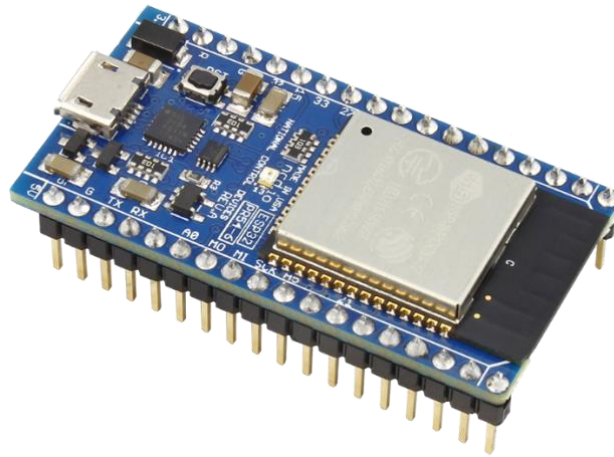


Composants électroniques

Table des matières

ESP32	2
Etat de l'art	2
Avantage de l'ESP32	3
Configurer l'ESP32 sur l'IDE Arduino	3
Glossaire	5
Ecran OLED I2C.....	6
Etat de l'art	6
Rôle dans le projet.....	6
Fonctionnement	7
Myoware.....	7
Etat de l'art	7
Rôle dans notre projet.....	8
Caractéristiques et utilisation.....	8
Signaux SIG et RAW	8
Alimentation et LED	8
Placement des électrodes en fonction de leurs couleurs (V11)	8
Electrodes	9
Etat de l'art	9
Rôle dans notre projet.....	9
Placement des électrodes.....	10
Volume de détection des électrodes et gel conducteur	11
Fonctionnement	11
Manette, Jack, Transistor	11
Etat de l'art	11
Rôle dans le projet.....	12
Fonctionnement	12
Bibliographie.....	14

ESP32



Etat de l'art

Dans ce projet nous avons besoin d'un microprocesseur pour rendre le projet viable et faisable. Lors de nos différents passés et de nos recherches, nous avons trouvé que le microprocesseur proposé par Arduino (ATMEGA328) avait trop peu de RAM et surtout une fréquence d'horloge trop basse. Notre choix c'est donc porté sur un ESP32 qui a une fréquence d'horloge plus élevée et plus de RAM. Dans notre projet, la vitesse de calcul et son traitement sont très important.

La taille des données à récupérer n'est pas très grande donc l'adressage n'est pas un problème. La mémoire flash nous importe peu étant donné que nous n'avons aucune sauvegarde de données après que le dispositif a été éteint.

Pour les entrées et les sorties nous avons besoins d'une entrée analogique (réception donnée des sensors) soit une par muscle, une sortie digitale ou analogique (envoi de données à la manette) et pour finir, une sortie et une entrée pour l'écran OLED.

Pour le prix : nous recherchons un produit réalisable pour le moins chère possible avec un objectif de 30/40€ de ventes, puis un ajout par muscle.

Comparatif ESP VS ATMEGA328 :

	ATMEGA328 (Arduino)	ESP32 (board MCU)
Adressage	8 bits	32 bits
Fréquence d'horloge	16 Mhz	240 Mhz
Mémoire Flash	32 kB	4 Mb
Mémoire RAM	2 kB	512 kB
E/S Numérique	14 (6 PWM)	19
E/S Analogiques	6	15
Taille	6,9 x 5,3 cm (Arduino Uno)	48 x 26 x 11,5 mm (NodeMCU)
Prix	16,25€ HT sur GoTronic	8€ seul ou 10€ HT sur board

Avantage de l'ESP32

Ce microcontrôleur permet de gérer des données de grandes tailles (adressage 32 bits) assez rapidement grâce à sa fréquence d'horloge nettement supérieure à celle d'Arduino (plus de 4 fois supérieure minimum). Sa mémoire flash est importante ici mais nous n'en n'aurons pas nécessairement l'utilité. Sa mémoire RAM est intéressante si l'on doit effectuer des filtres complexes, nous aurons peut-être besoin de stocker un certain nombre de variable. L'ESP32 possède ces avantages tout en gardant un rapport qualité / puissance / prix imbattable comparé aux autres microcontrôleurs disponibles sur le marché. Un autre point positif est au niveau de la taille, il est plus petit que la plupart des autres microcontrôleurs dans sa gamme de prix. Il est surtout beaucoup plus petit qu'une carte Arduino notamment.

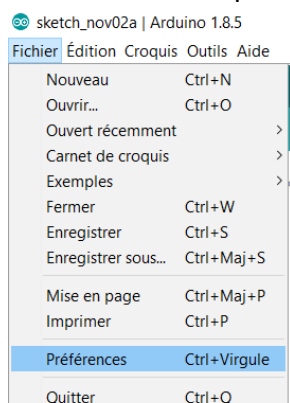
Enfin, et peut-être le point le plus important, le microcontrôleur est compatible avec l'IDE Arduino. Dans notre projet, le microcontrôleur permet de filtrer le signal des EMG sensors, envoyer des informations à l'écran et envoyé des signaux à la manette adaptative Xbox. C'est le cerveau et la pièce maitresse du projet !

Configurer l'ESP32 sur l'IDE Arduino

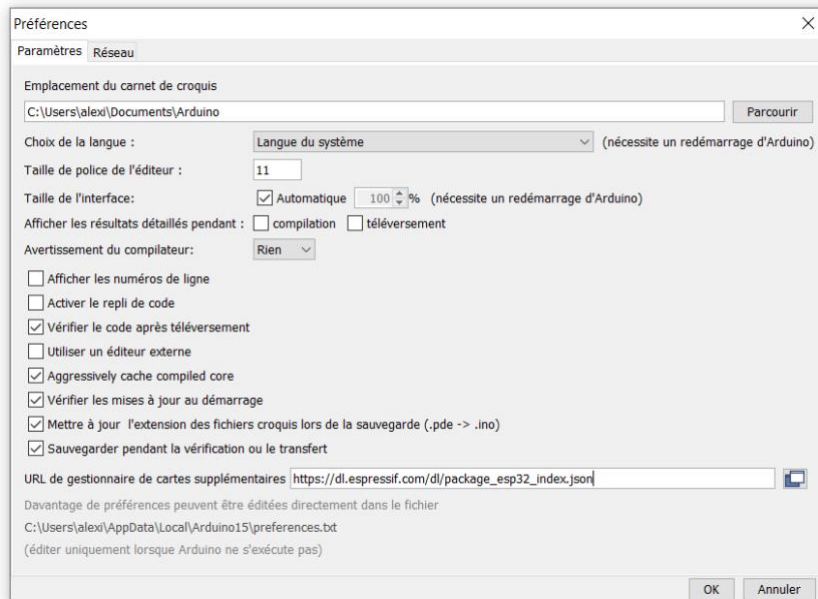


Dans un premier temps il faut télécharger la « bibliothèque » de l'ESP32 comme suit :

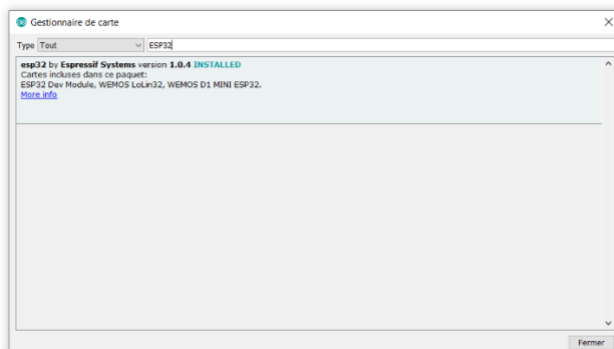
- Aller dans fichier puis sélectionner préférences



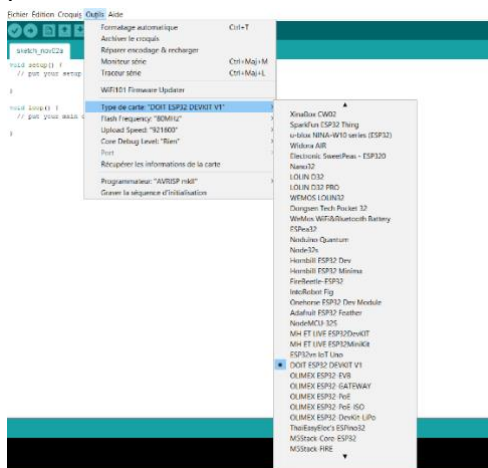
- Coller l'adresse https://dl.espressif.com/dl/package_esp32_index.json dans la barre « URL de gestionnaire de cartes supplémentaires » puis cliquer sur ok



- Une fois ceci fait, aller dans Outil – Type de carte – Gestionnaire de carte et entrer « ESP32 » dans la barre de recherche

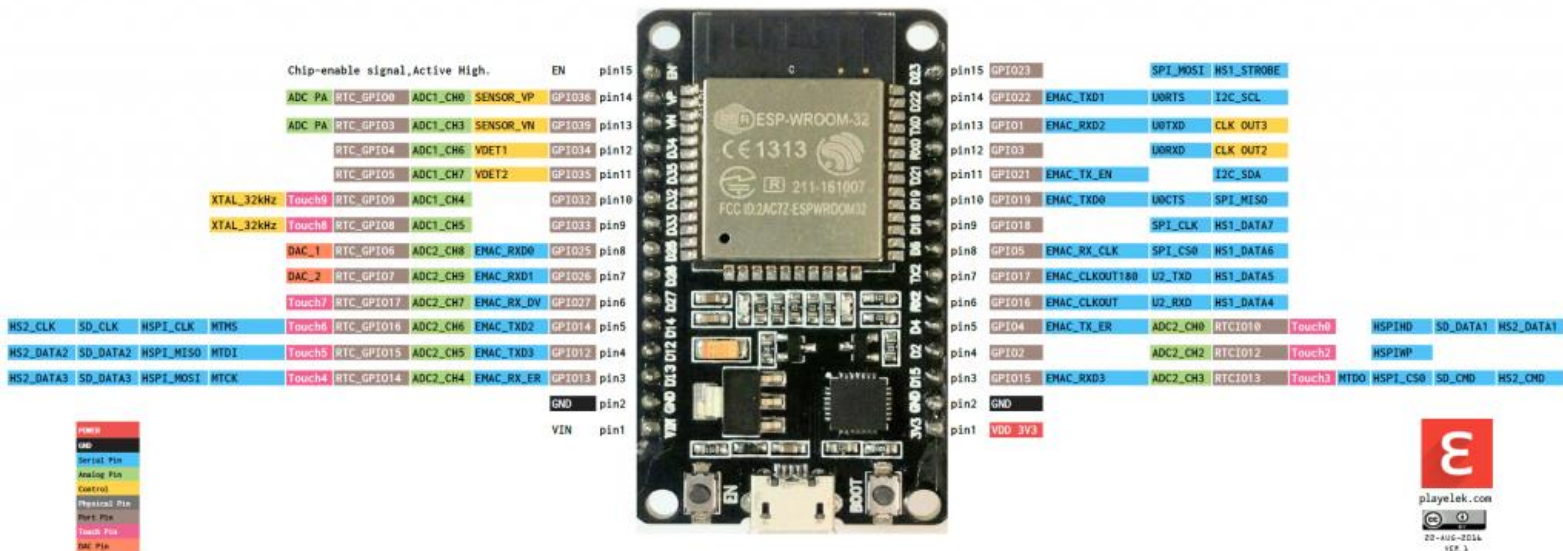


- Cliquer sur « Installer » et attendre quelques instants
- Sélectionner ensuite « DOIVT ESP32 DVKIT V1 » dans la liste des cartes installés pour pouvoir utiliser l'ESP32



- Le langage pour programmer l'ESP32 est le même que celui pour un Arduino Uno. Attention cependant à bien nommer les pins utilisés dans votre code.

DOIT ESP32 DEVKIT V1 PINOUT



Ci-dessus un schéma récapitulant les noms des pins de l'ESP32

Glossaire

Fréquence d'horloge : vitesse de calcul et transfert des données, caractérise la puissance de traitement ou le débit d'information (nombre d'instructions réalisable en 1 seconde)

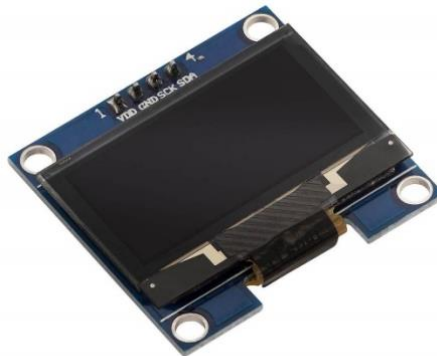
Mémoire RAM : sert à stocker des données temporaires (variables par exemple). Si cette mémoire vient à être dépassé, l'application peut crash sans plus d'informations. Ce crash sera de plus relativement aléatoire et risque de se traduire par un comportement complètement incohérent et anarchique du programme.

Mémoire flash : capacité à stocker des données après mise hors tension de l'appareil

Entrées/Sorties PWM : peuvent être utilisées en tant qu'entrée analogique mais seulement pour des données arrivant lentement car il faudrait mettre en place un filtre donc pas idéal dans notre cas.

Adressage : permet de savoir la taille des données que le microcontrôleur peut traiter. Plus il est élevé, plus on pourra traiter de données de grande taille (influe sur la RAM et la capacité de stockage)

Ecran OLED I2C



Etat de l'art

Nous avons choisi l'écran OLED 1.3 pouces I2C car il répondait aux nombreux besoins du projet. Par exemple :

- La possibilité de l'utiliser sur une carte de type ESP32 aussi bien sur le plan du software (en utilisant les bibliothèques et l'IDE Arduino) que sur le plan du hardware (surtout côté alimentation).
- Sa taille (3.7 cm) et sa lisibilité, mais aussi sa finesse (seulement 2.7 cm)
- Sa fiabilité
- Son prix très correct (<10€ pièce)
- Son branchement hyper simple et sa facilité d'utilisation, notamment avec la bibliothèque U8g2lib.h qui offre un grand éventail de possibilités.

L'avantage d'un écran OLED (*Organic Light-Emitting Diode*) par rapport à un écran LCD (*Liquid Crystal Display*) est dans un premier temps que la technologie OLED permet un rétroéclairage de la dalle qui permet une meilleure lisibilité et une dalle plus fine, dans un second temps l'écran en lui-même est plus petit et donc plus adapté dans un objectif de miniaturisation et de simplicité, enfin, le câblage est beaucoup plus complexe pour un écran LED, avec entre 10 et 16 branchements, ce qui impossible avec l'addition du myoware sur l'ESP32.

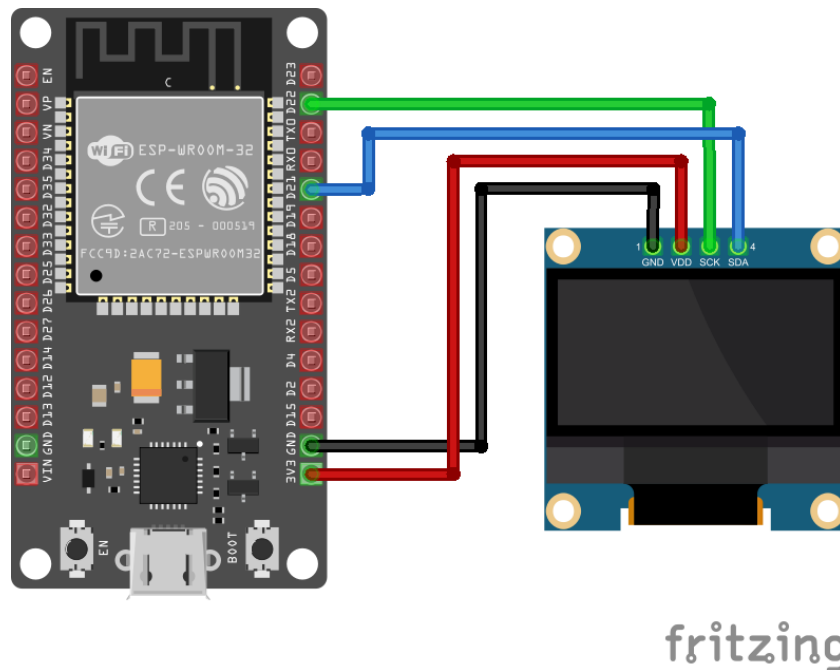
Rôle dans le projet

Le choix d'un écran OLED pour le système Play to Heal permet essentiellement de faciliter et d'améliorer l'expérience utilisateur. En effet, sur ce dernier pourront être affichées les instructions et étapes de la mise en place du système, assurant son bon fonctionnement. Ainsi le personnel de santé ou accompagnant des personnes en situation de handicap gagneront du temps et la personne utilisant la « manette » Play to Heal sera averti en temps réel de son évolution.

Elle permet aussi pour nous de faire des tests directement avec tout le système sans avoir à brancher le système à un ordinateur (qui ne servira, à terme, qu'à afficher le jeu ou le logiciel de rééducation).

Fonctionnement

Cet écran OLED a seulement besoin de 2 câbles d'alimentation (3.3V), et 2 câbles permettant l'affichage et le stockage d'informations. A Noter que l'écran fonctionne avec beaucoup de bibliothèques différentes. Nous avons décidé d'utiliser la bibliothèque U8g2lib.h pour sa simplicité d'utilisation, sa rapidité et sa lisibilité dans le code.



Myoware

Etat de l'art

Lorsque que nous avons repris le projet, l'équipe précédente utilisait des EMG Sensors de la marque Seed. Ces composants avaient deux inconvénients majeurs : l'imprécision des mesures et l'impossibilité d'obtenir un signal en fréquentiel. Afin de pallier ce problème nous avons effectué une recherche d'autres types de capteurs EMG permettant d'obtenir à la fois un signal de sortie présentant des variations en amplitude, et un autre signal avec des variations en fréquence. C'est à la suite de ces recherches que nous avons trouvé le Myoware, qui présentait ces caractéristiques recherchées et offrait la possibilité de l'alimenter en 3.3V ce qui était également un facteur important (tout notre système étant alimenté avec ce voltage).

Il est également à noter que là où l'EMG Sensor nécessitait l'usage d'électrodes distinctes devant être branchées, le Myoware peut théoriquement être placé directement sur la peau en utilisant ses électrodes intégrées. Nous ne nous en sommes néanmoins pas servis car pouvoir éviter quelques câbles ne nous a pas paru un gain suffisant par rapport au fait de devoir supporter le poids du Myoware et de son alimentation.

Rôle dans notre projet

Le Myoware (et le Cbale Shield qui y est relié) sont des composants clés de notre dispositif. En effet c'est le Myoware qui procède au traitement des données reçues par les électrodes et qui renvoie vers notre ESP32 un signal relatif à la contraction (ou non) du muscle visé.

Le Myoware peut renvoyer deux types de signaux, le premier présentant une variation de son amplitude en cas de contraction (pin SIG) et le second une variation en fréquence (pin RAW). Nous pouvons utiliser n'importe lequel de ces deux signaux, le SIG étant plus facile à traiter mais le RAW étant plus fiable.

Enfin le Myoware est dans notre dispositif soudé à un Cable Shield. C'est ce dernier qui nous permet d'utiliser des électrodes différentes de celles placées par défaut sur le Myoware. Le Cable Shield est très simple en lui-même, il ne sert que d'extension du Myoware avec une prise jack femelle dédiée aux électrodes.

Caractéristiques et utilisation

Signaux SIG et RAW

Le signal reçu depuis la pin SIG est plus facile à traiter car il suffit pour cela de définir un seuil dynamique définissant la présence ou l'absence de contraction à partir des valeurs moyenne du signal à l'état contracté et décontracté. Ce seuil est défini durant la procédure d'initialisation de l'appareil, et toute valeur du signal le dépassant sera considérée comme correspondant à une contraction du muscle.

Pour ce qui est du traitement du signal RAW, nous avons mis en place un filtre en cascade afin d'éliminer à la fois le bruit causé par des parasites ainsi que celui d'un muscle standard au repos. Cela signifie que si nous parvenons à observer un signal à la sortie de ce filtre en cascade, alors le muscle visé est contracté.

Alimentation et LED

Le Myoware demande une alimentation standard en 3.3V et une LED verte s'allume lorsque celui-ci est correctement alimenté. Une autre LED rouge s'allume quand le Myoware considère avoir détecté une contraction. Il est toutefois à noter que l'allumage de cette LED est hors de notre contrôle et n'est en aucune façon liée à nos définitions de seuil de contraction. Son allumage n'est donc pas nécessairement une preuve que notre dispositif considère la contraction comme valide.

Placement des électrodes en fonction de leurs couleurs (V11)

Pinout Relative to the Bottom View	v10 [DEV-13687]	v11 [DEV-14109]
Reference [R]	Blue	Black
End [E]	Red	Blue
Middle [M]	Black	Red

Electrodes



Etat de l'art

Au moment où notre équipe a repris le projet, le prototype d'alors utilisait des électrodes Sseed allant de paire avec l'EMG Sensor Sseed alors utilisé que nous avons plus tard remplacé par un Myoware (cf. document Myoware pour plus d'informations à ce sujet).

Il est rapidement apparu dans nos recherches que les électrodes en elles-mêmes ne semblaient pas être un élément déterminant dans la qualité de la réception du signal. En effet nous n'avons trouvé nulle part trace de document indiquant un fonctionnement particulier d'un type d'électrodes EMG par rapport à un autre.

Néanmoins, le placement des électrodes, le volume de détection des électrodes et la présence / qualité d'un gel conducteur entre l'électrode et la surface de la peau se sont révélés être des éléments déterminants dans la réception d'un signal musculaire.

Rôle dans notre projet

Afin de mesurer l'activité musculaire du sujet, nous avons besoin d'électrodes capables de mesurer une différence de potentiel électrique entre deux points d'un système biologique.

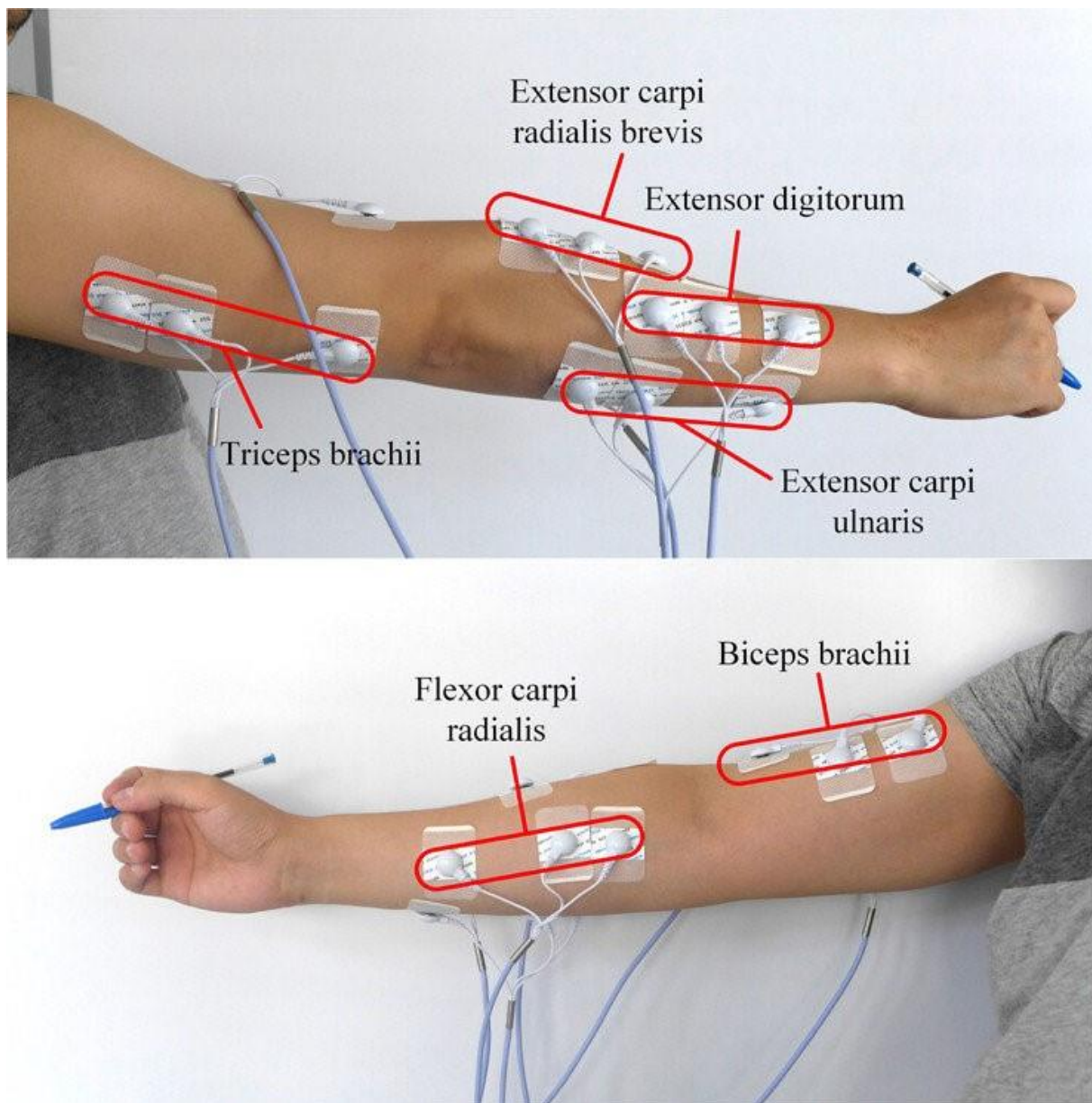
Ces électrodes sont placées sur un muscle que le sujet parvient à manipuler correctement, ou au contraire sur un muscle sur lequel il souhaite s'exercer. Une fois les électrodes branchées au reste du système, elles nous permettent de détecter une contraction et de relayer l'information jusqu'à notre dispositif résultant ainsi en une activation du bouton préalablement configuré de la manette adaptative XAC.

Il est à noter que le placement des dites électrodes peut parfois se révéler ardu. En effet une mauvaise disposition de ces dernières peut rendre l'appareil inopérant et il suffit d'une erreur de quelques centimètres pour diminuer gravement la précision de celui-ci.

Pour cette raison, nous indiquerons dans la dernière partie de ce document quelques exemples de placement corrects d'électrodes.

Placement des électrodes

Les jeux d'électrodes EMG sont composés de 3 électrodes à placer sur le muscle, chacune ayant un rôle particulier. L'une doit être placée au milieu du muscle que l'on veut mesurer et la suivante dans le prolongement de ce muscle, à la fin de ce dernier. C'est la différence de potentiel entre ces deux points qui va permettre à l'EMG Sensor auquel ces électrodes sont reliées de mesurer l'occurrence d'une contraction. On place enfin une troisième électrode sur une surface osseuse ou musculaire mais non-adjacente. Cette dernière électrode va servir de référence afin de diminuer l'impact du bruit sur la qualité de la réception. Il est à noter qu'un placement relativement permissif peut être fait sur le biceps, en utilisant le triceps comme référence.



Volume de détection des électrodes et gel conducteur

Au bout de chaque électrode d'un jeu, il est nécessaire d'attacher une « tête » adhésive permettant la réception du signal électrique. La taille de cette tête et sa lubrification sont des éléments clés d'une bonne réception de signal. Une tête plus petite permettra un placement plus précis sur des plaques motrices de taille réduite, permettant ainsi d'augmenter le nombre de muscles avec lesquels le dispositif peut être utilisé. En contrepartie, il est alors nécessaire de savoir exactement où placer les électrodes sous peine d'être incapable d'utiliser l'appareil. Mais même une fois correctement placées, les électrodes peuvent encore rencontrer une dernière difficulté : l'impédance à la surface.

Le signal électrique généré par une contraction musculaire est de faible amplitude (10 mV). Les électrodes doivent donc avoir une résistance minimale afin de détecter ce signal. Pour les cas les plus délicats, il est possible de réduire l'impédance à la surface de la peau en abrasant la surface de celle-ci et en la nettoyant à l'aide d'alcool à friction. On peut également appliquer un gel conducteur spécial, permettant ainsi d'augmenter encore la qualité de la réception par les électrodes

Fonctionnement

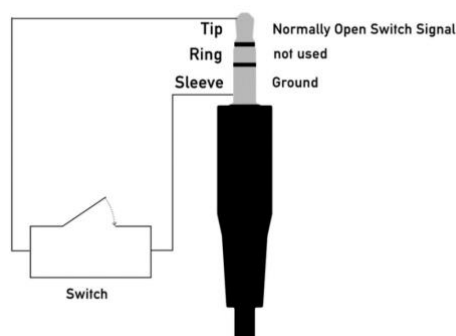
Le signal EMG résulte de la superposition de l'ensemble des potentiels d'action des unités motrices situées dans le volume de détection des électrodes, il permet ainsi de détecter des contractions musculaires. En effet le corps humain est généralement un bon conducteur électrique. Ainsi, les signaux électriques issus d'une contraction musculaire ont tendance à se propager jusqu'à la surface de la peau ce qui permet à des électrodes de détecter ces contractions.

Il est à noter que des parasites peuvent être détectés lors de mesures avec des électrodes. Ils peuvent être dus à un mauvais placement des électrodes, à la présence d'un corps étranger entre les électrodes et la peau ou à un mouvement dans les câbles des électrodes. Pour réduire ce bruit, différentes méthodes ont été mises en place, expliquées dans le document sur le Myoware.

Manette, Jack, Transistor

Etat de l'art

La connexion en digital (pas analogique) à la manette se fait par le biais de câbles JACK 3.5 (stéréo standard). Afin de cliquer un bouton, il faut que le tip et le sleeve du jack branché à ce bouton soient connectés. Exemple ci-dessous :



Le switch permet de cliquer ou non le bouton associé. Ce switch sera réalisé par un transistor, c'est un composant qui permet principalement de contrôler les signaux électriques en les amplifiant ou en servant d'interrupteur. Il peut aussi servir de redresseur de signal ou permettre de stabiliser la tension voire la détecter.

Dans le cadre de notre projet, nous avons choisi le transistor NPN (PN2222) pour les raisons suivantes :

- De petite taille
- Peut fonctionner en basse tension
- Très bon marché
- Faible impédance de sortie
- Longue durée
- Actions spontanées

Le PNP a les mêmes avantages mais est plus lent.

Rôle dans le projet

Dans le cadre de notre projet, nous allons utiliser des transistors pour effectuer le switch qui permet de cliquer le bouton ou non. Nous utilisons maintenant un NPN Transistor (le PN2222). Afin d'éviter de devoir couper le bout du jack male que nous voulons connecter au transistor, nous utilisons une prise jack femelle.

Fonctionnement

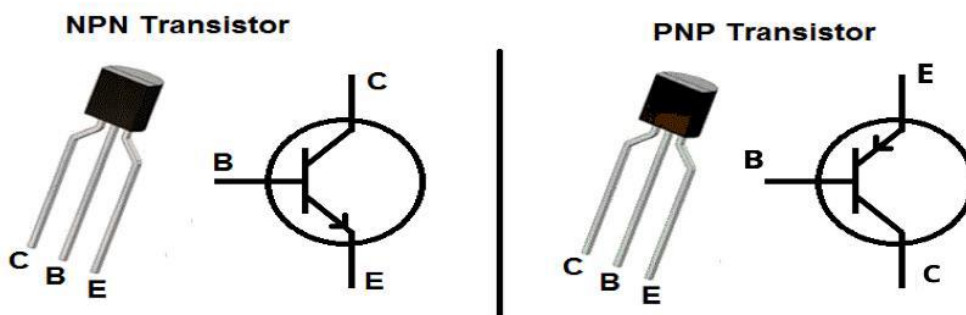
Transistor bipolaire (BJT)

Composé de 3 couches de semi-conducteurs dopées positivement et négativement. La couche P possède un manque d'e- et N un excès d'e-.

Deux types de BJT possibles NPN et PNP comprenant :

- Une base (B) : connectée à un potentiel négatif pour le NPN (positif PNP)
- Un émetteur (E)
- Un collecteur (C) : connecté à un potentiel positif pour le NPN (négatif PNP)

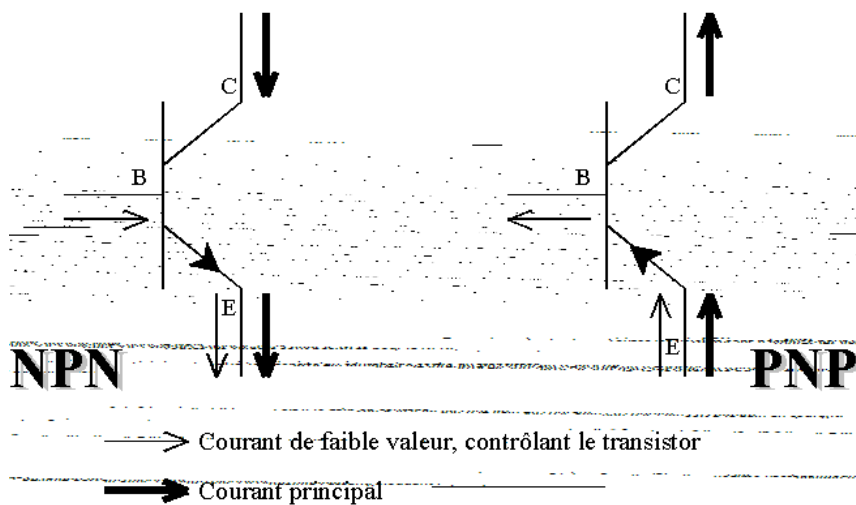
En sortie E le sens du courant est différent. Positif en NPN et négatif en PNP.



Régimes de fonctionnement :

1. Passant
 - Le courant B permet de laisser passer plus ou moins de courant entre C et E.
 - Une amplification du signal électrique est possible.
 - Entre C et E c'est un fil.
2. Régime saturé
 - Laisse passer la totalité du courant d'alimentation.
3. Bloquant
 - Aucun courant ne parcourt le transistor, l'interrupteur est ouvert

Le transistor permet donc de servir d'interrupteur commandé avec une faible énergie de commande.



Fiche technique du transistor PN2222 :

Electrical Characteristics $T_a=25^{\circ}\text{C}$ unless otherwise noted

Symbol	Parameter	Test Condition	Min.	Max.	Units
BV_{CBO}	Collector-Base Breakdown Voltage	$I_C=10\mu\text{A}, I_E=0$	60		V
BV_{CEO}	Collector Emitter Breakdown Voltage	$I_C=10\text{mA}, I_B=0$	30		V
BV_{EBO}	Emitter-Base Breakdown Voltage	$I_E=10\mu\text{A}, I_C=0$	5		V
I_{CBO}	Collector Cut-off Current	$V_{CB}=50\text{V}, I_E=0$		0.01	μA
I_{EBO}	Emitter Cut-off Current	$V_{EB}=3\text{V}, I_C=0$		10	nA
h_{FE}	DC Current Gain	$V_{CE}=10\text{V}, I_C=0.1\text{mA}$ $V_{CE}=10\text{V}, I_C=150\text{mA}$	35 100	300	
$V_{CE}(\text{sat})$	* Collector-Emitter Saturation Voltage	$I_C=500\text{mA}, I_B=50\text{mA}$		1	V
$V_{BE}(\text{sat})$	* Base-Emitter Saturation Voltage	$I_C=500\text{mA}, I_B=50\text{mA}$		2	V
f_T	Current Gain Bandwidth Product	$V_{CE}=20\text{V}, I_C=20\text{mA}, f=100\text{MHz}$	300		MHz
C_{ob}	Output Capacitance	$V_{CB}=10\text{V}, I_E=0, f=1\text{MHz}$		8	pF

* Pulse Test: Pulse Widths $\leq 300\mu\text{s}$, Duty Cycles $\leq 2\%$

Pour le transistor PN2222, la tension max est de 40 V et la tension de seuil est

Bibliographie

Manette adaptative : <https://www.xbox.com/fr-FR/accessories/controllers/xbox-adaptive-controller>

Transistor : https://www.amazon.fr/Adafruit-NPN-Bipolar-Transistors-PN2222/dp/B00XW2OK7M/ref=sr_1_3?dchild=1&hvadid=79852064419950&hvbmt=be&hvdev=c&hvqmt=e&keywords=pn2222&qid=1614073737&sr=8-3&tag=hydfirmsn-21

Jack male: <https://www.gotronic.fr/art-cordon-jack-ca35s05-15030.htm>

Electrodes: <https://www.gotronic.fr/art-cordon-cab-12970-26243.htm>

ESP32: <https://www.gotronic.fr/art-module-nodemcu-esp32-28407.htm>

OLED: https://www.gotronic.fr/art-afficheur-oled-1-3-i2c-tf051-28512.htm#compte_desc