

Bluetooth Low Energy Mesh Network based Smart Control Automation

Submitted in partial fulfilment of the requirements for the award of

Bachelor of Engineering degree in

Electronics and Communication Engineering

By

MANIBHARATHI TN (37130235)



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC | 12B Status by UGC | Approved by AICTE

JEPPIAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600 119.

March 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC | 12B Status by UGC | Approved by AICTE

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **MANIBHARATHI TN (371302325)**, who carried out the project entitled "**Bluetooth Low Energy Mesh Network based Smart Control Automation**" under our supervision from AUG 2020 to MARCH 2021

Internal Guide

Dr. R.NARMADHA,M.E.,Ph.D.

Head of the Department

Dr. T. RAVI, M.E., Ph.D.

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

T N MANIBHARATHI (37130235), hereby declare that the Project Report entitled “**Bluetooth Low Energy Mesh Network based Smart Control Automation**” done by us under the guidance of **Dr. R.NARMADHA at SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** is submitted In partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

DATE : (1)

PLACE : (2)

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to the Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

I convey my thanks to **Dr. N. M. NANDHITHA, M.E., Ph.D.**, Prof. & Dean, School of Electrical and Electronics Engineering and **Dr. T. RAVI, M.E., Ph.D.**, Head of the Department, Dept. of Electronics and Communication Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. R.NARMADHA, M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **Electronics and Communication Engineering** who were helpful in many ways for the completion of the project.

Abstract

In general Bluetooth Low Energy (BLE) has a leading edge with exceptional low energy technology. It has gained significant momentum in various application domains, such as grouping the street light , indoor positioning and home automation. BLE is a short- range communication technology , it needs a mesh network to cover a large area ,for the establishment of Bluetooth nodes with indirect contacts. Especially for the implication of real time communication , mesh networks support BLE specification. In our proposed method the real-time protocol , developed on top of BLE that allows for bounded packet delays over mesh networks. In addition to that priority support can also be achieved through BLE wireless communication technology In this proposed method,real time communication is implemented between BLE nodes and there is no direct contact(INTER LINK) between them. Effective device-to-device communication with the inception of FORWARD MESSAGING. It can support all operating systems eg; android ,IOS, etc., Similarly NRF52 , relays used to control various devices.The final result will be controlling the group of devices in a mesh Network using NRF52 Kit

CONTENTS

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	v
	LIST OF FIGURES	viii
1.	INTRODUCTION	1
1.3	DEVICE ROLL	2
1.4	GATT	3
1.5	BLUETOOTH MESH	5
2.	LITERATURE REVIEW	
2.1.	LITERATURE SURVEY	6
2.2	LITERATURE SUMMARY	8
3.	PROPOSED WORK	
3.1	BLOCK DIAGRAM OF PROPOSED WORK	9
3.2	DEVICES AND NODES	9
3.3	STATES AND PROPERTIES	12
3.3.1	Messages, States and Properties	12
3.3.2	STATE TRANSITIONS	13
3.3.1	BOUND STATES	13

3.4 COMPONENTS USED	13
3.4.1 nRF52 MESH BOARD	13
3.4.3 SEGGER EMBEDDED STUDIO	16
4. RESULTS AND DISCUSSION	17
4.1 UNPROVISIONED NODE	17
4.2 CIRCUIT DIAGRAM	17
4.3 PROVISIONING	18
4.8 FLASHING TO THE BOARD	36
4.9 RESULT	37
5. CONCLUSION AND FUTURE WORK	39
6. REFERENCES	40

LIST OF FIGURES

FIGURE No.	TITLE	PAGE No.
3.1	CIRCUIT diagram of the proposed work	9
3.4.1	nRF52 MESH BOARD	13
3.4.2	RELAY	15
4.1	ADDING THE UNPROVISIONED NODE	17
4.2	PROVISIONING	17
4.3	CREATING A GROUP	19
4.4	SUBSCRIBING THE MESH GROUP	19
4.5	CONTOURING THE THREE REALLY	20
4.6	FLASHING TO THE BOARD	36
4.8	NRF MESH BOARD-OFF	37
4.9	NRF MESH BOARD-On	37

CHAPTER 1

1.1 Bluetooth

Bluetooth has been effectively evolved since its delivery in 2000, when it was initially expected to fill in as a link substitution innovation. It before long started to rule remote sound and PC created items like remote mice and consoles. In 2010, Bluetooth LE offered the following stage, a major advance forward. Its effect was boisterous and discernible, particularly on Smartphones and tablets, just as in classifications. Remote correspondence frameworks dependent on network interchanges have demonstrated a compelling method to give huge scope establishment, broadening widths and giving toughness. Nonetheless, so far they depend on specialty innovation, not viable with most PCs, cell phones and extras that are claimed by customers or utilized in business. 120 organizations that are individuals from the Bluetooth SIG have taken an interest in the work needed to offer help for Bluetooth-based locales. This is a lot higher than expected, and addresses the interest for a worldwide norm, the force of the Bluetooth organization. The expansion of rocket network support demonstrates a specific sort of progress, with so much size as it permits it to be characterized as a worldview change in Bluetooth innovation.

1.2 Bluetooth Low Energy (BLE)

Bluetooth Low Energy (BLE) is a little remote correspondence innovation that can be utilized throughout a brief timeframe to empower brilliant gadgets to convey. Different gadgets you work with consistently, for example, your cell phone, savvy, wellness tracker, remote earphones and PC use BLE to make a consistent encounter between your gadgets. BLE is another standard Bluetooth association characterized by the Bluetooth Special Interest Group (SIG) zeroed in on low force. It has empowered gadget producers to add a low-power interface to existing arrangements. It has additionally been utilized to make extraordinary failure power gadgets, for example, reinforcements that can be controlled by a little cell battery for quite a long time or even years. BLE has a wide extent of possibilities and is used in various fields like prosperity,

wellbeing, security, home automation, home redirection, adroit industry and IoT (Internet of Things). It is also close to Smartphones and workstations that we use every day. Apple actually secured BLE on Smartphones with the appearance of the iPhone 4s out of 2011. Today, most Android and iOS devices accessible fuse BLE correspondence and correspondence with various devices. Most PDAs have acknowledged the fast apportionment of BLE and outfit device makers with induction to an overall BLE customer base.

1.3 Device roles

To understand how BLE works, we need to define some of the terms and concepts in play. The BLE device plays a central or peripheral role, sometimes called a client or server.

Central (Client)

A tool for initiating commands and requests and receiving responses.

Examples: computer, smartphone

Peripheral (server)

A tool for receiving and responding to orders and requests

Examples: Temperature sensor and heart rate monitor

The fringe character we regularly consider comparable to BLE gadgets like LE earphones, wellness trackers and pulse screens. Fringe is a gadget that advances association accessibility and gives an interface to correspondence. All things being equal, the gadget that distinguishes, associates and conveys peripherals assumes a key part - for instance PCs or cell phones. A middle will frequently filter for close by peripherals and show a rundown of identified gadgets. In the event that peripherals can be associated, the center point can be associated with a particular fringe and correspondence.

1.4 GATT

GATT represents Generic Attribute Profile, and characterizes the cycle by which two Bluetooth low power gadgets move information to and from utilizing administration and highlight ideas. It utilizes the Generic Data Protocol Attribute Protocol (ATT), which is utilized to store administrations, ascribes, and related information in a basic quest table with 16-cycle IDs for every section in the table. GATT is empowered when a different association is set up between two gadgets, which implies you have effectively gone through the GAP-controlled publicizing measure. The significant thing to recollect about GATT and associations is that associations are remarkable. This implies that BLE peripherals must be associated with one focal gadget (cell phone, and so forth) at a time! When associated with a fringe focal gadget, it will kill the promotions and different gadgets won't see or interface with it until the current association is broken. Setting up an association is the best way to permit two-way correspondence where the focal gadget can send fringe and significant information from.

1.4.1 Connected Network Topology

The accompanying graph ought to clarify the way that Bluetooth Low Energy gadgets work in an associated climate. A fringe must be associated with one focal gadget (like a cell phone) at a time, but the focal gadget can be associated with different peripherals. On the off chance that information should be traded between two peripherals, a custom letter box framework should be carried out where all messages go through the focal gadget. When an association is set up between a fringe and focal gadget, be that as it may, correspondence can occur in the two ways, which is not the same as the single direction broadcasting approach utilizing just promoting information and GAP.

1.4.2 GATT Transactions

A significant idea to comprehend with GATT is the worker/customer relationship. The fringe is known as the GATT Server, which holds the ATT query information and administration and trademark definitions, and the GATT Client (the telephone/tablet), which sends solicitations to this worker. All exchanges are begun by the fundamental

gadget, the GATT Client, which gets reaction from the auxiliary gadget, the GATT Server. When setting up an association, the fringe will propose a 'Association Interval' to the focal gadget, and the focal gadget will attempt to reconnect each association span to check whether any new information is accessible, and so forth. It's imperative to remember that this association stretch is truly an idea, however! Your focal gadget will be unable to respect the solicitation since it's bustling conversing with another fringe or the necessary framework assets simply aren't accessible.

1.4.3 Services and Characteristics

GATT exchanges in BLE depend on significant level, settled items called Profiles, Services and Characteristics, which can be found in the representation beneath:

Profiles

A Profile doesn't really exist on the BLE fringe itself, it's basically a pre-characterized assortment of Services that has been arranged by either the Bluetooth SIG or by the fringe architects. The Heart Rate Profile, for instance, consolidates the Heart Rate Service and the Device Information Service.

Services

Administrations are utilized to split information up into rationale substances, and contain explicit lumps of information called qualities. A help can have at least one attributes, and each assistance separates itself from different administrations through a one of a kind numeric ID called a UUID, which can be either 16-digit (for formally embraced BLE Services) or 128-bit (for custom administrations). A full rundown of authoritatively received BLE administrations can be seen on the Services page of the Bluetooth Developer Portal. In the event that you take a gander at the Heart Rate Service, for instance, we can see that this formally received help has a 16-bit UUID of 0x180D, and contains up to 3 trademark, however just the first is obligatory: Heart Rate Measurement, Body Sensor Location and Heart Rate Control Point.

Characteristics

The most minimal level idea in GATT exchanges is the Characteristic, which epitomizes a solitary information point (however it might contain a variety of related information, for example, X/Y/Z esteems from a 3-hub accelerometer, etc.). Similarly to Services, every Characteristic separates itself by means of a pre-characterized 16-bit or 128-digit UUID, and you're allowed to utilize the standard qualities characterized by the Bluetooth SIG (which guarantees interoperability across and BLE-empowered HW/SW) or characterize your own custom attributes which just your fringe and SW gets it. For instance, the Heart Rate Measurement trademark is required for the Heart Rate Service, and uses a UUID of 0x2A37. It begins with a solitary 8-cycle esteem depicting the HRM information design (regardless of whether the information is UINT8 or UINT16, and so on), and afterward proceeds to incorporate the pulse estimation information that coordinates with this config byte. Attributes are the primary concern that you will cooperate with your BLE fringe, so it's imperative to comprehend the idea. They are likewise used to send information back to the BLE fringe.

1.5 Bluetooth Mesh

Most Bluetooth LE gadgets speak with one another utilizing a basic highlight point network geography empowering balanced gadget interchanges. In the Bluetooth center determination, this is known as a 'piconet.' Imagine a cell phone that has set up a highlight direct association toward a pulse screen over which it can move information. One pleasant part of Bluetooth is that it empowers gadgets to set up numerous associations. That equivalent cell phone can likewise set up a highlight point association with a movement tracker. For this situation, the cell phone can discuss straightforwardly with every one of different gadgets, however different gadgets can't discuss straightforwardly with one another. Interestingly, a lattice network has a many-to-numerous geography, with every gadget ready to speak with each and every gadget in the cross section. Communication is accomplished utilizing messages, and gadgets can transfer messages to different gadgets so the start to finish correspondence every individual hub.

CHAPTER 2

2. LITERATURE REVIEW

2.1 LITERATURE SURVEY

Overview of Bluetooth Low Energy Mesh Networks: Survey of Communication and Security Protocols

Bluetooth Low Energy (BLE) Mesh Networks empower adaptable and dependable correspondences for low-power Internet of Things (IoT) gadgets. Most BLE-based lattice conventions are executed as overlays on top of the standard Bluetooth star geographies while utilizing piconets and scatternets. In any case, network geography support has expanded the weakness of BLE to security dangers, since a bigger number of gadgets can partake in a BLE Mesh organization. To address these worries, BLE rendition 5 improved existing BLE security highlights to manage different realness, honesty, and classification issues. Nonetheless, there is as yet an absence of point by point considers identified with these new security highlights. This overview looks at the latest BLE-based cross section network conventions and related security issues. In the initial segment, the most recent BLE-based cross section correspondence conventions are examined. The examination shows that the execution of BLE unadulterated lattice conventions stays an open exploration issue. In addition, there is an absence of auto-setup components to help bootstrapping of BLE unadulterated cross section organizations. In the subsequent part, late BLE-related security issues and weaknesses are featured. Solid Intrusion Detection Systems (IDS) are fundamental for distinguishing security penetrates to ensure against zero-day misuses. In any case, practical IDS answers for BLE Mesh networks stay an incipient exploration territory. Subsequently, a near overview of IDS approaches for related low-power remote conventions was utilized to delineate expected methodologies for upgrading IDS answers for BLE Mesh organizations.

Overview of Bluetooth Low Energy Mesh Networks: A Survey

Bluetooth Low Energy (BLE) has acquired critical force. Be that as it may, the first plan of BLE zeroed in on star geography organizing, which cutoff points network inclusion range and blocks start to finish way variety. Interestingly, other contending advancements defeat such limitations by supporting the lattice network geography. Therefore, the scholarly community, industry, and norms improvement associations have been planning answers for empower BLE network organizations. By the by, the writing comes up short on a combined view on this arising region. This paper completely studies cutting edge BLE network organizing. We initially give a scientific classification of BLE network arrangements. We at that point audit the arrangements, depicting the assortment of approaches that influence existing BLE usefulness to empower BLE network organizations. We recognize essential parts of BLE network arrangements and examine their benefits and downsides. At long last, we feature at present open issues.

Overview of Bluetooth Low Energy Industrial Wireless Mesh Networks

Mechanical remote sensor organizations are utilized to get sensor information that need ongoing preparing, along these lines they require unsurprising conduct and constant assurances. To be savvy, IWSNs are likewise expected to be ease and low force. In this unique situation, Bluetooth low energy (BLE) is a promising innovation, as it permits carrying out ease mechanical organizations. As BLE is a short reach innovation, a multihop network is expected to cover an enormous region. In any case, the as of late distributed Bluetooth network organizing particulars don't offer help for ongoing interchanges over multihop network organizations. To beat this constraint, this paper proposes the multihop ongoing BLE (MRT-BLE) convention, a continuous convention created on top of BLE, that takes into consideration limited bundle delays over network organizations. MRT-BLE likewise gives need support. This paper depicts in detail the MRT-BLE convention and how to execute it on business off-the-rack gadgets. Two sorts of execution assessment for the MRT-BLE convention are given. The first is a most pessimistic scenario start to finish defer examination, while the

subsequent one depends on the test results acquired through estimations on a genuine testbed.

Overview of Beyond beaconing: Emerging applications and challenges of BLE

As an arising innovation with uncommon low energy utilization and low-inactivity information transmissions, Bluetooth Low Energy (BLE) has acquired critical force in different application areas, like Indoor Positioning, Home Automation, and Wireless Personal Area Network (WPAN) correspondences. With different novel convention stack highlights, BLE is discovering use on asset obliged sensor hubs just as more impressive door gadgets. basically because of the reference points' energy proficiency and simplicity of arrangement.

2.2 LITERATURE SUMMARY

Numerous Algorithms and works have been proposed and created to accomplish a superior System. By the way, each proposed framework does not have a couple of highlights. Despite the fact that the downsides of one framework have been remunerated in the other, No framework in general gathering network in it , in our task we have built up the gathering control utilizing the nRF52 network board and individual execution too.

CHAPTER 3

3.1 Devices and Nodes

Devices that are important for a lattice network are called hubs and non-provisioned devices. The way toward changing over a non-provisioning gadget into a hub is classified "provisioning". Consider purchasing another Bluetooth light with network backing and bringing it home and setting it up. To make it a piece of your lattice organization, you can handle your current Bluetooth light switches and dimmers, you should give this. Provisioning is a safe strategy that includes numerous encryption keys for a non-provisioning gadget and is known to the provisioning gadget, normally a tablet or cell phone. One of these keys is known as an organization key or Network for short. You can peruse more about network security in the Security area. Every hub in the cross section network contains in any event one net, which contains this key, which makes the gadget an individual from the subsidiary organization, just as the actual hub. Different prerequisites should be met before the hub can be utilized, yet getting the organization through the provisioning cycle is an essential initial step.

3.2 BLOCK DIAGRAM OF PROPOSED WORK

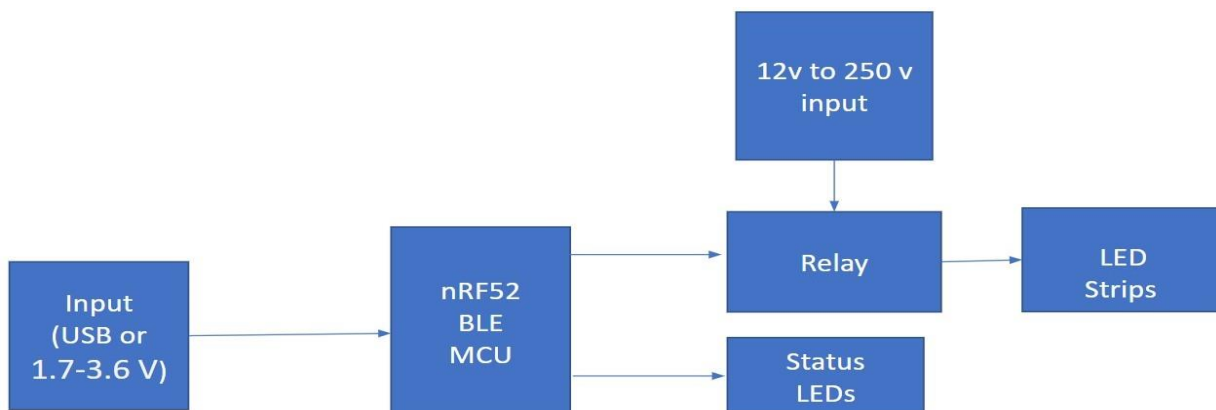


Fig.3.2 BLOCK DIAGRAM

3.2.1 Elements

A few hubs have more than one segment, every one of which is controlled freely. In Bluetooth network wording, these segments are called segments. At the point when a LED lighting item is added to a Bluetooth network organization, a solitary hub with three segments is shaped, one for each LED light.

3.2.2 Messages

In the event that a hub needs to check the situation with different hubs or control different hubs here and there, it sends the fitting message. On the off chance that a hub needs to report its status to different hubs, it communicates something specific. All correspondences in the lattice network are characterized as "message based" and a few message types, each with its own code. Messages can be categorized as one of two general classes; Accepted or obscure. Affirmed messages require input from the hubs that get them. Reaction fills two needs: it affirms that the message was gotten and returns the sender with the information related with the message beneficiary. On the off chance that the creator of the approved message doesn't get a reaction (s), the message might be sent again and in this manner the affirmed messages should be invalid. This implies that the impact of the approved message is a similar when it arrives at the hub more than once, when it is gotten just a single time. No reaction is needed for obscure messages.

3.2.3 Addresses

Messages should be sent from the location. Bluetooth network characterizes three kinds of addresses. A Unicast address exceptionally distinguishes a solitary piece. Unicast delivers are allotted to devices during the provisioning interaction. Gathering address is a multicast address that addresses at least one things. Gathering addresses are characterized by the Bluetooth Special Interest Group (SIG), otherwise called SIG static gathering addresses or doled out powerfully. 4 SIG characterized bunch addresses are characterized. These are called all-intermediaries, all-companions, all-transfers and all-hubs. The terms intermediary, front and hand-off

will be clarified later in this paper. Through the design application the client is relied upon to put dynamic gathering tends to that mirror the actual arrangement of the structure, for example, characterizing bunch tends to that fit each room in the structure. A virtual location is a location that can be appointed to at least one segments reaching out to at least one hubs. It appears as a 128-bit UUID esteem that can be related with any component, like a mark.

3.2.4 Publish/Subscribe

The demonstration of communicating something specific is called distributing. Hubs are designed to choose messages shipped off explicit locations for handling and membership. Commonly, messages are shipped off bunch or virtual locations. The end client comprehends the significance of gathering and virtual locations, they are not difficult to utilize and clear. We saw that the "Switch 1" hub bunch distributes the location for the kitchen. Hubs Lite1, Lite2 and Lite3 each buy in to the kitchen address, so get and measure distributed messages to this location. All in all, Light 1, Light 2 and Light 3 can be turned on or off with Switch 1. Switch 2 distributes the gathering address to the lounge area. Just Light 3 is bought in to this location, so it is the lone light turned on by Switch 2. This model likewise delineates the way that hubs can buy in to messages routed to numerous various locations. It is solid and basic. Additionally, notice how two hubs Switch 5 and Switch 6 distribute to a similar nursery address. The utilization of gathering/virtual locations in the general population/buy in correspondence model doesn't need the recreation of different hubs to eliminate, supplant or add new hubs to the organization. Look at what is engaged with introducing extra lighting in the lounge area. The new gadget will be added to the organization utilizing the provisioning cycle and the lounge area address will be arranged for membership. This change to the organization doesn't influence different hubs. Switch 2 will keep on distributing messages to the lounge area as in the past, however now Light 3 and New Light are reacting.

3.3 States and Properties

The parts might be in various states, which is shown by the idea of state esteems in the Bluetooth machine. State is a particular sort of significant worth in a component (in the worker model - see beneath). Notwithstanding the qualities, the states have added substance properties that are not reused in different cases. For instance, consider conventional light that can be turned on or off. Bluetooth characterizes the lattice of on nonexclusive status. The light has this status factor, the estimation of On enlightens the light, simultaneously mirrors the ordinary on-off state estimation of the offer and the light is turned off. The meaning of the term nonexclusive will be examined later. The properties are like expresses that have values related to a factor. In any case, in alternate ways they are totally different from the states. Perusers acquainted with Bluetooth LE will know about the highlights that are related in light of them and keep that they are information types that don't have the characterized practices related with them and can be reused in an assortment of settings. A character gives setting to depict a character. To see the value in the significance and handiness of the setting comparable to the highlights, for instance, consider the 8-cycle temperature state type temperature 8, which has many related properties, including the current indoor surrounding temperature and the current Dodor encompassing temperature. Both of these highlights permit the sensor to distribute sensor readings, which permits the accepting customer to decide the setting of the temperature esteem, which better comprehends its actual significance. Properties are coordinated into two areas: Manufacturer, which is a perused just classification and Administrator, which permits read-compose access.

3.3.1 Messages, States and Properties

Messages A framework that solicitations activities on the machine. Message Officially, the sort of message gave addresses an assortment of state or multi-state esteems. All messages reflect three expansive sorts, useful sorts upheld by Bluetooth Mesh. There are three sorts of truncations: GET, SET and STATUS. GET messages demand a given status esteem from at least one hubs. Send a STATUS message in

light of GET and it will have comparing state esteem. Set messages change the estimation of a specific status. The status message is returned in light of the approved set message, yet no reaction is needed to the obscure set message. Sends STATUS messages autonomously from GET messages, approved SET messages or different messages, for instance a clock running on the message sender. The particular status of the message movement is gotten from the message opcode. Highlights with a 16-bit property ID are plainly shown in open property related messages.

3.3.2 State Transitions

Changes starting with one state then onto the next are called state advances. Advances might be prompt or execute throughout some stretch of time called the change time. A state progress is probably going to affect the application layer conduct of a hub.

3.3.3 Bound States

Connections may exist between states whereby an adjustment in one triggers an adjustment in the other. Such a relationship is known as a state restricting. One state might be bound to different other states. For model, consider a light constrained by a dimmer switch. The light would have the two states, Generic OnOff and Generic Level with each bound to the next. Lessening the brilliance of the light until Generic Level has an estimation of nothing (completely diminished) brings about Generic OnOff progressing from On to Off.

3.4 Components used:

3.4.1 NRF52

The nRF52 DK is a solitary board improvement pack (DK) for Bluetooth Low Energy, Bluetooth lattice, ANT and 2.4 GHz exclusive applications utilizing the nRF52810 and nRF52832 SoCs. It works with advancement abusing all highlights of the nRF52810 and the nRF52832 SoCs. It incorporates a NFC receiving wire that rapidly empowers use of the NFC-A label fringe on the nRF52832. The unit offers admittance to all I/Os and

interfaces through edge connectors and has 4 LEDs and 4 catches which are client programmable. The unit is viable with the Arduino Uno Revision 3 norm, making it conceivable to utilize outsider safeguards that are viable to this norm

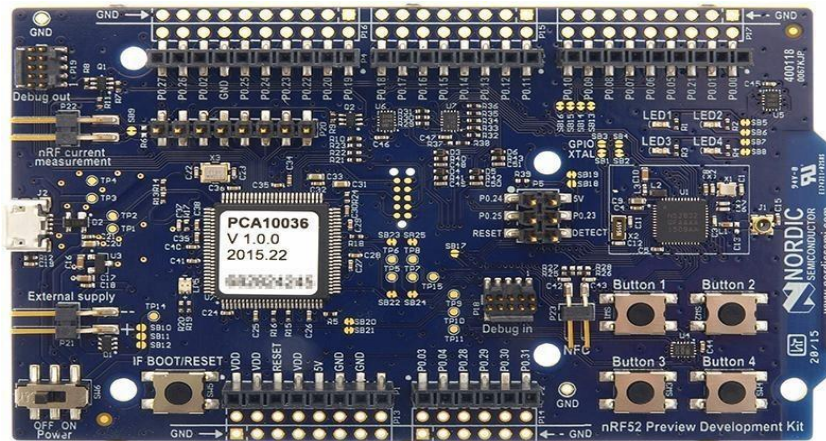


Fig.3.4.1 NRF52 mesh board

. It accompanies an on-board SEGGER J-Link debugger permitting programming and troubleshooting both the on-board SoC and outside SoCs through the investigate out header. It very well may be controlled by USB or by an outer source, yet additionally incorporates a CR2032 battery holder, empowering in-field testing of models. A scope of programming models are accessible from the nRF5 programming improvement pack (SDK) to help Bluetooth LE, ANT and 2.4 GHz applications. For Bluetooth network improvement, look at nRF5 SDK for Mesh. SEGGER Embedded Studio, Keil, GCC and IAR IDEs and toolchains are upheld.

Key Features:

- I. . Supports Bluetooth LE,
- II. . Bluetooth mesh,
- III. . ANT and 2.4 GHz proprietary development Bluetooth 5 feature support: 2 Mbps, Advertising Extensions and Channel Selection Algorithm (CSA #2)
- IV. . Arm® Cortex™ M4 with floating point support
- V. . DSP instruction set
- VI. . Arduino Uno Rev. 3 compatible for use with 3rd party shields
- VII. . All I/Os and interfaces are available via connectors

- VIII. · Segger J-Link OB programming/debugging supported
- IX. · Support for programming/debugging of external boards
- X. · Integrated 2.4 GHz PCB antenna
- XI. · NFC antenna
- XII. · Connector for external NFC antenna
- XIII. · Connector for RF measurements
- XIV. · Pins for power consumption measurements
- XV. · Coin-cell battery holder
- XVI. · 1.7-3.6 V supplied from battery, USB or external

3.4.2 SSR Relay



Fig.3.4.2 SSR Relay

Relay is an electronically worked switch. Its three terminals are NC, NO, COM. At the point when the relay loop isn't actuated, the relay will go about as an open circuit, when we enact the relay curl, it will go about as a shut circuit. Relays can be likewise utilized as a securing gadget, e.g Buchholz relay. It is an oil and gas incited relay that is utilized to shield the transformer from the short out. When there is a deficiency in the transformer, because of shortcoming current, the gas in the transformer gets hot and the air pockets will happen and which is given by the chamber lines to the relay and the caution circuit will be set off dependent on the power of the imperfection.

Relay pinout and Construction

NO (Normally open)

NC (Normally closed)

Common terminal

A Relay has an electromagnet, armature (portable contact or COM terminal) and two fixed contacts (NC, NO). In the underlying stage, the COM terminal will be near the NO terminal. At the point when we initiate electromagnet, electromagnet produces attractive motion which pulls in armature towards it and the situation of the armature changes from the COM to the NC.

Normally Close

This is a regularly shut association when the electromagnet is empowered, the transition delivered by the electromagnet will draw in the armature towards the NC terminal. As demonstrated in the subsequent figure

Normally Open

This is a Normally Open association when the electromagnet isn't actuated, at that point the armature will be near this terminal.

Application

1. To control the high power devices relays are used.
2. In Ac motor's stator, it is used to start connection to delta connection.

Segger Embedded studio

Installed Studio is the All-In-One answer for overseeing, building, testing and conveying inserted applications. This implies smooth, effective improvement tasks because of its wide scope of highlights. The amazing venture administrator empowers the administration of activities enormous and little. Adaptation control highlights empower programmed application arrangement. With a Visual Studio-like appearance, SEGGER's Embedded Studio improves both productivity and ease of use. Inserted designs presently appreciate the convenience that numerous product engineers know because of its natural yet adjustable format

CHAPTER 4

4.1 Adding the Unprovisioned node

Open the nRF mesh app in mobile then turn on the bluetooth , then go to the network to add the unprovisioned nRF mesh mode



Fig.4.1 Adding the Unprovisioned node

4.2 Provisioning

Provisioning is the interaction by which a gadget joins the cross section organization and turns into a hub. It includes a few phases, brings about different security keys being produced and is itself a safe cycle. Provisioning is cultivated utilizing an application on a gadget like a tablet. In this limit, the gadget used to drive the provisioning interaction is alluded to as the Provisioner.

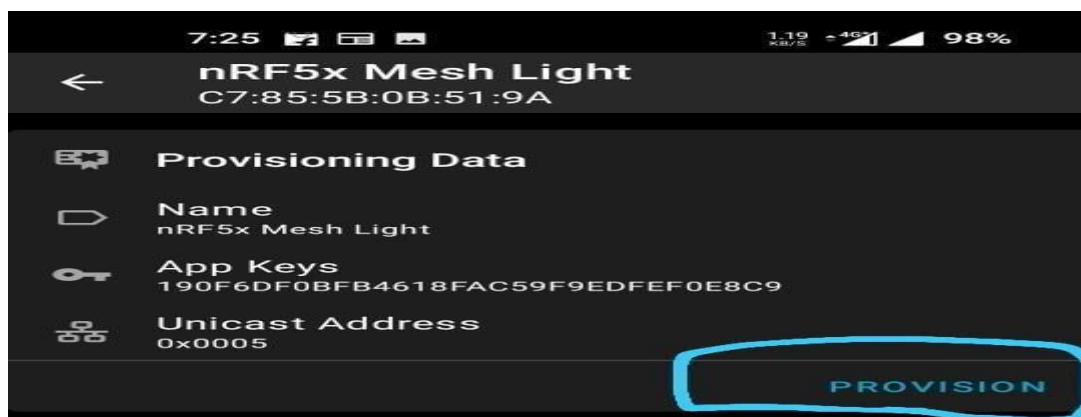


Fig.4.2 Provisioning

The provisioning process progresses through five steps and these are described next.

Step 1. Beaconing

On the side of different diverse Bluetooth network highlights, including however not restricted to provisioning, new GAP AD types (ref: Bluetooth Core Specification Supplement) have been presented, including the <<Mesh Beacon>> AD type. An unprovisioned gadget demonstrates its accessibility to be provisioned by utilizing the <<Mesh Beacon>> AD type in promoting bundles. The client may have to begin another gadget promoting in this manner by, for instance, squeezing a mix of catches or holding down a catch for a specific period of time.

Step 2. Invitation

In this progression, the Provisioner sends a solicitation to the gadget to be provisioned, as a Provisioning Invite PDU. The Beaconing gadget reacts with data about itself in a Provisioning Capabilities PDU.

Step 3. Exchanging Public Keys

The Provisioner and the gadget to be provisioned, trade their public keys, which might be static or fleeting, either straightforwardly or utilizing an out-of-band (OOB) technique

Step 4. Authentication

During the confirmation step, the gadget to be provisioned yields an irregular, single or multi-digit number to the client in some structure, utilizing an activity proper to its capacities. For instance, it may streak a LED a few times. The client enters the digit(s) yield by the new gadget into the Provisioner and a cryptographic trade happens between the two gadgets, including the irregular number, to finish the verification of every one of the two gadgets to the next.

Step 5. Distribution of the Provisioning Data

After confirmation has effectively finished, a meeting key is determined by every one of the two gadgets from their private keys and the traded, peer public keys. The meeting key is then used to get the resulting dissemination of the information needed to finish the provisioning interaction, including a security key known as the organization key (NetKey). In the wake of provisioning has finished, the provisioned gadget has the organization's NetKey, a cross section security boundary known as the IV Index and a Unicast Address, distributed by the Provisioner. It is presently known as a hub.

4.3 Creating a Group for controlling the relays from the NRF mesh app

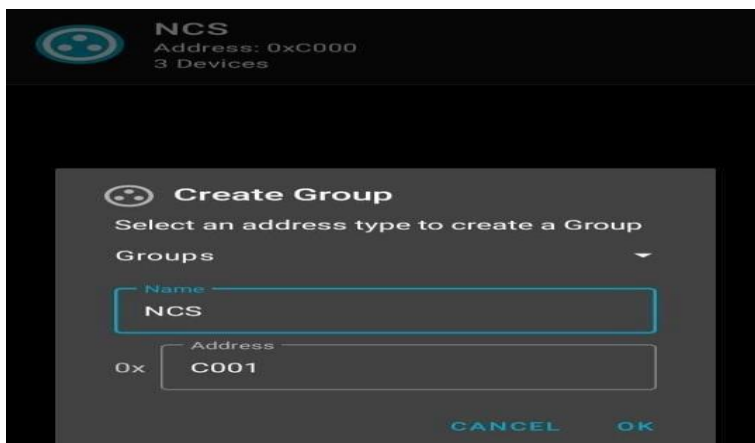


Fig.4.3 Group for controlling

4.4. Subscribing to that mesh group that was created to control the relays

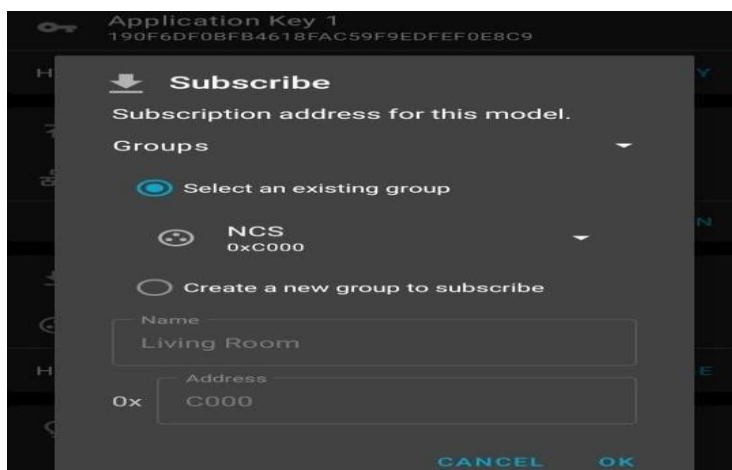


Fig.4.4. Subscribing to that mesh group

4.5. Controlling the 3 relays from the mesh app

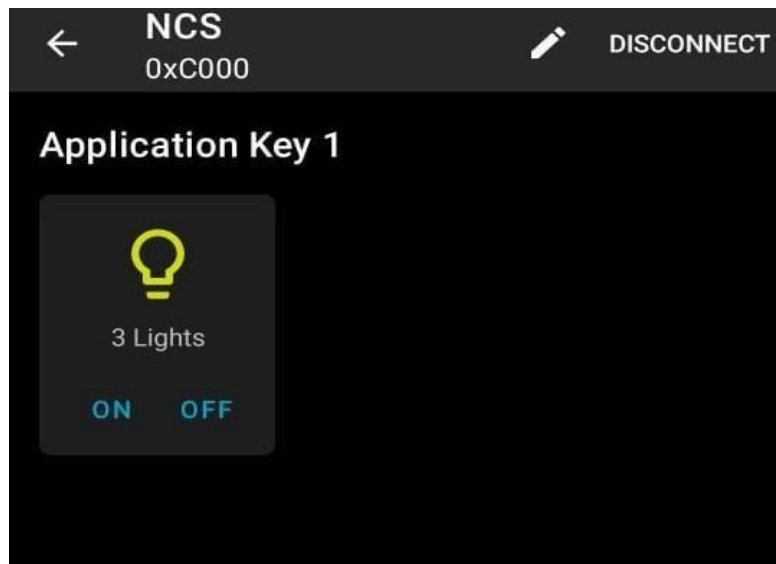


Fig.4.5 mesh app

Now we can control all the mesh node that are connected to the app can be controlled and we can verify the each node status, then we can control each node in individual or group manner

4.6 source code

`/* HAL based on the our requirement add all the extension in the SEGGER Embedded Studio for ARM 5.40c */`

`#include <stdint.h>`

`#include <string.h>`

`#include "boards.h"`

`#include "simple_hal.h"`

`#include "app_timer.h"`

`/* Core */`

```

#include "nrf_mesh_config_core.h"

#include "nrf_mesh_gatt.h"

#include "nrf_mesh_configure.h"

#include "nrf_mesh.h"

#include "mesh_stack.h"

#include "device_state_manager.h"

#include "access_config.h"

#include "proxy.h"

/* Provisioning and configuration with reference to the topic 4.1*/

#include "mesh_provisionee.h"

#include "mesh_app_utils.h"

/* Models */

#include "generic_onoff_server.h"

/* Logging and RTT */

#include "log.h"

#include "rtt_input.h"

/* Example specific includes */

#include "app_config.h"

#include "example_common.h"

#include "nrf_mesh_config_examples.h"

```

```

#include "light_switch_example_common.h"

#include "app_onoff.h"

#include "ble_softdevice_support.h"

/*****

* Definitions Controls if the model instance should force all mesh messages to
be segmented messages

*****/

#define ONOFF_SERVER_0_LED      (BSP_LED_0)

#define APP_ONOFF_ELEMENT_INDEX    (0)

/* Controls if the model instance should force all mesh messages to be
segmented messages. */

#define APP_FORCE_SEGMENTATION    (false)

/* Controls the MIC size used by the model instance for sending the mesh
messages. */

#define APP_MIC_SIZE            (NRF_MESH_TRANSMIC_SIZE_SMALL)

/*****

* Forward declaration of static functions

*****/

static void app_onoff_server_set_cb(const app_onoff_server_t * p_server, bool onoff);

static void app_onoff_server_get_cb(const app_onoff_server_t * p_server, bool *
p_present_onoff);

```

```

static void app_onoff_server_transition_cb(const app_onoff_server_t * p_server,

                                         uint32_t transition_time_ms, bool target_onoff);

/*****

* Static variables

*****/

static bool m_device_provisioned;

/* Generic OnOff server structure definition and initialization */

APP_ONOFF_SERVER_DEF(m_onoff_server_0,

                    APP_FORCE_SEGMENTATION,

                    APP_MIC_SIZE,

                    app_onoff_server_set_cb,

                    app_onoff_server_get_cb,

                    app_onoff_server_transition_cb)

/* Callback for updating the hardware state */

static void app_onoff_server_set_cb(const app_onoff_server_t * p_server, bool onoff)

{

    /* Resolve the server instance here if required, this example uses only 1
instance. */

    __LOG(LOG_SRC_APP, LOG_LEVEL_INFO, "Setting GPIO value: %d\n", onoff)

    hal_led_pin_set(ONOFF_SERVER_0_LED, onoff);

```

```
}
```

```
/* Callback for reading the hardware state */
```

```
static void app_onoff_server_get_cb(const app_onoff_server_t * p_server, bool *  
p_present_onoff)
```

```
{
```

```
    /* Resolve the server instance here if required, this example uses only 1  
instance. */
```

```
    *p_present_onoff = hal_led_pin_get(ONOFF_SERVER_0_LED);
```

```
}
```

```
/* Callback for updating the hardware state */
```

```
static void app_onoff_server_transition_cb(const app_onoff_server_t * p_server,  
                                           uint32_t transition_time_ms, bool target_onoff)
```

```
{
```

```
    __LOG(LOG_SRC_APP, LOG_LEVEL_INFO, "Transition time: %d, Target OnOff:  
%d\n",
```

```
          transition_time_ms, target_onoff);
```

```
}
```

```
static void app_model_init(void)
```

```
{
```

```
    /* Instantiate onoff server on element index APP_ONOFF_ELEMENT_INDEX */
```

```

ERROR_CHECK(app_onoff_init(&m_onoff_server_0,
APP_ONOFF_ELEMENT_INDEX));

__LOG(LOG_SRC_APP, LOG_LEVEL_INFO, "App OnOff Model Handle: %d\n",
m_onoff_server_0.server.model_handle);

}

/*****

static void node_reset(void)

{

__LOG(LOG_SRC_APP, LOG_LEVEL_INFO, "----- Node reset -----\\n");

hal_led_blink_ms(LED_MASK, LED_BLINK_INTERVAL_MS,
LED_BLINK_CNT_RESET);

/* This function may return if there are ongoing flash operations. */

mesh_stack_device_reset();

}

static void config_server_evt_cb(const config_server_evt_t * p_evt)

{

if (p_evt->type == CONFIG_SERVER_EVT_NODE_RESET)

{

node_reset();

}

}

```

```

#if NRF_MESH_LOG_ENABLE

static const char m_usage_string[] =

    "\n"

    "\t\t-----\n"

    "\t\t Button/RTT 1) LED state will toggle and inform clients about the state change.\n"

    "\t\t Button/RTT 4) Clear all the states to reset the node.\n"

    "\t\t-----\n";

#endif

static void button_event_handler(uint32_t button_number)

{

    /* Increase button number because the buttons on the board is marked with 1 to 4 */

    button_number++;

    __LOG(LOG_SRC_APP, LOG_LEVEL_INFO, "Button %u pressed\n",
button_number);

    switch (button_number)

    {

        /* Pressing SW1 on the Development Kit will result in LED state to toggle and trigger the STATUS message to inform client about the state change. This is a demonstration of

        state change publication due to local event. */

```

```

case 1:

{
    __LOG(LOG_SRC_APP, LOG_LEVEL_INFO, "User action \n");

    hal_led_pin_set(ONOFF_SERVER_0_LED,
!hal_led_pin_get(ONOFF_SERVER_0_LED));

    app_onoff_status_publish(&m_onoff_server_0);

    break;

}

/* Initiate node reset */

case 4:

{
    /* Clear all the states to reset the node. */

    if (mesh_stack_is_device_provisioned())

    {
#if MESH_FEATURE_GATT_PROXY_ENABLED

        (void) proxy_stop();

#endif

        mesh_stack_config_clear();

        node_reset();

    }

```

```

else

{

    __LOG(LOG_SRC_APP, LOG_LEVEL_INFO, "The device is unprovisioned.
Resetting has no effect.\n");

}

break;

}

default:

    __LOG(LOG_SRC_APP, LOG_LEVEL_INFO, m_usage_string);

    break;

}

}

static void app_rtt_input_handler(int key)

{

    if (key >= '1' && key <= '4')

    {

        uint32_t button_number = key - '1';

        button_event_handler(button_number);

    }

    else

```

```

{
    __LOG(LOG_SRC_APP, LOG_LEVEL_INFO, m_usage_string);
}
}

static void device_identification_start_cb(uint8_t attention_duration_s)
{
    hal_led_mask_set(LED_MASK, false);

    hal_led_blink_ms(BSP_LED_2_MASK | BSP_LED_3_MASK,
                    LED_BLINK_ATTENTION_INTERVAL_MS,
                    LED_BLINK_ATTENTION_COUNT(attention_duration_s));
}

static void provisioning_aborted_cb(void)
{
    hal_led_blink_stop();
}

static void unicast_address_print(void)
{
    dsm_local_unicast_address_t node_address;

    dsm_local_unicast_addresses_get(&node_address);
}

```

```

        __LOG(LOG_SRC_APP, LOG_LEVEL_INFO, "Node Address: 0x%04x \n",
node_address.address_start);
    }

static void provisioning_complete_cb(void)
{
    __LOG(LOG_SRC_APP, LOG_LEVEL_INFO, "Successfully provisioned\n");

#ifdef MESH_FEATURE_GATT_ENABLED

    /* Restores the application parameters after switching from the Provisioning
    * service to the Proxy */

    gap_params_init();

    conn_params_init();

#endif

    unicast_address_print();

    hal_led_blink_stop();

    hal_led_mask_set(LED_MASK, LED_MASK_STATE_OFF);

    hal_led_blink_ms(LED_MASK, LED_BLINK_INTERVAL_MS,
LED_BLINK_CNT_PROV);
}

static void models_init_cb(void)
{
    __LOG(LOG_SRC_APP, LOG_LEVEL_INFO, "Initializing and adding models\n");

```

```

    app_model_init();
}

static void mesh_init(void)
{
    mesh_stack_init_params_t init_params =
    {
        .core.irq_priority      = NRF_MESH_IRQ_PRIORITY_LOWEST,
        .core.lfclksrc          = DEV_BOARD_LF_CLK_CFG,
        .core.p_uuid            = NULL,
        .models.models_init_cb  = models_init_cb,
        .models.config_server_cb = config_server_evt_cb
    };

    uint32_t status = mesh_stack_init(&init_params, &m_device_provisioned);

    switch (status)
    {
        case NRF_ERROR_INVALID_DATA:

            __LOG(LOG_SRC_APP, LOG_LEVEL_INFO, "Data in the persistent memory
was corrupted. Device starts as unprovisioned.\n");

            __LOG(LOG_SRC_APP, LOG_LEVEL_INFO, "Reset device before starting of
the provisioning process.\n");

```

```

        break;

    case NRF_SUCCESS:

        break;

    default:

        ERROR_CHECK(status);

    }

}

static void initialize(void)

{

    __LOG_INIT(LOG_SRC_APP | LOG_SRC_FRIEND, LOG_LEVEL_DBG1,
LOG_CALLBACK_DEFAULT);

    __LOG(LOG_SRC_APP, LOG_LEVEL_INFO, "----- BLE Mesh Light Switch Server
Demo -----\\n");

    ERROR_CHECK(app_timer_init());

    hal_leds_init();

#ifdef BUTTON_BOARD

    ERROR_CHECK(hal_buttons_init(button_event_handler));

#endif

    ble_stack_init();

#ifdef MESH_FEATURE_GATT_ENABLED

    gap_params_init();

```

```

    conn_params_init();

#endif

    mesh_init();

}

static void start(void)
{
    rtt_input_enable(app_rtt_input_handler, RTT_INPUT_POLL_PERIOD_MS);

    if (!m_device_provisioned)
    {
        static const uint8_t static_auth_data[NRF_MESH_KEY_SIZE] =
STATIC_AUTH_DATA;

        mesh_provisionee_start_params_t prov_start_params =
        {
            .p_static_data      = static_auth_data,
            .prov_sd_ble_opt_set_cb = NULL,
            .prov_complete_cb = provisioning_complete_cb,
            .prov_device_identification_start_cb = device_identification_start_cb,
            .prov_device_identification_stop_cb = NULL,
            .prov_abort_cb = provisioning_aborted_cb,
        }
    }
}

```

```

        .p_device_uri = EX_URI_LS_SERVER

    };

    ERROR_CHECK(mesh_provisionee_prov_start(&prov_start_params));

}

else

{

    unicast_address_print();

}


mesh_app_uuid_print(nrf_mesh_configure_device_uuid_get());

ERROR_CHECK(mesh_stack_start());

__LOG(LOG_SRC_APP, LOG_LEVEL_INFO, m_usage_string);

hal_led_mask_set(LED_MASK, LED_MASK_STATE_OFF);

        hal_led_blink_ms(LED_MASK, LED_BLINK_INTERVAL_MS,
LED_BLINK_CNT_START);

}

int main(void)

{

    initialize();

    start();

```

```
for (;;)

{

    (void)sd_app_evt_wait();

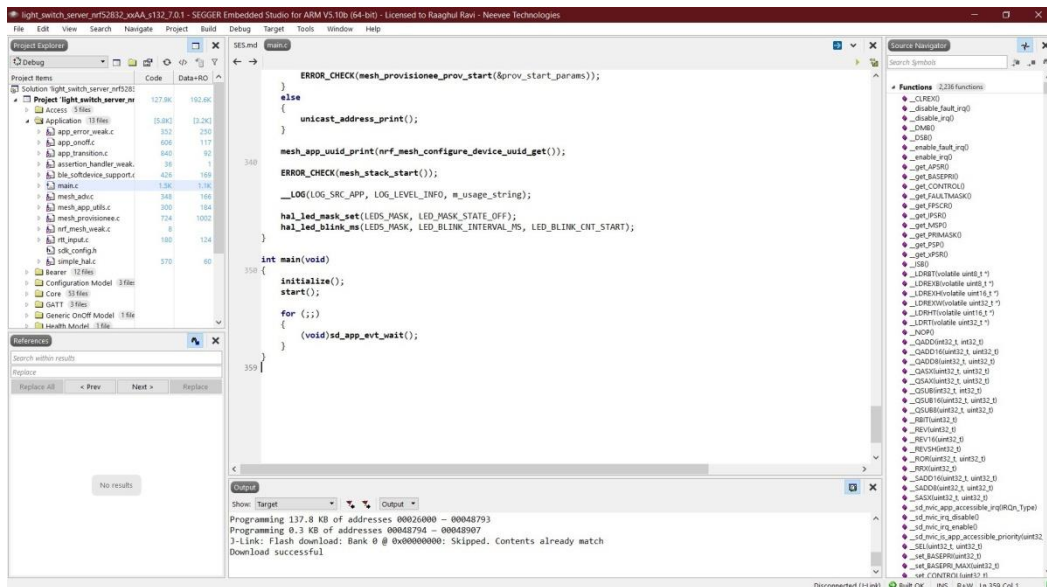
}

}
```

Connect the meshboard and by using the SEGGER Embedded Studio for ARM 5.40c run the code in the NRF mesh board add all the extensions in it and now connect it by using the bluetooth

4.7 Flashing to the board

Open the SEGGER Embedded Studio then add all the library functions then build the program then debug the code then connect the usb cable to the mesh board , do the same for all the mesh network to group them



Compiled code is now flashed to the NRF52 board

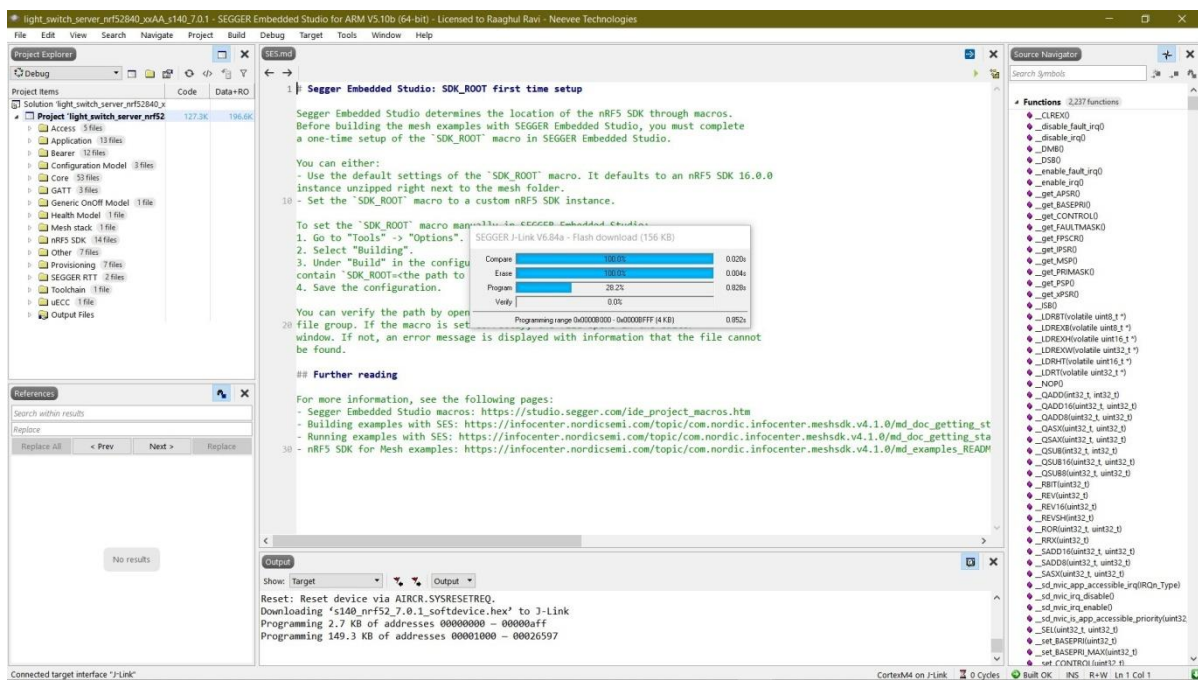


Fig.4.7 Flashing to the board

Result

Source code worker switch model code with our custom alteration is assemble and incorporated in the segger Embedded studio IDE ,ordered code is presently glimmered to the NRF52 board through segger implanted studio,since our NRF board isn't provisioned, presently it is an unprovisioned hub. NRF board is checked and included the NRF network app..Unprovisioned NRF board is provisioned in the NRF network application .A gathering was made for controlling the transfers from the NRF network application .NRF board was bought in to that cross section bunch for controlling the transfers from the lattice application .Relays that was associated with the NRF board can be presently controlled through the NRF network application

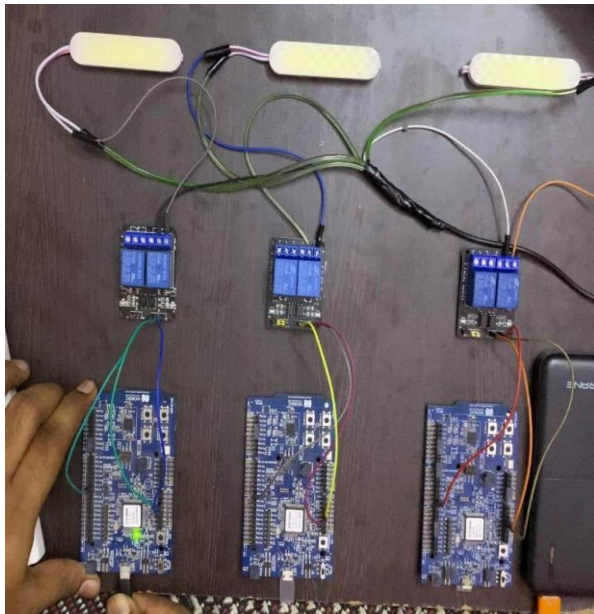


Fig.4.8 Nrf meshbord- Off

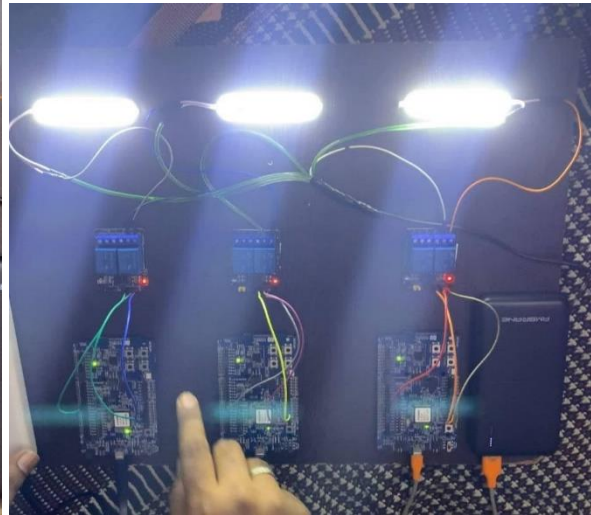


Fig.4.9 Nrf meshbord- ON

CHAPTER 5

5.Conclusion and Future work

It was an awesome learning experience for us while chipping away at this venture. This task took us through the different periods of venture improvement and gave me genuine understanding into the universe of inserted computer programming. The delight of working and the rush in question while handling the different issues and difficulties gave me a vibe of the designers business.

It was because of this undertaking I came to realize how expert inserted programming is planned. Since we can handle transfers from our cell phone application, we can handle any gadget like engine, TV, AC, clothes washer, and so on from our cell phone. This venture utilizes BLE network which was created as a main priority for low force utilization. This undertaking devours lesser force and has more proficiency. The primary part of this undertaking is to change over an ordinary gadget into a brilliant gadget, since we can handle the gadget and read the condition of the gadget that was associated with the hand-off

6.REFERENCE

- [1] Muhammad Rizwan Ghori ¹OrcID,Tat-Chee Wan ^{1,2,*}OrcID andGian Chand Sodhy ¹OrcID *Sensors* 2020, 20(12), 3590; “Bluetooth Low Energy Mesh Networks: Survey of Communication and Security Protocols”Received: 5 May 2020 / Revised: 15 June 2020 / Accepted: 15 June 2020 / Published: 25 June 2020
- [2] JianYang ChristianPoellabauer PramitaMitra CynthiaNeubecker ” Beyond beaconing: Emerging applications and challenges of BLE” Received 23 October 2018, Revised 3 September 2019, Accepted 22 September 2019, Available online 25 September 2019.
- [3] Seyed Mahdi Darroudi andCarles Gomez “ Bluetooth Low Energy Mesh Networks: A Survey”Received: 15 May 2017 / Revised: 17 June 2017 / Accepted: 20 June 2017 / Published: 22 June 2017
- [4] Ghori, M.R.; Wan, T.C.; Sodhy, G.C. Bluetooth Low Energy 5 Mesh Based Hospital Communication Network (B5M HCN). In *Advances in Cyber Security*; Springer: Singapore, 2020; Volume 1132, pp. 247–261. [Google Scholar]
- [5] Darroudi, S.M.; Gomez, C. Bluetooth Low Energy Mesh Networks: A Survey. *Sensors* 2017, 17, 1467. [Google Scholar] [CrossRef] [PubMed]
- [6] Gohel, S. Bluetooth Attendance System with Android Application for ERP. In Proceedings of the International Conference on Computing, Power and Communication Technologies (GUCON), Greater Noida, India, 28–29 September 2018; pp. 481–484. [Google Scholar]
- [7] Sthapit, P.; Gang, H.; Pyun, J. Bluetooth Based Indoor Positioning Using Machine Learning Algorithms. In Proceedings of the IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Jeju, Korea, 24–26 June 2018; pp. 206–212. [Google Scholar]
- [8] Bluetooth Core Specification Version 4.1, Bluetooth SIG, WA, USA, Dec 2013.

- [10] M. Siekkinen, M. Hienkari, J. K. Nurminen, and J. Nieminen, "How low energy is Bluetooth low energy? Comparative measurements with ZigBee/802.15.4," in Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW), Apr. 2012, pp. 232–237.
- [11] E. Toscano and L. L. Bello, "Comparative assessments of IEEE 802.15.4/ZigBee and 6LoWPAN for low-power industrial WSNs in realistic scenarios," in Proc. 9th IEEE Int. Workshop Factory Commun. Syst. (WFCS), May 2012, pp. 115–124.
- [12] R. Rondon, M. Gidlund, and K. Landernas, "Evaluating Bluetooth low energy suitability for time-critical industrial IoT applications," *Int. J. Wireless Inf. Netw.*, vol. 24, no. 3, pp. 278–290, Sep. 2017.
- [13] Bluetooth Core Specification Version 4.2, Bluetooth SIG, Kirkland, WA, USA, Dec. 2014.
- [14] Bluetooth Core Specification Version 5.0, Bluetooth SIG, Kirkland, WA, USA, Dec. 2016.
- [15] G. Patti, L. Leonardi, and L. Lo Bello, "A Bluetooth Low Energy real-time protocol for industrial wireless mesh networks," in Proc. 42nd Annu. Conf. IEEE Ind. Electron. Soc. (IECON), Firenze, Italy, Oct. 2016.

Document Information

Analyzed document	Report (6).pdf (D101563070)
Submitted	4/14/2021 10:14:00 AM
Submitted by	R.Narmadha
Submitter email	narmadha.enc@sathyabama.ac.in
Similarity	8%
Analysis address	narmadha.enc.sathya@analysis.urkund.com

Sources included in the report

SA	termpaper_pavan.pdf Document termpaper_pavan.pdf (D83778551)		1
W	URL: https://slideplayer.com/slide/16114883/ Fetched: 10/26/2019 6:04:17 PM		2
W	URL: https://www.mdpi.com/1424-8220/20/12/3590 Fetched: 4/14/2021 10:15:00 AM		3
W	URL: https://www.matec-conferences.org/articles/matecconf/pdf/2019/41/matecconf_csc201... Fetched: 10/27/2019 5:16:12 PM		2
W	URL: https://blog.csdn.net/qq_27114397/article/details/107127016 Fetched: 4/14/2021 10:15:00 AM		1
W	URL: https://baremetalprojects.com/2019/04/12/bluetooth-5-0-stairway-lights-bluetooth-m... Fetched: 11/18/2019 3:41:34 PM		5
W	URL: https://devzone.nordicsemi.com/f/nordic-q-a/39782/i-want-to-send-temperature-using... Fetched: 12/13/2019 3:01:45 PM		1
W	URL: https://infocenter.nordicsemi.com/topic/com.nordic.infocenter.meshsdk.v3.2.0/group... Fetched: 12/8/2020 5:28:31 PM		2
W	URL: https://infocenter.nordicsemi.com/topic/com.nordic.infocenter.meshsdk.v5.0.0/group... Fetched: 12/8/2020 5:28:31 PM		1
W	URL: https://devzone.nordicsemi.com/f/nordic-q-a/63304/send-string-from-server-to-clien... Fetched: 12/14/2020 3:52:01 AM		1
W	URL: https://devzone.nordicsemi.com/f/nordic-q-a/38838/sending-config-relay-set-message... Fetched: 10/23/2019 5:20:10 PM		2
W	URL: https://devzone.nordicsemi.com/f/nordic-q-a/34033/mesh-set-message-destination-address Fetched: 12/7/2019 1:18:25 AM		1