

آزمایش ۳ ریز پردازنده

محمد سپهر توکلی - ۹۸۳۱۱۱۱

محمد چوپان - ۹۸۳۱۱۲۵

پیش گزارش :

مشخصات فنی ماژول نمایشگر LCD کاراکتری 16x2 :

قابلیت نمایش دو سطر و 16 ستون کاراکتر را دارد. 16 پایه دارد.

پایه های 15 و 16 برای backlight نمایشگر است.

پایه های D0 تا D7 = پایه های دیتا (8 بیتی) هستند.

پایه VSS پایه ی اتصال به زمین است.

پایه VCC پایه ی تغذیه ی 5 ولتی است.

پایه VE کنتراست نمایش کاراکتر ها و پس زمینه را تنظیم میکند.

پایه RS(register select) برای انتخاب بین رجیستر ها است.

پایه E برای enable کردن نمایشگر است.

پایه RW برای خواندن و نوشتن روی نمایشگر است.

دلیل استفاده از پتانسیومتر :

دلیل استفاده از پتانسیومتر در مدار این است که پایه ی VE یک ولتاژ به عنوان ورودی میگیرد تا با توجه به آن کنتراست نمایشگر را مشخص کند. پس برای آنکه بتوانیم کنتراست را تغییر دهیم باید ولتاژ ورودی آن را تغییر دهیم و برای این که بتوانیم ولتاژ را در مدار تغییر دهیم از یک پتانسیومتر استفاده میکنیم و پایه ی وسط آن را به VE متصل میکنیم.

توابع مورد نیاز از کتابخانه LiquidCrystal:

1.LiquidCrystal :

این سازنده یک متغیر از نوع liquidCrystal ایجاد می کند.

پارامتر های ورودی :

Rs شماره پینی که از آردوینو به پایه rs متصل است

rw شماره پینی که از آردوینو به پایه rw متصل است

enable شماره پینی که از آردوینو به پایه E متصل است

d0 تا d7 شماره پین هایی که از آردوینو به پایه های دیتا متصل شده اند هستند.

Syntax

LiquidCrystal(rs, enable, d4, d5, d6, d7)

LiquidCrystal(rs, rw, enable, d4, d5, d6, d7)

LiquidCrystal(rs, enable, d0, d1, d2, d3, d4, d5, d6, d7)

LiquidCrystal(rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7)

2.begin :

ارتباط با lcd را شروع می کند. ابعاد صفحه نمایش را به عنوان ورودی می گیرد ، قبل از سایر متد ها باید فراخوانی شود.

Syntax

lcd.begin(cols, rows)

3.clear :

صفحه lcd را پاک میکند یعنی اگه پیکسلی روشن باشد آن را خاموش میکند و کرسر را به مختصات (0,0) نمایشگر می برد.

Syntax

`lcd.clear()`

4. setCursor :

مکان کرسر را مشخص میکند که در کدام سطر و ستون قرار گیرد. ورودی های آن سطر و ستونی است که میخواهیم کرسر به آنجا منتقل شود.

Syntax

`lcd.setCursor(col, row)`

5.write :

کاراکتری را روی lcd مینویسد و تعداد بایت های نوشته شده را باز میگرداند. ورودی آن دیتایی است که میخواهیم نوشته شود.

Syntax

`lcd.write(data)`

6.print :

text را روی lcd مینویسد و تعداد بایت های نوشته شده را باز میگرداند. ورودی اصلی آن دیتاییست که میخواهیم نوشته شود و یک ورودی optional دارد که مبنای اعدادی که نوشته میشوند را مشخص میکند.

Syntax

`lcd.print(data) lcd.print(data, BASE)`

7. noDisplay :

نمایشگر lcd را خاموش میکند (نوشته ی روی آن از بین نمی رود و فقط دیده نمی شود)

`lcd.noDisplay()`

8. scrollDisplayLeft

تکست و کرسر را یکی به چپ منتقل میکند.

Syntax

`lcd.scrollDisplayLeft()`

9. autoscroll :

automatic scrolling را برای lcd روشن میکند. با آمدن کاراکتر جدید، کاراکترهای قبلی یکی به جلو رانده میشوند.

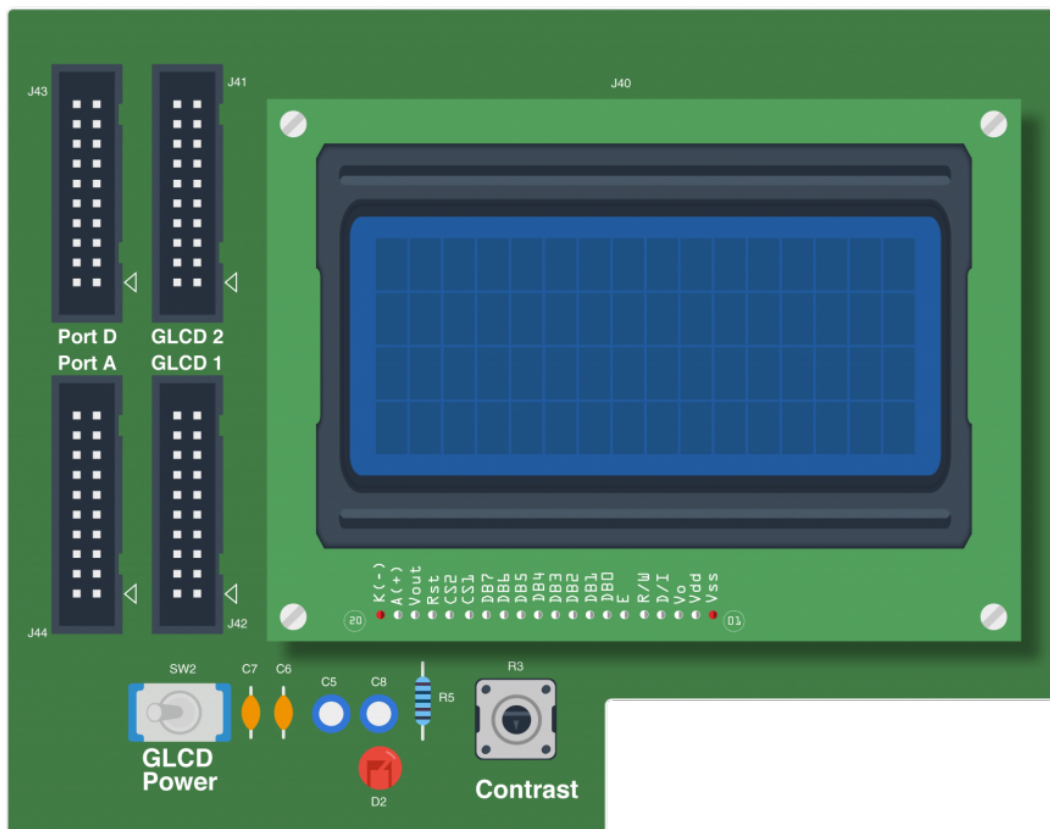
Syntax

`lcd.autoscroll()`

Graphical LCD

The graphical LCD has a 128×64 resolution and can be used to display monochrome images, custom texts in different fonts etc. Graphical LCD uses two ports, J41 and J42. By default, we can connect it to J43 (Port D) and J44 (Port A).

The switch SW2 should be turned ON to use the Graphical LCD.



شرح آزمایش :

برای بستن مدار، ابتدا برد ATmega2560 را انتخاب کرده و keypad و LCD را به پین های مشخص شده متصل کرده و مقاومت و پتانسیومتر مورد نیاز را نیز به برد متصل میکنیم. سر مثبت منبع ولتاژ را به VCC برد و سر منفی آن را به زمین وصل می کنیم.

برای برنامه ریزی برد ۴ فایل در نظر می گیریم هر کدام برای موارد خواسته شده در دستورکار.

برنامه ریزی را در Arduino ide انجام می دهیم و فایل کامپایل شده را روی مدار پروتیوس اجرا می کنیم :

مورد اول :

```
#include <LiquidCrystal.h>
#include <Keypad.h>

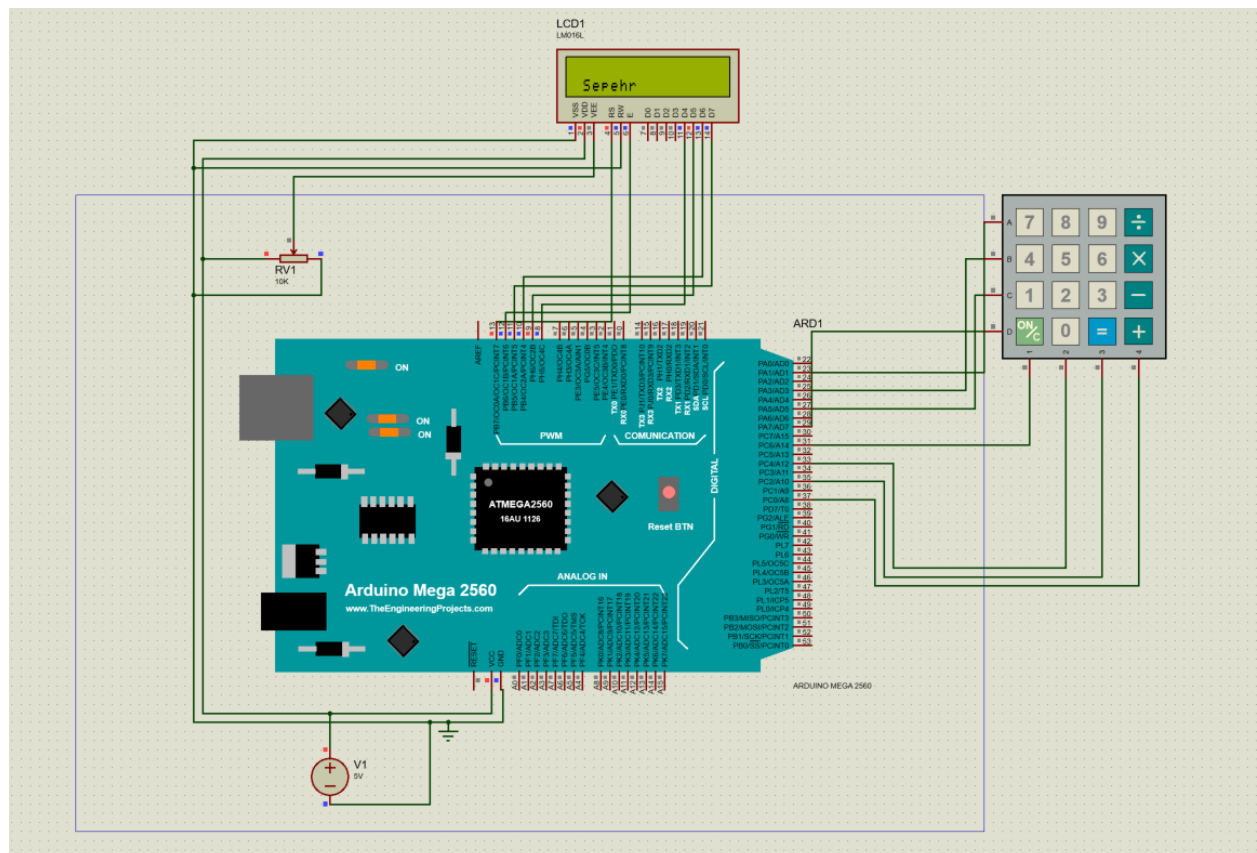
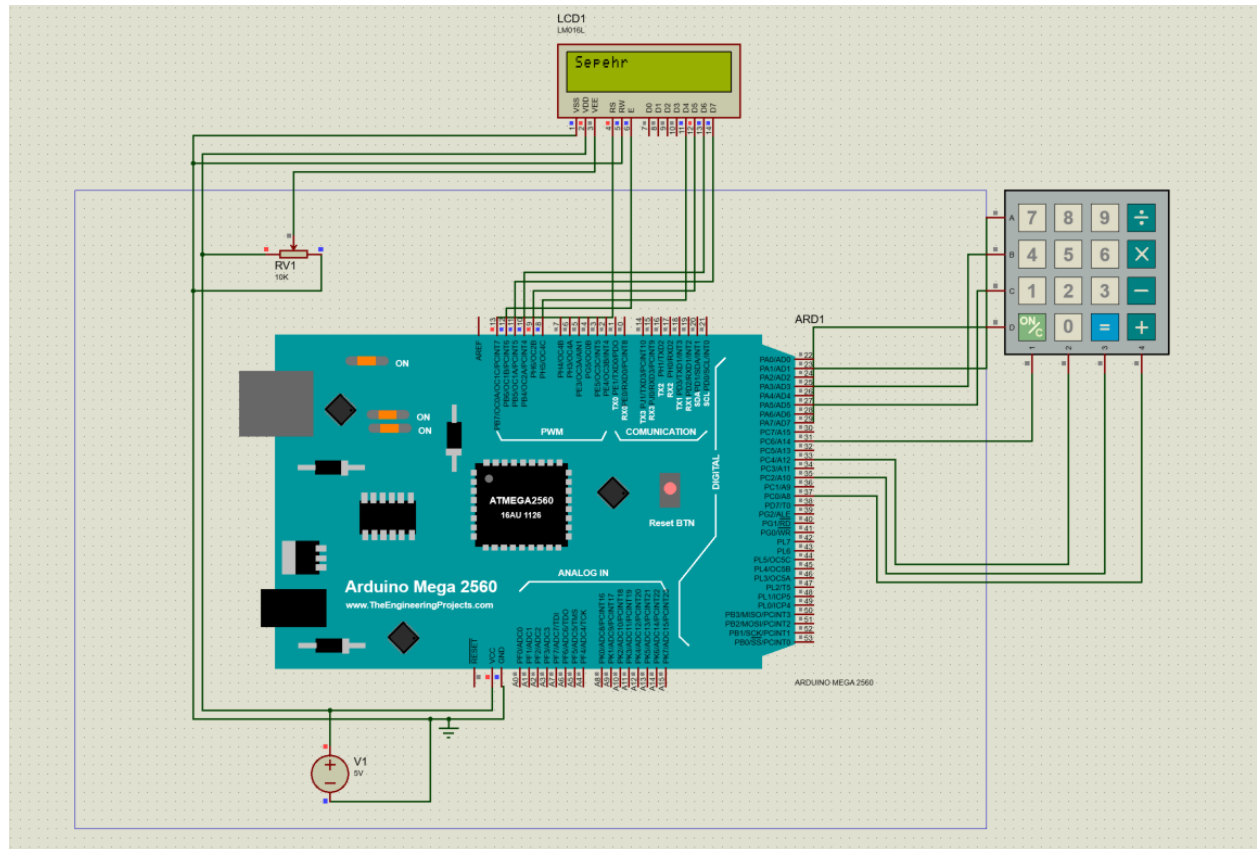
#define REGISTER_SELECT 13
#define ENABLE 12
#define D4 8
#define D5 9
#define D6 10
#define D7 11

LiquidCrystal lcd(REGISTER_SELECT, ENABLE, D4, D5, D6, D7);

//Shift Name
const String name = "Sepehr";
int nameLen = name.length();
int cursorRow=0;
int lastPosition=0;

void setup() {
    lcd.begin(16, 2);
    lcd.clear();
}

void loop() {
    if(lastPosition + nameLen <= 16){
        lcd.setCursor(lastPosition, cursorRow);
        lcd.print(name);
        delay(1000);
        lcd.clear();
        lastPosition ++;
    }
    else if(cursorRow == 0){
        cursorRow = 1;
        lastPosition = 0;
    }
    else if(cursorRow == 1){
        cursorRow = 0;
        lastPosition = 0;
    }
}
```



```
#include <LiquidCrystal.h>
#include <Keypad.h>

#define REGISTER_SELECT 13
#define ENABLE 12
#define D4 8
#define D5 9
#define D6 10
#define D7 11

LiquidCrystal lcd(REGISTER_SELECT, ENABLE, D4, D5, D6, D7);

//Keypad
const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
char keys[ROWS][COLS] = {
  {'7','8','9','/'},
  {'4','5','6','*'},
  {'1','2','3','-'},
  {'0','0','=','+'}
};
byte rowPins[ROWS] = {23, 25, 27, 29};
byte colPins[COLS] = {31,33, 35,37};

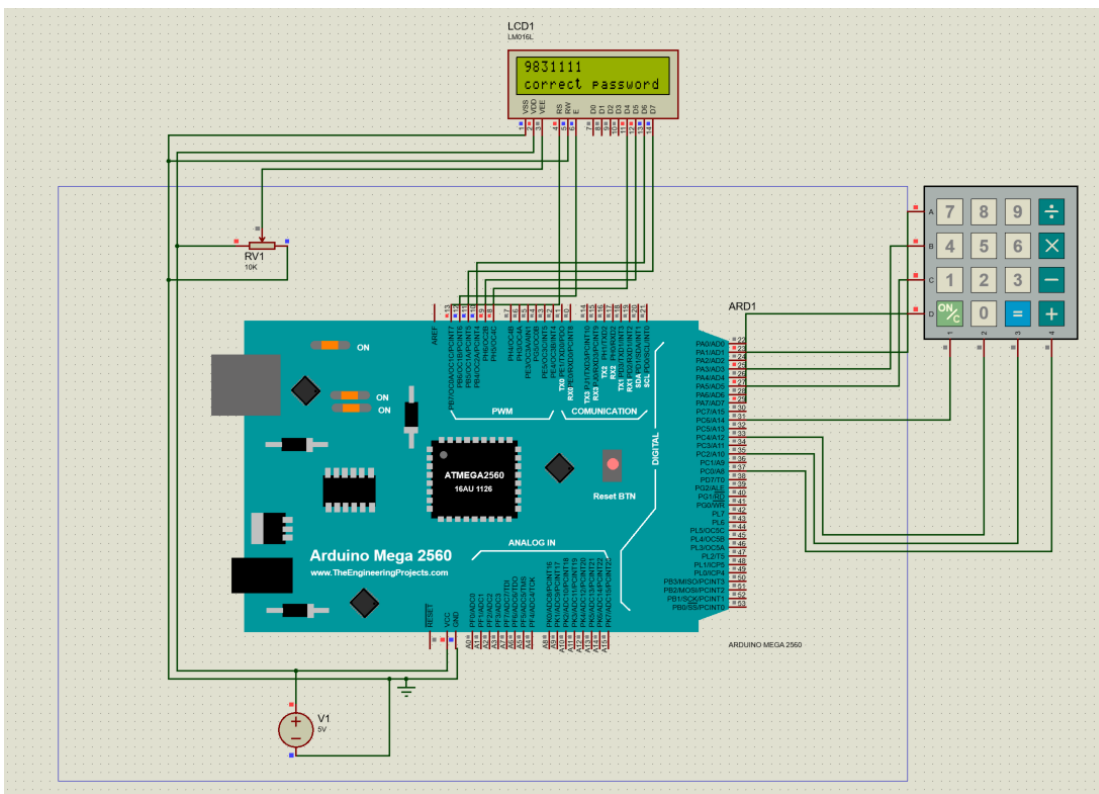
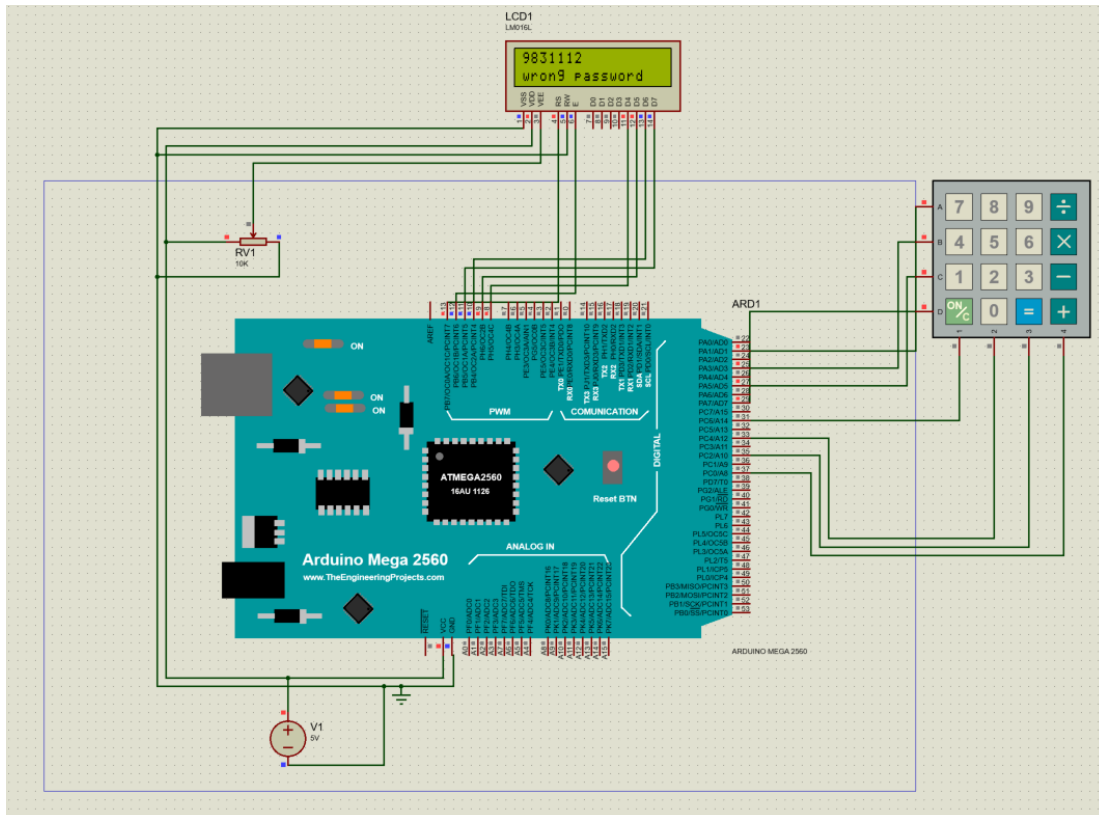
//make a new keypad object
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

char correctPassword[] = "98311111";
char enteredPassword[16];

void setup() {
  lcd.begin(16, 2);
  lcd.clear();
}
```



```
void loop() {  
  char key = keypad.getKey();  
  lcd.setCursor(0, 0);  
  if(strcmp(enteredPassword, NULL) == 0)  
    enteredPassword[0] = key;  
  else if(key != '*'){  
    strncat(enteredPassword, &key, 1);  
    lcd.print(enteredPassword);  
  }  
  else{  
    lcd.setCursor(0, 1);  
    if(strcmp(enteredPassword, correctPassword) == 0)  
      lcd.print("correct password");  
    else  
      lcd.print("wrong password");  
  }  
}
```



مورد سوم :

```
#include <LiquidCrystal.h>
#include <Keypad.h>

#define REGISTER_SELECT 13
#define ENABLE 12
#define D4 8
#define D5 9
#define D6 10
#define D7 11

LiquidCrystal lcd(REGISTER_SELECT, ENABLE, D4, D5, D6, D7);

//Keypad
const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
char keys[ROWS][COLS] = {
  {'7','8','9','/'},
  {'4','5','6','*'},
  {'1','2','3','-'},
  {'0','0','=','+'}
};
byte rowPins[ROWS] = {23, 25, 27, 29};
byte colPins[COLS] = {31,33, 35,37};

//make a new keypad object
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

//calculator
int num1;
int num2;
int res;

char str[16];
char str1[16];
char str2[16];

char operand;
```

```

void setup() {
    lcd.begin(16, 2);
    lcd.clear();
}

void loop() {
    char key = keypad.getKey();
    lcd.setCursor(0, 0);

    if(strcmp(str, NULL) == 0)
        str[0] = key;
    else if(key != '='){
        strncat(str, &key, 1);
        lcd.print(str);
        if(key == '+' || key == '-' || key == '/' || key == '*'){
            if(operand == NULL){
                operand = key;
                num1 = atoi(str1);
            }
        }
    }
    else{
        if(operand == NULL){
            if(strcmp(str1, NULL) == 0)
                str1[0] = key;
            else
                strncat(str1, &key, 1);
        }
        else{
            if(strcmp(str2, NULL) == 0)
                str2[0] = key;
            else
                strncat(str2, &key, 1);
        }
    }
}

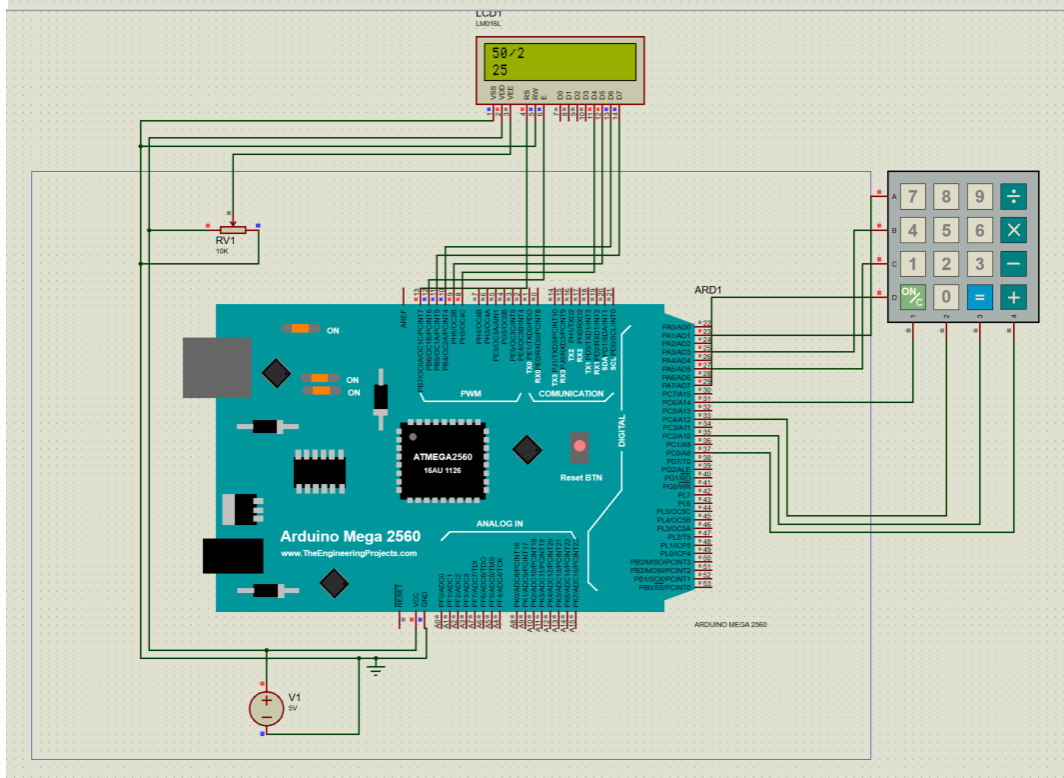
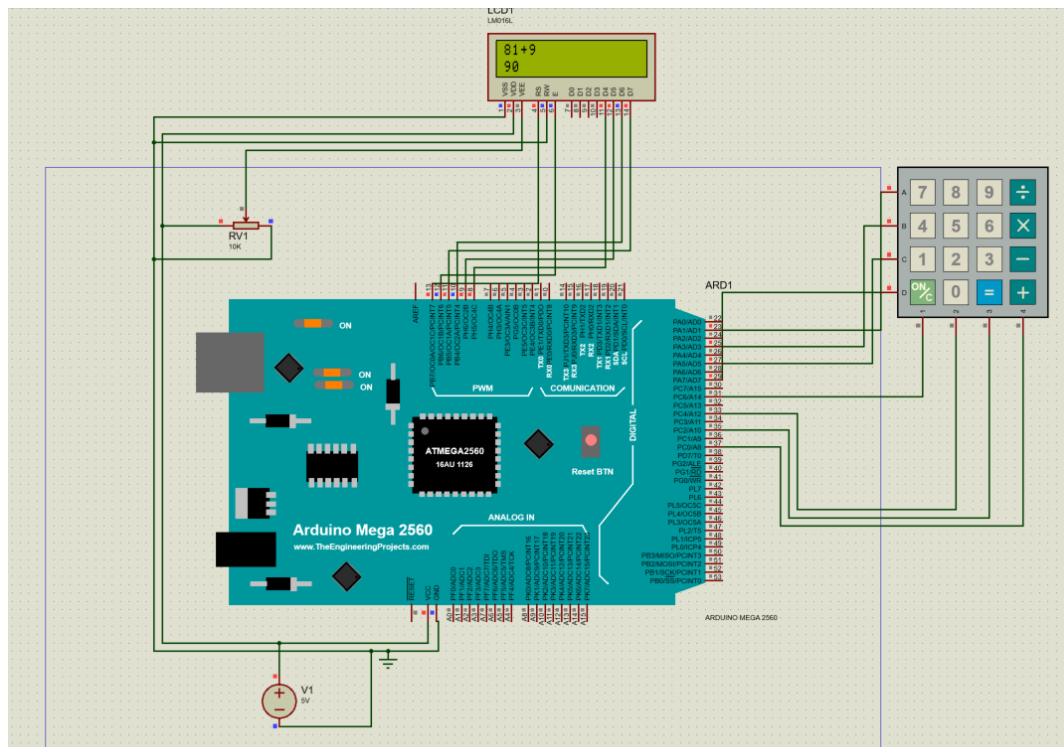
```

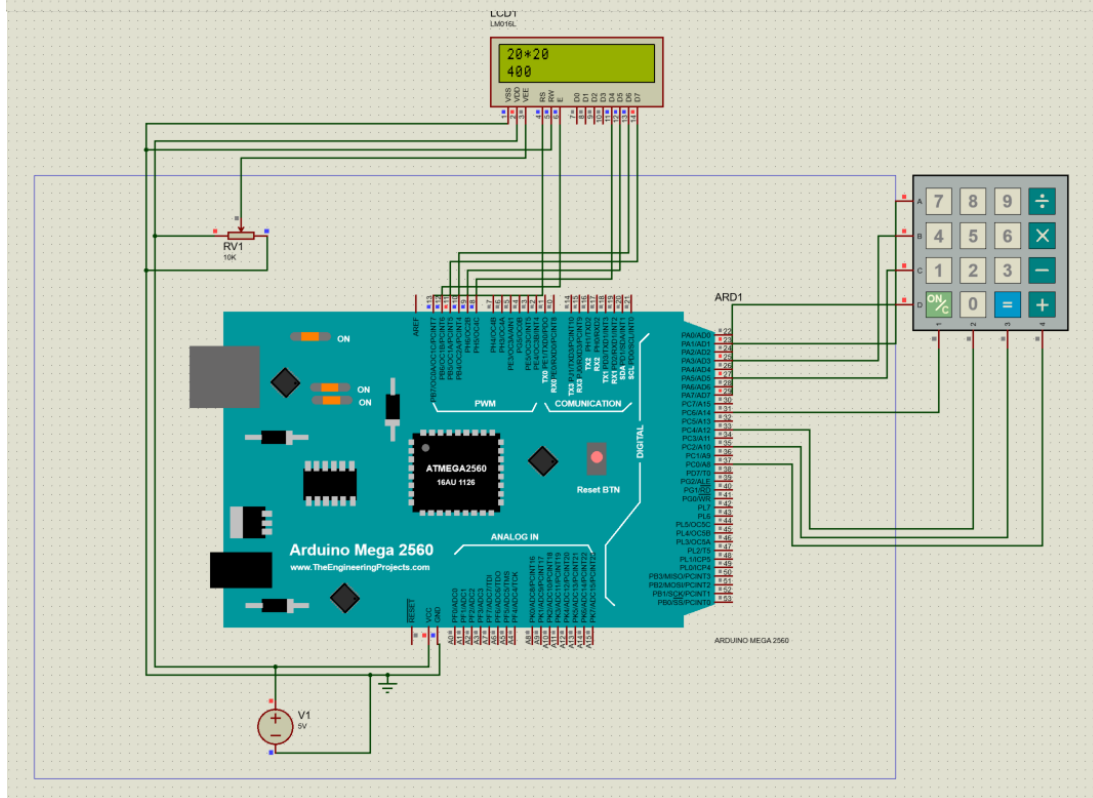
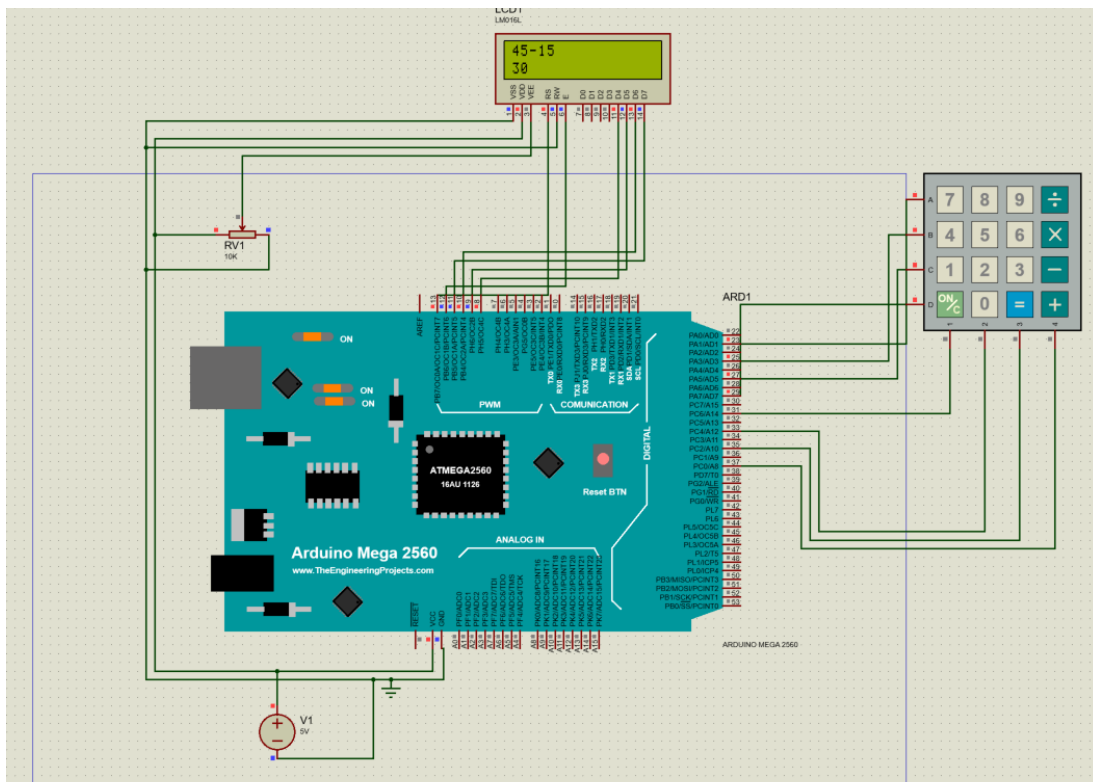
```
else if(key == '='){
    num2 = atoi(str2);

    switch(operand){
        case '+':
            res = num1 + num2;
            break;
        case '-':
            res = num1 - num2;
            break;
        case '*':
            res = num1 * num2;
            break;
        case '/':
            res = num1 / num2;
            break;
    }

    lcd.setCursor(0, 1);
    lcd.print(res);
}

}
```





```

#include <LiquidCrystal.h>

#define REGISTER_SELECT 13
#define ENABLE 12
#define D4 8
#define D5 9
#define D6 10
#define D7 11

LiquidCrystal lcd(REGISTER_SELECT, ENABLE, D4, D5, D6, D7);

int lastPosition = 0;
int cursorRow = 0;

//shiftChar
char shiftChar = 'S';

void setup() {
  lcd.begin(16, 2);
  lcd.clear();
}

void loop() {
  if(lastPosition >= 15){
    lcd.clear();
    lastPosition = 0;
    if(cursorRow == 0){
      cursorRow = 1;
      lcd.clear();
    }
    else{
      cursorRow = 0;
      lcd.clear();
    }
  }
  lcd.setCursor(0, cursorRow);
  lcd.print(shiftChar);
  if(lastPosition != 0){
    lcd.scrollDisplayRight();
  }
  lcd.setCursor(0, cursorRow);
  lcd.print(shiftChar);
  if(lastPosition != 0){
    lcd.scrollDisplayRight();
  }
  delay(1000);
  lastPosition ++;
}

```

