**Projet « smart-patate » :**

1-    Contexte du projet

2-    Résultat des expériences menées

3-    Explication du fonctionnement du circuit et du fonctionnement du capteur

4-    Présentation du prototype

5-    Bilan du projet

1. Contexte :

« Votre mission si vous l’acceptez, sera de réussir à transformer un simple légume en interrupteur intelligent. »

Nous avons un projet de deux jours pendant lequel nous sommes censés créer un capteur sachant différencier trois touchés différent.

1. Résultat des expériences menées :

Expérience 1 :

1. Pour le montage, cf : PowerPoint
2. Utilisation du GraphOscillo

* Avec l’aide du logiciel [Arduino](https://www.arduino.cc/en/Main/Software), du programme [PWMFreq4](https://sts.viacesi.fr/adfs/ls/), nous avons pu avec la visualisation du signal sous le programme [GraphOscillo](https://moodle-exia.cesi.fr/mod/resource/view.php?id=9345) pu observer les variations du signal en fonction de si notre doigt touchait l’électrode.
* Suite à l’utilisation du programme, on peut donc en tirer un tableau :
* Les résultats du tableau ont été observés grâce au programme qui donnais cette fenêtre de résultats. Voici quelques exemples. La partie entre les barres orange indique le moment ou un contact est effectué

1. Réalisation d’une courbe avec Scilab

* Avec l’aide du logiciel nous devions créer une courbe à l’aide du logiciel Scilab. Pour écrire des points dans un tableau avec ce logiciel, il faut utiliser la commande plot en suivant l’exemple suivant :

--> plot (500, 5, ʺ.rʺ)

* Le premier chiffre entré est l’abscisse. Pour notre courbe, nous avons choisi de représenter la tension en fonction de la fréquence. Le second chiffre est séparé par une virgule du premier et symbolise l’ordonnée. Enfin, la dernière information apportée à cette commande est le « .r ». Il indique la couleur du point, ici rouge. Par conséquent, « .b » représentera un point en bleu, etc…
* Voici un exemple du code rentré sous le logiciel, ainsi que de la courbe. Les point en rouges représentent l’action du touché (tension max et min) et les points rouges représentent la tension normale (tension max et min).

1. Réponse aux questions

* Pourquoi parle-t-on de capteur capacitif ?

On peut parler de capteur capacitif car celui détecte dans notre cas une conductivité du courant à l’aide de notre main.

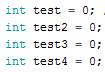
Le capteur que l’on « possède » aux mêmes propriétés que le capteur capacitif. En effet, il détecte à une très faible distance un matériau non métallique : la main.

* Déduisez-en quel composant le corps humain remplace dans le montage.

Le corps humain remplace une résistance et la bobine, car lorsque l’on touche le fil, l’état (la tension) change.

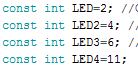
* Pourquoi le signal diminue-t-il ?

Car quand on ajoute une résistance à un circuit, la tension diminue.

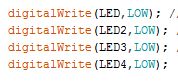
1. Explication du fonctionnement du circuit et du fonctionnement du capteur :

Dans cette partie du code on initialise nos quatre LED, la LED témoin (test4) et

les trois autres LED : test🡪1 doigt/test2🡪2 doigts/test3🡪à pleine main.



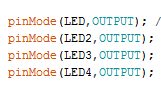
Ici on définit les broches sur lesquelles sont branchées les trois LED.



Ici, étant donné que la LED est configuré en sortie, on utilise cette fonction

pour dire si la LED est à un état haut ou bas. Ici, c'est l'état bas,

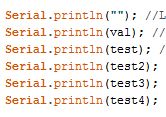
la LED est éteinte.



Ici on configure les LED comme étant des sorties.

https://scontent-lhr3-1.xx.fbcdn.net/v/t34.0-12/14971969_1170574589722847_354047847_n.png?oh=db3a5247b41bd6ef8abd6d559669bfb0&oe=582EA09B

Ici on précise qu’on utilise 200K en fréquence.

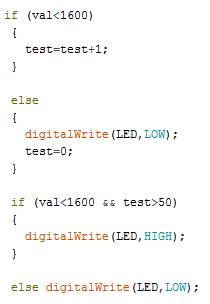


La Fct Serial.println affiche les données sur le port série.

Ici, cela permet d'envoyer une valeur sous forme d'octet simple.

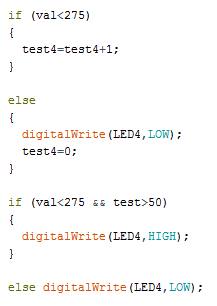
Affiche test en tant que nbr decimal sous la forme d'une chaîne de caractère

ASCII suivi d'un retour de chariot et d'un saut de ligne pour les trois tests.



Voilà le code de la LED témoin, c’est la LED verte qui nous sert de témoin,

elle reste allumée pour vérifier que la carte Arduino reçoive du courant.



Voilà le code de la dernière LED à s’éteindre, celle qui s’éteint lorsque l’on

prend la patate à pleine main.

( cf : Git pour le programme complet avec les commentaires )

1. Présentation du prototype :

A : Fonctionnement de notre interrupteur :

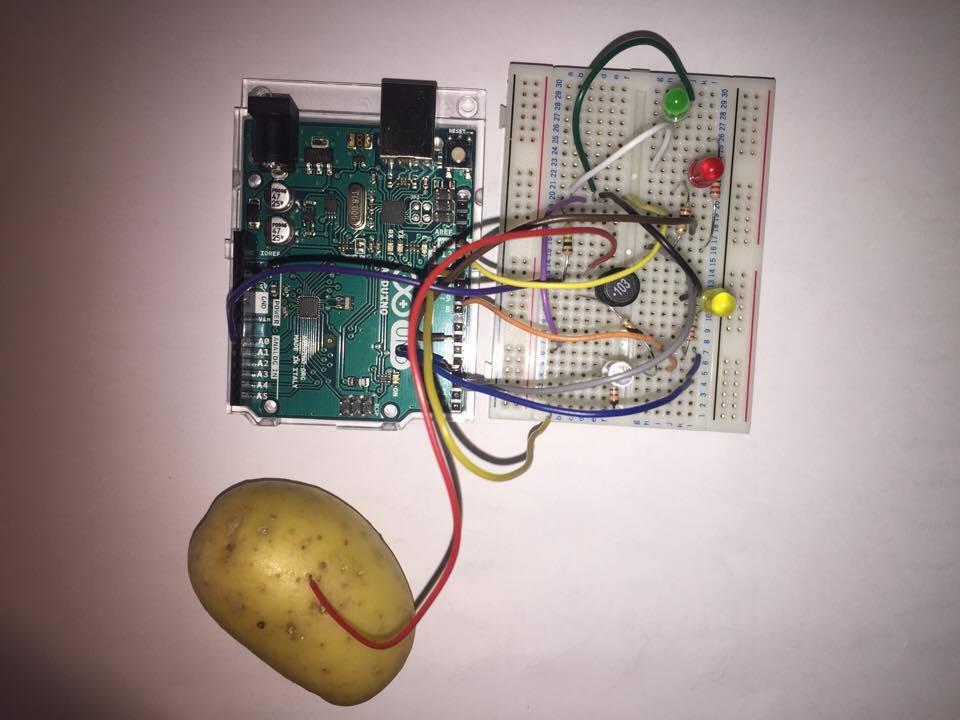
1a) Lorsque l’on n’appuie pas en continu sur la patate les trois LED sont allumées (la LED témoin est aussi allumée mais l’on considère qu’elle reste tout le temps allumée).

2a) Lorsque l’on touche la patate avec un doigt la LED rouge s’éteint.

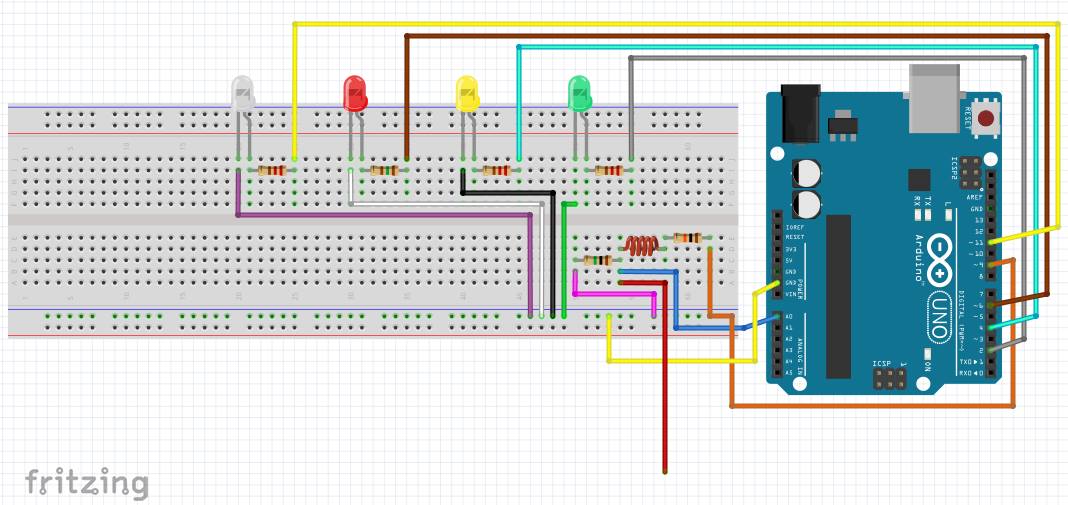
3a) Lorsque l’on fait de même avec deux doigts la LED rouge et la LED colorée jaune s’éteignent.

4a) Lorsque l’on prend la patate à pleine main les trois LED (rouge/verte et blanche) s’éteignent.

B : Notre prototype :



*(Notre prototype d’interrupteur intelligent)*



Bilan :