

Hệ thống chăm sóc vườn dâu tằm tự động
(Tài liệu hướng dẫn)

Mục Lục

1. Giới thiệu.....	2
2. Yêu cầu hệ thống	2
3. Cấu trúc đề tài	3
4. Hướng dẫn cài đặt và khởi chạy code	6
5. Cấu hình	7
6. Lưu đồ và sơ đồ	9
7. Ghi chú phát triển	16
8. Tham khảo	16
PHỤ LỤC	18

1. Giới thiệu

- **Mô tả đề tài:**

- Đề tài được thực hiện nhằm mục đích tạo ra một hệ thống chăm sóc vườn dâu tằm tự động, giảm thiểu cách chăm sóc thủ công và truyền thống.
- Đề tài cần giải quyết một số vấn đề như là có thể quan sát tình trạng của vườn từ xa, hệ thống thực hiện một số chức năng cơ bản một cách tự động.
- Hệ thống có một số chức năng chính như, truyền và nhận dữ liệu liên tục thông qua MCU ↔ Webserver cloud, giao tiếp với các sensor và điều khiển module một cách tự động và thủ công chỉ trên một màn hình cảm ứng.

- **Công nghệ sử dụng:**

- Ngôn ngữ lập trình: C/C++.
- Phần cứng sử dụng: ESP32, Motor, DHT11, PH-4602C, LDR, LED, TFT 4", ACS712.
- Phương pháp lập trình: OOP, RTOS.
- Công nghệ sử dụng: Google Firebase.

- **Tác giả:**

- Phạm Hữu Nghĩa (Tác giả), Nguyễn Xuân Hải (Đồng thực hiện).
- Thông tin liên hệ:
 - Email: huunghia250102@gmail.com.
 - SĐT: 0817819589

2. Yêu cầu hệ thống

- Yêu cầu phần cứng:

- Phần cứng sử dụng: ESP32, Motor, DHT11, PH-4602C, LDR, LED, TFT 4", ACS712.

- Yêu cầu phần mềm:
 - Các phần mềm cần thiết: `ArduinoIDE`, `EspTool`, `Sublime Text`.
 - Các thư viện cần thiết: `TFT_eSPI.h`, `Firestore.h`, `DHT.h`, `WiFi.h`, `LightSensor.h*`, `WaterSensor.h*`, `my_images.h*`, `pHSensor.h*`, `ACS712_ESP32.h*`, `ArduinoJSON.h`, `esp_task_wdt.h`.
- **Ghi chú:** Các thư viện có đánh `*` là các thư viện do đã tự viết, không cần phải tải về từ các nguồn khác.

3. Cấu trúc đề tài

```
Mulberry-Garden-Care-System/
├── doc/
│   ├── TaiLieu.docx
│   └── TaiLieu.pdf
├── gui/
│   ├── Icon/
│   │   └── *.png
│   └── Icon_Hex/
│       └── *.txt
├── lib/
│   ├── ACS712_ESP32/
│   │   ├── ACS712_ESP32.h
│   │   └── ACS712_ESP32.c
│   ├── LightSensor/
│   │   ├── LightSensor.h
│   │   └── LightSensor.c
│   ├── my_images/
│   │   └── my_images.h
│   ├── pHSensor/
│   │   └── pHSensor.h
```

```
|   |   └─ pHSensor.c
|   └─ touch/
|   |   └─ touch.h
|   └─ Water_Sensor/
|   |   └─ Water_Sensor.h
|   |   └─ Water_Sensor.c
└─ src/
    └─ main/
        └─ main.ino
    └─ web/
        └─ img/
            └─ *.png
        └─ firebase.js
        └─ index.html
        └─ script.js
        └─ styles.css
        └─ Web_React.code-workspace
└─ test/
└─ tool/
    └─ img2hex/
        └─ build/
        └─ dist/
        └─ Test/
        └─ img2hex.exe
        └─ img2hex.py
        └─ img2hex.spec
└─ LICENSE
```

- **Thư mục src/ - Mã nguồn chính**

Thư mục này chứa toàn bộ mã nguồn cốt lõi của đề tài.

- **main/:**

- Chứa tệp **main.ino**, là chương trình chính của đề tài. Đây là nơi triển khai các logic điều khiển, bao gồm khởi tạo các cảm biến, đọc dữ liệu, giao tiếp với các thiết bị, và thực hiện các chức năng chính của hệ thống.

- **Thư mục lib/ - Thư viện**

Thư mục này lưu trữ các thư viện riêng tự tạo và sử dụng trong đề tài.

- **ACS712_ESP32:**

- **ACS712_ESP32.h**: Tệp khai báo lớp ACS712, cung cấp các phương thức để đọc dữ liệu từ cảm biến dòng điện.
- **ACS712_ESP32.cpp**: Triển khai chi tiết các phương thức như đọc giá trị Ampere hoặc khởi tạo cảm biến.

- **LightSensor/:**

- **LightSensor.h**: Tệp định nghĩa lớp LightSensor và các phương thức như `begin()` (khởi tạo) và `isLightDetected()` (kiểm tra ánh sáng).
- **LightSensor.cpp**: Chứa phần triển khai thực tế của các phương thức đã được khai báo trong LightSensor.h.

- **my_images/:**

- **my_images.h**: Khai báo các mảng chứa mã hex để đẩy dữ liệu vào màn hình TFT.

- **pHSensor/:**

- **pHSensor.h**: Tệp định nghĩa lớp pHSensor.
- **pHSensor.cpp**: Chứa phần triển khai thực tế của các phương thức đã được khai báo trong pHSensor.

- **touch/:**

- **touch.h**: là một header để định nghĩa và cấu hình màn hình TFT như kích thước màn hình, độ xoay màn hình, và các hàm cấu hình khởi tạo khác như `touch_init()`.
- **WaterSensor/**:
 - **WaterSensor.h**: Tập định nghĩa lớp Water và các phương thức như `begin()` (khởi tạo) và `getWaterLevelCm()` (xuất ra mực nước đơn vị cm).
 - **WaterSensor.cpp**: Chứa phần triển khai thực tế của các phương thức đã được khai báo trong `WaterSensor.h`.
- **Thư mục doc/ - Tài liệu**

Thư mục này lưu trữ các tài liệu liên quan đến đề tài, bao gồm báo cáo và tài liệu hướng dẫn kỹ thuật.

 - **TaiLieu.docx và .pdf**:
 - Đây là tài liệu tham khảo, có thể bao gồm:
 - Mô tả chi tiết cấu trúc đề tài
 - Hướng dẫn sử dụng các thư viện.
 - Các sơ đồ kết nối.
 - **LICENSE**: Chứa thông tin bản quyền của đề tài (MIT License).

4. Hướng dẫn cài đặt và khởi chạy code

- Bước 1: Tiến hành clone dự án về máy bằng lệnh
`clone https://github.com/Jerguel02/Mulberry-Garden-Care-System.git` về máy
- Bước 2: Tiến hành giải nén và đưa các thư viện từ đường dẫn `Mulberry-Garden-Care-System/lib` vào nơi chứa thư viện của Arduino IDE hoặc để các file header và source tương ứng vào ngang hàng với file `Mulberry-Garden-Care-System/src/main/main.ino`.

- Bước 3: Tiến hành vào Arduino IDE và tải các thư viện còn lại về thông qua Arduino IDE hoặc thủ công từ Github.
- Bước 4: Tiến hành nạp code vào ESP32 và khởi chạy.

5. Cấu hình

- Cấu hình phần cứng:
 - Tiến hành kết nối phần cứng như bên dưới:

CTP_SDA	←-----→	G22
CTP_RTS	←-----→	G19
CTP_SCL	←-----→	G0
MISO	←-----→	G12
MOSI	←-----→	G13
SCK	←-----→	G14
LCD_SCL	←-----→	G22
LCD_RS	←-----→	G2
LCD_RST	←-----→	G27
LCD_CS	←-----→	G15
LCD_RS	←-----→	G2
LDR DIGITAL PIN	←-----→	G5
PH SENSOR (VN)	←-----→	G39
WATER SENSOR	←-----→	G35
DHT11	←-----→	G4
LCD_RS	←-----→	G2
PUMP PIN	←-----→	G17
MIST SPRAY PIN	←-----→	G23

LED PIN	←-----→	G16
ACS PUMP	←-----→	G32
ACS MIST SPRAY	←-----→	G34
ACS LED	←-----→	G33

- Cấu hình phần mềm:
 - Chỉnh sửa cấu hình Wifi và Google Sheet API cần thiết trong main.ino

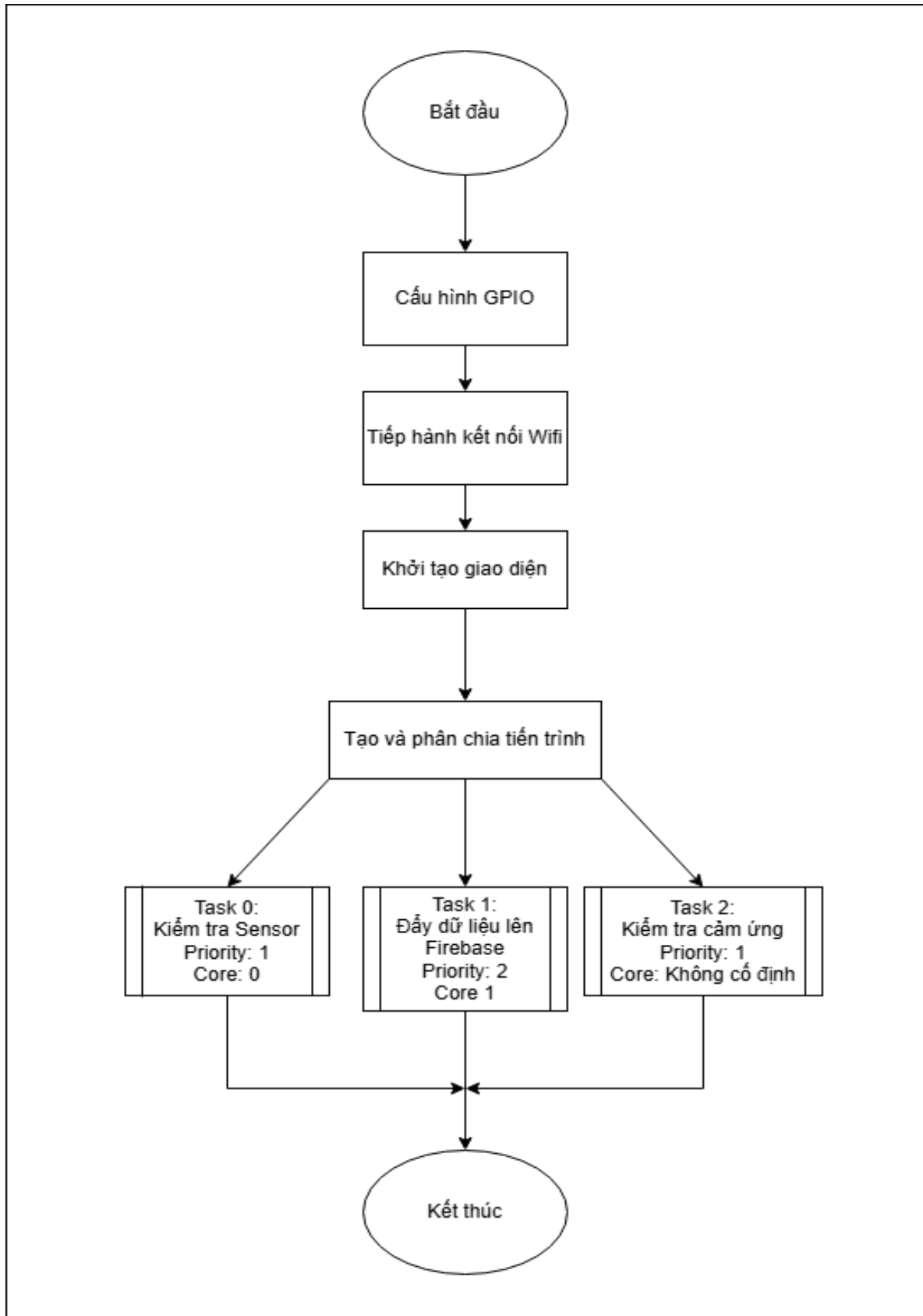
```

11  string WIFI_SSID = "YourSSID"
12  string WIFI_PASSWORD = "YourPassword"
50  #define DATABASE_URL "YourDatabaseURL"

```

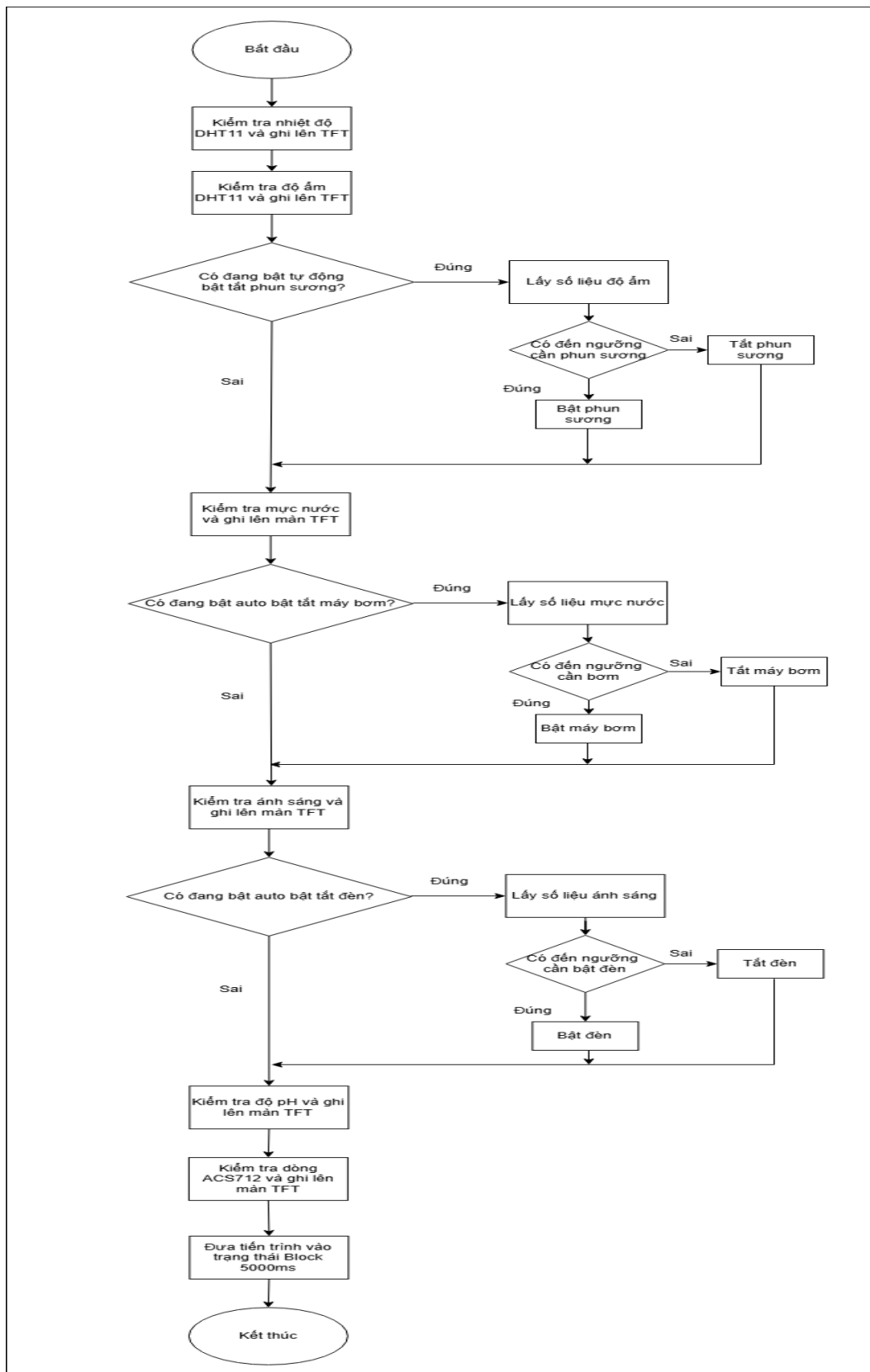
- Với
 - WIFI_SSID là tên wifi cần kết nối
 - WIFI_PASSWORD là mật khẩu của wifi đó
 - DATABASE_URL là liên kết đến Realtime Database của Firebase

6. Lưu đồ và sơ đồ



Hình 6.1. Lưu đồ tổng thể

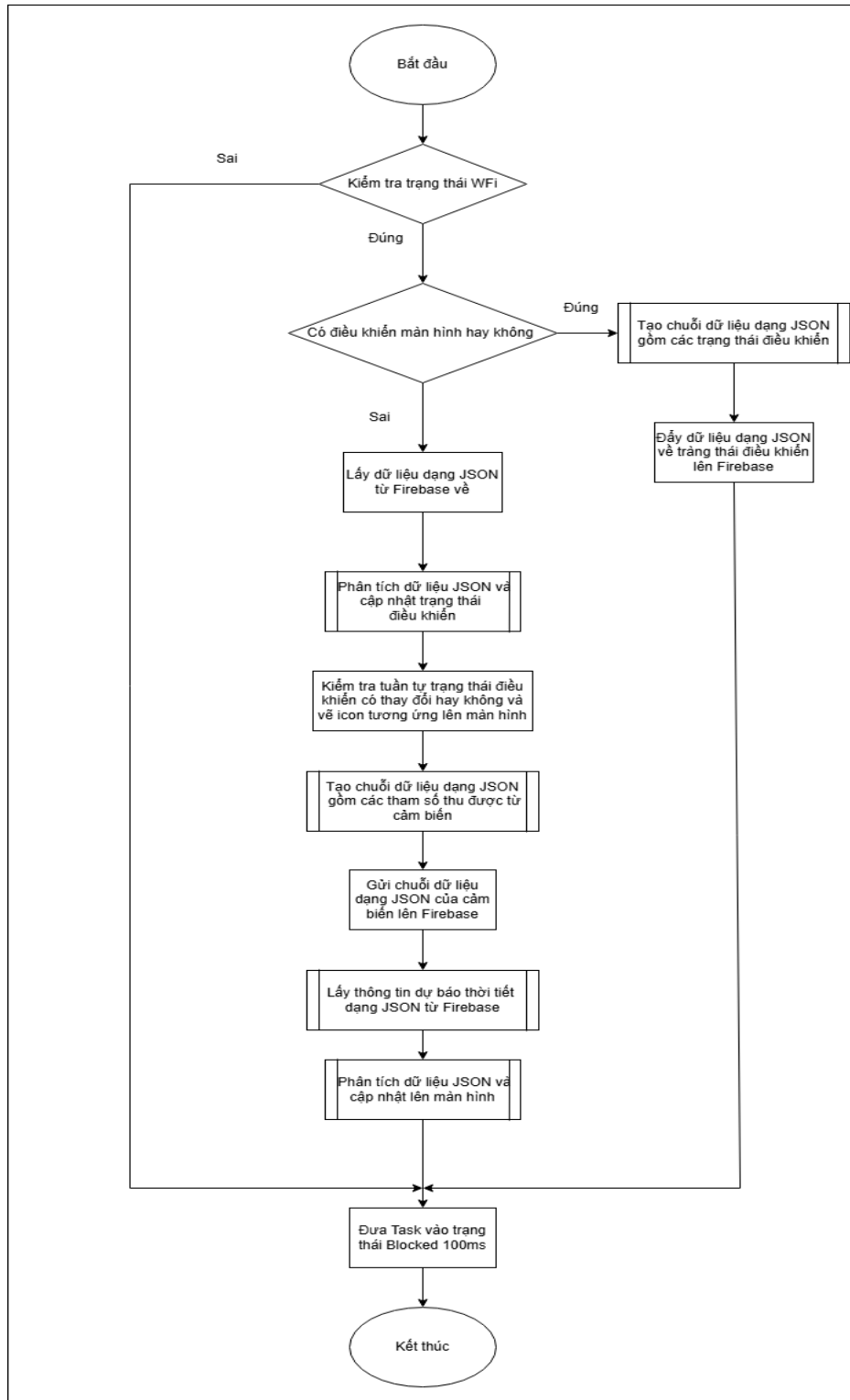
Hệ thống khởi động, cấu hình GPIO để thiết lập các chân làm đầu vào hoặc đầu ra, sau đó kết nối Wi-Fi để truyền nhận dữ liệu từ Firebase theo thời gian thực. Tiếp theo, giao diện điều khiển được khởi tạo để hiển thị dữ liệu. Hệ thống phân chia nhiệm vụ: Task 0 (đọc cảm biến, ưu tiên cao, Core 0), Task 1 (gửi dữ liệu lên Firebase, ưu tiên thấp, Core 1), và Task 2 (xử lý cảm ứng, ưu tiên cao, chạy trên bất kỳ Core). Cuối cùng, khi hoàn tất hệ thống chuyển sang trạng thái nghỉ.



Hình 6.2. Lưu đồ Task 0

Hệ thống bắt đầu kiểm tra nhiệt độ, độ ẩm từ cảm biến DHT11 và hiển thị lên màn hình TFT. Tiếp theo, nếu chức năng tự động phun sương được bật, hệ thống so sánh độ ẩm với ngưỡng yêu cầu để bật/tắt máy phun sương. Sau đó, hệ thống kiểm tra mức nước và kích hoạt máy bơm nếu mức nước thấp hơn ngưỡng khi chế độ tự động bơm nước đang bật. Tiếp đến, cường độ ánh sáng được kiểm tra, nếu thấp hơn ngưỡng yêu cầu, hệ thống sẽ bật đèn. Cuối cùng, hệ thống đo độ pH và dòng điện ACS712, hiển thị kết quả, chờ

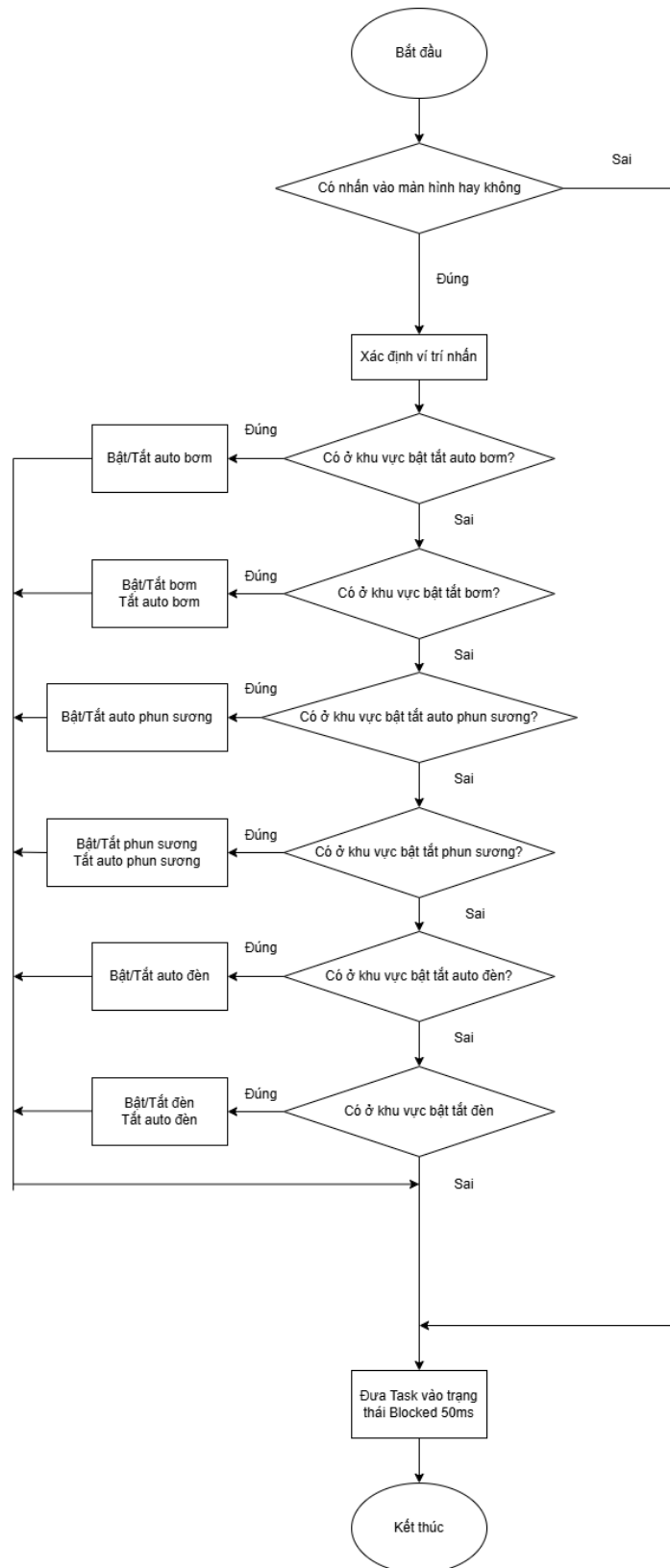
5000ms rồi lặp lại chu kỳ.



Hình 6.3. Lưu đồ Task 1

Hệ thống kiểm tra WiFi, nếu kết nối thành công, sẽ xử lý dữ liệu từ màn hình cảm ứng hoặc Firebase để cập nhật trạng thái thiết bị. Sau đó, hệ thống gửi dữ liệu cảm biến, nhận

dự báo thời tiết từ Firebase, hiển thị thông tin lên màn hình và chờ 100ms trước khi lặp lại chu kỳ, đảm bảo giám sát và điều khiển thiết bị theo thời gian thực.

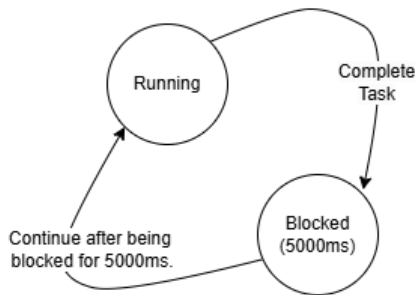


Hình 6.4 Lưu đồ Task 2

Quy trình bắt đầu bằng việc kiểm tra thao tác nhấn trên màn hình. Nếu không có, hệ thống quay lại trạng thái ban đầu. Nếu có, hệ thống xác định vị trí nhấn và kiểm tra các khu vực điều khiển (bơm, phun sương, đèn). Tại mỗi khu vực, nếu phù hợp, lệnh điều khiển được thực hiện; nếu không, tiếp tục kiểm tra. Sau khi xử lý xong, hệ thống chờ 50ms trước khi lặp lại. Quy trình đảm bảo xử lý chính xác mọi thao tác nhấn.

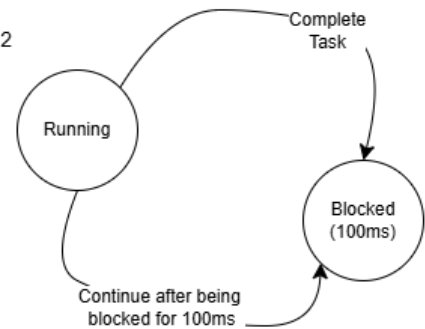
Task 0

Core 0, Priority 1



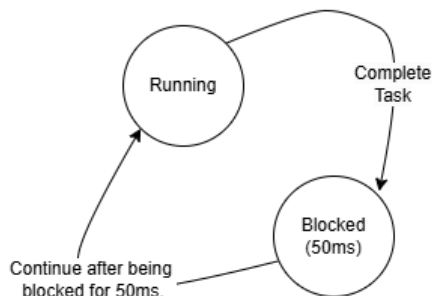
Task 1

Core 1, Priority 2



Task 2

No fixed core, Priority 1



Tương tác:

- Task 0 và Task 1 chạy trên các lõi riêng biệt, không can thiệp trực tiếp.
- Task 2 có thể chạy trên mọi lõi và có khả năng phát hiện các sự kiện chạm.

Hình 6.5 State Machine

Lưu đồ mô tả quy trình hệ thống khi người dùng tương tác với màn hình cảm ứng để điều khiển thiết bị. Nếu phát hiện thao tác nhấn, Task 1 sẽ tạm dừng, hệ thống xác định vị trí nhấn và thực hiện các chức năng tương ứng: bật/tắt chế độ tự động hoặc điều khiển thủ công thiết bị. Sau khi xử lý, Task 1 được khởi động lại, hệ thống chờ 50ms để tránh xung đột, rồi lặp lại chu kỳ, đảm bảo tương tác chính xác và ổn định.

7. Ghi chú phát triển

Mô hình thủy canh hiện tại đã đạt được những kết quả đáng khích lệ nhưng cần cải tiến để đáp ứng tốt hơn nhu cầu thực tế. Trong tương lai, việc nâng cấp phần mềm điều khiển, phát triển ứng dụng di động, và tích hợp AI để dự đoán nhu cầu nước, ánh sáng, dinh dưỡng sẽ là ưu tiên. Hệ thống cần hỗ trợ các nền tảng IoT như Google Home, Amazon Alexa và bổ sung điều khiển giọng nói, nhận diện hình ảnh. Về phần cứng, nâng cấp cảm biến, thiết kế mạch PCB nhỏ gọn, bền hơn và tích hợp năng lượng tái tạo như năng lượng mặt trời sẽ giúp giảm chi phí và bảo vệ môi trường. Hệ thống cũng cần chức năng cảnh báo thông minh và cơ chế tự động khắc phục sự cố. Đặc biệt, việc mở rộng quy mô và quản lý tập trung cho các mô hình lớn sẽ tối ưu hóa tài nguyên, nâng cao năng suất và thúc đẩy nông nghiệp bền vững.

8. Tham khảo

W. Gay, *FreeRTOS For ESP32-Arduino_Practical Multitasking Fundamental*, Elektor Publishing, 2020.

B. Amos, *Hands-On RTOS with Microcontrollers_Building real-time embedded systems using FreeRTOS, STM32 MCUs and SEGGER debug tools*, Packt Publishing, 2020.

B. Frain, *Responsive Web Design with HTML5 and CSS_Build future-proof responsive websites using the latest HTML5 and CSS techniques, 4th Edition*, Packt Publishing, 2022.

James F. Hancock. *Strawberries*. CABI Publishing, 2020

H. M. Resh, *Hydroponic Food Production*. Boca Raton, FL, USA: CRC Press, 2013.

T. T. M. Châu, "Ứng dụng thủy canh trong trồng dâu tây tại Đà Lạt," *Tạp chí Khoa học và Công nghệ Việt Nam*, 2020.

N. V. Tuấn, "Nông nghiệp công nghệ cao tại Đà Lạt: Thực trạng và giải pháp," *Tạp chí Khoa học Nông nghiệp*, 2018

A. Tannenbaum, *The Complete Guide to Hydroponics*, Alpha Books, 2018.

A. Trần, "Hệ thống màng dinh dưỡng (NFT) trong thủy canh," *Cây cảnh Hải Đăng*. <https://caycanhhaidang.com/ky-thuat-mang-dinh-duong>. (truy cập ngày 1 tháng 10 năm 2024)

C. Nguyễn, "Hệ thống thủy canh: Các mô hình và ứng dụng," *ttdaynghehotronongdan.baria-vungtau.gov.vn*, 2023. <https://ttdaynghehotronongdan.baria-vungtau.gov.vn>

vungtau.gov.vn/khoa-hoc-ky-thuat/-/view_content/content/55724/cac-mo-hinh-he-thong-thuy-can-h-co-ban-hien-nay . (truy cập ngày 2 tháng 10 năm 2024).

T. Vũ, "Các hệ thống thủy canh cơ bản," *Agri.vn*, 2023. <https://agri.vn/cac-mo-hinh-can-h-tac-thuy-can-h-co-ban-3>. (truy cập ngày 2 tháng 10 năm 2024).

Daryl K. S., *Serial Communications Developer's Guide*, 1st ed. Upper Saddle River, NJ: Prentice Hall, 1999

P. Horowitz and **W. Hill**, *The Art of Electronics*, 3rd ed. Cambridge, U.K.: Cambridge University Press, 2015.

Philips Semiconductors, "The I²C-bus specification and user manual," 1998.

V. Himpe, *I2C Bus: From Theory to Practice*, 1st ed. San Francisco, CA, USA: Prentice Hall, 2004.

Firebase, "Firebase Realtime Database," *Firebase Documentation*. <https://firebase.google.com/docs/database>. (truy cập ngày 1 tháng 10 năm 2024)

Google Developers, "Firebase Realtime Database," *Google Developers*. <https://developers.google.com/database>. (truy cập ngày 1 tháng 10 năm 2024)

Espressif Systems, "ESP32 Technical Reference Manual," Espressif Systems, [https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf]. (truy cập ngày 1 tháng 10 năm 2024)

Aosong Electronics Co., Ltd., "DHT11 Humidity & Temperature Sensor," *Datasheet*. https://components101.com/sites/default/files/component_datasheet/DHT11.pdf. (truy cập ngày 1 tháng 10 năm 2024)

SparkFun Electronics, "LDR - Light Dependent Resistor," *Datasheet*. <https://cdn.sparkfun.com/datasheets/Sensors/LightImaging/SEN-09088.pdf>. (truy cập ngày 1 tháng 10 năm 2024)

Texas Instruments, "Water Level Detection Using Capacitive Sensors," *IEEE Sensors Journal*, vol. 20, no. 5, pp. 2240-2246, 2020.

DFRobot, "PH sensor module PH-4502C," <https://www.dfrobot.com/product-1025.html>. (truy cập ngày 1 tháng 10 năm 2024)

Allegro Microsystems, "ACS712 Current Sensor," Allegro Microsystems <https://www.allegromicro.com/~media/Files/Datasheets/ACS712-Datasheet.ashx> (truy cập ngày 1 tháng 10 năm 2024)

LCDWiki, "4.0inch Capacitive SPI Module ST7796," *LCDWiki* http://www.lcdwiki.com/4.0inch_Capacitive_SPI_Module_ST7796 (truy cập ngày 1 tháng 10 năm 2024)

Songle Relay, "SRD-05VDC-SL-C Datasheet," *LCSC*.
https://www.lcsc.com/datasheet/lcsc_datasheet_2304140030_Ningbo-Songle-Relay-SRD-05VDC-SL-C_C35449.pdf. (truy cập ngày 1 tháng 10 năm 2024)

Thegioiic, "DP-521 Máy bơm áp suất phun sương 12VDC 2A 0.48MPa," *Thegioiic*.
<https://www.thegioiic.com/dp-521-may-bom-ap-suat-phun-suong-12vdc-2a-0-48mpa>.

(truy cập ngày 1 tháng 10 năm 2024)

Thegioiic, "AD20P-1230C Động cơ bơm chìm không chổi than 12VDC 240L/H 3W," *Thegioiic*.
<https://www.thegioiic.com/ad20p-1230c-dong-co-bom-chim-khong-choi-than-12vdc-240l-h-3w>. (truy cập ngày 1 tháng 10 năm 2024)

Thegioiic, "LED Hắt 3 bóng trắng SMD 5054 12VDC IP65," *Thegioiic*.
<https://www.thegioiic.com/led-hat-3-bong-trang-smd-5054-12vdc-ip65> (truy cập ngày 1 tháng 10 năm 2024)

PHỤ LỤC