Spring 4-18-2013

Design and Implementation of a Hybrid Technology Networking System

Sushanta Mohan Rakshit
University of Nebraska-Lincoln, sm.rakshit@gmail.com

DESIGN AND IMPLEMENTATION OF A HYBRID TECHNOLOGY NETWORKING SYSTEM

By

Sushanta Mohan Rakshit

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Master of Science

Major: Telecommunications Engineering

Under the Supervision of Professor Hamid R. Sharif-Kashani

Lincoln, Nebraska

May 2013

DESIGN AND IMPLEMENTATION OF A HYBRID TECHNOLOGY NETWORKING SYSTEM

Sushanta Mohan Rakshit, M.S.

University of Nebraska, 2013

Advisor: Hamid R. Sharif-Kashani

The safety of rail transport has always been the top priority for the Federal Railroad Administration (FRA). Legacy technology, like wayside monitoring, is still in place and is largely relied upon for detection of faults. Modern technology like Radio Frequency Identification (RFID) has been introduced recently. However, this is largely used to detect a particular railcar rather than to monitor it for problems. Wireless Sensor Network (WSN) technology is being evaluated by the railroads for real-time or near real-time monitoring of the status of railcars for timely response to problems and also for trend analysis.

ZigBee has been the networking protocol of choice for the railroads for its low power consumption and cost of implementation. The railroad scenario presents a long linear-chain like network topology which ZigBee was not designed to handle. It has been found that a ZigBee-only network in the railroad environment suffers from drawbacks like long synchronization delays, severe problems with route discovery and maintenance, aggregation of data errors leading to unacceptable packet loss rates, lack of a mechanism to decide traffic priority for critical packets, like alarm, so that they can reliably traverse the network to the collecting node in the locomotive etc.

Hybrid Technology Networking (HTN) protocol has been suggested which addresses the shortcomings of ZigBee in the railroad scenario. It proposes a standards-based multi-protocol approach that is well-suited for the railroad scenario. The current crop of sensor platforms does not provide an integrated environment for the implementation of HTN.

In this research work an integrated hardware platform for the implementation of the HTN protocol is designed and implemented. The guiding principle has been the adherence to standards. The test results using the hardware show that it provides inter-operability with available sensor platforms, can interface with other sensing hardware using standard protocols and provides communication capabilities exceeding that needed by HTN.

Dedicated to the countless thinkers and innovators who push the boundaries of knowledge today, for a better tomorrow.

Acknowledgments

I would, first and foremost, like to thank my advisor Dr. Sharif for his invaluable guidance and support during the course of my research work. I would also like to thank Dr. Michael Hempel for his timely inputs and new perspectives on the work that I was doing. I am also thankful to my colleagues Fahimeh, Pradhumna and Tao for their help and support during the time I was working on this research. I would specially like to mention Fahimeh for her unwavering support in helping me with laying out the PCB and teaching me how to work with the Cadence software suite.

I would like to sincerely thank the Federal Railroad Administration (FRA) for their generous funding of the project that has led to this research. I would like to mention John Punwani and Monique Stewart for their unwavering support of the project.

Although no amount of gratitude can ever be enough for them but still I would like to thank my parents for their unwavering support and belief in my dreams. I would not be here today writing these words without their constant inspiration and encouragement. Last, but not in the very least, a special mention for the angel with a flower tiara, who has looked over me and guided me through it all.

Table of Contents

List of Figures

List of Tables

List of Acronyms

ADC	Analog to Digital Converter
AEI	Automatic Equipment Identification
AF	Application Framework
AIB	APS Information Base
AP	Access Point
API	Application Programming Interface
APS	Application Support Sub-layer
APSDE	APS Data Entity
APSME	APS Management Entity
AREQ	Asynchronous Request
ARP	Address Resolution Protocol
bps	Bits per second
CAN	Controller Area Network
CRC	Cyclic Redundancy Check
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
CTS	Clear to Send
DAC	Digital to Analog Converter
DCF	Distributed Coordination Function
DHCP	Dynamic Host Configuration Protocol
DSS	Distribution System Services
DSSS	Direct Sequence Spread Spectrum
ED	Energy Detection

EPA	Environmental Protection Agency
FCC	Federal Communications Commission
FCF	Frame Control Field
FCS	Frame Check Sequence
FFD	Full Function Device
FHSS	Frequency Hopping Spread Spectrum
FRA	Federal Railroad Administration
GHz	Giga Hertz
GPIO	General Purpose Input Output
GPS	Global Positioning System
GTS	Guaranteed to Send
GUI	Graphical User Interface
HR-DSSS	High Rate-Direct Sequence Spread Spectrum
HTN	Hybrid Technology Networking
I2C	Inter-Integrated Circuit
I2S	Integrated Inter-chip Sound
ICMP	Internet Control Message Protocol
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IPv4	Internet Protocol version 4
ITU-T	International Telecommunication Union-

	Telecommunication
Kbps	Kilobits per second
LCD	Liquid Crystal Display
LLC	Logical Link Layer
LQI	Link Quality Indicator
LR-WPAN	Low Rate-Wireless Personal Area Network
MAC	Medium Access Control
Mbps	Megabits per second
MCPS	MAC Common Port Sub-layer
MCU	Micro Controller Unit
MHz	Mega hertz
MII	Media Independent Interface
MIMO	Multiple Input Multiple Output
MIPS	Million Instructions Per Second
MLME	MAC Layer Management Entity
MRDY	Master Ready
MSDU	MAC Service Data Unit
NLDE	Network Layer Data Entity
NLME	Network Layer Management Entity
NWK	Network
OBMCS	On-Board Monitoring and Control System
OFDM	Orthogonal Frequency Division

	Multiplexing
PAN	Personal Area Network
PCB	Printed Circuit Board
PHR	Protocol Header
PHY	Physical Layer
PID	Product Identification
PPDU	PHY Protocol Data Unit
RAM	Random Access Memory
RFD	Reduced Function Device
RFID	Radio Frequency Identification
RMII	Reduced Media Independent Interface
RSSI	Received Signal Strength Indicator
RTS	Request to Send
SAP	Service Access Point
SAR	Successive Approximation Register
SDHC	Secured Digital Host Controller
SHR	Synchronization Header
SMA	Sub Miniature version A
SPI	Serial Peripheral Interface
SRDY	Slave Ready
SREQ	Synchronous Request
SS	Station Services
TCP	Transmission Control Protocol
TFT	Thin Film Type

UART	Universal Asynchronous Receiver Transmitter
UDP	User Datagram Protocol
USB	Universal Serial Bus
VID	Vendor Identification
WEP	Wired Equivalent Privacy
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network
ZCL	ZigBee Cluster Library
ZDO	ZigBee Device Object
ZNP	ZigBee Network Processor

Chapter 1. INTRODUCTION TO STATUS MONITORING IN RAILROADS

1.1. Importance of Freight Railroad

Freight railroad is very important to the economic health of a country as it moves commodities between source and the market place, between the place of origin and the place where those commodities are used as raw materials. Hence freight railroads directly help businesses and industries thrive which in turn drives the economy of a country.

In the United States of America freight railroad accounts for 40% of the total freight volume [1], when measured in ton-miles. Ton-mile is defined as the movement of one ton of freight over a distance of one mile. The commodities moved by freight rail vary from coal, chemicals, food and related products, automobiles and their related products, lumber and wood products, minerals, metallic ores, petroleum and coal and other miscellaneous products [2-3]. Figure 1 and 2 illustrate this distribution.

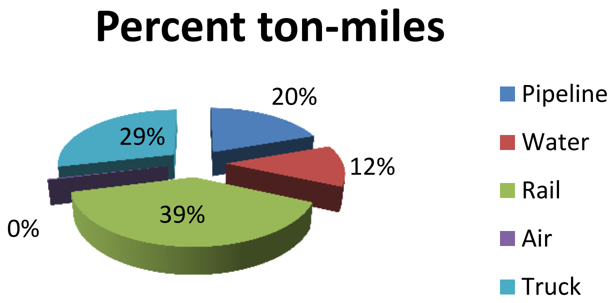


Figure 1.1. Percent in ton-miles of freight carried by different modes of transport

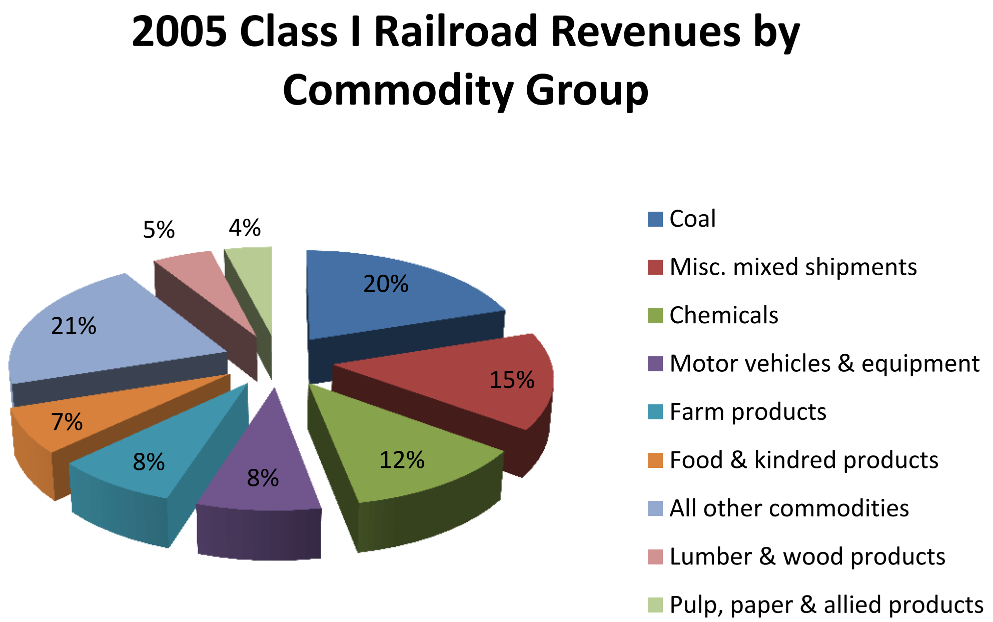


Figure 1.2. 2005 Class I Railroads Revenues by Commodity Group

There are several advantages of using freight railroad as well. They are presented as under:

a) A major advantage of freight rail is that it is fuel efficient. Studies have found that freight rail is as much as three times as fuel efficient as trucks, the other major mode of freight movement in the United States.

b) The movement of freight by rail is environment friendly. In today's day when the adverse effects of pollution on the environment is all too clear, emphasis is on clean modes of transport. Environment Protection Agency (EPA) estimates that freight rail produces only 9 percent of total transportation related nitrous emissions and 4 percent of total transportation related particulate emissions although it accounts for nearly 40 percent of inter-city ton-miles [2].

c) Freight railroad also helps take trucks off the road. It is estimated that a single freight rail takes 280 – 550 trucks off the road [1]. This contributes significantly to the alleviation of congestion from the road network.

d) Freight railroad is the safest way to carry hazardous materials. This stems from the fact that even though both trucks and rail carry hazardous material, the chances of a truck getting into an accident and releasing hazardous material is far higher than when using freight rail.

1.2. Reasons for Monitoring Freight Railroad

Each freight railcar consists of hundreds of moving mechanical parts, electrical wirings, goods and commodities that are prone to damage and maybe perishable.

The bearing in the wheels of a railcar may overheat and lead to a catastrophic failure and derailment. Electrical systems may fail leading to shut down of a refrigeration unit that would spoil the perishable goods that railcar was carrying. Doors of the railcar may be opened with the intention of stealing merchandise.

Any of these result in huge losses in the form of time and money for both the railroads and the businesses which depend on the goods transported by the railroads.

Each freight rail consists of close to hundred or more railcars. It becomes very difficult for the limited number of personnel on-board to monitor all the different parameter for each and every railcar.

Hence the ability to monitor status of various components and goods in a railcar becomes of critical importance. Also such monitoring can be used to look at and evaluate trends that can be used to upgrade maintenance schedules and put best practices in place.

1.3. Existing Methods for Railcar Status Monitoring

Currently employed methods of monitoring railcar status can be classified as the following:

a) **Wayside Monitoring:** In wayside monitoring certain parameters of a railcar are logged when it passes by the detector. The detector is generally placed outside stations or yards. They are generally used to monitor bearing temperature which is an indicator of bearing health. The data from the detectors are transmitted via backhaul networks to controlling stations where these are monitored and logged. Figure 3 shows a wayside monitoring installation.

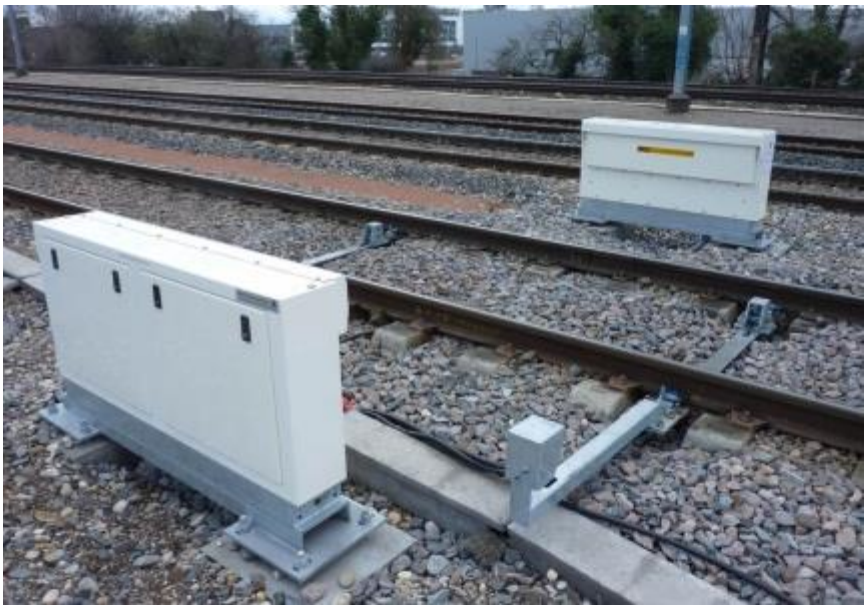


Figure 1.3: Wayside monitoring equipment in the field [5]

b) **Automatic Equipment Identification (AEI):** This form of monitoring is used just to identify the railcar or rail equipment that passed by the detector. It makes use of passive tags that are mounted on the side of railcars or on the rail equipment that is read by a reader mounted by the trackside. This technology makes use of Radio Frequency Identification technology [6] (RFID). Figure 4 shows an AEI setup in the field.

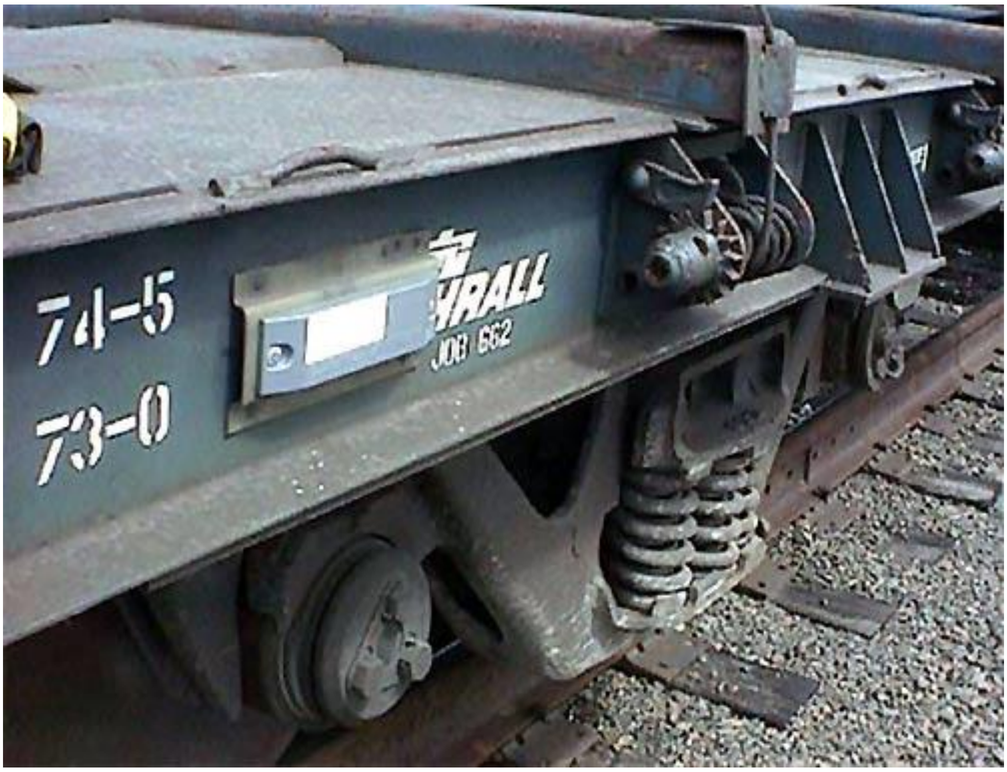


Figure 1.4: A railcar with an AEI tag mounted on its side [7]

c) On-Board Monitoring and Control System (OBMCS): This system puts the sensing equipment on-board the railcar. It has two distinct components. The sensors on-board a single railcar are interconnected by a CAN bus. While intra-train communication and control from the locomotive is achieved using IEEE 802.11(b) [8]. The system is also equipped with cellular radio and GPS for transmission of information to remote monitoring stations and location services. The system details are present in [9].

1.4. Drawbacks of Current Methods

The methods of railcar status monitoring mentioned briefly above suffer from some severe drawbacks. They are shortly discussed below:

a) Lack of real-time monitoring is a severe drawback plaguing the wayside monitoring technique. The wayside installations are spaced far apart from one another. The distance between two stations or yards may be in the hundreds of miles. Bearing failures are often catastrophic and happen very quickly. There is a good chance that it may take place between two stations. Also failure of the refrigeration unit, opening and closing of doors are events that cannot be effectively monitored by wayside installations.

b) The meager number of parameters that can be monitored by a wayside installation is also a noticeable drawback. These installations can measure bearing temperature and some very limited other parameters when a railcar passes by.

c) The OBMCS addresses these problems but introduces some unique to it. The use of a wired CAN bus within a railcar fixes the location of the sensors and also makes it hard to introduce a new sensor later on. The use of cellular radio is possible only in areas covered by cellular network. There are vast stretches of land, through which trains need to pass, where there is no cellular coverage. In those areas this system becomes non real-time. Also, the use of IEEE 802.11(b) [8], cellular radio and GPS [10] make the system power hungry. The use of dynamo to generate power to recharge the batteries needs invasive installation on the wheels.

1.5. WSN for Railcar Status Monitoring

Wireless Sensor Networks (WSN) involves sensor nodes that are capable of communicating with other sensor nodes and forming a network among themselves [11]. These nodes are capable of monitoring parameters and then communicating the readings through the network to the main collecting unit. Since the nodes are extremely resource constrained so the design of sensor nodes and network protocols are all guided by the principle of low power consumption. The continuing advancement in silicon science enables reduction of size of the sensor nodes which leads to lowering of cost. WSN has been successfully used in varied application environments as follows:

a) Area Monitoring

b) Environmental monitoring of forests, glaciers, air quality etc.

c) Industrial monitoring of machine health, data logging etc.

d) Agricultural monitoring of crops, soil humidity, cattle etc.

e) Smart home monitoring.

The deployment of WSN for railcar status monitoring is hence very feasible. It allows for real-time or near real-time monitoring. The absence of wires and self-forming and self-healing networks make the addition of new sensors very easy. Low power consumption profile increases the overall lifetime of the network and also since these can easily be battery powered, invasive installation is not required.

Hence the use of WSN for railcar status monitoring has significant advantages over the currently used methods of monitoring.

Chapter 2. OVERVIEW OF COMMUNICATION PROTOCOLS

This chapter will provide a brief overview of the involved communication protocols, namely IEEE 802.11, IEEE 802.15.4 and ZigBee.

2.1. IEEE 802.11

IEEE 802.11 [12] consists of a group of standards that are used to implement Wireless Local Area Network (WLAN) communication in the 2.4, 3.6, 5.0 and 60.0 GHz frequency bands. In 1999, the Wi-Fi Alliance [13] was formed which is a body that evaluates and certifies a product as Wi-Fi compliant. Wi-Fi is the name that has come to be commonly associated with products that conform to the IEEE 802.11 group of standards.

The following table shows the evolution of the IEEE 802.11 standard over the years with respect to a few parameters of interest:

802.11 Protocol	**Release Year**	**Frequency (GHz)**	**Bandwidth (MHz)**	**Data Rate (Mbps)**	**Modulation**
Initial	1997	2.4	20	1,2	DSSS, FHSS
a	1999	5.0	20	6,9,12,18,24 36,48,54	OFDM
b	1999	2.4	20	1,2,5.5,11	DSSS
g	2003	2.4	20	6,9,12,18,24 ,36,48,54	DSSS, OFDM

n	2009	2.4, 5	20	7.2, 14.4, 21.7, 28.9, 43.3, 57.8, 65, 72.2	OFDM
			40	15, 30, 45, 60, 90, 120, 135, 150	

Table 2.1: IEEE 802.11 releases over the years [8]

2.1.1. IEEE 802.11 Components

The IEEE 802.11 typical network architecture consists of the following components:

a) **Basic Service Set**: The IEEE 802.11 network is envisioned to consist of several interconnected cells. Each cell is called the Basic Service Set.

b) **Access Point:** Each Basic Service Set is to contain an Access Point or Base Station which controls the network within that cell.

c) **Distribution System:** If the WLAN consists of several cells, then the APs are connected by either a wired or wireless backbone. This is known as the distribution system.

d) **Extended Service Set:** The entire network consisting of the Basic Service Sets, APs and distribution system, viewed from the higher layers of the OSI-ISO stack appear as a single entity known as the Extended Service Set.

e) **Portal:** This is defined as a device which connects the 802.11 LAN to any other 802 LAN.

Figure 5 shows the conceptual diagram of an 802.11 WLAN consisting of most of the above components.

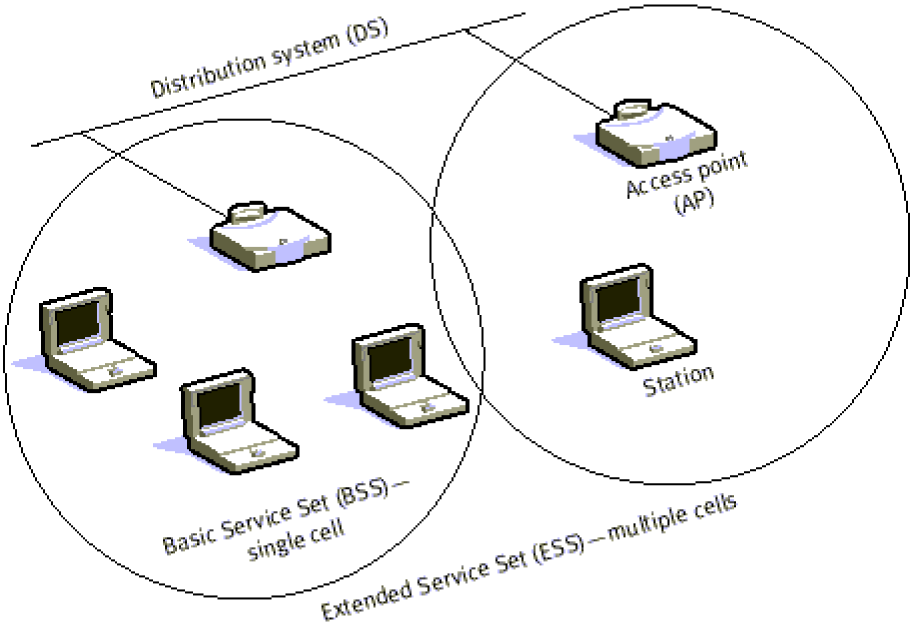


Figure 2.1: The IEEE 802.11 Infrastructure based network architecture [14]

2.1.2. IEEE 802.11 Network Types

The IEEE 802.11 standards provide for two distinct of network modes. The two modes are discussed as follows:

a) **Infrastructure mode:** This type of network depends on the cellular architecture and the presence of an Access Point through which all subscriber stations within a cell connect to the network. Figure 5 is an ideal representation of this type of network. The Distribution System or DS in an infrastructure based network needs to provide Distribution system services

(DSS) and Station services (SS). Distribution system services are related with node mobility and include Association, Re-association, Disassociation, Distribution and Integration. The Station services include Authentication, De-authentication, Privacy and MAC Service Data Unit (MSDU) delivery.

b) **Ad-hoc mode:** In this mode there is no Access Point present. There are just nodes which form a network among themselves. The services that are present in the Infrastructure mode are taken up by end-user stations, like beacon generation. Some functions, like frame relaying and power saving, available in the Infrastructure mode are not available in the ad-hoc mode. Figure 6 shows the layout of a typical ad-hoc network.

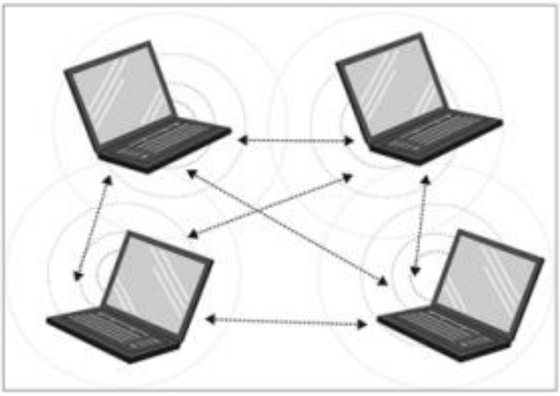


Figure 2.2: A few laptops in an ad-hoc IEEE 802.11 network [15]

2.1.3. IEEE 802.11 Protocol Stack

Figure 7 shows the IEEE 802.11 protocol stack showing the components of the Physical and the Data Link Layers.

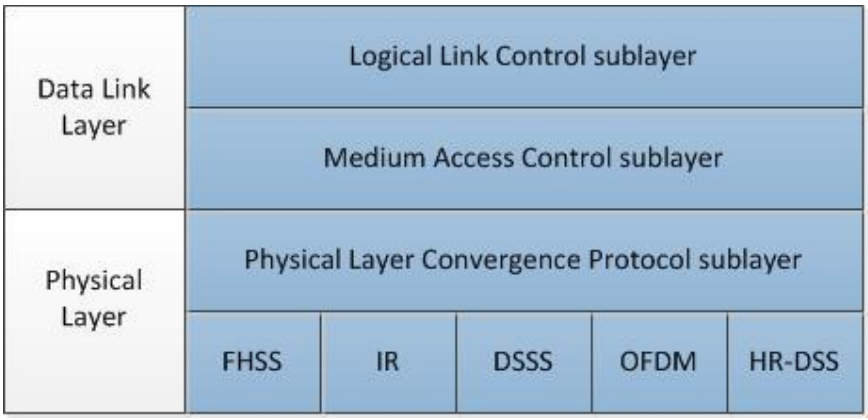

Data Link Layer
Logical Link Control sublayer
Medium Access Control sublayer
Physical Layer
Physical Layer Convergence Protocol sublayer
FHSS
IR
DSSS
OFDM
HR-DSS

Figure 2.3: IEEE 802.11 Protocol Stack [16]

2.1.3.1. IEEE 802.11 Physical Layer

The original IEEE 802.11 standard defined communication at 1 or 2 Mbps using Frequency Hopped Spread Spectrum (FHSS) or Direct Sequence Spread Spectrum (DSSS). Later on Orthogonal Frequency Division Multiplexing (OFDM) and High Rate-Direct Sequence Spread Spectrum (HR-DSSS) were introduced to support higher data rates.

When the FHSS technique is used the 2.4 GHz band is divided into 75 1 MHz sub-channels. Each pair of sender and receiver agrees on a hopping pattern and conversation between them is carried out with communication hopping between sub-channels according to the agreed hopping pattern. Each pair also chooses the hopping pattern in a way that interference is minimized. FHSS technique has the following characteristics:

a) Radio design is simple.

b) Data rate limited to 2 Mbps due to sub-channel restriction to 1 MHz by FCC.

c) Frequent hopping leads to high hopping overhead.

The use of DSSS technique allows the use of 13 22 MHz sub-channels in the 2.4 GHz band. All of these bands are partially overlapping with one another. Hence at any time there are just 3 sub-channels which are completely overlap free and communication is carried over any one of these channels. DSSS technique has the following characteristics:

a) Data rate is no longer limited to 2 Mbps.

b) Hopping is not used.

c) Chipping is used to increase immunity to noise. Each bit in the transmission sequence is converted into a unique series of bits called a chip and the entire signal is spread over the 22 MHz band to reduce the effect of noise and also build is redundancy that can be used for error correction. This leads to fewer retransmissions.

Figure 8 shows the utilization of the 2.4 GHz band using DSSS technique.

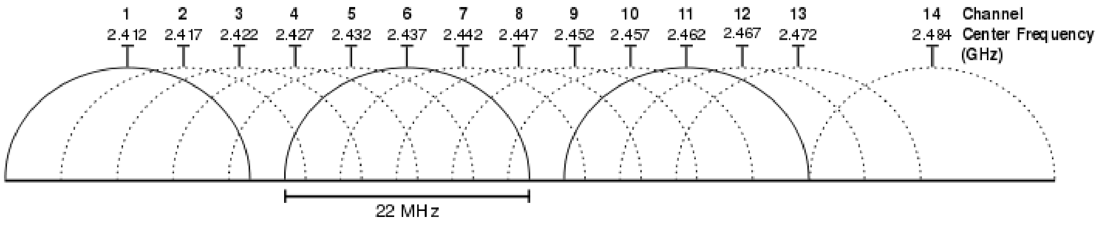


Figure 2.4: Channel assignment in the 2.4 GHz band for IEEE 802.11 [8]

2.1.3.2. IEEE 802.11 Data Link Layer

The IEEE 802.11 Data Link Layer consists of the Logical Link Control (LLC) sub-layer and the Medium Access Control (MAC) sub-layer.

The LLC is the same for IEEE 802.11 as it is for any other 802 communication protocol [17].

The MAC is unique to the IEEE 802.11. It provides the following features:

a) **Distributed Coordination Function** (DCF) deals with the problem of collision when two or more stations try to simultaneously transmit over the same channel. Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) is used for this purpose. When a station has data to transmit, it senses the channel and waits for a randomly generated time interval, if it finds the channel free. At the end of this time period if the channel is still free then the transmission is made. The receiving station receives the packet and if the CRC is alright, sends an acknowledgement back to the sender. If the sender receives the acknowledgement transmission terminates successfully. If the packet was lost or received in error or the acknowledgement was lost, the sender waits for some time and sends the packet again for a specified number of retries.

b) **Hidden node problem:** Often times there may arise a situation in which both the sending and receiving stations are visible to the Access Point but are outside of range from each other. In these cases the sender sends a Request to Send (RTS) to the access point. Since the Access Point is visible to all nodes in the cell, a Clear To Send (CTS) is only sent to the sender when there is no chance of collision happening during the ensuing transmission. This way the hidden terminal problem is tackled.

c) **Fragmentation and Reassembly:** Large packets are always more susceptible to corruption by noise and other inherent disturbances in the wireless channel. The IEEE 802.11 MAC layer provides for breaking a large packet into smaller chunks each framed by a MAC header and CRC for better reliability in transmission. On the receiving end it is the responsibility of the MAC layer for reassembling the fragments before passing them onto a higher layer.

2.1.4. IEEE 802.11 Security

Transmission security is one of the foremost concerns in any deployment of the IEEE 802.11 networks. Since it is over-the-air the ease of eavesdropping is much easier than in wired networks. The IEEE 802.11 standard defines a Wired Equivalent Privacy (WEP) for the security of communication. This has been shown to be very weak in the face of sophisticated attacks [18-20].

In summary, IEEE 802.11 provides for a high data rate long range communication protocol. The finer details of the protocol and its various implementation details can be found in [12].

2.2. IEEE 802.15.4

IEEE 802.15.4 [21] standard specifies the physical layer and the medium access layer for Low Rate-Wireless Personal Area Networks (LR-WPAN). It is designed for networks where communication is within devices close by and infrastructure for network formation and maintenance is virtually absent. It focusses on low power consumption, low data rate and small area of coverage. It is in sharp contrast to IEEE 802.11 or Wi-Fi where the networks deployed are principally infrastructure based and power consumption is not an issue but end-user experience is [22].

IEEE 802.15.4 was initially theorized to have a communication distance of 10 m and a maximum theoretical data rate of 250 kbps.

The two versions of IEEE 802.15.4 that are of interest in this research work and their brief introduction is provided in the table below:

IEEE 802.15.4 version	**Brief Introduction of features**
IEEE 802.15.4 – 2003	This was the first release of the IEEE 802.15.4 standard. It provided two physical layer bands, one in the 868 and 915 MHz and the other at 2.4 GHz.
IEEE 802.15.4 – 2006	This version of the standard upgraded the data rates possible in the 868 and 915 MHz bands. It also introduced new modulation schemes - three for the 868 and 915 MHz and one for the 2.4 GHz band.

Table 2.2: IEEE 802.15.4 versions [23]

2.2.1. LR-WPAN Device Architecture

The LR-WPAN device architecture is shown in figure 9 below. The IEEE 802.15.4 standard specifies the physical and the MAC layers only. The upper layers have to be implemented according to usage scenarios.

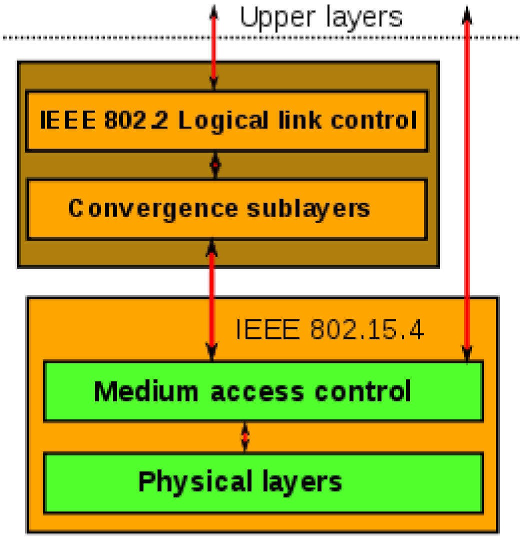


Figure 2.5: LR-WPAN Device Architecture [24]

The details of the layers and their interactions are provided in [21,25].

There are some popular upper layer implementations using IEEE 802.15.4 as the base and they include, but are not limited to, ZigBee [26], 6LowPAN [27], ISA-100.11a [28] and MiWi [29].

2.2.2. IEEE 802.15.4 Physical Layer

The physical layer of IEEE 802.15.4 specifies the radio communication capabilities of the protocol. The standard specifies several frequency ranges of operation, data rates and modulation techniques which are tabulated in the following table:

Frequency (MHz)	**Available Channels**	**Bit Rate (kbps)**	**Symbol Rate (kbaud)**	**Modulation Scheme**	**Receiver Sensitivity (dBm)**
868 – 868.6	1	20	20	BPSK	-92
902 - 928	10	40	40	BPSK	-92
868 – 868.6	1	250	12.5	ASK	-92
902 - 928	10	250	50	ASK	-92
868 – 868.6	1	100	25	O-QPSK	-92
902 - 928	10	250	62.5	O-QPSK	-92
2400 – 2483.5	16	250	62.5	O-QPSK	-86

Table 2.3: IEEE 802.15.4 Frequency bands of operation [30]

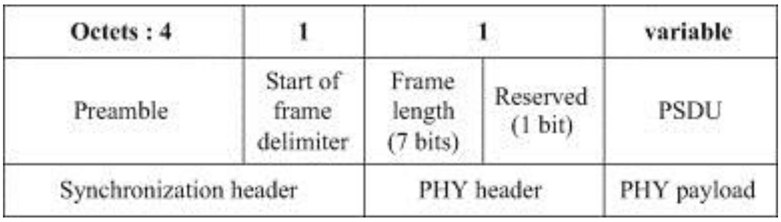
The physical layer of IEEE 802.15.4 provides two distinct services:

a) Data transmission and reception using physical layer protocol data units (PPDU).

b) Interface to the physical layer management entity which provides access to the upper layers as well as maintains information about the WPAN.

Direct Sequence Spread Spectrum technique is used for transmission which enhances the robustness of the communication in a non-perfect channel.

The data exchange is carried through using PPDUs the format of which is presented in the figure below:



Octets : 4	1	1		variable
Preamble	Start of frame delimiter	Frame length (7 bits)	Reserved (1 bit)	PSDU
Synchronization header		PHY header		PHY payload

Figure 2.6: Format of a IEEE 802.15.4 PPDU [21]

The SHR is used by the receiving device for synchronization, the PHR contains the frame length information and the PHY payload is variable length and contains the packet that is handed down from the IEEE 802.15.4 MAC layer.

The management functions that the IEEE 802.15.4 physical layer carries out have been listed below:

a) **Receiver Energy Detection (ED):** This is used by the network layer as a part of its channel selection algorithm.

b) **Link Quality Indication (LQI):** This is a representation of the strength and quality of the received packet. This is passed on to the higher layers and its use is completely dependent on the higher layers.

c) **Clear Channel Assessment:** Clear channel assessment is done using any one of the three methods outlined hereafter. If the channel sense returns energy above ED threshold then the channel is deemed to be busy. If on sensing the channel a carrier with IEEE 802.15.4 physical layer characteristics and spreading profile is detected then the channel is deemed to be busy. The last method is to use a combination of carrier sense and also detecting whether that energy is above ED threshold, to decide if the channel is busy or not.

2.2.3. IEEE 802.15.4 MAC layer

The IEEE 802.15.4 MAC layer provides the interface between the physical layer and the upper layers. Since the upper layers are not defined as part of the standard, they can be any of the implementations, like ZigBee, 6LoWPAN, mentioned before. The IEEE 802.15.4 MAC layer provides the following services [21]:

a) **MAC Data Services:** The entity known as MAC Common Port Layer (MCPS) provides data transmission and reception services between peer MAC layers.

b) **MAC Management Services:** The MAC Layer Management Entity (MLME) provides the interface through which layer management functions are accessed. It also maintains a database of objects for the MAC layer known as the Personal Area Network (PAN) information database. The MLME also has access to the MCPS for data transfer activities.

In general the IEEE 802.15.4 MAC layer is involved in the following network functions:

a) Beacon management.

b) Channel access.

c) Guaranteed-to-Send (GTS) management.

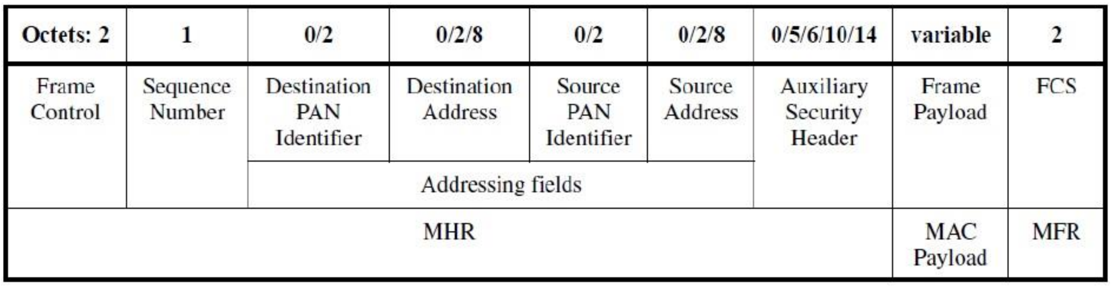
d) Frame validation.

e) Acknowledgement frame delivery.

f) Association

g) Dissociation.

The general MAC layer frame format is presented below and a short discussion is given thereon about the various fields within the frame.



<table>
<tr><th>Octets: 2</th><th>1</th><th>0/2</th><th>0/2/8</th><th>0/2</th><th>0/2/8</th><th>0/5/6/10/14</th><th>variable</th><th>2</th></tr>
<tr><td rowspan="2">Frame Control</td><td rowspan="2">Sequence Number</td><td>Destination PAN Identifier</td><td>Destination Address</td><td>Source PAN Identifier</td><td>Source Address</td><td rowspan="2">Auxiliary Security Header</td><td rowspan="2">Frame Payload</td><td rowspan="2">FCS</td></tr>
<tr><td colspan="4">Addressing fields</td></tr>
<tr><td colspan="7">MHR</td><td>MAC Payload</td><td>MFR</td></tr>
</table>

Figure 2.7: IEEE 802.15.4 MAC Frame format [21]

The MAC frame consists of the MAC header (MHR), a variable MAC payload and a Frame Check Sequence (FCS).

The MAC header consists of information needed for proper traversal of the data payload through the physical network to the intended receiver device. It consists of the following fields:

a) **Frame Control**: A 2 bytes long frame control field contains information related to frame type, addressing fields and other control flags. This will be discussed in details subsequently.

b) **Sequence number:** A 1 byte long sequence number gives an identifier to the frame that will be used by the destination node to generate an acknowledgement frame with the same sequence number.

c) **Destination PAN identifier:** A 2 bytes long field, when present, indicates a unique PAN identifier of the destination node.

d) **Destination address:** A 2 or 8 bytes long field, when present, that specifies the address of the intended recipient of the frame.

e) **Source PAN identifier**: A 2 bytes long field, when present, that specifies the unique PAN identifier of the origin of the frame.

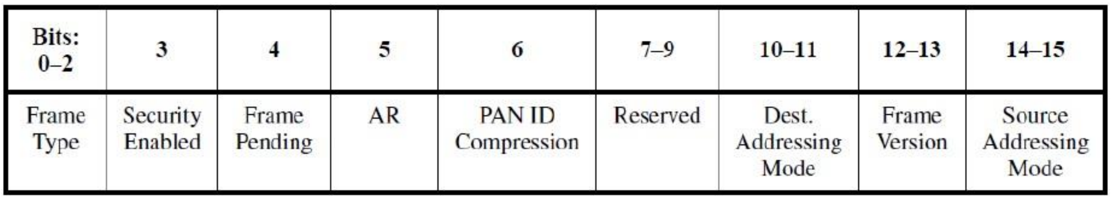
f) **Source address:** A 2 or 8 bytes long field, when present, that specifies the address of the origin of the frame.

g) **Auxiliary Security Header:** This field provides all the information necessary for security processing of the frame.

h) **Frame payload:** This field contains the data to be transmitted as handed down by the upper layers at the transmitter. If security is enabled then this field may be cryptographically protected.

i) **FCS field**: This field is a 16-bit ITU-CRC that is calculated over the entire MHR and MAC payload fields.

The structure of the Frame control field is shown in the figure below and its various bits and their functions are discussed.



Bits: 0–2	3	4	5	6	7–9	10–11	12–13	14–15
Frame Type	Security Enabled	Frame Pending	AR	PAN ID Compression	Reserved	Dest. Addressing Mode	Frame Version	Source Addressing Mode

Figure 2.8: IEEE 802.l5.4 MAC Frame control field [21]

The Frame control field bits are used as follows:

a) **Frame type**: Bits 0 and 1 are used to indicate the frame type. There are four types of frames namely Beacon, Data, Acknowledgement and MAC command.

b) **Security enabled**: This field if set to 1 indicates that the data in the payload will be encrypted and the auxiliary security header will be present in the MAC frame.

c) **Frame pending**: This bit will be set to 1 if the transmitter of the frame has more frames to send after the current one.

d) **Acknowledgment request**: This bit when set to 1 indicates that the recipient of the data or MAC command frame needs to acknowledge the receipt. An acknowledgement frame with the appropriate sequence number is sent back in that case.

e) **PAN ID compression field**: This bit indicates whether only one of the PAN identifiers will be present in the frame even though both source and destination addresses are present. If this field is 1 then when both source and destination addresses are present, only the destination PAN identifier will be present and the source PAN identifier will be assumed to be equal to the destination PAN identifier. If this field is set to 0, then the PAN identifier will be present only if the corresponding address fields are present.

f) **Destination and Source addressing modes**: These fields can be set to any one of the following values:

- PAN identifier and address fields are not present (00)
- Address field contains a short address (10)
- Address field contains an extended address (11)
- Reserved (01)

g) **Frame version**: The presence of a 0 in this field indicates that the frame is compatible with IEEE 802.15.4-2003 and a value of 1 indicates just an IEEE 802.15.4 frame. All other values are reserved.

The transfer of data between devices using IEEE 802.15.4 takes place in one of two modes. These are known as the Beacon Enabled mode and the non-Beacon

Enabled mode. The beacon enabled mode uses the slotted CSMA/CA algorithm whereas the non-Beacon enabled mode uses an un-slotted CSMA/CA algorithm [21]. The sequence of data transfer operations using these two modes are presented in the figures below.

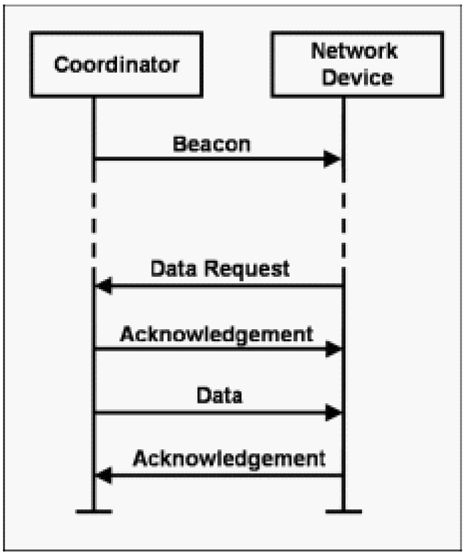


Figure 2.9: Sequence of steps when Coordinator has data to send to Network device in a beacon enabled mode [21]

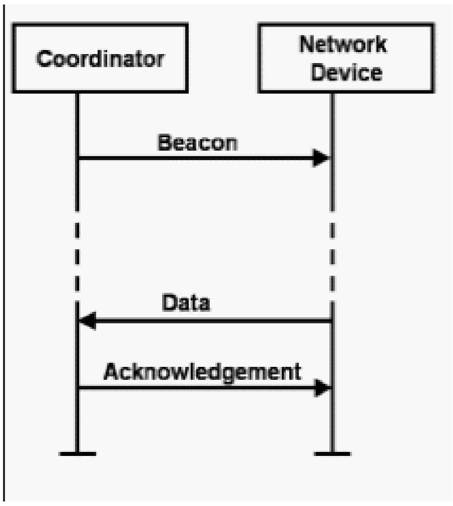


Figure 2.10: Sequence of steps when the Network device has data for Coordinator in a beacon enabled mode [21]

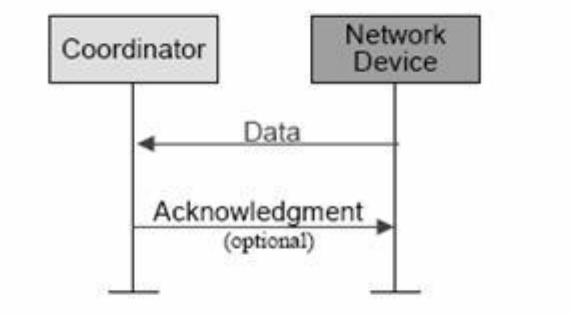


Figure 2.11: Sequence of steps for transfer of data in a non-beacon enabled network [21]

2.2.4. IEEE 802.15.4 Network Topologies

There are two types of network topologies that can be used within the IEEE 802.15.4 standard. These are briefly defined below [31]:

a) **Star Network**: In this topology there is a central PAN coordinator node and every other node communicates with this node.

b) **Peer-to-Peer Network**: In this topology there still exists the PAN coordinator node but the other nodes can also communicate between themselves without having to go through the coordinator node.

The two network topologies are represented in the figure below:

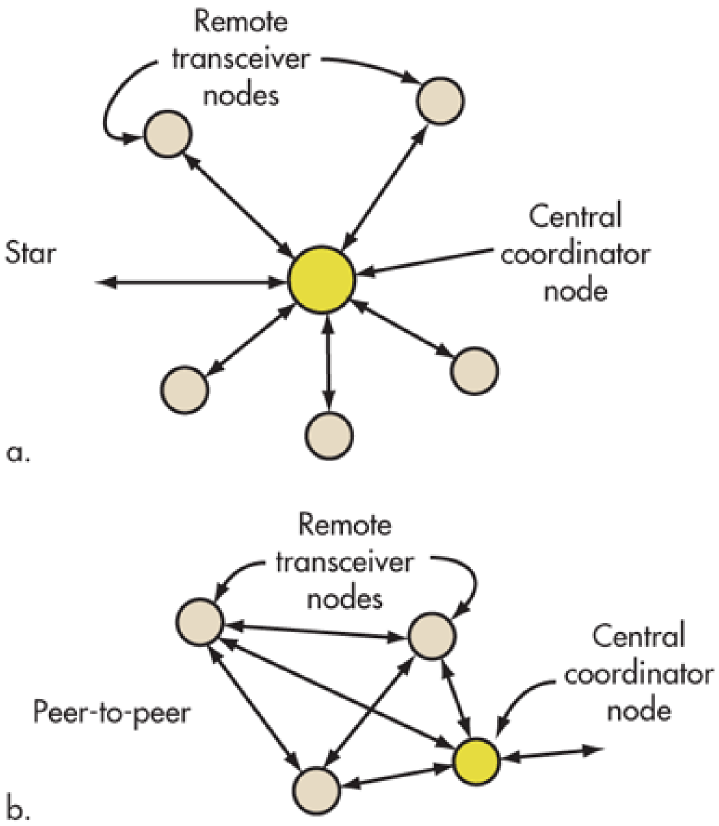


Figure 2.12: IEEE 802.15.4 network topologies a) Star and b) Peer-to-Peer

2.2.5. IEEE 802.15.4 Device Types

The different types of device classes that can exist within an IEEE 802.15.4 network are briefly presented below:

a) **Full Function Device (FFD):** These devices can send, receive as well as route data through them. They require more power to function.

b) **Reduced Function Devices (RFD)**: These are devices which can only send and receive data but cannot route traffic through them. They are generally end-devices in a network, typically a sensor or a switch. They can be very low power devices as they do not need to route traffic and hence can be out to sleep when not in operation.

c) **PAN Coordinator:** This is a special type of FFD which creates the network and also coordinates and manages it.

2.3. ZigBee

ZigBee is a standard which defines the network and application layer over the IEEE 802.15.4 physical and MAC layers for wireless sensor networks. It is maintained by ZigBee alliance, which an open, non-profit collaboration of hundreds of companies. Before any device is declared ZigBee compliant it has to undergo the ZigBee Certification Program [32]. ZigBee has the following characteristics:

a) Leads to low power consumption hence longer battery life.

b) Leads to small device footprint.

c) Allows mesh networking.

d) Self-forming, self-healing network.

e) Interoperability between devices from various vendors but conforming to the ZigBee standard.

f) Security mechanisms in place with 128 bit AES encryption possible.

In addition to the standard the ZigBee Alliance has defined many application profiles as well. Every data request in ZigBee is sent and received on an application profile. Some of the commonly used application profiles are Home Automation, Industrial Plant Monitoring, Commercial Building Automation, Telecom applications, Personal Home and Hospital care etc [33].

2.3.1. ZigBee Stack Architecture

The best way to understand how ZigBee builds on the IEEE 802.15.4 lower layers is by looking at the ZigBee stack architecture as provided in [33]. The same is replicated in the figure below:

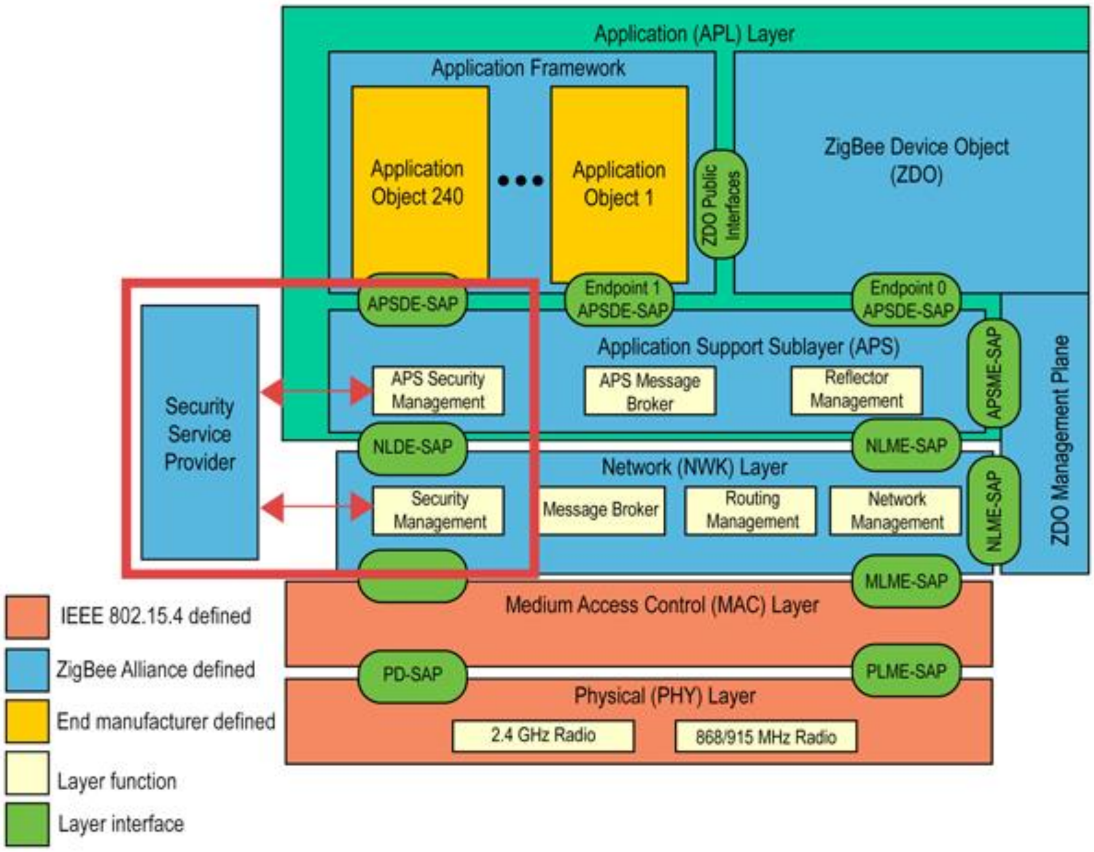


Figure 2.13: ZigBee stack architecture

Each component of the ZigBee standard will be briefly discussed in what follows

2.3.1.1. ZigBee Network Layer (NWK)

The ZigBee network layer resides just above the MAC layer as defined by IEEE 802.15.4 and is in charge of executing the following services [34]:

a) Routing of frames.

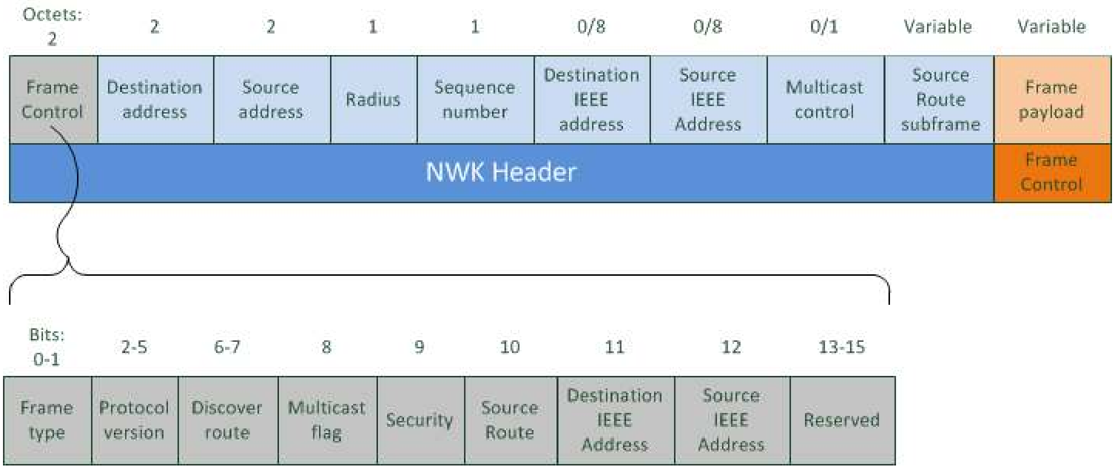
b) Discovering and maintaining route tables, one-hop neighbors and storing important information about neighbor devices.

c) Joining and Leaving of ZigBee network devices.

d) Providing cryptographic security to the transmitted frames.

The above functions are achieved by the NWK layer using the NWK Layer Data Entity (NLDE), used for data transmission via the NLDE-Service Access Point (NLDE-SAP), and the NWK Layer Management Entity (NLME) that manages the services via the NLME-Service Access Point (NLME-SAP).

The general NWK layer frame format is presented below and each of the fields in the frame are defined.



Octets: 2	2	2	1	1	0/8	0/8	0/1	Variable	Variable
Frame Control	Destination address	Source address	Radius	Sequence number	Destination IEEE address	Source IEEE Address	Multicast control	Source Route subframe	Frame payload
NWK Header									Frame Control

Bits: 0-1	2-5	6-7	8	9	10	11	12	13-15
Frame type	Protocol version	Discover route	Multicast flag	Security	Source Route	Destination IEEE Address	Source IEEE Address	Reserved

Figure 2.14: General ZigBee NWK Layer Frame Format [34]

The fields in the NWK Layer Frame are presented briefly below:

a) **Frame Control:** This field has information about how the frame is going to be treated. Each bit within the Frame control field has a specific meaning as follows:

- Frame type: This indicates the type of frame like NWK data or NWK command etc.
- Protocol Version: The value here indicates which version of ZigBee this frame is compliant with. For example, a value of 2 will indicate ZigBee 2006/2007/Pro.

- Discover Route: If this contains the value 1 then route discovery for this frame is enabled.
- Multicast Flag: If this bit is 0 then it indicates that the frame will be unicast to its destination.
- Security: If this bit is 0 then security for the frame is disabled and if it is 1 then security for the frame is enabled.
- Source Route: If this field is 0 then it means that the source route is not specified for the frame.
- Destination and Source IEEE addresses: A zero in each of these fields means that the frame does not contain the field marked with a zero. A 1 will indicate that the corresponding field is present in the frame.

b) **Destination Address:** This field contains the 16 bits long destination short address.

c) **Source Address:** This field contains the 16 bits long source short address.

d) **Radius:** This field contains the range of transmission as the maximum number of hops from source to destination. Each node on the way decrements this value by 1.

e) **Sequence number:** This is a number unique to the frame being sent and is used for proper re-assembly at the receiver end.

f) **Source and Destination IEEE Address:** If present these fields contain the 64 bits long source and destination IEEE addresses respectively.

g) **Multicast Control:** This field is present dependent on the Multicast flag in the Frame Control field.

h) **Source Route sub-frame**: This is present if the Source route flag in the Frame control field is set to 1. This indicates to the next hop node to check this sub-frame to determine the next hop in the route to the destination for this frame.

i) **Frame payload:** This contains the data that has been handed down from the higher layers.

2.3.1.2. ZigBee Application Layer

The application layer of ZigBee is composed of Application Framework (AF), ZigBee Device Object (ZDO) and Application Support Sub-layer (APS) [34].

The Application Support Sub-layer (APS) lies between the application and network layers and principally accomplish the following tasks:

a) It uses the APS Data Entity (APSDE) for exchange of data.

b) It uses the APS Management Entity (APSME) for exchange of management primitives.

c) The APS Information Base (AIB) contains constant and variable attributes.

d) When the APS receives an application payload, it adds the address and control fields, and passes the resulting payload to the lower layer.

e) APS frames are of the following types namely data, command, and acknowledge.

The Application Framework (AF) has the following characteristics:

a) Each device can have up to 240 application endpoints represented by a 8 bit address.

b) The application endpoint is used to transmit and receive data. It acts much like a port in TCP/IP communications.

c) Information generated by an application endpoint is called an attribute. A collection of attributes and commands working on the attributes is called a cluster. A collection of clusters is called the ZigBee Cluster Library (ZCL). A profile consists of a cluster, device ID and endpoint and together it forms an application. Each attribute, cluster and profile is represented by 16 bit IDs.

The ZigBee Device Object (ZDO) layer has the following functions to perform:

a) Service and device discovery.

b) Initialization of the coordinator.

c) Security management.

d) Binding management.

e) Network management.

The frame formats involved in the discussed layers and details about the layer functionalities can be found in [33].

2.3.2. ZigBee Device Types

ZigBee standard defines the following device types:

a) **Coordinator:** It is responsible for starting and maintaining the network. It is also responsible for assigning addresses to newly joined devices. It is a FFD as defined in the IEEE 802.15.4 standard.

b) **Routers:** These devices are used to route frames in the network and are likely to be always powered on. These are also FFDs as defined in the IEEE 802.15.4 standard.

c) **End-Devices:** These are the same as RFDs as defined in the IEEE 802.15.4 standard.

2.3.3. ZigBee Network Topologies

ZigBee standard defines the following network topologies for use:

a) Star topology: In this topology the central PAN coordinator node starts and maintains the network. Every other device can only communicate with and through the PAN coordinator.

b) Tree topology: In this topology beacon enabled mode is used in a hierarchical communication structure.

c) Mesh topology: In this topology peer-to-peer communication is allowed. This leads to a network that is more robust to failures because of the existence of redundant paths.

The Star and Mesh topologies are very much similar to the Star and Peer-to-Peer topology shown during the discussion of IEEE 802.15.4 topologies. The Tree topology introduced in ZigBee is shown in the figure below.

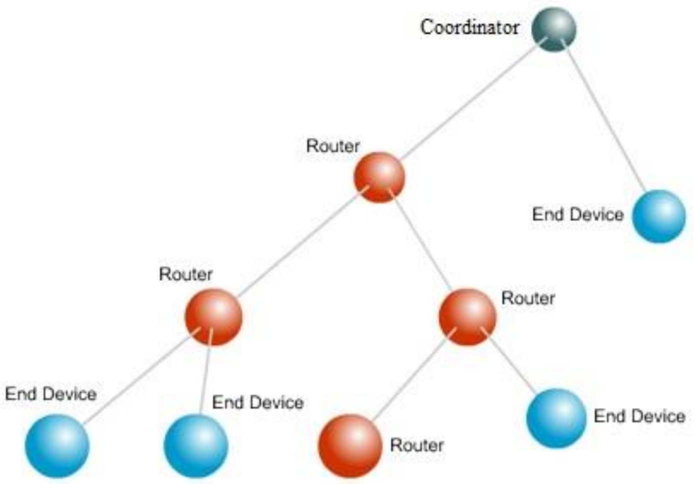


Figure 2.15: ZigBee Tree Topology [35]

Chapter 3. ISSUES WITH ZIGBEE IN RAILROAD ENVIRONMENTS

The North American Railroads are looking at the use of on-board real-time or near real-time monitoring of freight railcar status using WSN. ZigBee is the technology of choice for the railroads for its low power consumption and low cost of implementation.

In the railroad environment typically there will be at least hundred railcars strung together to form a freight rail. Each of the railcars will be equipped with several sensors. Let us assume that each railcar has only one sensor node. Then the data from each of these sensor nodes will have to travel hop-by-hop to the monitoring station in the locomotive. So the network topology that is most commonly found in the railroad environment is a long linear-chain like topology [36].

ZigBee was primarily designed for star topology [37]. Tests carried out at the Advanced Telecommunication Laboratory at the University of Nebraska, has shown that although ZigBee supports multi-hop communication but it is unsuitable for topologies involving as many hops as in the railroads environment [38 – 40].

Similar research studies carried out by others have also found similar issues and performance degradations with ZigBee [41] and also link layer performance bottlenecks as found in [42].

In the following pages we briefly revisit the main issues and challenges that were found to plague a ZigBee-only network in the railroads environment.

3.1. Route Discovery

A route in a network, wired or wireless, refers to the path that packets take from their origin to the intended destination. Each link from one node to its next communication partner is referred to as a hop. If a router has more than one hop, we refer to this as multi-hop routing [36].

ZigBee was designed for star topologies. In such a configuration, each node is in direct communication range of the PAN Coordinator. In this topology the hop count is at most 2 hops. To a limited degree, ZigBee also supports tree topologies, with the PAN Coordinator as the root node. However, it is severely restricted in the number of hops it supports for this topology. A hop count of 100 or more is far beyond the capability of ZigBee. In the chain-like topology we find on a freight train, ZigBee fails to discover routes after about 20 – 25 hops [36].

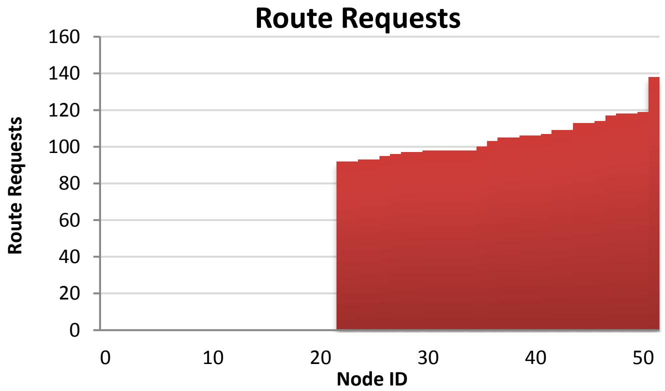


Figure 3.1: Number of unsuccessful route requests vs Node ID

3.2. Synchronization Delay

All nodes within a single PAN network are synchronized with each other. Each time the layout of the network changes due to node failure or disconnections, a resynchronization becomes necessary. This can also be triggered by fluctuating channel conditions that prevent a node from successfully communicating with the rest of the network for a period of time. Fluctuating channel conditions are a common occurrence in wireless networks. For a 100 railcar scenario, this synchronization can stretch to several minutes. During this time, no communication is possible between the unsynchronized nodes and the rest of the network [36].

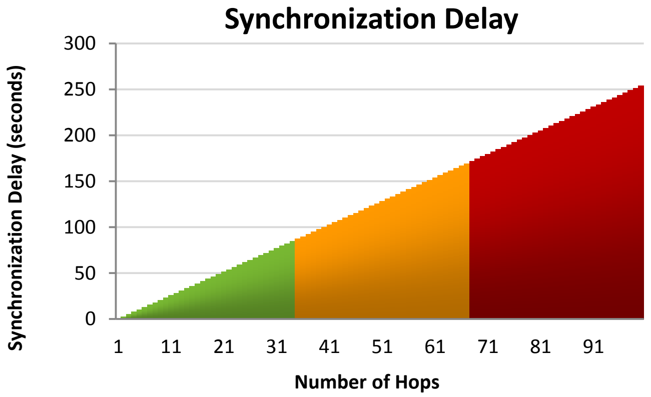


Figure 3.2: Time required for synchronization vs the maximum number of hops in network

3.3. Packet and Link Loss

If we study the behavior of wireless communication systems we can find that the probability of successful packet delivery decreases as the number of hops increases. As reported in [41], our analysis of this problem when applied to freight train monitoring using ZigBee. That analysis is shown in the following figure, demonstrating the end-to-end packet error rate for different hop counts and bit error rates. For this analysis we used a packet size of 127 bytes per packet [36].

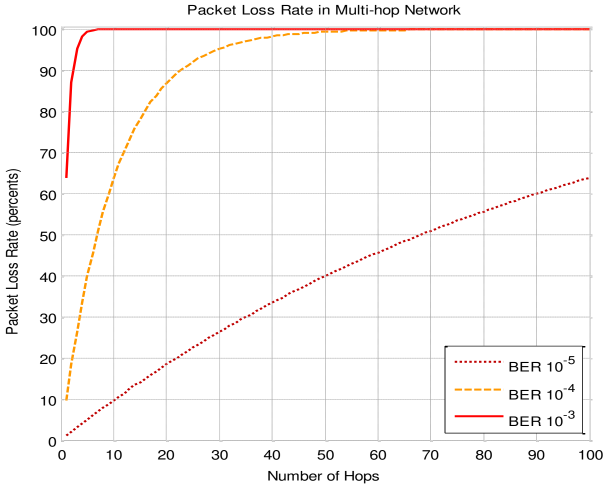


Figure 3.3: Packet Loss Rate percentage for different hop counts and Bit Error Rates

It can be seen, for a Bit Error Rate of 10^{-5}, which in wireless networks is a very low BER signifying good channel conditions, we still experience a packet loss of over 60% due to errors. Less than 40% of packets are being delivered successfully for a hop count of 100. Under more realistic conditions, with BERs between 10^{-3} and 10^{-4}, we can observe that none of the packets from a majority of nodes on the train will successfully be delivered to the locomotive. This means that the network will completely cease to function.

3.4. Lack of Quality-of-Service

Quality-of-Service refers to the ability of a network to allow transport of packets with special requirements, and appropriately allocate transmission resources to it. ZigBee does not have a way of assigning priorities to different packets.

This is extremely important in the freight train monitoring system. An important aspect of the envisioned system is the ability to generate alerts in case of equipment failure, the detection of dangerous substances or leaks, etc. But if these alert messages are treated the same way as a message containing periodic status updates from, say, a refrigeration unit, then it shares the same probability of being discarded by the network due to errors or congestion. It also will be entered into every queue at the end, like all other packets, and thus be significantly delayed. For alerting, this is simply not feasible [36].

3.5. Data Forwarding and Aggregation

A chain topology using ZigBee means that data from the last node needs to be relayed to the first node. In addition to relaying that information, however, each node also has its own data to report. Hence the amount of data quickly aggregates. By the time the nodes closest to the destination are reached the volume of data becomes very large, essentially overwhelming the network and causing significant congestion. This leads to excessively high packet loss. Simulation shows that if a freight train consists of more than 25 railcars then it cannot even sustain traffic of small 50-byte message per node generated every 20 seconds [36].

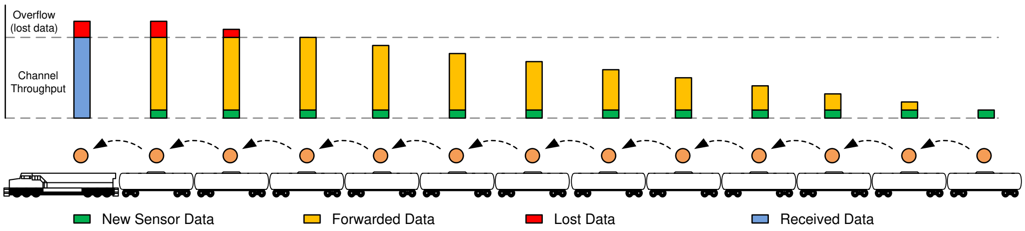


Figure 3.4: Amount of data present in channel as a progression of hops from the end of train

3.6. Network Lifetime

The nodes closest to the data destination have a lot more data to forward than nodes towards the end of the train. Hence, their increased number of operations also means that they consume a lot more energy and deplete their reserves much faster. Due to the point-of-failure issue, if one of those nodes fails, no data from behind that node will be able to reach the destination anymore. At this point, the network effectively fails to operate [36].

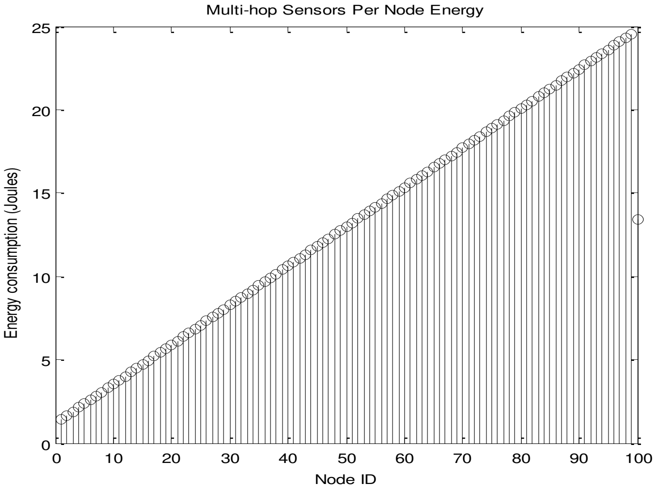


Figure 3.5: Energy consumption per node for a 100-railcar train

The battery life profile illustrates the claim that the nodes closer to the coordinator node run out of battery faster than the ones farther away from it. We can see that, although node 0 would be operational for around 2000 hours, the network will fail after less than 200 hours [36].

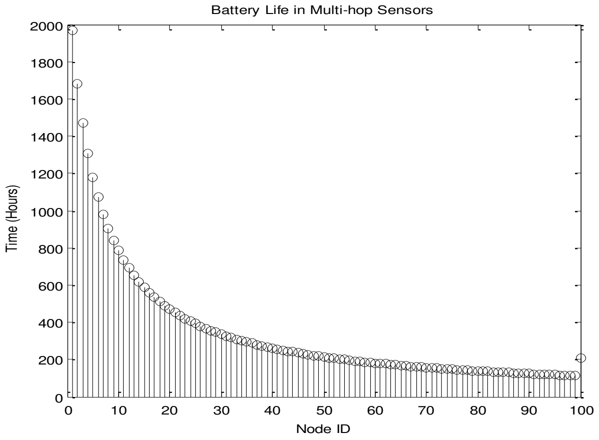


Figure 3.6: Battery life in hours for each node in a 100-railcar train

Chapter 4. OVERVIEW OF HYBRID TECHNOLOGY NETWORKING (HTN)

The Advanced Telecommunication Engineering Laboratory at the University of Nebraska-Lincoln has come up with a solution that addresses the issues that a ZigBee-only network faces in the railroads application environment. The solution is named Hybrid Technology Networking (HTN) protocol [43].

The basic idea behind the protocol envisions railcar monitoring as a two-step process.

In the first step clusters of sensors are formed spanning the length of the freight rail. Each of the clusters have only one coordinator node which is called the gateway. Each sensor node communicates via single or limited number of hops to the gateway. In the next step, the gateway aggregates the sensor data and relays it to the locomotive via Wi-Fi hops over similar gateways spanning the length of the freight rail [43].

The high-level overview of the Hybrid Technology Networking protocol is shown in the figure below:

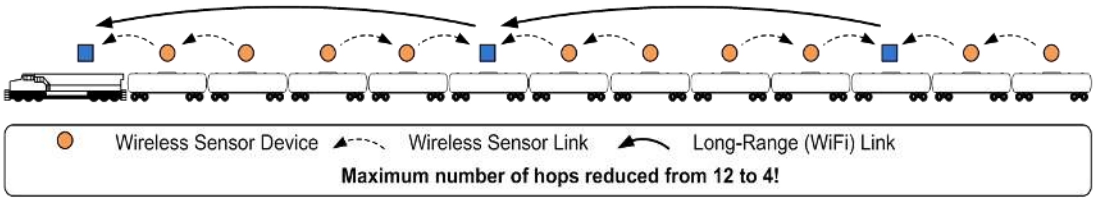


Figure 4.1: High-Level representation of Hybrid Technology Networking

Within the clusters ZigBee is used for communication. Since the clusters are small and of low depth the synchronization delay is drastically reduced.

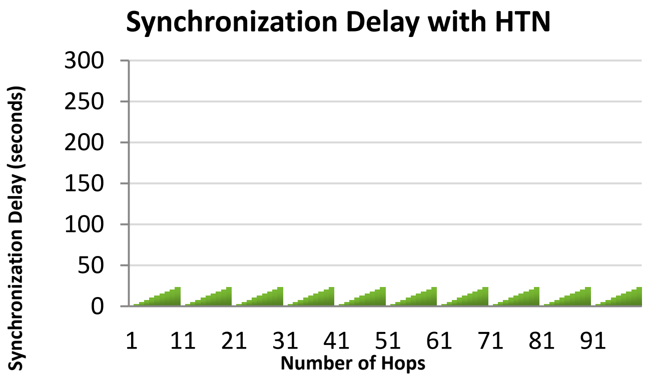


Figure 4.2: Time required for synchronization vs. the maximum number of hops in network in HTN

Since the ZigBee networks operate in small clusters the chance of the network getting overwhelmed due to data aggregation is also minuscule.

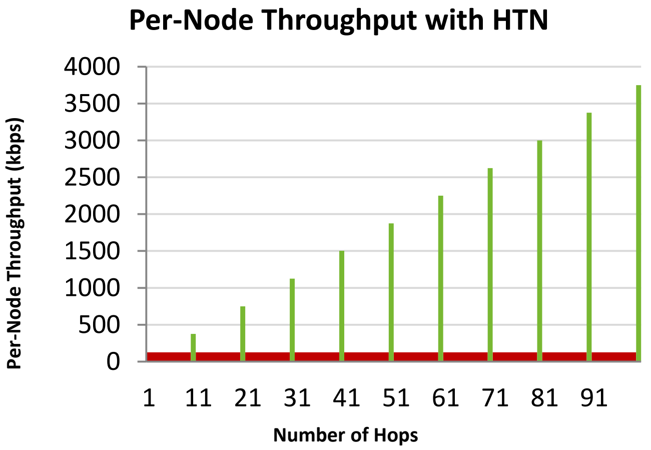


Figure 4.3: Per Node Throughput vs. the maximum number of hops in HTN

As can be seen in Figure 28 the per node throughput using ZigBee-only network is shown with the red bar, the improvement in throughput with HTN is clearly visible.

The protocol also suggests rotation of the gateway role among all the capable nodes in a cluster hence balancing the power consumption profile.

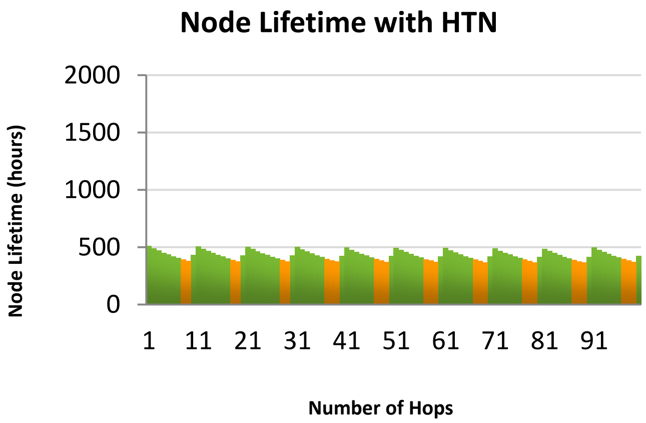


Figure 4.4: Node Lifetime vs. Total Number of Hops in HTN

The above simulation result clearly shows the improvement in network lifetime due to a balanced power consumption profile.

As mentioned in [43], simulation results with HTN shows that there are the following significant advantages over a ZigBee-only network in the railroads application environment:

a) 90% reduction is synchronization delay.

b) 40 times improvement in system throughput.

c) Increasing network lifetime by about 250%

d) Since HTN protocol sits on top of the underlying ZigBee and Wi-Fi protocols, an end to end Quality-of-Service mechanism can be implemented.

Chapter 5. PROBLEM STATEMENT

This chapter underscores the reason behind undertaking this research work.

As is evident from the discussions in chapter 3 the application of ZigBee-only networks to monitor railcar status is nowhere near an optimal solution. It suffers from severe drawbacks which defeat the very purpose of using WSN to monitor railcars.

A solution is presented in chapter 4 and it is seen that this solution addresses all of the issues faced by a ZigBee-only network in the application scenario. The performance improvements obtained from using this new protocol, called HTN protocol, is also evident from the simulation results presented by the authors.

The next step in exploring this solution is to implement it in hardware and actually deploy WSN using the HTN protocol and evaluate the results obtained from the field tests.

The first step towards this goal is to find a sensor platform that is capable of supporting HTN. The major components of such a sensor platform will be a Wi-Fi and a ZigBee radio available on the same platform.

The available sensor platforms like MICAz [44], Stargate [45] and recent development Libelium Waspmote [46] are all capable of providing the ZigBee communication part as well as the Wi-Fi communication part but not in an integrated solution. The current available sensor platforms will have to be plugged into one another to realize both the communication radios together.

Several technical and logistical problems arise from this approach. Some of these problems are listed below:

a) The data rate of plug-in boards will always be limited by the bus that is used to connect them. In all of the above cases, except the Stargate, the only available interfacing bus is the UART. This is slow speed and hence will cause a bottleneck for Wi-Fi traffic. In Stargate, the PCMCIA interface is used which removes the speed bottleneck but is an order of magnitude more power hungry than the UART.

b) The cost of obtaining various pieces of hardware from different vendors, assembling them and coding for the required firmware is high and is a logistical nightmare when the numbers of such platforms required are in millions.

c) The railroads are looking for standards compliant solution so that vendor independence is maintained. Some of the off-shelf solutions may not be standards compliant.

d) A custom hardware can be made with specific design goals in mind. The commercially available components were designed with some design goals and when they are brought together they may not meet the goals of the railroads.

As can be understood from the above discussion there are major advantages in designing a custom integrated hardware platform that will be standards compliant and also will meet and exceed any design goals that may be set by the railroads.

Chapter 6. HARDWARE DESIGN

The purpose of the hardware designed here-in is to provide an integrated platform for the implementation of the HTN protocol. Chapters 3 has shown the issues related with the use of ZigBee-only networks in the railroads scenario. Chapter 4 discusses about a solution that addresses the issues highlighted in chapter 3. In chapter 5 we provide the reasons why a new sensor hardware platform is necessary for the implementation of the HTN protocol.

In this chapter the hardware design has been presented in details. The operational requirements for the hardware have been discussed, followed by a general high level block diagram of the hardware. This is followed up by a discussion on each of the components involved in the design with respect to why they were chosen, how they are interfaced with the microcontroller etc. The design choices made during the design of the PCB have been discussed. Several snapshots of the hardware have been provided at the end of the section.

6.1. Requirements

The hardware design was guided by a set of requirements which originated from the overview of the HTN protocol and what it wishes to achieve.

The requirements are listed below:

a) The microcontroller that will form the core of the hardware platform needs to be fast as it has to deal with both ZigBee and Wi-Fi communication and also run a sizeable firmware involving many calculations. The microcontroller should be able to drive data transfer to and from ZigBee radio to Wi-Fi radio and vice-versa fast enough to ensure respectable throughput. At the same time the peripherals available with the microcontroller should be large. Keeping the overall low power budget in mind the microcontroller and its associated drive circuitry should not consume much power.

b) ZigBee radio hardware. The best option is to have an integrated solution having an on-board processor that will offload the ZigBee communication related tasks from the microcontroller. Needs to be able to transmit at a max of 3dBm output power and have a receive sensitivity of up to -87 dBm. The overall power consumption of this component should not exceed the 50 mA in any communication mode (Tx/Rx). Should operate in the 2.4 GHz band and have the flexibility to pick and cycle between the 16 available channels.

c) Wi-Fi radio hardware. The best option is to have an integrated solution having an on-board processor that will offload the TCP/IP stack functionality and radio interface tasks from the microcontroller. Should be low on power consumption, not exceeding 150 mA in any of the communication modes (Tx/Rx). Should be compliant to at least IEEE 802.11 a/b/g. Should support Ad-Hoc and infrastructure modes.

d) Since the HTN protocol mandates that the roles of the nodes may change depending on need, the nodes need to have some information stored on-board which facilitates this role change. Hence, a storage medium like SD Card is necessary.

e) The hardware must have capabilities to interface with a laptop or computer easily for diagnostics.

f) Since this hardware will need to interface with other hardware and sensor platforms, there must be multiple power supplies catering to various sensor platforms.

g) There must be debug interfaces on the hardware for initial development cycles.

h) The availability of development environment and APIs to reduce development time.

i) Low bill of materials.

6.2. Block Diagram of Design

The following figure shows the basic building blocks of the design. In later sub-sections the blocks are explained in more details.

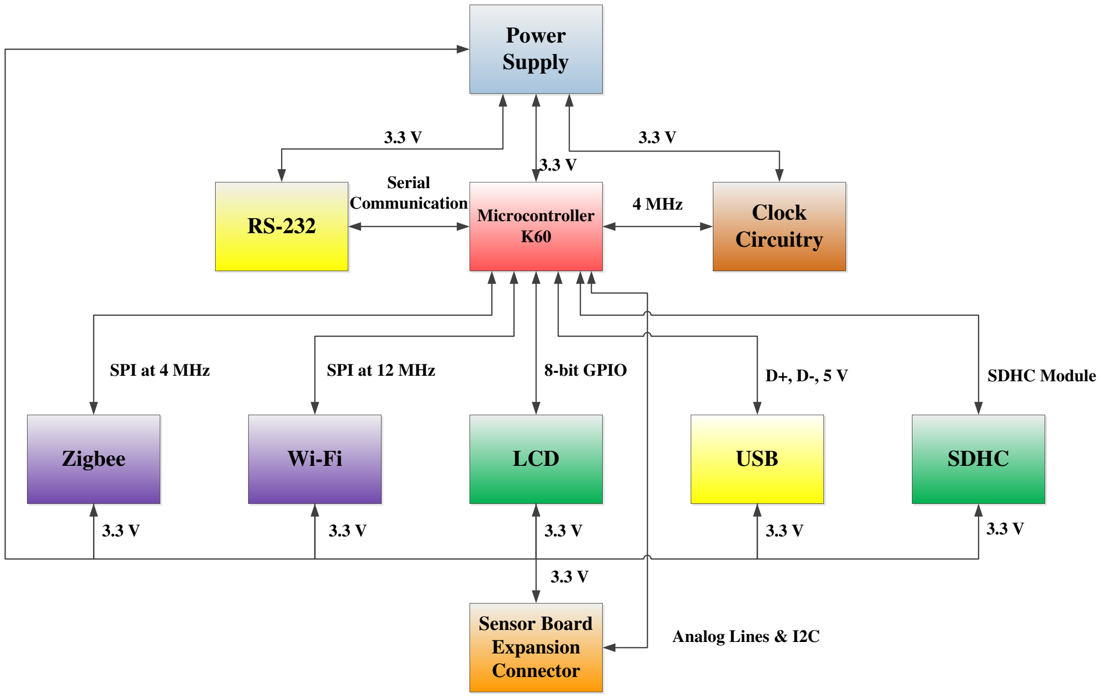


Figure 6.1: Hardware Block Diagram

6.3. Hardware Components

In the following each of the hardware components are introduced in detail.

6.3.1. Power Supply Circuitry

Figure A.1-4 in Appendix A shows the schematic of the power supply circuitry. In keeping with the requirement of interfacing platforms with this hardware that may need different levels of supply three different power supply options are provided. The board has 5 V, 3.3 V and 1.8 V options. 3.3 V is the main power supply voltage that runs all the peripherals on the board.

The LT1129-3.3 [47] IC is used to generate a 3.3 V from the 5 V supply. This chip consumes only 50 uA while operating and can deliver up to 700 mA of current at 3.3 V.

TPS73701 [48] IC is used to generate 1.8 V from 5 V supply. This chip consumes 0.3 mA while operating and can deliver up to 1 A of current at 1.8 V.

The supply of 5 V is taken from wall socket or USB and this is decided through a selector switch. The two regulator ICs can be individually disabled.

6.3.2. Clock Circuitry

Figure A.5 in Appendix A shows the schematic of the clock circuitry. The microcontroller runs off a 4 MHz external clock signal. The hardware board is given a choice of sources for this 4 MHz clock signal. Using selector switches one can use a 4 MHz crystal, a 4 MHz clock chip or source the 4 MHz from an external clock source.

This is done to give maximum flexibility to the source of the clock signal for the microcontroller.

The microcontroller also needs an external 32.768 kHz clock to operate.

An ECX-71 [49] tuning fork crystal is used for the 32.768 kHz clock source. Its load capacitance is 12.5 pF.

An ECS-3951 [50] SMD clock oscillator is used as one of the two 4 MHz clock sources for the microcontroller.

A Murata CSTCR4M00G53Z-R0 [51] ceramic resonator is used as the other source for the 4 MHz clock for the microcontroller.

6.3.3. Microcontroller

Figure A.6-7 of Appendix A shows the schematic for the microcontroller.

The microcontroller used for this hardware is the Freescale Kinetis K60 [52]. The features of this microcontroller are presented in the list below:

a) Operates at a voltage of 3.3 V in this case.

b) 100 MHz ARM-Cortex-M4 delivering up to 1.25 Dhrystone MIPS per MHz.

c) 512 KB flash memory and 128 KB RAM.

d) 10 different low power modes of operation.

e) Built in security modules for integrity of firmware once burnt in the microcontroller memory.

f) Two 16 bit SAR ADCs. Programmable Gain Amplifiers built in to the ADCs.

g) Two 12 bit DACs.

h) Three comparators containing a 6 bit DAC and a programmable input reference.

i) Several timer modules available including real time clock.

The communication modules available with this microcontroller are as follows:

a) Ethernet controller with MII and RMII interface to external PHY and hardware IEEE 1588 capability.

b) USB full-/low-speed On-the-Go controller with on-chip transceiver.

c) Two Controller Area Network (CAN) modules.

d) Three SPI modules.

e) Two I2C modules.

f) Six UART modules.

g) Secure Digital host controller (SDHC).

h) I2S module.

As can be seen from above, this microcontroller fits the requirements perfectly. The microcontroller has a maximum current consumption of 71 mA. The SPI interface is used to connect to the Wi-Fi and ZigBee modules. The SPI maximum speed is 12.5 MHz for the microcontroller. The abundance of analog lines on the microcontroller makes interfacing external sensors easy. The presence of I2C bus gives us the flexibility of interfacing digital I/O sensors as well. The integrated USB transceiver makes interfacing this microcontroller to a laptop or computer very easy.

The microcontroller is driven at a core clock speed of 96 MHz. The bus clock is 48 MHz. The USB module is driven at 48 MHz.

6.3.4. Wi-Fi Module

Figure A.8 in Appendix A shows the schematic of the Wi-Fi module.

The Wi-Fi module used in the design is an integrated solution with a microcontroller and RF front end built on-module. The module interfaces to the Kinetis microcontroller using the SPI bus at a SPI clock frequency of 12 MHz.

The Wi-Fi module used is the Redpine Signals module RS9110-N-11-22-05 [53]. Some of the capabilities of this module are listed below:

- Operates at a low power supply voltage of 3.3V.
- Small form factor of 22 mm X 28 mm
- Integrates a uFL antenna connector for external antenna connection. We use a 6 dBi gain rubber duck antenna.
- Integrates full TCP/IP stack with an option of bypassing it in the SPI mode. We use the TCP/IP stack within the module.
- Fully compliant with 802.11 b/g and single stream 802.11 n standards.
- Supports TCP, UDP, ARP, ICMP, IPv4 and DHCP.
- Supports infrastructure, ad-hoc and power save mode of operation.
- Current draw of 30 mA in transmit mode and 24 mA in receive mode at 2 Mbps throughput.

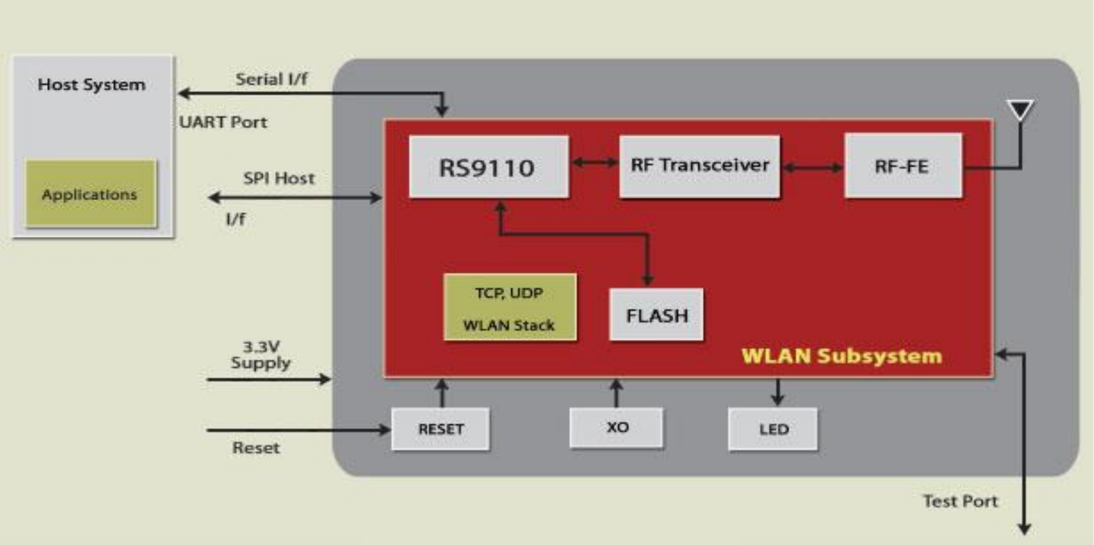


Figure 6.2: General high level block diagram of the Redpine Signals Wi-Fi module

The motivation behind using this module lies quite heavily on its low power consumption, availability of low power modes to further reduce overall current consumption and the integration of a fully functional Wi-Fi transceiver device with minimal processing requirement for the host microcontroller.

6.3.5. ZigBee Module

Figure A.9 in Appendix A shows the schematic of the ZigBee module.

The ZigBee module used in the design is the Texas Instruments CC2530 [54]. The interfacing circuitry and RF front end around this chip is designed and implemented. The RF front end consists of a signal conditioner, as the main component. This is the TDK-DEA202450BT-7210A1 signal conditioning Band Pass Filter for 2.4 GHz band. The circuit is connected to a SMA connector that uses a 6 dBi rubber duck antenna for transmission and reception of ZigBee signals.

Figure 6.3: Texas Instruments CC2530 System-on-Chip

The main characteristics of this chip which led to its selection are given below:

- Small 6 mm X 6 mm form factor with very few external components required to setup functional ZigBee communication.
- Standards compliant 802.15.4 implementation.
- Low power consumption of 24 mA for transmit and 29 mA for receive. Several low power modes are also available.
- High sensitivity of down to -91 dBm.
- High performance, low power consumption 8051 core with code pre-fetch built in the module.
- 256 KB of FLASH which is enough to run the Z-stack ZNP firmware.
- 8 KB of RAM with retention in all power modes.
- CSMA/CA hardware support.
- Accurate digital RSSI/LQI value reporting.

The CC2530 is used in the ZigBee Network Processor configuration. The host Kinetis microcontroller interfaces with this chip using SPI functioning at 4 MHz.

6.3.6. SDHC module

Figure A.10 Appendix A shows the schematic of the SDHC module.

The Secured Digital Host Controller [55] peripheral of the Kinetis K60 microcontroller is used to provide an on-board SD card storage option. In this implementation a standard size SD card is used. The supply required for operation is 3.3 V.

The data communication can be made at a maximum of 25 MHz, but since the requirements of this hardware do not place any specific constraint on this speed, for the sake of data integrity a low speed mode of 400 kHz is used. The 4-wire mode is used and hence at a time only a nibble can be exchanged physically from the microcontroller to the SD card and vice-versa.

6.3.7. USB module

Figure A.11 Appendix A shows the schematic of the USB module.

The hardware is provided with the facility to connect to a laptop or computer over USB. A micro USB connector is provided on the board for this purpose. The USB transceiver is built-in the microcontroller. It implements USB according to the standards specified at [56].

Special attention has been given to route the D+ and D- lines from the micro USB connector to the corresponding pins on the microcontroller. Both the traces have been kept the exact same length to prevent problems in USB communication that crop up if the lengths are different.

The USB is also used to power the hardware. A selector switch is provided on the power board to switch between wall-socket power supply and USB power supply.

6.3.8. RS-232 module

Figure A.12 Appendix A shows the schematic of the RS-232 module.

Although USB exists as a legitimate and high-speed mode for the hardware to exchange information with a laptop or computer, but this mode is easy to implement when only Mass Storage Device functionality is required. When program data needs to be exchanged with the remote device, a driver and other requisite interfacing software components need to be developed for proper communication.

Hence to lessen the development effort involved and also to keep alternate modes of communication available the hardware is equipped with a DB9 connector for communication with a laptop or PC using serial communication that can be monitored on the PC side using readily available terminal software like Hyperterminal. The RS-232 level conversion between the Laptop or PC and the Kinetis microcontroller is done using the well-known Texas Instruments MAX3232 RS-232 transceiver IC [57]. The IC operates from the 3.3 V supply and draws a maximum of 1 mA of supply current while operating. The transfer speed is a maximum of 250 kbps.

6.3.9. LCD module

Figure A.13 Appendix A shows the schematic of the LCD connector.

The USB and RS-232 interfaces are good for connectivity with a laptop or PC. When such a setup is absent, visual indication of processing state can be achieved with the on-board display LEDs. But for many parameters and complex conditions arising due to processing, LED display is not sufficient. Hence, the hardware is provided with a LCD.

The LCD used is a 2.8 inch TFT display from Displaytech – SDT028ATFT [58].

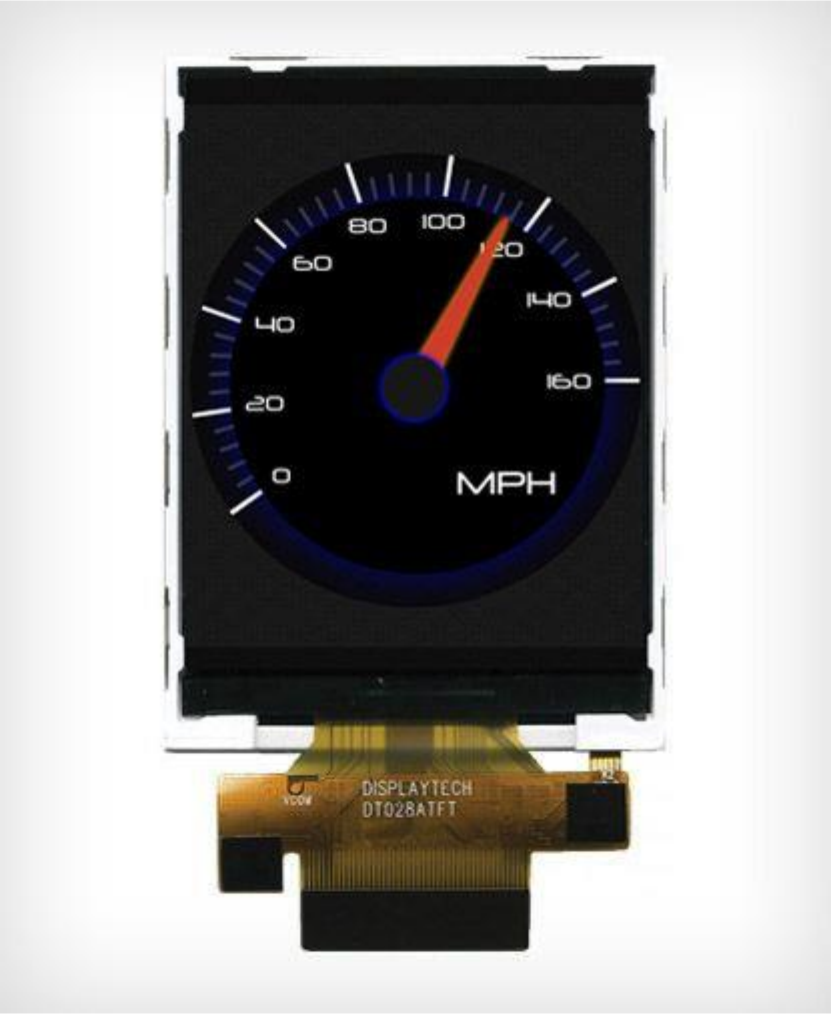


Figure 6.4: Displaytech SD028ATFT LCD with a picture displayed

It supports 240 RGB rows and 320 columns display. It has an integrated display controller ILI9341 [56]. The LCD is equipped with backlight. It operates from 3.3 V power supply. The interface to the microcontroller is over an 8-bit parallel bus made out of microcontroller GPIO lines and a few control lines also interfaced to microcontroller GPIO lines. The connector from the LCD goes in to a 45 pin FPC connector on the power board.

6.3.10. External Sensor Board Expansion Slot

Figure A.14 Appendix A shows the schematic of the sensor board expansion slot.

The hardware is given the ability to interface external analog and digital sensors or other relevant hardware through a 40-pin Hirose [59] connector. The available analog input and output lines, not already used, from the microcontroller are drawn to this connector. The I2C lines from the microcontroller are also drawn to this connector so that sensors with digital output can also be interfaced to the hardware. The need to supply power to the external sensor or hardware is addressed by tracing power supply lines to the connector as well.

6.3.11. PCB Layout

In this section the motivation behind the mechanical design choices will be presented shortly followed by the PCB layout of the hardware.

The number of components and peripherals on the hardware made the option of going with a single board quite infeasible at the very beginning. That would have resulted in a disproportionately large unmanageable board with long traces leading to latency and increased power consumption.

Since the power supply to the board can be from varied sources like battery, solar cells etc., it was decided to make the power supply circuitry separate from the main hardware. There is a connector provided on the main board to which the power board attaches. This gives the freedom to use different power boards, designed differently with different components, with the main hardware as long as the pins on the connector have the same lines connecting on both boards.

A large part of the power board was being left unused and the LCD had to be on a visible part of the hardware. The placement of the LCD on the main board again would have made the board very large. Instead the LCD was placed on the remaining area of the power board with the signal lines being routed up from the main board using board to board connectors. This not only results in efficient usage of space but also lends physical stability to the board by providing two connectors to share the load of the power board rather than just the one power connector.

Since both Wi-Fi and ZigBee have high speed signal lines going in to them, they are placed on opposite planes of the main board. This prevents interference between the high speed signal lines. Both the ZigBee and Wi-Fi portions of the circuit are placed close to the edge of the board so that the antenna connectors are easily accessible and also RF line trace lengths are short. The placement of the two parts of the circuits are also as far apart from each other as possible in order to minimize signal interference as they operate in the same frequency range although channel assignments, power outputs and spreading characteristics are different.

The SD card slot and the sensor expansion board connector need to be reached from the side as well as from the top, they are placed to the south end of the main board. The top side reachability is also the reason why the power board and the main board do not share the exact same dimensions.

To lend extra mechanical stability to the boards and provide easy access to the underside of the hardware there are four nut and bolt arrangement with 5 mm spacers in between holding the two boards together.

The PCB layout of the boards is provided in Appendix C.

6.3.12. HTNMote in Pictures

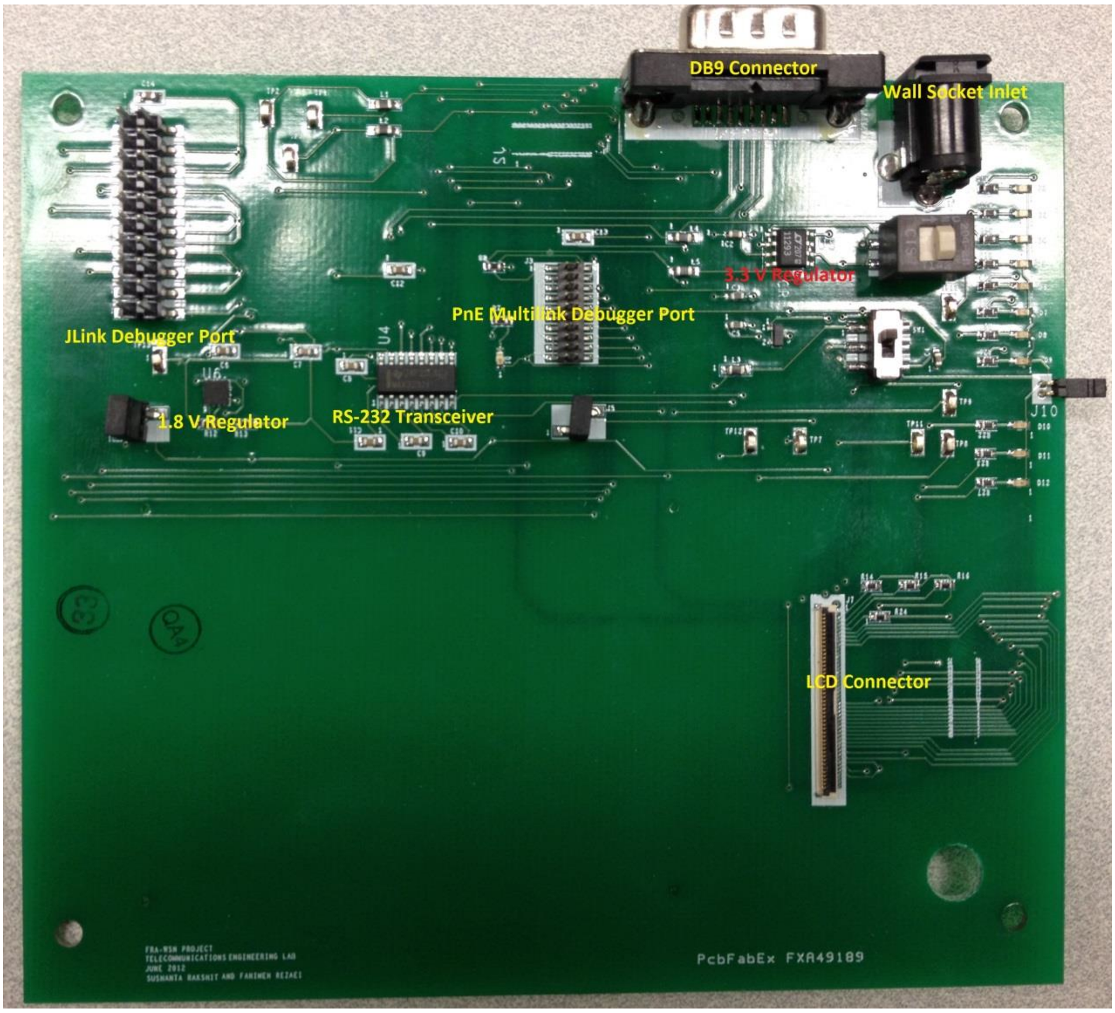


Figure 6.5: Power Board Top View

The above figure shows the Power Board from the top. The major components on the board are marked in yellow.

The connectors (DB9 and wall socket power supply) are placed towards the right hand top corner flush with the edge of the board.

The two switches seen in the figure are the power on/off switch and the switch which selects whether USB or wall socket power supply is the source of power for the board.

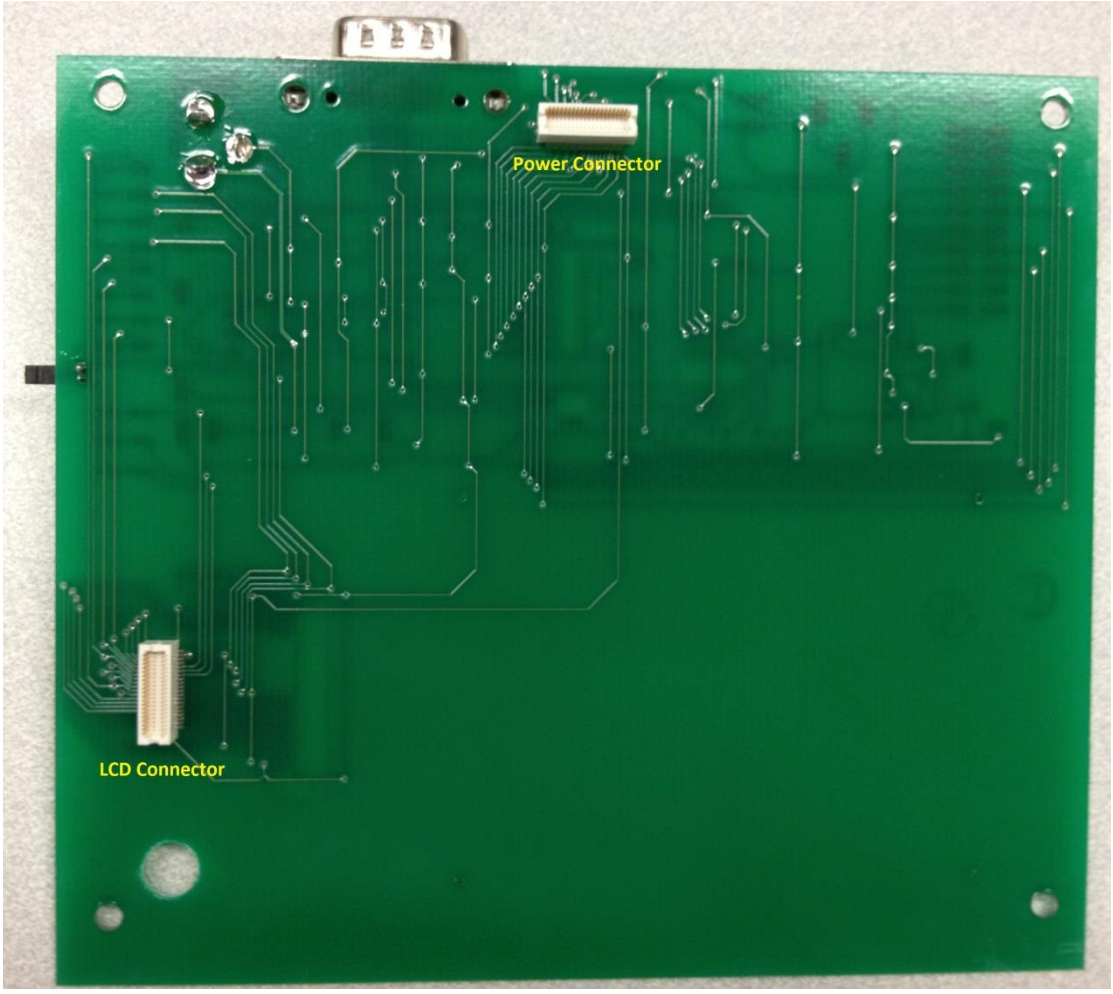


Figure 6.6: Power Board Bottom View

The above figure shows the Power board from the bottom. The two connectors here are the board-to-board power connector (top center) and the board-to-board LCD connector (left down). The power connector takes the power supply and RS-232 signal lines from the Power Board to the Main Board and vice-versa. The LCD connector takes the signal and data lines related to the LCD from the Main Board to the Power Board and vice-versa.

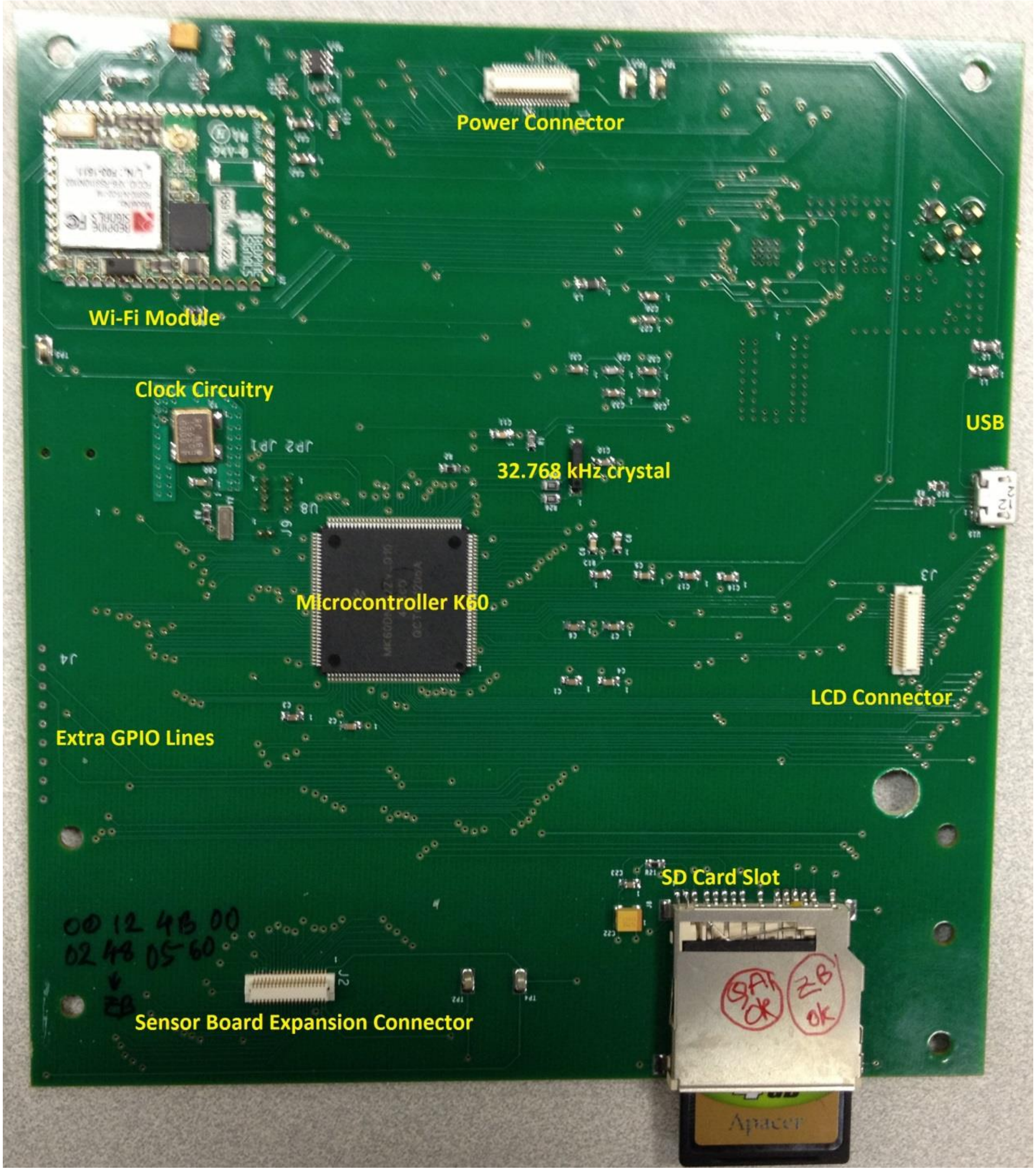


Figure 6.7: Main Board Top View

The above figure shows the Main Board from the top. The main components on the board are indicated using yellow texts.

The mating connectors for the power and LCD connectors on the Power Board can be seen.

There are some GPIO lines which we do not use on the microcontroller and they are routed to the left side edge of the board so that we can use those lines externally if required.

We can see that there is a guard band connected to the ground around the clock circuitry near the microcontroller. This is necessary as the high frequency clock sources are very sensitive to noise and external interference.

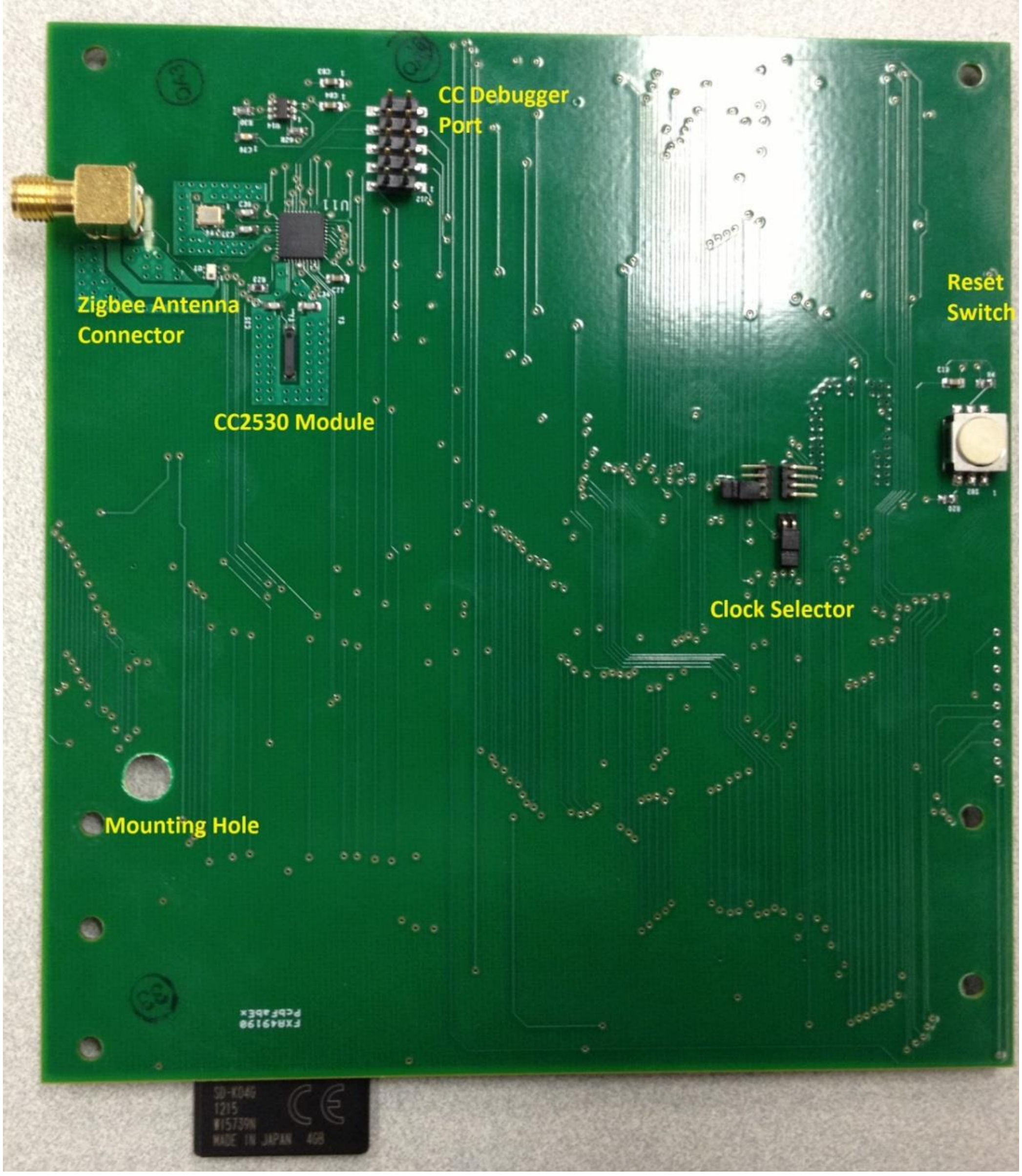


Figure 6.8: Main Board Bottom View

The above figure shows the Main Board bottom view. We can see that the guards connected to the ground plane are again placed around the clock circuitry for the ZigBee module. This is necessary as these high frequency clock sources are very sensitive to noise and external interference.

The CC-Debugger connector is provided so that the internal workings of the CC2530 chip can be traced and debugged during development.

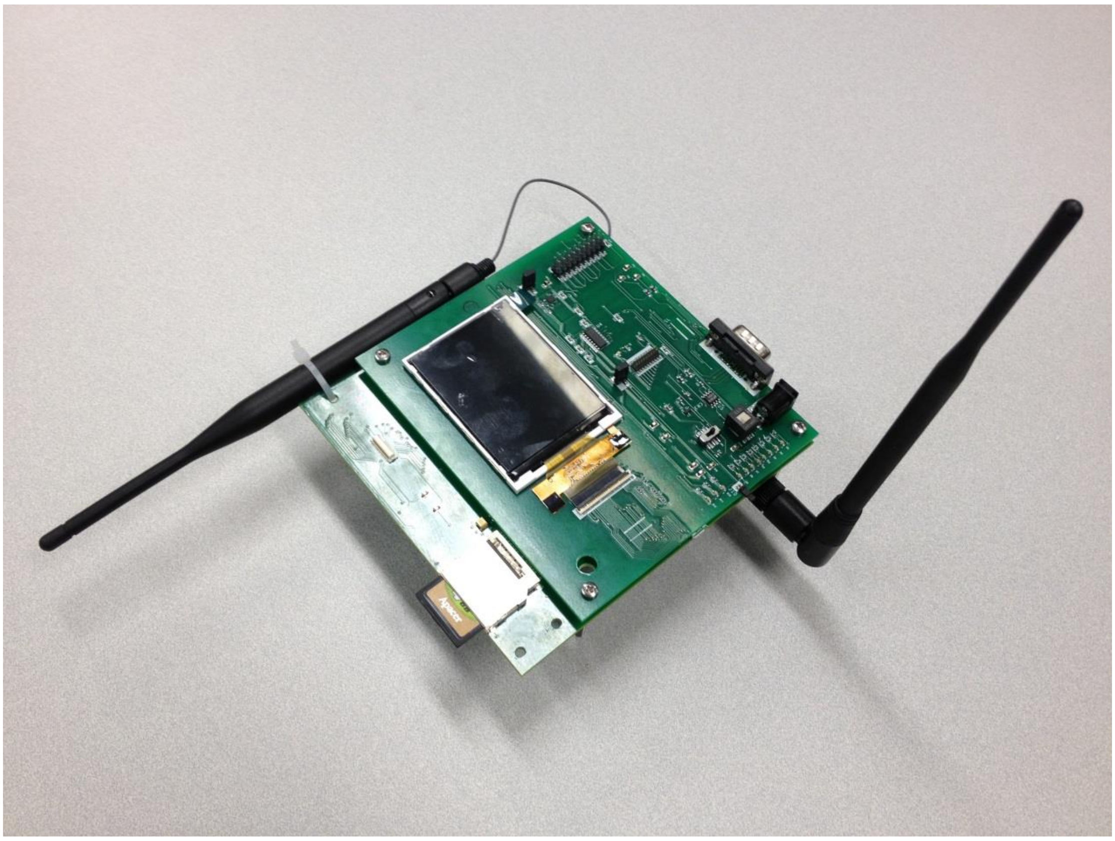


Figure 6.9: HTNMote - Assembled

The figure above shows the HTNMote fully assembled. The Main Board and the Power Board put together using nuts and bolts and spacers through the mounting holes.

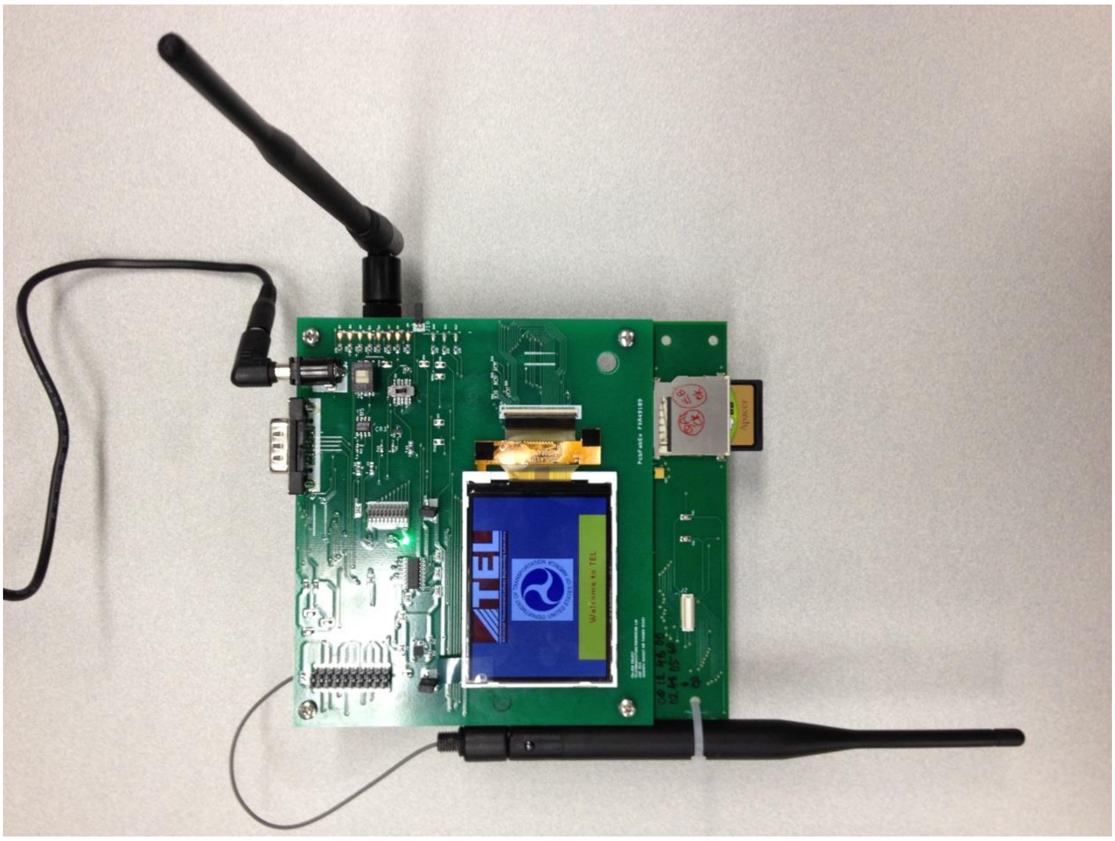


Figure 6.10: HTNMote – In action

The figure above shows the HTNMote powered by a wall socket supply. We can see the green power LED indicator glowing. The LCD is programmed to display some pictures and a line of text. The bank of indicator LEDs near the wall socket power supply inlet is also glowing as they are connected to the data lines of the LCD.

Chapter 7. SOFTWARE IMPLEMENTATION

The development platform used for writing the software for this hardware is the Freescale CodeWarrior for MCU v10.2 [60]. It is an Eclipse based IDE that allows C/C++ to be used as the coding language of choice. It has a Processor Expert module that slashes development time by automating low level register setup of the Kinetis microcontroller. The user is presented with a GUI to select the properties that the microcontroller needs to have for an implementation. The Processor Expert will automatically generate the low level code required to setup the microcontroller. The user can start from writing the actual system code by using the functions that the Processor Expert exposes. All these functions are wrappers for low level system calls.

7.1. Wi-Fi Implementation

The Wi-Fi module from Redpine Signals comes with an API that can be used to communicate over SPI with the host microcontroller [53].

This API contains two header files, namely rsi_config.h and rsi_global.h, which consist of all the variable parameter values that can be set by the user. For example, one such parameter can be the BSSID of the network to be created or joined in ad-hoc mode.

The API communicates with the host microcontroller with a series of low level hardware drivers that need to be modified according to the host microcontroller being used. Here, these drivers have been modified to work with the Kinetis K60 SPI peripheral.

The API consists of a central union that has all the variables needed to execute commands on the module and also to receive responses from the module as a result of executing those commands. The module carries the Wi-Fi functionalities out based on the values that are found in this one central union.

The module expects a little Endian format from the host. The SPI signal characteristics expected are a clock polarity of 0 and a rising edge clock phase. The commands issued to the module consist of 4 bytes C1, C2, C3 and C4 which together denote all the aspects of the command and ensuing data transfer to the module. Every command in the API has a unique value for C1, C2, C3 and C4.

The API has six different types of commands namely, initialization, memory read and write, frame read and write and register read.

The exchanges involved in each of these types are shown below [53]:

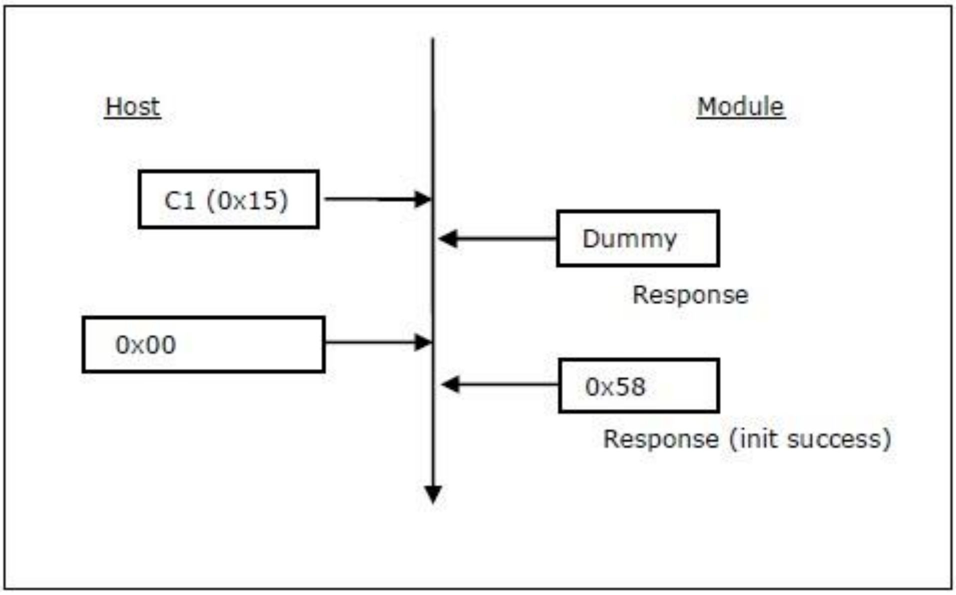


Figure 7.1: Slave initialization procedure

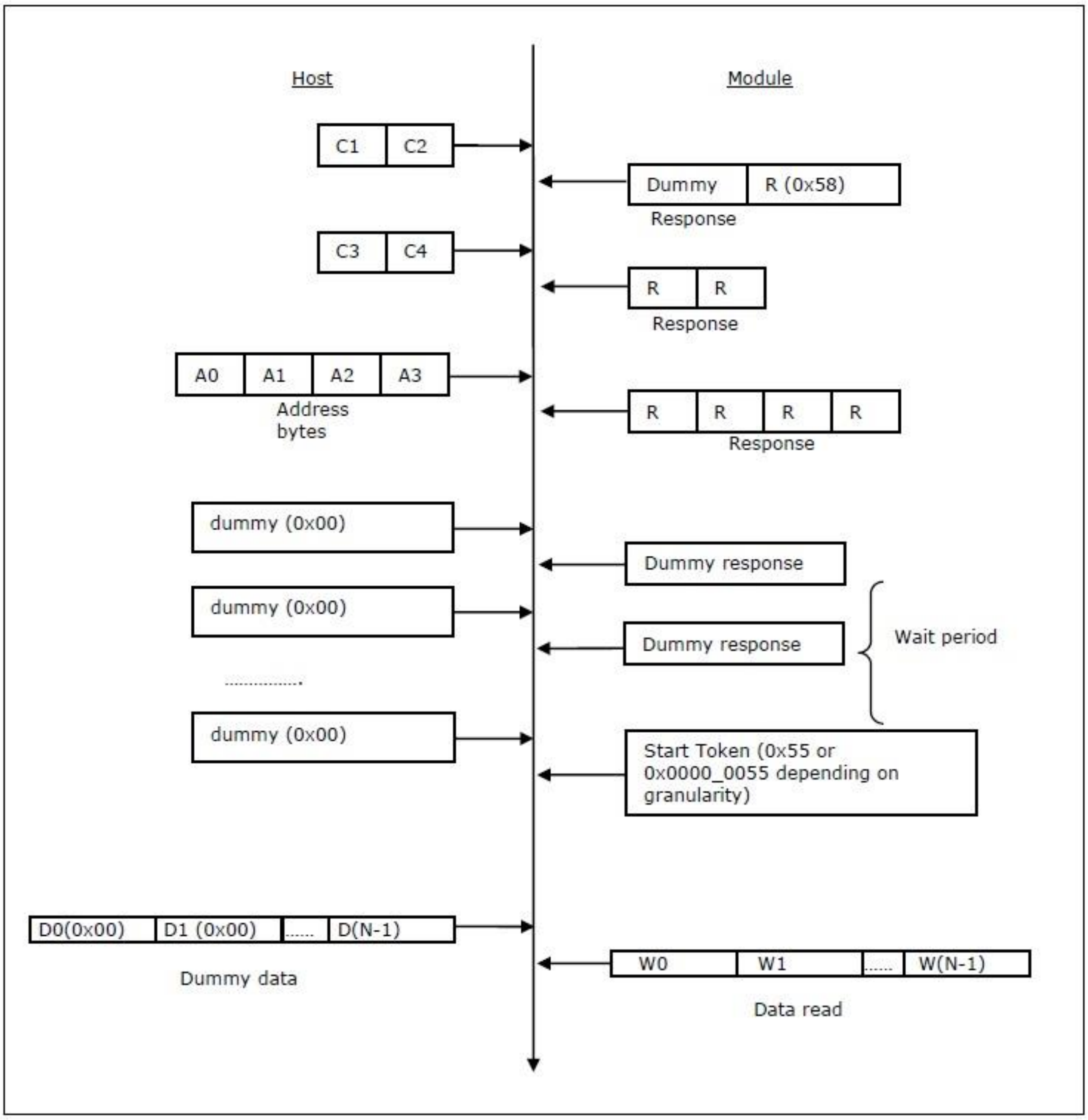


Figure 7.2: Memory Read or Master Read procedure

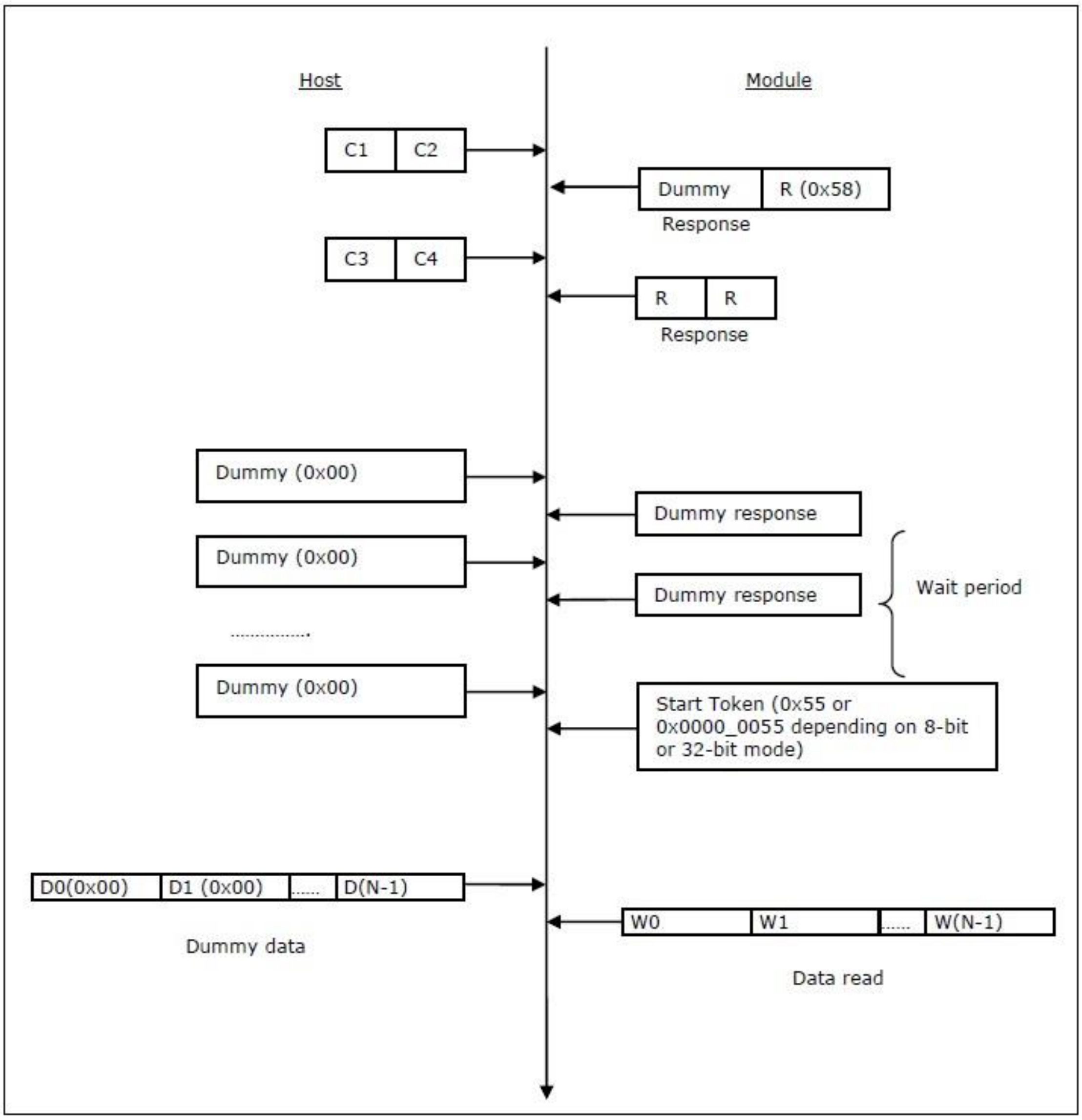


Figure 7.3: Frame read or Slave read procedure

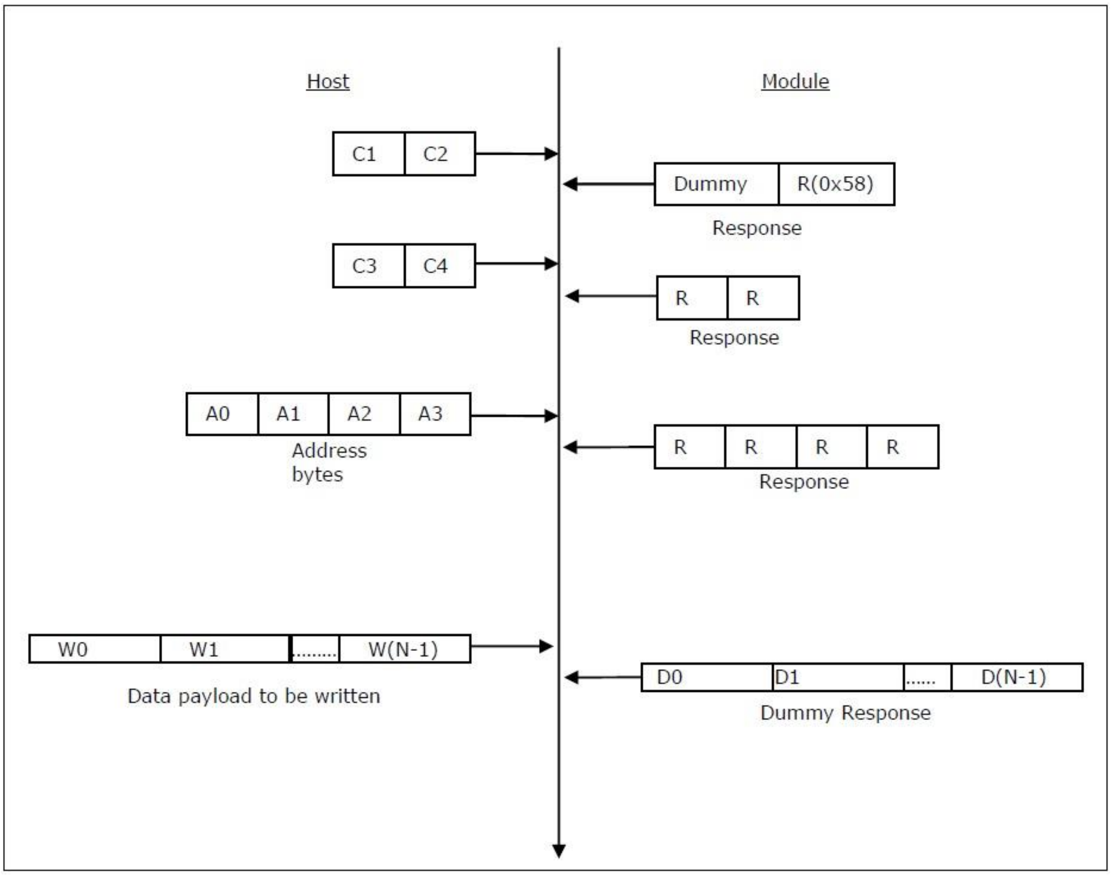


Figure 7.4: Memory Write or Master Write procedure

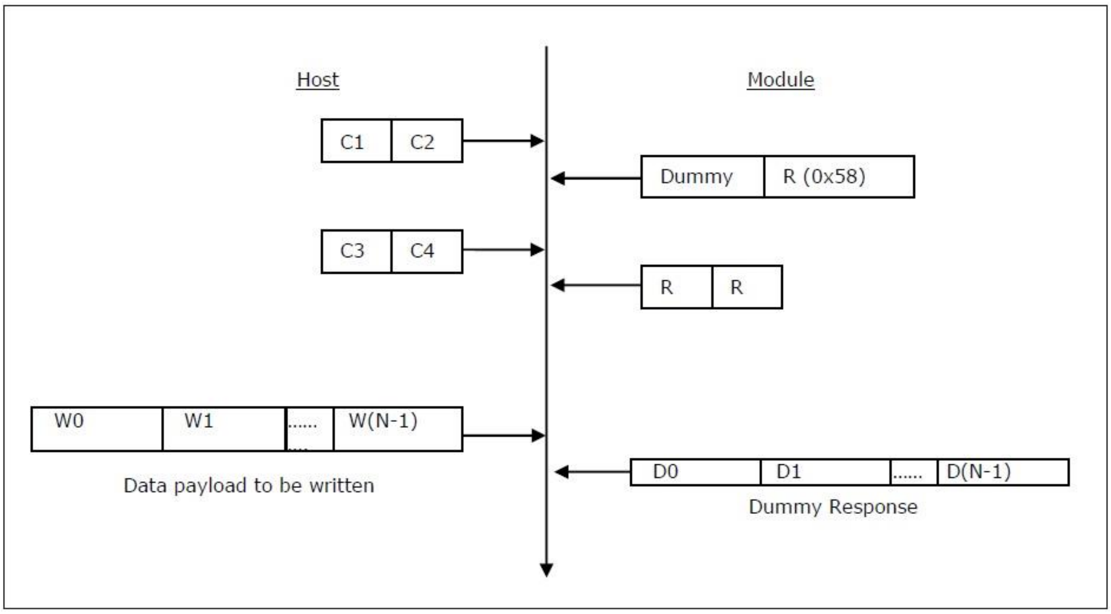


Figure 7.5: Frame write or Slave write procedure

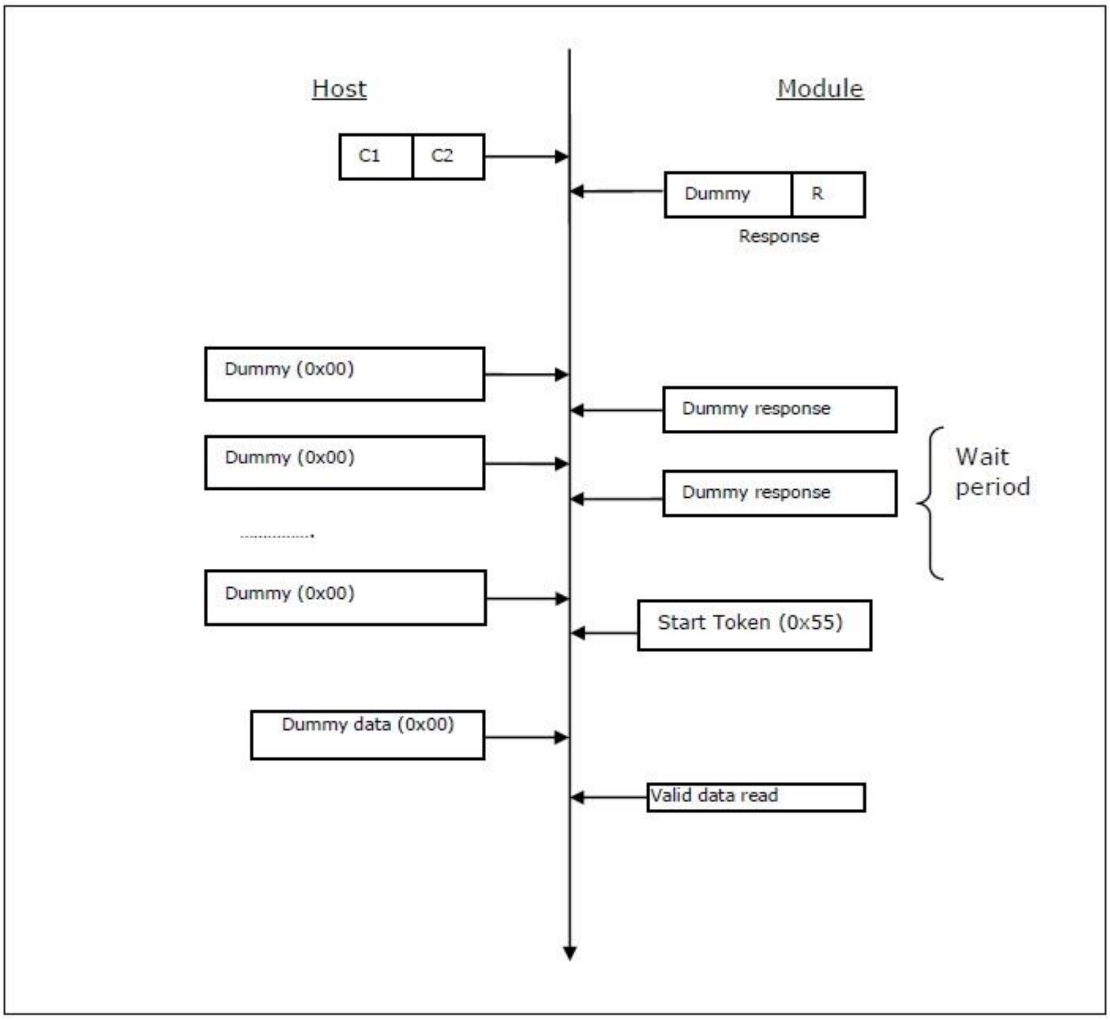


Figure 7.6: Register read procedure

Utilizing this API the code for Wi-Fi send and receive functionalities has been written. The corresponding flowchart is shown in Figure B.1 of Appendix B.

7.2. ZigBee Implementation

The ZigBee software has been implemented using the Z-Stack ZNP variant. The CC2530 is setup to be a ZigBee Network Processor with the Kinetis K60 being the host microcontroller communicating with the CC2530 over SPI. This scenario is shown in the figure below:

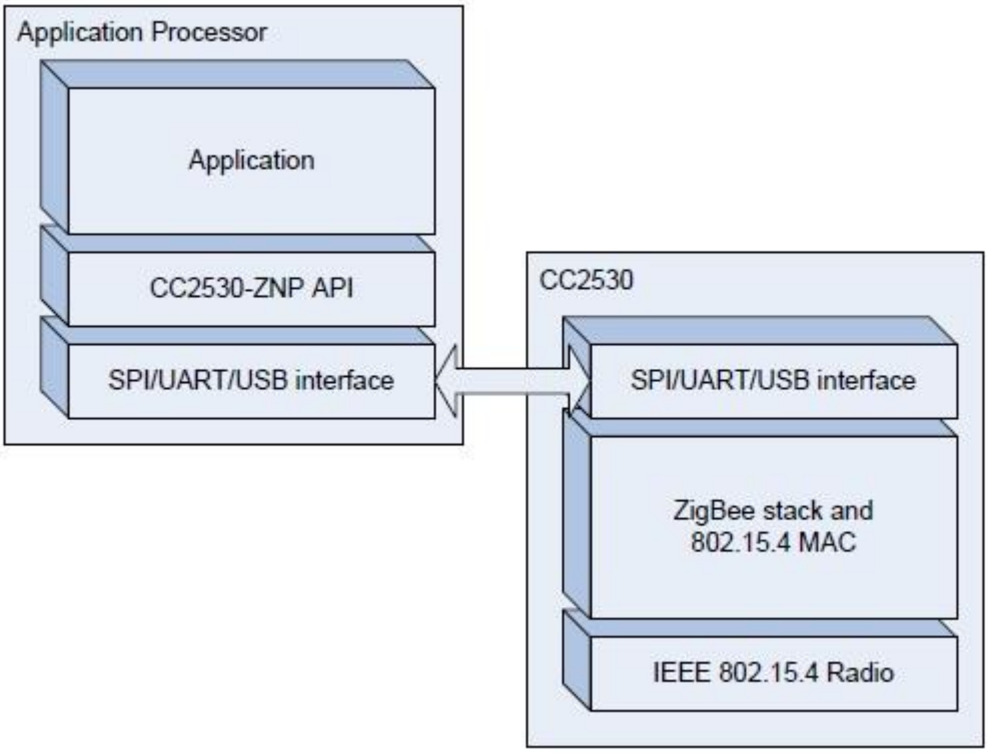


Figure 7.7: CC2530 in ZNP configuration (Application Processor is K60)

The SPI is clocked at the maximum speed possible for CC2530, 4MHz. The clock polarity is 0 and rising edge is used as the clock phase. MSB first bit ordering is used. Standard 4 wire SPI is in use [54].

Apart from the standard SPI lines the CC2530 interface also requires two additional control signal lines. These and their brief description are provided below [54]:

a) SRDY: Slave ready signal. This signal is set by the CC2530 when it is ready to receive or send data. It is set low when the CC2530 is ready to receive data.

b) MRDY: Master ready signal. It is an active low signal. This signal is set by K60 when it has data ready to send to the CC2530.

There are three types of command available under the CC2530 ZNP API. These are AREQ, POLL and SREQ. The details of these commands can be found in [54].

The CC2530 can be programmed as a ZNP in two modes, namely Simple API mode and the AF-ZDO mode. In this research the AF-ZDO mode is used as it offers better control over the CC2530 module as opposed to the Simple API.

The CC2530 ZNP software has three sections that it is divided into. These are presented briefly below [54]:

a) SYS Interface: This provides the application processor a low level interface to the CC2530 hardware and firmware. The peripherals on the CC2530 accessible through this interface are the ADC, NV items, hardware number generator and GPIO pins.

b) Configuration Interface: The commands in this interface allow us to set up various parameters of the CC2530 radio chip.

c) AF-ZDO interface: This interface consists of the Application Framework (AF) which is used by the application processor to register its application with the CC2530 thereby being able to send and receive data. The ZigBee Device Object (ZDO) interface allows access to all network management related functions.

Utilizing API the software for ZigBee packet transmission and reception is written. The flowchart outlining the code flow is shown in Figure B.2 of Appendix B.

7.3. SDHC Implementation

The SDHC interface code is written using the Processor Expert exposed function calls pertaining to the SDHC module in the K60. The block size of transfer is 512 bytes. A write, read-back and compare cycle is followed for every 512 bytes of transfer to ensure integrity of data written on to the SD card.

The flowchart for the SD card interface code is shown in figure B.3 of Appendix B.

7.4. USB Implementation

The USB on the hardware is implemented as a mass storage device with vendor ID (VID) of 0x1234 and a product ID (PID) of 0x5678.

The flowchart for the USB implementation code is shown in figure B.4 of Appendix B.

7.5. RS-232 Implementation

The serial communication software is set up to provide bi-directional communication capabilities to ensure monitoring from the remote terminal side as well as actuation from the remote terminal.

The software is currently setup to send a string for display on the remote terminal. It then waits for 2 bytes of input to come from the remote terminal. Once the inputs arrive they are echoed back for display on the remote terminal.

The communication baud rate is 9600 bauds and the character size is 8 bits with no flow control.

The flowchart for the RS-232 implementation code is shown in figure B.5 of Appendix B.

7.6. LCD Implementation

The LCD implementation is done using the embedded GUI (eGUI) [61] from Freescale. This is a software suite that is used to interface any TFT-LCD with a Freescale microcontroller.

This is a powerful but light-weight suite that has the capacity of rendering colors as well as the following graphics objects:

- Button
- Check Box
- Radio Button
- Gauge
- Icon
- Label
- Picture
- Graph
- Slider
- Menu
- Scroll Bar
- Console
- Text Box

It supports touch screen, multiple fonts and various screen sizes.

The structure of the software suites is layered. The lower layers contain the low level drivers that interface the microcontroller with the controller on the LCD module. As part of this research, new drivers were written to interface eGUI with the ILI9341 [62] display controller chipset being used with the LCD.

The eGUI suite block diagram is show in the figure below [61]:

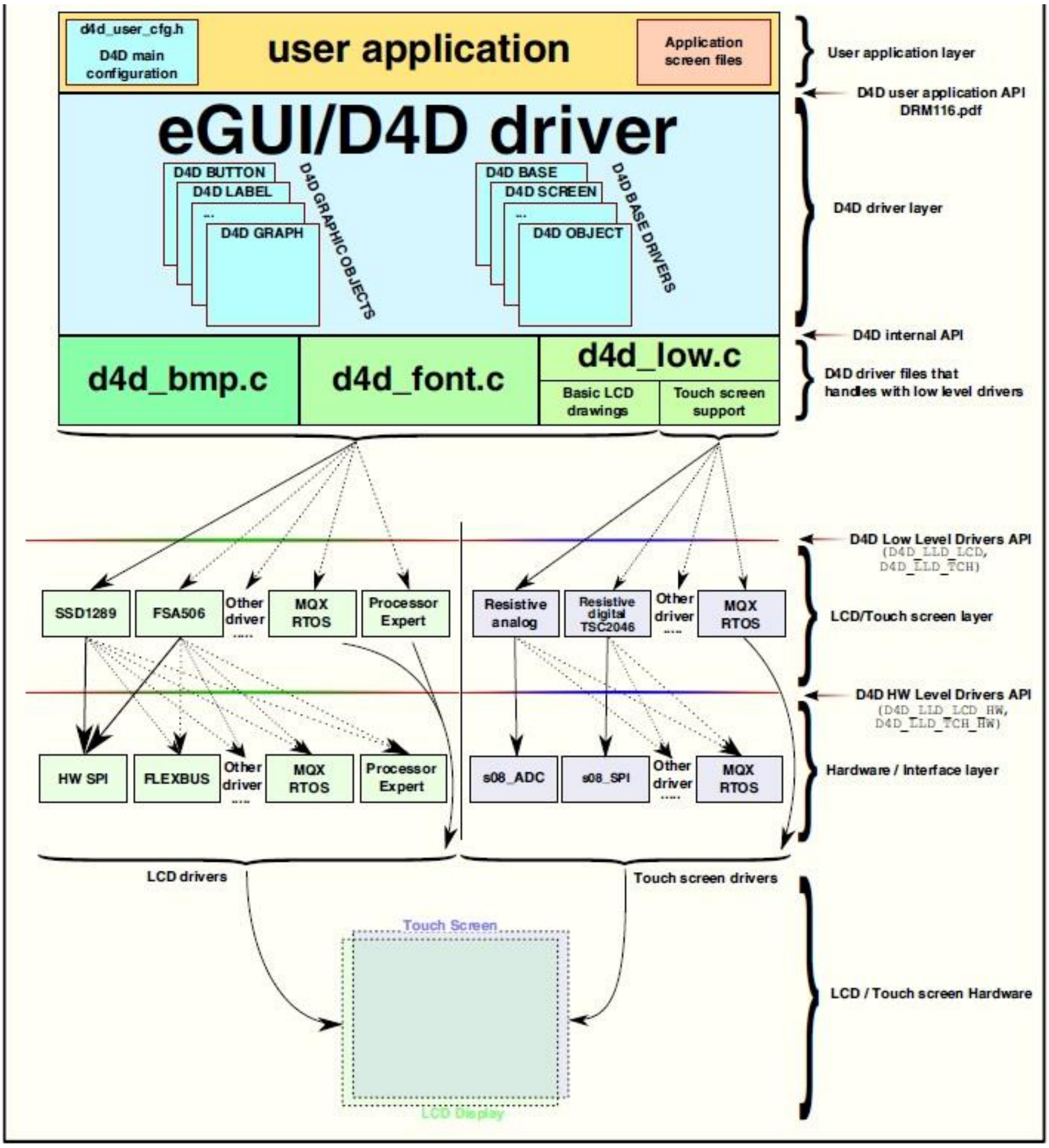


Figure 7.8: eGUI block diagram

The flowchart for LCD code is shown in figure B.6 of Appendix B.

Chapter 8. TESTS AND TEST RESULT ANALYSIS

Once the hardware platform was manufactured several tests were run on it. Several of these tests were functionality tests to confirm whether the module in the design was working as expected or not. The components mainly tested for functionality only were USB, RS-232, LCD and SDHC.

The other components, namely Wi-Fi and ZigBee, were tested for performance. This was essential as the HTN protocol needs the corresponding hardware to meet certain performance benchmarks for it to be suitable for implementation of the protocol.

In this chapter the first sub-section will consist of the test methodology, results obtained and result analysis for the performance tests. The second sub-section will consist of the functionality test outcomes.

8.1. Performance Tests

In the intended usage scenario of the HTNMote the most critical components on the board, apart from the microcontroller, are the Wi-Fi and the ZigBee modules. If these two modules do not perform above a certain level in comparison with other off-the-shelf devices then the HTNMote solution will be rendered infeasible. The following bunch of performance tests helps to show the capabilities of the communication hardware on the HTNMote.

8.1.1. Methodology

The performance tests have been divided in to three groups, namely, the current consumption tests, the Wi-Fi tests and the ZigBee tests. The methodology followed for both these tests are described in the rest of this sub-section.

8.1.1.1. Current Consumption Test Setup

In this research work, the long-run steady state current consumption of various peripherals on the HTNMote are measured. The test setup consists of 5 VDC, 1A supply provided to the HTNMote using a wall power supply adapter. The voltage regulators on the board convert this to the 3.3 V and 1.8 V on the board as required.

The current consumed from the wall socket source is measured. This gives us an idea of the overall current consumption from all components and thus any battery life model can be effectively come up with.

A multimeter is connected in series with the wall socket power adapter and the HTNMote and the current readings are collected.

8.1.1.2. Channel Emulator

A wireless channel emulator from Azimuth Systems ACE 400WB [63] is used to create wireless channels between the two end devices. The channel emulator is shown in figure below:

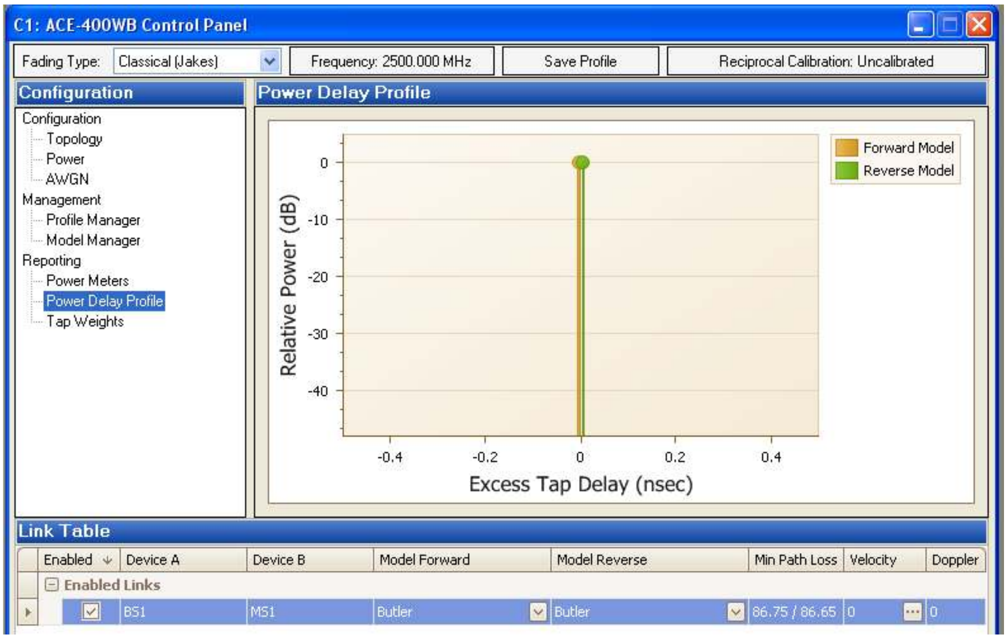
Figure 8.1: ACE 400WB wireless channel emulator

This piece of equipment is a computer controlled device. It is able to create accurate user-defined channel conditions between two devices at two ends of the channel. This allows us to do accurate real-world tests without having to make arrangements for elaborate field tests.

There are several advantages of using the channel emulator over real-world field tests. They are listed below:

- The use of the channel emulator allows the configuration of the physical channel exactly according to the specifications of the user. This means that an accurate physical channel containing a tree, a house, two tall buildings and a stone can be emulated. Whereas, finding a place in the field which has exactly this configuration of physical structures is very difficult to find.
- The other problem of the real-world is that the channel conditions are never static and not repeatable. There are hundreds of parameters that keep on changing every second in a real-world channel. Hence, if a device needs to be tested in several configurations under equivalent channel conditions then that becomes an impossible task in the field. However, with the channel emulator the user has strict control over the channel and hence repeatability is easily achieved.

A channel emulator is able to create any realistic real-world channel with great accuracy. In this research work only the ITU-T standard channel Butler model [64] is used. The power delay profile of the Butler channel model is shown in the figure below captured from an actual active session of the channel emulator:



C1: ACE-400WB Control Panel

Fading Type: Classical (Jakes)

Frequency: 2500.000 MHz

Reciprocal Calibration: Uncalibrated

Enabled	Device A	Device B	Model Forward	Model Reverse	Min Path Loss	Velocity	Doppler
⊟ Enabled Links							
☑	BS1	MS1	Butler	Butler	86.75 / 86.65	0	0

Figure 8.2: Power delay profile of the ITU-T Butler model

The Butler model is a static non-fading channel model that uses the identity matrix for the channel coefficients. It does not have any multipath components or scattering.

One of the common settings for the channel emulator common to both the Wi-Fi and ZigBee tests is that the MIMO antennas were configured to have no correlation.

8.1.1.3. Wi-Fi Tests Setup

The Wi-Fi tests consist of two tests that are performed to obtain the following characteristics of the hardware:

a) Throughput vs. Packet size

b) Packet Loss vs. Path Loss

The channel emulator setup for the Throughput versus Packet size test is as follows:

- The path loss between sender and receiver fixed at 67.55 dB
- The path loss between receiver and sender fixed at 67.65 dB
- Input side attenuation Port A : 0 dB
- Output side attenuation Port A : 0 dB
- Input side attenuation Port B : 0 dB
- Output side attenuation Port B side : 0 dB
- WiFi channel used is channel number 1 centered around 2.412 GHz.

The channel emulator setup for the Packet Loss versus Path Loss test is as follows:

- Input side attenuation Port A : 0 dB
- Input side attenuation Port B : 0 dB
- WiFi channel used is channel number 1 centered around 2.412 GHz.

The layout and logical interconnection of the devices for the test is given below:

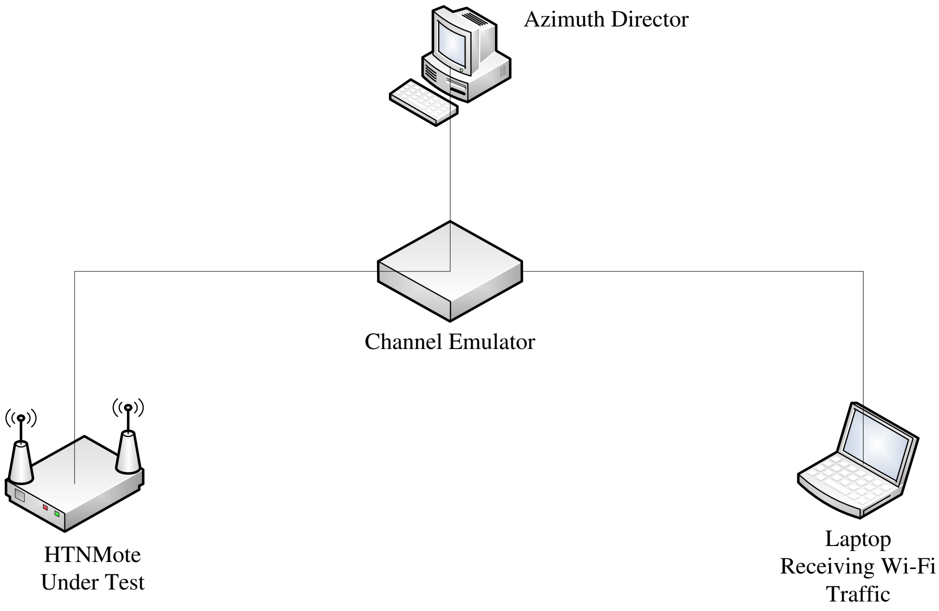
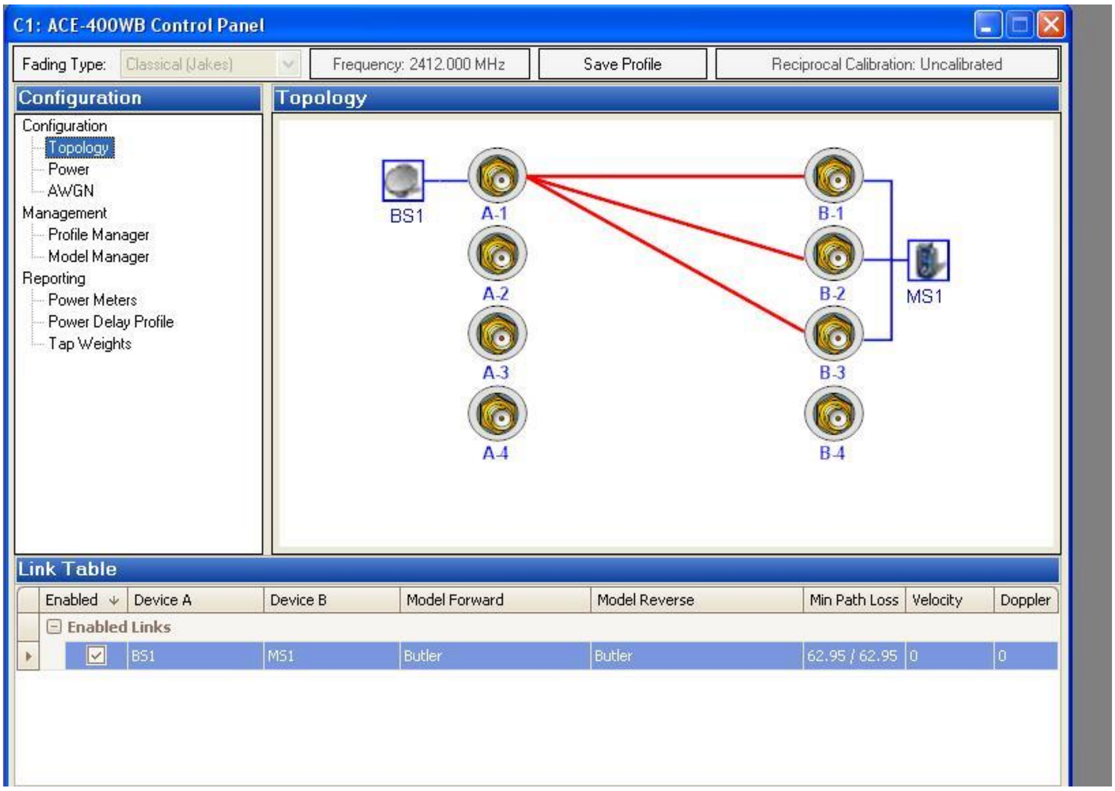


Figure 8.3: Layout and device interconnections for Wi-Fi tests

The channel configuration for the Wi-Fi tests on the channel emulator is shown in the figure below:



Fading Type: Classical (Jakes)

Frequency: 2412.000 MHz

Reciprocal Calibration: Uncalibrated

Enabled	Device A	Device B	Model Forward	Model Reverse	Min Path Loss	Velocity	Doppler
☑	BS1	MS1	Butler	Butler	62.95 / 62.95	0	0

Figure 8.4: The topology for the Wi-Fi tests

In the configuration above the device BS1 is connected to Port A1 of the channel emulator. The BS1 is the HTNMote device under test. The MS1 is the laptop which is the device that is the destination for the Wi-Fi traffic generated by the HTNMote device under test.

The HTNMote device under test is configured to join an ad-hoc network with SSID ‘FRAWSNNet’. It transmits at 16 dBm and the application code inside controls the packet payload size for transmission.

8.1.1.4. ZigBee Tests Setup

The ZigBee module is tested for the following characteristics:

a) Throughput vs Path Loss

b) Goodput vs Path Loss

c) Packet Loss Rate vs Path Loss

The test setup includes the device under test which transmits ZigBee packets of various payload size to the Coordinator device being monitored from Code Warrior using the debugger.

A Texas Instruments ZigBee packet sniffer [65] is used to monitor all packet exchanges happening in the channel. This allows us to later parse the packet trace and find the throughput, goodput and packet loss rate parameters.

The layout and device interconnections for the ZigBee tests are shown in the figure below:

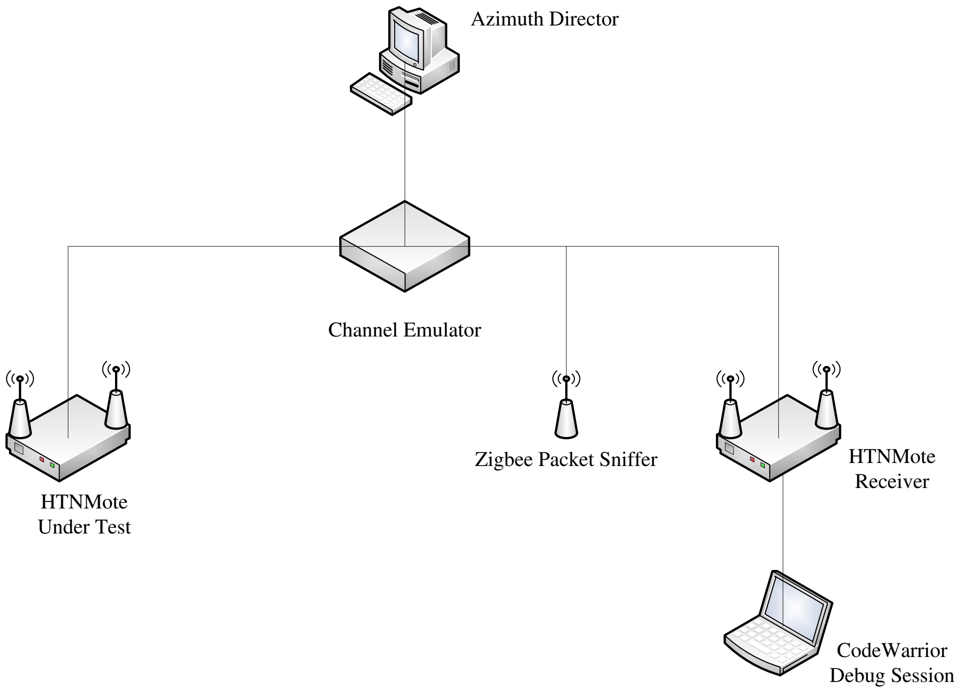
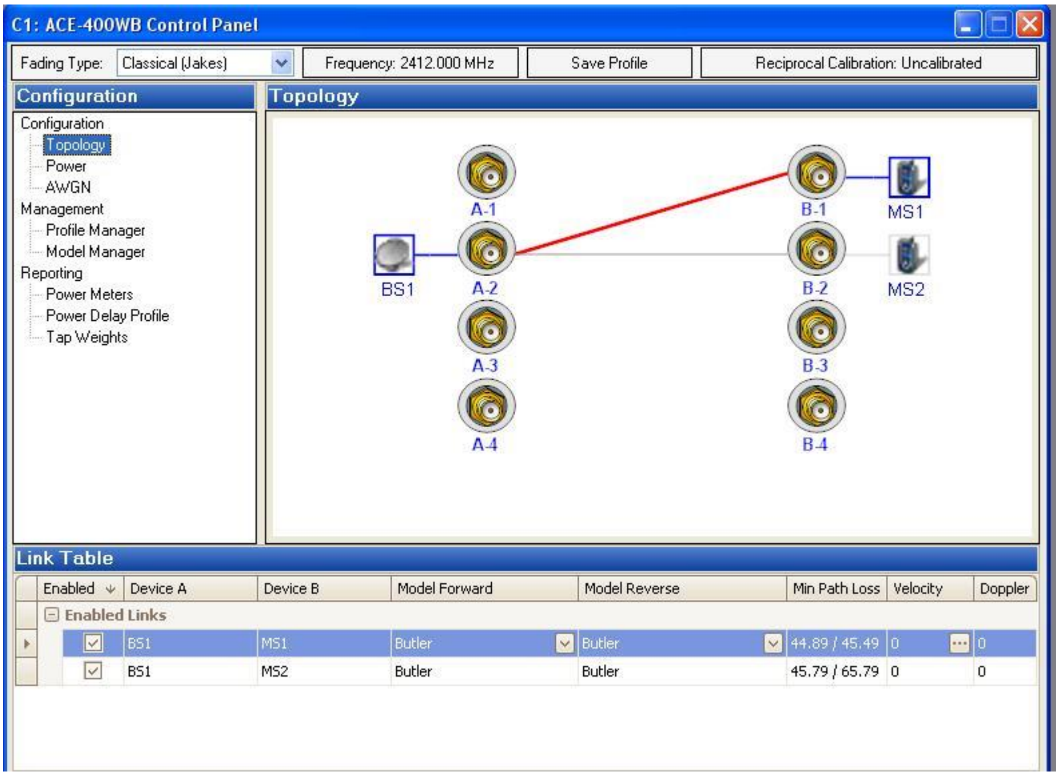


Figure 8.5: Layout and device interconnections for ZigBee tests

The channel emulator is setup in the following layout:



Fading Type: Classical (Jakes)

Frequency: 2412.000 MHz

Reciprocal Calibration: Uncalibrated

Enabled	Device A	Device B	Model Forward	Model Reverse	Min Path Loss	Velocity	Doppler
☑	BS1	MS1	Butler	Butler	44.89 / 45.49	0	0
☑	BS1	MS2	Butler	Butler	45.79 / 65.79	0	0

Figure 8.6: ZigBee tests channel emulator topology

In the above layout the device under test is denoted by the BS1. This is a HTNMote that is configured to be an end-device which sends ZigBee packets of payload sizes 2 and 84 bytes to the coordinator device. A total of 10001 packets are sent for every execution of a test.

MS1 denotes the coordinator device that receives packet transmissions from the device under test. It is monitored via a debugger from the Code Warrior IDE.

MS2 denotes the ZigBee packet sniffer. It is kept parallel to the device under test so that all the channel activity can be captured. The path from the device under test to the sniffer is not attenuated. All attenuations are placed in the path from the device under test to the coordinator device.

8.1.2. Current Consumption Test Results

The current consumption observed for various scenarios of interest are tabulated below:

Scenario	**Current Consumption**
HTNMote Switched on: voltage regulators, power on LED, RS-232, SD Card, USB, clock circuitry, microcontroller active. No LCD.	48.4 mA
HTNMote Switched on and a single indicator LED glowing	49.3 mA
HTNMote Switched on and RS-232 transaction on-going	51.5 mA
HTNMote Switched on and SD Card transfer on-going	50.2 mA
HTNMote Switched on and connected as USB mass storage device	59.3 mA
HTNMote Switched on and LCD displaying graphic	77.5 mA
HTNMote Switched on and ZigBee sending data to coordinator	80.1 mA
HTNMote Switched on and ZigBee	91.1 mA

receiving data from end-device	
HTNMote Switched on and WiFi module in idle but full power mode	163.4 mA
HTNMote Switched on and WiFi module sending data	197.1 mA
HTNMote Switched on and WiFi module sending data	190.1 mA

Table 8.1: HTNMote Current Consumption

When looking at the current consumption figures present in the table above we must keep note of the fact that all of these figures are not stand-alone consumption values for a corresponding component. Each of these values includes consumption due to all peripherals active as well as the code running in the microcontroller. This holistic view is necessary as in the field it is this overall current consumption which is of importance rather than individual component consumption.

8.1.3. Wi-Fi Test Results

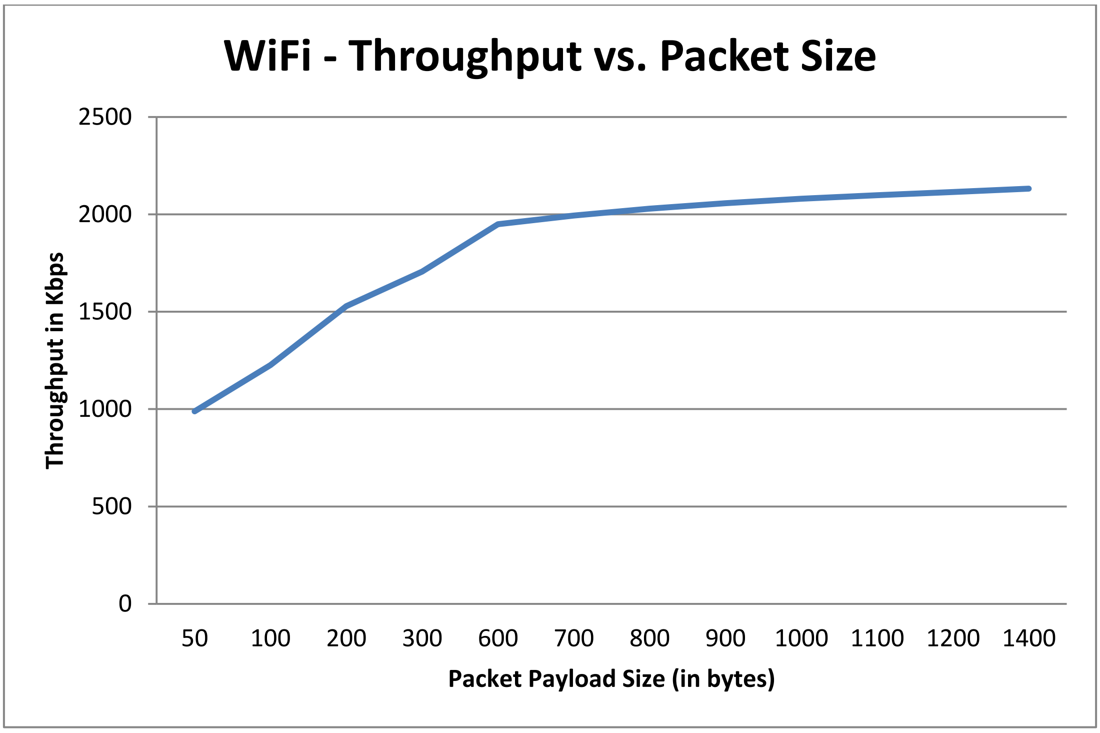


Figure 8.7: Throughput vs. Packet size test result

The above test is executed by keeping the channel conditions same and varying the transmitted packet payload size from 50 bytes to a Wi-Fi maximum packet payload size of 1400 bytes.

It is observed from Figure 8. That the lower packet payload sizes do not make the device Wi-Fi operate at the saturation level of throughput and hence the throughput gradually increases. As the packet payload size increases the throughput starts to level off. It can be seen that using packet payload sizes of 800 bytes and above will guarantee that the device is operating close to the maximum throughput possible given all the other conditions.

It can be seen that the Wi-Fi module allows nearly 2.2 Mbps as the maximum throughput.

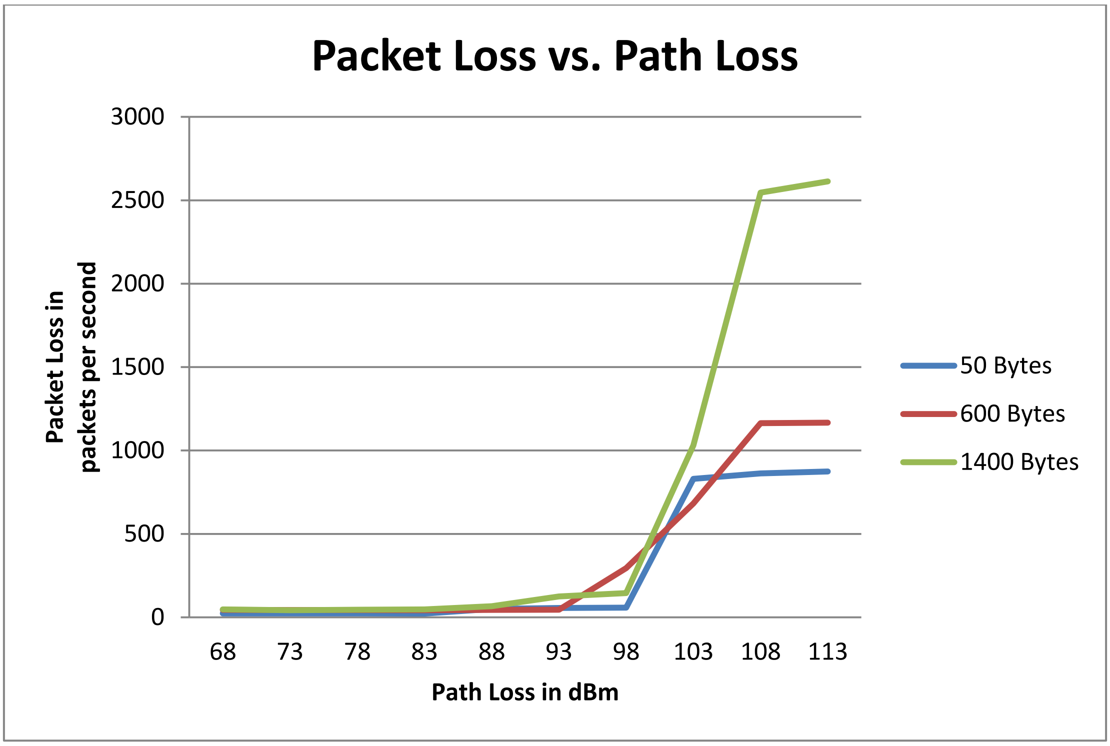


Figure 8.8: Packet loss vs. Path loss result

The Figure above shows that the overall packet loss for the Wi-Fi module is in the vicinity of a maximum of 32 packets per second when the payload size is 1400 bytes, the throughput is 2.2 Mbps and the path loss is 113 dB.

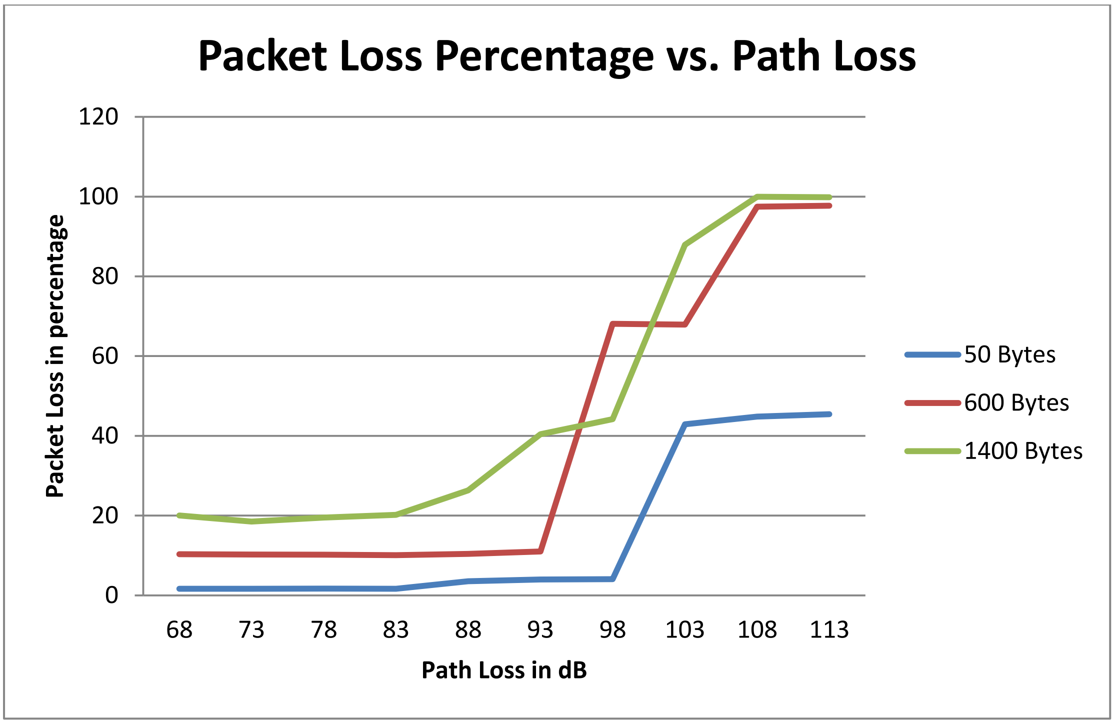


Figure 8.9: Packet loss percentage vs. Path Loss result

The above figure shows the overall percentage packet loss observed for a given path loss for different payload sizes at 2.412 GHz center frequency of operation. It is of importance to note that the above result will be affected by change in channel conditions and should be seen as the worst-case packet loss given similar conditions. We can see that the overall packet loss increases dramatically as the channel deteriorates. For smaller payload sizes we still are able to receive about 50% of the packets when there is in excess of 100 dB path loss. But as the payload size increases this amount becomes less and less and worst-case we observe near 100% packet loss for 1400 bytes payload size at a path loss of 100 dB and above.

We use the following formula convert the path loss value into a distance value:

$$Path\ Loss = 10n\log_{10}\left(\frac{4\pi d}{\lambda}\right)$$

In the above equation the path loss exponent (n) is assumed to be 2. The distance (d) between the sender and the receiver is in unit of meters. The wavelength (λ) is for the center frequency of 2.412 GHz. The path loss in the equation is in the unit of decibels (dB).

Using the above formula we can see that for a distance of separation of roughly 1 Km between the sender and the receiver the packet loss is a meager 10 - 12 packets per second when the transmission is very fast with 2.2 Mbps throughput using 1400 bytes packet size.

8.1.4. ZigBee Test Results

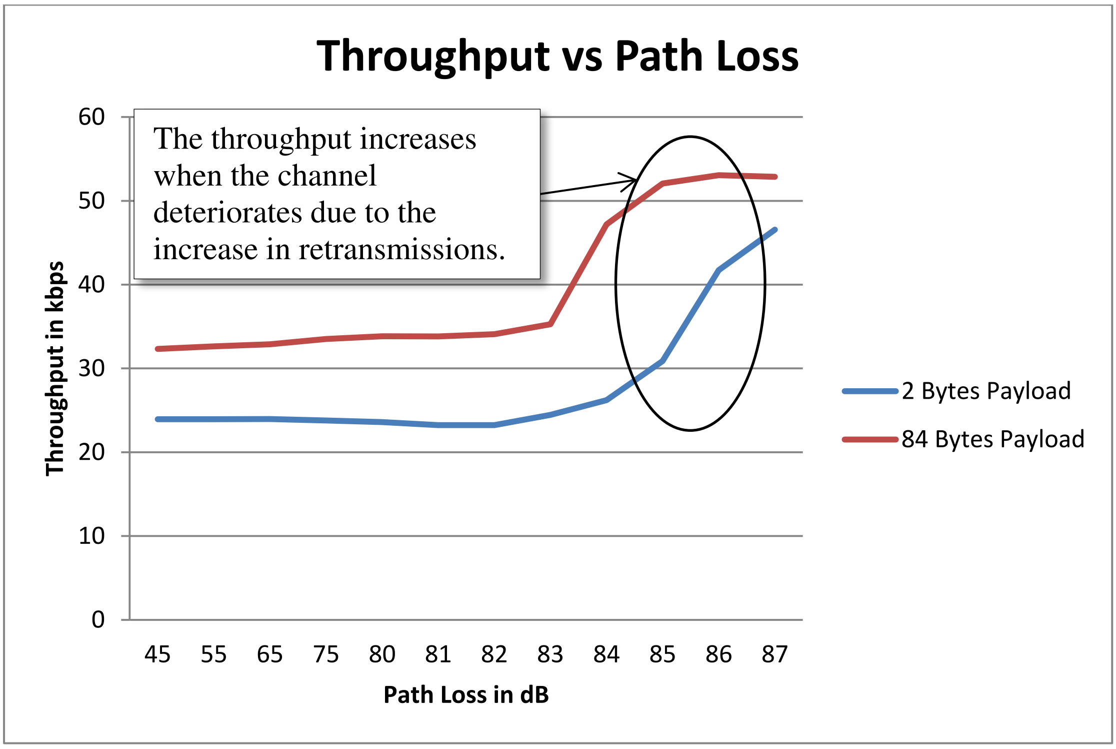


Figure 8.10: ZigBee throughput vs. path loss test result

The above figure shows that given the software interface provided by the CC2530 Z-Stack ZNP API, the maximum throughput achievable is roughly 52 kbps. We can see that this throughput value is towards the end of the graph where the channel is substantially deteriorated. The increase in throughput comes from re-transmissions in the channel.

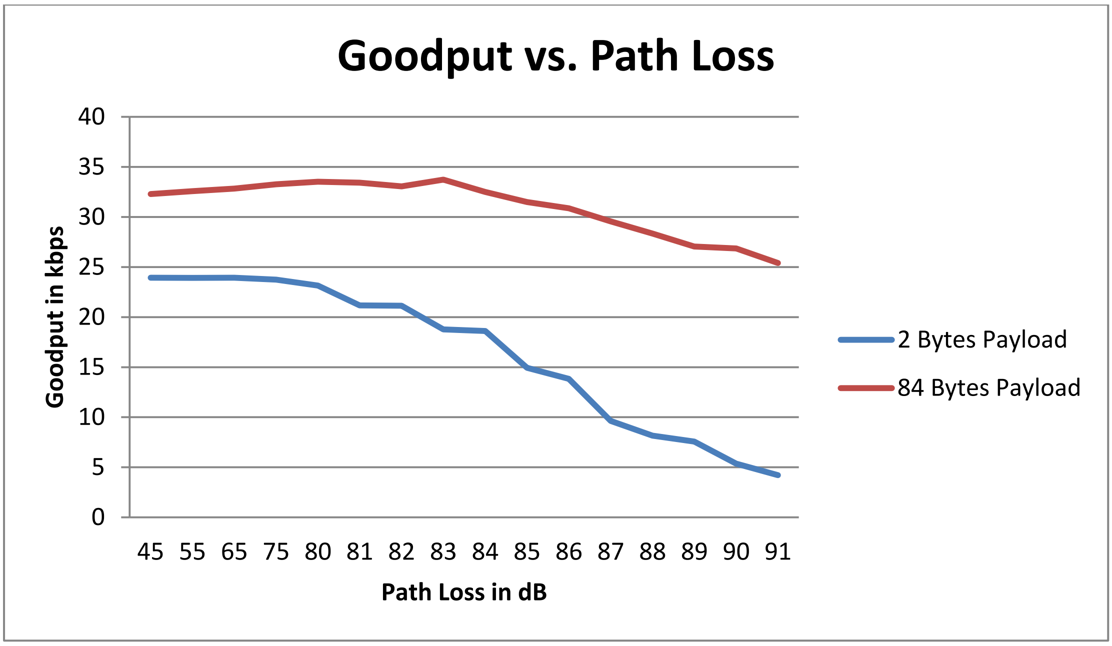


Figure 8.11: ZigBee goodput vs. path loss test result

The term goodput denotes the number of useful information bits that were transmitted per unit time from a source to the destination. As we can see from the above figure the goodput falls as the channel deteriorates. This is obvious as packet retransmissions and packet losses climb when the channel deteriorates. The above figure also shows us that the current hardware will be able to receive transmissions reliably down to 85 dB of path loss.

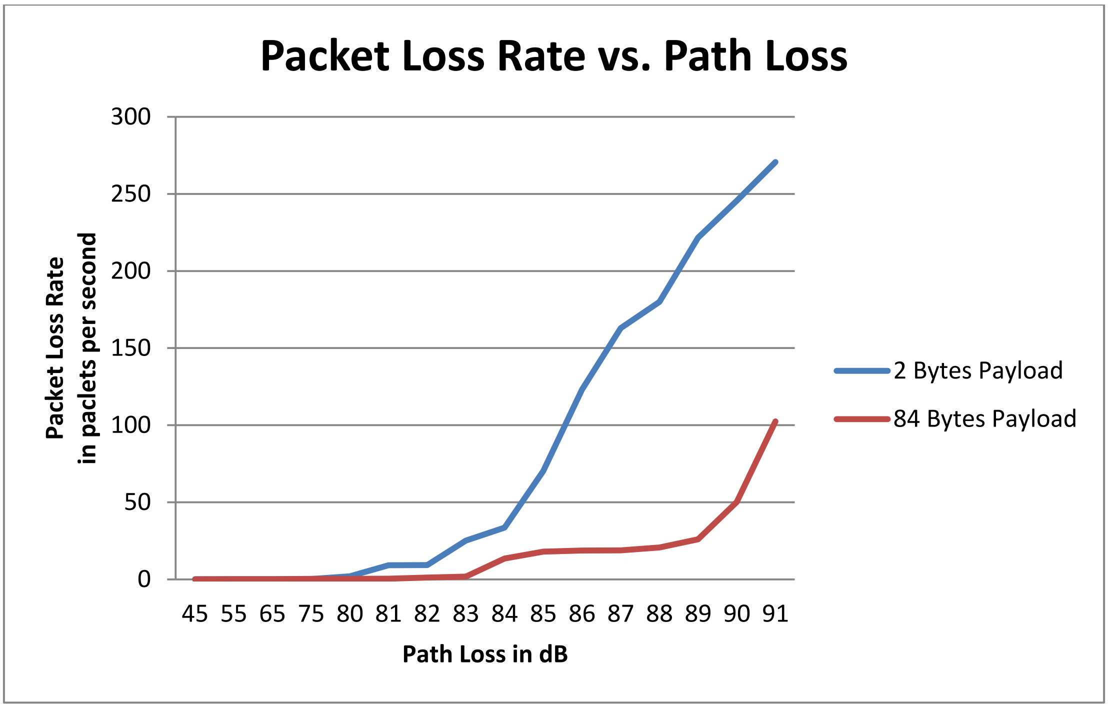


Figure 8.12: ZigBee packet loss rate vs. path loss test results

The above result confirms what the test results for throughput and goodput shows us. As the channel conditions deteriorate the number of retransmissions and packet loss increases. This means there is more traffic being generated and transmitted into the channel by the sender, this results in an increase in throughput. But the number of transmissions actually reaching the intended recipient goes down due to a bad channel and thus the goodput decreases.

8.1.5. Tests Result Analysis

In this section the analysis of the results obtained from the Wi-Fi and ZigBee tests will be carried out to determine performance improvements that the HTNMote offers over currently available sensor hardware platforms.

8.1.5.1. Wi-Fi Results Analysis

The Stargate gateway node [45] is a hardware platform that can interface with the MICAz motes and provide a Wi-Fi gateway using a Wi-Fi Network Interface Card in a PCMCIA slot provided in the hardware. The Wi-Fi transfer speeds on the Stargate range from 1.5 Mbps to 3.75 Mbps depending on the Wi-Fi settings on both the sender and receiver sides [66]. This is achieved at a cost of increased power consumption as the Stargate operates at 5 V and consumes on an average 300 mA of current when Wi-Fi is active [67].

A more recent development is the Wi-Fi expansion board [68] from Libelium for the Waspmote sensor platform. This board interfaces to the microcontroller core of the Waspmote using UART interface. This itself is a bottleneck for data transfer between the microcontroller and the Wi-Fi radio. In a practical scenario, the current consumed by the Waspmote hardware and the Wi-Fi board working together is close to 200 mA [68]. The maximum transmission throughput achievable when the UART baud rate is set to 57600 is 15.68 kbps.

The Wi-Fi results show us that there is a huge performance gain with HTNMote when the parameter under consideration is the throughput. The HTNMote provides a maximum throughput of close to 2.2 Mbps. This is an improvement of 140 times for similar current consumption.

Also when it comes to comparison with the higher power Stargate gateway, it must be kept in mind that the maximum 1.75 times increase in throughput comes at a significant increase in current consumption at a higher voltage, that is the power consumption is much higher.

8.1.5.2. ZigBee Results Analysis

The goodput test results show that the HTNMote hardware is capable of delivering up to 35 kbps in real-world application scenario at a path loss of 87 dB, which is comparable to the performance of MICAz and Waspmotes.

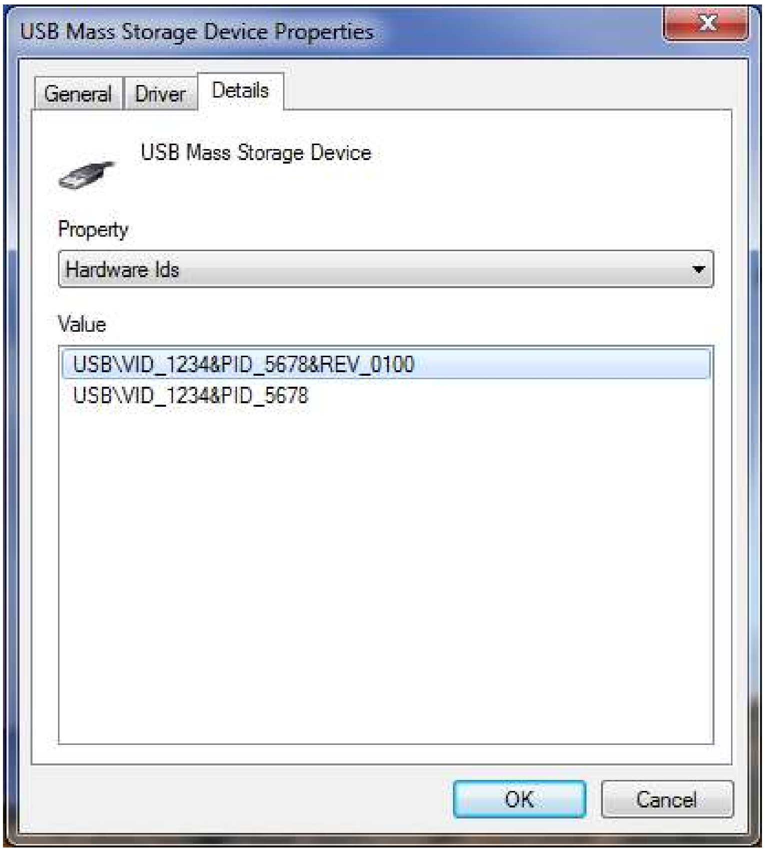
This is expected since the radio chip used on the MICAz is CC2420 from Texas Instruments and is a predecessor of the CC2530 used on the HTNMote but with similar architecture. The XBee ZigBee module [69] used with the Waspmote is theoretically capable of delivering better performance but is severely restricted by the UART interface between the microcontroller and the XBee module.

8.2. Functionality Tests

The USB, RS-232, LCD and SDHC are components of the hardware whose performance is not critical to the goals of the hardware but their proper functioning is. The following demonstrate the result of running the software implemented to make each of these modules work.

8.2.1. USB Test

The hardware is given a USB VID of 0x1234 and a PID of 0x5678. Upon execution of the USB code, the device is plugged into a computer using a USB cable. The device enumerates and the proper VID and PID can be seen from the Device Manager console.



USB Mass Storage Device Properties

USB Mass Storage Device

Property: Hardware Ids

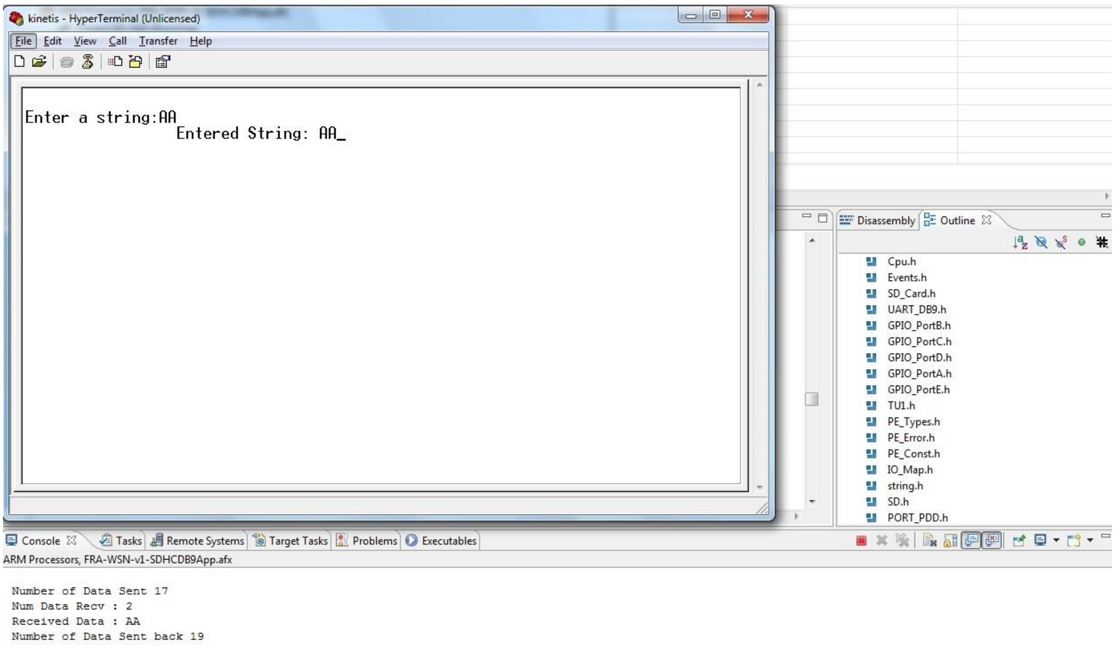
Value:

```
USB\VID_1234&PID_5678&REV_0100
USB\VID_1234&PID_5678
```

Figure 8.13: Device Manager screenshot for USB functionality test

8.2.2. RS-232 Test

When the software for RS-232 communication is executed, it is observed both on the Hyperterminal screen on the computer and on the CodeWarrior console that bi-directional communication is achieved successfully.



```
Enter a string:AA
                 Entered String: AA_
```

```
Number of Data Sent 17
Num Data Recv : 2
Received Data : AA
Number of Data Sent back 19
```

Figure 8.14: RS-232 bi-directional communication screenshot

8.2.3. LCD Test

The LCD successfully comes up after the code is executed as can be seen in the figure below:

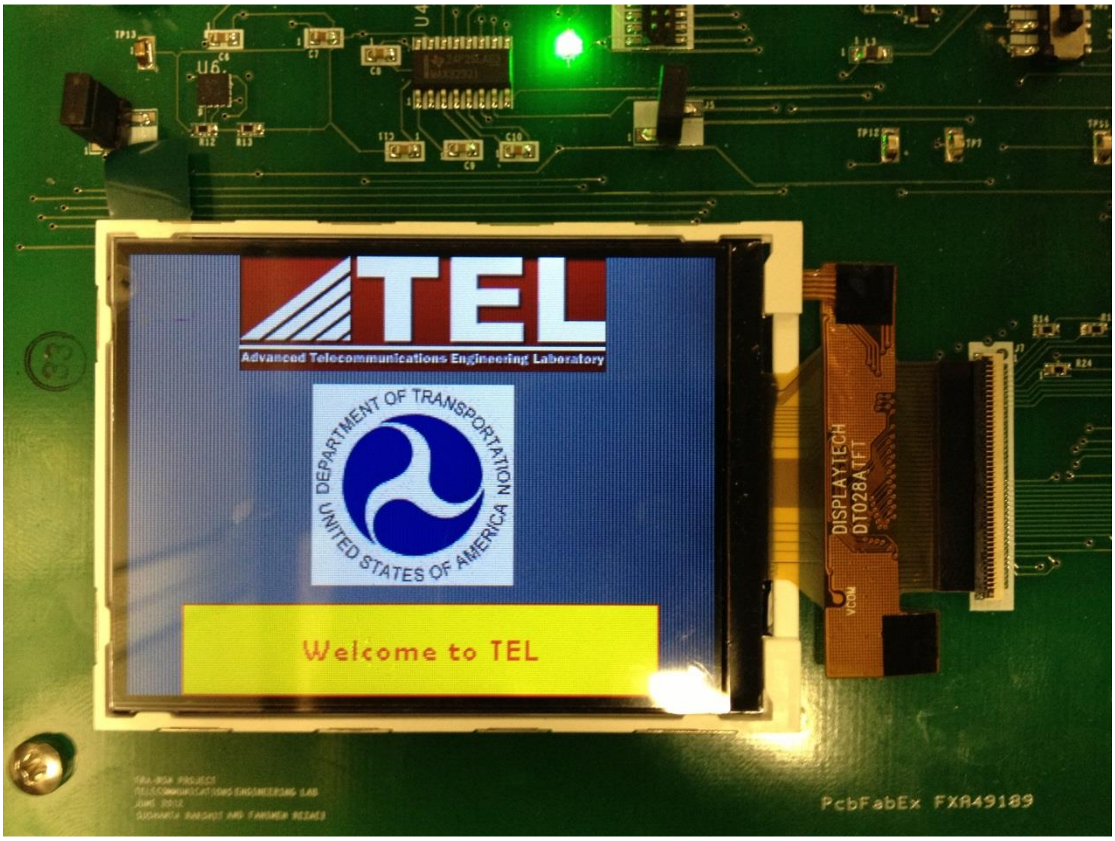
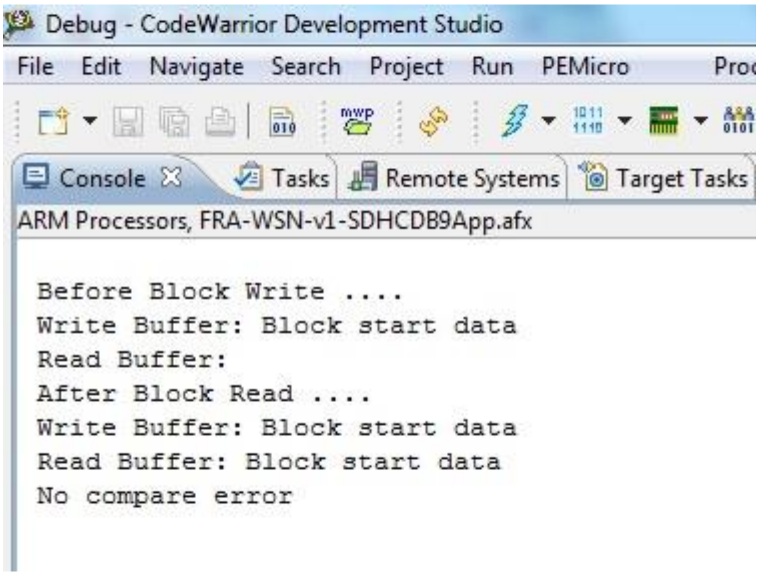


Figure 8.15: LCD output after code execution

8.2.4. SDHC Test

The SDHC software is executed to write a string to the SD card and then it is read back and compared with what was written. The result of the execution is shown in the figure below:


Debug - CodeWarrior Development Studio
File Edit Navigate Search Project Run PEMicro Proc
Console
Tasks
Remote Systems
Target Tasks

ARM Processors, FRA-WSN-v1-SDHCDB9App.afx

```
Before Block Write ....
Write Buffer: Block start data
Read Buffer:
After Block Read ....
Write Buffer: Block start data
Read Buffer: Block start data
No compare error
```

Figure 8.16: Partial screenshot of CodeWarrior console after SDHC code has executed

Chapter 9. SUMMARY, CONCLUSION AND FUTURE WORK

9.1. Summary and Conclusion

The Federal Railroad Administration is looking at WSN to monitor freight rail in real-time or near real-time. The freight rail is an important contributor to the health of the United States economy. Freight rail is also a greener mode of transport. The current methods of monitoring freight rail, like wayside monitoring etc., suffer from the drawbacks of not being able to monitor a large parameter set and the monitoring is not in real-time.

WSN is a natural choice in achieving the real-time monitoring goals. ZigBee is the protocol of choice for such deployments owing to its low cost and low power consumption.

However, the railroads deployment scenario is very unique when it comes to the topology in which the network exists. The long linear chain-like topology is one for which ZigBee was not designed. Hence, the use of ZigBee in such scenarios present significant problems.

These problems include:

a) Synchronization Delay

b) Route discovery issues

c) Packet and link loss

d) Lack of Quality-of-Service

e) Data forwarding and aggregation errors

f) Skewed network lifetime issues

A solution has been suggested by the Advanced Telecommunication Engineering laboratory at the University of Nebraska-Lincoln which addresses the above problems. This solution is called the Hybrid Technology Networking protocol. This envisions ZigBee to operate in small clusters and the data forwarding from one cluster to the next to happen using Wi-Fi. This has been shown to improve end-to-end throughput, balance power consumption and also is standards based.

The current available sensor hardware platform does not have an integrated solution for the application of Hybrid Technology Networking. Hence, the need to design an integrated standards-based sensor hardware platform that will at least be equivalent in performance.

The design of HTNMote is presented. It combines a Cortex M-4 microcontroller with both Wi-Fi and ZigBee radio capabilities residing on the same board. SD Card storage facility is present. Communication with the external world is carried out using RS-232 and USB apart from Wi-Fi and ZigBee.

The test results of the HTNMote hardware show that the Wi-Fi capability is far better than any solution available in the market now. The ZigBee performance is comparable to any other hardware currently available. Coupled with the possibilities of implementing extremely low power consuming code, as all the peripherals support power saving features, the HTNMote is an ideal platform to implement the Hybrid Technology Networking protocol.

9.2. Future Work

The initial version of HTNMote has been shown to be quite capable of handling the demands of the Hybrid Technology Networking protocol. It is already better than comparable products available in the market.

In future further research on HTNMote will be conducted with the following goals in mind:

a) Miniaturization: The next goal is to miniaturize the HTNMote hardware by doing away with components that are not required in the field. Miniaturization will further reduce latency and current consumption.

b) Improving performance: The current implementation has shown that even though capable hardware and a fast interface to that hardware exists but still performance bottlenecks come from the software implementation. In the next revision, software optimization with an eye to throughput and other performance improvement will be done.

c) Large deployment tests: In future, large deployments of HTNMotes will be carried out to test how the HTNMotes perform when they are in a network with several other HTNMotes.

REFERENCES

[1] http://www.fra.dot.gov/Page/P0362

[2] http://nationalatlas.gov/articles/transportation/a_freightrr.html

[3] AAR, Class I Railroad Statistics, http://www.aar.org/PubCommon/Documents/AboutTheIndustry/Statistics.pdf

[5] http://www.railway-technology.com/contractors/track/trackside-intelligence

[6] http://en.wikipedia.org/wiki/Automatic_equipment_identification

[7] http://www.railway-technology.com/contractors/signal/transcore/transcore2.html

[8] http://en.wikipedia.org/wiki/IEEE_802.11

[9] Edwards, M.C.; Donelson, J., III; Zavis, W.M.; Prabhakaran, A.; Brabb, D.C.; Jackson, A.S., "Improving freight rail safety with on-board monitoring and control systems," Rail Conference, 2005. Proceedings of the 2005 ASME/IEEE Joint , vol., no., pp.117,122, 16-18 March 2005

[10] http://www.gps.gov/

[11] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. 2002. Wireless sensor networks: a survey. Comput. Netw. 38, 4 (March 2002), 393-422.

[12] IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. (2012 revision). IEEE-SA. 5 April 2012.

[13] http://www.wi-fi.org

[14] http://www.tutorial-reports.com/wireless/wlanwifi/wifi_architecture.php

[15] http://damayanthiherath.wordpress.com/a-survey-on-ieee-802-11-wireless-lan-standards-and-physical-layer-issues

[16] http://it.siit.tu.ac.th/~u5322793290/its323/2.html

[17] http://www.radio-electronics.com/info/wireless/wi-fi/ieee-802-11-standards-

tutorial.php

[18] Lashkari, A.H.; Danesh, M.M.S.; Samadi, B., "A survey on wireless security protocols (WEP, WPA and WPA2/802.11i)," Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on , vol., no., pp.48,52, 8-11 Aug. 2009

[19] Majstor, F., "WLAN security threats & solutions," Local Computer Networks, 2003. LCN '03. Proceedings. 28th Annual IEEE International Conference on , vol., no., pp.650,, 20-24 Oct. 2003

[20] Sandirigama, M.; Idamekorala, R., "Security weaknesses of WEP protocol IEEE 802.11b and enhancing the security with dynamic keys," Science and Technology for Humanity (TIC-STH), 2009 IEEE Toronto International Conference , vol., no., pp.433,438, 26-27 Sept. 2009

[21] IEEE 802.15.4, “Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR WPANs)”, IEEE SA Standards Board, 2003.

[22] http://www.radio-electronics.com/info/wireless/ieee-802-15-4/wireless-standard-technology.php

[23] http://standards.ieee.org/about/get/802/802.15.html

[24] https://en.wikipedia.org/wiki/IEEE_802.15.4

[25] http://standards.ieee.org/about/get/802/802.2.html

[26] http://www.sensor-networks.org/index.php?page=0903503549

[27] Mulligan, Geoff, "The 6LoWPAN architecture", EmNets '07: Proceedings of the 4th workshop on Embedded networked sensors, ACM, 2007

[28] http://www.isa.org/

[29] David Flowers, Yifeng Yang, "The MiWi Wireless Networking Protocol is a simple protocol designed for low data rate, short distance, low-cost networks.", AN1066, Microchip.

[30] http://www.techonlineindia.com/article/12-02-09/an_introduction_to_wireless_sensor_network_concepts.aspx

[31] http://www.radio-electronics.com/info/wireless/ieee-802-15-4/mesh-networking-topology-topologies.php

[32] http://www.daintree.net/downloads/whitepapers/ZigBee_primer.pdf

[33] ZigBee Alliance: ZigBee Specification, ZigBee document 053474r13, 1 December 2006.

[34] Ata Elahi Ph.D.; Adam Gschwender,"ZigBee Wireless Sensor and Control Network",ISBN-10: 0-13-713485-1,Publisher: Prentice Hal,October 29, 2009

[35] http://www.jennic.com/elearning/ZigBee/files/html/module2/module2-5.htm

[36] S. M. Rakshit, M. Hempel, H. Sharif, J. Punwani, M. Stewart, and S. Mehrvarzi, “Challenges in current Wireless Sensor Technology for Railcar Status Monitoring for North America's Freight Railroad Industry”, in ASME/ASCE/IEEE Joint Rail Conference, 2012, pp. 1-9.

[37] J. Zheng, M. J. Lee, “Will IEEE 802.15.4 make ubiquitous networking a reality? A discussion on a potential low power low bit rate standard,” IEEE Communications Magazine, 2004, 42:140–146.

[38] P. Mahasukhon, H. Sharif, M. Hempel, T. Zhou, T.Ma, P. L. Shrestha, “A study on energy efficient multi-tier multi-hop wireless sensor networks for freight-train monitoring,” Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International,pp.297-301, 4-8 July 2011.

[39] P. Mahasukhon, H. Sharif, M. Hempel, T. Zhou, W. Wang, T.Ma, "Multi-Tier Multi-Hop Routing in Large-Scale Wireless Sensor Networks for Freight Train Monitoring," ITS World Congress, 2010.

[40] P. Mahasukhon, H. Sharif, M. Hempel, T. Zhou, W. Wang, T. Ma, "Multi-Tier Multi-Hop Routing in Large-Scale Wireless Sensor Networks for Real-Time Monitoring," IEEE Sensors, 2010.

[41] F. Osterlind, A. Dunkels, "Approaching the Maximum 802.15.4 Multi-hop Throughput," SICS Technical Report T2008:05, 2008.

[42] J. Misic, S. Shafi, V. B. Misic, "Avoiding the Bottlenecks in the MAC Layer in 802.15.4 Low-Rate WPAN", in Proc. of 11th International Conference on Parallel and Distributed Systems Workshops, 2005, pp. 363-367.

[43] S. M. Rakshit, M. Hempel, H. Sharif, J. Punwani, and M. Stewart, "Hybrid Technology Networking: A Novel Wireless Networking Approach for Real-Time Railcar Status Monitoring", in ASME Rail Transportation Division Fall Technical Conference, 2012, pp. 1-7.

[44] MICAz datasheet at MEMSIC: http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7- datasheets.html?download=148%3Amicaz

[45] www.willow.co.uk/html/spb400-_stargate_gateway.html

[46] http://www.libelium.com/products/waspmote

[47] LT1129-3.3 datasheet: www.linear.com/docs/2235

[48] TPS73701 datasheet: www.ti.com/product/tps73701

[49] ECX-71 32.768 kHz crystal: http://www.ecsxtal.com/store/pdf/ecx-71.pdf

[50] 4 MHz SMD Quartz crystal: http://www.ecsxtal.com/store/pdf/ecx-32.pdf

[51] Murata 4 MHz ceramic resonator:
http://datasheetz.com/data/Crystals%20and%20Oscillators/Resonators/490-1217-1-datasheetz.html

[52] Freescale Kinetis K60 microcontroller:
http://www.freescale.com/webapp/sps/site/taxonomy.jsp?code=K60_ETHERNET_CRYPTO_MCU

[53] Redpine signals Wi-Fi module:
http://www.redpinesignals.us/Modules_&_M2M_systems/Modules/Wi-Fi_Modules/Connect-io-n/RS9110-N-11-22.php

[54] CC2530: www.ti.com/product/cc2530

[55] SD Card Association: https://www.sdcard.org/

[56] USB standard: http://www.usb.org/developers/docs/

[57] RS-232 transceiver chip: www.ti.com/product/max3232

[58] Displaytech LCD: https://www.displaytech-us.com/2-8-inch-tft

[59] Board to Board Hirose connector datasheet:
http://www.mouser.com/ds/2/185/e53700036-7084.pdf

[60] Freescale CodeWarrior:
http://www.freescale.com/webapp/sps/site/homepage.jsp?code=CW_HOME

[61] Freescale eGUI:
http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=EGUI&tid=vanEGUI

[62] ILI9341 display controller datasheet:
http://www.displayfuture.com/Display/datasheet/controller/ILI9341.pdf

[63] Azimuth Systems, http://www.azimuthsystems.com/platforms-channel-

400wb.htm

[64] http://www.itu.int/en/Pages/default.aspx

[65] TI ZigBee Packet Sniffer: http://www.ti.com/tool/cc2531emk

[66] http://platformx.sourceforge.net/Documents/nuts/WiFiFAQ.html

[67] Kumar, A.; Namboothiri, P.G.; Deshpande, S.; Vidhyadharan, S.; Sivalingam, K.M.; Murty, S.A.V.S., "Testbed based throughput analysis in a Wireless Sensor Network," Communications (NCC), 2012 National Conference on , vol., no., pp.1,5, 3-5 Feb. 2012

[68] Wi-Fi Expansion Board: http://www.libelium.com/expansion_radio_board/

[69] Xbee Module datasheet: www.digi.com/pdf/ds_xbeezbmodules.pdf

APPENDIX A

Please contact the author at sushanta.rakshit@huskers.unl.edu for more information on this section.

APPENDIX B

Please contact the author at sushanta.rakshit@huskers.unl.edu for more information on this section.

APPENDIX C

Please contact the author at sushanta.rakshit@huskers.unl.edu for more information on this section.