Quantifying the Trade-off Between Latency and Power Consumption in Bluetooth Low Energy and Its Mitigation by Using a Wake-Up Receiver

Maryam Dodangeh
IC-Design group
University of Twente
Enschede, The Netherlands
m.dodangeh@utwente.nl

Mark S. Oude Alink
IC-Design group
University of Twente
Enschede, The Netherlands
m.s.oudealink@utwente.nl

Bram Nauta
IC-Design group
University of Twente
Enschede, The Netherlands
b.nauta@utwente.nl

***Abstract*—Bluetooth Low Energy (BLE) is one of the primary wireless communication protocols for Internet of Things (IoT) devices due to its inherently low energy consumption. BLE's duty-cycled scheme reduces energy consumption, but at the cost of latency. Wake-Up Receivers (WuRXs) have been proposed to mitigate this trade-off, but most attention seems to have been paid to the circuit design rather than assessing exactly how beneficial a WuRX could be. In this paper, we analyze the power consumption and latency impact of a 200μW WuRX added to a state-of-the-art commercial BLE transceiver for relevant IoT scenarios. The results show that the latency/power trade-off can be significantly relaxed for both peripheral and central devices in initiating a connection. Furthermore, if the role of central and peripherals can be changed (peripherals scan instead of advertise), the power consumption of the most energy-constrained devices can be further reduced in scenarios that require less than 200ms latency to initiate a connection. Also, for a maximum latency of 4s, a duty-cycled WuRX enables equal 60μW average power consumption for both the central and peripheral devices. This is extremely useful when both ends of the link face similar battery constraints, as would be the case in many IoT scenarios.**

***Index Terms*—Bluetooth Low Energy (BLE), Internet of Things (IoT), latency, power consumption, ultra-low power, Wake-Up Receiver (WuRX), Wireless Sensor Network (WSN)**

I. Introduction

In the upcoming decade, about 1 trillion Internet of Things (IoT) nodes are expected to be connected worldwide [1], forming wireless sensor networks for health and environmental monitoring, smart grids and smart homes, and industrial automation [2]. These nodes are often powered by coin cells which are hard and expensive to replace, requiring system and circuit design innovations to minimize power consumption for maximum battery life.

Wireless communication is often the most power-hungry part. Bluetooth Low Energy (BLE) is widely used in IoT scenarios due to its inherent low power consumption, and can further reduce power consumption by using a duty-cycled scheme [4]. The receiving end sleeps (consuming 1–10μW), but wakes up periodically to scan for incoming packets from the transmitting side (consuming 1–10mW). The average power consumption P_{avg} is then directly determined by how often the device wakes up and how long it stays on each time, posing a direct trade-off with latency. A similar trade-off exists in WiFi, where recently a WuRX concept has been standardized to break this trade-off [3]. Fig. 1 shows that by monitoring the RF environment and waking up the higher-power Main Receiver (MRX) only upon the detection of a certain event, such as a specific wake-up packet, average power consumption can be reduced dramatically in WiFi [3]. However, power reduction in BLE is different from WiFi as it is already optimized for low power rather than high throughput. As a result, the effect of a WuRX in BLE requires its own thorough examination.

This project is financed by Dialog, a Renesas Company, and Holland High Tech in a public-private-cooperation for research and development in the Hightech Systems & Materials topsector.

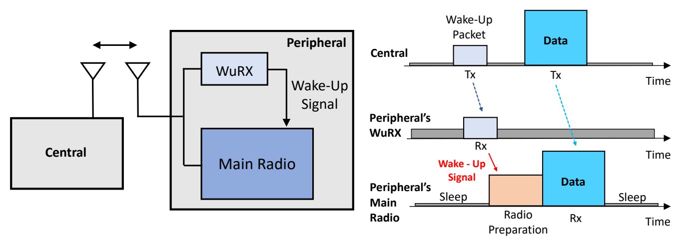


Fig. 1. Using a WuRX to trigger the main radio, based on [3].

A WuRX ideally has the same sensitivity as the MRX to avoid performance degradation, but may consume significantly less power by reducing the required Signal-to-Noise Ratio (SNR) at the antenna port. This reduced SNR-requirement can be achieved in two ways. One is by using back-channel communication, which means using the standard-compliant signals in a way that simplifies demodulation, e.g. bit or packet repetition [5], energy detection [1], [6], multiple channels reception and using techniques such as majority voting [7], [8]. The other is by changing the modulation into a simpler one which requires less power for demodulation such as on-off-keying (OOK) or pulse width modulation (PWM) [9]–[11].

Suggestions on how to physically implement such a WuRX abound in both BLE and WiFi [5]–[19], but little attention has been paid to the two more fundamental questions: What is P_{avg} for BLE devices for certain latency requirements and how much could a WuRX improve these numbers? Several sources tackle bits and pieces of the first question, e.g. some only explain the power profile of either central or peripheral in only one specific scenario [20]–[25], some examine the discovery latency and suggest protocols to reduce it [26], [27], and some suggest a mathematical model [28] or a framework [29] to measure the current consumption while some reduce it by optimizing the related parameters [30]. The second question has not been tackled at all yet. In this paper, we fill these gaps by thoroughly analyzing average power consumption and latency in BLE, their mutual interdependence, and the effect that a WuRX can have on these parameters.

This paper is organized as follows. Section II gives a brief overview of the relevant parts of the BLE standard. BLE's power consumption and latency for its different states are analyzed in section III, using data from a state-of-the-art BLE transceiver. Section IV examines the potential impact of always-on and duty-cycled WuRXs on these numbers. Section V applies the results to some relevant user scenarios, showing when a WuRX may or may not be beneficial and suggesting some BLE extensions to further improve power-efficiency in WuRX-enabled transceivers. Finally, the conclusions are drawn in section VI.

II. Relevant Specifications of the BLE Protocol

Here we will summarize only the parts of the full BLE (Bluetooth 5.2) protocol specification relevant for this paper [31]. BLE uses 40 RF channels (numbered 0–39) on a 2MHz grid from 2402MHz to 2480MHz. Channels 0–36 are used for data transfer during a connection, at a maximum data rate of 2Mbps using GFSK-modulation. Channels 37 (2402MHz), 38 (2426MHz), and 39 (2480MHz) are advertising channels (ACs) to initiate a connection, chosen to minimize interference from the main WiFi channels 1, 6, and 11. Sensitivity, defined as a packet error rate (PER) of 30.8%, with any bit error resulting in a packet error, should be at least -75dBm.

To establish a connection, the *peripheral* device sends advertising packets (APs) on any subset of the three ACs, one after the other. APs (with length L_{AP} of 8–37 bytes including fixed parts like preamble and access address) are sent at an "advertising interval" (T_{AI}) of 20ms to 10.24s, configurable in 625μs increments, plus a pseudo-random delay δ of 0–10ms to reduce collisions of advertisements from multiple devices. A larger T_{AI} leads to lower P_{avg} for the peripheral, but higher latency in initiating the connection. The *central* device scans the ACs, typically in round-robin fashion, for APs during the "scan window" (SW) with duration T_{SW} at a certain programmable "scan interval" (SI) with duration T_{SI} (10ms–10.24s), see Fig. 2. A larger ratio of T_{SW}/T_{SI} leads to higher P_{avg} for the central as the MRX is on for a higher fraction of time, but results in a lower latency to initiate the connection. A larger T_{SI} for the same T_{SW}/T_{SI} ratio also improves P_{avg} as the overhead of waking up the MRX and putting it to sleep is reduced, but it increases the latency.

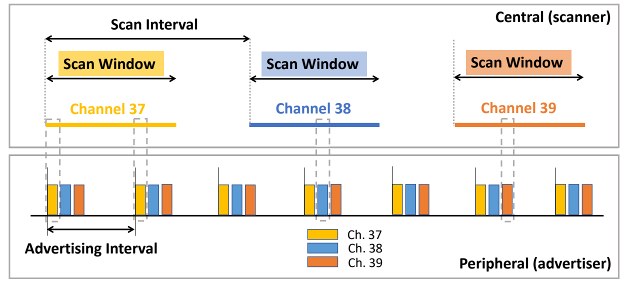


Fig. 2. BLE advertising and scanning mechanisms, based on [4].

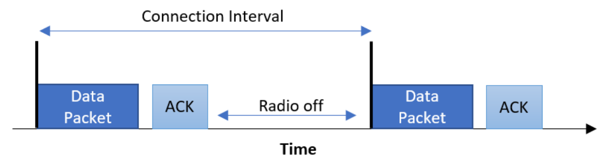


Fig. 3. BLE data communication in the connected state.

The combination of peripheral and central settings determines the actual P_{avg} and latency, and is typically preset per application, as detailed in section V. The actual latency is stochastic and depends on channel conditions, but for simplicity, we only consider the maximum latency under the assumption of BER=0%. In practice, when a BLE device is turned on, it goes to a "fast connection" state and advertises often (peripheral) or monitors the channel continuously (central) for a short time of typically several seconds. After that, it switches to a "low power" state with reduced T_{AI}, T_{SI} and/or T_{SW} [32].

When an AP is received, a connection request packet (CRP) is sent back on the same channel after a 150μs "inter-frame space" and a connection can be established. In the resulting connected state, a programmable "connection interval" (T_{CI}) (7.5ms to 4s in 1.25ms increments) defines connection events (CEs), during which at least 1 data packet (DP) needs to be transmitted, even if there is no data, see Fig. 3. DPs (with length L_{DP} of 6–257 bytes) have a 6-byte fixed part (including e.g. preamble and address) and a configurable data length of 0–251 bytes. Successful reception must be directly acknowledged; otherwise the DP needs to be retransmitted [4]. The central needs to wake up every T_{CI}, while the peripheral can skip a programmable number of consecutive CEs called "slave latency" (SL) without dropping a connection to stay asleep longer. Clearly, the BLE protocol is tailored towards minimizing the power consumption of the peripherals.

III. Power Consumption and Latency in BLE

Whereas latency is determined by the standard and its configuration settings, the power consumption is highly dependent on the specific hardware implementation. Table I lists the performance of several recently published 'experimental' low-

TABLE I
PERFORMANCE SUMMARY OF STATE-OF-THE-ART BLE TRANSCEIVERS.

	[33]	[34]	[35]	[36]	[37]	[38]	[39]	[40]
Type*	E	E	E	E	C	C	C	C
Tech. (nm)	28	65	40	22	55	-	-	-
Sens. (dBm)	-95	-94	-94	-96.4	-94	-97.5	-99	-104
Supply (V)	1	1	0.8	0.5	3	3	3	3
RX (mW)	2.75	2.3	5.3	1.9	6.6	7.8	7.5	21.9
TX** (mW)	3.7	5	8.2	7.8	10.5	9.6	10.2	23.7

* E: experimental, C: commercially available
** at 0dBm output power

power BLE transceivers [33]–[36], as well as the lowest power commercially available BLE transceivers [37]–[40]. Although the experimental devices outperform the commercial ones by about 3x in power consumption, they lack the completeness and robustness required for commercial use. Moreover, to calculate P_{avg}, more detailed information is needed than generally provided. The numbers in this paper have been provided to us by Dialog Semiconductor which has the lowest-power receiver of the commercial devices [37]. Therefore, we use the information from Dialog's BLE14531 device for illustrations and calculations of power consumption. A similar but publicly available set of numbers is available from Texas Instruments for an earlier-generation BLE transceiver CC2640/CC2642 [41].

This BLE device goes through several states in advertising, scanning and connected states, defined as:

- S_1 The wake-up state: a micro controller unit (MCU) wakes up from sleep and intializes the system, such as enabling the crystal oscillator and phase-locked loops (PLLs);
- S_2 The radio preparation state: the PLL is locking to the desired RF frequency and the MCU prepares the radio for transmitting or receiving packets;
- S_3 The Tx state: a peripheral sends an AP, a central transmits a CRP, or both devices transmit data in the connected state;
- S_4 Tx to Rx transition: the PLL needs to be locked to the proper channel or frequency;
- S_5 The Rx state: a peripheral scans an AC for a CRP, a central scans an AC for an AP, or both devices receive data packets in the connected state;
- S_6 Rx to Tx transition or inter-channel space where the PLL locks to the desired channel;
- S_7 Inter-frame space between two consecutive packets on the same channel in the connected state;
- S_8 The post-processing state: the MCU processes received packets and sets a timer to wake up for the next event;
- S_9 Sleep state: the radio is off, waiting to wake up again at the beginning of the next event.

The duration of state S_n is T_n, which may be defined by the protocol, its configurable settings, the amount of data, the specific hardware implementation, or a combination thereof. For example, T_5 is partly determined by the sleep-clock accuracy (SCA). A lower SCA typically means lower power in sleep mode, but requires an increased T_5 to guarantee conformance to the timing. The AI, SI or CI only impacts T_9. The states are (almost) identical for all of the other devices listed in Table I, and are therefore a good representation of general BLE device operation.

To make the analyses tractable, we use the following assumptions throughout the paper, which will all be properly substantiated further in this manuscript:

- WuRX consumes 200μW (active) and 0.1μW (sleep);
- MRX consumes 6.6mW (active);
- Transmitter (TX) consumes 10.5mW (active);
- Transceiver (MRX and TX) consumes 3μW (sleep);
- SNR and interference conditions result in negligible BER, so no retransmissions are necessary;
- WuRX sensitivity is equal to MRX sensitivity;
- Data transmission uses 1Mbps for 1μs per symbol;
- An AP has length of 27 bytes (L_{AP} = 27 bytes) and is transmitted on all the 3 channels;
- A DP has length of 6 bytes (L_{DP} = 6 bytes) as mainly empty packets with no data are transmitted to keep the connection alive;
- Exactly one data packet is transmitted in each connection interval;
- A WuRX adds 5ms latency to generate a wake-up signal;

A. Power Consumption

The average current I_{avg} and power consumption P_{avg} can be calculated as

$$I_{\text{avg}} = \frac{1}{T}\int_0^T I_{\text{BAT}}(t)\mathrm{d}t, \qquad P_{\text{avg}} = V_{\text{BAT}} I_{\text{avg}}, \qquad (1)$$

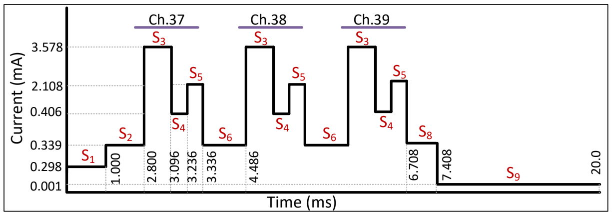
where T is equal to T_{SI} (scanning state), T_{AI} (advertising state), or T_{CI} (connected state), I_{BAT} is the current drawn from the battery, and V_{BAT} is the battery voltage of typically 3V for a small coin cell as often used in low-power devices.

While advertising, see Fig. 4a, the peripheral device wakes up, transmits an AP on a certain AC, scans that AC to potentially receive a CRP, and repeats it on all three ACs (as assumed here). During scanning, see Fig. 4b, the central device wakes up every T_{SI}, scans one AC for T_{SW}, and returns to sleep state.

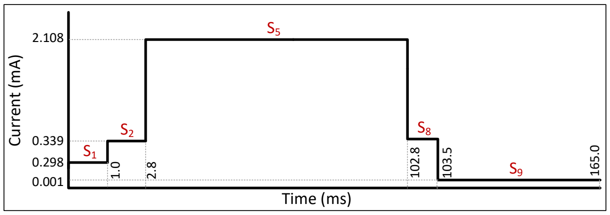
Once connected, both central and peripheral can send DPs. Assuming one DP of 6 bytes (L_{DP} = 6 bytes) for the peripheral, its current profile is shown in Fig. 4c.

B. Latency

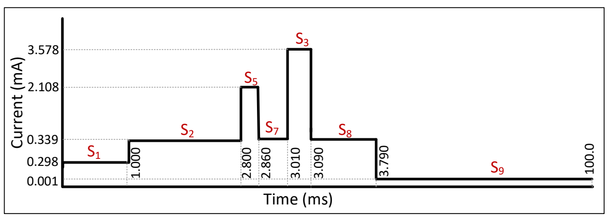
In the connected state, the central and peripheral can only transfer packets at the beginning of each CI. As a result, the maximum latency L_C in the connected state (denoted as $L_{C,\text{max}}$) is at most equal to T_{CI}. When not connected, a peripheral may start advertising when it has new data to initiate



(a) Peripheral advertising (L_{AP} = 27 bytes) at $T_{\text{AI}} = 20$ms: $P_{\text{avg}} = 865\mu$W.



(b) Central scanning ($T_{\text{SW}} = 100$ms and $T_{\text{SI}} = 165$ms): $P_{\text{avg}} = 3.82$mW.



(c) Connected peripheral transmitting (L_{DP} = 6 bytes, $T_{\text{CI}} = 100$ms): $P_{\text{avg}} = 51\mu$W.

Fig. 4. Battery current consumption in different states (numbers from [37]).

a connection. We define the latency L_D as the time difference between the start of the first AP and the end of the successful CRP reception as latency in initiating a connection.

Using Fig. 4a, $L_D \geq 3.336 - 2.800 = 0.536$ms, but $L_{D,\text{max}}$ will depend on both scanning and advertising parameters. When $T_{\text{AI}} < T_{\text{SW}}$, $L_{D,\text{max}}$ occurs when the peripheral starts transmitting the first AP right after the end of the SW, resulting in [42]:

$$L_{D,\text{max}} = \left\lceil \frac{T_{\text{SI}} - T_{\text{SW}}}{T_{\text{AI}}} \right\rceil T_{\text{AI}} + \delta_{\text{max}} \qquad (2)$$

with $\delta_{\text{max}} = 10$ms. When $T_{\text{AI}} \geq T_{\text{SW}}$, $L_{D,\text{max}}$ is difficult to calculate (see [42] for elaborate derivations).

Several suitable combinations of T_{SW}, T_{SI}, and T_{AI} have been defined for initiating a connection in practical scenarios, see Table II. These are referred to as BLE Discovery Processes (DiPs) and are often used in Bluetooth Application Programming Interfaces (APIs) [43]. For all DiPs $T_{\text{SW}} \geq 5T_{\text{AI}}$ to make it highly likely that at least one AP is properly received under realistic interference and SNR conditions in the first SW. For example, if a latency of no more than 90ms is allowed for a certain latency-critical scenario, the values of $T_{\text{SW}} = 100$ms, $T_{\text{SI}} = 165$ms, and $T_{\text{AI}} = 20$ms can be chosen. Our simulated $L_{D,\text{max}}$ in Table II were obtained through a 1000x repeated Monte Carlo simulation, and are seen to match (2). For the last row that $T_{\text{AI}} \geq T_{\text{SW}}$, $L_{D,\text{max}}$ is only derived through simulations.

TABLE II
PREDEFINED DIPS AND THEIR MAXIMUM LATENCIES.

Discovery Process (DiP)	T_{SW} [ms]	T_{SI} [ms]	T_{AI} [ms]	$L_{D,\text{max}}$ (sim)	$L_{D,\text{max}}$ eq. (2)
ultra-low latency (ULL)	4096	4096	20	20ms	10ms
low latency (LL)	100	165	20	90ms	90ms
balanced (BAL)	1024	4096	200	3.2s	3.2s
low power (LP)	512	5120	100	4.7s	4.7s
ultra-low power (ULP)	11.25	1280	1000	10min	-

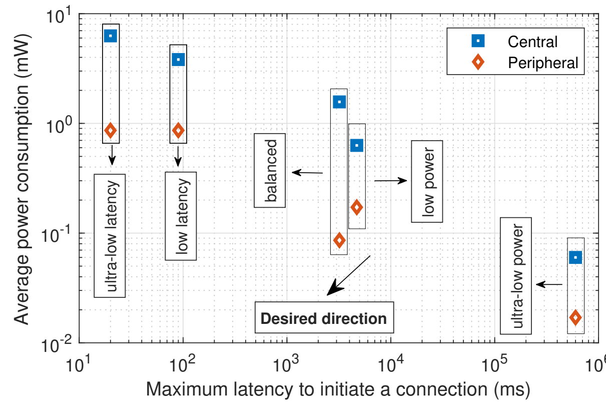


Fig. 5. P_{avg} vs. maximum latency to initiate a connection (data from [37]).

C. *Power consumption vs. latency in existing BLE devices*

Using Fig. 4a and 4b and (1), P_{avg} vs. $L_{D,\text{max}}$ is plotted for both central and peripheral in Fig. 5 for initiating a connection, showing that a central always has higher P_{avg} than a peripheral for these settings. Lower-latency scenarios increase P_{avg} for both peripheral and central devices due to more frequent scanning and advertising, showing the direct trade-off between power and latency. By decreasing both T_{SW} and T_{AI}, it is possible (to some extent) to reduce P_{avg} of a central without compromising on latency, but at the cost of increased P_{avg} of the peripheral.

Using Fig. 4c and (1), P_{avg} vs. T_{CI} (latency) for minimum and maximum length DPs is shown in Fig. 6, assuming only one packet in each CI. P_{avg} is (again) inversely proportional to CI until it starts to level out due to the limited SCA. For $L_{\text{DP}} = 257$ bytes, the power consumption is increased by 5.8x compared to $L_{\text{DP}} = 6$ bytes (i.e. only fixed header, no data payload) for small T_{CI}, but only by 2.1x at maximum T_{CI}.

IV. THE EFFECT OF USING A WURX ON POWER CONSUMPTION AND LATENCY

From the previous sections, it is clear that a reduced power consumption comes at the cost of latency. A WuRX could mitigate that trade-off if it can reduce P_{avg} at similar latency and sensitivity. The improvement in P_{avg} depends mainly on the ratio of WuRX and MRX active power. Additionally, a

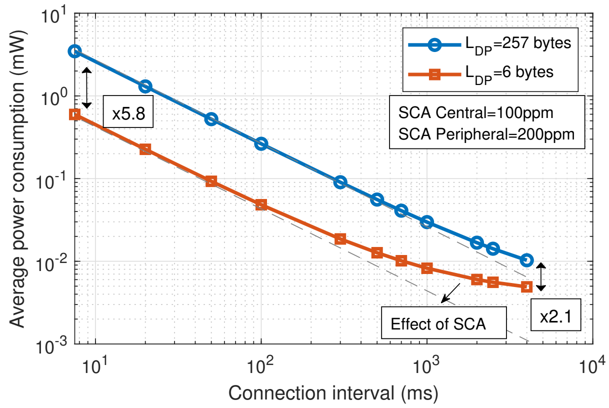


Fig. 6. P_{avg} of peripheral vs. CI (latency) in connected state for minimum and maximum data length (data from [37]).

TABLE III
SUMMARY OF STATE-OF-THE-ART BLE-COMPLIANT WURXS.

	[5]	[6]	[7]	[15]
CMOS Tech. (nm)	65	90	65	65
Sensitivity (dBm)	-80	-58	-57.5	-85
Power (μW)	230	164	150	220
Wake-up latency (ms)	0.2	N/A	N/A	N/A

WuRX will add some additional latency, as it takes time to process incoming information and wake up the MRX (discussed in section IV-B).

There have been quite a few 'experimental' publications regarding WuRXs recently, e.g. [5]–[19]. Standard-compliant WuRXs for BLE [5]–[7], [15] and Wi-Fi [16]–[19] (can) share the existing antenna, which is a must for size and cost reasons. Here we focus only on the BLE-compliant WuRXs [5]–[7], [15], see Table III.

One must pay extra attention to interference rejection while designing a WuRX as a high number of false wake-ups in the crowded 2.4GHz band will degrade the benefits of a WuRX. Based on Table III, we surmise that a power of 200μW is a realistic number in the near future for a WuRX at around -90dBm sensitivity.

A. Connected state

As shown in Fig. 6, for low throughput (L_{DP} = 6 bytes), which is often the case in IoT and BLE scenarios, $P_{\text{avg}} < 200\mu\text{W}$ for latencies above 20ms. Thus, an always-on 200μW WuRX is not beneficial once devices are connected. However, $P_{\text{avg}} \gg 200\mu\text{W}$ for both centrals and peripherals when they try to set up a connection, showing a potential benefit of using a WuRX. That is why the focus will be mainly on the effect of an always-on or duty-cycled WuRX on the power-hungry process of initiating a connection.

B. Initiating a connection

Different publications propose different wake-up schemes, see e.g. [5]–[7], [15], most of which need to receive all three APs from the peripheral in the three different ACs to improve interference-resilience (i.e. avoid false wake-ups). Based on Fig. 4a, the wake-up latency L_{WU} will be at least 4ms, so we assume $L_{\text{WU}} = 5\text{ms}$ (which is long enough to detect a reasonable packet length) to allow for 1ms of processing and waking up the MRX. Preferably, the procedure for initiating a connection by using a WuRX is fully BLE compliant and transparent to the peripheral (i.e. the peripheral does not need to know whether the central is equipped with a WuRX or not), which makes introducing a WuRX-enabled device in the existing eco-system significantly easier and more acceptable.

We propose the following procedure. The central MRX remains asleep until woken up by its WuRX after detection of a AP. As $L_{\text{WU}} \leq 5\text{ms}$, MRX continuously scans the channel after waking up to initiate a connection right at the next AI, resulting in $L_{D,\max} = 2T_{\text{AI}} + \delta_{\max}$. Since the MRX is only actively burning power for T_{AI}, while the 200μW WuRX takes care of the remaining time, significant power savings are possible (quantified later). This procedure, perhaps counter-intuitively, can also reduce the power consumption of the peripheral, as we will explain next.

Table IV quantifies the effect of a WuRX on P_{avg} and $L_{D,\max}$ to initiate a connection for both central and peripheral for the different DiPs using our proposed procedure. Adding a WuRX provides different options (last row of Table IV) for reducing P_{avg} and/or $L_{D,\max}$, one could 1) keep the P_{avg} of the peripheral the same as before (without a WuRX) by choosing the same T_{AI}, resulting in the maximum possible reduction in $L_{D,\max}$, 2) keep the $L_{D,\max}$ the same as before by choosing $T_{\text{AI}} = (L_{D,\max} - \delta_{\max})/2$, which provides the maximum possible reduction in the P_{avg} of the peripheral, and 3) reducing both P_{avg} of the peripheral and $L_{D,\max}$ simultaneously by choosing T_{AI} between the above-mentioned ranges. Note that P_{avg} of the central is reduced in all of the options. As stated before, these DiPs are defined for practical scenarios with interference and other realistic non-idealities ($T_{\text{SW}} \geq 5T_{\text{AI}}$). Although we do not take these non-idealities into account in our analysis, this still allows for a fair comparison between the cases with and without a WuRX. P_{avg} and $L_{D,\max}$ without a WuRX are derived as explained in section III-C, while they are calculated for the case with a WuRX as $L_{D,\max} = 2T_{\text{AI}} + \delta_{\max}$ and P_{avg} using our assumptions of 200μW for the WuRX when active, and 3μW for the sleep state of the MRX. Thus, latency now depends only on T_{AI}, as $\delta_{\max}$ is fixed at 10ms.

Without a WuRX, centrals are generally duty-cycled ($T_{\text{SW}} < T_{\text{SI}}$) to reduce power, requiring peripherals to use shorter AIs to initiate a connection within a certain time limit. For example, in the balanced DiP (BAL), for $L_{D,\max} = 3.2\text{s}$, $T_{\text{AI}} = 200\text{ms}$, $T_{\text{SW}} = 1024\text{ms}$, and $T_{\text{SI}} = 4096\text{ms}$. If the central has an always-on WuRX, different options can be chosen as explained before depending on the scenario. E.g. in BAL$_3$ option, $T_{\text{AI}} = 1\text{s}$ reduces $L_{D,\max}$ by 1.6x from 3.2s to 2s, while simultaneously reducing P_{avg} of the peripheral by 4.4x from 89μW to 20μW, and of the central by 7.7x from 1.57mW to 203μW. For the low-power DiP (LP), improvements similar to the BAL-mode can be obtained

TABLE IV
MAXIMUM LATENCY AND POWER CONSUMPTION OF BLE DEVICES WHEN INITIATING A CONNECTION WITH OR WITHOUT WURX ($L_{\text{AP}} = 27$ BYTES).

without WuRX							with WuRX					reduction			
DiP	L_D max	T_{SW} (ms)	T_{SI} (ms)	T_{AI} (ms)	P_{avg} central (μW)	P_{avg} periph. (μW)	duty-cycled WuRX	T_{AI} (ms)	P_{avg} central (μW)	P_{avg} periph. (μW)	L_D max	P_{avg} central	P_{avg} periph.	L_D max	option
ULL	20ms	4096	4096	20	6280	865	no	20	203	865	50ms	96.7%	0%	no	-
LL	90ms	100	165	20	3820	865	no	20	203	865	50ms	94.6%	0%	44%	LL_1
							no	40	203	434	90ms	94.6%	49.8%	0%	LL_2
							no	30	203	578	70ms	94.6%	33%	22%	LL_3
BAL	3.2s	1024	4096	200	1573	89	no	200	203	89	410ms	87%	0%	87%	BAL_1
							no	1600	203	13.5	3.2s	87%	85%	0%	BAL_2
							no	1000	203	20	2s	87%	77.5%	37.5%	BAL_3
LP	4.7s	512	5120	100	631	175	no	100	203	175	210ms	67.8%	0%	95.5%	LP_1
							no	2350	203	10	4.7s	67.8%	94%	0%	LP_2
							no	700	203	27.5	1.4s	67.8%	84%	70%	LP_3
							yes 30%*	300	63	60	*** 3.9s	90%	66%	17%	LP_4
ULP	10min	11.25	1280	1000	60	20	yes 10%**	2000	23	11.5	*** 1.5min	61.6%	42.5%	85%	-

* WuRX is on 1.5s every 5s ** WuRX is on 10s every 100s *** Based on Equation (2)

(LP-mode in Table IV). However, as $L_{D,\text{max}} = 4.7$s is quite relaxed, it is also a possibility to duty-cycle the WuRX itself, to be woken up by the same sleep clock as used by the MRX. For a duty-cycled WuRX, we again include the constraint that $T_{\text{SW}} \approx 5T_{\text{AI}}$ to 'guarantee' discovery in the first SW (LP_4 option in Table IV). As a result, a duty-cycled WuRX can further reduce P_{avg} of the central, but requires the peripheral to advertise with a somewhat shorter AI in comparison to LP_3, increasing its P_{avg} compared to the always-on WuRX. The best option then depends on the (relative) power constraints of the individual devices.

For the ultra-low power DiP (ULP), with $L_{D,\text{max}} = 10$min, an always-on WuRX increases P_{avg}, so duty-cycling the WuRX here is the only reasonable option. The bottom row of Table IV shows that when the WuRX uses $T_{\text{SW}} = 10$s and $T_{\text{SI}} = 100$s, while the peripheral uses $T_{\text{AI}} = 2$s, $L_{D,\text{max}}$ reduces by 6.7x from 10min to 1.5min, P_{avg} of the peripheral reduces by 1.7x from 20μW to 11.5μW, and P_{avg} of the central by 2.6x from 60μW to 23μW. Note that with such low average power levels, leakage currents are starting to limit the achievable improvement in power consumption.

The information in Table IV is visualized in Fig. 7. A WuRX clearly enables us to move towards the desired bottom-left corner. Without a WuRX, centrals always burn more power than the peripherals. However, by using a WuRX, for latencies smaller than around 200ms, peripherals burn more power. If one can reverse the role of central and peripherals in these scenarios (peripherals scan instead of centrals), even more power can be saved in the peripherals in the low-latency scenarios. Changing roles is not a current feature in BLE and is suggested in this paper for the first time.

Another interesting observation is that the 30% duty-cycled WuRX in low-power DiP option has an almost perfect balance in power consumption between central and peripheral, and may therefore be very suited for equal or equally-constrained devices. As here $P_{\text{avg}} \approx 60\mu$W is also quite low and latency is less than 4s, it looks promising for many IoT scenarios, such as in smart homes where numerous sensors need to transfer data among each other. Note that in such scenarios, there is no device with a "bigger battery" as in more traditional BLE smartphone-peripheral scenarios, where the central is allowed to burn more power than the peripheral.

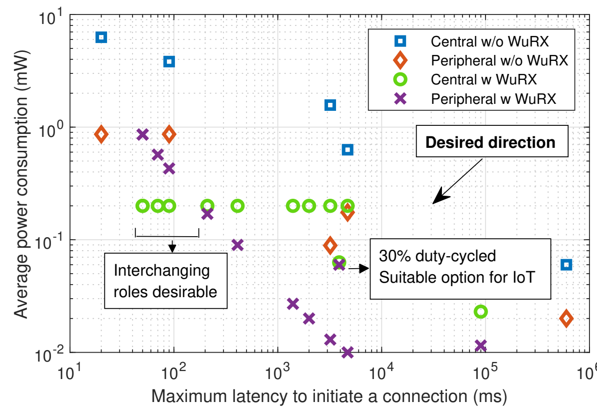


Fig. 7. Average power consumption vs. maximum latency to initiate a connection for both central and peripheral with and without a WuRX.

V. USER SCENARIOS

Having discussed the potential benefits of a WuRX in the previous sections, here we take a quantitative look at some popular BLE applications in IoT. All assumptions (see section III) apply here as well.

A. Smart door lock

A smart door lock is one of the most popular IoT devices, and found in many homes [44]. The door can be locked/unlocked automatically as the smartphone approaches it. Especially when unlocking, the user should experience no delay. The door lock battery needs to be replaced when it is depleted, so it is more power-constrained than the smartphone. A maximum delay of 100ms looks like an acceptable number, which makes the LL DiP of Table IV applicable. Therefore, it is logical to put the burden of the power-hungry scanning procedure on the smartphone (central, $P_{\text{avg}} = 3.82\text{mW}$) while the door lock (peripheral, $P_{\text{avg}} = 865\mu\text{W}$) is advertising.

However, by having a WuRX and using LL_2 option in Table IV, power consumption can be reduced by 94.6% in the smartphone, from 3.82mW to 203μW, and by 49.8% in the door lock, from 865μW to 434μW, while latency stays 90ms.

A WuRX gives us another option to further reduce the power consumption of the battery-constrained door lock by interchanging roles. The door lock can be the central and continuously scans the channel using a WuRX, reducing P_{avg} from 865μW to 203μW, a reduction of 76.5%. The smartphone can be the peripheral, sending APs at $T_{\text{AI}} = 40\text{ms}$ and bringing P_{avg} down from 3.82mW to 434μW, a reduction of 88.6% for the same 90ms latency. It is clear that by interchanging roles, a WuRX is able to further reduce the power consumption of the door lock than without changing roles (76.5% instead of 49.8%). Note that the smartphone only needs to advertise when it is in proximity of the door, which is generally known due to its GPS or WiFi-based location estimation.

Some door locks already use a proximity sensor to advertise only when somebody is approaching, significantly improving the battery life of the door lock. In such a case, a WuRX in the smartphone (central) can reduce its power consumption from 3.82mW to 203μW, a 94.6% reduction during the time it is actively scanning.

B. Smoke detectors and fire alarm

Another smart home application is a collection of smoke detectors connected to a central fire alarm. Here, the smoke detectors act as peripherals and advertise whenever they detect smoke. The fire alarm performs the central role and scans the channel to receive the APs.

Considering an acceptable maximum latency of around 3s for this scenario, the BAL DiP is applicable, resulting in $P_{\text{avg}} = 1.57\text{mW}$ for the central and $P_{\text{avg}} = 89\mu\text{W}$ for the peripheral (Table IV). If the fire alarm is equipped with a WuRX, its power consumption is reduced to 203μW, a 87% reduction. Additionally, the WuRX in the fire alarm would allow reduction of the advertising frequency of the smoke detectors. For example, choosing $T_{\text{AI}} = 1\text{s}$ for a 37.5% reduced maximum latency of only 2s, the peripheral power consumption is reduced by 77.5% from 89μW to 20μW. Thus, in this scenario, a WuRX can simultaneously improve latency and significantly extend battery life of both peripheral and central devices.

VI. Conclusions

There is a well-known trade-off between the latency and P_{avg} in BLE. We quantified these using one of the lowest-power commercially available devices, We also quantified, for the first time, the potential benefits of a 200μW WuRX for BLE.

To initiate a connection, a peripheral transmits APs and a central scans for these. We assume the WuRX is capable of detecting (rather than decoding) APs with the same sensitivity as the MRX can decode it. Upon detection of the AP, the WuRX wakes up the MRX, which then scans for the next AP. Without changing any configuration settings, this increases latency: for a maximum latency of $< 40\text{ms}$, such a WuRX is not useful. For higher tolerable latencies, the WuRX allows for significant improvements of latency and power consumption for both central and peripheral by using different scanning and advertising settings. For example, for the predefined low-power DiP, a WuRX can simultaneously reduce P_{avg} of the central by 67.8%, P_{avg} of the peripheral by 84%, and the maximum latency by 70%. Also, interchanging the role of centrals and peripherals was suggested for the first time in this paper to further reduce the power of the peripherals in initiating a connection in scenarios that need less than 200ms of latency. Finally, to further reduce P_{avg} in scenarios that allow for even higher latency, the WuRX itself may be duty-cycled. For example, for a maximum latency of 4 seconds, P_{avg} of both central and peripheral device can be reduced to around 60μW, which is a perfect fit for many IoT-scenarios where both ends of the connection may be equally battery-constrained.

Overall, a WuRX in BLE will significantly improve the energy efficiency of the devices on both ends of an unestablished link, except for the most latency-critical use cases.

Acknowledgment

We thank J. Prummel, M. Papamichail, and R. Kohlmann from Dialog for the useful discussions and detailed information on the DA14531 BLE chip.

References

[1] P. Bassirian, D. Duvvuri, D. S. Truesdell, N. Liu, B. H. Calhoun, and S. M. Bowers, "30.1 A temperature-robust 27.6nW 65dBm wakeup receiver at 9.6GHz X-band," in *2020 IEEE International Solid- State Circuits Conference - (ISSCC)*, 2020, pp. 460–462.

[2] N. C. Luong, D. T. Hoang, P. Wang, D. Niyato, D. I. Kim, and Z. Han, "Data collection and wireless communication in internet of things (IoT) using economic analysis and pricing models: A survey," *IEEE Communications Surveys Tutorials*, vol. 18, no. 4, pp. 2546–2590, 2016.

[3] D. Griffith, "Wake-up radio for low-power internet of things applications: An alternative method to coordinate data transfers," *IEEE Solid-State Circuits Magazine*, vol. 11, no. 4, pp. 16–22, 2019.

[4] J. Tosi, F. Taffoni, M. Santacatterina, R. Sannino, and D. Formica, "Performance evaluation of Bluetooth Low Energy: A systematic review," *Sensors*, vol. 17, no. 12, p. 2898, 2017.

[5] M. R. Abdelhamid, A. Paidimarri, and A. P. Chandrakasan, "A -80dBm BLE-compliant, FSK wake-up receiver with system and within-bit dutycycling for scalable power and latency," in *2018 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, 2018, pp. 1–4.

[6] M. Ding, P. Zhang, C. Lu, Y. Zhang, S. Traferro, G.-J. van Schaik, Y.-H. Liu, J. Huijts, C. Bachmann, G. Dolmans *et al.*, "A 2.4 GHz BLE-compliant fully-integrated wakeup receiver for latency-critical IoT applications using a 2-dimensional wakeup pattern in 90nm CMOS," in *2017 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*. IEEE, 2017, pp. 168–171.

[7] A. Alghaihab, J. Breiholz, H.-S. Kim, B. Calhoun, and D. D. Wentzloff, “A 150 μW -57.5 dBm-sensitivity Bluetooth Low-Energy back-channel receiver with LO frequency hopping,” in *2018 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*. IEEE, 2018, pp. 324–327.

[8] P.-H. P. Wang and P. P. Mercier, “28.2 a 220μW -85dBm sensitivity BLE-compliant wake-up receiver achieving -60dB SIR via single-die multi-channel FBAR-based filtering and a 4-dimentional wake-up signature,” in *2019 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2019, pp. 440–442.

[9] C. Salazar, A. Cathelin, A. Kaiser, and J. Rabaey, “A 2.4 GHz interferer-resilient wake-up receiver using a dual-IF multi-stage N-path architecture,” *IEEE Journal of Solid-State Circuits*, vol. 51, no. 9, pp. 2091–2105, 2016.

[10] A. A. Benbuk, N. Kouzayha, J. Costantine, and Z. Dawy, “Tunable, asynchronous, and nanopower baseband receiver for charging and wake-up of IoT devices,” *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[11] M. A. Karami and K. Moez, “An integrated RF-powered wake-up wireless transceiver with -26 dBm sensitivity,” *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[12] X. Liao, S. Xie, J. Xu, and L. Liu, “A 0.4 V, 6.4 nW, -75 dBm sensitivity fully differential wake-up receiver for WSNs applications,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, pp. 1–11, 2022.

[13] A. Dissanayake, H. L. Bishop, S. M. Bowers, and B. H. Calhoun, “A 2.4 GHz -91.5 dBm sensitivity within-packet duty-cycled wake-up receiver,” *IEEE Journal of Solid-State Circuits*, vol. 57, no. 3, pp. 917–931, 2022.

[14] H. Bialek, A. Binaie, S. Ahasan, K. R. Sadagopan, M. L. Johnston, H. Krishnaswamy, and A. Natarajan, “A passive wideband noise-canceling mixer-first architecture with shared antenna interface for interferer-tolerant wake-up receivers and low-noise primary receivers,” *IEEE Journal of Solid-State Circuits*, pp. 1–15, 2022.

[15] P. P. Wang and P. P. Mercier, “An interference-resilient BLE-compatible wake-up receiver employing single-die multi-channel FBAR-based filtering and a 4-D wake-up signature,” *IEEE Journal of Solid-State Circuits (early access)*, pp. 1–1, 2020.

[16] J. Im, H.-S. Kim, and D. D. Wentzloff, “A 220-μW -83-dBm 5.8-GHz third-harmonic passive mixer-first LP-WUR for IEEE 802.11 ba,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 67, no. 7, pp. 2537–2545, 2019.

[17] E. Alpman, A. Khairi, R. Dorrance, M. Park, V. S. Somayazulu, J. R. Foerster, A. Ravi, J. Paramesh, and S. Pellerano, “802.11 g/n compliant fully integrated wake-up receiver with -72 dBm sensitivity in 14-nm FinFET CMOS,” *IEEE Journal of Solid-State Circuits*, vol. 53, no. 5, pp. 1411–1422, 2018.

[18] J. Im, H.-S. Kim, and D. D. Wentzloff, “A 470μW -92.5 dBm OOK/FSK receiver for IEEE 802.11 WiFi LP-WUR,” in *ESSCIRC 2018-IEEE 44th European Solid State Circuits Conference (ESSCIRC)*. IEEE, 2018, pp. 302–305.

[19] R. Liu, A. B. KT, R. Dorrance, D. Dasalukunte, M. A. S. Lopez, V. Kristem, S. Azizi, M. Park, and B. R. Carlton, “An 802.11 ba 495μW -92.6 dBm-sensitivity blocker-tolerant wake-up radio receiver fully integrated with Wi-Fi transceiver,” in *2019 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*. IEEE, 2019, pp. 255–258.

[20] K. Lee, J. Koo, W. Sun, and S. Choi, “Power consumption modeling of BLE advertising devices,” in *Proceedings of the The 7th International Conference on Information and Communication Technology Convergence (ICTC 2016), Jeju Island, Korea*, 2016, pp. 19–21.

[21] A. Dementyev, S. Hodges, S. Taylor, and J. Smith, “Power consumption analysis of Bluetooth Low Energy, Zigbee and ANT sensor nodes in a cyclic sleep scenario,” in *2013 IEEE International Wireless Symposium (IWS)*. IEEE, 2013, pp. 1–4.

[22] R. Tei, H. Yamazawa, and T. Shimizu, “BLE power consumption estimation and its applications to smart manufacturing,” in *2015 54th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*. IEEE, 2015, pp. 148–153.

[23] Y.-Y. Ting and F. J. Lin, “A comparison and evaluation of different BLE connection methods for wearable devices,” in *2017 IEEE Conference on Standards for Communications and Networking (CSCN)*. IEEE, 2017, pp. 240–245.

[24] K. Cho, W. Park, M. Hong, G. Park, W. Cho, J. Seo, and K. Han, “Analysis of latency performance of Bluetooth Low Energy (BLE) networks,” *Sensors*, vol. 15, no. 1, pp. 59–78, 2015.

[25] E. Garcia-Espinosa, O. Longoria-Gandara, I. Pegueros-Lepe, and A. Veloz-Guerrero, “Power consumption analysis of Bluetooth Low Energy commercial products and their implications for IoT applications,” *Electronics*, vol. 7, no. 12, p. 386, 2018.

[26] E. Mackensen, M. Lai, and T. M. Wendt, “Bluetooth low energy (BLE) based wireless sensors,” in *SENSORS, 2012 IEEE*. IEEE, 2012, pp. 1–4.

[27] B.-R. Chen, S.-M. Cheng, and J.-J. Lin, “Energy-efficient BLE device discovery for internet of things,” in *2017 Fifth International Symposium on Computing and Networking (CANDAR)*. IEEE, 2017, pp. 75–79.

[28] F. J. Dian, “An analytical scheme for power consumption of battery-operated peripheral BLE nodes,” in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2019, pp. 1021–1026.

[29] A. Yousefi, K. Somaratne, and F. J. Dian, “Analysis of time synchronization based on current measurement for Bluetooth Low Energy (BLE),” in *2017 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE, 2017, pp. 602–607.

[30] A. A. Kherani, A. S. Chiddarwar, and A. K. Singh, “Power and QoS optimization for BLE-based accessory-smartphone communications,” in *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*. IEEE, 2015, pp. 394–399.

[31] BLE Core Specifications. [Online]. Available: https://www.bluetooth.com/specifications/bluetooth-core-specification/

[32] Bluetooth Generic Attributes. [Online]. Available: https://www.bluetooth.com/specifications/gatt/

[33] F. Kuo, S. Binsfeld Ferreira, H. R. Chen, L. Cho, C. Jou, F. Hsueh, I. Madadi, M. Tohidian, M. Shahmohammadi, M. Babaie, and R. B. Staszewski, “A Bluetooth Low Energy transceiver with 3.7-mW all-digital transmitter, 2.75-mW High-IF discrete-time receiver, and Tx/Rx switchable on-chip matching network,” *IEEE Journal of Solid-State Circuits*, vol. 52, no. 4, pp. 1144–1162, 2017.

[34] H. Liu, Z. Sun, D. Tang, H. Huang, T. Kaneko, Z. Chen, W. Deng, R. Wu, and K. Okada, “ADPLL-centric Bluetooth Low Energy transceiver with a 2.3-mW interference-tolerant hybrid-loop receiver in 65-nm CMOS,” *IEEE Journal of Solid-State Circuits*, vol. 53, no. 12, pp. 3672–3687, 2018.

[35] E. Bechthum, J. Dijkhuis, M. Ding, Y. He, J. Van Den Heuvel, P. Mateman, G. Van Schaik, K. Shibata, M. Song, E. Tiurin, S. Traferro, Y. Liu, and C. Bachmann, “30.6 a low-power BLE transceiver with support for phase-based ranging, featuring 5µs PLL locking time and 5.3ms ranging time, enabled by staircase-chirp PLL with sticky-lock channel-switching,” in *2020 IEEE International Solid- State Circuits Conference - (ISSCC)*, 2020, pp. 470–472.

[36] M. Tamura, H. Takano, H. Nakahara, H. Fujita, N. Arisaka, S. Shinke, N. Suzuki, Y. Nakada, Y. Shinohe, S. Etou, T. Fujiwara, F. Kondo, K. Yamamoto, T. Matsumoto, and Y. Katayama, “A 0.5-V BLE transceiver with a 1.9-mW Rx achieving 96.4 dBm sensitivity and -27-dBm tolerance for intermodulation from interferers at 6- and 12-MHz offsets,” *IEEE Journal of Solid-State Circuits*, vol. 55, no. 12, pp. 3376–3386, 2020.

[37] Dialog DA14531 Datasheet. [Online]. Available: https://www.dialog-semiconductor.com/products/connectivity/bluetooth-low-energy/products/da14530-and-da14531

[38] Nordic nRF5340 Datasheet. [Online]. Available: https://www.nordicsemi.com/Products/Low-power-short-range-wireless/nRF5340

[39] SiLabs ERF32BG22 Datasheet. [Online]. Available: https://www.silabs.com/wireless/bluetooth/efr32bg22-series-2-socs#

[40] Texas Instruments CC2652RSIP Datasheet. [Online]. Available: https://www.ti.com/document-viewer/CC2652RSIP/datasheet

[41] TI Bluetooth Power Calculator Tool. [Online]. Available: https://www.ti.com/tool/BT-POWER-CALC#description

[42] P. H. Kindt, M. Saur, M. Balszun, and S. Chakraborty, “Neighbor discovery latency in BLE-like protocols,” *IEEE Transactions on Mobile Computing*, vol. 17, no. 3, pp. 617–631, 2018.

[43] Bluetooth Application Programming Interface (API). [Online]. Available: https://android.googlesource.com/platform/packages/apps/Bluetooth/+/master/src/com/android/bluetooth/gatt/ScanManager.java

[44] Top 10 Most Popular IoT Devices In 2020. [Online]. Available: https://robots.net/tech-reviews/top-iot-devices/