A Touch and Pair system for battery-free 802.15.4/ZigBee Home Automation networks

Authors:
D.Condrau , L.Zimmermann ,
M.Gysel (InES), M. Würms (InES), M.Meli (InES),

Contact address:
Prof. Dr. Marcel Meli
Zürcher Hochschule für Angewandte Wissenschaften (ZHAW)
Institute of Embedded Systems (InES)
Technikumstrasse 9
CH-8400, Winterthur, Switzerland
Email : marcel.meli@zhaw.ch

Abstract:

In this paper, 2 problems affecting the acceptance of wireless devices by a wider public are introduced, and possible solutions are suggested. The first obstacle is linked to the necessity of replacing the batteries of autonomous wireless nodes. Although this problem can be solved to a certain extent by using battery-less devices, the energy need of flexible protocols such as ZigBee increases the complexity a device autonomously joining a network (including association and binding), leading to the second obstacle. A solution based on RFID components that allows the transfer of pairing information using a "Touch and Pair" system is presented. It is shown that a consumer device such as an iPod/iPhone can be modified to serve as a user friendly pairing device. Using ultra low power components, battery-less switches sending ZigBee compatible frames are built. Key network parameters can be transferred from the iPod/iPhone to the switch or other end-points and thus allow a fast and simple configuration of battery-less elements on the network.

1. Introduction and description of the issues.

The benefits of using wireless devices in building automation have been known for some time. Yet the market has been slow to accept them. There are certainly many reasons to that. Near the proliferation of non-compatible wireless systems, there are other important obstacles that slow down the widespread acceptance by the general public of wireless systems in building Automation. Two of these obstacles are considered here, especially with respect to the widely publicised 802.15.4/ZigBee standard [1,2]. These obstacles are:

a) The need to replace used batteries which can quickly become a nightmare when many devices are involved.
b) The complexity involved in the configuration (say pairing) of end points. At the best, this complexity restricts the freedom that should be provided by wireless systems. At the worst it excludes those persons not at ease with the use of complex PC-based configuration tools.

European Zigbee Developer's Conference, 2009. WEKA Fachmedien

1.1. Battery issue

Several efforts have been made to address the first problem. The power consumption of devices has been reduced, and protocols have been designed to work with a low power budget. As a result, 2 groups of devices dealing with low power can be found on the market.
There are on one hand, those who are trying to eliminate batteries and deliver battery-free devices. Those devices rely on energy from the environment [8,9,13,14]. The energy may come from intermittent sources such as electro-dynamic harvesters or Piezo elements, when the user activates the switch. It may also come from continuous sources such as Peltier elements or solar cells. In both cases, the power generated in a reasonable priced and sized device is in the order of tens to hundreds of micro Joules. The energy available is very small, leading those on this path to design protocols with little overhead. Such protocols require very little energy and also show little flexibility. The price to pay here is often that of proprietary systems, geared to special markets. [6,7]
On the other hand, there are those who have chosen a more open and flexible standard such as 802.15.4/ZigBee. The flexibility of such a protocol comes at the expense of a certain complexity (and overhead) which negatively affects the energy consumption. Those on that road have so far tried to solve the problems linked with the need to replace batteries by reducing the power consumption and improving the quality of batteries. This has led to extending the battery life of systems. Many wireless building automation devices can now boast years of battery life. However, the need to replace batteries sometimes in the future is still there. So is the daunting task of finding out used batteries and replacing them. A task all the more difficult when the number of wireless nodes in the network is high.
A better solution is undoubtedly a combination of battery-less devices and of a flexible protocol. Although these 2 requirements play against each other, it has already been successfully demonstrated that battery-less sensors of very low complexity (e.g. switches) can send 802.15.4 / ZigBee compatible frames [3]. Using about 100 micro Joules to send switch frames of about 30 bytes to another ZigBee device is no longer the issue. The problems that remain are mainly linked to other aspects of the ZigBee standard such as joining a network and binding a battery-less device to another end-device.

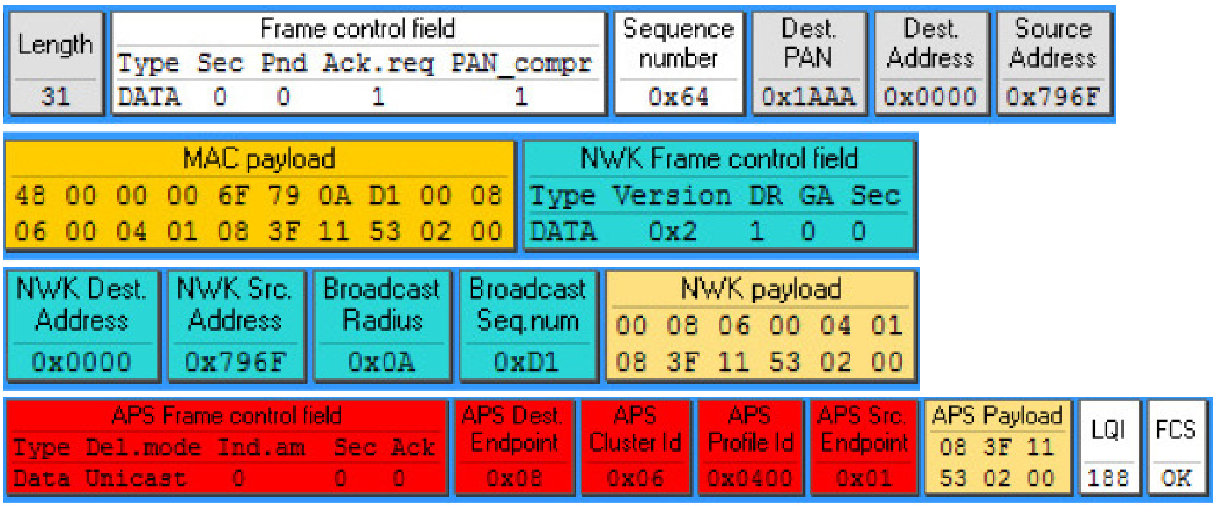
1.2. Pairing issue

There are normally several devices in a wireless network. These devices need to "agree" on parameters such as frequency band, network address ...etc. Device addresses should also be known to parties that are sending data to each other. In the case of ZigBee, a binding is needed between end-devices in order to ensure proper functioning.
Additionally, it might also be important to share some security information. All the relevant information should be distributed. In this work, we broadly describe the distribution of the information that is necessary for a successful communication (frequency band, PAN ID, device IDs, ...).
The amount of information to distribute is important. It increases dramatically with the number of nodes and cannot be remembered by most users. Furthermore, the lack of wires means that one cannot directly "see" the devices that are connected. The consequence is that special tools (often computers) may be needed for pairing. This places extra requirements on the users who will then tend to leave the installation to specialized personnel, resulting in an increase of costs (get the electrician to do it) and a loss of flexibility (you cannot do it yourself). Possibly, users may also lose interest in such a

demanding system. It is therefore important to find an easy and user friendly way to allow a user to pair wireless end points.
The issue of pairing is in principle found in many wireless systems. As soon as the size of information needed to configure the system grows, the binding button becomes very limited. [5]
In the case of battery-less devices for 802.15.4 / ZigBee, this issue takes another dimension. This is because it also addresses indirectly the issues of energy needs for a successful pairing.
In order to address these problems, we designed and tested a network where battery-less end-points are used, and pairing is done using a simple Touch and Pair system. End-points such as switches use different energy harvesting sources to generate 802.15.4/ZigBee compatible packets. Configuration parameters can be transmitted in a very secure way by bringing the devices together or by using a well accepted consumer product (iPOD or iPhone in our case). This makes the whole process very simple, fast and secure. No binding switches and no PC are required. Important parameters such as communication frequency, addresses of devices to pair, security information ...etc. are transmitted using a cheap and yet effective out of band channel (RFID in this case). The adopted strategy also presents the advantage of solving the problems associated with getting a battery-free device to join a given 802.15.4 network.

1.3. Energy needs

In what follows, we consider only the transmission of switch data. Switches are simple devices, most likely to be used without batteries. They are also likely to be among the most sold parts in a wireless building automation network. Lamps are normally connected to mains and thus should have no energy supply problems.
The energy needed in a battery-less wireless switch is a sum of the energy for the microcontroller and that needed by the transceiver. Typically, the microcontroller needs to be powered up, go through its start-up sequence, read the state of the sensor, initialise the transmitter with the proper parameters (pairing parameters), and give the order to send the data. Figure 1 shows a typical frame [3]. About 30 bytes are sent.



European Zigbee Developer's Conference, 2009. WEKA Fachmedien

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

Figure 1: Example of frame with switch data

A ZigBee device requires energy for the following tasks:

a) Scan task to find a network
b) Requesting association and joining a network
c) Binding
d) Passing messages (on, off, toggle, dimming)
e) Eventually reception of acknowledge
f) Other network activities such as rejoin, disassociation, frequency band changes

The most frequent activity is that of sending a message
We consider a device working at 2 Volts and requiring 15 mA in TX mode (for 0dBm). The energy need for transmitting at 250 Kb/S (4uS per bit) is:
15mA*2*4uS = 120nJ/bit (120 nano Joules per transmitted bit)
For transmitting 30 bytes of data, one will need about 300*120 nJ = 36uJ
This is about 1/4 of the energy available using a commercial energy harvester such as the ECO100. Energy is also used for the microcontroller activities, the start up phase of the transceiver and for updating the APS counter in non-volatile memory.

If done using the wireless link, operations such as joining the network, binding could last several milliseconds, and involve the transmission and reception of hundreds of bytes. Therefore, such tasks require much more energy than could be delivered by activating the energy harvester. These operations should occur just a few times in the life of a product. The data is then kept in non-volatile memory, and recalled every time a new message needs to be sent.

1.4. Problem of pairing (joining a network, binding)

Before a data exchange can take place between wireless parties, information such as frequency band, source address, destination address, network address must be known. Some of this information is part of the data frame that is sent to the receiving party. A commissioning tool can be used for the pairing procedure. It can also be done in a semi automatic way (for instance using a pairing button). A third possibility is to program the required data in the microcontroller at manufacturing or at installation time.
The first 2 ways require some energy, which is not available for battery-less switches that use only an intermittent harvester. In the case of continuous EH sources, a storage element might be used to accumulate enough energy before starting the procedure.

The third way reduces the flexibility of the system, and might even make it impossible for the user to reconfigure the system as wished, and thus nullify one important advantage of wireless systems.

2. The proposed solution

The solution we suggest is based on the use of a non-volatile table to store the pairing information at commissioning time. An RFID reader with appropriate software is used to configure the switch and lamp modules. The user first selects the function to pair using a graphic interface. By bringing the tool near the end-device, the proper information is written in the non-volatile memory. Each time the switch is activated (powered by the energy harvester), data is read from the RFID tag and used to build the communication frame that is then sent to control a load on the receiver side. The user can pair the switch with a different lamp, just by using the same easy procedure. Select the functions in the menu, and then touch the parts to pair.

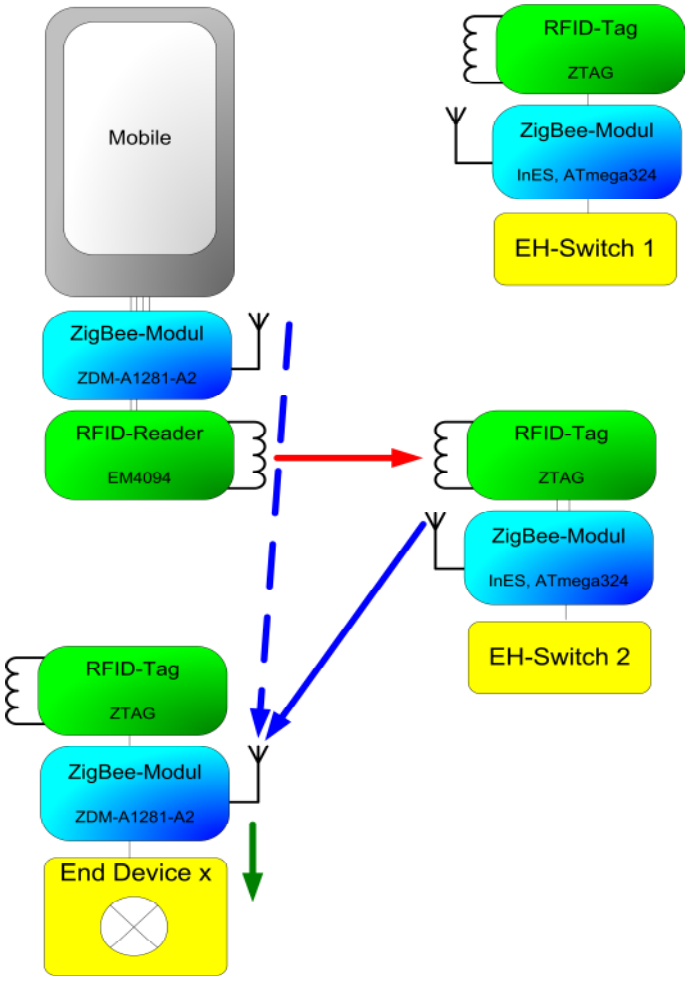


Figure 2 : Elements of the whole system

2.1. The hardware

Figure 2 shows all communication parts of a network set-up that proves the concept.

The switch end-point is a wireless battery-less switch, extended with an RFID Tag.

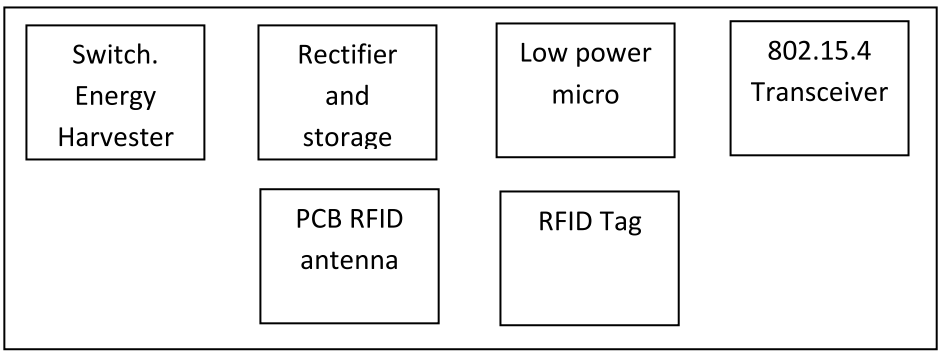


Figure 3: Block diagram of switch

The lamp end-point is a ZigBee wireless lamp, extended with an RFID Tag

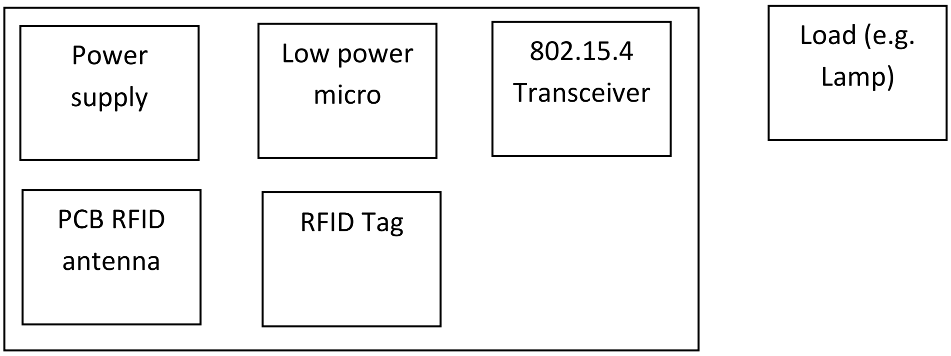


Figure 4: Block diagram of lamp end point

The pairing tool is the combination of an iPod/iPhone and a RFID reader extension board.

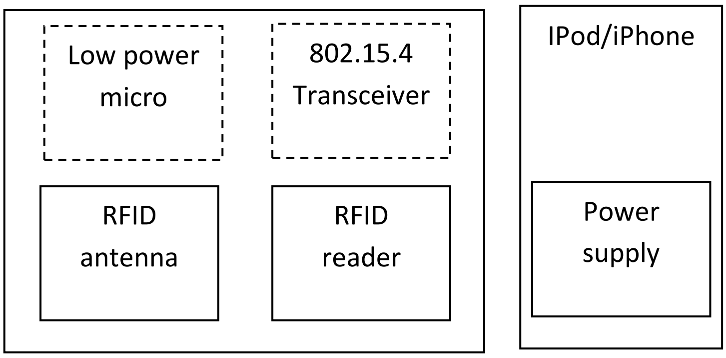


Figure 5: Block diagram of pairing device

The iPod communicates with the RFID reader using a serial link.
The RFID reader has an own microprocessor to ease programming and allow the timing constraints to easily be met while communicating with the tag.
The iPod can also be fitted with a ZigBee device. That will allow the iPod to join the wireless network, and afterwards transfer the parameters to the Switch.
In this work, the ZigBee device attached to the iPod was not used. Instead, pairing parameters were generated from the iPod according to the options chosen by the user.
The central element of the strategy is the RFID tag that is used to store the pairing information. That tag should have the following features:

- Enough room for the pairing parameters
- Dual ported.
    - One port is wireless as in the case of all RFID devices.
    - A second port should allow the end point microprocessor to read/write the information (wired link)
- Low power requirements, especially when read from the microprocessor.
- On chip power management
- Sufficient amount of read/write cycles to allow several pairing cycles during the product life of the end-device (thousands of programming cycles)

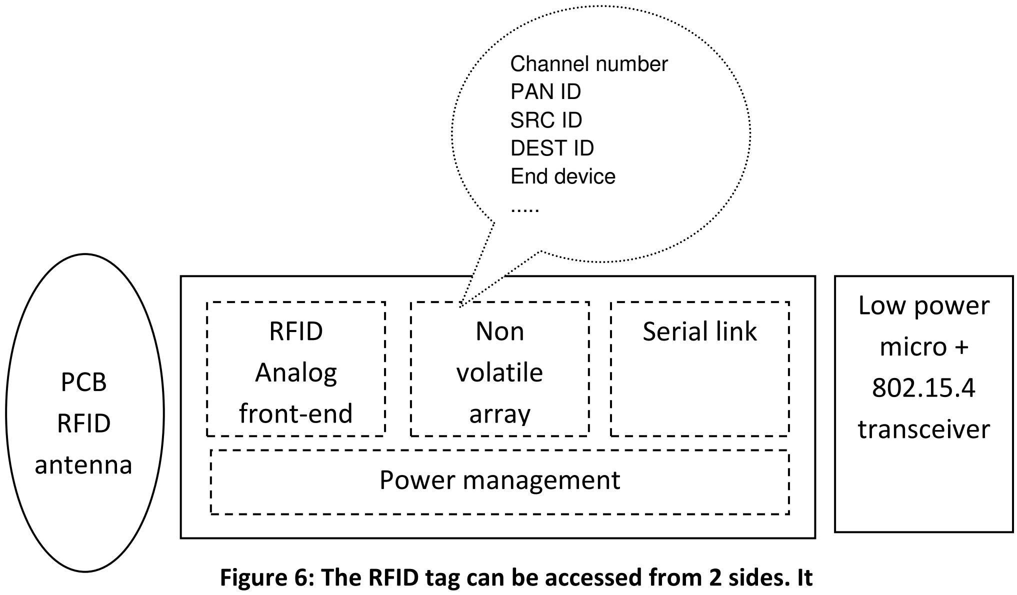


Figure 6: The RFID tag can be accessed from 2 sides. It holds the pairing parameters

Devices of the EM4233XL family of RFID chips manufactured by EM Marin are fully ISO15693 compliant, and fit these requirements. For this work, a device of the same firm, but of an older generation was used. New designs should be based on the new family.
For the RFID reader chip, there are several possibilities. For the work done here, we used the EM4094, a standard 13.56 MHz RFID reader from EM Marin.
For the battery-less tags, we used a combination of the 802.15.4 transceivers of Atmel and a low power microcontroller (EM6818 8-bit CoolRISC microcontroller from EM Marin, the low power AVR microcontrollers from Atmel) [10,11].

2.2. The Software

Lamp end-devices were implemented using a Full Function Device, with a ZigBee stack. Additionally, software was written to read information from the RFID tag, and use it in the network.
Switch end-devices were kept very simple. They simply send ZigBee compatible frames (see Figure 1), after reading the needed parameters from the RFID tag. In this way, a microcontroller with very little resources can be used.
The pairing device had software to allow the user to make the selection of elements to pair. It also had software running on the RFID reader, allowing data to be read/written from/to RFID tags.

a) Switch side
   1. Read information from tag when power is applied
   2. Use the information to initialise transceiver
   3. Send the message
   4. Optionally get the acknowledge and update failure/success fields

b) Lamp side
   1. Initialise the end-point with the information in tag
   2. Wait for communication
   3. Act on received switch commands

c) Pairing device side
   1. Menu for the user to select the actions
   2. Read contains of tags
   3. Compute the new contents of the RFID table according to the chosen pairing option
   4. write pairing information in tag

3. Tests and results

We built 2 different stations, each supporting 2 end-points, and fitted with the necessary electrical interface to control loads (240 volts). The 4 end-points were used to control lamps, radio, and ventilator.
4 Battery-less wireless switches were built, and fitted with RFID tags. Each switch was powered by an ECO100 energy harvester.
1 iPhone was fitted with the RFID reader and especially written User Interface software. The RFID reader was powered from the iPhone, but had an own microcontroller to read/write suitable RFID tags.
We could pair lamps and switches as needed, by bringing the reader within 2 cm of the tags.
The following combinations were done and successfully tested:
One switch to one load. Each load had an own switch.
One switch to many loads. Several loads could be controlled with one switch.
Many switches to one load. One load could be controlled with many switches.
The most critical element was the energy on the wireless switch side. That energy is needed to read the RFID tag first, to build the ZigBee frame, and then transmit it. The number of bytes read from the RFID tags had to be limited to less than 10. This is still enough for a network when short addresses are used. Some improvements are needed to allow the introduction of other pairing parameters.

Conclusions and future work

We showed that commissioning of 802.15.4/ZigBee devices is possible using dual-ported RFID tags. By adding the tags to end-points, pairing information can be programmed using a simple Touch and Pair system. In this system, the complexity of the procedure is hidden to the user by using a GUI linked to an RFID reader. The information in the tag can be read by the communication device, and used to send data with the appropriate parameters (frequency band, network and device identity, security, end-point address, ...).

This concept works for devices with enough energy, but also for battery-less devices. There is an extra benefit in the case of battery-free devices, since operations such as scan, association, or binding which need more energy are easy to handle.

Example of Far-Pairing Device with iPOD
From Padavath & Sauter, ZHAW 2009

Pairing with LF or HF RFID as presented here makes sense when devices are near enough to allow touch. There are situations when this nearness is not possible (e.g. Lamps on the ceiling). In such cases, Far-Pairing is needed. A practical design with this aspect in mind is in work, and the conclusions will be presented at a later date.

Acknowledgements

We wished to express our thanks to EM Microelectronic Marin for giving us information and samples of the RFID pairing and reader devices.
Our thanks also go to Anatec Switzerland, for delivering the 802.15.4 transceivers.

References

[1] 802.15.4 specifications
[2] ZigBee specifications
[3] A battery-less switch for use in 802.15.4 / ZigBee applications (Paper and presentation)
M. Meli, A. Bluem, M. Gysel. Second European Developer's ZigBee Conference. Munich June 2009
[4] Bidirectional 868/915 MHz wireless module powered with energy harverster.
M. Meli, S. Blättler, V. Kunz, M. Gysel
10. Wireless Technologies Kongress, September 2008, Bochum
[5] Low cost solutions to pairing issues in IEEE 802.15.4 networks
Marcel Meli, M. Gysel, M. Würms.
First European Developer's ZigBee Conference, Munich June 2007
[6] EnOcean: www.enocean.com
[7] lightningswitch: www.lightningswitch.com
[8] Private conversations a manufacturer of electrical switches
[9] Switch with EM9201. www.ines.zhaw.ch
[10] RFID devices. www.emmarin.ch
[11] Atmel 802.15.4 / ZigBee devices. www.atmel.com
[12] ZigBee Alliance. www.zigbee.org
[13] Communication gateway between wireless communication networks
Patent application of Schneider Electric
[14] Greenpeak. www.greenpeak.com