

Hướng dẫn sử dụng thư viện HT

Contents

I.	Thông tin & Điều khoản.	2
II.	CÀI ĐẶT THƯ VIỆN.	2
2.1.	Phiên bản thư viện	2
2.2.	Cách cài đặt.	2
2.3.	Cấu tạo thư viện.....	2
III.	KẾT NỐI VẬT LÝ.	3
IV.	CÁC PHƯƠNG THỨC ĐIỀU KHIỂN MÀN HÌNH.....	3
4.1.	Khởi tạo màn hình LCD.....	3
4.2.	Hiển thị số nguyên (âm hoặc dương).	3
4.3.	Hiển thị số thực có dấu chấm (".")......	4
4.4.	Hiển thị định dạng giờ.....	5
4.5.	Hiển thị định dạng nhiệt độ C.....	6

I. Thông tin & Điều khoản.

Cảm ơn các bạn đã tin và sử dụng thư viện này. Thư viện này được thiết kế để hoạt động với các màn hình 7 đoạn LCD 4 số do bên banlinhkien.vn bán. Các bạn có thể tham khảo hàng mua [tại đây](#).

Thư viện này được Tác giả người viết ra hướng dẫn này viết dựa theo thư viện của Anxzu. Tham khảo thư viện gốc [tại đây](#).

Thư viện này đã được xin phép thay đổi và chỉnh sửa lại để phù hợp với màn hình LCD của banlinhkien.vn.

Thư viện này hoàn toàn miễn phí sử dụng và thay đổi để phù hợp hơn trong quá trình sử dụng.

Chúng tôi không chịu trách nhiệm trước sự hỏng hóc của các thiết bị và các thiết bị liên quan khi sử dụng thư viện này mặc dù chúng tôi đã kiểm tra không có lỗi hoặc hỏng hóc với thiết bị trước khi phát hành thư viện.

Trong quá trình cài đặt hoặc sử dụng, các bạn gặp vấn đề gì vui lòng liên hệ:

Email: nhatthanh228@Gmail.com

Số dt: 0947321478

II. CÀI ĐẶT THƯ VIỆN.

2.1. Phiên bản thư viện

Phiên bản thư viện hiện thời bây giờ là: 1.3

**** Lưu ý:** tài liệu hướng dẫn này có thể thay đổi theo từng phiên bản thư viện mà không cần báo trước.

2.2. Cách cài đặt.

Tải File Zip từ đường dẫn phía dưới về. Giải nén ra thư mục tên “segmentLcd”

Có thể dùng một trong hai cách sau để cài đặt thư viện và sử dụng với Arduino.:

- Copy thư viện vào thư mục sau:
“C:\Program Files (x86)\Arduino\libraries”
- Hoặc Copy thư viện vào Documents\Arduino\libraries.

Trong hướng dẫn này, thư viện sẽ được copy vào Documents\Arduino\libraries

****Lưu ý:** Để nâng cao tính ổn định, thư viện được khuyến cáo copy vào thư mục chỉ định trước khi khởi động Arduino IDE.

2.3. Cấu tạo thư viện.

Thư viện này gồm có 3 files chính:

- segmentLcd.cpp – chứa mã nguồn bao gồm các phương thức điều khiển.
- segmentLcd.h – header khai báo các phương thức và các biến cần sử dụng.
- keywords.txt – chứa các từ khóa để cho Arduino IDE đổi màu chữ khi gọi các phương thức

III. KẾT NỐI VẬT LÝ.

Đối với màn hình này, yêu cầu 4 điểm kết nối:

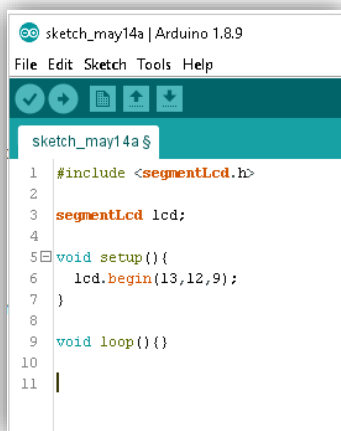
- Vcc – nối 3.3v+
- GND – nối Ground
- 13 – cs Pin
- 12 – wr Pin
- 9 – data Pin

****Lưu ý:** Tùy theo việc thiết kế mạch mà các điểm kết nối có thể thay đổi một cách linh hoạt. Khi thay đổi, phải khai báo trong chương trình.

IV. CÁC PHƯƠNG THỨC ĐIỀU KHIỂN MÀN HÌNH.

4.1. Khởi tạo màn hình LCD.

Để khởi tạo màn hình, vui lòng tham khảo đoạn mã sau:



```

1 #include <segmentLcd.h>
2
3 segmentLcd lcd;
4
5 void setup(){
6   lcd.begin(13,12,9);
7 }
8
9 void loop(){}
10
11

```

```

#include <segmentLcd.h> // thêm thư viện

segmentLcd lcd; //khởi tạo đối tượng "lcd"

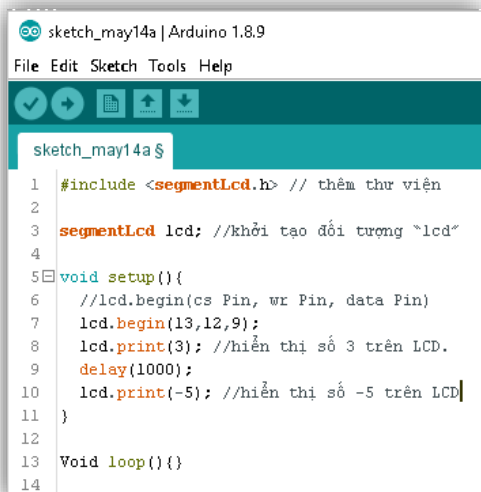
void setup(){
  //lcd.begin(cs Pin, wr Pin, data Pin)
  lcd.begin(13,12,9);
}

Void loop(){}

```

4.2. Hiển thị số nguyên (âm hoặc dương).

Để màn hình hiển thị số tự nhiên (âm hoặc dương) sử dụng hàm `print(Int Number)`. Chi tiết xem ví dụ dưới đây.



```

1 #include <segmentLcd.h> // thêm thư viện
2
3 segmentLcd lcd; //khởi tạo đối tượng "lcd"
4
5 void setup(){
6   //lcd.begin(cs Pin, wr Pin, data Pin)
7   lcd.begin(13,12,9);
8   lcd.print(3); //hiển thị số 3 trên LCD.
9   delay(1000);
10  lcd.print(-5); //hiển thị số -5 trên LCD
11 }
12
13 Void loop(){}
14

```

```

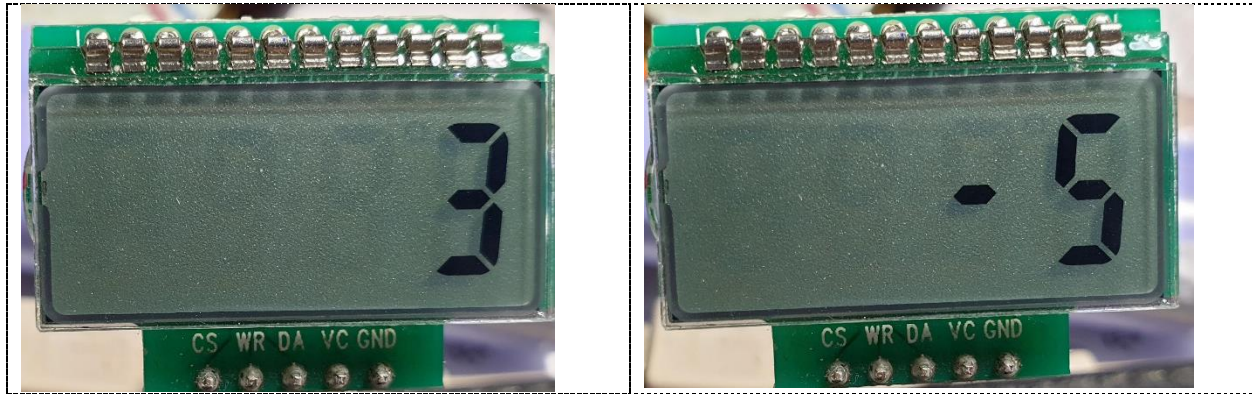
#include <segmentLcd.h> // thêm thư viện

segmentLcd lcd; //khởi tạo đối tượng "lcd"

void setup(){
  //lcd.begin(cs Pin, wr Pin, data Pin)
  lcd.begin(13,12,9);
  lcd.print(3); //hiển thị số 3
  delay(1000);
  lcd.print(-5); //hiển thị số -5
}

Void loop(){}

```



****Lưu ý:**

- Số sẽ luôn được hiển thị từ phía bên tay trái.
- Vì màn hình chỉ có 4 số nên số dương lớn nhất có thể hiển thị là "9999", và số âm nhỏ nhất có thể hiển thị là "-999".
Nếu vượt ngưỡng hai giá trị này, màn hình sẽ giữ nguyên "9999" và "-999".

4.3. Hiển thị số thực có dấu chấm (".").

Để màn hình hiển thị số thực có dấu phẩy (âm hoặc dương) sử dụng hàm `printFloat`(Floating number), chi tiết xem ví dụ phía dưới đây

```

sketch_may17a | Arduino 1.8.9
File Edit Sketch Tools Help

sketch_may17a §
1 #include <segmentLcd.h> //Thêm thư viện
2
3 segmentLcd lcd; //Khởi tạo đối tượng "lcd"
4
5 void setup(){
6   //lcd.begin(cs Pin, wr Pin, data Pin);
7   lcd.begin(13,12,9);
8   lcd.printFloat(3.14); //Hiển thị số 3.14
9   delay(1000);
10  lcd.printFloat(23.12); //Hiển thị số 23.12
11 }
12
13 void loop(){}

```

```

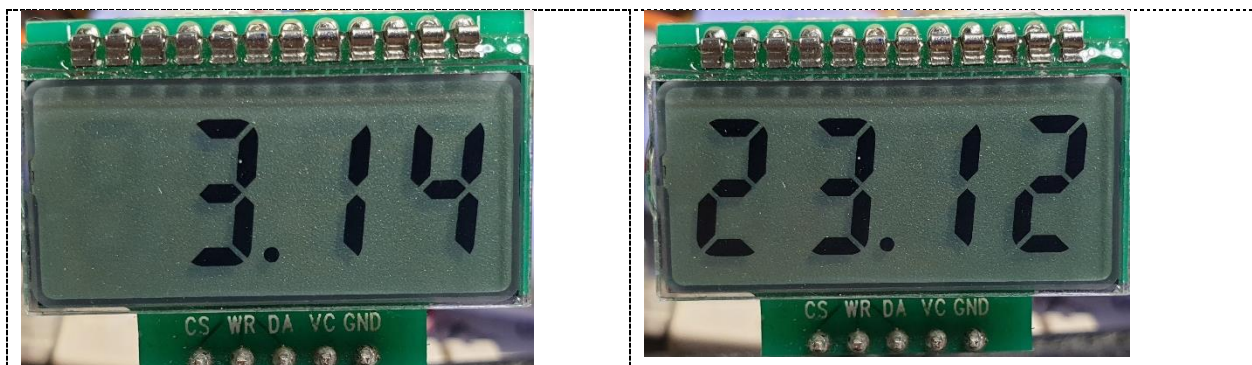
#include <segmentLcd.h> // thêm thư viện

segmentLcd lcd; //khởi tạo đối tượng "lcd"

void setup(){
  //lcd.begin(cs Pin, wr Pin, data Pin)
  lcd.begin(13,12,9);
  lcd.printFloat(3.14); //hiển thị số 3.14
  delay(1000);
  lcd.printFloat(23.12); //hiển thị số 23.12
}

Void loop(){}

```



****Lưu ý:**

- Số sẽ luôn được hiển thị từ phía bên tay trái.
- Tính đến thời điểm hiện tại, thư viện hỗ trợ xuất định dạng số thực có tối đa 2 chữ số đằng sau dấu chấm (".").
Ví dụ "3.14" hay "23.12".
Trong trường hợp số thực có nhiều hơn hoặc bằng 2 chữ số đằng sau dấu chấm ("."), số thực sẽ được tự làm tròn lên thành 2 chữ số đằng sau dấu chấm (".").
Ví dụ: "23.125" sẽ được in ra thành "23.13".
- Vì màn hình chỉ có 4 số nên số dương lớn nhất có thể hiển thị là "99.99", và số âm nhỏ nhất có thể hiển thị là "-9.99".
Nếu vượt ngưỡng hai giá trị này, màn hình sẽ giữ nguyên "99.99" và "-9.99".

4.4. Hiển thị định dạng giờ.

Để màn hình hiển thị giờ hoặc ngày sử dụng hàm `printTime(1st Int Number, 2nd Int Number)`, chi tiết xem ví dụ phía dưới đây:

```

sketch_may17a | Arduino 1.8.9
File Edit Sketch Tools Help
[Icons] Upload
sketch_may17a $
1  #include <segmentLcd.h> //Thêm thư viện
2
3  segmentLcd lcd; //Khởi tạo đối tượng "lcd"
4
5  void setup(){
6      //lcd.begin(cs Pin, wr Pin, data Pin);
7      lcd.begin(13,12,9);
8      lcd.printTime(02,15);
9  }
10
11 void loop(){}

```

```

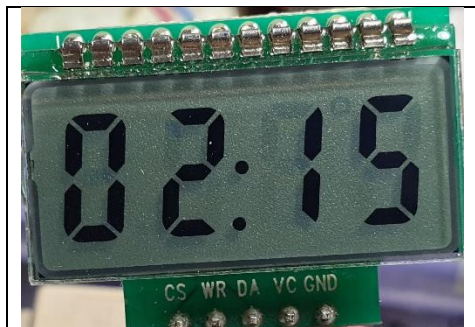
#include <segmentLcd.h> // thêm thư viện

segmentLcd lcd; //khởi tạo đối tượng "lcd"

void setup(){
    //lcd.begin(cs Pin, wr Pin, data Pin)
    lcd.begin(13,12,9);
    lcd.printTime(02,15); //In 2 giờ 15 phút
}

Void loop(){}

```

****Lưu ý:**

- Định dạng giờ số thứ nhất và số thứ 2 luôn có dạng là số nguyên dương.
- Vì là định dạng giờ nên số thứ nhất và số thứ 2 luôn luôn có giá trị cực đại là "99"
Nếu vượt ngưỡng hai giá trị này, màn hình sẽ giữ nguyên "99".

4.5. Hiển thị định dạng nhiệt độ C

Để màn hình hiển thị giờ hoặc ngày sử dụng hàm `printC`(Floating number), chi tiết xem ví dụ phía dưới đây:

```
sketch_may14a | Arduino 1.8.9
File Edit Sketch Tools Help

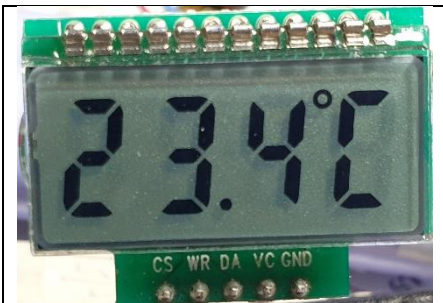
sketch_may14a $
1 #include <segmentLcd.h> // thêm thư viện
2
3 segmentLcd lcd; //khởi tạo đối tượng "lcd"
4
5 void setup(){
6   //lcd.begin(cs Pin, wr Pin, data Pin)
7   lcd.begin(13,12,9);
8   lcd.printC(23.4);
9 }
10
11 void loop() {}
12
```

```
#include <segmentLcd.h> // thêm thư viện
```

```
segmentLcd lcd; //khởi tạo đối tượng "lcd"
```

```
void setup(){
    //lcd.begin(cs Pin, wr Pin, data Pin)
    lcd.begin(13,12,9);
    lcd.printC(23.4); //In 23.4 *C
}
```

```
void loop() {}
```



****Lưu ý:**

- Vì màn hình chỉ có 4 số, ký tự cuối cùng chiếm mất một số nên số dương lớn nhất có thể hiển thị là "99.9", và số âm nhỏ nhất có thể hiển thị là "-9.9".
Nếu vượt ngưỡng hai giá trị này, màn hình sẽ giữ nguyên "99.9" và "-9.9".