Enhancing energy efficiency in tyre pressure and temperature monitoring systems

Praveen Uday Kalkundri[1], Veena Desai[2], Ravi Uday Kalkundri[1]
[1]Department of Electronics and Communication, Karnataka Law Society's Gogte Institute of Technology, Visvesvaraya Technological University, Belagavi, India
[2]Department of Computer Science and Engineering, Karnataka Law Society's Gogte Institute of Technology, Visvesvaraya Technological University, Belagavi, India

Article Info

ABSTRACT

This study addresses the pivotal challenge of enhancing power efficiency in tyre pressure and temperature monitoring systems (TPMS) for heavy vehicles and trailers. Employing field-programmable gate arrays (FPGA) and adaptive channel frequency hopping in Bluetooth low energy (BLE) communication, the research focuses on mitigating power consumption issues specific to heavy vehicles with multiple tyres. The proposed solution incorporates strategic BLE channel blocking and adaptive frequency hopping on the FPGA platform to alleviate channel congestion and interference, ultimately reducing TPMS power consumption. The FPGA's adaptability tailors frequency hopping strategies to automotive TPMS nuances, optimizing channel selection and minimizing energy-intensive processes. Empirical results showcase a significant reduction in power consumption, with the TPMS operating at 100 MHz during active mode consuming 66 mW, dropping to 11 mW in sleep mode, and reaching 0 mW in hibernate mode for the majority of operational time. This research establishes a practical FPGA-based approach for power optimization in commercial TPMS, promising heightened reliability, safety improvements, and environmental impact reduction in the automotive sector.

Corresponding Author:

Praveen Uday Kalkundri
Department of Electronics and Communication, Karnataka Law Society's Gogte Institute of Technology,
Visvesvaraya Technological University
Belagavi-590018, India
Email: pukalkundri@git.edu

1. INTRODUCTION

Tyre pressure monitoring systems (TPMS) have become a ubiquitous feature in modern cars, but they are also increasingly being adopted in heavy vehicles and trailers. In this essay, we will explore what TPMS is, the benefits of its implementation in heavy vehicles and trailers, and the practical considerations involved in implementing TPMS in these contexts. TPMS is an electronic system that monitors the air pressure and temperature of a vehicle's tyres. It uses sensors mounted on each wheel to gather data, which is then processed and transmitted to the vehicle's computer system. The dynamic behavior and safety of heavy vehicles and trailers are heavily reliant on tyre pressure [1]. Maintaining the appropriate tyre pressure is crucial for preventing accidents and optimizing vehicle performance. TPMS systems provide real-time monitoring of tyre pressure and temperature, allowing drivers to detect and address any issues before they become a major safety concern [2]. Furthermore, TPMS in vehicles, heavy vehicles, and trailers also offer several benefits. Firstly, it helps to enhance overall vehicle safety. By monitoring tyre pressure and

temperature in heavy vehicles and trailers, TPMS plays a critical role in maintaining vehicle stability and ensuring driver safety. The ability to detect deviations in tyre pressure and temperature in real time enables drivers to take immediate corrective action, mitigating the risk of tyre failure and potential accidents. Additionally, proper tyre pressure directly impacts the vehicle's handling and maneuverability, especially in heavy vehicles and trailers where the weight and load distribution can significantly influence tyre performance [3].

Moreover, the implementation of TPMS in heavy vehicles and trailers aligns with the industry's efforts to improve fuel efficiency and reduce carbon emissions. Underinflated tyres can increase rolling resistance, leading to decreased fuel efficiency and increased carbon emissions. Implementing a TPMS using various wireless technologies involves deploying sensor nodes to measure tyre pressure and temperature and transmitting this data wirelessly to a central monitoring system. A few of the technologies implemented are Bluetooth low energy (BLE), Wi-Fi, ZigBee, adaptive network topology (ANT), radio-frequency (RF) transceivers, cellular, long range (LoRa), and narrowband internet-of-things (NB-IoT) [4], [5]. Each wireless technology has its advantages and trade-offs in terms of power consumption, range, data rate, and implementation complexity. The choice of technology depends on specific requirements, such as the desired communication range, power efficiency, and data transmission characteristics for the TPMS application. Congestion can lead to reduced overall efficiency in the wireless communication process. Devices may spend more time and energy retransmitting data, negotiating communication parameters, or managing collisions, impacting the overall efficiency of power utilization. BLE devices often include power management features, such as low-power states and sleep modes. These features can help minimize power consumption during periods of inactivity or when the device is not actively transmitting or receiving data. Also, BLE, a low-power wireless technology for short-range applications, relies on frequency hopping to switch channels dynamically within the 2.4 GHz band. Adjusting the frequency hopping rate in a BLE system affects power consumption, necessitating careful consideration of trade-offs and application requirements. Implementing TPMS with BLE offers significant advantages, especially in heavy vehicles like multi-axle trucks or trailers with numerous tyres (10 to 36 or more) [6].

In the paper, we proposed a direct TPMS transmitter architecture using BLE on field-programmable gate arrays (FPGA) which sends a small byte of data. By blocking specific frequency channels, the adoptive frequency hopping algorithm in BLE leads to fewer retransmissions and collisions, contributing to lower power consumption. This can result in improved channel quality, reduced transmission errors, and more efficient use of power.

2. LITERATURE SURVEY

In the evolving landscape of automotive technology, TPMS is crucial for vehicle safety and efficiency. Concluding our exploration, we highlight distinctions between two TPMS types: direct and indirect [7]. This comparison sheds light on their functionalities and adoption considerations in modern automotive design. Direct TPMS, using pressure sensors within each tyre, provides real-time data on tyre pressure and temperature. This precise measurement enables immediate detection of variations, offering timely warnings, enhancing road safety, and preventing potential hazards. Notably, direct TPMS excels in providing specific information about individual tyres [8], enabling swift interventions. High-accuracy tyre pressure measurement ensures the detection of subtle changes for proactive hazard prevention.

However, direct TPMS is not without its challenges. The system's reliance on sensors within each tyre introduces additional costs during manufacturing and maintenance. Battery life in these sensors is a consideration, and the need for periodic replacement poses a potential inconvenience for vehicle owners. The initial investment and ongoing maintenance costs must be weighed against the system's benefits, especially in scenarios where budget constraints may be a determining factor for vehicle manufacturers or fleet operators. According to Kuncoro et al. [9], the tyre sensor's maximum current consumption is only 10 mA, and the enTyre current consumption for detecting, processing, and sending RF is expected to be around 40 mA, with a voltage of 4.3V resulting in total power usage of 172 mW [9]. Whereas a tyre pressure monitoring with a system on chip (SoC) from [10] employs a ZigBee transceiver with a supply voltage of 0.8 V and a power consumption of 0.83 mW [10]. In addition, Breglio et al. [11] indicated that the current consumption in active mode is 169 mAh at 3.7 V, with a total power consumption of 625.3 mW [11]. Despite these challenges, the precision and immediacy offered by Direct TPMS make it a preferred choice in various contexts, especially for high-performance vehicles, where accurate tyre pressure readings are of utmost importance.

Indirect TPMS relies on existing sensors and vehicle dynamics to estimate tyre pressure, introducing cost-effectiveness but some level of abstraction. Analyzing wheel speed and other factors alerts drivers to pressure variations. Cost savings come from using pre-existing sensors, reducing the need for extra hardware

and minimizing expenses [12]. Without individual sensors on each tyre, the design is simpler, eliminating the need for periodic replacements. Despite some limitations, like potential false positives, Indirect TPMS offers a pragmatic solution for mass-market and budget-friendly vehicles, aligning with the industry's cost-efficiency goals without compromising safety.

In automobile design, selecting between direct and indirect TPMS requires careful attention. Factors like vehicle use, budget, and desired precision play a crucial role. High-performance vehicles often find direct TPMS justified for optimal handling and safety. Conversely, for everyday commuting, cost-conscious manufacturing, and stable climates, Indirect TPMS emerges as an economical choice. Geographical and environmental context matters; direct TPMS excels in extreme climates, while indirect TPMS cost advantages shine in stable regions. The rise of electric and autonomous vehicles adds complexities, impacting the optimal TPMS choice. Integration into V2X communication networks holds promise for future development, enabling real-time tyre information sharing for collective safety.

3. DESIGN OF TPMS

Tyre failures from underinflation or overheating can lead to accidents, vehicle downtime, and increased costs. The experiment aims to create a system using BLE 5.0 and MicroBlaze to transmit tyre pressure and temperature data to a display device. The FPGA, with MicroBlaze, optimizes the system for power and timing. MicroBlaze collects data every 20 ms, processes it, and transmits it through BLE 5.0. This system integrates various sensors for adaptable tyre monitoring across vehicles, brands, and road conditions, aiming to enhance data accuracy and reliability.

Figure 1 shows a system that uses robust communication protocols for the seamless transmission of real-time tyre data to an external monitoring device or dashboard. This ensures that drivers and relevant stakeholders have immediate access to critical tyre information, enabling proactive responses to tyre-related issues. FPGA technology, with its flexibility and reconfigurability, offers a promising avenue for innovative applications in automotive safety. The motivation behind this research is to harness the potential of FPGA technology, particularly the MicroBlaze softcore internet protocol (IP), to create a high-performance, adaptable, and efficient TPMS [13]. Newer softcore processor designs are closing the performance gap with hard cores, making them increasingly competitive in terms of power efficiency Hardcore processors often have an edge in raw power efficiency at high-performance workloads, while softcore processors offer more flexibility for customized power optimization techniques. When a hard core's chip power is compared to that of a soft core, the soft core has significantly less chip power [14].

Developing a sophisticated TPMS using FPGA technology presents a challenging yet rewarding opportunity for scientific and engineering exploration. The aim is to overcome technical hurdles, contribute to automotive safety and FPGA-based system development, and align with green technology trends. Integrating BLE 5.0 ensures reliable, low-power data transmission, suitable for long-term, energy-constrained applications. BLE's efficiency in active and sleep modes makes it a power-efficient choice [15]. Implementing a frequency hopping algorithm enhances power efficiency by strategically avoiding interference and reducing retransmissions. This optimization aligns with sustainable, environmentally conscious applications.

According to ABI research, by 2021, 48 billion devices will be connected to the internet, with about 30% using Bluetooth technology. BLE is widely used in IoT applications due to its technical advances. Figure 2 illustrates the Bluetooth stack protocol, with the physical layer (PHY) at the bottom. Bluetooth 5.0 introduces two major additions to the physical layer compared to Bluetooth 4: LE M1, LE2M, and LE coded. LE M1 provides support similar to Bluetooth 4, while the newly added LE 2M offers a double data rate (up to 2Ms/s), and LE coded extends the range fourfold by increasing transmission power [16].
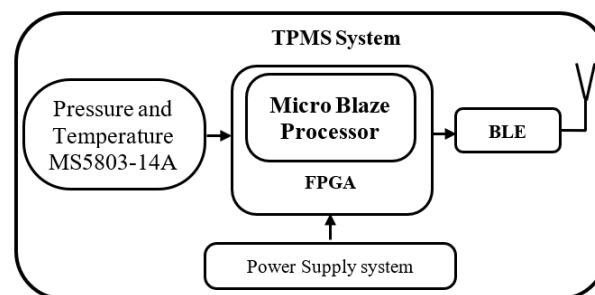

Figure 1. Block diagram of a TPMS transmitter

Figure 2. The Bluetooth low energy protocols

The Bluetooth special interest group (SIG) has initiated an adaptive frequency hopping (AFH) to reduce the interference between devices to improve the quality of the communication of devices in the same environment [17]. The different AFH techniques have their advantages and disadvantages and can be used depending on their necessity. It can also be implemented by different methods. The first generation in the 2.4 GHz band used only 79 channels out of 83.5 available channels in a random manner at 1,600 per second to skip the channel [18]. With the presence of any other device in the same environment, this type of hopping usually leads to occasional conflicts. Since it had no adaptive frequency hopping it was unable to avoid conflicts and cope in the same environment. Figure 3 shows a scenario of a wireless LAN (WLAN) and Bluetooth running without using adaptive frequency hopping [19].

Using the AFH allows the environment to identify the interference source on the fixed channels and then reject these blocked channels from the available channel list. This led to the remapping of the channels with a reduced number of available channels. Bluetooth requires at least 20 channels set. Figure 4 shows an environment where adaptive frequency hopping is used in Bluetooth and wireless LAN is compared [20]. Here we can observe that the AFH algorithm avoids multiple fading effects and avoids interferences.

The BLE operated at 2.4 GHz and in ISM 2.4 to 2.483 GHz. It has 40 RF channels which are 2 MH wide. Every RF channel is assigned with a distinct channel index which is also called a BLE channel [21]. The mapping between the BLE channel and the frequency is shown in Figure 5. Out of the 40 BLE channels, channels 37 to 39 are used for advertising, and the remaining 0 to 36 channels i.e. a total of 37 channels are used as data channels.
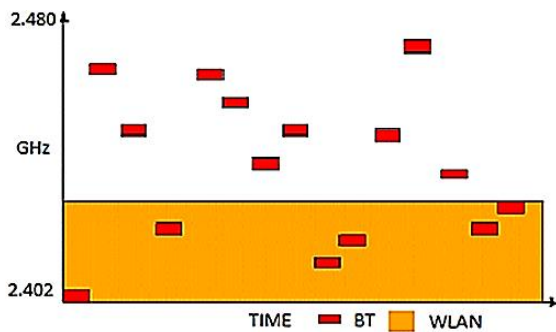

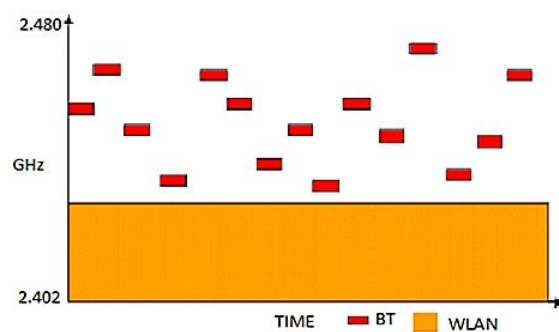Figure 3. WLAN and Bluetooth environment without adaptive frequency hopping

Figure 4. Adaptive frequency hopping is used in Bluetooth and wireless LAN communication
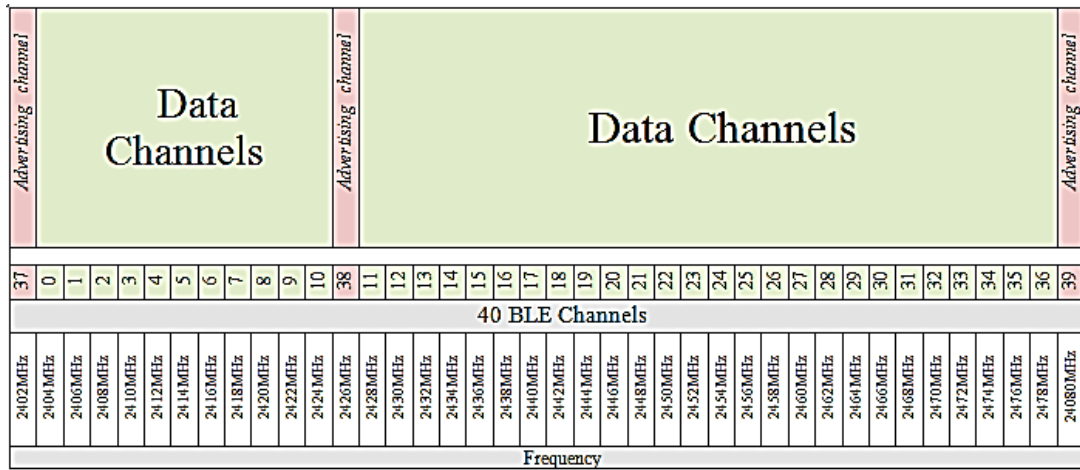
Figure 5. BLE channel and frequency mapping

The advertising channels are distributed across the 2.4 GHz frequency band. Thus, by widely spreading across the spectrum, the interference can be avoided from different devices working in the same spectrum as the WLAN. Exchanging data packets is mainly done on data channels. Whereas the advertising channels are used to connect indication packets, scan respond/request packets, and transmit advertising packets.

3.1. Channel hopping

Channel hopping mitigates interference in wireless communication, following Bluetooth standard rules and algorithms. The unlicensed 2.4 GHz industrial scientific and medical (ISM) spectrum often causes interference, especially with Bluetooth low energy (BLE). BLE's low-energy protocol may result in packet retransmissions to correct errors. Channel hopping provides a straightforward solution, ensuring uninterrupted broadcasting and communication even if a channel faces interference. BLE employs distinct channel hopping frequencies for connected devices and advertising. Figure 5 depicts the cyclic transmission of three advertising channels, starting from indices 37, 38, and 39. This cyclic listening approach is consistently used for initiation and scanning devices [22].

When a BLE connection is established, the device switches channels for each connection event, occurring periodically in a set interval called the connection interval. During a connection event, packets are sent on the same data channel, and a new channel is used for the next event. The Bluetooth core specification employs two-channel selection algorithms to avoid transmission faults. Channel mapping, categorized as good or bad based on specifications like packet error rate (PER) and signal-to-noise ratio (SNR), involves communication between the slave and the master. The master device identifies a good data channel and updates the channel map accordingly. If a bad channel is identified, the standard remapping procedure is executed based on the algorithms.

3.2. Data channel index selection

The master devices link layer categorizes the data channel as either a used channel or an unused channel for a connection. This is called data channel mapping. It uses at least two channels for mapping [23]. The host gives the channel arrangement guidelines to the link layer and the link layer uses this information. The slave gets the channel mapping information from the master and in case of any change in the mapping the master informs the slave as the slave cannot change the channel mapping.

3.2.1. Channel selection Algorithm #1

Algorithm #1 is the initial and compulsory algorithm, whereas Algorithm #2 is only initiated with Bluetooth 5. At the connection time, the master decides the hopping algorithm. Algorithm #1 only supports channel selection for connection events. It has two stages the first stage consists of calculating the unmapped channel index and the second mapping the data channels to this index from the groups of unused channels. The first connection events of a connection start with a 0 and further unmapped channels are mapped using (1) [24].

$$unmappedChannel = (lastUnmappedChannel + hopIncrement) \, mod \, 37 \qquad (1)$$

The last mapped channel is placed in the unmapped channel once the connection event is completed. If the unmapped channel is used already in the mapped channel, then the algorithm uses the channel for the next connection event and in case the channel is unused in channel mapping then the channel is remapped with one of the used channels in the channel map using the following algorithm.

$$remappingIndex = unmappedChannel \bmod numUsedChannels \qquad (2)$$

Figure 6 shows the complete procedure of Algorithm #1. If the unmapped channel is present in the mapped channel, then it is used as the data channel index for the connection event. If it is remapped, then a remapped table of the used channel is created in ascending order starting from zero. Then the remapped index from the remapped table is picked as the data channel index for the connection event.
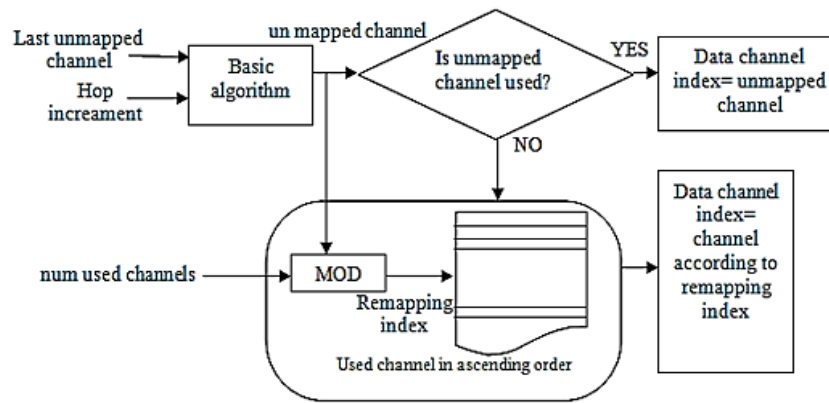

Figure 6. Flow diagram of data channel selection Algorithm #1

3.2.2. Channel selection algorithm #2

Channel Algorithm #2 is utilized for both advertising packets as well as for connection events where it generates an event channel index. Only for a legal device Algorithm #2 is used because it has more randomness. The general block diagram for the algorithm is explained in Figure 7. The 16-bit input counter revises every event that takes place. The data connections use the connection event counter and for periodic advertising, it is the event counter [10].
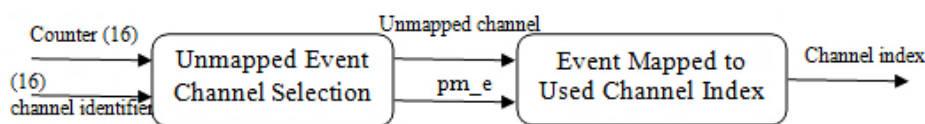

Figure 7. General block diagram of channel selection Algorithm #2

The algorithm consists of a 6-bit input, *'N'* number of channels classified as used channels. For a given connection or either to a periodic advertisement the 16-bit input channel identifier will be unchanging and can be calculated with an assess address as shown in (3).

$$Channel\ identifier = (Access\ Addresses\ 31-16)\ XOR\ (Access\ Addresses\ 15-0) \qquad (3)$$

The 'XOR' operation signifies the 16-bit bit-wise XOR operation. The permutation operation comprises individual bit-reversing the lower 8 input bits and the upper input bits as shown in Figure 8 [10].

The enTyre used channel is arranged in ascending order and assembled in the remapping table which is indexed from zero. The unmapped event channel method consists of two instances. Initially the pseudo-random number 'prn_e' is generated and then the unmapped channel index is derived from 'prn_e'. The unmapped channel is calculated as prn_e modulo 37. If the unmapped channel from the channel map is the channel index from the used channel, then the channel index is used for the event [25]. If its presence

forms the unused channel, then for the event the calculation from prn_e and the N-number of channels is remapped as (4).

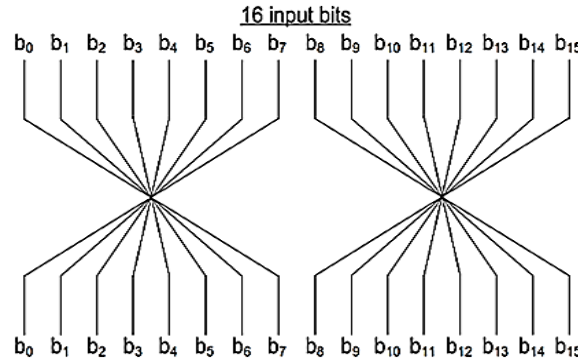$$remapping index = \lfloor ((N * prn_e)/2^{16}) \rfloor \qquad (4)$$

Figure 8. Permutation operation

The channel index for an event is indexed from the remapped table using the remapping index equation mentioned above. The overall procedure for event mapping to use channel index is shown in Figure 9. The actual implementation may vary depending on specific requirements, such as device roles (e.g., advertiser, scanner, initiator, or slave), connection parameters, and the presence of interference in the environment.
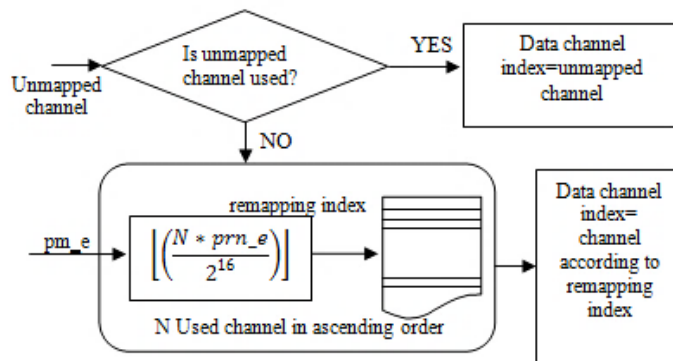

Figure 9. Flow diagram of event mapping to use channel index process

4.   RESULTS AND DISCUSSION
When optimizing power consumption in a BLE 5.0 device using channel selection Algorithms #1 and Algorithms #2 (CSA #1 and #2), the presence of bad channels significantly impacts the performance and power efficiency. Let's compare two scenarios: one with only 8 bad channels and the other with 16 bad channels. Utilizing power-saving modes and sleep periods when the device is not actively transmitting or receiving data helps optimize power consumption. And also ensure that the device transitions to low-power states during idle periods to minimize overall power consumption.

4.1. Low channel interference
Figure 10 shows a graphical user interface to generate the channel hopping for the desired channels. The total number of used channels is from 0 to 36 and the access address is 8E89BED6. Here we can observe that 8 channel numbers 0, 2, 10, 11, 20, 21, 30, and 31 are considered a bad channel, hence these channels are neglected in the mapping [26]. Figures 11 and 12 show the channel hopping pattern where the BLE channel selection can be seen with the increase in the connection event counter. Also, Figures 12 and 13 represent the histogram for 100 connection events of Algorithm #1 and Algorithm #2, respectively. We can see that the

bad channels are neglected. Algorithm #1 channel selection is more predictable; hence prawn for more interference and threat of attacks on the Bluetooth [27]. In Algorithm #2 the channel selection is calculated based on the current event counter and access address which looks semi-random and if a channel is blocked or a bad channel then the remapping algorithm remaps the channel to the next available channel.
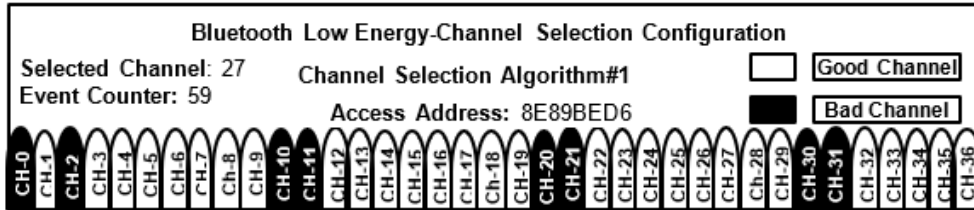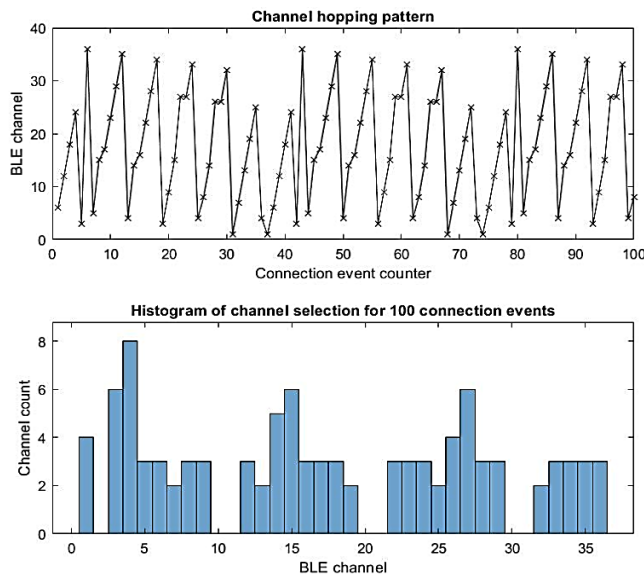

Figure 10. Channel selection configuration

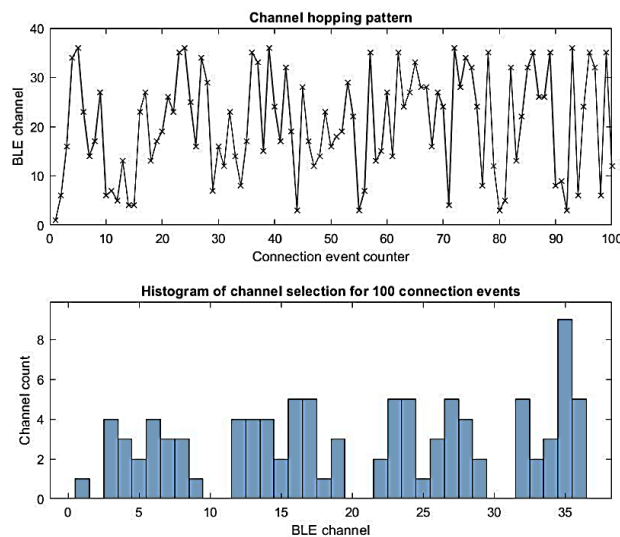Figure 11. Channel hopping pattern for Algorithm #1

Figure 12. Channel hopping pattern for Algorithm #2

4.2. Effect of high radio interference

Figure 13 shows the BLE channel selection configuration where there is a large number of bad or interference channels present. Out of 36 channels, 16 channels are considered bad channels. Bluetooth 5 can transmit a maximum of +20 dBm while Bluetooth 4 specifies around +10 dBm [28]. Thus, it increases the range of the device. Also, Bluetooth 5 has more RF channels than Bluetooth 4. Figure 14 shows channel selection Algorithm #1 as compared to channel selection Algorithm #2 where the channel selection Algorithm #2 has improved the interference tolerance of the Bluetooth radio. Figure 15 shows a condition where there is a high interference environment which caused to limit in the number of RF channels. Despite limiting the RF channels, it avoids congestion and is more resistant to interferences giving more stability to the communication event even in high radio interference [29].
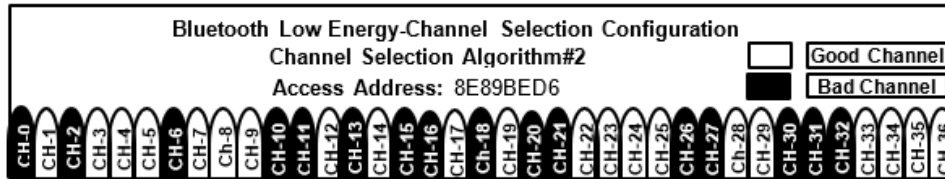

Figure 13. Channel selection configuration with multiple bad channels
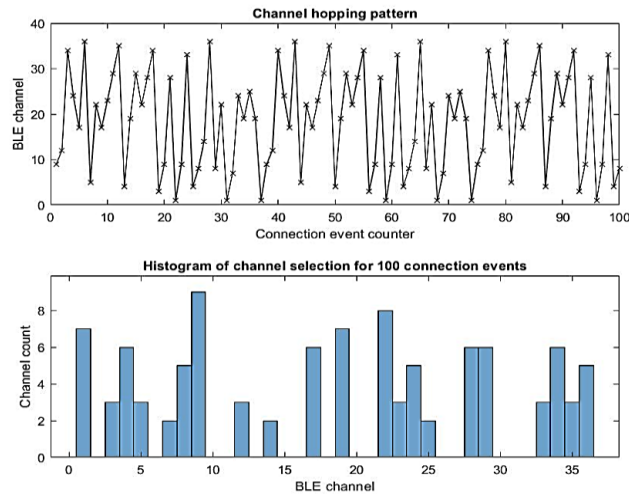

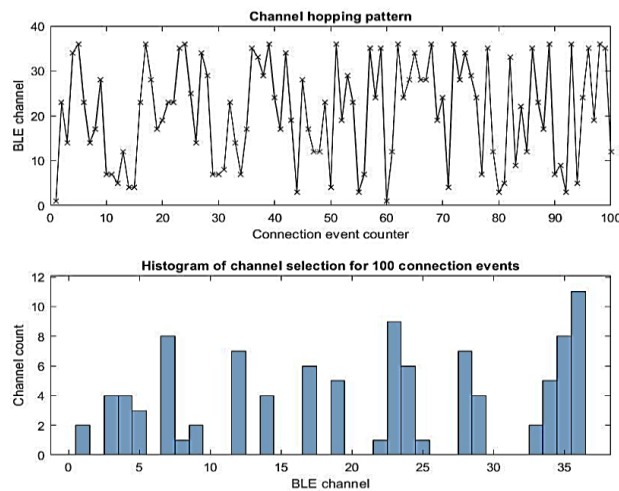Figure 14. Channel hopping pattern for Algorithm #1 with multiple bad channels

Figure 15. Channel hopping pattern for Algorithm #2 with multiple bad channels

When limiting the RF channel of the channel selection Algorithm #2, the adaptability helps in avoiding channels with poor performance and contributes to power optimization. Also, Optimize the timing of channel switches and ensure minimal processing time to conserve power [30]. Longer connection intervals and higher connection latencies also contribute to lower power consumption. In both cases, careful tuning of parameters, continuous monitoring, and adaptive strategies are essential to achieve the best compromise between power efficiency and communication reliability.

4.3. Power consumption

The Xilinx power estimator (XPE) is a key tool in project pre-design and pre-implementation phases, estimating FPGA design power consumption efficiently throughout the design cycle. Considering factors like resource usage, toggle rates, and I/O loading, XPE aids in planning power budgets and thermal management for device safety. By providing early visibility into power consumption, it streamlines design iteration, enabling quick, informed decisions. This accelerates development cycles and potentially shortens time-to-market. XPE verifies FPGA designs against power budget requirements, ensuring compliance with specified power constraints for the target application. The XPE gives the total device power which is calculated as (5).

$$Total\ Devices\ Power\ =\ Device\ Static\ +\ Design\ Static\ +\ Design\ Dynamic \qquad (5)$$

The device static power component denotes the fixed power consumption of the device, including idle states. Design static power refers to the specific design's static consumption, covering power used when components are not actively switching. Design dynamic power represents the power used during dynamic operation, linked to switching frequency. Managing these power components is vital in designing electronic systems, especially for power-efficient applications like mobile devices and IoT. Engineers strive to minimize power consumption while meeting performance needs. Table 1 displays the total on-chip power (66 mW) provided by XPE for the Spartan 6 Automotive device in a TQG144 package.

MicroBlaze offers power management modes, including sleep, hibernate, and suspend (idle) operations. These modes can be leveraged to reduce power consumption when the system is idle or not in use. Active power mode corresponds to the state when the field-programmable gate arrays (FPGA) are fully operational and actively processing data. In this mode, the FPGA executes instructions, performs computations, and engages with its peripherals and interfaces. Active power consumption is generally higher compared to other modes as the FPGA is utilizing its resources to execute tasks. Active power consumption is generally higher compared to other modes as the FPGA is utilizing its resources to execute tasks which can be observed in Table 1.

Table 1. Summary of on-chip power dissipation at a frequency of 100 MHz

	Active		Suspend		Hibernate	
Transceiver	0.000 W	0%	0 W	0%	0 W	0%
I/O	0.000 W	0%	0 W	0%	0 W	0%
Core dynamic	0.0051 W	78%	0 W	0%	0 W	0%
Device static	0.015 W	22%	0.011 W	100%	0 W	0%
Total on-chip power	0.066 W		0.011 W		0 W	

Suspend or idle power mode occurs when the FPGA is not actively processing data or executing instructions. However, it is still powered on and in a standby state, ready to resume active operation upon receiving further instructions. Power consumption during suspend mode is lower than the active mode but higher than the hibernate mode which is around 11 mW. The FPGA is in a low-power state, and it can quickly transition back to the active mode. When the results are compared to [9]–[11], it is clear that the power consumption in a TPMS that employs a softcore, Micro-Blaze on FPGA optimizes the power consumption while also reducing the system latency. In XPE the operating grade selected is industrial which provides a wide temperature range from -400 C to +1000 C. The Xilinx XPE On-chip panel provides detailed information about the power consumption of various on-chip components in an FPGA design.

Table 2 shows the on-chip panel in Xilinx XPE that provides insights into the power consumption of a specific on-chip component, such as a PLL clock, helping designers optimize their FPGA designs for power efficiency. It can be observed that the maximum power consumed by a PLL is 51 mW which is 77% of the total power. Definitely with proper trade-offs and power optimization in these resources a targeted power budget can be achieved.

Table 2. Total on-chip power consumed within a device

Resources		Power	
		(W)	(%)
Core dynamic	Clock	0.001	1
	Logic	0.000	0
	Block random access memory (BRAM)	0.000	0
	Digital signal processing (DSP)	0.000	0
	Digital clock manager (DCM)	0.000	0
	Phase locked loop (PLL)	0.051	77
Device static		0.015	22

5. CONCLUSION

This research paper has delved into the critical realm of optimizing power consumption in TPMS for vehicles and heavy vehicles, particularly trailers. The study focused on leveraging the capabilities of FPGA and innovative strategies like adaptive channel frequency hopping to address the power challenges associated with BLE communication. Adaptive frequency hopping is the most complex in classic Bluetooth. It requires examining the bad channel in real-time and then revising the hopping table and hence it requires the maximum resources of a microcontroller. In Bluetooth 5 the BLE used the newly introduced channel selection Algorithm #2 (CSA #2). In this, the frequency hopping is computed by the algorithm such that it avoids channel interference and also reduces the multipath fading effects. Through meticulous experimentation and analysis, we demonstrated that selectively blocking BLE channel numbers 0, 2, 10, 11, 20, 21, 30, and 31, coupled with the implementation of adaptive frequency hopping on FPGA results in a significant reduction in power consumption for TPMS, especially in heavy vehicles with multiple tyres. The FPGA played a pivotal role in enabling efficient optimization strategies, allowing for tailored channel selection and minimizing power-hungry operations. The findings underscore the importance of adapting frequency hopping strategies to the unique characteristics of heavy vehicles, where the number of tyres contributes to channel congestion. By strategically blocking certain channels, we mitigated interference, reducing the need for energy-intensive retransmissions and collisions. Our research not only contributes to the field of TPMS optimization but also addresses a critical aspect of automotive safety for heavy vehicles. The power-efficient strategies presented in this paper have the potential to enhance the overall reliability and longevity of TPMS in real-world, demanding applications.

REFERENCES

[1] Q. Xin, G. Jingfeng, G. Junjie, B. Ri, Y. Mingxing, and Z. Pian, "Automobile tire pressure monitoring technology and development trend," *Journal of Physics: Conference Series*, vol. 1314, no. 1, Oct. 2019, doi: 10.1088/1742-6596/1314/1/012100.

[2] L. M. Silalahi, M. Alaydrus, A. D. Rochendi, and M. Muhtar, "Design of tire pressure monitoring system using a pressure sensor base," *Sinergi*, vol. 23, no. 1, Feb. 2019, doi: 10.22441/sinergi.2019.1.010.

[3] M. A. Hassan, M. A. A. Abdelkareem, M. M. Moheyeldein, A. Elagouz, and G. Tan, "Advanced study of tire characteristics and their influence on vehicle lateral stability and untripped rollover threshold," *Alexandria Engineering Journal*, vol. 59, no. 3, pp. 1613–1628, Jun. 2020, doi: 10.1016/j.aej.2020.04.008.

[4] R. Mubashar, M. A. B. Siddique, A. U. Rehman, A. Asad, and A. Rasool, "Comparative performance analysis of short-range wireless protocols for wireless personal area network," *Iran Journal of Computer Science*, vol. 4, no. 3, pp. 201–210, Apr. 2021, doi: 10.1007/s42044-021-00087-1.

[5] R. R. Patil and A. A. Shinde, "Wireless tyre pressure measurement," *Indian Journal of Science and Technology*, vol. 14, no. 9, pp. 842–849, Mar. 2021, doi: 10.17485/ijst/v14i9.2296.

[6] A. Elhadeedy and J. Daily, "60 GHz Wi-Fi as a tractor-trailer wireless harness," in *2023 IEEE 13th Annual Computing and Communication Workshop and Conference, CCWC 2023*, Mar. 2023, pp. 1023–1028, doi: 10.1109/CCWC57344.2023.10099274.

[7] A. B. Muturatnam, N. V. Sridharan, A. P. Sreelatha, and S. Vaithiyanathan, "Enhanced tyre pressure monitoring system for nitrogen filled tyres using deep learning," *Machines*, vol. 11, no. 4, Art. no. 434, Mar. 2023, doi: 10.3390/machines11040434.

[8] B. Szczucka-Lasota, T. Węgrzyn, B. Łazarz, and J. A. Kamińska, "Tire pressure remote monitoring system reducing the rubber waste," *Transportation Research Part D: Transport and Environment*, vol. 98, Sep. 2021, doi: 10.1016/j.trd.2021.102987.

[9] C. B. D. Kuncoro, M.-F. Sung, C. Adristi, A. F. Permana, and Y.-D. Kuan, "Prospective powering strategy development for intelligent-tire sensor power charger application," *Electronics*, vol. 10, no. 12, Jun. 2021, doi: 10.3390/electronics10121424.

[10] A. Vasantharaj, N. Nandhagopal, S. A. Karuppusamy, and K. Subramaniam, "An in-tire-pressure monitoring SoC using FBAR resonator-based ZigBee transceiver and deep learning models," *Microprocessors and Microsystems*, vol. 95, Art. no. 104709, Nov. 2022, doi: 10.1016/j.micpro.2022.104709.

[11] G. Breglio *et al.*, "Development and testing of a low-cost wireless monitoring system for an intelligent tire," *Machines*, vol. 7, no. 3, Jul. 2019, doi: 10.3390/machines7030049.

[12] B. Huo, M. Zhang, S. Zhong, and F. Zhou, "Design of indirect tire pressure monitoring system based on wheel speed signal," *Journal of Physics: Conference Series*, vol. 2528, no. 1, p. 12045, Jul. 2023, doi: 10.1088/1742-6596/2528/1/012045.

[13] P. Kalkundri, H. Guhilot, and K. Ravi, "Soft embedded cores for FPGA," in *Lecture Notes in Networks and Systems*, vol. 321, Springer Nature Singapore, 2022, pp. 895–905.

[14] X. Yan, K. Huang, and J. Meng, "SoC design," in *Handbook of Integrated Circuit Industry*, Springer Nature Singapore, 2024, pp. 803–819.

[15] M. A. Al-Shareeda, M. A. Saare, S. Manickam, and S. Karuppayah, "Bluetooth low energy for internet of things: review, challenges, and open issues," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 31, no. 2,

pp. 1182–1189, Aug. 2023, doi: 10.11591/ijeecs.v31.i2.pp1182-1189.

[16] A. Aza, D. Melendi, R. García, X. G. Pañeda, L. Pozueco, and V. Corcoba, "Bluetooth 5 performance analysis for inter-vehicular communications," *Wireless Networks*, vol. 28, no. 1, pp. 137–159, Nov. 2022, doi: 10.1007/s11276-021-02830-9.

[17] M. Baert, J. Rossey, A. Shahid, and J. Hoebeke, "The Bluetooth mesh standard: an overview and experimental evaluation," *Sensors*, vol. 18, no. 8, Jul. 2018, doi: 10.3390/s18082409.

[18] G. Shan, B. U. Lee, S. H. Shin, and B. H. Roh, "Design and implementation of simulator for analysis of BLE broadcast signal collision," in *International Conference on Information Networking*, 2017, pp. 448–452, doi: 10.1109/ICOIN.2017.7899533.

[19] A. Shintani, "The design, testing, and analysis of a constant jammer for the Bluetooth low energy (BLE) wireless communication protocol," Robert E. Kennedy Library, Cal Poly.

[20] A. A. Eltholth, "Improved spectrum coexistence in 2.4 GHz ISM band using optimized chaotic frequency hopping for Wi-Fi and Bluetooth signals," *Sensors*, vol. 23, no. 11, May 2023, doi: 10.3390/s23115183.

[21] I. Natgunanathan, N. Fernando, S. W. Loke, and C. Weerasuriya, "Bluetooth low energy mesh: applications, considerations and current state-of-the-art," *Sensors*, vol. 23, no. 4, p. 1826, Feb. 2023, doi: 10.3390/s23041826.

[22] M. Afaneh, "Introduction," in *Intro to Bluetooth low energy*, Fishers Novel Bits, 2018.

[23] B.-Z. Pang, T. Claeys, D. Pissoort, H. Hallez, and J. Boydens, "Comparative study on AFH techniques in different interference environments," in *2019 IEEE XXVIII International Scientific Conference Electronics (ET)*, Sep. 2019, pp. 1–4, doi: 10.1109/ET.2019.8878594.

[24] *Covered core package Version: 5.0 [Low Energy Controller volume]*. Bluetooth SIG Proprietary, 2016.

[25] M. Woolley, "Bluetooth® Core specification version 5.0 feature enhancements," *Bluetooth*, 2021. Accessed Jan. 18, 2024. [Online], Available: https://www.bluetooth.com/wp-content/uploads/2019/03/Bluetooth_5-FINAL.pdf

[26] MathWorks, "Noncollaborative Bluetooth LE coexistence with WLAN signal interference," *Mathworks*, 2020. https://in.mathworks.com/help/bluetooth/ug/noncollaborative-bluetooth-le-coexistence-with-wlan-signal-interference.html (accessed Jan. 18, 2024).

[27] D. Acharya, "Performance measurements of Bluetooth 5 technique under interference," M.S. thesis, Degree Program in Wireless Communications Engineering, University of Oulu, 2019.

[28] "Bluetooth®5, Refined for the IoT," *Silicon Labs*. https://www.silabs.com/whitepapers/bluetooth-5-refined-for-the-iot (accessed Jan. 18, 2024).

[29] J. A. Afonso, A. J. F. Maio, and R. Simoes, "Performance evaluation of Bluetooth low energy for high data rate body area networks," *Wireless Personal Communications*, vol. 90, no. 1, pp. 121–141, Apr. 2016, doi: 10.1007/s11277-016-3335-4.

[30] Microchip Technology, "ISM band coexistence whitepaper," *Kleer*, 2007. Accessed Jan. 18, 2024. [Online], Available: https://ww1.microchip.com/downloads/en/DeviceDoc/Kleer_ISM_Coexistence.pdf

BIOGRAPHIES OF AUTHORS

Praveen Uday Kalkundri [ID] [img] [SC] [img] completed his Bachelor of Engineering in electronics and communication engineering in 2009 from Gogte Institute of Technology, Belgaum. He has industry experience in roles such as software engineer and senior software engineer. In 2011, he pursued a master's in technology in VLSI design in embedded systems from KLE Dr. M.S. Sheshgiri College of Engineering and Technology, Belgaum. Currently, Praveen is an assistant professor in the Department of Electronics and Communication Engineering at Gogte Institute of Technology and is pursuing his Ph.D. in VLSI design. He is a life member of the Indian Society for Technical Education. He can be contacted at email: pukalkundri@git.edu.

Veena Desai [ID] [img] [SC] [img] has been working at KLS, Gogte Institute of Technology, Belgaum since 1991. She has served as head of the Department of Electronics and Communication Engineering and Computer Science Engineering during this tenure. Dr. Veena holds a PhD in Electronics (VTU), a master's in computer network engineering (with Honors) and a bachelor's degree in electronics and communication engineering (with Distinction). She also holds a post graduate diploma in Cyberlaw from the Asian School of Cyber Laws, Mumbai University and has completed a Certificate course in IPR from WIPO. Dr. Veena has over 28 years of teaching experience and is involved in Guiding research scholars and teaching UG and PG students, her major areas of research include computational intelligence applications, network security, biomedical signal processing and agriculture issues. She can be contacted at email: veenadesai@git.edu.

Ravi Uday Kalkundri [ID] [img] [SC] [img] completed his Bachelor of Engineering in computer science and engineering and completed in 2009 from Gogte Institute of Technology, Belgaum. He worked in industries at different positions like software engineer and senior software engineer. In 2011, he went on to pursue a master's in technology in computer science and engineering, from Gogte Institute of Technology, Belgaum. Currently, he is working as an assistant professor at the Gogte Institute of Technology in the Department of Computer Science and Engineering, and currently pursuing his Ph.D. in the field of network security. His research interests are in the area of ad hoc networks specializing in the area of VANETS and security. He is a life member of the Indian Society for Technical Education. He can be contacted at email: ravi.kalkundri@gmail.com.