

Ultra-Low Power IoT Traffic Monitoring System

Siraj Muhammad, Hazem Refai
Electrical and Computer Engineering
University of Oklahoma
Tulsa, OK, USA
sirajmuhammad@ou.edu, hazem@ou.edu

Matthew Blakeslee
Oklahoma Department of Transportation
Oklahoma City, OK, USA
mblakeslee@odot.org

Abstract—Given the sizable anticipated proliferation of Internet of Things (IoT) devices, Forrester Research forecasts that the fleet management and transportation industry sectors will enjoy more growth than others. This may come as no surprise, since infrastructure (e.g., roadways, bridges, airports) is a prime candidate for sensor integration to provide real-time measurements and to support intelligent decisions. The predicted increase of deployed devices makes it difficult to calculate the amount of energy required for these functions. Current estimates suggest that 2 to 4% of worldwide carbon emissions can be attributed to the information and communication industry [1]. This paper presents novel algorithms designed to optimize power consumption of an intelligent vehicle counter and classifier sensors. Each was based on an event-driven methodology wherein a control block orchestrates the work of different components and subsystems. System duty-cycle is reduced through several techniques, and a reinforcement learning algorithm is introduced to control the system power policy, according to the traffic environment. Battery life for a sensor supported by a 2300 mAh battery was extended from 48-hour, adopted all-on policy to more than 400 days when leveraging the algorithms and techniques presented in this work.

Index Terms—IoT, low-power, ITS, sensors, reinforcement learning, DPM

I. INTRODUCTION

Vehicle detection and classification are essential components of intelligent transportation systems (ITS) and are integral to real-time traffic monitoring and analysis. Commuters, traffic administrators, and regulatory agencies require ITS for improving traffic control and management, as well as for planning trips and making routing decisions. A number of technologies have been leveraged for vehicle detection and surveillance. These include Inductive Loop Detectors (ILD), Weigh-In-Motion (WIM) systems, piezoelectric sensors, magnetic sensors, video image processors, microwave radars, and infrared sensors, to name a few. Additional details about these technologies and others can be found in [2]. Non-traditional technologies, like Bluetooth, have also been utilized for detecting vehicles and estimating highway travel time [3]. The era of Internet of Things (IoT), however, is redefining the objectives and standards of all systems. IoT devices are promoted as smart, reliable, and low-cost technologies that require little maintenance. For ITS systems to fit into the emerging IoT paradigm, they must adhere to inherent requirements of a model constrained by available resources for processing, memory, power, and security. This imposes additional difficulties

and challenges for ITS, especially given the nature of the applications they target. This work reports an extension of a project focused on designing a low-cost, reliable, and low-power intelligent vehicle detection and classification sensor (*i*VCCS) [4]. Research focuses on minimizing sensor power consumption while maintaining accurate vehicle detection, count, and speed estimates.

Power consumption is a major concern for wireless sensor node design relative to the effectiveness of utilizing a single element for unit processing, power management efficiency, algorithm operation, and, most importantly, overall component architecture of the system. Device and peripheral selection is a crucial first step [5]–[7]. Second generation *i*VCCS (*i*VCCS 2ndG) was designed to operate on low power. Literature suggests building a predictor to aid in estimating residual power in the main energy storage and to predict the amount of power harvested during future time slots [8], [9]. Such mechanism indicate which functionalities to disable and for how long, based on the corresponding peripheral power profile. In [8], researchers optimized the RF (radio frequency) transmission time, as it proved to be the most power-hungry element in their system. Optimal transmission interval was determined. Determining ideal software relative to energy consumption is critical. Given that the consumption associated with an algorithm is not optimized, it is pointless to select low-power components, as several techniques must be orchestrated to achieve satisfactory performance. Most energy management implementations involve intelligent use of peripherals and sensors in the system [8]–[11]. As such, this selection should be central to the power management process. An algorithm must be employed to determine the optimal time for enabling and disabling each module on the board, as well as for indicating the flexible control and switching between power schemes. A fixed profile operating the system in different situations is ineffective and inefficient. Authors in [12] modeled the Dynamic Power Management (DPM) algorithm to characterize system behavior as a time series of busy and idle periods, suggesting that various idle intervals require different sleep modes. They reasoned that switching necessitates more time and energy than nominal consumption. Accordingly, the researchers employed adaptive tree learning for policy selection and utilized a predictor to indicate idle states and durations.

In [13], researchers used Reinforcement Learning to adap-

tively choose the preferred power management policy for a given idle state wherein an action is taken and objectives are evaluated. Based on evaluation outcomes subject to constraints (e.g. performance), state-action pairs are rewarded/punished and a Q-value is assigned. A variation of the Q-learning algorithm, namely SARSA (State-Action-Reward-State-Action), was utilized to monitor and record all values in a Q-table. The authors assumed idle time is known and can be calculated. A more recent paper [14] also utilized a Q-learning algorithm to optimize power consumption in their video encoder System-on-Chip (SoC). Researchers in [15] proposed a machine learning algorithm to select the optimal policy online. A variant of reinforcement learning with weights was placed on policies, and a loss function was used to update weights. This group also assumed known idle times.

The balance of this paper is organized as follows. Section II summarizes the system platform, and Section III discusses various proposed techniques for enhancing system power consumption. A description of the reinforcement learning (RL) algorithm used to further optimize power consumption is presented in Section IV. Section V evaluates this proposed approach, and Section VI concludes the paper.

II. SYSTEM OVERVIEW

*i*VCCS 2ndG is a fully autonomous and low-power sensor designed to self-power and energy harvest without sacrificing performance. The updated *i*VCCS is carefully crafted in a compact design (45×30×6 mm) and combines high-performance, energy-efficient components equipped with a power management subsystem for minimizing consumption while maintaining accurate vehicle count, logging, and speed estimation. Fig. 1 shows the system board and its primary components. A battery gauge monitors battery capacity and nano-power load switches placed at the power lines of certain energy-hungry sensor components (e.g., RF wireless module, GPS receiver, SD card and others) are reported to the microcontroller unit (MCU).

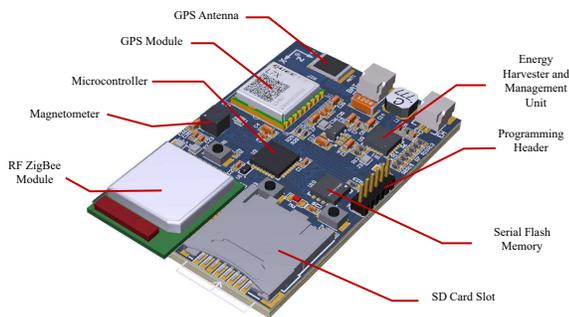


Fig. 1. Overview of the *i*VCCS components.

The sensor is built around an ultra-low-power platform from STMicroelectronics with an ARM CORTEX-M0+ 32-bit RISC core STM32L071 [16]. Kionix KMX62 is a six degrees-of-freedom magnetometer/accelerometer inertial sensor system

[17] that senses changes in the magnetic flux of the earth's surface when vehicles enter the vicinity. KMX62 is a reliable and power-efficient sensor, consuming 395 μ A at high-resolution mode and 1 μ A in standby mode. Magnetometer full-scale range is $\pm 1200 \mu$ T, and digital bit depth is 16 bits, resulting in a magnetic sensitivity of $\pm 0.0366 \mu$ T, which was $\pm 0.1 \mu$ T in *i*VCCS 1stG.

The system integrates a microSD card slot used for raw data acquisition (e.g., vehicle magnetic signature and accelerometer data) and timestamps of vehicle arrival and departure, as well as status messages. The microSD card is connected to the microcontroller through serial peripheral interface (SPI). An on-board 64 Mb serial NOR Flash memory serves as a secondary storage medium. Macronix's MX25R64 [18] is an ultra-low-power CMOS flash memory with minimum of 100,000 erase/program cycles and 20-year data retention; it features a typical standby current of 5 μ A, a maximum 4 mA read current, and 6 mA write current. Readers interested in further details on system components are encouraged to refer to [4], [19], [20].

III. SYSTEM ALGORITHMS AND ANALYSIS

A. Detection Algorithm

The detection algorithm developed in *i*VCCS 1stG is ported to the new design with modifications to fit the new platform [19]. The algorithm processes the magnetic flux sampled by KMX62 and detects the arrival and departure of a vehicle passing through the sensor zone. Vehicles have ferrous materials that cause disturbance in the local magnetic field and create a push-and-pull effect in flux lines as the vehicle passes. This produces fluctuations in the magnitude of the magnetic field. KMX62 is a tri-axial sensor that represents each sample point by three 16-bit words (i.e., x, y, z). The microcontroller calculates the magnitude and feeds it to the algorithm for processing. Three thresholds and three debounce timers govern the algorithm. R_{TH} represents the baseline threshold, O_{TH} the onset threshold, and H_{TH} the holdover threshold. These parameters define vehicle arrival and departure. Fig. 2 depicts a sample signature of a vehicle with thresholds and timers illustrated. Given that the magnitude reaches the O_{TH} threshold, a timestamp of the arrival is logged. If magnitude drops below H_{TH} , the vehicle is assumed departed, another timestamp is logged, and the vehicle counter adds one. In the absence of a vehicle in the vicinity, the magnitude should stay below R_{TH} . Given the magnitude rises between R_{TH} and H_{TH} , calibration procedure is executed. Debounce timers are crucial for eliminating misdetections and double detections. Onset debounce timer O_{DT} is used to filter glitches in the signature that result in misdetection. Double detections are a consequence of ferrous material distributed throughout the body of a vehicle. For example, long trucks with sizable separation between axles cause dips and steep fluctuations between O_{TH} and H_{TH} , resulting in counting a single vehicle twice. Holdover debounce timer H_{DT} is used to overcome this problem.

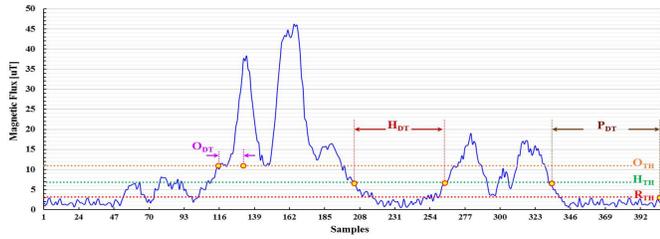


Fig. 2. Example of a vehicle signature.

B. Data Buffering Technique

As indicated in Section II, the microSD card is used to log timestamps and store status messages, as well as raw data. However, microSD cards are energy inefficient and counter intuitive in the low power paradigm, simply dumping all data and messages directly to the card. According to SanDisk microSD card specifications [21], Read and Write procedures can deplete up to 100 mA current consumption. Notably, it is impractical to turn the card on and off for each detected vehicle due to the time delay required for initializing the card each time it is powered on. Specifications also refer to an automatic sleep feature wherein the cards enter sleep mode when commands have not been received within 5 ms. Although cards consume $350 \mu\text{A}$ in this mode, this amount is inefficient. On-board ultra-low-power flash memory is used to buffer data before shifting it to the microSD card, primarily because it excels in energy efficiency and read/write performance when compared with SD cards. As such, this type of memory is suitable for instantaneous data logging.

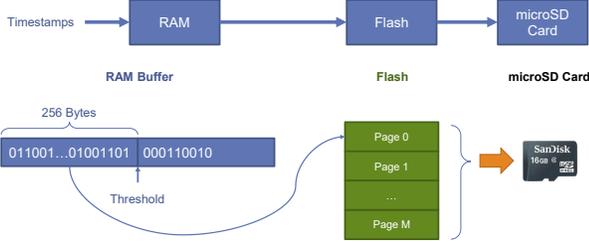


Fig. 3. Buffering technique.

MX25R64 exhibits a page-basis write functionality wherein page size is 256 bytes. The host can commence reading at any byte address, but can only write at the beginning of a page address. Hence, the microcontroller must track data length for writing in order to avoid gaps in the flash. Fig. 3 illustrates this procedure, and Alg. 1 provides a high-level description of the process.

C. Triggered Vehicle Detection

Typically, and in accordance with the configured data rate, the magnetometer regularly interrupts the microcontroller when a new sample is acquired in the buffer (i.e., Data Ready Interrupt [DRI]). Desired behavior interrupts MCU only when

Algorithm 1 Data Buffering Technique.

```

1: if new sector then
2:   erase sector
3: end if
4: Move: RAM Buffer  $\leftarrow$  Data
5: if RAM Buffer Size  $\geq$  256 then
6:   Move: Flash  $\leftarrow$  RAM[0 : 255]
7:   Move: RAM[0...255]  $\leftarrow$  RAM[256 : 511]
8: end if
9: if Flash Full Pages  $\geq$  Flash Threshold then
10:  SD Card: open file
11:  for each Flash page i do
12:    Move: SDCard  $\leftarrow$  Flash[i]
13:  end for
14:  Move: SDCard  $\leftarrow$  RAM[0 : 255]
15:  SD Card: close file
16: end if

```

a vehicle is approaching and triggers the detection algorithm. In addition to DRI, KMX62 magnetometer has two other interrupts, 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 (e.g., increasing or decreasing) and remains above that threshold for a specified number of samples (i.e., time). KMX62 also has a buffer capable of holding 64 samples of components x, y, and z.

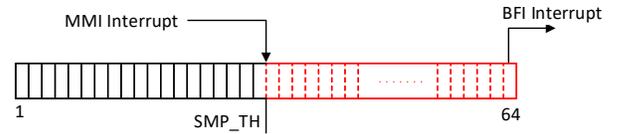


Fig. 4. KMX62 MMI and BFI interrupts operation.

Buffer functions in triggered mode. Given that a physical interrupt is caused by a digital engine (e.g., 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. Fig. 4 illustrates this mechanism. MMI is configured to trigger the buffer, and a BFI is routed to the MCU through one of the GPIOs [17]. SMP_TH is set to 63, which causes KMX62 to immediately interrupt the MCU after the first sample received at the event of an MMI interrupt when a vehicle approaches.

Alg. 2 describes how the manner in which the algorithm reads and controls data flow from the KMX62. MCU executes a dummy read for a number of samples, primarily because the first few samples might not relate to a vehicle signature. The algorithm is configured to feed the last 24 samples prior to vehicle approach and discard the first 40 samples from the buffer. After 24 samples are read and processed, and given that the system's state machine indicates detection, the MCU continues pulling new samples through Data Ready Registers

until vehicle departure.

Algorithm 2 Data flow control.

```

1: if DRI Flag then
2:   Read data from Data Ready registers
3: else if BFI Flag then
4:   Disable BFI
5:   Discard 40 samples
6:   while Samples Counter < 24 do
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

```

D. Empirical Power Consumption Analysis

In this section, the system’s power consumption is empirically analyzed by dividing the sensor’s one-hour cycle into states, and then measuring the current drained in each state. Assume the sensor is deployed on a highway with a 65 mph speed limit and vehicles are spaced 1 second apart with flow rate of 3089 passenger vehicles per hour in a single lane. Flash threshold of MCU buffers data from flash to microSD card is 10 pages (i.e., 2560 bytes). The sensor logs TA (time of arrival), TD (time of departure) and vehicle number in counter (N) in each direction. Additionally, sensor logs battery status each minute. During time laps of one hour, the sensor will advance through the following states: battery status log, vehicle log, flash-to-microSD card buffering, and standby.

```

a) [BAT: VOLT = 4021 mV, CAP = 2480 mAh, SOC = 95 %]<CR><LF>
   b) N01_TA@21225109.160156<CR><LF>
      N01_TD@21225109.519531<CR><LF>
      N01_N#73<CR><LF>

```

Fig. 5. a) Battery status log line b) Vehicle timestamp example.

During a one hour period, the sensor will log 60 lines of battery status at 51 bytes each. Fig. 5a. shows an example line.

$$B_{bat} = 60 \times 51 = 3060 \text{ [Bytes/Hour]} \quad (1)$$

For each detected vehicle, the timestamps and counter compose 54 bytes of logged data (or 56 bytes when the counter is four digits). See Fig. 5b.

$$B_{veh} = 3089 \times 54 = 166806 \text{ [Bytes/Hour]} \quad (2)$$

According to the aforementioned assumptions, the sensor will collect a total of 169,866 bytes in one hour. Since the MCU moves data from Flash to microSD card every 10 Flash pages (i.e., 2560 bytes), the results is 66 transfers.

Timing and drained current for each state is measured using a high-accuracy Fluke 289 multimeter [22]. Fig. 6 depicts current consumed by the sensor at different execution states. Each data transfer is 1 second in duration and consumes at most 30 mA. Average current for one hour is written as:

$$I_{avg} = \frac{T_B I_B + (3600 - T_B) I_d}{3600} = \frac{66 \times 30 + 3534 \times 4}{3600} = 4.48 \text{ [mA]}, \quad (3)$$

where T_B , I_B are buffering time and current, respectively. I_d is the detection/standby current. Thus, a 2300 mAh battery with up to an 80% derating factor should last for:

$$T_{Bat} = \frac{2300 \times 0.8}{4.48} = 411 \text{ [h]} \quad (4)$$

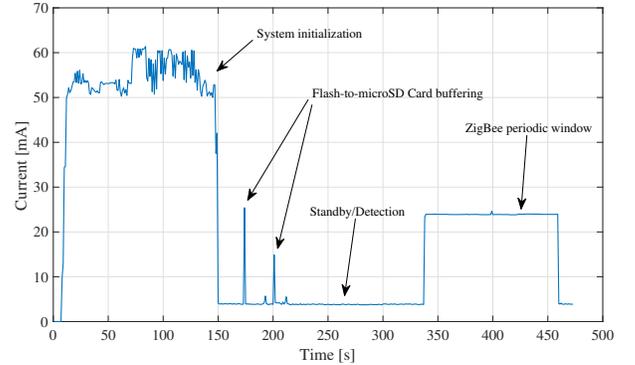


Fig. 6. Current consumption and different states.

IV. REINFORCEMENT LEARNING FOR POWER MANAGEMENT

Originally, the sensor exhibited an all-on policy, wherein all on-board components were turned on and the sensor lasted only 48 hours. Operation flexibility was added by turning specific components on and off, following a predetermined action flow and resulting in reduced current consumption, as shown in Fig. 6. Although power consumption was targeted for enhancement by incorporating the ARM Cortex-M0 micro-processor sleep mode, doing so caused misdetections and/or double counting for vehicles traveling with less than a 2-second following distance. This is attributed to a long wake-up time from sleep mode once a passing vehicle was detected. Nevertheless, in a low-traffic situation where vehicle following distance was more than 5 seconds, sleep mode is expected to perform flawlessly and to conserve power. A dynamic power management scheme is planned to manage this process.

A. Problem Formulation

The proposed system is modeled as a Markov Decision Process 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). Fig.

7 shows the system state transition diagram. Solid arrows represent agent transitions. Dashed arrows represent environment transitions. The agent (i.e., *i*VCCS) is not penalized for environment transitions.

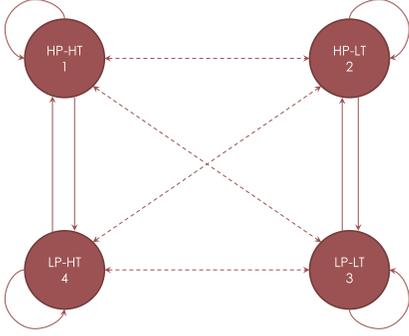


Fig. 7. State transition diagram.

The agent can perform two actions, namely switching to high-power mode or to low-power mode. Given that the agent takes action, a corresponding reward is given and a new experience (e.g., tuple of state, action, reward $\langle s, a, r \rangle$) is formed. Q-value $Q^*(s, a)$ is the expected value of an action a in state s , following the optimal policy; this is defined as:

$$Q^*(s, a) = \sum_{s'} P(s'|s, a)(R(s, a, s') + \gamma V^*(s')), \quad (5)$$

where $V^*(s)$ is the expected value following an optimal policy from state s . The Temporal Differences (TD) equation is used for estimating this value:

$$Q(s, a) = Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a') - Q(s, a)), \quad (6)$$

where $\max_{a'} Q(s', a')$ is the maximum Q-value for future state s' over all possible actions a' . $0 < \alpha \leq 1$ determines the weight of newer values compared with older values; γ is the discount factor; and r is the reward corresponding to an action.

Simulation demonstrated how the algorithm converges in approximately 500 steps and how the agent correctly learned the action necessary for each state.

B. Power Consumption Analysis

When employing RL algorithm, current drained by the system can be calculated utilizing a similar approach to the analysis detailed in Section III-D. 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. Fig. 9 represents current consumed when a vehicle is detected. Vehicle time spent traveling over the sensor determines length of detection

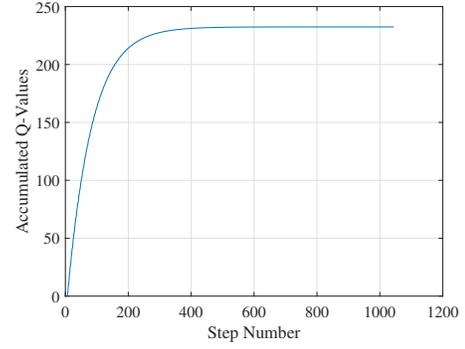


Fig. 8. Q-Matrix convergence.

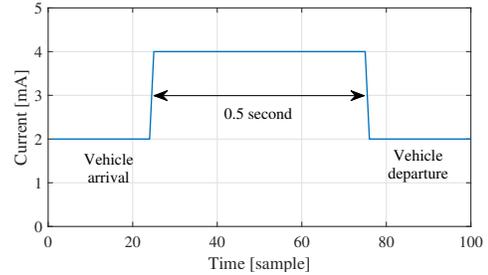


Fig. 9. Current consumed when MCU wakes up from sleep mode.

state, which was statistically calculated as 0.5 seconds on average.

Acknowledging previous assumptions, equation (3) can be rewritten to express current consumption, leveraging the RL algorithm:

$$I_{avg} = \frac{T_B I_B + N_v T_v I_d + (3600 - T_B - N_v T_v) I_s}{3600} \quad (7)$$

$$= 3.71 \text{ [mA]},$$

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 to:

$$T_{Bat} = \frac{2300 \times 0.8}{3.71} = 496 \text{ [h]} \quad (8)$$

V. EXPERIMENTAL SETUP AND RESULTS

The proposed system was tested in two scenarios: a) in lab using a train operating continuously for 24 hours and b) during an on-campus field test in which two sensors were deployed at the campus entrance for 24 hours.

A. Lab Test

The detection algorithm was validated, and battery life was examined in the lab test, wherein the sensor was placed under a miniature train track with a train operating for 24 hours at varying speeds. Fig. 10 illustrates the setup. Results were compared with the empirical analysis detailed in Section III-D.

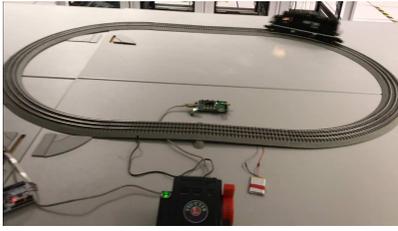


Fig. 10. Lab setup using a train. Sensor is placed under the track, and ZigBee AP is next to it.

Fig. 11 lists how often the sensors detected the train, along with reported speed calculated using the distance between the two sensors.

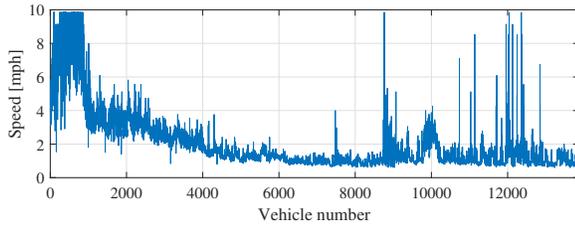


Fig. 11. Speed estimates of the train over 24 hours.

The sensor consumed only 11 mAh in over 24 hours (i.e., 0.458 mAh per hour), which is 10 times less than predicted in the analysis described in Section III-D.

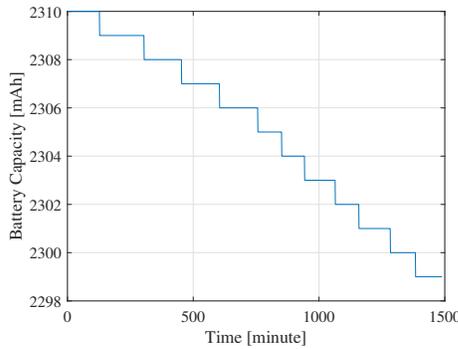


Fig. 12. Battery capacity of the sensor over 24 hours in a lab test.

B. On-Campus Deployment

Two sensors were deployed at the south entrance of the campus: one with the proposed DPM algorithm managing the power policy and the other running a plain version of the firmware. Fig. 13 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 post-processing stage, wherein detected vehicle timestamps were used with known separation distance (e.g., 2 meters) between



Fig. 13. Sensor setup on the entrance of campus.

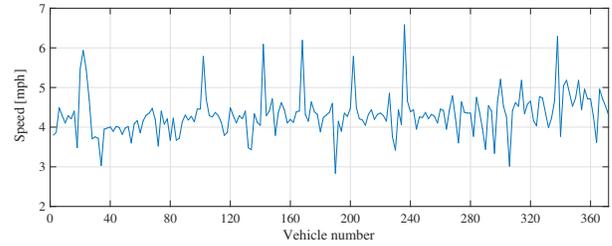


Fig. 14. Speed estimates of detected vehicles on campus.

the two sensors. Speed and power results are depicted in Fig. 14 and Fig. 15, respectively.

Average speed was 5 mph, which is fairly logical given sensor location at the entrance to the campus and a 10 mph speed limit. Processing the collected power data revealed that the sensor running original firmware (sans DPM) consumed approximately 18 mAh, resulting 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.

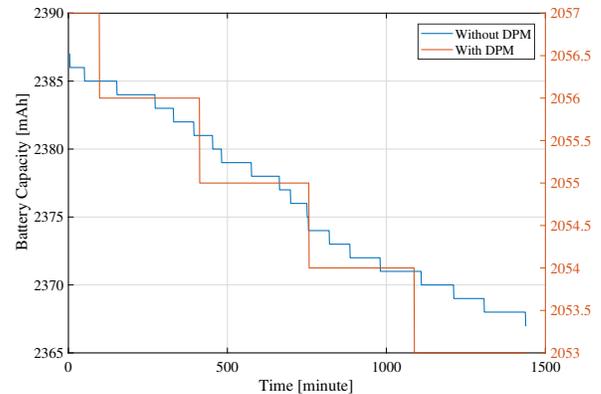


Fig. 15. Battery capacity of sensor with and without DPM over 24 hours.

VI. CONCLUSION

This research presented algorithms and methods for optimizing system response and reducing power consumption of an intelligent vehicle counter and classifier sensor (iVCCS). The initial sensor prototype adopted an all-on policy, wherein all components were active in spite of the fact that battery life was depleted in nearly 48 hours. Algorithms presented herein shift operation methodology from polling mode to an event-driven, interrupt-based process in which the system responds to vicissitudes of the surrounding environment. Conservative analysis revealed an 18-day battery life, whereas experimental results demonstrated overall system battery life was extended to over 100 days for a 2300 mAh battery.

A reinforcement learning approach was proposed for designing a dynamic power-management algorithm to further minimize power consumption. The algorithm observes the traffic and controls system power policy, according to the environment and sensor states. The algorithm was also compared with the previous one-policy optimization, and an enhancement of over 400 days was observed using the DPM scheme.

Future studies include additional field evaluations in various traffic environments (e.g., urban roadways and highways). Additionally, iVCCS 2ndG will be equipped with energy harvesting and management chips for hosting a solar panel through a dedicated port. Future research should aim toward a self-powered system by exploiting this capability.

ACKNOWLEDGMENT

The authors would like to thank ODOT Research Office for funding this study and acknowledge Dr. Walid Balid for his assistance in the development of this research.

REFERENCES

- [1] W. Vereecken, W. Heddeghem, M. Deruyck, B. Puype, B. Lannoo, W. Joseph, D. Colle, L. Martens, and P. Demeester, "Power consumption in telecommunication networks: overview and reduction strategies," *IEEE Communications Magazine*, vol. 49, no. 6, pp. 62–69, jun 2011. [Online]. Available: <http://ieeexplore.ieee.org/document/5783986/>
- [2] L. E. Y. Mimbela and L. a. Klein, "A Summary of Vehicle Detection and Surveillance Technologies used in Intelligent Transportation Systems."
- [3] M. T. Alamiri, "IoT Systems for Travel Time Estimation," Master's thesis, University of Oklahoma, 2017.
- [4] W. Balid, "Fully Autonomous Self-Powered Intelligent Wireless Sensor for Real-Time Traffic Surveillance in Smart Cities," Ph.D. dissertation, University of Oklahoma, 2016. [Online]. Available: <https://shareok.org/handle/11244/51828>
- [5] C. Park, P. H. Chou, Y. Bai, R. Matthews, and A. Hibbs, "An ultra-wearable, wireless, low power ECG monitoring system," in *IEEE 2006 Biomedical Circuits and Systems Conference Healthcare Technology, BioCAS 2006*, 2006, pp. 241–244.
- [6] M. T. Lazarescu, "Design of a WSN platform for long-term environmental monitoring for IoT applications," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 1, pp. 45–54, 2013.
- [7] O. Olorode and M. Nourani, "Reducing leakage power in wearable medical devices using memory nap controller," in *2014 IEEE Dallas Circuits and Systems Conference: Enabling an Internet of Things - From Sensors to Servers, DCAS 2014*, 2014, pp. 1–4.
- [8] 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. [Online]. Available: <http://digital-library.theiet.org/content/journals/10.1049/iet-com.2015.0223>
- [9] 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. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84933042177&partnerID=40&md5=c4cb75ed16f19dba343086717187bb91>
- [10] 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. [Online]. Available: <http://www.emeraldinsight.com/doi/10.1108/13565361211237662>
- [11] M. Hassanaliheragh, A. Page, T. Soyata, G. Sharma, M. Aktas, G. Mateos, B. Kantarci, and S. Andreescu, "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.
- [12] L. B. E.-Y. Chung and G. D. Micheli, "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. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=810661>
- [13] 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. [Online]. Available: <http://jes.eurasipjournals.com/content/2007/1/065478>
- [14] A. Iranfar, M. Zapater, and D. Aienza, "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. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=3125502.3125533>
- [15] 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.
- [16] 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>
- [17] Kionix, "Tri-Axis Magnetometer/Tri-Axis Accelerometer - KMX62." [Online]. Available: <https://www.kionix.com/product/KMX62-1031>
- [18] Macronix, "Macronix MX25R6435F Datasheet," pp. 1–86, 2017.
- [19] 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. [Online]. Available: <http://ieeexplore.ieee.org/document/8026566/>
- [20] W. Balid and H. H. Refai, "On the development of self-powered iot sensor for real-time traffic monitoring in smart cities," in *2017 IEEE SENSORS*, Oct 2017, pp. 1–3.
- [21] SanDisk Corporation, "SanDisk SD Card Product Manual," vol. 95035, no. 80, p. 123, 2004.
- [22] Fluke, "True-RMS AC Voltage and Current Logging Multimeter — Fluke 289." [Online]. Available: <http://en-us.fluke.com/products/digital-multimeters/fluke-289-digital-multimeter.html>