A battery-less switch for use in 802.15.4 / ZigBee applications

Marcel Meli, Alex Bluem

Marcel Meli
Institute of Embedded Systems
ZHAW, 8041 Winterthur, Switzerland
marcel.meli@zhaw.ch

Abstract:

Wireless switches to control light systems are important parts of wireless networks in building automation. For success, it is important to avoid frequent battery replacement, or even better to eliminate them. Although this is possible in some proprietary systems today, at the expense of some features, it is still a challenge for networks such as ZigBee. The main difficulties lie in the power available and the requirements for communication.

In this paper, we present the design of a battery-less wireless switch that can be integrated in 802.15.4 / ZigBee systems. We use a standard energy harvesting module for generating power when the switch is activated. The important and required power management aspects implemented in hardware and software are discussed, in the light of the network specifications.

After showing the challenges needed to overcome the use of a battery-less switch in wireless systems such as 802.15.4, we present our solutions and the results of some tests.

1) Introduction and proposal

Switches are very simple devices that are found in nearly every electrical system. Their installation in wired systems often requires the running of cables between switches, power sources and electrical devices (lamps, motors, etc). Switches are installed in millions worldwide, and are one of most used items in building automation. Residential houses and buildings have tens or hundreds of them. The use of cable often means that switches are confined to certain positions, and moving them to new positions infers costs for extra cable, labour and making/mending holes or channels in the walls. For these reasons, wireless switches have long been considered to have the potential of bringing needed flexibility in the installation of switches.

Systems exist on the market, which allows consumers to control electrical devices, using remote wireless switches. Many such devices can be found in "Do It Yourself" shops. Various wireless protocols are used, and most of these devices need batteries.

- The incompatibilities between the various protocols make it difficult to have a comprehensive or global solution, forcing the users to solve only part of their home or building automation problems. Furthermore, they are often bound to one or just a handful of manufacturers.
- The use of batteries in wireless switches pushes the user to take into account the replacement of batteries. That means monitoring their energy, finding the switches, replacing the batteries, disposing of old batteries. It might even mean having the right type of battery in store. Although many wireless switches are today so designed that their batteries will last for years, it is still be a daunting task for a building that has

hundreds of switches. Many people will clearly prefer to avoid having to replace batteries.

Recognising this need, the German firm "En Ocean" has designed and installed thousands of wireless switches that use an Energy Harvester as power source. The protocol is so designed as to work with a small amount of power. A description of this system can be found on the web site of the firm. [1] Switches are transmitters only, and send their data several times at different intervals. The data packets are very small, and carry little overhead. The high bit rate further helps to keep the message short. The whole system works in the 868MHz band, which is still less crowded than the 2.4 GHz band. These characteristics allow messages to be sent with a high probability of reaching their destination, which in turn reduces the need of a CCA (Clear Channel Assessment) and therefore saves costs and power.
The system does have some drawbacks:

- No clear channel assessment: Although the messages are small and sent several times, the collision problem increases with the number of active nodes.
- The system is still proprietary, and this could be a problem to reach a critical mass.

A system like 802.15.4 / ZigBee has more potential, thanks to the large base of firms supporting it. This advantage has lead to a beneficial competition among semiconductor manufacturers, which is slowly, but surely leading to better products, including in the area of power consumption. It is likely that in the coming years, even better components will appear on the market, facilitating the use of weak energy sources. Such factors have the potential to minimize the drawbacks related to the large overhead of 802.15.4 / ZigBee.

In this work, we have used a standard energy harvester of the firm En Ocean to power a microcontroller and a 2.4 GHz transceiver. As soon as power is applied, the microcontroller wakes up, initialises the transceiver, performs a CCA and sends data to a lamp endpoint. Most of the data sent is pre-programmed (channel, network identity, are defined in advance). These operations mean running the switch on about 100 micro joules.
In the present state, the switch cannot join a network using the power from the electro-dynamic Energy Harvester.
A good understanding of the system is required to determine energy critical steps and optimise the use of power. Furthermore, an appropriate choice of components is needed for the system.

In what follows, we will first examine the requirements needed to send the proper message from a switch to a lamp endpoint. Secondly, we will apply these to energy needs and the different steps required to send the message. Thirdly we will present a design that satisfies these requirements, and the results of some tests. In the last part, we will briefly discuss integration issues.

2) <u>Description</u>

a) Requirements for 802.15.4 and ZigBee switch and assumptions

A switch is one of the simplest devices (functionally) in an 802.15.4 based network.

- The message it sends can be coded in a simple bit (ON/OFF or Toggle).
- It does not need to get an answer back.
- It can very well be configured as a Reduced Function Device (RFD) in an 802.15.4 based network.

To be ZigBee compatible, a device must implement other functions, such as been capable to join or leave a network. In the case of a switch, these functions would be necessary mainly at

special times. Mostly, the device will be sending information about the state of the switch. We are not concerned with ZigBee compatibility in this work, but rather with sending the switch message in a way that can be understood by the endpoint. We assume that the switch is a RFD in a network, with defined frequency, addresses, etc. The issue of joining a network is briefly dealt with at the end of this paper. It will be discussed thoroughly in a different work. In what follows, we therefore focus on sending the state of the switch to another endpoint.
For sending switch message the following will be mandatory or optional:

- CCA must be implemented. The switch must check that the channel is free before sending data. This requires the presence of a receiver, with is linked to an important power consumption (above 15 mA in many cases)
- Sending a message. The switch should have enough energy to send the appropriate number of bytes. The power consumption of the transmitter will then depend on the range required.
- ACK (MAC layer). When the switch sends data, it can require the receiving party to send an acknowledgment. This confirmation can then be used by the switch to determine if the message has been received, and if it should be resent. We do not rate this as necessary in an application where the wireless switch is used to turn lights ON or OFF. In such applications, the user will most probably press it again if the lamps are not switched ON or OFF.
- APS layer Acknowledgment. This will be sent by the receiving party to the switch if it was so required. If the switch is programmed not to request APS layer acknowledgment, it also does not need to wait for it. The argument above about the necessity of a MAC ACK signal also applies here.
- Retransmission of data. For the reasons mentioned above, this is not necessary in the case of a switch that controls lamps.

It follows then that at the minimum, the switch will need to perform a CCA and send a certain number of bytes. Of the 4 defined frame types (beacon, data, acknowledgment, MAC command), only the data frame is of interest here.

A typical data packet on the 802.15.4 application will look as following [2].

octets	2	1	0/2	0/2/8	0/2	0/2/8	0/5/6/10/14	Variable	2
Fields MAC	Frame control	Sequence number	Dest. PAN	Dest. Addr.	Source PAN	Source Addr.	Security Header	Payload	FCS
Example	2	1	2	2	0	2	0	Variable	2

It will then required about 11 bytes + payload

A typical data packet on the ZigBee application layer will build on the example above, and look as follows [3].

Octets	2	2	2	1	1	0/8	0/8	0/1	var	var
Fields NWK	Frame control	Dest. Addr.	Source Addr.	Radius	Seq. Nber	Dest. Addr.	Source Addr.	Multicast cntr.	Src. Route subfr.	Frame payload

Octets	1	0/1	0/2	0/2	0/2	0/1	1	Variable	
Fields APS	Frame control	Dest. endpoint	Group Addr.	Cluster Id.	Profile Id.	Source endpoint.	APS counter	Payload	

We can deduce that, to be able to send switch commands for ZigBee, the following is needed:

- Clear channel assessment (switch on the receiver to verify channel activity)
- Transmit 30 + 2 + 6 = 38 bytes. (30 bytes of data + 2 bytes (FCS) on MAC layer, 6 Bytes for PHY overhead)

Near these requirements, the APS layer counter will also need to be incremented. This means that the switch should include a form of non volatile memory. Preferably, it should be a memory that can be read and written quickly, thousands of times, and with little energy. EEPROMs and FRAM are possibilities. In the case of EEPROM, the number of write cycles should be verified. For FRAMs, the value is very high. The write time of EEPROMs can be also an issue (several milliseconds). At the beginning, the APS counter value in the non volatile memory will be read, then incremented and written back.

b) Low power with microcontrollers and transceivers

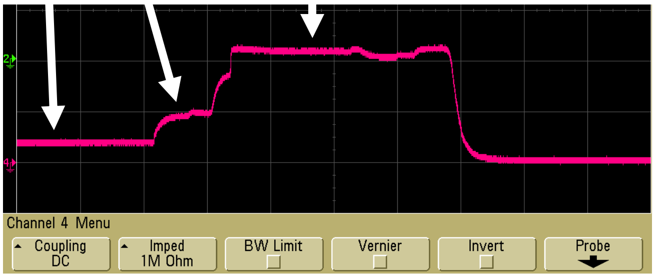
When dealing with low power design, it is important to understand how different pieces of hardware and software contribute to the power consumption. Each element that is not needed should be shut off, and one should avoid switching many blocks at the same time. It is therefore advantageous to use a microcontroller that allows such a control.

Near the current consumption at different active modes and the minimal operating voltage, there are other elements such as the delay counter after Reset that should be considered.
The microcontroller does not need a very precise clock, but its transceiver does. The microcontroller should therefore be allowed to run as soon as possible. A microcontroller that allows this will help save energy.

Fig. 1 shows the current consumption of the wireless switch. The contribution of transmitter and receiver can clearly be seen.

Fig. 1 *Dynamic current measurement showing that the current depends on the blocks and activities:*

Reset initialisation transceiver active



- Delay counter: As power is applied the processor starts up, and a delay counter is used to wait until the oscillator is stable.
- Initialisation: Different elements of the system are initialised. Only the CPU is active.
- The oscillator of the transceiver is started, and the CPU waits until it is stable
- The transceiver is initialised, and data to send is loaded.
- The receiver is started, for a CCA.
- The transceiver is activated, and data is sent.

In receive and transmit phases, much power is needed. It is thus important to keep the devices in these modes as shortly as possible.
Working with the lowest possible voltage saves energy. Therefore, components which can work at low voltages should have the priority.

c) Design of the switch

i) Hardware

Fig.2 shows a block diagram of the Switch Under Test (SUT). A ZigBit module of Meshnetics was chosen, based on the interesting low power data of this module. The ZigBit module is made up of the microcontroller from Atmel (ATmega1281) and the transceiver from the same firm of the firm Atmel (AT86RF230). These components are actually among the best in term of power consumption for 802.15.4 / ZigBee.
To supply the ZigBit module, the electro-dynamic Energy Harvester ECO100 of the firm En Ocean was used. The generated power was rectified using an appropriate diode (low voltage drop). A 57uF capacitor was used to accumulate the energy.
The microcontroller and the transceiver can work down to 1.8 Volt

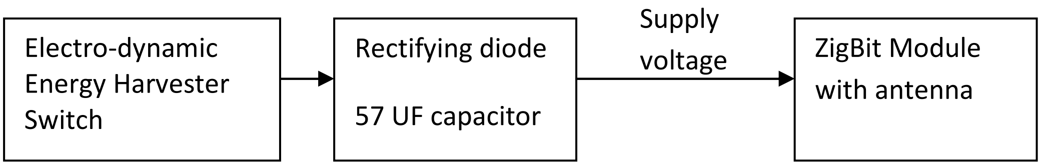

Electro-dynamic Energy Harvester Switch
Rectifying diode
57 UF capacitor
Supply voltage
ZigBit Module with antenna

Fig. 2 *Block diagram of the wireless switch*

ii) Software

The software in the ZigBit module was written in C and contains the following important elements:

- Fast start up to reduce power lost if there is a long countdown after Power On Reset.
- Initialisation of the transceiver
- Switch Receiver on
- CCA
- Send data
- Go into power down

3) Results

a) Available energy.

To measure the energy available, we used a load of 120 Ohm. We did not use the best components, therefore there is room for improvement. The switch delivers about 100 Micro Joules.

b) Testing the switch

i) Sending a switch frame

Set up.
In order to test the switch, we used the Beekit of Freescale (NVB and SRB Evaluation boards with ZigBee Beestack). Two elements of the kit were set up to run a switch/Lamp application. The first device (FRL) acted as the Coordinator and included the Lamp endpoint (LED). The second device acted as a switch endpoint (FRS). The data from FRS was captured, and programmed into our switch (SUT).

The program to send the switch data is run according to the following scenario, after the POR that follows the power generation resulting from activating the switch.

- Initialise transceiver
- Switch on receiver
- CCA
- Switch on transmitter
- Send data

SUT was then activated after FRS was activated a certain number of times.
The tests showed clearly that SUT could switch the Lamp ON or OFF.

Fig.3 shows the supply voltage of SUT when the switch is toggled and data is sent.
The voltage reaches a peak of 2.4 volts, that microcontrollers starts up, initialises the transceiver and data is sent before the voltage is too low.

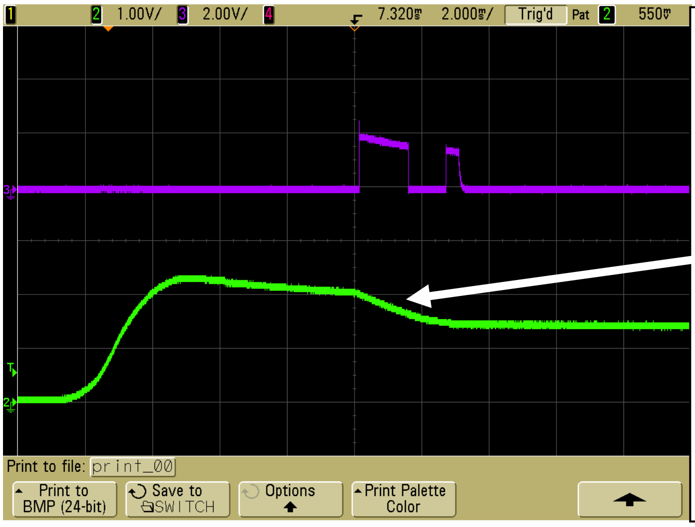
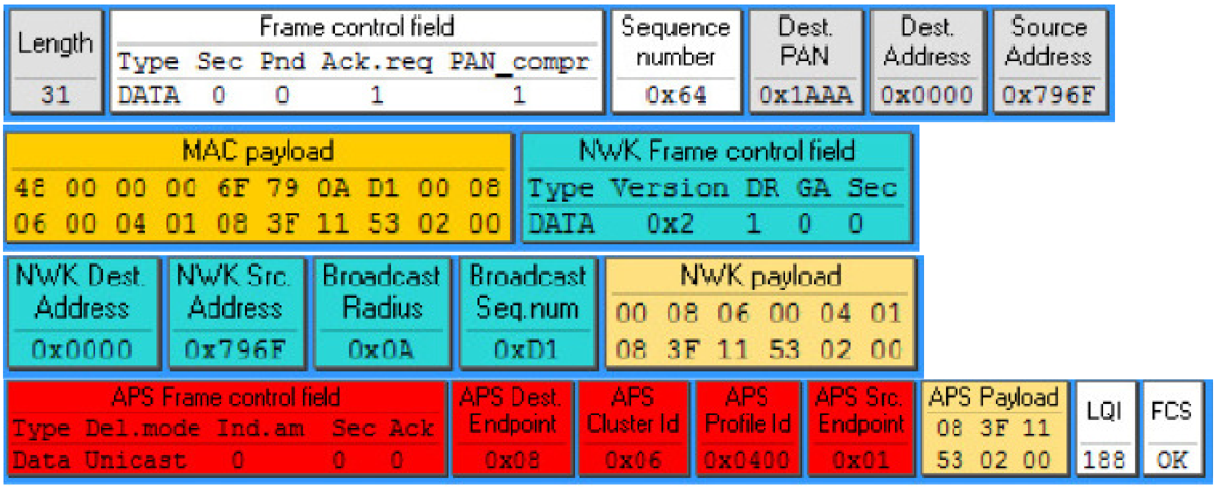


***Fig.3** GPIO pin monitoring activities (upper trace) and supply voltage (lower trace).*

Transceiver sends data

Fig.4 shows the data sent by the SUT, as captured by a sniffer

Fig. 4 Data from sniffer, MAC layer. CRC bytes not shown
Packet index: 33 Length: 31
Raw data (hex): 61 88 64 AA 1A 00 00 6F 79 48 00 00 00 6F 79 0A D1 00 08 06 00 04 01 08 3F 11 53 02 00
RSSI [dBm]: -39 Correlation value: 108 CRC OK: 1



Length	Frame control field Type Sec Pnd Ack.req PAN_compr	Sequence number	Dest. PAN	Dest. Address	Source Address
31	DATA 0 0 1 1	0x64	0x1AAA	0x0000	0x796F

MAC payload	NWK Frame control field Type Version DR GA Sec
48 00 00 00 6F 79 0A D1 00 08 06 00 04 01 08 3F 11 53 02 00	DATA 0x2 1 0 0

NWK Dest. Address	NWK Src. Address	Broadcast Radius	Broadcast Seq.num	NWK payload
0x0000	0x796F	0x0A	0xD1	00 08 06 00 04 01 08 3F 11 53 02 00

APS Frame control field Type Del.mode Ind.am Sec Ack	APS Dest. Endpoint	APS Cluster Id	APS Profile Id	APS Src. Endpoint	APS Payload	LQI	FCS
Data Unicast 0 0 0	0x08	0x06	0x0400	0x01	08 3F 11 53 02 00	188	OK

ii) Using all the energy to send (maximum packet length with the available energy)

In order to see how many bytes could be sent with the switch, we programmed the system to send as many bytes as possible after a CCA, and monitored the result on a sniffer. Knowing what we should get, we then counted the number of bytes that were properly sent. The switch could repeatedly send more than 40 bytes before stopping due to a lack of energy (voltage under the minimum operating voltage)

Fig. 5 shows the supply voltage of SUT
Fig.6 and Fig. 7 show the data captured by the sniffer; what should be and what is.

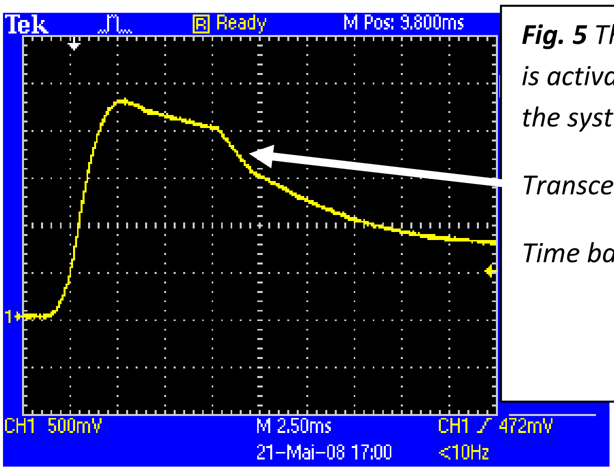


Fig. 5 *The voltage rapidly decreases as the transmitter is activated, and data is sent. Under a critical value, the system ceases to function properly.*

Transceiver active

Time base is 2.5 ms/div, voltage is 500mV/div

Fig. 6 Expected data CRC bytes not shown
Raw data (hex): 61 88 64 AA 1A 00 00 6F 79 48 00 00 00 6F 79 0A D1 00 08 06 00 04 01 08 3F 11 53 02 5A 69 67 42 65 65 20 43 6F 6E 66 65 6E 63 65 ***20 73 65 63 6F 6E 64 20 74 69 6D 65 20 3B 3C 3D 3E***

Fig. 7 Received data CRC bytes not shown
Raw data (hex): 61 88 64 AA 1A 00 00 6F 79 48 00 00 00 6F 79 0A D1 00 08 06 00 04 01 08 3F 11 53 02 5A 69 67 42 65 65 20 43 6F 6E 66 65 6E 63 65 ***70 77 7B 77 77 74 59 67 70 77 17 77 77 75 67 77 77***

4) Open questions.

We have shown that using hardware components that are available and the right software strategy, it is possible to send switch information to an end point in a 802.15.4 / ZigBee network. Since the energy harvester does not generate enough power to implement the association / disassociation functions, a strategy has to be found to incorporate the switch in the network. Some of the possibilities are:

- Implementation of the needed functions. Associate at installation using batteries, program the needed values in the non volatile memory of the switch memory and then let the system run without batteries (more memory required)
- Use an installation device (with enough energy) when the switches are installed. This device will read the identity of the switch, perform the association, get the needed wireless network information, and then program them in the switch memory. It will then be disconnected from the switch that will be left to run without batteries (sending ON/OFF information). This approach has the advantage of keeping the price of the switch very low (not much memory required, CPU resources low, even 4-bit micro can be used). This can be combined with a pairing strategy, and the needed information transferred using an out of band channel
- The switch itself can be used to provide the needed energy at installation. This will require the switch to be pressed several times at short intervals, and the microcontroller to decide on the functions to run based on its supply voltage. This might not be very comfortable, but as said earlier, this action should be required mainly at installation. (no extra batteries at installation, no extra devices for installation, a channel change can be followed). The device will need to have the required stack functions for association and disassociation.
- The switch can send its data on several channels to make up for possible change of channels.
- The switch can be made to monitor the acknowledgment, and change to the next predefined channel after a number of failures (no acknowledgement received).

5) Directions for future work

Based on what was said above, our work in this area will continue in the following lines:

- Use of a new low power RISC microcontroller capable of working down to 0.9 Volt, and consuming less power than the use used in this work.
- Explore the different proposals made above for integrating the switch in a network
- Use of pairing techniques to ease the deployment of such switches

6) Conclusion

In this work, we have discussed the important requirements for using a battery-free switch in 802.15.4 / ZigBee applications. We have presented a design that can be used to send compatible data to a lamp end point and shown that with the use of the current generation of microcontrollers and transceivers, it is possible to use battery-free switches with ZigBee devices. There are new microcontrollers and transceivers in development that will allow the system to consume even less energy, and thus open new possibilities.

For any commercial application, the issue of patents must be considered.

7) Thanks

We would like to express our thanks to the following groups who have contributed to the work.

- Representative of Meshnetics in Germany for providing the modules.
- Anatec Switzerland for providing various device samples.
- Freescale (Switzerland) for providing the Beekit.
- The wireless Team of InES for various help with the development work.

8) References

[1]Whitepaper
http://www.enocean.com/fileadmin/redaktion/pdf/white_paper/wp_wireless_sensor_solutions.pdf
[2] 802.15.4 specifications (2006)
[3] ZigBee specifications (2006)