1.1 Giới thiệu tổng quan về IOT

Thuật ngữ Internet of things (viết tắt là IoT) được hiểu một cách đơn giản là một mạng lưới vạn vật kết nối với nhau thông qua Internet. Chúng bao gồm các đồ vật, con người được cung cấp một định danh của riêng mình và tất cả có khả năng truyền tải hay trao đổi thông tin hay dữ liệu qua một mạng duy nhất mà không cần đến sự tương tác trực tiếp giữa người với người, hay người với máy tính.

IoT đã phát triển từ sự hội tụ của công nghệ không dây, công nghệ vi cơ điện tử và quan trọng hơn là sự có mặt của Internet. Nói đơn giản IoT là một tập hợp các thiết bị có khả năng kết nối mọi thứ lại với nhau với Internet và với thế giới bên ngoài để thực hiện một công việc nào đó.

Có thể nói IoT có mặt trong hầu hết các lĩnh vực đời sống của chúng ta hiện nay từ văn hóa, du lịch, giáo dục, y học, truyền thông,...Tất cả các lĩnh vực trên đều được ứng dụng từ các thành tựu của IoT cả. Như trước đây nếu chúng ta muốn thông tin liên lạc với nhau thì không còn cách nào khác đó là viết thư và truyền thư thông qua người giao nhận. Còn bây giờ thì thông qua các thiết bị thông minh, chúng được kết nối với nhau thông qua hệ thống wifi, 3G hay các ứng dụng riêng của từng hãng. Chưa đầy 10s chúng ta đã có thể gửi tin nhắn đến một người hay nhiều người khác. Bên cạnh đó thì còn có nhiều ứng dụng khác như:

* Quản lý môi trường Quản lý hệ thống máy móc
* Hệ thống mua sắm trực tuyến
* Hệ thống kiểm soát an ninh
* Nhà thông minh
* Ứng dụng quản lý toàn bộ thiết bị cá nhân thông qua việc động bộ

1.2 Giới thiệu hệ thống

Hệ thống giàn phơi/mái che thông minh mini là một mô hình mô phỏng việc kéo ra/thu vào giàn phơi ở ban công căn hộ, tập trung vào “mưa–nắng + an toàn hành trình”. Người dùng có thể bấm nút trên app/web (MQTT qua Wi-Fi nhà) để Mở/Đóng/Stop thủ công, hoặc có thể để thiết bị tự phân tích và đưa ra lịch đóng mở phù hợp, dựa theo thời tiết hiện tại.

1.3 Mục đích nghiên cứu

Mục tiêu đề tài:

* Giải bài toán “phơi tự động” cơ bản với chi phí thấp: cảm biến mưa–nắng → quyết định → điều khiển motor có giới hạn hành trình.
* Đánh giá độ tin cậy trong bối cảnh thực: tránh “đóng–mở liên tục” khi mưa/nắng chập chờn (dùng trễ xác nhận mưa vài giây, ngưỡng lux đủ nắng).
* Thử nghiệm điều khiển từ xa qua Wi-Fi + MQTT: gửi lệnh, nhận trạng thái tức thời; làm nền cho việc thêm lịch hẹn hoặc thông báo sau này.
* Giao diện người dùng thân thiện: Người dùng có thể theo dõi và điều khiển hệ thống từ xa qua ứng dụng di động hoặc website.

Đối tượng sử dụng:

Hệ thống được sử dụng cho các tòa nhà hay các hộ gia đình có không gian khiêm tốn

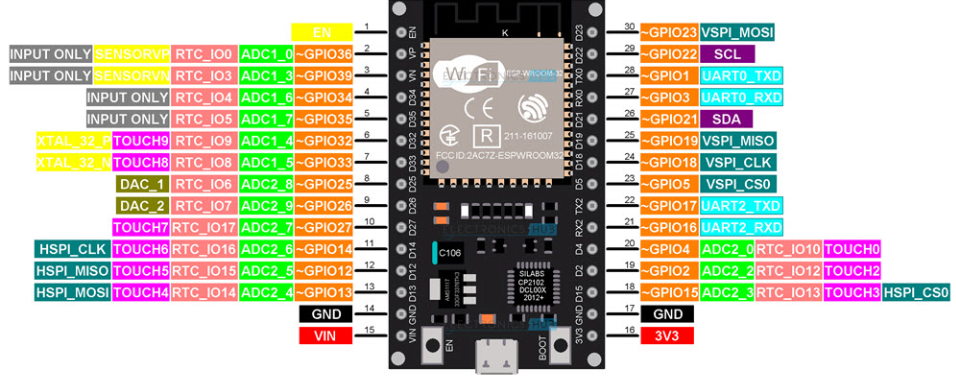
Lý thuyết:

* Hệ thống mô phỏng thao tác kéo ra/thu vào giàn phơi
* Hệ thống đọc cảm biến mưa (raindrop) và cảm biến ánh sáng BH1750 để quyết định trạng thái vận hành (mở khi nắng đủ, thu khi mưa)
* Thiết bị giao tiếp MQTT qua Wi-Fi để nhận lệnh từ xa (Open/Close/Stop), lưu trữ lịch sử đóng/mở và trạng thái/telemetry lên nền tảng đám mây, sau đó gửi về về dashboard/app.

1.4 Thiết bị phần cứng và nền tảng phần mềm

1.4.1 Các thiết bị phần cứng

a. Bộ kit ESP32 CP3102



ESP32 là SoC Wi-Fi 2.4 GHz + Bluetooth v4.2 (BR/EDR + BLE) tích hợp MCU Xtensa LX6 hai nhân (lên đến 240 MHz), SRAM ~520 KB, dùng module ESP32-WROOM-32(E/UE) gắn trên bo.

CP2102 là chip USB-to-UART của Silicon Labs, biến cổng USB máy tính thành cổng UART để nạp firmware và log serial. Trên DevKit, CP2102 còn nối DTR/RTS → EN/IO0 để tự vào chế độ bootloader khi nạp.

Thông số chính:

* CPU: Xtensa LX6 2 nhân, 80–240 MHz.
* Bộ nhớ: ~520 KB SRAM trong chip; Flash ngoài (thường 4 MB) trong module.
* Kết nối: Wi-Fi 802.11 b/g/n 2.4 GHz; Bluetooth BR/EDR + BLE v4.2.
* GPIO: tối đa 34 GPIO mức 3.3 V (một số chỉ input).
* Analog/TouchS: ADC 12-bit (tối đa 18 kênh), DAC 8-bit (GPIO25, 26), 10 kênh cảm ứng chạm.
* Peripherals: UART, SPI (VSPI/HSPI), I²C, I²S, PWM (LEDC), SD/SDIO, CAN (TWAI).
* Nguồn vào: USB/VIN 5 V; I/O chỉ 3.3 V (không 5 V trực tiếp vào GPIO).

Khái quát các chân của kit

* **Nguồn:** VIN(5V), 3V3, nhiều GND.
* **UART:** TX0/RX0 (nối CP2102 để nạp & log), có thêm UART1, UART2 remap linh hoạt.
* **I²C:** thường dùng GPIO21 (SDA) và GPIO22 (SCL) (có thể remap).
* SPI (VSPI mặc định): SCK(GPIO18), MOSI(GPIO23), MISO(GPIO19), CS(GPIO5).
* **ADC:** nhóm GPIO32–39…; GPIO34–39 chỉ input.
* **DAC:** GPIO25, GPIO26.
* **Touch:** ví dụ GPIO4, 12–15, 27, 32, 33…
* **Boot/Reset:** EN (reset), IO0 (giữ LOW khi reset để vào bootloader).
* GPIO6–11 nối flash QSPI trong module, không dùng làm GPIO.

Nguyên lý hoạt động

* **Cấp nguồn & nhận cổng COM:** cắm USB → CP2102 xuất hiện trên máy tính như cổng Serial.
* **Nạp firmware:** công cụ nạp (Arduino IDE/ESP-IDF esptool) kéo DTR/RTS để đặt IO0=LOW rồi reset EN → ESP32 vào UART bootloader → ghi flash → tự reset chạy ứng dụng.
* **Chạy ứng dụng:** lõi CPU thực thi, Wi-Fi/BLE hoạt động song song; ngoại vi (UART/I²C/SPI/ADC/DAC/PWM…) gán ra GPIO; hỗ trợ sleep/deep-sleep đánh thức bằng timer/GPIO.
* **Giao tiếp với các thiết bị ngoại vi:** Bộ kit hỗ trợ nhiều giao thức giao tiếp như UART, I2C và SPI. Các chân GPIO (General Purpose Input/Out- put) có thể được lập trình để đọc dữ liệu từ cảm biến hoặc điều khiển thiết bị khác. Ví dụ, chân ADCO có thể đọc tín hiệu analog từ cảm biến và chuyển đổi nó thành tín hiệu số để xử lý.
* **Xử lý dữ liệu:** Dữ liệu nhận được từ các cảm biến hoặc thiết bị ngoại vi được vi điều khiển xử lý thông qua mã lập trình đã nạp vào bộ nhớ Flash. Bộ kit có khả năng thực hiện các phép toán logic và quyết định dựa trên dữ liệu nhận được.
* **Gửi và nhận dữ liệu:** Sau khi xử lý, BỘ kit có thể gửi dữ liệu lên máy chủ hoặc nhận dữ liệu từ máy chủ thông qua giao thức HTTP, MQTT hoặc WebSocket. Điều này cho phép bộ kit hoạt động trong các ứng dụng IoT như giám sát, điều khiển từ xa và thu thập dữ liệu.
* **Tương tác người dùng:** Bộ kit có thể được kết nối với các thiết bị như màn hình LCD, cảm biến và động cơ, cho phép người dùng tương tác với hệ thống thông qua các ứng dụng di động hoặc web.

b. động cơ DC giảm tốc

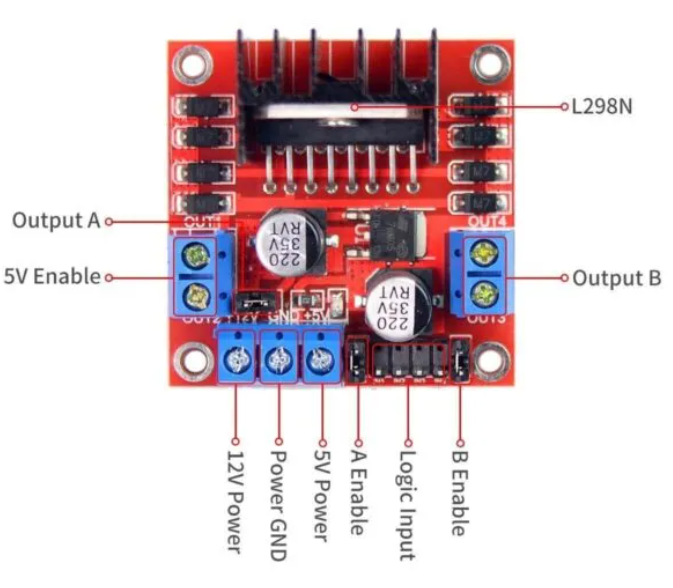
Là motor DC nhỏ có hộp số, lực tốt ở điện áp 6–9 V, phù hợp kéo/thu mô hình. (trong thực tế nên sử dụng động cơ có công suất lớn hơn)



Thông số kỹ thuật

* Điện áp danh định: 3–9 V
* Tốc độ không tải: phụ thuộc tỷ số truyền (ví dụ 30–300 rpm).
* Dòng không tải: 100–250 mA; dòng kẹt/stall: 0.8–1.5 A (tuỳ loại).
* Trục: đường kính 3–5 mm (gắn puly/ống cuộn).
* Cấp điện 9V qua module L298N

c. Module L298N



Là driver cầu H 2 kênh cho motor DC, nhận tín hiệu điều khiển từ ESP32.

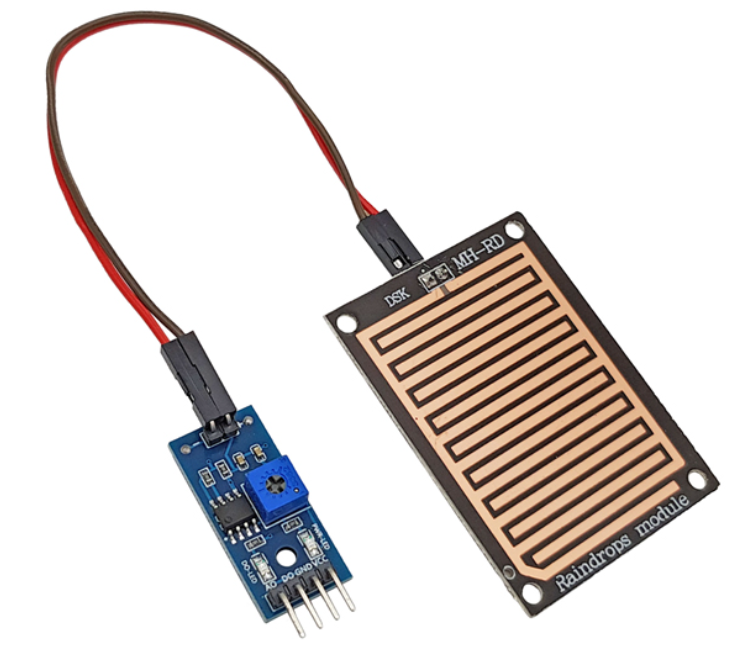
Thông số chính

* Điện áp phần công suất (VS): 5–35 V
* Dòng: tối đa ~2 A/kênh (đỉnh ngắn; thực tế nên ≤1–1.5 A liên tục, có tản nhiệt).
* Điện áp logic (Vss):5 V (lấy từ on-board regulator hoặc nguồn 5 V riêng).
* Sụt áp lớn (~1.5–2 V) → motor 9 V sẽ nhận ~7–7.5 V khi tải.

Cấp nguồn

* VS: từ buck 9 V; GND chung với ESP32.
* 5 V logic: dùng buck 5 V cấp cho ESP32; không dùng 5 V của L298N nuôi ESP32.

d. Cảm biến mưa



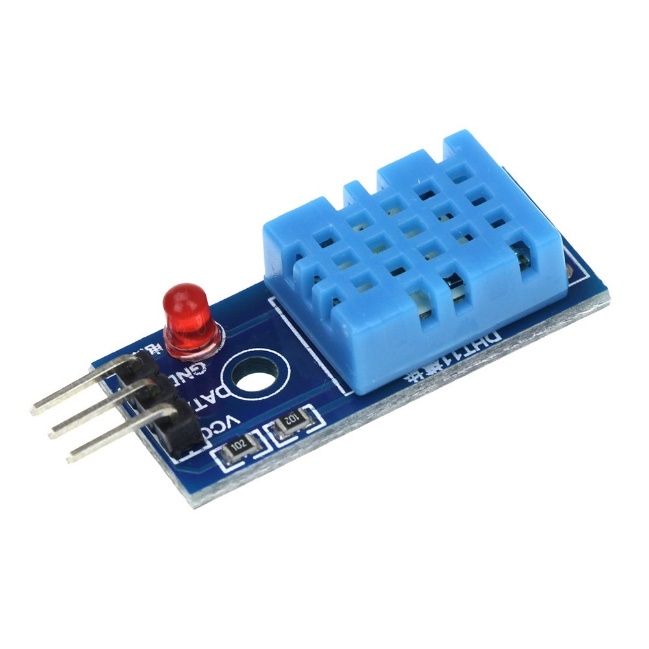
Là tấm dò mưa + board so sánh (LM393), xuất D0 (digital) theo ngưỡng chỉnh biến trở, và A0 (analog).

Thông số chính

* Điện áp hoạt động: 3.3–5 V (nhiều bản ghi 5 V, nhưng vẫn chạy 3.3 V).
* Ngõ ra D0: mức logic theo ngưỡng; A0: điện áp tỷ lệ ướt/khô.

Cấp nguồn: dùng 3.3 V để an toàn mức logic với ESP32, hoặc 5 V nhưng chỉ dùng D0 nếu bảo đảm tương thích 3.3 V

e. DHT11

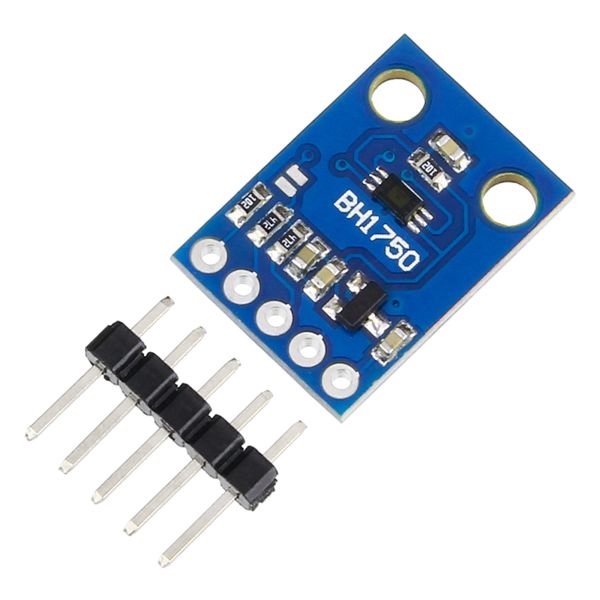


Là cảm biến nhiệt độ/độ ẩm giá rẻ để hiển thị môi trường (phục vụ thống kê/DryScore đơn giản).

Thông số chính

* Điện áp: 3.3–5.5 V.
* Nhiệt độ: 0–50 °C, độ chính xác khoảng ±2 °C.
* Độ ẩm: 20–80 %RH, độ chính xác khoảng ±5 %RH.
* Tần suất đo: ~1 Hz (1 lần/giây).
* Cấp nguồn 3.3 V (hoặc 5 V); GND chung.

f. BH1750



Là cảm biến ánh sáng đo **lux** trực tiếp, dùng để biết khi nào “đủ nắng” để có thể mở/đóng

Thông số chính

* Dải đo: ~1–65,535 lux (nhiều mode).
* I²C địa chỉ mặc định: 0x23 (0x5C tùy module).
* Điện áp: 3.3–5 V; logic I²C 3.3 V-tương thích.
* Cấp nguồn 3.3 V (hoặc 5 V); GND chung.

g. các linh kiện khác:

- Nguồn DC 12V: nguồn cấp chính cho thiết bị

- Module hạ áp Lm2596: Dùng để hạ áp từ 12V về 9V (nuôi motor) và 5V (cho ESP32)

- Breadboard, dây dupont...

1.4.2 Các nền tảng phần mềm

a. Arduino IDE

Arduino IDE là một phần mềm soạn thảo văn bản chính hãng, giúp viết code để nạp vào bo mạch Arduino một cách nhanh chóng, dễ dàng và hoàn toàn miễn phí.Tuy là phần mềm mã nguồn mở nhưng khả năng bảo mật thông tin của Arduino IDE là vô cùng tuyệt vời, khi phát hiện lỗi nhà phát hành sẽ vá nó và cập nhật rất nhanh khiến thông tin của người dùng không bị mất hoặc rò rỉ ra bên ngoài.

Khi người dùng viết mã và biên dịch, IDE sẽ tạo file Hex cho mã. File Hex là các file thập phân Hexa được Arduino hiểu và gửi đến bo mạch bằng cáp USB. Mỗi bo Arduino đều được tích hợp một bộ vi điều khiển, bộ vi điều khiển sẽ nhận file Hex và chạy theo mã được viết.

Arduino IDE sử dụng ngôn ngữ lập trình C/C++ rất phổ biến trong giới lập trình. Bất kỳ đoạn code nào của C/C++ thì Arduino IDE đều có thể nhận dạng, giúp các lập trình viên thuận tiện trong việc thiết kế chương trình lập cho các bo mạch Arduino.

Arduino IDE tích hợp với hơn 700 thư viện, được viết và chia sẻ bởi nhà phát hành Arduino Software và thành viên trong cộng đồng Arduino.

Arduino IDE có một giao diện đơn giản, dễ sử dụng giúp người dùng thuận tiện hơn trong thao tác. Dưới đây là một số tính năng chúng ta thường sử dụng:

* Nút kiểm tra chương trình (Verify): giúp dò lỗi phần code định truyền xuống bo mạch Arduino.
* Nút tải đoạn code vào bo mạch Arduino (Upload): giúp nhập đoạn code vào bo mạch Arduino.
* Vùng lập trình: người dùng sẽ viết chương trình tại khu vực này.
* Thanh Menu: gồm những thẻ chức năng nằm trên cùng như File, Edit, Sketch, Tools, Help

b. Firebase

Firebase là nền tảng phát triển ứng dụng di động của Google giúp chúng ta xây dựng, cải thiện và phát triển ứng dụng của mình. Nó có nhiều dịch vụ được sử dụng để quản lý dữ liệu từ bất kỳ ứng dụng Android, iOS hoặc web nào. Firebase cung cấp các dịch vụ:

* Xác thực: đăng nhập / đăng xuất và nhận dạng người dùng.
* Cơ sở dữ liệu thời gian thực: cơ sở dữ liệu NoSQL theo thời gian thực, được lưu trữ trên đám mây, dữ liệu được lưu trữ dưới dạng cấu trúc JSON;
* Cloud Firestore: cơ sở dữ liệu NoSQL theo thời gian thực, được lưu trữ trên đám mây, dữ liệu được dưới dạng "document";
* Lưu trữ đám mây: lưu trữ tệp có thể mở rộng để tải lên và tải xuống tệp
* Chức năng đám mây: chương trình phụ trợ hướng sự kiện, không có máy chủ; • Firebase hosting: lưu trữ web toàn cầu có hoặc không có tên miền tùy chỉnh

c. React:

React là một thư viện JavaScript phổ biến do Facebook phát triển, dùng để xây dựng giao diện người dùng (UI). React cho phép phát triển các thành phần giao diện độc lập, có thể tái sử dụng, giúp đơn giản hóa quá trình quản lý và cập nhật trạng thái của ứng dụng một cách hiệu quả. Với cơ chế "Virtual DOM", React giúp tăng tốc độ render và giảm tải cho trình duyệt, giúp ứng dụng trở nên mượt mà hơn. React cũng có một hệ sinh thái phong phú, hỗ trợ cả cho phát triển frontend và server-side rendering.

d. Flutter

Flutter là bộ SDK giao diện đa nền tảng do Google phát triển, dùng ngôn ngữ Dart. Điểm mạnh của Flutter:

* UI nhất quán, hiệu năng cao (render thẳng bằng engine Skia, không phụ thuộc native UI).
* Hot reload giúp dev nhanh.
* Một codebase cho Android, iOS, Web, Desktop (Windows/macOS/Linux).
* Hệ sinh thái plugin phong phú, dễ gắn với API/h phần cứng qua Platform Channels.

Ứng dụng IoT thường cần giao diện đẹp + realtime (dashboard, biểu đồ, điều khiển), kết nối đa giao thức (BLE, Wi-Fi, MQTT/WebSocket, HTTP), chạy trên nhiều thiết bị (điện thoại, tablet, web kiosk). Flutter đáp ứng tốt cả 3, lại có nhiều package sẵn sàng dùng.

1.4.3 Luồng hoạt động cơ bản

a. Khởi tạo & kết nối

Cấp nguồn: ESP32 khởi động, kiểm tra limit switch & I/O.

Kết nối Wi-Fi: vào STA, kết nối SSID đã lưu (hoặc mở cổng cấu hình nếu chưa có).

Kết nối MQTT: kết nối broker, đăng ký cmd/<deviceId>, gửi ack/<deviceId> = online + thông tin phiên bản.

b. Vòng lặp cảm biến → quyết định → chấp hành

Đọc cảm biến:

* BH1750 (lux) lấy trung bình N mẫu.
* Raindrop (D0) với bộ đếm mưa liên tục ≥ T xác nhận (mặc định 3 giây).
* DHT11: nhiệt độ/độ ẩm (phục vụ hiển thị/ghi log).

Ra quyết định (FSM):

* Nếu mưa xác nhận → trạng thái CLOSING (thu).
* Nếu không mưa & lux > ngưỡng nắng → OPENING (mở).
* Timeout hành trình (ví dụ 15s) → STOP + phát event TIMEOUT.

Điều khiển motor:

* Gửi PWM qua ENA (GPIO25), chiều quay bằng IN1/IN2; dừng mềm/hard stop khi cần.

Phát dữ liệu:

* Telemetry định kỳ (5–10s): telemetry/<deviceId> chứa {lux,rain,temp,rh,state,ts}.
* Event tức thời (limit, raining, timeout) lên event/<deviceId>.

3. Điều khiển từ app/web

* Người dùng nhấn Open/Close/Stop → app publish cmd/<deviceId> = OPEN|CLOSE|STOP.
* ESP32 nhận lệnh, ACK lên ack/<deviceId> (ví dụ OPENING), đồng thời cập nhật trạng thái cho dashboard.