



IUPUI

ECE-53301 - WIRELESS AND MULTIMEDIA COMPUTING



# MONITORING SYSTEM USING BLUETOOTH MESH NETWORK.

By,  
Akash Sunil Gaikwad  
Raghavan Naresh Sarangapani,  
Surya Kollazhi Manghat  
Siddhi Waman

# Introduction

Index

System  
Overview

Hardware  
Used

Software  
Used

- Introduction
- Need of BLE Mesh
- Requirements for the project

Software

Hardware

- Implementation
- Challenges faced
- Result & Conclusion
- Project Demonstration

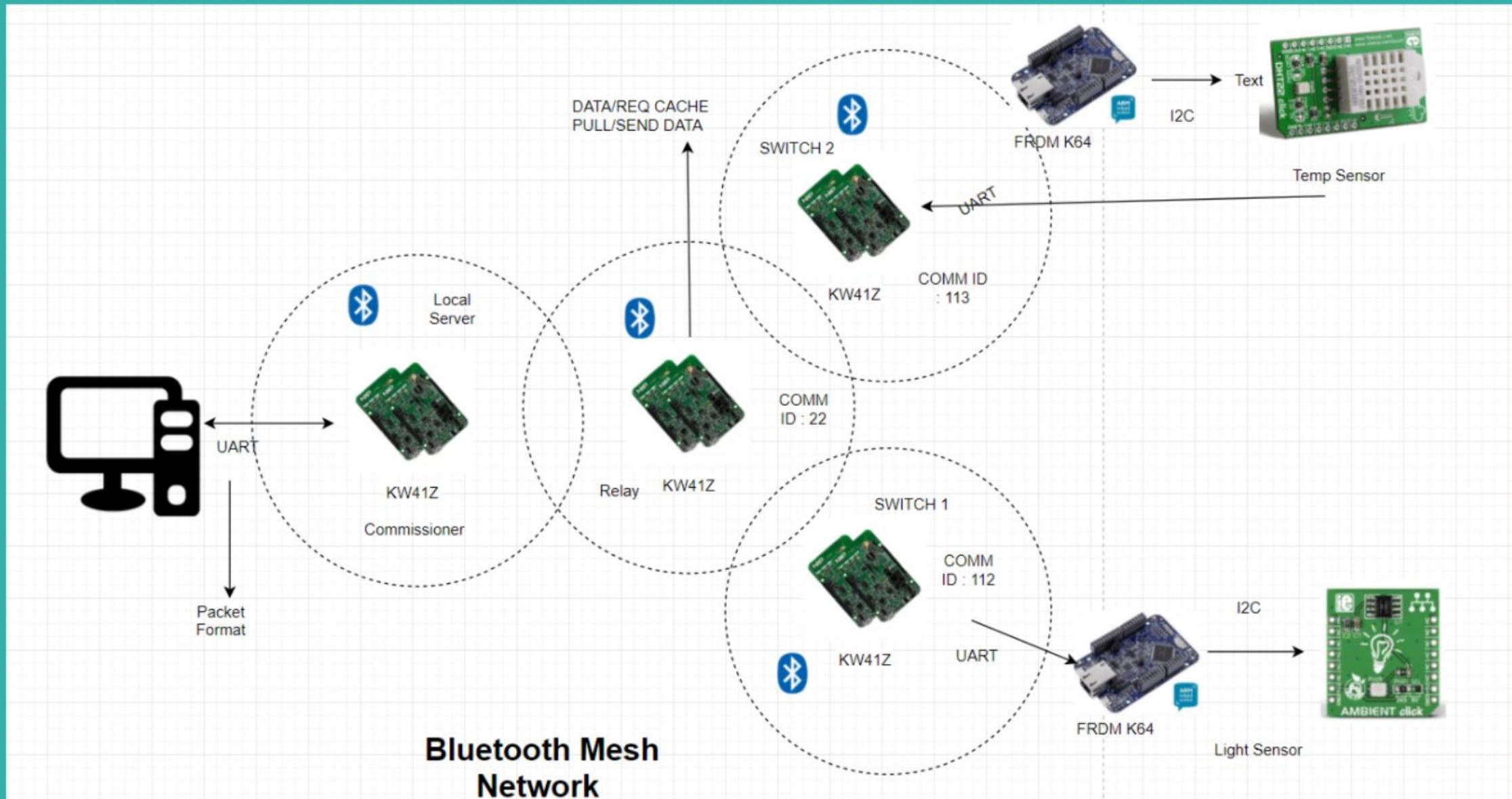


# Overview

- The goal of this project is to create a BLE mesh network to get data from various sensor placed in a wide range which sends the data to local Server for monitoring.
- The project uses the BLE Mesh.
- The data from these sensors will be sent using BLE mesh network of respective KW41Z boards to the destination node (KW41Z board), that is connected to the FRDM K64.
- Values from the temperature and ambient light sensors are updated to the local Server for monitoring.
- Ambient light and DHT uses I2C protocol and the UART protocol is used to transfer data.



# System Overview



# BLE Mesh Network

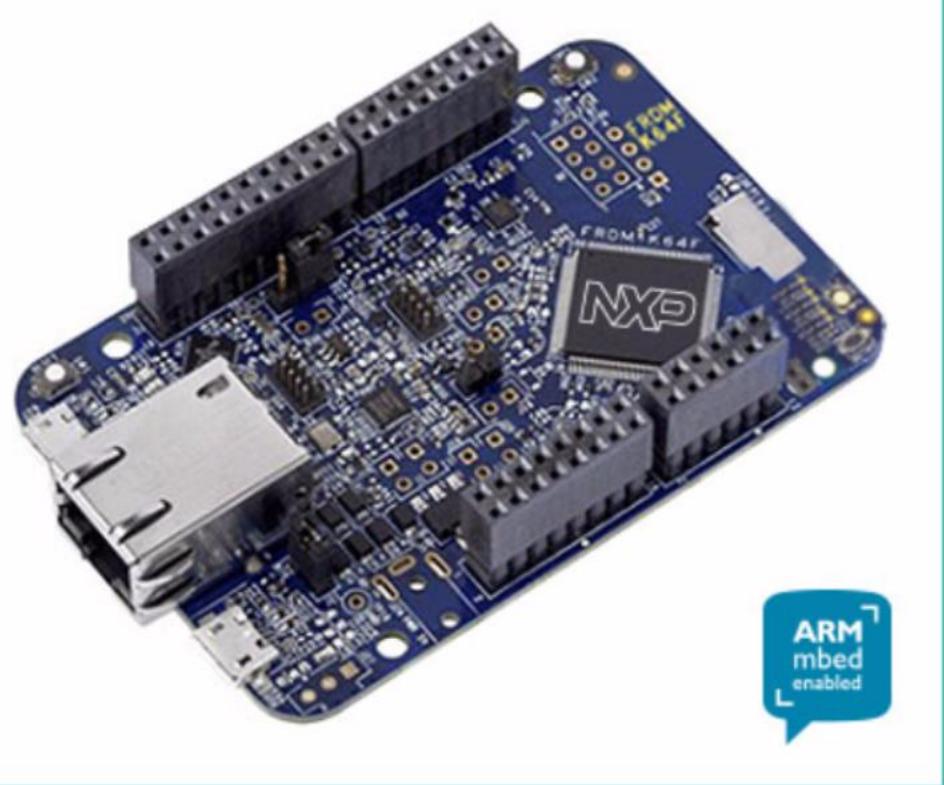


## **Hardware Used**

- **FRDM K64F Dev Board**
- **Temperature and Humidity Click**
- **Light Click**
- **KW41Z boards for mesh network**

**Dev  
Boards**

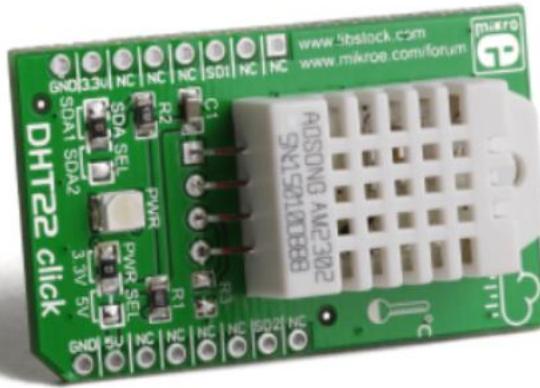
**Clicks**



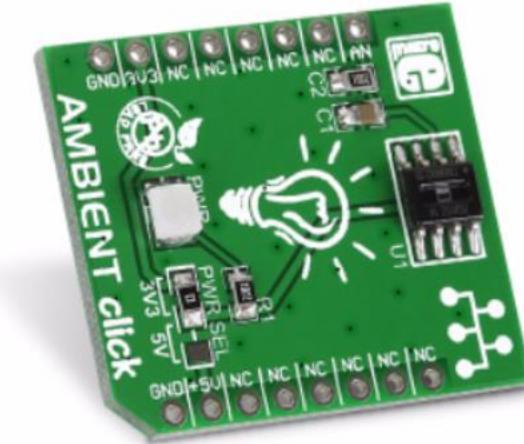
FRDM K64F

KW41Z Dev Board





## DHT22 click



# AMBIENT click

## **Software Used**

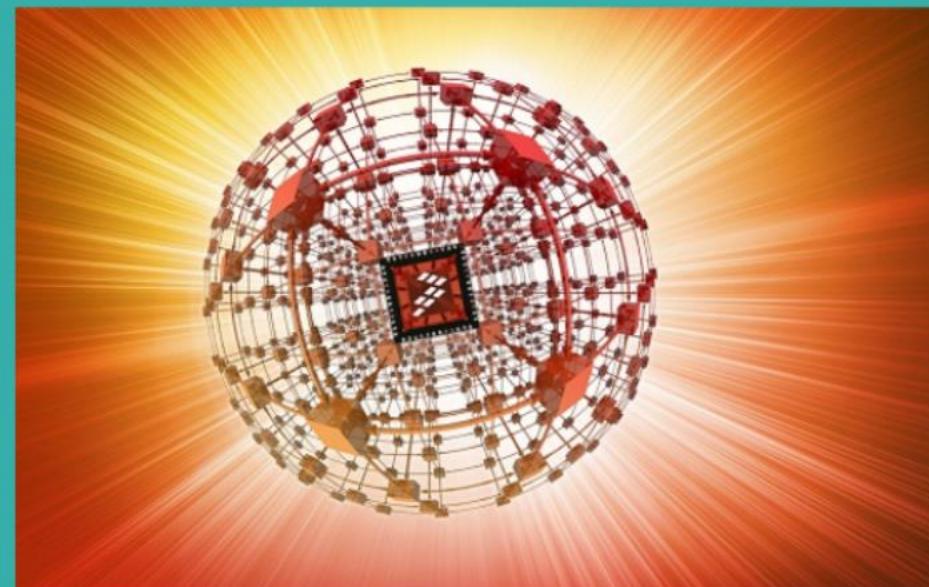
- Mbed Online Compiler
- Kinetis design studio

**Software**





mbed



# **REQUIREMENTS FOR THE PROJECT SOFTWARE & HARDWARE**

**FRDM K64F  
& KW41Z**

**CLICK  
SENSORS**

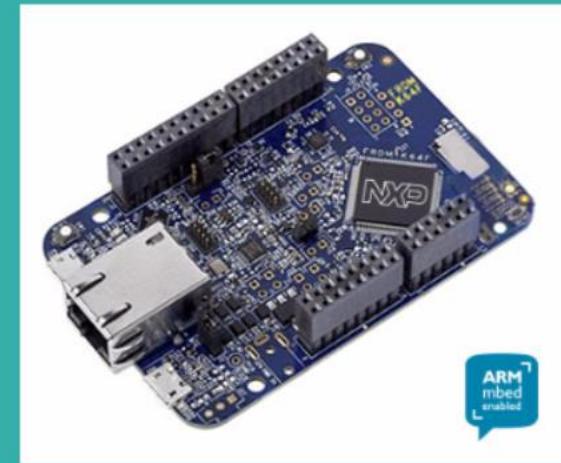
## Section :1

**FRDM K64F & KW41Z**



# FRDM K64F Development board

- The Freedom-K64F is an ultra-low-cost development platform by NXP
- 120 MHz, 1 MB flash memory, 256 KB RAM, low-power.
- Inbuilt accelerometer and magnetometer Sensor
- Flexible power supply option – OpenSDAv2 USB, Kinetis K64 USB, and external source
- **Arduino™ R3** compatible I/O connectors
- Compatible with **OpenSDA**, **Jlink** and **PEMicro** Debugging interface
- microSD Interface
- **SPI, I2C, UART, CAN** protocol and Ethernet



# KW41Z

## Multi-Protocol Radio

- 2.4 GHz Bluetooth Low Energy version 4.2 Compliant
- IEEE Std. 802.15.4 Standard Compliant

## Core and Memories

- Up to 48 MHz Arm® Cortex®-M0+ core
- Up to 512 KB Flash memory
- Up to 128 KB SRAM

## Supported Software

- BLE Host Stack and Profiles
- Generic FSK Link Layer Software
- Thread, 802.15.4 MAC, SMAC
- Thread + BLE Stack (Concurrent operation)
- Kinetis® Software Development Kit (SDK)
- FreeRTOS kernel and a bare-metal non-preemptive task scheduler
- Support for MCUXpresso and IAR



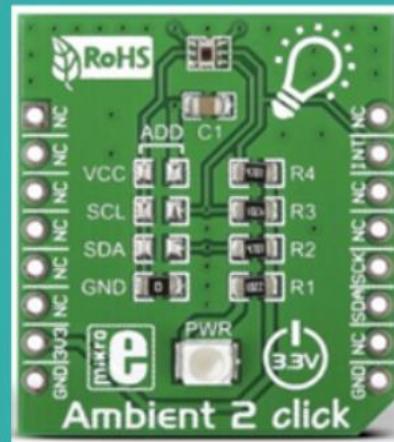
# **CLICK SENSORS**

Ambient Light  
Click Module

DHT22 Click  
Module

# Ambient Light Click Module

- Interface: mikroBUS™ standard I2C interface
- Measurement range: 0.01 lux - 83k lux
- Resolution: 23-bit
- On-board modules: OPT3001 Ambient Light sensor
- Power Supply: 3.3V or 5V
- It filters out 99% of the infrared light spectrum



# DHT22 Click Module



- **DHT22** is temperature and humidity measurement board
- **Interface:** mikroBUS™ standard one-wire interface
- **Measurement range:**
  - **Temperature:** -40°C-80°C with a half a degree precision
  - **Humidity:** 0-100% with 2% accuracy
- **Resolution:** 16-bit
- **On-board modules:** AM2302 temp/humidity sensor
- **Power Supply:** 3.3V or 5V

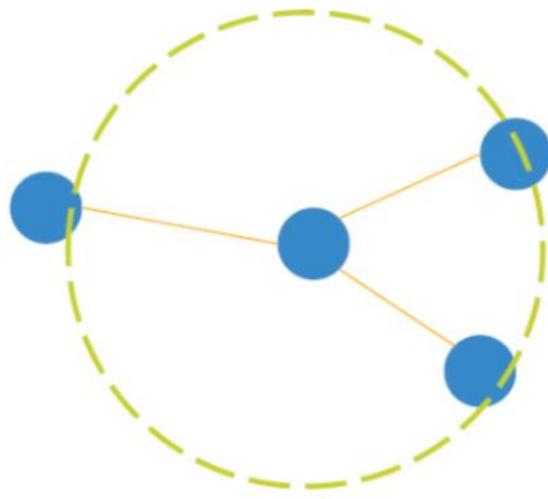
# IMPLEMENTATION OF BLUETOOTH MESH

Difference  
between BLE  
and BLE Mesh

# Difference between BLE and BLE Mesh

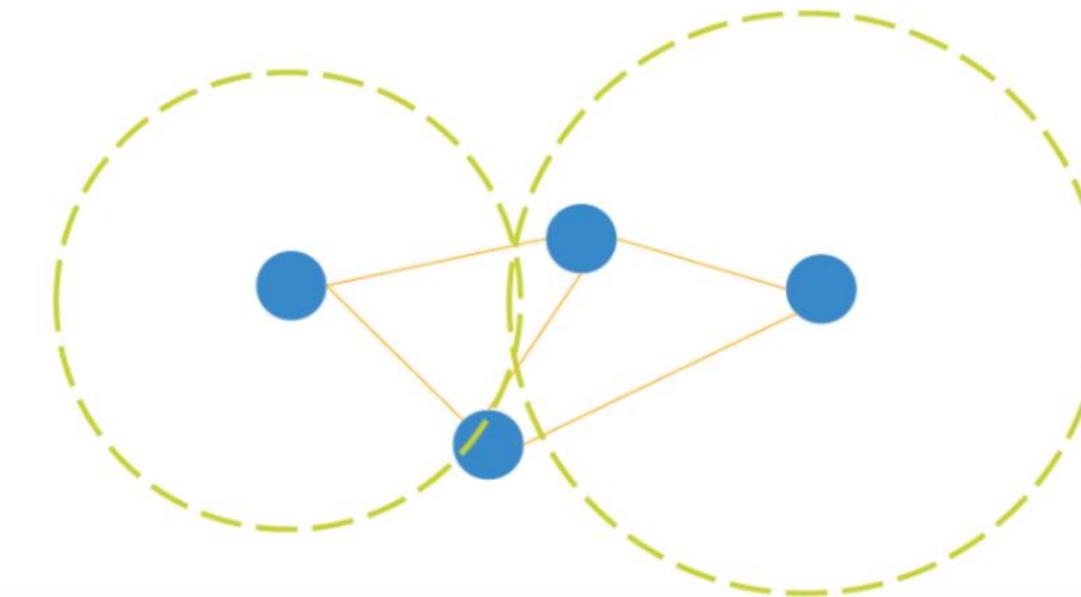
## Bluetooth LE

- **Two states** of a device: advertising and connected
- **Topology**: point-to-point or at best star
- **Range**: line of sight: protocol allows a datagram to be exchanged by two neighbor devices



## Bluetooth LE Mesh

- **A single state** of a device: advertising
- **Topology**: flooded mesh
- **Range**: configurable: protocol allows datagram to be relayed as many times as desired



The Bluetooth Mesh Network defines two topological roles:

- Leaf Nodes



Leaf

- Nodes that **never relay** packets, but only generate and/or process them
- Typically these are **battery-powered** devices that require an association with a Relay Node to be able to effectively participate in the network
- Examples: an energy-harvesting switch, a temperature sensor

- Relay Nodes



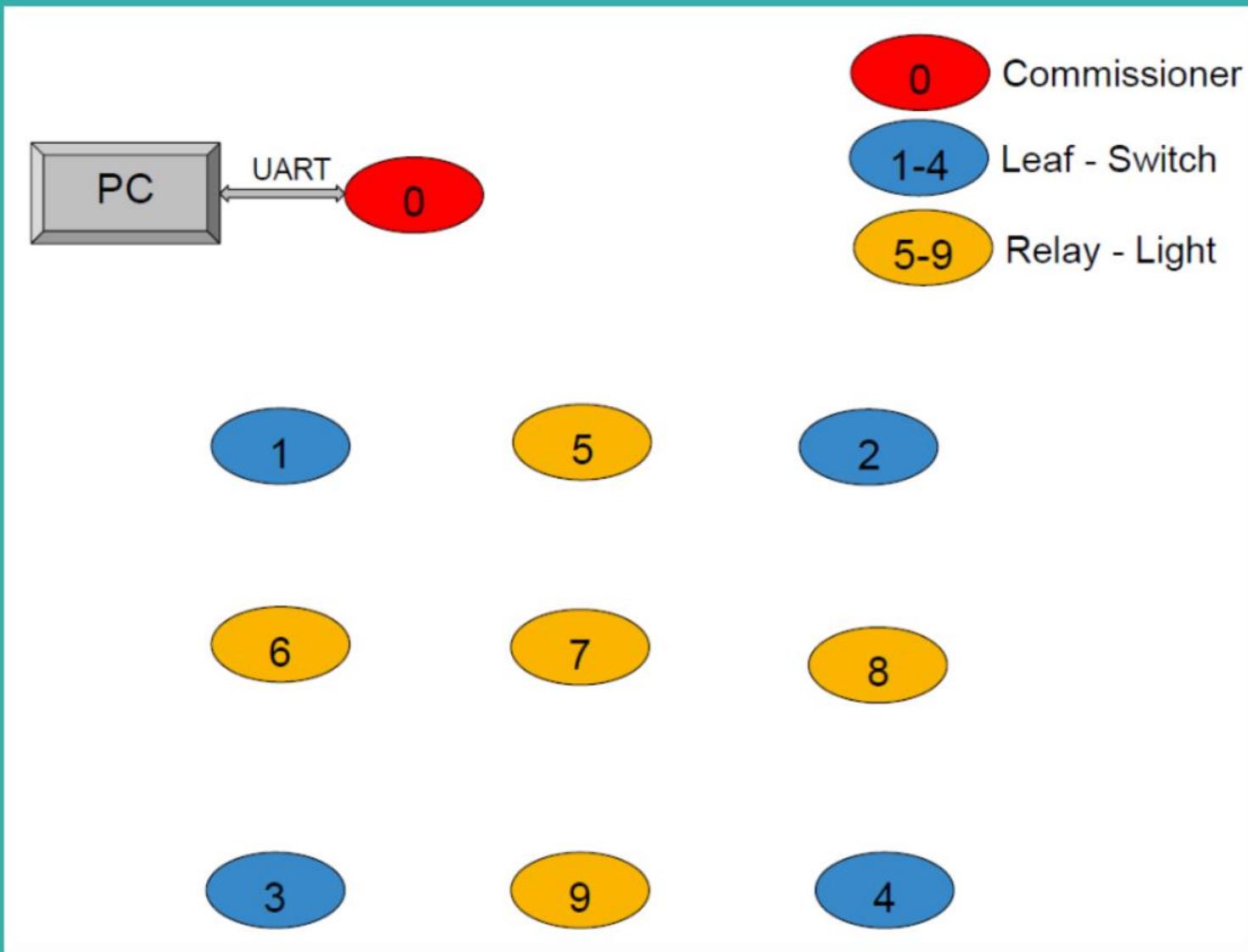
Relay

- Nodes that, in addition to generation and consumption of packets, also perform the **retransmissions** required by the mesh network
- Typically there are **mains-powered** devices
- Examples: a light-bulb, a power socket

# Mesh Commissioning

- *Commissioning* is the process of bringing a new device into the Bluetooth Mesh Network
- A device that is capable of commissioning a new device in the network is called *Commissioner Node* or, simply, *Commissioner*
- There can be multiple Commissioners in a certain Network
- Commissioning is a process performed in two phases:
  1. **Commissioning Link Establishment**
    - Consists of creating a **secure** link between the Commissioner and the new device
  2. **Network Credentials Distribution**
    - Consists of the distribution of the network security credentials from the Commissioner to the new device over the secure link

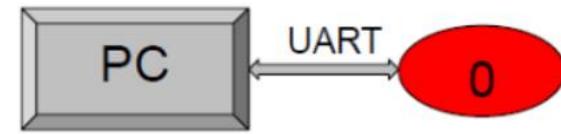
# Commissioning example



# Configuring Network

## Network configuration using terminal

- ***help***
  - List of commands
- ***status***
  - Get network status of device (ID, TTL, relay state)
- ***pub get/set***
  - Get/set Publish address for a Switch
- ***sub get/add/rem***
  - Get/edit Subscription List for a Light
- ***rel get/set***
  - Get/set Relay State
- ***ttl get/set***
  - Get/set TTL



- Terminal commands are translated into Mesh Configuration Model commands that are sent over-the-air by the Commissioner towards Mesh Nodes
- When Commissioner receives status messages over-the-air, it sends logs to the terminal

# Implementation Steps

- The two sensors Temperature and Light are initialized for transferring data.
- The Ambient light sensor & DHT uses the I2C protocol which is used to get the data from the sensor.
- UART protocol is used to transfer the data from sensors to KW41Z.
- Create BLE Network.
- Each KW41Z is assigned a unique ID.
- All the devices are subscribed(relay) and publish(leaf node) to the same multicast address.
- Data from Leaf nodes to Relay node (KW41Z) is sent using custom BLE.

# Implementations Steps

- The data is stored in Cache memory of the Relay.
- When requested the data is transferred to the Commissioner.



# RESULTS & CHALLENGES

Results

Challenges

# Serial terminal Value

# Commissioner

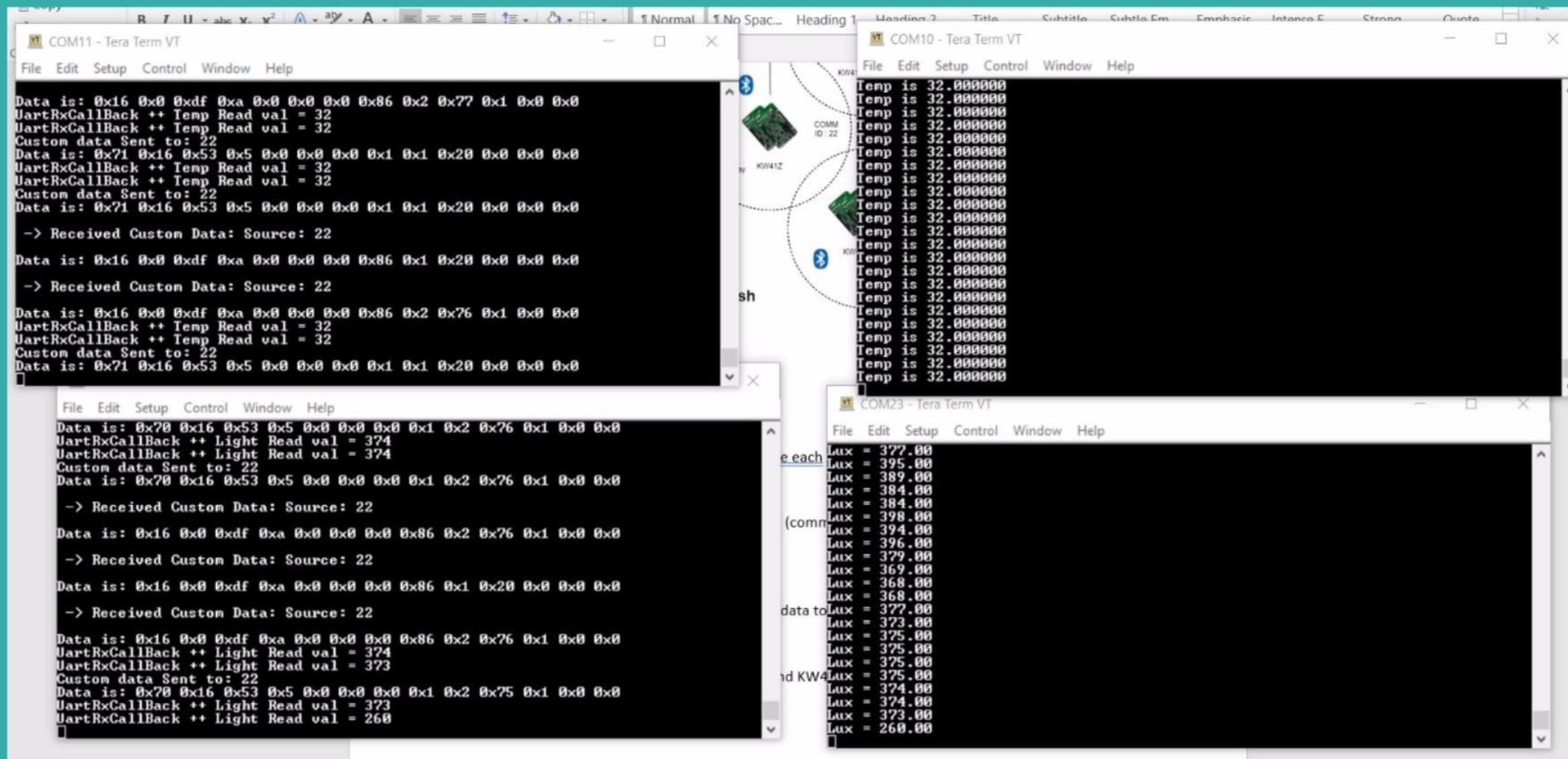
# Relay Node

```
esh_commissioner_frdmkw41z * Private functions fsl_os_abstraction.h

COM7:115200baud - Tera Term VT
File Edit Setup Control Window Help
BLE MESH >>> Received Temp is: 32
Received Light is: 384
Received Temp is: 32
Received Light is: 396
Received Temp is: 32
Received Temp is: 32
Received Light is: 368
Received Light is: 373
Received Temp is: 32
Received Light is: 375
Received Temp is: 32
Received Light is: 375
Received Temp is: 32
Received Light is: 374
Received Temp is: 32
Received Light is: 260
Received Temp is: 32
Received Light is: 274
Received Temp is: 32
Received Light is: 274
Received Temp is: 32
Received Light is: 260
Received Temp is: 32
Received Temp is: 32
Received Temp is: 32
***** osObjectAlloc.c *****
<
#454 osObjectAlloc.c
Data is: 0x16 0x16 0x53 0x5 0x0 0x0 0x0 0x1 0x1 0x20 0x0 0x0 0x0
Received Temp val is: 32 , Light val is: 264
Data is: 0x71 0x16 0x53 0x5 0x0 0x0 0x0 0x1 0x1 0x20 0x0 0x0 0x0
Received Temp val is: 32 , Light val is: 264
Custom data Sent to: 255
Custom data Sent to: 255
Data is: 0x16 0x0 0xdf 0xa 0x0 0x0 0x86 0x2 0x8 0x1 0x0 0x0
Data is: 0x70 0x16 0x53 0x5 0x0 0x0 0x0 0x1 0x2 0x7 0x1 0x0 0x0
Received Temp val is: 32 , Light val is: 263
Data is: 0x71 0x16 0x53 0x5 0x0 0x0 0x0 0x1 0x1 0x20 0x0 0x0 0x0
Received Temp val is: 32 , Light val is: 263
Data is: 0x70 0x16 0x53 0x5 0x0 0x0 0x0 0x1 0x2 0x14 0x1 0x0 0x0
Received Temp val is: 32 , Light val is: 276
Data is: 0x71 0x16 0x53 0x5 0x0 0x0 0x0 0x1 0x1 0x20 0x0 0x0 0x0
Received Temp val is: 32 , Light val is: 276
Custom data Sent to: 255
Custom data Sent to: 255
Data is: 0x16 0x0 0xdf 0xa 0x0 0x0 0x86 0x2 0x14 0x1 0x0 0x0
Data is: 0x70 0x16 0x53 0x5 0x0 0x0 0x0 0x1 0x2 0x91 0x1 0x0 0x0
Received Temp val is: 32 , Light val is: 401
Data is: 0x71 0x16 0x53 0x5 0x0 0x0 0x0 0x1 0x1 0x20 0x0 0x0 0x0
Received Temp val is: 32 , Light val is: 401
Data is: 0x70 0x16 0x53 0x5 0x0 0x0 0x0 0x1 0x2 0x8b 0x1 0x0 0x0
Received Temp val is: 32 , Light val is: 395
Data is: 0x71 0x16 0x53 0x5 0x0 0x0 0x0 0x1 0x1 0x20 0x0 0x0 0x0
Received Temp val is: 32 , Light val is: 395
Custom data Sent to: 255
Custom data Sent to: 255
Data is: 0x16 0x0 0xdf 0xa 0x0 0x0 0x86 0x2 0x8b 0x1 0x0 0x0
```

# Leaf Node

# FRDM K64F



# Challenges

- Experimenting with Temperature GATT profile.
- Switching to custom commands, implementing packet format through the network.
- Synchronizing data transfer through network.
- Alternating between Unicast, Multicast and Broadcast data transfer.



# **SUMMARY & FUTURE WORK**

**Summary of  
The Project**

**Future  
Enhancement**

**Demo**

# Summary of The Project

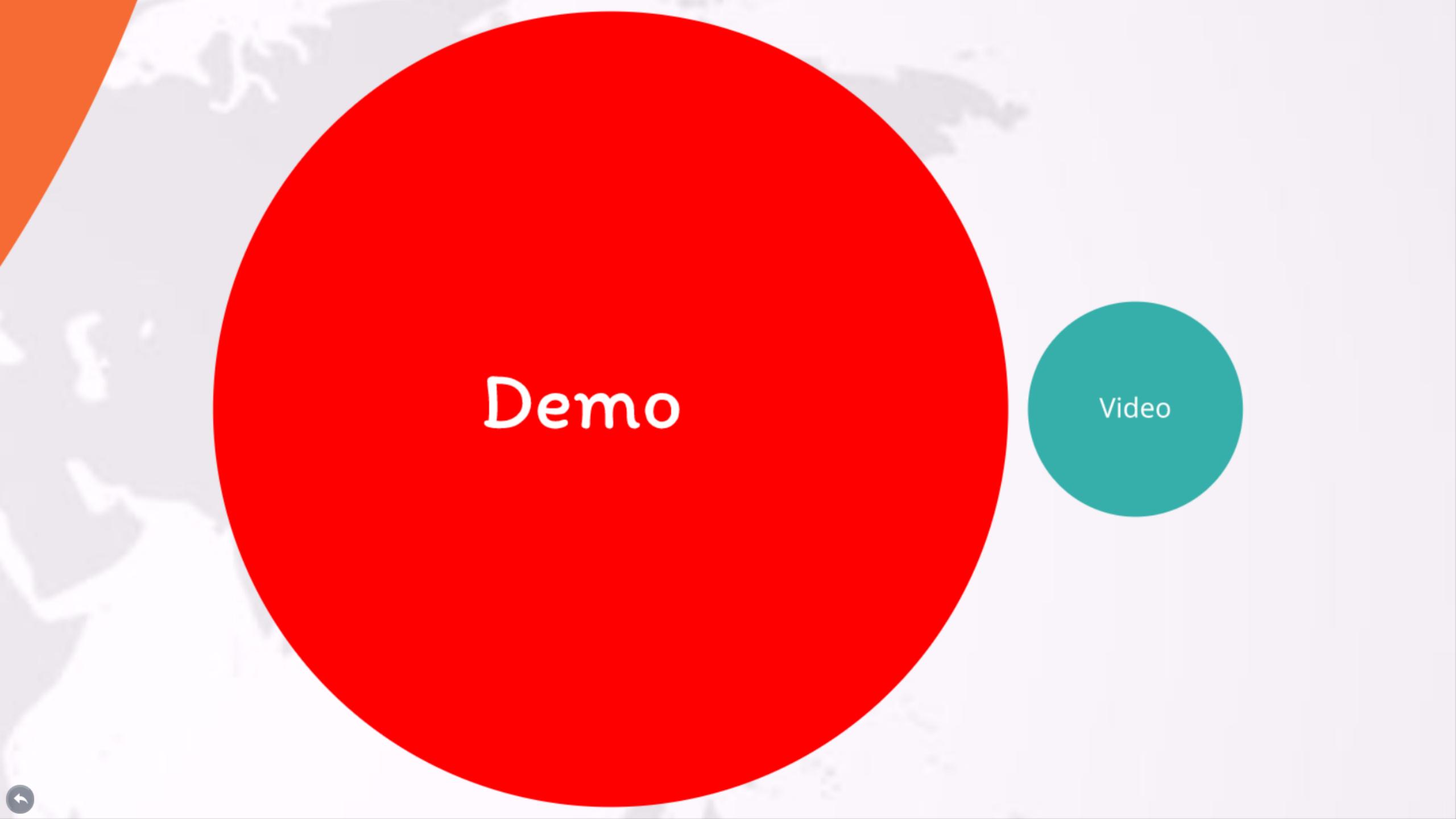
- Using BLE Mesh, we have successfully created the mesh network of MCU's which collects the data from the DHT and Ambient light sensors, and this data was sent to the local server.
- Values from the DHT and ambient light sensors are sampled regularly and the same is updated to the Local Server.
- Successfully created BLE mesh Network
- Setup the communication between FRDM K64F and KW41Z



# Future Enhancement

- Adding Friend Node to support the Low Power feature to operate efficiently, by storing messages
- IP addressing in the BLE Mesh is a natural step.
- Enhanced hardware for increasing the range of the each device.
- Send the local data to the cloud monitoring using any gateway ex. AT&T cellular shield.





Demo

Video

Please find the Demo video link below

<https://youtu.be/2JyAp0eowC0>

# Question Time?



# **Thank You !!!**

