

// Cet algorithme propose de piloter un bras-robot (ou tout autre robot équipé de moteurs électriques) grâce à une interface graphique. Ici, il suffit de cliquer à l'intérieur des cases (boutons) pour activer ou non la sortie correspondante. La case sélectionnée change alors de couleur.

:x0 , :y0 coordonnées initiales du centre de la case supérieure.
:xmax est l'abscisse de fin de la zone de sélection.
:longueur , :largeur sont les dimensions de chaque case.
:ymax, :ymin sont respectivement la plus grande ordonnée de la case supérieure et la plus petite ordonnée de la case inférieure. Elles définissent la zone de sélection.
:ys est l'ordonnée supérieure de chaque case selon l'indice de la boucle de balayage pendant la sélection des cases.
:sélection est une liste de 8 chiffres qui indique si la sortie correspondante est active (1) ou non (0).
:i est un indice de balayage pour repérer la case sélectionnée.
:listey contient les ordonnées supérieures et inférieures de chaque case-bouton.
:sél est le mot correspondant à la liste :sélection .
:s est le nombre correspondant au mot :sél pour être finalement validé par fixecanaux_1 .
:arrêt est une valeur booléenne qui interrompt la boucle de balayage pendant la sélection des cases.
:a , :b sont deux nombres qui autorisent la boucle de balayage tant que :a est inférieur à :b. modifiés par :arrêt, la boucle s'interrompt. La boucle de balayage n'est donc pas infinie.

//

```
pour interface
efftxt
si connecte 0 [
eff ct donne "x0 -100 donne "y0 217
teste clic?
donne "largeur 80 donne "hauteur 30
donne "xmax :x0 + :largeur
donne "listey []
// dimensions d'une case de sélection: :largeur x :hauteur pixels //
donne "x :x0 - entier (:largeur / 2) donne "y :y0 + entier (:hauteur / 2)
fcc noire donne "sél 0
répète 8 [
segment [ :x :y ] [ :x + :largeur :y ]
segment [ :x + :largeur :y ] [ :x + :largeur :y -:hauteur ]
segment [ :x + :largeur :y -:hauteur ] [ :x :y -:hauteur ]
segment [ :x :y -:hauteur ] [ :x :y ]
si boucle = 1 donne "objet [ Moteur pinces. ] // objet à modifier selon le type de robot. //
si boucle = 2 donne "objet [ Moteur poignet. ]
si boucle = 3 donne "objet [ Moteur avant-bras. ]
si boucle = 4 donne "objet [ Moteur bras. ]
si boucle = 5 donne "objet [ Moteur orientation du bras. ]
si boucle = 6 donne "objet [ Allumage de la diode. ]
si boucle = 7 donne "objet [ Inversion de la polarité. ]
si boucle = 8 donne "objet [ Déconnecte. ]
dessinetexte [ (:x + :largeur + 10) :y - entier (:hauteur / 3) 0] : objet
teste :listey = []
sivrai donne "listey ph :y :y -:hauteur
sifaux donne "listey liste :listey ph :y :y -:hauteur
teste boucle =8
sifaux [
colorie [ :x + 5 :y -3 ] rouge
dessinetexte [ (:x + :largeur + 10) :y - entier (:hauteur / 3) 0] : objet
dessinetexte [ :x + 15 :y - entier (:hauteur / 3) 0] ph [sortie n°] boucle
]
sivrai [
colorie [ :x + 5 :y -3 ] verte
dessinetexte [ :x + 15 :y - entier (:hauteur / 3) 0] :objet
]
donne "y :y - :hauteur - entier (:hauteur / 2)
]
donne "i 0 donne "sélection [ 0 0 0 0 0 0 0 0 ] donne "a 1 donne "b 2
donne "ymax saisis :listey [ 1 1] donne "ymin saisis :listey [ 8 2]
tantque :a < :b [ // 1 //
teste clic?
sivrai [ // 2 //
teste surécran?
sivrai [ // 3 //
donne "i 1 donne "arrêt faux
tantque (:arrêt = faux) [ // 4 //
si ((sourisx >= :x) et
(sourisx <= :xmax)) [ // 5 //
si ((sourisy <= :ymax) et
```

```

        (sourisy >= :ymin) [ // 6 //
tantque ((:i <= 7) et (sourisy <= saisis :listey [ :i 2])) donne "i :i + 1
    donne "ys saisis :listey [ :i 1]
teste :i = 8
sivrai [ // 7 //
    fcc verte
    dessinetexte [ :x + 15 (saisis :listey [ 8 1]) - entier (:hauteur / 3) 0] :objet
    fcc noire
    fixecanaux_1 0 déconnecte donne "arrêt vrai permute :a :b
    colorie [ :x + 5 :ys -3 ] rouge
    attends 0,1
    colorie [ :x + 5 :ys -3 ] verte
    dessinetexte [ :x + 15 (saisis :listey [ 8 1]) - entier (:hauteur / 3) 0] :objet
] // -7 //
sifaux [ // 7 //
    teste ((sourisy <= :ys) et
        (sourisy >= saisis :listey [ :i 2]))
        sivrai [ // 8 //
            donne "arrêt vrai
            teste ((item :sélection 9- :i) = 1)
            sivrai [ // 9 //
                remplace :sélection 0 [ 9- :i ]
                colorie [ :x + 5 :ys -3 ] rouge
            dessinetexte [ (:x + 15) :ys - entier (:hauteur / 3) 0] ph [sortie n°] :i
            ] // -9 //
            sifaux [ // 9 //
                remplace :sélection 1 [ 9-:i ]
                colorie [ :x + 5 :ys -3 ] tilleul
            dessinetexte [ (:x + 15) :ys - entier (:hauteur / 3) 0] ph [sortie n°] :i
            ] // -9 //
        fcc eau dessinetexte [ :x :ymin - entier (:hauteur / 3) 0] ph "fixecanaux_1 :sél fcc noire
        donne "sél "
        répète 8 donne "sél mot :sél item :sélection boucle
        donne "i 1 tantque ((item :sél :i) = 0) donne "i :i + 1
        donne "s "
        répète 9 - :i [ donne "s mot :s item :sél :i donne "i :i + 1 ]
        donne "sél nombre :s
        donne "sél translate :sél (-2)
        fixecanaux_1 :sél
        dessinetexte [ :x :ymin - entier (:hauteur / 3) 0] ph "fixecanaux_1 :sél
            ] // -8 //
        ] // -7 //
        ] // -6 //
        ] // -5 //
        ] // -4 //
        ] // -3 //
        teste surécran?
    ] // -2 //
    teste clic?
    ] // -1 //
]
fin

interface

```