

Propulsion automatisée de légumes par déflagration : Commande à distance d'un lanceur de tubercules comestibles



Étudiants :

Esteban MOUTY
Louis COUTURIER
Paul TESSE

Florine CHEVRIER
Paul SMACQUE
Rémy FNEICH

Enseignant-responsable du projet :
Clément BRUNET

Date de remise du rapport : 15/06/2020

Référence du projet : STPI/P6/2020 – N°42

Intitulé du projet : Propulsion automatisée de légumes par déflagration : Commande à distance d'un lanceur de tubercules comestibles

Type de projet : *Expérimental*

Objectifs du projet : L'objectif de notre projet est de théoriser et créer le lanceur de tubercules comestibles ou "patator" le plus performant possible en coopération avec 3 autres groupes étudiants de STPI2. Notre groupe doit notamment créer un circuit électrique avec un système d'électrovanne, qui doit être également capable de mettre en place la routine d'injection et d'allumage du lanceur créé par les autres groupes, l'ensemble devant être pilotable par un microcontrôleur lui-même connecté à un smartphone en bluetooth afin de recevoir les instructions de l'opérateur. En somme, il s'agit de créer un système capable de déclencher à distance, grâce à un smartphone, la déflagration permettant la propulsion du condiment, de façon à assurer la sécurité de l'utilisateur.

Mots-clefs du projet : *Electronique, Arduino, Développement, Réalisation*

Table des matières

Notations	5
Introduction	6
1 Méthodologie, organisation du travail	7
2 Travail réalisé et résultats	8
2.1 Réalisation de l'application	8
2.1.1 Cahier des charges	8
2.1.2 Développement	8
2.1.3 Résultats obtenus	10
2.2 L'électrovanne	11
2.2.1 Dans quel domaine l'utilise-t-on?	11
2.2.2 Electrovanne « tout ou rien » et proportionnelle	11
2.2.3 Comment fonctionne une électrovanne	12
2.2.4 Notre électrovanne	13
2.3 Modélisation du circuit électrique de commande à distance	13
2.3.1 Le circuit de commande	14
2.3.2 L'interrupteur électronique	15
2.3.3 Le circuit d'alimentation	15
2.3.4 Conclusion	18
3 Arduino	19
3.1 Recherche & Preparation	19
3.1.1 Choix des composants	19
3.1.2 Principe de fonctionnement	20
3.2 Premières manipulations, les bases de l'arduino	21
3.2.1 Allumer une LED	21
3.2.2 Allumer plusieurs LED grâce aux touches du clavier	21
3.3 Commande à distance	22
3.3.1 Allumer une LED à distance	22
3.3.2 Fonctionnement de l'AOP	23
Conclusion	25
Bibliographie	26
Annexes	28

Remerciements

Nous adressons ces remerciements aux personnes qui nous ont aidés dans la réalisation de ce projet.

En premier lieu, nous remercions Clément Brunet, doctorant et enseignant à L'INSA de Rouen. En tant que Directeur de projet, il nous a tout d'abord fourni un sujet original et intéressant, et malgré la crise sanitaire, il a su nous guider dans notre travail et nous a aidé à trouver des solutions pour avancer.

Nous tenons aussi à remercier Sylvain Petit et les élèves des autres groupes qui ont aussi travaillé sur le projet. Nous souhaitons particulièrement remercier Mathias Van Audenhove et Gabriel Feltgen pour leur aide précieuse à la relecture et à la correction de notre projet.

Notations et Acronymes

Notation à définir : définition

Acronymes à définir : — STPI : Sciences et Techniques Pour l'Ingénieur

Introduction

Dans le cadre de notre deuxième année de formation à l'INSA, il nous est demandé de réaliser un projet de physique en groupe. Le but pédagogique de ce projet est double, il permet aux étudiants d'appliquer leurs enseignements dans un cadre plus concret et nécessitant organisation et communication au sein d'un groupe. Ce dernier point est particulièrement important tout ingénieur doit être apte à coopérer avec ses pairs et savoir s'adapter à des contraintes temporelles.

Nous avons donc travaillé sur la réalisation du projet "Propulsion automatisée de légumes par déflagration" avec 3 autres groupes d'étudiants issues de STPI2. Nous étions notamment en charge de la partie "Commande à distance d'un lanceur de tubercules comestibles" du projet. Le rôle de notre groupe était donc de concevoir et de fournir un lanceur fonctionnel, performant et utilisable depuis une distance de sécurité.

La réalisation de ce projet, outre le divertissement, est l'occasion de mettre en pratique des connaissances acquises au cours de notre cursus STPI, mais aussi de les développer et de les expérimenter dans de nombreux domaines. En effet, le design d'un tel système demande des connaissances en résistance des matériaux afin d'assurer la pérennité du lanceur, ainsi que la sécurité des utilisateurs. Afin de permettre une déflagration optimale, l'application de connaissances en mécanique des fluides ainsi qu'en chimie sont nécessaires. Outre le lanceur, il faut également pouvoir prévoir la trajectoire que va emprunter notre projectile, ce qui nécessite des capacités en mécanique du point.

Enfin, dans notre groupe nous avons notamment utilisé les apprentissages de l'E.C P3 électricité, au travers de la création, mais nous avons également pu expérimenter l'utilisation d'une carte Arduino, qui crée un pont intéressant entre l'E.C P3 et les E.C informatiques que nous avons également pu suivre. Enfin, nous avons dû réaliser une application mobile.

L'acquisition de ces nouvelles compétences et la mise en pratique en autonomie de nos connaissances nous a permis d'entrevoir un nouveau champ de possibilités dans les expérimentations que nous pourrions réaliser.

Ainsi, malgré les circonstances actuelles qui ne nous ont pas permis de réaliser le projet dans les conditions attendues et d'atteindre l'objectif initialement prévu, nous considérons cette expérience comme un réel atout et un gain de confiance dans le cadre de nos études et de notre future carrière professionnelle.

Chapitre 1

Méthodologie, organisation du travail

Afin de mener le projet à bien il était nécessaire de diviser le travail entre les 6 membres du groupe. Nous avons donc adopté la disposition suivante :

- Paul SMACQUE : Arduino et montages associés
- Esteban MOUTY et Paul Tessé : Modélisation et montage du circuit alimentation et commande
- Louis COUTURIER : Électrovanne et montages associés
- Florine CHEVRIER et Rémy FNEICH : Conception de l'application et mise en forme du rapport sur L^AT_EX

Nous avons également décidé d'adopter, sur les conseils de M.Brunet et en accordance avec les autres groupes, le calendrier suivant :

- Fin Février/Début Mars : Design effectué et rassemblement des composants
- Avril : Montage et test de l'appareil
- Mai : Rédaction de notre rapport au propre et continuation des tests afin de le rendre fonctionnel ou de l'améliorer
- Juin : Préparation de la soutenance orale

Chapitre 2

Travail réalisé et résultats

2.1 Réalisation de l'application

2.1.1 Cahier des charges

Notre cahier des charges est relativement simple. L'application doit fonctionner sur la majorité des téléphones possédés au sein du groupe, être simple d'utilisation et le développement devait être réalisable dans un temps acceptable.

2.1.2 Développement

Étant donné que la majorité des membres du projet possède un téléphone fonctionnant sous Android nous avons décidé de travailler avec ce système d'exploitation. De plus, le développement sous iOS s'annonçait plus complexe que celui sous Android. L'installation d'applications en développement sur iOS demande des manipulations supplémentaires, ce qui n'est pas le cas sur Android où il suffit d'autoriser l'installation d'application ne provenant pas du Google Store.

Une fois le système d'exploitation choisi, il était nécessaire de savoir quelles fonctionnalités notre application devait adopter pour répondre à nos besoins. Nous avons donc obtenu la liste de fonctionnalités suivante :

- Affichage d'une liste permettant de choisir l'appareil auquel l'on veut se connecter en Bluetooth.
- Information sur l'état de la connexion (Connecté/Déconnecté).
- Envoi de données à l'appareil auquel on est connecté.
- Prévoir un affichage des informations reçues au cas où, grâce à un capteur, une amélioration soit possible plus tard dans le projet.
- Une interface intuitive et graphiquement plaisante.

Une fois ces objectifs fixés, il s'agit de choisir la plateforme de développement que nous souhaitons utiliser. Nous avons retenu deux choix principaux : Android Studio et MIT App Inventor.

Android Studio est un environnement de développement orienté vers la création d'applications sur Android. Il a l'avantage de pouvoir incorporer un émulateur de smartphone Android afin de pouvoir tester facilement nos créations sans avoir à importer et installer

l'application sur un téléphone pour chaque version. De plus, comme il s'agit d'un environnement de développement les seules limites de notre développement sont les capacités de notre matériel, nos connaissances du développement d'application Android et des langages de programmation Java et HTML.

MIT App Inventor est une application web créée par Google et maintenue par le Massachusetts Institute of Technology qui offre une interface extrêmement intuitive. L'objectif et tout l'avantage de cette application sont la simplicité de développement qu'elle offre. Par ailleurs, aucun code n'est à écrire. Seule une trame logique entre l'algorithme et l'assemblage de "bloc" doit être réalisée. Néanmoins de cette simplicité découle un défaut, nous n'avons pas le code source de l'application et nous ne pouvons pas créer une application visuellement plaisante à partir de cet outil.

Nous avons décidé de réaliser une application avec le MIT App Inventor puis si le temps nous le permettait, de développer une seconde application à l'aide de Virtual Studio. En effet, cela avait l'avantage de nous permettre de réaliser une application opérationnelle rapidement permettant d'atteindre les objectifs principaux et de réaliser les tests nécessaires.

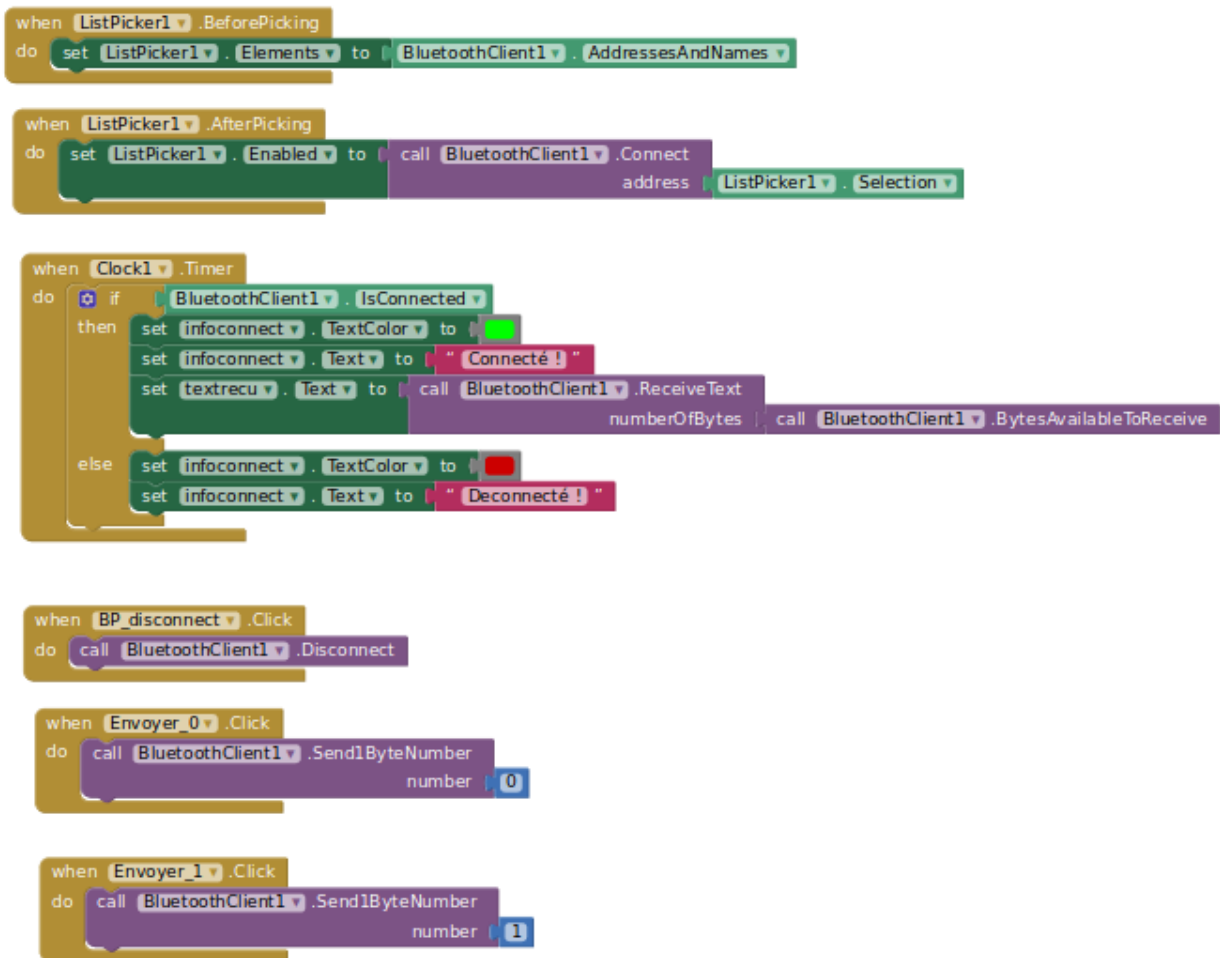


FIGURE 2.1 – Blocs logiques de l'application

Le premier et le second bloc permettent de définir le fonctionnement de la liste des éléments connectés avant et après connexion à la cible. Dans les deux cas il s'agit de présenter la liste des cibles connues par le téléphone et de permettre à l'utilisateur de choisir celle à

laquelle il souhaite se connecter.

Le troisième bloc permet de définir le fonctionnement de l'indicateur de l'état de connexion affichant "Déconnecté" en rouge ou "Connecté" en vert, ainsi qu'une ligne permettant éventuellement l'affichage de données reçues.

Le quatrième bloc permet de créer un bouton "Déconnexion" permettant de mettre fin à la connexion Bluetooth avec la cible.

Les deux derniers blocs définissent chacun un bouton, l'un permettant d'envoyer un bit 0 à la cible, l'autre un bit 1.

2.1.3 Résultats obtenus

Voici à quoi ressemble le code de notre application que vous pouvez observer ci-dessus.



FIGURE 2.2 – Interface de l'application

L'application fonctionne comme décrit précédemment :

- L'information "Déconnecté!" en rouge indique l'état de la connexion et après connexion affiche le texte "Connecté!" en vert.
- L'onglet Bluetooth permet d'afficher les appareils sur lesquels il est possible de se connecter. Il est évidemment à la charge de l'utilisateur d'être à portée de l'Arduino. (voir annexe).
- Le texte "Info reçu" est remplacé dans le cas où l'Arduino tente de transmettre des informations au téléphone en Bluetooth.

- Les boutons “Envoyer 0” ou “Envoyer 1” permettent d’envoyer un bit égal à 0 ou 1 au système auquel le système est connecté
- Le bouton “Déconnexion” permet de mettre fin à la connexion Bluetooth

Nous avons donc réussi à répondre à notre cahier des charges, à l’exception de l’aspect esthétique qui laisse à désirer. En effet, l’application MIT app Inventor semble avoir du mal à s’accommoder à la taille des téléphones actuels, ce qui explique le positionnement étonnant des informations et des boutons. Nous avons considéré qu’il s’agissait ici du positionnement le plus acceptable et permettant de s’adapter au mieux aux différents smartphones en notre possession.

Enfin, pour nous assurer que l’application marchait correctement nous avons effectué un test permettant de nous assurer des diverses fonctionnalités et de la compatibilité avec l’Arduino. Comme décrit précédemment dans la partie s’intéressant à la carte Arduino, nous avons simplement allumé une LED lors de l’envoi d’un bit 1 à l’Arduino depuis l’application, puis nous l’avons éteinte en envoyant un bit 0. Cette expérience fut concluante, nous avons donc considéré que l’application était suffisamment satisfaisante.

2.2 L’électrovanne

L’électrovanne est un composant important de notre circuit puisqu’il va nous permettre de gérer l’injection de gaz dans la chambre de combustion.

2.2.1 Dans quel domaine l’utilise-t-on ?

L’électrovanne fonctionne exactement comme une vanne classique, c’est-à-dire qu’elle bloque ou non le passage d’un fluide. Dans notre cas, le fluide sera un gaz explosif comme un mélange de méthane ou de propane avec de l’air.

L’électrovanne est principalement utile pour l’alimentation, le sectionnement de tuyauteries et la purge de celles-ci. De plus, elle a de multiples applications telles que les pompes à chaleur, les chauffages ou les lave-vaisselles par exemple. Cependant, elle sert surtout pour les réseaux de traitement des eaux, les réseaux incendies ou encore les réseaux de gaz.

Nous allons maintenant regarder plus en détail l’électrovanne et expliquer les différents composants pour comprendre comment celle-ci fonctionne.

2.2.2 Électrovanne « tout ou rien » et proportionnelle

La première particularité est la distinction entre proportionnelle et « tout ou rien ». Proportionnelle signifie que l’on va pouvoir contrôler le débit du fluide lors de son passage à travers la vanne. En fonction de la tension ou du courant électrique appliqué, l’ouverture entre le clapet et la paroi est plus ou moins grande. On en déduit que la vanne est proportionnelle à cette tension. Au contraire, une électrovanne « Tout ou rien » laisse juste s’écouler le fluide. L’orifice de passage est toujours le même et ne varie pas au cours du temps. Parmi ces électrovannes, il faut en distinguer deux types :

- Les électrovannes normalement ouvertes.
- Les électrovannes normalement fermées.

L’application d’une tension sur la première va la fermer tandis que la vanne va s’ouvrir pour la deuxième. Dans notre cas, nous utiliserons une électrovanne normalement fermée.

2.2.3 Comment fonctionne une électrovanne

Tout d'abord, il est important de distinguer les deux types d'électrovannes que sont :

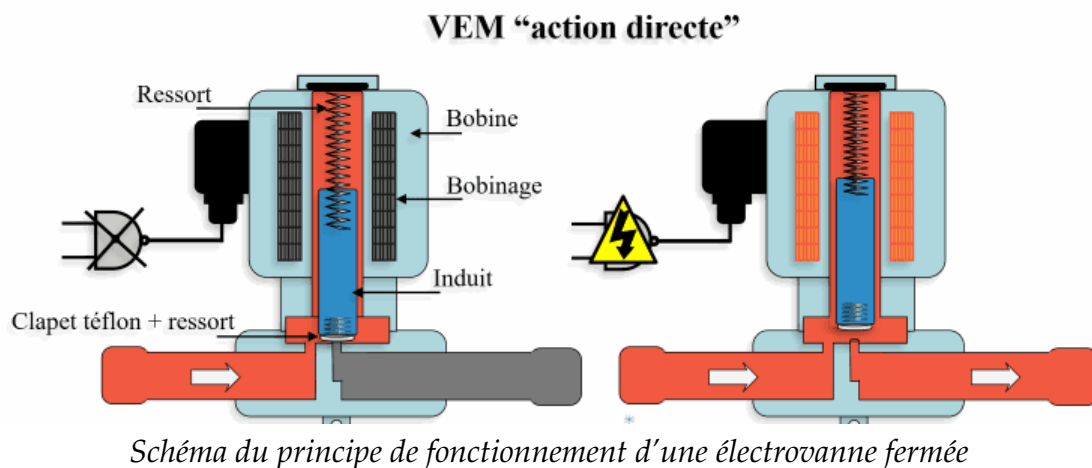
- Les électrovannes « solénoïde » à commande directe.
- Les électrovannes à membrane assistée.

Pour commencer, nous expliquerons le premier type, puis nous le comparerons au second type dans le cas d'une électrovanne normalement fermée.

Une électrovanne peut être séparée en deux parties bien distinctes. La première partie est le haut de l'électrovanne appelée tête magnétique dans laquelle se trouvent une bobine, un ressort, une culasse et une bague de déphasage.

La bobine va induire un champ magnétique lorsque celle-ci est actionnée. Enroulés autour d'un cylindre et fixes par rapport au bâti, les fils de cuivre sont très efficaces et sont fréquemment utilisés pour la bobine. La culasse, située à l'une des extrémités du ressort est fixe elle aussi. Elle accroît les effets du champ magnétiques lors du fonctionnement. L'autre extrémité du ressort est associée au noyau (voir deuxième partie). Enfin, la bague de déphasage, située entre la culasse et le noyau mobile va permettre de limiter les vibrations liées au ressort.

Le rôle de l'assemblage des différentes pièces décrites précédemment est de faire monter ou descendre le noyau dans le tube. En position de repos, le ressort est étiré mais lorsque l'on actionne la bobine, celui-ci va se rétracter jusqu'à toucher la culasse entraînant ainsi avec lui le noyau.



Expliquons ce schéma afin de mieux comprendre. Celui-ci est constitué de deux orifices de raccordement qui sont bloqués par un clapet ou une membrane, selon le type d'électrovanne. Là encore un ressort est fixé au clapet. Le clapet va servir de barrière au fluide entrant. En effet, très étanche et résistant, il bloque le fluide en position de repos. Le corps est lui aussi à l'épreuve des fluides grâce aux garnitures d'étanchéité entre la tête et le corps et au niveau du clapet. Le noyau ou « induit » est un cylindre fixé au ressort de noyaux d'un côté et au ressort à clapet de l'autre. Il est en translation par rapport au bâti et va permettre la montée et la descente du clapet.

Si l'on résume, il suffit d'envoyer un courant électrique assez puissant pour déclencher la bobine qui par induction fera monter ou descendre le clapet étanche. C'est ce modèle d'électrovanne que nous utiliserons dans notre projet.

L'électrovanne à membrane directe est légèrement différente du point de vue du fonctionnement mais l'objectif final reste le même. Cette électrovanne fonctionne grâce à une différence de pression entre l'orifice d'entrée et l'orifice de sortie (entre l'amont et l'aval). Quand aucune tension n'est appliquée, le clapet bouche l'orifice pilote (du côté aval). La bobine est juste utilisée pour donner le signal d'ouverture. De plus, une membrane a été rajoutée entre le clapet et les orifices et un différentiel de pression est nécessaire. Celui-ci, va permettre de réguler la montée et la descente du clapet en fonction de la pression P1 à l'entrée et de la pression P3 en sortie.

Lorsque l'on alimente la bobine, l'orifice pilote va s'ouvrir, créant ainsi une différence de pression entre l'entrée et la sortie. Le fluide situé dans la cheminée, c'est à dire au-dessus de la membrane, à la pression $P2 = P1$ au début, va s'enfuir par l'orifice pilote ouvert. La pression P1 à l'entrée va alors être plus forte que P2 elle-même plus forte que P3 ($P1 > P2 > P3$). La fermeture du clapet stoppe l'écoulement du fluide et augmente la pression P1, ce qui va garantir que la vanne est bien fermée.

2.2.4 Notre électrovanne

Pour réaliser le circuit d'alimentation, nous avons besoin des caractéristiques électriques de l'électrovanne qui nous a été fournie, et notamment de l'intensité et de la tension nécessaires à son bon fonctionnement. Nous avons ainsi trouvé la fiche technique d'une électrovanne extrêmement similaire à la nôtre, celle type 470 de la marque Burkert (photo en annexe). Notre électrovanne nécessite donc 24 Volts pour fonctionner de manière optimale. Or $I=P/U$ avec $P=2W$ et $U=24V$, nous obtenons donc $I=83mA$.

Sur la fiche technique, nous pouvons voir que cette électrovanne est faite pour faire circuler des gaz. Elle est donc bien étanche. Ce point est capital, puisque beaucoup d'électrovannes servent à alimenter des circuits en eau, et particulièrement des circuits d'arrosage de jardin. Or ces systèmes ne sont pas étanches, ce qui n'est pas envisageable puisqu'on nous faisons circuler un gaz explosif.

Au sujet du gaz, nous avons donc dû vérifier que sa composition et celle de l'électrovanne et des tuyaux ne soient pas conflictuelles. Le gaz choisi est un mélange de propane. Les tuyaux transportant le gaz sont très probablement en caoutchouc. Nous avons cherché des informations au sujet d'une incompatibilité entre le propane et le caoutchouc, sans en trouver. Ainsi, le propane ne semble pas réactif avec les composants des tuyaux. De plus, ces derniers ainsi que l'électrovanne ont été créés pour être utilisés dans des laboratoires, et pour convoier ce gaz explosif. Notre montage semble donc relativement sûr pour notre utilisation.

2.3 Modélisation du circuit électrique de commande à distance

Une fois les composants choisis, il fallait imaginer un circuit de commande et d'alimentation afin de permettre de piloter via l'Arduino l'injection du gaz dans la chambre de combustion, ainsi que le déclenchement de la combustion grâce à une étincelle. Nous avons donc longtemps réfléchi aux différentes problématiques liées à ce circuit électrique. (Voir Annexe 4.6)

2.3.1 Le circuit de commande

Le protocole de fonctionnement

Dans un premier temps, nous nous sommes interrogés sur le protocole de fonctionnement du patator, dans le but de mieux visualiser la façon dont allaient être pilotés l'électrovanne et la bougie.

Notre circuit de commande devait permettre de déclencher l'injection du gaz, puis la combustion de celui-ci. Les deux étapes devaient se faire de manière bien distinctes, pour éviter tout risque de déclenchement de la combustion alors que l'électrovanne était encore ouverte. Nous en avons déduit que les deux composants ne seraient jamais actionnés de manière simultanée, ce qui peut avoir une incidence sur notre circuit de commande.

De plus, notre montage électrique devait être de taille raisonnable, afin de pouvoir être fixé au patator, ce qui nous imposait de prendre en compte ce facteur de taille du circuit et des autres composants.

L'injection se faisant grâce à une électrovanne tout ou rien, nous en avons déduit que la façon de piloter ce composant à distance serait de l'alimenter uniquement lors de l'injection. De cette manière, l'électrovanne serait normalement fermée et s'ouvrirait pour laisser passer le gaz lorsque celle-ci serait alimentée.

Pour ce qui est de la bougie, il en serait de même. On alimenterait celle-ci afin de produire une étincelle lorsque le gaz serait dans la chambre.

Nous avons une idée de circuit de commande. Le principe était de pouvoir contrôler l'alimentation de la bougie et de l'électrovanne. Nous pourrions ainsi donc déclencher l'électrovanne puis la bougie, ce qui permettrait de piloter à distance la projection de la pomme de terre.

Réflexions sur le circuit de commande

Nous avons naturellement pensé à un circuit de commande se basant sur des interrupteurs contenus dans le circuit d'alimentation des actionneurs. Les interrupteurs seraient ouverts initialement, puis fermés afin d'alimenter les deux actionneurs pendant le temps nécessaire. Une fois leur tâche effectuée, l'interrupteur serait ouvert à nouveau pour interrompre leur fonctionnement.

Nous avons donc cherché comment nous pourrions piloter ces interrupteurs avec notre module Arduino.

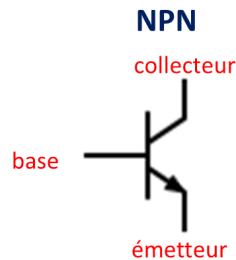
L'utilisateur transmettrait l'ordre de déclencher l'injection du gaz via l'application. Le téléphone, relié en bluetooth à notre carte Arduino transmettrait donc l'instruction de fermer l'interrupteur pendant le temps nécessaire à l'injection, laissant ainsi passer le fluide par l'électrovanne, maintenant ouverte. En appliquant ce même principe de fonctionnement à la bougie, nous pourrions également commander la génération de l'étincelle à distance.

Il ne nous restait plus qu'à trouver quel composant pourrait faire office d'interrupteur pilotable par de l'Arduino.

2.3.2 L'interrupteur électronique

Nous avons effectué quelques recherches pour trouver un composant que l'on pourrait insérer dans le circuit d'alimentation et qui servirait d'interrupteur pilotable par notre carte Arduino. Nous nous sommes donc intéressés aux transistors.

Un transistor est un dispositif semi-conducteur à trois électrodes actives, qui permet de contrôler un courant ou une tension sur l'électrode de sortie (le collecteur) grâce à une électrode d'entrée (la base). Il existe plusieurs types de transistors : transistors bipolaires, à effet de champ, unijonction et hybride.



Les transistors NPN et PNP sont des semi-conducteurs bipolaires. Contenant deux types de silicium légèrement différents, ils acheminent l'électricité à l'aide des polarités des transporteurs de charges, les trous et les électrons. La différence entre un transistor NPN et PNP est que pour autoriser le passage du courant du collecteur à l'émetteur, il faut appliquer une tension relativement positive à la base pour un transistor NPN et une tension relativement négative pour un PNP. Nous avons donc choisi un transistor NPN.

L'idée serait donc d'alimenter la base des transistors NPN grâce à notre carte Arduino lorsque l'utilisateur donne l'instruction de déclencher l'injection ou la combustion. De cette manière, nous pourrions piloter à distance le circuit d'alimentation et donc la bougie ainsi que l'électrovanne.

Vous trouverez donc une ébauche de notre circuit électrique dans les annexes (Figure...), avec le circuit de commande comportant le montage Arduino pilotant la base de deux transistors, un pour chaque actionneur. Nous avons également dû penser à une ébauche de circuit d'alimentation.

2.3.3 Le circuit d'alimentation

La structure du circuit d'alimentation

Nous avons choisi de placer l'électrovanne et la bougie sur deux branches en parallèle. Ce choix a été motivé par le fait que les deux composants ne devaient pas fonctionner en même temps. Ce montage en parallèle permet de couper l'alimentation des deux actionneurs et donc de les commander très facilement.

De plus, ce montage permet d'éviter l'utilisation de deux circuits d'alimentation, un pour chaque actionneur, et donc d'optimiser la place prise par notre circuit.

Enfin, la tension et l'intensité peuvent être facilement adaptées en fonction des caractéristiques de chaque actionneur et sans impacter le reste du circuit.

Néanmoins, ce montage présentait une complication : l'apparition d'un nœud, et donc une division de l'intensité dans chaque branche, ce qui aurait pu poser problème. Cependant, ce problème n'a pas lieu d'être au vu du mode de fonctionnement de ce circuit. En effet, comme nous l'avons spécifié précédemment, les deux composants fonctionnent alternativement. Cela implique que les deux transistors ne seront jamais fermés en même temps. C'est donc comme s'il n'y avait aucun nœud.

Les caractéristiques électriques de nos composants

Comme expliqué précédemment dans notre rapport, nous avons reçu de M. Brunet pour notre circuit un accumulateur, ainsi qu'une électrovanne tout ou rien. L'accumulateur permet de délivrer une tension de 9V. Les transistors NPN sont alimentés par la carte Arduino, délivrant une tension d'alimentation de 5V. Pour ce qui est de la carte Arduino en elle-même, elle nécessite une tension d'entrée de 7 à 12V. Il faudrait donc une petite batterie ou bien un autre accumulateur pour pouvoir l'alimenter. L'alimentation des transistors et de la carte Arduino ne pose donc pas de problème.

L'électrovanne, quant à elle, nécessite une tension d'entrée de 24V pour pouvoir fonctionner, d'après ce que nous avons pu trouver dans la documentation technique. La tension délivrée par l'accumulateur, n'est donc pas suffisante. Autre caractéristique importante, d'après nos calculs de puissance, l'électrovanne a besoin d'une intensité de 0,08A pour pouvoir fonctionner.

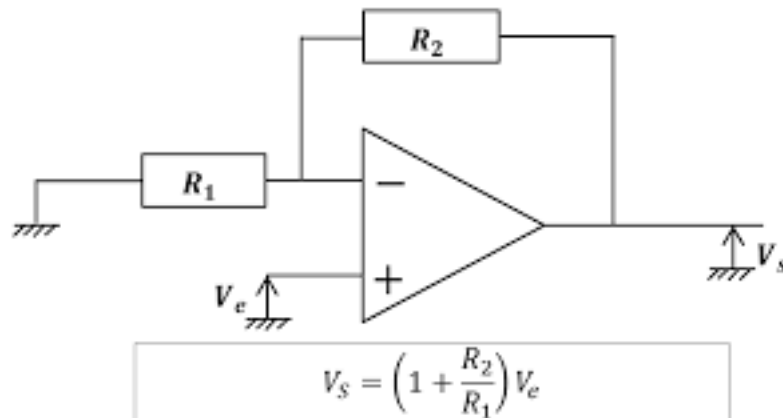
Pour ce qui est de la bougie, nous avons décidé de la remplacer par deux électrodes espacées de 1mm. Ce choix a été motivé d'une part par la difficulté de trouver une bougie adaptée à notre montage et d'autre part, par la découverte dans nos recherches sur le sujet, d'une méthode offrant une meilleure synergie avec l'Arduino utilisant deux fils en guise d'électrode. Pour pouvoir générer une étincelle entre les deux fils, nous avons besoin d'après nos recherches d'une tension de 3000V. Encore une fois, la tension délivrée par notre accumulateur n'était pas suffisante.

Montage amplificateur

Nous avons donc besoin d'amplifier la tension d'entrée fournie par l'accumulateur. Pour cela, nous avons tout de suite pensé à utiliser un montage utilisant un amplificateur opérationnel.

Ce montage sert à amplifier notre tension d'entrée afin que la tension en sortie de l'AOP corresponde à la tension nécessaire pour faire fonctionner la bougie et l'électrovanne. Il existe différentes sortes de montages : montage suiveur, montage inverseur, montage en cascade, montage non-inverseur, etc.

Le montage que nous avons choisi est le montage non-inverseur afin d'avoir un gain et une tension en sortie positive. Voici le schéma d'un montage non-inverseur classique :



On peut voir d'après l'expression de V_s que le gain est caractérisé par le rapport $\frac{R_2}{R_1}$. Cela signifie qu'en adaptant notre couple de résistances, on peut obtenir une vaste plage de tensions de sortie. Une fois les tensions de fonctionnement connues pour les actionneurs, il est facile d'adapter la tension de sortie du montage amplificateur.

Pour avoir une idée du dimensionnement des résistances, nous avons appliqué la loi d'Ohm :

$$U_s = \left(1 + \frac{R_2}{R_1}\right) * U_e$$

$$\Leftrightarrow R_2 = \left(\frac{U_s}{U_e} - 1\right) * R_1$$

Ensuite, en posant $U_s = 24V$ (tension électrovanne), $U_e = 9V$ et une valeur arbitraire de 100 Ohms pour R_1 , on trouve une valeur théorique pour la résistance R_2 de 166 Ohms, ce qui est tout à fait raisonnable.

Pour pouvoir générer les 3000V requis pour l'étincelle, on pose $U_s=3000V$ et avec $R_1 = 5$ Ohms, on obtient cette fois-ci une valeur théorique de R_2 de 1662 Ohms, ce qui est tout à fait raisonnable.

Ainsi, afin de pouvoir fournir une tension suffisante pour nos deux actionneurs, nous avons choisi d'utiliser la deuxième résistance de manière à pouvoir fournir 3000V en sortie de l'AOP. L'ajout de résistances devant l'électrovanne permettra d'abaisser la tension à 24V et ainsi protéger le composant.

Le problème de l'amplification de tension était donc réglé, mais restait encore à vérifier que l'intensité du courant délivré en sortie du montage amplificateur était adaptée aux besoins de nos deux actionneurs. (voir Figure Annexe 2.7)

Intensité du courant dans le circuit

Nous rappelons que les deux transistors n'étant jamais fermés en même temps, déterminer l'intensité en entrée des deux actionneurs revient à déterminer l'intensité en sortie de notre montage amplificateur.

En faisant l'hypothèse que la puissance en entrée de notre montage amplificateur est conservée nous pouvons ainsi déterminer l'intensité en sortie du montage.

Calcul de la puissance entrante P_e :

D'après une mesure expérimentale de l'intensité délivrée par l'accumulateur, on obtient $I_e = 0.4$ A.

$$P_e = U_e * I_e$$

Application numérique : $P_e = 9 * 0.4 = 3.6 \text{ W}$

Nous utilisons les relations suivantes : $I_S = P_S / U_S$ et $P_S = P_E$

Application numérique : $I_S = 3.6 / 3000 = 1.2 * 10^{-3} \text{ A}$

D'après nos calculs théoriques, l'intensité en sortie de notre montage amplificateur est de $1.2 * 10^{-3} \text{ A}$. Ce résultat est satisfaisant pour générer l'étincelle car il n'y a pas de contrainte d'intensité, cependant cela n'est pas suffisant pour pouvoir alimenter l'électrovanne car d'après nos calculs il fallait 0.08 A .

Nous nous sommes alors demandés si avec notre montage amplificateur dimensionné pour délivrer 24 V seulement, l'intensité serait suffisante. Avec la même formule pour déterminer I_s mais en prenant pour valeur de $U_s 24 \text{ V}$ on obtient $I_s = 0.125 \text{ A}$, ce qui est largement suffisant. (voir Figure Annexe 2.8)

2.3.4 Conclusion

Suite à notre étude, nous avons décidé de dédoubler notre montage. En effet, nous étions initialement partis de l'idée de ne faire qu'un seul circuit pour piloter et alimenter à la fois l'électrovanne et les électrodes. Nous avons pu voir d'après nos calculs théoriques que nos solutions sont viables individuellement, mais pas lorsqu'il s'agit d'alimenter et de piloter les deux actionneurs avec le même montage. C'est pour cela que nous avons décidé de créer deux circuits avec une batterie 9V , un montage amplificateur opérationnel non-inverseur, un transistor piloté par notre carte Arduino et pour finir notre carte Bluetooth.

Vous trouverez donc les schémas de nos deux montages en Annexe (Figure 2.9 et 2.10)

Chapitre 3

Arduino

3.1 Recherche & Preparation

3.1.1 Choix des composants

Pour contrôler à distance le patator nous avons tout de suite pensé à utiliser une carte arduino et une carte bluetooth permettant d'envoyer des informations à l'arduino qui agirait en conséquence sur le patator. Nous avons donc entamé des recherches concernant ces composants.

La carte arduino fut assez simple à trouver, il en existe plusieurs modèle mais notre choix s'est porté sur l'Arduino Uno. Nous avons ensuite cherché des modules bluetooth que nous pourrions connecter à notre arduino. Nous en avons trouvé deux, la carte HC-05 et la carte HC-06 que nous avons comparé, voici un tableau récapitulatif de nos recherches :

	HC-05	HC-06
Nombre de broches	6	4
Modes disponibles	Esclave & maître	Esclave
Prix	10€	10€

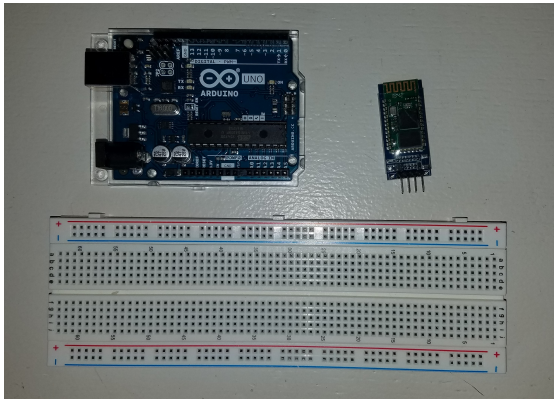


FIGURE 3.1 – Carte bluetooth HC-05



FIGURE 3.2 – Carte bluetooth HC-06

Le mode « esclave » ne peut que recevoir des demandes d'appairage tandis que le mode « maître » peut proposer à un autre élément bluetooth de s'appairer avec lui. Nous ne voulions pas spécialement que notre patator puisse lui même envoyer des demande d'appairage c'est pourquoi notre choix s'est porté sur la carte bluetooth HC-06.



Notre matériel :

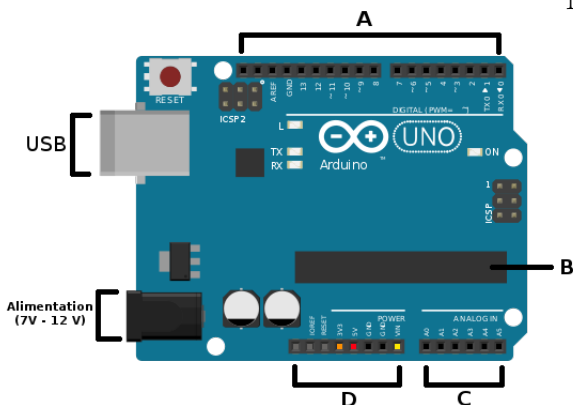
- La carte Arduino Uno;
- Le module bluetooth HC-06;
- Breadboard;

3.1.2 Principe de fonctionnement

La carte Arduino UNO

L'Arduino Uno est un microcontrôleur qui permet de contrôler des éléments mécaniques (LED, moteur, écran, ...). C'est un système très simple d'utilisation qui permet de réaliser tout ce qui nous passe par la tête : création de radars, de drones, de robots, systèmes d'alarmes, etc.

La carte Arduino est également munit de deux ports. Tout d'abord un port USB permettant d'alimenter la carte et de téléverser les programmes dans le microcontrôleur. Et un port d'alimentation (7V – 12V) qui permet d'alimenter l'arduino une fois en fonction (c'est-à-dire lorsque qu'elle n'est pas reliée à l'ordinateur). Enfin nous pouvons trouver des micro ports (appelés pins) différent que nous allons détailler à partir du schéma suivant :



Notre matériel :

- **A** : ce sont les pins digitales (0,1) ou "tout ou rien" car elles offrent en sortie du 0V ou du 5V et acceptent en entrée du 5V sur le même principe
- **B** : le microcontrôleur
- **C** : il s'agit des pins analogiques qui offrent en sortie une valeur entre 0V et 5V
- **D** : différents pins d'alimentation :
 - Rouge : sortie 5v (+)
 - Orange : sortie 3,3V (+)
 - Noire : les masses (-)
 - Jaune : entrée reliée à l'alimentation (7V-12V)

FIGURE 3.3 – Schéma d'une Arduino UNO

La carte bluetooth HC-06

Le module HC-06 possède 4 broches :



FIGURE 3.4 – Notre carte HC-06

- **RX** : broche de réception, connectée au pin de transmission (TX) de l'Arduino
- **TX** : broche de transmission, connectée au pin de réception (RX) de l'Arduino
- **GND** : masse, connectée au pin GND de l'Arduino
- **+5V** : plus souvent appelé VCC, broche de l'alimentation, connectée au pin 3,3V ou 5V de l'arduino

3.2 Premières manipulations, les bases de l'arduino

Quand nous avons commencé ce projet, aucun de nous n'était familier avec l'arduino. Nous avons donc dut tout apprendre en partant des systèmes les plus simples à faire : allumer des LED.

3.2.1 Allumer une LED

Nous avons réalisé un montage très simple (Voir schéma du montage en annexe 3.17 page 32) qui nous permettait d'allumer une LED.

Pour se faire, nous avons connecté la cathode au GND de la carte arduino et l'anode au pin digitale 13. Ainsi, pour allumer la LED, il fallait que notre Arduino délivre 5V au pin 13; c'est le programme qui est chargé d'indiquer à la carte quel pin alimenter. Nous avons utilisé le programme Blink, disponible dans la bibliothèque intégrée au logiciel Arduino IDE, qui allumait et éteignait la LED toutes les secondes.

3.2.2 Allumer plusieurs LED grâce aux touches du clavier

La deuxième étape était d'échanger des informations avec notre carte arduino. Pour cela nous avons trouvé un montage (Voir schéma du montage en annexe 3.18 page 33) permettant d'allumer différentes LED avec les touches du clavier (Vous trouverez le code que nous avons utilisé en annexe 3.3.2 page 34).

Ainsi il nous suffisait de rentrer soit « R », « J », « V » et « B » pour allumer respectivement les LED rouge, jaune, verte et bleue, ou « E » pour éteindre les LED. Nous envoyions les informations à l'arduino grâce au moniteur série du logiciel Arduino IDE.

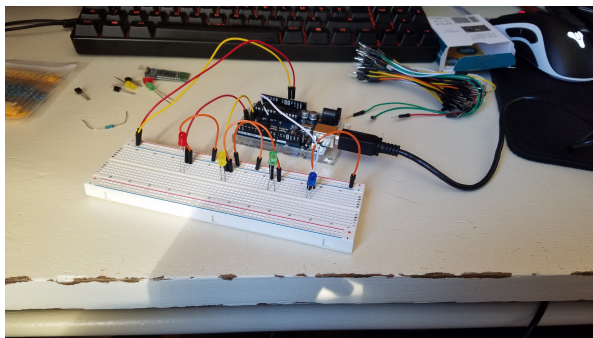


FIGURE 3.5 – Notre montage, toute LED éteintes

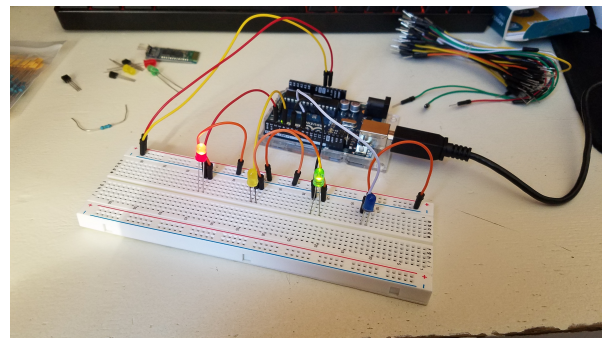


FIGURE 3.6 – LED rouge et verte allumées

3.3 Commande à distance

3.3.1 Allumer une LED à distance

Encore une fois vous trouverez un schéma de notre montage en annexe 3.19 page 35.

Nous avons commencé par chercher des programme sur internet pour allumer une LED à distance, mais aucun ne compilait. C'est pourquoi nous avons décidé de faire notre propre programme que voici :

```

#include <SoftwareSerial.h>
SoftwareSerial hc06(2,3); //pour definir les ports RX et TX servant a la
    communication
int info;

5 //Ce bloc sert a initialiser le Pin 13 comme le pin vers la LED et a
    initialiser la vitesse de communication de la carte HC-06
void setup(){
    pinMode(13, OUTPUT);
    hc06.begin(9600);
}

10 //Ce bloc est une boucle qui dans un premier temps lit les informations
    envoyer par la carte bluetooth, puis si l'info est 1 alors elle allume
    la LED si c'est 0 elle l'eteint
void loop(){
    while(hc06.available()>0){
        if(info==2){
15         delay(10);
        }
        info=hc06.read();
    }
    if(info==1){
20     digitalWrite(13, HIGH);
    }
    else if(info==0){
        digitalWrite(13, LOW);
    }
25     info=2;
}

```

SoftwareSerial hc06(2,3) permet d'informer l'arduino que ses ports RX et TX sont les pins 2 et 3. Ainsi la carte sait quel pin regarder pour recevoir les informations de la carte bluetooth. Notre variable info est un integer qui reçoit 1 ou 0 et qui respectivement allume et éteint la LED avec les commandes digitalWrite(<pin>, <HIGH ou LOW>), HIGH permettant de délivrer 5V et LOW de délivrer 0V.

Nous avons donc testé notre montage et notre code avec l'application que nous avons développé. La LED s'allume lorsqu'on envoi 1 et s'éteint lorsqu'on envoi 0, tout fonctionne parfaitement.

Notre système de commande à distance fonctionne, il ne nous restait plus qu'à tester

notre montage théorique. Pour ce faire nous avons commencé par étudier les AOP.

3.3.2 Fonctionnement de l'AOP

L'amplificateur Opérationnel étant un élément central dans notre montage, nous devons tout d'abord nous assurer que nous saurions l'utiliser.

Nous avons utilisé les datasheets des 3 AOP que nous avons (voir bibliographie). Nos 3 AOP avaient 8 pattes selon 2 configurations possibles :

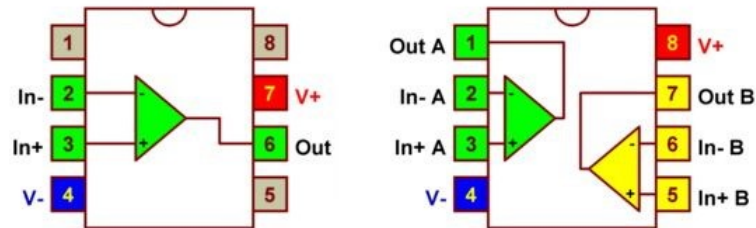
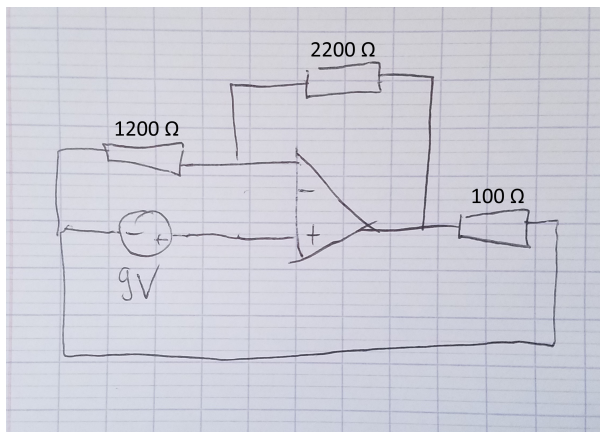


FIGURE 3.7 – Branchements interne de nos AOP

Nous avons réalisé différents montages, voici quelques schémas et photos :



Notre AOP était alimenté en 18V grâce à 2 piles 9V branchées en série. Nous avons en entrée 9V et devons théoriquement avoir 24V en sortie de l'AOP. Seulement, le meilleur résultat que nous ayons eu était de 3,5V en sortie.

FIGURE 3.8 – Schéma théorique du montage de test

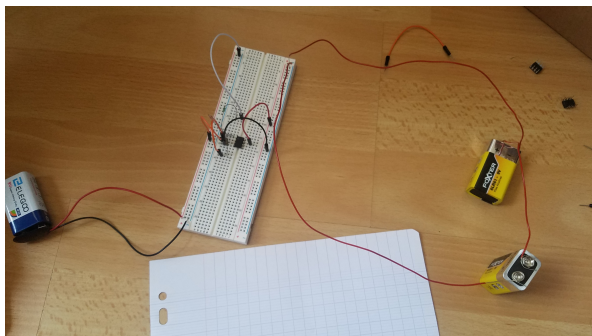


FIGURE 3.9 – Montage AOP

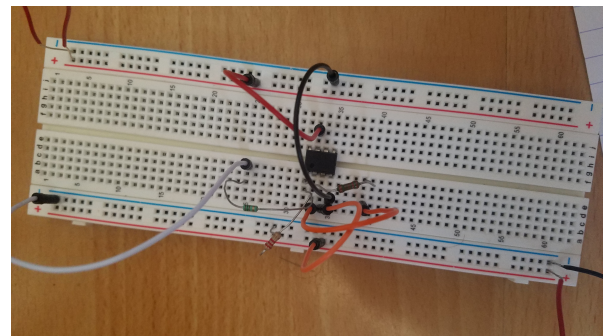


FIGURE 3.10 – Montage AOP

En somme nous n'avons pas réussi à avoir de résultats concluants : soit nous avons 0V en sorti de notre AOP, soit nous avons une tension quasi égale à celle en entrée et très faible

face au résultats théoriques attendues.

Nous n'avons donc pas put aboutir le projet, cependant si nous avons réussi à faire fonctionner l'AOP et à avoir ainsi 24V en sortie de celui-ci, nous aurions put facilement faire fonctionner l'électrovanne.

Conclusion et perspectives

Afin de conclure, nous aimerions revenir sur les connaissances que nous avons mis en application et sur les compétences que nous avons développées. Nous nous sommes confrontés à la création d'un circuit devant répondre à des exigences précises avec nos propres moyens, de la conception théorique au choix des pièces en passant par la modélisation sur le programme. Nous avons également appris à utiliser une carte Arduino, à réaliser des montages à partir de celle-ci et à apprendre à la programmer à l'aide d'un langage C que nous ne maîtrisions que peu. Enfin, nous avons découvert les différents moyens de développer une application, nous avons acquis les bases de programmation en Java ainsi que de l'utilisation d'Android Studio et nous avons réussi à produire une application simple.

Néanmoins, durant ce projet nous avons appris une grande leçon, à savoir qu'il n'est tout simplement pas possible d'être prêt à toute éventualité. Dans notre cas, l'imprévu fut principalement la pandémie actuelle et les complications qu'elle a apportées. Cependant, cette situation nous a forcés à trouver des moyens pour nous adapter afin de pouvoir présenter le travail le plus abouti et intéressant possible.

Cela a notamment pu mettre à l'épreuve l'organisation au sein du groupe, nous avons effectivement dû trouver de nouveaux moyens pour communiquer et nous entraider, mais nous avons également eu besoin d'apprendre à nous faire confiance pour avancer. Ce qui nous a permis au final d'acquérir de l'expérience dans la gestion de l'aspect humain d'un projet, et ce dans des conditions relativement catastrophiques.

Enfin, nous sommes fiers de notre projet et du résultat que nous avons réussi à vous proposer. Nous espérons pouvoir monter dans l'avenir un système de propulsion automatisée de légumes par déflagration inspiré du design et des leçons résultant de ce projet.

Bibliographie

- [1] V.M. SHMELEV, "On Minimum Energy of Ignition of Methane-Air mixtures", Russian Journal of Physical Chemistry B, Vol 3 N°1, p75-81, 2009.
- [2] COURS SUR LES TRANSISTORS , <http://robert.cireddu.free.fr/SI/Cours%20sur%20les%20transistors.pdf> (Valide à la date du 15/04/2020)
- [3] COURS SUR LES AMPLIFICATEURS OPÉRATIONNELS, <https://melusine.eu.org/syracuse/immae/mpsi/physique-chimie/electrocinetique/05.pdf> (Valide à la date du 03/05/2020)
- [4] COMMENT GÉNÉRER UNE ÉTINCELLE AVEC UN MONTAGE ARDUINO , <https://electronics.stackexchange.com/questions/312828/generate-electric-spark-with-arduino-high-voltage-boost-converter> (Valide à la date du 03/05/2020)
- [5] FICHE DES ELECTROVANNES BURKERT, https://www.burkert.fr/fr/content/download/6978/228998/file/01_Electrovannes_FR_version%20light.pdf?id=95 (Valide à la date du 08/05/2020)
- [6] ELECTROVANNE : FONCTIONNEMENT, TERMINOLOGIE ET TYPES DE CONSTRUCTION, https://www.asconumatics.eu/images/site/upload/_fr/pdf1/00005fr.pdf (Valide à la date du 10/04/2020)
- [7] VANNE ÉLECTROMAGNÉTIQUE OU SOLÉNOÏDE, <https://www.abcclim.net/vanne-electromagnetique-solenoid.html> (Valide à la date du 21/05/2020)
- [8] COMMENT FAIRE SES PROPRES APPLICATIONS POUR DES PROJETS ARDUINO EN UTILISANT MIT APP INVENTOR, <https://howtomechatronics.com/tutorials/arduino/how-to-build-custom-android-app-for-your-arduino-project-using-mit-app-inventor/> (Valide à la date du 20/04/2020)
- [9] ALLUMER DES LED AVEC LES TOUCHES DU CLAVIER : <https://fr.flossmanuals.net/arduino/console-de-lumiere/> (Valide à la date du 19/04/2020)
- [10] DATASHEET DE L'AOP TL081CP : https://www.alldatasheet.com/view.jsp?Searchword=TL081cp%20datasheet&gclid=Cj0KCQjwoPL2BRDxARIsAEMm9y9tyZxGQom7y1_1lw2hw4YgTtYG7JsX3_73TvgEarZfunVycj507aoaAhtvEALw_wcB (Valide à la date du 18/05/2020)
- [11] DATASHEET DE L'AOP TL082CN : https://www.alldatasheet.com/view.jsp?Searchword=TL082cn%20datasheet&gclid=Cj0KCQjwoPL2BRDxARIsAEMm9y-1yKdzNGcD3az6Nu2FZy5_qPVHivw-SciWeckQFkGwb6TA2281UnQaAkIdEALw_wcB (Valide à la date du 18/05/2020)

[12] DATASHEET DE L'AOP TL082CP :

<https://www.alldatasheet.com/view.jsp?Searchword=TL082CP>

(Valide à la date du 18/05/2020)

Annexes



FIGURE 3.11 – Photo d’une électrovanne Type 470 de la marque Burkert

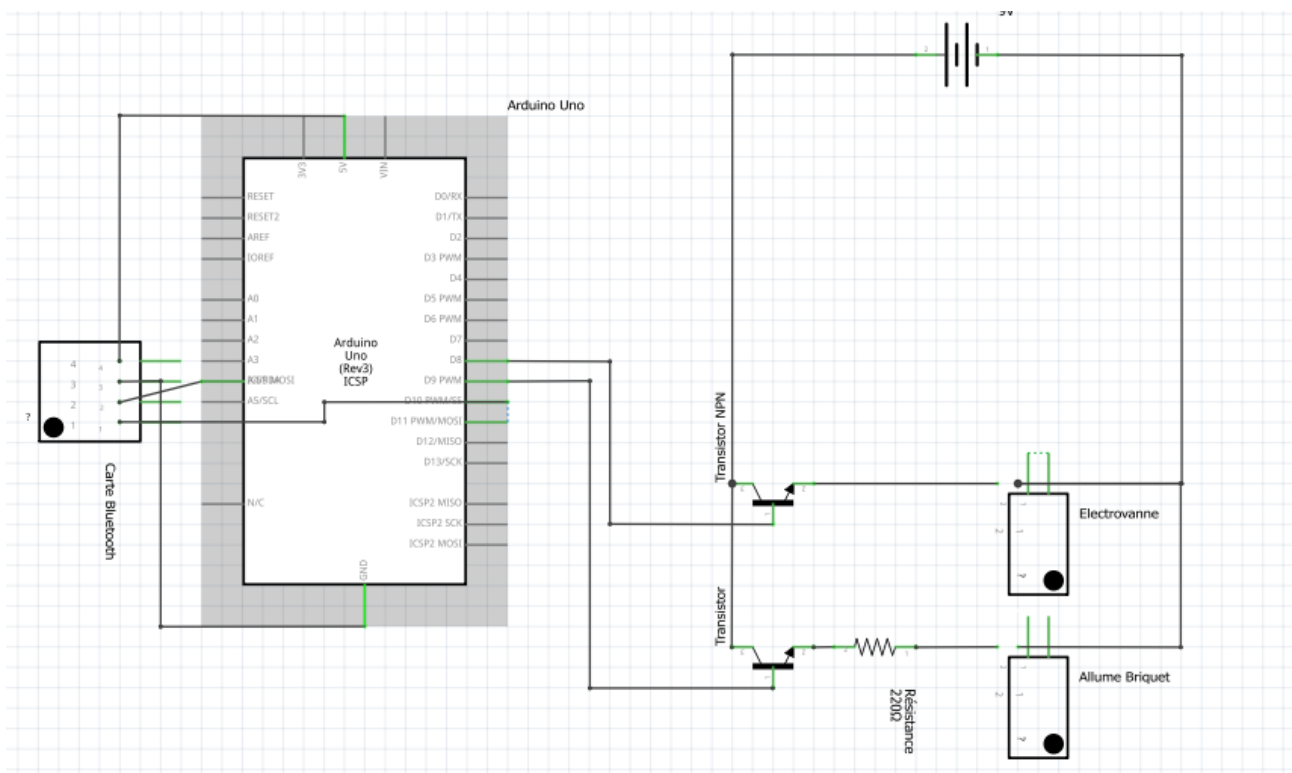


FIGURE 3.12 – Ébauche du schéma du circuit électrique

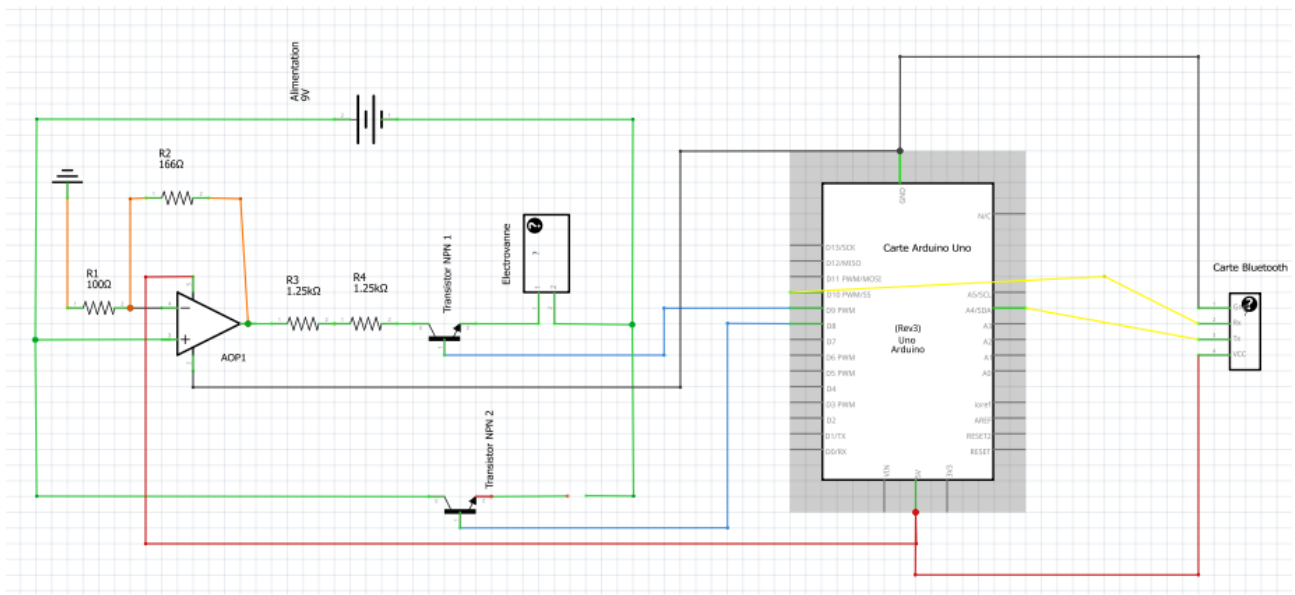


FIGURE 3.13 – Schéma du montage amplificateur

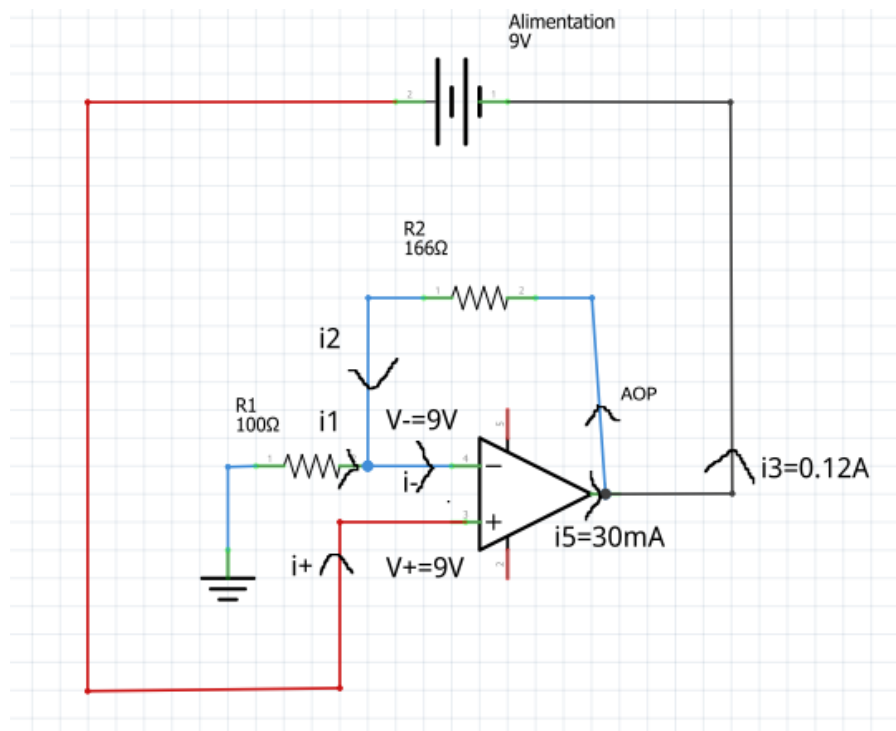
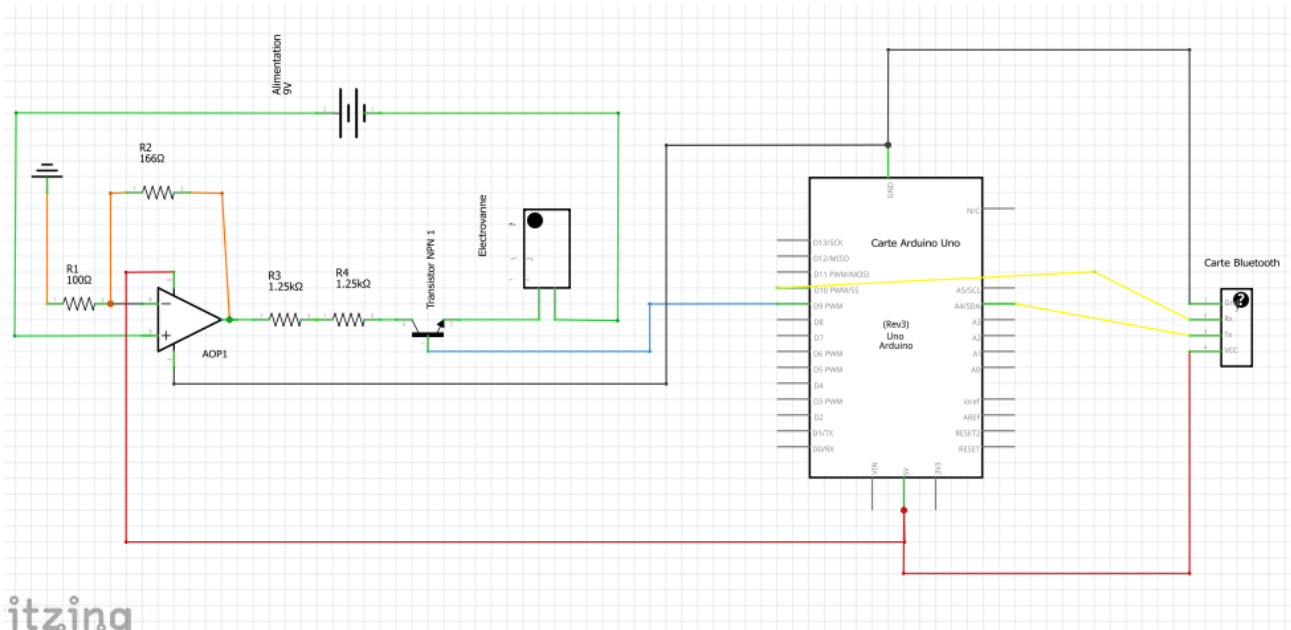
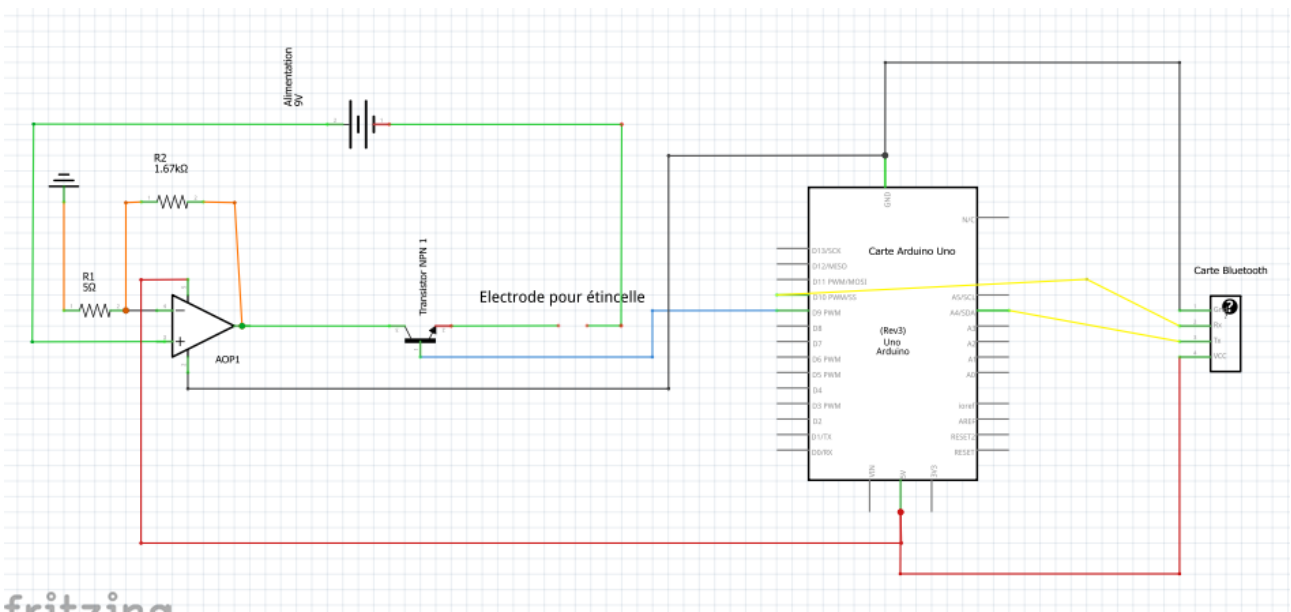


FIGURE 3.14 – Schéma détaillé du montage amplificateur utilisé pour les calculs



itzing

FIGURE 3.15 – Schéma du montage théorique permettant l’injection contrôlée à distance du gaz

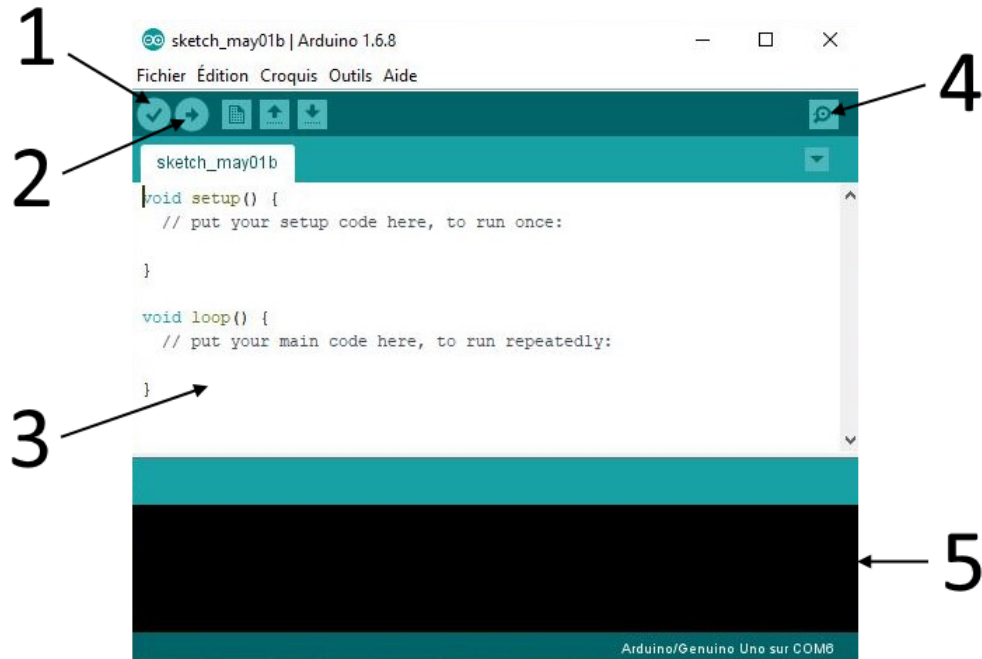


fritzing

FIGURE 3.16 – Schéma du montage théorique permettant l’étincelle contrôlée à distance

Le logiciel Arduino IDE

Il s'agit du logiciel le plus utilisé pour faire de l'Arduino. Celui-ci permet de coder un programme et de le téléverser sur la carte connectée à l'ordinateur. Il propose également une bibliothèque assez complète de programmes pré-faits, nous en utiliserons un dans la partie 3.2.1. Voici l'interface détaillée du logiciel :



- **1** : permet de compiler le programme pour vérifier qu'il n'y a pas d'erreur
- **2** : permet de téléverser le programme sur la carte Arduino connectée à l'ordinateur; il est important de préciser que lorsqu'on téléverse, le logiciel va automatiquement compiler le programme et nous donner les messages d'erreur
- **3** : zone pour écrire le programme
- **4** : ouvre un moniteur série qui permet d'envoyer des informations à l'Arduino et d'afficher les informations reçues ou envoyées par l'Arduino (nous l'utiliserons dans la partie 3.2.2)
- **5** : console qui permet d'afficher les erreurs rencontrées lors de la compilation

Allumer une LED

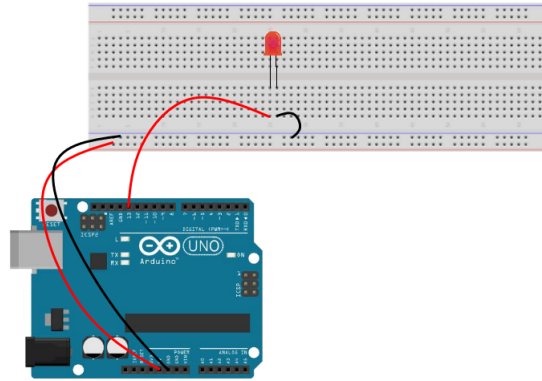


FIGURE 3.17 – Schéma du montage Arduino pour allumer une LED

Allumer plusieurs LED grâce aux touches du clavier

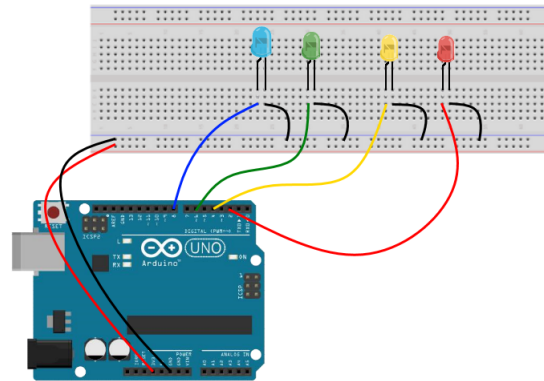


FIGURE 3.18 – Schéma du montage Arduino pour allumer plusieurs LED à partir des touches du clavier

```

//Declaration des pin sur lesquelles sont connectees les LED
int pinLed3 = 8; // LED Bleue
int pinLed2 = 6; // LED Verte
int pinLed1 = 4; // LED Jaune
5 int pinLed0 = 2; // LED Rouge

//Declaration de la variable contenant la valeur de la touche saisi au
  clavier
int octetRecu;
void setup() {
10
    // Initialisation de la communication serielle
    Serial.begin(9600);

    // Configuration des pin en sortie
15 pinMode(pinLed0, OUTPUT);
    pinMode(pinLed1, OUTPUT);
    pinMode(pinLed2, OUTPUT);
    pinMode(pinLed3, OUTPUT);
}
20 void loop() {
    // Verifie si il y a une donnee serielle disponible
    if (Serial.available() > 0) {
        // Lecture de l'octet present dans le buffer
        octetRecu = Serial.read();
25
        if (octetRecu == 'R' || octetRecu == 'r') { //Si l'octet recu est
            egal a R ou r
            digitalWrite(pinLed0, HIGH); //Allumer la LED connectee a
                pinLed0
            Serial.println("LED Rouge allumee"); //Afficher "LED Rouge
                allumee" dans le moniteur serie
        }
    }
}

```

```

30  if (octetRecu == 'J' || octetRecu == 'j') { //Si l'octet reçu est
    egal a J ou j
    digitalWrite(pinLed1, HIGH); //Allumer la LED connectee a
    pinLed1
    Serial.println("LED Jaune allumee"); //Afficher "LED Jaune
    allumee" dans le moniteur serie
    }
35  if (octetRecu == 'V' || octetRecu == 'v') { //Si l'octet reçu est
    egal a V ou v
    digitalWrite(pinLed2, HIGH); //Allumer la LED connectee a
    pinLed2
    Serial.println("LED Verte allumee"); //Afficher "LED Verte
    allumee" dans le moniteur serie
    }
40  if (octetRecu == 'B' || octetRecu == 'b') { //Si l'octet reçu est
    egal a B ou b
    digitalWrite(pinLed3, HIGH); //Allumer la LED connectee a
    pinLed3
    Serial.println("LED Bleue allumee"); //Afficher "LED Bleue
    allumee" dans le moniteur serie
    }
45  if (octetRecu == 'E' || octetRecu == 'e') { //Si l'octet reçu est
    egal a E ou e
    //Eteindre les LED connectees aux pinLed0, pinLed1, pinLed2 et
    pinLed3
    digitalWrite(pinLed0, LOW);
    digitalWrite(pinLed1, LOW);
50  digitalWrite(pinLed2, LOW);
    digitalWrite(pinLed3, LOW);
    //Afficher "LED eteinte" dans le moniteur serie
    Serial.println("LED eteinte");
    }
55  }
}

```

Allumer une LED en bluetooth

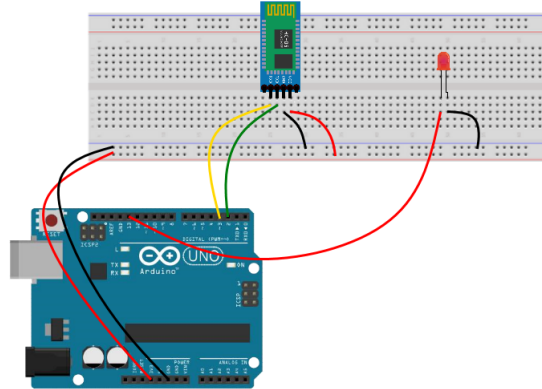


FIGURE 3.19 – Schéma du montage arduino permettant d'allumer une LED à distance

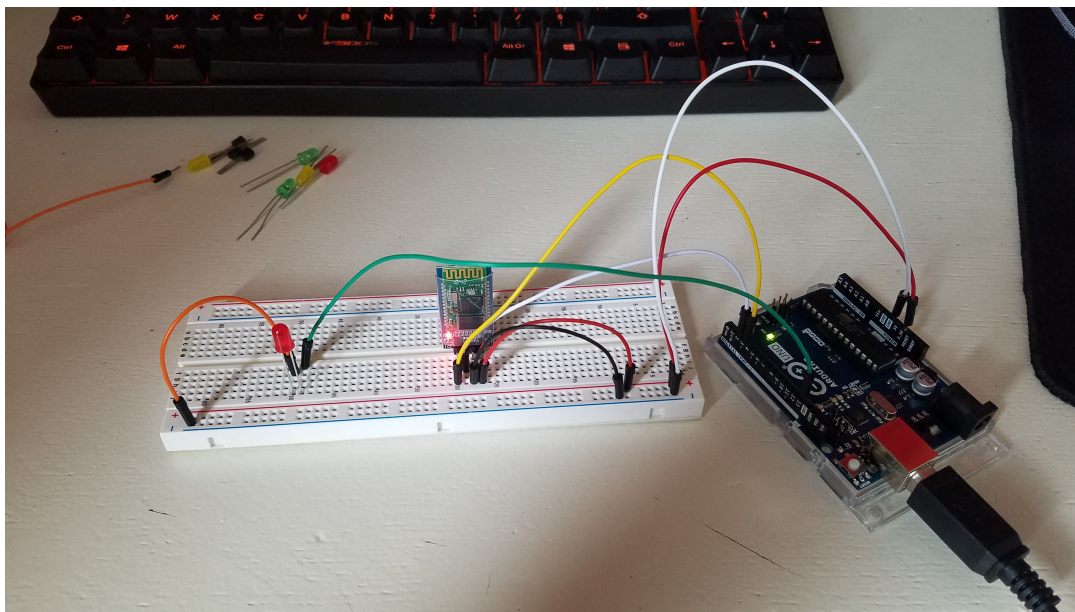


FIGURE 3.20 – Montage pour l'allumage d'une LED à distance