

ARPRO - IRP - Lab n°1

Matthieu LAMIELLE

Muruo WANG

Exercice 1

Code version 1 :

```
1  edit MLMU1.version1
2  ; Matthieu LAMIELLE et Muruo WANG
3
4  ;inputs
5  part.det.P1 = 1004
6  part.det.P2 = 1005
7  part.det.R2 = 1008
8  part.det.R1 = 1007
9
10 ;output
11 conveyor.start = 1
12 conveyor.stop = -1
13 conveyor.g = -2
14 conveyor.d = 2
15
16 signal 5
17 tool suction.pad
18
19 ;Tool check
20 openi
21
22 ;Debut
23 move wait.loc
24
25 ;On vérifie qu'une pièce est en place
26 wait sig(part.det.P1) or sig(part.det.P2)
27
28 ;On regarde où est située la pièce à prendre
29 if sig(part.det.P1) then
30     set P=P1
31 else
32     set P=P2
33 end
34
35 ;On va la chercher
36 appro P, 100
37 moves P
38 closei
39 departs 50
40
41 ;On la pose sur le tapis
42 appro place,50
43 moves place
44 openi
45 departs 100
46
47 ;On le fait partir vers la droite
48 signal(conveyor.d)
49 signal(conveyor.start)
50
51 ;Dès que la pièce est au detector 2 on stop le tapis
52 wait sig(part.det.R2)
53 signal(conveyor.stop)
54
55 ;On attend
56 timer 1 = 0
57 wait timer(1) > 1
58
59 ;On redémarre dans l'autre sens
60 signal(conveyor.g)
61 signal(conveyor.start)
62
```

```

62 |
63 | ;Quand la pièce est au detector 1 on stop le tapis
64 | wait sig(part.det.R1)
65 | signal(conveyor.stop)
66 |
67 | ;On prend la pièce sur le tapis (à detector 1)
68 | appro grasp.loc.stopped, 100
69 | moves grasp.loc.stopped
70 | closei
71 | departs 50
72 |
73 | ;On pose la pièce en P3
74 | appro P3,50
75 | moves P3
76 | openi
77 | departs 100
78 |
79 | move wait.loc
80 |
81 | e
82 |
83 | load pickconv.lc
84 | load alltools.v2

```

Code version 2 :

```

1 | edit MLMU1.version2
2 | ; Matthieu LAMIELLE et Muruo WANG
3 |
4 | ;inputs
5 | part.det.P1 = 1004
6 | part.det.P2 = 1005
7 | part.det.R2 = 1008
8 | part.det.R1 = 1007
9 |
10 | ;output
11 | conveyor.start = 1
12 | conveyor.stop = -1
13 | conveyor.g = -2
14 | conveyor.d = 2
15 |
16 | signal 5
17 | tool succion.pad
18 |
19 | ;Tool check
20 | openi
21 |
22 | speed 30 always
23 | speed 100 mmmps always
24 |
25 | ;Debut
26 | move wait.loc
27 |
28 | ;On vérifie qu'une pièce est en place
29 | wait sig(part.det.P1) or sig(part.det.P2)
30 |
31 | ;On regarde où est située la pièce à prendre
32 | if sig(part.det.P1) then
33 |     set P=P1
34 | else
35 |     set P=P2
36 | end

```

```

37
38 ;On va la chercher
39 appro P, 100
40 moves P
41 closei
42 departs 50
43
44 ;On la pose sur le tapis
45 appro place,50
46 moves place
47 openi
48 departs 100
49
50 ;On place le robot en attente
51 appro grasp.loc.stopped, 150
52
53 ;On fait partir le tapis vers la droite
54 signal(conveyor.d)
55 signal(conveyor.start)
56
57 ;Dès que la pièce est au detector 2 on commence le timer
58 wait sig(part.det.R2)
59 timer 1 = 0
60 wait timer(1) > 3
61
62 ;On redémarre dans l'autre sens après un delais de 1s
63 signal(conveyor.stop)
64 signal(conveyor.g)
65 timer 1 = 0
66 wait timer(1) > 1
67 signal(conveyor.start)
68
69 ;On remet le timer à 0 au passage du detector 2
70 wait sig(part.det.R2)
71 timer 1 = 0
72
73 ;Quand la pièce est au detector 1 on calcule la vitesse
74 wait sig(part.det.R1)
75 vitesse = 700/timer(1)
76 speed (vitesse*sqrt(2)) mm/s
77
78 ;On prend la pièce sur le tapis (à detector 1)
79 set grasp.loc.mov = shift(grasp.loc.stopped by 0, 150, 0)
80 openi
81 moves grasp.loc.mov
82 closei
83 departs 50
84
85 ;On pose la pièce en P3
86 appro P3,50
87 moves P3
88 openi
89 departs 100
90
91 move wait.loc
92
93 e
94
95 load pickconv.lc
96 load alltools.v2

```

Questions :

What are the reasons for your program to fail ?

It takes time for the robot to accelerate at its ordered speed. So the cylinder goes too far and the suction pad comes too late.

How to teach the grasp.loc.stopped point to the robot ?

After the execution of the program, when we have to take back the cylinder, we can use the remote control to reach the aimed position (grasp.loc.stopped)

Exercice 2

```
1 edit ex2
2 ;Matthieu LAMIELLE et Muruo WANG
3 |
4 ;Initialisation des outils
5 tool pen
6 openi
7 |
8 ;Initialisation position du robot
9 moves wait.loc
10 signal 5
11 |
12 ;inputs
13 glue.empty = 1009
14 glue.recharged = 1010
15 |
16 ;recharge le glue
17 reacti glue.empty, recharger, 50
18 |
19 |
20 appro board:A, 50
21 moves board:A
22 closei
23 |
24 moves board:B
25 break
26 |
27 moves board:C
28 break
29 |
30 moves board:D
31 break
32 |
33 moves board:A
34 break
35 |
36 departs 50
37 |
38 move wait.loc
39 |
40 set board = board:shift(null by 10,10,0)
41 |
42 openi
43 |
44 e
45 |
46 edit recharger
47 |
48 ;inputs
49 glue.recharged = 1010
50 |
51 ;On repère le point où on s'est interrompu
52 set interruption = here
53 set arrival = dest
54 openi
55 departs 50
56 ;On va recharger
57 appro recharge, 50
58 moves recharge
59 wait sig(glue.recharged)
60 departs 50
61 |
```

```

61
62 ;On revient à l'endroit où on s'était arrêté
63 appro interruption, 50
64 moves interruption
65 closei
66
67 ;On finit l'instruction a laquelle on s'était arrêté
68 moves arrival
69
70 e
71
72 load glue.lc
73 load alltools.v2

```

Questions :

1.Should the glue default event be handled synchronously?

We have to detect whether there is glue left or not during all the process even between two points, so we use the synchronous mode.

2.Which are the Val/V+ instructions that can be used?

We can use “reacti” inside our program or run a parallel program to check glue using execute/abort in the main function.

3.Which one will you choose and why?

“reacti” is adapted because we can stop and go back at the point we left after recharging.

4.What is the solution to make sure that the object for which a glue occurred is correctly glued along the whole trajectory?

We have to record both the position and destination of the tool when it runs out of glue, so we can go back to the exact location.

Exercice 3

```
1 edit ex3
2 ;Matthieu LAMIELLE et Muruo WANG
3
4 ;Tool
5 tool gripper
6
7 ;Initialisation
8 signal 5
9
10 ;Définir le frame
11 set Palette = frame(origin,pt.x,pt.xy,origin)
12
13 move wait.loc
14 openi
15
16 i=0
17 j=0
18
19 for i=0 to 2
20     for j=0 to 1
21
22         set P = shift(null by 82*i, 59*j, 0)
23         set drop.loc = shift(drop.loc by 0, 0, 15)
24
25         appro palette:P, 50
26         moves palette:P
27         closei
28         departs 50
29
30         appro drop.loc, 50
31         moves drop.loc
32         openi
33         depart 50
34     end
35 end
36 move wait.loc
37 e
38 load alltools.v2
39 load pallet.lc
.
```

Exercice 4

Code version 1 :

```
1 edit ex4V1
2 ;Matthieu LAMIELLE et Muruo WANG
3
4 speed 30 always
5 speed 100 mm/s always
6
7 tool null
8 stop.condition = 1009
9
10 ;On bouge manuellement le robot dans sa position de depart
11 set P = here
12
13 ;On definit la distance totale à parcourir
14 distancetot = 200
15
16 i = 0
17
18 ;On definit la longueur du déplacement intermediaire
19 smallld = 5
20
21 timer 1 = 0
22
23 do
24     set etape = shift(P by 0,0,-i*smallld)
25     moves etape
26     i = i+1
27 until(sig(stop.condition) or (i*smallld > distancetot))
28
29 vitesse = (i*smallld)/timer(1)
30 type "La vitesse moyenne est de ", vitesse
31
32 e
```


Code version 2 :

```
1 edit ex4V2
2 ;Matthieu LAMIELLE et Muruo WANG
3
4 ;Mouvement gardé
5 execute 1 parallele, -1
6
7 tool null
8 speed 30 always
9 speed 100 mm/s always
10
11 ;On bouge manuellement le robot dans sa position de depart
12 set P = here
13 distancetot = 200
14
15 timer 1 = 0
16 ;Decendre
17 set fin = shift(P by 0,0,-distancetot)
18 moves fin
19 break
20
21 vitesse = (distancetot)/timer(1)
22 type "La vitesse moyenne est de ", vitesse
23
24 abort 1
25
26 e
27
28 edit parallele
29
30 ;On definit une condition d'arrêt
31 stop.condition = 1009
32
33 if sig(stop.condition) then
34     brake
35 end
36
37 e
```

Which solution is the best ?

The parallel program works the best because it doesn't interrupt the calculation during each small displacement.

Why is the average speed always lower than the desired speed ?

For both program, the maximum speed isn't reached instantaneously so the average speed is lower than the maximum speed. In addition for the first program we had a condition to be tested for each displacement so the speed is slower than the improved program.