

ПРИКЛАД ВИКОНАННЯ

Мета: Вивчити принципи побудови пристроїв Інтернету речей на основі фреймворку QPF (Quantum Platform Framework) та освоїти методику розробки програмного забезпечення з використанням парадигми активних об'єктів і скінченних автоматів.

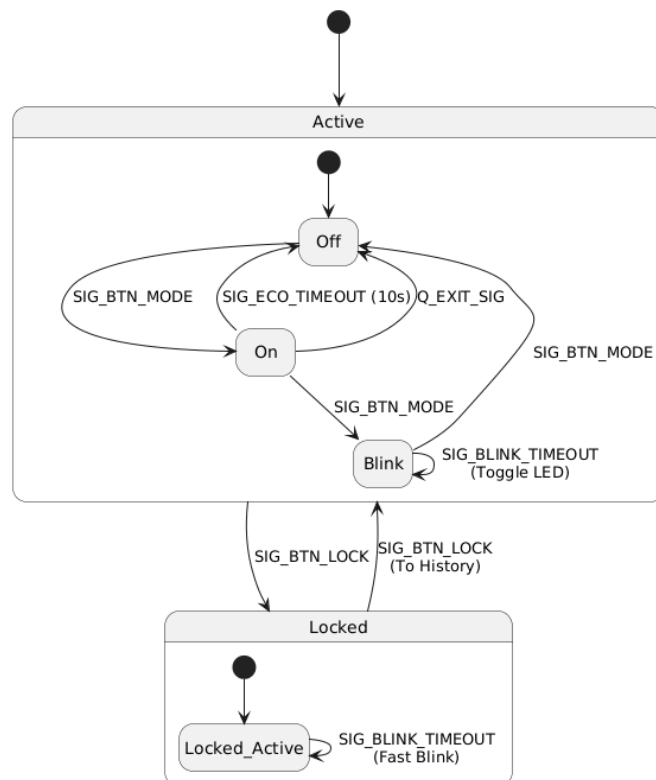
Завдання: розробити "Інтелектуальний контролер освітлення".

Варіант: додати стан **ECO_MODE**: якщо в стані **ON** не було натискань кнопок протягом 10 секунд — автоматичний перехід у **OFF**.

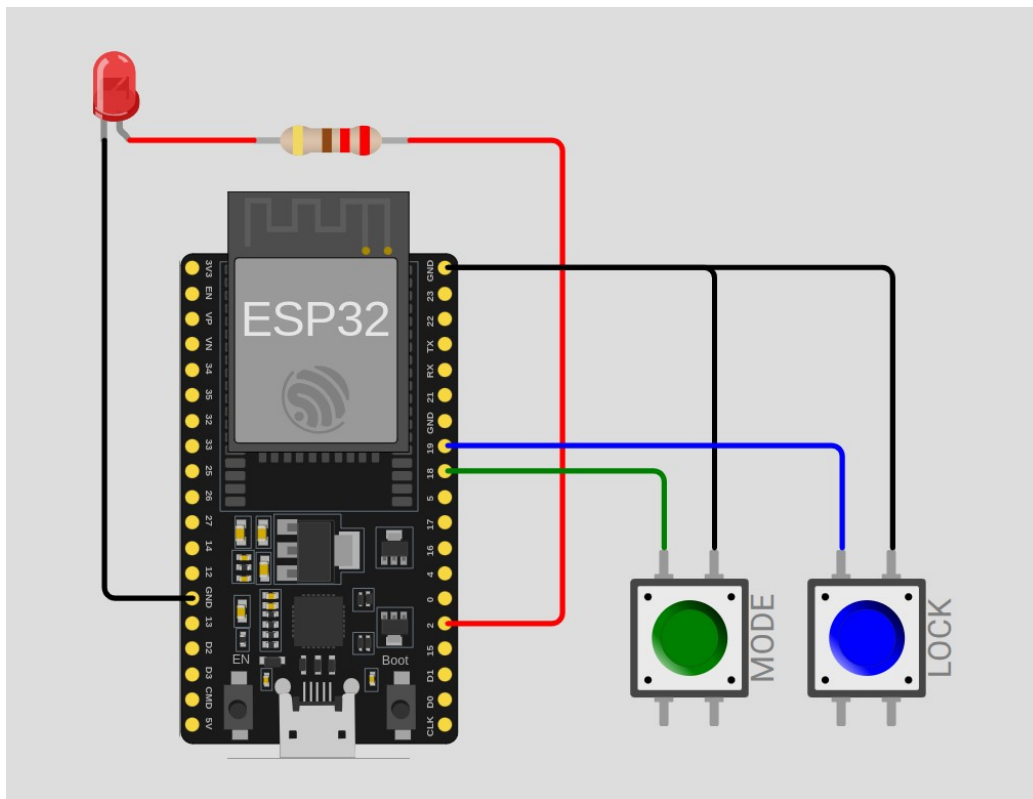
Розробка UML-діаграми станів

Діаграма станів має наступну ієрархію:

- **Active (Супер-стан):** Відповідає за загальну поведінку.
 - **Off:** Стан спокою (LED вимкнено).
 - **On:** Робочий режим (LED увімкнено, працює ECO-таймер).
 - **Blink:** Режим блимання (LED блимає з частотою 2 Гц).
- **Locked:** Стан блокування (система ігнорує кнопку MODE, LED швидко мерехтить).



Розроблена схема для симуляції у Wokwi



На розробленій схемі зображено підключення зовнішніх компонентів до плати розробника ESP32 DevKit V1, яке включає один світлодіод та дві тактові кнопки. Червоний світлодіод під'єднано до цифрового контакту GPIO 2. Схема його живлення реалізована через червоний провідник, який подає сигнал на струмообмежувальний резистор, послідовно з'єднаний з анодом світлодіода, тоді як катод світлодіода через чорний провідник замкнено на контакт заземлення (GND) у нижній лівій частині плати.

Система керування складається з двох кнопок, що працюють за принципом замикання на землю. Ліва кнопка, позначена як "MODE", підключена зеленим провідником до контакту GPIO 18. Права кнопка з підписом "LOCK" з'єднана синім провідником із контактом GPIO 19. Обидві кнопки мають спільну лінію заземлення, організовану через чорний провід, забезпечуючи подачу логічного нуля на вхід мікроконтролера при натисканні.

Логи роботи програми

ets Jul 29 2019 12:21:46

```
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:1156
load:0x40078000,len:11456
ho 0 tail 12 room 4
load:0x40080400,len:2972
entry 0x400805dc
BSP Initialized
BSP Initialized
SM: Initial -> Active
SM: Active Entry
STATE: OFF
EVENT: BTN_MODE -> Switching to ON
STATE: ON (ECO Timer started)
EVENT: BTN_MODE -> Switching to BLINK
STATE: BLINK
EVENT: BTN_MODE -> Switching to OFF
STATE: OFF
EVENT: BTN_MODE -> Switching to ON
STATE: ON (ECO Timer started)
EVENT: BTN_LOCK -> Locking system
STATE: LOCKED
EVENT: BTN_LOCK -> Unlocking (Restoring History)
SM: Active Entry
STATE: ON (ECO Timer started)
EVENT: ECO TIMEOUT -> Auto OFF
STATE: OFF
EVENT: BTN_MODE -> Switching to ON
STATE: ON (ECO Timer started)
EVENT: BTN_MODE -> Switching to BLINK
STATE: BLINK
EVENT: BTN_LOCK -> Locking system
STATE: LOCKED
EVENT: BTN_LOCK -> Unlocking (Restoring History)
SM: Active Entry
STATE: BLINK
EVENT: BTN_MODE -> Switching to OFF
STATE: OFF
```

Висновки: в ході виконання лабораторної роботи було реалізовано програмне забезпечення для мікроконтролера ESP32 з використанням фреймворку QPF.

Переваги підходу Event-Driven (QPF) порівняно з Bare Metal:

- **Реактивність:** Система миттєво реагує на події (натискання кнопок) завдяки механізму переривань, які генерують події для автомата.
- **Структурованість:** Використання ієрархічних автоматів (HSM) дозволило винести загальну логіку (наприклад, обробку кнопки блокування або ігнорування інших кнопок) на вищий рівень абстракції, що зменшує дублювання коду.
- **Масштабованість:** Додавання нових станів (наприклад, "LOCKED") або таймерів (ECO Timer для варіанту 1) не вимагає переписування основного циклу loop(), а реалізується через окремі функції-обробники.
- **Енергоефективність:** Оскільки основний цикл порожній, а планувальник запускається лише при настанні подій, це дозволяє процесору переходити в режим сну (Idle) між обробкою подій.

До недоліків можна віднести вищий поріг входження та складнішу на початковому етапі інфраструктуру коду (необхідність створення класів подій, сигналів, BSP) порівняно з простим "супер-циклом".

Поклик на репозиторій:

<https://github.com/Field-Effect-Transistor/PSIR3QPFrameworkExample>