# Présentation Générale

## Projet

### Contacts

* Vincent AUGUEY (VAU), initiateur du projet (vincearcade50@gmail.com).
* Thomas AUGUEY (TAU), analyste-programmeur (avaace@hotmail.fr).

### Objectif

Le produit étudié doit permettre d’automatiser et contrôler les équipements électriques de son habitat via le réseau informatique. En d’autres termes : Faire de la Domotique.

Ainsi, les équipements contrôlés manuellement pourront l’être à distance via un terminal informatique.

### Motivations

Ce projet, initié par Vincent AUGUEY (VAU), est une façon de partager une aventure technique et personnelle. La possibilité d’une commercialisation serait une concrétisation si la finalité le permet.

### Besoin

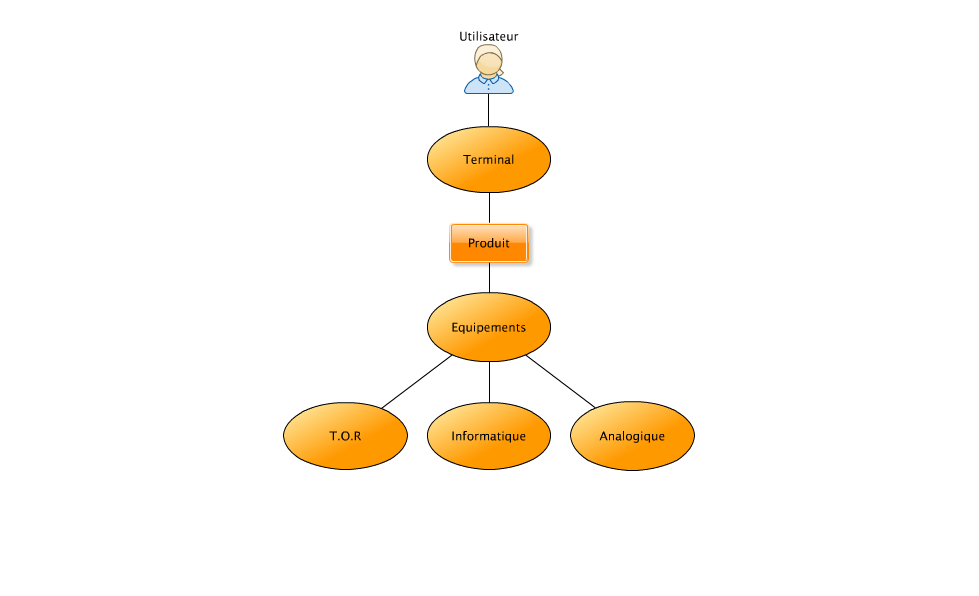
L’utilisateur de l’habitat doit :

* Pouvoir exécuter des ordres et recevoir des notifications de ses équipements via un ou plusieurs terminaux informatiques (ordinateur, tablette et smartphone).
* Activer/Désactiver le pilotage informatique. Le contrôle manuel ne doit pas être affecté par l’installation.

### Contraintes

* Aucune date limite, aucune concurrence. Le projet peut mûrir doucement et évoluer sans contreparties.

## Environnement



### Equipements

Les équipements représentent un élément de l’habitat qui possède au moins 2 particularités :

* Une alimentation électrique
* Une fonction pilotable

### Equipement T.O.R (tout ou rien)

Equipement basique ne pouvant avoir que 2 états : Marche ou Arrêt (On/Off). Exemples : Ampoule (Eteint/Allumé) ; Porte et Fenêtre (Ouvert/Fermé) ; Chauffage (Marche/Arret) ; etc…

### Equipement Analogique

Equipement possédant un état variable entre 0 et 1. Ce type d’équipement est contrôlé via une variation de tension électrique. Exemples : Lumière tamisé (0.5) ; Volet roulant (entrouvert=0.33) ; etc…

### Equipement Informatique

Equipement possédant plusieurs états et pouvant communiquer sur réseau informatique. Exemples : Un ordinateur (Allumé, Eteint, Ouvrir navigateur, Exécuter scan antivirus, …) ; Porte (Fermer, Ouvrir) ; Chauffage (Mode, Température)

### Terminal

Un poste informatique permettant de communiquer sur le réseau et possédant une interface utilisateur. Il peut s’agir d’une tablette, un smartphone, un ordinateur ou encore une télécommande.

### Utilisateur

L’utilisateur donne les commandes via l’interface du terminal.

# Présentation Technique

## Système



**(5) Config**

**(6) BDD**

(3) Serveur de Communication

### Base de données

* Rôle
* Pourquoi ?

### Clients

* Tablette
* Smart Phone
* PC
* Android
* Simplicité

### Administration

* PC / Windows
* Super Utilisateur
* Sécurité

### Serveur

* Arduino
* Communication
* Configuration

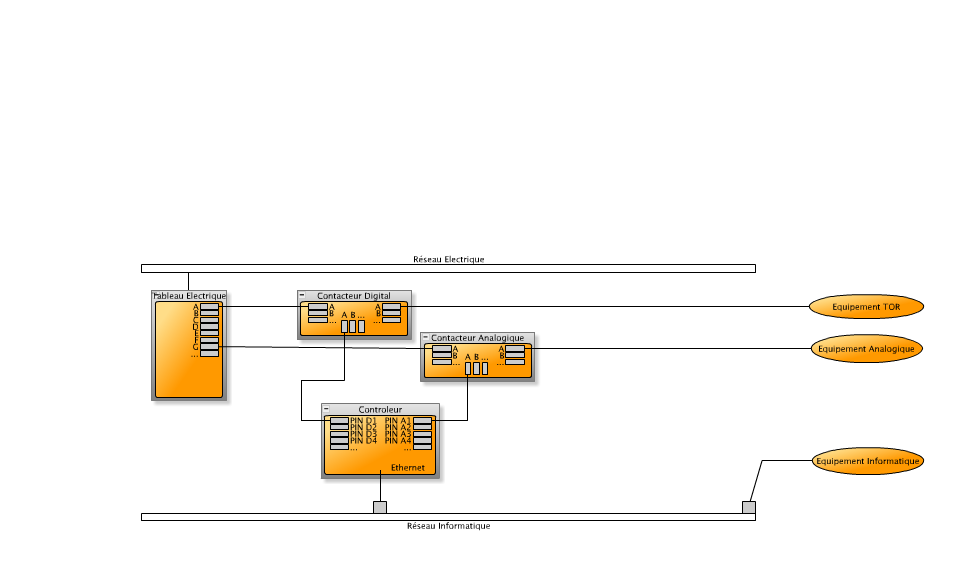
### Equipements

* Matériel électrique
* Arduino Mini

### Communication

* Plusieurs équipements sur un serveur
* Plusieurs Clients sur un serveur
* 1 PC de programmation

## Réseau



## Arduino

Le projet part du constat suivant : A l’heure actuelle, l’aménagement domotique d’un habitat coûte plusieurs milliers d’euros (voir étude concurrence). Ce qui est un très onéreux pour la plupart des gens.

Parallèlement, les systèmes Arduino (voir : www.**arduino**.cc) permettent à moindre coût de s’équiper de cartes d’entrées-sorties programmables réalisant le travail d’un automate. Ce système permet au grand public de réaliser des projets de développement. Arduino propose gratuitement les outils techniques et pour réaliser un système domotique.

Le projet Arduino consiste à développer un système permettant aux développeurs d’intégrer facilement leurs propres équipements.

Ce système comprend :

* Un logiciel de management de l’installation
* Plusieurs interfaces de pilotage
* Un programme arduino server permettant le pilotage des équipements

### Licence

* Exploitation commerciale
* Partage des sources

### Possibilités technique

* Connecteurs analogique
* Connecteurs digital
* Bus série

### Limite technique

* Connecteurs
* Connecter plusieurs cartes grâce au multiplexage (bus i2c)

### Programmation

* IDE
* Port série / USB

### Modèles

* Arduino MEGA

### Modules

* WIFI
* Ethernet
* GSM
* Détecteur de présence
* Détecteur lumière
* Ecran LCD
* Radio

## Equipement

Un équipement est un microcontrôleur doté d’entrées-sorties. Un équipement peut être une carte Arduino ou tout autre équipement capable de communiquer avec l’Arduino-Server.

### Fonctions

* Recevoir un message en série
* Envoyer un message en série
* Lire un message texte (conversion au format machine => Commande,Configuration,etc…)
* Exécuter une commande
* Exécuter une configuration

## Arduino Server

L’Arduino-Server est une carte équipée d’une interface réseau, de capacités de stockages et de performances accrues. Cette carte communique directement les applications clientes et d’administrations pour piloter les équipements.

L’Arduino-Server est un équipement à part entière et possède les mêmes fonctions qu’un équipement de base. Cependant, l’Arduino-serveur écrit/reçoit les messages de l’administration via le réseau informatique.

### Fonctions

* ++ Fonctions d’un équipement
* Recevoir un message en réseau
* Envoyer un message en réseau
* Transmettre un message en série

## Message

Le message structure une information sur le réseau. Un message peut prendre plusieurs formes :

* Une commande
* Une configuration
* …

### Format

Un message texte est codé au format ASCII.

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Taille (bytes)** | **Définition** | **Commentaire** |
| 0 | 3 | Signature | ="MSG" |
| 3 | 3 | Type | Type du message, prend l’une des valeurs suivantes :  "CMD" = Commande  "CFG" = Configuration |
| 6 | 6 | CodeEquip | Code de l’équipement concerné par le message |
| 12 | ? | Paramètres | "Nom=Valeur;…" |

### Message de commande

Une commande permet de donner une instruction formatée à un équipement.

Une commande est structuré par :

* Un code commande permettant d’identifier le type de commande
* Un jeu de paramètres associatifs Nom/Valeur

Exemples de messages de commandes :

|  |  |  |  |
| --- | --- | --- | --- |
| Émetteur | Code message | Message | Paramètres |
| Desktop Admin | CMD | Lit la valeur d’un Pin digital | **CodeCmd** : "DIN"  **PinNum** : Numéro de pin |
| Desktop Admin | CMD | Ecrit la valeur d’un Pin digital | **CodeCmd** : "DOUT"  **PinNum** : Numéro de pin  **Value** : Valeur (0 ou 1) |
| Desktop Admin | CMD | Lit la valeur d’un Pin analogique | **CodeCmd** : "AIN"  **PinNum** : Numéro de pin |
| Desktop Admin | CMD | Ecrit la valeur d’un Pin analogique | **CodeCmd** : "AOUT"  **PinNum** : Numéro de pin  **Value** : Valeur (0-255) |
| Desktop Admin | CMD | Active la LED interne | **CodeCmd** : "LEDH" |
| Desktop Admin | CMD | Désactive la LED interne | **CodeCmd** : "LEDL" |
| Arduino Server | STE | Retourne la valeur d’un Pin | **Value** : Valeur  **PinNum** : Numéro de pin |

### Message d’état

Un état est une propriété persistante d’un équipement.

Un état est représenté par une valeur à un instant T.

Un identificateur unique distingue un état.

Exemples de messages d’état:

|  |  |  |  |
| --- | --- | --- | --- |
| Émetteur | Code message | Message | Paramètres |
| Desktop Admin | STA | Définit le mode de défilement des leds | **CodeState** : "PIN\_MODE"  **Value** : Linear |

## Client

L’application cliente doit donner au client une interface pour piloter les équipements de l’installation. L’interface doit être sécurisée et posséder une interface la plus ergonomique possible.

### Interface

* Visualisation

### Fonctions

* Envoyer un message en réseau
* Lire la configuration d’un équipement
* Afficher la configuration d’un équipement

## Administration

Le programme d’administration gère une base de données des équipements. Elle permet de reconfigurer et de piloter l’Arduino server.

### Interface

### Fonctions

* Recevoir un message sur le réseau
* Envoyer un message sur le réseau
* Lire la configuration d’un équipement
* Ecrire la configuration d’un équipement
* Modifier la configuration d’un équipement
* Afficher la configuration d’un équipement
* Ajouter, supprimer, modifier un équipement

## Base de données

### Entités

* Serveur
* Equipement
* Commande
* Etat

### Fonctions

* Créer / Modifier / Supprimer un server
* Créer / Modifier / Supprimer un équipement