EN3240: Embedded Systems Engineering
Assignment 2 — IoT Project

Name: John Doe
Index No: XXXXX

May 1, 2023

This is an individual assignment!
Due Date: 30 July 2023 by 11.59 PM

Instructions

Follow the steps given below. Add the Node-RED flow as a JSON file and Arduino code to the same folder. Make a ZIP file with these files and submit it to Moodle as XXXXX.zip, where XXXXX is your index number.

Project Steps

Medibox is a smart IoT device designed to help users manage their medications effectively. The device is a box-like structure that can be easily placed on a table or mounted on a wall. It has a sleek and modern design with a transparent cover that allows users to see the contents of the box.

Medibox comes equipped with several features that make it a valuable asset for managing medication schedules. The device has a built-in LCD that shows the date, time, and upcoming medication schedule. The device also has a speaker that provides audible reminders to take medication at the scheduled time.

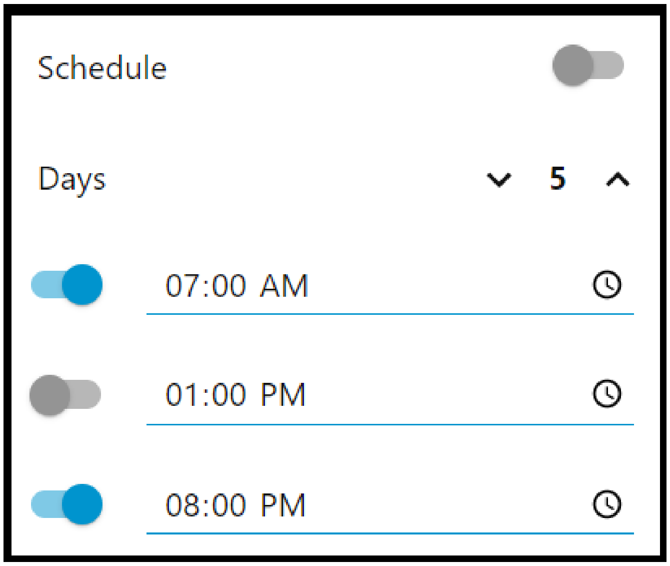
Medibox has an easy-to-use interface that allows users to input medication information, dosage, and schedule. The device has multiple compartments that can be labeled and assigned to different medications. The compartments are also fitted with sensors that detect whether medication has been removed or not, ensuring accurate tracking of medication intake.

In addition to its core features, Medibox can also be connected to the internet and synchronized with a Node-RED Dashboard. This enables users to receive notifications on their mobile devices, track medication history, set commands, and share information with healthcare professionals.

To create a Medibox prototype with IoT features, use Node-RED to design a dashboard and ESP32 as the Medibox controller. The project should include the following key features, which can be implemented using either real hardware or the Wokwi platform.

- Main buzzer
  - Medibox is equipped with a main buzzer, which is used to notify the user of different events or commands.
  - Include a main switch on the Node-RED dashboard to control the buzzer and properly synchronize it with the buzzer module to improve the overall user experience and provide greater flexibility in managing the Medibox device.
- Scheduler
  - The scheduler feature of Medibox is designed to allow users to schedule and automate the switching on of the buzzer at appropriate times.
  - Use a switch in the dashboard to toggle the scheduler feature on and off.
  - Allow users to select the number of days to repeat the Medibox scheduler, with automatic synchronization to reduce the remaining number of days when each day passes.

  - Enable users to select up to three alarms per day on Medibox, with switches for each time to disable or enable the alarms.
  - Figure 1 displays a sample group.



Schedule

Days 5

07:00 AM

01:00 PM

08:00 PM

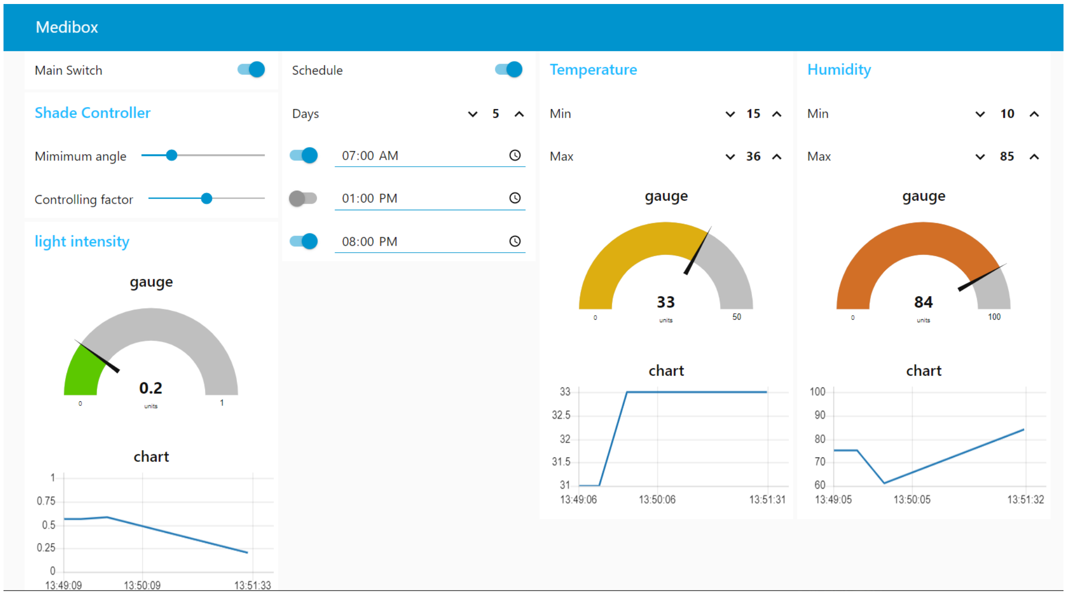
Figure 1: Scheduler Group

- Temperature and humidity measurements are necessary for certain medicines.
  - Incorporate a sensor in Medibox to measure the temperature and humidity levels.
  - Display temperature and humidity measurements accurately on the Node-RED dashboard using a gauge to show real-time values and plots to show historical values.
  - Allow users to set maximum and minimum thresholds for temperature and humidity levels. If the temperature and humidity measurements go beyond the thresholds, display a warning notification on the Node-RED dashboard to alert users of the situation.
- It is essential to monitor light intensity when storing certain medicines as they may be sensitive to sunlight.
  - To measure the intensity of light, it is recommended to use a Light Dependent Resistor (LDR).
  - Use a separate group to display the light intensity on the Node-RED dashboard. Within this group, it is recommended to use a gauge to display the real-time intensity and a plot to visualize past variations.
  - To ensure consistency in the display of intensity values, use a range of 0 to 1, where 0 represents the minimum possible value of intensity and 1 represents the maximum possible value of intensity.
- A shaded sliding window has been installed to prevent excessive light from entering the Medibox.
  - The shaded sliding window is connected to a servo motor responsible for adjusting the light intensity entering the Medibox. The motor can adjust its angle between 0-180 degrees based on the lighting conditions. This enables the system to dynamically regulate the amount of light entering the Medibox to ensure optimal storage conditions for sensitive medicines.
  - The following equation represents the relationship between the motor angle and the intensity of light entering the Medibox:
    $\theta = \theta_{\text{offset}} + (180 - \theta_{\text{offset}}) \times I \times \gamma$
    where,

        - θ is the motor angle
        - θ_{offset} is the minimum angle (default value of 30 degrees)
        - I is the intensity of light, ranging from 0 to 1
        - γ is the controlling factor (default value of 0.75)

- Different medicines may have different requirements for the minimum angle and a controlling factor is used to adjust the position of the shaded sliding window.
    - To enable the user to adjust the minimum angle and controlling factor, create a new group in the Node-RED dashboard.
    - Use a separate group to display the light intensity on the Node-RED dashboard. Within this group, it is recommended to use a gauge to display the real-time intensity and a plot to visualize past variations.
    - To adjust the system's minimum angle and controlling factor, it is recommended to use two slider controls in the Node-RED dashboard. The first slider control should range from 0 to 120, allowing the user to adjust the minimum angle of the shaded sliding window as needed. The second slider control should range from 0 to 1, enabling the user to adjust the controlling factor used to calculate the motor angle.
- Create a separate "Settings" tab in Medibox for users to make adjustments to the device's settings.
    - Create a separate group for adjusting the buzzer settings in Medibox, with a dropdown menu to select the buzzer type (continuous/repeated on-off), a field for entering the buzzer delay, and a field for setting the buzzer frequency.
    - Design a separate group for toggling key features on and off, with individual switches for temperature, humidity, and light intensity. If a user switches off a feature, it should be removed from the main tab.

Figure 2 and Figure 3 show a sample dashboard for the project.
Note: **This does not include all the required features**. This is to get an idea about the dashboard. You are free to choose colors and placements of components.



Days 5

07:00 AM

01:00 PM

08:00 PM

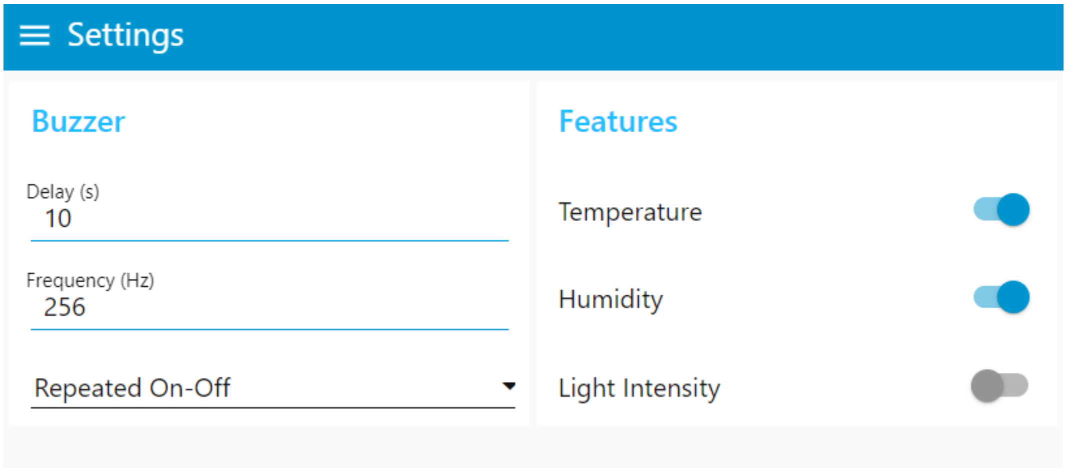
Temperature Min 15

Temperature Max 36

Humidity Min 10

Humidity Max 85

Figure 2: Sample dashboard [Main Tab].



Settings

Buzzer

Delay (s)
10

Frequency (Hz)
256

Repeated On-Off

Features

Temperature

Humidity

Light Intensity

Figure 3: Sample dashboard [Settings Tab].

Figure 4 shows the basic architecture of the project. **Use, test.mosquitto.org/ as the broker**.

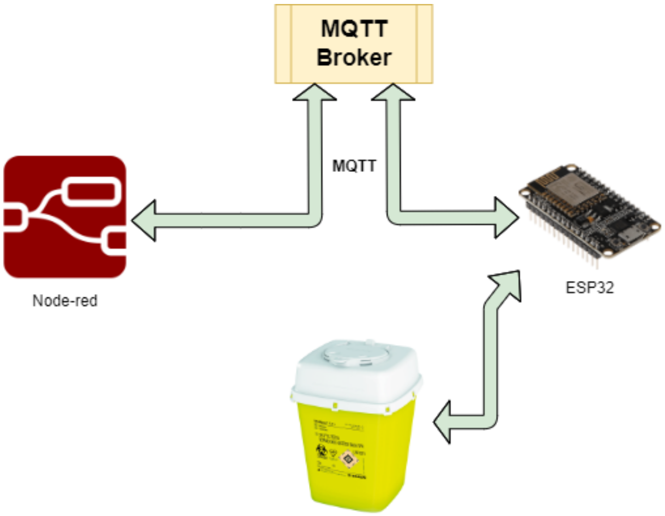


Figure 4: High-level architecture.