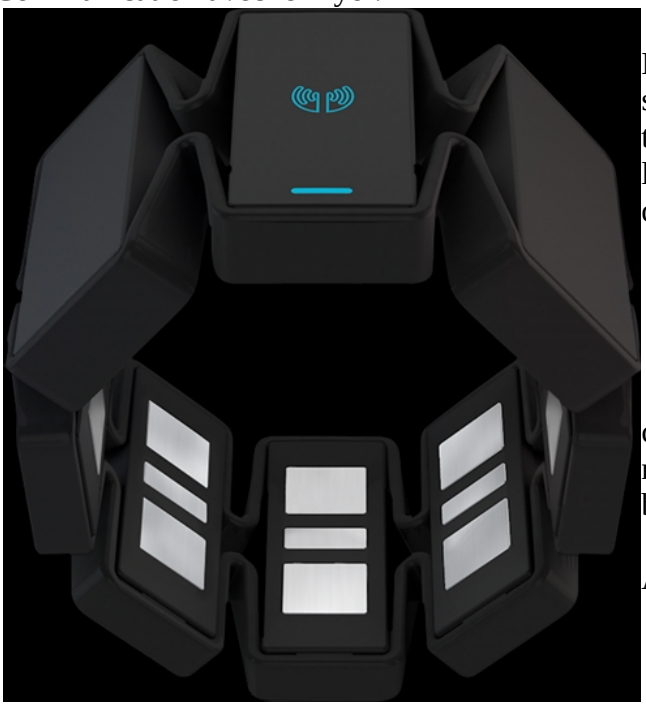
Communication
SSC-32U/Kinect & Myo

Documentation du
Projet Robot Arm

Communication avec le Myo :



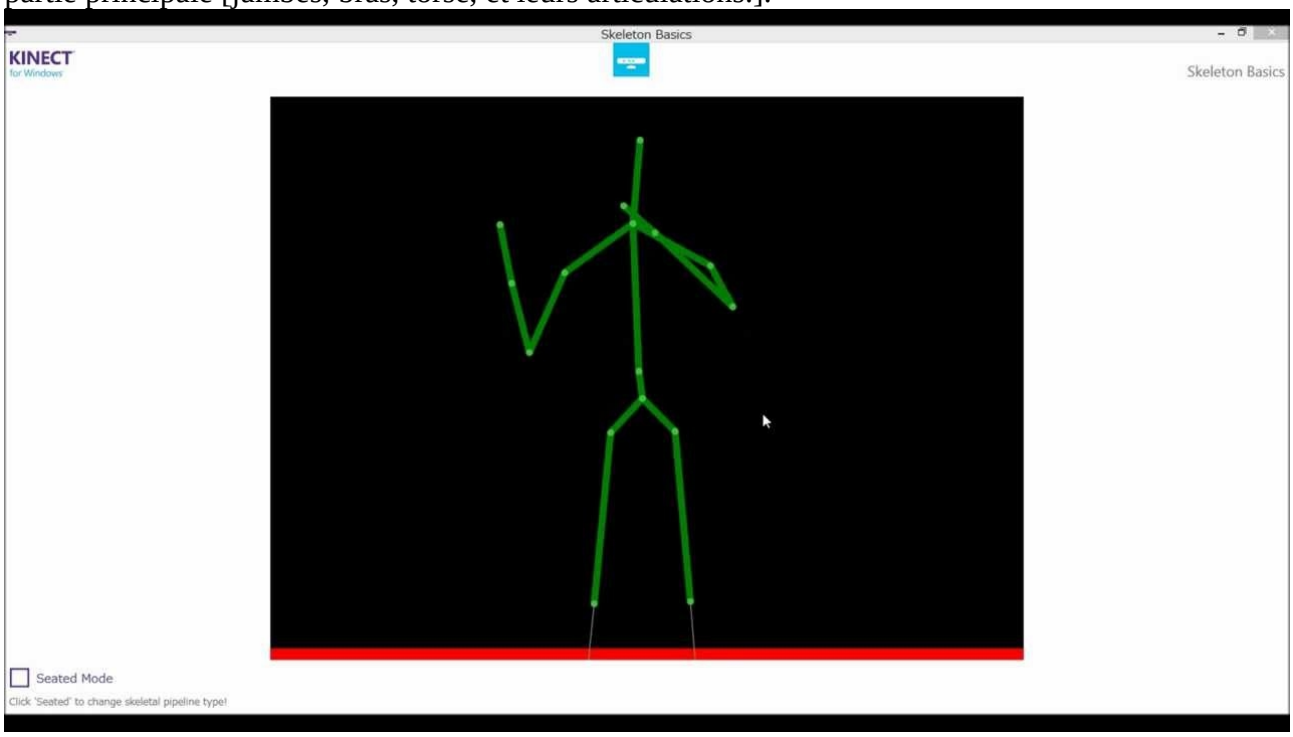
La communication avec le Myo et le robot a simplement été réalisée grâce au mappage des touches, possible à l’aide du logiciel fournit sur le site web grâce au code d’activation se trouvant dans le boîtier du Myo.

En effet il a suffit de se servir dumotoréducteur commandant les pinces comme acteur pour le mappage afin de faire bouger la pince à l’aide du bracelet Myo.

Aucun ajout de ligne de code n’a été nécessaire.

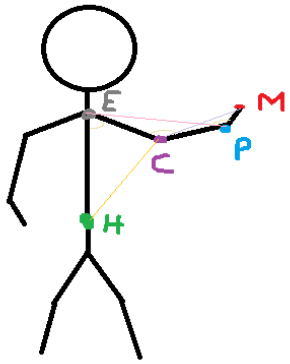
Communication entre Kinect et SSC-32U :

- Skeleton WPF
Il s’agit d’un logiciel créer par Microsoft et en open source permettant à la Kinect de détecter et représenter un corps humain au travers de la Kinect sous la forme d’un squelette représentant les partie principale [jambes, bras, torse, et leurs articulations.].



Le code a été importé afin de pouvoir être modifié comme présenté dans la section suivante.

- Calcul des angles



$$\cos \widehat{CPM} = \frac{\sqrt{(yp-yc)^2 + (xp-xc)^2} + \sqrt{(yp-ym)^2 + (xp-xm)^2} - \sqrt{(ym-yc)^2 + (xm-xc)^2}}{2 * \sqrt{(yp-yc)^2 + (xp-xc)^2} * \sqrt{(yp-ym)^2 + (xp-xm)^2}}$$

$$\cos \widehat{ECP} = \frac{\sqrt{(yp-yc)^2 + (xp-xc)^2} - \sqrt{(yp-ye)^2 + (xp-xe)^2} + \sqrt{(ye-yc)^2 + (xe-xc)^2}}{2 * \sqrt{(yp-yc)^2 + (xp-xc)^2} * \sqrt{(ye-yc)^2 + (xe-xc)^2}}$$

$$\cos \widehat{HEC} = \frac{-\sqrt{(yh-yc)^2 + (xh-xc)^2} + \sqrt{(yc-ye)^2 + (xc-xe)^2} + \sqrt{(yh-yc)^2 + (xh-xc)^2}}{2 * \sqrt{(yc-ye)^2 + (xc-xe)^2} + \sqrt{(yh-yc)^2 + (xh-xc)^2}}$$

Tout d'abord, avant de se lancer dans le calcul des angles, il a fallut supprimer la représentation des parties qui ne nous intéressaient pas pour le projet.
Comme les jambes, et le bras gauche.

Une fois ces dernières retirées, nous avons du calculer le cosinus des trois angles nous intéressant : Poignet, Coude et Épaule, respectivement CPM, ECP et HEC.

M pour Main et H pour Hanche.

Une partie du problème fut de transcrire l'équation en C# pour l'insérer dans le code.
Du au résultat que l'on devait obtenir, les variable pour les angles sont de types « double » afin d'être suffisamment spacieuse pour afficher les résultats.

On se retrouve donc pour COS CPM avec deux variables Poignet/Coude et Poignet/Main afin de simplifier l'équation, donnant les deux lignes de codes suivantes.

```
double longueurPoignetCoude = Math.Sqrt(Math.Pow(PoignetD.Position.Y - CoudeD.Position.Y, 2) + Math.Pow(PoignetD.Position.X - CoudeD.Position.X, 2));
```

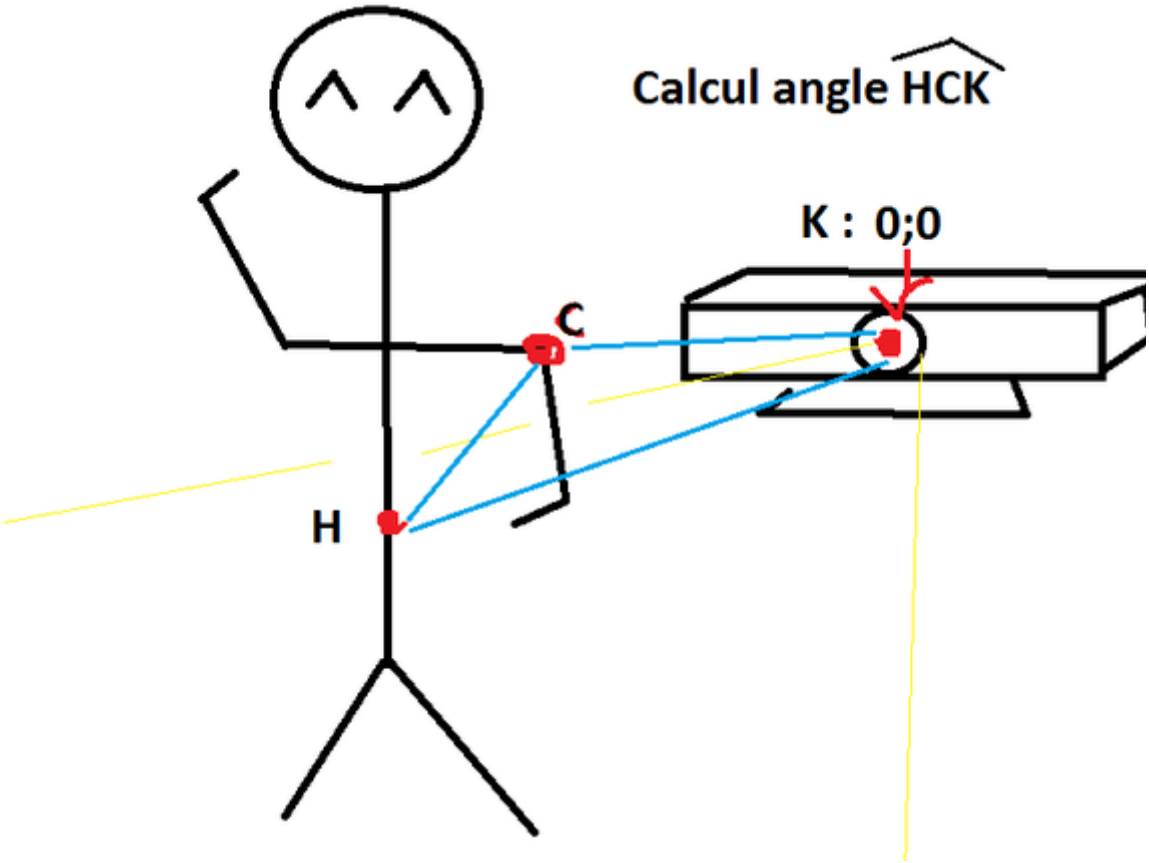
```
double longueurPoignetMain = Math.Sqrt(Math.Pow(PoignetD.Position.Y - MainD.Position.Y, 2) + Math.Pow(PoignetD.Position.X - MainD.Position.X, 2));
```

Ces deux lignes serviront ensuite à être imbriquées dans l 'équation globale :

anglePoignet = Math.Acos((Math.Pow(longueurPoignetCoude, 2) + Math.Pow(longueurPoignetMain, 2) - Math.Pow(longueurMainCoude, 2)) / (2 * longueurPoignetCoude * longueurPoignetMain)) * 180 / Math.PI;

Il a ensuite fallut faire de même pour les autres angles.

Du moins, cela concerne uniquement la variable Y, autrement dit, notre déplacement latéral par rapport à la Kinect.
Il a ensuite fallut réaliser le déplacement en Z, soit, la profondeur.



Comme montrer sur le dessin, cet angle HCK se calcul entre la Hanche, le Coude et la Kinect, et permet de déterminer la profondeur, essentiel pour une parfaite représentation des mouvements.

Il a donc fallut créer trois nouvelles variable double : longueurHancheCoudeZ, longueurHancheKinect et longueurCoudeKinect .

Ces variables ont été calculée de la même façon que les autres et, comme les autres, ont pour but de réduire la consistance de l'équation finale pour éviter tout risque d'erreur.

L'angle HCK qui permet la détection en Z a ensuite été créer comme les autre angles en Y, c'est à dire que les variables ont été intégrée par la suite dans l 'équation globale, donnant comme résultat :

angleHanche = Math.Acos((Math.Pow(longueurHancheKinect, 2) - Math.Pow(longueurCoudeKinect, 2) + Math.Pow(longueurHancheCoudeZ, 2)) / (2 * longueurHancheKinect * longueurHancheCoudeZ)) * 180 / Math.PI;

Par la suite, il a simplement fallut intégrer, sans modification ni ajustement, le code de communication.