University of Thi-Qar Journal Vol.12 No.3 SEP 2017
Web Site: **https://jutq.utq.edu.iq/index.php/main**
Email: journal@jutq.utq.edu.iq

Design and Implementation of Z-Wave Sniffer on TI-CC1125 Transceiver

https://doi.org/10.32792/utq/utj/vol12/3/9

Ali Kareem Abdulrazzaq
Collage of Engineering / Thi-Qar University

المستخلص:

مقترح من خلال هذا البحث تنفيذ متحكم رئيسي بالمنزل الذكي بالاعتماد على تكنولوجيا الاتصال Z-Wave و والمرسل والمستقبل TI-CC1125. المنزل الذكي يحتوي عدة انظمة منها منظومة الاضائة، منظومة الحماية، متحسسات الحرائق، السطيرة على درجة الحرارة والرطوبة، و واجهزة اخرى مربوطة عبر شبكة لا سلكية. بدلا من استخدام من متحكم منفصل لكل منظومة، نقترح في هذا البحث تصميم منظومة تحكم رئيسية لعرض ومتابعة جميع المنظومات الفرعية من خلال استنشاق الاشارات المرسلة والمستقبلة لهذه الانظمة. تكنولوجيا Z-Wave لتواصل الاسلكي اختيرت في هذا البحث. وحدة الارسال والاستقبال المستخدمة لهذا الغرض هي TI-CC1125 منتج شركة Texas Instruments (TI) في وضع الاستنشق (sniffer mode). البارمج المستخدمة في برمجة النظام هي Smart RF Studio 7 وthe Code Composer Studio (CCS) ، كلاهما منتج لنفس الشركة (TI). تعرض البيانات الملتقطة على الشاشة المثبتة على جهاز الاستلام فضلا عن عرضة على الحاسوب بعد ربط الجهاز عبر وصلة UART والتي ممكن بعد ذلك معالجتها لانجاز وضيفة معينة.

Abstract

This paper proposed a new smart home master control system based on Z-wave network and the TI-CC1125 transceiver. The smart home system includes lighting system, security system, smoke sensors, environment control and other equipment which are connected to a wireless communication network. Instead of having a specific controller for each system, in this paper we suggest having a master controller that monitor all systems status via sniffing their transceiver signals. Z-wave is the selected communication technology for this research. The Texas instrument lower power consumption transceiver (TI-CC1125) is used for this purpose in the sniffer mode of operation. Smart RF Studio 7 and the Code Composer Studio (CCS), both produced by Texas Instrument, are used for programming the Transceiver. The sniffed packets are displayed by the on-board LCD screen as well as the on PC application over UART interface, which can be processed later for a specific function.

Keywords: Smart Home, Z-Wave, RX Sniff Mode, Wake on Radio, and CC1125 Transceiver.

1. Introduction

The receiver current is very important parameter, especially for the battery powered devices. Sniff Mode is a very efficient concept to reduce the RX current. The Sniffer is a device that eavesdrops the network traffic by grabbing the traveling information over the communication medium. In this mode, the device wakes up from the sleep state to RX and go back to sleep periodically with a defined duty cycle [1].

This concept is strongly enhanced by CC1125 Transceiver due to its short settling time. CC1125 Transceiver is produced by Texas Instruments, with operating frequency in the range of the Industrial, Scientific, and Medical (ISM) frequency band. Home automation become very popular in the recent years, to integrates electrical devices (i.e. Comfort Function, Security Function, Medical etc.) in a house with each other, this means, many devices working in the same closed area, however these devices are not sending and receiving signals all the time, therefore the RX Sniff Mode would be the best choice to monitor the signals transferred between different devices. One of the most powerful home automation technologies is the Z-Wave protocol, which is very well standardized by the International Telecommunication Union.

Z-Wave is low power wireless communication technology mainly used in home applications. Z-Wave oriented to interconnect many functions dealing with security, lighting, climate conditioning door locks etc. to perform automation on these functions [2].

2. CC1125 Transceiver Evaluation Board

This work is mainly implemented on the low power RF transceiver, CC1125 Transceiver Evaluation Board (TRXEB) produced by Texas Instruments. CC1125 is fully integrated transceiver chip for low power and high performance. Figure 1 shows the PCB top view that handles the Sub 1GHz transceiver in terms of CC1125 Transceiver, microcontroller and some other peripherals [3].

CC1125 Transceiver is mainly uses the Industrial, Scientific and Medical radio bands (164-192MHz, 274-320MHz, 410-480MHz and 820-960MHz.), in this project and according to the European regulations, the 820-960MHz has been considered [4,5]. CC1125 supports hardware packet handling features including data buffering, link quality indication, and automatic low-power data receiving via Wake-On-Radio (WOR).

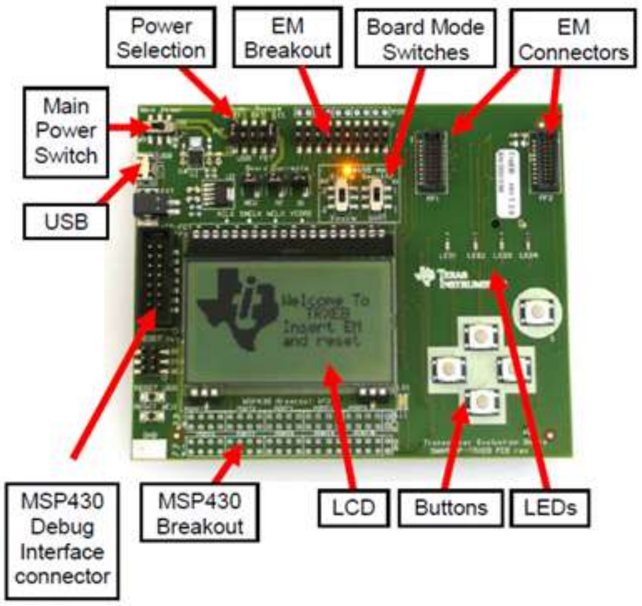


Figure 1: CC1125 Transceiver Evaluation Board with EM and LCD connected [3]

CC1125 supports hardware packet handling features including data buffering, link quality indication, and automatic low-power data receiving via Wake-On-Radio (WOR) [6].

3. Sniffer Mode Theory

The Sniffer Mode is especially needed in case it is not known when the transmitter sends a packet. This concept becomes more important for battery powered nodes, in which it is not required to be in RX mode all time, and it is sufficient to periodically wake-up from the sleeping mode and return to it if no valid packet is received. This periodic operation is adjusted by Wake-On-Radio feature [6,7].

3.1 RX Sniff Mode Usage

The normal RX Mode keeps the receiver always waiting for a packet to be sure that it receives the packet when the transmitter starts sending. In this mode the transmitted packet doesn't require long preamble (4-bits is sufficient), figure 2 illustrates this mode [8].

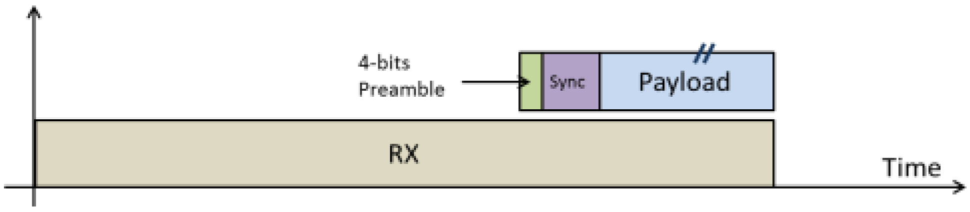


Figure 2: Ordinary RX Mode [8]

The transmitted packet requires longer preamble to be recognized by the receiver in the Sniff Mode, as it is important to receive 4-bits at least. Thereby the receiver can enter the sleeping mode and wake-up periodically, so that, and as it is illustrated in figure 3 the wake-up interval time is depending on the preamble length.

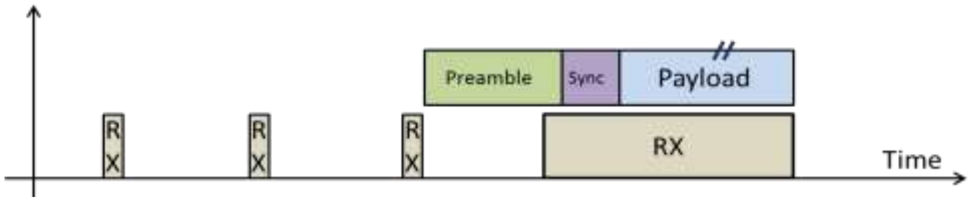


Figure 3: RX Sniff Mode [8]

4. Sniffer Software Design for the Proposed System

At first, the SmartRF Studio software tool has been used in RX Mode to find the suitable registers configurations of the CC1125 transceiver that match the Z-Wave RF PHY layer parameters. Then use these registers settings as a starting point to write the embedded code for designing the Sniffer using C language. The Code Composer Studio (CCS) is used as a development environment.

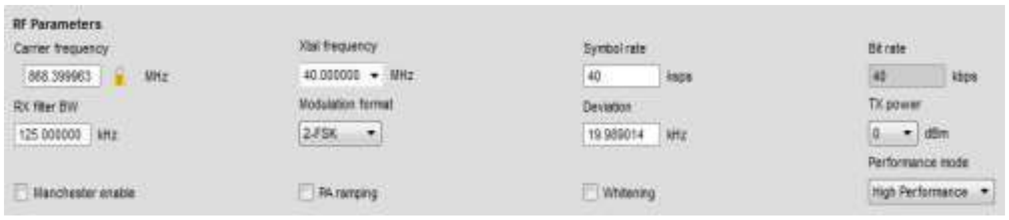
Z-Wave with 40Kbps data rate has been considered in this design as a source node, however, it is very well possible to use this design to sniff other Z-Wave devices packets after taking into account the differences in the PHY layer parameters (i.e. data rate, frequency, coding format, etc.).

4.1 SmartRF Studio Configuration

This section will cover the registers values selecting techniques, which can be divided into two parts. The first part includes the parameters that depend on the Z-Wave standard specifications. The second part deals with the unknown parameters of the transmitted frame which may be different from device to another.

4.1.1 SmartRF Configuration in RX Mode

Z-Wave technology has standard parameters that describe the PHY layer specifications. These specifications have to be considered during the configuration of the SmartRF to receive a correct packet. Figure 4 shows the first step configuration.



RF Parameters

Carrier frequency: 868.399963 MHz
RX filter BW: 125.000000 kHz
Xtal frequency: 40.000000 MHz
Modulation format: 2-FSK
Symbol rate: 40 ksps
Deviation: 19.989014 kHz
Bit rate: 40 kbps
TX power: 0 dBm
Performance mode: High Performance
☐ Manchester enable
☐ PA ramping
☐ Whitening

Figure 4: SmartRF Studio configuration

After configuring the SmartRF Studio with these values, it is very easily to read the value of the register(s) for every configured field. Table 1 shows the radio requirements for 40Kbps data rate packet and the corresponding registers values. It is important to note that some of the registers may correlate the setting of different parameters, where each register consist of subfields each field defined the specific parameter.

As a result, it is important to finish the entire configuration with the SmartRF Studio before considering the value of the registers as final values and copy it to the next design step.

4.1.2 Configuration of Other Registers to Receive Correct Packet

Configuring the SmartRF Studio with the given standards Z-Wave RF parameters is not enough to receive a correct packet, therefore some more registers have to be configured first to receive the correct packet (transmitted packet from the source node).

Table 1: Registers configuration according to Z-Wave specifications

Z-Wave Radio Specification	**Corresponding Register in CC1125**	**Register Value**
Frequency= 868.4MHz	FREQ2 FREQ1 FREQ0	0x56 oxD7 0x0A
Data Rate= 40Kbps	DRATE2 DRATE1 DRATE0	0x90 0x62 0x4E
Coding type: NRZ	MDMCFG1	0x46
Modulation Format: FSK	SYNC_CFG1 MODCFG_DEV_E	0x0B 0x05
Frequency Deviation = 20KHz	DEVIATION_M MODCFG_DEV_E	0x06 0x05

One of the important parameters to be configured is the sync word. Without choosing the write Sync Word it is not possible to receive the packet and read it from the First in First out (FIFO). The key point here is to configure the Sync Word registers SYNCx of (where x is from 1 to 4) the CC1125 transceiver with the preamble value; thereby the receiver will receive the whole packet including the preamble. Table 2 shows the value of the Sync Word registers values that have been used.

The bandwidth is set according to Carson's rule, which states that the bandwidth frequency is larger than or equal to the double of the sum of the data rate and the frequency deviation [10].

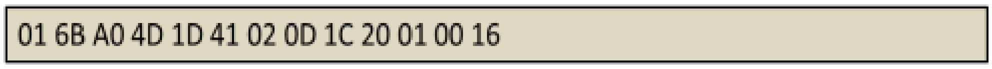
In this design the data rate is 40Kbps and the deviation is 20 KHz, therefore the BW can be chosen to be larger or equal to 120 KHz. The nearest value offered by the SmartRF Studio is 125 KHz. Up to this level of the configuration, and after run the SmartRF Studio in Packet RX testing mode, it is possible to receive packets. However, the setting of the other registers is tunable until receiving the correct packet by comparing the received data with the originally transmitted packet (assuming that the original packet is known).

Table 2: Sync Word registers values

Register	Value
SYNC3	0xAA
SYNC2	0xAA
SYNC1	0xAA
SYNC0	0xAA

4.1.3 Evaluation the Received Packet

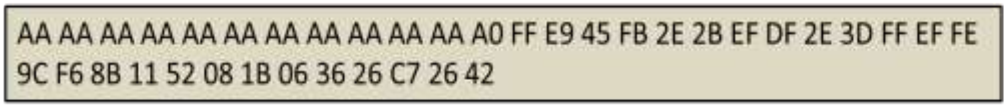
Here the received packet is evaluated and compared with the transmitted one to make sure that the received packet carries the expected information from the source node. The transmitted packet from the Z-Wave device is shown in figure 5.



```
01 6B A0 4D 1D 41 02 0D 1C 20 01 00 16
```

Figure 5: Transmitted packet from Z-Wave device

According to the setting of registers that described in the previous sections, the received packet is shown in figure 6.



```
AA AA AA AA AA AA AA AA AA AA AA A0 FF E9 45 FB 2E 2B EF DF 2E 3D FF EF FE
9C F6 8B 11 52 08 1B 06 36 26 C7 26 42
```

Figure 6: Received Packet

From the Z-Wave basics, the PHY layer frame is started with preamble pattern (the sequence of “1010” in binary) and then Start of Header Delimiter “00001111”. By excluding the preamble pattern and the start of the header field, the rest data is expected to be the payload information.

Indeed, after invert (One's complement) the payload data (data after the start header field) the originally transmitted packet is obtained. At this point, it is important to define how long the packet is expected to be, in order to read it and drop the rest data. These steps to extract the wanted information from the received packet can be easily implemented by a simple code in C language during future design steps which have been already accomplished during this work.

4.2 Sniffer Design with Code Composer Studio (CCS)

Code Composer Studio (CCS) is an integrated development environment (IDE) for Texas Instruments (TI) microcontrollers and application processors. It is a Windows and Linux application used to develop and debug embedded applications [11].

The second step of the Sniffer design is implemented by the CCS. The Design in this level is represented by the code that was written in the C language. The registers configuration from the SmartRF Studio is used, as well as the real Sniffer function, Wake on Radio (WOR) feature is initiated, and registers that govern the duty cycle of WOR mode have been set.

4.2.1 Software Code Design

Figure 7, illustrate the Sniffer code in terms of the flow chart, in which the main design steps have been introduced. In this code design, the initiation of the clock system, peripherals, and their interfaces is done in the main function. As soon as the Sniff Mode is activated, the Sniffer source code started to be executing; the SPI interface to the transceiver is going to be initiated, then this followed by the registers configuration, and activation of the Sniff Mode. The Transceiver can enter a low power mode with an infinitely loop or implement another piece of code and look for new packets with defined time interval. However, termination condition for this mode can be defined to enter another mode or return to the main menu.

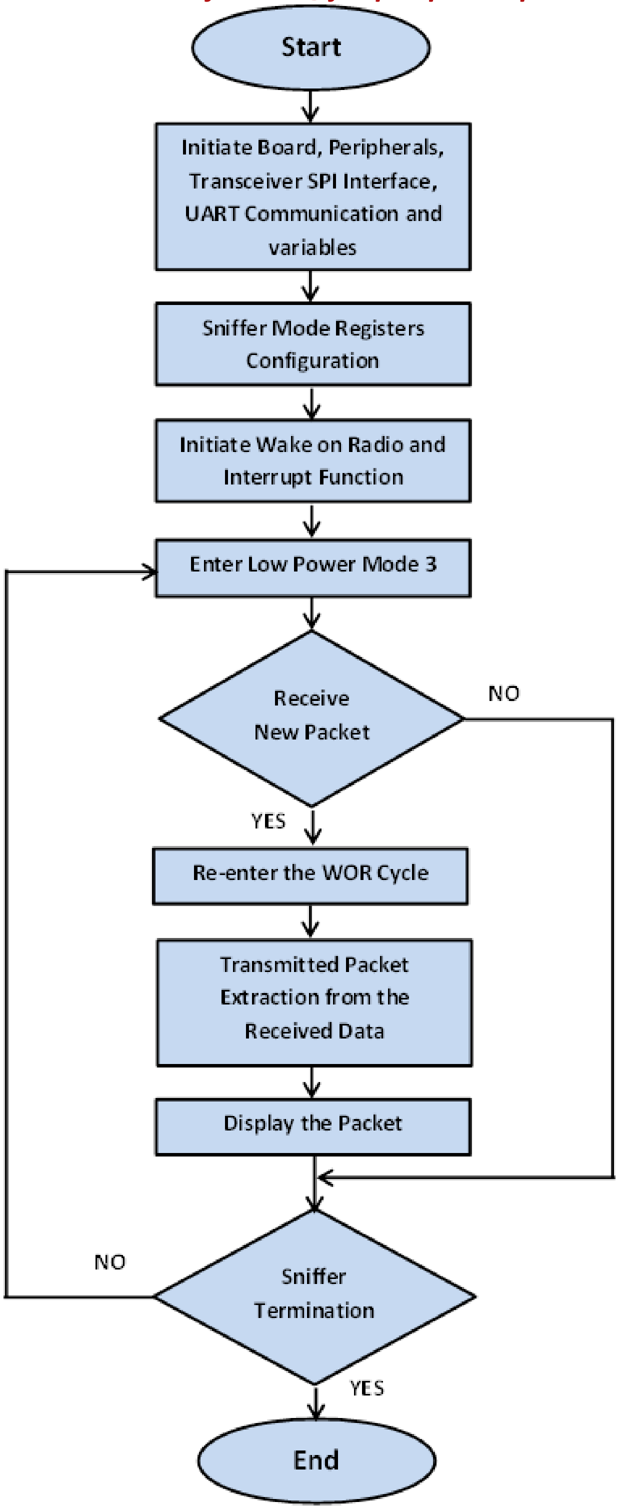

Start
Initiate Board, Peripherals, Transceiver SPI Interface, UART Communication and variables
Sniffer Mode Registers Configuration
Initiate Wake on Radio and Interrupt Function
Enter Low Power Mode 3
Receive New Packet
NO
YES
Re-enter the WOR Cycle
Transmitted Packet Extraction from the Received Data
Display the Packet
NO
Sniffer Termination
YES
End

Figure 7: Code design flow chart

4.2.2 Hardware Initialization

As the first step in the code, it is obligatory to initiate the clock system, peripherals (i.e. LCD display, Buttons, LEDs, etc.) which are used during the code implementation, variables, as well as the transceiver SPI interface. LCD display also accessed by SPI interface, which is shared with the SPI Flash, therefore it is important to disable the communication with the flash and initiate it with the LCD. Table 3 shows the functions that have been used in this section.

Table 3: Initiation Functions

Purpose	Code Function
Select 32-kHz watch crystal oscillator (XT1 mode)	halMcuStartXT1();
Initiate the MCU with clock speed	halMcuSetSystemClock (systemClockSpeed)
Initiate the LEDs and turn them off	halLedInit();

Initiate the buttons and enable their timer	halButtonsInit();
	halButtonsInterruptEnable();

Initiate the LCD and clear the screen	halLcdInit();
	halLcdClear(0);

Initiate the transceiver RF SPI interface	trxRfSpiInterfaceInit (systemClockSpeed);
Initiate the UART communication	UARTInit();

4.2.3 Registers Configuration

The special function has been used to configure the transceiver with the register values required for the sniffer function. Registers values are defined as a constant table, each register has its own memory address which is defined in one of the header files. The function will read this address and write the register value to it.

4.2.4 Initiate the Wake-On-Radio Feature

Initiate the WOR feature includes calling the interrupt service routine which is responsible for configuring the registers that govern the sniffer function. Table 4 shows the functions that have been used to connect the RX interrupt service routine to the interrupt function, and enter a WOR cycle including interrupt clear flag and enable transceiver interrupt. By these two functions, the Radio switched from idle to WOR mode. Table 5 shows the registers values and the corresponding function.

Table 4: Connect ISR to interrupt and enter a WOR cycle functions

Purpose	**Code Function**
Connect ISR function to interrupt from GDO0 (P1.7)	trxIsrConnect(&radioRXISR);
TRX from Idle to WOR	trxIdleWor( );

Table 5: Initiate Wake-On-Radio registers setting

Register	**Value**	**Function**
AGC_CS_THR	0x19	Set AGC carrier sense threshold
AGC_CFG0	0xCF	Set RSSI valid count to minimum: 2
AGC_CFG1	0xA9	Set AGC window size to 8 samples Set AGC settle wait time to 24 SAMPLES
SETTLING_CFG	0x03	Turning off the Auto-Calibration
WOR_CFG1	0x08	Setting WOR mode to NORMAL_MODE & WOR timer resolution to HIGH_RESOLUTION

WOR_EVENT0_MSB	0x02	Setting Event0 timeout, to 20ms
WOR_EVENT0_LSB	0x08	
WOR_CFG1	0x0E	Setting Event1 timeout, to 1ms
WOR_CFG0	0x24	Enable RC calibration to send calibration storb
	0x20	Disable calibration
RFEND_CFG1	0x0E	Setting RX timeout to never
RFEND_CFG0	0x09	Setting RX termination of RSSI

This followed by updating the LCD display indicating that the transceiver entered the new mode.

4.2.5 Receiving Packets

After entering the WOR cycle the transceivers is ready to receive new packets. As soon as a new packet has been received the data read from the FIFO and stored in a matrix buffer.

Re-entering the WOR mode to receive new packet is done by using the same function mentioned before for this purpose. However, this implies the use of multi-dimensions array to store the received packet(s) that may be received during the evaluation and displaying the previous one. The received packet has to be prepared to extract the originally transmitted information by the Z-Wave device, according to the steps defined in section 4.2.1. To monitor the data that have been received there is more than one possibility. One option could use the LCD display which is available with CC1125 transceiver board. Another option is to use UART serial communication with PC to monitor the data one the PC, using a software application like Putty.

5. Conclusion

In this research work, the Z-Wave Sniffer has been designed and implemented on CC1125 Transceiver. The design done in two steps; first, the SmartRF used to find the right registers setting then these registers values exported to be used in the Sniff Mode configuration using the Code Composer Studio.

One of the important aspects of the design is choosing the sync word same as the preamble of the packet to be received, thereby the packet will be received with the preamble pattern. Received packet entered in a special function to extract the originally transmitted information from it. This function includes scanning the received binary sequence looking for the starting header delamination "00001111", then exclude the data before it (the preamble pattern). The rest data will represent the original transmitter information after doing the one's complement. The received message has been passed to a computer where it can be analyzed and understand the message transmitter status.

6. Future Work

As future work, the same research principle can be considered to design a Sniffer for other smart home protocols. The same design can be used to sniff multi packets with different data rates simultaneously; this can be implemented by more than one way, like oversampling with high data rate, or switching between two registers configuration by using timer interrupt; however, this required accurate timing calculation which depends on the preamble length of the transmitted packets.

7. References

[1] Huo, L. (2014). A Comprehensive Study of Passive Wake-up Radio in Wireless Sensor Networks. Delft University of Technology.

[2] Mendes, T., Godina, R., Rodrigues, E., Matias, J., & Catalão, J. (2015). Smart Home Communication Technologies and Applications: Wireless Protocol Assessment for Home Area Network Resources. Energies, 8(7), 7279–7311. https://doi.org/10.3390/en8077279

[3] Texas Instruments. SmartRF Transceiver Evaluation Board "TrxEB" User's Guide. Retrieved March 15, 2014, from http://www.ti.com/lit/ug/swru294a/swru294a.pdf

[4] Mendes, T., Godina, R., Rodrigues, E., Matias, J., & Catalão, J. (2016). Smart Home Communication Technologies and Applications: Wireless Protocol Assessment for Home Area Network Resources. Energies, 8(7), 7279–7311. https://doi.org/10.3390/en8077279

[5] Mendes, T., Godina, R., Rodrigues, E., Matias, J., & Catalão, J. (2015). Smart Home Communication Technologies and Applications: Wireless Protocol Assessment for Home Area Network Resources. Energies, 8(7), 7279–7311. https://doi.org/10.3390/en8077279

[6] Texas Instruments. Ultra-High Performance RF Narrowband Transceiver (CC1125). Retrieved March 15, 2014, from http://www.ti.com/lit/ds/swrs120a/swrs120a.pdf

[7] Macii, D., Ageev, A., & Somov, A. (2009). Power consumption reduction in wireless sensor networks through optimal synchronization. 2009 IEEE Intrumentation and Measurement Technology Conference, I2MTC 2009, 1352–1356. https://doi.org/10.1109/IMTC.2009.5168665

[8] Texas Instruments. CC112X/CC1175 Low-Power High Performance Sub-1 GHz RF Transceivers/Transmitter. Retrieved March 15, 2014, from http://www.ti.com/lit/ug/swru295c/swru295c.pdf

[9] Kumberg, T., Schink, M., Reindl, L. M., & Schindelhauer, C. (2017). T-ROME: A simple and energy efficient tree routing protocol for low-power wake-up receivers. Ad Hoc Networks. https://doi.org/10.1016/j.adhoc.2017.02.003

[10] Kumberg, T., Schink, M., Reindl, L. M., & Schindelhauer, C. (2017). T-ROME: A simple and energy efficient tree routing protocol for low-power wake-up receivers. Ad Hoc Networks. https://doi.org/10.1016/j.adhoc.2017.02.003

[11] Texas Instruments. Code Composer Studio IDE. Retrieved March 15, 2014, from http://www.ti.com/lit/ml/sprue40/sprue40.pdf