Building a blimp using nano-drones for BLE mesh

1. Problem and Scope

2. System Design

3. Implementation Plan

4. Engineering and Software

1. Problem and Scope
2. System Design
3. Implementation Plan
4. Engineering and Software

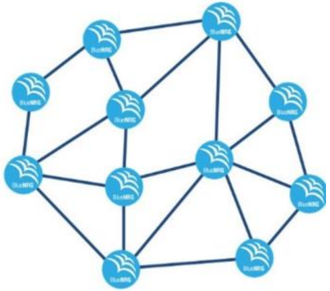
Problem Statement

Objectives:

- Build an indoor blimp using nano drone
- Use it to gather information using BLE
- Coordinate with BLE mesh to:
  - Collect data and communicate with Mesh
  - Link BLE central devices to Mesh

(a) Drone

(b) Blimp Model



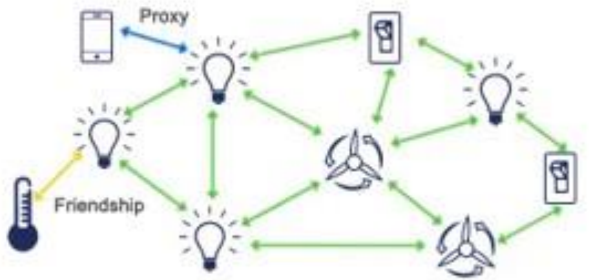
(c) BLE Mesh Example

Challenges:

- Increase on-air time with single charge.
- Coordinating BLE tasks with blimp motion.

Applications and Scope

- Scope
  - Create a BLE Mesh
  - Configure blimp as BLE gateway to Mesh
  - Increase on-air time of blimp
- Applications
  - Inventory Management in warehouses
  - Firmware Over The Air Updates



1. Problem and Scope
2. System Design
3. Implementation Plan
4. Engineering and Software

System Design

- Mesh network (Application)
  - 3 Nodes (1 Proxy Node + 2 Mesh Nodes)
  - Provisioner: Smart phone
  - Blimp (BLE 4.0 radio nRF51822)

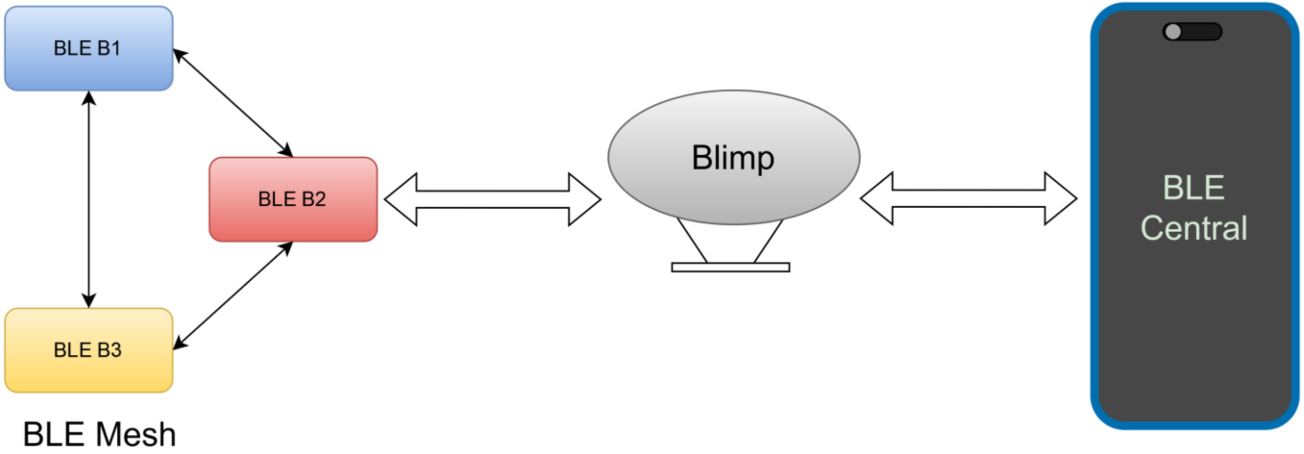


Figure 3: Milestone 1 Application

System Design

Node B1:

- 1 Element with Generic on/off State
- Physical actuation LED

Node B2:

- Proxy node translating MESH messages to classic BLE services and characteristics

Node B3:

- 1 Element with Generic level State
- Physical actuation FAN

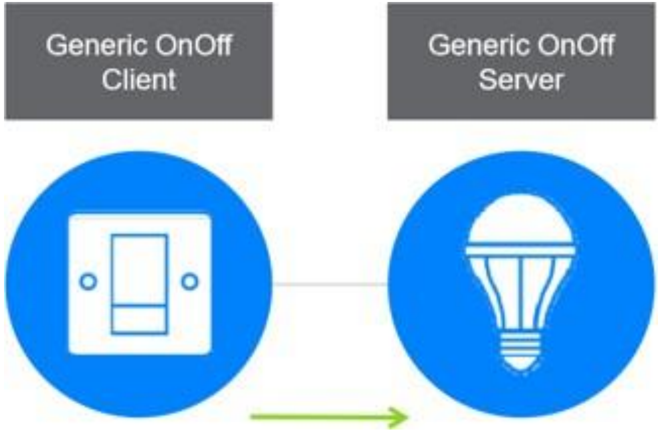


Figure 4: Generic Models

Bluetooth Meshing Challenges

- Crazy Flie 2.1 uses nRF51822
- No support for meshing
- Conventional point-to-point BLE
- Blimp cannot be directly integrated to mesh
- Can communicate with proxy node only
- Alternative - nRF52/ nRF53 family of SoCs

1. Problem and Scope

2. System Design

3. Implementation Plan

4. Engineering and Software

Mesh Implementation Plan

- Node B1 (nRF52840DK):
  - LED light node
  - Proxy function disabled
  - Powered by Li-ion cell
- Node B2 (nRF52832DK):
  - Proxy node link for BLE central
  - Proxy function enabled
  - Powered by wall socket
- Node B3 (nRF52840DK):
  - Fan node with variable speed
  - Proxy function disabled
  - Powered by Li-ion cell
- Development using VSCode, Zephyr and

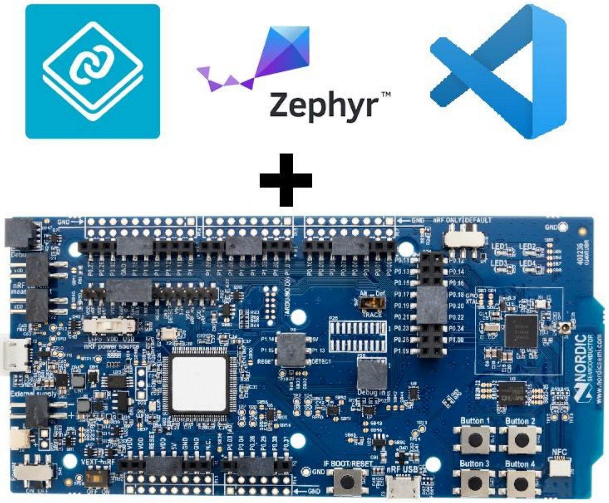


Figure 5: Hardware and Software

Blimp Implementation Plan

- Balloon Materials:
  - Mylar or Melinex (Superior retention)
  - Latex (Higher lift poor retention)
- Buoyant gas:
  - Hydrogen (lift capacity $\approx$ 1.12 gm/litre)
  - Helium (lift capacity $\approx$ 1.02 gm/litre)
  - Helium preferred due to safety concerns
- Nano Drone:
  - Crazy Flie 2.0
  - Total weight 27gm
  - Controller: Android Phone

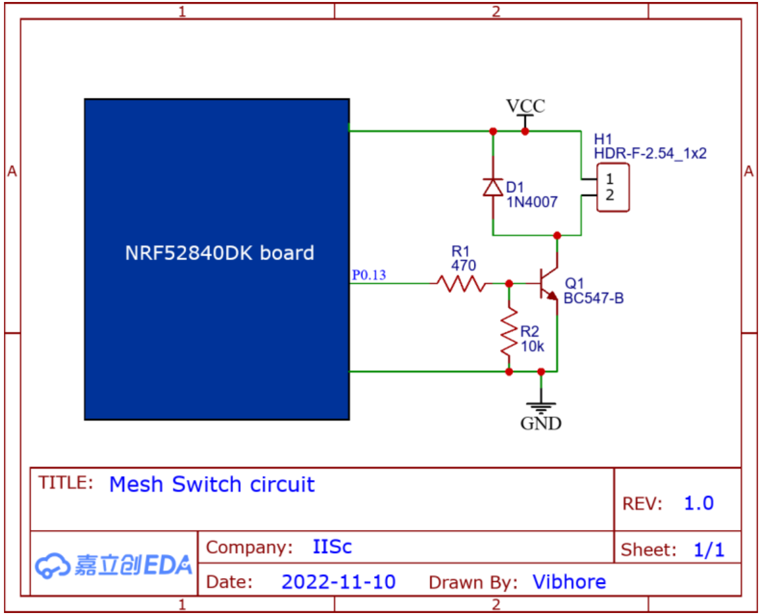
Progress

- Mesh Implementation
  - 3 Nodes Mesh programming ✓
  - Provisioning the nodes into Mesh ✓
  - Communication to mesh using Proxy ✓
  - Implement actuation circuit for nodes
  - Use nRF51822 to send mesh messages
- Blimp Implementation
  - Balloon Material and filler gas ✓
  - Test motor orientation and control
  - Modify BLE radio firmware for mesh

1. Problem and Scope
2. System Design
3. Implementation Plan
4. Engineering and Software

Hardware: Mesh Node

- Fan and Light Control
- Max Current 200 mA
- Diode : 1N4007 fly-back
- GPIO : Port 0 Pin 13



TITLE: Mesh Switch circuit

REV: 1.0

Company: IISc

Sheet: 1/1

Date: 2022-11-10 Drawn By: Vibhore

Figure 6: Switch circuit for high current loads

Hardware : Drone-I

- Controller : Arduino Nano BLE 33
  - Based on nRF52840
  - Only Accelero & Gyro sensors present
- Motor : Coreless 720 motors with 55mm propeller
- MOSFET : AO3400 N-channel MOSFET
- Diode : 1N4007 fly-back
- ToF sensor : VL53L0X
- LDO : TLV71333 3.3V
- Battery : 150 mAh Li-Po 1S

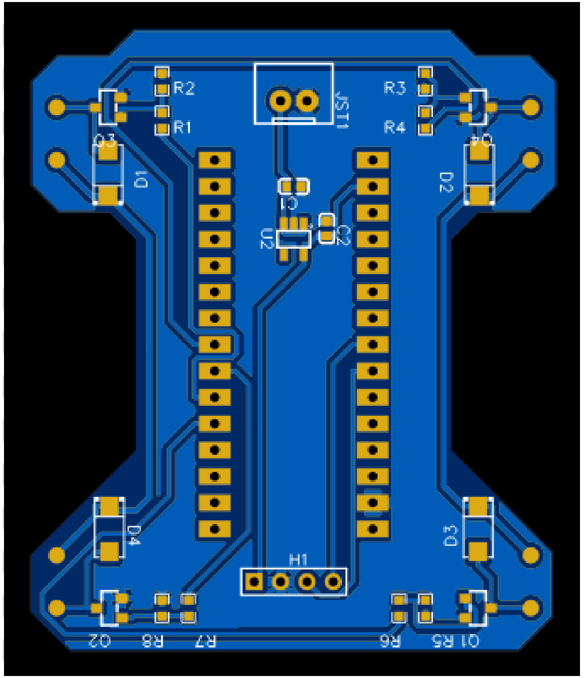


Figure 7: PCB design for drone (Top View)

Hardware : Drone-II

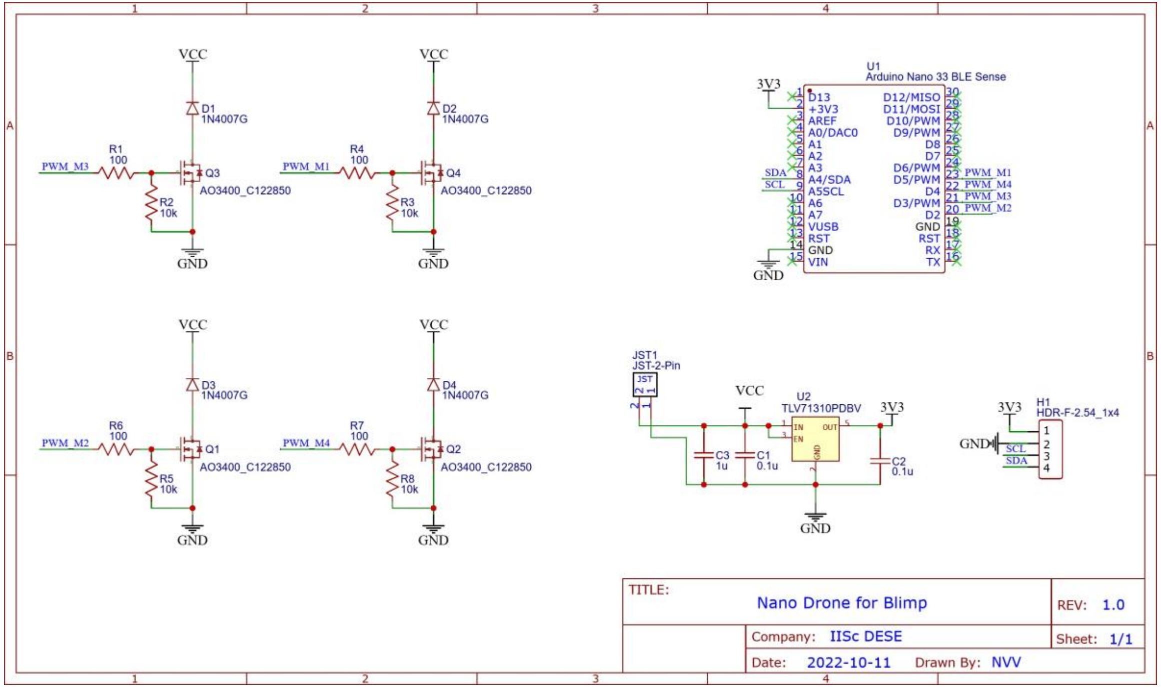


Figure 8: Schematic of Nano Drone Controller

Hardware : Drone-III

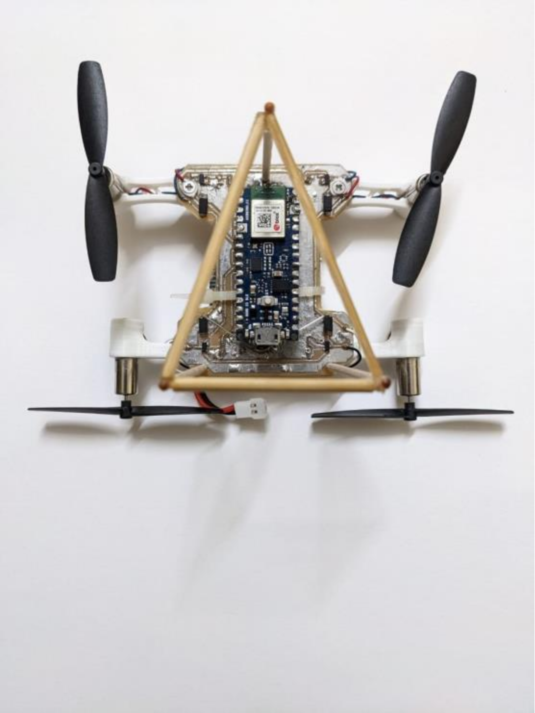


Figure 9: Design for drone (Top View)

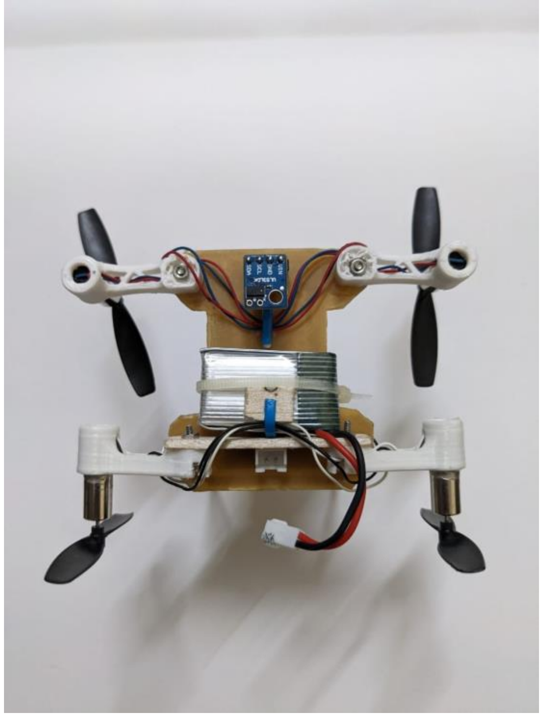


Figure 10: Design for drone (Bottom View)

Software: Mesh

- Boards : nRF52840DK & nRF52832DK
- IDE : VS Code
- Toolchain : nRF connect SDK V2.1.0
- Provisioner : nRF Mesh App

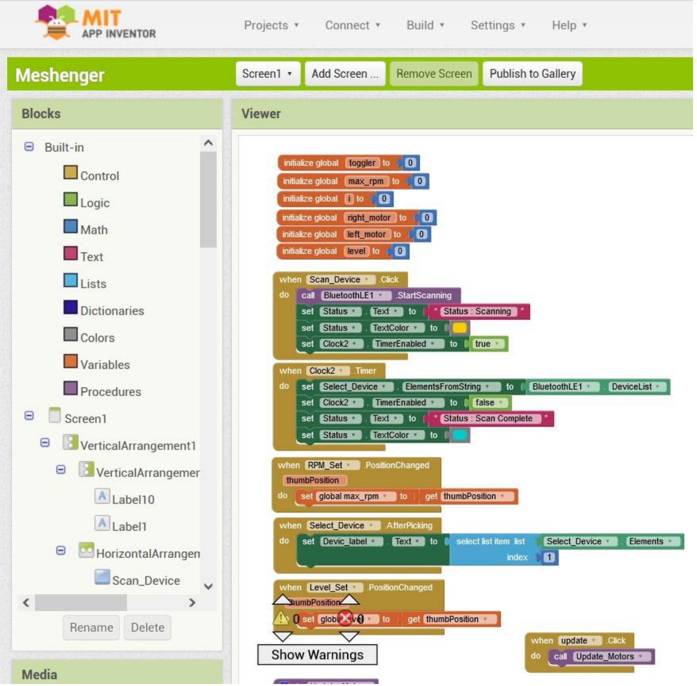
Software: Blimp firmware

- Board : Arduino Nano BLE 33
- IDE : Arduino IDE
- Libraries: Scheduler and Adafruit ToF
- BLE role: Central and Peripheral

Software: Blimp Control App-I

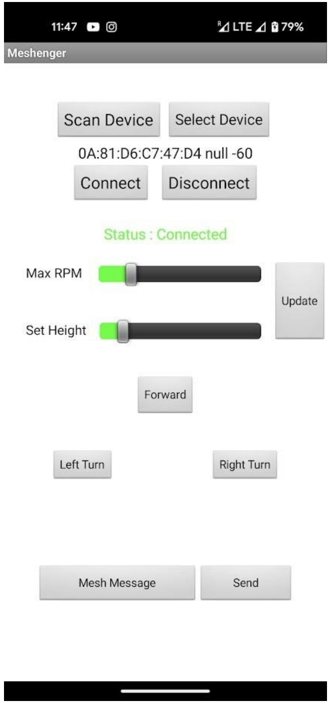
- Android App for Blimp navigation
- Development platform : MIT App Inventor 2
  - BLE connection for Blimp control
  - Mesh message communication

Software: Blimp Control App-II



```
initialize global toggler to 0
initialize global max_rpm to 0
initialize global i to 0
initialize global right_motor to 0
initialize global left_motor to 0
initialize global level to 0

when Scan_Device .Click
do  call BluetoothLE1 .StartScanning
    set Status . Text to " Status : Scanning "
    set Status . TextColor to
    set Clock2 . TimerEnabled to true

when Clock2 .Timer
do  set Select_Device . ElementsFromString to BluetoothLE1 . DeviceList
    set Clock2 . TimerEnabled to false
    set Status . Text to " Status : Scan Complete "
    set Status . TextColor to

when RPM_Set .PositionChanged
  thumbPosition
do  set global max_rpm to get thumbPosition

when Select_Device .AfterPicking
do  set Devic_label . Text to select list item list Select_Device . Elements
                                            index 1

when Level_Set .PositionChanged
  thumbPosition
do  set global level to get thumbPosition

when update .Click
do  call Update_Motors
```

Figure 11: Mit App Inventor



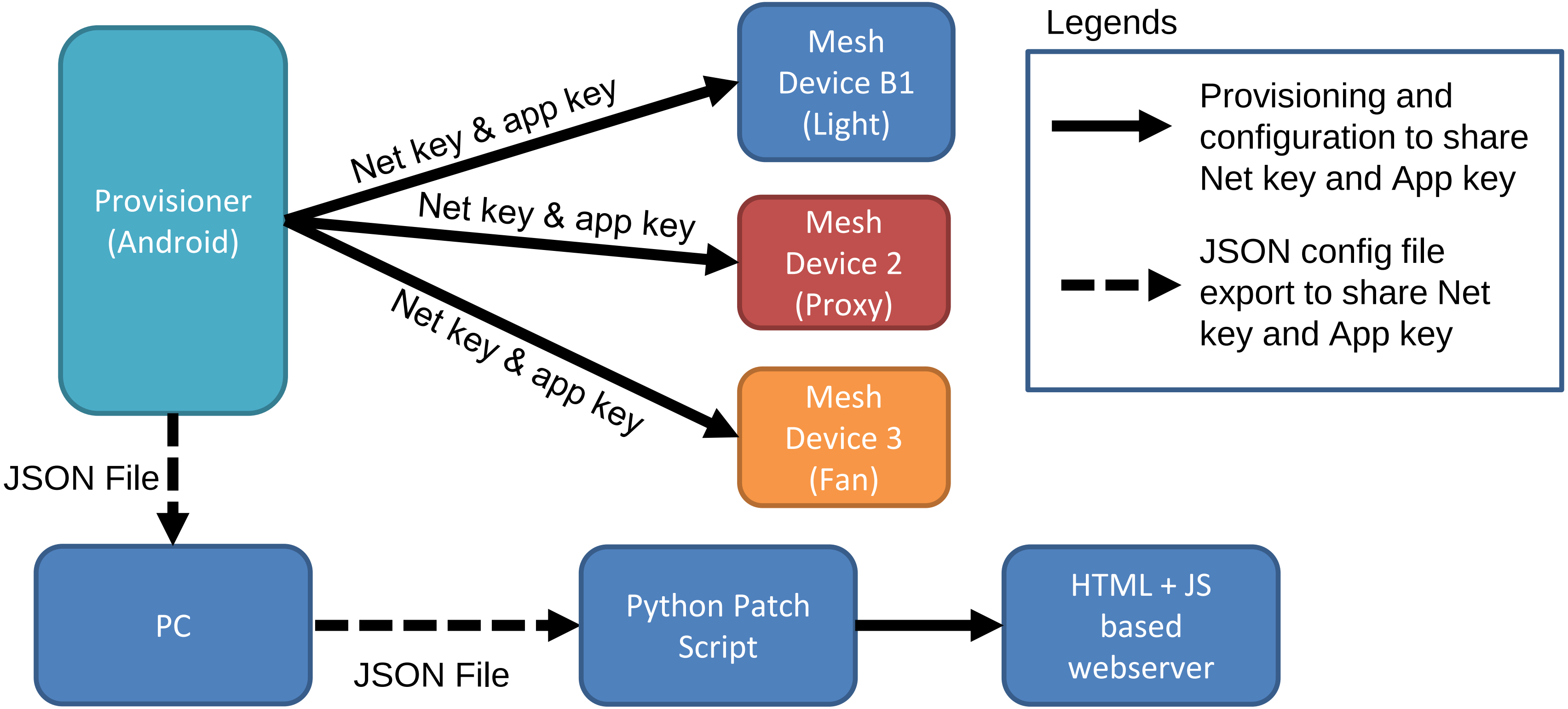
0A:81:D6:C7:47:D4 null -60

Status : Connected

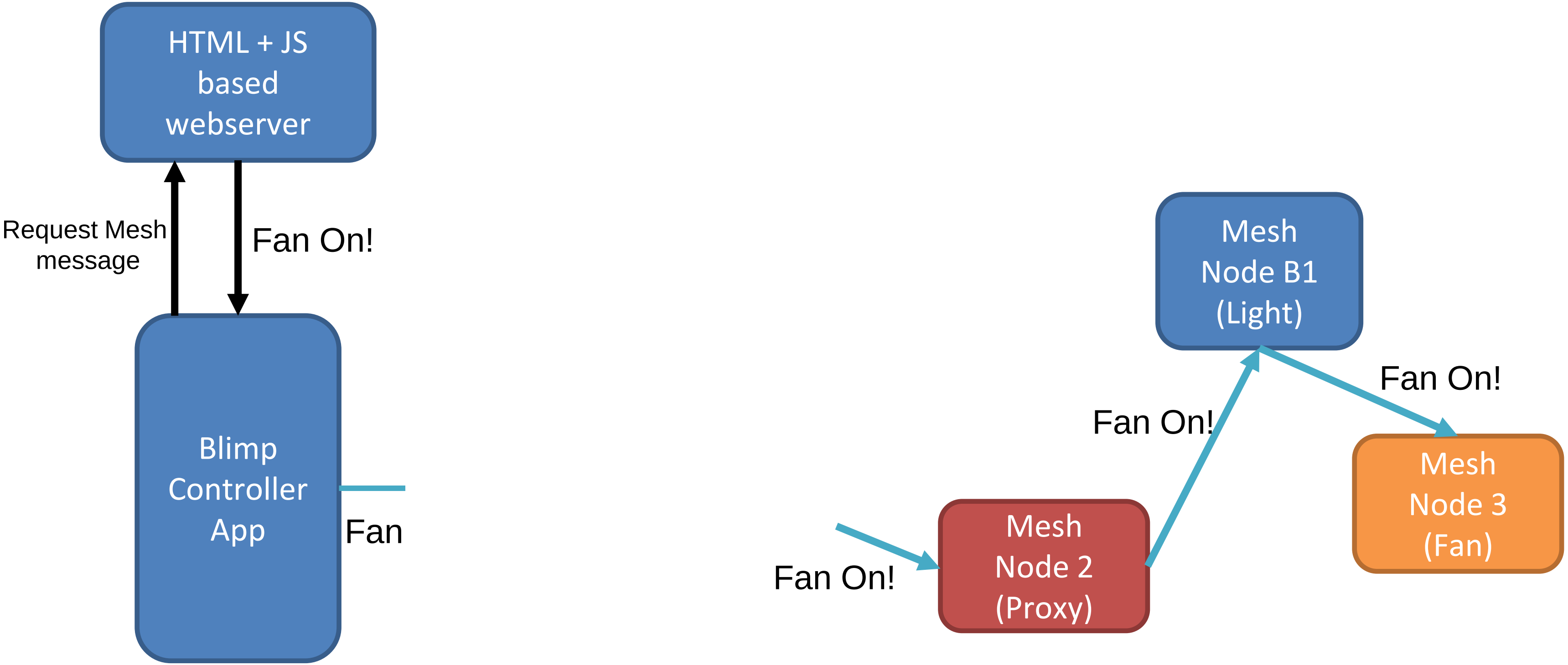
Figure 12: Blimp App

Pre-Demo Activities

Mesh Configuration



Mesh Message Transport



HTML + JS based webserver

Request Mesh message

Fan On!

Blimp Controller App

Fan

Fan On!

Mesh Node 2 (Proxy)

Fan On!

Mesh Node B1 (Light)

Fan On!

Mesh Node 3 (Fan)

Why BLE on a drone?

- We essentially have a BLE post man
- Capabilities inherited from Mesh
- Advantage:
  - Mobility and increased range compared to stationary BLE
  - Maintains low power compared to traditional drones
  - Suitable to “divide and conquer” large meshes
- Why not WiFi/Zigbee?
  - WiFi overheads very large for simple IoT applications
  - Zigbee mesh at best supports 100 devices

Numbers!

Parameter	Value	Unit
Weight	45	gm
Range (LoS)	19	meters
Flight Time*	>10	minutes

*Subject to throttle and blimp buoyancy

Some Application Areas

- Warehouse Inventory Management
  - Dedicated mesh network for each Rack
  - Prevents battery drain due to unnecessary flooding
- Building Automation
  - Configuring BLE meshes in hotels
  - Health Monitoring of nodes
- Mobile Beacon for advertisement
  - Advertising to smartphones in malls

References

- https://devzone.nordicsemi.com
- https://www.bluetooth.com/wp-content/uploads/2019/03/Mesh-Technology-Overview.pdf
- https://www.bluetooth.com/bluetooth-resources/bluetooth-mesh-proxy-kit/
- https://www.bluetooth.com/blog/bluetooth-mesh-developer-study-guide-v2-0/

Acknowledgment

- Prabhakar Sir
- Jamadagni Sir
- Shruti Ma’am
- Lab team
- Centre for Cryogenic Technology

Behind the Scenes