

ELE3312

Microcontrôleurs et applications

Projet final

Version préliminaire

Automne 2024

Sommaire

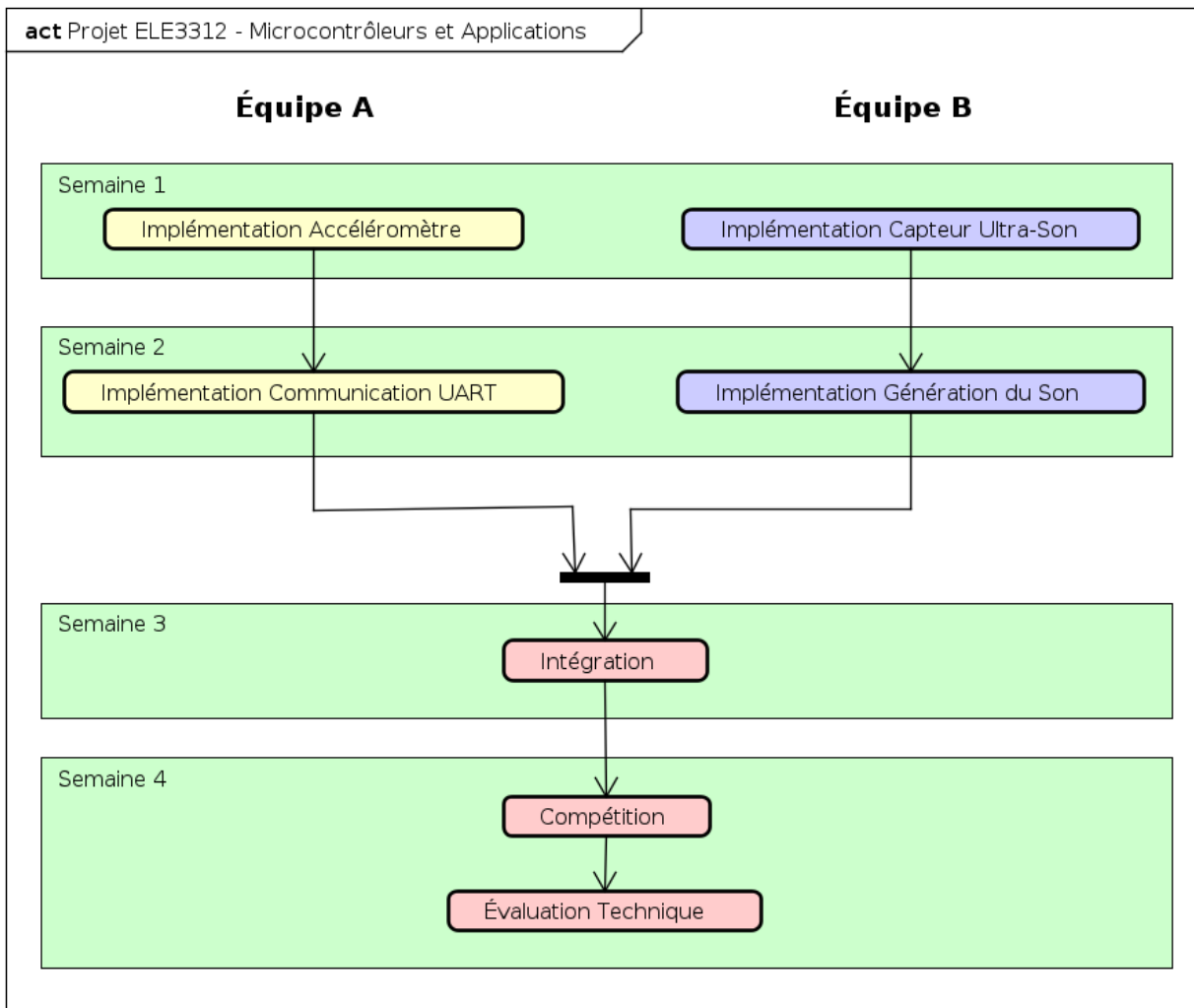
Introduction.....	3
Description de l'application.....	4
Directives particulières	5
Travail à effectuer.....	5
Semaine 1	6
Objectif A1 : Accéléromètre.....	6
Objectif B1 : Capteur de distance à ultrason	8
Semaine 2	11
Objectif A2 : Communication UART.....	11
Objectif B2 : Génération du son	12
Semaine 3 : Intégration	14
Semaine 4 : Évaluation	15
Compétition : Évaluation fonctionnelle du jeu (en classe)	15
Évaluation technique (en laboratoire)	16

Introduction

Ce projet final s'effectue en équipe de 4 membres, et couvre les dernières semaines de la session. Nous allons utiliser les connaissances acquises en cours et en laboratoire, afin de développer une application embarquée en C sur notre carte de développement NUCLEO-F446RE.

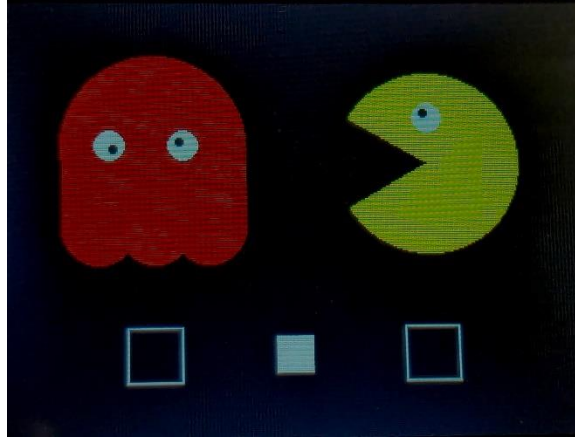
Notre objectif sera d'implémenter une version dérivée du jeu **Pac-Man** ! Un jeu d'arcade où le joueur contrôle un personnage qui se déplace dans un labyrinthe en mangeant des pièces pour gagner des points tout en évitant des fantômes.

Pendant les premières semaines du projet, l'équipe est divisée en sous-équipe de 2 étudiants (sous-équipe A et B), et chacune de ces équipes sera amenée à compléter des objectifs distincts. À partir de la troisième semaine, les équipes A et B mettront en commun leur partie respective pour compléter le jeu. À la dernière semaine, chaque équipe de 4 fera une démonstration de leur jeu lors d'une compétition en classe, et sera évaluée sur les aspects techniques en laboratoire.

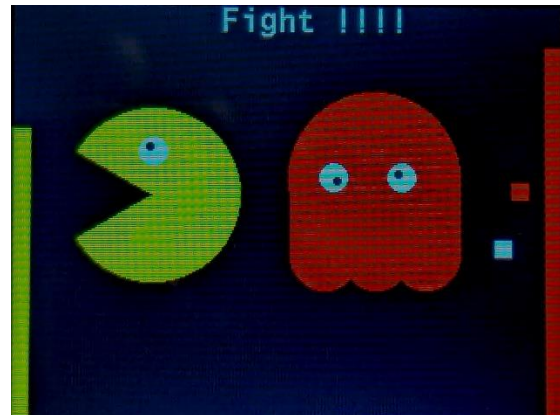
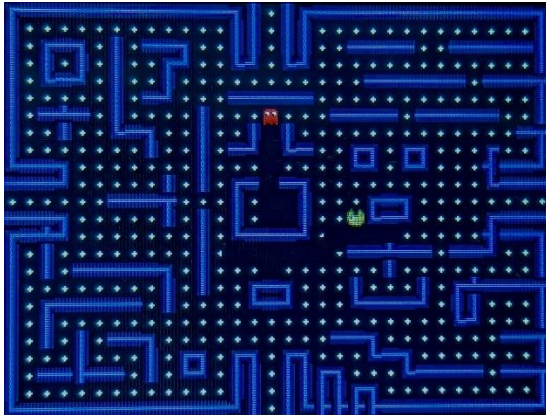


Description de l'application

Notre version de Pac-Man se joue avec 2 joueurs (donc 2 cartes) sur le même labyrinthe. Chaque joueur contrôle un personnage (Pac-Man ou fantôme) qu'il sélectionne en début de partie en inclinant l'accéléromètre dans la direction du personnage choisi à l'écran.



Après la sélection, chaque joueur se déplace sur le labyrinthe via l'accéléromètre dans le but de trouver l'adversaire.



Lorsque les joueurs se rencontrent, une phase de combat s'enclenche.

Pour gagner, il faut être le premier à remplir sa barre d'attaque au complet ! La barre d'attaque se remplit lorsque nous arrivons à aligner la distance de notre main (ou d'un objet) au capteur ultrason – représentée par le point blanc (photo haut droite) avec un point vertical généré à l'écran (point rouge sur la photo).

Lorsque le joueur arrive à s'aligner sur le point, un son spécifique est émis et le point se régénère aléatoirement à une position différente. La fréquence du son émis dépendra de la position du point sur l'axe vertical de l'écran.

Le gagnant est indiqué à l'écran, la partie se termine, et le jeu recommence automatiquement.

Directives particulières

- Les équipes sont strictement de 4 ou de 3 membres.
- Si une équipe est composée de 3 membres, le travail à effectuer sera adapté pour la sous-équipe B d'un seul membre.
- **Chaque semaine vous serez obligatoirement évalué sur la réalisation de l'objectif de la semaine passée (ex : le travail de la semaine 1 est évalué la semaine 2).**
- L'évaluation finale peut se reporter de semaine en semaine jusqu'au dernier laboratoire si au moins une personne de votre équipe se présente préparée pour faire des mesures au laboratoire.
- Toute référence à l'écran est faite en mode paysage.
- Un projet de départ avec des éléments logiciels déjà implémentés vous sera fourni, il est fortement conseiller de se baser dessus afin de monter votre jeu.

Travail à effectuer

Barème

Semaine 1*	6 pts
Semaine 2*	3 pts
Semaine 3	8 pts
Compétition	3 pts
Total	/20 pts

*L'évaluation de ces 2 parties est indépendante pour chaque sous-équipe.

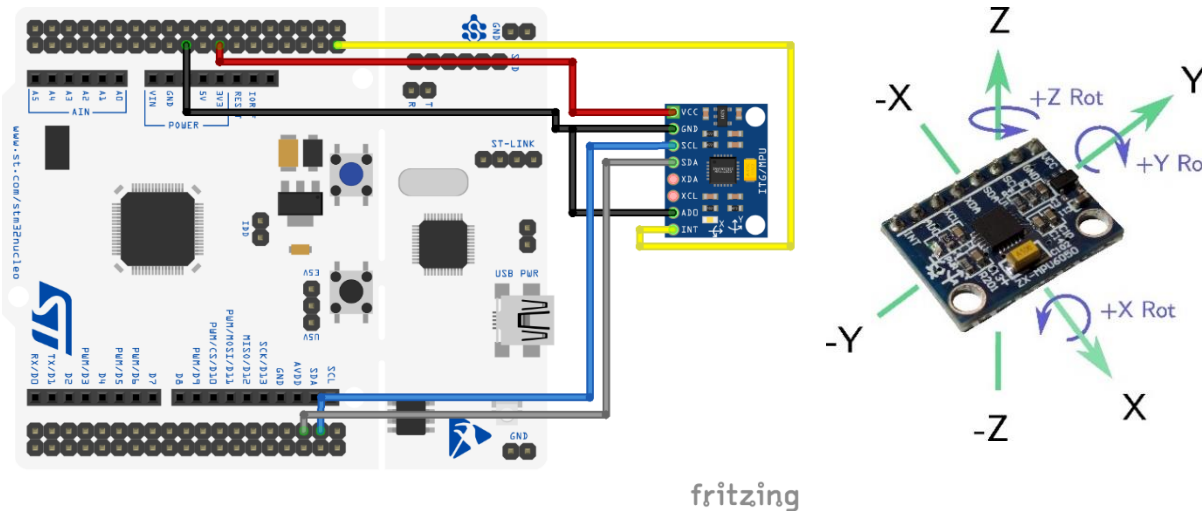
Semaine 1

Objectif A1 : Accéléromètre

Le premier objectif de l'équipe A est d'interfacer l'accéléromètre avec le microcontrôleur.

Le **MPU-6050** est un capteur de mouvement qui combine un accéléromètre et un gyroscope sur une seule puce. Il est couramment utilisé dans des applications telles que les drones, les robots et les dispositifs de réalité virtuelle pour détecter l'orientation et le mouvement.

Le capteur fonctionne sur un principe de mesure de l'accélération linéaire et de la vitesse angulaire, offrant des données précises pour le suivi de la posture et de la dynamique. Il communique généralement via une interface I2C, ce qui facilite son intégration avec des microcontrôleurs comme notre carte NUCLEO. De plus, il inclut un capteur de température, ajoutant une dimension supplémentaire à ses capacités de détection.



Pour communiquer avec le capteur, il faut d'abord trouver son adresse I2C. L'adresse est déterminée par le potentiel sur la broche ADO. Le montage proposé ci-haut met le potentiel sur cette broche à 0 V (GND). Utilisez le manuel fourni pour déterminer l'adresse I2C pour ce montage.

Les microcontrôleurs ARM Cortex M4 du fournisseur STM possèdent des périphériques spécifiés pour communiquer avec des composants I2C. Activez le périphérique I2C1 dans STM32CubeMX et choisissez le I2C Speed Mode "Fast Mode". Activez les interruptions et générez le code.

Fonctions HAL proposées pour la communication avec l'accéléromètre :

- HAL_I2C_IsDeviceReady
- HAL_I2C_Mem_Read_DMA
- HAL_I2C_Mem_Write_DMA

Dans une première étape, vérifiez que le capteur est accessible par le microcontrôleur, lisez la valeur du registre "WHO_AM_I" et comparez-la avec la valeur citée dans le manuel.

Ensuite, configurez le capteur en écrivant les valeurs du tableau suivant dans les registres correspondants.

Configuration du capteur ([description des registres du MPU-6050](#)) :

Registre	Tâche à réaliser
PWR_MGMT_1	Reset Sleep Mode, Select Internal Clock
GYRO_CONFIG	FS_SEL = ± 250 °/s, désactiver Selftest
ACCEL_CONFIG	AFS_SEL = $\pm 2g$, désactiver Selftest
CONFIG	Bandwidth = 94 Hz
INT_ENABLE	DATA_RDY_EN = 1

Le capteur est désormais prêt pour mesurer des accélérations.

Pour déclencher la lecture des nouvelles valeurs d'accélération, on va utiliser le signal d'interruption généré par le capteur sur la broche INT (que nous avons activé via le registre INT_ENABLE). Dans la configuration du microcontrôleur, mettez la broche PC11 en mode "GPIO_EXTI11" et activez dans la partie "NVIC" de la section "GPIO" : la génération des interruptions "EXTI line[15:10] interrupts".

Implémentez ensuite la routine du gestionnaire d'interruption "**void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)**" dans votre fichier de code source.

Déclenchez dans cette fonction la lecture des valeurs X et Y d'accéléromètre en mode DMA et implémentez le gestionnaire d'interruption pour la réception des données d'accéléromètre. Pour obtenir la valeur d'accélération, il faut diviser le 16 bit entier par la valeur qui correspond à la configuration du registre ACCEL_CONFIG.

Barème

Vous devez être capable de détecter le capteur, vérifier son identité, de le configurer correctement, de mesurer l'accélération dans les 3 axes, et afficher leur valeur sur l'écran en temps réel.

/1 pt	Détection de la présence du capteur
/1 pt	Vérification de l'identité du capteur (par lecture du registre "WHO_AM_I")
/2 pts	Configuration du capteur
/2 pts	Mesure d'accélération en X, Y et Z par interruption externe

Objectif B1 : Capteur de distance à ultrason

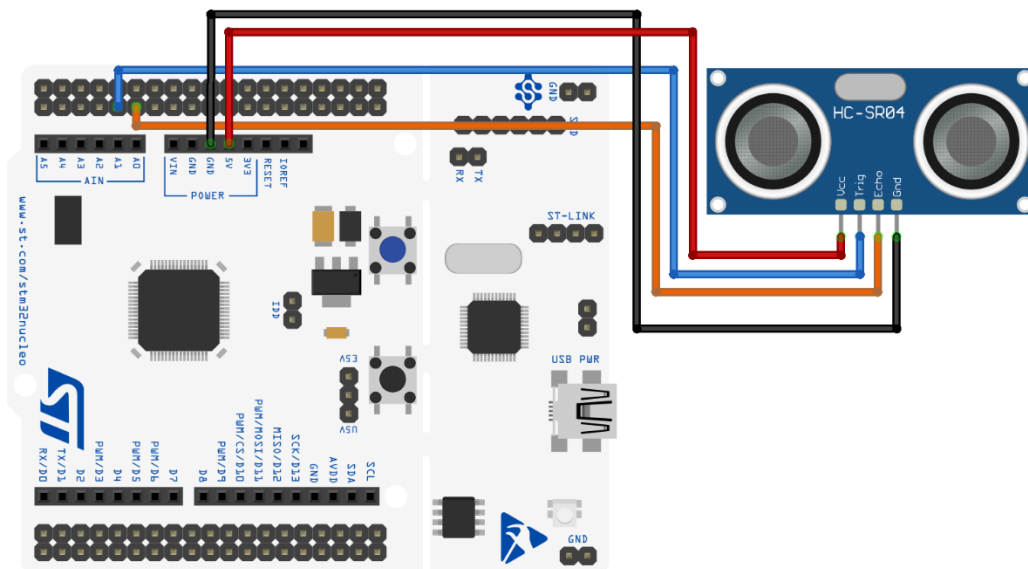
Le premier objectif de l'équipe B est d'interfacer le capteur de distance ultrason avec le microcontrôleur.

Le capteur **HC-SR04** utilise un signal sonore haute fréquence à 40kHz (inaudible à l'oreille humaine) afin d'évaluer sa distance par rapport à un objet. À la réception d'un signal de déclenchement (broche TRIG), il émet un ultrason qui sera réfléchi lorsqu'il atteint un objet. Un récepteur va alors recevoir l'onde réfléchie un certain temps après son émission initiale. Ensuite, le capteur envoie une impulsion (broche ECHO) qui va durer un temps proportionnel à la distance mesurée.

Pour déterminer la durée de l'impulsion envoyée par le capteur sur la broche ECHO, vous **devez obligatoirement** utiliser un périphérique de type Timer en mode Input Capture.

Ci-bas la description des broches et un montage suggéré avec l'utilisation du Timer 2 (par exemple) :

Broches du HC-SR04	Description	Connexion à la carte Nucleo
VCC	Broche d'alimentation.	Attention au modèle !!*
TRIG (Entrée)	Broche utilisée pour déclencher une mesure de propagation ultrason.	PA1
ECHO (Sortie)	Broche envoyant une impulsion d'une durée proportionnelle à la distance mesurée.	PA0
GND	Broche de mise à la masse.	GND



fritzing

***En principe, le modèle du capteur fourni (soit HC-SR04) est prévu pour du +5V. Si vous avez un HC-SR04P dans votre kit, il ne fonctionne qu'en +3V3V.**

Les broches PA0, PA1, PA15 et PB2 de votre microcontrôleur sont compatibles avec le 5V, mais les autres ne le sont pas nécessairement. **Soyez très prudents. Ne connectez JAMAIS une tension supérieure à 3.3V à une broche de votre microcontrôleur qui n'est pas prévue pour cela, sinon vous risquez de le détruire.**

Vérifier à partir des photos ci-dessous :



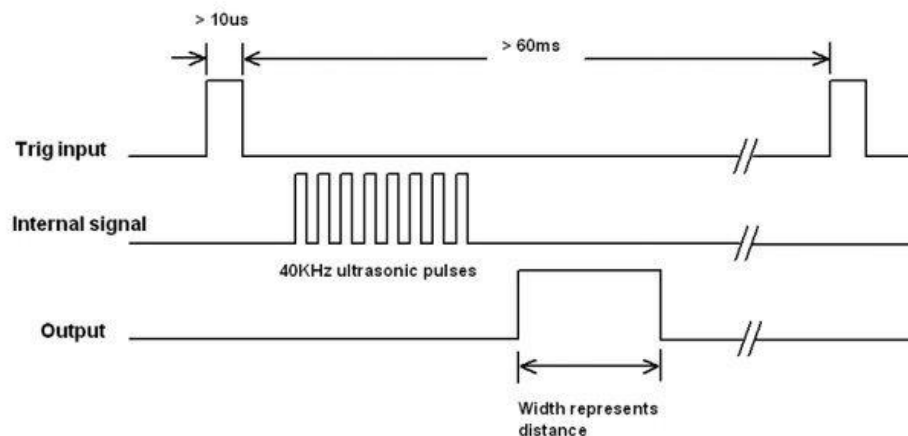
Fonctionnement spécifique

Processus de mesure de distance :

1. Émettre un signal HIGH sur la broche TRIG pendant 10µs pour déclencher la mesure.
2. Attendre le retour du signal ultrason émis par le capteur sur l'objet visé.
3. Détecter et **mesurer avec un Timer** le signal HIGH sur la broche ECHO (qui aura une durée égale au temps de propagation du signal ultrason).
4. Calculer la distance entre le capteur et l'objet :

$$\text{Distance (m)} = \frac{\text{Vitesse du son dans l'air } \left(\frac{\text{m}}{\text{s}}\right) * \text{Temps de propagation (s)}}{2}$$

Diagramme temporel des signaux lors du processus de mesure :



Détails d'utilisation du mode Input Capture direct mode sur un Timer :

- Une interruption doit être générée sur un front montant et un front descendant du signal d'entrée ;
- Une routine d'interruption `HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim)` est à implémenter ainsi :
 - Lorsqu'un front montant est détecté, la valeur du compteur est enregistrée dans le registre CCR que vous pouvez lire à l'aide d'une fonction HAL (vous devez vous rappeler de cette valeur) ;
 - Au prochain front descendant, on lit de nouveau la valeur du registre CCR (la différence avec la valeur précédente nous donne la durée du signal HIGH) ;
- **Attention : la configuration du *prescaler* et *ARR* influence la précision de votre mesure.**
- La fréquence de votre mesure doit être de 5Hz (5x par seconde).

Important : Le déclenchement de la mesure de distance (envoi d'un signal sur TRIG) **doit se faire par interruption, donc la variable contenant la mesure doit se mettre à jour continuellement en arrière-plan (pas de *polling mode*)**. Si le capteur de distance ne détecte aucun objet, la variable doit conserver sa valeur précédente.

Barème

Vous devez être capable d'envoyer l'impulsion TRIG, de mesurer la distance et de l'afficher sur l'écran en temps réel. Le capteur devrait être en mesure de fournir une mesure sur une distance allant de 2 à 60 cm.

/1 pt	Fréquence de la mesure (5Hz = 2pts, 1Hz = 1pt)
/2 pts	Justesse de la mesure (2cm = 2pts, 5cm = 1.5pts, 10cm = 1pt, 20cm = 0.5pt)
/3 pts	Déclenchement de la mesure et mesure par interruption

Semaine 2

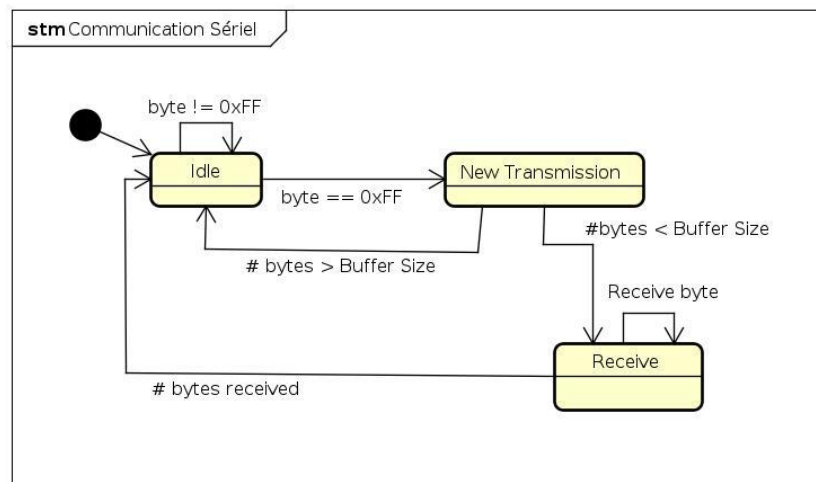
Objectif A2 : Communication UART

Le second objectif de l'équipe A est de mettre en place la communication UART pour échanger des données avec une autre carte. Pendant le jeu, vous devez échanger des informations de position, différents états du jeu, et l'état du combat en utilisant l'interface UART. Le fonctionnement ainsi que les détails techniques sont présentés ci-dessous.

Pour faire la connexion entre deux cartes NUCLEO il faut configurer l'UART5 pour éviter des interférences avec l'écran. Gardez les paramètres par défaut (Baud Rate 115200, Word Length 8 Bits, Parity None, Stop Bits 1, etc.). Activez le DMA UART5_TX et activez également l'interruption globale. Connectez les broches PD2 (Rx) et PC12 (Tx) d'une carte avec les broches PC12 (Tx) et PD2 (Rx) d'une autre carte de façon qu'une broche Rx soit connecté avec une broche Tx.

Astuce : Pour vous permettre de faire des tests de manière individuelle, dans un premier temps, vous utiliserez une seule carte et vous court-circuiterez Tx avec Rx de votre carte, de sorte que vous recevrez tous les messages que vous enverrez.

La réception des signaux sériels se fait en utilisant le mode interruption. Dans le gestionnaire d'interruption, vous implémentez la machine d'état montrée dans le schéma suivant. Cette implémentation permet de détecter de nouveaux messages et de lire des messages d'une longueur variable. L'émission des messages peut être effectuée en mode DMA.



Barème

Vous devez être capable d'échanger des messages en temps réels pour afficher les positions du Pac-Man et du fantôme ou de l'état de la phase combat par exemple. L'information doit être correcte (la même) sur les deux écrans.

/1 pt	Transmission et réception de message à partir de l'UART
/1 pt	Envoie de message de différentes tailles possibles
/1 pt	Communication entre les deux cartes en temps réel

Objectif B2 : Génération du son

Le second objectif de l'équipe B est d'être capable de jouer ces notes à partir du microcontrôleur :

Note	Fréquence (Hz)
B3	246.94
C4	261.63
D4	293.66
D4#	311.13
E4	329.63
F4#	369.99
G4	392.00
A4	440
B4	493.88
C5	523.25
D5	587.33
D5#	622.25
E5	659.26
F5#	698.46
G5	783.99

Ces notes seront jouées lors de la phase de combat, à chaque fois que l'on réussira à s'aligner à l'aide du capteur de distance ultrason sur le point généré sur l'axe vertical (point rouge ci-dessous) :



L'axe vertical de l'écran (240 pixels) sera divisé en 15 sections n pixels. Chaque section représentera une note du tableau ci-dessus (B3 pour la section commençant à 0 pixel jusqu'à G5 pour la section terminant à 240 pixels). La note jouée est donc déterminée en fonction de la position du point sur lequel on s'est aligné à l'écran.

Pour générer un son à partir du microcontrôleur, vous devez utiliser une des 2 méthodes ci-dessous.

Méthode de génération :

- Équipe B composée de 2 membres ;
vous **devez** (obligatoirement) utiliser le DAC en DMA (méthode du laboratoire 8).
- Équipe B composée d'1 membre ;
vous pouvez utiliser deux ondes carrées inverses (méthode du laboratoire 2, énoncé 2) ou la méthode de votre choix (il n'y a pas d'obligation).

Barème

Vous devez être capable de jouer chacune des notes présentées dans le tableau avec un signal PWM (selon la méthode requise) et afficher cette note à l'écran. La note jouée doit changer en fonction de la position de notre main par rapport au capteur de distance.

/1 pt	Génération d'un son à une fréquence déterminée
/1 pt	Changement de la note en fonction de la distance mesurée par le capteur ultrason
/1 pt	Génération du son avec la méthode requise

Semaine 3 : Intégration

À la troisième semaine, vous intégrez les différentes parties implémentées pendant les deux semaines précédentes. Le but est qu'à la semaine 4, notre jeu fonctionne minimalement tel que décrit à la section **Description de l'application**.

Vous devez intégrer :

- L'accéléromètre (partie A1) pour le choix du personnage dans le menu de départ.
- L'accéléromètre (partie A1) pour le mouvement du personnage dans le labyrinthe.
- La communication série (partie A2) pour échanger l'état du jeu entre les deux cartes.
- Le capteur de distance ultrason (partie B1) pour le gain de puissance pendant le combat.
- La génération du son (partie B2) pour émettre une note lors d'un gain de puissance pendant le combat.

Barème

/0.25 pt	Choix du personnage via l'accéléromètre
/0.25 pt	Déplacement de son personnage via l'accéléromètre
/0.75 pt	Déplacement du personnage adverse en temps réel
/0.5 pt	Rafraichissement de l'écran à 50Hz minimum
/0.5 pt	Justesse du gain de puissance pendant le combat
/0.75 pt	Justesse de la note émise lors d'un gain de puissance
/0.75 pt	Mise à jour de la barre de puissance de l'adversaire en temps réel
/0.75 pt	Fin de partie synchronisée entre les deux cartes
/1.5 pts	Fonctionnement général de l'application (jeu agréable à jouer)
/2 pts	Amélioration/personnalisation du jeu

Semaine 4 : Évaluation

Compétition : Évaluation fonctionnelle du jeu (en classe)

Vous devez finaliser le jeu pour permettre minimalement de jouer tel que décrit dans la section **Description de l'application**. En outre, il vous appartient d'ajouter des fonctionnalités pour rendre le jeu plus intéressant et attractif, obtenir les meilleures notes de la part de vos collègues et possiblement gagner la compétition.

Pour l'évaluation, vous réaliserez une vidéo de démonstration du jeu que vous déposerez sur YouTube (en mode non-référencé) et vous enverrez le lien au plus tard le mercredi 27 novembre à 8h30 dans le dépôt Moodle préparé à cet effet. Dans cette vidéo, vous présenterez votre équipe et votre prototype comme si c'était un produit commercial. Mettez l'accent sur tout ce qui pourrait vous démarquer de vos « concurrents ».

La vidéo doit durer 90 secondes maximum. Si vous dépassez, ne serait-ce que d'une seconde, vous perdez automatiquement un point.

Barème

/3 pts	Évaluation par les pairs selon la méthode ci-dessous
/1 pt	Bonus de la meilleure démonstration lors de la compétition

Note : Le gagnant devra faire une démonstration live d'une partie en mode deux joueurs et tenir au moins une minute sans bug :)

Vous serez évalués par les autres équipes au moyen d'une version simplifiée du suffrage par jugement majoritaire.

En gros, vous devrez noter chaque équipe avec une mention parmi les suivantes : **Excellent (3pts)**, **Très bien (2pts)**, **Bien (1pt)**, **Passable**, **insuffisant ou très insuffisant (0pt)**.

Votre note sera celle qui rallie minimum 50% des suffrages.

Par exemple, si vous avez recueilli les votes suivants :

- 15% Excellent
- 45% Très bien
- 35% Bien
- 5% Passable, insuffisant ou très insuffisant

Au moins 50% des étudiants pensent que vous méritez « Très bien » ou plus (45% + 15% des votes). Vous obtiendrez donc la note « Très bien », soit 2pts.

Soyez bons joueurs ! On se réserve le droit de supprimer des voteurs ou des votes trop éloignés de la tendance ou encore d'ajuster les notes pour parer à toute situation imprévue.

Que le meilleur gagne !!

Évaluation technique (en laboratoire)

La dernière semaine est dédiée à l'évaluation de toutes les composantes du projet. Un horaire sera établi pour que chaque équipe puisse se faire évaluer en laboratoire.

Chaque étudiant.e doit se faire évaluer, c'est-à-dire répondre aux questions des chargés de laboratoire. Si un.e étudiant.e n'est pas capable d'expliquer certaines parties du code, sa note sera modulée en fonction.

Pour toutes les équipes ; vous déposerez sur Moodle la veille de votre évaluation :

1. Une archive de votre projet Keil, incluant le fichier de configuration STM32Cube.
2. Un document PDF qui détaille très précisément:
 - a. Sur quelles broches du microcontrôleur vous connectez quels signaux.
 - b. Un schéma électrique de votre montage.
 - c. Des photos de votre montage.
 - d. Le lien vers votre vidéo de démonstration.
 - e. Tout détail pertinent pour être capable de faire des mesures sur votre prototype.

Ces informations doivent permettre au chargé de laboratoire de refaire votre montage rapidement et de programmer votre carte pour faire des vérifications le cas échéant.