TAMPEREEN TEKNILLINEN YLIOPISTO

DANISH SAADI

DESIGN AND DEVELOPMENT OF A ROBUST, RELIABLE REMOTE INPUT DEVICE FOR VIRTUAL REALITY ENVIRONMENTS

Master of Science Thesis

Examiner: Prof. Timo D. Hämäläinen
Instructor: Dipl. Wolfgang Schotte
Examiner and topic approved by the Faculty Council of the Faculty of Pervasive Computing on 20 July' 2015

ABSTRACT

Tampere University of Technology
Master's Degree Programme in Information Technology
DANISH SAADI: Design and Development of a Robust, Reliable Remote Input Device for Virtual Reality Environments
Master of Science Thesis, 63 pages
March 2016
Major: Digital and Computer Electronics
Examiner: Professor Timo D. Hämäläinen
Keywords: Wireless Communication, RF-module, Atmega88 and ATtiny88 micro-controller, Virtual Reality, Reliability Enhancement, ISM band, Bi-directional Protocol.

Visualization is the most important and helpful part of any scientific research activity. Almost all of the research and development requires specific and in some cases real-time visualization of the simulation. HLRS visualization department support scientists and engineers with visual analysis of the simulated data computed by high performance computers. With time the complexity and size of the simulated data set is increasing and in order to keep up with the technology, new and state of the art techniques are being utilized for data visualization.

Currently the visualization department is working on a 3D virtual reality environment called CAVE which allows to simulate data sets intuitively. This virtual reality environment need some kind of control and navigation capability form humans and for this purpose a prototype of a remote device is being developed. This remote device is used to control and navigate scenes and simulation models within the virtual reality environment of CAVE. The first prototype of this remote device consists of AVR atmega micro-controller and a RF module operating at 2.4 GHz. The same set of devices is present on the receiver side as an AVR controller and a RF module which is then connected to the CAVE via USB.

The main objective of this thesis work is to implement a reliable, robust and low-latency bi-directional communication protocol between multiple remote devices and a receiver device inside CAVE. Also improve the power efficiency of the remote device regarding run-time and stand-by time, redesign the remote device to support the target application with feedback to the user.

In order to improve the reliability first the reliability is defined regarding our environment requirements, then various key factors which reduce the reliability are studied and their solutions are implemented and finally the implementation is evaluated. The latency will remain a key point in all of the implementations regarding robustness and reliability of wireless link. The communication protocol will be extended to handle multiple remote devices with bi-directional capability with one receiver. The goal of making the remote device maximum power efficient is investigated by exploiting the different operating modes of the RF module during the run-time and stand-by time of the remote device. Latency will remain a trade-off with power efficiency and reliability.

PREFACE

I would like to take this moment and show my regards to HLRS hight performance computing research institute in Stuttgart, which allowed me to perform my master thesis project in their respective institute. Specially the Visualization department and the visualization team provided the enabling environment in which this master thesis was successfully accomplished. Mr. Uwe Woessner and Mr. Wolfgang Schotte helped me personally during my project and guided me through their discussion on various topics. I would also like to mention that the laboratory in which I worked for these six months provided a perfect environment for research oriented activity.

Tampere University of Technology always remained helpful throughout the process, from registration of the thesis topic and supervision related issues. TUT provided all the necessary documentation which allowed me to start my master thesis at HLRS. Prof. Timo D. Hämäläinen supervised my master thesis and guided me in the matter related to the project. He also advised in the process of writing this master thesis and helped to improve the organization of this document.

Finally I want to pay my regards to my family who always supported me in difficult and good times. They always kept me motivated and focused me on the real goals to be achieve in this life. Specially my mother who kept me in her prayers throughout my masters education. My father who always motivated and supported me in pursuing a professional career abroad. My brother and sisters whom I missed a lot during this period and was not able to share some of the beautiful moments of their life. I also wanted to apologize to my brother in law and nephew whose birth took place while I was abroad. I hope that all of this sacrifice may really worth something in the end.

30.3.2016

Danish Saadi

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACK	Acknowledgement
AP	Access Point
ASN	Absolute Sequence Number
BER	Bit Error rate
CAVE	Virtual Reality Environment Room
CE	Chip Enable
CRC	Cyclic Redundancy Check
CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance
DSSS	Direct Sequence Spread Spectrum
EB	Enhanced Beacon Message
EN_AA	Enable Auto Acknowledgement
FHSS	Frequency Hopping Spread Spectrum
FIFO	First In First Out
GFSK	Gaussian Frequency Shift Keying
GHz	Giga Hertz
ISM	Industrial Scientific and Medical radio band
ISR	Interrupt Service Routine
Kbps	Kilo Bit Per Second
MISO	Master in Slave out
MOSI	Mater out Slave in
No_ACK	No Acknowledgement
OSI	Open System Interconnection model
PDR	Packet Delivery Ratio
PER	Packet Error Rate
PIC	Family of Microcontrollers
RFM	Radio Frequency Module
RSSI	Relative Signal Strength
RX	Receiver Mode
SCK	Synchronous Clock Microcontroller pin
SNR	Signal to Noise Ratio
SPI	Serial Peripheral Interface

SS	Select Signal
Tx	Transmission Mode
USART	Universal Synchronous Asynchronous Receiver Transmitter
USB	Universal Serial Bus
Wlan	Wireless Local Area Network

1 INTRODUCTION

This chapter will provide the basic understanding of this master thesis work. First the brief explanation about embedded wireless communication systems is provided. Since this thesis work is part of an on going project, a brief introduction of the previous work and argument for the decisions taken in the past within the scope of this project is necessary. Then the problem description is detailed with specific objectives to be achieved and the reasons why those objective are relevant within the project environment is discussed. In the last section the organization of this master thesis document and how the various chapters are connected is explained.

1.1 Embedded Wireless Applications

Embedded system is a special purpose small computer which is designed for the needs to control electronic devices and in which it can be easily concealed. It can only perform pre-defined tasks and can be optimized with regards to the specific application in terms of size and cost [1]. In brief, an embedded system is any computer that is a component of a larger system and that relies on its own microprocessor [2].

Connectivity is the key for modernization and up to an estimate 3 billion embedded devices were manufactured in 2008 and two-third of them were connected. Embedded devices cannot operate as a stand alone system any longer and wireless connectivity has provided an opportunity to communicate in a cost effective way. Short range and long range wireless communication protocols have already seen an increase in application for embedded systems and as suggested in [3] the trend of embedded wireless application is in the area of System-on-Chip architecture. Which means an embedded microcontroller can be integrated with a RFM module on a single chip. A very sound evidence of similar implementation was mentioned in [4] when a short range and low rate wireless technology ZigBee module was successfully interfaced with PIC microcontroller for wireless remote control of industrial electrical system parameters like voltage and current. The number of wireless connected devices is increasing as never before and it is predicted that it might go upto 40 billion [5]. The basic reason for this increase is IoT which will offer an overall connectivity from home appliances to personal gadgets and beyond, it refers to the interconnection of heterogeneous smart embedded devices within the internet infrastructure [6]. The most important point to

remember is that the era of one person per computer is over and the time of many computers per person has begun [7].

1.2 Prior Work

This master thesis work was carried out as a part of larger project CAVE. CAVE is a 3D virtual reality environment which is used for the simulation of large scientific data sets for the research purpose. It provides an accurate 3D simulation using a special mouse or 3D glasses creating an accurate rendering for the human eye and offer a resolution of 1920 by 1200 pixels [8].

In order to further enhance the capability of the humans to navigate within this 3D virtual reality environment and exercise control, a remote device is developed. This remote device is callled 'CyberStick' and uses a short range wireless communication module RFM73. The decision to opt for wireless communication was based on the requirements of the working environment inside CAVE. The wired communication cannot provide the mobility required to navigate in the CAVE. Wired communication was considered to be more reliable, faster, secure than wireless communication, but recently wireless networks has also provided reasonably high reliability, easy installation and cost effectiveness [9]. Security is not a concern in our scenario and data speed in terms of milliseconds is adequate. The most important factor was the mobility so that the user can move freely within the environment without worrying about the wires.

There are several other technology standards which operate in 2.4 GHz ISM free band such as Bluetooth LE, ZigBee, Infrared but RFM73 Transceiver was selected. Bluetooth LE have almost similar characteristics as the RFM73 like range is similar 10 meters, 79 operating channel with 1MHz width, 1mW of TX power for 10m range but the transmission bit rate is 750 Kbps to 305 kbps [11][12] which is lower than RFM transmission bit rate of 1Mbps to 2Mbps [13]. Since CAVE is an interactive environment the throughput of the data transmission is critical. Zigbee was also a strong choice for this project with lower power consumption of 0.5mW less than that of RFM73 2mW but again the max transmission bit rate is 250Kbps [12] which is considerably lower than that of RFM73. Infrared technology was disqualified based on its range.

1.3 Problem Description

The prototype of the remote input device which is developed to be used within the virtual reality environment of CAVE have lots of potential for improvement. The main objective of this thesis work is to improve the working of the CyberStick within the

aspect of wireless communication, latency, power efficiency and use of multiple CyberSticks.

ISM is a license free industrial, scientific and medical 2.4 GHz frequency band, this feature of the ISM band promoted the emergence of various technologies within 2.4 GHz range and thus made it vulnerable in terms of reliability of the short range wireless communication [14]. As our remote input device also operates within the 2.4 GHz ISM band the reliability will be a concern due to the presence of other technologies within CAVE such as Bluetooth, Wlan etc. In order to provide the perception of reality within virtual reality environment stability plays a very important role, which means the display lag between end-to-end. Degradation of the stability in virtual reality environment can cause Oscillopsia in which visual world appears to swim or oscillate in space [15]. The impact of latency on humans was studied in which response to two different environments were compared, one was normal and non-threatening and the other was to evoke a fear response [16]. Than participants were subject to low and high latency and it was concluded that participants subjected to low latency have higher sense of presence of the virtual environment [16]. Since the remote input device works interactively within the CAVE environment, it is highly desirable to keep the latency as low as possible to provide a high sense of reality.

The CyberStick is a stand alone remote input device and is not connected to any power source at all times. Rather it has the storage for two AAA cell batteries and cannot operate forever without replacing those AAA cell batteries. Due to this aforementioned problem it is necessary to design and implement a better power management solution. The virtual reality environment of CAVE is not a single user environment and since it is still in development there are lots of demos continuously carried out throughout the year. Also after the completion and when it provided in the market the basic purpose of this virtual reality environment is to provide simulation for large data sets to the scientist and researchers, and often it will be required for simultaneous use by multiple users. For this purpose the remote input device CyberStick should be able to communicate with CAVE with multiple of its clones. Another requirement for this interactive communication is to have a feedback from CAVE to the CyberStick in order to provide high sense of reality and reliability.

2 CYBERSTICK

Figure 2.1 CyberStick Remote Input Device

This chapter provides the detailed explanation of the CyberStick electronics and its communication network. First the most important discussion about embedded radio frequency communication device is presented. This discussion will allow the reader to get in-depth understanding of the RFM functionalities crucial to understand the complex ideas developed in this thesis. Mainly the different RFM modes, its message transmission types and the concept of data pipes are explained. In the second part another embedded device the microcontrollers used within the CyberStick and within its counterpart the receiver device are introduced. Since these microcontrollers have vast area of applications and enormous functionalities, the discussion will be limited to the used microcontroller peripherals. In the third portion of this chapter the common attribute in microcontrollers and in the RFM, the power down functionalities of both devices is introduced. This thesis also involves the implementation of ideas developed during the research and in order to implement, the explanation about the development and communication environment for this thesis project is necessary. The last part of this chapter is dedicated to the direction of research and development based on the OSI reference model.

2.1 Radio Frequency Module

Figure 2.2 RFM73 ChipSet [13]

As discussed in the previous chapter the radio frequency module RFM73 was selected for the short range wireless communication in CyberStick. RFM73 is a GFSK transceiver. The term transceiver refers to a device which has the same circuitry for transmitter and receiver. The RFM73 operates in the ISM frequency band within the frequency range of 2400 MHz to 2483 MHz. It can be tuned to 1 to 83 different frequency values with the band limit of 1 MHz by setting the value in the RF_CH register. The receiver should also be tuned with the same frequency so as to communicate with the transmitter. The formula for setting the frequency in RF_CH register is,

$$\text{Frequency} = 2400 + \text{RF\_CH (MHz)}, \qquad (2.1)$$

The RFM73 also has the capacity to transmit packets with different output power and also can vary its bit transmission rate from 250Kbps to 2Mbps [13]. The RFM73 can be programmed to switch between transmitter device and the receiver device. It uses a burst mode (to transmit same instance of data several times by overlooking the necessary steps before each packet transmission) transmission to increase the data throughput and makes it easier to listen on the receiver side. The RFM has two FIFOs, one is called RX FIFO and the other is called TX FIFO. The purpose of both FIFOs is simple, is to store data after receiving or before transmitting. The storage capacity of both FIFOs is three slots each of 32 bytes. It also contains a packet processing unit which automatically takes the data out from the packet when the RFM is configured as receiver and bundle up the data as required by the designed specifications when configured as a transmitter. The packet processing unit also perform the cyclic redundancy check on the received packet and after successful result it transfers the payload to the RX FIFO. The structure of the packet is provided in ***Figure 2.3*** for better

explanation. The data length of the payload is variable from 1 to 32 bytes and can be configured by setting up the appropriate register value.

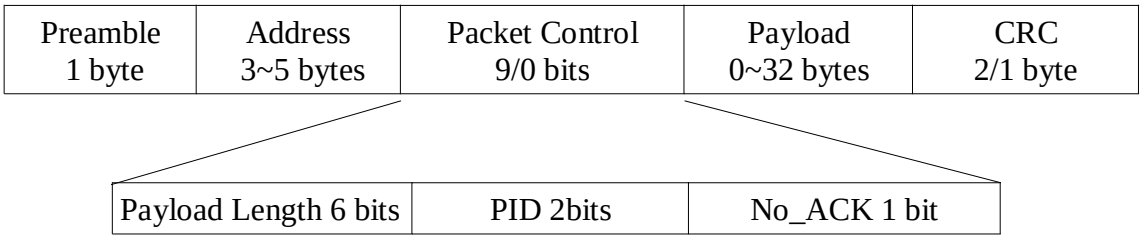

Figure 2.3 RFM Transmission Packet Format [13]

In order to enable multiple transmitters to communicate with the single receiver the RFM73 provides the concept of data pipes upto 6 which is explained later in this chapter. The RFM73 have very low power consumption and due to all the attributes just mentioned it is an appropriate choice for our application.

2.1.1 Special Features

The RFM have some hardware built in functionalities, some are part of normal operation and others can be enabled by setting up the values in specified registers. The knowledge of these features or functionalities helps in developing the broad understanding of RFM. These feature also enhance the reliability of the wireless link between transmitter and the receiver. The explanation of some of the important features are given below:

- **Auto Retransmission**

This feature provides the user with the possibility of transmitting same payload multiple times and used in Transmitter device. This feature has to be enabled and the maximum retransmission value have to be set in the specific register. The purpose of this feature is save time in case packet is not received by the receiver and has to be transmitted again. Auto retransmission works with another feature called Auto Acknowledgement.

- **Auto Acknowledgement**

Auto acknowledgement is used when RFM is configured as a Transmitter device and allows the receiver to send acknowledgement payload everytime it receives a packet from transmitter. If auto acknowledgement is not received the transmitter will keep transmitting the original packet until the number of retransmission reaches the user defined maximum value. The maximum number of retransmission can be set from 0 to 15.

- **Auto Retransmission Delay**

Auto retransmission delay specifies the time between the retransmission of the same packet. This delay have to be set with care since we need to wait for atleast the amount of time necessary to receive the auto acknowledgement. If auto acknowledgement in not received and auto retransmission delay is elapsed than the transmitter can send the same packet again until it reaches the maximum retransmission value.

- **Auto Retransmission Count**

This feature allows to set the maximum number of retransmissions possible in case auto acknowledgement is not received. This also allows the programmer to monitor the number of retransmission and any decision taken based on intermediate number of retransmissions.

- **TX_DS Interrupt**

TX_DS interrupt provides the programmer the opportunity to increase the reliability. Normal TX_DS is asserted when the data is sent from the transmitter, but if auto acknowledgement is enabled this interrupt waits till the acknowledgement packet is received from the receiver and then it is asserted. This feature gives the opportunity and decision making power whether the wireless channel is reliable or not.

2.1.2 TX and RX Modes

The RFM73 is a transceiver and can perform both roles of a transmitter and a receiver. In order to provide the control of RFM73 to the user, the RFM73 can be programmed to operate as the transmitter device and the receiver device by using the PRIM_RX bit.

For Transmitter device: PRIM_RX = 0,
For Receiver device: PRIM_RX = 1,

Once the device type is selected it does not mean that it can only operate as Transmitter or Receiver rather it also has the capacity to operate interchangeably between RX and TX modes without any interference from the user as a part of some built-in functionalities, but the device type will still remain the same even when it is operating differently. There are three more modes which are used as intermediate states when going into TX mode or RX mode. These are Power Down mode (explained in detail later in this chapter), Standby-I mode and Standby-II mode. The importance and functionality of these modes can be explained with the help of state diagrams.
It is shown in ***Figure 2-4*** and ***Figure 2-5*** that the default mode is Power Down mode and the next immediate mode has to be Standby-I mode. From where RFM73 can go into TX mode or RX mode depending on the value of PRIM_RX bit. Standby-I mode is

used to reduce the power consumption of the RFM and also minimize the time to go into TX mode or RX mode.

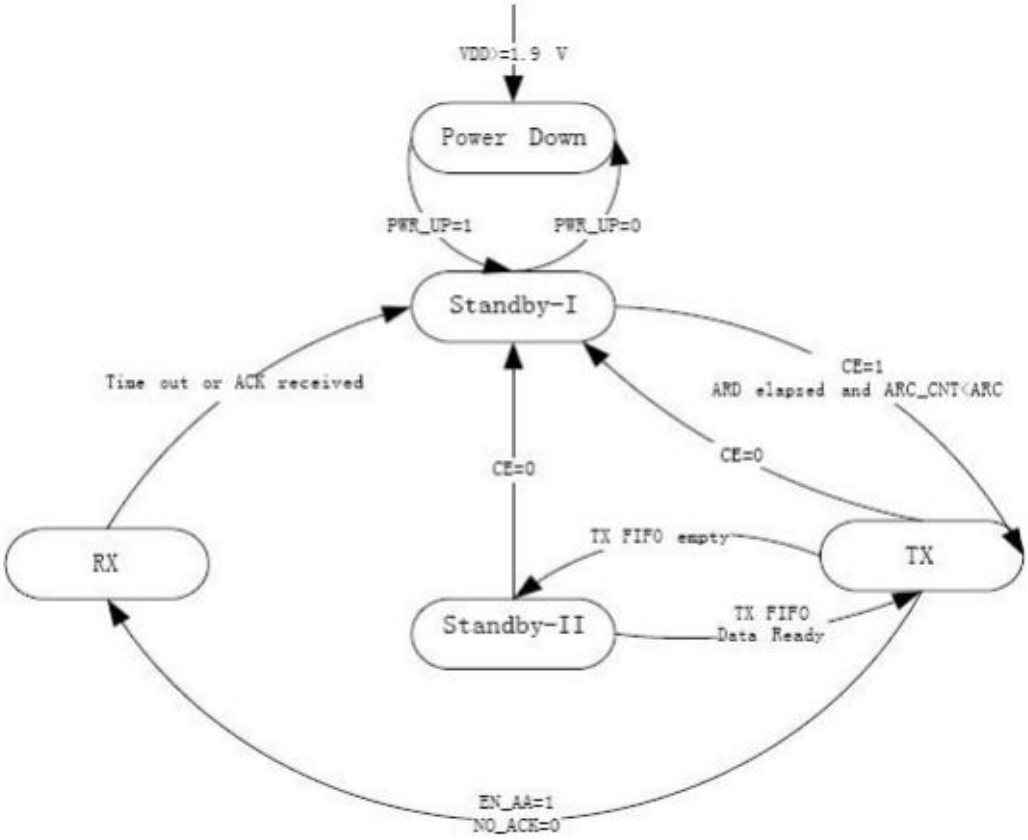

Figure 2.4 RFM73 State Diagram for PTX mode when PRIM_RX=0 [13]

Lets first take the state diagram in ***Figure 2.4***, which explains the operation of RFM for Transmitter device. RFM can switch to TX mode from Standby-I mode if CE bit is set high and TX FIFO have payload to transmit. RFM cannot remain in TX mode for more than 4ms and if TX FIFO is empty then it will switch to standby-II mode. From where it can come back to TX mode if TX FIFO gets new payload to transmit. From Standby-II mode and TX mode, if CE is set to 0 the RFM can go back to Standby-I mode at anytime as well. There is another possibility that when RFM is configured as transmitter device can switch to receiver mode after transmitting payload if auto retransmit is set (EN_AA) and acknowledgement is required (NO_ACK).

When RFM = Transmitter Device,
TX mode → RX mode : if EN_AA =1 and NO_ACK =0,

From the RX mode it will go back to Standby-I mode if the ack is received or the time is out which was set to wait for the acknowledgment. In case the acknowledgment was

not received and time was out than from Standby-I mode the RFM will go to TX mode and retransmit the packet, if the value of the auto retransmission counter is less than the value set for auto retransmission count and the auto retransmission delay had elapsed. When value of auto retransmission counter reaches the value set for maximum auto retransmission, the RFM will send the new packet from TX FIFO in TX mode.

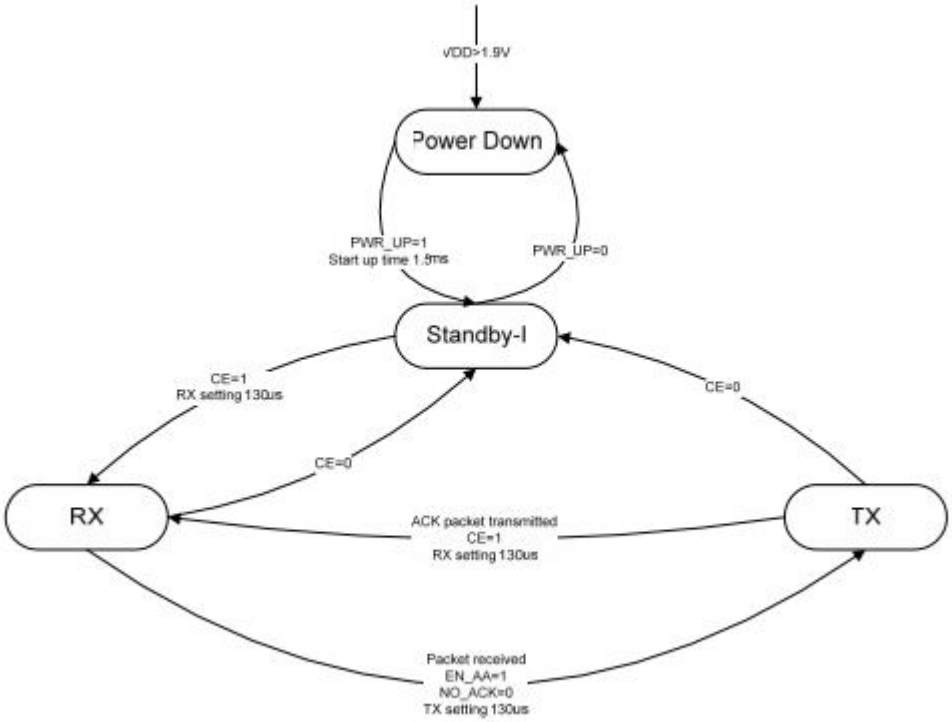

Figure 2.5 RFM73 State Diagram for PRX mode when PRIM_RX=1 [13]

To understand the RFM operation as a Receiver device, consider state diagram in ***Figure 2.5***. From Standby-I mode the RFM can switch to RX mode if CE is assigned high(1) and if the received packet contains auto acknowledgement(No_Ack = 0) than the RFM goes into TX mode and sends the acknowledgement packet. From where it can either go back to Standby-I mode or again to RX mode depending on the value of CE.

2.1.3 Transmitted Payloads

The RFM73 allows three different types of payload transmission. These payload types are related to the TX or RX mode of the RFM. Two types are allowed in the TX mode and one in the RX mode. The packet is transmitted only in the TX mode but in order to use the specific payload type the current RFM state is important, the RFM goes into the TX mode as part of built-in functionality to transmit if we use the payload transmission type only allowed in RX mode. The in depth understanding of these three different types of payload is necessary in order to understand the reliable bi-directional communication mechanism which was developed in this project.

- **Auto Acknowledgement Required**

This type of payload transmission is used in TX mode and it allows the receiver to send the acknowledgement packet. When receiver receives the packet and its packet processing unit interpret the different fields of packet, from where it understands that an acknowledgement packet has to be sent. The transmitter waits for the acknowledgement packet for the user defined period of time and retransmit the packet if does not receive the acknowledgement packet. This type of messaging allows the user to increase the reliability of wireless communication or atleast figure out whether the wireless link is reliable or not.

- **Auto Acknowledgement Not required**

Another type payload transmission which can considered as normal messaging is when acknowledgement is not required. The receiver will not send an acknowledgement packet when it receives this type of packet. With this type of messaging the reliability of the wireless link cannot be established. This type is also used when RFM is in TX mode.

- **With Acknowledgement Packet**

This is a unique type of packet transmission because it is only allowed when RFM is in RX mode. The logic of this type of packet transmission is that if the receiver receives a packet of which acknowledgement has to be sent, the packet processing unit of the receiver side will bundle up the data payload stored in the TX FIFO with the acknowledgement packet to be sent. The receiver will go into TX mode for small period of time to sent the acknowledgement packet and uses this opportunity to send the user defined payload with the acknowledgement packet. The pre condition to use this type of packet transmission is that the packet received from the transmitter has to be of Auto Acknowledgement Required type and the TX FIFO of the receiver already have the payload to be transmitted. This type of packet transmission ensures the reliability of the wireless link as well as reduce the latency of message transmission from the receiver side.

2.1.4 Data Pipes

Wireless communication network often designed to communicate among more than two devices rather than one-to-one communication between a transmitter and a receiver. The RFM also allows the user to communicate between several of its clones. To enable this functionality and to identify messages from various different of its clones, RFM uses the concept of Data Pipes.

RFM allows the 1:6 star network in which one RFM module is configured as Receiver device and other six of its clones are configured as Transmitter devices. The data pipe concept allows the receiver to differentiate among the messages received from these transmitters so that the receiver can understand which message is from which transmitter. The RFM can be programmed with one of six unique addresses if it is a transmitter device. The receiver can be programmed with all of the six addresses and does not have to switch among the addresses so as to listen to the specific transmitter. Transmitter should communicate at a specific address and the receiver should have that same address saved to listen and both to be configured with the same operating frequency. The most important point is that even if all the six transmitters are communicating at the same frequency the receiver will listen to all of the addresses which it is configured to listen and can differentiate among those messages as well. All six data pipes can be considered as different communication channels and are allowed to have all the attributes such as various type of packet transmissions, different payload lengths, operating at different frequencies etc.

2.2 AVR Microcontrollers

The RFM has to be controlled at real time to perform wireless communication and has to be interfaced with a programmable controller. For this purpose two Atmel microcontrollers have been interfaced with RFM modules, one for the transmitter side (CyberStick) and one for the receiver side. The CyberStick contains Attiny88 and the receiver side have Atmega88. Both microcontrollers are 8-bit, low power and high performance programmable controllers. RFM requires an SPI interface to be controlled or programmed and both Atmega88 and Attiny88 have SPI interfaces. Some of the more distinguished features of Atmega88 and Attiny88 which are also used in this project are given below:

2.2.1 Timers/Counters

The Attiny88 and Atmega88 both have 8-bit and 16-bit timers/counters. These timers allow the programmer to make decisions at runtime based on the specific time or count. Microcontrollers require clocks to run and same clocks are used to control the timers. The basic operation of the timer is that it start counting as soon as it is initialized and when reaches a specific value it generates an interrupt at which point microcontroller stops the processing and jump to the ISR which is assigned to that timer. After the execution of the ISR the microcontroller will restart the processing where it had left at time of the interrupt generation. The clock allows the timer to count and in order to slow the counting of the timer, the clock can be prescaled by use of the prescaler provided in the microcontroller. The timer can be configured to generate interrupt at a specific value or when it overflows.

2.2.2 Pin Change Interrupt

Pin change interrupt proved to be very crucial in the design of CyberStick and allowed to make decisions when a button is pressed on the CyberStick. Pin change interrupt allows the microcontroller to sense any high or low on the specific pin, which is configured for pin change interrupt and execute its ISR which can be programmed by the user. This interrupt can also be use to power up the microcontroller when it is in the sleep mode or power down mode.

2.3 Power Down Modes

CyberStick is a stand alone device as mentioned earlier and requires two AAA batteries to run. To allow the device to run for a longer period of time without changing of batteries, the efficient use of the available power is necessary. CyberStick requires some kind of efficient power management system but as the receiver side is continuously connected to the CAVE it does not require any power management. CyberStick contain two devices the RFM and Attiny88 and both posses a power down state. When the CyberStick is not in use these power down states can be utilized to reduce the power consumption of the device. This changing into power down states can have an impact on the latency of the device, but this will be discuss later in this thesis. Pin change interrupt is used in the implementation of a power down functionality on the microcontroller, which also force the RFM to go into power down mode as well.

2.4 Communication Environment

The CyberStick and the receiver device both perform wireless communication using their respective RFMs which are cotrolled by microcontrollers in real-time. The communication can be in any direction from CyberStick to receiver device and back forth depending on the designed application. The data can only be generated from CyberStick as it is considered the remote device and the receiver will receive the data and will forward it to the COVISE, the system software running on the CAVE. The only possibility of the opposite direction flow will be in case their has to be a response from the CAVE in return for the received data. This point will be elaborated in detail in the chapter of Future Work. Different protocols and devices are part of this communication environment which are detailed below:

2.4.1 Development Environment

The first step towards carrying out a project is set up a development environment. Since this thesis project is based on the implementation and verification of the ideas and techniques developed during the last six month period, the user friendly development environment played a crucial role in the successful completion this project and saved a

lot of time. Most programmers have their favourite programming editors and tool chains, and some devices have some specific tools and softwares. Since we were using Atmel Atmega88 and Atiny88 which does not have an specific programming editor but GNU toolchain provides a plug-in for Eclipse editor. This GNU plug-in and AVR DUDE programmer was configured for the Eclipse editor and the programming of microcontrollers were close to one click after successful compilation. This GNU toolchain also contain the compiler and the AVR DUDE had the programmer software for the microcontrollers. The application development was carried out in C language.

2.4.2 SPI Interface

SPI uses the concept of Master and Slave to communicate. There will be one Master but several Slaves are possible. SPI is used for two purposes in our communication environment, one for the control of the RFM from microcontroller and other is to program the microcontroller using programmer device. In this project microcontroller behave as the master and the RFM behave as slave. SPI is being used on both the transmitter device and the receiver device for communicating with RFM and programming of Atmega88 and Attiny88. SPI requires four pins to communicate with a device the SCK, MOSI, MISO and the SS. These pins are also available in Atmega88 and Attiny88 as shown in ***Figure 2.6*** in black coloured pins.

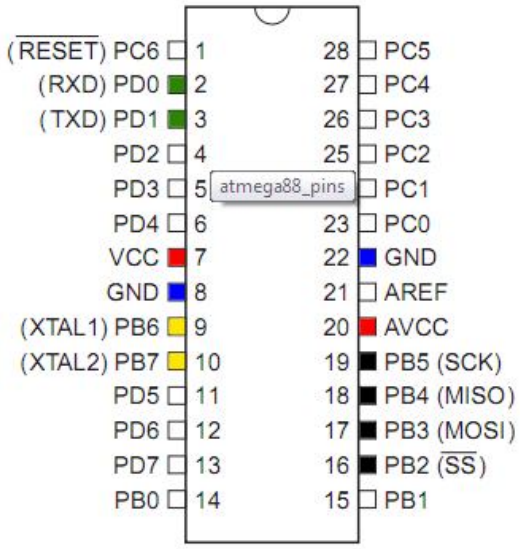

Figure 2.6 Atmega88 pinout courtesy to codeendlife.com

2.4.3 USART Interface

USART is another serial communication interface provided on the microcontroller Attiny88. As SPI is already being used for communicating with RFM, the USART interface was utilized for the purpose of debugging messages. It is full duplex communication with synchronous or asynchronous modes. Two registers are dedicated

in the microcontroller for communication using USART one is the Rx register and the other is the Tx register. The serial communication using USART requires the use of three pins on the microcontroller Rx, Tx and XCK pins as shown in ***Figure 2.6*** in green coloured pins. This interface is only used on the receiver side because debugging messages can be transmitted wirelessly from transmitter to the receiver side and then publish on the terminal.

2.4.4 Data Flow

The data is generated from CyberStick and then wirelessly communicated to the receiver device which is connected to the COVISE. CyberStick consists of three buttons and one touch and motion sensor. The user can navigate within the COVISE environment by using these three buttons and one sensor. The pressure and motion sensor allows the user to scroll up-down-left-right. The microcontroller continuously check the states of these input sources and if the state is changed it sense an input data has arrived. Once the input data is sensed it stores the states of the input sources into 32 byte long variable and send it to the RFM using SPI interface. The microcontroller on the CyberStick also handle the modes of the RFM which were discussed earlier in this chapter and depending on the type of transmission also configured by the microcontroller, the RFM will start transmitting the data. The RFM is designed so as soon as it receives the command and the data, its packet processing unit will start transmission on the specified frequency. The microcontroller is programmed by using the Eclipse environment as stated earlier and allows the programmer to define the functionality of the microcontroller and the RFM.

Once the data is received by the receiver RFM, the microcontroller on the receiver side takes the data and transfer it to the COVISE using USB interface. A USB interface is integrated into the receiver side to allow the receiver device to communicate with the COVISE. A serial communication interface using USART is also integrated to the receiver device to communicate the debugging messages during development of the prototype. The RFM gives an interrupt on arrival of a new packet and the microcontroller on the receiver side is configured to listen to that interrupt at all times. As the microcontroller listens to the interrupt it sends a command to read data from the RFM and RFM send the data using SPI interface to the microcontroller.

2.5 OSI Layers and Research Areas

The OSI is a reference model for how applications can communicate over a network [17]. It helps in understanding the connections among different stages of a network and the operations carried out within each stage. The main idea behind developing this

model was to specify a universal model for communication network so that various products developed by different vendors can interoperate.

For the ease of understanding the ideas and techniques developed and implemented during this project can be divided into the stages of OSI reference model. The communication scenario for our project is concerned with last two layers the physical layer and the data link layer. The partition of the communication network used in this project into OSI layers is given in **Figure.**

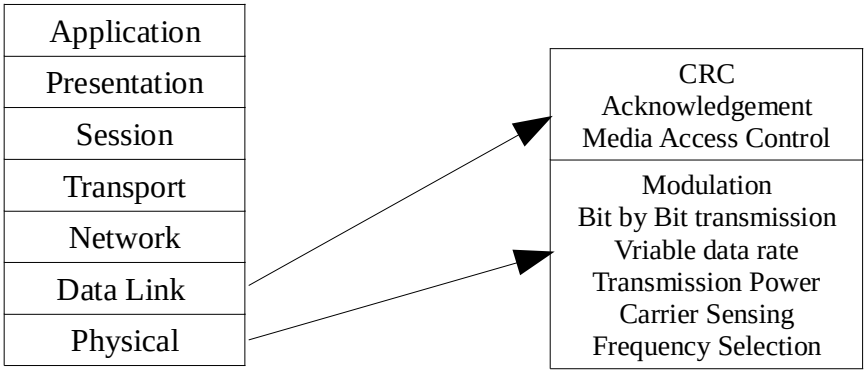

Figure 2.7 OSI Layers and Concerned Topics

2.5.1 Physical Layer

Physical layer is the first layer of the OSI model and it is concerned with hardware used for communication. This layer encompass all the issues from receiving data from the data link layer to the transmission. The main areas of interest for this layer is signal modulation so that the data can be transmitted in binary, bit synchronization for synchronous serial communication, carrier sensing and collision detection, forward error correction/channel coding and frequency selection for wireless communication [18].

The implementation of physical layer in this project scenario is on the RFM transceiver, as the RFM handles the modulation, bit-by-bit transmission, variable data rates, transmission power, carrier sensing, SNR of the transmitted signal, maximum range of the transmission and frequency selection. Some of the functionalities implemented on the RFM are related to other layers which will be mentioned later in this chapter. All these physical layer attributes implemented on RFM can be used through the control software on the microcontroller and can be changed at runtime as well. Since the

hardware implementation on the RFM cannot be changed there is no room for research and modification at the physical layer.

2.5.2 Data Link Layer

The data link layer controls the exchange of data between nodes on the same network and allow the upper layers to access the physical medium. It controls how data is placed on the medium and received from the medium using techniques such as media access control and error detection and correction for errors occur at the physical layer [19]. The data link layer is further divided into two sub layers Logical Link Control and Media Access Control. The logical link control sub layer manages flow control and error control over the physical medium of the data link layer. It also assign the sequence numbers to frames and track acknowledgements [20]. Media access control sub layer defines the decision making process to allow which node can access the medium at a particular time in the network, since if two nodes on the same network send messages at the same time over the physical medium will cause collision. It can also provide encapsulation of upper layer packet into frames and frame synchronization.

The implementation of the data link layer is divided into RFM and microcontroller. The CRC error detection is implemented on the RFM as part of its hardware implementation and it is within the domain of data link layer. The packet addressing and packet formatting is also handled in the RFM and is part of data link layer. The most important part of this layer is the media access control which has to be implemented as part of the C functionality in the microcontroller to stop interference in the wireless communication. Another important technique to improve the reliability of the wireless link is the acknoweldegement, can be implemented as C functionality on the microcontroller or already provided hardware implementation of the auto acknowledgement on the RFM can be utilized. Since there are some functionalities have to be implemented through software on the microcontroller and they fall under the scope of data link layer, thus it is clear that the data link layer will be the layer in which most of the research and implementation will be carried out.

3 STATE OF THE ART

This chapter will focus in detail on the different issues or problems identified during the design and development of the remote input device. These problems has been categorized into three areas. The reliability of the wireless communication, the impact of Latency when CyberStick communicate with the receiver device and use of multiple remote input devices at the same time.

In first part of this chapter reliability is discussed in detail. Issues in ISM 2.4 GHz band, effect of interference in 2.4 GHz wireless communication, transmission power and finally various solution are presented based on research of already existing 2.4 GHz band standards and techniques. Then the impact of latency will be discussed based on the solutions presented in previous section for increasing the reliability of wireless communication and different solutions will be compared for the trade-off with increase in latency. Finally the complications which will arise due to the use of multiple CyberSticks are mentioned and the possible solutions will be presented.

3.1 Reliability In Wireless Communication

Reliability in wireless communication is defined as per the requirement of the application, for some cases reliability can be defined as data integrity which means data being transferred from transmitter to the receiver with 100% accuracy and for other cases such as video and audio it is all about tolerance as to delay and corrupted data. Reliability of a communication channel can also be defined as the minimum error probability at maximum data rate at which reliable communication is possible [21].

Reliable wireless communication in our project scenario can be characterized at two different layers of the OSI model. As discussed in previous chapter the concerned OSI layers with respect to the development of our remote input device are Physical layer and the Data link layer. Reliability at physical layer can be characterized using certain measurements such as SINR, BER, SER, PER, RSSI and outage probability. The data link layer improves the reliability by removing the corrupt packets received from the physical layer and the measure for such an activity at this layer is PDR

One of the main objective of this thesis project is to achieve a reliable communication between the remote device and the receiver. This chapter is dedicated for the discussion

on how to increase the reliability and efficiency of the remote input device and for this purpose first we need to identify the causes which decreases the reliability in a short range wireless communication environment.

3.1.1 Interference in ISM Frequency Band

ISM frequency band is free to utilize for various short range wireless communication devices. Mostly within the range from 10m to 100m, which include a Wlan device for the whole building or smart sensor networks such as fire alarms, remote devices, Bluetooth in hand held electronic products etc. With the technological advancement more and more embedded wireless devices are introduced in the market and since the ISM is a license free world wide frequency band suitable for low power wireless communication it is the prime focus of these latest wireless technologies. As there is no co-existence mechanism exists for all these wireless technologies to use the ISM band in a collaborative manner, which results in a considerable degradation in the performance of such devices and interference in the ISM band. To further explain the phenomena of interference consider Figure 3.1, in which interference problem is shown as an overlap over time and frequency between two commonly used standards in the ISM 2.4 GHz band the Wlan and Bluetooth. The Wlan is using DSSS which is utilizing 22 MHz wide band to communicate and the Bluetooth is occupying different 1 MHz wide signals using Frequency Hopping technique [22].

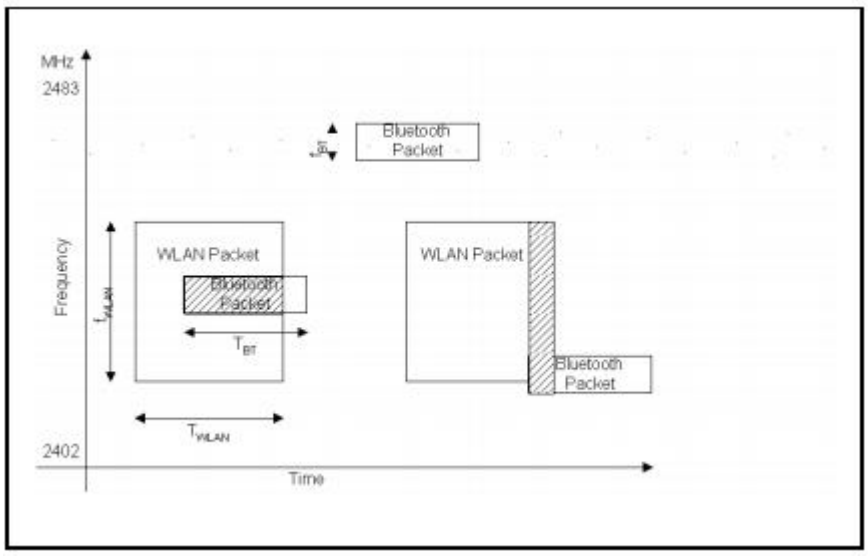

Figure 3.1 Time and Frequency Collisions in the 2.4 Ghz ISM band [22]

The short range wireless communication between the CyberStick and the wireless device also utilizes 1 MHz wide frequency which can be selected from 2400 to 2483 MHz range. The reason for this 1 MHz channel selection is due to transceiver design of

the RFM. It is safe to assume that interference with Wlan and Bluetooth devices will cause major interruption in the communication between CyberStick and receiver device connected to CAVE system software COVISE, as Wlan is always installed in most research & development environments and Bluetooth is available in almost all of the hand held electronics products. Moreover presence of other devices communicating over 2.4 GHz band is possible such as fire alarm network, microwave, laptops etc. The list of commonly used technologies used in the 2.4 GHz ISM band is given in the **Table 3.1**. [23]. For all the above mentioned reasons we can declare that interference due to the presence of other wireless devices communicating over 2.4 GHz band is the main concern in the degradation of reliable wireless communication in the virtual reality environment of CAVE. The reliability enhancement techniques for the mitigation of the impact of interference in ISM band can be related to both the physical layer and data link layer. These techniques are presented later in this chapter.

Protocols	Bluetooth	UWB	ZigBee	Wi-Fi
Frequency Band	2.4 GHz	3.1-10.6 GHz	2.4 GHz	2.4; 5 GHz

Table 3.1 Common Technologies in ISM band [23]

3.1.2 Received Signal Strength

The power of the transmitted signal has to be set according to the regulations and hence the overall design of the wireless communication system to achieve reliability has to be carefully planned. The power of the received signal is always less than the transmitted signal from the transmitter and is inversly proportional to the distance between a transmitter and the receiver. This indicates the important role the receiver has to play in order to extract the desired information from the received power attenuated signal. The design of the receiver and the demodulation technique used by the receiver is the crucial factor in achieving the miniumum requirements of the reliable communication. The most important point as mentioned in [24], the received signal strength should always be higher than the combination of noise and interference power for reliable detection of the received bits. It is also mentioned in [24] that the path loss which is the difference between the power of a signal when it is transmitted and when it is received is directly proportional to the central frequency the signal is transmitted on. Which means that as higher the operating frequency of a wireless technology the distance on which it can reliably communicate will become shorter given that the transmission power remain constant. Also it can be inferred that in case there are multiple receivers for a single transmitter, the receivers which are in close proximity as compare to other receivers will be able to receive signal from the transmitter with higher strength and hence can

communicate with high reliability than other receivers placed at distant location. This phenomena also add another dimension to the problem of interference called hidden node shown in ***Figure 3.2*** as there are more wireless devices within the range the higher the probability of the interference, specially if the distance between the desired communicating wireless device is more than another undesired wireless device.

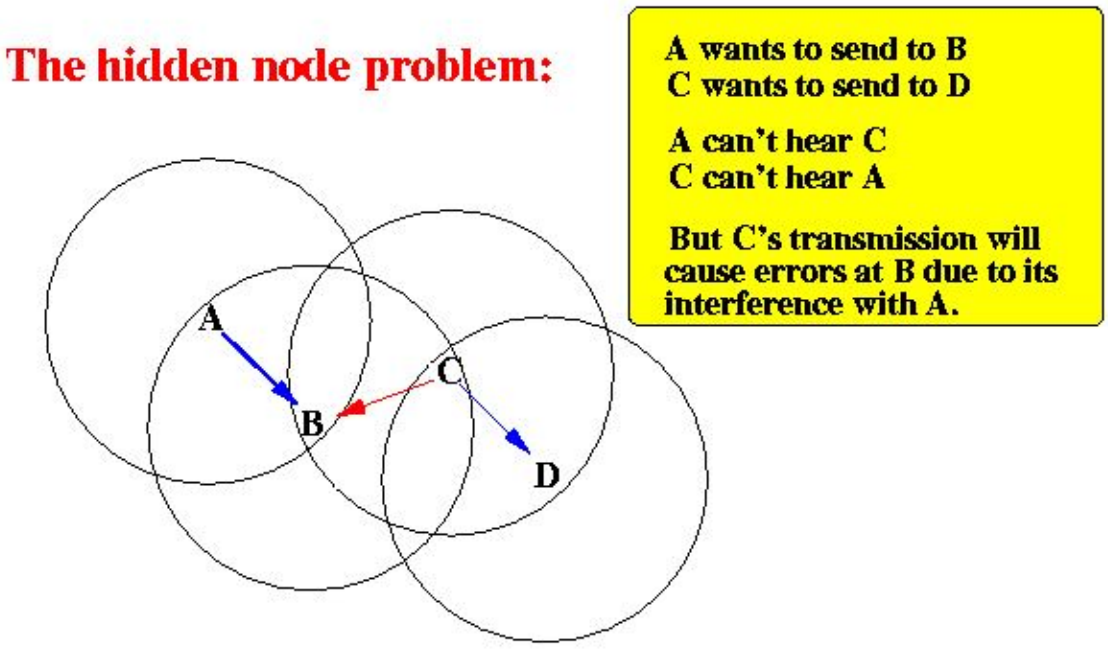

Figure 3.2 Pictorial Representation of Hidden Node problem courtesy to mathcs.emory.edu

The transmission power of the transmitted signal and design of the receiver to extract the desired information from the received signal is an attribute of the physical layer and is handled by the RFM in our case. Hence there is not much scope of research and implementation in this area. Only the transmission power of the RFM can be varied by using the appropriate registers allocated to set the transmission power of the RFM from a range of -10 dBm to 5 dBm.

3.1.3 Multipath and Shadowing

Fading is the attenuation of the quality of a signal cause by the environment in which wireless communication is taking place. The media in which fading occurs is termed as fading channel. Fading is mainly caused due to two reason one multipath propagation of waves and second due to presence of an obstacle in the path of the wave propagation which is termed as Shadowing. The environment in which transmitter and receiver are

communicating impact the reliability of data transmitted and received. The presence of the reflecting objects around the transmitter or receiver will cause multipath propagation of the waves, which can result in constructive or destructive interference. Multiple copies of the same signal will be created with respect to delay, phase shift and signal strength attenuation when the transmitted signal is reflected by reflecting material. This can result in multiple signals with the same information arriving at the receiver as shown in ***Figure 3.3***.

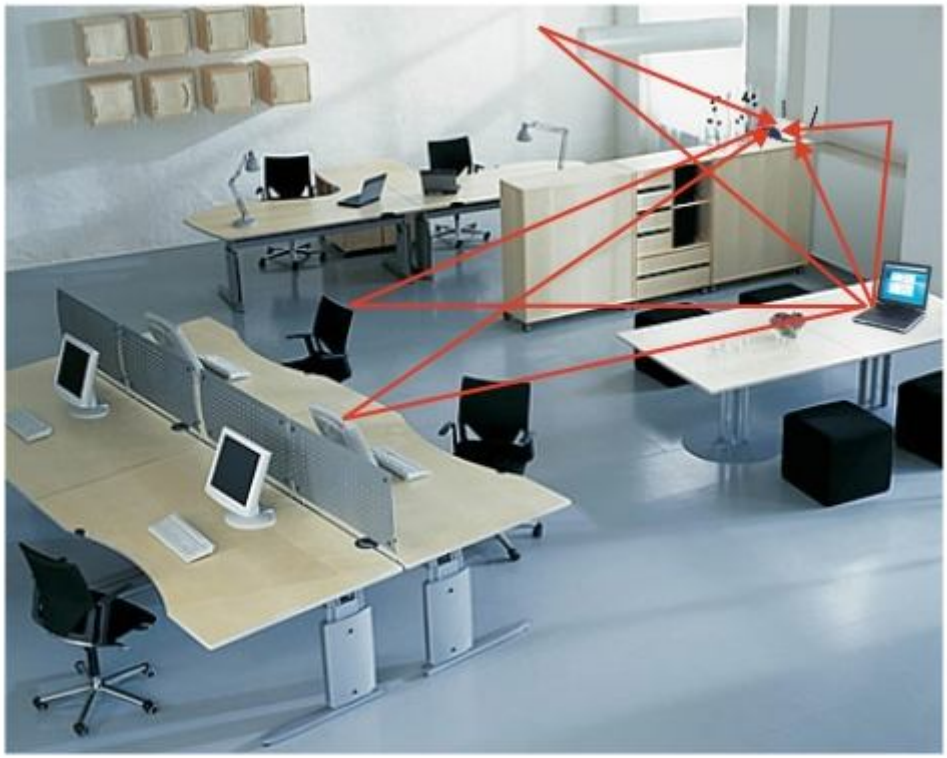

Figure 3.3 Representation of a possible Multipath scenario courtesy to nari.ee.ethz.ch

The virtual reality environment of CAVE is designed to provide a sense of reality by covering all the sides with screens. The screens can be considered as reflecting material as their surface are like mirrors. The CAVE environment has a potential to generate multipath propagation of the waves and can cause constructive or destructive interference which can be termed as concern for the reliable communication. Virtual reality CAVE has the space to be used by multiple users at the same time which means more than one human body is likely to be present inside CAVE and this situation can cause the effect of Shadowing as one body can become an obstacle in wave propagation

from the CyberStick of another user. The solution of multipath lies within the design of the receiver which is outside of the scope of this thesis project.

3.2 Reliability Enhancement Techniques

Various wireless technology standards in ISM band implement different strategies for reliability enhancement at physical layer and data link layer. There has been some research carried out to allow a collaborative mechanism for different wireless technologies to operate in the same environment but mainly most ideas are developed for the collision avoidance and error correction. Most common wireless technologies operating in the ISM band are Bluetooth, Wlan and home automation devices. In this part reliability enhancement techniques used in ISM band wireless technology standards are discussed and later on these techniques will be compared based on our project requirements.

3.2.1 Forward Error Correction Techniques

The introduction of errors by the communication medium is a significant area of concern in any communication environment and to increase the reliability of a wireless communication systems various error control strategies are proposed and one of them is Forward Error Correction. The FEC allows the wireless communication system to correct errors in the received data in an efficient and accurate manner from the transmitter. If there are 'n' number of bits to be sent, the FEC will add 'k' number of bits and 'n+k' bits will be transmitted where 'k' are the amount of redundant bits. Upon receiving the data the receiver will be able to detect errors using the 'k' redundant bits and based on the FEC algorithm the receiver will correct data without asking the transmitter for retransmission. A simplistic idea of a FEC encoder is given in the ***Figure 3.4***. There are several FEC techniques presented in different research articles such as Turbo codes and short forward error-correcting codes [25][26].

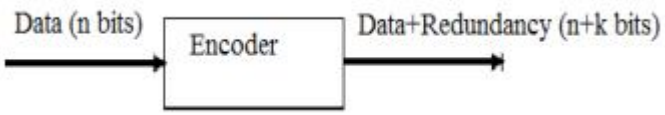

Figure 3.4 Forward Error Correction Technique [25]

The implementation of any sort Forward Error Correction algorithm is not possible as it require enhancements in hardware which is in our case the pre-defined RFM

transceiver. The RFM provides some level of error detection by performing cyclic redundancy check and discard any message which fails CRC test.

3.2.2 Automatic Repeat Request

Automatic Repeat Request protocol is another error control mechanism commonly used in wireless communication systems [27]. If the transmitter does not receive an acknowledgement message from the receiver, it means that either the packet was not received or it was corrupted during transmission through the medium. In which case the transmitter starts auto-retransmission until it receives an acknowledgement message or after retransmitting specific number of times without an acknowledgement it cancels the transmission of that particular packet. The pictorial representation of ARQ is given in ***Figure 3.5*** given below.

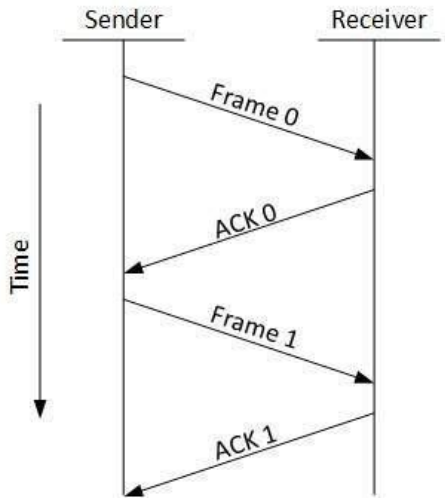

Figure 3.5 ARQ Error Control mechanism courtesy to tutorialspoint.com

ARQ error control mechanism can be implemented on hardware and also at upper layers. The implementation at upper layers than the physical layer will increase the latency of the overall communication system which is also proved during this project. In the chapter of Result and Verification the comparison of the time taken by auto acknowledgement implemented at physical layer and manual acknowledgement implemented in application software on the microcontroller is provided. The RFM provides an efficient hardware implementation of ARQ mechanism and hence it is sufficient for the latency requirement of our project and later on these techniques will be compared based on our project requirements.

3.2.3 Beacon Frame

802.11 type wireless stations utilize various type of frames to perform wireless data communications such as data frames, management and control frames[28]. Beacon frame can be termed as a management frames periodically transmitted from the central wireless station to provide all the necessary information to communicate with various wireless devices available in the vicinity of the central wireless station (AP) [29]. Beacon messaging is an essential requirement in the reliability enhancement when one central wireless station is communicating with multiple wireless devices especially in the scenario of synchronized data transfer between the devices and it also allows the wireless devices to use reliability techniques such as FHSS and DSSS [28].

3.2.4 CSMA/CA

Collision detection is very difficult in a wireless network because of the hidden node phenomena and also due to the fact that a node cannot transmit and listen at the same time [30][31]. CSMA/CA allows the node to avoid collisions before even they occur. In order to avoid or keep the number of collisions in a wireless network to a minimum, nodes are allowed to send data only if they cannot sense any carrier and the medium is idle[32][33]. This technique allows the medium to be used by multiple stations or nodes. The medium is shared by more than one node simply by allowing the nodes to transmit at their allotted time. Before each transmission a node listens to the medium to determine that if there is any another transmission taking place on the medium. If a carrier is detected the node has to wait for specific amount of time and then listen to the medium again. Upon determination node is allowed to transmit and wait for the acknowledgement, in case the acknowledgement is not received which means that a collision has occurred for various number of reasons. RTS or CTS can be used to avoid such collisions by one node specifically allowing one other particular node to transmit, in which case collision will be avoided as no other node is allowed to transmit. CSMA/CA in ***Figure 3.6*** is commonly implemented in WLAN applications as in a WLAN network an AP can see all the other nodes but some nodes may not be able to see each other (hidden node problem) due to distance and transmission power.

CSMA/CA is a data link layer MAC protocol and is possible to be implemented in scenario of this project. The RFM transceiver provides the hardware implementation of carrier sensing functionality at a particular frequency, which means sense before transmit idea can be implemented which is the most essential point in CSMA/CA protocol. The microcontroller which is controlling the RFM module has to be programmed for carrier sensing before it allows the RFM to transmit. If a carrier is sensed before transmission the implementation of waiting time is also possible as microcontroller posses timers which can be used to wait till a particular time. RTS and

CTS can also be easily implemented as a specific message code has to be embedded in a normal RFM packet to be transmitted so that when receiver collect the message it understands that only it is allowed to send and no probability of collision is possible.

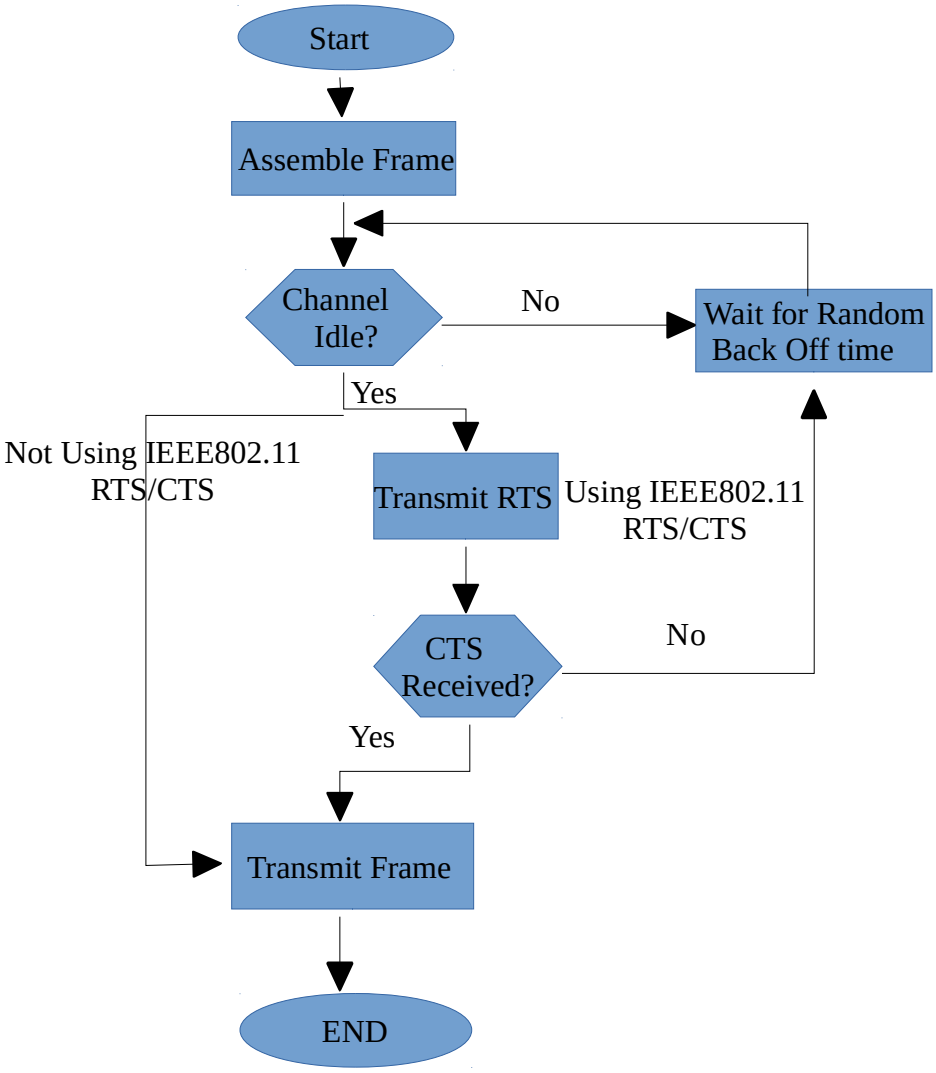

Figure 3.6 Flow Chart Diagram of CSMA/CA protocol
Courtesy slidshare.cdn.com

3.2.5 Spread Spectrum

Spread spectrum is a technique used in wireless communication in which a signal is transmitted by spreading over a frequency band rather than just using a specific frequency on which it can be easily transmitted. By using spread spectrum transmission the impact of noise, jamming and other types of interferences can be reduced. The spread spectrum transmitter transmits at the same power as of the narrow band one, but

since the power is spread of several frequencies the power density for each frequency is reduced and hence it can achieve low power transmission. It is due to these attributes of spread spectrum of interference resistant it has become a common application on most of the wireless communication devices such as cell phones, Wlans, cordless phones, radio modem devices etc [34]. Spread spectrum is further divided in to two types DSSS and FHSS. The DSSS uses the same carrier frequency to transmit the whole signal by spreading the signal over the band and FHSS uses different carrier frequencies within the band producing higher SNR as compare to DSSS [35].

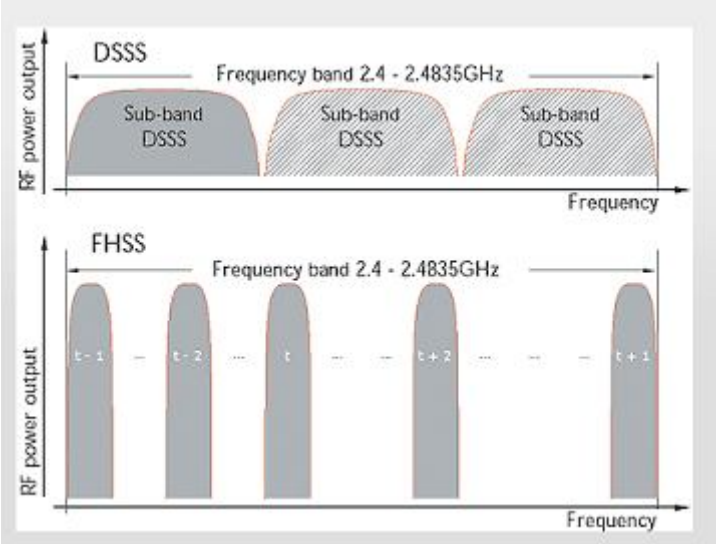

Figure 3.7 Pictorial Representation of DSSS and FHSS courtesy to "Industrial Ethernet" by Bruno Forgue

The implementation of DSSS or FHSS only possible at the physical layer [36] as these techniques require a specific design of the transmitter and the receiver. The RFM transceiver does not support FHSS or DSSS and hence cannot be implementable in this project and later on these techniques will be compared based on our project requirements.

3.2.6 Time Slotted Channel Hopping

Channel hopping was discussed in the previous section as part of the FHSS technique in which a transmitter continuously change or hop among various channels or frequencies within a specific band. The main bottleneck is that the receiver has to know about this random or defined hopping sequence in advance or it has to synchronize with the transmitter at runtime. Concept of time slots allows the nodes to synchronize with respect to time and communicate on different frequencies based on their hopping sequence. TSCH protocol utilize this concept of time slots and allows the reliable

communication in the scenario of wireless network in which the wireless communication nodes belong to the same network and hence can be synchronized. TSCH divides the time into equal slots in which the period of each slot depends on the application. The time slot is defined as amount of time for a message to be sent from node A to node B and a acknowledgement message is received from node B. It also include the time for the operational change of the node from transmitter to the receiver. In the scenario of common wireless networks with data bytes of 127 to be transmitted, 3ms time is enough and 1ms for acknowledgement including some time for the change of transmitter to receiver of a node which means 10ms for each time slot is considered adequate. The slots are then combined into a slot frame, which consist of transmit, receive or sleep slots. A network schedule is developed based on the application needs which runs the slot frames in a continuous manner over each node. The Node TSCH schedule specifies for each node whether it has to transmit or listen and at which frequency it should operate on and the address of the node it has to communicate with. The schedule will make sure that the two nodes are configured for the communication to take place for example if node A is transmitting to node B than node B should be configured as receiver. It is also possible that a single node can transmit for multiple nodes configured as receivers. The concept of ASN is used to synchronize the overall operations of each node with other nodes and each node can access the current ASN of the network at all times. Based on the knowledge of current ASN a node can calculate the frequency it has to hop on from the look up table of different frequencies available. TSCH is higly dependent on the time synchronization of the nodes within the network and hence each node must have a clock source. Since different clock sources can result in a time drift for each node, periodic time synchronization of the network clock of the node with its neighbour is required. A node will be defined as time source neighbour so that all other nodes can synchronize their network clocks with the time source neighbour node. For time synchronization purpose another idea is opted in the TSCH which is in every message exchanged between the nodes, the message will contain their time information and hence nodes can synchronize during the communication as well. EB type messaging can also be utilized after every now then for information exchange between the nodes [37].

TSCH is MAC layer protocol and can be implemented by defining the functionality on microcontroller controlling the RFM. TSCH offers a lot ideas which are very interesting with this project requirements. Time slots can be easily implemented and can provide reliability as well. Channel hopping is another important idea which will allow the CyberStick and the receiver to find the interference free frequency at real time and use that frequency for communication. The timers provided in the microcontroller have time resolution of microseconds and can easily provide interrupts after every specific amount of milliseconds. In our scenario receiver is connected to a virtual reality environment

which can provide clock at all times but the CyberStick is a remote device and is not connected to a separate clock, which means that time synchronization for CyberStick based on its own clock is not possible. As a whole TSCH provides with lot of techniques can be used in the development of reliable bi-directional protocol and these ideas will be discussed in the next chapter of Development of Bi-directional Protocol.

3.2.7 Channel Blacklisting

TSCH allows the channel hopping on certain frequencies defined by network schedule but in case a node observe interference on a particular frequency it will not try to avoid this channel next time when it has operate on that channel. In order to solve this problem Wireless Hart protocol proposes a concept of channel blacklisting [38]. Channel blacklisting means that if a frequency on which lot other wireless devices operating on than that channel will be included into a blacklist. This blacklist of frequencies on which there is lot of interference will be maintained and the nodes will try to avoid these frequencies to communicate. An implementation of channel blacklisting for A-TSCH is presented in [39], in which channel quality estimation will be performed and based on the results channels can be declared as unreliable and added in the blacklist. Another technique which can be utilized for the decision making of blacklisting a channel is spectrum sensing. As channel quality estimation is time consuming and it is difficult to implement, the spectrum sensing provides an easy solution to the problem. In TSCH time slots a specific time slot has to be defined for carrier sensing and the blacklist has to be periodically updated. It is also quite important that all the nodes share this information on the blacklist. Channels can be included into the blacklist and can excluded as well at all times and the nodes will perform adaptive frequency hopping based on the order of blacklisted channels.

A-TSCH is also fall into the MAC layer protocol as it is a variation of TSCH and most of the ideas presented in A-TSCH are implementable in this project scenario. Spectrum sensing can be performed by using the carrier sensing functionality of the RFM and a blacklist can be managed inside the microcontroller.

3.3 Latency in CyberStick Communication Environment

Latency is a very important remote input device CyberStick design aspect and it is imperative to keep latency on the CyberStick communication with COVISE systems software running on CAVE as minimum as possible. There was no benchmark for the latency described in the problem description of this master thesis work, but during the research and implementation of different reliability enhancement techniques a minimum range for latency value was evident. Mainly CyberStick is developed for human interaction and provide the capability for navigation within the virtual reality

environment, which also helps in making an educated guess for the minimum latency requirement. The goal is to achieve the reliability in wireless communication keeping the latency value such that no impact is sensed on the human interaction with the virtual reality environment.

3.3.1 Impact of Reliability Enhancement on Latency

Reliability enhancement techniques discussed in the previous section are the main cause for increase in latency. Latency can be considered the main bottleneck in the implementation of reliability enhancement techniques and is the deciding factor is selection among those techniques. Acknowledgement messages is first basic step towards achieving reliable communication which also increases the latency. The receiver upon arrival of a packet from the transmitter will send an acknowledgement packet back to the transmitter. The communication time is doubled for single successful message transmission and in case due to interference in the frequency channel and acknowledgement is not received, other techniques will be implemented such as auto retransmission which will also result in increase in latency. To detect interference prior to the message transmission a carrier sensing technique can be implemented, in the RFM transceiver used in this project this technique can only be used in the receiver mode which means the device have to be switched around every time carrier sensing is used. Also to determine the exact time period for which carrier sensing can be done on single frequency or multiple frequencies is a design challenge and will result in further addition in latency. Based on the results of carrier sensing if interference is detected than that frequency have to be avoided and transmitter frequency will be change to different value, but first it has to be determined that which frequencies are currently not occupied by other wireless devices. An important relevant factor is that these electronic components used in transmitter and receiver can perform these communications in terms of milliseconds as we are operating in 2.4 GHz ISM band. This wireless communication within some milliseconds allows the implementation of these reliability enhancnment techniques without being any difference sensed by the human perception depending on the way these techniques are organized in a wireless device.

3.3.2 Design Issues

Three design problems were identified which will definitely result in the increase of latency of CyberStick communication. First major problem was the RFM turn-around from receiver to transmitter and vice versa, most importantly synchronization in the turn-around is required. After intense working with RFM it was realized that the time taken by the RFM to switch from transmitter to receiver was longer than from RFM switching from receiver to transmitter. The transmitter was sending the message and the receiver was transmitting the acknowledgement packet but since receiver transmitting

too quickly and the transmitter was not timely turn-around into receiver the acknowledgement packet from the receiver was lost. This result in additional delay on the receiver side as receiver has to wait for the transmitter turn-around, resulting in additional latency in the overall single communication process. Second major design problem which resulted in further increase latency was frequency synchronization. As discussed previously that when acknowledgement is not received or interference is detected using carrier sensing, the operating frequency of wireless link have to changed and this change will happen at run time. This means that the other wireless device have to search for the operating frequency of the first one. In our case receiver has the capacity for carrier sensing and frequency switching, and the CyberStick will search and synchronize to the receiver operating frequency at run time. This will also result in latency increase as more number of frequency channels to search through will cost additional delay per frequency channel.

3.3.3 Multiple Cybersticks

One of the main objective of this master thesis work is to enlarge the scope of the bi-directional communication from single CyberStick single receiver to multiple CyberSticks single receiver. Maintaining the reliability aspect of the wireless communication and keeping the latency minimum. To allow multiple CyberSticks to communicate with single receiver the concept of time slots and scheduled messaging is suggested in the reliability enhancement techniques section. Through this concept a particular time slot will be allotted to each CyberStick to communicate with receiver and all the other reliability enhancement techniques will be carried out within this time slot. This time scheduled messaging will allow each CyberStick to communicate with the receiver without interfering with each others messages. This will have a multifold impact on the latency as more number of CyberSticks means more time period after each CyberStick is allowed to communicate. In case CyberStick communicate a message before or after it is allowed than it has to wait for a specific amount of time. Much more worse scenario in case the frequency is switched on the receiver side and all the CyberSticks have to find the new operating frequency of the receiver, this problem will require careful design of protocol and definitely increase in latency.

4 RELIABLE BI-DIRECTIONAL PROTOCOL

This chapter will further build on the reliability enhancement techniques presented in the previous chapter. The pros and cons for each reliability enhancement technique will be discussed considering latency as the crucial factor. The deep thoughts will be presented in the form of arguments about the decisions taken in the development and implementation of the reliable bi-directional protocol. The argument about the sequence of these reliability enhancement techniques and the implementation decision on either side receiver or CyberStick will be discussed. These reliability enhancement techniques will be compared and finally a reliable bi-directional protocol will be presented.

4.1 Brainstorming

The first step in the development of a new algorithm or set of rules is generation of various ideas. Luckily the short range radio frequency communication is heavily researched subject and continuously new ideas are being developed in this field. In order to grasp those ideas all the major research in the topic of reliability enhancement in ISM 2.4 Ghz band was read, whether it was in the form of 2.4 Ghz wireless standards such as Wlan, Bluetooth etc or individual techniques proposed in the research papers. This research helped in developing the knowledge required to cherry pick the techniques which were relevant to our application scenario. All these techniques are already presented in the previous chapter State Of The Art. The next step was to select those techniques which can be implemented theoretically on the hardware used in this project, as it is clear that some of the techniques require hardware support. Than these techniques were tested individually on the hardware for the verification. When the basic ideas were implemented which can provide the platform for further implementation of complex ideas, than development of reliable bi-directional protocol was started. All the verified techniques which were passed for implementation are given in the Table 4.1.

	ACK	***TSCH***	***CSMSA***	***Data Pipe***
1	Auto Ack	Time Slots	Carrier Detection	Address Switch
2	Manual Ack	Beacon Frame	RTS	Dummy Pipe
3	Message with Ack	Channel Blacklist	CTS	
4		Channel Hopping	Frequency Synch	

Table 4.1 Implementable Reliability Enhancement Techniques

4.2 Basic Idea

After the selection of basic reliability enhancement techniques used to mitigate interference, a first concept of the possible algorithm was put on paper. This first concept can be easily understood by the use of flow chart. The flow chart in ***Figure 4.1*** depict the software implemented as a result of selection of basic reliability enhancement techniques and the order of the execution and decision making mechanism in the implemented software.

The first stage is initialize, in which the microcontroller initializes the RFM .
Initialization values for RFM (Receiver & CyberStick) :

Auto Retransmission = 15 max. ,	Auto Retransmission Delay = 4 m.s. max
Operating Frequency = 2435 Mhz ,	Air Data Rate = 2 Mbps max.
Output Power in TX mode = 5 dbm max. ,	Payload Length = 32 bytes max.

In second stage of the basic concept of the algorithm acknowledgement was implemented. As acknowledgement message is the most basic and easiest technique to achieve reliability in wireless communication. It was implemented on both CyberStick and receiver side. Which means upon every receival of a message, the receiver side will send an acknowledgement message. The manual acknowledgement type was implemented and the reason was that at a later stage may be we need a feedback message from COVISE, and in that case manual acknowledgement will be useful and we can use the same implementation on the receiver side for both purposes. The basic idea was limited to the acknowledgement stage and now we can determine whether the wireless link between CyberStick and the Receiver device is reliable or not.

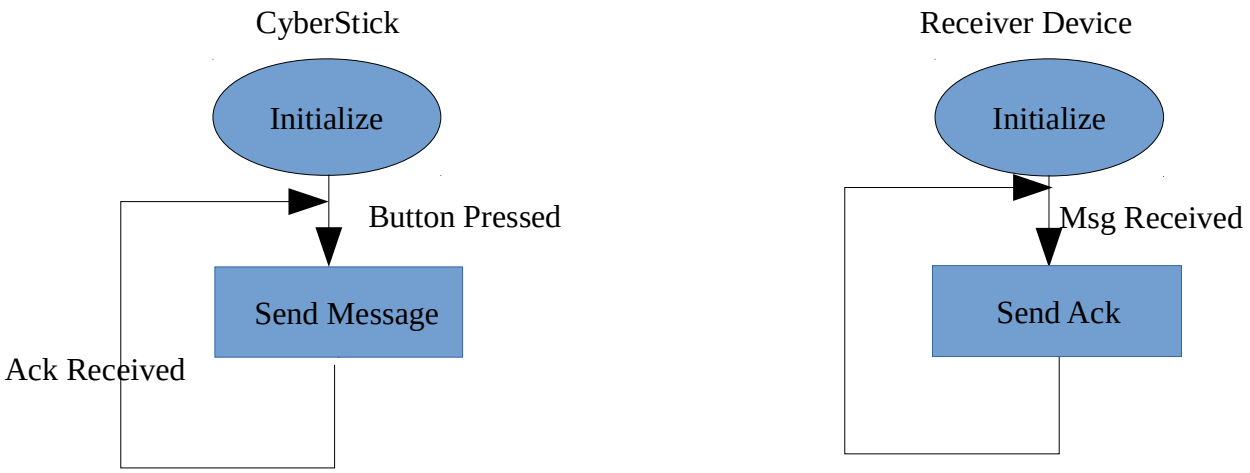

Figure 4.1 Basic Idea Implemented on CyberStick and Receiver Device

4.3 Maintain Reliability

The basic idea only allow to recognize that there is some reliablity issue with the wireless link. It does not determine whether it is due to interference from other wireless devices or for any other reason. The second step was to introduce a interference detection and frequency synchronization incase operating frequency is switched. This interference detection can be run continuously after a time interval to monitor the whole spectrum band of the RFM. Carrier detection is provided as a hardware support functionality in the RFM. The carrier detection is implemented on the Receiver side because the receiver device is connected to COVISE and have access to continuous power source. Due to power optimization the CyberStick will not remain online whole period of time. With the carrier detection implementation the receiver can identify interference from other wireless devices, and in case interference is detected it has to switch frequency. This technique can also be termed as frequency hopping, but in frequency hopping the frequency is switched continuously disregarding the interference detection. This technique can be configured with the implementation of time slots which is discussed later in this chapter.

Now algorithm is modified as shown in ***Figure 4.2*** and can adapt to the environment in which it is operating and also provides a solution in case acknowledgement is not received. If the CyberStick sends a message and does not receive an acknowledgement, it means the receiver has switched its operating frequency and the CyerStick should search for receiver frequency. The receiver will run carrier detection after every specific period of time and if carrier is detect it will switch its frequency.

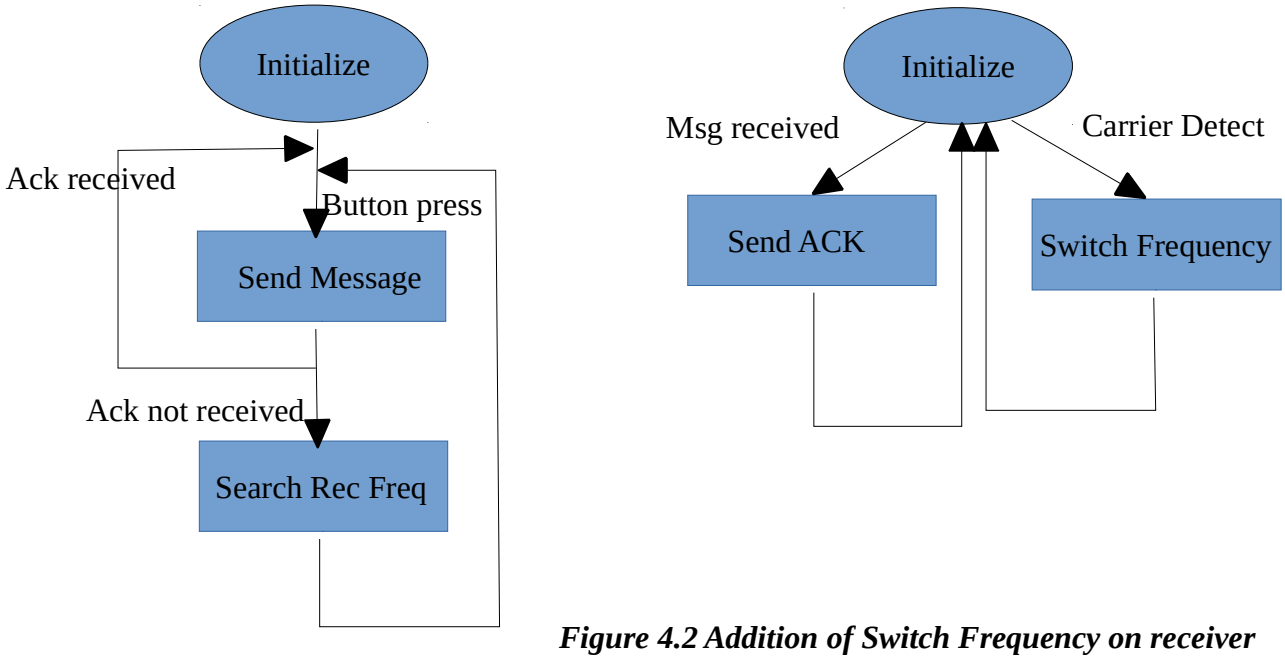

Figure 4.2 Addition of Switch Frequency on receiver side and Search Frequency Functionality on CyberStick

4.4 Time Slots on Receiver Device

To further develop the protocol, usage of time slots for various purposes was necessary. It will allow to put a timing constraint on the message exchange between CyberStick and the receiver. Time slots could also be implemented on the CyberStick side, but the main reason for the implementation of time slots on the receiver side was that at later stage this protocol will also handle multiple CyberSticks. The receiver will remain single and will control the overall environment.

The interference which can be caused due to the presence of multiple CyberSticks can be eliminated if the receiver have the decision making power that which CyberStick is allowed to communicate. Time slots will help defining the time of this communication so that receiver can switch to other CyberSticks based on time spent. Since the decision making power is given to the receiver device, the implementation of receival of acknowledgments is also shifted to the receiver side. The receiver will send the message which means CTS and the CyberStick will send an auto acknowledgment packet to indicate that it has successfully received the CTS message. Auto acknowledgement will help to decrease the time of a single communication between the receiver and the CyberStick and hence time period of each slot will also be decreased. Another modification was carried out which also helped in increasing the efficiency and the reliability. The CyberStick will send its payload with auto acknowledgement packet, which means that there will always be a single communication within a time slot and we can guarantee the timing of this communication. The time slots length was fixed at 10 ms which is enough to send CTS to CyberStick from receiver and the auto acknowledgement reception on receiver. If the receiver sends CTS to CyberStick and do not receive acknowledgement packet, it will term this situation as interference from other wireless devices and start carrier detection funcntionality. Upon determination it will switch to a suitable frequency channel and the CyberStick will search this frequency by listening to all the frequencies within the 2400 to 2483 spectrum. To reduce the search frequency functionality time on the CyberStick, the receiver will sent CTS after every 2ms if it does not receive an acknowledgement packet from the CyberStick. Which means CyberStick have to listen to each frequency for not more than 3ms and ultimately reduce the search frequency time significantly.

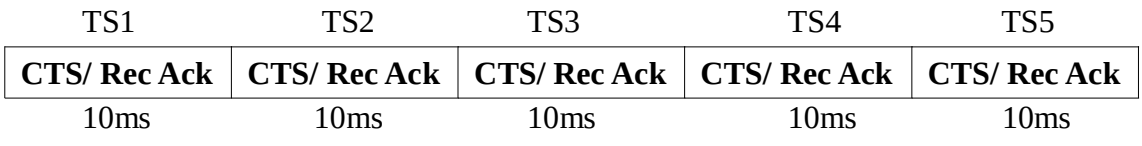

Figure 4.3 Message Packet Frames on Receiver Device

4.5 Dummy Communication Pipe

The next big step was to allow multiple CyberSticks to communicate with the receiver device. With the usage of time slots it looks implementable and two solutions were worked out to solve this problem. The first was to allow multiple CyberSticks to use different frequencies to communicate with the receiver, in which case time slots will not be necessary. The second was to utilize single frequency but to use the concept of address switching and dummy pipe with time slots.

The second choice was selected for implementation over the first one. The drawbacks to allow each CyberStick a different specific frequency to communicate with the receiver means that we should have more than one frequency channels. In the license free ISM 2.4 GHz band there is already a scarcity of free channels and to occupy more than one channel will not be wise. Furthermore if we communicate on more than one channel than we have to monitor them for interference and in case if interference is detected in one channel the whole communication will be stopped. Another minor issue is to identify a specific CyberStick during communicating with multiple CyberSticks. This problem could have been solved by identifying each CyberStick with its each operating frequency. Before we were using specific frequency which was set at the time of initialization, but in real environment it is unrealistic as we cannot determine before hand which frequency is free to communicate on as the receiver might change its operating frequency at any time. The search receiver operating frequency functionality have to be utilized as soon as the CyberStick comes online and if each CyberStick has a different frequency to operate on it might listen to CTS message from receiver for different CyberStick.

The advantages of opting for the second solution to use time slots as a platform for communicating with multiple CyberSticks on single frequency by using address switching and dummy pipe were, that now there is only one operating frequency and to monitor one channel and to detect interference over is easier and reduce latency of the functionality. The problem of identification and inter-interference among CyberSticks was solved by using the concept of data pipes which was explained in detail in the chapter of State of the Art. To solve the problem of how to find out which frequency to operate on from CyberStick perspective and to notify the receiver that a specific CyberStick have came online dummy communication pipe was implemented. Which is the communication link for any CyberStick which comes online and after exchange of first message from receiver it will switch to its specific address. The dummy communication pipe utilizes address0 of the RFM out of six available addresses and the CyberStick will always send message to this dummy communication link to allow any new CyberStick to exchange the first message.

4.6 Finalized Protocol

The process of the development of the bi-directional reliable protocol have been explained and now the finalized protocol is presented in the form of pseudo code which is given below.

Pseudo Code of Bi-directional Protocol for CyberStick :

```
Search receiver frequency () //at address0 (dummy link) until found
{
        IF (receiver frequency found == true)
        {
                switch to specific address ( for eg CyberStick1 = address1 )
                and send handshake message with Auto ACK
        }
}

While(true)             // within main function
{
        Search receiver frequency()

        IF (Receive CTS)
        {
                IF (button press == true)
                send (Auto ACK+ Button press Msg)
        }
        ELSE
                send only Auto ACK

        IF (CTS not received time > (10 * number of CyberSticks online))
                Search receiver frequency()

        IF ( Button not pressed time > time out specified for CyberSticks)
                Goto power down mode
                When come back online Search receiver frequency()
 }
```

Pseudo Code of Bi-directional Protocol for Receiver Device:

Initialize with specific frequency

Start Time Slots of 10ms

```
While()
{
        Send CTS on address0

        IF ( Auto ack received on Address0 == true)
                CyberStick detected ( identify CyberStick from it handshake msg)

        Send CTS at all new addresses of detected CyberSticks
                each time slot for each CTS to each CyberStick address

        IF auto ack not received for any one of the detected CyberStick than remove
        that Time slot.

        IF auto acknowledgement not received for all the CyberSticks than switch
        frequency.

}
```

5 EMBEDDED SW DEVELOPMENT

This chapter will allow the readers to follow the step by step embedded software development and implementation carried out throughout this project. The main focus will be on the explanation of the software development and implementation on the microcontroller of various ideas developed for reliable bi-directional protocol explained in the previous chapter. Which microcontroller peripherals were used for the implementation of specific techniques and which RFM characteristics were utilized.

5.1 Feedback Messages

The first important implementation on the CyberStick was the software development for feedback messages. This implementation was different from the auto acknowledgement functionality provided in the RFM. As this implementation was carried out with the software implementation on the microcontrollers on CyberStick and the receiver device. This implementation allows the CyberStick communication environment to enable manual acknowledgement and the feedback messages from the COVISE to the CyberStick. It also helped in verification for later implementation of feedback messages based on the application running on the COVISE which is discussed in detail in the chapter of Future Work. The most important technique used in this idea is the RFM turnaroud, which means that if RFM is communicating as the receiver it will turn around and start communicating as the transmitter. As this was the first implementation step in this thesis project, a simple idea was implemented that the CyberStick will send a message and than waits for the feedback message. The CyberStick will act as transmitter and the receiver device will act as the receiver for the first message transfer, than the receiver will turn around and sends a feedback message automatically and the CyberStick should be configured as the receiver at that point. To achieve the specific timing was difficult so the idea was kept simple to implement for the time being. For every received message on the receiver device, a feedback message will be sent and the CyberStick should wait until it gets the feedback message.

5.2 Wake Up Interrupt

The second implementation was carried to save energy on the CyberStick remote device as it has to run on power batteries with limited amount of power to be utilized. CyberStick consist of two electronic components the Attiny88 microcontroller and the

RFM transceiver. Both of these electronic components have power down modes which are utilized for this implementation. The idea which was implemented was that if the CyberStick is not used for the specific amount of time than it will go into the power down mode. And when any button is pressed on the CyberStick it will come back online immediately. The timers provided in the Attiny88 used to calculate the time since the last button pressed on the CyberStick and as soon as that time is passed the CyberStick will go into power down mode. The ISR will be enabled on the Attiny88 before going into power down mode to allow the Attiny88 to come back online when the next button is pressed. The buttons on the CyberStick were connected to Attiny88 pins which were configured as ISR sensitive. As soon as the button is pressed a pin change interrupt is generated at Attiny88 and the CyberStick will come back online.

5.3 Debugging Interface

After the first two software implementations it was realized that there is no interface provided for the debugging messages other than the use of LEDs to indicate successful message reception. There was no interface to see the values for example how much time taken in the communication between CyberStick and the receiver device. Therefore a hardware and software implementation for the debugging interface was carried out on the receiver side as it can be connected to the PC. The USART interface provided at the Atmega88 was utilized for this purpose which required soldering of two extra pin connections on the receiver hardware. These pins were than connected to the USART-USB bridge which can display data on the linux terminal using USB port on the PC. Any type of message which has to be displayed on the terminal contains an identifier within the packet so that the receiver understands that this information has to be communicated on the USART interface for display on terminal. The most difficult debugging message implementation was for timing values as they were floating point numbers. The serial communication of the USB-UART bridge handles one character at a time, so the floating point timing values have to be divided into single decimals and than to be communicated through debugging interface. A separate software implementation was carried out for the debugging timing messages.

5.4 Interference Detection

To allow the interference free communication a hardware implementaiton of frequency carrier detection functionality is provided in the RFM. The problem is that radio frequency communication takes place within milliseconds and it is difficult to detect. A software implementation was carried out for the utilization of the hardware implementation to detect the carrier with certainty. The RFM was allowed to detect carrier multiple times for the same frequency within a for loop so as make sure that frequency is free to be utilized. The whole spectrum will be searched from 2400 to 2483

Mhz range and each frequency multiple times. The detected frequencies will than be saved to allow the decision making later for which frequency to be communicate on.

5.5 Scheduled Messaging

Latency for the CyberStick communication with the receiver was a key concern from the beginning of this project. It was realized that in order to keep latency low and enhance the reliability a type of timing constraint is necessary of the communication to give a sense of assurance of the communication. Also it was necessary to implement scheduled messaging to allow later development of the bi-directional protocol. For this purpose the timers on the Attiny88 were utilized and were configured to generate an interrupt after every 20 ms. The button can be pressed at any time but luckily human sensibility does not differentiate within few milliseconds. So the idea which was implemented that regardless of the time of the button pressed, the CyberStick can only communicate with the receiver device within those 20 ms. The button pressed will be saved and message will be sent for that button press in the beginning of the 20 ms time period and than the CyberStick will wait for the acknowledgement. As 20 ms was enough time to get an acknowledgement from the receiver device.

5.6 Frequency Synchronization

The communication between the RFM modules of the CyberStick and the receiver device can only take place if both the RFMs are tuned to the same frequency. Previously both the RFMs were configured to the same frequency to easily communicate with each other. But with the implementation of Frequency Detection functionality on the either side, it was required that a frequency synchronization functionality at runtime should be implemented on the other side. One side can switch frequency based on the frequency detection functionality results and than the other should search the current operating frequency at runtime. The functionality was designed to send messages on all frequencies one by one and than wait for the acknowledgement on every frequency. The scheduled messaging was also utilized for frequency synchronization. With every interrupt generated at after every 20 ms a frequency is switched and message is sent on that frequency. Than the device will wait for the acknowledgement for 4 ms and if the acknowledgement is not received it will retransmit the message on the same frequency until next interrupt of 20 ms time period is generated. This process will continue to repeat itself until it gets an acknowledgement on a particular frequency and than the device will shift to normal communication mode.

5.7 Time Slots With Auto Acknowledgements

Further software development and implementation was carried out for scheduled messaging concept which later resulted in the implementation of Time Slots which can encompass other ideas as well. The concept of Auto Acknowledgement and manual acknowledgement was discussed in the topic of RFM in the second chapter. It was verified that the latency for auto acknowledgement was much less than that of manual acknowledgement. As the auto acknowledgement functionality was provided as the hardware implementation on the RFM which can be utilized easily. Time slots were implemented as the platform to implement further ideas like transceiver turn around, carrier detection and auto acknowledgements.

The concept of time slots was implemented on the receiver device using the timers in Atmega88 microcontroller. The timers will generate an interrupt every 20 ms and sends a beacon message which requires auto acknowledgement message from the CyberStick. If there is no auto acknowledgement message the receiver will consider interference and will start carrier detection functionality. With auto acknowledgement utilization in the receiver side it was necessary to transciever turn around on the receiver RFM, as the RFM goes into RX mode automatically in auto acknoweldgement messaging and comes back in TX mode itself as well. This transciever turn around was also implemented on the CyberStick side. Carrier detection was not part of the time slots at this point but it can be considered to be included within every 20 ms time slot. Time slots were only implemented on the receiver side as it is the controller of the overall communication with the CyberStick. It is safe to say that at this point the receiver is converted to transmitter and the CyberStick is changed to receiver.

5.8 Latency Minimization

Latency of the wireless communication between the CyberStick and the receiver device was further reduced by single message communication in each time slot rather two message communication. Previously the receiver will send a message which requires an auto acknowledgement and upon receiving the beacon message from receiver the CyberStick will send the auto acknowledgement packet and than its own information message if the button was pressed. This technique required transceiver turnaorund on the CyberStick side and on the receiver side and increases latency. With the implementation of With Acknowledgement Payload, the length of time slot was reduced to 10 ms. The receiver will send a beacon message on the start of every 10 ms time slot and upon receiving the beacon message from the receiver the CyberStick will send an acknowledgement packet as discussed before but now it send the button press information with this acknowledgement packet as well. Resulting in reduction in latency

as their no need for transceiver turn around on either side and also reduce the overall latency of frequency synchronization functionality.

5.9 Implementation of Address Switching

To allow multiple CyberSticks to communicate with single receiver on single frequency the concept of address switching was developed and implemented. It will allow to avoid the interference which can be caused by more than one CyberSticks communicating at the same time. The RFM module provides the concept of data pipes for the same reason. The CyberStick is configured with a unique pipe address and the receiver device have all the addresses to recognize which CyberStick it is trying to communicate with. The receiver will start communicating on pipe0 initially and as soon as the CyberStick comes online it also try to find current receiver frequency by listening to pipe0. When the frequency synchronization is completed at pipe0 the CyberStick will than shift to its predefined data pipe address and the receiver at this point will also know which CyberStick have found the receiver operating frequency. The receiver from here on will also send a beacon message on the recently online CyberStick pipe address as well and pipe0 as before. In case the receiver does not get an acknowledgement message on the specific pipe of the recently online CyberStick, it will consider rather the CyberStick went offline or there is some interference at this frequency. In case receiver changes its frequency at runtime due to interference the cyberstick will switch back again to address of pipe0 until it finds the correct frequency and then switch to its predefined address.

6 RESULTS & VERIFICATION

This chapter describes the verification process used in the course to verify the implementation of this master thesis project. Every major implementation part was accompanied by the corresponding verification and the verification results were than analyzed to be acceptable or not in our application scenario. This chapter will detail the three verification environments used througout this project the linux terminal, usage of LEDs and the prototype of CyberStick.

In first part of this chapter all the verifications resulted by the use of linux terminal connected to the debugging interface on the receiver are discussed. The timing difference between manual acknowledgement and auto acknowledgement is analyzed, and through the timing verification the proof for usage of auto acknowledgement is given. Then we define the process of how interference at different frequencies was detected and how the results were verified. The concept of address switching and use of different communication pipes is verified using terminal. The second part of this chapter describes the verifications which were entitled to the use of LEDs on CyberStick and on the receiver device. The successful reception of a message and the acknowledgment was verified with the help of LEDs. The frequency synchronization between the CyberStick and the receiver was also indicated with the help of a certain LED on the CyberStick. The indication of the CyberStick device mode for power up or power down is also expressed by lighting up LEDs in a specific manner. The third part of this chapter belong to the verification of power management implemented in this project by developing a prototype of the CyberStick on breadboard.

6.1 Linux Terminal

Linux terminal in connection with the serial interface on the receiver device provided the most utilized verification mechanism in this project. The serial interface on the receiver device is actually a UART interface of the Atmega88 microcontroller, which means any data using 8-bit characters can be sent using serial UART communication. The serial interface of the receiver device is connected to a USB-UART bridge and the linux terminal can establish a serial connection with USB-UART bridge using USB interface on the desktop computer. The screen command provides the functionality through which a linux terminal can be attached to the USB interface to listen to the serial data arriving from USB-UART bridge. The CyberStick is connected with the

receiver device via wireless communication link and can also send verification and debugging messages to be publish on the terminal. The debugging messages from CyberStick carry a specific code to identify on the receiver side that which message are to be sent to the terminal and which messages have to be forward to the COVISE. This verification environment is given in ***Figure 6.1***

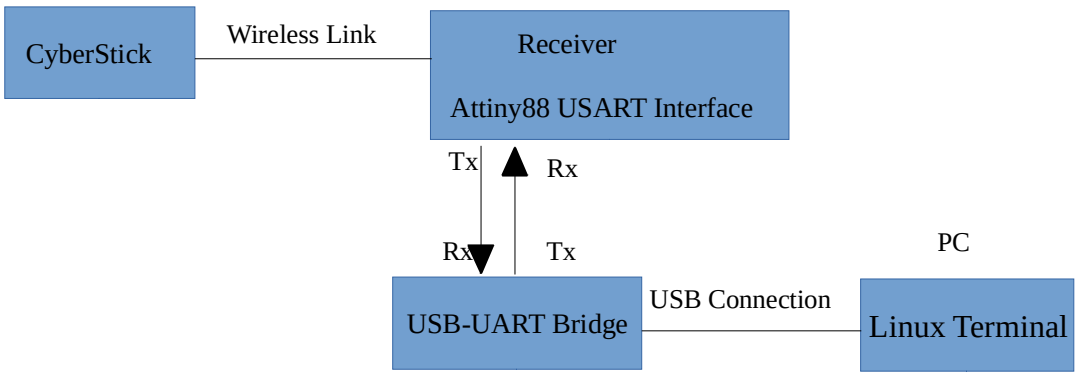

Figure 6.1. Verification Environment for Debugging Messages

6.1.1 Acknowledgement Timing Comparison

Two different type acknowledgements the auto acknowledgement and the manual acknowledgement have been identified in the beginning of this master thesis work for achieving reliability in the wireless communication between the CyberStick and the receiver device. As both of the implementations will increase latency and the implementation of any one of them was necessary. The selection was carried out based on the latency comparisons of the two implementations. For this purpose first manual acknowledgement was implemented on the CyberStick and the receiver device. So that when the CyberStick sends a message it initiates a timer and when it receives an acknowledgement from the receiver the timer stops and save the time spent. This saved time period is sent to the receiver with the next message accompanied with a special code so that receiver can identify that this message is carrying information to be publish on the terminal. The same process is then repeated for the auto acknowledgement implementation and it was verified that the messages with auto acknowledgement functionality have 60% to 70% less latency that of the manual acknowledgement. Timings are usually of integer or float or double type and cannot be communicated via serial interface, thus all the digits within the time value have to be converted into separate character of 8-bit to be publish on the terminal. The actual timing details of both implementation are presented in the form of graph in ***Figure 6.2 .***

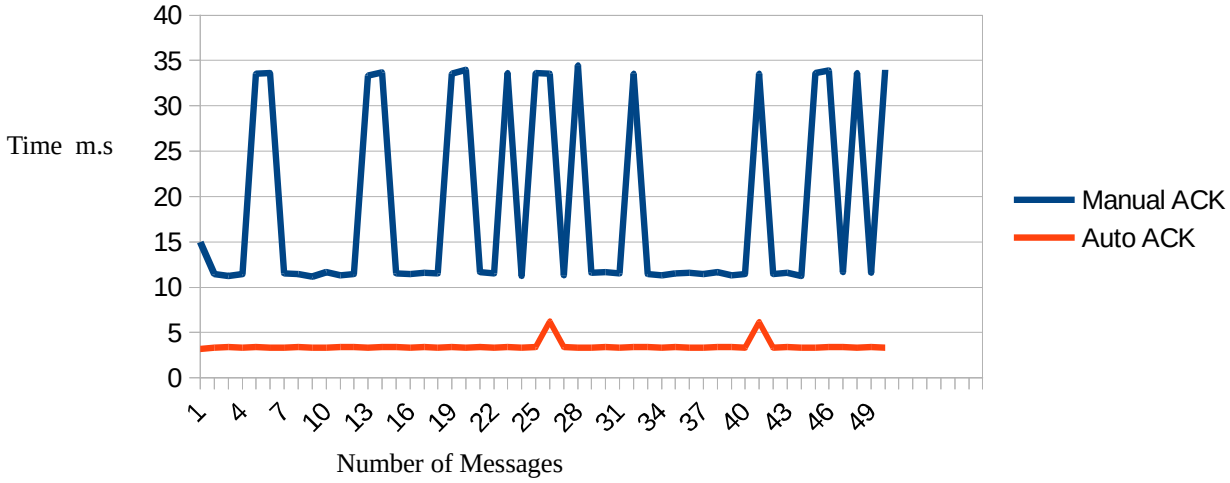

Figure 6.2 Acknowledgement Timing Comparison

6.1.2 Occupied Frequency Channels

To identify the currently occupied frequencies in the communication environment the implementation of a carrier sensing functionality was carried out on the receiver side. The RFM already provides the hardware implementation of the functionality and can detect carrier at the frequency which is set on the RfM register. In case a carrier is detect on that particular frequency one bit in the RF_CH register of the RFM is set high. In order to identify that particular frequency, the integer value of that frequency is sent to the terminal every time a carrier is detected. The functionality was programmed so that microcontroller changes the frequency form 2400 MHz to 2483 MHz one by one and than wait to detect a carrier. If a carrier is detected that particular frequency value is saved and the microcontroller changes the frequency of the RFM. In the end all carrier detected frequencies were sent to the terminal to be published.

6.1.3 Data Pipe Address Verification

The concept of data pipes provided in the RFM datasheet was utilized to allow multiple CyberSticks to communicate with single receiver at the same time. The RFM has the capacity to listen to six data pipes at the same time when configured as receiver. But to avoid interference among the CyberSticks the receiver sends the beacon message to allow specific CyberStick to communicate. During this implementation of data pipes concept it was realized that a method of verification is required that the receiver is performing the address switching correctly and can realize that a CyberStick has gone into power down mode or went offline for good and the receiver should switch to address0. Each CyberStick is given an address selected from the six addresses in the data sheet of the RFM. The address0 has to remain free for new CyberStick to come

online and from address1 to address5 can be assigned to CyberSticks. The receiver should send CTS at address0 and in case an acknowledgement is received for the CTS, the receiver should start sending CTS at the address of that received acknowledgement CyberStick. For the verification of this concept CyberStick will come online and then went to power down mode. The receiver should change beacon message sending address to address1 from address0 and when CyberStick goes into power down mode it should change its CTS sending address to address0. Everytime the address switching took place on the receiver, the receiver publish its new address on which it will send CTS message on the terminal via the serial debugging interface. Through this process it was verified that the receiver is performing accordingly.

6.2 LED Utilization for Verification

LEDs are a common element in the development of electronic systems whether they are analog or digital systems. They are easily interfaced with other electronic components and provide means for test and verification of an electronic circuit. The use of LEDs during this project was also to provide the CyberStick and the receiver device to communicate with the humans, mainly to indicate whether a process in completed or not and for debugging purposes. As shown in ***Figure 6.3*** the receiver device have one RED colour LED and the CyberStick have three different colour LEDs (***Figure 2.1*** CyberStick buttons are LEDs). During the application development of this thesis project these different LEDs were assigned for temporary indications and some for permanent indications when the product is ready to use.

Some temporary verifications involve RED LED blink every time a message is received at the receiver side and a green LED on the CyberStick to signal for successful acknowledgement reception. Another important indication using yellow LED on CyberStick is to indicate whether CyberStick is in power down mode or normal operation mode. The yellow LED is turned off if CyberStick goes into power down mode. Another important step was to implement some sort of indication when the CyberStick is able to find current receiver operating frequency. The functionality was designed so that every time the receiver changes its frequency the CyberStick will start searching for the current operating frequency and the user has to be notified that the wireless link betwenn the CyberStick and receiver device is established or not. The green LED on the CyberStick is used for established connection between the CyberStick and the receiver device.

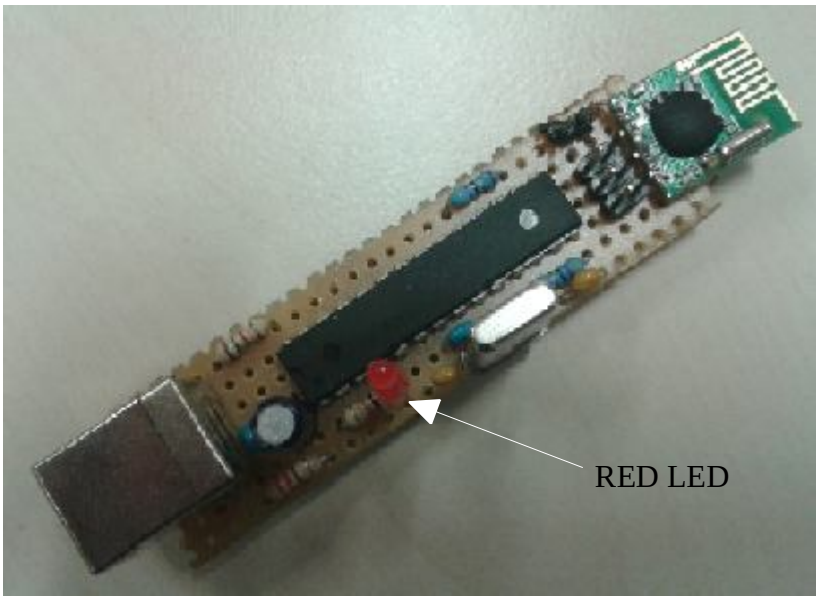

Figure 6.3 RED LED in the Receiver Device

6.3 CyberStick Prototype

CyberStick is a remote input device contain all the required circuitry in a 3D printed remote style box. To perform some of the verifications for the power management ideas the CyberStick circuitry connections have to be disconnected and modified. For this purpose it was realized that a prototype of the device must be built on the breadboard with temporary connections. This prototype helped in confirming the power consumptions of the RFM and Attiny88 at various operating stages provided in the respective data sheets. For better power management on the CyberStick the power down modes of both the RFM and the Attiny88 is utilized. The current consumption and supply voltage values for both devices are given below.

Supply Voltage = 3.3 v (for both Attiny88 and RFM) ….....
Internal Oscillator frequency = 8MHz
temperature = 25 Degree Centigrade

Attiny States	*Measured*	*Data Sheet*
Idle	0.7 mA	0.69 mA
Active	2.6 mA	2.8 mA
Power Down	0.00 uA	0.14 uA

Table 6.1 Comparison of Attiny88 Power Consumptions with Utilized Power Down mode

RFM Operating States	*Measured*	*Data sheet*
Standby-I	--	50 uA
Standby-II	261 uA	330 uA
Power Down	12 uA	2.5 uA
Transmitter (5dBm output power)	24.2 mA	23 mA

Table 6.2 Comparison of Power Consumptions of different Operating States of RFM

7 DISCUSSION

This is the last part of this master thesis document and helps to gain insight in the possible future improvements in the CyberStick prototype which were realized during the development of the prototype. This chapter also provides the conclusion of this master thesis work.

7.1 Future Implementation Enhancements

During the development of the reliable bi-directional protocol for CyberStick wireless communication various key enhancement areas were realized. Which will further enhance the reliability of the CyberStick communication, improve knowledge of the CyberStick status and provides more interactive communication with the COVISE system software running on CAVE.

7.1.1 Offline Message

During the implementation of the power down mode on CyberStick, it was realized that while receiver is continuously interacting with the CyberStick and the status knowledge of the CyberStick is crucial for the receiver device. As the receiver is configured t send CTS to every online CyberStick and the more the CyberSticks are present more time will be required to communicate with a particular CyberStick because the receiver can only communicate with one CyberStick at a time. It is proposed that an implementation of offline message from the CyberStick which is about to go offline will reduce the communication latency of the CyberSticks which are currently online. Before going offline each CyberStick will send a going offline message to notify the receiver so that the receiver will not send CTS on that address pipe here onwards and understands that device is went offline and there is no problem with the communication channel.

7.1.2 Super Frame

Super Frame [39] consists of time slots which are already implemented on the receiver device to avoid interference among the various CyberSticks. Time slots provides a specific amount of time to the receiver to exchange one message with each CyberStick within every slot. To allow the receiver to continuously monitor the wireless environment in which it is operating carrier detection is necessary. Carrier detection was

also implemented but to utilize its benefits a strategy is required just like with time slot communication with CyberSticks. Super frame will allow further enhancements in achieving reliability by allowing periodic carrier detection of the overall 2.4 Ghz operating band. Carrier detection for the overall 2.4 Ghz band cannot be implemented within time slots and require an implementation of super frame. Depending on the number of CyberSticks are allowed to communicate an appropriate super frame consisting of specific number of time slots can be defined. By having the interference information on all 2.4 Ghz operating frequencies, the best frequency can be selected when interference occurs on the current operating wireless frequency.

7.1.3 Hardware Modification

To provide an more interactive communication with the COVISE some hardware modifications can be carried out. It is proposed that for some target application running on COVISE some kind of feedback messaging will improve the sense of reality of the user. With the implementation of CTS messages on receiver device which can be used as feedback messages from the COVISE, only some hardware modification is required. For example an implementation of a vibrating motor on CyberStick is suggested.

7.2 Conclusion

This master thesis work was based on the development of a remote input device for virtual reality environment. The reliability of the wireless communication was the major focus of the research carried in this project. All the major reliability enhancement techniques utilized for 2.4 Ghz ISM band were focused and based on their implementation requirements and our application scenario some of the techniques were selected. The implementation of these selected techniques resulted in achieving reliability. Another major focus was to develop a protocol so that multiple remote devices can communicate with a single receiver while keeping latency minimum. A reliable bi-directional protocol is presented encompassing the implemented reliability enhancement techniques for multiple CyberSticks communicating with single receiver and maintaining low latency.

The power efficiency was another achievement of this project. It will allow to use CyberStick for a long period of time, while the user does not have to worry to switch off or on the CyberStick. Some further implementations are suggested to further improve the remote input device and its operation. In the end the overall project was considered beneficial for HLRS and the successful completion of master thesis requirement needed to graduate as a masters degree student for Tampere University of Technology.

LIST OF REFRENCES

[1] NC State University, ”Embedded Computer Systems”, http://www.ece.ncsu.edu/research/cas/ecs.

[2] Wayne Wolf, ”What is Embedded Computing”, Computer, vol. 35, no. 1, pp. 136-137, 2002.

[3] Sukriti Jalali, ”Trends and Implication in Embedded Systems Development,” http://www.tcs.com/sitecollectiondocuments/whitepapers/tcs_hitech_whitepaper_Trends-Implications-Embedded-Systems-Development.pdf, 2009.

[4] Lu Mai, Min Zaw Oo, ”Design and Construction of Microcontroller Based Wireless Remote Controlled Industrial Electrical Appliances Using Zigbee Technology”, International Journal of Scientific Research Engineering and Technology (IJSRET), vol. 3, no. 1, pp. 79-84, April 2014.

[5] ABI Research, ”The Internet of Things Will Drive Wireless Connected Devices to 40.9 Billion in 2020”, https://www.abiresearch.com/press/the-internet-of-things-will-drive-wireless-connect/, published Aug 2014.

[6] Muhammad Qutab-ud-din, ”Enhancements and Challenges in IEEE 802.11AH-A Sub-Gigahertz Wi-Fi for IOT Application”, Master of Science Thesis, ch. Introduction, pp. 1, November 2015.

[7] Jan Beutel. ”Design and Deployment of Wireless Networked Embedded Systems”, Doctor of Technical Science Dissertation, ch. Introduction, pp. 1, August 1973.

[8] HLRS, ”Virtual Reality”, https://www.hlrs.de/organization/av/vis/research/vr/.

[9] Navpreet Kaur, Sangeeta Monga, ”Comparisons of Wired and Wireless Networks: A Review”, International Journal of Advanced Engineering Technology, vol. 5, no. 2, pp. 34-35, April-June 2014.

[11] Phil Smith, ”Comparing Low-Power Wireless Technologies”, http://www.digikey.com/en/articles/techzone/2011/aug/comparing-low-power-wireless-technologies, Digi-Key Electronics, Article Library, 2011.

[12] Turning Technologies, ”RF Interoperability”, Version 1.3, https://www.turningtechnologies.com/pdf/UserGuides/RFInteroperabilityDoc.pdf, October 2013.

[13] HOPERF Electronics, ”Low Power High Performance 2.4 GHz GFSK Transceiver”, Datasheet, V2.0.

[14] Wenqi Guo, William M.Healy, MengChu Zhou, ”Impacts of 2.4 GHz ISM Band Interference on IEEE 802.15.4 Wireless Sensor Network Reliability in Buildings”, IEEE Transactions on Instrumentations and Measurements, vol. 61, no. 9, September 2012.

[15] Robert S. Allison, Laurence R. Harris, Michael Jenkin, Urszula Jasiobedzka, James E. Zacher, ”Tolerance of Temporal Delay in Virtual Environments”, Virtual Reality, 2001. Proceedings. IEEE, pp. 247-254, March 2001.

[16] M. Meehan, S. Razzaque, M.C Whitton, F.P Brooks, ”Effect of Latency on Presence in Stressful Virtual Environments”, Virtual Reality, 2003. Proceedings. IEEE, pp. 141-148, 2003.

[17] Margaret Rouse, Daniel kroon ”OSI reference model (Open System Interconnection)”, http://searchnetworking.techtarget.com/definition/OSI.

[18] Techopedia, ”Physical layer”, https://www.Techopedia.com/definition/8866/physical-layer.

[19] Highteck, ”Datalink Layer”, http://www.highteck.net/EN/DataLink/Data_Link_Layer.html.

[20] Margaret Rouse, Steve Curtis, ”Logical Link Control Layer”, Techtarget, http://searchnetworking.techtarget.com/definition/Logical-Link-Control-layer.

[21] Ismail Guvenc, Sinan Gezici, Zafer Sahinoglu, Ulas C. Kozat, ”Reliable Communications for Short-range Wireless Systems”, Book, ISBN 978-0-521-76317-2, Ch. Short-range wireless communication and reliability, pp – 8.

[22] Golmie N, ”Interference in the 2.4 GHz ISM Band: Challenges and Solutions”, National Institute of Standards and Technology, http://w3.antd.nist.gov/pubs/golmie.pdf.

[23] Chakkor Saad, Baghouri Mostafa, El Ahmadi Cheikh, Hajraoui Abderrahmane, ”Comparative Performance Analysis of Wireless Communication Protocols for Intelligent Sensors and Their Application”, International Journal of Advanced Computer Science and Applications (IJACSA), vol. 5, no. 4, 2014.

[24] Ismail Guvenc, Sinan Gezici, Zafer Sahinoglu, Ulas C. Kozat, ”Reliable Communications for Short-range Wireless Systems”, Book, ISBN 978-0-521-76317-2, Ch. Short-range wireless communication and reliability, pp – 10.

[25] Ranjeet Singh Tomar, G.S Tomar, ”Efficiency Enhancement Techniques for Wireless Communication Systems”, IJSST, vol. 11, no. 1.

[26] Sheng Tong, Xidian Univ. Xidian, Dengsheng Lin, kavicic A., Baoming Bai, Li Ping, ”On Short Forward Error-Correcting Codes for Wireless Communication Systems”, Computer Communications and Networks, Proceedings of 16th International Conference, pp. 391-396, 2007.

[27] Margaret Rouse, Sandi Seidi, ”Automatic Repeat Request”, TechTarget, http://searchnetworking.techtarget.com/definition/automatic-repeat-request.

[28] Jim Geier, ”802.11 Beacons Revealed”, Tutorial Article, http://www.wi-fiplanet.com/tutorials/print.php/1492071.

[29] Vishal Gupta, Mukesh Kumar Rohil, ”Information Embedding in IEEE 802.11 Beacon Frame”, National Conference on Communication Technologies & its impact on Next Generation Computing CTNGC 2012 proceedings published by International journal of Computer Applications (IJCA).

[30] Miquel Olive, Ana Escudero, ”Study of different CSMA/CA IEEE802.11 based-Implementations”, EUNICE 1999 Contribution.

[31] Viral V. Kapadia, Sudarshan N. Patel, Rutvij H. Jhaveri, ”Comparative Study of Hidden Node Problem and Solutions using different Protocols and Techniques” Journal of Computing, Vol.2, Issue 3, March 2010.

[32] ”Federal Standard 1037C”, Glossary of Telecommunications Terms, Published on August 7’ 1996.

[33] ”American National Standard”, ATIS Telecom Glossary, 11 April’ 2016.

[34] ”Introduction to Spread Spectrum”, By Randy Roberts http://sss-mag.Com/ss.html#tutorial

[35] ”Tutorial on Spread Spectrum”, Parabaka Parabakaran, N.L Polytechnic College http://www.eetimes.com/document.asp?doc_id=1271899.

[36] ”Layers One & Two of 802.11 Wlan Security”, SANS Institute, https://www.sans.org/reading-room/whitepapers/wireless/layers-two-80211-wlan-security-14.

[37] T. Watteyne Ed., M. Palattela, L. Grieco, ”Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things: Problem Statement”, Internet Engineering Task Force (IETF), May 2015.

[38] Mark Nixon, ”A Comparison of WirelessHART and ISA100.11a”, White Paper, Published on 23 September’2012.

[39] Peng Du, Dr. George Roussas, ”Adaptive Time Slotted Channel Hopping for Wireless Sensor Networks”, Computer Science and Electronic Engineering Conference (CEEC), pp 29-34, 12-13 September’2012.