

به نام خدا

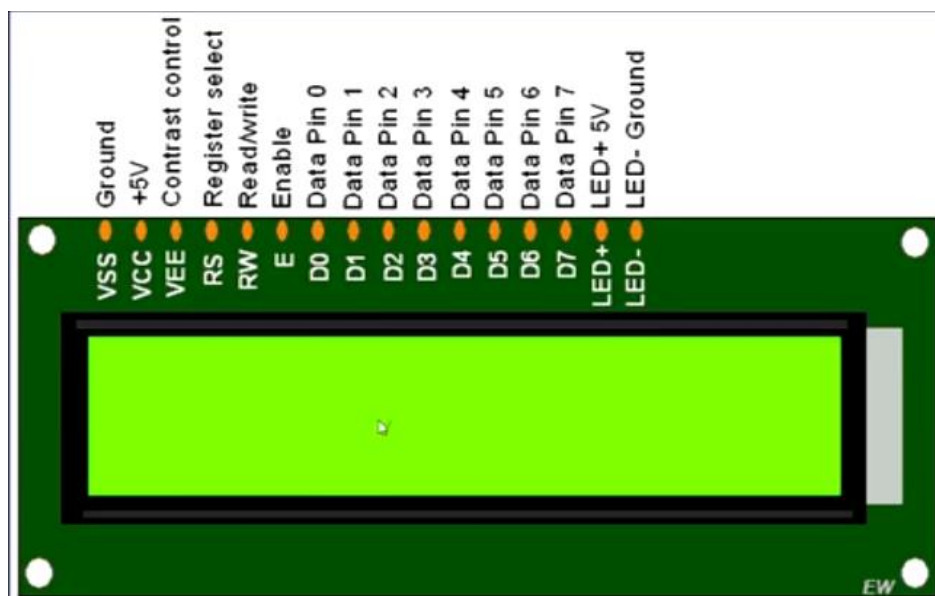
10/18/2021

گزارش آزمایش شماره 3

آزمایشگاه ریزپردازنده و زبان اسمبلی

محمد جواد زندیه , ابوالفضل بکیاسای کیوی
دانشگاه صنعتی امیرکبیر دانشکده مهندسی کامپیوتر

- مشخصات فنی ماژول نمایشگر ال سی دی کاراکتری 16x2 و دلیل استفاده از پتانسیومتر در مدار:



در هر یک از 2 سطری که دارد می تواند 16 کاراکتر را نمایش دهد. هر کاراکتر در یک ماتریس 5x7 نمایش داده می شود. این lcd دارای دو عدد ثبات با نام های data , command می باشد.

پایه های این lcd در بالا نمایش داده شده اند که دو پایه LED+ 5v و LED- Ground به 5 و 0 ولت متصل می شوند. پایه های D0 تا D7 برای انتقال و دریافت داده می باشند (8 Data Pins). همچنین یک پایه فعال ساز به نام Enable دارد که میتوان مشخص کرد که lcd فعال یا غیرفعال باشد. پایه Register select برای انتخاب یکی از دو ثبات که اشاره کردیم یعنی data , command می باشد. دو پایه هم به نام های vcc, ground دارد که کاربردشان مشخص است.

علت استفاده از پتانسیومتر در مدار را میتوان برای تنظیم contrast صفحه استفاده کرد یعنی اختلاف رنگ پیکسل های روشن و خاموش را تنظیم می کند.



- تعریف مختصر توابع مورد نیاز از کتابخانه LiquidCrystal :

LiquidCrystal()

یک متغیر از جنس LiquidCrystal می سازد. صفحه نمایش می تواند توسط 4 یا 8 خط داده کنترل شود.

`begin()`

پیش از صدا زدن هر کتابخانه دیگر نیاز است این کتابخانه صدا زده شود تا رابط را به صفحه lcd مشخص کند و تعداد سطر و ستون lcd را نیز به عنوان ورودی می گیرد (در اینجا به صورت 2, 16 می باشد)

`clear()`

صفحه lcd را پاک می کند و موقعیت cursor را روی 0,0 یعنی بالا چپ ست می کند.

`setCursor()`

موقعیت cursor را روی سطر و ستون ورودی داده شده قرار می دهد و موقعی که بخواهیم بنویسیم روی lcd از آنجا شروع به نوشتن می کند.

`write()`

یک کاراکتر را روی صفحه lcd نمایش می کند.

`print()`

یک متن را روی صفحه lcd نمایش می کند.

`noDisplay()`

صفحه نمایش را خاموش می کند، بدون آنکه متنی که در حال نمایش است را از دست بدهد.

`scrollDisplayLeft()`

محتوای نمایش داده شده روی lcd و یا cursor را به اندازه یک اسپیس به چپ شیفت می دهد.

`autoscroll()`

پیمایش خودکار LCD را روشن می کند. این باعث می شود که هر کاراکتر خروجی در صفحه نمایش ، کاراکترهای قبلی را یک واحد به جلو منتقل دهد.

اطلاعات بیشتر از کتابخانه های گفته شده:

[LiquidCrystal Library](#)

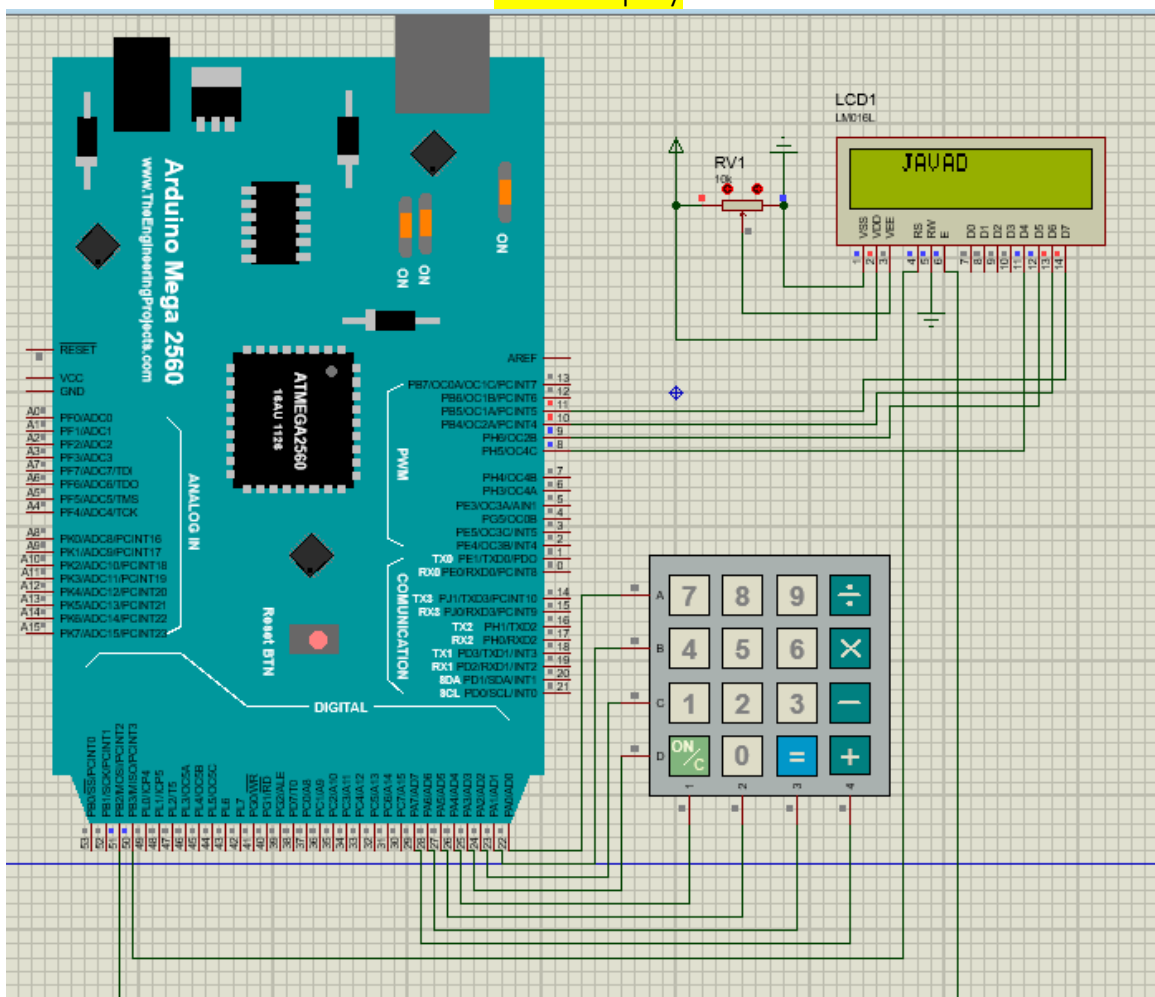
• در رابطه با LCD های گرافیکی توضیح دهید.

دارای وضوح 128x64 پیکسل می باشد. برای نمایش تصاویر تک رنگ استفاده می شود و نسبت به lcd های کاراکتری می توان متون و شکل هارا به صورت بهتری نشان داد و همچنین میتوان نمودار را هم روی آن نمایش داد درحالی که در lcd های کاراکتری نمیتوان این کار را کرد.

• هر یک از پایه های LCD برای چه هدفی استفاده می شوند؟

1. Vss = زمین یا GND
2. Vcc = تغذیه (5 ولت)
3. VO(Display Contrast Pin) = تنظیم شدت نور صفحه
4. RS(Register Select) = انتخاب ثبت data or command
5. RW(Read/Write) = پایه read , write
6. Enable = پایه E
7. D0 – D7 = پایه های دیتا (8 بیتی)
8. Anode = A
9. Cathode = K

ScrollDisplay



code

```
#include <LiquidCrystal.h>

#define RS_PIN 50
#define EN_PIN 51
#define D4_PIN 8
#define D5_PIN 9
#define D6_PIN 10
#define D7_PIN 11

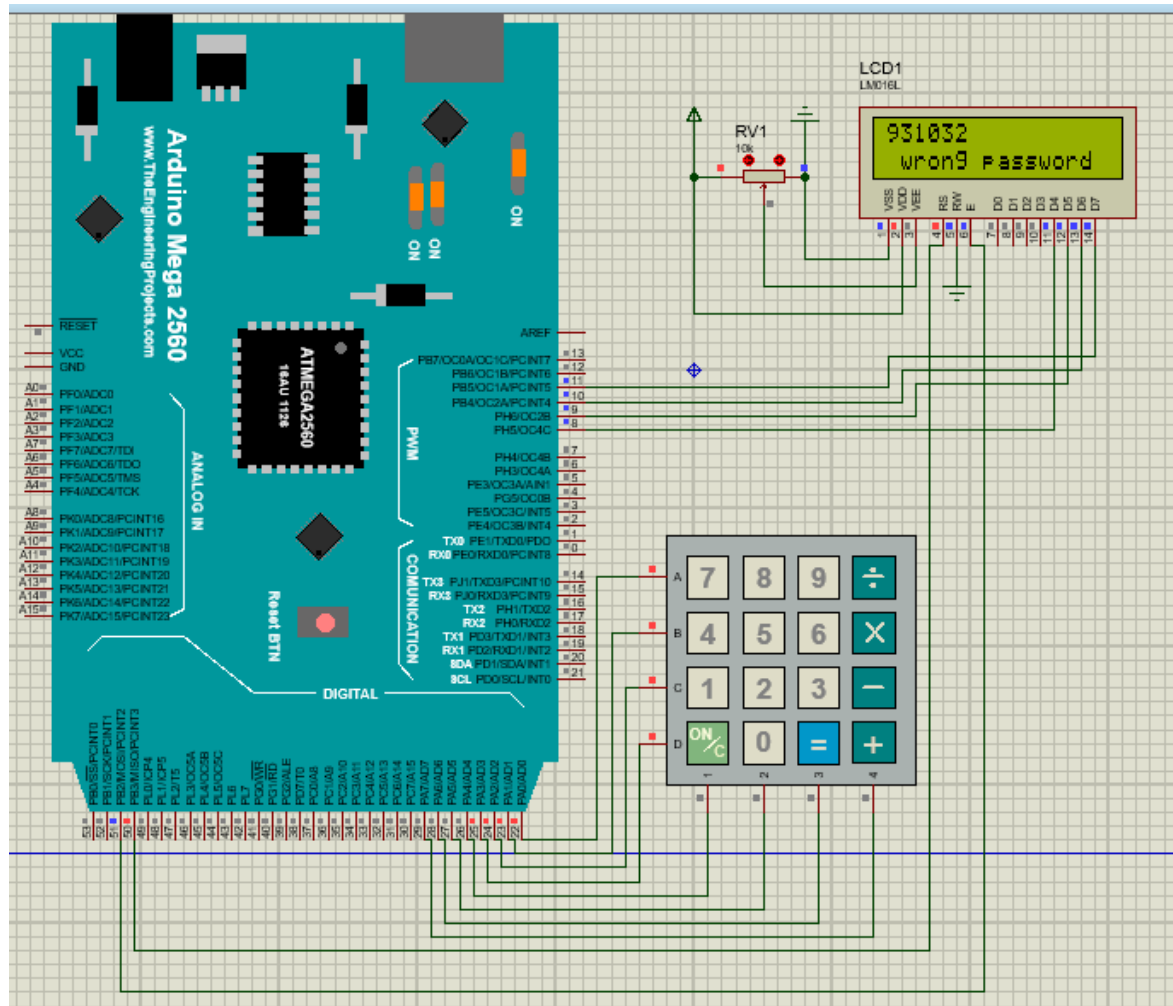
LiquidCrystal lcd(RS_PIN, EN_PIN, D4_PIN, D5_PIN, D6_PIN, D7_PIN);
char *my_name = "JAVAD";
int my_name_len = strlen(my_name);
int row = 0;

void setup() {
    // put your setup code here, to run once:
    lcd.begin(16, 2);
}

void loop() {
    // put your main code here, to run repeatedly:
    lcd.clear();
    lcd.setCursor(0, row);
    lcd.print(my_name);
    delay(1000);
    for(int i = 0; i < (16 - my_name_len); i++){
        lcd.scrollDisplayRight();
        delay(1000);
    }
    row = (row + 1) % 2;
}
```

Done compiling.

PassChecking



code

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

//defining pins
#define RS_PIN 50
#define EN_PIN 51
#define D4_PIN 8
#define D5_PIN 9
#define D6_PIN 10
#define D7_PIN 11

//global variables
String my_pass = "9831032";
String cur_input = "";
```

```

//lcd object initializing
LiquidCrystal lcd(RS_PIN, EN_PIN, D4_PIN, D5_PIN, D6_PIN, D7_PIN);

//keypad object initializing
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
  {'7','8','9','/'},
  {'4','5','6','*'},
  {'1','2','3','-'},
  {'C','0','=','+'}
};
byte rowPins[ROWS] = {22, 23, 24, 25};
byte colPins[COLS] = {26, 27, 28, 29};
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

void setup() {
  // put your setup code here, to run once:
  lcd.begin(16, 2);
  lcd.clear();

  keypad.addEventListener(keypadEvent);
}

void loop() {
  // put your main code here, to run repeatedly:
  char key = keypad.getKey();
}

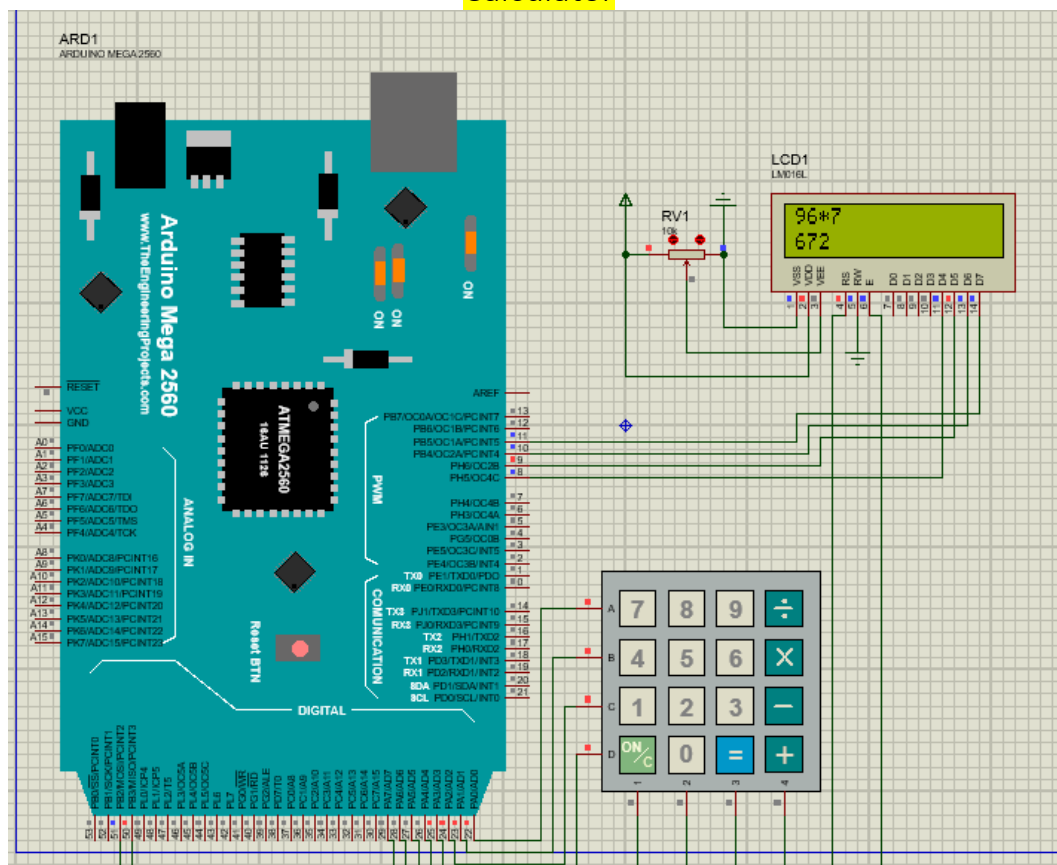
```

```

void keypadEvent(KeypadEvent key) {
  if(keypad.getState() == PRESSED){
    if((key >= '0') & (key <= '9')){
      lcd.write(key);
      cur_input += key;
    }
    else if(key == '*') {
      lcd.setCursor(0, 1);
      if(cur_input == "9831032")
        lcd.print("correct password");
      else
        lcd.print(" wrong password ");
      delay(1000);
      cur_input = "";
      lcd.clear();
    }
  }
}

```

Calculator




```

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

//defining pins
#define RS_PIN 50
#define EN_PIN 51
#define D4_PIN 8
#define D5_PIN 9
#define D6_PIN 10
#define D7_PIN 11

String op1 = "";
String op2 = "";
String operation = "";

//lcd object initializing
LiquidCrystal lcd(RS_PIN, EN_PIN, D4_PIN, D5_PIN, D6_PIN, D7_PIN);

//keypad object initializing
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
  {'7','8','9','/'},
  {'4','5','6','*'},
  {'1','2','3','-'},
  {'C','0','=','+'}
};
byte rowPins[ROWS] = {22, 23, 24, 25};
byte colPins[COLS] = {26, 27, 28, 29};
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

void setup() {
  // put your setup code here, to run once:
  lcd.begin(16, 2);
  lcd.clear();
  keypad.addEventListener(keypadEvent);
}

void loop() {
  // put your main code here, to run repeatedly:
  char key = keypad.getKey();
}

```

```

void keypadEvent(KeypadEvent key) {
    if(keypad.getState() == PRESSED){
        if((op1 == "") & (op2 == "") & (operation == ""))
            lcd.clear();

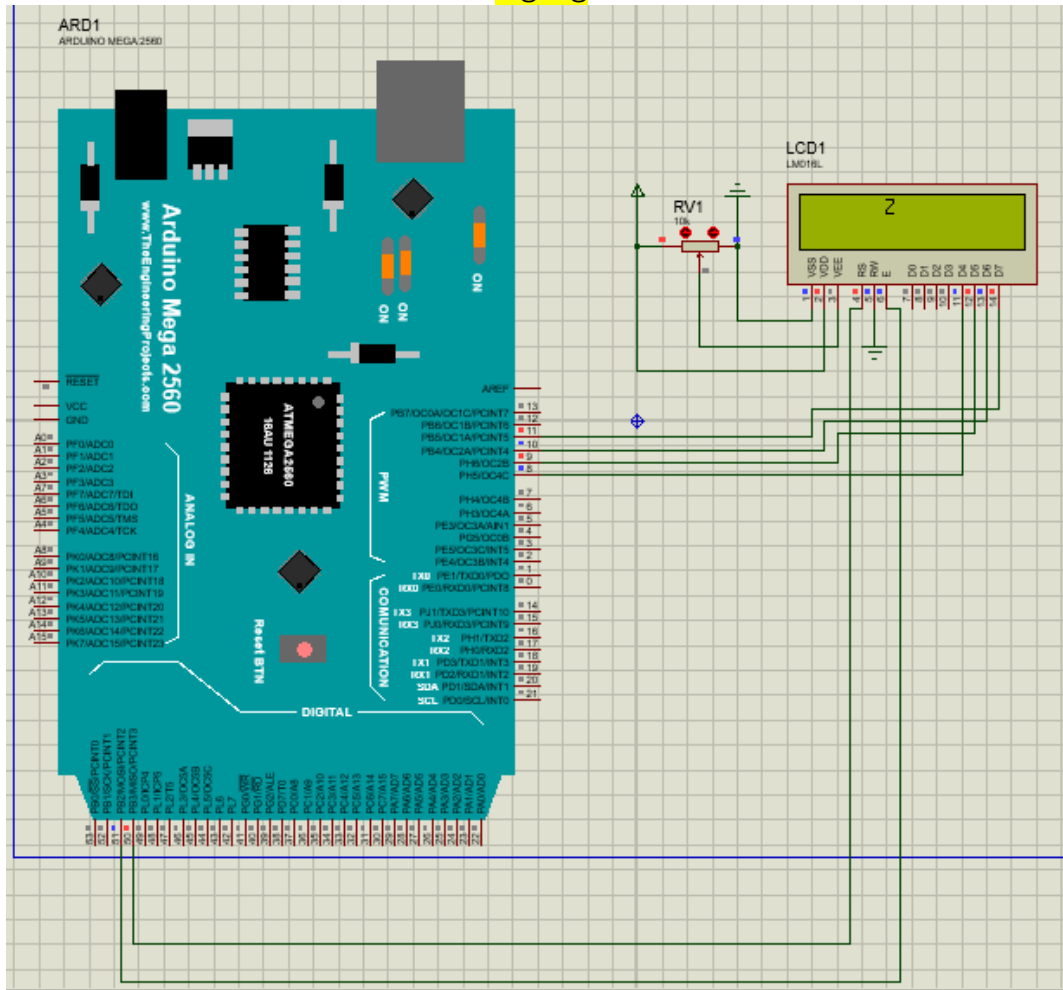
        if(key == 'C'){
            op1 = "";
            op2 = "";
            operation = "";
            lcd.clear();
        }else if((key >= '0') & (key <= '9')){
            // 0 < key < 9
            lcd.write(key);
            if(operation == "")
                op1 += key;
            else
                op2 += key;
        }else if((key != '=') & (operation == "") & (op1 != "")){
            // it means that key is one of +/* operations in its correct place
            lcd.write(key);
            operation += key;
        }else if((key == '=') & (operation != "") & (op2 != "")){
            lcd.setCursor(0, 1);

            if(operation == "+")
                lcd.print(op1.toInt() + op2.toInt());
            else if(operation == "-")
                lcd.print(op1.toInt() - op2.toInt());
            else if(operation == "*")
                lcd.print(op1.toInt() * op2.toInt());
            else if(operation == "/")
                lcd.print(op1.toInt() / op2.toInt());

            op1 = "";
            op2 = "";
            operation = "";
        }
    }
}

```

Done compiling.



code

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

//defining pins
#define RS_PIN 50
#define EN_PIN 51
#define D4_PIN 8
#define D5_PIN 9
#define D6_PIN 10
#define D7_PIN 11

char my_char = 'Z';
int row = 0;
int col = 0;

LiquidCrystal lcd(RS_PIN, EN_PIN, D4_PIN, D5_PIN, D6_PIN, D7_PIN);
```

```
void setup() {  
  // put your setup code here, to run once:  
  lcd.begin(16, 2);  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  lcd.clear();  
  lcd.setCursor(col, row);  
  lcd.write(my_char);  
  delay(1000);  
  
  row = (row+1)%2;  
  col = (col+1)%16;  
}
```