DESIGN & DEVELOPMENT OF A RASPBERRY PI DAUGHTER BOARD TO SUPPORT BLUETOOTH COMMUNICATION USING UART

UNIVERSITY OF KHARTOUM

By

Ali Kamal Ali Ahmed

INDEX NO. 124065

Supervisor

M.Sc. Eng. Mohannad Ahmed Mohammed Alhassan

A REPORT SUBMITTED TO

University of Khartoum

In partial fulfillment of the requirements for the degree of

B.Sc. (HONS) Electrical and Electronic Engineering

(ELECTRONIC ENGINEERING)

Faculty of Engineering

Department of Electrical and Electronic Engineering

October 2017
I declare this report entitled “Design & Development of a Raspberry Pi Daughter Board to support Bluetooth communication using UART” is my own work except as cited in references. The report has been not accepted for any degree and it is not being submitted currently in candidature for any degree or other reward.

Signature: _______________________

Name: _________________________

Date: _________________________
DEDICATION

• To my Mother, for her tireless support, for always pushing me to my full potential, never-ending love and devotion.
• To my father for his support, love and encouragement throughout my university carrier.
• To my beloved brothers and sisters.
ACKNOWLEDGEMENT

- M.Sc.Eng. Mohannad Ahmed Mohammed Alhassan, our mentor and guiding light; whose concern is always that we learn from anything that we do.
- M.Sc.Eng. Yusra Osman Eltahir Babiker, our liaison at Nile Research Center, who didn’t spare anything in trying to help us envision and realize our design.
- Eng. Mohammed Salah Alwali, our software mentor, none of the design could be accomplished without your teachings.
ABSTRACT

Since the late 1970's, personal computers have been a rapidly growing part of society. Starting from small kits to modern supercomputers, PC’s have become an essential part of our lives. However, the number of people who understand the working of a computer is incredibly small compared to the number of users. In 2012, a small company based in the UK called the Raspberry Pi Foundation launched two models of a small and cheap computer with different specifications, with the intention of promoting and teaching basic computer science in schools and colleges. Due to their low costs (between $25 - $35) they have become extremely popular with hobbyists and small project leaders.

The Raspberry Pi provides many interfaces including: General purpose IO pins, USB, Ethernet, UART serial connection and audio/video outputs. However, it does not provide any forms of wireless communication. Due to the rapid increase in Smartphone users, Laptops and other mobile devices, wireless forms of data transfer is rapidly becoming the standard.

For this reason, we decided to Design and Develop a daughter board for the Raspberry Pi that will allow it to communicate with other mobile devices using the standard protocol, Bluetooth.
منذ أواخر السبعينات، كانت الحواسيب الشخصية جزءًا لا يتجزأ من نمو المجتمع. بدءًا من مجموعات صغيرة إلى الحواسيب الفائقة الحديثة، أصبحت أجهزة الكمبيوتر جزءًا أساسًا من حياتنا. ومع ذلك، فإن عدد الأشخاص الذين يفهمون عمل جهاز الكمبيوتر صغير بشكل لا يصدق مقارنة بعدد المستخدمين. في عام 2012، أطلقت شركة صغيرة مقرها في المملكة المتحدة تدعى منظمة الراسبري باي نموذجين من جهاز كمبيوتر صغير ورخيص مع مواصفات مختلفة، بهدف تحسين وتعليم علوم الحاسب الأساسية في المدارس والكليات. نسبة إلى انخفاض تكاليفها (بين 25 $ - 35 $) أصبحت أجهزة الكمبيوتر شعبية للغاية مع الهواة وقادة المشاريع الصغيرة.

منظمة الراسبري باي توفر العديد من واجهات التواصل من ضمنها: الغرض العام لدبابيس الادخال/الإخراج، يو إس بي، إيثرنت، يو-ارت الاتصال التسلسلي ومخرجات الصوت / الفيديو.

ومع ذلك، فإنه لا يوفر أي شكل من أشكال الاتصالات اللاسلكية. ونظراً للزيادة السريعة في مستخدمي الهواتف الذكية وأجهزة الكمبيوتر المحمولة وغيرها من الأجهزة النقالة؛ أصبحت الأشكال اللاسلكية لنقل البيانات ذات السرعة العالية المعيار الأساسي.

لذا السبب، قررنا تصميم وتطوير لوحة إلكترونية صغيرة لتلك التي صممتها شركة الراسبري باي التي من شأنها أن تسمح لها بالتعامل مع الأجهزة النقالة الأخرى باستخدام تقنية البروتوكول (البلوتوث).
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>...........................................................................................................</td>
<td></td>
</tr>
<tr>
<td>CHAPTER ONE</td>
<td>INTRODUCTION ...............................................................................</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Brief History ..................................................................................</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>Rise of Embedded Computers ................................................................</td>
<td>2</td>
</tr>
<tr>
<td>1.3</td>
<td>The Internet of Things .....................................................................</td>
<td>3</td>
</tr>
<tr>
<td>1.4</td>
<td>Solving the Connectivity Problem ..................................................</td>
<td>3</td>
</tr>
<tr>
<td>1.5</td>
<td>Objectives .......................................................................................</td>
<td>4</td>
</tr>
<tr>
<td>1.6</td>
<td>Thesis Layout ...................................................................................</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>...........................................................................................................</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>...........................................................................................................</td>
<td>5</td>
</tr>
<tr>
<td>CHAPTER TWO</td>
<td>TECHNOLOGY OVERVIEW ..................................................................</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>Wireless Communication ....................................................................</td>
<td>5</td>
</tr>
<tr>
<td>2.1.1</td>
<td>Wireless Standards ..........................................................................</td>
<td>5</td>
</tr>
<tr>
<td>2.2</td>
<td>Bluetooth Standard ..........................................................................</td>
<td>7</td>
</tr>
<tr>
<td>2.2.1</td>
<td>Bluetooth Frequency Bands ................................................................</td>
<td>10</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Common Bluetooth Versions and Classes .........................................</td>
<td>10</td>
</tr>
<tr>
<td>2.3</td>
<td>RF Design .........................................................................................</td>
<td>12</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Antennas ............................................................................................</td>
<td>13</td>
</tr>
<tr>
<td>2.4</td>
<td>Computer Aided Design Programs .....................................................</td>
<td>17</td>
</tr>
<tr>
<td>2.4.1</td>
<td>High frequency simulation tools ......................................................</td>
<td>17</td>
</tr>
<tr>
<td>2.4.2</td>
<td>PCB Design Tools .............................................................................</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>...........................................................................................................</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>...........................................................................................................</td>
<td>19</td>
</tr>
<tr>
<td>CHAPTER THREE</td>
<td>DESIGN AND IMPLEMENTATION .........................................................</td>
<td>19</td>
</tr>
</tbody>
</table>
3.1 Design Vision

1. Picking Components

2. PCB Design

3. Design Fabrication

4. Debugging/Verification

3.1.1 The Full-Design route

3.1.2 Ready-made Modules

3.2 Design Implementation

3.2.1 Circuit Layout

3.2.2 Software Setup

3.3 Testing Methodology:

3.3.1 Transfer rate test

3.3.2 File size

3.3.3 Data corruption

3.3.4 Range test

3.3.5 Power consumption test

3.4 Example demonstration

3.4.1 Case 1: Simple File transfer

3.4.2 Case 2: SSH Protocol Control

CHAPTER FOUR

RESULTS & DISCUSSION

4.1 Results

4.1.1 Fully designed module tests

4.1.2 Transfer rate
CHAPTER FIVE

CONCLUSIONS

5.1 Problem statement and solution .................................................. 48
5.2 Problem implementation ............................................................. 48
5.3 Solution Result ............................................................................. 48
5.4 Project Outcome ........................................................................... 48
5.5 Project Limitations ....................................................................... 49
5.6 Personal Gain and Future Work .................................................... 49

REFERENCES ...................................................................................... 50
LIST OF FIGURES

Figure 1: Processing Power growth over time.......................................................... 1
Figure 2: A Raspberry Pi 1 board ............................................................................. 2
Figure 3: Bluetooth Network Topology .................................................................... 7
Figure 4: Serial Cable VS Bluetooth Raspberry Pi interface .................................... 9
Figure 5: RF Transducer Block Diagram ................................................................. 12
Figure 6: System Block Diagram .......................................................................... 21
Figure 7: System Flowchart .................................................................................. 21
Figure 8: Recommended Circuit Diagram for the CSR1012 .................................... 23
Figure 9: Schematic library, Schematic symbol for CSR1012 IC ............................ 24
Figure 10: System Schematic ............................................................................... 25
Figure 11: PCB Library, footprint for CSR1012 IC ................................................. 26
Figure 12: Final PCB Design ............................................................................... 27
Figure 13: Patch micro-strip being simulated in microwave office ....................... 28
Figure 14: Meander Antenna Design .................................................................... 28
Figure 15: a) CNC Drilling Machine b) Traced- PCB ............................................. 29
Figure 16: (A) Final Product (B) CSR 10X Family USB Programmer .................... 30
Figure 17: HC-05 Pinout .................................................................................... 32
Figure 18: Raspberry Pi Pinouts; (a) Model 1 A+ (b) Model 2 B+ ............................. 33
Figure 19: Raspberry Pi to HC-05 Interfacing; (a) Model 2 B+ (b) Model 1 A+ ........ 33
Figure 20: Final Setup ...................................................................................... 34
Figure 21: Raspi-Config menu .......................................................................... 35
Figure 22: BlueTerm-Raspberry Pi SSH interface example .................................... 40
Figure 23: 1-1.3GHz S11 VS Frequency using (a) Simulation (b) Network analyzer test .... 41
Figure 24: 2-3GHz S11 VS Frequency Simulation .................................................. 42
Figure 25: Antenna TL impedance test .................................................................. 42
Figure 26: Default baud rate ............................................................................... 43
Figure 27: Optimized baud rate .......................................................................... 44
LIST OF TABLES

Table 1: Wireless communication standards comparison ................................................................. 6
Table 2: Bluetooth Classifications ..................................................................................................... 11
Table 3: Antenna design types comparison ......................................................................................... 16
Table 4: BT modules comparison ...................................................................................................... 22
Table 5: Ready-made Bluetooth modules comparison ........................................................................ 31
Table 6: A list of other useful AT Commands .................................................................................... 37
Table 7: Transmission Rate test ........................................................................................................ 43
Table 8: Pairing Range test ................................................................................................................ 44
Table 9: Hash test ............................................................................................................................... 45
Table 10: Power Consumption test .................................................................................................... 45
Table 11: Bluetooth v2.0 compliance ................................................................................................. 45
## LIST OF ABBREVIATIONS

<table>
<thead>
<tr>
<th>AT</th>
<th>Attention</th>
<th>PIN</th>
<th>Personal Identification Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATM</td>
<td>Automatic Teller Machine</td>
<td>RF</td>
<td>Radio Frequency</td>
</tr>
<tr>
<td>BLE</td>
<td>Bluetooth Low Energy</td>
<td>RISC</td>
<td>Reduced Instruction Set Architecture</td>
</tr>
<tr>
<td>BT</td>
<td>Bluetooth</td>
<td>RPi</td>
<td>Raspberry Pi</td>
</tr>
<tr>
<td>CAM</td>
<td>Computer Aided Manufacturing</td>
<td>RS-232</td>
<td>Recommended Standard 232</td>
</tr>
<tr>
<td>CAD</td>
<td>Computer Aided Design</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
<td>RX</td>
<td>Receive</td>
</tr>
<tr>
<td>DIY</td>
<td>Do It Yourself</td>
<td>SD</td>
<td>Secure Digital</td>
</tr>
<tr>
<td>EDA</td>
<td>Electronic Design Automation</td>
<td></td>
<td>SDRAM  Synchronous Dynamic Random Access Memory</td>
</tr>
<tr>
<td>EDR</td>
<td>Enhanced Data Rate</td>
<td>SFF</td>
<td>Small Form Factor</td>
</tr>
<tr>
<td>FEM</td>
<td>Finite Element method</td>
<td>SPI</td>
<td>Serial Peripheral Interface</td>
</tr>
<tr>
<td>FM</td>
<td>Frequency Modulation</td>
<td>SPP</td>
<td>Serial Port Profile</td>
</tr>
<tr>
<td>GND</td>
<td>Ground</td>
<td>SSH</td>
<td>Secure Shell</td>
</tr>
<tr>
<td>HSP</td>
<td>Headset Profile</td>
<td>SSP</td>
<td>Secure Simple Pairing</td>
</tr>
<tr>
<td>I2C</td>
<td>Inter-Integrated Circuit</td>
<td>TX</td>
<td>Transmit</td>
</tr>
<tr>
<td>IC</td>
<td>Integrated Circuit</td>
<td>UART</td>
<td>Universal Asynchronous Receiver-Transmitter</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>IrDA</td>
<td>Infrared Data Association</td>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
<tr>
<td>ISM</td>
<td>Industrial, Scientific, Medical</td>
<td>VCC</td>
<td>IC Power-supply Pin</td>
</tr>
<tr>
<td>MD5</td>
<td>Message Digest algorithm 5</td>
<td>VGA</td>
<td>Video Graphics Array</td>
</tr>
<tr>
<td>MIC</td>
<td>Military Industry Corporation</td>
<td>Wi-Fi</td>
<td>Wireless Fidelity</td>
</tr>
<tr>
<td>OUI</td>
<td>Organization Unique Identifier</td>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
<tr>
<td>PDA</td>
<td>Personal Digital Assistant</td>
<td>WPAN</td>
<td>Wireless Personal Area Network</td>
</tr>
<tr>
<td>PDF</td>
<td>Portable Document Format</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER ONE
INTRODUCTION

1.1 Brief History

There is no denying that the invention of the computer was the most contributing factor in the technological advances achieved in the past 50 years. Since their inception, Computers have been used in automation, space travel calculations, various simulations, and protein folding.

Moore’s Law – which states that the number of transistors in a given area doubles every 18-24 months – has been the governing factor in processing power increases achieved from new silicon fabrication technologies. Due to this, computers today are exponentially more powerful than the ones used a few decades ago.

![Figure 1: Processing Power growth over time](image)

However, due to almost reaching the physical limit of silicon transistor sizes; the number of transistors and consecutively processing power of new CPUs has been seeing a much lower rate of increase causing Moore’s Law to be pronounced dead and the industry to look for other means.
to improve. This led to the need to utilize what power is already available in better ways. This approach paved the way for Small form factor (SFF) and Embedded computers.

### 1.2 Rise of Embedded Computers

Embedded computers are small, special purpose systems that are designed to perform pre-defined tasks. Generally they are encapsulated into the device they are designed to control unlike general-purpose computers. Examples of these systems are ATMs, cell phones, printers, thermostats, calculators, PDAs, and video game consoles. [1]

The introduction of embedded systems along with SFF computers led to the development of many low cost computers for projects that do not require high power processing/specialization. The most popular of these devices is the Raspberry Pi (RPi). [2]

![Figure 2: A Raspberry Pi 1 board](image)

Originally developed to help get children interested in computer sciences, the RPi quickly became a hobbyist’s dream as the device was perfect for DIY projects and small scale ideas. This
led the development of the RPi to focus on its use as a controller and the implementation of Daughter Board modules that add a plethora of new features.

1.3 The Internet of Things

The easier availability of internet access around the globe led to the creation of the Internet of Things (IoT), a concept which emphasizes that everything can and should be connected to the internet where it can be monitored and controlled. Imagine turning on the air conditioner at home to set the optimum temperature while you are on your way back from work. Or on a bigger scale a “Smart City” where traffic lights, water/energy distribution, alarm control, and many other utilities are all interconnected to maintain optimum control. [3]

With IoT in mind, projects made using the RPi became more ambitious, incorporating more peripherals and using increasingly more complex code to achieve their intended results. The desire to be always online put more pressure into adding more connectivity options to the RPi, leading to more connectivity ports being added to new iterations of the device.

Some projects such as robotics required the RPi to be connected to a moving part, which made using wires or cables to connect to it a hindrance. Other applications required long range access or multiple connections to be made in a relatively small space, rendering using wires a huge hassle and a chore to modify. Wireless communication provided a solution to these problems while its main drawbacks (lower transfer rates, inconsistent channel strength) were a non-issue to RPi projects.

1.4 Solving the Connectivity Problem

Newer RPi units (version 3+) have built in wireless connectivity available, but since even the old versions sold millions of units globally, the issue of implementing wireless communication for these units becomes a necessity. In this paper, Bluetooth serial communication is implemented via Raspberry Pi’s UART serial interface.
1.5 Objectives

1. Establish a working Wireless interface between 2 Raspberry Pi units.
2. Explore additional uses for a wireless interface on Raspberry Pi.

1.6 Thesis Layout

- **Chapter 2** provides some needed background information on wireless while going into detail on how Bluetooth works. It details various software tools that are used in the design and shows the basics of antenna
- **Chapter 3** Describes the intended courses of action, the full design procedure, the realization of the design into usable hardware. It also describes how the hardware and software
- **Chapter 4** Shows how the testing was conducted the results of the performed tests, a brief discussion on each test and their discussion.
- **Chapter 5** is comprised of project conclusion, approach discussion, and possible future work.
CHAPTER TWO

TECHNOLOGY OVERVIEW

2.1 Wireless Communication

Wireless communication, or sometimes simply wireless, is the transfer of information or power between two or more points that are not connected by an electrical conductor. The most common wireless technologies use radio waves. With radio waves distances can be short, such as a few meters for Bluetooth or as far as millions of kilometers for deep-space radio communications. It encompasses various types of fixed, mobile, and portable applications, including two-way radios, cellular telephones, personal digital assistants (PDAs), and wireless networking. Other examples of applications of radio wireless technology include GPS units, garage door openers, wireless computer mice, keyboards and headsets, headphones, radio receivers, satellite television, broadcast television and cordless telephones.

2.1.1 Wireless Standards

There are various communication methods that can be used to implement a wireless link between two points. Table (1) compares attributes for the most common standards [4], which are:

1. **Infrared**: A common, inexpensive, and easy to use wireless communication technology using IR light. It’s very similar to visible light except for its slightly longer wavelength.
2. **Zigbee**: a wireless technology developed as an open global standard to address the unique needs of low-cost, low-power wireless networks. The Zigbee standard operates on the IEEE 802.15.4 physical radio specification and operates in unlicensed bands including 2.4 GHz, 900 MHz and 868 MHz. [5]
3. **Bluetooth**: A Wireless Personal Area Network (WPAN) using frequency hopping spread spectrum for short-range transmission of digital audio and data using radio waves. It operates on the unlicensed industrial, scientific, medical (ISM) 2.4 GHz band.
4. **Wi-Fi**: Also operates in the ISM 2.4 GHz and 5 GHz bands. It’s a lot faster and has more range than Bluetooth but comes at a higher cost per device.
### Table 1: Wireless communication standards comparison

<table>
<thead>
<tr>
<th>Name</th>
<th>Wi-Fi</th>
<th>Bluetooth</th>
<th>Zigbee</th>
<th>Infrared</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE Standard</td>
<td>802.11b/g/n/ac</td>
<td>802.11a</td>
<td>802.14</td>
<td>802.11</td>
</tr>
<tr>
<td>Interface Type</td>
<td>UART, SPI</td>
<td>UART</td>
<td>UART</td>
<td>IrDA</td>
</tr>
<tr>
<td>Range</td>
<td>400 meters</td>
<td>10 meters</td>
<td>90 meters</td>
<td>1m</td>
</tr>
<tr>
<td>Transfer Rate</td>
<td>150mbps</td>
<td>2.1mbps</td>
<td>250 kbps</td>
<td>115.2kbps</td>
</tr>
<tr>
<td>Password Protection</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Line of Sight</td>
<td>Not required</td>
<td>Not required</td>
<td>Not required</td>
<td>Required</td>
</tr>
<tr>
<td>Relative Cost</td>
<td>Medium-High</td>
<td>Medium</td>
<td>High</td>
<td>Ultra-low</td>
</tr>
<tr>
<td>IC availability</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>IC size</td>
<td>Large</td>
<td>Small</td>
<td>Small</td>
<td>Tiny</td>
</tr>
</tbody>
</table>

Considering all the above standards features and their compatibility with the intended design and using a process of elimination, it can be observed that:

- Infrared isn’t a viable option due to its very low range, lack of security, very low throughput and the requirement of having no objects between the 2 points of connection (direct line of sight).
- Zigbee has very high range and matches Bluetooth and Wi-Fi in other areas, but the low throughput, high relative cost and low IC availability makes it an undesired choice.
- Wi-Fi offers feature parity with Bluetooth and exceeds it in almost all areas. But due to its relatively higher cost, larger IC size and the fact that the Raspberry Pi’s UART interface won’t be using all of Wi-Fi’s high throughput makes Bluetooth a better option.
- Bluetooth delivers the best balance between throughput, cost, size, range, and high availability on its custom ICs.

For the aforementioned reasons as well as cost-efficiency, Bluetooth was chosen as the communication standard.
2.2 Bluetooth Standard

Bluetooth networks (commonly referred to as piconets) use a master/slave model to control when and where devices can send data. In this model, a single master device can be connected to up to seven different slave devices. Any slave device in the piconet can only be connected to a single master. [6]

Figure 3: Bluetooth Network Topology

Every single Bluetooth device has a unique 48-bit address, commonly abbreviated BD_ADDR. This will usually be presented in the form of a 12-digit hexadecimal value. The most-significant half (24 bits) of the address is an organization unique identifier (OUI), which identifies the manufacturer. The lower 24-bits are the more unique part of the address. [7]

Bluetooth devices can also have user-friendly names given to them. These are usually presented to the user, in place of the address, to help identify which device it is.

Creating a Bluetooth connection between two devices is a multi-step process involving three progressive states:

1. Inquiry – If two Bluetooth devices know absolutely nothing about each other, one must run an inquiry to try to discover the other. One device sends out the inquiry request, and
any device listening for such a request will respond with its address, and possibly its name and other information.

2. **Paging (Connecting)** – Paging is the process of forming a connection between two Bluetooth devices. Before this connection can be initiated, each device needs to know the address of the other (found in the inquiry process).

3. **Connection** – After a device has completed the paging process, it enters the connection state. While connected, a device can either be actively participating or it can be put into a low power sleep mode.

   - **Active Mode** – This is the regular connected mode, where the device is actively transmitting or receiving data.
   - **Sniff Mode** – This is a power-saving mode, where the device is less active. It’ll sleep and only listen for transmissions at a set interval (e.g. every 100ms).
   - **Hold Mode** – Hold mode is a temporary, power-saving mode where a device sleeps for a defined period and then returns back to active mode when that interval has passed. The master can command a slave device to hold.
   - **Park Mode** – Park is the deepest of sleep modes. A master can command a slave to “park”, and that slave will become inactive until the master tells it to wake back up.

**Bonds** are created through one-time a process called **pairing**. When devices pair up, they share their addresses, names, and profiles, and usually store them in memory. The also share a common secret key, which allows them to bond whenever they’re together in the future.

Pairing usually requires an **authentication process** where a user must validate the connection between devices. The flow of the authentication process varies and usually depends on the interface capabilities of one device or the other. Sometimes pairing is a simple “Just Works” operation, where the click of a button is all it takes to pair (this is common for devices with no UI, like headsets). Other times pairing involves matching 6-digit numeric codes. Older, legacy (v2.0 and earlier), pairing processes involve the entering of a common PIN code on each device. The PIN code can range in length and complexity from four numbers (e.g. “0000” or “1234”) to a 16-character alphanumeric string.
Bluetooth profiles are additional protocols that build upon the basic Bluetooth standard to more clearly define what kind of data a Bluetooth module is transmitting. While Bluetooth specifications define how the technology works, profiles define how it’s used.

The profile(s) a Bluetooth device supports determine(s) what application it’s geared towards. A hands-free Bluetooth headset, for example, would use headset profile (HSP), while a Nintendo Wii Controller would implement the human interface device (HID) profile. For two Bluetooth devices to be compatible, they must support the same profiles.

The most important profile is the Serial Port Profile (SPF) as it replaces serial communication interfaces (like RS-232 or in this case, UART) with Bluetooth; SPP is great for sending bursts of data between two devices. It’s is one of the more fundamental Bluetooth profiles (Bluetooth’s original purpose was to replace RS-232 cables after all).

Using SPP, each connected device can send and receive data just as if there were RX and TX lines connected between them. Two Raspberry Pi’s for example, could converse with each other from across rooms, instead of from across the desk.

Figure 4: Serial Cable VS Bluetooth Raspberry Pi interface
2.2.1 Bluetooth Frequency Bands

Bluetooth frequencies are all located within the 2.4 GHz ISM band. The ISM band typically extends from 2400 MHz to 2 483.5 MHz (i.e. 2.4000 - 2.4835 GHz). Bluetooth channels are spaced 1 MHz apart, starting at 2402 MHz and finishing at 2480 MHz. This can be calculated as $2401 + n$, where $n$ varies from 1 to 79. This arrangement of Bluetooth channels gives a guard band of 2 MHz at the bottom end of the band and 3.5 MHz at the top.

The industrial, scientific, and medical radio band (ISM band) refers to a group of radio bands or parts of the radio spectrum that are internationally reserved for the use of radio frequency (RF) energy intended for scientific, medical and industrial requirements rather than for communications. ISM bands are generally open frequency bands, which vary according to different regions and permits. In some countries the ISM band allocation does not allow the full range of frequencies to be used. In France, Japan and Spain, the hop sequence has to be restricted to only 23 frequencies because of the ISM band allocation is smaller.

There are also some Bluetooth frequency accuracy requirements for Bluetooth transmissions. The transmitted initial center frequency must be within $\pm 75$ kHz from the receiver center frequency. The initial frequency accuracy is defined as being the accuracy before any information is transmitted and as such any frequency drift requirement is not included.

2.2.2 Common Bluetooth Versions and Classes

Bluetooth has been constantly evolving since it was conceived in 1994. The most recent update of Bluetooth, Bluetooth v4.0, is just beginning to gain traction in the consumer electronics industry, but some of the previous versions are still widely used. Here’s a rundown of the commonly encountered Bluetooth versions:

2.2.2.1 Bluetooth v1.2

The v1.x releases laid the groundwork for the protocols and specifications future versions would build upon. Bluetooth v1.2 was the latest and most stable 1.x version.

These modules are rather limited compared to later versions. They support data rates of up to 1 Mbps (more like 0.7 Mbps in practice) and 10 meter maximum range.
2.2.2.2 Bluetooth v2.1 + EDR

The 2.x versions of Bluetooth introduced enhanced data rate (EDR), which increased the data rate potential up to 3 Mbps (closer to 2.1 Mbps in practice). Bluetooth v2.1, released in 2007, introduced secure simple pairing (SSP), which overhauled the pairing process.

2.2.2.3 Bluetooth v3.0 + HS

It has an optimum speed of 24 Mbps. The data is actually transmitted over a Wi-Fi (802.11) connection. Bluetooth is only used to establish and manage a connection.

2.2.2.4 Bluetooth v4.0 and Bluetooth Low Energy

Bluetooth 4.0 split the Bluetooth specification into three categories: classic, high-speed, and low-energy. Classic and high-speed are a reference to Bluetooth versions v2.1+EDR and v3.0+HS respectively. The real standout of Bluetooth v4.0 is Bluetooth low energy (BLE). BLE is a massive overhaul of the Bluetooth specifications, aimed at very low power applications. It sacrifices range (50m instead of 100m) and data throughput (0.27 Mbps instead of 0.7-2.1 Mbps) for a significant savings in power consumption. BLE is aimed at peripheral devices which operate on batteries, and don’t require high data rates, or constant data transmission.

**Bluetooth classes** are determined from the operating range, they are defined as follows

<table>
<thead>
<tr>
<th>Class</th>
<th>Operating Range</th>
<th>Maximum Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100 meters</td>
<td>100mw</td>
</tr>
<tr>
<td>2</td>
<td>10 meters</td>
<td>2.5mw</td>
</tr>
<tr>
<td>3</td>
<td>1 meter</td>
<td>1mw</td>
</tr>
</tbody>
</table>

- For this design, Bluetooth v2.1 was chosen because its data rate closely matches the maximum data rate of the Raspberry Pi.
- The range of Bluetooth v2.1 (10m) is very close to the range at which the majority of practical applications for the Raspberry Pi are implemented.
Due to the Raspberry Pi’s low number of USB ports (only two), a USB implementation was ruled out which led to the adaptation of a Serial/Bluetooth design.

2.3 RF Design

An RF transducer is a device that sends and receives data sent over a specific Radio frequency. Though individual RF transducers may differ from one another, the majority share the same building blocks.

![RF Transducer Block Diagram](image)

**Figure 5: RF Transducer Block Diagram**

The above figure shows the block diagram for an RF transducer. The design of an RF system can be said to be the design of these blocks.

There are two main ways to design a RF circuit; first, by designing every RF block on separate ICs/circuits and designing the interfaces between them, or secondly, by designing a single IC which contains most or all the RF blocks within it.

The first design path provides more flexibility and freedom in controlling the properties of each block, can be maintained more easily (as blocks can be tested or replaced without affecting other blocks), but requires matching circuits to developed between each block and next, to insure a consistent internal impedance (50 ohm).
The second design path provides a more compact final solution and does not require matching circuits except at the input and output ports, due to the extremely short distances between blocks within an IC.

2.3.1 Antennas
Antenna or aerial refers to an electronic device used for transmission of radio or other electromagnetic waves. It converts electricity into electromagnetic waves. Antenna was firstly built by a German physicist named Heinrich Hertz. Antennas are very important parts of an electric system as they play the role of a connector between the transmitter and free space or free space and the receiver.

It is used for focusing waves into a narrow beam for transmission. It collects the signals for the receiver end. Antennas are employed in various systems and in different forms like in radio and television broadcasting, point to point radio communication, wireless LAN, radar, etc. Transmitting antennas are used for radiating electromagnetic radiation in a desired direction, whereas a receiving antenna receives electromagnetic radiation. Size and shape of antennae is in accordance with the size of the signal’s wavelength.

2.3.1.1 Properties of Antennas
Antenna Gain: The parameter that measures the degree of directivity of antenna’s radial pattern is known as gain. An antenna with a higher gain is more effective in its radiation pattern. Antennas are designed in such a way that power raises in wanted direction and decreases in unwanted directions.

\[ G = \frac{\text{power radiated by an antenna}}{\text{power radiated by reference antenna}} \]

2.3.1.2 Reflection coefficient:
In physics and electrical engineering, the reflection coefficient is a parameter that describes how much of an electromagnetic wave is reflected by an impedance discontinuity in the transmission medium. It is equal to the ratio of the amplitude of the reflected wave to the incident wave, with each expressed as phasors. For example, it is used in an electrical transmission line to calculate how much of the electromagnetic wave is reflected by an impedance. The reflection coefficient
of a load is determined by its impedance $Z_L$ (Load impedance) and the characteristic impedance of the transmission line $Z_0$.

$$\Gamma = \frac{V^-}{V^+} = \frac{Z_L - Z_0}{Z_L + Z_0}$$

**Aperture:** This aperture is also known as the effective aperture of the antenna that actively participate in transmission and reception of electromagnetic waves. The power received by the antenna gets associated with collective area. This collected area of an antenna is known as effective aperture.

$$P_r = P_d \times A \text{ watts}$$

$$A = \frac{Pr}{P_d} \text{ m}^2$$

**Directivity and Bandwidth:** The directive of an antenna is defined as the measure of concentrated power radiation in a particular direction. It may be considered as the capability of an antenna to direct radiated power in a given direction. It can also be noted as the ratio of the radiation intensity in a given direction to the average radiation intensity. Bandwidth is one of the desired parameters to choose an antenna. It can be defined as the range of frequencies over which an antenna can properly radiates energy and receives energy.

**Polarization:** An electromagnetic wave launched from an antenna may be polarized vertically and horizontally. If the wave gets polarized in the vertical direction, then the $E$ vector is vertical and it requires a vertical antenna. If vector $E$ is in horizontal way, it needs a horizontal antenna to launch it. Sometimes, circular polarization is used, it is a combination of both horizontal and vertical ways.

**Effective Length:** The effective length is the parameter of antennas that characterizes the efficiency of the antennas in transmitting and receiving electromagnetic waves. Effective length can be defined for both transmitting and receiving antennas. The ratio of EMF at the receiver input to the intensity of the electric field occurred on the antenna is known as receivers’ effective length. The effective length of the transmitter can be defined as the length of the free space in
conductor, and current distribution across its length generates same field intensity in any direction of radiation.

$$\text{Effective Length} = \frac{\text{Area under non-uniform current distribution}}{\text{Area under uniform current distribution}}$$

### 2.3.1.3 Basic Antenna Types

**Dipoles and Monopoles** - These antennas are greatly preferred in the field of wireless communication due to their omnidirectional feature and low cost. Monopole antenna is half of a dipole antenna. Monopole uses a ground plane in place of the second wire. A dipole antenna is made by using two terminals. Radio frequency current slows into these poles. The poles are symmetrical and are of equal lengths. It extends in just opposite directions from the feed point. Monopole antenna possesses a single element. These are very easy to build and install, and therefore have gained some popularity. They can be installed in various forms like in commercial flat top versions. A typical gain for a dipole antenna is 2dB, and the bandwidth is generally around 10%.

**Loop Antenna** – This antenna is generally designed for AM broadcast and the Long wave lengths. It is directional. They are further categorized into small loop and large loop types. Usually, these antennae are small in size. It is made of a conductor which is bent into the shape of a closed curve like a circle or square. The larger the size of loop antennae is always considered to be better. One wavelength perimeter loop works like a folded dipole antenna. However, its impedance is greater than that of half wavelength dipole antennae. Multi turn loop antennas are not good for transmission due to high losses and high level of heat dissipation.

**Micro-strip Antenna** – These were invented by Bob Munson in 1972. These antennas became highly popular during the 1970s. These antennas are also known as the patch antennas. Patch refers to the rectangular component which is photo-etched from one side of the board. The majority of elements are fed by a coaxial conducted which is soldered to the reverse part of the ground plate. It can be considered as a planar antenna which is separated from a ground plate by a dielectric medium like air. Their simple 2 D physical geometry and design makes then inexpensive and easy to manufacture.
**Helical Antenna** – These are also known as helical antennas. A travelling wave antenna which is shaped in the form of a corkscrew is a famous type of helical antennae. Its wider bandwidth and easy construction are two key benefits. The current in these antennas travel along the antenna and the phase keeps on changing continuously. It works best of pitch angles between 12 and 14 degrees. Unlike dipole, which radiates normal to its axis, these antennas are circularly polarized.

**Horn Antenna** – Its flared appearance is the reason behind its name. They operate as high as 150 GHz. Usually they are used as microwave antennas. Its performance is easily controllable. They possess a directional radiation pattern with a high antenna gain. They exhibit very little loss. It is usually used as the active component of a dish antenna. This antenna is generally used in unification with waveguide feeds. It is generally made of a small rectangular cylindrical metal tube which is closed from one end. This tube flares into an open ended conical or pyramidal shaped of horn from the other side.

**Table 3: Antenna design types comparison**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Mono/Dipole</th>
<th>Loop</th>
<th>Micro-strip</th>
<th>Helical</th>
<th>Horn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matching</td>
<td>Requires matching circuit between antenna and transmission line.</td>
<td>Impedance matched by connecting the feed line to a smaller feed loop inside the area surrounded by the main loop.</td>
<td>Matching and phase adjustment can be performed with printed micro-strip feed structures.</td>
<td>Matching impedance is changed from the nominal 50 ohms to between 25 and 35 ohms base impedance.</td>
<td>Shape of horn acts as a matching transformer.</td>
</tr>
<tr>
<td>Size</td>
<td>Bulky</td>
<td>Bulky</td>
<td>Small</td>
<td>Large</td>
<td>large</td>
</tr>
<tr>
<td>Cost</td>
<td>Low</td>
<td>Low</td>
<td>low</td>
<td>High due to size</td>
<td>Medium</td>
</tr>
<tr>
<td>Directivity</td>
<td>omnidirectional</td>
<td>Plane of loop</td>
<td>omnidirectional</td>
<td>omnidirectional</td>
<td>Plane of Horn</td>
</tr>
<tr>
<td>Antenna Gain</td>
<td>5-6dBi</td>
<td>-</td>
<td>6-9dBi</td>
<td>-</td>
<td>Low</td>
</tr>
<tr>
<td>Coverable</td>
<td>Not coverable</td>
<td>Not coverable</td>
<td>Coverable</td>
<td>Not coverable</td>
<td>Not coverable</td>
</tr>
</tbody>
</table>
2.4 Computer Aided Design Programs

Now a day, there are many CAD design programs specialized in electronic circuit and PCB design. In this project, a high frequency simulation tool is required in order to determine the most optimal antenna design, as well as a circuit simulation and PCB design tool.

2.4.1 High frequency simulation tools

The following are high frequency analysis tools currently available on the market:

- **ANSYS HFSS**: A high-frequency electromagnetic field simulator, provided by ANSYS (a company that specializes in engineering simulation programs). HFSS offers state-of-the-art solver technologies based on finite element, integral equation, asymptotic, and advanced hybrid methods to solve a wide range of microwave, RF and high-speed digital applications. HFSS is based on Finite Element Method (FEM) which is more accurate for designing antennas; however it is not easy to use and is rather difficult to obtain.

- **CST Microwave Suite**: A part of a family of simulation tools, provided my Dassault Systems, called CST Studio Suite. It provides fast and accurate 3D simulation of high frequency devices. It enables the fast and accurate analysis of antennas, filters, couplers, planar and multi-layer structures and SI and EMC effects. CST microwave suite is based upon Finite Integration Technique (FIT), which is also popular among antenna designers due to ease in simulations, but provides less accurate results then its FEM counterpart.

- **Microwave Office**: Produced by National Instruments within its AWR Design Environment, Microwave office is a high frequency simulation tool that provides an intuitive interface. AWR software is used for radio frequency (RF), microwave and high frequency analog circuit and system design. Typical applications include cellular and satellite communications systems and defense electronics including radar, electronic warfare and guidance systems.

2.4.2 PCB Design Tools

The most popular EDA design tools are
• **Altium Designer:** PCB design software, an electronic design automation (EDA) software package for printed circuit board, FPGA and embedded software design, and associated library and release management automation. It is developed and marketed by Altium Limited of Australia. Many universities’ circuit specialties have the specific courses to learn how to use Altium, and almost all of the circuit companies have used it for their work.

• **EAGLE:** A scriptable electronic design automation (EDA) application with schematic capture, printed circuit board (PCB) layout, auto-router and computer-aided manufacturing (CAM) features. EAGLE stands for Easily Applicable Graphical Layout Editor and is developed by CadSoft Computer GmbH. The company was acquired by Autodesk Inc. in 2016.

• **AutoTRAX DEX:** AutoTRAX DEX is a Windows operating system schematic design and PCB layout program with 3D part and board visualization. Unlike other EDA programs, AutoTRAX DEX integrated all of the schematic and PCB design into a single XML project file. It uses parametric parts which define the parts schematic symbols, PCB footprint and 3D package models using numeric parameters. AutoTRAX DEX uses the Microsoft Windows platform and runs on Windows XP, Windows Vista and Windows 7. The file format is open and viewable with any text editor as it is XML based, and the XML schema is fully documented.

• **KiCad:** a free open source software suite for electronic design automation (EDA). It facilitates the design of schematics for electronic circuits and their conversion to PCB designs. KiCad was originally developed by Jean-Pierre Charras. It features an integrated environment for schematic capture and PCB layout design. Tools exist within the package to create a bill of materials, artwork, Gerber files, and 3D views of the PCB and its components.
CHAPTER THREE
DESIGN AND IMPLEMENTATION

3.1 Design Vision

- There are 2 proposed courses of action to solve the wireless connectivity issue
  
  **Full design:** Design and implement a fully functioning daughter board from scratch.
  
  **Ready-made Modules:** Select and purchase a pre-fabricated module from the online e-tailers as a contingency plan in case the fully designed daughter board fails to function.

- The two courses of actions are executed simultaneously in order to ensure optimum success chance for the project.

The general steps taken to implement the full design are

1. **Picking Components**
   Checking available online ICs from reputable e-tailers (Digikey, Mouser, Farnell) and picking the best suited SoC for the design.

2. **PCB Design**
   Altium designer was chosen as the EDA tool because of its industry-wide usage, relatively easy to use UI, and the availability of local courses in its use.

3. **Design Fabrication**
   Done in the Military Industry Corporation’s Alzargaa fabrication plant, with the written consent of the Department of Electrical and Electronics Engineering.

4. **Debugging/Verification**
   Testing is done on the fabricated samples to verify that they are in a fully operational state, debugging is done in case any errors in operations are found.

3.1.1 The Full-Design route

Due to how wide spread the Bluetooth wireless communication protocol has become in consumer electronics, many companies manufacture RF integrated circuit for the Bluetooth spectrum. As Bluetooth communication is commonly used for small electronic devices, (i.e. phones, wireless mice or keyboards, etc.), the compact nature of integrated RF circuits is more
desirable then the flexibility afforded by individual block design. Therefore, many manufactures provide complete System on Chip (SOC) IC for Bluetooth transducers.

This greatly simplifies the design process, as it now only requires a choice of IC and antenna design.

The proposed system architecture consists of two Raspberry Pi and two Bluetooth modules which are connected through UART, as shown in fig. (6). The desired flow chart for the system is shown in figure (7).

3.1.1.1 Main components

1. Raspberry Pi
Raspberry Pi is a small sized computer board consists of RISC architecture based processor ARM 11, 700 MHz processor. It is equipped with SoC “BCM 2835” providing onboard 512 MB of SDRAM, an interface for output unit such as VGA Monitor, touch screen and input unit such as keyboard, keypad, and mouse.

2. Serial interface
Serial Communication is a form of communication providing bits of bytes being transferred serially. It includes I2C, SPI, UART, and USB. The Raspberry Pi comes equipped with SPI, UART and I2C interfaces.

3. Bluetooth Daughterboard Module
Consists of a serial interface, a Bluetooth transceiver and a System-On-Chip (SoC) IC that handles Serial-to-Bluetooth translation when sending, and Bluetooth-to-Serial when receiving
Figure 6: System Block Diagram

Figure 7: System Flowchart
### 3.1.1.2 Daughter board Component Selection

In order to realize the design, the main component must be selected first. This is the Microcontroller which will handle the Bluetooth-Serial Interface. The table below shows a list of Bluetooth RF integrated circuits.

<table>
<thead>
<tr>
<th>Name</th>
<th>Manufacturer</th>
<th>Supply Voltage (V)</th>
<th>Current (mA)</th>
<th>Range (m)</th>
<th>Description</th>
<th>Datasheet Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCF87852</td>
<td>Philips</td>
<td>1.8</td>
<td>13</td>
<td>-</td>
<td>BT SoC</td>
<td>[8]</td>
</tr>
<tr>
<td>HMC310MS</td>
<td>Hittite</td>
<td>3</td>
<td>24</td>
<td>1-10</td>
<td>2.4 GHz RF Transceiver</td>
<td>[9]</td>
</tr>
<tr>
<td>CC2510FX</td>
<td>Texas Instruments</td>
<td>3.6</td>
<td>17-23</td>
<td>10</td>
<td>2.4 GHz RF Transceiver SoC</td>
<td>[10]</td>
</tr>
<tr>
<td>STLC2690</td>
<td>ST. Micro</td>
<td>2.5</td>
<td>30</td>
<td>10-100</td>
<td>Bluetooth and FM transceiver SoC</td>
<td>[11]</td>
</tr>
<tr>
<td>450-0053</td>
<td>TiWi5</td>
<td>3-4</td>
<td>100-165</td>
<td>-</td>
<td>W-LAN, BT transceiver</td>
<td>[12]</td>
</tr>
<tr>
<td>CSR1012</td>
<td>CSR</td>
<td>3.6-5</td>
<td>50</td>
<td>10</td>
<td>BT SoC</td>
<td>[13]</td>
</tr>
</tbody>
</table>

- The CSR1012 QFN IC provides a Bluetooth transceiver circuit, RF modulation and demodulation, ADC/DCA units, internal processing, memory and a UART serial communication terminal. This, as well as its cheap price and considerable availability in the market place, makes it an ideal chip for the Bluetooth daughterboard.

The other passive components (capacitors, inductors, ROMs, etc.) are selected with the main factor being compatibility with the CSR1012.
3.1.1.3 Circuit Schematics

The first step in designing a PCB is to design or acquire a circuit schematic. The choice of the CSR 1012 IC greatly simplified the design process as the manufacture provides a recommended circuit. This circuit was used as well as the Altium Designer’s schematic tool to produce a circuit schematic for the Bluetooth module.

Figure 8: Recommended Circuit Diagram for the CSR1012

Altium Designer operates based on user libraries; these include both component schematic symbols and PCB (footprints). These libraries can be updated, modified and augmented by the programs library editor. A schematic symbol library was created for the required components of the circuit, and then used to produce the circuit schematic.
Figure 9: Schematic library, Schematic symbol for CSR1012 IC
Figure 10: System Schematic
In order to start the PCB layout design, PCB footprints must be made for all the components. A PCB footprint or land pattern is the arrangement of pads (in surface-mount technology) or through-holes (in through-hole technology) used to physically attach and electrically connect a component to a printed circuit board. Recommended footprints are usually provided by the Manufacturer in the component’s datasheet. Each component must have a footprint made and assigned to it. Altium also provides a PCB footprint library editor, for the design of the land patterns, and they are assigned to the components in the schematic sheet.

After which, a pcb board design can begin. Altium automatically places the footprints of the components into the PCB editor, and illustrates the required connections based on the schematic. After which the user must decide on the location of the components and place the required track connections.

Figure 11: PCB Library, footprint for CSR1012 IC
The antenna can’t be modeled in Altium Designer, Use of ICCT is needed and results should be imported back into Altium.

### 3.1.1.4 Antenna Simulation
Microwave office is a High frequency simulation program that can be used to test antenna design against several parameters. Due to the program’s intuitive and easy to use user interface, availability and previous experience with the program, it was deemed this to the most suitable option.

In order for the antenna to perform reliably and accurately, the design should hold up against certain RF specifications. Most of the Bluetooth SIG (special interest group) specifications are fulfilled by the SOC IC. The properties relevant to the antenna design are the reflection coefficient at 2.4 GHz and the antenna gain.

As the daughterboard should be small, low cost, and is only required to provide a 10-100m range, a Micro-strip antenna is the most suited to the project needs.

There are several shapes and designs used in micro-strip antennas; the majority of Bluetooth-Serial communication modules use a meander micro-strip antenna. The meander shape was chosen due to

![Figure 12: Final PCB Design](image-url)
Figure 13: Patch micro-strip being simulated in microwave office

- It’s a widely used industry standard that is easy to implement
- Its relatively small and compact, taking small volume while providing high effective length.

A simulation of the meander antenna was done using microwave office AWR EMSight simulator, the results showed compliance with the required specifications as shown in the results section of chapter 4.

Figure 14: Meander Antenna Design
3.1.1.5 Fabrication & Verification

Fabrication was done at the MIC.

Figure 15: a) CNC Drilling Machine b) Traced- PCB

The steps taken to complete fabrication are:

1. The completed design GERBER and NCDRILL files was handed over to MIC personnel.
2. The files were used to instruct the CNC machines to create the fabricated PCB containing the traces/routes of the design.
3. Component soldering was done by hand due to the unavailability of solder masks.
The final product was tested using a USB programmer for the CSR1012 to check if the module is working. However, the unit showed no response when voltage was applied. Several configurations were tried but all of them resulted in the same outcome.

### 3.1.1.6 Possible Failure Reasons

1. The copper sheet thickness was extremely thin and could be easily damaged while fabricating.

2. Components could not be automatically soldered on the PCB. Instead, tedious (and possibly erroneous) manual soldering was required.

3. The inability to use plated vias forced the use of free wires as a connection between the source ground and the ground plane.

Due to the above reasons, the Full-design approach could not be implemented. Hence, a move to the ready-made module course of action was required.
3.1.2 Ready-made Modules

There are widely available ready-made modules that can be connected to the Raspberry Pi pins in order to provide Bluetooth connectivity, going through the available catalogue of popular online retailers (DigiKey, Mouser, Farnell). After a brief review of available choices, the final candidates were selected and shown in Table (5).

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Bluetooth Protocol</th>
<th>Operating Voltage (V)</th>
<th>Breakout board included</th>
<th>Cost ($)</th>
<th>Datasheet link</th>
</tr>
</thead>
<tbody>
<tr>
<td>USR-BLE100</td>
<td>V4.1</td>
<td>1.9-5.5</td>
<td>No</td>
<td>7</td>
<td>[14]</td>
</tr>
<tr>
<td>EGBT-BT</td>
<td>V2.0+EDR</td>
<td>3.3</td>
<td>No</td>
<td>12</td>
<td>[15]</td>
</tr>
<tr>
<td>SPBT2632C2A</td>
<td>V3.0</td>
<td>2.5</td>
<td>No</td>
<td>16</td>
<td>[16]</td>
</tr>
<tr>
<td>HC-05</td>
<td>V2.0+EDR</td>
<td>3.3-5</td>
<td>Yes</td>
<td>11</td>
<td>[17]</td>
</tr>
</tbody>
</table>

From the above table, the HC-05 was picked due to

- Its operating voltage closely matches the Raspberry Pi.
- The inclusion of a breakout board.
- Relative low cost
- Wide availability of online documentation and resources.


3.2 Design Implementation

Full component list:

4. Raspberry Pi 1 Model A+
5. Raspberry Pi 2 Model B+
6. 2x HC-05 BT-Serial Modules
7. 8x Female-to-Female Jumper cables.
8. 2x Micro-USB cables

3.2.1 Circuit Layout

In UART, the wiring is spliced; meaning each Transmission (Tx) port is connected to the Reception port (Rx) on the other side.

The HC-05 Pinout is as follows

![HC-05 Pinout](image)

**Figure 17: HC-05 Pinout**

There is a pin number difference between the Raspberry Pi 1 and 2 models as shown in figure (10). [18]
Figure 18: Raspberry Pi Pinouts; (a) Model 1 A+ (b) Model 2 B+

Hence, the required connections are:

Figure 19: Raspberry Pi to HC-05 Interfacing; (a) Model 2 B+ (b) Model 1 A+
The 2 Micro-USB cables are used to provide power to the Raspberry Pi models, the final hardware setup looks like this

![Figure 20: Final Setup](image)

3.2.2 Software Setup

3.2.2.1 Raspberry Pi

1. The Raspberry Pi comes with an unused Secure Digital (SD) card slot, two SD memory units were loaded with official Raspbian operating system which is based on Debian Linux. For optimum compatibility, the system needs to be updated to the latest version.

   The following commands were ran in the Linux Terminal.

   ```
   sudo apt-get update
   sudo apt-get upgrade
   ```

   These 2 commands download the latest kernel/packages from online repositories.

2. By default the Raspberry Pi’s serial port is configured to be used for console input/output. This can help to fix problems during boot, or to log in to the Pi if the video and network are not available.
To be able to use the serial port to connect and talk to other devices (e.g. a modem, a printer), the serial port console login needs to be disabled.

```
sudo raspi-config
```

This brings up the screen in figure (10).

![Raspi-Config menu](image)

**Figure 21: Raspi-Config menu**

From this menu, go to advanced options > Serial > then select “No” at the prompt for the login shell to be enabled.

3. To setup the HC-05 or establish a serial port connection, a program named Minicom has to be installed.

```
sudo apt-get install minicom
```

With a quick system reboot after this is done, the Serial Interface is ready to be utilized.

3.2.2.2 HC-05

The HC-05 has 2 operational modes:

- **Normal Operation**: where the module sends and receives data as requested.
- **AT-Command Mode**: used to configure the module and setup various values such as baud rate, Master/Slave mode, password change, Address binding….etc.

There are two AT modes of functionality; both accessed using the AT-Command Mode button [19]:

35
1. Constant Baud Rate mode: accessed via pressing the button while powering up the device then releasing it immediately afterwards. The baud rate is set to a constant 9600 and all other commands are accepted.

2. Variable Baud Rate mode: accessed via holding the button pressed for the whole duration of configuration. The baud rate is set to 38400 but can be changed, all other commands are accepted.

Utilizing the above information; powering on the Raspberry Pi and the HC-05 into Variable Baud Rate mode, A terminal window is then opened and the following is entered.

```
sudo minicom -b 38400 -o -D /dev/ttyAMA0
```

This establishes a serial connection to the device (ttyAMA0 is the code used for the first serial device connected to the Pi) using 38400 baud rate. To check if the connection is successful an “AT” command is sent in the bus to which the response should be “OK”

```
AT  (sent)
OK (received)
```

the HC-05 is now connected in configuration mode, the next step is to know the unit’s Bluetooth address.

```
AT+ADDR? Returns a line
```

```
AT+ADDR:1234:56:abcdef Where 12:34:56:ab:cd:ef is the Bluetooth address of the module.
```

Next is setting the role of modules, one as master and the other as slave as required by Bluetooth communication protocol. This is achieved via

```
AT+ROLE=0 Sets role to Slave
AT+ROLE=1 Sets role to Master
```

For a connection to be made the password has to be identical for the two modules, this is achieved by running this command on both units

```
AT+PSWD=6573 (6573 is the new password chosen)
```
By default, the Raspberry Pi’s UART serial interface has a maximum baud rate of 115200, to set a matching rate in the modules the following command is used:

\[ \text{AT+UART}=115200,0,0 \]

sets baud rate to 115200, number of stop bits to 1, number of parity bits to 0.

For the modules to automatically establish a connection to each other without waiting for user consent, they have to be bound to each other. Essentially removing the ability for any other device to connect to a module besides its bound pair.

\[ \text{AT+BIND}=1234:56:abcdef \]

with the address belonging to the other model. This is run on each module consecutively.

Once all the above commands are executed, the modules will continuously seek each other when they are active, and automatically establish a connection, keeping it up until they are powered off.

Table 6: A list of other useful AT Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
<th>Parameter</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+NAME=&lt;Param&gt;</td>
<td>OK</td>
<td>Name</td>
<td>Set device. name</td>
</tr>
<tr>
<td>AT+RESET</td>
<td>OK</td>
<td>None</td>
<td>Power cycle dev.</td>
</tr>
<tr>
<td>AT+ORGL</td>
<td>OK</td>
<td>None</td>
<td>Reset device to original config.</td>
</tr>
<tr>
<td>AT+RMAAD</td>
<td>OK</td>
<td>None</td>
<td>Delete all auth. devices from pair list</td>
</tr>
<tr>
<td>AT+STATE?</td>
<td>+STATE:&lt;Param&gt;</td>
<td>Initialized, Pairable, Ready, Paired…etc.</td>
<td>Check device status</td>
</tr>
</tbody>
</table>

Optimizing the baud rate:

The HC-05 module supports baud rates up to 1382400, which is about 13 times as much as the default UART rate of the Raspberry Pi (115200) [20]. The limiting factor on the UART speed is
the UART clock as it only allows baud rates of UART_CLK/16. The clock is set to 3 MHz by default, limiting it to 187500baud. To drive 1Mbps, a UART clock of at least 16 MHz is needed. To do this, configuration parameters need to be set in /boot/config.txt by adding the line

```plaintext
init_uart_clock=16000000
```

This allows a maximum baud rate of 16M/8 which allows a theoretical maximum of 2M baud rate. All that is needed to use this optimized rate is changing the corresponding HC-05 (via AT-mode) and Minicom’s baud rates.

```plaintext
AT+UART=921600,0,0  
```

(921600 was picked due to 1382400 being unstable while transmitting)

And for Minicom

```plaintext
sudo minicom -b 921600 -o -D /dev/ttyAMA0
```

### 3.3 Testing Methodology:

Various tests were performed to determine connectivity and power characteristics.

1. All transfers are conducted using Minicom’s zmodem mode.
2. MD5 Hashing is achieved using md5sum utility.

#### 3.3.1 Transfer rate test

- The two Raspberry Pi units are placed at different ranges of each other, with and without direct line of sight.
- The test was done in both default baud rate and optimized baud rate modes.
- A file transfer of 500KB is started with its completion time being recorded.
- The test was repeated with different file types to check their effect on transfer rate.

#### 3.3.2 File size

- Transmission of different file sizes at a 5 meter distance to determine the effect it has on transfer rate.
3.3.3 Data corruption
- Using MD5 hashing methods, the hashes of sent and received files were compared to check for transmission data loss.

3.3.4 Range test
- The two units are placed at varying ranges, with and without direct line of sight.
- The pairing of the units is checked and documented.

3.3.5 Power consumption test
- A multi-meter was used to read the power consumption of the modules, reading the power between their VCC and GND lines.
- The test was conducted while the modules were in Idle and Active states.
- The test was repeated 3 times with the average result documented.

3.4 Example demonstration

3.4.1 Case 1: Simple File transfer
Using Minicom’s zmodem mode, simple files can be transferred from one device to another using the UART serial bus. No additional setup is required after the initial state is achieved.

3.4.2 Case 2: SSH Protocol Control
The command line of a Raspberry Pi can be accessed remotely from other Bluetooth enabled devices on the same network using Secure Shell (SSH) protocol [21]. Only access to the command line will be available, not the full desktop interface but that is more than enough to control the Pi or do routine tasks. To perform this:

1. Use the Raspi-config window (shown in fig. 10) to enable remote login access by going to advanced options > Serial > then select “Yes” at the prompt for the login shell to be enabled.
2. Configure the HC-05 to set it to accept incoming connections. This is done by unbinding it from the other unit. The easiest way to do this is
   \[AT+ORGL\] (resets the module to default settings)
   \[AT+PSWD=6573\] (sets password to 6573)
3. Any Bluetooth enabled device is able to establish a connection using the pair password. For this example an Android phone was used, utilizing the “BlueTerm” [22] application. When establishing the connection, the user will be asked to input the pairing password (6573) afterwards a command line on the screen will prompt the user to enter the raspberry pi’s username and password (username: pi, password: raspberry).

4. Once the connection is established, the user will be able to fully control the device as if a terminal is open. Running codes, editing files, and even shutting down the system are examples of user control.

Figure 22: BlueTerm-Raspberry Pi SSH interface example
CHAPTER FOUR
RESULTS & DISCUSSION

4.1 Results

- All tests were repeated 10 times, with the average result noted after removing the highest and lower values obtained.
- The measurement margin of error for all tests is 0.1%, any results that have difference lower than that are considered to be identical.

4.1.1 Fully designed module tests

4.1.1.1 S11 Reflection coefficient antenna test

The meander antenna simulation and network analyzer test.

Figure 23: 1-1.3GHz S11 VS Frequency using (a) Simulation (b) Network analyzer test
4.1.1.2 **Antenna transmission line impedance test**

Network analyzer results for transmission line impedance

![Graph 1](image)

**Figure 24: 2-3GHz S11 VS Frequency Simulation**

**Figure 25: Antenna TL impedance test**
4.1.2 Transfer rate

Using the default baud rate of 115200 and the optimized rate of 921600 using a 500KB sample

<table>
<thead>
<tr>
<th>Distance (meters)</th>
<th>Transmission time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Default baud rate</td>
</tr>
<tr>
<td>0 (Point-Blank)</td>
<td>578</td>
</tr>
<tr>
<td>1</td>
<td>579</td>
</tr>
<tr>
<td>3</td>
<td>579</td>
</tr>
<tr>
<td>5</td>
<td>577</td>
</tr>
<tr>
<td>10</td>
<td>578</td>
</tr>
<tr>
<td>15</td>
<td>Fail</td>
</tr>
<tr>
<td>20</td>
<td>Fail</td>
</tr>
</tbody>
</table>

4.1.3 File size

The test was done using both the default and optimized baud rates, as shown in fig (15) and (16)
4.1.4 Range

Table 8: Pairing Range test

<table>
<thead>
<tr>
<th>Distance (meters)</th>
<th>Pairing Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Success</td>
</tr>
<tr>
<td>15</td>
<td>Success</td>
</tr>
<tr>
<td>20</td>
<td>Success</td>
</tr>
<tr>
<td>21</td>
<td>Success</td>
</tr>
<tr>
<td>22</td>
<td>Success</td>
</tr>
<tr>
<td>23</td>
<td>Fail</td>
</tr>
</tbody>
</table>
4.1.5 Data Corruption
Using 500KB transmission files, hashed using md5sum

Table 9: Hash test

<table>
<thead>
<tr>
<th>Distance (meters)</th>
<th>Hash Comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Identical</td>
</tr>
<tr>
<td>10</td>
<td>Identical</td>
</tr>
<tr>
<td>11</td>
<td>Different</td>
</tr>
<tr>
<td>12</td>
<td>Different</td>
</tr>
</tbody>
</table>

4.1.6 Power Consumption

Table 10: Power Consumption test

<table>
<thead>
<tr>
<th>State</th>
<th>Power consumed (milliampere)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idle</td>
<td>3</td>
</tr>
<tr>
<td>Active</td>
<td>41</td>
</tr>
<tr>
<td>Transmission</td>
<td>41</td>
</tr>
<tr>
<td>Reception</td>
<td>41</td>
</tr>
</tbody>
</table>

4.1.7 Bluetooth Standard Compliance

Table 11: Bluetooth v2.0 compliance

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Bluetooth v2.0+EDR</th>
<th>HC-05 Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>Up to 10m</td>
<td>10m</td>
</tr>
<tr>
<td>Transmission Rate</td>
<td>Up to 2.1Mbits/s</td>
<td>7Kbits/s</td>
</tr>
</tbody>
</table>
4.2 Discussion

4.2.1 S11 Reflection coefficient antenna test
For an antenna to be deemed acceptable, the S11 parameter should be less than -10dB for its operating frequency. The network analyzer has a limited spectrum going up only to 1.3GHz. A simulation of the 1-1.3GHz was done in order to show a correlation between the simulation results and the actual properties of the antenna. The simulation results for the 1-1.3GHz spectrum was approximately the same as the network analyzer results and therefore the 2.4GHz simulated results can be assumed to be approximately accurate.

4.2.2 Antenna transmission line impedance test
The Network analyzer results showed that the transmission line for the antenna was within a 10% margin of the desired 50 Ohms. This result was maintained for the majority of frequencies tested between 1 to 1.3 GHz spectrum. It would therefore be acceptable to extrapolate impedance of 50 ± 5 ohm for our desired 2.4 – 2.48 GHz spectrum.

4.2.3 Transfer Rate
Files transferred with the range of 0 to 10 meters all arrived successfully, we within acceptable error margins and all files were deemed to be identical. Files transferred at ranges larger than 10 meters where subject to errors outside acceptable margins and didn’t not arrive at distance above 20 meters.

4.2.4 File Size
All sizes recorded a result between in the (0.864 - 0.866) range for the default rate and (6.941 – 6.943) for the optimized rate, which is well within the margin of error.

4.2.5 Range
The maximum pairing range is found to be 22 meters. Pairing range doesn’t necessarily constitute successful transmission at this distance, only that a connection can be established and maintained.
4.2.6 **Data Corruption**

The maximum range for error-less transmission is found to be 10 meters. Going above 10 meters, the files transmissions were completed, however with errors that cannot be corrected using error-detection.

4.2.7 **Power Consumption**

Power consumption is within typical Bluetooth transmission standards. Active, transmission, and reception states all consume the same power.

4.2.8 **Standard compliance**

The module is found to be compliant with Bluetooth v2.0+EDR standard. The effective range of the module (10 meters) puts it into Class 2 as specified by table (2).
CHAPTER FIVE
CONCLUSIONS

5.1 Problem statement and solution
Though the Raspberry Pi computer system is extremely versatile in its uses, its lack of a wireless communication interface severely limits its potential. In order to solve this weakness without the use of too many connectivity ports, a full Bluetooth-serial daughter board was designed and implemented making use of the Raspberry Pi UART interface in order to provide a wireless communication channel.

5.2 Problem implementation
Through the use of several computer aided design programs (Altium, Microwave office, ICCT, etc.), a Bluetooth-UART daughter board was designed and simulated. Other applications where used to configure and test the device in order to implement a successful interface with the Raspberry Pi. An emergency ready-made module was purchased to be used as a contingency plan in case the first approach fails.

5.3 Solution Result
A Bluetooth-UART daughter board was designed and implemented. The daughter board was tested against high frequency specification with the use of numerus simulation techniques, and was deemed to be functional. The completed PCB device was unfortunately unresponsive. It was determined that due to certain fabrication issues, the USB programmer was unable to maintain a connection to the CSR1012 IC. An effective solution to this problem was not reached due to insufficient time for a comprehensive debugging to be implemented. Hence, the ready-made module was used to achieve the desired project results.

5.4 Project Outcome
Wireless Connectivity between two Raspberry Pi models was achieved using the daughter board module. The daughter board module was used to connect/provide new uses for the Raspberry Pi units.
5.5 Project Limitations

The module used is only compliant with Bluetooth v2.0+EDR standard, which has a lower range and transfer rate than newer Bluetooth standards. The module itself comes in a breakout board, which makes integrating it as an actual daughter board that can be connected to the Raspberry Pi impossible. Instead, the use of Jumper cables was required.

5.6 Personal Gain and Future Work

During the process of completing this project, I was able to accumulate many new skills as well as a solid understanding of High frequency electronics and circuit design. I became proficient in the use of both Altium Designer and Microwave Office CAD programs. Gain an effective knowledge of the PCB manufacturing process as well as general knowledge of the Raspberry Pi computer system.

Future versions or this daughterboard may include:

- Use the Full-Design route instead of Ready-made modules.
- More applications for the HC-05 interface with a Raspberry Pi such as using the Pi as a way to make “dumb” TV’s smart, or using BlueTerm with the HC-05 to provide headless (without a monitor) Raspberry Pi’s with a useful output interface.
- Implementing another Wireless interface with the Raspberry Pi, possibly Zigbee or Wi-Fi.
- Explore using the HC-05 with other devices, such as providing Bluetooth connectivity to desktop PCs which generally lack that option.
REFERENCES


   https://er.educause.edu/articles/2016/6/the-internet-of-things-is-here


   https://www.bluetooth.com/specifications/


   https://learn.sparkfun.com/tutorials/bluetooth-basics


https://www.csrsupport.com/download/47278/CSR1012 Data Sheet CS-238833-DS.pdf


http://www.hobbytronics.co.uk/datasheets/EGBT-bluetooth-datasheet.pdf


https://moduleforces.blogspot.com/2017/06/hc-05-datasheet-and-how-to-enter-at.html


http://fw.hardijzer.nl/?p=138

https://www.raspberrypi.org/documentation/remote-access/ssh/
