORT_DIR P4DIR
```

LED port direction

Definition at line 28 of file main.c.

2.3.2.8 ONE_SECOND

```
#define ONE_SECOND (32768-1)
```

One second calculated with the ACLK

Definition at line 40 of file main.c.

2.3.3 Function Documentation

2.3.3.1 ADC12ISR()

```
__interrupt void ADC12ISR (
             void  )
```

Updating the current digit using A/D conversion.

Reads the change using ADC interrupt vector and along with that updates current digit on the LCD. Only the high nibble is used per module 10.

Definition at line 185 of file main.c.

References ASCII_CONV, conversion_allowed, data_cnt, lcd_command(), lcd_display(), and LED_PORT.

2.3.3.2 check_password()

```
void check_password ( )
```

Password check.

Checks password guess validity by comparing the master password and the password guess.

Definition at line 100 of file main.c.

References correct, data_cnt, guess, password, and turn_on_led().

Referenced by main().

2.3.3.3 main()

```
int main (
            void  )
```

Main program.

Definition at line 122 of file main.c.

References BUTTON_EDGE_SEL, BUTTON_FLAG, BUTTON_INT_EN, BUTTON_PORT_DIR, check_password(), conversion_allowed, data_cnt, guess, lcd_initialization(), LED_PORT, LED_PORT_DIR, and pressed.

2.3.3.4 Timer_A_CCR0_ISR()

```
__interrupt void Timer_A_CCR0_ISR (
            void  )
```

One second elapsed ISR.

Counts the number of seconds that have passed since the adequate LED has been turned on. If 3 seconds have passed, LED is turned off and the number of seconds is reset. ISR is unavailable until next password guess validation.

Definition at line 211 of file main.c.

References conversion_allowed, correct, lcd_begin(), lcd_command(), LED_PORT, and seconds.

2.3.3.5 turn_on_led()

```
void turn_on_led ( )
```

Turning on LEDs.

Turns on the adequate LED with respect to the validity of the password guess.

Definition at line 65 of file main.c.

References correct, lcd_command(), lcd_text(), LED_PORT, and ONE_SECOND.

Referenced by check_password().

2.3.4 Variable Documentation

2.3.4.1 conversion_allowed

```
volatile uint8_t conversion_allowed = 1
```

Conversion allowed condition

Definition at line 56 of file main.c.

Referenced by ADC12ISR(), main(), and Timer_A_CCR0_ISR().

2.3.4.2 correct

```
volatile uint8_t correct = 1
```

Correct guess indicator

Definition at line 54 of file main.c.

Referenced by check_password(), Timer_A_CCR0_ISR(), and turn_on_led().

2.3.4.3 data_cnt

```
volatile uint8_t data_cnt = 0
```

Password guess counter

Definition at line 50 of file main.c.

Referenced by ADC12ISR(), check_password(), and main().

2.3.4.4 guess

```
volatile uint8_t guess[4] = {0,0,0,0}
```

Password guess

Definition at line 48 of file main.c.

Referenced by check_password(), and main().

2.3.4.5 password

```
volatile uint8_t password[4] = {1,2,3,4}
```

Lock password

Definition at line 46 of file main.c.

Referenced by check_password().

2.3.4.6 pressed

```
volatile int8_t pressed = 0
```

Button press indicator

Definition at line 52 of file main.c.

Referenced by main().

2.3.4.7 seconds

```
volatile uint8_t seconds =0
```

Counter of seconds

Definition at line 44 of file main.c.

Referenced by Timer_A_CCR0_ISR().

Index