

## Prototypage

---

# Rapport Projet Prototypage

Développement d'un jeu Mario Bros piloté par capteur capacitif

---

Yasmine SALMOUNI

Yassmina BARA

**Juin 2024**

## Table des matières

<b>1</b>	<b>Entrée en contexte d'un système sensitif avec interface capacitive</b>	<b>2</b>
1.1	Fonctionnement et principes des capteurs capacitifs . . . . .	2
1.2	Fonctionnement et principes des systèmes sensitifs à base de capteurs capacitifs	2
<b>2</b>	<b>Principe du circuit conditionneur</b>	<b>3</b>
2.1	Principe de fonctionnement . . . . .	3
2.2	Convertisseur tension-courant . . . . .	4
2.3	Le comparateur à hystérésis . . . . .	6
2.4	Association du convertisseur tension-courant et du comparateur à hystérésis .	7
2.5	Conception de l'oscillateur : fonctionnement en boucle fermée . . . . .	7
2.6	Circuit global en alimentation unique . . . . .	8
<b>3</b>	<b>conception du circuit conditionneur</b>	<b>9</b>
<b>4</b>	<b>Notre projet : développement d'un jeu Mario Bros piloté par capteur capacitif</b>	<b>10</b>
4.1	Présentation du prototype : fonctionnement et fonctions remplies par le démonstrateur . . . . .	10
4.2	Programmation du microcontrôleur STM32 . . . . .	11
4.2.1	Configuration IOC . . . . .	11
4.2.2	Développement du fichier main.c . . . . .	14
	<b>Liste des figures</b>	<b>16</b>

# 1 Entrée en contexte d'un système sensitif avec interface capacitive

## 1.1 Fonctionnement et principes des capteurs capacitifs

Un capteur capacitif est un dispositif électronique qui utilise la capacité électrique pour détecter divers types de changements dans son environnement. Le capteur convertit cette variation de capacité en un signal électrique, qui est ensuite analysé pour déterminer la nature de l'interaction.

Imaginons que je déplace mon doigt autour d'un capteur capacitif. Ce capteur peut être considéré comme une des plaques d'un condensateur, tandis que mon doigt agirait comme la deuxième plaque. Lorsque je bouge mon doigt, la distance entre mon doigt et le capteur varie, modifiant ainsi la capacité électrique du capteur. Cette variation est ce qui permet au capteur de détecter et de répondre aux mouvements du doigt. En effet, la formule d'un condensateur plan montre que la capacité est inversement proportionnelle à la distance entre les plaques :

$$C = \epsilon_r \epsilon_0 \frac{A}{d}$$

où :

- $C$  est la capacité,
- $\epsilon_r$  est la permittivité relative du matériau entre les plaques (dans ce cas, l'air et potentiellement une partie de votre doigt),
- $\epsilon_0$  est la permittivité du vide,
- $A$  est la surface effective des plaques qui se font face,
- $d$  est la distance entre votre doigt et la surface du capteur.

Notons que dans le cas d'un capteur capacitif et dans le cas spécifique du doigt en mouvement, la grandeur  $A$  — qui représente la surface effective d'interaction entre le capteur et le doigt — peut également varier si le mouvement du doigt modifie cette zone d'interaction.

## 1.2 Fonctionnement et principes des systèmes sensitifs à base de capteurs capacitifs

Un système sensitif, également appelé système sensoriel ou système de détection, est un dispositif ou un ensemble de dispositifs capables de détecter des changements ou des stimuli dans leur environnement et de répondre en conséquence. Ces systèmes sont conçus pour percevoir des informations physiques ou chimiques, comme la lumière, la température, la pression, ou encore la composition chimique. Ils convertissent ces informations en signaux électriques ou autres formes de données interprétables, permettant ainsi une interaction adéquate.

Un système sensitif avec interface capacitive intègre un capteur capacitif capable de détecter les variations dans son environnement. Ce capteur convertit les fluctuations de capacité électrique en signaux électriques, permettant ainsi au système de répondre de manière appropriée aux changements détectés.

## 2 Principe du circuit conditionneur

### 2.1 Principe de fonctionnement

La fonction principale d'un circuit conditionneur est de préparer et de modifier les signaux en sortie des capteurs capacitifs afin qu'ils soient compatibles et exploitables par des dispositifs de traitement ultérieurs. Le circuit conditionneur effectue plusieurs tâches : il amplifie les signaux faibles, filtre les bruits indésirables et, si nécessaire, convertit les signaux analogiques en formats numériques. Dans le cas de notre projet, les signaux en sortie du circuit conditionneur sont traités par un microcontrôleur : la carte Nucleo F301K8T6. Sur cette carte, nous avons programmé le microcontrôleur pour effectuer une conversion analogique-numérique (ADC) du signal et calculer la fréquence correspondante à partir des données reçues.

Dans le cadre de notre projet, nous avons utilisé une électrode agissons comme un capteur capacitif. Le signal en sortie de l'électrode est conditionné par le circuit conditionneur avant d'être envoyé à la carte Nucleo F301K8T6.

Voici les étapes simplifiées de ce conditionnement :

- Le circuit commence par détecter les changements dans la tension qui proviennent de l'électrode. Ce sont ces changements qui indiquent une interaction ou une modification de l'environnement proche de l'électrode.
- Après avoir détecté ces variations de tension, le circuit les transforme en courant à l'aide d'un convertisseur tension-courant (Figure 1). Cette transformation est nécessaire pour que le signal puisse être traité plus facilement dans les étapes suivantes.
- Le signal peut être traité par le comparateur à hystérésis de la manière suivante : le comparateur compare la tension générée (issue de la conversion du courant) avec deux seuils de tension définis (haut et bas). Lorsque la tension dépasse le seuil haut, la sortie du comparateur bascule à un état haut. Lorsqu'elle redescend en dessous du seuil bas, la sortie bascule à un état bas. Grâce à l'hystérésis, le comparateur évite les réactions excessives aux petites variations, stabilisant ainsi la réponse du circuit.
- La sortie du comparateur est renvoyée à l'entrée du convertisseur tension-courant. Cela signifie que le signal de sortie du comparateur influence directement le signal de tension initial. Par exemple, lorsque la sortie du comparateur est haute, elle pourrait réduire le signal de tension, et lorsque la sortie est basse, elle pourrait l'augmenter. Ce processus se répète continuellement. Le signal de sortie du comparateur ajuste le signal d'entrée du convertisseur tension-courant, ce qui modifie à nouveau le signal de courant envoyé au comparateur. Cette interaction continue crée des oscillations stables.

## 2.2 Convertisseur tension-courant

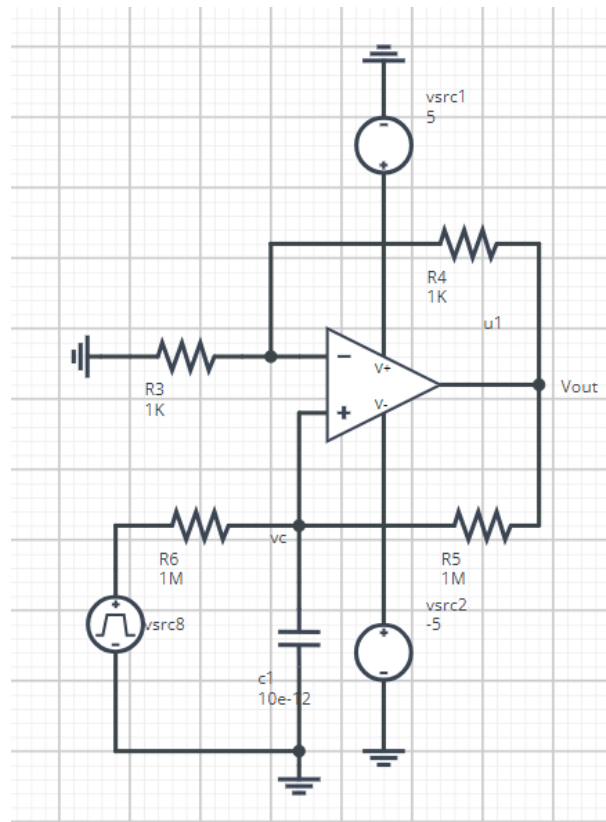


FIGURE 1 – Circuit électrique d'un convertisseur tension-courant

Le circuit est conçu à l'aide du logiciel Partquest, avec les valeurs suivantes :

- $R_3 = R_4 = 1 \text{ k}\Omega$
- $R_5 = R_6 = 1 \text{ M}\Omega$
- Alimentation de l'AOP :  $\pm 5 \text{ V}$
- grandeurs internes de l'amplificateur opérationnel :
  - GBW (Gain Bandwidth Product) :  $1 \times 10^8 \text{ Hz}$
  - SR\_MAX (Slew Rate Maximal) :  $1 \times 10^8 \text{ V/s}$
  - $R_{in} = 1 \times 10^{10} \Omega$
  - $V_{O\_PP} = 6 \text{ V}$  (Amplitude de sortie crête-à-crête)

Exprimons le courant de sortie du convertisseur tension-courant en fonction de la tension d'entrée.

Supposons que l'amplificateur est idéal

$$V^+ = V^-$$

$$V^- = V_{\text{out}} \left( \frac{R_3}{R_3 + R_4} \right) \quad (\text{Pont diviseur de tension})$$

Notons  $V_{in}$  La tension en entrée du convertisseur tension-courant.

$$V^+ = \left( \frac{V_{in}}{R_6} + \frac{V_{out}}{R_5} \right) \left( \frac{1}{R_6} + \frac{1}{R_5} + \frac{1}{Z_C} \right) \quad (\text{Millmann})$$

En remplaçant  $V_{out}$  dans  $V^+$  on obtient :

$$V^+ = \frac{V_{in}}{R_6 \left( \frac{1}{R_6} + \frac{1}{R_5} + \frac{1}{Z_C} \right)} + \frac{R_6 \left( \frac{R_4+R_3}{R_5 R_3} \right) V^+}{R_6 \left( \frac{1}{R_6} + \frac{1}{R_5} + \frac{1}{Z_C} \right)}$$

D'où

$$V^+ = \frac{V_{in}}{1 + R_6 \left( \frac{1}{R_5} + \frac{1}{Z_C} - \frac{R_4+R_3}{R_5 R_3} \right)}$$

On note  $i_{cap}$  l'intensité du courant qui traverse le condensateur du circuit convertisseur analogique

$$i_{cap} = \frac{V^+}{Z_C}$$

donc

$$i_{cap} = \frac{V_{in}}{Z_C + R_6 \left( \frac{Z_C}{R_5} + 1 + \frac{Z_C(R_4+R_3)}{R_5 R_3} \right)}$$

La condition pour que  $i_{cap}$  soit indépendante de  $Z_C$  est la suivante :  $R_4 R_6 = R_3 R_5$

Dans ce cas là,

$$i_{cap} = \frac{V_{in}}{R_6}$$

La figure suivante représente la tension appliquée en entrée du convertisseur tension-courant  $V_{in}$  :

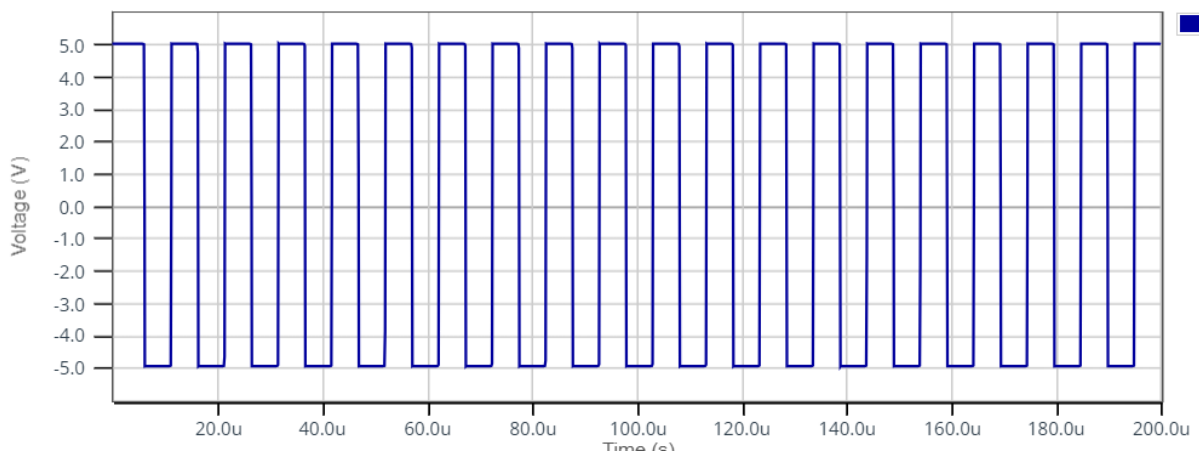


FIGURE 2 – Représentation de  $V_{in}$  en fonction du temps

## 2.3 Le comparateur à hystérésis

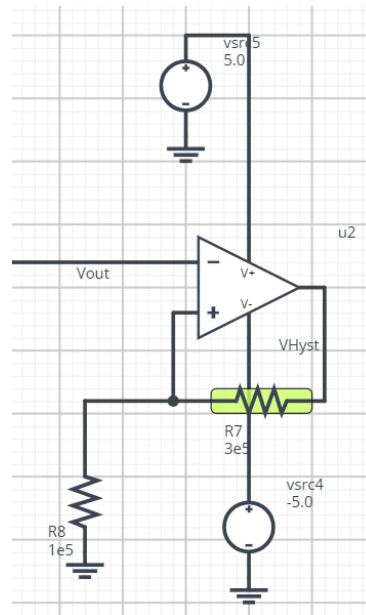


FIGURE 3 – Circuit électrique d'un comparateur à hystérésis

Pour réaliser ce circuit, les valeurs suivantes ont été utilisées :

- Les deux résistances ont une valeur de  $10\text{ k}\Omega$ .
- L'alimentation est de  $\pm 5\text{ V}$ .
- La tension d'entrée a une amplitude de  $5\text{ V}$ .

En simulant le circuit, on obtient les résultats de la figure suivante :

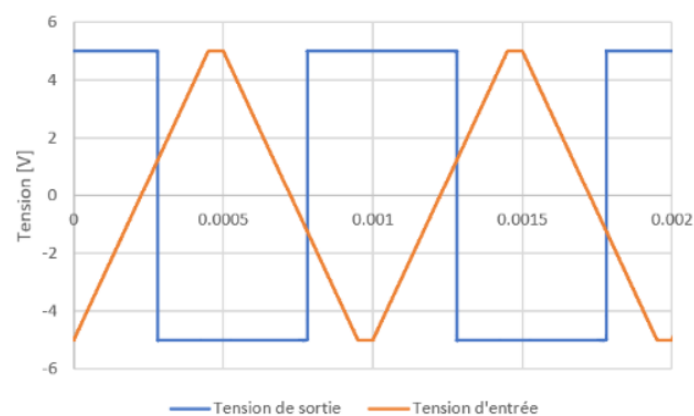


FIGURE 4 – Tensions de sortie et d'entrée du comparateur à hystérésis

Nous retrouvons le fonctionnement stabilisateur du comparateur à hystérésis expliqué dans le principe de fonctionnement du circuit conditionneur.

## 2.4 Association du convertisseur tension-courant et du comparateur à hystérésis

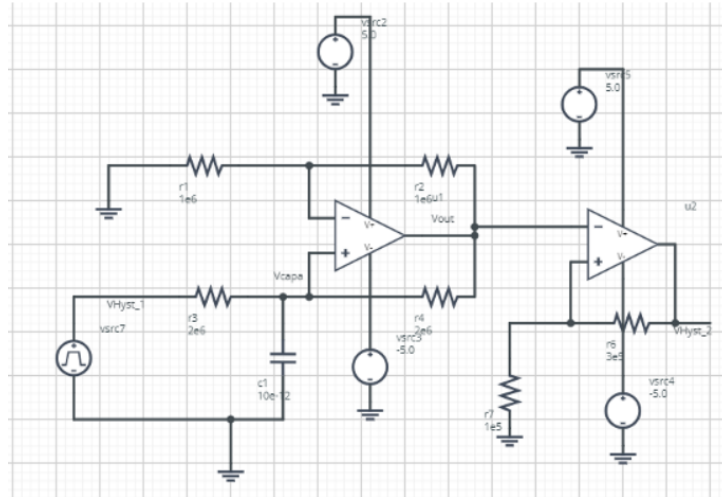


FIGURE 5 – Circuit électrique de l'association du convertisseur tension-courant et du comparateur à hystérésis

Nous obtenons les résultats de la figure suivante :

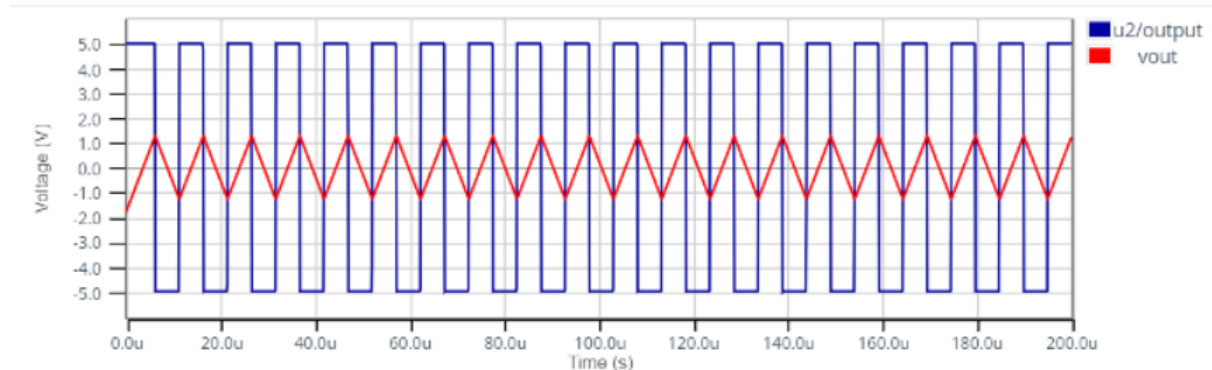


FIGURE 6 – Tensions de sortie du convertisseur tension-courant et du comparateur à hystérésis

Nous constatons que le signal en sortie du comparateur à hystérésis bascule à -5V quand le signal en sortie du convertisseur tension-courant vaut 1.25, et bascule vers 5V quand ce dernier vaut -1.25.

## 2.5 Conception de l'oscillateur : fonctionnement en boucle fermée

L'oscillateur est obtenu en reliant la sortie du comparateur à hystérésis à l'entrée du convertisseur tension-courant.

la figure suivante représente le fonctionnement en boucle fermée de l'oscillateur :



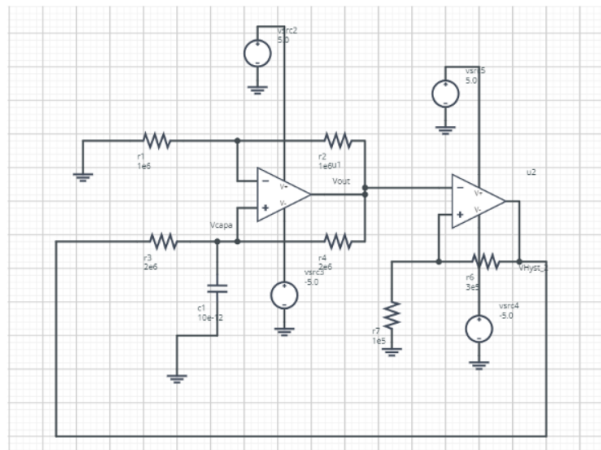


FIGURE 7 – Circuits électrique de l'oscillateur en boucle fermée

Remarque : nous obtenons les mêmes résultats de la figure 6, ceci s'explique par le fait que le tension en sortie du comparateur à hystérésis est identique à celui en entrée du convertisseur (signal carré  $\pm 5V$ ).

## 2.6 Circuit global en alimentation unique

La figure suivante représente le circuit global ou nous avons regroupé toutes les alimentations en une seule :

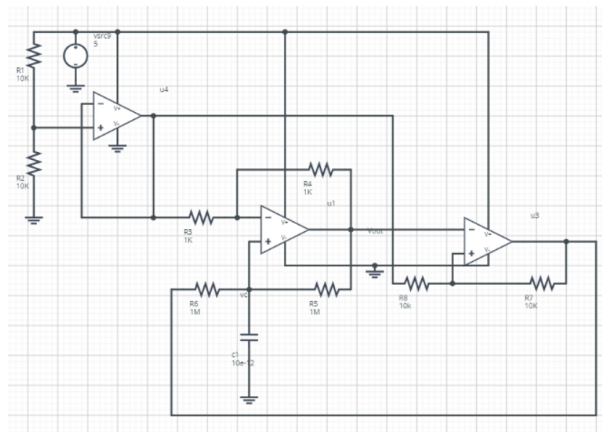


FIGURE 8 – Circuits électrique global

On note bien que les deux résistances  $R_1$  et  $R_2$  constituent un diviseur de tension ; la tension à l'entrée non inverseuse de l'amplificateur à gauche vaut  $\frac{5}{2} = 2.5V$ .

### 3 conception du circuit conditionneur

Nous avons utilisé le logiciel KiCad pour tracer le circuit conditionneur disponible sur ecampus. KiCad est un outil open source de conception assistée par ordinateur (CAO) pour la création de schémas électroniques et de circuits imprimés (PCB).

Les étapes suivies pour la réalisation sont les suivantes :

1. Nous avons d'abord dessiné le circuit conditionneur en utilisant les composants nécessaires, tels que les résistances, les condensateurs et les amplificateurs opérationnels (Figure 9).
2. Chaque composant du schéma a été associé à une empreinte correspondant à son boîtier physique pour le placement sur le PCB.
3. À partir du schéma électrique, nous avons généré le layout du PCB en plaçant les composants sur la carte et en traçant les pistes pour connecter les composants selon le schéma (Figure 10).
4. Nous avons vérifié le PCB pour nous assurer qu'il respecte les règles de conception et qu'il n'y a pas d'erreurs de routage.

Une fois le circuit tracé et le PCB généré, nous avons exporté les fichiers Gerber nécessaires à la fabrication du circuit imprimé.

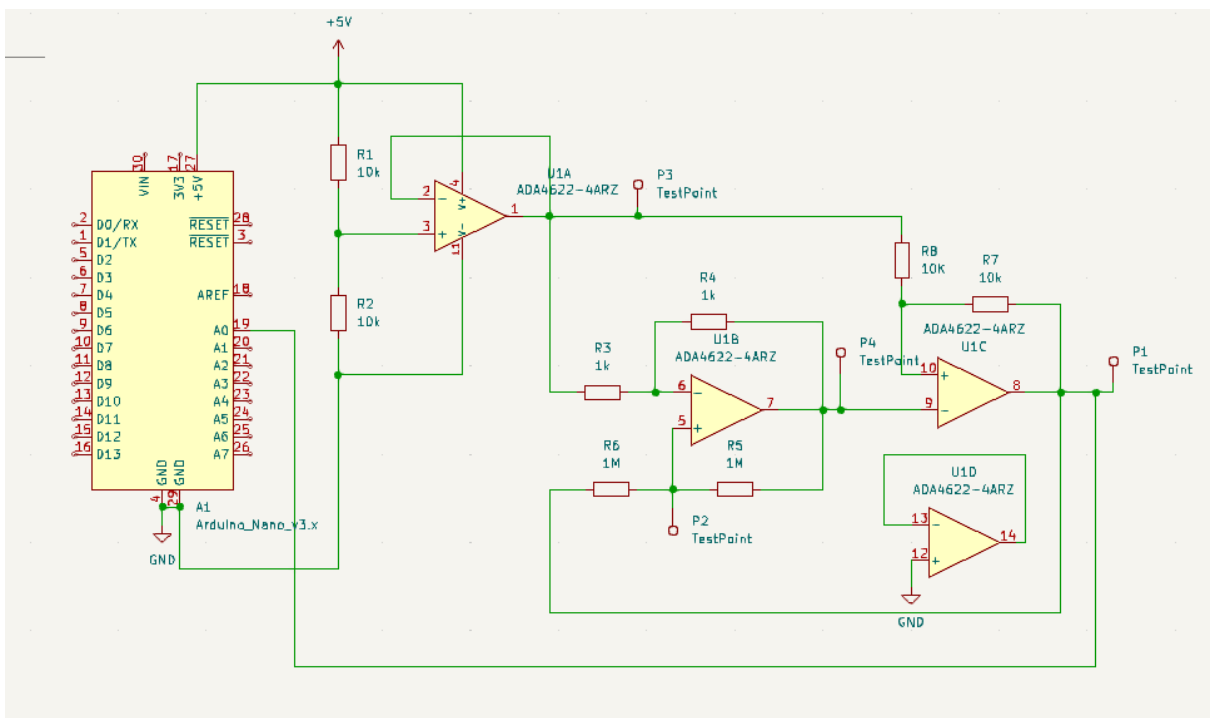


FIGURE 9 – circuit conditionneur dessiné sur kicad

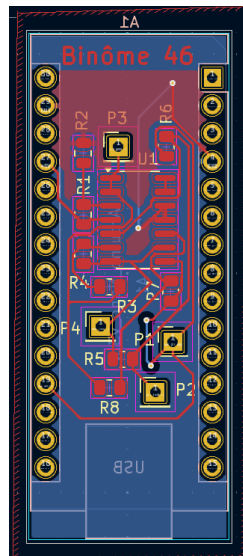


FIGURE 10 – PCB généré sur kicad

## 4 Notre projet : développement d'un jeu Mario Bros piloté par capteur capacitif



FIGURE 11 – Mario, prêt pour pour partir à l'aventure

### 4.1 Présentation du prototype : fonctionnement et fonctions remplies par le démonstrateur

Dans le cadre de notre projet de prototypage, nous avons développé un jeu interactif inspiré de Mario Bros, piloté par un e électrode capacitive. Ce capteur est intégré au circuit conditionneur conçu sur kicad, lui-même connecté à la carte Nucleo F301K8T6. Le fonctionnement repose sur les interactions physiques : les mouvements de la main à proximité du capteur modifient sa capacité, ce qui se traduit par des variations de tension. Ces variations sont converties en courant par un convertisseur tension-courant.

Sur la carte Nucleo, le microcontrôleur a été programmé pour convertir le signal analogique en numérique (ADC). Il analyse ensuite les données pour calculer la fréquence d'oscillation

du signal reçu. Cette fréquence est utilisée pour contrôler les déplacements de Mario dans le jeu : une augmentation de la fréquence provoque un déplacement à droite de Mario, simulant une main se rapprochant du capteur, tandis qu'une diminution entraîne un mouvement vers la gauche, correspondant à un éloignement de la main.

Le jeu a été intégralement codé en Python, ce qui permet une grande flexibilité dans la programmation des interactions et des commandes du jeu.

## 4.2 Programmation du microcontrôleur STM32

### 4.2.1 Configuration IOC

La configuration IOC du projet STM32 est représentée dans la figure suivante :

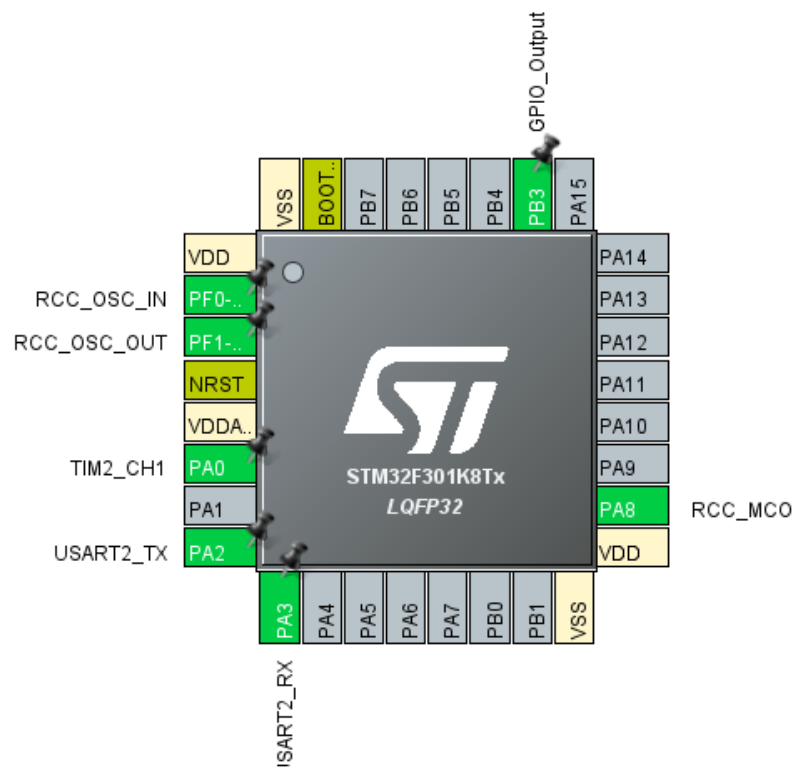


FIGURE 12 – Configuration IOC du projet STM32

Nous avons configuré principalement les PINs d'horloge, les deux ports de transmission et réception de l'UART2, et le port PA0 en mode TIMER2.

La configuration de l'UART est la suivante :

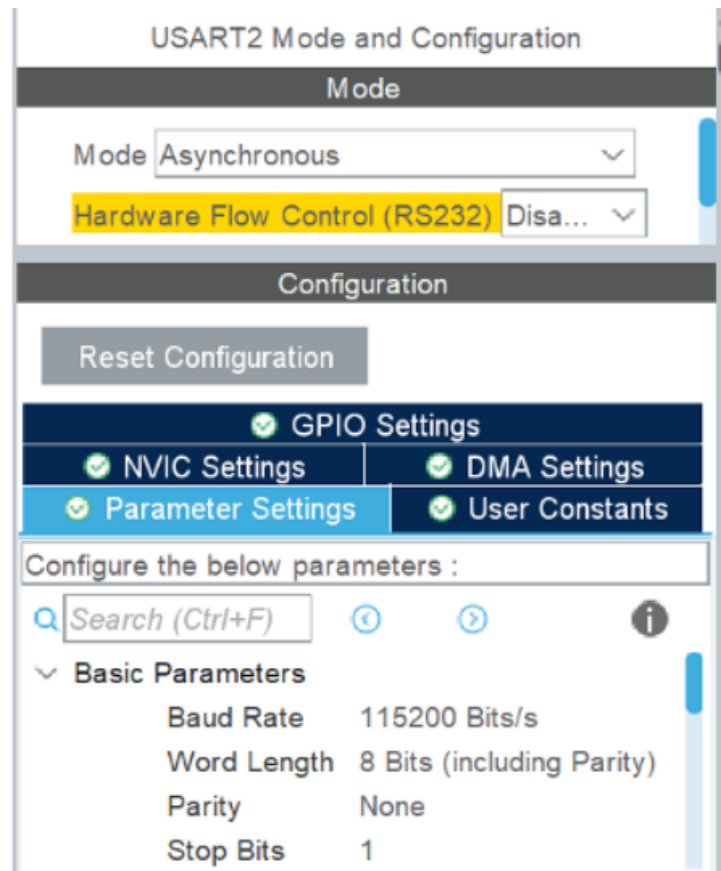


FIGURE 13 – Configuration de l'UART du projet STM32

Nous avons configuré l'UART en mode asynchrone avec une vitesse de transmission (Baud Rate) valant 115200 Bits/s.

La configuration du timer2 est la suivante :

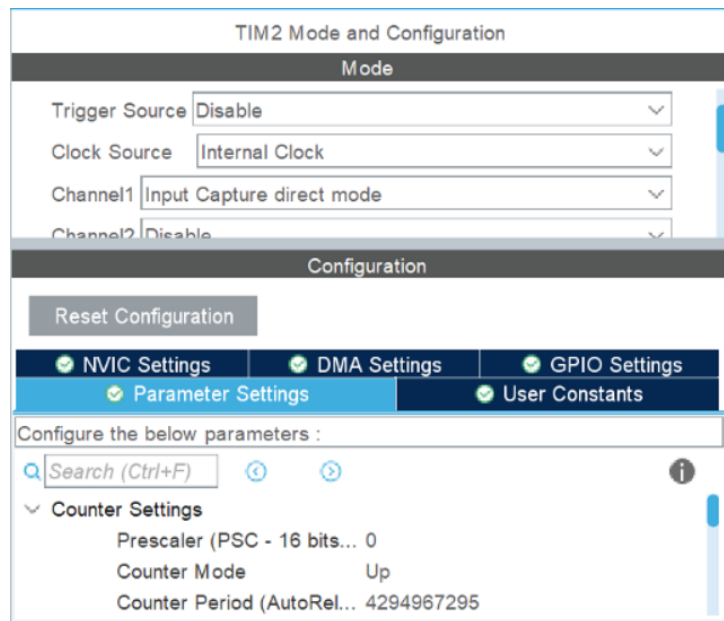


FIGURE 14 – Configuration du timer2 du projet STM32

Le port PA0 est lié au Channel 1. Nous avons donc activé ce Channel. Nous avons également activé la source d'horloge.

La figure suivante montre la configuration du timer 2 en mode interruption :

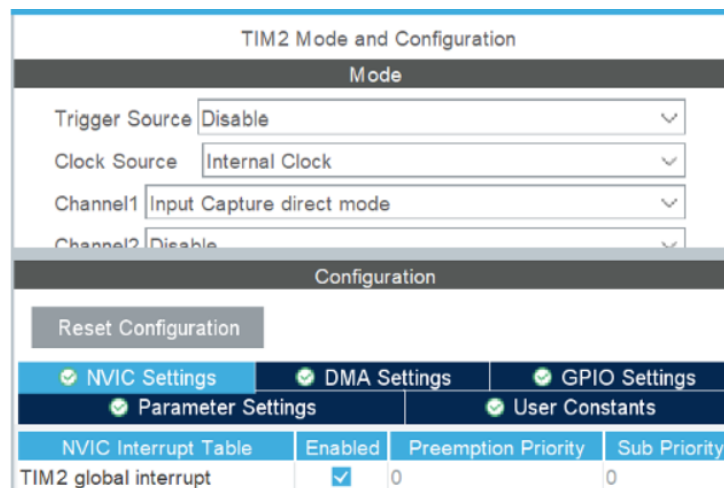


FIGURE 15 – Configuration du timer 2 en mode interruption

### 4.2.2 Développement du fichier main.c

La figure suivante facilite la compréhension de la logique adoptée pour le calcul de la fréquence :

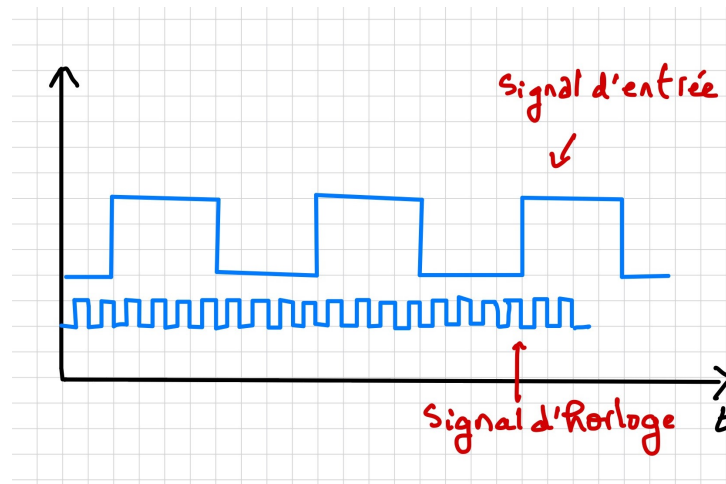


FIGURE 16 – logique adoptée pour le calcul de la fréquence

Pour calculer la fréquence, il suffit de calculer la période du signal d'entrée. Pour ce faire, nous allons calculer le nombre de coups d'horloge entre deux fronts montants du signal d'entrée, et donc on va utiliser une fonction callback représentée dans la figure suivante :

```

283 /* USER CODE BEGIN 4 */
284 void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim) {
285     if(htim->Instance==TIM2) {
286         if(htim->Channel==HAL_TIM_ACTIVE_CHANNEL_1) {
287             cpt=1;
288             courant_temps_capture=HAL_TIM_ReadCapturedValue(htim, TIM_CHANNEL_1);
289             duree=courant_temps_capture-dernier_temps_capture;
290             dernier_temps_capture=courant_temps_capture;
291             frequency=32000000/duree;
292         }
293     }
294     /*if(cpt){
295         sprintf(msg, "demi-periode: %ld\r\n", duree);
296         HAL_UART_Transmit(&huart2, (uint8_t *) msg, sizeof(msg), HAL_MAX_DELAY);
297     }*/
298 }
299 /* USER CODE END 4 */

```

FIGURE 17 – Code de la fonction callback utilisée pour le calcul de la fréquence

Il s'agit d'une fonction de rappel activée lorsqu'une interruption de capture d'entrée se produit. Cette interruption est déclenchée par le timer 2 lorsqu'il détecte une valeur d'entrée spécifique, typiquement une transition sur un signal d'entrée. Dans notre cas, cette valeur d'entrée correspond à une variation de capacité au niveau de l'électrode.

Dans le corps de la fonction, nous vérifions initialement si l'interruption a été déclenchée par le timer 2. Bien que cette vérification ne soit pas strictement nécessaire — puisqu'un seul port est configuré en mode timer — elle est maintenue pour assurer une cohérence dans le code. Nous utilisons ensuite la fonction HAL\_TIM\_ReadCapturedValue pour mesurer l'instant du premier front montant du signal. La valeur capturée est stockée dans la variable

courant\_temps\_capture. Par la suite, nous calculons la durée entre ce temps de capture courant et le précédent, stocké dans la variable dernier\_temps\_capture — cette dernière ayant été initialisée à 0 dans l'entête du code. Après le calcul, dernier\_temps\_capture est mise à jour avec la valeur courante obtenue par HAL\_TIM\_ReadCapturedValue. La variable duree contient le nombre de coups d'horloge entre deux fronts montants du signal d'entrée. Pour calculer la période du signal, nous multiplions simplement duree par la période d'horloge du système, qui est de 32 MHz. La fréquence du signal est ensuite déduite en prenant l'inverse de la période calculée.

Une fois la fréquence d'oscillation obtenue, nous pouvons la transmettre au code Python qui sera développé ultérieurement. Cette transmission s'effectuera via l'UART2, comme illustré dans la figure ci-dessous.

```
108  /* Infinite loop */
109  /* USER CODE BEGIN WHILE */
110  while (1)
111  {
112      if(cpt){
113          sprintf(msg, "%ld\r\n", frequency);
114          HAL_UART_Transmit(&huart2, (uint8_t *) msg, sizeof(msg), HAL_MAX_DELAY);
115      }
116  /* USER CODE END WHILE */
117
```

FIGURE 18 – Code de transmission de la fréquence via l'UART2



## Table des figures

1	Circuit électrique d'un convertisseur tension-courant . . . . .	4
2	Représentation de $V_{in}$ en fonction du temps . . . . .	5
3	Circuit électrique d'un comparateur à hystérésis . . . . .	6
4	Tensions de sortie et d'entrée du comparateur à hystérésis . . . . .	6
5	Circuit électrique de l'association du convertisseur tension-courant et du comparateur à hystérésis . . . . .	7
6	Tensions de sortie du convertisseur tension-courant et du comparateur à hystérésis . . . . .	7
7	Circuits électrique de l'oscillateur en boucle fermée . . . . .	8
8	Circuits électrique global . . . . .	8
9	circuit conditionneur dessiné sur kicad . . . . .	9
10	PCB généré sur kicad . . . . .	10
11	Mario, prêt pour pour partir à l'aventure . . . . .	10
12	Configuration IOC du projet STM32 . . . . .	11
13	Configuration de l'UART du projet STM32 . . . . .	12
14	Configuration du timer2 du projet STM32 . . . . .	13
15	Configuration du timer 2 en mode interruption . . . . .	13
16	logique adoptée pour le calcul de la fréquence . . . . .	14
17	Code de la fonction callback utilisée pour le calcul de la fréquence . . . . .	14
18	Code de transmission de la fréquence via l'UART2 . . . . .	15