Monitoring System Using Bluetooth Mesh Network

Akash Sunil Gaikwad (2000116696)
Raghavan Naresh Sarangapani (0003440222)
Siddhi Waman (2000235669)
Surya Kollazhi Manghat (2000257775)

ECE-53301 – Wireless and Multimedia Computing

Indiana University – Purdue University Indianapolis.
420 University Blvd, Indianapolis, IN 46202
Year: Fall 2017

PROJECT REPORT

Monitoring System Using Bluetooth Mesh Network

BY
A.S. Gaikwad
R.N. Sarangapani
S. Waman
K.M. Surya

Indiana University – Purdue University Indianapolis.
Year: Fall 2017

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature:

Name: Akash Sunil Gaikwad (2000116696)

Signature:

Name: Raghavan Naresh Sarangapani (0003440222)

Signature:

Name: Siddhi Waman (2000235669)

Signature:

Name: Surya Kollazhi Manghat (2000257775)

Date: 12/06/2017

ACKNOWLEDGEMENT

We would like to express our gratitude and appreciation to all those who gave us the possibility to complete this report. A special thanks to our Project Guide, Professor Mohamed El-Sharkawy, whose help, stimulating suggestions and encouragement, helped us to coordinate our project especially in writing this report and given us his full effort in guiding the team in achieving the goal as well as his encouragement to maintain our progress in track.

We would also like to acknowledge with much appreciation of the crucial role of the University library and University Labs, which gave the permission to use all required resources to complete our project.

CONTENTS

ABSTRACT

Bluetooth is one of the World's best-known brands and one of the most ubiquitous wireless communication technologies. Bluetooth has not stood still. Since its first incarnation Bluetooth has been carefully and systematically improved and it has continued to keep pace with market requirements. Bluetooth mesh networking is the latest chapter in this incredible technology story. In our project we made use of mesh networking capability of BLE to create a wide range monitoring system which sends data to local server and can be used to signal the presence of a hazard, requiring urgent situation. We succeeded to implement the BLE mesh network which collects data from Light and Temperature sensors and updated to local server. The future work on this project involves adding Friend node to support the Low Power feature to operate efficiently, by storing messages and sending the data from local server to the cloud monitoring using gateway such as AT&T cellular shield. IP addressing and enhanced hardware for increasing the range will be an add on to the project.

CHAPTER 1

INTRODUCTION

Bluetooth mesh allows to establish a many-to-many (m:m) relationship between wireless devices. Devices may relay data to other devices not in direct radio range of the originating device. In this way, mesh networks can span very large physical areas and contain large numbers of devices. Instead of creating a connection using the GAP Central-Peripheral interaction and exchanging data over GATT or L2CAP, a mesh node sends data by placing it in an advertising packet. Another mesh node receives the data by scanning for advertisement.

FRDM KW41Z development boards that include M0+ core microcontroller and BLE radio, are configured to act as Leaf nodes and Relay nodes to build the mesh network. Two sensor devices such as Ambient Light sensor and DHT22 Temperature sensor, are connected to Leaf node through a FRDM K64F development board, which detects the physical stimulus from its environment. Relay node retransmit the advertising packet to propagate the message throughout the mesh to reach the destination node. In this way the sensor data from the Leaf node is transmitted from monitoring environment to the destination server.

To send a multicast message, the destination field is set to a multicast address. Receiving and processing these packets is defined under a publish-subscribe paradigm, which states that a node must be subscribed to a multicast address, or a publish address, to process the application message contained by the packet. The nodes with sensors (Leaf Nodes) are set to publish a multicast address and Relay is subscribed to the same multicast address to receive data from the sensor nodes. This data is further transmitted to the commissioner node through a broadcast addressing, and then to a local sever through UART protocol.

CHAPTER 2

REQUIREMENTS FOR THE PROJECT

2.1. Hardware

1. FRDM K64F Development Board.

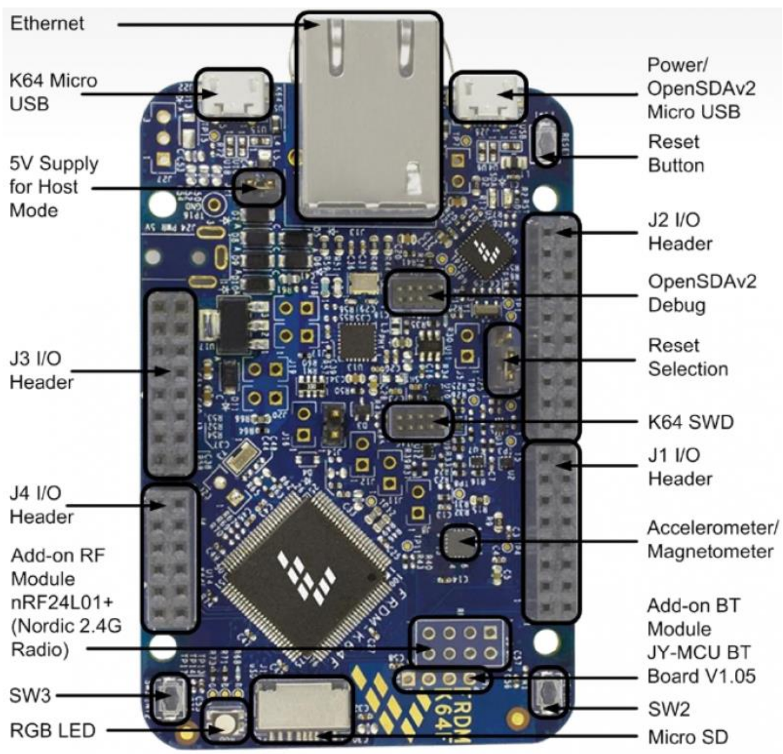


Figure 1.

The Freedom-K64F is an ultra-low-cost development platform by NXP. The features of the FRDM-K64F hardware are as follows:

- MK64FN1M0VLL12 MCU (120 MHz, 1 MB flash memory, 256 KB RAM, low-power, crystal-less USB, and 100 LQFP)
- Dual role USB interface with micro-B USB connector.
- RGB LED.
- FXOS8700CQ – accelerometer and magnetometer.
- Two user push buttons.
- Flexible power supply option – OpenSDAv2 USB, K64 USB, and external source.
- Easy access to MCU input/output through **Arduino R3TM** compatible I/O connectors.
- Programmable **OpenSDAv2** debug circuit supporting the CMSIS-DAP Interface software that provides:

- Mass storage device (**MSD**) flash programming interface.
- CMSIS-DAP debug interface over a driver-less USB HID connection providing run control debugging and compatibility with IDE tools.
- Virtual serial port interface
- Open-source CMSIS-DAP software project: github.com/mbedmicro/CMSIS-DAP

- **SPI, I2C, UART, CAN** protocol and Ethernet.
- Add-on RF module: nRF24L01+ Nordic 2.4GHz Radio.
- Add-on Bluetooth module: JY-MCU BT board V1.05 BT.
- The primary components and their placement on the hardware assembly are explained in Figure 1.

2. FRDM KW41Z Development Board.

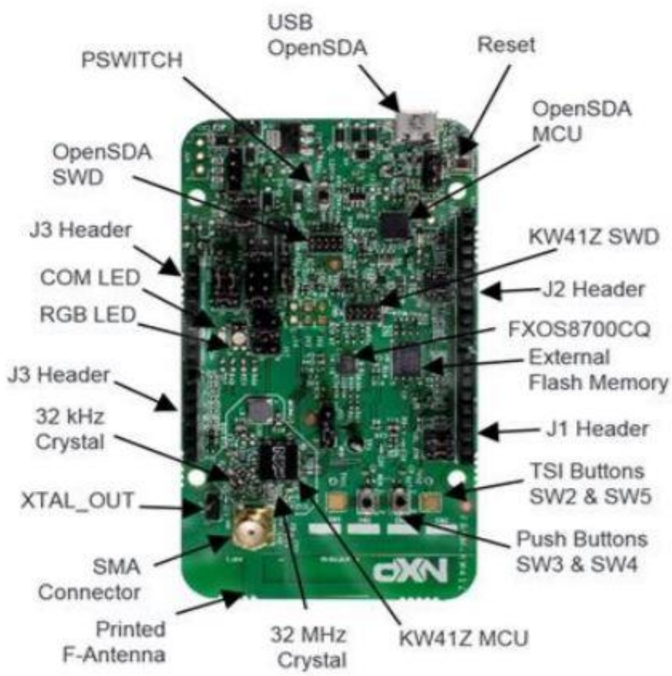


Figure 3. FRDM-KW41Z component placement

The FRDM KW41Z development board has the following features:
• NXP's ultra-low-power KW41Z Wireless MCU supporting BLE, Generic FSK, and IEEE Std. 802.15.4 (Thread) platforms.
• IEEE Std. 802.15.4-2006 compliant transceiver supporting 250 kbps O-QPSK data in 5.0 MHz channels, and full spread-spectrum encoding and decoding.
• Fully compliant Bluetooth v4.2 Low Energy (BLE).

Core and Memories

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

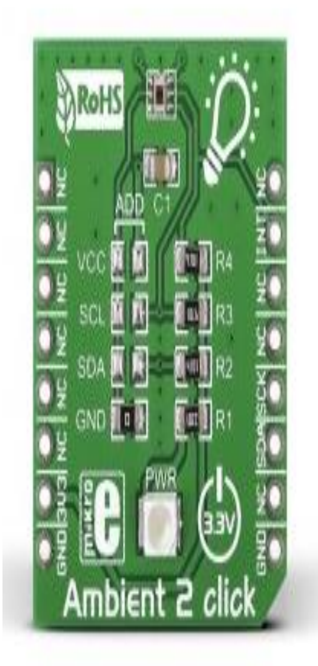
Supported Software

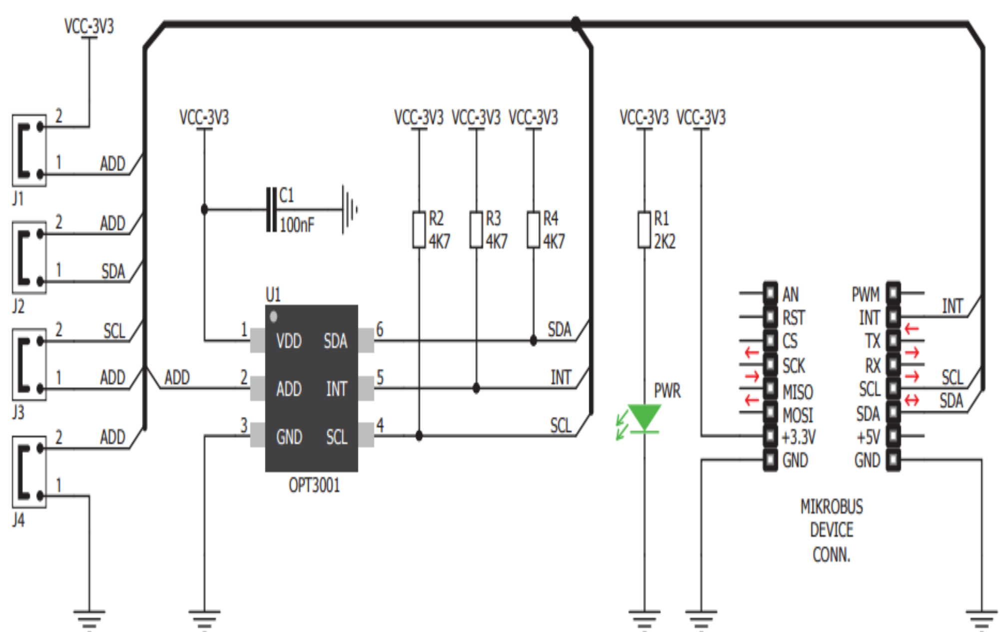
- BLE Host Stack and Profiles.

- Generic FSK Link Layer Software.
- Thread, 802.15.40 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.

3. Ambient light Click Module.

Ambient 2 click carries TI's OPT3001 Ambient Light sensor. It's a small (2mm x 2mm) single-chip lux meter that measures only the visible part of the light spectrum from any kind of source (mimicking the way humans see light). It does so by filtering out 99% of infrared light. Communication with the target MCU is done through mikroBUS I2C pins, with an additional INT pin which can be used for triggering wake-up events (offloading the MCU).

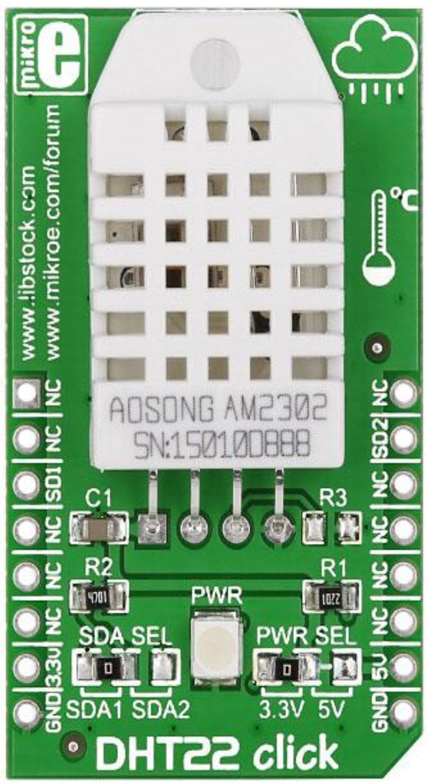


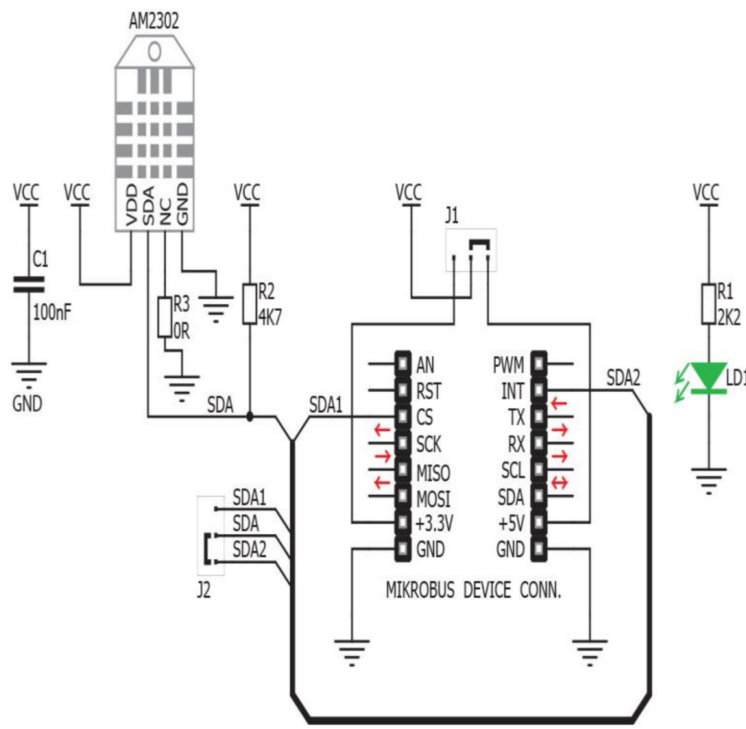


- **Interface:** mikroBUS™ standard I2C interface.
- **Measurement range:** 0.01 lux – 83k lux.
- **Resolution:** 23 bit.
- **On board modules:** OPT3001 Ambient Light Sensor.
- **Power Supply:** 3.3V or 5V.

4. DHT22 Click Module.

DHT22 is temperature and humidity measurement board. It utilizes exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements is connected with 8-bit single-chip computer. Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in type of programme in OTP memory, when the sensor is detecting, it will cite coefficient from memory. Small size & low consumption & long transmission distance(20m) enable DHT22 to be suited in all kinds of harsh application occasions.



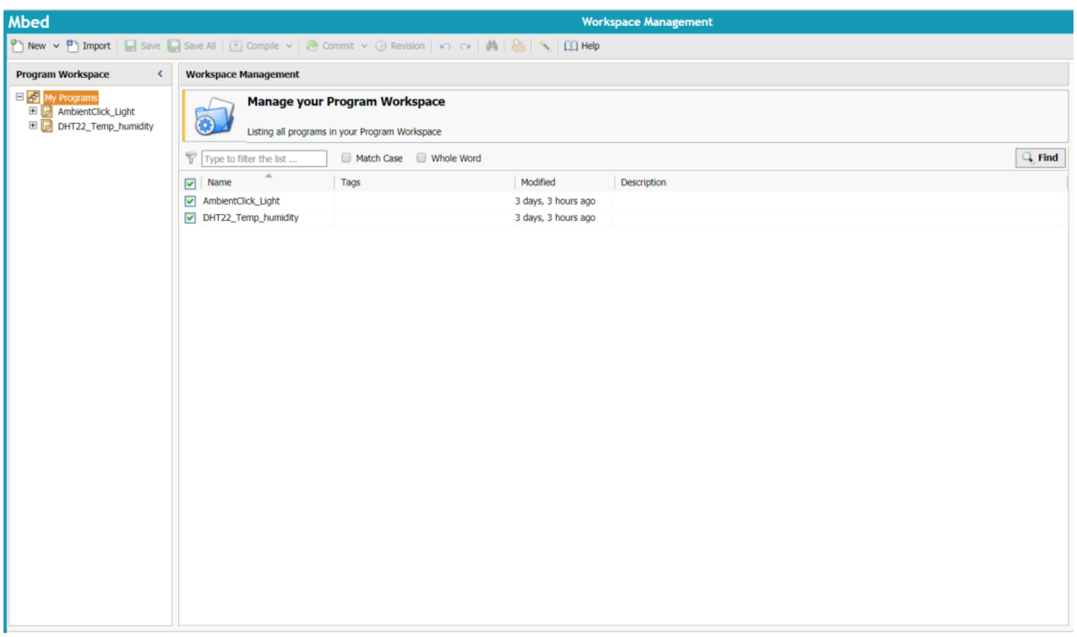


- **Interface**: mikroBUS™ standard one-wire interface.
- **Measurement range**:
  - **Temperature**: -40 degree Celsius to 80 degree Celsius with a half a degree precision.
  - **Humidity**: 0-100 % with 2% accuracy.
- **Resolution**: 16- bit
- **On board Modules**: AM2302 temp/humidity sensor.
- **Power Supply**: 3.3V or 5V.

2.2 Software

1.Mbed

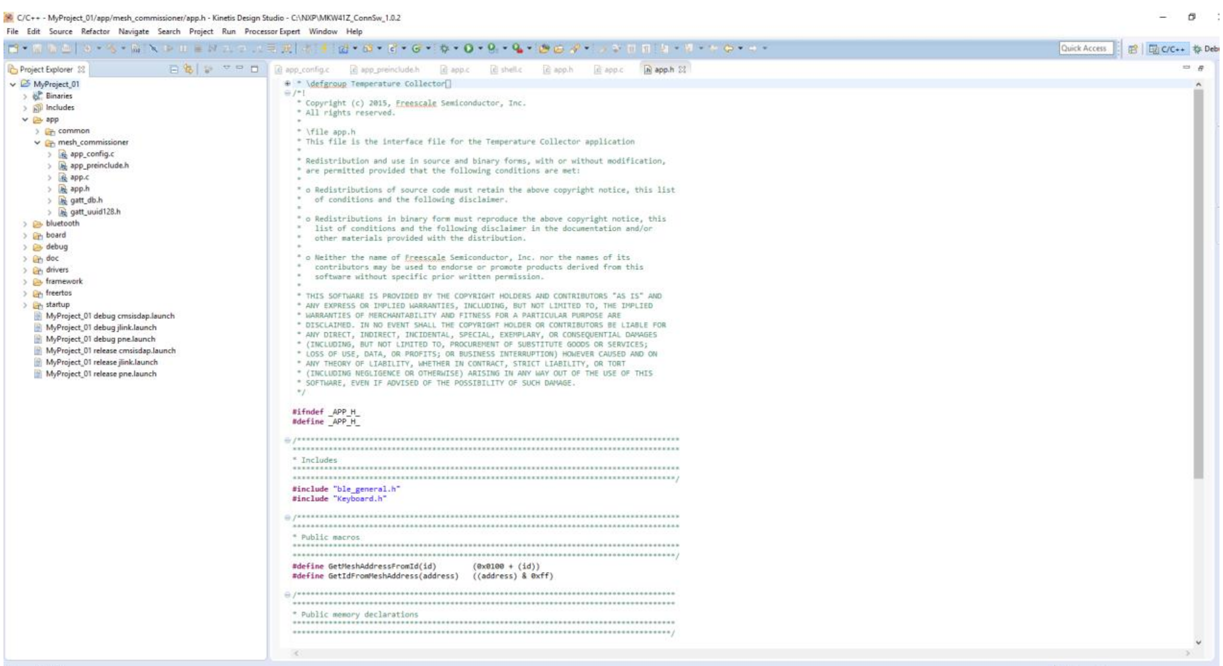
The mbed OS is used to write applications that run on embedded devices, by providing the layer that interprets the application's code in a way the hardware can understand. Application code is written in C++. It uses the *application programming interfaces* (APIs) that mbed OS provides. These APIs allow your code to work on different microcontrollers in a uniform way. This reduces a lot of the challenges in getting started with microcontrollers and integrating large amounts of software.



2. KDS_IDE: Kinetis® Design Studio Integrated Development Environment (IDE)

The Kinetis Design Studio software development tool is a GNU/Eclipse-based development environment for Freescale Kinetis devices. It supports Cortex-M based Kinetis devices and integrates with Processor Expert and Kinetis Software Development Kit. KDS supports SEGGER J-Link/J-Trace, P&E USB Multilink Universal/USB Multilink Universal FX and CMSIS-DAP debug adapters and uses the newlib-nano C runtime library. This runtime library helps reduce the memory footprint of an embedded application GDB debugger with support for the following debug interface hardware:

- SEGGER J-Link (w/SEGGER GDB Server, Windows, Linux, Mac).
- P&E Multilink (w/P&E GDB Server, Windows, Linux).
- CMSIS-DAP (w/OpenOCD GDB and OpenSDA embedded circuit,Windows, Linux).
- Command Line (CL) debugging with GDB and OpenOCD TCL.

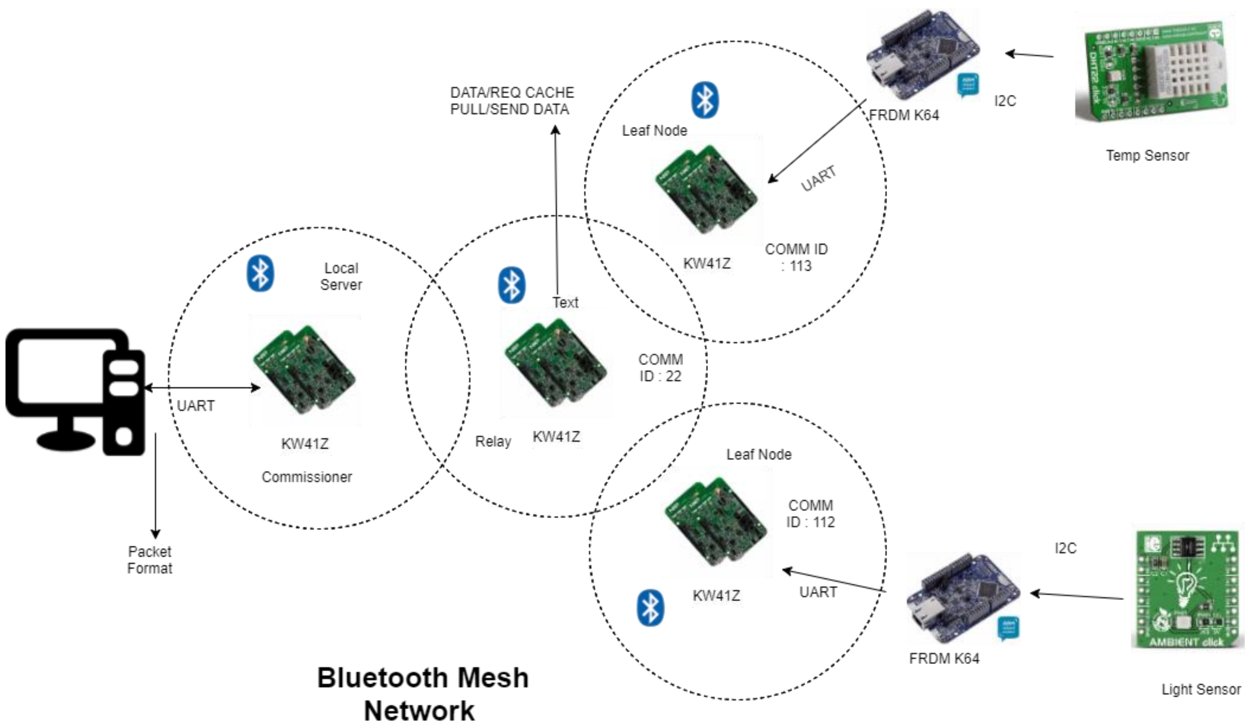


CHAPTER 3

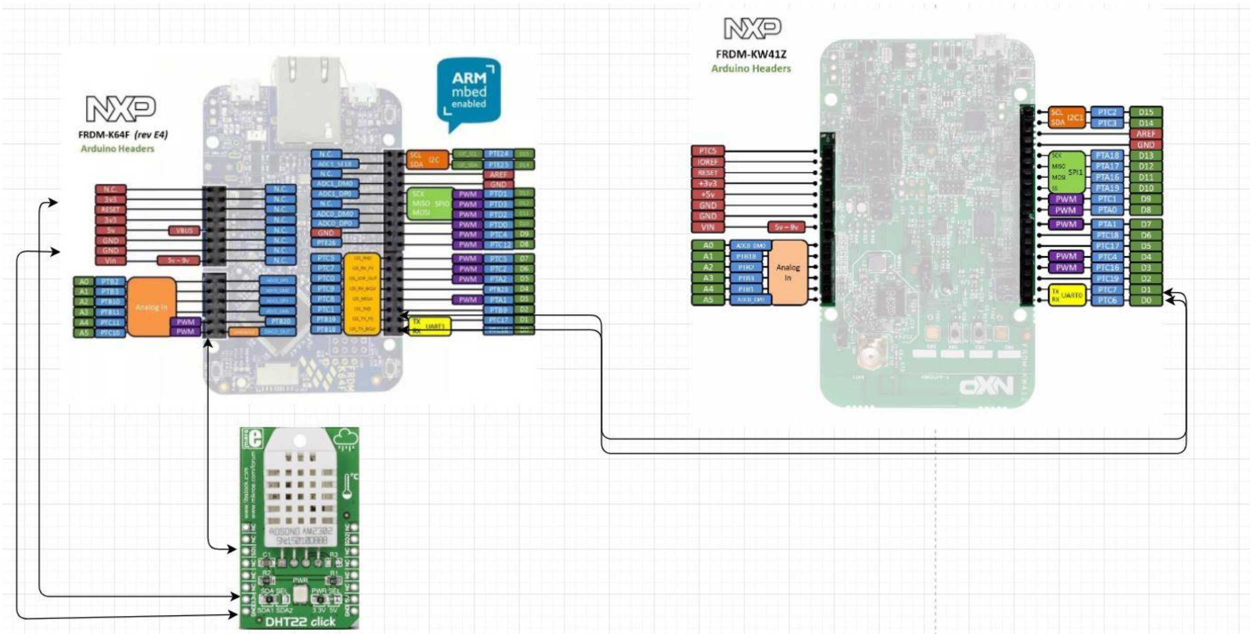
IMPLEMENTATION

The Implementation of the project is done with the following steps.

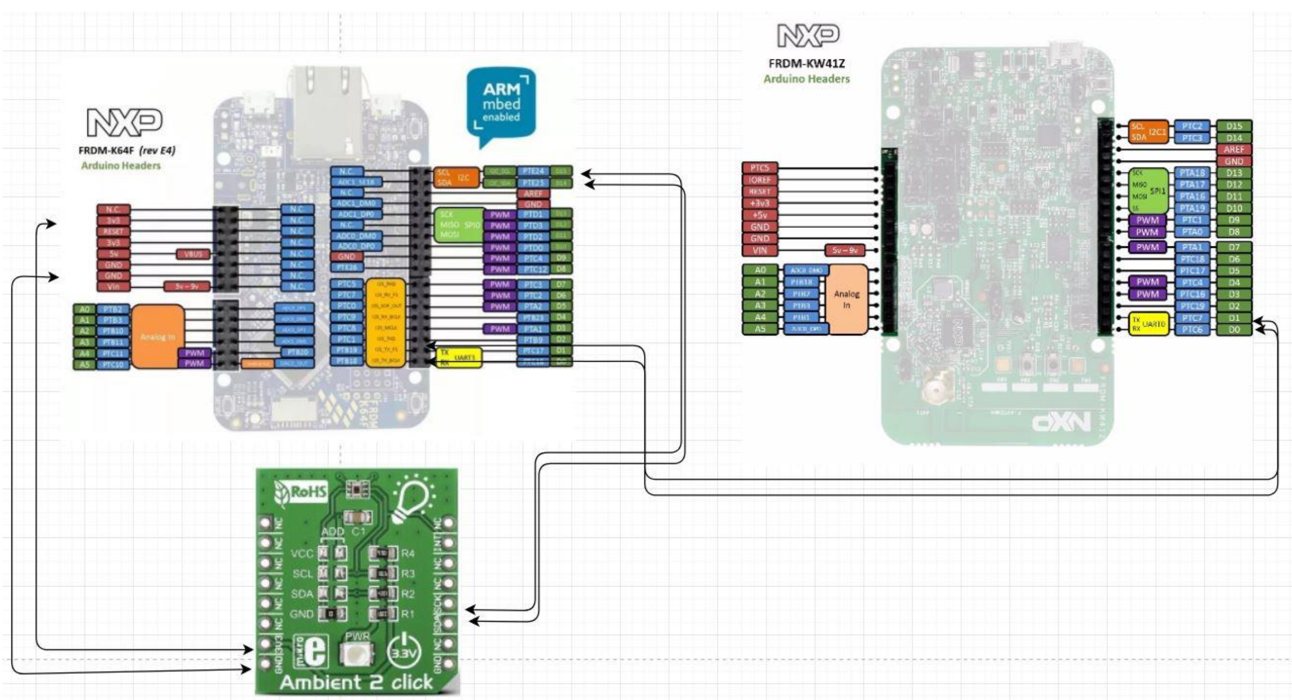
1. **System Overview**

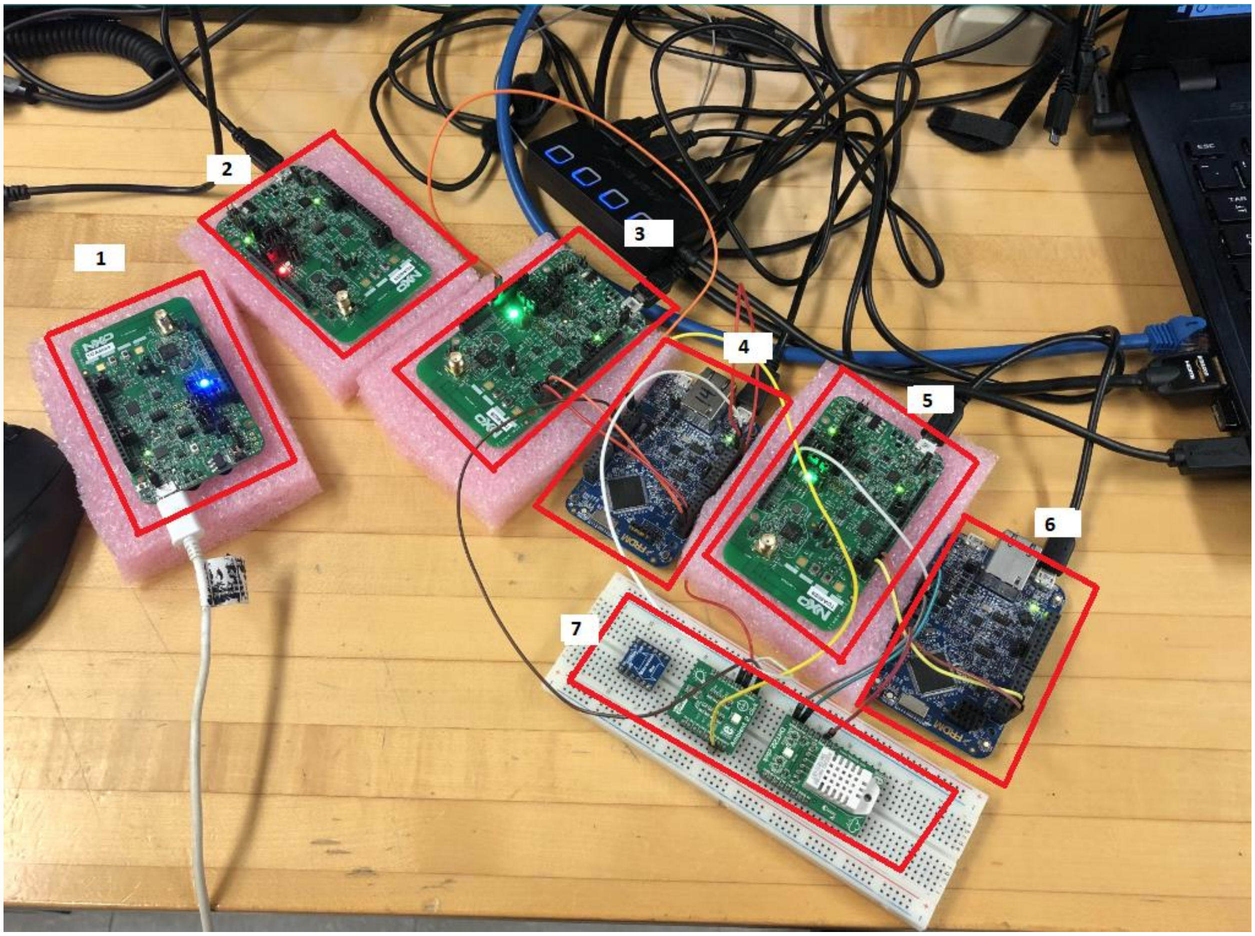


Connection Diagram of DHT22 Temperature sensor



Connection Diagram of Ambient 2 Click Light Sensor





Actual setup

a) The DHT22 Temperature and Humidity sensor module is connected to FRDM K64F (6) using I2C Protocol. Which is then connected to the KW41Z (Leaf Node (5)) using UART.
b) Ambient Light click module is connected to another FRDM K64(4) using I2C, which then transmit the sensor data to the second Leaf Node (3).
c) The Relay Node (1) uses Multicast addressing and publish-subscribe paradigm along with the leaf nodes to transmit the data from the sensor location to the commissioner (2) location.

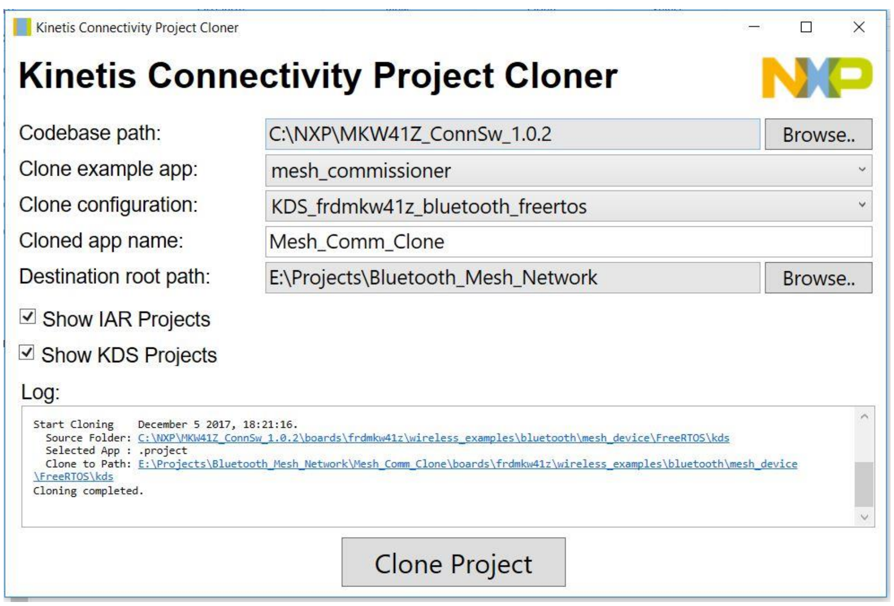
d) Commissioner application is configured as client for the 2 profiles (Temperature and Illuminance) and commissioner uses terminal commands to send Configuration Client commands.

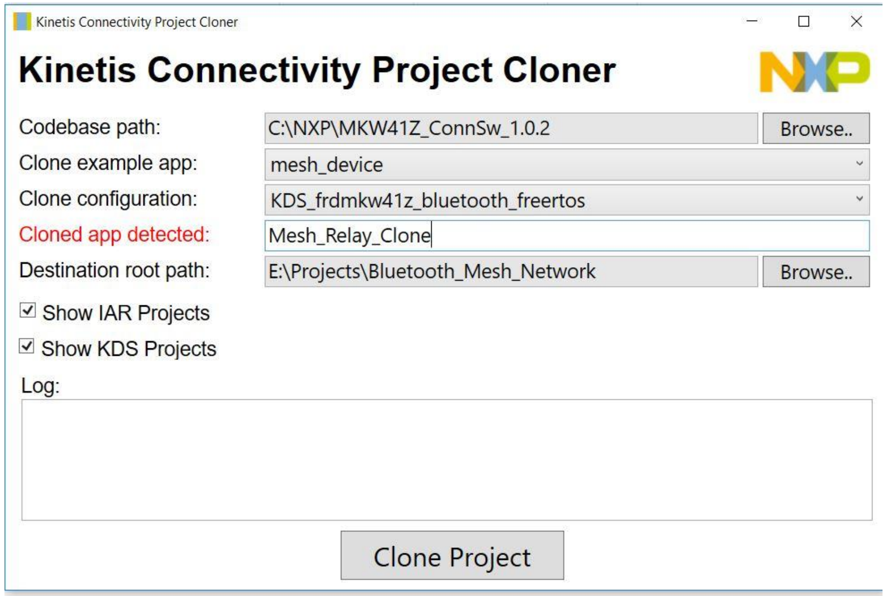
e) The sensor data from the leaf nodes are stored in the cache memory of the Relay (1). Relay act as a forwarding node in the BLE mesh network.

f) The data from the relay is transferred to the commissioner (2) on request from the commissioner. The UART connection from the commissioner node to the local server display data on Tera Term Terminal on local server.

Steps to run BLE Mesh code:

1. Install KDS, Jlink, Project cloner and teraterm.
2. Clone the following three projects:
    a. meshcommissioner – Clone to Folder Mesh_Comm_Clone.
    b. meshdevice – Clone to Folder Mesh_Relay_Clone
    c. meshdevice – Clone to Folder Mesh_Leaf_Clone
3. Select Clone configuration: KDS_frdmkw41z_bluetooth_freertos.





4. In project Mesh_Leaf_Temp_Clone copy the three files from \Bluetooth_Mesh_Network_Project_Source\Mesh_Leaf_Temp_Files folder to corresponding folders mentioned below.
    a. File 1: app.c , app.h – Copy to the folder: \Mesh_Leaf_Temp_Clone\middleware\wireless\bluetooth_1.2.2\examples\mesh_device\
    b. File 2: app_preinclude.h – Copy to the folder: \Mesh_Leaf_Temp_Clone\boards\frdmkw41z\wireless_examples\bluetooth\mesh_device\FreeRTOS\

5. In project Mesh_Leaf_Light_Clone copy the three files from \Bluetooth_Mesh_Network_Project_Source\Mesh_Leaf_Light_Files folder to corresponding folders mentioned below.
    a. File 1: app.c , app.h – Copy to the folder: \Mesh_Leaf_Light_Clone\middleware\wireless\bluetooth_1.2.2\examples\mesh_device\
    b. File 2: app_preinclude.h – Copy to the folder: \Mesh_Leaf_Light_Clone\boards\frdmkw41z\wireless_examples\bluetooth\mesh_device\FreeRTOS\

6. In project Mesh_Relay_Clone copy the three files from \Bluetooth_Mesh_Network_Project_Source\Mesh_Relay_Files folder to corresponding folders mentioned below.
    a. File 1: app.c , app.h – Copy to the folder:

\Mesh_Relay_Clone\middleware\wireless\bluetooth_1.2.2\examples\mesh_device\

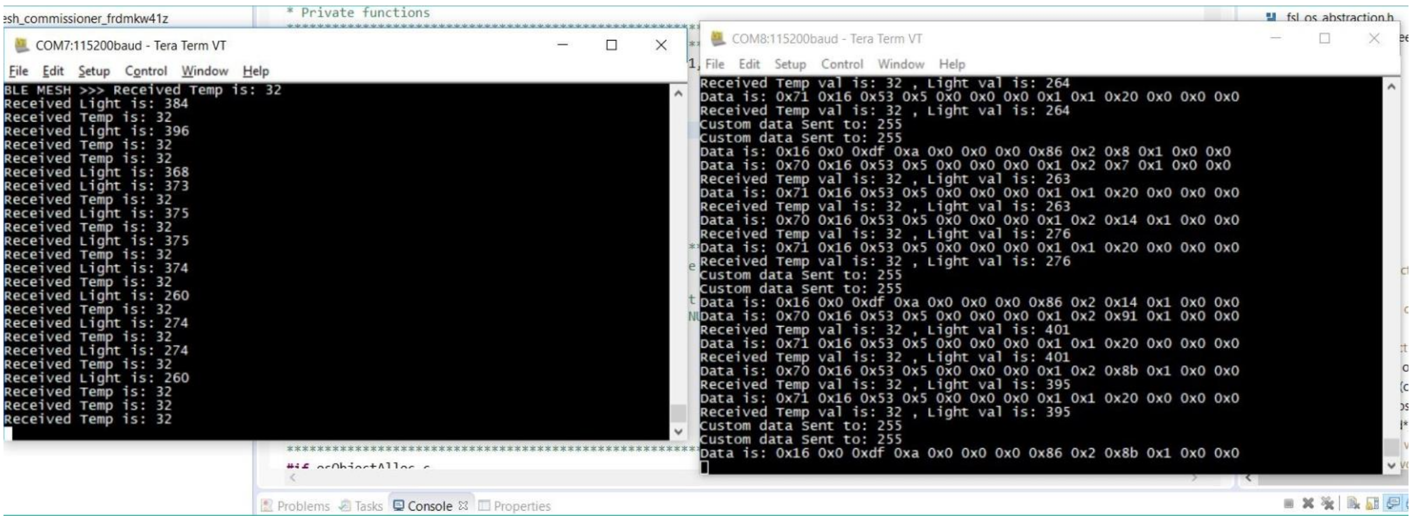
   b. File 2: app_preinclude.h – Copy to the folder: \Mesh_Relay_Clone\boards\frdmkw41z\wireless_examples\bluetooth\mesh_device\FreeRTOS\

7. In project Mesh_Comm_Clone copy the two files from \Bluetooth_Mesh_Network_Project_Source\Mesh_Comm_Files folders to corresponding folders mentioned below.
   a. File 1: shell.c – Copy to the folder: \Mesh_Comm_Clone\middleware\wireless\framework_5.3.2\Shell\Source\
   b. File 2: app.c – Copy to the folder: \Mesh_Comm_Clone\middleware\wireless\bluetooth_1.2.2\examples\mesh_commissioner\

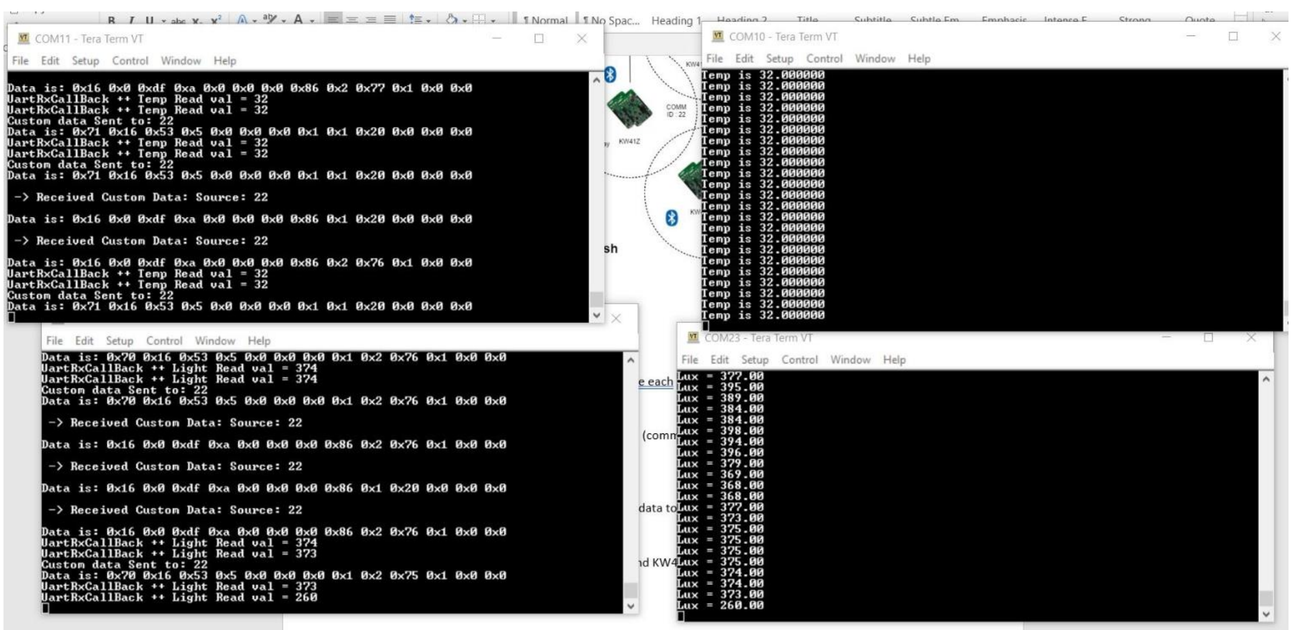
8. Build the projects.
9. Select Debug configurations and disable Enable SW0 in Startup tab and load firmware to corresponding boards.
10. Verify output via Tera term with configuration:
    a. Corresponding port number for J-Link from device manager.
    b. Baud Rate : 115200

CHAPTER 4

RESULT



Measure of Temperature and Illuminance on Tera Term Terminal -1) Commissioner Node 2)Relay Node



Measure of Temperature and Illuminance on Tera Term Terminal 1) Leaf Nodes 2)Corresponding FRDMK64 of each Leaf Node

CONCLUSION

The BLE mesh network of MCU's was successfully created that collects the data from DHT and Ambient light sensors. The values from the DHT and ambient light sensors are sampled regularly and the same is updated to the Local Server. We could also set up communication between FRDM K64F and KW41Z successfully.

CHAPTER 5

REFERENCES

1. https://www.bluetooth.com/
2. https://www.nxp.com
3. https://en.wikipedia.org/wiki/Bluetooth_Low_Energy
4. https://developer.mbed.org/compiler/#nav:/;
5. https://developer.mbed.org/cookbook/Homepage