myDomotique

Manuel d'installation

(version manuelle - Buster)

1. Installation de raspbian

Reportez vous aux tutos du net

Formatez la carte SD en FAT32 puis,

sous windows, utilisez Win32DiskImager ou Etcher

sous linux avec Etcher ou la commande :

sudo dd bs=1M if=votreImage.img of=/dev/sdc

Avec /dev/sdc le périph de la carte ; attention ce n'est pas sdc1

Pour information :

Sauvegarde:

sudo dd if=/dev/sdc |gzip -9 > ./imageRasp.img.gz

Restauration :

gunzip ./iamgeRasp.img.gz | sudo dd of=/dev/sdc

1. Premier démarrage

Le ssh n'étant activé par défaut il faut obligatoirement démarrer avec un écran un clavier.

Ensuite, faites Préférences/Configuration du Raspberry Pi,

onglet ***Interfaces***, **validez SSH, I2C, Série, GPIO**

onglet ***Localisation***, Réglez les quatre paramètres.

ssh en root

1. Ouvrez le fichier de configuration : sudo nano /etc/ssh/sshd\_config.
2. Trouvez cette ligne : #PermitRootLogin prohibit-password.
3. Remplacez la par celle-ci : PermitRootLogin yes. ...
4. Mise à jour

Après avoir booté,

sudo apt-get update

sudo apt-get upgrade

sudo apt-get install -y dos2unix

Rmq : L'upgrade est long (env 30mn)

1. Installation du reseau

Deux possibilités :

1. Avec un routeur en point d'accès Wifi. Dans ce cas nous n'aurons pas besoin ni du module RTC ni du service NTP. Après le §5, allez directemnt au §9.
2. Sans routeur : Dans ce cas, la raspberry doit server de point d'accès Wifi et de serveur de temps (NTP) pour les cameras.

* Le réseau est maintenant géré par le package : **ifupdown**
* Auquel il faut ajouter le package **resolv**.**conf**
* Il existe des scripts de (re)configuration (**ifplugd**, **guessnet**, **ifsceme**, …)
* Le package **wpasupplicant** est nécessaire pour utiliser WPA/WPA2 sur le wifi
* Pour les tests réseau le package **ifupdown-extra** dispose entre autres du scripts **network-test** qui donne toutes les info sur les connexions

1. Installation réseau (Cas 1 : avec routeur)

**Editer /etc/dhcpcd.conf**

sudo geany /etc/dhcpcd.conf &

Il doit contenir :

interface wlan1

static ip\_address=192.168.2.1/24

nohook wpa\_supplicant # on n'utilise pas wpa\_supplicant pour wlan1

############################

# Example static IP configuration:

interface wlan0

static ip\_address=192.168.0.100/24

static ip6\_address=fe80::8218:dbfb:ec51:1f8b/64

static routers=192.168.0.254

static domain\_name\_servers=192.168.0.254 8.8.8.8 fd51:42f8:caae:d92e::1

interface eth0

static ip\_address=192.168.0.99/24

#static ip6\_address=fd51:42f8:caae:d92e::ff/64

static routers=192.168.0.254

static domain\_name\_servers=192.168.0.254 8.8.8.8 fd51:42f8:caae:d92e::1

**Editer /etc/wpa\_supplicant/wpa\_supplicant.conf**

network = {

ssid= " "

psk= " "

key\_mngt=WPA-PSK

}

1. Serveur web

sudo apt-get install -y apache2

Changement des droits dossier apache :

sudo chown pi.www-data /var/www/html -R

sudo chmod 770 /var/www/html -R

Puis recopier le site dans "/var/www/html"

1. PHP

sudo apt-get install -y php

1. Mysql ou Mariadb

sudo apt-get install -y mariadb-client mariadb-server php-mysql

sudo apt-get install -y default-libmysqlclient-dev

sudo apt-get install -y libmariadbclient-dev-compat

sudo apt autoremove

Pour acceder à mysql en root

sudo mysql -u root

... connexion

use mysql;

update user set plugin='' where user='root';

flush privileges;

exit;

puis :

mysql --user=root mysql

mysql> …

CREATE USER 'gilles'@'localhost' IDENTIFIED BY 'gilles';

GRANT ALL PRIVILEGES ON \*.\* TO 'gilles'@'localhost' WITH GRANT OPTION;

CREATE USER 'gilles'@'%' IDENTIFIED BY 'gilles';

GRANT ALL PRIVILEGES ON \*.\* TO 'gilles'@'%' WITH GRANT OPTION;

CREATE USER 'admin'@'localhost' IDENTIFIED BY 'admin';

GRANT RELOAD,PROCESS ON \*.\* TO 'admin'@'localhost';

SET PASSWORD FOR 'root'@'localhost'=PASSWORD('');

La dernière ligne permet de se connecter avec login root sans mdp pour l'installation de phpmyadmin.

1. phpmyadmin :
   1. installation

sudo apt-get install -y phpmyadmin

* Choisissez "apache2" à configurer,
* répondez oui à la proposition de configuration,
* puis entrez les informations demandées :
  + Nom de la base : phpmyadmin
  + Nom administrateur : root (et non pas phpmyadmin)
  + Mot de passe : root.

Au besoin consultez :

<https://doc.ubuntu-fr.org/phpmyadmin>

En cas de non fonctionnement relancez la configuration avec :

sudo dpkg-reconfigure phpmyadmin

Remettez le mot de passe de root :

mysql -u root

puis

SET PASSWORD FOR 'root'@'localhost'=PASSWORD('root');

* 1. Importez la base "Surveillance"

connectez vous sur mysql avec phpmyadmin, créez une base vide nommée "Surveillance" avec un interclassement "utf8\_bin",

puis importez fichier "Surveillance1.sql" disponible dans le répertoire "Instal-Rasp"

ou bien avec le client mysql (mdp = root) :

mysql -u root -p Surveillance < Surveillance1.sql

1. Librairie curl

sudo apt-get install -y libcurl4-openssl-dev

**IMPORTANT** *(bug = dh key too small) :*

sudo geany /etc/ssl/openssl.cnf

Rechercher (à la fin du fichier)

CipherString=DEFAULT@SECLEVEL=2

Le commenter

#CipherString=DEFAULT@SECLEVEL=2

1. Serveur FTP (pure-ftp)
   1. Installation

sudo apt-get install -y pure-ftpd pure-ftpd-common

sudo groupadd ftpgroup

sudo useradd -g ftpgroup -d /dev/null -s /etc ftpuser

puis,

sudo geany /etc/pure-ftpd/conf/PAMAuthentication &

no

sudo geany /etc/pure-ftpd/conf/DontResolve &

yes

cd /etc/pure-ftpd/auth/

sudo ln -s ../conf/PureDB 50puredb

* 1. Création des utilisateurs

sudo useradd – g -u 21 -s /bin/false -d /home/pi/Surveillance gilles

sudo pure-pw useradd ***gilles*** -u ftpuser -g ftpgroup -d /home/pi/Surveillance

-> mdp de gilles : gilles

sudo pure-pw useradd ***pi*** -u ftpuser -g ftpgroup -d /home/pi

-> mdp de pi : *<celui que vous avez mis au début de l'install>*

sudo pure-pw useradd ***www-data*** -u ftpuser -g ftpgroup -d /var/www/html

-> mdp de gilles : www-data

**Ou bien plus simple et préfarable pour cette appli :**

*(on place directement les utilisateurs en pi)*

sudo pure-pw useradd ***gilles*** -u pi -g pi -d /home/pi/Surveillance

sudo pure-pw useradd ***www-data*** -u pi -g pi -d /var/www/html

sudo pure-pw useradd ***pi*** -u pi -g pi -d /home/pi

Ajoutons pi au groupe ftpgroup

sudo usermod -a -G ftpgroup pi

sudo usermod -a -G ftpgroup www-data

Création de la base de pure-ftp

sudo pure-pw mkdb

Puis, on relance :

sudo /etc/init.d/pure-ftpd restart

Pour verifier ou chg mdp:

sudo pure-pw show gilles

sudo pure-pw passwd gilles -m

Suppression d'un utilisateur virtuel :

sudo pure-pw userdel UTILISATEUR –m

–m permet de ne pas reconstruire la base

1. wiringPi

A partir de raspbian 2019-09 :

sudo apt-get install -y wiringpi

Sur version plus ancienne :

sudo apt-get install -y git git-core

git clone git://git.drogon.net/wiringPi

cd wiringPi

sudo su

./build

Rmq : Le clonage est long …

Dans le répertoire de wiringPi, Testez :

|  |
| --- |
| root@raspberrypi:/home/pi/wiringPi/wiringPi# **gpio readall** |

1. OpenCV

sudo apt-get install -y cmake gcc g++ libx11-dev libxt-dev libxext-dev libgraphicsmagick1-dev libopencv-dev

1. tinyXML

Sert pour lire la config du système placée dans le repertoire Domotik.

sudo apt-get install -y libtinyxml2-dev [libtinyxml-dev]

1. Mosquitto

Elle sert pour la communication Wifi entre les capteurs de température et le relai de pilotage de la VMC.

**Principe :**

* Les capteurs de température publient un message contenant la température toutes les N secondes avec un TOPIC= DOMO/CaptXY (XY= code du capteur dans la table)
* L'application publie un message de TOPIC = DOMO/relai avec une valeur à 0 ou 1 pour piloter la VMC

**Installation :**

sudo apt-get install -y mosquitto mosquitto-clients

sudo apt-get install -y mosquitto libmosquitto-dev

**Pour tester le fonctionnement :**

Sur une console tapez :

mosquitto\_sub -h localhost -t DOMO/+

Sur une seconde console :

mosquitto\_pub -h localhost -t DOMO/relai -m "1"

ex : -t DOMO/Capt21 -m "12.34"

1. Installation réseau (Cas 2 : sans routeur)

Installons le **hostapd** pour le point d'accès wifi, **dnsmasq** comme serveur DHCP et DNS.

sudo apt-get install -y dnsmasq hostapd

Stoppez les services et rebootez

sudo systemctl stop dnsmasq

sudo systemctl stop hostapd

sudo reboot

* 1. Partie "Wifi"

**Les drivers :**

Pour wlan0 il est en natif dans le noyau. Pour votre clé wifi supplémentaire, la commande lsusb vous donne des informations.

Si vous voyez RTL8188CUS dans la ligne de votre clé et qu'elle ne fonctionne pas, reportez vous au net pour installer le driver correspondant. On les trouve souvent sur github ou sur <http://www.fars-robotics.net/>.

Choisir la version en fonction de la version de noyau (uname-a)

Exemple pour Ralink MT7601U

* Récupérer le fichier MT7601U sur github
* le copier dans /lib/firmware
* ajouter mt7601u dans /etc/module
  1. Configuration de l'IP statique wlan1

Editer **/etc/dhcpcd.conf**

sudo geany /etc/dhcpcd.conf &

Ajoutez :

interface wlan1

static ip\_address=192.168.2.1/24

nohook wpa\_supplicant # on n'utilise pas wpa\_supplicant pour wlan1

* 1. Configuration du serveur DHCP

Editez **/etc/dnsmasq.conf**

sudo geany /etc/dnsmasq.conf &

Ajoutez :

interface=wlan1

dhcp-range=192.168.2.20,192.168.2.40,24h

#EXEMPLE ipfixe pour les cameras

dhcp-host=28:f3:66:6c:db:b7,WansWiew2,192.168.2.52

* 1. Configuration du point d'accès

Editez ou créez **/etc/hostapd/hosaptd.conf**

sudo geany /etc/hostapd/hosapd.conf &

Ajoutez :

interface=wlan1

#driver=rtl871xdrv

driver=nl80211

#

ssid=DomoSSID

#mode g=2.4G, plus lent mais plus grande portée

hw\_mode=g

channel=7

#ieee80211n=1

wmm\_enabled=0

macaddr\_acl=0

auth\_algs=1

ignore\_broadcast\_ssid=0

wpa=2

wpa\_key\_mgmt=WPA-PSK

wpa\_passphrase=***DomoMdP***

wpa\_pairwise=TKIP

rsn\_pairwise=CCMP

Indiquons au système de ne pas utiliser l'interface wlan1 en recherche de réseau, éditez à nouveau **/etc/dhcpcd.conf** et ajoutez la ligne suivante à la fin :

#denyinterfaces wlan1

**Lancez les services**

sudo service hostapd start

sudo service dnsmasq start

Pour lancer hostapd en démon au démarrage, éditez :

sudo geany /etc/default/hostapd &

Remplacez la ligne #DAEMON\_CONF="" par le chemin au fichier de conf

DAEMON\_CONF= "/etc/hostapd/hostapd.conf"

**Après avoir rebooté**, vous devriez voir votre réseau :

ssid : …

MdP = …

La commande suivante permet de lister les réseaux wifi :

iwlist s

* 1. Configuration (éventuelle) du routage et masquerade

Rappel : l'IP Masquerading permet à vos clients dhcp de se connecter à internet

Editez **/etc/sysctl.conf** et décommentez cette ligne :

net.ipv4.ip\_forward=1

Ajoutez la masquerade pour rediriger le traffique sur eth0:

sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

Sauvez les règles iptables

sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"

Editez **/etc/rc.local** et ajoutez juste avant "exit 0" la récupération des règles au boot

/sbin/iptables-restore < /etc/iptables.ipv4.nat

Attention rc.local n'est plus lancé automatiquement, donc :

Editez **/etc/bash.bashrc** et ajoutez à la fin

/etc/rc.local

Rebootez

1. Module de temps RTC (ds3231)

**Cette partie est nécessaire uniquement s'il n'y a pas de routeur (sinon allez au § 9)**

* 1. Installation

sudo apt-get install -y python-smbus i2c-tools

Vérifiez le fonctionnement en tapant :

sudo i2cdetect -y 1

Vous devez avoir une entrée en 68 *(soit 68 soit UU)*

Si ça ne fonctionne pas,avez-vous validé l'I2C ? Puis essayez :

sudo modprobe i2c-dev

sudo modprobe i2c-bcm2708

sudo modprobe rtc-ds1307

Ensuite,

sudo geany /etc/modules &

Ajoutez les lignes :

i2c-dev

i2c-bcm2708

rtc-ds1307

Notifiez au système la présence du composant

sudo su

echo ds3231 0x68 > /sys/class/i2c-adapter/i2c-1/new\_device

exit

* 1. Utilisation avec systemd

Editez le fichier **/boot/config.txt**

sudo geany /boot/config.txt &

précisons qu’on veut utiliser le bus i2c et un module à base de DS3231 en ajoutant :

dtparam=i2c\_arm=on

dtoverlay=i2c-rtc,ds3231

Notez que ds3231 peut être remplacé par pcf8523 ou ds1307 ...

Créer ou récupérer le fichier **/etc/systemd/system/rtc-init.service**

sudo geany /etc/systemd/system/rtc-init.service &

Contenu :

[Unit]

Description=RTC Clock Setup and Time Sync

Before=cron.service

[Service]

Type=oneshot

ExecStart=/usr/lib/systemd/scripts/rtc-setup.sh

[Install]

WantedBy=multi-user.target

Ceci défini un nouveau service qui ne s’exécute qu’une fois et est lié à un script qu’il faut également écrire. C’est le fichier **/usr/lib/systemd/scripts/rtc-setup** :

sudo geany /usr/lib/systemd/scripts/rtc-setup &

Contenu :

#!/bin/sh

hwclock -s --utc

echo "System Time synced with RTC Time"

Ne pas oublier de le rendre exécutable avec :

sudo chmod a+x /usr/lib/systemd/scripts/rtc-setup

Et enfin, nous pouvons activer le service en question avec

sudo systemctl enable rtc-init

* 1. Configuration

Configuration du système pour utiliser automatiquement le module RTC

|  |
| --- |
| sudo geany /etc/rc.local & |

Ajoutez à ce fichier les deux lignes suivantes :

|  |  |
| --- | --- |
|  | /bin/echo ds3231 0x68 > /sys/class/i2c-adapter/i2c-1/new\_device  /sbin/hwclock -s |

Désactiver et enlever le service [**fake-hwclock**](https://packages.debian.org/unstable/main/fake-hwclock) qui tente de reproduire le fonctionnement d’une horloge temps réel quand on en dispose pas

sudo update-rc.d fake-hwclock disable

sudo apt-get -y remove fake-hwclock

sudo update-rc.d -f fake-hwclock remove

Maintenant démarrons l'horloge matérielle d'origine

sudo geany /lib/udev/hwclock-set &

Commentez les lignes

#if [ -e /run/systemd/system ] ; then

# exit 0

#fi

Après un reboot, la commande

sudo hwclock -D -r

devrait vous donner :

Une image contenant texte

Description générée automatiquement

L'horloge du RTC n'étant pas forcément à l'heure ...

Au besoin, définissez manuellement la date du système manuellement

sudo date -s "2019-05-25 18:50:00"

Enregistrez la date dans le RTC

sudo hwclock -w

puis pour relire :

sudo hwclock -r

1. Serveur NTP

**Cette partie est nécessaire uniquement s'il n'y a pas de routeur (sinon allez au § 9)**

sudo apt-get install -y ntp

sudo geany /etc/ntp.conf &

ajoutez au paragraphe restrict (env ligne 47) :

restrict 192.168.2.0 mask 255.225.225.0

broadcast 192.168.2.255

1. Modification rc.local

Verifiez / Recopiez le fichier /etc/rc.local

Ajoutez :

/bin/echo ds3231 0x68 | sudo /usr/bin/tee /sys/class/i2c-adapter/i2c-1/new\_device

Attention le /etc/rc.local n'étant plus appelé automatiquement, pensez à le rendre exécutable et à modifier le crontab (voir paragraphe sur cron ci-après)

1. cron

Editez le crontab et reproduisez les lignes ci-dessous :

sudo crontab -e

# m h dom mon dow command

# La ligne suivante ne sert qu'avec motion

#\* \* \* \* \* /home/pi/DomoScript/testMotion

#0 9 \* \* 5 reboot

0 11 \* \* 4 /home/pi/DomoScript/cleanAppli

\* \* \* \* \* /home/pi/DomoScript/testAppli

@reboot /etc/rc.local

@reboot /home/pi/DomoScript/testAppli

Pour info :

* reboot le vendredi
* cleanAppli le jeudi
* testAppli toutes les minutes
* rc.local à chaque reboot

1. Application

*Copiez le répertoire Instal-Rasp dans /home/pi, puis :*

cd Instal-Rasp

dos2unix install\*

dos2unix conf/\*

dos2unix DomoScript/\*

chmod a+x install\*.sh

./install-Appli.sh

**IMPORTANT**

**Pensez à faire dos2unix sur les fichiers du répertoire**

**- Domotik**

**- DomoCayenne**

**- DomoScript**

**- DomoDossier**

**Rendre exécutable tous les fichiers batch du répertoire /home/pi/DomoScript**

**Effacer les pipes 'Domotik/ent' et 'Domotik/sort'**

1. Réglages (Cas Rasp en point d'accès Wifi)

N'EST PLUS NECESSAIRE !

Repérez l’adresse MAC wifi de votre ESP. Les exemples de commandes suivantes vous permettront de la trouver :

nmap-sP 192.168.2.0/24

arp -a 192.168.2.62

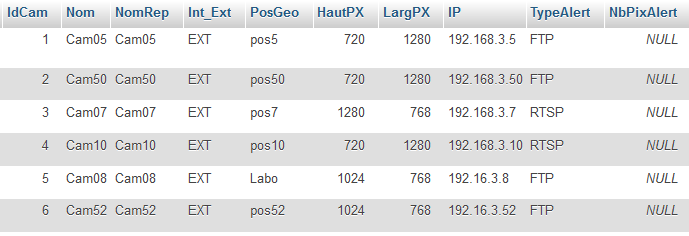
Reportez-la dans le fichier "/etc/dhcpcd.conf" comme indiqué à la fin.

Ajoutez là dans le fichier de conf "~/Domotik/configAppli.txt"

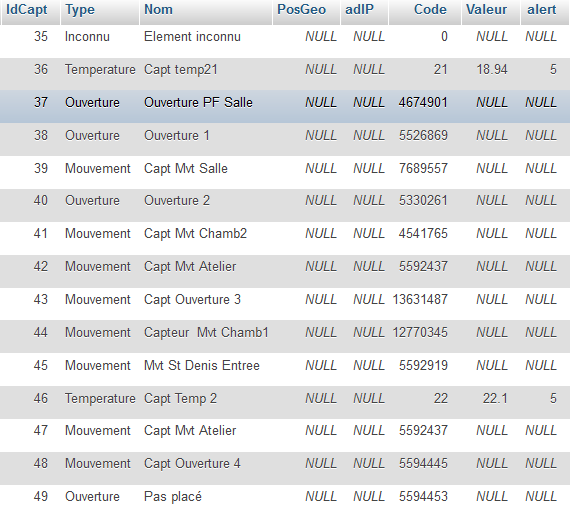
Dans le fichier "~/Domotik/configAppli.txt" pensez à modifier au besoin le numéro de la carte SIM.

**Exemples de tables :**

**Table CAMERA**



**Table CAPTEUR**

Exemple (voir tableau plus loin) :

**Exemple de fichier /etc/dnsmasq.conf**

interface=wlan1

dhcp-range=192.168.2.20,192.168.2.40,24h

#EXEMPLE ipfixe pour les cameras

dhcp-host=28:f3:66:6c:db:b7,WansW52,192.168.2.52

dhcp-host=00:0C:43:B0:19:73,Wanscam49,192.168.2.49

dhcp-host=00:7E:56:F3:51:78,SZSinoN50,192.168.2.50

dhcp-host=a0:9d:c1:1d:65:0d,SZSinoN\_8,192.168.2.8

dhcp-host=cc:50:e3:c7:36:c4,ESP\_C736C4,192.168.2.62

1. Capteur de temperature par Arduino

* Récupérer le fichier « .ino »,
* Vérifier le numéro de capteur.
* Téléverser

1. Module ESP (température et pilotage VMC)

Installation du **driver pour la carte USB/Série** FTD1232 :

<https://www.ftdichip.com/FTDrivers.htm>

ou fichier dans repertoire du projet

Installation du **logiciel Arduino** :

<https://www.arduino-france.com/tutoriels/comment-programmer-lesp8266-avec-arduino/>

* Récupérez le fichier « .ino », vérifiez
  + les adresses IP,
  + le numéro de capteur
  + les noms de topic Mosquitto
  + L'adresse de la carte ESP dans le fichier **/home/pi/Domotik/confAppli.xml**  
    Théoriquement en IP fixe = 192.168.0.ZZ avec 51 <= ZZ <=59

1. Le port COM doit être branché avant de faire la coupure de courant qui fait le reset de la carte.
2. Positionnez la broche GPIO0 à 0 (fil volant)
3. Alimenter (ou faire coupure)
4. Téléversez le programme sur l’ESP
5. Retirez la masse de la broche GPIO0 avant de couper l’alimentation
6. Cayenne sur Raspberry
   1. Pour développer directement avec le E/S du GPIO :

* Sur la rasp, aller sur le site "cayenne.mydevice.com"
* se connecter et ajouter un device raspberry
* Suivre la démarche (installation automatique)
* Ajouter les devices (sensors) en indiquant les numéro de pin
* Les identifiants de la raspberry sont dans le fichier "/etc/mydevice/AppSettings.ini"
  1. Pour ajouter des widgets sur Cayenne :
* Récupérer sur GitHub ***Cayenne-MQTT-CPP*** et l'installer par : sudo make
* Adapter le code suivant exemple

**Sur Cayenne :**

* Créer un nouveau **device type "Arduino"**
* Récupérer les ***identifiants*** proposés, les recopier dans le code
* Connecter le code, les widgets en entrée s'ajoutent automatiquement
* Ajouter les widgets (uniquement booléens) de sortie

1. Configuration des cameras

autoriser rtsp

serveur de temps : 192.168.2.1 ou site si routeur

dimension image : 1024 x 768 ou autres …

LAN en DHCP

Wifi : static 192.168.2.XX avec 2 <= XX <= 49 ou 192.168.0.YY avec 70 <= YY <= 70

Eventuellement faire détecter le ssid de votre réseau

Associer si besoin l’adresse MAC de la camera à une IP dans /etc/dnsmasq.conf

1. Capteur de mouvements

Ces capteurs envoient une trame sur la fréquence de 433MHz. Cette trame est composée d’un entier codé sur 24 bits (3 octets). Ce code peut être modifié grâce aux cavaliers.

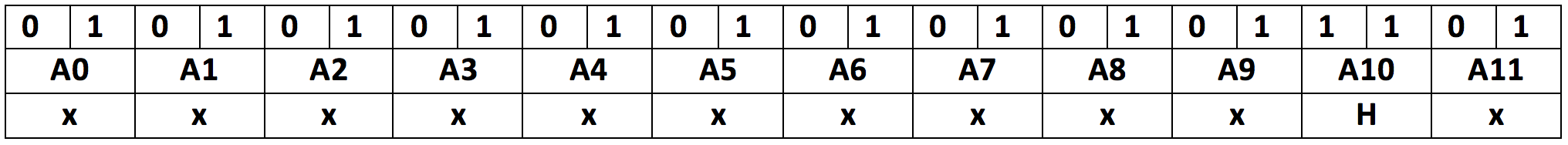
|  |  |
| --- | --- |
| Chaque cavalier modifie 2 bits, on a donc 3 possibilités :   * 01 (sans cavalier) * 00 (Low) * 11 (High)   Pour les 12 adresses de A0 à A12 | 0: 00 00  1: 00 01  3: 00 11  4: 01 00  5: 01 01  7: 01 11  12=C: 11 00  12=D: 11 01  15=F: 11 15 |

NOTA : Nos capteurs de mouvement devront avoir une adresse qui commence par C2 ou C3

C2XXXX ou C3XXXX

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| image4 | Par exemple si l’on place le cavalier A10 en position H (High) nous obtiendrons le code : |



Ce qui nous donne en décimal: 5592413 = 0x55555D

1. Capteur d’ouverture

Principe identique aux capteurs de mouvement.

Une image contenant équipement électronique, circuit

Description générée automatiquement

Au besoin mettre J4 sur 1.5

Attention : La sérigraphie H/L peut être inversée !

NOTA : Nos capteurs d'ouverture devront avoir une adresse qui commence par C4 : C4XXXX

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1. Capteur de température

**Sur Arduino (communication en 433) :**

Le mot émis est composé de 3 octets :

* 11 bits de poids faible = temp = (int)(temp +56)\*10)
* 13 bits qui représente le Num de catpteur avec 16 < Num < 126

**Sur ESP8266 (communication en wifi):**

* Passage par le serveur MQTT installé sur la Raspberry

1. Résumé des fréquences 433

|  |  |
| --- | --- |
| Détection de mouvement | C2XXXX  C3XXXX |
| Capteur d'ouverture | C4XXXX |
| Capteur de température | FFXXXX  11 bits poids faible = Temp  13 bits poids fort = Num capteur avec  16 < Num < 126 |
| Relai Marche / Arrêt | M = A0000 = 655360  A = B0000 = 720896 |
| Porte clés | 7AXXXX  555503 |

1. Inventaire capteurs

|  |  |  |  |
| --- | --- | --- | --- |
| **Ouverture** | | **Mouvement** | |
| c45555 |  | c355d5 | Entrée St Denis |
| c45554 |  | c2dc29 | (fixe) Test |
|  |  |  | Salle Trivy |
|  |  |  | Chb2 Trivy |
|  |  |  | Atelier Trivy |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. GPIO de la Raspberry

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Carte proto** | **Carte 2020** |  |
| 2 | 5V | 5V |  |
| 9 | Masse | Masse |  |
| 13 | Data RX 433 |  |  |
| 15 | E= coupure alim | E= coupure alim |  |
| 16 |  |  |  |
| 17 | 3.3V | 3.3V |  |
| 18 | E=W8= 0 si courant secteur |  |  |
| 20 |  |  |  |
| 24 | S=W8=Data Emetteur 433 |  |  |
| 28 | S=W31=Pilotage alim Emet433 |  |  |

Initialisation de la broche du relai (Klaxon)

Fichier DomoScript/confGPIO.sh

#!/bin/bash

sudo sh -c 'echo "22" > /sys/class/gpio/export &'

sleep 1

#gpio en sortie gpio3 = BCM22

#/usr/bin/gpio export 3 out #gpio3

sudo sh -c 'echo out > /sys/class/gpio/gpio22/direction'

# Init à 1 pour repos

sudo sh -c 'echo 1 > /sys/class/gpio/gpio22/active\_low'

#initialisation ; position inverse mettre à 0

# gpio write <pi> <value>

#/usr/bin/gpio write 3 0

sudo sh -c 'echo 0 > /sys/class/gpio/gpio22/value'

1. Utilisation du port série interne de rapberry

Voir option 1 de : <https://www.framboise314.fr/le-port-serie-du-raspberry-pi-3-pas-simple/>

Dans ce cas le bluetooth ne fonctionnera plus !

1. **Valider le port série dans la configuration du raspberry (sudo raspi-config).**
2. **Editer /boot/cmdline.txt**

Il doit ressembler à

dwc\_otg.lpm\_enable=0 console=ttyAMA0,115200 kgdboc=ttyAMA0,115200 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait

Retirez toutes les références à ttyAMA0 qui est le nom du port série. Votre fichier doit maintenant ressembler à ceci :

dwc\_otg.lpm\_enable=0 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait

Supprimez la partie

console=serial, 115200

pour désactiver la console série au boot.

1. **Désactivez le module bluetooth dans le fichier /boot/config.txt**

Ajoutez à la fin :

dtoverlay=pi3-miniuart-bt

Pour ne pas utiliser UART1, Enlevez si besoin

**enable\_uart = 1**

[Optionnel] -> Configurations du port :

stty -F /dev/ttyAMA0 -a

sudo stty -F /dev/ttyAMA0 115200 cs8 -cstopb -onlcr -echo -echoe -echok -opost

1. **Config Rasp**

Allez dans configuration Raspberry, onglet "Interfaces"

Activez le port série et désactivez la console série

1. **Pour accéder au port série en mode "Utilisateur"**

sudo usermod -aG dialout pi

ou

sudo usermod -aG dialout $USER

1. **Attention**

Le port série peut s'appeler **/dev/ttyS0**

qui est dans le group tty (pas dialout), donc :

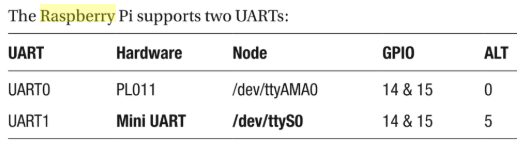
sudo usermod -aG tty pi

Pour connaitre celui qui est utilisé :

ll /dev/serial0

Voir page 178 de :

<https://books.google.fr/books?id=Ks10DwAAQBAJ&pg=PA177&lpg=PA177&dq=raspberry+xon/xoff&source=bl&ots=xpucvN-w0I&sig=ACfU3U1uukqjY-eh0kjTS6zlugeJKluJCw&hl=fr&sa=X&ved=2ahUKEwiV4r-xhojpAhUBx4UKHaMzA30Q6AEwBnoECAYQAQ#v=onepage&q=raspberry%20xon%2Fxoff&f=false>



1. **Rebooter !**

**Pb d'accès SSL :**

Dans /etc/ssl/openssl.cnf

Supprimer la dernière ligne :

#CipherString …

1. Choix des adresses IP (conf avec routeur)

**Raspberry :**

* eth0= 192.168.0.99
* wlan0=192.168.0.100

**Cameras :**

* 192.168.0.51 à 59

**ESP :**

* 192.168.0.61 à 69

**Les paragraphes suivants ne sont pas à faire !!!**

1. /\*\* Libserial \*\*/

sudo apt-get install -y libboost-all-dev python-sip-dev

sudo apt-get install -y libserial0 libserial-dev [libserial-doc]

Voir :

<http://libserial.sourceforge.net/tutorial.html#setting-the-flow-control-type>

**Sur anciennes distributions :**

Décompresser libserial-X-Y-Z.tar.gz (disponible dans le répertoire *Instal-Rasp*)

cd libserial-0.6.0rc2

./configure

make

sudo make install

1. Evénement de la base (pour info)

DROP EVENT IF EXISTS cleanAlerte;

DELIMITER $$

CREATE EVENT cleanAlerte

ON SCHEDULE EVERY 1 MONTH

STARTS '2021-09-01 00:00:00'

ON COMPLETION PRESERVE -- permet de conserver l'événement

-- après son exécution

DO BEGIN

DELETE FROM ALERTE

WHERE DateAlerte < (CURRENT\_DATE - INTERVAL 92 DAY);

END$$

DELIMITER ;

1. Fichier de conf "confAppli.xml"

<?xml version="1.0" encoding="utf-8"?>

<!-- Settings pour appli Domotik

Version du 29/05/2020

-->

<Domotik>

<!-- PAR NOEUDS -->

<Param>

<IP\_Interrupt>192.168.0.61</IP\_Interrupt>

<num\_carte>+33769813456</num\_carte>

<mdpSMS>3456</mdpSMS>

<portSerie>/dev/ttyAMA0</portSerie>

<cle3G>0</cle3G>

<emiss433>1</emiss433>

<klaxonFil>0</klaxonFil>

<alimSec>0</alimSec>

<cayenne>1</cayenne>

<mms\_APN>mmsfree</mms\_APN>

<mms\_Centre>mms.free.fr</mms\_Centre>

<mms\_CentreH>http://212.27.40.255</mms\_CentreH>

<mms\_CentreIP>212.27.40.225</mms\_CentreIP>

<mms\_Proxy>212.27.40.225</mms\_Proxy>

<mms\_Port>80</mms\_Port>

</Param>

<Bdd>

<hostBdd>localhost</hostBdd>

<base>Surveillance</base>

<login>gilles</login>

<mdp>gilles</mdp>

</Bdd>

<!-- PAR ATTRIBUTS -->

<Cayenne

numCaptTemp="24"

username="00f66980-8126-11e9-ace6-4345fcc6b81e"

password="43a7ae8aabc694f8cb51a526609c5354a8249f3e"

clientID="911804c0-4b87-11ea-b301-fd142d6c1e6c"

/>

<!--

St Denis : "911804c0-4b87-11ea-b301-fd142d6c1e6c"

Trivy-Nouveau : "bde75370-8250-11ea-883c-638d8ce4c23d"

Test : "d834fc40-646f-11ea-ba7c-716e7f5ba423"

-->

<Ftp

url="ftp://perso-ftp.orange.fr/Images/"

logMdp="gillesaucaigne@orange.fr:GillesA\*21"

/>

<Mqtt

hostname="localhost"

port="1883"

username="admin"

password="admin"

topicRECEP="DOMO/+"

topicEMISS="DOMO/relai"

/>

<!--

marche : freq qui actionne le relai 433

arret : freq qui arrete le relai 433

pinalim : pin qui alimente la carte d'em 433

(31 sur ancienne carte ; 3 sur nouvelle)

pintx : pin des datas de la carte d'em 433

(10 sur ancienne carte ; 0 sur nouvelle)

pinrelai : pin qui pilote le relai FILAIRE

(3 sur ancienne carte ; 13 sur nouvelle)

pinRF : pin de reception des datas de la carte Recep 433 (2)

pindefalim : pin qui détecte la coupure de courant(21)

-->

<Relai

marche="655360"

arret="720896"

pinalim="3"

pintx="0"

pinrelai="13"

pinRF="2"

pindefalim="21"

/>

</Domotik>

Table des matières

[myDomotique 1](#_Toc57206420)

[Manuel d'installation 1](#_Toc57206421)

[(version manuelle - Buster) 1](#_Toc57206422)

[1. Installation de raspbian 1](#_Toc57206423)

[2. Premier démarrage 1](#_Toc57206424)

[3. Mise à jour 1](#_Toc57206425)

[4. Installation du reseau 1](#_Toc57206426)

[5. Installation réseau (Cas 1 : avec routeur) 2](#_Toc57206427)

[6. Installation réseau (Cas 2 : sans routeur) 2](#_Toc57206428)

[6.1. Partie "Wifi" 2](#_Toc57206429)

[6.2. Configuration de l'IP statique wlan1 3](#_Toc57206430)

[6.3. Configuration du serveur DHCP 3](#_Toc57206431)

[6.4. Configuration du point d'accès 3](#_Toc57206432)

[6.5. Configuration (éventuelle) du routage et masquerade 4](#_Toc57206433)

[7. Module de temps RTC (ds3231) 4](#_Toc57206434)

[7.1. Installation 4](#_Toc57206435)

[7.2. Utilisation avec systemd 5](#_Toc57206436)

[7.3. Configuration 5](#_Toc57206437)

[8. Serveur NTP 6](#_Toc57206438)

[9. Serveur web 6](#_Toc57206439)

[10. PHP 6](#_Toc57206440)

[11. Mysql ou Mariadb 7](#_Toc57206441)

[12. phpmyadmin : 7](#_Toc57206442)

[12.1. installation 7](#_Toc57206443)

[12.2. Importez la base "Surveillance" 7](#_Toc57206444)

[13. Librairie curl 8](#_Toc57206445)

[14. Serveur FTP (pure-ftp) 8](#_Toc57206446)

[14.1. Installation 8](#_Toc57206447)

[14.2. Création des utilisateurs 8](#_Toc57206448)

[15. wiringPi 8](#_Toc57206449)

[16. OpenCV 9](#_Toc57206450)

[17. tinyXML 9](#_Toc57206451)

[18. Mosquitto 9](#_Toc57206452)

[19. Modification rc.local 9](#_Toc57206453)

[20. cron 9](#_Toc57206454)

[21. Application 10](#_Toc57206455)

[22. Réglages (Cas Rasp en point d'accès Wifi) 10](#_Toc57206456)

[23. Capteur de temperature par Arduino 12](#_Toc57206457)

[24. Module ESP (température et pilotage VMC) 12](#_Toc57206458)

[25. Cayenne sur Raspberry 12](#_Toc57206459)

[25.1. Pour développer directement avec le E/S du GPIO : 12](#_Toc57206460)

[25.2. Pour ajouter des widgets sur Cayenne : 12](#_Toc57206461)

[26. Configuration des cameras 13](#_Toc57206462)

[27. Capteur de mouvements 13](#_Toc57206463)

[28. Capteur d’ouverture 14](#_Toc57206464)

[29. Capteur de température 14](#_Toc57206465)

[30. Résumé des fréquences 433 14](#_Toc57206466)

[31. Inventaire capteurs 14](#_Toc57206467)

[32. GPIO de la Raspberry 15](#_Toc57206468)

[33. Utilisation du port série interne de rapberry 16](#_Toc57206469)

[34. Choix des adresses IP (conf avec routeur) 17](#_Toc57206470)

[35. /\*\* Libserial \*\*/ 17](#_Toc57206471)

[36. Fichier de conf "confAppli.xml" 18](#_Toc57206472)