FINAL REPORT ~ FHWA-OK-19-03

DEVELOPMENT OF INTELLIGENT VEHICLE COUNTING AND CLASSIFICATION SENSOR (*i*VCCS)

Siraj Muhammad, Ph.D. Student
Hazem H. Refai, Ph.D.

School of Electrical and Computer Engineering (ECE)
Gallogly College of Engineering
The University of Oklahoma
Norman, Oklahoma

October 2019

OKLAHOMA
Transportation

The Oklahoma Department of Transportation (ODOT) ensures that no person or groups of persons shall, on the grounds of race, color, sex, religion, national origin, age, disability, retaliation or genetic information, be excluded from participation in, be denied the benefits of, or be otherwise subjected to discrimination under any and all programs, services, or activities administered by ODOT, its recipients, sub-recipients, and contractors. To request an accommodation please contact the ADA Coordinator at 405-521-4140 or the Oklahoma Relay Service at 1-800-722-0353. If you have any ADA or Title VI questions email ODOT-ada-titlevi@odot.org.

The contents of this report reflect the views of the author(s) who is responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the views of the Oklahoma Department of Transportation or the Federal Highway Administration. This report does not constitute a standard, specification, or regulation. While trade names may be used in this report, it is not intended as an endorsement of any machine, contractor, process, or product.

DEVELOPMENT OF INTELLIGENT VEHICLE COUNTING AND CLASSIFICATION SENSOR (*i*VCCS)

FINAL REPORT ~ FHWA-OK-19-03
ODOT SP&R ITEM NUMBER 2303

Submitted to:
Office of Research and Implementation
Oklahoma Department of Transportation

Submitted by:
Siraj Muhammad, Ph.D. Student
Hazem H. Refai, Ph.D.
School of Electrical and Computer Engineering
The University of Oklahoma – Tulsa

OKLAHOMA
Transportation

October 2019

TECHNICAL REPORT DOCUMENTATION PAGE

1. REPORT NO.	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.
FHWA-OK-19-03		

4. TITLE AND SUBTITLE	5. REPORT DATE
Development of Intelligent Vehicle Counting and Classification Sensor (*i* VCCS)	Oct 2019
	6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S)	8. PERFORMING ORGANIZATION REPORT
Siraj Muhammad Hazem H. Refai, Ph.D.	The University of Oklahoma 202 W. Boyd St., Room 104 Norman, OK. 73019-0631
9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. WORK UNIT NO.
School of Electrical and Computer Engineering, The University of Oklahoma	
	11. CONTRACT OR GRANT NO. ODOT SPR Item Number 2303
12. SPONSORING AGENCY NAME AND ADDRESS	13. TYPE OF REPORT AND PERIOD COVERED
Oklahoma Department of Transportation Office of Research and Implementation 200 N.E. 21st Street, Room G18 Oklahoma City, OK 73105	Final Report Oct 2016 - Sep 2019
	14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES

16. ABSTRACT

The Internet of Things (IoT) is reshaping our world. Soon our lives will be based on smart technologies. According to IHS Markit forecasts, the number of connected devices will grow from 15.4 billion in 2015 to 30.7 billion in 2020. Forrester Research reports that fleet management and transportation sectors lead others in IoT growth. This may come as no surprise, since infrastructure (e.g., roadways, bridges, airports,) is a prime candidate for sensor integration, providing real-time measurements to support intelligent decisions.

This report presents the design, development, and implementation for a novel, autonomous, and intelligent wireless sensor for various traffic surveillance applications. Research activities advanced three generations of the proposed sensor, each adding intelligence, performance, accuracy, and lifetime to its predecessor. The first generation serves a proof-of-concept design/implementation of a novel, fully-autonomous, intelligent wireless sensor for real-time traffic surveillance. Multi-disciplinary, innovative integration of state-of-the-art, ultra-low-power embedded systems; smart physical sensors; and wireless sensor networks, coupled with intelligent algorithms will address component composition of the developed platform, namely Intelligent Vehicle Counting and Classification Sensor (*i*VCCS). Second generation (G) *i*VCCS introduced algorithms for optimizing power consumption and were based on an event-driven methodology wherein a control block orchestrates the work of various components and subsystems powered by a reinforcement learning algorithm. *i*VCCS third G incorporated advanced wireless communication capabilities via Bluetooth Low Energy 5 (BLE 5) with higher data rates, long-range operation, and over-the-air firmware upgrades. The data storage unit has been upgraded with an on-board flash array, and the power management subsystem has been simplified by eliminating energy harvesting. The sensor is now powered by a high capacity (i.e., 10000 mAh), ultra-wide-range operating temperatures and compact size batteries that use Lithium-Thionyl chemistries.

17. KEY WORDS	18. DISTRIBUTION STATEMENT
Intelligent Transportation System (ITS), Traffic Surveillance, Vehicles Detection, WSN, IoT, Low-Power	No restrictions. This publication is available from the Office of Research and Implementation, Oklahoma DOT (ODOT).

19. SECURITY CLASSIF. (OF THIS REPORT)	20. SECURITY CLASSIF. (OF THIS PAGE)	21. NO. OF PAGES	22. PRICE
Unclassified	Unclassified	77	N/A

SI* (MODERN METRIC) CONVERSION FACTORS

APPROXIMATE CONVERSIONS TO SI UNITS

SYMBOL	WHEN YOU KNOW	MULTIPLY BY	TO FIND	SYMBOL
		LENGTH		
in	inches	25.4	millimeters	mm
ft	feet	0.305	meters	m
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
		AREA		
in^2	square inches	645.2	square millimeters	mm^2
ft^2	square feet	0.093	square meters	m^2
yd^2	square yard	0.836	square meters	m^2
ac	acres	0.405	hectares	ha
mi^2	square miles	2.59	square kilometers	km^2
		VOLUME		
fl oz	fluid ounces	29.57	milliliters	mL
gal	gallons	3.785	liters	L
ft^3	cubic feet	0.028	cubic meters	m^3
yd^3	cubic yards	0.765	cubic meters	m^3
		NOTE: volumes greater than 1000 L shall be shown in m^3		
		MASS		
oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short tons (2000 lb)	0.907	megagrams (or "metric ton")	Mg (or "t")
		TEMPERATURE (exact degrees)		
^oF	Fahrenheit	5 (F-32)/9 or (F-32)/1.8	Celsius	^oC
		ILLUMINATION		
fc	foot-candles	10.76	lux	lx
fl	foot-Lamberts	3.426	candela/m^2	cd/m^2
		FORCE and PRESSURE or STRESS		
lbf	poundforce	4.45	newtons	N
lbf/in^2	poundforce per square inch	6.89	kilopascals	kPa

APPROXIMATE CONVERSIONS FROM SI UNITS

SYMBOL	WHEN YOU KNOW	MULTIPLY BY	TO FIND	SYMBOL
		LENGTH		
mm	millimeters	0.039	inches	in
m	meters	3.28	feet	ft
m	meters	1.09	yards	yd
km	kilometers	0.621	miles	mi
		AREA		
mm^2	square millimeters	0.0016	square inches	in^2
m^2	square meters	10.764	square feet	ft^2
m^2	square meters	1.195	square yards	yd^2
ha	hectares	2.47	acres	ac
km^2	square kilometers	0.386	square miles	mi^2
		VOLUME		
mL	milliliters	0.034	fluid ounces	fl oz
L	liters	0.264	gallons	gal
m^3	cubic meters	35.314	cubic feet	ft^3
m^3	cubic meters	1.307	cubic yards	yd^3
		MASS		
g	grams	0.035	ounces	oz
kg	kilograms	2.202	pounds	lb
Mg (or "t")	megagrams (or "metric ton")	1.103	short tons (2000 lb)	T
		TEMPERATURE (exact degrees)		
^oC	Celsius	1.8C+32	Fahrenheit	^oF
		ILLUMINATION		
lx	lux	0.0929	foot-candles	fc
cd/m^2	candela/m^2	0.2919	foot-Lamberts	fl
		FORCE and PRESSURE or STRESS		
N	newtons	0.225	poundforce	lbf
kPa	kilopascals	0.145	poundforce per square inch	lbf/in^2

*SI is the symbol for the International System of Units. Appropriate rounding should be made to comply with Section 4 of ASTM E380. (Revised March 2003)

Acknowledgements

PI Dr. Refai and his research team recognize Oklahoma Department of Transportation (ODOT) for providing funds to support project research activities. Additionally, ODOT personnel are highly acknowledged for support in coordinating highway deployments, testing developed system, and engaging in insightful discussions throughout project execution.

Table of Contents

Table of Figures

List of Tables

Executive Summary

The Internet of Things (IoT) is reshaping our world. Soon our lives will be based on smart technologies. According to IHS Markit forecasts, the number of connected devices will grow from 15.4 billion in 2015 to 30.7 billion in 2020. Forrester Research reports that fleet management and transportation sectors lead others in IoT growth. This comes as no surprise, since infrastructure (e.g., roadways, bridges, airports) is a prime candidate for sensor integration, providing real-time measurements to support intelligent decisions.

This report presents the design, development, and implementation of a novel, autonomous, and intelligent wireless sensor for various traffic surveillance applications. Research activities advanced three generations of the proposed sensor, each adding intelligence, performance, accuracy, and lifetime to its predecessor.

The first generation serves a proof-of-concept design/implementation for a novel, fully-autonomous, intelligent wireless sensor for real-time traffic surveillance. Multi-disciplinary, innovative integration of state-of-the-art, ultra-low-power embedded systems; smart physical sensors; and wireless sensor networks, coupled with intelligent algorithms will address the component composition of the developed platform, namely Intelligent Vehicle Counting and Classification Sensor (*i*VCCS). Second generation (G) *i*VCCS introduced several algorithms for optimizing power consumption and were introduced based on an event-driven methodology wherein a control block orchestrates the work of various components and subsystems. Two operation modes (i.e., HP and LP) were designed for iVCCS 2nd G to balance a tradeoff between consumed energy and response time, which affects detection accuracy relative to traffic flow. A Reinforcement Learning (RL) approach was proposed to design a dynamic power management (DPM) algorithm for observing traffic and controlling the system's power policy according to the environment and agent states. Experimental results showed that overall battery life of the system was extended to over 200 days for a 2300 mAh battery.

*i*VCCS third G incorporated advanced wireless communication capabilities via Bluetooth Low Energy 5 (BLE 5) with higher data rates, long-range operation, and over-the-air firmware upgrades. The data storage unit has been upgraded with an on-board flash array, and the power management subsystem has been simplified by eliminating the energy harvesting. The sensor is now powered by a high capacity (i.e., 10000 mAh), ultra-wide-range operating temperatures and compact size batteries that use Lithium-Thionyl chemistries. The sensor was deployed on several field tests on highways to evaluate its performance. Constant communication with the sensors was achieved throughout the deployments, demonstrating ability for real-time data reporting to access points. The system achieved detection accuracy as high as 98% and on average 88.66%. Reported speed values were compatible with speed limits of their respective locations.

Several enclosure designs were proposed to protect sensors from environmental effects and mechanical forces exerted by passing vehicles. A robust and light-weight enclosure design that meets necessary requirements was designed and manufactured by NCTronics, Inc.

Chapter 1 Introduction

1.1 Introduction

Transportation systems are at the heart of our socio-economic system and play a major role in shaping the trajectory of economic strength and quality of life in a given community. For a strong and continually prosperous economy, it is necessary for the underlying infrastructure to be equipped and up-to-date with technological advancements common in other industry sectors. Hence, it is imperative that transportation systems integrate and are interoperative with others.

Intelligent Transportation System (ITS) technology has accelerated since the turn of the century resulting from a variety of technological forces requiring efficient utilization of the transportation infrastructure. Although other industry sectors (e.g., information technology, communications) have evolved at an even faster pace, the 2010-2011 ITS America Annual Report informs [1]:

- 77% of fixed bus route agencies have real-time arrival data.
- 94% of toll roads have electronic collection.
- 70% of the population is covered by 511 systems in 38 states.
- Thousands of miles of highway and arterial roads are managed under Traffic Management Centers surveillance.

ITS is one of the most promising sectors for excitement over the Internet of Things (IoT), primarily due to the impact transportation system elements will have on domains connected to the Internet cloud. Smart traffic signals and surveillance systems enable more convenient traffic planning by communicating traffic conditions and flow to connected vehicle on-board systems, which, in turn, will aid in saving an enormous amount of time attributed to traffic congestion delays. ITS America reports that each year the average American spends 40 hours sitting in traffic [2]. Moreover, public safety vehicles, buses, and commercial vehicle fleets will benefit from a smart, connected infrastructure that communicates valuable information about traffic and road conditions. Cisco Systems has already introduced their high-level holistic Cisco Connected Roadways system [3].

Vehicle detection and classification constitute essential ITS system components. However, integral real-time traffic monitoring and analysis functionality proves problematic. Commuters, traffic administrators and agencies must rely on such systems for improving traffic control and management, as well as decision-making for trip planning and routing.

Technologies used for vehicle detection and surveillance can be categorized into two types:

1. In-roadway sensors, including Inductive Loop Detectors (ILD), magnetic sensors, piezoelectric sensors, and Weigh-In-Motion (WIM) sensors.
2. Off-roadway sensors, including video image processors, microwave radars, infrared sensors, ultrasonic sensors, and passive acoustic arrays sensors.

Interested readers can find additional details about these technologies in [4]. Existing technologies, like Bluetooth, have also been used to detect vehicles and estimate highway travel time [5].

Regardless of facilitating technologies, the era of IoT is redefining objectives and standards for many systems. IoT devices are expected to be smart, reliable, low-cost, and free from significant maintenance. For an ITS systems to fit into the IoT paradigm, it must adhere to the inherent need for constrained

resources for processing, memory, power, and security functionality. These requirements could impose further difficulties and challenges for ITS, especially given the nature of applications they target. For example, some Automatic Vehicle Classification (AVC) systems cannot save collected data for more than several days. WIM systems, on the other hand, can store data for up to a year. Problems with power consumption persist, whether at the processing or communication stage at the point when data is transmitted to an access point for further processing.

The work detailed in this report is an extension to a project that focused on designing a low-cost, reliable, and low-power intelligent vehicle counter and classification sensor (*i*VCCS) system [6] in which the sensor exploits the physical phenomenon of magnetic field disturbance caused by ferrite materials in the body of a vehicle. The sensor relies on a triaxial magnetometer to measure magnetic field before measurements are transmitted to a processing platform. Previous work (i.e., first generation *i*VCCS [*i*VCCS 1stG]) proved the design concept with high accuracy and reliability [6]–[8]. A second generation *i*VCCS (*i*VCCS 2ndG) was designed to leverage a more powerful, 32-bit microcontroller equipped with an on-die digital signal processing (DSP) core. Special attention to power consumption led to the development of *i*VCCS 2ndG platform.

1.2 Magnetometer Theory of Operation

Geomagnetic field is a force field that surrounds the earth's surface and flows from the earth's interior into the space. Similar to a field generated by a simple bar magnet tilted, the force is currently estimated at an 11.5° angle. Although intensity is in constant flux and has experienced large-amplitude variations over the past 800 thousand years [9], the field is considered rather uniform at any region on the earth's surface, ranging between 25 and 65 µT (0.25 to 0.65 gauss).

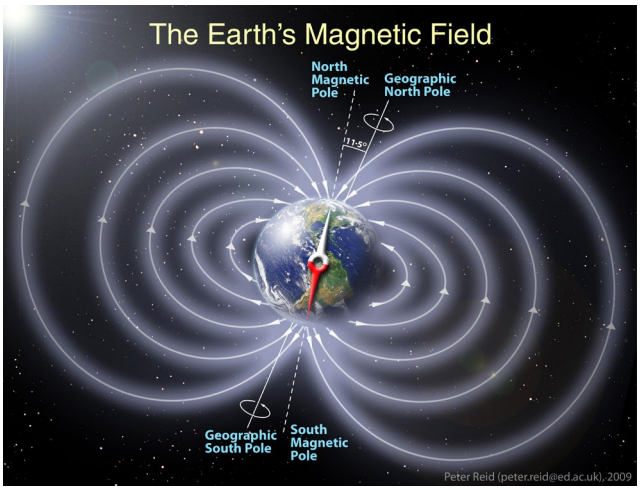

Figure 1. Magnetic field of the earth [10].

Ferrous materials in traveling vehicles cause small, local disruptions in the geomagnetic field. Due to high magnetic permeability of these ferrous materials, the flux lines of the field are absorbed and distorted in a non-linear form when a vehicle passes through the field, as shown in Figure 2. Several

factors are at play in the resulting intensity magnitude and direction of field changes (e.g., speed, density, and size mechanical form).

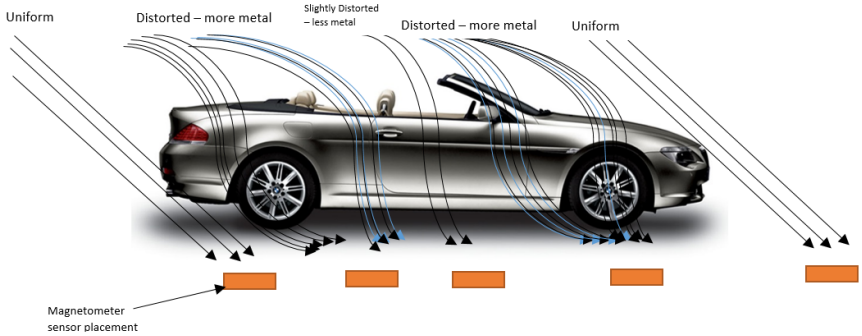

Figure 2. Disturbance in magnetic flux lines caused by a vehicle.

Measured disturbance is known as a vehicle magnetic signature, which is unique and differs from one vehicle to another. The disturbance can be modeled as a number of magnetic dipoles [11]. Signatures can be measured by magnetometers, which report three geomagnetic components of the field: north B_x, east B_y, and vertical B_z.

1.3 Related Work

Utilizing magnetic sensors for vehicle detection began in early 1978 [12] when a fluxgate magnetic sensor was used to actuate a lighting system as a vehicle passed the sensor. A recent study [13] proposed a 2-axis MAG for detecting vehicle driving direction. A detection rate of 99% was observed when traveling vehicles passed closely to the sensor. Performance degraded to 89% as the signal-to-noise ratio (SNR) decreased. A two-threshold, four-state machine algorithm was proposed in [14] for vehicle detection using a 3-axis AMR sensor. An active magnetic detection method was introduced in [15]. Although this method solved the baseline drift problem, it was not efficient in power, cost, or size. Authors in [16] proposed a short-time transform detection and recognition algorithm using a MAG sampled at 2KHz. Work proposed in [17] integrated an IEEE 802.15.4 transceiver with 32-bit MCU and 1-axis AMR for a vehicle counting and collision warning application. A 3-axis MAG was used in [18] for vehicle detection in parking lots. In [19], a street parking system using a MAG was introduced, and in [20], researchers proposed a vehicle parking detection method using a normalized cross-correlation of a 3-axis MAG signal. Authors in [21] proposed a scheme for identifying the heading direction of a moving vehicle using a 2-axis MAG. A wireless link budget study for intersection monitoring using MAG was proposed in [22].

Vehicle speed estimation is a key parameter for traffic surveillance applications [23]. Essential applications demanding speed estimation included length-based vehicle classification [24]; travel time estimation [25]; ramp-metering queue length estimation [26]; work zone safety [27]; curve warning [28]; vehicle emissions estimation [29]; and traffic light control, among many others. Lately, solutions based on cost- and power-efficient sensors (e.g., acoustic, MAG, and ACCEL) have become increasingly popular. Each of these has advantages and disadvantages. For example, acoustic and ultrasound sensors are very sensitive to dirt and background noise (e.g., vehicles or wind). In this study, vehicle speed was estimated by means of MAG sensors.

While technological advancements have reached a point at which application performance is ensured, energy efficiency remains challenging, especially given IoT integration. Modeling power consumption at the early stages of development and pre-deployment becomes decisive. Authors in [30] propose a general methodology to simulate wireless sensor node power consumption by profiling three essential aspects of WSNs: networking, sampling, and processing. Although electrical component characteristics and wireless interfaces play a major role in determining power consumption, a particular standard, component, and technology becomes crucial when developing an algorithm. A model for performance evaluation is necessary for optimization.

Literature has suggested building a predictor for estimating remaining power in the main energy storage, as well as the amount of power harvested during future time slots [31], [32]. Such mechanisms aid in determining which functionalities to disable and for how long based on a corresponding peripheral power profile. This stage of development can benefit from the power consumption model proposed in previous references. In [31], authors optimized RF transmission time—the most power-hungry system element—to determine an optimal transmission interval.

Software development is dependent upon energy consumption. Given that a nonoptimal algorithm is executed for energy consumption, selecting low-power components is inconsequential. In fact, several techniques must then be orchestrated to achieve a satisfactory performance. Most energy management implementations involve smart and intelligent use of peripherals and system sensors [31]–[34]. Indeed, the heart of the power management process should be an algorithm that determines when to enable and disable each module on the board, in addition to flexibly controlling and switching between power schemes. This is preferred over a fixed profile that runs the system in its many situations. Such an algorithm is commonly referred to as Dynamic Power Management (DPM).

Researchers have defined and tried to resolve the DPM problem. Authors in [35] modeled system behavior as a time series of busy and idle periods, suggesting that various idle intervals require different sleep modes, primarily because switching takes time and more energy than nominal consumption. They employed adaptive tree learning as a method for solving policy selection wherein a predictor is used to indicate when and for how long idle states occur.

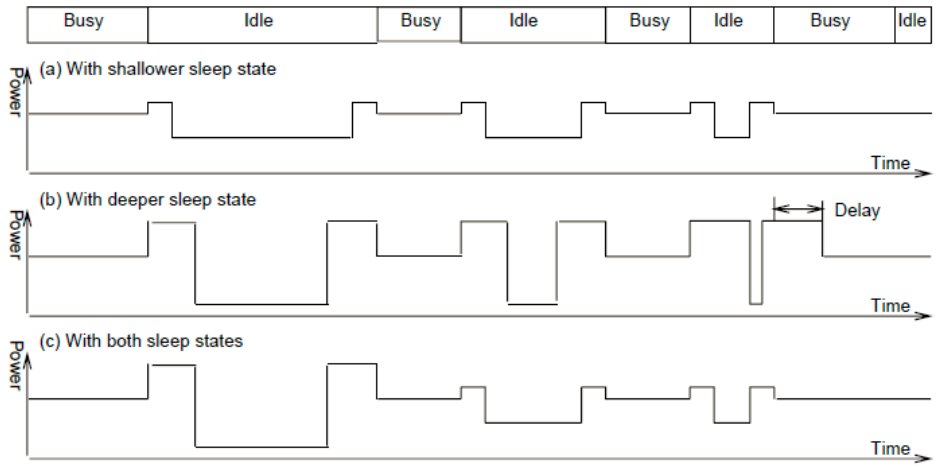

Figure 3 Changing system sleep mode according to load [35].

In [36], authors used RL to adaptively choose the optimal power management policy in a given idle state, wherein an action is taken and objectives are evaluated. Based on outcomes, subject-to-constraints (e.g., performance) state-action pairs are rewarded/punished and assigned a Q-value. A variation of the Q-Learning algorithm, namely SARSA (State-Action-Reward-State-Action), is used to monitor values in a Q-table. Idle time is known and can be calculated according to assumptions.

A more recent paper [37] used the Q-learning algorithm to optimize power consumption in a video encoder System-on-Chip (SoC).

Authors in [38] proposed a machine learning algorithm to select the optimal policy online. The algorithm considers a variant of RL with weights placed on policies and a loss function to update weights. Assumed idle times are known, although the way in which loss function affects weight is not clear.

Chapter 2 *i*VCCS Second Generation

*i*VCCS 2ndG is a fully autonomous system designed to achieve self-powered, low-power operation through energy harvesting without sacrificing performance. The new system is carefully crafted in a compact design (45×30×6 mm) that combines high-performance, energy-efficient components and is equipped with a power management subsystem for minimizing energy consumption while maintaining accurate vehicle count, logging, and speed estimation. Figure 4 shows the interconnections among various components. A gauge monitors and reports battery capacity to the microcontroller unit (MCU). Nano-power load switches are placed at the power lines of certain energy-hungry sensor components, including the RF wireless module, GPS receiver, and SD card, among others. Energy-efficient algorithms were developed to accurately detect, count, estimate speed, and control various system components [39]. Figure 5 shows top and bottom views of the printed circuit board.

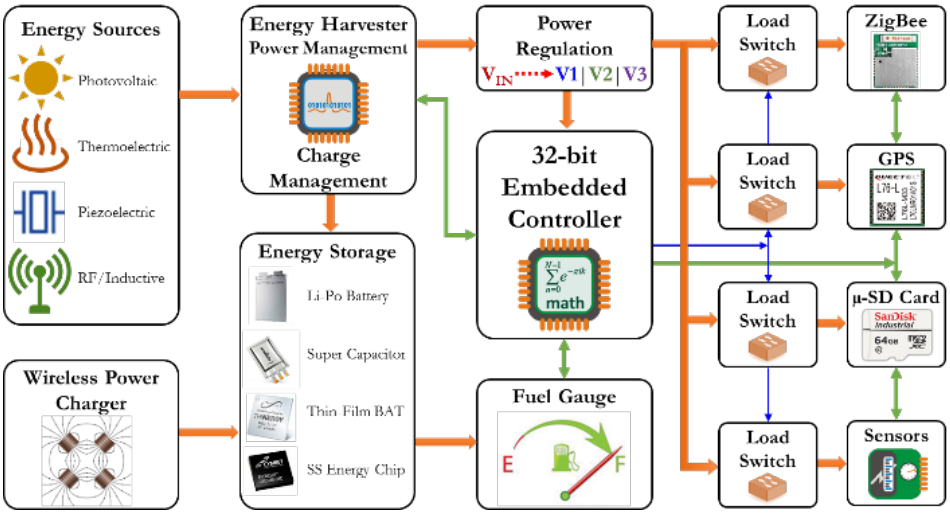

Figure 4. Overview of system components [6].

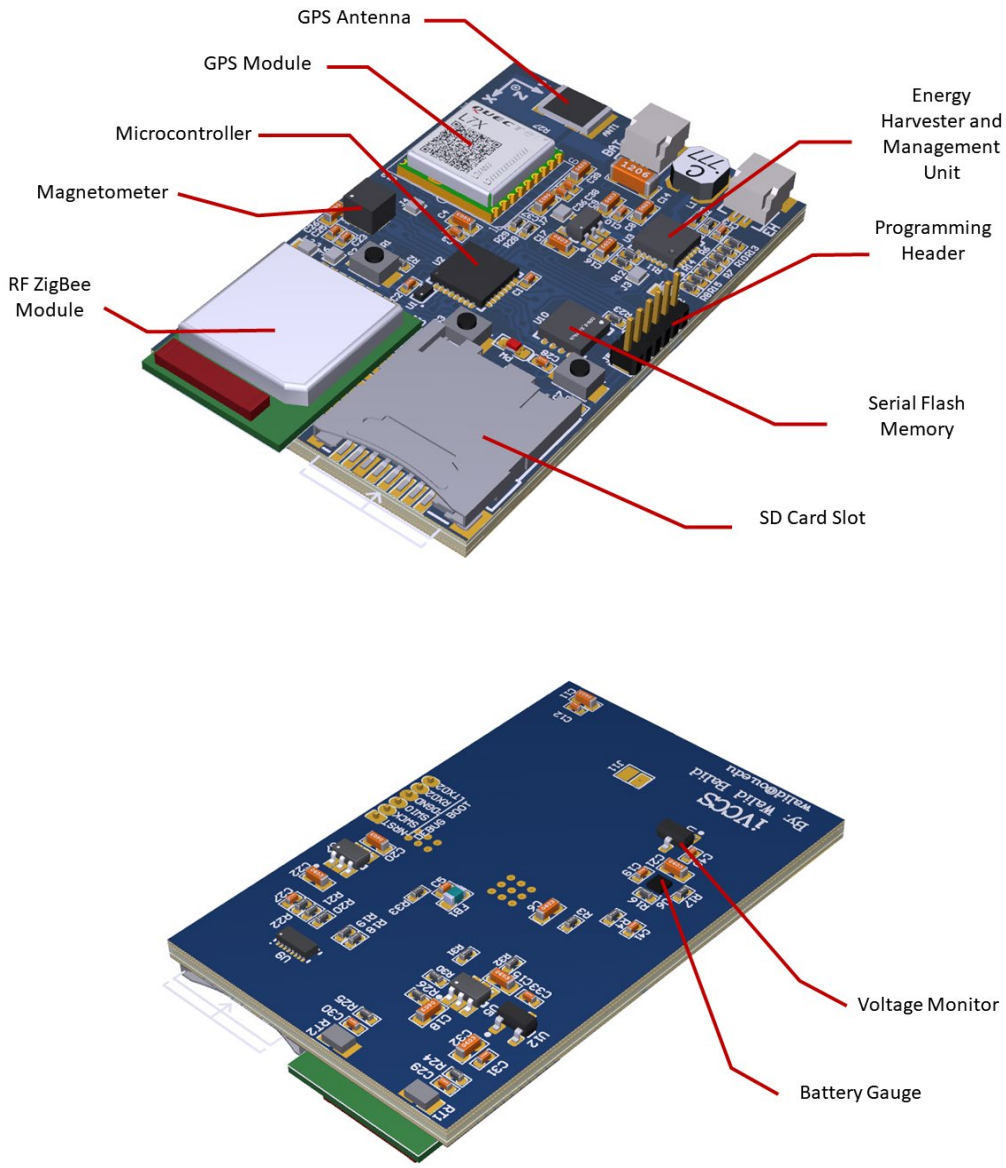

Figure 5. *i*VCCS PCB components.

2.1 Platform Overview

2.1.1 Microcontroller

At the heart of the sensor system lies an ultra-low power platform manufactured by STMicroelectronics, equipped with an ARM CORTEX-M0+ 32-bit RISC core – STM32L071 [40]. This microcontroller is characterized by number of features that make it appealing for low-power applications [41]. These include:

- 0.29 µA standby mode (3 wakeup pins)

- 0.43 µA stop mode (16 wakeup lines)
- 0.86 µA stop mode + RTC + 20 KB RAM retention
- Down to 93 µA/MHz in run mode
- 5 µs wakeup time (from flash memory)
- Core from 32 kHz up to 32 MHz max
- 1 to 25 MHz crystal oscillator
- 32 kHz oscillator for RTC with calibration
- Up to 192 KB flash memory with ECC
- 20 KB RAM

2.1.2 Magnetometer

Kionix KMX62 is a 6 degrees-of-freedom magnetometer/accelerometer inertial sensor system [42] that senses changes in the magnetic flux of earth's surface due to vehicles passing in its vicinity. KMX62 is a very reliable and power-efficient sensor, consuming 395 µA at high-resolution mode and 1 µA in standby mode. The magnetometer's full-scale range is ±1200 µT, and digital bit depth is 16 bits, resulting in a magnetic sensitivity of ±0.0366 µT compared to ±0.1 µT in *i*VCCS 1stG.

2.1.3 ZigBee Module

The radio frequency (RF) frontend utilizes a ZigBee transceiver from ZLG based on an NXP JN5168 microcontroller [43]. AW516x is a low-power, high-performance ZigBee module that incorporates the IEEE 802.15.4 standard and supports a variety of protocols on top (e.g., FastZigBee, ZNET, ZigBee-PRO, and RF4CE). The AW5161P0CF module used in *i*VCCS 2ndG has a small footprint (13.5×16.5 mm) and features a ceramic antenna. Receiver sensitivity is -95 dBm, consuming 21 mA, and transmitter power is 2.5 dBm with 18 mA current consumption. The engine features a deep sleep mode with 100 nA typical current. System data transfer rate is 2 Mbps.

2.1.4 GPS

Time synchronization is achieved via a GPS module manufactured by Quectel. The L76-L GPS receiver module integrates GLONASS and GPS systems [44] and provides a built-in, low-noise amplifier (LNA) for improved performance for 33 tracking channels, 99 acquisition channels, and 210 PRN channels. The module features EASY technology [44] and allows L76-L to automatically calculate and predict orbits using ephemeris data stored in the module's internal flash. This reduces time-to-fix even in indoor situations with poor signal levels. Current consumption is 25 mA in acquisition mode and 7 µA in backup mode.

2.1.5 Real-Time Clock

STM32L071 also offers an internal Real-Time Clock (RTC) and features low power operation [41]. It has on-the-fly correction capabilities with a range of 1 to 32767 RTC clock pulses, allowing the system's 1 Hz clock to synchronize with the Pulse Per Second (PPS) signal of the GPS module subsequent to fix acquisition. The internal RTC is driven by external clock of 32.768 kHz, ultra-low power oscillator manufactured by SiTime [45], which offers a current consumption of <1 µA and frequency stability of ±5, ±10, ±20 ppm options over temperature, as well as the world's smallest footprint at 1.5×0.8 mm CSP.

2.1.6 Data Storage

The system integrates a microSD card slot used for raw data acquisition (e.g., vehicle magnetic signature or accelerometer), in addition to timestamps for vehicle arrival/departure and status messages. The micro-SD card is connected to the microcontroller through serial peripheral interface (SPI). An on-board 64 Mb serial NOR flash memory is also available as a secondary storage medium. Macronix's MX25R64 [46] is an ultra-low power CMOS flash memory with a minimum of 100,000 erase/program cycles and 20-year data retention. MX25R64 features a typical 5 µA standby current, a maximum 4 mA read current, and 6 mA write current.

2.1.7 Battery Gauge

As part of the power management subsystem, a smart battery gauge is integrated for monitoring battery capacity and protecting it from deep discharge and over charge, especially for Li-Po batteries. Texas Instruments BQ27621-G1 [47] was used, as it provides advanced algorithms for calculating remaining battery capacity (mAh), state-of-charge (%), battery voltage (mV), and temperature (°C).This technology also requires little to no configuration and can be accessed through a 400 kHz I^2C interface. Figure 6 shows a simplified schematic of gauge-battery pack connection.

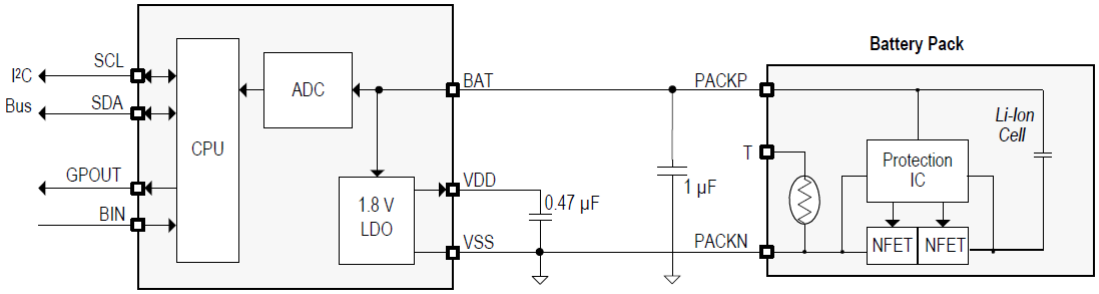

Figure 6. Simplified schematic of the battery gauge unit [47]**.**

2.1.8 Energy Harvester and Management Unit

ADP5091 [48] is an ultra-low PMU that harvests energy from photovoltaic (PV) and thermoelectric generator (TEG) sources. The unit can efficiently convert power from sources with a range as low as 6 µW and as high as 600 mW without losing significant energy (i.e., sub-microwatt). The chip features a cold-start circuit with input voltage as low as 380 mV. After cold-start, the regulator can operate at an input between 80 mV to 3.3 V. Stable DC-to-DC boost conversion is realized through the use of a maximum power point tracking (MPPT) controller.

2.2 System Algorithms

In essence, the algorithms in 1st and 2ndG are identical in functionalities to offer. However, the different designs necessitate software code changing to ensure accurate implementation of these functionalities. Another important aspect of *i*VCCS 2ndG features low-power operation. Notably, storage and communication are two of the most energy-hungry components of an embedded system. In this research, algorithms developed for mitigating power consumption induced by the microSD card, ZigBee, and continuous data fetching from the magnetometer KMX62 are discussed.

2.2.1 Data Buffering Technique

A microSD card is used for logging timestamps and storing other status messages and raw data. However, microSD cards are energy inefficient and counter intuitive to the adapted low power paradigm, primarily because they simply dump data and messages directly to the card. According to SanDisk microSD card specifications [49], read/write procedures can take up to 100 mA in current consumption. Turning the card on/off when data must be stored is impractical due to required time delay for repeated card initialization each time the card is powered on. Specifications also refer to the automatic sleep feature wherein cards enter sleep mode, given no commands are received within 5 ms. In this mode, cards consume 350 µA, which indicates inefficiency for system duration. Consequently, the on-board, ultra-low power flash memory is utilized to buffer data before shifting it to the microSD card, as the memory excels in energy efficiency and read/write performance when compared to the microSD card. The result is suitable for instantaneous data logging.

1: **IF** New Sector **THEN**
2: Erase Sector
3: END IF
4: RAM Buffer ← Data
5: **IF** RAM Buffer Size ≥ 256 **THEN**
6: Move data: Flash ← RAM
7: Shift remaining data in RAM
8: END IF
9: **IF** Flash Full Pages ≥ Flash Threshold **THEN**
10: Open new file in SD card
11: **FOR EACH** Flash Page i **DO**
12: SD card ← Flash page i
13: END FOR
14: SD card ← RAM Buffer
15: Close file in SD card
16: END IF

Figure 7 High level description of data buffering technique.

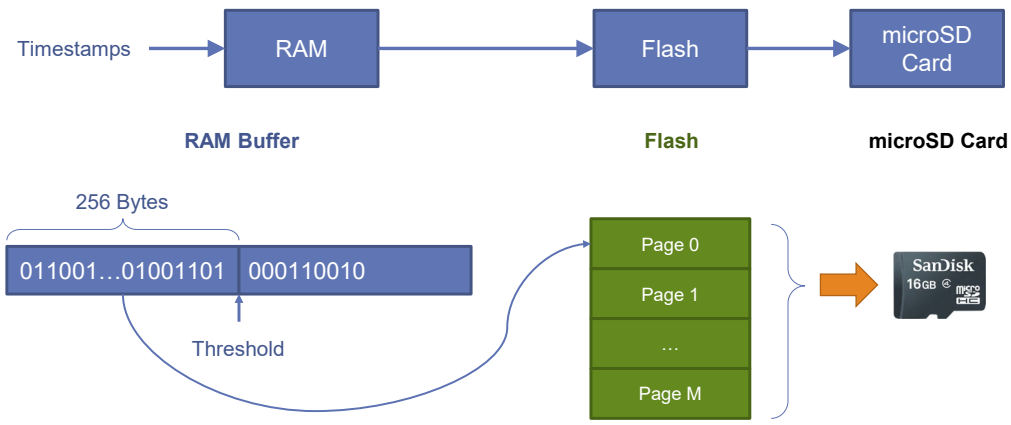

Figure 8. Block diagram illustrating data buffering technique.

MX25R64 exhibits a page basis write functionality with 256-byte page size. Host can commence reading at any byte address, although it can write only at the beginning of a page address. Hence, the microcontroller must retain length of data required for the write function to avoid creating gaps in the flash memory. Figure 8 illustrates the procedure, and Figure 7 provides a high-level description of the algorithm.

2.2.2 Triggered Vehicle Detection

Typically—and in accordance with configured data rate (i.e., Data Ready Interrupt [DRI]), the magnetometer interrupts the microcontroller at any time in which a new sample is acquired in the buffer. Desired behavior occurs only when an approaching vehicle interrupts MCU, thus, triggering execution of the detection algorithm. KMX62 magnetometer 2 is interrupted, as are DRI, Magnetometer Motion Interrupt (MMI), and Buffer Full Interrupt (BFI). MMI is issued when the difference between two consecutive samples on one axis reaches a programmed threshold in a specific direction (i.e., either increasing or decreasing) and stays above the threshold indicated for a specific number of samples (i.e., time). The KMX62 buffer can operate in triggered mode and hold 64 samples of three components: x, y, z. Given a physical interrupt is caused a digital engine (i.e., magnetometer or accelerometer), a trigger event is asserted, and SMP_TH number of samples prior to the event are retained. Sample collection continues until the buffer is full. Data is reported in chronological order, as explained in Figure 9. MMI is configured to trigger the buffer, and BFI is routed to MCU through one of the GPIOs [42]. SMP_TH is set to 63, which causes KMX62 to immediately interrupt MCU subsequent to the reception of the first sample received after an MMI interrupt (i.e., vehicle approach/detection).

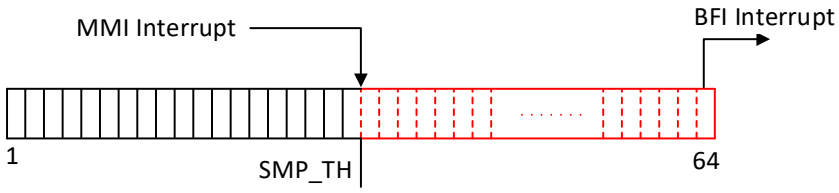

Figure 9. MMI and BFI operation in KMX62 magnetometer.

Figure 10 describes the way in which the algorithm reads and controls data flow from KMX62. MCU completes a dummy read for a number of samples, primarily because the first few samples might not relate to a vehicle's signature. The algorithm is configured to consider the last 24 samples prior to vehicle approach, discarding the first 40 samples from the buffer. After 24 samples are read and processed—and given the state machine of the system still indicates detection, the MCU continues pulling new samples through Data Ready Registers until the vehicle has departed.

2.2.3 Communication Scheme

The primary role of RF in this system is to report data (i.e., count and timestamps) to an access point (AP). This function is completed once each day, typically at midnight. However, the system is also capable of interacting with user requests. Notably, the ZigBee module is one of the most power-consuming units on board, and constantly turning it on will quickly deplete the battery.

```
1:  IF DRI Flag THEN
2:  Read data from DRI registers
3:  ELSE IF BFI Flag THEN
4:  Disable BFI
5:  Discard 40 samples
6:  WHILE Samples Counter < 24 THEN
7:  Read data from Samples Buffer
8:  Increase Samples Counter
9:  END WHILE
10: IF State = Detection THEN
11: Enable DRI
12: ELSE
13: Enable BFI
14: END IF
15: END IF
```

Figure 10 High level description of triggered vehicle detection algorithm.

MCU is not required to use RF interface, except for initial status reporting after sensor is initialized and after data reporting at midnight. To accommodate user/AP requests, MCU must activate the ZigBee module for one minute, during which time it sends a status message reporting battery charge level and number of vehicles counted. Given that a command is received within this minute, the timer is reset, allowing additional commands. If none are received, RF module is shut down.

The optimization methods and algorithms introduced in the previous chapters can be deemed hard-coded workarounds to minimize power consumption, as they lack flexibility and resilience for adapting to varying conditions. Although some policies lead to significant energy savings, cost to the system is response time and accuracy. Hence, proper orchestration between alternate methods and policies is necessary. Given the DPM problem, an algorithm must control power/performance tradeoff according to workload on the system. DPM and RL are described in this chapter in an effort to further improve system power consumption.

2.3 Reinforcement Learning for Power Management

2.3.1 Introduction to Reinforcement Learning

Reinforcement Learning (RL) is a branch of Artificial Intelligent (AI) algorithms that mimics a natural way of gaining knowledge and experience (i.e., learning through trial and error). Typically, this technique is intended for classifying algorithms into one of two AI categories: supervised and unsupervised learning. Supervised learning agents are trained on a set of labeled data provided by an external expert, while unsupervised learning agents are trained on data that has no labels with a goal of finding hidden structure within unlabeled data [50]. Many researchers argue that RL falls somewhere between these two categories. In fact, RL can be considered an unsupervised method because it does not rely on expert, accurate-based data set examples [50]. Instead of finding hidden structures, RL attempts to maximize sparse rewards based on necessary agent learning to behave in a given environment.

Although the concept is natural and intuitive, several challenges are characteristic of RL that are not characteristic of other types of machine learning (e.g., dilemma of exploration and exploitation). Example questions include the following. Should the agent *exploit* the knowledge gained by selecting actions to maximize its rewards or should the agent *explore* by selecting alternative untried actions for building a more realistic estimate and understanding of the environment? With regard to the credit assignment, how should one determine rewards and penalties associated with actions that lead to an arbitrary state?

The most common way of formalizing an RL problem is utilizing Markov Decision Processes (MDPs) [51], consisting of:

- finite state space S,
- set of available actions A,
- reward function $R: S \times A \rightarrow R$, and
- system dynamics function $P: S \times S \times A \rightarrow [0,1]$,

which can be written as $P(s'|s,a)$ where:

$$\sum_{s' \in S} P(s'|s,a) = 1 \quad \forall\, s \in S, \forall\, a \in A$$

In other words, function P determines the probability of transitioning to state s', given that the agent remains in state s and performs action a. However, in an RL problem the agent has no initial or prior knowledge of the rewards function nor the system dynamics function.

Q-Learning is a standard RL algorithm in which agent experience is expressed as sequences of states, actions, and rewards:

$$< s_0, a_0, r_1, s_1, a_1, r_2, \dots >$$

In state s_0, the agent performed action a_0 and was rewarded r_1, resulting in transitioning to state s_1, and so on. Interactions with the environment form agent experience accumulated over time, with agent aiming to maximize experience value—typically realized as a discounted future reward. In Q-Learning, this value is given by function $Q^*(s,a)$, defined as the expected value of the cumulative discounted reward of performing action a in state s, and, hence, following the optimal path.

In general, the discounted future reward is defined as the immediate reward the agent receives for the current action taken and the future reward. Given a finite series of experiences

$$< s_0, a_0, r_1, s_1, a_1, r_2, \dots, s_{n-1}, a_{n-1}, r_n, s_n >,$$

total reward would be

$$R = r_1 + r_2 + r_3 + \cdots + r_n$$

Therefore, future reward accumulated from time t onward is

$$R_t = r_t + r_{t+1} + r_{t+2} + \cdots + r_n$$

However, due to the stochastic nature of the environment, the future reward for a given action is not guaranteed to be the same when chosen next. Thus, a discount factor is introduced to impart less significance on future rewards when compared with a current reward, as shown below:

$$R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^{n-t} r_n$$

where $0 \leq \gamma < 1$ is the discount factor. If $\gamma = 0$, then the algorithm will be short-sighted, relying on only the immediate reward. If $\gamma = 1$, total future reward would be the same. Thus, it would make sense if identical actions consistently reap the same rewards in a deterministic environment. A more balanced approach would be $\gamma = 0.9$.

In this case, the discounted reward can be rewritten as

$$\begin{aligned} R &= \sum_{i=1}^{\infty} \gamma^{i-1} r_i \\ &= r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots + \gamma^{i-1} r_i + \cdots \\ &= r_1 + \gamma(r_2 + \gamma(r_3 + \cdots)) \end{aligned}$$

Let R_t be the reward accumulated from time t

$$\begin{aligned} R_t &= r_t + \gamma\big(r_{t+1} + \gamma(r_{t+2} + \cdots)\big) \\ &= r_t + \gamma R_{t+1} \end{aligned}$$

Q-function will be discussed in more detail in subsequent sections of this chapter.

2.3.2 RL for *i*VCCS

Originally, the sensor exhibited an all-on policy, wherein all on-board components were turned on. This approach was not an intuitive approach, and the sensor lasted barely 40 hours. A more flexible operation was added wherein certain components were turned on and off following a predetermined action flow, resulting in reduced current consumption. To further enhance power consumption, sleep mode of ARM Cortex-M0 microprocessor was incorporated into the operation. However, this caused misdetections and/or double counting vehicles with less than two-second following distance, as a consequence of long wake-up time from sleep mode following detection of a passing vehicle. Sensor power cycle can be divided into several phases, namely

1. System initialization,
2. Data buffering,
3. ZigBee periodic communication, and
4. Standby/detection.

System initialization is a one-time phase and, therefore, has a negligible effect on long-term power consumption. Data buffering is part of the data-driven phase and is related to the number of detected vehicles and traffic flow. This phase was incorporated to reduce current consumed by microSD card and operation duty-cycle. ZigBee periodic communication was also implemented to reduce operation duty-cycle. Although sleep mode drains current at 2 mA, standby phase drains at 4 mA. The problem with sleep mode, as mentioned earlier, is that system response time increases and causes misdetections. Notably, in a low traffic condition (e.g., vehicle following distance is more than 5 seconds), sleep mode is expected to perform faultlessly while at the same time conserve more power. A DPM was introduced to solve this problem.

2.3.3 Problem Formulation

An RL algorithm based on Q-Learning was proposed as an approach to intelligently optimize power policies selection in a given state.

Assume the following:

- System can
  - observe the environment (i.e., traffic congestion) and
  - measure power consumption.
- Environment behavior can be modeled as high traffic (HT)/low traffic (LT).
- Power policies available to the agent (i.e., sensor) include
  - High Power (HP) mode and
  - Low Power (LP) mode.

The system is modeled as an MDP with four-state space: High-Power High-Traffic (HP-HT), High-Power Low-Traffic (HP-LT), Low-Power High Traffic (LP-HT), and Low-Power Low-Traffic (LP-LT). Figure 11 shows the state transition diagram of the system. Solid arrows are transitions made by the agent, and dashed arrows are transitions made by environment. Although agent is not penalized for transitions made by the environment, transitions should have high penalty in the reward matrix to prevent agent from taking such an action.

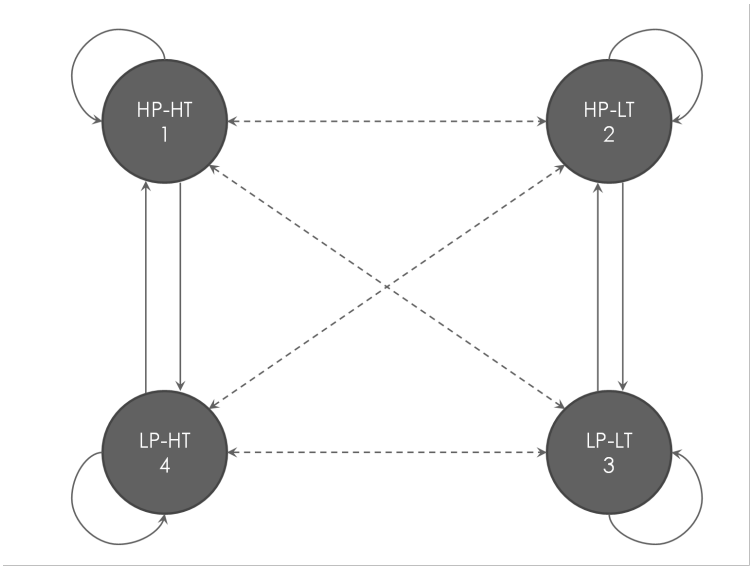

Figure 11. State transition diagram.

The agent (i.e., *i*VCCS) can switch to either high-power or low-power mode. Reward function is defined as an R matrix, as indicated below.

$$R = \begin{bmatrix} 15 & -50 & -50 & -50 \\ -50 & -50 & 15 & -50 \\ -50 & -20 & 15 & -50 \\ 15 & -50 & -50 & -50 \end{bmatrix}$$

A tuple of state, action, reward $< s, a, r >$ forms an experience in the table. Q-function constructs the Q-table by giving the expected value of reward (i.e., Q-value). The Q-table can be either randomly initialized or zero initialized, as indicated below.

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

2.3.4 Bellman Equation vs. Temporal Differences Equation

The proposed model consists of four states, as discussed earlier, namely HP-HT, HP-LT, LP-HT, and LP-LT. Although the agent is able to control its state in HP or LP, the environment controls sensor state through traffic, placing it in high or low traffic. Sensor should read information for either to determine its state. When the agent performs an action, a corresponding reward from R matrix is selected and a new experience (i.e., tuple of $< s, a, r >$) is formed. This result is the Q-value $Q^*(s,a)$, which is the expected value of action a in state s, and then following optimal policy, defined as

$$Q^*(s,a) = \sum_{s'} P(s'|s,a)[\mathcal{R}^a_{ss'} + \gamma V^*(s')],$$

where $V^*(s)$ is the expected value of following an optimal policy from state s.

Bellman equation is used to estimate this value, as follows:

$$Q(s,a) = r + \gamma \max_{a'} Q(s',a'),$$

where $\max_{a'} Q(s',a')$ is the maximum Q-value in the experience table (Q-table) for future state s' over all possible actions a'. Bellman equation provides the maximum future reward as the reward received for the action taken in the current state s plus the maximum future reward for the next state s'. In some RL problems, newer values of Q can be given additional weight as they are deemed more accurate to increase their influence. To weigh later experiences, the following Temporal Differences (TD) equation can be used:

$$Q(s,a) = Q(s,a) + \alpha\left(r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right).$$

- $0 < \alpha \leq 1$ determines the weight of newer values compared to older ones.
- γ is the discount factor.
- r is the reward corresponding to an action from R matrix.

Both cases were simulated for studying convergences. Figure 12 shows that Bellman equation converges faster than the TD equation by a factor of 10 times.

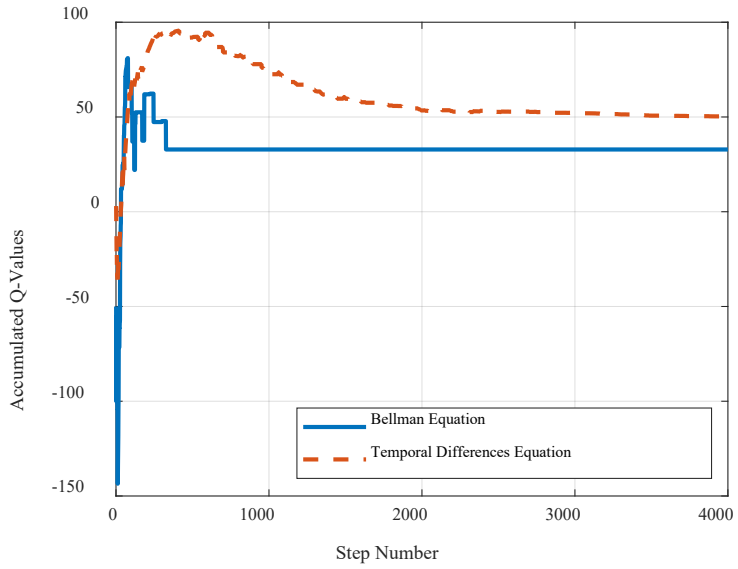

Figure 12. Q-Matrix convergence: Bellman vs. TD equation.

By looking at the normalized Q-matrix for both cases, it is clear that the agent was rewarded for going to optimal states 1 and 3, which correspond to HP-HT and LP-LT, respectively. Although not optimal, some rewards were given for transitioning into state 2 from state 3, since this action does not affect detection and counting accuracy at the expense of power consumption. Figure 13 and Figure 14 depict the normalized experience matrices (i.e., Q-Matrix).

Figure 13. Normalized Q-Matrix for Bellman equation.

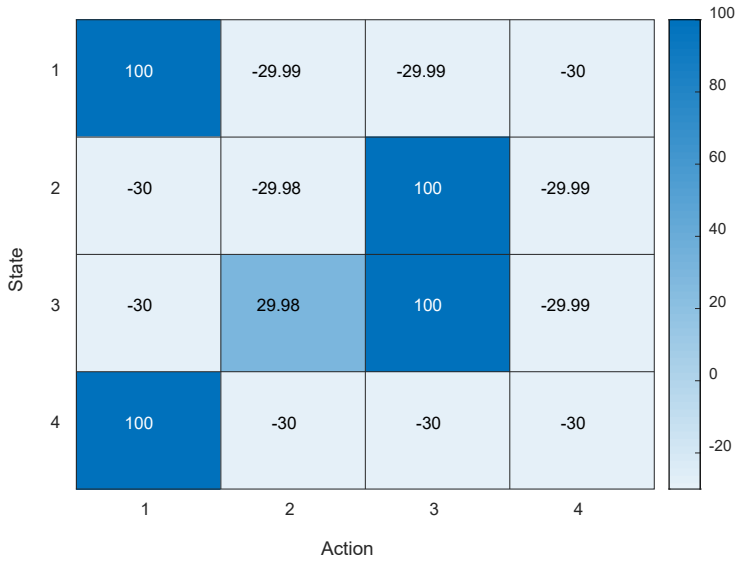

Figure 14. Normalized Q-Matrix for TD equation.

A small difference can be observed in the Q-Matrix and accumulated Q-values of TD equation. This is attributed to the fact that TD is a suboptimal solution to Bellman functional equation [52]. States and actions 1, 2, 3, and 4 refer to HP-HT, HP-LT, LP-LT, and LP-HT, respectively. Given the fast convergence of the Bellman equation, it is prudent to be chosen as the system Q-function.

2.3.5 Reducing Number of Actions

In the previous section, the simulation was accomplished when agent had the option to move to any state. However, the agent was penalized for choosing actions that resulted in states controlled by the environment. In this section, the number of actions is reduced to LP and HP.

This approach eliminates the possibility of choosing non-permitted actions, such as those controlled by the environment. It also enhances convergence time by a factor of half, compared to results obtained earlier. Figure 15 and Figure 16 show Q-Matrix convergence and Q-Matrix normalized.

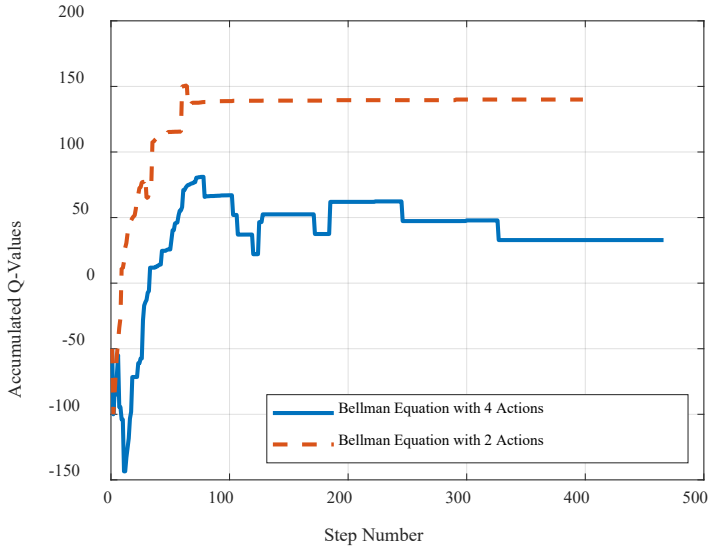

Figure 15. Bellman equation convergence: 2 actions vs 4 actions.

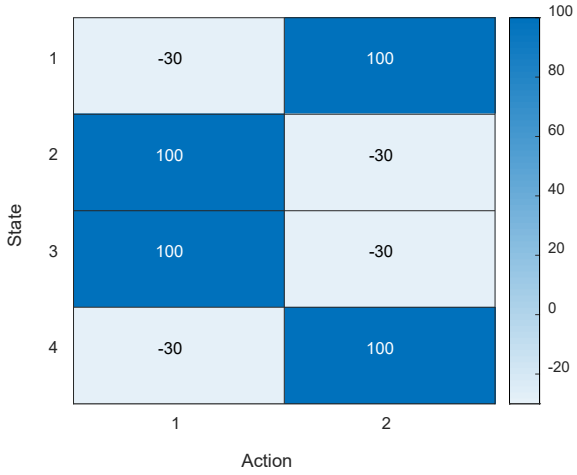

Figure 16. Q-Matrix using two actions.

2.3.6 Power Consumption Analysis

When employing an RL algorithm, circuit current drained by the system can be calculated utilizing a similar approach to the one described in [39], [53]. In this scheme, the algorithm places the microcontroller core in sleep mode during standby state, reducing current consumed to 2 mA. Given that a vehicle approaches the sensor zone, microcontroller experiences an interruption, exits sleep mode, and switches to detection state. Consequently, drained current becomes a function of the number of vehicles detected by the sensor. Figure 17 represents current consumed when a vehicle is detected. Vehicle time spent traveling over the sensor determines length of detection state, which was statistically calculated as an average 0.5 seconds.

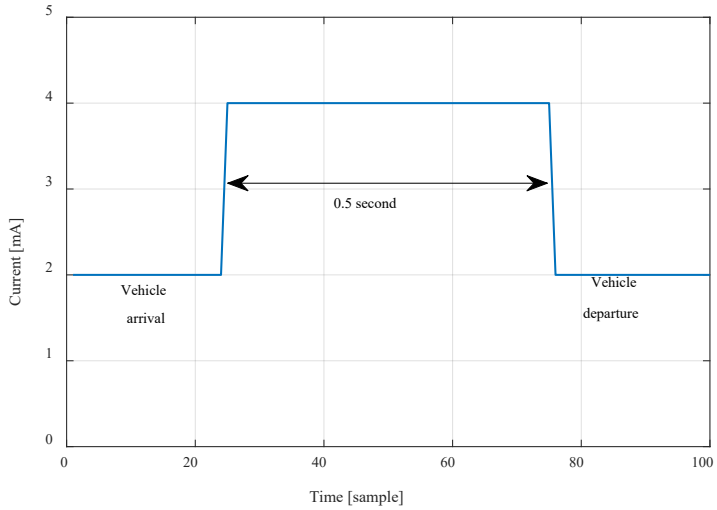

Figure 17. Current consumed when MCU wakes from sleep mode.

Acknowledging previous assumptions, average current equation can be rewritten to express current consumption, leveraging the RL algorithm, as shown below.

$$I_{avg} = \frac{T_B I_B + N_v T_v I_d + (3600 - T_B - N_v T_v) I_s}{3600} = 3.71\ [mA],$$

where N_v is number of vehicles detected per hour; T_v is the detection period; and I_s is the sleep state current. Given a 2300 mAh battery, battery life is extended, as follows:

$$T_{Bat} = \frac{2300 \times 0.8}{3.71} = 496\ [h]$$

2.3.7 Simulation using Real-World Data

Simulations detailed above were provided with synthetic data, where the agent is randomly assigned a state and is required to take appropriate action. This process was repeated over 5000 episodes, with each episode commencing in a random state and terminated when agent reaches goal state.

To simulate real-word scenarios, a code was developed to use the same sensor (e.g., *i*VCCS 2ndG) for combining vehicle timestamps collected during past deployments and forming the state of the agent using two variables: power policy (i.e., LP or HP) and traffic trend (i.e., HT or LT), the latter defined as vehicles detected per minute. A threshold of 10 vehicle/min separates high from low traffic. The algorithm was tested using two data sets: one collected on campus for 24 hours and another on Britton Highway for 3 hours. Power policy 1 represents low power; 2 represents high power. The following figures show traffic trend and power policy chosen over time for both data sets. Since campus traffic is rather low, the agent chose to keep power policy on low (i.e., 1). High traffic on Britton Highway caused the agent to maintain HP (i.e., 2). Switching between HP and LP can be observed at the beginning and end of the graphs for Britton Highway, on which traffic increases and decreases due to lane closure by traffic control for installation.

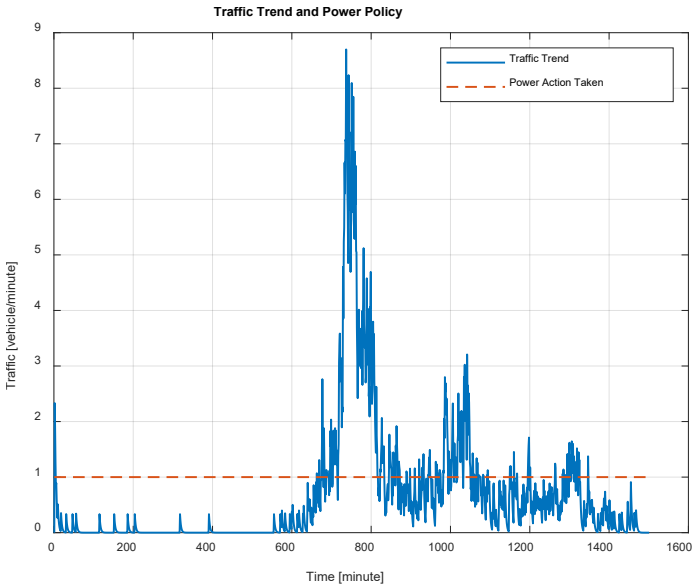

Figure 18. Traffic trend and power policy on campus.

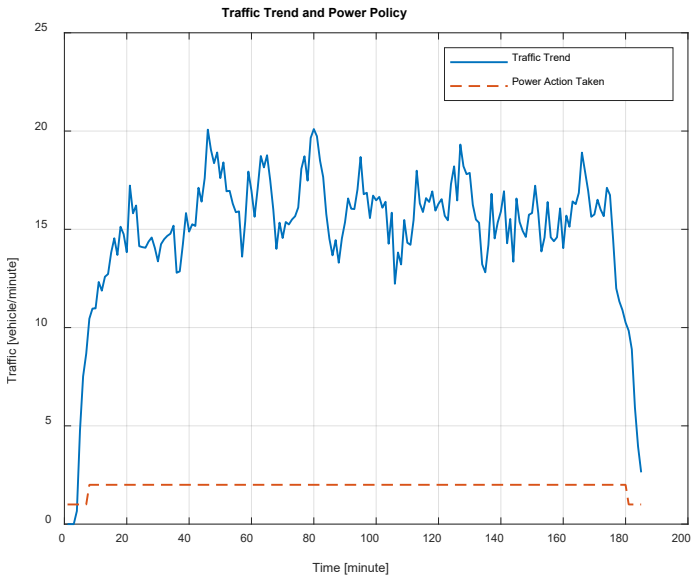

Figure 19. Traffic trend and power policy on Britton Highway.

2.4 Experiments and Results

The system was tested in two scenarios: a lab test using a train continuously running for 24 hours and a field test wherein two sensors were deployed at the south entrance of campus for 24 hours. The sensor captured vehicle count and speed estimates for individual cars. Reported speeds were in line with expectations and nominal values of a real-world setting.

2.4.1 Lab Test

In the lab test, a sensor was placed under a miniature train track with a train operating for 24 hours at varying speeds. Figure 20 illustrates the setup. The detection algorithm was validated, and battery life was examined.

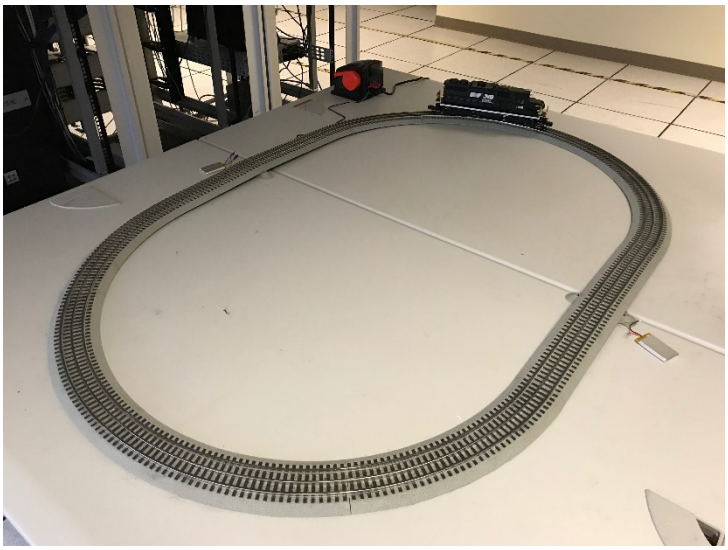

Figure 20. Lab test using a train and two sensors placed under the track.

Figure 21 lists the frequency at which the sensors detected the train, along with reported speed, which was calculated using distance between the two sensors.

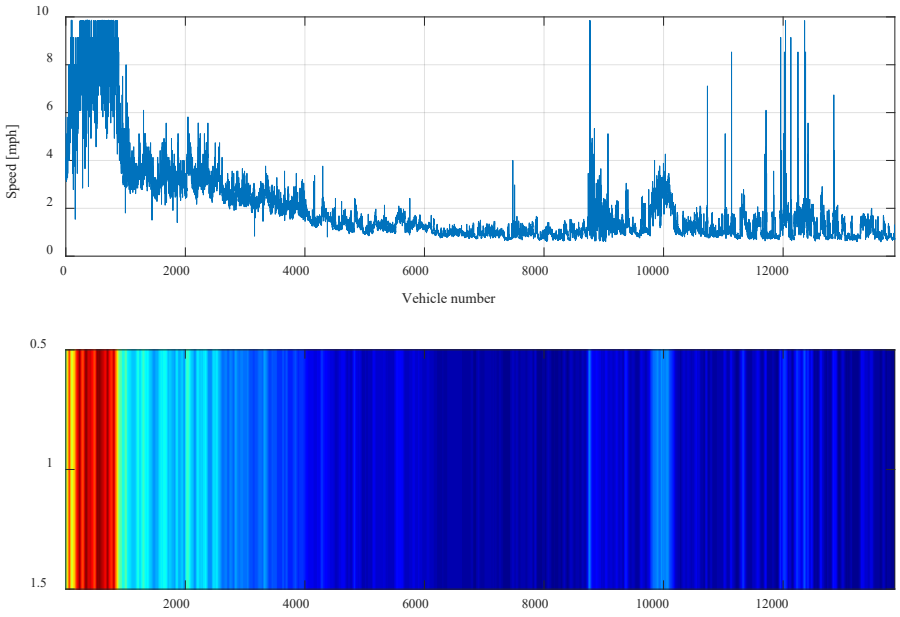

Figure 21. Lab test: Speed estimates of the train for a time period over 24 hours.

The sensor consumed only 11 mAh in over 24 hours (i.e., 0.458 mAh per hour), as shown in Figure 22. This amount is 10 times less than predicted in the analysis.

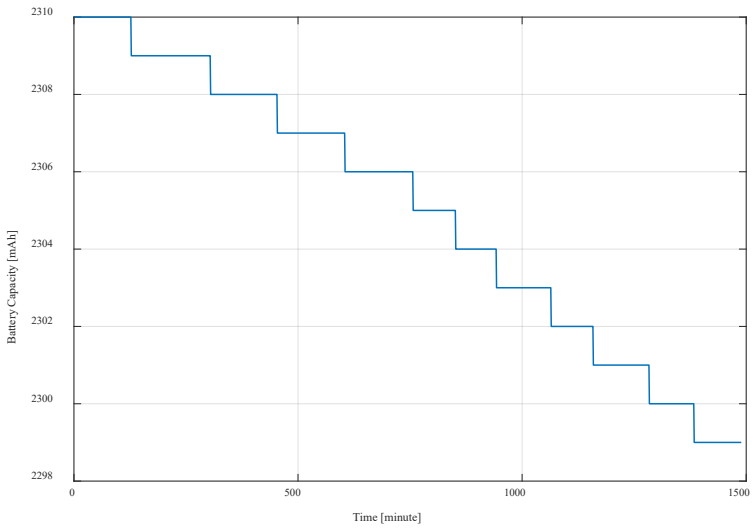

Figure 22. Lab test: Battery capacity of the sensor for a time period over 24 hours.

2.4.2 On-Campus Test

For the field test, two sensors were deployed at the south entrance of campus: one with the proposed DPM algorithm managing the power policy and the other running a plain version of the firmware. Figure 23 depicts the sensors' locations. Sensors were deployed for 24 hours, collecting the number of vehicles

that entered the campus, time of arrival and departure, as well as battery capacity for comparing power consumption of both versions of firmware. Speed was calculated in a postprocessing stage, wherein detected vehicle timestamps were used with known separation distance (e.g., 2 meters) between sensors.

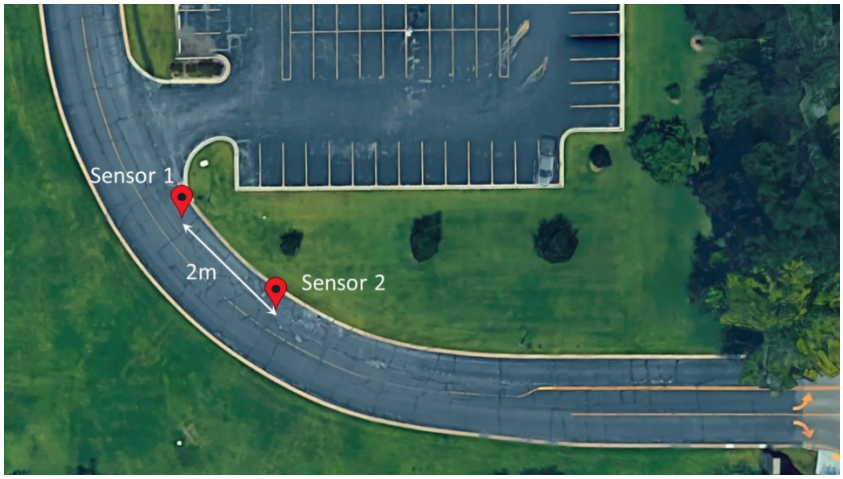

Figure 23. Campus field test setup.

Average speed was 6 mph, which is fairly logical given sensor location at the campus entrance and speed limit of 10 mph. Processing collected power data revealed that the sensor running original firmware (i.e., sans DPM) consumed approximately 18 mAh and resulted in a battery life of over 100 days.
Alternatively, the sensor executing the DPM algorithm consumed only 4 mAh over 24 hours, indicating a battery life of over 400 days. Both estimates assumed a 2300 mAh battery with a derating factor of 80%. The original sensor detected only 11 more vehicles (2.9%) than the sensor executing the DPM algorithm.

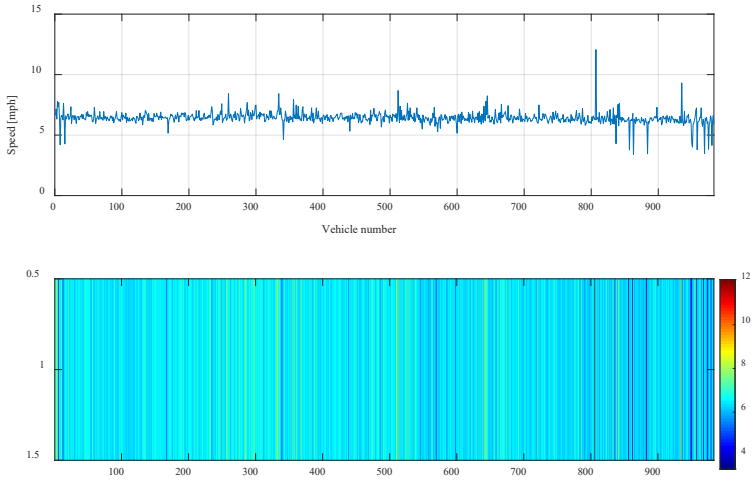

Figure 24. Campus field test: Number of vehicles detected and speed estimates.

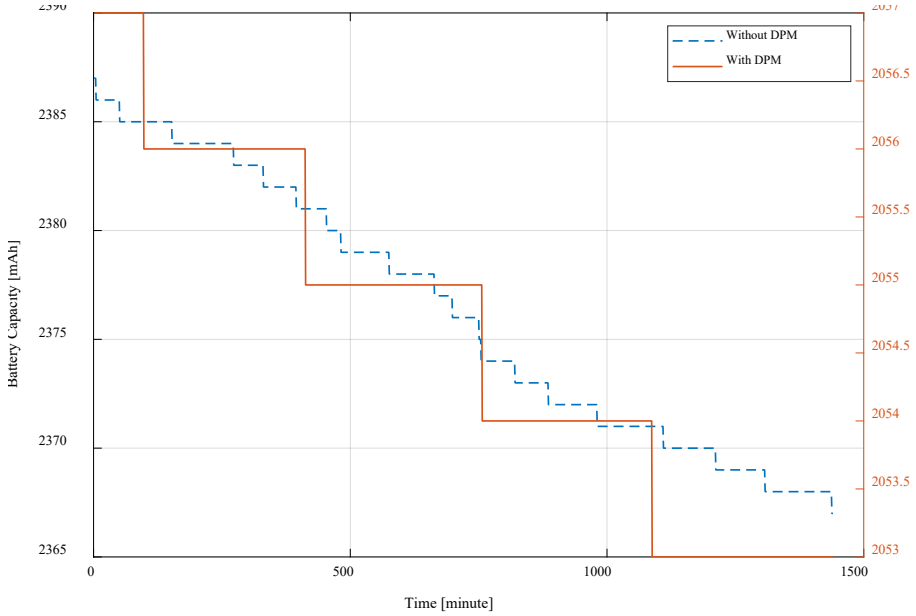

Figure 25. Battery capacity of sensor with and without DPM for a time period over 24 hours.

Chapter 3 *i*VCCS Third Generation

Sensor design was revised to reduce power consumption and provide additional processing power and improved communications capabilities. The first radical design change was replacing the microcontroller unit with an ARM CORTEX-M4F 32-bit core to provide additional processing power. More importantly, the new microprocessor has a floating-point unit that speeds up calculations and offers additional digital signal processing capabilities. The microSD card was replaced with an array of on-board flash memories to reduce its tremendous amount of consumed energy. The power management subsystem was simplified by eliminating energy harvesting. Instead, the sensor is powered by a very high capacity, ultra-wide-range operating temperature, compact size batteries that use Lithium-Thionyl chemistries.

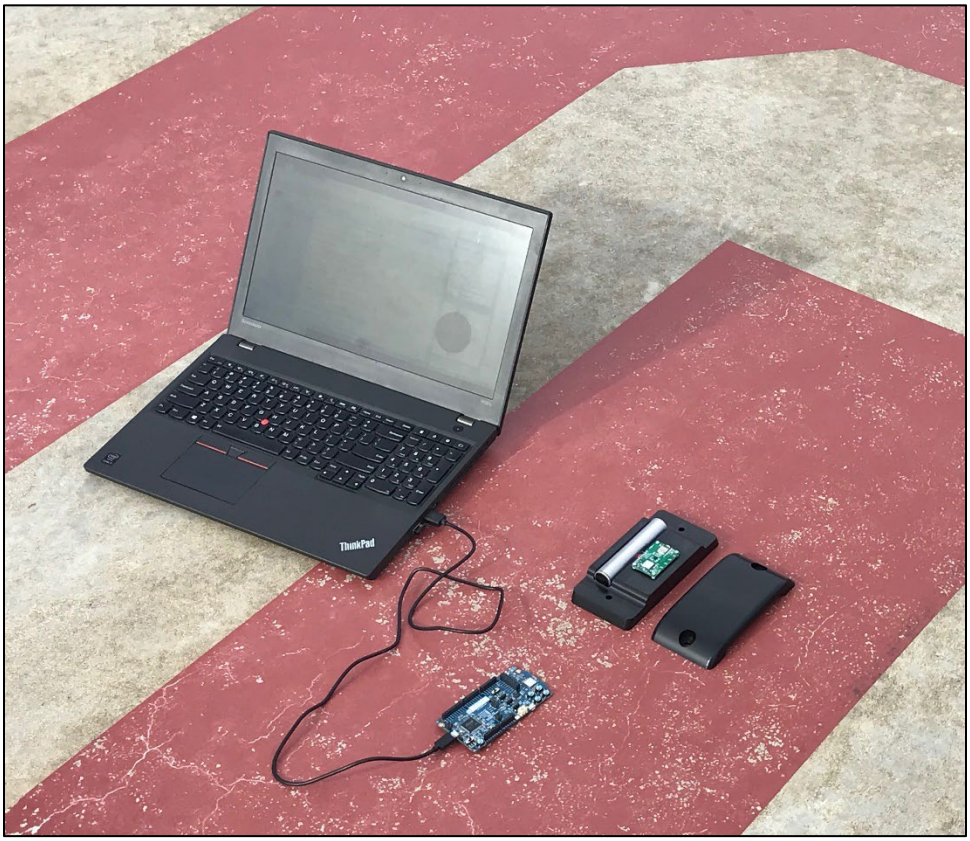

Figure 26. The new *i*VCCS 3rd generation system.

A microcontroller module (BMD-340) from Rigado based on the nRF52840 SoC from Nordic Semiconductor now manages the radio. This module allows for multiple wireless protocols (e.g., Bluetooth 5, IEEE 802.15.4 with ZigBee & Thread, and other proprietary protocols from Nordic Semiconductor). The following figure depicts the block diagram of the module.

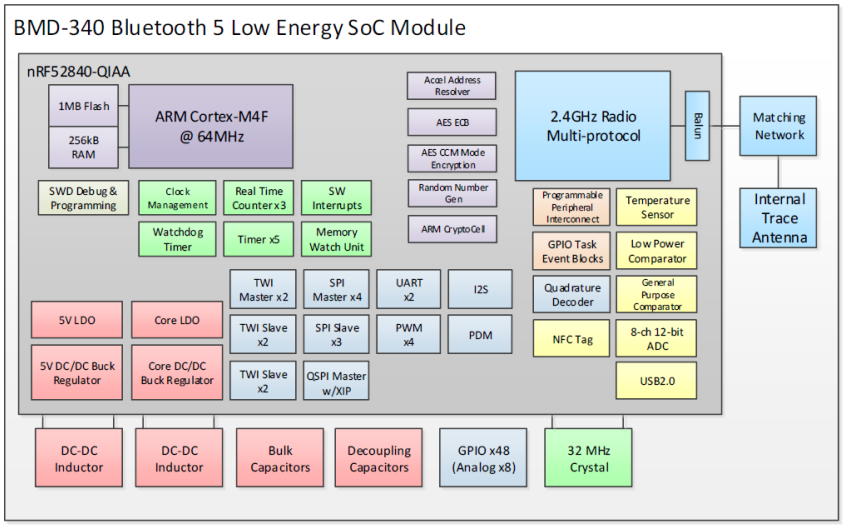

Figure 27. Block diagram of the nRF52840 SoC module.

The following table lists the 2.4 GHz transceivers integrated with module.

Table 1. Specifications of the 2.4 GHz transceiver

Specs	**Value**
Protocols	**BT 5 LE** Concurrent Peripheral / Central (S140)
Protocols	**Thread** (OpenThread-based)
Protocols	Zigbee
Frequency	2.360GHz to 2.500GHz
IEEE Standard 802.15.4 Modulation	**OQPSK** @ 250kbps
Bluetooth Low Energy 5 Modulation	**GFSK** @ 2Mbps, 1Mbps + 500kbps/125kbps coded
IEEE Standard 802.15.4 Receiver sensitivity	-100 dBm
Bluetooth Low Energy 5 Receiver sensitivity	**-96 dBm** (1Mbps), -103 dBm (125kbps coded)
Transmit power	+8dBm to -40 dBm
Receiver sensitivity	**-96** dBm (BLE mode)
RSSI	-20 to -90 dBm, 1 dB resolution
Antenna	Integrated antenna

While the BMD-340 module lacks a Real-Time Clock unit, the unit provides high accuracy counters and allows developers to implement their own calendar and time system. Counters are shared with the protocol stack of the wireless front-end; therefore, the implementation will drift a few seconds a day. Notably, the *i*VCCS speed estimation technique cannot tolerate such a time drift. Hence, a high accuracy, temperature compensated real-time clock was added to the system.

Table 2. Power consumption of the module

Specs	Value
Radio Tx	**13.6mA** @ +8dBm, **5.3mA** @ 0dB (DCDC, 3V)
Radio Rx	**5.4mA** @ 1 Mbps (DCDC, 3V)
CPU – Running	**52µA**/MHz running from flash, **3.3mA** @ 64MHz
CPU – Running	**44µA**/MHz running from RAM, **2.8mA** @ 64MHz
CPU – Idle	**1.5µA** in ON mode, no RAM retention, with RTC
CPU – Idle	**0.97µA** in ON mode, no RAM retention, all blocks IDLE
CPU – Idle	**0.4µA** in OFF mode, **+2nA** per 4kB RAM retention

In the following sections, the main features that differentiate the 3rdG from its predecessors are discussed.

3.1 Wireless Connectivity using Bluetooth Low Energy 5

The new sensor is designed around the nRF52480 module from Nordic Semiconductor, which features full Bluetooth 5 support in both hardware and software, including new long range, high throughput, advertising extensions, and improved coexistence features. Moreover, the software development kit provides support for Bluetooth mesh for building products and infrastructure employing mesh networking. The module has a new built-in radio interface that supports multiple protocols—in addition to Bluetooth 5, as well as 802.15.4-based protocols, such as Thread, ANT, and 2.4 GHz proprietary protocols.

Bluetooth 5 was chosen for *i*VCCS sensor design and application purposes, mainly due to high throughput when compared with previously used ZigBee. In addition, the Bluetooth protocol is ubiquitously available on many devices (i.e. smartphones, tablets, edge nodes), which eliminates the need to design a special access point (AP) to communicate with the sensor.

3.1.1 Bluetooth Low Energy 5 Basics

Attribute Profile (ATT)

Bluetooth Core specifications define the attribute profile as

"a protocol for discovering, reading, and writing attributes on a peer device."

Devices referred to as "servers" are used to expose a set of attributes and their associated values to peer devices (or "clients"). A client can inquire and discover attributes exposed by the server. Attributes can be read and written by clients and unsolicitedly sent by the server.

Generic Attribute Profile (GATT)

According to Bluetooth Core specifications, the Generic Attribute Profile (GATT) is defined as

"The Generic Attribute Profile (GATT) defines a service framework using the Attribute Protocol. This framework defines procedures and formats of services and their characteristics."

This means that GATT defines how data are packaged, presented, and transferred using BLE.

Services

"A service is a collection of data and associated behaviors to accomplish a particular function or feature."

In other words, a service is a group of information that is relevant to a particular type. For example, a device might have a service for sensor readings, and another service to report the status of battery charge. The Bluetooth SIG defines a number of standard services (e.g. blood pressure service, heart rate service, device information service).

Characteristics

"A characteristic is a value used in a service along with properties and configuration information about how the value is accessed and information about how the value is displayed or represented."

This implies that the characteristic is the actual value and information of a service. A characteristic definition contains a descriptor that communicates the value and configuration of the server (e.g., security and access parameters, and units).

3.1.2 Object Transfer Service

The Object Transfer Profile (OTP) defines the required set of rules and characteristics of an Object Transfer Service (OTS), which enables a client to create, modify, read, and write objects on a particular server. OTP allows for bulk data transfer over Bluetooth Low Energy (BLE) transport layer, which is compatible with Bluetooth v4.1 or later. OTP Bluetooth Profile Specification defines two roles, as follows.

"The Object Server is the device that interacts with an Object Client for the reading or writing of object data and exposes object metadata. The Object Client is the device that initiates reading, writing or other object action from the Object Server."

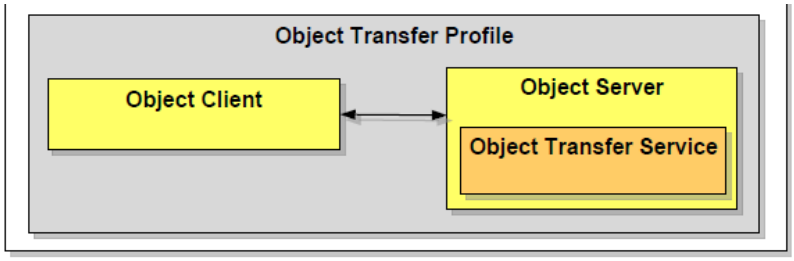

Figure 28 Diagram that explains the relationship between service and profile roles.

This service could be employed in *i*VCCS design to allow for magnetic signature transfer capability, where *i*VCCS sensor would act as server and an AP (e.g., smartphone, tablet, laptop) would act as client.

A reference design has been altered to resemble an *i*VCCS scenario for confirming feasibility of implementing the service and accomplishing required use case.

3.1.3 UART over BLE using Nordic UART Service (NUS)

In the previous revision, *i*VCCS 2ndG featured a wireless command-line interface (CLI) that permitted the AP to interact with the sensor to read, write, and configure different parameters on the sensor. Traditionally, Bluetooth Classic supported a Serial Port Profile (SPP) that was based on the RFCOMM protocol. SPP emulates a serial cable to provide a substitute for existing RS-232. Unfortunately, no standard serial profile is defined in BLE specifications, although the specifications allow for custom services implementations. Nordic Semiconductor provides a custom UART service called “Nordic UART Service (NUS)” for this purpose. Since this is a custom implementation, not all Bluetooth devices will recognize the service, and therefore, a custom software on the APs must be present to interact with the sensor.

Nordic Semiconductor also provides sample Android and iOS applications that could be used to communicate with connected Bluetooth servers that expose the NUS service. Future *i*VCCS system designs will incorporate a custom smartphone and tablet application.

The NUS exposes two characteristics: one for transmitting and one for receiving, as seen from the peer. Regarding RX characteristic, the peer can send data to the device by writing to the RX Characteristic of the service. For TX characteristic, given that the peer has enabled notifications for the TX Characteristic, the application can send data to the peer as notifications.

In the reference design from the SDK, data transmitted over BLE is routed to UART interface. This design was modified to suit *i*VCCS needs. Instead of a UART handler, a function removes data from the BLE packet and routes it to an application data handler for implementing the user application.

```
/**@snippet [Handling the data received over BLE] */
static void nus_data_handler(ble_nus_evt_t* p_evt) {
  if (p_evt->type == BLE_NUS_EVT_RX_DATA) {
    NRFX_LOG_DEBUG("Received data from BLE NUS.");
    app_data_handler((char*)p_evt->params.rx_data.p_data, p_evt->params.rx_data.length);
  }
}
```

The function `app_data_handler` is user implementation dependent, wherein command handling is implemented. An example function is given below.

```
void ble_data_handler(char* p_data, uint16_t length)
{
    // Make sure that the terminal null is there
    char* data = 0;
    if (p_data[length-1] != '\0')
    {
        data = malloc((length + 1) * sizeof(char));
        strncpy(data, p_data, length);
        data[length] = '\0';
    }
    else
        data = p_data;

    NRFX_LOG_INFO("%s", (uint32_t)data);

    bluetooth_send("Got your data, you said:\n");
    bluetooth_send(data);

    free(data);
}
```

Sending data to the AP is made easy with a function that provides all necessary preparations and calls to send a packet over BLE to the connected peer.

```
/**@brief Function for sending data over BLE to the connected peer.
 */
ret_code_t bluetooth_send(char* data)
{
   ret_code_t err_code = NRF_SUCCESS;
   uint16_t length = strlen(data);

   err_code = ble_nus_data_send(&m_nus, data, &length, m_conn_handle);
   if ((err_code != NRF_ERROR_INVALID_STATE) &&
      (err_code != NRF_ERROR_RESOURCES) &&
      (err_code != NRF_ERROR_NOT_FOUND))
   {
      APP_ERROR_CHECK(err_code);
   }
   return err_code;
}
```

3.1.4 Bluetooth Stack Initialization

Before all aforementioned functions can be used in an application, the Bluetooth stack should first be initialized. Additionally, several configurations should be made to GATT, Generic Access Profile (GAP), and other required services running on the device. The Nordic SoftDevice Bluetooth stack is a precompiled binary file loaded into a certain location in Flash with reserved RAM space. It is important to know that the sensor's main application and SoftDevice should never overlap in Flash or in RAM address spaces. Special care must be taken when designating these values to prevent unpredicted behavior during deployment. In out design, the linker script has been modified to account for such considerations.

Functionality related to Bluetooth connectivity has been bundled in a single library containing all related initialization and configuration functions. When a packet arrives, user needs only to call a general initialization function and pass the data handler to be called.

```
/**@brief Function for initializing everything related to Bluetooth to
 *       get it ready.
 */
void bluetooth_init(bluetooth_rx_data_handler_t data_handler)
{
   app_data_handler = data_handler;
   // Initialize app_timer
   ret_code_t err_code = app_timer_init();
   APP_ERROR_CHECK(err_code);
   ble_stack_init();
   gap_params_init();
   gatt_init();
   services_init();
   advertising_init();
   conn_params_init();

   NRFX_LOG_INFO("Bluetooth is initialized successfully.");
}
```

Such programming ensures that the right functions are called in the right order and also sets up advertising packets, which determines how the device is seen from the peer point of view. User can then call another function to commence the advertising procedure. The totality of code can be summarized in two lines in the main source file:

```
/* ==== Initialize SoftDevice and start Bluetooth ==== */
   bluetooth_init(ble_data_handler);
   bluetooth_advertising_start();
```

3.1.5 BLE 5 Long-Range Connectivity

The newly released BLE 5 standard introduces a new physical layer that features a robust modulation scheme and forward error correction (FEC) mechanism. Although this increases the immunity of transmitted packets against noise, it decreases achieved data rate. However, the result is an improved range of communication without sacrificing more power, thus the name BLE Long-Range. Notably, BLE

devices that advertise utilizing this new physical layer cannot communicate with legacy devices, and coordination between peers is required to avoid communication disruption, which can be managed by the stack provided in the host system.

3.2 Over-the-Air Device Firmware Upgrade (OTA DFU)

Given the intended operation environment and the difficulty of installing the sensor on the ground, this feature renders crucial, as it allows the system to update its firmware components on the field without interrupting its operation.

Over-the-air (OTA) device firmware upgrade (DFU) is realized by including a custom bootloader to manage this functionality. A bootloader is a small and special piece of firmware that resides at a defined address in the microcontroller memory. Usually, its sole purpose is loading the application firmware address into the memory for executing user code. The bootloader is responsible for:

- booting into an application,
- activating new firmware,
- optionally, entering DFU mode where DFU transports are activated and new firmware can be delivered, and
- feeding the watchdog timer.

Figure 29 shows the placement of the bootloader, application, and other settings in the system's memory.

The bootloader settings page is considered an interesting component in that it is a non-volatile page that stores DFU information, including

- current firmware size and CRC-32 code,
- new firmware size and CRC-32 code,
- current firmware versions (application and bootloader),
- progress of the firmware update,
- progress of the firmware activation, and
- transport layer-specific data.

DFU procedures are carried out by copying the new firmware to a temporary location in memory, and then exchanging it with old firmware after its checksum, digital signature, and requisites are validated. After the application signals its desire for DFU, the bootloader takes responsibility for the process. When the bootloader restarts, rather than loading the application address, it broadcasts a custom service for firmware upgrade on the BLE advertising channels and waits for the user to connect and send the new firmware package. If no connection occurs within two minutes, the bootloader stops advertising and reloads the old firmware application. Figure 30 shows a flow diagram of the process.

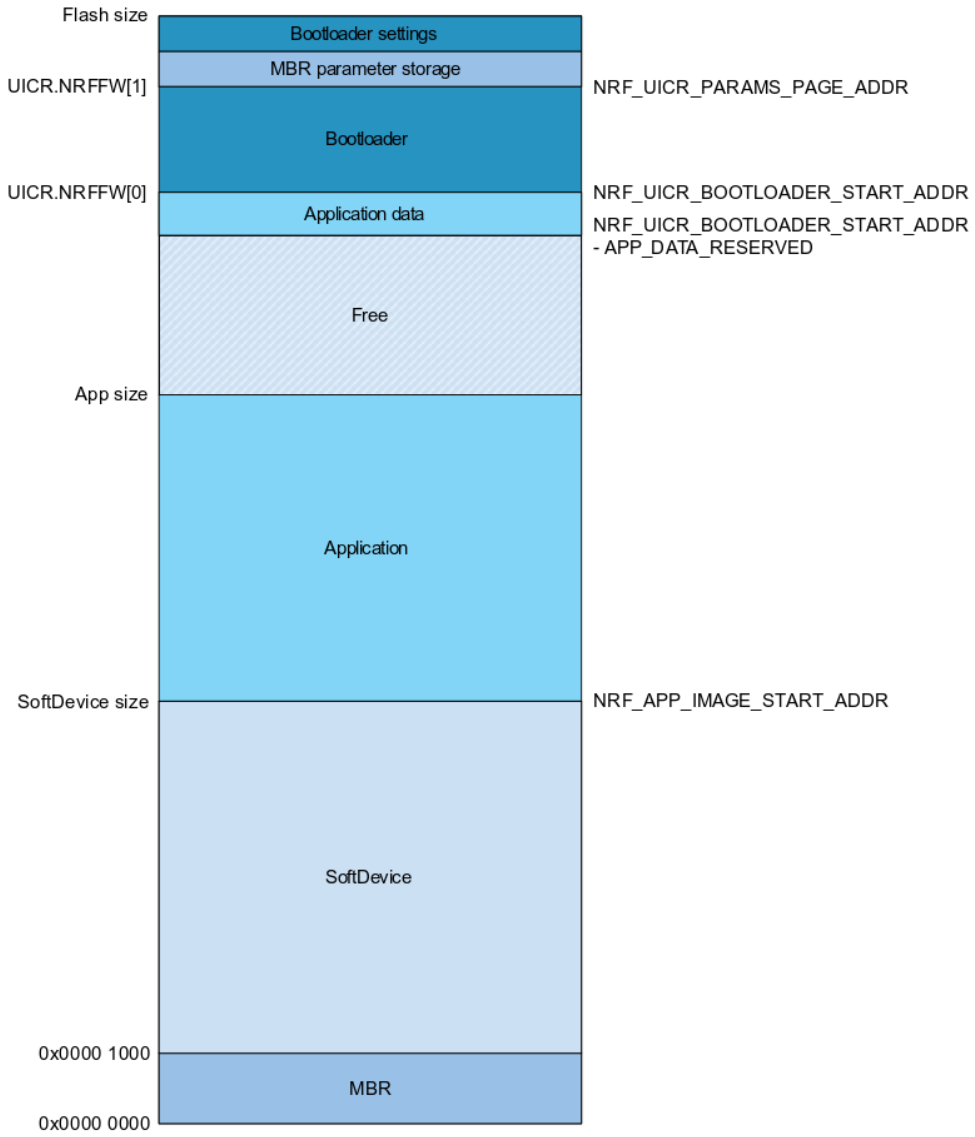

Figure 29. Memory layout of *i*VCCS with custom bootloader.

The DFU feature updates the following in a single package:

- application
- SoftDevice (i.e., BLE stack)
- bootloader
- bootloader and SoftDevice

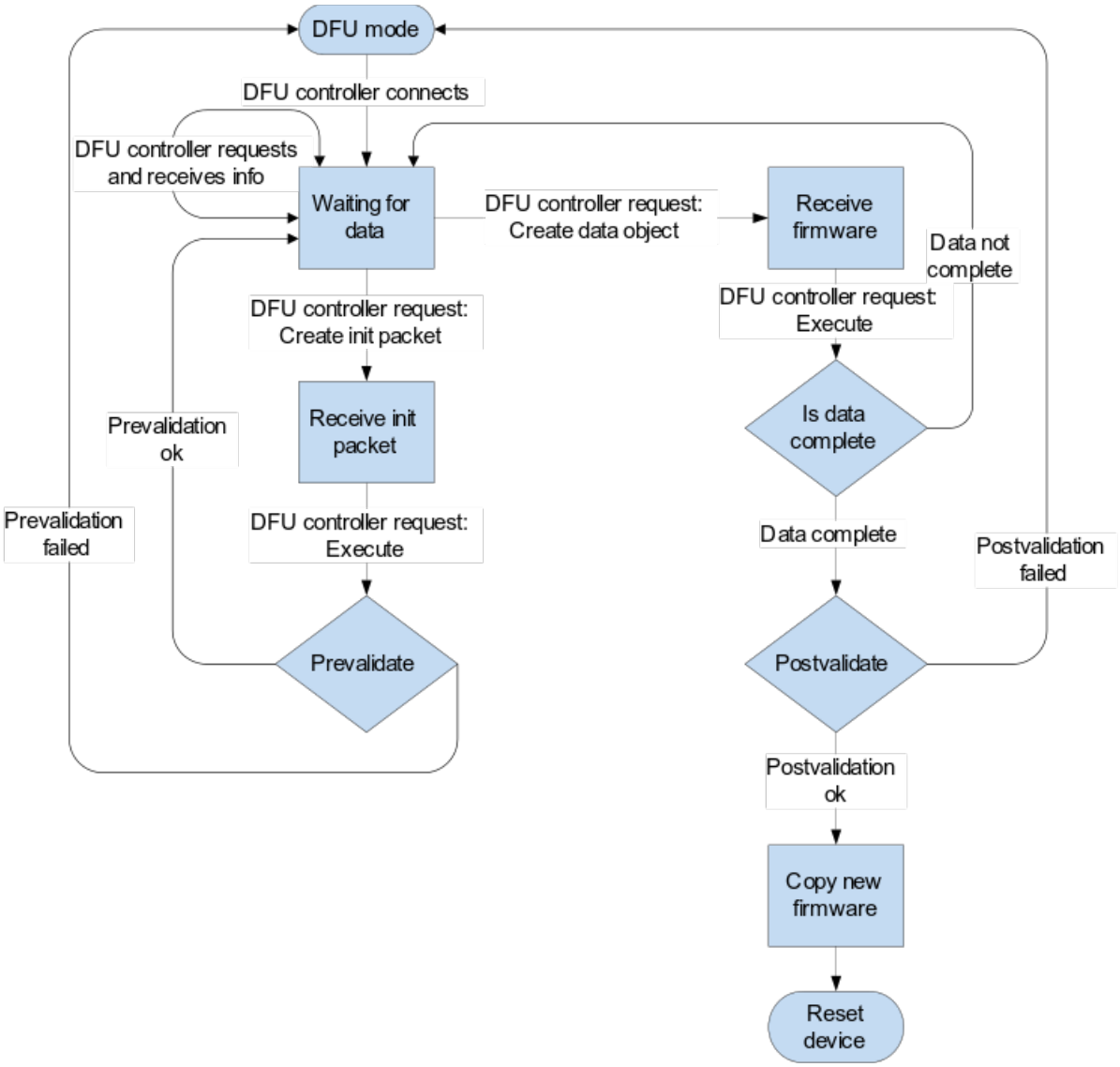

Figure 30. Process flow of DFU procedure.

3.3 Low Power Operation

Low power operation is achieved via: 1) supporting low-power modes by the host microcontroller (nRF52480), 2) optimizing clock and power management components, and 3) reducing current consumed by the Bluetooth and Radio interface. Enhancements for each category (i.e., system low-power modes, clock and power management, and BLE 5 connectivity) are listed below.

A. System Low-Power Modes

- nRF52840 has two power saving modes: SYSTEM_ON and SYSTEM_OFF.
- SYSTEM_OFF mode offers the deepest power saving mode, wherein core functionality and all tasks are powered down. The system can be woken up through a configured digital or analog pin, which effectively restarts the system.

- SYSTEM_OFF is implemented as an option, reducing power to 590 µA. Since a vehicle waking the system from SYSTEM_OFF will not be counted, this power mode could be used if the system remains idle for a certain amount of time.
- SYSTEM_ON mode has two sub modes: low-power and constant latency.
    - In constant latency mode, system wakeup is minimized by leaving on only a certain number of resources during sleep.
    - In low-power mode, the power management unit (PMU) dynamically optimizes the power supply to different system components according to need and pending request, although doing so comes at the cost of varying CPU wakeup latency.
- By incorporating SYSTEM_ON low-power mode into the sensor system, current consumption is reduced from ~ 8 mA to approximately 1.8 mA.
- It is worth noting that in SYSTEM_ON BLE interface is turned on and able to send and receive data packets. In SYSTEM_OFF, BLE interface and other system peripherals are powered down, requiring the entire system to reset when awoken.

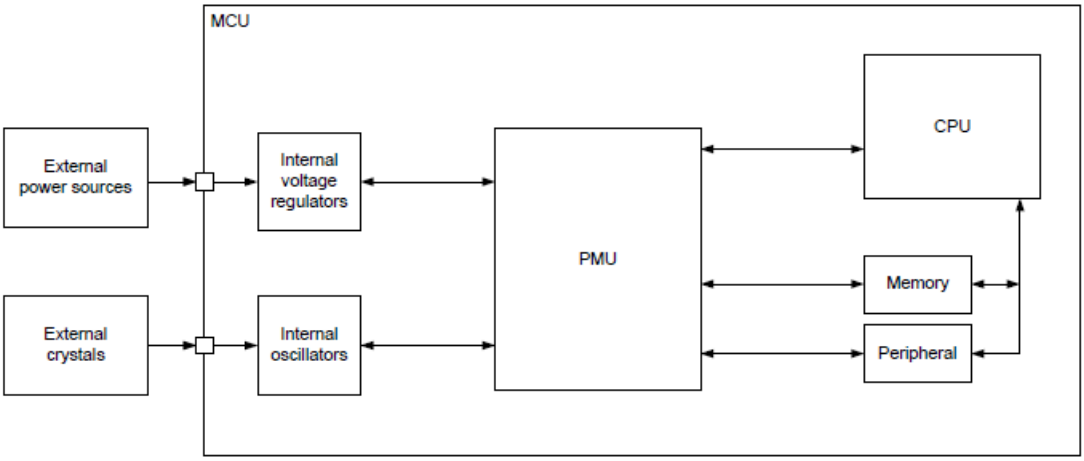

Figure 31. nRF52840 power management unit (PMU).

B. Clock and Power Management

- nRF52840 SoC encompasses two main supply regulator stages: REG0 and REG1.
- Each regulator stage has two options: DC/DC buck converter and low-dropout regulator (LDO). DC/DC regulators are enabled.
- Power consumption with SYSTEM_ON low-power and DC/DC converters is reduced to 1.27 mA in idle state.
- Enable external Low-Frequency Crystal Oscillator (LFXO), which offers better accuracy and lower power consumption.
- QSPI peripheral consumes approximately 500 µA. Functions to initialize and turn off QSPI peripheral are implemented. Flash and QSPI should only be needed when buffering data from RAM to flash array unit.
- Idle state current consumption with SYSTEM_ON, DC/DC converter, BLE connected to a central peer (AP), and QSPI off is 750 µA. This is further reduced to 605 µA when BLE stack and radio are powered down.

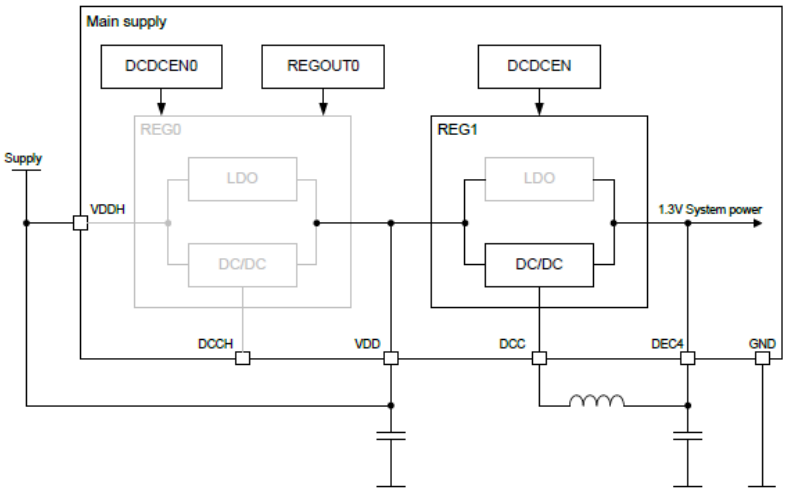

Figure 32. nRF52840 REG1 stage configured as DC/DC Buck converter.

C. BLE 5 Connectivity

- System power consumption is reduced by means of duty cycling BLE advertising.
- BLE advertising parameters were adjusted to the following: ADV Interval = 500 ms and ADV timeout = 120 s. These settings correspond to two advertising packets every second—240 packets in total. Subsequently, advertising times out and transitions into idle state.
- The previous adjustment reduced average current consumed in advertising state to 23 µA from 235 µA (according to Nordic's online power profiler tool).
- After advertising timeout (e.g., 2 minutes), BLE stack stops and radio interface is shutdown, conserving approximately 90 µA.
- A timer is implemented to reinitialize BLE stack and restart advertising after 10 minutes in idle state.
- Total measured averaged current consumed by the sensor is 695 µA while advertising and 605 µA without advertising.

After the aforementioned enhancements were implemented in the sensor firmware, current measurements were collected for 25 minutes without connecting to a peer BLE device. Figure 33 shows measurements over time. In idle state (i.e., when the sensor is not advertising), consumed current is approximately 602 µA. In advertising state, a maximum current of 753 µA was observed. Given that a vehicle approaches the sensor and is detected, the sensor draws approximately 3.75 mA, as a result of the CPU requesting the high frequency clock that drives the main application.

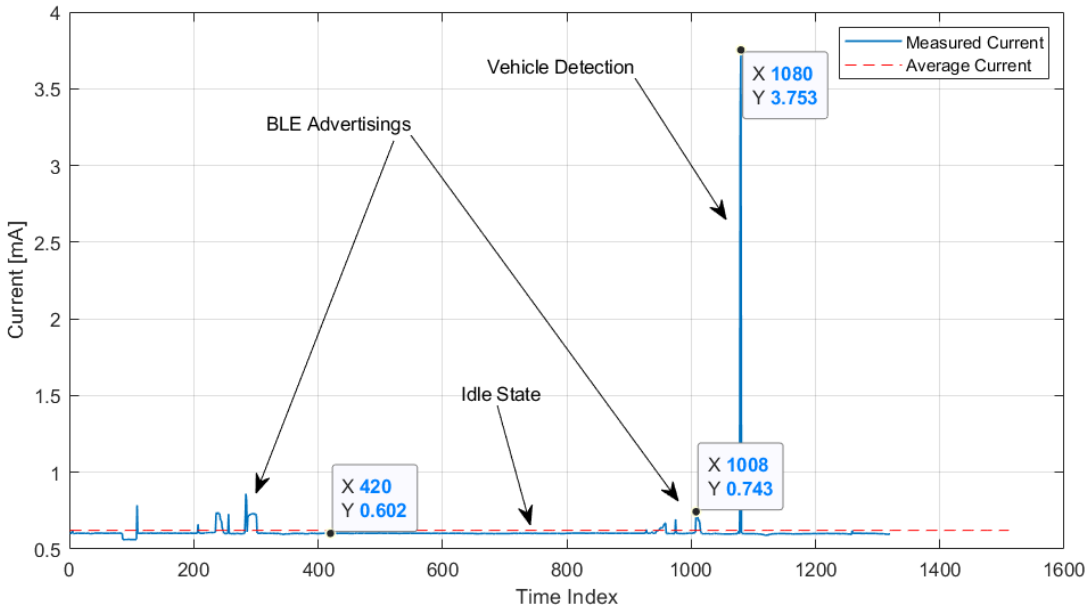

Figure 33. Electrical current measurements for *i*VCCS 3G.

On average, the sensor drew 621 µA over 25 minutes. Assuming the sensor is to be powered with 13 Ah batteries, battery life is two years. Notably, GPS, Flash memory array, and real-time clock operations are not taken into consideration.

3.4 Advanced Battery Gauge

The new sensor design features a 10 Ah Lithium Thionyl Chloride (Li-SOCl2) battery, which requires a special gauging unit due to its unique chemistry. Texas Instrument bq35100 was chosen for this purpose.

The bq35100 battery fuel gauge provides a highly configurable solution for non-rechargeable lithium batteries without requiring forced discharge. The module uses voltage, current, and temperature data to provide State-of-Health (SOH) and End-of-Service (EOS) information.

When first installing a new battery (and before the module gauges the new battery and reports to the host system), the gauge must first be prepared. This includes setting up the gauge configuration, gauging mode, calibration, and battery parameters (e.g., chemistry, cell voltage, capacity).

Sensor firmware supports commands to simplify interacting with the gauge, including dynamic commands to access internal registers of the gauge, such that the sensor acts as a transparent router between AP (user) and fuel gauge.

With end-of-service (EOS) mode, the battery gauge module performs only a health check on the installed battery. Given the chemistry of used Lithium Thionyl batteries, the Texas Instrument bq31500 battery gauge could only report whether or not the unit reached EOS. However, the goal is quantifying the lifetime of the power unit and providing estimated battery life. Hence, another mechanism was introduced in *i*VCCS 3rdG. Because this solution continually “counts” the coulomb energy discharged from the battery once a start command is sent to the gauge, the counter requires that the gauge is

powered all the time. The gauge will save accumulated value on its internal flash storage after receiving a stop command.

One challenge to this solution is that the internal flash has a limited lifetime—precisely 100,000 write cycles, per the datasheet reports, before wearing out/becoming inoperative. To protect the flash from excessive writing commands and to prevent loss of accumulated coulombs values, given a sudden restart (e.g., watchdog timer trigger), a repetitive procedure has been developed to send the required sequence of commands to write values on the flash every 30 minutes.

The firmware validates that the battery gauge is configured to work in the accumulated coulomb mode upon its start and commands the gauge to commence counting as soon as the sensor is powered on. An interrupt-driven timer also commences with gauge start-up to ensure values are stored on flash every 30 minutes.

Although the battery gauge and Real-Time Clock (RTC) use the Twisted Wire Interface (TWI) engine of the microcontroller, they are connected on two different sets of physical pins. When one of the two peripherals have access to the TWI bus, the other does not. This is accomplished by alternating the configuration when the battery gauge is enabled/disabled, as the gauge is only required when an EOS check is requested. As part of the new implementation, however, this functionality has changed, given the new requirement for constant power to the battery gauge. Hence, instead of alternating the TWI bus between the two peripherals on gauge enable/disable, the system is now reconfigured whenever the gauge attempts to access the bus—regardless of gauge power state. For example, if an indication of the remaining battery capacity is requested, the system transfers the TWI bus from the RTC to the battery gauge, and a series of transactions are carried out on the TWI bus, before the RTC assumes the bus back when the gauge is done. Several commands are available to report different parameters on the battery status.

3.5 NOR Flash Array

An array of four serial NOR flash units were equipped with the sensor's main processor to store data during field deployments. Each unit has 8 MB of data space, for a total of 32 MB. All four units are connected to the host system via Quad-SPI interface, which provides at least twice the transfer rate of regular SPI protocol. Power to each unit is controlled by a low-current load switch. The host system can select which unit to write to or read data from by enabling the corresponding power switch and selecting its chip select pin (CS).

The same API used in *i*VCCS 2nd G was ported to *iVCCS* 3rd G with the addition of a control function that coordinates to which unit the data should be written. Moreover, as a result of the page-basis write functionality exhibited by memories, data buffering should occur in the host system prior to writing to Flash. This process is similar to *i*VCCS 2nd G.

Memory addresses of four units are combined to form a global address space ranging from 0x00 to 0x2000000. The algorithm calculates which unit a given global address belongs to and to what local address it translates. The two equations below are used to calculate this information:

$$Flash\ Unit = floor\left(\frac{Global\ Address}{Flash\ Unit\ Size}\right) + 1$$
$$= floor\left(\frac{Global\ Address}{0x800000}\right) + 1$$

$$Local\ Address = Global\ Address\ \%\ Flash\ Unit\ Size$$
$$= Global\ Address\ \%\ 0x800000,$$

where $floor(x)$ is a function to round a real number x downward, effectively returning the largest integral value not greater than x, and $\%$ is the modulus operator.

After the unit is determined, the host system enables power upon request for a Read or Write operation. Power is disabled afterwards to conserve energy.

By default, when the system starts the global address is reset to 0x00. However, in the event of error(s) and fault(s), the system might not be able to recover without a full system restart. This would cause data overwrite when deployed in the field since the global address was reset to 0x00. To prevent data overwrite each time the system starts, the flash address space looks for the first empty page and sets the global address to that location. The algorithm progresses through the first four bytes of each page and switches to the corresponding flash unit as it scans through addresses.

3.6 Miscellaneous Features

3.6.1 Watchdog Timer (WDT)

A countdown watchdog timer using the low-frequency clock source (LFCLK) offers configurable and robust protection against application lock-up.

The watchdog is implemented as a down-counter that generates a TIMEOUT event when wrapped after counting down to 0. When the watchdog timer commences, the counter loads with the value specified in the configuration registers. The counter is reloaded given that a request is granted.

The watchdog timeout period is given by:

$$timeout\ [s] = \frac{CRV + 1}{32768}$$

where CRV is a configuration register that maintains reload value.

The watchdog is reset by a call to function `wdt_feed()`, which is called in several places in the source code (e.g., main application loop, wait routines, while loops pending flags, initialization routines). Because routines wherein WDT is reset are guaranteed to timeout and return, it is acceptable to reset the WDT in them. However, when the CPU reaches a point where it is stuck in a dead loop, e.g., hard faults not handled by the firmware, the WDT expires and issues a system reset ensuring reliable recovery.

3.6.2 Command-Line Interface (CLI)

To facilitate user interaction with the sensor, a command-line interface (CLI) has been introduced in the embedded system. Each command has a description for explaining the purpose of the command and the way in which it is executed with any sub commands and options.

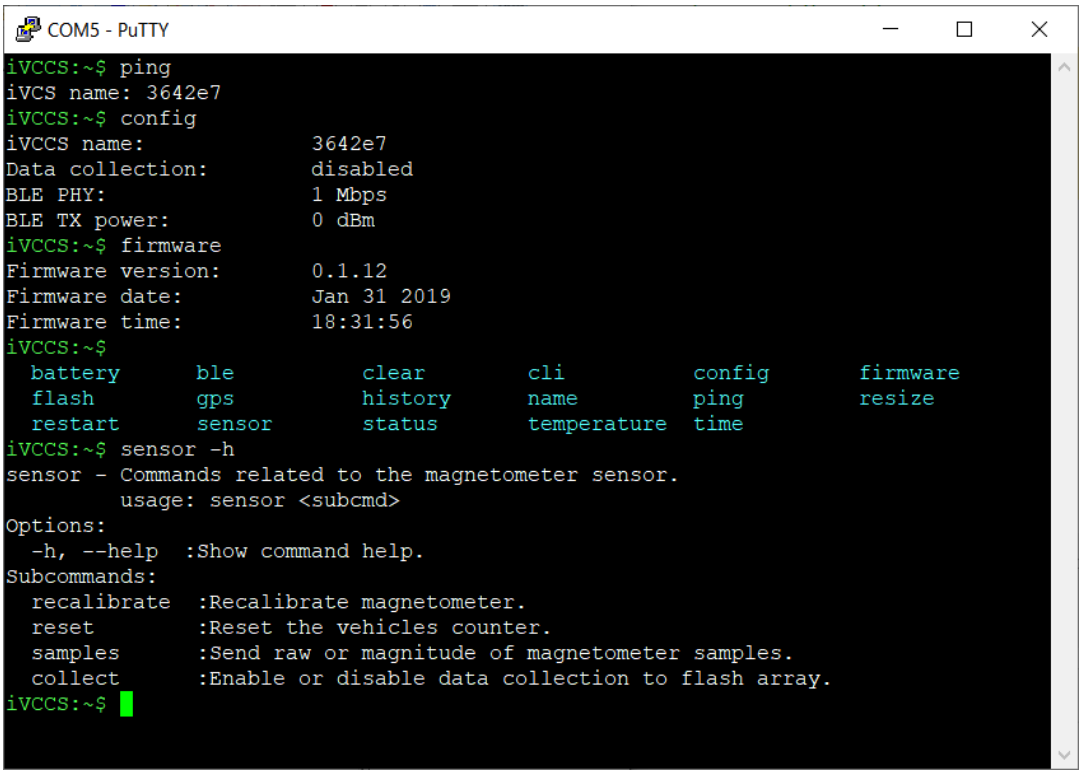

Figure 34. Command Line terminal used to interact with the sensor.

3.6.3 Sensor Configurations Module

Some configurations might be persistent with power loss. Like resolving automatic address resolve was developed in the Flash storage, it is advisable to save some settings in non-volatile memory and load them at system boot up. Therefore, a new module was added to the firmware where settings and configurations can be defined, stored and loaded into RAM whenever required. The configuration module uses the on-chip flash storage to save different variables such as BLE TX power, BLE PHY mode, sensor node name, and data collection setting. This on-chip flash storage is the same one used to hold firmware code in. Two pages, each is 1024 bytes, were reserved for the configuration module. This flash has a limited number of write/erase cycles, and should not be exceeded, otherwise, flash cells start to wear out. To protect the flash from a possible wear damage, an automatic garbage collection is used when writing data.

3.6.4 Wireless Data Retrieval

Data collected by the sensor are stored in a serial flash array on-board. The ability to retrieve the data and download it to a host system, e.g., PC, requires a technique that the sensor should support in order to access that data. For this purpose, a download command was introduced in the sensor firmware and exposed to the user through the command-line interface (CLI).

The flash array is interfaced to the sensor's microcontroller via a Quad-SPI protocol. Therefore, reading from any flash unit should occur in multiple of 4 bytes. When a read operation occurs, the flash unit from which data is read must be specified, and of course, starting address. When data is stored across

multiple units, it becomes cumbersome to keep track of addresses and units in this case. Flash API facilitates a `read()` function for this purpose. This function takes a global address and number of bytes to read. However, forwarding data from flash units to CLI (terminal), requires memory to be allocated in RAM. When data requested is too big to fit in RAM as one chunk, i.e., ~ 10 KB, the download operation fails. In this case, the reading process should be divided into small chunks and data should be forwarded to the CLI recursively and automatically until all data requested by the user are read and transmitted to CLI.

3.7 Remote Access and Control

The introduced wireless capabilities made it easier to integrate the sensor system with the cloud and control from a remote host machine. Such scenario would allow for remote configuration, as well as data download from the field.

REECE (Roadside Embedded Extensible Computing Equipment) is a Linux box designed to store and process sensor data connected via different types of interfaces (e.g., ADC, USB, RS232), as in Figure 35. REECE controls communication between sensors and the cloud using a WebSocket connection to facilitate real-time sensing and environment monitoring. Alternatively, the REECE could be used as an intermediate host that can be accessed through Telnet or SSH session over an internet gateway using an LTE modem, for example.

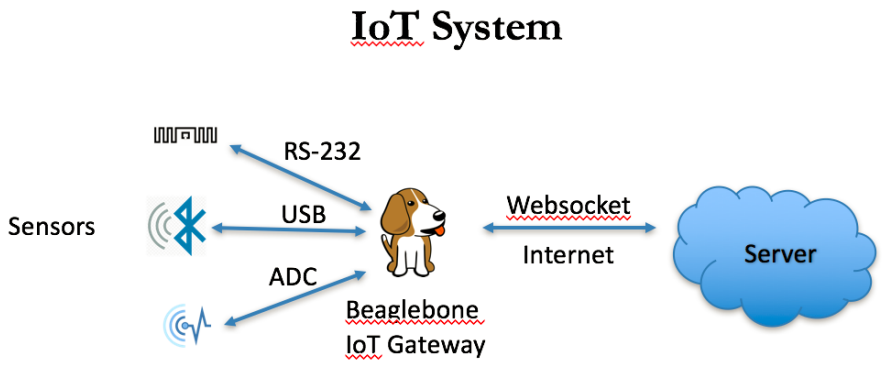

Figure 35 IoT system architecture.

The REECE is connected to an iVCCS access point (AP) through USB. When sensors are advertising, the AP sends a connection request to all advertising iVCCS nodes. Figure 36 shows the setup. The AP can be reached remotely now using the IP address of the gateway (RV50 modem) by starting an SSH session with the Linux system on the REECE and opening the serial port of the AP.

Figure 36 Remote access setup. RECCE is connected to RV50 and iVCCS AP. Sensors are connected to AP over BLE.

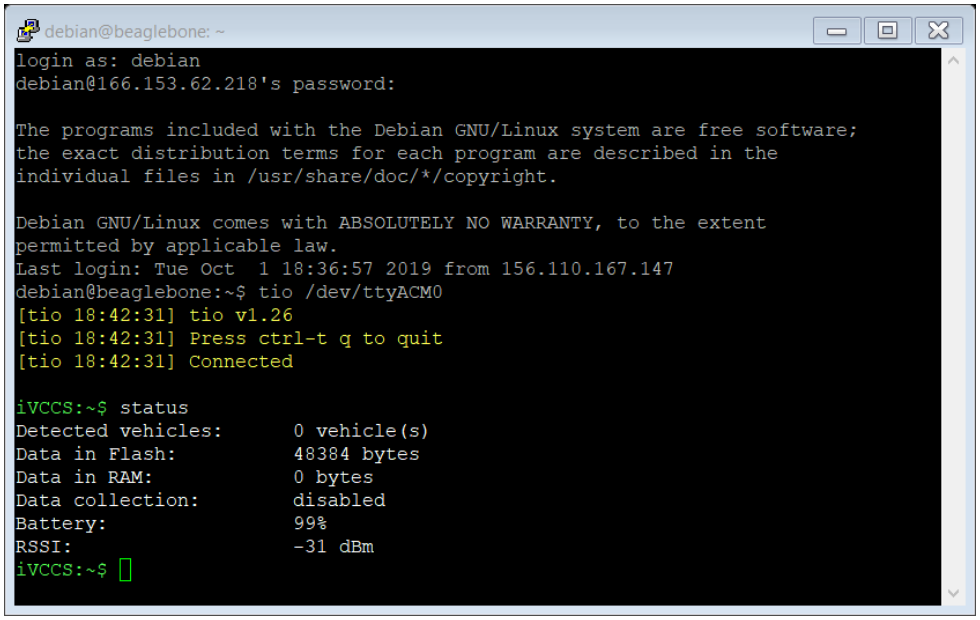

Figure 37 iVCCS sensors accessed remotely through REECE.

A server can connect to a deployed AP and inquire all connected sensors for data, e.g., timestamps or vehicles counts, as well as sensors status. Alternatively, the REECE can be programmed to periodically pull data from sensors and then upload it to server over a WebSocket.

The AP can be placed in a cabinet with the REECE and an external antenna can be connected to the AP, which can also provide additional signal-to-noise (SNR) gain and better link performance. An example of such configuration is discussed in section 3.8.3 and Figure 46.

3.8 Deployments and Results

*i*VCCS 3rd G was tested and deployed in various settings to test and debug the firmware and to evaluate system-level performance. Location, setup, and results of each deployment are discussed below.

3.8.1 On-Campus Test

The complete system—including batteries and new enclosure—was tested continuously for three days on the OU-Tulsa campus. Operations were as expected, with advertising every 12 minutes. This time frame allowed APs to pull information on vehicle counter and data storage status, and to interact with CLI. Communication with the sensor was also tested using a smartphone utilizing a Nordic app for evaluation. The setup is shown in Figure 38.

Figure 38. Sensor placement on the road.

Even though it rained during the test, the sensor experienced no interruptions and operated as expected. The sensor collected more than 2000 vehicle timestamps over the course of three days from vehicles entering from the south entrance of the OU-Tulsa campus. Data was downloaded through the CLI. A snippet of the downloaded data file is shown in Figure 39.

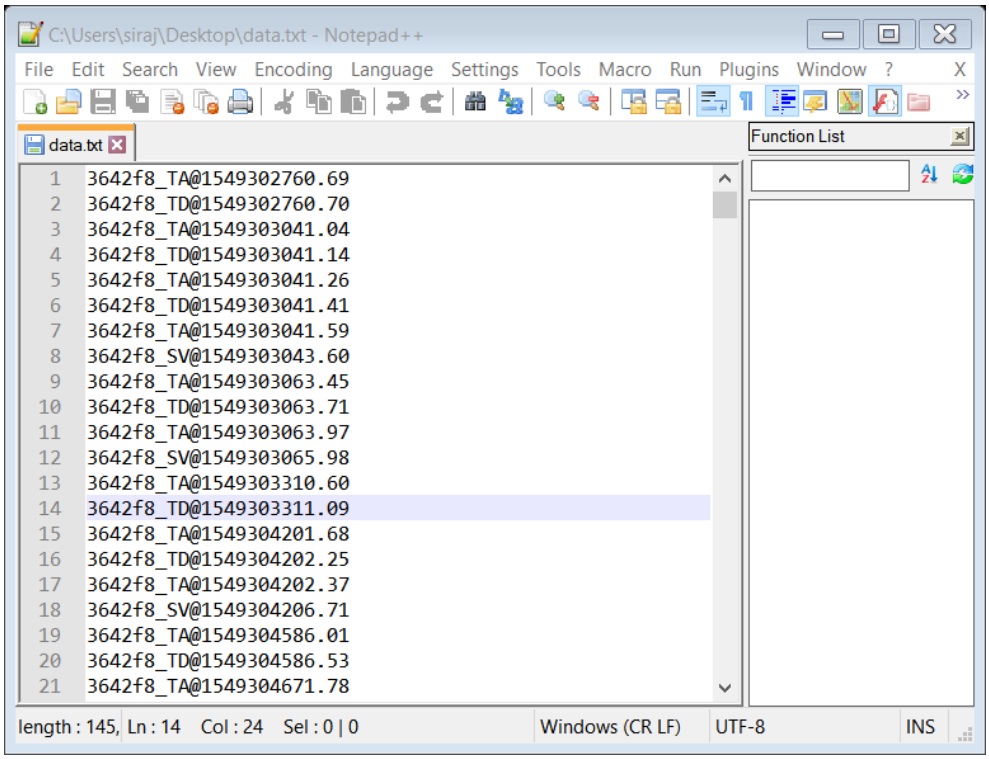

Figure 39. Snippet of the downloaded data from campus test.

The purpose of this test was to ensure acceptable sensor operation on a long-term deployment and to detect faults that might arise. It is important to note that detection accuracy and speed estimation of approaching vehicles were beyond the scope of this test; this work was left for highway deployments where ground truth data is available via other systems (e.g., Road Runner Kit 3, AVC systems, or cameras).

3.8.2 First Deployment on I-44

The system was tested on I-44 for three days from April 24 to 26, 2019. Sensors were collected from the field on April 29. Some sensors were still operating and collecting data, while others ceased working due to problems discussed later in this report. Figure 40 and Figure 41 show system placement and setup on I-44.

The sensors collected on average over 8,000 vehicles per day. Road-side cameras were setup to validate collected data collected. However, only three hours of video were recorded on any given day. Despite the fact that some sensors experienced reset events due to poor wiring inside the enclosure (which was resolved in subsequent deployments) and shocks from vehicles driving over the enclosure, the system rebooted and loaded predefined configurations successfully, ensuring uninterrupted data collection.

Problems during deployment affected data quality. Notably, new sensors must perform flash initialization prior to data collection deployment by sending a chip erase command through the sensor CLI. Sensors 2 and 4 failed to record data on flash memory due to a glitch in this array initialization, which prevented data storage. Both sensors were successfully used to wirelessly collect data on the second and third day during the three hours video was recorded.

Figure 40. Placement of sensors on I-44 and 126th St in Tulsa, OK.

Figure 41. Laptop used to configure and communicate with two sensors in lane, through a directional antenna connected to an access point on the road side.

Although sensors endured pressure from heavy trucks passing over the newly fabricated enclosures, small sensor pocket space caused GPS module RF shields to compress. Consequently, GPS performance was heavily degraded (i.e., sensors detected and collected vehicles timestamps, but data was out of sync since the GPS could not lock to new constellation). This issue was reported to NCTronics, the company

that designed and manufactured the enclosure. The company has since modified the design to mitigate the problem. A new, waterproof unit has also been designed.

Given the problems discussed above, most collected data could not be validated with ground truth reference.

One sensor was missing from the field when they were collected on April 29, 2019. Evidence suggested that someone broke the enclosure and took the senor (See Figure 39).

Figure 42. Broken enclosure as found on I-44 on Monday April 29, 2019.

3.8.3 Second Deployment on US-412

For the second deployment, iVCCS 3rd generation was installed on US-412 West for three and a half hours on June 11, 2019. The system was tested for vehicle detection accuracy, as well as real-time vehicle classification based on vehicle length. The system was also integrated with Bluetooth Travel Time system to assign classified vehicles with Bluetooth MAC addresses.

Three sensors were installed on the ground with a separation distance of 4.5 meters. A camera was also installed roadside to validate collected data, as depicted in Figure 43.

Several problems from previous deployments were avoided.

1. Instead of using electrical tape, sensors were placed into soldered enclosures with shorter power cables to prevent power loss and minimize the chance of resetting the system.
2. Sensors were fixed inside the enclosures without the plastic screw-guard legs, giving the sensor room to adjust when vehicles drive over the enclosure and bend the cover. See Figure 44. Readers will recall that a too small enclosure led to damaged GPS modules.

Also, leveraging firmware version 1.0.0 in this deployment fixed the problem of writing data to flash storage.

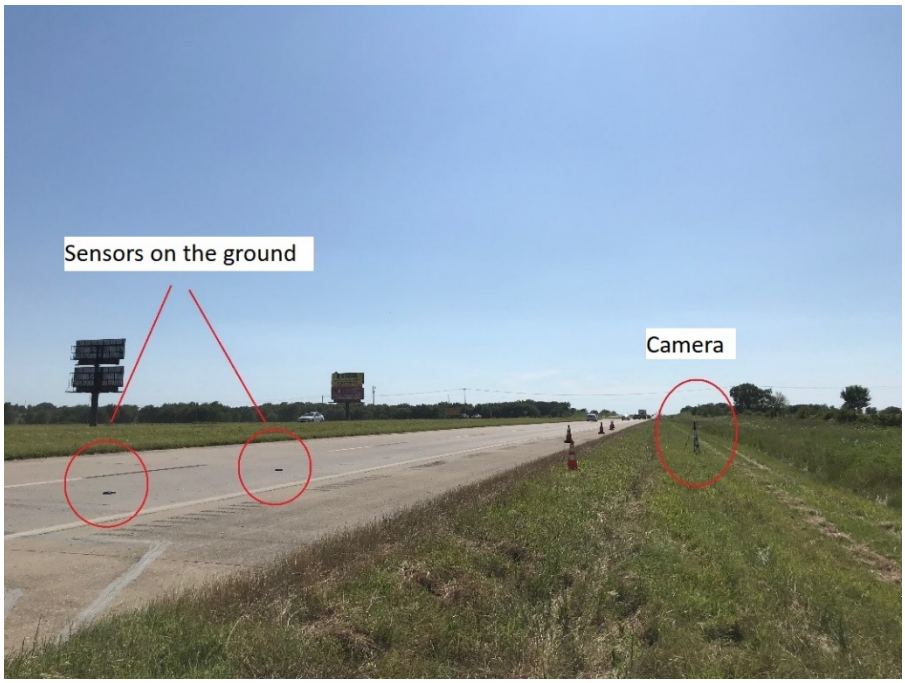

Figure 43. Layout of the deployment on US-412 West.

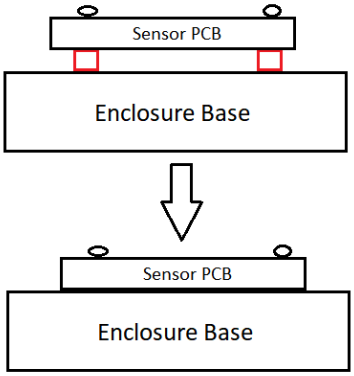

Figure 44. Fixing the sensor inside enclosures without screw guards.

Communication with the sensors was constant throughout the deployment, allowing for real-time data reporting to AP. A 30-degree antenna (See Figure 46) on the AP side was used and placed level to the ground. Site geography was helpful, allowing the antenna to be mounted on a pole at ground level that was lower than the road (See Figure 45).

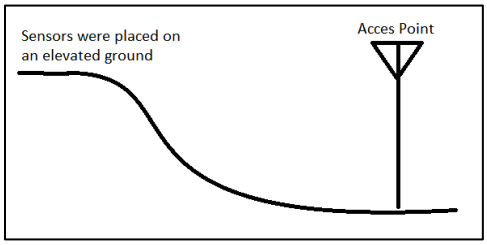

Figure 45. Antenna placement at the level of the sensors.

Figure 46. Access point antenna on the pole.

The three sensors collected 1006, 1077, and 1207 vehicles, respectively. Video recording showed 1073 vehicles. The recording was manually processed to log vehicle count and class. After eliminating outlier vehicles (e.g., those that changed lanes while travelling over the sensors), 970 vehicles were observed.

Time-duration distribution of detected vehicles (i.e., difference between time of arrival and time of departure) from all deployed sensors are depicted in Figure 47. The three sensors have nearly identical distributions. Most detected vehicles spent 0.1 to 0.25 seconds within the sensor's detection zone. Average length of detected vehicles based on class is 4.5 meters. Given the 65-mph speed limit on US-412, average detection time was 0.15 seconds, which conforms with Figure 47.

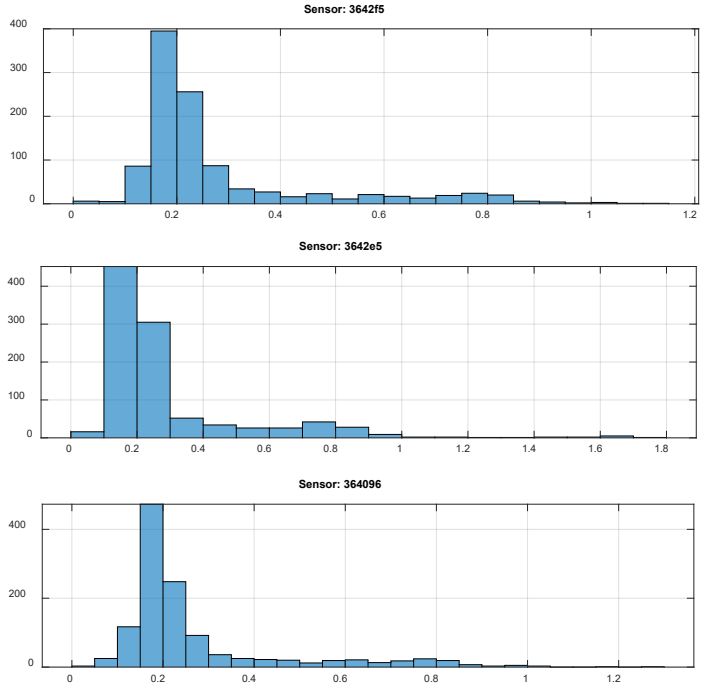

Figure 47. Time duration histogram of detected vehicles.

To further validate collected data, data from two sensor files were a) filtered out by removing data points from the time range of the camera (e.g., 12:30:00 PM to 4:00:00 PM) and b) cross-correlated by finding each timestamp from the first data file and comparing with the second file, taking into

consideration separation distance between sensors. This process added a calculated time tolerance of 0.20 seconds. 925 vehicles were identified, indicating a 95.36% detection accuracy.

Again, since separation distance is known, speed can be estimated from the collected timestamps. Figure 48 shows a plot of speed per detected vehicle. Average speed was 62.68 mph.

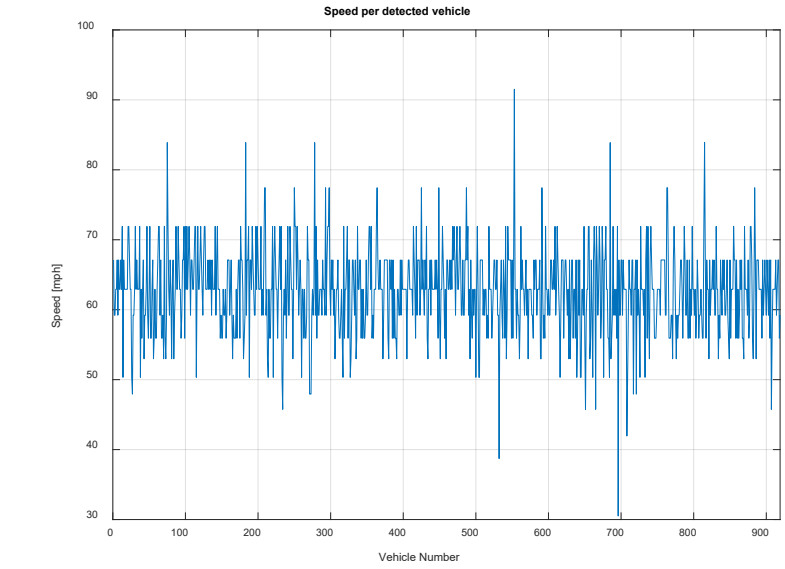

Figure 48. Estimated speed from deployed sensors.

3.8.4 Third Deployment on US-412

A third deployment was carried out on US-412 West, as well, for six hours on July 23rd, 2019 from 10:00 AM to 4:00 PM. Similar to the previous deployment, the system was tested for vehicle detection accuracy and real-time vehicle classification based on vehicle length. Four sensors—two on each lane—were installed on the ground at a separation distance of 4.5 meters. A roadside camera was also set up to validate collected data.

Figure 49. Panoramic view of the sensors and access point location.

AP setup was the same as the previous deployment and utilized the 30-degree antenna. Like before, the AP was able to communicate with all four sensors at a transmit power of 0 dBm during the time they were configured.

The four sensors detected a total of 1066 vehicle between 10:50 AM – 11:40 AM, and 2:00 PM – 3:36 PM. Distributions are provided in the table below.

Table 3. Detection accuracy in each lane.

Sensor ID	364096 - 3642F5	3642F7 - 3642E5
Lane	Slow	Fast
Detected	750	316
Ground Truth	763	363
Accuracy	98.29%	87.05%

Figure 50 and Figure 51 depict speed per detected vehicle on both right and left lanes, respectively.

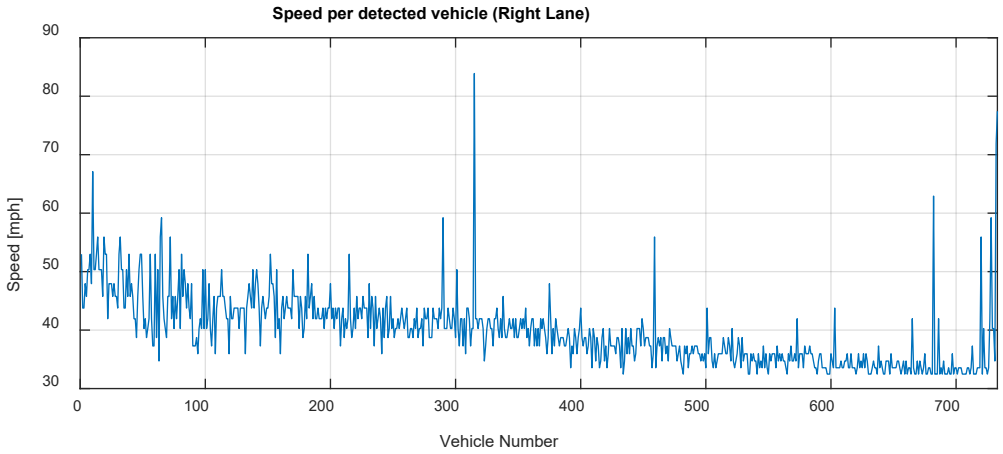

Figure 50. Estimated speed from deployed sensors on the right lane.

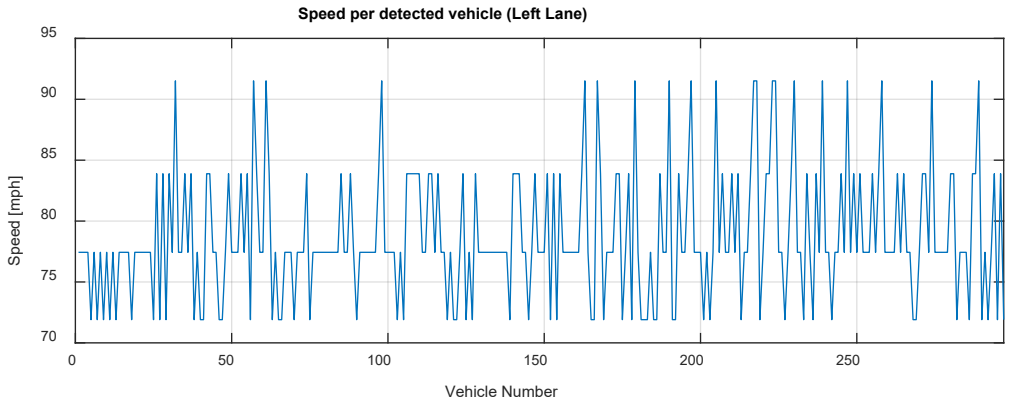

Figure 51. Estimated speed from deployed sensors on the left lane.

Sensors were powered using 8000 mAh batteries and consumed an average of 27.24 mAh during a six-hour deployment. It is worth noting that sensors were connected to the AP at all times and transmitted real-time timestamps at 0 dBm. After dropped packets were observed, Tx power was changed mid-deployment to 4 dBm to maintain link connectivity.

Table 4. Energy consumed by the sensors during the 6-hour deployment.

Sensor ID:	364096	3642f5	3642f7	3642e5
Consumed energy:	27.75 mAh	27.66 mAh	27.16 mAh	26.37 mAh

3.8.5 Fourth Deployment on US-412

A fourth deployment on August 15, 2019, entailed installing four sensors on two lanes—two on each—for three and a half hours. The same location and setup used in the previous deployment was repeated. An AP was set up, even though it was not used to communicate with the sensors since real-time operation was not the objective of this deployment. This enabled us to measure sensor power consumption in a non-real-time setting.

1165 vehicles were distributed between the hours of 10:30 AM and 2:00 PM, as indicated in the table below.

Table 5. Detection accuracy in each lane.

Sensor ID	3642E5 – 3642FA	3642F7 – 3642F1
Lane	Fast	Slow
Detected	317	848
Ground Truth	492	864
Accuracy	64.43%	98.15%

Fast lane data demonstrates lower accuracy than the slow lane. This is attributed to time synchronization issue found later in one of the two sensors on the fast lane. Figure 52 and Figure 53 depict speed per detected vehicle on both fast and slow lanes, respectively.

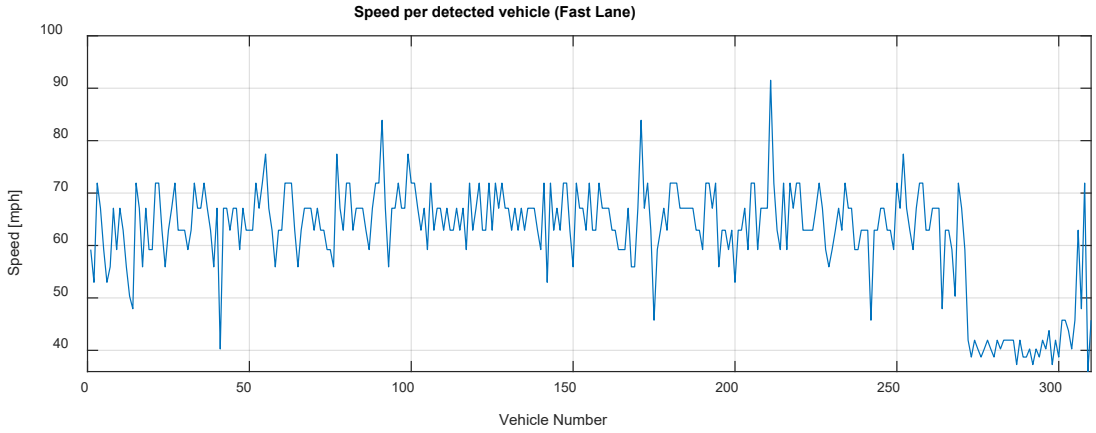

Figure 52. Estimated speed from deployed sensors on the right (fast) lane.

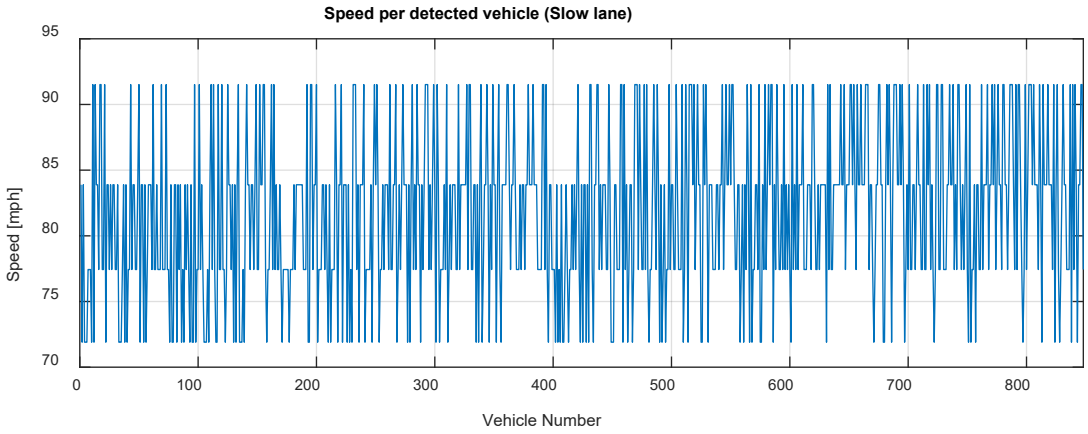

Figure 53. Estimated speed from deployed sensors on the left (slow) lane.

Sensors were powered using 4200 mAh batteries and consumed an average of 10.39 mAh during the three-and-a-half hours.

Table 6. Energy consumed by the sensors during the deployment.

Sensor ID:	3642F7	3642F1	3642E5	3642FA
Consumed energy:	12.59 mAh *	9.69 mAh	10.61 mAh *	8.67 mAh

* Sensors connected to the AP periodically to verify data collection.

Chapter 4 Conclusion

Real-time traffic monitoring systems play a major role in the transition toward smart cities and more efficient ITS. Autonomous traffic sensing is at the heart of smart city infrastructure, wherein smart wireless sensors are used to measure traffic flow, predict congestion, and adaptively control traffic routes. Such information enables a more efficient use of resources and infrastructure.

The research detailed in this report presented the design, development, and implementation of a novel, autonomous, and intelligent wireless sensor for various traffic surveillance applications, each adding intelligence, performance, accuracy, and lifetime to its predecessor.

*i*VCCS 1stG is a portable, inexpensive, and easy-to-install into highway, roadway, or even roadside surfaces without intrusive labor. A novel, multi-threshold-based detection algorithm was also introduced, wherein a drift in geomagnetic reference field baseline threshold—due to variations in temperature, background noise, relative Earth's magnetic field over time, or aging—is adaptively auto-compensated in real-time. This method will aid in solving common problems reported in literature by keeping magnetic signal variation at a minimum; hence, providing reliable vehicle speed estimation under congested traffic, low- and high-speed conditions.

*i*VCCS 2ndG was built around a more powerful processing platform utilizing a 32-bit microcontroller. The 1stG firmware was ported to the new platform in C language for accommodating robustness and speed, which is critical for an improved response time without an increase in power consumption. As part of the process, some algorithms were modified, while others were introduced to shift the operation methodology from a polling mode to an event-driven, interrupt-based procedure, wherein the system responds to interactions from the surrounding environment.

Two operation modes (i.e., HP and LP) were designed for *i*VCCS 2nd G to balance a tradeoff between consumed energy and response time, which affects detection accuracy relative to traffic flow. A Reinforcement Learning (RL) approach was proposed to design a dynamic power management (DPM) algorithm for observing traffic and controlling the system's power policy according to the environment and agent states. Experimental results showed that overall battery life of the system was extended to over 200 days for a 2300 mAh battery [53].

This research was culminated with a state-of-the-art, *i*VCCS 3rdG sensor. The design has been substantially revised in order to enhance power consumption and improve processing power and communications capabilities. The microSD card was replaced with an array of on-board flash memories to reduce the enormous amount of energy consumed by the microSD cards. The power management subsystem has been simplified by eliminating energy harvesting. Alternatively, the sensor will be powered by a high capacity (i.e., 10000 mAh), ultra-wide-range operating temperature, compact-size batteries using Lithium-Thionyl chemistries.

More importantly, *i*VCCS 3rdG supports the new Bluetooth Low Energy 5 as the radio interface, bringing desired IoT paradigm features, such as enhanced data rates, long-rage operation, and over-the-air updates. BLE 5 enables communication, configuration, and data download from sensors using smartphones and tablets, as well as APs on laptops and embedded roadside gateways.

All three generations were tested in several field studies and evaluation deployments during the associated research. Eight field tests were conducted using *i*VCCS 1stG on highways and urban roads in

the state of Oklahoma under various traffic conditions. System performance evaluation was conducted using real-time data, offline data, video images, and reports from the highly accurate Roadrunner Kit 3.

*i*VCCS 2ndG was tested in two scenarios: 1) a lab test using a train running continuously for 24 hours and 2) a field test wherein two sensors were deployed at the south entrance of campus for 24 hours. The sensor captured vehicle count and speed estimate for individual cars. Reported speeds were in line with the expectations and nominal values of a real-world setting.

*i*VCCS 3rdG was deployed on five field tests performed on highways and on The University of Oklahoma – Tulsa campus. Detection accuracy and speed estimation, as well as power consumption of the system in different scenarios, were evaluated during these deployments. Several problems were discovered and solved. The system achieved detection accuracy as high as 98% and on average 88.66%. Reported speed values were compatible with speed limits of their respective locations.

Several enclosure designs were proposed to protect sensors from environmental effects and mechanical forces exerted by passing vehicles. Additionally, materials with minimum effect on radio communication quality were researched. These investigations were crucial for design, as these affect both communication with the sensor using BLE and the GPS ability to receive and lock-on clear constellation. A robust and light-weight enclosure design that meets necessary requirements was designed and manufactured by NCTronics, Inc.

References

[1] ITS (Intelligent Transportation Society of America), “Annual Report 2010-2011,” 2011.

[2] ITS (Intelligent Transportation Society of America), “Rise of the Real-Time Traveler,” 2015.

[3] Cisco, “Cisco Connected Roadways – IoT in Roads, Traffic, Transit - Cisco.” [Online]. Available: https://www.cisco.com/c/en/us/solutions/industries/transportation/connected-roadways.html. [Accessed: 14-Apr-2018].

[4] L. E. Y. Mimbela and L. a Klein, “A Summary of Vehicle Detection and Surveillance Technologies used in Intelligent Transportation Systems.”

[5] M. T. Alamiri, “IoT Systems for Travel Time Estimation,” University of Oklahoma, 2017.

[6] W. Balid, “Fully Autonomous Self-Powered Intelligent Wireless Sensor for Real-Time Traffic Surveillance in Smart Cities,” University of Oklahoma, 2016.

[7] W. Balid, H. Tafish, and H. H. Refai, “Intelligent Vehicle Counting and Classification Sensor for Real-Time Traffic Surveillance,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2017.

[8] W. Balid and H. H. Refai, “Real-Time Magnetic Length-Based Vehicle Classification: Case Study for Inductive Loops and Wireless Magnetometer Sensors in Oklahoma State,” 2018.

[9] Y. Guyodo and J.-P. Valet, “Global changes in intensity of the Earth’s magnetic field during the past 800[thinsp]kyr,” *Nature*, vol. 399, no. 6733, pp. 249–252, 1999.

[10] NASA, “2012: Magnetic Pole Reversal Happens All The (Geologic) Time,” 2015. [Online]. Available: https://www.nasa.gov/topics/earth/features/2012-poleReversal.html. [Accessed: 15-Apr-2018].

[11] N. Wahlstrom and F. Gustafsson, “Magnetometer Modeling and Validation for Tracking Metallic Targets,” *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 545–556, Feb. 2014.

[12] S. V. Marshall, “Vehicle detection using a magnetic field sensor,” *IEEE Transactions on Vehicular Technology*, vol. 27, no. 2, pp. 65–68, 1978.

[13] N. Wahlstrom *et al.*, “Classification of driving direction in traffic surveillance using magnetometers,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1405–1418, 2014.

[14] K. Liu, H. Xiong, and H. He, “New method for detecting traffic information based on anisotropic magnetoresistive technology,” in *Proc. SPIE 8783, Fifth International Conference on Machine Vision (ICMV 2012): Computer Vision, Image Analysis and Processing*, 2013, p. 87830S.

[15] L. Yalong, L. Bin, and W. Zhiqiang, “A novel method of automatic vehicle detection based on active magnetic theory,” in *2010 International Conference on Information, Networking and Automation (ICINA)*, 2010, vol. 2, pp. V2-382-V2-385.

[16] J. L. J. Lan and Y. S. Y. Shi, “Vehicle detection and recognition based on a MEMS magnetic sensor,” *2009 4th IEEE International Conference on Nano/Micro Engineered and Molecular Systems*, pp. 404–408, 2009.

[17] A. Kaadan and H. H. Refai, “iICAS: Intelligent intersection collision avoidance system,” in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, 2012, pp. 1184–1190.

[18] A. Daubaras and M. Zilys, “Vehicle Detection based on Magneto-Resistive Magnetic Field Sensor,” *Electronics and Electrical Engineering*, vol. 118, no. 2, pp. 27–32, Feb. 2012.

[19] Z. Zhang, X. Li, H. Yuan, and F. Yu, “A Street Parking System Using Wireless Sensor Networks,” vol. 2013, no. c, 2013.

[20] H. Zhu and F. Yu, “A Vehicle Parking Detection Method Based on Correlation of Magnetic Signals,” vol. 2015, 2015.

[21] D. Liu *et al.*, “Direction identification of a moving ferromagnetic object by magnetic anomaly,” *Sensors and Actuators A: Physical*, vol. 229, pp. 147–153, 2015.

[22] R. D. Komguem *et al.*, “WARIM: Wireless Sensor Networks Architecture for a Reliable Intersection Monitoring,” in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2014, pp. 1226–1231.

[23] Federal Highway Administration, “Traffic Monitoring Guide,” *Fhwa*, no. October, p. 462, 2013.

[24] B. Coifman and S. Kim, “Speed estimation and length based vehicle classification from freeway single-loop detectors,” *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 4, pp. 349–364, Aug. 2009.

[25] C. H. Wu, J. M. Ho, and D. T. Lee, “Travel-time prediction with support vector regression,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 276–281, 2004.

[26] X. Sun and R. Horowitz, “Set of New Traffic-Responsive Ramp-Metering Algorithms and Microscopic Simulation Results,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1959, pp. 9–18, Jan. 2006.

[27] C. B. Winkler, “Work Zone Safety ITS Smart Barrel for an Adaptive Queue-Warning System,” no. February, 2005.

[28] C. Monsere, C. Nolan, R. Bertini, E. Anderson, and T. El-Seoud, “Measuring the Impacts of Speed Reduction Technologies: Evaluation of Dynamic Advanced Curve Warning System,” *Transportation Research Record*, vol. 1918, no. 1, pp. 98–107, 2005.

[29] A. J. Kean, R. a Harley, and G. R. Kendall, “Effects of Vehicle Speed and Engine Load on Motor Vehicle Emissions,” *Environmental Science & Technology*, vol. 37, no. 17, pp. 3739–3746, 2003.

[30] B. Martinez, M. Monton, I. Vilajosana, and J. D. Prades, “The Power of Models: Modeling Power Consumption for IoT Devices,” *IEEE Sensors Journal*, vol. 15, no. 10, pp. 5777–5789, Oct. 2015.

[31] I. Joe and M. Shin, “Energy management algorithm for solar-powered energy harvesting wireless sensor node for Internet of Things,” *IET Communications*, vol. 10, no. 12, pp. 1508–1521, Aug. 2016.

[32] P. Kamalinejad, C. Mahapatra, Z. Sheng, S. Mirabbasi, V. C. M. Leung, and Y. L. Guan, “Wireless energy harvesting for the Internet of Things,” *IEEE Communications Magazine*, vol. 53, no. 6, pp. 102–108, 2015.

[33] K.-Y. Chan, H.-J. Phoon, C.-P. Ooi, W.-L. Pang, and S.-K. Wong, “Power management of a wireless sensor node with solar energy harvesting technology,” *Microelectronics International*, vol. 29, no. 2, pp. 76–82, May 2012.

[34] M. Hassanalieragh *et al.*, “Health Monitoring and Management Using Internet-of-Things (IoT) Sensing with Cloud-Based Processing: Opportunities and Challenges,” *Proceedings - 2015 IEEE International Conference on Services Computing, SCC 2015*, pp. 285–292, 2015.

[35] L. B. and G. D. M. E.-Y. Chung, “Dynamic power management using adaptive learning tree,” *ICCAD ’99 Proceedings of the 1999 IEEE/ACM international conference on Computer-aided design*, pp. 274–279, 1999.

[36] V. L. Prabha and E. C. Monie, “Hardware Architecture of Reinforcement Learning Scheme for Dynamic Power Management in Embedded Systems,” *EURASIP Journal on Embedded Systems*, vol. 2007, pp. 1–6, 2007.

[37] A. Iranfar, M. Zapater, and D. Atienza, “A machine learning-based approach for power and thermal management of next-generation video coding on MPSoCs,” *Proceedings of the Twelfth IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis Companion - CODES ’17*, pp. 1–2, 2017.

[38] K. Bhatti, C. Belleudy, and M. Auguin, “Power management in real time embedded systems through online and adaptive interplay of DPM and DVFS policies,” *Proceedings - IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, EUC 2010*, pp. 184–191, 2010.

[39] S. Muhammad, “Intelligent Power Aware Algorithms for Traffic Sensors,” University of Oklahoma, 2018.

[40] STMicroelectronics, “STM32L071KB - Ultra-low-power ARM Cortex-M0+ MCU with 128-Kbytes Flash, 32 MHz CPU - STMicroelectronics.” [Online]. Available: http://www.st.com/en/microcontrollers/stm32l071kb.html. [Accessed: 08-Apr-2018].

[41] M. C. U. Arm, “STM32L071x8 STM32L071xB,” no. March. 2016.

[42] Kionix, “Tri-Axis Magnetometer/Tri-Axis Accelerometer - KMX62.” [Online]. Available: https://www.kionix.com/product/KMX62-1031.

[43] ZLG, “Nano Type ZigBee Wireless Module.” [Online]. Available: http://www.zlg.com/wireless/wireless/product/id/136.html. [Accessed: 09-Apr-2018].

[44] Quectel, “GNSS L76 - Quectel Wireless Solutions.” .

[45] SiTime, “SiT1552: 1.2mm2, NanoPower, ±10 ppm 32 kHz TCXO | SiTime.” .

[46] Macronix, “Macronix MX25R6435F Datasheet.” pp. 1–86, 2017.

[47] Texas Instrument, “BQ27621-G1 System-Side Fuel Gauge with Dynamic Voltage Correlation, Battery Gas Gauge | TI.com.” [Online]. Available: http://www.ti.com/product/BQ27621-G1/description. [Accessed: 09-Apr-2018].

[48] Analog-Devices, “Adp5091-2.” .

[49] SanDisk Corporation, “SanDisk SD Card Product Manual,” vol. 95035, no. 80, p. 123, 2004.

[50] R. S. Sutton and A. G. Barto, *Reinforcement Learning : An Introduction*. 2015.

[51] D. L. Poole and A. K. Mackworth, *Artificial intelligence: Foundations of computational agents*, vol. 9780521519. 2010.

[52] J. Skach, I. Puncochar, and F. L. Lewis, “Optimal active fault diagnosis by temporal-difference learning,” in *2016 IEEE 55th Conference on Decision and Control (CDC)*, 2016, no. Cdc, pp. 2146–2151.

[53] S. Muhammad, H. Refai, and M. Blakeslee, “Ultra-Low Power IoT Traffic Monitoring System,” in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, 2018, pp. 1–7.

[54] *Bluetooth Core Specifications*. Bluetooth SIG, 2019.

Appendix:

Proposition for Electronic Circuit Enclosure Anchoring to Asphalt

Preface

Anchoring components to asphalt are challenging. This notion is corroborated with evidence from a literature review and testimony from experienced users. Expectations are that properly affixing components to asphalt requires fastening with concrete that underlies the asphalt or utilizing grout to bond the anchor into a precisely cut or drilled hole. However, considerations for lag shields and wedge anchors (i.e., drop-in anchors) revealed that these solutions are not viable, as both are designed for concrete and require constant side pressure for anchors to stay in place.

The proposition in this brief offers TAPCONS® as the "best" fit for successful alternative fastening. Although TAPCONS are designed for concrete and might likely suffer from problems described above, we recommend that a controlled experimental setup should be devised for a suitable solution. The enclosure design and its anchoring are separate issues that require equal attention and research.

Proposed Installation Hardware to Test

Since wedge anchors, as well as lag shields, have been dismissed for aforementioned difficulties, the following suggestion warrants investigation. We propose utilizing hardware designed for concrete and installing the hardware with PL construction adhesive for binding the anchor to asphalt. Because the enclosure must be removed from the pavement after only a short period of time, a **two-piece installation setup** is suggested. Following is a list of required hardware.

1. HANGERMATE®: ¼"-20 end for fastening a hex cap machine screw with a 3" concrete-type thread for fastening into a precisely drilled and cleaned hole that is then filled with PL construction adhesive.
2. The strongest PL construction adhesive available. Some products are better than others.
3. Grade 8 ¼"-20 hex cap screw with built-in thread-locking patch for fastening the enclosure to anchors. Fastenal is a reliable supplier that offers certified components.
4. Grade 8 hex cap screw without thread-locking patch could also be considered. Given this choice, liquid Loctite should be applied to lock the thread and prevent loosening from shock and vibration. Loctite is anaerobic substance (i.e., hardens in the absence of air). For installation, a bottle is typically filled half-way, making the substance remain in liquid form. While some Loctite thread-locking fluids offer permanent locking, we advised using one that permits disassembly.
5. An SAE-type ¼" flat washer is required to distribute the load under the hex cap screw when installing the enclosure. Simulation showed considerable yielding when a washer was not used. Notably, USS-type washers are configured with alternative dimensions and will not fit with other components.
6. Compressed air for cleaning drilled holes.

Suggested Installation Procedure

Care should be exercised in drilling and cleaning holes, as this is key for proper installation.

- Drill ¼" diameter hole at a depth of 3".
- Clean the hole with compressed air for proper anchor binding to asphalt, using PL construction adhesive.
- Drill ½" diameter countersink hole at a depth of ¾". Verify that hole depths are accurate by using marking tape on drill bits. Depth tolerance is PLUS 1/8" MINUS 0" (i.e., holes can be 1/8" deeper, but not shallower than the local road grade).

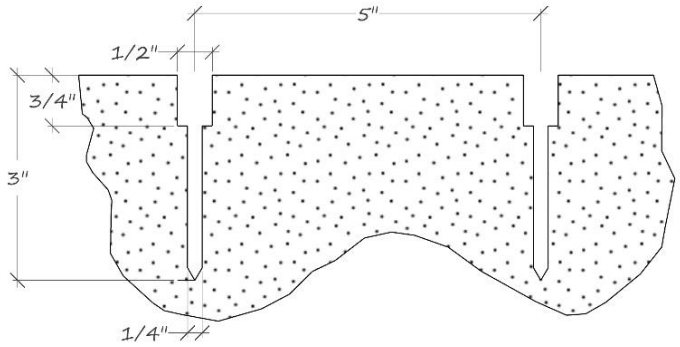

Figure 54 Holes depth, separation, and diameter.

- Inject PL adhesive into both holes.
- Do not over-fill holes to avoid spill over into the pavement. Excess adhesive could seep into the internal ¼"-20 thread, rendering the anchor unusable.

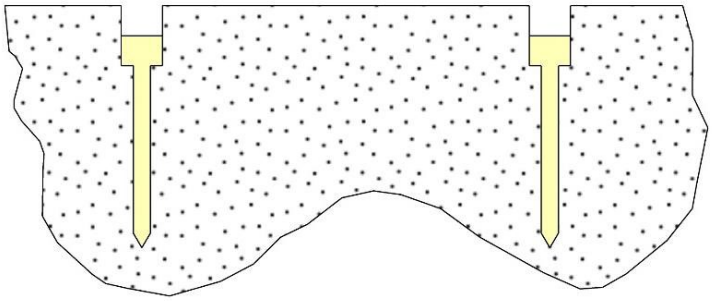

Figure 55 Holes filled with adhesive.

- Drive the HangerMate using a ½" driver socket as far as it will reach. Approximately, ¼" will remain above ground. A 1/4" cap screw can be used to sink the extra ¼".
- Attach a hex cap screw to the thread and continue to drive the ¼" HangerMate until it is flush with the pavement. The Hanger Mate can be installed an additional 1/8", although an alternate installation screw will be required. Sinking the HangerMate more than 1/8" below the surface has proven problematic.

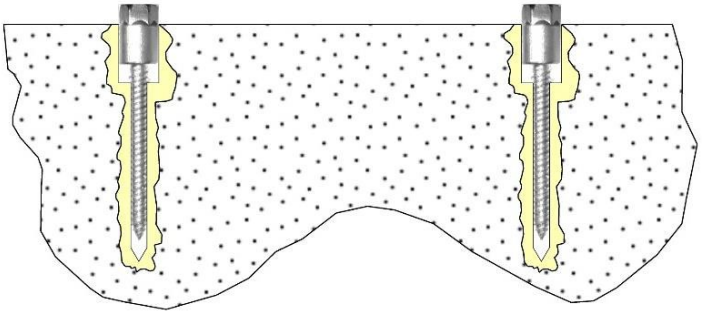

Figure 56 HangerMate inserted into the adhesive.

- Alternatively, a spanner wrench can be used to avoid locking the driving cap screw in the thread.
- Wipe-off excess PL glue, ensuring glue does not get into the thread.
- Repeat for the second hole.
- Remove the temporary ¼" hex cap screw used to drive the anchor the last ¼", given that this alternative technique was used to set the anchor into the hole.
- Leave anchors in place for 24 hours, as most PL construction adhesives require a 24-hour cure time.

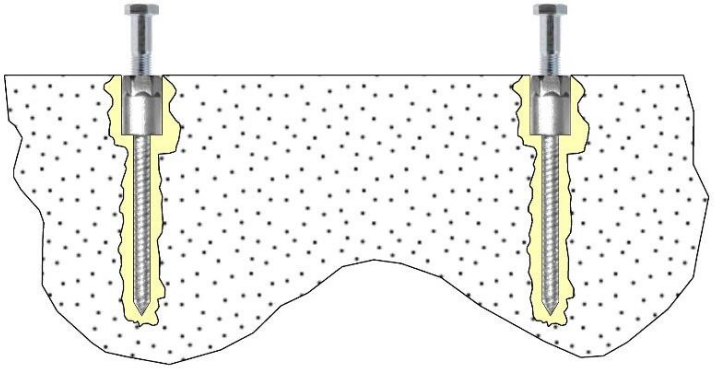

Figure 57 Hex cap screw inserted partially in the threads.

- After applying Loctite to the threads, use a 7/16" socket driver to install the enclosure, along with the supplied ¼"-20 x L 1 1/8".
- Screws will be shortened by 1/8" from a standard ¼"-20 x L 1 ¼" when installing using current Revision 10. The enclosure will be modified later in order to utilize a standard-length hex cap screw.
- Standard-length screws will be supplied as a backup (i.e., in case an anchor is driven beyond the 1/8" tolerance.
- Recall that screws must be installed with washers to spread the load and prevent yielding.
- Future enclosures will utilize screws with thread locking patches. Also, enclosure will be redesigned to accommodate standard-length hex cap screws.

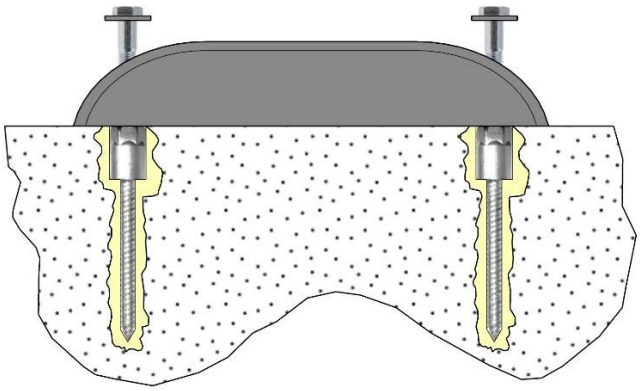

Figure 58 Sensor enclosure placed on the ground and tightened with screws.

Note that guidelines detailed above are merely suggestions derived through simulation. However, no guarantee is implied, as the procedure has not been field-tested. We suggest consulting with highway and city authorities who regularly perform this type of work (e.g., city crews installing road signs).

A controlled experiment to derive data-based specifications will eventually be needed, as simulation results for asphalt are unreliable.

Potential Alternatives

Few suppliers offer anchors designed for use with asphalt. A short list of alternatives is offered below. Notably, currently available anchor sizes are incompatible with the proposed enclosure, although some suppliers offer custom-made hardware. Inquiries should be made directly with the manufacturer.

- WS Hampshire Asphalt Anchors
- Screw Fix
- BoltHold Asphalt Anchor