Abstract

SIMKITS: AN EXPLORATION OF TANGIBLE INTERFACE FOR PROGRAMMING AND CIRCUIT SIMULATION

Keywords: TUI, tangible programming, circuit simulation, logic simulation, cube, mobile

Tangible User Interfaces (TUIs) have increased the interactive space between human and computer. They also have developed a strong interlink between the perception and cognition of people. Several studies have performed to identify the effects of TUIs in learning and education.

SimKits is a research project that explores the possibility of having interactive cubes as an educational platform. It consists of a set of tangible cubes which detects and communicates with neighbors, day to day objects (e.g. smartphones) and a server (i.e. smartphone/tablet/laptop). It allows children to enhance their learning capabilities particularly in areas such programming and electronic circuits in a uniquely interactive manner.

This report presents the design and implementation of SimKits. Highlights of our research include the establishment of multiple BLE (Bluetooth Low Energy) connections between the clients (cubes) and the server (Smart phone) and overall functionality of SimKits. The report also describes the four main applications for SimKits (Maze Game, Tangible Programming Application, Logic Simulation and Circuit Simulation Application). This research also has a promising perspective in improving children's computational thinking. The concept can be extended to a wide range of interactive entertainment and learning applications with few modifications to the current design.

This is dedicated our beloved parents and teachers

for their kind affection and encouragement

ACKNOWLEDGEMENT

We would also like to convey our sincere gratitude to our mother of guidance Prof.(Mrs) Dileeka Dias, the supervisor of our project, for her remarkable contribution to our project. Throughout this time period, her massive support and advices should be appreciated. She pioneered our project by giving us targets and tracking us regularly. She was with us in every difficult moment as a part of our group project. We appreciate her facilitation of funding for our project.

We would also like to acknowledge our Head of the Department Dr. Ajith Pasqual for providing us the necessary guidance and allowing access to the required facilities within the department.

Our kind gratitude is conveyed to Prof. Rohan Munasinghe, the coordinator of the final year projects for his guidance and encouragement to complete tasks within the time frame. We also thank to laboratory instructors who help us lot to continue this project.

We also express thanks to academic and nonacademic staff of University of Moratuwa and our parents for helping and being with us to success this project.

Table of Contents

List of Figures

List of Tables

List of Acronyms

IoT	Internet of Things
LOS	Line-Of-Sight
BER	Bit Error Rate
ROM	Read Only Memory
RAM	Random Access Memory
MCU	Main Controller Unit
UART (USART)	Universal Asynchronous Receiver/Transmitter (Universal Synchronous and Asynchronous Receiver/Transmitter)
SPI	Serial Peripheral Interface
I^2C (TWI)	Inter-Integrated Circuit (Two Wire Interface)
RTC	Real Time Clock
Open GL ES	OpenGL (Open Graphic Library) for Embedded Systems
HCI	Human Computer Interaction
GUI	Graphical User Interface
TUI	Tangible User Interface
dTUI	Distributed Tangible User Interface
VUI	Voice User Interface
SNUI	Sensor Network User Interface
SoC	System on Chip
BLE	Bluetooth Low Energy
SIG	Special Interest Group
GATT	General Attribute Profile
ATT	Attribute Protocol
UUID	Universally Unique Identifier
GFSK	Gaussian Frequency Shift Keying
AES	Advanced Encryption System

1 INTRODUCTION

1.1 Introduction to the project

People like ‘smart’ devices which can interact with each other to do multiple tasks at the same time. However interaction alone does not enhance usability. Tangibility, affordability and physical manipulations also affect the usefulness of the device/interface.

With the development of technology people require ‘computational thinking’ [1] and interactivity. Even though programming helps to develop computational thinking [2] , complex syntaxes and texts have become a hindrance for children to master it.

This research project explores the possibility of having interactive tangible interface as an educational platform. Adapting the concepts of tangible interfaces, ‘SimKits’ is built as set of cubes which can communicate with day-to-day objects and address the difficulties in programming and circuit simulation. There are very few applications which have used tangibility and interactivity for education. ‘Tern’ [3] is one such tangible programming language which has proven to enhance the learning capability of children.

SimKits (Figure 1) consist of a set of tangible cubes and a server (e.g. Smart phone) which processes data and runs the applications. Each SimKit cube consist of a Bluetooth Low Energy (BLE) device (nRF51822), induction neighbor detection, an accelerometer, a touch sensor, a flash memory, a rechargeable battery unit and a TFT LCD display. These cubes can detect their neighbors using touch sensors or induction and communicate with the server through BLE. Android platform is used to develop four applications/games; tangible maze application, tangible programming application, logic simulation and circuit simulation application. Each application is targeted for a dedicated age group to enhance knowledge on programming and circuit simulation. Tangible maze application helps primary school children to learn basic programing concepts and reasoning, while tangible programming application is targeted for O/L and A/L students to learn complex text based programming with visual outputs. Circuit and logic simulation are for O/L and A/L students who have some basic knowledge on circuit components. All those applications process interactions in real-time and display the current results on cubes and the screen of the phone. For example; tangible programming will display errors in syntaxes or semantics during compiling and show step-by-step results during the execution.

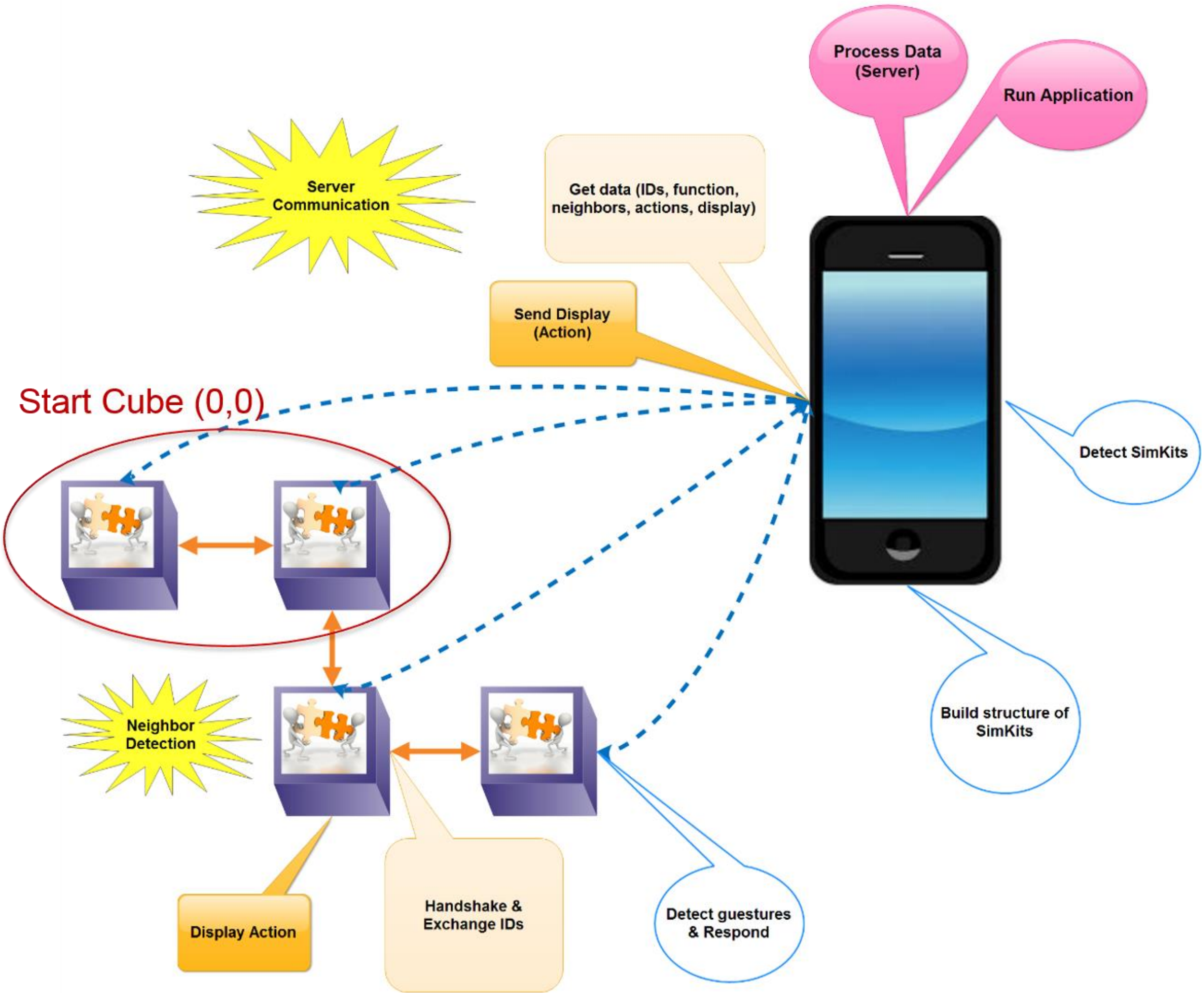


Figure 1: Overall operation of SimKits

The usability and learnability should be tested on a targeted user group through survey analysis. Still the project is in the research and development cycle and evaluating the SimKits' effectiveness is the next step.

The interaction space is not limited to educational purposes. This can also use as a wireless mouse or a remote controller or to play multiplayer games on mobile. Apart from that, cubes can act as add-ons for the mobile to show time or cancel a phone call. Due to the communication capability with day-to-day objects, the interaction space will be developed with the growth of those smart objects' technologies.

1.2 Motivation of the Project

Today many object try to be ‘smart’ by connecting to networks and perform distributed tasks through ‘internet of things’ (IoT). Similarly people also want to connect with networks to share ideas, thoughts and relax their minds. But most seek ‘social networks’, a virtual society where free thoughts and expressions face no barriers. This has made people to bond with their smart devices (especially mobile phone and laptop), which makes them interactive via networks, games, movies and messaging.

Abacus, one of the greatest invention in history; is barely used among children, due to lack of interactivity and support for modern complexity. Still children play LEGO blocks due to their tangibility and challenging nature. They also improve visual memories and reasoning of children. So we needed to add the interactivity with day-to-day objects and tangibility to an educational platform to enhance the learning capabilities of children. To help with reasoning and computational thinking, we decided to include programming for the educational platform, while retaining the powerful concepts of Abacus and LEGO blocks.

1.3 Objectives of the Project

Improving the interaction with day-to-day objects to enhance the learning capability of children (possibility of having interactive cubes as an educational platform) on programming and circuit simulation can be considered as the main objective of this project. Apart from that, overcoming the difficulties in learning programming and implementing circuits using actual devices is considered as another target of this project.

1.4 Scope of the Project

Several applications have come to the market to enhance interactivity using TUIs. But there are only few products which are developed for education using TUIs. Project was started to fulfil the need of TUI as an educational platform. Identification of the user interaction space was the critical part of the project. To achieve the objectives of project, a new hybrid platform was built based on TUIs. SimKits is developed to enhance the knowledge of children while enabling entertainment. Evaluating the usability and affordance of SimKits is very important since it may open SimKits to the modern dream worlds of children.

1.5 Practical implementation and results

To achieve the objectives of project, four applications and a cube set was built targeting primary, ordinary and advance level students. Initial cube set consist of ten cubes to support fairly complex steps in games. Android was selected for running applications on mobile. For neighbor detection induction communication and touch sensing was used while communication between server and cubes was archived using BLE.

Basic applications and target groups

- Programming
  1. Maze game - Primary school children
  2. Tangible program - O/L and A/L students
- Simulation
  3. Circuit simulation - O/L and A/L students
  4. Logic simulation - O/L and A/L students

User can interact with game (cubes) through shaking, flipping, touching cubes and gestures on mobile phone. In maze game user has to reach the goal by fulfilling the conditions (escaping wrong moves) and changing cube conditions. In tangible programming, user can make simple code by cubes and see the results while executing in real time. Circuit simulation enable users to use cubes to build circuits on mobile without limiting to any particular hardware component.

1.6 Outline of Project Report

The first chapter of this report introduces the project, its motivation and objectives. The second chapter gives an insight to the existing researches and literature related to the background of this project. These novel interfaces were introduced recently and still in the development stage. Third chapter describes about software and hardware platform selection. Next chapter discusses the methodology and implementation in details. It explain novel features of SimKits and implementation process. It also addresses the issues encountered and how they were tackled for a better design. And the last chapter conclude the project with future improvements and design possibilities.

2 LITERATURE REVIEW

The existing literature was explored in order to identify the limitations of the existing systems and to avoid the potential misleading paths. It also helped to identify the potential, novel designs and new areas to implement the project more effectively. Apart from them it helped to identify the different approaches for solving similar issues.

2.1 Human Computer Interaction

Human computer interaction (HCI) is about designing novel interactive interfaces between human and computer (any computational device), which enable user to fulfill needs and requirements efficiently and effectively. Graphical User Interfaces (GUIs), Tangible User Interfaces (TUIs, e.g. Sifteo Cubes, Audio Cubes, etc.) and Voice User Interfaces (VUIs, e.g. Google Talk, Apple Siri) are few of such interfaces which resulted from HCI.

2.2 Tangible Interfaces

Tangible interfaces have increased the physical manipulation over virtual reality to interact with digital information. Merrill et al. [4] divides the TUIs in to two categories; 'TUIs with handles operate by sensing the user's manipulation of each handle, and displaying a co-located visual representation of the data being manipulated' and 'featuring physical objects that directly embody the digital information or media that they represent'. Fitzmaurice et al. [5] introduces the concept of 'handles' which can manipulate digital world using physical objects. Merrill et al. [4] also highlight the advantages of TUIs over GUIs including the support for two-handed inputs, reduced cognitive load and increased directivity between the hand gestures and actual computation.

Marshall [6] explains the possible learning benefits of TUIs over other interfaces. Some of them are; higher knowledge gain due to interlink between perception and cognition, increased engagement and reflection, increased collaborative learning, development of thinking due to manipulation of concrete physical objects (Piagetian developmental theory) particularly in young children. Horn et al. [7] also illustrate the advantages of thoughtfully designed TUIs over GUIs in the context of informal education.

2.3 Tangible Programming

Programming helps children to learn reasoning, computational thinking and increases visual memories and numerical senses [2]. But due to the complexity of syntaxes, unfamiliar texts/words children find difficulties in learning programming [8]. MIT media lab has introduced ‘Scratch’ [9], a programming language to create interactive stories, music, games, art, and animations easily and share those creations on the web. This eliminated several barriers in learning programming and introduced a novel path for programming. Hood [10] describes an innovative method for using LEGO® bricks to teach programming and other computing concepts. They also have shown that “the use of LEGOs® increases the tactile and kinesthetic aspects of the learning experience and helps to make abstract concepts more concrete” [10].

Tangible programming is a new view point of programming to enhance the learning concepts efficiently. It interacts with physical objects by sensor technology and transforms the logic in physical world in to program logic [11]. Wang et al. [11] have divided the tangible programming in to two types based on the fact tangible blocks are built with electronic modules or not. They also have introduced a tangible programming tool 'T-Maze' which enable 5 to 9 years old children to create their own maze maps and complete some maze escaping tasks. It uses a camera to catch the programming sequence of blocks & sensors and give real-time feedback on sematic correctness.

On the other hand Horn et al. [3] have introduced ‘Tern’, a tangible programming language for middle school and late elementary school students, consists of wooden blocks to form physical computer program (offline). They have added programming concepts such as action commands, loops, branches and subroutines for advance usage. Tern differs from other programming languages due to pure symbolic representation without texts. It also can be used to control virtual roles and control robots. In [12] Horn et al. have identified situations where tangible interaction proves less useful and alternative interaction styles by moving towards ‘hybrid approach’.

Wang and et al. [11] explore a new tangible programming tool ‘TanPro-Kit’ which was designed for children aged 5 to 9 focusing on playfulness, learnability, affordability and handleability. It consists of programming wooden blocks and a LED pad to simulate and create maze games.

2.4 Cubes

There are different types of researches which are related with the TUIs. The concept of cube-based tangible user interfaces and it’s usage in learning applications is not new to the world. Most tangible interfaces take the shape of cubes due to the handlability and ease of piling. It also enable user to feel the 3D nature of tangible interfaces. There are several cube interfaces including ‘Sifteo cubes’ [4] and ‘Audio cubes’ [13] which are used for useful digital manipulations. “Sifteo Cube (Siftable [14])” is one commercial product with interactive games, calculations and neighboring detection. It comes with three cubes and one server to play interactive games. Users are enabled to configure and install new games to server and build games on Sifteo platform.

Lucia et al. [15] have developed learning cube which act as a general learning platform while supporting multiple choice test system and vocabulary trainer. The ‘i-Cube’ [16] which contains the details of a digital manipulative consisting of two main applications: ‘MusiCube Arranger’ and ‘Spelling Cube’. In here they have compared the differences with the manipulatives and focused on giving a more interactive learning tool for children. Orientation and Motion sensing with full 3-d spatial awareness has been implemented within these i-Cubes. Active Cubes [17] enable users to construct and interact with 3D modelling environment while allowing them to simulate. Due to its real time multimodal and spacial capabilities it can be used in interactive entertainment applications.

‘Blinky Blocks’ [18] is a novel cube implementation which talks about requirements of physically distributed ensembles and discusses the use of the distributed programming language ‘Meld’ to program ensembles of blocks. It discusses the hardware design used to create 100 of modules while keeping the connectivity between neighbors. Each block consists of 8-bit microcontroller, RGB LEDs, an accelerometer, a magnetometer, a speakers and a microphone. From the concepts of modular robotics and tangible interaction, they managed to run any number of blocks without knowing the total number of blocks. Each block can detect its current orientation and talk with neighbors independent of topology. Programmer also does not have to think about topology changes since program has reconfigurable, computational control, scalable and localization capability. So when a cube is added, removed or changed Meld runtime will automatically update the rainbow (test program) and keep it consistent with the current ensemble.

2.5 Logic simulation and circuit simulation

Circuit and Logic Simulation can be considered as one of the most tedious task to do in terms of simulation aspect. SPICE has been already used as a tool to aid in the teaching of basic circuit techniques. Most of the simulation applications are not tangible and makes it less attractive to children. Due to this reason the concept of TUI in circuit & logic simulation comes into play where children can interactively learn the basics of electronics while entertaining themselves.

There are some researches which are involved in TUI for circuit simulation & logic simulation. TOUCHSPICE [19] is one of such application where they use cubes as TUI to simulate circuit simulation.

2.6 Existing products and interfaces

Table 1 shows the existing products/interfaces and their capabilities. But SimKits tries to combine certain capabilities of those to fill the efficiency of interactivity with day to day objects through programming and circuit simulation.

Table 1: Existing products/interfaces and their features

	Tangibility	Programming Interface	Simulation interface	Gaming Interface	Real-time interactivity	Multi functionality	Interaction with day to day objects	3D awareness
Sifteo Cube	√			√	√	√		
Scratch		√	?					
Active Cubes	√		√		√	√		√
Audio Cubes	√		√		√	√		
Tern	√	√		√				
T-Maze	√	√		√				
i-Cube	√				√	√		√
TanPro-Kit	√	√		√		√		
Sifteo SPICE	√		√		√	√		
TouchSpice	√		√		√	√		

3 DESIGN SELECTIONS

3.1 Software (Mobile application)

In any modern electronic device, software plays a major role same as the hardware platform. Therefore it is compulsory to choose the correct and more suitable software platform in order to increase the performance in terms of efficiency, productivity and accuracy.

3.1.1 Android

Though Android Operating system in smart phones is still relatively new to the market, it has achieved an immense progress within the last couple of years. Android was designed by Google for mobile phones which are easier to handle. Main reasons behind selecting the Android over others can be categorized as following.

- It is open source – Anyone can access the android platform and develop applications as their own choice.
- Multitasking – Many applications can run simultaneously. For an instance while browsing Facebook we can listen to a song.
- Easy to learn – Anyone who has a programming knowledge can learn Android easily since it is based on java.
- Most of the Smart phones have Android as their Operating System.

Due to the above reasons we selected Android as our main software platform to build the applications which are interactively communicating with the SimKits.

3.1.2 iOS

iOS is developed by Apple Inc which only support Apple products. It is not open source and need Apple computer and license to create and install applications to i-phones and i-pads.

3.1.3 Selection

Table 2 contrast the differences between Andoid OS and iOS. Due to lot of advantages in programming and community, Android was selected for application development.

Table 2: Comparison between Andoid OS and iOS (Source : http://www.diffen.com/difference/Android_vs_iOS)

Developer	Google	Apple Inc.
Source model	Open source	Closed, with open source components.
Customizability	A lot. Can change almost anything.	Limited unless jailbroken
Available on	Many phones and tablets, including Kindle Fire(modified android), LG, HTC, Samsung, Sony, Motorola, Nexus, and others. Also, Google Glasses	iPod Touch, iPhone, iPad, Apple TV (2nd and 3rd generation)
OS family	Linux	OS X, UNIX
Programmed in	C, C++, Java	C, C++, Objective-C
Open source	Kernel, UI, and some standard apps	The iOS kernel is not open source but is based on the open-source Darwin OS.
Interface	Touch screen, Smartwatch	Touch screen

3.2 Hardware selection

As described above, the hardware platform is a key role in any electronic product which heavily contributes to the performance of it. Therefore there should be a screening process before applying any hardware platform for a particular product. The features of the product must be carefully analyzed and according to the expected outcomes, a suitable hardware platform must be selected.

3.2.1 Bluetooth

All the Bluetooth devices operate within the 2.4GHz frequency band which is same as microwave frequencies. The Bluetooth technology divides the 2.4 GHz band into 79 channels and employs channel hopping techniques such that Bluetooth frequencies are always changing which makes Bluetooth different from other technologies.

3.2.1.1 Bluetooth Low Energy

Bluetooth Low Energy or Bluetooth Smart came to the market as a part of Bluetooth 4.0 Core Specification. It can be considered as a highly optimized version of Bluetooth classic which had several limitations and short comings. But in reality BLE has a totally different architecture which makes it more optimized in terms of power consumption. This was originally designed by Nokia and later it is being adopted by Bluetooth Special Interest Group (SIG). The hidden truth behind the immense growth of BLE is correlated with the growth of mobile computing and specially smartphones. BLE was adopted by Apple and Samsung at the earliest stages and it helped to the massive implementation of BLE.

- Optimized Gaussian Frequency Shift Keying (GFSK) – BLE uses the GFSK modulation scheme with a modulation index between 0.45 and 0.55 which is higher than Bluetooth Classic.
- Tight Security – BLE authentication and Encryption is implemented based on a 128-bit Advanced Encryption System (AES) developed by U.S. government to secure the data. Its dual-mode capability can be considered as another main aspect.
- Adaptive frequency hopping – BLE uses the same frequency hopping technique which is used for Bluetooth Classic which minimizes the interferences from other technologies within the band.

BLE can be considered as a totally new technology which has efficient discovery and connection set-up and short data package transferring. Figure 2 indicates the Block diagram of the Bluetooth Low Energy Core which has the basic components.

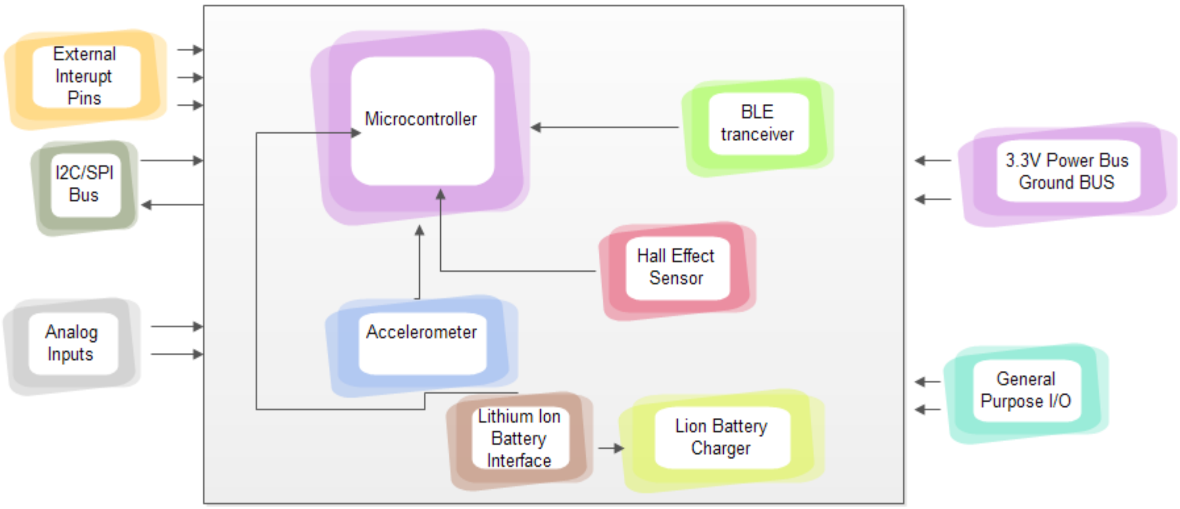


Figure 2: BLE Core Block diagram

3.2.1.2 Reasons behind selecting BLE platform

There were several reasons behind selecting BLE over other wireless standards other than above specified technical aspects. Those reasons can be listed as following,

- Low Power Consumption
- Less Expensive
- New arising technology which is compatible with all the smart devices.
- Ease of use and Integration

3.2.1.3 Comparison between BLE and other wireless standards

For the wireless devices which are seeking for low power rather than going for high speed transfer rates, BLE is the natural choice, Bluetooth Smart Core can be considered as a powerful tool which is offering a wide range of applications including iPhone and Android applications. Not only that, but also it is embedded as Bluetooth Low Energy module which links to cutting edge technology devices including Tablets, iPods and Smart phones. Its small size helps to use it in many applications. Table 3 clearly indicates the plus and negative points of BLE over other technologies.

Table 3: Comparison between existing wireless technologies (Source: http://www.iebmedia.com/index.php?id=8294&parentid=63&themeid=255&hft=67&showdetail=true&bb=1)

	Bluetooth (Classic)	WLAN	802.15.4 (ZigBee)	BLE
Data throughput	+/-	++	-	-
Robustness	++	+/-	+/-	++
Range	50-1000m	50-300m	75m/node	10-300m
Local System Density	++	-	+/-	++
Roaming	+/-	++	-	+/-
Large Scale network	-	+/-	++	++
Low latency	++	+/-	-	++
Paring speed	++	+/-	+	++

Power consumption	+	-	++	+++
Cost	+	-	+	++

3.2.1.4 GATT Protocol

Generic Attribute Profile defines how the data is being exchanged over the BLE connection. GATT uses the Attribute Protocol (ATT) which is acting as the transport protocol to exchange data between the devices. There are two main sectors known as Services and Characteristics. All the data is hierarchically organized in sections which are known as Services while pieces of users data is known as Characteristics.

Client

The GATT client corresponds to the ATT client. The GATT client must first perform service discovery in order to know about the server's attributes. After completing the service discovery then it can read characteristics or write characteristics which are found in the server.

Server

Similarly GATT server corresponds to the ATT server which receives requests from the clients and sends response or acknowledgements back. Every BLE connection must have at least one sever which can also send server initiated updates when necessary.

UUIDs

UUID stands for Universally Unique Identifier which is a 128- bit (16 bytes) number. Normally a UUID is considered as a globally unique identifier. Not only on Bluetooth, but also in many other applications UUIDs are used immensely.

Figure 3 clearly describes the GATT server hierarchy where it can have multiple services and within each service it can have multiple characteristics consisted of descriptors.

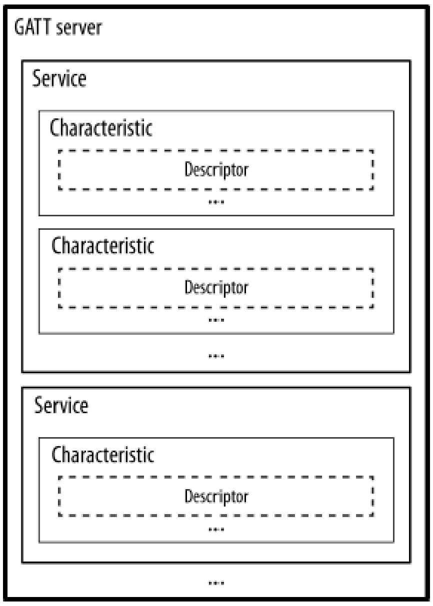


Figure 3: GATT Hierarchy

3.2.1.5 Multiple BLE Connections

In order to have multiple BLE connections it is said that theoretically for each connection we must have a separate BluetoothGattCallback which uses a great deal of resources which is inefficient. But using only one BluetoothGattCallback and multiple instances of BluetoothGatt we were able to successfully establish multiple BLE connections between clients (SimKits Cubes) and the server (In our case a smart phone).

Here we have used only one Service for all the sensors (Touch sensor, accelerometer etc.) and within the service BluetoothGatt instances have created and instantiated. Within the BluetoothGattCallback we have attempted to discover services for each BluetoothGatt instance and all the states of the connection (whether connected or disconnected) were updated using a broadcastUpdate method.

3.2.2 Communication Transceiver

In SimKits, all the processed sensor data has to be transmitted to server for further processing and results should transferred back on cubes. Since BLE has been chosen there were several transceivers to achieve this purpose. But selecting the component had to be decided on power consumption, output transmit power, receiver sensitivity, achievable data rate, library support, ease of using (e.g. soldering), etc. Table 4 compares the widely available BLE support transceivers and their characteristics.

Table 4: Comparison between considered BLE transceivers (Source: DigiKey and Nordic web sites and corresponding data sheets)

	CC2540 (CC2540F256RHAR)	CC2541 (CC2541F256RHAR)	nRF8001 (NRF8001-R2Q32-T)	nRF51822 (NRF51822-QFAA-T)
Input Voltage	2 - 3.6V	2 - 3.6V	1.9 - 3.6V	1.8 - 3.6V
RX Current (Max)	22.1mA	20.2mA	14.6mA	13.4mA
TX Current (Max)	31.6mA	18.2mA	12.7mA	16mA
Power output (Max)	8dBm	0dB	0dB	4dBm
Receiver sensitivity (rate = 1Mbps)	-93dBm	-93dB	-93dBm	-96dBm
Frequency	2.4GHz	2.4GHz	2.4GHz	2.4GHz
Data Rate	250 kbps , 500kbps, 1Mbps	250 kbps , 500kbps, 1Mbps, 2Mbps	1Mbps	250 kbps , 1Mbps, 2Mbps
BER	< 1%	< 1%	< 1%	< 1%
Bluetooth Protocol	BLE v4.0	BLE v4.0	BLE v4.0	BLE v4.0, Gazell (nRF24L01)
SoC	In built 8-bit 8051, 256kB Flash, 8kB RAM	In built 8-bit 8051, 256kB Flash, 8kB RAM	No SOC	In built 32-bit ARM Cortex M0 16MHz, 256kB Flash, 16kB RAM
Interfaces	USB, 2 * USART	I2C interface , 2 * USART	SPI, ACI, DTM UART	Pin configurable I2C, 2 * SPI, UART

Library Support	Very High, Free samples	Very High, Free samples	High	Low, but SDK with examples
BLE stack	Proprietary	Proprietary	In-built	Open use softdevices (S110, S120, S130)
Cost (20 units) per unit	$4.61	$4.25	$3.83	$4.17

Texas Instruments' CC2451 and Nordic's nRF51822 are competitive chips with SoC and BLE stacks. Even though nRF51822 need $100 worth SDK to purchase with lesser support, it consumes lesser power and can be programmed using (free version is restricted) low cost Keil IDE instead of highly costly IAR IDE (one month trial). It also has lot of customizable features with powerful in-built ARM processor and free soft devices with dedicated BLE stack from user code. So nRF51822 was chosen as both processing unit and transceiver. But due to the complexity of antenna placement and soldering, it was decided to use nRF51822 breakout board available on eBay (see Appendix AnRF51822 module imported from eBay).

3.2.3 Neighbor Detection Sensors

One of the main tasks of project was to map the relative locations with each SimKits to find the connections and draw the results on server (e.g. mobile phone) screen. Since cubes could be packed very closely (Figure 4), each cube should be able to detect its four neighbors (assuming cubes are used in 2D) accurately when there is neighbor change and inform to server instantly.

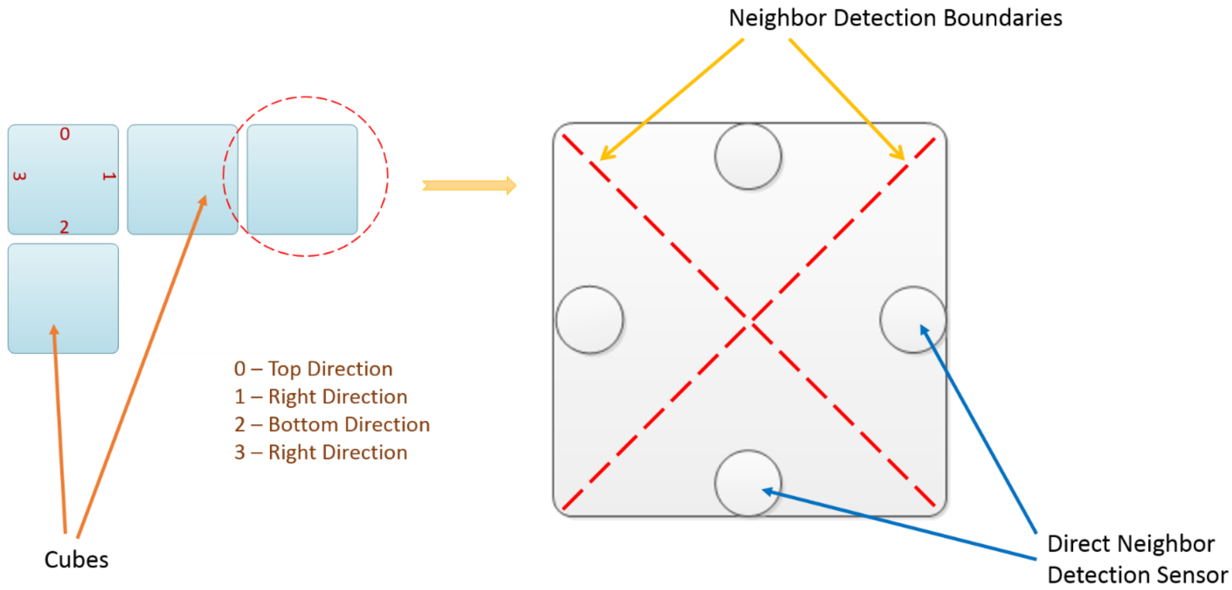


Figure 4: Optimal (direct) neighbor detection sensor placement

When cubes are closely packed (Figure 4) neighbor detection sensors' sensing capabilities should be confined to limited distance to minimize intra-cube interference which eventually determined by cube dimensions. Following subsections will describe available technologies for neighbor detection with their capabilities and limitations.

3.2.3.1 IrDA Infrared Transceivers

The Infrared Data Association (IrDA) is an industry driven interest group [20] which sets standard protocols for wireless infrared data communication. These standards are specially focused on 'last meter' data transferring using LOS with very low BER.

General characteristics of IrDA transceivers [20]:

- Range: standard: 1 m; low power to low power: 0.2 m; standard to low power: 0.3 m, The 10 GigaIR also define new usage models that supports higher link distances up to several meters.
- Angle: minimum cone ±15°
- Speed: 2.4 kbit/s to 1 Gbit/s (depend on physical layer protocol)
- Modulation: baseband, no carrier
- Infrared window
- Wavelength: 850–900 nm

e.g: For SimKits following TFBS4711 sensor was considered (Table 5);

Table 5: Characteristics of TFBS4711

Data Rate	115.2kbs (SIR)
Supply Voltage	2.7 V ~ 5.5 V
Idle Current, Typ @ 25°C	75µA
Standards	IrPHY 1.2
Size	6.0mm x 3.0mm x 1.9mm
Link Range, Low Power	1m
Unit Price (80 units) – DigiKey	$3.48

Pros:

- Can configure easily to support UART
- Low BER due to LOS and hardware support
- Compact modules with small dimensions

Cons:

- Since range is about 20cm , reducing it to 2 cm is challenging
- Due to cone angle, inter-cube interference has considerable effect
- Need LOS without any obstacles
- Need special microcontroller to support four IrDA protocols
- Higher cost due to the need of four sensors per cube
- Interference from other light sources

3.2.3.2 NFC/RFID transceivers

RFID transceivers are active RFID modules which can act as both writer and reader. For SimKits each transceiver must transmit its neighbors and read its neighbors using NFC. Depending on the frequency of operation and detection range, commonly used 13.56MHz frequency is considered.

Table 6 describes the characteristics of PN512 [21] without antenna

Table 6: PN512 characteristics

RF Type	Read / Write
Frequency	13.56MHz
Link Range	Up to 50mm
Max Current	100mA
Features	ISO14443-A, ISO14443-B
Supply Voltage	2.5 V to 3.6 V
Unit Price (80 units) – DigiKey	$5.75

Pros:

- Support 8-bit parallel interface, SPI interface (up to 10Mb/s) , Serial UART (up to 1228.8kBaud), I²C interface (up to 3400kBaud) for direct reading
- Support ISO/IEC 14443 A/MIFARE, ISO/IEC 14443 B Read/Write modes
- Less interference and shorter distance
- Programmable I/O pins

Cons:

- Higher cost due to the need of four sensors per cube
- Miniaturizing the cube will affect the range of communication

3.2.3.3 Induction Transceivers

“An inductor capacitor circuit (LC circuit) acts as a transmitter as well as a receiver on inductive filed” [22]. Jeevan and David [22] have examined different possible configuration of those sensor circuits and developed several protocols to act as a transceiver.

Pros:

- Very low cost due to lesser components (L,C and op-amps)
- Can configure range arbitrarily
- Can control intra-cube interference easily

- Can configure to support multiple protocols
- Can work through solid plastics or other materials
- No need of special ‘windows’ unlike IR transceivers

Cons:

- No off-the-shelf sensor
- Need to determine configurations according to requirements
- Lack of references

3.2.3.4 Selection

Table 7 compares and contrasts the different characteristics of available neighbor detection sensors.

Table 7: Comparison between neighbor detection sensors under normal conditions

	IrDA Infrared Transceiver (TFBS4711)	NFC/RFID transceiver (PN512)	Induction transceiver
Power Consumption	Very Low	High	Low (controllable)
Intra-Cube interference	None	Low	Low (controllable)
Inter-Cube interference (distance between < 2 cm)	Low	None	Very Low (controllable)
Inter-Cube interference (distance between > 2 cm)	Very High	None	Very Low (controllable)
Interfaces	IrDA/UART	I2C/SPI/UART	customizable
Communication protocol	IrPHY	ASK	Custom
Cost per cube	High	Very High	Low
Accuracy	High	Very High	Low

Due the cost effectiveness and configurable protocols for low power consumption, novel induction transceivers were used for neighbor detection.

3.2.4 Microcontroller Selection

Microcontroller is the brain of each cube. It should process each event and handle them correct order in order to smooth the data flow among sensors, actuators and transceivers. It also has to make sure that each sensor get sufficient time to read/write data and function correctly without interfering to other modules. One of the main concern in selecting microcontroller is the capability to handle sensors, process data and transmit processed data (sufficient ROM, RAM, Flash memory, processing speed and interfaces) Another main consideration for SimKits is the power consumption (low) and operation time (larger).

Table 8 compares the three different MCUs which suit for SimKits.

Table 8: Comparison of suitable microcontrollers for SimKits

	ATmega328P (ATMEGA328P-AU)	PIC24FJ128 (PIC24FJ128GA306)	nRF51822 (NRF51822-QFAA-T)
Core size	8-bit	16-bit	32-bit
Core processor	AVR	Modified Harvard	ARM
Flash Memory	32KB (16K x 16)	128KB	256KB
RAM	2K x 8 bits	8K x 8 bits	16K x 8 bit
Speed	20MHz	32MHz	16MHz
Active power (Max under standard conditions)	0.2mA	0.15mA	16mA
BLE support	NO	NO	YES
Interfaces	I²C, SPI, UART/USART	I²C, IrDA, LIN, SPI, 4 * UART/USART	Pin configurable I2C, 2 * SPI, UART
Supply Voltage	1.8 - 5.5 V	2 – 3.6 V	1.8 – 3.6 V
Cost (20 units) per unit	$3.16	$3.37	$3.17
Programming IDE	Arduino (free), IAR	MPLAB (free version)	Keil IDE (free version)
Recommended Programmer	AVRISP	PickKit3	JLink

Considering the requirements of project (interfaces, speed and power), nRF51822 is highly appropriate due to its BLE support, low cost (no need of additional BLE transceiver) and acceptable MPU parameters.

3.2.5 Accelerometer

The need of using an accelerometer is to detect tilt, shake, and orientation of the cube. Most of accelerometers provide those features as their in built functions and some of accelerometers does not provide them. So the user has to use their own algorithms to detect those gestures using accelerations in each axis. Some accelerometers provide only analog output for each axis (X, Y and Z). Due to the limited number of pins in NRF51822 SOC, we could not go to that kind of accelerometer. So the best option was MMA7660 which is widely available in the market for lower prices. Although it does not give higher resolution acceleration outputs for each axis, we did choose it because the capability of gesture detection through interrupts. It has limited number of registers and initial settings are comparably easier than other accelerometers who communicate with I2C.

3.2.6 Touch sensor

The need of touch sensor is to detect neighbors and improve the interactivity of cubes among users. We had to use the touch sensor instead of using a LCD with touch sensing because of design limitations. LCD with touch sensor are quite larger than the size limitations of our cube design. So we had to move for a touch sensor to fill the lack of LCD with touch capability. For our application we just need only 4 touch pins. But we used MPR121 touch sensor which have 12 touch sense pins for the future development purposes. It communicate with I2C so we can share the I2C pins which used to communicate with accelerometer.

3.2.7 LCD

As mentioned earlier we could not find any suitable size LCD with touch sensing capability. So we looked for a suitable size LCD which fits into our casing. The only one LCD we found was ST7735 LCD module 128*160 pixels. OLED displays were deselected because they were not widely available in the market with the size of what we needed. ST7735 is a single chip LCD driver with 262K colors. This LCD can be directly connected to microprocessor with SPI/ 8 bit/ 9 bit/ 16 bit/ 18 bit parallel interface. It stores display data in its RAM (128*160*18 bits). Color depth was software programmable and it was really helpful in flash memory utilization. We used SPI interface to communicate with LCD module although it is slower than using parallel interface since we had to utilize the pins of NRF51822 module.

3.2.8 Power supply

The battery is LIPO 3.7 V 400mAh. So it needs to regulate to 3.3v and recharge. To regulate the battery voltage to 3.3V, MIC5219 is used. It gives 500mA output regulated current which is sufficiently enough for the normal working condition of the cube. It has the ability to produce a low noise output voltage, reverse polarity protection, current and thermal limiting and near zero shut down current.

The battery needs to be charged whenever its voltage is lower than 3.4V. It is not practical to remove the battery from cube every time when it needs to be charged. Solution was to add a charging circuit to the PCB design. MCP73831 was selected as the charger.

3.2.9 PCB design

Altium 14 was used to design the PCB of the cube. Before moving in to the PCB design, we tested all hardware components with NRF51822 on a breadboard to ensure that there are no bugs in the hardware design. Once the hardware platform was confirmed, we moved to draw the schematic on Altium software.

Most of the libraries for components were not there in Altium so we had to design them ourselves. The biggest challenge was to shrink the PCB as much as possible. So we used smallest surface mount components which can be soldered by hand. PCB is double layered to reduce the size of the PCB. Multilayer PCB could shrink the PCB to twice the double layer. But multilayer PCB manufacturing cost is much higher than that of double layer PCB.

PCB routing phase was the most difficult and time consuming part of PCB designing. Routing was done by grouping components into certain areas and placing each net in reasonable places to avoid noises and other thermal impacts. Ground plane was added to reduce noises in the design. The top layer and the bottom layer of the routed PCB are displayed in the APPENDIX C.

4 METHODOLOGY AND IMPLEMENTATION

4.1 SimKits

SimKits consists of set of tangible cubes which manipulate states through interaction and a server which does data processing and result simulation. As a proof of concept, authors decided to implement four different games/applications to demonstration the use of SimKits as an educational platform.

Games/Applications

- Tangible Maze Application
- Tangible Programming Application
- Logic simulation Application
- Circuit simulation Application

4.1.1 Connectivity and Operation

Figure 5 activity diagram describe the overall operation of SimKits (cubes and server) in a compact form.

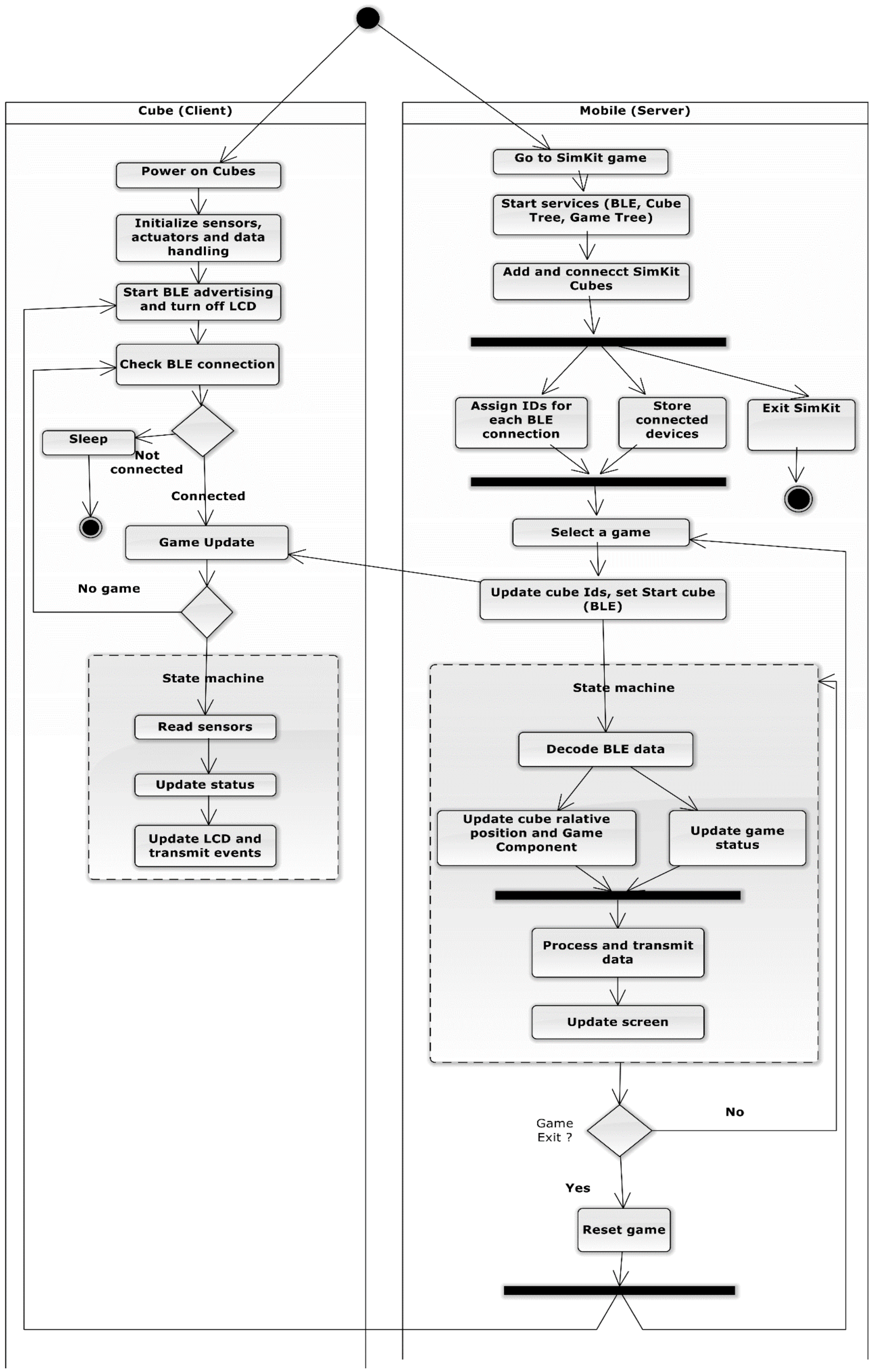


Figure 5: Activity diagram of SimKits

As shown in Figure 5 after adding cubes to server's network (i.e. connected through BLE) it will automatically select a 'Start' cube (first cube connected through BLE) to determine the origin of coordinates for mapping and drawing. When a new cube is placed near 'start' cube it will be added to 'Cube Tree' (a tree which keep the relative locations respect to 'start' cube) as shown in Figure 1.

4.1.2 Interactions on Cubes

When a cube is in a game state it will detect gestures though touch sensor and accelerometer and trigger events. According to those triggers LCD will be updated to display corresponding image. Every image on LCD has three properties (Figure 6), 'game type', 'game operator' and 'game component', except 'start' which is unique for all games which start/execute the game actions.

- Two consecutive flips – go to 'game change' or 'keep current game' request screen
- Flip – change 'game operator'
- Shake – change 'game component'

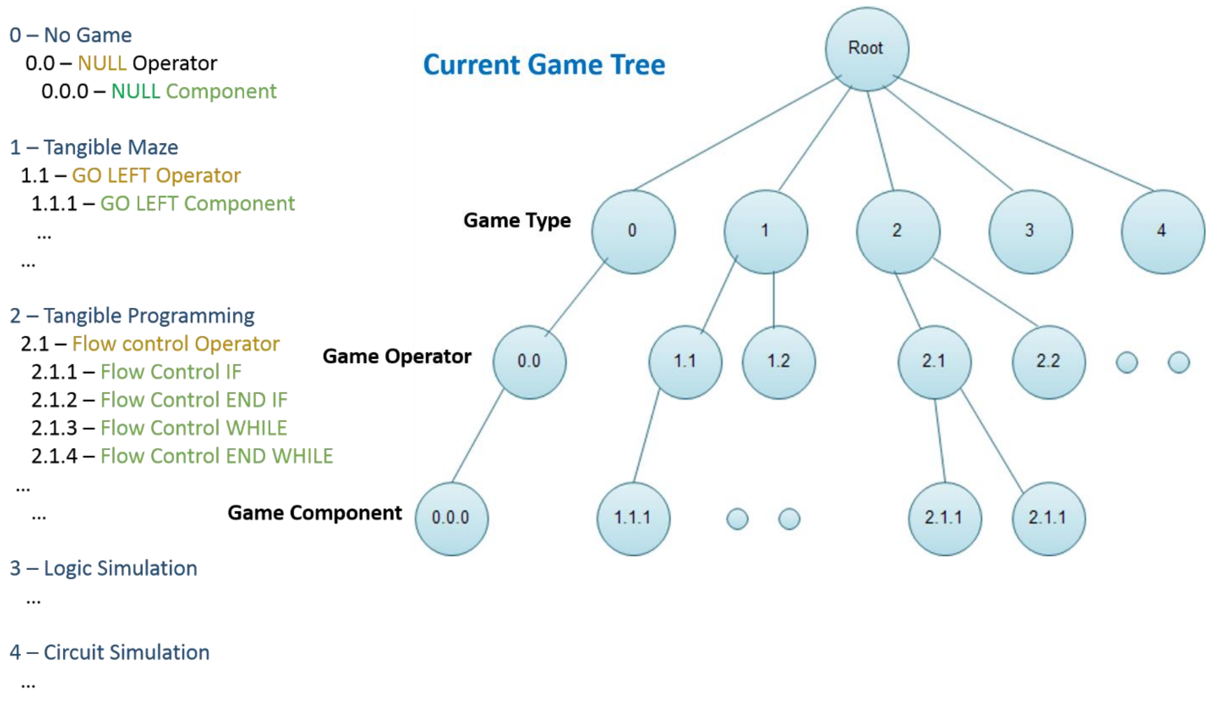


Figure 6: Game tree

4.1.3 Interaction on Server (Mobile)

User can interact with SimKits not only using the interactions mentioned in 4.1.2 but also touch and shake on mobile. But those are determined by the game application. Table 9 shows the available interaction on mobile.

Table 9: Mobile interaction on applications/games

Game	Interaction	Action (see Figure 6)
Tangible Programming	Single touch (tap) on start	Update/compile/execute program
	Single touch (tap)	Change game component
	Long touch (press)	Change game operator
Circuit simulation	Single touch (tap) on component selection	Change game operator
	Single touch (tap) on drawn component	Change value of component

4.2 SimKit Cubes

Cubes communicate with the server (e.g. Android smart phone) using BLE and exchange data with each other using neighbor detection. They update their status from sensor information while updating the display and sending status' changes to server. Cubes act as physical handlers for virtual world inside game scenarios to manipulate game states and simulate results.

4.2.1 Operation

Figure 7 illustrate the operation of cubes using state machine which consist of sub-state machines. Those state machines make sure each sensor/actuator/transceiver get enough MPU time to process data and transmit.

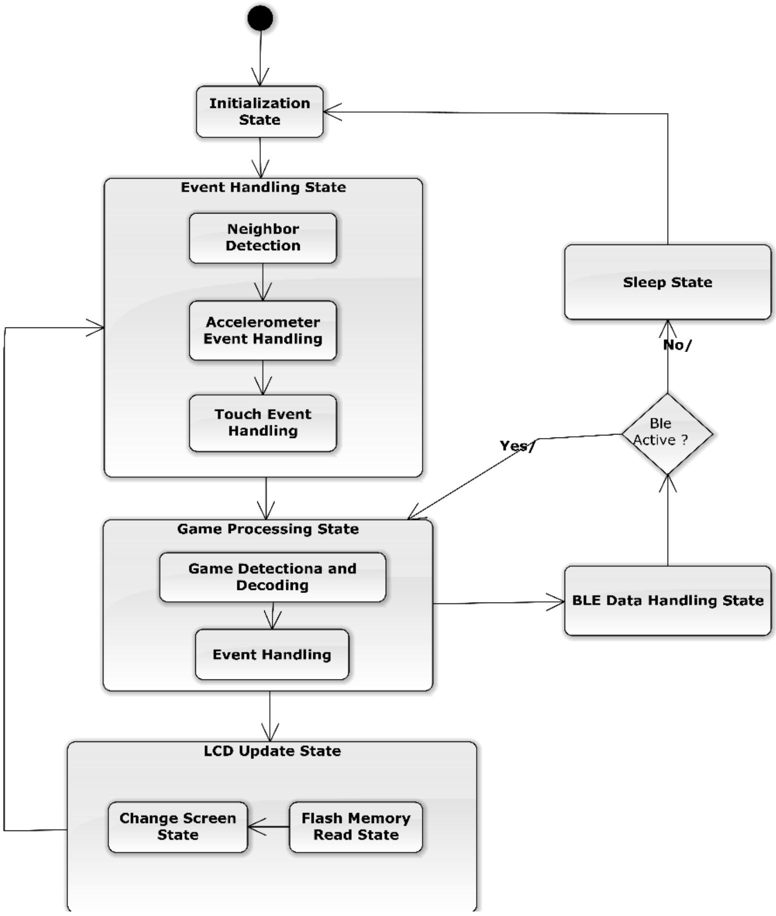


Figure 7: State machine of SimKit Cubes

4.2.2 Hardware Implementation

Hardware high level system diagram is shown in Figure 8, indicating the systems and subsystems which cooperate with each other directly or indirectly to fulfil the functional requirements of SimKits.

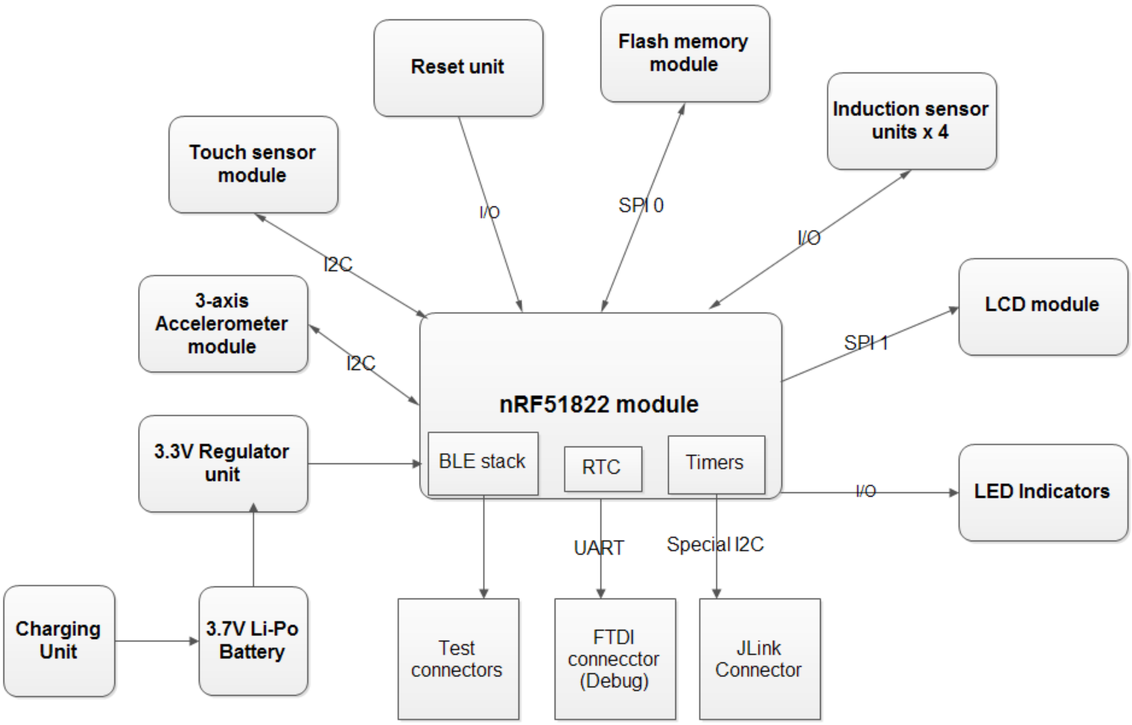


Figure 8: Hardware Interconnection

4.2.3 Casing and PCB

Casing is the main key to interact the user in appearance. User basically sees the product and after feels its functions. So coming up with good design is a huge challenge with casing design. But according to product cycle outer design is developed with the development of its features. The better solution is come up in developing stage and not in the initial stage. Therefor we modeled the design in solid works as shown in Figure 9. Basic requirement was the cubic shape. We had to place an open on top surface for the LCD display. There are also small openings for switch, battery charger and touch pins. It should be like a box because we had to keep circuit board inside the design. So we first of all take the dimension of the box and cover.

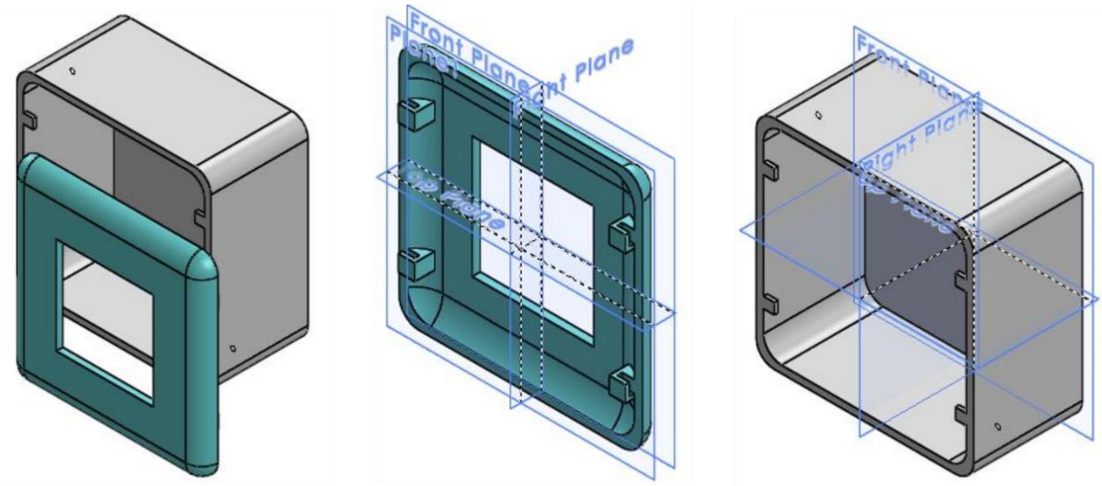


Figure 9: SolidWorks design of casing

Due to the required huge cost we had to abandon the idea of manufacturing these cubes using 3D printing or Rapid prototyping. Since this is a researched based product, spending huge amount of money for a cube might be a waste in this stage. Then we had another option to buy normal boxes from shops which are widely available. As the alternative way we bought boxes which are suitable for our shape and redesigned by milling machine to make it as our casing (Figure 10).

Figure 10: Redesigned cube casing

Figure 11 shows the populated PCB of SimKits highlighting the main components.

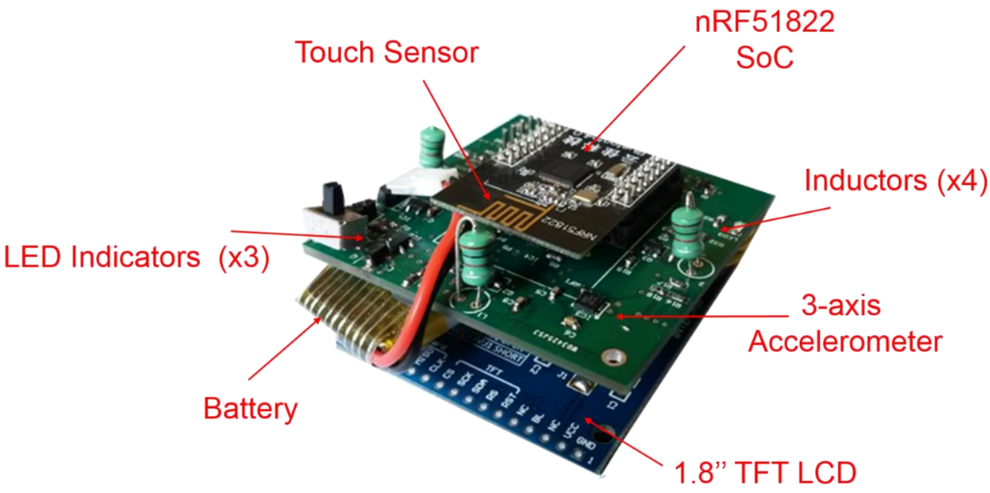


Figure 11: Complete hardware platform

4.2.4 Library writing and modification

Due to the lack of library support following libraries (Table 10) have been created/modified.

Table 10: Libraries modified or created

Library	Purpose	Modified	Added
Ble service (ble_nus.c)	support BLE RX and TX	YES	
LCD (lcd_1_8_tft.c)	Update and draw images, shapes, characters		YES
Flash memory (memory_Chip.c)	Read and write to flash		YES
Bridge (flash_lcd_bridge.c)	Handle connection between flash memory and LCD		YES
Accelerometer (mma7660.c)	Get readings, interrupts from sensor		YES
Induction Communication (induct_radio.c)	TX and RX through induction transceivers		YES
Timers (timer.c)	Global timer with multiple instances (32)		YES
Data structures (bit_frame.c, stack_buffer.c,	Handle data efficiently		YES

cube_event_fifo.c, cube_game.c)			
UART, SPI, TWI (nrf_uart.c, …)	To support requirements of SimKits	YES	

4.2.5 *Induction Communication*

As mentioned in the (chap. Induction Receivers) induction communication has several advantages over the other existing transceivers. But [22] does not specify any suitable configuration for a particular case and components has to be selected depending on the requirement.

4.2.5.1 *Operation*

The simplest transceiver circuit using induction is shown in Figure 12Figure 12: Simple LC circuit used in induction communication.

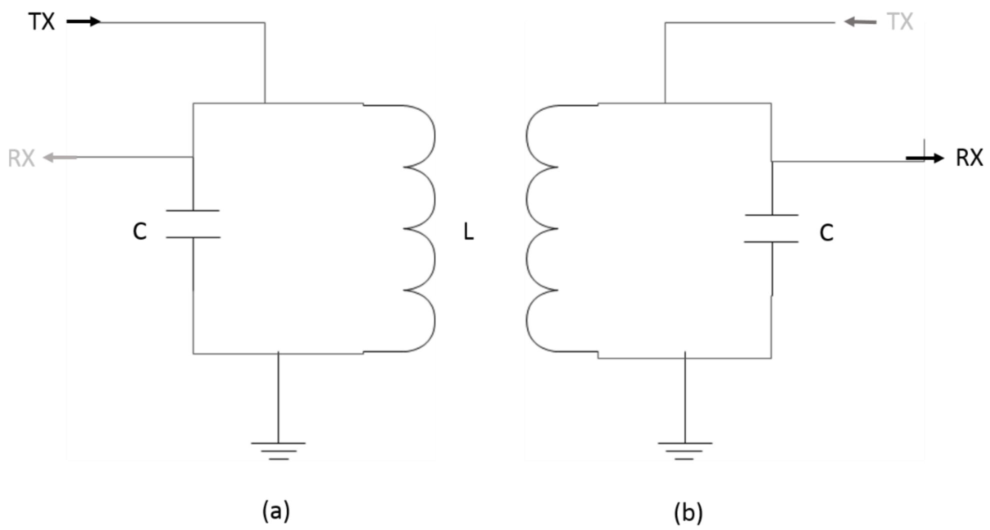


Figure 12: Simple LC circuit used in induction communication

When a pulse/pulse train is applied to the 'TX' pin of parallel LC circuit (Figure 12 (a)) (a.k.a tank circuit) it will resonate in its resonance frequency according to $f = \frac{1}{2\pi\sqrt{LC}}\ Hz$. Then voltage change across inductor has a damped oscillation or damping wave depending on the LC circuit's characteristic equation. When an identical parallel LC circuit (Figure 12 (b)) is placed parallel to the L in transmit circuit Figure 12 (a), 'hit's will be induced on receiving circuit Figure 12 (b) as shown in Figure 13. The amplitude of hits depends on distance between inductor coils and medium properties ([22]) and time gap is determined by the larger pulse gap (not consecutive pulses) between input pulses/pulse train. Using these hits the transmit signal can be determined after amplifying and filtering the received signal at ‘RX’. But to achieve this only one unit (either transmitter or receiver) should be active enabling half-duplex communication.

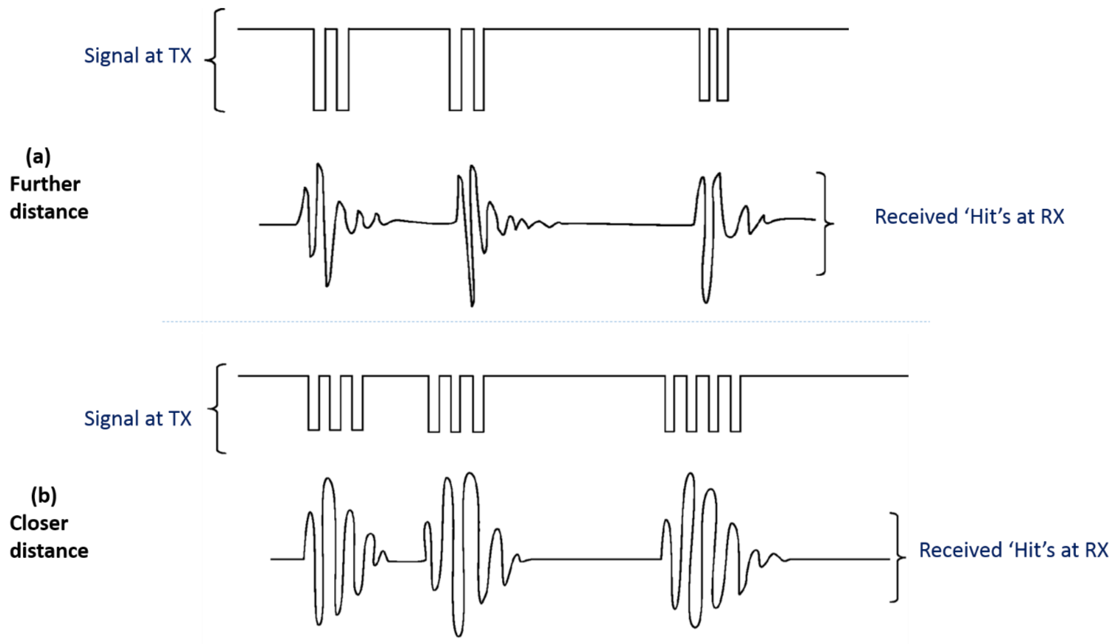


Figure 13: [22] Signal characteristics of parallel LC circuit for pulses at (a) further distance (b) closer distance

The time gap between pulse bursts can be used as symbol representation (encoding/modulation) for induction communication. For error correction, a suitable channel coding scheme should be set up with suitable synchronization protocol by detecting start and end characters of the received signal.

4.2.5.2 Implementation

Due to the interference of BLE stack with normal operation, it was determined to use one of hardware RTC available (nRF51822 has two RTCs) for neighbor detection instead of timer interrupts. Since it had minimum 31us cycle time (=T), capacitor and inductor was determined to support resonance frequency of 1/31us ≈ 32kHz. From practical experiments and theoretical knowledge a new circuit, Figure 14 was designed (which is not in [22]) supporting induction communication up to 20 mm. But its BER was not extensively studied due to time limitations. Main difficulty designing the circuit was to find op-amp which has higher slew rate and operate 0 - 3.3V which operate compatible to nRF51822.

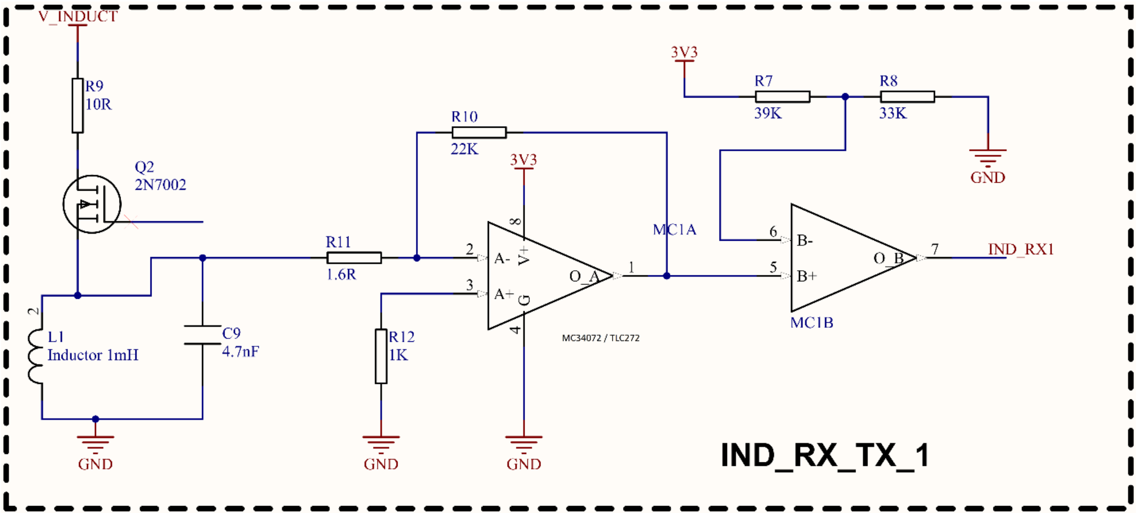


Figure 14: Circuit of designed Induction transceiver

For bit transmission modified encoding scheme from [22] was used. This consist of four source symbols, 'start character', 'zero character', 'one character' and 'end character' (see Figure 15). Each higher level symbol (e.g. numbers, ASCII characters) are encoded (converting to binary and adding odd parity) using those symbols and added to transmit buffer (FIFO). When a source symbol is read from transmit buffer, depending on the symbol it will transmit a single pulse whose width is a multiple of above RTC clock period (tested using T x 2 and timer interrupts of 1 tick = 4 ns = T/8) and wait for multiples of pulse width for the induced pulse to damp (tested using start_character_gap (1) = 4 T , one_character_gap (2) = 6T, zero_character_gap (3) = 8T, end_character_gap (5) = 10T). When a high-to-low transition occurs in receiver pin, another timer will count the ticks (1 tick = 4ns) between those transitions and determine the character within +/- 1T difference. If gap is less than 1T, it will be taken as a distortion of a single pulse. Then it will go to a sub state machine waiting for other symbols. From the time gaps transmitted source symbols (characters) are determined and shifted to find the integer value. If pulse gap is greater than 10T+T, it will determine as end of character (or timeout of waiting for further characters) and parity is checked for received combined symbol. Depending on error detection, either it will be discarded or added to higher level symbol buffer. For more details see "induct_radio.c".

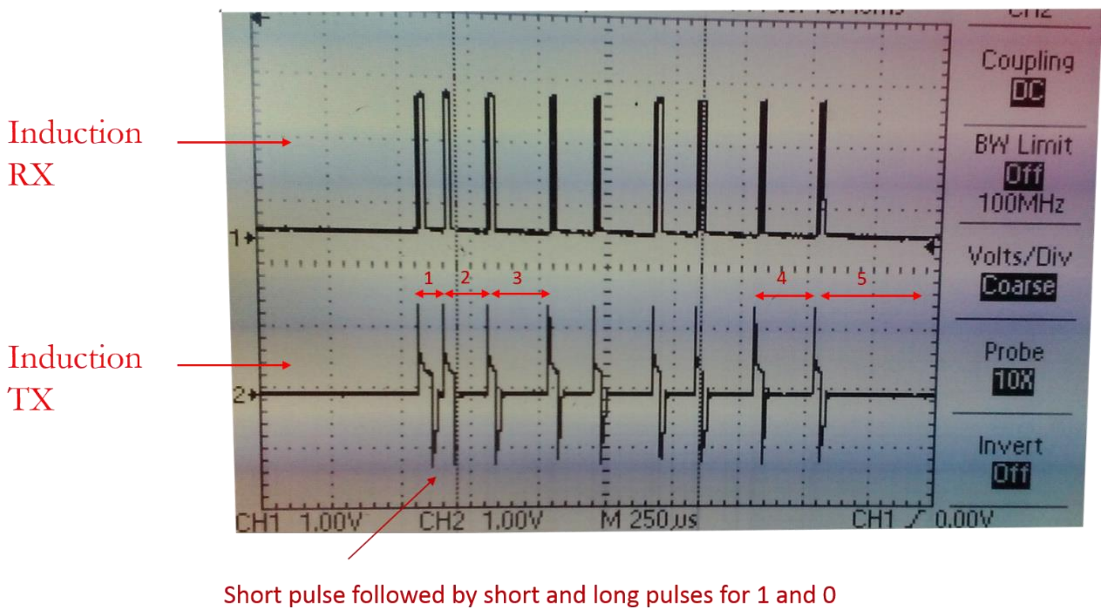
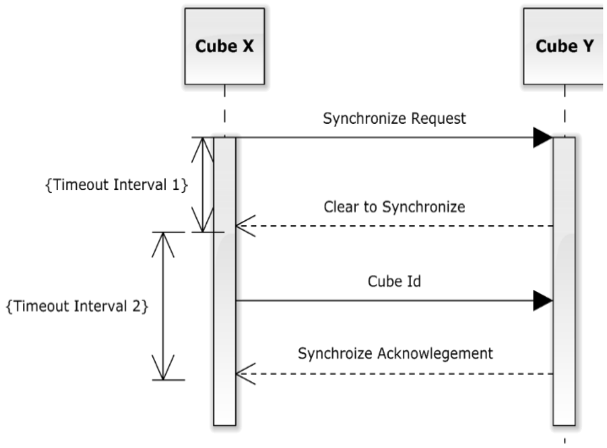


Figure 15: Encoded 42 (base 10) = [start, 1, 0, 1, 0, 1, 0, odd parity(0), end character]

4.2.5.3 High level state machine

For synchronizing of higher level symbols ‘three way hand shake protocol’ was used as shown in Figure 16.



State : function → next state

0: Init -> 1

RX data

1: get SYNC_REQ → 2
2: got SYNC_REQ, send SYNC_CLR → 3
3: start receiving data → 4 ,
if timeout → 2
4: send SYNC_ACK → 11

TX data

11: try to connect with others by sending SYNC_REQ → 12
if no need to transmit → 1
12: wait for SYNC_CLR and send my ID → 12
if timeout → 11
13: wait for SYNC_ACK, reset sending requests → 1
if timeout → 11

Figure 16: handshake protocol and state machine

4.2.5.4 Touch for neighbor detection

Due to the shape of SimKits (cube) there are limitations to detect neighbors directly (i.e. when cubes come closer they will automatically detect neighbor cubes). But for some applications this will be a hindrance for making connections between cubes.

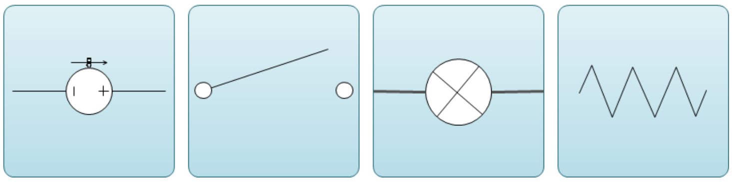


Figure 17: A simple circuit to light up a bulb

As shown in Figure 17, to connect the open (unconnected) end of current source to open (unconnected) end of resistor, user has to use either two ground connectors or make a connection between those two open ends. So touch is also used (one touching one end and the other) to make virtual connections between far away cubes.

4.2.6 Power consumption

Table 11 shows the power consumption of current SimKit implementation.

Table 11: Power consumption of Cube assuming all units are operating simultaneously

Component	**Typ. Current**	**Max Current**
Accelerometer	2 uA	47 uA
LCD + Backlight	25+30 mA = 50 mA	30+30 mA = 60 mA
Touch Sensor	29 uA	29 uA
BLE	22.5 mA	26.5 mA
MPU	.2 mA	0.5 mA
Total	**72.73 mA**	**87.08 mA**

4.3 SimKit Android Applications

4.3.1 Requirements

Following requirements should be satisfied to run the SimKit Android Application on mobile phone;

- OS :
    - Above Android 4.3
- Hardware support :
    - Open GL ES 2.0 or above
    - Bluetooth V4.0 or higher support (Bluetooth Low Energy support)

4.3.2 Operation

Overall operation of base Android application is illustrated in Figure 18.

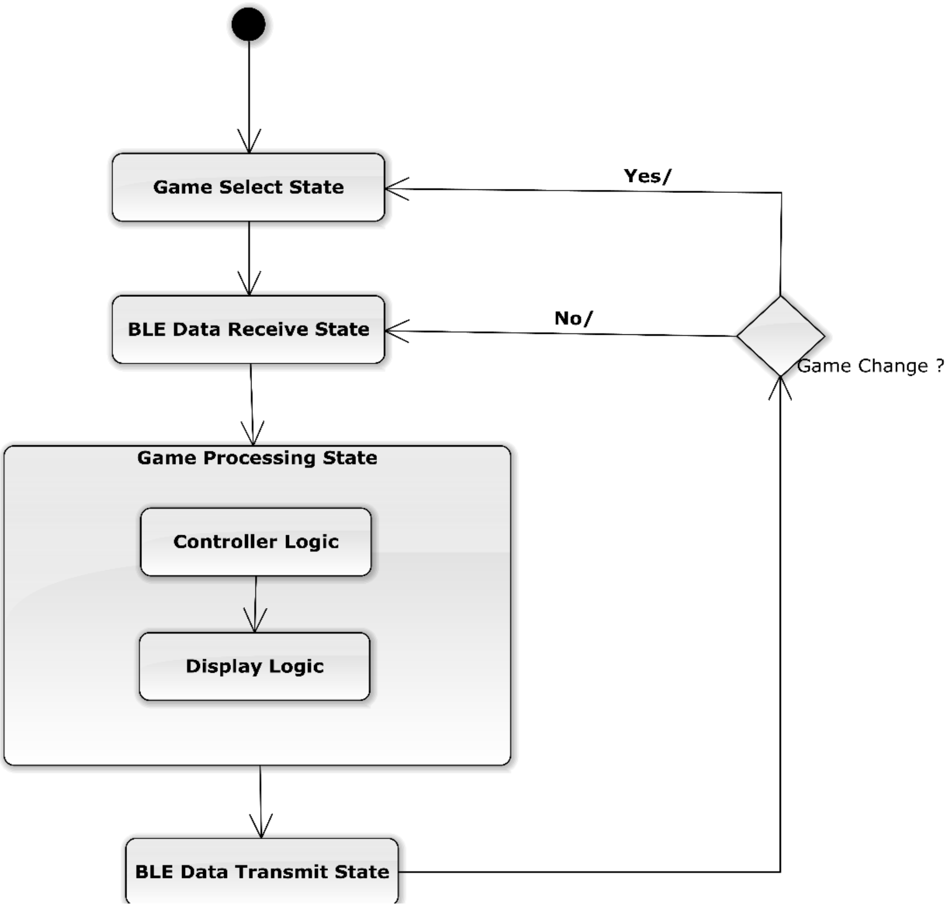


Figure 18: State machine on Mobile (Server)

Figure 19 describes the relationship between fundamental classes in SimKit application. Among them ‘BleService’ (control the RX and TX on android), ‘CubeLinkedTree’ (update and keep the relative locations of cubes) and ‘SimKitDecoder’ (decode the received commands through BLE and direct to corresponding controllers) plays the critical roles in running application.

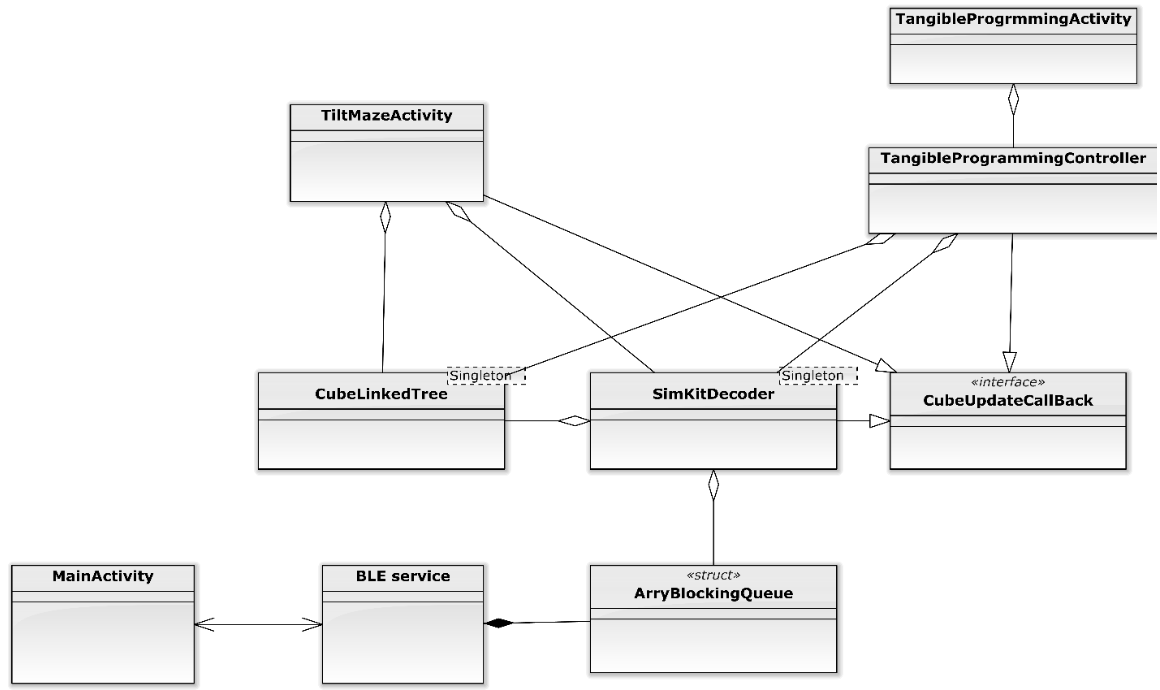


Figure 19: Class diagram of software platform

4.3.3 Base Application

The android base platform (Figure 20) is designed in order to connect all the clients or the SimKits to the server (Smart Phone) and then direct to any of the main applications (Circuit Simulation, Logic Simulation, Tangible Programming and Maze Game) according to the user preference. The base platform is designed in such a way where user can store all the available SimKits at the first time he/she is using them. At later times if the stored SimKits are available around the server, it'll automatically connect to the server. By means of this method, at any time we can add another SimKit cube to the network and if it is near the server it'll connect to the server.

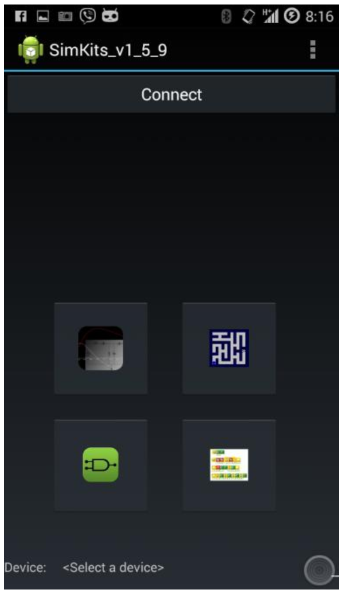


Figure 20: Android Base Platform

4.3.4 Tangible Maze

This is the application (Figure 21) where user can give the directions using SimKits Cubes to play the game. After getting the neighbor relations the user can give a sequence of steps to follow. For an instance first go Right, then go down, next turn Left. Likewise after preparing the correct sequence to the goal user can give the command to execute the sequence by pressing the Start Cube (The first cube which is connected to the server is taken as the Start cube and all the other cubes' positions are taken relative to the Start cube in the coordinate system.)

Figure 21: Maze Game

In order to finish this game with the lesser number of steps, the user must think from the end point where back tracking is coming into play. This game helps the user to improve his thinking and reasoning capabilities.

The operation of the Maze game is clearly indicated in Figure 23. We can give the sequence of directions using the cubes as following.

Figure 22: Maze Game Direction Sequence

After arranging the correct sequence of directions in order to approach to the goal, we must touch the Start cube in order to execute the sequence. If the correct sequence is given using the cubes we will be able to see the ball in the maze game rolling according to the sequence of directions given. If there is a wrong direction among the sequence the ball will stop the rolling.

Pros:

- Interactive game engine
- Help to learn reasoning and back tracking techniques
- Improve the logical thinking.

Cons:

- Neighboring should be done before starting the game by touching the cubes.
- Need many cubes if we need to give the direction sequence at once.

4.3.5 Tangible Programming

Tangible programming application helps to visualize how program execution happens step-by-step without the prior knowledge of computer programming. It consists of standard set of programming syntaxes including arithmetic operations, logical operations, flow control operations, constants and variables. When the user (target: O/L or A/L students) builds a program on cubes (Figure 23 (b)), it will update the relative locations on screen (Figure 23 (a)) assigning specific colors for specific operators. When the user touch on ‘start’ cube or corresponding image on mobile it start to compile. If there are missing parts in program, application will give an error message highlighting the error occurred cubes. After compiling successful, if user touch ‘start’ again it will display how execution happen step by step with constant time delay (user can configure). If runtime error happens (e.g. uninitialized variables, assign values to non-variable operators, etc.) the particular error caused blocks will be highlighted with corresponding error message.

(a)

(b)

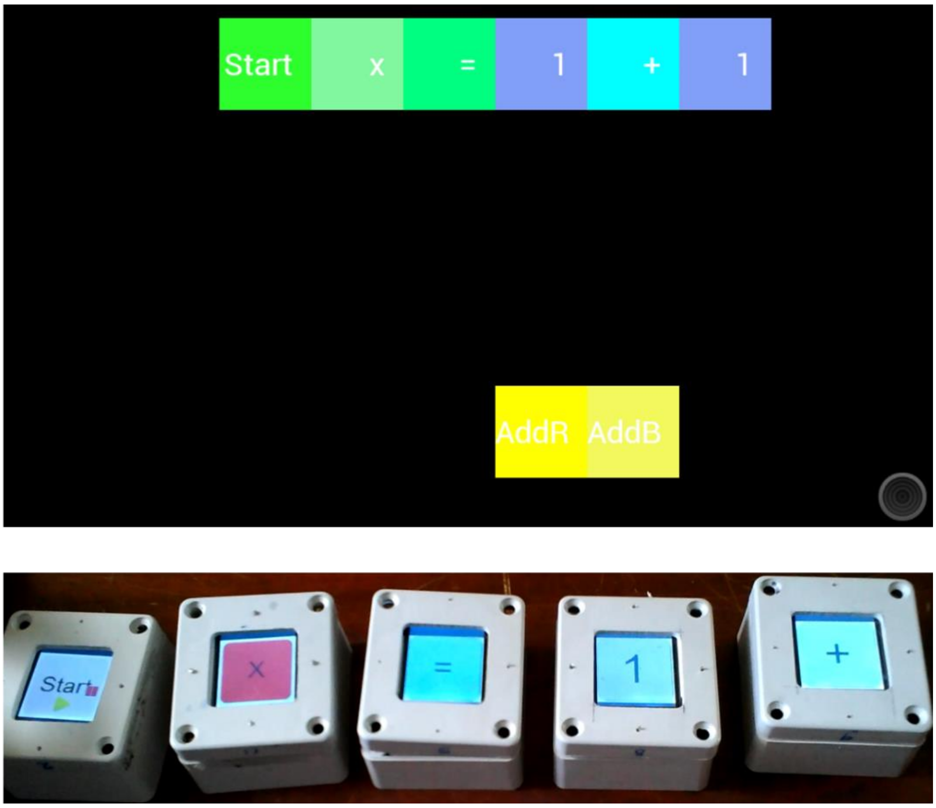


Figure 23: Tangible programming screen shot (a) mobile (b) cube set

Tangible programming consist of three steps

1) Update program
    a. update cube neighbors
    b. identify relative locations
    c. find the execution order relative to ‘start’
2) Compile program
    a. Start new thread (to display step by step)
    b. Separate flow control statements, assign statements from rest
    c. If ‘assign’ find variable and set the value of the expression from assign symbol (=) to end of line
    d. If ‘flow control’ (if, end_if, while, end_while), find the condition, total iterations from the end of condition to end flow control block
    e. Add them to stack for processing
    f. If errors occur display them
3) Execute program
    a. Start new thread (to display step by step)

b. Select control statements (assign and flow control) one after other in execution order
c. Execute each statement
d. If errors occur display them and stop

Pros:

- Display error location with reason
- Execute each step-by-step
- Can configure step time
- Can program moderately complex program without physical cubes
- Real-time updating, compiling and execution
- Support variables, constants and flow control operators

Cons:

- Text based programming language
- Cannot change the lastly updated component to its previous state

4.3.5.1 Technical Requirements

- For display graphics OpenGL ES 2.0 ([23]) is used, which eliminate the need to worry about the model, view point and projections of components
- For expression execution (i.e. evaluate condition true or false, do arithmetic/logical/relational operation, etc.) open source 'JEval' library ([24]) from Source Forge is used.

4.3.6 Circuit Simulation

The following android application (Figure 24) is designed to draw analog circuits where each component can be assigned with corresponding values. After assigning the neighbor relations to the SimKits cubes by touching the neighbor cubes the corresponding cubes will be drawn on the Circuit Simulation application. Also we can change the component using the android application or using the SimKits by flipping the cube. The new state will be updated both in the application the cube display.

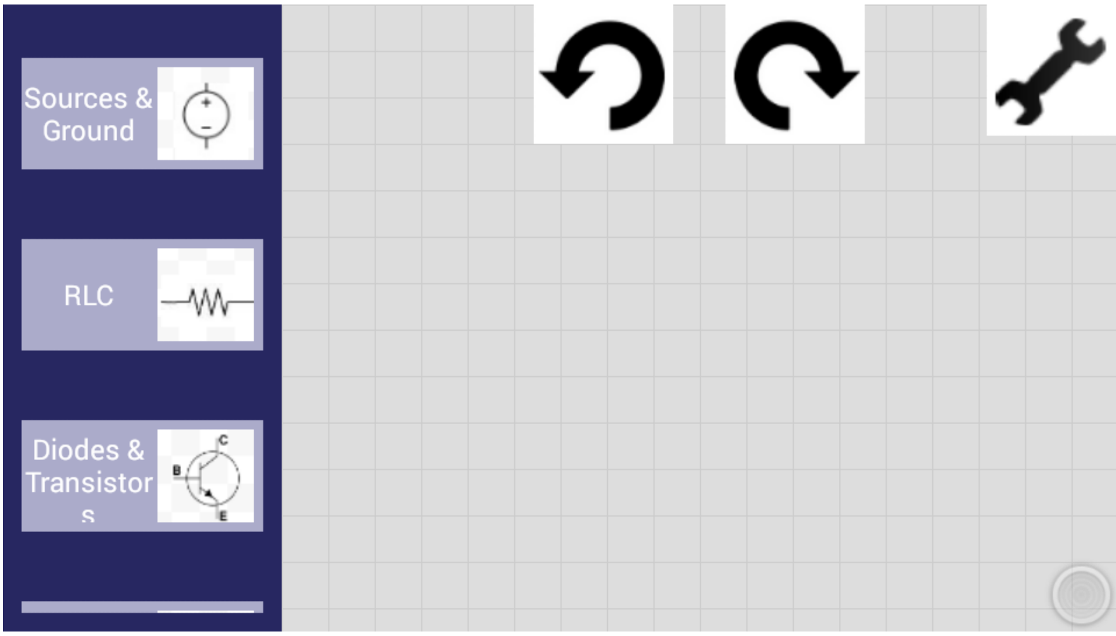


Figure 24: Circuit Simulation Application

This application is capable of saving the designed circuits. As a future improvement we can use the android application to show how the electrons are going through the circuit and various kinds of graphs to analyze the circuits created using SimKits cubes. Also we can use this to prove the concepts of electricity such as Ohm's law. Here is a simple example (Figure 25) of how the SimKits cubes are interacted with the Simulation application.

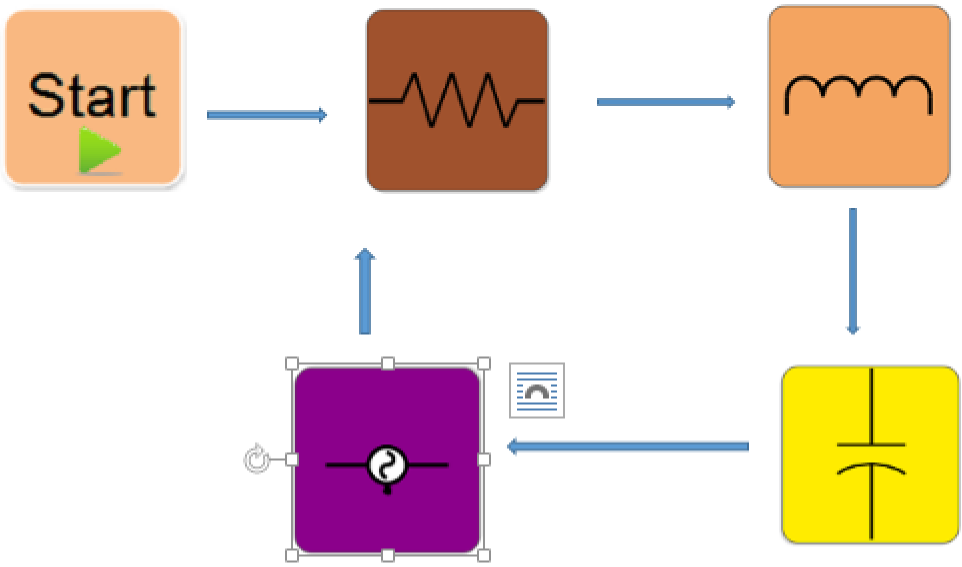


Figure 25: SimKits in Circuit Simulation Mode

4.3.6.1 Operation

Here are the steps to follow in Circuit Simulation.

- First of all we must connect all the cubes to the server or the smart phone.
- After connecting we must go the circuit simulation application.
- Then all the SimKits' screen will be updated by the default inductor image except the Start cube which is connected first to the server.
- Next we can connect the neighbors by touching the relevant touch sensors (There are four touch sensors around the TFT display).
- When we connect the neighbors according to the above figure, android application in the smart phone should be updated as following.
- Whenever we change the component by flipping the SimKit cube, corresponding component should be changed within the android application.

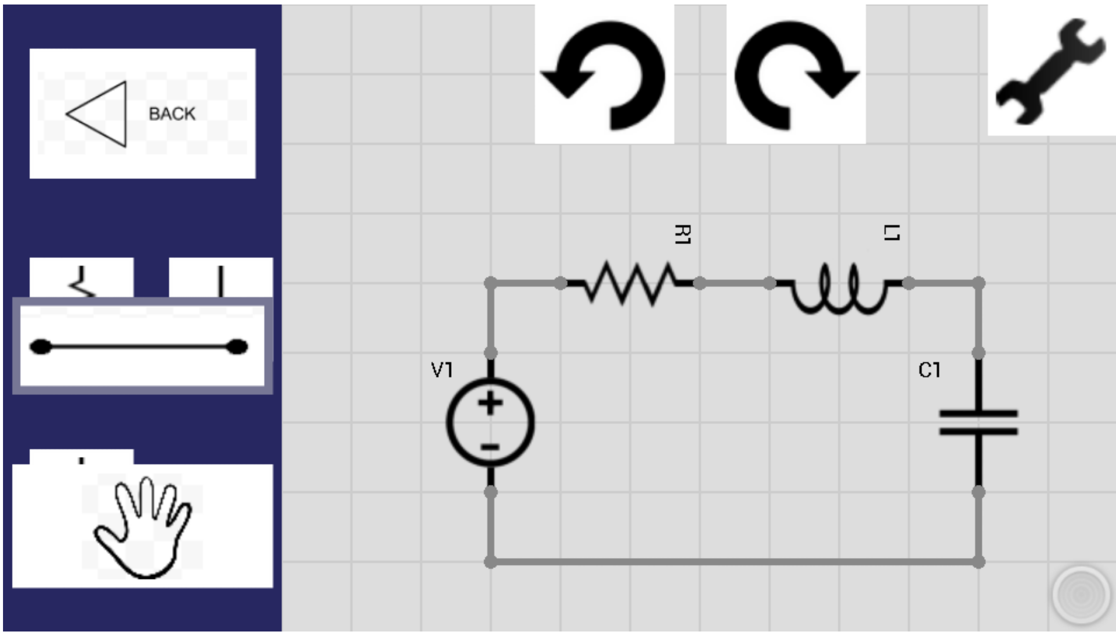


Figure 26: Updated Android application when SimKits are connected

Not only that, but also whenever we change the component using the android application (Figure 26), corresponding SimKit cube should be updated with the relevant component. Components values can be changed through the android application. For an instance we can give the resistance value for the selected resistor.

4.3.7 Logic Simulation

The following android application (Figure 27) is designed to draw digital circuits using basic gates such as AND gate, OR gate and NOT gate. This application is same as the circuit simulation application where connections can be made based on the neighbor relations given by the SimKits cubes.

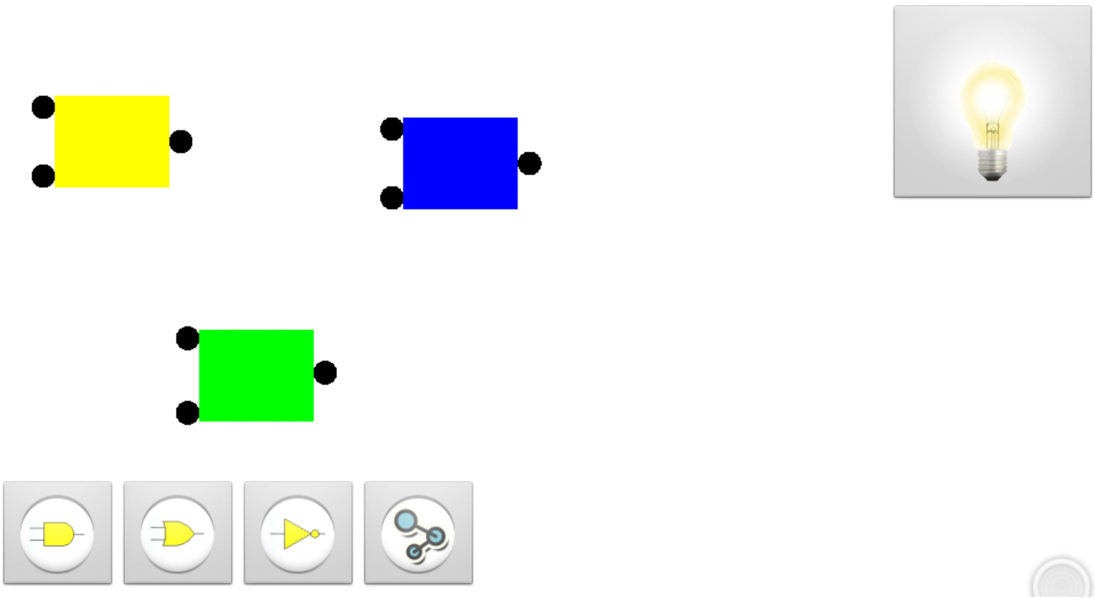


Figure 27: Logic Simulation Application

We can change the components (Figure 27, Figure 29) in the cube by just flipping it. Next component will be displayed in the TFT screen. This application was designed in order to give the basic understanding of Logical circuits to the children. While having the fun, the learning makes more effective. Following figures indicates some of the components which are stored in the SimKit cubes' memory.

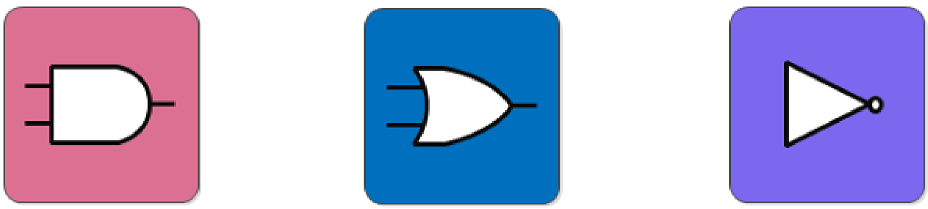


Figure 28: AND, OR, NOT gates respectively

Figure 29: Input and Output

4.3.7.1 Operation

- First connect all the cubes to the server
- Then go to the Logic simulation application which is indicated by a Logic Gate button
- All the TFT screens will be updated with a default AND gate except the Start cube
- Next connect the neighbors by touching the relevant touch sensors
- Change the components as users preference
- The corresponding circuit will be drawn in the android application

4.4 Experiments and Results

4.4.1 Test groups

We are going to select three sets of user groups to evaluate SimKits. Those user groups can be listed as;

- Primary school children
- O/L & A/L students
- University students

These SimKits should be tested with the above mentioned groups in order to get a clear picture of what modifications should be added to the system.

4.4.2 Results

Still these SimKits cubes were tested with the members of the group and some other university students. According to the responses, the response time of the system should be lower in order to improve the interactivity and performance of the project.

4.5 Issues Encountered

- Interaction finalizing
- Game selecting
- Handling BLE communication between multiple nRF51822 clients and the server
- BLE communication delays and data loss
- Neighbor Detection using inductors while having immense noise
- Hardware SPI communication between nRF51822 and TFT LCD display
- Image Compression in order to utilize the memory efficiently.

5 DISCUSSION AND FUTURE WORK

SimKits is a research and product development project. It was designed as an interactive educational platform and has the functionality of tangible programming and circuit implementation.

There are several reasons to believe that this project is an opening for the learning things in an interactive way. Here we summarize the project using its main objectives namely interactivity & playfulness.

- Interactivity – SimKits are TUIs which interact with day-to-day objects (e.g. smart phones, computer, tablets etc).
- Playfulness – Simplicity and tangibility makes SimKits more attractive for children and help to overcome the difficulties in learning programming and implementing circuits.

Due to the multiple connections and packet delays in Bluetooth, current prototype has a considerable responsive time which limit the speed of performance. But this may have a good market potential as a result of its novel and interactive features.

5.1 Future Improvements

Bluetooth multiple connections caused unexpected delays and packet losses during implementation of SimKits. There were no reference related to performance evaluation of BLE multiple connections and the stability of multiple connections. So we should study the performance of multiple connections and find ways to improve response time (e.g. separate service for each sensor…) and eliminate unnecessary delays and losses. Adapting complex data formats for induction communication may eliminate client-server communication delays.

Also we are hoping to improve the android applications in order to give a better output. For an instance we can improve the circuit simulation application to simulate the electrons flow when a voltage source is given which will help children to feel the concepts of electronics. Not only that, but also we should add real time graphing which corresponding to the circuits drawn using SimKits.

5.2 Other Applications

Since SimKit is an interactive tangible user interface, there are many application spaces where it can be used.

- Interaction with phone
  - Cancel a call
  - Show time on cube
  - Playing games
- Simulating A/L physics theories such as acceleration/motion and logic circuits
- Fluid display
- Act as a wireless mouse/ remote controller

REFERENCES

[1] J. M. Wing, "Computational Thinking," *Communications of the ACM - Self managed systems CACM*, pp. Pages 33-35, 2006.

[2] A. P. Ambrosio, L. S. Almeida, J. Macedo and A. H. R. Franco , "Exploring core cognitive skills of Computational Thinking," in *PPIG proceedings*, 2014.

[3] . S. Horn and . J. K. Jacob, "Tangible programming in the classroom with tern," in *CHI EA '07 CHI '07 Extended Abstracts on Human Factors in Computing Systems*, 2007.

[4] D. Merrill, J. Kalanithi and P. Maes, "Siftables: towards sensor network user interfaces," in *TEI '07 Proceedings of the 1st international conference on Tangible and embedded interaction*, 2007.

[5] G. W. Fitzmaurice, H. Ishii and W. A. S. Buxton, "Bricks: laying the foundations for graspable user interfaces," in *CHI '95 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1995.

[6] P. Marshall, "Do tangible interfaces enhance learning?," in *TEI '07 Proceedings of the 1st international conference on Tangible and embedded interaction*, 2007.

[7] M. S. Horn, E. T. Solovey, R. J. Crouser and R. J. Jacob, "Comparing the use of tangible and graphical programming languages for informal science education," in *CHI '09 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2009.

[8] C. Kelleher and R. Pausch, "Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers," *Journal ACM Computing Surveys (CSUR),* vol. Volume 37, no. Issue 2, pp. Pages 83-137 , 2005.

[9] . Brennan, . M. Hernández and . Resnick, "Scratch: creating and sharing interactive media," in *CSCL'09 Proceedings of the 9th international conference on Computer supported collaborative learning*, 2009.

[10] . S. Hood and . J. Hood, "Teaching programming and language concepts using LEGOs," in *ITiCSE '05 Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education*, 2005.

[11] D. Wang, C. Zhang and H. Wang, "T-Maze: a tangible programming tool for children," in *IDC '11 Proceedings of the 10th International Conference on Interaction Design and Children*, 2011.

[12] M. S. Horn, R. J. Crouser and . U. Bers, "Tangible interaction and learning: the case for a hybrid approach," *Journal Personal and Ubiquitous Computing,* vol. Volume 16, no. Issue 4, pp. Pages 379-389 , 2012.

[13] . Schiettecatte and . Vanderdonckt, "AudioCubes: a distributed cube tangible interface based on interaction range for sound design," in *TEI '08 Proceedings of the 2nd international conference on Tangible and embedded interaction*, 2008.

[14] "Sifteo," Sifteo Inc, [Online]. Available: https://www.sifteo.com/cubes. [Accessed 07 2014].

[15] L. Terrenghi, M. Kranz, P. Holleis and A. Schmidt, "A cube to learn: a tangible user interface for the design of a learning appliance," *Personal and Ubiquitous Computing,* vol. Volume 10 , no. Issue 2-3, pp. Pages 153-158 , 2006.

[16] L. K. F. J. T. W. S. Wooi Boon Goh, "The i-Cube - Design Considerations for Block-based Digital Manipulatives and their Applictions," in *Proceeding - DIS '12 Proceedings of the Designing Interactive Systems Conference*, 2012.

[17] Y. I. ,. M. A. ,. Y. K. ,. F. K. Ryoichi Watanabe, "The soul of ActiveCube: implementing a flexible, multimodal, three-dimensional spatial tangible interface," *Computers in Entertainment (CIE)* , vol. 2, no. 4, p. 15, 2004.

[18] . T. Kirby, . Ashley-Rollman and . C. Goldstein, "Blinky blocks: a physical ensemble programming platform," in *CHI EA '11 CHI '11 Extended Abstracts on Human Factors in Computing Systems,* 2011.

[19] K. C. Peters, "TOUCHSPICE: PHYSICAL-VIRTUAL CIRCUIT EMULATOR," 2012.

[20] "Infrared Data Association," [Online]. Available: http://en.wikipedia.org/wiki/Infrared_Data_Association. [Accessed 09 2014].

[21] N. Semiconductors, "PN512," [Online]. Available: http://www.nxp.com/products/identification_and_security/nfc_and_reader_ics/nfc_front_end_solutions/series/PN512.html. [Accessed 09 2014].

[22] J. Kalanithi and D. Merrill, "Data communication and object localization using inductive coupling". USA, Patent US8611817 B2, 17 12 2013.

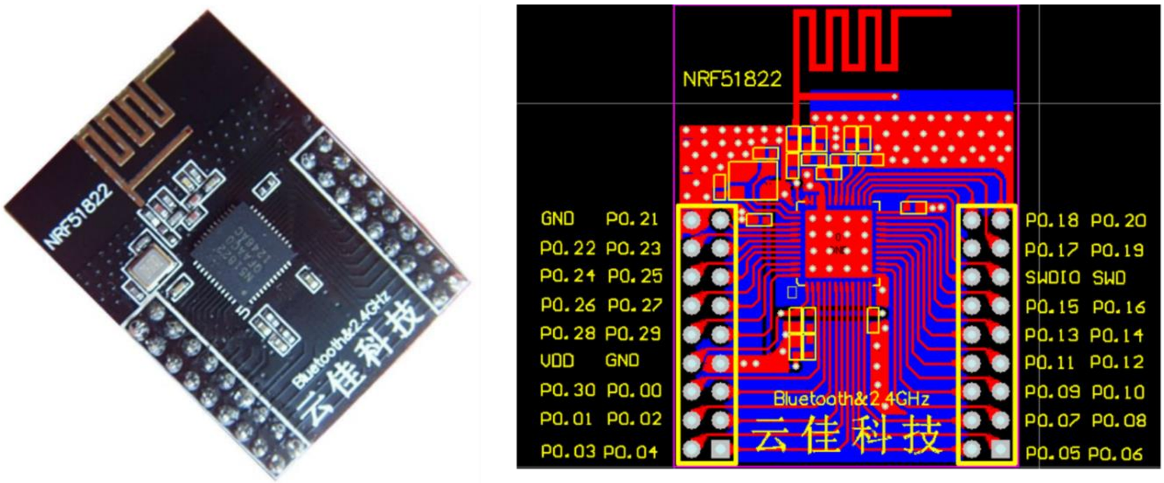
[23] K. Group, "OpenGL ES (The Standard for Embedded Accelerated 3D Graphics)," [Online]. Available: https://www.khronos.org/opengles/. [Accessed 12 2014].

[24] R. Breidecker, "JEval," [Online]. Available: http://jeval.sourceforge.net/. [Accessed 09 2014].

[25] [Online]. Available: http://en.wikipedia.org/wiki/Bluetooth_low_energy.

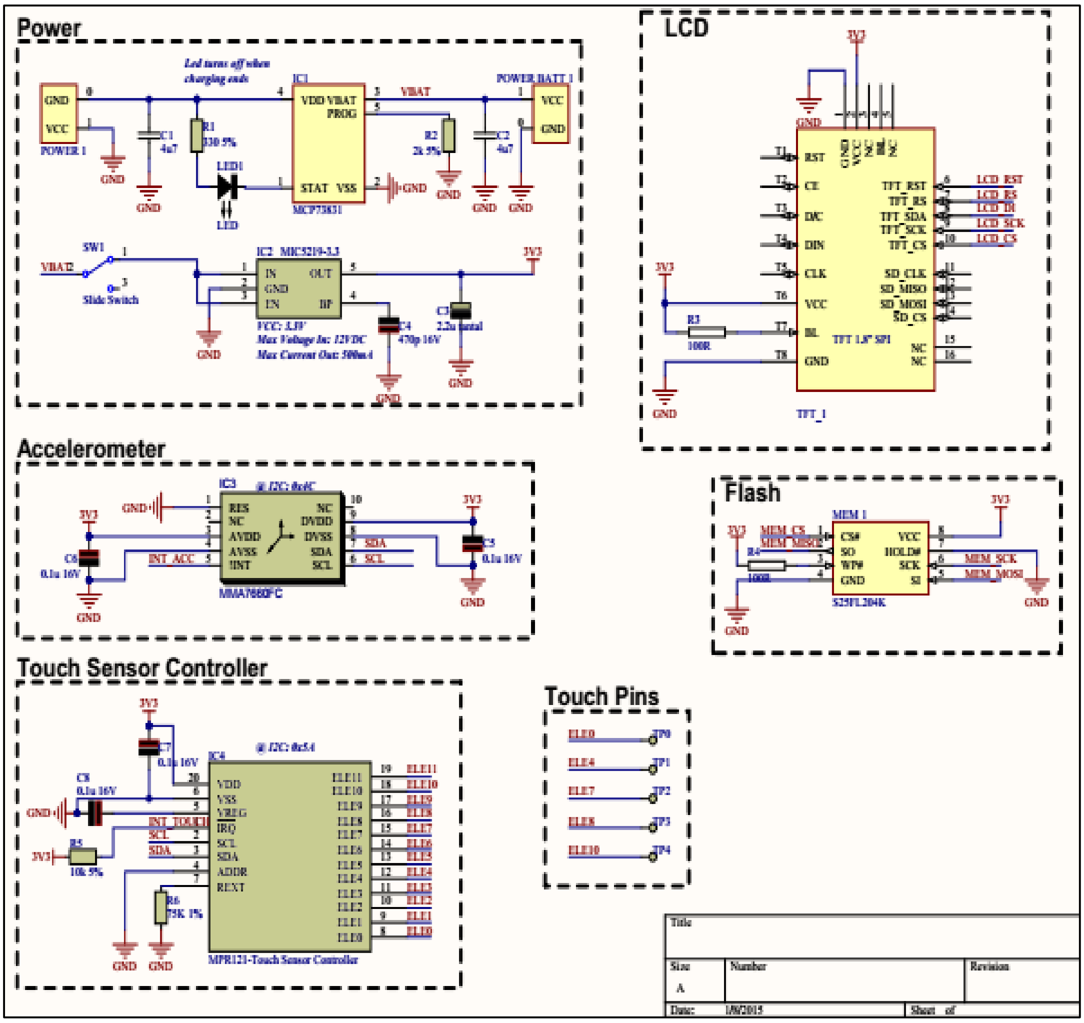
[26] [Online]. Available: http://www.medicalelectronicsdesign.com/article/bluetooth-low-energy-vs-classic-bluetooth-choose-best-wireless-technology-your-application.

APPENDIX A: NRF51822 MODULE

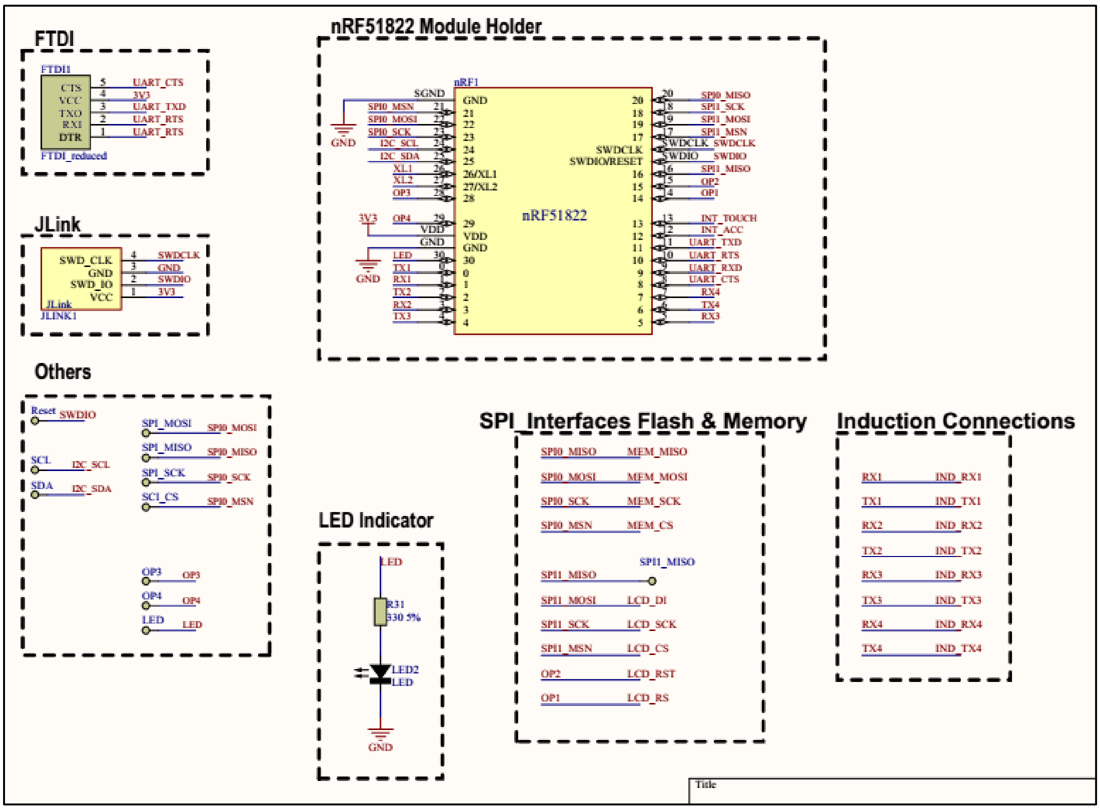


nRF51822 module imported from eBay

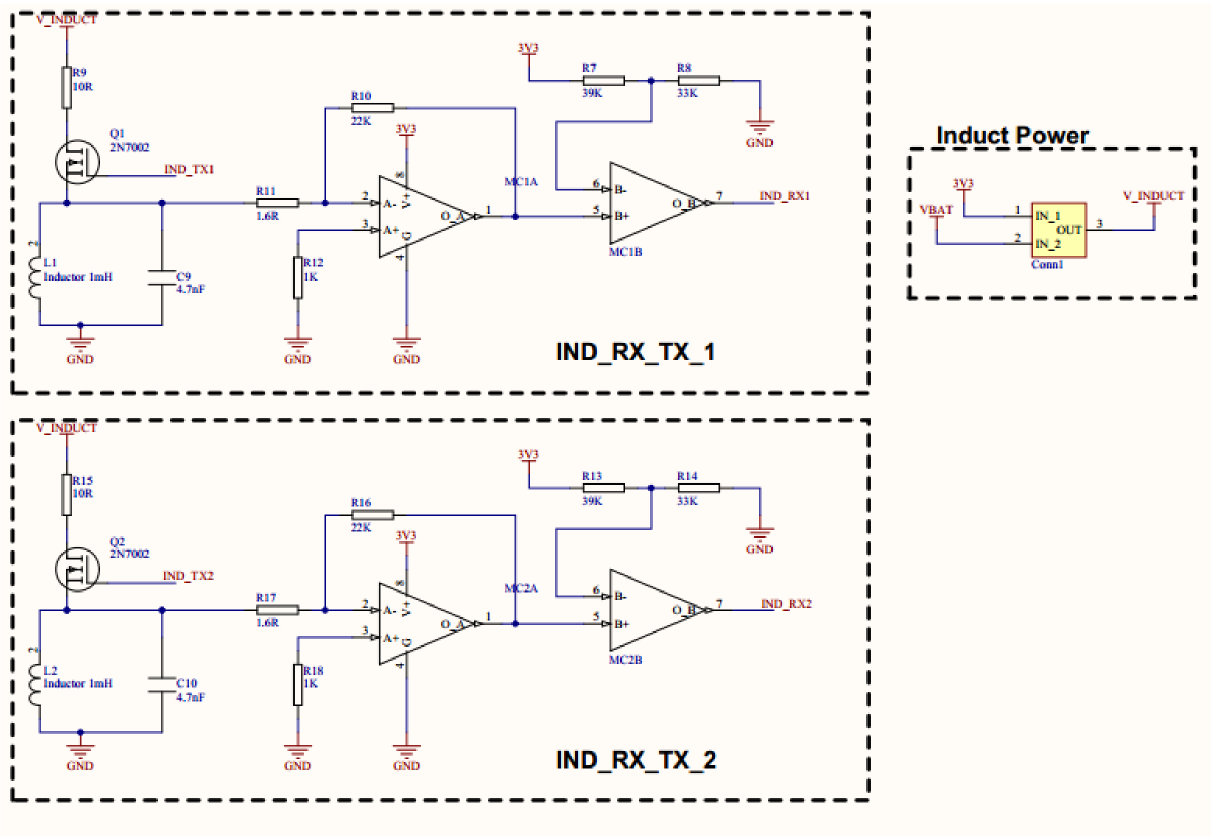
APPENDIX B : CIRCUIT DIAGRAM



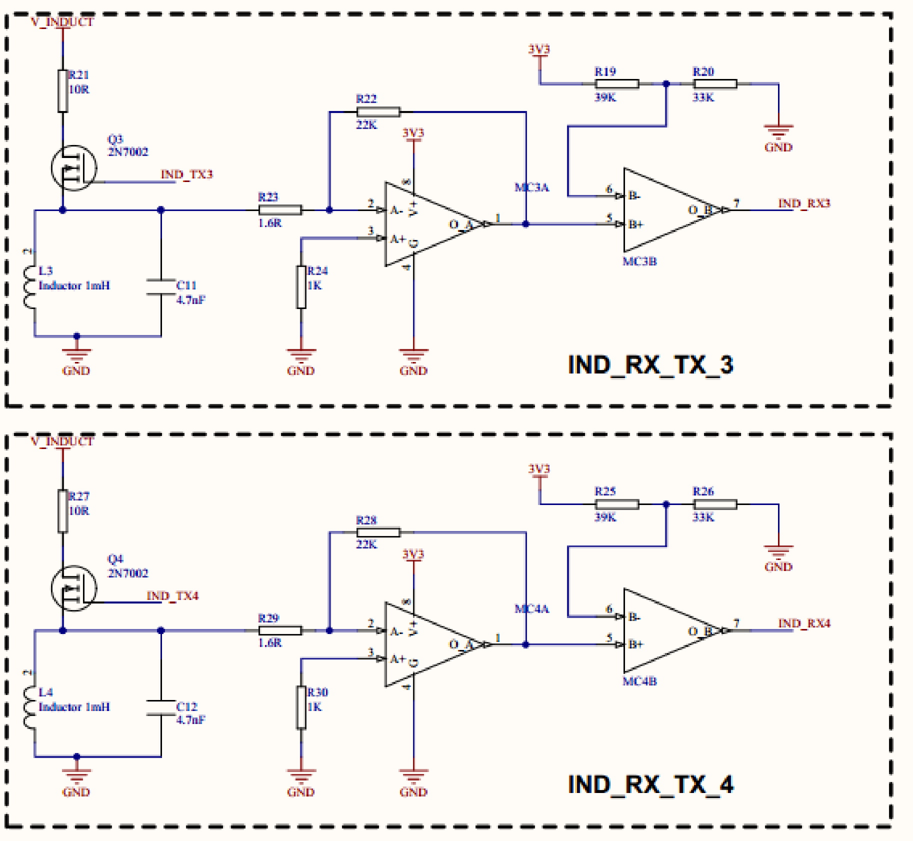
Peripheral circuit schematic



Central circuit schematic

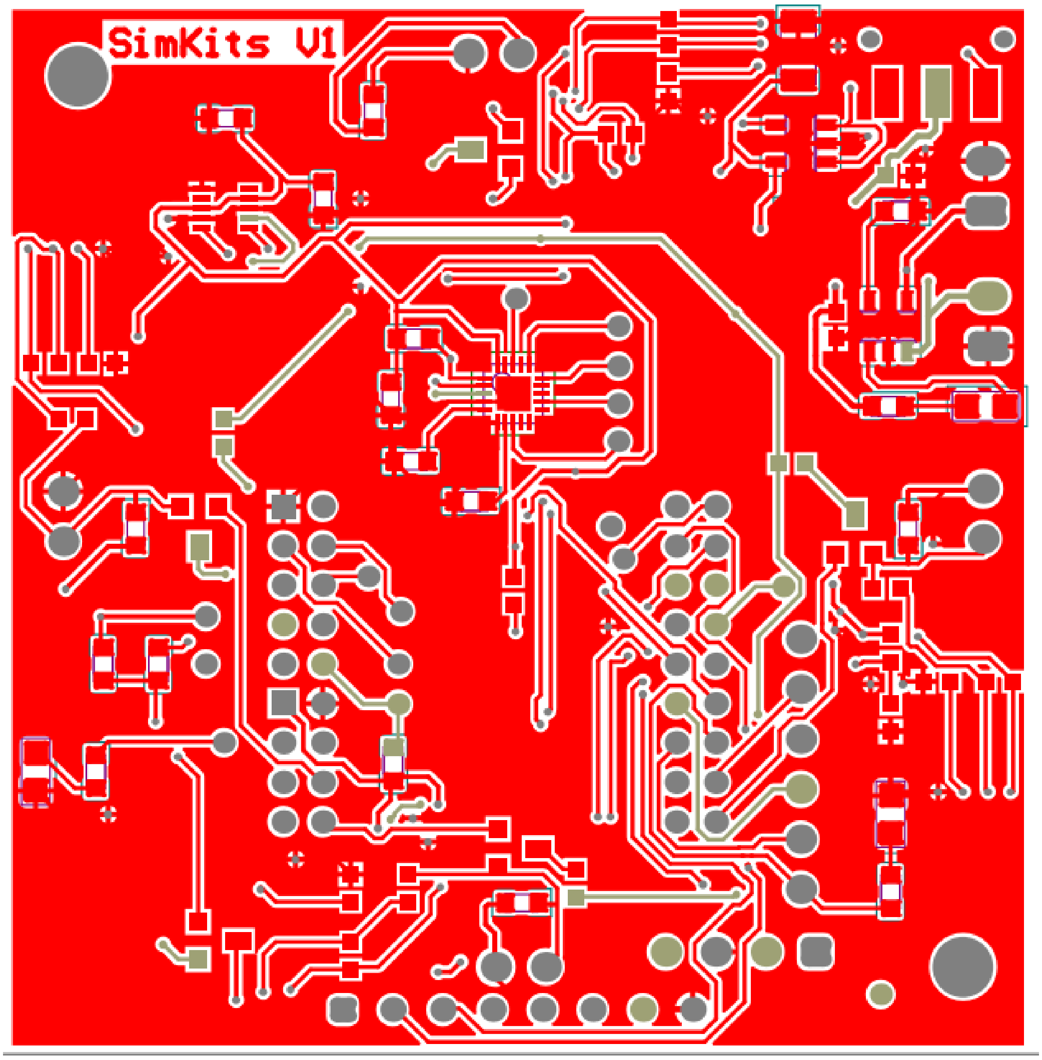


Induction transceiver schematic1

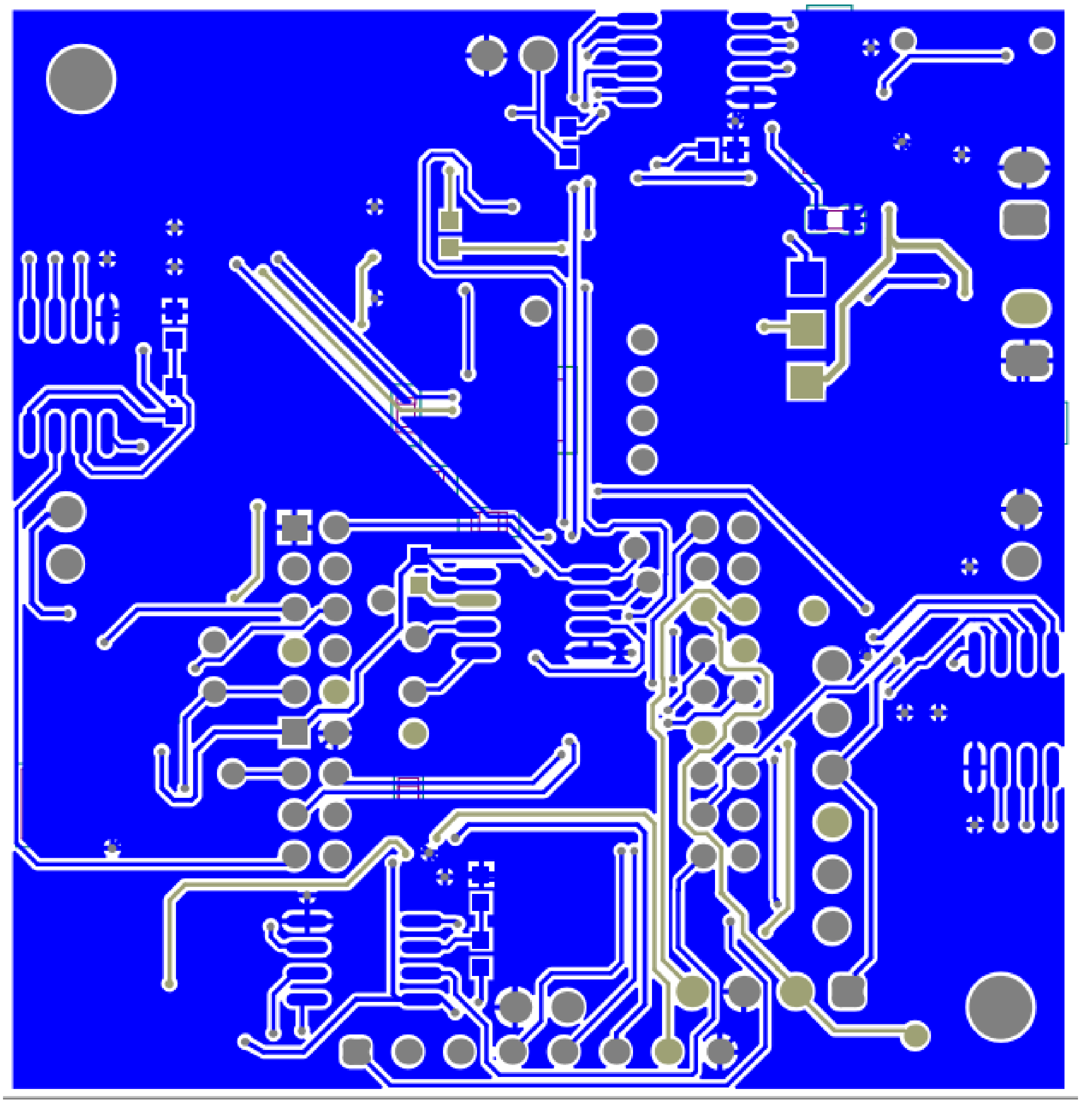


Induction transceiver schematic2

APPENDIX C: PCB LAYOUT



Top copper layer of PCB



Bottom Copper Layer of PCB