

**TRƯỜNG ĐẠI HỌC TRẦN ĐẠI NGHĨA**  
**KHOA CÔNG NGHỆ THÔNG TIN**



## **BÁO CÁO**

### **BÀI TẬP LỚN MÔN HỌC** **VI XỬ LÝ VÀ ỨNG DỤNG**

**Đề tài: “Điều Khiển Lò Áp Trứng Gà”**

**GVHD: Hoàng Xuân Tuân**

**Sinh viên thực hiện:**

**Tổng Thành Anh**

**Tôn Nữ Nguyên Hậu**

**Nguyễn Lê Xuân Phước**

**Lớp: 16DDS06031**

**TP. HỒ CHÍ MINH, THÁNG 12 NĂM 2018**

TRƯỜNG ĐẠI HỌC TRẦN ĐẠI NGHĨA  
KHOA CÔNG NGHỆ THÔNG TIN

## **BÁO CÁO**

### **BÀI TẬP LỚN MÔN HỌC VI XỬ LÝ VÀ ỨNG DỤNG**

**Đề tài: “Điều Khiển Lò Áp Trứng Gà”**

**GVHD: Hoàng Xuân Tuân**

**Sinh viên thực hiện:**

**Tổng Thành Anh**

**Tôn Nữ Nguyên Hậu**

**Nguyễn Lê Xuân Phước**

**Lớp: 16DDS06031**

**TP. HỒ CHÍ MINH, THÁNG 12 NĂM 2018**

## MỤC LỤC

LỜI NÓI ĐẦU .....	4
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI.....	5
1.1. Tổng quan về lò ấp trứng gà.....	5
1.2. Sơ lược về hệ thống và sơ đồ khối của hệ thống .....	5
1.2.1. Sơ lược hệ thống .....	5
1.2.2. Sơ đồ khối hệ thống.....	5
CHƯƠNG 2: XÂY DỰNG MÔ HÌNH HỆ THỐNG VÀ CÁC THIẾT BỊ CHỨC NĂNG .....	6
2.1. Giới thiệu chung về Arduino.....	6
2.2. Boar Arduino UNO R3 .....	7
2.3. LCD 1602 .....	11
2.4. Cảm biến LM35.....	17
CHƯƠNG 3: MÔ PHỎNG.....	19
KẾT LUẬN .....	25
TÀI LIỆU THAM KHẢO .....	26

## LỜI NÓI ĐẦU

Ngày nay khoa học công nghệ ngày càng phát triển, vi điều khiển AVR và vi điều khiển PIC ngày càng thông dụng và hoàn thiện hơn, nhưng có thể nói sự xuất hiện của Arduino vào năm 2005 tại Italia đã mở ra một hướng đi mới cho vi điều khiển. Sự xuất hiện của Arduino đã hỗ trợ cho con người rất nhiều trong lập trình và thiết kế, nhất là đối với những người bắt đầu tìm tòi về vi điều khiển mà không có quá nhiều kiến thức, hiểu biết sâu sắc về vật lý và điện tử. Phần cứng của thiết bị đã được tích hợp nhiều chức năng cơ bản và là mã nguồn mở. Ngôn ngữ lập trình trên nền Java lại vô cùng dễ sử dụng tương thích với ngôn ngữ C và hệ thư viện rất phong phú và được chia sẻ miễn phí. Chính vì những lý do như vậy nên Arduino hiện đang dần phổ biến và được phát triển ngày càng mạnh mẽ trên toàn thế giới. Ngày nay, các bộ vi điều khiển đang có ứng dụng ngày càng rộng rãi trong các lĩnh vực kỹ thuật và đời sống xã hội, đặc biệt là trong kỹ thuật tự động hoá và điều khiển từ xa. Giờ đây, với nhu cầu chuyên dụng hoá, tối ưu (thời gian, không gian, giá thành), tính bảo mật, tính chủ động trong công việc... ngày càng đòi hỏi khắt khe. Việc đưa ra công nghệ mới trong lĩnh vực chế tạo mạch điện tử để đáp ứng những yêu cầu trên là hoàn toàn cấp thiết mang tính thực tế cao.

Với nền kinh tế ở vùng nông thôn đô thị đang được đẩy mạnh phát triển. Việc chọn đề tài thiết kế mạch điều khiển nhiệt độ trong lò ấp trứng với mục đích góp phần vào nghiên cứu phát triển ngành nghề đang được sử dụng khá phổ biến. Việc sử dụng lò ấp, máy ấp trong thời gian gần đây khá phổ biến. Nó phù hợp với sự phát triển nhanh chóng của nền chăn nuôi nước nhà. Lò ấp được thiết kế với những công suất và yêu cầu kỹ thuật khác nhau tùy theo yêu cầu về giống gia cầm cần ấp nở cũng như quilmô ấp nở. Có những giống gia cầm cần yêu cầu về kỹ thuật làm lò ấp khá cao như gà trắng (gà công nghiệp)... khi đó việc xây dựng lò ấp cũng phải có những tiêu chuẩn khắt khe hơn như: sự giữ nhiệt, đều nhiệt, giữ ẩm, đều ẩm, cách tạo nhiệt ẩm cho lò... Đối với những giống gia cầm dễ ấp nở hơn như gà ta, gà Sao... thì yêu cầu kỹ thuật của lò cũng đơn giản hơn.

Trong các lò ấp trứng, yêu cầu cần phải cung cấp lượng nhiệt đầy đủ và liên tục thì năng suất của lò mới cao. Nếu trong quá trình ấp trứng mà mất nhiệt thì trứng dễ hỏng. Do vậy cần phải có hệ thống kiểm soát, quản lý xem lò có được cung cấp nhiệt đầy đủ hay không, và có thể biết dễ dàng ngăn trứng nào không được cung cấp nhiệt để sửa chữa

## CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

### 1.1. Tổng quan về lò ấp trứng gà

Trong các lò ấp trứng gia cầm, yêu cầu cần phải cung cấp lượng nhiệt đầy đủ và liên tục và cao thì năng suất và độ chính xác cao nhất của lò mới cao. Nếu trong quá trình ấp trứng mà mất nhiệt thì trứng dễ hỏng. yêu cầu của ấp trứng gia cầm là sai lệch nhiệt độ ấp (tùy từng thời kỳ) không quá 0,1 - 0,2 độ C (Tất nhiên còn nhiều yêu cầu khác nữa như độ ẩm, độ thông thoáng, độ vô trùng....). Mục tiêu là nâng cao tỷ lệ nở và tỷ lệ chính phẩm Do vậy cần phải có hệ thống kiểm soát, quản lý xem lò có được cung cấp nhiệt đầy đủ hay không, và có thể biết dễ dàng ngăn trứng nào không được cung cấp nhiệt để sửa chữa. Vì vậy về nguyên tắc mạch dùng vi điều khiển là hay nhất có thể đáp ứng được các yêu cầu. Hiện nay tất cả các máy ấp công nghiệp đều dùng theo cách này. Chính vì vậy chúng em đã quyết định thiết kế hệ thống đo và điều khiển nhiệt độ sử dụng cảm biến kết hợp với vi xử lý.

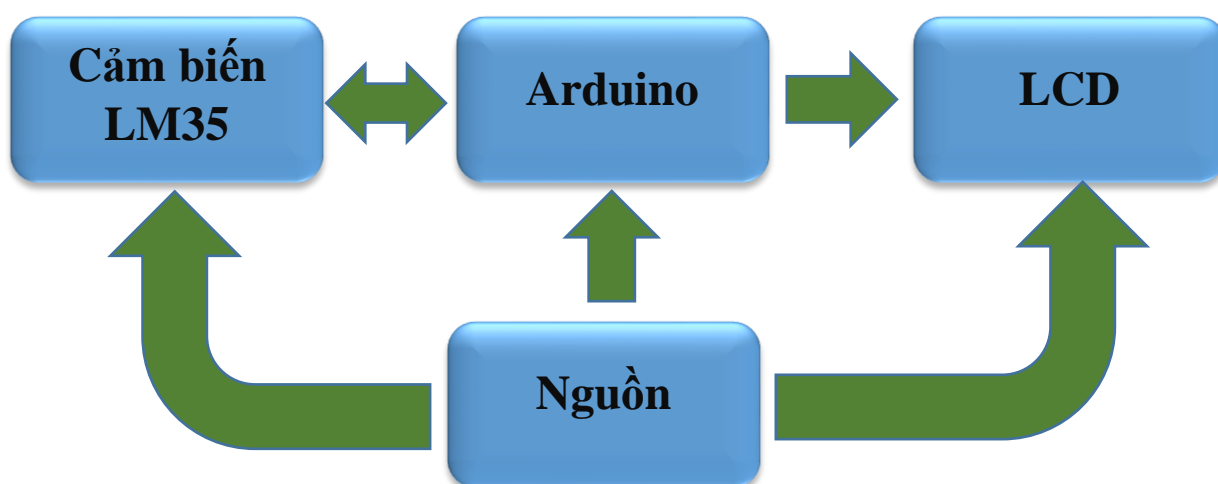
### 1.2. Sơ lược về hệ thống và sơ đồ khối của hệ thống

#### 1.2.1. Sơ lược hệ thống

Bộ mạch được điều khiển bởi vi điều khiển ADUINO đóng vai trò điều khiển và nhập xuất dữ liệu từ các thiết bị giao tiếp với nó, điển hình như cảm biến nhiệt độ LM35 và xuất nhập dữ liệu đọc nhiệt độ từ cảm biến sau đó giao tiếp với thiết bị LCD mã hoá dữ liệu để hiển thị kết quả ra màn hình với dữ liệu của cảm biến.

Phần cứng sau khi thi công hoàn thành thì vi điều khiển được nạp chương trình qua các lệnh được thiết kế bằng ngôn ngữ C trên Arduino IDE như những tập lệnh điều khiển vi điều khiển để mạch hoạt động một cách trơn tru.

#### 1.2.2. Sơ đồ khối hệ thống



## CHƯƠNG 2: XÂY DỰNG MÔ HÌNH HỆ THỐNG VÀ CÁC THIẾT BỊ CHỨC NĂNG

Để tạo ra một máy ấp trứng phải đáp ứng các yêu cầu cơ bản sau:

- Bảo đảm nhiệt độ thích hợp ( $37-38^{\circ}\text{C}$ ) và tương đối ổn định theo từng giai đoạn của mỗi đợt ấp;
- Bảo đảm thông gió thoáng khí nơi ủ ấp;

### 2.1. Giới thiệu chung về Arduino

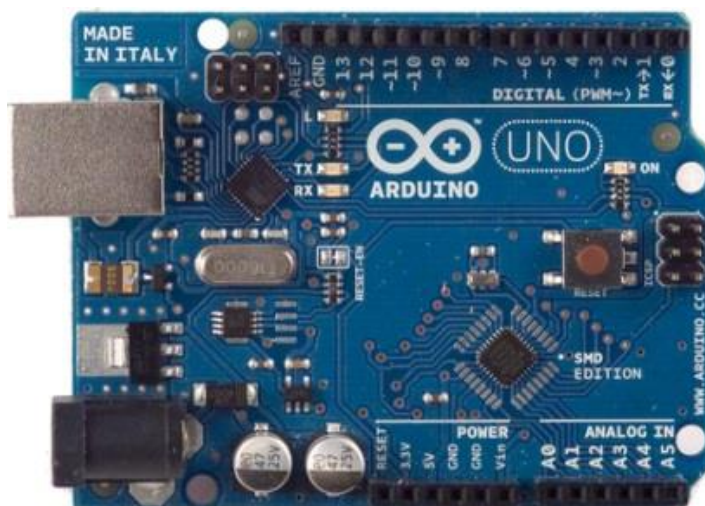
Arduino thực sự đã gây sóng gió trên thị trường người dùng DIY trên toàn thế giới trong vài năm gần đây, gần giống với những gì Apple đã làm được trên thị trường thiết bị di động. Số lượng người dùng cực lớn và đa dạng với trình độ trải rộng từ bậc phổ thông lên đến đại học đã làm cho ngay cả những người tạo ra chúng phải ngạc nhiên về mức độ phổ biến.

Arduino là gì mà có thể khiến ngay cả những sinh viên và nhà nghiên cứu tại các trường đại học danh tiếng như MIT, Stanford, Carnegie Mellon phải sử dụng; hoặc ngay cả Google cũng muốn hỗ trợ khi cho ra đời bộ kit Arduino Mega ADK dùng để phát triển các ứng dụng Android tương tác với cảm biến và các thiết bị khác. Arduino thật ra là một bo mạch vi xử lý được dùng để lập trình tương tác với các thiết bị phần cứng như cảm biến, động cơ, đèn hoặc các thiết bị khác. Đặc điểm nổi bật của Arduino là môi trường phát triển ứng dụng cực kỳ dễ sử dụng, với một ngôn ngữ lập trình có thể học một cách nhanh chóng ngay cả

Với người ít am hiểu về điện tử và lập trình. Và điều làm nên hiện tượng Arduino chính là mức giá rất thấp và tính chất nguồn mở từ phần cứng tới phần mềm. Chỉ với khoảng \$30, người dùng đã có thể sở hữu một board Arduino có 20 ngõ I/O có thể tương tác và điều khiển chừng ấy thiết bị Arduino ra đời tại thị trấn Ivrea thuộc nước Ý và được đặt theo tên một vị vua vào thế kỷ thứ 9 là King Arduino. Arduino chính thức được đưa ra giới thiệu vào năm 2005 như là một công cụ khiêm tốn dành cho các sinh viên của giáo sư Massimo Banzi, là một trong những người phát triển Arduino, tại trường Interaction Design Institute Ivrea (IDII). Mặc dù hầu như không được tiếp thị gì cả, tin tức về Arduino vẫn lan truyền với tốc độ chóng mặt nhờ những lời truyền miệng tốt đẹp của những người dùng đầu tiên. Hiện nay Arduino nổi tiếng tới nỗi có người tìm đến thị trấn Ivrea chỉ để tham quan nơi đã sản sinh ra Arduino.

## 2.2. Boar Arduino UNO R3

Nhắc tới dòng mạch Arduino dùng để lập trình, cái đầu tiên mà người ta thường nói tới chính là dòng Arduino UNO. Hiện dòng mạch này đã phát triển tới thế hệ thứ 3 (R3). Bạn sẽ bắt đầu đến với Arduino qua thứ này. Bạn có thể dùng Arduino Nano cũng được nhưng mình khuyên bạn nên dùng cái này.

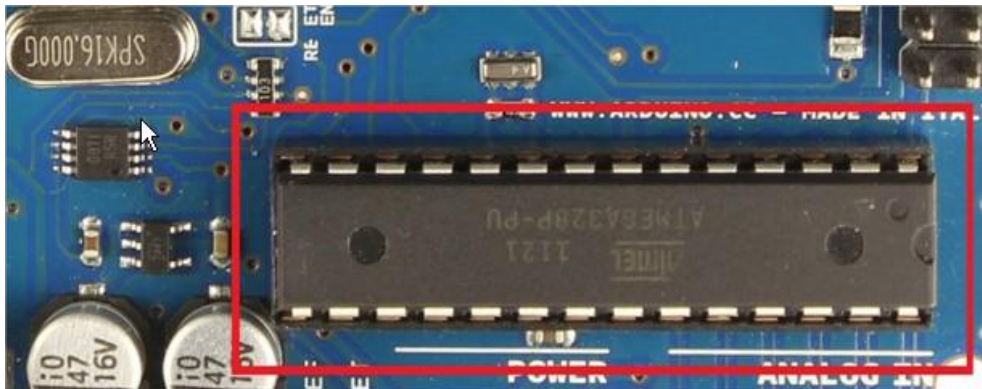


### ✚ Một vài thông số của Arduino

Vi điều khiển	ATmega328 họ 8bit
Điện áp hoạt động	5V DC (chỉ được cấp qua cổng USB)
Tần số hoạt động	16 MHz
Dòng tiêu thụ	khoảng 30mA
Điện áp vào khuyên dùng	7-12V DC
Điện áp vào giới hạn	6-20V DC
Số chân Digital I/O	14 (6 chân hardware PWM)
Số chân Analog	6 (độ phân giải 10bit)
Dòng tối đa trên mỗi chân I/O	30 mA
Dòng ra tối đa (5V)	500 mA
Dòng ra tối đa (3.3V)	50 mA
Bộ nhớ flash	32 KB (ATmega328) với 0.5KB dùng bởi bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)



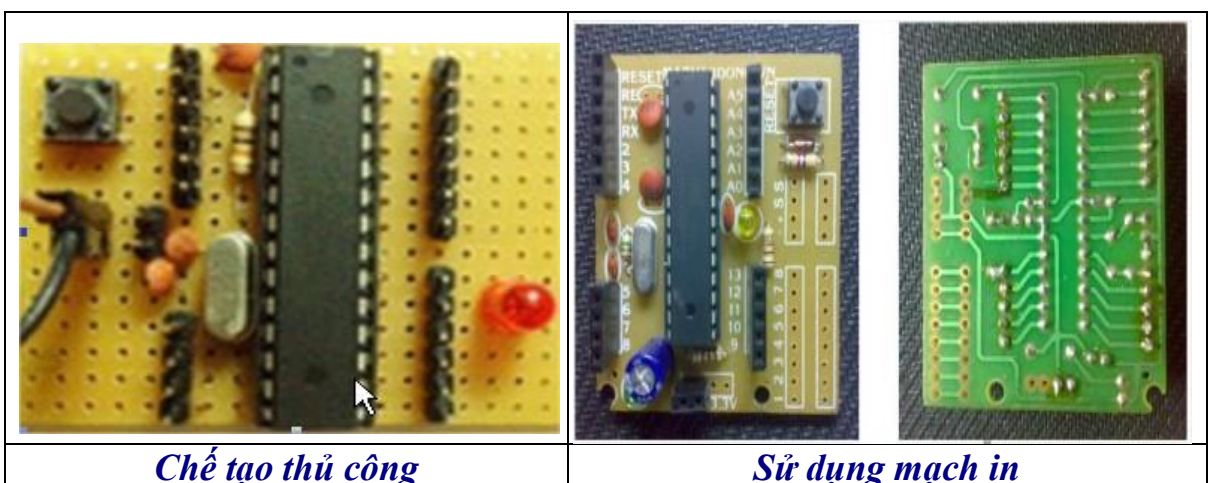
## Vi điều khiển



Arduino UNO có thể sử dụng 3 vi điều khiển họ 8bit AVR là ATmega8, ATmega168, ATmega328. Bộ não này có thể xử lý những tác vụ đơn giản như điều khiển đèn LED nhấp nháy, xử lý tín hiệu cho xe điều khiển từ xa, làm một trạm đo nhiệt độ - độ ẩm và hiển thị lên màn hình LCD,....

Thiết kế tiêu chuẩn của Arduino UNO sử dụng vi điều khiển ATmega328 với giá khoảng 90.000đ. Tuy nhiên nếu yêu cầu phần cứng của bạn không cao hoặc túi tiền không cho phép, bạn có thể sử dụng các loại vi điều khiển khác có chức năng tương đương nhưng rẻ hơn như ATmega8 (bộ nhớ flash 8KB) với giá khoảng 45.000đ hoặc ATmega168 (bộ nhớ flash 16KB) với giá khoảng 65.000đ.

Ngoài việc dùng cho board Arduino UNO, bạn có thể sử dụng những IC điều khiển này cho các mạch tự chế. Vì sao ? Vì bạn chỉ cần board Arduino UNO để lập trình cho vi điều khiển. Trên thực tế, bạn không cần phải dùng Arduino UNO trên các sản phẩm của mình, thay vào đó là các mạch tự chế để giảm chi phí như hình dưới đây:





## **Năng lượng**

Arduino UNO có thể được cấp nguồn 5V thông qua cổng USB hoặc cấp nguồn ngoài với điện áp khuyến dùng là 7-12V DC và giới hạn là 6-20V. Thường thì cấp nguồn bằng pin vuông 9V là hợp lý nhất nếu bạn không có sẵn nguồn từ cổng USB. Nếu cấp nguồn vượt quá ngưỡng giới hạn trên, bạn sẽ làm hỏng Arduino UNO.

## **Các chân năng lượng**

- **GND (Ground):** cực âm của nguồn điện cấp cho Arduino UNO. Khi bạn dùng các thiết bị sử dụng những nguồn điện riêng biệt thì những chân này phải được nối với nhau.
- **5V:** cấp điện áp 5V đầu ra. Dòng tối đa cho phép ở chân này là 500mA.
- **3.3V:** cấp điện áp 3.3V đầu ra. Dòng tối đa cho phép ở chân này là 50mA.
- **Vin (Voltage Input):** để cấp nguồn ngoài cho Arduino UNO, bạn nối cực dương của nguồn với chân này và cực âm của nguồn với chân GND.
- **IOREF:** điện áp hoạt động của vi điều khiển trên Arduino UNO có thể được đo ở chân này. Và dĩ nhiên nó luôn là 5V. Mặc dù vậy bạn không được lấy nguồn 5V từ chân này để sử dụng bởi chức năng của nó không phải là cấp nguồn.
- **RESET:** việc nhấn nút Reset trên board để reset vi điều khiển tương đương với việc chân RESET được nối với GND qua 1 điện trở 10KΩ.

## **Lưu ý:**

- ❖ Arduino UNO không có bảo vệ cắm ngược nguồn vào. Do đó bạn phải hết sức cẩn thận, kiểm tra các cực âm – dương của nguồn trước khi cấp cho Arduino UNO. Việc làm chập mạch nguồn vào của Arduino UNO sẽ biến nó thành một miếng nhựa chặn giấy. mình khuyên bạn nên dùng nguồn từ cổng USB nếu có thể.
- ❖ Các chân 3.3V và 5V trên Arduino là các chân dùng để cấp nguồn ra cho các thiết bị khác, không phải là các chân cấp nguồn vào. Việc cấp nguồn sai vị trí có thể làm hỏng board. Điều này không được nhà sản xuất khuyến khích.
- ❖ Cấp nguồn ngoài không qua cổng USB cho Arduino UNO với điện áp dưới 6V có thể làm hỏng board.
- ❖ Cấp điện áp trên 13V vào chân RESET trên board có thể làm hỏng vi điều khiển ATmega328.
- ❖ Cường độ dòng điện vào/ra ở tất cả các chân Digital và Analog của Arduino UNO nếu vượt quá 200mA sẽ làm hỏng vi điều khiển.

- ❖ Cấp điện áp trên 5.5V vào các chân Digital hoặc Analog của Arduino UNO sẽ làm hỏng vi điều khiển.
- ❖ Cường độ dòng điện qua một chân Digital hoặc Analog bất kì của Arduino UNO vượt quá 40mA sẽ làm hỏng vi điều khiển. Do đó nếu không dùng để truyền nhận dữ liệu, bạn phải mắc một điện trở hạn dòng.

Khi mình nói rằng bạn “*có thể làm hỏng*”, điều đó có nghĩa là chưa chắc sẽ hỏng ngay bởi các thông số kĩ thuật của linh kiện điện tử luôn có một sự tương đối nhất định. Do đó hãy cứ tuân thủ theo những thông số kĩ thuật của nhà sản xuất nếu bạn không muốn phải mua một board Arduino UNO thứ 2. Khi mình nói rằng bạn “*có thể làm hỏng*”, điều đó có nghĩa là chưa chắc sẽ hỏng ngay bởi các thông số kĩ thuật của linh kiện điện tử luôn có một sự tương đối nhất định. Do đó hãy cứ tuân thủ theo những thông số kĩ thuật của nhà sản xuất nếu bạn không muốn phải mua một board Arduino UNO thứ 2.

### **Bộ nhớ**

Vi điều khiển Atmega328 tiêu chuẩn cung cấp cho người dùng:

- ❖ **32KB bộ nhớ Flash:** những đoạn lệnh bạn lập trình sẽ được lưu trữ trong bộ nhớ Flash của vi điều khiển. Thường thì sẽ có khoảng vài KB trong số này sẽ được dùng cho bootloader nhưng đừng lo, bạn hiếm khi nào cần quá 20KB bộ nhớ này đâu.
- ❖ **2KB cho SRAM (Static Random Access Memory):** giá trị các biến bạn khai báo khi lập trình sẽ lưu ở đây. Bạn khai báo càng nhiều biến thì càng cần nhiều bộ nhớ RAM. Tuy vậy, thực sự thì cũng hiếm khi nào bộ nhớ RAM lại trở thành thứ mà bạn phải bận tâm. Khi mất điện, dữ liệu trên SRAM sẽ bị mất.
- ❖ **1KB cho EEPROM (Electrically Erasable Programmable Read Only Memory):** đây giống như một chiếc ổ cứng mini – nơi bạn có thể đọc và ghi dữ liệu của mình vào đây mà không phải lo bị mất khi cúp điện giống như dữ liệu trên SRAM.

### **Các cổng vào/ra**

Arduino UNO có 14 chân digital dùng để đọc hoặc xuất tín hiệu. Chúng chỉ có 2 mức điện áp là 0V và 5V với dòng vào/ra tối đa trên mỗi chân là 40mA. Ở mỗi chân đều có các điện trở pull-up từ được cài đặt ngay trong vi điều khiển ATmega328 (mặc định thì các điện trở này không được kết nối).

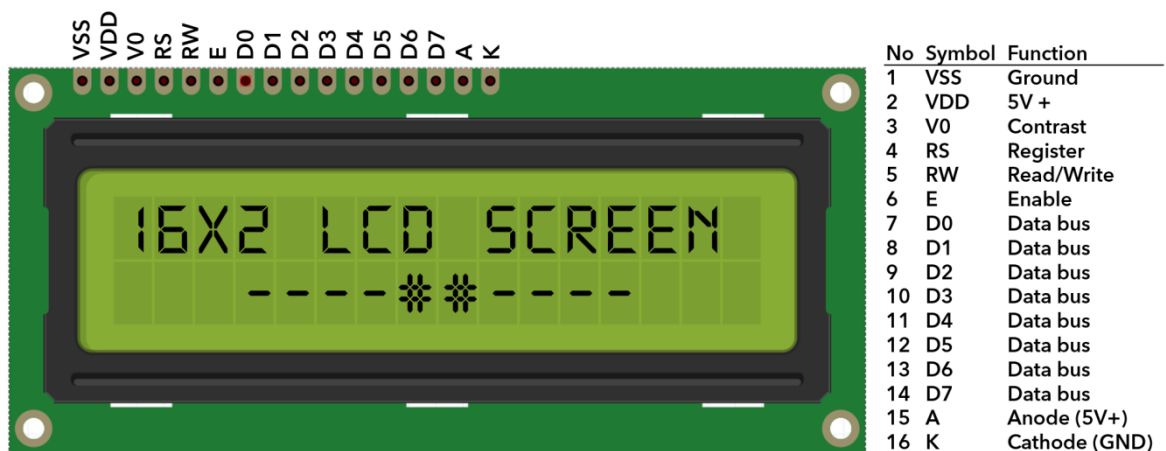
Một số chân digital có các chức năng đặc biệt như sau:

- ❖ **2 chân Serial:** 0 (RX) và 1 (TX): dùng để gửi (transmit – TX) và nhận (receive – RX) dữ liệu TTL Serial. Arduino Uno có thể giao tiếp với thiết bị khác thông qua 2 chân này. Kết nối bluetooth thường thấy nói nôm na chính là kết nối Serial không dây. Nếu không cần giao tiếp Serial, bạn không nên sử dụng 2 chân này nếu không cần thiết
- ❖ **Chân PWM (~): 3, 5, 6, 9, 10, và 11:** cho phép bạn xuất ra xung PWM với độ phân giải 8bit (giá trị từ  $0 \rightarrow 2^8-1$  tương ứng với  $0V \rightarrow 5V$ ) bằng hàm `analogWrite()`. Nói một cách đơn giản, bạn có thể điều chỉnh được điện áp ra ở chân này từ mức 0V đến 5V thay vì chỉ cố định ở mức 0V và 5V như những chân khác.
- ❖ **Chân giao tiếp SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ngoài các chức năng thông thường, 4 chân này còn dùng để truyền phát dữ liệu bằng giao thức SPI với các thiết bị khác.
- ❖ **LED 13:** trên Arduino UNO có 1 đèn led màu cam (kí hiệu chữ L). Khi bấm nút Reset, bạn sẽ thấy đèn này nhấp nháy để báo hiệu. Nó được nối với chân số 13. Khi chân này được người dùng sử dụng, LED sẽ sáng.

Arduino UNO có 6 chân analog ( $A0 \rightarrow A5$ ) cung cấp độ phân giải tín hiệu 10bit ( $0 \rightarrow 2^{10}-1$ ) để đọc giá trị điện áp trong khoảng  $0V \rightarrow 5V$ . Với chân **AREF** trên board, bạn có thể để đưa vào điện áp tham chiếu khi sử dụng các chân analog. Tức là nếu bạn cấp điện áp 2.5V vào chân này thì bạn có thể dùng các chân analog để đo điện áp trong khoảng từ  $0V \rightarrow 2.5V$  với độ phân giải vẫn là 10bit.

Đặc biệt, Arduino UNO có 2 chân A4 (SDA) và A5 (SCL) hỗ trợ giao tiếp I2C/TWI với các thiết bị khác.

### 2.3. LCD 1602



1. VSS: tương đương với GND - cực âm
2. VDD: tương đương với VCC - cực dương (5V)
3. Contrast Voltage (Vo): điều khiển độ sáng màn hình
4. Register Select (RS): điều khiển địa chỉ nào sẽ được ghi dữ liệu
5. Read/Write (RW): Bạn sẽ đọc (read mode) hay ghi (write mode) dữ liệu? Nó sẽ phụ thuộc vào bạn gửi giá trị gì vào.
6. Enable pin: Cho phép ghi vào LCD
7. D0 - D7: 8 chân dữ liệu, mỗi chân sẽ có giá trị HIGH hoặc LOW nếu bạn đang ở chế độ đọc (read mode) và nó sẽ nhận giá trị HIGH hoặc LOW nếu đang ở chế độ ghi (write mode)
8. Backlight (Backlight Anode (+) và Backlight Cathode (-)): Tắt bật đèn màn hình LCD.

### **Các thanh ghi**

- ❖ **Thanh ghi IR:** Mỗi lệnh được nhà sản xuất LCD đánh địa chỉ rõ ràng. Người dùng chỉ việc cung cấp địa chỉ lệnh bằng cách nạp vào thanh ghi IR. Ví dụ: Lệnh “hiển thị màn hình và con trỏ” có mã lệnh là 00001110
- ❖ **Thanh ghi DR :** Thanh ghi DR dùng để chứa dữ liệu 8 bit để ghi vào vùng RAM DDRAM hoặc CGRAM ( ở chế độ ghi) hoặc dùng để chứa dữ liệu từ 2 vùng RAM này gửi ra cho MPU (ở chế độ đọc).
- ❖ **Cờ báo bận BF:** (Busy Flag) Khi đang thực thi các hoạt động bên trong, LCD bỏ qua mọi giao tiếp với bên ngoài và bật cờ BF( thông qua chân DB7 khi có thiết lập RS=0, R/W=1) lên để cho biết nó đang “bận”.
- ❖ **Bộ đếm địa chỉ AC :** (Address Counter) Khi một địa chỉ lệnh được nạp vào thanh ghi IR, thông tin được nối trực tiếp cho 2 vùng RAM (việc chọn lựa vùng RAM tương tác đã được bao hàm trong mã lệnh).

Sau khi ghi vào (đọc từ) RAM, bộ đếm AC tự động tăng lên (giảm đi) 1 đơn vị.

### **Bộ nhớ LCD**

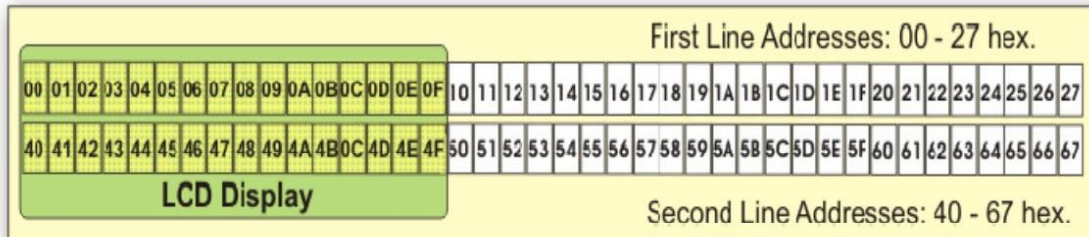
**Vùng RAM hiển thị DDRAM :** (Display Data RAM)

**Vùng CGROM:** Character Generator ROM Vùng CGRAM : (Character Generator RAM)

### **DDRAM**

Đây là vùng RAM dùng để hiển thị, nghĩa là ứng với một địa chỉ của RAM là một ô kí tự trên màn hình.

## DDRAM Memory



## CGROM

Chứa các mẫu kí tự loại 5x7 hoặc 5x10 điểm ảnh/kí tự, và định địa chỉ bằng 8 bit.

		4 higher bits of address															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
4 lower bits of address	xxxx0000	CG RAM (5)		0	0	P	`	P						-	タ	ミ	αp
	xxxx0001 (2)			!	1	A	Q	a	q					。	ア	チ	ㄥăq
	xxxx0010 (3)			"	2	B	R	b	r					「	イ	ツ	×βθ
	xxxx0011 (4)			#	3	C	S	c	s					」	ウ	テ	ε∞
	xxxx0100 (5)			\$	4	D	T	d	t					、	エ	ト	μΩ
	xxxx0101 (6)			%	5	E	U	e	u					・	オ	ナ	1σÜ
	xxxx0110 (7)			&	6	F	V	f	v					ヲ	カ	ニ	ヨρΣ
	xxxx0111 (8)			'	7	G	W	g	w					ア	キ	ヌ	ラgπ
	xxxx1000 (1)			(	8	H	X	h	x					イ	ク	ネ	リJヌ
	xxxx1001 (2)			)	9	I	Y	i	y					ウ	ケ	ル	リ-y
	xxxx1010 (3)			*	:	J	Z	j	z					エ	コ	ハ	レjチ
	xxxx1011 (4)			+	;	K	[	k	<					オ	サ	ヒ	ロ*ヲ
	xxxx1100 (5)			,	<	L	¥	l	l					カ	シ	フ	ワΦ円
	xxxx1101 (6)			-	=	M	]	m	}					ユ	ズ	ヘ	ンも÷
	xxxx1110 (7)			.	>	N	^	n	→					ヨ	セ	ホ	°ん
	xxxx1111 (8)			/	?	O	_	o	←					ッ	ソ	マ	°ö■



Mẫu kí tự đồ họa riêng.

**Table 5** Relationship between CGRAM Addresses, Character Codes (DDRAM) and Character Patterns (CGRAM Data)

**For  $5 \times 8$  dot character patterns**

[illegible]

- Notes:
1. Character code bits 0 to 2 correspond to CGRAM address bits 3 to 5 (3 bits: 8 types).
  2. CGRAM address bits 0 to 2 designate the character pattern line position. The 8th line is the cursor position and its display is formed by a logical OR with the cursor.  
Maintain the 8th line data, corresponding to the cursor display position, at 0 as the cursor display.  
If the 8th line data is 1, 1 bits will light up the 8th line regardless of the cursor presence.
  3. Character pattern row positions correspond to CGRAM data bits 0 to 4 (bit 4 being at the left).
  4. As shown Table 5, CGRAM character patterns are selected when character code bits 4 to 7 are all 0. However, since character code bit 3 has no effect, the R display example above can be selected by either character code 00H or 08H.
  5. 1 for CGRAM data corresponds to display selection and 0 to non-selection.
- \* Indicates no effect.

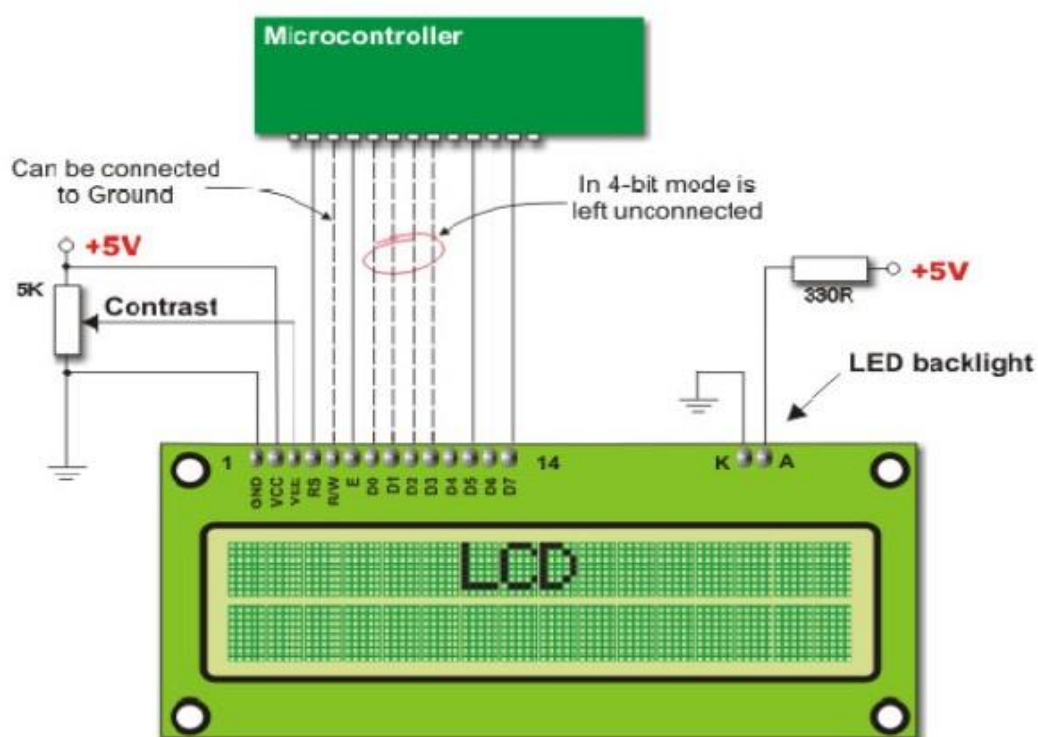
## Tập lệnh:

Tập lệnh	Mã nhị phân										Mô tả	Thời gian thực thi
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Xoá hiển thị	0	0	0	0	0	0	0	0	0	1	Xoá hiển thị và đưa con trỏ về vị trí ban đầu (địa chỉ 0).	1.64mS
Con trỏ ban đầu	0	0	0	0	0	0	0	0	1	*	Trả con trỏ về vị trí ban đầu (địa chỉ 0). Ngoài ra đưa hiển thị đã bị dịch chuyển về vị trí ban đầu. Nội dung bộ nhớ hiển thị dữ liệu(DDRAM) không thay đổi.	1.64mS
Thiết lập chế độ	0	0	0	0	0	0	0	1	I/D	S	Thiết lập hướng di chuyển của con trỏ tăng/giảm(I/D=0:giảm,I/D=1:tăng), chỉ rõ dịch chuyển hiển thị (S=0:không dịch chuyển hiển thị,S=1:dịch chuyển hiển thị). Hoạt động này được thực hiện trong suốt quá trình đọc/ghi dữ liệu	40uS
Điều khiển hiển thị	0	0	0	0	0	0	1	D	C	B	Bật/tắt hiển thị (D=0:tắt,D=1:bật) nhưng dữ liệu vẫn lưu trong DDRAM, bật/tắt con trỏ(C=0:tắt,C=1:bật) và bật tắt con trỏ nhấp nháy tại vị trí của kí tự (B=0:tắt,B=1:bật).	40uS
Dịch chuyển con trỏ/hiển thị	0	0	0	0	0	1	S/C	R/L	*	*	Dịch chuyển con trỏ/hiển thị qua trái/phải mà không phải đọc/ghi lại dữ liệu (S/C=0:di chuyển con trỏ,S/C=1:di chuyển hiển thị), (R/L=0:dịch trái,R/L=1:dịch phải). nội dung DDRAM không thay đổi.	40uS
Thiết lập chức năng	0	0	0	0	1	DL	N	F	*	*	Khởi tạo giao diện của độ dài dữ liệu (DL=0:độ dài 4 bit,DL=1:8 bit), số hàng hiển thị(N=0:1 hàng ,N=1 :2 hàng) và phông chữ(F=0:5x7,F=1:5x10).	40uS
Thiết lập địa chỉ CGRAM	0	0	0	1	Địa chỉ CGRAM						Thiết lập địa chỉ bộ nhớ tạo kí tự (CGRAM), dữ liệu được gửi/nhận sau thiết lập này	40uS

Tập lệnh	Mã nhị phân										Mô tả	Thời gian thực thi
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Thiết lập địa chỉ DDRAM	0	0	1	Địa chỉ DDRAM							Thiết lập địa chỉ bộ nhớ tạo kí tự (DDRAM), dữ liệu được gửi/nhận sau thiết lập này	40uS
Đọc cờ bận	0	1	BF	Địa chỉ CGRAM / DDRAM							Đọc cờ bận Busy-flag (BF),kiểm tra xem hệ thống có đang thực thi 1 lệnh đã được nhận trước đó không.(BF=1:hệ thống đang thực hiện tác vụ bên trong,khi BF=0 thì lệnh tiếp theo mới được thực thi)	0uS
Ghi dữ liệu đến CGRAM/DDRAM	1	0		Ghi dữ liệu							Ghi dữ liệu đến CGRAM/DDRAM.	40uS
Đọc dữ liệu từ CGRAM/DDRAM	1	1		Đọc dữ liệu							Đọc dữ liệu từ CGRAM/DDRAM.	40uS



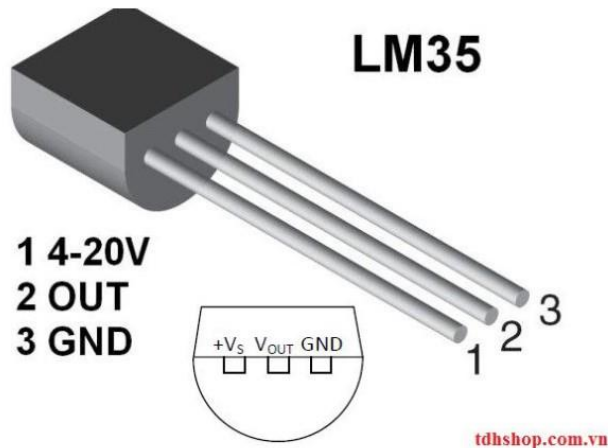
Mã (Hex)	Lệnh đến thanh ghi của LCD
1	Xoá màn hình hiển thị
2	Trở về đầu dòng
4	Giả con trỏ (dịch con trỏ sang trái)
6	Tăng con trỏ (dịch con trỏ sang phải)
5	Dịch hiển thị sang phải
7	Dịch hiển thị sang trái
8	Tắt con trỏ, tắt hiển thị
A	Tắt hiển thị, bật con trỏ
C	Bật hiển thị, tắt con trỏ
E	Bật hiển thị, nhấp nháy con trỏ
F	Tắt con trỏ, nhấp nháy con trỏ
10	Dịch vị trí con trỏ sang trái
14	Dịch vị trí con trỏ sang phải
18	Dịch toàn bộ hiển thị sang trái



## 2.4. Cảm biến LM35

Cảm biến nhiệt độ LM35 là một loại *cảm biến tương tự* rất hay được ứng dụng trong các ứng dụng đo nhiệt độ thời gian thực. Vì nó hoạt động khá chính xác với sai số nhỏ, đồng thời với kích thước nhỏ và giá thành rẻ là một trong những ưu điểm của nó. Vì đây là cảm biến tương tự (analog sensor) nên ta có thể dễ dàng đọc được giá trị của nó bằng hàm `analogRead()`.

**Giới thiệu về cảm biến LM35:**



**LM35 là một cảm biến nhiệt độ analog:**

Nhiệt độ được xác định bằng cách đo hiệu điện thế ngõ ra của LM35.

→ Đơn vị nhiệt độ: °C.

→ Nhiệt độ thay đổi tuyến tính: 10mV/°C

TO-92  
Plastic Package



BOTTOM VIEW  
DS005516-2

Order Number LM35CZ,  
LM35CAZ or LM35DZ  
See NS Package Number Z03A

*Sơ đồ chân của LM35*

**LM35 không cần phải chỉnh nhiệt độ khi sử dụng.**

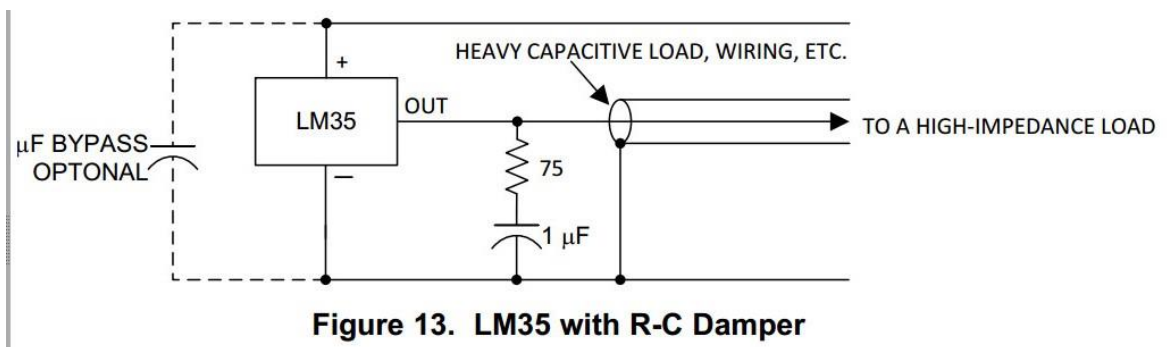
**Cảm biến LM35 hoạt động bằng cách cho ra một giá trị hiệu điện thế nhất định tại chân Vout (chân giữa) ứng với mỗi mức nhiệt độ.**

Như vậy, bằng cách đưa vào chân bên trái của cảm biến LM35 hiệu điện thế 5V, chân phải nối đất, đo hiệu điện thế ở chân giữa bằng các pin A0 trên arduino (giống y hệt cách đọc giá trị biến trở), bạn sẽ có được nhiệt độ (0-100°C) bằng công thức:

```
float temperature = (5.0*analogRead(A0)*100.0/1024.0);
```

Với LM35, bạn có thể tự tạo cho mình mạch cảm biến nhiệt độ sử dụng LM35 và tự động ngắt điện khi nhiệt độ vượt ngưỡng tối đa, đóng điện khi nhiệt độ thấp hơn ngưỡng tối thiểu thông qua module rơ le...

**LM35 thay đổi nhiệt độ nhanh và chính xác.**



**Sơ đồ nối dây:**



## CHƯƠNG 3: MÔ PHỎNG

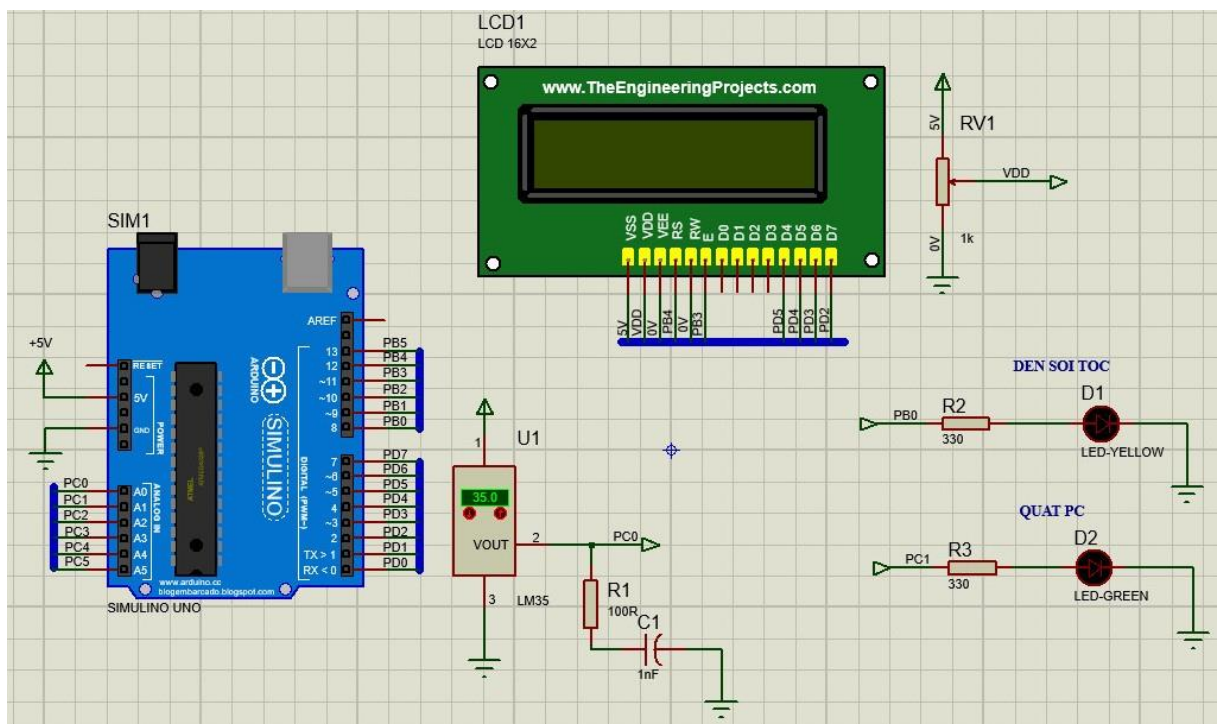
### Mô phỏng trên phần mềm Protues.

Phần mềm protues là phần mềm chuyên dụng cho việc thiết kế mạch điện tử, cũng như mô phỏng các đề tài về điện tử. Một số ưu điểm của phần mềm này như:

- Thư viện linh kiện phong phú.
- Hỗ trợ các thiết bị đo
- Hỗ trợ thiết kế mạch in.

Vì vậy Protues là phần mềm có ưu điểm vượt trội để dễ dàng hỗ trợ thực hiện các thiết kế, mô phỏng trước khi lắp đặt nhằm khắc phục tối đa các lỗi có thể xảy ra.

### Sơ đồ Mạch Mô Phỏng



✓ **Code chương Trình:**

```
#include <LiquidCrystal.h>
#include <Arduino.h>
#include <Wire.h>

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
int den=8;
int quat=A1;
void setup() {
  //câu hình thành ghi DDRB output
  DDRB=DDRB|B0000001;
  DDRC=DDRC|B0000010;
  // Cấu hình chân analog kết nối tới cảm biến LM35 INPUT
  DDRC=DDRC&B1111110;

  pinMode(den,OUTPUT);
  pinMode(quat,OUTPUT);
  lcd.begin(16,2);

  lcd.print("May Ap Trung Ga");
  lcd.setCursor(0,1);
  Serial.begin(9600);
}
void loop(){
  delay(1000);
  lcd.clear();
  int NhietDo;
  while (1){
    /*
      * theo quy tac tam suat ta co
      * cứ 5000mv thi tuong ung 1024 adc(2^10-1)
      * vậy điện áp reading
      * dien ap=reading*5000/1024
      * theo datasheet ta co
      * cu 10mV thi tuong ung 1 do C
      * vay dien ap          nhiet do
```

```

*  => nhietdo=dienap/10
*  hay nhietdo=(reading*5000/1024)/10
*  hay nhietdo=(reading*500/1024)
*/

//đọc giá trị từ cảm biến LM35
int reading = analogRead(PINC&B00000001);
float temp=(((float)reading*500.00)/1024.00);
NhietDo=(float)temp;
lcd.setCursor(0,0);
lcd.print("NHIET DO:");
lcd.print(NhietDo);
lcd.write(233);
lcd.print("C");
if(NhietDo>37){
    lcd.setCursor(0,1);
    lcd.print("WARRING!:HOT  ");
    PORTC|=B00000010;//BAT QUAT
    digitalWrite(den,LOW);//Tắt đèn khi nhiệt độ lên cao hơn 37°C
    delay(2000);
}
if(NhietDo<37){
    lcd.setCursor(0,1);
    lcd.print("WARRING!:COOL  ");
    PORTB|=B00000001;//BẬT ĐÈN
    digitalWrite(quat,LOW);//Tắt quạt khi nhiệt độ bé hơn 37 độ C
    delay(2000);
}
if((NhietDo==37)&&(NhietDo<38)){

```

```

        lcd.setCursor(0,1);
        lcd.print("BINH THUONG ");
        PORTB|=B00000001;//BẬT ĐÈN
        digitalWrite(quat,LOW);//Tắt quạt khi nhiệt độ bình thường
        delay(2000);
    }
    delay(1000);
}
}

```

✓ Giải thích code:

Tổng quan: chương trình sẽ kiểm tra mức điện áp ngõ ra của LM35 tương ứng với nhiệt độ của môi trường hiển thị trên Lcd và Serial Monitor. Nhiệt độ của môi trường được tính bằng độ C. Bắt đầu chương trình ta khai báo thư viện LCD và định nghĩa chân cũng như các biến cần dùng cho toàn bộ chương trình.

*#include <LiquidCrystal.h>*

*LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // định nghĩa chân cho LCD*

Để tính toán đúng nhiệt độ môi trường chúng ta cần lưu ý những điểm sau:

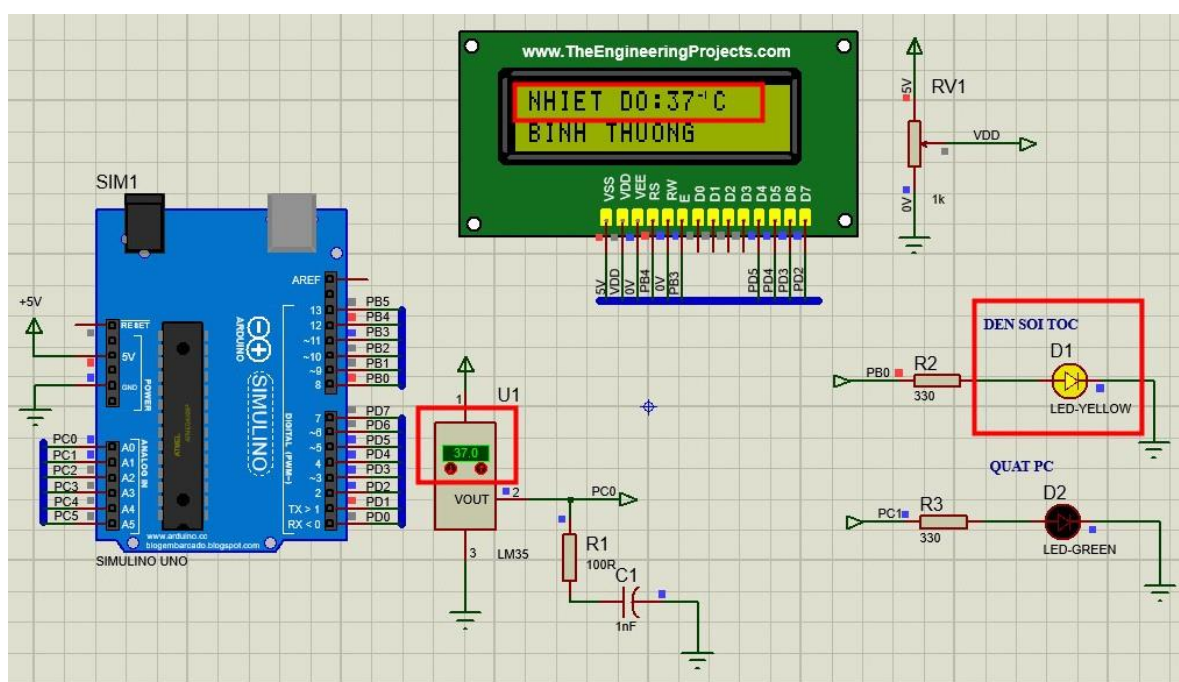
Bộ chuyển đổi ADC gồm 10 bit tức là 1024 mức.

Đối với LM35 thì điện áp ngõ ra tuyến tính với nhiệt độ của môi trường. Hệ số chuyển đổi điện áp sang nhiệt độ là  $10\text{mV}/1^{\circ}\text{C} = 0,01\text{V}/1^{\circ}\text{C}$ . Xây dựng công thức tính nhiệt độ.

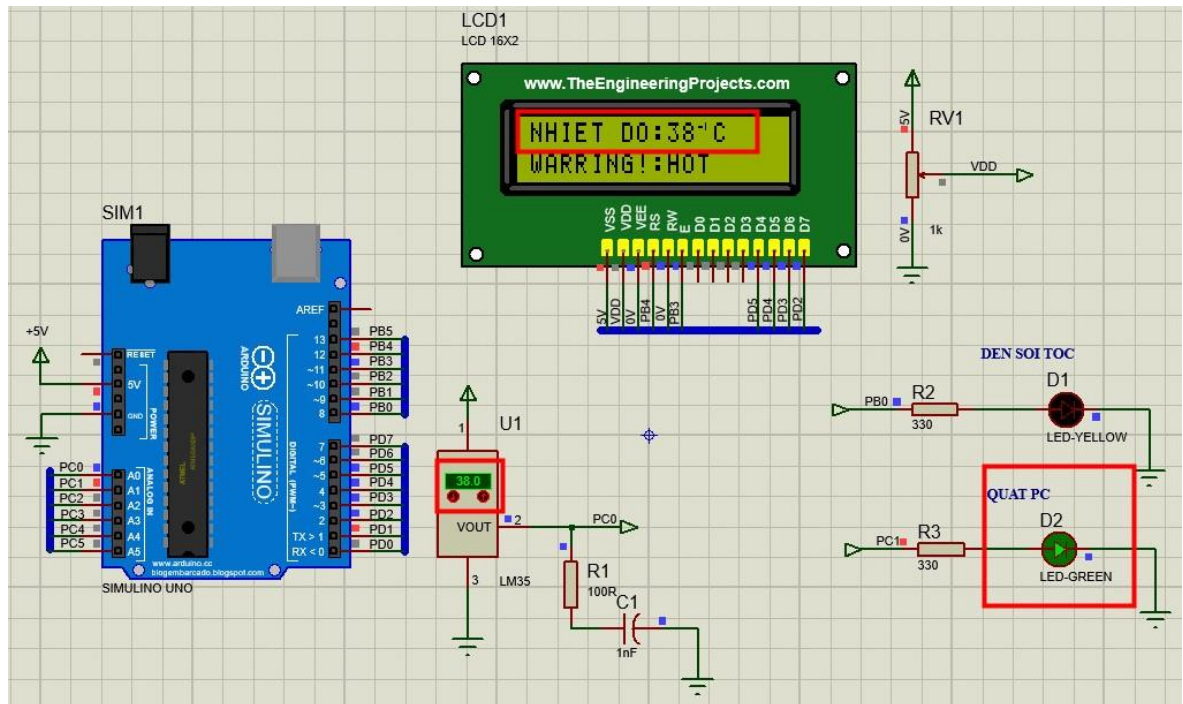
Ta biết: 1,1 volt (1100 mV) có 1024 mức biểu diễn, vậy 1 mức sẽ là  $1,1/1024$  (volt), để chuyển đổi từ điện áp sang nhiệt độ thì ta chia tiếp cho 0,01V. Từ đây ta thấy cứ 1 mức chuyển đổi của ADC tương ứng với  $1,1/1024 \cdot 0,01 = 0,10742188^{\circ}\text{C}$ . Như vậy chúng ta chỉ cần đọc giá trị đầu vào ở chân A0 (giá trị nằm trong khoảng 0 - 1023).



1. Khi  $T \leq 37^{\circ}\text{C}$



2. Khi  $T > 37^{\circ}\text{C}$



## KẾT LUẬN

Qua thời gian làm bài tập lớn với nội dung: **Thiết kế hệ thống điều khiển lò ấp trứng gà** nhóm chúng em đã thiết kế và xây dựng được hệ thống điều khiển lò ấp trứng gà tự động. Thiết kế đo nhiệt độ ( $^{\circ}\text{C}$ ), dựa trên cơ sở đó để điều khiển động cơ hoạt động. VD: trong lò ấp trứng nhiệt độ trên  $37^{\circ}\text{C}$ , thì hệ thống sẽ tự động bật quạt để giảm nhiệt độ xuống để đạt đúng khoảng nhiệt độ yêu cầu để ấp trứng, còn dưới  $37^{\circ}\text{C}$  thì hệ thống tự động bật đèn sưởi đốt hoạt thiết bị làm nóng lên khoảng nhiệt độ yêu cầu. Thiết kế LCD hiển thị đưa ra các thông tin trên LCD đưa thông tin cho người sử dụng các thông tin để điều chỉnh nhiệt độ cho thích hợp điều khiển hệ thống.

Trong quá trình thực hiện, lập trình cho mạch đo nhiệt độ, độ ẩm gặp phải nhiều khó khăn khác nhau như: do phải nghiên cứu nhiều tài liệu nước ngoài, datasheets,... dẫn đến nhiều chỗ dịch sai, dịch nhầm dẫn đến áp dụng các hàm, câu lệnh bị sai ý nghĩa, cấu trúc..., trong quá trình viết code gặp phải nhiều lỗi phát sinh mà không tìm ngay ra nguyên nhân cần đầu tư thời gian để giải quyết, nhiều linh kiện (linh kiện đo độ ẩm và nhiệt độ DHT11, DHT22 hoặc linh kiện cảm biến độ ẩm HS1101, HS220....) rất khó để tìm được thư viện chuẩn để lập trình...

Do thời gian có hạn việc thiết kế hệ thống của em vẫn còn nhiều sai sót. Hệ thống vẫn chưa tối ưu, việc điều khiển từ xa lấy và phát thông tin chưa thành công. Phát triển các dữ liệu trên máy tính chưa phát huy được hết các tính năng tốt nhất. Em rất mong được sự ủng hộ và giúp đỡ của thầy để đề tài chúng em thực hiện được hoàn thiện hơn và có thêm nhiều cải tiến đáng kể và ứng dụng tốt hơn vào thực tiễn.

## TÀI LIỆU THAM KHẢO

1. Massimo Banzi (2009), *Getting Started with Arduino*, O'Reilly Media.
2. Michael Margollis and Nicholas Weldin (2009), *Arduino Cookbook*, O'Reilly Media.
3. <http://arduino.vn/>
4. <http://arduino.cc/>