

Ex16-LCD-Ana

1

Generated by Doxygen 1.7.5

Tue Jun 19 2012 21:26:01

Contents

1	File Index	1
1.1	File List	1
2	File Documentation	3
2.1	delay.c File Reference	3
2.1.1	Detailed Description	4
2.1.2	Function Documentation	4
2.1.2.1	Delay	4
2.1.2.2	Delay_Us	4
2.2	delay.c	5
2.3	delay.h File Reference	6
2.3.1	Detailed Description	7
2.3.2	Define Documentation	7
2.3.2.1	Delay200uS_count	7
2.3.2.2	Delay_15mS_Cnt	7
2.3.2.3	Delay_1mS_Cnt	7
2.3.2.4	Delay_1S_Cnt	7
2.3.2.5	Delay_2mS_Cnt	7
2.3.2.6	Delay_5mS_Cnt	8
2.3.2.7	Fcy	8
2.3.3	Function Documentation	8
2.3.3.1	Delay	8
2.3.3.2	Delay_Us	8

2.4	delay.h	9
2.5	Ex16-LCD-Ana.h File Reference	10
2.5.1	Detailed Description	10
2.5.2	Function Documentation	11
2.5.2.1	Initialize	11
2.5.3	Variable Documentation	12
2.5.3.1	dirty	12
2.5.3.2	message	12
2.6	Ex16-LCD-Ana.h	13
2.7	Initialize.c File Reference	13
2.7.1	Detailed Description	14
2.7.2	Function Documentation	14
2.7.2.1	Initialize	14
2.8	Initialize.c	15
2.9	lcd.c File Reference	17
2.9.1	Detailed Description	17
2.9.2	Function Documentation	17
2.9.2.1	LCDcountedstring	17
2.10	lcd.c	18
2.11	lcd.h File Reference	19
2.11.1	Detailed Description	20
2.11.2	Define Documentation	20
2.11.2.1	LCDclear	20
2.11.2.2	LCDhomet	20
2.11.2.3	LCDleft	20
2.11.2.4	LCDline2	21
2.11.2.5	LCDposition	21
2.11.2.6	LCDright	21
2.11.2.7	LCDshift	21
2.11.3	Function Documentation	21
2.11.3.1	LCDcommand	21

2.11.3.2	LCDcountedstring	22
2.11.3.3	LCDinit	23
2.11.3.4	LCDletter	25
2.12	lcd.h	26
2.13	lcd_intern.h File Reference	26
2.13.1	Detailed Description	27
2.13.2	Define Documentation	28
2.13.2.1	LCD_DATA	28
2.13.2.2	LCD_DATAPORT	28
2.13.2.3	LCD_DATATRIS	28
2.13.2.4	LCD_ENABLE	28
2.13.2.5	LCD_ENABLE_TRIS	28
2.13.2.6	LCD_RS	28
2.13.2.7	LCD_RS_TRIS	28
2.13.2.8	LCD_RW	29
2.13.2.9	LCD_RW_TRIS	29
2.14	lcd_intern.h	29
2.15	LCDcommand.c File Reference	30
2.15.1	Detailed Description	30
2.15.2	Function Documentation	30
2.15.2.1	LCDcommand	30
2.16	LCDcommand.c	31
2.17	LCDcountedstring.c File Reference	32
2.17.1	Detailed Description	33
2.17.2	Function Documentation	33
2.17.2.1	LCDcountedstring	33
2.18	LCDcountedstring.c	33
2.19	LCDinit.c File Reference	34
2.19.1	Detailed Description	35
2.19.2	Function Documentation	35
2.19.2.1	LCDinit	35

2.20 LCDinit.c	36
2.21 LCDletter.c File Reference	38
2.21.1 Detailed Description	38
2.21.2 Function Documentation	38
2.21.2.1 LCDletter	39
2.22 LCDletter.c	40
2.23 main.c File Reference	40
2.23.1 Detailed Description	42
2.23.2 Function Documentation	42
2.23.2.1 _FICD	42
2.23.2.2 _FOSC	42
2.23.2.3 _FOSCSEL	42
2.23.2.4 _FPOR	42
2.23.2.5 _FWDT	42
2.23.2.6 main	42
2.23.3 Variable Documentation	44
2.23.3.1 szMessage	44
2.24 main.c	44
2.25 Tmr6Interrupt.c File Reference	46
2.25.1 Detailed Description	46
2.25.2 Function Documentation	47
2.25.2.1 __attribute__	47
2.25.3 Variable Documentation	47
2.25.3.1 delayCount	47
2.26 Tmr6Interrupt.c	47

Chapter 1

File Index

1.1 File List

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

delay.c	Routines used to provide delays for the LCD routines	3
delay.h	Declarations for LCD delay routines	6
Ex16-LCD-Ana.h	Global declarations for Ex16-LCD-Ana	10
Initialize.c	Initialization for Ex16-LCD-Ana	13
lcd.c	LCD routines	17
lcd.h	LCD definitions	19
lcd_intern.h	Definitions used within LCD routines	26
LCDcommand.c	Send a command to the LCD	30
LCDcountedstring.c	Send a specific number of characters to the LCD	32
LCDinit.c	Initialize the LCD	34
LCDletter.c	Send a character to the LCD	38
main.c	Mainline for Ex16-LCD-Ana	40

[Tmr6Interrupt.c](#)

Timer 6 interrupt service routine 46

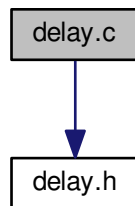
Chapter 2

File Documentation

2.1 delay.c File Reference

Routines used to provide delays for the LCD routines.

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



Functions

- void `Delay` (unsigned int delay_count)
Delay for a specific count.
- void `Delay_Us` (unsigned int delayUs_count)
Delay for a specified number of microseconds.

Variables

- unsigned int **temp_count**

2.1.1 Detailed Description

Routines used to provide delays for the LCD routines.

Definition in file [delay.c](#).

2.1.2 Function Documentation

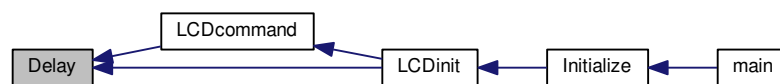
2.1.2.1 void Delay (unsigned int *delay_count*)

Delay for a specific count.

Definition at line 10 of file [delay.c](#).

```
{  
    temp_count = delay_count +1;  
    asm volatile("outer: dec _temp_count");  
    asm volatile("cp0 _temp_count");  
    asm volatile("bra z, done");  
    asm volatile("do #3200, inner" );  
    asm volatile("nop");  
    asm volatile("inner: nop");  
    asm volatile("bra outer");  
    asm volatile("done:");  
}
```

Here is the caller graph for this function:



2.1.2.2 void Delay_Us (unsigned int *delayUs_count*)

Delay for a specified number of microseconds.

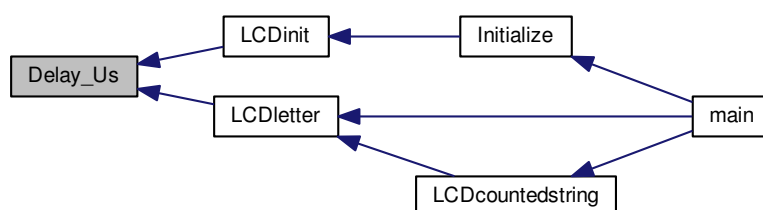
Definition at line 24 of file [delay.c](#).

```

{
    temp_count = delayUs_count +1;
    asm volatile("outer1: dec _temp_count");
    asm volatile("cp0 _temp_count");
    asm volatile("bra z, done1");
    asm volatile("do #1500, inner1" );
    asm volatile("nop");
    asm volatile("inner1: nop");
    asm volatile("bra outer1");
    asm volatile("done1:");
}

```

Here is the caller graph for this function:



2.2 delay.c

```

00001
00005 #include "delay.h"
00006
00007 unsigned int temp_count;
00008
00010 void Delay( unsigned int delay_count )
00011 {
00012     temp_count = delay_count +1;
00013     asm volatile("outer: dec _temp_count");
00014     asm volatile("cp0 _temp_count");
00015     asm volatile("bra z, done");
00016     asm volatile("do #3200, inner" );
00017     asm volatile("nop");
00018     asm volatile("inner: nop");
00019     asm volatile("bra outer");
00020     asm volatile("done:");
00021 }
00022
00024 void Delay_Us( unsigned int delayUs_count )
00025 {
00026     temp_count = delayUs_count +1;
00027     asm volatile("outer1: dec _temp_count");
00028     asm volatile("cp0 _temp_count");
00029     asm volatile("bra z, done1");
00030     asm volatile("do #1500, inner1" );
00031     asm volatile("nop");

```

```

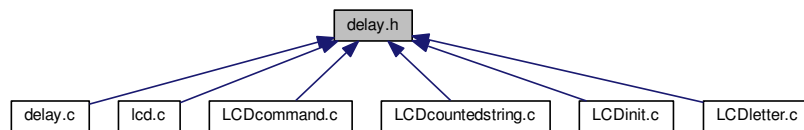
00032     asm volatile("inner1: nop");
00033     asm volatile("bra outer1");
00034     asm volatile("done1:");
00035 }
00036

```

2.3 delay.h File Reference

Declarations for LCD delay routines.

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



Defines

- #define [Delay200uS_count](#) (Fcy * 0.0002) / 1080
Counts for a 200 us delay.
- #define [Delay_15mS_Cnt](#) (Fcy * 0.015) / 2950
Counts for a 15 ms delay.
- #define [Delay_1mS_Cnt](#) (Fcy * 0.001) / 2950
Counts for a 1 ms delay.
- #define [Delay_1S_Cnt](#) (Fcy * 1) / 2950
Counts for a 1 second delay.
- #define [Delay_2mS_Cnt](#) (Fcy * 0.002) / 2950
Counts for a 2 ms delay.
- #define [Delay_5mS_Cnt](#) (Fcy * 0.005) / 2950
Counts for a 5 ms delay.
- #define [Fcy](#) 16000000
Instruction clock Hz.

Functions

- void [Delay](#) (unsigned int delay_count)

Delay for a specific count.

- void [Delay_Us](#) (unsigned int delayUs_count)

Delay for a specified number of microseconds.

2.3.1 Detailed Description

Declarations for LCD delay routines.

Definition in file [delay.h](#).

2.3.2 Define Documentation

2.3.2.1 `#define Delay200uS_count (Fcy * 0.0002) / 1080`

Counts for a 200 us delay.

Definition at line 15 of file [delay.h](#).

2.3.2.2 `#define Delay_15mS_Cnt (Fcy * 0.015) / 2950`

Counts for a 15 ms delay.

Definition at line 23 of file [delay.h](#).

2.3.2.3 `#define Delay_1mS_Cnt (Fcy * 0.001) / 2950`

Counts for a 1 ms delay.

Definition at line 17 of file [delay.h](#).

2.3.2.4 `#define Delay_1S_Cnt (Fcy * 1) / 2950`

Counts for a 1 second delay.

Definition at line 25 of file [delay.h](#).

2.3.2.5 `#define Delay_2mS_Cnt (Fcy * 0.002) / 2950`

Counts for a 2 ms delay.

Definition at line 19 of file [delay.h](#).

2.3.2.6 #define Delay_5mS_Cnt (Fcy * 0.005) / 2950

Counts for a 5 ms delay.

Definition at line 21 of file [delay.h](#).

2.3.2.7 #define Fcy 16000000

Instruction clock Hz.

Definition at line 7 of file [delay.h](#).

2.3.3 Function Documentation

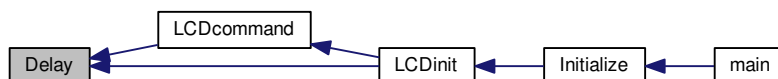
2.3.3.1 void Delay (unsigned int *delay_count*)

Delay for a specific count.

Definition at line 10 of file [delay.c](#).

```
{
    temp_count = delay_count +1;
    asm volatile("outer: dec _temp_count");
    asm volatile("cp0 _temp_count");
    asm volatile("bra z, done");
    asm volatile("do #3200, inner" );
    asm volatile("nop");
    asm volatile("inner: nop");
    asm volatile("bra outer");
    asm volatile("done:");
}
```

Here is the caller graph for this function:



2.3.3.2 void Delay_Us (unsigned int *delayUs_count*)

Delay for a specified number of microseconds.

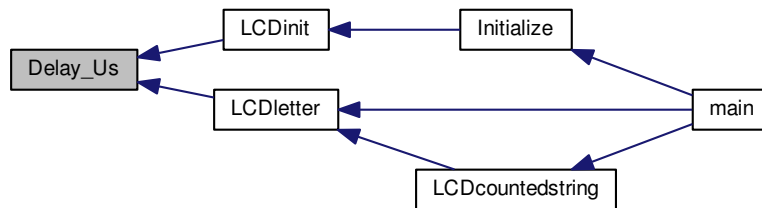
Definition at line 24 of file [delay.c](#).

```

{
    temp_count = delayUs_count + 1;
    asm volatile("outer1: dec _temp_count");
    asm volatile("cp0 _temp_count");
    asm volatile("bra z, done1");
    asm volatile("do #1500, inner1" );
    asm volatile("nop");
    asm volatile("inner1: nop");
    asm volatile("bra outer1");
    asm volatile("done1:");
}

```

Here is the caller graph for this function:



2.4 delay.h

```

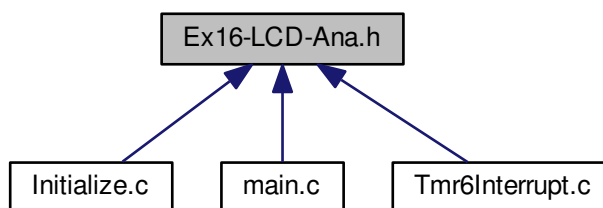
00001
00005 //define Fcy 14754600
00007 #define Fcy 16000000
00008
00010 void Delay( unsigned int delay_count );
00012 void Delay_Us( unsigned int delayUs_count );
00013
00015 #define Delay200uS_count (Fcy * 0.0002) / 1080
00016
00017 #define Delay_1mS_Cnt (Fcy * 0.001) / 2950
00018
00019 #define Delay_2mS_Cnt (Fcy * 0.002) / 2950
00020
00021 #define Delay_5mS_Cnt (Fcy * 0.005) / 2950
00022
00023 #define Delay_15mS_Cnt (Fcy * 0.015) / 2950
00024
00025 #define Delay_1S_Cnt (Fcy * 1) / 2950
00026

```

2.5 Ex16-LCD-Ana.h File Reference

Global declarations for Ex16-LCD-Ana.

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



Functions

- void [Initialize](#) (void)
Initialization for Ex16-LCD-Ana.

Variables

- EXTERN int [dirty](#)
Dirty flag - if non-zero display is updated.
- EXTERN int [message](#)
Current message number to display.

2.5.1 Detailed Description

Global declarations for Ex16-LCD-Ana. File: [Ex16-LCD-Ana.h](#) Author: jjmcd

Created on June 19, 2012, 9:28 AM

Definition in file [Ex16-LCD-Ana.h](#).

2.5.2 Function Documentation

2.5.2.1 void Initialize (void)

Initialization for Ex16-LCD-Ana.

Initializes the ports and then initializes timer 6 Initializes the dirty flag and message number

Definition at line 36 of file [Initialize.c](#).

```
{
    // Set the instruction clock speed
    //
    // Fcy 40 MIPS
    // DOZE = Fcy/8 = 011
    // DOZEN = 1
    // PLLPRE 2 = 00000
    // PLLDIV 40 = .38 = 0x26 = 0 0010 0110
    // PLLPOST 2 00
    //ROI   DOZE  DOZEN  FRCDIV  PLLPOST X   PLLPRE
    // 15 14 13 12 11 10 9 8   7 6   5 4 3 2 1 0
    // 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

    CLKDIV = 0x0000;
    PLLFBD = 0x0026;

    // Fcy 20 MIPS
    // PLLPRE 2 = 00000
    // PLLDIV 40 = .38 = 0x26 = 0 0010 0110
    // PLLPOST 4 01
    //ROI   DOZE  DOZEN  FRCDIV  PLLPOST X   PLLPRE
    // 15 14 13 12 11 10 9 8   7 6   5 4 3 2 1 0
    // 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
/*
    CLKDIV = 0x0008;
    PLLFBD = 0x0026;
*/

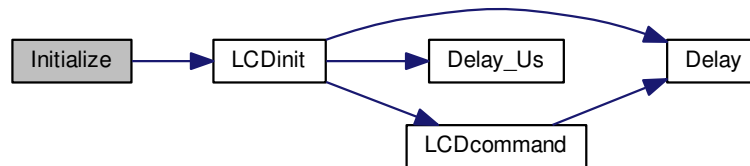
    TRISA = 0;           // All PORTA pins outputs
    LATA = 0x0001;       // Right LED on

    // Set timer 6 for right LED
    // Explanation ...
    // Timer 6 will increment every 128 instruction cycles
    // Once the count reaches 50,000, the timer 6 interrupt will fire
    // and the count will be reset
    PR6 = 50000;         // Timer 6 counter to 50,000
    TMR6 = 0;            // Clear timer 6
    T6CON = 0x8030;      // 1:256 prescale, timer on, Clock Fcy
    IEC2bits.T6IE = 1;   // Enable Timer 6 interrupt

    // Initialize the LCD
    LCDinit();

    // Initialize global variables
    dirty = 0;           // Message dirty flag
    message = 0;         // Current message number
}
```

Here is the call graph for this function:



Here is the caller graph for this function:



2.5.3 Variable Documentation

2.5.3.1 EXTERN int dirty

Dirty flag - if non-zero display is updated.

Definition at line 19 of file [Ex16-LCD-Ana.h](#).

2.5.3.2 EXTERN int message

Current message number to display.

Definition at line 21 of file [Ex16-LCD-Ana.h](#).

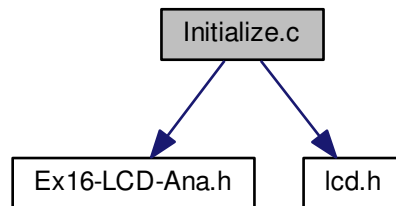
2.6 Ex16-LCD-Ana.h

```
00001
00011 #ifndef EX16_LCD_ANA_H
00012 #define EX16_LCD_ANA_H
00013
00014 #ifdef __cplusplus
00015 extern "C" {
00016 #endif
00017
00019 EXTERN int dirty;
00021 EXTERN int message;
00022
00024 void Initialize( void );
00025
00026
00027 #ifdef __cplusplus
00028 }
00029 #endif
00030
00031 #endif /* EX16_LCD_ANA_H */
00032
```

2.7 Initialize.c File Reference

Initialization for Ex16-LCD-Ana.

#include "Ex16-LCD-Ana.h" #include "lcd.h" Include dependency graph for Initialize.c:



Defines

- #define **EXTERN** extern

Functions

- void [Initialize](#) (void)

Initialization for Ex16-LCD-Ana.

2.7.1 Detailed Description

Initialization for Ex16-LCD-Ana.

Definition in file [Initialize.c](#).

2.7.2 Function Documentation

2.7.2.1 void Initialize (void)

Initialization for Ex16-LCD-Ana.

Initializes the ports and then initializes timer 6
Initializes the dirty flag and message number

Definition at line 36 of file [Initialize.c](#).

```
{
    // Set the instruction clock speed
    //
    // Fcy 40 MIPS
    // DOZE = Fcy/8 = 011
    // DOZEN = 1
    // PLLPRE 2 = 00000
    // PLLDIV 40 = .38 = 0x26 = 0 0010 0110
    // PLLPOST 2 00
    //ROI   DOZE  DOZEN  FRCDIV  PLLPOST X   PLLPRE
    // 15 14 13 12 11 10 9 8   7 6   5 4 3 2 1 0
    // 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

    CLKDIV = 0x0000;
    PLLFBD = 0x0026;

    // Fcy 20 MIPS
    // PLLPRE 2 = 00000
    // PLLDIV 40 = .38 = 0x26 = 0 0010 0110
    // PLLPOST 4 01
    //ROI   DOZE  DOZEN  FRCDIV  PLLPOST X   PLLPRE
    // 15 14 13 12 11 10 9 8   7 6   5 4 3 2 1 0
    // 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
/*
    CLKDIV = 0x0008;
    PLLFBD = 0x0026;
*/

    TRISA = 0;           // All PORTA pins outputs
    LATA = 0x0001;       // Right LED on

    // Set timer 6 for right LED
    // Explanation ...
    // Timer 6 will increment every 128 instruction cycles
```

```

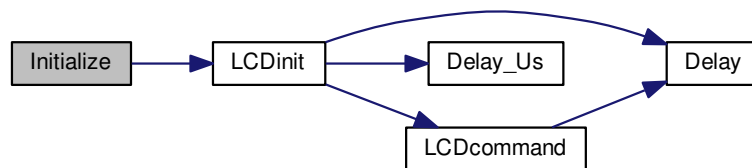
// Once the count reaches 50,000, the timer 6 interrupt will fire
// and the count will be reset
PR6 = 50000;           // Timer 6 counter to 50,000
TMR6 = 0;              // Clear timer 6
T6CON = 0x8030;        // 1:256 prescale, timer on, Clock Fcy
IEC2bits.T6IE = 1;     // Enable Timer 6 interrupt

// Initialize the LCD
LCDinit();

// Initialize global variables
dirty = 0;             // Message dirty flag
message = 0;           // Current message number
}

```

Here is the call graph for this function:



Here is the caller graph for this function:



2.8 Initialize.c

```

00001
00007 #if defined(__PIC24E__)
00008 #include <p24Exxxx.h>
00009

```

```

00010 #elif defined (__PIC24F__)
00011 #include <p24Fxxxx.h>
00012
00013 #elif defined (__PIC24H__)
00014 #include <p24Hxxxx.h>
00015
00016 #elif defined (__dsPIC30F__)
00017 #include <p30Fxxxx.h>
00018
00019 #elif defined (__dsPIC33E__)
00020 #include <p33Exxxx.h>
00021
00022 #elif defined (__dsPIC33F__)
00023 #include <p33Fxxxx.h>
00024
00025 #endif
00026
00027 #define EXTERN extern
00028 #include "Ex16-LCD-Ana.h"
00029
00030 #include "lcd.h"
00031
00032
00033
00036 void Initialize( void )
00037 {
00038     // Set the instruction clock speed
00039     //
00040     // Fcy 40 MIPS
00041     // DOZE = Fcy/8 = 011
00042     // DOZEN = 1
00043     // PLLPRE 2 = 00000
00044     // PLLDIV 40 = .38 = 0x26 = 0 0010 0110
00045     // PLLPOST 2 00
00046     //ROI   DOZE  DOZEN  FRCDIV  PLLPOST X   PLLPRE
00047     // 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
00048     // 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
00049
00050     CLKDIV = 0x0000;
00051     PLLFBD = 0x0026;
00052
00053     // Fcy 20 MIPS
00054     // PLLPRE 2 = 00000
00055     // PLLDIV 40 = .38 = 0x26 = 0 0010 0110
00056     // PLLPOST 4 01
00057     //ROI   DOZE  DOZEN  FRCDIV  PLLPOST X   PLLPRE
00058     // 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
00059     // 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
00060 /*
00061     CLKDIV = 0x0008;
00062     PLLFBD = 0x0026;
00063 */
00064
00065     TRISA = 0;           // All PORTA pins outputs
00066     LATA = 0x0001;      // Right LED on
00067
00068     // Set timer 6 for right LED
00069     // Explanation ...
00070     //   Timer 6 will increment every 128 instruction cycles
00071     //   Once the count reaches 50,000, the timer 6 interrupt will fire
00072     //   and the count will be reset
00073     PR6 = 50000;        // Timer 6 counter to 50,000
00074     TMR6 = 0;           // Clear timer 6
00075     T6CON = 0x8030;     // 1:256 prescale, timer on, Clock Fcy
00076     IEC2bits.T6IE = 1;  // Enable Timer 6 interrupt
00077
00078     // Initialize the LCD
00079     LCDinit();
00080

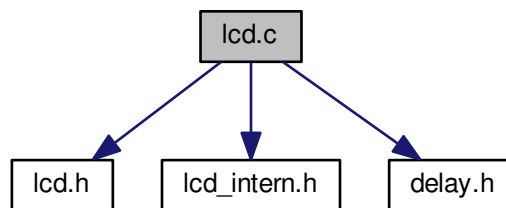
```

```
00081 // Initialize global variables
00082 dirty = 0; // Message dirty flag
00083 message = 0; // Current message number
00084
00085 }
```

2.9 lcd.c File Reference

LCD routines.

`#include "lcd.h" #include "lcd_intern.h" #include "delay.h"` Include dependency graph for lcd.c:



Functions

- void `LCDcountedstring` (unsigned char *data, unsigned char count)
Send a counted string to the LCD.

2.9.1 Detailed Description

LCD routines.

Definition in file [lcd.c](#).

2.9.2 Function Documentation

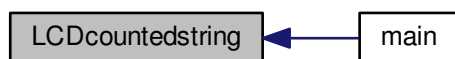
2.9.2.1 void `LCDcountedstring` (unsigned char * *data*, unsigned char *count*)

Send a counted string to the LCD.

Definition at line 47 of file [lcd.c](#).

```
{
    while ( count )
    {
        LCDletter( *data++ );
        count --;
    }
}
```

Here is the caller graph for this function:



2.10 lcd.c

```
00001
00007 #if defined(__PIC24E__)
00008 #include <p24Exxxx.h>
00009
00010 #elif defined (__PIC24F__)
00011 #include <p24Fxxxx.h>
00012
00013 #elif defined(__PIC24H__)
00014 #include <p24Hxxxx.h>
00015
00016 #elif defined(__dsPIC30F__)
00017 #include <p30Fxxxx.h>
00018
00019 #elif defined (__dsPIC33E__)
00020 #include <p33Exxxx.h>
00021
00022 #elif defined(__dsPIC33F__)
00023 #include <p33Fxxxx.h>
00024
00025 #endif
00026
00027 #include "lcd.h"
00028 #include "lcd_intern.h"
00029 #include "delay.h"
00030
00031
00032 /*
00033  For Explorer 16 board, here are the data and control signal definitions
00034  RS -> RB15
00035  E  -> RD4
00036  RW -> RD5
00037  DATA -> RE0 - RE7
```



```

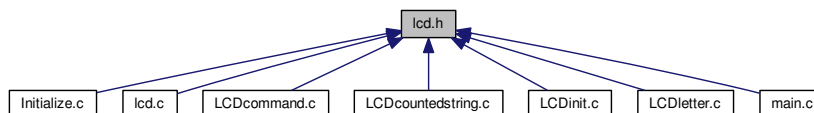
00038 */
00039
00040
00041
00042 /*****
00043 /***** LCD SUBROUTINE *****/
00044
00045
00047 void LCDcountedstring( unsigned char *data, unsigned char count )
00048 {
00049     while ( count )
00050     {
00051         LCDletter( *data++ );
00052         count --;
00053     }
00054 }

```

2.11 lcd.h File Reference

LCD definitions.

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



Defines

- #define **LCDclear**() LCDcommand(0x01)
Clear the LCD display and home cursor.
- #define **LCDhomet**() LCDcommand(0x02)
Set the LCD cursor to home.
- #define **LCDleft**() LCDcommand(0x10)
Move the LCD cursor to the left.
- #define **LCDline2**() LCDcommand(0xC0)
Position the LCD cursor to the second line.
- #define **LCDposition**(a) LCDcommand(0x80 + (a & 0x7f))
Set the LCD cursor position.
- #define **LCDright**() LCDcommand(0x14)
Move the LCD cursor to the right.
- #define **LCDshift**() LCDcommand(0x1C)

Shift the LCD display.

Functions

- void [LCDcommand](#) (char cmd)
Send a command to the LCD.
- void [LCDcountedstring](#) (unsigned char *data, unsigned char count)
Send a counted string to the LCD.
- void [LCDinit](#) (void)
Initialize the LCD.
- void [LCDletter](#) (char data)
Send a character to the LCD.

2.11.1 Detailed Description

LCD definitions.

Definition in file [lcd.h](#).

2.11.2 Define Documentation

2.11.2.1 `#define LCDclear() LCDcommand(0x01)`

Clear the LCD display and home cursor.

Definition at line 26 of file [lcd.h](#).

2.11.2.2 `#define LCDhomet() LCDcommand(0x02)`

Set the LCD cursor to home.

Definition at line 28 of file [lcd.h](#).

2.11.2.3 `#define LCDleft() LCDcommand(0x10)`

Move the LCD cursor to the left.

Definition at line 22 of file [lcd.h](#).

2.11.2.4 `#define LCDline2() LCDcommand(0xC0)`

Position the LCD cursor to the second line.

Definition at line 30 of file [lcd.h](#).

2.11.2.5 `#define LCDposition(a) LCDcommand(0x80 + (a & 0x7f)`

Set the LCD cursor position.

Definition at line 32 of file [lcd.h](#).

2.11.2.6 `#define LCDright() LCDcommand(0x14)`

Move the LCD cursor to the right.

Definition at line 20 of file [lcd.h](#).

2.11.2.7 `#define LCDshift() LCDcommand(0x1C)`

Shift the LCD display.

Definition at line 24 of file [lcd.h](#).

2.11.3 Function Documentation

2.11.3.1 `void LCDcommand(char cmd)`

Send a command to the LCD.

Definition at line 32 of file [LCDcommand.c](#).

```
{
    LCD_DATA &= 0xFF00; // prepare RD0 - RD7
    LCD_DATA |= cmd; // command byte to lcd
    LCD_RW = 0; // ensure RW is 0
    LCD_RS = 0;
    LCD_ENABLE = 1; // toggle E line
    Nop();
    Nop();
    Nop();
    LCD_ENABLE = 0;
    Delay(Delay_5mS_Cnt); // 5ms delay
}
```

Here is the call graph for this function:



Here is the caller graph for this function:



2.11.3.2 void LCDcountedstring (unsigned char * *data*, unsigned char *count*)

Send a counted string to the LCD.

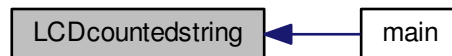
Definition at line 47 of file `lcd.c`.

```
{  
    while ( count )  
    {  
        LCDletter ( *data++ );  
        count --;  
    }  
}
```

Here is the call graph for this function:



Here is the caller graph for this function:



2.11.3.3 void LCDinit (void)

Initialize the LCD.

Definition at line 32 of file [LCDinit.c](#).

```

{
    // 15mS delay after Vdd reaches nnVdc before proceeding with LCD
    initialization
    // not always required and is based on system Vdd rise rate
    Delay(Delay_15mS_Cnt); // 15ms delay

    /* set initial states for the data and control pins */
    LCD_DATA &= 0xFF00;
    LCD_RW = 0; // R/W state set low
    LCD_RS = 0; // RS state set low
    LCD_ENABLE = 0; // E state set low

    /* set data and control pins to outputs */
    LCD_DATATRIS &= 0xFF00;
    LCD_RW_TRIS = 0; // RW pin set as output
    LCD_RS_TRIS = 0; // RS pin set as output
    LCD_ENABLE_TRIS = 0; // E pin set as output

    /* 1st LCD initialization sequence */
    LCD_DATA &= 0xFF00;
  
```

```

LCD_DATA |= 0x0038;
LCD_ENABLE = 1;
Nop();
Nop();
Nop();
LCD_ENABLE = 0; // toggle E signal
Delay(Delay_5mS_Cnt); // 5ms delay

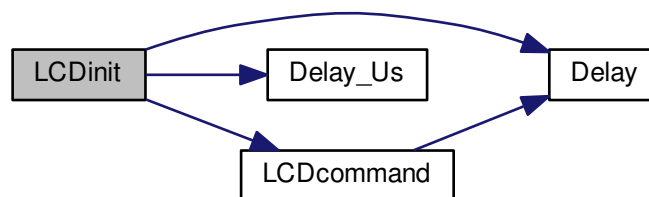
/* 2nd LCD initialization sequence */
LCD_DATA &= 0xFF00;
LCD_DATA |= 0x0038;
LCD_ENABLE = 1;
Nop();
Nop();
Nop();
LCD_ENABLE = 0; // toggle E signal
Delay_Us(Delay200uS_count); // 200uS delay

/* 3rd LCD initialization sequence */
LCD_DATA &= 0xFF00;
LCD_DATA |= 0x0038;
LCD_ENABLE = 1;
Nop();
Nop();
Nop();
LCD_ENABLE = 0; // toggle E signal
Delay_Us(Delay200uS_count); // 200uS delay

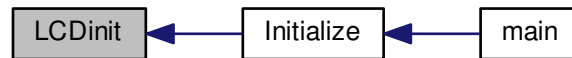
LCDcommand(0x38); // function set
LCDcommand(0x0C); // Display on/off control, cursor blink off (0x0C)
LCDcommand(0x06); // entry mode set (0x06)
}

```

Here is the call graph for this function:



Here is the caller graph for this function:



2.11.3.4 void LCDletter (char data)

Send a character to the LCD.

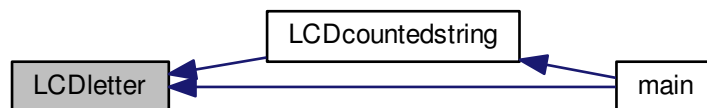
Definition at line 33 of file [LCDletter.c](#).

```
{  
    LCD_RW = 0; // ensure RW is 0  
    LCD_RS = 1; // assert register select to 1  
    LCD_DATA &= 0xFF00; // prepare RD0 - RD7  
    LCD_DATA |= data; // data byte to lcd  
    LCD_ENABLE = 1;  
    Nop();  
    Nop();  
    Nop();  
    LCD_ENABLE = 0; // toggle E signal  
    LCD_RS = 0; // negate register select to 0  
    Delay_Us(Delay200uS_count); // 200uS delay  
    Delay_Us(Delay200uS_count); // 200uS delay  
}
```

Here is the call graph for this function:



Here is the caller graph for this function:



2.12 lcd.h

```

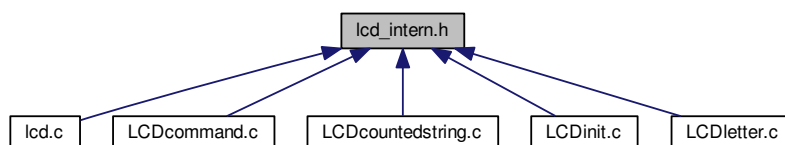
00001
00006 /***** LCD FUNCTION PROTOYPES *****/
00007
00009 void LCDinit( void );           // initialize display
00011 void LCDcommand( char cmd );    // write command to lcd
00013 void LCDletter( char data );     // write data to lcd
00015 void LCDcountedstring ( unsigned char *data, unsigned char count );
00016
00017
00018 /***** LCD COMMAND FUCNTION PROTOTYPES *****/
00020 #define LCDright()  LCDcommand( 0x14 )
00021
00022 #define LCDleft()   LCDcommand( 0x10 )
00023
00024 #define LCDshift()  LCDcommand( 0x1C )
00025
00026 #define LCDclear()   LCDcommand( 0x01 )
00027
00028 #define LCDhomet()   LCDcommand( 0x02 )
00029
00030 #define LCDline2()    LCDcommand( 0xC0 ) // (0xC0)
00031
00032 #define LCDposition(a)  LCDcommand( 0x80 + ( a & 0x7f) )

```

2.13 lcd_intern.h File Reference

Definitions used within LCD routines.

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



Defines

- #define [LCD_DATA](#) LATE
LCD data port latch.
- #define [LCD_DATAPORT](#) PORTE
LCD data port.
- #define [LCD_DATATRIS](#) TRISE
LCD data port direction register.
- #define [LCD_ENABLE](#) LATDbits.LATD4
LCD Enable pin.
- #define [LCD_ENABLE_TRIS](#) TRISDbits.TRISD4
LCD Enable direction register bit.
- #define [LCD_RS](#) LATBbits.LATB15
LCD Register select pin.
- #define [LCD_RS_TRIS](#) TRISBbits.TRISB15
LCD Register select direction register bit.
- #define [LCD_RW](#) LATDbits.LATD5
LCD Read/Write pin.
- #define [LCD_RW_TRIS](#) TRISDbits.TRISD5
LCD Read/Write direction register bit.

2.13.1 Detailed Description

Definitions used within LCD routines. This file contains definitions of the various connections to the LCD on the Explorer 16 board. They are uninteresting outside the LCD routines.

Definition in file [lcd_intern.h](#).

2.13.2 Define Documentation

2.13.2.1 #define LCD_DATA LATE

LCD data port latch.

Definition at line 49 of file [lcd_intern.h](#).

2.13.2.2 #define LCD_DATAPORT PORTE

LCD data port.

Definition at line 51 of file [lcd_intern.h](#).

2.13.2.3 #define LCD_DATATRIS TRISE

LCD data port direction register.

Definition at line 53 of file [lcd_intern.h](#).

2.13.2.4 #define LCD_ENABLE LATDbits.LATD4

LCD Enable pin.

Definition at line 37 of file [lcd_intern.h](#).

2.13.2.5 #define LCD_ENABLE_TRIS TRISDbits.TRISD4

LCD Enable direction register bit.

Definition at line 45 of file [lcd_intern.h](#).

2.13.2.6 #define LCD_RS LATBbits.LATB15

LCD Register select pin.

Definition at line 35 of file [lcd_intern.h](#).

2.13.2.7 #define LCD_RS_TRIS TRISBbits.TRISB15

LCD Register select direction register bit.

Definition at line 43 of file [lcd_intern.h](#).

2.13.2.8 #define LCD_RW LATDbits.LATD5

LCD Read/Write pin.

Definition at line 33 of file [lcd_intern.h](#).

2.13.2.9 #define LCD_RW_TRIS TRISDbits.TRISD5

LCD Read/Write direction register bit.

Definition at line 41 of file [lcd_intern.h](#).

2.14 lcd_intern.h

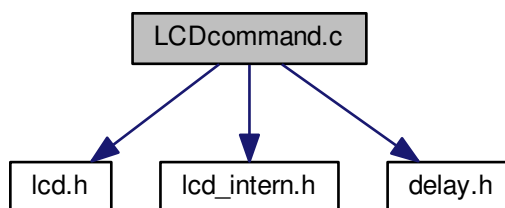
```
00001
00009 /*
00010  * File:    lcd_intern.h
00011  * Author:  jjmcd
00012  *
00013  * Created on June 19, 2012, 12:57 PM
00014  */
00015
00016 #ifndef LCD_INTERN_H
00017 #define LCD_INTERN_H
00018
00019 #ifdef __cplusplus
00020 extern "C" {
00021 #endif
00022
00023 /*
00024  * For Explorer 16 board, here are the data and control signal definitions
00025  * RS -> RB15
00026  * E -> RD4
00027  * LCD_RW -> RD5
00028  * DATA -> RE0 - RE7
00029  */
00030
00031 // Control signal data pins
00033 #define LCD_RW LATDbits.LATD5
00034
00035 #define LCD_RS LATBbits.LATB15
00036
00037 #define LCD_ENABLE LATDbits.LATD4
00038
00039 // Control signal pin direction
00041 #define LCD_RW_TRIS TRISDbits.TRISD5
00042
00043 #define LCD_RS_TRIS TRISBbits.TRISB15
00044
00045 #define LCD_ENABLE_TRIS TRISDbits.TRISD4
00046
00047 // Data signals and pin direction
00049 #define LCD_DATA LATE
00050
00051 #define LCD_DATAPORT PORTE
00052
00053 #define LCD_DATATRIS TRISE
00054
00055 #ifdef __cplusplus
```

```
00056 }  
00057 #endif  
00058  
00059 #endif /* LCD_INTERN_H */  
00060
```

2.15 LCDcommand.c File Reference

Send a command to the LCD.

```
#include "lcd.h" #include "lcd_intern.h" #include "delay.-  
h" Include dependency graph for LCDcommand.c:
```



Functions

- void [LCDcommand](#) (char cmd)
Send a command to the LCD.

2.15.1 Detailed Description

Send a command to the LCD.

Definition in file [LCDcommand.c](#).

2.15.2 Function Documentation

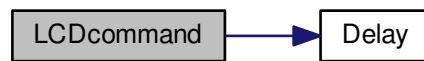
2.15.2.1 void LCDcommand (char cmd)

Send a command to the LCD.

Definition at line 32 of file [LCDcommand.c](#).

```
{
    LCD_DATA &= 0xFF00; // prepare RD0 - RD7
    LCD_DATA |= cmd; // command byte to lcd
    LCD_RW = 0; // ensure RW is 0
    LCD_RS = 0;
    LCD_ENABLE = 1; // toggle E line
    Nop();
    Nop();
    Nop();
    LCD_ENABLE = 0;
    Delay(Delay_5mS_Cnt); // 5ms delay
}
```

Here is the call graph for this function:



Here is the caller graph for this function:



2.16 LCDcommand.c

```
00001
00007 #if defined(__PIC24E__)
00008 #include <p24Exxxx.h>
00009
00010 #elif defined (__PIC24F__)
00011 #include <p24Fxxx.h>
00012
00013 #elif defined(__PIC24H__)
00014 #include <p24Hxxx.h>
00015
```

```

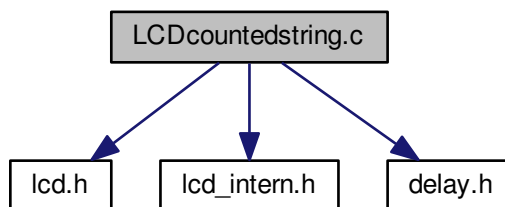
00016 #elif defined(__dsPIC30F__)
00017 #include <p30Fxxxx.h>
00018
00019 #elif defined (__dsPIC33E__)
00020 #include <p33Exxxx.h>
00021
00022 #elif defined(__dsPIC33F__)
00023 #include <p33Fxxxx.h>
00024
00025 #endif
00026
00027 #include "lcd.h"
00028 #include "lcd_intern.h"
00029 #include "delay.h"
00030
00032 void LCDcommand( char cmd )          // subrouitune for lcd commands
00033 {
00034     LCD_DATA &= 0xFF00; // prepare RD0 - RD7
00035     LCD_DATA |= cmd; // command byte to lcd
00036     LCD_RW = 0; // ensure RW is 0
00037     LCD_RS = 0;
00038     LCD_ENABLE = 1; // toggle E line
00039     Nop();
00040     Nop();
00041     Nop();
00042     LCD_ENABLE = 0;
00043     Delay(Delay_5mS_Cnt); // 5ms delay
00044 }
00045

```

2.17 LCDcountedstring.c File Reference

Send a specific number of characters to the LCD.

```
#include "lcd.h" #include "lcd_intern.h" #include "delay.-
h" Include dependency graph for LCDcountedstring.c:
```



Functions

- void [LCDcountedstring](#) (unsigned char *data, unsigned char count)
Send a counted string to the LCD.

2.17.1 Detailed Description

Send a specific number of characters to the LCD.

Definition in file [LCDcountedstring.c](#).

2.17.2 Function Documentation

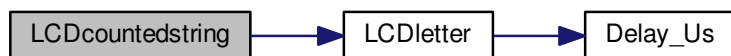
2.17.2.1 void LCDcountedstring (unsigned char * data, unsigned char count)

Send a counted string to the LCD.

Definition at line 33 of file [LCDcountedstring.c](#).

```
{  
    while (count)  
    {  
        LCDletter(*data++);  
        count--;  
    }  
}
```

Here is the call graph for this function:



2.18 LCDcountedstring.c

```
00001  
00007 #if defined(__PIC24E__)  
00008 #include <p24Exxxx.h>  
00009  
00010 #elif defined (__PIC24F__)  
00011 #include <p24Fxxxx.h>
```

```

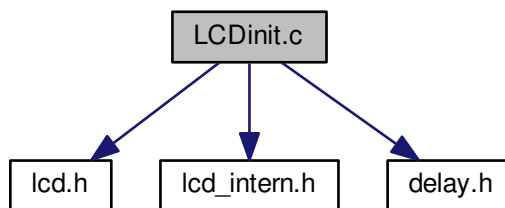
00012
00013 #elif defined(__PIC24H__)
00014 #include <p24Hxxxx.h>
00015
00016 #elif defined(__dsPIC30F__)
00017 #include <p30Fxxxx.h>
00018
00019 #elif defined (__dsPIC33E__)
00020 #include <p33Exxxx.h>
00021
00022 #elif defined(__dsPIC33F__)
00023 #include <p33Fxxxx.h>
00024
00025 #endif
00026
00027 #include "lcd.h"
00028 #include "lcd_intern.h"
00029 #include "delay.h"
00030
00031
00033 void LCDcountedstring( unsigned char *data, unsigned char count)
00034 {
00035     while (count)
00036     {
00037         LCDletter(*data++);
00038         count--;
00039     }
00040 }

```

2.19 LCDinit.c File Reference

Initialize the LCD.

```
#include "lcd.h" #include "lcd_intern.h" #include "delay.-
h" Include dependency graph for LCDinit.c:
```



Functions

- void [LCDinit](#) (void)
Initialize the LCD.

2.19.1 Detailed Description

Initialize the LCD.

Definition in file [LCDinit.c](#).

2.19.2 Function Documentation

2.19.2.1 void LCDinit (void)

Initialize the LCD.

Definition at line 32 of file [LCDinit.c](#).

```
{
// 15mS delay after Vdd reaches nnVdc before proceeding with LCD
// initialization
// not always required and is based on system Vdd rise rate
Delay(Delay_15mS_Cnt); // 15ms delay

/* set initial states for the data and control pins */
LCD_DATA &= 0xFF00;
LCD_RW = 0; // R/W state set low
LCD_RS = 0; // RS state set low
LCD_ENABLE = 0; // E state set low

/* set data and control pins to outputs */
LCD_DATATRIS &= 0xFF00;
LCD_RW_TRIS = 0; // RW pin set as output
LCD_RS_TRIS = 0; // RS pin set as output
LCD_ENABLE_TRIS = 0; // E pin set as output

/* 1st LCD initialization sequence */
LCD_DATA &= 0xFF00;
LCD_DATA |= 0x0038;
LCD_ENABLE = 1;
Nop();
Nop();
Nop();
LCD_ENABLE = 0; // toggle E signal
Delay(Delay_5mS_Cnt); // 5ms delay

/* 2nd LCD initialization sequence */
LCD_DATA &= 0xFF00;
LCD_DATA |= 0x0038;
LCD_ENABLE = 1;
Nop();
Nop();
Nop();
LCD_ENABLE = 0; // toggle E signal
Delay_Us(Delay200uS_count); // 200uS delay
```

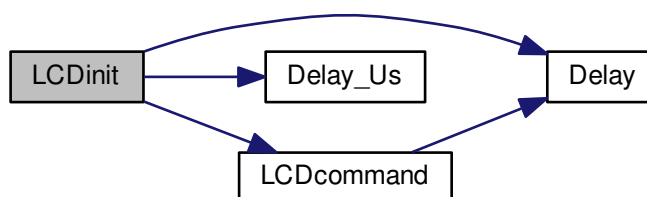
```

/* 3rd LCD initialization sequence */
LCD_DATA &= 0xFF00;
LCD_DATA |= 0x0038;
LCD_ENABLE = 1;
Nop();
Nop();
Nop();
LCD_ENABLE = 0; // toggle E signal
Delay_Us(Delay200uS_count); // 200uS delay

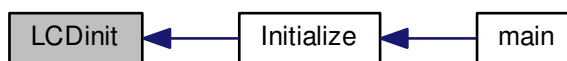
LCDcommand(0x38); // function set
LCDcommand(0x0C); // Display on/off control, cursor blink off (0x0C)
LCDcommand(0x06); // entry mode set (0x06)
}

```

Here is the call graph for this function:



Here is the caller graph for this function:



2.20 LCDinit.c

```

00001
00007 #if defined(__PIC24E__)
00008 #include <p24Exxxx.h>

```

```

00009
00010 #elif defined (__PIC24F__)
00011 #include <p24Fxxxx.h>
00012
00013 #elif defined (__PIC24H__)
00014 #include <p24Hxxxx.h>
00015
00016 #elif defined (__dsPIC30F__)
00017 #include <p30Fxxxx.h>
00018
00019 #elif defined (__dsPIC33E__)
00020 #include <p33Exxxx.h>
00021
00022 #elif defined (__dsPIC33F__)
00023 #include <p33Fxxxx.h>
00024
00025 #endif
00026
00027 #include "lcd.h"
00028 #include "lcd_intern.h"
00029 #include "delay.h"
00030
00032 void LCDinit( void) {
00033     // 15mS delay after Vdd reaches nnVdc before proceeding with LCD
    initialization
00034     // not always required and is based on system Vdd rise rate
00035     Delay(Delay_15mS_Cnt); // 15ms delay
00036
00037     /* set initial states for the data and control pins */
00038     LCD_DATA &= 0xFF00;
00039     LCD_RW = 0; // R/W state set low
00040     LCD_RS = 0; // RS state set low
00041     LCD_ENABLE = 0; // E state set low
00042
00043     /* set data and control pins to outputs */
00044     LCD_DATATRIS &= 0xFF00;
00045     LCD_RW_TRIS = 0; // RW pin set as output
00046     LCD_RS_TRIS = 0; // RS pin set as output
00047     LCD_ENABLE_TRIS = 0; // E pin set as output
00048
00049     /* 1st LCD initialization sequence */
00050     LCD_DATA &= 0xFF00;
00051     LCD_DATA |= 0x0038;
00052     LCD_ENABLE = 1;
00053     Nop();
00054     Nop();
00055     Nop();
00056     LCD_ENABLE = 0; // toggle E signal
00057     Delay(Delay_5mS_Cnt); // 5ms delay
00058
00059     /* 2nd LCD initialization sequence */
00060     LCD_DATA &= 0xFF00;
00061     LCD_DATA |= 0x0038;
00062     LCD_ENABLE = 1;
00063     Nop();
00064     Nop();
00065     Nop();
00066     LCD_ENABLE = 0; // toggle E signal
00067     Delay_Us(Delay200uS_count); // 200uS delay
00068
00069     /* 3rd LCD initialization sequence */
00070     LCD_DATA &= 0xFF00;
00071     LCD_DATA |= 0x0038;
00072     LCD_ENABLE = 1;
00073     Nop();
00074     Nop();
00075     Nop();
00076     LCD_ENABLE = 0; // toggle E signal

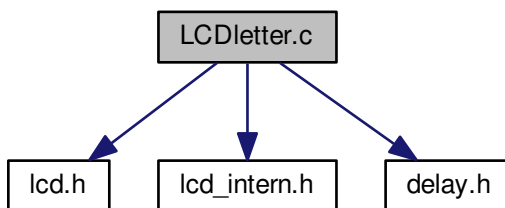
```

```
00077     Delay_Us(Delay200uS_count); // 200uS delay
00078
00079     LCDcommand(0x38); // function set
00080     LCDcommand(0x0C); // Display on/off control, cursor blink off (0x0C)
00081     LCDcommand(0x06); // entry mode set (0x06)
00082 }
```

2.21 LCDletter.c File Reference

Send a character to the LCD.

```
#include "lcd.h" #include "lcd_intern.h" #include "delay.-
h" Include dependency graph for LCDletter.c:
```



Functions

- void [LCDletter](#) (char data)
Send a character to the LCD.

2.21.1 Detailed Description

Send a character to the LCD.

Definition in file [LCDletter.c](#).

2.21.2 Function Documentation

2.21.2.1 void LCDletter (char data)

Send a character to the LCD.

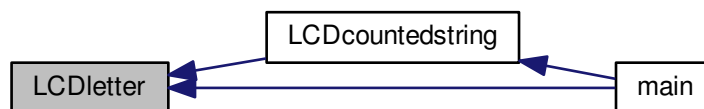
Definition at line 33 of file [LCDletter.c](#).

```
{  
    LCD_RW = 0; // ensure RW is 0  
    LCD_RS = 1; // assert register select to 1  
    LCD_DATA &= 0xFF00; // prepare RD0 - RD7  
    LCD_DATA |= data; // data byte to lcd  
    LCD_ENABLE = 1;  
    Nop();  
    Nop();  
    Nop();  
    LCD_ENABLE = 0; // toggle E signal  
    LCD_RS = 0; // negate register select to 0  
    Delay_Us(Delay200uS_count); // 200uS delay  
    Delay_Us(Delay200uS_count); // 200uS delay  
}
```

Here is the call graph for this function:



Here is the caller graph for this function:



2.22 LCDletter.c

```

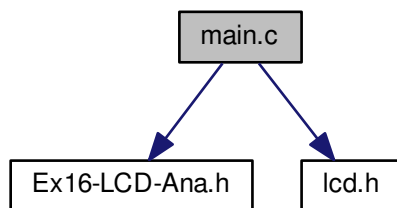
00001
00007 #if defined(__PIC24E__)
00008 #include <p24Exxxx.h>
00009
00010 #elif defined (__PIC24F__)
00011 #include <p24Fxxx.h>
00012
00013 #elif defined (__PIC24H__)
00014 #include <p24Hxxx.h>
00015
00016 #elif defined (__dsPIC30F__)
00017 #include <p30Fxxx.h>
00018
00019 #elif defined (__dsPIC33E__)
00020 #include <p33Exxxx.h>
00021
00022 #elif defined (__dsPIC33F__)
00023 #include <p33Fxxx.h>
00024
00025 #endif
00026
00027 #include "lcd.h"
00028 #include "lcd_intern.h"
00029 #include "delay.h"
00030
00031
00033 void LCDletter( char data )          // subroutine for lcd data
00034 {
00035     LCD_RW = 0; // ensure RW is 0
00036     LCD_RS = 1; // assert register select to 1
00037     LCD_DATA &= 0xFF00; // prepare RD0 - RD7
00038     LCD_DATA |= data; // data byte to lcd
00039     LCD_ENABLE = 1;
00040     Nop();
00041     Nop();
00042     Nop();
00043     LCD_ENABLE = 0; // toggle E signal
00044     LCD_RS = 0; // negate register select to 0
00045     Delay_Us(Delay200uS_count); // 200uS delay
00046     Delay_Us(Delay200uS_count); // 200uS delay
00047 }

```

2.23 main.c File Reference

Mainline for Ex16-LCD-Ana.

#include "Ex16-LCD-Ana.h" #include "lcd.h" Include dependency graph for main.c:



Functions

- [_FICD](#) (ICS_PGD1 &JTAGEN_OFF)
Communicate on PGC1/EMUC1 and PGD1/EMUD1, JTAG is Disabled.
- [_FOSC](#) (POSCMD_XT &FCKSM_CSECMD)
XT Oscillator Mode, Clock switching is enabled, Fail-Safe Clock Monitor is disabled.
- [_FOSCSEL](#) (FNOSC_PRIPLL &IESO_OFF)
Primary Oscillator (XT, HS, EC) w/ PLL, Start up with user-selected oscillator.
- [_FPOR](#) (FPWRT_PWR64)
Power-on reset timer 64 ms.
- [_FWDT](#) (FWDTEN_OFF)
Watchdog timer enabled/disabled by user software.
- int [main](#) (void)
Mainline for Ex16-LCD-Ana.

Variables

- unsigned char [szMessage](#) [4][17]
Table of messages to be displayed.

2.23.1 Detailed Description

Mainline for Ex16-LCD-Ana. This application is intended to show use of the timer and the LCD. A flag is passed from the ISR to the mainline to indicate time to update the display.

A second line of the display contains the message number, to demonstrate LCD cursor positioning.

File: [main.c](#) Author: jjmcd

Created on June 19, 2012, 9:27 AM

Definition in file [main.c](#).

2.23.2 Function Documentation

2.23.2.1 `_FICD (ICS_PGD1 & JTAGEN_OFF)`

Communicate on PGC1/EMUC1 and PGD1/EMUD1, JTAG is Disabled.

2.23.2.2 `_FOSC (POSCMD_XT & FCKSM_CSECMD)`

XT Oscillator Mode, Clock switching is enabled, Fail-Safe Clock Monitor is disabled.

2.23.2.3 `_FOSCSEL (FNOSC_PRIPLL & IESO_OFF)`

Primary Oscillator (XT, HS, EC) w/ PLL, Start up with user-selected oscillator.

2.23.2.4 `_FPOR (FPWRT_PWR64)`

Power-on reset timer 64 ms.

2.23.2.5 `_FWDT (FWDTEN_OFF)`

Watchdog timer enabled/disabled by user software.

2.23.2.6 `int main (void)`

Mainline for Ex16-LCD-Ana.

Blink two LEDs and display a number of messages on the LCD

Pseudocode:


```

Initialize()
Clear the LCD display
do forever
    if the dirty flag is set
        clear the dirty flag
        clear the display
        display the current message
        increment the message number
        display the message number
    if we are at the end of messages
        point to the first message

```

Definition at line 90 of file `main.c`.

```

{
    // Initialize ports and variables
    Initialize();

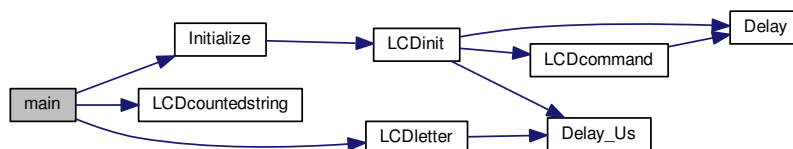
    // Clear the screen
    LCDclear();

    // Display a friendly warning message
    LCDcountedstring((unsigned char *)"In Principio      erat Verbum ",28);

    while (1)
    {
        // If the message needs to be updated
        if ( dirty )
        {
            // Remember we did it
            dirty = 0;
            // Clear the display
            LCDclear();
            // Display the current message
            LCDcountedstring(szMessage[message],16);
            // Point to the next message
            message++;
            // Position cursor to the middle of line 2
            LCDposition( 0x40+5);
            // Display the message number
            LCDletter(0x30+message );
            // If we are at the end of the messages
            if ( message > 3 )
                // point back to the first message
                message = 0;
        }
    }
}

```

Here is the call graph for this function:



2.23.3 Variable Documentation

2.23.3.1 unsigned char szMessage[4][17]

Initial value:

```

{
    "Message One      ",
    "msg num 2        ",
    "Number three     ",
    "I am number four"
}

```

Table of messages to be displayed.

Definition at line 64 of file [main.c](#).

2.24 main.c

```

00001
00018 /*****
00019  * Software License Agreement
00020  *
00021  * GPLv2+
00022  *
00023  *****/
00024
00025
00026 #if defined(__PIC24E__)
00027 #include <p24Exxxx.h>
00028
00029 #elif defined (__PIC24F__)
00030 #include <p24Fxxx.h>
00031
00032 #elif defined(__PIC24H__)
00033 #include <p24Hxxx.h>
00034
00035 #elif defined(__dsPIC30F__)

```

```

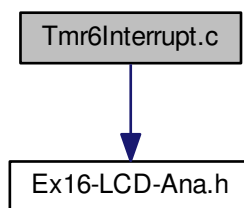
00036 #include <p30Fxxxx.h>
00037
00038 #elif defined (__dsPIC33E__)
00039 #include <p33Exxxx.h>
00040
00041 #elif defined (__dsPIC33F__)
00042 #include <p33Fxxxx.h>
00043
00044 #endif
00045
00046 #define EXTERN
00047 #include "Ex16-LCD-Ana.h"
00048 #include "lcd.h"
00049
00050 // Configuration fuses
00051 //
00053 _FOSCSEL( FNOSC_PRIPLL & IESO_OFF );
00055 _FOSC( POSCMD_XT & FCKSM_CSECMD );
00057 _FWDTP( FWDTPN_OFF );
00059 _FPOR( FPWRT_PWR64 );
00061 _FICD( ICS_PGD1 & JTAGEN_OFF );
00062
00064 unsigned char szMessage[4][17] =
00065 {
00066     "Message One      ",
00067     "msg num 2        ",
00068     "Number three     ",
00069     "I am number four"
00070 };
00071
00073
00090 int main(void)
00091 {
00092     // Initialize ports and variables
00093     Initialize();
00094
00095     // Clear the screen
00096     LCDclear();
00097
00098     // Display a friendly warning message
00099     LCDcountedstring((unsigned char *)"In Principio      erat Verbum ",28);
00100
00101     while (1)
00102     {
00103         // If the message needs to be updated
00104         if ( dirty )
00105         {
00106             // Remember we did it
00107             dirty = 0;
00108             // Clear the display
00109             LCDclear();
00110             // Display the current message
00111             LCDcountedstring(szMessage[message],16);
00112             // Point to the next message
00113             message++;
00114             // Position cursor to the middle of line 2
00115             LCDposition( 0x40+5);
00116             // Display the message number
00117             LCDletter(0x30+message );
00118             // If we are at the end of the messages
00119             if ( message > 3 )
00120                 // point back to the first message
00121                 message = 0;
00122         }
00123     }
00124 }
00125 }

```

2.25 Tmr6Interrupt.c File Reference

Timer 6 interrupt service routine.

`#include "Ex16-LCD-Ana.h"` Include dependency graph for Tmr6Interrupt.c:



Defines

- `#define EXTERN extern`

Functions

- `void __attribute__((__interrupt__, auto_psv))`
Timer 6 Interrupt Service Routine.

Variables

- `int delayCount`
Counter used to delay toggling dirty flag.

2.25.1 Detailed Description

Timer 6 interrupt service routine. Whenever Timer 6 expires, this routine toggles the rightmost 2 LEDs. After 5 interrupts, it sets the dirty flag causing the mainline to display a new message on the LCD.

Definition in file [Tmr6Interrupt.c](#).

2.25.2 Function Documentation

2.25.2.1 void __attribute__((__interrupt__, auto_psv))

Timer 6 Interrupt Service Routine.

Gets executed whenever Timer 6 expires

Pseudocode:

```

Clear timer interrupt flag
Toggle right 2 LEDs (XOR LATA with 3)
increment delayCount
if delayCount > 5
    Set dirty flag
    Reset delay count

```

Definition at line 50 of file [Tmr6Interrupt.c](#).

```

{
    IFS2bits.T6IF = 0;           // Clear timer interrupt flag
                                // This is always the first order of
                                // business in an interrupt routine

    LATA ^= 0x0003;              // Toggle right 2 LEDs
    delayCount++;                // Increment delayCount
    if ( delayCount > 5 )        // Only update display every 5
    {                             // toggles of LEDs
        dirty = 1;               // Set the dirty flag
        delayCount = 0;          // Reset the delayCount
    }
}

```

2.25.3 Variable Documentation

2.25.3.1 int delayCount

Counter used to delay toggling dirty flag.

Definition at line 35 of file [Tmr6Interrupt.c](#).

2.26 Tmr6Interrupt.c

```

00001
00011 #if defined(__PIC24E__)
00012 #include <p24Exxxx.h>
00013
00014 #elif defined (__PIC24F__)
00015 #include <p24Fxxxx.h>
00016
00017 #elif defined(__PIC24H__)
00018 #include <p24Hxxxx.h>
00019
00020 #elif defined(__dsPIC30F__)

```

```
00021 #include <p30Fxxxx.h>
00022
00023 #elif defined (__dsPIC33E__)
00024 #include <p33Exxxx.h>
00025
00026 #elif defined (__dsPIC33F__)
00027 #include <p33Fxxxx.h>
00028
00029 #endif
00030
00031 #define EXTERN extern
00032 #include "Ex16-LCD-Ana.h"
00033
00035 int delayCount;
00036
00038
00050 void __attribute__((__interrupt__, auto_psv)) _T6Interrupt( void )
00051 {
00052     IFS2bits.T6IF = 0;           // Clear timer interrupt flag
00053                                 // This is always the first order of
00054                                 // business in an interrupt routine
00055
00056     LATA ^= 0x0003;             // Toggle right 2 LEDs
00057     delayCount++;               // Increment delayCount
00058     if ( delayCount > 5 )       // Only update display every 5
00059     {                             // toggles of LEDs
00060         dirty = 1;              // Set the dirty flag
00061         delayCount = 0;         // Reset the delayCount
00062     }
00063 }
```