

<b>PROJET :</b>	<b>Robotique</b>	<b>Année :</b>	2016-2017
		<b>Semestre :</b>	S2
<b>NOM1 :</b>	FEKKAR	<b>Groupe TP :</b>	C2
<b>NOM2 :</b>	JEYAKUMAR		

## 1 *Présentation du projet*

Le bras robot MeArm est un dispositif mécanique, électrique et informatique accomplissant automatiquement des mouvements.

Dans ce travail de projet, le but est de rendre fonctionnel un bras robot de type MeArm sans sa pince. Pour cela, nous devons adapter le design du robot car on nous impose d'utiliser les servomoteurs disponibles.

Cela pose certaines problématiques. En effet, en adaptant le robot avec les servomoteurs disponibles, nous devons changer, agrandir et rétrécir certaines pièces ainsi que modifier leurs positionnements. De plus, il faut réfléchir aux conséquences que cela implique sur les servomoteurs, la structure et les mouvements du bras.

En plus de cela, nous allons faire de la programmation à l'aide d'Arduino Uno, du câblage électrique et nous allons utiliser certains logiciels libres tel qu'Inkscape.

Nous avons donc travaillé sur ces axes durant le projet :

- Planification de celui-ci afin de répondre aux caractéristiques du cahier des charges
- Organisation des tâches à effectuer
- Etude des spécifications du bras robot MeArm original
- Etude des spécifications du bras robot que l'on nous demande de réaliser
- Modification du design du bras robot pour le rendre compatible avec les moteurs disponibles
- Programmation du bras à l'aide d'Arduino afin de pouvoir manœuvrer le robot à l'aide d'un joystick ou d'un slider
- Réalisation du prototype de la partie mécanique
- Câblage de la partie électrique

## 2 *CDC TECHNIQUE FINAL*

N°	Nom Fonction	Critères	Valeur limite	Pourcentage d'avancement
F1	Positionner le bras robot en obéissant à une consigne	Erreur angulaire Vitesse	<10 deg >0,5cm/s	75%
F2	Garantir la sécurité de l'utilisateur et du matériel	Normes électriques basse tension.	Voir normes	50%
F3	Maintenir la stabilité du bras	Base à l'horizontale	<10 deg	100%

	robot			
F4	Positionner le bras robot avec des servomoteurs à nombre limités	Utiliser les servomoteurs : <ul style="list-style-type: none"> <li>• HS-322HD</li> <li>• MG90S</li> </ul>	Maximum 1 MG90S par groupe	100%
F5	Modifier le design du bras robot afin de l'adapter au matériel disponible	Structure solide		100%
F6	Permettre la programmation du bras	Pilotage par Arduino Uno		100%

### 3 PROFIL DE SATISFACTION FINALE

Fonction	Libellé	Critère	Score (++,+,-,-,-)	Commentaire
F1	Déplacer	Erreur angulaire (<10 deg) Vitesse(>0,5cm/s)	+	Nous n'avons pas trouvé de moyen précis afin de déterminer avec certitude l'erreur angulaire ainsi que la vitesse de déplacement du bras robot. Cependant, à vue d'œil, on peut clairement voir que la vitesse est >0,5cm/s et que le positionnement du robot est précis.
F2	Sécurité	Normes électriques basse tension : $50V < U_n \leq 1000V$	+	La carte ESPLORA d'Arduino fonctionne en 5V. Nous sommes donc dans le domaine TBTS. Nos branchements se font à partir de cette carte. Cependant, certains de nos câbles sont parfois dénudés, mais on ne peut pas se blesser car nous sommes en TBTS
F3	Stabilité	Horizontal : <10 deg	+	La base du robot est très stable et paraît horizontale.

				Cependant, nous n'avons aucun moyen de vérifier si la base est horizontale à 100%
F4	Positionner	Utiliser les servomoteurs : <ul style="list-style-type: none"> <li>• HS-322HD</li> <li>• MG90S</li> </ul>	++	Le positionnement du bras robot se fait avec : 1 MG90S 2 HS-322HD
F5	Structure	Solide	++	Robot en bois
F6	Programmer	Pilotage par Arduino Uno	++	Utilisation du slider pour effectuer les mouvements

### Estimez-vous avoir rempli le contrat ?

Nous estimons avoir rempli le contrat. En effet, nous avons réussi à rendre opérationnel le bras robot MeArm avec le matériel imposé. Concernant le cahier des charges, nous avons tenté de le respecter au maximum. En effet, nous avons réussi à positionner le bras précisément à l'aide de la carte Arduino ESPLORA et d'un programme. Cependant nous n'avons pas trouvé le moyen de calculer la vitesse du robot ni le moyen de calculer l'erreur angulaire. Mais le robot nous paraît précis et se déplace à plus de 0.5 cm/s à vue d'œil. Concernant la sécurité, nous utilisons une carte Arduino ESPLORA alimentée en 5V, ce qui correspond au domaine TBTS ce qui respecte les normes électriques BT.

### Quelles sont les difficultés rencontrées et les solutions originales apportées ?

Une des principales difficultés était de maîtriser le logiciel Inkscape. En effet, tant que l'on ne maîtrisait pas ce logiciel, on ne pouvait pas modifier correctement le bras robot.

Une fois le logiciel maîtrisé, nous avons rencontré des problèmes au niveau des premiers prototypes. En effet, les premiers prototypes ne comportaient pas assez de modifications. De plus, les prototypes étaient en carton (carton de calendrier). Donc souvent, les pièces soumises à des forces tel que la force exercée par la palette du servomoteur étaient trop fortes pour le carton et donc il cassait.

Lorsque le prototype cassait, nous devions retourner au Fablab, sachant que celui-ci n'ouvrait que les jeudis, ce qui était très problématique.

Une fois que nous avons modifié des pièces à l'aide d'Inkscape, d'autres pièces non modifiées nous posaient des problèmes et donc nous devions modifier aussi celles-ci. Nous avons donc au total modifié une quinzaine de pièces.

La partie programmation qui paraît simple nous a pris beaucoup de temps car nous ne connaissions

pas la programmation par Arduino. Nous avons donc dû chercher des informations sur des bibliothèques Arduino sur internet par exemple.

Le prototype final en carton nous a aussi posé problème car nous n'avions pas mesuré l'épaisseur du bois mis à notre disposition. Pour remédier à ce problème, nous avons utilisé un couteau pour augmenter les trous de certaines pièces et les trous pour les vices.

Quelles sont les améliorations prioritaires à apporter selon vous ?  
(si vous aviez plus de temps/plus de moyens ?)

Les améliorations prioritaires à apporter selon nous réside dans le fait de contrôler avec plus de précisions les mouvements du bras robot. En effet, nous le contrôlons à l'aide du slider de la carte ESPLORA. Donc si nous avions plus de temps et plus de connaissances, nous aurions fait un programme qui aurait fixé une vitesse de rotation pour une meilleure maniabilité du bras robot.

De plus, si nous avions eu plus de temps, nous aurions conçu une base plus grande afin de pouvoir y mettre la carte Arduino ESPLORA. Nous aurions aussi réalisé le robot dans une matière de type plastique.

## **4 BILAN PERSONNEL ET PROFESSIONNEL**

Thomas : Ce projet m'a permis de connaître le logiciel Inkscape. Je maîtrise donc maintenant les bases de ce logiciel qui me sera peut-être utile pour d'autres projets. Le projet m'a permis de découvrir la découpe laser ainsi que la programmation sous Arduino. Je me suis aussi rendu compte de la difficulté de réaliser un simple objet électronique.

Concernant l'organisation du projet, il faudrait des séances d'EREP dès le début du semestre afin de nous présenter le projet pour avoir les idées claires avant la pré-étude. Il faudrait aussi plus de séances au Fablab

Sébastien : Ce projet s'est révélé très enrichissant dans la mesure où nous avons appris à créer un prototype ou à programmer dans un nouveau langage. En effet le travail que nous avons effectué en équipe sera un élément essentiel dans nos prochains projets.

De plus, ce projet nous a permis d'appliquer nos connaissances en programmation et en modélisation d'objet à l'aide du logiciel Inkscape.

Ce que je changerais dans l'organisation de ce projet c'est d'avoir plus de matériel pour l'ensemble des binômes, que ce soit au niveau des cartes Arduino ou des servomoteurs. Et en ce qui concerne l'Icam, il aurait fallu avoir plus de séances pour pouvoir travailler davantage sur le robot au Fablab.

## **5 ANNEXE**

→ Programme Final

```

#include <Servo.h> // Bibliothèque pour les servomoteurs
#include <Esplora.h> // bibliotheque pour la carte esplora

int Pos; // Variabbe pour la fonction map du slider

Servo s1; // Declaration du servo 1
Servo s2; // Declaration du servo 2
Servo s3; // Declarartion du servo 3

void setup() {

  Serial.begin(9600); // Intialisation du moniteur serial

  s1.attach(1); // le servo1 branché sur la patte 1
  s2.attach(0); // le servo2 branché sur la patte 0
  s3.attach(8); // le servo2 branché sur la patte 8

  Esplora.writeRGB(0,0,0); // Initalisation de la led
}

void loop() {

  // Partie moniteur pour avoir la valeur du potentiomètre du slider
  int Slider = Esplora.readSlider(); // Déclaration variable pour le slider
  Serial.print("\nSlider: "); // Ecrit "Slider:" dans le moniteur série
  Serial.print(Slider); // Ecrit la valeur du slider dans le moniteur série
}

```

```

int button, button1, button2; // Déclaration de variables pour les boutons

button = Esplora.readButton(SWITCH_LEFT); // Variable button associé au bouton du Gauche
button1 = Esplora.readButton(SWITCH_UP); // Variable button1 associé au bouton du Haut
button2 = Esplora.readButton(SWITCH_RIGHT); // Variable button2 associé au bouton de Droite

if (button == LOW) // Pour utiliser le servo1 il faut rester appuyé sur le bouton de gauche
{
  Esplora.writeRGB(255,0,0); // Quand on appuie sur le bouton de Gauche, la led s'allume en Rouge
  Pos=map(Slider,1023, 0, 0, 180); // Fonction map pour convertir la valeur du slider (0 et 1023) à la valeur du servomoteur (0 et 180)
  s1.write(Pos); // le servo 1 lit la valeur
  delay(15);
}

Esplora.writeRGB(0,0,0); // On reinitialise la led

if (button1 == LOW) // Pour utiliser le servo2 il faut rester appuyé sur le bouton du haut
{
  Esplora.writeRGB(0,255,0); // Quand on appuie sur le bouton du Haut, la led s'allume en Vert
  Pos=map(Slider,1023, 0, 0, 180); // Fonction map pour convertir la valeur du slider (0 et 1023) à la valeur du servomoteur (0 et 180)
  s2.write(Pos); // le servo 2 lit la valeur
  delay(15);
}

Esplora.writeRGB(0,0,0); // On reinitialise la led

```

```

if (button2 == LOW) // Pour utiliser le servo3 il faut rester appuyé sur le bouton de droite
{
  Esplora.writeRGB(0,0,255); // Quand on appuie sur le bouton du Droite, la led s'allume en Bleu
  Pos=map(Slider,0, 1023, 0, 180); // Fonction map pour convertir la valeur du slider (0 et 1023) à la valeur du servomoteur (0 et 180)
  s3.write(Pos); // le servo 3 lit la valeur
  delay(15);
}

Esplora.writeRGB(0,0,0); // On reinitialise la led

```

## ➔ Programme Python

```

#include <Servo.h> // Bibliothèque pour les servomoteurs

int pos1; // Déclaration variable pour les boucles for

Servo s1; // Déclaration servomoteur 1
Servo s2; // Déclaration servomoteur 2
Servo s3; // Déclaration servomoteur 3

void setup() {
  Serial.begin(9600); // Initialisation du moniteur série
  s1.attach(1); // Le servomoteur est branché sur patte 1
  s2.attach(0); // Le servomoteur est branché sur patte 0
  s3.attach(8); // Le servomoteur est branché sur patte 8
}

```

```

void loop() {
  if (Serial.available())
  {
    switch(Serial.read())
    {
      // Case '0' pour que le servomoteur 1 tourne dans un sens puis dans l'autre sens
      case '0' : for(pos1 = 0; pos1>180; pos1+=1)
        {
          s1.write(pos1);
          delay(8);
        }

        for(pos1 = 180; pos1>=0; pos1--=1)
        {
          s1.write(pos1);
          delay(8);
        }
        break;
    }
  }
}

```

```

// Case '1' pour que le servomoteur 2 tourne dans un sens puis dans l'autre sens
case '1' : for(pos1 = 0; pos1>180; pos1+=1)
  {
    s1.write(pos1);
    delay(8);
  }

  for(pos1 = 180; pos1>=0; pos1--=1)
  {
    s1.write(pos1);
    delay(8);
  }
  break;

// Case '0' pour que le servomoteur 3 tourne dans un sens puis dans l'autre sens
case '2' : for(pos1 = 0; pos1>180; pos1+=1)
  {
    s1.write(pos1);
    delay(8);
  }

  for(pos1 = 180; pos1>=0; pos1--=1)
  {
    s1.write(pos1);
    delay(8);
  }
  break;

default : break;

```

#### Jupyter QtConsole

File Edit View Kernel Window Help

Jupyter QtConsole 4.2.1

Python 3.6.0 [Anaconda 4.3.1 (64-bit)] (default, Dec 23 2016, 11:57:41) [MSC v.1900 64 bit (AMD64)]  
 Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.

? -> Introduction and overview of IPython's features.

%quickref -> Quick reference.

help -> Python's own help system.

object? -> Details about 'object', use 'object??' for extra details.

In [1]: `import serial` # on importe cette bibliothèque pour pouvoir communiquer entre la carte arduino et python

In [2]: `ser=serial.Serial('COM3',9600)` # on crée une variable pour pouvoir communiquer entre python et la carte arduino

In [3]: `while 1:` # crée une boucle infinie

...: `val=raw_input("Entrer une valeur entre 0 et 2");` # on crée une variable qui va lire la valeur qu'on va entrer dans python et qui va être renvoyée à la carte arduino

...: `ser.write(val)` # la variable 'ser' va lire la valeur que l'on aura entrée

...: |