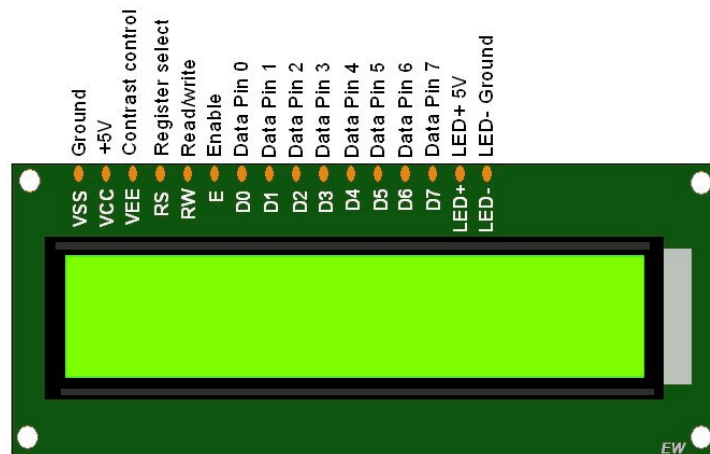


## GIAO TIẾP VỚI MÀN HÌNH LCD 16x2

### 1. Giới thiệu màn hình LCD 16x2

Thông thường màn hình LCD 16x2 được điều khiển bởi chip HD44780 của hãng Hitachi. Nó có thể hiển thị 1 hoặc 2 hàng với 16 ký tự mỗi hàng. Ngoài các ký tự trong bảng chữ cái, chữ số từ 0 đến 9 nó còn có thể hiển thị một số ký tự đặc biệt. Ngoài ra người dùng có thể tự tạo cho mình một số ký tự hoặc biểu tượng trên LCD16x2.

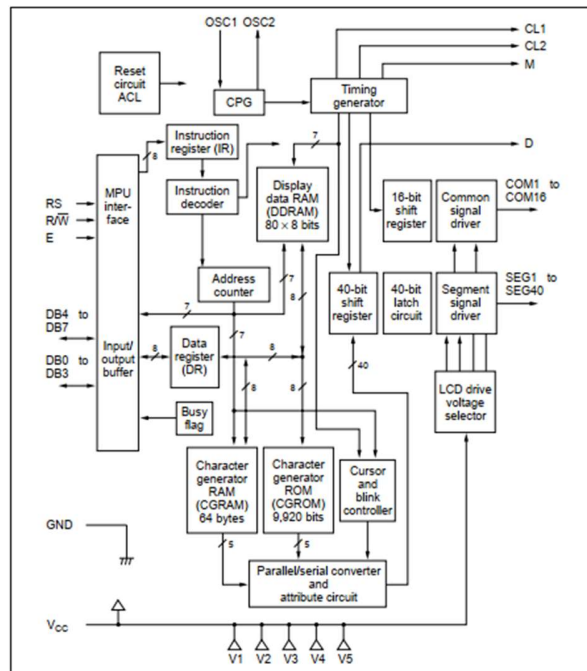


Hình 1. Màn hình LCD16x2 và các chân tín hiệu.

### 2. Các chân tín hiệu của LCD16x2

CHÂN	KÝ HIỆU	MỨC	MÔ TẢ
1	VSS	0VDC	Chân GND
2	VCC	5.0VDC	Điện áp cấp cho mạch điều khiển
3	VEE	0-VCC	Điều chỉnh độ tương phản cho LCD
4	RS	H/L	Chân lựa chọn thanh ghi dữ liệu (DATA Register) hoặc thanh ghi lệnh (Instruction Register) H: Data register, L: Instruction Register
5	R/W	H/L	Chân đọc ghi dữ liệu. H: Đọc dữ liệu từ LCD về VĐK. L: Ghi dữ liệu từ VĐK ra LCD
6	E	H,H->L	Khi chân này chuyển từ mức logic H sang L, dữ liệu từ các chân DB0-DB7 được ghi vào LCD
7-14	DB0-DB7	H/L	Dữ liệu truyền giữa MCU và LCD.
15	ANODE		Điện áp cấp cho Led nền.
16	CATHODE		GND của Led nền

### 3. Chip điều khiển LCD16x2 HD44780



**Hình 2. Sơ đồ khối chip HD44780 của hãng Hitachi**

#### 3.1. Thanh ghi

HD44780 có hai thanh ghi 8 bit: thanh ghi lệnh IR và thanh ghi dữ liệu DR.

##### 3.1.1. Thanh ghi lệnh IR (Instruction Register)

Thanh ghi lệnh IR ghi các lệnh điều khiển như xóa màn hình, dịch chuyển con trỏ và thông tin địa chỉ cho Display Data RAM (DDRAM) và Character Generator RAM (CGRAM). Thanh ghi lệnh IR chỉ được ghi từ VĐK.

##### 3.1.2. Thanh ghi dữ liệu DR (Data Register)

Thanh ghi DR lưu trữ giá trị đọc hoặc ghi từ DDRAM hoặc CGRAM. Khi lệnh thông tin địa chỉ được ghi vào thanh ghi IR thì thông tin từ DDRAM hoặc CGRAM sẽ được ghi vào thanh ghi DR.

Thanh ghi DR hoặc IR được lựa chọn thông qua chân tín hiệu RS

RS	R/W	Nội dung
0	0	Dữ liệu từ DB0-DB7 được ghi vào thanh ghi IR
0	1	Đọc trạng thái cờ báo bận (chân DB7) và bộ đếm địa chỉ (DB0-DB6)
1	0	Ghi dữ liệu từ DB0-DB7 lên DDRAM hoặc CGRAM
1	1	Đọc dữ liệu từ DDRAM hoặc CGRAM thông qua các chân DB0-

		DB6
--	--	-----

### 3.2. Cờ báo bận *BF (Busy Flag)*

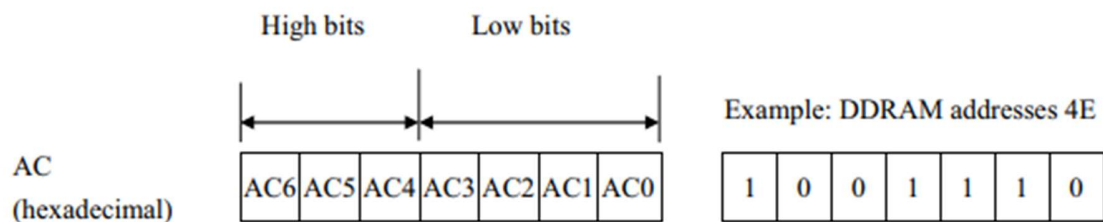
Khi cờ báo bận bằng 1, bộ điều khiển bên trong LCD đang ở chế độ thực hiện các lệnh bên trong, trường hợp này sẽ không nhận các lệnh kế tiếp. Khi  $RS = 0$  và  $R/W = 1$  thì tín hiệu tại chân DB7 sẽ ở mức cao. Lệnh kế tiếp được ghi sau khi chân DB7 về mức 0.

### 3.3. Bộ đếm địa chỉ *AC (Address Counter)*

Bộ đếm địa chỉ gán địa chỉ cho cả thanh ghi DDRAM và CGRAM.

### 3.4. Thanh ghi dữ liệu hiển thị *DDRAM (Display Data RAM)*

Thanh ghi DDRAM dùng lưu trữ dữ liệu hiển thị dưới dạng mã ký tự 8 bit. Nó có khả năng mở rộng tới 80 ký tự 80x8bit. Hình dưới minh họa mối tương quan giữa địa chỉ DDRAM và vị trí hiển thị trên LCD



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

2-Line by 16-Character Display

### 3.5. Bộ nhớ sinh ký tự *CGROM (Character Generator ROM)*

Bộ nhớ CGROM sinh ký tự 5x8 hoặc 5x10 điểm ảnh từ mã ký tự 8 bit. Xem bảng dưới





For 5 \* 8 dot character patterns

Character Codes ( DDRAM data )	CGRAM Address	Character Patterns ( CGRAM data )
7 6 5 4 3 2 1 0	5 4 3 2 1 0	7 6 5 4 3 2 1 0
High Low	High Low	High Low
<div> <div>0 0 0 0 0 0 0 0</div> <div>0 0 0 0 0 0 0 0</div> </div>	<div> <div>0 0 0</div> <div>0 0 0</div> </div>	<div> <div>Character pattern( 1 )</div> <div>Cursor pattern</div> <div>Character pattern( 2 )</div> <div>Cursor pattern</div> </div>
<div> <div>0 0 0 0 0 0 0 1</div> </div>	<div> <div>0 0 1</div> </div>	
<div> <div>0 0 0 0 0 1 1 1</div> </div>	<div> <div>1 1 1</div> </div>	

#### Hình 4. Mẫu ký tự với độ phân giải 5x8 điểm ảnh.

For 5 \* 10 dot character patterns

Character Codes (DDRAM data)		CGRAM Address		Character Patterns (CGRAM data)	
7 6 5 4 3 2 1 0		5 4 3 2 1 0		7 6 5 4 3 2 1 0	
High	Low	High	Low	High	Low
0 0 0 0 0 0 0 0	0 0	0 0 0 0 0 0	0 0		
		0 0 0 0 1			
		0 0 0 1 0			
		0 0 0 1 1			
		0 0 1 0 0			
		0 0 1 0 1			
		0 0 1 1 0			
		0 0 1 1 1			
		0 1 0 0 0			
		1 0 0 0 1			
		1 0 1 0 0			
		1 1 1 1 1			
		1 1 1 1 1			

### Hình 5. Mẫu ký tự với độ phân giải 5x10 điểm ảnh.

#### 4. Tập lệnh điều khiển HD44780

LỆNH	MÃ LỆNH										Mô tả	Thời gian
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Xóa hiển thị trên LCD và đặt địa chỉ của DDRAM trong bộ đếm địa chỉ bằng 0	1,53ms
Return Home	0	0	0	0	0	0	0	0	1	-	Đặt địa chỉ 0x00 cho DDRAM từ thanh ghi AC và đưa con trỏ về vị trí gốc. Nội dung của thanh ghi DDRAM vẫn giữ nguyên.	1,53s
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	I/D thiết lập hướng di chuyển của con trỏ và S thiết lập sự dịch chuyển hiển thị.	39us
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Các bit điều khiển Hiển thị (D), con trỏ (C) và nhấp nháy (B)	39us
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	-	-	Di chuyển con trỏ và dịch chuyển hiển thị nhưng không làm thay đổi nội dung thanh ghi DDRAM	39us
Function Set	0	0	0	0	1	DL	N	F	-	-	Thiết lập độ dài dữ liệu giao tiếp (DL: 8-bit/4-bit), số lượng dòng hiển thị (N: 2 dòng/1dòng) và kiểu font(F:5x10/5x8)	39us
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Đặt địa chỉ CGRAM từ bộ đếm địa chỉ	39us
Set DDRAM Address	0	0	01	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Đặt địa chỉ DDRAM từ bộ đếm địa chỉ	39us
Read Busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Bất cứ khi nào có tác vụ nội trong LCD BF sẽ bằng 1. Nội dung của thanh ghi AC cũng có thể đọc cùng.	0us
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Ghi dữ liệu vào RAM(DDRAM/CGRAM)	43us
Read Data From RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Đọc dữ liệu từ RAM(DDRAM/CGRAM)	43us

##### 4.1. Entry Mode Set

- I/D = 1: tăng
- I/D = 0: giảm
- S = 1: dịch chuyển ký tự theo I/D.

##### 4.2. Display ON/OFF Control

- D = 1: bật hiển thị.
- D = 0: tắt hiển thị.
- C = 0: tắt con trỏ.
- C = 1: bật con trỏ
- B = 0: tắt chế độ nhấp nháy.
- B = 1: bật chế độ nhấp nháy.

#### 4.3. Dịch chuyển con trỏ hoặc hiển thị

Dịch chuyển con trỏ hoặc hiển thị sang trái hoặc phải mà không cần phải đọc/ghi dữ liệu hiển thị. Chức năng này được dùng để hiệu chỉnh hay tìm kiếm. Với chế độ hiển thị hai hàng con trỏ sẽ dịch chuyển xuống hàng thứ 2 nếu nó đi qua ký tự thứ 40 của hàng đầu tiên. Lưu ý rằng dòng 1 và 2 đều cùng dịch chuyển tại cùng thời điểm.

- S/C = 0: dịch chuyển con trỏ.
- S/C = 1: dịch chuyển ký tự.
- R/L = 0: dịch chuyển sang trái.
- R/L = 1: dịch chuyển sang phải.

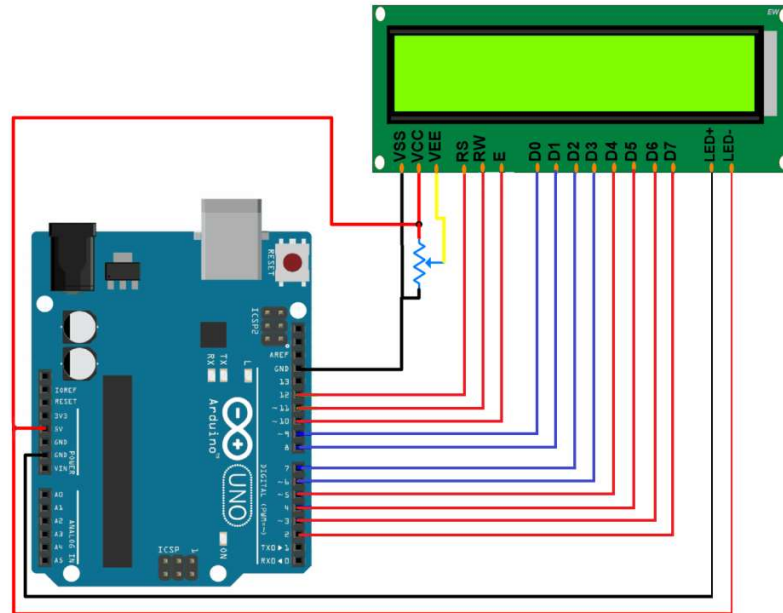
#### 4.4. FunctionSet

- DL = 0: giao tiếp 4bit dữ liệu (1byte được truyền 2 lần, lần 1 4bit cao, lần 2 4bit thấp)
- DL = 1: giao tiếp 8bit dữ liệu.
- N = 0: hiển thị một hàng ký tự (hàng trên).
- N = 1: hiển thị 2 hàng ký tự.
- F = 0: sử dụng font ký tự 5x8 dot.
- F = 1: sử dụng font ký tự 5x10 dot.

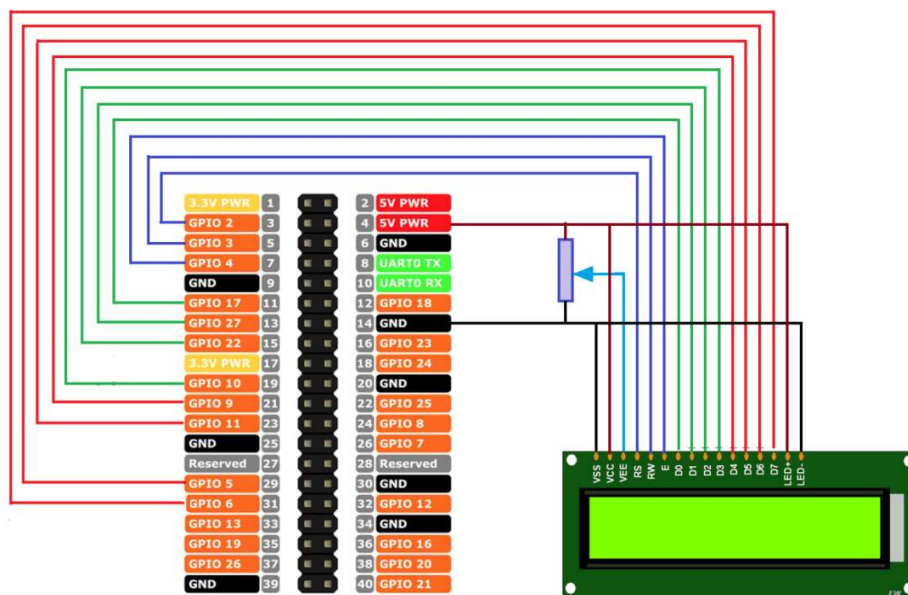
#### 4.5. Một số lệnh cơ bản được xây dựng từ

TT	Mã Hex	Lệnh
1	0x01	Xóa màn hình
2	0x02	Đưa con trỏ về góc
3	0x04	Dịch chuyển con trỏ sang trái
4	0x05	Dịch chuyển các ký tự hiển thị sang phải
5	0x06	Dịch chuyển con trỏ sang phải
6	0x07	Dịch chuyển các ký tự hiển thị sang trái
7	0x08	Tắt hiển thị ký tự và con trỏ
8	0x0A	Tắt hiển thị ký tự nhưng con trỏ vẫn bật
9	0x0C	Hiển thị ký tự nhưng tắt con trỏ
10	0x0E	Hiển thị ký tự, con trỏ nhấp nháy
11	0x0F	Hiển thị ký tự, con trỏ nhấp nháy
12	0x10	Dịch chuyển vị trí con trỏ sang trái
13	0x14	Dịch chuyển vị trí con trỏ sang phải
14	0x18	Dịch chuyển toàn bộ ký tự hiển thị sang trái
15	0x1C	Dịch chuyển toàn bộ ký tự hiển thị sang phải
16	0x80	Đưa con trỏ về ô đầu tiên hàng 1.
17	0xC0	Đưa con trỏ về ô đầu tiên hàng 2.
18	0x38	Hiển thị 2 dòng ký tự 5x8.

## 5. Kết nối LCD16x2 với Arduino và Raspberry Pi



Hình 6. Sơ đồ kết nối LCD16x2 với Arduino UNO.



Hình 7. Sơ đồ kết nối LCD16x2 với Raspberry Pi



CHÂN KẾT NỐI GIỮA LCD16x2 VỚI ARDUINO UNO					
LCD	ARDUINO	LCD	ARDUINO	LCD	ARDUINO
RS	12	D1	8	D5	4
R/W	11	D2	7	D6	3
EN	10	D3	6	D7	2
D0	9	D4	5		

CHÂN KẾT NỐI GIỮA LCD16x2 VỚI RASPBERRY PI					
LCD	RASPBERRY	LCD	RASPBERRY	LCD	RASPBERRY
RS	GPIO2	D1	GPIO21	D5	GPIO11
R/W	GPIO3	D2	GPIO22	D6	GPIO5
EN	GPIO4	D3	GPIO10	D7	GPIO6
D0	GPIO17	D4	GPIO9		

Dữ liệu truyền giữa VĐK và LCD16x2 được thực hiện theo 2 cách:

- Truyền 8 bit: ngoài 3 chân kết nối với RS, R/W,E VĐK cần 8 chân để kết nối với các chân dữ liệu D0-D7 của LCD16x2.
- Truyền 4 bit: ngoài 3 chân kết nối với RS, R/W,E VĐK chỉ cần 4 chân để kết nối với các chân dữ liệu D4-D7 của LCD16x2. Dữ liệu được truyền lên LCD làm 2 lần. Lần 1 là 4bit cao, lần 2 là 4bit thấp.

Định nghĩa và thiết lập các chân VĐK giao tiếp với LCD16x2 như sau:

#### ARDUINO UNO

```
#define RSpin 12
#define RWpin 11
#define ENpin 10
#define D0 9
#define D1 8
#define D2 7
#define D3 6
#define D4 5
#define D5 4
#define D6 3
#define D7 2
uint8_t DATApin[8] = {D0, D1, D2, D3, D4, D5, D6, D7};
void setup(){
  pinMode(ENpin,OUTPUT);
  pinMode(RWpin,OUTPUT);
  pinMode(RSpin,OUTPUT);
  for(int i = 0; i < 8; i++){
    pinMode(DATApin[i], OUTPUT);
  }
}
```

#### RASPBERRY PI

```
import RPi.GPIO as GPIO
import time
RSpin = 2
RWpin = 3
ENpin = 4
D0 = 17
D1 = 27
D2 = 22
D3 = 10
D4 = 9
D5 = 11
D6 = 5
D7 = 6
DATApin = [D0, D1, D2, D3, D4, D5, D6, D7]
GPIO.setmode(GPIO.BCM)
GPIO.setup(ENpin,GPIO.OUT)
GPIO.setup(RWpin,GPIO.OUT)
GPIO.setup(RSpin,GPIO.OUT)
GPIO.setup(DATApin,GPIO.OUT)
```

## 6. Chương trình giao tiếp giữa VĐK và LCD16x2.

Xây dựng một số hàm căn bản nhằm giao tiếp giữa VĐK và LCD16x2.

Hàm writeLCD có nhiệm vụ chuyển đổi số nguyên 8 bit ra mã nhị phân với các chân tương ứng từ D0 - D7 của LCD16x2. Dữ liệu từ các chân này được LCD16x2 nhận khi chân tín hiệu EN chuyển từ mức cao xuống mức thấp.

Hàm command2LCD ghi dữ liệu từ VĐK sang thanh ghi lệnh của LCD16x2 bằng cách kéo chân RS xuống mức 0.

Hàm data2LCD ghi dữ liệu từ VĐK sang thanh ghi dữ liệu của LCD16x2 bằng cách kéo chân RS lên mức 1.

Hàm Str2LCD hiển thị chuỗi ký tự lên LCD16x2

### ARDUINO UNO

```
void writeLCD(uint8_t byte)
{
    for(int i = 0; i < 8; i++){
        digitalWrite(DATApin[i], (byte >> i) & 0x01);
    }
    digitalWrite(ENpin,1);
    delayMicroseconds(500);
    digitalWrite(ENpin,0);
}
void command2LCD(uint8_t cmd)
{
    digitalWrite(RSpin,0);
    digitalWrite(RWpin,0);
    writeLCD(cmd);
}
void data2LCD(uint8_t data)
{
    digitalWrite(RSpin,1);
    digitalWrite(RWpin,0);
    writeLCD(data);
}
void Str2LCD(String str)
{
    int i = 0;
    while(str[i] != '\0')
    {
        data2LCD(str[i]);
        i++;
    }
}
```

### RASPBERRY PI

```
def writeLCD(byte):
    for i in range(8):
        GPIO.output(DATApin[i],(byte >> i)& 0x01)
        GPIO.output(ENpin,1)
        time.sleep(0.0005)
        GPIO.output(ENpin,0)
        time.sleep(0.0005)

def command2LCD(cmd):
    GPIO.output(RSpin,0)
    GPIO.output(RWpin,0)
    writeLCD(cmd)

def data2LCD(data):
    GPIO.output(RSpin,1)
    GPIO.output(RWpin,0)
    writeLCD(data)

def Str2LCD(str):
    n = len(str)
    for i in range(n):
        data2LCD(ord(str[i]))
```

Ví dụ: hiển thị dòng chữ DHCN – DHQGHN lên LCD16x2

#### ARDUINO UNO

#### RASPBERRY PI

```
void setup()
{
//Thêm các dòng lệnh sau vào code Setup
  command2LCD(0x01);
  command2LCD(0x38);
  command2LCD(0x0C);
  command2LCD(0x80);
  Str2LCD("DHCN - DHQGHN");
}
```

Các hàm sử dụng trong code:

- 0x01: xóa màn hình hiển thị.
- 0x38: hiển thị 2 dòng, 5x8 điểm ảnh, chế độ giao tiếp 8 bit.
- 0x0C: hiển thị ký tự và ẩn con trỏ.
- 0x80: di chuyển con trỏ về vị trí đầu của hàng thứ nhất.

#### 7. Bài tập:

- Viết hàm khởi tạo LCD với các chức năng:
  - Hiển thị 1 hoặc 2 dòng ký tự.
  - Chọn font chữ 5x8 điểm ảnh hoặc 5x10 điểm ảnh.
- Viết hàm bật hoặc tắt con trỏ.
- Viết hàm hiển thị hàng chữ lên LCD với chức năng canh phải, canh trái, canh giữa.
- Với các chuỗi lớn hơn 16 ký tự, viết hàm chạy chữ lần lượt từ phải qua trái sao cho hiển thị liên tục hàng chữ trên.