



# گزارش آزمایش شماره ۲

## هدف آزمایش

نمایش یک استرینگ روی LCD و جابجایی متن روی آن به کمک دکمه های موجود

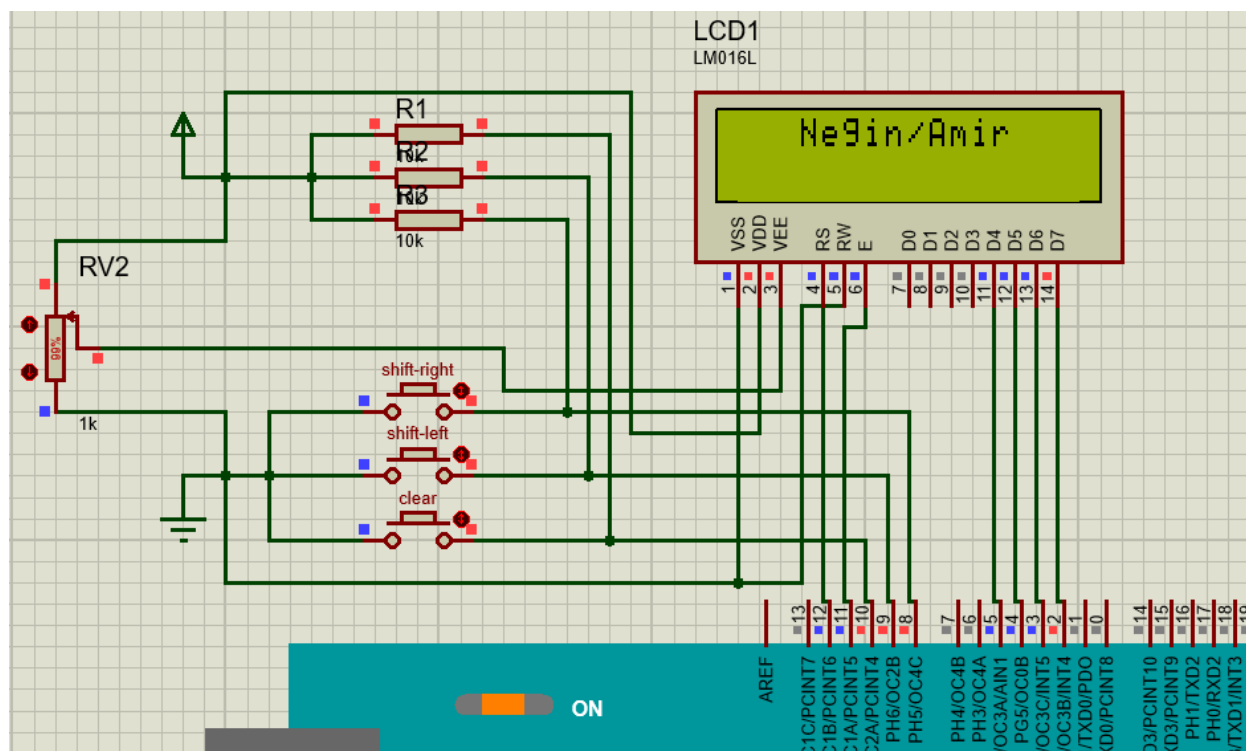
## پیاده سازی مدار در proteus

به ۳ دکمه نیاز داریم، در نتیجه به ازای هرکدام از آنها به یک مقاومت ۱۰ کیلو اهمی نیز احتیاج داریم. پیاده‌سازی کلیدها را بصورت active low انجام می‌دهیم تا در واقع وقتی دکمه فشرده شد مقدار منطقی آن ۰ شود. Pin های ۸ تا ۱۰ برد را به این دکمه‌ها اختصاص می‌دهیم. در شکل پایین صفحه کنار هر دکمه نوشته شده که هر کدام چه کاری انجام می‌دهد.

برای LCD، از LM016L استفاده می‌کنیم و مطابق جدول زیر pin های آن را به برد آردوئینو متصل می‌کنیم:

Arduino board pin	LCD pin
12	RS
11	EN
5	D4
4	D5
3	D6
2	D7

شکل مدار به فرم زیر خواهد بود:





## گزارش آزمایش شماره ۲

## کد در Arduino IDE

```
#include <LiquidCrystal.h>
#define RS 12
#define EN 11
#define D4 5
#define D5 4
#define D6 3
#define D7 2
#define DELAY_PERIOD 500
LiquidCrystal lcd(RS, EN, D4, D5, D6, D7);
int btn1 = 1;
int btn2 = 1;
int btn3 = 1;

void setup() {
    // set up the LCD's number of columns and rows:
    lcd.begin(16, 2);
    lcd.setCursor(0,0);
    lcd.clear();
    lcd.print("Negin/Amir");
    // initializing buttons
    pinMode(8 , INPUT);
    pinMode(9 , INPUT);
    pinMode(10, INPUT);
}

void loop() {
    btn1 = digitalRead(8);
    btn2 = digitalRead(9);
    btn3 = digitalRead(10);
    if (btn3 == LOW) { // clear screen
        lcd.clear();
        delay(DELAY_PERIOD);
    }else if (btn2 == LOW){ // shift left
        lcd.scrollDisplayLeft();
        delay(DELAY_PERIOD);
    }else if (btn1 == LOW){ // shift right
        lcd.scrollDisplayRight();
        delay(DELAY_PERIOD);
    }
}
```



## گزارش آزمایش شماره ۲

### توضیح کد

در بالای کد تعداد مقادیر ثابت define شده اند که شماره پورت های مورد نظر، مقادیر رشته هایی که قرار است نمایش داده شوند و همچنین میزان تاخیر را شامل میشوند.

در قسمت setup علاوه بر مشخص کردن pin های مربوط به دکمه ها، مقدار اولیه موجود روی LCD را روی آن پرینت میکنیم. در ابتدا پس از قرار دادن موقعیت cursor به کمک تابع setCursor، جهت پاک کردن صفحه از محتوای احتمالی قبلی clear میکنیم و سپس اسممان را print میکنیم.

در تابع loop هر بار چک میکنیم کدام دکمه ها فشرده شده اند (مقدار ۰ منطقی). دو دکمه‌ی اول برای شیفت دادن به سمت راست و چپ میباشند. برای این کار از تابع آماده‌ی scrollDisplayLeft برای شیفت به چپ و از scrollDisplayRight برای شیفت به راست استفاده میکنیم. نکته قابل ذکر در این دو مورد این است که اگر دکمه را مدت کوتاهی فشار دهیم ممکن است به جای یک کارکتر، چندین کارکتر شیفت پیدا کند. دلیل این است که ما هرقدر هم بتوانیم پس از فشردن دکمه سریع دست خود را از روی آن برداریم، باز هم سرعتی کم تر از سرعت پردازنده خواهیم داشت. فلسفه‌ی delay به کار رفته در تابع loop جلوگیری از همین مشکل است.

همچنین دکمه سوم برای پاک کردن LCD استفاده میشود. برای این منظور باز هم از تابع clear استفاده میکنیم.