

## Contrôle d'accès intelligent

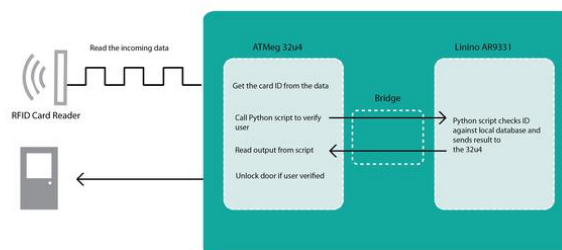
Pour les industriels ou les professionnels soucieux de maîtriser les allés et venues dans leurs locaux, on veut réaliser un système capable de contrôler l'accès avec un programme intelligent qui enregistrera les heures d'entrées ou de sortie du personnel et bloquera ou non l'accès à certaine zone à tout moment à partir d'une application Smartphone et de badges RFID



Matériel mis en œuvre :



Utilisation d'un lecteur de carte RFID, d'un Arduino YUN qui décode la trame et retrouve l'ID de la carte, il la compare à sa base de donnée locale et autorise ou non l'accès en fonction des droits alloués, ces droits peuvent être modifié à tout moment à partir d'une application Android.



L'avantage de l'Arduino Yun c'est qu'il dispose d'un noyau linux capable d'héberger un serveur LAMP et d'exécuter un script Python.

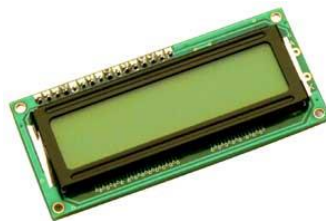
Matériel nécessaire

Arduino YUN avec sa carte  $\mu$ SD

Lecteur RFID



Afficheur

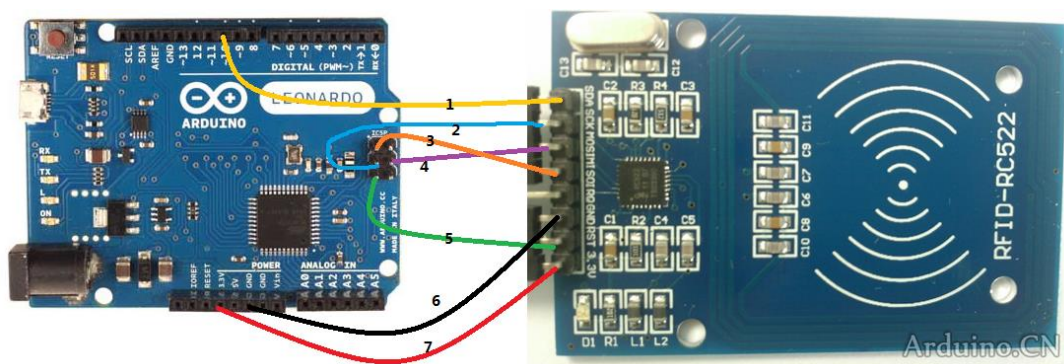


en option module Bluetooth, LED, Buzzer ...

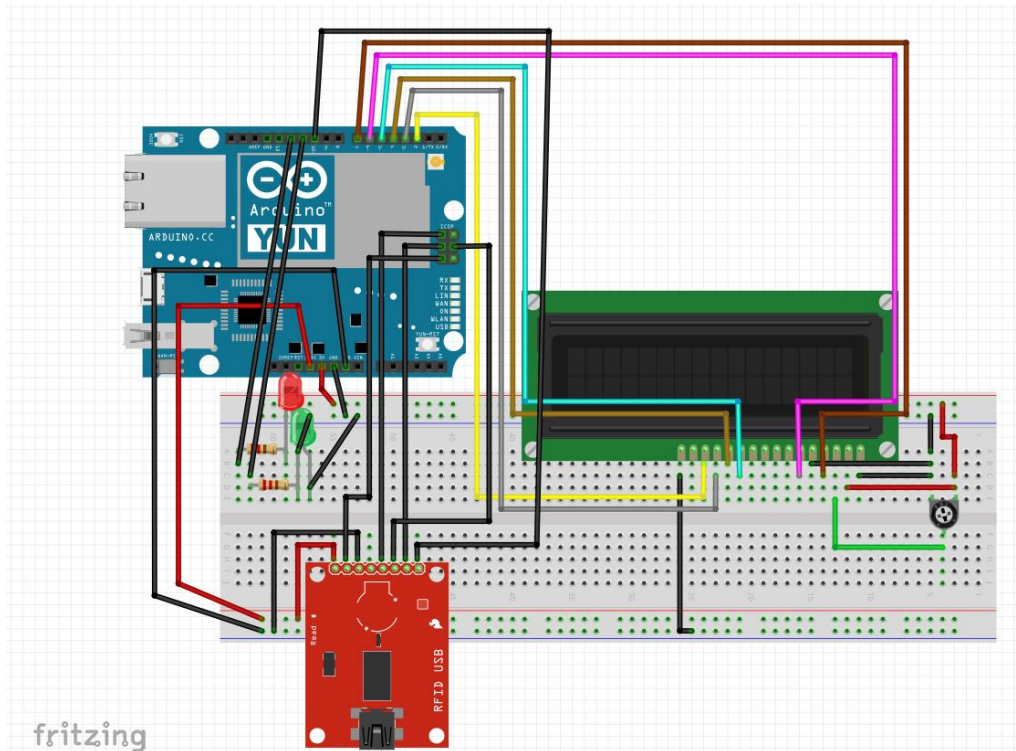
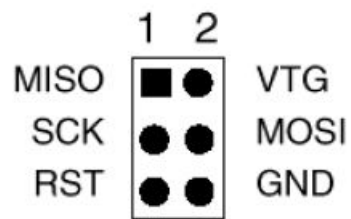


Le montage du capteur RFID

Pour raccorder un lecteur RC522 à un YUN il faut utiliser le port SPI

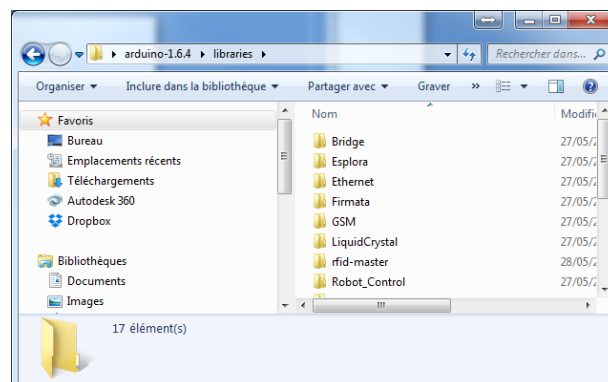


Le brochage du connecteur est le suivant la broche 10 est utilisé pour le SDA et le module est a alimenter en 3.3 V



Pour tester le montage utiliser la bibliothèque <https://github.com/miguelbalboa/rfid>

Copier le répertoire rfid-master dans la librairie Arduino



Copier le programme suivant pour tester le lecteur

/\*

\* .....

\* This is a MFRC522 library example; see <https://github.com/miguelbalboa/rfid>

\* for further details and other examples.

\*

\* NOTE: The library file MFRC522.h has a lot of useful info. Please read it.

\*

\* Released into the public domain.

\* -----

\* Example sketch/program showing how to read data from a PICC (that is: a RFID

\* Tag or Card) using a MFRC522 based RFID Reader on the Arduino SPI interface.

\*

\* When the Arduino and the MFRC522 module are connected (see the pin layout

\* below), load this sketch into Arduino IDE then verify/compile and upload it.

\* To see the output: use Tools, Serial Monitor of the IDE (hit Ctrl+Shft+M).

\* When you present a PICC (that is: a RFID Tag or Card) at reading distance

\* of the MFRC522 Reader/PCD, the serial output will show the ID/UID, type and

\* any data blocks it can read. Note: you may see "Timeout in communication"

\* messages when removing the PICC from reading distance too early.

\*

\* If your reader supports it, this sketch/program will read all the PICCs

\* presented (that is: multiple tag reading). So if you stack two or more

\* PICCs on top of each other and present them to the reader, it will first

\* output all details of the first and then the next PICC. Note that this

\* may take some time as all data blocks are dumped, so keep the PICCs at

\* reading distance until complete.

\*

\* Typical pin layout used:

\* -----

	MFRC522	Arduino	Arduino	Arduino	Arduino	Arduino
	Reader/PCD	Uno	Mega	Nano v3	Leonardo/Micro	Pro Micro
* Signal	Pin	Pin	Pin	Pin	Pin	

\* -----

* RST/Reset	RST	9	5	D9	RESET/ICSP-5	RST
-------------	-----	---	---	----	--------------	-----

* SPI SS	SDA(SS)	10	53	D10	10	10
----------	---------	----	----	-----	----	----

* SPI MOSI	MOSI	11 / ICSP-4	51	D11	ICSP-4	16
------------	------	-------------	----	-----	--------	----

* SPI MISO	MISO	12 / ICSP-1	50	D12	ICSP-1	14
------------	------	-------------	----	-----	--------	----

* SPI SCK	SCK	13 / ICSP-3	52	D13	ICSP-3	15
-----------	-----	-------------	----	-----	--------	----

\*/

```
#include <SPI.h>
```

```
#include <MFRC522.h>
```

```

#define RST_PIN          9           //
#define SS_PIN           10          //

MFRC522 mfrc522(SS_PIN, RST_PIN);  // Create MFRC522 instance

void setup() {
    Serial.begin(9600);              // Initialize serial communications
    with the PC

    while (!Serial);                // Do nothing if no serial port is opened
    (added for Arduinos based on ATMEGA32U4)

    SPI.begin();                    // Init SPI bus
    mfrc522.PCD_Init();             // Init MFRC522
    ShowReaderDetails();            // Show details of PCD
    Serial.println("Scan PICC to see UID, type, and data blocks...");
}

void loop() {

    // Look for new cards
    if ( ! mfrc522.PICC_IsNewCardPresent() ) {
        return;
    }

    // Select one of the cards
    if ( ! mfrc522.PICC_ReadCardSerial() ) {
        return;
    }

    // Dump debug info about the card; PICC_HaltA() is automatically
    mfrc522.PICC_DumpToSerial(&(mfrc522.uid));

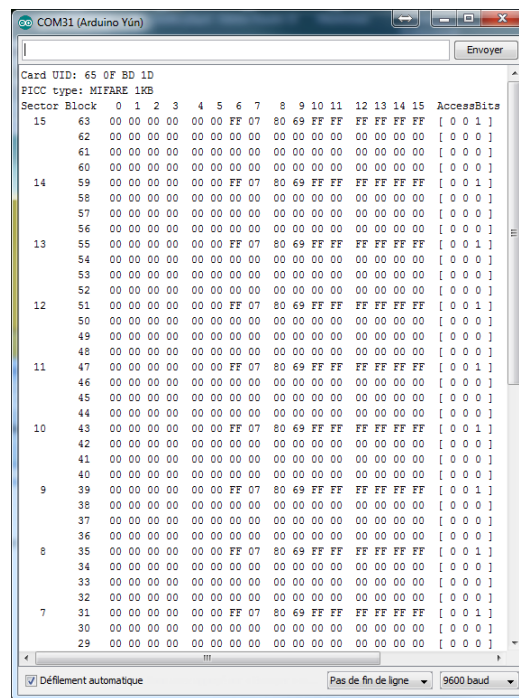
}

void ShowReaderDetails() {

    // Get the MFRC522 software version
    byte v = mfrc522.PCD_ReadRegister(mfrc522.VersionReg);
    Serial.print("MFRC522 Software Version: 0x");
    Serial.print(v, HEX);
    if (v == 0x91)
        Serial.print(" = v1.0");
    else if (v == 0x92)
        Serial.print(" = v2.0");
}

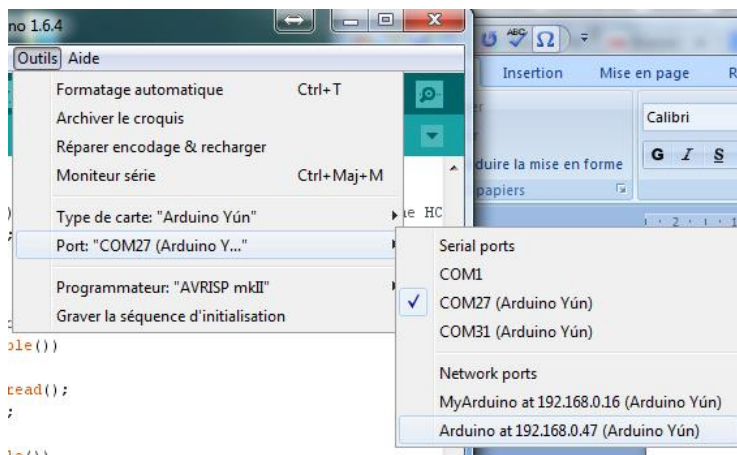
```

Passer un badge devant le lecteur pour vérifier le résultat sur le moniteur série

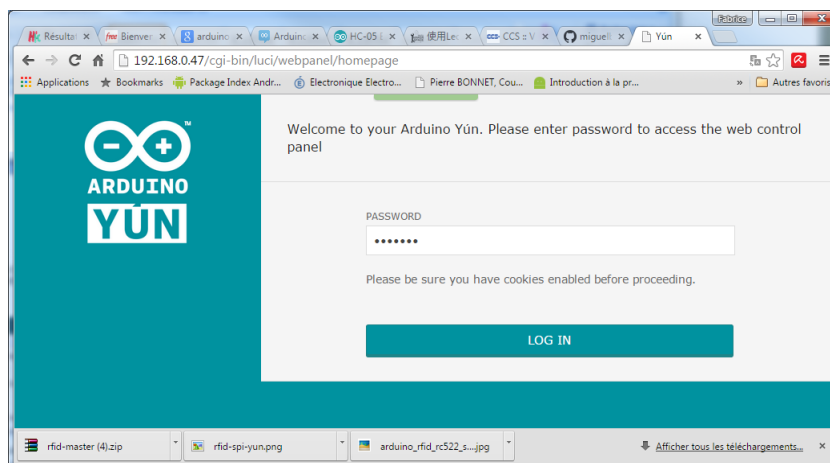


## Réalisation du programme de gestion intelligente de badge

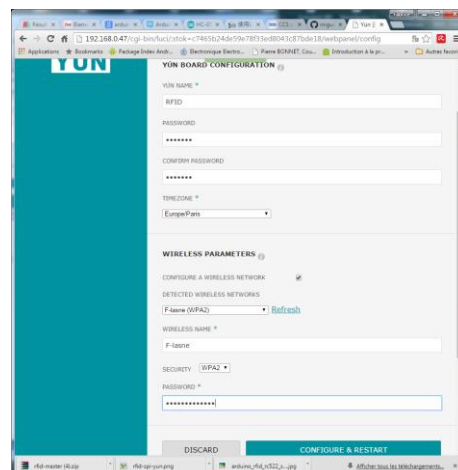
Il faut commencer par configurer la liaison Wifi de l'Arduino le plus rapide étant de passer par un câble RJ45 et de le brancher sur un réseau qui dispose d'un DHCP et de trouver l'adresse IP dans l'éditeur.



Ici en 192.168.0.47 (Arduino) avec un navigateur on peut configurer la liaison wifi le password par défaut est arduino



Le wifi est configuré par défaut comme point d'accès en 192.168.240.1, on change le nom en RFID avec comme password arduino-007 pour relier la carte au wifi local puis on redémarre la carte pour prendre en compte la configuration.





On peut passer par Luci pour configurer le serveur linux embarqué

[Applications](#) | [Bookmarks](#) | [Package Index And...](#) | [Electronique Changelog...](#) | [Autres favoris](#)

Rfid ID OpenWRT/NTY Attitude Adjustment 1 | Load: 0.31 0.23 0.14 Changes... Arduino Web Panel Administration

Status System Network Logout

Overview Firewall Routes System Log Kernel Log Processes Realtime Graphs

### Processes

This list gives an overview over currently running system processes and their status.

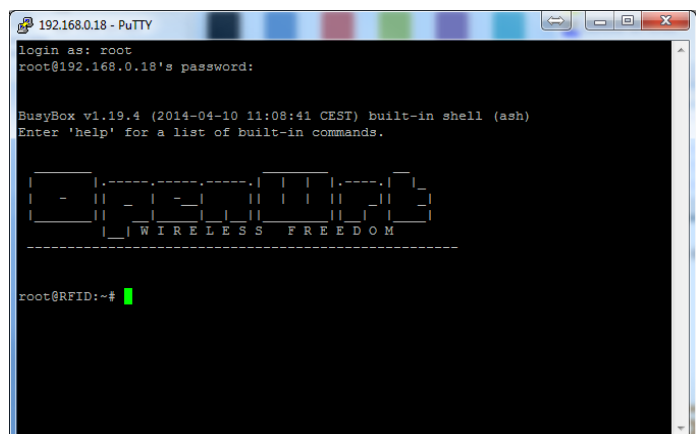
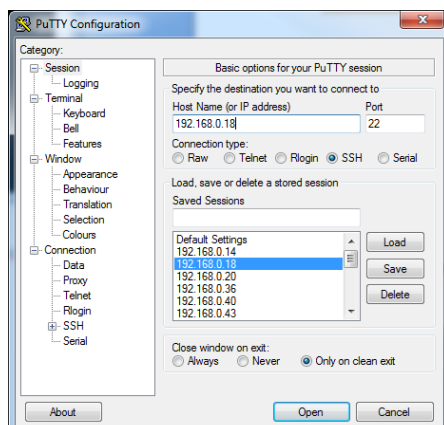
PID	Owner	Command	CPU usage (%)	Memory usage (%)	Hang Up	Terminate	Kill
1	root	[init]	0%	2%			
2	root	[kthread]	0%	0%			
3	root	[ksmoffd@0]	0%	0%			
5	root	[kworker@0]	0%	0%			
6	root	[klibsd]	0%	0%			
7	root	[kworker@ur1]	0%	0%			
90	root	[sync_supers]	0%	0%			
92	root	[bd-defauld]	0%	0%			
94	root	[klibsd]	0%	0%			
106	root	[dhcdd]	0%	0%			
132	root	[ksawpd@0]	0%	0%			
180	root	[fnctofy_mark]	0%	0%			
211	root	[ath79-spi]	0%	0%			
222	root	[mtmblock3]	0%	0%			
227	root	[mtmblock1]	0%	0%			
232	root	[mtmblock2]	0%	0%			
237	root	[mtmblock3]	0%	0%			
242	root	[mtmblock4]	0%	0%			
247	root	[mtmblock5]	0%	0%			
252	root	[mtmblock6]	0%	0%			
257	root	[mtmblock7]	0%	0%			
310	root	[kworker@0:1]	0%	0%			
328	root	[scsi_ah_0]	0%	0%			
329	root	[usb-storage]	0%	0%			
473	root	[kworker@0:2]	0%	0%			
528	root	[rfcc_gad_mw1]	0%	0%			
530	root	[flush-mtd-unmap]	0%	0%			
573	root	[rcS] bin/sh /etc/mtd/rcS	0%	2%			
576	root	logger -s -g s -t sysstat	0%	2%			
617	root	[df@0211]	0%	0%			

hd-master (4.2ip)    rfd-spj-jungun    Afficher tous les téléchargements...

Pour gérer les accès il faut disposer d'une base de donnée sur le serveur linux de l'Arduino et pour cela installer mysql .

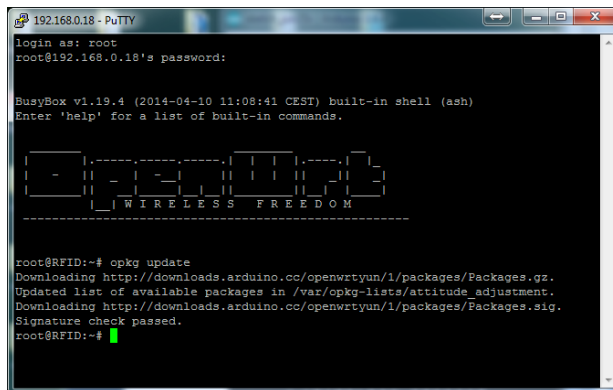


Pour installer des applications sur l'Arduino il faut passer par un client SSH comme putty avec l'adresse ip 192.168.0.18 du réseau wifi login root password arduino-007





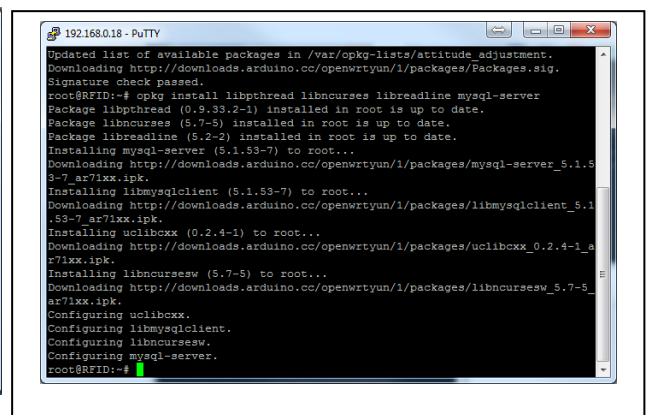
La première chose à faire est de mettre à jour le système avec la commande `opkg update`



```
192.168.0.18 - PuTTY
login as: root
root@192.168.0.18's password:

BusyBox v1.19.4 (2014-04-10 11:08:41 CEST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

root@RFID:~# opkg update
Downloading http://downloads.arduino.cc/openwrt/1/packages/Packages.gz.
Updated list of available packages in /var/opkg-lists/attitude_adjustment.
Downloading http://downloads.arduino.cc/openwrt/1/packages/Packages.sig.
Signature check passed.
root@RFID:~#
```



```
192.168.0.18 - PuTTY
Updated list of available packages in /var/opkg-lists/attitude_adjustment.
Downloading http://downloads.arduino.cc/openwrt/1/packages/Packages.sig.
Signature check passed.
root@RFID:~# opkg install libpthread libncurses libreadline mysql-server
Package libpthread (0.9.33.2-1) installed in root is up to date.
Package libncurses (5.7-5) installed in root is up to date.
Package libreadline (5.2-2) installed in root is up to date.
Installing mysql-server (5.1.53-7) to root...
Downloading http://downloads.arduino.cc/openwrt/1/packages/mysql-server_5.1.53-7_ar71xx.ipk.
Installing libmysqlclient (5.1.53-7) to root...
Downloading http://downloads.arduino.cc/openwrt/1/packages/libmysqlclient_5.1.53-7_ar71xx.ipk.
Installing uclibcxx (0.2.4-1) to root...
Downloading http://downloads.arduino.cc/openwrt/1/packages/uclibcxx_0.2.4-1_ar71xx.ipk.
Installing libncursesw (5.7-5) to root...
Downloading http://downloads.arduino.cc/openwrt/1/packages/libncursesw_5.7-5_ar71xx.ipk.
Configuring uclibcxx.
Configuring libmysqlclient.
Configuring libncursesw.
Configuring mysql-server.
root@RFID:~#
```

Et installer les paquets suivants pour la distribution OPENWRT

```
opkg install libpthread libncurses libreadline mysql-server
sed -i 's,^datadir.*,datadir      = /srv/mysql/g' /etc/my.cnf
sed -i 's,^tmpdir.*,tmpdir       = /tmp,g' /etc/my.cnf
mkdir -p /srv/mysql
mysql_install_db --force
```

Si vous avez un message d'erreur du type

*Neither host 'RFID' nor 'localhost' could be looked up with*

*/usr/bin/resolveip*

*Please configure the 'hostname' command to return a correct*

*hostname.*

*If you want to solve this at a later stage, restart this script*

*with the --force option*

créer un fichier host dans /etc

avec

```
127.0.0.1 localhost
192.168.0.18 RFID
```

Puis lancer la commande

`/etc/init.d/mysqld start` si vous avez un message d'erreur

`mysql_install_db --force` (ça passe comme ça chez moi ...)

Lancer le serveur mysql et créer un mot de passe pour l'utilisateur root ('pass')

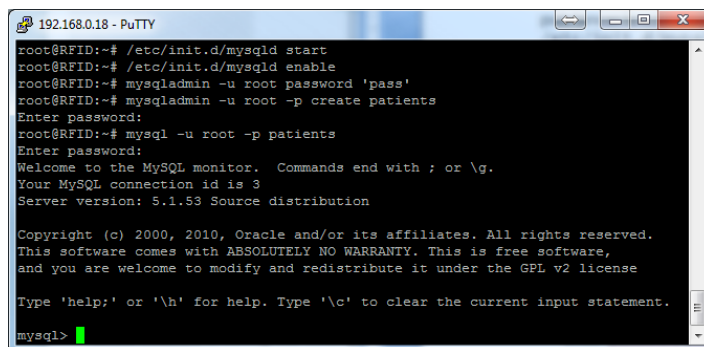
```
/etc/init.d/mysql start
/etc/init.d/mysql enable
mysqladmin -u root password 'pass'
```

Créer une table patients

```
mysqladmin -u root -p create patients
Enter password pass
```

Se connecter a l'interface de commande

```
mysql -u root -p patients avec pass comme password
```



```
192.168.0.18 - PuTTY
root@RFID:~# /etc/init.d/mysql start
root@RFID:~# /etc/init.d/mysql enable
root@RFID:~# mysqladmin -u root password 'pass'
root@RFID:~# mysqladmin -u root -p create patients
Enter password:
root@RFID:~# mysql -u root -p patients
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.1.53 Source distribution

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

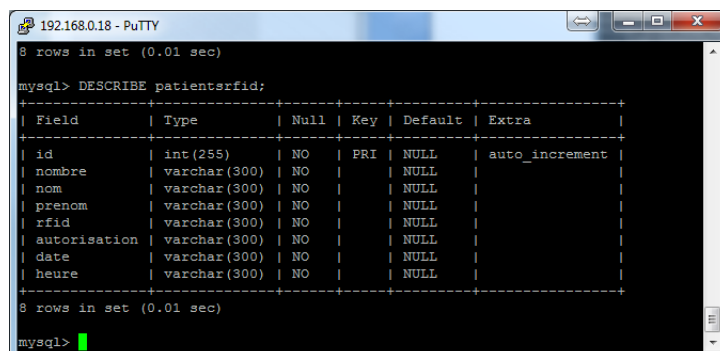
mysql>
```

créer un fichier patientsrfid

```
CREATE TABLE `patientsrfid` ( `id` int(255) NOT NULL AUTO_INCREMENT, `nombre`
varchar(300) NOT NULL, `nom` varchar(300) NOT NULL, `prenom` varchar(300) NOT NULL, `rfid`
varchar(300) NOT NULL, `autorisation` varchar(300) NOT NULL, `date` varchar(300) NOT
NULL, `heure` varchar(300) NOT NULL, PRIMARY KEY (`id`) ) ENGINE=MyISAM DEFAULT
CHARSET=latin1 AUTO_INCREMENT=0;
```

voir la table crée [SHOW TABLES;](#)

voir son contenu [DESCRIBE patientsrfid;](#)



```
192.168.0.18 - PuTTY
8 rows in set (0.01 sec)

mysql> DESCRIBE patientsrfid;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(255) | NO | PRI | NULL | auto_increment |
| nombre | varchar(300) | NO | | NULL | |
| nom   | varchar(300) | NO | | NULL | |
| prenom | varchar(300) | NO | | NULL | |
| rfid  | varchar(300) | NO | | NULL | |
| autorisation | varchar(300) | NO | | NULL | |
| date  | varchar(300) | NO | | NULL | |
| heure | varchar(300) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.01 sec)

mysql>
```

On créer également une table controle pour vérifier les horaires de passage

```
CREATE TABLE `controle` ( `id` int(255) NOT NULL AUTO_INCREMENT, `rfid` varchar(300)
NOT NULL, `heure` varchar(300) NOT NULL, `date` varchar(300) NOT NULL, PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=0;
```

```
192.168.0.18 - PuTTY
mysql> describe controle;
+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+
| id     | int(255)      | NO   | PRI | NULL    | auto_increment |
| rfid   | varchar(300)  | NO   |     | NULL    |                |
| heure  | varchar(300)  | NO   |     | NULL    |                |
+-----+
3 rows in set (0.01 sec)

mysql> select * from controle;
+-----+
| id | rfid   | heure                |
+-----+
| 1  | 650FBD1D | 26/06/2015 01:57:27 |
+-----+
1 row in set (0.00 sec)

mysql>
```

On peut entrer la première valeur de carte à la main pour vérifier le fonctionnement de la base.

```
INSERT INTO patientsrfid
(id,nombre,nom,prenom,rfid,autorisation,date,heure) VALUES
('','1','Lasne','Fabrice','650FBD1D','0','25/06/2015','17:58');
```

Vérifier que la table c'est bien rempli.

```
select * from patientsrfid;
```

```
192.168.0.18 - PuTTY
mysql> INSERT INTO patientsrfid (id,nombre,nom,prenom,rfid,autorisation,date,heure) VALUES ('','1','Lasne','Fabrice','650FBD1D','0','25/06/2015','17:58');
Query OK, 1 row affected, 1 warning (0.01 sec)

mysql> select * from patientsrfid
-> ;
+-----+
| id | nombre | nom   | prenom | rfid   | autorisation | date       | heure |
+-----+
| 1  | 1       | Lasne | Fabrice | 650FBD1D | 0            | 25/06/2015 | 17:58 |
+-----+
1 row in set (0.00 sec)

mysql>
```

Pour traiter les requêtes à partir d'un serveur web il faut installer PHP5



Quitter le serveur mysql avec la commande `exit` et installer PHP5 avec la commande

```
opkg install php5 php5-cgi
```

Si vous avez un message d'erreur faites un `opkg update` avant ;)

```
192.168.0.18 - PuTTY
mysql> opkg install php5 php5-cgi
-> Ctrl-C -- exit!
Aborted
root@RFID:~# opkg install php5 php5-cgi
Unknown package 'php5'.
Unknown package 'php5-cgi'.
Collected errors:
 * opkg_install_cmd: Cannot install package php5.
 * opkg_install_cmd: Cannot install package php5-cgi.
root@RFID:~# opkg update
Downloading http://downloads.arduino.cc/openwrt/1/packages/Packages.gz.
Updated list of available packages in /var/opkg-lists/attitude_adjustment.
Downloading http://downloads.arduino.cc/openwrt/1/packages/Packages.sig.
Signature check passed.
root@RFID:~# opkg install php5 php5-cgi
Installing php5 (5.4.5-3) to root...
Downloading http://downloads.arduino.cc/openwrt/1/packages/php5_5.4.5-3_ar71xx.ipk.
Installing libxml2 (2.7.8-2) to root...
Downloading http://downloads.arduino.cc/openwrt/1/packages/libxml2_2.7.8-2_ar71xx.ipk.
Installing php5-cgi (5.4.5-3) to root...
Downloading http://downloads.arduino.cc/openwrt/1/packages/php5-cgi_5.4.5-3_ar71xx.ipk.
Configuring libxml2.
Configuring php5.
```

Configuration du serveur web uHTTPd de l' arduino avec VI (editeur linux)

`vi /etc/config/uhttpd`

Appuyer sur i pour éditer le fichier et supprimer le # devant la ligne list interpreter ". php = / usr / bin / php-cgi taper sur la touche ESC pour passer en mode commande et taper :wq puis sur entrer pour enregistrer et quitter l'éditeur

Redémarrer le serveur uHTTPd pour prendre en compte les modifications

`/etc/init.d/uhttpd restart`

Installer Mysql for PHP5 et Python



On installe la passerelle entre python et php avec les commandes suivantes :

```
opkg install php5-mod-mysql
sed -i 's,;extension=mysql.so,extension=mysql.so,g' /etc/php.ini
opkg install python-mysql
```

```
192.168.0.18 - PuTTY
#      list interpreter      ".cgi=/usr/bin/perl"

# Lua url prefix and handler script.
# Lua support is disabled if no prefix given.
root@RFID:~# /etc/init.d/uhttpd restart
root@RFID:~# opkg install php5-mod-mysql
Installing php5-mod-mysql (5.4.5-3) to root...
Downloading http://downloads.arduino.cc/openwrt-yun/1/packages/php5-mod-mysql_5.4.5-3_ar71xx.ipk.
Configuring php5-mod-mysql.
root@RFID:~# sed -i 's,;extension=mysql.so,extension=mysql.so,g' /etc/php.ini
root@RFID:~# opkg install python-mysql
Installing python-mysql (1.2.2-1) to root...
Downloading http://downloads.arduino.cc/openwrt-yun/1/packages/python-mysql_1.2.2-1_ar71xx.ipk.
Configuring python-mysql.
root@RFID:~#
```

Voila le serveur LAM est installé sur le YUN

Code PYTHON



Pour communiquer entre l'Arduino et la base de donnée Mysql on utilisera un langage python

On se connecte à la base pour vérifier que le badge a l'autorisation d'ouverture avec le script valid.py

On vérifie que le N° RFID correspond à un nom dans la base si oui on regarde s'il est autorisé ou pas

```
import MySQLdb
import sys

host = "127.0.0.1"
user = "root"
passwd = "pass"
base = "patients"

while True:
    db= MySQLdb.connect(host,user,passwd,base)
    cur = db.cursor()
    resultat = cur.execute("""SELECT nom FROM patientsrfid WHERE rfid
LIKE %s ORDER BY id""",(sys.argv[1],))
    nom = cur.fetchall()

    if (resultat == 1 ):
        cur1 = db.cursor()
        resultat1 =cur1.execute("""SELECT nom FROM patientsrfid
WHERE rfid LIKE %s AND autorisation ="1" ORDER BY id""",(sys.argv[1],))

        if (resultat1 == 1 ):
            print "1" , nom
            sys.exit(1)
        else:
            print "2" , nom
            sys.exit(1)
    else:
        print "0"
        sys.exit(1)
```

la fonction renvoie 0 1 ou 2

0 le badge n'est pas reconnu

1 Le badge existe et est autorisé on renvoie le nom

2 Le badge existe mais n'est pas autorisé on renvoie le nom

Pour stoker les horaires d'entrées/ sorties on réalise le script control.py

```
import MySQLdb
```

```
import sys
```

```
import time
```

```
host = "127.0.0.1"
```

```
user = "root"
```

```
passw = "pass"
```

```
base = "patients"
```

```
temps = time.strftime("%d/%m/%Y %H:%M:%S", time.localtime())
```

```
db= MySQLdb.connect(host,user,passw,base)
```

```
cur = db.cursor()
```

```
resultat = cur.execute(" INSERT INTO controle (rfid,valeur) VALUES (%s,%s)  
", (sys.argv[1], temps,))
```

```
if (resultat == 1):
```

```
    print 1
```

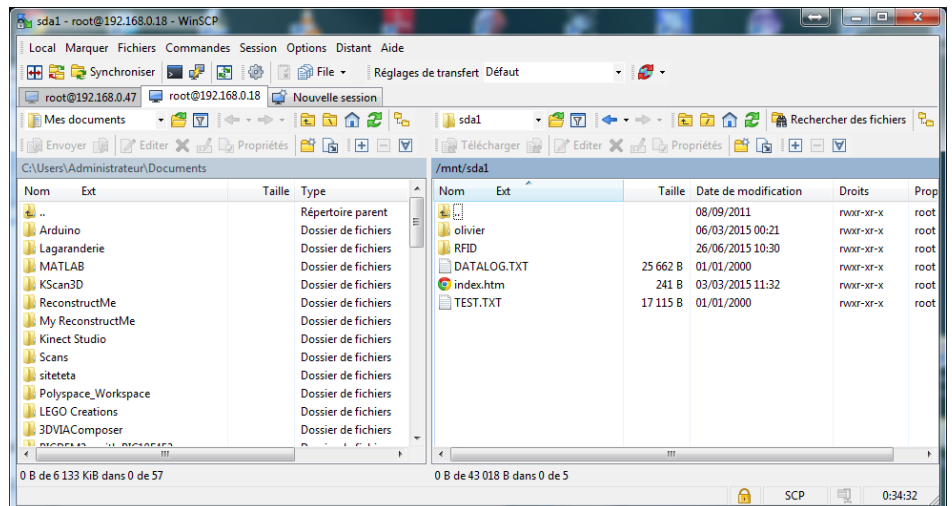
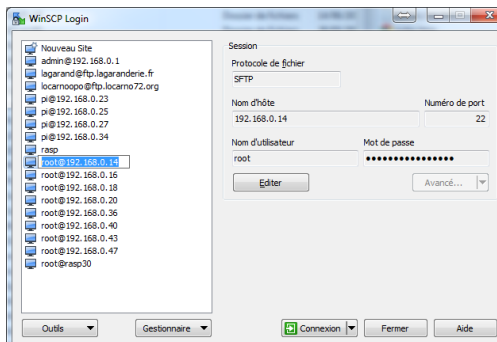
```
    sys.exit(1)
```

```
else:
```

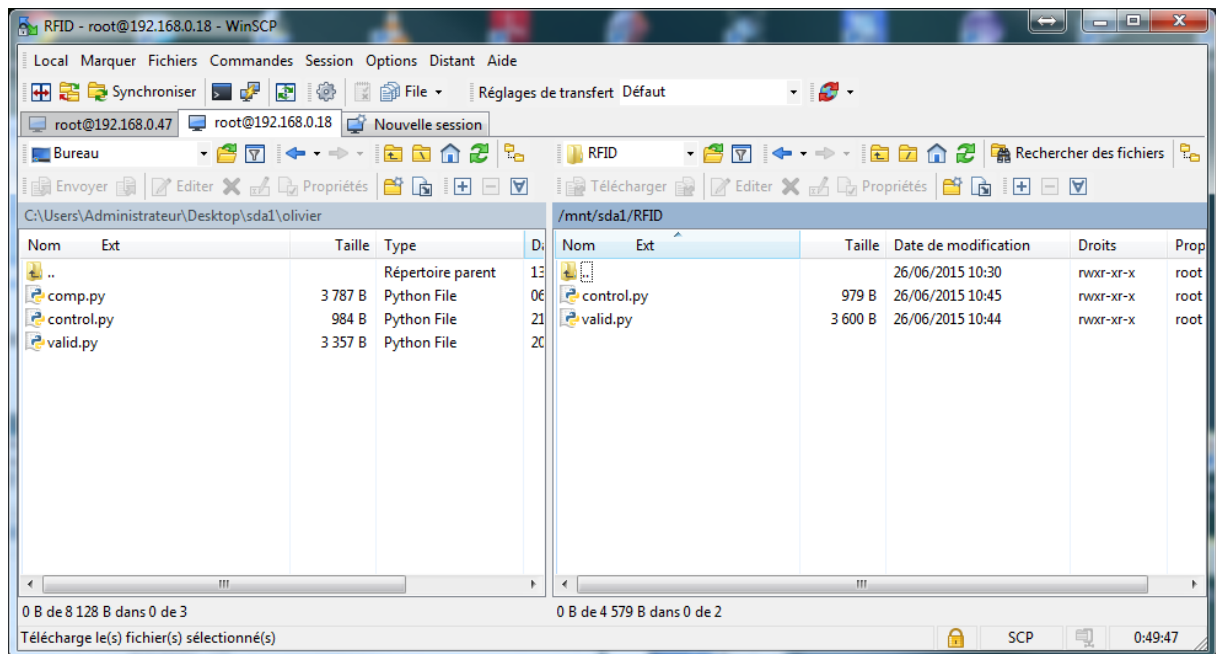
```
    print 2
```

```
    sys.exit(1)
```

Ces deux fichiers doivent être déposés dans le YUN, j'utilise WinSCP pour le transfert vers la carte SD



je place les scripts dans un répertoire RFID dans mnt/sda1 (sur la carte µSD)



Le passage d'un badge devant le lecteur doit lancer ce script qui fait appel à la base de donnée et qui lui retourne l'autorisation

code Arduino avec l'afficheur et les LED

1er programme vérification du fonctionnement du lecteur et de l'afficheur avec deux cartes RFID  
N°650FBD1D Fabrice et 0571CC1D Olivier

//Programme : RFID - Controle d'accès intelligent

//Autor : F.LASNE 2015

/\*

\* -----

\* This is a MFRC522 library example; see <https://github.com/miguelbalboa/rfid>

\* for further details and other examples.

\*

\* NOTE: The library file MFRC522.h has a lot of useful info. Please read it.

\*

\* Released into the public domain.

\* -----

\*/

#include <SPI.h>



```

#include <MFRC522.h>

#include <LiquidCrystal.h>

#include <Bridge.h>

#include <Process.h>


// Pins used to connect the display to the Arduino
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);


#define RST_PIN 8
#define SS_PIN 9


MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance.


Process date; //gestion du temps Arduino.


void setup()
{
    Serial.begin(9600); // Initialize serial communications with the PC
    while (!Serial); // Do nothing if no serial port is opened (added for Arduinos based on ATMEGA32U4)
    SPI.begin(); // Init SPI bus
    // Bridge.begin(); // initialisation mode Bridge
    mfrc522.PCD_Init(); // Init MFRC522
    mfrc522.PCD_SetAntennaGain(mfrc522.RxGain_max); //Set Antenna Gain to Max- this will increase
    reading distance

    // initialize digital pin 12 et 13 as an output.
    pinMode(12, OUTPUT);
    pinMode(13, OUTPUT);


    /*---Init temp ---*/
    if (!date.running()) {
        date.begin("date");
        date.addParameter("+%T");
    }
}

```

```

    date.run();
}

/*--- FIN init temp ---*/

Serial.println("Lecture Carte ...");

Serial.println();

lcd.begin(16, 2);

mensageminicial();
}

void loop()
{
    digitalWrite(12, LOW); // turn the LED off by making the voltage LOW
    digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
    // Look for new cards
    if ( ! mfrc522.PICC_IsNewCardPresent())
    {
        return;
    }

    // Select one of the cards
    if ( ! mfrc522.PICC_ReadCardSerial())
    {
        return;
    }

    lcd.clear();

    lcd.setCursor(0,0);

    lcd.print("Verification");

```

```

//lecture UID du Badge

//Serial.print("UID du tag :");

String contenu= "";

//byte lettre;

for (byte i = 0; i < mfrc522.uid.size; i++)
{
    //Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? "0" : "");

    //Serial.print(mfrc522.uid.uidByte[i], HEX);

    contenu.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? "0" : ""));

    contenu.concat(String(mfrc522.uid.uidByte[i], HEX));
}

contenu.toUpperCase();

Serial.println("Le GUID est:");

Serial.print(contenu);

Serial.println();

Serial.println("Verification ...");

// verifRFID(contenu);

if (contenu == "650FBD1D") //UID 1 - fabrice
{
    Serial.println("Bonjour Fabrice !");

    Serial.println();

    lcd.clear();

    lcd.setCursor(0,0);

    lcd.print("Bonjour Fabrice !");

    lcd.setCursor(0,1);

    lcd.print("Ouverture porte!");

    digitalWrite(12, HIGH); // turn the LED on (HIGH is the voltage level)

    delay(3000);

    mensageminicial();
}

```

```

if (contenu == "0571CC1D") //UID 2 - OLIVIER
{
    Serial.println("Bonjour Olivier");
    Serial.println();
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Bonjour Olivier");
    lcd.setCursor(0,1);
    lcd.print("Acces REFUSE");
    digitalWrite(11, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(3000);
    mensageminicial();
}
}

void mensageminicial()
{
    lcd.clear();
    lcd.print("Passer le badge");
    lcd.setCursor(0,1);
    lcd.print("sur le lecteur");
}

```

Chaque carte affiche un message unique

Pour utiliser le script python on utilise le programme suivant avec la fonction verifRFID

Programme N°2 complet avec heure de sortie

```

//Programme : RFID - Controle d'accès intelligent
//Autor : F.LASNE 2015
/*

```

\* -----

\* This is a MFRC522 library example; see <https://github.com/miguelbalboa/rfid>

\* for further details and other examples.

\*

\* NOTE: The library file MFRC522.h has a lot of useful info. Please read it.

\*

\* Released into the public domain.

\* -----

\*

\* Typical pin layout used:

	MFRC522	Arduino	Arduino	Arduino	Arduino	Arduino
	Reader/PCD	Uno	Mega	Nano v3	Leonardo/Micro	Pro Micro
* Signal	Pin	Pin	Pin	Pin	Pin	
* RST/Reset	RST	9	5	D9	RESET/ICSP-5	RST
* SPI SS	SDA(SS)	10	53	D10	10	
* SPI MOSI	MOSI	11 / ICSP-4	51	D11	ICSP-4	16
* SPI MISO	MISO	12 / ICSP-1	50	D12	ICSP-1	14
* SPI SCK	SCK	13 / ICSP-3	52	D13	ICSP-3	15
* /						

```
#include <SPI.h>
#include <MFRC522.h>
#include <LiquidCrystal.h>
#include <Bridge.h>
#include <Process.h>
#include <SoftwareSerial.h>
```

```
// Pins used to connect the display to the Arduino
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
```

```
// Pins used to connect the Bluetooth module HC06
SoftwareSerial mySerial(10, 11); // RX, TX
```

```
// Pins used to connect the RFID
#define RST_PIN 8
#define SS_PIN 9
```

```
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance.
```

```
Process date; //gestion du temps Arduino.
```

```
void setup()
{
  Serial.begin(9600);           // Initialize serial communications with the PC
  while (!Serial);             // Do nothing if no serial port is opened (added for Arduinos based on
  ATMEGA32U4)                  // Rajout d'un port com pour lecteur bluetooth (pas serial1 a cause du
  mySerial.begin(9600);         // bridge)
  // mySerial.println("BONJOUR");
  SPI.begin();                 // Init SPI bus
  // Serial.println("SPI OK ...");
  Bridge.begin();              // initialisation mode Bridge
  // Serial.println("BRIDGE OK ...");
  mfrc522.PCD_Init();          // Init MFRC522
```

```
mfr522.PCD_SetAntennaGain(mfr522.RxGain_max); //Set Antenna Gain to Max- this will increase reading distance
```

```
// initialize digital pin 12 et 13 as an output.  
pinMode(12, OUTPUT);  
pinMode(13, OUTPUT);
```

```
/*---Init temp ---*/  
if (!date.running()) {  
    date.begin("date");  
    date.addParameter("+%T");  
    date.run();  
}  
/*--- FIN init temp ---*/
```

```
Serial.println("Lecture Carte ...");  
Serial.println();
```

```
lcd.begin(16, 2);  
messageminicial();  
}
```

```
void loop()  
{  
    digitalWrite(12, LOW); // turn the LED RED OFF  
    digitalWrite(13, LOW); // turn the LED GREEN OFF
```

```
// Look for new cards  
if ( ! mfr522.PICC_IsNewCardPresent())  
{  
    return;  
}
```

```
// Select one of the cards  
if ( ! mfr522.PICC_ReadCardSerial())  
{  
    return;  
}
```

```
lcd.clear();  
lcd.setCursor(0,0);  
lcd.print("Verification");
```

```
//lecture UID du Badge  
//Serial.print("UID du tag :");  
String contenu= "";  
//byte lettre;  
for (byte i = 0; i < mfr522.uid.size; i++)  
{  
    //Serial.print(mfr522.uid.uidByte[i] < 0x10 ? "0" : "");  
    //Serial.print(mfr522.uid.uidByte[i], HEX);  
    contenu.concat(String(mfr522.uid.uidByte[i] < 0x10 ? "0" : ""));  
    contenu.concat(String(mfr522.uid.uidByte[i], HEX));  
}
```

```
contenu.toUpperCase();
```

```

Serial.println("Le GUID est:");
Serial.print(contenu);
Serial.println();
Serial.println("Verification ...");
verifRFID(contenu);

```

```

}

```

```

void verifRFID(String IDrfid){
  Process verif;
  verif.begin("python");
  verif.addParameter("/mnt/sda1/RFID/valid.py");
  verif.addParameter(IDrfid); // Appel de la fonction en python valid.py
  verif.run();
  while (verif.available()){
    String resultatPython = verif.readString(); // lecture du resultat de la fonction
    resultatPython.trim(); // 0 ou 1 ou 2 en string

```

```

    String val = resultatPython.substring(0,1);
    Serial.println(val);

```

```

    if(val == "0"){ // badge RFID non reconnu
      Serial.println("Acces Refuse !");
      Serial.println();
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("NON VALIDE");
      lcd.setCursor(0,1);
      lcd.print("Acces REFUSE");
      digitalWrite(12, HIGH); // led rouge allumée
      delay(3000);
      messageminicial();
      break;

```

```

    }
    else if (val == "1"){ //Badge vérifié et OK

```

```

      Serial.println("Acces Autorise");
      resultatPython = resultatPython.substring(5,(resultatPython.length()-5));
      Serial.println(resultatPython);
      Serial.println();
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("Bonjour M ");
      lcd.print(resultatPython);
      lcd.setCursor(0,1);
      lcd.print("Ouverture porte!");
      digitalWrite(13, HIGH); // LED vert allumée
      delay(3000);
      controle(IDrfid);
      messageminicial();
      break;

```

```

    }
    else { //Badge vérifié OK mais pas autorisé

```



```

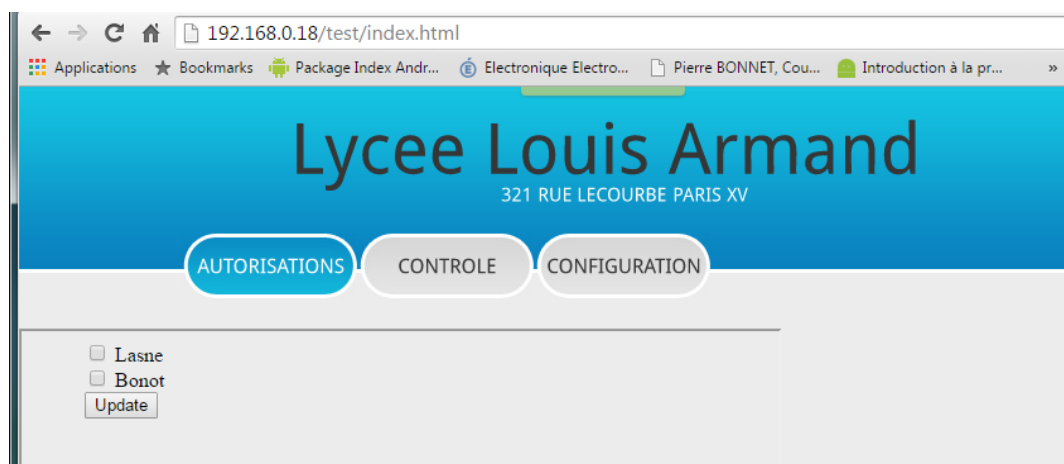
Serial.println("Acces Refusé");
resultatPython = resultatPython.substring(5,(resultatPython.length()-5));
Serial.println(resultatPython);
Serial.println();
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Bonjour M ");
lcd.print(resultatPython);
lcd.setCursor(0,1);
lcd.print("NON Autorise");
digitalWrite(12, HIGH);          // LED rouge allumée
delay(3000);
messageminicial();
break;
}
}
}

void controle(String IDrfid){
  Serial.println("ajout contrôle");
  Process control;
  control.begin("python");
  control.addParameter("/mnt/sda1/RFID/control.py");
  control.addParameter(IDrfid); // Appel de la fonction en python controle qui ajoute l'heure de sortie
  control.run();
}

void messageminicial()
{
  lcd.clear();
  lcd.print("Passer le badge");
  lcd.setCursor(0,1);
  lcd.print("sur le lecteur");
}

```

Nous allons créer une interface pour gérer les accès à partir de pages WEB hébergées dans l'Arduino. Une page pour valider les autorisation une page pour le contrôle des entrées/ sortie et une pour entrer les noms et les numéros de badges



Elle fait appel à une page qui liste les entrées Mysql et qui affiche sous forme de cases à cocher les autorisations (pas si simple ...)

```

<?php
include('config.php');

if(isset($_POST['id'])) //la condition sera true si il y a un tableau d'id

{
    $listId=$_POST['id'];
    if(isset($_POST['CBTest'])) // si il y a au moins un checkbox coché
    {
        ;

        $check=$_POST['CBTest'];
        $noCheck = array_diff($listId, $check); //difference entre list complete des id et id cherchée
        $sizeNocheck=sizeof($noCheck);

        //var_dump($noCheck);

        foreach( $check as $key => $val) //on update d'abord les id qui ont été checké
        {
            mysql_query("UPDATE patientsrfid SET autorisation= 1 WHERE id='$val'") or die(mysql_error());
        }

        foreach($noCheck as $key => $val) //on update ensuite les id qui non checké
        {
            mysql_query("UPDATE patientsrfid SET autorisation= 0 WHERE id='$val'") or die(mysql_error());
        }
    }

    else //si il ni a eu aucune checkbox cochée

    {
        foreach($listId as $key => $val)
        {
            mysql_query("UPDATE patientsrfid SET autorisation= 0 WHERE id='$val'") or die(mysql_error());
        }
    }

}

?>
<form action="" method="post">
<?php

$test = mysql_query("SELECT * FROM patientsrfid") or die(mysql_error());

while ($resultat = mysql_fetch_array($test))
{
    if($resultat['autorisation'] == 1)
    {
        $checked="checked";
    }
    else
    {
        $checked="";
    }
}

```

?>

```
<input type="checkbox" id="chk_<?=$val['autorisation'];?>" name="CBTest[]" value="<?=$resultat['id'];?>"
<?=$checked;?> />
<input type="hidden" name="id[]" value="<?=$resultat['id'];?>" />
<label for="chk_<?=$resultat['id'];?>"><?=$resultat['nom'];?></label><br/>

<?php }?>

<input type="submit" value="Update" />
</form>
</body>
</html>
```

La page controle



Elle donne la liste des utilisateurs et permet de sélectionner un badge

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Contrôle de sortie</title>
</head>
<body>
<div align="center">
<h1>Liste des sorties</h1><hr>
</div>
<?php
include("config.php");
$query = "SELECT * FROM patientsrfid ORDER BY id";
$result = mysql_query($query) or die("Erreur de la table SQL");
if ($row = mysql_fetch_array($result)){
echo "<table border = '1'>";
echo "<tr> ";
echo "<td><b>Numero</b></td> ";
echo "<td><b>Nom</b></td> ";
echo "<td><b>Prenom</b></td> ";
echo "<td><b>RFID</b></td> ";
echo "<td><b>Autorisation</b></td> ";
```

```

echo "</tr> ";
do {
echo "<tr> ";
echo "<td>".$row["nombre"]."</td> ";
echo "<td>".$row["nom"]."</td> ";
echo "<td>".$row["prenom"]."</td> ";
echo "<td>".$row["rfid"]."</td> ";
echo "<td>".$row["autorisation"]."</td> ";
echo "<td><a href=test.php?rfid=".$row["rfid"].">Voir</a></td>";
echo "</tr> ";

} while ($row = mysql_fetch_array($result));
echo "</table>";

echo "</table> ";

} else {

echo "'Pas de donnée dans la table'";

}

?>
</body>
</html>

```

Elle fait appel à une page test.php qui ressort tout les passage d'un badge, en cliquant sur voir on a les heures de passage



id	rfid	heure
1	650FBD1D	26/06/2015 01:57:27
2	650FBD1D	26/06/2015 02:04:00
6	650FBD1D	27/06/2015 01:06:27
10	650FBD1D	27/06/2015 01:37:54

Les pages ajout et effacement permettent de gérer la base de données



**Ajouter des badges**

Numéro

Nom

Prenom

RFID

[Effacer un badge](#)



**Effacer un enregistrement**

Nombre à effacer  
1

[Ajouter](#)

## conf.php

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Contrôle de sortie</title>
</head>
<body>
<div align="center">
<h1>Ajouter des badges</h1><hr>
</div>
<form name="ajout" id="ajout" method="post" action="ajout.php" >
  <div align="center">Numéro
  <br/><input name="nombre" /><br/>
  Nom
  <br/><input name="nom" /><br/>
  Prenom
  <br/><input name="prenom" /><br/>
  RFID
  <br/><input name="rfid" /><br/>
  <br/>
  <input type="submit" name="Ajouter" value="Ajouter"/>
  </div>
</form>
<div align="center"><br>

  <a href=efface.php>Effacer un badge<a>
  <br>

</body>
</html>
```

## ajout.php

```
<?php
$nombre=$_POST['nombre'];
$nom=$_POST['nom'];
$prenom=$_POST['prenom'];
$rfid=$_POST['rfid'];

include("config.php");

$query="INSERT INTO patientsrfid (nombre, nom, prenom, rfid)
VALUES('$nombre','$nom','$prenom','$rfid')";
$result=mysql_query($query) OR DIE("Erreur ecriture base de donnée".mysql_error());
mysql_close();
?>
<div align="center">Nouvel enregistrement<br/>
  <br>
  <a href=conf.php>Ajout</a>
  <br>
  <a href=efface.php>Effacer</a></div>
```

## efface.php

```
<HTML>
<HEAD>
<TITLE>Effacer</TITLE>
</HEAD>
<BODY>
<div align="center">
```

```

<h1>Effacer un enregistrement</h1>
<br>

<?php
include("config.php");
echo '<FORM METHOD="POST" ACTION="efface1.php">Nombre a effecer<br>';

//Commande SQL
$query="Select nombre From patientsrfid Order By id";
$result=mysql_query($query);

echo '<select name="nombre">';

while ($row=mysql_fetch_array($result))
{echo '<option>'. $row["nombre"];}
mysql_free_result($result)
?>

</select>
<br>
<INPUT TYPE="SUBMIT" value="EFFACER">
<br>
<br>
<a href=conf.php>Ajouter<a>;
</FORM>
</div>

</BODY>
</HTML>

```

## efface1.php

```

<HTML>
<HEAD>
<TITLE>Effacer un badge</TITLE>
</HEAD>
<BODY>
<?php
include("config.php");

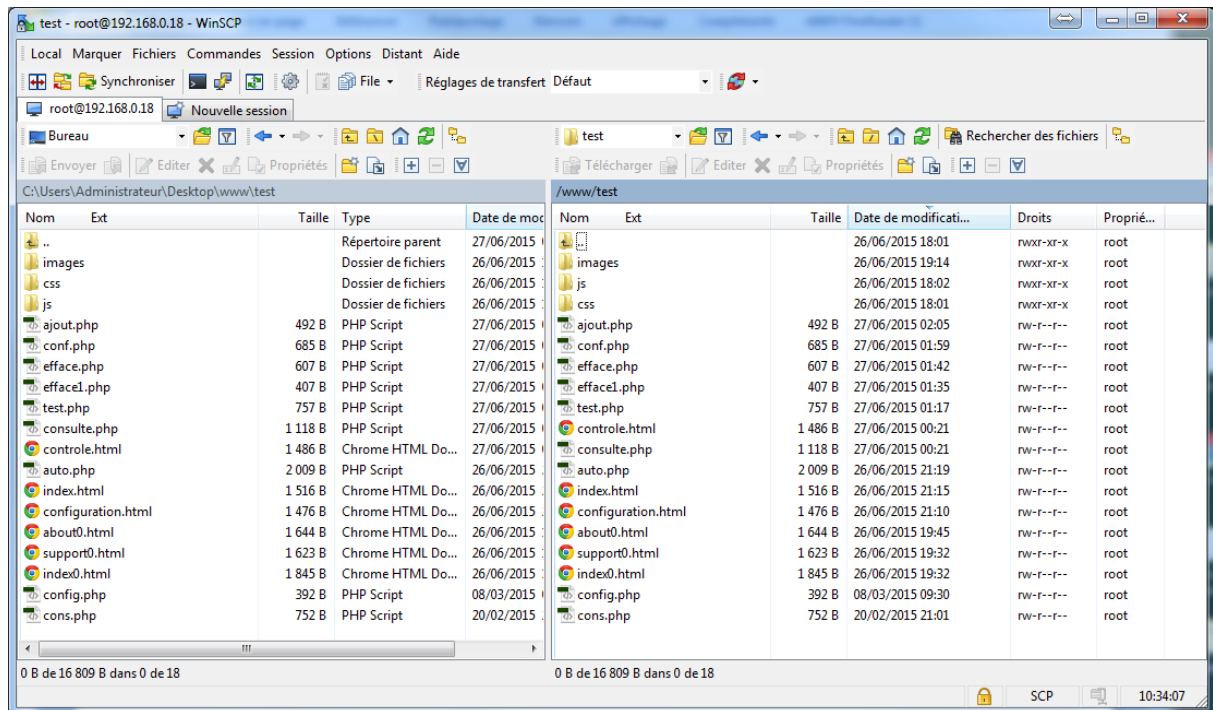
$query="Delete From patientsrfid Where nombre = '". $_POST['nombre']. "'";
mysql_query($query);
?>

<h1><div align="center">Effacer un enregistrement</div></h1>
<div align="center">Votre enregistrement a bien été effacé</div>
<div align="center"><a href="conf.php">Retour</a></div>

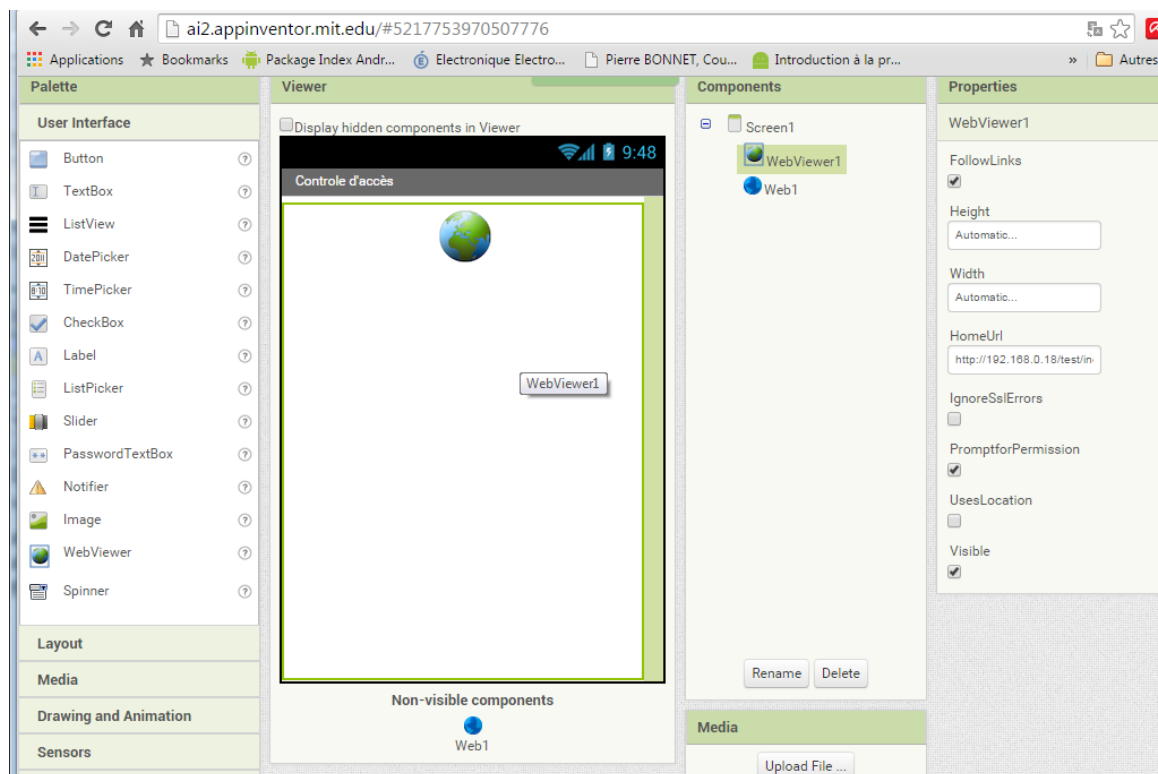
</BODY>
</HTML>

```

Ces fichiers sont à placer dans le rep www de l'arduino avec WinSCP



Réalisation de l'application Android sous App Inventor créer une application avec uniquement le lien vers le site web



Pour avoir une application entièrement autonome on peu repasser le Yun en point d'accès et connecter son smart phone à ce wifi

Le bouton reset du WiFi est situé à côté du connecteur ASB-A. Quand vous pressez ce bouton, la led WLAN se mettra à clignoter.



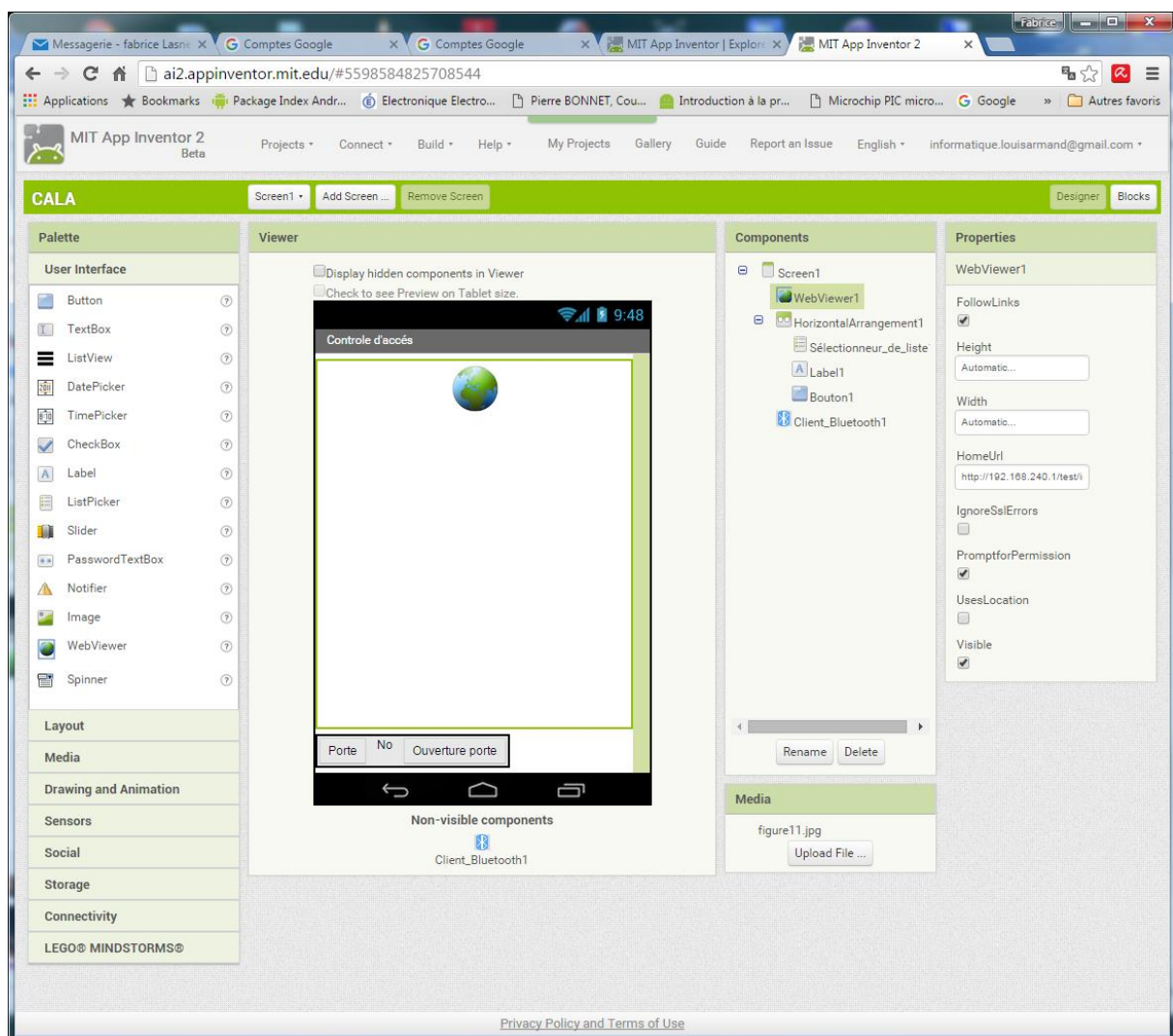
Si vous changez de réseau et ne pouvez plus vous connecter sans fil au Yún par le biais de son interface web, vous pouvez réinitialiser la configuration réseau du Yún en appuyant sur le bouton reset du WiFi pendant plus de 5 secondes, mais moins de 30.

Pour configurer l'application Bluetooth à partir d'Appinventor

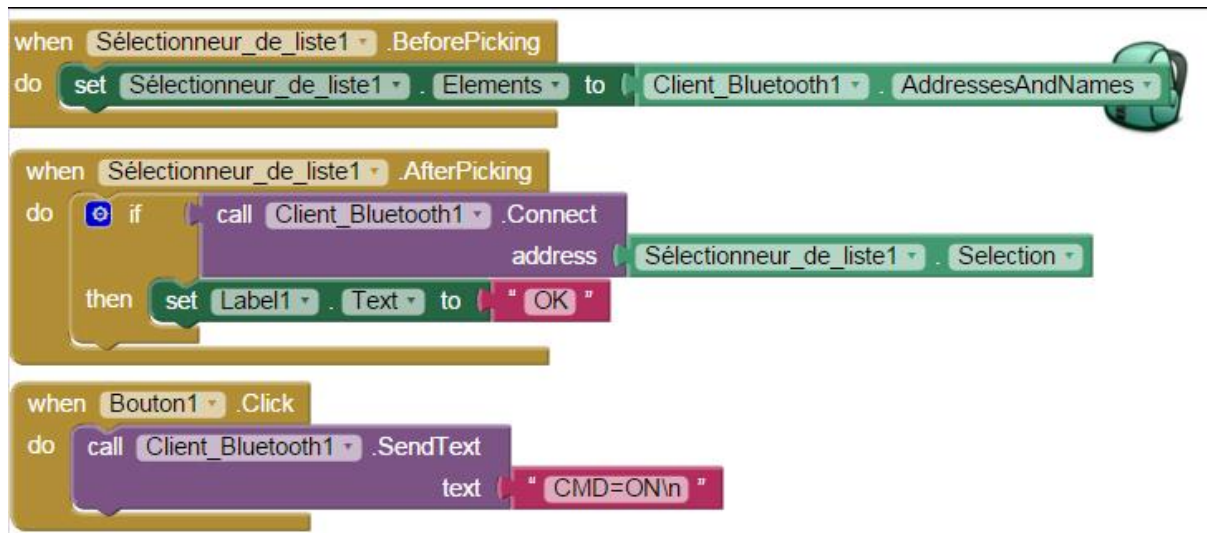
Connectez-vous sur <http://appinventor.mit.edu/explore/> avec un compte Google

Créez un projet avec un WebViewer à l'adresse IP de votre site Arduino ici <http://192.168.240.1/test/index.html>

Ajouter un sélecteur de liste listpicker, un label et un bouton et dans la fenêtre connectivity un client Bluetooth



Dans la partie Blocks rentrez le code suivant :



Le sélectionneur de liste vous donne les appareil Bluetooth a proximité pour choisir l'appareil a connecter ici le 20 :14 :04 :24 :30 :12 HC-06 une fois l'appareil connecté la LED du HC-06 passe en rouge fixe et l'appareil est connecté un appui sur le BP envoie la commande CMD=ON sur la liaison série on utilise les pates 10 et 11 pour la communication avec le HC-06

```
// Pins used to connect the Bluetooth module HC06
```

```
SoftwareSerial mySerial(10, 11); // RX, TX
```

```
mySerial.begin(9600);           // Rajout d'un port com pour lecteur bluetooth (pas serial1 à cause du bridge)
```

a la reception de caractères sur la liaison serie issue du Bluetooth

```
// if (mySerial.available() > 0) {serialA = mySerial.read();Serial.println(serialA);}
```

```
while(mySerial.available())
```

```
{
```

```
  delay(3);
```

```
  char c = mySerial.read();
```

```
  messageRecu += c;
```

```
}
```

```
if (messageRecu.length() >0)
```

```
{
```

```
  Serial.println(messageRecu);
```

```
  if (messageRecu == "CMD RED=ON\\n")
```

```
{digitalWrite(13, HIGH);  
  
delay(3000);  
  
Serial.println("led on");  
  
}
```

La porte s'ouvre (LED verte allumé) si le message reçu est CMD RED=ON\n.

Il ne reste plus qu'à faire la découpe du portail à partir des fichier SolidWorks, imprimer le pignon avec une imprimante 3D réaliser le câblage du moteur et des fins de course pour rendre l'ensemble fonctionnel.