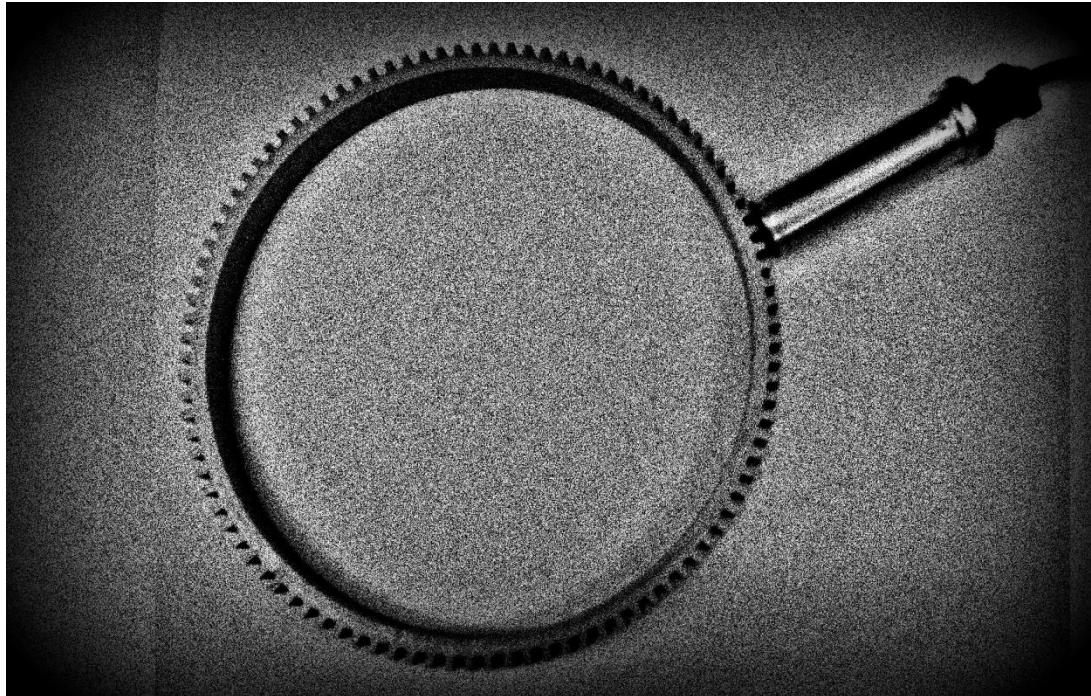


# CHALMERS



## Energy harvesting Wheel Speed Sensor

*Master of Science Thesis*

DHASARATHY PARTHASARATHY

*Department of Microtechnology and Nanoscience*

CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden, 2012





**CHALMERS**

**VOLVO**

Energy harvesting Wheel Speed Sensor

DHASARATHY PARTHASARTHY

Department of Microtechnology and Nanoscience  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Göteborg, Sweden 2012

**Energy harvesting Wheel Speed Sensor**  
**DHASARATHY PARTHASARATHY**

© DHASARATHY PARTHASARATHY, 2012

Department of Microtechnology and Nanoscience,  
Chalmers University of Technology,  
SE-412 96 Göteborg  
Sweden  
Telephone +46 (0)31-772 1000

# Energy harvesting Wheel Speed Sensor

DHASARATHY PARTHASARATHY

Department of Microtechnology and Nanoscience,

Chalmers University of Technology

## **Abstract:**

This thesis presents a prototype energy harvesting autonomous sensor, called the Autonomous Wheel Speed Sensor (AWSS), that is targeted for operation in the Electronic Braking System (EBS) of vehicles. In order to monitor the rotational state of a wheel, the EBS currently uses a passive Wheel Speed Sensor (WSS) which is a variable reluctance electromagnetic transducer. In the existing EBS setup, one WSS is used per wheel, each of which is connected to the EBS system using extensive cabling.

This project presents the first successful attempt at converting the WSS into an *energy harvesting wireless wheel speed sensor* or in other words an *Autonomous Wheel Speed Sensor* (AWSS), which provides information about the rotational state of the wheel to the EBS system over a wireless link. Unlike most wireless sensors which use batteries, the AWSS employs energy harvesting to power itself by simultaneously using the WSS electromagnetic transducer as an energy harvester as well as a sensor. By thus making an autonomous WSS, the amount of cables in the existing WSS assembly can be reduced, leading to savings in material, assembly and maintenance costs.

The self-powered prototype AWSS successfully implements periodic wireless transmission of the wheel speed along with near real-time wheel lock detection, at a duty cycle of less than 5.9%. The AWSS has been built using readily available COTS components and uses a proprietary low-power standard for wireless communication.

The prototype AWSS implemented in this project successfully demonstrates that the WSS is capable of being used as an energy harvesting transducer. In the experimental setup used in this project, the WSS yields harvestable power of  $11.4 \mu W - 1 mW$  for speed ranges of  $\sim 5 - 55 \text{ kmph}$ . This opens up the possibility of using the variable reluctance sensor setup to harvest energy from any rotational assembly and use this harvested energy to power autonomous sensors.

This prototype system is intended for operation in AB Volvo vehicle applications and the project is a partnership between Chalmers University of Technology and Volvo Group Trucks Technology – Advanced Technology and Research, Göteborg, Sweden.

Keywords: energy harvesting sensor, wheel speed sensor, variable reluctance sensor

## Acknowledgements

It is not often that one is given five months of time, an Aladdin's cave of a lab, the indulgence and guidance to build something as fun as an autonomous sensor right up from scratch. Sure enough, it's been the most amazing five month period of learning, and now that the project is done, I turn my attention to the most important part: a shout-out of 'tack så mycket!' to everybody involved with the project.

I begin with thanking my thesis examiner Per Lundgren at Chalmers. Thanks Per, for the opportunity, the advice and the tough reviews which made sure that my attention to detail wasn't wavering.

I would then like to thank Alejandro Cortes at Volvo Technology (VTEC) for giving me this incredible opportunity to work with the state of the art of sensor technology.

I then thank Roy Johansson at VTEC for his invaluable guidance. Thanks Roy, first for thinking up this project, then for all the early morning chats at 0730 where we always put my design through scrutiny and most important of all, for helping me (an automotive noob) by answering my never ending questions about trucks, though I think you've to work more on your fake Aussie accent!

I save my biggest thanks for my supervisor Mikaela Öhman at VTEC. I couldn't thank you enough, Mikaela, for giving me this opportunity along a simple mandate to knock myself out! Thanks for being a great mentor and for your confidence in letting me freely frame the contents of my thesis.

## Abbreviations:

ABS	Anti-lock Braking System
ALL	Application Logic Layer
AP	Access Point (of a sensor network)
API	Application Programming Interface
ASR	Anti-slip Regulation
AWSS	Autonomous Wheel Speed Sensor
AWSSN	Autonomous Wheel Speed Sensor Network
DAQ	Data Acquisition
DSP	Digital Signal Processing/Processor
EBS	Electronic Braking System
ECU	Electronic Control Unit
ED	End Device (of a sensor network)
EHU	Energy Harvesting Unit
EMC	Electromagnetic Compatibility
ez430	TI ez430-RF2500 development tool
FSM	Finite State Machine
GSM	Global System for Mobile communications
HAL	Hardware Abstraction Layer
IDE	Integrated Development Environment
ISR	Interrupt Service Routine
JTAG	Joint Test Action Group
LAN	Local Area Network
LPM	Low Power Mode (MSP430 operational mode)
LPM	Low Power Mode
MCU	Microcontroller Unit
MPPT	Maximum Power Point Tracking
PAN	Personal Area Network
PHY	Physical layer of a communication protocol stack
PM	Pseudo Modulator
PP	Peak to Peak
RMS	Root Mean Square
RPM	Revolutions Per Minute
RT	Random Telegraph process
RTOS	Real-Time Operating System
SoC	System on Chip
SPI	Serial Peripheral Interface bus
TDMA	Time Division Multiple Access
TI	Texas Instruments Inc.
TPMS	Tire Pressure Monitoring System
UART	Universal Asynchronous Receiver/Transmitter
WER	Wheel Emulation Rig
WISCON	Wireless Sensor Concept Node
VLSI	Very Large Scale Integration
VR	Variable Reluctance
WSN	Wireless Sensor Network
WSS	Wheel Speed Sensor
WSU	Wireless Sensing Unit

## **Contents**

<i>Abstract:</i> .....	4
<i>Acknowledgements</i> .....	5
<i>Abbreviations:</i> .....	6
<i>Contents</i> .....	7
<i>List of Figures:</i> .....	9
<i>List of Tables:</i> .....	11
<b>Chapter 1 - Introduction</b> .....	12
1.1 Summary of key contributions of this project:.....	12
1.2 Background of this project: .....	13
1.3 A guide to the contents of this report: .....	13
<b>Chapter 2 - Autonomous sensors in vehicles</b> .....	14
2.1 Autonomous sensors:.....	14
2.2 The Wireless Sensor Network (WSN) paradigm: .....	14
2.3 Energy autonomy using Energy Harvesting:.....	15
2.4 Autonomous Wireless Sensor Networks (WSN) in vehicles: .....	16
<b>Chapter 3 - The Wheel Speed Sensor (WSS) and its applications</b> .....	18
3.1 The Wheel Speed Sensor (WSS): .....	18
3.2 The WSS as an energy source: .....	21
3.3 Primary application of the WSS - The Electronic Braking System (EBS):.....	23
3.4 The role of the WSS in the EBS: .....	23
3.5 WSS installation as part of the EBS: .....	24
3.6 Other Variable Reluctance (VR) sensor applications in a vehicle:.....	25
<b>Chapter 4 - The Autonomous Wheel Speed Sensor (AWSS)</b> .....	27
4.1 The functional architecture of an AWSS node: .....	27
4.2 The Autonomous Wheel Speed Sensor Network (AWSSN): .....	28
4.3 Challenges in implementing the AWSSN:.....	28
4.4 Potential benefits:.....	30
<b>Chapter 5 - Prototype specification</b> .....	32
5.1 Prototype network architecture:.....	32
5.2 Prototype functionality:.....	33
5.3 Parametric model:.....	33
5.4 General design principles for increasing availability of autonomous sensors: .....	44

<b>Chapter 6 - Prototype development setup .....</b>	<b>47</b>
6.1 The Wheel Emulation Rig (WER):.....	47
6.2 Hardware choices for the ED and AP:.....	48
6.3 Putting it all together - the hardware development bench:.....	51
6.4 Software development: .....	52
6.5 System modeling:.....	52
<b>Chapter 7 - Prototype implementation.....</b>	<b>53</b>
7.1 Modular stack definition: .....	53
7.2 Hardware Layer: .....	53
7.3 The software layers: .....	55
7.4 The AWSSN cluster library: .....	59
<b>Chapter 8 - Prototype characterization.....</b>	<b>60</b>
8.1 Characterizing the Wheel Emulation Rig (WER):.....	60
8.2 Characterizing the ED/AWSS node:.....	66
8.3 Illustrative end-to-end application scenario:.....	79
<b>Chapter 9 – Conclusions .....</b>	<b>81</b>
9.1 Summary of results: .....	81
9.2 Suggestions for further development.....	82
<b>Bibliography .....</b>	<b>84</b>
<b>Appendix A - The Random Telegraph (RT) process.....</b>	<b>88</b>
<b>Appendix B - Hardware Abstraction Layer (HAL) user/programmer guide .....</b>	<b>90</b>
<b>Appendix C - Application Logic Layer (ALL) programmer/user guide.....</b>	<b>92</b>
<b>Appendix D - AWSSN Cluster definition .....</b>	<b>94</b>
<b>Appendix E - Parametric register .....</b>	<b>95</b>
<b>Appendix F – Prototype characterization test cases.....</b>	<b>98</b>
<b>Appendix G – Confidential information.....</b>	<b>101</b>

## List of Figures:

Figure 1 : A WSN mesh network [9] .....	15
Figure 2 : Architecture of an energy harvesting wireless sensor.....	16
Figure 3: An illustration of WSS assembly near the wheel of a truck [15].....	18
Figure 4 : The WABCO WSS and a depiction of its internal structure .....	19
Figure 5 : The WSS along with a pole wheel .....	19
Figure 6 : Power and efficiency of an electrical circuit as a function of the load [16] .....	21
Figure 7 : Knorr-Bremse EBS used in some Volvo trucks [27] .....	23
Figure 8 : WSS in the EBS setup [27].....	24
Figure 9 : WSS mounting on the front axle of Volvo D68 - with deflector and protection shield [15]......	25
Figure 10 : WSS harness mounting procedure for the front axle [29].....	25
Figure 11 : A camshaft (left) and a crankshaft (right).....	26
Figure 12 : The Autonomous Wheel Speed Sensor (AWSS) .....	27
Figure 13 : The Autonomous Wheel Speed Sensor Network (AWSSN).....	28
Figure 14: Conceptual depiction of the targeted prototype - a two node AWSSN.....	32
Figure 15 : Targeted prototype AWSSN setup .....	32
Figure 16 : The 'leaky bucket' model of the energy harvesting ED node .....	36
Figure 17 : Classification of WSU functionality.....	39
Figure 18 : Design principles for increasing availability of autonomous sensing .....	46
Figure 19 : Experimental setup for the WSS - (a) the WER and (b) & (c) a closer look at the pole wheel and VR sensor setup.....	47
Figure 20 : CBC-Eval-09 functional description due to [44].....	48
Figure 21 : SimpliciTI communication stack [49] .....	50
Figure 22 : Topologies supported by SimpliciTI [49].....	50
Figure 23 : Prototype hardware development and test bench .....	51
Figure 24 : Modular definition of an AWSS node.....	53
Figure 25 : Hardware Layer of the ED node .....	54
Figure 26 : AWSSN program structure.....	55
Figure 27 : State machine implementation of ED ALL .....	57
Figure 28 : Flow diagram depicting AP ALL logic .....	59
Figure 29 : WER transfer characteristics - signal voltage .....	61
Figure 30 : WER transfer characteristics - signal frequency .....	62
Figure 31 : WSS COMSOL 2D model.....	64
Figure 32 : Simulated result of the magnetic flux density.....	65
Figure 33 : WSS COMSOL model simulated results simulation results .....	66
Figure 34: WSS signal distortion due to loading .....	66
Figure 35 : WSS signal pulse conversion – high frequency case .....	67
Figure 36 : WSS signal pulse conversion - low frequency case.....	67
Figure 37 : WSS unlock event - high speed.....	68
Figure 38 : WSS lock event - high speed braking .....	69
Figure 39 : WSS lock event - low speed braking.....	69
Figure 40 : ED startup with average current draw of <b>6.59 mA</b> and discharge capacity of <b>0.03 µAh</b> .....	70
Figure 41 : ED event sensing and transmission.....	71
Figure 42 : Wheel unlock event transmission with average current draw of <b>7.85 mA</b> and a discharge capacity of <b>0.01µAh</b> .....	71
Figure 43: ED attribute sensing and transmission .....	72
Figure 44 : Attribute sensing and transmission in a high speed case with an average current of <b>3.78 mA</b> and a discharge capacity of <b>0.01 µAh</b> .....	73
Figure 45 : Attribute sensing and transmission in a low speed case with an average current of <b>2.41 mA</b> and a discharge capacity of <b>0.06 µAh</b> .....	73
Figure 46 : Variation of $\lambda_{transition}$ with $t$ for $\lambda_{transition} = 0.3$ .....	74
Figure 47 : Plot of duty cycle as a function of attribute transmission interval with $\lambda_{transition} = 0.15$ .....	75

<i>Figure 48: Effect of event transitions on the duty cycle of operations.....</i>	76
<i>Figure 49 : Average discharge due to the ED as a function of attribute transmission interval.....</i>	76
<i>Figure 50 : Variation of average power consumption with attribute transmission interval .....</i>	77
<i>Figure 51 : ED cold start example.....</i>	78
<i>Figure 52 : AWSSN end-to-end illustration .....</i>	80
<i>Figure 53 : RT process for <math>\mathbf{X}(\mathbf{0}) = \mathbf{1}</math> .....</i>	88
<i>Figure 54 : RT process for <math>\mathbf{X}(\mathbf{0}) = \mathbf{0}</math> .....</i>	89

## List of Tables:

<i>Table 1 : Wireless networks for automotive applications due to [13] .....</i>	17
<i>Table 2 : WSS functional parameters.....</i>	34
<i>Table 3: EHU parametric specification .....</i>	37
<i>Table 4: WSS pulse converter truth table.....</i>	40
<i>Table 5 : WSU wheel speed attribute sensing specification.....</i>	40
<i>Table 6 : WSS rotational state sensor truth table .....</i>	41
<i>Table 7 : WSU event sensing specification .....</i>	41
<i>Table 8 : WSU RF communication specification .....</i>	41
<i>Table 9 : WSU duty cycle specification.....</i>	42
<i>Table 10 : ED power consumption parameters.....</i>	43
<i>Table 11 : EHU parameters as implemented by the CBC-Eval-09.....</i>	49
<i>Table 12 : WSU wheel speed attribute sensing parameters as implemented by ez430 (MSP430F2274).....</i>	49
<i>Table 13 : ED parameters implemented by the Hardware Layer .....</i>	54
<i>Table 14 : ED parameters implemented by the HAL .....</i>	56
<i>Table 15 : ED parameters implemented by the ALL.....</i>	58
<i>Table 16 : WSS parameters as implemented by the WER .....</i>	63
<i>Table 17 : WSS COMSOL model parameters .....</i>	65
<i>Table 18 : Experimental measurement of Hardware Layer parameters .....</i>	70
<i>Table 19 : Experimental measurement of event sensing parameters .....</i>	72
<i>Table 20 : Experimental measurement of attribute sensing parameters .....</i>	74

# Chapter 1 - Introduction

*"In the 1980s, the PC revolution put computing at our fingertips.*

*In the 1990s, the Internet revolution connected us to an information web that spans the planet.*

*And now the next revolution is connecting the Internet back to the physical world we live in-in effect, giving that world its first electronic nervous system.*

*Call it the Sensor Revolution: an outpouring of devices that monitor our surroundings in ways we could barely imagine a few years ago. Some of it is already here. The rest is coming soon."*

- US National Science Foundation 'The Sensor Revolution - A special report' [1].

Sensors and sensor networks are heralding an age of astounding technological possibilities, glimpses of which are already being seen. Machines equipped with sensors that are completely in tune with the ambient environment are fast breaking out of the realm of science fiction and becoming commonplace. This race for a technological fantasy world depends upon the mastering of a wide range of technologies, among which ranks the technology of Energy harvesting. This technology, which imparts intelligent embedded sensor devices the ability to scavenge energy from the ambient environment, gives a special meaning to the term 'ambient intelligence'. Energy harvesting, along with other key enabling technologies such as microelectronics, nanotechnology, signal processing and wireless communications, have already set the pace for rapid developments in this ongoing sensor revolution. The automotive electronics scene has also been an active participant in the sensor revolution, where new sensing technologies are continuously redefining the ways and means of controlling transportation. Energy harvesting sensors have now gained traction in the automotive industry as it has been identified as a technology that has real potential to broaden horizons.

This chapter presents a summary of conclusions and contributions of this thesis project in section 1.1 and a guide to the contents of the report in section 1.2.

## **1.1 Summary of key contributions of this project:**

The first contribution of this project is proving the concept that the Wheel Speed Sensor (WSS) can be used as an energy harvesting transducer to power autonomous sensors. In the experimental setup used in this project, the WSS has been measured to produce harvestable power in the range of  $11.4 \mu W$  –  $1 mW$  for a wheel speed range of  $5.8$  –  $54.6 kmph$ . These power levels have been shown to sustain carefully engineered wireless sensor operations.

The next contribution is the development of a prototype Autonomous WSS (AWSS) which uses the power harvested from the WSS, in addition to simultaneously using the WSS as a sensor. The prototype AWSS is tested to be capable of comfortably transmitting wheel speed information at intervals of  $\geq 1 s$  while consuming less than  $0.65 mW$  (under certain modeling assumptions) of power on average with a duty cycle of less than 5.9% for a wheel that keeps rotating for ~80% of the time. It is also capable of communicating wheel start/stop events with a worst case delay of  $100 ms$ , a time interval which can be easily reduced in future iterations.

Based on the design of the prototype AWSS, a set of general design principles have been derived, which would help in engineering energy harvesting sensors with real-time capabilities. These design

principles, which aim at increasing the availability of the autonomous sensor using highly abstracted passive stimulation, is the next contribution of this project.

This project also recommends the serious consideration of the Variable Reluctance sensing assembly as a platform for harvesting energy from any rotating assembly. This harvested energy can be used to power sensors nearby. For example, by adapting the pole-wheel and VR sensor combination to fit inside the tire, enough energy can be harvested to power the Tire Pressure Monitoring System (TPMS).

The entire project was conducted with an eye on the future and the prototype was designed and developed in such a way that it lays down the framework for further development. A basic mathematical model of the autonomous sensor has been developed, with an emphasis on incorporating stochastic models. A modular stack definition for operations in each sensor node has also been developed along with software platforms for each layer of the stack.

## **1.2 Background of this project:**

This thesis project is a spinoff of the Wireless Sensor Concept Node (WISCON) project that is jointly being undertaken by Volvo Technology Corporation, Chalmers University of Technology and Halmstad University. The aim of WISCON is the survey of technology and development of prototypes, in order to build a knowledge base for energy harvesting in automotive applications. The project is in its second year of activity and this author, as part of WISCON, has participated in compilation of surveys of energy harvesting technologies and applications [2], and energy harvesting power electronics [3]. These technology surveys form the technical basis of this thesis project.

## **1.3 A guide to the contents of this report:**

The idea of autonomous sensors and their potential application in automotive control systems is presented in Chapter 2. The Wheel Speed Sensor (WSS) which is targeted for rendering autonomy, its capabilities as a sensor and as an energy harvester and its application in the Electronic Braking System (EBS) of a truck among others, are presented in Chapter 3. The idea of an Autonomous Wheel Speed Sensor (AWSS), which combines the usage of the WSS as a signal and energy source, its implementation challenges and benefits are presented in Chapter 4. A minimal behavioral/mathematical model of the AWSS system that is geared towards the prototype implementation is presented in Chapter 5 along with system level functionality implemented in the prototype. The lab development setup is presented in Chapter 6 which describes the hardware and wireless communication standard chosen for implementing the prototype. The modular stack definition of an AWSS node along with a description of the implementation of each layer of the stack is presented in Chapter 7. The characterization of the prototype system and the results of characterization are presented in Chapter 8. Finally, a summary of results and suggestions for enhancements is presented in Chapter 9.

## **Chapter 2 - Autonomous sensors in vehicles**

Sensors are pervasive in today's vehicles providing data for performance, safety, and convenience and comfort functions. The data provided by these sensors are made available to one of the many electronic control units (ECU) in the vehicle, which process this data and perform control actions. The sensors used in vehicles are of incredible variety ranging from the Oxygen sensor for regulating particulate matter in the exhaust, the Manifold Air Pressure (MAP) sensor to measure the mass air flow rate in the engine to decide fuel injection, right up to climate sensors for regulating the climate in the cab. Both the number and the variety of sensors used in a vehicle have been continuously increasing, a trend which is set to continue in the future. The main drivers behind such ever increasing sensor deployments in vehicles are increased operational efficiency on one hand and regulation/legislation on the other hand which mandate the deployment of sensors. The usage of sensors in vehicles is expected to expand over the years with the increase in 'X-by-wire', the electrical/electronic control of mechanical systems. A combination these and many more factors have made the automotive sensors market, according to one market research report [4], a \$14.5billion industry in 2011 and projected to increase to \$20billion in 2016. The average number of sensors in a vehicle, according to a review paper submitted in 2008 [5], is 40 in the North American market which is projected to increase to 70 in 2013.

This chapter presents the idea of autonomous sensors in section 2.1, the Wireless Sensor Networks (WSN) paradigm that is used to realize this idea in 2.2. The well-established technology for providing energy autonomy to these autonomous sensors is presented in 2.3. Finally the extent of autonomous sensors in vehicles is presented in 2.4.

### **2.1 Autonomous sensors:**

While sensor applications have been steadily increasing in vehicles, the world of sensors itself has been experiencing exciting developments due to breakthroughs in the areas microelectronics, micro- and nanotechnology. The integration of solid-state sensors with associated electronics and embedded networking services had given rise to autonomous sensing devices, i.e. independent embedded systems communicating data and events for machine consumption. The idea of autonomous sensors has been around for quite some time and gained traction with the definition of the Sensor Web by Delin et al. [6] at NASA's Jet Propulsion Laboratory (JPL). Paraphrasing from their definition, an autonomous sensor has at the least:

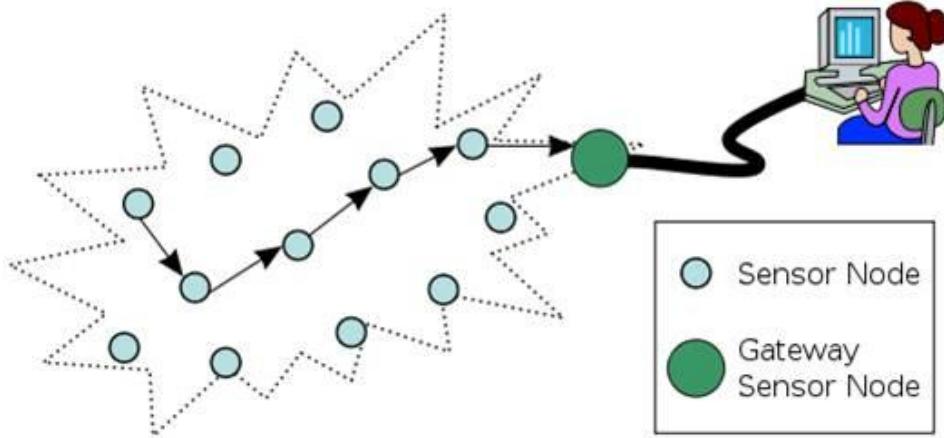
1. Processing autonomy - the sensor has its own CPU in the form of a Microcontroller/ASIC/FPGA
2. Communication autonomy - the sensor has means of communicating wirelessly
3. Energy autonomy - the sensor has means of powering itself

In very simple terms, autonomous sensors are smart embedded devices, which are not wired to anything else, and have capabilities to operate independently. Such autonomous devices can be widely distributed spatially and deployed in locations that are inaccessible.

### **2.2 The Wireless Sensor Network (WSN) paradigm:**

The idea of networked sensors is an inevitable consequence of spatial distribution of autonomous sensors. This idea was born quite early with the first sensor network going operational in the mid-1970s. The history of sensor networks is quite fascinating, beginning with the Sound Surveillance System (SOSUS) deployed by the US Navy on the Atlantic seabed to detect Soviet submarines [7].

Academic interest in sensor networks began with the definition of the paradigm known as 'Smart dust' in a research proposal submitted by the University of California, Berkeley (UCB) in 1997 [8]. The term Sensor Network or Wireless Sensor Network (WSN) is commonplace today to describe this paradigm of spatially distributed networked embedded sensor devices, each of which is called as a sensor node or a mote. Nodes in a sensor network communicate by transmitting a message to another node in its neighborhood, which in turn passes on the message towards the destination.



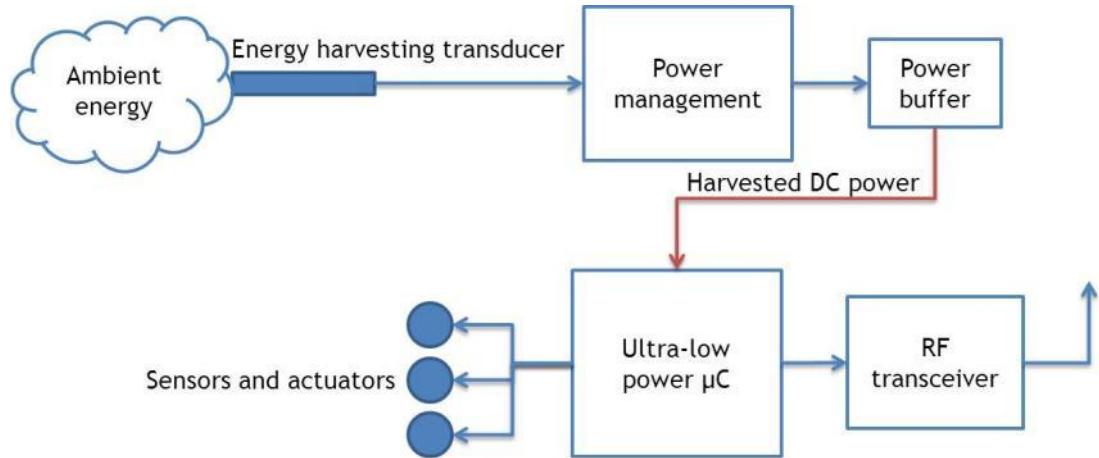
**Figure 1 : A WSN mesh network [9]**

WSN are capable of deploying sensors covering huge geographical areas, reaching inaccessible places thereby providing remote sensing capabilities of far-reaching proportions. On the other hand it is also suitable for deployment in small physical areas such as a house or a room for Information Technology (IT)-based utilization of resources in the target environment. WSN represents the ultimate deployment model for sensors where each sensor is tasked with data collection and reporting in a completely autonomous manner. WSN communication protocols available today, such as Zigbee and Bluetooth, are highly capable and accomplish functionality specified in all the seven layers of the ISO Open Systems Interconnection (OSI) standard.

### 2.3 Energy autonomy using Energy Harvesting:

For too long a time batteries have been (and still are) at the forefront of providing energy autonomy. Frequent battery replacements and environmentally safe disposal concerns, however, have led to the evolution of techniques for recycling/scavenging or harvesting energy from the ambient environment, an approach that is known as energy harvesting.

Energy harvesting (or ambient energy harvesting to be more accurate), refers to scavenging small amounts of power from the ambient environment to power autonomous sensors. Harvested power levels of  $100 \mu W - 100 mW$  are typical, since CMOS VLSI based ultra-low power systems can implement a wide range of functions using such small amounts of power. In order to achieve complete autonomy using energy harvesting, a wireless sensor node usually has the following architecture:



*Figure 2 : Architecture of an energy harvesting wireless sensor*

Ambient sources of energy include ambient vibrations, temperature gradients and solar energy, among many more. In most cases, a dedicated energy harvesting transducer is utilized to convert a chosen form of ambient energy into raw electrical power. A comprehensive study of energy harvesting research and development can be found, from among a plethora of sources, in [10] and [11]. Another major area of energy harvesting research is power management whose role is to convert the raw harvested electrical energy into usable DC power. Power electronics for energy harvesting is quite a mature area with many COTS offerings, while simultaneously being a field of active research, whose state of the art can be gleaned from [12]. With energy harvesting and wireless communication, a sensor node taking the above architectural form becomes autonomous.

As mentioned in section 1.2, a survey of energy of energy harvesting technologies and energy harvesting power management have been compiled as part of WISCON in [2] and [3] respectively.

#### **2.4 Autonomous Wireless Sensor Networks (WSN) in vehicles:**

Electronic Control Units (ECUs) in vehicles have paid most attention to processing autonomy, with wired networked embedded devices being the order of the day. In a typical architecture of automotive control systems, ECUs with autonomous processing capabilities are placed on communication buses such as Controller Area Network (CAN) or Local Interconnect Network (LIN) buses. While at a system level the control units are more or less independent of each other, at each control unit the functionality is quite centralized. Also, since sensors are connected to individual ECUs, the idea of 'Sensing as a service' and networked sensors is still at its infancy. The 'universe' of a sensor in a vehicle currently extends only as much as the ECU it serves.

Communication autonomy is also in its infancy, reflected by the fact that the number of wireless sensors in a vehicle is currently very low. The newly introduced TPMS is often the only wireless sensor in a vehicle. This trend is set to reverse as communication autonomy has the potential to introduce significant cost savings by the removal of electrical cables, which are often several kilometers in aggregated length, along with savings in the associated assembly effort. In addition, new wireless sensors could be deployed in areas which were impossible to reach using cables, while they can also be used for applications which are too trivial to justify wiring costs.

The idea of wireless sensors and WSNs in vehicles does provide tantalizing prospects and has already been the focus of significant amount of research. In a review by D'Orazio et al. [13] an extensive classification of in-vehicle communication networks based on application and data rate requirements is

provided. According to this review, current wireless technologies that are suitable for automotive applications are:

<b>Wireless networking standard</b>	<b>Representative data rate</b>	<b>Applications</b>
Ultra-Wide Band (UWB)	100Mbps	Infotainment, body electronics
WLAN/Wi-Fi	1Mbps	Infotainment
Bluetooth	500Kbps	Infotainment
Zigbee	150Kbps	Body electronics

*Table 1 : Wireless networks for automotive applications due to [13]*

From this classification, it can be seen that wireless sensors and networks are currently being considered only for infotainment applications, where wireless LAN technologies are becoming popular. PAN technologies are being advocated for infotainment and body electronics applications. However according to this review, wireless sensors are not considered for safety and 'X-by-wire' applications.

Energy autonomy for automotive sensors is slightly more mature compared to communication autonomy. While the automotive industry has pioneered technologies which recycle wasted energy, for example - waste heat recovery from vehicle exhaust using thermoelectric materials, ambient energy harvesting for automotive sensors has only begun to start getting attention. The automobile is quite rich in ambient sources of energy such as ambient vibrations, heat, solar and electromagnetic energy. A review of possible energy harvesting sources is presented in [14] and the list of possible sources is definitely not limited to this.

This project aims to develop a prototype for proving the concept of using autonomous sensors in an automotive safety application. The prototype autonomous sensor would include provision autonomy in all three areas of processing, power and communication.

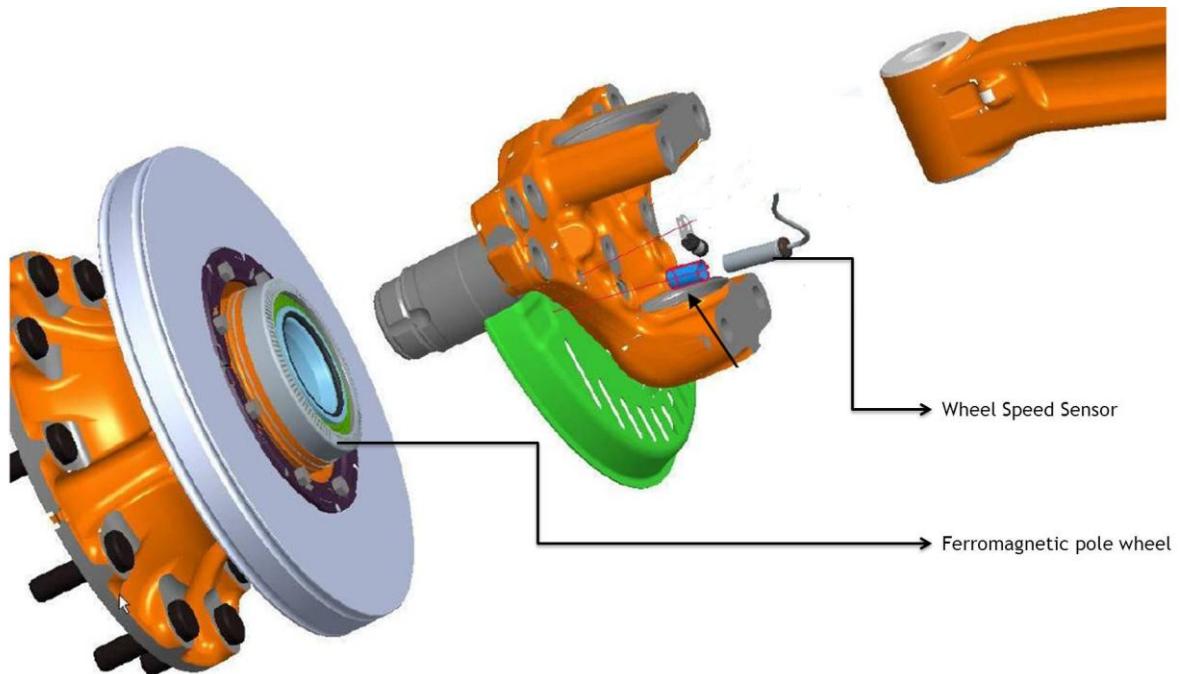
## Chapter 3 - The Wheel Speed Sensor (WSS) and its applications

The Wheel Speed Sensor (WSS) is the automotive sensor that has been chosen to be made autonomous in this project. The WSS transducer has certain capabilities, which are suitable for using it simultaneously as a sensor and an energy harvester.

This chapter introduces the WSS in section 3.1 and the possibility of using the WSS as an energy harvester along with a review of comparable harvesting techniques is presented in section 3.2. The primary application of the WSS in trucks is presented in sections 3.3, 3.4 and 3.5 while other applications are presented in section 3.6.

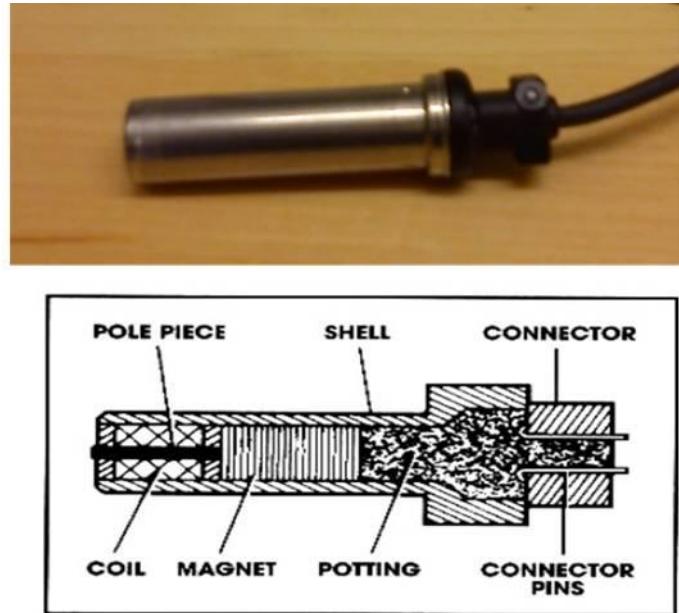
### 3.1 The Wheel Speed Sensor (WSS):

The WSS is used to sense the state of rotation of a wheel in a contactless manner, using a magnetic sensor and a pole wheel that rotates along with the vehicle wheel. A sinusoidal signal produced by the WSS is a direct measure of the rotational speed of the wheel. In a vehicle, one WSS is provided for each wheel in a deployment that is illustrated by the following figure:



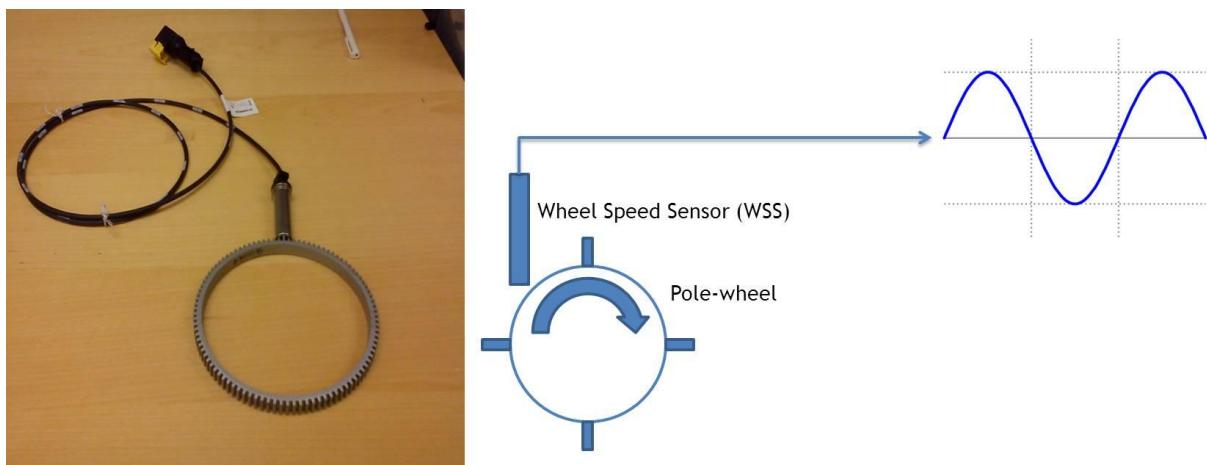
*Figure 3: An illustration of WSS assembly near the wheel of a truck [15]*

The WSS falls under the category of Variable Reluctance (VR) sensors, a class of electromagnetic sensors that is popular for sensing displacement. A typical VR sensor consists of a permanent magnet to which a ferromagnetic core (or pole piece) is attached, with a coil wound around the core. A WSS procured from WABCO (<http://www.wabco-auto.com/>) is used in the project, whose conceptual description can be seen below:



*Figure 4 : The WABCO WSS and a depiction of its internal structure*

The permanent magnet acts as a source of magnetic flux, which when interrupted, induces a voltage in the coil. The sensor, sometimes called magnetic pickup, is therefore placed near a flux path whose reluctance (resistance to the flow of magnetic flux) varies in proportion to the feature that is being sensed. In this case, the variable flux path is provided with a gear wheel also called the pole-wheel, tooth-wheel or tone-wheel. The pole wheel, in the case of wheel speed measurement, is mounted on the axle of the wheel and therefore has the same RPM speed as the wheel. The WSS along with the pole wheel it is used with for RPM measurements is shown below:



*Figure 5 : The WSS along with a pole wheel*

The principle of operation of the WSS is to vary the reluctance of the flux path, depending upon speed of rotation, which in turn varies the flux that is linked by the coil. This induces a voltage proportional to the reluctance of the flux path, in accordance with the Maxwell-Faraday law. The flux of the permanent magnet placed near the wheel is variably linked by the teeth in the case of radial pole wheel and by the gaps in case of the axial pole wheel. The induced voltage is therefore proportional to the rate of change of flux as well as the distance between the VR sensor and the pole wheel.

### **3.1.1 A note on WSS terminology:**

The terminology for referencing the WSS setup and its constituent parts are not usually consistent and therefore the terminology used in this report will be clarified here.

1. VR sensor/VR magnetic sensor/VR magnetic pickup/Magnetic pickup - this refers to the magnetic element shown in Figure 4.
2. WSS - this term will be used to describe the complete assembly of the pole wheel and the VR magnetic sensor. However in certain places WSS will also refer to the VR sensor, a reflection of the common usage of the term WSS to refer to the magnetic sensor alone.

### **3.1.2 WSS signal voltage:**

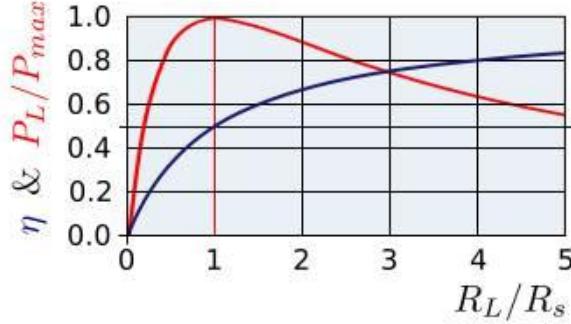
While the WSS operation is based on the Maxwell-Faraday law, a mathematical model for the output signal voltage of the WABCO WSS has not been developed because of the reluctance of the manufacturer to share the architectural details of the sensor in the duration of the project. Therefore in this project, empirical data is used to assess the signal voltage of the WSS.

The speed of rotation of the pole-wheel primarily determines the amplitude of the WSS output signal voltage. Apart from the speed of rotation of the wheel, the WSS output signal amplitude also depends upon a number of parameters such as intensity of the magnetic field, number of turns in the induction coil around the magnet, magnetic properties of the pole wheel, the air gap between the pole wheel and the sensor, etc. Though the output voltage depends upon such an extensive list of parameters, most of them are typically set during installation and not controllable later. For most practical purposes the voltage depends only upon the speed of rotation of the pole-wheel.

The main feature of interest from a sensing point of view is the frequency of the induced voltage which is proportional to the wheel rotation speed. While it is the frequency and not the amplitude of the induced voltage that is of primary interest for sensing purposes, it is advantageous (for reasons described in section 3.2.2) to tweak the assembly to get as strong a signal as possible. An illustration of the signal voltage from sample measurements provided by WABCO is shown in Appendix G. The figure presents output signal amplitude as a function of the air gap for different WSS. For an 100 Hz output it can be seen that the signal voltage has peak-to-peak amplitude of about 3 V. This data is simply an illustration because, as explained earlier, the signal characteristics depend upon the assembly. A detailed presentation of the WSS signal characteristics along with empirical data for the experimental assembly used in this project is presented in section 8.1.

### **3.1.3 WSS signal power:**

The WSS has constant, mostly resistive (in the sub-kHz range), source impedance ( $R_s$ ) which, if exploited properly, makes it possible to draw significant amounts of power from it. The maximum power transfer theorem in circuit theory states that maximum power is drawn from a circuit if the load impedance matches the source impedance. However, the disadvantage of impedance matching is that the efficiency of power extraction is 50%. This scenario can be represented by the graph below:



*Figure 6 : Power and efficiency of an electrical circuit as a function of the load [16]*

Under conditions of maximum power transfer, the WSS average signal power is given by:

$$P_{WSS,max} = \frac{V_{RMS}^2}{4R_s} \quad (I)$$

Here  $V_{RMS}$  is the average signal voltage under a certain speed of rotation of the pole wheel. Assuming a source resistance of about  $1\text{ k}\Omega$ , for the voltage produced as shown in Appendix G, it is possible to extract about  $0.28\text{ mW}$  from the WSS which may be sufficient for powering certain applications. Here again this data is only an illustration of maximum power extraction and the empirical power data measured on the experimental setup used in this project is presented in section 8.1.

### 3.2 The WSS as an energy source:

While the application of the WSS as a signal source, i.e. a sensor, is fairly common, the main research angle of this project is determining the capabilities of the WSS as a power source. Electromagnetic generators based on the Maxwell-Faraday law are widely used for generating electrical power. This principle which is applied in the AC motor is also equally applicable to electromagnetic meso/micro-energy harvesters of various types, which generate power using induction.

It has been established in section 3.1.1 that the WSS is just another electromagnetic generator with a permanent magnet acting as the flux source and the pole wheel varying the reluctance. An illustrative case of the power that is available for extraction from this signal is provided in section 3.1.3. The primary source of this extractable energy is the kinetic energy of the rotating wheel and therefore the WSS is an electromagnetic energy harvester extracting energy from ambient rotational kinetic energy. This combination of sensing and power harvesting capabilities creates a unique convergence, which opens up the possibility of using the WSS both as the signal source as well as the energy source.

In order to place the WSS in the context of electromagnetic energy harvesters a review of research in the area of electromagnetic energy harvesting and rotational kinetic harvesting is presented below.

#### 3.2.1 A review of electromagnetic and rotational kinetic energy harvesters:

The WSS uses ambient rotational kinetic energy as its energy source and generates power by electromagnetic transduction. This represents the convergence of the areas of electromagnetic energy harvesting and rotational energy harvesting.

Electromagnetic transducers for ambient kinetic energy harvesting have been widely discussed in literature. In most of these cases, the ambient kinetic energy is in the form of linear acceleration in one to three dimensions. A comprehensive review of electromagnetic transducers for ambient vibration

harvesting has been presented in [17]. The authors conclude from their survey that in its current state of research, electromagnetic vibration harvesters are in the range of  $0.1 - 100 \text{ cm}^3$  in size and  $100 \text{ nW} - 100 \text{ mW}$  in output power. Most of the harvesters use the resonance phenomenon and produce highest amount of energy in a narrow range of mechanical vibrations.

When it comes to rotational kinetic energy harvesting a variety of designs has been reported in literature. Among the piezoelectric transduction techniques, Manla et al. [18] have reported a non-resonant piezoelectric transduction mechanism where the centripetal force due to the rotation of a wheel is used to make a ball bearing impact a piezoelectric material. The authors have reported measured power of  $4 \text{ mW}$  at a rotational speed of  $800 \text{ RPM}$ . Hu et al. [19] have proposed flexible piezo generators to harvest energy from the deformation of tires during rotation, meant for powering a Tire Pressure Monitoring System (TPMS) node. Khameneifar et al. [20] have proposed a piezoelectric cantilever with a tip mass that is attached to a rotating hub with a simulated output power of  $7.7 \text{ mW}$ . A similar structure using a magneto-electric transducer which is a combined magnetostrictive/piezoelectric laminate, producing  $157.4 \mu\text{W}$  has been reported by Wen et al. [21]. Gu and Livermore [22] have reported a radially oriented self-tuning piezoelectric cantilever beam.

Rotational harvesting designs involving electromagnetic transduction of rotation include Conrad [23] who has reported pendulum based designs for harvesting power from both linear and rotational inertia of a ship's propeller. The reported device is designed to be embedded in the propeller and is reported to produce an output power of  $10 - 100 \text{ mW}$  in the rotational speed range of  $100 - 350 \text{ RPM}$ . Toh et al. [24] have reported a modification of a conventional generator by attaching an unbalanced mass to the rotor of the DC generator. The design reportedly produces harvested power up to  $1 \text{ W}$  at a relatively high rotational speed of  $8000 \text{ RPM}$ . Wang et al. [25] have reported a well-weighted pendulum, i.e. a pendulum with one or more weights which adjusts the oscillation frequency to meet the rotational frequency and uses electromagnetic transduction to convert this oscillation to electrical power. The authors have explored a deployment in the wheel of a vehicle in order to power a TPMS node and have reported measured power in the range of  $0.9 - 2.6 \text{ mW}$  depending upon the speed of rotation.

### **3.2.2 The WSS as an energy harvester:**

The WSS, like many of the reviewed examples, uses the stator-rotor combination for inertial harvesting. The meso-scale dimensions of the WSS is quite advantageous because, as shown in [26], scaling laws for electromagnetic transduction is such that higher open circuit voltages can be produced by meso scale wire wound coils. The WSS is easily capable of producing open circuit voltages, as will be shown in section 8.1, in the  $1 - 10 \text{ V}$  range. This relatively high range of open circuit voltage is its biggest advantage because with a source impedance of  $1 \text{ k}\Omega$  at maximum power transfer, this open circuit voltage provides a power of  $0.13 - 12.5 \text{ mW}$ . This is comparable to the range of extractable power reported in literature.

Structurally, the WSS system needs an elaborate assembly to fix the pole wheel to the axle and hold the VR sensor close to the pole wheel. The structural assembly of the WSS in a truck is presented in section 3.5 while the experimental WSS setup is presented in section 6.1. The elaborate assembly procedure makes the WSS less integrated as a structure when compared to the reviewed harvesters. However the advantage of using the WSS as an energy harvester is its wide acceptance in rotation sensing applications. The pole-wheel and VR sensor assembly can be used in any wheel-axle assembly such as those found in engines, generators and motors and similar machinery. The meso-scale WSS setup however restricts its use to meso-scale machinery, where size is not a concern. The Variable

Reluctance (VR) transduction technology is quite mature and can be used directly, without any modifications, as an energy harvester.

### 3.3 Primary application of the WSS - The Electronic Braking System (EBS):

The EBS acts as the platform for providing electronic control of braking services in vehicles. The primary function of EBS is the electronic actuation of pneumatic systems to engage brakes in vehicles. In addition to the core braking function, the EBS continuously monitors the condition of the brakes in order to detect malfunctions. The EBS platform also integrates a number of associated functionality such as Anti-lock Braking System (ABS), Traction Control System (TCS), Electronic Stability Control (ESC), etc. The figure below depicts an EBS that is installed in some Volvo trucks.

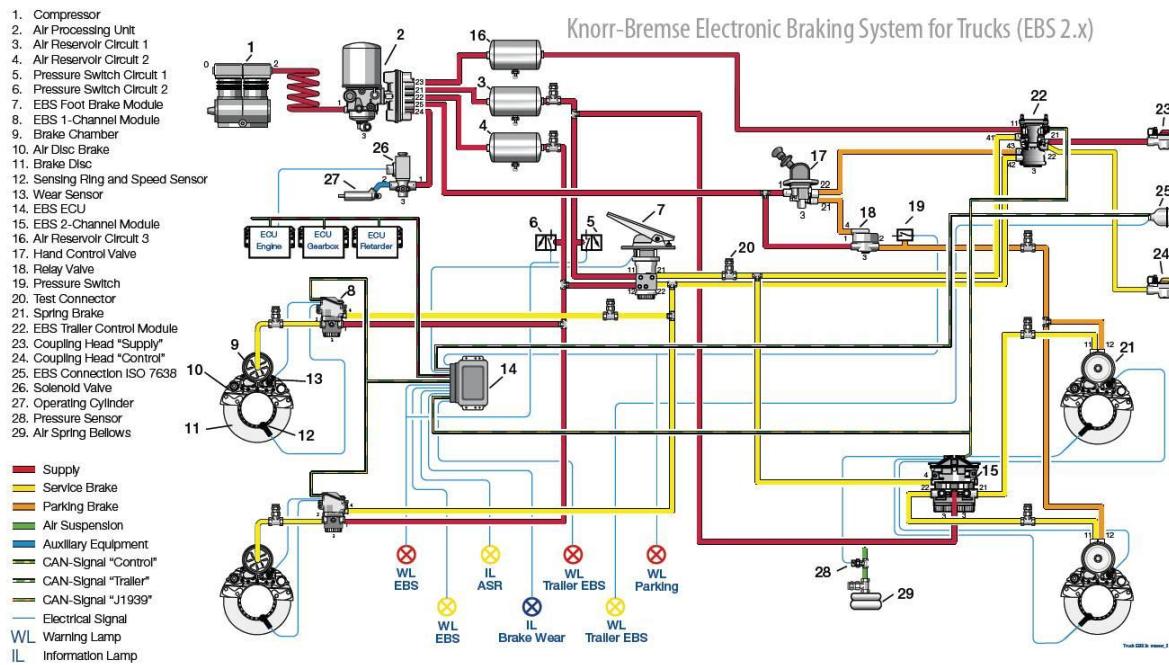


Figure 7 : Knorr-Bremse EBS used in some Volvo trucks [27]

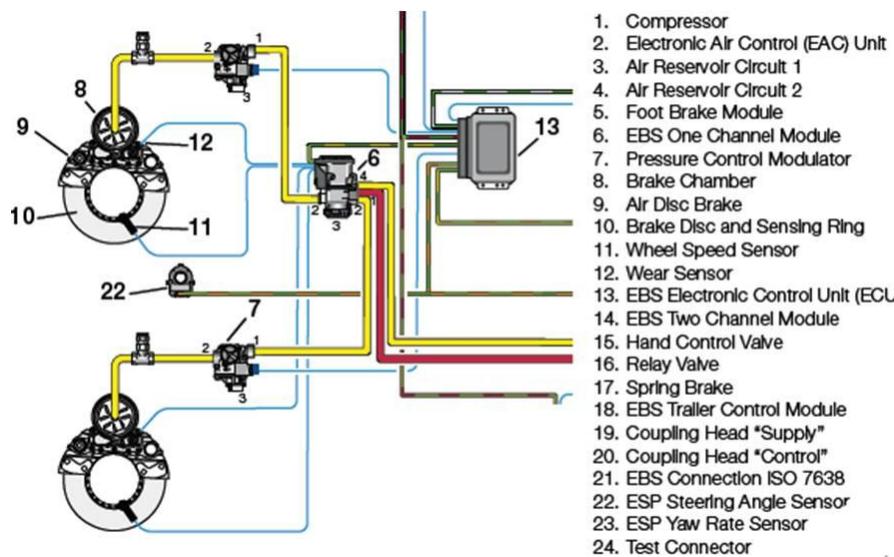
### 3.4 The role of the WSS in the EBS:

The WSS provides real-time information about the rotational speed of the wheel to which it is connected, and therefore acts as a primary signal source for the EBS Channel Module (otherwise known as the EBS Modulator). The Channel Module processes the WSS signal and produces the instantaneous wheel speed data which is made use of for the following control functions:

1. **Vehicle speed indicator** – The speed data that is displayed on the dashboard is derived from the WSS signal
2. **Gear selection** - In the I-Shift intelligent gear shift system found in Volvo trucks [28] uses the instantaneous vehicle speed along with other characteristic data to decide the gear that has to be engaged.
3. **Anti-lock Braking System (ABS)** – Whenever brakes are applied the WSS signal is used to detect if the wheel is rotating/unlocked or not rotating/locked. If a wheel is locked while the brake is applied, there are chances that the wheel is sliding instead of rotating which causes maneuvering problems. Detecting and controlling undesirable wheel locking falls under the purview of the ABS.

4. **Anti-slip Regulation (ASR)/Traction control** – This is a secondary function of the ABS system where the intention is to prevent loss of traction. Possible differences in traction, i.e. the grip of the wheel on the road, between different wheels are derived from the instantaneous wheel speeds (available from the WSS). Corrective action is taken by a combination of throttling and applying brakes on selected wheels to equalize traction.
5. **Electronic Stability Control (ESC)** – When a vehicle is in operation there is a possibility of loss of steering control. This happens where the actual direction of the vehicle does not match the intended direction of motion set using the steering wheel, which is possible during evasive swerves, under steering or over steering. In such cases, corrective action is taken to restore stability by applying brakes to counter-steer. The actual direction of the vehicle is calculated using measured lateral acceleration, the vehicle yaw and the wheel speeds (using the WSS).

The instantaneous wheel speed is quite an important piece of information and therefore its usage may not be limited to the above list of applications.



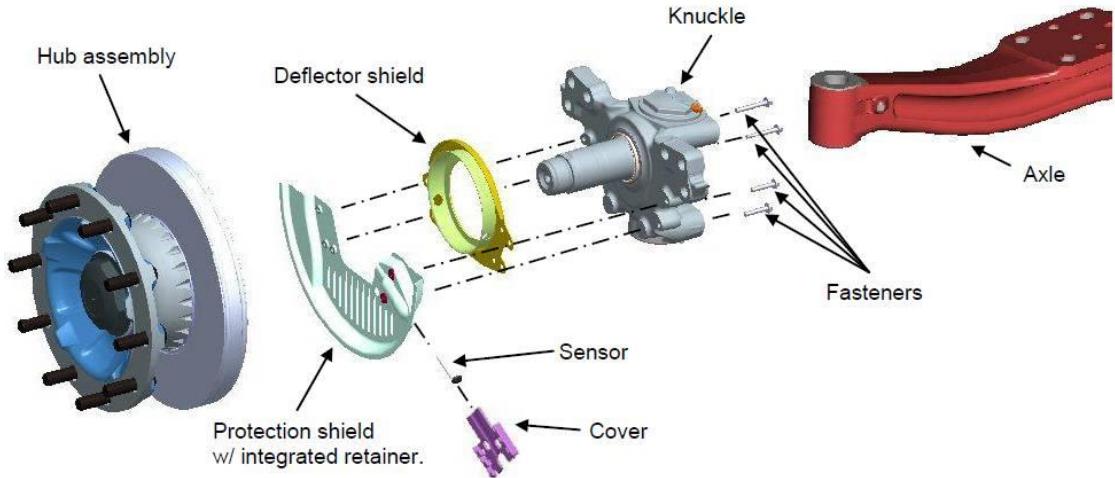
*Figure 8 : WSS in the EBS setup [27]*

### 3.5 WSS installation as part of the EBS:

WSS installation in Volvo trucks can be broadly divided into two phases which are

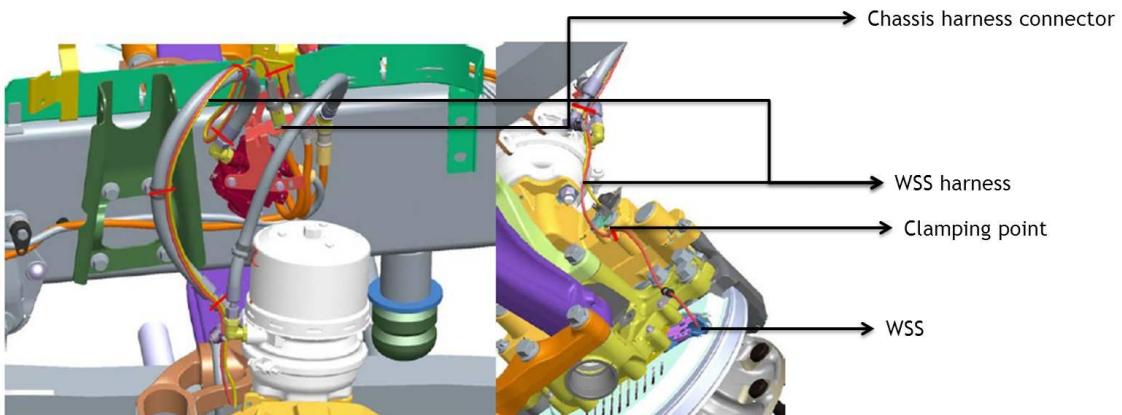
1. WSS mounting on axle - the installation of the sensor close to the wheel on the axle which is usually done by the axle manufacturer
2. WSS harness mounting - the installation of cables connecting the sensor mounted on the axle to the EBS modulator which is done by the vehicle manufacturer

The WSS installation on an axle is an elaborate process which varies depending upon the type of the axle and the type of the brake used. A detailed presentation of WSS installation in different axles can be found in the Volvo assembly document [15], out of which the mounting process for a front wheel with a disc brake is presented here for illustrative purposes. The WSS assembly on the front axle of a Volvo D68 truck is shown below.



**Figure 9 : WSS mounting on the front axle of Volvo D68 - with deflector and protection shield [15]**

Once the WSS is mounted on the axle, the next assembly step is to mount the harness connecting the WSS to the EBS modulator. The harness mounting procedure, just like the WSS mounting on the axle, depends upon the axle. Harness mounting procedures for different axles are described in the Volvo installation requirements document [29]. The assembly of the harness typically involves connecting the routing cables, by securely fastening them using clamps at different points, to ensure that the electrical connection is not degraded or lost.

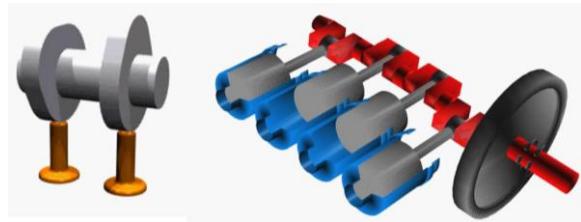


**Figure 10 : WSS harness mounting procedure for the front axle [29]**

The figure above illustrates the harness clamping process for connecting the WSS to the chassis harness connector. The red loops in the picture indicate clamping points, one of which is marked, which are affixed manually costing assembly effort. The total cost for the harness material and assembly, as shown in Appendix G is about 5.3 EUR. A cost analysis of the WSS axle assembly has not been presented because it is the harness assembly that stands to be replaced as explained further on.

### **3.6 Other Variable Reluctance (VR) sensor applications in a vehicle:**

The same pole wheel and VR sensor setup used in the WSS is reused for similar applications in the vehicle. For example, it is used for a number of position and speed sensing operations in the engine such as the crankshaft position sensor and the camshaft position sensor. An illustration of the camshaft and the crankshaft is shown below.



*Figure 11 : A camshaft (left) and a crankshaft (right)*

By using VR sensors, the positions of the camshaft and crankshaft can be sensed which is then used to determine the cylinder position and the engine stroke which in turn plays an important role in fuel injection. This sensing mechanism can be extrapolated and the VR sensor can be used for position/speed sensing in any rotational frame of reference.

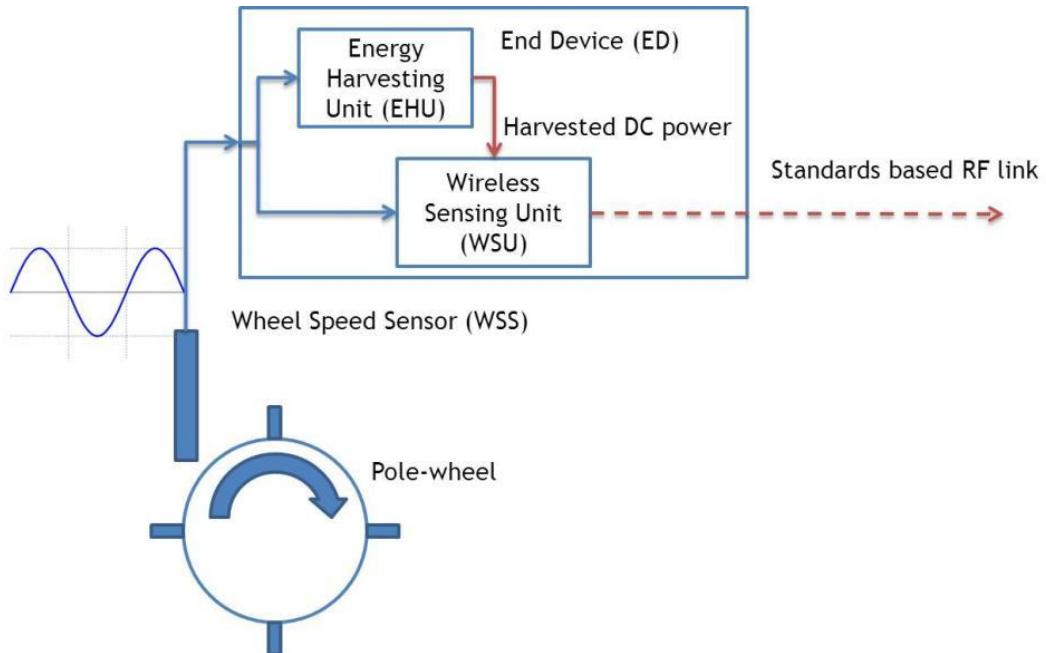
## Chapter 4 - The Autonomous Wheel Speed Sensor (AWSS)

The harness assembly cost for the WSS, shown in section 3.5, in combination with the energy harvesting capabilities of the WSS, shown in section 3.2, makes the WSS a reasonable target for rendering autonomy. The synergy of sensing and energy harvesting capabilities of the WSS sensor to create an Autonomous WSS (AWSS) is a notion that is quite attractive.

This chapter presents the functional architecture of the proposed AWSS node and its subsequent integration in a sensor network in sections 4.1 and 4.2. The challenges of implementing such a system and the potential benefits offered by the system are presented in sections 4.3 and 4.4.

### 4.1 The functional architecture of an AWSS node:

Synergizing the sensing and energy harvesting capabilities of the VR magnetic sensor lies at the heart of converting the passive WSS into an autonomous WSS, and the architecture of the AWSS node reflects this idea as shown below.



*Figure 12 : The Autonomous Wheel Speed Sensor (AWSS)*

By integrating the WSS with an energy harvesting mechanism and a wireless sensing platform, which includes a microcontroller and RF transceiver, the new embedded device called the AWSS is created. Because of the wireless communication of the AWSS, the harness assembly of the passive WSS stands to be eliminated. A brief description of the constituent parts of the AWSS follows.

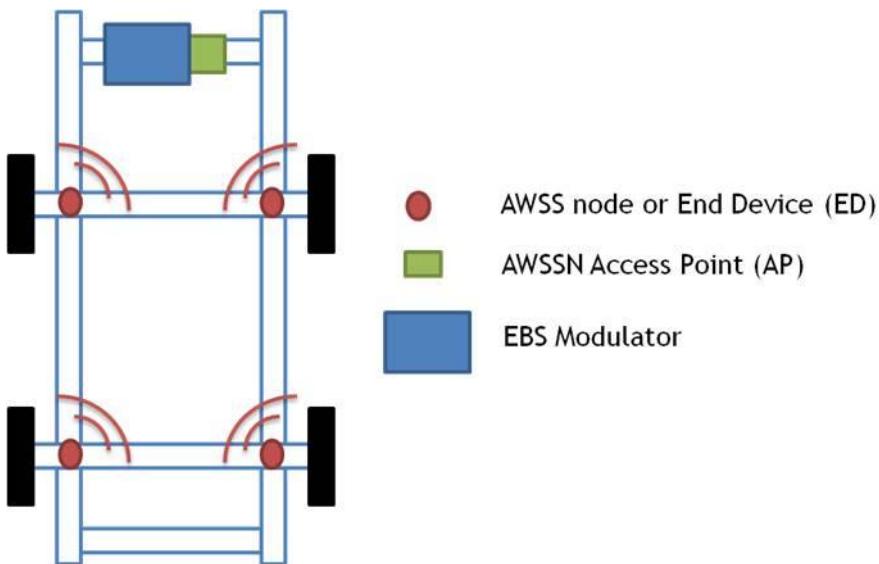
**The Energy Harvesting Unit (EHU):** The EHU would be responsible for converting the raw sinusoidal signal output of the WSS into usable DC power. As pointed out in section 2.3 the research area of energy harvesting power management is purely dedicated towards accomplishing this process chain efficiently. The EHU in combination with the VR sensor act as the power source for the AWSS and therefore provides power autonomy.

**The Wireless Sensing Unit (WSU):** The WSU is responsible for implementing the node level management, sensing and messaging operations of the AWSS. It is therefore responsible for providing

the AWSS with processing and communication autonomy. Powered by the EHU, the WSU senses and processes the WSS signal output. Based on the processed results it makes determinations and communicates them, some in real time. The WSU must perform all these operations, while consuming the least amount of power possible. This precludes the usage of ultra-low power hardware processing platforms.

#### **4.2 The Autonomous Wheel Speed Sensor Network (AWSSN):**

Reflecting the WSS deployment, one AWSS node needs to be deployed per wheel, and the data from each AWSS has to be collected and provided to the EBS system. In order to integrate each individual AWSS into the EBS, a WSN infrastructure is required. A WSN would take care of setting up and managing the network of AWSS nodes while seamlessly routing data between them and the EBS. A potential AWSSN could take the following form where one AWSS node or an End Device (ED) is attached to every wheel in the vehicle and all of them communicate with an Access Point (AP).



*Figure 13 : The Autonomous Wheel Speed Sensor Network (AWSSN)*

The ED, which is the AWSS node, has already been described, while the AP is a typical WSN sink node, which acts as a bridge between the EBS and the WSN. It routes the data between each ED and the EBS while in addition creating and maintaining the network of autonomous nodes. While there is no question that such a WSN would be quite a challenge to implement, the technology for such a network is already available and ripe for prototyping.

#### **4.3 Challenges in implementing the AWSSN:**

The application of autonomous sensors, which use energy harvesting and wireless communications, in a critical real-time control system like the EBS raise a lot of concerns. These concerns, along with a few more challenges, which have to be overcome to realize the network of autonomous sensors, are as follows.

- 1. Availability concerns:** The biggest concern of an autonomous sensor is whether it is available at a certain point of time to perform an operation. Here the term ‘availability’ is defined, by the field of reliability engineering, as the proportion of time for which a system is functioning. If a system experiences a failure, which renders it incapable of performing an operation, it is in a state of non-availability. Availability is heavily used for evaluating the performance of

equipment in the telecom and IT industries. In energy harvesting and in wireless sensors, ‘high availability’ is a derivative definition where, for example, efficiently harvesting power and judicious usage of this harvested power ultimately contributes towards high availability. High availability of autonomous sensors is a definition that is commonly used in this report.

One way of increasing availability is to increase the amount of energy that is being harvested. Increasing the efficiency of transduction and power extraction is hotly being pursued as [10], [11], [12] and [30] would show. Another approach is the hybrid energy harvesting technique where multiple ambient sources of energy are tapped to increase the harvested energy levels. Examples include the combination of kinetic and RF energy harvesting using a single coil proposed in [31] and the hybrid ambient light and thermal harvester proposed in [32].

Another way of increasing availability is judicious usage of the harvested power, i.e. to reduce the duty cycle of operation. A set of design principles have been proposed in section 5.4, which calls for abstracted passive sensing to reduce duty cycle and power consumption. Ways of improving availability are not limited to the options described here and there is little doubt that as the technologies related to autonomous sensors mature, the availability levels would improve.

2. ***Real-time task execution concerns:*** As pointed out in [13], wireless sensors for critical automotive applications in the areas of safety and X-by-wire are currently not being seriously considered due to concerns ranging from latency to interference. In addition to being wireless, an autonomous node is also powered by energy harvesting which only adds up to the real-time execution concerns.

Real-time scheduling in energy harvesting sensors is in a state of infancy and a sample of the research activity in this area is presented here. Scheduling mechanisms for recurrent tasks have been proposed by Audet et al. [33] and lazy scheduling of tasks with non-deterministic arrival times proposed by Moser et al. in [34]. Gu and He propose a mechanism for bounding delays in communication in [35] while Liu et al. propose the adaptation of the well-known Dynamic Voltage and Frequency Scaling (DVFS) technique for energy harvesting sensors in [36] as do Dehghan and Kargahi in [37].

Though it is currently not a widespread application area of WSN, real time wireless sensing and control is certainly not new. Implementation of real-time scheduling for wireless control and sensing, especially in the industrial process control area, is actively being researched. The WirelessHART open communication standard seems to be a popular choice in this area as reported in [38], [39] and [40]. WirelessHART uses a TDMA based transmission mechanism, more akin to voice communication standards like GSM, which makes real-time operations possible. Though WirelessHART is not used in this project, the intention here is to illustrate that real-time wireless sensing and control is definitely achievable.

3. ***Integration concerns:*** A significant area of concern is the integration of the newly added electronics to the WSS assembly. A suitable mounting point for the autonomous sensor and suitable packaging must be chosen to integrate the CMOS electronics so that the node electronics is able to withstand the rigorous environment close to the wheel. The mounting point, in many ways, decides whether this application is feasible or not since the new assembly could turn out to be more elaborate than the existing assembly.

4. **New assembly procedures:** Procedures have to be developed to program and assemble these new ECUs in the vehicle. This would present a number of challenges, for example, a particular AWSS node has to be coupled both physically and in program with a particular wheel so that the modulator can recognize the exact wheel to which a particular message applies. This is not a major technical challenge but it does require intensive data collection and management for maintaining node level information.
5. **EMC concerns:** With the introduction of new wireless sensors, extensive evaluation needs to be done to ensure that the addition of this sensor does not affect any of the electronics in the neighborhood. The radio emissions must be balanced with signal power requirements of the wireless communication standard being used.

#### **4.4 Potential benefits:**

While the development of the AWSSN presents many challenges, there are potential gains from such an application which are listed below.

1. **Assembly cost savings:** The AWSSN could potentially eliminate all cable and connector costs listed in Appendix G and save a maximum of 5.3 *EUR* per sensor. However, net savings depend very much upon the chosen method of integration and assembly of the proposed AWSS node. If the WSS mounting point on the axle is determined to be too hostile for the newly added electronics, then a mounting point that is further away has to be chosen, which incurs associated cabling and assembly costs. While the determination of net cost savings would certainly be non-trivial, a cost of more than 5 *EUR* for mounting the WSS in the current form does warrant a serious consideration of this alternative method.
2. **Warranty savings:** The use of the autonomous WSS instead of the wired version has the potential to reduce aftermarket maintenance. Like any other harness, the WSS harness is responsible for a majority of the reported aftermarket WSS failures. These harness connectivity problems and associated failure can be potentially reduced using real-time wireless communication links. In addition, diagnosing problems in a workshop becomes much simpler as there is no harness to trace faults.
3. **Customization savings:** Due to cable length variations for different axles, there is a considerable customization effort where, for every axle, a separate WSS configuration is developed, stocked and maintained. Each WSS, with a particular harness configuration constitutes a ‘part’, which has to be individually developed and stocked. Because of the wireless link, the AWSS ‘part’ can be made agnostic to the axle on which it is mounted and this can drastically reduce the customization effort.
4. **Possible savings from EBS architectural changes:** There could be several changes in the EBS system architecture, due to the deployment of new wireless sensors, which could lead to savings. For example, if all the nodes communicate wirelessly with the modulator, one or two access points could replace the speed sensing electronics that is present in each modulator. The modulator performs actuation functions which can possibly be re-architected to make use of the intelligent sensors deployed near the wheel.
5. **New EBS applications:** The AWSS node increases the level of distribution of intelligence in the EBS system which opens up the possibilities of new applications. For example, it would

be possible to sense various wheel events such as wheel lock, wheel slip, etc. at the wheel and control them using this new distributed system.

6. **New wheel management applications:** While the AWSS concentrates specifically on wheel speed monitoring, the idea of using the VR sensor as an energy source need not be restricted to this particular application. If the entire pole wheel and VR sensor setup is moved to the wheel hub or even inside the tires, then it would be possible to use the harvested energy to power any number of wheel monitoring sensors. This broadens the horizons of the VR sensor which can basically act as an energy harvesting transducer platform for any wheel-axle assembly.

One such wheel management application is the TPMS where a pressure sensor is mounted inside the wheel to monitor air pressure. Because of the inaccessible nature of the sensor, a significant amount of research is being directed towards finding autonomous energy sources for this sensor. In-tire vibration and strain energy harvesters seem to be popular choices, as pointed out by a few examples such as [19], [41] and [42]. Alternatively some of the proposals target RF energy harvesting as seen in [43]. Being a wheel-axle assembly, the VR sensor and pole wheel combination can be adapted to fit inside the tire to power the TPMS. Tire pressure monitoring may not have as critical real-time execution requirements as the EBS system and therefore energy harvesting sensors can be used with a higher amount of confidence in such an application.

## Chapter 5 - Prototype specification

As a starting point, the simplest version of the AWSSN, as described in section 4.2, is targeted for implementation in this project. While the prototype cannot target all of the perceived challenges listed in section 4.3, it will target the most important operational concerns of availability and capability for real-time task execution.

The architecture of the targeted prototype sensor network is presented in section 5.1 and the functions chosen to implement in this prototype system is listed in section 5.2. A systematic parametric specification of the prototype system is specified in section 5.3 and a set of general design principles for implementing any real time energy harvesting autonomous system is presented in section 5.4.

### 5.1 Prototype network architecture:

The AWSSN in its simplest form consists of one AWSS node acting as the ED and one AP which receives messages from the ED and relays them to the EBS modulator. An AWSSN consisting of only two nodes is the target of implementation. This is depicted by the following figure.

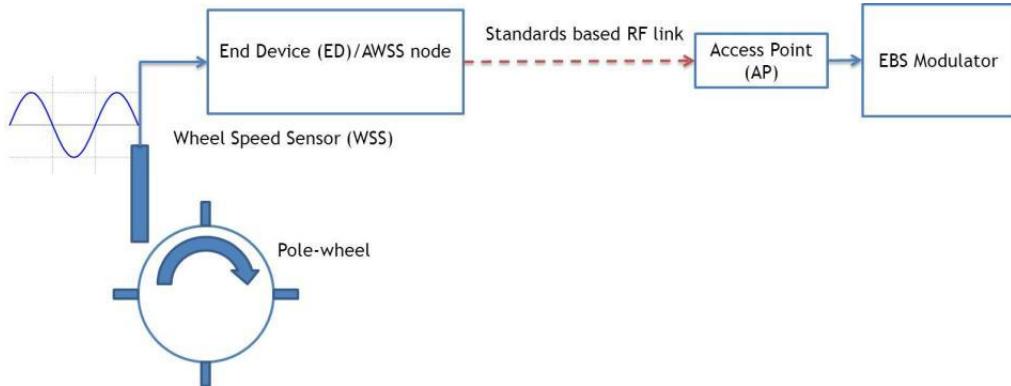


Figure 14: Conceptual depiction of the targeted prototype - a two node AWSSN

However the EBS modulator in its current form is designed to operate only with the raw WSS signal and cannot process discrete messages. Since modification of the modulator is not possible in the limited time that is available for this project, an additional simplification is done by implementing a Matlab based Pseudo Modulator (PM). The PM simply displays messages relayed by the AP, in a setup that is depicted by the following figure.

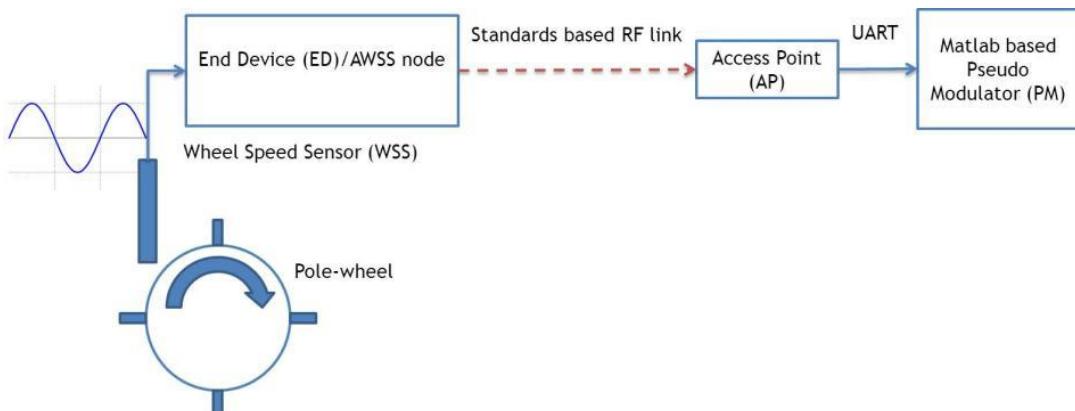


Figure 15 : Targeted prototype AWSSN setup

The choice of the two-node WSN abstracts away many challenges in real-time wireless scheduling such as Media Access Control (MAC), network level routing and many more challenges posed by a multiple node WSN. All these challenges are left to be tackled in future iterations of the project.

## 5.2 Prototype functionality:

In the EBS system deployed in trucks the WSS signal data is used for a number of applications, as described in section 3.4. Among these applications the ABS is targeted and the prototype AWSSN system implements a representative sample of functions intended for the ABS application. From the performance of these representative functions, results can be drawn to evaluate applicability to other EBS applications. It must be noted at this point that no concern apart from high availability and real-time task execution are considered in the prototype implementation. Keeping in mind the time-limited nature of the project, the following functionality were chosen for implementation as a representative set for the ABS.

1. **Periodic wheel speed sensing** - the ED shall sense and transmit the speed of the wheel periodically to the AP which relays the message to the PM.
2. **Wheel rotational state sensing** - the ED shall sense the starting/stopping of the wheel and instantaneously transmit an event to the AP which is relayed to the PM. This message would signal that the wheel has started/stopped rotating.

The periodic wheel speed sensing and transmission represents the data stream which operates at a far lower duty cycle than the passive WSS and the rotational state sensing represents the event stream.

*An emphasis on event-driven communication and control, in place of data-driven communication, in a network of autonomous sensors is one of the key recommendations of the project.* Continuously transmitting wheel speed data makes poor use of the processing capabilities of the AWSS and the distribution of intelligence. While data and events can be combined, the best case scenario of operation with the AWSS node would be a completely event-driven model where the node would handle most of the management functions on-site and communicate only important events up the hierarchy.

The main reasoning behind this idea is the fact that abstracted events are likely to occur less frequently in time and therefore can potentially reduce the duty cycle of operation, consequently improving the availability. This principle motivates most of the design choices of the prototype. Extrapolating this idea, when it comes to applications such as ASR, ESC, etc. the recommended course of operation is to detect events such as loss of control and loss of stability at lower levels of system hierarchy and communicate them to points of control/for actuation.

## 5.3 Parametric model:

A systematic specification of the targeted AWSSN, as shown in Figure 15, is the first step in choosing the hardware, designing and eventually developing the prototype. The parametric definitions in this section are quite extensive and for ease of reference, a table of all parameters (the parametric register) has been compiled and presented in Appendix E. The notations of the different parameters are used consistently throughout the report and in the source code of the software.

### 5.3.1 Specifying the WSS:

As noted earlier, the WSS is the primary object of interest for sensing and energy harvesting and in many ways dictates the requirements and limitations of the system. The WSS generates a sinusoidal

voltage signal at its output whose frequency depends upon the speed of rotation of the pole wheel. The WSS can be specified using the following parameters.

Notation	Description
$g_a$	Air gap between the pole wheel and VR sensor (m)
$d_{wheel}$	Diameter of the wheel (m)
$n_{teeth}$	Number of teeth in the pole wheel
$s_{min}$	Minimum wheel speed (Kmph)
$s_{max}$	Maximum wheel speed (Kmph)
$V_1$	Sensor output voltage for 1Hz (V/Hz)
$f_{min}$	Minimum pole wheel teeth frequency (Hz)
$f_{max}$	Maximum pole wheel teeth frequency (Hz)
$t_{min}$	$1/f_{max}$
$t_{max}$	$1/f_{min}$
$V_{min}$	Sensor output voltage at $f_{min}$
$V_{max}$	Sensor output voltage at $f_{max}$
$Z_{WSS}$	Source impedance of the WSS ( $\Omega$ )
$P_{min}$	Minimum average signal power with matched impedance (W)
$P_{max}$	Maximum average signal power with matched impedance (W)
$P_{mean}$	Mean average signal power with matched impedance (W)
$\lambda_{transition}$	The average number of rotational transitions per hour
$p_{lock}$	Proportion of time for which the wheel is locked i.e. not rotating

*Table 2 : WSS functional parameters*

The speed at which the wheel of a truck rotates is a logical place to begin modeling as it is this rotational speed that is represented by the WSS output. The conversion from speed in Kmph to frequency in Hz can be done using the conversion factor

$$c_{vTof} = \frac{0.278 * n_{teeth}}{\pi d_{wheel}} \quad (2)$$

Using this, the frequency parameters  $f_{min}$  and  $f_{max}$  can be derived from  $s_{min}$  and  $s_{max}$  using

$$f = c_{vTof} * s \quad (3)$$

It should be noted here that  $f_{min} > 0$  because the WSS is not capable of detecting a speed of 0. Since the relation between the frequency and the induced voltage in a VR sensor is linear, according to the Maxwell-Faraday law, the voltage parameters  $V_{min}$  and  $V_{max}$  can be derived from the frequency parameters  $f_{min}$  and  $f_{max}$  and  $V_1$  using

$$V = V_1 * f \quad (4)$$

The output power limits  $P_{min}$  and  $P_{max}$  can be derived from the voltage limits  $V_{min}$  and  $V_{max}$  using

$$P = \frac{V_{RMS}^2}{4Z_{WSS}} \quad (5)$$

As the wheel is put through its paces, its speed converges to a statistical average  $P_{mean}$ . If the instantaneous wheel speeds are uniformly distributed, then this mean average power is given by

$$P_{mean} = \frac{(P_{max} + P_{min})}{2} \quad (6)$$

It must be noted that the choice of a uniform distribution is only for illustration and the idea is valid with any other statistical distribution. In practical cases, the distribution of wheel speeds is highly unlikely to be uniform, but the distribution is chosen nevertheless for simplicity. The average power in (6) is applicable only if the wheel is rotating continuously. If the proportion of time for which the wheel is not rotating ( $p_{lock}$ ) is considered then the average power is given by

$$P_{mean} = \frac{P_{max} + P_{min}}{2} (1 - p_{lock}) \quad (7)$$

The above reasoning follows elementary probability theory and is generic enough to be applied with any statistical model of wheel rotation.

As an illustration, the transition of the rotational state of the wheel can be modeled using the well-known Random Telegraph (RT) process, described in Appendix A. The RT process is a readymade bi-stable random process where the transition between the states follows a Poisson distribution. Since the state probabilities converge, as shown in Figure 53 and Figure 54, the RT process model is modeled in a way that is sensitive to knowledge of the initial state. As the wheel of a truck is meant to be rotating most of the time, the assumption is made that the wheel starts from an unlocked position. According to the RT process model in an interval of time  $(0, t)$  we have

$$p_{lock} = P\{X(t) = 0 | X(0) = 1\} = e^{-\lambda t} \sinh(\lambda_{transition} t) \quad (8)$$

The choice of  $X(0) = 1$  turns out to be necessary because, from (41) and (44), we have

$$\min[P\{X(t) = 0 | X(0) = 1\}] = 0$$

$$\min[P\{X(t) = 0 | X(0) = 0\}] = 0.5$$

This shows that a choice of  $X(0) = 0$  is quite restrictive for modeling the wheel rotation because it is necessary for  $p_{lock}$  to be less than 0.5. The sensitivity of the knowledge of initial state however reduces exponentially with the Poisson parameter as shown in (46).

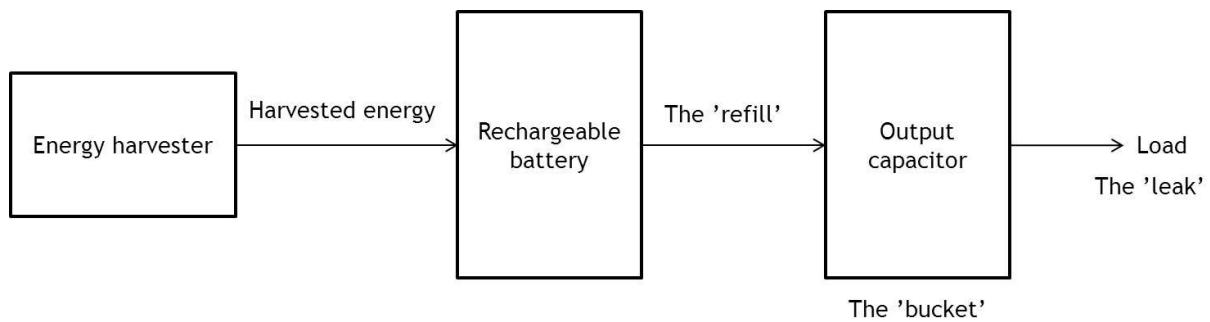
The parameter  $\lambda_{transition}$  in (8), is the average number of rotational state transitions per hour. Using (8) in (7),

$$P_{mean} = \frac{P_{max} + P_{min}}{2} e^{-\lambda t} \cosh(\lambda_{transition} t) \quad (9)$$

The choice of the RT process may not be optimal for modeling wheel rotation because of the convergence of state occupation probabilities. As time progresses, in the RT process model, both states become equally likely to be occupied as shown in Figure 53. The fitness of the model can therefore be evaluated only after careful examination of wheel speed data. Another simplification in the above model is that the distribution of wheel speeds and the locking of the wheel has been modeled independently, which is also not true in reality. The main intention here is to illustrate and promote stochastic modeling of the wheel speed for analyzing the available power as well as the sensing activity.

### 5.3.2 Modeling the AWSS node using the 'leaky bucket' analogy:

The ED is the primary object of interest in this project and represents a major part of the implementation effort. The operation of the energy harvesting node ED can be described as using water from a 'leaky bucket'. Here the water in the bucket represents the energy that is available in the EHU at the disposal of the WSU and the leak represents the actual usage of energy from the bucket. The EHU always tries to replenish the leak of energy that is drained by the WSU. The bucket here is normally a capacitor, or in some cases a supercapacitor, capable of providing power in bursts. The EHU could either replenish the leak directly from the harvester input or use a bucket of its own, for example a rechargeable battery. The analogy is depicted by the following figure:



**Figure 16 : The 'leaky bucket' model of the energy harvesting ED node**

For sustainable operation the refill rate must be able to sustain the leak rate, which in effect captures the essence of operation of an energy harvesting system. The leaky bucket analogy has been used to model the operation of the ED node and design its functionality. In the above figure, the harvester, rechargeable battery and capacitor are part of the EHU and the WSU acts as the load.

### 5.3.3 Specifying the EHU of the ED:

The power that is available from the WSS in an average sense is modeled by ( 7 ) and in the specific case of an RT process model by ( 9 ). The EHU must be able to successfully harvest this energy from the WSS output signal and supply a regulated DC voltage. The EHU parameters can be listed as follows:

Notation	Description
$E_{min}$	Minimum harvestable voltage at the energy harvester input (V)
$E_{max}$	Maximum harvestable voltage at the energy harvester input (V)
$Z_{min}$	Minimum allowed transducer source impedance ( $\Omega$ )
$Z_{max}$	Maximum allowed transducer source impedance ( $\Omega$ )
$D_{bat}$	The capacity of the rechargeable battery ( $\mu\text{Ah}$ )
$R_{bat}$	Effective series resistance of the battery( $\Omega$ )
$C_{out}$	Output capacitor (F)
$V_{cap}$	Instantaneous supply voltage across $C_{out}$ (V)
$D_{cap}$	Instantaneous discharge from $C_{out}$ ( $\mu\text{Ah}$ )
$V_{maxCap}$	Maximum supply voltage across $C_{out}$ (V)
$V_{minCap}$	Minimum allowable voltage to which $C_{out}$ can discharge (V)
$V_{chargeCap}$	The charging voltage of $C_{out}$ (V)
$D_{peakCap}$	Maximum allowable peak discharge ( $\mu\text{Ah}$ )

Table 3: EH parametric specification

With the conditions

$$E_{min} \leq V_{min} < V_{max} \leq E_{max} \quad (10)$$

$$Z_{min} \leq Z_{WSS} \leq Z_{max} \quad (11)$$

The capacitor  $C_{out}$  provides the output power to the load which in turn must draw the power in bursts. In one burst the load draws a current of  $I_{RMS}$  in time  $t_{discharge}$  and discharges  $C_{out}$  from  $V_{maxCap}$  to  $V_{cap}$ . The discharge of the capacitor is modeled as follows.

$$t_{discharge} = RC_{out} \ln \frac{V_{maxCap}}{V_{cap}} \quad (12)$$

Though this discharge is exponential, if the discharge magnitude is small enough, it can be approximated as a linear discharge. In such a case

$$R = \frac{V_{avg}}{I_{RMS}} = \frac{V_{maxCap} + V_{cap}}{2I_{RMS}} \quad (13)$$

Using (13) in (12) we have

$$t_{discharge} I_{RMS} = \frac{C_{out}(V_{maxCap} + V_{cap})}{2} \ln \frac{V_{maxCap}}{V_{cap}} \quad (\text{As}) \quad (14)$$

The product of the average current drawn and the time for which it draws this current gives the charge that is drawn out from the buffer capacitor. The expression in ( 14 ) expresses the discharge in units of  $As$  which is converted into units of  $\mu Ah$  as follows

$$D_{cap} = \frac{277.8 * C_{out}(V_{maxCap} + V_{cap})}{2} \ln \frac{V_{maxCap}}{V_{cap}} \quad (15)$$

Expressing the discharge in terms of a current-time product is very useful for practical embedded system development where the current that is drawn and the time for which is drawn are easily relatable in software and hardware.

Even though  $V_{cap}$  can go down close to 0 V, this is not desirable because most CMOS circuits need a supply voltage of at least 1.8 V to operate. In addition, too much fluctuation in  $V_{cap}$  may also not be desirable because there may be circuit applications which depend upon the magnitude of the supply voltage. This implies that the discharge of  $C_{out}$  should be carefully regulated to ensure that  $V_{cap}$  stays within well-defined limits. In this model these limits are set by  $V_{maxCap}$  and  $V_{minCap}$ , with preferably a small difference between them. These voltage limits can be converted into an equivalent discharge limit which can be expressed as follows.

$$D_{peakCap} = \frac{277.8 * C_{out}(V_{maxCap} + V_{minCap})}{2} \ln \frac{V_{maxCap}}{V_{minCap}} \quad (16)$$

The WSU must make sure to not draw a charge that is higher than  $D_{peakCap}$  so that the voltage across the capacitor is maintained between  $V_{maxCap}$  and  $V_{minCap}$ .

Having modeled the discharge, attention can now be turned to modeling the charging of  $C_{out}$  by the rechargeable battery. As shown in the leaky bucket analogy, the leak of charge from  $C_{out}$  due to consumption by the load is replenished by the rechargeable battery. When  $C_{out}$  has been discharged to  $V_{cap}$ , the time taken for the battery to charge it back to  $V_{maxCap}$  is given by

$$t_{charge} = -R_{bat}C_{out} \ln \frac{V_{maxCap} - V_{chargeCap}}{V_{cap} - V_{chargeCap}} \quad (17)$$

Therefore after discharging  $C_{out}$  to  $V_{cap}$ , the WSU must necessarily refrain from further discharging the capacitor for at least  $t_{charge}$  amount of time, a necessity which arises out of the indefinite nature of periodic transmission.

While being nearly accurate models of the charge and discharge processes, ( 14 ) and ( 17 ) need to be simplified further to be practically usable. For this, another approximation is made that the discharge from  $V_{maxCap}$  to  $V_{cap}$  is quick enough such that the average value of the drop is  $V_{maxCap}$ . While this discharge profile does not reflect the exponential discharge it may just work if the discharge is below, for example, 1% of  $V_{maxCap}$ . In such a case, we have

$$t_{discharge} = \frac{C_{out} V_{maxCap}}{I_{RMS}} \ln \frac{V_{maxCap}}{V_{cap}} \quad (18)$$

The above equation can equivalently expressed, based on ( 16 ), as

$$D_{cap} = 277.8 * C_{out} V_{maxCap} \ln \frac{V_{maxCap}}{V_{cap}} \quad (19)$$

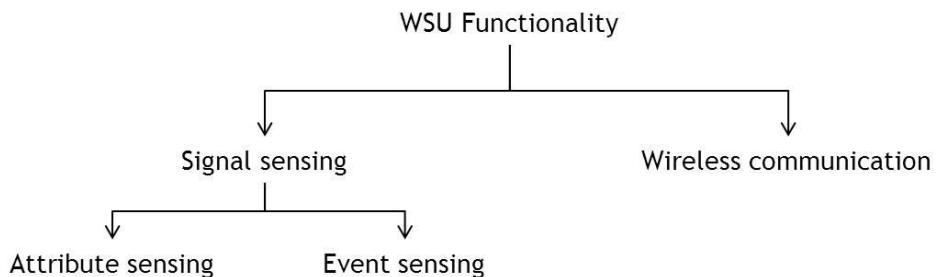
While being approximate, the expressions in ( 17 ) and ( 19 ) are more practically usable as the only unknowns would be  $V_{cap}$  and  $t_{charge}$  while all other values can either be measured or set. The value of  $t_{charge}$  effectively sets the limits on the periodic transmission period and therefore the duty cycle of operation of the WSU.

### 5.3.4 Specifying the WSU of the ED

The operation of the WSU can be described as follows.

1. The WSU must be able to sense and process sinusoidal waves ranging from  $f_{min}, V_{min}$  to  $f_{max}, V_{max}$ .
2. As long as the WSS output signal frequency  $f > 0$ , the WSU must be able to sense if  $f$  drops to zero (or below a certain threshold) indicative of the fact that the wheel is locked and immediately transmit a message indicating this event.
3. As long as  $f = 0$  (or below a certain threshold), the WSU must be able to sense if  $f$  crosses zero or the threshold indicating that the wheel has unlocked and transmit a message communicating this event.
4. Sensed results should be transmitted on a standards based RF link
5. The WSU must be able to save energy by having a duty cycle of  $<1$

Based on the above statements, a classification of ED functionality can readily be formulated as follows.



**Figure 17 : Classification of WSU functionality**

Here attribute sensing refers to the sensing of transient attributes in the WSS system, which in this case is the periodic speed sensing and transmission.

**WSU sensing model:** The attribute sensing functional requirement implies that the WSU should have a frequency sensor. Time period sensing was chosen to implement frequency sensing because of the ready availability of digital time period sensing techniques and the ability to realize these techniques without consuming too much power. Here, time period sensing uses a counter to determine the time elapsed between successive rising/falling edges of the WSS signal output.

It has to be noted that the WSS signal varies not only in frequency but also in amplitude. For pure frequency sensing purposes, it is advantageous to abstract away these variations in amplitude and

present a consistent signal that varies only in frequency. In order to perform this conversion a specific interface circuit called the ‘WSS pulse converter’ is now defined which implements the logic:

<b>WSS signal input</b>	<b>WSS Pulse converter output</b>
$V_{inst} > V_{pulseTH}$	<i>CMOS logic 1 (3 to 3.8 V)</i>
$V_{inst} < V_{pulseTH}$	<i>CMOS logic 0 (-0.3 to 0 V)</i>

*Table 4: WSS pulse converter truth table*

Here  $V_{inst}$  is the instantaneous voltage of the WSS output signal. Typically a switching threshold of 0 V is not possible due to voltage hysteresis, and is usually around a few hundred millivolts. The time period sensing functionality, as implemented by the pulse converter and the time period sensor, can be specified as follows.

<b>Notation</b>	<b>Description</b>
$w_t$	Sensing window (number of rising edges to sense for calculation and averaging)
$b_t$	Timer word length (bits)
$f_t$	Timer clock rate
$t_{attSense}$	Time taken to sense and process an attribute (s)
$t_{attInterval}$	Minimum interval between successive transmissions of the attribute (s)

*Table 5 : WSU wheel speed attribute sensing specification*

So that we have conditions:

$$f_t > f_{max} \quad (20)$$

$$b_t > \log_2 f_{max} \quad (21)$$

$$w_t \geq 2 \quad (22)$$

Since attribute sensing senses the time period of a pulse wave, the sensing duration depends upon the instantaneous frequency of the signal. Keeping up the assumption of uniformly distributed speeds, the attribute sensing time can be expressed using the expected value of this distribution.

$$t_{attSense} = \frac{w_t}{2} (t_{max} + t_{min}) \quad (23)$$

The attribute transmission interval is chosen to be fixed in the prototype implementation and must necessarily be longer than the longest attribute sensing period.

$$t_{attInterval} \geq w_t * t_{min} \quad (24)$$

When it comes to event sensing, the WSU must be able to sense the event, i.e. the locked or unlocked state of the wheel, and communicate it. Though the lock status can be sensed from the periodic speed attribute, such sensing does not satisfy real-time event sensing needs and calls for a dedicated

rotational state sensor. The rotational state sensor is a bi-stable multivibrator, whose two states represent the rotational states, as depicted by the following truth table.

<b>WSS sinusoidal input</b>	<b>Rotational state sensor output</b>
$f_{sin} > f_{lock}$	<i>CMOS logic 1 (3 to 3.8 V)</i>
$f_{sin} < f_{lock}$	<i>CMOS logic 0 (-0.3 to 0 V)</i>

**Table 6 : WSS rotational state sensor truth table**

The transitions of the rotational state sensor output are indicative of the change of state from rotating to stationary and these transitions are primary objects of interest. The event sensing functionality can be specified by:

<b>Notation</b>	<b>Description</b>
$f_{lock}$	Lock threshold frequency (Hz)
$t_{rise}$	The rise time of the output signal indicating an unlock event (s)
$t_{fall}$	The fall time of the output signal indicating a lock event (s)
$t_{eventSense}$	Time taken to sense a transition event (s)
$t_{eventInterval}$	The average time between two transition events (s)

**Table 7 : WSU event sensing specification**

When it comes to modeling  $t_{eventSense}$ , one has to consider the fact that the events occur in a non-deterministic manner. Therefore for highest reliability of event detection, the event sensing period must be indefinite.

$$t_{eventSense} = \infty \quad (25)$$

The average duration between transitions is given by:

$$t_{eventInterval} = \frac{3600}{\lambda_{transition}} \quad (26)$$

**WSU communication model:** After sensing and processing the WSS signal, the WSU must then communicate results to the AP. The communication functionality can be specified as follows.

<b>Notation</b>	<b>Description</b>
$b_{PHY}$	PHY level packet length (bits)
$r_{RF}$	Data rate of transmission (kbps)
$t_{transmit}$	The time taken for transmitting $b_{PHY}$ (s)

**Table 8 : WSU RF communication specification**

Here,  $b_{PHY}$  would include the attribute/event payload and the overhead that is added to it by the WSN communication standard. Parameters such as the PHY level overhead and the data rate are well specified by the WSN communication standard.

$$t_{transmit} = \frac{b_{PHY}}{r_{RF}} \quad (27)$$

Apart from communicating attributes and events the WSU must also be able to maintain the link between the ED and the AP according to the WSN standards, but a detailed specification of WSN management is beyond the scope of this project.

**WSU duty cycle model:** The WSU and hence the ED consumes energy when it senses and transmits the attributes and events. This system, or indeed any energy harvesting system, is better off if it has an operating duty cycle of << 1(100%), so that it is active for a significantly lower amount of time compared to the time when it is not. This is mainly because of the intermittent availability of the ambient energy and finite EHU capacity, which precludes the sensor from operating continuously. New parameters can now be defined to define the duty cycle of the ED:

Notation	Description
$t_{attActive}$	The active period for attribute sensing and transmission (s)
$t_{eventActive}$	The active period for event sensing and transmission (s)
$d$	Duty cycle of operation

*Table 9 : WSU duty cycle specification*

In the attribute sensing case, the active period is quite straightforward as the sensing time  $t_{attSense}$  is determinable. Therefore we have

$$t_{attActive} = t_{attSense} + t_{transmit} \quad (28)$$

When it comes to event sensing, we have a sensing period that is indefinite, which in practical cases could last several minutes. If an active sensor is used, then this indefinite active period - which could draw steady power for many minutes - would potentially violate the discharge limits set by ( 14 ).

*In order for the rotational state sensor, or indeed any sensor in an energy harvesting system, to be indefinitely active, it must necessarily use passive means of sensing.*

Now if the rotational state sensor is passive, an indefinite active period has no impact on the active duty period. The event sensing function is now considered active only when a transition occurs which triggers a radio transmission.

$$t_{eventActive} = t_{transmit} \quad (29)$$

Since in most practical cases  $t_{attActive} \ll t_{attInterval}$  and  $t_{eventActive} \ll t_{eventInterval}$ , duty cycle is given by:

$$d = \frac{t_{attActive}}{t_{attInterval}} + \frac{t_{eventActive}}{t_{eventInterval}} \quad (30)$$

The above equation can be equivalently expressed using ( 26 ) as

$$d = \frac{t_{attActive}}{t_{attInterval}} + \frac{\lambda_{transition} * t_{eventActive}}{3600} \quad (31)$$

With event sensing and transmission now in place, it is not necessary to transmit the attributes in the duration the wheel is locked. This means that the sensor is not active for the entire period when the wheel is locked and this significantly reduces the duty cycle as seen below.

$$d = \frac{t_{attActive}(1 - p_{lock})}{t_{attInterval}} + \frac{\lambda_{transition} * t_{eventActive}}{3600} \quad (32)$$

In ( 32 ) the advantage of event sensing and transmission becomes apparent because it decreases the most influential term of the duty cycle by a factor of  $(1 - p_{lock})$ .

It is not surprising to note that passive sensing is better in every case which is shown by the fact that if the pulse converter is passive,  $t_{attSense}$  is essentially 0 so that

$$t_{attActive} = t_{transmit} \quad (33)$$

This reduction in  $t_{attActive}$  further brings down the duty cycle of operations. Therefore, while passive sensors are best to have for both deterministic and non-deterministic sensing, it is an absolute requirement for non-deterministic sensing in autonomous sensors.

**WSU average discharge and power consumption model:** The ED functionality has now been broken down into a set of discrete operations, namely event and attributes sensing. The average discharge and power consumption, during the operational lifetime of the ED, can be calculated if the discharge and the current drawn due to each individual operation are known. Defining new parameters:

Notation	Description
$I_{attActive}$	RMS current drawn during attribute sensing and transmission (A)
$D_{attActive}$	Buffer capacitor discharge due to attribute sensing ( $\mu\text{Ah}$ )
$I_{eventActive}$	RMS current drawn during event sensing and transmission (A)
$D_{eventActive}$	Buffer capacitor discharge due to event sensing ( $\mu\text{Ah}$ )
$D_{avg}$	Average discharge ( $\mu\text{Ah}$ )
$I_{avg}$	Average current consumption (A)
$P_{avg}$	Average power consumption (W)

**Table 10 : ED power consumption parameters**

The average discharge is only a slight modification of the duty cycle expression ( 32 ) and is given by

$$D_{avg} = \frac{D_{attActive} * t_{attActive} * (1 - p_{lock})}{t_{attInterval}} + \frac{D_{eventActive} * \lambda_{transition} * t_{eventActive}}{3600} \quad (34)$$

Similarly the average current that is drawn is given by

$$I_{avg} = \frac{I_{attActive} * t_{attActive} * (1 - p_{lock})}{t_{attInterval}} + \frac{I_{eventActive} * \lambda_{transition} * t_{eventActive}}{3600} \quad (35)$$

Assuming the theoretical discharge model specified in ( 18 ), the average power consumption is given by

$$P_{avg} = V_{maxCap} * I_{avg} \quad (36)$$

### 5.3.5 Specifying the AP and the PM:

The AP and the PM parts of the AWSSN are not systematically specified as part of this project, because of maximum focus on the ED. The AP is not powered by energy harvesting, and acts only as a relay node for forwarding the data received via the wireless link to the PM. The PM is a simple Matlab script running on a laptop and receiving messages from the AP via a USB interface. The PM merely displays the messages that are relayed by the AP. Due to the choice of focusing primarily on the ED in this project, the AWSSN prototype will operate under the following assumptions.

1. The AP is permanently active and is continuously receiving which implies that the ED is permanently on the network hosted by the AP. In addition, no corrective action is implemented in the ED in the event of the AP not being active.
2. The AP never solicits any information from the ED or issues any commands to the ED. Therefore the ED never listens for reception from the AP.
3. The PM similarly never solicits any information from the AP and acts as a passive listener

It should be noted that the above assumptions are in place only for simplifying the prototype implementation due to limited availability of time. The system is definitely capable of shedding this level of passivity and operating without any of the above assumptions once the associated functionality is developed.

## 5.4 General design principles for increasing availability of autonomous sensors:

As stated earlier, in order to address availability and real-time task execution in the AWSS node, the approach taken in this prototype implementation focuses heavily on duty cycle reduction. It has been shown in ( 32 ) that the usage of passive sensing of high abstraction can significantly reduce the duty cycle. This design idea can be extrapolated to a general case of any autonomous sensor node as follows.

**Definition:** Passively-active sensing mechanisms are those which have active circuitry but consume levels of power that are small enough for them to be considered passive. The stimuli derived from these mechanisms can be termed as passive stimuli.

**Illustration:** Passively-active sensors are just another name for ultra-low power sensors, which are similar to passive sensors in the sense that they consume insignificant amounts of power, while differing from them in the sense that they need to be powered.

**Principle 1:** In order to make real-time determinations, which are non-deterministic in time, an autonomous sensor necessarily needs stimulating mechanisms that are either passive or passively-active.

**Reasoning:** This principle is derived from ( 25 ) and its associated reasoning in section 5.3.4. To reiterate the idea in ( 25 ), in order to perform non-deterministic operations, an autonomous sensor node needs stimulating mechanisms which have to be active for indeterminate periods of time. In other words, they must be almost permanently active. If a sensing or stimulating mechanism has to be permanently active, it must not draw significant amounts of power. This is because in energy harvesting sensors, steady power consumption is detrimental. Since passive sensing/stimulating mechanisms do not satisfy many sensing requirements, ultra-low power active circuits are the answer to this problem. Such sensors can be active indefinitely as long as drain currents are tolerable, and perform non-deterministic operations in real time.

**Illustration:** This design principle is already being used, for example, in ultra-low power processing platforms. The sleep oscillator in an ultra-low power microcontroller is active when the microcontroller is sleeping and stimulates it when the sleep period elapses. This sleep timer circuit draws a negligible amount of current, typically a few nano-amperes when it is active.

This same principle can be extended for any other stimulating mechanism such as the rotational state sensor in the AWSS. Since the rotational state can change in a non-deterministic fashion, the rotational state sensor, similar to the sleep oscillator, has to be active even when the microcontroller is sleeping. While being active it should ideally draw a few nano-amperes of current which is a manageable level of drain. This allows the node to sleep until it is stimulated non-deterministically, while losing only small amounts of power between stimulations. On the other hand if the rotational state sensor is an active sensor and draws a few millamps during active operation, it can only be operated in deterministic bursts which mean that it is possible to miss certain events.

Ultra-low power active sensors are becoming quite common with the integration of solid state MEMS/NEMS sensors with CMOS electronics. Examples of such sensors are solid-state accelerometers, solid-state pressure sensors, etc. Such ultra-low power sensors are indispensable for making real-time determinations in an energy harvesting autonomous sensor.

**Principle 2:** As long as an autonomous sensor uses passive sensing and stimulation mechanisms, the duty cycle of operations can possibly be reduced by increasing the level of abstraction of the passive stimuli.

**Corollary:** A decrease in duty cycle increases the availability of the autonomous sensor node

**Reasoning:** In any autonomous sensor, there is always a desire to reduce the duty cycle of operations. Even though ultra-low power sensing can drastically reduce the duty cycle, there are active periods (such as radio transmissions), which cannot be avoided. In the case of such active operations, the duty cycle can be reduced by reducing the frequency of the stimuli which triggers these operations.

**Illustration:** A simple example of this principle is the fact that an autonomous sensor that reacts to change in weather (a higher abstraction of temperature, pressure, humidity, etc.) operates less frequently than a sensor that reacts to changes in temperature alone. If passive sensing is used for sensing temperature, pressure, etc. individually then the abstracted ‘weather’ stimulus is not only passive but is also less frequent. In the case of the AWSS, this principle is illustrated by the change from data-driven to event-driven models proposed in section 5.2. Take the case of determining whether the wheel is rotating or not, which can be derived by monitoring the output of the passive WSS sensor. Even if the monitoring mechanism is passive, it has to be monitored very frequently. However, if the level of abstraction is increased by directly sensing the rotational state, then the node is stimulated directly by a lock/unlock event, which massively reduces the duty cycle. For the

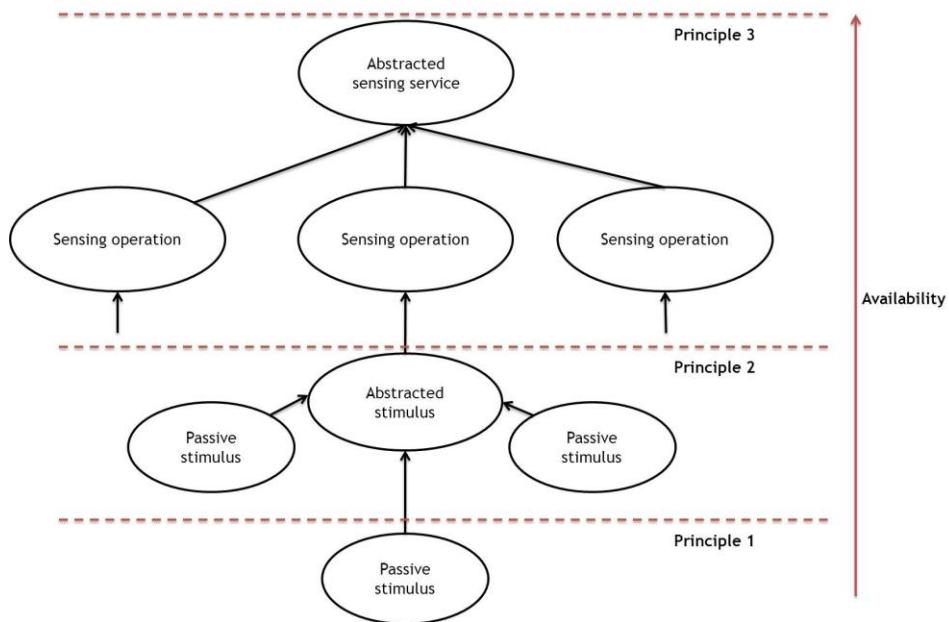
rotational state sensor example it has been shown in ( 32 ) that the duty cycle can be reduced by the proportion of time for which the wheel is locked.

**Principle 3:** The abstraction hierarchy need not be restricted to the sensor node and can be extended beyond, abstracting the operations of one or more autonomous sensors into a unified service, thereby improving the quality and availability of the service.

**Corollary:** An increase in availability and quality of a sensing service increases the reliability of real-time operations using autonomous sensors.

**Illustration:** This principle is readily derived from the WSN model, where abstracting the sensing operations in the network as a service and using more than one sensor for the same operation, increases the reliability of sensing. The node redundancy is necessary because any sensing service provided by the network should not depend upon a single node, which enables the availability of a single sensor node to be defined quite loosely. While this principle seems straightforward, it has to be especially highlighted for the automotive sensing use cases where redundancy is used only for critical functions. The idea of redundancy can be extended to any autonomous sensing case where the nodes are typically inexpensive. For example using two autonomous climate sensors instead of one, when they are CMOS based inexpensive circuits, may not make a huge difference in cost, but can make a significant improvement in the quality of the sensing service. In the AWSS case, having two or even three AWSS nodes per wheel and abstracting their operations into a single service would play a significant part in improving their availability.

It can be clearly seen that the principles are themselves hierarchical in nature and ultimately combine to improve the availability of autonomous sensing. The above defined principles can be alternatively depicted by the following figure:



**Figure 18 : Design principles for increasing availability of autonomous sensing**

While the above principles lay down the framework for tackling availability concerns by reducing the duty cycle, it is however apparent that this method is only the tip of the iceberg. Availability and real-time concerns from many other perspectives such as wireless communication and more complicated task models have not been addressed in the prototype.

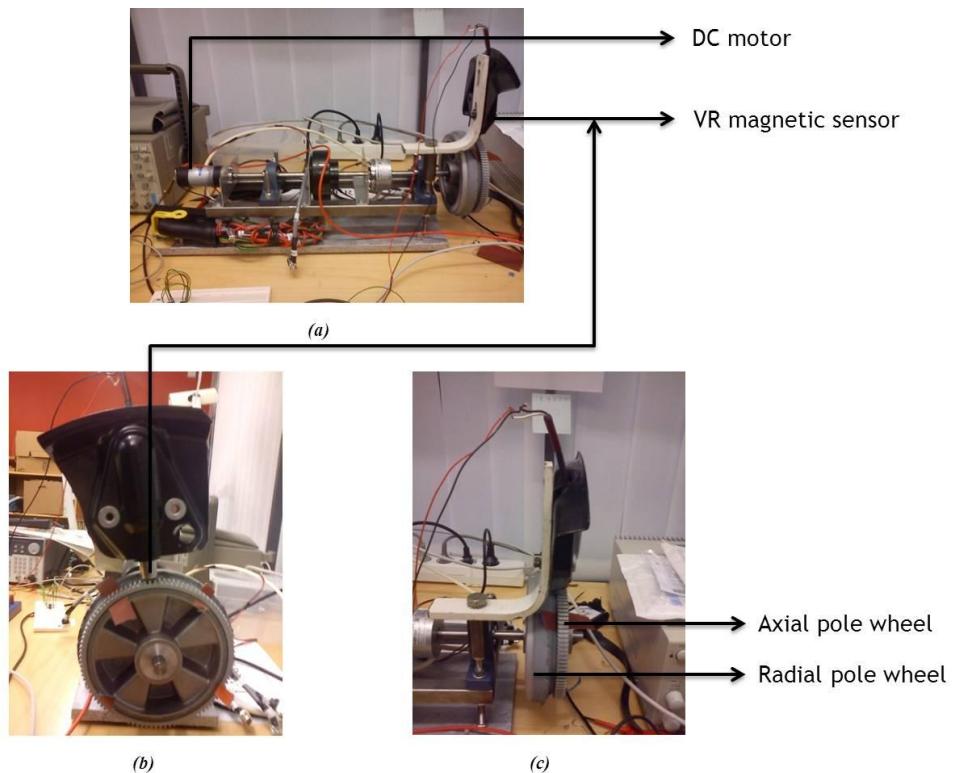
## Chapter 6 - Prototype development setup

This chapter will focus on the most important equipment and tools in the development and experimental setup. The experimental setup is chosen to reflect the system architecture illustrated in Figure 15.

The wheel emulation setup is described in section 6.1 and the hardware for implementing the AWSS node and network is presented in sections 6.2. The hardware and software development and system modeling frameworks are described in sections 6.3, 6.4 and 6.5.

### 6.1 The Wheel Emulation Rig (WER):

The most basic need for developing the prototype AWSS is to emulate a rotating pole wheel and its associated VR sensor and this is done using a custom made Wheel Emulation Rig (WER).



**Figure 19 : Experimental setup for the WSS - (a) the WER and (b) & (c) a closer look at the pole wheel and VR sensor setup**

The main constituent parts of the WER are:

1. The DC motor - A 24V DC motor which rotates an axle on which the pole wheel is mounted. The speed of rotation of the motor, and therefore the wheel, depends upon the DC input voltage magnitude and the amount of load placed on the axle. The DC motor is controlled using a standard DC power supply which emulates requested torque.
2. The pole wheel - a gear wheel, made of a ferromagnetic material. The pole wheel used in the WER has a diameter of approximately 13 cm and has 100 teeth.

- The VR magnetic sensor - the magnetic pickup used in the WER is a commercially available sensor sourced from the manufacturer WABCO, shown in Figure 4. VR WSS sensors from WABCO are used in many Volvo vehicle applications.

The most important concern is of course the range of rotational speeds that this emulation rig has to offer. Data on the range of rotational speeds and the translation of this data into actual truck wheel speeds is presented in section 8.1.

## 6.2 Hardware choices for the ED and AP:

As pointed out earlier, this project is a prototyping effort, where COTS components will be used as much as possible. The specified functional requirements of the system play an important role in choosing the hardware. However, there are also certain non-functional requirements which have to be considered:

1. Ease of development – the components should not impose unreasonable development and integration effort
2. Flexibility of development – the components should have a reasonable choice of usage options for experimenting
3. Ready availability – the components should not have unreasonable delivery times

The non-functional requirements turned out to be more important than the functional requirements, especially due to the time-limited nature of the project. Based on the functional and non-functional requirements, the following hardware choices were made.

### 6.2.1 Choice of EHU - Cymbet CBC-Eval-09:

As described in section 4.1 the EHU should be able to implement the complete energy harvesting chain, which typically involves rectification, maximum power extraction, intermediate storage and output voltage regulation. The energy harvesting development platform used for this project is the Cymbet CBC-Eval-09 universal energy harvesting evaluation kit [44]. The kit provides options for interfacing a variety of energy harvesting transducers ranging from solar to electromagnetic transducers. At the heart of the energy harvesting kit is the CBC-915ACA energy processor which implements the MPPT algorithm for efficient power extraction. The kit implements the entire power management tool chain and the sanitized power is used to charge the EnerChip CBC51100 rechargeable battery.

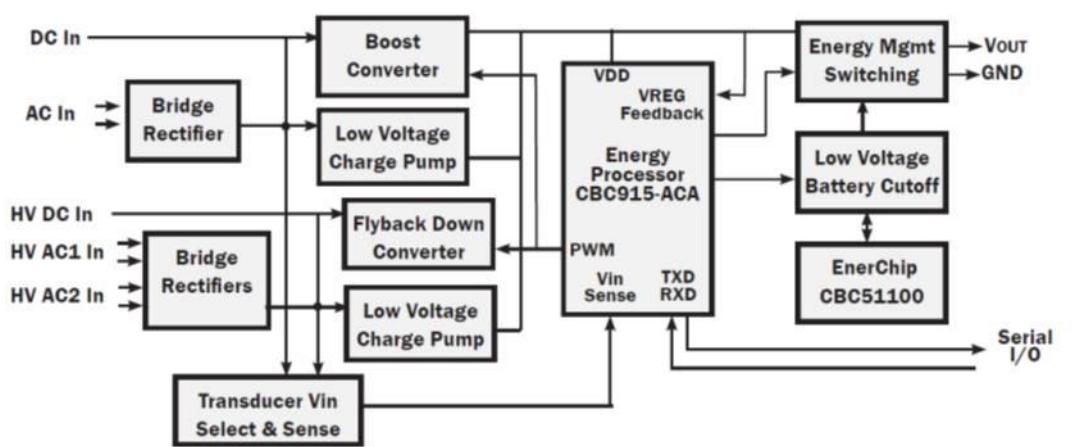


Figure 20 : CBC-Eval-09 functional description due to [44]

The WSS falls under the category of a low voltage, constant source impedance, AC output, voltage sensor which would be connected to the ‘AC in’ terminals seen in Figure 20. Also as described in [44]:

<b>EHU parameter</b>	<b>Value as implemented by CBC-Eval-09</b>
$E_{min}$	600 mV
$E_{max}$	2 V
$Z_{min}$	100 Ω
$Z_{max}$	100 kΩ
$D_{bat}$	100 μAh
$R_{bat}$	1 kΩ
$C_{out}$	1000 μF
$V_{chargeCap}$	3.8 V
$V_{maxCap}$	Can be chosen in the range 3.6 V to less than 3.8 V
$V_{minCap}$	Can be chosen in the range 1.8 – 3 V

*Table 11 : EHU parameters as implemented by the CBC-Eval-09*

The kit has a connector with power and UART Tx/Rx lines to interface with a MCU. More information on the CBC-915ACA energy processor can be found in [45].

### **6.2.2 Choice of WSU and AP – Texas Instruments ez430-RF2500:**

Both the WSU and AP have been chosen to be implemented by the same wireless sensing platform. The sensing platform chosen for this project is the Texas Instruments (TI) ez430-RF2500 kit [46], which is a development tool consisting of two ez430 sensor boards. Each sensor board combines the TI MSP430F2274 16-bit ultra-low power MCU [47] and the TI CC2500 ultra low power RF transceiver [48]. Of the two ez430 sensor boards available in the kit, one is programmed to be the WSU of the ED and the other as the AP.

The choice of TI’s MSP430 family of ultra-low power microcontrollers is based on its high levels of acceptance in the user community and most importantly, TI is among the very few vendors who provide a complete COTS ultra-low power wireless sensing platform. The project also has an implicit objective of evaluating the TI MSP430 based platform for energy harvesting applications. The signature characteristic of the MSP430 is its ultra-low power operation where, for example, the MSP430F2274 draws an active current of less than 300 μA at a 2.2 V supply. In addition a wide range of low power operational modes are also available which draw less than 1 μA. The CC2500 is an ultra-low power proprietary RF transceiver and on the ez430 kit, the CC2500 is connected to the MSP430 as a SPI peripheral.

<b>WSU parameter</b>	<b>Value as implemented by ez430 (MSP430F2274)</b>
$w_t$	Programmable
$b_t$	16 bits
$f_t$	Programmable 12 kHz – 8 MHz

*Table 12 : WSU wheel speed attribute sensing parameters as implemented by ez430 (MSP430F2274)*

The minimum timer clock of 12 kHz offers a reasonable resolution for sensing periods of frequency  $f_{max}$ .

### 6.2.3 Choice of a WSN communication standard:

The choice of the WSN standard is partially restricted by the choice of the ez430-RF2500 development kit. The CC2500 RF transceiver is not 802.15.4 compliant, ruling out the usage of any standard that is based on 802.15.4 such as Zigbee or 6LoWPAN. Among the standards that can be implemented using CC2500 is TI's SimpliciTI WSN standard [49] that is designed for ultra-low power operation. The SimpliciTI standard specifies the following three layer stack with a minimal set of application layer functionality.

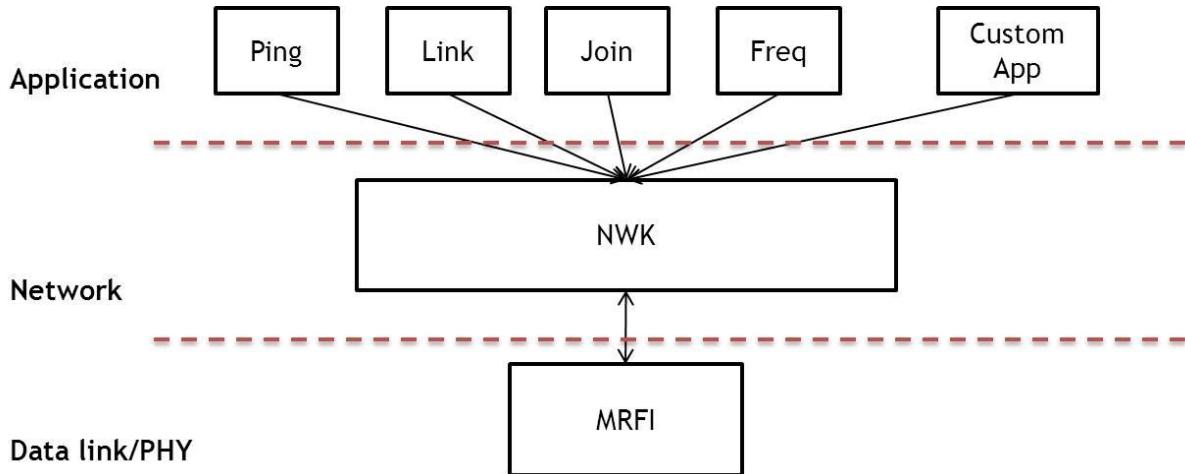


Figure 21 : SimpliciTI communication stack [49]

TI also provides an open source implementation (<http://www.ti.com/tool/simpliciti>) of the SimpliciTI stack with a minimal API which can be readily used with the ez430-RF2500 sensor board. The proven low power operation of SimpliciTI, in combination with knowledge of existing energy harvesting applications using SimpliciTI, prompted its choice as the standard used in this project.

SimpliciTI supports three device profiles where a node can be an End Device, Access Point or a Range Extender. The overlap of terminology with the AWSSN definition is to be noted here: the AP of the AWSSN need not be the Access Point of a SimpliciTI network. SimpliciTI supports two topologies which are a peer-to-peer link between two nodes or a star topology with the Access Point acting as the anchor node.

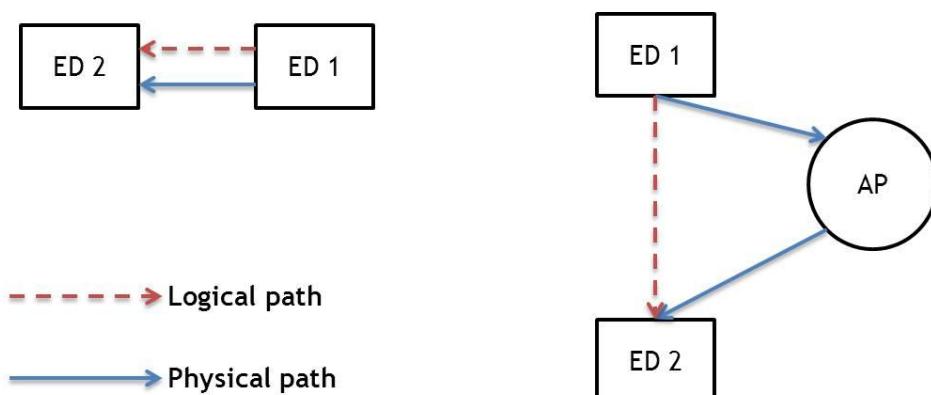


Figure 22 : Topologies supported by SimpliciTI [49]

Each node has a unique address and, in either topology, Range Extenders can be used as a relay between any two nodes. In the prototype application, the bi-directional peer-to-peer link will be used to implement the standards based RF link.

#### **6.2.4 Choice of pulse converter – Maxim MAX3096:**

The conversion of a sinusoidal signal into a unipolar pulse waveform is quite simple, and ICs are available for specifically converting VR signals. Examples of such IC solutions include Maxim's MAX992X and ON Semiconductor's NCV7001, which are TTL logic compatible 5 V ICs. Another way of converting sinusoidal waves to pulse waves is to use a differential line receiver, which is used to convert bipolar RS-4xx bus signals to unipolar RS-232 signal levels. Maxim's MAX3096 [50] is a CMOS compatible differential line receiver, which has been chosen as the pulse converter IC in this project. The MAX3096 has an active current draw of 2.4 mA, which is reasonable for the targeted prototype application.

#### **6.2.5 Choice of rotational state sensor – voltage comparator based on MCP6541:**

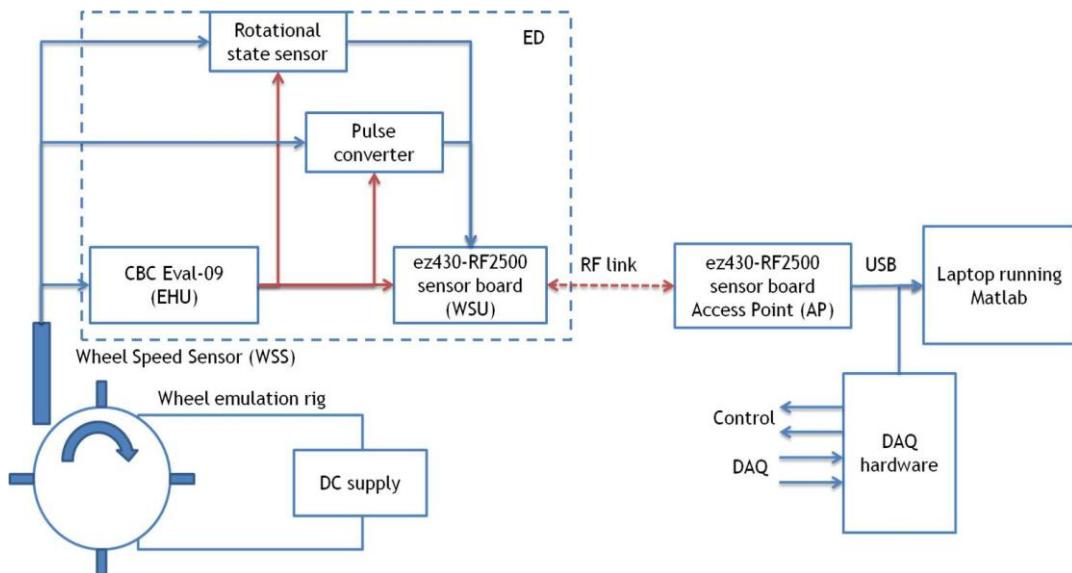
In order to determine signal presence asynchronously for event sensing, a simple comparator circuit based on Microchip's MCP6541 [51] ultra-low power comparator is used. The ultra-low power comparator draws less than 1 $\mu$ A during active operation, which means that it can be permanently active.

#### **6.2.6 One active and one passive sensor:**

When one takes a look at the choices of sensing interfaces, it can be seen that the pulse converter based on MAX3096 is an active sensor while the rotational state sensor based on the MCP6541 is a passive sensor. The rotational state sensor conforms with *Principle 1* and *Principle 2* defined in section 5.4 while the pulse converter violates them. This choice is deliberate to demonstrate the adoption and violation of the design principles that have been defined.

### **6.3 Putting it all together - the hardware development bench:**

The above described development equipment are put together to form the complete hardware test bench. A conceptual depiction of the test bench is shown below.



**Figure 23 : Prototype hardware development and test bench**

## **6.4 Software development:**

MSP430 application programming has been done using C, and the IDEs and tool chains used for the embedded software development are as follows.

**IAR Embedded Workbench for TI MSP430:** For programming the MSP430 in a Windows environment, the IAR Embedded Workbench for MSP430 [52] has been used. The IDE bundles a C compiler, JTAG programmer and debugger into a single application program for convenient MSP430 software development.

**GCC based open source tool chain:** Open source MSP430 development tools for Linux based development environments have also been used, albeit to a lesser extent. The open source gcc based tool chain consists of the mspgcc C and C++ compiler for the MSP430 platform, the mspdebug JTAG programmer and the mspgdb debugger. More information on the open source tool chain can be found in [53].

## **6.5 System modeling:**

System modeling and simulation has been used at various stages of the development process using the following tools

**Matlab:** This has been used for multiple purposes in this project, from mathematical/behavioral modeling to controlling DAQ hardware. Matlab release R2011b has been used in this project.

**COMSOL Multiphysics:** Multiphysics modeling and simulation for the pole wheel and VR sensor setup have been done as part of this project. COMSOL Multiphysics [54] is a commercially available Finite Element Analysis (FEA) based multiphysics simulation tool. The electromagnetic physics interface along with rotational mechanics interface has been used to create a 2D model of the rotating wheel with a VR sensor placed nearby. COMSOL Multiphysics version 4.2a has been used in this project.

**Synopsys Saber:** All electrical and electronic simulation as part of the project was done using the popular Saber tool from Synopsys.

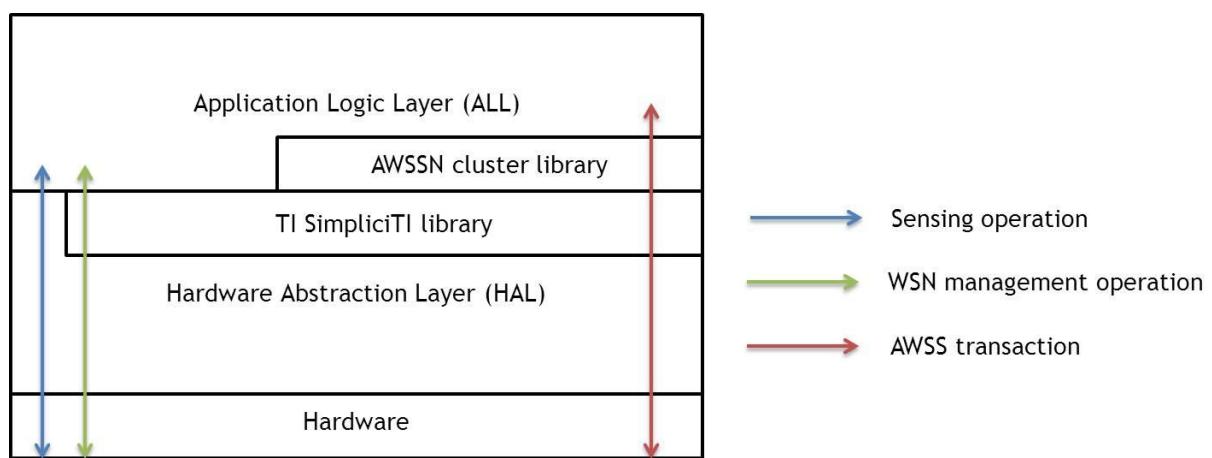
## Chapter 7 - Prototype implementation

The implementation of the AWSSN prototype involves hardware and software development activities using the development setup which was presented in Chapter 6.

The modular definition of each node is described in section 7.1 and the individual layers of the operational stack are defined in sections 7.2 and 7.3. The application level service library for the AWSS node is defined in section 7.4.

### 7.1 Modular stack definition:

As described earlier, each AWSS node is part of a bigger network of similar nodes. Each node has the following modular definition, clearly demarcating the scope of operations.



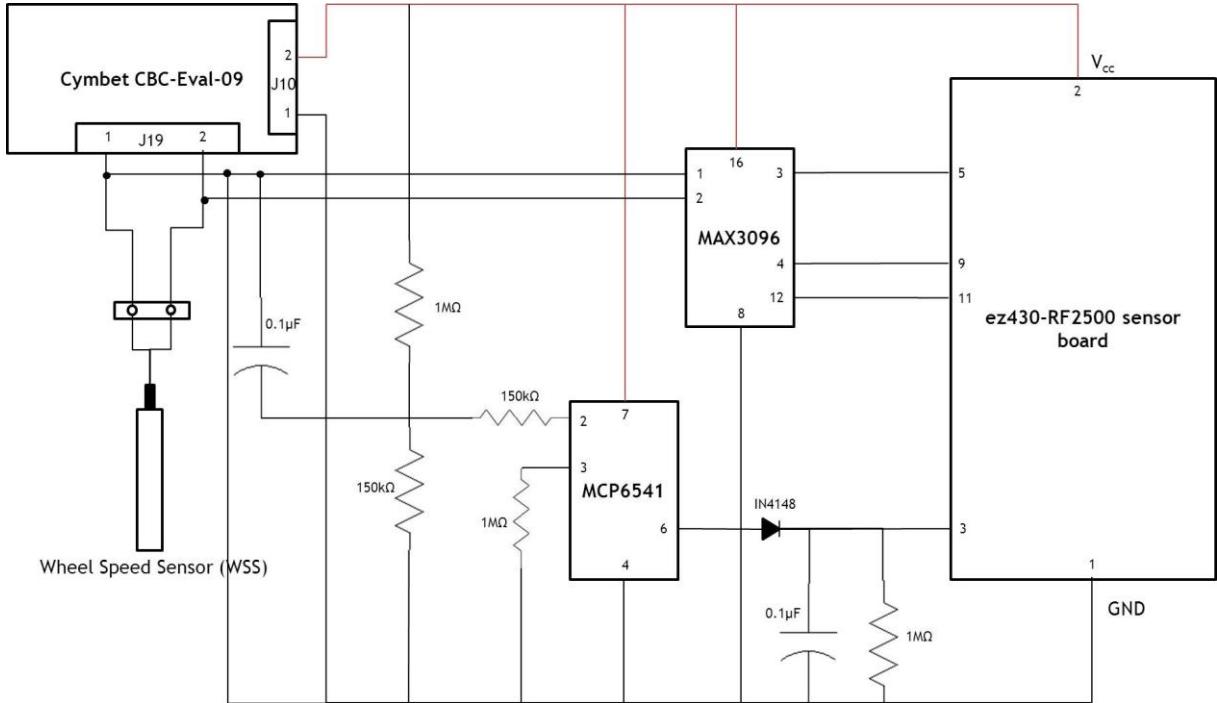
*Figure 24 : Modular definition of an AWSS node*

Due to the simplicity of operations conducted by each node, the complete system can be defined by a stack which spans only three layers. It must be noted that the communication protocol stack has three layers of its own as defined in section 6.2.3. Each layer of the modular stack is defined below.

### 7.2 Hardware Layer:

The Hardware Layer for the ED represents an integration of multiple COTS hardware modules. The ED Hardware Layer schematic is depicted in Figure 25. It should be noted that in the schematic, the Cymbet CBC-Eval-09 and the ez430-RF2500 sensor board are PCBs, whose schematics can be found in [44] and [46] respectively.

The pulse converter circuit uses the MAX3096 integrated differential receiver while the rotational state sensor is implemented as a simple voltage comparator circuit using the MCP6541. Because of the linear relationship between voltage and frequency of the WSS, the truth table specified in Table 6 is implemented as a simple voltage comparator, whose output is rectified to provide a steady digital signal.



**Figure 25 : Hardware Layer of the ED node**

The switching threshold of the pulse converter is decided by the input hysteresis of the MAX3096 which, according to [50], is  $45mV$ . When it comes to the rotational state sensor, as described earlier, the frequency threshold is converted into an equivalent voltage threshold which is decided by the voltage divider at the inverting input. The lock voltage threshold is given by

$$V_{lock} = V_{CC} \frac{150k\Omega}{150k\Omega + 1M\Omega} = 0.13V_{CC} \quad (37)$$

Here  $V_{CC}$  is the supply voltage seen in the schematic in Figure 25. This lock voltage can be converted into its equivalent frequency using the parameter  $V_1$  as

$$f_{lock} = \frac{V_{lock}}{V_1} \quad (38)$$

It should be noted that this threshold is dependent upon  $V_{CC}$ , which spans a  $3 - 3.6 V$  range. This corresponds to a  $V_{lock}$  of  $0.39 - 0.47V$  RMS of the WSS output. In addition the rise and fall times of the rotational state sensor are decided by the RC arrangement on pin 6 of the MCP6541 and can be found in the table below.

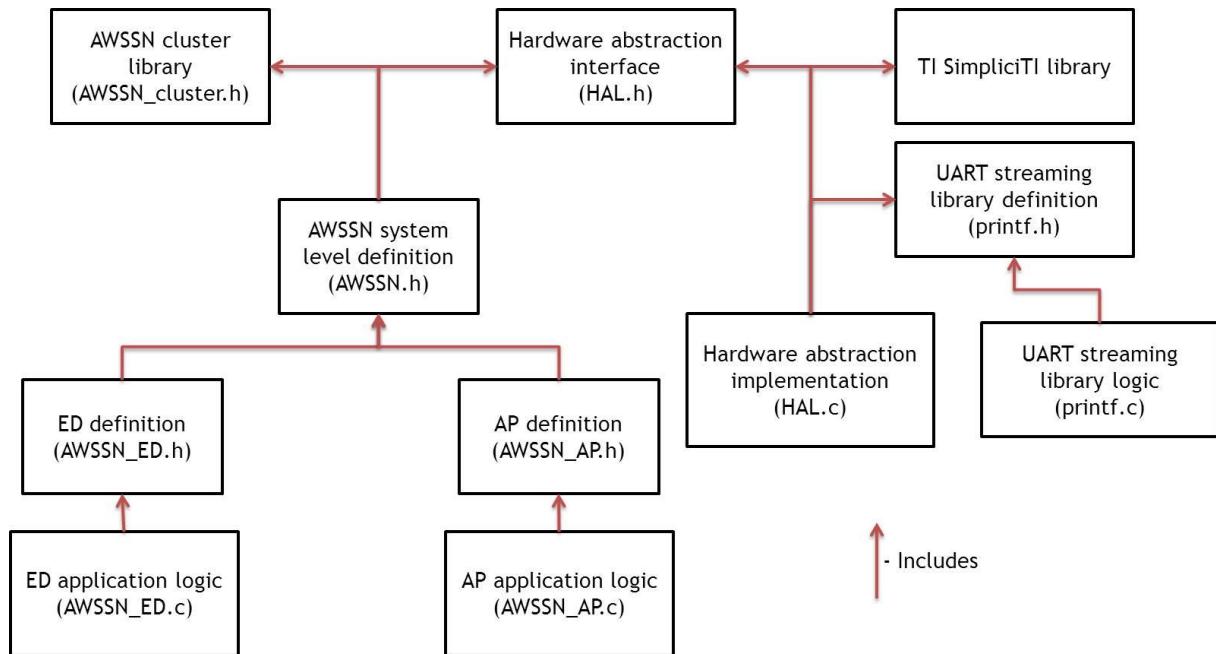
ED parameter	Value as implemented by the Hardware Layer
$V_{pulseTH}$	$45 m$
$f_{lock}$	$0.13 * \frac{V_{CC}}{V_1}$
$t_{rise}$	$3.5 \mu s$ ( $35 \Omega$ IN4148 ON resistance)
$t_{fall}$	$100 ms$

**Table 13 : ED parameters implemented by the Hardware Layer**

Though the discharge time is high, the high discharge resistance has been chosen to keep the ripple amplitude low. However it must be noted that this is only for demonstration and the circuit can be reconfigured as needed.

### 7.3 The software layers:

All layers above the Hardware Layer in Figure 24 are implemented in software and this would be an ideal place to introduce the program structure for implementing the software layers. All software modules are implemented in C.



**Figure 26 : A WSSN program structure**

The programming style adopted for the prototype development includes the following choices:

1. All memory allocation is static for simplicity
  2. There is only one process that is active all the time, multi-tasking is not implemented

The following sections describe each software layer reflecting the program structure depicted above.

### **7.3.1 Hardware Abstraction Layer (HAL):**

The HAL is completely unified in the prototype application in the sense that it is consistently used by both the ED and the AP. The aim of the HAL is to abstract system level like an RTOS. A programmer's/user's guide for the HAL can be found in Appendix B. The services provided by the HAL in the prototype application are:

**Initialization services:** This is usually the first step in system startup that is used to initialize hardware, software and the network and bring the node to a steady (usually a low-power) state. Node initialization routines have been carefully regulated to ensure that long bursts of current are not drawn during node start up. The initialization routine initializes all necessary hardware and connects the node to the network.

**Sleep and wakeup services:** Very often the MCU has to go into the sleep mode in order to conserve power. The MSP430 provides one active mode and four sleep modes or Low Power Modes (LPMs),

out of which, only LPM3 is utilized in the prototype. In LPM3 the MSP430 consumes less than  $1 \mu A$  at room temperature. The HAL provides services to sleep in and wake up from LPM3 to resume operations.

**Interrupt callback services:** The HAL provides callbacks for selected MSP430 interrupts and when an interrupt occurs, the associated callback is executed. Interrupts usually occur when the microcontroller is sleeping and the HAL interrupt callback provides mechanisms to signal whether the microcontroller should wake up or continue sleeping after an interrupt.

**Snooze service:** The signature use case of the snooze service is to pause for a short while during a long sequence of operations. This is important in an energy harvesting sensor in order to stop drawing steady current for a short while, before resuming operations.

**WSN services:** The HAL abstracts away all WSN operations and provides simple and consistent services for node utilization. The prototype HAL provides services only for the ED to join the WSN and transmitting a message. The WSN used is the SimpliciTI 'Link' network service where the AP acting as a SimpliciTI End Device listens for a link request, so that the ED (another SimpliciTI end device) links to it. As specified in section 5.3.5, the AP must be up and running by the time the ED is seeking to join. WSN services for the AP are not well encapsulated in the prototype and the ALL freely makes use of SimpliciTI services in violation of the modular stack. The default SimpliciTI implementation sets a data rate of  $r_{RF} = 250 \text{ Kbps}$ .

**MCU services:** Here again due to a lower degree of encapsulation in the prototype, a number of peripheral services are exposed as simple MCU control services. Examples include starting and stopping MSP430 peripherals, for example, the timer, starting and stopping external ICs connected to the MSP430, for example, the MAX3096. As part of the MCU services, the HAL currently implements the timer clock  $f_t = 12 \text{ kHz}$  using the MSP430's ACLK internal oscillator. This is an unstable oscillator whose frequency drifts between  $9$  and  $15 \text{ kHz}$  depending upon the temperature and supply voltage. In spite of this drift, this clock has been chosen to run the timer because it is the lowest frequency oscillator on board and calibration is done periodically to correct for drifts in the frequency. For a production application using time period sensing, a stable clock needs to be used to run the timer.

In summary, the system level parameters set by the HAL are:

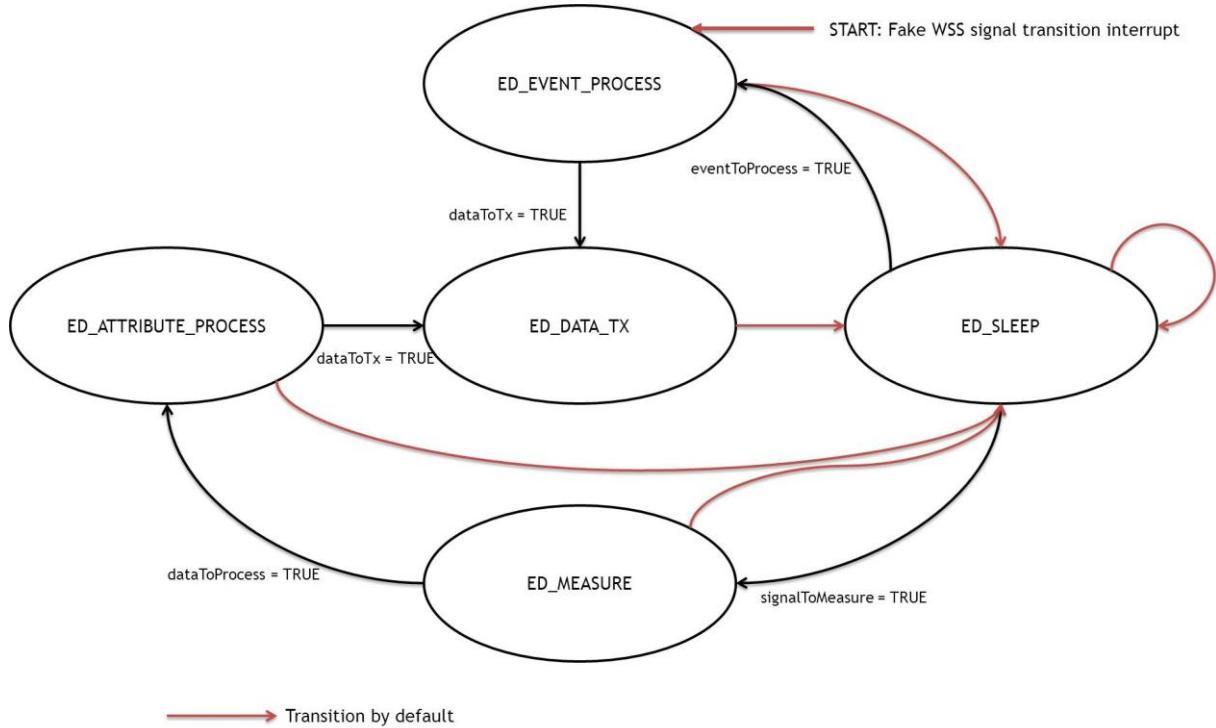
ED parameter	Value as implemented by the HAL
$r_{RF}$	$250 \text{ Kbps}$
$f_t$	$12 \pm 3 \text{ kHz}$

*Table 14 : ED parameters implemented by the HAL*

### 7.3.2 Application Logic Layer (ALL):

The ALL implements all node level operations by making use of the services provided by the HAL. These operations include node management functions like starting up and shutting down the node, joining and leaving the WSN, sensing the external stimuli and transacting data with other nodes. As mentioned earlier, the encapsulation of functions is not perfect in this prototype, and this is especially so in the AP node because of the lower amount of attention that was paid to it in this time-limited project. The style of ALL implementation also differs significantly between the ED and the AP. The ED ALL is implemented as a state machine, while the AP ALL is not. Detailed descriptions of the prototype ALL in the ED and AP are as follows:

**ALL in the ED node:** The ED node's ALL is implemented as a 5-state machine which is conceptually depicted by the following figure:



*Figure 27 : State machine implementation of ED ALL*

It can be easily seen that the state machine logic makes liberal use of transition variables, four to be precise, for five states in this implementation. A lower level of encapsulation is the primary reason for such a high number of transition variables and consolidation of these is definitely possible. In the prototype, however, the transition variables are considered as signals, allowing for a completely asynchronous style of operations. A brief description of each state is provided in this section while detailed information on the implementation of each state in the ED ALL can be found in Appendix C.

**ED\_EVENT\_PROCESS:** This state is responsible for processing asynchronous events, namely the WSS signal transition events, and signaling if the wheel is locked or unlocked. If this state decides to transmit an event, it populates the transmission buffer and signals a change to the state *ED\_DATA\_TX*; else it signals a transit to *ED\_SLEEP* by default.

**ED\_DATA\_TX:** This state is responsible for transmission of events and attributes to the AP. The state exposes a transmission buffer which any other state can use to populate a maximum of 1 message. When appropriately signaled *ED\_DATA\_TX* will transmit the message in the buffer to the AP. It should be noted that acknowledgement by the AP is not implemented in the prototype and therefore there is no re-transmission logic. This state always signals a transit to *ED\_SLEEP* once it finishes transmission.

**ED\_SLEEP:** This is the anchor state of the FSM, which is used to force the ED to sleep, so that it consumes almost no power while being in this state. The ED uses the HAL's sleep service while ensuring that all external peripherals are also turned off. Any interrupt would wake up the microcontroller from HAL's sleep, after which there could possibly be a state transition. An external event such as a WSS signal transition would trigger an immediate state transition to *ED\_EVENT\_PROCESS*. Internally, the sleep state signals a transit to *ED\_MEASURE*, after

periodically determining if the WSS signal is present, and this is an example of breaking encapsulation. The state signals a transit to itself by default (to continue sleeping) and the microcontroller wakes up only if the state machine transits out of *ED\_SLEEP*.

***ED\_MEASURE***: This state is responsible for making measurements which are periodic in nature. The state machine transits to this state from *ED\_SLEEP* if external signals (the WSS signal) are available for measurement. In order to increase modularity, a callback is provided by this state, which can be implemented by the ALL to setup measurements while teardown of the set measurements is usually done in an interrupt callback. If this state makes a successful measurement it populates a data buffer and signals a transition *ED\_ATTR\_PROCESS* to process the raw measured data. If it does not make a successful measurement, the FSM transits to *ED\_SLEEP* by default.

***ED\_ATTR\_PROCESS***: This state is responsible for processing the raw measured data and creating attributes out of it. If this state determines that the processed data constitutes a valid attribute, it populates the transmission buffer and signals a transition to *ED\_DATA\_TX*. If it determines that the processed data does not constitute a valid event or attribute it signals a transit to *ED\_SLEEP* by default.

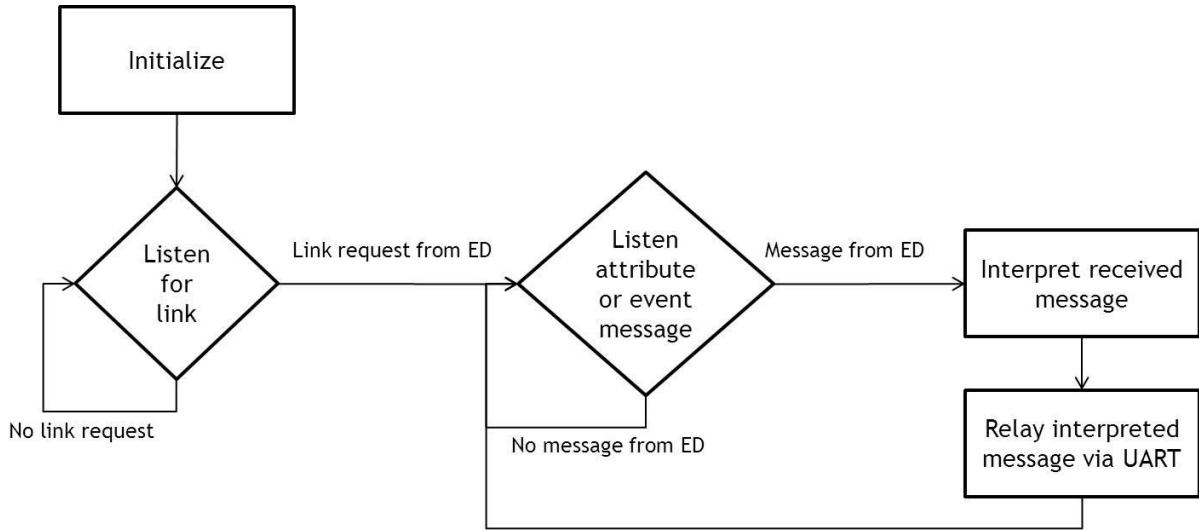
Before the state machine is started, the ALL initializes the node using the HAL's initialization services and a few other initialization operations. The final initialization operation before starting the state machine is faking a WSS signal transition interrupt. This emulated external event is recognized by the state machine, which transits to the *ED\_EVENT\_PROCESS* state to begin operations. The HAL provides a service to emulate the external interrupt (see Appendix B).

***Parametric choices***: The ALL logic has been implemented by choosing the parameter  $V_{min} = 3V$  and  $V_{max} = 3.6V$ , which consequently sets  $D_{peak} = 0.167\mu Ah$ . While this choice is not used to control any operation in the ALL, it acts as the containment limit. All operations have been implemented so that they do not cross the discharge limit. Additional parameters implemented by the ALL include an application payload of 2 bytes in the case of event transmission, and 3 bytes in the case of attribute transmission. This payload in combination with protocol overhead decides  $b_{PHY} = 16 - 17$  bytes.

ED parameter	Value as implemented by the ED ALL
$V_{minCap}$	3 V
$V_{maxCap}$	3.6 V
$D_{peak}$	0.1671 $\mu Ah$
$b_{PHY}$	16 – 17 bytes

*Table 15 : ED parameters implemented by the ALL*

***ALL in the AP node***: The ALL implementation in the AP node is quite flat with minimal amount of modularity. Most of the AP functionality is reused from the default AP source that is provided with the ez430-RF2500 kit. TI provides an open source example for a temperature sensor monitor (<http://www.ti.com/lit/zip/slac139>) whose AP code has been reused with adaptations to use the HAL. The AP ALL can be described by the following logic flow diagram.



**Figure 28 : Flow diagram depicting AP ALL logic**

The operation of the AP is quite straightforward and the above figure reflects its use as a relay node, where all messages received from the ED are relayed (via the UART) to the PM. It can also be seen that the AP is continuously listening, satisfying the minimal AP specification set in section 5.3.5.

#### 7.4 The AWSSN cluster library:

The AWSSN cluster library is a service abstraction definition within the ALL, for standardizing all transactions between nodes. The term cluster is borrowed from the Zigbee world where ‘Cluster’ refers to a standard definition of services that a Zigbee node can offer. Reflecting this definition, the AWSSN Cluster library defines the services that are offered by an AWSS node. Such a systematic definition of services is agnostic to a particular WSN standard and can completely encapsulate all AWSSN services. The position of the cluster library in an AWSS node’s operational stack is clearly visible in Figure 24.

Three elemental definitions in the cluster are:

1. Attributes – dynamic time bound values which are transacted more or less synchronously
2. Events – occurrences which are transacted asynchronously
3. Values – static information which are not transacted

Implementation details of the prototype cluster library are available in Appendix D. The definition in the cluster is, for example, used by the AP to interpret the messages (Figure 28) received from AP before it relays them via the UART stream.

Currently the definitions in the cluster are shared by the AP and the PM and there is a definite need for separation of definitions between them, something that can be considered for the future. While the AWSSN cluster definition in the prototype is germinal at best, it lays down the framework for extension. One can certainly imagine high level services defined as part of the cluster library to enable full-fledged service oriented operation.

Among the various system level parameters, the WSU sensing parameter  $w_t$  is set in the cluster library. This is an example of application level configuration of the system using the cluster library. As the system implementation grows, more and more parameters can be inducted into the cluster library and used in services as needed.

## Chapter 8 - Prototype characterization

The operation of the prototype sensor network is characterized and analyzed in this section. Most of the attention is paid to the WER and the ED, while the AP and the PM are put through a relatively lower level of scrutiny. Most characterization test cases are manual, since test automation was not attempted due to lack of proper equipment.

This chapter presents the characteristics of the WER in section 8.1, the characteristics of the ED node in section 8.2 and an end-to-end illustrative case of the prototype sensor network in section 8.3.

### 8.1 Characterizing the Wheel Emulation Rig (WER):

As described in 6.1, the WSS used in this project is emulated using the WER and therefore all electrical characterization has been done using the WER setup. The objective of this characterization is to determine the operational characteristics, such as the range of RPM speeds, the range of output voltage and range of harvestable power of the WSS. As described in the functional specification, WSS characteristics depends upon the fundamental parameter  $n_{teeth}$ , which in the wheel emulation rig has a value of:

$$n_{teeth} = 100$$

This value depends upon the choice of the pole wheel and is not a variable parameter in the WER test setup. Another fundamental parameter which the WSS characteristics would depend upon is the air gap ( $g_a$ ) between the WSS and the pole wheel. In an in-vehicle WSS setup,  $g_a$  is not a control parameter and is set to about 0.1 mm. On the other hand in the WER, while  $g_a$  happens to be controllable, it is usually set to a distance of:

$$g_a = 0.25 \text{ mm}$$

This setting is empirical to ensure that the signal voltage levels of the WSS output do not exceed the absolute maximum ratings of the various electronic components.

#### 8.1.1 Electrical characteristics of the Variable Reluctance (VR) magnetic sensor:

The source impedance of the VR sensor can be readily measured and is found to be:

$$\begin{aligned} R_{WSS} &= 1.2 \text{ k}\Omega \\ L_{WSS} &= 831.7 \text{ mH} \end{aligned}$$

The low inductance of the VR sensor renders it almost completely resistive in the sub-1 kHz range in which the WER operates. This means that the power factor is quite low and therefore reactive losses in energy harvesting are reasonably low. Therefore, the reactance of the VR sensor has been ignored for power calculations.

$$Z_{WSS} = 1.2 \text{ k}\Omega$$

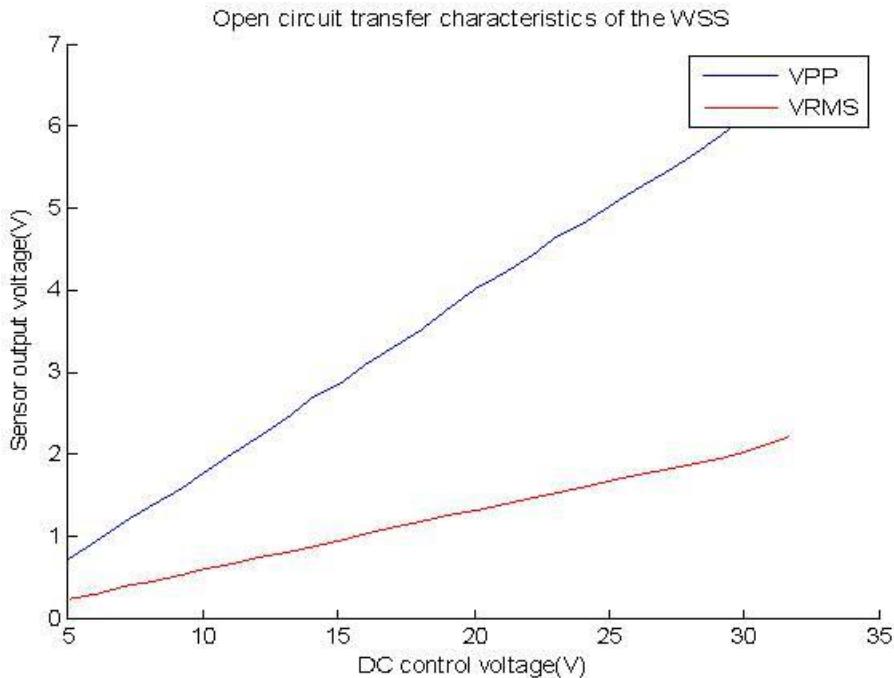
#### 8.1.2 WSS signal characteristics:

As described earlier, the WSS output signal is sinusoidal in nature, where both the amplitude and frequency depend upon the speed of rotation of the pole wheel. Since this speed is controlled by the applied DC voltage, this control voltage will be used as the reference parameter. While the control voltage may not be as good a reference as speed of rotation of the wheel, it is nevertheless used here

for simplicity. In addition, the RPS speed of the wheel (frequency of the WSS signal) can be represented in terms of wheel speed in *kmp/h* by assuming that the WER setup has a wheel of a certain diameter. In this case the diameter is assumed to be 1 *m*.

$$d_{wheel} = 1 \text{ m}$$

This assumption is practical where, for example in Volvo FH16 trucks, the wheel diameter can be anything between 996 *mm* and 1012 *mm*. For a particular  $g_a = 0.25 \text{ mm}$ , the WSS signal characteristics is essentially given by the transfer characteristics of the WER. By varying the DC control voltage of the rig, the speed of rotation of the pole wheel is varied. For different speeds of rotation, the output signal is measured to determine its voltage and frequency. These transfer characteristics in effect decide the range  $f_{min}, V_{min}$  to  $f_{max}, V_{max}$  that has to be measured by the AWSS node and, additionally, the range of power that is available for harvesting. The transfer characteristics of the WSS are given by the following plot:



**Figure 29 : WER transfer characteristics - signal voltage**

As seen in the plot, the variation in output voltage is a reasonably linear function of the control voltage, as one would expect from an inductive sensor, in accordance with the Maxwell-Faraday induction law. From the measured transfer characteristics, the average power available for energy harvesting using ( 5 ) is calculated to be:

$$P_{min} = 11.4 \mu\text{W}$$

$$P_{max} = 1 \text{ mW}$$

It should be noted that this range has been measured with  $g_a = 0.25 \text{ mm}$  and with a less than perfect WER assembly. For lower values of  $g_a$ , higher open circuit voltages and hence higher harvested average power is possible. In order to calculate the statistical average power that is available for harvesting, using the RT process model of wheel rotation, a choice of  $\lambda_{transition} t = 0.3$  is made, which yields  $p_{lock} = 0.23$  according to ( 8 ). This lock proportion is quite realistic because trucks tend to be operated for significantly higher proportions of time compared to when they are stationary. If the

wheel remains rotating for almost 80% of the time, the average amount of power for uniformly distributed speeds according to ( 7 ) is

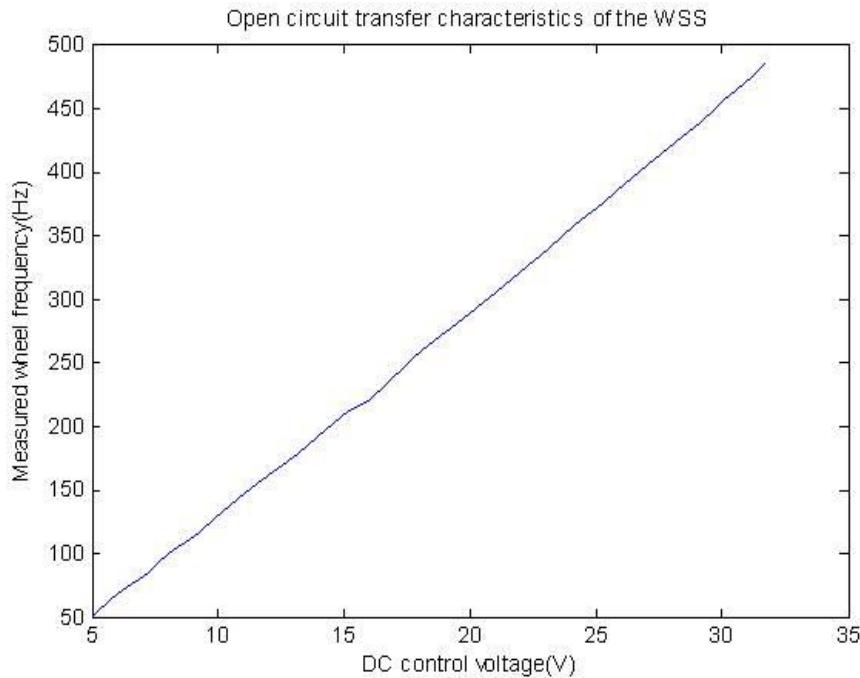
$$P_{mean} = 0.39 \text{ mW} \quad ( 39 )$$

The imperfections in the WER, such as wheel run-off and sensor run-off, can be expressed in terms of the non-linearity of change in the WSS signal voltage with the change in control voltage. For the above measurements,

$$\frac{dV_{PP}}{dV_{DC}} \cong 226 \text{ mV} \pm 45 \text{ mV}$$

$$\frac{dV_{RMS}}{dV_{DC}} \cong 75 \text{ mV} \pm 12 \text{ mV}$$

The above derivatives are expressed in terms of the mean and standard deviation of the discrete derivative. The non-linearity happens to be quite significant at 19.9% and 16% respectively for the PP and RMS cases. The output signal frequency characteristic is depicted by the plot below:



**Figure 30 : WER transfer characteristics - signal frequency**

Here again the change in frequency and its non-linearity are given by:

$$\frac{df}{dV_{DC}} = 16.3 \pm 1.9 \text{ HzV}^{-1}$$

This represents a non-linearity of 11.7% which is also quite significant.

The reason for this non-linearity is that the WER is hand-tooled and hand-assembled in a machine workshop, and is not constructed to meet with truck installation standards. The variation in distance between the teeth and the sensor and the sensor run-off - the variation in relative position between the

WSS and the pole wheel - contribute to such non-linearities. Having knowledge of these non-linearities can help in making statistical approximations.

From the above set of measurements, we have

$$f_{min}, V_{min} = 51 \text{ Hz}, 732 \text{ mVPP} = 51 \text{ Hz}, 234 \text{ mVRMS}$$

$$f_{max}, V_{max} = 485 \text{ Hz}, 6.7 \text{ VPP} = 485 \text{ Hz}, 2.2 \text{ VRMS}$$

For an assumed truck wheel of diameter 1 m, using (3) we have

$$s_{min} = 5.77 \text{ kmph}$$

$$s_{max} = 54.85 \text{ kmph}$$

It should be noted that wheel speed in a truck is usually in the 10 – 100 kmph range, while the WER spans half of this range.

The change in output frequency with output voltage is also of importance as it helps in calculating the system parameter  $V_1$ . From Figure 29 and Figure 30, we have:

$$\begin{aligned}\frac{df}{dV_{pp}} &= 74.4 \pm 14.5 \text{ HzV}_{pp}^{-1} \\ \frac{df}{dV_{RMS}} &= 222.6 \pm 34 \text{ HzV}_{RMS}^{-1}\end{aligned}$$

This represents deviations of 19.5% and 15.2% respectively. The parameter  $V_1$  can be calculated as

$$V_1 = \frac{1}{\left(\frac{df}{dV}\right)} \cong 13.3 \text{ mV}_{pp} \text{Hz}^{-1} \text{ or } 4.5 \text{ mV}_{RMS} \text{Hz}^{-1}$$

To summarize, the WSS parameters as implemented by the WER are as follows:

<b>WSS parameter</b>	<b>Value as implemented by the WER</b>
$d_{wheel}$	1 m (assumption)
$n_{teeth}$	100
$s_{min}$	5.77 Kmph
$s_{max}$	54.85 Kmph
$V_1$	$13.3 \text{ mV}_{pp} \text{Hz}^{-1}$ or $4.5 \text{ mV}_{RMS} \text{Hz}^{-1}$
$f_{min}$	51 Hz
$f_{max}$	485 Hz
$t_{min}$	2.1 ms
$t_{max}$	19.6 ms
$V_{min}$	732 mVPP or 234 mVRMS
$V_{max}$	6.7 VPP or 2.2 VRMS
$Z_{WSS}$	1.2 kΩ
$P_{min}$	$11.4 \mu\text{W}$
$P_{max}$	1 mW
$P_{mean}$	0.39 mW

*Table 16 : WSS parameters as implemented by the WER*

**A note on the signal characteristics data:** While parsing the WER transfer characteristics data it is necessary to understand the influence of  $g_a$  in deciding the amplitude of the signal. The value of  $g_a$ , while adjustable in the WER, cannot be considered reliable because of the quality of the experimental setup. Since a smaller portion of the effort was invested in making the WER, the quality of the  $g_a$  adjustment setup is only reasonable, which makes accurately setting  $g_a$  quite a challenge. For the measurement data presented in earlier sections,  $g_a$  was set to 0.25 mm and was verified to be so, but during consequent measurement operations,  $g_a$  was not similarly regulated, which means that signal amplitude data is not the same as what is presented here. Hence to reiterate, signal amplitude values in the WER have to be taken with a pinch of salt because of possible variations in  $g_a$ .

### 8.1.3 Behavioral model of the WSS:

In order to gain more insight into the WSS, a behavioral model was created using COMSOL Multiphysics. COMSOL provides options for modeling the WSS setup in both two and three dimensions. While 3D models mirror the reality best, they require a significantly higher amount of computational resources. In addition, a key requirement of the WSS behavioral model is to model a rotating pole wheel. COMSOL version 4.2a used in this project does not have in-built capabilities to model 3D rotation.

With this limitation in mind, a decision was made to model the WSS in 2D using the Rotational Machinery Magnetic (*rmm*) physics interface. The *rmm* interface has built-in support for modeling 2D rotation about a certain point with a certain angular velocity. The following figure depicts the WSS COMSOL behavioral model:

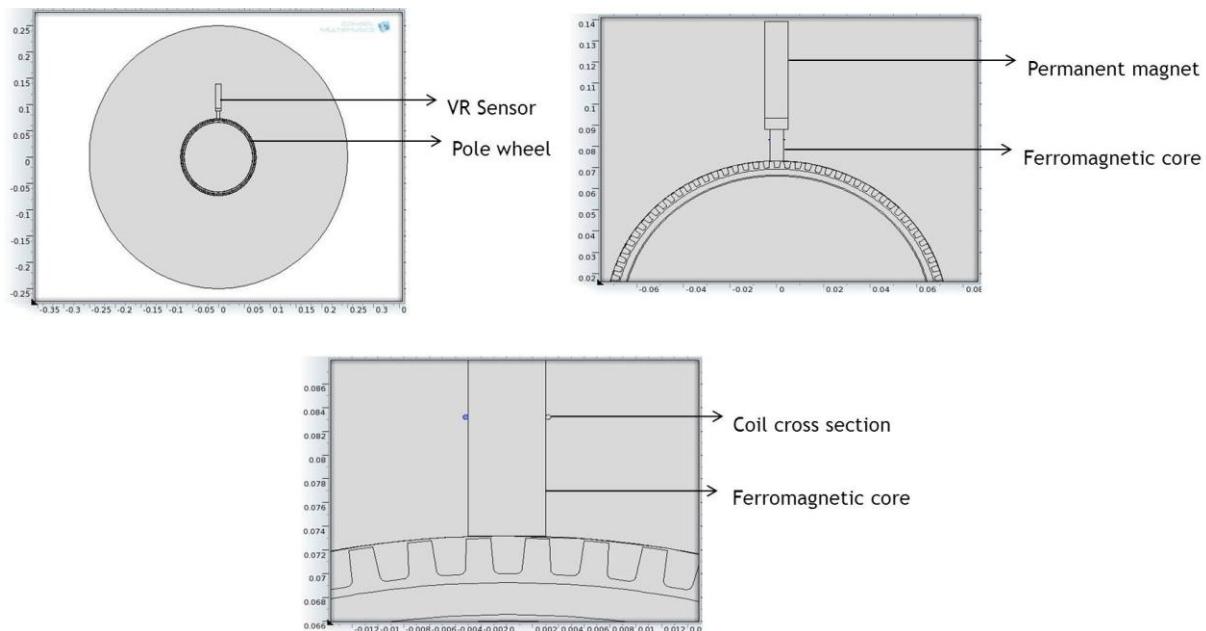


Figure 31 : WSS COMSOL 2D model

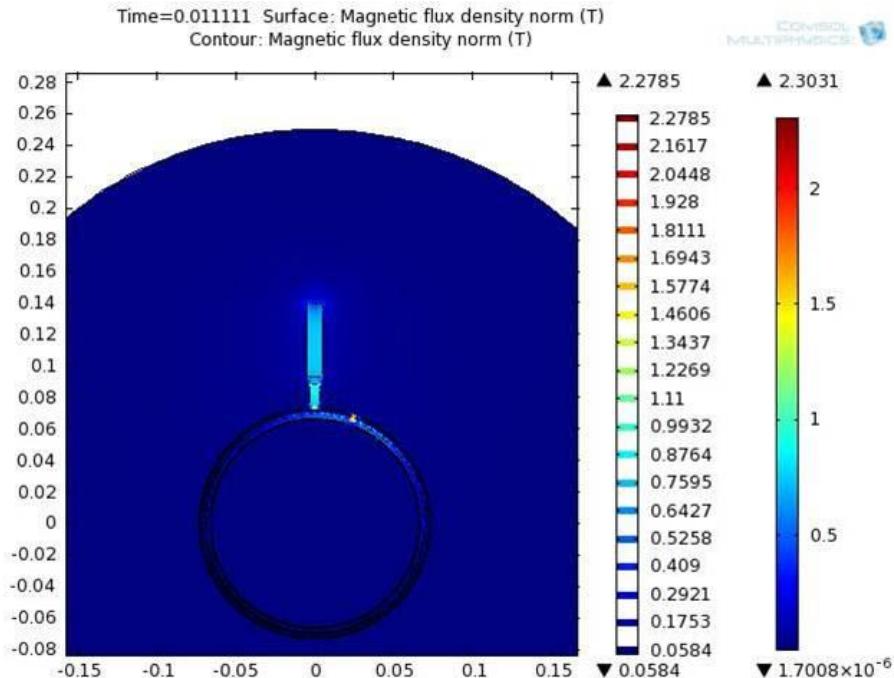
All domains in the model, apart from the VR Sensor and the pole-wheel, are modeled to be air. The coil is modeled by two circles representing the wire cross-section on either side of the ferromagnetic core. It should be noted that the pole-wheel geometry is derived from a CAD geometry file while the VR Sensor geometry was hand drawn. These two separate models were imported and then assembled to form the final WSS 2D model. The ‘Ampere’s law’ electromagnetic interface is assigned to all domains but the magnet, to simulate Maxwell’s laws. A separate Ampere’s law setup is assigned to the domain of the magnet where its remnant magnetic flux density was specified. In addition a ‘Prescribed

'Rotational Velocity' interface is assigned to the pole wheel, where its rotational velocity is specified. The model is described by an exhaustive list of parameters, the most pertinent of which are listed here.

Parameter	Value
Radius of the magnet	1 cm
Length of the magnet	4.6 cm
Radius of the ferromagnetic core	0.5 cm
Length of the ferromagnetic core	2 cm
Relative permeability of ferromagnetic core	4000 ( <i>Soft iron</i> )
Radius of coil cross section	0.2 mm
Number of turns in the coil	500
Air gap between the VR Sensor and pole wheel	0.1 mm
Number of teeth in the pole wheel	100
Relative permeability of pole-wheel material	4000 ( <i>Soft iron</i> )
Wheel speed	5 RPS

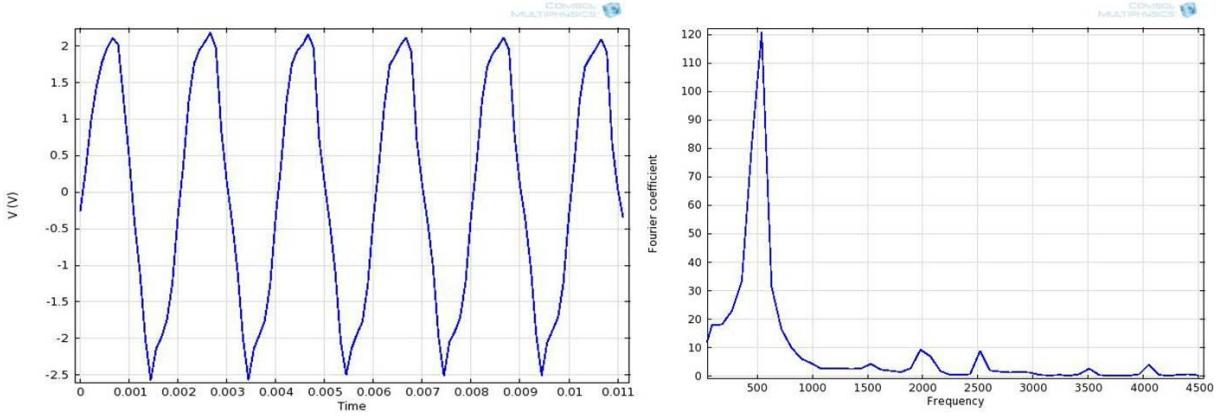
*Table 17 : WSS COMSOL model parameters*

The above parameters are empirical and determined using cursory measurements. Detailed information about the properties of the VR sensor was not shared by the sensor manufacturer due to intellectual property concerns, a fact which brought the COMSOL modeling activity in the project to a standstill. However, the model is suitably constructed and parameterized for further development. With the parameters in Table 17, a snapshot of the simulation results of the WSS COMSOL 2D model is presented below.



*Figure 32 : Simulated result of the magnetic flux density*

One must note that in the figure above the highlighted tooth in the plot of the simulation results, due to the method of visualization in COMSOL, is actually in front of the VR sensor. A snapshot of the simulated voltage signal and its frequency spectrum:



**Figure 33 : WSS COMSOL model simulated results simulation results**

The 2 V, 500 Hz signal corresponds to rotational speed of 5 RPS of a 100 teeth pole-wheel, subject to the parameters defined in Table 17.

## 8.2 Characterizing the ED/AWSS node:

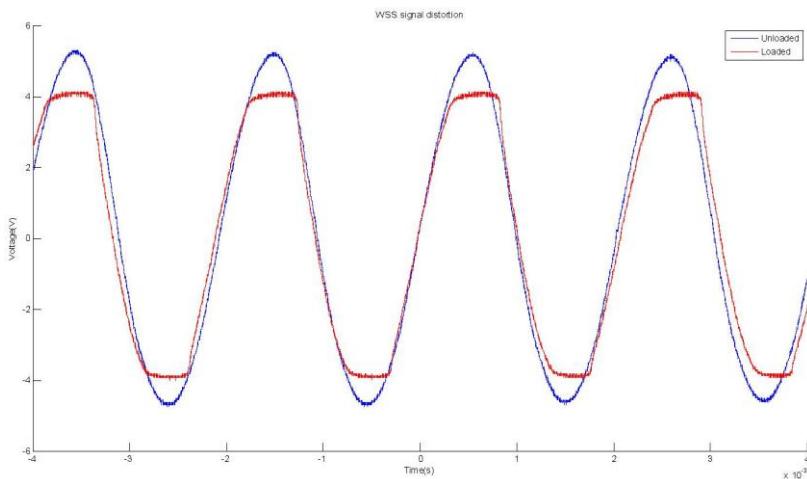
The operation of the ED node which implements the AWSS is characterized and analyzed in this section. The procedure for characterizing each of the operations is presented in Appendix F.

### 8.2.1 Characterization pre-conditions:

The system level parameters have their values configured by different modules, and a summary of all the parameters with their default values have been provided in the parametric register in Appendix E. These default values are used as pre-conditions for all the following characterization test cases.

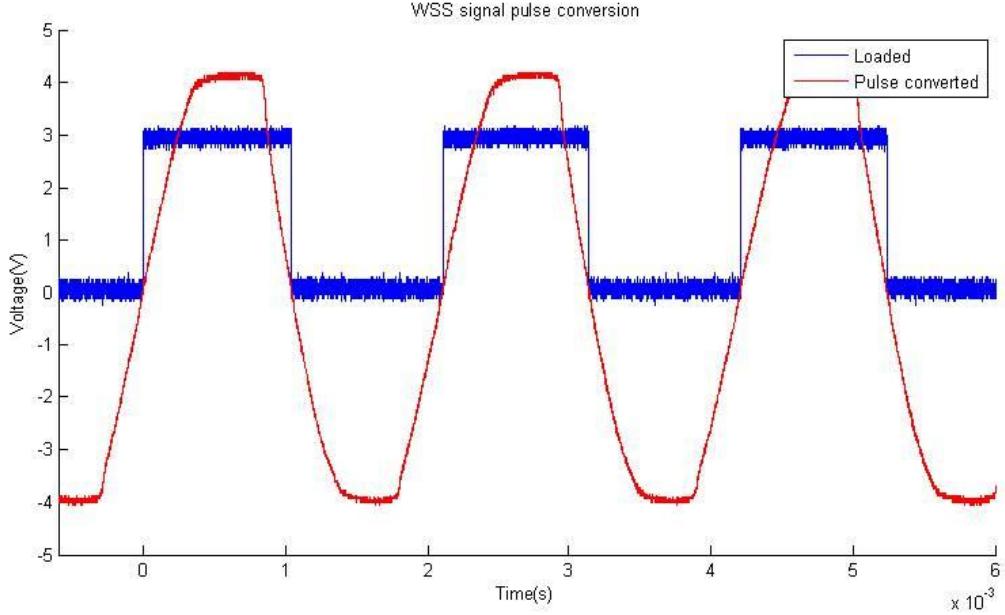
### 8.2.2 ED Hardware Layer signal characteristics:

Signal characteristics of the MCU and the RF transmitter, which form the heart of the Hardware Layer, are provided in section 6.2, while the signal characteristics of the WSS pulse converter and the rotational state sensor are presented here. The WSS output signal is routed to the EHU, which loads the WSS with matched impedance, which distorts the signal, and it is this distorted WSS signal that is provided to the rest of the Hardware Layer. In order to illustrate the distortion characteristics, the test procedure described in Appendix F is followed, giving the following results.



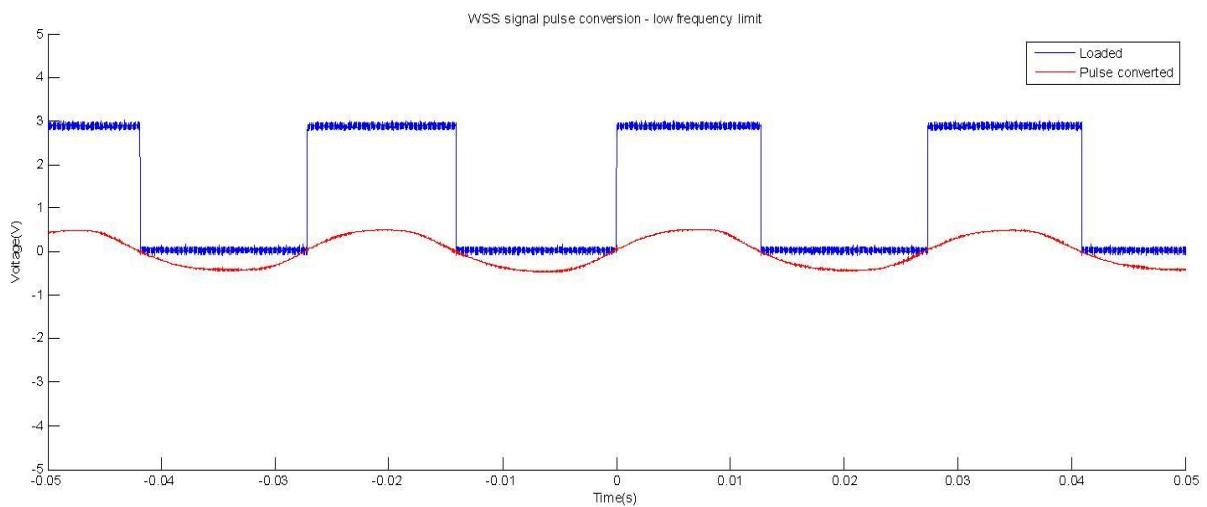
**Figure 34: WSS signal distortion due to loading**

The plot in Figure 34 illustrates the distortion in the signal due to loading by the EHU, i.e. the CBC-Eval-09. As measurements show in Appendix F, there is minimal distortion ( $35.29 \times 10^{-3}^\circ$ ) in the phase since reactive elements are not introduced by the EHU. It should be noted in the above figure, the waveforms are captured at different points of time, since they are the voltage at the same circuit node at different loading conditions. Therefore phase information should not be interpreted from the above figure, but from the measurements that are presented in Appendix F.



**Figure 35 : WSS signal pulse conversion – high frequency case**

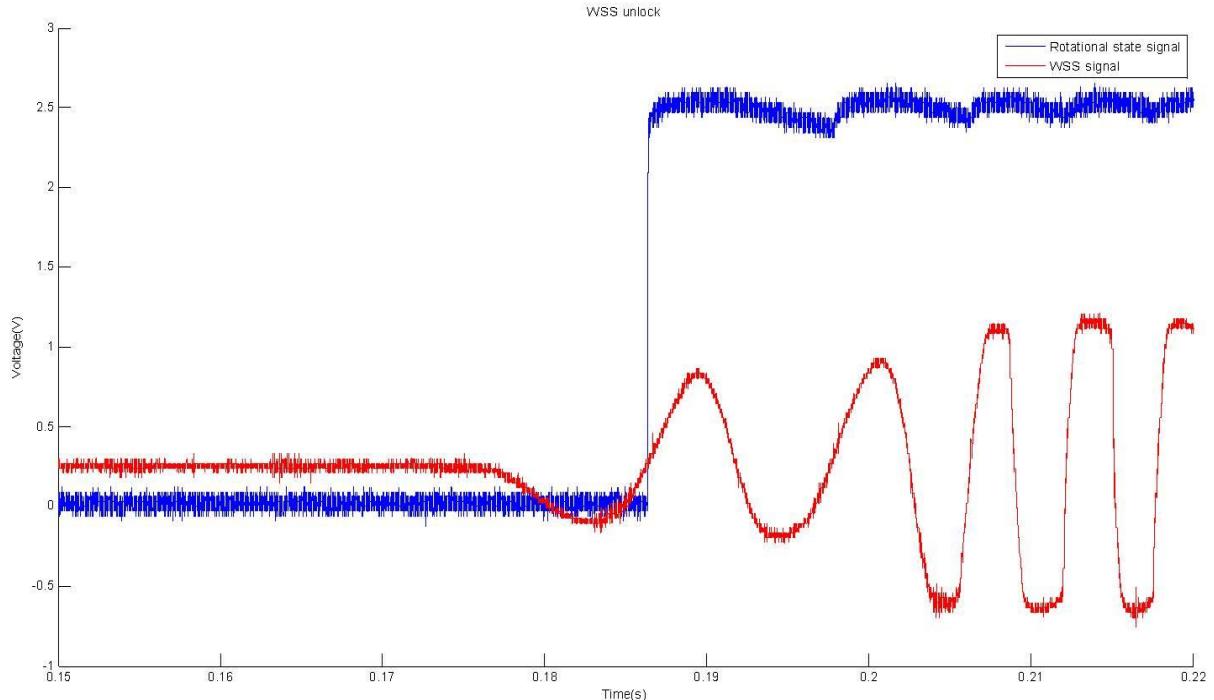
The pulse converted signal that is used for attribute sensing is shown in Figure 35. The measured difference in phase is less than  $1^\circ$ , consistent with the stated propagation delays of  $10\text{ ns}$  in [50], which is acceptable since the minimum signal period is around  $2\text{ ms}$ . The signal characteristics in a low frequency case are as follows, where the phase difference is still less than  $1^\circ$ .



**Figure 36 : WSS signal pulse conversion - low frequency case**

The above figure also shows the drop in voltage of the WSS signal to less than 1 V at low rotational speeds. This is a demonstration of the use of the pulse converter circuit, which maintains a steady 0 – 3 V pulse despite the variations in the WSS signal amplitude.

The signal characteristics of the rotational state sensor, which determines whether the wheel is rotating (unlocked) or not (locked), is presented next. The lock to unlock state transition is depicted by the following plot, beginning with a high speed unlock which is in effect a 0 –  $\sim 50 \text{ kmph}$  transition in a few seconds.

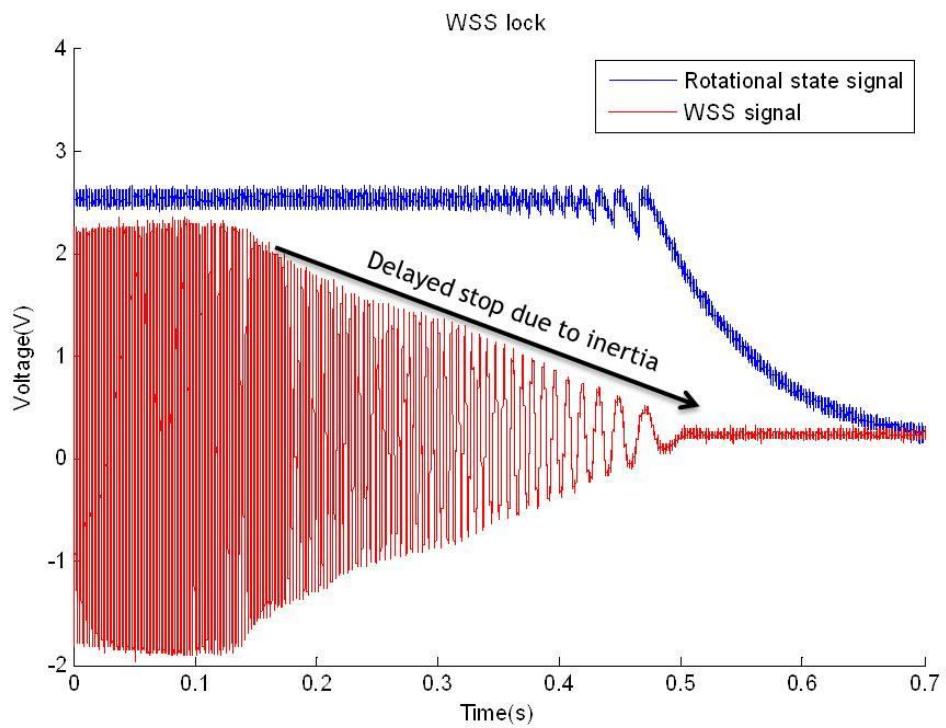


**Figure 37 : WSS unlock event - high speed**

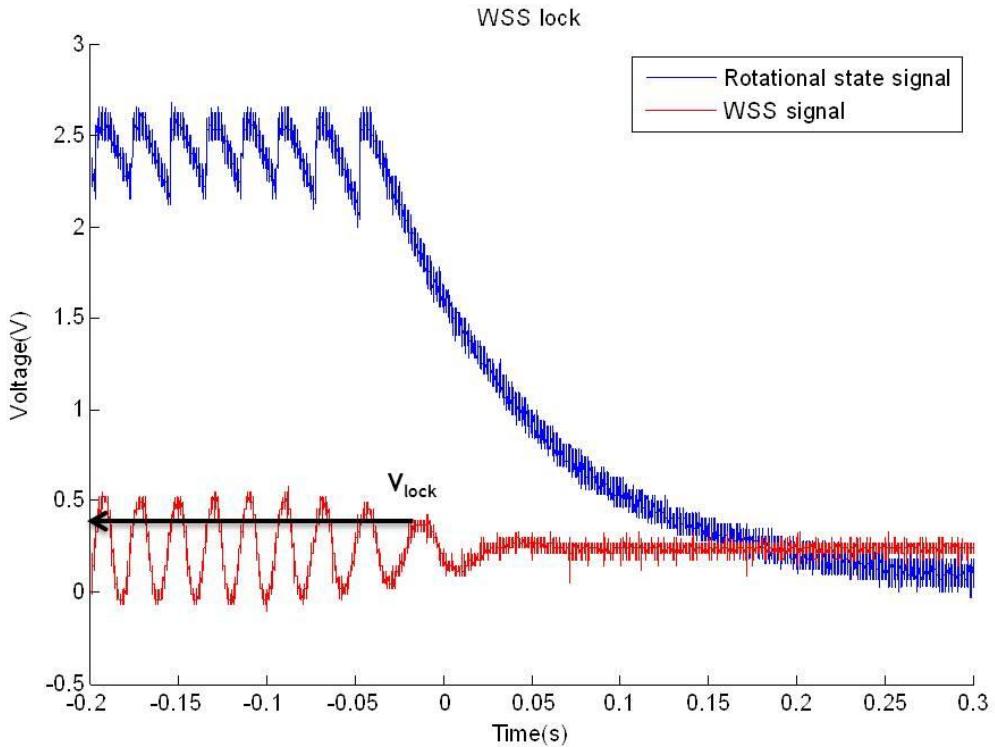
As seen in the plot above, the rise in the rotational state signal is quite instantaneous and is measured to be close to  $3.7 \mu\text{s}$ , reflecting the  $t_{rise}$  chosen by design (Table 14). The signal rises when the WSS signal crosses about 0.4 V and charges via a forward biased diode which explains the short charging time. The ripples in the state signal are clearly visible with amplitude of close to 0.2 V.

A corresponding high speed lock event, a  $\sim 50 - 0 \text{ kmph}$  drop, is shown in Figure 38. The lock event, as opposed to the unlock event, takes some time due to rotational inertia. However in the prototype, the characteristic of interest is the fall time of the rotational state signal, since this transition acts as the interrupt to signal the lock event. The measured worst case fall time in this case is quite high at 99.3 ms, which reflects  $t_{fall}$  of the rotational state sensor chosen by design (Table 14). However, it has to be noted that this discharge rate is configurable and can be made to fit the requirements of a potential production application. The slow discharge can be sped up at the cost of increased ripples in the state sensor output.

A smooth transition of the WSS signal can be seen in the low speed lock event in Figure 39, where the brake is applied when the wheel is rotating at  $< 5 \text{ kmph}$ . This transition makes the measurement of the lock threshold simpler, where a  $V_{lock}$  of close to 0.45 V can be easily observed. This observed value is consistent with the observation in Figure 37.



*Figure 38 : WSS lock event - high speed braking*



*Figure 39 : WSS lock event - low speed braking*

It should be noted that the rise and fall times have been measured to qualify the worst case scenario. According to [47], the MSP430F2274 has worst case Schmitt trigger levels of 2.25V for the rising

edge and  $0.75\text{ V}$  for the falling edge. In the test,  $80 - 20\%$  limits are chosen, which translates to  $2.4\text{ V}$  for the rising edge and  $0.6\text{ V}$  for the falling edge.

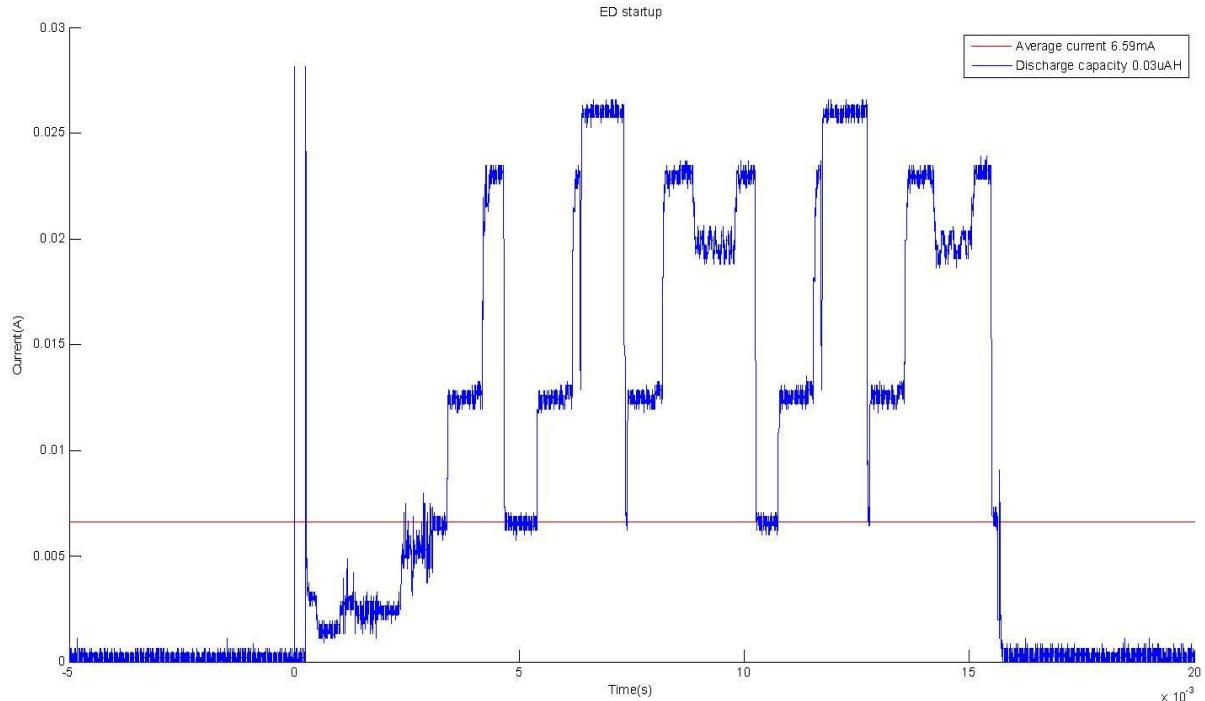
Hardware Layer parameter	Measured value
$t_{rise}$	$3.7\text{ }\mu\text{s}$
$t_{fall}$	$99.3\text{ ms}$
$V_{lock}$	$0.45\text{ V}$

*Table 18 : Experimental measurement of Hardware Layer parameters*

### 8.2.3 ED Discrete operations:

Functionality of the ED can be broken down into a set of discrete operations such as startup, event transmission, attributes transmission, etc., which simplifies node characterization. The discrete operations in the prototype ED and their characteristics, especially their peak power consumption, are as follows. The test procedures can be found in Appendix F.

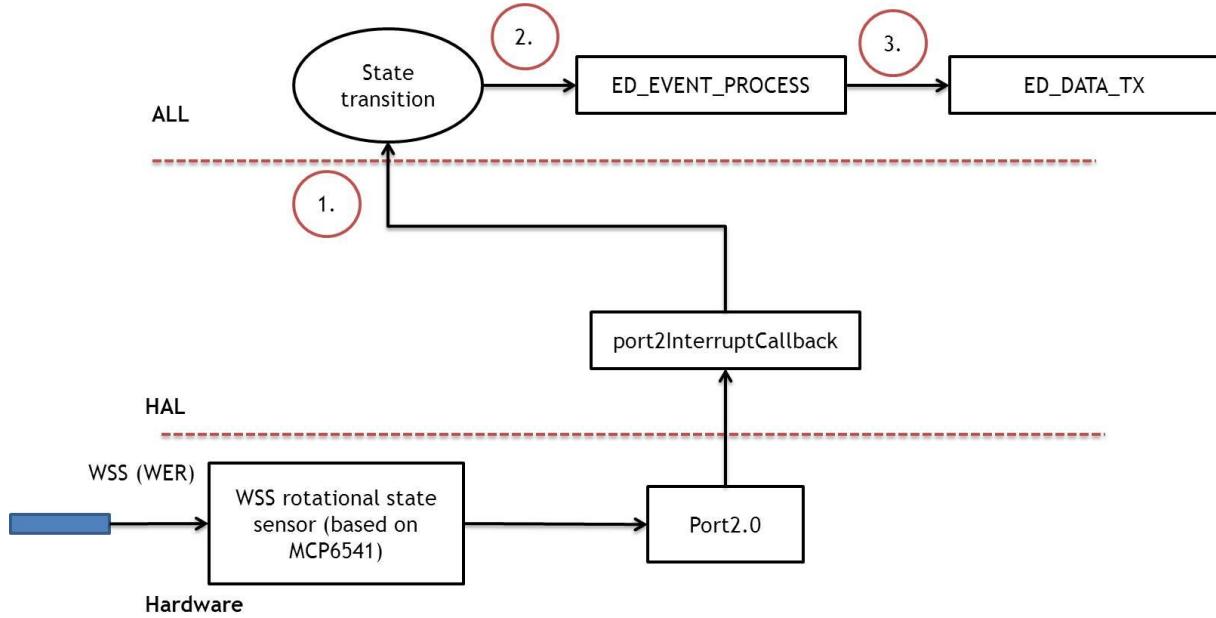
**ED startup:** When the ED starts up, the MSP430 and the CC2500 goes through initialization, while the MAX3096 is put to sleep. The current consumed by the startup operation:



*Figure 40 : ED startup with average current draw of  $6.59\text{ mA}$  and discharge capacity of  $0.03\text{ }\mu\text{Ah}$*

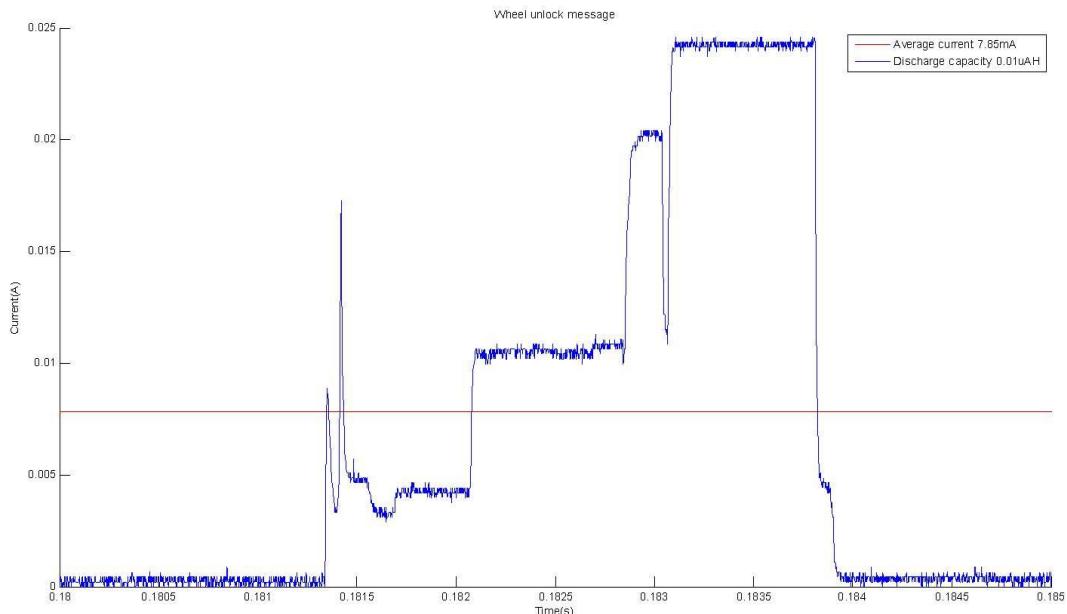
It can be seen that startup of the ED lasts for about  $15\text{ ms}$  during which it draws an average current of  $6.59\text{ mA}$ . This translates to a capacity of  $0.03\text{ }\mu\text{Ah}$  discharged from the  $1000\text{ }\mu\text{F}$  output capacitor, which is well below  $D_{peakCap}$ . This extremely low startup power is one of the biggest advantages of the ez430 sensor board.

**ED event processing:** The event sensing and transmission operation is depicted by the following figure.



**Figure 41 : ED event sensing and transmission**

The transition of the rotational state sensor signal acts as an interrupt trigger for the event sensing operation. A rising edge of this signal triggers the transmission of an unlock event, while the falling edge triggers a lock event. The current consumption for lock/unlock events:



**Figure 42 : Wheel unlock event transmission with average current draw of 7.85 mA and a discharge capacity of 0.01 μAh**

It can be seen that the lock event transmission lasts between 2.5 – 3 ms and draws an average of 7.85 mA, translating into a discharge of 0.01 μAh. The current drawn in the case of a lock event is identical to the unlock event because they differ only in message content. We therefore have:

Event sensing parameter	Measured value
$t_{eventActive}$	3 ms
$I_{eventActive}$	7.85 mA
$D_{eventActive}$	0.01 $\mu$ Ah

Table 19 : Experimental measurement of event sensing parameters

**ED attributes processing:** Periodically, the ED senses the pulse converted WSS signal, measures it and transmits the measured result. The attributes processing operation is depicted by the following figure.

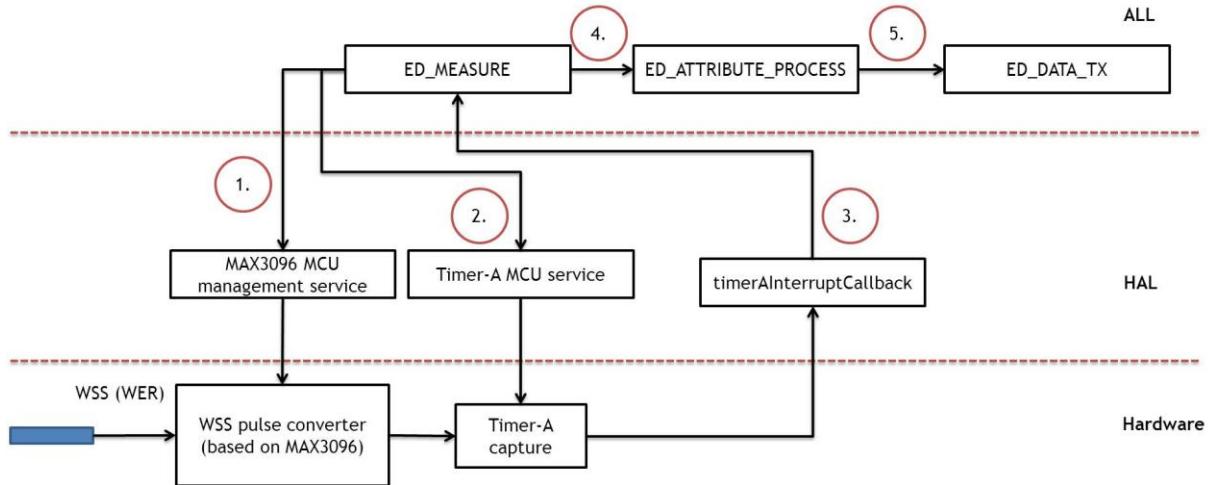
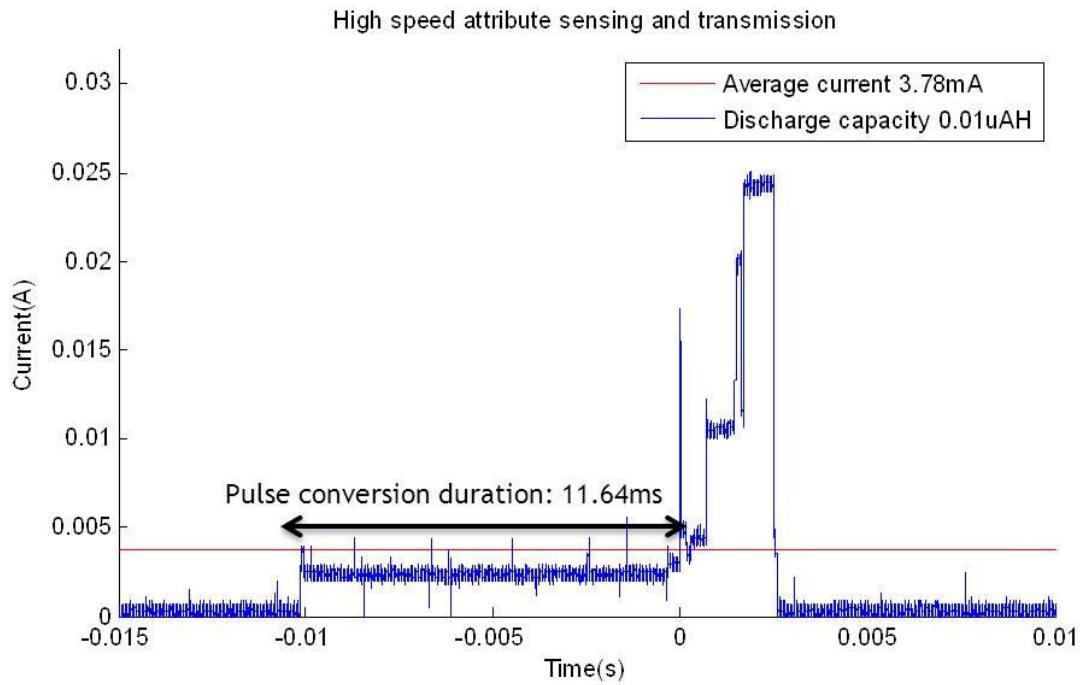
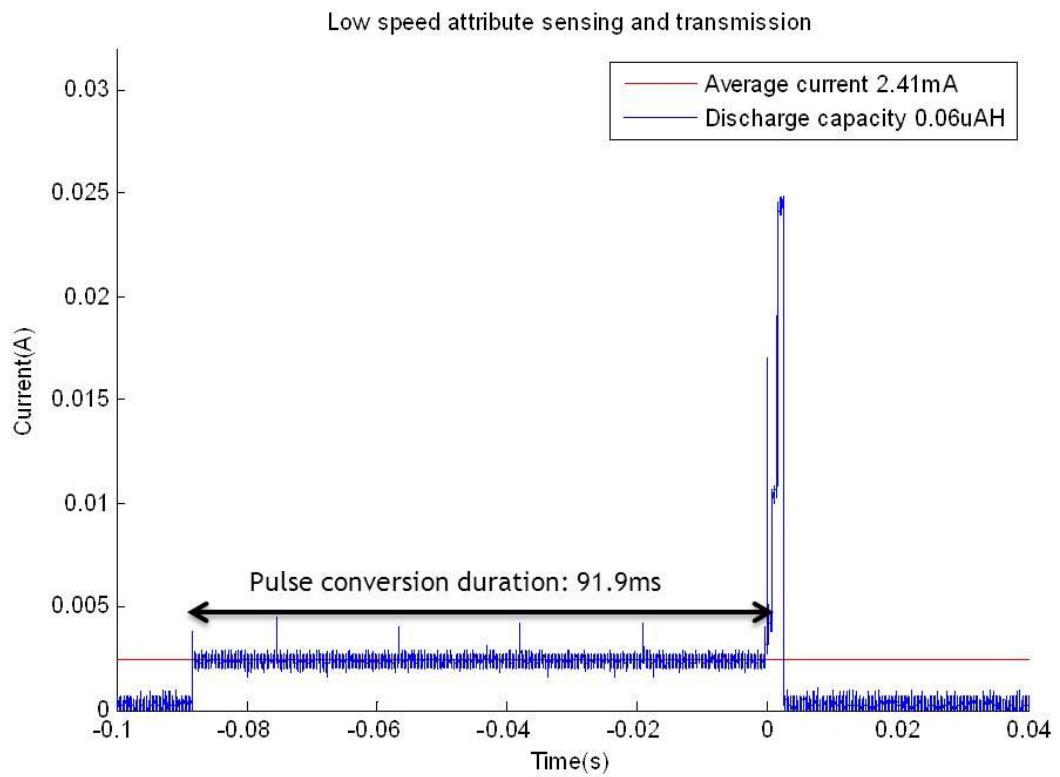


Figure 43: ED attribute sensing and transmission

The Hardware Layer is essentially used as a rising edge sensor, where the specified number of rising edges ( $w_t$ ) are captured, to calculate the average time lapse between two rising edges. This count is then transmitted to the AP, from which the wheel speed is derived. It must be noted here that the pulse converter remains active for  $w_t$  number of periods and therefore the total active time depends upon the frequency of the signal being sensed. If a high frequency signal is sensed, the pulse converter is active for a lower amount of time compared to sensing a low frequency signal. Power consumption at these extremes with the WER is presented in Figure 44 and Figure 45. The pulse conversion duration of 11.64 ms, pointed out in Figure 44, represents the lower end of steady current drawn by the pulse conversion circuitry. The higher end of the limit, measuring 91.9 ms can be seen in Figure 45. It can be seen that the active duration has increased proportionally with the reduction in speed, as theorized in section 5.3.4.



**Figure 44 :** Attribute sensing and transmission in a high speed case with an average current of 3.78 mA and a discharge capacity of 0.01  $\mu$ Ah



**Figure 45 :** Attribute sensing and transmission in a low speed case with an average current of 2.41 mA and a discharge capacity of 0.06  $\mu$ Ah

From the above measurements, we have:

Attribute sensing parameter	Minimum	Average	Maximum
$t_{attActive}$	14.7 ms	54.7 ms	94.8 ms
$I_{attActive}$	2.41 mA	3.09 mA	3.78 mA
$D_{attActive}$	0.01 $\mu$ Ah	0.035 $\mu$ Ah	0.06 $\mu$ Ah

Table 20 : Experimental measurement of attribute sensing parameters

The average values have been calculated by considering the uniform speed distribution model. It should be noted that the values across a column do not have a one-one relation. The one-one time and current values can be found in Figure 44 and Figure 45.

#### 8.2.4 ED Duty cycle calculation:

Now that the values of  $t_{attActive}$  and  $t_{eventActive}$  have been measured and found to be in compliance with theory, the average duty cycle of operation can be calculated using ( 32 ). In order to model the locking probability, the RT process model, defined in ( 8 ), is used with a reasonable value for  $\lambda_{transition}$ . With the initial state assumption of  $X(0) = 1$  (due to reasons presented in ( 8 )), a modeling choice of  $\lambda_{transition} t = 0.3$  is made. With this, according to ( 8 ) we have  $p_{lock} = 0.23$ . As explained earlier, this choice in modeling  $p_{lock}$  is made to reflect the fact that truck operators will try to keep trucks running most of the time, which means that the wheel would be rotating most of the time. However a weakness of the RT process model is revealed here, where the transition rate is heavily influenced by the choice of the observation window.

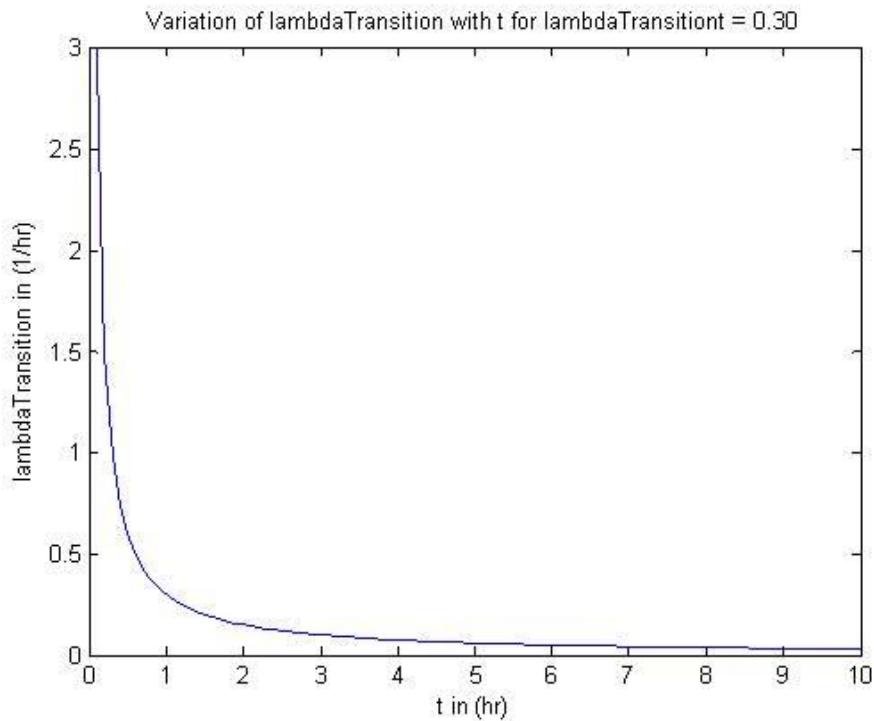


Figure 46 : Variation of  $\lambda_{transition}$  with  $t$  for  $\lambda_{transition} t = 0.3$

It can be easily seen that high transition rates can only be modeled for extremely short observation windows. Considering the long lifetime of a rotating wheel, a reasonable observation window of two hours ( $t = 2$ ) has been chosen in this model. As seen in Appendix E, a default value of  $t_{attInterval} = 5$  s is used in the ED characterization test cases. Using ( 32 ), for the two hour time window, we have  $d = 0.0085$ . The ED therefore operates with a duty cycle of about 0.85% on average for a 5 s

transmission interval. As reasoned in section 5.3.4, this duty cycle can be reduced further if the attribute sensing is made passive, reducing  $t_{attActive}$  to 0.

The theoretical limits of  $t_{attInterval}$  and  $d$  for this operational profile can now be calculated. According to (18), for worst case attribute transmission with  $D_{attActive} = 0.06 \mu Ah$  and  $t_{attActive} = 94.8 ms$ ,  $V_{cap}$  is discharged to 3.39 V. In order to recharge  $V_{cap}$  to  $V_{maxCap} = 3.6 V$ , according to (17), a charging interval of at least 0.72 s is needed. This worst case recharge interval is safe enough to sustain node operations at the lowest speeds of the WER and can be considered as the minimum usable attribute transmission interval. Based on this transmission interval limit, duty cycles for various transmission intervals can now be calculated using (32). A pictorial representation of the transmission interval limit and the associated duty cycle limit is presented in Figure 47.

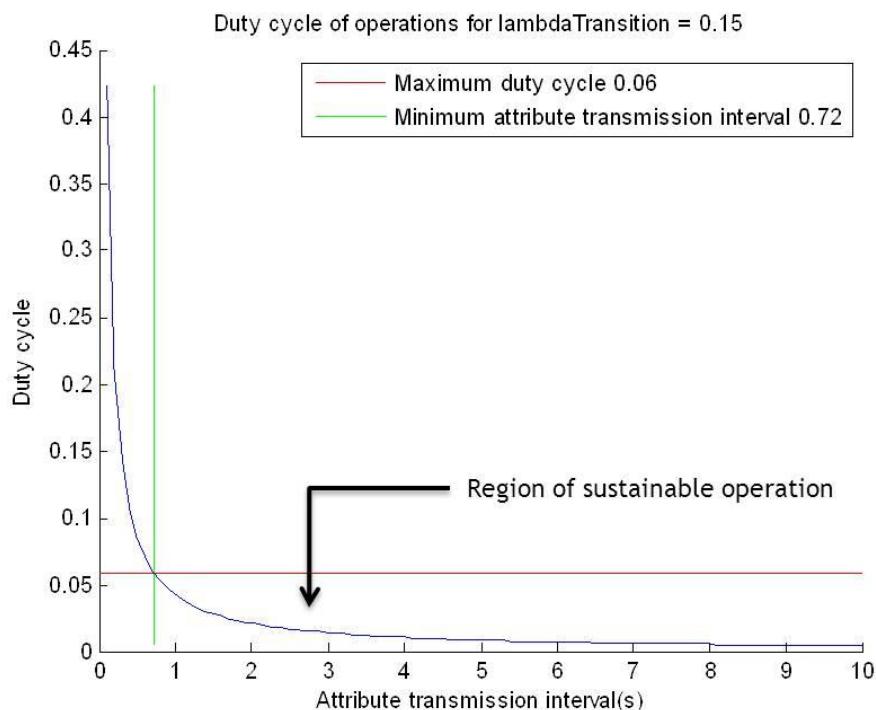
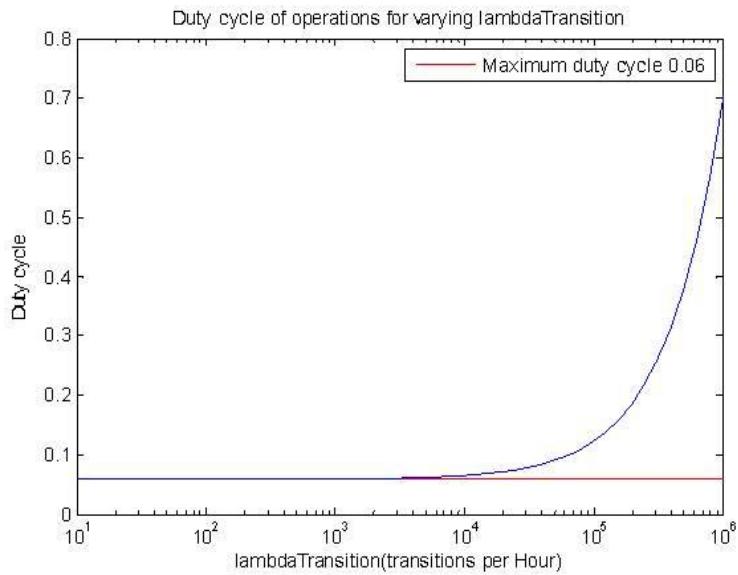


Figure 47 : Plot of duty cycle as a function of attribute transmission interval with  $\lambda_{transition} = 0.15$

It can be seen from the Figure 47 and (32) that both  $d = 0$  and  $t_{attInterval} = 0$  are asymptotes. While a  $d = 0$  asymptote is valid, the  $t_{attInterval} = 0$  is an illusion that is created by the approximations  $t_{attActive} \ll t_{attInterval}$  and  $t_{eventActive} \ll t_{eventInterval}$  used in (32). If these approximations are violated, (32) will not hold. Violating these approximations in design is not advisable because  $d$  would dramatically increase and will come close to 1, since observation periods will become equal to active periods. Also, while not apparent in (32), there cannot be a practical case where  $t_{attInterval} < t_{attActive}$  and  $t_{eventInterval} < t_{eventActive}$ .

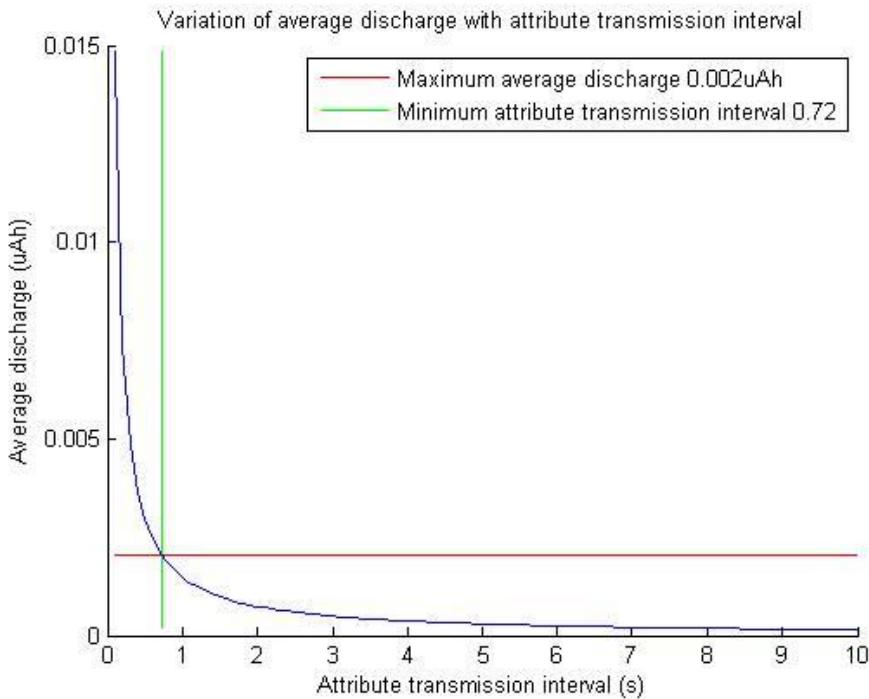
With transmission intervals higher than the worst case limit of 0.72 s, the duty cycle stays under 5.88%. The effect of event transitions on the duty cycle is minimal as a transmission event lasts for a mere 3 ms and is relatively infrequent. This is also evident from the plot in Figure 48, which shows that the duty cycle limit is exceeded only if the number of transitions per hour crosses 1000.



**Figure 48: Effect of event transitions on the duty cycle of operations**

### 8.2.5 ED charge and power consumption:

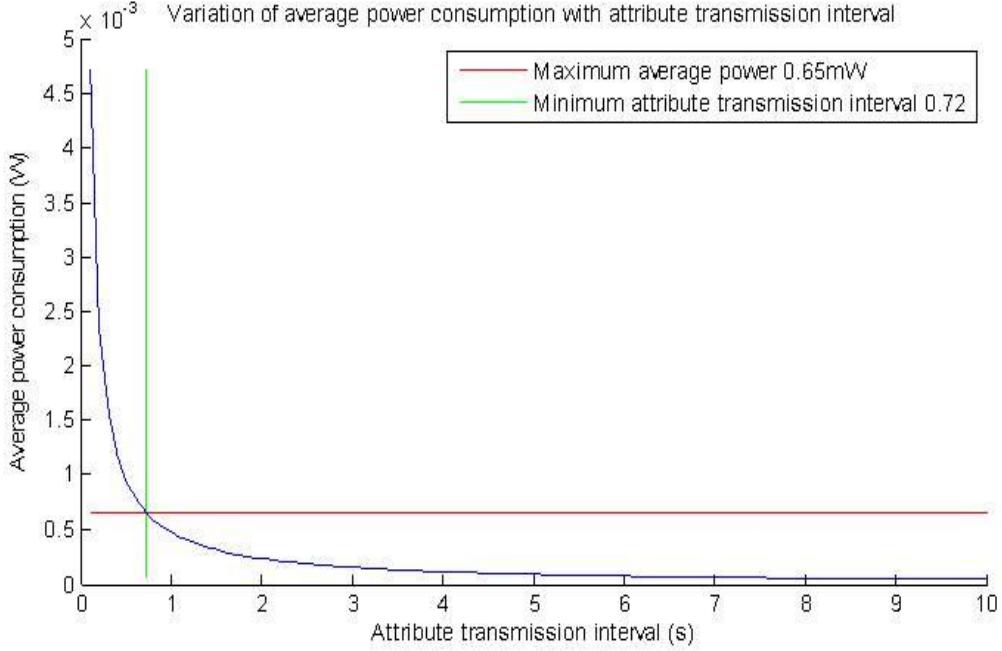
Using ( 34 ), ( 35 ) and ( 36 ), the amount of charge that is consumed on average by the ED, for  $\lambda_{transition}t = 0.3$ , as a function of attribute transmission interval is depicted in Figure 49.



**Figure 49 : Average discharge due to the ED as a function of attribute transmission interval**

The average discharge in the region of sustainable operation is well below  $0.002 \mu Ah$ . However the peak discharge is a much more important statistic than the average discharge and it has been shown

that  $D_{attActive}$ ,  $D_{eventActive}$  and  $D_{startup}$  are well below  $D_{peak}$ . The average power consumption of the ED is depicted by the following figure.



**Figure 50 : Variation of average power consumption with attribute transmission interval**

Here again, the asymptotic nature of both axes is due to the approximation of power and discharge expressions based on ( 32 ). It can also be seen that the average power consumption of the node for a lock probability of 0.23 is greater than the corresponding  $P_{mean}$  of 0.39 mW according to ( 39 ). Therefore the sensor operation may not be sustainable on average for uniformly distributed speeds. It must be noted that it is highly unlikely that speeds are uniformly distributed and therefore a full characterization of the distribution of wheel speeds is necessary to determine the average available power and the average power consumed by the sensor. It must also be noted that the power leaked by various components of the ED has not been considered here.

### 8.2.6 Active vs. Passive Sensing:

In the worst case, the pulse converter actively draws 0.06  $\mu Ah$  of charge which is about 35% of the peak discharge capacity ( $D_{peakCap}$ ). At this worst case average current draw of 2.41 mA, according to ( 18 ),  $C_{out}$  will only be able to sustain the pulse converter for slightly over 0.2 s before discharging below  $V_{minCap}$ . This discharge rate, when periodic, is unsustainable because  $C_{out}$  needs more than 1 s to charge back to  $V_{maxCap}$ . This makes it clear that the active pulse converter cannot be active indefinitely, reinforcing the need for passive sensing mechanisms for non-deterministic sensing, as laid out in section 5.4.

The passive sensor, on the other hand, which draws about 600 nA according to [51], would discharge  $C_{out}$  below  $V_{minCap}$  in slightly over 1000 s. This is more than enough time for the rechargeable battery in the EHU to charge  $C_{out}$  back to  $V_{maxCap}$ . In effect, the voltage is quite stable at  $V_{maxCap}$  when the passive sensor is active and therefore ensures high availability as pointed out in section 5.4.

The high availability of the AWSS in detecting transition events is empirically demonstrated in an end-to-end illustrative scenario, presented in section 8.3.

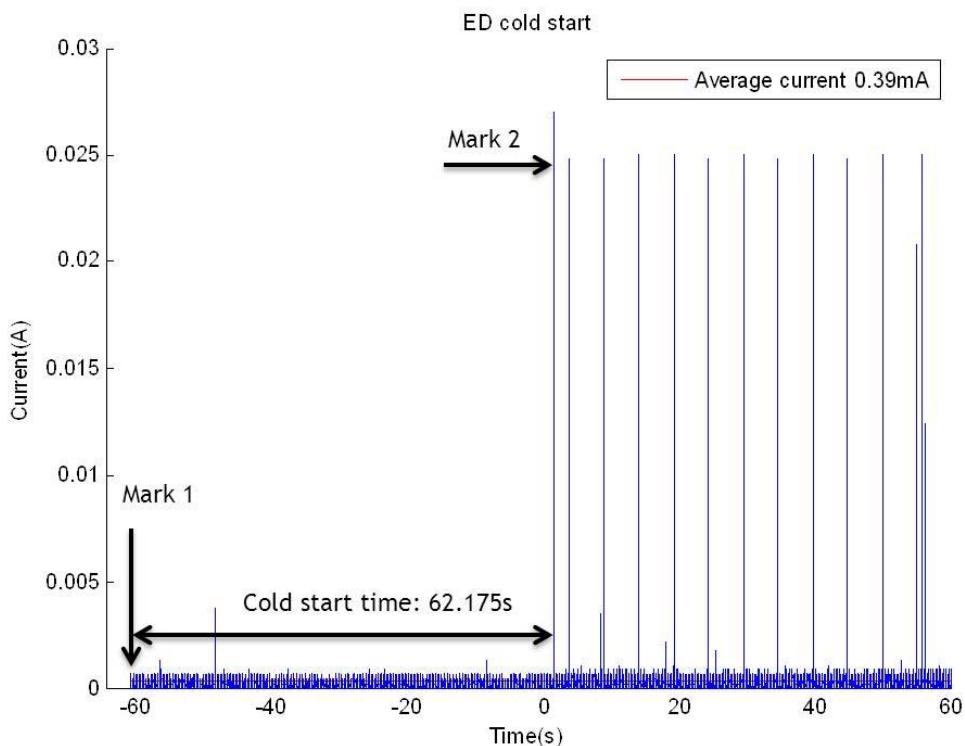
It should however be noted that the rechargeable battery in the Cymbet CBC-Eval-09 would not be able to sustain a  $600\text{ nA}$  draw indefinitely, since the battery would stop recharging  $C_{out}$  when it drains to  $\sim 20\%$  of its capacity. The time for which the rechargeable battery can sustain the passive rotational state sensor has been measured in the prototype to be slightly over *2 hours*. Measuring this ‘cold stop’ time has been quite difficult because of the loading effect of oscilloscope or multimeter probes, which quickly drain  $C_{out}$  when it is not actively being recharged.

### 8.2.7 ED cold start:

The ED can be defined to undergo a ‘cold start’ if the initial output voltage available from the EHU is  $0\text{ V}$ . Once the wheel starts rotating and the energy harvested from the rotation charges the EHU buffers, power is available for the ED to start up and operate. In the cold start case, therefore, the wheel is rotating when the ED starts up. There are many cases where a cold start is possible, for example:

1. During installation when the buffer capacitor has not yet been charged and will be charged for the first time
2. During long periods of inactivity, hours in the case of the prototype and days or months in the case of a production application

In a cold start case, the buffer has to charge with a finite non-zero load, which affects the cold start duration. A sample cold start case is presented in Figure 51, which follows the test procedure laid out in Appendix F. The sample cold start case that is presented here is the best case scenario, i.e. the case where the wheel starts and very quickly speeds up to its maximum rotational speed.



*Figure 51 : ED cold start example*

The best case cold start duration in the prototype is quite high at  $62.175\text{ s}$ , which would definitely be unacceptable in a production application. A reduction in drain currents, in future iterations, would bring down the cold start time. The above figure illustrates a cold start followed by a sequence of attribute and event transmissions lasting  $\sim 60\text{ s}$ . As seen from the plot the average current drawn during this entire sequence is about  $400\text{ }\mu\text{A}$ , which will go down as total transmission duration increases and  $p_{lock}$  comes into effect.

### 8.3 Illustrative end-to-end application scenario:

The end-to-end execution scenario presented in this section provides a system level view of operations. As always the test procedure can be found in Appendix F. In the end-to-end scenario, the most useful interface is the Matlab console of the PM along with the WER control voltage. As the WER is put through its paces the messages displayed by the PM in this illustrative case is shown in Figure 52.

The numbers marking each case in Figure 52 is explained below.

1. ***First event message:*** The first message that is transmitted by the ED is always an event message. This stems from the transition event emulation that was explained in section 7.3.2 and Figure 27. This emulation prompts the ED to determine the current rotational state of the wheel and transmit this state to the AP. In this case since the wheel was rotating when the ED started, the first message was a wheel unlocked event notification. If the wheel had not been rotating when the ED starts (a hot start case) then the first message would have been a wheel locked notification.
2. ***Attribute messages:*** Unless there is an immediate change in the rotational state of the wheel after the first event message, the ED periodically transmits attribute messages as seen in the figure. As the WER control is varied, the corresponding variation in wheel speed can be seen from the data stream, but not in real time.
3. ***Wheel lock and unlock (multiple transmissions):*** When the WER control voltage is switched off, the wheel stops rotating and a lock notification is seen in real-time and when it starts rotating, an unlock notification is received. No attribute messages are received in the intervening period. As seen in the event stream in 3, there are times when multiple event notifications are received. This is a sporadic and as yet unresolved bug in the prototype arising from the slow transitions in the rotational state signal, which spans many milliseconds in the worst case.
4. ***Wheel lock and unlock (clean transmission):*** In this event sequence, a clean transmission of one notification per event can be seen.
5. ***Gradual reduction in speed:*** This sequence depicts a gradual reduction in speed which stops the wheel and triggers a lock notification. When the speed picks up, an unlock notification is received before receiving attribute messages.

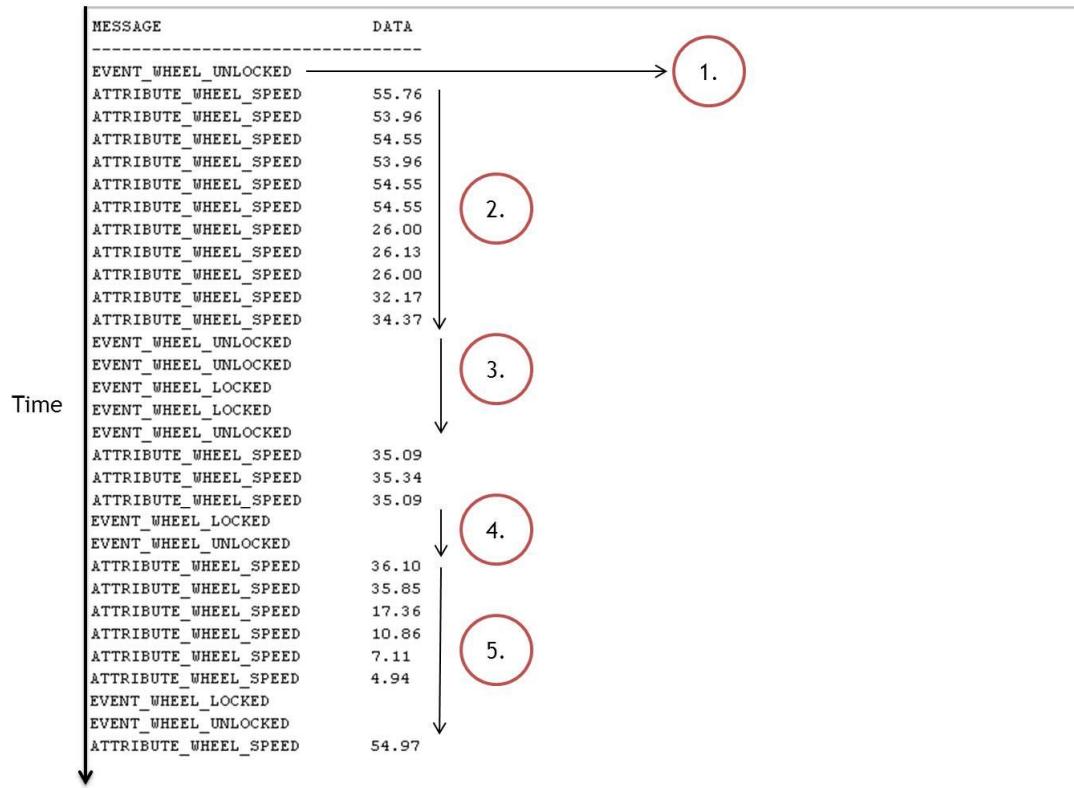


Figure 52 : AWSSN end-to-end illustration

## Chapter 9 – Conclusions

The concept that the Wheel Speed Sensor (WSS) can be used as an energy harvesting transducer has been successfully proven, and an Autonomous Wheel Speed Sensor (AWSS) has been successfully developed. The AWSS combines the sensing and energy harvesting capabilities of the WSS. This chapter presents a summary of the main results in section 9.1 and suggestions for further development in section 9.2.

### 9.1 Summary of results:

The VR electromagnetic transduction setup of the WSS has been determined to be a viable energy harvesting transducer. Under conditions of maximum power transfer, the WSS is found to be capable of producing power levels of  $11.4 \mu W - 1 mW$  for a speed range of  $\sim 30 - 290 RPM$  or equivalently a vehicle speed range of  $\sim 6 - 55 kmph$ , assuming a wheel of diameter  $1 m$ . These power levels have been measured in an experimental wheel setup, with an air gap of  $0.25 mm$  between the VR sensor and the pole wheel. In a WSS assembly in a truck, with an air gap of  $0.1 mm$  and a speed range of  $10 - 100 kmph$ , the amount of power produced by the transducer would only increase.

The power harvested from the WSS has been proven to be sufficient for simple autonomous wheel speed monitoring. A prototype AWSS node has been created by integrating readily available COTS components such as the Cymbet CBC-Eval-09 energy harvesting kit, the TI MSP430 ultra-low power microcontroller and the TI CC2500 proprietary  $2.4 GHz$  transceiver. The AWSS uses the light weight SimpliciTI standard for wireless peer-to-peer communication with an access point. Periodic transmission of wheel speed and near real-time transmission of wheel stop/start events have been successfully implemented in the prototype. Both these operations have been implemented with an average power consumption of less than  $0.65 mW$  (under certain modeling assumptions), with an average duty cycle of less than 5.88%, for a wheel that is rotating  $\sim 80\%$  of the time. Apart from implementing a simple two-node Autonomous Wheel Speed Sensor Network (AWSSN), a network with many more autonomous nodes has been proposed along with its benefits and challenges.

Under optimal integration and assembly, the AWSS can save  $\leq 5 EUR$  in assembly and material costs per sensor. Here it must be noted that each commercial vehicle could have up to 8 sensors and AB Volvo annually produces a few hundred thousand commercial vehicles. Additional savings are possible in after-market maintenance of the installed sensor, and by eliminating the need to develop and maintain sensors which are customized for different axles. There is, however, a long road ahead for a full-fledged sensor network solution as functional concerns, such as availability and real-time task execution, and non-functional concerns, such as new assembly procedures and EMC concerns, have to be addressed.

Duty cycle reduction is the main tactic that has been used in this project to address availability and real-time execution concerns of the autonomous node. The usage of passive stimulation mechanisms has been shown to provide high availability for performing non-deterministic sensing. A set of design principles have been codified which defines how passive stimuli of high abstraction can be applied to increase the availability by reducing the duty cycle. By drawing less than  $1 \mu A$ , the passive rotational state sensor has been demonstrated to be highly available and capable of sensing wheel stop/start events within  $100 ms$ . In addition, the usage of the rotational state sensor decreases the duty cycle and power consumption by an amount that is equal to the proportion of time for which the wheel is not rotating.

A basic mathematical model of the operation of the autonomous sensor based on the ‘leaky bucket’ analogy has been developed. While the model presented in the project is specifically designed for the AWSS, the idea can be extrapolated to any energy harvesting sensor.

A C-based software platform for energy harvesting sensors has been developed, which can be reused and extended to create an RTOS for energy harvesting sensors.

## 9.2 Suggestions for further development

Being a prototype, the AWSS node developed in this project can be improved in a multitude of ways. In addition, the idea of autonomous sensing of wheel behavior, and the idea of using the VR sensor as an energy harvesting transducer can also be extended in many ways.

### 9.2.1 Extending the AWSS:

The following are suggestions for continued work on the AWSS node itself.

**AWSS functionality:** The pulse converter has been chosen to be active only for contrasting it with passive sensing. It can readily be made passive and the rotational state sensor can be dispensed with as the rotational state signal can be derived from the pulse converter output. This would reduce the duty cycle and power consumption to much lower levels. The rotational state output can also be modified to respond much more quickly than the current 100 ms. A critical problem which needs to be solved is the high cold-start time (~60 s), which can be lowered by reducing the cold-start load on the EHU. Solving this current drain problem would also bring down the cold stop time which is currently quite low at 2 hours. In addition, for accurate characterization of available power levels and sensor power consumption, a statistical model of the distribution of wheel speeds must be developed.

**AWSS Hardware Abstraction Layer (HAL) improvements:** The HAL can be expanded into an energy harvesting RTOS by incorporating state of the art real-time scheduling, energy monitoring services, etc. SimpliciTI needs to be replaced with standards like WirelessHART, which incorporates real-time communication and robust sensor network management.

**AWSS integration and assembly in the truck:** It should be noted that the question of saving costs by using an autonomous WSS instead of a wired WSS has not been completely answered, and this is mainly because of integration and assembly concerns. Ideally the AWSS would have to be integrated with the VR magnetic sensor, but the deployment location of the WSS may be too hostile for the node electronics. This problem has to be dealt with either by using packaging or choosing a safe mounting point, which in turn would decide assembly costs.

### 9.2.2 Extending the idea of autonomous sensing of wheel characteristics:

While the prototype system developed in this project sticks to monitoring the speed of the wheel, it need not be restricted to this application.

**New autonomous sensing applications:** Branching away from speed monitoring, the incorporation of autonomous sensing for other EBS application, like ASR, ESC, etc. can be investigated. While adding new sensing applications, usage of the guidelines in section 5.4 is recommended for increasing availability. Highly active research in the area of transducers, and the development of MEMS/NEMS solid-state ultra-low power inertial sensors, indicates that passive sensing of high levels of abstraction is definitely possible. Sensors can be developed for sensing events like loss of traction and loss of response to steering using such passive inertial sensors, which could lead to autonomous sensing and control of every aspect of the wheel.

***Re-architecting the EBS:*** The AWSS represents a step in the path towards increased distribution of intelligence and control in the EBS. In order to fully utilize the potential of the distributed system, the architecture of the EBS may need to be rearranged extensively. As sensing becomes more autonomous, actuation and management should not be too far behind. As the EBS is made more distributed, savings in cost and possibilities of new applications may arise.

### **9.2.3 Extending the idea of the WSS as an energy harvester:**

The fact that the WSS is a viable power source is a key conclusion of this project and a major recommendation is to reuse this idea in other wheel-axle assemblies.

***A power source for wheel management applications:*** The idea of using a rotating pole-wheel with a stationary magnet nearby, or conversely a rotating magnet with a stationary wheel, can be used to harvest power from the rotational motion of the wheel. This harvested energy can then be used to power any wheel management application, for example the Tire Pressure Monitoring System (TPMS). The presence of an autonomous source of energy also opens up possibilities of new autonomous wheel management applications.

***A power source for rotational assemblies:*** The same energy harvesting idea can be extended to any rotational assembly in the vehicle, such as the camshaft in the engine, or the differential in the powertrain. The amount of energy that stands to be harvested using this technique is considerable and can be used to power new sensing applications in these assemblies. The idea need not be restricted to trucks alone and can be extended to machinery of any kind with rotational assemblies, for example in manufacturing plants, to power new autonomous sensors.

## Bibliography

- [1] US National Science Foundation, "The Sensor Revolution - A special report," [Online]. Available: [http://www.nsf.gov/news/special\\_reports/sensor/overview.jsp](http://www.nsf.gov/news/special_reports/sensor/overview.jsp). [Accessed 4 May 2012].
- [2] Volvo Technology Corporation, "Energy harvesting for sensors in commercial vehicles," VTEC 6200 BMS no. 06200-3214, 2011.
- [3] Volvo Technology Corporation, "Energy harvesting power management," VTEC 6200 BMS no. 06200-3223 , 2011.
- [4] Market Reasearch.com, "Global Sensors Market For Automotive Applications (2011 - 2016)," [Online]. Available: <http://www.marketresearch.com/MarketsandMarkets-v3719/Global-Sensors-Automotive-Applications-6837870/>. [Accessed 22 April 2012].
- [5] W. J. Fleming, "New Automotive Sensors—A Review," *IEE Sensors Journal*, vol. 8, no. 11, pp. 1900-1921, 2008.
- [6] K. Delin and S. P. Jackson, "The Sensor Web: A New Instrument Concept," in *Symposium on Integrated Optics*, San Jose, 2001.
- [7] J. Bousquin, "A History of DARPA's Contributions to Antisubmarine Warfare," DARPA, US Department of Defense, 2008.
- [8] K. S. Pister, "Smart Dust," UC Berkeley,'other educational', 1997.
- [9] Wikipedia, The Free Encyclopedia, "Wireless sensor network," [Online]. Available: [http://en.wikipedia.org/wiki/Wireless\\_sensor\\_network](http://en.wikipedia.org/wiki/Wireless_sensor_network). [Accessed 30 April 2012].
- [10] S. Priya and D. J. Inman, Energy Harvesting Technologies, Springer, 2009.
- [11] S. Beeby and N. White, Energy Harvesting for Autonomous Systems, Artech House, 2010.
- [12] G. D. Szarka, B. H. Stark and S. G. Burrow, "Review of Power Conditioning for Kinetic Energy Harvesting Systems," *IEEE Transactions on Power Electronics*, vol. 27, no. 2, pp. 803-815, 2012.
- [13] L. D'Orazio, F. Visintainer and M. Darin, "Sensor networks on the car: State of the art and future challenges," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2011.
- [14] H. Zervos, "Energy harvesting for automotive applications," IDTechEx Ltd., 2011.
- [15] Volvo 3P, "SENSOR & SHIELD INSTALLATION, Document type: ASSEMBLY REQUIREMENTS, Document No.21052828, Issue index 03, Volume No. 01," 2010.
- [16] Wikipedia, The Free Encyclopedia, "Maximum Power Transfer Theorem," [Online]. Available: [http://en.wikipedia.org/wiki/Maximum\\_power\\_transfer\\_theorem](http://en.wikipedia.org/wiki/Maximum_power_transfer_theorem). [Accessed 30 April 2012].

- [17] D. Spreemann and Y. Manoli, in *Electromagnetic Vibration Energy Harvesting Devices*, Springer, 2012, pp. 2-11.
- [18] G. Manla, N. White and J. Tudor, "Harvesting energy from vehicle wheels," in *Solid-State Sensors, Actuators and Microsystems Conference*, 2009.
- [19] Y. Hu, C. Xu, Y. Zhang, L. Lin, R. L. Snyder and Z. L. Wang, "A Nanogenerator for Energy Harvesting from a Rotating Tire and its Application as a Self-Powered Pressure/Speed Sensor," *Advanced Materials*, vol. 23, no. 35, pp. 4068-4071, 2011.
- [20] F. Khameneifar, M. Moallem and S. Arzanpour, "Modeling and Analysis of a Piezoelectric Energy Scavenger for Rotary Motion Applications," *Journal of Vibration and Acoustics*, vol. 133, no. 1, p. 011005(6 pages), 2011.
- [21] W. Yumei, L. Ming, L. Ping, Y. Jin and D. Xianzhi, "A rotation energy harvester using cantilever beam and magnetostrictive/piezoelectric composite transducer," in *PowerMEMS*, 2010.
- [22] G. Lei and C. Livermore, "Passive self-tuning energy harvester for extracting energy from rotational motion," *Applied Physics Letters*, vol. 97, no. 8, p. 081904(3 pages), 2010.
- [23] S. D. Conrad, "Development of an inertial generator for embedded applications in rotating environments," Massachusetts Institute of Technology, 2007.
- [24] T. T. Toh, A. Bansal, G. Hong, P. D. Mitcheson, A. S. Holmes and E. M. Yeatman, "ENERGY HARVESTING FROM ROTATING STRUCTURES," in *PowerMEMS*, Freiburg, Germany, 2007.
- [25] Y.-J. Wang, C.-D. Chen and C.-K. Sung, "Design of a frequency-adjusting device for harvesting energy from a rotating wheel," *Sensors and Actuators A: Physical*, vol. 159, no. 2, pp. 196-203, 2009.
- [26] T. O'Donnell, C. Saha, S. Beeby and J. Tudor, "Scaling effects for electromagnetic vibrational power generators," in *Symposium on Design, Test, Integration and Packaging of MEMS/MOEMS*, 2006.
- [27] Knorr-Bremse, "EBS Product Catalogue," 2011.
- [28] Volvo Trucks, "I-Shift: Transmission with intelligence - Product information," [Online]. Available: [http://www.volvotrucks.com/trucks/na/en-us/products/engines/ishift/pages/ishift\\_ILF.aspx](http://www.volvotrucks.com/trucks/na/en-us/products/engines/ishift/pages/ishift_ILF.aspx). [Accessed 7 May 2012].
- [29] Volvo 3P, "CLAMPING FOR EBS-SENSORS, Document type: INSTALLATION REQUIREMENT, Document No. 21591897, Issue Index 01, Volume No. 01," 2010.
- [30] R. Silva, K. Farinholt and G. Park, "Hybrid energy sources for embedded sensor nodes," in *Industrial and Commercial Applications of Smart Structures Technologies*, 2011.
- [31] A. Hehr, G. Park and K. Farinholt, "Hybrid energy harvesting/transmission system for embedded

- devices," in *Industrial and Commercial Applications of Smart Structures Technologies*, 2012.
- [32] Y. K. Tan and S. Panda, "Energy Harvesting From Hybrid Indoor Ambient Light and Thermal Energy Sources for Enhanced Performance of Wireless Sensor Nodes," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 9, pp. 4424-4435, 2010.
- [33] D. Audet, L. C. de Oliveira, N. MacMillan, D. Marinakis and K. Wu, "Scheduling recurring tasks in Energy Harvesting sensors," in *IEEE INFOCOMM*, 2011.
- [34] C. Moser, D. Brunelli, L. Thiele and L. Benini, "Real-time scheduling for energy harvesting sensor nodes," *Real-Time Systems*, vol. 37, no. 3, 2007.
- [35] Y. Gu and T. He, "Bounding Communication Delay in Energy Harvesting Sensor Networks," in *International Conference on Devices Circuits and Systems (ICDCS)*, 2010.
- [36] S. Liu, Q. Wu and Q. Qiu, "An adaptive scheduling and voltage/frequency selection algorithm for real-time energy harvesting systems," in *Design Automation Conference*, 2009.
- [37] M. Dehghan and M. Kargahi, "Adaptive checkpoint placement in energy harvesting real-time systems," in *Iranian Conference on Electrical Engineering*, 2010.
- [38] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, M. Nixon and W. Pratt, "WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control," in *Proceedings of the 2008 IEEE Real-Time and Embedded Technology and Applications Symposium*, 2008.
- [39] A. Saifullah, Y. Xu, C. Lu and Y. Chen, "Real-Time Scheduling for WirelessHART Networks," in *Proceedings of the 2010 31st IEEE Real-Time Systems Symposium*, 2010.
- [40] D. Chen, M. Nixon and A. Mok, *WirelessHART(TM): Real-Time Mesh Network for Industrial Automation*, Springer, 2010.
- [41] Q. Zheng, H. Tu, A. Agee and Y. Xu, "Vibration energy harvesing device based on asymmetric air-spaced cantilevers for Tire Pressure Monitoring System," in *PowerMEMS*, 2009.
- [42] E. Westby and E. Halvorsen, "Design and Modeling of a Patterned-Electret-Based Energy Harvester for Tire Pressure Monitoring Systems," *IEEE/ASME Transactions on Mechatronics*, vol. PP, no. 99, pp. 1-11, 2011.
- [43] S. Gruber, H. Reinisch, H. Unterassinger, M. Wiessflecker, G. Hofer, M. Klammlinger, W. Pribyl and G. Holweg, "A passively powered BAW-based multi-standard in-tire identification and monitoring system," in *7th International Symposium on Communication Systems Networks and Digital Signal Processing (CSNDSP)*, 2010.
- [44] Cymbet Corporation, "Datasheet - CBC EVAL 09," [Online]. Available: <http://www.cymbet.com/pdfs/DS-72-13.pdf>. [Accessed 16 March 2012].
- [45] Cymbet Corporation, "Datasheet - CBC915-ACA," [Online]. Available: <http://www.cymbet.com/pdfs/DS-72-15.pdf>. [Accessed 06 March 2012].

- [46] Texas Instruments Inc., "eZ430-RF2500 Development Tool User's guide," [Online]. Available: <http://www.ti.com/lit/ug/slau227e/slau227e.pdf>. [Accessed 1 May 2012].
- [47] Texas Instruments Inc., "Datasheet - MSP430F22x2, MSP430F22x4 Mixed Signal Microcontroller (Rev. F)," 14 July 2011. [Online]. Available: <http://www.ti.com/lit/ds/symlink/msp430f2274.pdf>. [Accessed 16 March 2012].
- [48] Texas Instruments Inc., "CC2500 Low-cost low-power 2.4GHz RF Transceiver," [Online]. Available: <http://www.ti.com/lit/ds/swrs040c/swrs040c.pdf>. [Accessed 1 May 2012].
- [49] Texas Instruments Inc., "SimpliciTI Simple Modular RF Network Specification," 2011.
- [50] Maxim Integrated Products, "Datasheet - MAX3095/3096 Quad RS-422/RS-485 Receivers," [Online]. Available: <http://datasheets.maxim-ic.com/en/ds/MAX3095-MAX3096.pdf>. [Accessed 5 May 2012].
- [51] Microchip Technology, "Datasheet - MCP6541/1R/1U/2/3/4 Push-Pull Output Sub-Microamp Comparators," [Online]. Available: <http://ww1.microchip.com/downloads/en/devicedoc/21696f.pdf>. [Accessed 5 May 2012].
- [52] IAR Systems, "Product page - IAR Embedded Workbench for TI MSP430," IAR Systems, [Online]. Available: <http://www.iar.com/en/Products/IAR-Embedded-Workbench/TI-MSP430/>. [Accessed 16 March 2012].
- [53] MSPGCC Project, "MSPGCC Wiki," [Online]. Available: [http://sourceforge.net/apps/mediawiki/mspgcc/index.php?title=MSPGCC\\_Wiki](http://sourceforge.net/apps/mediawiki/mspgcc/index.php?title=MSPGCC_Wiki). [Accessed 06 March 2012].
- [54] COMSOL AB, "COMSOL Multiphysics homepage," [Online]. Available: <http://www.comsol.com/>. [Accessed 26 March 2012].

## Appendix A - The Random Telegraph (RT) process

The well-known Random Telegraph (RT) process has been used to model the rotational state of the wheel. The bi-stable version of the RT process is described by:

$$X(t) = 0/1$$

The number of transitions in  $(0, t)$  between the two states  $X(t) = 0$  and  $X(t) = 1$  is Poisson distributed with parameter  $\lambda t$ . Therefore if  $N(t)$  is the number of transitions in  $(0, t)$  then we have

$$P\{N(t) = n\} = \frac{e^{-\lambda t} (\lambda t)^n}{n!}$$

The average number of transitions in  $(0, t)$  is therefore given by

$$E\{N(t)\} = \lambda t$$

If an initial state of  $X(0) = 1$  is assumed, in the interval  $(0, t)$ ,

$$P\{X(t) = 1 | X(0) = 1\} = P\{N(t) = even\} = e^{-\lambda t} \cosh(\lambda t) \quad (40)$$

$$P\{X(t) = 0 | X(0) = 1\} = P\{N(t) = odd\} = e^{-\lambda t} \sinh(\lambda t) \quad (41)$$

And the expected value

$$E\{X(t) | X(0) = 1\} = e^{-\lambda t} \cosh(\lambda t) \quad (42)$$

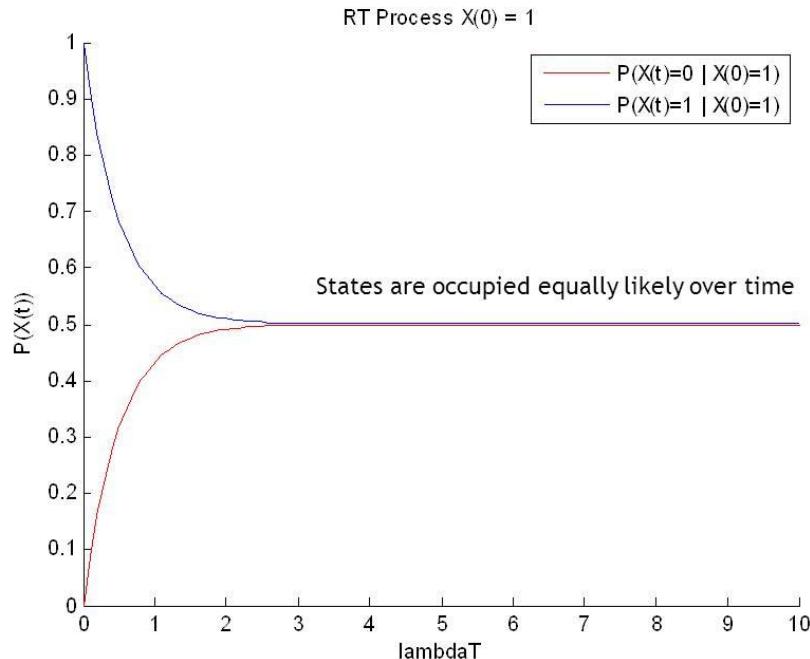


Figure 53 : RT process for  $X(0) = 1$

As shown in the figure above, as the observation interval increases, both the states are equally likely to be occupied.

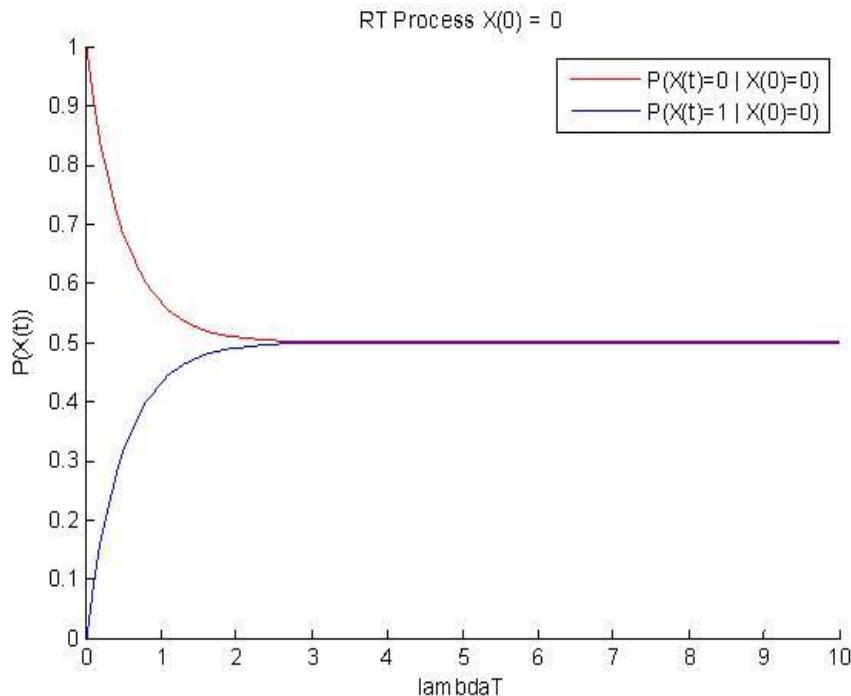
However, if the initial state is assumed to be 0, i.e. if  $X(0) = 0$ , then the probabilities become inverted, i.e.

$$P\{X(t) = 1 \mid X(0) = 0\} = P\{N(t) = \text{odd}\} = e^{-\lambda t} \sinh(\lambda t) \quad (43)$$

$$P\{X(t) = 0 \mid X(0) = 0\} = P\{N(t) = \text{even}\} = e^{-\lambda t} \cosh(\lambda t) \quad (44)$$

$$E\{X(t) \mid X(0) = 0\} = e^{-\lambda t} \sinh(\lambda t) \quad (45)$$

The plot of the probability function,



**Figure 54 : RT process for  $X(0) = 0$**

From (42) and (45) it can be seen that

$$\lim_{\lambda t \rightarrow \infty} E\{X(t) \mid X(0) = 1\} = \lim_{\lambda t \rightarrow \infty} E\{X(t) \mid X(0) = 0\} = 0.5 \quad (46)$$

This implies that the expected values (and state probabilities as seen in Figure 53 and Figure 54) converge to 0.5 as  $\lambda t$  increases, i.e. the influence of the knowledge of the initial state decays exponentially.

## Appendix B - Hardware Abstraction Layer (HAL) user/programmer guide

HAL definitions can be found in *HAL.h* and its implementation in *HAL.c*. The services provided by the HAL are described below.

**Initialization services:** Separate initialization routines are used for ED and AP, due to different SimpliciTI configurations and differences in hardware. The function signatures are

```
void halInitAP(void)
```

```
void halInitED(void)
```

**Sleep/wakeup service:** The macro *SLEEP\_3()* can be used to put the MSP430 to sleep in LPM3 and *WAKEUP\_3()* can be used to wake it up from LPM3

**Interrupt callback service:** Interrupt callback services are provided for Timer-A and Port 2 using the interrupt type definition *MSPInterrupt\_t*.

The interrupt callback has the signature:

```
typedef boolean_t (* MSPInterruptCallback_t)(MSPInterrupt_t)
```

A *booleant* return value of *TRUE* executes *WAKEUP\_3*, while a *booleant* return value of *FALSE* does not execute *WAKEUP\_3*, so that the MSP430 would continue sleeping.

**Port 2 interruption hack:** Few pins in Port 2 are used in the ez430 board by SimpliciTI for communication between the MSP430 and the CC2500. In order to have a common ISR for HAL and SimpliciTI, the following changes have to be made in the open source SimpliciTI implementation

1. In the file *mrfi\_board.c*, comment out *BSP\_ISR\_FUNCTION*
2. In the file *mrfi\_radio.c*, comment out the prototype for the function *MRFI\_GpioIsr*
3. In the file *mrfi.h*, add a prototype for the function *MRFI\_GpioIsr*

After making these changes, the port 2 ISR is implemented in the *HAL.c* by including a call to *MRFI\_GpioIsr*.

**Snooze service:** This is used to pause in LPM3 for roughly one second. The function has the signature

```
void snooze(void)
```

**WSN services:** Currently the HAL only provides WSN services for the ED which are

```
void joinNetwork(void)
```

```
boolean_t transmitMessage(unsigned char * msg, unsigned char msgLen)
```

Both services utilize the SimpliciTI peer network. The *joinNetwork* service links to the AP which is up and running (a pre-requisite) while the *transmitMessage* service sends a SimpliciTI peer frame to the AP. WSN services for the AP are currently not well-encapsulated and are provided by *halInitAP* and the SimpliciTI library itself.

**MCU services:** Representing a bulk of the services provided by the HAL, MCU services can be classified as follows.

1. *VLO calibration:* This can be used to determine the current frequency of the unstable VLO. This function has the signature

```
unsigned int calibrateVlo(void)
```

2. *Timer-A compare mode services:* This can be used to operate Timer-A CC0, CC1 and CC2 in the compare mode for a user-set count level. Macros of the form *START\_TIMER\_A\_COMPAREX()* and *STOP\_TIMER\_A\_COMPAREX()* have been defined where X can be 0, 1 or 2.
3. *WSS signal transition stimulus services:* The rotational state sensor output 1 is connected to Port2.0 and the following macros provide interrupt setup and teardown services.

```
WSS_SIGNAL_TRANSITION_INPUT_SETUP()
```

```
WSS_SIGNAL_TRANSITION_INTERRUPT_TEARDOWN()
```

Signal transition interrupts are of two kinds, a rising-edge interrupt which can be set up using *WSS\_SIGNAL\_PX\_INTERRUPT\_SETUP()* for detecting signal presence (PX) and *WSS\_SIGNAL\_AX\_INTERRUPT\_SETUP()* for detecting a signal absence (AX). Using these services both locking and unlocking states can be detected based on the current state. In addition, the following service is provided to emulate an interrupt.

```
WSS_CREATE_SIGNAL_TRANSITION_INTERRUPT()
```

4. *MAX3096 management services:* The MAX3096 IC has control signals which are connected to Port4.4 and Port4.6. Services are provided to setup the control signals using *MAX3096\_SETUP()*, turn on the IC using *MAX3096\_ON()* and to turn it off using *MAX3096\_OFF()*.
5. *WSS pulse signal management services:* The output of the MAX3096 is connected to Port2.2. Services are provided to setup the connection using *WSS\_PULSE\_INPUT\_SETUP()*. A rising edge capture can be started with a configurable timeout using

```
WSS_MEASUREMENT_START(measurementTimeout)
```

The rising edge capture can be stopped using *WSS\_SIGNAL\_CAPTURE\_STOP()*, while the measurement can be stopped using *WSS\_MEASUREMENT\_STOP()*.

## Appendix C - Application Logic Layer (ALL) programmer/user guide

This appendix describes the implementation of the ALL layer in the prototype AWSSN system. The ALL in the ED is defined in *AWSSN\_ED.h* and implemented in *AWSSN\_ED.c*, while the ALL for the AP is defined in *AWSSN\_AP.h* and implemented in *AWSSN\_AP.c*. Provision for common definitions can be found in *AWSSN.h*. The ALL in both the ED and AP is semi-encapsulated, and the ALL in the ED is implemented as a state machine.

**State machine based ED ALL:** ALL status is maintained by the variable *EDStatus* of type *EDStatus\_t*. This status variable holds the state transition signal variables and all state transitions are controlled by the function *void transitState(void)*. The individual states are only allowed to set the transition signals and are not allowed to set the next state. It should be noted that this is a design style and access protection is not implemented in the prototype. The state transition diagram is found in Figure 27.

The implementation details of each state are as follows:

1. ***ED\_EVENT\_PROCESS*:** This state is responsible for processing all asynchronous events and all logic is concentrated in the function *void processEvent(void)*. The state machine transits to this state if the flag *EDStatus.eventToProcess* is *TRUE*. The state variable *EdEvtControl* of type *EDEventControl\_t* is the state variable which keeps track of the event that has occurred. This state variable could potentially be expanded into a queue which is currently abstracted by the signal *EDStatus.eventToProcess*. The only event implemented in the prototype is the WSS signal transition event:

*EXT\_EVT\_WSS\_SIGNAL\_CHANGE.*

The transition signaling is done in the *port2InterruptCallback* when the port interrupt is triggered by the external signal transition. When a signal transition event occurs, the logic for this event in *processEvent()* checks the state of the signal to decide whether the wheel is locked or not and sets the inverse interrupt to wait for the inverse transition. Once the event is processed the state sets *EDStatus.dataToTx* to signal transmission of the event true and exits the state.

2. ***ED\_DATA\_TX*:** This state is responsible for transmitting events and attributes to the AP. It exposes a transmission buffer *txBuf* of type *EDTxBuf\_t* which can be used by any other state to populate exactly one message. Setting *EDStatus.dataToTx* to *TRUE* signals a transition to this state, after which it uses HAL's *transmitMessage()* function to transmit the message in the buffer, and transit by default to *ED\_SLEEP*. Acknowledgement by the AP is not implemented in the prototype and therefore there is no re-transmission logic.
3. ***ED\_SLEEP*:** This state forces the ED to sleep in a time-bound manner by using HAL's *SLEEP\_3()*. The state uses a global variable *EDSleepControl* of type *EDSleepControl\_t* to configure the sleep duration. This state always uses Timer-A CC2 in the compare mode for this sleep and the transition after the expiration of the sleep interval is decided in the *timerInterruptCallback*. If the WSS signal is present, a transition to *ED\_MEASURE* is set in the *timerInterruptCallback* and the HAL is signaled to execute *WAKEUP\_3()*. If no signal is present after the elapsed time period, the state transits

to itself. The sleep state can however be woken up by a WSS signal transition signaled by the *port2InterruptCallback* in which case it transits to *ED\_EVENT\_PROCESS*.

4. ***ED\_MEASURE***: This state is responsible for making raw measurements of external signals after which it populates a data buffer. In this case the external signal is the WSS pulse signal and the data buffer is *wssMeasurementBuffer* of type *WSSMeasurementBuffer\_t*. This data type is not well encapsulated and is used for controlling the measurement and the measurement state in addition to storing raw measured data. A transition to this state is signaled by setting *EDStatus.signalToMeasure* to *TRUE* which in this case is done by *ED\_SLEEP* if the WSS signal is present. This state exposes a measurement callback with signature

```
boolean_t measurementSetupCallback void
```

This callback, for increased modularity, is used to setup measurements after which the state goes to sleep (*SLEEP\_3()*) if so signaled by the return type. The reason for having this setup callback is because a lot of measurements can be done in low power modes which the callback can exploit. It should be noted that if the callback signals the microcontroller to go to sleep, it must ensure that an interrupt will eventually wake up the MCU. Measurement teardown and signals for state transition in this case is done in the interrupt callback.

5. ***ED\_ATTR\_PROCESS***: This state is responsible for processing the raw data collected by *ED\_MEASURE* which is populated in the *wssMeasurementBuffer*. All logic in this state is implemented by *void processWSSData(void)*, where the measured WSS signal data is used to calculate its time period and make an attribute out of it. This attribute is then populated in the *txBuf* and a transition to *ED\_DATA\_TX* is signaled. If the state does not successfully constitute an attribute it transits to *ED\_SLEEP* by default.

***Non-state machine based AP ALL***: The AP ALL is non-state machine based and has a very low level of encapsulation. Most of the AP ALL source is reused from the example application mentioned in section 7.3.2 and the only significant modification done here is to adapt this code to use the HAL interface.

## Appendix D - AWSSN Cluster definition

The AWSSN cluster is in essence the contract of transaction between various nodes in the AWSSN. A brief description of the prototype cluster definition:

**Attributes:** Only one attribute is defined in the prototype cluster:

1. **ATTRIBUTE\_MEAN\_PERIOD\_COUNT:** This defines the wheel speed attribute, as the wheel speed is actually transmitted as the average number of counts of a timer representing the time period of the WSS pulse signal. This attribute carries an unsigned integer (two bytes) representing the current average period count and the wheel speed is derived from this attribute using a process that is described later on.

**Events:** Three events are defined in the cluster out of which two are used and these two are described here:

1. **EVENT\_WHEEL\_LOCKED:** This event signals that the wheel was previously rotating and has now stopped rotating.
2. **EVENT\_WHEEL\_UNLOCKED:** This event signals that the wheel was previously not rotating and has now started rotating.

**Values:** The static values defined in the prototype cluster are:

1. **VALUE\_TIMER\_REF\_CLOCK:** The frequency of the timer whose elapsed count is represented by *ATTRIBUTE\_MEAN\_PERIOD\_COUNT*.
2. **VALUE\_MEAN\_PERIOD\_COUNT\_SCALE:** An integer which is a power of 10 used to pre-scale the value carried by *ATTRIBUTE\_MEAN\_PERIOD\_COUNT* for increasing accuracy.
3. **VALUE\_ATTRIBUTE\_MEAN\_PERIOD\_COUNT\_TX\_PERIOD\_S:** This floating point value sets the periodic transmission interval of *ATTRIBUTE\_MEAN\_PERIOD\_COUNT*.
4. **VALUE\_WSS\_WT:** This value is an integer which specifies the number of rising edges to capture and determine the average time period. This value sets the WSS parameter  $w_t$  specified in section 5.3.4. This value should not be set to less than 2.
5. **VALUE\_FREQ\_LOWER\_LIM:** This value sets the lowest frequency that needs to be sensed and any frequency less than this value is considered to be zero.

The values *VALUE\_TIMER\_REF\_CLOCK* and *VALUE\_MEAN\_PERIOD\_COUNT\_SCALE* are used to derive the wheel rotation frequency using the conversion

$$f = \frac{VALUE_TIMER_REF_CLOCK * VALUE_MEAN_PERIOD_COUNT_SCALE}{value(ATTRIBUTE_MEAN_PERIOD_COUNT)}$$

And this frequency can be converted into wheel speed using ( 2 ). All these conversions are implemented in the PM.

## Appendix E - Parametric register

This appendix presents all the system level parameters in a unified manner, the default value of each parameter and in which system module they are set.

Parameter	Default value	Additional remarks	Set by
$d_{wheel}$	1 m	Assumed for modeling	-
$n_{teeth}$	100	-	WER
$s_{min}$	5.77 kmph	-	WER
$s_{max}$	54.85 kmph	-	WER
$V_1$	13.3 mV <sub>pp</sub> Hz <sup>-1</sup> or 4.5 mV <sub>RMS</sub> Hz <sup>-1</sup>	-	WER
$f_{min}$	51 Hz	-	WER
$f_{max}$	485 Hz	-	WER
$t_{min}$	2.1 ms	-	WER
$t_{max}$	19.6 ms	-	WER
$V_{min}$	732 mVPP or 234 mVRMS	-	WER
$V_{max}$	6.7 VPP or 2.2 VRMS	-	WER
$Z_{WSS}$	1.2 kΩ	-	WER
$P_{min}$	11.4 μW	Calculated based on maximum power transfer	WER
$P_{max}$	1 mW	Calculated based on maximum power transfer	WER
$\lambda_{transition}$	0.15	Assumed for modeling	-
$p_{lock}$	0.23	Calculated based on the RT process model using $\lambda_{transition}t = 0.3$	-
$P_{mean}$	0.39 mW	Calculated based on model	WER
$E_{min}$	600 mV	-	Cymbet CBC-Eval-09
$E_{max}$	2 V	-	Cymbet CBC-Eval-09
$Z_{min}$	100 Ω	-	Cymbet CBC-Eval-09
$Z_{max}$	100 kΩ	-	Cymbet CBC-Eval-09
$D_{bat}$	100 μAh	-	Cymbet CBC-Eval-09
$V_{limBat}$	3 V	-	Cymbet CBC-Eval-09
$D_{effectiveBat}$	Listed for each discrete operation further on	-	-
$R_{bat}$	1 kΩ	-	Cymbet CBC-

			Eval-09
$C_{out}$	$1000 \mu F$	-	Cymbet CBC-Eval-09
$V_{maxCap}$	$3.6 V$	Assumed for modeling and design	-
$V_{minCap}$	$3 V$	Assumed for modeling and design	-
$V_{chargeCap}$	$3.8 V$	-	Cymbet CBC-Eval-09
$D_{peakCap}$	$0.1671 \mu Ah$	Calculated for modeling and design	-
$V_{pulseTH}$	$45 mV$	-	MAX3096
$w_t$	5	Configurable	AWSSN cluster library
$b_t$	16 bits	-	MSP430 Timer-A
$f_t$	$12 kHz$ unstable	-	MSP430 ACLK set in <i>HAL.c</i>
$t_{attSense}$	54.5 ms on average	Calculated and verified by measurement	-
$t_{attInterval}$	5 s	Configurable	AWSSN cluster library
$f_{lock}$	$7.44V_{CC} Hz$	Set by design	Rotational state sensor
$t_{rise}$	$3.5 \mu s$	Set by design	Rotational state sensor
$t_{fall}$	100 ms	Set by design	Rotational state sensor
$t_{eventSense}$	$\infty$	Indefinite measurement	-
$t_{eventInterval}$	360 s	Calculated from model assumptions	-
$b_{PHY}$	16 – 17 bytes	Configurable	14 bytes of overhead set by SimpliciTI and 1 – 2bytes of application payload set by <i>eh-wss_ED.c</i>
$r_{RF}$	250 Kbps	Configurable	SimpliciTI library MRFI implementation
$t_{transmit}$	0.51 – 0.54 ms	Theoretically calculated but not verified by measurement as measurement procedure is quite elaborate	-
$t_{attActive}$	10.5 – 98 ms	Theoretically calculated and	-

		verified by measurement. Allow for 3ms transmission time in measured value	
$t_{eventActive}$	3 ms	Measured	CC2500
$d$	< 5.88%	Calculated from measured and assumed model parameters	-
$I_{attActive}$	2.41 – 3.78 mA	Measured	MAX3096 + MSP430 + CC2500
$D_{attActive}$	0.01 – 0.06 $\mu Ah$	Calculated from measured values	-
$I_{eventActive}$	7.85 mA	Measured	MSP430 + CC2500
$D_{eventActive}$	0.001 $\mu Ah$	Calculated from measured values	-
$D_{avg}$	< 0.002 $\mu Ah$	Calculated from measured and assumed model parameters	-
$I_{avg}$	0.18 mA	Calculated from measured and assumed model parameters	-
$P_{avg}$	0.65 mW	Calculated from measured and assumed model parameters	-

## Appendix F – Prototype characterization test cases

This section will describe the test cases to characterize the prototype as described in section 8.2.

### ***ED Hardware Layer signal characteristics:***

An illustrative procedure for characterizing the WSS signal at the Hardware Layer is as follows:

1. Set voltage and frequency measurement for the chosen channel in the WaveSurfer
2. Set the WER DC control voltage to a required value and switch it on. In this illustration, the voltage is set to the maximum allowed control voltage in the WER.

$$V_{DC} = 31.7 \text{ V}$$

3. Connect the WaveSurfer probe directly across the WSS terminals and make a note of the frequency

$$f_{unloaded} = 483.34 \text{ Hz}$$

4. Record the waveform as *wssUnloaded.dat*
5. Connect the WSS terminals to J19 of CBC-Eval-09 while keeping the probe connected
6. Observe the frequency changes until the CBC-Eval-09 goes into regulation

$$f_{loaded} = 484.8 \text{ Hz}$$

7. When the CBC-Eval-09 goes into regulation (the three indicator lights blink) record the waveform as *wssLoaded.dat*
8. Connect the MAX3096 *G* and *G'* pins directly to supply and ground rails respectively and connect a 3V supply.
9. Connect a second probe to the Y1 output of MAX3096 and observe the frequency

$$f_{pulse} = 485.5 \text{ Hz}$$

10. Record the waveform as *wssPulseConvertedMax.dat*
11. Measure the phase difference between the loaded WSS signal and the pulse converted signal

$$\Phi_{pulse-loaded} = 35.29 \times 10^{-3}^\circ$$

12. Reduce the DC control voltage to close to 4V which comes close to a WSS frequency of 40Hz approaching the lower limit of sensing and measure the WSS signal frequency

$$V_{DC} = 4.3 \text{ V}$$

$$f_{loaded} = 34.5 \text{ Hz}$$

13. Measure the pulse wave frequency and the phase difference

$$f_{pulse} = 34.75 \text{ Hz}$$

$$\Phi_{pulse-loaded} = 18.01 \times 10^{-3}^\circ$$

14. Record the pulse waveform as *wssPulseConvertedMin.dat*
15. Connect the second probe to the Output1 pin of the MCP6541 to measure the rotational state sensor signal characteristics
16. Set the scope to trigger on the positive edge of the signal on the second probe and switch on the WER

17. Record the waveform as *wssUnlock.dat*

18. Measure the rise time of the rotational state signal

$$t_{rise-unlock} = 3.69 \mu s$$

19. Switch on the WER and set the scope to trigger on the falling edge to record the lock signal

20. Switch off the WER and record the triggered wave form as *wssLock.dat*

21. Measure the fall time of the rotational state signal

$$t_{rise-unlock} = 99.33 ms$$

22. Run the Matlab script *EDHardwareLayerSignalCharacteristics.m* to generate the following plots

The generated plots and analysis are presented in section 8.2.2.

#### ***ED discrete operations:***

1. Connect the AP and run *testMSP430SerialComm.m*

2. Connect the probe across the current sensing resistor and set the scope to trigger on positive edge of  $200mV$ .

$$R_{currentSense} = 14.1 \Omega$$

3. Turn on the power to the ED and the ED startup current consumption will trigger the signal capture

4. Record the captured waveform as *edStart.dat* and measure the total active period

$$t_{EDStart} = 15.47 ms$$

5. Once the ED has started, to record a lock event, once a message is received on the Matlab console, switch off the WER DC supply within 5 seconds. The wheel lock event transmission will trigger signal capture

6. Record the captured waveform as *wssLock.dat* and measure the active period.

$$t_{EDLock} = 2.52 ms$$

7. Reset the capture and switch on the WER. The transmission of the unlock event will trigger a signal capture.

8. Record the captured waveform as *wssUnlock.dat* and measure the active period

$$t_{EDUnlock} = 2.57 ms$$

9. Now the attribute transmission operation at a low speed will be measured. Reduce the WER DC voltage to  $5V$  and set the trigger for capture.

10. The next attribute transmission will take place within 5 seconds and will trigger a capture.

11. Record this waveform as *wssAttributeLowSpeed.dat* and measure the active period

$$t_{EDATTLowSpeed} = 94.8 ms$$

12. Note the measured frequency from the Matlab console

$$f_{lowSpeed} = 56.46 Hz$$

13. Measure the duration of the flat portion of the sensed voltage which represents the current drawn by the pulse conversion circuit. This represents 5 periods of the signal of  $53.46 Hz$  and should be close to  $88.6 ms$  and in this case is found to be  $91.9 ms$ .

14. Increase the WER voltage to its maximum of  $31.7V$  and make a note of the measured frequency.

$$f_{highSpeed} = 488.8 Hz$$

15. Trigger a measurement similar to the low speed attribute measurement and record the waveform as *wssAttributeHighSpeed.dat*. Record the active period

$$t_{EDATTHighSpeed} = 14.7 \text{ ms}$$

16. Similarly measure the MAX3096 current draw duration which should be close to  $10.2 \text{ ms}$  and in this case is found to be  $11.64 \text{ ms}$ .
17. Run the Matlab script *EDDiscreteOperations.m* to generate the plots.

The generated plots and analysis are presented in section 8.2.3.

***ED cold start:***

1. Drain the CBC-Eval-09 buffer capacitor by briefly shorting the output voltage leads.
2. Connect the AP and run the Matlab script *testMPS430SerialComm.m*
3. Connect the output terminals to the ED power lines, the system is now cold.
4. Connect the scope probe across the current sense resistor to measure the current drawn by the ED. Ensure that the settings are set to sufficiently record data for  $\sim 150 \text{ s}$ .
5. Turn up the WER voltage to its maximum and turn it on. Mark this time instance as *Mark1*
6. Wait for the scope to show a spike representing ED startup, now the system has cold started mark this time as *Mark2*
7. About  $2\text{s}$  after *Mark2* the Matlab console output should show the message *WHEEL\_UNLOCKED*
8. Allow the ED to transmit the speed attribute a few times, 5 in this case, which is visible both on the scope and the Matlab console
9. After the required number of attribute transmissions have elapsed, turn off the WER and check for the message *WHEEL\_LOCKED*
10. Record the scope waveform as *coldStart.dat*
11. Measure the time elapsed between *Mark1* and *Mark2*, this is the cold start time

$$t_{coldstart} = 62.175 \text{ s}$$

12. Run the Matlab script *EDColdStart.m* to generate the plots

The plots and analysis are presented in section 8.2.4.

***End-to-end application scenario:***

1. Drain the CBC-Eval-09 buffer capacitor by briefly shorting the output voltage leads.
2. Connect the AP and start the PM by running the Matlab script *pseudoModulator.m*
3. Connect the output terminals to the ED power lines, the system is now cold.
4. Turn up the WER voltage to its maximum and turn it on.
5. Wait for the PM to display *WHEEL\_UNLOCKED*
6. About  $5\text{s}$  after the *WHEEL\_UNLOCKED* message, the PM starts displaying the wheel speed attribute messages
7. Vary the speed of rotation by varying the WER DC control voltage and the changes in speed will be displayed by the PM at an  $5\text{s}$  interval.
8. Switch off the WER and observe the *WHEEL\_LOCKED* message being displayed instantaneously by the PM.
9. Switch back on the WER to observe the *WHEEL\_UNLOCKED* message being displayed instantaneously
10. Gradually reduce the speed by reducing the DC control voltage to below  $4V$  and observe a *WHEEL\_LOCKED* message as the voltage crosses below about  $4V$ .

An illustrative sequence of the above test case is presented in section 8.3.

## **Appendix G – Confidential information**

# **Company confidential information**

(Information available in a Volvo internal version of the report)