# 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

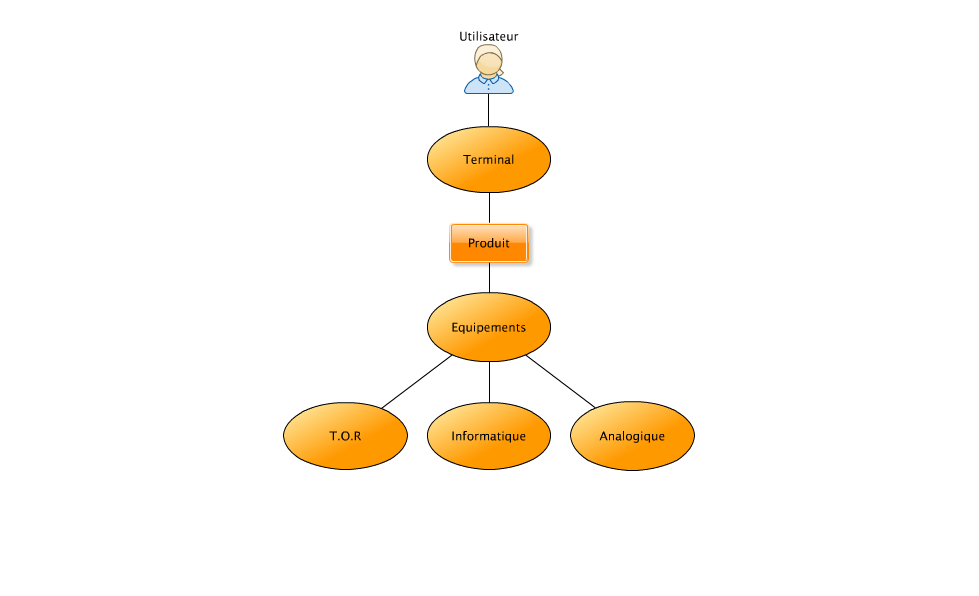
L’utilisateur de l’habitat doit :

* 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

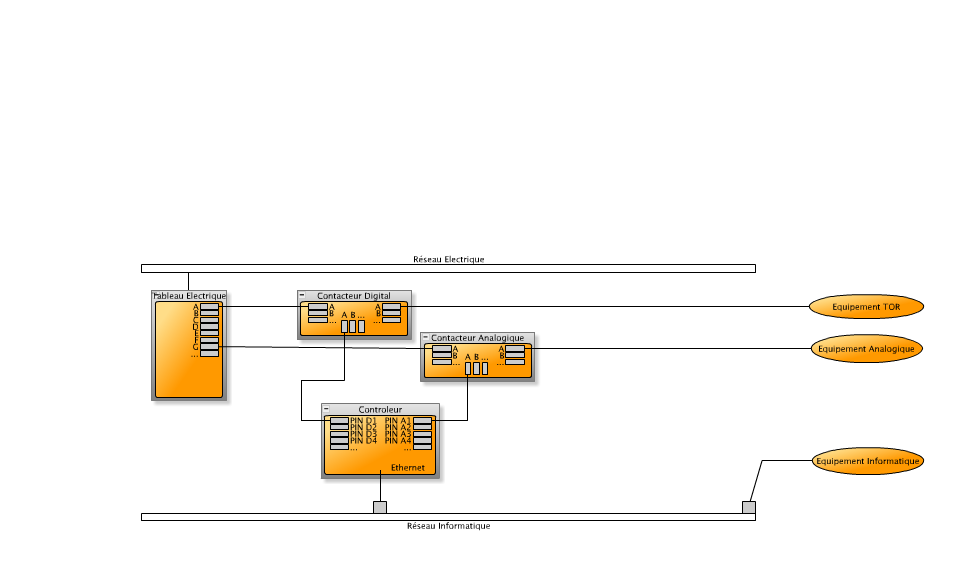
### Equipements

* 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.

Le système comprend :

* Un logiciel de management de l’installation
* Plusieurs interfaces de pilotage
* Un programme Arduino-Server permettant de communiquer avec les clients
* *n* Programmes Arduino pour chaque équipement

### Licence

* Exploitation commerciale
* Partage des sources

### Possibilités technique

* Connecteurs analogique
* Connecteurs digital
* Bus série
* Démultiplier le nombre de connecteurs en multiplexage (bus i2c)

### Limite technique

* Nombres de connecteurs d’entrées/sorties

### Programmation

* IDE
* Port série / USB

### Modèles

* Arduino MEGA

### Modules

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

## Arduino Server

L’Arduino-Server est une carte Arduino équipée d’une interface réseau, de performances et de capacités de stockages accrues. L’Arduino-Server à la responsabilité de servir de point de communication entre les équipements et le réseau informatique (applications clientes).

L’Arduino-Server possède les mêmes propriétés qu’un équipement et de ce fait, il permet de réaliser des tâches qui lui sont propres.

### Définition

* Une adresse MAC
* Une adresse IP

### Fonctions

En plus des fonctions d’un équipement, l’Arduino-Serveur permet de :

* Recevoir un message du réseau
* Envoyer un message sur le réseau
* Connection au réseau

## Equipement

Un équipement est une carte Arduino (microcontrôleur doté d’entrées-sorties) permettant de communiquer avec l’Arduino-Server.

L’équipement réalise des actions qui lui sont propre dans un système fermé.

### Définition

* Un identifiant

### Fonctions

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

## Objets

Les objets représentent les éléments d’un équipement, ils permettent à la fois d’identifier et de généraliser certaines fonctionnalités d’un équipement.

Les objets sont un moyen de normaliser des sous-ensembles d’équipements.

### Définition

Un équipement définit :

* Des états
* Des commandes
* Des configurations
* Un identifiant

### Fonctions

* Evènement périodique

### Exemple

Prenons l’exemple d’un interrupteur On/Off, celui-ci est souvent commun à plusieurs équipements de l’installation.

Un interrupteur comprend :

* 2 Etats :
  + Ouvert
  + Fermé
* 2 Commandes :
  + Ouvrir
  + Fermer
* 2 Configurations :
  + Numéro de Pin en entrée
  + Numéro de Pin en sortie

## Message

La communication entre les équipements Arduino et  les applications clientes repose sur un système de messagerie.

Un message permet l’une des actions suivantes :

* Envoyer une commande
* Envoyer / Recevoir une configuration
* Recevoir un état

### Commande

Une commande permet de donner une instruction à un équipement.

Sa structure se définit ainsi :

* Un code commande permettant d’identifier l’action à réaliser
* Un jeu de paramètres associatifs Nom/Valeur

Commandes standards d’un équipement:

|  |  |  |  |
| --- | --- | --- | --- |
| **Émetteur** | **Type** | **Message** | **Paramètres** |
| Desktop Admin | CMD | Lit la valeur sur un Pin digital | **CodeCmd** : "DIN"  **PinNum** : Numéro de pin |
| Desktop Admin | CMD | Ecrit la valeur sur un Pin digital | **CodeCmd** : "DOUT"  **PinNum** : Numéro de pin  **Value** : Valeur (0 ou 1) |
| Desktop Admin | CMD | Lit la valeur sur un Pin analogique | **CodeCmd** : "AIN"  **PinNum** : Numéro de pin |
| Desktop Admin | CMD | Ecrit la valeur sur 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" |

**Etat**

Un état est une propriété persistante définit par l’équipement. Elle indique un état du système à un temps T.

Exemples d’un message d’état:

|  |  |  |  |
| --- | --- | --- | --- |
| **Émetteur** | **Type** | **Message** | **Paramètres** |
| Arduino Server | STE | Retourne la valeur d’un Pin | **Value** : Valeur  **PinNum** : Numéro de pin |

### Configuration

Une configuration est une propriété persistante définit par l’utilisateur. Elle indique les consignes de fonctionnements désirées.

Exemples d’un message de configuration:

|  |  |  |  |
| --- | --- | --- | --- |
| **Émetteur** | **Type** | **Message** | **Paramètres** |
| Desktop Admin | CFG | Définit le mode de défilement des leds | **CodeCfg** : "PIN\_MODE"  **Value** : "LINEAR" |

### Format de données

Un message est toujours constitué d’un en-tête définit comme ceci :

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Taille (bytes)** | **Définition** | **Commentaire** |
| 0 | 3 | Signature | "MSG" |
| 3 | 3 | Type | Type de message  :  "CMD" : Commande  "CFG" : Configuration  (Définit le format des données après l’en-tête) |
| 6 | 1-128 | CodeEquip | Code de l’équipement concerné |
| ? | 1 | Séparateur | " ;" |
| ? | 1-128 | CodeObjet | Code de l’objet concerné |
| ? | 1 | Séparateur | " ;" |
| **Fin de l’en-tête** | | | |

Suivi de *n* paramètres :

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Taille (bytes)** | **Définition** | **Commentaire** |
| 18 | 1-128 | Nom | Nom du paramètre |
| ? | 1 | Séparateur | Séparateur de chaine  "=" |
| ? | 1-128 | Valeur | Valeur du paramètre |
| ? | 1 | Séparateur | Séparateur de fin  ";" |
| **Fin de paramètre** | | | |

## 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