

TP 1

NATURE DES INFORMATIONS TRANSITANT DANS UN SYSTÈME

TP : CRÉER UN JEU DE TYPE « PONG » MONO OU MULTIJOUEUR

Compétences

Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs.

Connaissances associées

Forme et transmission du signal.
Capteur

Objectif du TP :

- L'objectif est de créer un jeu de type « pong ». La raquette devra être pilotée par un potentiomètre

Objectif pédagogique

- Découvrir l'acquisition d'un signal analogique
- Découvrir le capteur potentiométrique
- Réaliser l'interface graphique associée à un capteur

Prérequis:

- Fiche sur mblock
- Proportionnalité
- Notions de coordonnées dans un plan.

1 MATÉRIEL ET LOGICIELS NÉCESSAIRES

- Mblock doit être préalablement installé
- Une carte Arduino Uno
- Un câble USB
- Une carte Groove Starter Kit
- Un (ou deux) capteur(s) potentiométriques sur l'entrée analogique A0 (et A1)
- Une (ou deux) nappe(s) de raccordement

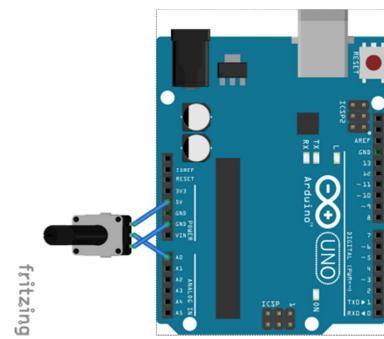


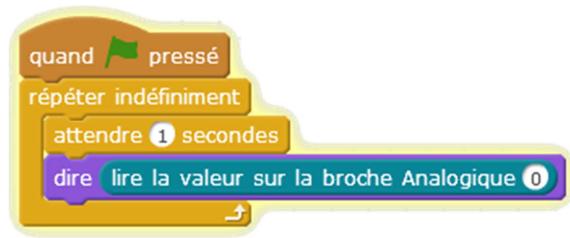
Schéma de câblage équivalent
(sans carte Groove)

2 ACTIVITÉS PRÉLIMINAIRES

2.1 Mise en œuvre du potentiomètre

Activité :

- Réaliser le branchement de la carte et des capteurs.
- Réaliser sur mblock le programme ci-contre.
 - ⇒ Quel est le but du programme ?
- En mode « On-line » téléverser le programme sur la carte.
 - ⇒ Quelle sont les valeurs minimales et maximales données par le potentiomètre.



2.2 Adaptation de l'information

Activité : régler le gain du capteur

- Estimer la variation angulaire du potentiomètre.
- On souhaite qu'en position initiale, (angle nul), le panda affiche 0 et qu'en position maximale, le panda affiche l'angle maximal.
 - ⇒ Quelle est la relation de proportionnalité entre l'angle du potentiomètre ?
 - ⇒ Modifier le programme pour que le panda affiche l'angle ?

Activité : modifier l'échelle de mesure angulaire

- On souhaite maintenant centrer l'information soit centrée, c'est-à-dire qu'on veut que le panda affiche 0 à mi-course.
 - ⇒Modifier le programme pour tenir compte de cette nouvelle contrainte.

Activité : modifier l'échelle de mesure linéaire

- La fenêtre d'affichage est comprise entre -240 et +240 pixels en abscisses
 - ⇒ Modifier le programme pour que le panda donne la position du potentiomètre entre -240 et 240.

3 RÉALISATION DU JEU

Activité : modifier le comportement de la raquette

- Ouvrir le fichier TP_01_05_Elève.sb2.
 - ⇒ Modifier le script de la raquette pour qu'elle suive le mouvement du potentiomètre.

4 CHALLENGE

- Modifier le programme pour pouvoir jouer à 2 joueurs !

**FORMATION À L'ENSEIGNEMENT DE
L'INFORMATIQUE AU COLLÈGE**

**INFORMATIQUE
COLLÈGE**

TP 2

NATURE DES INFORMATIONS TRANSITANT DANS UN SYSTÈME

TP : COMMANDER UN MOTEUR À COURANT CONTINU

Compétence

Écrire, mettre au point (tester, corriger) et exécuter un programme commandant un système réel et vérifier le comportement attendu.

Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs.

Connaissances associées

Systèmes embarqués.

Capteur, actionneur, interface

Objectif du TP :

- L'objectif est de piloter un moteur à partir d'un capteur potentiométrique ou de tout autre capteur (ultra son, capteur de température, capteur de lumière...).
- L'application finale pourrait être la commande d'un store en fonction de la luminosité d'une chambre.

Objectif pédagogique

- Découvrir le pilotage d'un moteur à courant continu.
- Réaliser un programme et l'implémenter dans un système embarqué

Prérequis:

- Fiche sur mblock
- Connecter un potentiomètre et acquérir l'information.

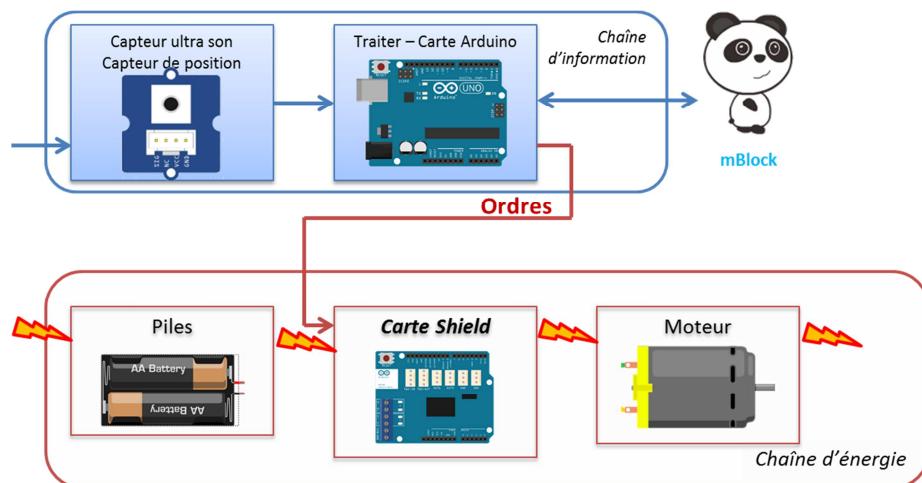
1 MATÉRIEL ET LOGICIELS NÉCESSAIRES

- Mblock doit être préalablement installé
 - Une carte Arduino Uno
 - Une carte Arduino Shield
 - Un câble USB
 - Une carte Groove Starter Kit
 - Un capteur potentiométrique sur l'entrée analogique A0
 - Une nappe de raccordement
 - Un moteur à courant continu
 - Une alimentation externe (piles, batterie).

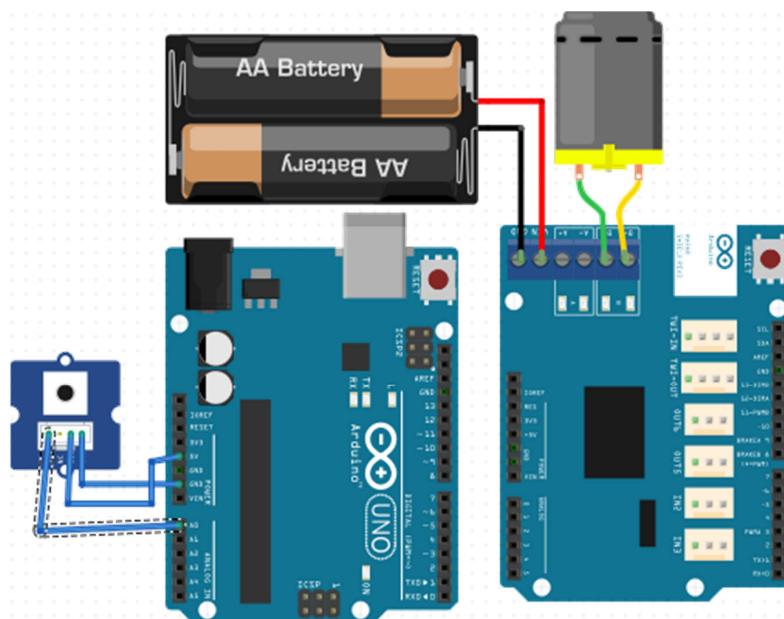
Remarque :

Une carte Arduino n'est pas une carte permettant de piloter des actionneurs demandant de la puissance. Le courant sur les sorties est limité à 40 mA. On utilise donc une carte « Motor shield ». Celle-ci permet en effet de délivrer un courant allant jusqu'à 4 A.

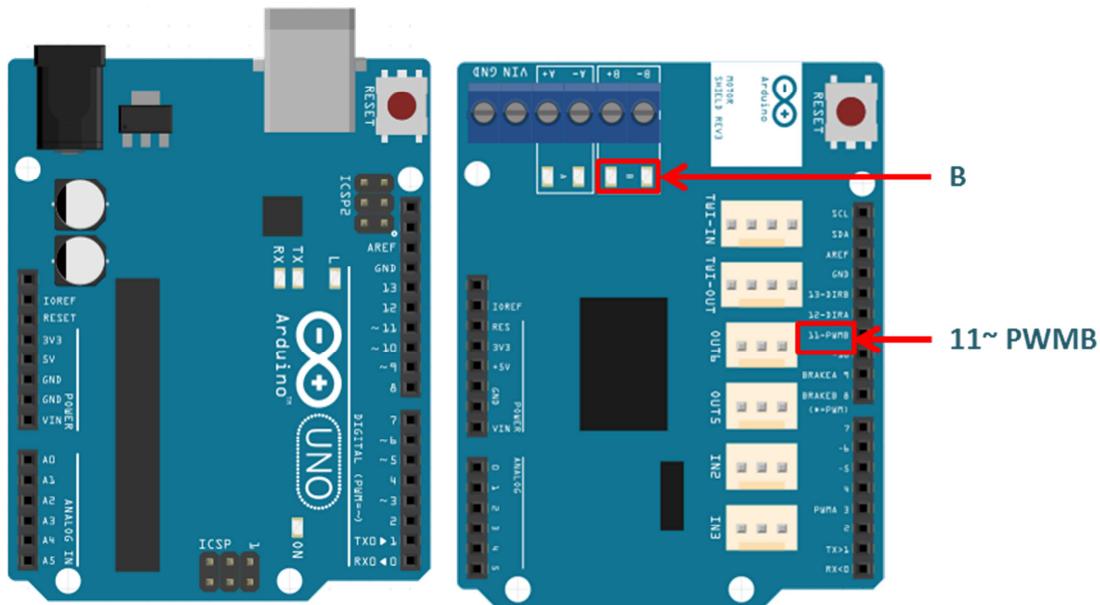
Avec la carte Arduino, ainsi que la carte shield, la chaîne fonctionnelle, devient la suivante.



La carte shield vient se brancher sur la carte Arduino. Le plan de câblage est le suivant :



Si l'alimentation est de 9V, on pourra donc envoyer au maximum 9V au moteur. Pour piloter le moteur, on va commander la sortie numérique 11 de la carte Arduino. La valeur de commande varie de 0 à 255. Si on note v la valeur adressée à la sortie 11, la tension d'alimentation du moteur sera $\frac{v}{255} \cdot 9$ (tension d'alimentation de 9V).



2 ACTIVITÉS PRÉLIMINAIRES

2.1 Mise en œuvre du potentiomètre

Activité : brancher le potentiomètre

- ❑ Réaliser les branchements.
 - ❑ Réaliser sur mblock le programme ci-contre.
 - ❑ En mode « On-line » téléverser le programme sur la carte.
 - ⇒ Quel est le but de ce programme ?
 - ⇒ Quel est le rôle de la division par 4 ?.



2.2 Mise en œuvre du moteur

Activité : brancher le moteur

- Réaliser le programme ci-contre.
 - Brancher les piles.
 - En mode « On-line » téléverser le programme sur la carte.
 - ⇒ Observer le comportement du moteur.



3 PILOTAGE DU MOTEUR

3.1 Pilotage du moteur par le potentiomètre

Activité : piloter le moteur

- En utilisant les deux activités précédentes, réaliser un programme permettant de piloter le moteur à partir du potentiomètre.

3.2 Pilotage du moteur en fonction de la luminosité

Objectif :

- Préparer un programme permettant de faire tourner le moteur en fonction d'un seuil.
 - Réaliser une application pour système embarqué.

Activité : câbler et programmer le capteur de luminosité

- ❑ Câbler le capteur de luminosité.
 - ⇒ Réaliser un programme permettant de déterminer un seuil de jour et de nuit.

Activité : câbler et programmer le moteur en fonctionnement de la luminosité

- Réaliser un programme permettant d'allumer le moteur avec un PWM de 255 en fonction d'un seuil de luminosité de 500.

Activité : réaliser une application embarquée

- Réaliser un programme permettant d'être embarqué sur la carte Arduino et le tester dans la salle.

4 CHALLENGE

Activité : réaliser une application embarquée

- Vous avez remarqué que le moteur tourne toujours dans le même sens.
 - ⇒ Réaliser un programme pour que le moteur tourne dans un sens quand la luminosité passe au-dessus d'un certain seuil et pour qu'il tourne dans le sens opposé quand la luminosité passe au-dessous de ce seuil.
- Modifier le programme pour que le système réponde au cahier des charges suivant : la luminosité doit rester entre deux seuils.
 - ⇒ On définit un seuil haut à 700.
 - ⇒ On définit un seuil bas à 300.
 - ⇒ Lorsqu'on passe au-dessus du seuil haut :
 - Le moteur tourne dans un sens pendant 5s.
 - ⇒ Lorsqu'on passe au-dessous du seuil bas :
 - Le moteur tourne dans l'autre sens pendant 5s.

FORMATION À L'ENSEIGNEMENT DE L'INFORMATIQUE AU COLLÈGE

INFORMATIQUE
COLLÈGE

TP 3

LE JEU DU SIMON

Compétences

Analyser le comportement attendu d'un système réel et décomposer le problème posé en sous problèmes afin de structurer un programme de commande.

Écrire, mettre au point (tester, corriger) et exécuter un programme commandant un système réel et vérifier le comportement attendu.

Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs.

Connaissances associées

Notions d'algorithme et de programme.

Notion de variable informatique

Déclenchement d'une action par un événement, séquences d'instructions, boucles instructions conditionnelles.

Objectif du TP

- Les élèves devront réaliser en plusieurs étapes la programmation du jeu du Simon expliqué dans la suite.

Objectif pédagogique

- L'objectif est de découvrir la notion d'entrées et sorties tout ou rien (TOR) ou digitales ainsi que les instructions telles que les boucles et les conditions.

1 PRÉSENTATION DU JEU

Le Simon est un jeu permettant de tester la mémoire. Il s'agit de reproduire une séquence lumineuse sans se tromper.

Simon éclaire une des quatre couleurs des boutons rétroéclairés (rouge, bleu, jaune, vert) et produit un son toujours associé à cette couleur. Il faut alors appuyer sur la touche de la couleur qui vient de s'allumer. Le jeu répète la même couleur puis ajoute au hasard une nouvelle couleur. Il faut alors reproduire cette nouvelle séquence entièrement. Chaque fois que la séquence est reproduite correctement, Simon ajoute une nouvelle couleur. Le score maximal qu'il est possible d'atteindre est une séquence de 200.



2 PRÉPARATION

2.1 Matériel et logiciel nécessaires

Pour cette fiche, il vous faut :

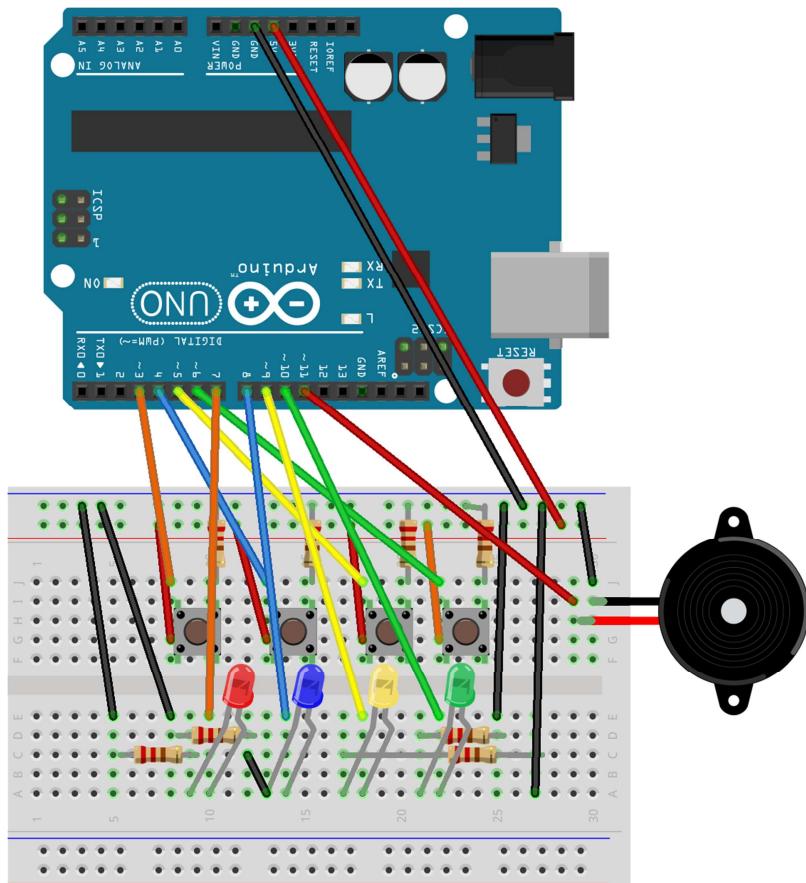
- Un ordinateur avec le logiciel Mblock installé ;
- Une carte Arduino Uno ;
- Un câble USB ;
- Une plaque d'essai ;
- 4 boutons poussoirs et 4 résistances de 1K ;
- 4 leds (une jaune, une rouge, une bleue, une verte) ;
- 4 résistances de 200 Ohms ;
- Un buzzer ;
- Des straps mâles-mâles.



2.2 Câblage

Les leds, boutons poussoir et buzzer sont directement montés sur la plaque d'essai à l'aide des résistances.

Le câblage suivant indique les connections à réaliser.



fritzing

Le tableau suivant recense les pins digitaux associés aux leds et boutons poussoirs.

Pins Digitaux Arduino	Type
3	Bouton rouge
4	Bouton bleu
5	Bouton jaune
6	Bouton vert
7	Led rouge
8	Led bleue
9	Led jaune
10	Led verte
11	Buzzer

2.3 Préparation de mBlock

- Brancher la carte Arduino par l'intermédiaire du câble USB.
- Lancer le logiciel mBlock.
- Dans le menu Connecter, choisir Port Série et sélectionner le COM disponible.
- Cliquer ensuite dans Connecter sur Mettre à jour le microprogramme pour travailler directement en liaison avec la carte Arduino (mode On line).
- À la fin du téléchargement, cliquer sur Fermer.

3 ACTIVITÉ 1 – DÉCOUVERTE DES SORTIES DIGITALES

Activité :

- Créer le programme suivant en faisant glisser depuis Pilotage les blocs Arduino (verts) et depuis Contrôle ou Événements les autres blocs.
- Cliquer sur  dans la scène pour lancer le programme en mode On-line. Le stopper en cliquant sur .

 - ⇒ Quel est le but du programme ?
 - ⇒ À quoi correspondent le niveau haut et le niveau bas ?
 - ⇒ Comment appelle-t-on ce type d'informations ?

- Modifier le programme pour faire clignoter plus rapidement la led.



Activité :

- Ajouter de nouvelles instructions pour faire clignoter les 4 leds en même temps.

Activité :

- Modifier les instructions pour que les leds s'allument successivement (rouge puis bleu puis jaune puis vert). On réalise ainsi un chenillard.

Activité :

- Modifier une instruction pour que le chenillard n'exécute que trois fois la suite de couleurs

4 ACTIVITÉ 2 – DÉCOUVERTE DES ENTRÉES DIGITALES

Activité:

- Réaliser le programme suivant en faisant glisser depuis Pilotage les blocs Arduino (verts), depuis Apparence le bloc dire Bonjour (violet) et depuis Contrôle ou Événements les autres blocs.
- Cliquer sur  dans la scène pour lancer le programme en mode On-line. Le stopper en cliquant sur . Appuyer sur le bouton rouge puis le relâcher.
 - ⇒ Qu'observe-t-on ?
 - ⇒ Quelles sont les valeurs disponibles lors de l'appui ou non sur le bouton poussoir.
- Comment appelle-t-on ce type d'information ?



Nous allons maintenant réaliser un programme permettant de contrôler l'allumage des leds en fonction de l'appui sur le bouton poussoir.

Activité :

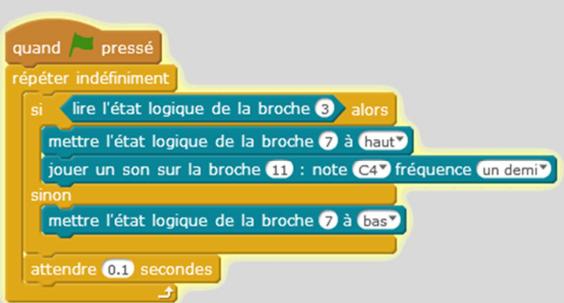
- Modifier le programme pour prendre en compte une instruction de type Condition (dans Contrôle).
- Tester le programme.
- Modifier le programme pour faire la même chose sur chaque led de couleur (l'appui sur le bouton rouge allume la led rouge, l'appui sur le bouton jaune allume la led jaune).



Le petit plus

Activité :

- Ajouter la commande de Pilotage pour jouer un son sur le buzzer branché sur la broche 11. Tester différentes notes ou différentes fréquences.

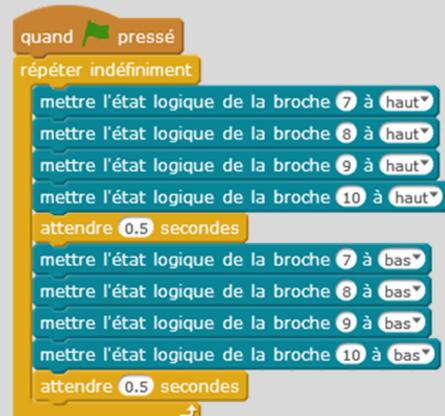


5 ACTIVITÉ 3 – GESTION DE DIFFÉRENTS MODES – FONCTIONS

Dans le jeu du Simon, il est possible de sélectionner différents modes de jeu (jeu à deux, répétition d'une suite inventée par le joueur, jeu de rapidité...). Nous allons voir que ce type de comportement nécessite de faire des tests et gérer les événements en fonction de conditions.

Activité :

- Créer un programme qui fasse clignoter toutes les leds en même temps. Tester votre programme.



Activité :

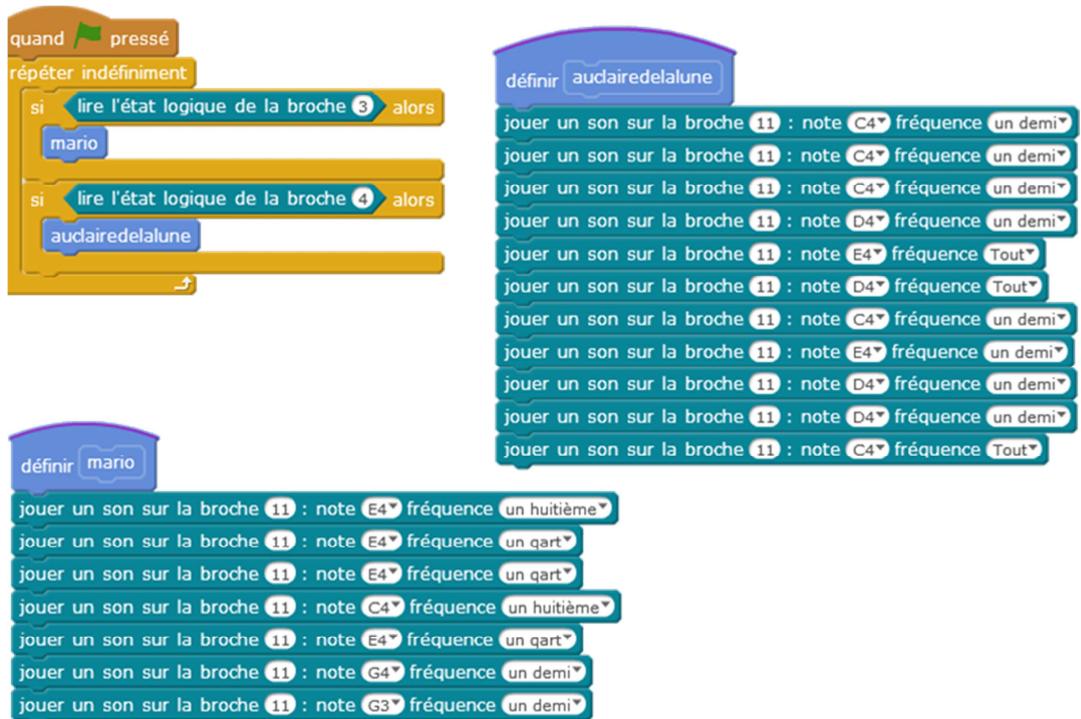
- Cliquer dans Blocs et variables sur Créer un bloc. Renseigner le nom **blink** (clignotement). Un bloc apparaît dans la fenêtre de programmation. Il s'agit de la définition d'un sous-programme ou fonction.
- Prendre toutes les instructions mises dans répéter indéfiniment et les faire glisser dans le bloc Définir **blink**.
- Dans répéter indéfiniment faire glisser depuis Blocs le bloc **blink**. Tester le comportement.



Par cette technique, il est possible de faire plusieurs fonctions et de définir un programme principal qui utilisera ces fonctions.

Activité :

- Créer de 2 à 4 fonctions différentes que vous nommerez comme vous voulez en utilisant les leds ou le buzzer (on peut créer ainsi 4 musiques !).
- Dans le programme principal, introduire des conditions en fonction de l'appui sur les boutons pour appeler une des fonctions créées.



6 ACTIVITÉ 4 – GESTION DU JEU SIMON

Le programme principal du jeu est défini de la manière suivante.



6.1 Fonction SimonDit

La fonction SimonDit est proposée ci-contre. Une variable N est introduite.

Activité :

- Réaliser ou ouvrir le fichier JeuSimon
 - ⇒ Quelle valeur peut prendre N au cours de l'exécution du programme.
 - ⇒ Que se passe-t-il si N est égale à 2.



6.2 Fonction Attente

La fonction Attente permet d'attendre que le joueur appuie sur une des 4 touches.

6.3 Fonction Action

Ensuite la fonction Action est appelée dès qu'une touche est appuyée. Une nouvelle variable APPUI est utilisée. Lorsqu'une touche est appuyée, la broche Entrée logique correspondante est mise au niveau logique Bas. A l'inverse lorsqu'elle n'est pas appuyée, cette entrée vaut Haut.

Activité :

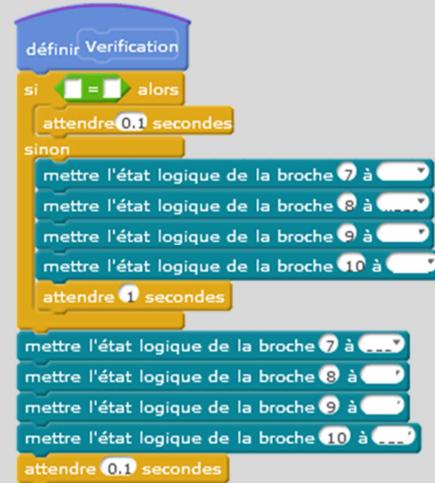
- Réaliser cette fonction ou ouvrir le fichier JeuSimon
 - ⇒ Indiquer ce qu'il se passe lorsque le bouton rouge est appuyé.

6.4 Fonction Vérification

Le jeu procède ensuite à une vérification pour savoir si la touche appuyée correspond bien à ce qui était demandé. Si la touche ne correspond pas, toutes les leds s'allument une seconde puis elles s'éteignent pendant une seconde. Si la touche correspond, les leds s'éteignent toutes et la procédure complète peut redémarrer.

Activité :

- Ouvrir le fichier JeuSimon
 - ⇒ Compléter le sous-programme Verification en renseignant les éléments vides pour que la spécification précédente soit respectée.
 - ⇒ Tester votre programme



Activité :

- En conservant les mêmes fonctions, que faut-il modifier pour respecter le jeu complet du Simon, c'est-à-dire répéter une séquence augmentant progressivement.
- On pourra réaliser ce programme en introduisant une variable de type Liste dans laquelle on stockera les valeurs à jouer.

FORMATION À L'ENSEIGNEMENT DE L'INFORMATIQUE AU COLLÈGE

INFORMATIQUE
COLLÈGE

TP 4

INTERAGIR AVEC UN JEU VIDÉO PAR UN PÉRIPHÉRIQUE BLUETOOTH

Compétences

Analyser le comportement attendu d'un système réel et décomposer le problème posé en sous problèmes afin de structurer un programme de commande.

Écrire, mettre au point (tester, corriger) et exécuter un programme commandant un système réel et vérifier le comportement attendu.

Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs.

Programmer des scripts de déroulant en parallèle.

Connaissances associées

Notions d'algorithme et de programme.

Déclenchement d'une action par un événement, séquences d'instructions, boucles instructions conditionnelles.

Capteur, actionneur, interface.

Objectif du TP

L'objectif est de découvrir comment utiliser un périphérique Bluetooth avec un jeu vidéo. Excepté sa configuration, son utilisation reste semblable à celui d'un clavier ou souris.

Les élèves devront modifier un programme existant pour prendre en compte une des fonctionnalités du périphérique (accéléromètre) plutôt qu'une autre déjà proposée.

1 PRÉPARATION

1.1 Matériel et logiciel nécessaires

Pour cette fiche, il vous faut :

- Un ordinateur avec le logiciel Scratch installé (cf. fiche outil) ;
- Le logiciel GlovePie40 et le fichier script_wiimote ;
- Une Wiimote ;
- Une clé Bluetooth 4.0 ;



1.2 Appairage de la Wiimote

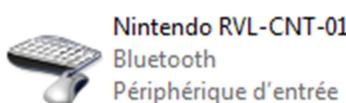
Cette opération n'est à effectuer que la première fois que la Wiimote est reconnue sur un PC équipé de Windows.

Lorsque la clé Bluetooth (type 4.0) est connectée au PC et que les drivers sont bien installés, un icône  apparaît en bas à droite de l'écran dans la barre des tâches.

Cliquer sur cet icône et choisir Ajouter un périphérique. Appuyer sur le bouton rouge de la Wiimote. Cliquer sur Appairer sans utiliser de code de couplage.



Des drivers sont installés sur le PC spécifiquement pour le périphérique et le nom suivant apparaît dans la



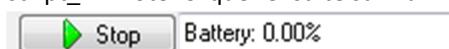
fenêtre de recherche de périphériques

. Une fois l'opération réussie le PC et

la manette sont appairés. La Wiimote doit clignoter.

1.3 Reconnaissance de la Wiimote

Lancer le logiciel GlovePie en cliquant sur GlovePIE_Bird_5DT.exe. Cliquer sur File/Open et choisir script_wiimote. Cliquer ensuite sur Run. L'indication Battery doit être affichée.



1.4 Lancement de Scratch

Lancer le logiciel Scratch et ouvrir le fichier WiimoteCapteurs.sb2

2 ACTIVITÉ 1 – DÉCOUVERTE DES CAPTEURS DE LA WIIMOTE

Lancer le logiciel Scratch et ouvrir le fichier WiimoteCapteurs.sb2

Activité:

- Cliquer sur  dans la scène pour lancer le programme. Bouger la Wiimote et appuyer sur chaque bouton. Vérifier que tous les boutons marchent.
 - ⇒ Indiquer autour de quels axes vis-à-vis de la Wiimote se font les mouvements associés à gx, gy, gz, rx, ry, rz
 - ⇒ Quelles sont les valeurs que peut prendre chacune de ces grandeurs
 - ⇒ À quoi correspond la variable Shakiness. Quelles sont les valeurs qu'elle peut prendre ?

3 ACTIVITÉ 2 – MISE EN PLACE DES DÉPLACEMENTS DES PERSONNAGES

Ouvrir maintenant le fichier BougerWiimote.sb2. Ce fichier contient 3 acteurs : le chat, un crabe et une épée. L'objectif du jeu va être de déplacer le chat avec la Wiimote de manière à ce qu'il aille toucher avec son épée le crabe qui se déplace aléatoirement.



Cliquer sur le chat

Activité :

- Compléter la boucle infinie pour qu'à chaque fois qu'une des touches du pad directionnel de la Wiimote est égal à 1, le chat s'oriente à droite (90) ou à gauche (-90) et avance de +10 à droite ou -10 à gauche. Pour le déplacement vertical, on ajoutera +10 ou -10 à y pour que le chat monte ou descende

ajouter 10 à y

avancer de 10

s'orienter à 90°

En cliquant sur l'onglet Costume situé à côté de l'onglet Script, vous pouvez constater que le chat possède deux costumes qui sont associés à son déplacement. Pour changer de costume, on utilise les instructions suivantes :

costume suivant

basculer sur costume costume1 ▾

Activité :

- Ajouter un passage au costume suivant à chaque fois que la croix directionnelle de la wiimote est enfoncée.
- Ajouter une condition pour que si aucune touche de la croix directionnelle n'est enfoncée, on bascule automatiquement sur le costume1.

Les déplacements du chat sont donc prêts

Activité :

- Essayer de modifier les conditions d'appui sur les touches du pad directionnel pour utiliser à la place l'inclinaison de la Wiimote. Remettre les touches si le résultat n'est pas concluant.

Le déplacement du crabe est plus simple. Celui-ci doit avancer de 10 indéfiniment en changeant de costume à chaque fois. Tous les dix déplacements, il tourne d'un angle aléatoire entre 0 et 360°.

Activité :

- Compléter la première boucle infinie du comportement du crabe (cliquer sur l'icône du crabe). Tester le fonctionnement.

Le comportement de l'épée étant un peu plus complexe, on ne l'étudie pas et on laisse le comportement par défaut (l'épée suit le chat et s'oriente correctement en fonction des secousses faites par la Wiimote)

4 ACTIVITÉ 3 – INTERACTION ENTRE LE CRABE ET L'ÉPÉE

La dernière étape du jeu consiste à modifier le comportement du crabe lorsque celui-ci est touché par l'épée et que la Wiimote était secouée au-dessus d'un niveau donné.

Si ces deux conditions sont réunies alors on fait augmenter la taille du crabe (comme s'il explosait) en utilisant

mettre l'effet oeil de poisson ▾ à 100

l'instruction pendant 0.5 s et on joue le son pop puis on positionne de façon aléatoire le chat sur l'écran (x est compris entre -240 et 240, y est compris entre -180 et 80).

Activité:

- Mettre en place ce comportement dans la boucle infinie incomplète et tester le comportement du jeu.

Pour aller plus loin, on peut ajouter une variable nbTouche qui vaut 0 au début et qui lorsqu'on atteint 10 touches fait dire au chat : J'ai gagné et le fait miauler.

FORMATION À L'ENSEIGNEMENT DE L'INFORMATIQUE AU COLLÈGE

INFORMATIQUE
COLLÈGE

TP 5

TP : CRÉER UNE APPLICATION DE DOMOTIQUE

Compétence

Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs.

Connaissances associées

Forme et transmission du signal.
Capteur, Entrées/sorties

Objectif du TP :

- L'objectif est de créer une application sur smartphone qui permettra d'allumer / éteindre une lampe (led) à distance ou changer son intensité lumineuse à l'aide d'une interface dédiée

Objectif pédagogique

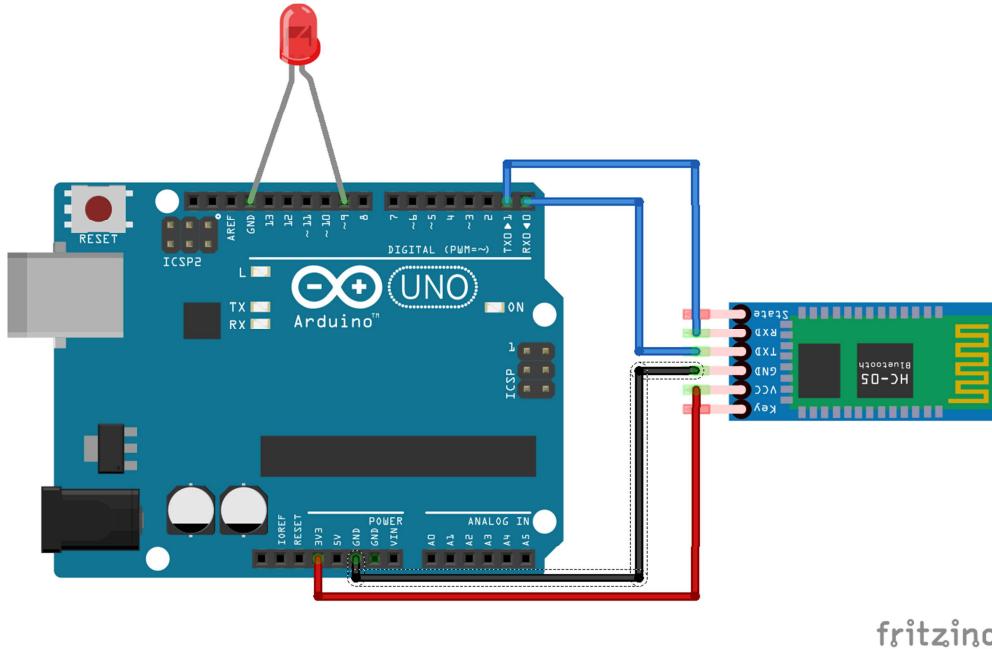
- Découvrir les éléments graphiques disponibles pour smartphone
- Découvrir les sorties logiques et analogiques
- Réaliser l'interface graphique associée à une sortie

Prérequis:

- Fiche sur AppInventor
- Fiche Arduino

1 MATÉRIEL ET LOGICIELS NÉCESSAIRES

- Une carte Arduino Uno
- Une led
- Une tablette android
- Une connexion internet wifi
- Une carte Bluetooth HC-05 et 4 câbles mâle-femelle



2 PRÉPARATION

- Charger au préalable dans la carte Arduino le fichier toolbox_arduino_v4.ino en ayant pris soin de débrancher les pattes RX et TX de la carte Arduino. Utiliser la fiche Arduino si nécessaire.
- Cette opération n'est à faire qu'une seule fois et peut être réalisée en dehors du TP une bonne fois pour toute.
- Brancher la carte Bluetooth et la led comme indiqué sur le schéma de câblage.

3 ALLUMER/ÉTEINDRE UNE LED PAR BOUTON

3.1 Ouverture d'une application sous AppInventor

Activité :

- Se connecter à AppInventor par l'adresse suivante et utiliser le compte gmail créé.
- Dans Projets cliquer sur Importer le projet aia de mon ordinateur et choisir le fichier ArduinoAppinventor.aia

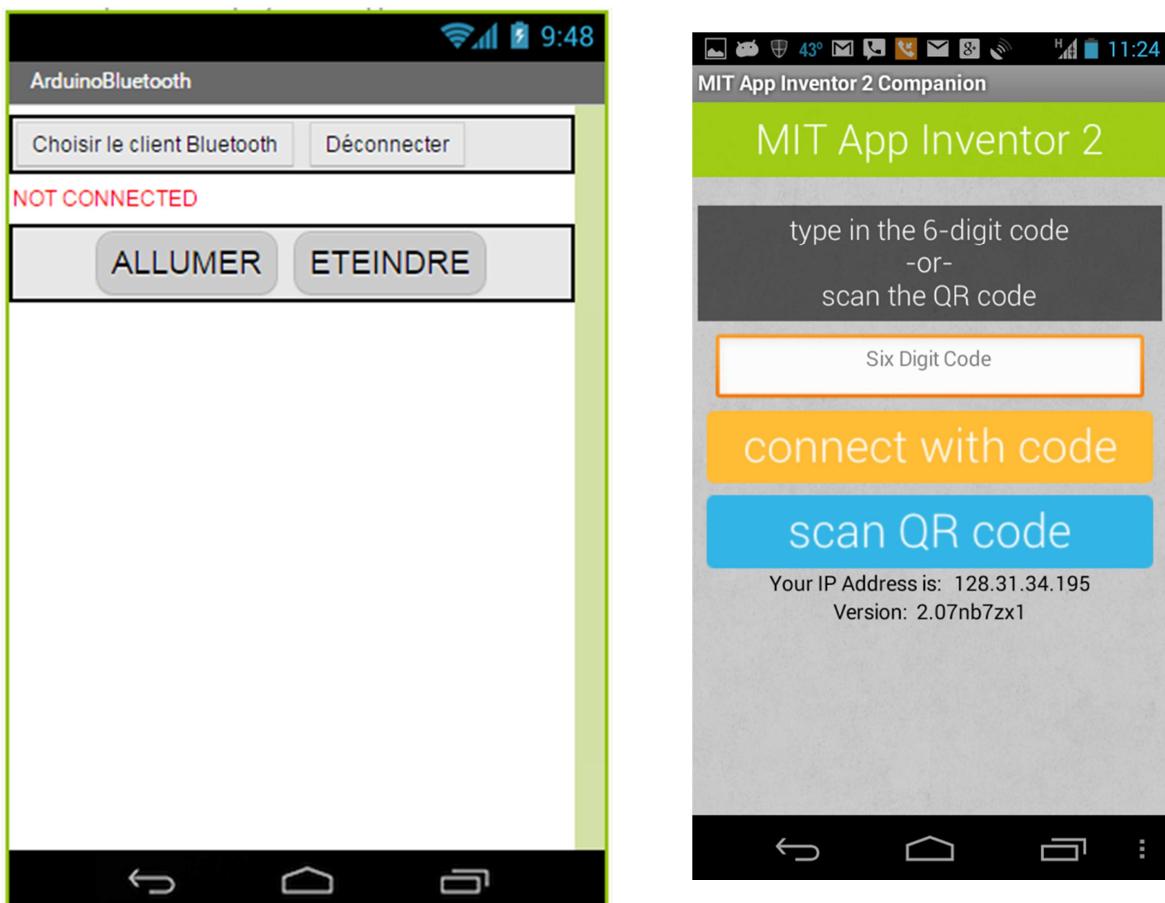
La partie communication Bluetooth est déjà présente dans cette interface.

Se référer à la fiche AppInventor pour de plus amples informations sur la description des menus si besoin.

3.2 Préparation de l'interface graphique

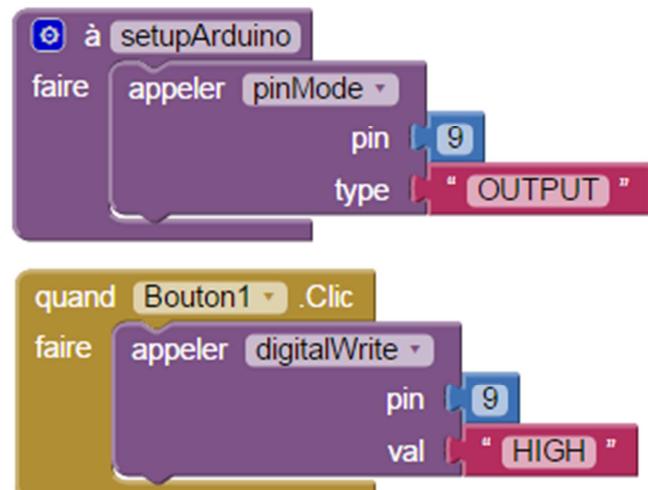
Activité : Ajouter deux boutons

- Faire glisser une zone d'arrangement horizontale puis modifier ses propriétés de largeur en choisissant Remplir parent.
- Faire glisser deux boutons et changer les propriétés pour obtenir une interface de ce type



Activité : Appliquer une fonction à chaque bouton

- Cliquer sur Blocs en haut à droite.
- Faire glisser depuis Procédure la fonction permettant de définir une sortie digitale sur le pin 9 (led)
- Faire ensuite glisser en cliquant sur Bouton1 la procédure qui permet de réaliser des instructions lorsqu'on clique sur le bouton. Ajouter l'instruction digitalWrite (dans Procédure) qui permet de mettre au niveau haut (HIGH) ou au niveau bas (LOW) le pin 9 de l'Arduino.
 - ⇒ Compléter de la même manière l'action associée au bouton2 (éteindre la led).



L'interface est maintenant prête, vous allez pouvoir tester.

Cliquez sur Connecte / Compagnon AI dans AppInventor. Sur votre tablette, assurez-vous que le bluetooth est bien activé puis cliquez sur AI Companion et appuyez sur Scan QRcode (cf. image ci-dessus). L'application se lance.

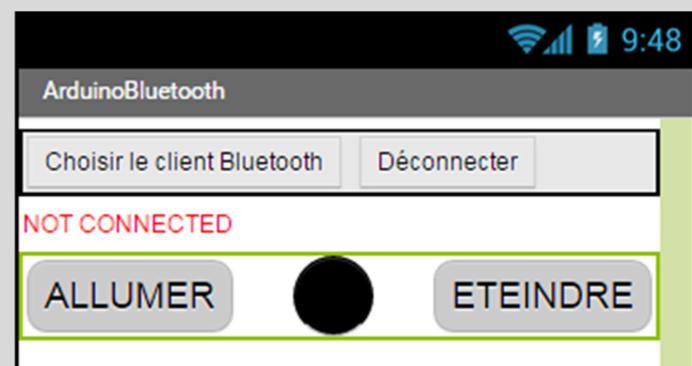
Activité : Test de l'allumage de la led

- Appuyer sur Choisir son périphérique et sélectionner le périphérique HC-05-1 s'il est disponible.
- Cliquer ensuite sur Allumer / Eteindre pour voir la led répondre correctement
- Appuyer sur Connecte/Réinitialiser connexion pour arrêter l'application. Si vous pensez à vous déconnecter du Bluetooth, vous pouvez observer vos changements dans l'application directement sur la tablette en laissant le compagnon connecté.

4 AMÉLIORATION DE L'APPLICATION

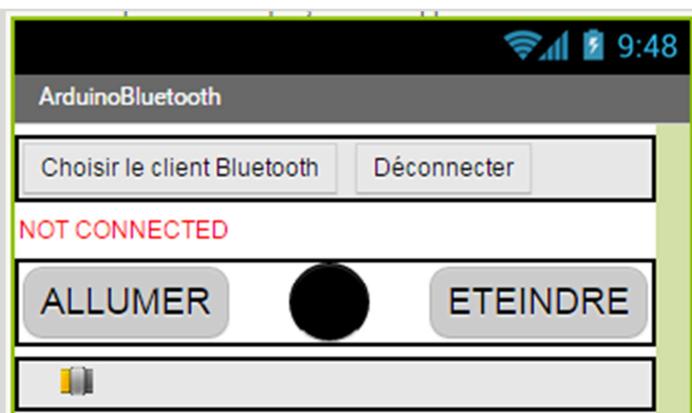
Activité : modifier le comportement des boutons

- Ajouter un bouton sans texte entre les deux boutons Allumer / Eteindre et mettre la couleur Noire par défaut.
 - ⇒ Modifier la zone de programmation pour que le bouton central (non actif) change de couleur comme la led !



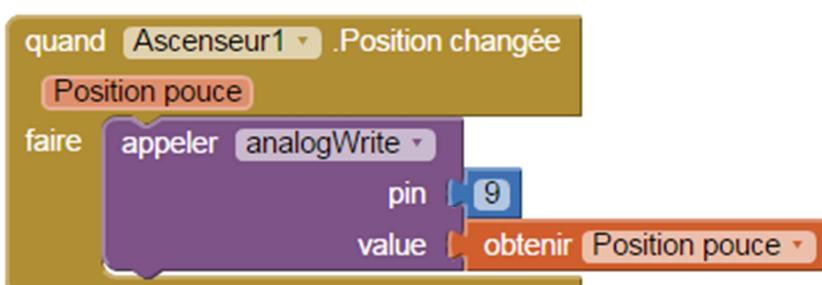
Activité : Ajouter un curseur

- Ajouter une nouvelle zone d'arrangement horizontal en dessous de la précédente et faire glisser un ascenseur horizontal.
 - ⇒Modifier / ajouter des labels vides pour que l'agencement soit satisfaisant
 - ⇒ Mettre des bornes allant de 0 à 255 pour l'ascenseur



Activité : Action associée au curseur

- Dans la partie programmation (blocs), faire glisser le bloc Quand ascenseur position changée et ajouter les autres blocs pour obtenir la commande suivante.
 - ⇒ Tester votre application en faisant varier le curseur. Que constate-t-on au niveau de la led ?



COMMUNICATION ENTRE LES SYSTÈMES

TP 6

TP : CONFIGURER UN CLIENT ET UN SERVEUR POUR UNE APPLICATION SIMPLE

Remarque :

Cette activité s'appuie sur celle proposée par Christian Weiss

<http://electronique.ac-bordeaux.fr/Contributions.php>

Connaissances

Comprendre le fonctionnement d'un réseau informatique

**Programme de
technologie**

Composants d'un réseau, architecture d'un réseau local, moyens de connexion d'un moyen informatique.

Notion de protocole, d'organisation de protocoles en couche, d'algorithme de routage

Internet

Objectif du TP :

- L'objectif est de configurer un ordinateur serveur et un ordinateur client.

Objectif pédagogique

- Trouver l'adresse IP d'un ordinateur
- Visualiser une trame

Prérequis:

- Application Client – Serveur
- Installation de Wireshark

1 MATÉRIEL ET LOGICIELS NÉCESSAIRES

- Au moins 2 PC reliés à un même réseau ;
 - Les mini logiciels « client et serveur » (fichier socket.zip disponible dans les exécutables)
 - Le logiciel Wireshark (disponible dans les exécutables)
 - Un convertisseur hexadécimal (par exemple
<http://sebastienguillon.com/test/javascript/convertisseur.html>)

2 ACTIVITÉS PRÉLIMINAIRES

- ❑ Un ordinateur doit être désigné comme étant le serveur.
 - ❑ Les autres ordinateurs doivent être les clients.

2.1 Déterminer l'adresse IP d'une machine

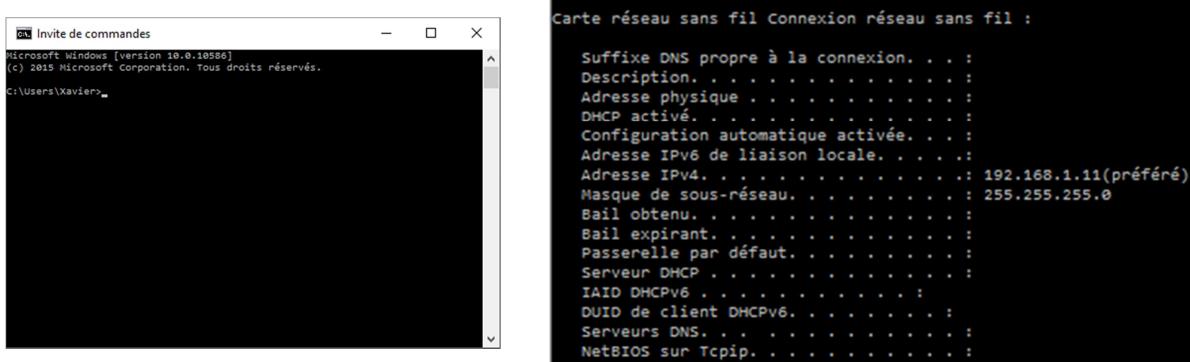
Objectif :

- #### □ Déterminer les adresses IP des machines du réseau.

Il existe plusieurs méthodes pour connaître l'adresse IP d'une machine.

2.1.1 Méthode 1 : Ligne de commande

- Exécuter CMD
 - Exécuter la ligne de commande ipconfig /all

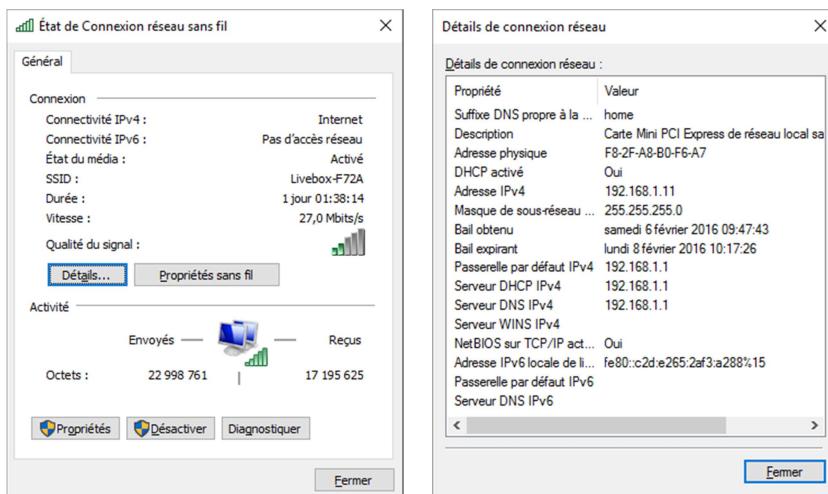


Dans la liste affichée :

- ❑ repérer la carte réseau utilisée : ici Carte réseau sans fil ;
 - ❑ repérer l'adresse IPv6 : ici 192.168.1.11 ;
 - ❑ repérer le masque de sous-réseau : 255.255.255.0.

2.1.2 Méthode 2 : interface graphique

- ❑ Aller dans les connexions réseaux (méthode dépendant de votre version de windows) ;
 - ❑ Rechercher les propriétés ou l'état de la connexion
 - ❑ Cliquer sur Détails



Dans la liste affichée :

- ❑ repérer la carte réseau utilisée : ici Carte réseau sans fil ;
 - ❑ repérer l'adresse IPv6 : ici 192.168.1.11 ;

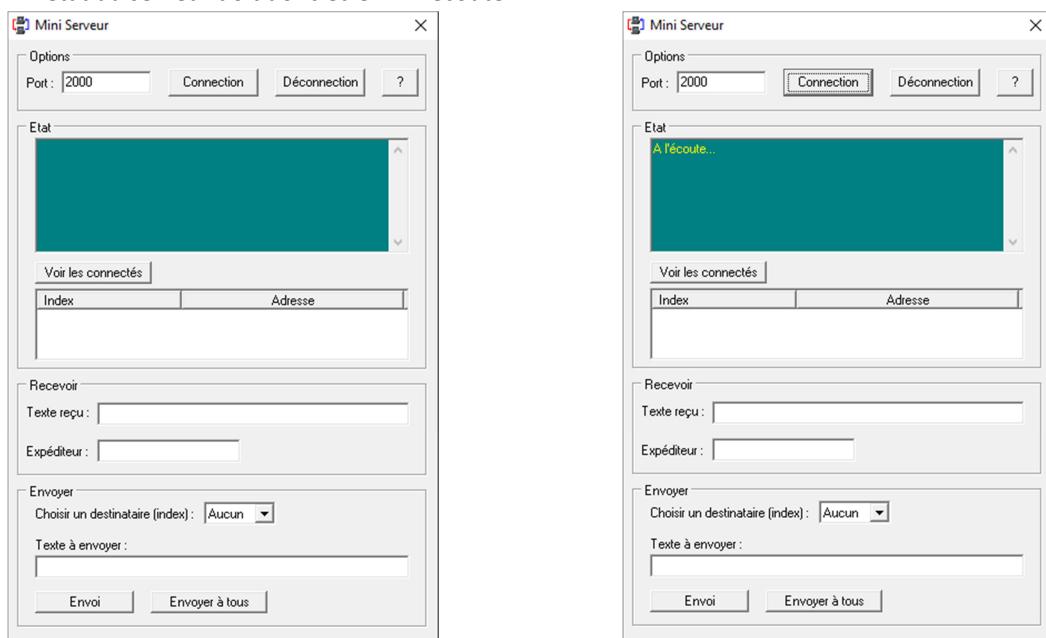
- repérer le masque de sous-réseau : 255.255.255.0.

Activité : Adresses IP

- Trouver l'adresse IP de l'ordinateur devant faire office de serveur et de celui (ou ceux) devant faire office de client.

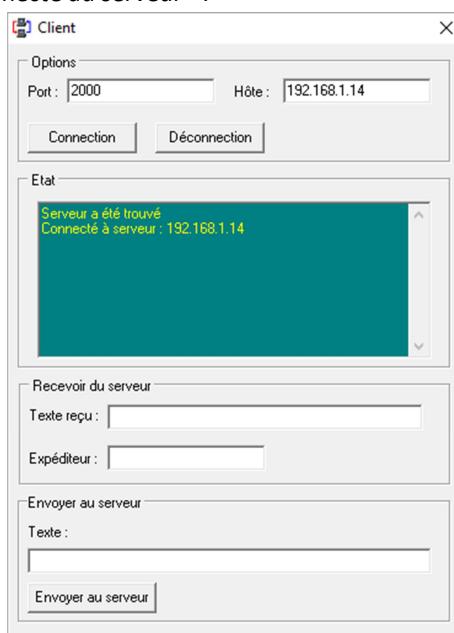
2.2 Configuration du serveur

- Ouvrir l'application serveur.
- Conserver le port 2000.
- Cliquer sur « Connecter ».
- Autoriser l'accès par le pare feu si Windows vous le demande.
- L'état du serveur doit donc être « A l'écoute »



2.3 Configuration du client

- Ouvrir l'application client.
- Dans la case « Hôte », saisir l'adresse IP du serveur.
- Cliquer sur « Connecter ».
- L'état doit afficher « Connecté au serveur ».



3 ENVOI DE MESSAGES

3.1 Utilisation de l'application client – serveur

Activité :

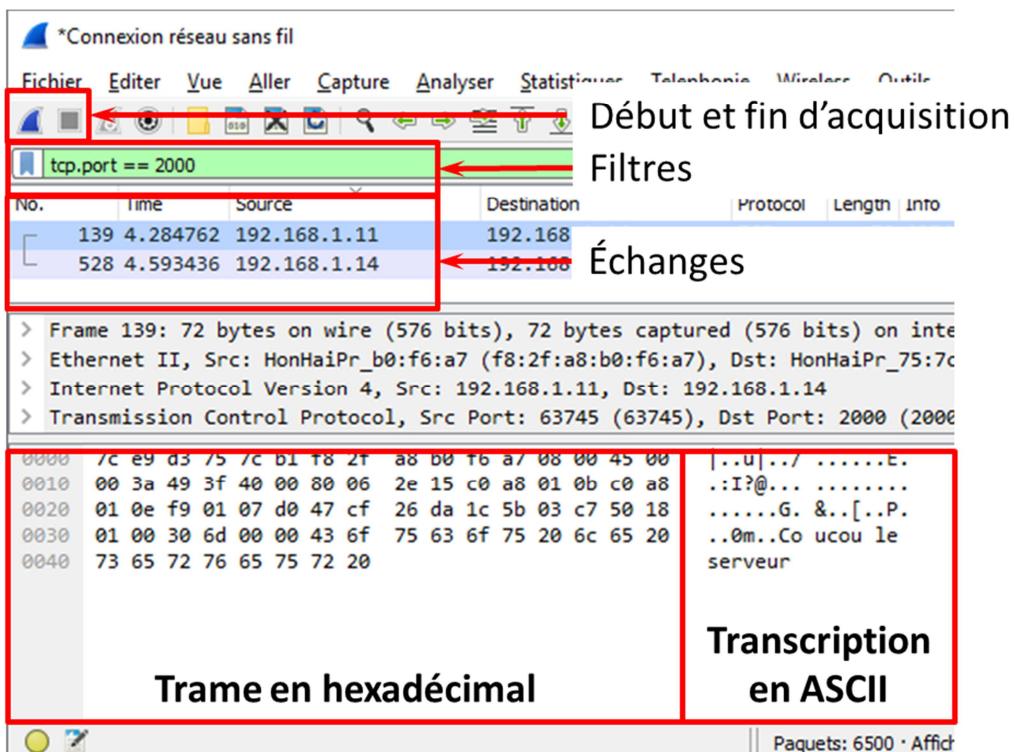
- Envoyer des messages du client au serveur et observer que le message a bien été transmis.
- Envoyer des messages du serveur au(x) client(s).
 - ⇒ Expliquer pourquoi le client ne peut envoyer de messages qu'au serveur ?
 - ⇒ Expliquer pourquoi le serveur peut envoyer un message à un seul client ou à tous les clients ?

3.2 Visualisation de la trame

Remarque :

Cette partie nous semble un peu difficile à appréhender au niveau du collège.

- Ouvrir Wireshark
- Sélectionner la connexion utilisée.
- Dans la zone de filtrage, saisir la ligne suivante pour ne visualiser que les paquets qui nous intéressent : `tcp.port == 2000`
- Démarrer la capture de paquets .
- Envoyer un message au serveur.
- Arrêter la capture de paquets .



Activité :

- En utilisant wireshark, repérez dans la trame codée en hexadécimal :
 - ⇒ Le code hexadécimal de l'adresse IP Source ;
 - ⇒ Le code hexadécimal de l'adresse IP Destination.
- En utilisant un codeur hexadécimal vers décimal :
 - ⇒ Vérifier que les adresses IP de la source et du destinataire codés dans la trame correspondent à ceux du client et du serveur.