A Low-Power, Tricky and Very Easy to Use Sensor Network Gateway Architecture with Application Example

Susanne Kenner
and Klaus Volbert

Technical University of Applied Sciences
Regensburg, Germany
Email: {susanne1.kenner, klaus.volbert}@oth-regensburg.de

***Abstract*—Wireless Sensor Network gateways are key components to connect smart environments with Internet of Things and to support Wireless Sensor Network applications. With regard to the increasing amount of smart devices, communicating with each other in Home area networks or over the Internet, low cost and low power devices are desirable. In this paper, we present such a low cost and low power gateway and system architecture with an example application in the Smart Home environment to control smart devices with a mobile application. We show an easy way to use low cost and low power hardware to expand the existing home network just with an USB-Stick without any extra power supply. The gateway we developed needs only 0.3 % of the average power consumption of the used network router.**

Keywords–Sensor Networks; Sensor Network Architecture; Sensor Network Gateway; Low-Power Devices; Internet of Things.

I. INTRODUCTION

The Internet of Things (IoT) is the next Internet generation, which expands the present Internet and machine-to-machine communication of smart and active devices to a larger global network of connected devices and sensors. 24 billion devices are expected in 2020 [1], communicating with each other and making smart decisions based on huge amounts of data. For this reason, many research works are in process in the IoT field. The IoT consists of different sectors like home automation systems, health applications, logistic applications, warehousing, industrial production, smart grid and many other sectors. The Radio Frequency Identification (RFID) technology and Wireless Sensor Networks (WSN) are key technologies and wide-spread in the IoT sectors to collect sensor data and control and connect devices to the Internet [1]. WSNs are used for many applications. In health care sector, they can be used to analyze the sleep of people which have an infrequent breathing during sleep, for fall detection systems or a better and faster diagnostic through doctors. In the agriculture industry, WSNs can help to increase the harvest by tracking moisture, nutrients, lightning and temperature values. Energy monitoring, weather monitoring and use cases in the industry are only a few application areas. In [2], the authors give a detailed review of the named application areas. An example work for an agriculture industry use case is presented in [3] and in [4], an application in the Smart Home area is presented, which combines the RFID and WSN technologies. WSNs consist of many nodes, which track data from sensors and transmit them to a gateway. The gateway collects data from all sensors and often visualizes and analyzes them. In the IoT the gateway is a key component, which connects the WSNs to the Internet and therefore, the use cases named above were taken to the next level. The amount of nodes in WSNs and the expected amount of devices in the IoT are one of the reasons for developing low cost and low power devices in this environment. Another reason is the transition in energy policies, especially in Europe. The increasing power demand and the transition from conventional to renewable energy sources have an impact on the stability of the power grid [5]. Low power IoT gateways can support Smart Grid applications to stabilize the power grid or to analyze smart environments. For example, they can be used as gateways for collecting energy values in a low voltage grid as described in [6] or for monitoring data from energy analyzers in a small smart grid scenario as presented in [7]. Additionally, low power devices contribute to the reduction of the CO_2 emission, which was decided by the European Parliament [8]. In this paper we present a Wireless Sensor Network gateway architecture with focus on low cost hardware and low power functions.

This paper is structured as follows. Section II introduces the gateway design idea and names some similar works on this topic. Section III describes the eZ430 Development kit we used for developing the gateway and in Section IV the gateway architecture is explained in detail, followed by an application example in Section V. Section VI shows the power consumption of the gateway and finally, Section VII concludes this paper.

II. MOTIVATION AND RELATED WORK

Smart devices like smart fridges, smart washing machines, automatic controllable heating systems, window shutters and switchable outlets, controlled by a radio control unit, are integrated in modern households. Several solutions for home automation systems like RWE Smart Home [9], Telekom Smart Home system [10] or HomeMatic [11] are offered in the market but each manufacturer develops their own closed system with a special communication protocol and gateway device. Additionally, the systems often are not power optimized and need an additional power supply. Besides the named Smart Home systems, the Startup Codeatelier first followed the same approach as we, but they do not describe how they couple the USB-Stick with the router [12]. In this paper, we show how simple a router could be expand by an USB gateway to get WSN functionality at the home network and connect them to the IoT. There are existing several gateway architectures in literature as named in the following. In [13] the authors describe the design and implementation of an embedded Linux gateway for monitoring smart home health system data. A flexible gateway with different communication interfaces is presented in [14]. The gateway architecture modules are replaceable depending on the application running at the system. Further gateway architectures and systems are presented in [15] and [16]. Another interesting approach show the authors in [17] by using a Smartphone as a (mobile) gateway. Combined with an Arduino hardware board the Smartphone receives data from sensors and transmits data to the Internet. However, this approach needs an extra hardware like the Arduino board, with extra power supply, for connecting sensors and transmitting data to the Smartphone.

In contrast to the referred works above, we present a gateway architecture without extra power supply by integrating the gateway in the existing home network infrastructure. Almost every household uses a router with an USB-Port, which provides USB devices like printers and external hard disks in the network. We exploit this feature to connect a USB-Stick as gateway for a home automation system via the USB-Port of the router with the home network. Through the usage of low cost hardware and free available software we present a low cost and very easy to use sensor network gateway. The most similar approach to our one is presented in [18]. The authors also develop a gateway, which communicates over the USB-Interface with an USB-Host like a PC or RaspberryPi to monitoring data from a WSN. Different USB communication classes are possible and they tested the system with the Virtual Serial Port and Ethernet Emulation classes. These USB communication classes need a system driver on the host which means that the usage is restricted on devices with driver support. In our system the USB Mass Storage class, which needs only a standard driver provided by all USB host devices, is used. Therefore, it is possible to connect our gateway to any host device without needing a driver. On this basis it is possible to expand existing network devices like routers in the home network so that they operate as the interface to the IoT as described in this paper. We combine different known technologies to design a gateway, which emulates a file system and communicates over the USB port of a router with the network.

III. eZ430-Chronos development kit

The eZ430-Chronos kit is a development kit from Texas Instruments including a programmable watch, an access point with radio module and a programming interface as shown in Figure 1. The kit is available for the following different ISM/SRD-Band frequencies: 915 MHz (America), 433 MHz (worldwide usable) and 868 MHz (Europe and India). The microcontroller in the watch case is a CC430F6137 with integrated sub-1GHz wireless radio module based on the CC1101. The eZ430 consists of a 96-segment LCD Display, a pressure sensor, three-axis accelerometer, temperature and battery voltage sensor. Additionally, a heart rate monitor can be connected, communicating over the BlueRobin protocol [19]. The Access point (middle usb device in Figure 1) is an USB-based microcontroller and consists of a MSP430F5509 microcontroller and a CC1101 radio module for communication with the watch [20]. There are two different access points available. We use the newer release of the eZ430 Chronos kit with the white access point. The older eZ430 Chronos kit with black access point contains an CC1111 radio module. The MSP430F5509 is an ultra-low-power 16-bit RISC CPU with USB interface, 24 KB flash memory and 6 KB RAM. The supply voltage is between 1.8 to 3.6 VCC. It supports one active (AM) and 6 low power (LPM0 - LPM4.5) operating modes. Table I shows power consumption according to interesting operating modes [21]. Due to the characteristics like low power (our application mainly runs in LPM 3), low costs and the free availability of hard- and software, we use this kit in our system. For developing a gateway for home automation systems we only need the access point of the kit.

Figure 1. Texas Instruments eZ430 Chronos - Development Kit

TABLE I. MSP430F5509 OPERATING MODES AND POWER CONSUMPTION [21]

Operating mode	Description	power consumption
AM	All clocks active	115 - 195 μA/MHz
LPM 3	Standby mode - CPU, MCLK, FLL loop control, DCOCLK, DCO's generator disabled	1.4 - 2.1 μA
LPM 4	CPU, MCLK, ACLK, FLL loop control, DCOCLK, DCO dc generator disabled, Crystal oscillator is stopped, Complete data retention	1.1 μA
LPM 4.5	Internal regulator disabled, No data retention	0.18 μA

IV. Gateway Architecture

In our work we create a low power gateway for home automation. This gateway should be integrated in existing network infrastructure to save energy and costs. To realize this, we designed the architecture shown in Figure 2. The access point described above was programmed as a gateway and coupled with a FRITZ!Box router. The gateway communicates over the USB interface with the router by emulating a file system and is mounted as USB flash device in the integrated NAS-System. At the other side the gateway is able to communicate over the radio module with different devices at home to monitor sensor data or control them. The FRITZ!Box router supports data access over FTP, HTTP and the Server Message Block (SMB) protocol, so it is possible to request data by different clients. The router is also able to make the data available through the Internet by registering the FRITZ!Box router at

 ISBN: 978-1-61208-490-9

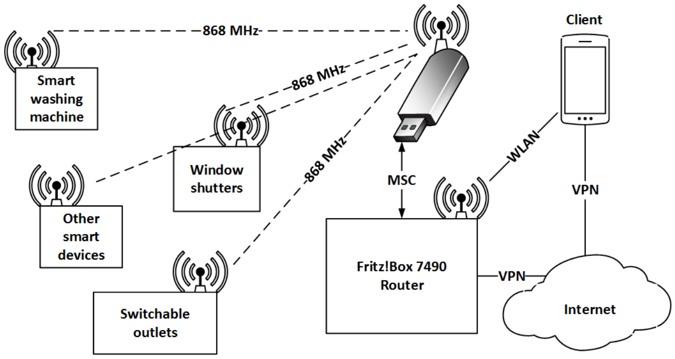

Figure 2. System architecture

the AVM domain service. In our scenario we implemented an Android application which uses the SMB protocol to request data. Additionally it is possible to set up a VPN (Virtual private network) connection between the FRITZ!Box router and the Android client, so we communicate over a secure channel from outside the home network.

A. USB Mass Storage Class (MSC)

The MSC is one of the USB device classes, which are defined for the communication between an USB-Device and an USB-Host. The device classes summarize devices with similar functions to standardize the communication interface. Each host provides a standard driver for each defined USB device class and so also for MSC devices. The MSC profile provides an interface for reading and writing file systems [22] [23].

Texas Instruments provides an example code package for training and testing with the different USB communication classes for free. This package is available for Windows, Linux and MAC systems and can be downloaded at [24]. The package demonstrates the USB-Development with MSP430 devices and provides the USB communication device class (CDC), human interface device (HID) and mass storage class (MSC). We used the included example for file system emulation `M1_FileSystemEmulation` and make some changes for our scenario. Some hardware settings were adjusted to the MSP430F5509 of the access point. The emulated file system in the example with a size of 83456 byte is too big for the available hardware. As described in Section III the access point only has a flash memory of 24 KB. Approx. 18 KB is needed for the program code, so we changed the emulated file system size to 4 KB by creating an FAT12 file system with 7 sectors, whereas each sector has a size of 512 byte. A FAT12 file system normally consists of 2 file allocation tables (FATs), but due to the limited flash memory size we set the value for provided FATs in the BIOS Parameter Block (BPB) to 1. The first three sectors are used by the FAT12 file system, sector 4 includes the "data_log.txt" file and the remaining sectors are free at the beginning. By connecting the gateway to a host, the free available space will be used by the operating system. At the FRITZ!Box router this makes some trouble for the reason that the FRITZ!Box router indexing the files and overrides the flash memory at the gateway. In addition, the FRITZ!Box sets the mounted flash drive as write-protected, so it is impossible to exchange data with the gateway. To avoid this situation we create one file for each sector so the free available space at the gateway is zero and the operating system cannot write index or temp data on sectors. Therefore, the FRITZ!Box do not set the mounted flash drive as write-protected and it is possible to exchange data between gateway and router via the file system.

B. Communication

There are two communication ways implemented as shown in Figure 3. At the first one the client writes data via the SMB protocol over the network into the file "data_log.txt" mounted at the router NAS for controlling smart devices at home network. The gateway reads the flash memory area for this file every 2 seconds, controlled by a timer, and read out the command codes. Independent on these codes it controls the smart devices by transmitting a radio sequence. The communication in the opposite direction is based on the smart devices. The smart devices like power outlets send data

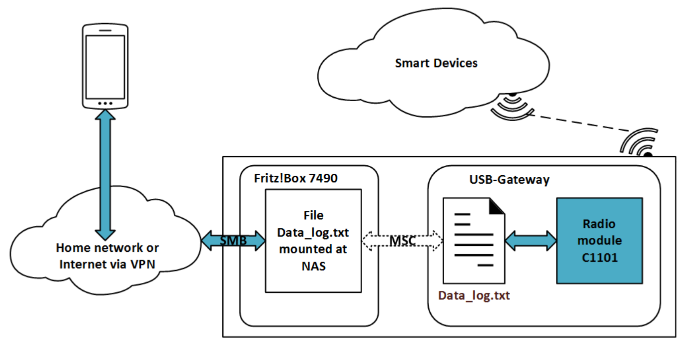

Figure 3. Gateway details

via radio sequence to the gateway. Subsequently the gateway software transforms the measurement in a human-readable value and writes this value into the flash memory area of a file. On the reason that the USB protocol is host driven the router as host takes no note of the changed file. The host caches the data read at USB plug in for the reason the host does not expect changes. A reset triggered through the Watchdog timer of the gateway forces the host to re-read the file system. In the case of using the FRITZ!Box router the remount of the USB flash drive has a duration time of $\sim$ 10 seconds. After this time the files are available again. To realize the radio communication between the gateway and the smart devices we integrate parts of the radio code from the original access point code example into the emulated file system example.

V. APPLICATION EXAMPLE

We test the low power gateway and the architecture described above in a small test environment. At this a little smart home environment with a switchable power outlet and 7 window shutters is used as proof of concept scenario. The switchable power outlet is a FS20 ST-4 with a Breaking capacity of 16 A at 230 V and a built-in timer to switch devices on or off depending on time [25]. The FS20 RSU-2 window shutters are flush-mounted sockets also with a built-in timer for time-depending operations [26]. The devices are controllable through the FS20 protocol via radio at the 868 MHz SRD band. This protocol is only used as example to demonstrate the low power gateway. In the following the FS20 system is described and the example application is presented.

 ISBN: 978-1-61208-490-9

A. FS20 System

The FS20 system is a smart home radio control system with many different devices in the home automation sector [27]. The system is distributed by the companies Conrad Electronics and ELV, which is the copyright holder of the FS20 protocol and devices. There are controllable devices like switchable power outlets, electric window shutters, rain sensors, temperature sensors, etc. These devices are controllable via transmitters like radio remote control or gateway devices. The receivers can be integrated in the home automation system through the following described addressing scheme. It consists of 4 different address types and a home code. Each remote control or gateway has its own home code address with a length of 8 digits, so it is possible to run more radio systems side by side in the same environment. The receivers are addressable with up to 4 addresses with a length of 4 digits, one for each address type. There are 225 single, 15 function group, 15 local master addresses and 1 global master address. With function group addresses it is possible to control several receivers of the same kind with one command. The local master address forms spatial groups, e.g., to switch all devices in one room on or off by sending one command. The global master address also forms one group with several receivers, e.g., all devices in a house [26]. The FS20 communication protocol operates at a frequency of 868.35 MHz and uses the amplitude modulation [25]. Table II shows some different commands for controlling smart FS20 devices as described in [26]. According to the device type some commands have different effects. At window shutters all commands for dimming up and for switching on are used to switch the device on and run them up. All commands for dimming down and for switching off are used to run the windows shutters down and switch them off. The commands can be expand by an extension byte. With this one it is possible to control the devices depending on time settings as described in [26]. A more detailed protocol description can be found at web.

TABLE II. FS20 COMMAND CODES [26]

FS20 command	Description
0x00	device off
0x01	device on, 6.25% brightness level 1
0x02	device on, 12.5% brightness level 2
...	...
0x0f	device on, 93,75% brightness level 16 (max)
0x10	device on
0x11	device on, set to last brightness level
0x12	toggle between off and on with last brightness level
0x13	dim up to the next brightness level
0x14	dim down the brightness level
0x15	changing between dim up and down
0x18	device off, depending on timer
0x19	device on, depending on timer
0x1a	device on, last brightness level, depending on timer

B. Proof of concept application

For the proof of concept application, the access point of the eZ430 Chronos additionally was configured for FS20 data transmission. The main settings for the CC1101 radio module is listed in table III. For each device we create a control frame, for example to switch the power outlet on or off. The control of the devices and the access to the file system at the gateway is encapsulated through an Android application. The Android client is connected to the FRITZ!Box router via home network or over the Internet as described above. Each smart device is controllable through one button, which switches the power outlet on or off or pulls up and down the window shutters. The client application saves the commands of the buttons in the ”data_log.txt” file, which are not visible for the user. We use 1 Byte for each command as described in table IV. After these 18 bytes there are 6 more bytes available. For example these could be used for pulling up and down the window shutters at set times, for time synchronization between gateway and USB-Host or for additional devices. The values saved at ”data_log.txt” are checked periodically by a timer. Every two seconds the gateway checks if the values at the flash memory of the ”data_log.txt” are changed. According to the changes the gateway triggers the equivalent action and transmits the control frame for the command via the radio module to the device. The command structure could be optimized by using just one bit for one command or by using XML-files. Further protocol extensions are possible to support different devices, communicating via other protocols, e.g., z-Wave [28]. Through the separation of control and data flow efficient low power energy monitoring is possible. As described above the gateway is able to write received data from radio devices into the flash memory. Received energy values from energy analyzers can be saved in a second file on the emulated file system and read out via the MSC Profile from the host respectively the Smartphone application.

TABLE III. CC1101 MAIN SETTINGS

Register	Value	Description
MCSM0	0x18	XOSC-Timeout 149-155μs, calibration at IDLE to TX
PKTLEN	0x23	Packet length: 35 Bytes
PKTCTRL0	0x04	Fixed packet length modus, CRC calculation disabled, data-whitening disabled
FREQ2 FREQ1 FREQ0	0x21 0x65 0xE8	Frequency: 868,35 MHz
MDMCFG4 MDMCFG3	0x87 0x93	Data rate: 5.00107 kBaud $\sim$ 5 kBaud
MDMCFG2	0x30	Preamble disabled, Synchronization word disabled, Amplitude modulation
AGCCTRL2	0x43	Highest DVGA-asset can not be used
FREND0	0x11	Amplitude modulation: Setting for transmitting a 1 in PATABLE(1)

TABLE IV. APPLICATION CONTROL COMMANDS

Position in ”data_log.txt”	Operation when Byte changed
Byte 1 / Byte 2	Pulls up/down all window shutters
Byte 3 / Byte 4	Pull up/down window shutter 1
...	...
Byte 15 / Byte 16	Pull up/down window shutter 7
Byte 17 / Byte 18	Turn power outlet on/off

VI. Power consumption

The MSP430F5509 microcontroller and the CC1101 radio module are low power devices as described above. In the following, we show typical current consumption values for the used hardware. Based on these values we show the power consumption of our developed gateway. The total amount of power consumption consists of the consumption of the MSP430F5509, depending on clock frequency and power mode, and the consumption of the radio module, depending on radio power output and operation mode. As shown in table

TABLE V. POWER CONSUMPTION MSP430F5509

Clock Frequency	1.8 V Min / Max mW	3.0 V Min / Max mW	3.6 V Min / Max mW
8 MHz	1.66 / 2.81	2.76 / 4.68	3.31 / 5.62
16 MHz	3.31 / 5.62	5.52 / 9.36	6.62 / 11.23
25 MHz	5.18 / 8.78	8.63 / 14.63	10.35 / 17.55
LPM3	0.0025 / 0.0038	0.0042 / 0.0063	0.005 / 0.0076

I the current consumption of the MSP430F5509 in active mode is between 115 and 195 μA/MHz, and in LPM3 mode between 1.4 and 2.1 μA. Depending on these values, table V shows the power consumption at different supply voltage levels and clock frequencies for the active operation mode and LPM3. The maximum power consumption with frequency of 25 MHz and a supply voltage of 3.6 is 17.55 mW and in LPM3 mode the maximum value is 0.0076 mW. For the CC1101 radio

TABLE VI. POWER CONSUMPTION CC1101

Supply Voltage	IDLE	TX	RX
1.8 V	3.06 mW	54 mW	27 mW
3.0 V	5.1 mW	90 mW	45 mW
3.6 V	6.12 mW	108 mW	54 mW

module we consider the current consumption in IDLE State, receiving (RX) and transmitting (TX) mode. In IDLE state the radio module needs 1.7 mA, in RX mode 15 mA, in TX mode 30 mA (SLEEP state with 0.2 μA is negligible) [29]. Table VI shows the power consumption for these modes, depending on the different supply voltage levels. The maximum values are 6.12 mW for IDLE state, 108 mW for TX and 54 mW for RX mode. Due to the fact, that the application is in IDLE state most of the time, the total amount of power consumption of our system is approx. $17.55mW + 6.12mW = 23.67mW$. Depending on the frequency of sending commands, the power consumption for TX mode must be added. In our example application we mainly use window shutters. In a household the persons typically run up the window shutters at morning and run them down at evening. Under this assumption the TX mode is entered twice at a day for some microseconds. With regard to the statutory duty cycle, which limits time the gateway can transmit at 868 MHz band (1% transmitting time in 1 hour), the power consumption results as follows. A transmitter can only transmit data at 868 MHz band for 0.01 h in 1 h, so the maximum power consumption for RX in this time is $108 \cdot 10^{-3}W \cdot 0.01h = 0.00108Wh$. Additionally with the $23.67 \cdot 10^{-3}Wh$ from above we get a power consumption of 0.02475 Wh for our gateway. These values could be reduced by using a lower supply voltage as described in the tables or by using a lower radio output power level. However, in comparison with the power consumption of the FRITZ!Box router the values of the gateway are negligible. The FRITZ!Box router has an average power consumption of 9.3 Wh and a maximum of 27 Wh [30]. As described in the router manual, the average power consumption was determined with an active DSL connection, active WLAN, active DECT with one connected phone and one connected LAN device without data transfer.

VII. Conclusion

In this paper, we presented a low power and low cost sensor network gateway, which could be integrated in existing network infrastructures at home. Due to the fact that the gateway is just a little USB-Stick and it communicates over the USB-MSC Profile with a router, we present a gateway, which needs no extra power supply. The power consumption of the gateway is just approx. 0.3 % of the power consumption of the router. Therefore, the consumption of the gateway is negligible. The application example shows the easy use of the system and further applications like monitoring of sensor data or controlling washing machines or alarm systems are possible. The gateway is also useable in the industry by energy service providers to read out power and water consumption over the internet or by utility companies to monitoring the power grid. The next steps are the optimization of the command structure, implementing a flexible application for different smart devices with different protocol stacks and analyzing performance parameters and the possible amount of concurrently supported sensors in the system.

Acknowledgment

This work was supported by the Regensburg Center of Energy and Resources (RCER) and the Technology- and Science Network Oberpfalz (TWO). Further information under www.rcer.de.

References

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of things (iot): A vision, architectural elements, and future directions,” Future Generation Computer Systems, vol. 29, no. 7, 2013, pp. 1645–1660, DOI: 10.1016/j.future.2013.01.010.

[2] H. M. Ammari et al., “A survey of sensor network applications and architectural components,” Ad Hoc & Sensor Wireless Networks, vol. 25, 2015, pp. 1–44, ISSN: 1551-9899.

[3] T. Gomes, J. Brito, H. Abreu, H. Gomes, and J. Cabral, “Greenmon: An efficient wireless sensor network monitoring solution for greenhouses,” in Industrial Technology (ICIT), 2015 IEEE International Conference on. IEEE, 2015, pp. 2192–2197, DOI: 10.1109/ICIT.2015.7125420.

[4] S. Hussain, S. Schaffner, and D. Moseychuck, “Applications of wireless sensor networks and rfid in a smart home environment,” in Communication Networks and Services Research Conference, 2009. CNSR’09. Seventh Annual. IEEE, 2009, pp. 153–157, ISBN: 978-1-4244-4155-6.

[5] X. Fang, S. Misra, G. Xue, and D. Yang, “Smart gridthe new and improved power grid: A survey,” Communications Surveys & Tutorials, IEEE, vol. 14, no. 4, 2012, pp. 944–980, DOI: 10.1109/SURV.2011.101911.00087.

[6] G. Zenger, S. Kenner, K. Volbert, T. Waas, and M. Kucera, “Acquiring energy data from a medium-voltage grid for future smart grid solutions: A practical smart grid application example realized by use of cellular communication networks of the 2 nd and 3 rd generation,” in Intelligent Solutions in Embedded Systems (WISES), 2013 Proceedings of the 11th Workshop on. IEEE, 2013, pp. 1–8, ISBN: 978-3-00-042899-9.

[7] S. Kenner, R. Thaler, M. Kucera, K. Volbert, and T. Waas, "Smart grid architecture for monitoring and analyzing, including modbus and rest performance comparison," in Intelligent Solutions in Embedded Systems (WISES), 2015 12th International Workshop on. IEEE, 2015, pp. 91–96, ISBN: 978-88-87548-06-8.

[8] "Europe 2020 - a strategy for smart, sustainable and inclusive growth," 2010, URL: http://eurlex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2010:2020:FIN:EN:PDF [accessed 2016-04-06].

[9] "Rwe smart home," URL: https://www.rwe-smarthome.de/web/cms/de/2768534/home/ [accessed 2016-04-06].

[10] "Telekom smart home," URL: https://www.smarthome.de/was-ist-smarthome [accessed 2016-04-06].

[11] "Homematic smart home," URL: http://www.homematic.com/ [accessed 2016-04-06].

[12] "Homee - Smart Home so easy as Lego," URL: http://www.siio.de/homee-smart-home-so-einfach-wie-lego/ [accessed: 2016-03-16].

[13] M. Mittek, J. D. Carlson, and L. C. Pérez, "Design and implementation of a low-cost embedded linux gateway for smart home health monitoring," in Electro/Information Technology (EIT), 2014 IEEE International Conference on. IEEE, 2014, pp. 485–490, DOI: 10.1109/EIT.2014.6871812.

[14] S. Guoqiang, C. Yanming, Z. Chao, and Z. Yanxu, "Design and implementation of a smart iot gateway," in Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing. IEEE, 2013, pp. 720–723, DOI: 10.1109/GreenCom-iThings-CPSCom.2013.130.

[15] C. Kruger, A. Abu-Mahfouz, and G. Hancke, "Rapid prototyping of a wireless sensor network gateway for the internet of things using off-the-shelf components," in Industrial Technology (ICIT), 2015 IEEE International Conference on. IEEE, 2015, pp. 1926–1931, DOI: 10.1109/ICIT.2015.7125378.

[16] B. D. S. Campos et al., "Design and construction of a wireless sensor and actuator network gateway based on 6lowpan," in EUROCON-International Conference on Computer as a Tool (EUROCON), 2011 IEEE. IEEE, 2011, pp. 1–4, ISBN: 978-1-4244-7486-8.

[17] S. K. Dhar, S. S. Bhunia, S. Roy, and N. Mukherjee, "Enabling smartphone as gateway to wireless sensor network," in Recent Advances in Information Technology, RAIT-2014 Proceedings. Springer, 2014, pp. 19–26, ISBN: 978-81-322-1855-5.

[18] A. Voinescu, D. Tudose, and D. Dragomir, "A lightweight, versatile gateway platform for wireless sensor networks," in Networking in Education and Research, 2013 RoEduNet International Conference 12th Edition. IEEE, 2013, pp. 1–4, ISSN: 2068-1038, ISBN: 978-1-4799-2599-5.

[19] "Bm wireless ltd.&co.kg - bluerobin^{TM} technology," URL: http://www.bmwireless.com/index.php/references_bmwireless.html [accessed 2016-04-06].

[20] "ez430-chronos^{TM} development tool - user guide," URL: http://www.ti.com/lit/ug/slau292g/slau292g.pdf [accessed: 2016-03-21].

[21] "The msp430f5509 mixed signal microcontroller - datasheet," URL: http://www.ti.com/product/MSP430F5509/description [accessed: 2016-03-21].

[22] J. Axelson, USB complete: the developer's guide. Lakeview research LLC, 2015, ISBN: 978-1-931448-28-4.

[23] ——, USB mass storage: designing and programming devices and embedded hosts. lakeview research llc, 2006, ISBN: 978-1-931448-05-5.

[24] "Usb developer package 5.0.01," URL: http://software-dl.ti.com/msp430/msp430_public_sw/mcu/msp430/MSP430_USB_Developers_Package/latest/index_FDS.html [accessed: 2016-03-22].

[25] "Wireless switch socket "fs20 st"," URL: http://www.produktinfo.conrad.com/datenblaetter/600000-624999/623004-an-01-en-FUNK_SCHALTSTECKDOSE_FS_20_ST.pdf [accessed 2016-04-06].

[26] "Radio controlable window shutter," URL: http://www.produktinfo.conrad.com/datenblaetter/1000000-1099999/001093002-an-01-de-FS20_RSU_2_UP_FUNK_ROLLLADENSCHALTER.pdf [accessed 2016-04-06].

[27] "Fs20 - radio control system," URL: http://www.elv.de/fs20-funkschaltsystem.html [accessed 2016-06-13].

[28] "zwave alliance," URL: http://z-wavealliance.org/ [accessed 2016-04-06].

[29] "Cc1101 - low-power sub-1ghz rf transceiver," URL: http://www.ti.com/lit/ds/swrs061i/swrs061i.pdf [accessed: 2016-06-13].

[30] "Fritz!box manual," URL: https://avm.de/fileadmin/user_upload/DE/Handbuecher/FRITZ_Box/Handbuch_FRITZ_Box_7490.pdf [accessed 2016-06-13].