



دانشگاه صنعتی اصفهان
دانشکده برق و کامپیوتر

دستورکار آزمایشگاه طراحی مدارهای واسط

آزمایش چهارم

تهیه کننده: مهران صفایانی

پاییز ۹۵

آشنایی با نمایشگرهای ال سی دی کاراکتری

❖ اهداف :

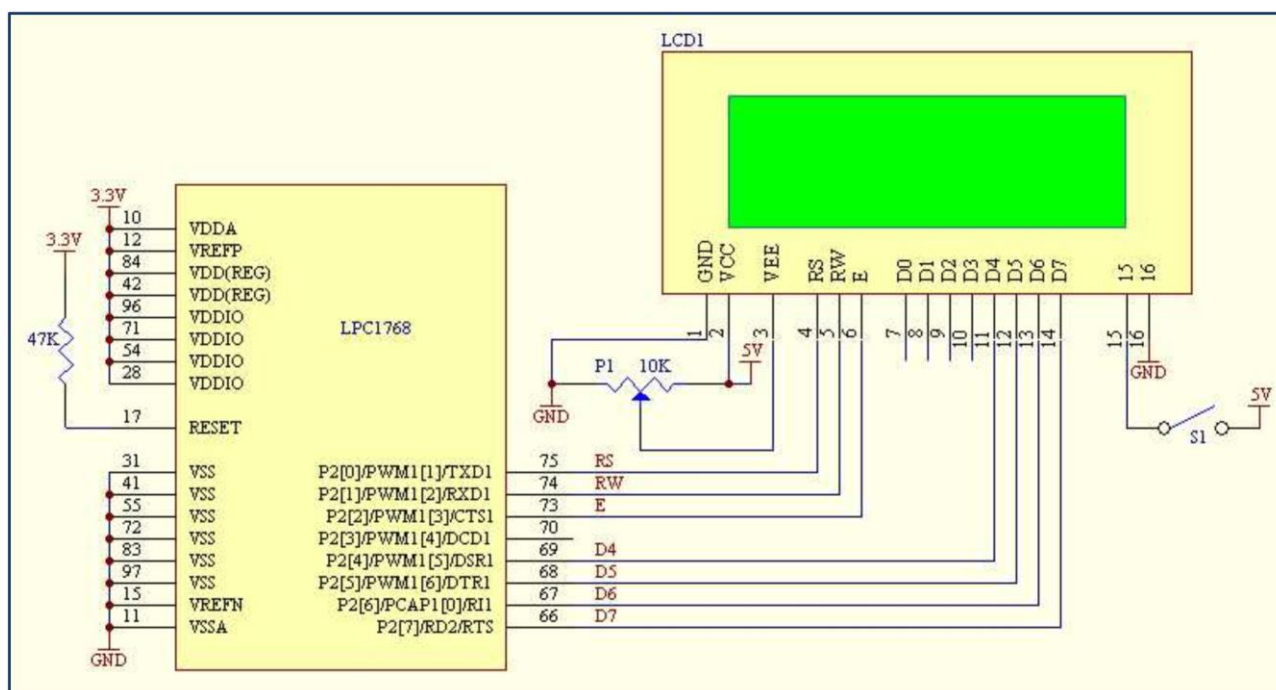
• راه اندازی ال سی دی کاراکتری و کار با آن

❖ مقدمه :

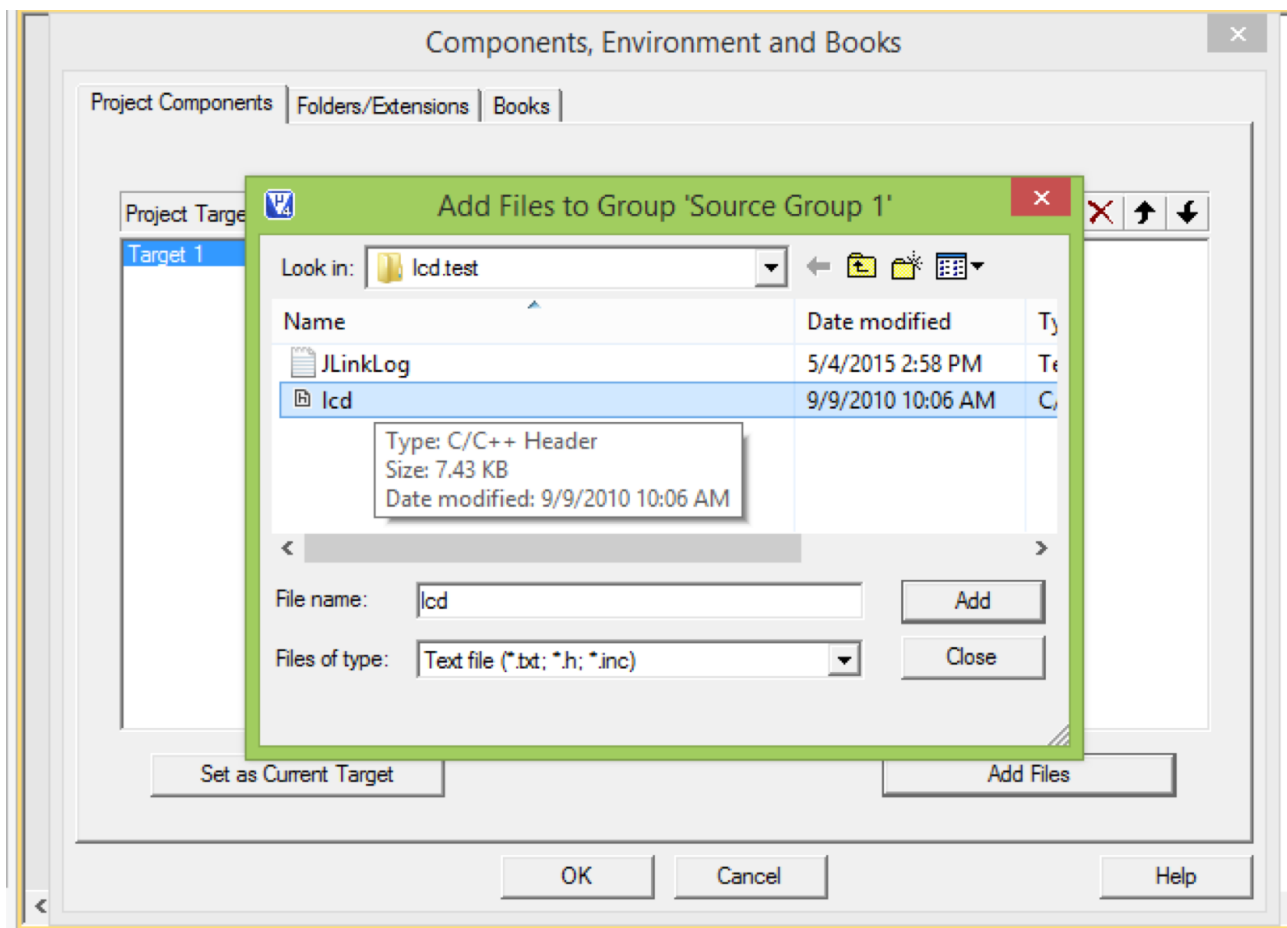
هدف از این آزمایش آشنایی و کار با نمایشگرهای ال سی دی کاراکتری است .

❖ راه اندازی ال سی دی کاراکتری:

Lcd کارکتری یک نمایشگر کریستال مایع است که توسط یک تراشه به نام *HD44780* داریو می شود. این تراشه وظیفه گرفتن اطلاعات از میکروکنترلر و نمایش آن بروی صفحه کریستال مایع را بر عهده دارد .



به منظور استفاده از *LCD* از هدر فایل *LCD.h* که کتابخانه ای شامل توابع کار با *LCD* است استفاده می کنیم. برای استفاده از این کتابخانه باید فایل *LCD.h* در مسیر پروژه خود کپی و سپس این کتابخانه را به پروژه خود اضافه می نماییم .



بعد از آن باید پیکر بندی مربوط به پین ها و پایه های میکرو کنترلر و *LCD* را تعریف کنیم . برای این کار طبق دستور زیر عمل می کنیم :

```
#define LCD_LPC1768
#define LCD_PORT_2
#define LCD_RS 0
#define LCD_RW 1
#define LCD_E 2
#define LCD_DB4 4
#define LCD_DB5 5
#define LCD_DB6 6
#define LCD_DB7 7
```

سپس با استفاده از دستور *include* به کامپایلر اعلام می نماییم قصد استفاده از هدر *LCD* را داریم :

```
#include "lcd.h"
```

❖ آشنایی با هدر فایل lcd.h :

- `lcd_init()` : این تابع جهت آماده سازی میکروکنترلر به منظور ارتباط با `lcd` استفاده می شود که می بایست حتماً در ابتدای برنامه اصلی نوشته شود .
- `lcd_clear()` : از این تابع جهت پاک کردن صفحه نمایش استفاده می شود .
- `lcd_gotoxy(x,y)` : از این تابع جهت آدرس دهی مکان نما استفاده می شود.
- `lcd_putsf()` : از این تابع برای چاپ رشته (String) استفاده می شود.
`lcd_putsf(char flash *str)`
- `lcd_putchar(char c)` : برای چاپ یک کارکتر استفاده می شود.
- `lcd_puts(char* str)` : برای چاپ متغیرهای عددی استفاده می شود.
- `lcd_shift_right(int)` : برای شیفت راست عبارت روی صفحه `lcd` به اندازه عدد داخل آرگومان داخل تابع استفاده می شود.
- `lcd_shift_left(int)` : برای شیفت چپ عبارت روی صفحه `lcd` به اندازه عدد داخل آرگومان داخل تابع استفاده می شود.
- `lcd_command(hex)` : این تابع جهت ارسال فرمان های کنترلی `lcd` استفاده می شود . در جدول زیر کدهای کنترلی مربوط به این `lcd` آورده شده است.

| توضیح عملکرد command | Command Code |
|------------------------------|--------------|
| پاک کردن صفحه نمایش | 0x01 |
| بازگشت به home | 0x02 |
| جا به جایی مکان نما به چپ | 0x04 |
| جا به جایی مکان نما به راست | 0x06 |
| جا به جایی نمایش به چپ | 0x05 |
| جا به جایی نمایش به راست | 0x07 |
| نمایش خاموش و مکان نما خاموش | 0x08 |
| نمایش خاموش و مکان نما روشن | 0x0A |
| نمایش روشن و مکان نما خاموش | 0x0C |
| نمایش روشن و مکان نما روشن | 0x0E |
| مکان نما به صورت چشمک زن | 0x0F |

برنامه نمونه: برنامه ای بنویسید که عبارت *arm* را در مکان (۱۰۱) *lcd* نوشته و عبارت *microcontroller* در سطر دوم بنویسید .

```
#include <LPC17xx.h>
#define LCD_LPC1768
#define LCD_PORT_2
#define LCD_RW      1
#define LCD_E        2
#define LCD_DB4       4
#define LCD_DB5       5
#define LCD_DB6       6
#define LCD_DB7       7
#include "lcd.h"

main() {
    lcd_init();
    lcd_clear();
    lcd_gotoxy(1,1);
    lcd_putsf("ARM");
    lcd_gotoxy(2,1);
    lcd_putsf("microcontroller");
    while (1) {}
}
```

❖ دستور کار:

- I. برنامه ای بنویسید که اسم و فامیلتان را بر روی LCD نوشته و نمایش دهد .
- II. برنامه نوشته شده در قسمت الف را به گونه ای تغییر دهید که متن نوشته شده بر روی LCD به صورت روان از سمت راست LCD خارج شود و پس از خروج کامل آن دوباره از سمت چپ LCD وارد شود.
- III. برنامه ای بنویسید که با زدن یک سوئیچ از ۱ تا ۹۹ به صورت صعودی بشمارد و با زدن یک سوئیچ دیگر به صورت نزولی بشمارد .
- IV. برنامه ی یک ولوم را بنویسید که با فشردن هر بار یک سوئیچ به عدد پیشفرض (صفر) یک واحد افزوده شود و با فشردن سوئیچ دیگر یک واحد کاهش دهد. رنج مجاز عداد صفر تا ۱۰۰ می باشد .