

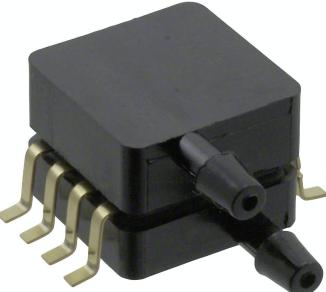
Documentation Souffl'utile



1. Hardware	1
2. Software	2
2.1. Arduino - Mesurer la pression du souffle	2
2.2. Snake - Utiliser la donnée détectée	2
2.2.1. Le jeu	2
2.2.2. Les Settings	4
2.2.3. La fin de jeu	6

1. Hardware

Liste des composants :

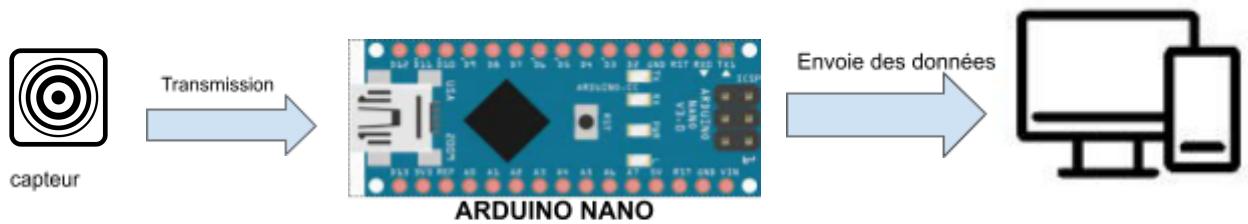
capteur MPXV7002DP :	Carte Arduino NANO	Smatree Bras de Support de Webcam
		

DSD TECH Module Bluetooth	piles Alcalines Type 9V	Boîtier
		

2. Software

La partie software est composée d'un code Arduino repris d'un ancien projet Open Source ainsi que d'un mini-jeu en Python 3.

2.1. Arduino - Mesurer la pression du souffle



Lorsque l'utilisateur souffle où aspire, le capteur de pression transmet l'information en continu au microcontrôleur. Plus le souffle est fort , plus la valeur transmise sera élevée et l'aspiration est forte, plus la valeur sera faible. Durant notre phase de test, nous obtenions une plage de valeur entre 0 et 1000.

2.2. Snake - Utiliser la donnée détectée

Nous avons décidé de faire un snake contrôlé par le souffle pour qu'un utilisateur puisse être autonome sur le jeu sans faire intervenir de tiers-personne.

Nous nous sommes basés sur un snake codé dans un [tutoriel](#) et qui utilise la bibliothèque python Pygame (v2.0.1). Pour faire communiquer l'Arduino à l'IDE Python, nous avons utilisé Pyserial (v3.5) qui permet de lire les données d'un certain port.

2.2.1. Le jeu

Le jeu est composé de deux fenêtres: le snake et les settings. La navigation entre chaque fenêtre est entièrement contrôlée par le souffle. Observons les contrôles du Snake.



Screen de la fenêtre **Snake**

Fenêtre Snake	
Expiration courte	Droite (<i>par défaut</i>)
Inspiration courte	Gauche (<i>par défaut</i>)
Expiration longue	Quitter
Inspiration longue	Settings



Screen de la fenêtre **Settings**

Fenêtre Settings	
Expiration courte	Modifier snake speed
Inspiration courte	Modifier contrôles du snake
Expiration longue	Quitter
Inspiration longue	Snake (quitter les Settings)



Screen de la fenêtre **Modifier snake speed**

Fenêtre Settings/Modifier snake speed	
Expiration courte	Settings (quitter la modification de la vitesse du snake)
Inspiration courte	<i>None</i>
Expiration longue	Augmenter la vitesse
Inspiration longue	Diminuer la vitesse



Screen de la fenêtre **Modifier les contrôles du snake**

Fenêtre Settings/Modifier les contrôles du snake	
Expiration courte	Définir comme Tourner à droite
Inspiration courte	Définir comme Tourner à droite



Screen de la fenêtre **Mort**

Fenêtre Settings/Modifier les contrôles du snake	
Expiration longue	Quitter
Inspiration longue	Continuer

2.2.2. Le code

Le code est composé de trois classes:

- Snake: le jeu
- Arduino: communiquer avec l'arduino et retourne l'action (souffle court etc)
- Button: créer un bouton

Arduino retourne une valeur (entre 0 et 1000). Pour le convertir en commande nous regardons les 3 dernières valeurs captées et fixons des seuils. C'est donc un tuple de trois dernières valeurs que l'on appellera T:

$$T = (530, 541, 540)$$

Si l'on n'applique aucune action sur la manette, les valeurs restent autour de 530.

Pour contrôler la sensibilité nous avons donc défini des seuils: 600 pour une expiration et 400 pour une aspiration. A partir de ces seuils nous avons définis des conditions pour les commandes:

Conditions	Exemple	Commande	valeur renvoyée par la classe Arduino
Les deux dernières valeurs de $T > 600$	$T = (529, 832, 911)$	Expiration longue	1
Les deux dernières valeurs de $T < 400$	$T = (499, 330, 11)$	Inspiration longue	2
$T[0] < 600 \ \& \ T[1] > 600 \ \& \ T[2] < 600$	$T = (531, 807, 550)$	Expiration courte	3
$T[0] > 400 \ \& \ T[1] < 400 \ \& \ T[2] > 400$	$T = (502, 23, 541)$	Aspiration courte	4

Dans le code, self.outputs correspond à toutes les valeurs de pressions captées depuis la dernière commande.