

Les différents pseudocodes du programme

Fonction pour calculer les coordonnées en fonction des angles θ_1 et θ_2 , à partir du modèle géométrique.

Calcul_co()

$$x = L * (\sin(\theta_2) - \sin(\theta_1))$$

$$y = L * (\cos(\theta_2) + \cos(\theta_1))$$

Fonction pour calculer le point le plus proche de la position actuelle, sur le cercle de centre (X_c, Y_c) de rayon l .

Pt_le_plus_proche_sur_le_cercle()

Si $X = 0$ *//Sur l'abscisse du repère du cercle*

$$X_0 = 0$$

Si $Y > Y_c$

$$Y_0 = Y_c + l$$

Sinon

$$Y_0 = Y_c - l$$

Sinon Si $Y = 0$ *//Sur l'ordonnée du repère du cercle*

$$Y_0 = 0$$

Si $X > 0$

$$X = X_c + l$$

Sinon

$$X = X_c - l$$

Sinon *//Point pas sur les axes du repère du cercle*

$$\alpha = \text{atan}(Y/X)$$

$$X_0 = X_c + l * \cos(\alpha)$$

$$Y_0 = Y_c + l * \sin(\alpha)$$

Fonction pour calculer le point le plus proche permettant de tracer un trait de 5cm dans le carré de 5cm par 5cm. Les tableaux Xa et Ya sont les coordonnées dans les angles du carré.

Pt_le_plus_proche_valable()

```
id = 0
max = 0
Pour i allant de 1 à 4 //Calcul de la distance maximale par rapport au 4 coins
    d = ( X - Xa[i])2 + ( Y - Ya[i])2 )1/2
    Si d > max
        max = d
    Si max < 5
        id = i
Si max ≥ 5 //Trait possible depuis la position actuelle
    id = 0
Si id = 0
    fin
Sinon //Déplacement vers le point le plus proche valable
    α = atan( abs(Ya[id] - Y) / abs(Xa[id] - X) )
    Si id = 1 ou id = 2
        Y0 = Ya[id] + 5*sin(α)
    Sinon
        Y0 = Ya[id] - 5*sin(α)
    Si id = 2 ou id = 3
        X0 = Xa[id] + 5*cos(α)
    Sinon
        X0 = Xa[id] - 5*cos(α)
Fin
```

Fonction pour parcourir le cercle

Dessiner_Cercle()

```

Pt_le_plus_proche_sur_le_cercle() //Définir point de départ du parcours
Aller(X0,Y0) //Déplacement vers le point de départ du parcours
α = atan(Y0/X0)
cpt = 0
test = 1
Pour i allant de α à α+360, par pas de 0.1° //Parcours du cercle
    X0 = Xc + l*cos(i) //coordonnée objectif sur x
    Y0 = Yc + l*sin(i) //Coordonnée objectif sur y
    Aller(X0,Y0) //Aller à l'objectif
    Attendre(10/3600) //pause pour un déplacement de 10 s
    cpt = cpt + 1
    Si cpt = 200 et pointillé //Si écriture en pointillé
        Dessiner(test)
        test = !test //Inversion de la position de la plaque

```

Fonction pour mettre la plaque de dessin en position haute ou basse

Dessine(test)

```

Si test
    Servo3 en position haute //Position pour écrire
Sinon
    Servo3 en position basse //Position pour se déplacer sans écrire

```

Fonction pour aller au coordonnées (X,Y) à partir du modèle géométrique inverse

Aller(X,Y)

```

α = atan(X0/Y0)
l = (X02 + Y02)1/2
α3 = acos( (2*L2 - l2)/(2*L2) )
α2 = 90 - α3/2
t1 = α1 + α2

```

t2 = $\alpha_2 - \alpha_1$

Servo gauche en position t2

Servo droite en position t1

Fonction pour dessiner une ligne de 5 cm dans le carré de 5cm par 5cm

Dessiner_ligne()

```
Pt_le_plus_proche_valable()      //Définir le point de départ
Dessine(0)                       //Lever le stylo
Aller(X0,Y0)                     //Coordonnée défini par Pt_le_plus_proche_valable
Dessine(1)                       //Baissier le stylo
d = 0.1
cpt = 0
Plage_angle()                   //Fonction qui renvoie deux angles a1 et a2
 $\alpha$  = nombre aléatoire entre a1 et a2
Tant que cpt < 50/d //Parcours du trait
    cpt = cpt + 1
    X = X + d*cos( $\alpha$ )          //Calcul du nouveau point
    Y = Y + d*sin( $\alpha$ )          //Calcul du nouveau point
    Aller(X,Y)                   //Déplacement vers le prochain point
    Attendre(10/(50/d))          //Pause pour un parcours de 10s
```

Fonction pour trouver la plage d'angle possible au point (X,Y)

Plage_angle()

a1 = -1

a2 = 0

Pour a allant de 0 à 360

$X_i = X + 5*\cos(a)$

$Y_i = Y + 5*\sin(a)$

Si $X_i \in [-l, l]$ et $Y_i \in [Y_c - l; Y_c + l]$ //Point du cercle dans le carré pour l'angle a

```
        Si a1 = -1
            a1 = a           //Angle minimum
        Sinon
            a2 = a           //Angle maximum
    retourner a1 et a2
```

Fonction principale

Main()

```
    Si BP_Cercle
        Dessiner_Cercle()           //Dessiner le cercle
    Si BP_Trait
        Dessiner_Trait()           //Dessiner le trait
    Si BP_Type
        var_type = !var_type       //Définir le type de trait
    //Déplacement selon le joystick
    dX : entrée du joystick entre -1 et 1
    dY : entrée du joystick entre -1 et 1
    StepX = Vmax*dX
    StepY = Vmax*dY
    X = X + StepX
    Y = Y + StepY
    Attendre(1/100)
```