

Laboratory Exercise 10

Đỗ Hải Dương – 20194528

Assignment 3

Code:

```
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|.eqv WHEREY 0xffff8040 # Integer: Current y-location of MarsBot
11|.text
12|main:
13|    #jal TRACK # draw track line
14|    addi $a0, $zero, 90 # Marsbot rotates 90* and start running
15|    jal ROTATE
16|    jal GO
17|    nop
18|
19|sleep1: addi $v0,$zero,32 # Keep running by sleeping in 5000 ms
20|    li $a0,5000
21|    syscall
22|    jal UNTRACK # keep old track
23|    nop
24|    jal TRACK # and draw new track line
25|
26|    nop
27|goRIGHTDOWN: addi $a0, $zero, 135 # Marsbot rotates 210*
28|    jal ROTATE
29|    nop
30|sleep2: addi $v0,$zero,32 # Keep running by sleeping in 5000 ms
31|    li $a0,1000
32|    syscall
33|    jal UNTRACK # keep old track
34|    nop
35|    jal TRACK # and draw new track line
36|    nop
37|
38|goDOWN: addi $a0, $zero, 180 # Marsbot rotates 210*
39|    jal ROTATE
40|    nop
41|sleep3: addi $v0,$zero,32 # Keep running by sleeping in 5000 ms
42|    li $a0,3000
43|    syscall
44|    jal UNTRACK # keep old track
45|    nop
46|    jal TRACK # and draw new track line
47|    nop
48|
```

```

49 goLEFTDOWN: addi $a0, $zero, 225 # Marsbot rotates 330*
50             jal ROTATE
51             nop
52 sleep4:     addi $v0,$zero,32 # Keep running by sleeping in 5000 ms
53             li $a0,1000
54             syscall
55             jal UNTRACK # keep old track
56             nop
57             jal TRACK # and draw new track line
58             nop
59
60 goLEFT:     addi $a0, $zero, 270 # Marsbot rotates 330*
61             jal ROTATE
62             nop
63 sleep5:     addi $v0,$zero,32 # Keep running by sleeping in 5000 ms
64             li $a0,2000
65             syscall
66             jal UNTRACK # keep old track
67             nop
68             jal TRACK # and draw new track line
69             nop
70
71 goUP:       addi $a0, $zero, 0 # Marsbot rotates 330*
72             jal ROTATE

```

```

73             nop
74 sleep6:     addi $v0,$zero,32 # Keep running by sleeping in 5000 ms
75             li $a0,4350
76             syscall
77             jal UNTRACK # keep old track
78             nop
79             jal TRACK # and draw new track line
80             nop
81
82 goRIGHT:
83             addi $a0, $zero, 90 # Marsbot rotates 330*
84             jal ROTATE
85             nop
86 sleep7:     addi $v0,$zero,32 # Keep running by sleeping in 5000 ms
87             li $a0,2000
88             syscall
89             jal UNTRACK # keep old track
90             nop
91             jal TRACK # and draw new track line
92             nop
93
94 goUP2:
95             addi $a0, $zero, 0 # Marsbot rotates 330*
96             jal ROTATE

```

```

97     nop
98 sleep8:
99     addi $v0,$zero,32 # Keep running by sleeping in 5000 ms
100     li $a0,2000
101     syscall
102     jal UNTRACK # keep old track
103     nop
104     jal TRACK # and draw new track line
105     nop
106
107 sleep:
108     addi $v0,$zero,32 # Keep running by sleeping in 1000 ms
109     li $a0,0
110     syscall
111     j STOP
112
113 end_main:
114 #-----
115 # GO procedure, to start running
116 # param[in] none
117 #-----
118 GO:
119     li $at, MOVING # change MOVING port
120     addi $k0, $zero,1 # to logic 1,

```

```

121     sb $k0, 0($at) # to start running
122     jr $ra
123     nop
124 #-----
125 # STOP procedure, to stop running
126 # param[in] none
127 #-----
128 STOP:
129     li $at, MOVING # change MOVING port to 0
130     sb $zero, 0($at) # to stop
131     jr $ra
132     nop
133 #-----
134 # TRACK procedure, to start drawing line
135 # param[in] none
136 #-----
137 TRACK:
138     li $at, LEAVETRACK # change LEAVETRACK port
139     addi $k0, $zero,1 # to logic 1,
140     sb $k0, 0($at) # to start tracking
141     jr $ra
142 #-----
143 # UNTRACK procedure, to stop drawing line
144 # param[in] none

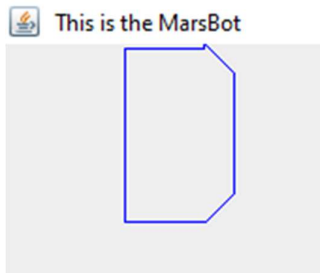
```

```

145 #-----
146 UNTRACK:
147     li $at, LEAVETRACK # change LEAVETRACK port to 0
148     sb $zero, 0($at) # to stop drawing tail
149     jr $ra
150     nop
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 ROTATE:
160     li $at, HEADING # change HEADING port
161     sw $a0, 0($at) # to rotate robot
162     jr $ra
163     nop
164

```

Kết quả:



Chữ D (tên Dương)

Assignment 4

Code:

```
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  .eqv e 0x65
8  .eqv x 0x78
9  .eqv i 0x69
10 .eqv t 0x74
11 .text
12     li $k0, KEY_CODE
13     li $k1, KEY_READY
14     li $s0, DISPLAY_CODE
15     li $s1, DISPLAY_READY
16 loop: nop
17 WaitForKey:
18     lw $t1, 0($k1)           # $t1 = [$k1] = KEY_READY
19     beq $t1, $zero, WaitForKey # if $t1 == 0 then Polling
20 ReadKey:
21     lw $t0, 0($k0)           # $t0 = [$k0] = KEY_CODE
22     j check_e
23 WaitForDis:
24     lw $t2, 0($s1)           # $t2 = [$s1] = DISPLAY_READY
25     beq $t2, $zero, WaitForDis # if $t2 == 0 then Polling
26 ShowKey:
27     sw $t0, 0($s0)           # show key
28     nop
29     j loop
30 check_e:
31     beq $t3, e, check_x      # if character e exist then check x
32     bne $t0, e, WaitForDis    # if character != e then continue
33     add $t3, $t0, $0          # else $t3 = 'e'
34     j WaitForDis
35
36 check_x:
37     beq $t4, x, check_i      # if character x exist then check i
38     bne $t0, x, reset         # if character != i then continue
39     add $t4, $t0, $0          # else $t4 = 'x'
40     j WaitForDis
41 check_i:
42
43     beq $t5, i, check_t      # if character i exist then check t
44     bne $t0, i, reset         # if character != t then continue
45     add $t5, $t0, $0          # else $t5 = 't'
46     j WaitForDis
47
48 check_t:
```

```

49      beq $t0, t, exit          # if character t exist then exit
50      j reset
51                                     # if character != t then continue
52 reset: li $t3, 0               # reset 'e' to 0
53        li $t4, 0               # reset 'x' to 0
54        li $t5, 0               # reset 'i' to 0
55        j WaitForDis
56
57 exit:  li $v0, 10
58        syscall
59

```

Kết quả:

