

Le Thermostat Intelligent

S. Eppe, S. Vansummeren, A. Vermeir

Version 1.0 - October 7, 2015

1 Introduction

Le thermostat de chauffage central est l'un des contrôleurs à boucle de rétroaction les plus anciens et simples. Dans sa forme originale, il mesure la température actuelle de la pièce chauffée. Si la température est en dessous d'un seuil désiré (appelé la valeur de consigne), le thermostat enclenche le chauffage central. La chaudière chauffe alors l'eau qui circule dans les radiateurs du bâtiment. Les radiateurs, à leur tour, réchauffent les pièces dans lesquelles ils ont été installés. Lorsque le thermostat observe que la température de consigne est atteinte, il éteint alors la chaudière. Evidemment, on doit prendre soin d'éviter tout dépassement de la température de consigne ou l'oscillation de température autour de cette même température de consigne. Dans le cadre de ce projet, votre objectif sera la réalisation d'un thermostat **multi-zones et intelligent**.

Multi-zones... Un thermostat traditionnel est généralement installé dans une pièce centrale dont la température est la plus représentative pour le bâtiment entier. Ceci implique que, lorsque la température de consigne est atteinte pour cette pièce, la chaudière s'éteigne. Cependant, les pièces voisines n'ayant pas atteint la température consigne resteront donc plus froides. L'utilisation de communications sans fil permet de dépasser cette limitation. En effet: on peut diviser un bâtiment en plusieurs zones thermiques, et équiper chaque zone d'un "thermostat individuel" ; un microcontrôleur équipé de capteurs de présence et de température, et ayant la capacité d'ouvrir ou de fermer la vanne thermostatique des radiateurs présents dans sa zone. Les thermostats individuels communiquent avec le thermostat central, qui contrôle la chaudière. Dans les zones où la température de consigne est atteinte, les vannes sont fermées pour éviter toute surchauffe. Lorsque chaque zone atteint la température désirée (ou lorsque le contrôleur observe qu'étant donné la température de l'eau, le résultat souhaité peut être facilement atteint), la chaudière s'éteint. La figure 1 illustre ce principe.

... et intelligent Les thermostats plus récents tiennent compte de paramètres additionnels, tels que la température extérieure ou l'inertie thermique du bâtiment, pour réduire significativement la consommation d'énergie de chauffage. La nouvelle génération de thermostats, les "smart thermostats" s'inscrivent dans cette approche et proposent des options telles que :

- Le contrôle du thermostat à distance via le World Wide Web.
- La détection de la présence des occupants du bâtiment (soit par capteur infrarouge, ou par géolocalisation de smart-phones), abaissant automatiquement la température ciblée si le bâtiment n'est pas occupé et/ou préchauffant le bâtiment en anticipant la présence de ses occupants (par exemple, via notification explicite de l'utilisateur, sa géolocalisation ou par l'apprentissage de ces habitudes).

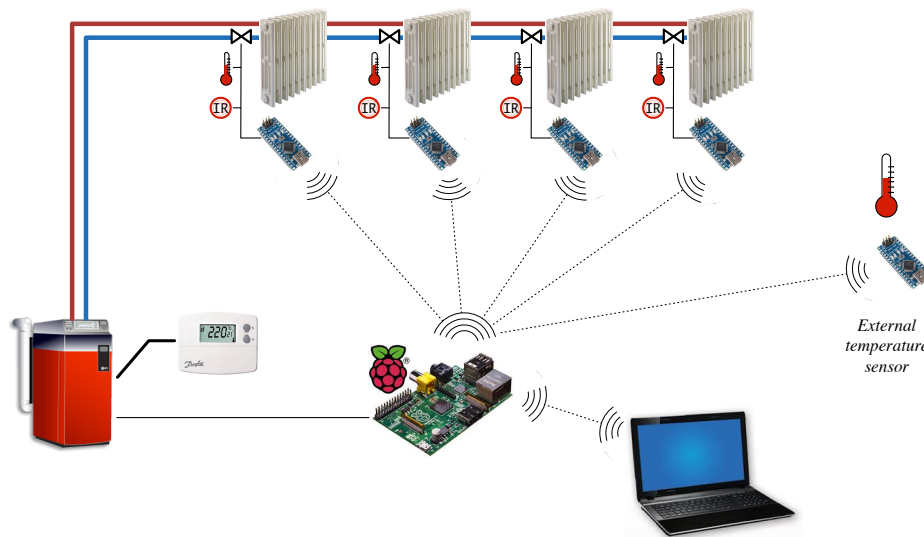


Figure 1: Structure général d'un thermostat multi-zone.

Pour une présentation plus approfondie des “smart thermostats” disponibles dans le commerce, le lecteur peut s'en référer à l'Annexe A ¹.

2 Objectifs et exigences

2.1 Prototypes

L'objectif de ce projet est de concevoir et d'implémenter un système multi-zones et intelligent de thermostats. Pour ce faire, deux prototypes doivent être développés.

- Les **thermostats individuels**. Chaque thermostat individuel couvre une zone et est doté de deux capteurs et d'un actuateur:
 - Un capteur de température mesurant la température de la zone.
 - Un capteur infrarouge détectant la présence de personnes dans la zone.
 - Un actuateur permettant de commander l'ouverture ou la fermeture de la vanne thermostatique.

Les mesures faites par les capteurs sont transmises par le thermostat individuel au thermostat central. Les commandes d'ouverture/fermeture de la vanne sont envoyées par l'unité centrale et reçues par le thermostat individuel. Toutes les communications se font par une communication sans fil. Dans le cadre du projet, seul un prototype de thermostat individuel sera à réaliser. Par ailleurs, pour simplifier le montage, la vanne thermostatique sera connectée à une résistance thermique qui simulera un système de chauffage central (chaudière, tuyaux de transfert d'eau chaude, radiateurs). Cette structure simplifiée est illustrée dans la Figure 2.

- Le **thermostat central**. Il communique sans fil avec les thermostats individuels, agrégeant les lectures de leurs capteurs. Le thermostat central est pourvu de l'algorithme de contrôle, qui est responsable de deux tâches : (1) commander les vannes thermostatiques des zones où

¹Notez qu'en 2014, Google a acquis Nest Labs, producteur de l'un des tout premiers thermostats “smart”, pour plus de 3 milliards de dollars.

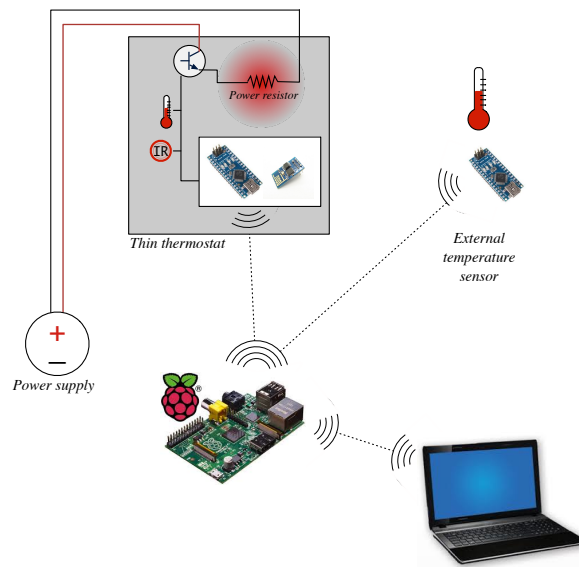


Figure 2: Configuration simplifiée

le chauffage est nécessaire ; et (2) déterminer dans quelles zones le chauffage est (ou sera) requis. Nous détaillons ces deux tâches ci-dessous.

1. Pour commander individuellement la vanne thermostatique de chaque zone de chauffage, l'algorithme doit avant tout déterminer, sur base des trois températures (courante dans la zone, de consigne pour la zone et la température extérieure) le niveau d'actuation de la vanne (entre 0%, c'est-à-dire fermée, et 100%, entièrement ouverte). Le thermostat central communique ensuite le niveau d'actuation calculé à chaque thermostat individuel, qui met à jour l'ouverture de la vanne. Vous devrez développer un modèle qui permet l'actuation des vannes au niveau désiré. L'algorithme devra également empêcher la surchauffe (significative) et des oscillations (significatives) de la température autour de la température de consigne.
2. L'algorithme de contrôle déterminera également les moments pour d'activation et de désactivation est également responsable des décisions liées à la nécessité de chauffer une zone. Nous discernons deux niveaux de détection:

Détection réactive : le contrôleur utilise uniquement le capteur de et que la température actuelle est inférieure à la température de consigne.

Détection intelligente , conférant le caractère "smart" au thermostat: en utilisant le World Wide Web, l'utilisateur peut notifier l'unité centrale qu'il ou elle sera rentré chez lui/elle dans x minutes et que, d'ici-là, la température de consigne dans les zones Y et Z devra être atteinte. En utilisant cette information, l'unité centrale peut anticiper et déterminer sa stratégie de chauffage optimale.

Au cours du projet, vous serez d'abord amenés à implémenter un algorithme de contrôle basé sur une détection réactive. Dans un second temps, vous implémenterez l'algorithme de la détection intelligente.

Dans le cadre du debugging et de l'évaluation de votre algorithme, il est nécessaire que l'état du système entier puisse être suivi en externe. Pour cela, l'unité centrale devra proposer un tableau de bord consultable depuis un navigateur web.

Veuillez observer la présence d'un appareil complémentaire dans le schéma, qui communique avec le thermostat central : il s'agit du **thermostat externe**. Son rôle est de capter la tempéra-

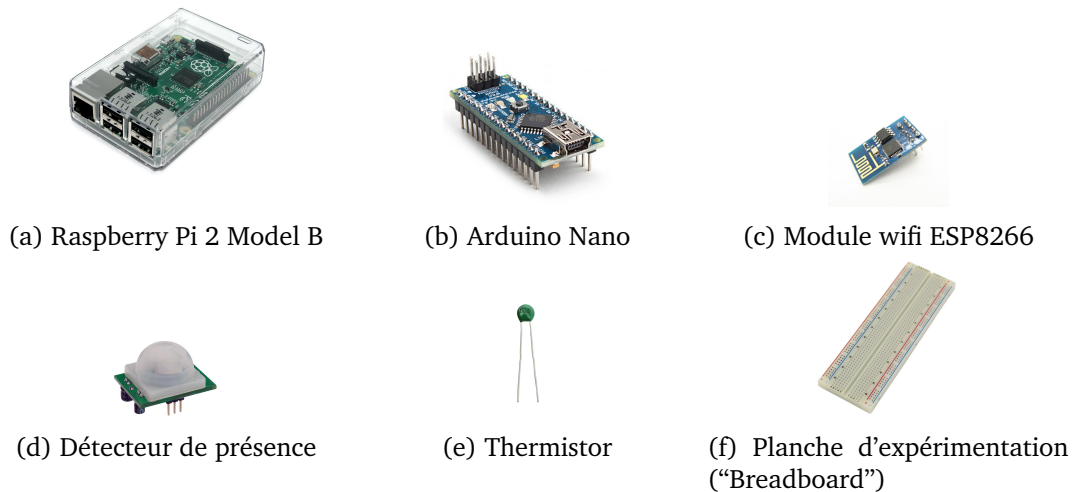


Figure 3: Eléments matériels mis à votre disposition.

ture extérieure et de la communiquer au thermostat central.

2.2 Protocole de communication

Les thermostats individuels et le thermostat central communiqueront par WIFI en utilisant le protocole de communication (basé sur HTTP [8]) décrit en Annexe B. Il vous sera demandé d'implémenter ce protocole par vous-même et de vous conformer strictement à ses spécifications.

2.3 Matériel fourni

Une large gamme de plateformes matérielles et de microcontrôleurs sont envisageable pour l'implémentation des thermostats individuels et du thermostat central. Par expérience, cependant, leur configuration peut se révéler chronophage. Pour éviter toute perte de temps dans la résolution d'erreurs de configuration, nous vous fournissons un matériel présélectionné et configuré (voir Figure 3). Vous ne devrez donc pas vous procurer du matériel additionnel.

- Le thermostat central doit être implémenté sur un Raspberry Pi 2 modèle B [6] (un “Pi” pour faire court). Le Raspberry Pi est un ordinateur bon marché, de la taille d'une carte de crédit. Bien que compact, il constitue un ordinateur complet avec mémoire principale, processeur et une mini carte SD, jouant le rôle du disque dur. Votre Pi fonctionnera sous le système d'exploitation Linux; aura le langage de programmation Python pré-installé, pour communiquer avec les thermostats individuels. L'algorithme de contrôle central devra donc être implémenté en Python sur le Pi.
- Le thermostat individuel sera implémenté en utilisant un microcontrôleur Arduino Nano [2]. Pour la communication sans fil, un module WIFI esp8266 vous sera fourni (cf. Figure 3b). L'Arduino est généralement programmé en utilisant un langage propriétaire appelé Arduino Programming Language [3], qui est une variante du langage C++. L'apprentissage autonome de ce langage ainsi que du traitement des signaux d'entrée (signaux provenant des capteurs) et de sortie d'un Arduino, font partie du projet.

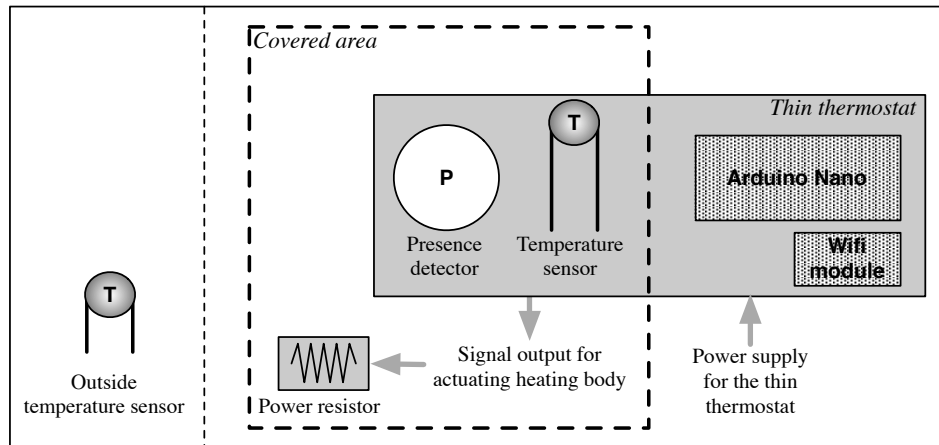


Figure 4: Représentation schématique de l'environnement physique de validation.

- Enfin, vous recevrez un kit comportant les différents modules et des câbles qui vous permettent de construire un prototype de thermostat individuel. Ce kit contient un thermistor et un capteur infrarouge pour implémenter les capteurs de température et de présence, respectivement. La résistance de puissance n'est pas fournie, car elle fait partie de l'environnement physique de validation (voir supra). Notez cependant, que pour commander cette résistance thermique, l'Arduino devra produire un signal de sortie adéquat sur l'un de ses canaux de sortie.

Tout équipement procuré doit être rendu avant l'évaluation finale du projet.

2.4 Environnement physique

En supplément du matériel mentionné ci-dessus, nous vous fournirons également un **environnement physique**. Ce dernier est constitué d'une boîte (Figure 4) dans laquelle vous pouvez insérer votre thermostat individuel et le tester. Cet environnement a deux objectifs : (1) procurer un système de chauffage simple (en comparaison avec une chaudière domestique) qui fonctionne pour un petit volume à chauffer, et qui vous permettra de tester votre thermomètre individuel dans un environnement simple à contrôler, et pour de courts laps de temps et (2) procurer un capteur additionnel qui communiquera sans fil la température extérieure à votre thermostat central.

Chaque groupe aura l'environnement physique complet à sa disposition approximativement 2 semaines après le début du projet, l'objectif étant que vous vous familiarisiez d'abord avec le "board" de l'Arduino Nano, le module wifi et les capteurs (infrarouges et thermiques).

2.5 Logiciels fournis

Des exemples et des bibliothèques de code, illustrant l'utilisation du module wifi ESP8266 sur un Arduino, ainsi que la programmation d'un serveur web sur le Pi sont téléchargeables sur le dépôt GIT ² du projet, dont l'URL est le suivant:

<http://wit-projects.ulb.ac.be/rhocode/TRAN-H-201/2015-2016/Supplied-Software>.

²<http://git-scm.com/documentation>

Ce répertoire contient également un programme que vous pouvez utiliser pour tester votre implémentation du protocole de communication sur votre thermostat central.

2.6 Résultats attendus

En résumé, les résultats attendus comportent :

- Conception et implémentation (programmation) de l'algorithme de contrôle en python et implémentation du thermostat individuel (assemblage et programmation), y compris l'implémentation du protocole de communication tel que décrit dans l'Annexe B.
- Des indicateurs de qualité. Il vous est demandé de développer un ou plusieurs critères qui vous permettront d'évaluer quantitativement l'efficacité de vos réalisations. Ces critères devraient également vous permettre de classer les différentes solutions considérées et/ou développées au cours du projet. Quelques exemples : une mesure de la surchauffe, de la vitesse à laquelle la température de consigne est atteinte, une mesure de l'énergie consommée, etc.
- Stratégie de test. Développez (et décrivez dans le rapport) une stratégie de test pour vous assurer du bon fonctionnement de vos réalisations, tant au niveau de l'unité que du système intégré.
- Travail de groupe. Votre évaluation finale tient compte de la qualité de votre travail d'équipe (répartition des tâches, mise en commun, entraide, communication de groupe, tant au niveau collaboratif que dans la résolution de conflits, etc.).
- Instructions de montage du thermostat individuel. Ces instructions doivent entièrement spécifier l'interface physique du thermostat individuel, de telle sorte qu'il puisse être utilisé par autrui. En particulier, elles doivent comprendre le plan de câblage du thermostat individuel aux capteurs et à la résistance thermique.

Nous attirons votre attention sur certains points :

- Même si vous ne réaliserez qu'un seul prototype de thermostat individuel, votre thermostat central doit avoir la capacité de gérer un nombre indéfini de thermostats individuels.
- La structure de votre algorithme de contrôle doit lui permettre d'être fonctionnel dans un environnement (de validation physique) ayant des caractéristiques thermiques différentes (par exemple, un environnement fortement isolé). En d'autres mots: ne "hard-codez" pas les caractéristiques thermiques de l'environnement physique que nous vous procurons. Dans votre rapport final, nous vous demandons d'inclure une explication de la manière dont votre système peut s'adapter à un environnement ayant des caractéristiques thermiques différentes.
- Etant donné la complexité des équations (l'équation de la chaleur est une équation aux dérivées partielles dont la résolution, mêmes par des méthodes numériques, dépasse le cadre de ce projet), nous vous suggérons fortement de considérer le système (la pièce/zone à chauffer) comme une boîte noire dont vous ignorez la loi de comportement thermique.

3 Délivrables

Pour l'évaluation à mi-parcours. Vous devrez avoir produit un rapport intermédiaire, incluant une bibliographie détaillée. Pour permettre une évaluation par tous les membres du jury, votre rapport intermédiaire devra inclure une page qui liste les membres du groupe (nom, matricule d'étudiant, e-mail, photo). Vous présenterez l'état d'avancement de votre projet orallement, y compris l'état de vos prototypes.

Pour l'évaluation finale. Les livrables sont :

- Le rapport final (incluant une bibliographie détaillée, et une page mentionnant les membres du groupe);
- les prototypes fonctionnels (thermostats central et individuel);
- le code produit pour les deux prototypes;
- les instructions de montage du thermostat individuel; et
- une présentation orale du projet devant le jury.

Vous devrez aussi restituer le matériel d'implémentation qui vous a été fourni.

Remise des livrables. Vous créerez un dépôt GIT³ dans le groupe TRAN-H-201/2015-2016 que vous trouverez sous l'URL

`http://wit-projects.ulb.ac.be/rhodecode/TRAN-H-201/2015-2016`

pour y envoyer vos rapport, code, et les instructions de montage au jury. Le nom d'utilisateur et le mot de passe pour ce système correspondent à votre NetID ULB/VUB. Le répertoire doit être libellé `group<X>`, où `<X>` est votre numéro de groupe. Ce répertoire doit être rendu privé. Il vous est recommandé de créer ce répertoire *le plus vite possible* pour éviter les difficultés techniques de dernière minute, et de l'utiliser en tant que système de contrôle de version et de synchronisation des changements tout au long du projet.

Astuce. Si vous tentez d'envoyer ("push") un large volume de données à un répertoire GIT via HTTP ou HTTPS, il est possible que vous soyez confrontés au message d'erreur suivant: `error: RPC failed; result=22, HTTP code = 411..` Celui-ci est vraisemblablement causé par la configuration GIT qui limite certaines opérations HTTP à 1 Mega octet. Pour changer cette limite dans votre répertoire local :

```
git config http.postBuffer *bytes*
```

où `*bytes*` est le nombre maximum de bytes permis par push.

4 Milestones du projet et critères d'évaluation

Le projet est découpé en 4 étapes ("milestones") décrites ci-dessous. Chaque étape sera évaluée sur 5 points, constituant au total votre évaluation sur 20. Si vous complétez les étapes 1 et 3 seulement, vous obtiendrez une cote maximale de 10/20. Pour chaque étape, la cote obtenue dépendra du ou des prototypes développées, ainsi que de la manière dont vous avez pu travailler en groupe, gérer le projet, validé votre approche via des critères de qualités pertinents, et communiquer votre bonne compréhension de la problématique et de votre travail.

Notez que certaines étapes peuvent être développées en parallèle, alors que d'autres ont en prérequis la réalisation d'autres parties du projet. Vous êtes bien entendu libre dans le choix de votre organisation quant à la réalisation de livrables. Cependant, dans les 3 semaines suivant le début du projet, vous devrez avoir établi et discuté un plan de travail avec votre tuteur, et donné une description claire de votre approche.

³<http://git-scm.com/documentation>

Milestone 1: Thermostat individuel

- Configuration de l'Arduino, apprendre à programmer avec Sketch
- Câblage de l'Arduino, interfaçage avec les capteurs et actuateurs.
- Implémentation des parties du protocole de communication utiles à la réalisation du thermostat individuel;
- Tests d'unité : la lecture des capteurs est-elle correcte ? La commande de la vanne fonctionne-t-elle ?
- Les instructions de montage du thermostat individuel.

Milestone 2: Thermostat central avec algorithme de contrôle réactif.

- Configuration du Raspberry Pi;
- Etude du background, modélisation approximative de l'algorithme de contrôle;
- Conception et implémentation de l'algorithme de contrôle basé sur une détection réactive;
- Implémentation des parties du protocole de communication utiles à la réalisation du thermostat central;
- Tests d'unité du protocole de communication sur base du simulateur inclus dans les logiciels fournis.

Milestone 3: Intégration et validation Se base sur les réalisations des étapes 1 et 2, et comporte :

- Intégration complète des thermostats central et individuel;
- Validation entière du système intégré avec l'environnement de validation physique
- Analyse critique et synthèse des performances du système;
- Modifications du thermostat central ou individuel si requis;

Milestone 4: Thermostat central avec algorithme de contrôle "smart"

- Modélisation du comportement anticipatif de l'algorithme de contrôle;
- Modélisations liées au chauffage via système dynamique (déterminer l'instant optimal pour la marche/arrêt de la chaudière);
- Design et implémentation de l'algorithme de contrôle basé sur un algorithme de détection "smart";

References

- [1] Requests: Http for humans. <http://www.python-requests.org/en/latest/>.
- [2] Arduino. Getting started with arduino. <https://www.arduino.cc/en/Guide/HomePage>.
- [3] Arduino. Sketch. <https://www.arduino.cc/en/Tutorial/Sketch>.
- [4] Internet Assigned Numbers Authority. Media types. <http://www.iana.org/assignments/media-types/media-types.xhtml>.
- [5] Bottle: Python web framework. <http://bottlepy.org/docs/dev/index.html>.
- [6] Raspberry Pi Foundation. Raspberry pi - teach, learn and make with raspberry pi. <https://www.raspberrypi.org/>.

- [7] Ecma-404 the json data interchange standard. <http://www.json.org>.
- [8] James Marshall. Http made really easy. <http://www.jmarshall.com/easy/http/>.

A Aperçu de thermostats “smarts” dans le commerce

Un certain nombre de “smarts” thermostats ont déjà été réalisés. Parmi les plus connus :

- le thermostat NEST (<https://nest.com>), qui a été racheté par Google pour plus de 3 milliards de dollars il y a quelques années.
- le Honeywell Lyric (<http://lyric.honeywell.com/>) – Honeywell réalise des thermostats depuis la fin du 19e siècle.
- l’Ecobee 3 (<https://www.ecobee.com>).
- le tado (<https://www.tado.com/gb/heatingcontrol-savings>)

Tous ces thermostats sont connectés par WIFI à internet et peuvent être contrôlés depuis une application sur smartphone ou tablette. De plus:

- le thermostat NEST a des capteurs de mouvements et apprend les habitudes des utilisateurs durant une phase de configuration qui dure une semaine. S’il détecte une absence, il abaissera la température désirée, s’il détecte une présence, il l’augmentera. Si l’arrivée d’un utilisateur est anticipée, il allumera déjà la chaudière. Plusieurs thermostats NEST peuvent communiquer entre eux et s’échanger les lectures de leurs capteurs de mouvements (bien qu’ils aient un agenda automatisé individuel).
- le Honeywell Lyric ne possède pas d’algorithme d’apprentissage (il prétend que la plupart des habitudes des utilisateurs sont trop irrégulières pour pouvoir apporter un plus). Cependant, il supporte la géolocalisation: basé sur la localisation des smartphone couplés (tracés), il baissera la température si l’un de ces derniers est au delà d’un certain rayon, et la remontera lorsque l’un des smartphones se rapproche à nouveau.
- l’Ecobee 3 mesure la température à l’extérieur sur base de sa géolocalisation et de d’informations météorologiques issues du web. Il a également un certain nombre de capteurs externes qui communiquent avec le thermostat central. Ainsi, si vous travaillez dans votre bureau mais que le thermostat est dans le salon, vous aurez quand même du chauffage (La température demandée étant la moyenne entre toutes les zones où la présence est détectée.).
- le tado utilise la géolocalisation pour vérifier si les utilisateurs sont à l’extérieur, vérifie les prévisions météorologiques pour éviter toute surchauffe et apprend de ses erreurs (via boucle de rétroaction).

B Protocole de communication

Le thermostat central communique avec les thermostats individuels et le thermostat extérieur en utilisant le protocole décrit dans cette section. Remarquez qu’une conformité stricte à ce protocole est requise.

B.1 Background général

Toute communication est effectuée en utilisant HTTP [8]. HTTP est le protocole que les navigateurs utilisent pour télécharger des pages web depuis Internet. Il est également utilisé pour beaucoup d’autres applications, avec de communiquer sur le World Wide Web.

HTTP est un protocole de type *request-reply*. Cela signifie qu’une partie, dénommée le *http client* effectue une requête à une autre partie, dénommée le *http server*⁴. Le serveur inspecte la requête et évalue s’il peut fournir une réponse adéquate, puis fournit cette dernière. Pour pouvoir effectuer des requêtes, le client nécessite l’adresse IP du serveur (par exemple, 192.168.10.1) et le port (par exemple, 8080) que le serveur inspecte pour recevoir les requêtes. Le serveur s’exécute généralement en continu, vérifiant s’il y a de nouvelles requêtes auxquelles il devrait répondre.

Techniquement, tant les requêtes que les réponses sont des chaînes de textes formatées selon certaines conventions. Ces conventions ne sont pas importantes dans le cadre de ce projet, car vous utiliserez des librairies (e.g., [5, 1]) existantes prévues à cet effet.

Par contre, il est important de comprendre qu’une requête est composée de plusieurs parties distinctes :

- La **méthode HTTP**, qui spécifie l’action à entreprendre. La table 1 liste et décrit les différentes méthodes valides.
- Le **chemin** sur lequel la méthode doit être appliquée. Par exemple, pour obtenir la page web `https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol`, un navigateur web devra convertir le nom d’hôte `en.wikipedia.org` en une adresse IP, se connecter à cette adresse via un port pré-défini (par défaut : port 80), et faire une requête de méthode GET sur le chemin `path wiki/Hypertext_Transfer_Protocol`.
- Un certain nombre d’**en-têtes** (“Headers”). Une en-tête est une paire (clé, valeur), dont le nom de certaines clés est standardisé. Les en-têtes permettent de fournir plus d’informations sur la manière dont certaines requêtes doivent être traitées.
- Le **corps de la requête** (“Entity body”). correspond aux données envoyées du client au serveur. Le corps est généralement vide pour les requêtes de type GET. Cependant, pour des requêtes PUT ou POST, lorsque le client envoie des données au serveur, le corps contient les données en questions.

Techniquement, le corps de la requête est un chaîne de caractères. L’en-tête `Content-Type` spécifie la manière dont cette chaîne doit être interprétée. Par exemple, une en-tête `Content-Type` ayant la valeur `image/png` précise que le corps doit être interprété comme une image au format PNG. Une en-tête `Content-Type` de valeur `text/plain` spécifie un corps devant être interprété comme du texte. La liste des `Content-Type` officielle peut être trouvée en référence [4].

La réponse du serveur est également structurée en différentes parties.

⁴Dans le contexte d’un navigateur internet, ce dernier est le client, et le serveur auquel il se connecte est le serveur http, souvent appelé serveur web.

Method	Explanation
GET	demande une ressource au serveur (page HTML , données, etc.)
POST	envoie des données au serveur
PUT	crée ou met à jour une ressource sur le serveur
DELETE	efface une ressource
OPTIONS	demande les différentes méthodes HTTP valides pour un chemin donné
HEAD	similaire à GET, mais ne récupère que les en-têtes, pas le corps

Table 1: HTTP methods

- Un **code de statut**. Par exemple, un code de 200 signifie que la requête a été correctement traitée. Un code de 400 signifie au contraire que le client a effectué une requête erronée , et que le serveur ne peut y pourvoir.
- Un certain nombre d'**en-têtes**, ayant la même utilité que pour les requêtes.
- Le **corps**, qui contiendra éventuellement les données envoyées par le serveur au client. Ici aussi, l'en-tête Content-Type spécifie la manière dont le corps doit être traité.

B.2 Le protocole

Lorsqu'un thermostat individuel est sous tension, il se connecte au réseau sans fil dont le SSID est "TRANH201-GXX" (où XX est votre numéro de groupe), et obtient une adresse IP. Votre Raspberry Pi étant configuré en tant qu'hôte du réseau sans fil, il doit être allumé pour que le thermostat individuel puisse s'y connecter. Lorsque le thermostat individuel est connecté et a bien reçu une adresse IP, il doit s'enregistrer auprès du thermostat central. Durant cette phase, le thermostat central agit en tant que serveur HTTP, et le thermostat individuel est le client. Après l'enregistrement, le thermostat central contacte régulièrement les thermostats individuels enregistrés pour obtenir la lecture de leurs capteurs, et éventuellement modifier l'ouverture des vannes. Durant cette période, le thermostat central agit en tant que client HTTP, et les thermostats individuels sont les serveurs.

Le protocole d'enregistrement

Le thermostat central est configuré pour systématiquement obtenir l'adresse IP 192.168.10.1. Le serveur HTTP du thermostat central qui est à l'écoute des requêtes doit être configuré pour écouter le port 8080.

Pour s'enregistrer, les thermostats individuels se connectent au thermostat central à l'adresse IP et au port susmentionnés et exécute une requête PUT sur le chemin /register avec un corps de requête de type application/json [7], et de contenu suivant ce modèle :

```
{ "ip": "192.168.10.51", "port": 9000, "type": "thin" }
```

Ici, les valeurs mentionnées dans les champs ip et port doivent être remplacées par les adresses IP des thermostats individuels, et par les ports auxquels les serveurs HTTP s'exécutent, respectivement. Le thermostat central sauvegarde ces valeurs, pour ultérieurement pouvoir interroger les capteurs via requêtes aux appareils enregistrés.

Le champ type peut prendre deux valeurs distinctes : thin (pour les thermostats individuels) et outside (pour le thermostat extérieur).

Le thermostat central répond aux requêtes d'enregistrement avec un code 200 OK si la requête est correcte, et 400 BAD REQUEST sinon.

Interactions avec les thermostats individuels

A intervalles réguliers, le thermostat central effectuera des requêtes HTTP aux clients enregistrés (aux adresses IP et ports spécifiés à l'étape précédente).

– Demande de la lecture du capteur thermique

- Request: GET /temperature
- Response: 200 OK avec un corps de content-type "text/plain" contenant une valeur, par exemple "298.15". La valeur est exprimée en Celsius.

– Demande de la lecture du capteur de présence

- Request: GET /presence
- Response: 200 OK avec un corps de content-type "text/plain" contenant une valeur booléenne, par exemple "true" (présence détectée) ou "false" (présence non détectée).

– Actuateur de la vanne thermostatique

- Request: PUT /valve avec un corps de content-type "text/plain" contenant une valeur entière comprise entre 0 et 100 inclus. (exprimé en pourcents)
- Response: 400 BAD REQUEST si requête illégale. 200 OK si requête valide.

– Demande de la valeur de l'actuateur

- Request: GET /valve
- Response: 200 OK avec un corps de content-type "text/plain" contenant une valeur entière comprise entre 0 et 100 inclus. (exprimé en pourcents)

Notez que l'appareil mesurant la température externe (thermostat externe) ne peut que mesurer la température ; il répondra 400 BAD REQUEST à toute autre requête.

C Connection au Raspberry Pi

Il y a deux manières de se connecter au Raspberry Pi :

1. Connectez le Pi à un moniteur externe en utilisant le câble HDMI fourni. Ensuite, branchez un clavier et une souris USB.
2. Connectez vous au réseau sans fil ayant comme SSID "TRANH201-GXX" de votre Raspberry Pi, XX étant votre numéro de groupe, ensuite, utilisez SSH ⁵ pour vous connecter à l'adresse IP de votre Pi, c'est à dire 192.168.10.1.

Votre nom d'utilisateur est pi et votre mot de passe est raspberry. Vous pouvez aussi brancher un câble ethernet au port correspondant du Pi, ce dernier essayera automatiquement d'obtenir une adresse IP via DHCP, et de se rendre disponible sur votre réseau local.

⁵En utilisant la commande linux ssh ou le programme sous windows PUTTY (<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>).