



ECOLE DES MINES DE SAINT ETIENNE
CAMPUS G. CHARPAK PROVENCE

RAPPORT - PROJET PROTOTYPAGE

Documentation

Maïssa ELAFANI
Nour MHANNA

17 novembre 2024

Remerciements

Nous tenions à remercier M. Blayac de nous avoir guidé durant les séances de prototypage en nous expliquant le principe de fonctionnement du conditionneur et M. Alsaadoui afin de nous avoir aidé à maîtriser les logiciels COMSOL.

Nous remercions également M. Bernier et M. Delattre d'avoir contribué à la mise en place de notre projet avec des conseils et des commentaires toujours très précieux pour réussir.

Table des matières

1	Introduction	3
2	Compréhension du projet	4
2.1	Contexte	4
2.2	Principe de fonctionnement	5
3	Présentation de la synoptique du système	6
4	Modélisation du système capacitif	7
4.1	Le circuit conditionneur	7
4.1.1	La charge à courant constant	8
4.1.2	Le convertisseur courant-tension	9
4.1.3	Le comparateur à hystérésis et le fonctionnement en boucle fermé .	10
4.2	Modélisation et simulation Comsol du système	11
5	Mise en œuvre du système capacitif	13
5.1	Description des composants du système	13
5.1.1	Microcontrôleur	13
5.1.2	Conditionneur	14
5.1.3	Écran OLED	14
5.1.4	Bouton poussoir	15
5.2	Réalisation du socle accueillant les électrodes	15
5.3	Programmation du microcontrôleur	16
6	Améliorations possibles	18
7	Conclusion	19
8	Annexes	20

1 Introduction

Les technologies capacitives et particulièrement les capteurs capacitifs offrent plusieurs avantages tels qu'une sensibilité accrue, une réponse rapide aux interactions en temps réel, une certaine fiabilité mais aussi une simplicité de conception qui rend leur utilisation intéressante dans de nombreux domaines.

Dans le cadre du projet Prototypage du semestre 8 à l'École des Mines de Saint-Étienne, nous avons pu concrétiser les principes de fonctionnement des capteurs capacitifs en ayant comme objectif de concevoir et de mettre en place un système novateur exploitant cette technologie.

Pour répondre à cette problématique, nous avons décidé de réaliser une boîte à votes intelligente qui comptabilise le nombre de bulletins de vote inséré en temps réel.

Au sein de ce rapport, vous trouverez une présentation du système en détaillant son utilité ainsi que son fonctionnement. Puis, nous décrirons la modélisation du système ainsi que les simulations effectuées. Enfin, nous décrirons la mise en œuvre de notre projet.

2 Compréhension du projet

2.1 Contexte

"Le principe de souveraineté réside essentiellement dans la nation."

Article 3 de la Déclaration des droits de l'homme et du citoyen

Le droit de vote est un droit fondamental dans tous les gouvernements démocratiques. Il est accordé aux citoyens pour exprimer leur volonté à l'occasion d'un scrutin et nous permet de participer activement au processus démocratique afin de faire entendre notre voix. Le vote représente alors un enjeu important et il est de notre devoir de faire en sorte qu'il soit respecté comme tel.



FIGURE 1 – Boite à votes classique

Malheureusement, les moyens mis en place ne sont pas forcément adéquats à l'importance du droit de vote et seulement un décompte des votes est fait en général et les études montrent qu'il existe une erreur humaine d'environ 1% dans un environnement contrôlé.

En France, lors des élections présidentielles en 2022, 35 923 707 personnes auraient votés d'après le journal "Le Monde" ce qui représente en réalité un intervalle compris entre [35 564 469, 36 282 944] personnes.

C'est avec cette pensée en tête que nous avons décidé de créer "Smart Vote", une boîte à votes intelligente qui permet de comptabiliser en temps réel le nombre de bulletins

de vote dans la boîte. Cela permet ainsi de compter en amont le nombre de vote avant d'effectuer une deuxième fois un décompte à "échelle humaine".

Par ailleurs, cette boîte peut être utilisé dans de multiples contextes qui requiert le décompte de ce qui y est présent. Et notre projet se montre donc utile et nécessaire pour de nombreuses fonctions.

2.2 Principe de fonctionnement

Pour mettre en place notre système, nous avons choisi de placer deux électrodes dans la fente du couvercle de la boîte dans la disposition suivante :

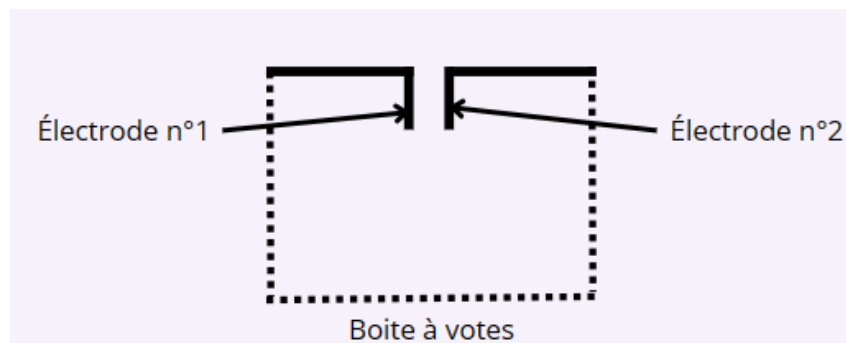


FIGURE 2 – Schéma général du système

Or, la formule régissant la capacité d'un condensateur est la suivante :

$$C = \epsilon_r C_0 \quad (1)$$

avec C_0 , la capacité entre les plaques dans diélectrique et ϵ , la permittivité relative

L'objectif en mettant en place ce dispositif est que lorsqu'une enveloppe contenant un bulletin de vote passera entre les deux électrodes, la permittivité relative variera car elle ne sera plus seulement égale à la permittivité relative du vide mais à celle du vide et de la feuille cartonnée représentant le bulletin de vote. Cela entraînera la variation de la capacité et donc de la fréquence et de la période. La détection de ce changement de période permettra la détection de la présence de l'enveloppe entre les deux électrodes permettant ainsi la mise en place de notre système.

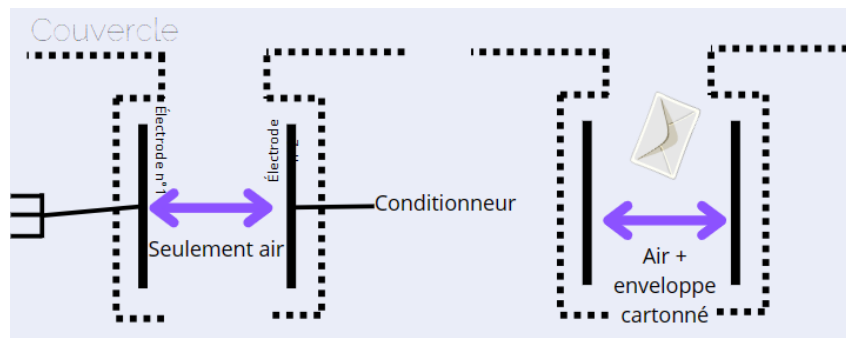


FIGURE 3 – Schéma explicitant le fonctionnement du capteur

3 Présentation de la synoptique du système

Dans cette partie, nous analyserons les différents éléments composant notre application.

Le circuit électronique basique est composé d'un conditionneur, d'un microcontrôleur. À ce système, nous avons ajouter un bouton poussoir et un écran OLED facilitant la gestion de l'interface pour l'utilisateur.

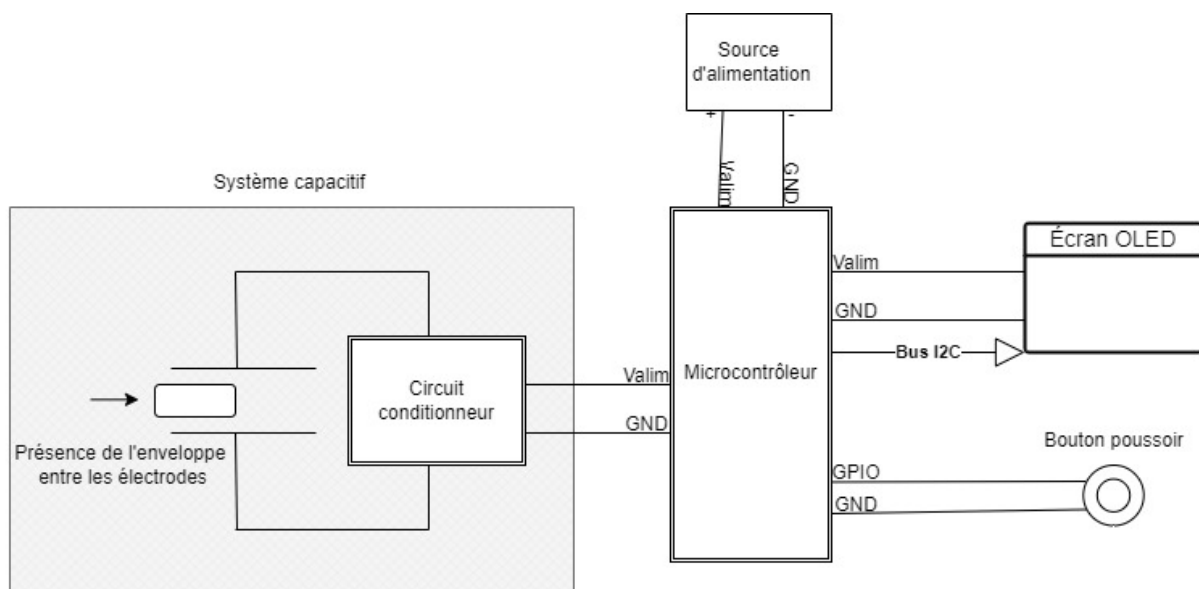


FIGURE 4 – Synoptique du circuit

Le système se décompose comme suit :

- Le capteur qui est composé des deux électrodes détecte les votes entrants dans la fente de la boîte.
- Le conditionneur traite le signal brut du capteur et pourra amplifier le signal, le filtrer ou le convertir en une forme plus adaptée à un traitement ultérieur.

- Le microcontrôleur analyse le signal conditionné. Il prendra des décisions en fonction des informations reçues et du programme implémenté au sein de ce dernier.
- L'écran OLED qui permettra d'afficher l'écran d'accueil ainsi que le nombre de votes en temps réel.
- Le bouton poussoir qui permettra le démarrage du compteur et agira comme un bouton reset pour ré-initialiser le compteur.

4 Modélisation du système capacitif

4.1 Le circuit conditionneur

La variation de capacité captée par les électrodes est mise en forme par un circuit conditionneur permettant la transduction en un signal électrique interprétable par un microcontrôleur et dans notre cas, de la fréquence d'un signal carré.

Dans le cadre de ce projet, les capacités des électrodes sont d'environ 10 pF avec des variations de l'ordre de quelques pF. Par conséquent, il est nécessaire d'utiliser des technologies d'acquisition compatibles avec ces ordre de grandeur. Ainsi, deux approches se distinguent pour mesurer la capacité : la première repose sur la mesure de la tension tandis que la deuxième se base sur la mesure du temps de charges.

Dans notre cas, nous privilégierons la seconde méthode afin de réduire l'impact des parasites.

Pour ce faire, nous avons alors conçu un circuit conditionneur et avons, en amont, utilisé *Partquest Explore*, un logiciel de conception, de modélisation, de simulation et d'analyse pour les circuits électroniques. Grâce à cette phase de modélisation, nous avons pu simuler le fonctionnement du circuit conditionneur dans diverses conditions.

Par ailleurs, le système conditionneur est composé de plusieurs éléments que l'on analysera dont :

- une charge à courant constant,
- un convertisseur courant-tension
- et un comparateur à hystérésis.

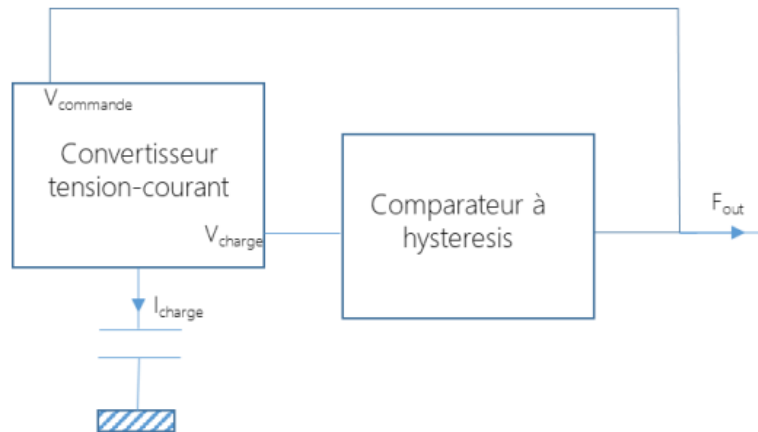


FIGURE 5 – Synoptique du circuit conditionneur

4.1.1 La charge à courant constant

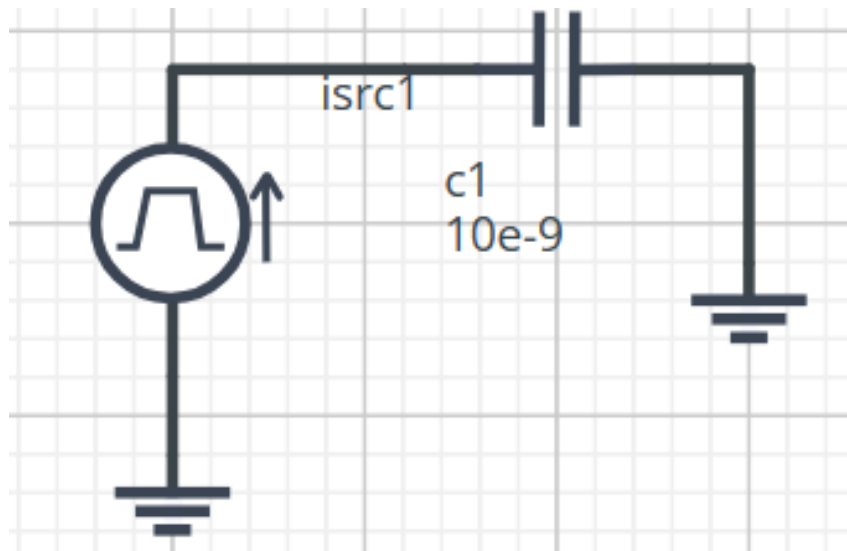


FIGURE 6 – Circuit de la charge à courant constant

Pour étudier la charge à courant constant, on crée un circuit comprenant un générateur de courant ainsi qu'une capacité de 10 pF.

On met, en entrée, un signal carré et on obtient une tension de charge V_{charge} qui est un signal triangulaire.

Par ailleurs, la relation liant le courant et la tension est :

$$I = C \frac{dV}{dt} \quad (2)$$

Et on obtient finalement :

$$V_{charge} = \frac{I}{C} \frac{T_{periode}}{2} \quad (3)$$

On acquiert alors les graphes suivants :

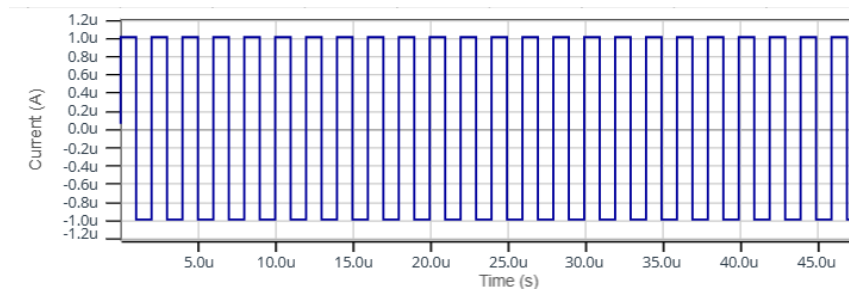


FIGURE 7 – Graphique représentant le courant en entrée du circuit en fonction du temps

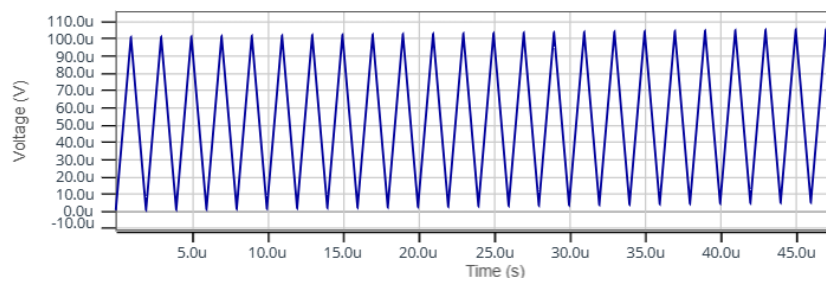


FIGURE 8 – Graphique représentant la tension V_{charge} en sortie en fonction du temps

4.1.2 Le convertisseur courant-tension

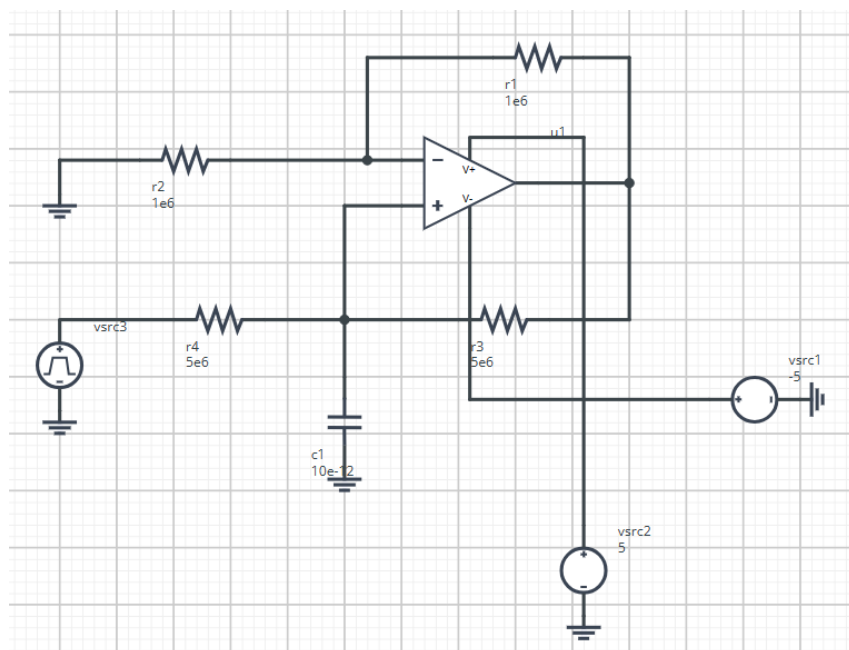


FIGURE 9 – Circuit du convertisseur courant-tension

La fonction principale du convertisseur courant-tension est de réguler le courant de charge traversant la capacité.

En appliquant un pont diviseur de tension et la formule de Millmann, on obtient :

$$i = \frac{V_{in}}{R_1} \quad (4)$$

avec $R_1 = R_2 R_3 = R_4$

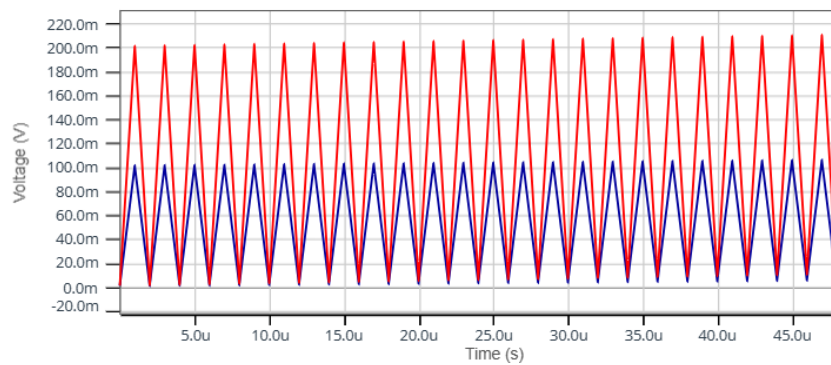


FIGURE 10 – Graphique représentant la tension aux bornes du condensateur (en bleu) et la tension à la sortie de l'amplificateur (en rouge) en fonction du temps

On obtient bien un signal amplifié en sortie du montage.

4.1.3 Le comparateur à hystérésis et le fonctionnement en boucle fermée

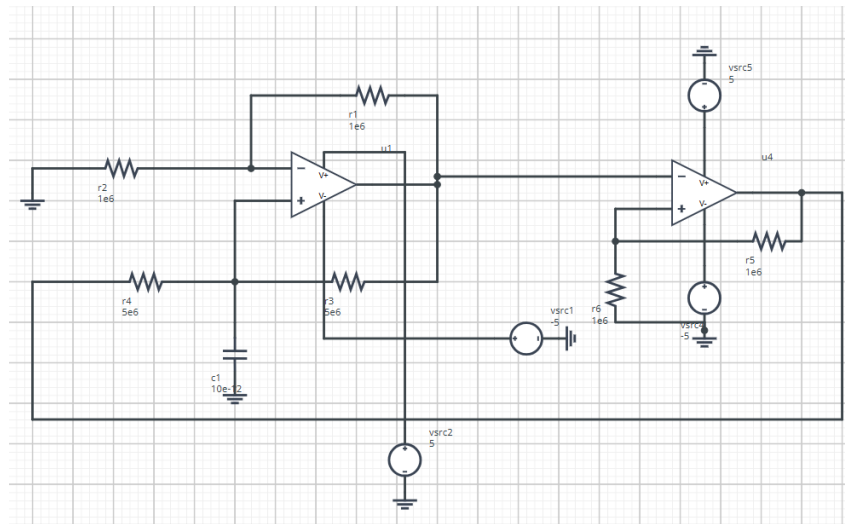


FIGURE 11 – Circuit du comparateur à hystérésis

On ajoute au montage précédent un comparateur à hystérésis.
0.2cm Le comparateur par hystérésis impose une valeur d'alimentation V_{cc} en fonction de son entrée.

Par ailleurs, on obtient la relation suivante :

$$F = C \frac{2}{R_4} \quad (5)$$

Le rôle du conditionneur est donc, effectivement, de traduire la variation de la capacité en une grandeur électrique que le microcontrôleur peut traiter.

4.2 Modélisation et simulation Comsol du système

Comsol est un logiciel permettant la modélisation et la simulation des comportements physiques de dispositifs réels.

Pour ce faire, nous allons étudier la création d'un champ électrostatique et lorsqu'un objet (l'enveloppe dans notre cas) passe entre les deux électrodes, la permittivité relative va varier perturbant ainsi le champ électrostatique. Cette variation va induire une variation de la valeur de capacité et qui pourra être par suite utilisée pour détecter le passage de l'enveloppe.

On sait que la formule qui décrit la capacité créée entre deux électrodes rectangulaires est :

$$C = \frac{\varepsilon_0 \varepsilon_r S}{d} \quad (6)$$

Donc d'après l'équation précédente, on peut constater que le passage de l'enveloppe, qui est un diélectrique, va provoquer une variation de la permittivité relative. Par conséquent, la capacité du milieu va varier.

En fait, avant de passer à la conception, il était nécessaire de réaliser une modélisation de notre idée sur un logiciel pour s'assurer que le dimensionnement des électrodes et de l'ensemble était correct. Cela est dû au fait que le conditionneur possède une capacité parasite qui est branchée en parallèle avec la capacité des électrodes. Pour cela, nous devons nous assurer que la variation de capacité est au moins plus grande que la valeur de la capacité parasite qui est de l'ordre de 8 pF.

Pour cela, nous avons utilisé le logiciel *Comsol*, qui est un outil puissant pour les simulations physiques.

Au début, on a construit la géométrie, puis défini les matériaux utilisés. Ensuite, nous avons choisi les propriétés physiques à utiliser. Et enfin, nous avons choisi le maillage et nous avons calculé notre étude en paramétrage et en étude stationnaire.

Les résultats de l'étude stationnaire ont permis de visualiser la distribution du champ électrique et du potentiel autour des électrodes.

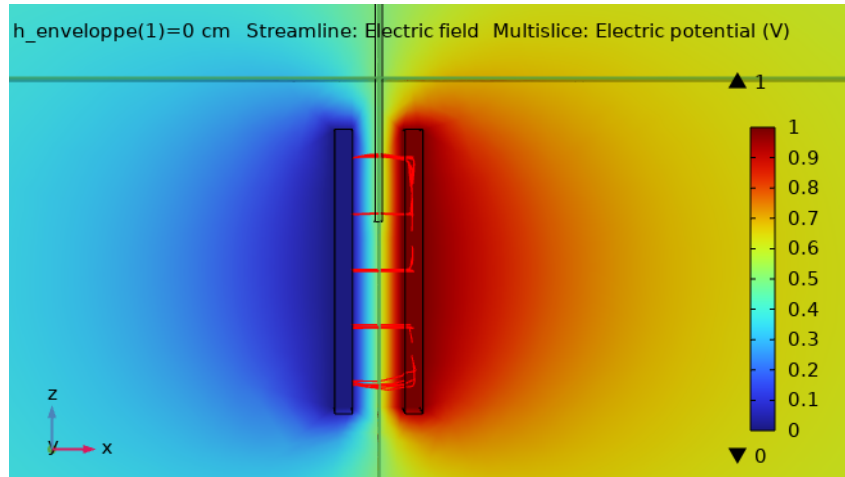


FIGURE 12 – Potentiel électrique

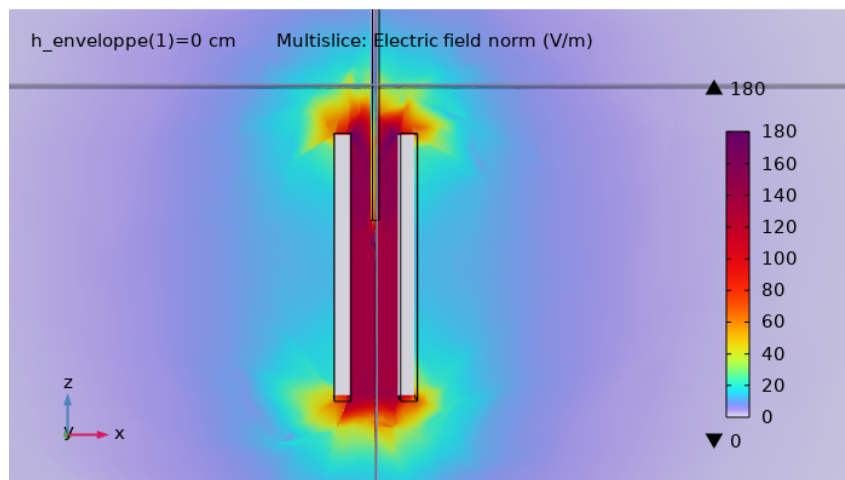


FIGURE 13 – Champs électrique

La courbe montrant la variation de la valeur de la capacité selon la position de l'enveloppe a été obtenue grâce à l'étude du paramètre sweep.

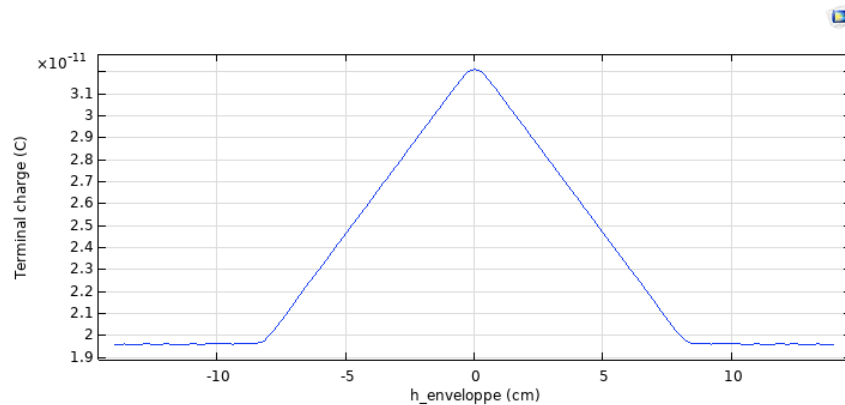


FIGURE 14 – Variation de la capacité des électrodes en fonction de la position de l'enveloppe

D'après la figure précédente, nous remarquons que la capacité maximale est atteinte lorsque le paramètre $h_{enveloppe}$ vaut 0. Cela correspond au cas où il est au milieu des deux électrodes, ce qui est tout à fait logique. En effet, la permittivité relative du papier vaut 2. Par conséquent, d'après l'équation 4, nous obtenons que la capacité va augmenter lorsque le milieu diélectrique augmente en permittivité.

5 Mise en œuvre du système capacitif

Dans cette partie, nous traiterons la réalisation du système capacitif du soudage à la programmation de ce dernier.

5.1 Description des composants du système

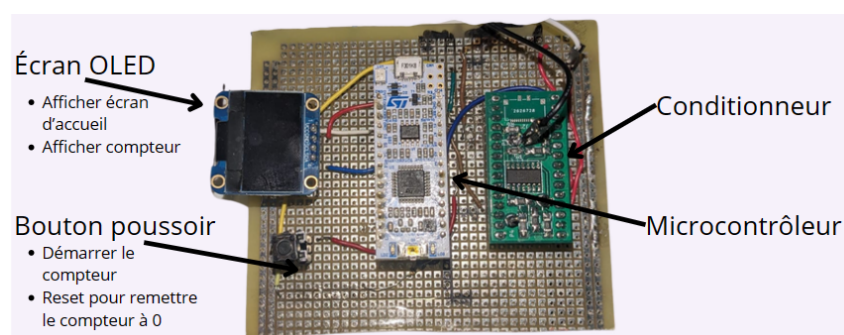


FIGURE 15 – Description du système

5.1.1 Microcontrôleur

Pour réaliser ce système, il est nécessaire d'utiliser un microcontrôleur et nous nous sommes servis du NUCLEO F301 que nous avons programmé grâce à l'environnement ARDUINO IDE de STM.

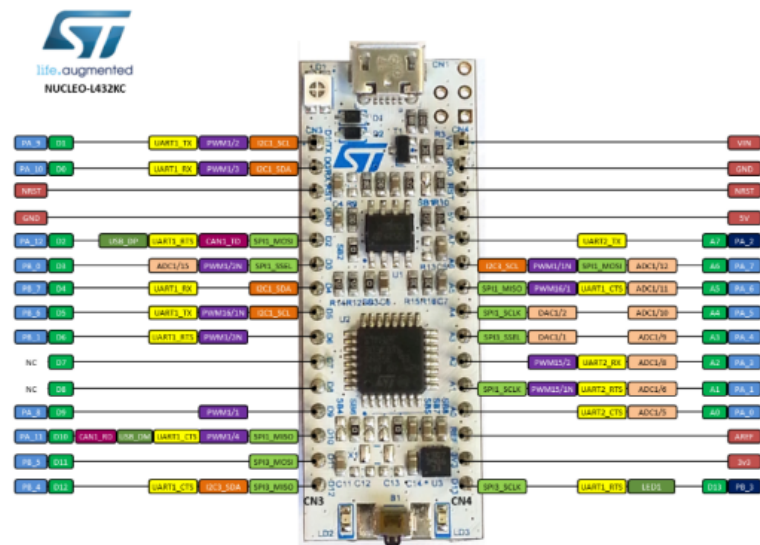


FIGURE 16 – Configuration du NUCLEO F301

5.1.2 Conditionneur

Lors des séances de TP, nous avons pu souder le circuit conditionneur sur un PCB. Puis nous avons effectué plusieurs tests pour contrôler les différents points du circuit.

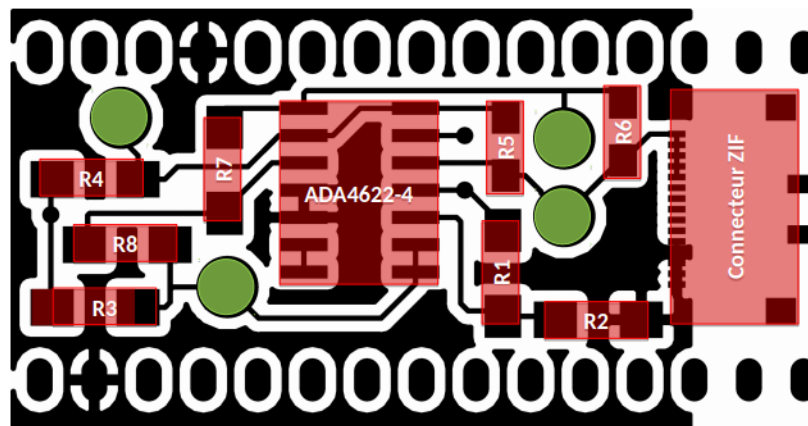


FIGURE 17 – Configuration du conditionneur à souder

5.1.3 Écran OLED

Pour faciliter la compréhension de l'interface utilisateur, nous avons décidé d'utiliser un écran OLED SSD1306. Cet écran communique grâce à un bus I2C avec le microcontrôleur.



FIGURE 18 – Écran OLED

5.1.4 Bouton poussoir

On travaillera aussi avec un bouton poussoir qui sera géré par le microcontrôleur à travers une connexion GPIO entrée.



FIGURE 19 – Bouton poussoir

5.2 Réalisation du socle accueillant les électrodes

Nous avons tout d'abord utiliser le logiciel *Blender* pour modéliser en 3D le couvercle de la boîte à vote.

Nous avons voulu personnaliser au maximum le couvercle qui comporte les 2 électrodes pour gérer au mieux la sensibilité du système capacitif et par conséquent, obtenir le décompte le plus précis et fiable dans notre cas.

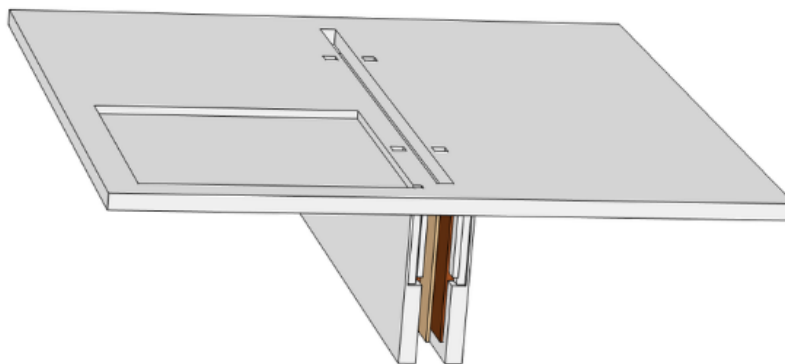


FIGURE 20 – Modélisation 3D de notre système à 2 électrodes

On peut observer que lors de la modélisation 3D, nous avons inséré des cavités au sein de chaque paroi pour permettre l'insertion des électrodes de manière simple et rapide.

Cependant, le papier cartonné étant difficilement détectable, il ne permet pas une variation de la capacité hautement significative. Par conséquent, avant d'imprimer le couvercle et pour optimiser au maximum la sensibilité du capteur, nous avons utilisé le logiciel *Comsol* comme décrit précédemment.

5.3 Programmation du microcontrôleur

On peut représenter l'algorithme sous la forme de l'algorithme suivant :

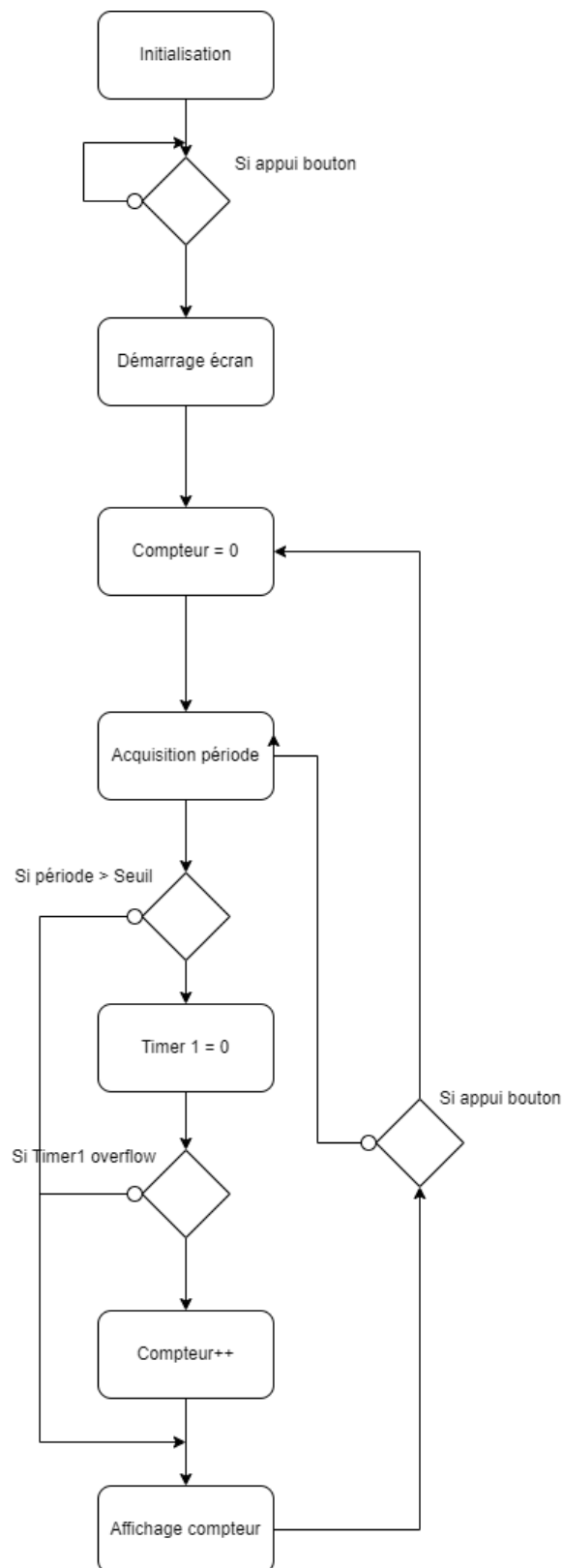


FIGURE 21 – Algorithme du main

Afin de comptabiliser le nombre de votes, nous allons tout d'abord observer le changement de variation de la période (et donc de la capacité) puis nous allons analyser si cette variation a lieu sur plus d'une seconde et demi (on pose la période du Timer1 comme

étant égale à une seconde et demi), alors on comptabilise un vote.

Après plus calibrages de manière empirique, une seconde et demi nous a paru comme étant la période la plus proche de la réalité.

Par ailleurs, le bouton poussoir agit comme un bouton On/Off au début du programme puis il se comporte comme un bouton reset qui permet de ré-initialiser le compteur à 0.

6 Améliorations possibles

Plusieurs améliorations sont possibles au niveau de la modélisation du couvercle et des électrodes.

En effet, les électrodes étant peu sensibles aux bulletins cartonnés de vote, il faudrait augmenter la variation de capacité lors de l'insertion d'une enveloppe dans la boîte. Pour ce faire, on a plusieurs solutions : on peut agrandir la surface des électrodes, réduire la distance entre les deux électrodes ou encore créer une morphologie des électrodes non-linéaire.

7 Conclusion

Pour conclure, ce projet a été pertinent car il nous a permis d'appréhender des logiciels tels que *Partquest Explorer*, *Comsol* ou encore l'environnement *Arduino IDE*. De plus, nous avons pu réaliser un projet de l'idée de conception à sa réalisation en utilisant nos compétences en informatique et en électronique acquises au sein du cursus ISMIN, ce qui rend concret notre apprentissage en école d'ingénieurs.

8 Annexes

```

while(button_state == false)
{
    if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_4) == GPIO_PIN_RESET) {
        button_state = true;
    }
}
button_state = false;
ssd1306_DrawLogo(); // Affichage démarrage du compteur

/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */

while (1)
{
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    while(button_state == false)
    {
        HAL_TIM_IC_Start_IT(&htim2, TIM_CHANNEL_1);
        sprintf(str, "%d", cpt); //affichage du compteur permanent
        ssd1306_SetCursor(50, 25);
        ssd1306_WriteString(str, Font_16x26, White);
        ssd1306_UpdateScreen();

        if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_4) == GPIO_PIN_RESET) {
            button_state = true;
        }
    }
    button_state = false;
    cpt = 0;
    ssd1306_Init();
}
/* USER CODE END 3 */
}

```

FIGURE 22 – Code - Main

```

void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef* htim2)
{
    if (htim2->Instance == TIM2)
    {
        if (first == 1)
        {
            tab[0] = HAL_TIM_ReadCapturedValue(htim2, TIM_CHANNEL_1);
            first = 0;
        }
        else
        {
            tab[1] = HAL_TIM_ReadCapturedValue(htim2, TIM_CHANNEL_1);
            value = tab[1] - tab[0];
            frequency = HAL_RCC_GetPCLK1Freq()/value;
            first = 1;
            HAL_TIM_IC_Stop_IT(htim2, TIM_CHANNEL_1);
        }
        if (value>1100)
        {
            if (tim1_started == 0)
            {
                HAL_TIM_SET_COUNTER(&htim1, 0);
                HAL_TIM_Base_Start_IT(&htim1);
                tim1_started = 1;
            }
        }
        else if (value<1100)
        {
            HAL_TIM_Base_Stop_IT(&htim1);
            tim1_started = 0;
        }
    }
}

```

FIGURE 23 – Code - Interruption du Timer2

```

void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
    if (htim->Instance == TIM1)
    {
        HAL_GPIO_WritePin(GPIOB, LD3_Pin , led_state);
        led_state = (led_state == 0) ? 1 : 0;
        {
            cpt++;
        }
    }
}

```

FIGURE 24 – Code - Interruption du Timer1

```

void ssd1306_DrawLogo()
{
    ssd1306_Init();
    ssd1306_Fill(Black);
    ssd1306_SetCursor(0, 0);
    ssd1306_WriteString(".....DEMARRAGE.....", Font_7x10, White);
    ssd1306_DrawBitmap(32,16,logo_vote,64,64,White);
    ssd1306_UpdateScreen();
    HAL_Delay(4000);
    ssd1306_Fill(Black);
    ssd1306_UpdateScreen();
}

```

FIGURE 25 – Code - Initialisation démarrage de l'écran OLED