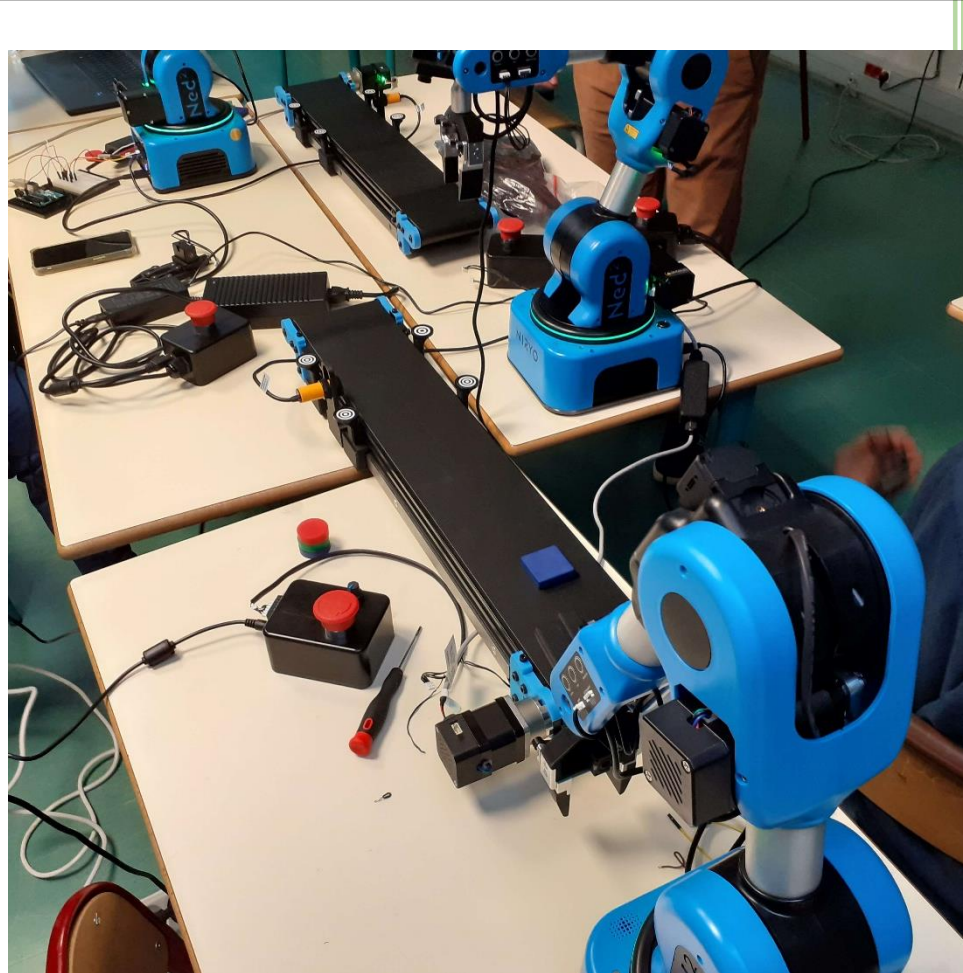


# SAE5

## Pilotage du convoyeur par le NED2 et la carte Arduino



TOGBOSSI Kevin

THIAM Mohammed

GEII

SAE5

## Sommaire

<b>Introduction .....</b>	<b>2</b>
<b>I- Utilisation du logiciel Niryo studio.....</b>	<b>3</b>
<b>II- Programme avec pick and place.....</b>	<b>6</b>
<b>III- Programme avec vision .....</b>	<b>7</b>
<b>IV- Projet SAE.....</b>	<b>10</b>
<b>1) Introduction .....</b>	<b>10</b>
<b>2) Schéma électrique .....</b>	<b>10</b>
<b>3) Programme Arduino .....</b>	<b>11</b>
<b>4) Programme Blockly.....</b>	<b>12</b>
<b>Conclusion:.....</b>	<b>14</b>

## Introduction

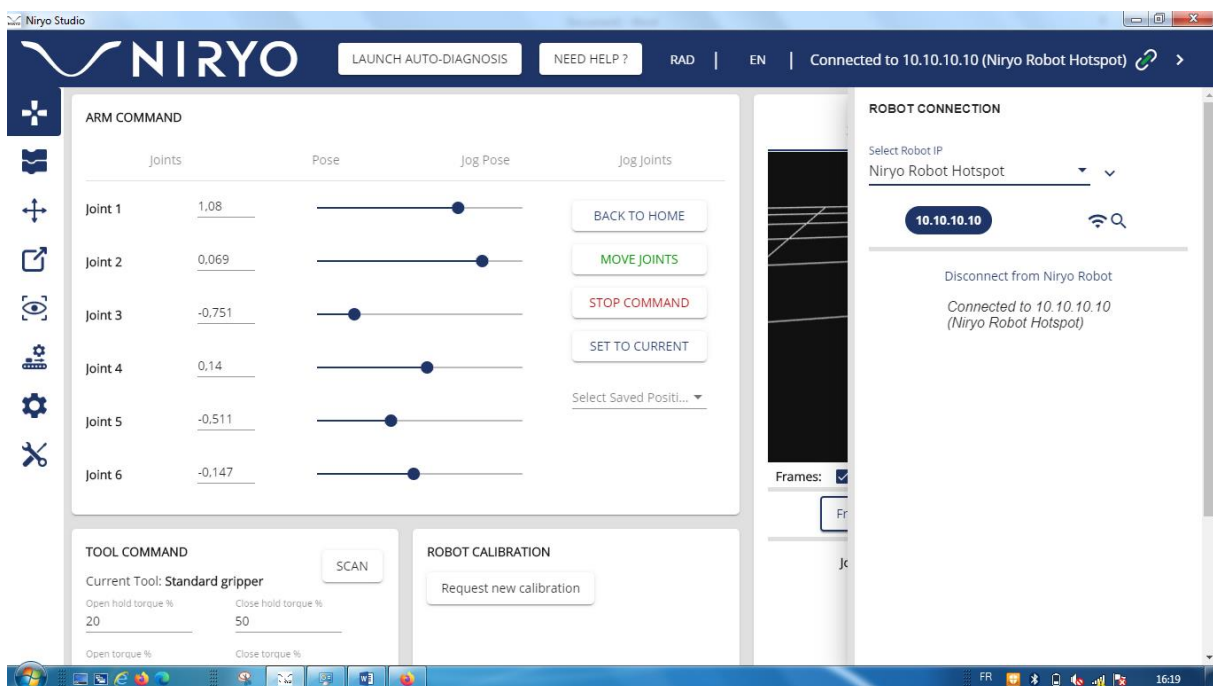
Dans ce projet nous avons utilisé le robot Niryo Ned 2. Dans un premier temps nous avons pris connaissance du langage de programmation Blockly pour le robot. C'est une bibliothèque Google inspirée du langage Scratch développée par le MIT aux États-Unis. Une fois que nous avons pris connaissance du langage nous allons réaliser un projet de SAE en collaboration avec les autres robots Ned 2 commandés par nos camarades.

## I- Utilisation du logiciel Niryo studio

Pour commencer on a connecté notre PC au réseau wifi du robot.









Puis sur l'application Niryo Studio en local hotspot avec l'adresse 10.10.10.10.




Après s'être connecté au réseau, on lance le calibrage automatique sur l'application puis on regarde les positions pour lesquelles le robot percute un obstacle dans notre zone de travail et on les note ci-dessous :

## ARM COMMAND

Joints		Pose	Jog Pose	Jog Joints
Joint 1	<input type="text" value="0"/>			<div>BACK TO HOME</div> <div>MOVE JOINTS</div> <div>STOP COMMAND</div> <div>SET TO CURRENT</div> <div>Select Saved Position ▼</div>
Joint 2	<input type="text" value="-0,27"/>			
Joint 3	<input type="text" value="-1,25"/>			
Joint 4	<input type="text" value="0"/>			
Joint 5	<input type="text" value="0"/>			
Joint 6	<input type="text" value="0"/>			

Joints (rad)	Joints (rad)
1.80	-2.73
-0.54	-0.69
0.25	1.02
0.16	0.21
-0.38	-0.73
0.00	-0.15
<div>SAVE</div>	<div>SAVE</div>

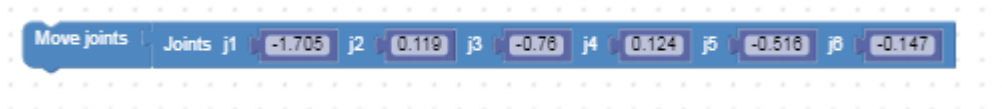
Pour pouvoir déplacer les 6 axes du robot on peut soit resté appuyer sur le bouton free motion situé sur la tête du robot ou on peut rester appuyer sur le bouton free motion situé dans l'application.



Frames: <input checked="" type="checkbox"/> TCP <input checked="" type="checkbox"/> Base link		Legend: → >	
<div>Free Motion</div>		<div>STOP</div>	
<b>Joints (rad)</b> -1.71 0.12 -0.76 0.13 0.52		<b>Position</b> x : -0.0: y : -0.2: z : 0.23 <b>Orientation</b> Roll : 0.1	

## 1) Mode apprentissage du robot Nyrio

1.1) Pour déplacer le robot il existe 2 manières. La première manière pour que le robot atteigne la position est le mode free motion et le deuxième moyen le mode bloc.

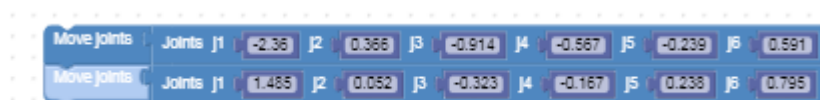


1.2) Les angles maximaux du robot sont : J6= 2.53 ; -2.53, J5 = 1.92, J4 = 2.09 et -2.07, J3 = 1.51 et -1.34, J2 = 0,61 et -1.84, J1 = 2.94 et -2.96

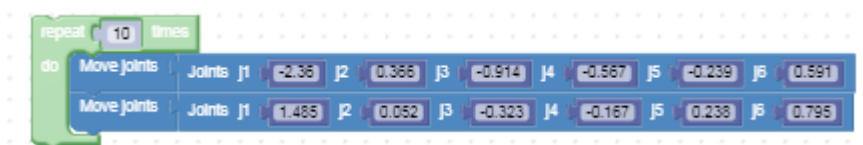
Pour sauvegarder la position on appuie sur le bouton Save ou on clique sur Save sur l'application ce qui nous créera un bloc comme ci-dessous. Et si on déplace le robot a la main on peut le remettre à cette position en appuyant sur le bouton Play dans l'application.

## 2) Programmation sous blocky du robot Nyrio Ned

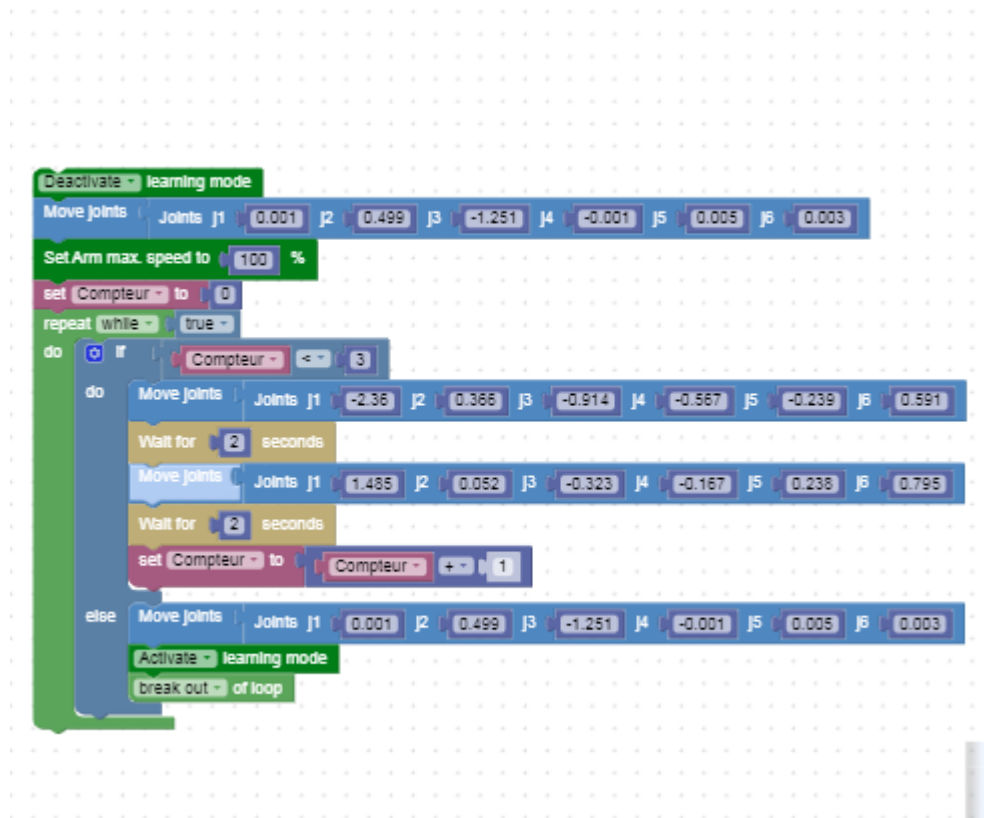
2.1) Pour programmer sous blocky les position P1 et P2 on va dans l'onglet mouvement et on sélectionne le block move joints. Ensuite on configure les positions des 6 axes en entrant les valeurs dans les cases de J1 à J6.



2.2) Pour répéter 10 fois le mouvement P1 et P2 on va dans l'onglet « Logic » et on sélectionne le bloc « repeat »

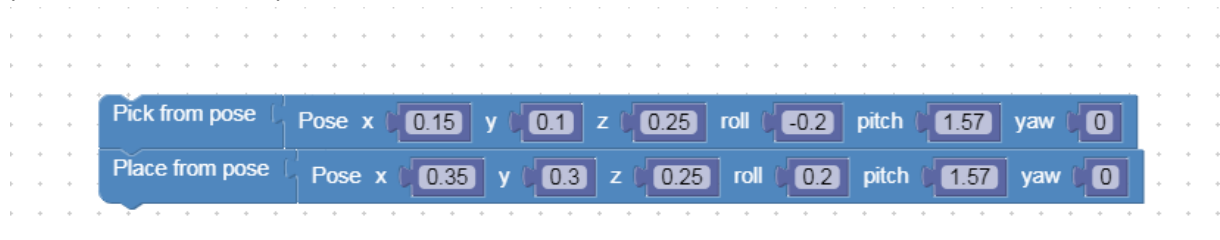


3.3)



## II- Programme avec pick and place



Pour faire un « pick and place » en blockly on va dans l'onglet « movement » puis on sélectionne le block « pick from pose » pour attraper une pièce et le block « place from pose » pour déposer la pièce dans un endroit précis



### III- Programme avec vision


Pour programmer avec vision, on va dans l'onglet « vision » puis on sélectionne le boutons « + » dans Workspaces pour configurer un workspace . Puis un onglet s'ouvre avec les indications à faire pour configurer avec le robot le workspace, on met l'outil pique sur le port outils du robot puis on pointe le milieu des cercles sur les 4 coins du workspace en enregistrant à chaque fois la position. Cela va servir à délimiter l'espace de travail du robot avec la camera.

#### WORKSPACES

Select a workspace  

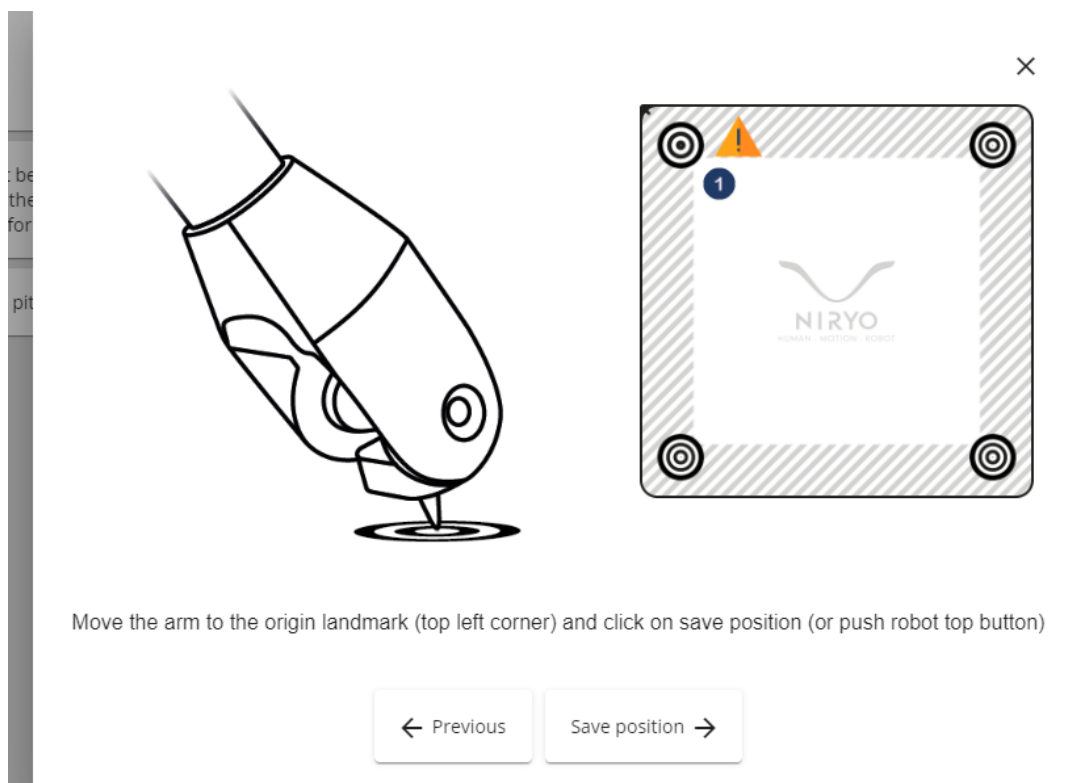
#### CAMERA STREAMING SETTINGS

Enable vision module



Everytime your want to use vision capabilities, the robot must be in an observation position where the 4 landmarks of the workspace area are fully visible by the camera and that the camera is as parallel as possible to the workspace.  
(For more information, please consult the user manual)

Tips: To place the camera in an optimal position, favor a position with pitch values = 1.57, yaw = 0 and a roll  $\approx$  0 (to be adjusted according to vision / gripper)

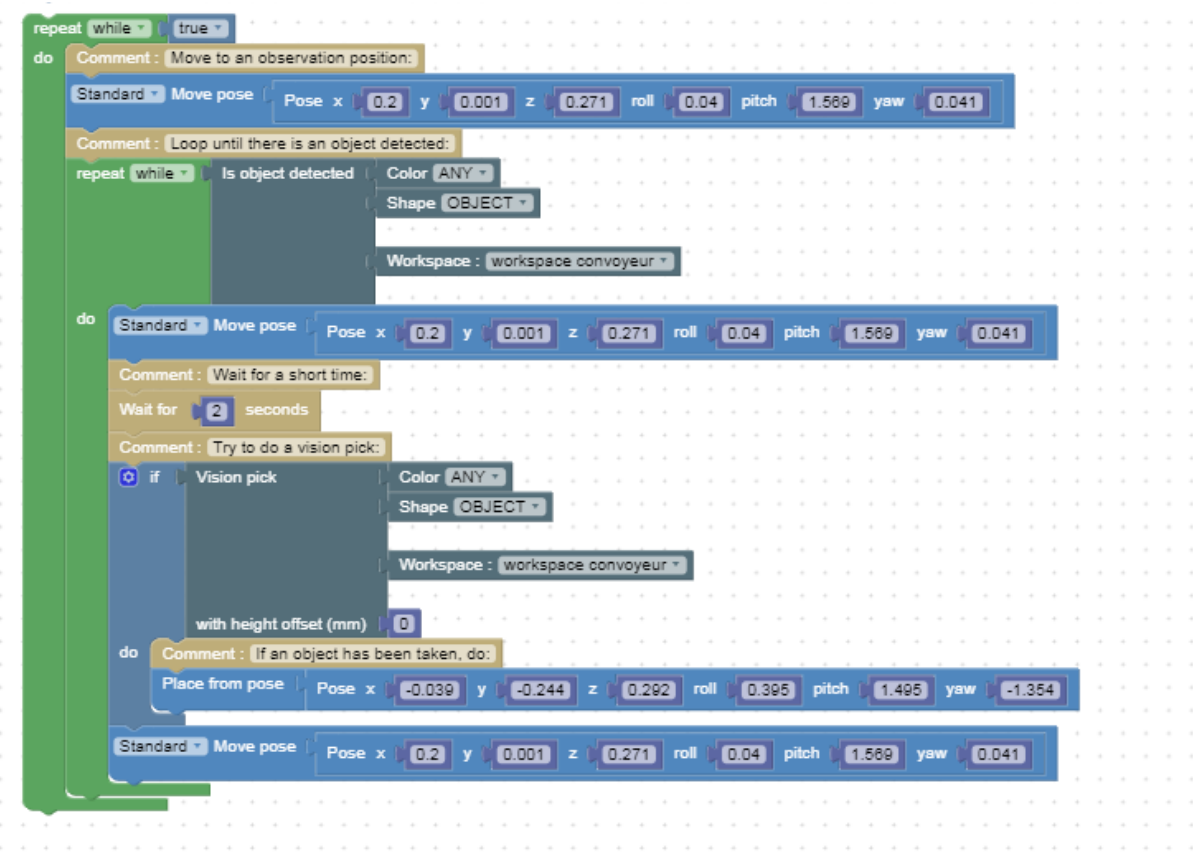


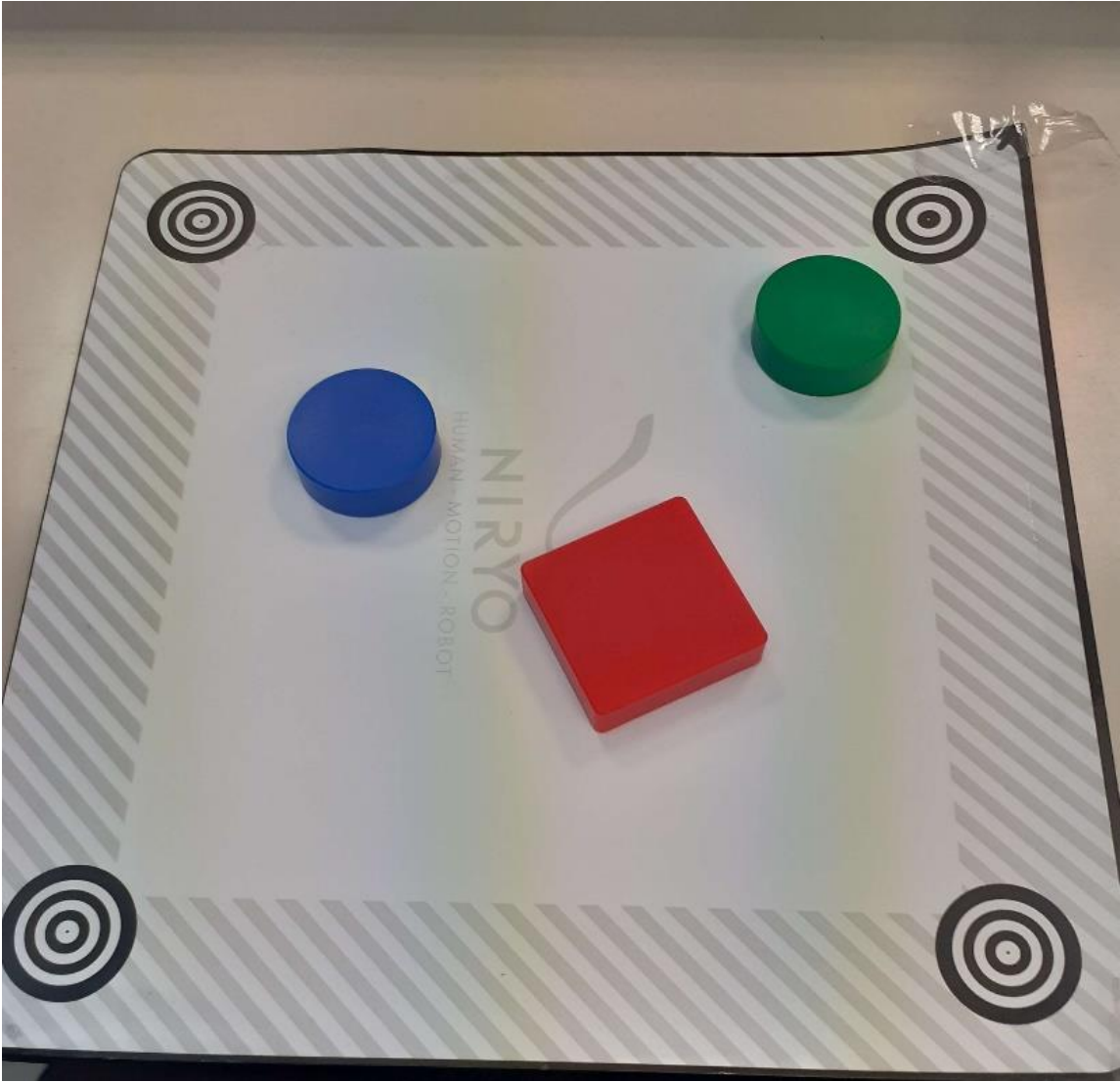


Workspaces du robot :



Après avoir configuré le workspace. On va déterminer avec le robot la position d'observation pour qu'il voit les 4 cercles du workspace et on détermine la position de dépôt de l'objet. Puis on va créer un programme Blockly pour que le robot prenne une pièce quand il le détecte avec la caméra sur le workspace et le déplace sur la position déterminée





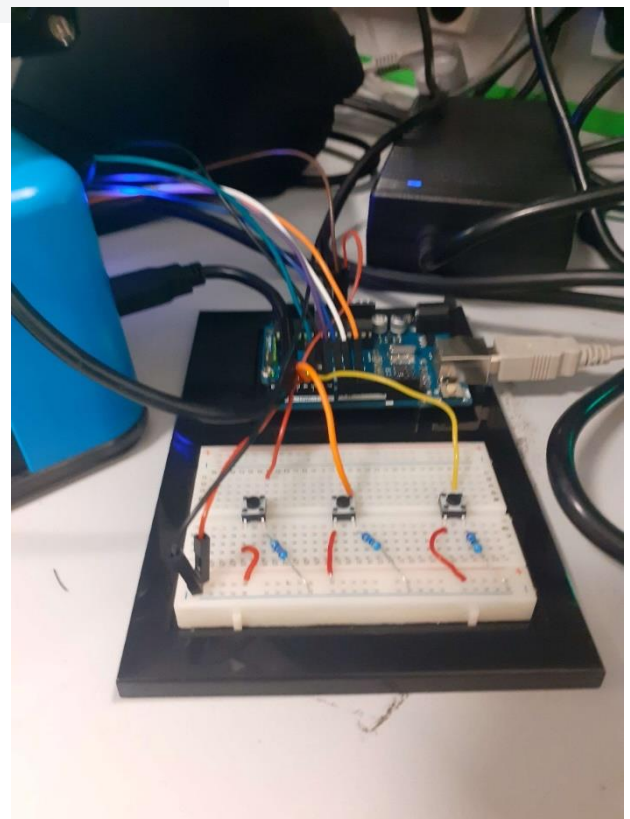
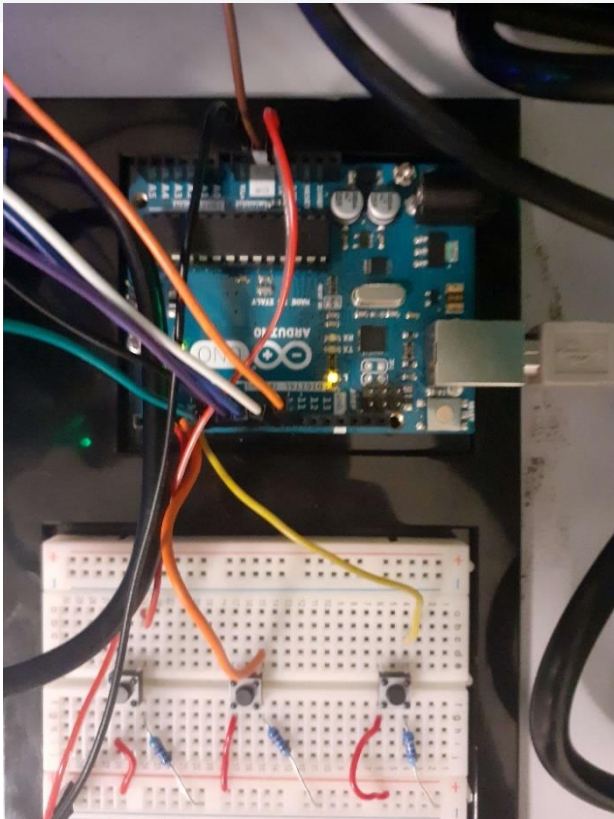
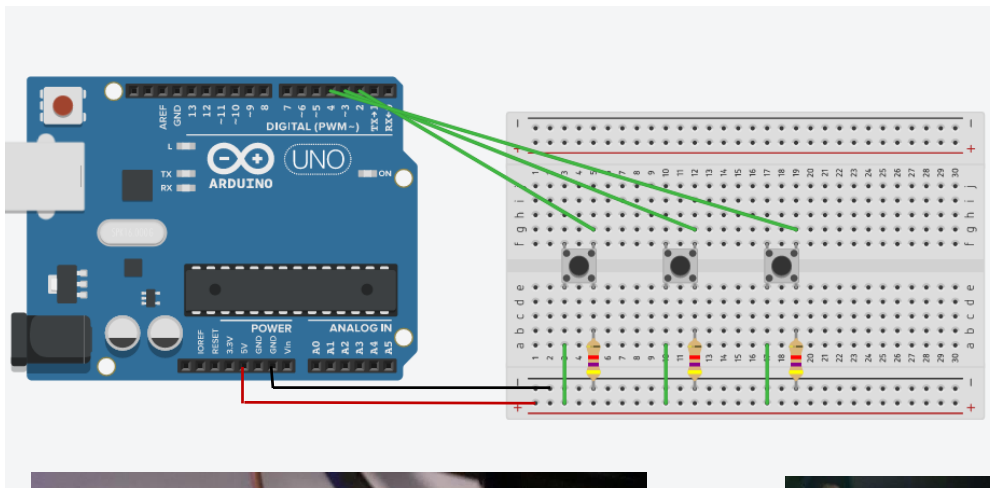
## IV- Projet SAE

### 1) Introduction

Dans cette SAE nous allons créer une interface de contrôle commande du convoyeur par robot Nyrio-NED 2 pour effectuer une chaine industrielle avec les 3 bras robot et les convoyeurs via un pilotage par une carte Arduino-Uno.

### 2) Schéma électrique

Après avoir câblé sur la plaque les 3 boutons poussoirs on va câbler les entrées du robot sur l'Arduino sur les port digital 5,6,7 et les sorties du robot sur les port digital 8,9,10



Tâche 1 : mise en place du fonctionnement avec les boutons poussoir + Tâche 2 : Collaborer avec un autre bras robot en utilisant le convoyeur

### 3) Programme Arduino

Les programmes Arduino sont les mêmes pour les deux tâches.

```
saeb-robot

int bouton1 = 2;
int bouton2 = 3;
int bouton3 = 4;
int out1 = 8;
int out2 = 9;
int out3 = 10;
void setup() {
  pinMode(bouton1, INPUT);
  pinMode(bouton2, INPUT);
  pinMode(bouton3, INPUT);
}

void loop() {
  int pick = digitalRead(bouton1);
  int place = digitalRead(bouton2);
  int reset = digitalRead(bouton3);
  int pickout = digitalRead(out1);
  int placeout = digitalRead(out2);
  int resetout = digitalRead(out3);

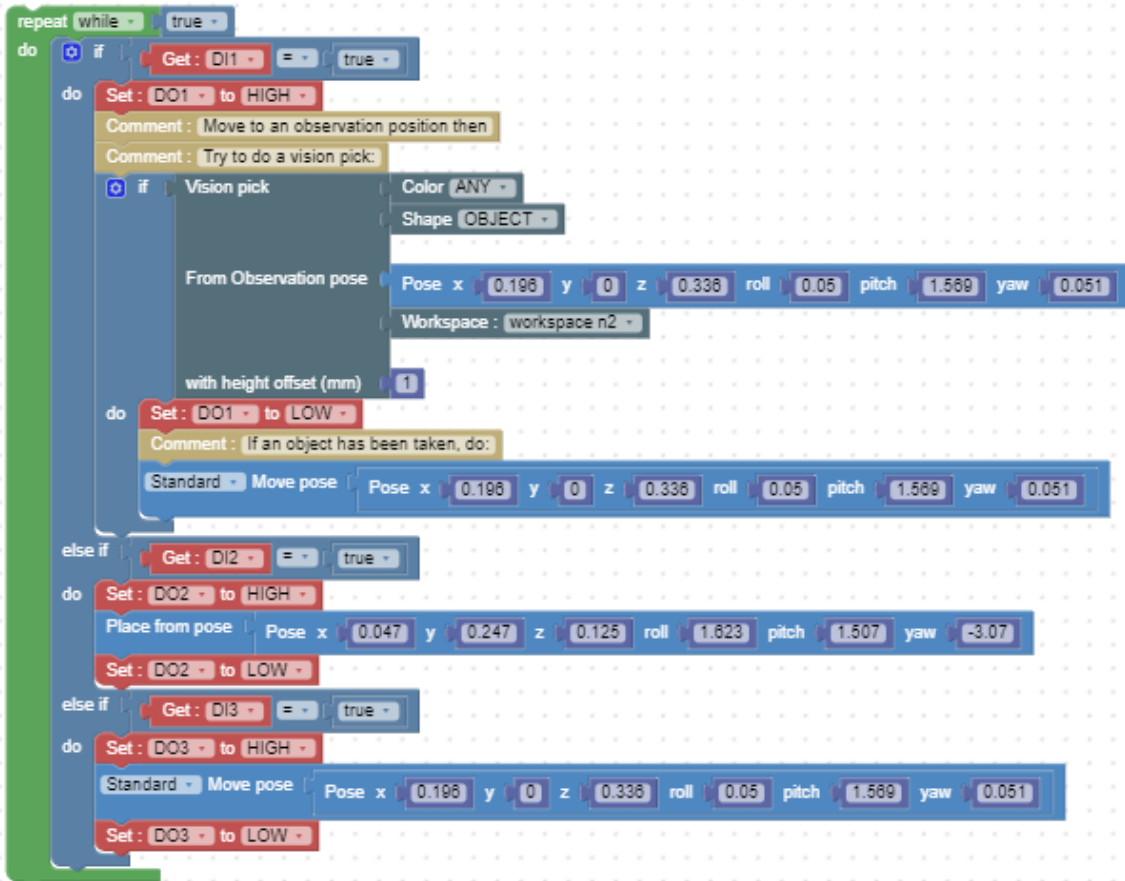
  if(pick == HIGH)
  {
    digitalWrite(5, HIGH);
  }
  else if(place == HIGH)
  {
    digitalWrite(6, HIGH);
  }
  else if(reset == HIGH)
  {
    digitalWrite(7, HIGH);
  }
  if(pickout == HIGH)
  {
    digitalWrite(5, LOW);
  }
  if(placeout == HIGH)
  {
    digitalWrite(6, LOW);
  }
  if(resetout == HIGH)
  {
    digitalWrite(7, LOW);
  }
}
```

Sur Arduino, on configure les entrées des 3 boutons poussoirs. Puis on utilise la fonction (digitalRead) pour lire l'état des boutons poussoirs et des sorties du robot. On crée des conditions pour mettre à l'état haut les entrées du bras robot selon le boutons poussoir appuyer.

#### 4) Programme Blockly

Programme blockly tâche 1:

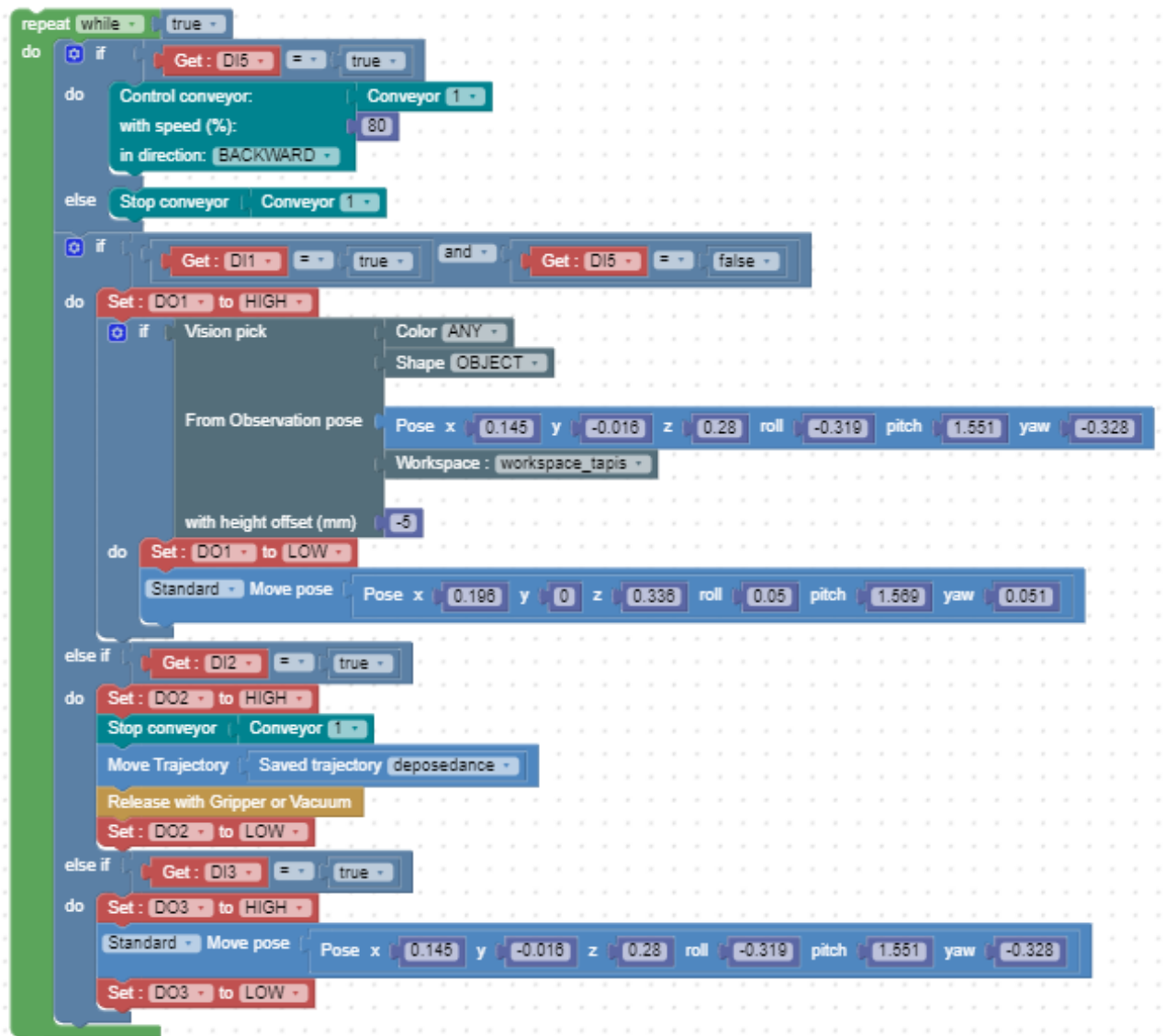
Dans le programme blockly on fait plusieurs conditions quand on appuie sur les bouton poussoirs qu'on a câblés sur les entrées DI1, DI2, DI3 du robot. On active les sorties D01, D02, D03 du robot



selon les conditions. La sortie D01 sert à faire la vision pour récupérer une pièce détecter par la caméra sur le workspaces. La sortie D02 sert à poser la pièce dans une position déterminer. La sortie D03 sert à faire revenir le robot dans sa position initiale.

Programme blockly tâche 2:

On programme la tâche 2 avec blockly. Tout d'abord on va mettre la condition pour que si le capteur



ne détecte pas un objet le convoyeur fonctionne sinon le convoyeur s'arrête. Puis on a repris le programme de la tâche 1 en changeant le workspaces avec celui du convoyeur et les position d'observation et de dépose

## Conclusion:

Pour conclure, on a appris à utiliser le bras robot avec le convoyeur et le logiciel Niryo puis on a fait un projet en collaboration avec d'autre bras robot afin de mettre en place une situation réel d'automatisation industriel.