Experiences and Measurements with Bluetooth Low Energy (BLE) Enabled and Smartphone Controlled Embedded Applications

David Tauchmann and Axel Sikora
Laboratory Embedded Systems and Communication Electronics, University of Applied Sciences Offenburg, D77652 Offenburg, Germany
Email: dtauchma@stud.hs-offenburg.de, axel.sikora@hs-offenburg.de

***Abstract*—Bluetooth Low Energy extends the Bluetooth standard in version 4.0 for ultra-low energy applications through the extensive usage of low-power sleeping periods, which inherently difficult in frequency hopping technologies. This paper gives an introduction into the specifics of the Bluetooth Low Energy protocol, shows a sample implementation, where an embedded device is controlled by an Android smart phone, and shows the results of timing and current consumption measurements.**

***Index Terms*—bluetooth low energy, bluetooth smart, energy consumption, android, smart phone**

I. Introduction

Since its early beginnings in 1999, the Bluetooth standard is known for its good stability, its real-time characteristics, and its low power consumption in active mode. Due to these characteristics, around 9 billion Bluetooth enabled devices were sold until e/o 2012; a total of more than 20 billion devices is expected until 2017 [1]. However, due to the characteristics of the Frequency Hopping Spread Spectrum (FHSS) scheme, ultra low energy regimes with extended low power (sleeping) periods haven’t been possible, as resynchronization times to re-enter an existing link significantly contributed to the energy budget. Thus, a good number of low-energy applications, be it battery-powered, be it energy autarkic, were lost to alternative technologies. For applications in home and building automation, major competitors are the EnOcean Radio Protocol (ERP) [2] and possibly ZigBee [3] or Z-Wave [4]. For consumer electronics and smart accessories for sports fitness, health, and wellness the ANT / ANT+-protocol [5] has won a decent market share in the past. And of course, many other proprietary technologies are around.

In order to also address these markets and to use its excellent installation basis in smart phones, the Bluetooth Special Interest Group (BSIG) included the Bluetooth Low Energy (BLE) into its v4.0 [6]. Its energy savings are mostly attributed to changes in the FHSS schemes and to a modified modulation scheme.

Manuscript received April 1, 2014; revised July 16, 2014.

This contribution presents the architecture and the characteristics of the BLE technology, shows the implementation of a sample application from the authors, and compares the results of their own measurements with the parameters given in the datasheets. For this, the contribution is structured as follows:

Ch. II gives a short overview on the major features of Bluetooth Low Energy. Ch. III describes our sample implementation and our test and measurement setup. Using this setup, the results observed Ch. IV were observed. These results are also compared to other competing communication solutions. Finally, ch. V gives a short conclusion.

II. Architecture

A. Device Types

Bluetooth devices supporting the legacy Bluetooth and BLE are called Dual Mode Devices and are marketed under the registered trademark “Bluetooth Smart Ready”. Devices supporting BLE only are called Single Mode Devices and marketed under the registered trademark “Bluetooth Smart” (cf. Fig. 1). The older Bluetooth versions, i.e. 2.1 and 3.0, are known as „Bluetooth Classic” [7].

Figure 1. Registered trademarks for bluetooth smart (left) and bluetooth smart ready (right)

It should be well noted that changes relate to all layers of the protocol, from the physical layer through to the communication models on the application layer. The changes on the physical layer imply the backward incompatibility of Bluetooth classic devices with Bluetooth Smart devices.

B. Physical Layer Issues

Both Bluetooth versions are operated in the 2.4 GHz-ISM-band, are using a Frequency Hopping Spread Spectrum (FHSS) scheme and a Gaussian Frequency Shift Keying (GFSK) modulation. The specifications of the physical layers mainly differ in three aspects:

doi: 10.12720/ijeee.3.4.292-296

- BLE uses a modulation index of 0.5 (close to a Gaussian Minimum Shift Keying (GMSK) scheme), compared to 0.35 for Classic Bluetooth technology. This change lowers the power consumption in active transmitting and receiving states and also improves the range of BLE versus Classic Bluetooth.
- BLE uses only 40 channels, 2 MHz wide, while Classic Bluetooth uses 79 channels, 1 MHz wide. The 40 BLE channels are shown in Fig. 2.
- Three channels, which are located between the most widely used Wireless LAN channels, are used for device discovery and connection setup. These channels, also known as “advertising channels”, are used to search for other devices or promote its own presence to devices that might be looking to make a connection.

In comparison, Classic Bluetooth technology uses 32 channels for the same task. This reduction is essential to reduce the time for scanning. According to the Bluetooth propaganda, like [8] BLE has to switch “on” for just 0.6 to 1.2ms to scan for other devices using its three advertising channels. Classic Bluetooth, instead, requires 22.5ms to scan its 32 channels. The power savings may be significant: BLE holds the promise to consume 10 to 20 times less energy than Classic Bluetooth technology to locate other nodes.

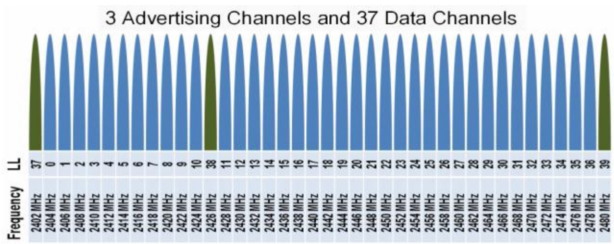

Figure 2. Channel distribution for 2.4 GHz BLE

C. Device Roles and Topologies

Whereas Bluetooth Classic supports master-slave-topologies, only, BLE supports additional roles, i.e. Broadcaster, Observer, Central, and Peripheral, which allows a much simpler and more energy efficient ad-hoc communication. Together with the increased number of active nodes within a network, theses aspects significantly to the flexibility and energy efficiency.

III. Sample Application

A. Task Description

In the project, an embedded light switch shall be controlled from an Android based app on a smart phone in such a way that bidirectional (read/write) operations are supported and that measurements of energy consumption and timing behavior can be performed.

B. Embedded Device Selection

BLE modules are available from various manufacturers, like CSR [9], Texas Instruments [10], Panasonic [11] or Bluegiga [12]. Due to direct support from TI, no comparison was performed between the different vendors, but the TI platforms CC2540EM and CC2541EM were used together with the SmartRF05 Evaluation Board (EB). A comparison of the key characteristics of the two transceiver SOCs is shown in Table I.

TABLE I. Timing and Electrical Specifications for CC2540 and CC2541 Devices from TI [13], [14]

	CC2540	CC2541
device type	System-on-Chip	
time: power mode 1 → active mode	4 µs	
time: power mode 2 or 3 → active mode	120 µs	
current consumption power mode 1 (short wake-up)	235 µA	270 µA
current consumption power mode 2 (sleep timer on)	0,9 µA	1 µA
current consumption power mode 3 (external interrupts)	0,4 µA	0,5 µA
current consumption standard receive mode	15,8 mA	14,7 mA
current consumption standard transmit mode (0 dBm)	21,0 mA	14,3 mA

C. Smart Phone Selection

Bluetooth 4.0 is supported with Android versions 4.3 and higher. At the time of the device selection in October 2013, only two devices were available with Android 4.3: a Nexus 7 and a Samsung Galaxy S4. Due to availability reasons, the Samsung smart phone was selected.

D. Embedded Device Implementation

The embedded application is based on TI’s “BLE Software Stack” in version 1.4.0, being available at [15]. The application programming was executed in ANSI-C using “IAR Embedded Workbench for 8051” [16] within TI-RTOS, a free Real-Time Operating System for TI Devices [17].

Figure 3. Screenshot of the android app for light control, including a lightweight application protocol logger

E. App Implementation

The programming of the smart phone app is done in Java using the Android SDK [18] and is based on the demo project "BluetoothLeGatt," where Gatt stands for "Generic Attribute Profile" and allows the reading and writing of the parameters, the so-called "characteristics". Fig. 3 shows a screenshot of the light app, which also includes a lightweight application protocol logger.

F. Measurement Setup

The measurement setup is shown in Fig. 4. A Tektronix TDS2002 is used to measure the voltage drop over a shunt resistor into the CC254xEM. The size of the shunt resistor varies with the measurement range.

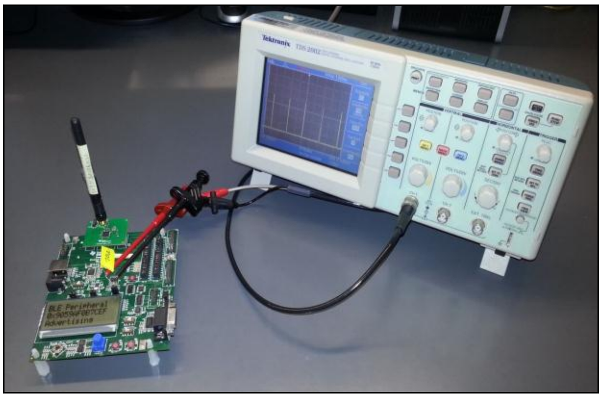

Figure 4. Setup for timing and current measurements using an oscilloscope and the SmartRF05 board with the CC2541EM

Further reference measurements for the validation of the results described below have been performed.

G. Monitoring Network Traffic

The network traffic is monitored with the help of TI's BLE Packet Sniffer eavesdropping all packets with a USB dongle. Fig. 5 shows a sample output with additional manual decoding.

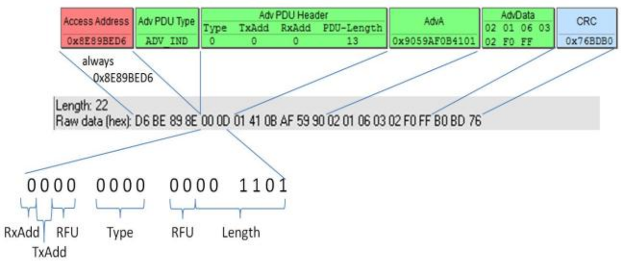

Figure 5. Decoding of an advertising packet being monitored by the BLE packet sniffer

IV. Measurement Results and Comparisons

A. Timing Measurements

In the initial state, the peripheral device is in the advertising "state looking for a central node and sending periodic advertising events (cf. Fig. 6). The periodicity *T_advEvent* is measured as 108ms and consists of the pre-defined interval *advInterval* of 100ms and the variable delay *advDelay* of 8ms.

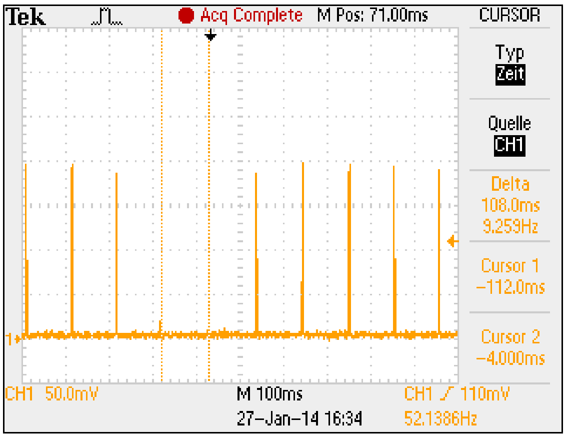

Figure 6. *I*(*t*)-Diagram of periodic transmission of advertising events

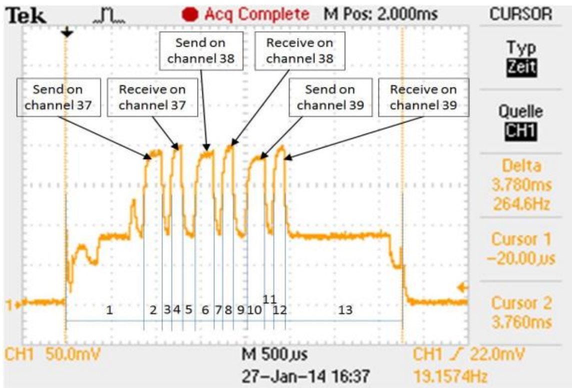

Figure 7. Detailed *I*(*t*)-diagram of advertising events

TABLE II. Timing of Unsuccessful Advertising Events on All Three Advertising Channels

	delta time [µs]	total time [µs]
1	860	860
2	220	1080
3	100	1180
4	120	1300
5	150	1450
6	200	1650
7	100	1750
8	120	1870
9	150	2020
10	200	2220
11	100	2320
12	2120	2440
13	1310	3750

Zooming into Fig. 6 allows a more detailed analysis (cf. Fig. 7). During the first 860 µs the peripheral wakes up and performs various local operations. Then, the Advertising Events ("ADV_IND" PDU) are sent out subsequently on the three Advertising Channels #37, #38 and #39. After each transmission, the transceiver switches into receive mode and waits for the answer of a central node. If there is no answer on channel #37, then the advertisement continues on the next channel #38. Consequently, Fig. 7 shows the case, when no central node responds. In this case, the microcontroller wakes up for a total of 3.75ms to look for a central node. A detailed listing of the different phases can be found in Table II. The peak current consumption in transmit mode is around 19mA, in receive mode it may reach 20mA. The current consumption without transceiver activity is in the range of 8mA.

After the setup of the connection, the data exchange can be started. The timing behavior of such a two-way communication is shown in Fig. 8.

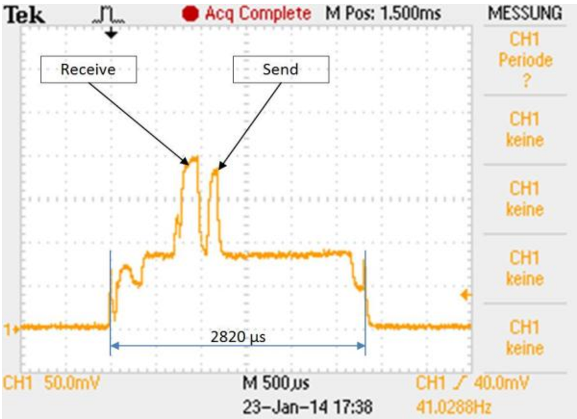

Figure 8. $I(t)$-Diagram of two-way communication

B. Energy Measurements

Based on the measurements of Fig. 8 the required electrical discharge can be calculated. Due to the simple (rectangular) characteristics of the diagram, an easy piece-wise integration is performed, as shown in Fig. 9.

The discharge for one transaction can be added up:

$$Q_{trans} = 11\text{ mA} * 250\ \mu\text{s} + 10\text{ mA} * 100\ \mu\text{s} + 8\text{ mA} * 2.9\text{ ms} = 26.95\ \mu\text{As} \quad (1)$$

C. Lifetime Calculations

In the power-down mode, we measured an average current of the device:

$$I_{Sleep} = 0.33\ \mu\text{A} \quad (2)$$

$$Q_{bat} = 230\text{ mAh} = 828\text{ As} \quad (3)$$

Assuming an energy reservoir of a typical coin cell (CR2032) with a capacity of the lifetime of such a BLE communication system can be calculated to the values of Table III. These are of course best case estimations, where real-life parameters would decrease this idealistic performance. The most important amongst them are:

- Power consumption of the application itself,
- Retransmissions and reconnection after network failures,
- Self-discharge of the battery, and
- Temperature dependencies of the electric parameters.

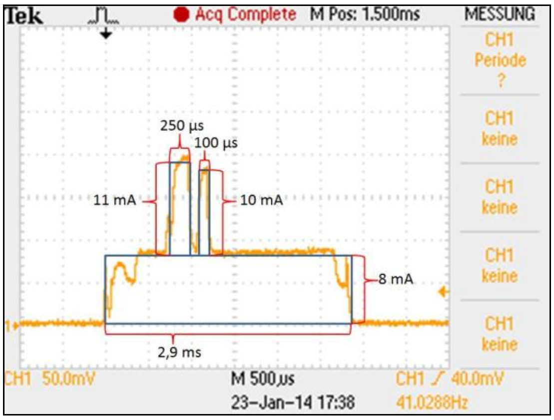

Figure 9. $I(t)$-Diagram of two-way communication for calculation of discharge

TABLE III. BEST CASE ESTIMATION FOR LIFETIME OF A BLE SYSTEM WITH PERIODIC DATA TRANSFER

transaction each x second	Q_{tot} in interval x	trans-actions	life time	
			seconds	years
x	$Q_{tot} = Q_{trans} + I_{Sleep} * x$	$n_{trans} = Q_{bat} / Q_{tot}$	$t = n_{transactions} * x$	
30	36.85 µAs	22.5E6	674E6	30
10	30.25 µAs	27.4E6	274E6	10
1	27.28 µAs	30.4E6	30.4E6	1

D. Comparisons

In comparison with other short-range wireless network (SRWN) protocols, it becomes clear that BLE is now in the same order of magnitude. Basically all devices for competing SRWN protocols, like the ones mentioned in ch. I, draw between 10 and 20 mA in transmit or receive modes. The sleeping current of the devices is not so much influenced from the communication protocol, but from the device implementation.

The energy consumption mainly depends on the duty cycle, i.e. the data rate, the transmission efficiency (ratio of net data bits to gross channel bits including preamble, frame check sequences, and other coding overhead) and the network management overhead. The latter very much depends on the actual installation parameters and is therefore not well comparable.

Although not being optimized for high data-rate, BLE is still the fastest with regard to data rate. BLE is operated at 1Mbps, where ZigBee operates at 250kbps, EnOcean at 125kbps, DECT-ULE at 112.5kbps, and Z-Wave at mere 40kbps.

V. CONCLUSIONS

With the new BLE approach, Bluetooth has now patched things up with regard to energy efficient connections with low-energy, low-complexity peripherals. It is now at par with other, still being strongest, when data is transmitted periodically. But the broadcast modes help also in other use cases.

Even though not all measured values were as good as indicated by earlier publications, the results are still good enough to be for real life projects, where Bluetooth can enjoy an unrivaled proliferation in smart phones, tablets, and other smart handhelds.

ACKNOWLEDGMENT

The authors wish to thank Geir Lauritsen from Texas Instruments for his multiannual and steady support in many of their activities, including this project. The authors are also grateful for the financial support from the Research Council of Offenburg University of Applied Sciences.

REFERENCES

[1] *Bluetooth SIG Analyst Digest Q4 2012*, Bluetooth Special Interest Group, 2013.

[2] Information Technology-Home Electronic Systems (HES)-Part 3-10: Wireless Short-Packet (WSP) Protocol Optimized for Energy Harvesting-Architecture and Lower Layer Protocols, ISO/IEC 14543-3-10:2012

[3] (2014). ZigBee Alliance. [Online]. Available: http://www.zigbee.org

[4] Z-Wave World. [Online]. Available: http://zwaveworld.com/

[5] (2014). This is ant - The wireless sensor network solution. [Online]. Available: http://www.thisisant.com/

[6] Specification, Core Version 4.0, Bluetooth SIG, Inc, 2014.

[7] *CC2540/41 Bluetooth Low Energy Software Developer's Guide (Rev. F)*, Texas Instruments, Inc., 2012.

[8] J. Decuir, *Bluetooth 4.0 Low Energy, Standards Architect*, CSR plc, 2010.

[9] (2013). Bluetooth smart. CSR plc. [Online]. Available: http://www.csr.com/products/technology/low-energy

[10] *Wireless Connectivity*, Tools and Software for Bluetooth/Bluetooth Low Energy, Texas Instruments, Inc., 2014.

[11] Bluetooth low energy. Panasonic. [Online]. Available: http://www.panasonic.com/industrial/electronic-components/rf-modules/bluetooth-low-energy/index.aspx.

[12] *Bluetooth 4.0 Modules*, Bluegiga, Bluetooth Smart Ready, 2013.

[13] *2.4-GHz Bluetooth Low Energy System-On-Chip (Rev. F),* SWRS084F, Texas Instruments, Inc. Jun. 2013.

[14] *2.4-GHz Bluetooth Low Energy and Proprietary System-on-Chip (Rev. D),* SWRS110D, Texas Instruments, Inc., Jun. 2013.

[15] Bluetooth low energy software stack and tools. [Online]. Available: http://www.ti.com/tool/ble-stack

[16] *Integrated Development Environment And Optimizing C/C++ Compiler for 8051*, IAR Embedded Workbench for 8051, 2014.

[17] Ti-Rtos: Real-Time operating system (RTOS). [Online]. Available: http://www.ti.com/tool/ti-rtos

[18] Android Developer Portal. Get the Android SDK. [Online]. Available: http://developer.android.com/sdk/index.html

D. Tauchmann has done a three-year technical apprenticeship in mechatronics at Robert Bosch GmbH in Buehl (Germany). He has earned his B.Eng, in Mechatronics (including teachers' qualification) at Offenburg University of Applied Sciences. He is currently pursuing a M.Sc. in Mechatronics (including teachers' qualification) at Offenburg University of Applied Sciences) and working as research assistant in the Laboratory Embedded Systems and Communication Electronics.

Dr.-Ing. Axel Sikora holds a diploma of Electrical Engineering and a diploma of Business Administration, both from Aachen Technical University. He has done a Ph.D. in Electrical Engineering at the Fraunhofer Institute of Microelectronics Circuits and Systems, Duisburg, with a thesis on SOI-technologies. After various positions in the telecommunications and semiconductor industry, he joined at the Baden-Wuerttemberg Cooperative State University Loerrach in 1999 as Professor. In 2011, he joined Offenburg University of Applied Sciences, where he holds the professorship of Embedded Systems and Communication Electronics.

His major interest is in the system development of efficient, energy-aware, autonomous, secure, and value-added algorithms and protocols for wired and wireless embedded communication. He is founder and head of Steinbeis Transfer Center Embedded Design and Networking (stzedn).

Dr. Sikora is author, co-author, editor and co-editor of several textbooks and numerous papers in the field of embedded design and wireless and wired networking. Amongst many other duties, he serves as member of the Steering Board of Embedded World Conference.

