Alessandro Taccola - Théo Viguier - Brandon Aguessy

# INTERNET OF THINGS

TEAM PEAKY BLINDERS

15 JANVIER 2020

# TABLE DES MATIERES

# Description of the project

Nowadays, connected objects are more and more present in our lives. A connected object is an object equipped with sensors with a communication network to transmit and receive data. For our IoT project, we have decided to build a smart home. We will be able to handle the opening and closing of the connected shutters, to handle the lights and also to check the temperature remotely. You will be able to open and close the shutter with an application or a remote control, or even may be vocal control using google assistant. A simple "Ok google" will allow you to take control of the shutter. Another feature will the possibility to program the opening or closing of the shutter or the lights when you want or to wake up naturally with the daylight.

We are building a smart home. We mainly want to control lights, shutters and temperature. The system will be autonomous, but also will be controllable with a remote control, and the vocal command on a phone (Google Assistant). It is also possible to share with a smartphone if the window is open. You could go outside and check on your phone your window degree of openness.

# Related Work

Today, when we are talking about connected shutters, there are a lot of solutions which are propose on the web.

The first solution we have found is the Connected roller shutter pack, you can centralise the opening and closing of your shutters with a single click. It's possible to control roller shutters individually or centrally, program the closing and opening of the roller shutters, view if your shutter is open or close with an application if you are not at home.

The second solution we have found is the Shutter control. The functions are : Open and close the shutter, step up/step down, slat orientation, timer, schedule (includes sunrise and sunset synchronization).

The third solution we have found is the Bosch Smart Home. It allows the user to manage light and electronic shutters.

## Our Approach

Our final purpose is to create a system which allow the user to handle his home devices. Our purpose is a kind of smart home. The user will have several means to control light and shutters.

All the device (shutters, light) are connected to a relay. So, the main problematic is how could we manage the relay and with which kind of tools?

Our technical approach is defined with three main technology:

Zigbee:

Zigbee will allows the ESP 32 and all the relay to communicate together. In this situation Zigbee module looks like WIFI hotspot. To build our system, two ZigBee module are useful. One is the coordinator and the other one is the end device. The first one is linked to the ESP32 and IR remote, either all the 'master' corposant. The second one is connected with all of the relay, these will allow to turn on the light or open the shutters.

ESP32 with mobile application:

We will link to the ZigBee master to the ESP 32. This corposant is the more useful to connect external controller (mobile application, website…) due to it ability to be link with WIFI and Bluetooth. Our mobile application will be built with react native language. It will have only one feature. Open and close device, the user could select which device he would like to turn on or turn of. With the application it will be easier to select the device that we would manage. In the application each device will have name contrary to remote where each one will be a number.

IR remote with ESP32:

IR remote is our first proposition to build device manager. It is easier to connect than mobile application. Controller could turn on or turn of any device save in the system.

## Software design

There will be some kind of centralization where every module will communicate with a main card. It will be either through WIFI, or ZigBee. The software will be divided in multiples parts.

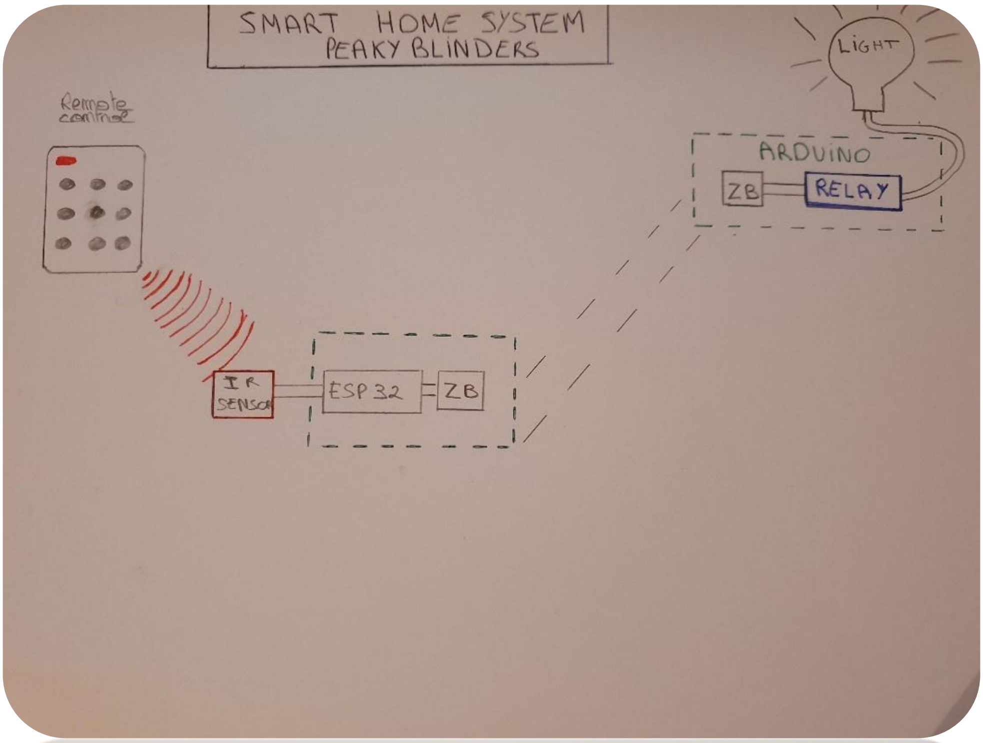
- Communicate with the different modules
- Handle the IR remote commands and decide what to do
- When a command from the main board is received, act on the output (ie: the relays).
- Use ZigBee module

We use ZigBee in order to build automation for a home(smart home). This home automation system will allow us to control lights and shutters through a remote control and our self-mobile application.

Why using it ?

The ZigBee module itself is not very expensive (30 euros) and does not need much power to work and is a wireless mesh network. For home appliances, we need low latency communication and it is exactly what we have with this module. The physical range can vary from 10 to 20 meters, which is good enough for a home.

Here is a little schema of our solution:



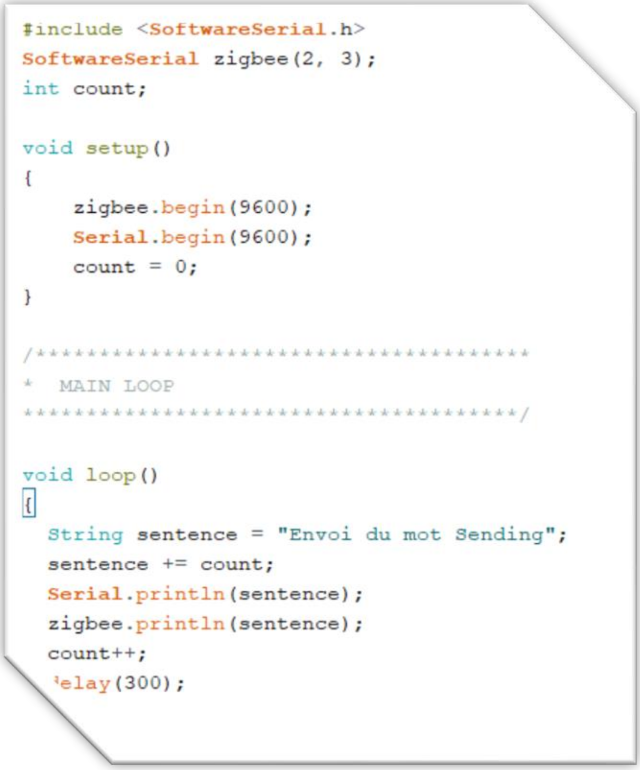
We can see that we have 2 ZigBee modules, 1 remote control, 1 relay, 1 IR sensor, and an ESP32.

How does it work?

On the one hand, our remote control is link to an IR sensor with the ESP32 and a ZigBee module. On the other hand, we have a ZigBee module with a relay connected to a light. Each ZigBee module will be associated to an Arduino, so we need also 2 Arduino (Or esp32). The objective is to assure the communication between the 2 Boards.

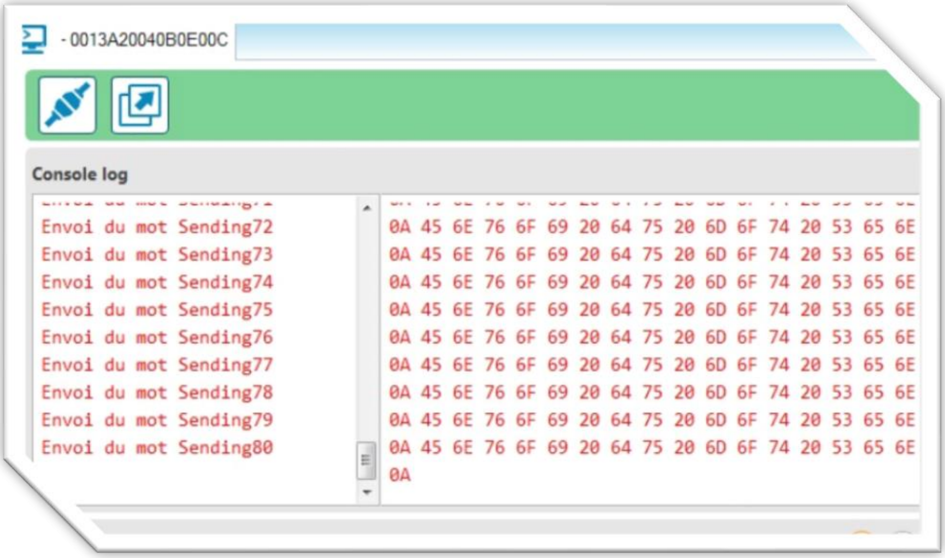
The most complicated part is to realize the communication between the two Arduino or Arduino/ESP32.

For example, if we want to transfer data to XCTU software, we have to use this code for the Arduino which is sending:



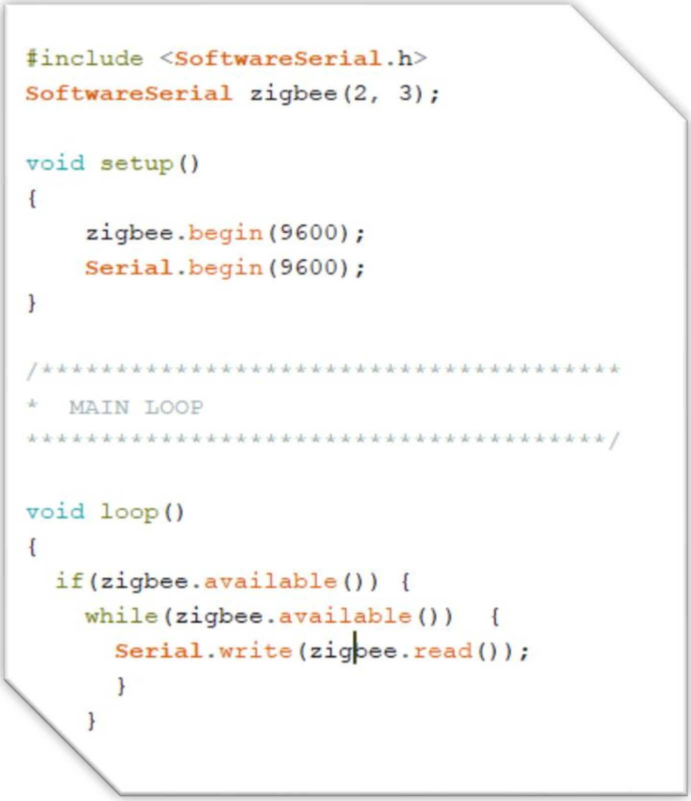
```
#include <SoftwareSerial.h>
SoftwareSerial zigbee(2, 3);
int count;

void setup()
{
    zigbee.begin(9600);
    Serial.begin(9600);
    count = 0;
}

/*****************************************
*  MAIN LOOP
*****************************************/

void loop()
{
  String sentence = "Envoi du mot Sending";
  sentence += count;
  Serial.println(sentence);
  zigbee.println(sentence);
  count++;
  'elay(300);
```

We are supposed to obtain this in XCTU:



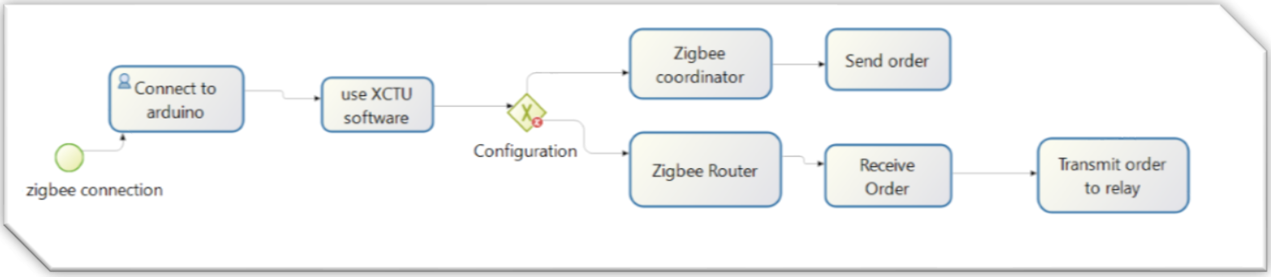
```
Envoi du mot Sending72     0A 45 6E 76 6F 69 20 64 75 20 6D 6F 74 20 53 65 6E
Envoi du mot Sending73     0A 45 6E 76 6F 69 20 64 75 20 6D 6F 74 20 53 65 6E
Envoi du mot Sending74     0A 45 6E 76 6F 69 20 64 75 20 6D 6F 74 20 53 65 6E
Envoi du mot Sending75     0A 45 6E 76 6F 69 20 64 75 20 6D 6F 74 20 53 65 6E
Envoi du mot Sending76     0A 45 6E 76 6F 69 20 64 75 20 6D 6F 74 20 53 65 6E
Envoi du mot Sending77     0A 45 6E 76 6F 69 20 64 75 20 6D 6F 74 20 53 65 6E
Envoi du mot Sending78     0A 45 6E 76 6F 69 20 64 75 20 6D 6F 74 20 53 65 6E
Envoi du mot Sending79     0A 45 6E 76 6F 69 20 64 75 20 6D 6F 74 20 53 65 6E
Envoi du mot Sending80     0A 45 6E 76 6F 69 20 64 75 20 6D 6F 74 20 53 65 6E
                           0A
```

Then we have to set up the Arduino which will receive data.

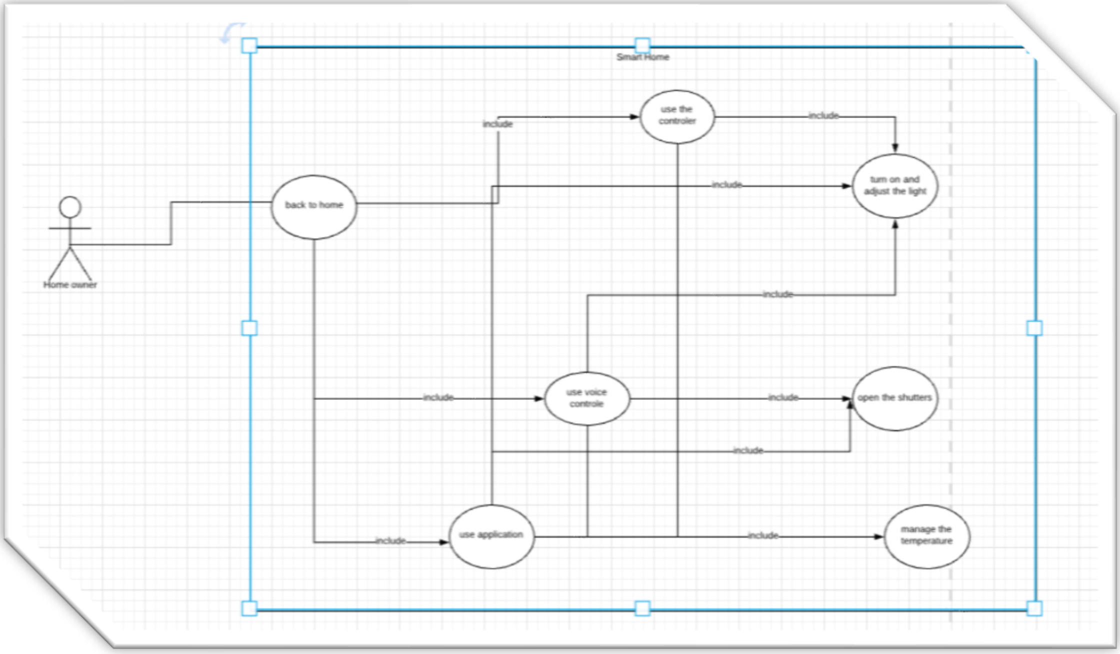


```
#include <SoftwareSerial.h>
SoftwareSerial zigbee(2, 3);

void setup()
{
    zigbee.begin(9600);
    Serial.begin(9600);
}

/****************************************
*  MAIN LOOP
****************************************/

void loop()
{
  if(zigbee.available()) {
    while(zigbee.available())  {
      Serial.write(zigbee.read());
      }
    }
```

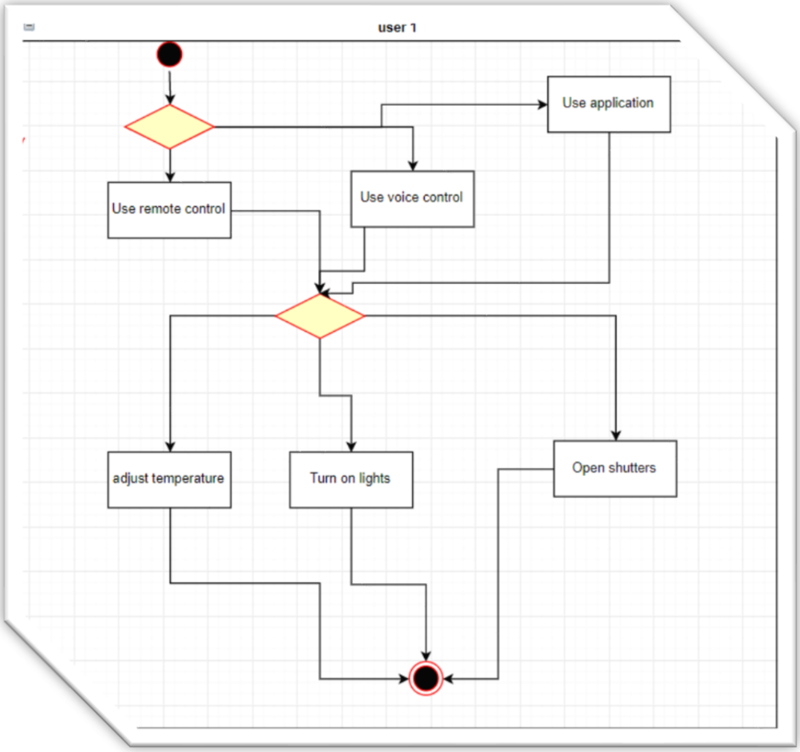
To resume the behaviour of ZigBee when we try to connect it to our system. We have realised a activity diagram.



## Use case diagram of our solution

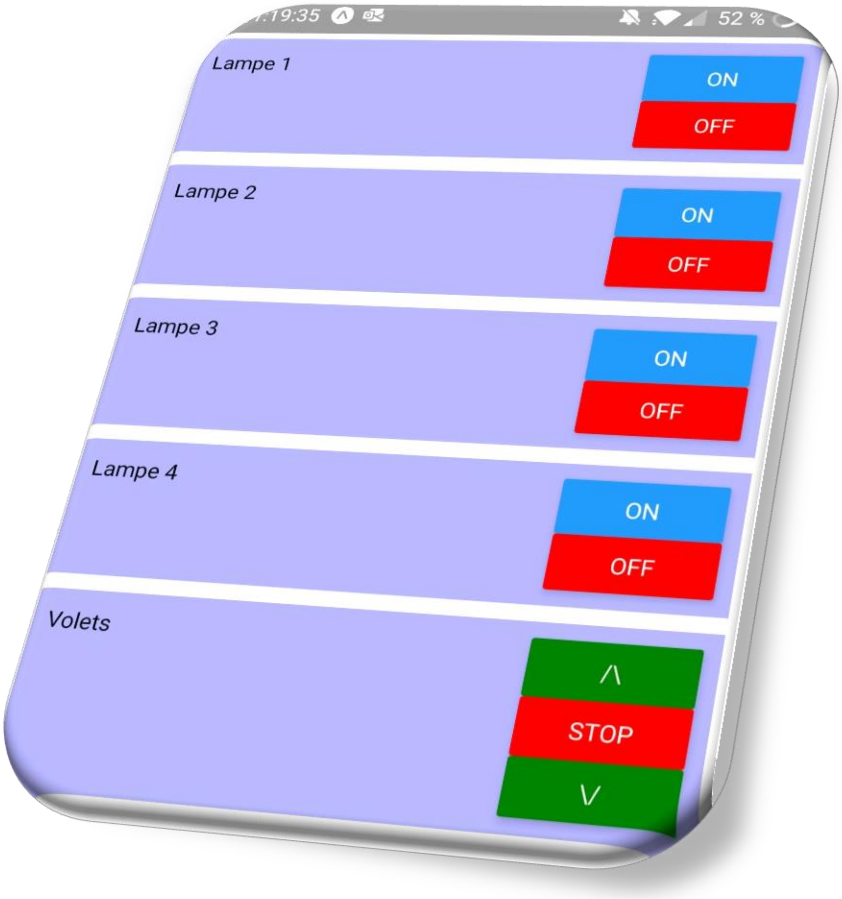


## Activity diagramme of our solution



We have also decided to design a mobile application to handle our smart home with WIFI using React Native.

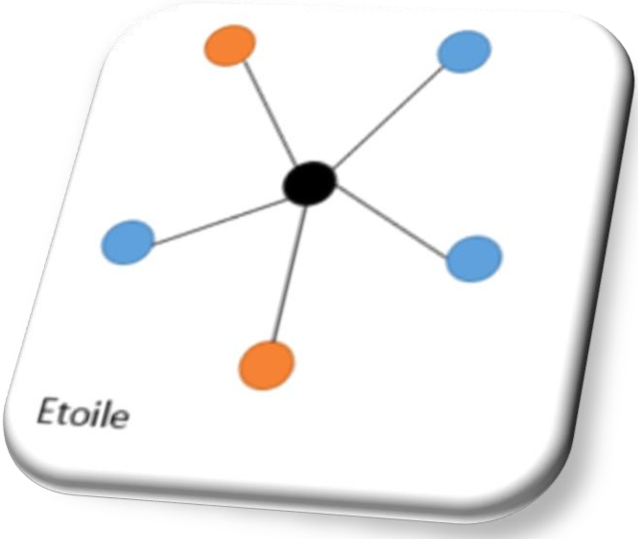
Design of the mobile application



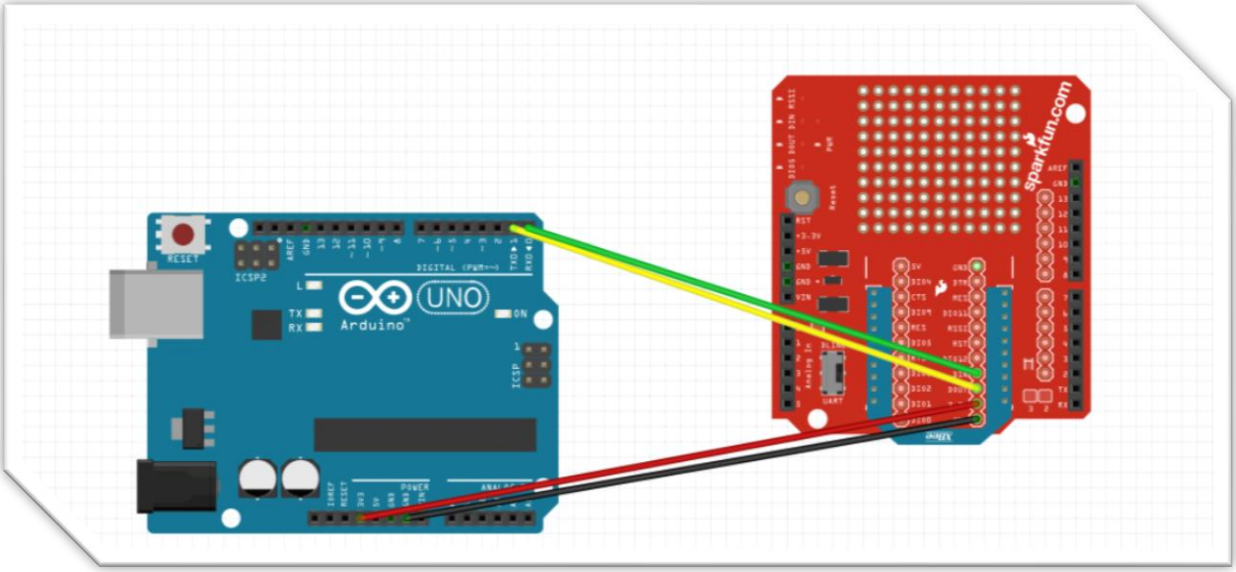
## Hardware design

The modules will be connected to ZigBee capable boards and to relays. They will be some kind of gateway between the main module and the remote modules. The shutters module will have also a distance sensor to detect the proper opening or closing of the shutters. The main module will have all the mandatory cards connected to it (Zigbee, IR receiver, WIFI)

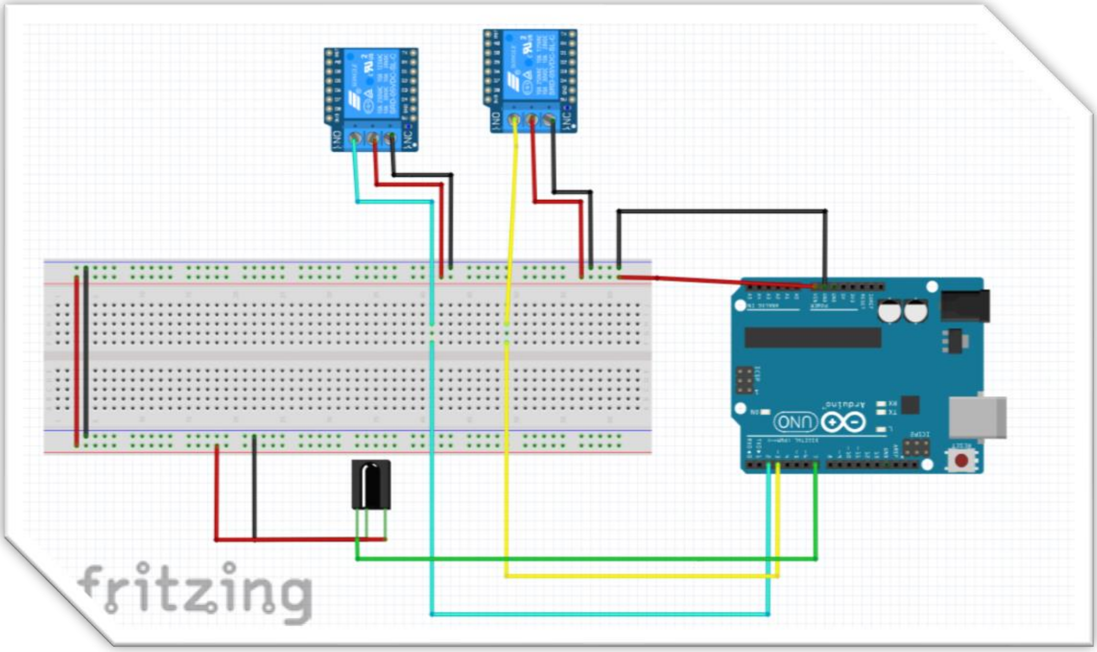
To build our ZigBee architecture we will use star topology. We have chosen this one because it is easier to implement, and it is efficient for less than ten devices.



We also realized a fritzing schema representing the connection between ZigBee and Arduino.



To connect IR remote controller to Arduino and to manage the relay we have used schema, below: (this schema is the basis, then it will be re-use with ESP32)



## Components list

- Hardware :
  - Rasberry Pi or arduino (we will choose later according to our needs)
  - Relay 7.89 euros

    https://www.amazon.fr/SeeKool-canaux-module-Arduino-Raspberry/dp/B07D8RF6MK/ref=sr_1_1_sspa?__mk_fr_FR=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crid=1OHK7S41SLL7K&keywords=relay+arduino&qid=1574289242&sprefix=Relay%2Caps%2C251&sr=8-1-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUExODZPOVBDUlhHNzNDJmVuY3J5cHRlZElkPUEwMTcwMDUzOENGMU9HUTlEUFFVJmVuY3J5cHRlZEFkSWQ9QTA0Mjk2NDcxSE1JUFlWSjFCQkdCJndpZGdldE5hbWU9c3BfYXRmJmFjdGlvbj1jbGlja1JlZGlyZWN0JmRvTm90TG9nQ2xpY2s9dHJ1ZQ==

  - Connection wires 0.99 euros

    https://www.amazon.fr/Paperllong%C2%AE-Multifonctionnel-Connexion-d%C3%A9finition-Utilisation/dp/B0818ZJQ82/ref=sr_1_2?__mk_fr_FR=%C3%85M%C3%85%C5%BD%C3%95%C3%91&keywords=o+Connection+wires&qid=1574289319&sr=8-2

  - Bluetooth or wifi module (ESP 32) use to connect arduino to phone app (useless with rasberry pi 4)
  - Diode 6.99 euros

https://www.amazon.fr/ATPWONZ-SuperBright-multicolore-%C3%A9mettant-couleurs/dp/B06X3VT6TD/ref=sr_1_fkmr0_2?__mk_fr_FR=%C3%85M%C3%85%C5%BD%C3%95%C3%91&keywords=o+Diode&qid=1574289625&sr=8-2-fkmr0

- Controller 15.99 euros

  https://www.amazon.fr/Timorn-Joystick-r%C3%A9paration-contr%C3%B4leur-Controller/dp/B07PNHCN21/ref=sr_1_1_sspa?__mk_fr_FR=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crid=3C1ALMTDJURP7&keywords=joystick+controller&qid=1574289744&sprefix=joystick+con%2Caps%2C140&sr=8-1-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEyMFBFTUpWOUU3RFNYJmVuY3J5cHRlZElkPUEwMTQxNDM4WE0yWk5HUjFLVk8zJmVuY3J5cHRlZEFkSWQ9QTAzMzA5MDYzUFdCQVpWSkZTSDJZJndpZGdldE5hbWU9c3BfYXRmJmFjdGlvbj1jbGlja1JlZGlyZWN0JmRvTm90TG9nQ2xpY2s9dHJ1ZQ==

- Electronic shutters already installed
- Electronic range finder
- Coordinator device for zigbee
- End-device for zigbee
- Zigbee routor 39.99 euros

  https://www.amazon.fr/Dresden-ConBee-Electronique-II/dp/B07PZ7ZHG5/ref=asc_df_B07PZ7ZHG5/?tag=googshopfr-21&linkCode=df0&hvadid=343205865330&hvpos=1o1&hvnetw=g&hvrand=16892286452071340399&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9056561&hvtargid=pla-699901973751&psc=1&tag=&ref=&adgrpid=66498663062&hvpone=&hvptwo=&hvadid=343205865330&hvpos=1o1&hvnetw=g&hvrand=16892286452071340399&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9056561&hvtargid=pla-699901973751

- Connect ampoule 35.25 euros

  https://www.amazon.fr/Eglo-ampoule-e27-blanc-15-5/dp/B0767HDPZX/ref=sr_1_fkmr0_1?__mk_fr_FR=%C3%85M%C3%85%C5%BD%C3%95%C3%91&keywords=o+Connect+ampoule&qid=1574290111&s=computers&sr=8-1-fkmr0

- SIM900 GSM GPRS module to send SMS 24.99 euros ( to send a sms if the window is open)

  https://www.amazon.fr/AZDelivery-SIM-900-bouclier-Arduino/dp/B01M9J4N56/ref=sr_1_fkmr2_1?__mk_fr_FR=%C3%85M%C3%

85%C5%BD%C3%95%C3%91&keywords=o+SIM900+GSM+GPRS+module&qid=1574290171&sr=8-1-fkmr2

- Temperatur sensor

# Anything else

I would like to introduce you our project application on disabled people. We can split two kind of disabled people, these which are stucked in bed and in wheelchair. With their situation, they don't have access to all their house windows, some are too high for example. Our project could be useful in these situations. Disabled people don't need to always keep controller with them. With our project their just need to say "open shutters" to deal with the system and open their shutter. It is more user friendly than the actual system with controller.

# Final solution

So, finally with our project the user could manage his light and shutters with two means: IR Controller and Smartphone application. After weighing pros and cons, we finally decided to use a raspberry pi instead of the ESP32 with our mobile application because we felt better with this stuff.

Here are two shorts videos showing the system:

https://youtu.be/_7JDXzcm5Do

https://youtu.be/hbYk7gBboac