

KilnRegulator3.0 (Y1819-S004)

Introduction

Ce projet a pour but d'aider les céramistes dans le processus de cuisson. Notre rôle sera de proposer un système capable de contrôler une cuisson en se greffant aux fours de potiers. Ce genre de système existe déjà mais coûte plusieurs centaines voire milliers d'euros selon les modèles. Proposer un produit bien moins cher est donc l'une de nos priorités.

Les systèmes déjà existants sont fiables mais proposent peu de fonctionnalités. Les possibilités de personnalisation de cuisson étant réduites, nous souhaitons les élargir en proposant une application mobile permettant de créer et de modifier des programmes de cuisson précis. Cette application aura aussi pour but de suivre l'évolution des cuissons au cours du temps.

Cependant, pour convaincre les céramistes, il faudra proposer un système facile à prendre en main ressemblant aux produits qu'ils ont l'habitude de manipuler. Finalement, notre problématique est d'arriver à concevoir un système simple contrôlant une cuisson de poterie, moins coûteux que les produits existants sur le marché et proposant de nouvelles fonctionnalités facilitant le travail des céramistes.

Description de la solution

Pour répondre à la problématique développée précédemment, nous proposerons deux modules qui communiqueront ensemble : une application mobile et un système embarqué (voir Fig. 1).



Figure 1 : Schéma représentatif des éléments majeurs intervenant dans le projet

Personas :

- Ludovic, 36 ans, est potier depuis que son père lui a légué son atelier. En plus de son lieu de travail, son père lui a laissé son savoir faire et une extrême minutie pour son travail. Ludovic aime donc contrôler avec précision le processus de création de ses œuvres. Il utilise un nouveau four qui lui a coûté cher, mais qui lui permet d'obtenir une fiabilité très élevée. Fan de technologie, il aimerait utiliser son téléphone pour contrôler son four, tout en gardant la fiabilité qu'il estime plus que

nécessaire. Il souhaite aussi que le système agisse automatiquement ou au moins l'avertisse lors d'une anomalie (par exemple, lors d'une coupure de courant).

- Amélie, 56 ans, céramiste depuis son plus jeune âge, maîtrise largement son domaine. Son atelier fonctionne très bien, surtout grâce aux prix qui battent la concurrence. En fait, Amélie fait ce travail par passion et l'argent ne l'intéresse pas plus que ça. Par contre, son matériel coûte très cher, et elle n'ose pas le remplacer de peur qu'elle soit obligée d'augmenter ses prix. Elle aurait donc besoin d'un système de gestion de cuisson pas cher, voire même réutilisable pour ses différents fours.
- Albert, 44 ans, est un artisan touche-à-tout. Il travaille avec une petite équipe pour fabriquer de nombreux types d'objets en manipulant plusieurs matériaux. Il vend ses œuvres aux marchés de sa ville ou à des particuliers sur Internet. Albert est expérimenté dans la céramique, mais cela lui arrive de faire des erreurs. Il oublie souvent qu'il a lancé une cuisson, et ne vérifie pas assez son état. Il aimerait donc un outil afin de suivre l'évolution de sa cuisson quand il est occupé ailleurs.

Application mobile

Scénarios d'usage

Scénario 1 :

- 1) Lancer l'application
- 2) Connecter l'application au four
- 3) Choisir le programme numéro 3 correspondant à la porcelaine
- 4) Démarrer la cuisson
- 5) Après 3 heures de cuisson, vérifier la température actuelle du four
- 6) Être alerté de la fin de cuisson

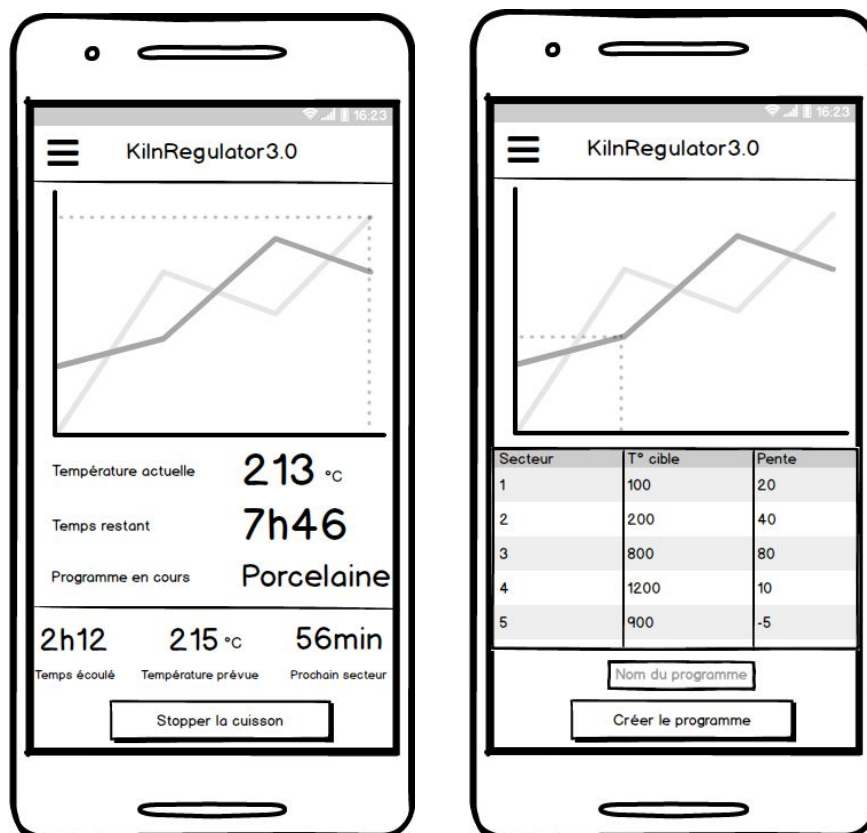
Scénario 2 :

- 1) Lancer l'application
- 2) Connecter l'application au four
- 3) Créer un nouveau programme
- 4) Identifier les segments du programme
- 5) Enregistrer le programme
- 6) Démarrer le programme créé
- 7) Après 2 heures de cuisson, être alerté qu'un problème est survenu
- 8) Stopper la cuisson
- 9) Consulter l'évolution de la température pour identifier le problème

Scénario 3 :

- 1) Réaliser une cuisson avec de la faïence
- 2) Trois semaines plus tard, consulter l'historique des cuissons passées
- 3) Identifier la cuisson de la faïence réalisée précédemment
- 4) Mémoriser le programme utilisé
- 5) Modifier le programme en augmentant la température d'un segment de 10°C
- 6) Démarrer une cuisson avec le nouveau programme

Maquettes



Figures 2 & 3 : Maquettes du suivi de la cuisson et de la création d'un programme

Plateforme embarquée

L'application mobile agit sur une plateforme embarquée reliée au four. Cette plateforme doit d'une part assurer le contrôle de la température en suivant du mieux que possible la courbe de température désirée, et d'autre part communiquer avec l'application mobile pour recevoir les commandes et envoyer les données de la cuisson courante. Elle se doit d'être fiable et de prévenir au mieux les incidents ainsi que réagir de manière appropriée lorsqu'ils surviennent.

Pour des raisons de facilité du prototypage, nos choix matériel se concentrent sur des plateformes type Arduino et Raspberry Pi ainsi que des capteurs/actionneurs qui peuvent s'interfacer facilement avec ceux-ci. Plusieurs solutions ont été étudiées comme l'utilisation d'une plateforme Arduino Uno seule, Arduino Mega seule, Raspberry Pi seule ou Raspberry Pi et Arduino Uno, en comparant le prix, les contraintes de chacune et les fonctionnalités offertes. La dernière solution est retenue puisque d'une part elle répond le mieux aux contraintes imposées (robustesse, fiabilité associées à un faible coût) et que d'autre part, elle propose le plus de flexibilité pour l'implantation des fonctionnalités. Dans ce système, l'Arduino s'occupe de la régulation de température et de la gestion des incidents tandis que la Raspberry Pi s'occupe de la communication et de l'enregistrement des données.

Liste du matériel

Ci-dessous la composition de la plateforme matérielle dans un premier temps :

Qt	Nom	Prix unitaire (€ TTC)	Coût total (€ TTC)
1	Arduino Uno	19,50	19,50
1	Raspberry Pi 3 B	40,00	40,00
1	Carte SD 8 GB	9,00	9,00
1	Alimentation secteur 5V 3A	12,00	12,00
1	Module MAX13855	7,80	7,80
1	Relai basse tension	3,00	3,00
1	Afficheur HD44780	5,50	5,50
3	Bouton poussoir	0,30	0,90
10	Câble Dupont	0,20	2,00
1	Optocoupleur (ex. IL250)	2,00	2,00
	Total		101,70

Ces composants ont été choisis en particulier pour leur popularité et donc la facilité à se les procurer et à travailler avec. La carte Raspberry Pi 3 pourra être remplacée par exemple par une Raspberry Pi Zero W (23 €) ou une Orange Pi Zero (15 €) pour réduire les coûts. De même, l'Arduino Uno pourra être remplacé par une plateforme équivalente à un coût plus faible.

L'afficheur LCD pourra être remplacé par un écran 128x128 pour afficher la courbe de température par la suite. Les boutons poussoirs pourront servir à sélectionner/démarrer/arrêter un programme directement depuis le four sans passer par l'application.

Il reste à prévoir un boîtier pour l'intégration, l'impression 3D pourra convenir pour du prototypage mais pas pour le produit à intégrer sur le four, un boîtier robuste et plus résistant aux hautes températures sera nécessaire.

Une maquette mettant en œuvre une Raspberry Pi 3, un Arduino Uno, un thermocouple et son module MAX6675, un élément chauffant et un module de contrôle triac+optocoupleur est visible en [annexe 1](#).

Contrôle de température

Le contrôle de température se fait en boucle fermée grâce au principe du contrôleur PID. Les coefficients appliqués à chacun des paramètres affecteront les performances du contrôleur, et sont à déterminer expérimentalement. Étant donné que chaque four aura ses propres caractéristiques, une phase de calibration est nécessaire.

L'Arduino mesure en continu la température à l'aide du thermocouple (par l'intermédiaire du convertisseur MAX13855) et détermine la correction à appliquer.

L'Arduino actionne un premier relai de faible puissance, qui agit alors sur un second relai de forte puissance permettant le passage du courant dans la résistance afin d'augmenter la température du four. Réciproquement, le relai est actionné pour bloquer le passage du courant afin de diminuer la température du four, en fonction de la correction à appliquer. L'Arduino contrôle que l'action a bien été effectuée grâce à l'optocoupleur qui mesure un changement d'état (il doit prendre en compte que le courant est alternatif). La connectique présente sur le four propose peut être déjà un moyen de vérifier l'état de la résistance.

Gestion des incidents

Les incidents suivants sont à gérer avec différents niveaux de criticités et tentative de récupération :

- Valeurs incohérentes récurrentes en provenance du thermocouple : défaut du thermocouple ou du convertisseur
- Dérive trop importante entre température cible et température actuelle : profil incohérent, défaut de l'élément chauffant ou de son contrôle
- Courant circulant dans l'élément chauffant lorsqu'il ne devrait pas chauffer (détecté par l'optocoupleur) : défaut du contrôle de l'élément chauffant
- Courant ne circulant pas dans l'élément chauffant lorsqu'il devrait chauffer (détecté par l'optocoupleur) : défaut de l'élément chauffant ou de son contrôle
- La Raspberry Pi ne reçoit pas de réponse ou reçoit des données corrompues de l'Arduino : défaut de l'Arduino ou du lien de communication
- L'Arduino ne reçoit pas de réponse ou reçoit des données corrompues de la Raspberry Pi : défaut de la Raspberry Pi ou du lien de communication
- Panne de courant
- Arrêt d'urgence (distant ou local)

Communication

Raspberry Pi/Arduino

Connexion USB classique avec port série virtuel, le protocole de communication devra assurer un minimum d'intégrité des données. On pourra utiliser un protocole de plus haut niveau comme *MessagePack* et ajouter des sommes de contrôle par exemple après réception du profil de température, et des messages de notification de bonne réception (s'assurer que la commande d'arrêt a bien été transmise par exemple).

Raspberry Pi/Application mobile

Dans un premier temps, la Raspberry Pi fera office de point d'accès WiFi pour permettre une connexion directe du terminal mobile. Les communications se feront à l'aide d'une API REST avec un serveur NodeJS hébergé sur la Raspberry Pi.

Dans un second temps, la Raspberry Pi pourra être configurée pour se connecter à un point d'accès WiFi existant.

Enfin, il est envisageable d'ajouter la prise en charge du Bluetooth, avec côté client et côté serveur une passerelle pour réutiliser l'API REST existante.

Planning

Prototype 1 – semaine 1 :

- Lancer un programme prédéfini (programme déjà chargé)
- Connecter l'application au boîtier

- Lancer une cuisson depuis l'application (start / stop)
- Réguler en boucle fermée la température
- Afficher la température sur l'application
- Gérer les erreurs critiques

Prototype 2 — semaine 2 :

- Configurer le programme
- Réceptionner le programme
- Enregistrer le programme

Prototype 3 — semaine 3 :

- Afficher le suivi de la cuisson
- Récupérer les erreurs non critiques
- Gérer les erreurs
- Alerter en cas de problème

Prototype 4 — semaine 3 et 4 :

- Choisir un programme
- Afficher un programme
- Modifier un programme
- Importer / exporter / dupliquer un programme
- Afficher la température et les erreurs sur un écran 7 segments

Prototype 5 — semaine 4 :

- Enregistrer les suivis des cuissons
- Afficher le déroulement d'une cuisson
- Afficher l'historique des cuissons

Prototype 6 :

- Importer / exporter un historique
- Annoter des programmes
- Annoter des historiques de cuisson
- Envoyer une notification en cas d'erreur
- Envoyer une notification en fin de cuisson
- Consulter à distance l'état de la cuisson

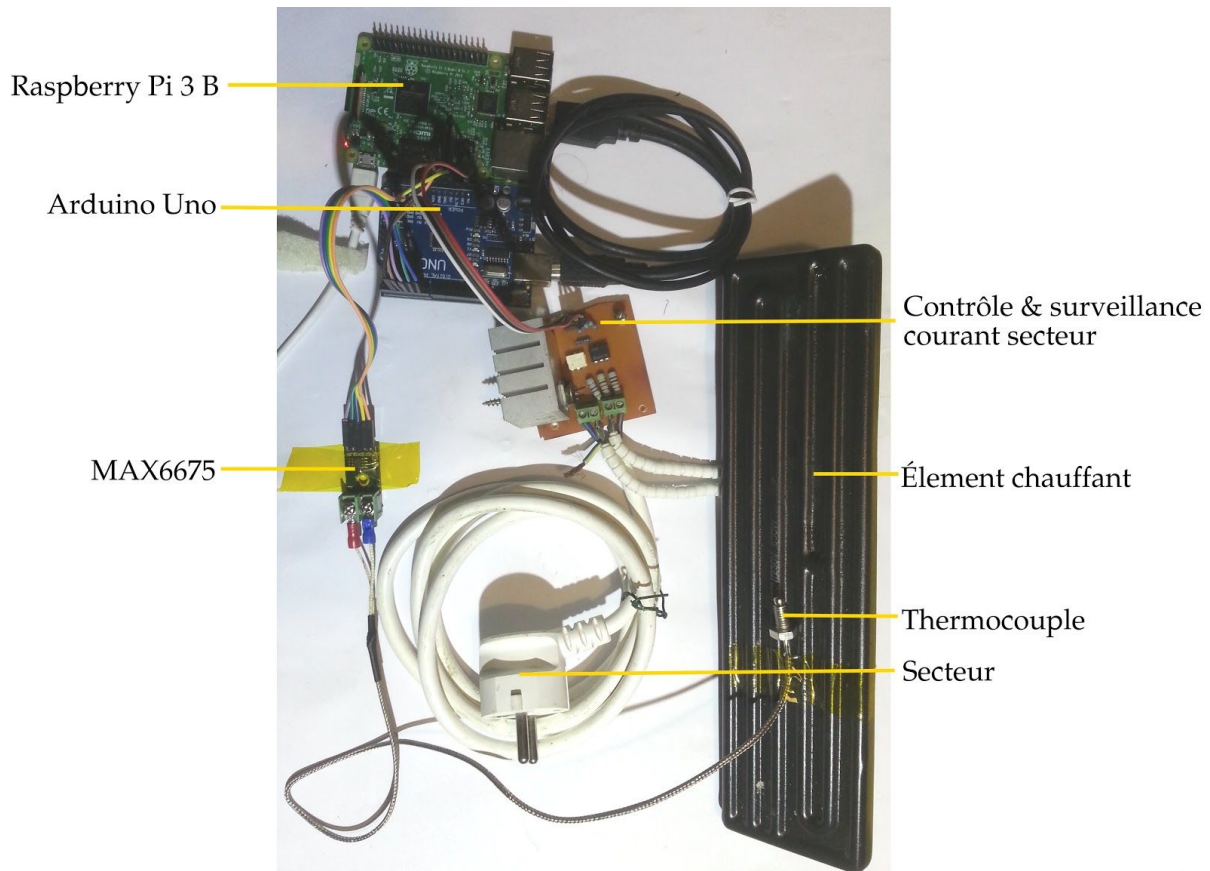
Conclusion

Après avoir discuté avec des céramistes, nous nous rendons compte à quel point notre outil pourrait faciliter leur quotidien. Cela nous motive pour la suite. Selon le planning que nous avons établi, nous sommes confiants quant à la faisabilité du projet, que ce soit du point de vue embarqué ou mobile.

Nous ne pouvons pas accéder à toutes les demandes des céramistes pour la durée qui nous est accordée. Si le projet a pour vocation de continuer, nous avons d'autres fonctionnalités à proposer. Nous pourrions mettre en place un système d'échanges et de partage de cuissons et de conseils. Nous souhaiterions apporter un système très demandé par les céramistes permettant de fermer le bouchon automatiquement après une durée fixée. Enfin, nous aimerions ajouter du machine learning à notre système afin qu'il puisse conseiller ses utilisateurs quant à la cuisson à programmer.

Annexes

Annexe 1



Figures 4 : Maquette de la plateforme embarquée de contrôle de température