

RAPPORT DE PROJET – EI4 AGI

PROJET DOMOTIQUE DAIO « Domotique All-In-One »

Projet réalisé par

Paul MEGUEULE
Fabien NOIR
Nicolas ROULLEAU
Florian ROUSSEL

Projet encadré par

Bertrand COTTENCEAU
Mehdi LHOMMEAU



REMERCIEMENTS

Tout d'abord, nous tenons à remercier tout particulièrement et à témoigner toute notre reconnaissance aux personnes suivantes, pour leur dévouement et leur soutien dans la concrétisation de ce projet ingénieur :

- M. Medhi LHOMMEAU, responsable projet, pour ses conseils éclairés, sa patience, sa disponibilité et pour la confiance qu'il nous a accordée dès l'ébauche du projet et tout au long de ces cinq mois.
- M. Bertrand COTTENCEAU, responsable projet, pour nous avoir accordé toute la confiance nécessaire pour élaborer ce projet librement, et avoir mis à notre disposition tous les moyens disponibles.
- M. Jean-Baptiste FASQUEL, enseignant-chercheur en traitement d'images, pour son investissement et sa contribution.
- M Hassan BOULJROUFI, assistant ingénieur, pour toute l'aide et les conseils apportés durant ce projet.
- L'ISTIA et l'ensemble des enseignants pour leur coopération professionnelle tout au long de cette expérience et pour avoir partagé avec nous, une partie de leurs savoir-faire et de leurs expériences professionnelles.

SOMMAIRE

I. INTRODUCTION.....	6
II. BESOINS ET OBJECTIFS DU PROJET	7
1. CONTEXTE	7
2. MOTIVATION	8
3. LES ENJEUX	8
4. OBJECTIFS ET CONTRAINTES	9
a. Les objectifs techniques.....	9
b. Les objectifs économiques.....	9
c. Les délais	9
III. GESTION DE PROJET	10
1. L'ÉQUIPE.....	10
2. LA PLANIFICATION DE PROJET ET LES OUTILS DE GESTION.....	11
a. Cahier des charges	11
b. Diagramme GANTT	12
c. Diagramme PERT.....	13
3. RÉPARTITION DES TÂCHES ET DES TECHNOLOGIES	14
a. Répartition des tâches	14
b. Répartition des technologies.....	14
IV. DÉVELOPPEMENT TECHNIQUE.....	15
1. LA STRATÉGIE	15
a. La stratégie prévisionnelle	15
b. Architecture matérielle	16
c. Architecture logicielle	17
2. LES SOLUTIONS MATÉRIELLES	18
a. Alimentation électrique	18
b. Radio.....	19
c. Raspberry Pi.....	19
d. Arduino	19
e. Le module d'éclairage	20
f. Le module de température	20
g. Camera.....	20
h. Maquette	21
3. LES SOLUTIONS LOGICIELLES.....	22
a. Les technologies	22
CGI.....	22

CSS	22
HTML	22
JAVASCRIPT	22
JSON	22
Langage C – Atmel AVR.....	23
PHP	23
Python	23
XML.....	23
b. Leurs applications.....	24
Site web.....	24
Protocole radio	26
Commande manuelle	28
Gestion des scénarios	29
Caméra - Configuration et paramétrage.....	32
V. BILAN DU PROJET	33
1. Apports individuels et collectifs	33
2. Conclusion générale.....	34
VI. PERSPECTIVES	35
VII. BIBLIOGRAPHIE.....	36

Table des figures

Figure 1 - Diagramme GANTT	12
Figure 2 - Diagramme PERT	13
Figure 3 - Répartition des tâches	14
Figure 4 - Répartition des technologies	14
Figure 5 - Architecture matérielle	16
Figure 6 - Architecture logicielle	17
Figure 7 - Schéma électrique	18
Figure 8 - Module radio	19
Figure 9 - Raspberry Pi	19
Figure 10 - Arduino	19
Figure 11 - Relai	20
Figure 12 - TMP102.....	20
Figure 13 - Caméra	20
Figure 14 - Schéma de la maquette.....	21
Figure 15 - Réalisation de la maquette	21
Figure 16 - Exemple de panneau	24
Figure 17 - Gestion des caméras	25
Figure 18 - Gestion des panneaux.....	25
Figure 19 - Adresses dans le protocole radio.....	26
Figure 20 - Commande manuelle.....	28
Figure 21 - Fichier crontab	29
Figure 22 - Fonctionnement logiciel de la gestion des scénarios	31
Figure 23 - Configuration et paramétrage de la caméra	32

I. INTRODUCTION

Dans le cadre de notre seconde année du cycle ingénieurs en Automatique et Génie Informatique à l'ISTIA, il nous est proposé un projet de 4 mois nous permettant de mettre en pratique nos connaissances et nos compétences professionnelles au travers d'un cahier des charges ayant pour finalité la conception et le développement d'une application industrielle en accords avec nos intérêts professionnels.

Ayant une passion commune pour le Raspberry Pi et la domotique, notre groupe composé de Paul MEGUEULE, Fabien NOIR, Nicolas ROULLEAU et Florian ROUSSEL, a saisi l'opportunité d'exploiter cet intérêt commun pour soumettre l'ébauche d'un projet personnel innovant au responsable de l'année EI4 M. Mehdi LHOMMEAU.

II. BESOINS ET OBJECTIFS DU PROJET

1. CONTEXTE

Les premiers développements de la domotique sont apparus au milieu des années 1980 grâce à la miniaturisation des systèmes électroniques et informatiques. Dès lors, l'industrie a concentré ses expérimentations sur le développement d'automates, d'interfaces et d'outils apportant confort, sécurité et assistance au sein d'un édifice.

Aujourd'hui, la domotique se définit comme étant l'ensemble des techniques et technologies permettant de superviser, d'automatiser, de programmer et de coordonner les tâches de confort, de sécurité, de maintenance et plus généralement de services dans l'habitat individuel ou collectif. La domotique inclut également les notions de communications inter-équipements, d'environnement autonome et d'interactions multimodales.

La domotique s'est imposée dans différents domaines, autant dans l'industrie que chez le particulier. Cependant, les entreprises peinent à diversifier leurs offres et la plupart des investissements industriels réalisés à ce jour dans ce domaine se sont soldés par des échecs.

Pour réaliser une installation domotique, les fabricants se concentrent sur les trois technologies suivantes :

- Les ondes radio (essentiellement le wifi ou la radio fréquence)
- Les courants porteurs en ligne (CPL)
- Les technologies filaires (bus)

Malgré un nombre de supports physiques restreints, il existe de nombreux protocoles propriétaires et libres qui, pour la plupart du temps, ne sont pas interopérables : la majorité des fabricants développent un protocole qui est propre à leurs produits, et qui par conséquent n'est pas ou peu supporté par d'autres systèmes. Parmi les plus connus, on retrouve le protocole X10, zWave, Ethernet, Dali et Chacon.

Au-delà de la non-interopérabilité, équiper sa maison reste très coûteux et nécessite généralement des travaux si cela n'a pas été pris en compte lors de la construction. De nombreuses contraintes entourent la domotique et freinent la notoriété de ce domaine qui peine à se faire une place chez les particuliers.

2. MOTIVATION

Partant de ce postulat, nous est venu l'idée de combler cette insatisfaction chez le public en quête d'automatisation et de confort à petit prix. Nous avons ainsi décidé de réaliser une solution domotique pour le particulier qui soit abordable financièrement, ne nécessitant pas de travaux, modulable, tout en alliant robustesse et stabilité.

C'est avec ces objectifs en tête, que nous avons créé « DAIO », la « Domotique All-In-One ».

3. LES ENJEUX

Les enjeux du projet sont technologiques, économiques et humains.

En premier lieu, c'est le défi technologique qui nous a attirés. Comme évoqué précédemment, la domotique fait intervenir une notion de multimodalité, c'est-à-dire la capacité d'un produit à exploiter plusieurs technologies pour communiquer. Ainsi, nous avons voulu à travers ce projet, tester notre capacité à faire communiquer différentes technologies entre elles. S'ajoute à cela la notion de modularité : nous voulions proposer une solution facilement intégrable dans le foyer du grand public.

Outre la technique, l'aspect économique nous a également motivés dans notre entreprise : pouvons-nous proposer un produit abordable avec les moyens à disposition et le temps imparti ? En effet, si la domotique ne se développe pas depuis une trentaine d'années, c'est en partie à cause de son prix. Les professionnels l'affirment, ce coût élevé est entretenu d'une part par le manque de notoriété (pas de besoin du grand public, manques d'informations) et d'autre part par le lobbying que l'on trouve dans ce domaine. Comme le monde de la maison rassemble divers métiers propres (chauffage, électricité, surveillance, etc...), il est difficile pour l'ensemble de s'accorder sur des choix communs afin de limiter le coût des produits. De ce fait, pour un ensemble cohérent, il faut bien souvent installer des passerelles entre les différents moyens de communication de la maison, ce qui ajoute bien entendu un surcoût. C'est pourquoi nous avons eu l'idée de développer une solution à faible coût et de travailler sur l'uniformité des différents services proposés.

Enfin et non sans importance, c'est le fait de travailler en groupe et l'opportunité de réaliser un projet de A à Z tout en exploitant les compétences de chacun qui nous ont séduits.

4. OBJECTIFS ET CONTRAINTES

a. Les objectifs techniques

Étant pour le moment dans un projet universitaire limité en moyens et en temps, nous avons décidé de restreindre notre projet domotique, en sélectionnant les solutions à développer parmi toutes les possibilités permises.

Ainsi, nous avons projeté dans un premier temps de mettre en place un serveur domotique qui sera le cerveau du système, ainsi que des modules de température et de consommation électrique. Ajouté à ce cœur de projet, nous entreprenons l'étude d'une alimentation électrique adaptée, le choix d'un protocole radio ouvert et simple à mettre en place ainsi que la réalisation d'une maquette et d'un site web. Ces derniers seront l'aboutissement du projet, à même de démontrer tout l'intérêt de DAIO.

b. Les objectifs économiques

Comme évoqué précédemment, un des enjeux du projet est économique. Un prix « maîtrisé » est synonyme de produit grand public. C'est pourquoi nous avons décidé d'étudier toutes les solutions qui seraient à même de baisser le prix de notre projet : alimentation électrique, plateforme de prototypage, modules radios, librairies open source, etc.

De même, un de nos objectifs est d'uniformiser les différentes parties du système : celles-ci doivent pouvoir communiquer sur les mêmes principes (ondes radios, protocoles, etc.) afin que l'ajout éventuel de matériel ne soit pas contraignant.

c. Les délais

Le projet débute le mercredi 4 décembre et s'achève le vendredi 18 avril, soit un peu plus de quatre mois. Afin de terminer ce projet ambitieux à temps, il est important de correctement le gérer et de le tenir à jour grâce aux outils de gestion adéquats. Dans cette optique, nous utilisons des outils autant présents dans le domaine universitaire que dans le monde du travail : journal de bord, diagramme de Gantt, diagramme Pert, etc.

III. GESTION DE PROJET

1. L'ÉQUIPE

Notre équipe de projet est composée de personnalités complémentaires provenant de différents cursus :

- Paul MÉGUEULE, DUT en Génie Electrique et Informatique Industrielle option Automatismes Industriels puis une licence en Electronique et Télécommunication option Domotique.
- Fabien NOIR, fondateur et chef du projet, DUT en Génie Electrique et Informatique Industrielle option Informatique Industrielle
- Nicolas ROULLEAU, DUT en Génie Electrique et Informatique Industrielle avec un quatrième semestre en Ingénierie des Systèmes Électriques et Électroniques
- Florian ROUSSEL, BTS IRIS (Informatique et Réseaux pour l'Industrie et les Services techniques)

Nos diverses expériences professionnelles et personnelles au cours de ces années d'études, nous ont permis d'aiguiser notre curiosité et de nous ouvrir à d'autres domaines et technologies. Nos centres d'intérêts, souvent abordés au cours de nos discussions quotidiennes, nous ont rassemblés cette année autour d'un projet domotique, faisant intervenir le nano-ordinateur de l'année 2012, le Raspberry Pi.

2. LA PLANIFICATION DE PROJET ET LES OUTILS DE GESTION

Pour accompagner le développement du projet dès le stade d'ébauche, un cours de gestion de projet a rapidement été mis en place. Ce cours introduit la méthodologie à suivre et les outils nécessaires au bon déroulement d'un projet. Étant dans une dimension ingénieur, cette gestion est d'autant plus importante que le respect des délais, des coûts et de la performance est essentiel dans la conception d'un système complexe. La gestion de projet permet d'autre part de distribuer les travaux à réaliser entre les membres de l'équipe mais également de créer une base de référence permettant de surveiller les écarts et l'évolution du projet afin d'assurer sa continuité.

a. Cahier des charges

Décrivant l'ensemble des conditions attachées à l'exécution du projet, le cahier des charges nous a permis dans un premier temps, de définir le contexte, les enjeux, les objectifs techniques ainsi que les livrables et les axes de développement envisagés. En organisant nos idées, nous avons ainsi pu vérifier la concordance et la faisabilité de notre projet.

b. Diagramme GANTT

Le diagramme de GANTT est un outil efficace exploitant des données brutes tel que les dates de début et de fin et les durées de chacune des tâches afin de générer une visualisation de l'avancement du projet. Il permet de donner une vue globale des tâches à réaliser, des responsabilités et des ressources associées, de l'idée jusqu'à la mise en service en passant par l'analyse des exigences, l'étude de faisabilité, la conception fonctionnelle, les spécialisations, la réalisation et enfin, les tests. Aussi, il est possible d'organiser une gestion des ressources, leurs disponibilités, leurs coûts, etc.

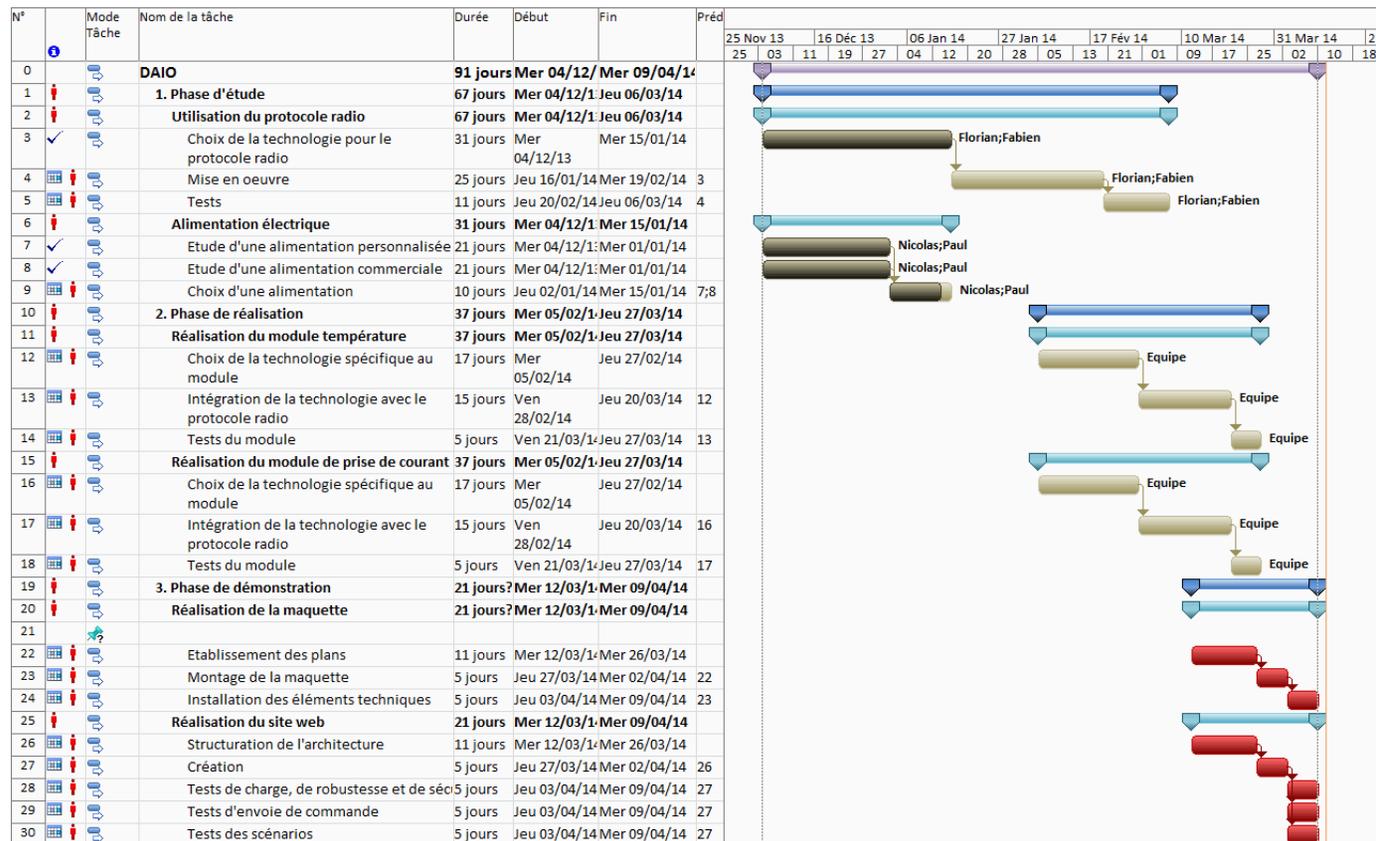


FIGURE 1 - DIAGRAMME GANTT

c. Diagramme PERT

Associé au diagramme GANTT, le diagramme PERT permet de mettre en évidence les liaisons entre les différentes tâches afin d'en déduire un chemin dit « critique ». L'état de chacune des tâches évolue ensuite en fonction de la progression du projet et met en évidence la succession des livrables. Pour une question de lisibilité, nous avons seulement détaillé ci-dessous une partie du diagramme PERT à travers la hiérarchie de deux des phases de notre projet.

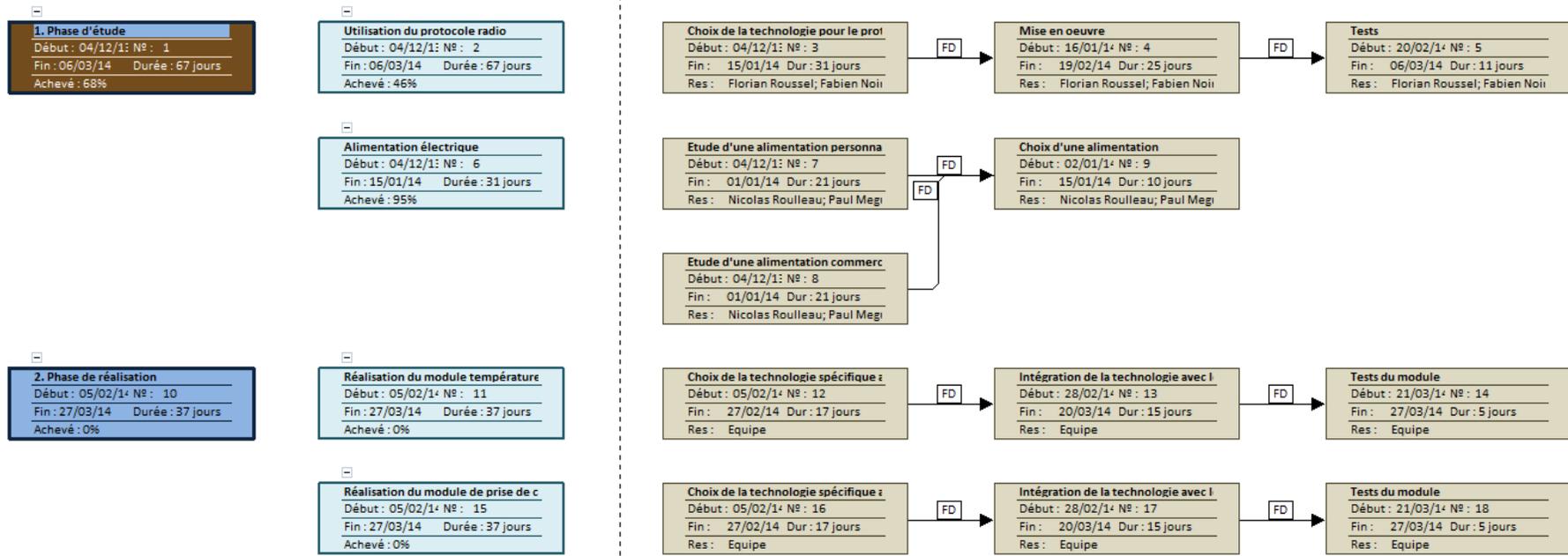


FIGURE 2 - DIAGRAMME PERT

3. RÉPARTITION DES TÂCHES ET DES TECHNOLOGIES

a. Répartition des tâches

Le diagramme ci-dessous illustre la répartition du temps attribué à chacune des grandes étapes du projet. C'est une conclusion graphique résultant de notre gestion de projet, nous permettant d'identifier au premier coup d'œil les travaux chronophages et l'homogénéité entre les tâches.

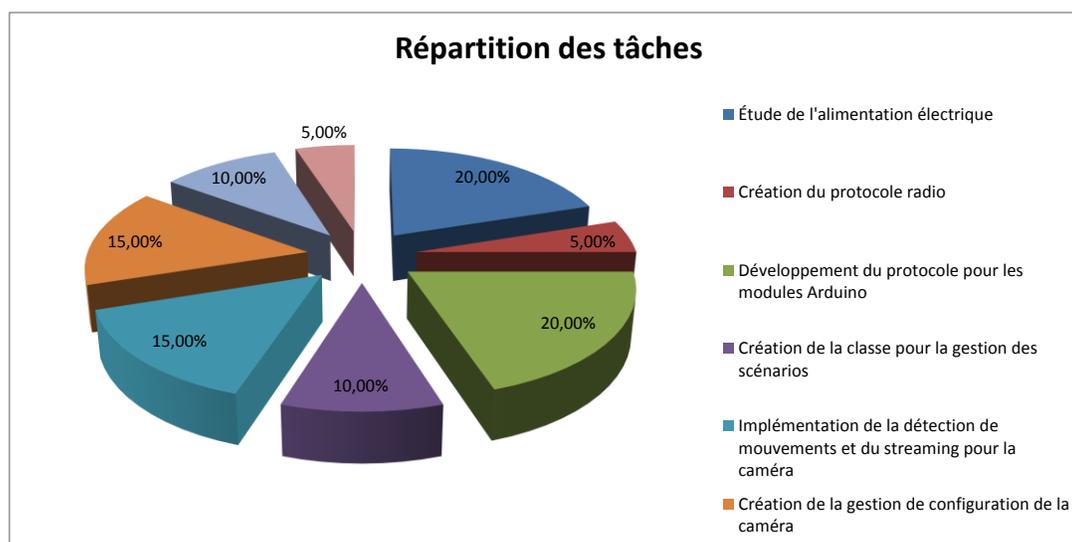


FIGURE 3 - REPARTITION DES TACHES

b. Répartition des technologies

Le diagramme ci-dessous est une visualisation des différentes technologies abordées : le C, le PHP et le Python sont les trois langages auxquels nous avons accordé le plus de temps au cours du développement technique.

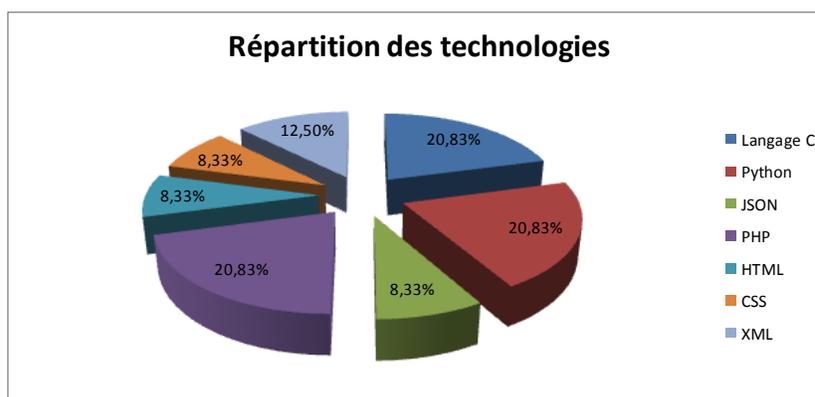


FIGURE 4 - REPARTITION DES TECHNOLOGIES

IV. DÉVELOPPEMENT TECHNIQUE

1. LA STRATÉGIE

a. La stratégie prévisionnelle

Ce projet étant personnel, notre première mission fut de définir nous-mêmes une stratégie prévisionnelle ainsi que les objectifs à atteindre. Bien entendu, cette stratégie a évolué au cours du temps afin de satisfaire nos exigences, mais aussi les contraintes auxquelles nous avons fait face. Nous avons ainsi tissé notre réflexion, avec pour fil conducteur la relation entre l'utilisateur et la maison.

Tout d'abord l'utilisateur : même si celui-ci ne fait pas partie à proprement parler du développement technique, il est primordial d'identifier le public concerné. Quel est le profil de l'utilisateur type ? Voilà le fondement de notre réflexion. Nous avons finalement privilégié une clientèle novice en développant une solution facile d'utilisation, la plus intuitive possible.

À la manière d'une interface homme-machine, le site web permet d'interagir avec le système, sans nécessiter de connaissances pointues de la part de l'utilisateur pour être configuré et utilisé. Ce site inclut une vue d'ensemble du système, les outils de gestion des modules, les pages de configuration de la caméra et des scénarios. Cette partie logicielle est hébergée par le Raspberry Pi.

Le nano-ordinateur est le cœur du projet : il joue le rôle d'intermédiaire entre l'utilisateur et l'environnement lui-même. Grâce à une communication radio maître-esclave, le Raspberry Pi doit être capable d'organiser les consignes de l'utilisateur pour les restituer sous forme de requête à l'Arduino maître.

Relié par une simple liaison série, le rôle de l'Arduino maître est de redistribuer par radio les requêtes aux Arduino esclaves.

Ces derniers sont des unités autonomes distantes, parfois de plusieurs dizaines de mètres, dont nous avons simplifié le fonctionnement de sorte qu'elles puissent seulement envoyer des informations lorsqu'elles en ont reçu l'ordre.

Enfin, nous avons agrémenté notre architecture domotique d'une caméra qui nous permet de proposer de la surveillance ainsi que du streaming vidéo dans notre interface web.

On notera que le contrôle d'éclairage et la gestion de température s'opèrent grâce aux Arduino, au contraire de la caméra, contrôlée sans intermédiaire matériel depuis le Raspberry Pi.

L'utilisateur peut agir de deux manières différentes sur le système DAIO : en utilisant des commandes manuelles ou des commandes automatisées par des scénarios.

Dans sa version finale, notre solution fournit un ensemble de composants contrôlables par l'utilisateur grâce au site web. Nous avons tenus à mettre en avant le contrôle d'éclairage (simulé par des DELs contrôlées par un relai), la gestion de la température ainsi que la vidéo-surveillance.

b. Architecture matérielle

Le diagramme ci-dessous résume notre paragraphe précédent sous la forme d'un schéma :

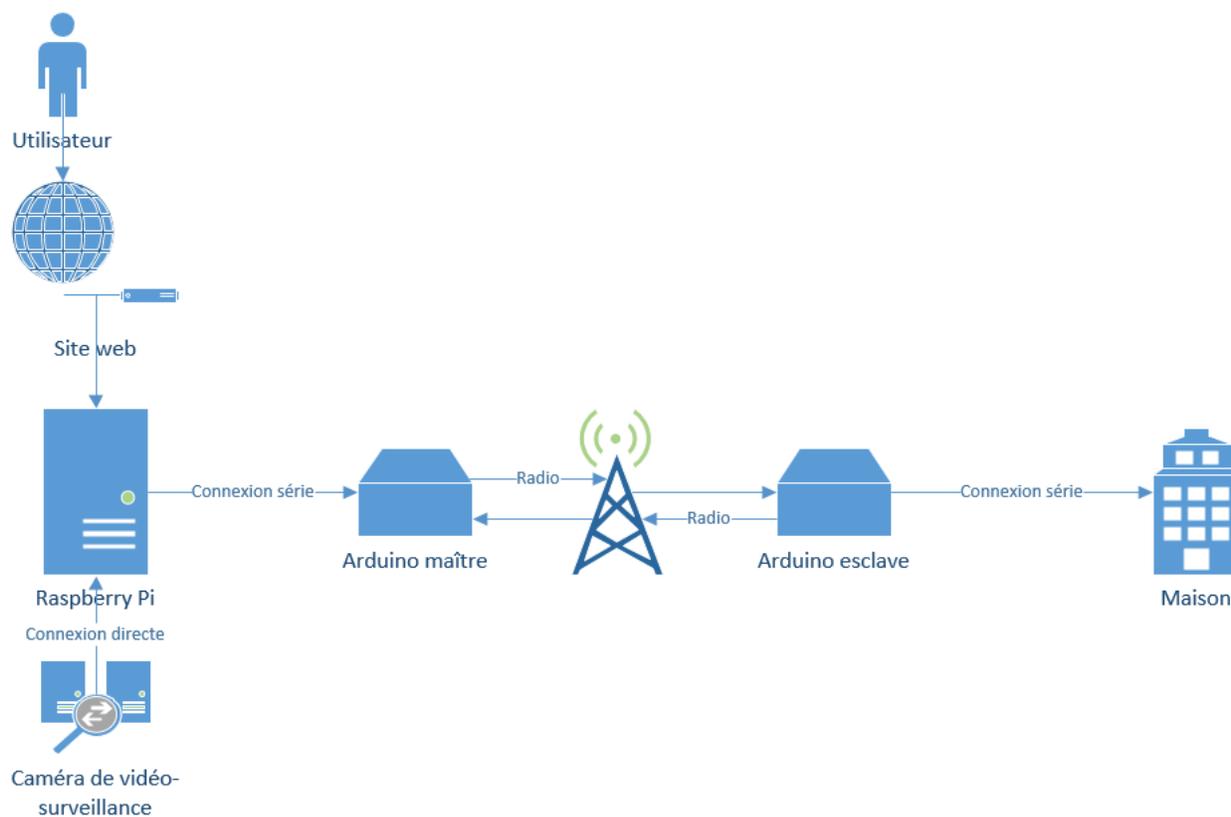


FIGURE 5 - ARCHITECTURE MATERIELLE

c. Architecture logicielle

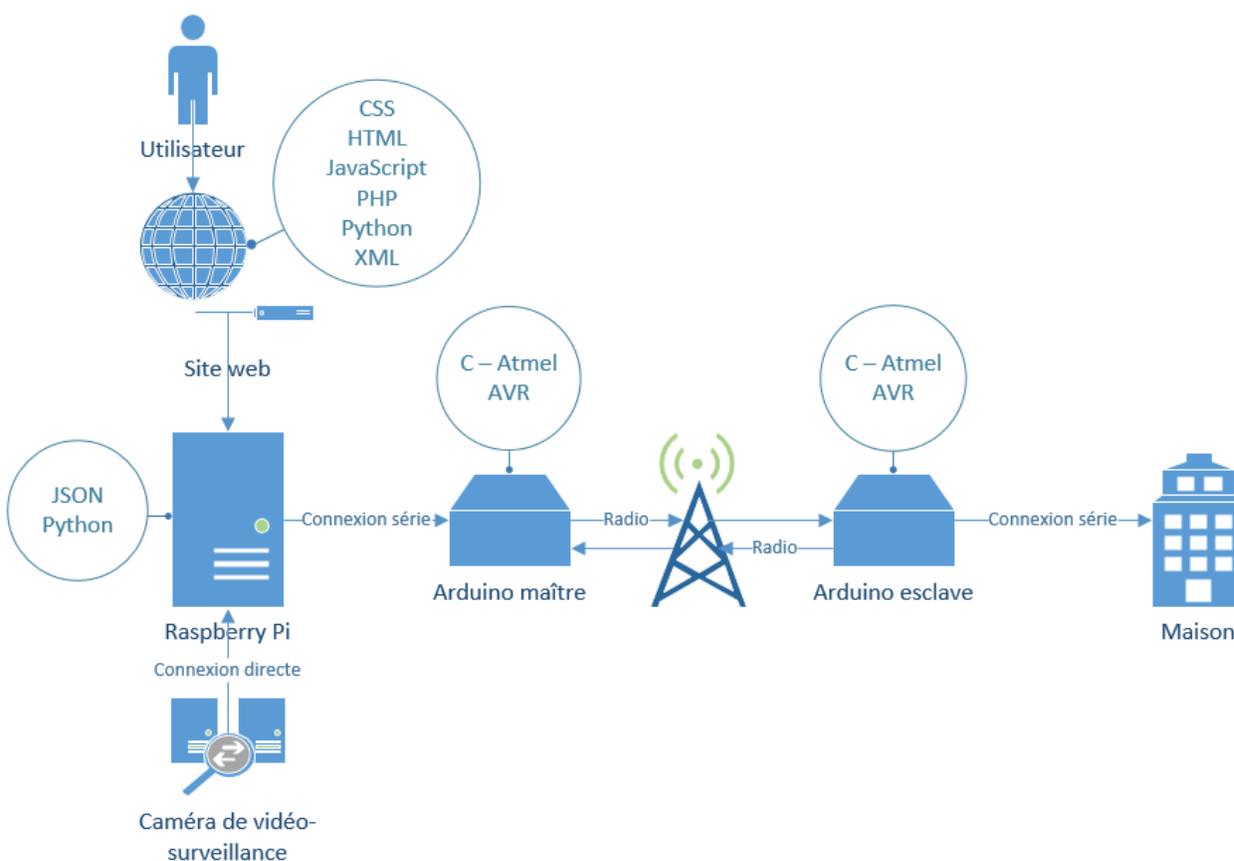


FIGURE 6 - ARCHITECTURE LOGICIELLE

Pour une question de lisibilité et de compréhension, l'intégralité du fonctionnement logiciel sera exprimée dans les paragraphes suivants.

2. LES SOLUTIONS MATÉRIELLES

a. Alimentation électrique

Comme énoncé dans les objectifs techniques, nous avons entrepris l'étude d'une alimentation électrique. Faut-il acheter une alimentation électrique dans le commerce ou la fabriquer nous-même ? Nous avons étudié différents critères tel que le prix et les performances pour faire un choix au plus proche des exigences de notre projet. En effet, DAIO étant un système autonome, le choix de l'alimentation est primordial afin de garantir le contrôle de l'habitation sans interruption.

Après avoir défini les besoins électriques nécessaires, nous avons entamé une phase de recherche de composants afin de limiter au maximum les coûts. Suite à cela, nous avons dessiné les premières ébauches, tout en prenant conseil auprès de M. Hassan et M. Guignard.

Ci-dessous, le schéma de la dernière version :

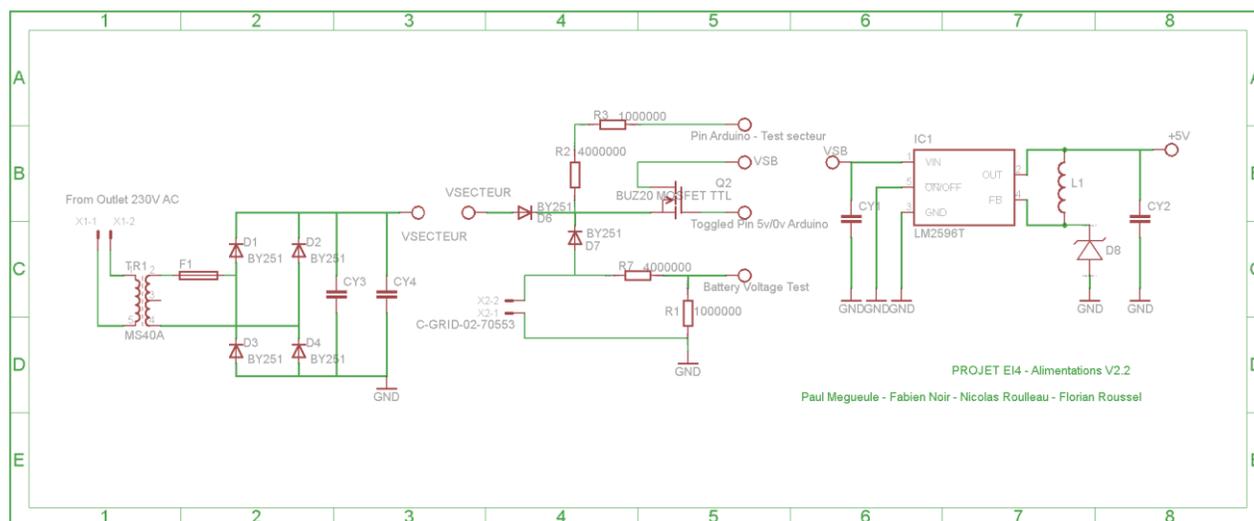


FIGURE 7 - SCHEMA ELECTRIQUE

Le concept comprend, de gauche à droite, un transformateur AC-DC 230V - 12V, un circuit de lissage, un circuit de contrôle testant la tension secteur et un convertisseur DC-DC, dans le but de fournir une tension de 5V au serveur. De plus, nous avons laissé un emplacement afin d'accueillir une batterie pour rendre le système totalement autonome.

Cependant, la conception d'un tel circuit est coûteuse en temps et en argent. DAIO étant à l'état de projet, nous avons finalement opté pour l'achat d'une alimentation dans le commerce, qui ne restreindrait en aucun cas les tests sur une future maquette.

b. Radio



FIGURE 8 - MODULE RADIO

Les modules radios permettent, une fois connectés à une Arduino, de communiquer par ondes-radios à 433 MHz. Dans notre projet, nous les utilisons pour que l'Arduino maître puisse communiquer avec les Arduino esclaves.

c. Raspberry Pi



FIGURE 9 - RASPBERRY PI

Le Raspberry Pi est un nano-ordinateur qui comporte un système d'exploitation Linux. Il nous sert de cerveau au sein de notre système. C'est lui qui va comporter l'ensemble de notre site web ainsi que l'interaction avec les Arduino.

d. Arduino



FIGURE 10 - ARDUINO

Arduino est une carte électronique qui comporte un microcontrôleur programmable. Dans notre solution, nous en utilisons, d'une part pour implémenter la fonctionnalité du module, et d'autre part, pour connecter les émetteurs et récepteurs d'ondes radios.

e. Le module d'éclairage



FIGURE 11 - RELAI

Le relais permet, une fois connecté à une Arduino de contrôler divers appareils électriques. Dans notre projet, nous utilisons 7 relais pour contrôler des DELs présentes dans la maquette.

f. Le module de température



FIGURE 12 - TMP102

Seconde application domotique, le module de température inclut une Arduino et un TMP102, un capteur de température économe avec une faible consommation de $10\mu\text{A}$ et facile à mettre en place grâce à une liaison série I2C.

g. Camera



FIGURE 13 - CAMERA

Le module de caméra se branche grâce à une nappe CSI directement sur le Raspberry Pi. Elle a été développée par la fondation Raspberry Pi. Par conséquent, toutes les bibliothèques sont disponibles. Dans notre projet, la Pi Camera intervient dans la vidéosurveillance ainsi que dans le streaming.

h. Maquette

Pour donner du réalisme à notre projet, nous avons investi du temps dans la conception d'une maquette, une solution économique et représentative de l'utilisation future de DAIO, adaptée à un projet étudiant de cette envergure.

Ci-dessous un aperçu de la maquette sur papier :

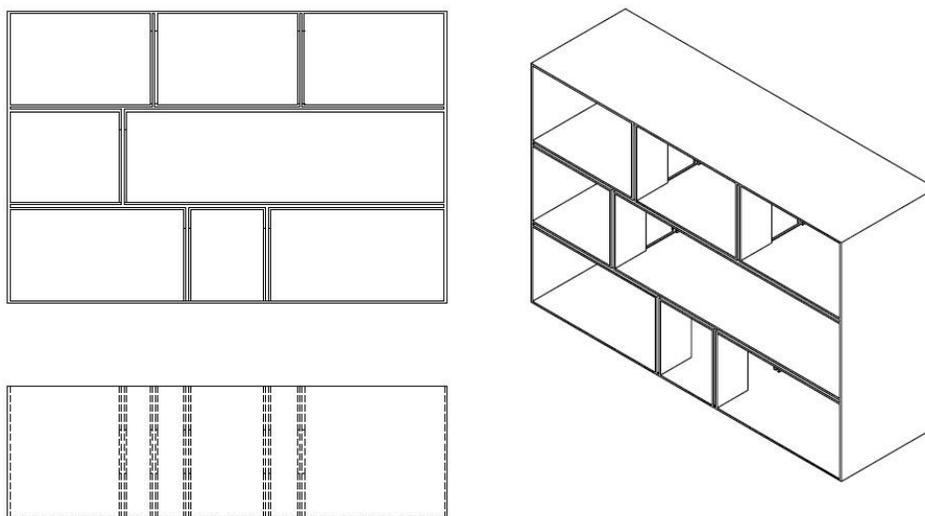


FIGURE 14 - SCHEMA DE LA MAQUETTE

Étant donné la taille modérée de la maquette et en vue d'intégrer les technologies dans celle-ci, la phase de conception fut guidée par les mots « simplicité » et « modularité ». Les phases de tests se sont retrouvées facilitées par ce raisonnement en amont.



FIGURE 15 - REALISATION DE LA MAQUETTE

3. LES SOLUTIONS LOGICIELLES

a. Les technologies

CGI

L'interface Common Gateway Interface abrégé CGI est une passerelle qui permet de faire communiquer des logiciels tournant sur le même ordinateur. Avec CGI, on peut écrire des scripts dans différents langages en les interconnectant.

CSS

De l'anglais Cascading Style Sheets (les feuilles de style en cascade), le CSS constitue un langage informatique permettant d'ajouter des contraintes de mise en forme graphique dans des documents web, dont les normes sont établies par le World Wide Web Consortium (W3C).

HTML

L'Hypertext Markup Language, abrégé HTML, est un langage conçu pour représenter les pages web. Ce langage de balisage permet de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, etc.

JAVASCRIPT

JavaScript, abrégé JS, est un langage de programmation de scripts principalement utilisé dans les pages web interactives.

JSON

JavaScript Object Notation, abrégé JSON, est un format de données textuelles et génériques permettant de représenter de l'information structurée comme le permet le langage XML par exemple.

Langage C – Atmel AVR

Le C est un langage impératif, généraliste, issu de la programmation système. Inventé au début des années 1970 pour réécrire UNIX, il est devenu un des langages les plus utilisés. Le langage C – Atmel AVR est utilisé pour la programmation destinée aux microcontrôleurs.

PHP

PHP : Hypertext Preprocessor, abrégé PHP, est un langage de programmation libre principalement utilisé pour produire des pages Web dynamiques.

Python

Python est un langage de programmation structuré et orienté objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser.

XML

L'Extensible Markup Language, abrégé XML, est un langage de balisage permettant de définir différents espaces de noms, via l'usage de chevrons encadrant les balises.

b. Leurs applications

Site web

Le site web permet de centraliser l'ensemble des informations que l'utilisateur peut observer sur son système domotique. Il peut contrôler à distance les éléments du système, par exemple, les DELs dans la maison.

Le site se veut le plus épuré possible, grâce à une interface intuitive permettant à l'utilisateur de rapidement trouver les informations de son système. Un tableau de bord reprend notamment les différents panneaux disponibles.

#	Date	Time	Amount
3326	10/21/2013	3:29 PM	\$321.33
3325	10/21/2013	3:20 PM	\$234.34
3324	10/21/2013	3:03 PM	\$724.17
3323	10/21/2013	3:00 PM	\$23.71
3322	10/21/2013	2:49 PM	\$8345.23
3321	10/21/2013	2:23 PM	\$245.12
3320	10/21/2013	2:15 PM	\$5663.54
3319	10/21/2013	2:13 PM	\$943.45

FIGURE 16 - EXEMPLE DE PANNEAU

Nous avons ainsi intégré les actions nécessaires pour la gestion des panneaux :

- Ajouter
- Gérer
- Supprimer

Ces outils nous permettent de créer, pour le moment, deux types de panneaux : un affichage numérique et un panneau de contrôle type interrupteur avec retour d'état.

Nous pouvons ainsi :

- Allumer / éteindre les DELs et récupérer leurs états
- Récupérer la température en degrés Celsius ou Fahrenheit à chaque rafraichissement de page

Voici des images représentant la gestion de modules sur le site.

Gestion des caméras

Ajouter une caméra

Nom de la caméra :

Le nom de la caméra doit contenir de 3 à 20 caractères et uniquement des caractères alphanumériques.

Caméras activées (URL : http://localhost/stream.php)

Nom	Token	Supprimer
Batman	7508d78512	

FIGURE 17 - GESTION DES CAMERAS

Gestion des modules

Ajouter un module

Nom du module :

Le nom du module doit contenir de 3 à 20 caractères et uniquement des caractères alphanumériques.

Adresse du module :

L'adresse du module doit contenir 2 caractères et uniquement des caractères hexadécimaux (0 à 9, A à F).

Type de module :

Le type doit correspondre à votre module.

Modules activés

Nom	Type	Adresse	Supprimer
Maison	Courant	aa	
Exterieur	Température	bb	

FIGURE 18 - GESTION DES PANNEAUX

Protocole radio

Pour pouvoir communiquer entre les modules Arduino, il a fallu définir un protocole de communication. Agissant dans un environnement client-serveur, le serveur va donner les ordres en radiofréquence auprès des modules pour effectuer des actions. Les modules Arduino, n'agissent qu'en tant qu'esclave et attendent d'avoir une instruction de commande en provenance du serveur pour renvoyer l'information. La communication s'effectue sous forme de trame. Celle-ci respecte l'architecture détaillée ci-dessous.

Le protocole fonctionne en deux parties :

- La première va permettre de définir les adresses de l'émetteur et du récepteur.

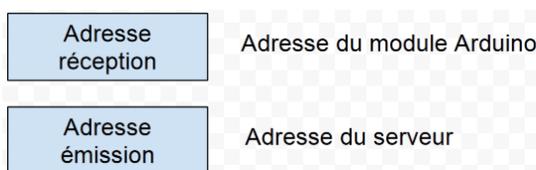


FIGURE 19 - ADRESSES DANS LE PROTOCOLE RADIO

- La seconde en quatre niveaux va permettre de spécifier les attentes que nous avons vis-à-vis du module. Ces attentes sont définies par un système de « mots-clés ».
 - Niveau 1 : Va définir l'action à réaliser par le module
 - Mots-clés de ce niveau :
 - **GET** : Récupération d'une information, soit l'état d'une broche soit les informations d'un capteur.
 - **SET** : Changement d'état d'une broche.
 - Niveau 2 : Indique le type de module que nous allons interroger
 - Mots-clés de ce niveau :
 - **TMP** : Capteur de température.
 - **CUR** : Capteur de courant.
 - Niveau 3 : Indique l'instruction voulue sur le module
 - Mots-clés de ce niveau :
 - **INF** : Récupération de l'information.
 - **ON** : Mise d'une broche à 1.
 - **OFF** : Mise d'une broche à 0.
 - Niveau 4 : Précision sur les numéros des modules à interroger ou le type d'informations que l'on souhaite obtenir (exemple : degrés Celsius ou Fahrenheit pour la température).
 - Mots-clés de ce niveau :
 - **ALL** : Action à effectuer sur toutes les broches.
 - **Numéro_des_broches** : On peut spécifier le numéro des composants qui vont subir une action. Ce mot-clé est utilisé dans le contrôle du relai de courant. Un exemple de trame à ce niveau serait **1,2,3**.
 - **CEL** : Demande de la température en degrés Celsius.
 - **FAH** : Demande de la température en degrés Fahrenheit.

Le fonctionnement du protocole en situation réelle, sera expliqué dans le paragraphe suivant.

Commande manuelle

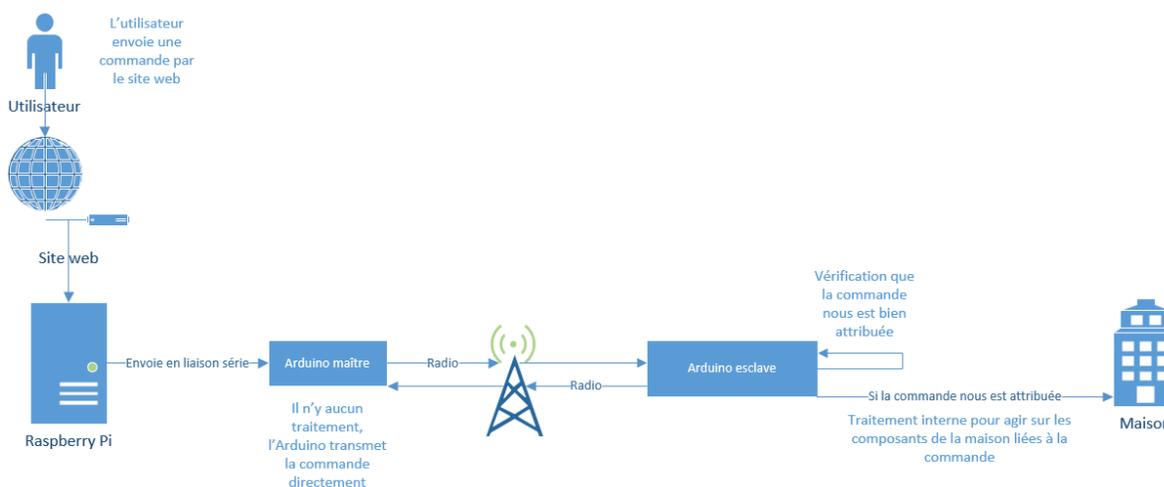


FIGURE 20 - COMMANDE MANUELLE

Lorsque que le serveur souhaite obtenir ou faire une action auprès d'un module, il va envoyer une trame. Le Raspberry Pi va créer la trame en interne, en fonction des scénarios configurés ou de la commande désirée via le site web par l'utilisateur. Le serveur va transmettre la commande à l'Arduino maître qui va effectuer une diffusion en radiofréquences. Si l'Arduino maître ne reçoit pas de réponse après 2 secondes suivant l'émission de la trame, il considère que le récepteur ne l'a pas reçue.

Dans l'Arduino esclave, il y a deux étapes bien distinctes. La première consiste à déterminer si la commande reçue est bien pour lui. Pour cela, on se base sur l'adresse de réception présente dans la trame. Dans un second temps, une fois que nous sommes sûrs de devoir traiter la trame, nous devons découper la trame afin de déterminer les 4 couches du protocole.

Par un système de comparaison de mot-clé, l'Arduino va savoir au fil des couches ce qu'elle doit effectuer et quel composant sera affecté par un changement. Si tous les mots-clés sont reconnus, on renvoie la trame initiale (en intervertissant les adresses, car l'adresse de réception devient l'adresse serveur) auquel on ajoute soit la confirmation de l'action soit la donnée qui nous a été demandée.

Au cas où il y aurait une erreur lors de l'identification des mots-clés, on arrête le traitement à la couche où l'on a détecté l'erreur et on renvoie la trame initiale auquel on ajoute un message d'erreur.

Par exemple, si le serveur à l'adresse « 0x01 » souhaite connaître la température du module nommé « Température_Chambre » à l'adresse « 0x10 », il peut envoyer la trame :

1001GET|TMP|INF|CEL

Et le module lui répondrait la trame :

0110GET|TMP|INF|CEL|22.12

Gestion des scénarios

Les scénarios vont permettre à l'utilisateur de définir des plages horaires auxquels des évènements particuliers vont être effectués. Par exemple, éteindre la lumière du salon tous les jours à 23h00.

Pour pouvoir effectuer une gestion des scénarios au sein de notre solution, nous allons nous servir du démon cron présent sur les systèmes Unix. Pour interagir avec celui-ci, nous allons utiliser le fichier crontab présent nativement sur ces systèmes.

Présentation de cron et de crontab

Cron est un démon utilisé pour programmer des tâches automatiques devant être exécutées à des moments précis. Chaque utilisateur du système Unix possède son propre fichier crontab, lui permettant d'indiquer les actions qui devront être faites par cron.

Fonctionnement de crontab

Le fichier crontab contient l'ensemble des tâches que devra effectuer le démon cron. Pour ce faire, il faut respecter une syntaxe définie et remplir les champs requis. Ci-dessous, un extrait du document officiel de crontab.

Une ligne du fichier correspond à une tâche, qui contient les champs suivants dans cet ordre précis :

minute heure jour_du_mois mois jour_de_la_semaine commande

Exemple du fichier crontab présent dans « /etc/ » de base :

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
```

FIGURE 21 - FICHER CRONTAB

Champs		Valeurs autorisées	
Minute	0-59	-	* / , -
Heure	0-23	-	* / , -
Jour du mois	0-31	-	* / , -
Mois	0-12	3 premières lettres du mois en anglais JAN-DEC	* / , -
Jour de la semaine	0-7	3 premières lettres du jour en anglais SUN-SAT	* / , -
Commande	On indique la commande Unix que l'on souhaite exécuter		

Il est important pour la compréhension des lignes, de saisir le fonctionnement des caractères spéciaux énoncés en troisième colonne des valeurs autorisées.

Astérisque (*)

Ce symbole permet de notifier que l'action devra être effectuée toutes les minutes/heures/etc. dépendant du champ.

Slash (/)

Permet d'effectuer une incrémentation définit d'un intervalle de temps. Il permet de spécifier, par exemple, avec la structure */10 (pour les minutes) que l'évènement sera exécuté avec un intervalle de 10 minutes à partir de la minute 0.

Virgule (,)

Permet de séparer des éléments d'une liste, notamment pour les mois et jours de la semaine dans la langue anglaise. Par exemple, « JAN, FEB, MAR » ou encore « MON, WED, FRI ».

Tiret (-)

Définit un intervalle de temps sur le champ. Par exemple, si on définit pour le mois 1-4, on définit inclusive entre janvier et avril.

Fonctionnement logiciel de la gestion des scénarios

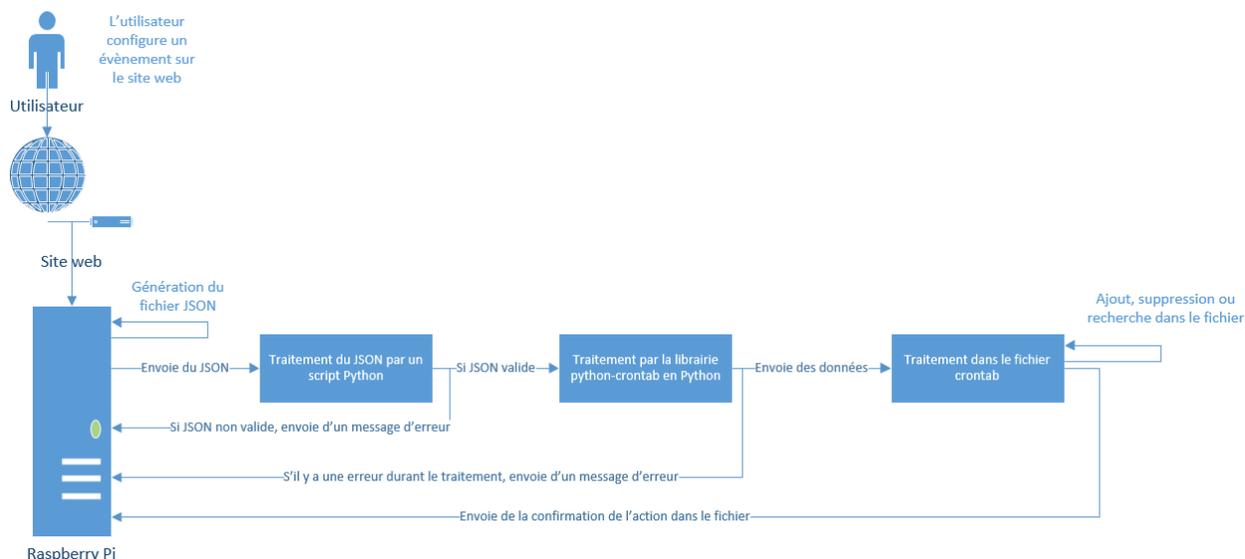


FIGURE 22 - FONCTIONNEMENT LOGICIEL DE LA GESTION DES SCENARIOS

Lorsque l'utilisateur souhaite ajouter un événement, il doit remplir un formulaire dans le site web. Une fois celui-ci rempli, le serveur génère un fichier JSON. Il existe trois modèles de JSON différents, dans le but d'optimiser le traitement par les scripts Python :

- Pour la création
- Pour la suppression
- Pour la recherche

Une fois créée le JSON est envoyé à un script Python qui va être en charge de définir si les paramètres du fichier sont cohérents avec l'action demandée. Si c'est le cas, le script formalise les données pour la librairie « python-crontab » auquel cas on renvoie un message d'erreur. Par la suite, « python-crontab » agit en « boîte noire » et effectue les actions auprès du fichier crontab présent sur l'ensemble des systèmes Unix présenté précédemment.

Tout au long du processus les messages d'erreurs envoyés au serveur sont au format JSON.

Caméra - Configuration et paramétrage

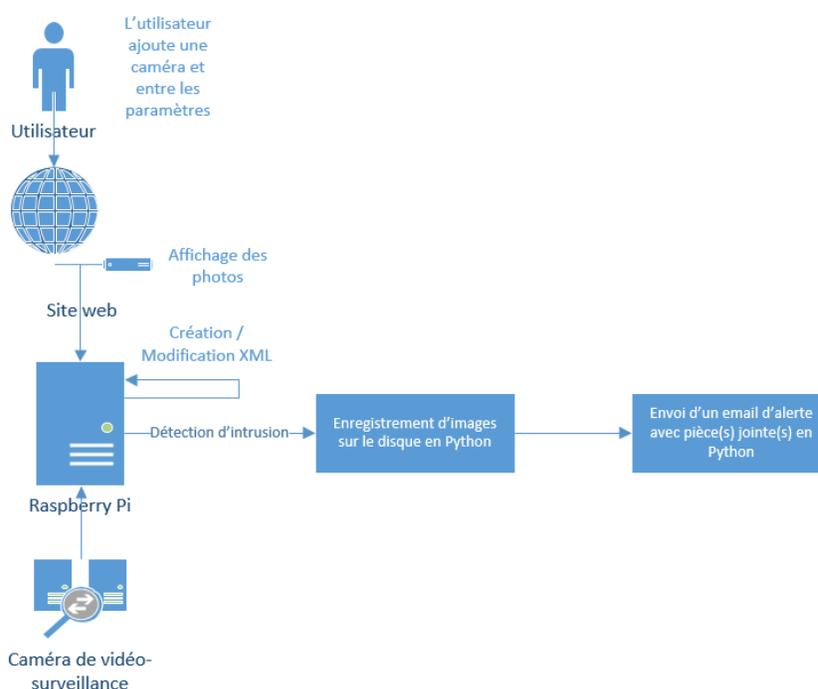


FIGURE 23 - CONFIGURATION ET PARAMETRAGE DE LA CAMERA

Dans l'optique d'avoir un produit le plus modulable possible, nous laissons à l'utilisateur l'opportunité de régler la caméra de surveillance. Pour ce faire, nous avons implémenté une interface web en html, dans laquelle nous renseignons les paramètres de la caméra : nom, résolution, sens de rotation, nombres d'images à enregistrer, seuil de détection, etc.

Ensuite le serveur récupère les données grâce à une interface CGI, le programme vérifie que le contenu est correct avant de créer un fichier XML.

Pour faire la validation des données, nous utilisons des expressions régulières (motifs décrivant un ensemble de caractères possibles) pour chacun des champs.

Lors de la première mise en service, l'utilisateur devra indiquer tous les paramètres et de ce fait compléter le fichier XML. Il est possible, par la suite, de mettre à jour le fichier.

Un second script traite le fichier XML en extrayant les données et en les envoyant à la caméra.

Afin de garantir la sécurité de l'utilisateur, nous avons développé une solution de surveillance grâce à une caméra de surveillance. Lorsqu'un mouvement survient, des photographies sont prises, stockées en local, puis envoyées par email en pièce jointe à l'utilisateur en guise de notification.

Le script de détection fonctionne de la manière suivante : les images sont tout d'abord converties en noir et blanc, puis on effectue une différence de seuil entre deux images consécutives. S'il y a une trop grande différence entre les deux images, le mode de détection se déclenche.

V. BILAN DU PROJET

1. Apports individuels et collectifs

Paul Mégueule

Lors de ce projet, j'ai eu l'occasion de travailler en équipe ainsi que de d'utiliser de nombreux langages de programmations. Ce sont des points importants, car j'ai pu élargir mes connaissances dans le domaine technique, à la fois en effectuant mes tâches individuelles mais aussi grâce à l'échange dans le groupe.

De plus, j'ai pu remarquer que la gestion d'un projet d'une telle envergure n'est pas évidente et nécessite donc un suivi constant et sérieux.

Le fait d'être amené à développer une telle application de A à Z est représentatif du travail que l'on pourrait réaliser en entreprise. C'est donc un exercice très enrichissant.

Fabien Noir

Ayant découvert la domotique lors d'un stage pour le Solar Décathlon, il m'est paru pertinent de proposer à mes camarades de participer au projet DAIO. N'ayant que rarement eu la possibilité d'encadrer un projet, l'échange fut le maître mot au sein du groupe. Chacun ayant ses propres idées et concepts, il a fallu prendre des décisions importantes pour finir le projet à temps. De plus, l'apport dans le domaine technique et intellectuel fut très enrichissant.

Nicolas Roulleau

En adéquation avec mes centres d'intérêt, l'idée de créer un système domotique à immédiatement piquée ma curiosité. Ce projet est une synthèse en lui-même, tant par sa structure pluridisciplinaire que par les capacités et les aptitudes qu'il exige. Cette complexité technique nous pousse rapidement à travailler sur la gestion de projet et les relations humaines, afin d'organiser les hommes et leurs idées. Pour la première fois, il m'est apparu évident que le management de l'équipe était bien plus important que le développement et la production de résultats. La gestion de projet prend alors tout son sens, dans un florilège de connaissances.

Florian Roussel

Ce projet m'a permis de mettre en corrélation un grand nombre de technologies vues en cours et en dehors. Leurs associations permettent d'aboutir à un système fiable et qui répond aux attentes que nous nous étions fixées en début de projet. Le fait de concilier différents langages peut faire peur au début, mais en se fixant des objectifs par fonctionnalités, j'ai réussi à prendre conscience de l'importance de chacune dans le projet. Au-delà de l'aspect technique, j'ai pu remarquer que l'aspect humain et comme à chaque fois dans un projet, la part la plus importante. Malgré quelques désaccords sur le début du projet, nous avons réussi à accorder nos envies pour aboutir à une solution de qualité qui satisfait l'ensemble des membres.

2. Conclusion générale

Cette année, nous avons pris les devants en proposant notre propre vision de la domotique. De notre réflexion est né un projet pour le moins enrichissant et formateur. Pour la première fois, nous avons mené une étude du papier à la mise en service du produit. En imaginant ce projet, nous avons en tête quelques bases et méthodes scolaires concernant l'élaboration d'une solution domotique, mais nous ne connaissons ni les processus en amont, ni les processus en aval. En maîtrisant le projet dans sa globalité, nous avons eu un aperçu complet du processus de fabrication d'un système domotique, des compétences et des connaissances qui y sont attachées.

D'autre part, la domotique a cette particularité de rassembler divers corps de métiers : programmation, informatique, électronique, électrotechnique, et bien d'autres. Cette pluridisciplinarité fut un obstacle que nous avons surmonté grâce aux compétences multiples et à la polyvalence de notre équipe. Cette plurivalence a notamment été mise à profit dans l'attribution des tâches afin que chacun tire parti de ses accomplissements.

Enfin, les temps impartis à la réalisation du système furent bref et il a fallu faire preuve de flexibilité et de persévérance, parfois pour respecter les délais, parfois pour respecter les contraintes technologiques imposées par le projet. Somme toute, nous avons retrouvé lors de ces quatre mois, les compétences, les contraintes mais aussi l'excitation d'un projet d'entreprise.

VI. PERSPECTIVES

Ce projet est avant tout un POC (Proof Of Concept), c'est-à-dire qu'il nous a permis d'affirmer qu'aujourd'hui, il est possible de faire de la domotique de qualité à moindre coût au prix de quelques efforts.

Il conviendra, cependant, d'élargir le nombre de modules, en gardant à l'esprit que les modules sont passifs et parfois gourmands en énergie.

On pensera par exemple à :

- Une gestion de chauffage
- Une station météorologique
- Un système d'irrigation pour le jardin
- Une gestion de l'état de chaque ouverture (portes et fenêtres)
- Un système de diffusion sonore intelligent
- Un système d'alerte incendie

Les seules limitations restent l'imagination et les connaissances techniques.

VII. BIBLIOGRAPHIE

Bldr. **How's the weather? TMP102 + Arduino.** [en ligne]

Disponible sur : <http://bldr.org/2011/01/tmp102-arduino/> (Consulté le 13/04/2014)

JSONLint. **JSONLint - The JSON Validator.** [en ligne]

Disponible sur : <http://jsonlint.com/> (Consulté le 02/04/2014)

PHPDoc. **phpDocumentor.** [en ligne]

Disponible sur : www.phpdoc.org/ (Consulté le 02/04/2014)

Python. **Documentation for picamera -- Picamera 1.2 documentation.** [en ligne]

Disponible sur : <http://picamera.readthedocs.org/en/release-1.2/> (Consulté le 02/04/2014)

Python. **lxml 3.3.3.** [en ligne]

Disponible sur : <https://pypi.python.org/pypi/lxml/3.3.3> (Consulté le 02/04/2014)

Python. **Python 3.4.0 documentation.** [en ligne]

Disponible sur : <https://docs.python.org/3/> (Consulté le 02/04/2014)

Python. **python-crontab 1.7.2.** [en ligne]

Disponible sur : <https://pypi.python.org/pypi/python-crontab> (Consulté le 02/04/2014)

VirtualWire. **VirtualWire library for Arduino and other boards.** [en ligne]

Disponible sur : <http://www.airspayce.com/mikem/arduino/VirtualWire/> (Consulté le 02/04/2014)

Wikipédia. **cron - Wikipedia, the free encyclopedia.** [en ligne]

Disponible sur : <http://en.wikipedia.org/wiki/Cron> (Consulté le 02/04/2014)

PROJET DOMOTIQUE

DAIO

« Domotique All-In-One »

Projet réalisé par : **Paul Mégueule - Fabien Noir - Nicolas Roulleau - Florian Roussel**

Projet encadré par : **Bertrand Cottenceau - Mehdi Lhommeau**

Résumé

Ce rapport relate la conception du système domotique **DAIO** "Domotique All In One" ayant pour but l'**automatisation** d'un habitat pour plus de confort, de sécurité et d'**assistance**. Cette étude fut conduite dans le cadre du projet tuteuré EI4 2013-2014 à l'**ISTIA** d'Angers. La création d'une solution domotique pour le particulier s'est révélée être indispensable pour combler l'insatisfaction du public en quête de services et de fonctions facilitant sa vie quotidienne à domicile. Par l'intermédiaire d'une interface web, l'utilisateur gère la **supervision** de l'édifice, en générant des requêtes depuis un serveur **Raspberry Pi** vers des modules **Arduino** distants. Une fois configuré, le système est capable d'interagir avec l'**environnement** de façon autonome, de prendre des initiatives et de garantir le bien-être et la quiétude exigés par l'utilisateur.

MOTS-CLÉS: DAIO, automatisation, assistance, ISTIA, supervision, Raspberry Pi, Arduino, environnement

Abstract

This paper reports the design of the home automation system **DAIO** "Domotique All In One" aimed at **automating** a habitat for comfort, security and **assistance**. This study was conducted within the tutored project EI4 2013-2014 at the **ISTIA** of Angers. Creating a home automation solution for the particular turned out to be essential to make up for the lack of satisfaction of the public seeking for services and features facilitating their daily life at home. Through a web interface, the user manages the **supervision** of the house, by generating queries from a **Raspberry Pi** server to remote **Arduino** modules. Once set up, the system is able to interact autonomously with the **environment**, to take initiatives and ensure the well-being and the tranquility required by the user.

KEYWORDS : DAIO, automating, assistance, ISTIA, supervision, Raspberry Pi, Arduino, environment