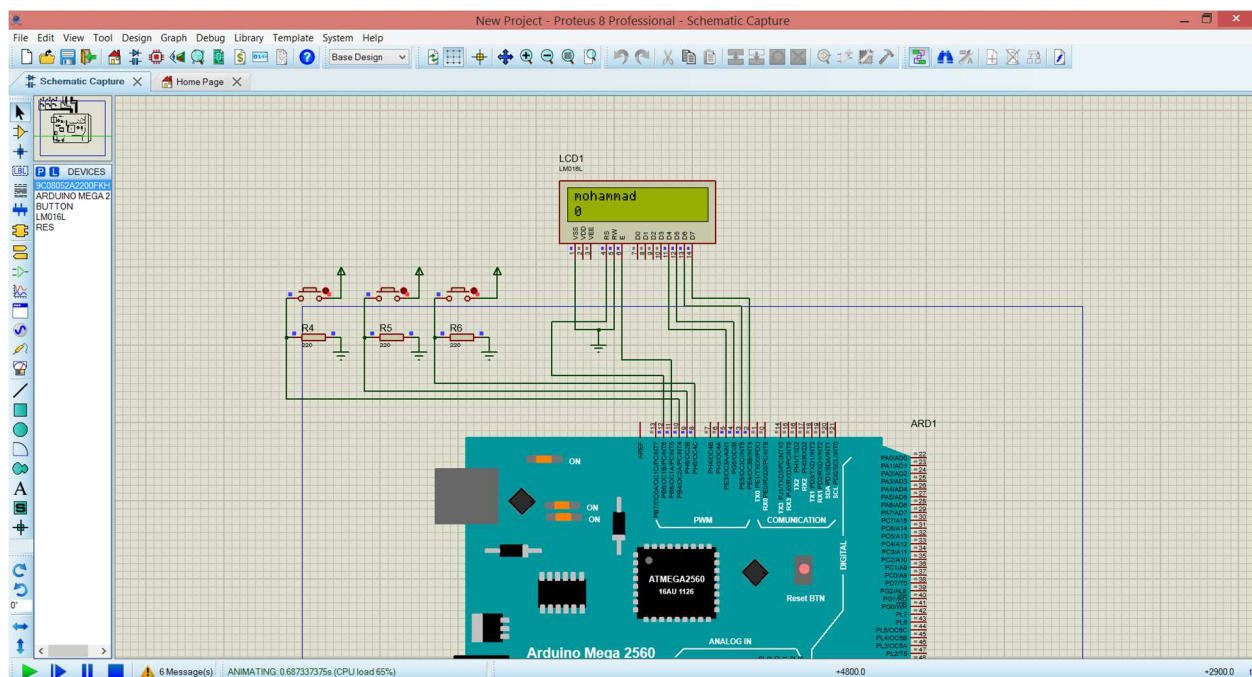


mohammad kazemi

9731050

roham zende del

9731088



همان طور که در مدار بالا مشاهده می کنید باتن ها به صورت active high بسته شده اند.

VSS, RW هم به زمین وصل شده اند.

E (پایه ی Enable) و RS (انتخاب رجیستر) هم به پایه هایی که در کد برنامه آردوینو ذکر شده وصل شده اند.

همین طور پایه های دیتا (D4 تا D7) هم به پایه های که در برنامه آردوینو ذکر شده وصل شدند.

## پایه های LCD مورد استفاده

شماره پایه	سمبول	نحوه اتصال پایه
1	Vss	اتصال به زمین
2	Vdd	اتصال به +5V
3	VEE یا Vo	تنظیم کنتراست LCD
4	RS	کنترل رجیستر
5	RW	انتخاب مد خواندن یا نوشتن
6	E	فعال سازی LCD
7 – 14	D0 - D7	گذرگاه 8 تایی اطلاعات و دستورالعمل
15، 16	کاتد LED، آند LED	آند و کاتد LED پس زمینه



LCD LM016L

## کد آردوینو از 2

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("mohammad");
}
```

ابتدا کتاب خانه ی LiquidCrystal.h را به برنامه اضافه میکنیم.

6 مقدار int تعریف کرده و برای جایگذاری آن ها در ساختن یک شی از کلاس LiquidCrystal استفاده می کنیم ( که در واقع برای تنظیم پایه های LCD از آن ها استفاده می کنیم.

برای معرفی LCD و مشخص کردن ابعاد مورد استفاده از تابع زیر استفاده می کنیم.

lcd.begin(cols, rows)

همچنین برای نمایش کاراکتر در LCD از تابع زیر استفاده می کنیم.

lcd.print(data)

```

void loop() {
  // buttons
  int button1 = digitalRead(8);
  int button2 = digitalRead(9);
  int button3 = digitalRead(10);

  if(button1 == HIGH){
    lcd.scrollDisplayLeft();
    delay(200);
  }
  if(button2 == HIGH){
    lcd.scrollDisplayRight();
    delay(200);
  }
  if (button3 == HIGH){
    lcd.clear();
  }

  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis() / 1000);
}

```

3 باتن تعریف می کنیم.

و با سه دستور شرطی چک می کنیم که اگر هر باتن فشرده شود کدی که داخل شرط مربوط به فشرده شدن آن باتن است اجرا شود.

اگر باتن شماره 1 فشرده شود محتوای نوشته شده در LCD یک واحد به سمت چپ انتقال داده میشود.

`lcd.ScrollDisplayLeft()`

اگر باتن شماره 2 فشرده شود محتوای نوشته شده در LCD یک واحد به سمت راست انتقال داده میشود.

`lcd.ScrollDisplayRight()`

اگر باتن شماره 3 فشرده شود صفحه نمایش LCD پاک می شود.

`lcd.clear()`

توسط تابع **setcursor** یک موقعیت خاص در ال سی دی برای مکان نما انتخاب می‌کنیم.

```
lcd.setCursor(col, row)
```

## توابع دیگر در کتابخانه LiquidCrystal

توسط تابع **home** مکان نما در بالا سمت چپ ال سی دی قرار می‌گیرد.

```
lcd.home()
```

از تابع **write** برای نوشتن کارکتر در ال سی دی استفاده می‌شود.

```
lcd.write(data)
```

از تابع **cursor** برای نمایش به صورت زیر خط **underscore** استفاده می‌شود. در واقع یک خط زیر کاراکتر کشیده می‌شود. همچنین برای عدم نمایش **underscore** از تابع **nocursor** استفاده می‌کنیم.

```
lcd.cursor()
```

```
lcd.noCursor()
```

توسط تابع **blink** مکان نما به صورت چشمک زن در ال سی دی نمایش داده می‌شود.

همچنین توسط تابع **noblink** از حالت چشمک زن خارج می‌شود.

```
lcd.blink()
```

```
lcd.noBlink()
```

توسط تابع **display** ال سی دی روشن و همچنین توسط تابع **nodisplay** ال سی دی خاموش می‌شود.

```
lcd.display()
```

```
lcd.noDisplay()
```