

بسمه تعالی

نام دانشجو : مجتبی منافپور

شماره دانشجویی : 14025361037

درس : سیستم های نهفته و بیدرنگ

نام استاد : دکتر محمد حسین محمد زاده

گزارش جامع پروژه سیستم کنترل گلخانه (IoT Greenhouse Control System)

مقدمه

با پیشرفت تکنولوژی اینترنت اشیا (IoT)، پیاده سازی سیستم های هوشمند در حوزه کشاورزی و گلخانه ها به منظور نظارت و کنترل بهتر شرایط محیطی اهمیت یافته است. هدف این پروژه، طراحی و پیاده سازی یک سیستم کنترل هوشمند گلخانه بر پایه ESP32 با امکان مانیتورینگ از طریق اپلیکیشن موبایل Flutter است.

این پروژه، ترکیبی از سخت افزار (ESP32, LCD)، سنسورها، نرم افزار (WebServer، Application). است که به کاربران امکان ارتباط و کنترل از راه دور پارامترهایی مانند دما، رطوبت و زمان بندی آبیاری را می دهد.

تجهیزات و قطعات سخت افزاری

برد ESP32: هسته اصلی سیستم، با قابلیت ارتباط WiFi، اجرای وب سرور، کنترل سنسورها و عملگرها (سیستم خنک کننده، گرمایشی و آبیاری).

سنسور DHT11: برای اندازه گیری دما و رطوبت محیط.

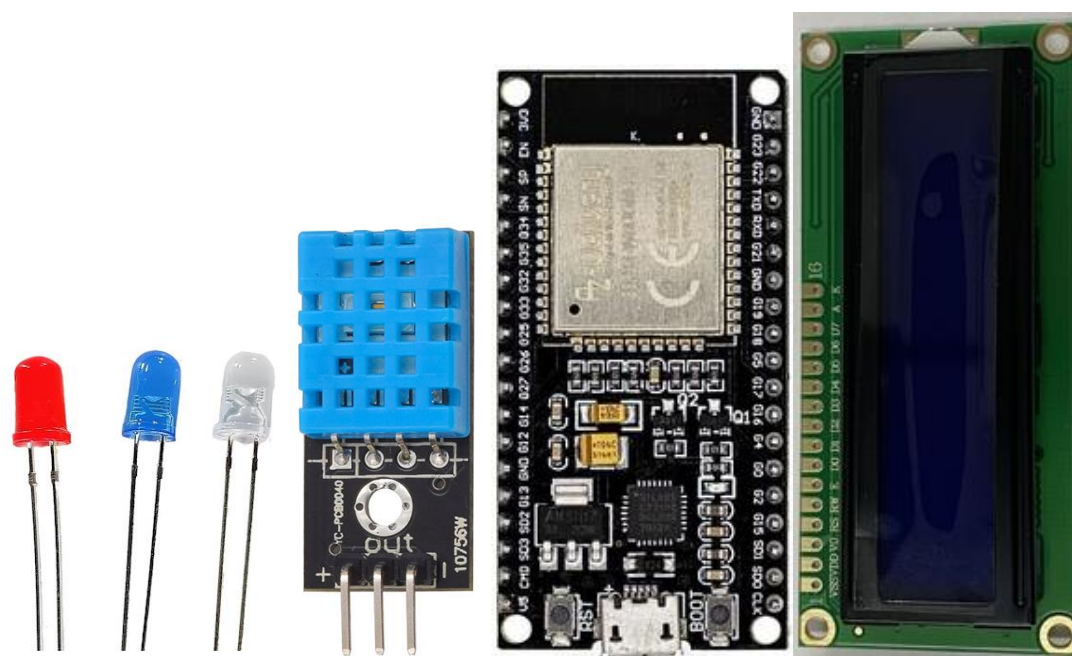
نمایشگر LCD 16x2: جهت نمایش لحظه ای دما، رطوبت و ساعت.

ماژول **NTPClient**: برای دریافت زمان دقیق از طریق اینترنت.

LED: برای نمایش وضعیت فن، بخاری و پمپ آب.

BreadBoard and Wires: برای اتصالات موقت.

تغذیه : تأمین برق ESP32 با اتصال به USB



3xLED

DHT11

ESP32

LCD(16x2)

طراحی اولیه

در مرحله طراحی با توجه دیاگرام Design Flow که در زیر نمایش داده شده است پیش رفتیم و قسمت Specification را با استفاده از زبان lePUS3 که یک زبان ترسیم گرافیکی شئ گراست در نرم افزار StarUML ترسیم کردیم.

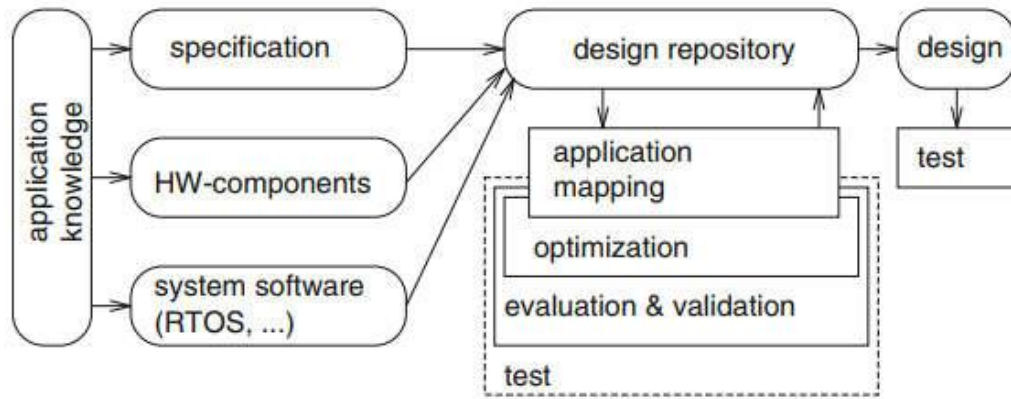
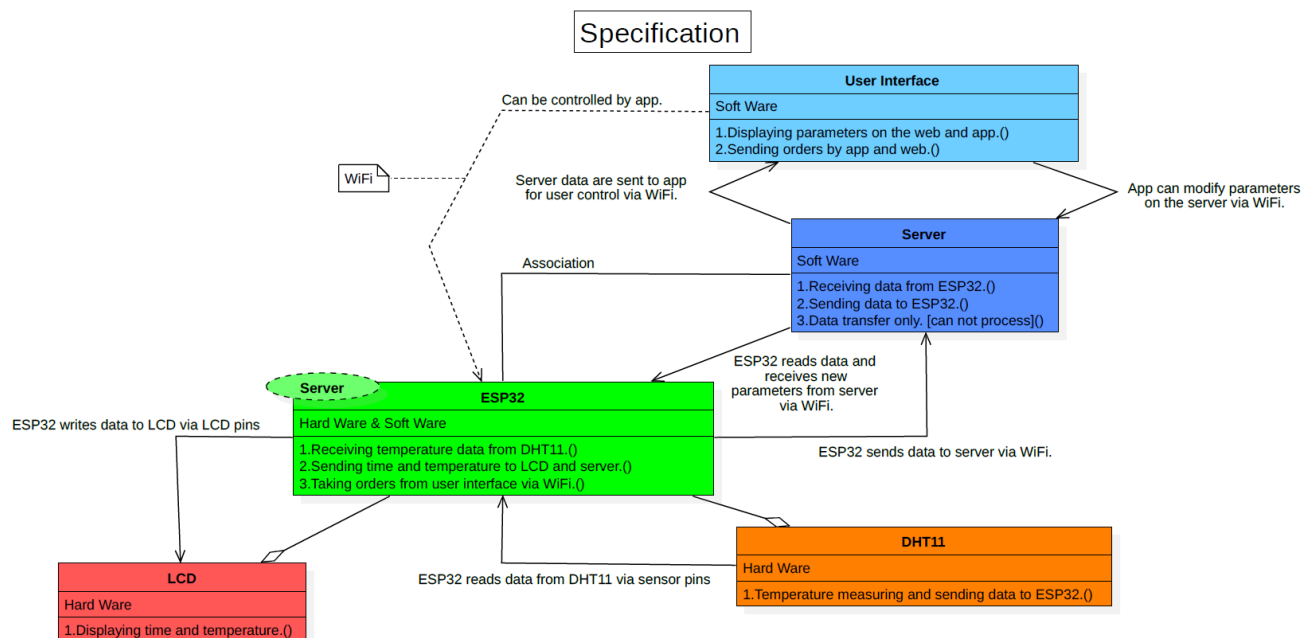


Fig. 1.8 Simplified design information flow

مدل سطح سیستم (System-Level Models) :

اصطلاح سطح سیستم تعریف دقیقی ندارد، اما در اینجا منظور از آن، کل سیستم نهفته ، سیستمی است که پردازش اطلاعات در آن قرار گرفته است.

Design Flow (Specification)



نرم افزارها و محیط های توسعه

Arduino IDE: جهت توسعه و آپلود کد بر روی ESP32.

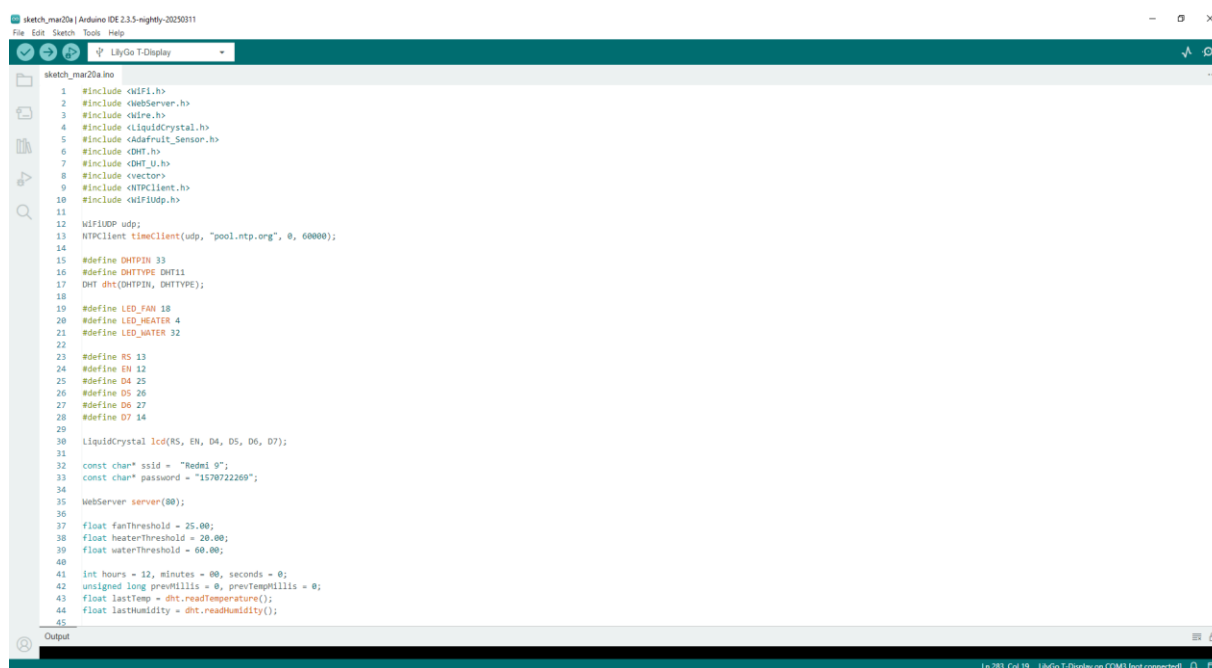
Wokwi: برای شبیه سازی سخت افزار در مراحل ابتدایی.

Visual Studio Code (VS Code): برای توسعه اپلیکیشن Flutter.

Android Studio: برای اجرا و تست اپلیکیشن روی دستگاه واقعی یا شبیه ساز.

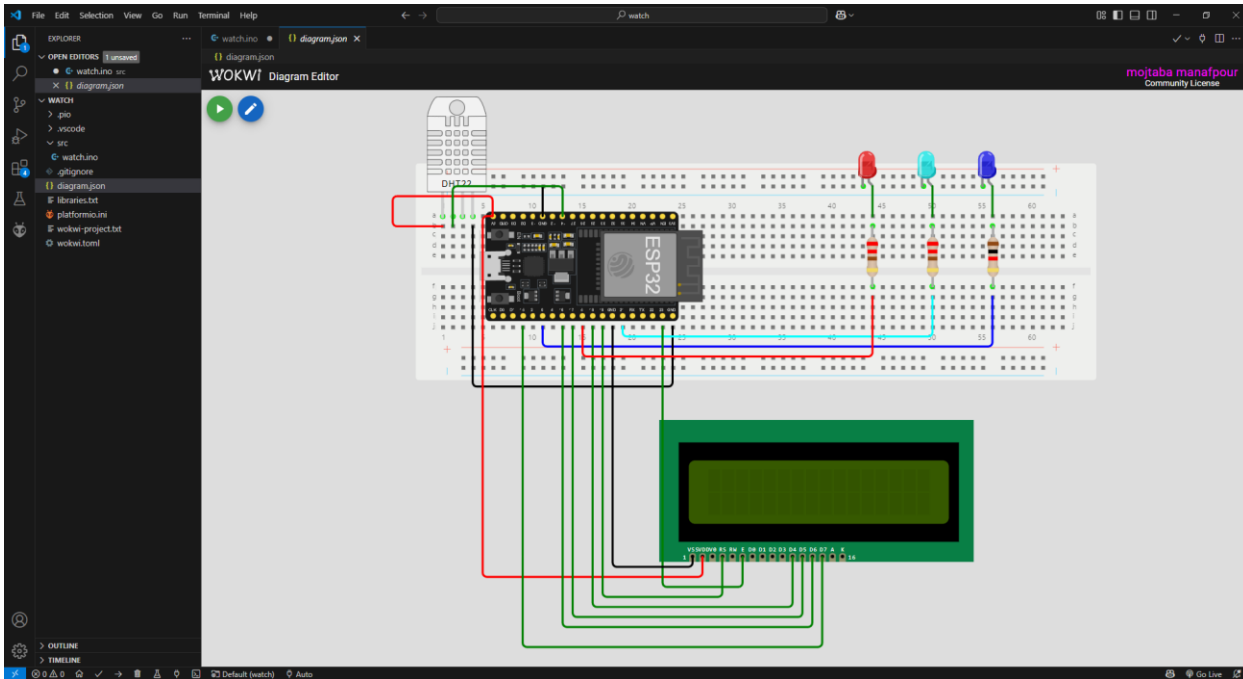
Flutter SDK: برای توسعه رابط کاربری موبایل اندروید با پشتیبانی از تم شب و روز

Arduino IDE

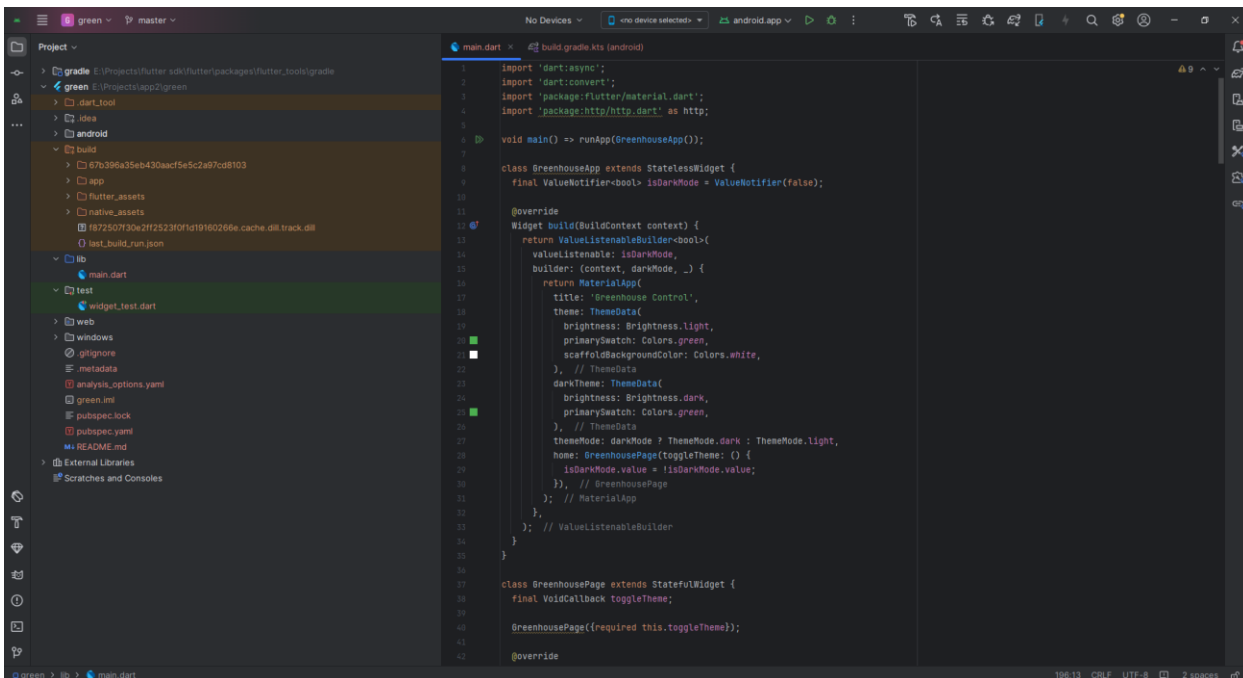


```
1 #include <WiFi.h>
2 #include <WebServer.h>
3 #include <Wire.h>
4 #include <LiquidCrystal.h>
5 #include <Adafruit_Sensor.h>
6 #include <DHT.h>
7 #include <DHT_U.h>
8 #include <vector>
9 #include <NTPClient.h>
10 #include <WiFiUdp.h>
11
12 WiFiUDP udp;
13 NTPClient timeClient(udp, "pool.ntp.org", 0, 60000);
14
15 #define DHTPIN 33
16 #define DHTTYPE DHT11
17 DHT dht(DHTPIN, DHTTYPE);
18
19 #define LED_FAN 18
20 #define LED_HEATER 4
21 #define LED_WATER 32
22
23 #define RS 13
24 #define EN 12
25 #define D4 25
26 #define D5 26
27 #define D6 27
28 #define D7 14
29
30 LiquidCrystal lcd(RS, EN, D4, D5, D6, D7);
31
32 const char* ssid = "Radmi 9";
33 const char* password = "1578722289";
34
35 WebServer server(80);
36
37 float fanThreshold = 25.00;
38 float heaterThreshold = 28.00;
39 float waterThreshold = 60.00;
40
41 int hours = 12, minutes = 00, seconds = 0;
42 unsigned long prevMillis = 0, prevTempMillis = 0;
43 float lastTemp = dht.readTemperature();
44 float lastHumidity = dht.readHumidity();
45
```

Wokwi in VS code



Android Studio



قابلیت‌های سیستم

نمایش پارامترهای لحظه‌ای

زمان لحظه‌ای با استفاده از (NTPClient)

دما (Celsius) و رطوبت (درصد/)

وضعیت سیستم خنک کننده، گرمایشی و آبیاری

کنترل خودکار عملگرها

فن فعال می‌شود اگر دما از آستانه تعیین شده بیشتر شود و رطوبت کافی باشد.

سیستم گرمایشی فعال می‌شود اگر دما کمتر از حد مجاز باشد.



سیستم آبیاری براساس زمان بندی آبیاری فعال می‌شود.



تنظیم آستانه‌ها از طریق وب و اپلیکیشن




کاربر می‌تواند از طریق اپلیکیشن یا صفحه وب مقادیر آستانه برای فن، سیستم گرمایشی و آبیاری و رطوبت را وارد کند. این مقادیر به ESP32 ارسال و ذخیره شده و در تصمیم‌گیری‌های کنترلی استفاده می‌شوند.


Web Page


72



 10:30



 192.168.54.2



Greenhouse Control Panel

Time: **10:30:50**

Temperature: **24.80 C**

Humidity: **70.00 %**

Fan Threshold: **25.00**

Heater Threshold: **20.00**

Humidity Threshold: **60.00**

Fan Threshold:

Heater Threshold:

Water Threshold:

Set Values

Set Watering Schedule

Start Time:

End Time:

Add Watering Time

Current Watering Schedule

مدیریت برنامه ریزی آبیاری

افزودن بازه های زمانی برای آبیاری خودکار.

حذف زمان بندی ها از طریق وب و اپلیکیشن.

اجرای دقیق بر اساس ساعت داخلی.

اپلیکیشن موبایل

دارای رابط کاربری مدرن و ساده.

نمایش لحظه ای وضعیت ها.

قابلیت تغییر بین حالت شب و روز.

افزودن و حذف زمان های آبیاری.

تغییر آستانه ها.

همگام سازی کامل با وب سرور ESP32.


Mobile APP(Dark & Light Them)


72


4.5G


DEBUG


Greenhouse Dashboard


 **Time: 10:30:5**


 Temp: 24.50 °C

 Humidity: 72.00 %

 Fan: 25.00

 Heater: 20.00


 Water: 60.00


 **Set Thresholds**

Fan Threshold

Heater Threshold


Humidity Threshold


 Set Thresholds

 **Add Watering Schedule**

Start Time (HH:MM)

End Time (HH:MM)

 Add Watering


 **Watering Schedule**


72


4.5G


DEBUG


Greenhouse Dashboard


 **Time: 10:29:57**


 Temp: 24.50 °C

 Humidity: 72.00 %

 Fan: 25.00

 Heater: 20.00


 Water: 60.00


 **Set Thresholds**

Fan Threshold

Heater Threshold


Humidity Threshold


 Set Thresholds

 **Add Watering Schedule**

Start Time (HH:MM)

End Time (HH:MM)

 Add Watering

 **Watering Schedule**

ساختار کد (ESP32) Arduino

تعریف پایه‌ها، پین‌های LED و LCD

تعریف ساختار WateringSlot برای زمان‌بندی.

استفاده از WebServer برای ایجاد API های /status، /set، /add_watering، /delete_watering.

ارسال پاسخ JSON از طریق /status برای اپلیکیشن Flutter.

نمایش زمان، دما و رطوبت روی LCD.

بررسی زمان فعلی برای آبیاری خودکار و فعال‌سازی پمپ آب.

ساختار کد Flutter

دریافت اطلاعات لحظه‌ای از /status به صورت JSON.

نمایش داده‌ها در یک Card.

TextField برای ورودی آستانه‌ها و زمان‌بندی.

ارسال POST به /set و /add_watering.

مدیریت لیست برنامه آبیاری با افزودن و حذف.

استفاده از ValueNotifier برای تعویض تم روز و شب.

یکپارچه‌سازی کامل با کد ESP32.

نکته مهم: رطوبت به صورت Field جدید humidity در JSON اضافه شده و در اپ نیز خوانده می‌شود.

نقاط قوت پروژه

عملکرد دقیق و هماهنگ سخت افزار و اپلیکیشن.

قابلیت کارکرد آفلاین از طریق شبکه محلی WiFi.

ظاهر کاربرپسند اپلیکیشن با دو حالت شب و روز.

خواندن همزمان دما و رطوبت و به روزرسانی روی LCD.

کنترل از راه دور با اتصال به USB.

نتیجه گیری

این پروژه توانسته است یک سیستم کنترل دقیق و کامل برای گلخانه طراحی و پیاده سازی کند. تعامل بین سخت افزار و اپلیکیشن به خوبی انجام شده و کاربر می تواند از طریق موبایل به راحتی وضعیت را کنترل کند.