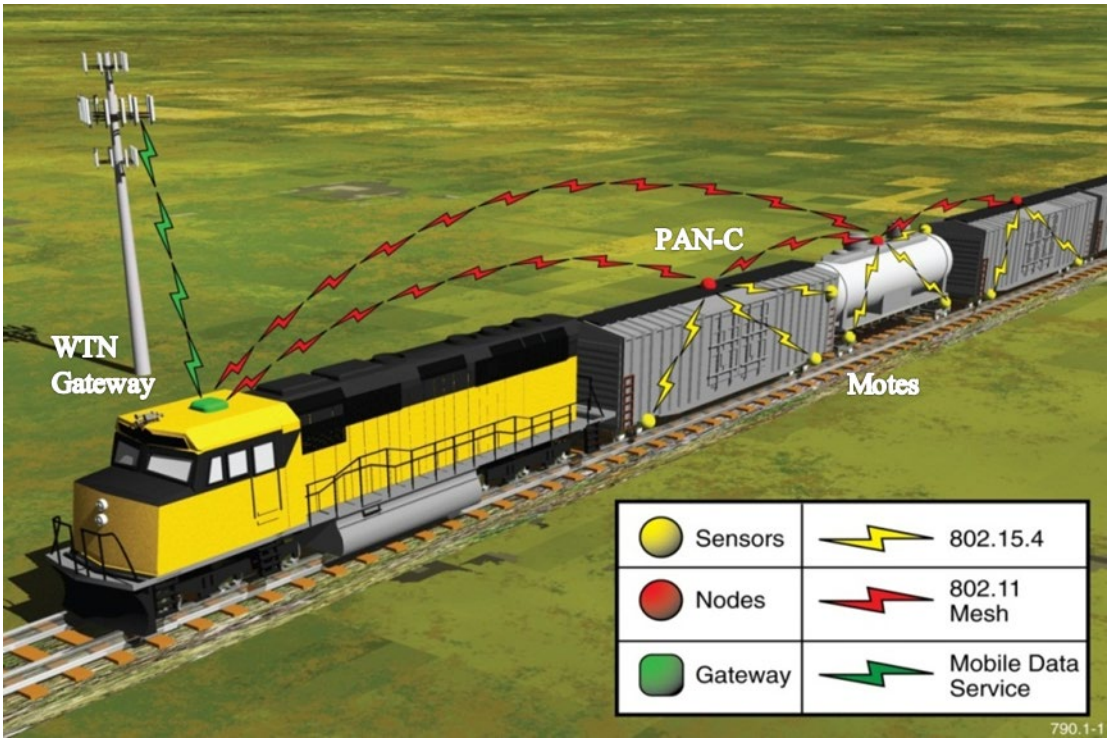
U.S. Department of Transportation

Federal Railroad Administration

Onboard Monitoring and Control – Wireless Sensor Networks Systems Integration

Office of Research,
Development,
and Technology
Washington, DC 20590



DOT/FRA/ORD-21/21

Final Report
July 2021

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	July 2021	Technical Report

4. TITLE AND SUBTITLE
Onboard Monitoring and Control –
Wireless Sensor Networks Systems Integration

5. FUNDING NUMBERS
DTFR53-12-D-00003L
Task Order 0011

6. AUTHOR(S)
Tom Campbell*, Dr. Hamid Sharif**

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
*QinetiQ North America
350 Second Avenue
Waltham, MA 02451-1196

**University of Nebraska – Lincoln
Telecommunications Engineering Laboratory
1110 S 67th Street
Omaha, NE 68182

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Department of Transportation
Federal Railroad Administration
Office of Research, Development, and Technology
Washington, DC 20590

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
DOT/FRA/ORD-21/21

11. SUPPLEMENTARY NOTES
COR: Tarek Omar, Ph.D.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
This document is available to the public through the FRA website.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)
A practical wireless train network (WTN) for freight trains is defined. A hybrid technology network utilizing ZigBee within each railcar and WiFi to create an ad hoc train network is researched. Motes and PAN-coordinators have been fabricated for demonstration testing. Network software were developed and embedded for testing. A specialized wheel bearing sensor was developed; it contains a thermistor, a 3-axis accelerometer, and a microphone. The sensor components were fabricated and the firmware has been developed. Raw data can be acquired or the fast Fourier transforms can be compared to thresholds for alerts. A demonstration was designed and built to activate and deactivate electronically deployed hand brakes developed by Sharma and Associates. This can enable the activation of hand brakes remotely from the locomotive for every railcar. The WTN was developed to self-initiate as the train leaves the departure yard. Activation is based on the attachment of the End-of-Train device and forward acceleration in each railcar. WiFi beacons from the gateway are detected by each railcar in series as they accelerate forward. That timing determines the railcar order. Direction of forward accelerations determines the railcar orientation. The data are automatically assembled as the train departs and is uploaded to the enterprise.

14. SUBJECT TERMS
Wireless train network, hybrid technology network, wheel bearing sensor, train health monitoring, wireless sensor network, ad hoc network, freight train network

15. NUMBER OF PAGES
70

16. PRICE CODE

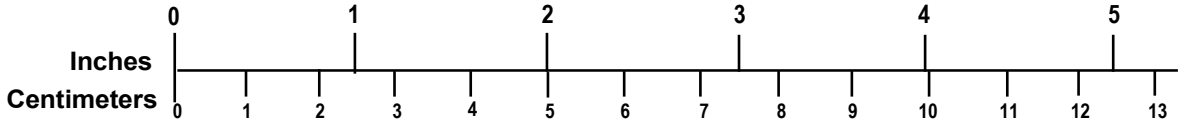
17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18; 298-102

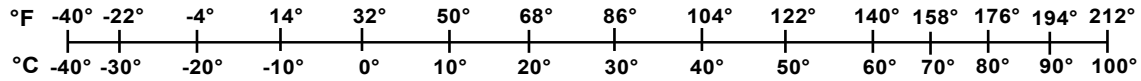
METRIC/ENGLISH CONVERSION FACTORS

ENGLISH TO METRIC	METRIC TO ENGLISH
LENGTH (APPROXIMATE)	**LENGTH (APPROXIMATE)**
1 inch (in) = 2.5 centimeters (cm)	1 millimeter (mm) = 0.04 inch (in)
1 foot (ft) = 30 centimeters (cm)	1 centimeter (cm) = 0.4 inch (in)
1 yard (yd) = 0.9 meter (m)	1 meter (m) = 3.3 feet (ft)
1 mile (mi) = 1.6 kilometers (km)	1 meter (m) = 1.1 yards (yd)
	1 kilometer (km) = 0.6 mile (mi)
AREA (APPROXIMATE)	**AREA (APPROXIMATE)**
1 square inch (sq in, in^2) = 6.5 square centimeters (cm^2)	1 square centimeter (cm^2) = 0.16 square inch (sq in, in^2)
1 square foot (sq ft, ft^2) = 0.09 square meter (m^2)	1 square meter (m^2) = 1.2 square yards (sq yd, yd^2)
1 square yard (sq yd, yd^2) = 0.8 square meter (m^2)	1 square kilometer (km^2) = 0.4 square mile (sq mi, mi^2)
1 square mile (sq mi, mi^2) = 2.6 square kilometers (km^2)	10,000 square meters (m^2) = 1 hectare (ha) = 2.5 acres
1 acre = 0.4 hectare (he) = 4,000 square meters (m^2)	
MASS - WEIGHT (APPROXIMATE)	**MASS - WEIGHT (APPROXIMATE)**
1 ounce (oz) = 28 grams (gm)	1 gram (gm) = 0.036 ounce (oz)
1 pound (lb) = 0.45 kilogram (kg)	1 kilogram (kg) = 2.2 pounds (lb)
1 short ton = 2,000 pounds (lb) = 0.9 tonne (t)	1 tonne (t) = 1,000 kilograms (kg) = 1.1 short tons
VOLUME (APPROXIMATE)	**VOLUME (APPROXIMATE)**
1 teaspoon (tsp) = 5 milliliters (ml)	1 milliliter (ml) = 0.03 fluid ounce (fl oz)
1 tablespoon (tbsp) = 15 milliliters (ml)	1 liter (l) = 2.1 pints (pt)
1 fluid ounce (fl oz) = 30 milliliters (ml)	1 liter (l) = 1.06 quarts (qt)
1 cup (c) = 0.24 liter (l)	1 liter (l) = 0.26 gallon (gal)
1 pint (pt) = 0.47 liter (l)	
1 quart (qt) = 0.96 liter (l)	
1 gallon (gal) = 3.8 liters (l)	
1 cubic foot (cu ft, ft^3) = 0.03 cubic meter (m^3)	1 cubic meter (m^3) = 36 cubic feet (cu ft, ft^3)
1 cubic yard (cu yd, yd^3) = 0.76 cubic meter (m^3)	1 cubic meter (m^3) = 1.3 cubic yards (cu yd, yd^3)
TEMPERATURE (EXACT)	**TEMPERATURE (EXACT)**
[(x-32)(5/9)] °F = y °C	[(9/5) y + 32] °C = x °F

QUICK INCH - CENTIMETER LENGTH CONVERSION



QUICK FAHRENHEIT - CELSIUS TEMPERATURE CONVERSIO



For more exact and or other conversion factors, see NIST Miscellaneous Publication 286, Units of Weights and Measures. Price $2.50 SD Catalog No. C13 10286

Updated 6/17/98

Contents

Illustrations

Tables

Executive Summary

The goal of the proposed research is to develop a wireless train network (WTN) and associated hardware/firmware that can provide real-time monitoring of railcar health, issue alerts, control actuators, and provide automatic train consist information to the railroad. This was accomplished by developing cost-effective sensor and actuator network motes that can be located throughout a railcar to monitor critical parameters such as wheel bearing condition, accelerations for feedback on hunting, status of hatches and doors, and lading condition. The motes can also be used to control actuators, such as remotely controlling electronically deployed hand brakes (EDHB).

WTN solutions for the railroad environment require an overall system of sensors, motes, and a locomotive gateway that provides near-real-time data transmission from individual sensors to the data collection gateway. By simplest definition, "motes" are small computers for remote sensing. In a mesh network of low-power sensor nodes, motes are used to create wireless sensor networks capable of processing, gathering, and communicating information gathered from various sensors.

The purpose of the railroad WTN is to continuously collect sensor data concerning the operating conditions on a railcar. The WSN, based on that data, can deliver early indicators, warnings, and alerts of interest in near real-time according to configurable rules to any or all of train operators, railcar and track owners, and cargo owners.

The WTN system is composed of four major components:

- Sensor: The component that performs the functional task. A network will support a multitude of sensors.
- Mote or Node: The communication component that enables the sensor to transmit its information across the network. The mote may be integrated with a sensor component, or separate from the sensor.
- Railcar Network Coordinator: The personal area network coordinator, or PAN-C, accumulates all sensor data on a railcar and communicates that information to the locomotive gateway. All sensors and their motes must be able to communicate with their PAN-C.
- WTN Gateway: The centralized location where all PAN-C nodes will transmit their data. All nodes must be able to communicate with the WTN gateway. In train operations, the gateway is a unit on the locomotive at the front of the train.

Transportation Technology Center, Inc. (TTCI) developed the hardware and firmware for testing the WTN. TTCI also developed a wheel bearing sensor containing a thermistor, 3-axis accelerometer, and microphone for operation through motes on the WTN. Additionally, hardware was developed for actuating the EDHBs remotely from the locomotive.

1 Introduction

The program goal is to provide an open-architecture, ad hoc, wireless sensor network (WSN) to enable the real-time monitoring of critical railcar parameters such as wheel bearing condition, and the actuation of components such as electronically deployed hand brakes (EDHB), from the locomotive. An in-transit sensing system that continuously monitors factors such as wheel bearing temperature and vibration could be tied into predictive maintenance to reduce wheel failures and enable trains to safely operate at higher speeds. Further, it could monitor the environmental conditions of sensitive cargo, provide rail-container security information, and provide real-time location for logistic tracking. Other applications for a wireless sensor network include acoustic bearing detectors, air and brake pressure monitors, automatic railcar order and orientation, intrusion and tamper detection, cargo data, and consist configuration time history data.

1.1 Background

Previous work at the Telecommunications Engineering Laboratory of the University of Nebraska-Lincoln showed that a wireless sensor network could be used to monitor sensors and control actuators. Low-power networks, such as ZigBee, were shown to lose data packets with many data hops to transmit information along a long freight train. A hybrid system was developed where low-power ZigBee networks could be used for local clusters, such as on individual railcars. The long-distance data hopping would then be conducted via an ad hoc WiFi network. This combination of technologies was its hybrid technology network (HTN).

1.2 Objectives

The program goal is to adapt the HTN approach to a freight train wireless network and demonstrate that network with representative sensors and actuators. The sensors selected were a suite of wheel bearing sensors, and the actuator was the electronically deployed hand brake (EDHB) developed by Sharma and Associates. The developed network hardware and firmware will be tested at the Transportation Technology Center. The program will develop the detailed method for network formation and the hardware suitable for Transportation Technology Center, Inc. (TTCI) field testing.

1.3 Overall Approach

The program explores using existing network systems where possible. The emphasis is on the network implementation and demonstration of its practical application on freight trains. Consideration of typical freight train application and best use for the railroad back office will be important drivers. The research team will design and build sufficient motes for a realistic test at TTCI. Where sensors were not available, custom sensors will be designed, built, and tested.

1.4 Organization of the Report

Section 2 describes the wireless train network (WTN). Section 3 describes the system architecture. Section 4 describes the network hardware, and Section 5 describes the firmware. Section 6 summarizes the results and presents the conclusions and next steps. A list of references is included in Section 7, and Abbreviations and Acronyms are included at the end of this report. The Appendix contains the network mode of operation and draft specifications.

2 Wireless Sensor Network System Description

In WSNs the goal is to realize an event-driven monitoring and actuation solution comprised of small low-power devices sometimes also referred to as "motes." These motes represent nodes in the network architecture and are used for data acquisition from attached sensing devices, for actuating attached devices, and for wireless communications between nodes of the network. The primary goal in WSN research is to minimize the power requirements without impacting system operation. Ideally, in event-driven WSN's data is communicated from sensors to data sinks or gateways in near-real-time, in order to be able to act upon critical events in a timely fashion. Typically, WSNs are designed to operate in a mesh-like network topology, enabling every node to transmit, receive and relay information.

For WSNs specifically targeting freight railcar monitoring, continuous monitoring and operation are vital. Railcar component failure and other critical events can happen at any time. Based on collected sensor information, the WSN can then issue early fault indicators, warnings, and alerts of interest in near-real-time, and in accordance with configurable rules, to the train operators, the railcar owner and/or track owners, as well as the shipper of the cargo.

The WSN system is composed of three major components:

- Sensor: The component that performs the functional task. A network will support a multitude of sensors.
- Mote or Node: The communication component that enables the sensor to transmit its information across the network. The mote may be integrated with a sensor component, or separate from the sensor.
- Gateway: The centralized location where all sensors will transmit their data. All sensors must be able to communicate with the gateway. In train operations, the gateway is a unit on the locomotive at the front of the train.

The WSN concept involves mounting motes on railcars where they can monitor for unusual conditions like overheated wheel bearings. The motes form a radio communications network with multi-hop communications. The motes use this network to communicate with gateways – more powerful laptop-sized computers with longer-range radios and broad communications capabilities. The gateways can be mounted in the locomotive, enabling continuous condition monitoring of the attached consist, or on the track wayside.

2.1 Ad Hoc Mesh Network

The Telecommunications Engineering Laboratory (TEL) at the University of Nebraska-Lincoln has independently conducted extensive theoretical and laboratory tests and confirmed, similar to other industry evaluations, that ZigBee is not a feasible technology candidate for realizing end-to-end communication throughout the entire length of a freight train. To form a ZigBee network that encompasses all sensors in a freight train, the network is comprised of an exceedingly large number of hops. This chain-like topology creates numerous problems ranging from a short maximum hop count, unacceptable delays in network formation, synchronization, and route discovery, to severe limitations in aggregate data throughput and data delivery reliability.

One of the biggest problems encountered in ZigBee is the limitation to the maximum device depth, which is closely correlated with the maximum-supported hop count. In ZigBee, the

maximum network depth is 7, which results in a maximum hop count of 14 if the coordinator is the central device. This is clearly of insufficient length for even a short freight train. Even in ZigBee Pro, which in theory does not impose a network depth limit due to its stochastic addressing scheme, a practical limitation is encountered due to route management that effectively restricts the device depth to 20 for most implementations, and hence results in a maximum hop count of 40 – still far less than would be required for freight train. With the very short frames in ZigBee, route discovery is restricted to short routes only, in order to be able to report the discovered route information to the requesting node. If the route length exceeds the carrying capacity of the frame, route discovery will fail entirely, as shown in Figure 1.

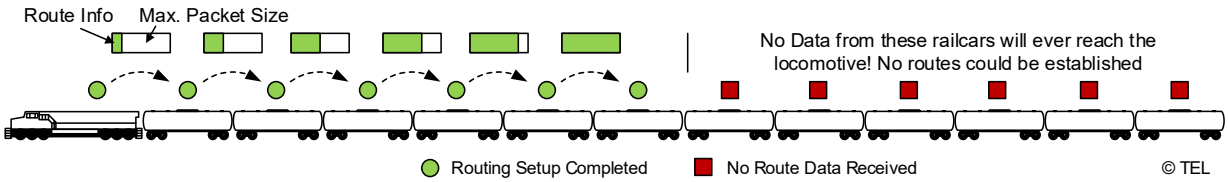


Figure 1 Limitation of Nodes Due to Packet Size

Another major problem is the limited throughput provided by ZigBee. Its theoretical limit is 250 kilobits/second (kbps), far less than data rates commonly provided by WiFi. With multiple neighboring sensors operating on the same channel, such as in the chain-like topology for freight trains, these nodes are effectively sharing the same channel and have to divide the total channel throughput among them. Combined with the data aggregation required to both forward data through multiple sensors in order for it to reach the destination and start the delivery process for the data generated by the sensor itself, the link quickly exceeds its throughput capacity and hence data is forced to be discarded, as illustrated in Figure 2.

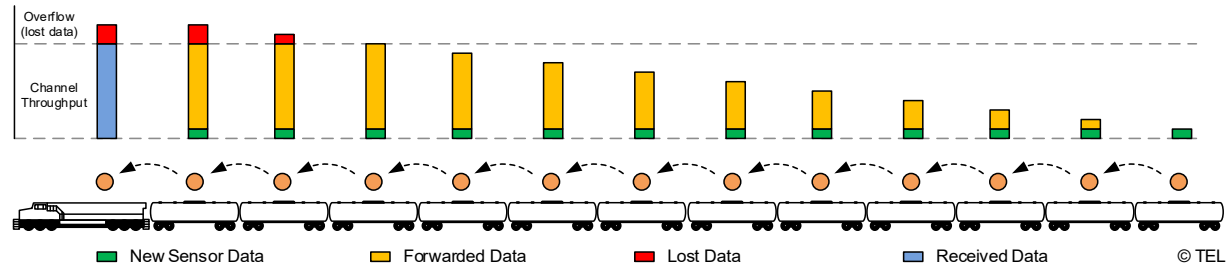


Figure 2 Potential Data Loss with Channel Throughput Limitations

TEL developed a multi-technology approach to freight train wireless sensor networks that it calls a hybrid technology network (HTN) [1]. HTN combines the benefit of low-power, short-range communication of ZigBee with high-performance, long-distance communication with WiFi between multiple sensor clusters. ZigBee is based on IEEE 802.15.4, with maximum data rates of 250 kbps and 16 channels at 2.4–2.4835 GHz and communication range of 100 ft. WiFi is based on IEEE 802.11, with a possible range of 100 m to 1 km. WiFi offers better range and higher data rates – from 1 Mbps to over 1 Gbps – than ZigBee. The combination of the two protocols offers low-power communications for most nodes, or motes, in the network with ZigBee and higher range and performance to communicate with the network gateway at the locomotive with WiFi.

The HTN is divided into clusters. Each cluster is an independent ZigBee network that is conveniently organized as an individual railcar. This maintains a small network within a railcar that ensures no data loss from network saturation. The WiFi throughput is significantly greater at around 6 Mbps for IEEE 802.11b. The goal was for each mote to contain both ZigBee and WiFi in order to maximize each node's flexibility to take on any role within the HTN network, as well as for redundancy.

Fifty-two commercial wireless network platforms were reviewed for application to WSN on freight trains. The selection criteria included the ability for both ZigBee and WiFi communications and the availability of sensors and interfaces. The Waspmote system by Libelium was downselected as the most promising commercial off-the-shelf (COTS) solution. Waspmote was the only one with the option for two radios operating simultaneously, as needed for the HTN.

Waspmote was tested at TEL using a wireless channel emulator from Azimuth Systems (ACE-400WB). The emulator allows full test repeatability and control over the RF environment using ITU-defined channel models. The models tested include Butler, with line-of-sight path loss only; Pedestrian A, that has some multipath and is equivalent to open area; Pedestrian B, that has significant multipath and is a city equivalent; Vehicular A, with velocity and higher path delay for multipath; and Vehicular B, with very large delays and high reflected signal strength. Path loss in the testing was determined in dB and converted to distance using $L = 10n \log d$, where d is the range, L is the path loss in dB, and n is the exponent. The exponent n was taken as 2.8 to reflect a moderate city environment. The exponent would be 2 in free space.

Two packet sizes were emulated with both ZigBee and ZigBee Pro. ZigBee Pro had significantly better range and was selected. Tests were at 2.42 GHz. Researchers discovered that WiFi on the Waspmote was unstable and that ad hoc mode is not recommended by the manufacturer. Additionally, they found that UART is used to control WiFi and limits speed to 115 kbps. Therefore, they determined that Waspmote was not suitable for HTN. The program continued with a custom-designed mote.

Table 1 Waspmote Data Loss Range

	Zigbee		Zigbee-Pro	
Packet Size	10 Byte	74 Byte	10 Byte	74 Byte
Butler Model	100 m	40 m		
Pedestrian A	20 m	30 m	100 m	95 m
Pedestrian B	60 m	85 m	95 m	95 m
Vehicle A	60 m	70 m	130 m	125 m
Vehicle B	15 m		110 m	110 m

The WSN was configured as a ZigBee Pro ad hoc network on each railcar with a dedicated personal area network coordinator (PAN-C) that had both ZigBee and WiFi radios. The PAN-C coordinates the railcar ZigBee network and becomes a node in the ad hoc WiFi network that communicates with the train gateway. The concept of a WTN has evolved as more descriptive of the capability of the network in device actuation as well as in deploying sensors for monitoring. The ad hoc WiFi network communicates with the WTN gateway, which is the interface to the train communication system to the back office. In the case of the illustration in Figure 3, the WTN gateway (GW) communicates via Ethernet to the Wi-Tronix WiPU locomotive back office interface. A feature of the WTN is that the network forms automatically once the consist is formed and ready to move. The End-of-Train device activation is the mechanism that triggers network formation.

The railcar ZigBee network is illustrated in Figure 4. Railcar motes only utilize ZigBee radios to save power and minimize form factor. Motes are designed to contain sensors within them, or to interface with external sensors or actuators. Sensors can be either self-powered or powered from the motes. Actuators are assumed to be self-powered.

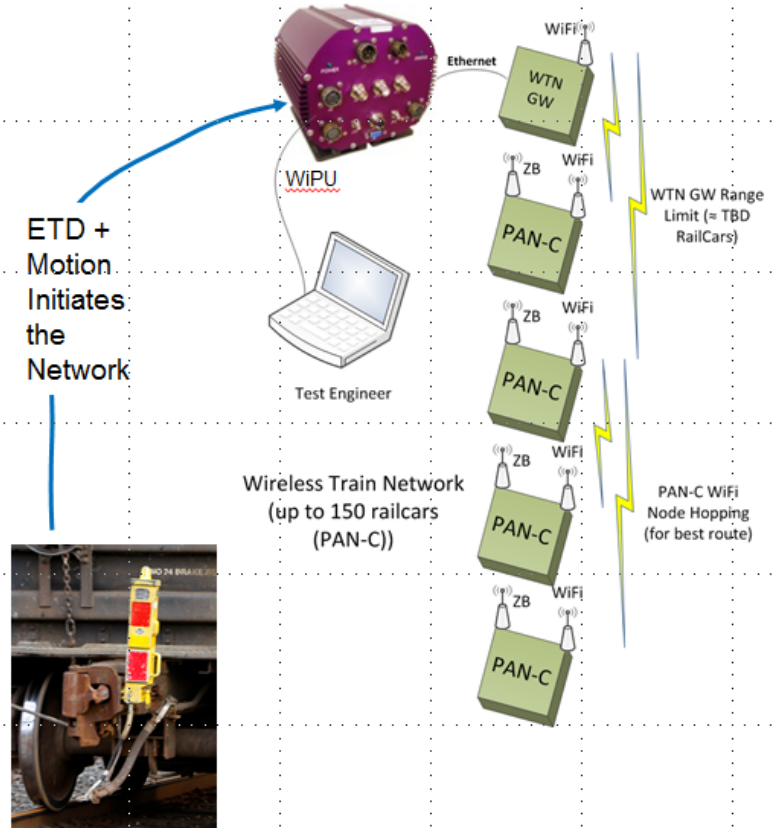


Figure 3 Wireless Train Network Schematic

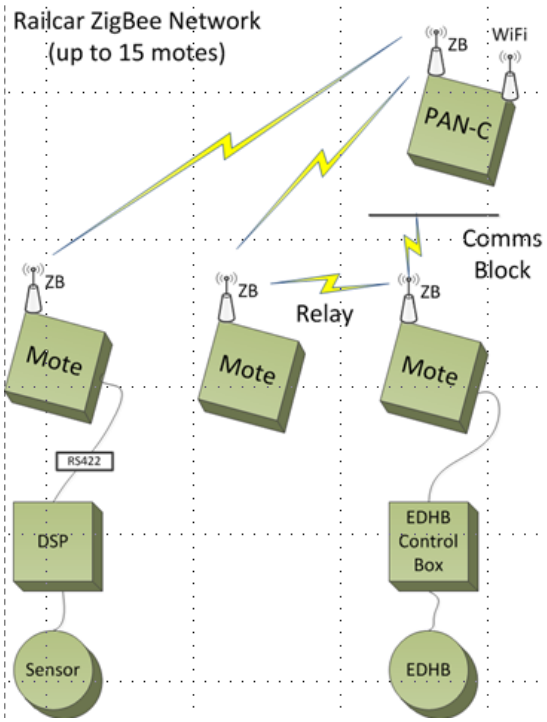


Figure 4 Railcar Zigbee Network

3 System Architecture

3.1 General System Layout

One of the core principles within an HTN is the multi-technology communication paradigm. This allows the network to be organized around groups of nodes called clusters. Each node within a cluster utilizes the low-power radio technology to communicate with the other nodes in the same cluster. Adopting terminology from ZigBee, one of the most popular low-power radio technologies for such applications, each cluster is its own PAN. Each PAN is created and maintained by a node with a special role, called the PAN coordinator. Researchers adopted the same principle and call the node that operates as this role the "PAN-C." All other nodes within the PAN are simply called motes. Each PAN has a single PAN coordinator, but can have multiple motes. The PAN-C, in this study's HTN architecture, is the gateway from the intra-cluster low-power network to the inter-cluster high-performance long-range network. This long-range network interconnects all PAN-Cs within a train consist and connects them to the third node role, the WTN. The WTN node interfaces the entire consist wireless network to the wired network infrastructure of the active locomotive.

To summarize, the nodes that are simply "motes" use only a single-radio architecture, with only the low-power radio present on each node. The PAN-C nodes utilize two radios and act as a gateway between the low-power radio domain and the long-range radio domain. The WTN node type, finally, utilizes two network technologies and acts as a gateway between the long-range radio domain and the wired Ethernet domain on board the locomotive.

This architecture enables a highly structured yet very flexible system operation. It combines in an optimal way the benefits of multiple, different networking technologies. In the current project, researchers chose the following technologies:

- Low-Power Radio: ZigBee, based on the ZigBee Alliance specifications and built upon an IEEE 802.15.4 radio.
- Long-Range Radio: WiFi, utilizing the IEEE 802.11b/g/n specifications in a compatible radio module.
- Wired Networking: Ethernet, using a 10/100 Mbps 100-BASE-TX interface built upon IEEE 802.3 specifications.

Overall message flow in the resulting architecture is shown in the Figure 5.

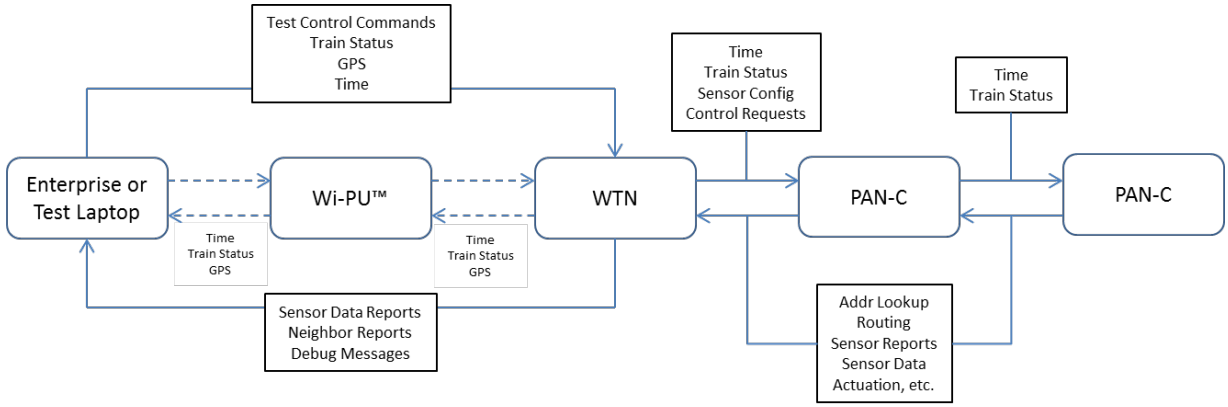


Figure 5 Wireless Train Network Message Flow

3.2 Overall Mote Architecture

Within each mote the design will utilize a layered architecture, following design principles derived, but not strictly adhering to, the Open Systems Interconnection (OSI) layer stack. In most typical network applications, the OSI 7-layer stack is adopted to particular constraints and requirements. For sensor network applications, this results in a rather varied number of layers being employed. All architectures, however, will have physical layer(s) at the bottom and application layer(s) at the top. This is true for the HTN mote architecture as well. The overall structure is shown in Figure 6.

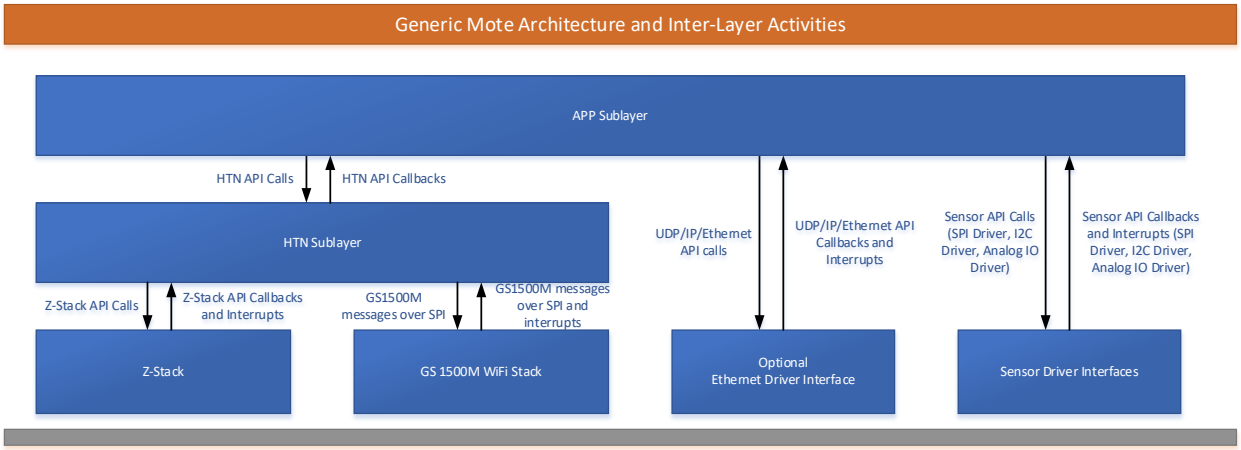


Figure 6 Mote Architecture

HTN utilizes two different radios, and provides functionalities that tie both together into a single system design. Therefore, in this project's architecture for HTN, there were two individual protocol stacks present and integrated: a ZigBee protocol stack called Z-Stack, provided by Texas Instruments that is ZigBee compliant, as well as a network stack built on top of WiFi, utilizing the IEEE 802.11b/g/n protocol family as well as providing IP and TCP/UDP functionality. In the above figures, both protocol stacks are shown as single blocks. In reality, they themselves consist of multiple protocol layers. A ZigBee stack architecture is shown in Figure 7.

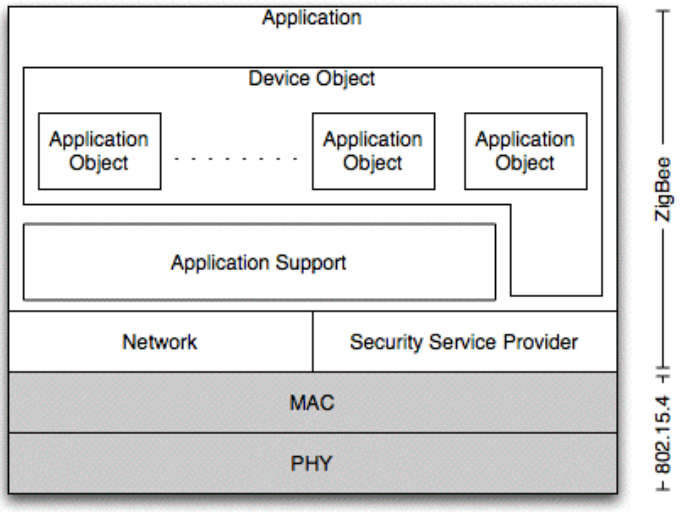


Figure 7 Zigbee Stack Architecture

The physical layer and medium access control (MAC) layer are defined by IEEE 802.15.4. The ZigBee Alliance then defined the networking and application support (APS) layers above that. From the perspective of ZigBee, it would then interface or integrate with an *application layer* above.

Similarly, the WiFi stack consists of the 802.11 PHY and MAC layers, where the MAC layer is part of the link layer definition. Above that is the IP network layer, followed by the TCP/UDP session layer, as shown below:

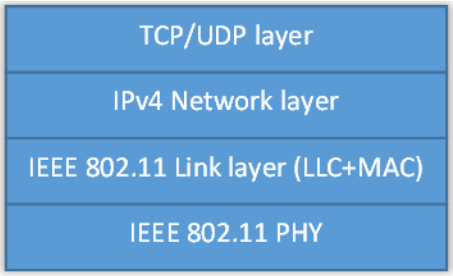


Figure 8 WiFi Stack Layers

Once again, from the perspective of the WiFi stack, it would then interface to the application layer above it.

In the current HTN design, the application layer is a split-architecture. The bottom portion is the HTN sublayer. It interfaces downward to both the ZB stack as well as the WiFi stack. HTN manages both radios' functionalities. In addition, it also implements a routing sublayer for the WiFi stack, as it by itself is only capable of single-hop communication. The *application sublayer* is then implemented above the HTN sublayer. The APP sublayer interfaces to HTN and other system devices, such as sensors, power management, other network interfaces, etc.

As seen from the HTN stack architecture, there are numerous intra-mote interfaces being utilized. Within a protocol stack these are often referred to a *service access points* (SAP). These exist at the boundary between two layers. There are also API-provided interfaces to hardware

components, such as the sensors, other protocol stacks, and more. This report will refer to these in later sections when discussing the messages being exchanged over these interfaces.

3.3 Inter-Mote Interfaces

The mote architecture shown in the section above is utilized to achieve network communication between motes as well as other devices on the HTN network. These interfaces are shown in the following figure:

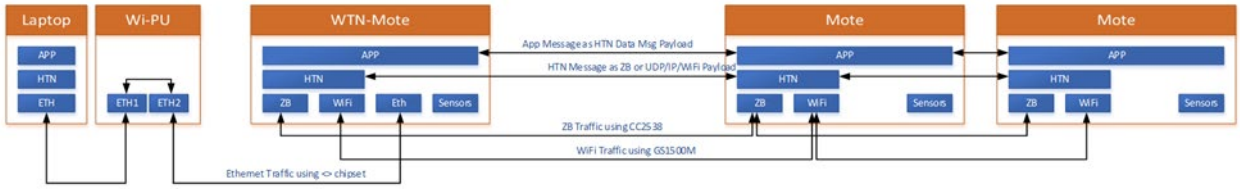


Figure 9 Wireless Train Network Communication Interfaces

From left to right, the figure shows a test conductor laptop, connected to an on-locomotive Wi-PU™, which in turn is connected to the WTN gateway mote. The interfaces between these three devices are all based on a standard Ethernet network connection running a full IPv4 stack. The Wi-PU will act as a switch in this network, enabling the laptop to interface with the WTN gateway. The WTN gateway is a typical HTN mote, except for the addition of a third network interface – an Ethernet port. It acts as the gateway between the HTN network and the outside world. It operates using firmware specialized for this purpose.

The HTN network exchanges messages at various levels. For example, an application layer message is carried over this network within the payload portion of the HTN protocol's data message. These HTN data messages in turn are transported either over the ZB network or over the WiFi network.

3.4 Mote Activity State Machine

In addition to the overall mote architecture, it is also important to understand, from a high-level perspective, the mote's operations. These are illustrated in the following two *finite state machines* (FSM). Akin to a flowchart, they illustrate activity states and transitions between them. The first FSM illustrates the HTN layer functionality:

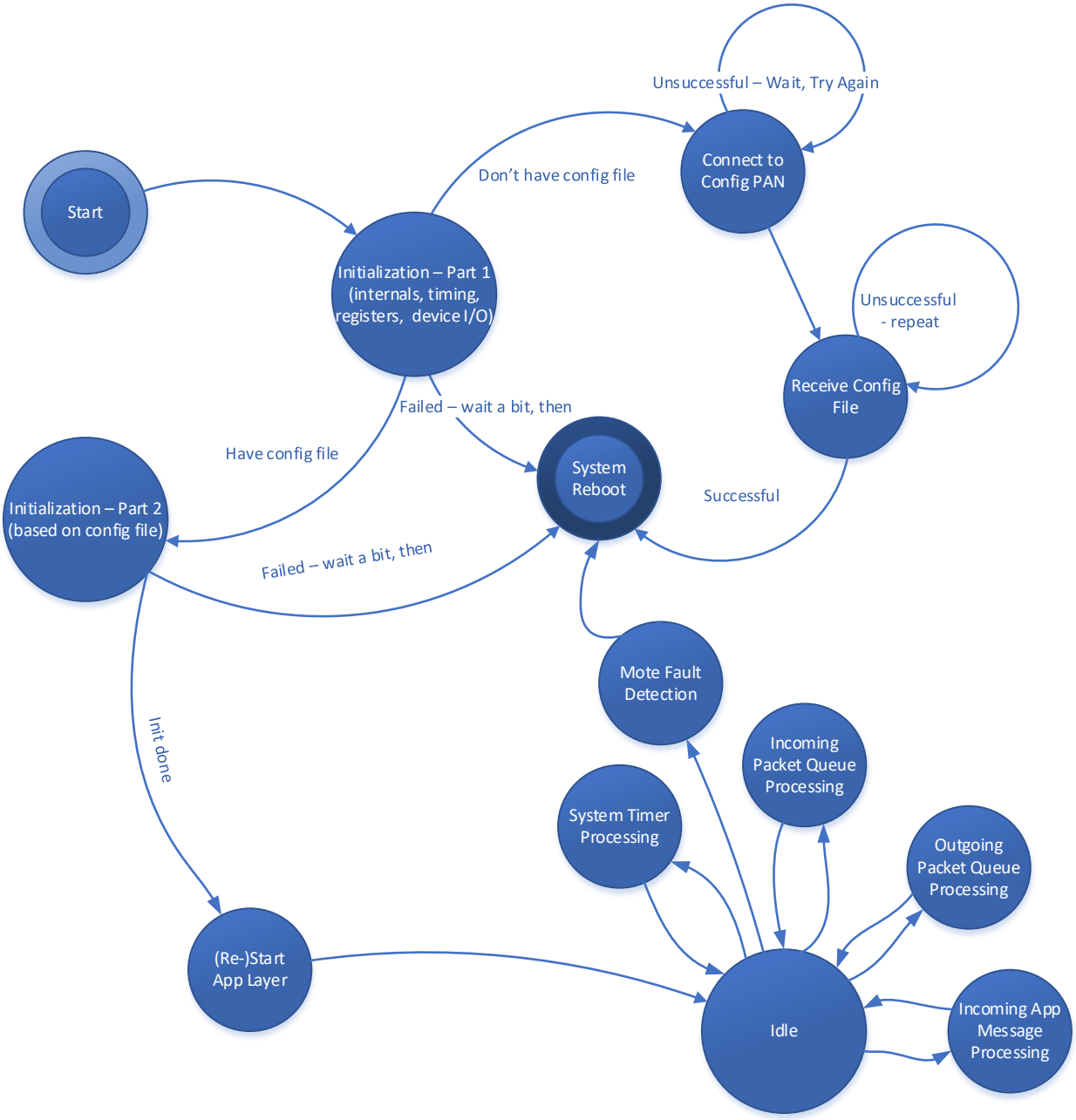


Figure 10 Finite State Machine for HTN Functionality

It shows that after startup, the HTN layer will attempt to validate the presence of configuration information. If not, it will transition into a state where it can acquire this information through a designated ZB PAN from the outside world. Once provisioned, it will reboot and execute initialization again, but now including the present configuration information. Once initialization is completed it will trigger the start of the application layer within the HTN motes. The startup of the application layer also transitions the HTN layer into idle mode. From within idle mode a series of different events will result in a variety of different state transitions. For the sake of

brevity, project researchers limited the level of detail shown to just these activities, not their internal activity details.

Similarly, a state machine is defined for the application layer. Its design is very similar to the HTN layer architecture. Once initialization is completed, the mote's app layer will transition to idle and execute different states based on triggers. The app layer state machine is shown in the following figure:

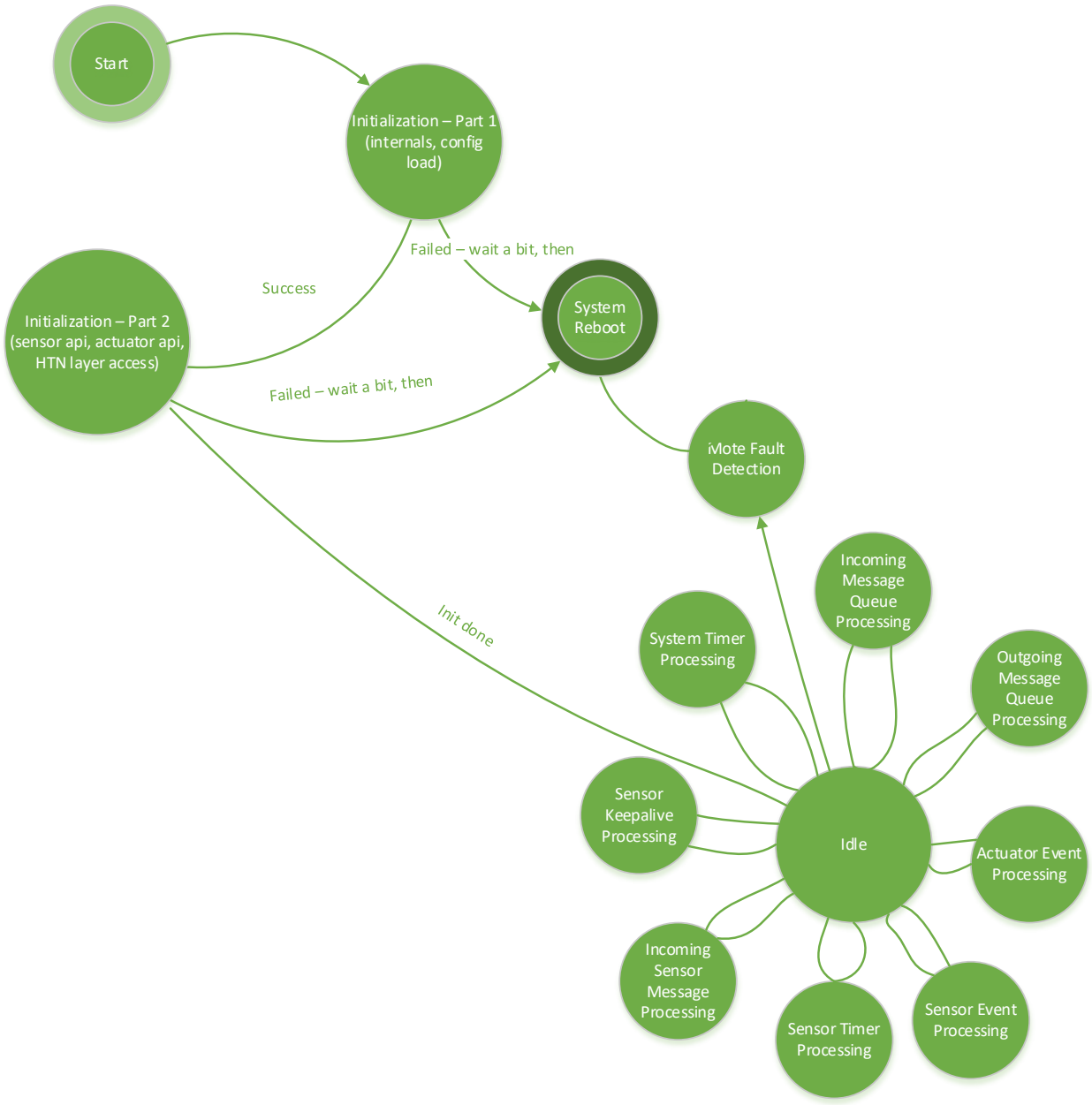


Figure 11 Finite State Machine for Application Layer

3.5 Role of Witronix Wi-PU

The Wi-PU (Figure 12) is a ruggedized computer that remotely sends asset data in real time. The Wi-Tronix Wi-PU is an integral aspect of the overall design of this system. It is the project network's interface to the locomotive and the train state. The Wi-PU is expected to provide the following functionalities to the WTN:

Figure 12 Wi-Tronix Wi-PU

1. Provide time information for the train network to set the real-time clock. This process is initiated always by the WTN, as a query directed from it to the Wi-PU. The Wi-PU then sends its current time back in a corresponding message. It is never sent as an unsolicited update by the Wi-PU.
2. Provide GPS location information. This process is initiated always by the WTN, as a query directed from it to the Wi-PU. The Wi-PU then sends its current location, speed, etc., back in a corresponding message. It is never sent as an unsolicited update by the Wi-PU.
3. Provide train formation status information. Primarily, this indicates that the End-of-Train device has been attached and is in communication with the Head-of-Train device. This indicates that consist formation is completed and may be used as a train network start indicator. This also indicates when the train's ETD is removed or missing. This update is sent as an unsolicited update by the Wi-PU to the WTN. It may also be requested by the WTN from the Wi-PU. It may be acknowledged by the WTN back to the Wi-PU, using regular HTN ACK.
4. Provide train movement information. Primarily, this indicates a change in status between moving to stationary. This is used to regulate sensor activity levels. This update is sent as an unsolicited update by the Wi-PU to the WTN. It may also be requested by the WTN from the Wi-PU. It may be acknowledged by the WTN back to the Wi-PU.
5. Provide requests to acquire sensor information or to actuate mote-attached devices such as the EDHB. Requests are sent by the Wi-PU unsolicited to the WTN, which directs them to the appropriate mote or motes. These requests may be acknowledged by the motes. The motes then send requested information (in case of sensor readings) or status

indications (in case of actuation) back to the Wi-PU. These messages may be acknowledged by the Wi-PU back to the originating mote.

During the TTCI testing, the Wi-PU may be abstracted through the use of a test laptop, which will fulfill the same requirements and handle the same message flow as the Wi-PU. This reduces the time required until testing at TTCI can commence. It does in no way diminish the role of the Wi-PU in the overall design of this system.

3.6 Role of the Enterprise

Another key element to this system is the *enterprise*. The enterprise represents any outside entity interfacing with the on-train network. This can be a dispatch center, a test laptop, or other similar infrastructure elements.

It interfaces to the train network through the Wi-PU. This enterprise may connect to it via a wired interface, such as in the case of a test laptop, or via a wireless interface, be that cellular, satellite, trackside, or other technologies.

The purpose of the enterprise is:

- To receive and log updates about critical state changes of a train consist including type, location and time.
- To be able to remotely retrieve or request information from the train network.
- To receive inventory information.
- And to receive and log notifications about discovered dark cars, including time, location, and railcar identification.

In addition, when acting as a test laptop, it also provides a visual interface for debugging and logging of further operational information from the train network. This may include information such as network topology, event logs, battery status, etc. This may then also be used to request the actuation of EDHBs and other devices attached via the train network, essentially becoming a replacement of a visual interface for the train engineer to operate actuators and sensors.

A final role limited to only the test laptop is to assume responsibilities of the Wi-PU itself. This is a special case and not considered an element of the final product. It then fulfills all roles otherwise provided by the Wi-PU with respect to the interface to the WTN.

3.7 Addressing

This is an important aspect to clarify, especially since it involves a variety of different network types, all utilizing different addresses.

3.7.1 IP Address Space

IP addresses are assigned to all the WiFi radios within the HTN network for each consist. This includes the radio on the WTN as well as all the PAN-Cs. All WiFi IP addresses internal to the HTN are randomly assigned from within the 10.x.x.x Class-A IP address space. Each train consist uses the entire 10.x.x.x Class-A as its internal address space. The WTN, through its Ethernet port, acts as the network address translator to the outside network connections. The WTN Ethernet port IP address assignment is outside of the scope of this document.

The addresses for the WiFi radios inside the HTN network can be chosen in several different ways. The preferred method is to use the lower 3 bytes of the WiFi radio's MAC address as the 3 bytes to complete the 10.x.x.x address. There are only three reserved addresses:

10.0.0.0 – This is an IP specification reserved address and signifies the network itself.

10.255.255.255 – This is an IP specification reserved address and signifies the broadcast address for this network.

10.0.0.1 – This is the first IP address of the 10.x.x.x block and shall be used as a special address permanently reserved for the WTN device. This does not replace the WTN's actual WiFi IP address, but rather acts as a placeholder in case an HTN PAN-C does not know the actual IP address of the WTN, but needs to communicate with it. This always signifies that routing through the WiFi network shall follow the path through the parents, which ultimately always leads to the WTN.

3.7.2 ZigBee Address Space

We use standard ZB addressing, with the possibility of using short addresses or extended addresses.

3.7.3 Application Layer and HTN Layer Address Format

At the application and HTN layer, devices are addressed using a combination of RailcarID, SensorID, and the SensorPort.

RailcarID follows the industry standard notation of reporting mark and number. It is thus comprised of 1 to 5 letters, followed by 1 to 6 digits. An example is "SHPX 462477." For this project's addresses, reporting marks are always allocated 5 character slots and numbers are always allocated 6 digit slots. Slots that are not used are filled with a space (" ") character. All reporting marks with less than 5 characters are always left aligned. All numbers are always left aligned. Hence spaces used as fillers are always trailing spaces, never leading spaces. For example: "SHPX ", but never " SHPX".

The SensorID identifies the type and location of the actual mote on the railcar. It uses a two-letter identifier of the overall position type (wheels, door, coupler, EDHB, PAN-C, WTN, etc.). It adds to that a 1-to-3-letter designator of the specific position (when wheel, examples are L1, R3, etc.; when door it can be L, R1, etc.; when EDHB it indicates position of EDHB based on the end A", "B", etc. that it is mounted on, especially on segmented railcars; for PAN-C it designates the location of the PAN-C, such as "TOP", for WTN it is "CAB"). Finally, it adds to that a four-position alphanumeric Inventory ID that is the hex representation of the lower two bytes of the device's hardware address. For motes, this hardware address is derived from the ZigBee radio's hardware address. For PAN-Cs and WTNs, this is derived from the WiFi radio's hardware address. For the type and position field the same rule from the reporting mark about the use of space as a fill character in the event that fewer characters are used than there are slots allocated applies. For example, the position L1 would be represented as "L1 ", but never as " L1" or "L 1". Similarly, a position label of T would always be represented as "T ", never " T " or " T".

Each mote may have the option to control one or more attached devices, such as sensors and EDHBs. For addressing a specific attached device for specific actions, such as engaging a handbrake or reporting on a specific sensor, the attached device is referenced via a sensor's Port

identifier that is provided as part of the address. This is a one-digit numeric designator. The first attached device is on port 1, the second on port 2, and so on.

The complete address format for application/HTN layer addresses thus has the following format as shown in Figure 13.

Reporting Mark	Reporting Number	Type	Position	Identifier	Port
5 Letters	6 Digits	2 Letters	3 Letters/ Digits	4 Letters/ Digits	1 Digit

Figure 13 Application/HTN Layer Address Format

As a result, the entire application/HTN layer address occupies 21 alphanumeric characters. In this designation, letters are case-insensitive. No special characters are allowed, except for a space (“ ”). Thus, the entire set of allowed characters is 26 letters, 1 special character and 10 digits. These 37 characters can be represented by 6 bits, allowing a compressed representation of the above 21 characters in 126 bits. Researchers added 2 bits to achieve a full 128 bits, or 16 bytes. The extra 2 bits are reserved, are always the leading 2 bits so that the bit sequence itself is right-justified, and are always set to 0 in the current specification. The sequence for transmission of these 16 bytes uses big endianness. That is, the most significant byte is transmitted first, and is the byte that includes the 2 fill bits as the 2 most significant bits.

Individual characters are represented using the following encoding in Table 2.

Table 2 Character Code

Character	Decimal Value	Character	Decimal Value	Character	Decimal Value
0	0	D	13	Q	26
1	1	E	14	R	27
2	2	F	15	S	28
3	3	G	16	T	29
4	4	H	17	U	30
5	5	I	18	V	31
6	6	J	19	W	32
7	7	K	20	X	33
8	8	L	21	Y	34
9	9	M	22	Z	35
A	10	N	23	<space>	36
B	11	O	24		
C	12	P	25		

3.7.4 Special Addresses

Due to the use of only 37 characters per position, but the possibility of the 6 bits to encode 64 different values, there is a lot of room for special addressing.

3.7.4.1 Special Addresses for RailcarID

For the RailcardID, researchers specified that a reserved address for the lead (controlling) locomotive. This is used when a mote is unaware of the actual address because it hasn't yet learned it from network communication, but still needs to send something to the locomotive. To do so, the reporting mark is represented by all-1 bits, but the reporting number is represented by all-0 bits. Hence, the 66 bits comprising this address portion are:

1111 1111 1111 1111 1111 1111 1111 11 : 0000 0000 0000 0000 0000 0000 0000 0000 0000

If a broadcast is needed to address all railcars, the RailcarID uses all-1 bits for both portions, e.g.,:

1111 1111 1111 1111 1111 1111 1111 11 : 1111 1111 1111 1111 1111 1111 1111 1111 1111

This is only allowed in messages from the test laptop, Wi-PU, or WTN to the HTN network, but never by specific PAN-Cs or motes. Responses back would always use a specific source and destination address.

3.7.4.2 Special Addresses for SensorID

Similarly, the SensorID provides flexibility in addressing. For example, all sensors on a railcar can be addressed by setting the type, position, and identifier fields to all-1 bits, such that the SensorID would look as follows, denoted as bit values:

1111 1111 1111 : 1111 1111 1111 1111 11 : 1111 1111 1111 1111 1111 1111

Special addresses also are possible, and are identified by the type being all-1 bits and the identifier being all-0 bits. The defined special addresses are:

WTN:	1111 1111 1111 : 0000 0000 0000 0000 00 : 1111 1111 1111 1111 1111 1111
PAN-C:	1111 1111 1111 : 0000 0000 0000 0000 01 : 1111 1111 1111 1111 1111 1111
Wi-PU:	1111 1111 1111 : 0000 0000 0000 0000 10 : 1111 1111 1111 1111 1111 1111
Test Laptop:	1111 1111 1111 : 0000 0000 0000 0000 11 : 1111 1111 1111 1111 1111 1111

If all sensors of a specific type are to be addressed, for example all EDHBs, then the specific type is encoded and the position and identifier fields are set to all-1 bits, such as:

<EDHB> : 1111 1111 1111 1111 11 : 1111 1111 1111 1111 1111 1111

Also, if a specific sensor and location is sought to be addressed, but the exact SensorID is not known, its type and position can be encoded, but the identifier field can be set to all-1 bits. This will address all corresponding sensors of that type, in that specified location, if there are multiple instances. However, this will typically not be the case. The L1 wheel sensor, for example, could then be addressed as shown below. Please note that the position is shown as it would be used in

the message. Any unused characters are always set to the <space> character, trailing in the designator:

<Wheel> : “L1 “ : 1111 1111 1111 1111 1111 1111

If no sensor port is needed as part of the address, the sensor port is set to the value 0. If all sensor ports are to be addressed, the port field is set to all-1 bits. Otherwise, valid sensor port values are 1 through 9.

3.8 Message Handling

It is the HTN layer’s role to provide the network connectivity between all devices. It utilizes both WiFi and ZigBee to accomplish efficient and reliable communication. However, this mandates that the HTN layer can successfully utilize the various different address formats, and translate between them as needed. This process is illustrated using an example as follows:

When a WTN receives a message from the Wi-PU or test laptop to be delivered to a specified device, based on the provided application/HTN layer address using the format above, it needs to determine what railcar this sensor is on in order to send it to the correct PAN-C of that specified railcar. It uses a look-up table that contains entries of the following pair information:

(Reporting Mark, Reporting Number) : (PAN-C WiFi IP address)

This allows the WTN to determine the IP, and then send the application/HTN layer message encapsulated in an IP/UDP message over WiFi to the PAN-C using the IP address information from the look-up table. The system provides mechanisms to populate this table at the WTN.

When the PAN-C receives the message via WiFi, it first extracts the application/HTN layer message from the payload and determines the SensorID from the application/HTN layer destination address. It uses this to once again consult a look-up table. This table provides information about the ZigBee network address of the specific mote that is being addressed. The entries in the look-up table represent the following information:

(Sensor Type, Sensor Position, Sensor Identifier) : (ZigBee PAN Extended/Short Address)

In ZigBee, addresses are always addressable using their *extended address* (EAddr). This is a 64-bit address globally unique to the device and is the ZB radio’s hardware address. Optionally, the PAN coordinator can also assign 16-bit *short addresses* (SAddr) to motes within its PAN. These are not globally unique and only valid for the duration of the PAN. This greatly reduces the overhead and, if available, will be used within each railcar’s PAN functionality. Their scope, through the look-up at the PAN-C, is limited to only the railcar and thus no address conflicts are created by their use.

The PAN-C then encapsulates the application/HTN layer message in the ZB message payload section and sends it to the correct mote using the address from the look-up table.

When the mote receives the message via ZB it extracts the application/HTN layer message from the payload and determines the functionality requested by this message.

3.9 Interface Mitigation

To limit interference, researchers employed frequency separation: In the firmware, the channels in which the low-power radio operates and in which the high-bandwidth radio operates are kept mutually exclusive in the frequency domain. In Figure 14 below, the IEEE 802.11 channels are 22 MHz wide and spaced 5 MHz apart in the frequency domain. There are a total of 14 overlapping IEEE 802.11 channels. The IEEE 802.15.4 channels are 2 MHz wide and spaced apart by 5MHz in the same 2.4 GHz frequency spectrum. In the implementation channel, only the three non-overlapping 802.11 channels 1, 6, and 11, are used. Interference between 802.11 and 802.15.4 is then controlled by utilizing only any of the channels 15–24 that fall outside of the frequency spectrum occupied by selected IEEE 802.11 channel.

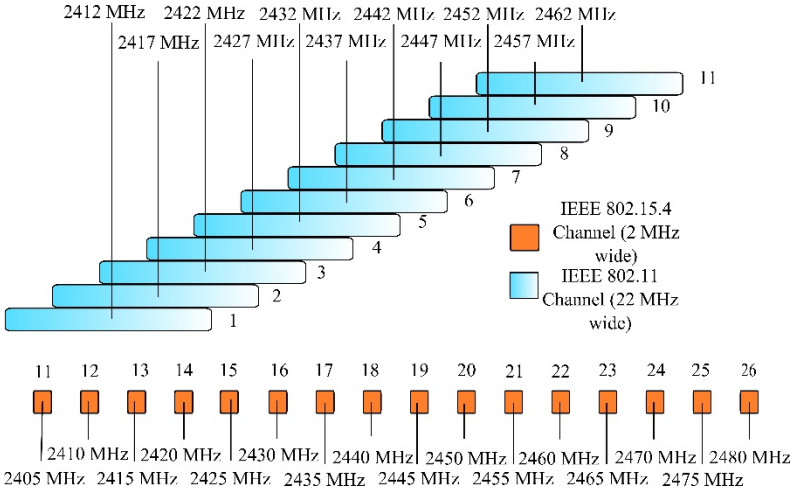


Figure 14 IEEE 802.11 and 802.15.4 Channel Frequencies

3.10 Material Interface

The railcars on which the motes are deployed are made of various ferrous and non-ferrous materials and hence unevenly absorb electromagnetic radiation. This severely limits the possible range of communication between the motes. To provide connectivity between motes and ensure alternate pathways of communication in the event that a mote fails, a placement strategy such as the one shown in the figure below was followed. The lines drawn in the figure highlight some of the possible routes packets may take from a source mote to the destination mote without requiring transmission through the railcar to motes on the other side.

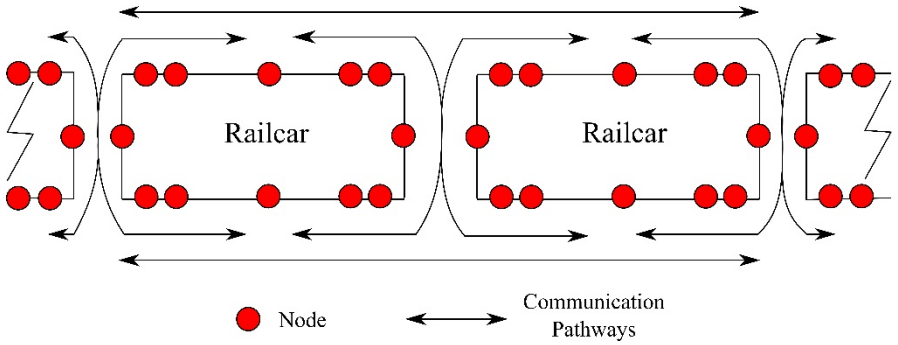


Figure 15 Mote Communication Pathways

4 Wireless Train Network System Hardware

The program has chosen to operate in the open access 2.4 GHz ISM frequency bands. This is consistent with a low-power radio system and is consistent with frequency interference mitigation techniques.

The system uses primarily alert based reporting with the enterprise retaining the ability to request overall status and individual sensor data. The enterprise can also set alert thresholds for sensors. The system is not configured for data streaming such as for video feeds as that is not consistent with the low-power goal of the sensors and motes.

The system is required to be self-powered either by primary batteries with a 3- to 5-year life, or with rechargeable capabilities. Investigations show there are several techniques for energy harvesting that could be implemented, such as with wheel bearing generators feeding a railcar energy storage system. However, fully exploring this energy harvesting system was beyond the scope of the study. With implementation of electronically controlled pneumatic (ECP) braking, the powering of a wireless train network would be more readily achieved.

The program initially examined available network platforms to implement the WTN. The assumption from previous work was that the range and bandwidth was not sufficient with a ZigBee Pro network alone and that the system needed to accommodate both WiFi and ZigBee radios simultaneously. Of the available systems, only the Waspmote series had that capability. It was later discovered that Waspmote did not accommodate an ad hoc mesh network and so was abandoned. The project then aimed to develop a custom system. Of important consideration in the design process for these custom systems were the power requirements of the ZigBee and WiFi modules. The CC2538 ZigBee module by Texas Instruments (TI) and the GS1500M WiFi module from Gainspan were selected as a result. An Ethernet module was also added for connection to the Wi-Tronix WiPU system by the WTN gateway mode.

4.1 Microcontroller

Researchers chose the TI CC2538 as the microcontroller for the mote hardware. It is based on the Cortex-M3 ARM architecture and operates at 3.3 V supply. The frequency of operation is 32 MHz. It provides 512 KB of flash memory for firmware storage and 32 KB RAM. The microcontroller supports over-the-air firmware upgrade allowing remote update of firmware.

The microcontroller supports all standard peripheral interface technologies like Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I2C), Universal Synchronous Asynchronous Receiver Transmitter (USART) and Universal Serial Bus (USB).

The microcontroller provides secure communication using hardware acceleration for various encryption algorithms like AES128/256, SHA2, ECC 128/256, and RSA.

A 12-bit, 8 channel, analog-to-digital converter can interface to sensors and devices providing analog inputs.

The microcontroller also contains a built-in IEEE 802.15.4-compatible radio used for ZigBee communication in the project's implementation. The radio can operate at a maximum of 250 kbps data rate at a maximum output power of 7 dBm and a maximum receiver sensitivity of -97 dBm. The radio hardware consumes 20 mA of current for receive and 24 mA of current to transmit at 0 dBm and 3.3 V supply.

Several power-saving modes are provided, allowing varying degrees of memory retention while reducing power consumption. The lowest current consumption of the microcontroller is 1.3 uA in sleep mode. These parameters make the CC2538 an ideal choice – lending a balance between capabilities, performance, and power saving.

4.2 IEEE 802-11 WiFi Module

The WiFi communication capability is added to the hardware using a Gainspan GS1500M module. The use of a module reduces the board space required and also optimizes power consumption.

The capabilities of this module are summarized in Table 3 below.

Table 3 GS1500M Gainspan WiFi Module Specifications

Radio Protocol	Supports IEEE 802.11b, 11g, 11n single stream
RF Output Power (Typical)	14 dBm (802.11b), 12dBm (802.11n)
RF Operating Frequency	2.412 - 2.484 GHz
Supported Data Rates	802.11b (CCK): 1, 2, 5.5, 11 Mbps 802.11g (OFDM): 6, 9, 12, 18, 24, 36, 48, 54 Mbps 802.11n (1x1 HT20): MCS0-7
Antenna Options	PCB Trace or uFL connector for external antenna
Operating Temperature	Industrial (-40° to +85°C)
Security Protocols	WEP, 802.11i WPA/WPA2 Personal Security (AES and TKIP), Enterprise Security (EAP-FAST, EAP-TLS, EAP-TTLS, PEAP)
Networking Protocols	UDP, TCP/IP (IPv4), DHCP, ARP, DNS, SSL, HTTP/HTTPS Client and Server
Certifications and Compliance	FCC, IC, ETSI, RoHS, Wi-Fi Alliance
I/O Interface	UART (2), SPI (2), I2C, ADC (2), WAKE, ALARM (2), GPIOs, PWM, JTAG
Dimensions	1.45 in x 0.9 in
Power Source	3.3V

The module is used at 7 dBm output power. The IEEE 802.11b is used at a data rate of 2 Mbps. The microcontroller interface to the module is over SPI at a data transfer speed of 2 Mbps. The uFL connector is used for an external whip antenna allowing an additional gain of 5 dBm. Ad hoc. self-healing network formation and UDP transport protocol is used for communications.

4.3 Network Hardware Implementation

Network components are illustrated in Figure 16. The mote housing contains a CC2538 evaluation board from TI that contains the ZigBee system and the 2.4 GHz IEEE 802.15.4 communications. The printed circuit board (PCB) contains the application layers and the external interfaces. The mote connects to the EDHB through the PCB and GPIO interfaces. The wheel bearing sensor connects to the mote through an RS422 connection, as shown.

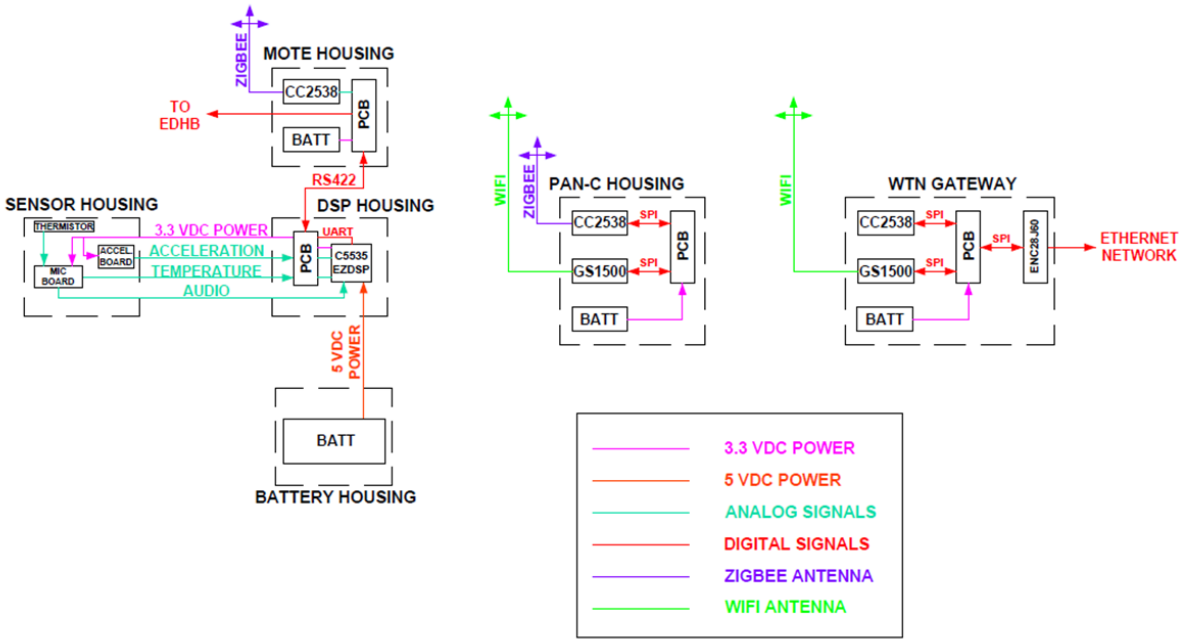


Figure 16 Wireless Train Network Components

The wheel bearing sensor consists of three housings: the sensor housing, mounted to the wheel bearing housing, and the digital signal processing (DSP) housing and battery housing, mounted within the truck frame. The sensor housing contains the three sensors: a thermistor used as the hot box detector, a microphone (MIC board) for the acoustic detector, and an accelerometer (ACCEL board) for detecting bearing vibrations.

The mote hardware is shown in Figure 17. The CC2538 ZigBee board sits atop the PCB motherboard. The CC2538 board mounts to the PCB through two 20-pin connectors. Researchers found, after delivery of the motes to UNL for network testing, that those 20-pin connectors were installed in a mirror image of what they should have been. This created the need to recall the motes and rework the boards. The external view of the completed motes with antennas is found in Figure 18.

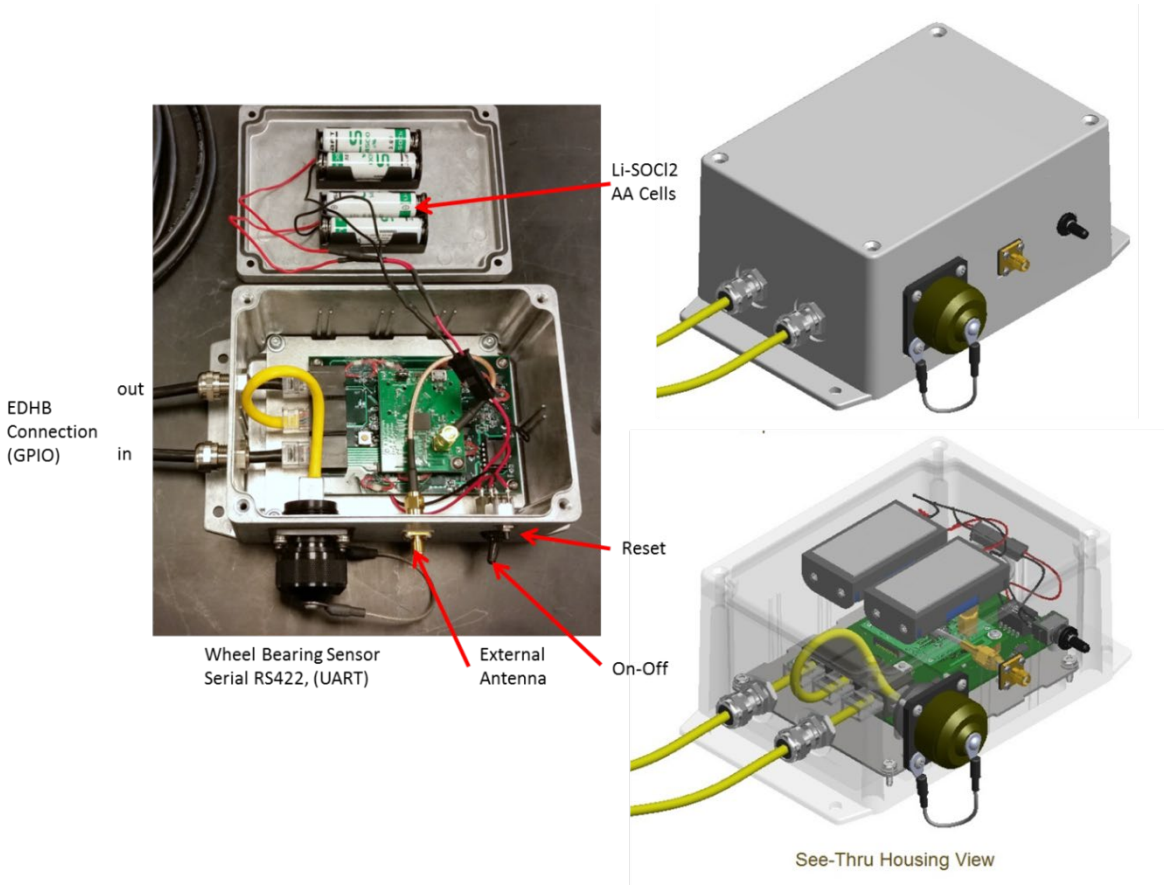


Figure 17 Sensor Motes Electronics

During analytical evaluation of the power usage in the motes, researchers determined there was a tradeoff required in the type of ZigBee network for the railcars. The initial concept was for an ad hoc ZigBee network that would allow motes to act as repeaters for other motes where communications to the PAN-C was lost. This is an ideal situation. However, to accomplish that, the mote receivers needed to be on at all times. The resulting power requirements made a practical, self-powered mote impractical. To avoid this situation, the ZigBee network was configured as a star network. Each mote must be able to be in communication with the PAN-C and cannot repeat signals. To enable this network in evaluation, each mote was equipped with an external antenna to ensure communications with the PAN-C. This may not be a requirement in a production system. The motes wake and are queried in set intervals. This query period is currently set at 10-minute intervals and the motes are in sleep mode in between this interval. All motes will wake and transmit alerts off-cycle when attached sensors indicate that condition. The motes are powered with four AA Li-SOCl2 cells. The anticipated power usage in a star network is estimated to provide a 3.5-year battery life. This is close to the desired 5-year battery life associated with normal railcar maintenance cycles. Note that the mote circuit boards have a coulomb counter implemented, so that actual power draw during testing will be measured as compared to the predicted values.

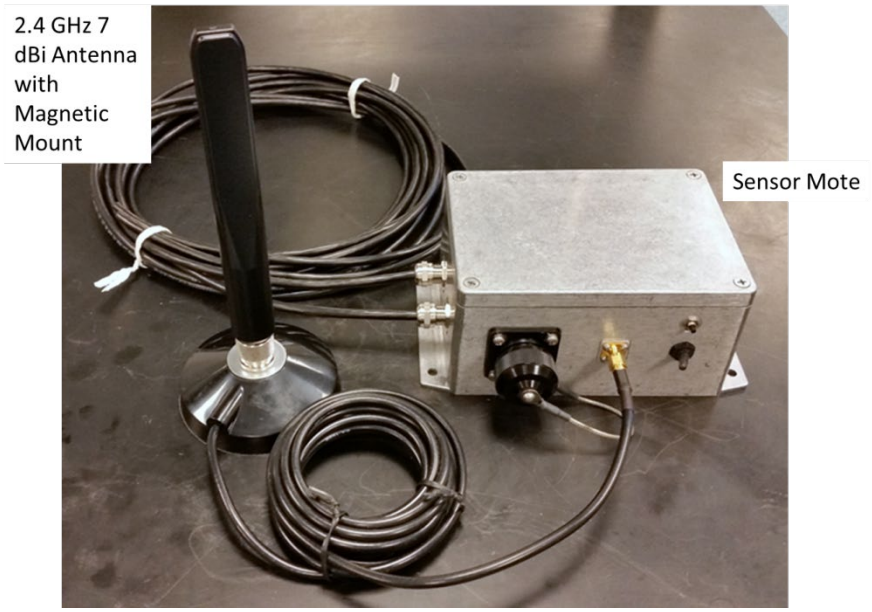


Figure 18 Sensor Motes with External Antenna

The PAN-C device is shown in Figure 19. The PAN-C controls both the ZigBee network and an ad hoc mode in the train WiFi network. The board shown in the figure also has an Ethernet port that is enabled when used as the WTN gateway. There are three boards in the PAN-C: the motherboard, the CC2538 for ZigBee control, and the GS 1500M for WiFi control. Both ZigBee and WiFi are in the 2.4 GHz ISM bands. ZigBee has 16 channels and WiFi has 3 channels of greater bandwidth. When the WiFi network is formed, the WiFi channel is selected and maintained. The ZigBee network channels are searched for the best reception at each occasion, which avoids interference with the WiFi channels.

In practice, the PAN-C should be installed on the railcar with the antennas positioned for best reception. Motes and sensors should be mounted on the railcar and configured to be specifically part of that particular railcar's network with its ID. This will avoid mixed communications between networks, both on the train and in the classification yard.

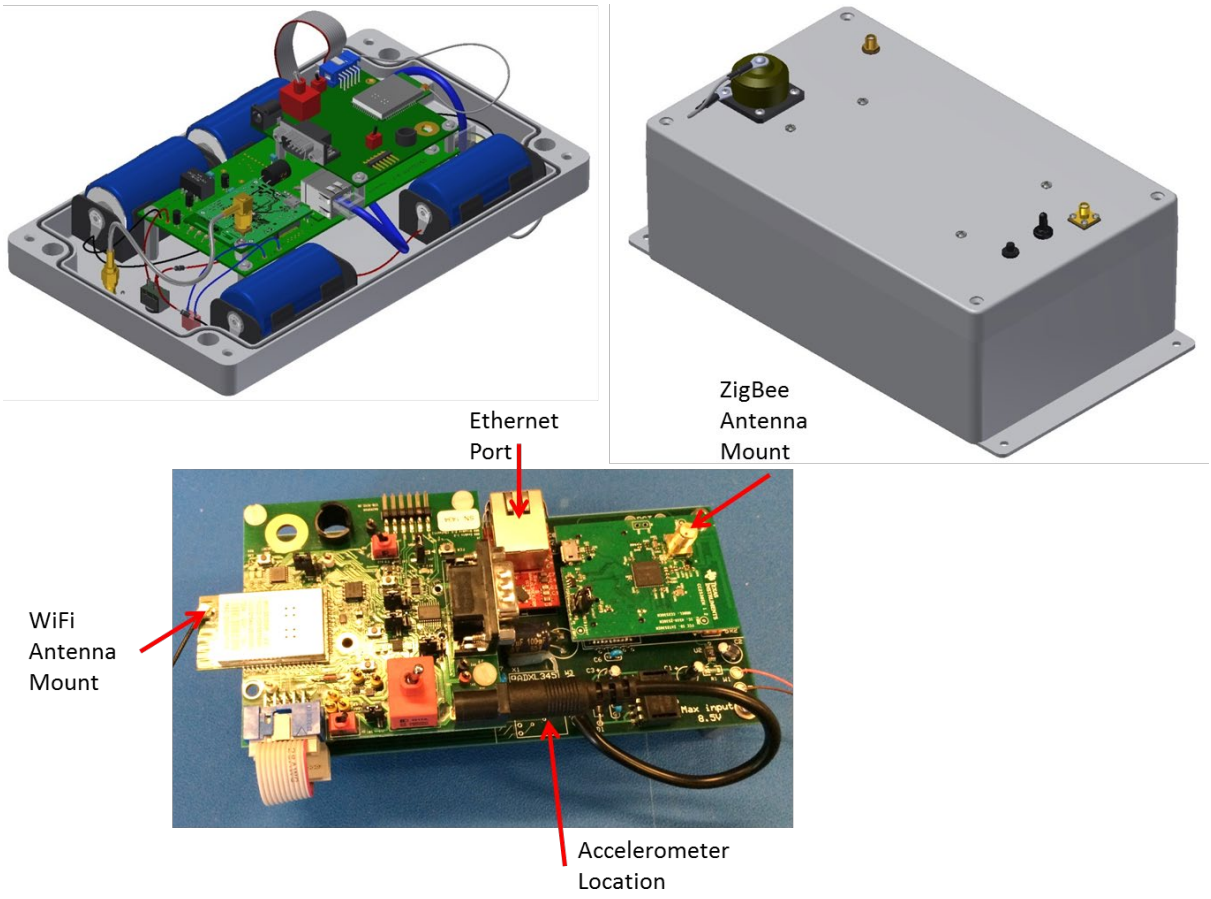


Figure 19 PAN-C and Electronics

4.4 Power Usage

4.4.1 Mote Power

The motes use the TI CC2538 wireless microcontroller for 2.4 GHz IEEE 802.15.4 and ZigBee applications. The board draws 2 µA in sleep mode and 13 mA in wake mode. Communications power draw is 20 mA receiving and 34 mA transmitting. Dirnberger [2] found that railcars are typically spend 65 percent of their time in yards. He also discovered that once a railcar is part of a consist, 18 percent of the total year is rolling time and 17 percent of the time is dwell time. Project researchers used these times to estimate annual power draws for the motes and PAN-Cs.

While in the yard, a railcar will be impacted several times. Researchers assumed 10 impacts when arriving in a yard, 20 impacts in the classification yard, and 10 impacts in the departure yard prior to the arrival of a locomotive. So there are an average of 40 impacts per yard visit. The mote activates based on acceleration from impact and scans for a network. Scanning 16 ZigBee channels for 2 seconds each, the power draw can be 0.3 mA-hr per yard visit.

The 2008 BNSF Annual Report and 2013 AAR statistics show an average of 40 trains/locomotive moves annually. This can also be assumed to represent the number of yard

visits per railcar annually. The power draw for 65 percent of the year as yard visits can then be calculated to be 486 mA-hr.

In an ad hoc mode of operation, the motes are always on and maintaining a radio link in Rx mode. The mote draws 1300 mA-hr for the typical train run-time of 39 hours (18 percent roll time for 40 trains). For 40 trains per year, the annual power draw is 52,000 mA-hr.

In a star topology rather than an ad hoc network, the motes only function as reduced functionality devices in a ZigBee network. They are on only when awakened by an alert, acceleration, or at proscribed times for messages. In this case, the total power draw is 5200 mA-hr. which is about a 2-year life for four AA Li-SOCL2 batteries.

4.4.2 PAN-C Power

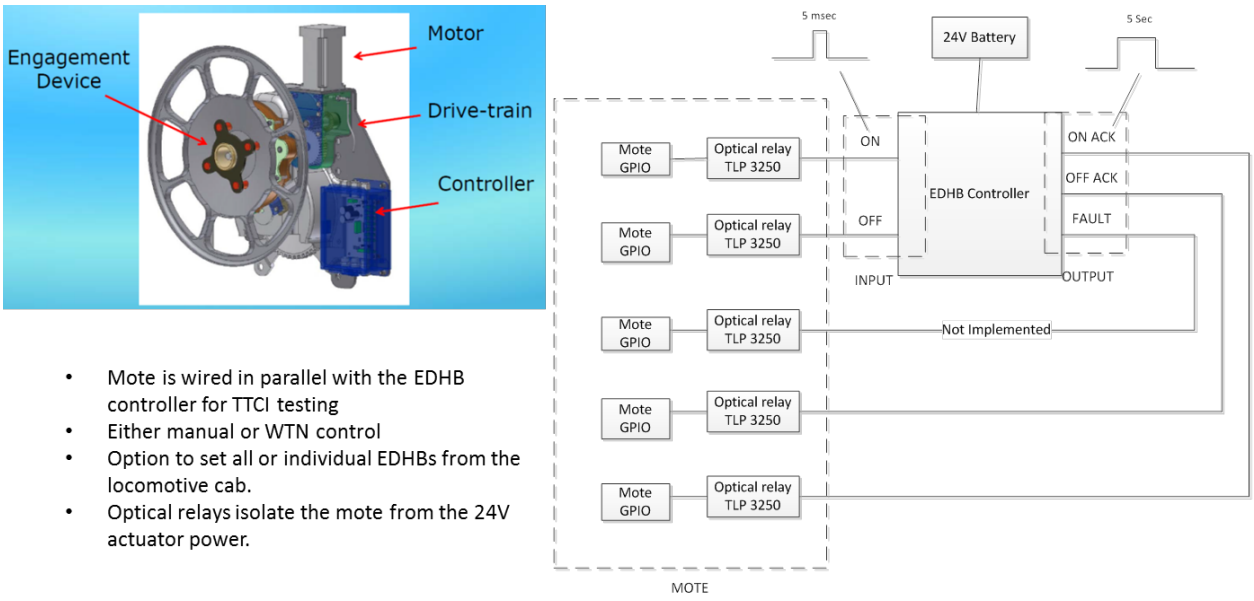
The PAN-C uses both the CC2538 ZigBee microcontroller and the GS1500M WiFi microcontroller from GainSpan. WiFi receive power is 65 mA and transmit power is 200 mA. The PAN-C will search for a WiFi network once awake from acceleration. Once it joins the WiFi network, it remains on and attempts to form the ZigBee network for that railcar.

As an ad hoc WiFi network, the PAN-C is on in receive mode for the entire time the network is formed. It is estimated that the PAN-C draws 3,350 mA-hr per train run and 134 A-hr annually. With four D-sized Li-SOCL2 batteries, the expected battery life is 0.5 year.

The power draw has not been optimized. Improvements are definitely possible. However, this analysis does lead to the conclusion that energy harvesting will be necessary to avoid frequent maintenance and battery replacement.

4.5 Electronically Deployed Hand Brake Actuation Using the Wireless Train Network

The motes are configured to work with a number of sensors and actuators. The current implementation is to operate the EDHBs developed by Sharma & Associates – already installed on railcars at TTC. The EDHB and its interface with the motes are shown in Figure 20. The motes duplicate the controls that the EDHB currently has through its box manual controls. The on-off control is through the motes' GPIO connections and is isolated by optical relays to decouple the electronics from the EDHB electronics. The mote can duplicate all EDHB controls and enables remote activation of all EDHBs in a train consist from the locomotive.



- Mote is wired in parallel with the EDHB controller for TTCI testing
- Either manual or WTN control
- Option to set all or individual EDHBs from the locomotive cab.
- Optical relays isolate the mote from the 24V actuator power.

Figure 20 Electronically Deployed Hand Brake Interface with Sensor Motes

The operation logic of the EDHB with the WTN is illustrated in Figure 21. The WiPU informs the WTN of the train speed. If there is motion, the EDHB cannot be engaged. The locomotive (enterprise) requests EDHB be engaged and that signal be passed to the WTN GW. The request is forwarded to all railcars in the network, and the PAN-C directs the request to the appropriate mote controlling the EDHB. A 5-msec pulse activates the EDHB controller. Both EDHB on and off signals are acknowledged to the locomotive for each actuator. The WTN GW and test laptop, or eventually the WiPU, coordinate the information for the engineer with red/green lights to indicate actuation of the EDHBs throughout the consist.

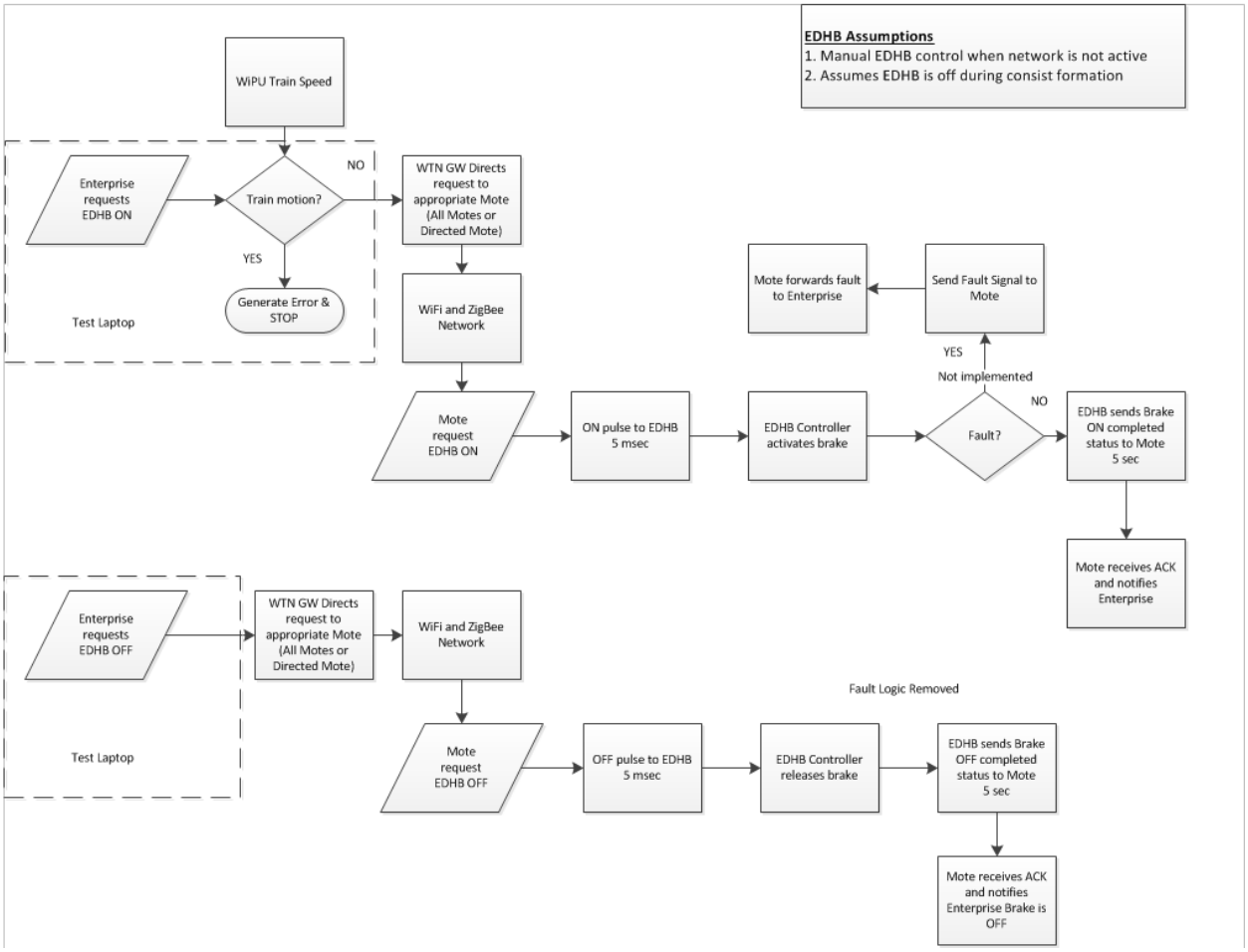


Figure 21 EDHB Implementation Logic

4.6 Wheel Bearing Sensors

The base program was developed to demonstrate the WTN while using wheel bearing sensors that duplicated the information from wayside hot box detectors and acoustic detectors. QNA developed a wheel bearing sensor that incorporates wheel bearing temperature measurement and an onboard microphone for acoustic information when mounted on the wheel bearing housing. In addition, since the most relevant information of a bearing's health may be found with an accelerometer, one was integrated into the sensor as well. The QNA wheel bearing sensor has the following functionality:

- Temperature
- Acoustic response
- Acceleration

This sensor was developed as both a research tool and as a means to evaluate the effectiveness of the WTN. There have been no plans to develop a detailed bearing health study on this program, as that work has been conducted by previous researchers. But the sensor developed to date has the potential to add to the data regarding bearing health and degradation.

The wheel bearing sensor schematic is shown in Figure 22. The mote functions as the communications component to the network. The advanced functionality of the sensor is primarily driven by the data processing requirements for the accelerometer and the microphone. The sensor is self-powered, with a separate battery pack. The sensor itself consists of two boards, one that contains the microphone and the thermistor and the other the accelerometer, as shown in Figure 23. Data processing occurs in the DSP housing containing a TI C5535 EZDSP board.

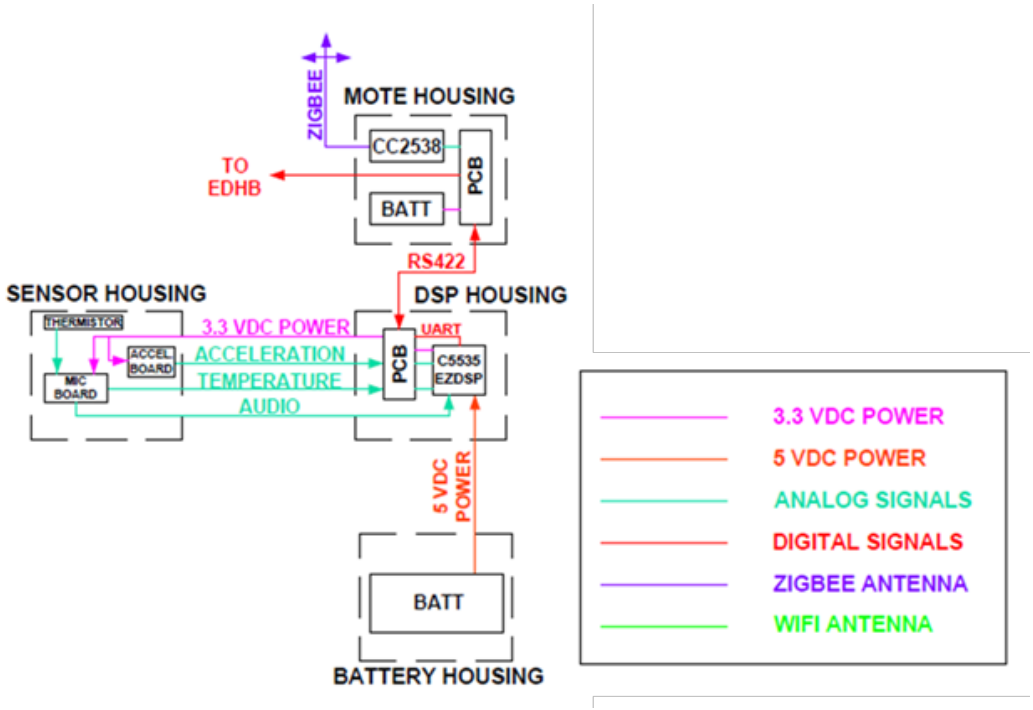


Figure 22 Wheel Bearing Sensor Schematic

The wheel bearing sensor and its mounting configuration are shown in Figure 24. The sensor itself is shown in Figure 25. The sensor is mounted to the wheel bearing housing for a direct measurement of the temperature and vibratory environment. The sensor has a thermistor mounted in the housing. The direct heat path from the housing to the thermistor will produce accurate housing temperatures. Research has shown the correlation between the housing temperature and the bearing temperature is offset by about 20 °C.

The sensor has two PCBs. One is for the accelerometer and the other for the microphone and thermistor. The microphone is directed toward the housing through an opening in the housing. The 3-axis accelerometer is hard-mounted to the sensor housing with the board oriented vertically to minimize deflections.

To save on development costs, the electronics for DSP were designed using evaluation boards mounted in a truck well due to their size. Additionally, the DSP housing is self-powered with a battery pack that is also located in the well location.

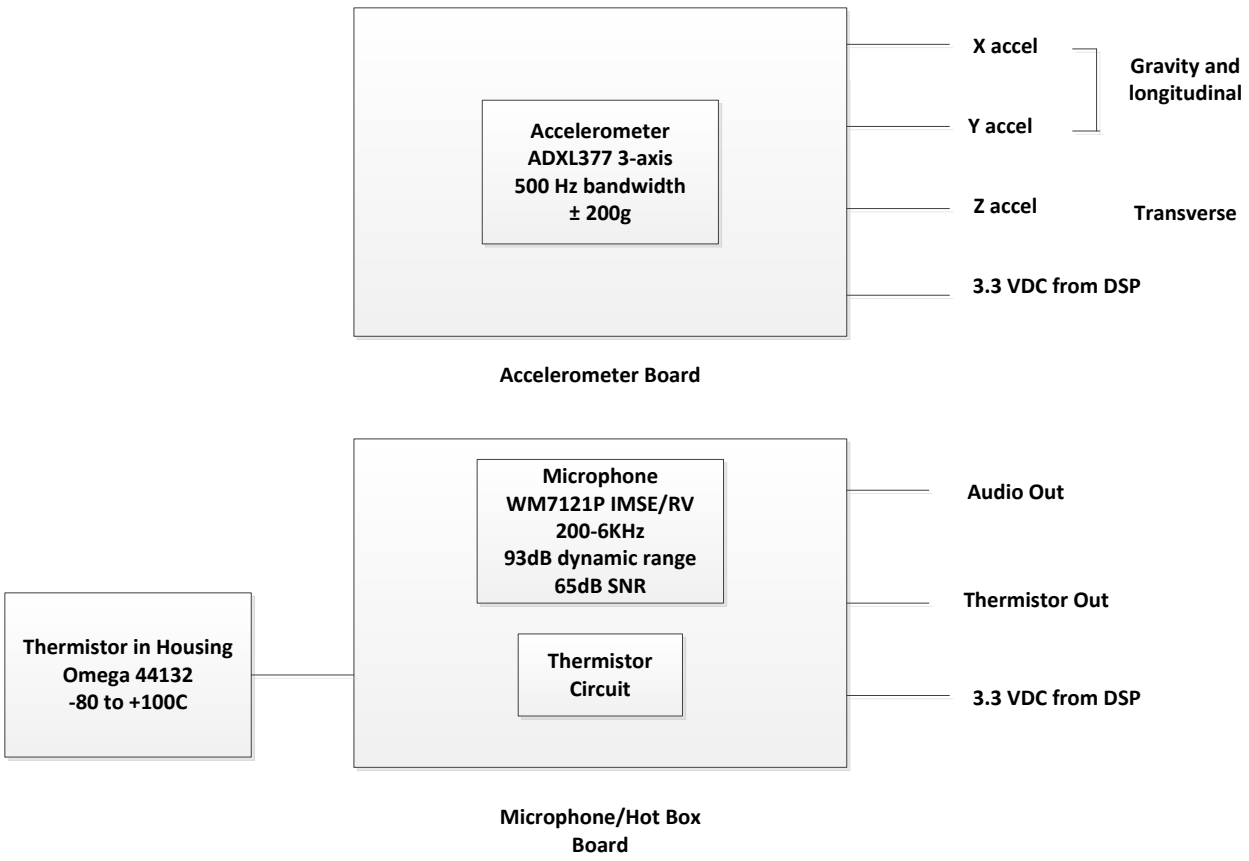


Figure 23 Wheel Bearing Sensor Details

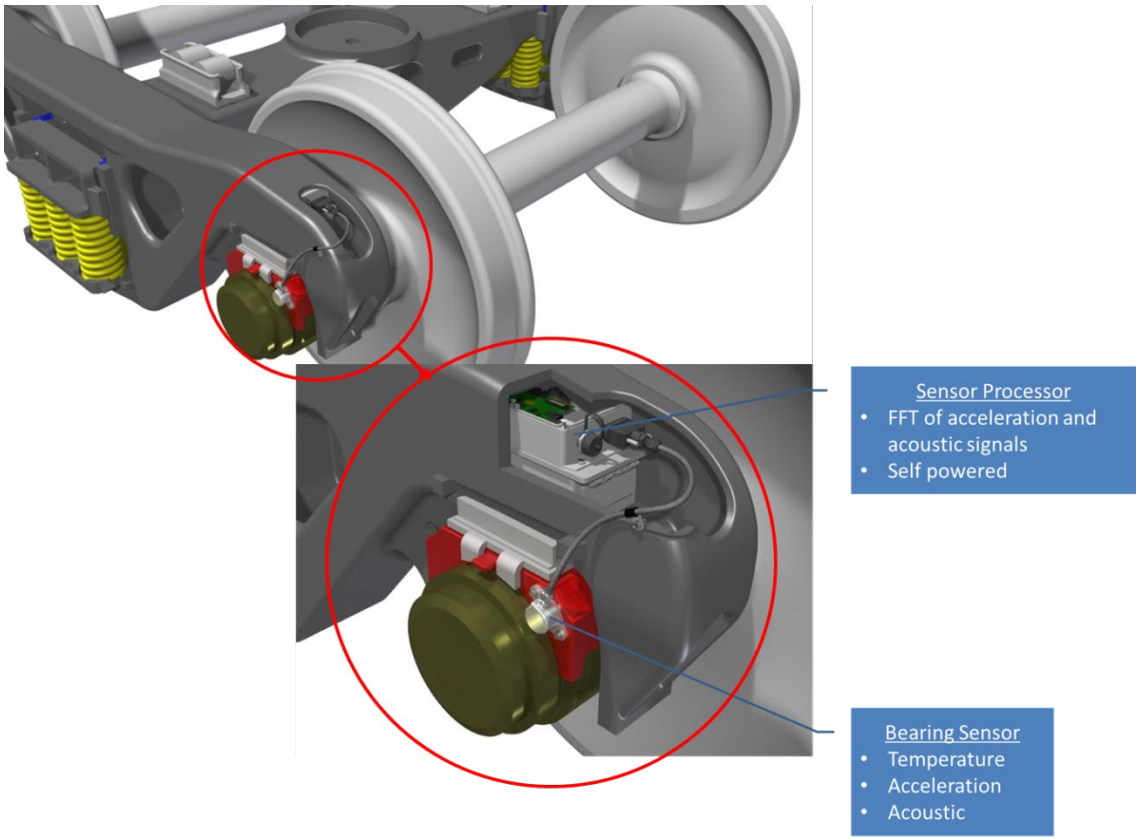


Figure 24 Wheel Bearing Sensor and Mote Location

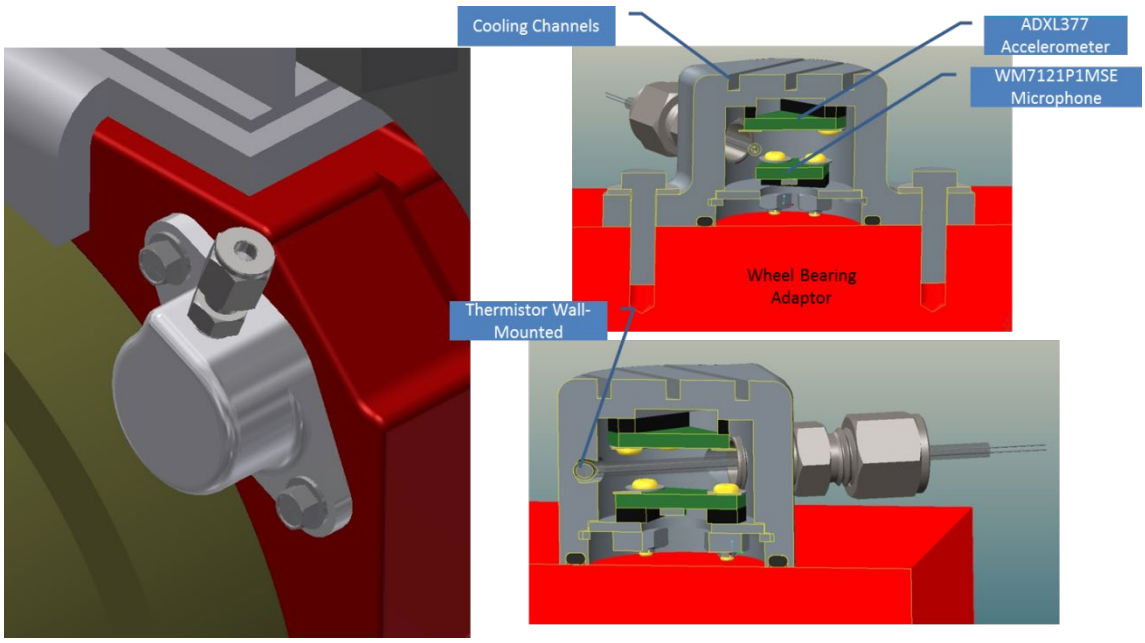


Figure 25 Wheel Bearing Sensor Capability

The sensor and the DSP boards are designed to store the data as well as conduct fast Fourier transforms (FFTs) on the data. The FFT is conducted in real time and evaluated against three alert levels: (1) Warning, (2) Alert, and (3) Critical. Alerts are triggered in specific frequency bands that correlate to bearing critical frequencies. Each frequency band has its own alert levels. A lower bound threshold is implemented to limit repeated alerts. During WTN testing, initial test runs will establish baselines. Thresholds will be artificially set to trigger alerts in order to test the sensor functionality and its implementation in the WTN. Later work can establish the use of this sensor in wheel bearing health monitoring.

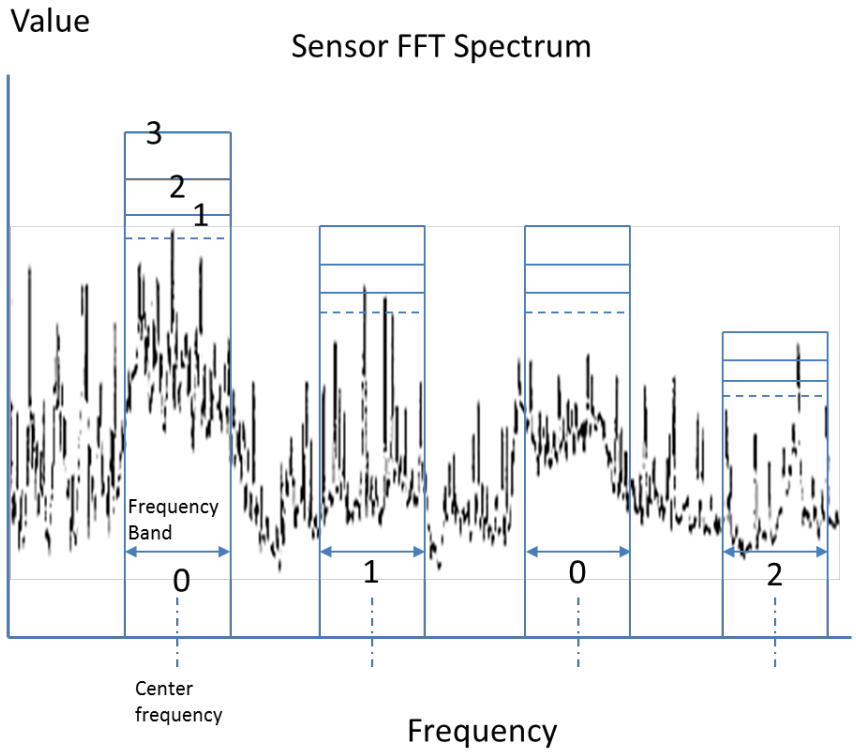


Figure 26 Microphone and Accelerometer FFT Alerts

Details of the microphone, accelerometer and thermistor logic is shown Figures 27 to 29. Threshold FFT sound levels are set at 100 Hz and 500 Hz to three levels (30 dB, 40 dB, and 50 dB). These are the three alert levels for warning, alert, and critical messages. These threshold levels can be modified via commands from the enterprise, such as from a dispatch center. The option to simply record sound data is enabled on command for data acquisition. A similar logic is in place for the accelerometer. Maximum FFT g-loads are calculated in all three axes. A moving 10-minute average is calculated and compared to the given acceleration threshold levels for frequencies of interest. All thresholds and frequencies are adjustable by the enterprise after network formation. Temperature of the bearing housing is measured at 1 sample/minute. A 10-minute moving average is calculated and compared to the temperature thresholds.

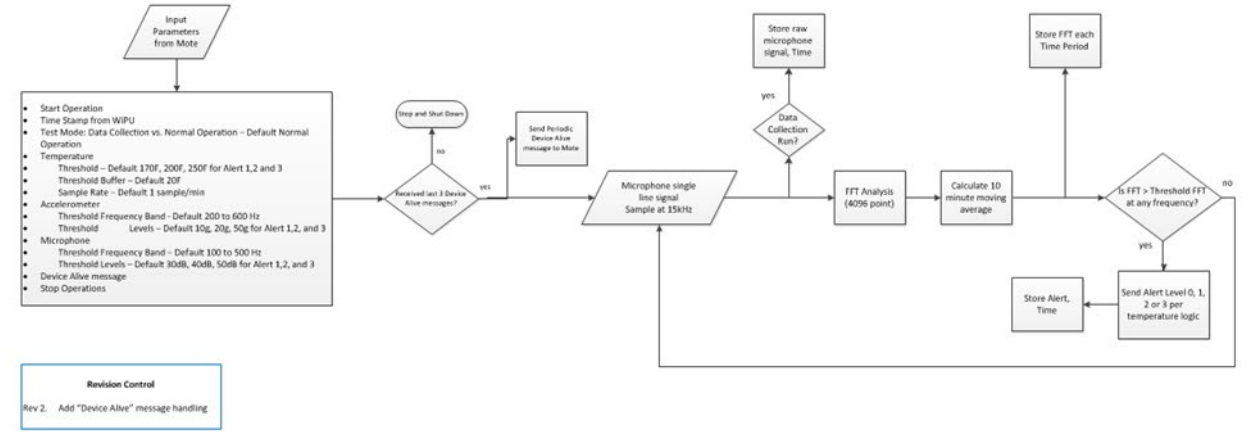


Figure 27 Microphone Implementation Logic

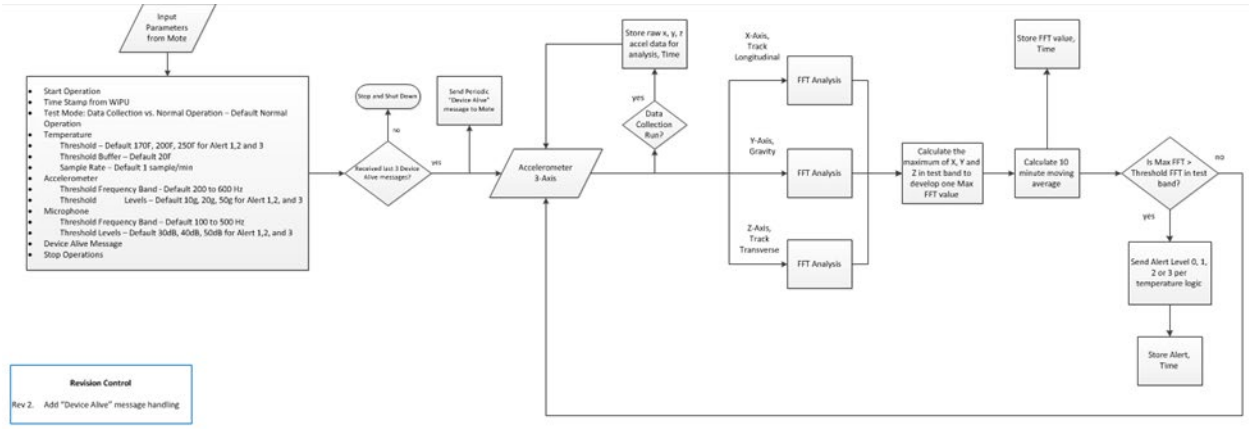


Figure 28 Accelerometer Implementation Logic

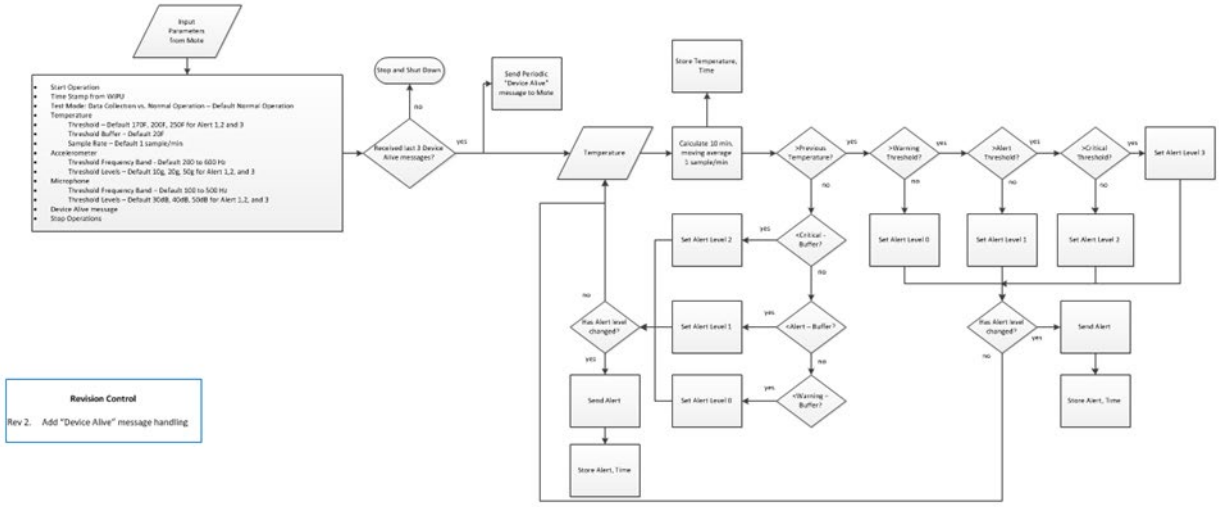


Figure 29 Thermistor Implementation Logic

4.7 Network Formation

The WTN is formed after the train is assembled in the departure yard. The keys to beginning network formation are installation of the End-of-Car device (EOD) and acceleration as the train departs, as shown in Figure 31. Train acceleration is felt by the PAN-C accelerometers. The WTN gateway receives a signal from the WiPU or Head-of-Train device that the EOD has been installed. The WTN gateway then sends beacons looking for PAN-C communication. Each railcar PAN-C wakes and searches for a WiFi beacon. When received, it sends an acknowledgement and joins the WTN network.

Since there is slack in the couplers, the railcars will accelerate in order from the locomotive to the back. So the time of receipt of the PAN-C signal directly correlates to the railcar order in the consist. With accelerometers always oriented in the same direction, toward the A-end of the railcar, the sign of the acceleration is a direct correlation to the orientation of the railcar in the consist. So the act of joining the WTN WiFi network also automatically defines the order and orientation of the railcars in the consist, as in Figure 30. This feature has not been implemented in the base program due to funding constraints. Details of the consist formation can be found in the Appendix.

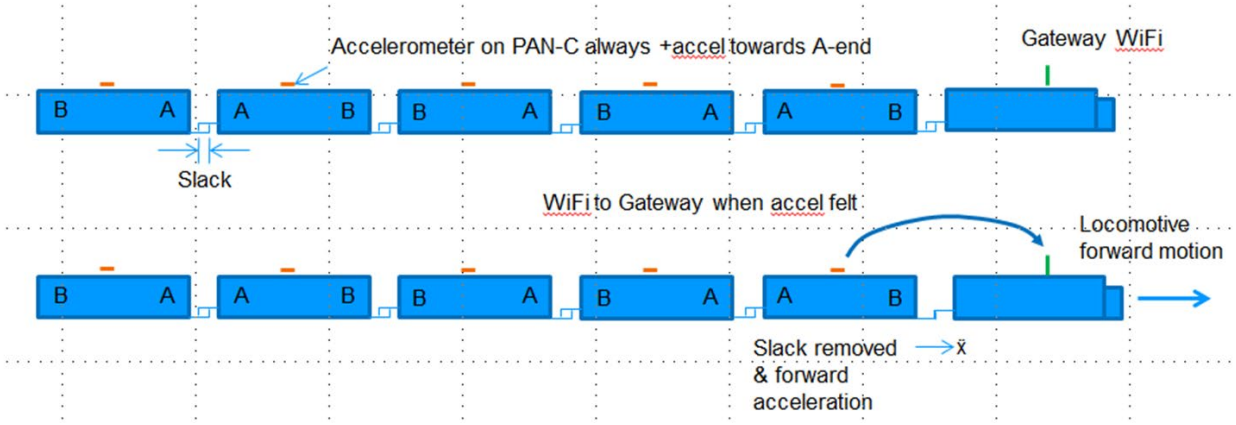


Figure 30 Automatic Consist Order and Orientation with the Wireless Train Network

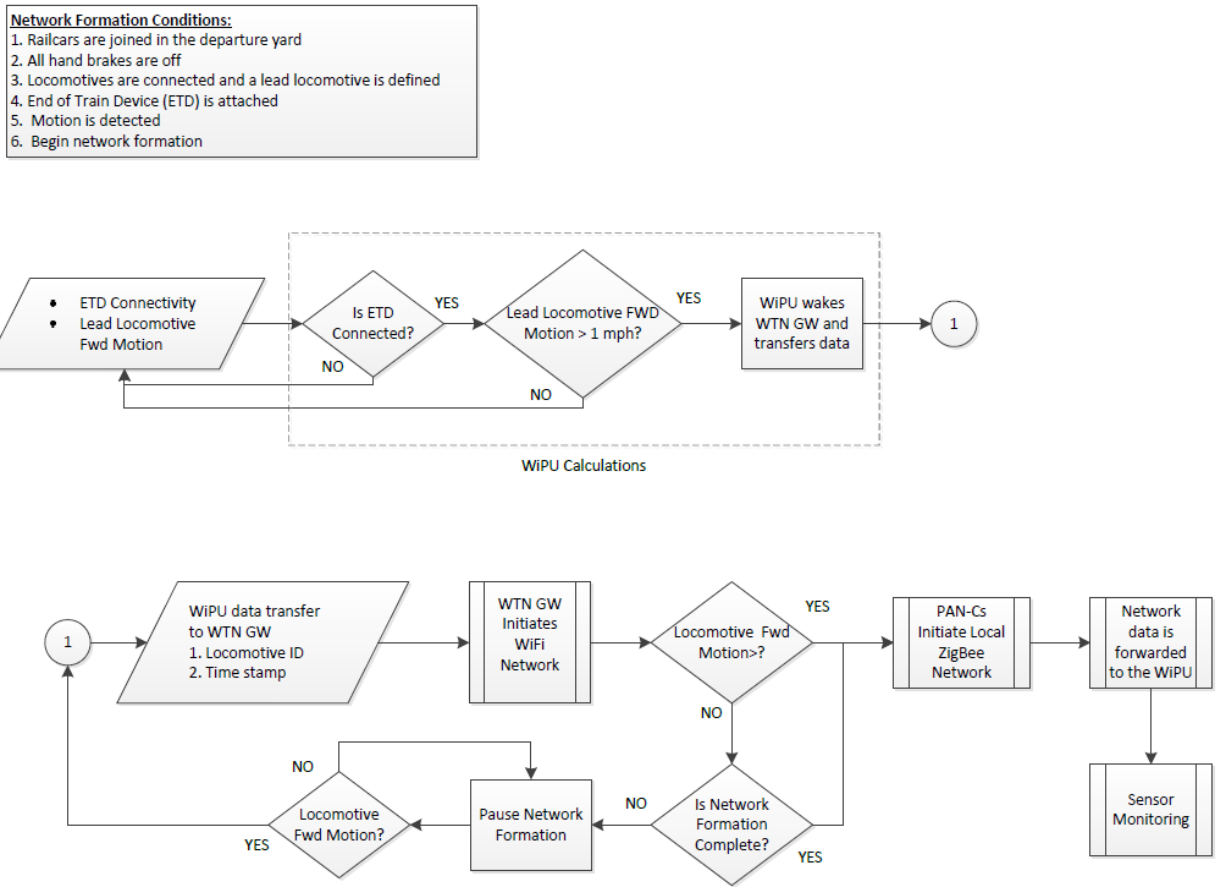


Figure 31 Network Formation Logic

Following the PAN-C joining the WiFi WTN network, the PAN-Cs then send ZigBee beacons for the motes. Motes wake periodically based on their accelerations. If the railcar is not in a train consist and moving, the PAN-C will not have sent a beacon, so the motes go back to sleep. If the train consist is formed, then the motes will hear the beacon and join the railcar ZigBee network. The mote will not confuse adjacent railcar beacons as they were configured for a particular railcar when originally mounted. The motes return to sleep mode and will wake and look for

queries and will report status periodically (currently in 10-minute intervals) to conserve power. This enables the 2-year battery life currently estimated for the motes.

5 Firmware Implementation

The firmware is implemented as a modular structure where functionality can be added or removed, depending on the capabilities required by a particular mote in the network. The sensor motes are capable of ZigBee communication only and thus have firmware capable of general processing and ZigBee communication. The cluster heads have the added module capable of driving the IEEE 802.11 WiFi communication. The WTN gateway mote on the locomotive, in addition to ZigBee and WiFi, can also communicate with the WiPU over wireless or wired Ethernet. The sensor motes can communicate with an acoustic wheel bearing sensor and the EDHB.

5.1 Real-Time Operation System

The RTOS used is the Z-stack from TI. This is directly compatible with the CC2538 microcontroller used as the brains of each mote. This version of Z-stack is capable of self-healing IEEE 802.15.4 mesh network with up to 400 motes in a single cluster. The RTOS has built-in support for many-to-many as well as many-to-one routing. The over-the-air firmware upgrade option is also made available as part of this RTOS software stack. The Z-stack allows use of the sleep modes offered by CC2538 to allow power-efficient firmware development.

In a cluster, the Z-stack RTOS differentiates between three different classes of a device:

a. ZigBee Coordinator: This is a fully functional device (FFD) which forms the network. This mote sets up the channel and Personal Area Network ID and then starts the network. All the security protocols to be used in the network and their settings are established by this device type. Once the network is set up, the coordinator devolves to the role of a router or may drop off the network totally.

b. Router: This is also a FFD that allows routing, assisting the children mote to communicate in a power-efficient way by allowing children motes to go to sleep and allows new devices to join the network.

c. End-Device: This device type is generally found in the motes that form the leaf motes of the network in a cluster. They do not have the responsibility to maintain network infrastructure and hence can go to sleep to save power. These are also called reduced function devices (RFD).

The general structure of the network in a ZigBee cluster formed using motes running the Z-stack RTOS and application firmware is as follows:

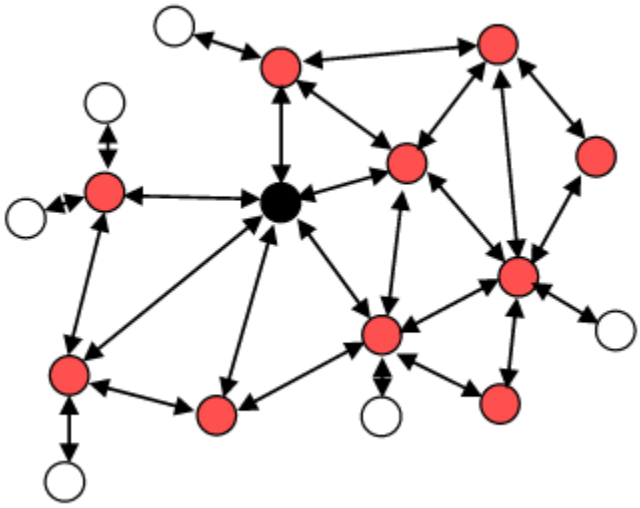


Figure 32 General Zigbee Cluster Configuration Diagram

In the figure above the mote in black is the cluster head (PAN-C) equipped with both ZigBee and WiFi radio. The motes in red are the router motes relaying traffic from the motes which are associated to them as children. The motes in white are the RFD leaf motes.

5.2 Execution Structure

The firmware is structured around processes. Every process has its own pool of memory and shares the global message queue to listen to messages from other processes or to put in messages destined to other processes. Every process has a task ID number assigned to it. Any software or hardware interrupt preempts any running process to execute the associated interrupt service routine before resuming execution where the previously running process was preempted.

The firmware is designed to be event-driven. The system goes into a low-power mode of the microcontroller when there is no event of interest occurring in the system. Any timely activity scheduled, like general housekeeping of routing tables, polling of sensors, etc., generate an event. Each of these events generates a message that is put in the global message queue destined for a process. The process contains an event loop that keeps listening to this queue for events addressed to it. Upon receiving one such message it executes the callback function associated with the event type. The flow is illustrated in Figure 33.

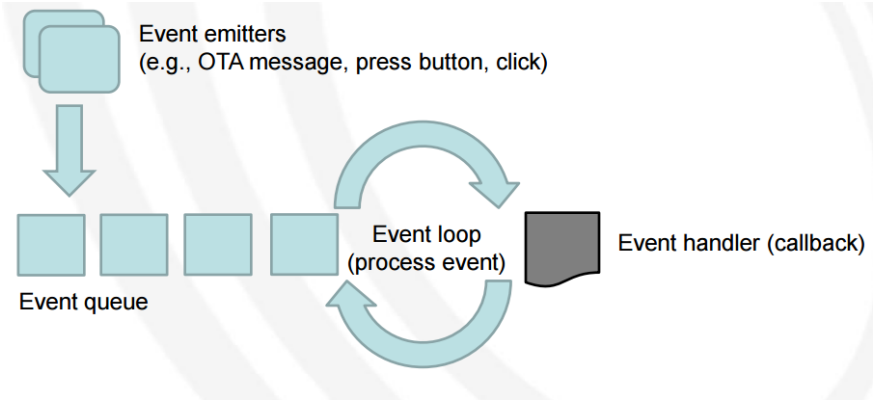


Figure 33 Event Driven Firmware Loop Flow

Every process has an Init() method, used to initialize required parameters at power-up or reset – then the event processing loop takes over. The process is illustrated in Figure 34.

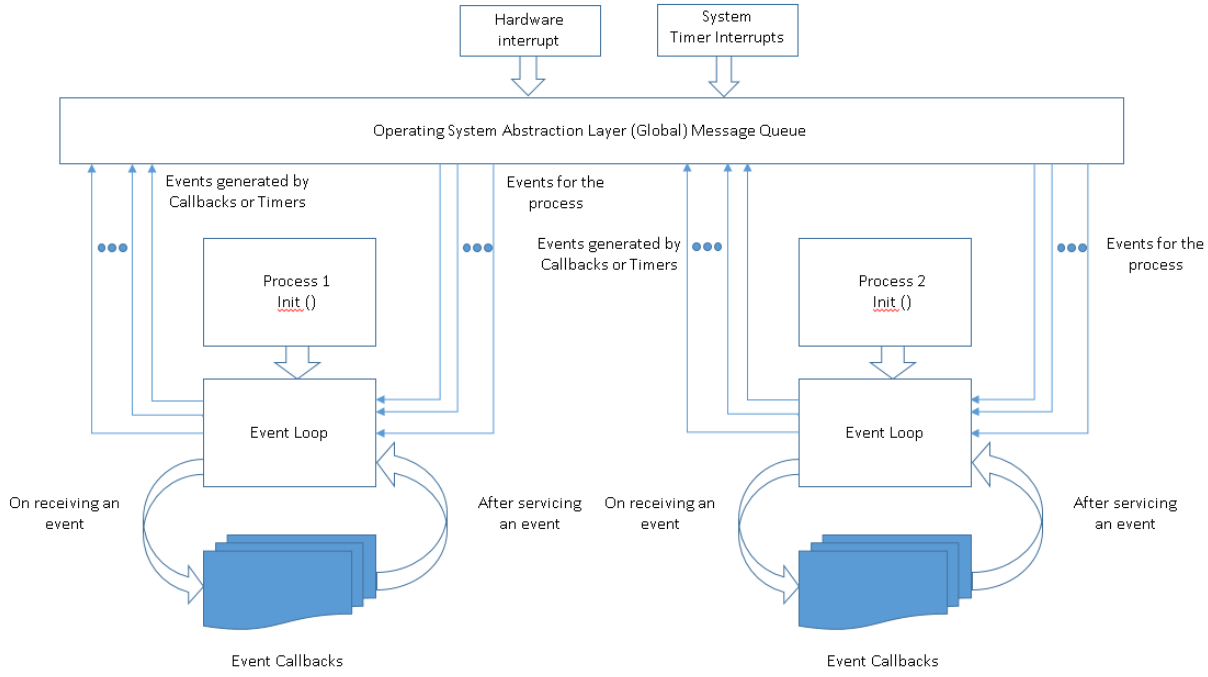


Figure 34 Event Processing Loop

The round-robin scheduling is used in the firmware. In the absence of any interrupts, the tasks run to completion instead of being preempted. In case an interrupt occurs, the execution follows the path described at the beginning of this section.

The task scheduling methodology of the firmware is shown in Figure 35.

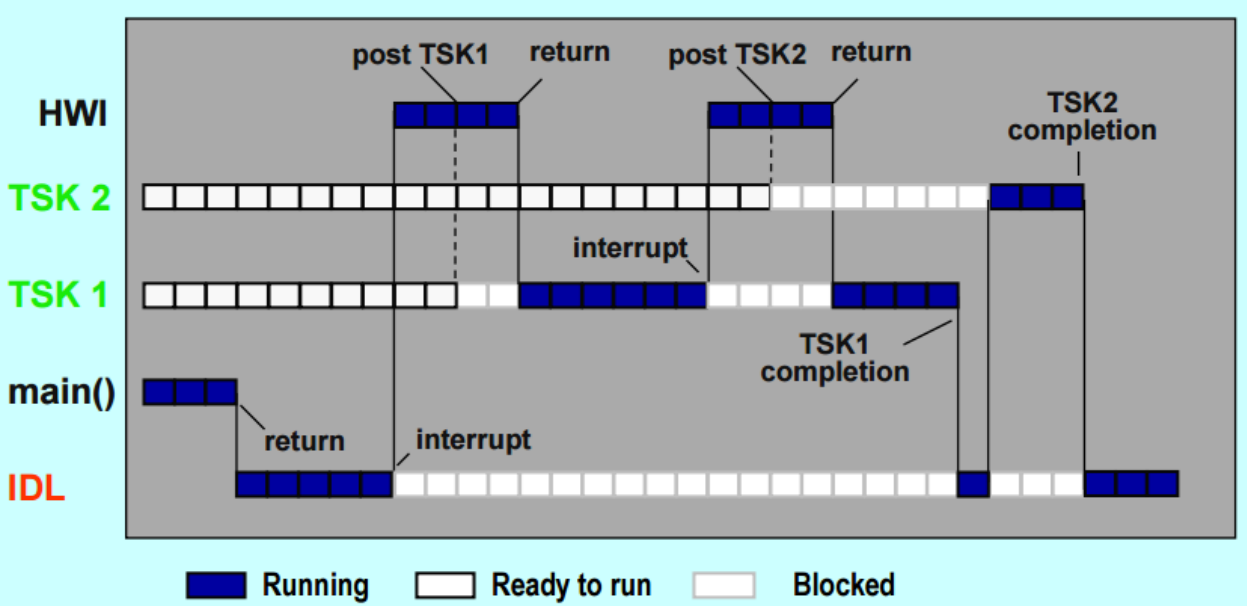


Figure 35 Task Scheduling Methodology

During the idle states indicated by IDL in the above diagram, the system moves to a low-power state to save power.

5.3 Processes

The following processes run as part of the firmware over and above the processes required by the RTOS:

a. **WiFi radio monitoring process:** This process deals with the activity to and from the WiFi module. The task of initializing the module at power-up or reset is performed by this module. The module also configures the interface between the module and the microcontroller over SPI at a rate of 2 Mbps. It maintains a data packet queue each for incoming packets over WiFi and outgoing packet destined for transmission over WiFi. The queue is set up to service 10 incoming packets and 10 outgoing packets at any one time. Any packets arriving at the incoming queue or outgoing queue over and above this capacity is silently discarded.

b. **AODV routing protocol process:** The ad hoc WiFi connectivity between all the cluster heads follow the Ad-hoc On-Demand Vector Routing protocol to discover, maintain and repair routes. This process is responsible for route discovery and route maintenance by servicing route discovery requests from the WiFi radio monitoring process and addressing failed route notifications from the same process, prompting a route repair.

c. **ZigBee radio monitoring process:** This process services the traffic over the ZigBee radio. Analogous to the WiFi radio monitoring process, this process too maintains an incoming and an outgoing queue for ZigBee packets. The capacity of queues is 10 packets. The packets beyond this capacity are silently dropped.

d. **Ethernet process:** This process controls the communication over the Ethernet with the WiPU in the applicable motes. Since the communication between the WiPU and WTN

gateway mote will be sparse, the process only maintains an incoming and outgoing queue of three messages in each queue.

e. **HTN protocol message parser process:** This process deals with interpreting the received messages over the WiFi or ZigBee radio and executing applicable methods accordingly. It also generates reply messages where applicable and hands them over to the corresponding radio control process.

f. **Ethernet message parser process:** This process is tasked with the handling of messages received from and messages destined to the WiPU.

g. **Housekeeping process:** This process posts periodic events that scan all the tables and queues for the age of the artifacts in them. Any artifact that exceeds the maximum age set for that kind of data – for example, packets destined for transmission, neighbor table entries, etc., – are deleted from the corresponding tables and queues. This task is also the one that updates the age of the artifacts in case they are still not over the age threshold.

h. **Update process:** This process is started at power-up or reset. It reads the non-volatile memory of the microcontroller for system parameters. If the process detects that the system properties are inconsistent or the mote has not yet been configured for the network, it switches to a default PAN ID and listens on a default channel over the ZigBee radio for configuration updates. The process is also responsible for controlling the configuration update process once an update source is detected. At the end of the update process flow the mote will be reset. In case the proper system parameters and configuration data is found, the process switches to the normal flow by creating the other processes mentioned above.

5.4 Optimizations

There are several optimizations conducted in the firmware that allow the application to be light-weight, responsive, robust, and power-efficient.

a. All tables and queues are declared as static structures with pre-allocated memory. This reduces the processing overload due to dynamic memory allocation and related garbage collection.

b. The Z-stack RTOS provides the capability of 16 events per process. This is restrictive in certain situations. The firmware provides up to 256 events per process, removing the RTOS constraint.

c. Capabilities of the RTOS beyond what is required is removed to reduce the Flash and RAM footprint of the RTOS stack.

d. Every layer of processing does not create an incoming and outgoing data queue. There are single incoming and outgoing queues for each radio interface. There are block descriptors that facilitate the other processes' access to data in these queues. The block descriptors contain information related to the start and end location of a packet in a queue and all other details as retry counters, age counters, etc. Whenever a process needs access to data or information about the data in a queue, it looks up the corresponding block descriptor. This reduces the overhead caused by repeated copying of the same data and also reduces the memory required by the application to function.

e. Packets of certain types, like alerts, can be assigned higher priority. These packets will be placed in the outgoing or incoming queues at the head such that they get processed at the earliest. This allows for quality-of-service for the HTN packets flowing through the system.

f. The acoustic wheel bearing sensor and the EDHB are checked on interrupt and not polled. This frees up processing power for the rest of the system.

g. Whenever activity is not detected for a defined period of time, the mote moves into a low power state to save energy.

5.5 Wheel Bearing Sensor Interface

The AWBS is connected to the microcontroller over an SPI interface operating at 16 MHz. The sensor consists of a DSP, which requires setting up with relevant parameters. These parameters are set in the microcontroller's non-volatile memory during configuration. The firmware sends these to the sensor when the sensor switches on. The firmware is also capable of bi-directional communication with the sensor along with dynamic disconnect detection.

The following messages are exchanged between the AWBS and the firmware:

a. **Heartbeat or Device Alive:** This message is sent periodically from the mote to the sensor package and from the sensor to the mote. The absence of this message from any side for a preset interval will indicate catastrophic fault and the other side will assume that the mote/sensor is dead. On the mote side, if a sensor fault is detected, a message can be sent to the laptop via HTN for logging of the event. This message is called the Sensor Fault Alert Message, shown further below.
b. **Negative Acknowledgement:** This message is sent when a previous message was faulty or contained invalid parameters.
c. **Acknowledgement:** This message is sent when a previous message was okay.
d. **Start Sensor Operation:** This message is sent to the sensor to begin operation.
e. **Stop Sensor Operation:** This message is sent to the sensor to stop operation.
f. **Mode:** This message is sent to the sensor to select an operating mode: 0-normal; 1-data collection.
g. **Time:** This message is sent to the sensor to provide a UNIX time for it, counted as seconds elapsed since the epoch of 1970-01-01.
h. **Temperature Threshold:** This message is sent to the sensor to set the temperature alert thresholds. All thresholds are 16 bit values representing degree Celsius.
i. **Accelerometer Band:** This message is sent to the sensor to set the accelerometer frequency band limits.
j. **Accelerometer Threshold:** This message is sent to the sensor to set the accelerometer alert thresholds.
k. **Microphone Band:** This message is sent to the sensor to set the microphone frequency band limits.
l. **Microphone Threshold:** This message is sent to the sensor to set the microphone alert thresholds.

m. **Temperature Alert:** This message is used by the sensor to proactively send alert indications from the temperature sensor in the package. This is coupled with a GPIO interrupt line from sensor to mote.
n. **Accelerometer Alert:** This message is used by the sensor to proactively send alert indications from the accelerometer in the package. This is coupled with a GPIO interrupt line from sensor to mote.
o. **Microphone Alert:** This message is used by the sensor to proactively send alert indications from the microphone sensor in the package. This is coupled with a GPIO interrupt line from sensor to mote.
p. **Battery Status:** This message is used by the sensor to proactively send updates about the sensor packages battery charge.
q. **Sensor Data Read Request:** This message is sent to the sensor when we need to read the current value or status of a particular sensor or set of sensors.
r. **Sensor Data Read Reply:** This message is sent by the sensor back to the mote with the requested data.

5.6 Electrically Deployed Hand Brake Interface

The configuration parameters determine if there is an EDHB attached to the mote. If it is, then the EDHB is actuated on commands from the WTN gateway, and the status of the EDHB is also read back and sent to the WTN gateway.

There are four lines connecting to the EDHB for purposes of actuation. There is another set of four lines on which the status is read back from the EDHB. The EDHB asserts the signal on each status line for 5 seconds; this is read by the firmware to determine the status of each of the four lines. This information is then sent to the WTN gateway.

6 Conclusion

The program to date has defined a practical wireless train network for freight trains: a hybrid technology network utilizing ZigBee technology within each railcar and WiFi technology to create an ad hoc train network. The network is based on custom hardware using off-the-shelf PCBs for development. Motes and PAN-Cs have been fabricated for demonstration testing at TTC. The network software has been developed and compiled into the hardware for test evaluation. Errors in hardware assembly have delayed network test and evaluation at TTC.

A specialized wheel bearing sensor has been developed for mounting on retainer blocks. The sensor contains three measurement systems: a thermistor, a 3-axis accelerometer, and a microphone system for acoustic measurements. The sensor components were fabricated and the firmware has been developed. Data can be output has straightforward data acquisition, or it can be compared to thresholds for initiating alerts.

The WTN can be used with actuators. A demonstration was designed and built to activate and deactivate EDHBs developed by Sharma & Associates. This can enable the activation of hand brakes remotely from the locomotive for every railcar.

The WTN was developed to self-initiate from a locomotive as it leaves the departure yard. Activation is based on attachment of the End-of-Train device and forward acceleration as measured in each railcar. WiFi beacons from the locomotive gateway are detected by each railcar in series as they accelerate forward. That timing determines the railcar order in the train. The direction of forward accelerations determines the railcar orientation. That data is automatically assembled as the train departs and can be uploaded to the enterprise.

7 References

1. Sushanta, M.R, Hempel, M., Sharif, H., Punwani, S. K., and Stewart, M. (October 2012). Hybrid Technology Networking: A Novel Wireless Networking Approach for Real-Time Railcar Status Monitoring. *Proceedings of the ASME 2012 Rail Transportation Division Fall Technical Conference.* Omaha, NE.
2. Dirnberger, J.R., and Barkan, C.P. (September 2006). Improved Railroad Classification Yard Performance Through Bottleneck Management Methods. AREMA.

Abbreviations and Acronyms

Abbreviation or Acronym	Name
AAR	American Association of Railroads
ACCEL	Accelerometer
ACK	Acknowledgement
ADC	Analog to Digital Converter
AODV	Ad hoc On-Demand Vector routing
API	Application Program Interface
APP	Application Layer
APS	Application Sublayer
ARM	Advanced RISC Machines architecture
AWBS	Acoustic Wheel Bearing Sensor
COTS	Commercial Off-The-Shelf
DSP	Digital Signal Processing
DSP	Digital Signal Processing
EAddr	Extended Address
ECP	Electronically Controlled Pneumatic brakes
EDHB	Electronically Deployed Hand Brake
EOD	End-Of-Train Device
ETD	End-of-Train Device
FFD	Full Function Device
FFT	Fast Fourier Transform
FRA	Federal Railroad Administration
FSM	Finite State Machine
GHz	Gigahertz
GPIO	General Purpose Input/Output
HTN	hybrid Technology Network
Hz	Hertz
I2C	Inter-Integrated Circuit
ID	Identification
IDL	Idle

Abbreviation or Acronym	Name
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IPv4, IPv6	Internet Protocol version 4, Internet Protocol version 6
ISM	Industrial, Scientific, and Medical Radio Band
ITU	International Telecommunication Union
kbps	Kilobit per second
MAC	Medium Access Control layer
MIC	Microphone
OSI	Open Systems Interconnection model
PAN-C	Personal Area Network Coordinator
PCB	Printed Circuit Board
PHY	Physical layer
QNA	QinetiQ North America
RAM	Random Access Memory
RF	Radio Frequency
RFD	Reduced Function Device
RTOS	Real Time Operating System
RX	Receive Mode
SAddr	Short Address
SAP	Service Access Point
SPI	Serial Peripheral Interface
TCP/UDP	Transmission Control Protocol/User Datagram Protocol
TEL	Telecommunications Engineering Laboratory
TTCI	Transportation Technology Center, Inc
TX	Transmit Mode
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
WSN	Wireless Sensor Network
WTN	wireless train network
ZB	ZigBee

8 Appendix: Mote and Network Operation – Draft Specifications

Date: 3/9/15

Revision: 6

Mote and Network Operation – Draft Specifications

8.1 Mote Installation

1. Initial mote installations for a railcar in maintenance facility or siding away from functioning PAN-Cs other than that of the installed railcar.
2. Two types of motes are used. A dedicated PAN Coordinator (PAN-C) and Full Function Devices (FFD) or motes. There are currently no plans for Reduced Function Devices or end motes. Only the PAN-C has both ZigBee and WiFi capability. The motes (FFD) have only ZigBee capability. Future use of the term "motes" refers to FFD with only ZigBee and PAN-C implies both ZigBee and WiFi. The PAN-C has a larger battery pack to accommodate higher power requirements for WiFi. Summary of mote differences:
    a. PAN-C (1 per railcar, up to 100 per train for 100 car consist)
        i. ZigBee and WiFi (CC2538 microcontroller + GS1500M WiFi module)
        ii. 2.4GHz ZigBee and WiFi
        iii. Larger battery pack (Size T.B.D.)
        iv. Accelerometer and potentially a gyroscope
            1. Single axis accelerometer
            2. Always oriented with "front" pointed toward the B-end of the railcar
            3. Accelerometer installed with primary axis towards "front" of the PAN-C
        v. Installed at high point of railcar for best WiFi line-of-sight
    b. Mote (2-12 T.B.D. per railcar, for a 100 car consist up to 1200 per train)
        i. Only ZigBee (only CC2538 microcontroller)
        ii. Smaller battery pack (size T.B.D.)
        iii. No accelerometer
        iv. Arbitrary installation orientation
    c. WTN Gateway (1 per locomotive. Multiple locomotives possible but only one active WTN Gateway. Likely the lead locomotive)
        i. Both the CC2538 processor (CC2538SF53)and the GS1500M WiFi
        ii. Only a WiFi mote
        iii. Have accelerometers. Each locomotive mote reports the time difference from receipt of WiFi beacon to start of acceleration and motion. Distributed power locomotives are included in the automatic consist

makeup. In distributed power, all locomotives have a mote, but only the lead locomotive has the active WTN Gateway mote.

iv. Power T.B.D.
    1. Same battery pack as PAN-C for WiFi
    2. Alternatively DC powered from WiPU or DC converter

3. Insert mote battery
4. Initial startup check on systems
5. Starts in sleep mode with initial configuration parameters
6. Insert sensor battery – Sensor start-up protocol
7. Mote and sensor are physically installed on railcar 1st such as mote and hot box
8. Test Sensors
    a. Test sensor is a single housing with thermocouple, microphone and accelerometer to be mounted to the wheel bearing retainer housing
    b. A separate housing holds the sensor processor to be located in a truck side frame recess
    c. Sensor processor is a C5535 DSP developer board (TMDX5535EZDSP)
    d. Cable connection between the sensor and the processor
    e. Cable connection between the sensor processor and the mote.
    f. Sensor to mote Application Programming Interface (API) is T.B.D.
9. Waits for a test sensor connection (external plug-in)
10. With a test sensor connection
    a. Wakes up mote
    b. Checks for sensor information through API – stores data
        i. Initializes the sensor
        ii. Uploads sensor status, ID, type and battery state
    c. Waits for a railcar ID and location configuration
    d. Railcar ID's are of the format XXXX 000000
11. Requires <u>manual setting of railcar ID</u> and location on the railcar (BNSF1074, L1 to L4 and R1 to R4 sequenced from B-end (with hand brake)). Need a compact way to represent location. Configure with ID prior to joining intra-car ZigBee(ZB) network. Needs a laptop, iPhone app or other input device to configure mote at installation. Use laptop for initial tests. <u>Now motes always form an intra-car network only with other motes with same railcar ID.</u>
    a. Accomplished through ZB connection to special PAN ID when sensor mote determines it needs configuration data. This special PAN ID is only served by the mote configuration device, which is the PAN coordinator for this special PAN. No other device shall have this role.
    b. The motes periodically scan for the special PAN and, when found, connect to it using pre-configured parameters.
    c. Motes then identify themselves and request bulk configuration update.

d. Mote configuration devices, upon approval, provide bulk configuration update to requesting mote.
e. Mote receives bulk configuration update, disconnects from PAN, and then performs a soft reset. The soft reset shall result it discovery of the valid configuration data and their use for all subsequent startup activities.

12. Railroad mechanics will have the mote configuration device (laptop for now), or will access spares located on locomotives
13. Each mote now has its unique ID from factory, railcar ID, and railcar installed location prior to joining a network.
14. Install the PAN-C at high point of the railcar for best WiFi line-of-sight.
15. The ZigBee network formation is delayed until after consist formation to save power.
16. All motes are in sleep mode awaiting the PAN-C network activation
    a. Clarification: Motes periodically wake up in order to scan for their expected railcar PAN ID. If found, they connect. The PAN-C will only activate the PAN coordinator functionality (and thus starting up the PAN for the railcar) when the consist link via WiFi has been formed.
17. PAN-C ZigBee network activation follows joining the WiFi train network as discussed below.
18. Premise:
    a. Railcars sorted for consists in the classification or hump yard
    b. Switchers bring segments of railcars from the classification yard to the departure yard to form the consist
    c. When the train is ready to be formed, the locomotives join the consist and are connected.
    d. The End of Train Device (ETD) is attached to the last railcar when the train is formed and ready to leave the yard (457.9375 MHz at 2W for 3-5 mile transmission). Head End control signals at 452.9375 MHz.
    e. The wireless train network (WTN) forms as follows as the train leaves the departure yard.
        i. The WTN will determine the railcar order by timing the railcar acceleration and/or jolt when starting to move.
        ii. The railcar orientation will be determined by the direction of the acceleration as each PAN-C is oriented towards the railcar B-end
        iii. The ZigBee network forms following the PAN-C forwarding the acceleration timing to the locomotive gateway.

8.2 Consist Order, Begin WiFi Network

19. Railcars in the departure yard prior to mounting the ETD have all motes in sleep mode
20. The PAN-C in each railcar is waiting for a jolt (accelerometer) and forward motion (gyroscope) to turn on

21. Lead locomotive Head of Train Device (HTD) receives the ETD signal that the train is formed
22. Assume the Wi-Tronix WiPU™ receives this signal from the HTD and forwards the acknowledgement to our WTN Gateway (digital on/off)
23. WTN Gateway is activated by this signal
24. WTN Gateway records the time of activation
25. WTN Gateway begins to send beacons to search for nodes in a WiFi ad hoc network
    a. Initially no railcar motion, so no PAN-C is awake and no response to the beacon.
    b. Other moving railcars in the area are either not connected to a train, so they are in sleep mode, or they are connected to a train, so will not respond to a new gateway beacon.
    c. Need any PAN-C to join a WiFi network only if both a gateway is sending a beacon (meaning the ETD is attached) and the PAN-C feels a jolt and is moving forward. Once joined to a WTN, the PAN-C must ignore other gateway beacon requests.
26. Sometime later, the locomotive begins to leave the departure yard with the train
27. As each railcar detects an acceleration (>1g for ex.), local time is recorded in the PAN-C and the sign of the acceleration is recorded
    a. Clarification: This could use a threshold-supporting accelerometer such as the ADXL345
28. PAN-C sends its railcar ID, the DELTA between the current time of message transmission and the local time the event occurred. This way, no clock synchronization is needed, which would not be present prior to a fully formed network, and acceleration sign to the WTN Gateway
29. PAN-C always relays messages over the WiFi link. WiFi messages are prioritized highest.
30. PAN-C then waits to forward other railcar signals as subsequent cars are activated.
31. Locomotive may stop during departure
    a. PAN-C continues to monitor gyroscope and forward motion. Should forward motion stop due to train stopping in the yard during exiting, the PAN-C resets for the next accel/motion event. A new accel signals a new timing start
    b. If locomotive stops during consist formation in the departure yard, the WTN Gateway resets and awaits new formation.
32. The ZB operation is relatively low-power, start the ZB network immediately after the railcar successfully connected to the locomotive via the WTN. This way, it is available for reporting sensor information even when the train is stopped.
33. WTN Gateway has all railcar accel timing, ID and calculated signal latency from routing information.
34. WTN Gateway calculates the consist order from the timing from locomotive starting motion.

35. WTN Gateway uses the acceleration sign to determine the railcar orientation: (+) is B-end forward, (-) is A-end forward for example
36. WTN Gateway stores information, formats it and sends to the Enterprise through the WiPU

8.3 Forming the ZigBee Networks

37. The PAN-C sends a beacon to the railcar motes that awakens the motes from sleep mode
38. Motes receive the beacon, checks the railcar ID, and if matches, joins the network with their unique ID.
39. Motes had previously connected with their associated sensors for sensor type and location.
40. PAN-C develops a mapping of mote, sensor, location and network ID for later communication with the WTN
41. In order to connect with motes with poor RSSI to the PAN-C, each mote sends a beacon to find unreported motes.
42. If unreported motes are located, those motes report to the PAN-C through appropriate ZigBee motes in a data relay
43. PAN-C uploads its ZigBee network information to the WTN Gateway once the ZigBee network is formed.
44. WTN Gateway maps the network for access by the Enterprise

8.4 Forming the Wi-Fi inter-car train consist network

45. To this point, the Gateway has received the railcar ID's and established the order and orientation of the cars. Also, each ZigBee network has been formed on each railcar.
46. WiFi ad hoc network is formed for the WTN.
47. As each PAN-C completes the ZigBee formation, it begins searching for an ad hoc WiFi network and a path to upload its ZigBee network information to the WTN Gateway.
48. WiFi WTN ad hoc network is formed
49. All railcar information is forwarded to the WTN Gateway
50. WTN Gateway transmits all pertinent information to the Enterprise
    a. Need to define WTN Gateway functionality and interface with WiPU
51. Train leaves the yard with railcar order and orientation information and all intra-car networks formed and the inter-car Wi-Fi network formed
52. All motes enter sleep mode awaiting:
    a. Sensor alert
    b. Network request for status
    c. Network request for stored data
    d. Network download changing alert thresholds
    e. Engineering initiated consist change
53. Cutting and adding railcars
    a. All cut and add activities are with a stopped train.

b. WTN stays active when a train is stopped.
c. WTN closes out when the train is stopped and the ETD is removed.
    i. When ETD is removed, a signal is sent to the HTD that must be forwarded through the WiPU and to the WTN Gateway.
    ii. The WTN Gateway sends a signal (message) to the WiPU to be uploaded to the Enterprise as an official close out of the train network
d. Once the ETD is removed and the WTN is closed, the system waits for a new network formation
e. All motes enter sleep mode
f. When the locomotive is ready to depart with a new consist configuration and the ETD is installed, the network formation process begins again as before.
g. New consist data is uploaded to the WTN Gateway, WiPU and the Enterprise.
h. No active participation of the train engineer or conductor is required.

8.5 Sensors

54. Sensors can be powered and monitored from the mote or self-powered to push data to the mote
55. In normal operation, mote is in low power mode
56. Four potential mote/sensor modes
    a. Self-powered sensors, mote waits for a sensor alert or request from PAN-C
    b. Mote-powered sensors, the mote supplies power but the sensor electronics control uploading alerts to the mote
    c. Mote sensor, the mote supplies power and all controls for the mote.
    d. Mote actuator, the mote powers and interacts with an actuator adaptor to control self-powered actuators
57. Self-powered sensors for TTCI tests
    a. Self-powered sensors have built-in electronics to control the sensor
    b. Hot-Box
        i. Sensor samples temperature every 1 minutes (adjustable sample rate)
        ii. All adjustments are through the mote setup software
        iii. Sensor calculates a moving average over a 10 minute window (adjustable)
        iv. Track the weighted average temperature to avoid reporting on data spikes.
        v. Compare weighted average temperature to the threshold (adjustable threshold)
            1. Record weighted average temperature
            2. If approaches threshold within 10%, send the temp value and alert status 1 to the mote (warning)
            3. If next reading is above the alert 1 level, but alert 1 has already been sent, no action.
            4. If exceeds threshold, send temp and alert 2 to mote (actionable)

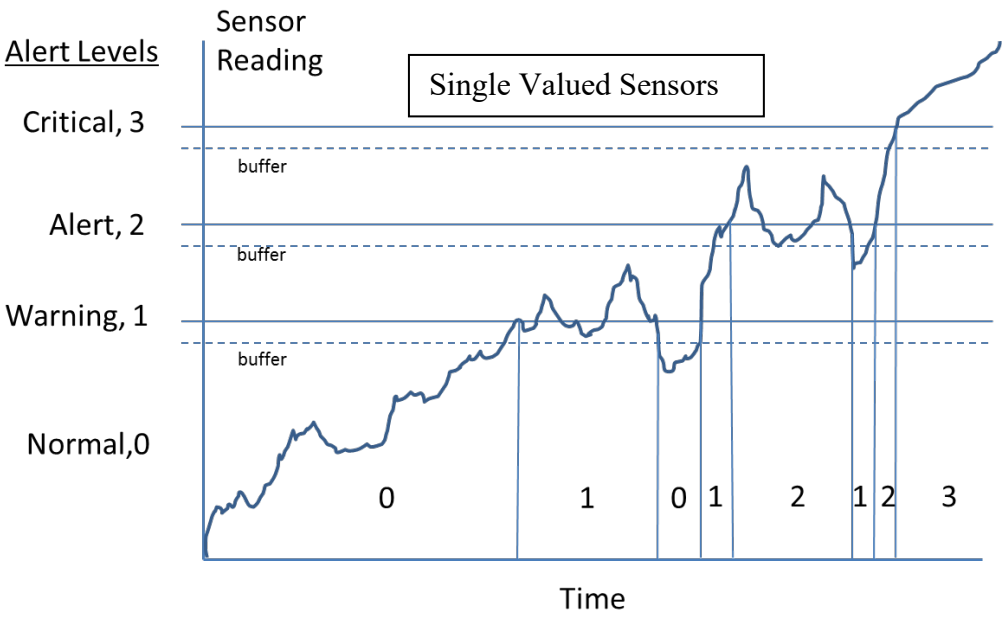
5. If next reading is above the alert 2 level, but alert 2 has already been sent, no action.
6. If exceeds threshold by 10%, send temp and alert 3 to mote (critical)
7. If next reading is above the alert 3 level, but alert 3 has already been sent, no action.
8. If weighted average reduces below the latest alert level, upload an updated temp value and alert level. (for example if the train slows, gives engineer some control)
9. Store weighted average temperature and time stamp for each reading for later download or to upload to the mote on request.
10. At a later date, implement a temperature trending algorithm to be more predictive.

c. Accelerometer (584-ADXL377BCPZ-R7)
   i. Currently selected accelerometer capable of +-200g at from 0.5Hz to 1300Hz.
   ii. Download threshold limits spectrum from the mote on setup
   iii. Frequency range to 1300Hz
   iv. Accelerometer is in sleep mode until activated once the ZigBee network is formed with an activation signal from the mote. ZigBee network is formed only after the railcar is in motion.
   v. Sample data at a factor above max frequency (10X for 13000Hz sample rate)
   vi. Store data in a circular buffer of some duration (10 min)
   vii. Perform real time FFT over a continuously moving 5 min. window
   viii. Compare g-load spectrum to threshold in 1/3rd octave bands
   ix. For current testing, send an alert if threshold is exceeded in any band. (WiPU then interprets the alert signal and communicates to the engineer)
   x. Control of thresholds at setup can limit the bands of interest. May depend on which bands become most sensitive to faults.
   xi. Once an alert is sent, it remains on until thresholds are reduced to some percent below the threshold (20%). Intent is to prevent excessive alert cycling
   xii. A cleared alert must be across all frequency bands. If the band that triggered the alert falls to 80% of the threshold, but other bands exceed the 80% of threshold, the alert stays on.
   xiii. Upon receiving a data dump request from the mote, the sensor will upload the contents of the 10 min circular buffer with a times stamp. If required, data collection can stop in that upload time.

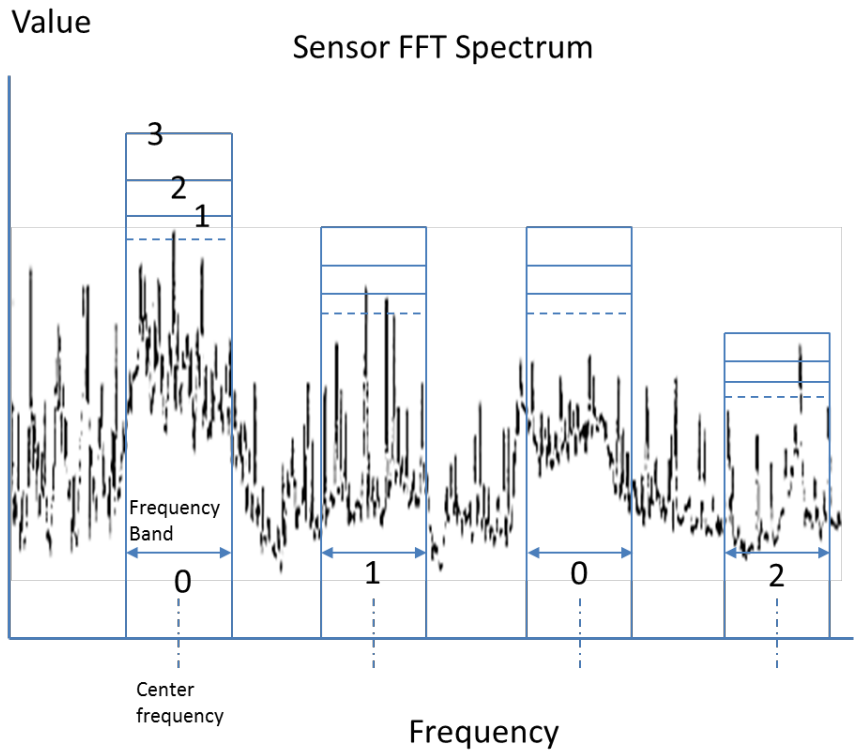
xiv. When railcar stops moving, the accelerometer (and the entire sensor) enters a sleep mode.
xv. Mote sends a signal to wake the sensor either from initial ZigBee network formation, GPS or gyroscope indicated motion begins from the PAN-C, or if mote receives a request for data upload.

d. Microphone (238-WM7121PIMSE/RV)
   i. Mote pushes microphone power spectrum threshold limits to the sensor processor.
   ii. -40C to 100C operating temp
   iii. 62 Hz to 15,000 Hz range
   iv. Microphone sensor is in sleep mode until activated once the ZigBee network is formed with an activation signal from the mote. ZigBee network is formed only after the railcar is in motion.
   v. Sample data at a factor above max frequency (3X for 45,000Hz sample rate)
   vi. Store data in a circular buffer of some duration (10 min)
   vii. Perform real time FFT over a continuously moving 5 min. window
   viii. Compare dB levels to the threshold bands and extract the highest alert level across all bands
   ix. Store the alert levels at each band
   x. Compare alerts to previously stored values. If alert levels are falling, degrade alert level if below the buffer value.
   xi. If a previous alert was sent,
      1. Send no update if status hasn't changed
      2. Send a new alert status if a change
   xii. If no previous alert, send the alert if >0 level, otherwise no action if all bands at 0 alert level

58. Mote functions
   a. Low power mode waiting for a threshold exceedance signal from the sensor
   b. Periodic inquiry of the sensor status and sensor parameter data
   c. Mote sensor alerts should be at 3 levels for all sensor types for consistency



d. For temperature example, [Warning, Alert, Critical, Buffer %] [250F, 205F, 170F, 10%]
e. Alert Value upload to the mote is 0, 1, 2, or 3
f. Mote inputs three alert thresholds for single value sensors that correspond to warning, alert, and critical levels. Also included is a buffer percent or value so that degrading an alert level requires a percentage below the threshold to avoid excessive alert cycling.
g. Thresholds for multivalued sensors such as FFT accelerometer or acoustic values



h. Threshold values are frequency and bandwidth dependent
i. Alert values contain warning, alert and critical values per frequency band
j. Overall sensor alert is the largest monitored value
k. 1 or many bands can be monitored
l. Threshold format is [time window, No. of bands, Band 1 center freq, Band 2 bandwidth, Band 1 Warning, Band 1 Alert, Band 1 Critical, … Band n Critical, buffer percent]
m. This is a lot of input for threshold, but is probably necessary. What is our current limit on #bands?
n. In the example in the figure, alert level 2 is uploaded to the mote.
o. FFT thresholds are evaluated each time window increment. Sensor remembers the band and alert level and lowers the level in each band if values fall below the buffer level. Then sensor reports the highest of all bands in that time window.

8.6 Hot Box Setup

59. Setup program pushes temperature threshold temperature as in the figure , sample rate (1 sample/min) and moving average window (10 min.) to the sensor on railcar installation
60. Wheel location is saved on the mote (L1, R4, etc)
61. Wake up with temperature threshold exceedance signal. (Do we need a wake-up signal 1st, followed by the temp and alert status?)
62. Receive the temp and alert status from the sensor, immediately forward to the PAN-C for distribution to the WTN Gateway mote.

63. Wait a period of time (how long?) for receipt acknowledgement from the WTN Gateway, if no receipt, resend message to the PAN-C.
64. Mote returns to sleep mode until next alert
65. In this case, control of alerts is controlled by the sensor since it is self-powered with onboard controller.
66. Mote receives a request for raw data from the Enterprise through the WiPU, WTN Gateway and PAN-C. (What is that signal?). Mote requests moving average temp and time stamp data from sensor. Receives the sensor upload, and forwards back through the PAN-C, WTN Gateway and WiPU to be uploaded to the Enterprise.

8.7 Accelerometer Setup

67. Mote pushes threshold spectrum limits to the sensor on setup
68. In the TTCI test case, the mote must push temperature, acceleration spectrum and sound dB spectrum threshold limits according to the figure for spectrum thresholds.
69. Once ZigBee network is formed (car is moving at that point), the mote sends an activation signal to the sensor to wake up. Accelerometer begins taking data.
70. If railcar (and train) stops temporarily, the PAN-C sends a signal to the mote that the railcar is in a stop mode
71. In a stop mode, the mote signals the sensors to stop collecting data and enter a sleep mode
72. Once the railcar begins moving, the PAN-C signals the mote to reactivate and the mote passes the signal to the sensors and data collection begins again.
73. Mote stays on low-power mode, waiting for a signal from the sensors to upload alerts, waiting for a change in status from the PAN-C (stopped train) or waiting for a signal from the PAN-C to upload a particular sensor's data.
74. When the mote receives an acceleration alert, it 1st receives a notice of pending alert, allowing time to wake, and then it receives the alert (value of 0, 1, 2, or 3).
75. Microphone – behaves the same as the accelerometer sensor.
76. Both accelerometer and microphone are examples of a spectrum sensor and the mote interface should be the same for both.

8.8 Actuator Functions

77. Motes function to activate a device rather than actually power them.
78. Motes can supply 3.3V power. Supply 4mA or 20mA max.
79. Motes can provide an actuation signal only, or supply the signal with power
80. Devices can communicate status back through the WTN to the engineer through the WiPU
81. Mote is configured at setup with the device type, and location
82. Mote and actuator are in sleep mode and are activated only when needed
83. In general, a device can be activated through an action by the engineer or at specific instances of train configuration. In both cases, the request for an action on the device must

come through the WiPU to the WTN Gateway, to the PAN-C and forwarded to the mote for action.

84. Once the mote receives the request for an action on the device, the mote wakes and sends a signal to the device to await an action request
85. Device actuators generally require
    a. Action signal (set brake): digital pulse for setting action, repeat send after a time period (3min?) if status is not returned as actuated.
    b. Reverse action (release brake): digital pulse to release brake, signal repeated at a time period (3 min?) if status does not return released
    c. Status return signal from device actuator: actuated (set)=1, reversed action (released) = 2, unknown status =3
        i. Clarification: We need to determine if this will be a logic signal or an analog signal. Logic signal per signal line will give us only a single bit, unless we establish and define framing for it (for example, a preamble sequence followed by a specific number of bits). An analog signal could represent different values as voltage ranges. For example: reversed action==(v<0.2V), actuated==(v>3.1V), error or unknown==(1V<v<2V).
    d. Separate signal from the device actuator is a fault code limited in some way per the mote requirements. Perhaps a 3 digit number?
        i. Clarification: Same limitation as above.
        ii. Typical sequence
            1. Engineer requests hand brake on either a particular car or on all cars
            2. WiPU forwards the request to the WTN Gateway
            3. WTN Gateway cross references addressed to forward the request to the cars PAN-C
            4. PAN-C forwards the request to the appropriate mote. Need a railcar location nomenclature for all potential locations (wheels, doors, brakes, etc)
            5. Mote receives the request and wakes up from sleep or low power mode
            6. Mote sends a signal to the actuator to wake and await a signal
            7. Mote sends a “set brake “ signal to the actuator
            8. EDHB brake actuates and starts to set the brake
            9. Brake is set after 2 minutes
            10. A “set” status = 1 is sent to the mote and the mote stops activate signal re-sends
            11. Mote forwards status to the WTN Gateway and the WiPU to inform the engineer the device is set
            12. Actuator and mote return to a low power or sleep mode

13. Engineer requests brake release and the actions above repeat.
14. Mote sends the “release” request, received by the actuator and actuator sends a “release” status = 2 to the mote when released
15. If the status is not changed after a time period (3 min?), the release request is resent.
16. Errors in the actuator a passed to the mote via a fault code and forwarded to the WTN Gateway and the WiPU

8.9 Wi-Tronix WiPU Interface – TTCI Testing Only

86. Train Motion Procedures
87. Locomotive is running and connected to the train
88. Identify the controlling locomotive
89. End of Train Device (EOT) installed
90. EOT ID entered into HOTD and connection is confirmed
91. Successful brake line pressure test completed
92. Engine run switch set to run, needed for locomotive to move
93. Throttle moves to position 1,2 or 3 depending on the grade
94. Slow acceleration until the slack is removed
95. EOT reports forward motion to HOTD
96. More rapid acceleration to get to train speed
    a. Controlling Locomotive WiPU Interface
97. Install WTN Gateway (GW) nearby for Ethernet connection to WiPU
98. WiPU sends signal to WTN GW that it is controlling locomotive
99. WTN GW requires that signal to initiate WiFi beacon transmission
100. WiPU sends WTN GW signal to initiate the network once the EOT has connected to the HOTD and the break pressure has been checked and confirmed
101. WTN GW sends the WiFi beacon and waits for railcars and other power units to send ID and delta times from car acceleration start and receipt of the beacon.
102. Train begins to move
103. WTN GW accumulates inputs from network, calculates the train order and car orientation
104. EOT sends a signal to the HOTD that it is moving
105. WiPU sends signal to the WTN GW that all cars are moving and to stop waiting for car timing signals
106. WTN GW uploads the consist information to the WiPU for upload to the TTCI Enterprise – data format TBD
    a. Initiate the Test network
107. ZB network IDs uploaded to the WTN GW and stored
108. WTN GW has initial threshold alert values stored from installation
109. TTCI Enterprise uploads new thresholds to WiPU.

110. WiPU forwards thresholds for temperature, accelerometer and acoustic monitors to WTN GW
111. New thresholds are forwarded to sensors as soon as arrived at the WTN GW
112. Data format TBD
113. Test train run continues
114. Sensor Alerts
    a. When a sensor alert is developed, the alert goes through the PAN C to the WTN GW and then to the WiPU
    b. Alert consists of a railcar ID, sensor type, sensor location and alert level
    c. WiPU sends the alert to the TTCI Test Enterprise
115. Actuator
    a. TTCI Enterprise sends a request for application of the EDHB to the WiPU
    b. WiPU checks that there is no motion (and other checks as required)
    c. WiPU forwards the actuation request to the WTN GW
    d. WTN GW sends a signal to activate all EDHB devices
    e. WTN GW cross-indexes the locations of EDHB devices and forwards the signal.
    f. PAN-C sends signal to actuators
    g. Actuator mote sends signal to actuator
    h. Actuator returns the status signal
    i. Mote forwards status signal to WTN GW and WiPU
    j. WiPU uploads actuator status to TTCI Enterprise
    k. Fault signals from actuator are also forwarded as needed.
    l. TTCI Enterprise requests EDHB to be turned off.
    m. WiPU sends request to WTN GW
    n. WTN GW responds with status and/or fault code
    o. WiPU forwards status and fault code to TTCI Enterprise