



# GROUP 3

## SMART BIKELIGHT

s224025 - Emil Broge Johansen

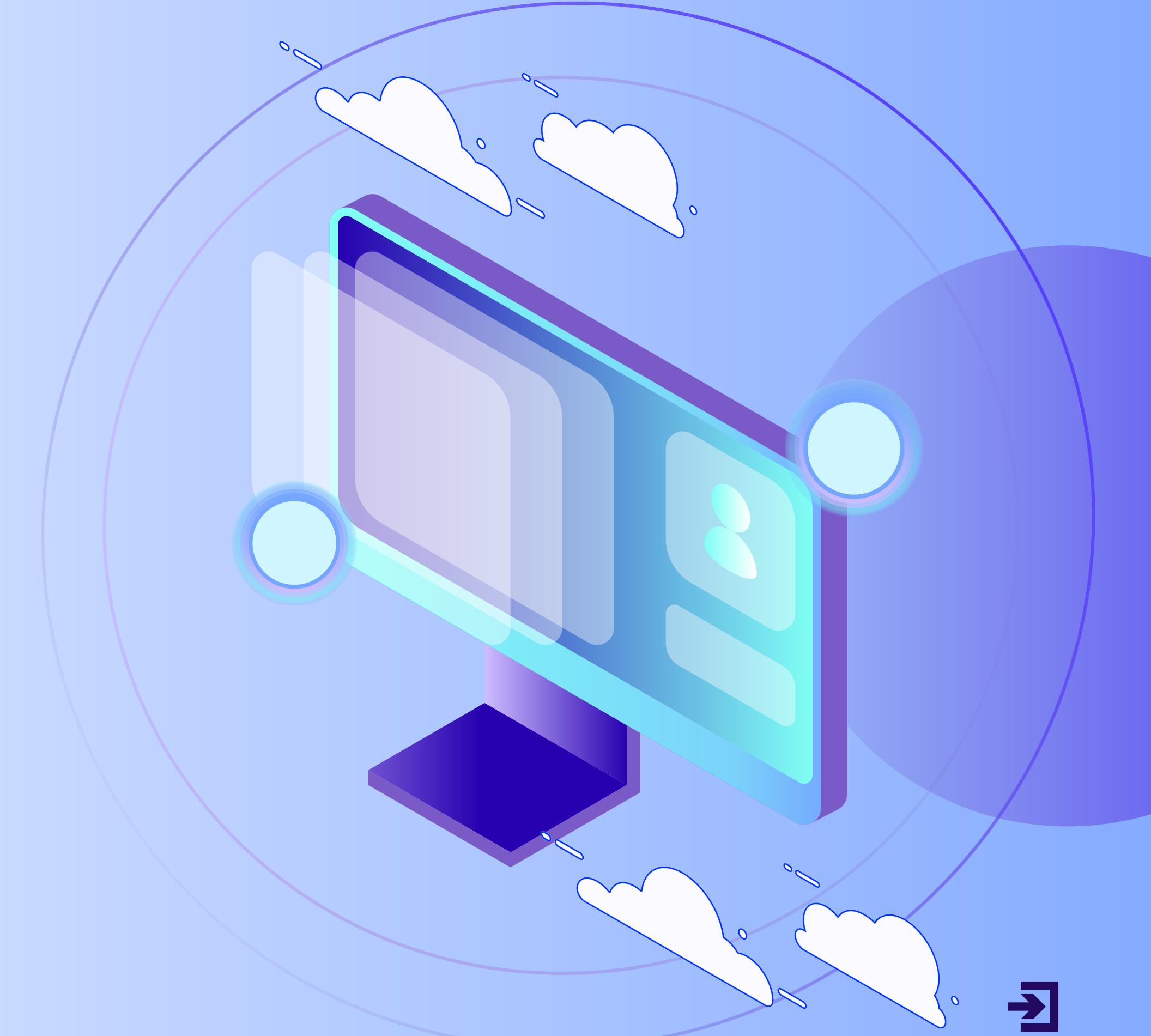
s202935 - Victor Lyhne Jensen

s223987 - Asbjørn Schroll Graae

s061795 - Henrik Bach

s220579 - Jonathan Rolsted Jensen

s243549 - Manuel Imanse



# DEFINITION AND SCOPE

---



AUTOMAICALLY TURN ON IN THE  
DARK...

THEFT PREVENTION!

....A BIKE LIGHT BUT SMARTER ?

SENSORS & WIRELESS  
COMMUNICATION

SHOULD IDEALLY SEEM  
“NORMAL”

...ALL WHILE MAINTANING  
BATTERY LIFE



# HARDWARE OVERVIEW

## Sensing

Accelerometer  
(ADXL345)

Light Sensor  
(LDR)

Push buttons

## ESP32

Microcontroller

GPIO etc.

Battery monitoring  
(ADC)

Charging and regulating  
circuitry 3.3V

Battery (Li-ion) 4.2V

## Connectivity

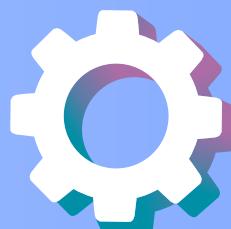
WiFi

LoRa Module  
(RN2483)

(BLE)

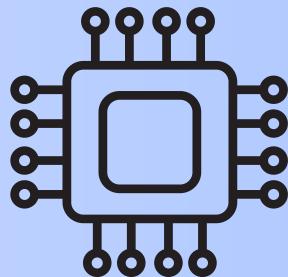
POWER  
LED

Transistor & PWM



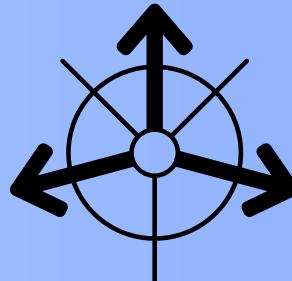
# HARDWARE COMPONENTS

COMPACT, MODULAR, AND POWER-EFFICIENT.



## ESP32 Microcontroller

- CPU managing all operations.
- Dual-core processor, GPIO pins, with Wi-Fi and BLE, UART for LoRaWAN, and deep sleep modes.
- Interfaces with all sensors and modules via I2C, analog, digital, and PWM.



## Motion Sensor

- Accelerometer and gyroscope (3axis) to detect bike movement or orientation changes.
- I2C with an interrupt pin to wake the ESP32 from deep sleep.



## Photoresistor

- Measures ambient light to automatically turn the LED on/off.
- Connected to an analog pin & is read via ADC.



## LED & Brightness Control

- Provides light with adjustable brightness via a transistor
- Connected to a PWM pin (LED\_PIN) for brightness control (0-255 range).

# HARDWARE COMPONENTS



COMPACT, MODULAR, AND POWER-EFFICIENT.



## Battery Monitoring System

- Tracks battery voltage, percentage, and estimated runtime.
- Voltage divider to measure cell voltage.
- (Charging status pin to detect charging state.)
- User inputs to switch modes or wake the device.
- Two buttons with pull-down resistors, managed by the OneButton library for click/long-press detection.



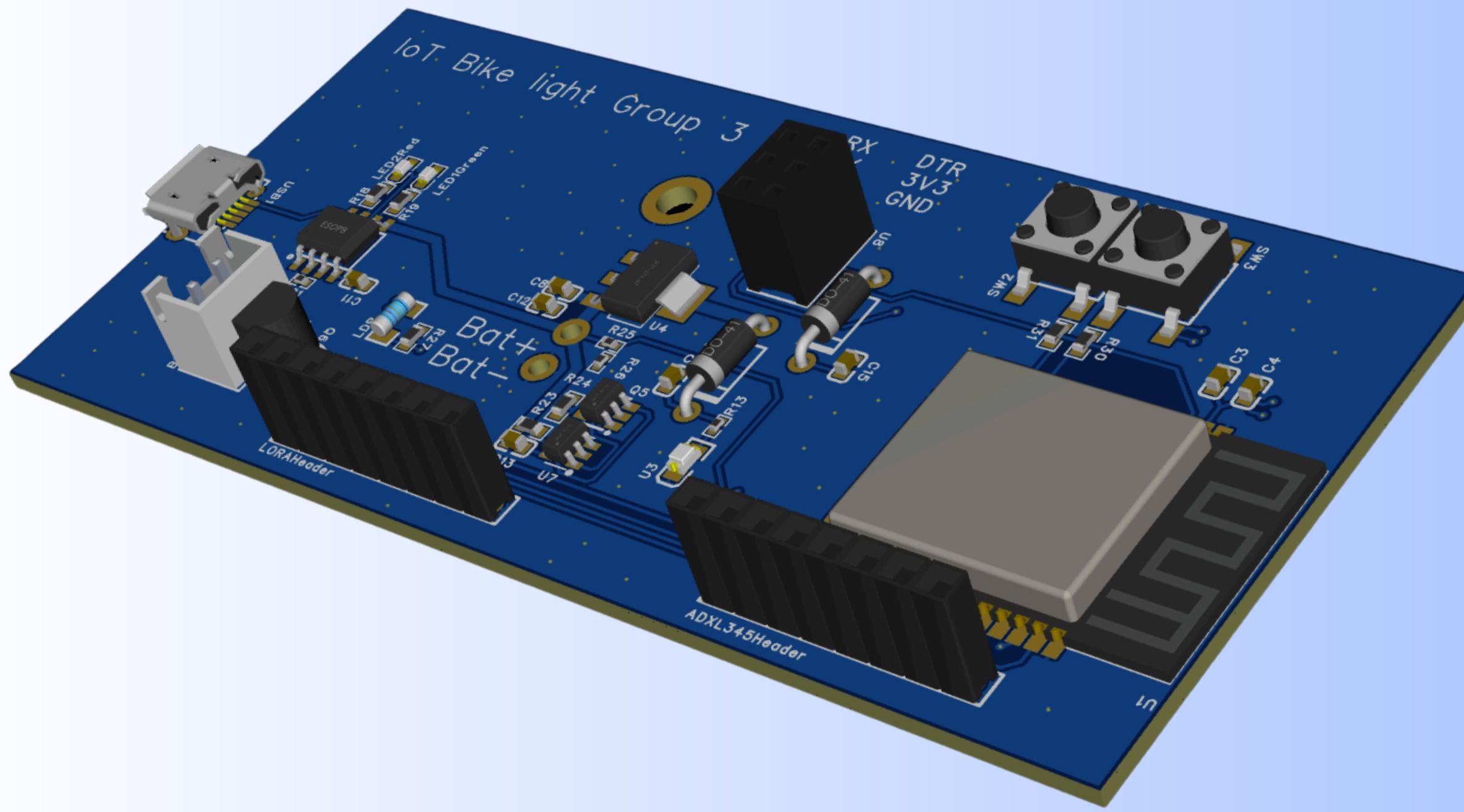
## Buttons



## LoRaWan Module

- Enables long-range, low-power communication for status updates and remote commands.
- UART interface (pins RX, TX), with a reset pin (RST) and wake-up via RX.
- Uses Over-The-Air Activation (OTAA) with APP\_EUI and APP\_KEY for The Things Network (TTN).

# PCB HARDWARE IMPLEMENTATION

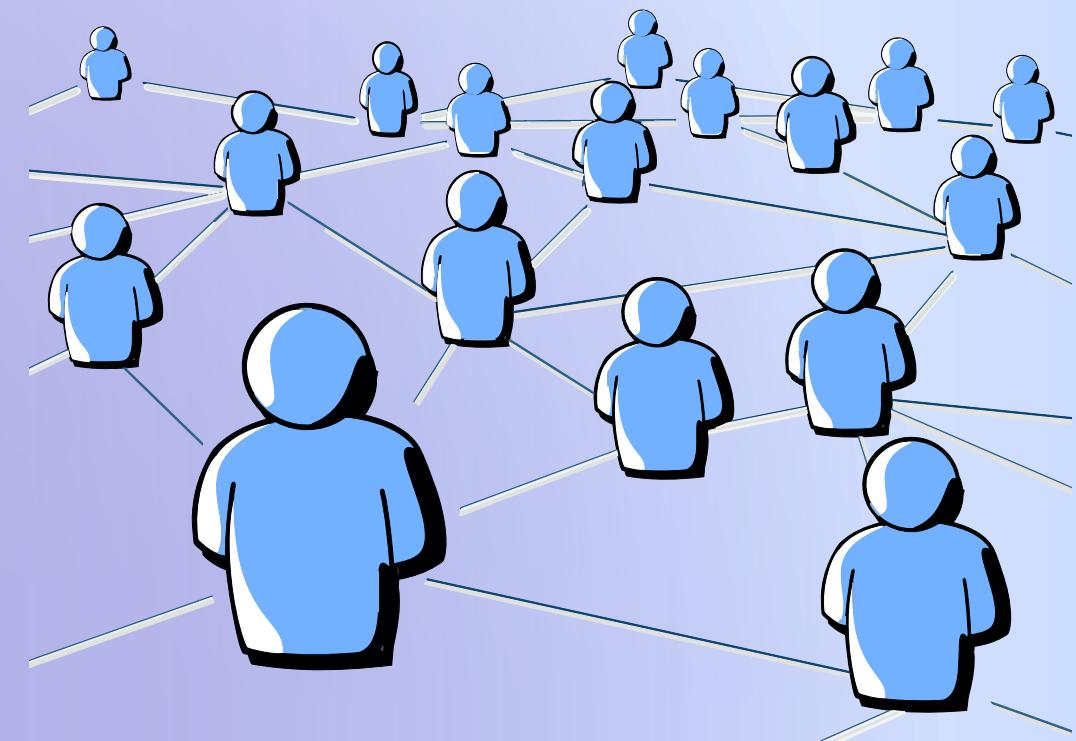




- **LoRaWAN Integration with TTN:** Interfaces with The Things Network (TTN) to handle long-range device updates and commands.
- **MQTT Broker:** Uses a internal broker to manage publish-subscribe communication for real-time data exchange between FE and BE.
- **App Connectivity:** Links with the Flutter Test and -Flow-based front end to deliver updates, controls, and notifications to users.
- **Scalability:** Designed to support multiple devices and growing data demands efficiently.
- **Reliability:** Ensures consistent communication between the bike light device and user interface.



# COMMUNICATION PROTOCOLS



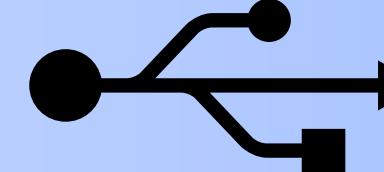
## LoRaWAN:

Sends status (battery, Wi-Fi scan) and receives commands.  
Uses RN2483 LoRa module via UART; supports OTAA join with retries.



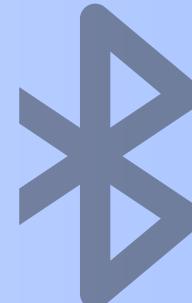
## Wi-Fi

Geo Location; Reports top 3 networks' MAC addresses plus RSSI via LoRaWAN.



## UART / I2C

For communicating with the different module (RN2483 and ADXL345)



## BLE (Bluetooth Low Energy)

Proximity unlocking; scans for 3s, checks RSSI and UUID.



- **Platform:** Built using FlutterFlow, a low-code platform for cross-platform Flutter apps.
- **User Interface:** Features a responsive dashboard for real-time device status.
- **LED Control:** Allows users to toggle the LED on/off and adjust brightness directly from the app.
- **Alerts and Notifications:** Push notifications for critical events like motion detection or low battery.
- **User-Friendly Design:** Ensures accessibility with a lightweight, intuitive interface for cyclists.
- **Real-Time Monitoring:** Provides live updates on device status for seamless interaction.
- **(BLE Integration:** Enables proximity-based unlocking and local control of the bike light via smartphone.)

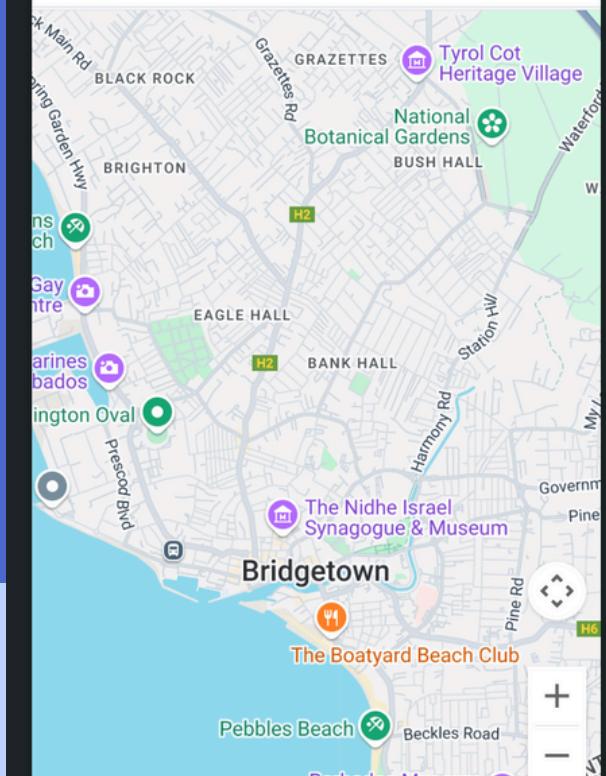


# MOBILE APP

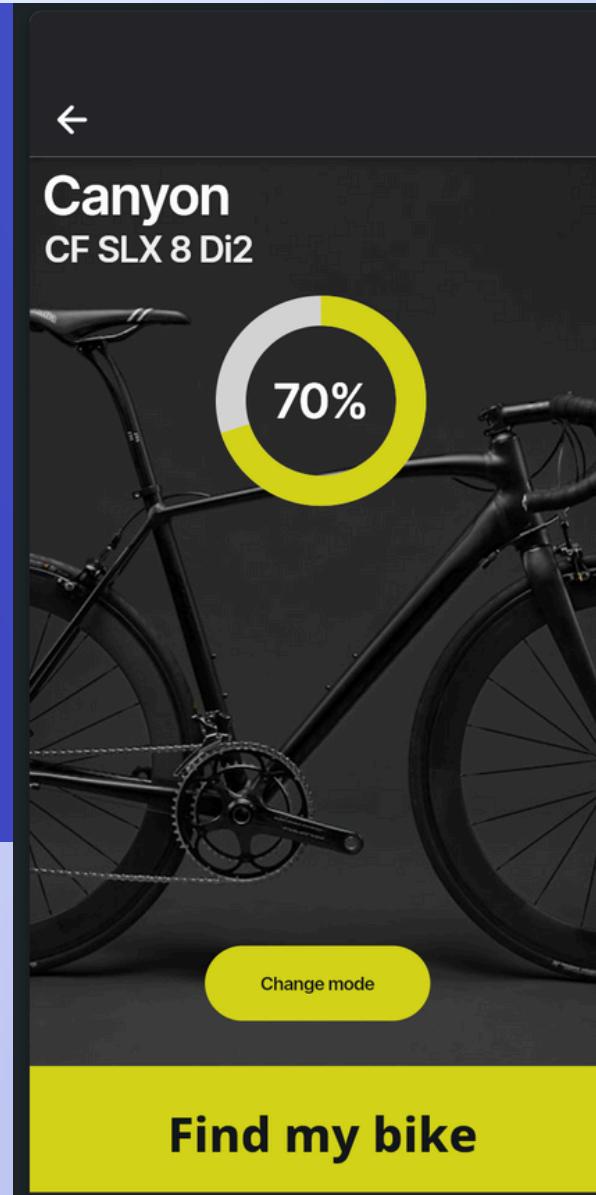


Find My Bike

Last location: Mar 8, 2022



Ping Location



# HOW IT WORKS

## 1. Startup:

- Powers on, initializes in Storage mode, and enters deep sleep.
- Waits for wakeup by buttons

## 2. Mode Switching:

- Button long press → Park mode (Btn1) / Storage Mode (Btn2).
- Motion/Standing still /(BLE unlock) → Toggles Active/Park mode.

## 3. Operation:

- **Active:** LED adjusts to light, monitors battery and motion sensor. No LoRa here for improved responsiveness.
- **Park:** Checks motion, wakes up every interval → scans Wi-Fi, pings LoRaWAN.
- **Storage:** Sleeps, wakes only on button press or every interval → sends battery life.
- **Stolen:** (activated via LoRa): Will wake up every time interval, send location + do an LED warning blinking sequence.

## 4. Communication:

- LoRaWAN for remote control; Wi-Fi for location.

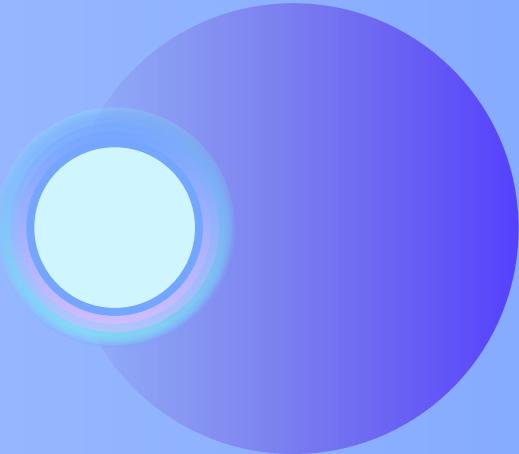
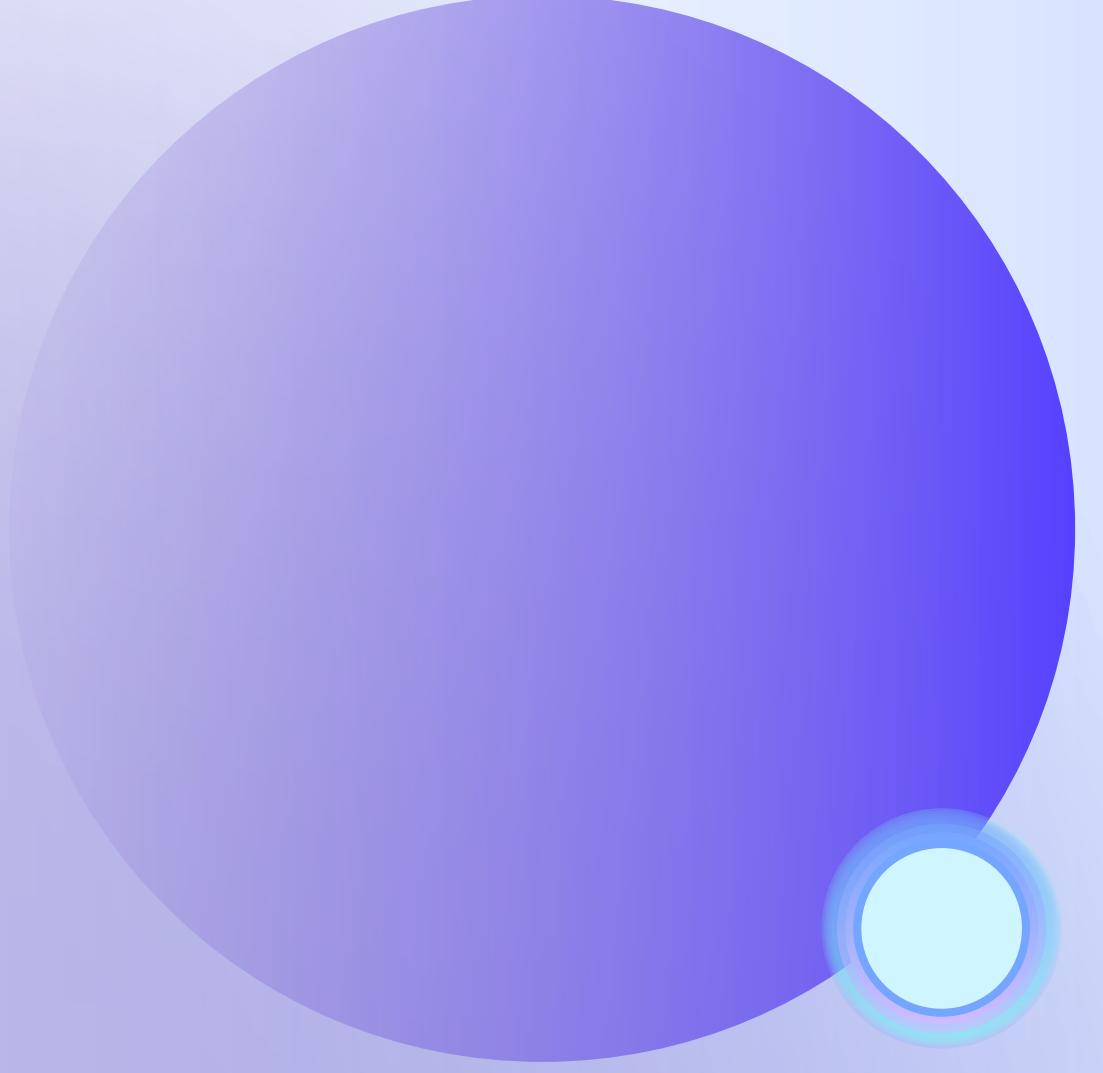
## 5. Shutdown:

- Double click BTN1 or BTN2 to go into permanent sleep mode.
- (Until buttons are pressed, NO timer wakeup).



# FURTHER IMPROVEMENTS

- BLE
- True shutdown of external modules
  - LoRa Module
  - Accelerometer
- Running LoRa communication asynchronously
  - Utilize both cores on ESP
- Implementing App completely
- Better indicator (battery level eg.)
- “Fixing” bad coverage
  - LTE vs. LoRaWAN



**THANK YOU!**