



Prises, Interrupteurs, volets roulants par commande WEB et ou vocale.

Utile dans la vie de tous les jours, la domotique. Prises, Interrupteurs, volets roulants par commande WEB et ou vocale.

 Difficulty Medium

 Duration 5 hour(s)

 Categories Electronics

 Cost 50 EUR (€)

Contents

Introduction

Step 1 - Les prises électriques

Step 2 - Emetteur 433Mhz et carte Ethernet

Step 3 - Les commandes en page WEB

Step 4 - Les interrupteurs et les commandes de volets roulants

Step 5 -

Comments

Introduction

Il permet de commander à distance des fonctions électriques de la maison comme : • Des prises • Des lumières • Des Volets roulants Il est composé de trois parties : • La partie commande basée sur une page web. J'ai aussi fait une commande vocale avec un Raspberry PI (<https://www.blog-de-michel.fr/raspberry-pi-domotique/>) • La partie émission des commandes basée sur un Arduino, carte Ethernet et émetteur 433Mhz.

• La partie réception (prises, interrupteurs et modules réceptions volets roulants).

Materials

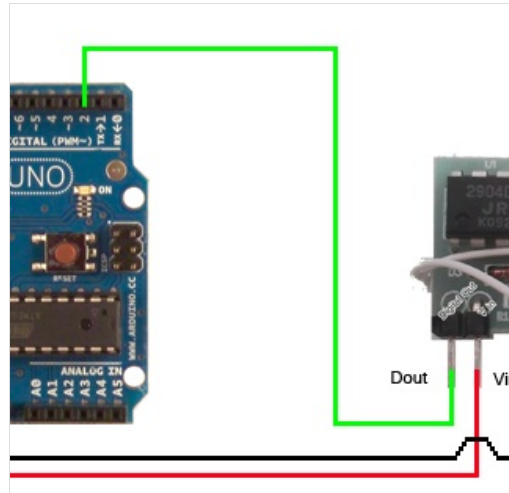
Raspberry PI Arduino pro mini 1 ou 2 ... Carte Ethernet Récepteur émetteur 433Mhz Alimentation 5V Alimentation 220V en 5V

Tools

Ordinateur pour programmation. Fer a souder, fil

Step 1 - Les prises électriques

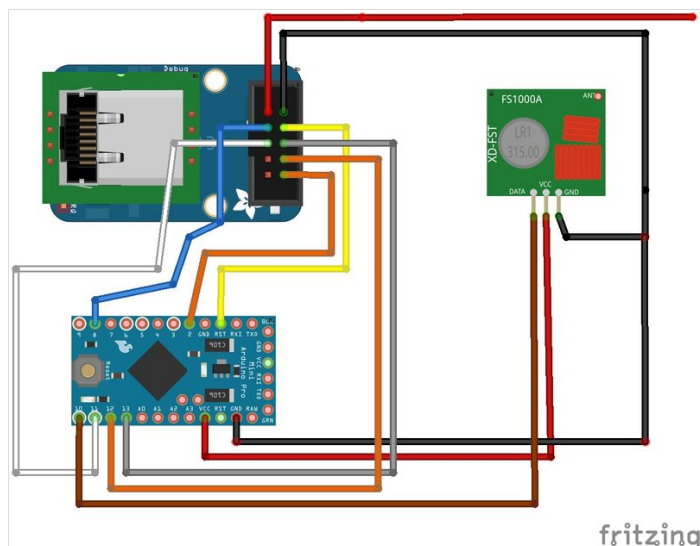
Je vais commencer par la fin, la partie réception. Pour l'instant j'ai trois sortes de récepteurs différents (prises, interrupteurs et modules réceptions volets roulants) mais je vais commencer par la commande des prises électriques. (Une petite lampe, un écran, un disque dur externe...) Des prises achetées dans un magasin de bricolage (Monsieur Bricolage) 15€ les trois avec la télécommande. Il faut des prises « simples » pas avec un code tournant donc souvent le meilleurs marchés. J'ai d'abord trouvé le code d'émission en utilisant la librairie RcSwitch. Un Arduino avec un récepteur 433Mhz et le code d'exemple de la librairie « receivedemo_simple » permet d'obtenir le code, le nombre de bit et le protocole bien noter tout cela pour la suite. De ce côté-là pas plus à faire il suffit de brancher la prise avec un accessoire et il sera commandé par notre émetteur avec le bon code. Lien librairie RcSwitch : <https://github.com/sui77/rc-switch>



Step 2 - Emetteur 433Mhz et carte Ethernet

Maintenant j'ai des codes il me faut un module pour les transmettre. J'ai d'abord fait des essais simple avec un Arduino et un émetteur 433Mhz et toujours avec la librairie RcSwitch et l'exemple « SendDemo » Si tout va bien vous devez pouvoir allumer et éteindre votre prise électrique à distance mais pour l'instant c'est encore moins portable que la télécommande... Je voulais un émetteur connecté sur mon réseau pour lui envoyer des commandes à distances. J'ai fait le choix d'une carte Ethernet que j'ai connecté sur un arduino ProMini en ajoutant un émetteur 433Mhz pour envoyer les codes suivant les autres reçues. En fait c'est simple la carte Ethernet aura une IP sur mon réseau il suffit d'ajouter à la suite de cette IP ce qu'on veut faire du genre LED1 = ON ... Lien librairie Ethercard : <https://github.com/jcw/ethercard>
Code Arduino :

https://github.com/Michelgard/Arduino_433Mhz/blob/master/ether_net_et_emmeteur.ino Une fois le montage terminé et le code dans l'Arduino un simple navigateur nous permet de faire les essais en tapant dans la barre d'adresse l'IP de la carte Ethernet et l'action à faire. Quand on ajoute des accessoires il suffit de reprendre le code pour ajouter le 'if' avec le bon code de la prise, de l'interrupteur ...



Step 3 - Les commandes en page WEB

Bon on a bien avancé mais il faut encore taper une adresse dans un navigateur avec un code pas forcément facile à retenir ... Mais c'est déjà sympa non ! Là il faut simplement faire quelque page web. J'ai fait le choix d'un hébergement à la maison sur un Raspberry PI. Faible consommation, facile à mettre en route, installation Apache facile et bien documenté sur le net. Et il peut tourner sans problème 24/24 et 7/7 bien sûr pas d'écran et pas de clavier juste une alim et une connexion à la box. Celui que j'utilise pour la domotique et même en direct sur la box j'utilise pas le wifi. Donc pour faire cette page j'utilise plusieurs choses. Bien sûr HTML, JavaScript avec AJAX et quelques images de boutons. J'utilise aussi une connexion avec une base de donnée pour garder en mémoire la position des boutons en ON/OFF. L'AJAX permet une mise à jour de la page pour chaque bouton permettant d'utiliser plusieurs pages de modification. Comme cela si je change un bouton sur mon téléphone le même bouton sur ma tablette va changer de position dans les 5 secondes. Chaque boutons est un formulaire HTML (en JavaScript) et il lance une fonction `emetteur()` qui va envoyer sur le réseau l'adresse IP du module Ethernet avec en argument l'accessoire et le ON ou OFF. Le fopen va être envoyé x fois suivant un argument passer à la fonction, souvent 2 fois pour être sûr de la bonne réception avec une bonne moyenne. Dans mon cas avec deux envoies j'ai un taux de réussite de 99% dans mon appartement. Et bien sûr il y a une mise à jour de la donnée dans la base. La base de données pour la position des boutons :

```
https://github.com/Michelgard/Arduino_433Mhz/blob/master/Tabl
e%20position_prise J'ai des boutons simple pour les interrupteurs
(j'y reviens plus loin) car c'est le même codes qui fait ON et OFF mais
je change quand même la valeur dans la base car j'ai aussi une
application qui me retourne la position de chaque appareil sur mon
téléphone. fonction emetteur($led, $on_off, $nb){ global $bdd;
$url="http://xx.xxx.xx.xx:80/?" . $led . "=" . $on_off;
for ($i=0; $iexec($sql); } Les interrupteurs AUTO sous certain
boutons sont là pour automatiser l'allumage des lampes en cas
d'absence. Dans une base de données j'ai les horaires (allumage et
extinction) je lance une tache CRON à chaque heure qui vérifie s'il
faut faire quelque chose et si c'est le cas envoie l'ordre au module
Ethernet. Code Auto CRON :
https://github.com/Michelgard/Arduino_433Mhz/blob/master/auto
_cron.php Le code complet de l'interface web est là :
https://github.com/Michelgard/Arduino_433Mhz
```



Step 4 - Les interrupteurs et les commandes de volets roulants

Donc maintenant vous avez des prises électrique qui de commande avec un module Arduino/Ethernet et par une page WEB. J'ai aussi des interrupteurs. Un interrupteur tactile avec une télécommande et en plus en 433Mhz. Donc rien de plus facile on recommence par trouver le code de la télécommande et on va pouvoir commander nos lumières de la même façon. Ces interrupteurs ont plusieurs avantages, ils sont jolies et tactile ça fait un très belle effet dans la maison et commandable en 433Mhz sans problème. J'ai aussi intégré une commande pour volet roulant dans mon installation domotique. Au début j'avais pensé à acheter un module à encastré sous les boutons de commande. Mais j'avais peur que le module soit trop important pour mes boîtiers (installation électrique assez ancienne) et que les codes soit tournant. Je me suis dit que maintenant j'avais la maîtrise des code avec la librairie RcSwitch et pourquoi pas faire le récepteur ... Un Arduino avec un récepteur 433Mhz et une platine avec deux relais pour simuler les boutons. Dans mon émetteur j'invente un code au format des autres accessoires (prises et interrupteurs) et dans mon programme Arduino je vérifie si le code est reçu. Les codes :

```
define FreqMonte 4232320 define FreqStop 4232322 define FreqDescente 4232324 Et dans la boucle : value = Reception(); if (value == FreqMonte){ Et la fonction réception : long Reception(){ value = 0; if (mySwitch.available()) { value = mySwitch.getReceivedValue(); mySwitch.resetAvailable(); } return value; } Le code complet :
```

https://github.com/Michelgard/Arduino_433Mhz/blob/master/Commandes_Volet.ino J'ai gardé la possibilité d'utiliser les boutons de commande classique mais maintenant ils passent par l'Arduino.

Je vous invite aussi à aller voir un complément avec une commande vocale de tout cela basé sur un Raspberry PI et un programme écrit en Python (<https://www.blog-de-michel.fr/raspberry-pi-domotique/>)

Et bien sur pour toute précision une seule adresse : contact@blog-de-michel.fr

Step 5 -
