

IoT Smart Watch

Проект годинника інтернет речей
Автор: Гончаренко Андрій

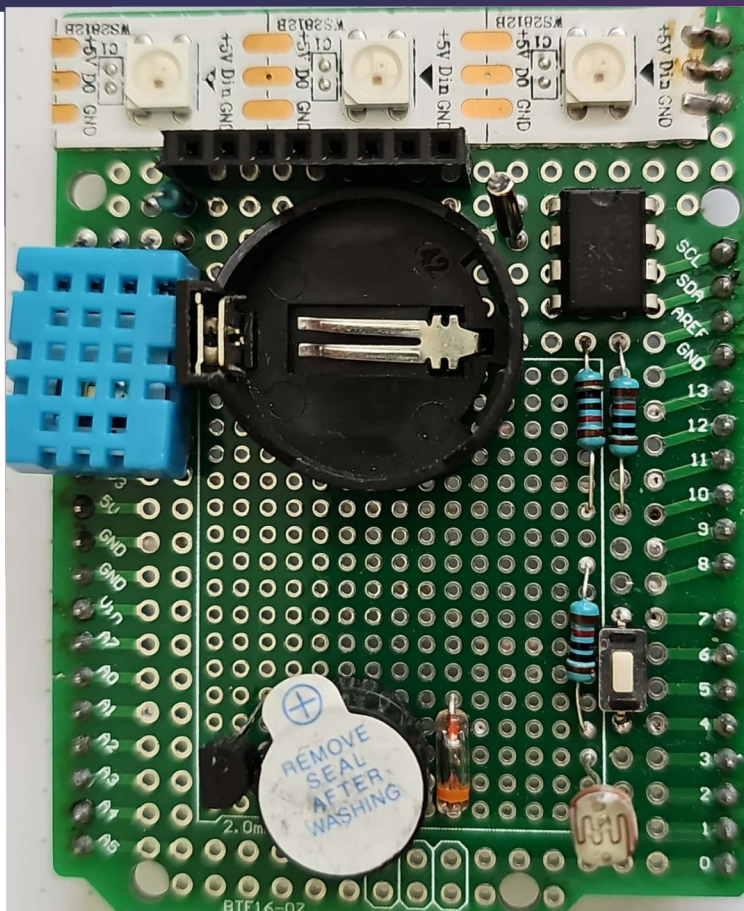
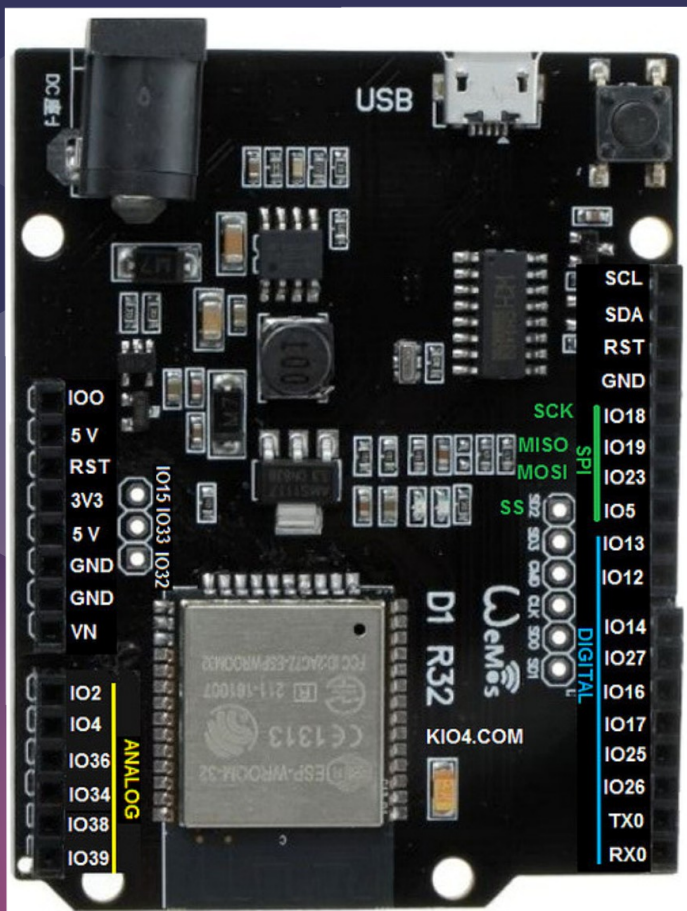
Вступ

- IoT (Internet of Things) - це концепція, яка описує мережу фізичних пристроїв, обладнаних сенсорами, програмним забезпеченням та здатністю взаємодіяти між собою та з іншими пристроями через Інтернет. Ці пристрої можуть бути різних типів: від простих датчиків температури до складних систем управління будинком або індустріальними процесами.
- Основна ідея IoT полягає в тому, щоб надати речам, які нас оточують, здатність сприймати інформацію про своє середовище, комунікувати між собою та приймати рішення на основі зібраної інформації. Це відкриває безліч можливостей у різних сферах, включаючи домашнє господарство, медицину, транспорт, енергетику та багато іншого.
- У моєму проекті, "IoT Smart Watch", я використовую принципи IoT для розробки інтелектуального годинника, який не лише надає базові функції годинника, але й може збирати дані, взаємодіяти з іншими пристроями та надавати розширені можливості для користувача.
- Проект "IoT Smart Watch" поєднує сучасні технології, зручність використання та підключення до Інтернету речей, щоб створити інноваційний та функціональний пристрій, який підвищує здатність взаємодіяти зі світом навколо нас.

Огляд смарт-годинника як пристрою Інтернету речей

- Мій проект "IoT Smart Watch" базується на ідеї смарт-годинника, як пристрою Інтернету речей. Давайте поглянемо на основні можливості та функції, які реалізовані в моєму проекті згідно коду, доступному за посиланням https://github.com/taburyak/IoT_Smart_Watch
- Збір даних: Смарт-годинник обладнаний датчиком DHT11 який збирає інформацію про температуру і вологість в приміщенні. Демонструє інформацію на кольоровому дисплеї і відправляє данні в хмарний сервіс <https://thingspeak.com/> . Спостерігати за зібраною інформацією можна на публічному каналі <https://thingspeak.com/channels/2193007>
- Доступ до Інтернету: Завдяки можливості підключення до Інтернету, смарт-годинник може отримувати оновлення програмного забезпечення, синхронізувати дані з хмарними сервісами та отримувати доступ до різних онлайн-ресурсів. Наприклад, на поточний момент цей годинник має доступ до статистики мого youtube каналу <https://www.youtube.com/channel/UCHMrCiNnJB6PrRaDDZEyZQ> і демонструє на дисплей кількість підписників, кількість переглядів і кількість опублікованих на каналі відео. А також має доступ до погодного сервісу <https://openweathermap.org/> завдяки якому можемо отримувати багато погодних даних в тій місцевості де знаходиться годинник.

Залізяччя



Пристрій складається з основної плати ESP32 arduino formfactor та допоміжної плати arduino formfactor де напаяні: роз'єм для дисплею, датчик DHT11, BUZZER, RTC + батарейний відсік, датчик освітленості, 3 програмовані RGB світлодіоди WS2812B, та кнопка користувача.

Засоби розробки

- Основна IDE розробки це Visual Studio Code з плагіном Platform.io. Що дозволяє комфортно писати код, мати автодоповнення слів і звичне кольорове підсвічування коду.
- Також плагін platformio дозволяє мати з коробки компілятор, доступ до репозиторію з популярними бібліотеками та одночасно всі доступні framework для обраної платформи. Наприклад для ESP32 це буде Arduino framework та Espressif IoT Development Framework + FreeRTOS з коробки.
- Код і бібліотеки написані мовою C++

Архітектура проекту

- Самий вищий рівень проекту це **«Машина станів»**
- **"Машина станів"** - це поняття в програмуванні, що описує модель поведінки системи або об'єкта, який може перебувати в різних станах і здійснювати переходи між ними відповідно до певних подій або вхідних сигналів.
- Основна ідея машини станів полягає в тому, що об'єкт може перебувати у конкретному стані, і цей стан визначає, які дії відбуваються або які подальші стани можуть бути досягнуті. Коли виникає певна подія або вхідний сигнал, машина станів переходить в інший стан згідно визначених правил.
- **"Машина станів"** включає такі ключові поняття:
 1. **Стан:** Це конкретний стан, в якому перебуває об'єкт або система. Наприклад, "вимкнено", "увімкнено", "працює", "зупинено", «З'єднання з мережею встановлено / відсутнє».
 2. **Подія:** Це подія або вхідний сигнал, що спричиняє перехід машини станів із поточного стану в інший. Наприклад, "ввімкнути", "вимкнути", "почати роботу", "зупинити", «сигнал втрачено», «пройшов певний відлік часу».
 3. **Перехід:** Це правило, яке визначає, які дії відбуваються при переході машини станів від одного стану до іншого в результаті виникнення певної події. Наприклад, при події "ввімкнути" машина станів може перейти зі стану "вимкнено" до стану "увімкнено" і виконати відповідні дії.
 4. **Дії:** Це дії або операції, які виконуються при переході машини станів від одного стану до іншого. Наприклад, зміна параметрів, виконання функцій або взаємодія з іншими об'єктами.
- Машини станів є потужним інструментом для моделювання складного поведінки систем та об'єктів у програмуванні. Вони дозволяють організовувати програмний код у логічні блоки, які керують поведінкою системи залежно від її поточного стану і вхідних подій. Це сприяє зрозумілості, модульності та підтримці великих програмних проектів.

Модулі проекту

- На рівень нижче маємо певні модулі для роботи пристрою, такі як:
 - 1) Модуль для налаштування wifi з'єднання, зберігання конфігурації wifi, та встановлення з'єднання з WiFi роутером
 - 2) Модуль для ініціалізації і отримання даних від сенсора DHT11
 - 3) Модуль який виводить на дисплей необхідну інформацію
 - 4) Модуль для обробки і створені переривань часових проміжків часу.
 - 5) Модуль роботи з хмарними сервісами такі як доступ до погоди, запит статистики ютюб каналу і надсилання зібраної інформації від сенсору DHT на спеціальний сервіс для накопичення даних і його аналіз.

Бібліотеки проекту

- Ще на рівень нижче присутні бібліотеки для забезпечення роботи всіх модулів на рівень вище. Та логіка взаємодії між модулями.

Планується реалізувати додатково

- Яскравість дисплею в залежності від навколишнього освітлення
- ESP32 має внутрішній RTC модуль. Але з-за того, що я використовую готову плату з ESP32, на ній не передбачено підключення кварцу, конденсаторів і батарейки до вбудованого модуля RTC. Тому буде інтегровано зовнішній RTC DS1307 який взаємодіє по шині I2C. За допомоги нього матиму точний час та дату без з'єднання до мережі і синхронізації з часовою службою.
- Buzzer для сповіщення сигналу будильника, або попередження про штормову погоду або дощ що наближається.
- Світлодіодну стрічку з 3 програмованих RGB світлодіодів для реалізації показу станів пристрою та відображення якихось подій. Наприклад стан без мережі, або з мережею. Граничних станів температури і вологості, тощо.

Контактна інформація

E-mail: taburyak@gmail.com

GitHub:

https://github.com/taburyak/IoT_Smart_Watch

YouTube:

<https://www.youtube.com/channel/UCHMrCiNnJB6PrRraDDZEyZQ>

Blog: <https://stm32withoutfear.blogspot.com/>