Testing a Commercial Sensor Platform for Wideband Applications based on the 802.15.4 standard

David Guillén Jiménez and Franck Le Gall*
Easy Global Market, 2000 Route des Lucioles, 06410 Biot, France
{david.guillen, franck.le-gall}@eglobalmark.com

Abstract

Wireless Sensor Networks are increasing their capacity to handle larger amounts of data. These networks, which were born with the purpose of collecting environmental information, could be also used for wideband applications, e.g. transmitting audio and / or video data streams. The European project EAR-IT [1] uses acoustic detection to make smarter cities. The idea is to make large-scale deployments and use sounds to create innovative applications. Current technology can be a limitation in terms of performance. This document shows what parameters are relevant to be measured in a 802.15.4 Wireless Sensor Network to determine its suitability to wideband transmission of multimedia streams. TST-Sistemas is the commercial sensor platform chosen for the evaluation and qualification.

Keywords: 802.15.4, Personal Area Network (PAN), Wireless Sensor Networks (WSN), Internet of Things (IoT), TST Sistemas, Audio transmission, Smart Cities, Surveillance

1 Introduction

The Internet of Things (IoT) is becoming a large sensitive skin that covers the Earth and offers real-time information to the humankind. We started gathering small amounts of information, like temperature, humidity, etc. and nowadays we are interested in any data we can retrieve and process. Audio and video are new targets in Wireless Sensor Networks (WSNs) based on the standard IEEE 802.15.4. Despite of their constraints, these networks are able to handle constant flows of encoded data. One of the tasks undertaken in the European project EAR-IT was to deploy a multi-hop WSN able to stream compressed audio. The European project EAR IT [1] focuses on audio recognition and acoustic event detection in smart cities. In addition to our own deployments, we used the large-scale WSN deployed in Santander (over 10,000 sensors), inside the project SmartSantander [2], to stream audio over this low-resource infrastructure. The different performances observed between the SmartSantander WSN (based on Waspmote) and those we deployed (based on Telosb) made us develop a technique to measure their most relevant characteristics [3, 4]. This technique is actually independent of the nature of the data transmitted, so it is not limited to audio. In our pursuit of new commercial platforms that could fit or overcome the requirements to stream audio (or other wideband applications) we searched for more powerful processing devices. There have already been studies about multimedia sensors although few of them consider timing on realistic hardware constraints for sending/receiving flows of packets [2, 5, 6, 7]. In this paper, we will present the experimentations needed to enable real-time multi-hop streaming of encoded wideband data with low-resource devices. The motivation of this paper is to state the suitability of a commercial sensor platform for wideband applications using a standard validation method that takes into account the most relevant characteristics for this kind of transmissions. This method is detailed in [3]. The commercial

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, volume: 6, number: 1, pp. 24-36
*Corresponding author: Easy Global Market, 2000 Route des Lucioles, 06410 Biot, France, Tel: 0033-620-035-420

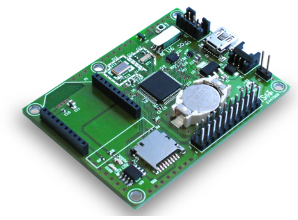


Figure 1: TST-Sistemas' mote

platform chosen is TST Sistemas [8] (figure 1). TST-Sistemas and Easy Global Market are two independent companies. The author warranties total independence and objectivity presenting the results of this evaluation process.

2 Goals and sections

The goal of this paper is to analyze the performance of TST Sistemas motes in wideband multimedia applications. Section 3: "The TST-Sistemas' Platform" describes some of the technical details related to the hardware to be tested and the way it is programmed. Section 4: "Performance Tests" describes how the tests were undertaken and the results obtained. Section 5: "Test results analysis" analyzes the results obtained and evaluates each individual key characteristic for wideband transmissions. Section 6: "Conclusions" summarizes the reasons that determine the suitability of the configuration possibilities and the efficiency of the data transmissions.

3 The TST-SISTEMAS' Platform

Not all the information related to TST-Sistemas' mote [9] is going to be relevant. In this paper the focus is on the mote processing capabilities and the radio efficiency to send and forward messages. To program TST-Sistemas' platform there is an Application Interface (API) integrated in a development environment. It is critical to know what options are available, especially regarding the radio configuration and the data encoding. The API version used for these validation process was the 1.6. This API is provided by the company through its website [10].

3.1 Mote characteristics

Table 1 describes the main characteristics of a TST-Sistemas mote. The clock frequency is higher than

Table 1: TST-Sistemas' mote characteristics

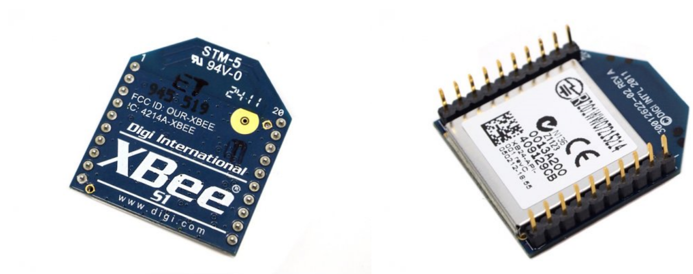
Microcontroller	32 bits STM with ARM Cortex-M3 core
Clock Frequency	72 MHz
Flash Memory	1 MB
RAM Memory	96 kB
Serial Interfaces	3 UART, 2 I2C, 1 SPI
Input / Output Ports	Up to 6 analog, up to 20 digital
Timer resolution	0.1 ms

other platforms tested in [3] what foresees a better performance in terms of computing speed.

3.2 Radio characteristics

The radio of a TST-Sistemas' mote is a separated radio-chip compliant with the 802.15.4 standard (figure 2). In the test we used an "XBee 802.15.4 - S1" [11]. Each radio-chip has a 64-bit MAC address written in the bottom side.

3.3 Mote characteristics



(a) Top view (b) Bottom view

Figure 2: XBee 802.15.4 - S1 radio-chip

This radio-chip is programmed using AT commands. These are the relevant configuration options [12] of the radio-chip and the analysis when using them through TST-Sistemas' API:

1. **Baud Rate (ATBD)** ATBD: AT command to set up the Baud Rate The baud rate defines the speed used to exchange data between the mote and the radio-chip. The communication between the

Table 2: Baud rate options

Baud Rates	
0	1200 bps
1	2400 bps
2	4800 bps
3	9600 bps
4	19200 bps
5	38400 bps
6	57600 bps
7	115200 bps

mote and the radiochip must be fast enough to deal with wideband applications. The API does not explicitly offer [13] the option to communicate to radio-chip using 115200 bps, but is useable. We use a baud rate of 115200 bps for the tests. Table 2 shows a list of the available baud rate configurations.

2. **API Operation (ATAP)** The API parameter defines the data structure that the mote and the radio-chip are using to communicate. The API specifies how commands, command responses and module status messages are sent and received from the module using a UART Data Frame. Table 3

shows a list of the available baud rate configurations. ATAP is the AT command to set up the APi

Table 3: API Operation

API operation	
0	API disabled
1	API enabled
2	API enabled with escaped characters

operation.Unfortunately the TST-Sistemas API does not support the value "ATAP = 2". This is a handicap when we work with raw data flows, e.g. audio or video, because all the characters are needed. If escaped characters are not allowed there is a risk of spoiling the integrity of the data. We use the value ATAP = 1 for the tests.

3. **MAC Mode (ATMM)** The MAC mode defines the behavior of the radio communications at the link layer. It can enable the use of ACK messages and also the use of an overhead. ACK are not needed and they would slow down the communications. ATMM is the AT command to set

Table 4: MAC Mode options

MAC Mode	
0	Digi Mode
1	802.15.4 (no ACKs)
2	802.15.4 (with ACKs)
3	Digi Mode (no ACKs)

up the Mac Mode (table 4). Unfortunately the TST-Sistemas API does not support the value "ATMM = 1" nor "ATMM = 2". This is a handicap regarding the integration of this platform with other 802.15.4 WSNs. If the rest of the network is not aware of this overhead they will not understand the messages [14]. We use the value ATMM = 3 for the tests because ATMM = 1 it is not available. Performance Tests Tests aim to show the response of the hardware in terms of radio-packet generation and relay times. Or, in other words, how fast we can send and forward data to transmit audio or video. Each test is described and followed by the results. The configuration used for all the tests is:

- Baud rate (ATDB) = 115200 bps
- API Operation (ATAP) = 1
- MAC Mode (ATMM) = 3

3.4 Transmission time

The first step is to measure the transmission time of one single message, making use of the following method:

1. The TST-Sistemas' mote's timer starts to count.

2. A radio packet is built, from the header to the payload.
3. The radio packet is sent to the radio.
4. The timer is stopped after the sending operation.
5. The value registered by the timer is sent through the UART for analysis.
6. Operations 1-5 are repeated 1000 times as fast as possible for a 5-byte payload.
7. Operations 1-6 are repeated increasing the payload size by 5 bytes each time until a 100-byte payload.
8. Average sending time are calculated for each payload size (5, 10, 15 . . . 90, 95 and 100).

The average transmission times (ATT) for each payload size are displayed on Figure 3 and exhibit a linear dependence of time to send a message vs the payload size.

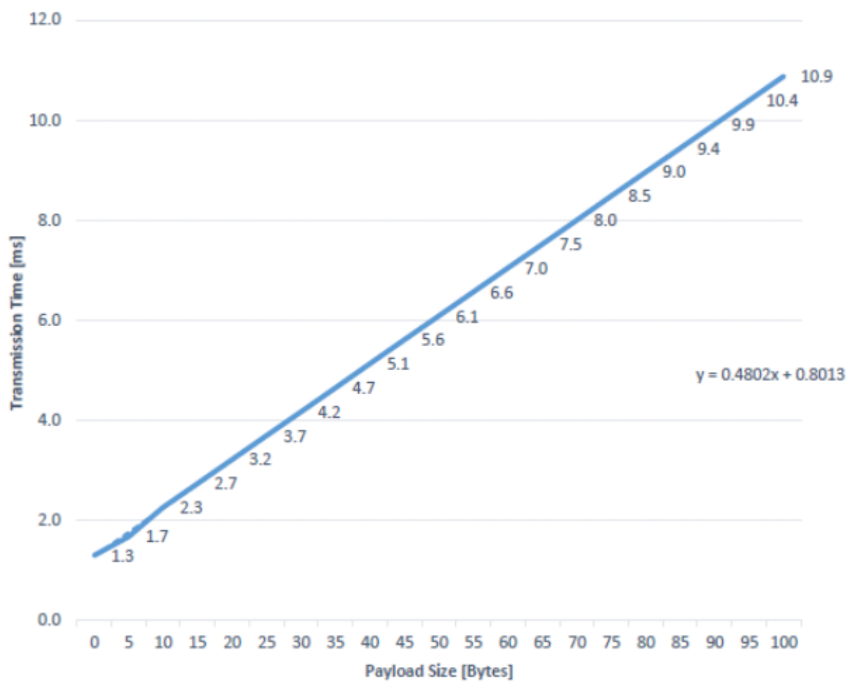


Figure 3: Average Transmission Time vs Payload Size

3.5 Two consecutive transmissions time

In wideband applications streams of data are sent, which implies consecutive transmissions. Therefore, we need to measure the time needed to send two messages in a row and compare it with the time needed to send one. This test evidences the efficiency for fast transmission rates. To measure the transmission time the following method is followed:

1. The TST-Sistemas' mote's timer starts to count.
2. A radio packet is built, from the header to the payload.

3. The radio packet is sent to the radio.
4. A second radio packet is built.
5. The second radio packet is sent to the radio.
6. The timer is stopped after the second sending operation.
7. The value registered by the timer is sent through the UART for analysis.
8. Operations 1-5 are repeated 1000 times as fast as possible for a 5-byte payload.
9. Operations 1-6 are repeated increasing the payload size by 5 bytes each time until a 100-byte payload.
10. Average sending time are calculated for each payload size (5, 10, 15 ... 90, 95 and 100).

The average two consecutive transmissions times (ATCTT) for each payload size are displayed on figure 4 and exhibit a linear dependence of time to send 2 consecutive messages vs the payload size.

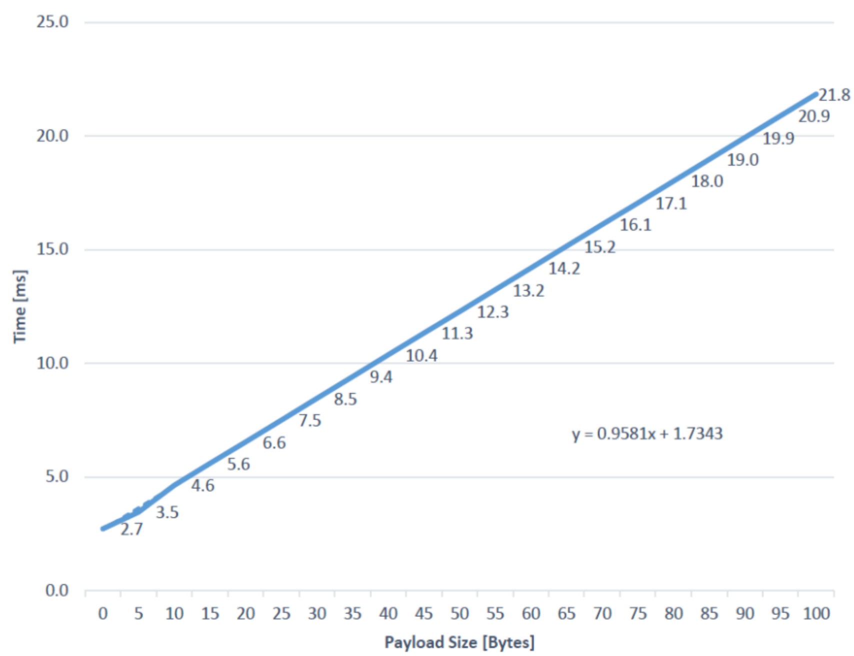


Figure 4: Two Consecutive Transmission Time vs Payload Size

3.6 Time between consecutive transmissions

An analysis can be undertaken to calculate the time needed between consecutives transmissions using the two previous tests. The absolute difference is calculated doing a simple subtraction. The minuend is the measured time for two consecutive transmissions and the subtrahend is the measured time for a transmission multiplied by 2. The relative difference is calculated doing the percentage of a simple division. The dividend is the absolute difference value and the divisor is the measured time for two consecutive transmissions. Observation are listed in table 5.

$$\text{Absolute difference } [ms] = ATCTT - 2 * ATT \quad (1)$$

$$\text{Relative difference } [\%] = 100 * \text{Absolute difference}/ATCTT \quad (2)$$

Table 5: Time between consecutive transmissions

Payload Size [Bytes]	ATT [ms]	ATCTT [ms]	Absolute Difference [ms]	Relative Difference [%]
5	1.67	3.46	0.12	3.47
10	2.26	4.64	0.12	2.59
15	2.72	5.6	0.16	2.86
20	3.22	6.55	0.11	1.68
25	3.69	7.51	0.13	1.73
30	4.17	8.46	0.12	1.42
35	4.65	9.42	0.12	1.27
40	5.13	10.37	0.11	1.06
45	5.61	11.33	0.11	0.97
50	6.09	12.28	0.1	0.81
55	6.57	13.24	0.1	0.76
60	7.05	14.2	0.1	0.7
65	7.53	15.15	0.09	0.59
70	8	16.11	0.11	0.68
75	8.48	17.06	0.1	0.59
80	8.96	18.02	0.1	0.55
85	9.44	18.97	0.09	0.47
90	9.92	19.93	0.09	0.45
95	10.4	20.88	0.08	0.38
100	10.88	21.84	0.08	0.37

3.7 Relay time

The relay time is the elapsed time from the moment the mote listens to a radio-message until the mote finishes its forwarding. To measure the relay time the following method is followed:

1. A first TST-Sistemas' mote sends a 5-byte payload radio-packet.
2. A second TST-Sistemas' mote receives and re-sends the same radio-packet.
3. A third mote, located in between the other two, collects the original radio-packets and the forwarded ones. The serial number of each packet and its timestamp is stored for later analysis.
4. Operations 1-3 are repeated ten times every 0.25 seconds.
5. Operations 1-4 are repeated increasing the size of the payload by 5 bytes until the payload size is 100 bytes.
6. Average relaying times are calculated for each payload size (5, 10, 15 ... 90, 95 and 100).

The average relay times for each payload size is displayed in figure 5.

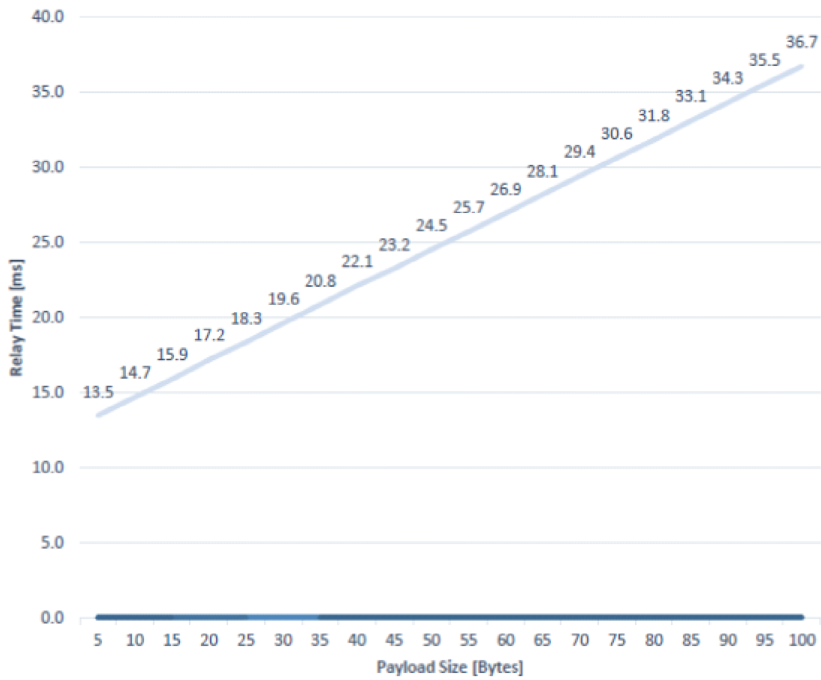


Figure 5: Average Relay Time vs Payload Size

4 Test results analysis

Test results can be separated in two categories: hardware performance and programming options.

4.1 Hardware performance

We compared TST.Sistema's motes with two of the most well-known and long-established commercial platforms: Waspmote and Telosb. The full comparison is included in the APPENDIX. This analysis focuses on the 100 byte radio packet. The reason is that longer radio packets have proportionally smaller header information and the efficiency is higher. These packets are the most interesting for wideband transmissions and therefore we analyze them in detail.

1. In terms of Average Transmission Time we can observe that:

   - A TST-Sistemas mote needs the 95.6% of the time a Waspmote mote needs to send a message.
   - A TST-Sistemas mote needs the 53.6% of the time a Telosb mote needs to send a message.

   In this category the performance is very good.

2. In terms of Two Consecutive Transmissions we can observe that:

   - A TST-Sistemas mote needs the 133.7% of the time a Waspmote mote needs to send two consecutive messages.
   - A TST-Sistemas mote needs the 86.3% of the time a Telosb mote needs to send two consecutive messages.

   In this category the performance is acceptable.

3. In terms of Relay Time we can observe that:

   - A TST-Sistemas mote needs the 60.1% of the time a Waspmote mote needs to relay a message.
   - A TST-Sistemas mote needs the 130.6% of the time a Telosb mote needs to relay a message.

   In this category the performance is acceptable.

The analysis of the hardware performance shows very acceptable results. These characteristics are desirables for wideband applications

4.2 Programming options

The main configuration options used are:

1. Baud rate (ATDB) = 115200 bps

   For wideband operations it is convenient to transmit the data as fastest as possible. We used a baud rate of 115200 bps to communicate the mote and the radio-chip. This speed works correctly. This characteristic makes it suitable for wideband applications.

2. API Operation (ATAP) = 1

   The API operation defines how we communicate to the radio-chip.

   - ATAP = 0 implies no API operation. The radio chip only sends or receives AT commands. This kind of transmission is too slow and complex to be suitable for wideband transmissions.
   - ATAP = 1 is the chosen value for the tests. The use of this configuration can introduce some control characters that can be misunderstood:
     - 0x7D - Escape control character. Indicates that next byte is escaped.
     - 0x11 0x13 - These bytes are software flow control characters

     . Audio encoding can generate any character so it is mandatory to escape control characters to avoid errors. If part of the audio is encoded using the value 0x7E and this is not escaped it will be recognized as a new packet regardless of the length. This characteristic makes the mote not suitable for wideband applications.
   - ATAP = 2 would be the right choice. It allows control character escaping. However, this is not a permitted value in the configuration options.

3. MAC Mode (ATMM) = 3 The MAC mode defines how the link layer behaves. This value controls the enablement of ACKs packets and the addition of some “extra bytes”. ATMM = 3 is the chosen value for the tests. It does not generate nor wait for an ACK packet, but it attaches 2 bytes to the header. The fact it does not introduce ACKs make it suitable for wideband communications. However the addition of two extra bytes generate compatibility problems. Motes running the 802.15.4 standard will think that the two last bytes of the header are, instead of that, the two first bytes of the payload. This generates interoperability problems that makes the platform less suitable and not fully compliant with 802.15.4 WSN

The analysis of the programming options shows some important lacks. These characteristics are unsuitable for wideband applications and for interoperability with strictly compliant 802.15.4 WSN.

5 Conclusion

After evaluating the performance of TST-Sistemas' motes for wideband communications using the 802.15.4 standard, these are the conclusions:

1. TST-Sistemas' platform shows a good performance in terms of processing data and transmitting it over a WSN, especially when increasing the size of the payload. These characteristics are very suitable for wideband applications.

2. TST-Sistemas's API is not complete enough to handle data properly. The fact that API mode escaping characters cannot be used makes it less suitable for wideband applications. This is an option that should be available. Later API version 2.2 is still missing this possibility.

To conclude, we can say that despite the good hardware performance the platform is not ready for wideband applications yet. This is due to the missing configuration options in the API

Acknowledgments

We would like to thank Dr. Congduc Pham, professor in the University of Pau, France for the guidance, contributions and previous work on this subject.

References

[1] EAR-IT Consortium, "EAR-IT." [Online]. Available: http://ear-it.eu/

[2] M. Rahimi, R. Baer, O. I. Iroezi, J. C. Garcia, J. Warrior, D. Estrin, and M. Srivastava, "Cyclops: In situ image sensing and interpretation in wireless sensor networks," in *Proc. of the 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys'05), San Diego, USA*. ACM, November 2005, pp. 192–204.

[3] C. Pham and P. Cousin, "Benchmarking low-resource device test-beds for real-time acoustic data," in *Proc. of the 9th International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities (TridentCom'14), Guangzhou, China, Lecture Notes of the Institute for Computer Sciences*, vol. 137. Springer International Publishing, May 2014, pp. 97–106.

[4] C. Pham, P. Cousin, and A. Carer, "Real-time on-demand multi-hop audio streaming with low-resource sensor motes," in *Proc. of the 9th IEEE International Workshop on Practical Issues in Building Sensor Network Applications (IEEE SenseApp'14), Edmonton, Canada*. IEEE, September 2014, pp. 539–543.

[5] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, pp. 921–960, 2007.

[6] S. Misra, M. Reisslein, and G. Xue, "A survey of multimedia streaming in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 10, 2008.

[7] S. Soro and W. Heinzelman, "A survey of visual sensor networks," *Advances in Multimedia*, vol. 2009, no. Article ID: 640386, 2009.

[8] TST-Sistemas, "English homepage," December 2014. [Online]. Available: http://www.tst-sistemas.es/en

[9] ——, "mote description and technical details," December 2014. [Online]. Available: Available:http://www.tst-sistemas.es/Docs/TSmoTe-EN.pdf

[10] ——, "Api version 1.6 description," December 2014. [Online]. Available: http://api.tst-sistemas.es/apitsmart-1.6

[11] Digi International Inc., "Xbee 802.15.4 (series 1). general information about the radio module specifications," December 2014. [Online]. Available: http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/point-multipoint-rfmodules/xbee-series1-module#specs

[12] Sparkfun, "Xbee 802.15.4 datasheet. full description. all configuration options," pp. 28–35, December 2014. [Online]. Available: https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf

[13] TST-Sistemas, "Api version 1.6. specific point about baud rate configuration," December 2014. [Online]. Available: http://api.tst-sistemas.es/apitsmart-1.6/structtsmart__xbee__802__config__device__t.html

[14] Digi International Inc., "Xbee 802.15.4 (series 1). clarification between digi mode and strict 802.15.4 mode," December 2014. [Online]. Available: http://www.digi.com/support/kbase/kbaseresultdetl?id=2158

Author Biography

David Guillén Jiménez received the B.S. degree in the University Miguel Hernández of Elche, Spain and the M.S. degree in the University Politécnica of Madrid, Spain. He started his career as a researcher in European projects during 2008 -2009, getting in contact with the IoT world. The three following years he worked in the industrial and energetic sectors as a software developer, system administrator and tester. Since the end of 2013 he came back to the research European projects and he worked as an expert for E.T.S.I. in the TTCN-3 testing language.

Franck Le Gall has a PhD in Physics and Telecommunications. After 5 years in technical development (he authored several patents and scientific publications) in the field of optoelectronic components , he spent a further eight years in the set-up and management of large international and multi-cultural industrial R&D projects. He directed more than 10 large scale projects and studies related to the evaluation and monitoring of innovation and technical programs, including recent coordination of the PROBE-IT support action (socio-technico-economic benchmarking of IoT deployments) and BUTLER IP (developing context awareness for horizontal IoT/M2M). In October 2013, in joined the company eglobalmark as COO. Beside office development, his activities focuses on increasing value of research projects outcomes through the implementation of standards conformance and interoperability validation schemes.

Appendix

Table 6: Average Transmission Time [ms] (ATT)

Payload Size [Bytes]	ATT TST [ms]	ATT Waspmote [ms]	ATT Telosb / Advanticsys [ms]
5	1.67		
10	2.26	2.35	13.7
15	2.72		14.1
20	3.22	3.36	14.5
25	3.69		14.8
30	4.17	4.36	15.2
35	4.65		15.6
40	5.13	5.36	15.9
45	5.61		16.3
50	6.09	6.37	16.7
55	6.57		17
60	7.05	7.37	17.34
65	7.53		17.8
70	8	8.37	18.1
75	8.48		18.5
80	8.96	9.38	18.9
85	9.44		19.2
90	9.92	10.38	19.6
95	10.4		20
100	10.88	11.38	20.3

Table 7: Average Two Consecutive Transmission Time [ms] (TCTT)

Payload Size [Bytes]	TCTT TST [ms]	TCTT Waspmote [ms]	TCTT Telosb / Advanticsys [ms]
5	3.46		
10	4.64	7.03	18.6
15	5.6	7.55	19
20	6.55	8.07	19.4
25	7.51	8.58	19.7
30	8.46	9.1	20.1
35	9.42	9.62	20.5
40	10.37	10.13	20.9
45	11.33	10.65	21.2
50	12.28	11.17	21.6
55	13.24	11.68	22
60	14.2	12.2	22.3
65	15.15	12.72	22.7
70	16.11	13.23	23.1
75	17.06	13.75	23.4
80	18.02	14.27	23.8
85	18.97	14.78	24.2
90	19.93	15.3	24.6
95	20.88	15.82	24.9
100	21.84	16.33	25.3

Table 8: Average Relay Time [ms] (ART)

Payload Size [Bytes]	ART TST [ms]	ART Waspmote [ms]	ART Telosb / Advanticsys [ms]
5	13.46		
10	14.68	18	16.40
15	15.86		17.40
20	17.18	21.43	17.90
25	18.34		18.7
30	19.62	26.06	20
35	20.82		20.5
40	22.1	31.32	20.6
45	23.23		20.7
50	24.5	37.26	21.9
55	25.69		22.5
60	26.94	41.34	22.7
65	28.15		24.3
70	29.4	46.06	25
75	30.6		24.8
80	31.79	50.74	25.5
85	33.08		26.3
90	34.29	54.45	27.8
95	35.51		27.9
100	36.7	61.03	28.1

The above tables show a comparison between Waspmote, Telosb and TST-Sistemas platforms. The values for the two first platforms were obtained from the paper “Benchmarking low-resource device test-beds for real-time acoustic data” [3]. Results are given for ATT (table 6), TCTT (table 7) and ART (table 8]).