

Laboratory Exercise 10.2 – Report:

Lê Văn Duẩn - 20194508

1. Assignment 3

- Mã nguồn:

```
1 .eqv HEADING 0xffff8010      # integer: an angle between 0 and 359
2                               # 0 : North (up)
3                               # 90: East (right)
4                               # 180: South (down)
5                               # 270: West (left)
6 .eqv MOVING 0xffff8050       # boolean: whether or not to move
7 .eqv LEAVETRACK 0xffff8020   # boolean: (0 or non-0)
8                               # whether or not to leave a track
9 .eqv WHEREX 0xffff8030      # integer: current x-location of Marsbot
10
11 .eqv WHEREY 0xffff8040      # integer: current y-location of marsbot
12
13 .text
14 main:   addi $a0, $0, 180 # Marsbot rotates 180 * running and start running
15         jal ROTATE
16         nop
17         jal GO
18         nop
19 sleep:  addi $v0, $0, 32 # keep running by sleeping in 500ms
20         li $a0, 500
21         syscall
22 #-----
23 goline: addi $a0, $0, 90 # Marsbot rotates 90 * running and start running
24         jal ROTATE
25         nop
26 sleep1: addi $v0, $0, 32 # keep running by sleeping in 500ms
27         li $a0, 500
28         syscall
29 #-----
30 sleep2: jal TRACK
31         nop
32         addi $v0, $0, 32 # keep running by sleeping in 1200ms
33         li $a0, 1200
34         syscall
35
36         jal UNTRACK
37         nop
```

```

38 goDown1: jal TRACK
39         nop
40         addi $a0, $0, 135
41         jal ROTATE
42         nop
43 sleep3: addi $v0, $0, 32
44         li $a0, 1000
45         syscall
46
47         jal UNTRACK
48         nop
49 -----
50 goDown2: jal TRACK
51         nop
52         addi $a0, $0, 180
53         jal ROTATE
54         nop
55 sleep4: addi $v0, $0, 32
56         li $a0, 2000
57         syscall
58
59         jal UNTRACK
60         nop
61 -----
62 goDown3: jal TRACK
63         nop
64         addi $a0, $0, 225
65         jal ROTATE
66         nop
67 sleep5: addi $v0, $0, 32
68         li $a0, 1000
69         syscall
70         jal UNTRACK
71         nop
72 ..

```

```

72 #-----
73 goDown4: jal TRACK
74         nop
75         addi $a0, $0, 270
76         jal ROTATE
77         nop
78 sleep6: addi $v0, $0, 32
79         li $a0, 1200
80         syscall
81
82         jal UNTRACK
83         nop
84 #-----
85 goDown5: jal TRACK
86         nop
87         addi $a0, $0, 360
88         jal ROTATE
89         nop
90 sleep7: addi $v0, $0, 32
91         li $a0, 3390
92         syscall
93
94         jal UNTRACK
95         nop
96 #-----
97 goU1:    addi $a0, $0, 90
98         jal ROTATE
99         nop
100 sleepU1: addi $v0, $0, 32
101         li $a0, 3000
102         syscall
103 #-----
104 end_main: jal STOP
105         nop
106         j end

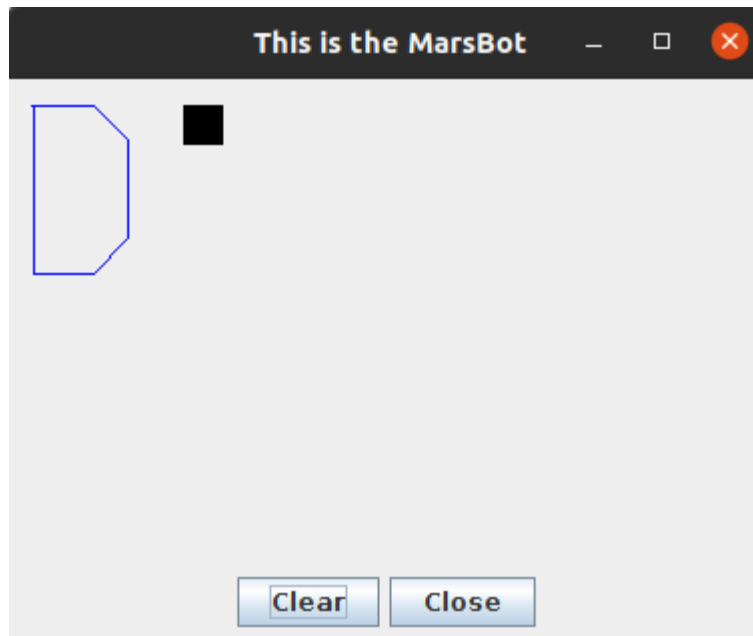
```

```

107 #-----
108 # GO procedure, to start running
109 # param[in] none
110 #-----
111 GO: li $at, MOVING # change MOVING port
112 addi $k0, $zero, 1 # to logic 1,
113 sb $k0, 0($at) # to start running
114 nop
115 jr $ra
116 nop
117
118 #-----
119 # STOP procedure, to stop running
120 # param[in] none
121 #-----
122 STOP: li $at, MOVING # change MOVING port to 0
123 sb $zero, 0($at) # to stop
124 nop
125 jr $ra
126 nop
127
128
129
130 #-----
131 # TRACK procedure, to start drawing line
132 # param[in] none
133 #-----
134 TRACK: li $at, LEAVETRACK # change LEAVETRACK port
135 addi $k0, $zero, 1 # to logic 1,
136 sb $k0, 0($at) # to start tracking
137 nop
138 jr $ra
139 nop
140
141 #-----
142 # UNTRACK procedure, to stop drawing line
143 # param[in] none
144 #-----
145 UNTRACK: li $at, LEAVETRACK # change LEAVETRACK port to 0
146 sb $zero, 0($at) # to stop drawing tail
147 nop
148 jr $ra
149 nop
150
151 #-----
152 # ROTATE procedure, to rotate the robot
153 # param[in] $a0, An angle between 0 and 359
154 # 0 : North (up)
155 # 90: East (right)
156 # 180: South (down)
157 # 270: West (left)
158 #-----
159
160
161
162 ROTATE: li $at, HEADING # change HEADING port
163 sw $a0, 0($at) # to rotate robot
164 nop
165 jr $ra
166 nop
167 end:

```

- Kết quả chạy mô phỏng:



- Giải thích:

Hàm ROTATE-> chỉnh góc của con trỏ để đi tiếp

Hàm STOP -> set giá trị của MOVING về 0 để dừng con trỏ

Hàm GO -> set MOVING về 1 và con trỏ sẽ di chuyển theo góc được chỉ định qua hàm ROTATE

Hàm TRACK -> set LEAVETRACK về 1 và bắt đầu vẽ

Hàm UNTRACK -> ngược lại với TRACK, set LEAVETRACK về 0 và dừng vẽ

Các nhãn loop để set thời gian con trỏ di chuyển theo hướng vừa được set up bằng hàm ROTATE để vẽ các đoạn theo ý muốn

Cách vẽ chữ D trong Duẩn:

+ di chuyển con trỏ đi xuống với góc 180 và đi sang phải với góc 90 những ko set hàm Track nên sẽ có khoảng cách với lề

+ vẽ đường ngang trên bằng cách tiếp tục với góc 90 nhưng sẽ dùng TRACK để vẽ

+ Ở đoạn cong chỉnh góc về 135 để nó di chuyển xuống dưới

+ Vẽ đường đi xuống bằng góc 180

- + tiếp tục vẽ cong vào bằng góc 225
- + vẽ nét ngang dưới đi sang trái với góc 270
- + cuối cùng là nét đi lên để hoàn thành bằng góc 360
- + thời gian sleep hay chiều dài của các nét đã được tính toán và điều chỉnh để chữ vẽ ra không bị lệch

2. Assignment 4

- Mã nguồn:

```

1  .eqv KEY_CODE 0xFFFF0004      # ASCII code from keyboard, 1 byte
2  .eqv KEY_READY 0xFFFF0000     # =1 if has a new keycode ?
3                                # Auto clear after lw
4  .eqv DISPLAY_CODE 0xFFFF000C  # ASCII code to show, 1 byte
5  .eqv DISPLAY_READY 0xFFFF0008 # =1 if the display has already to do
6                                # Auto clear after sw
7
8  .text
9      li $k0, KEY_CODE
10     li $k1, KEY_READY
11
12     li $s0, DISPLAY_CODE
13     li $s1, DISPLAY_READY
14
15 loop: nop
16 WaitForKey:
17     lw $t1, 0($k1)              # $t1 = [$k1] = KEY_READY
18     nop
19     beq $t1, $zero, WaitForKey  # if $t1 == 0 then Polling
20     nop
21
22     #-----
23 ReadKey:
24     lw $t0, 0($k0) # $t0 = [$k0] = KEY_CODE
25     nop
26
27     #-----
28 WaitForDis:
29     lw $t2, 0($s1)              # $t2 = [$s1] = DISPLAY_READY
30     beq $t2, $zero, WaitForDis  # if $t2 == 0 then Polling
31     nop
32
33     #-----
34 ShowKey:
35     add $t2, $t0, $0
36     addi $t0, $t0, 8
37     sw $t0, 0($s0) # show key

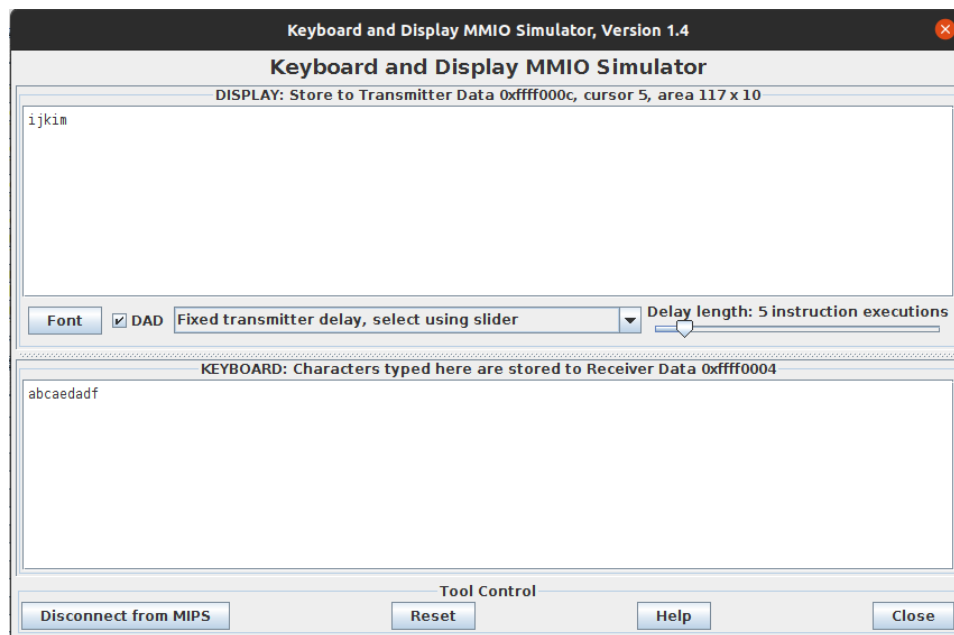
```

```

38 # Kiểm tra D
39 CheckD:
40     beq $t2, 'd', Exit_CL # Nếu là chu d thì dừng lại
41     beq $t2, 'D', Exit_CL # Nếu là chu D thì dừng lại
42     j loop
43 #-----
44
45 # Nếu là exit thì thoát chương trình
46 Exit_CL:
47     li $v0, 10
48     syscall

```

- Kết quả chạy:



- Giải thích:

Chương trình có 1 vòng loop nhận các kí tự đọc vào và hiển thị

Ở nhãn showKey chúng ta sẽ mã hóa để hiển thị ra màn hình bằng cách đưa giá trị cần show ra vào địa chỉ 0(\$s0), vì mssv là 20194508 nên mã hóa lệch 8 nên tăng giá trị \$t0 thêm 8 và lưu vào 0(\$s0)

Ở nhãn CheckD dùng để dừng chương trình khi nhập vào kí tự 'd' hoặc 'D' bằng cách kiểm tra \$t2 (là giá trị \$t0 lúc đọc vào).