Wireless Vehicle Presence Detection Using Self-Harvested Energy

By

Frazer K. Noble

A thesis in partial fulfilment of the requirements for the degree of

Master of Engineering

In

Mechatronics

Massey University,

Albany,

New Zealand

2009

Abstract

Rising from the “excess demand” modern societies and economies place on limited road resources, congestion causes increased vehicle emissions, decreases national efficiency, and wastes time (Downs, 2004). In order to minimise congestion’s impacts, traffic management systems gather traffic data and use it to implement efficient management algorithms (Downs, 2004). This dissertation’s purpose has been the development of a distributable vehicle presence detection sensor, which will wirelessly provide vehicle presence information in real time. To address the sensor’s wireless power requirements, the feasibility of self-powering the device via harvested energy has been investigated. Piezoelectric, electrostatic, and electromagnetic energy harvesting devices’ principles of operation and underlying theory has been investigated in detail and an overview presented alongside a literature review of previous vibration energy harvesting research. An electromagnetic energy harvesting device was designed, which consists of: a nylon reinforced rubber bladder, hydraulic piston, neodymium magnets, and wire-wound coil housing. Preliminary testing demonstrated a harvested energy between 100mJ and 205mJ per axle. This amount is able to be transferred to a 100Ω load when driven over at speeds between 10km/h and 50km/h. Combined with an embedded circuit, the energy harvester facilitated the development of a passive sensor, which is able to wirelessly transmit a vehicle’s presence signal to a host computer. The vehicle detected event is displayed via a graphical user interface. Energy harvesting’s ability to power the embedded circuit’s wireless transmission, demonstrated the feasibility of developing systems capable of harvesting energy from their environment and using it to power discrete electronic components. The ability to wirelessly transmit a vehicle’s presence facilitates the development of distributable traffic monitoring systems, allowing for remote traffic monitoring and management.

Acknowledgements

First and foremost, I would like to offer my deepest gratitude to the mentors of this research: Dr. Johan Potgieter, Dr. Fahkrul Alam, and Dr. Peter Xu; who, with their guidance and tuition allowed for the completion of this dissertation.

For their invaluable advice and services, I would also like to thank Eddie Rodgers, Jamie McIntyre, and Gordon Hein.

As usual, the unconditional support of my family and loved ones is something always appreciated; as such, I would like to acknowledge my mother and father, Cristine and David Noble; my brother and sister, Rhys and Frances-Elise Noble; partner, Monique Hall; grandparents, Dawn and Kingsley Collins, and Stella and Jack Noble. Their support, both direct and indirect, provided a bastion of confidence during times of difficulty.

For those who I have gained knowledge from indirectly, your work has provided a rich source of information that has only furthered my own abilities, and I thank you.

Lastly, I would like to thank the staff and lecturers of Massey University's School of Engineering and Advanced Technology at Albany for the interest shown in the project and their freely given advice.

Table of Contents

List of Figures

Name **Page**

List of Tables

Name **Page**

Chapter 1

Introduction

To society as a whole, traffic congestion: when vehicles on any given road travel below its designed capacity, is undesirable. Traffic congestion misallocates resources, causes economic inefficiency, and wastes time. “Traffic delays caused by congestion worsen air pollution, raise national fuel consumption, and add heavy costs to goods delivery” (Downs, 2004); however, congestion is the principal means for allocating limited road space at peak time and a symptom of the real problem: “excess demand”. The “excess demand” placed on roads rises from modern societies’ and economies’ organisation: requiring people to commute and work similar hours so that they are able to interact effectively. While undesirable, society cannot eliminate the resulting “excess demand” (Downs, 2004), so for metropolitan areas with large populations, systems need to be in place in order to minimise the impacts of congestion.

In theory, there are four possible ways to address the disparity between supply and demand for road space: ration road space by charging drivers a user fee, expand current road networks’ capacity, increase public transport, and endure congestion. Increasing network space to handle peak time congestion results in excess space during off-peak times and may even exacerbate the issue by inducing traffic, offsetting any gain in efficiency (Kay, 1998, Downs, 2004). Congestion charging has been a feature of road schemes worldwide and is used to reduce traffic volumes during peak times, raising “road pricing” high enough to discourage travelling via private vehicle; improve access to public transport, and enable accurate traffic monitoring (Diamond Consulting Services, 2007); however, while tolling has been recommended by economists, it is seen as “grossly unfair”, discriminating against those unable to afford it, and “just another tax” in an already heavily taxed economy (Downs, 2004). Since drivers constitute a significant electoral force, governments of all levels have been hesitant to

impose this on its electorates for fear of losing the peoples' mandate to govern. Increased public transport, in theory, would greatly reduce the number of private vehicles on the road thereby reducing peak time congestion; however, the share of trips made by public transport in comparison to private vehicles is significantly less, meaning that services would have to increase significantly to match demand, and even if public transport was increased, and people used it, the resulting decrease in congestion would not be significant enough to remove congestion (Downs, 2004). Thus, the only feasible solution to congestion is to accept it and develop traffic management systems capable of monitoring traffic and optimising traffic flow in order to minimise levels of congestion and its impacts.

One such traffic management system is the New Zealand Transport Agency's Auckland Traffic Management Centre: 'ATTOMS', which manages Auckland's motorways and key regional arterial roads; improving "road user experience" by utilising traffic information gathered from technical infrastructure. ATTOMS operates 24 hours, 365 days a year, with dynamic traffic information through the Advanced Traffic Management System (ATMS) and the Auckland Region traffic signals system (SCATS). Operators constantly monitor automatic systems and live video to feed back information to partner organisations to assist in clearing traffic incidents effectively and update variable message signs (NZ Transport Agency, 2009).

As part of Auckland's ATMS, inductive loop detectors, which measure the variance of a coil loop's inductance caused by a vehicle temporarily acting as the loop's "core", are used for vehicle presence detection. A coil of wire, of several feet in diameter and containing several loops, is placed into grooves cut into the road, an inductance meter connected, and the grooves filled with filling compound. When a vehicle passes over the coil, the coils' inductance rises, and the change can be used by the ATMS to determine traffic speed and density.

When traffic becomes heavy, ATTMS sends a message via fibre optic cables to an automated server which switches on-ramp signals on. The signals manage the flow of traffic onto the motorway during peak periods and other busy times and only operate when needed to improve traffic flows; at other times the system can be switched off completely when traffic volumes decreases. The system is made up of a number of traffic detectors and associated algorithms that detect when queues are building and it changes the signal phasing to speed up traffic through the on-ramp. Depending on location, additional cameras and detectors are installed on the approach to the on-ramp to measure the ramp and local road traffic volumes (NZ Transport Agency, 2009). Data gathered has shown the following results since ATTOMS' on ramp traffic management system was commissioned:

- 12 % improved travel speeds
- 18 % increase in throughput
- Safer merging and motorway incidents being cleared up to 15 minutes faster to restore normal traffic flows

In sections of Auckland's motorways where ramp signals have been commissioned in the southbound direction between the central city and the Ellerslie-Panmure, Mt Wellington and East Tamaki Interchanges, peak period traffic flows have been significantly improved with shortened periods of congestion. The motorway is carrying significantly more traffic during peak periods than before and speeds have increased (NZ Transport Agency, 2009).
In order to provide dynamic traffic information to traffic management systems, it can be argued that embedded sensors, like those used by Auckland's ATMS, need to provide information in real time in order to be as effective as possible.

Minimising the impacts of congestion through traffic management has provided motivation for this dissertation; as such, the purpose of this research has been to investigate issues related to the development of a distributable vehicle detection

sensor and prototype a system capable of communicating a vehicle's presence in order to provide data towards the development of a traffic management system. The research has aimed at developing a system that is autonomous as possible; investigating energy harvesting in an attempt to address the wireless power requirements of distributed sensor nodes. The literature review has focused on energy harvesting, investigating the predominant harvesting methods: electrostatic, piezoelectric, and electromagnetic, discussing the underlying theory of each technique, and presenting previous research into the development of energy harvesting devices. Research has also investigated and discussed wireless sensor nodes and their architecture in order to provide a framework for which the vehicle presence detection sensor can be designed around.

The dissertation consists of six chapters: Chapter 1, Introduction, discusses the problem of congestion, traffic management, and the research's purpose; Chapter 2, Literature Review, presents and discusses previous electromagnetic, piezoelectric, and electrostatic energy harvesting research; Chapter 3, Foundation of Energy Harvesting, provides an overview of electromagnetic, piezoelectric, and electrostatic energy harvesting techniques' theory and principles of operation; Chapter 4, Vehicle Presence Detection Sensor and Reception System Design, discusses the design and development of a combined electromagnetic energy harvester, embedded circuit, wireless transceiver, and graphical user interface; Chapter 5, System Integration and Testing, discusses the testing of each individual system component and the outcomes, and integration and testing of the combined system components and the outcome; Chapter 6, Conclusion, discusses the research's contribution, the implications, and identifies areas where further research could be undertaken.

Chapter 2

Literature Review

2.1 Introduction

In the past the mobility of networks and potential operating environments of condition monitoring applications have been constrained by their reliance on access to mains power and the “wired” nature of their communication systems. The development of wireless networking standards, like IEEE’s 802.11 and 802.15.4 protocols, removed the need to be physically tethered to a network and enabled cost effective, remote access to end devices.

Removing the need for physical inspection and costly copper wiring, wireless communication facilitated the replacement of previously hard-wired devices with the development of wireless sensor nodes and condition monitoring devices. Capable of being placed into situations where hard-wired devices were impractical, such systems can be used to wirelessly transmit the state of a machine or process. Typically isolated form conventional power sources, wireless sensors require some form of wireless power. Conventionally, wireless devices operate on battery power and employ efficient energy budgets in order to last as long as possible; however, the reliance on batteries is a less than ideal solution: batteries are inherently finite devices, requiring periodic replacement or recharging; are bulky in nature, constricting the miniaturisation of devices; and contain hazardous chemicals, requiring environmentally friendly disposal. The reliance on battery power instils the need to be maintained in a wireless device; which, for applications where accessibility is an issue, presents a significant problem.

Projecting future power consumption to reduce to tens of μW to hundreds of μW (Amirtharajah and Chandrakasan, 1998), energy harvesting has been proposed

as a means of augmenting or replacing wireless devices' power supplies by converting ambient energy into electricity. Wireless devices incorporating either a hybrid battery and/or harvested energy power supply can be thought of as truly autonomous. Independent and self-contained, such systems do not need to be maintained; enabling their placement into situations where previously battery powered devices were infeasible. Example harvested energy powered applications could include: ingestible medical sensors, systems free of batteries' hazardous chemicals which harness power from blood flow or blood pressure; wireless condition monitoring devices, powered from the phenomenon being measured and wirelessly transmitting a machine's state, thereby, removing the need for physical inspection and providing real-time information; or embedded structural sensors, embedded into the material of the structure being monitored and communicating structural change wirelessly.

2.2 Wireless Sensor Nodes

Figure 1 demonstrates the general structure of a wireless sensor node with energy harvesting ability, which consists of: energy harvesting device, energy storage, processor unit, sensor, wireless transceiver, and buffer management (Niyato, Hossain, Rashid, and Bhargava, 2007).

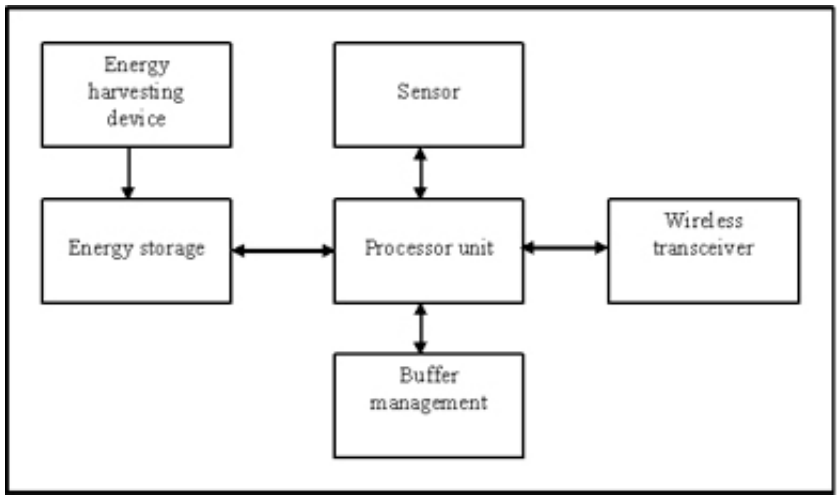

Figure 1: General structure of a wireless sensor node

As mentioned, conventional wireless devices operate on battery power, which for isolated applications may not be ideal; however, rechargeable alkaline, lead-acid, of Li-ion batteries can also be used as a reservoir for power generated by wireless sensor nodes' energy harvesting devices. Alternatively, super capacitors, electrochemical capacitors, or double layer capacitors offer a compromise between "electronic" or "dielectric" capacitors, such as: ceramic, tantalum, film, and electrolytic, and batteries; providing high energy, power, and capacitance densities (AVX, 2009). Although limitations, including: high ESR and capacitance loss exist, products like AVX's 560mF BestCap super capacitor have an ESR of approximately 0.025Ω at 1kHz (AVX, 2009). Irrespective of the device used, wireless sensor node's energy storage component's purpose is to temporarily store energy for the sensor node's components to use later.

The wireless sensor node's wireless transceiver receives and transmits application data packets and can utilise the ZigBee, 802.15.4; Bluetooth, 802.15.1; or WIFI, 802.11 standards based, wireless networking protocols. ZigBee operates in three frequency bands: 868MHz, 916MHz, and 2.4GHz, and can achieve data rates up to 250kb/s at 2.4GHz. Frequency division multiplexing allows networks to coexist within the same location, dividing the frequency bands in up to 16 channels. The 802.15.4 protocol defines two types of devices: full-function devices (FFD), which have complete network functionality and typically act as routers or co-ordinators, and reduced function devices (RFD), which support a reduced set of functions and typically are used as end devices. RFD devices are often used in sensor nodes (Nardis and Benedetto, 2007) which only require simple commands or reduced functionality in order to accommodate low power consumption as part of energy budgets. ZigBee and Bluetooth are suitable for short-range communications with ranges around 30m and 80m depending on line-of-sight. Capable of data rates in the Mb/s, WIFI's range can be used for sensor nodes that require longer transmission.

The buffer management's objective is to store incoming data packets and schedule them for future transmission. Typical wireless sensor nodes buffer sensor data and data packets received from other sensor nodes in the nodes' network; however, the size of the buffer is dependant on energy constraints placed on the system, as such, efficient buffer management schemes are required (Niyato *et al*, 2007).

Wireless sensor nodes' processors tend to be microcontrollers. Their flexibility, size, low current draw, and functionality provide an interface nodes can use to incorporate sensors and wireless transceivers into their architecture. Mid-range microcontrollers' built in ADC convert sensors' outputs into digital values, which in turn can be used to generate wireless transceivers' data packets. Energy management algorithms are often implemented via the processor unit; toggling a node's component's on and off. Most mid-range microcontrollers often incorporate built in power management function and can operate in SLEEP modes, reducing their current draw.

Wireless sensor node's energy harvesting devices convert ambient energy into electricity and can be used to augment or replace finite power supplies, like batteries. Energy harvesting has typically been implemented using piezoelectric, electrostatic, and electromagnetic approaches and focussed on harvesting energy from vibrating mediums; however, thermal, RF, and solar based approaches can also convert energy found in a sensor node's environment into electricity. Electromagnetic harvesting devices utilise Faraday's Law of Induction, harvesting energy from a coils' induced voltage as a magnet passes through it. Piezoelectric harvesting devices utilise their intrinsic piezoelectric properties to convert mechanical actuation of their active materials into electricity. Electrostatic energy harvesting harvesters act in a similar manner to capacitors; two electrodes are polarised, one electrode is pulled away from its fixed counter part, and the area between the electrodes varied, which in turn affects the

capacitance. Charge is harvested from the displaced electrode; pulling charge off the plate.

With respect to traffic management, energy harvesting can facilitate the development of a self-sufficient, traffic monitoring device, capable of wirelessly transmitting real-time traffic information; as such, the purpose of this literature review has been to investigate the feasibility of generating enough power to operate a wireless sensor node, for the purpose of vehicle presence detection, using predominant energy harvesting techniques, discuss the techniques' operating principles, and compare what has been achieved by recent researchers. Of particular interest will be the amounts of energy generated per device based on the technique used and factors affecting the harvesting of energy.

2.3 Electromagnetic Vibration Energy Harvesting

To address the "difficulty of powering remote microsytems" (Williams and Yates, 1995), Williams and Yates proposed generating electricity by harvesting energy from a vibrating medium using a 5mmx5mmx1mm electromagnetic system consisting of: a spring suspended proof-mass, magnet, coil, dashpot, and housing. Harmonic analysis led to the following design goals: proof-mass' mass should be as large as possible within device volume, proof-mass displacement should be as large as possible, spring and proof-mass' mass should be designed so that the resonant frequency matches application's frequency, load impedance should be low enough so as not to damp the proof-mass' oscillation, and the type of transducer used to convert mechanical energy into electrical has no effect on the maximum amount of power generated. For a $16mm^3$, 15mg silicon mass and damping factor of 0.3, the predicted power generated for a 70Hz vibration was 1µW and 100µW at 330Hz.

Amirtharajah's and Chandrakasan's (1998) proposed electromagnetic vibration harvester consisted of: a spring suspended, combined coil and proof-mass, fixed magnet and housing, and dashpot. Their generator had a coil impedance of 10Ω,

0.5g proof-mass, 174Nm^{-1} spring coefficient, and resonant frequency of 94Hz. For an initial displacement a peak voltage of 180mV was observed. A vibration source, with frequency and amplitude of 2Hz and 2cm, was used to simulate the average output power dissipated in the system's dashpot to be 400µW. The feasibility of powering Amirtharajah's and Chandrakasan's DSP system via energy harvesting was demonstrated by enabling up to 11,700 DSP cycles and 2,340 DSP operations.

Neil, Ching, Wong, Wen, Philip, Leong, and Wen (2002) demonstrated a magnet based micro power generator that harvested energy from the environment and converted it into electrical energy. For their generator of volume 1cm^3, AC voltages of 2.0V_{pk-pk} to 4.4V_{pk-pk} with maximum power of 200µW_{rms} to 800µW_{rms} were achieved between frequencies of 60Hz to 100Hz with no more than 200µm input vibration amplitude.

Another resonant electromagnetic harvester and integrated energy harvesting circuit is described by Cao and Lee (2006). Consisting of a wire-wound coil housing, base frame and screw, spring, and magnet, their system harvested energy from the coils' induced voltage when the magnet oscillated out of phase with the harvester's vibration. Cao's and Lee's energy harvesting circuit implemented feed-forward and feed-back DC-DC PWM voltage step-up. For a 40Hz resonant frequency, the energy harvester and harvesting circuit were able to achieve a peak open circuit voltage of approximately 16V and harvest a maximum power level of 35mW.

A series of electromagnetic harvesters were designed by Glynne-Jones, Tudor, Beeby, and White (2004) in order to supply power for an intelligent sensor network. Designed around cantilevered beams, neodymium magnets acting as proof-mass' combined with the cantilevered beam to form a resonating structure. The first harvester's magnets were oriented so that its coil was half covered; whereas, the second harvester had four magnets and fully enclosed its coil. For a resonant frequency of 322Hz, displacement amplitude 13µm, 4.73mV_{rms} was generated. A peak voltage of 8.92mV_{rms} was observed for an optimum load of 0.603Ω. The maximum power generated by the first generator was found to be

37μW. From practical experimentation, the second generator produced a peak output of 3.9mW and an average power level of 157μW.

Duffy and Caroll (2004) proposed harvesting energy from the 67W dissipated by a person as they walk; presenting two electromagnetic harvesters designed to fit in a shoe. The first generator's 5mmx10mm magnets were designed to slide along the length of the shoe, inducing a voltage in a series of connected coils. For a controlled 5Hz frequency, peak to peak voltages of 0.8V and 250mV_{rms} were observed in individual coils; however when connected together, induced voltage were shown to be negated because of the opposing polarities induced in series coils. It was demonstrated that 456mV_{rms} was achieved for direct series configured coils, and 634mV_{rms} achieved for opposing series. The second generator utilised an opposing magnet approach; where, one which magnet remained fixed to the sole of the shoe, and the other free to move in the vertical direction. For series opposition configuration peak voltages of 704mV_{pk-pk} and 43.8mV_{rms} were obtained from experimentation. Measurements were performed using 3.3Ω and 1Ω resistive loads for the sliding and opposing magnet structures. The sliding magnet generator was able to generate up to 8.5mW while the opposing magnet structure generated only 0.23mW.

Torah, Beeby, Tudor, O'Donnel, and Roy (2006) developed a cantilevered beam, vibration powered, electromagnetic energy harvester. Consisting of: cantilevered beam suspended combined tungsten proof-mass and neodymium magnets, zintec keeper, plastic base, and fixed copper coil, energy was harvested from the coils' induced voltage when it experienced a varying magnetic flux caused by the oscillating proof-mass and magnets. For a 600 turn, 25μm thick wire coil with 100Ω impedance, two cantilevered beam generators were prototyped. For 1mmx1mmx1.5mm and 2.5mmx2mmx1.5mm magnets; 57Hz and 60Hz resonant frequencies and peak voltages 87mV_{rms} and 39mV_{rms} were observed. The optimum load was determined to be 150Ω. For an optimised generator, a maximum power of 17.87μW was generated with a peak voltage of 52mV_{rms} for a 56.6Hz, 0.588ms^{-2} vibration.

Kulkarni, Kouhharenko, Tudor, Beeby, O'Donnel, and Roy (2007) presented two MEMS electromagnetic “microgenerators”; one with a wire-wound copper coil and the other an electro-deposited coil. Placed in a recessed silicon paddle designed to vibrate laterally, the coils experienced an increased change in flux; which, was caused by a soft magnetic layer concentrating bulk magnets' flux. For 106mm^2 device volumes, maximum power outputs 148nW and 23nW were achieved for 8.08kHz, 3.9ms^{-2} and 9.83kHz, 9.8ms^{-2} vibrations and load resistance of 52.7Ω.

Torah, Tudor, Patel, Garcia, and Beeby (2007) presented a vibration powered, electromagnetic system; harvesting energy to power a “radio-frequency linked microsystem”. There harvester consisted of: cantilevered beam suspended combined tungsten proof-mass and neodymium magnets, zintec keeper, plastic base, and fixed copper coil. When the harvester was vibrated the proof-mass and magnets oscillated out of phase with the vibration and the coil experienced a vary amount of magnetic flux. For a 2300 turn, 12μm thick wire coil with 1.6kΩ impedance, the generator was capable of producing 45μW$_{rms}$ of power into an optimum load of 4kΩ.

2.4 Piezoelectric Vibration Energy Harvesting

Glynne-Jones, Beeby, and White (2001) investigated the use of piezoelectric ceramics as a means of converting the strain energy in a bi-morph beam into electricity. Operating in the d_{31} mode, their piezoelectric energy harvester consisted of: cantilever beam suspended proof-mass, PZT piezoelectric material, and fixed reference point. When vibrated, the proof-mass would oscillate and load the active material, causing it to become polarised, which in turn could be harvested. The harvester's resonant frequency was 80.1Hz and its optimum load resistance for extracting power 333kΩ. For amplitudes: 0.53mm, 0.80mm, 0.90mm, power levels of: 0.6μW, 1.5μW, and 2.1μW were achieved. For beam amplitudes between 0.2mm and 1mm, load voltages between 0.21V and 1.20V were achieved.

Optimising a piezoelectric energy harvester's output, Ottman, Hofmann, and Lesieutre (2003) developed a DC-DC converter, energy harvesting circuit which increased power transferred to a load by up to 325%. For a resonant frequency of 1KHz, two 1.81”x1.31”x0.01” piezoelectric elements demonstrated a peak power output of 9.73mW. For an optimised energy harvesting circuit duty cycle, Ottman *et al* (2003) were able to harvest 30.66mW. For their frequency of operation, Ottman *et al’s* energy harvesting circuit was 65% efficient.

Kim, Clark, and Wang (2005) analysed the electro-mechanic properties of circular plate structures and piezoelectric materials. For a unmodified plate structure there was no net charge produced; the plates structure meant there was areas of tensile and compressive strain and hence positive and negative polarisation. For a regrouped electrode structure a net charge was achieved when adding polarisation constructively rather than together. For a fixed 50.8mm plate with aluminium substrate and 9.7kPa pressure, an inner radius of $.707^{th}$ the maximum resulted in maximum power generation. Experimentation validation of Kim *et al’s* (2005) analysis was performed by Kim, Clark, and Wang (2005). For a 0.127mm thick PZT layer and 0.508mm thick aluminium substrate, between 0V to 10.174V were generated for devices' capacitance of 350.8F.

For the purpose of powering a wireless sensor's transmission, Tan, Iloe, and Panda (2006) harvested energy from a piezoelectric push button igniter and stored it in a double layered “ultracapacitor”. Tan *et al’s* (2006) energy harvesting circuit rectifies the 5kV to 7kV generated, storing it in a 2.2µF capacitor which in turn clamps the DC voltage to 8V. The capacitor's voltage is then regulated and then used to power a 12-bit transmission. The peak power required during transmission and standby is 2.64mW and 0.33µW.

2.5 Electrostatic Vibration Energy Harvesting

Proposed to convert “ambient mechanical energy” into electrical energy for powering autonomous, low-power systems, Meninger, Mur-Miranda, Amirtharajah, Chandrakasan, and Lang (2001) developed a MEMS electrostatic energy harvester. For their device, 8µW of usable power was expected to be

generated from a 2520Hz vibration. When the harvester's “floating mass” is at full displacement, the system's “interdigitated” combs yield a value of 2pF; when completely closed, they yield 260pF. The maximum gap voltage is limited to 8V.

Tsutsumino, Suzuki, Kasagi, and Sakane (2006) presented the work done in developing a micro electret generator for energy harvesting applications. Using “MEMS-friendly” high performance electret material with charge density of up to $1.37mCm^{-2}$, and subjecting their system to $2mm_{pk\text{-}pk}$, 20Hz vibrations a maximum generated power of 38µW was achieved. The ideal capacitance of the system, when completely overlapping and conversely, was 17.7pF and 0pF; however, the system experienced substantial parasitic capacitance which meant the systems actual capacitance, when completely overlapping and not, was 39.8pF and 25.4pF. By using guard electrodes Tsutsumino *et al* (2006) was able to reduce parasitic capacitance down to 2.3pF so that the systems overall capacitance, when completely covered and not, was 22.4pF and 2.3pF.

A variable capacitance energy harvester was presented by Yen and Lang (2006).Their system consisted of a diode based charge pump, inductive fly back circuit, and cantilever beam variable capacitor. Vibrated at a resonant frequency of 1.56kHz, Yen and Lang’s generator was capable of generating up to 1.8µW of power via a 650pF, $43.56cm^2$ capacitor. For their system, an efficiency of 19.1% was demonstrated.

2.6 Conclusion

Typical energy harvesting systems have utilised vibration as a common source of ambient energy for harvesting. Piezoelectric, electromagnetic, and electrostatic harvesting techniques have been predominantly used and implemented an inertial approach. For a given vibration, a “spring-mass” system's proof-mass oscillates out of phase with the vibration, in turn driving the harvesting element. Electromagnetic harvesters utilise Faraday's Law of Induction, harvesting energy from a coils' induced voltage caused by an applied vary magnetic flux. Piezoelectric harvesters utilise their intrinsic piezoelectric properties, harvesting charge from the active material as it is mechanically loaded. Electrostatic

harvesters are in effect capacitors, harvesting charge from polarised plates once the area between them is varied. However, type of transducer used to convert mechanical energy into electrical has no effect on the maximum amount of power generated (Williams and Yates, 1995).

The output performance of vibration harvesting devices is affected by their system's load impedance and electrical parameters, where output voltage scale linearly with electrical parameters; resonant and active frequencies, having the resonant frequency at the operating frequency maximises displacement of the proof-mass and in turn power transferred to the load (Williams and Yates, 1995); the spring constant and proof-mass, increasing the spring constant or reducing the proof-mass' mass will increase the natural frequency of the mechanical system (Amirtharajah and Chandrakasan, 1998); the frequency and amplitude of the vibrating medium, the higher the frequency and greater the amplitude of displacement, the greater the energy transferred to the load (Roundy, 2005).

To improve output performance of energy harvesting systems, modifying electrical parameters rather than mechanical properties is more productive. Power output scales linearly with electrical parameters and scales by the square root with mechanical parameters. To maximise device mechanical performance, inertial based harvesters should maximise their proof-mass' mass and its displacement, have their proof-mass' mass and spring coefficient designed so that the system's resonant frequency matches the application's operating frequency, and damping as low as possible so as not to damp the proof-mass' oscillations. As part of Faraday's Law of Induction, magnetic field strength, the number loops in a coil, and the rate at which the magnetic flux varies have a direct impact on the voltage induced in electromagnetic energy harvesters. Maximising these parameters alongside mechanical parameters will result in improved energy harvesting.

Tables 1, 2, 3 demonstrate the amount of power generated by harvesters, application frequency, and vibrating medium's displacement, harvesting device volume, and proof-mass' mass.

Reference	Power (μW)	F (Hz)	A (ms-2)	Volume (mm3)	Mass (g)
Amirtharajah	400	94	-	-	0.5
Cao	35000	40	-	-	-
Neil	860	100	-	$1cm^3$	-
Glynne-Jones	37	322	-	-	-
	3900	-	-	-	-
Torah	17.8	56.6	0.59	150	-
Kulkarni	0.148	8080	3.9	106	-
	0.023	9830	9.8	106	-
Torah	45	56.6	0.59	150	-
Caroll	8500	5	-	-	-
	230	5	-	-	-

Table 1: Electromagnetic energy harvesters' device characteristics

Reference	Power (μW)	F (Hz)	A (ms-2)	Volume (mm3)	Mass (g)
Glynne-Jones	2.1	80.1	-	-	-
Ottman	30700	1000	-	$0.0237in^2$	-

Table 2: Piezoelectric energy harvesters' device characteristics

Reference	Power (μW)	F (Hz)	A (ms-2)	Volume (mm3)	Mass (g)
Meninger	8	2520	-	75	-
Tsutsumino	38	20	-	-	-
Yen	1.8	1560	-	$43.56cm^3$	-

Table 3: Electrostatic energy harvesters' device characteristics

Generated power levels below 100μW are believed to be insufficient to power wireless radio communication (Glynne-Jones *et al*, 2001); however, a continuous supply may not required, so energy generated can be stored temporarily (Williams and Yates, 1995). Energy harvesting circuits have been used to step-up, step-down, rectify, or modulate harvesting device's output, periodically operating wireless transceivers when enough energy was available. Up to 325% more power can be transferred to a load using energy harvesting circuits (Ottman *et al*, 2003). Cao's and Lee's (2006) energy harvesting circuit achieved an open circuit power level of 35mW. Tan *et al's* piezoelectric energy harvesting circuit

demonstrated radio transmission was possible when powered entirely from harvested energy; requiring a peak power level of 2.64mW. The voltage multiplier used in Torah *et al's* work demonstrated the ability to step up 456mV_{rms} to 2.2V.

Based on the articles reviewed, powering an embedded sensor or wireless sensor node via energy harvesting, when combined with an energy harvesting circuit, is feasible. From the three main techniques of energy harvesting, it has been shown that generating power levels in the order of µW and mW is possible. With respect to the purpose of the research, it will be investigated whether it will be possible to develop a system capable of harvesting enough energy from a vehicle presence detection sensor's environment and use it to power a wireless transmission containing data related to the detection of a vehicle's presence.

Chapter 3

Foundation of Energy Harvesting

3.1 Vibration Harvesting's Inertial Transduction Mechanism

Much of the research into energy harvesting has been based on converting a vibrating medium's kinetic energy into electricity. Piezoelectric, electromagnetic, and electrostatic energy harvesters utilise vibration's inherent displacement to drive their respective harvesting elements. The most common approach to interfacing the energy harvester's harvesting element to the vibrating medium is to use an inertial spring-mass approach. Referred herein as the inertial transduction mechanism, it contains:

- Proof-Mass
- Energy Harvester Housing
- Spring or Cantilevered Beam
- Damper

Figure 2 shows a schematic of the inertial transduction mechanism, where M, K, D, z(t), and y(t) represent the proof-mass, spring, damper, and the proof-mass' and vibrating medium's displacements as functions of time. Assuming the vibrating medium's mass is significantly larger than that of the proof-mass', and that the vibrating medium is a source of infinite energy; the vibrating medium will be unaffected by the presence of the transduction mechanism.

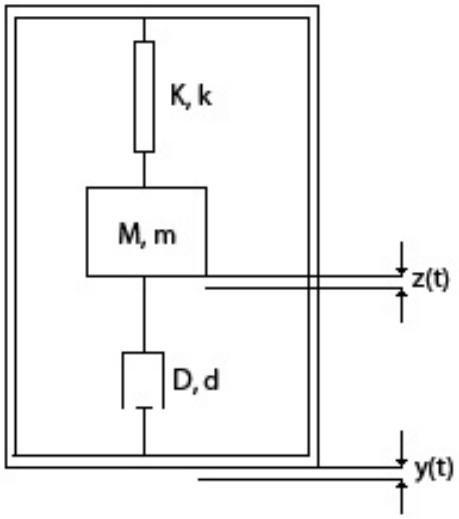

Figure 2: Inertial transduction mechanism schematic

The transduction mechanism's proof-mass' displacement can be derived from the dynamic forces acting on the proof-mass. Vibrated with the medium's displacement: y(t), the proof-mass' relative displacement with respect to the housing is:

$$m\ddot{z}(t) + d\dot{z}(t) + kz(t) = -m\ddot{y}(t) \qquad 3.1.1$$

Where, m, k, and d denote the proof-mass' mass, the spring coefficient, and the damping coefficient.

The force acting on the proof-mass is equal to that on the spring, where:

$$F = kz(t) \qquad 3.1.2$$

The instantaneous power transfer in the proof-mass: p(t), is derived from the product of the force on the mass and its velocity:

$$p(t) = kz(t)(\dot{z}(t) + \dot{y}(t)) \qquad 3.1.3$$

When driving a harvesting element, there will be damping present and a net transfer of mechanical energy to electrical energy. Assuming the vibrating medium's oscillation is sinusoidal, where $y(t) = Y_0 \cos(\omega t)$, the net transferred power will be:

$$P = \frac{m\zeta Y_0^2 (\frac{\omega}{\omega_n})\omega^3}{(1-(\frac{\omega}{\omega_n})^2)^2 + (2\zeta \frac{\omega}{\omega_n})^2} \qquad 3.1.4$$

Where, ζ, ω_n, Y_0, and ω denote the damping factor, resonant angular frequency, vibrating medium's amplitude of displacement, and angular frequency.

For a vibration powered energy harvesting device, the amount of transferred energy is dependent on the damping factor. For a large damping factor, the amount of transferred mechanical energy is greatest when the operating frequency is closest to the resonant frequency; however, as the operating frequency varies away from the resonant's, the amount of transferred energy drops quickly. In contrast, a low damping factor means that the overall transferred energy is less when the operating frequency is closest to the resonant angular frequency; however, as the frequency shifts, the amount of generated energy doesn't drop as quickly as that of a heavily damped system. Based on equation 3.1.4, it is obvious for a matching angular frequency and resonant frequency, the vibrating medium's displacement and inertial transduction mechanism's proof-mass's mass also plays a factor in transferring power; the larger they are the greater the energy transferred.

3.2 Piezoelectric Energy Harvesting

3.2.1 Piezoelectric Energy Harvesting Theory

Discovered in 1880 by Jacques and Pierre Curie, the piezo effect is the ability of certain crystalline and ceramic materials to become polarised in response

to mechanical stress or strain. The piezoelectric effect in natural materials such as rochelle salts, quartz, and tourmaline is very small; however, modern materials like barium titanate and lead zirconate titanate (PZT) demonstrate considerably improved characteristics.

Before piezoelectric materials, like PZT, are polarised, individual crystallites have symmetric unit cells. At temperatures below their Curie temperature the material's lattice structure deforms and becomes asymmetric, where individual crystallites exhibit polarisation; i.e. the PZT material's individual cells demonstrate piezoelectric properties. Because of the random distribution of the material's crystallites' domains, the material does not demonstrate an overall piezoelectric property; however, because of the material's ferroelectric nature it is possible to permanently align the domains via a process called polling; where the material is heated beyond its Curie temperature and exposed to a strong magnetic field. The magnetic field realigns the piezoelectric material's ions in the direction of the magnetic field's polarity. Following the process of polling, the material is cooled, resulting in the permanent polarisation of the piezoelectric material. At this point the material exhibits piezoelectric properties; changing dimensions when exposed to an electric field and polarising when deformed.

Because of the anisotropic nature of piezoelectric material, i.e. piezoelectric material's properties differ according to the direction of measurement, the piezoelectric effect is dependent on direction. The IEEE defines the direction in which the ions are realigned as the polling axis; labelling it the 3rd direction or Z axis. The Z axis has the same positive and negative sense as the applied magnetic field. The remainder of the coordinate system is analogous to a right-handed orthogonal system, mapping X to the 1st direction, Y to the 2nd direction, and Z to the 3rd direction.

Piezoelectric materials are characterized by several coefficients; including:

- d_{ij}: Strain coefficients (m/V) or charge output coefficients (C/N). The strain developed per unit of electric field strength applied (V/m) or charge density developed (C/m^2) per given stress (N/m^2) due to the sensor's or actuator's piezoelectric material properties
- g_{ij}: Voltage coefficients or field output coefficients (Vm/N). Open-circuit electric field (V/m) developed per applied mechanical stress (N/m^2) or strain developed per applied charge density (C/m^2) due to the sensor's or actuator's piezoelectric material properties
- k_{ij}: Coupling coefficients. The ratios describing the conversion of mechanical to electrical energy or vice versa. K^2 is the ratio of energy, mechanical or electrical, stored to energy applied

Piezoelectric coefficients with double subscripts link electrical and mechanical quantities, where the first subscript gives the direction of the electrical field associated with the applied voltage, or the charge produced. The second subscript gives the direction of the mechanical stress or strain. Several piezoelectric material constants may be written with a superscript which specifies either a mechanical or electrical boundary condition. The superscripts are T, E, D, and S, signifying:

- T : Constant Stress = Mechanically Free
- E : Constant Field = Short Circuit
- D: Constant Electrical Displacement = Open Circuit
- S: Constant Strain = Mechanically Clamped

For example, K_3^T expresses the relative dielectric constant, K, measured in the 3rd direction with no mechanical clamping applied.

The piezoelectric constants relating the mechanical strain produced by an applied electric field are termed the strain constants, or the d_{ij} coefficients. Expressed in terms of meters per volts, d_{ij}:

$$d_{ij} = \frac{\text{Strain Applied}}{\text{Applied Magnetic Field}} \qquad 3.2.1.1$$

The d_{33} coefficient denotes force acting along the 3rd axis and onto the same surface which charge is collected. The d_{31} coefficient denotes charge being collected on the same surface as before; however, the force is acting along the 1st axis. Lastly, the d_{15} coefficient denotes the charge being collected on electrodes at right angles to the polling axis and the applied mechanical stress is shear.

Piezoelectric generators that rely on a compressive strain applied perpendicular to the electrodes exploit the d_{33} coefficient of the material whilst those that apply a transverse strain parallel to the electrodes utilise the d_{31} coefficient. The power output achieved in the compressive mode can be improved by increasing the piezoelectric element's thickness or by using multi-layer stacks. Compressive loading, however is not a practical coupling mechanism for vibration energy harvesting in the majority of applications. Typically, in the case of piezoelectric films of piezoelectric elements bonded onto substrates, the elements are coupled in the transverse direction. Such an arrangement provides mechanical amplification of the stress.

The piezoelectric constants relating the electric field produced by a mechanical stress are termed the voltage constants, or the g coefficients. The units are expressed as volts per meter per Newton per square meter.

$$g_{ij} = \frac{\text{Open Circuit Electric Field}}{\text{Applied Mechanical Stress}} \qquad 3.2.1.2$$

A g_{33} coefficient indicates that the electric field and the mechanical stress are both along the polarization axis. A g_{31} coefficient signifies that the pressure is applied at right angles to the polarization axis, but the voltage appears on the same electrodes as in the g_{33} case. A g_{15} coefficient implies that the applied stress is shear and that the resulting electric field is perpendicular to the polarization axis.

High g coefficients favour large voltage output, and are sought after for sensors. Although the g coefficients are called voltage coefficients, it is also correct to say the g_{ij} coefficients are ratios of strain developed over the applied charge density. g_{ij} coefficients have units of Newton per meter over coulombs per square meter.

$$g = \frac{\text{Strain Developed}}{\text{Applied Charge Density}} \qquad 3.2.1.3$$

The relative dielectric constant of piezoelectric material: K, is the ratio of the material's permittivity: ε, to the permittivity of free space: ε_0, in an unconstrained condition.

$$K = \frac{\varepsilon}{\varepsilon_0} \qquad 3.2.1.4$$

Whereas the relative dielectric constant is strictly a material property, the capacitance is a quantity dependant on the type of material and its dimensions. Piezoelectric material's capacitance is calculated by multiplying the relative dielectric constant with the permittivity of free space: 8.9×10^{-12} F/m, and electrode surface area, then dividing by the thickness separating the electrodes. Expressed in farads, C:

$$C = \frac{K\varepsilon_0 A}{t} \qquad 3.2.1.5$$

At frequencies far below resonance, piezoelectric ceramic transducers are fundamentally capacitors. Consequently, the voltage coefficients are related to the charge coefficients by the dielectric constant as, in a capacitor, the voltage is related to the charge by the capacitance.

Equations related to piezoelectric material and capacitors are:

- q=CV
- $d_{33} = K_3^T \varepsilon_0 g_{33}$
- $d_{31} = K_3^T \varepsilon_0 g_{31}$
- $d_{15} = K_1^T \varepsilon_0 g_{15}$

Where, K_x^x is the coupling coefficient. At resonance, the dielectric constant reduces by a factor: ($1-K^2$).

The direct piezoelectric effect can be described as:

$$\Delta q = k\Delta F \qquad 3.2.1.6$$

Where, Δq is the piezoelectric electrodes' change in charge, ΔF is the force applied to the piezoelectric material, and k is a proportionality factor determined by the sensitivity of a harvesting device's piezoelectric element.

3.2.2 Principle of Piezoelectric Vibration Energy Harvesting

Piezoelectric harvesters utilise a similar transduction mechanism to that of electrostatic and electromagnetic harvesters: connecting an inertial load to

harvester housing via a cantilevered beam, diaphragm, or anchored plate. The linkage connecting inertial load to housing tends to either be made from or incorporate piezoelectric material. When the housing is combined with a vibrating medium, the inertial load vibrates out of phase; loading the active piezoelectric material and polarizing it. The polarization can be harvested via the piezoelectric element's electrodes.

Piezoelectric devices produce power effectively, even at low frequencies. Typical devices have quick response and low damping and are suitable for use in resonant devices; however, the capacitance of piezoelectric elements dominates its impedance. Typically a real load is employed to maximize power generation.

3.3 Electrostatic Energy Harvesting

3.3.1 Electrostatic Energy Harvesting Theory

The conventional arrangement of an electrostatic energy harvesting element is a parallel-plate capacitor; consisting of two isolated, parallel conductive plates of area A, separated by distance d. Assuming that the devices have no material between plates; when charged they will have equal but opposite charges. The charge q and potential difference V for a capacitor are proportional to each other; that is:

$$q = CV \qquad 3.3.1.1$$

Where, C represents the device's capacitance, which is independent of charge and voltage; instead depending on device geometry.

Once a device's geometry is known its capacitance can be calculated, assuming that there is a charge q on the plates, by calculating the electric field $\vec{E}$ and potential difference V between them. Gauss' Law can be used to relate the electric field between the capacitor's plates to charge q:

$$\varepsilon_0 \oint \vec{E} d\vec{A} = q \qquad 3.3.1.2$$

Where, q is the charge enclosed by the Gaussian surface and $\varepsilon_0 \oint \vec{E} d\vec{A}$ is the net flux through the surface.

Assuming, that whenever flux passes through the surface, the electric field vector $\vec{E}$ will have uniform magnitude E and be parallel with vector $d\vec{A}$; equation 3.3.1.2 reduces to:

$$q = \varepsilon_0 EA \qquad 3.3.1.3$$

Where, A is the area of the Gaussian surface through which the flux passes.

Potential difference between capacitor's plates is related to the electric field by:

$$V_f - V_i = -\int_i^f \vec{E} d\vec{s} \qquad 3.3.1.4$$

Where, the integral is evaluated along any path between the capacitor's plates. For a path that follows an electric field line, the vectors; $\vec{E}$ and $d\vec{s}$ dot product equates to $-E.ds$; thus, 3.3.1.4's right half of the equation reduces to:

$$V_f - V_i = -\int_-^+ E ds \qquad 3.3.1.5$$

Where, limits + and – denote the direction of the path taken between the plates; starting on the negative and ending on the positive plate.

Negating fringing, $\vec{E}$ is assumed to be constant throughout the area between the capacitor's plates. For a Gaussian surface that encloses the positive plate's charge, equation 3.3.1.3 can be rewritten as:

$$q = \varepsilon_0 EA \qquad 3.3.1.6$$

Where, A is the area of the plates. Equation 3.3.1.4 can be rewritten as:

$$V = \int_{-}^{+} Eds = E\int_{0}^{d} ds = Ed \qquad 3.3.1.7$$

Substituting equations 3.3.1.6 and 3.3.1.7 into $q = C/V$:

$$C = \frac{\varepsilon_0 A}{d} \qquad 3.3.1.8$$

Equation 3.3.1.8 clearly shows that the capacitance of a parallel plate capacitor is indeed dependant on the geometry of the device, namely its surface area and plate separation.

Work required to charge a capacitor is stored in the form of electrical potential energy U in the fields between the capacitor's plates. For a point charge transferred from one plate to the other, the voltage between the plates will be proportional to the point charge and capacitance. The work required to bring the capacitor to full charge is stored as potential energy in the capacitor, so that:

$$U = \frac{q^2}{2C} = \frac{1}{2}CV^2 \qquad 3.3.1.9$$

Equation 3.3.1.9 holds for any geometry.

If the area between a capacitor's plates is filled with an insulating material or *dielectric,* the capacitance can be increased based on the material's properties. Referred to as the dielectric constant: K; the second effect of introduction of a dielectric material is to limit the potential difference V_{MAX}, which can be applied between the plates.

The capacitance of a capacitor with dielectric material can be written as:

$$C = K\frac{\varepsilon_0 A}{d} \qquad 3.3.1.10$$

If the charge on the plates is held constant, the perpendicular force between the plates: F, is given by:

$$F = \frac{1}{2}\frac{Q2d}{\varepsilon A} \qquad 3.3.1.11$$

If the voltage between the plates is held constant the perpendicular force between the plates: F, is given by:

$$F = \frac{1}{2}\frac{\varepsilon A V^2}{d^2} \qquad 3.3.1.12$$

For an electrostatic energy harvester utilising an electret, the maximum power generated and ideal load resistance of an electrostatic generator: P_{MAX} and R_{MAX}, is a function of the charge density, plate distance, electret thickness, and overlapping surface area.

$$P_{MAX} = \frac{\sigma^2 \frac{dA(t)}{dt}}{\frac{4\varepsilon_0\varepsilon_1}{d}\left(\frac{\varepsilon_1 g}{\varepsilon_2 d}+1\right)} \qquad 3.3.1.13$$

$$R_{MAX} = \frac{1}{\varepsilon_0 \frac{dA(t)}{dt}}\left(\frac{d}{\varepsilon_1}+\frac{g}{\varepsilon_2}\right) \qquad 3.3.1.14$$

Where, ε_0, ε_1, and ε_2 are the permittivity of a vacuum, dielectric constant of an electret, and permittivity of air, and σ, d, A, and g are the surface charge, electret thickness, area, and gap distance.

3.3.2 Principle of Electrostatic Vibration Energy Harvesting

Electrostatic systems can essentially be thought of as capacitors that transduce power by performing work against charge injected into their plates. Systems generally consist of a proof-mass, parallel plate capacitor, and harvester housing; where, the proof-mass tends to be connected to one of the capacitor's plates and allowed to move out of phase with a vibrating medium. The capacitor's second electrode is connected to the harvester's housing; thus, when coupled with a vibrating medium, the proof-mass vibrates out of phase with the vibration, changing the area covered by or distance between plates, and generating power. Typical systems operate by pre-charging parallel plates at maximum capacitance and removing energy at minimum capacitance.

Electrostatic harvesters have tended towards being developed for MEMS applications due to developments made in micro-machining. For electrostatic devices, high output power and efficiency require high device capacitances, partly to overcome parasitic capacitances; unfortunately, this is a major challenge because of the difficulty of combining small gaps with long travel;

furthermore, the need for pre-charging is a disadvantage; however, this can be avoided by use of quasi permanently charged materials: electrets.

3.4 Electromagnetic Energy Harvesting

3.4.1 Electromagnetic Energy Harvesting Theory

From Coulomb's Law:

$$F = k\frac{|q_1||q_2|}{r^2} \qquad 3.4.1.1$$

We know that point charges of magnitude q_1 and q_2, separated by distance r, experience an electrostatic force of attraction of repulsion. q_1 and q_2 set-up vector electric fields, where the electric field at any point due to the charges is defined as:

$$\vec{E} = \frac{\vec{F}}{q_0} \qquad 3.4.1.2$$

Where, the magnitude of the electric field $\vec{E}$ at point P is $E = F/q_0$ and the direction of $\vec{E}$ is in the same direction as the force $\vec{F}$ acting on the point charges. The SI unit for the electric field is the Newton per coulomb: N/C.

If a point charge is moving with uniform velocity, such as current in a wire, a secondary effect takes place: magnetism. Elementary particles, such as electrons, posses an intrinsic magnetic field around them and in certain materials add together, resulting in a net magnetic field around the material, as demonstrated by permanent magnets.

The SI unit for the magnetic field is the Tesla, where:

$$1T = 1\frac{Newton}{(coulomb)(meter/\sec ond)} \qquad 3.4.1.3$$

However, the magnetic field is often given in units of Gauss, G, where:

$$1T = 10^4 G \qquad 3.4.1.4$$

The magnetic field is often referred to as the magnetic induction or the magnetic flux density; however, it is important to distinguish the magnetic field from the magnetic field strength, H, where:

$$B = \mu H \qquad 3.4.1.5$$

Which is quite different but also referred to as the magnetic field.

The amount of magnetic field that passes through a loop is called the magnetic flux, ϕ, where:

$$\phi = \int \vec{B} d\vec{A} \qquad 3.4.1.6$$

Assuming the loop lies within a plane perpendicular to the magnetic field, the magnetic flux can be written as:

$$\phi = BA \qquad 3.4.1.7$$

3.4.2 Faraday's Law of Induction and Lenz's Law

The experimental work of Michael Faraday showed that a changing magnetic field linking a wire loop introduces a voltage in the loop. The induced voltage is proportional to the rate of change of the magnetic flux through the loop,

which can change with respect to time in several ways. The loop can be fixed in space, while the magnetic field changes with time; for example, when it is produced by a magnet moving back and forth through the loop, or the wire loop can also be moving or changing its shape while in a static magnetic field. The polarity of the voltage is given by Lenz's Law: It causes a current in the wire loop which produces a magnetic field opposing the change in flux. Combining Lenz's Law, which determines the induced voltage's sign, with Faraday's experimental results, Faraday's Law can be written as:

$$v = -N\frac{d\phi}{dt} \qquad 3.4.2.1$$

Where, v is the induced voltage in the loop, ϕ is the flux linking the loop, and N is the number of turns the loop has.

3.4.3 Principle of Electromagnetic Energy Harvesting

Conventionally engineered, with few electromagnetic MEMS systems documented; typical electromagnetic harvesters consist of: spring or cantilevered arm, proof-mass, housing, permanent magnet, and wire coil. Proof-mass and magnet are usually combined in order to create an inertial load, which is suspended inside the housing due to the spring or arm. When combined with a vibrating medium, the load experiences a relative displacement, which electromagnetic harvesters utilise; driving the combined proof-mass and magnet through the coil.

Faraday’s Law of Induction dictates that a wire loop linked by a magnetic field and experiencing a varied magnetic flux has a voltage induced in it. Although the varying magnetic flux can be provided in two ways: either by moving a permanent magnet towards of away from a fixed coil or by moving a coil towards of away from a fixed permanent magnet, the majority of documented systems have used a fixed coil. The induced voltage is proportional to the rate

of change of the magnetic flux with respect to time and the number of loops in the coil.

The overall power produced by electromagnetic systems is comparable to that of piezoelectric; however, it does scale with the device size; decreasing as the device becomes smaller. One of the key factors affecting power generation is the velocity of the inertial load. At low vibration amplitude and frequency it is difficult to generate significant levels of power

As discussed, the damping factor affects the net amount of energy transferred from the inertial transduction mechanism. In order to determine the damping factor for an electromagnetic device, the damping force needs to be calculated. For an electromagnetic harvester, the damping force on a wire terminated with a load: R, moving through a magnetic field is:

$$F = \frac{(Bl)^2}{R} v \qquad 3.4.3.1$$

Where, B, l, R, v denote the magnet's flux density, the length of the wire, load resistance, and the velocity of the moving electromagnetic component. Since damping force is proportional to the velocity, the damping factor can be derived, where:

$$F = \frac{(Bl)^2}{2\omega_n mR} \qquad 3.4.3.2$$

Hence, the damping factor can be controlled by varying the electromagnetic transducer's properties and energy harvester's load in order to achieve optimum performance.

Chapter 4

Vehicle Presence Detection Sensor and Reception System Design

4.1 Introduction

The goal of the project was to develop a distributable vehicle presence detection sensor and reception system, and investigate whether energy harvesting was capable of providing enough power to operate the sensor's wireless transmission. Based on wireless sensor nodes' architecture, the proposed vehicle presence detection sensor consisted of: an energy harvesting device, rectifier, DC/DC converter, microcontroller, and wireless transceiver. For the purpose of transmitting vehicle presence data, the sensor was only required to operate passively i.e. when a vehicle drove over the sensor, the harvesting element provided enough energy to a combined rectifier, DC/DC converter, microcontroller, and wireless transceiver so that they would be able to wirelessly transmit the detection of a vehicle. When the energy harvesting device was not being actuated, the system would remain in a powered down state. The sensor's microcontroller, with non-volatile EEPROM memory, was proposed to record the vehicle's presence each time the sensor was actuated, and generate the data packets transmitted by the sensor's wireless transceiver. A host computer with a corresponding wireless transceiver was proposed to act as a receiver, processing the received data packet and presenting the vehicle presence detected event via a graphical user interface. Figure 3 depicts the proposed system's architecture and Figure 4 demonstrates the sensor's architecture. Since the proposed sensor's energy harvesting device's volume was considerably larger than those discussed in the literature review, electromagnetic harvesting's scalable performance was utilised and the harvesting device implemented an electromagnetic approach to energy harvesting. It was expected that a large

enough electromagnetic harvesting mechanism would be able to generate enough energy to operate the proposed sensor's embedded circuit and power a wireless transmission. For speeds starting at 10km/h and incrementing in 10km/h stages up to 50km/h, testing of the complete fabricated device involved driving a test vehicle over the harvesting device's bladder and recording the waveform generated across an impedance matched load.

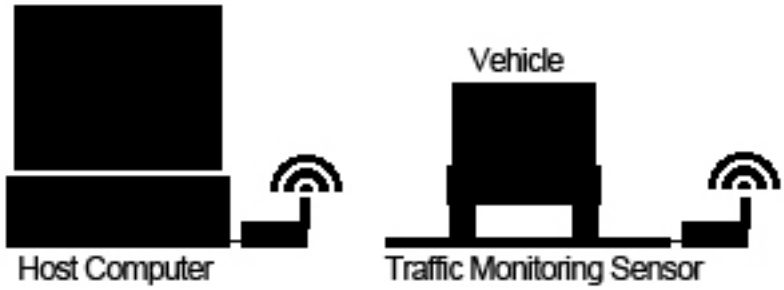

Figure 3: Traffic monitoring system's architecture. The vehicle presence detection sensor transmits the presence of the vehicle to the host computer

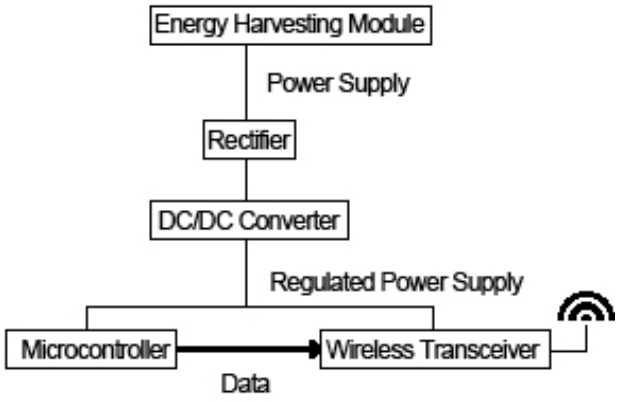

Figure 4: Traffic monitoring sensor's architecture. The energy harvesting device provides energy to the system, which is regulated and used to power the microcontroller and wireless transceiver

4.2 Electromagnetic Energy Harvester Design

The energy harvesting device's purpose was to provide enough energy to the vehicle presence detection sensor's embedded circuit in order to power the

wireless transmission of a vehicle detected event. In order to determine whether electromagnetic, electrostatic, or piezoelectric energy harvesting was to be used, the harvesting techniques' ability to integrate with a sensor circuit or system was investigated. Initial concepts looked at methods involving pressure pads; however, it was argued that in order to harvest energy from a particular point on the road, vehicles would have to consistently drive over that point; thus a “lane wide” approach was proposed. Several devices were conceptualised; including, an embedded electrostatic harvester, which covered the width of a lane and harvested energy from small deflections caused by a vehicle, and embedded piezoelectric harvesting plates; however, their embedded nature opposed the research's distributable device principle. A bladder approach was proposed and designed to run across the width of a lane, translating vehicles’ impact forces laterally and actuating a harvesting system. In theory, once the bladder was driven over a volume proportional to the bladders internal diameter and width of the tyre would be displaced and could be used to drive some form of pressure system. Piezoelectric stack actuators, which harvest energy from a d_{33} mode actuation, were proposed to be placed at the end of the bladder and harvest energy per bladder compression, as per their piezoelectric property; however, the cost of a typical stack actuator was considerable and as such, more cost effective approaches were investigated.

An electromagnetic approach was proposed to harvest energy, as per Faraday's Law of Induction, by driving a magnetic load through a series of coils per bladder actuation. Electromagnetic technologies' maturity meant that development of, and use of, complicated production techniques was not required, which meant a cost effective solution could be easily manufactured. Electromagnetic systems' “scalability”, i.e. the larger a device's electromagnetic parameters the larger the induced voltage, was also a key attraction to implementing energy harvesting via an electromagnetic approach. An electrostatic harvester was also proposed; however, it was argued that electrostatic harvesting was more suited to MEMS

applications and as such not investigated. In the end, an electromagnetic approach was decided to potentially yield the greatest power output.

The proposed electromagnetic energy harvester design, Figure 5, consisted of: a nylon reinforced rubber bladder, incompressible fluid, hydraulic piston and spring return, magnetic load, and wire-wound coil hosing. When the proposed device's bladder was driven over, the fluid contained therein would be displaced, driving a hydraulic piston's head and in turn displacing a magnet through a series of coils. As per Faraday's Law of induction, as the magnets passed through the wire-wound coil housing, a voltage would be induced in housing's coils. The induced voltage could then be harvested and used to power an embedded system. When the vehicles tires moved off the bladder the hydraulic piston's spring return caused the magnet and piston to return to its rest position; inducing a voltage of opposite polarity to that experienced by the positive displacement of the magnets.

Figure 5: CAD model for proposed electromagnetic energy harvester

As part of the harvesting device's design, different bladder materials were compared; the "atypical" garden hose was too stiff, demonstrating poor malleability and return characteristic; fuel lines, similarly were too stiff and also demonstrated poor malleability and return characteristics. Nylon reinforced rubber tubing's material; however, demonstrated excellent malleability and return characteristics. The nylon reinforced tube was able to be compressed easily, and because of the nylon reinforcement, prevented portions not compressed from expanding, which meant that the bladder's displaced contents were able to translate the compressive force. 3D mechanical CAD software: SolidWorks, was used to develop a CAD model in order to visualise the proposed harvesting

mechanism's form. SolidWorks integrated CAM package: COSMOS Motion, was used to validate the design by simulating its operation when subjected to a simulated load.

Utilising Faraday's Law of Induction, the proposed harvesting device had three main design goals: maximise magnetic field strength, maximise rate at which the magnet would be displaced, and maximise the number of the coils' loops. Since magnets' field strengths vary depending on their chemical compounds, Neodymium Iron Boron, Samarium Cobalt, Alnico, and Ceramic or Ferrite magnets' field strengths were compared; typically Neodymium magnets have the greatest magnetic field density, with permanent field strengths of up to 1.38T.

Figure 6: Wire-wound coil housing and coils

As per Duffy's and Caroll's (2004) observation of negated induced voltage in series connected coils, the harvesting device's coils were designed to be configured in "opposing series"; connecting the end of the first coil to the end of the second and the beginning of the second to the beginning of the third. In order to see whether or not this would have an impact on the quantities of induced voltage, a magnetic load was attached to the hydraulic piston's threaded shaft, pushed out to a maximum displacement, released, and pulled through the harvester's coils (Figure 6). Connected in series opposition and series, Figure 7 shows the coils' resultant induced voltages.

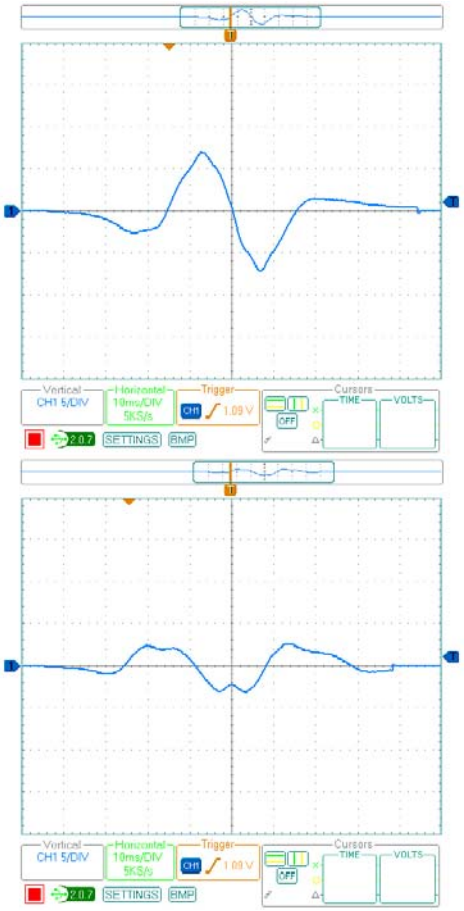

Figure 7: Opposing series and series coil configurations' voltage waveforms

In contrast to Duffy's and Caroll's observation, preliminary testing of the coils showed, irrespective of configuration, a net induced voltage; however, as Figure 7 demonstrates, the peak voltage of an opposing series configuration was considerably higher than that of the series coil configuration's. It is thought that the wire-wound coil housing's ridges, separating each coil segment, were large enough to minimise the impacts of negated voltages, if not remove it completely, which explains the disparity between series opposition coils' and series coils' induced voltages.

For Figure 7's opposing series and series coils' induced voltages, approximately 580mJ and 180mJ of energy was generated; as such, an opposing series coil configuration was used for the harvesting device's coil arrangement.

Once the design of the energy harvesting device was finalised; the device was fabricated.

The first stage involved disassembling a recycled hydraulic piston and inserting springs along its shaft. The hydraulic piston's body was placed in a lathe and a cutting tool used to remove the piston's end-stops. The piston shaft was removed and springs fed along it; thus, creating the harvesting device’s spring return. Selection of the appropriate spring was experimentally performed until a reasonable spring return characteristic was achieved. Replacement end-caps were clamped onto the combined hydraulic piston and spring return. In order for the harvesting mechanism's bladder to be attached to the hydraulic piston, the piston's housing end was machined to a diameter slightly greater than the bladder's internal diameter, and grooves added so as to prevent the bladder from coming loose. The bladder was connected to the machined end of the hydraulic piston's housing and clamped in place with a reinforce hose-clamp. A plug was fabricated to be placed in the remaining bladder end. Figure 8 shows the combined bladder and hydraulic piston.

Figure 8: Combined bladder and hydraulic piston

To see if the proposed harvesting device's mechanical design would work, the bladder was filled with oil, the plug clamped in the bladder's end, and driven over with a test vehicle’s available spare wheel. The hydraulic piston's actuation was observed; partially validating the energy harvesting device's design.

The coil housing was fabricated on a lathe; where, the wire-wound coil housing's coil segments were machined with a cutting tool. To interface the coil housing with the hydraulic piston, an adapter was machined and the piston's remaining

end threaded. Figure 9 pictures the combined wire-wound housing and piston-housing adapter.

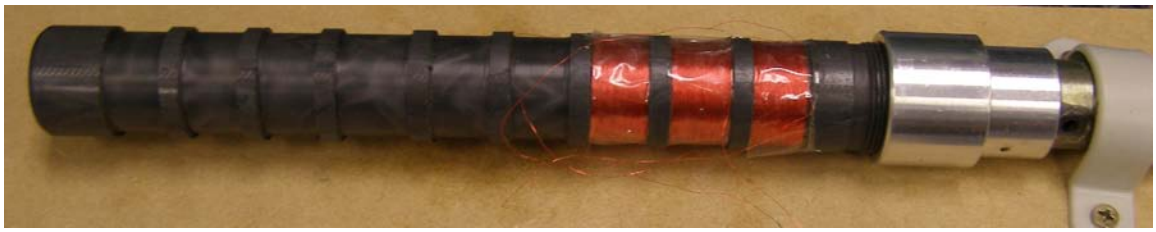

Figure 9: Wire-wound coil housing, coils, and adapter

The widths of the test vehicle's tires were measured and the volume of fluid displaced when the bladder was driven over calculated. Based on the hydraulic piston's head surface area, an average piston stroke of approximately 200mm was expected; as such, only three of the coils were initially wound onto the housing. To wind the coils on the coil housing was placed in a lathe, and used to pull wire onto the housing. To prevent injuries, the lathe was set to operate at its lowest speed.

An adapter, as pictured in Figure 10, was fabricated and used to connect the magnetic load to the piston's shaft's threaded end.

Figure 10: Magnetic load and adapter

A test platform was fabricated from a sheet of MDF and the harvesting mechanism bolted to it using hose saddles. Wooden wedges were fabricated

using a belt sander and nailed to the platform so as to hold the bladder in place when driven over.

Figure 11 shows the finished energy harvesting device and test platform.

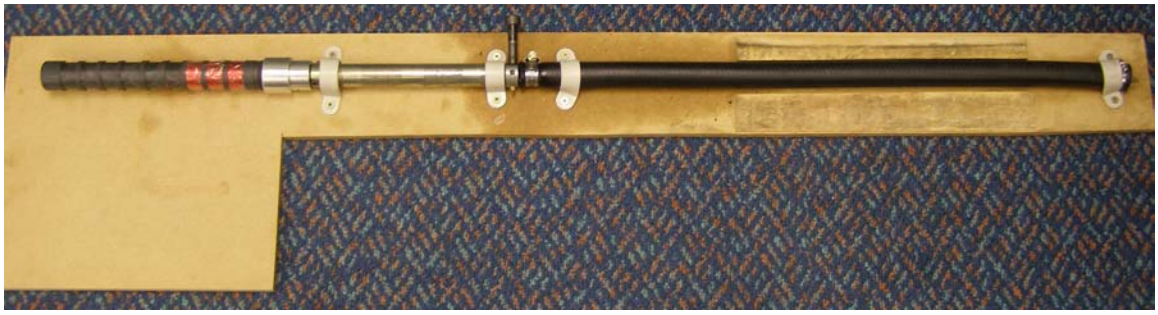

Figure 11: Completed energy harvesting device and test platform

4.3 Design of a Energy Harvesting Device Interface and Vehicle Presence Detection Embedded Circuit

In terms of detecting a vehicle's presence, the embedded circuit provided the hardware required to generate the wireless transceiver's data packets and the sensor's “sensing” functionality. While the embedded circuit's architecture (Figure 12) was based on wireless sensor nodes' and consisted of: rectifier, DC-DC converter, microcontroller, and wireless transceiver, the circuit's architecture was designed so that the combined energy harvesting device and embedded circuit formed a passive vehicle detection sensor. In theory, every time a vehicle actuated the energy harvesting device, power would be supplied to the embedded circuit and the proposed architecture's microcontroller turned on. Since the embedded circuit would only be powered when the energy harvester's bladder was driven over, the microcontroller would interpret a reset as a vehicle's presence, recording the reset and generating a data packet for the wireless transceiver; hence, passively detecting a vehicle's presence.

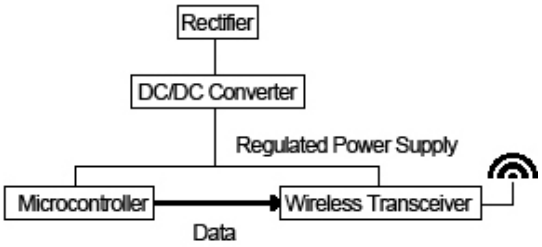

Figure 12: Vehicle presence detection sensor's embedded architecture

Based on preliminary testing of the electromagnetic energy harvesting device's rectified and smoothed 1.5V output, a DC/DC converter was designed to "step-up" or "boost" the output to 3.3V; a level compatible with discrete digital components. To minimise the circuit's power requirement, components with low quiescent current have been used. To ensure the proposed embedded circuit's energy requirements were less than the energy generated by the energy harvester, the sum of the energy required for each component was calculated. To ensure the circuit's operation, the sum of the required energy was multiplied by a factor of 2 and still less than the minimum 100mJ generated.

Microchip's range of nanoWatt microcontrollers utilise Microchip's power management technology's low power features; including, sleep currents as low as 100nA, real time clock currents of 1µA, and low power system voltage monitors; ultra fast 1µs wakeup and on the fly clock frequency change with two-speed start-up, and allowing the CPU core to run a different speed than the peripherals (Microchip, 2009). Devices such as Microchip's PIC16F688 microcontroller (Figure 13), which requires an active current of between 11µA at 32kHz, 2.0V and 220µA at 4MHz, 2.0V and a standby current of 50nA at 2.0V, were ideal for low power consuming applications such as this one.

Figure 13: Microchip's PIC16F688 microcontroller

In order to provide a regulated harvester output, Linear Technology's high efficiency, fixed frequency step-up DC/DC converters can be used. They operate from input voltages below one volt, and have active currents in the order of micro amps; minimizing power requirements in portable applications. Other features include micro amp shut-down currents, anti ringing control, open-drain power good output, and thermal shut-down and current limit. Linear Technology's LTC3401's (Figure 14) 38µA quiescent current in Burst Mode, 2.6V to 5.5V adjustable output, operating input voltage between 0.5V to 5.5V, 1A current switching, and < 1µA shut-down current made the device ideal for providing the regulated power supply required for the embedded circuit's microcontroller and wireless transceiver. Another feature of the DC/DC converter is its SHDN pin, which could be used to prevent the microcontroller and wireless transceiver from powering up if not enough energy is available. Maxim's MAX9064 "ultra-small", Low-Power, Single Comparator" (Maxim, 2008) are ideal for power constrained applications and feature: input voltages between -0.3V to 5.5V, internal $0.2V_{REF}$, operate at supply voltage as low as 1V, and consume only 700nA. The MAX9064 comparator will be used as part of a hysteresis system; preventing the LTC3401 from turning on until the harvesters output is sufficient enough to operate the sensor's embedded circuit.

Figure 14: Linear Technology's LTC3401 DC/DC converter

In line with the proposed traffic monitoring sensor's passive operation and in order to minimise power consumption, the wireless transceiver was designed to only transmit and not receive data. IEEE's 802.15.4 and ZigBee standards-based

protocols provide the infrastructure required for wireless sensor network applications, where 802.15.4 defines the physical and MAC layers, and ZigBee defines the network and application layers. For wireless sensor applications, like this project's, the ZigBee standard addresses wireless transmitter's power consumption, being designed to support low power consumption per transmission. Digi's wireless XBee RF modules provide the end-point connectivity to Drop-in networking components, enabling cost effective and an easy to deploy approach to integrating ZigBee into a sensor system. Digi's XBee 802.15.4 (Series 1) ZigBee module (Figure 15), with RF data rate between 1200b/s to 250kb/s, indoor/urban range of 100ft (30m), outdoor/RF line of sight of 300ft (100m), TX current of 45µA, RX current 50µA, and operating voltage between 2.8V and 3.4V, provides an effective method to integrating wireless communication into the traffic monitoring sensor.

Figure 15: Digi's XBee 802.15.4 (Series 1) ZigBee module

In order to rectify the traffic sensor's harvesting module's output, ON Semiconductor's MBRS130LT3 surface mount Schottky power rectifiers (Fig. 2.14) with a maximum forward voltage drop of 0.395V at 1A will be used as part of a conventional bridge rectifier circuit (Figure 16) to convert the AC nature of the generated waveform into a DC signal. Schottky diode's low forward voltage drop means that more energy will be transferred to the load rather than lost in general purpose diode's typical 0.7V forward voltage drop.

Figure 16: MBRS130LT3 surface mount Schottky power rectifier

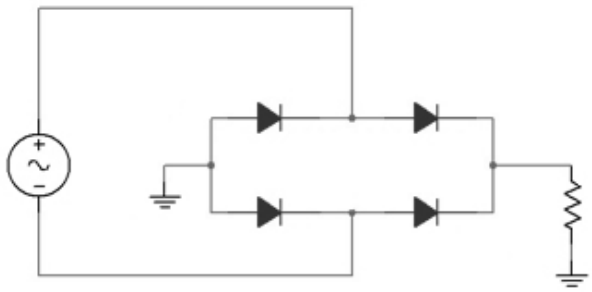

Figure 17: Rectifier schematic

Initially a general purpose diode rectifier and 1 capacitor was designed to rectify and smooth the harvesting mechanism's generated waveform; however, the forward voltage drop across the rectifier's diodes, typically 0.7V, was considered to be too great; preventing the first 0.7V generated from being harvested. Schottky diodes, with a voltage drop of approximately 0.1V to 0.3V, were proposed to be used in order to enable a greater amount of energy to be transferred to the load. Comparison of a general purpose and Schottky diode rectifier was simulated in Multisim, as demonstrated in Figure 18.

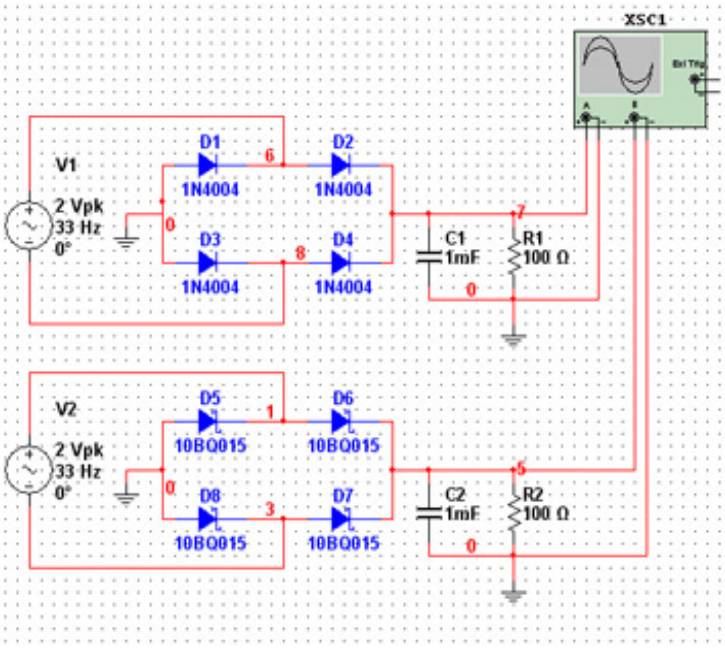

Figure 18: Comparison of general purpose and Schottky diode rectifiers

Based on the magnitude of the simulated rectifiers' voltages measured across the 100Ω loads (Figure 19), it was concluded that a Schottky diode based rectifier would transfer more energy to the load in comparison to a general purpose rectifier. For a 100ms period, the Schottky and general purpose rectifiers' waveforms' energy was calculated; from the simulation, approximately: 220mJ and 35mJ, was transferred to their respective load. The Schottky rectifier's greater transferred energy demonstrated the advantage of using a diode with low forward voltage drop and amount of *lost* energy caused by the general purpose diodes' forward bias.

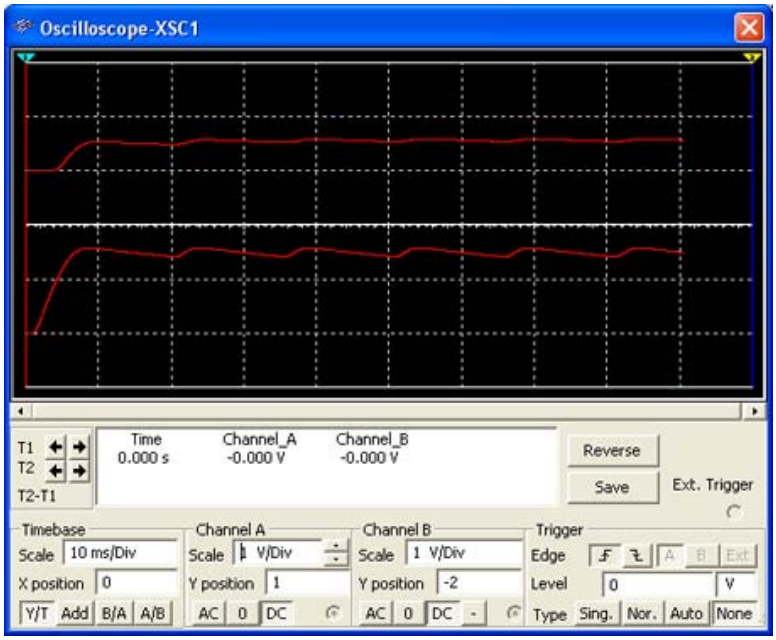

Figure 19: General purpose and Schottky diode rectifiers' output voltage measured across a 100Ω load

Having simulated the performance of a Schottky diode based rectifier, its performance was compared to that of a Dickson voltage multiplier; which intrinsic ability to rectify AC voltage meant that it could potentially provide further increased rectification efficiencies. Figure 20 demonstrates the simulated diode charge pump and rectifier comparison circuit.

Figure 21 shows the simulated outputs of the charge pump and rectifier. Essentially equal, the rectifier's output was marginally higher, by approximately 20mV, when compared to that of the charge pump. Further tested for a range of supply voltages and frequencies, the outputs remained similar; as such, the decision to implement the rectification with the Schottky diode rectifier was arbitrary.

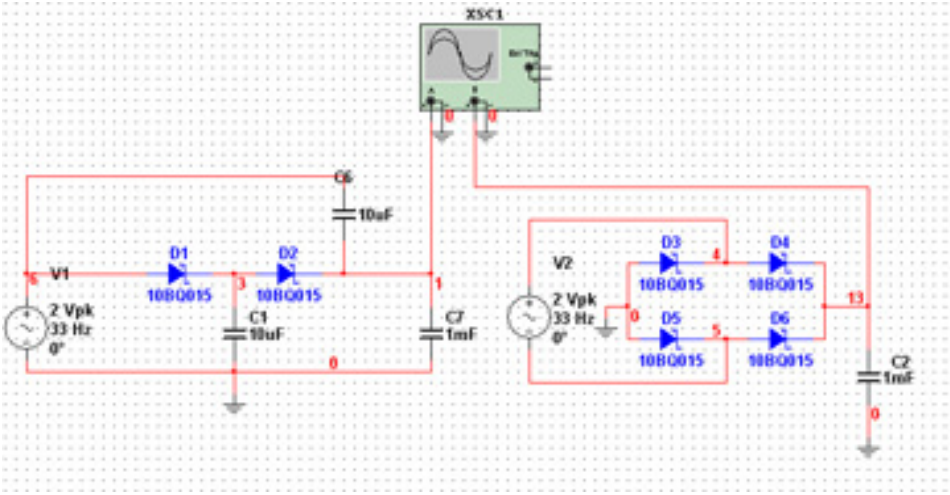

Figure 20: Schottky diode charge pump and rectifier comparison circuit

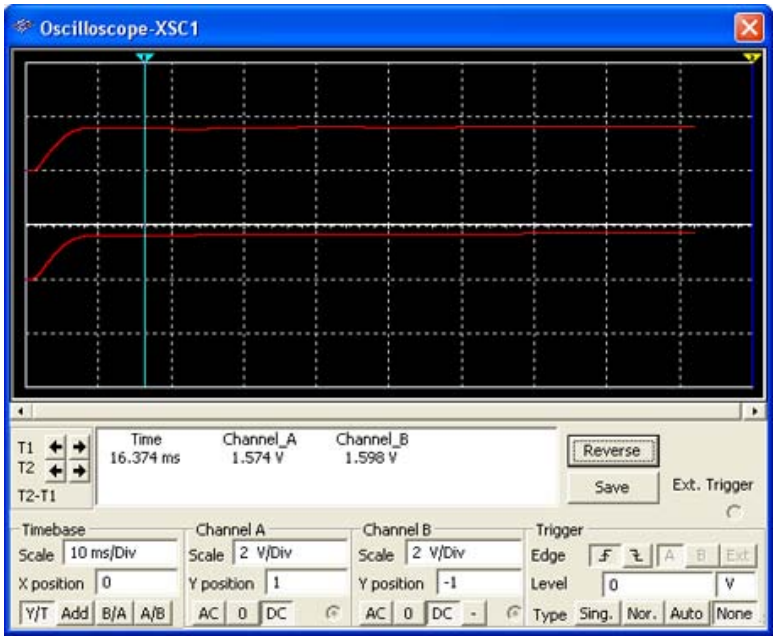

Figure 21: Schottky diode and rectifier circuit's output

Although simulated, the Schottky diode rectifier's 1.5V output closely matches the output of the energy harvesting device's actual rectified and smoothed output; as such, the embedded circuit has been designed around this voltage.

Since the energy harvester's rectified and smoothed output: 1.5V, was too low to operate discrete digital components, Linear Technologies' LTC3401 DC/DC converter has been used. The converter's low quiescent current and start-up voltage was ideal for the application and was used to step the 1.5V up to 3.3V. Figure 22 shows the DC/DC converter module's components.

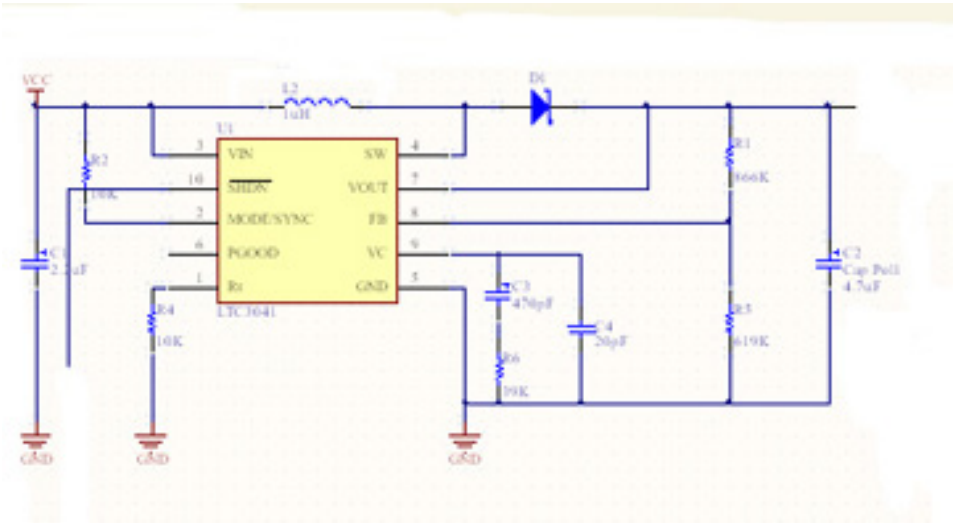

Figure 22: LTC3401 circuit schematic

In order to provide a consistent, and reliable, supply voltage to the sensor's embedded circuit's components, a soft start circuit, consisting of a comparator, was designed to prevent the LTC3401 DC/DC converter from turning on until the rectifier's output rose up to 1.5V. Pictured in Figure 23, a simulated circuit used an external 0.2V comparator reference voltage and voltage divider to set the circuit's hysteresis band's upper and lower thresholds to 1.5V and 0V. Figure 24 shows the comparator's simulated operation; going high as the voltage rises to

approximately 1.5V. The main purpose of the soft start circuit was to prevent the DC/DC converter from turning on before its supply voltage stabilised.

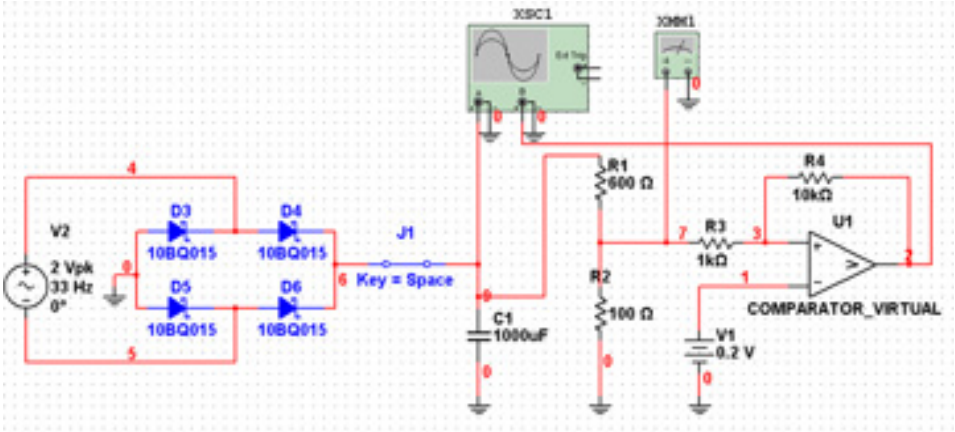

Figure 23: Rectifier and soft start circuit schematic

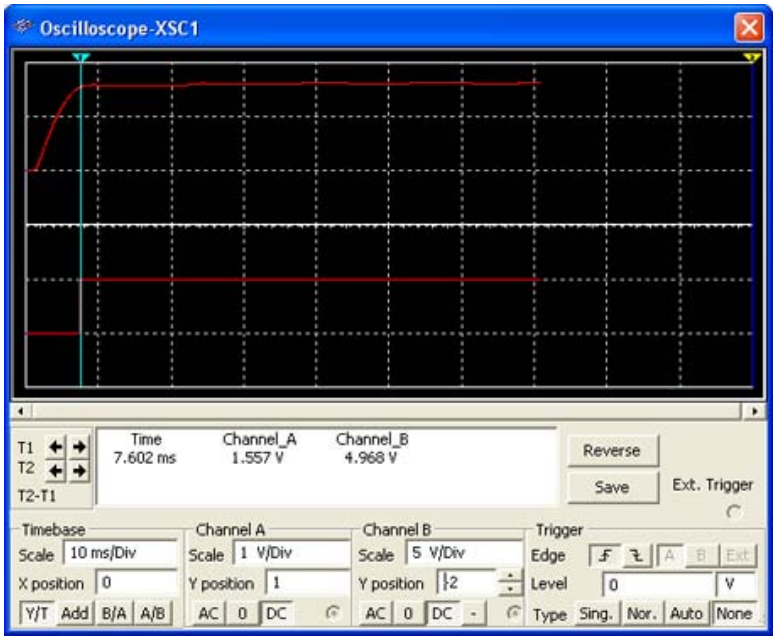

Figure 24: Rectifier and soft start circuit's simulated outputs

In order to implement the designed soft start circuit, a comparator capable of turning on, and operating, at very low voltages was required. Comparison of leading manufacturers: Linear, Texas IC, Maxim, and National Semiconductor, available comparators led to the Maxim's ultra-small, low-power, MAX9064 comparator being selected. The MAX9064's low operating voltage was ideal for

the purpose of acting as a soft-start component, preventing the harvesting circuit's DC/DC converter, embedded controller, and wireless adapter from turning on before the generator's rectified output had stabilised.

Choice of the embedded controller was based on the need to read and write to non-volatile memory and write data to a serial port. Since few system components were required, Microchip's mid-range PIC16 microcontrollers were investigated and the PIC16F688 was chosen based on its 256 bytes of EEPROM data memory, incorporation of Microchip's nanoWatt technology, and EUSART module. To interface the PIC microcontroller with the rest of the harvesting circuit, its $\overline{MCLR}$ and V_{DD} pins were tied to the positive supply rail, V_{SS} tied to ground, and its RA0/AN0/C1IN+/ICSPDAT/ULPWU, RA1/AN1/C1IN-/VREF/ICSPCLK, MCLR, V_{DD}, and V_{SS} pins connected to a RS12 in serial programmer socket.

In order to implement wireless transmission of a vehicle's presence, wireless protocol 802.15.4 and ZigBee was investigated. As discussed by previous researchers, ZigBee provides a low-power, cost effective method to integrate wireless communication into a wireless sensor application. A number of commercially available devices were looked at and the criteria for selection of the final system based on the devices' supply voltage, transmit and receive currents, RF data rate, range, and operating frequency band. Several devices were found to be suitable for the application; including:

- Meshnetic's ZigBit ZigBee Modules
    - Supply voltage: 1.8V to 3.6V
    - Transmit current: 18mA
    - Receive current: 19mA
    - Sleep current: 6µA

- Microchip's MRF24JMA ZigBee Module

- Supply voltage: 2.4V to 3.6V
- Transmit current: 23mA
- Receive current: 19mA
- Sleep current: 2µA

Initially, previous experience with Microchip products, MRF24J40MA ZigBee module's low transmit and receive currents, and the ability to be surface mounted provided an attractive option to implement wireless connectivity; however, upon further investigation is was discovered that the module required further programming of a embedded host controller and manual configuration of the ZigBee stack; complicating the design process unnecessarily, and a stand alone OEM module was looked for and found in Digi's range of XBee ZigBee modules.

Digi's wireless XBee Multi-point RF modules provide end-point connectivity to Drop-in networking components, enabling a cost effective and easy to deploy approach to integrating ZigBee into a sensor system. Inter-operable with other Drop-in network products, such as routers, end devices, and coordinators, Digi's XBee modules allow for the ability to swap-out XBee ZigBee modules with XBee ZigBee Pro devices; allowing for improved range at the cost of increased transmit and receive currents. The selected device used was Digi's XBee 802.15.4 (Series 1) ZigBee module. The key advantage of the XBee modules is the lack of need to be programmed or further developed; the OEM modules were ideal for plug-and-play applications where wireless communication can be quickly integrated.

Powered from the DC/DC converter's output, the XBee adapter Tx and Rx pins were designed to be connected to the PIC16F688's RX and TX pins; thus when the microcontroller's EUSART module transmitted a value, the XBee module received it; transmitting it wirelessly.

In order to layout the schematic of the designed embedded circuit, Altium's unified design suite: Altium Designer 6, was used. From its extensive device library, common devices could be searched for and dragged onto a newly created PCB project's *.SchDoc file's workspace. For the embedded circuit a new PCB project *.PcbPrj file was created and a blank schematic file added to it. The project's schematic's, surface mount resistors, with footprint size 0805, surface mount inductors, and through hole capacitors were dragged from Altium's Common Devices library onto the schematic. Altium's Microchip library provided the schematic and footprint for the PIC16F688 microcontroller. For devices such as the: LTC3401 DC/DC converter, MAX9064 comparator, and XBee ZigBee Adapter, custom schematics and footprints were developed. A new schematic library *.SchLib file was created and a LTC3401 entry added to the list of the library's schematics; the device's schematic shape was drawn and pins added. Next a new PCB library *.PcbLib file was created, a new footprint entry for the LTC3401 added, and the device's footprint generated based on a template MSOP footprint's dimensions. The *.PcbLib's LTC3401 footprint was then imported into the *.SchLib LTC3401 schematic entry; thus creating an integrated component. The process was repeated for the MAX9064 and XBee ZigBee adapter.

Once the energy harvesting and sensor circuit's components were connected the *.SchDoc file was compiled to ensure there was no errors. Figure 25 depicts the final embedded circuit's schematic.

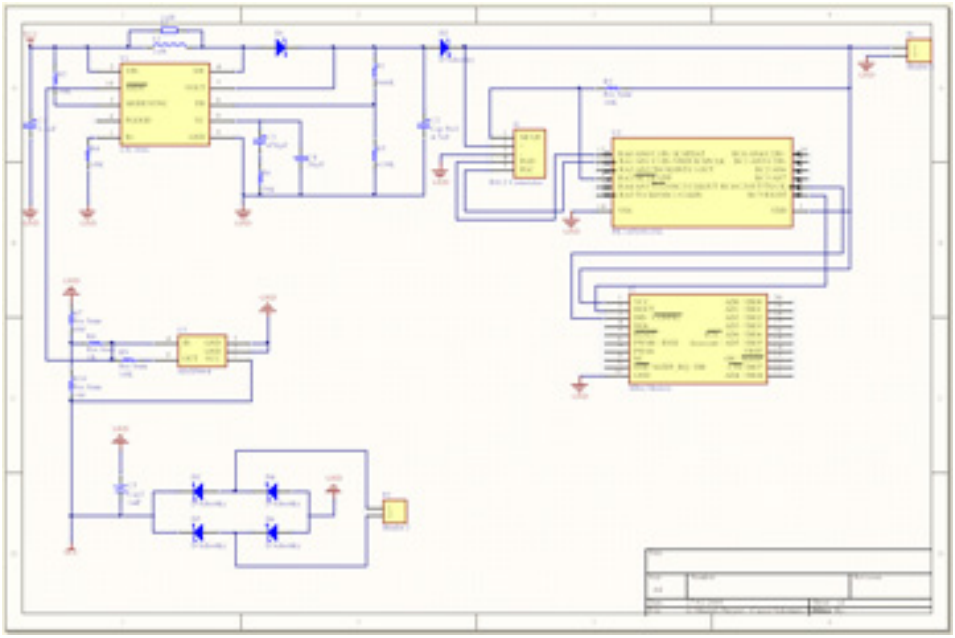

Figure 25: Sensor's embedded circuit schematic

A *.PcbDoc was created and added to the embedded circuit's project, and Altium Designer's Update PCB instruction invoked; populating the *.PcbDoc file with the footprints of the schematic's components. Devices were then dragged onto the *.PcbDoc's workspace and placed as optimally as possible. Either by manually routing or the designer suite's auto-route function, the harvesting circuit's tracks were routed. Figure 26 shows the routed embedded circuit.

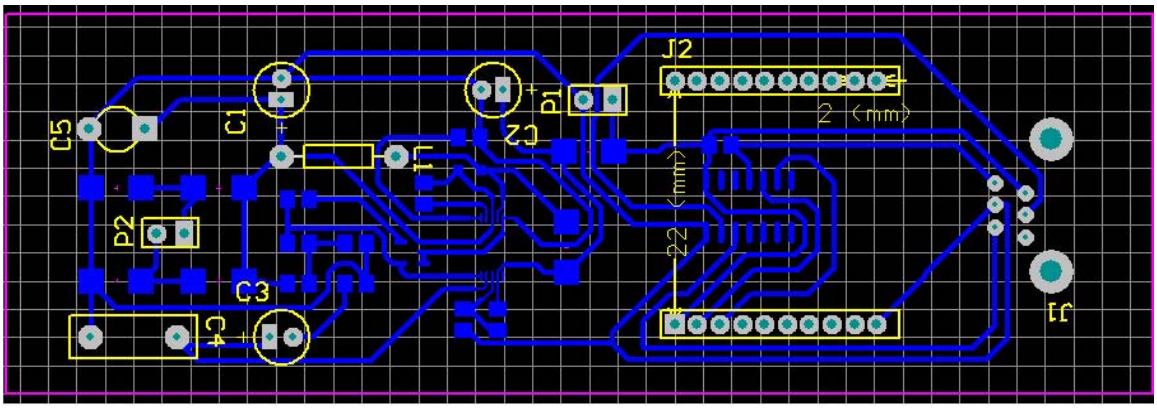

Figure 26: Routed embedded circuit's PCB board

Once the embedded circuit's PCB was populated and its layout finalised, Altium's designer suite was used to generated the *.txt, *.GBL, and *.GKO Gerber and NC-Drill files necessary for fabricating the PCB.

Once transferred to the PCB fabricator's host computer, the *.txt, *GBL, and *.GKO files were imported into the application responsible for generating the file which would be passed to the fabricator's application; where the *txt, *.GBL, and *.GKO files were used to define the bottom layer, board outline, and required drilling operations. Once the files were imported, the board outline, bottom layer, and holes were visible; a milling operation was added to the edges of the board so as to allow for the fabricated PCB to “snapped off” the master copper sheet used. After inspection of the generated file, it was saved to be imported into the fabricator's application.

Before any milling or drilling was performed, the dimensions of the master copper sheet, from which the PCB would be cut, was loaded into the fabricator's application.

The application's work-space's bottom left corner was defined by moving the camera to the vertical edge of the copper board and moving it down till the corner's vertex became visible, then the Head>Camera instruction was invoked. At this point the fabricator's tool was above the corner point, and the current co-ordinate set to 0,0. The previously generated file was imported and the board layout positioned on the application's work-space. Since the prototyped PCB was designed to be a single sided board, a Single Sided Board Phase instruction set was loaded.

The fabricator's shield was opened and copper board was then placed into the fabricator's workspace. To ensure that the milling tool's resulting cut depth was large enough to isolate the PCB's tracks, but small enough to allow for the MAX9064's MSSOP package to be soldered to the board; the fabricator's tool was loaded with a universal 2.0 mill and moved to a barren area of the copper sheet. The head was lowered and a drilling operation performed. The Camera>Head instruction was invoked and the diameter of the milled hole

shown. For a hole diameter greater than 0.25 mil of an inch the tool was raised, Head>Camera instruction invoked, the tool moved down 1mm and the process repeated. When the milling operation was perfected the tool was returned home and the first fabrication stage executed; drilling pilot holes for future drilling operations. The second and third phases drilled out the PCB's holes and milled the PCB's tracks. The board outline was milled and the milling stage to allow the board to be “snapped off” performed; at which point the physical PCB board was finished.

All that remained was for the board's components to be soldered on. Surface mount component's small footprints presented difficulty when soldering them on; the MAX9064's MSSOP package very much so. Figure 27 pictures the completed board.

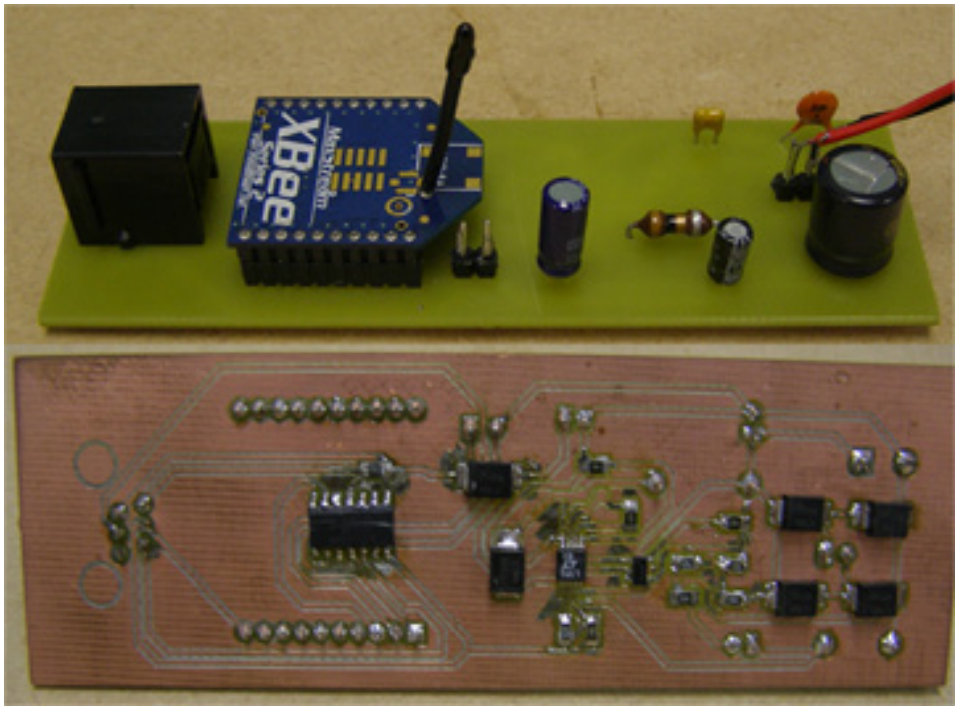

Figure 27: Sensor’s fabricated embedded circuit and components

4.4 Embedded Microcontroller’s Sensing Application

As part of the vehicle presence detection sensor's embedded circuit, Microchip's mid-range PIC16F688 microcontroller has been used to record a vehicle's actuation of the energy harvester; storing the event in the PIC's non-volatile EEPROM memory. Because the vehicle detection sensor operates passively i.e. power is only supplied to the sensor's embedded circuit when the energy harvester is driven over by a vehicle's axles, the PIC16F688 records and generates vehicle presence detection data packets in terms of vehicle axles. Differentiation between the beginning and end of a vehicle is outside the embedded application's scope; rather focussing on processing the detection of a vehicle's axle and generating the data packets required for transmission.

4.4.1 PIC16F688 Microcontroller

The microcontroller's features include:

- Wide operating voltage: 2.0V to 5.5V
- High-endurance Flash/EEPROM Cell:
    - 100,000 write flash endurance
    - 1,000,000 write EEPROM endurance
- Low power features:
    - Standby current: 50nA @ 2.0V
- Operating current:
    - 11µA @ 32kHz, 2.0V
    - 220µA @ 4MHz, 2.0V
- Enhanced USART module:
    - Auto-baud detect
    - Supports RS-232
- Incorporates Microchip's nanoWatt technology
- In-circuit serial programming (ICSP) via pins

The PIC16F688's low operating current and voltage features, enhanced Flash/EEPROM cell, and EUSART module was ideal for use with the sensor and embedded circuit. Its oscillator's wide variety of potential clock sources allows for the microcontroller to be used in a number of applications, while minimising power consumption and maximising performance. Clock modes include:

- EC: External clock with I/O on OSC2/CLKOUT
- LP: 32 kHz low-power crystal mode
- XT: Medium gain crystal or ceramic resonator oscillator mode
- HS: High gain crystal or ceramic resonator mode
- RC: External resistor-capacitor (RC) with FOSC/4 output on OSC2/CLKOUT
- RCIO: External resistor-capacitor (RC) with I/O on OSC2/CLKOUT
- INTOSC: Internal oscillator with FOSC/4 output on OSC2 and I/O on OSC1/CLKIN
- INTOSCIO: Internal oscillator with I/O on OSC1/CLKIN and OSC2/CLKOUT.

Clock sources are configured via the FOSC<2:0> bit of the configuration word.
The internal clock can be generated from two sources, HFINTOSC or LFINTOSC, depending on the frequency of the clock source. The Oscillator Control, OSCCON, register controls the system clock and frequency selection options. The OSCCON register contains the following bits:

- Frequency selection bits (IRCF)
- Frequency Status bits (HTS, LTS)
- System clock control bits (OSTS, SCS)

Able to be erased and written electrically, mid-range PICs, like the PIC16F688, are equipped with up to 256 bytes of non-volatile, electrically-erasable programmable read-only memory. The data EEPROM is both readable and writeable; however, it is not mapped directly to memory, but indirectly addressed through special function registers:

- EECON1
- EECON2
- EEDAT
- EEDATH
- EEADDR
- EEADDRH

When working with the data memory block, EEDAT holds the 8-bit read or write value and EEADDR holds the EEPROM data location being accessed. Since the PIC16F688 has only 256 bytes of data memory, the addresses range from 0x0000 to 0x00FF. Registers EEDAT and EEDATH are used when accessing programming memory; where, EEDAT and EEDATH form a two byte, 14-bit word of data, and registers EEADDR and EEADDRH form a two byte, 12-bit address.

When code protected the device CPU is able to access data and program memory; however, the programming device is prevented from doing so.
The EECON1 register provides control of access of the EEPROM memory; where, the register's EEPGD bit determines whether program or data memory is accessed. EECON1 control bits RD and WR imitate read and write operations respectively; however, for write operations EECON1's WREN bit needs to be set before writing to EEPROM memory. The PIR1 interrupt flag EEIF will set when a write sequence completes and will need to be cleared in software.

The EECON2 register is used exclusively for writing to the EEPROM data memory; reading it will result in all '0's being returned.

The PIC16F688's EUSART module is a serial input or output communications peripheral, containing required clock generators, shift registers, and buffers. Able to be configured as either an asynchronous, full-duplex or synchronous, half-duplex system, the EUSART is able to communicate with external peripheral devices. Also known as the Serial Communications Interface, the module includes the following capabilities:

- Full-duplex Asynchronous Transmit and Receive
- Two-character Input Buffer
- One-character Output Buffer
- Programmable 8-bit or 9-bit Data Packets
- Address Detection in 9-bit mode
- Input Buffer Overrun Error Detection
- Received Character Framing Error Detection
- Half-duplex Synchronous Master
- Half-duplex Synchronous Slave
- Programmable Clock Polarity in Synchronous Modes

The PIC16F688's EUSART module is also able to automatically detect and calibrate baud rates. The operation of the EUSART module is controlled through three registers:

- Transmit Status and Control (TXSTA)
- Receive Status and Control (RCSTA)
- Baud Rate Control (BAUDCTL)

4.4.2 Embedded Sensing Application Overview

The PIC16F688's ability to read and write to its EEPROM data memory allows for the number of vehicle axles driven over the energy harvesting device to be stored in non-volatile memory; preserving the vehicle axle count value even when power is removed from the PIC. The purpose of the application was to provide means to access the PIC16F688's EEPROM data memory, so as to store the count value, and transmit the value via the PIC's EUSART module.

SFRs: EECON1, EEADR, and EEDAT, are used to indirectly access the PIC16F688's EEPROM data memory. In order for the application to write data to memory, the location's address needs to be loaded into the EEADR register and the EEDAT loaded with appropriate data. With regards to the embedded application, two byte variables: EE_Count_L and EE_Count_H, have been designed to hold the upper and lower bits of a 16-bit value, so as to store the vehicle count as an integer value.

In order for the PIC16F688 to transmit the contents of the EE_Count_L and EE_Count_H bytes, the EUSART will need to be configured correctly. Since the device doesn't need to provide a master clock to peripheral devices, the serial port has been configured in asynchronous mode by clearing the TXSTA's SYNC bit. The SPBRG register will require an appropriate value to be loaded in it; which, for a 16-bit, asynchronous mode and 1Mhz clock, the baud rate generator value is: 25. The serial port will need to be enabled, so the RCSTA register's serial port enable bit, SPEN, will need to be set. To define packet size to 8-bits, the default length of the ZigBee adapter's, the TXSTA register's TX9 bit will be cleared.

The pseudo-operation of the application is outline below:

- Initialise the device, define processor, include required *.inc files, set OSCCON register, configure ports PORTA and PORTB, set ANSEL register, configure EUSART module, declare variables, in particular EE_Count_L and EE_Count_H; initialise EEPROM memory addresses.

- Read EEPROM data memory's first two addresses and load contents into EE_Count_L and EE_Count_H.
- Increment the combined EE_Count_L and EE_Count_H values, writing the results to the EUSART's TXSTA register.
- When transmitted, write the EE_Count_L and EE_Count_H variables' contents to the first two EEPROM data memory locations.

The proposed application outline should increment the value stored in the PIC16F688's EEPROM whenever the device experiences a reset caused by its MCLR pin being pulled low or having its power supply removed.

4.4.3 Embedded Sensing Application Source Code

```
;-----
; Processor
;-----
      processor 16F688
      #include <P16F688.inc>
;-----
; Error Level
;-----
errorlevel -302
```

The compiler directive processor has been used to define the embedded application's microcontroller as a PIC16F688. The #include directive adds the PIC16F688's *.inc file to the project and provides definitions for each of the microcontroller's registers.

The compiler directive errorlevel provides control of the warnings and messages generated by the MPLAB IDE. In this case, the assembler message, 302: register in operand not in bank 0, is suppressed.

The main reason for the MPASM assembler message 302 to be suppressed is because the assembler or linker can't tell which path code will take, and will always generate this message for any variable not in bank 0. Since variables may be stored in multiple banks it was decided that suppressing this warning would be helpful

```
;-----
; Configuration Bits
;-----
    __CONFIG _CP_OFF & _CPD_OFF & _FCMEN_OFF & _IESO_OFF & _BOD_OFF &
_MCLRE_ON & _INTRC_OSC_NOCLKOUT & _PWRTE_OFF & _WDT_OFF
```

The __CONFIG directive sets the processor's configuration bits, and can be placed in source code or programmed via the MPLAB IDE. The configuration options are included in the *.inc file and are bitwise ANDed together using the & character in order to declare multiple configuration bits.
From the above directive and configuration options, the PIC16F688 will have the following settings:

- _CP_OFF: Program Memory Code Protection Disabled
- _CPD_OFF: Data Memory Code Protection Disabled
- _FCMEN_OFF: Fail-Safe Clock Monitor Disabled
- _IESO_OFF: Internal External Switch-over Mode Disabled
- _BOD_OFF: Brown-Out-Reset Disabled
- _MCLRE_ON: Pin Function is
- _INTRC_OSC_NOCLKOUT: INTOSCIO Oscillator: I/O Function on RA4/OSC2/CLKOUT Pin, I/O Function on RA5/OSC1/CLKIN
- _PWRTE_OFF: PWRT Disabled
- _WDT_OFF: WDT Disabled

Where the program and data memory is not code protected, so that variables stored in both can be read and written to; the PIC16F688's pin is used to generate a reset when pulled low; the internal oscillator is used for the device's clock source; the power-up and watchdog timers are disabled since they are not used.

```
;-----
; Macros
;-----
Bank0 MACRO
    bcf STATUS,RP0
    bcf STATUS,RP1
```

```
    ENDM
Bank1 MACRO
    bsf STATUS,RP0
    bcf STATUS,RP1
    ENDM
Bank2 MACRO
    bcf STATUS,RP0
    bsf STATUS,RP1
    ENDM
Bank3 MACRO
    bsf STATUS,RP0
    bsf STATUS,RP1
    ENDM
```

As discussed, because of the PIC16F688's architecture registers are stored across multiple memory banks. The STATUS register's bank select bits, RP<1:0>, are used to select bank 0 through to bank 3. The above macros remove the need to directly change the bank select bits; instead, only requiring a macro to appropriate bank macro to change the current bank.

```
;-----
; Variables
;-----
    cblock 0x20
    temp_wreg
    temp_status
    temp
    delay_L
    delay_H
    endc
EE_Addr_L equ 0x70
EE_Addr_H equ 0x71
EE_Addr equ 0x72
EE_Data equ 0x73
EE_Count_L equ 0x74
EE_Count_H equ 0x75
```

The cblock directive declares a block of program memory starting from a initial address. Variables temp_wreg, temp_status, temp, delay_L, and delay_H are declared using the directive; where, temp_wreg is stored at address 0x20, temp_status is stored at address 0x21, temp is stored at 0x22, delay_L is stored at 0x23, and delay_H is stored at 0x24. The variables EE_Addr_L, EE_Addr_H, EE_Data, EE_Count_L, and EE_Count_H are

declared using the equ directive, where each variable is equal to the value defined.

Variables temp_wreg and temp_status are used to hold the WREG and STATUS registers' data values as part of the interrupt service routine and variables delay_L and delay_H are used as part of a delay function. Variable temp is used as a debugging variable to ensure data is transferred between registers correctly.

Variables EE_Addr_L, EE_Addr_H, EE_Addr, EE_Data, EE_Count_L, and EE_Count_H are used as part of the functions used to read and write to the PIC16F688's EEPROM.

```
;-----
; Program
;-----
      org 0x0000 ; Reset Vector
      goto Initialisation
      org 0x0004 ; Interrupt Vector
      movwf temp_wreg
      swapf STATUS,W
      movwf temp_status
;-----
; Interrupt Service Routines
;-----
      goto ISR_Done
      ISR_Done:
      swapf temp_status,W
      movwf STATUS
      swapf temp_wreg,f
      swapf temp_wreg,W
      retfie
;-----
; Initialisation
;-----
Initialisation: ; Pic Initialisation Routine
      Bank1
      movlw b'00100000'
      iorwf TRISC,f
      movlw b'11111111'
      iorwf TRISA,f
      clrf ANSEL
      movlw b'01100001'
      movwf OSCCON
      Bank0
      clrf TMR0
      movlw .25
      movwf SPBRG
      bcf TXSTA,TX9
```

```
bsf TXSTA,TXEN
bcf TXSTA,SYNC
bsf TXSTA,BRGH
bsf RCSTA,SPEN
goto Main
```

Starting at the reset vector address 0x0000, the first instruction is a branch to the initialisation code. The Bank1 macro is called and the STATUS register's bank select bits set correspondingly. The binary value b'00100000' is written to the WREG and an inclusive or operation with the TRISC register performed, the output of which is put into the TRISC register. The EUSART module's TX and RX pins are part of PORTC, so the purpose of setting and clearing the TRISC's corresponding bits enables the pins to receive input and transmit. The binary value b'11111111' is written to the WREG and an inclusive of operation with the ANSEL register is performed. This sets all PORT register pins as digital inputs or outputs. The binary value b'01100001' is written to the WREG and transferred to the OSCCON register. The OSCCON register controls the PIC16F688's oscillator setting, and with the previous binary value loaded into it sets the clock to 4MHz and uses the internal oscillator to provide the clock source. Macro Bank0 is called and the TMR0 register cleared. The literal decimal value 25 is written to the WREG and transferred to the SPBRG register. The SPBRG register controls the baud rate of EUSART serial port. The value 25 is taken from the data sheet and corresponds to a baud rate of 9600. The TXSTA register's TX9 bit is cleared, setting the length of data packets to 8-bits;the TXSTA register's TXEN bit is set, enabling the EUSART's transmitter; the TXSTA register's SYNC bit is cleared, configuring the device to operate in asynchronous mode; the TXSTA register's BRGH bit is set, in order to achieve the desired baud rate, and the RCSTA register's bit SPEN is set, in order to enable the EUSART serial port. The last instruction branches to the main routine.

Starting at the interrupt vector 0x0004, the first instruction loads the temp_wreg with the current WREG register's data and swaps the STATUS register's with the WREG's. The reason why a swapf instruction is used is

since swapf doesn't change the WREG's contents. Then the WREG, containing the STATUS register's contents, is transferred to temp_status. Once the WREG and STATUS registers are stored the program is able to execute any interrupt service routines. Afterwards, the ISR_Done routine is branched to and the temp_wreg and temp_status registers' contents written to the WREG and STATUS registers. The retfie instruction returns control from the interrupt service routine to the Main routine of the program.

```
EE_Read:
    Bank1
    movf EE_Addr,W
    movwf EEADR ; Move WREG into EEPROM address
    bcf EECON1,EEPGD
    bsf EECON1,RD
    movf EEDAT,W
    Bank0
    return
EE_Write:
    Bank1
    Wait2Start:
    btfsc EECON1,WR
    goto Wait2Start
    movf EE_Addr,W
    movwf EEADR
    movf EE_Data,W
    movwf EEDAT
    bcf EECON1,EEPGD
    bsf EECON1,WREN
    movlw 0x55
    movwf EECON2
    movlw 0xAA
    movwf EECON2
    bsf EECON1,WR
    nop
    nop
Wait2End:
    btfsc EECON1,WR
    goto Wait2End
    bcf EECON1,WREN
    Bank0
    return
```

Since the registers responsible for reading and writing to the EEPROM data memory are in bank1, macro Bank1 is called and the STATUS register set accordingly. The contents of the EE_Addr variable is written to the WREG and written to the EEADDR register. The EECON1 register's EEPGD bit is

cleared, configuring the register to access data memory. Next the EECON1 register's RD bit is set, initiating a read sequence. The contents of the EEPROM data memory's EEADDR address is loaded into EEDATA and written to the WREG. The Bank0 macro is called and the method returns to the Main routine.

The EE_Write routines process of writing to the EEPROM's data memory is considerably more complex than reading. Again, the first instruction is a call to Bank1. The next instruction checks to see if there is currently a write operation being performed; if so, the routine loops till finished. The desired EEPROM address is written to the WREG and transferred into the EEADDR register. The data to be stored in the data memory is taken from the EE_Data variable, stored in the WREG, and written to EEDATA register. The EECON1 register's PGD bit is cleared, configuring the register to access data memory. Next the EECON1 register's bit WREN is set and write sequences enabled. Literal hexadecimal value 0x55 is written to the WREG and transferred to the EECON2 register. Then literal hexadecimal value 0xAA is written to the WREG and transferred to the EECON2 register. The process of writing 0x55 and 0xAA to the EECON2 register is a required stage of writing data to the EEPROM data memory. EECON1's WR bit it is set, initiating the write operation.

The Wait2End routine polls the ECCON1's WR to check to see if the write sequence completes. Due to PIC16F688 silicon errata it is not possible to use interrupts for checking the end of write sequences; The EEIF flag may be cleared inadvertently when performing operations on the PIR1 register simultaneously with the completion of an EEPROM write. This condition occurs when the EEPROM write timer completes at the same moment that the PIR1 register operation is executed. Once the write sequence has completed EECON1's WREN bit is cleared, preventing further writes to the EEPROM.

The Bank0 macro is called and control returned to the Main routine.

```
Transmit:
    Bank0
    BusyWait:
    btfss PIR1,TXIF
    goto BusyWait
    movwf TXREG
    Bank0
    return
```

The transmit routine first calls the Bank0 routine, since the TXREG is located in the first bank. The peripheral interrupt register, PIR1, is polled to check if the transmit register is busy, if not the contents of the WREG is transferred to the TXREG and the EUSART module transmits it. Control is return to the Main routine.

```
Delay:
    ;2493 cycles
    movlw 0xEC
    movwf delay_H
    movlw 0x03
    movwf delay_L
Delay_0
    decfsz delay_H, f
    goto $+2
    decfsz delay_L, f
    goto Delay_0
    ;3 cycles
    goto $+1
    nop
    ;4 cycles (including call)
    return
```

The Delay routine uses instruction cycles to implement a simple delay method. A two byte counter is decremented 2493 times in order to implement a 13ms delay.

```
;-----
; Main
;-----
Main: ; Main Program Function
    movlw 0x00
    movwf EE_Addr
    call EE_Read
    movwf EE_Count_L
    movlw 0x01
```

```
    movwf EE_Addr
    call EE_Read
    movwf EE_Count_H
    movlw 0xFF
    subwf EE_Count_L,W
    btfsc STATUS,Z
    goto Increment_H
    goto Increment_L
Increment_L:
    incf EE_Count_L,W
    movwf EE_Count_L
    goto End_TX
Increment_H:
    incf EE_Count_H,W
    movwf EE_Count_H
    goto End_TX
End_TX:
    call Delay
    movf EE_Count_L,W
    call Transmit
    movf EE_Count_H,W
    call Transmit
    movf EE_Count_L,W
    movwf EE_Data
    movlw 0x00
    movwf EE_Addr
    call EE_Write
    movf EE_Count_H,W
    movwf EE_Data
    movlw 0x01
    movwf EE_Addr
    call EE_Write
    sleep
    goto$
```

The first instruction of the Main routine writes the literal hexadecimal value 0x00 into the WREG and transfers it to the address of the EE_Addr, so that the contents of EE_Addr is equal to 0x00. The EE_Read routine is called and the contents of the EEPROM address 0x00 returned and written to the WREG. The contents of the WREG are then transferred into the EE_Count_L register. Literal hexadecimal value 0x01 is written to the WREG and transferred into the EE_Addr variable's address. The EE_Read routine is called again and the contents of the EEPROM's 0x01 memory address returned and written to the WREG. The contents of the WREG are then transferred to the EE_Count_H register.

Literal hexadecimal value 0xFF is written to the WREG and subtracted from the contents of the EE_Count_L register and the results stored in the WREG. If the resultant was 0 then the contents of the low byte EE_Count_L register is equal to 255 and the high byte register is incremented.

The last instruction branches to the End_Tx routine.

The Delay routine is initially called in order to wait 3000 instruction cycles or 10ms so that the ZigBee module is able to start correctly and receive data written to the EUSART's TXREG register. If the ZigBee module has not had enough time to start-up the microcontroller would end up writing to the device before it could process the data.

The EE_Count_H variable's contents is written to the WREG and the Transmit function called, writing the data to the transmit buffer. The same is repeated for the EE_Count_H variable's contents.

Once the count variables are transmitted they are written into the PIC16F688's EEPROM data memory.

4.5 Wireless Transceiver Design

Since the purpose of the wireless transceiver was to provide the means to interface the combined energy harvester and embedded circuit with the host computer; the first stage of the transceiver's development was to investigate means for the host computer to connect to the XBee module. Initially it was thought to implement the Universal Serial Bus (USB) stack using one of Microchip's PIC18F4550 microprocessors; however, much like the MRF24J40MA module, the PIC18F4550 also needed to be further programmed; presenting unnecessary development delay. As such another option was looked for and found in Future Technologies Design International ltd's (FTDI) range of USB products.

FTDI specialize in converting legacy peripherals to USB and provided the easiest route to implementing communication via USB by combining their USB-Serial and USB-FIFO products with their royalty-free USB Virtual COM Port driver

(VCP). The VCP driver greatly reduced the development time of the end application, as discussed in the next chapter, eliminating the need for a dedicated USB driver to be developed.
For the wireless transceiver, FTDI's FT232R USB-UART IC was an ideal choice; the IC provided a USB to asynchronous serial UART interface while handling the entire USB protocol in a single chip. The FT232R provided the means for communication between the XBee module's UART and the host computer's USB port.

Using Altium's unified designer suite: Altium Designer 6, the wireless transceiver's schematic and PCB design followed a similar process to that described for the embedded circuit; common devices, such as resistors, were dragged from the Common Devices library onto the *.SchDoc file's workspace and their surface mount footprints set to 0805. The FT232R's schematic and footprint was provided from the FTDI library file and dragged onto the schematic file's workspace. For the PMOS and XBee ZigBee adapter custom schematic and footprint libraries were designed and included in the *.SchDoc file. Once the layout of the circuit was completed, it was compiled and the *.PcbDoc file populated via the Update PCB instruction. Figure 28 shows the final schematic of the wireless transceiver.

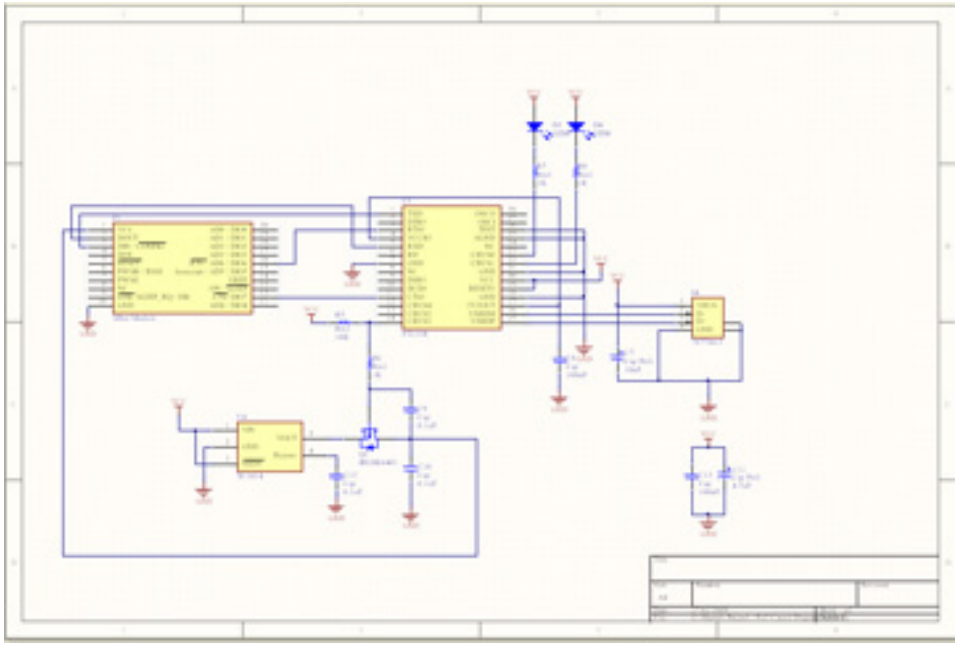

Figure 28: Wireless transceiver's circuit schematic

As demonstrated in the above schematic, the wireless transceiver derived its supply voltage from the host computer's USB port; while a 3.3V low drop-out (LDO) regulator was used to provide power to the XBee module. The FT232R's pins DP and DM were connected to the USB-B socket which provided the data signals to the chip. The FT232R's pins TXD and RXD were connected to the corresponding XBee data in and data out pins in order to properly communicate with the module's UART. The default CBUS values were used to drive the RX and TX light emitting diodes (LED), with RXLED# and TXLED# connected to the FTDI's CBUS1 and CBUS2 pins via 270Ω series resistors.

As per the requirement of USB bus powered applications, when in USB suspend mode the application, including external logic, can only draw a total current of less than 2.5mA. The FT232R provided an effective method of minimizing current draw via the PWREN# pin; which, as demonstrated in the previous figure, controls the gate of an International Rectifier IRLML6402 P-Channel MOSFET. The circuit consisting of a 1kΩ series resistor and 0.1µF capacitor was used to limit the current surge from when the MOSFET turns on, else without it the

transient power surge, caused when the MOSFET turns on, will reset the FT232R or the USB host controller. To ensure the PWREN# pin is powered from the VCCIO at all times, the VCCIO pin was connected to the 3V3OUT pin on the FT232R. The purpose of connecting the VCCIO pin to 3V3OUT pin was double, not only ensuring the PWREN# pin to function correctly but also setting the voltage level of the FT232R's digital input and output pins to 3.3V as required to communicate with the XBee DOUT and DIN pins.

Placed as optimally as possible, the PCB file's components were routed using a combination of Altium Designer's auto-route tool and manual routing. Figure 29 pictures the routed wireless transceiver PCB.

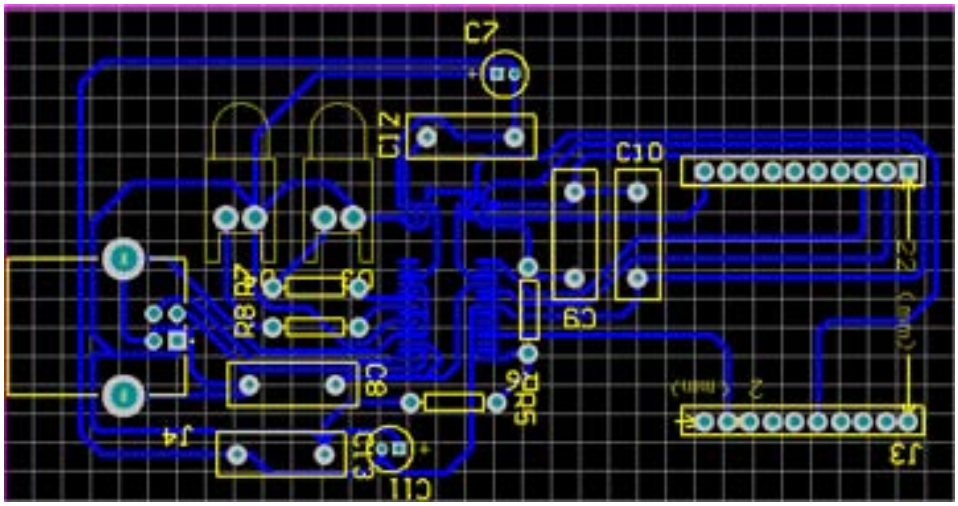

Figure 29: Routed wireless transceiver's PCB

Following almost the exact fabrication procedure as the embedded circuit's fabrication; Altium's designer suite was used to generated the wireless transceiver's *.txt, *.GBL, and *.GKO Gerber and NC-Drill files necessary for fabricating the PCB.

Once transferred to the PCB fabricator's host computer, the wireless transceiver's files were imported into the application responsible for generating the file which would be passed to the fabricator's application. The *txt, *.GBL, and

*.GKO files were used to define the bottom layer, board outline, and required drilling operations. Once the files were imported, the board outline, bottom layer, and holes were visible; a milling operation was added to the edges of the board so as to allow for the fabricated PCB to “snapped off” the master copper sheet used. After inspection of the generated file, it was saved to be imported into the fabricator's application.

Before any milling or drilling was performed, the dimensions of the master copper sheet, from which the PCB would be cut, was loaded into the fabricator's application.

The application's work-space's bottom left corner was defined by moving the camera to the vertical edge of the copper board and moving it down till the corner's vertex became visible, then the Head>Camera instruction was invoked. At this point the fabricator's tool was above the corner point, and the current co-ordinate set to 0,0. The previously generated file was imported and the board layout positioned on the application's work-space. Since the prototyped PCB was designed to be a single sided board, a Single Sided Board Phase instruction set was loaded.

To ensure that the milling tool's resulting cut depth was large enough to isolate the PCB's tracks, but small enough to allow for the MAX9064's MSSOP package to be soldered to the board; the fabricator's tool was loaded with a universal 2.0mm mill and moved to a barren area of the copper sheet. The head was lowered and a drilling operation performed. The Camera>Head instruction was invoked and the diameter of the milled hole shown. For a hole diameter greater than 0.25mil of an inch the tool was raised, Head>Camera instruction invoked, the tool moved down 1mm and the process repeated. When the milling operation was perfected the tool was returned home and the first fabrication stage executed; drilling pilot holes for future drilling operations. The second and third phases drilled out the PCB's holes and milled the PCB's tracks. The board outline was milled and the milling stage to allow the board to be “snapped off” performed; at which point the physical PCB board was finished.

Figure 30 pictures the wireless transceiver's fabricated board after components were soldered on.

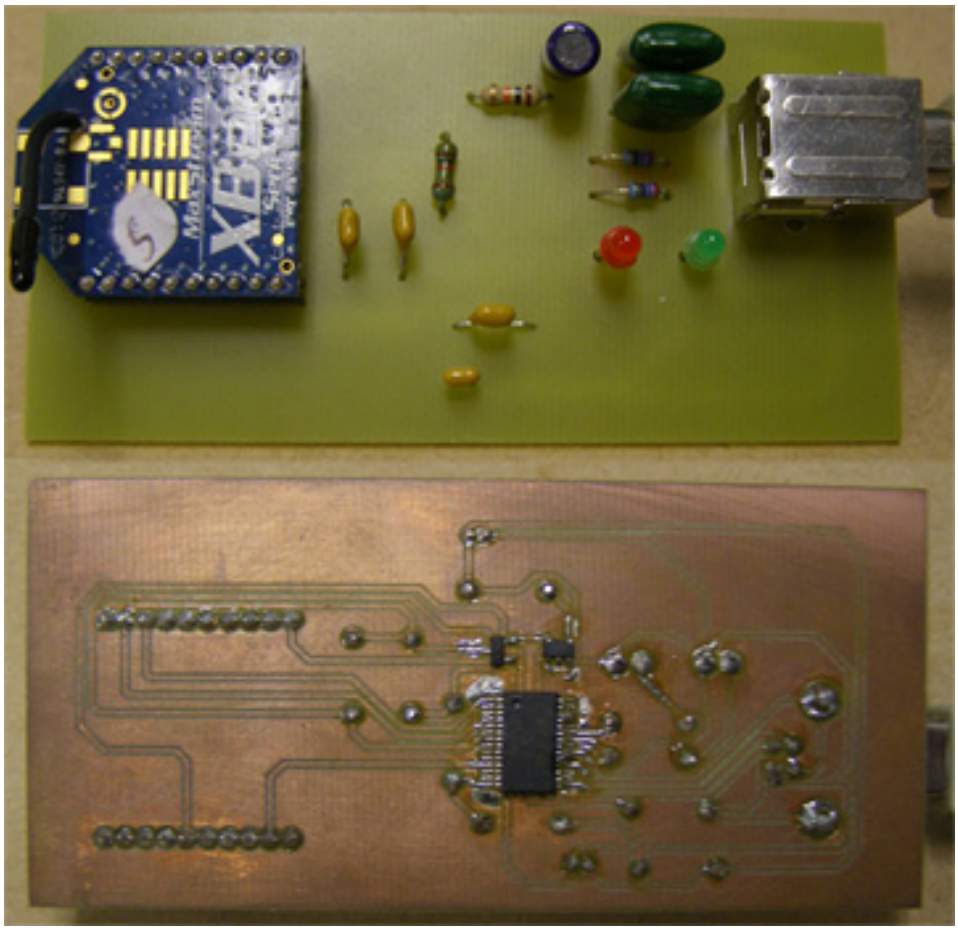

Figure 30: Wireless transceiver's fabricated board and components

4.6 Vehicle Detected Graphical User Interface

4.6.1 Graphical User Interface Application Overview

Designed to provide an interface between the end-user and wireless transceiver's and host computer's hardware, the end application was developed using Microsoft's .Net framework and programmed in Microsoft's Visual C# Studio. The .Net framework's programming interface provided a comprehensive abstraction of the WIN32 API; making it possible to develop a non-trivial graphical user interface in a relatively short period of time. The

proposed application accessed the host computer's serial port service through the .Net framework, receiving data transmitted by the wireless transceiver's FT232R's UART. Received data packets were processed and passed to the application's user interface's methods, and displayed as an axle count at the time received. The proposed application's graphical interface was designed to also provide a visual method of setting the serial port's baud rate, parity, packet size, stop bits, flow control, and COM port.

4.6.2 Graphical User Interface Design

Creating a new Windows Forms Application by selecting the Windows Forms template, the Visual Studio's Designer View provided the blank slate required to develop the end application's graphical user interface. The name of the project was saved as ZigBee_Transceiver, and by default, the Visual C# Studio labelled the project's form as Form1. In order to populate the project's form, Visual Studio's toolbox was opened and controls clicked and dragged onto it. The main object associated with interfacing the end-user with the host computer's serial ports and transceiver hardware was serialPort1. Dragged onto the form, the object allows the application to access its SerialPort class through the System.IO.Ports namespace. StatusStrip object statusStrip1 was created by dragging the control onto the form.

Next the MenuStrip object menuStrip1 was created by dragging it onto the form, populating it with standard items, and editing it so that only the *File>Exit* item was available. A click event was programmed for the menuStrip1's item by double clicking the object.

The TabControl object tabControl1 was created by dragging the control onto the form and docking it in the form's parent container. Tab-pages tabPage1 and tabPage2 were added to the tabControl1 object's collection and the default text size and font specified to 10pt, Microsoft Sans Serif. tabPage1's text property was set to: Data, and tabPage2's text property set to: Properties.

The Button control's objects: buttonStart, buttonStop, buttonWrite, buttonClear, and buttonExit, were added to Form1 and their respective text

and name properties set to: Start, buttonStart; Stop, buttonStop; Write, buttonWrite; Clear, buttonClear; Exit, buttonExit. Click events were programmed for these objects by double clicking on each of the them.

RichTextBox and TextBox objects richTextBoxReceivedData and textBoxTransmit were created by dragging the controls onto Form1 and setting their respective name properties to richTextBoxReceivedData and textBoxTransmit.

Once the *"Data"* tab-page was populated with controls, the *"Properties"* tab-page was selected and controls added to it.

ComboBox objects: comboBoxCOMPorts, comboBoxParity, comboBoxBaudRate, comboBoxStopBits, comboBoxFlowControl, and comboBoxDataBits, were dragged onto the form and their respective name properties set to: comboBoxCOMPorts, comboBoxParity, comboBoxBaudRate, comboBoxStopBits, comboBoxFlowControl, and comboBoxDataBits.

Figure 31 and Figure 32 show the designed end application's serial data communications and properties windows.

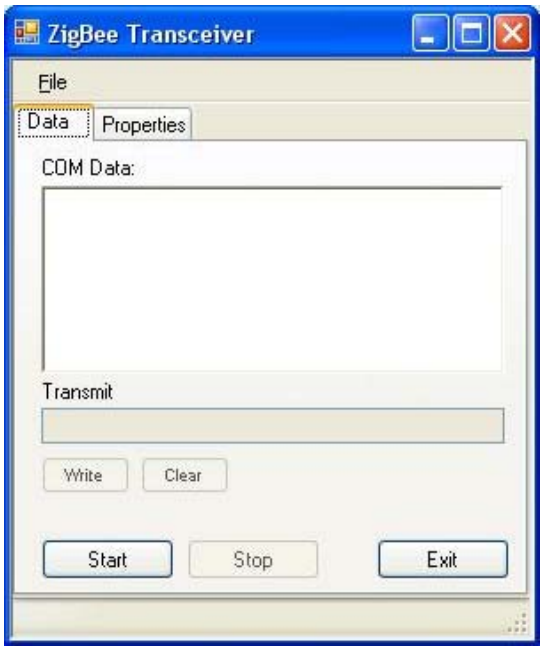

Figure 31: Graphical user interface's communication window

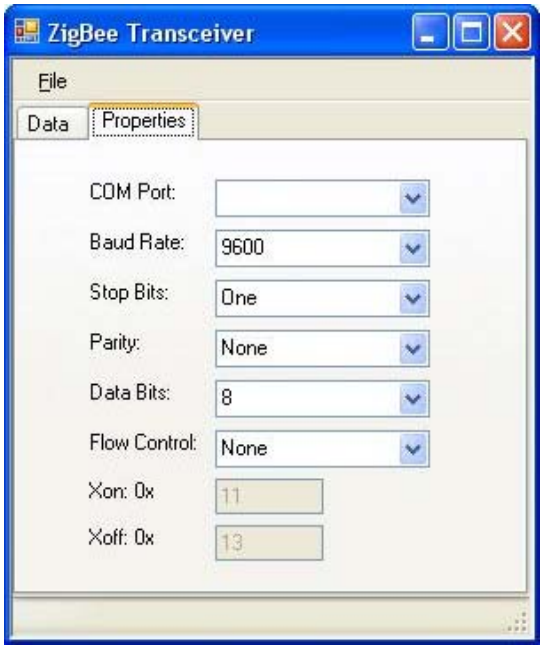

Figure 32: Graphical user interface's settings window

4.6.3 Graphical User Interface Application Source Code

During programming of the application, the Visual C# Studio generated a large amount of source code, providing the code for monotonous operations

such as creating a window and setting its default size and origin's coordinates.

Using the partial keyword, the Visual Studio split the end application's program and Form1 class's code across three source files:

- Program.cs
- Form1.cs
- Form1.Designer.cs

The Form1.cs and Form1.Designer.cs source files contained much of the code related to providing the interface between end-user and the transceiver's hardware and host computer's serial ports.

This first section of code relates to the declaration of Form1.cs' namespaces and Form1 initialisation.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
using System.Threading;
namespace ZigBee_Transceiver
{
        public partial class Form1 : Form
        {
                public Form1()
                {
                        InitializeComponent();
                }
        }
}
```

When repeatedly calling a class from the same namespace, the using directive has been used to shorten statement lengths. For example, using

Systems.IO.Ports; has been used to instruct the compiler to use the System.IO.Ports namespace when encountering the SerialPort class.

When generating the application's source files, the Visual Studio has created the class Form1 within the ZigBee_Transceiver namespace.

When the Form1 class is created its constructor calls Form1.Designer.cs's InitializeComponent method, which is responsible for generating the designed Windows Form's visual layout and setting the properties of the form's controls.

```
private void Form1_Load(object sender, EventArgs e)
{
        comboBoxBaudRate.Enabled = true;
        comboBoxCOMPorts.Enabled = true;
        comboBoxDataBits.Enabled = true;
        comboBoxFlowControl.Enabled = true;
        comboBoxParity.Enabled = true;
        comboBoxStopBits.Enabled = true;
        textBoxXoffChar.Enabled = true;
        textBoxXonChar.Enabled = true;
        textBoxXoffChar.Enabled = false;
        textBoxXonChar.Enabled = false;
        buttonExit.Enabled = true;
        buttonStart.Enabled = true;
        buttonStop.Enabled = false;
        buttonWrite.Enabled = false;
        buttonClear.Enabled = false;
        textBoxTransmit.ReadOnly = true;
        richTextBoxReceivedData.ReadOnly = true;
        comboBoxCOMPorts.Items.AddRange(SerialPort.GetPortNames());
        comboBoxParity.Items.AddRange(Enum.GetNames(typeof(Parity)));
        comboBoxParity.Text = "None";
        comboBoxStopBits.Items.AddRange(Enum.GetNames(typeof(StopBits)));
        comboBoxStopBits.Text = "One";
        string[] stringPacketsize = new string[] { "5", "6", "7", "8" };
        comboBoxDataBits.Items.AddRange(stringPacketsize);
        comboBoxDataBits.Text = "8";
        string[] stringBaudrates = new string[] { "300", "600", "1200", "1800", "2400",
        "3600", "4800", "7200", "9600", "14400", "19200", "28800", "38400", "57600" };
        comboBoxBaudRate.Items.AddRange(stringBaudrates);
        comboBoxBaudRate.Text = "9600";
        string[] stringFlowControl = new string[] {"None", "RTS-CTS", "Xon-Xoff"};
        comboBoxFlowControl.Items.AddRange(stringFlowControl);
        comboBoxFlowControl.Text = "None";
        textBoxXoffChar.Text = "13";
        textBoxXonChar.Text = "11";
        serialPort1.DataReceived += new
        SerialDataReceivedEventHandler(serialPort1_DataReceived);
}
```

A Load event is generated when Form1 loads and the Form1_Load method called. The method initialises Form1's controls' states by setting the combo-boxes', buttons',and text-boxes' enabled property as either true or false.
The ComboBox class's addrange method is used to add items to the objects' drop-down lists.
By calling the SerialPort resource's GetPortNames method, combo-box comboBoxCOMPorts's drop-down list is loaded with a string array of the host computer's available COM ports.
By enumerating the names of the types of Parity, combo-box comboBoxParity contained the list of available parities. The combo-box's default text property was set as “None”.
Achieved in the same way as the comboBoxParity's content, comboBoxStopBits contained the enumerated range of the stop-bit values. The combo-box's default text property was set to “One”.
The contents of string array stringPacketsize were added to the comboBoxDataBits's drop down list, and the combo-box's default text property set to “8”.
The contents of string array stringBaudrates were added to the comboBoxBaudRate's drop down list, and the combo-box's default text property set to “9600”.

Lastly, a new event handler for the serialPort1 object's DataReceived event was defined.

```
private void buttonWrite_Click(object sender, EventArgs e)
{
        if (!serialPort1.IsOpen)
        {
                return;
        }
        char[] buff = new char[1];
        for (int i = 0; i < textBoxTransmit.Text.Length; i++)
        {
                buff[0] = textBoxTransmit.Text[i];
                serialPort1.Write(buff, 0, 1);
                }
}
```

When clicked, button buttonWrite generates a Click event which calls the buttonWrite_Click method.

If the serialPort1 object's IsOpen property is not true, that is IsOpen returns a false boolean value, the application breaks from the method; otherwise, a 1 byte character buffer, buff, is declared.

For the length of the textBoxTransmit's text property's contents, a loop writes the character at index i into buff's index [0], which is then written to the serial port's transmit buffer and synchronously transmitted.

```
private void buttonClear_Click(object sender, EventArgs e)
{
    textBoxTransmit.Clear();
}
```

When clicked, button buttonClear generates a Click event which calls the buttonClear_Click method and clears the textBoxTransmit's text property's contents.

```
    private void buttonStart_Click(object sender, EventArgs e)
    {
            if (this.comboBoxCOMPorts.Text.CompareTo(System.String.Empty) != 0)
            {
                    serialPort1.PortName = this.comboBoxCOMPorts.Text;
                    serialPort1.BaudRate = Convert.ToInt32(comboBoxBaudRate.Text);
                    serialPort1.Parity = (Parity)Enum.Parse(typeof(Parity),
    comboBoxParity.Text);
                    serialPort1.StopBits = (StopBits)Enum.Parse(typeof(StopBits),
    comboBoxStopBits.Text);
                    serialPort1.DataBits = Convert.ToInt32(comboBoxDataBits.Text);
                    try
                    {
                            serialPort1.Open();
                    }
                    catch
                    {
                            MessageBox.Show("Failed to open selected port.\nIs it open in
    another application?\n(" + ")");
                    }
                    finally
                    {
                            if (serialPort1.IsOpen)
                            {
                                    serialPort1.RtsEnable = true;
                                    serialPort1.DtrEnable = true;
                                    serialPort1.WriteTimeout = 5000;
                                    buttonStart.Enabled = false;
```

```
                                buttonStop.Enabled = true;
                                buttonWrite.Enabled = true;
                                buttonClear.Enabled = true;
                                textBoxTransmit.ReadOnly = false;
                                comboBoxBaudRate.Enabled = false;
                                comboBoxCOMPorts.Enabled = false;
                                comboBoxDataBits.Enabled = false;
                                comboBoxFlowControl.Enabled = false;
                                comboBoxParity.Enabled = false;
                                comboBoxStopBits.Enabled = false;
                                textBoxXoffChar.Enabled = false;
                                textBoxXonChar.Enabled = false;

                                }
                        }
                }
        }
```

When clicked, button buttonStart generates a Click event which calls the buttonStart_Click method.

The method first performs a string comparison on the combo-box comboBoxCoMPorts's text property's contents to ensure that an appropriate COM port value has been selected from the combo-box's drop-down list.

When a COM port has been selected, text properties of combo-boxes comboBoxParity, comboBoxBaudRate, comboBoxStopBits, and comboBoxDataBits are copied to the serialPort1 object's appropriate Parity, BaudRate, StopBits, and DataBits properties.

A try-catch attempts to open serialPort1 with the object's Open method. If it fails, the MessageBox class shows a “Failed to open the selected port. Is it open in another application?” message; otherwise, serialPort1's IsOpen property returns true and buttonStart is disabled and buttons buttonStop, buttonWrite, and buttonClear enabled. The combo-boxes comboBoxCOMPorts, comboBoxParity, comboBoxBaudRate, comboBoxStopBits, and comboBoxDataBits are disabled once the serial port is opened.

```
private void buttonStop_Click(object sender, EventArgs e)
{
        try
        {
                // Close the serial port
                serialPort1.Close();
```

```
        }
        catch (System.Exception)
        {
                MessageBox.Show("Failed to close port.");
        }
        finally
        {
                textBoxTransmit.Clear();
                buttonStart.Enabled = true;
                buttonStop.Enabled = false;
                buttonWrite.Enabled = false;
                buttonClear.Enabled = false;
                textBoxTransmit.ReadOnly = true;
                comboBoxBaudRate.Enabled = true;
                comboBoxCOMPorts.Enabled = true;
                comboBoxDataBits.Enabled = true;
                comboBoxFlowControl.Enabled = true;
                comboBoxParity.Enabled = true;
                comboBoxStopBits.Enabled = true;

                if (comboBoxFlowControl.Text == "Xon-Xoff")
                {
                        textBoxXonChar.Enabled = true;
                        textBoxXoffChar.Enabled = true;
                } else {
                        textBoxXonChar.Enabled = false;
                        textBoxXoffChar.Enabled = false;
                }
        }
}
```

When clicked, button buttonStop generates a Click event which calls the buttonStop_Click method.

A try-catch attempts to close the serialPort1 object by calling the Close method. If an exception is generated, the MessageBox class shows a “Failed to close port." message; otherwise, buttonStart is enabled, buttons buttonStop, buttonWrite, and buttonClear disabled, and the combo-boxes comboBoxCOMPorts, comboBoxParity, comboBoxBaudRate, comboBoxStopBits, and comboBoxDataBits enabled.

```
private void buttonExit_Click(object sender, EventArgs e)
{
        if (serialPort1.IsOpen)
        {
                try
                {
                        serialPort1.Close();
                }
                catch (System.Exception)
```

```
                {
                        MessageBox.Show("Failed to close selected port.");
                }
        }
        serialPort1.DataReceived -= new
        SerialDataReceivedEventHandler(serialPort1_DataReceived);
        Application.Exit();
}
```

When clicked, button buttonExit generates a Click event which calls the buttonExit_Click method.

If the serialPort1's IsOpen property is true, a try-catch attempts to close the serial port by calling the Close method. If successful, serialPort1's DataReceived event is removed and the application closed; however, if unsuccessful, a “Failed to close selected port.” message is shown.

```
private void exitToolStripMenuItem_Click(object sender, EventArgs e)
{
        if (serialPort1.IsOpen)
        {
                try
                {
                        serialPort1.Close();
                }
                catch (System.Exception)
                {
                        MessageBox.Show("Failed to close selected port.");
                }
        }
        serialPort1.DataReceived -= new
        SerialDataReceivedEventHandler(serialPort1_DataReceived);
        Application.Exit();
}
```

When clicked, the menu item generates a Click event which calls the exitToolStripMenuItem_Click method. The method performs the exact same operation as the buttonExit_Click method.

```
private void comboBoxFlowControl_SelectedIndexChanged(object sender, EventArgs e)
{
        if (comboBoxFlowControl.Text == "Xon-Xoff")
        {
                // Enable our Xon and Xoff text boxes
                textBoxXonChar.Enabled = true;
                textBoxXoffChar.Enabled = true;
        }
        else
        {
```

```
                // Disable our Xon and Xoff text boxes
                textBoxXonChar.Enabled = false;
                textBoxXoffChar.Enabled = false;
            }
        }
```

When the selected index of comboBoxFlowControl is changed, a SelectedIndexChanged event is generated and the comboBoxFlowControl_SelectedIndexChanged method called.

If the comboBoxFlowControl's text property is equal to “Xon-Xoff”, the text-boxes textBoxXonChar and textBoxXoffChar are enabled; else, they are disabled.

```
private void serialPort1_DataReceived(object sender,
System.IO.Ports.SerialDataReceivedEventArgs e)
{
    // This event handler fires each time data is received by the serial port.
    // Read available data from the serial port and display it on the form.
    // This event does not run in the UI thread, so need to
    // use delegate function
    //string strErg = serialPort1.ReadExisting();
    // Need to take the received hex value and convert it so that it may be displayed
    properly
    try
    {
            byte[] buffer = new byte[serialPort1.BytesToRead];
            serialPort1.Read(buffer, 0, 2);
            int count = buffer[0] + buffer[1];
            this.BeginInvoke(new EventHandler(delegate{
                SetTheText(string.Format("Axle Count: {0} at ",count)+DateTime.Now);
            }));
            Application.DoEvents();
    }
    catch (System.Exception)
    {
        MessageBox.Show("Failed to process byte reception.");
    }

}
```

When serialPort1 receives data, a DataReceived event occurs and the serialPort1_DataReceived method is called.

Because the SerialPort class does not create events on the same thread as Form1's, an invoke of a separate event handler is required in order to update the user interface's text-boxes.

A try-catch attempts to declare the byte buffer, buffer, and read two bytes from serialPort1's received buffer. The integer variable count takes the two bytes and adds them together.
The required invoke of a delegate thread passes a string containing the count variable to the SetTheText method.

```
private void SetTheText(string Text)
{
        richTextBoxReceivedData.AppendText(Text + '\n');
}
```

Method SetTheText operates in the Form1's thread and appends the received string Text to user interface's richTextBoxReceivedData text-box.

Chapter 5

System Integration and Testing

5.1 Testing of Individual System Components

5.1.1 Electromagnetic Energy Harvester Testing

Assuming $x(t)$, a real-valued signal, represents the voltage across a resistance R, it produces a current $i(t) = x(t)/R$. The instantaneous power of the signal is $Ri(t)^2 = x(t)^2/R$, and the energy expended during the incremental interval dt is $x(t)^2/Rdt$. In general, it is not known whether $x(t)$ is a voltage or current signal, and in order to normalize power, it can be assumed that R=1Ω. Hence, the instantaneous power associated with signal $x(t)$ is $x(t)^2$. The signal energy over a time interval of length is defined as

$$E_{2L} = \int_{-L}^{+L} | x(t)^2 | dt \qquad 5.1.1.1$$

The total energy in the signal over the range $t \in (-\infty, \infty)$ is defined as:

$$E = \lim_{L\to\infty} \int_{-L}^{+L} | x(t) |^2 \, dt \qquad 5.1.1.2$$

The average power is defined as:

$$P = \lim_{L\to\infty} \left[\frac{1}{2L} \int_{-L}^{L} | x(t) |^2 \, dt \right] \qquad 5.1.1.3$$

For a limit of $0 \leq E \leq \infty$, signal $x(t)$ is said to be a energy signal, and upon inspection have zero power. Alternatively, if the limit of signal is $0 \leq P \leq \infty$ it is defined as a power signal, possessing infinite energy. These equations define the energy and power, respectively, of any arbitrary signal $x(t)$ and can be used to calculate the energy and power produced by the prototyped generator.

To quantify the mechanism's performance at speeds between 10km/h and 50km/h, a test vehicle was driven over the harvesting mechanism's bladder and the voltage generated across a load measured. In order to ensure maximum energy transfer, the load's impedance was matched to the harvesting mechanism's coils' impedance of 100Ω.

Excited by the test vehicle moving at 10km/h, Figure 33 pictures the voltage generated across the load by the prototyped harvesting mechanism.

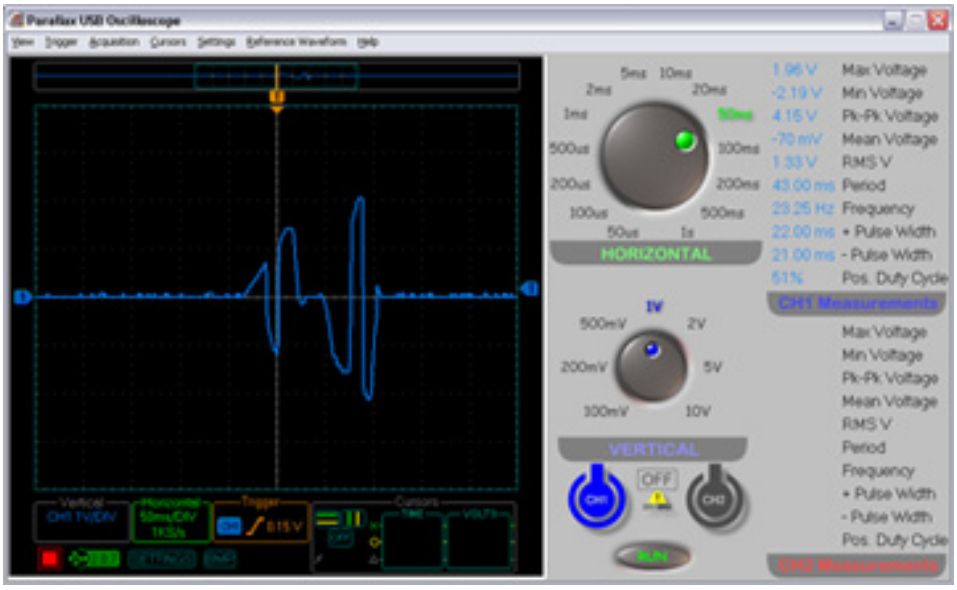

Figure 33: Waveform measured across the load at 10km/h

Split into the displacement and return phases, Figure 33's pictured waveform's first half was generated when the harvesting mechanism's

bladder was driven over, and the magnetic load travelled through the mechanism's coils and coil housing. The second half of the waveform is the result of the bladder's applied force being removed and the hydraulic piston's spring-return pulling the magnetic load back through the coils.

Arranged in a series opposition configuration, the voltages induced in the harvesting mechanism's coils were added constructively as designed.

The first observation made from Figure 33’s pictured waveform, was that the amplitudes of the return phase's induced voltage peaks were generally higher than that of the displacement phase's. It was concluded that damping generated by the hydraulic piston's spring-return during the displacement phase, prevented the magnetic load from being displaced as quickly as the piston's spring-return could pull the magnetic load back; thus, as per Faraday's Law, inducing less voltage in the coils.

The second observation was that the mid-region of Figure 33’s pictured waveform experienced a relatively steady state or pause. This was due to the magnetic load being held at maximum displacement for a short time; thus, the coils did not experience a varying magnetic flux and no voltage was induced.

The harvesting mechanism's induced voltage and generated waveform's change between positive polarity, during the displacement phase, and negative polarity, during the return phase, experimentally validates Lenz's and Faraday's Laws. For the waveform's 150ms period and harvesting device’s 10km/h excitation, it was calculated that approximately 205mJ was transferred to the load.

Excited by the test vehicle moving at 20km/h, Figure 34 pictures the voltage generated across the load by the prototyped harvesting mechanism.

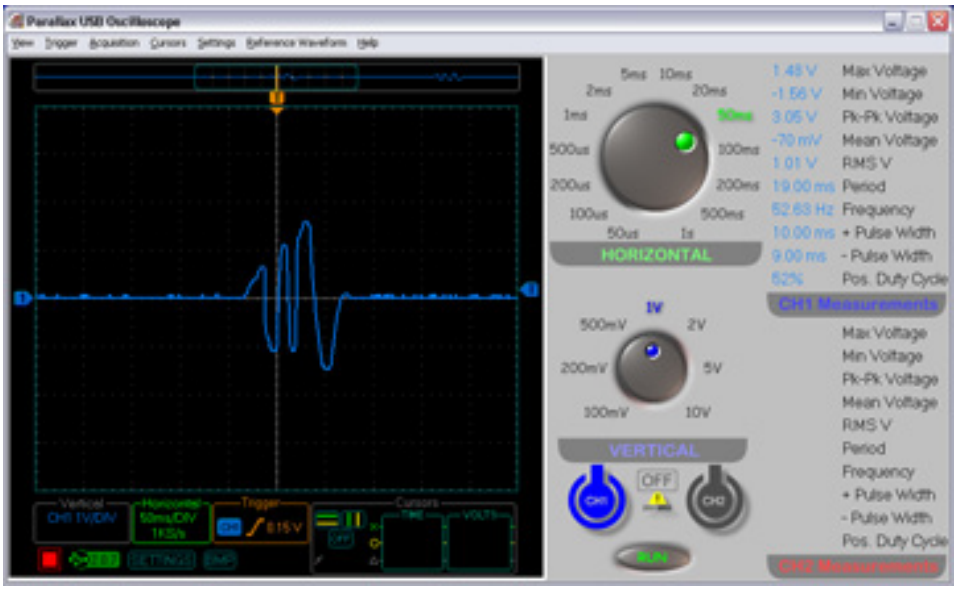

Figure 34: Waveform measured across the load at 20km/h

Unlike the previous figure, the waveform pictured in Figure 34 was considerably shorter in time; approximately 100ms in duration.

The difference between the amplitudes of the displacement phase's induced voltage peaks and the return phase's was noticeable; generally being smaller than the return phase's. Damping, as concluded, significantly affected the amount of voltage induced during the first half of the harvesting mechanism's operation.

Approximately 178mJ of energy was transferred to the load when the harvesting mechanism was excited by a 20km/h velocity.

Excited by the test vehicle moving at 30km/h, Figure 35 pictures the voltage generated across the load by the prototyped harvesting mechanism.

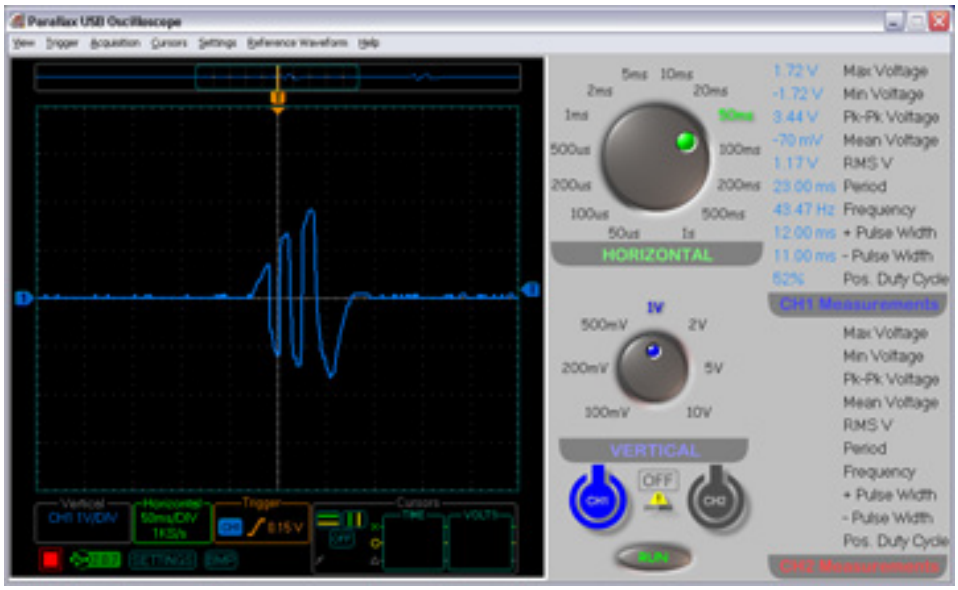

Figure 35: Waveform measured across the load at 30km/h

Similar to the previous figure's pictured waveform's shape and length, Figure 35's lasted for approximately 100ms. It was also observed, as in previous figures, that the amplitudes of the return phase's induced voltage peaks were greater than the displacement phase's.

For an excitation of 30km/h, the energy harvesting device was able to transfer up 136mJ of energy to the load.

Excited by the test vehicle moving at 40km/h, Figure 36 pictures the voltage generated across the load by the prototyped harvesting mechanism.

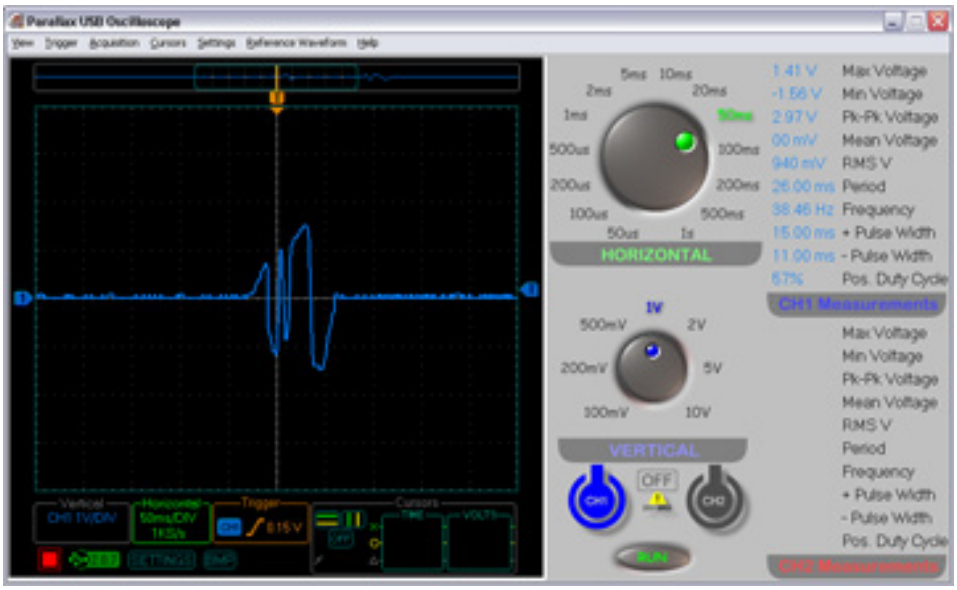

Figure 36: Waveform measured across the load at 40km/h

As pictured in Figure 36, the period of the voltage measured across the load was approximately 100ms in length and experienced a relatively quick transition from positive to negative polarity when changing from the displacement phase to the return phase.

It was concluded the short duration of the last coils' induced voltage was due to the test vehicle's speed, preventing the magnetic load from being fully displaced.

Approximately 100mJ of energy was transferred to the load.

Excited by the test vehicle moving at 50, Fig. 3.05 pictures the voltage generated across the load by the prototyped harvesting mechanism.

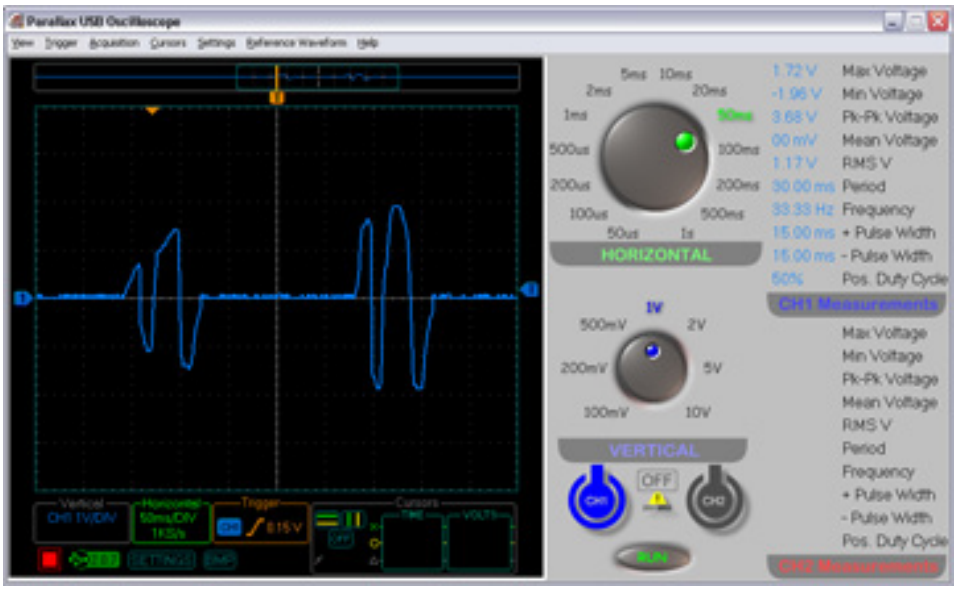

Figure 37: Waveform measured across the load at 50km/h

In contrast to the previous figures, Figure 37's pictured waveform showed that the harvesting mechanism's magnetic load did not pass through the prototyped coil housing's third coil. The waveform pictured showed only two coils experiencing an induced voltage. It was concluded that the speed of the vehicle, much like the conclusion drawn from the previous experiment, prevented the magnetic load from being fully displaced.

As such, when the prototyped system was excited by a 50km/h velocity approximately only 180mJ of energy was able to be transferred to the load.

In order to interface the energy harvesting mechanism's output with sensor circuitry and produce a steady DC voltage, a general purpose diode rectifier bridge was designed and fabricated. As before, a load equivalent to the mechanism's coils' 100Ω impedance was placed between the rectifier's output and ground in order to maximise power transferred.

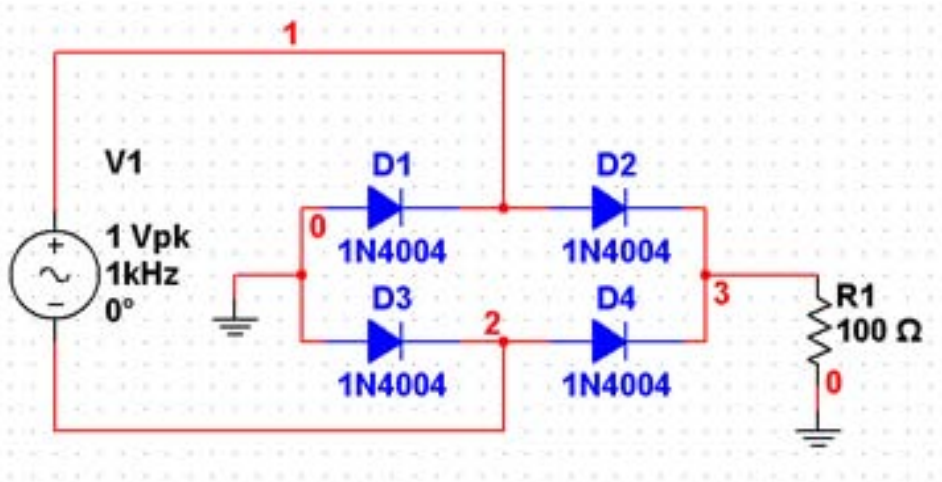

Figure 38: General purpose diode rectifier circuit schematic

To test its performance, and effect on the generated waveform, the rectifier was connected to the fabricated energy harvesting device, which was driven over at 20km/h and 30km/h. The generated rectified waveforms, presented in Figure 38, were compared to that generated across the load from the previous test stage.

For the rectified outputs generated at 10km/h and 30km/h, approximately 100mJ and 230mJ was generated. Compared to the previous load terminated test cases, rectified waveforms seem to transfer more energy to the load at higher speeds, and were able to achieve considerably higher amplitudes of induced voltage.

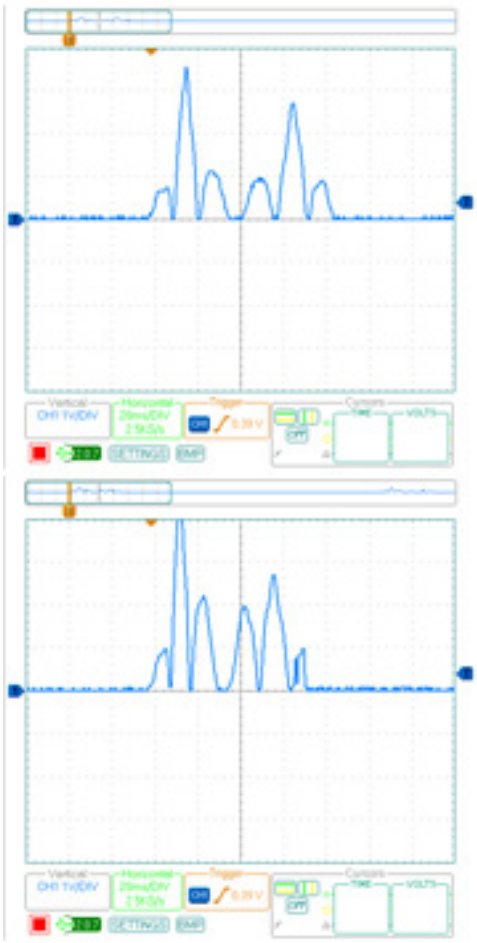

Figure 39: Rectified waveforms generated at 10km/h and 30km/h

From observing the combined energy harvesting device's and general purpose diode rectifier's generated waveforms it was obvious a capacitor was needed to smooth the output.

The general purpose diode rectifier was modified so that a capacitor was connected in parallel with the 100Ω load. To test the performance of the modified rectifier, the combined energy harvesting device and modified general purpose rectifier were tested in the same method as previous experiments.

1µF, 10µF, 100µF, 470µF, and 1000µF capacitors were initially experimented with and the effects they had on the load's waveform noted. In the end a 1000µF capacitor was used. Figure 40 shows the waveform measured across the combined diode rectifier, capacitor and load when driven over at 30km/h

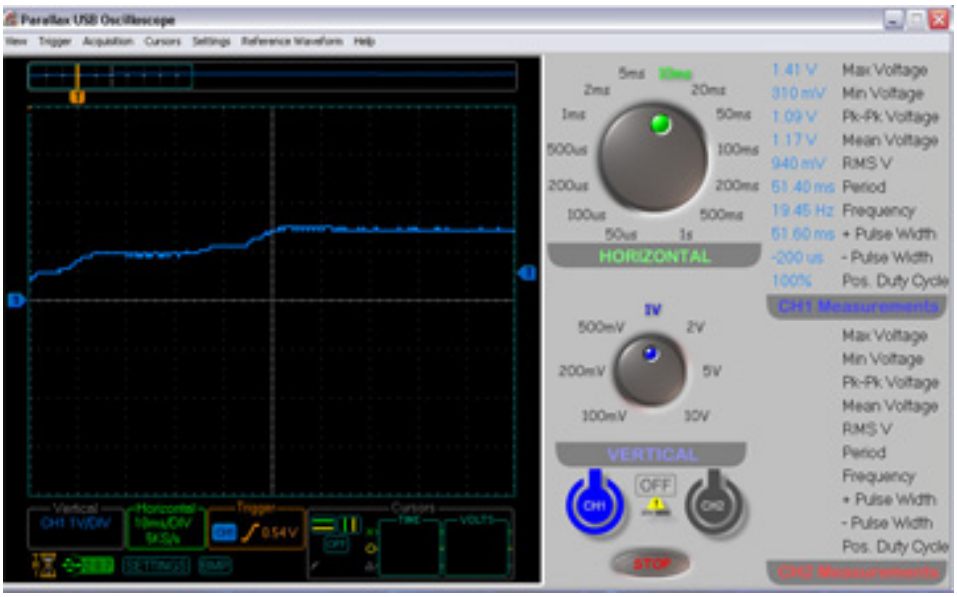

Figure 40: Rectified and smoothed waveform measured across the load

For a series of test runs over the harvesting mechanism, it was observed that the peak voltage generated, once rectified and smoothed, reached a maximum voltage of 1.5V.

5.1.2 Embedded Microcontroller's Application Testing

Having developed the embedded application, Microchip's MPLAB SIM was set as the application's debugger via the Debugger>Select Tool>MPLAB SIM menu, and used to emulate the embedded PIC16F688 microcontroller. Once the application's source files were assembled, linked, and compiled, the application's instructions were stepped through and the MPLAB IDE's Watch window used to monitor the PIC's SFRs.

The key purposes for the application's testing were to ensure: the PIC's EEPROM data memory was able to be read and written to, and the PIC's EUSART module was properly configured. To show characters "transmitted" by the MPLAB SIM debugger, the Sim1 Uart1 window was enabled via the Debugger>Settings menu. The Watch register was used to show the contents of the EEDAT, EEADDR, EECON1, and WREG SFRs, and EE_Data, EE_Data, EE_Count_L, and EE_Count_H memory addresses.

During debugging, breakpoints were added to the application just before and immediately after the first and last instructions of the EE_Read method; therefore, the Watch window would show the contents of the EEPROM's addressing and data SFR's and application's variables before and after calling the method. Making the application and using the MPLAB SIM “Run” instruction caused the application to break at the planned breakpoint. The Watch window, as pictured in Figure 41, shows the contents of the EE_DAT, EE_ADDR, EECON1, and WREG SFRs and EE_Data, EE_Count_L, and EE_Count_H variables.

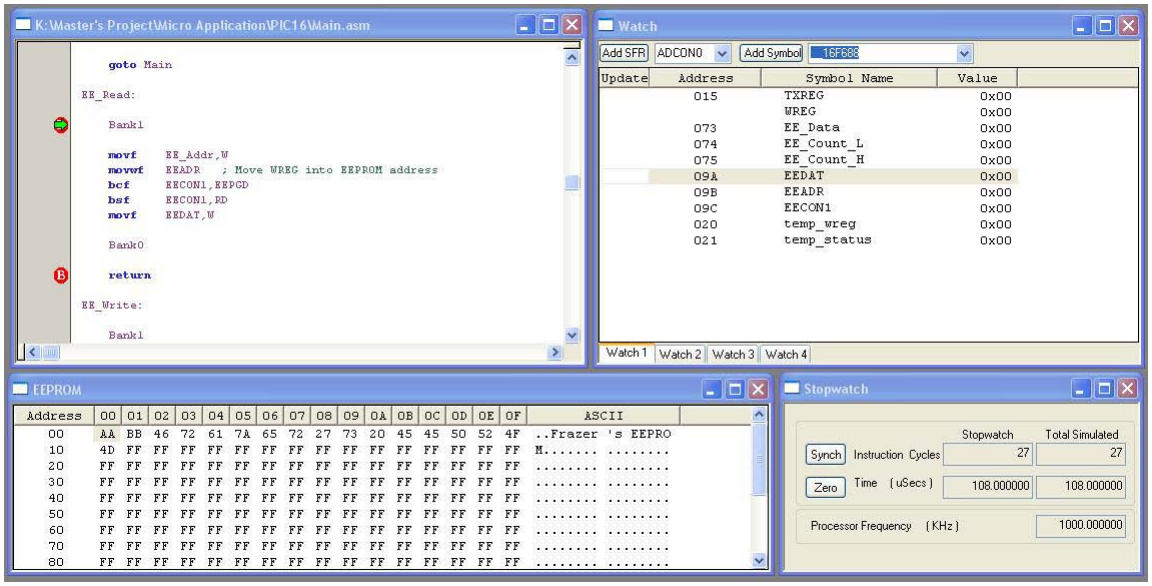

Figure 41: EEPROM SFRs' and variables' contents before EE_Read method execution

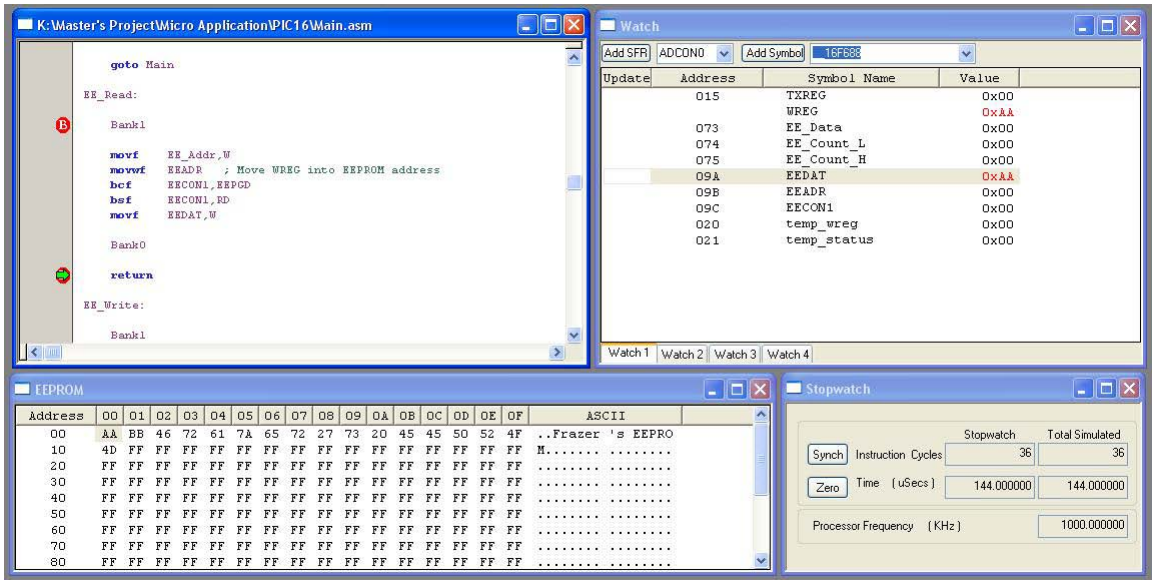

Figure 42: EEPROMM SFRs' and variables' contents after EE_Read method execution

Clicking the debugger's “Run” button a second time caused the application to break just before the EE_Read method branches back to the main application thread. Figure 42 shows the SFRs’ and variables’ contents; clearly the application is able to read the contents of the EEPROM data memory, since the WREG and EEDAT registers are loaded with the contents of the first EEPROM memory address' contents: 0xAA.

The same process was followed for testing the application's ability to write to the PIC's data memory: breakpoints were added just before and immediately after the EE_Write method. Figure 43 shows the contents of the EEPROM’s SFRs and PIC16F688’s EEPROM data memory. Running the simulator caused the application to break at the first breakpoint; where variables EE_Data and EE_Addr hold the address of the data memory where the contents of EE_Data will be written to.

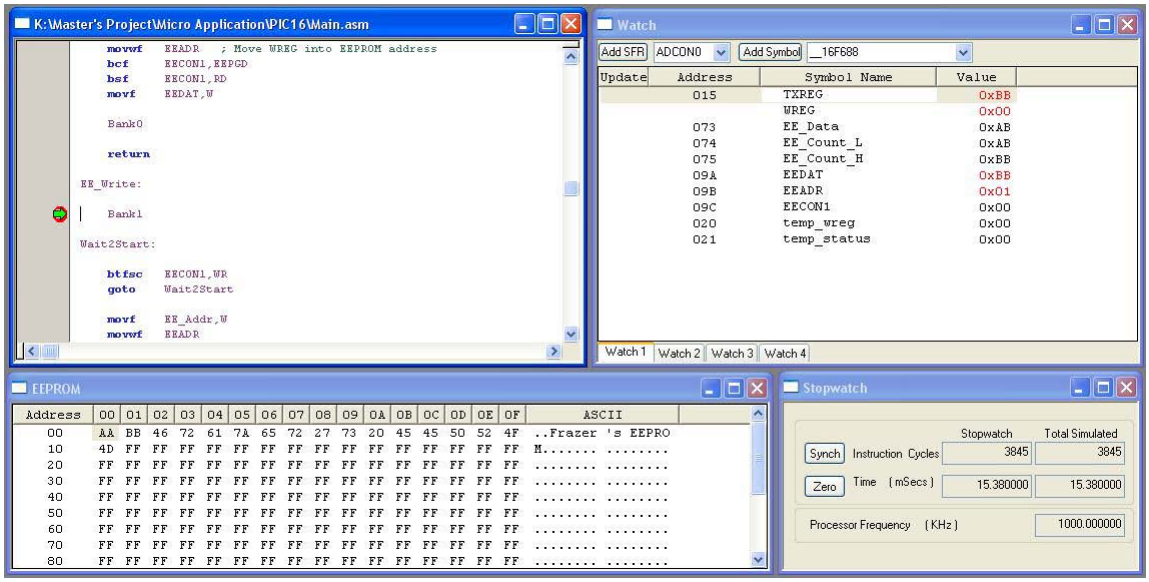

Figure 43: EEPROM SFRs' and variables' contents before EE_Write method execution

Clicking the “Run” button a second time caused the application to break just before the EE_Write method returns to the main thread. Figure 44 shows the contents of the EEPROM SFR and EEPROM data memory after the EE_Write method.

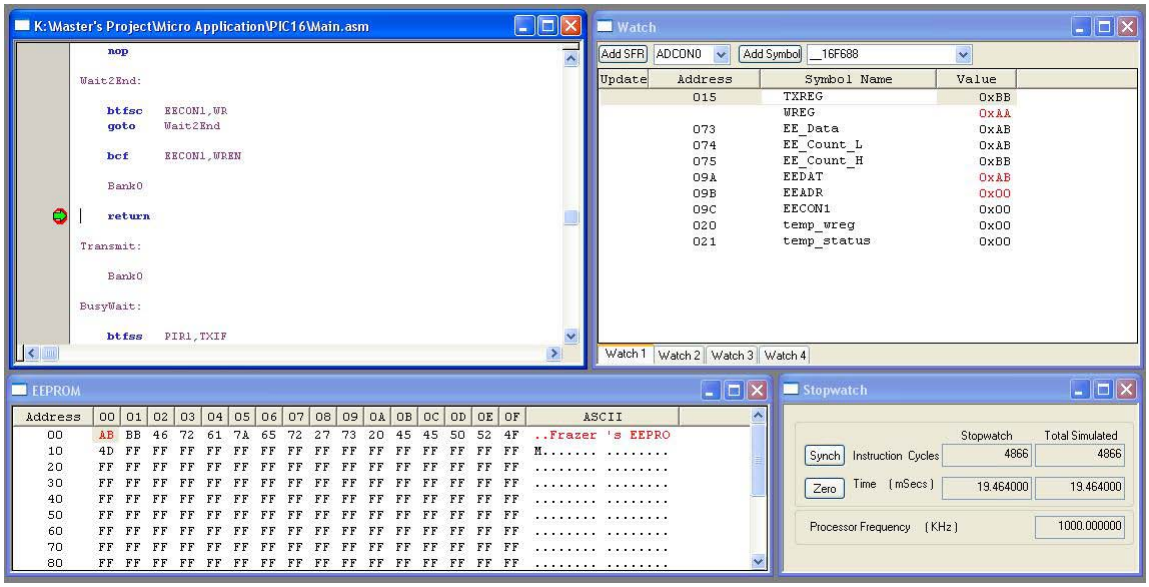

Figure 44: EEPROM SFRs' and variables' contents after EE_Write method execution

Clearly the application is able to write to the EEPROM data memory as its first addresses' contents have been incremented as is intended.

Lastly the EUART's ability to transmit the contents of the EEPROM data memory was tested by placing breakpoints before and after the Transmit Method. If the module was configure correctly the SIM Uart1 window would show an ASCII character based on the value stored in the EEPROM data memory's first two addresses. Clicking the MPLAB SIM's “Run” button caused the application to break just before the Transmit method; Figure 3.13 shows the states of the PIC16F688's SFRs and application variables before the Transmit method.

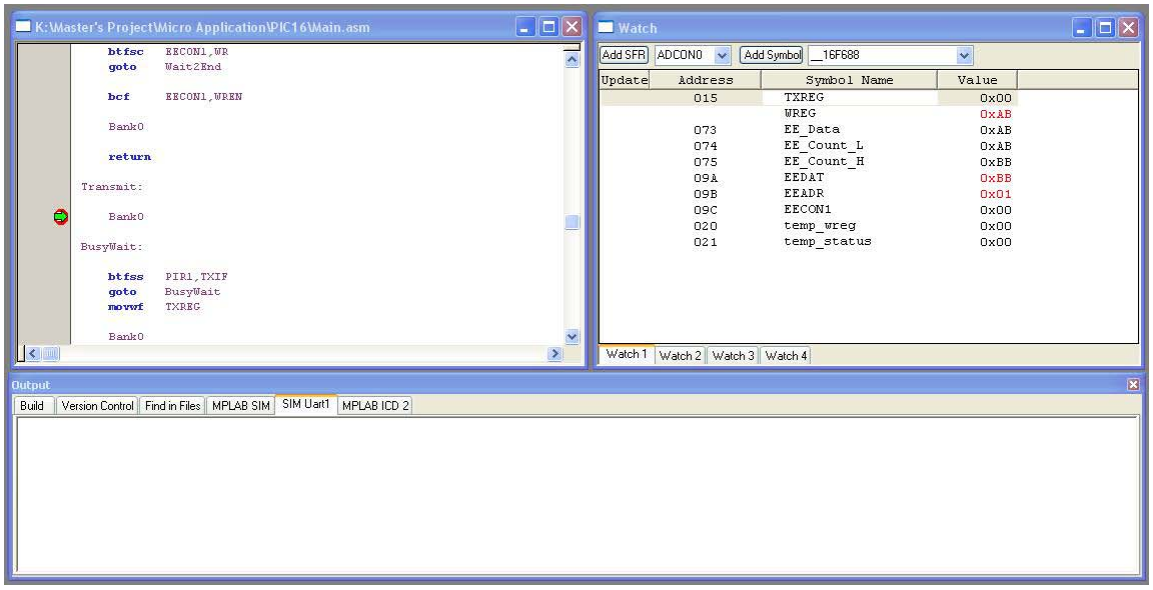

Figure 45: EUART's SFRs' and variables' contents before Transmit method execution

Clicking the “Run” button again caused the application to break just before returning control to the application's main thread. Figure 46 demonstrates the state of the contents of the EUSART’s SFRs and the SIM Uart1 window; clearly the 0xAB hex value has been transmitted as the ASCII character '+'.

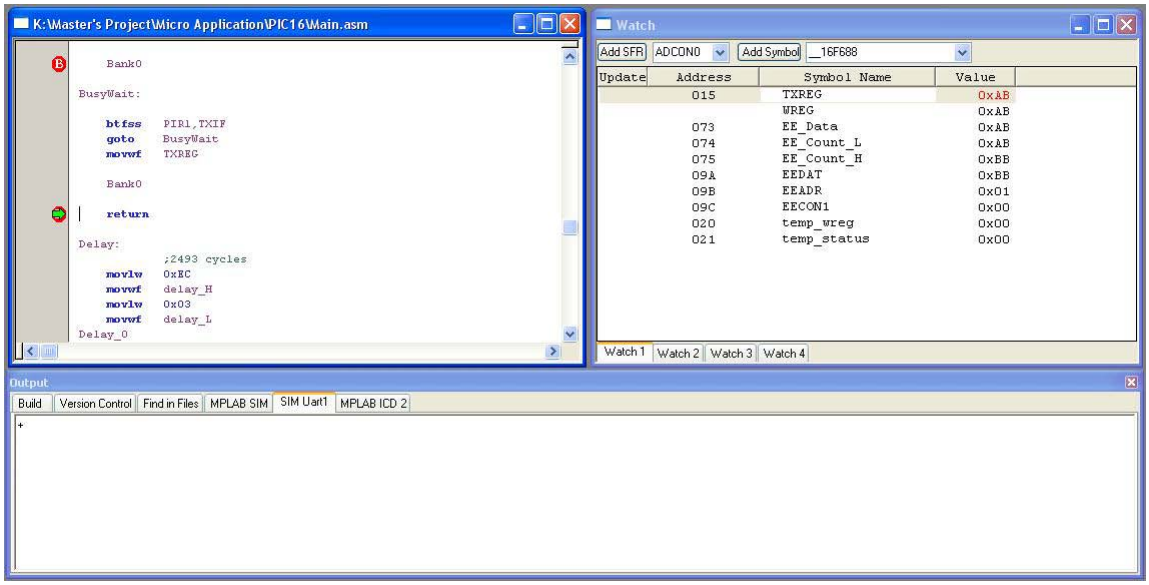

Figure 46: EUART's SFRs' and variables' contents after Transmit method execution

To demonstrate the application's combined ability to read, write to, and transmit the PIC16F688's EEPROM data memory's content's, breakpoints were removed, and the MPLAB SIM's “Reset” and “Run” buttons are pressed 256 times; transmitting the ASCII character equivalent of the memory's contents. Figure 47 shows the SIM Uart1's window for the final test.

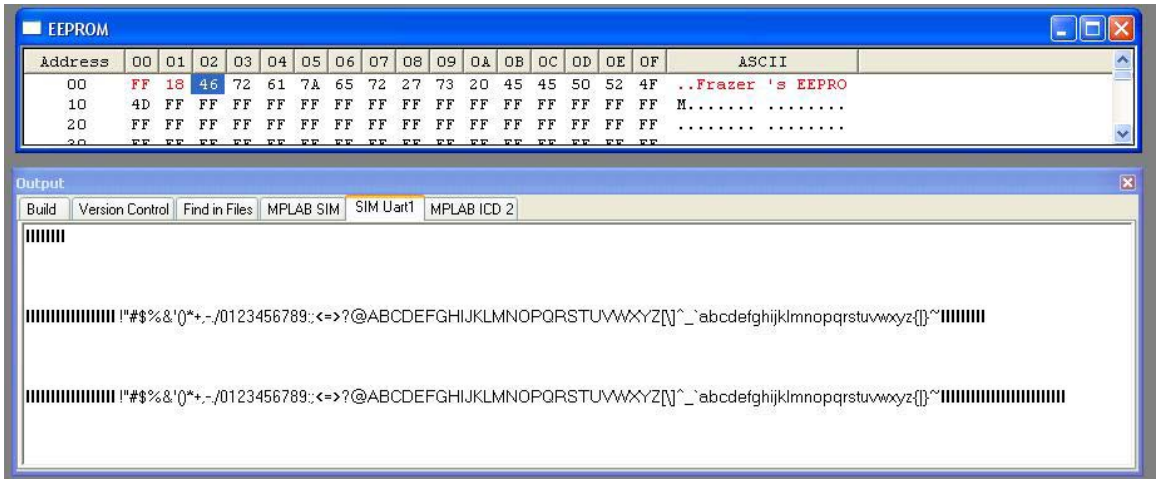

Figure 47: SIM UART1’s output window

The application's ability to read, write to, and transmit the contents of the PIC16F688's EEPROM data memory validated the embedded application's

design; showing that it was able to increment a value upon reset, write it to EEPROM, and transmit it via the PIC16F688's EUSART module. With regards to the energy harvesting and embedded sensor circuit, the working embedded application provided the means to communicate with the vehicle detection sensor's wireless transceiver.

5.1.3 Wireless Transceiver Testing

In order to test the functionality of the prototyped wireless transceiver, a USB XBee development board and the prototyped transceiver were connected to the host PC's USB ports. By configuring one device to read and the other write, two instances of Digi's XBee firmware update software were used to set-up a link between the devices. Initial results showed the prototyped wireless transceiver received 100 percent of transmitted data. The experiment was repeated a second time with the transceiver transmitting. As per the first attempt, the development board received 100 percent of the transmitted data. The results clearly showed the prototyped transceiver worked and as such would be suitable to be used as part of the overall system.

5.1.4 Graphical User Interface Application Testing

Testing of the end application consisted of: ensuring it was able to write to and read the serial port correctly, as well as be interfaced with the wireless transceiver's hardware.

Initially, two of XBee's ZigBee development boards were used to implement a wireless NULL modem and two instances of the end application used to communicate between them; each instant of the end application used a different COM port, so as not to conflict with the other.

For the purpose of testing the application, the serialPort1_DataReceived method was modified to:

```
private void serialPort1_DataReceived(object sender,
System.IO.Ports.SerialDataReceivedEventArgs e)
{
```

```
        try
        {
                string Str;
                Str = serialPort1.ReadExisting();
                SetTheText(Str);
        }
        catch (System.Exception)
        {
                MessageBox.Show("Failed to process byte reception.");
        }
}
```

Where the serialPort1's *ReadExisting* method read all immediately available bytes in the port's buffer, passing them to the Set*TheText* delegate to be appended to the richTextBoxReceivedData's text property.

Once the instances' respective COM ports were selected, the applications' “Start” buttons were clicked.

The string: “Hello World”, was typed into the first instant's “Transmit” text-box's field and its “Write” button clicked. As expected, the second instant received the “Hello World” string and appended it to the end of its “COM Data” text-box.

For continuities sake, the “Hello World” string was typed into the second instant's “Transmit” text-box's field and its “Write” button clicked. As expected, the first instant received the “Hello World” string and appended it to the end of its “COM Data” text-box.

The end application's instances' “Clear” buttons were pressed and the “Transmit” text-boxes' contents cleared.

Finally the “Stop” buttons were pressed and the instances closed without any errors.

Thus, the end application's ability to write to and read the host computer's serial ports was experimentally proven.

Before connecting the wireless transceiver's hardware to the host computer, FTDI's virtual COM port, VCP, drivers were installed. Without them the host computer would be unable to enumerate the wireless transceiver's hardware and the application would fail.

Once the VCP drivers were installed the transceiver was connected to one of the host computer's USB ports and one of the ZigBee development boards removed.

A similar methodology to before was used to test the transceiver hardware's, host computer's, and end application's interfacing:

Once the end applications' instances' respective COM ports were selected, the applications' “Start” buttons were clicked.

The string: “Hello World”, was typed into the first application instant's “Transmit” text-box's field and its “Write” button clicked. As expected, the second application instant received the “Hello World” string and appended it to the end of its “COM Data” text-box.

Again, for continuities' sake, the “Hello World” string was typed into the second application instant's “Transmit” text-box's field and the “Write” button clicked. As expected, the first application instant received the “Hello World” string and appended it to the end of its “COM Data” text-box.

Thus, the end application's ability to interface with the wireless transceiver's hardware and host computer's serial ports was experimentally proven.

5.2 System Integration and Testing

Having validated the operation of the vehicle presence detection sensor's and end system's energy harvester, embedded circuit and application, wireless transceiver, and end application, the individual components were integrated, and whether they worked together experimentally investigated. For the purpose of testing the combined embedded circuit, wireless transceiver, and end application, the embedded circuit was connected to the MPLAB IDE via the ICD2 debugger and provided with a supply voltage equivalent to the harvesting device's output of 1.5V. The embedded circuit's PIC16F688 was reset using the IDE's Release

from Reset and Hold in Reset instruction. Having selected the wireless transceiver's corresponding COM port, the resets were transmitted, received by the wireless transceiver, and related to the end user via the end application in terms of axle count and time received.

Once the embedded circuit was connected to the energy harvesting device, the harvester's bladder was driven over at speeds between 10 km/h and 50 km/h, at incremental 10 km/h steps. Initially, the circuit's soft start component's hysteresis' high threshold was too high, preventing the LTC3401 DC/DC converter from providing power to the PIC16F688 and XBee module; as such, the values of the voltage divider's resistors were changed so as to set the threshold to approximately 1.35V, allowing the circuit to turn on just before reaching the harvesting mechanism's maximum output voltage. It is though that with the load connected the voltage generated under loading was less than expected; however, for the harvesting device's generated output, the DC/DC converter was still able produce a stable 3.3V output, and power the embedded PIC16F688 and XBee ZigBee adapter's transmission.

Figure 48 demonstrates the combined system's functionality, demonstrating the graphical user interface's output for a 30km/h test run.

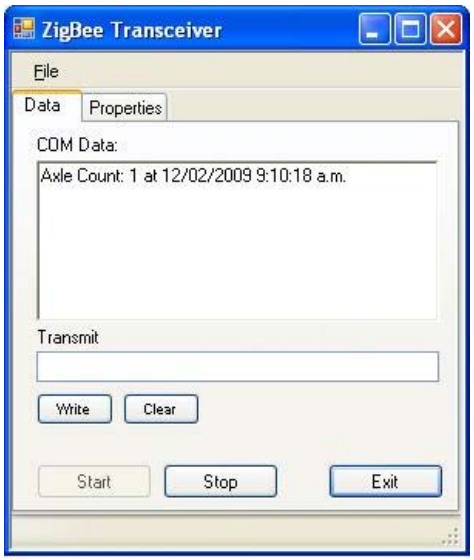

Figure 48: Graphical user interface demonstrating reception of a vehicle detect event

Chapter 6

Conclusion

A result of modern societies' and economies' organisation, traffic congestion significantly impacts nations' economic efficiency, increases pollution, causes the misallocation of scant resources, and wastes time. Congestion cannot be solved by building more roads, tolling, or increasing public transport; irrespective of how championed they are by lobby groups. Congestion is in fact the only solution to populated regions' “excess demand” for limited road resources, and in order to minimise congestion's impacts efficient traffic management systems like Auckland's ATTOMS traffic management system's on-ramp embedded sensors and traffic lights are required. The need for real-time traffic information has provided motivation for this research, spurring the development of a vehicle presence detection sensor and end system in order to implement a distributable sensor traffic management system.

A “self-powered” vehicle presence detection sensor, harvesting energy from a vehicle as it drives over its energy harvesting device's rubber bladder, has been developed and the ability to wirelessly transmit the presence of a vehicle via a 802.15.4 wireless transceiver and present the event via a graphical user interface demonstrated.

The vehicle presence detection sensor's electromagnetic energy harvesting device was demonstrated to be capable of generating between 100mJ and 205mJ per actuation caused by a vehicle axle between speeds of 10 km/h and 50km/h. The sensor's embedded circuit consisted of Microchip's PIC16F688 microcontroller, Linear Technologies' LTC3401 DC-DC converter, Maxim's 9064 Low-power comparator, Digi's XBee ZigBee module, and a full-bridge rectifier. For every actuation caused by a vehicle's axle the energy harvester was able to provide enough energy to cause the 9064 comparator to enable the LTC3401 DC-DC converter, which in turn provided a 3.3 V supply voltage to the PIC16F688 and XBee module. When powered up the PIC16F688 read an

EEPROM address, incrementing the value and writing it to its EUSART module. The ESUART module provided the data packet for the XBee module to transmit.

The end system received the transmitted data packet, writing it to the wireless transceiver's receive buffer, which in turn was converted to USB by FTDI's FT232 USB to UART converter. The C# end application read a virtual COM port and placed the data packet on the screen in terms of time received and vehicle count.

The ability to for a distributable system to wireless transmit the detection of a vehicle's presence in real-time not only means that traffic management systems, like ATTOMS, have the ability to monitor any part of a arterial or suburban roads, allowing for systems to have an improved data resolution, but areas which do not have a traffic management system can easily be implemented due to the vehicle presence detection sensor's distributable nature. Two sensors a fixed distance apart can be used to determine vehicle speeds and traffic flow using a time of flight approach.

Based on the experience gained from development of the vehicle presence detection sensor and end system, there are a number of improvements that could be made in future work: replacement of the sensor's ZigBee wireless module with a lower power consuming device should decrease the embedded sensor's power requirements; mathematical modelling of the energy harvesting module could be performed in order to develop an optimised harvesting device; electrostatic and piezoelectric energy harvesting ability to power a wireless sensor node could be investigated; whether or not an embedded energy harvesting vehicle presence detection can be developed could be investigated; force analysis, regarding the bladder down force/spring coefficient, could be performed; a smaller piston head are could be used in order to maximise piston stroke and thus the number of coils a magnet could be pushed through. Future research could also investigate ways of determining vehicle class types from axle count based on the time between vehicle presence detection events and vibration optimisation of coupled electromagnetic and spring-mass systems.

References

Active materials Laboratory. (2008). Piezoelectric and material theory. Retrieved April 18, 2008 from http://aml.seas.ucla.edu/home.htm.

Virginia Tech Digital Library and Archives. (2008). Piezoelectric theory Chapter Two. Retrieved April 20, 2008 from scholar.lib.vt.edu/theses/available/etd-042299-150134/unrestricted/Chapter2.pdf.

Amirtharajah, R., & Chandrakasan, A. P. (1998). Self-powered signal processing using vibration-based power generation. *IEEE Journal of Solid-State Circuits, 33*(5), 687 - 695. DOI: 10.1109/4.668982.

AVX. (2009). AVX BestCap Ultra-low ESR High Power Pulse Supercapacitors. Retrieved April 17, 2009 from http://www.avx.com

Beeby, S. P., Tudor, M. J., & White, N. M. (2006). Energy harvesting vibration sources for microsystems applications. *Measurement Science and Technology, 17*(12), 175 - 195.

Bishop, R. H. (Ed.). (2002). *The Mechatronic Handbook*. USA: CRC Press Ltd.

Cao, X., & Lee, Y. (2006). Design and Fabrication of Mini Vibration Power Generation System for Micro Sensor Networks. *2006 IEEE International Conference on Information Acquisition, 20 - 23 August 2006*, Page(s): 91 - 95. DOI: 10.1109/ICIA.2006.305868.

Cao, X., Chiang, W., King, Y., & Lee, Y. (2007). Electromagnetic Energy Harvesting Circuit with Feedforward and Feedback DC-DC PWM Boost Converter for Vibration Power Generator System. *IEEE Transactions on Power Electronics, 22*(2), 679 – 685. DOI: 10.1109/TPEL.2006.890009.

Diamond Consulting Services. (2007, 06 Dec). Automatic Vehicle Detection and Classification and Automatic Vehicle Identification Technology. White Paper.

Downs, A. (2004). *Still Stuck in Traffic: Coping with Peak-hour Traffic Congestion*. Washington, D.C.: Brookings Institution Press.

Duffy, M., & Carroll, D. (2004). Electromagnetic generators for power harvesting. *In Power Electronics Specialists Conference 2004 and PESC 04, 20 - 25 June 2004*, Page(s): 2075 – 2081, Volume 3.

Electrical and Computer Engineering Department of University of Toronto. (2008). Charge Pumps: An overview. Retrieved April 10, 2008 from www.eecg.utoronto.ca/~kphang/ece1371/chargepumps.pdf.

Farrel, J. (2008). *Mircosoft Visual C# 2005: An Introduction to Object-Oriented Programming*. Canada: Thomson Learning, Inc.

Glynne-Jones, P., Beeby, S.P., & White, N.M. (2001). Towards a piezoelectric vibration-powered microgenerator. *IEEE proceedings of Science, Measurement and Technology, March 2001*, 148(2), Page(s): 68 – 72. DOI 10.1049/ip-smt:20010323.

Glynne-Jones, P., Tudor, M. J., Beeby, S. P., & White, N. M. (2004). An electromagnetic, vibration-powered generator for intelligent sensor systems. *Sensors and Actuators A: Physical*, 110, 344 – 349.

Griffiths, I., & Adams, M. (2003). *Net Windows Forms in a Nutshell*. Beijing; Cambridge: O'Reilly and Associates, Inc.

Halliday, D., Resnick, R., & Walker, J. (2001). *Fundamentals of Physics* (6th ed): John Wiley& Sons, Inc.
Horowitz, P. (1989). *The art of electronics* (2nd ed). Cambridge, England; N.Y, USA: Cambridge University Press, 1989.

Kay, J. H. (1998). *Asphalt Nation: How the Automobile Took over America, and how We Can Take it Back*. University of California Press

Kim, S., Clark, W. W., & Wang, Q. (2005). Piezoelectric Energy Harvesting with a Clamped Circular Plate: Analysis. *Journal of Intelligent Material Systems and Structures, 16*(10), 847 - 854. DOI: 10.1177/1045389X05054044.

Kim, S., Clark, W. W., & Wang, Q. (2005). Piezoelectric Energy Harvesting with a Clamped Circular Plate: Experimentation. *Journal of Intelligent Material Systems and Structures, 16*(10), 855 - 863. DOI: 10.1177/1045389X05054043.

Meninger, S., Mur-Miranda, J. O., Amirtharajah, R., Chandrakasan, A., & Lang, J. H. (2001). Vibration-to-Electric Energy Conversion*. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 9*(1), 64 - 76. DOI: 10.1109/92.920820.

Kulkarni, S., Koukharenko, E., Tudor, J., Beeby, S., O'Donnell, T., & Roy, S. (2007). Fabrication and Test of Integrated Micro-scale Vibration Based Electromagnetic Generator. *Solid-State Sensors, Actuators and Microsystems Conference 2007, 10 - 14 June 2007*, Page(s):879 - 882. DOI: 10.1109/SENSOR.2007.4300271.

LaFollette, R., Singh, P., Broadhead, J., & Reisner, D. (2002). Development of a Fuzzy-Logic Managed Microscopic Battery. *Proceedings of IEEE Sensors 2002, 12 – 14 June 2002*, Page(s): 1193 – 1198, Volume 2. DOI: 10.1109/ICSENS.2002.1037284.

Maxim. (2008). Ultra-Small, Low-Power Single Comparators in 4-Bump and 5-SOT23. Retrieved Nov 1, 2008 from http://www.maximic.com

Microchip. (2009). nanoWatt Technology Overview. Retrieved April 16, 2009 from http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2628&param=en533162

Neil, N. H., Ching, H., Wong, Y., Wen, J. Philip, H., Leong, W., & Wen, Z. (2002). A laser-micromachined multi-modal resonating power transducer for wireless sensing systems. *Sensors and Actuators A: Physical*, 97 - 98, 685 - 690. DOI: 10.1016/S0924-4247(02)00033-X.

Nardis, L. D., & Benedetto, M. D. (2007). Overview of the IEEE 802.15.4/4a standards for low data rate Wireless Personal Data Networks. *4th Workshop on positioning, navigation and communication 2007* (WPNC ’07). Hannover, Germany.

Niyato, D. Hossain, E., Rashid, M. M., & Bhargava, V. K. (2007). Wireless sensor networks with energy harvesting technologies: a game-theoretic approach to optimal energy management. *Wireless Communications, IEEE. 14*(4), 90-96. DOI: 10.1109/MWC.2007.4300988

NZ Transport Agency. (2009). About ATTOMS. Retrieved April 14, 2009 from http://traffic.transit.govt.nz/About.do

NZ Transport Agency. (2009). About ramp signals. Retrieved Apr 14, 2009 from http://www.transit.govt.nz/projects/rampsignalling/about/

Ogata, K. (2002). *Modern Control Engineering*. N.J., USA: Prentice-Hall, Inc.

Paradiso, J. A., & Starner, T. (2005). Energy scavenging for mobile and wireless electronics Energy Scavenging*. Pervasive Computing, IEEE, 4*(1), 18 - 27. DOI: 10.1109/MPRV.2005.9.

Piezo Systems, Inc. (2008). Piezoelectric Terminology. Retrieved April 22, 2008 from http://www.piezo.com/tech1terms.html .

Predko, M. (2005). *123 PIC Microcontroller Experiments for the Evil Genius*. USA:The McGraw-Hill Companies, Inc.

Rashid, M. H. (2004). *Power Electronics: Circuits, Devices, and Applications* (3rd ed). Upper Saddle River, N.J, USA: Prentice-Hall, Inc.

Roundy, S. (2005). On the Effectiveness of Vibration-based Energy Harvesting. *Journal of Intelligent Material Systems and Structures, 16*(10), 809 - 823. DOI: 10.1177/1045389X05054042.

Sanchez, J, & Canton, M. P. (2007). *Microcontroller Programming: the Microchip PIC*. Boca Raton, FL, USA: CRC Press.

Seeman, M. D., Sanders, S. R., & Rabaey, J. M. (2007) An Ultra-Low-Power Power Management IC for Wireless Sensor Nodes. *IEEE Custom Integrated Circuits Conference 2007, CICC `07, 16 – 19 September 2007*, Page(s) 567 – 570. DOI: 10.1109/CICC.2007.4405795.

Sharp, J. (2006). *Microsoft Visual C# 2005: Step by Step*. Redmond, Washington, USA: Microsoft Press.

Sodano, H. A., Inman, D. J., & Park, G. (2005) .Comparison of Piezoelectric Energy Harvesting Devices for Recharging Batteries. *Journal of Intelligent Material Systems and Structures, 16*(10), 799 - 807. DOI: 10.1177/1045389X05056681.

Soliman, S. S., & Srinath, M. D. (1998). *Continuous and Discrete Signals and Systems* (2nd ed). Upper Saddle River, N.J, USA: Prentice-Hall, Inc.

Thompson, M. (2006). *Intuitive Analog Circuit Design: a Problem-solving Approach Using Design Case Studies*. Durlington, MA, USA: Newnes.

Torah, R., Beeby, S. P., Tudor, M. J., O'Donnell, T., & Roy, S. (2006). Development of a Cantilever Beam Generator Employing Vibration Energy Harvesting. *The 6th Int. Workshop on Micro and Nanotechnology for Power Generation and Energy Conversion Applications (PowerMEMS 2006), November 29th - December 1st 2006*, Berkeley, USA. Retrieved from the University of Southampton Web site: http://eprints.ecs.soton.ac.uk/13356/.

Torah, R. N., Tudor, M. J., Patel, K., Garcia, I. N., & Beeby, S. P. (2007). Autonomous Low Power Microsystem Powered by Vibration Energy Harvesting. *2007 IEEE Sensors, 28 - 31 October 2007*, Page(s): 264 – 267. DOI: 10.1109/ICSENS.2007.4388387.

Tsutsumino, T., Suzuki, Y., Kasagi, N., & Sakane, Y. (2006). Seismic Power Generator Using High-Performance Polymer Electret. *19th IEEE International Conference on Micro Electro Mechanical Systems 2006, MEMS 2006 Istanbul, 2006*, Page(s): 98 - 101. DOI 10.1109/MEMSYS.2006.1627745.

Tsutsumino, T., Suzuki, Y., Kasagi, N., & Sakane, Y. (2007). Electromechanical Modelling of Micro Electret Generator for Energy Harvesting. *International Solid-State Sensors, Actuators and Microsystems Conference 2007 and TRANSDUCERS 2007, 10 – 14 June 2007*, Page(s): 863 – 866. DOI: 10.1109/SENSOR.2007.4300267.

Williams, C. B., & Yates, R. B. (1995). Analysis of a Micro-electric Generator for Microsystems. *The 8th International Conference on Solid-State Sensors and Actuators 1995, Eurosensors IX, and Transducers `95, 25 - 29 June 1995*, Page(s): 369 – 372. Retrieved from the IEEE website: http://ieeexplore.ieee.org/.

Wu, J., & Chang, K. (1998). MOS Charge Pumps for Low-Voltage Operation. *IEEE Journal of Solid-State Circuits, 33*(4), 592 - 597. DOI: 10.1109/4.663564.

Yen, B. C., & Lang, J. H. (2006). A variable-Capacitance Vibration-to-Electric Energy Harvester*. IEEE Transactions on Circuits and Systems I: Regular Papers, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 53*(2), 288 – 295. DOI: 10.1109/TCSI.2005.856043.

Zhong-Xian, L., Xiao-Ming, Y., & Zongjin, L. (2006). Application of Cement-Based Piezoelectric Sensors for Monitoring Traffic Flows. *Journal of Transport Engineering, 132*(7), 565 – 573.