|  |
| --- |
| TRƯỜNG ĐẠI HỌC TÂY NGUYÊN  **KHOA KHTN&CN**  Tuyển sinh Sau Đại học – Trang tuyển sinh – Trường Đại học Tây Nguyên  **BÁO CÁO ĐỒ ÁN**  **LẬP TRÌNH ARDUINO DÙNG LED 7 ĐOẠN**  **HIỂN THỊ NHIỆT ĐỘ**  **Sinh viên thực hiện:**  **Nguyễn Văn Huy**  **Trần Gia Bảo**  **Đỗ Thị Hồng Ngọc**  **Trần Văn Lâm**  **Trần Thanh Ngân**  **Đỗ Xuân Tiến**  **Lớp: Công nghệ thông tin K18**  **Môn học: Lập trình ứng dụng mã nguồn mở**  **Giảng viên hướng dẫn: ThS. Đoàn Vũ Thịnh**  **Đắk Lắk, tháng 9 năm 2021** |

**MỤC LỤC**

[LỜI NÓI ĐẦU 1](#_Toc85347984)

[1. GIỚI THIỆU 2](#_Toc85347985)

[1.1. Giới thiệu về Arduino 2](#_Toc85347986)

[1.2. Tổng quan về linh kiện và các phần mềm thiết kế 3](#_Toc85347987)

[1.2.1. Led 7 đoạn 3](#_Toc85347988)

[1.2.2. Kit Arduino Uno R3 5](#_Toc85347989)

[1.2.3. LM35 (Cảm biến nhiệt độ) 7](#_Toc85347990)

[1.2.4. Resistor 8](#_Toc85347991)

[1.2.5. App Blynk 9](#_Toc85347992)

[1.2.6. Giới thiệu về Arduino IDE và ngôn ngữ lập trình Arduino 10](#_Toc85347993)

[2. PHƯƠNG PHÁP NGHIÊN CỨU 11](#_Toc85347994)

[2.1. Mô phỏng nguyên lý hoạt động bằng phần mềm ISIS 11](#_Toc85347995)

[2.2. Thiết kế mạch in với sprint-layout 11](#_Toc85347996)

[2.3. Thiết kế mạch in với Ares 13](#_Toc85347997)

[2.4. Lập trình 15](#_Toc85347998)

[2.4.1. Khai báo thư viện, biến và hàm Setup 15](#_Toc85347999)

[2.4.2. Hàm ChuyenDoC 16](#_Toc85348000)

[2.4.3. Hàm Loop 16](#_Toc85348001)

[2.5. Lập trình App bằng Blynk 17](#_Toc85348002)

[2.5.1. Khai báo thư viện,kết nối wifi 17](#_Toc85348003)

[2.5.2. Hàm gửi dữ liệu từ DHT11 18](#_Toc85348004)

[2.5.3. Hàm Setup và Loop 19](#_Toc85348005)

[2.6. App C# và chuyển dữ liệu nhiệt độ lên Web 21](#_Toc85348006)

[2.6.1. Đối với mô phỏng ISIS Professtional 21](#_Toc85348007)

[2.6.2. Đối với phần Code Arduino 23](#_Toc85348008)

[2.6.3. Ứng dụng C# 24](#_Toc85348009)

[3. KẾT QUẢ 31](#_Toc85348010)

[4. PHÂN CÔNG NHIỆM VỤ VÀ ĐÓNG GÓP 32](#_Toc85348011)

[5. TÀI LIỆU THAM KHẢO 33](#_Toc85348012)

**DANH MỤC HÌNH ẢNH**

[Hình 1.1 Những thành viên khởi xướng Arduino 2](#_Toc85348087)

[Hình 1.2 Bo mạch Arduino Uno R3 3](#_Toc85348088)

[Hình 1.3 Led 7 đoạn 3](#_Toc85348089)

[Hình 1.4 Mô phỏng 2 loại Led 7 đoạn 4](#_Toc85348090)

[Hình 1.5 Mô phỏng Led 7 đoạn 4](#_Toc85348091)

[Hình 1.6 Cấu tạo của Arduino Uno R3 5](#_Toc85348092)

[Hình 1.7 Cảm biến nhiệt độ LM35 7](#_Toc85348093)

[Hình 1.8 Điện trở các loại 8](#_Toc85348094)

[Hình 1.9 Mô tả cấu tạo điện trở cacbon 8](#_Toc85348095)

[Hình 1.10 Giao diện phần mềm Blynk 9](#_Toc85348096)

[Hình 1.11 Giao diện Arduino IDE 10](#_Toc85348097)

[Hình 2.1 Mô phỏng mạch Arduino trên ISIS 11](#_Toc85348098)

[Hình 2.2 Vẽ mạch in trên Sprint-Layout 12](#_Toc85348099)

[Hình 2.3 Mạch in hoàn thiện hơn 12](file:///D:\HOC_TAP\LapTrinhNhung\De_Tai_Led_7Doan\Báo%20cáo\Báo%20cáo.docx#_Toc85348100)

[Hình 2.4 Mạch in hoàn chỉnh trước khi in 12](#_Toc85348101)

[Hình 2.5 Mô phỏng mạch dự kiến sau khi hoàn thành việc in 13](#_Toc85348102)

[Hình 2.6 Thiết kế mạch in trên ISIS (Ares) 13](#_Toc85348103)

[Hình 2.7 Mạch trước của bảng mạch 14](#_Toc85348104)

[Hình 2.8 Mặt sau của bảng mạch 14](#_Toc85348105)

[Hình 2.9 Code Khai báo thư viện, kết nối wifi 17](#_Toc85348106)

[Hình 2.10 Code hàm gửi dữ liệu từ DHT11 18](#_Toc85348107)

[Hình 2.11 Hàm Setup và Loop 19](#_Toc85348108)

[Hình 2.12 Các thông số cho nhiệt độ và độ ẩm trên Blynk 19](#_Toc85348109)

[Hình 2.13 Giá trị nhiệt độ, độ ẩm hiển thị trên biểu đồ 20](#_Toc85348110)

[Hình 2.14 Kết quả sau khi hoàn thành App 20](#_Toc85348111)

[Hình 2.15 Sô đồ chuyển dữ liệu 21](#_Toc85348112)

[Hình 2.16 COMPIM trong sơ đồ mạch 21](#_Toc85348113)

[Hình 2.17 Các thuộc tính của COMPIM 22](#_Toc85348114)

[Hình 2.18 Kết nối 2 cổng COM bằng Virtual Serial Port 22](#_Toc85348115)

[Hình 2.19 Khởi tạo giao tiếp nối tiếp 23](#_Toc85348116)

[Hình 2.20 Code bổ sung trong vòng lặp loop 23](#_Toc85348117)

[Hình 2.21 Code chuyển độ F 23](#_Toc85348118)

[Hình 2.22 Giao diện hiển thị nhiệt độ bằng C# 24](#_Toc85348119)

[Hình 2.23 Khai báo cổng Port 24](#_Toc85348120)

[Hình 2.24 Code khai báo biến 25](#_Toc85348121)

[Hình 2.25 Lấy danh sách cổng COM 25](#_Toc85348122)

[Hình 2.26 Code kết nối tới cổng COM 25](#_Toc85348123)

[Hình 2.27 Code ngắt kết nối cổng COM 25](#_Toc85348124)

[Hình 2.28 Code mở led 7 đoạn 26](#_Toc85348125)

[Hình 2.29 Code chuyển đổi nhiệt độ 27](#_Toc85348126)

[Hình 2.30 Code lưu file Text 27](#_Toc85348127)

[Hình 2.31 Code nút Clear 27](#_Toc85348128)

[Hình 2.32 Kết quả hiển thị nhiệt độ trên Web 28](#_Toc85348129)

[Hình 2.33 Code đọc file text 28](#_Toc85348130)

[Hình 2.34 Code thiết kế web 29](#_Toc85348131)

[Hình 2.35 Code cập nhật trang web 30](#_Toc85348132)

[Hình 2.36 Nội dung file server.php 30](#_Toc85348133)

[Hình 3.1 Kết quả hiển thị nhiệt độ trên mạch mô phỏng và Web 31](#_Toc85348134)

[Hình 3.2 Kết quả hiển thị nhiệt độ trên App 31](#_Toc85348135)

**DANH MỤC BẢNG BIỂU**

[Bảng 1.1 Bảng chi tiết kỹ thuật trong Arduino 6](#_Toc85280701)

[Bảng 1.2 Bảng thông số kỹ thuật của cảm biến nhiệt độ LM35 7](#_Toc85280702)

# 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 thểnói 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.

Trên cơ sở kiến thức đã học trong môn học: Tin học đại cương và một số môn học khác...cùng với những hiểu biết về các thiết bị điện tử, chúng em đã quyết định thực hiện đề tài: Thiết kế mạch đo nhiệt độ sử dụng board Arduino, hiển thị trên 4 led 7 thanh và truyền phát không dây sử dụng module R3 với mục đích để tìm hiểu thêm về Arduino, làm quen với các thiết bị điện tử và nâng cao hiểu biết cho bản thân. Do kiến thức còn hạn hẹp, thêm vào đó đây là lần đầu chúng em thực hiện đồ án nên chắc chắn không tránh khỏi những thiếu sót, hạn chế vì thế chúng em rất mong có được sự góp ý và nhắc nhờ từ thầy giáo để có thể hoàn thiện đề tài của mình.

Chúng em xin chân thành cảm ơn giảng viên **ThS. Đoàn Vũ Thịnh** đã giúp đỡ chúng em rất nhiều trong quá trình tìm hiểu ,thiết kế và hoàn thành đề tài đồ án này.

*Đắk Lắk, tháng 10 năm 2021*

***Ký tên nhóm trưởng***

**Nguyễn Văn Huy**

# GIỚI THIỆU

## Giới thiệu về Arduino

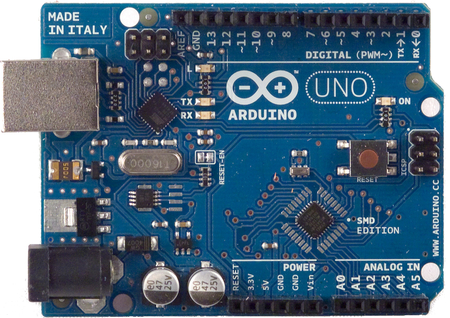
Arduino thực sự đã gây sóng gió trên thị trường người dùng DIY (là những người tự chế ra sản phẩm của mình) trên toàn thế giới trong vài năm gần đây, gần giống với những gì Apple đã làm được trên thị trường thiết bị di động. Số lượng người dùng cực lớn và đa dạng với trình độ trải rộng từ bậc phổ thông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



Hình 1.1 Những thành viên khởi xướng Arduino

Arduino thật 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 120.000 VNĐ, người dùng đã có thể sở hữu một board Arduino uno R3 có 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 Arduin. Arduino chính thức được đưa ra giới thiệuvào 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 Instistute 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

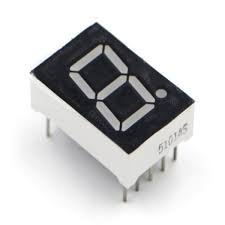


Hình 1.2 Bo mạch Arduino Uno R3

## Tổng quan về linh kiện và các phần mềm thiết kế

### Led 7 đoạn

Giới thiệu: Led 7 đoạn là thiết bị hiển thị điện tử để hiển thị số. Khi mỗi đoạn chiếu sáng thì một phần của chữ số sẽ được hiển thị. Led 7 đoạn được sủ dụng rộng rãi trong đồng hồ số,máy tính.



Hình 1.3 Led 7 đoạn

**Cấu tạo:**

LED 7 đoạn bao gồm 8 LED được kết nối song song để có thể thắp sáng hiển thị số  “0, 1, 2, 3, 4, 5,

7, 8, 9, A, b, C, d, E, F, …”.

Mỗi đoạn Led được đánh dấu từ A tới G.

Đoạn thứ tám gọi là “chấm thập phân” (Decimal Point) ký hiệu DP được sử dụng khi hiển thị số không

phải là số nguyên

Phân loại LED 7 đoạn:

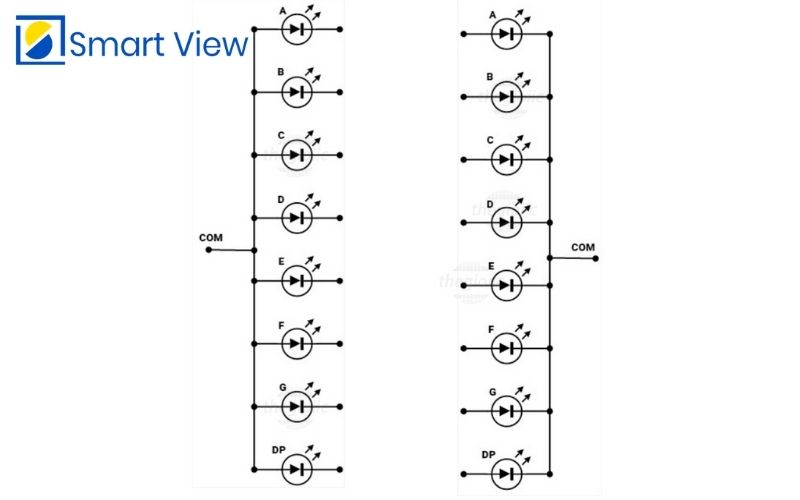
Dựa vào các cực được nối, có thể phân loại LED 7 đoạn như sau:

Loại dương chung (Common Anode): nếu cực dương (anode) của tất cả 8 LED được nối với nhau và

các cực âm (cathode) đứng riêng lẻ.

Loại âm chung (Common Cathode): nếu cực âm (cathode) của tất cả 8 LED được nối với nhau và các

cực dương (anode) đứng riêng lẻ.



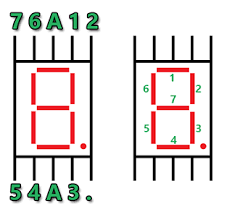
Hình 1.4 Mô phỏng 2 loại Led 7 đoạn

**Chân ngõ ra**

LED 7 đoạn có 10 chân, trong đó 8 chân được nối với LED (1, 2, 3, 4, 5, 6, 7, và A).

Tùy vào loại LED 7 đoạn, hai chân giữa được đánh dấu COM hoặc dương chung hoặc âm

chung của các LED.

****

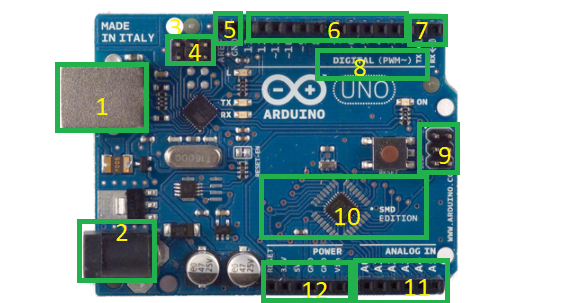
Hình 1.5 Mô phỏng Led 7 đoạn

### Kit Arduino Uno R3

**Arduino UNO R3** là một loại [**bo mạch**](https://maysanxuattudong.com/vi-mach-dien-tu-la-gi/) vi điều khiển, được sử dụng phổ biến trong họ [**Arduino**](https://maysanxuattudong.com/vi-mach-dien-tu-la-gi/). Chúng được phát hành vào năm 2011, và là phiên bản thứ 3 mới nhất của bảng Arduino. Mạch kit này được phát triển dựa trên ATmega328P với mục đích kiểm soát và giữ chức năng là bộ vi điều khiển.

 Phiên bản hiện tại của Arduino Uno R3 đi kèm với giao diện USB, 6 chân đầu vào analog, 14 cổng kỹ thuật số **I / O** được sử dụng để kết nối với các mạch điện tử, thiết bị bên ngoài. Trong đó có 14 cổng I / O, 6 chân đầu ra xung [***PWM***](https://www.vietnic.vn/ic-dieu-xung-pwm) cho phép các nhà thiết kế kiểm soát và điều khiển các thiết bị mạch điện tử ngoại vi một cách trực quan.

Arduino Uno R3 được kết nối trực tiếp với máy tính thông qua USB để giao tiếp với phần mềm lập trình IDE, tương thích với Windows, MAC hoặc Linux Systems, tuy nhiên, Windows thích hợp hơn để sử dụng. Các ngôn ngữ lập trình như C và C ++ được sử dụng trong IDE.



Hình 1.6 Cấu tạo của Arduino Uno R3

1. **USB:** Cổng giao tiếp USB có 2 chức năng: cấp nguồn cho board mạch và truyền thông nối tiếp với máy tính trong việc nạp chương trình hay giao tiếp nối tiếp.
2. **Jack Power:** Nguồn cấp cho board mạch Arduino. Có 2 loại nguồn có thể sử dụng được là nguồn xoay chiều (AC) tối đa 6V và nguồn một chiều (DC) tối đa 5V.
3. **RESET:** Đặt lại trạng thái ngay khi nạp chương trình
4. **ISCP:** Chân giao tiếp với USB, tín hiệu giao tiếp có thể giám sát từ đây.
5. **AREF và GND**: GND hay chân Mass hay chân đất dùng để cấp nguồn 0V cho các modue khác có kết nối với Arduino. Trong khi chân AREF được dùng để phối hợp với các chân Analog Input (11) để điều chỉnh dải điện áp đầu vào cho ADC 10 bit.
6. **DIGITAL Input/Output**: 12 chân tín hiệu số được dùng làm đầu vào hoặc đầu ra tại mỗi thời điểm.
7. **Serial Transmition:** 2 chân RXD và TXD được dùng trong truyền thông nối tiếp. Khi 2 chân này cũng có thể được cấu hình làm đầu vào hoặc đầu ra tín hiệu số.
8. **PWM:** Chân băm xung có ký hiệu là ~ dùng để tạo ra chuỗi xung vuông trong điều khiển tốc độ động cơ hay hiển thị LED sáng dần hay tối dần.
9. **ICSP:** Giao tiếp với Atmega328.
10. **IC lập trình ATMEGA16u2**
11. **ANALOG Input:** Tín hiệu chuyển đổi **ADC** được đưa vào các chân này. Ngoài ra, các chân này cũng có thể sử dụng để làm đầu vào/ra tín hiệu số.
12. **POWER:** Cung cấp nguồn 5V, **GND** cho các thành phần mở rộng.

|  |  |
| --- | --- |
| 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, trong đó 0.5KB bootloader |
| SRAM | 2 KB (ATmega328) |
| EEPROM | 1KB (ATmega328) |

Bảng 1.1 Bảng chi tiết kỹ thuật trong Arduino

### LM35 (Cảm biến nhiệt độ)

LM35 là một cảm biến nhiệt độ được sử dụng rộng rãi. Nó hiển thị các giá trị dưới dạng điện áp đầu ra thay vì độ C.

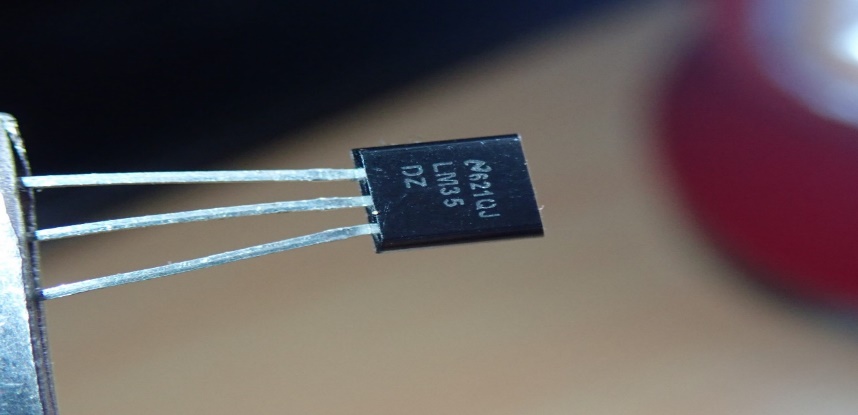
LM35 hiển thị giá trị điện áp cao hơn cặp nhiệt điện và có thể không cần khuếch đại điện áp đầu ra.

Điện áp đầu ra của LM35 tỷ lệ với nhiệt độ C. Hệ số thang đo là 0,01 V / ° C.

Một đặc điểm quan trọng nhất là nó chỉ lấy 60 micromps từ nguồn và có khả năng tự gia nhiệt thấp.

Cảm biến nhiệt độ LM35 có nhiều gói khác nhau như gói giống transistor kim loại T0-46, gói giống transistor nhựa TO-92, gói dán 8 chân SO-8.

Khi sử dụng, cảm biến LM35 không dùng để điều chỉnh nhiệt độ cho từng thiết bị. trong thực tế thì độ chính xác của cảm biến sẽ khác nhau khi ở các môi trường làm việc khác nhau. Cụ thể là: ở nhiệt độ phòng độ chính xác là **¼ độ C**, nhiệt độ ngoài khoảng từ 2 đến 150 độ C thì độ chính xác là **¾ độ C.**



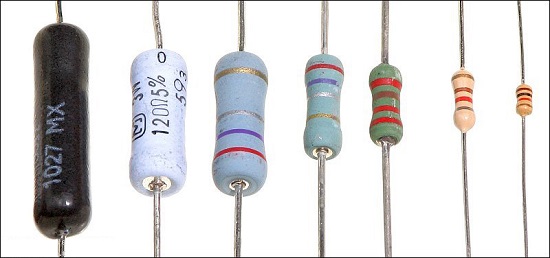
Hình 1.7 Cảm biến nhiệt độ LM35

|  |  |
| --- | --- |
| **Điện áp hoạt động:** | 4~20VDC |
| **Công suất tiêu thụ:** | khoảng 60uA |
| **Khoảng đo:** | -55°C đến 150°C |
| **Điện áp tuyến tính theo nhiệt độ:** | 10mV/°C |
| **Kiểu chân: 4.3 × 4.3mm** | 4.3 × 4.3mm |
| **Kích thước:** | 4.3 × 4.3mm |
| **Sai số:** | 0.25°C |

Bảng 1.2 Bảng thông số kỹ thuật của cảm biến nhiệt độ LM35

### Resistor

**Điện trở** hay **Resistor** là một linh kiện điện tử thụ động gồm 2 tiếp điểm kết nối, thường được dùng để hạn chế cường độ dòng điện chảy trong mạch, điều chỉnh mức độ tín hiệu, dùng để chia điện áp, kích hoạt các linh kiện điện tử chủ động như transistor, tiếp điểm cuối trong đường truyền điện và có trong rất nhiều ứng dụng khác. Điện trở có mặt ở mọi nơi trong thiết bị điện tử và như vậy điện trở là linh kiện quan trọng không thể thiếu được, trong mạch điện, điện trở.



Hình 1.8 Điện trở các loại

Điện trở công suất có thể tiêu tán một lượng lớn điện năng chuyển sang nhiệt năng có trong các bộ điều khiển động cơ, trong các hệ thống phân phối điện.

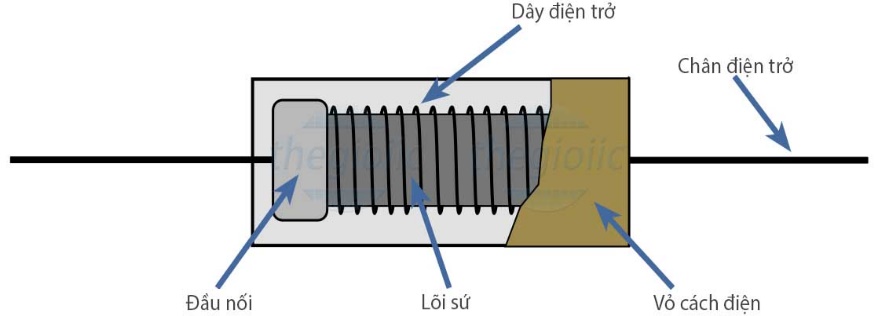
Nói tóm tắt, điện trở là sự cản trở dòng điện của một vật dẫn điện, nếu một vật dẫn điện tốt thì điện trở nhỏ, vật dẫn điện kém thì điện trở lớn, vật cách điện thì điện trở là vô cùng lớn.

Điện trở thường được tạo ra từ kết tủa kim loại tinh khiết (như niken) hoặc màng oxit (như oxit thiếc) vào trụ gốm cách điện hoặc chất nền.

Đơn vị điện trở là Ω (Ohm), mΩ (milliohm), KΩ (kilohm), MΩ (megohm)

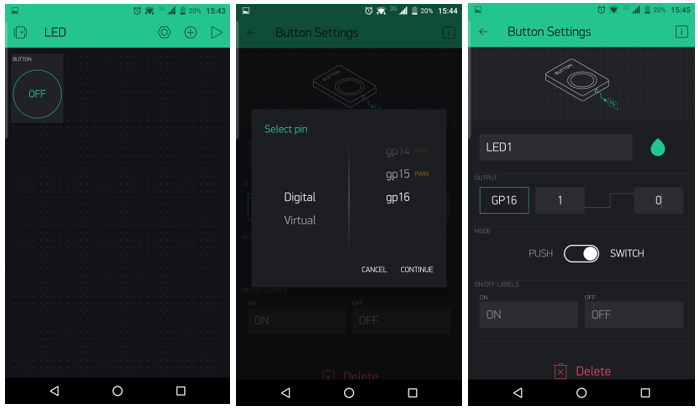
1 mΩ = 0.001 Ω, 1KΩ = 1000 Ω, 1MΩ = 1000 K Ω = 1000.000 Ω

Các điện trở mắc nối tiếp có giá trị tương đương có công thức là: ***Rtd = R1 + R2 + R3*** và nếu chỉ có hai điện trở song song thì tính bằng công thức: ***Rtd = R1\* R2 / (R1 + R2*).** Trong mọi thời điểm, Công suất P(W) tiêu thụ bởi một điện trở có trở kháng R(Ω) được tính bởi công thức: P= U\*I =I2\*R= U2/R. Trong đó U (V) là điện áp trên điện trở và I (A) là dòng điện đi qua nó.



Hình 1.9 Mô tả cấu tạo điện trở cacbon

### App Blynk



Hình 1.10 Giao diện phần mềm Blynk

Blynk là một ứng dụng chạy trên nền tảng iOS và Android để điều khiển và giám sát thiết bị thông qua internet. Blynk không bị ràng buộc với những phần cứng cụ thể nào cả, thay vào đó, nó hỗ trợ phần cứng cho bạn lựa chọn như Arduino, Raspberry Pi, ESP8266 và nhiều module phần cứng phổ biến khác.

Thông tin về Blynk:

* Nhà phát hành: Pavel Bayborodin
* Năm phát hành: 2014
* Nền tảng: Andriod
* Ngôn ngữ viết: Java
* Phiên bản hiện hành: 2.27.21
* Cấu trúc của Blynk:

Có ba thành phần chính trong nền tảng:

* **Blynk App** - cho phép tạo giao diện cho sản phẩm của bạn bằng cách kéo thả các widget khác nhau mà nhà cung cấp đã thiết kế sẵn.
* **Blynk Server** - chịu trách nhiệm xử lý dữ liệu trung tâm giữa điện thoại, máy tính bảng và phần cứng. Bạn có thể sử dụng Blynk Cloud của Blynk cung cấp hoặc tự tạo máy chủ Blynk riêng của bạn. Vì đây là mã nguồn mở, nên bạn có thể dễ dàng intergrate vào các thiết bị và thậm chí có thể sử dụng PC làm server của bạn.
* **Library Blynk** – support cho hầu hết tất cả các nền tảng phần cứng phổ biến - cho phép giao tiếp với máy chủ và xử lý tất cả các lệnh đến và đi.

Tính năng của Blynk:

* Cung cấp API & giao diện người dùng tương tự cho tất cả các thiết bị và phần cứng được hỗ trợ.
* Kết nối với server qua nhiều kết nối: Wifi, Bluetooth, Ethernet, USB, GSM…
* Các tiện ích trên giao diện được nhà cung cấp dễ sử dụng.
* Thao tác kéo thả trực tiếp giao diện mà không cần viết mã.
* Dễ dàng tích hợp và thêm chức năng mới bằng cách sử dụng các cổng kết nối ảo được tích hợp trên blynk app.

### Giới thiệu về Arduino IDE và ngôn ngữ lập trình Arduino

Arduino IDE (Arduino Integrated Development Environment) là một chương trình phần mềm mã nguồn mở cho phép người dùng viết và tải lên mã lên vi điều khiển.



Hình 1.11 Giao diện Arduino IDE

Ứng dụng này hoạt động được cả 3 môi trường Windows, Linux và MacOS, mã nguồn của Arduino IDE được viết bởi C/C++ và tương thích hầu hết các board Arduino. Chương trình viết trên IDE có thể là C hoặc C++, sau khi biên dịch được nạp trực tiếp lên board mạch lập trình thông qua cổng USB.

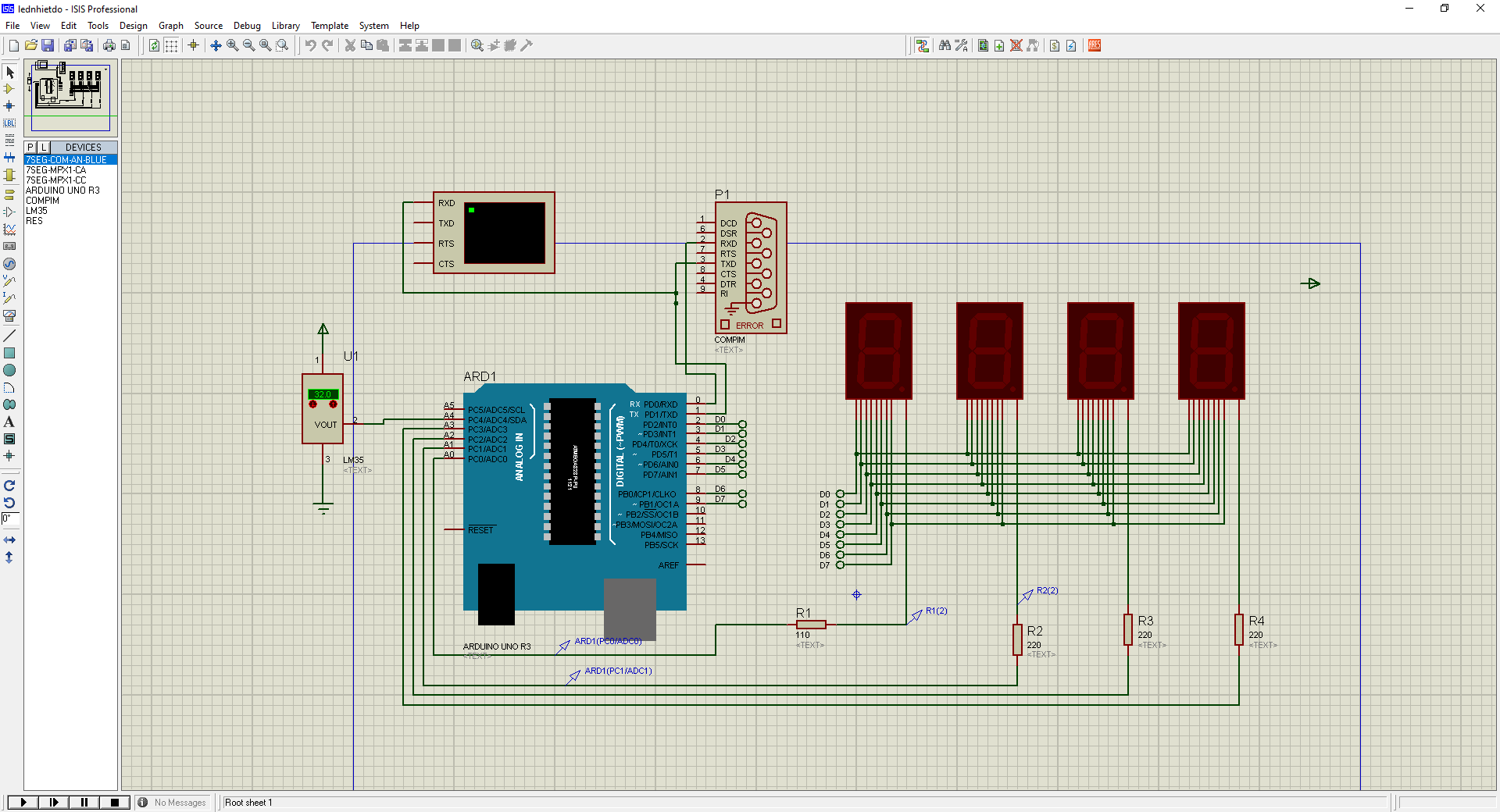
Ngôn ngữ lập trình có thể được mở rộng thông qua các thư viện C++. Và do ngôn ngữ lập trình này dựa trên nền tảng ngôn ngữ C của AVR nên người dùng hoàn toàn có thể nhúng thêm code viết bằng AVR vào chương trình nếu muốn. Hiện tại, Arduino IDE có thể download từ trang chủ <https://www.arduino.cc/en/software>

# PHƯƠNG PHÁP NGHIÊN CỨU

## Mô phỏng nguyên lý hoạt động bằng phần mềm ISIS

Các thiết bị mô phỏng bao gồm:

* Arduino R3
* 4 đèn 7 Leg: Hiển thị nhiệt độ
* LM35: đo nhiệt độ phòng
* COMPIM: trao đổi dữ liệu với máy tính thông qua cổng COM (sử dụng cổng COM2)
* Terminal: Hiển thị dữ liệu nhiệt độ



Hình 2.1 Mô phỏng mạch Arduino trên ISIS

Nối 2 đầu dây RXD và TXD vào Arduino R3 để trao đổi dữ liệu.

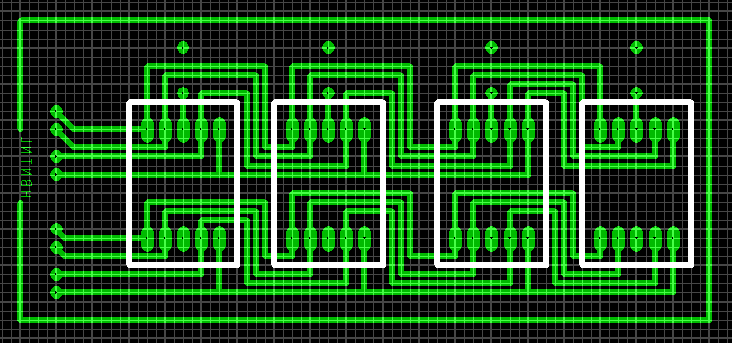
Các LED được nối với nguồn A0,A1,A2,A3 qua điện trở

Các LED được nối với nhau qua cổ D0,D1,D2,D3,D4,D5,D6,D7 để hiển thị dữ liệu nhiệt độ

LM35 được nối đất và nguồn, và nối vào cổng A4 của Arduino để đo nhiệt độ.

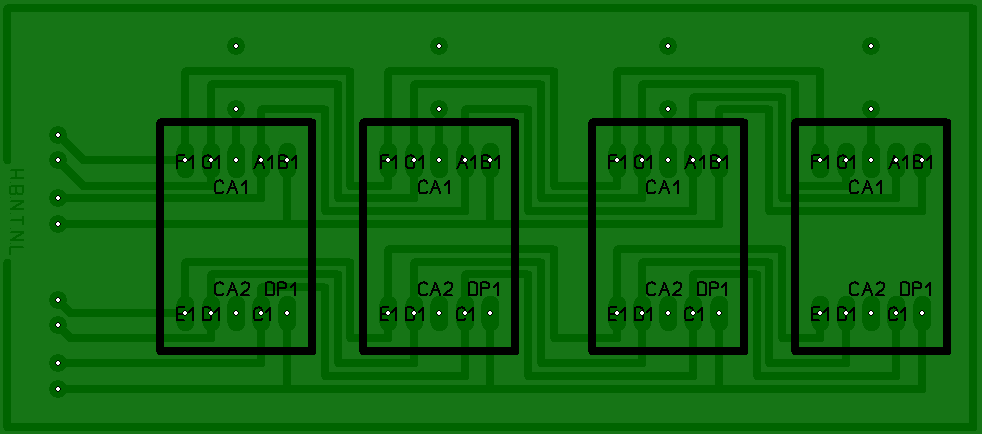
## Thiết kế mạch in với sprint-layout

Mô tả: Mạch in sẽ bao gồm các chân để nối 4 con led 7 đoạn và 4 con trở tương ứng, cùng với việc đi dây giữa các chân. (được thiết kế trên công cụ sprint-layout 5.0)



Hình 2.2 Vẽ mạch in trên Sprint-Layout

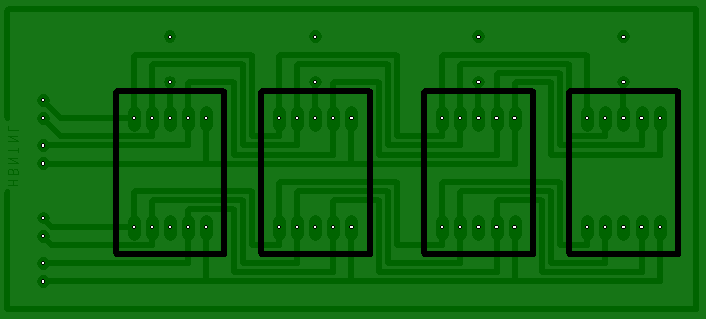
Đi dây trên bảng mạch theo thứ tự:

* B1-B1-B1-B1

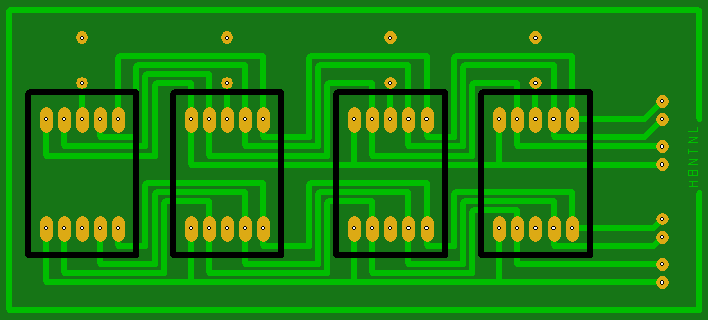
Hình 2.3 Mạch in hoàn thiện hơn

* A1-A1-A1-A1
* G1-G1-G1-G1
* F1-F1-F1-F1
* C1-C1-C1-C1
* D1-D1-D1-D1
* E1-E1-E1-E1
* DP1-DP1-DP1-DP1
* Các chân CA1 sẽ được nối với 4 điện trở.

(Các dây khi đi sẽ ko trùng lên nhau trên mạch)



Hình 2.4 Mạch in hoàn chỉnh trước khi in



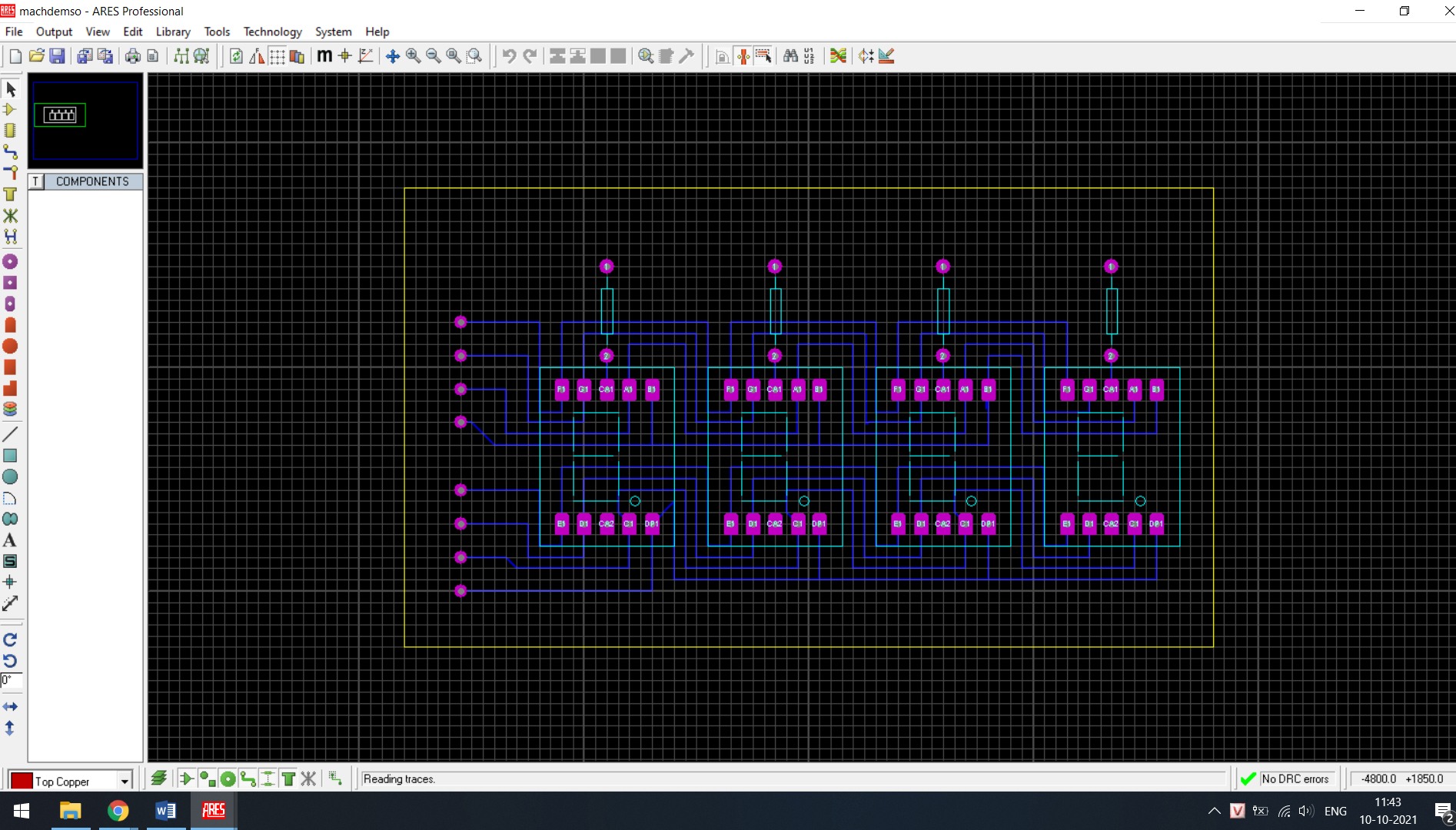
Hình 2.5 Mô phỏng mạch dự kiến sau khi hoàn thành việc in

## Thiết kế mạch in với Ares

ARES là ứng dụng cho phép bạn xem và in các bản thiết kế PCB được tạo bằng chương trình ARES

Professional được tích hợp trong chương trình Proteus, có định dạng LAY.

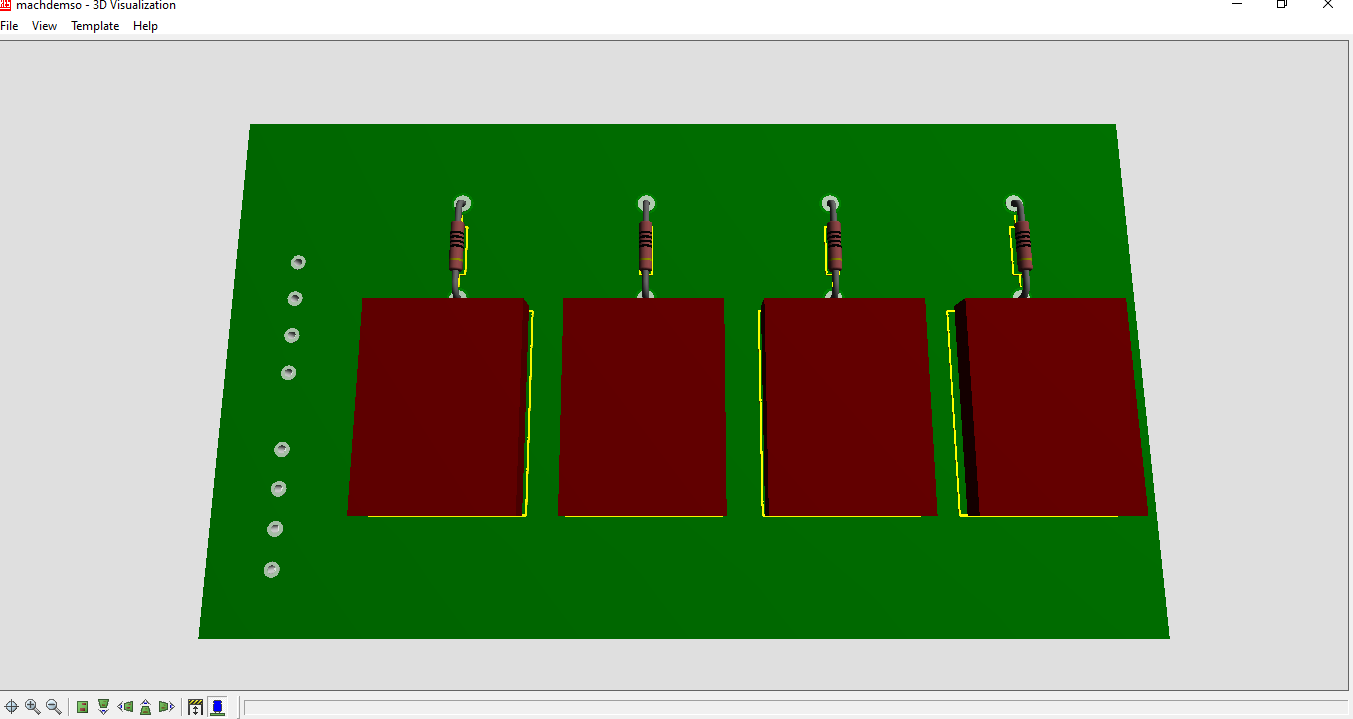
Bên dưới là bản thiết kế hoàn chỉnh



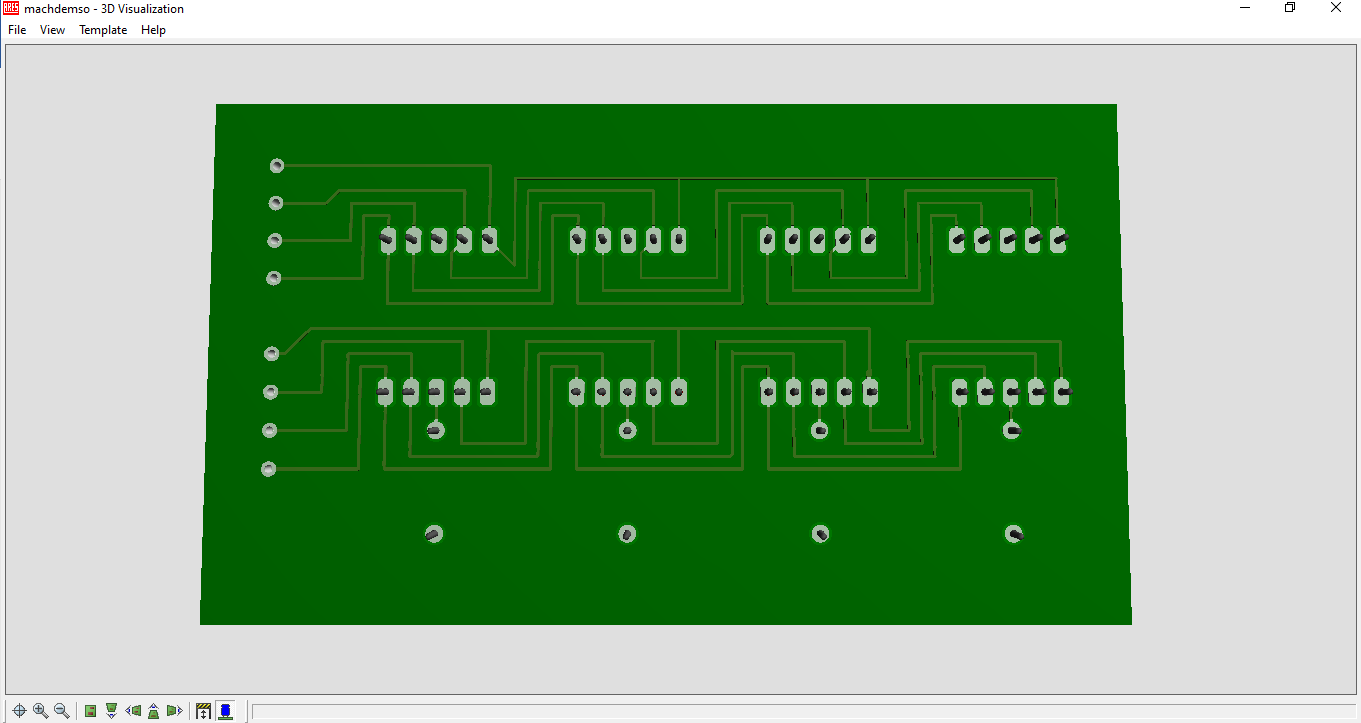
Hình 2.6 Thiết kế mạch in trên ISIS (Ares)

Mô tả: Mạch in sẽ bao gồm các chân để nối 4 con led 7 đoạn và 4 con trở tương ứng, cùng với việc đi

dây giữa các chân.



Hình 2.7 Mạch trước của bảng mạch



Hình 2.8 Mặt sau của bảng mạch

## Lập trình

### Khai báo thư viện, biến và hàm Setup

#include<SevSeg.h>

SevSeg sevSeg;

unsigned int myTemp;

void setup() {

  Serial.begin(9600);

  Serial.println("==================");

  Serial.println("Welcome to Nhom 3");

  Serial.println("==================");

  byte numDigits = 4;

  byte ccDPins[] = {A0, A1, A2, A3};

  byte segDPins[] = {2,3,4,5,6,7,8,9};

  //(hardwareConfig,numDigitsIn,digitPinsIn[], segmentPinsIn[],

  //resOnSegmentsIn,updateWithDelaysIn,leadingZerosIn, disableDecPoint)

  sevSeg.begin(COMMON\_ANODE, numDigits, ccDPins, segDPins, false, false, false, false);

  // Thiết lập độ sáng

  sevSeg.setBrightness(100);

}

Khai báo thư viện SevSeg để thao tác với Led 7 đoạn.

Khai báo biến sevSeg để sử dụng các hàm trong thư viện SevSeg, biến myTemp để lưu trữ nhiệt độ

hiển thị lên Led 7 đoạn.

Hàm Setup thiết lập một số thiết bị như: Serial.begin() thiết lập giao tiếp nối tiếp giữa bo Arduino và

một thiết bị khác(như là máy tính), sevSeg.begin() thiết lập Led 7 đoạn có những thuộc tính nào.

### Hàm ChuyenDoC

unsigned int ChuyenDoC()

{

  float temp = (float) 100 \* (5.0 / 1024) \* analogRead(A4);

  Serial.print("Nhiet do trong phong la: ");

  // In temp (nhiệt độ) lên màn hình và lấy 2 số thập phân

  Serial.print(temp, 2);

  Serial.print(char(223));

  Serial.println("C");

  temp = 100 \* temp;

  return ((unsigned int)temp);

}

Hàm ChuyenDoC lấy điện thế từ LM35 thông qua chân A4 của Arduino, sau đó tính toán sao cho ra

giá trị nhiệt độ và trả về giá trị nhiệt độ đó. Ngoài ra, hàm này còn hiển thị nhiệt độ lên Serial.

### Hàm Loop

void loop() {

  // Lấy thời gian kể từ khi arduino được khởi động

  unsigned long prMillis = millis();

  while( millis() - prMillis < 2000)

  {

    // làm mới màn hình

    sevSeg.refreshDisplay();

  }

  myTemp = ChuyenDoC();

  // Thiết lập số hiển thị và lấy 2 số thập phân

  sevSeg.setNumber(myTemp, 2, LOW);

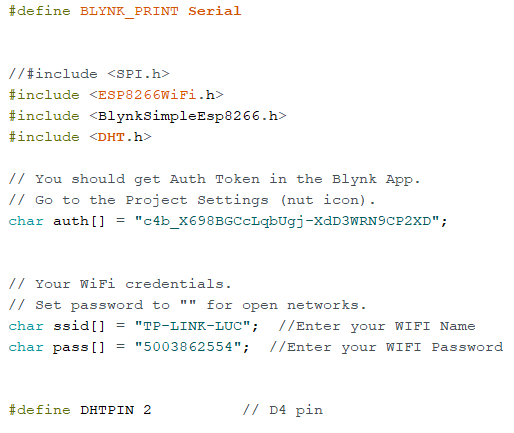
}

Hàm loop dùng để hiển thị nhiệt độ đoạn được lấy từ hàm ChuyenDoC lên Led 7 đoạn. Cứ mỗi sau 2

giây Led 7 đoạn sẽ được làm mới.

## Lập trình App bằng Blynk

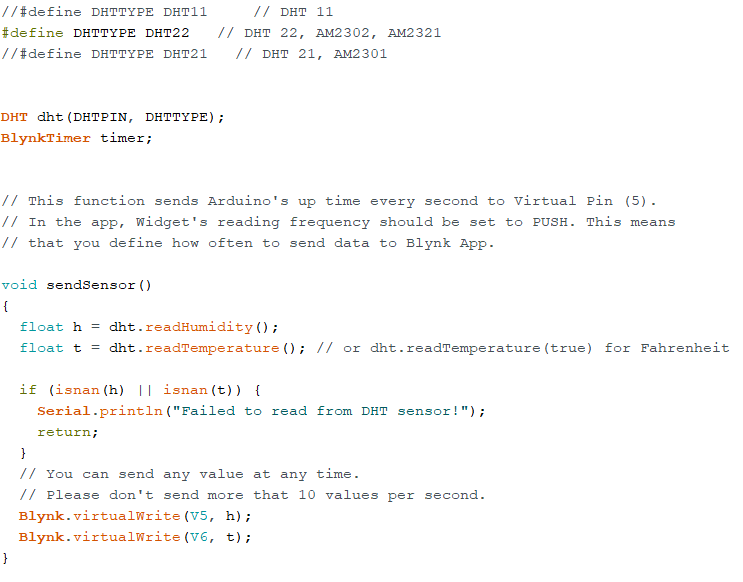
### Khai báo thư viện,kết nối wifi



Hình 2.9 Code Khai báo thư viện, kết nối wifi

Khai báo thư viện nodeMCU, thư viện Blynk và thư viện cho DHT, mã Auth để xác thực kết nối từ App Blynk đến Arduino, password và wifi kết nối NodeMCU đến Arduino và Blynk.

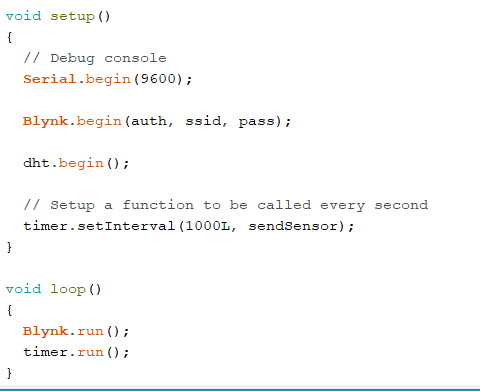
### Hàm gửi dữ liệu từ DHT11



Hình 2.10 Code hàm gửi dữ liệu từ DHT11

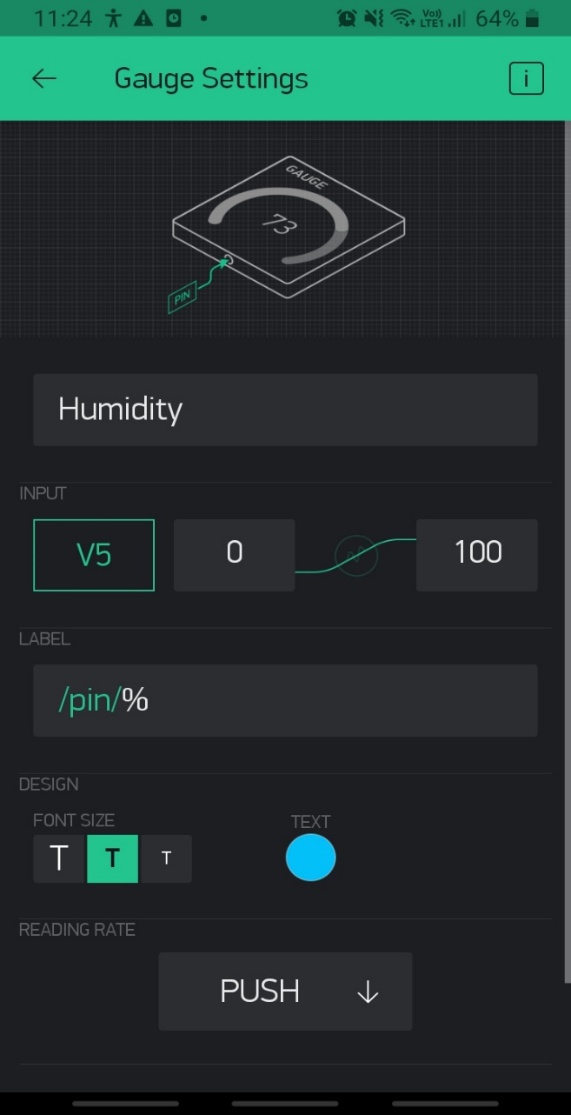
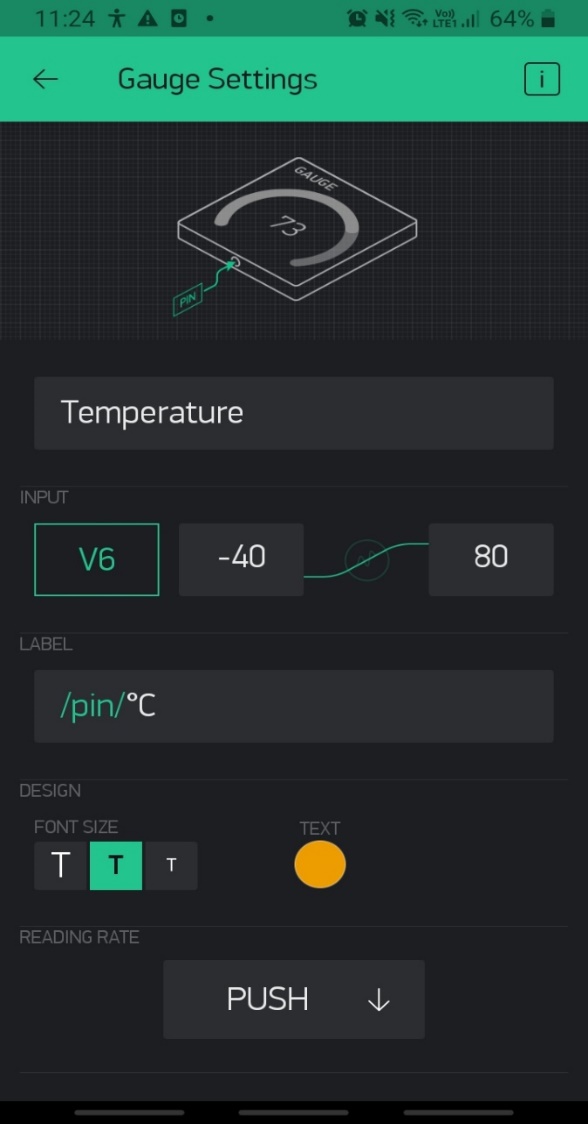
Khai báo hàm gửi dữ liệu nếu dữ liệu là False thì hiển thị Failed to read from DHT sensor còn nếu True thì xuất ra giá trị nhiệt độ và độ ẩm trên App Blynk các chân virtualWrite tương ứng với các chân kết nối trong Blynk.

### Hàm Setup và Loop

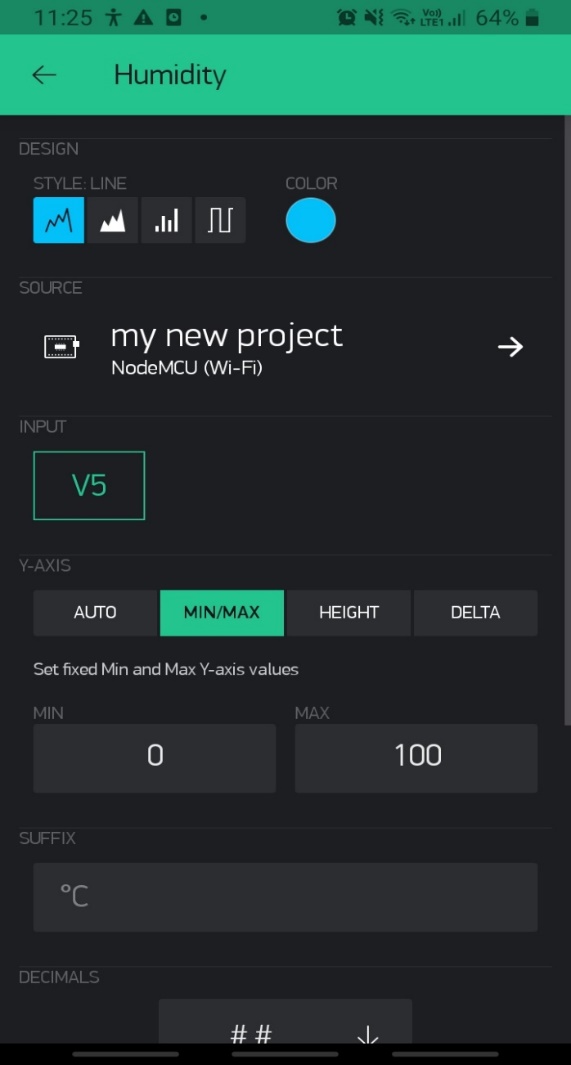
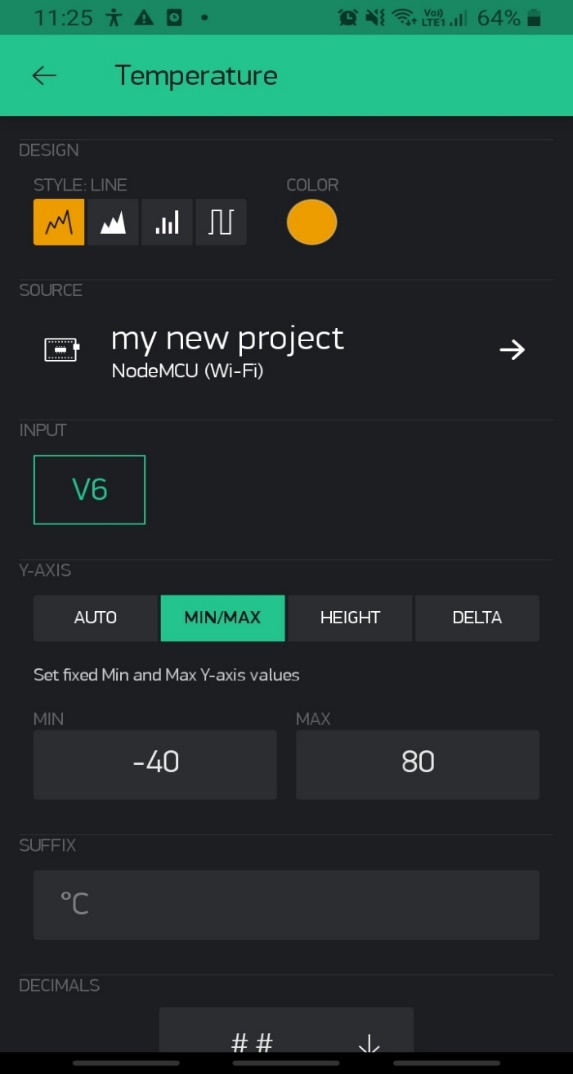


Hình 2.11 Hàm Setup và Loop

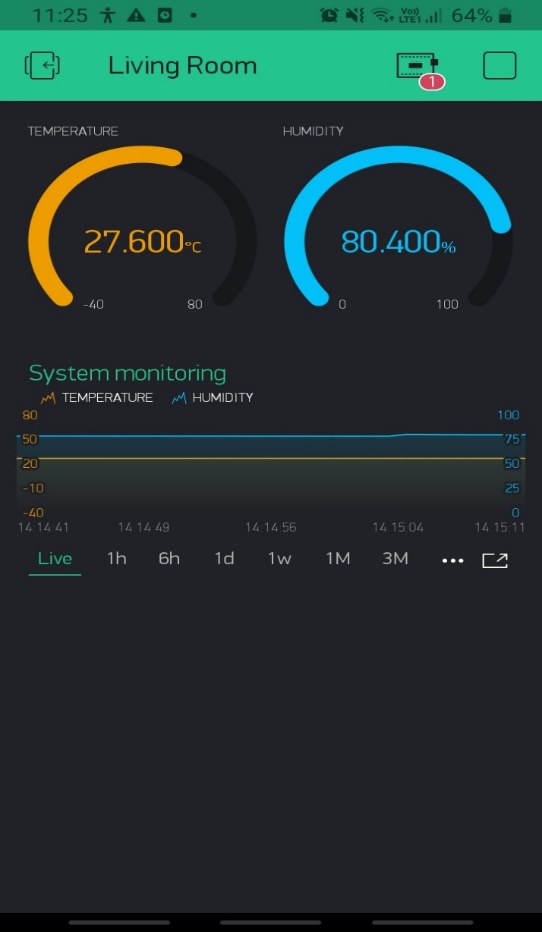
Khởi tạo giá trị cho cổng kết nối , mỗi 1s thì thì gửi gias trị một lần , hàm loop khởi động giá trị trên App và thời gian chạy.



Hình 2.12 Các thông số cho nhiệt độ và độ ẩm trên Blynk



Hình 2.13 Giá trị nhiệt độ, độ ẩm hiển thị trên biểu đồ



Hình 2.14 Kết quả sau khi hoàn thành App

## App C# và chuyển dữ liệu nhiệt độ lên Web

Phiên bản Isis personal 7 không hỗ trợ mô phỏng thiết bị Card để đọc ghi dữ liệu ra file Text

nên nhóm em sử dụng phương pháp là trao đổi dữ liệu ứng dụng với C#

**Việc truyền dữ liệu nhiệt độ trên Web được thực hiện theo phương pháp:**

Dữ liệu từ Arduino (Sử dụng isis mô phỏng)

-> trao đổi dữ liệu nhiệt độ với ứng dụng viết bằng C#

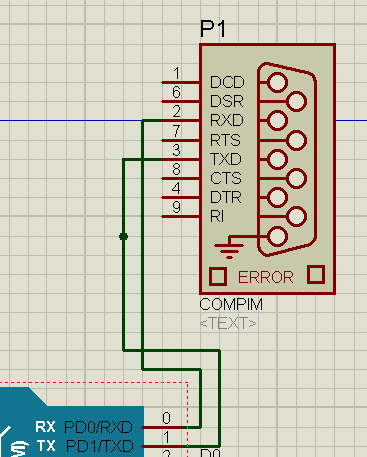
-> Đưa nhiệt độ từ C# lên Web lập trình bằng php

Hình 2.15 Sô đồ chuyển dữ liệu

### Đối với mô phỏng ISIS Professtional

Trong mô phỏng isis bổ sung thêm một thiết bị dùng để kết nối và trao đổi dữ liệu với máy tính.

Đó là COMPIM. Một thiết bị được kết nối thông qua cổng COM, thực hiện nối 2 đầu dây RXD và TXD vào Arduino R3 để trao đổi dữ liệu.

****

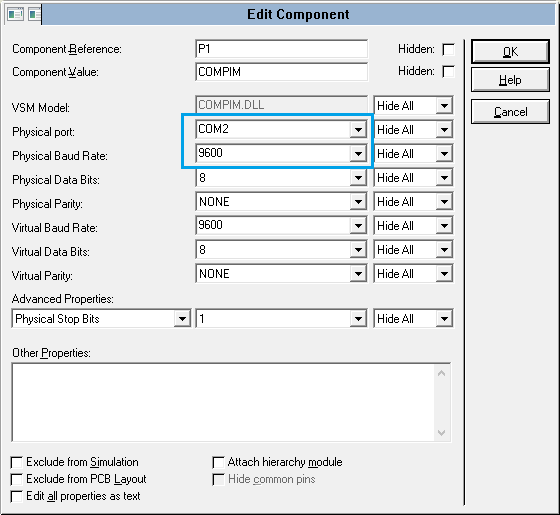
Hình 2.16 COMPIM trong sơ đồ mạch

**Thông tin tiết lập COMPIM**

Cổng COM sử dụng: COM2

Tốc độ Baud: 9600 (tốc độ truyền dữ liệu 9600 bit mỗi giây) đây là tốc độ phù hợp và được thiết lập

đa số.



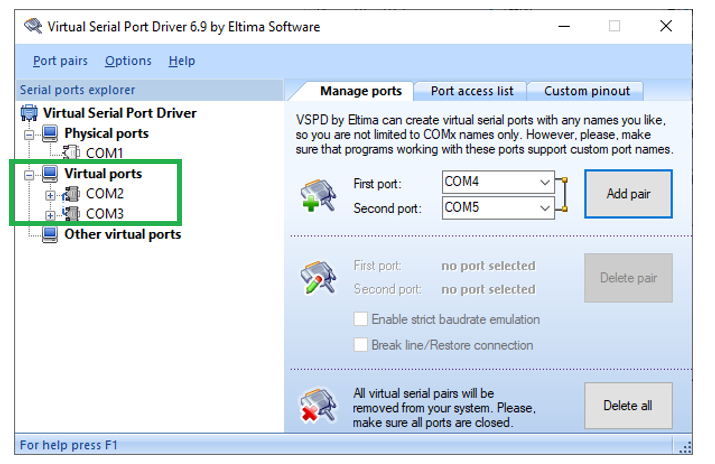
Hình 2.17 Các thuộc tính của COMPIM

Sử dụng phần mềm **Virual Serial Port** để kết nối 2 cổng COM với nhau.

Trong đó COMPIM đã được kết nối với cổng COM2

Ứng dụng C# sẽ được kết nối với cổng COM3

Từ đó trao đổi dữ liệu được với nhau

****

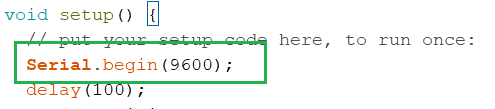
Hình 2.18 Kết nối 2 cổng COM bằng Virtual Serial Port

### Đối với phần Code Arduino

Sử dụng chuẩn giao tiếp truyền thông nối tiếp UART. Hay còn gọi là Serial.

Khai báo thêm Serial.begin(9600);

Trong đó 9600 là tốc độ baud



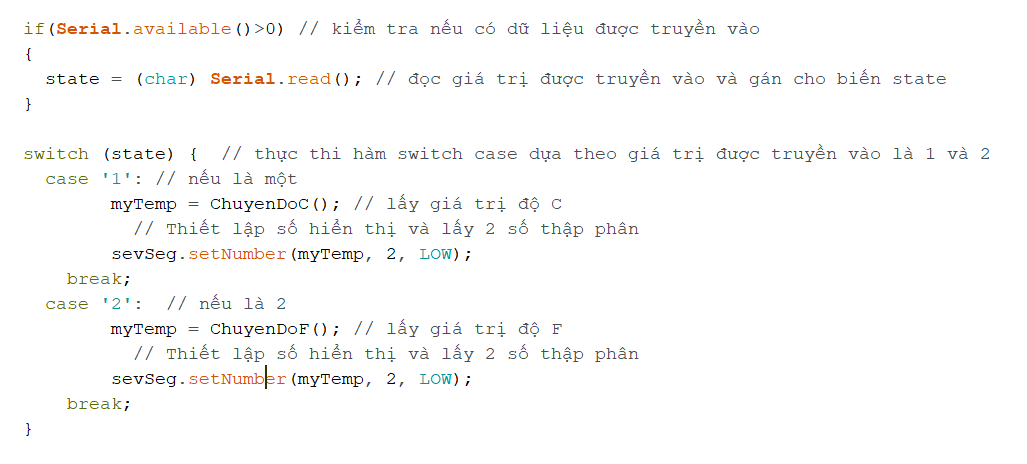
Hình 2.19 Khởi tạo giao tiếp nối tiếp

Sau đó kiểm tra Serial nếu có dữ liệu được truyền vào thì thực hiện đọc dữ liệu đó và thực hiện hiển thị theo nhiệt độ được yêu cầu tuỳ theo giá trị nhận được là 1 hoặc 2.

1: Đo độ C

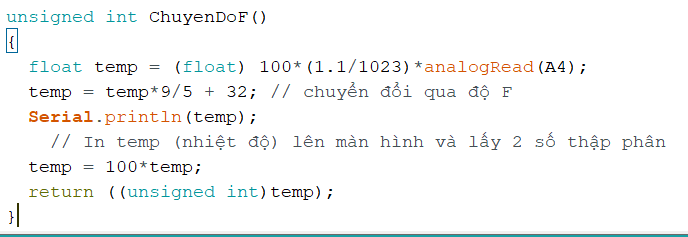
2: Đo độ F

Đây là chương trình được bổ sung trong vòng lặp loop()



Hình 2.20 Code bổ sung trong vòng lặp loop

Bổ sung thêm chương trình tính độ F, để chuyển đổi hiển thị trên 7 Leg

****

Hình 2.21 Code chuyển độ F

Thiết lập bên isis và code Arduino đã xong, giờ chuyển sang phần C#

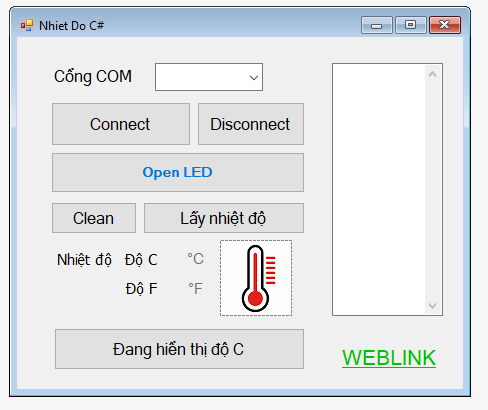
### Ứng dụng C#

Phần mềm sử dụng: Visual Studio 2017

**Giao diện:**

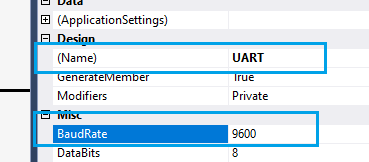
Bao gồm:

* Danh sách cổng COM,
* 2 nút kết nối và ngắt kết nối cổng COM,
* Open Led: Bật Led trên isis
* Clean: dọn dữ liệu hiển thị ở box bên phải
* Lấy nhiệt độ đo được hiện tại ở box bên phải
* Độ C và độ F -> dữ liệu được đưa lên Web qua file text
* Box bên phải -> dữ liệu được đưa lên Web qua file text
* Đang hiển thị độ C: nhấn vào sẽ chuyển đổi sang độ F và hiển thị trên LED
* Weblink: liên kết mở Web hiển thị nhiệt độ



Hình 2.22 Giao diện hiển thị nhiệt độ bằng C#

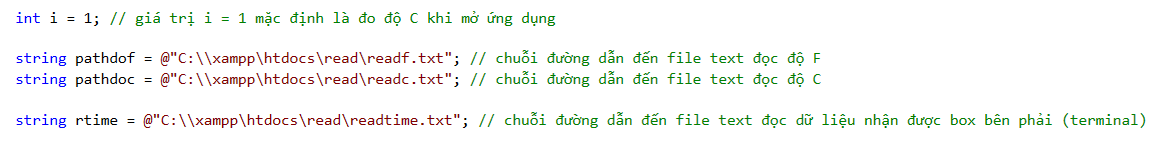
Khai báo cổng PORT vào form với name là UART và tốc độ BaudRate 9600 tương ứng



Hình 2.23 Khai báo cổng Port

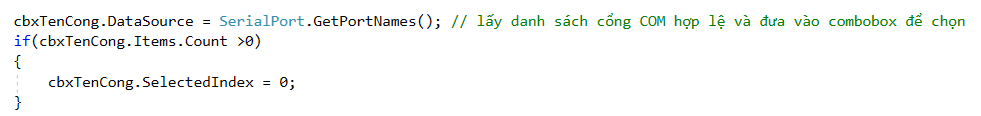
**Khai báo chung:**

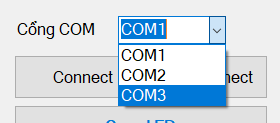
Các biến và chuỗi được sử dụng trong toàn bộ chương trình



Hình 2.24 Code khai báo biến

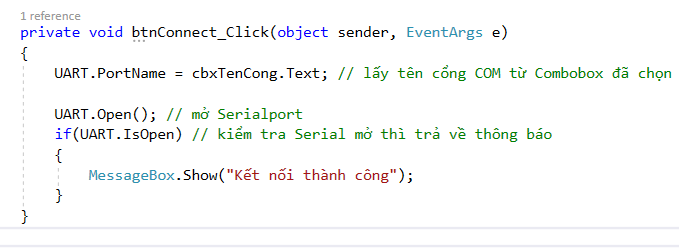
**Lấy ra danh sách các cổng COM**





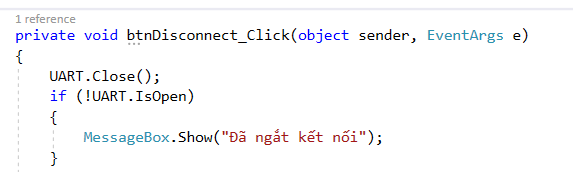
Hình 2.25 Lấy danh sách cổng COM

Kết nối tới cổng COM được chọn



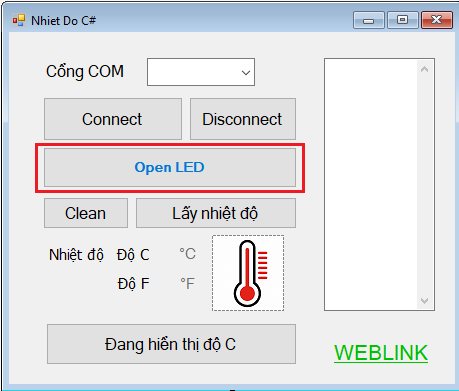
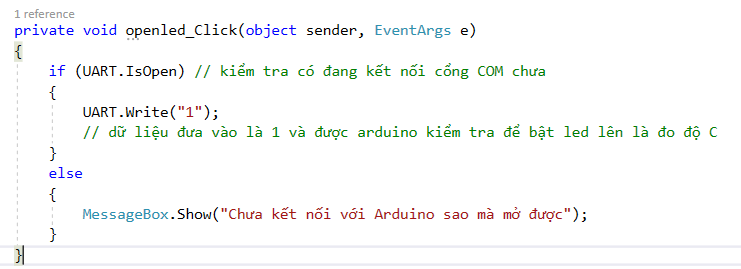
Hình 2.26 Code kết nối tới cổng COM

Ngắt kết nối với cổng COM

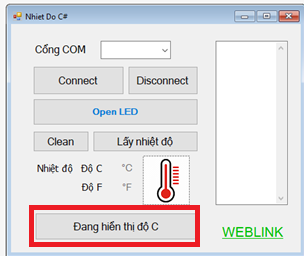


Hình 2.27 Code ngắt kết nối cổng COM

**OpenLed: mở led 7 đoạn lên thông qua C#**

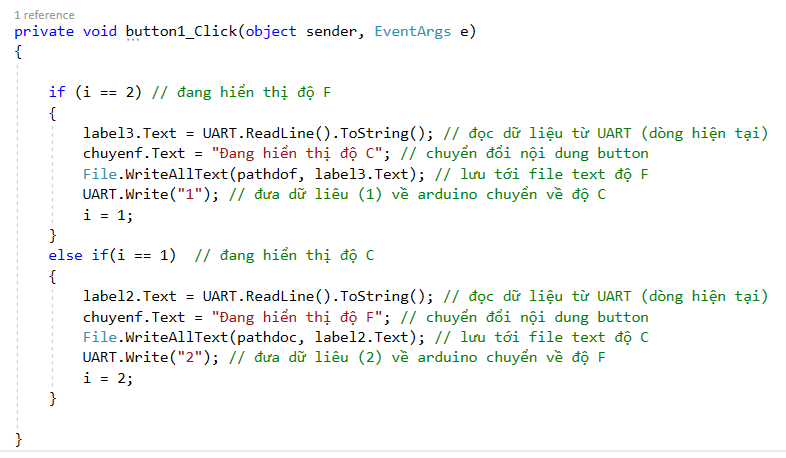


Hình 2.28 Code mở led 7 đoạn

**Thực hiện việc chuyển đổi dữ liệu về nhiệt độ**

Đây là chương trình khi nhấn vào button chuyển nhiệt độ giữa độ C và độ F. Bao gồm lấy nhiệt độ ra

lưu trong file text để hiển thị lên Web



Hình 2.29 Code chuyển đổi nhiệt độ

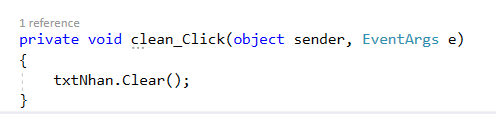
Việc chỉ lấy nhiệt độ hiện tại (không chuyển đổi sang độ F) thì được thực hiện đơn giản hơn phía trên

và cũng sẽ đưa dữ liệu lên web thông qua việc lưu file Text



Hình 2.30 Code lưu file Text

Nút Clean dọn dữ liệu từ box bên phải (terminal)

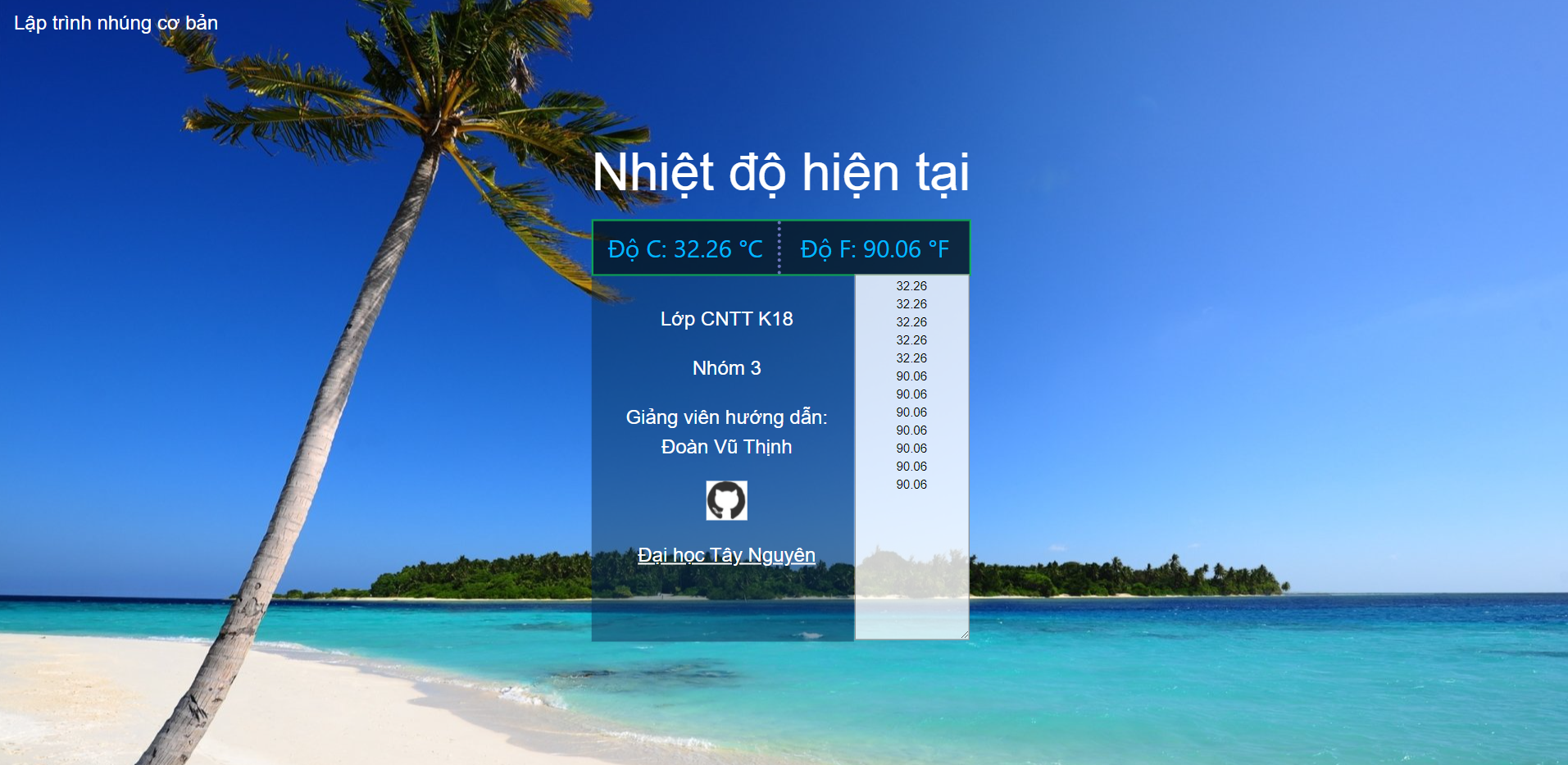


Hình 2.31 Code nút Clear

**2.6.3 Web**

Ngôn ngữ lập trình được sử dụng: html, css, javascript, php

Phương pháp: Đọc file text bằng php sau đó đưa lên html



Hình 2.32 Kết quả hiển thị nhiệt độ trên Web

**PHP**

Với php được sử dụng để đọc file text:

Bao gồm 3 file text được lưu dữ liệu từ C#

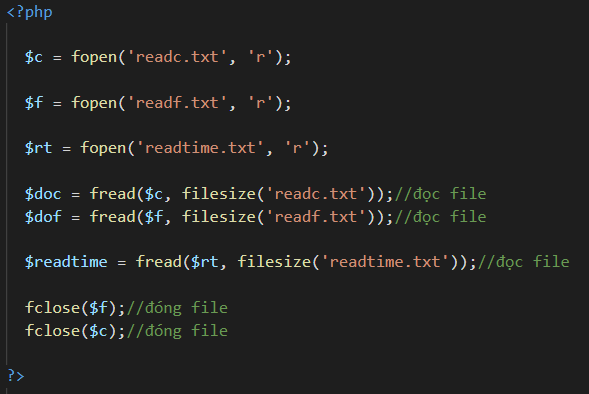
* Readc.txt: nhiệt độ C
* Readf.txt: nhiệt độ F
* Readtime.txt: Hiển thị dữ liệu nhiệt độ (Terminal)

Lưu các biến $c, $f, $rt sử dụng hàm fopen để ở các file tại các đường dẫn.

Fread để đọc file đó bao gồm đường dẫn tới các biến đã lưu để đọc, filesize trỏ tới đường dẫn để

đọc toàn bộ nội dung có trong file text đó. Được lưu trong các biến docc, docf, contents1

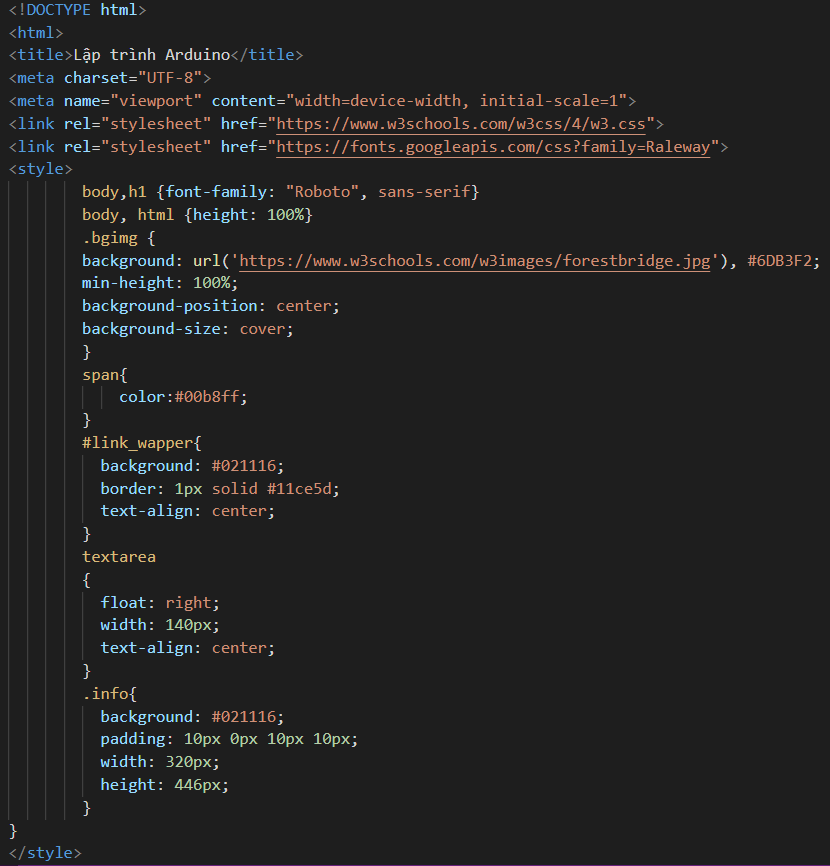
Fclose để đóng file Text đó.

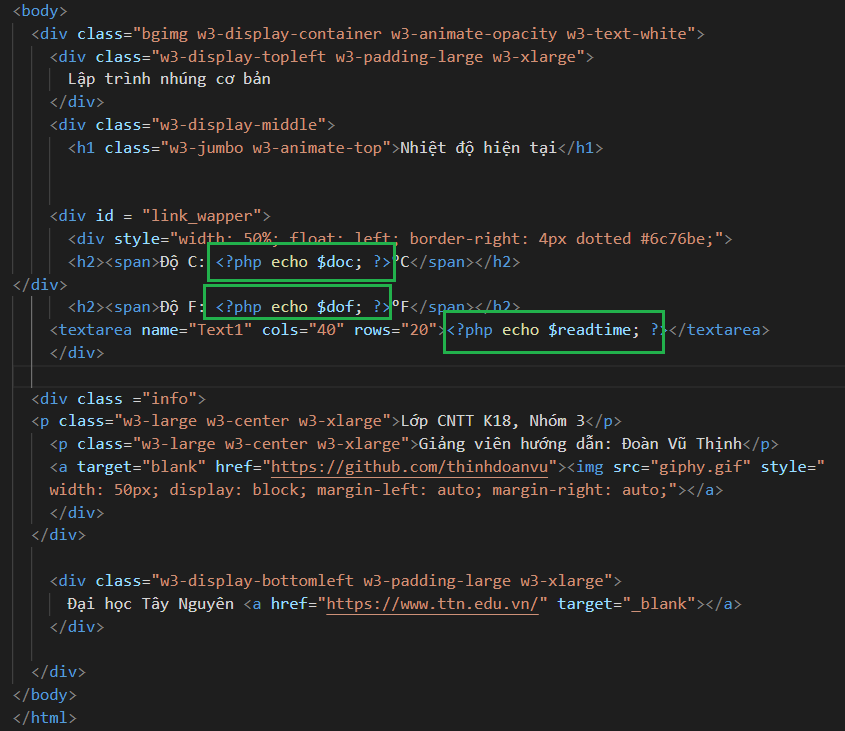


Hình 2.33 Code đọc file text

**HTML:**

sử dụng thêm cả **CSS** vào thẻ style để chỉnh sửa giao diện cho đẹp





Hình 2.34 Code thiết kế web

Trong html truyền vào các biến $doc, $dof, $readtime đã lưu trên php để hiển thị dữ liệu

**JavaScrip**

Hàm khởi tạo XMLHttpRequest

Cập nhật trang web mà không cần tải lại trang

Bên dưới là cú pháp JavaScript phổ biến để sử dụng đối tượng XMLHttpRequest

Trong đó server.php là dữ liệu sẽ được cập nhật theo thời gian.



Hình 2.35 Code cập nhật trang web

Nội dung file server.php Bao gồm đoạn php đọc file text và một đoạn nhỏ html nằm trong Read.php

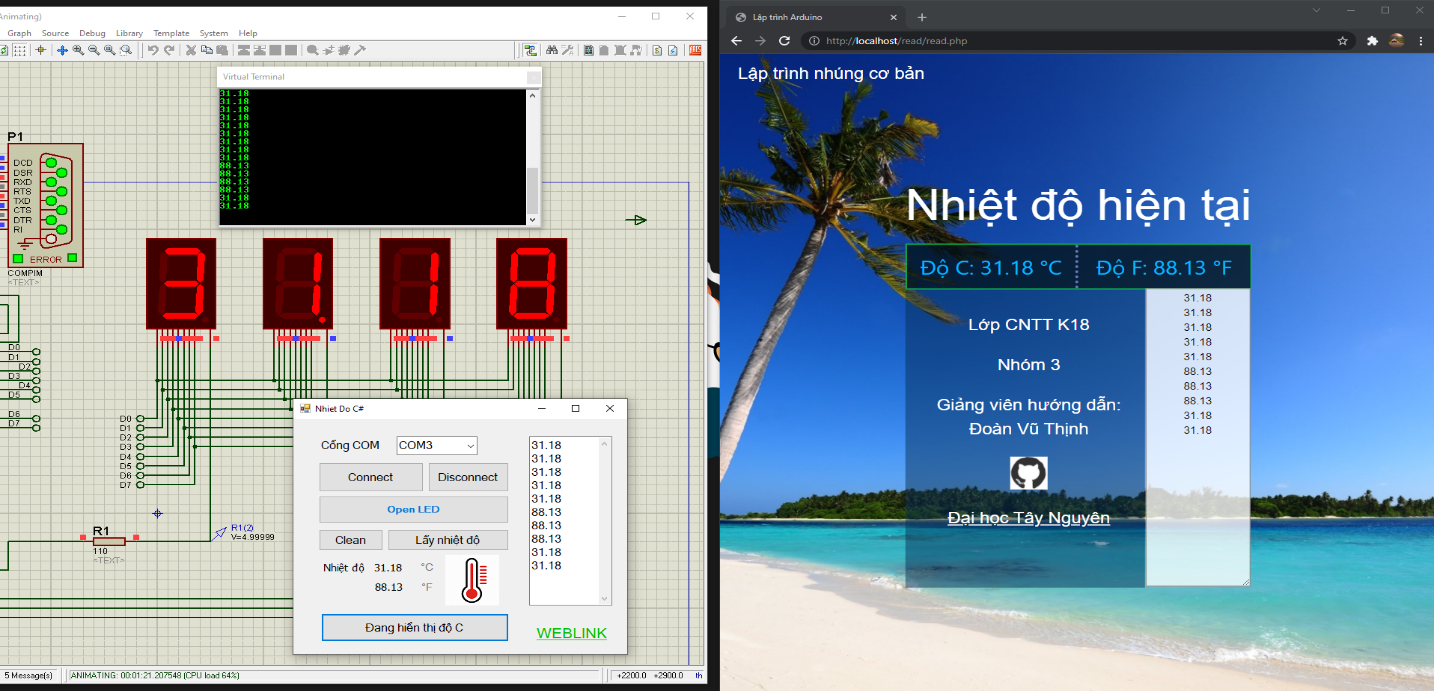
ở trên, đây là nội dung sẽ được update theo thời gian mà không cần tải lại trang.



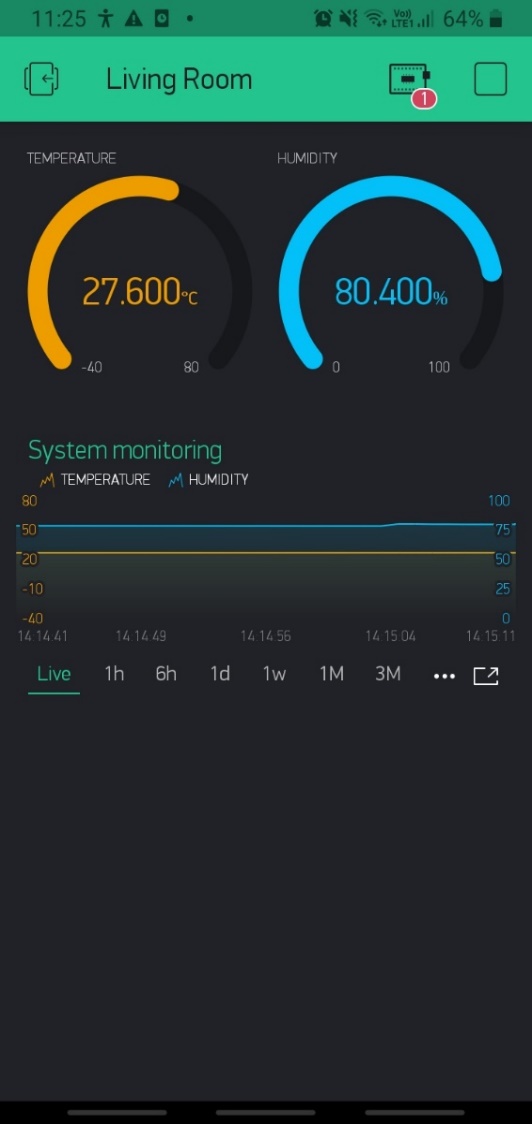
Hình 2.36 Nội dung file server.php

# KẾT QUẢ

Do chưa có mạch thật nên chỉ thực hiện trên mạch mô phỏng.



Hình 3.1 Kết quả hiển thị nhiệt độ trên mạch mô phỏng và Web



Hình 3.2 Kết quả hiển thị nhiệt độ trên App

Link Demo:

# PHÂN CÔNG NHIỆM VỤ VÀ ĐÓNG GÓP

|  |  |  |
| --- | --- | --- |
| **STT** | **Họ và tên** | **Nhiệm vụ** |
| 1 | Nguyễn Văn Huy (Trưởng nhóm) | Phân công nhiệm vụ  Tìm kiếm đề tài và tài liệu hướng dẫn  Mô phỏng mạch Leg 7 đoạn – Arduino trên isis Professtional  Viết code trên C# kết nối với Arduino  Viết code kết nối và trao đổi dữ liệu trên Arduino với C#  Xuất dữ liệu nhiệt độ hiển thị trên Web      Viết báo cáo phương pháp nghiêm cứu App C#, Web |
| 2 | Trần Gia Bảo (Phó nhóm) | Tìm kiếm đề tài và tài liệu hướng dẫn  Viết code arduino hiển thị nhiệt độ trên Led 7 đoạn  Viết báo cáo chi tiết về code trên Arduino  Chỉnh sửa và hoàn thiện báo cáo chung  Làm powerpoint báo cáo |
| 3 | Đỗ Thị Hồng Ngọc | Hiển thị nhiệt độ,độ ẩm trên App Blynk   Hỗ trợ hoàn thiện báo cáo và sản phẩm. |
| 4 | Đỗ Xuân Tiến | Viết báo cáo chung |
| 5 | Trần Thanh Ngân | Thiết kế bảng mạch trên Ares Professtional  Viết báo cáo Led 7 đoạn |
| 6 | Trần Văn Lâm | Thiết kế bảng mạch trên Sprint-layout 5.0 |

# 5. TÀI LIỆU THAM KHẢO

**[Tiếng Việt]**

[1] Cộng đồng Arduino Việt Nam, 01/10/2021, từ <<http://arduino.vn/>>

[2] THINH DOAN VU, 02/10/2021, từ <<https://github.com/thinhdoanvu/Arduino>>

[3] Arduino là gì và ứng dụng của nó, 02/10/2021, từ <<https://quantrimang.com/arduino-la-gi-va-ung-dung-cua-no-trong-cuoc-song-145388>>

**[Tiếng Anh]**

[1] Dean Reading, SevSeg, 01/10/2021, từ <<https://github.com/DeanIsMe/SevSeg>>

[2] XML HttpRequest, 10/10/2021, từ <<https://www.w3schools.com/xml/xml_http.asp>>

[3] Proteus And Visual C# (Arduino Uno Simulator), 10/10/2021, từ <<https://www.youtube.com/watch?v=O0lHtehdcbM>>