**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH  
BỘ MÔN CÔNG NGHỆ THÔNG TIN**

A blue and yellow logo

Description automatically generated with low confidence

**BÁO CÁO THỰC TẬP CHUYÊN MÔN**

**ĐỀ TÀI: XÂY DỰNG WINFORM APP**

**ĐIỀU KHIỂN THIẾT BỊ ĐIỀU HOÀ NHIỆT ĐỘ**

**Thông tin sinh viên thực hiện:**

Họ tên : Nguyễn Cao An Mã sinh viên : 6151071032

Lớp : CQ.61.CNTT Hệ : Chính quy

Ngành đào tạo : Công nghệ thông tin Khoá : 61

Email : 6151071032@st.utc2.edu.vn Số điện thoại : 0915579872

**Thông tin giảng viên hướng dẫn:**

Họ tên : Trần Thi Dung Học vị : Thạc sĩ

Email : ttdung@utc2.edu.vn Số điện thoại : 0388389579

Đơn vị công tác: Đại học Giao thông Vận tải Phân hiệu Thành phố Hồ Chí Minh

Họ tên : Trần Quốc Khánh Học vị : Kỹ sư

Email : tqkhanh@utc2.edu.vn Số điện thoại : 0384742790

Đơn vị công tác: Đại học Giao thông Vận tải Phân hiệu Thành phố Hồ Chí Minh

Thành phố Hồ Chí Minh, ngày 1, tháng 7, năm 2023

MỤC LỤC

[LỜI CẢM ƠN 1](#_Toc139569769)

[NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN 2](#_Toc139569770)

[CHƯƠNG 1 : TỔNG QUAN ĐỀ TÀI 3](#_Toc139569771)

[**I. Giới thiệu đề tài: 3**](#_Toc139569772)

[**II. Lý do chọn đề tài: 3**](#_Toc139569773)

[**III. Mô tả bài toán cần giải quyết: 3**](#_Toc139569774)

[**IV. Mục tiêu thực hiện đề tài 4**](#_Toc139569775)

[**V. Mục đích thực hiện đề tài 4**](#_Toc139569776)

[**VI. Phương pháp nghiên cứu: 4**](#_Toc139569777)

[CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT 6](#_Toc139569778)

[**I. WinForm: 6**](#_Toc139569779)

[**II. IoT: 7**](#_Toc139569780)

[**III. Arduino: 8**](#_Toc139569781)

[**IV. SQL: 9**](#_Toc139569782)

[**V. Serial Port: 10**](#_Toc139569783)

[**VI. Thư viện OneWire: 11**](#_Toc139569784)

[**VII. Thư viện DallasTemperature: 12**](#_Toc139569785)

[**VIII. API OpenWeather: 12**](#_Toc139569786)

[**IX. Phần cứng: 13**](#_Toc139569787)

[**1.Arduino UNO R3 13**](#_Toc139569788)

[**2. Cảm biến nhiệt độ DS18B20 14**](#_Toc139569789)

[**3. Đèn LED 15**](#_Toc139569790)

[**4.Board Test: 16**](#_Toc139569791)

[CHƯƠNG 3 : PHÂN TÍCH THIẾT KẾ HỆ THỐNG 18](#_Toc139569792)

[**I. Mô tả các yêu cầu chức năng của hệ thống: 18**](#_Toc139569793)

[**II. Mô tả các yêu cầu phi chức năng của hệ thống: 18**](#_Toc139569794)

[**III. Tổng quan chức năng của hệ thống: 19**](#_Toc139569795)

[**1. Usecase tổng quát 19**](#_Toc139569796)

[**2. Usecase Xem thông tin nhiệt độ: 20**](#_Toc139569797)

[**3. Usecase Điều khiển nhiệt độ: 20**](#_Toc139569798)

[**4. Sơ đồ phân rã chức năng (BFD): 21**](#_Toc139569799)

[**IV. Xây dựng cơ sở dữ liệu: 21**](#_Toc139569800)

[**1. Bảng cơ sở dữ liệu: 21**](#_Toc139569801)

[**2. Bảng mô tả chi tiết cơ sở dữ liệu 21**](#_Toc139569802)

[CHƯƠNG 4 : XÂY DỰNG CHƯƠNG TRÌNH 22](#_Toc139569803)

[**I. Đăng nhập: 22**](#_Toc139569804)

[**II. Xem thông tin thời tiết 23**](#_Toc139569805)

[**III. Điều khiển thiết bị điều hoà nhiệt độ 24**](#_Toc139569806)

[CHƯƠNG 5: KẾT LUẬN 27](#_Toc139569807)

[**I. Kết quả đạt được 27**](#_Toc139569808)

[**II. Hạn chế còn tồn tại của hệ thống 27**](#_Toc139569809)

[**III. Hướng phát triển hệ thống trong tương lai 28**](#_Toc139569810)

[TÀI LIỆU THAM KHẢO 29](#_Toc139569811)

# DANH MỤC HÌNH ẢNH

[Hình 2.1: Arduino UNO R3 13](#_Toc139569747)

[Hình 2.2: Cảm biến nhiệt độ DS18B20 14](#_Toc139569748)

[Hình 2.3: Đèn LED 15](#_Toc139569749)

[Hình 2.4: Board Test 16](#_Toc139569750)

[Hình 3.1: Sơ đồ Usecase tổng quát 19](#_Toc139569751)

[Hình 3.2: Sơ đồ Usecase xem thông tin nhiệt độ 20](#_Toc139569752)

[Hình 3.3: Sơ đồ Usecase điều khiển nhiệt độ 20](#_Toc139569753)

[Hình 3.4: Sơ đồ phân rã chức năng (BFD) 21](#_Toc139569754)

[Hình 3.5: Ảnh Bảng users 21](#_Toc139569755)

[Hình 4.1: Giao diện đăng nhập 22](#_Toc139569756)

[Hình 4.2: Thông báo thông tin đăng nhập không hợp lệ 22](#_Toc139569757)

[Hình 4.3: Thông báo đăng nhập thành công 23](#_Toc139569758)

[Hình 4.2: Giao diện InfoWeather 23](#_Toc139569759)

[Hình 4.5: Giao diện Setting 24](#_Toc139569760)

[Hình 4.6: Phần mền đưa ra khuyến chị điều chỉnh nhiệt độ 25](#_Toc139569761)

[Hình 4.7: Trạng thái thiết bị hoạt động 25](#_Toc139569762)

[Hình 4.8: Thiết bị đang hoạt động (mô phỏng bằng đèn LED) 26](#_Toc139569763)

[Hình 4.9: Trạng thái thiết bị tắt 26](#_Toc139569764)

[Hình 4.10: Thiết bị đang tắt (mô phỏng bằng đèn LED) 27](#_Toc139569765)

# DANH MỤC BẢNG BIỂU

[Bảng 3.1: Bảng mô tả chi tiết cơ sở dữ liệu 21](#_Toc139569766)

# LỜI CẢM ƠN

Quá trình hoàn thành đồ án không chỉ là một thử thách về kiến thức mà còn là một hành trình phát triển cá nhân và chuyên môn của em. Em xin gửi lời cảm ơn sâu sắc và lòng biết ơn chân thành đến quý thầy cô **Bộ môn Công nghệ thông tin Trường Đại học Giao Thông Vận Tải phân hiệu tại Thành phố Hồ Chí Minh** đã dành thời gian và công sức để hướng dẫn và hỗ trợ chúng em trong suốt thời gian thực hiện đề tài.

Quý thầy cô đã không ngừng truyền đạt kiến thức, chia sẻ kinh nghiệm và tạo điều kiện để em có thể áp dụng những kiến thức đã học vào thực tế. Sự tận tâm và sự chỉ bảo của quý thầy cô đã giúp em vượt qua những khó khăn, khám phá và phát triển khả năng tiềm năng của chúng em.

Thực tập chuyên môn không chỉ là một nhiệm vụ cá nhân, mà còn là sự kết hợp của sự hỗ trợ từ quý thầy cô. Em xin chân thành cảm ơn tất cả quý thầy cô bộ môn, đặc biệt là cô Trần Thị Dung và thầy Trần Quốc Khánh vì sự đồng hành và sự tin tưởng vào khả năng của em. Điều này đã truyền cảm hứng và động lực cho em để vượt qua mọi thử thách và hoàn thành đề tài một cách xuất sắc.

Em xin gửi lời chúc tốt đẹp nhất đến tất cả các quý thầy cô bộ môn. Mong rằng thầy cô sẽ tiếp tục mang đến sự tận tâm và sự cống hiến trong việc truyền đạt kiến thức cho các thế hệ sinh viên tương lai.

Trân trọng.

# NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..…………………………………………………..………………………..………………………………………………………………………………………………………………………………………………………………………………………………………….…………………………………………………………………………………………………...……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

# CHƯƠNG 1 : TỔNG QUAN ĐỀ TÀI

## Giới thiệu đề tài:

Trong thời đại công nghệ hiện đại, việc sử dụng thiết bị điều hòa nhiệt độ đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày của chúng ta. Thiết bị này giúp chúng ta duy trì một môi trường sống và làm việc thoải mái và tiện nghi. Tuy nhiên, việc điều khiển và quản lý nhiệt độ trong không gian sống và làm việc vẫn còn gặp một số thách thức.

Với mục tiêu tạo ra một ứng dụng WinForm để điều khiển thiết bị điều hòa nhiệt độ, đề tài thực tập chuyên môn này đã được tiến hành. Qua quá trình thực hiện đề tài, em đã áp dụng các phương pháp nghiên cứu và phát triển ứng dụng WinForm để xây dựng một giao diện người dùng thân thiện và dễ sử dụng. Đồng thời, em cũng đã tích hợp và hiển thị thông tin nhiệt độ từ thiết bị điều khiển, cho phép người dùng điều khiển và điều chỉnh nhiệt độ, cùng với tính năng cài đặt nhiệt độ ở chế độ auto lấy dữ liệu điều kiện của API OpenWeather

## Lý do chọn đề tài:

Do nhu cầu điều khiển và quản lý nhiệt độ trong không gian sống và làm việc ngày càng tăng. Thiết bị điều hòa nhiệt độ đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày của chúng ta, giúp duy trì một môi trường sống và làm việc thoải mái và tiện nghi. Tuy nhiên, việc điều khiển và quản lý nhiệt độ vẫn còn gặp một số thách thức.

Chúng ta cần một giao diện người dùng thân thiện và dễ sử dụng để điều khiển thiết bị điều hòa nhiệt độ. WinForm App cung cấp một giao diện người dùng đồ họa trực quan và dễ sử dụng, giúp người dùng tương tác và điều khiển thiết bị một cách thuận tiện và hiệu quả. Với giao diện người dùng được thiết kế một cách rõ ràng và trực quan, người dùng có thể dễ dàng điều chỉnh và kiểm soát nhiệt độ theo nhu cầu của mình.

Một lý do khác là tính linh hoạt và dễ dàng mở rộng của ứng dụng WinForm điều khiển thiết bị điều hòa nhiệt độ. Bằng cách sử dụng các thành phần và công cụ có sẵn trong viện dẫn.NET, chúng ta có thể linh hoạt tạo ra các chức năng và tính năng mới cho ứng dụng. Điều này giúp người dùng có thể tùy chỉnh và quản lý nhiệt độ một cách linh hoạt và hiệu quả.

Lý do tiếp theo là khả năng điều khiển và điều chỉnh nhiệt độ từ xa thông qua ứng dụng WinForm. Việc này mang lại sự tiện lợi và linh hoạt cho người dùng, cho phép họ điều chỉnh nhiệt độ từ xa.

## Mô tả bài toán cần giải quyết:

Trong thời đại 4.0, công nghệ phát triển như vũ bão thì việc ứng dung công nghệ để đáp ứng những nhu cầu sống cơ bản của con người là vô cùng cần thiết. Để đảm bảo nhiệt độ trong không gian làm việc được thoả mái thì các cá nhân, công ty, xí nghiệp cần một phần mềm quản lí nhiệt độ để có thể nắm được các thông tin như nhiệt độ thực tế hiện tại, nhiệt độ dự báo của các nguồn thông tin khác, để rồi có thể sử dụng phần mềm để bật tắt các thiết bị điều hoà nhiệt độ để đưa nhiệt độ về mức mong muốn

## Mục tiêu thực hiện đề tài

**Tạo ra một ứng dụng WinForm dễ sử dụng và thân thiện với người dùng: Mục tiêu là tạo ra một giao diện người dùng đồ họa trực quan, giúp người dùng dễ dàng tương tác và điều khiển thiết bị điều hòa nhiệt độ một cách thuận tiện và hiệu quả.**

**Tích hợp và hiển thị thông tin nhiệt độ từ thiết bị điều khiển: Mục tiêu là kết nối và lấy dữ liệu nhiệt độ từ thiết bị điều khiển, sau đó hiển thị thông tin nhiệt độ lên giao diện người dùng của ứng dụng WinForm. Điều này giúp người dùng có thể theo dõi và kiểm soát nhiệt độ trong không gian sống hoặc làm việc của mình.**

**Điều khiển và điều chỉnh nhiệt độ từ xa: Mục tiêu là cho phép người dùng điều khiển và điều chỉnh nhiệt độ từ xa thông qua ứng dụng WinForm. Việc này giúp tiết kiệm thời gian và năng lượng, cho phép người dùng điều chỉnh nhiệt độ theo nhu cầu và tạo ra một môi trường thoải mái và tiết kiệm năng lượng.**

**Cung cấp các tính năng bổ sung: Mục tiêu là tích hợp các tính năng bổ sung như lịch trình đặt nhiệt độ, cảnh báo khi nhiệt độ vượt quá giới hạn, và thống kê nhiệt độ theo thời gian. Điều này giúp người dùng có thể tùy chỉnh và quản lý nhiệt độ một cách linh hoạt và hiệu quả.**

**Đảm bảo một số yêu cầu phi chức năng: Hiệu suất của phầm mềm được đảm bảo, phản hồi nhanh trong quá trình đọc ghi dữ liệu nhiệt độ lên xuống Arduino. Ứng dụng đảm bảo độ tin cậy, độ chính xác cao. Giao diện người dùng thân thiện, trực quan, dễ sử dụng**

## ****Mục đích thực hiện đề tài****

Lý do chọn đề tài này là vì nhu cầu điều khiển và quản lý nhiệt độ trong không gian sống và làm việc ngày càng tăng. Xây dựng một ứng dụng WinForm để điều khiển thiết bị điều hòa nhiệt độ không chỉ mang lại sự tiện lợi và linh hoạt cho người dùng, mà còn đóng góp vào việc tiết kiệm năng lượng và tạo ra một môi trường sống và làm việc thoải mái hơn.

## Phương pháp nghiên cứu:

Tìm hiểu về thiết bị điều khiển nhiệt độ: Nghiên cứu về các thiết bị điều khiển nhiệt độ có sẵn trên thị trường, hiểu về cách hoạt động và giao tiếp với thiết bị.

Phân tích yêu cầu và thiết kế ứng dụng: Xác định các yêu cầu chức năng và giao diện người dùng cho ứng dụng WinForm. Thiết kế giao diện người dùng và các chức năng điều khiển nhiệt độ.

Xây dựng ứng dụng WinForm: Sử dụng ngôn ngữ lập trình C# và công cụ phát triển Visual Studio, xây dựng ứng dụng WinForm theo thiết kế đã được xác định. Tạo các thành phần giao diện người dùng, kết nối và gửi yêu cầu điều khiển đến thiết bị điều khiển nhiệt độ.

Kiểm thử và sửa lỗi: Tiến hành kiểm thử ứng dụng để đảm bảo tính ổn định và đáp ứng đúng các yêu cầu chức năng. Sửa lỗi và cải thiện hiệu suất của ứng dụng.

Đánh giá và tối ưu: Đánh giá hiệu suất và tính năng của ứng dụng thông qua việc sử dụng thực tế và thu thập phản hồi từ người dùng. Tối ưu hóa ứng dụng để đảm bảo tốc độ và hiệu suất tốt nhất.

Triển khai và triển khai: Chuẩn bị ứng dụng để triển khai vào môi trường thực tế. Cài đặt ứng dụng trên các thiết bị điều khiển nhiệt độ và cung cấp hướng dẫn sử dụng cho người dùng.

Đánh giá kết quả và kết luận: Đánh giá kết quả của đề tài và rút ra kết luận về hiệu suất và tính khả thi của ứng dụng WinForm điều khiển thiết bị điều hòa nhiệt độ.

Phương pháp nghiên cứu này sẽ đảm bảo quá trình xây dựng ứng dụng WinForm điều khiển thiết bị điều hòa nhiệt độ diễn ra một cách có hệ thống và đạt được mục tiêu đề ra.

# CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT

## WinForm:

Windows Forms (WinForms)là một công nghệ phát triển ứng dụng desktop trong .NET Framework của Microsoft. Nó cung cấp một cách dễ dàng để tạo và quản lý các giao diện người dùng trong các ứng dụng Windows. Dưới đây là cơ sở lý thuyết đầy đủ của WinForms:

* Form (Biểu mẫu):
  + Form là cửa sổ chính của ứng dụng WinForms, nơi chứa các thành phần giao diện người dùng khác nhau.
  + Mỗi Form có một tên duy nhất và có thể chứa nhiều controls (điều khiển) khác nhau.
* Control (Điều khiển):
  + Control là các thành phần giao diện người dùng như nút (Button), hộp văn bản (TextBox), danh sách (ListBox), và nhiều hơn nữa.
  + Các control được sử dụng để hiển thị thông tin, chấp nhận đầu vào từ người dùng và thực hiện các hành động.
* Events (Sự kiện):
  + Events là các hành động xảy ra trên controls, chẳng hạn như nhấp chuột, nhấn phím hoặc thay đổi giá trị.
  + Mỗi control có một danh sách các events mà bạn có thể gắn kết với các phương thức để xử lý các sự kiện đó.
* Handlers (Xử lý sự kiện):
  + Handlers là các phương thức được gọi khi một sự kiện xảy ra.
  + Có thể viết các phương thức xử lý sự kiện để thực hiện các hành động cụ thể khi sự kiện xảy ra.
* Properties (Thuộc tính):
  + Properties là các thuộc tính của controls và form, cho phép bạn cấu hình và tùy chỉnh các thành phần giao diện người dùng.
  + Ví dụ: màu nền, kích thước, vị trí, văn bản, và nhiều thuộc tính khác.
* Layout (Bố cục):

Layout là cách các controls được sắp xếp và tổ chức trên form.

* Bạn có thể sử dụng các layout managers như FlowLayoutPanel, TableLayoutPanel, hoặc định vị bằng tay để quản lý vị trí và kích thước của các controls.
* Data Binding (Ràng buộc dữ liệu):
  + Data binding là quá trình liên kết dữ liệu từ nguồn dữ liệu (như cơ sở dữ liệu hoặc danh sách) tới các controls.
  + Điều này cho phép tự động cập nhật dữ liệu trên các controls khi dữ liệu thay đổi.
* Dialogs (Hộp thoại):
  + Dialogs là các cửa sổ phụ xuất hiện trong quá trình làm việc với ứng dụng.
  + Các dialog thường được sử dụng để yêu cầu đầu vào từ người dùng hoặc hiển thị thông báo.
* Threading (Luồng):
  + Threading là quá trình sử dụng nhiều luồng để thực hiện các tác vụ đồng thời trong ứng dụng.
  + Điều này giúp tránh tình trạng đứng chương trình khi thực hiện các tác vụ mất thời gian.

WinForms cung cấp một cấu trúc và các công cụ hỗ trợ để phát triển ứng dụng desktop Windows dễ dàng và linh hoạt. Với các khái niệm và thành phần trên, bạn có thể thiết kế và xây dựng giao diện người dùng phong phú và tương tác trong ứng dụng của mình.

## IoT:

IoT (Internet of Things) là một hệ thống mạng liên kết các thiết bị vật lý và đối tượng thông qua internet, cho phép chúng giao tiếp và trao đổi dữ liệu. Dưới đây là cơ sở lý thuyết đầy đủ của IoT:

* Thiết bị IoT (IoT Devices):
  + Thiết bị IoT là các thiết bị vật lý như cảm biến, bộ điều khiển, máy móc, thiết bị đeo tay, và nhiều thiết bị khác, có khả năng thu thập và truyền dữ liệu.
* Kết nối (Connectivity):
  + Kết nối là quá trình liên kết các thiết bị IoT với nhau và với mạng internet.
  + Các công nghệ kết nối phổ biến trong IoT bao gồm Wi-Fi, Bluetooth, Zigbee, LoRa, và 4G/5G.
* Mạng (Network):
  + Mạng là hệ thống cơ sở hạ tầng để truyền dữ liệu giữa các thiết bị IoT và các hệ thống xử lý dữ liệu.
  + Mạng có thể là mạng cục bộ (Local Area Network - LAN), mạng rộng (Wide Area Network - WAN), hoặc mạng không dây (Wireless Network).
* Nền tảng IoT (IoT Platform):
  + Nền tảng IoT là một hệ thống phần mềm để quản lý và kiểm soát các thiết bị IoT, thu thập và xử lý dữ liệu, và cung cấp các dịch vụ và ứng dụng IoT.
  + Nền tảng IoT cung cấp giao diện lập trình ứng dụng (API) để phát triển các ứng dụng IoT.
* Dữ liệu (Data):
  + Dữ liệu là thông tin thu thập từ các thiết bị IoT, bao gồm các giá trị đo lường từ cảm biến, trạng thái hoạt động và thông tin khác.
  + Dữ liệu IoT có thể được sử dụng để phân tích, điều khiển và đưa ra quyết định thông minh.
* Bảo mật (Security):
  + Bảo mật là một yếu tố quan trọng trong IoT để bảo vệ dữ liệu và thiết bị khỏi các mối đe dọa và tấn công.
  + Các biện pháp bảo mật IoT bao gồm xác thực, mã hóa dữ liệu, kiểm soát truy cập và giám sát.
* Ứng dụng IoT (IoT Applications):
  + Ứng dụng IoT là các ứng dụng và giải pháp được xây dựng trên nền tảng IoT để giải quyết các vấn đề và mang lại lợi ích trong nhiều lĩnh vực như nông nghiệp, y tế, công nghiệp, giao thông và gia đình thông minh.
* Phân tích dữ liệu (Data Analytics):
  + Phân tích dữ liệu là quá trình xử lý và phân tích dữ liệu IoT để tìm ra thông tin quan trọng, xu hướng và mô hình từ dữ liệu thu thập.
  + Phân tích dữ liệu IoT giúp tăng cường khả năng dự đoán, quản lý tài nguyên và cải thiện hiệu suất.
* Quản lý thiết bị (Device Management):
  + Quản lý thiết bị là quá trình quản lý và kiểm soát các thiết bị IoT, bao gồm cài đặt, cập nhật phần mềm, theo dõi trạng thái và phát hiện lỗi.
  + Quản lý thiết bị IoT giúp đảm bảo hoạt động ổn định và hiệu quả của các thiết bị trong hệ thống.

IoT đang ngày càng phát triển và mang lại nhiều tiềm năng trong việc kết nối và tận dụng dữ liệu từ các thiết bị vật lý. Với cơ sở lý thuyết trên, bạn có thể hiểu và áp dụng IoT trong các ứng dụng và giải pháp của mình.

## Arduino:

Arduino là một nền tảng phát triển phần cứng và phần mềm mã nguồn mở, được sử dụng rộng rãi trong việc xây dựng các dự án điện tử và IoT. Dưới đây là cơ sở lý thuyết đầy đủ của Arduino:

* Arduino Board:
  + Arduino Board là một bo mạch điện tử nhỏ gọn, có thể lập trình để thực hiện các hoạt động điều khiển và giao tiếp với các linh kiện khác.
  + Arduino Board bao gồm một vi điều khiển (microcontroller), các chân giao tiếp (pins), và các thành phần khác như nguồn cấp và chân kết nối.
* Vi điều khiển (Microcontroller):
  + Vi điều khiển là bộ xử lý trung tâm của Arduino Board, có khả năng thực hiện các lệnh và điều khiển các thiết bị ngoại vi.
  + Arduino sử dụng vi điều khiển AVR hoặc ARM, có tốc độ xử lý và bộ nhớ lưu trữ khác nhau.
* Chân Giao tiếp (Pins):
  + Chân Giao tiếp là các chân trên Arduino Board được sử dụng để kết nối và giao tiếp với các linh kiện và thiết bị khác nhau.
  + Có hai loại chân giao tiếp trên Arduino: chân số (digital pins) và chân analog (analog pins).
* Ngôn ngữ lập trình (Programming Language):
  + Arduino sử dụng một phiên bản đơn giản của ngôn ngữ C/C++ để lập trình.
  + Ngôn ngữ lập trình Arduino cung cấp các thư viện (libraries) và hàm (functions) để tương tác với các linh kiện và thực hiện các hoạt động điều khiển.
* IDE Arduino (Arduino Integrated Development Environment):
  + IDE Arduino là một môi trường phát triển tích hợp, được sử dụng để viết và tải chương trình vào Arduino Board.
  + IDE cung cấp các công cụ và giao diện đồ họa để lập trình, gỡ lỗi và tải chương trình.
* Thư viện (Libraries):
  + Thư viện là một tập hợp các mã nguồn đã được viết sẵn để giúp lập trình viên thực hiện các chức năng cụ thể một cách dễ dàng.

## SQL:

SQL (Structured Query Language) là một ngôn ngữ lập trình được sử dụng để quản lý và truy vấn cơ sở dữ liệu quan hệ (Relational Database Management System - RDBMS). Dưới đây là cơ sở lý thuyết đầy đủ của SQL:

* Cơ sở dữ liệu (Database):
  + Cơ sở dữ liệu là một tập hợp các dữ liệu có tổ chức được lưu trữ trong hệ thống quản lý cơ sở dữ liệu (Database Management System - DBMS).
  + Có nhiều loại cơ sở dữ liệu, trong đó phổ biến nhất là cơ sở dữ liệu quan hệ (Relational Database).
* Bảng (Table):
  + Bảng là một đơn vị lưu trữ dữ liệu trong cơ sở dữ liệu quan hệ.
  + Bảng gồm các hàng (rows) và cột (columns), trong đó mỗi hàng đại diện cho một bản ghi (record) và mỗi cột đại diện cho một thuộc tính (attribute).
* Truy vấn (Query):
  + Truy vấn là một yêu cầu được gửi đến cơ sở dữ liệu để truy xuất hoặc thay đổi dữ liệu.
  + SQL cung cấp các câu lệnh truy vấn như SELECT, INSERT, UPDATE, DELETE để thực hiện các hoạt động trên cơ sở dữ liệu.
    - Câu lệnh SELECT:
      * Câu lệnh SELECT được sử dụng để truy vấn dữ liệu từ cơ sở dữ liệu.
      * Cú pháp SELECT column1, column2 FROM table WHERE condition; được sử dụng để lựa chọn các cột và hàng dữ liệu dựa trên điều kiện.
    - Câu lệnh INSERT:
      * Câu lệnh INSERT được sử dụng để thêm dữ liệu mới vào cơ sở dữ liệu.
      * Cú pháp INSERT INTO table (column1, column2) VALUES (value1, value2); được sử dụng để chèn dữ liệu vào bảng.
    - Câu lệnh UPDATE:
      * Câu lệnh UPDATE được sử dụng để cập nhật dữ liệu trong cơ sở dữ liệu.
      * Cú pháp UPDATE table SET column1 = value1, column2 = value2 WHERE condition; được sử dụng để cập nhật giá trị của các cột dựa trên điều kiện.
    - Câu lệnh DELETE:
      * Câu lệnh DELETE được sử dụng để xóa dữ liệu khỏi cơ sở dữ liệu.
      * Cú pháp DELETE FROM table WHERE condition; được sử dụng để xóa các hàng dữ liệu dựa trên điều kiện.
    - Ràng buộc (Constraints):
      * Ràng buộc là các quy tắc được áp dụng cho cơ sở dữ liệu để đảm bảo tính toàn vẹn và hợp lệ của dữ liệu.
      * Ví dụ: ràng buộc khóa chính (primary key) để đảm bảo duy nhất giá trị của một cột trong bảng.
    - Liên kết (Join):
      * Liên kết là quá trình kết hợp dữ liệu từ nhiều bảng dựa trên một điều kiện chung.
      * Có nhiều loại liên kết như INNER JOIN, LEFT JOIN, RIGHT JOIN và FULL JOIN.

SQL cung cấp một cách mạnh mẽ để quản lý và truy vấn cơ sở dữ liệu quan hệ. Với các khái niệm và câu lệnh trên, bạn có thể tạo, truy vấn và thao tác dữ liệu trong cơ sở dữ liệu của mình.

## Serial Port:

Cổng Serial (Serial Port) là một giao diện truyền thông mạng dùng để truyền dữ liệu theo chuỗi bit (bit-by-bit) qua một kết nối vật lý. Nó được sử dụng rộng rãi trong các ứng dụng truyền thông và giao tiếp giữa các thiết bị điện tử, bao gồm máy tính, vi điều khiển, máy in, modem và nhiều thiết bị khác.

Cổng Serial sử dụng một số dây truyền thông để truyền dữ liệu. Các dây quan trọng bao gồm:

* Dây TX (Transmit): Dùng để truyền dữ liệu từ nguồn gốc (như máy tính) đến thiết bị nhận (như vi điều khiển).
* Dây RX (Receive): Dùng để nhận dữ liệu từ thiết bị gửi (như vi điều khiển) đến nguồn đích (như máy tính).
* Dây GND (Ground): Dùng để định mức chung giữa các thiết bị.

Cổng Serial hoạt động dựa trên nguyên tắc truyền dữ liệu theo chuỗi bit. Dữ liệu được chia thành các khung (frames) và truyền bit theo thứ tự từ bit thấp nhất đến bit cao nhất. Mỗi khung dữ liệu bao gồm các bit điều khiển (start bit, stop bit) để đồng bộ hóa và đảm bảo đúng thứ tự truyền nhận dữ liệu.

Các thông số cấu hình quan trọng của Cổng Serial bao gồm tốc độ truyền (baud rate), độ dài khung dữ liệu (data frame length), kiểu dữ liệu (parity), và số bit stop (stop bits). Tốc độ truyền xác định số lượng bit truyền qua kênh trong một đơn vị thời gian, độ dài khung dữ liệu xác định số bit dữ liệu trong mỗi khung, kiểu dữ liệu kiểm soát tính chẵn lẻ của dữ liệu, và số bit stop xác định số bit dừng sau mỗi khung dữ liệu.

Cổng Serial cung cấp một phương thức truyền dữ liệu đơn giản và đáng tin cậy giữa các thiết bị. Nó đã được sử dụng rộng rãi trong quá khứ và vẫn còn được sử dụng trong nhiều ứng dụng hiện đại. Tuy nhiên, với sự phát triển của công nghệ, cổng Serial dần được thay thế bởi các giao diện truyền thông mới như USB, Ethernet và Bluetooth.

## Thư viện OneWire:

Thư viện OneWire là một thư viện phần mềm được sử dụng rộng rãi trong các dự án điện tử để giao tiếp với các cảm biến và thiết bị sử dụng giao tiếp OneWire. Giao tiếp OneWire là một giao thức đơn dây đơn giản và tiết kiệm tài nguyên, cho phép người dùng kết nối và giao tiếp với nhiều thiết bị trên cùng một dây duy nhất.

Thư viện OneWire cung cấp một loạt các chức năng và phương thức để thực hiện giao tiếp OneWire. Bằng cách sử dụng thư viện này, người dùng có thể tìm kiếm và xác định địa chỉ của các thiết bị OneWire được kết nối, đọc và ghi dữ liệu từ các thiết bị, cũng như điều khiển và cấu hình chúng.

Một trong những tính năng quan trọng của thư viện OneWire là khả năng tìm kiếm và xác định địa chỉ của các thiết bị OneWire. Thông qua việc gửi các tín hiệu và nhận phản hồi từ các thiết bị, thư viện OneWire cho phép người dùng định danh và xác định các thiết bị được kết nối một cách dễ dàng và chính xác. Điều này rất hữu ích khi bạn có nhiều thiết bị OneWire trong mạch và cần xác định chính xác địa chỉ của từng thiết bị.

Thư viện OneWire cũng cung cấp các phương thức để đọc và ghi dữ liệu từ các thiết bị OneWire. Bằng cách sử dụng các phương thức này, người dùng có thể truy cập và lấy dữ liệu từ các cảm biến nhiệt độ, cảm biến ánh sáng, bộ nhớ EEPROM và nhiều loại thiết bị khác. Điều này cho phép người dùng thu thập thông tin từ các cảm biến và sử dụng dữ liệu thu được để điều khiển các thiết bị và thực hiện các chức năng khác.

Thư viện OneWire cũng cho phép người dùng điều khiển và cấu hình các thiết bị OneWire. Bằng cách gửi các lệnh và tín hiệu điều khiển, người dùng có thể điều chỉnh các thiết bị OneWire để thực hiện các chức năng cụ thể hoặc thay đổi cấu hình của chúng. Điều này tạo ra sự linh hoạt và tùy chỉnh trong việc sử dụng các thiết bị OneWire theo nhu cầu của dự án.

Thư viện OneWire thường được sử dụng trong các dự án Arduino và các hệ thống nhúng khác để giao tiếp với các cảm biến và thiết bị OneWire. Với cú pháp dễ hiểu và các chức năng mạnh mẽ, thư viện OneWire giúp người dùng dễ dàng và tiện lợi trong việc truyền và nhận dữ liệu từ các thiết bị OneWire. Nó giúp người dùng tập trung vào việc phân tích và sử dụng dữ liệu thu được từ các cảm biến và thiết bị này, thúc đẩy sự phát triển và ứng dụng của công nghệ OneWire trong các dự án điện tử.

## Thư viện DallasTemperature:

Thư viện DallasTemperature là một thư viện phần mềm được sử dụng trong các dự án điện tử để giao tiếp và điều khiển các cảm biến nhiệt độ của hãng Dallas Semiconductor. Thư viện này cung cấp các phương thức và chức năng đơn giản để đọc và điều khiển các cảm biến nhiệt độ 1-Wire của Dallas Semiconductor.

Cảm biến nhiệt độ 1-Wire của Dallas Semiconductor là một loại cảm biến thông minh, sử dụng giao tiếp OneWire để truyền dữ liệu và điều khiển. Các cảm biến này có khả năng đo nhiệt độ chính xác và có thể được sử dụng trong nhiều ứng dụng khác nhau như điều khiển nhiệt độ, giám sát môi trường, và hệ thống tự động hóa.

Thư viện DallasTemperature hỗ trợ nhiều loại cảm biến nhiệt độ của Dallas Semiconductor như DS18B20, DS18S20, DS1822 và nhiều loại khác. Các phương thức và chức năng trong thư viện cho phép người dùng dễ dàng đọc giá trị nhiệt độ từ cảm biến, thiết lập độ phân giải, đọc và ghi các giá trị điều khiển của cảm biến, và thực hiện các chức năng khác liên quan đến cảm biến nhiệt độ.

Thư viện DallasTemperature được sử dụng phổ biến trong các dự án Arduino và các hệ thống nhúng khác. Với cú pháp dễ sử dụng và tính năng mạnh mẽ, thư viện này giúp người dùng dễ dàng tích hợp và sử dụng các cảm biến nhiệt độ của Dallas Semiconductor trong các ứng dụng của mình. Bằng cách sử dụng thư viện DallasTemperature, người dùng có thể thu thập dữ liệu nhiệt độ chính xác và sử dụng chúng để điều khiển các hệ thống và quá trình theo nhu cầu của dự án.

Tóm lại, thư viện DallasTemperature là một công cụ hữu ích để giao tiếp và điều khiển các cảm biến nhiệt độ 1-Wire của Dallas Semiconductor. Với tính năng đa dạng và dễ sử dụng, thư viện này giúp người dùng tận dụng tối đa tiềm năng của các cảm biến nhiệt độ trong các dự án điện tử và ứng dụng của mình.

## API OpenWeather:

API OpenWeather là một dịch vụ cung cấp thông tin thời tiết từ trang web OpenWeatherMap. Được phát triển để cung cấp dữ liệu thời tiết chính xác và chi tiết cho các ứng dụng và trang web, API OpenWeather cho phép người dùng truy cập và sử dụng thông tin thời tiết từ hàng ngàn địa điểm trên toàn thế giới.

Với API OpenWeather, người dùng có thể lấy thông tin về nhiệt độ, độ ẩm, tốc độ gió, áp suất, mô tả thời tiết, dự báo trong tương lai và nhiều thông tin khác về thời tiết. Dữ liệu được cung cấp bằng cách sử dụng các yêu cầu HTTP và trả về dưới dạng các tệp tin JSON hoặc XML, tùy thuộc vào yêu cầu của người dùng.

API OpenWeather cung cấp các tùy chọn linh hoạt để truy xuất thông tin thời tiết. Người dùng có thể lấy dữ liệu thời tiết cho một địa điểm cụ thể, lấy dữ liệu dự báo theo giờ, ngày hoặc tuần, và thậm chí tìm kiếm thông tin thời tiết theo tọa độ địa lý. Điều này cho phép người dùng tùy chỉnh và lấy thông tin thời tiết theo nhu cầu của dự án hoặc ứng dụng của mình.

API OpenWeather cũng cung cấp các tính năng bổ sung như tra cứu thông tin về mặt trời, mặt trăng và thời tiết trong quá khứ. Người dùng có thể lấy thông tin về thời tiết trong quá khứ hoặc dự báo thời tiết cho các ngày trong tương lai.

Với tính linh hoạt và khả năng tùy chỉnh của nó, API OpenWeather đã trở thành một công cụ phổ biến trong việc tích hợp thông tin thời tiết vào các ứng dụng và trang web. Các nhà phát triển có thể sử dụng API OpenWeather để hiển thị thông tin thời tiết trực quan và hữu ích cho người dùng, cũng như tạo ra các tính năng và dịch vụ dựa trên dữ liệu thời tiết.

Tóm lại, API OpenWeather là một dịch vụ cung cấp thông tin thời tiết chính xác và chi tiết từ OpenWeatherMap. Với khả năng truy xuất thông tin thời tiết linh hoạt và tùy chỉnh, API OpenWeather đã trở thành công cụ quan trọng trong việc tích hợp thông tin thời tiết vào các ứng dụng và trang web, giúp người dùng tiếp cận và sử dụng thông tin thời tiết một cách thuận tiện và hiệu quả.

## Phần cứng:

### 1.Arduino UNO R3

A picture containing electronic component, circuit component, electronic engineering, electronics

Description automatically generated

#### Hình 2.1: Arduino UNO R3

Arduino UNO R3 là một phiên bản nâng cấp của bo mạch phát triển Arduino Uno. Được phát triển bởi Arduino.cc, nó là một trong những bo mạch phổ biến nhất và được sử dụng rộng rãi trong cộng đồng Maker và IoT.

Arduino UNO R3 được thiết kế nhỏ gọn và dễ sử dụng, với các tính năng mạnh mẽ và khả năng mở rộng linh hoạt. Bo mạch này sử dụng vi điều khiển ATmega328P, có tốc độ xung nhịp 16MHz và bộ nhớ Flash 32KB. Nó cung cấp 14 chân GPIO (General Purpose Input/Output) và 6 chân ADC (Analog-to-Digital Converter), cho phép người dùng kết nối và điều khiển nhiều loại cảm biến và thiết bị ngoại vi khác nhau.

Arduino UNO R3 cũng có các tính năng như giao tiếp USB, UART (Universal Asynchronous Receiver/Transmitter) và I2C (Inter-Integrated Circuit), cho phép nó kết nối với máy tính hoặc các thiết bị khác để truyền dữ liệu và lập trình. Nó cũng có một cổng nguồn DC và cổng ICSP (In-Circuit Serial Programming), giúp người dùng nạp chương trình vào vi điều khiển dễ dàng.

Arduino UNO R3 được hỗ trợ bởi một môi trường phát triển tích hợp (IDE) miễn phí và dễ sử dụng, cho phép người dùng viết và tải các chương trình Arduino một cách nhanh chóng. Cộng đồng Arduino cũng rất lớn và sẵn sàng chia sẻ kiến thức và dự án, giúp người dùng tận dụng tối đa tiềm năng của bo mạch.

Với khả năng mở rộng và tính linh hoạt, Arduino UNO R3 là một công cụ lý tưởng cho các dự án điện tử, IoT và robot. Dễ sử dụng và có giá cả phải chăng, nó là lựa chọn hàng đầu cho cả người mới bắt đầu và những người có kinh nghiệm trong lĩnh vực này.

### 2. Cảm biến nhiệt độ DS18B20

A black wire with a silver metal rod

Description automatically generated

#### Hình 2.2: Cảm biến nhiệt độ DS18B20

Cảm biến nhiệt độ DS18B20 là một cảm biến kỹ thuật số chính xác và linh hoạt trong việc đo nhiệt độ. Được sản xuất bởi hãng Maxim Integrated, DS18B20 sử dụng giao tiếp OneWire, cho phép người dùng kết nối nhiều cảm biến trên cùng một dây duy nhất.

DS18B20 có thể đo nhiệt độ trong khoảng từ -55°C đến +125°C với độ chính xác cao và sai số nhỏ. Nó cung cấp độ phân giải 12 bit, cho phép đo nhiệt độ với độ chính xác tới 0.0625°C. Điều này làm cho DS18B20 trở thành một lựa chọn lý tưởng cho các ứng dụng yêu cầu đo nhiệt độ chính xác như trong các hệ thống điều khiển nhiệt độ, quản lý môi trường hoặc các dự án IoT.

Cảm biến DS18B20 có thiết kế nhỏ gọn và dễ dàng tích hợp vào các thiết bị khác nhau. Nó hoạt động với điện áp 3.3V hoặc 5V và có chế độ tiết kiệm năng lượng, giúp tiết kiệm pin và kéo dài thời gian sử dụng.

Cảm biến DS18B20 cung cấp dữ liệu nhiệt độ dưới dạng số, cho phép người dùng dễ dàng đọc và xử lý thông qua vi điều khiển hoặc bo mạch Arduino. Nó cũng có khả năng ghi nhớ địa chỉ duy nhất cho mỗi cảm biến, giúp người dùng dễ dàng phân biệt và quản lý nhiều cảm biến trên cùng một hệ thống.

Với tính linh hoạt và độ chính xác cao, cảm biến nhiệt độ DS18B20 là một công cụ quan trọng trong các ứng dụng đo nhiệt độ. Dễ sử dụng và có giá cả phải chăng, nó phổ biến trong cộng đồng Maker và được sử dụng rộng rãi trong các dự án IoT, hệ thống giám sát và điều khiển, hoặc các ứng dụng yêu cầu đo nhiệt độ chính xác.

### 3. Đèn LED

Several different colored lights

Description automatically generated

#### Hình 2.3: Đèn LED

Đèn LED 5mm là một loại đèn LED phổ biến được sử dụng rộng rãi trong các ứng dụng chiếu sáng, điện tử và IoT. Với kích thước nhỏ gọn và dễ dàng tích hợp, đèn LED 5mm cung cấp ánh sáng mạnh mẽ và độ sáng cao.

Đèn LED 5mm được thiết kế để hoạt động với điện áp từ 1.9V đến 2.1V, và dòng điện từ 10mA đến 20mA. Điều này giúp người dùng có thể dễ dàng kết nối đèn LED với nguồn điện thông thường như pin, bộ nguồn hoặc bo mạch điều khiển.

Với độ sáng từ 6000 MCD đến 8000 MCD, đèn LED 5mm tạo ra ánh sáng rõ ràng và mạnh mẽ. Điều này làm cho nó trở thành một lựa chọn tốt cho các ứng dụng yêu cầu ánh sáng sáng, như đèn chiếu sáng, đèn flash, hay các dự án điện tử sáng tạo.

Đèn LED 5mm có thể được phủ màu, cho phép người dùng lựa chọn các màu sắc khác nhau như đỏ, xanh, vàng, trắng, v.v. Điều này mang lại sự linh hoạt và sáng tạo cho các dự án thiết kế ánh sáng.

Với các thông số kỹ thuật như điện áp, dòng, độ sáng và kích thước nhỏ gọn, đèn LED 5mm là một giải pháp chiếu sáng linh hoạt và hiệu quả. Dễ dàng sử dụng và có giá cả phải chăng, đèn LED 5mm rất phổ biến trong cộng đồng Maker và được sử dụng rộng rãi trong các dự án điện tử, chiếu sáng và IoT.

### 4.Board Test:

A white circuit board with red and blue lines

Description automatically generated

#### Hình 2.4: Board Test

Board test trong IoT là một công cụ quan trọng để kiểm tra và xác định hiệu suất và chất lượng của các thiết bị IoT. Một board test được thiết kế để kiểm tra và đánh giá các thành phần và chức năng của một board điều khiển IoT, bao gồm việc kiểm tra các cảm biến, kết nối mạng, giao tiếp và các chức năng khác.

Board test trong IoT cho phép người dùng kiểm tra các yếu tố quan trọng như độ chính xác, độ ổn định và độ tin cậy của các thành phần và chức năng trong board điều khiển. Nó giúp đảm bảo rằng các thiết bị IoT hoạt động đúng theo đặc tả và đáp ứng các tiêu chuẩn chất lượng và an toàn.

Các board test IoT thường được tích hợp với các chức năng tự động, cho phép người dùng thực hiện các bài kiểm tra và thu thập dữ liệu một cách tự động và hiệu quả. Các kết quả kiểm tra và dữ liệu thu thập được có thể được phân tích và đánh giá để đưa ra các cải tiến và điều chỉnh cho thiết bị IoT.

Board test trong IoT có vai trò quan trọng trong quá trình phát triển và sản xuất các thiết bị IoT. Nó giúp đảm bảo rằng các thiết bị được sản xuất đáp ứng các yêu cầu và tiêu chuẩn chất lượng, đồng thời giúp giảm thiểu rủi ro và chi phí do lỗi trong quá trình sử dụng.

Với sự phát triển nhanh chóng của IoT và sự gia tăng về số lượng và đa dạng của các thiết bị IoT, board test đã trở thành một yếu tố quan trọng trong việc đảm bảo chất lượng và hiệu suất của các thiết bị này. Các công ty và tổ chức trong lĩnh vực IoT đều cần sử dụng board test để đảm bảo sự thành công và đáng tin cậy của các sản phẩm và dịch vụ IoT.

# CHƯƠNG 3 : PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Mô tả các yêu cầu chức năng của hệ thống:

Đối với đề tài “Xây dựng Winform App Điều khiển thiết bị điều hoà nhiệt độ”, hệ thống sẽ bao gồm các chức năng sau:

* Hiển thị thông tin thiết bị: Hệ thống cần hiển thị thông tin về nhiệt độ hiện tại, trạng thái hoạt động của thiết bị điều hoà nhiệt độ.
* Điều khiển nhiệt độ: Hệ thống cho phép người dùng điều chỉnh nhiệt độ của thiết bị điều hoà. Người dùng có thể tăng hoặc giảm nhiệt độ theo ý muốn.
* Chế độ hoạt động: Hệ thống cần hỗ trợ các chế độ hoạt động của thiết bị điều hoà như chế độ auto hoặc manual,…
* Báo cáo: Hệ thống cần cung cấp các báo cáo về nhiệt độ để người dùng có thể kiểm tra và phân tích hoạt động của thiết bị.
* Đăng nhập và quản lý người dùng: Hệ thống cần hỗ trợ chức năng đăng nhập và quản lý người dùng, để người dùng có thể truy cập và quản lý thông tin của mình.
* Giao diện thân thiện và dễ sử dụng: Hệ thống cần có giao diện người dùng thân thiện, dễ sử dụng và có thể tương tác dễ dàng với các thành phần điều khiển nhiệt độ.
* Tích hợp với thiết bị điều khiển: Hệ thống cần có khả năng kết nối và giao tiếp với thiết bị điều khiển nhiệt độ để truyền thông tin và điều khiển thiết bị.

## Mô tả các yêu cầu phi chức năng của hệ thống:

Các yêu cầu phi chức năng của đề tài “Xây dựng Winform App Điều khiển thiết bị điều hoà nhiệt độ” bao gồm:

* Độ tin cậy: Hệ thống cần đảm bảo tính ổn định và tin cậy trong việc điều khiển thiết bị điều hòa nhiệt độ. Điều này đảm bảo rằng ứng dụng luôn hoạt động đúng và không gây ra sự cố không mong muốn.
* Hiệu suất: Hệ thống cần có hiệu suất tốt, đảm bảo thời gian phản hồi nhanh khi người dùng thay đổi cài đặt nhiệt độ hoặc chế độ hoạt động.
* Giao diện người dùng thân thiện: Ứng dụng cần có giao diện người dùng đơn giản và dễ sử dụng, cung cấp các nút và thanh trượt để điều chỉnh nhiệt độ và chế độ của thiết bị điều hòa.
* Bảo mật: Hệ thống cần có các biện pháp bảo mật để đảm bảo rằng chỉ có người dùng được ủy quyền mới có thể truy cập và điều khiển thiết bị điều hòa.
* Tương thích: Ứng dụng cần tương thích với các thiết bị điều hòa nhiệt độ phổ biến trên thị trường, đảm bảo rằng người dùng có thể sử dụng ứng dụng với bất kỳ thiết bị nào mà họ sở hữu.
* Tích hợp: Hệ thống cần có khả năng tích hợp với các hệ thống khác, chẳng hạn như hệ thống quản lý nhà thông minh hoặc hệ thống điều khiển từ xa khác.
* Độ bảo mật và quyền riêng tư: Hệ thống cần đảm bảo an toàn thông tin và quyền riêng tư của người dùng, đảm bảo rằng dữ liệu cá nhân không bị lộ ra ngoài hoặc bị truy cập trái phép.

## Tổng quan chức năng của hệ thống:

Application Winform điều khiển thiết bị điều hoà cho phép người dùng có thể biết được nhiệt độ phòng hiện tại của mình thông qua cảm biến và điều khiển các thiết bị điều hoà thông qua giao diện WinFormApp

* Đăng nhập;
  + - Người dùng phải đăng nhập để có thể sử dụng được các chức năng của phầm mềm
* Thông tin thời tiết:
  + - Người dùng có thể xem được các thông tin của thời tiết như: Thành phố, nhiệt độ, độ ẩm, tốc độ gió,…
* Thiết lập nhiệt độ:
  + - Người dùng phải kết nối với Arduino để có thể nhận và gửi các yêu cầu đến thiết bị
    - Người dùng có thể xem được thông tin nhiệt độ hiện tại thông qua cảm biến
    - Người dùng có thể lựa chọn chế độ để bật tắt các thiết bị
    - Đối với chế độ Auto: Phầm mềm sẽ dựa vào nhiệt độ hiện tại và nhiệt độ lấy từ API OpenWeather để đưa ra kiến nghị
    - Đối với chế độ Manual: Người dùng có thể tuỳ chỉnh nhiệt độ mong muốn

### Usecase tổng quát

A diagram of a diagram

Description automatically generated

#### Hình 3.1: Sơ đồ Usecase tổng quát

Để sử dụng đươc các chức năng của hệ thống thì người dùng cần phải đăng nhập vào hệ thống

### Usecase Xem thông tin nhiệt độ:

A picture containing text, diagram, circle, line

Description automatically generated

#### Hình 3.2: Sơ đồ Usecase xem thông tin nhiệt độ

Sau khi đăng nhập vào hệ thống, người dùng có thể chọn chức năng xem thông nhiệt độ. Hệ thống sẽ gửi yêu cầu lấy thông tin từ cảm biến nhiệt độ để cho biết nhiệt độ thực hiện ở thời điểm hiện tại, bên cạnh đó hệ thống cũng gửi yêu cầu để lấy thông tin thời tiết từ nguồn dữ liêu API OpenWeather.

### Usecase Điều khiển nhiệt độ:

A picture containing text, diagram, circle, line

Description automatically generated

#### Hình 3.3: Sơ đồ Usecase điều khiển nhiệt độ

Sau khi đăng nhập vào hệ thống, người dùng có thể chọn chức năng điều khiển nhiệt độ. Người dùng có thể điều chỉnh nhiệt độ mong muốn bằng cách nhập giá trị mới trên giao diện. ệ thống gửi yêu cầu điều khiển nhiệt độ tới thiết bị điều hoà nhiệt độ. Thiết bị điều hoà nhiệt độ nhận yêu cầu và thực hiện điều chỉnh nhiệt độ theo yêu cầu của người dùng. Hệ thống cập nhật trạng thái hiển thị của thiết bị điều hoà nhiệt độ trên giao diện.

### Sơ đồ phân rã chức năng (BFD):

A picture containing text, screenshot, diagram, font

Description automatically generated

#### Hình 3.4: Sơ đồ phân rã chức năng (BFD)

## Xây dựng cơ sở dữ liệu:

### Bảng cơ sở dữ liệu:

A screenshot of a computer

Description automatically generated

#### Hình 3.5: Ảnh Bảng users

### Bảng mô tả chi tiết cơ sở dữ liệu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu giá trị | Khoá | Ý nghĩa |
| 1 | id | INT | ✓ | Định danh duy nhất cho mỗi người dùng trong hệ thống |
| 2 | username | VARCHAR |  | Lưu trữ tên người dùng của mỗi tài khoản trong hệ thống |
| 3 | password | VARCHAR |  | Lưu trữ mật khẩu của mỗi tài khoản trong hệ thống |

##### Bảng 3.1: Bảng mô tả chi tiết cơ sở dữ liệu

# CHƯƠNG 4 : XÂY DỰNG CHƯƠNG TRÌNH

## Đăng nhập:

A screenshot of a login screen

Description automatically generated

#### Hình 4.1: Giao diện đăng nhập

Để sử dụng được phần mềm. người dùng phải sử dụng tài khoản được cấp để đăng nhập

Nếu người dùng đăng nhập với thông tin không hợp lệ, phần mềm sẽ báo lỗi và yêu cầu người dùng kiểm tra lại UserName và Password

A screenshot of a computer

Description automatically generated with low confidence

#### Hình 4.2: Thông báo thông tin đăng nhập không hợp lệ

Nếu người dùng đăng nhập thành công, phần mềm sẽ hiển thị thông báo chào mừng và chuyển đến giao diện InfoWeather

A screenshot of a computer

Description automatically generated with medium confidence

#### Hình 4.3: Thông báo đăng nhập thành công

## Xem thông tin thời tiết

A screenshot of a computer

Description automatically generated with low confidence

#### Hình 4.2: Giao diện InfoWeather

Ở giao diện InfoWeather, người dùng có thể xem được các thông tin về thời tiết như: Thành phố, nhiệt độ, tốc độ gió, độ ẩm, mây. Các dữ liệu của thông tin thời tiết này được lấy từ API OpenWeaTher.

Nếu người dùng nhấn nút Exit chương trình sẽ đóng lại

Nếu người dùng nhấn nút Next, chương trình sẽ chuyển sang giao diện Setting

## Điều khiển thiết bị điều hoà nhiệt độ

A screenshot of a computer

Description automatically generated with medium confidence

#### Hình 4.5: Giao diện Setting

Ở giao diện Setting, người dùng có thể xem được thông tin nhiệt độ thực tế thông qua cảm biến, xem được thông tin nhiệt độ lấy dữ liệu từ API OpenWeather, chọn chế độ để điều chỉnh nhiệt độ

Nếu người dùng chọn chế độ Auto, phần mềm sẽ lấy giá trị nhiệt độ từ API OpenWeather so với nhiệt độ thực tế thông qua cảm biến rồi đưa ra đề nghị điều chỉnh nhiệt độ

A screenshot of a computer

Description automatically generated with medium confidence

#### Hình 4.6: Phần mền đưa ra khuyến chị điều chỉnh nhiệt độ

Nếu người dùng chọn chế độ Manual, phần mềm sẽ kích hoạt ô “Tuỳ chỉnh nhiệt độ”, người dùng sẽ nhập nhiệt độ mong muốn vào ô nhiệt độ. Nếu nhiệt độ người dùng mong muốn thấp hơn nhiệt độ hiện tại của cảm biến thì thiết bị sẽ hoạt động, đèn trạng thái trong ô Thiết bị sẽ chuyển sang màu xanh. Nếu nhiệt độ người dùng mong muốn cao hơn nhiệt độ hiện tại của cảm biến thì thiết bị sẽ hoạt động, đèn trạng thái trong ô Thiết bị sẽ chuyển sang màu đỏ.

A picture containing text, screenshot, font, rectangle

Description automatically generated

#### Hình 4.7: Trạng thái thiết bị hoạt động

A close up of wires and wires

Description automatically generated

#### Hình 4.8: Thiết bị đang hoạt động (mô phỏng bằng đèn LED)

A screenshot of a computer

Description automatically generated with low confidence

#### Hình 4.9: Trạng thái thiết bị tắt

A picture containing electronics, electrical wiring, electronic engineering, cable

Description automatically generated

#### Hình 4.10: Thiết bị đang tắt (mô phỏng bằng đèn LED)

# CHƯƠNG 5: KẾT LUẬN

## Kết quả đạt được

Sau quá trình thực hiện dưới sự hướng dẫn của quý thầy cô bộ môn, phần mềm Winform Điều khiển thiết bị điều hoà nhiệt độ đã đạt được các tính năng cơ bản như sau:

* Thực hiện lấy được thông tin thời tiết từ API OpenWeather như: Thành phố, nhiệt độ, tốc độ gió, độ ẩm,…
* Ứng dụng lập trình nhúng với Arduino UNO R3 vào Project
* Thực hiện được các thao tác gửi và nhận giá trị, dữ liệu từ cảm biển thông qua Arduino với giao diện của Winform
* Xây dựng được giao diện phần mềm khá trực quan
* Hệ thống hoạt động ổn định, dữ liệu từ hệ thống đáng tin cậy, độ chính xác cao

## Hạn chế còn tồn tại của hệ thống

* Các chức năng của phần mềm chưa đầy đủ để ứng dụng vào thực tế:
* Chưa áp dụng được công nghệ IoT điều khiển qua wifi hoặc mạng dữ liệu
* Cơ sở dữ liệu còn nhỏ
* Chức năng Auto chưa được hoàn chỉnh để tối ưu việc gợi ý và điều chỉnh nhiệt độ cho người dùng
* Chức năng ghi lại các lần thay đổi nhiệt độ vào log chưa thực hiện được
* Chưa điều khiển thiết bị tự động theo thời gian thực

## Hướng phát triển hệ thống trong tương lai

* Tìm hiểu, trau đồi kiến thức để hoàn thiện những chức năng chưa hoàn thiện
* Xây dựng thêm giao diện để trực quan hơn
* Tìm hiểu công nghệ wifi để áp dụng
* Xây dựng cơ sở dữ liệu hoàn thiện hơn
* Tìm hiểu thuật toán để chế độ Auto gọi ý tối ưu việc điều chỉnh nhiệt độ cho người dùng
* Tìm hiểu về log để ứng dụng vào việc ghi lại lịch sử thay đổi nhiệt độ
* Tìm hiểu thêm về Timer để liên tục cập nhật nhiệt độ theo thời gian thực

# TÀI LIỆU THAM KHẢO

[1] Nạp code vào Arduino UNO R3: <http://arduino.vn/bai-viet/402-huong-dan-nap-chuong-trinh-don-gian-cho-arduino-uno-r3>

[2] Sử dụng API OpenWeather và các thông số thời tiết cần sử dụng trong bài: <https://openweathermap.org/api>

[3] Hướng dẫn sử dụng chi tiết API OpenWeather: <https://youtu.be/BG6stfOqhd8>

[4] Kiến thức cơ bản về cổng kết nối SerialPort: <https://mesidas.com/serial-port/#:~:text=Serial%20Port%20(hay%20c%E1%BB%95ng%20n%E1%BB%91i,s%E1%BB%AD%20d%E1%BB%A5ng%20c%E1%BB%95ng%20n%E1%BB%91i%20ti%E1%BA%BFp>

[5] Thư viện OneWire: <https://arduino.esp8266.vn/projects/onewire.html>