LCD DRIVER WITH AWU

MCD APPLICATION LAB
Version 1.0
24/01/08

Table of Contents

lcd Module Index

lcd Module Documentation

Here is a list of all modules:

lcd File Index

lcd File Documentation

Here is a list of all documented files with brief descriptions:

lcd Module Documentation

CLK:

Functions

- void **CLK_Init** (void)
- void **CLK_Switch** (char clock, char previous_clock)

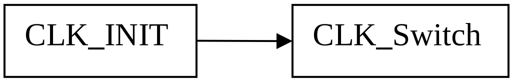
Function Documentation

void CLK_Init (void)

All Clock Controller settings have to be put in this function clock master, clock switch if any, HSI prescaler, CCO, clock gating system... No parameter
Definition of the function at line 35 of file clk.c:

⇒ References CLK_Switch()
⇒ Referenced by main()

Here is the call graph for this function:



void CLK_Switch (char *clock*, char *previous_clock*)

Allow to switch from one clock to another and to switch off the previous one or not.
Clock to switch ON (value put in CLK_SWR), values available in CLK_reg.h if previous clock has to be kept: write 0 as parameter otherwise take predefined value in CLK_reg.h (value put in CLK_ICKR or CLK_ECKR)
Definition of the function at line 52 of file clk.c:

⇒ Referenced by CLK_Init().

Io_port :

Functions

- void **IOport_Init** (void)
- void **IO_Toggle1** (void)
- void **IO_Toggle2** (void)

Function Documentation

void IOport_Init (void)

IOport_Init: Initialization of the I/Os (COMs + SEGs) + 2 externals ITs on PA4/PA5. No parameter
Definition of the file at line 45 of file io_port.c:

⇒ Referenced by LCD_On()

void IO_Toggle1 (void)

Applies once Vdd to segments which have to be ON and Vss to segments which have to be OFF with COM line of this phase set to low level and other set at Vdd/2, and once voltage inverted to the ones applied previously on segments with COM line of this phase set to high level and other set at Vdd/2; If clock master is LSI, there is a switch on HSI during the interrupt to speed up the processing. No parameter
Definition at line 135 of file io_port.c

⇒ References first_CC1, index, and LCDRAM
⇒ Referenced by AWU_It()

void IO_Toggle2 (void)

As it is difficult to implement the contrast control, COM and SEG lines are set low, to have a fixed “dead time” and so the contrast can not be changed. If clock master is LSI, there is a switch on HSI during the interrupt to speed up the processing. No parameter
Definition at line 251 of file io_port.c

- ⇒ References index
- ⇒ Referenced by AWU_It()

LCD

Functions

- void **Clr_LCDRAM** (void)
- void **LCD_On** (void)
- void **Convert** (char *c)
- void **Write_char** (char car, u8 pos)
- void **LCD_Display** (char *str)

Function Documentation

void Clr_LCDRAM (void)

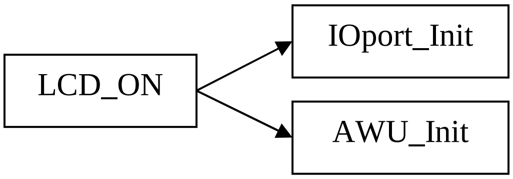
Clear the LCD RAM. No parameter
Definition at line 106 of file lcd.c:

void LCD_On (void)

Switch LCD ON (takes AWU and I/Os ressources), enable the interrupts. No parameter
Definition at line 117 of file lcd.c:

- ⇒ References IOport_Init() and AWU_Init().
- ⇒ Referenced by main()

Here is the call graph for this function:



void Convert (char * *c*)

Convert an ASCII char to a LCD coding contained in letter or number table.
`Character` Definition at line 126 of file lcd.c:

- ⇒ References digit, letter, and number

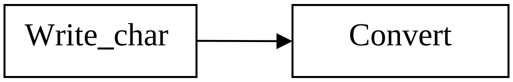
void Write_char (char *car*, u8 *pos*)

Write a char in the LCD frame buffer. ASCII value of the character LCDRAM index [0:3] where to write the LCD coding value.
Definition at line 151 of file lcd.c:

⇒ References Convert(), digit, and LCDRAM.

⇒ Referenced by LCD_Display().

Here is the call graph for this function:



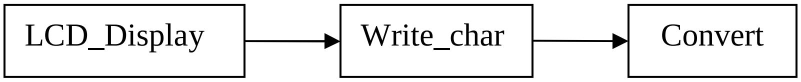
void LCD_Display (char * *str*):

Allow to display a string of 8 characters on the LCD, this function only has to be called from the main. String of 8 characters to display.
Definition at line 197 of file lcd.c:

⇒ References Write_char().

⇒ Referenced by main()

Here is the call graph for this function



Main

Functions

- void **main** (void)
- interrupt void **dummy** ()

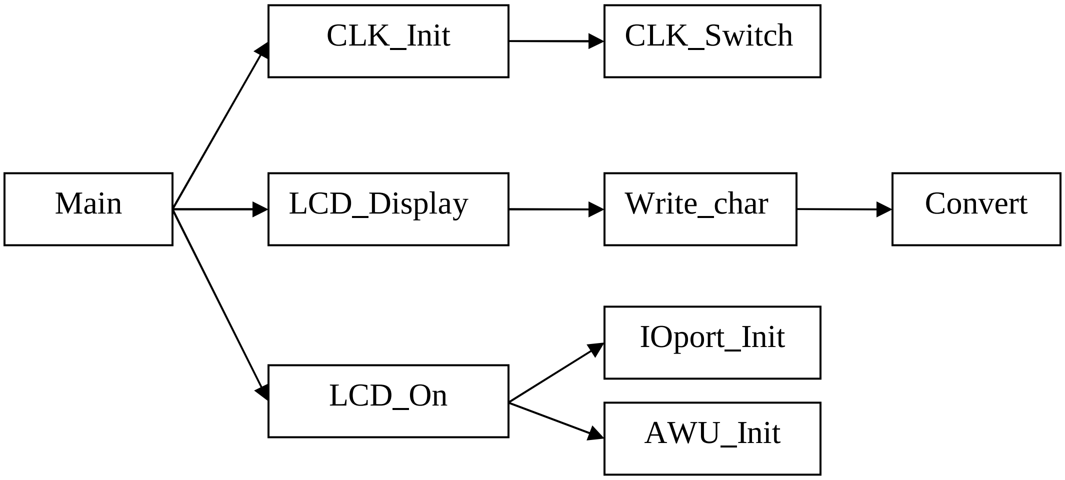
Function Documentation

void main (void)

Main routine. No parameter
Definition at line 37 of file main.c:

⇒ References CLK_Init(), LCD_Display(), and LCD_On().

Here is the call graph for this function:



AWU

Functions

- void **AWU_Init** (void)
- interrupt void **AWU_It** (void)

Function Documentation

void AWU_Init (void) :

Enable AWU and set the AWU timebase.
Definition at line 37 of file awu.c.

⇒ Referenced by LCD_On().

interrupt void AWU_It (void):

AWU Interrupt function: Enable IO_Toggle1 and then IO_Toggle2. No parameter.
Definition at line 45 of file awu.c:

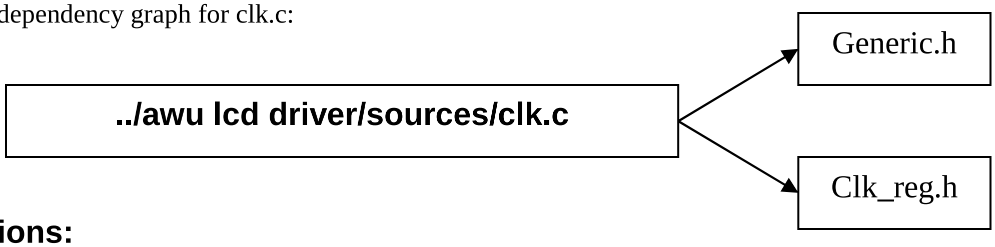
⇒ References IO_Toggle1(), IO_Toggle2() and first.

lcd File Documentation

../awu lcd driver/sources/clk.c File Reference

```
#include "clk_reg.h"
#include "generic.h"
```

Include dependency graph for clk.c:



Functions:

- void **CLK_Init** (void)
- void **CLK_Switch** (char clock, char previous_clock)

Detailed Description:

Clock Controller settings
Definition in file **clk.c**

../awu lcd driver/sources/stp7_interrupt_vector.c File Reference

```
#include "generic.h"
```

Include dependency graph for stp7_interrupt_vector.c:

../awu lcd driver/sources/stp7_interrupt_vector.c
Generic.h

Functions:

- void **NonHandledInterrupt** ()
- void **AWU_It** ()

Variables:

- void (*const **_vectab** [])()

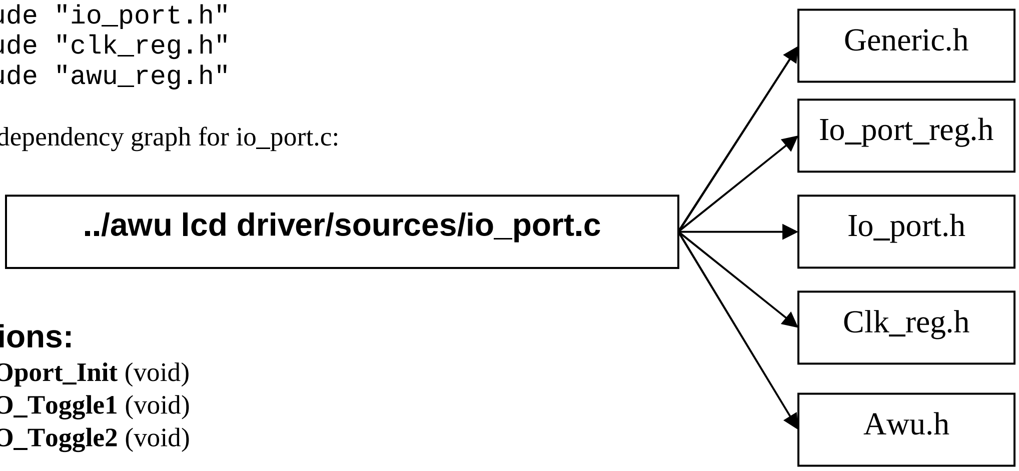
Detailed Description:

Interrupt vectors table mapping
Definition in file **stp7_interrupt_vector.c**

../awu lcd driver/sources/io_port.c File Reference

```
#include "generic.h"
#include "io_port_reg.h"
#include "io_port.h"
#include "clk_reg.h"
#include "awu_reg.h"
```

Include dependency graph for io_port.c:



Functions:

- void **IOport_Init** (void)
- void **IO_Toggle1** (void)
- void **IO_Toggle2** (void)

Variables:

- u8 **first_CC1**
- u8 **index**

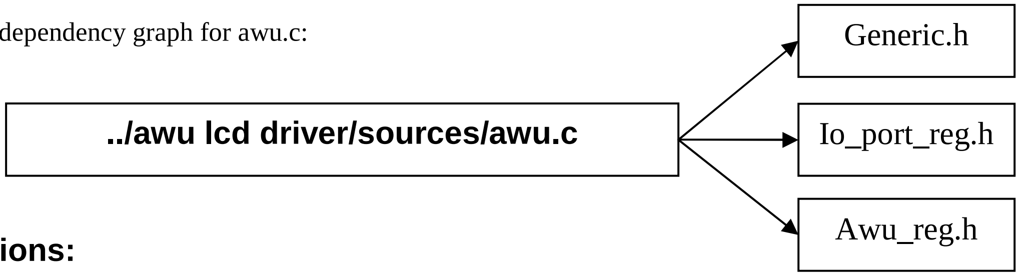
Detailed Description:

IOs configuration depending on the number of COM/SEG defined in generic.h and AWU function to toggle the IOs.
Definition in file **io_port.c**.

../awu lcd driver/sources/awu.c File Reference

```
#include "awu_reg.h"
#include "io_port_reg.h"
#include "generic.h"
```

Include dependency graph for awu.c:



Functions:

- void **AWU_Init** (void)
- interrupt void **AWU_It** (void)

Variables:

- u8 **first** = 0

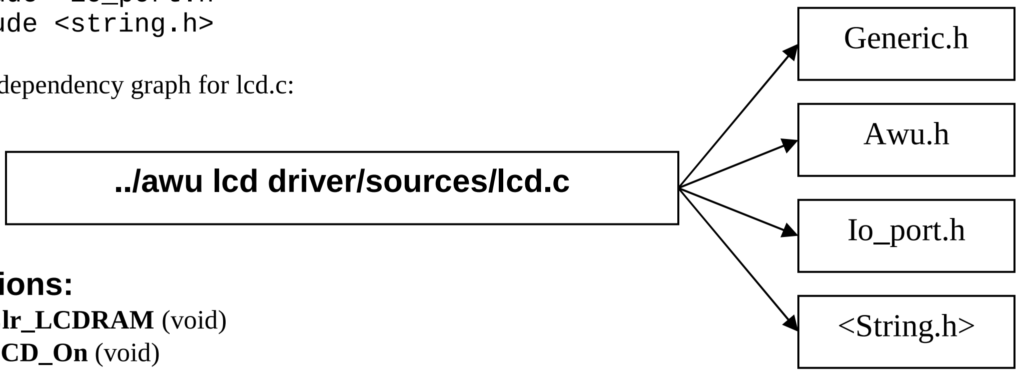
Detailed Description:

AWU function managing LCD display with the dedicated timebase
Definition in file **awu.c**

../awu lcd driver/sources/lcd.c File Reference

```
#include "generic.h"
#include "awu.h"
#include "io_port.h"
#include <string.h>
```

Include dependency graph for lcd.c:



Functions:

- void **Clr_LCDRAM** (void)
- void **LCD_On** (void)
- void **Convert** (char *c)
- void **Write_char** (char car, u8 pos)
- void **LCD_Display** (char *str)

Variables:

- u8 **LCDRAM** [COM *(SEG/8)]
- const unsigned int **number** [10] = {0xA665, 0x0320, 0x2A54, 0x2A34, 0x0E30, 0x2C34, 0x2C74, 0x2990, 0x2E74, 0x2E34}
- const unsigned int **letter** [29] = {0x2E70,0x323C,0x2444,0x322C,0x2C54,0x2C50,0x2474,0x0E70, 0x300C,0x0264,0x1109,0x0444,0x8760,0x8661,0x2664,0x2E50,0x2665,0x2E51,0x2C34,0x3008,0x0664, 0x05C0,0x06E1,0x8181,0x8180,0x2184,0x4000,0x0002,0x0000}
- u8 **digit** [(SEG/8)]

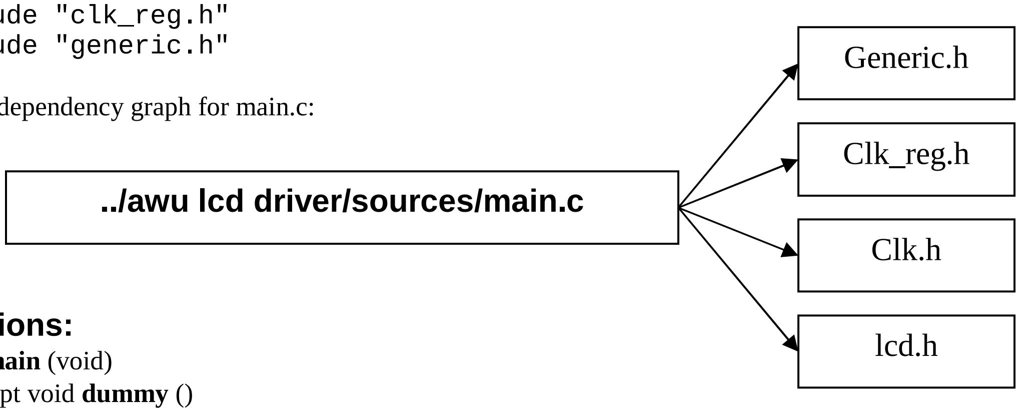
Detailed Description:

All the LCD functions: LCD RAM clear, LCD ON, LCD Display its purpose. Definition in file **lcd.c**

../awu lcd driver/sources/main.c File Reference

```
#include "lcd.h"
#include "clk.h"
#include "clk_reg.h"
#include "generic.h"
```

Include dependency graph for main.c:



Functions:

- void **main** (void)
- interrupt void **dummy** ()

Detailed Description:

Main file using the LCD driver
Definition in file **main.c**

Index