Báo cáo tuần 11

Bài 3:

Mã nguồn:

```
------<u>-----</u>
1 .eqv HEADING Oxffff8010 # Integer: An angle between 0 and 359
 2
                                # 0 : North (up)
                                 # 90: East (right)
3
                                 # 180: South (down)
 4
 5
                                 # 270: West (left)
 6 .eqv MOVING 0xffff8050
                                 # Boolean: whether or not to move
7
   .eqv LEAVETRACK 0xffff8020
                                 # Boolean (0 or non-0):
                                 # 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: jal goDOWN
13
        nop
          jal ROTATE
14
15
          nop
          jal GO
16
17
          nop
18
           jal sleep8000
19
          nop
           jal goRIGHT
20
21
          nop
            jal sleep2000
22
23
            nop
            jal TRACK
24
25
            nop
26
            jal sleep2000
27
            nop
28
            jal goUP
29
            nop
            jal sleep1000
30
31
            nop
            jal goLEFT
32
33
            nop
            jal sleep1000
34
            nop
35
36
            jal goDOWN
37
            nop
            jal sleep8000
38
39
            nop
40
            jal goLEFT
41
            nop
42
            jal sleep1000
```

```
43
              nop
              jal goUP
44
45
              nop
              jal sleep4000
46
47
              nop
48
              jal goRIGHT
49
              nop
50
              jal sleep8000
51
              nop
              jal goUP
52
53
              nop
              jal sleep4000
54
55
              nop
              jal goLEFT
56
57
              nop
              jal sleep1000
58
59
              nop
              jal goDOWN
60
61
              nop
              jal sleep8000
62
63
              nop
64
            jal goRIGHT
65
            nop
            jal sleep1000
66
            nop
67
68
            jal STOP
            nop
69
70
    end main:
            addi $v0, $0, 10
71
72
            syscall
73
    goASKEW:addi $a0, $zero, 45
                                 # Marsbot rotates 45*
74
            add $s0, $0, $ra
75
76
            jal ROTATE
77
            nop
            add $ra, $s0, $0
78
79
            jr $ra
80
            nop
            addi $a0, $zero, 0
                                    # Marsbot rotates 0*
81
   goUP:
82
            add $s0, $0, $ra
            jal ROTATE
83
84
            nop
```

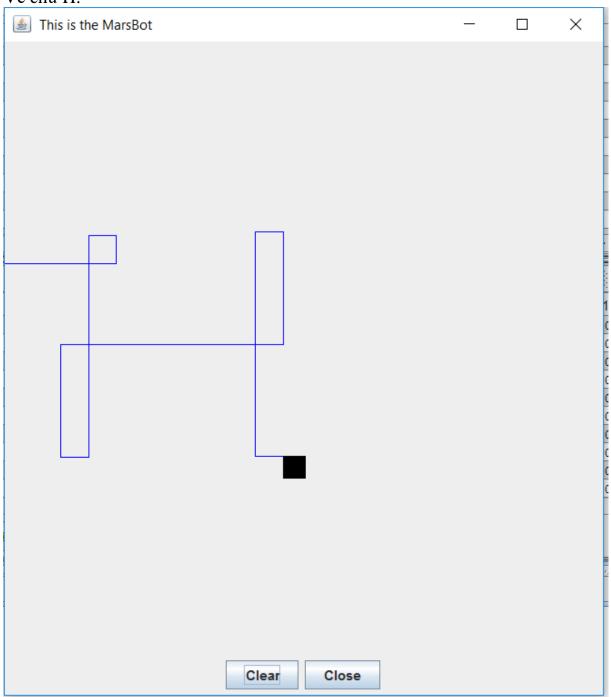
```
85
              add $ra, $s0, $0
              jr $ra
 86
 87
              nop
     goDOWN: addi $a0, $zero, 180
                                         # Marsbot rotates 180*
 88
              add $s0, $0, $ra
 89
              jal ROTATE
 90
 91
              nop
 92
              add $ra, $s0, $0
 93
              jr $ra
 94
              nop
     goRIGHT: addi $a0, $zero, 90
                                      # Marsbot rotates 90*
 95
              add $s0, $0, $ra
 96
 97
              jal ROTATE
 98
              nop
              add $ra, $s0, $0
 99
              jr $ra
100
101
              nop
102 goLEFT: addi $a0, $zero, 270
                                         # Marsbot rotates 270*
              add $s0, $0, $ra
103
              jal ROTATE
104
105
              nop
106
            add $ra, $s0, $0
107
            jr $ra
108
            nop
109
110 sleep1000: addi $v0,$zero,32
                                # Keep running by sleeping in 1000 ms
111
            li $a0,1000
112
            syscall
            add $s0, $0, $ra
113
                                  # keep old track
114
            jal UNTRACK
115
            nop
            jal TRACK
                                  # and draw new track line
116
            nop
117
            add $ra, $s0, $0
118
            jr $ra
119
120
            nop
121 sleep4000:
122
            addi $v0,$zero,32
                                  # Keep running by sleeping in 1000 ms
123
            li $a0,4000
124
            syscall
            add $s0, $0, $ra
125
            jal UNTRACK
126
                                  # keep old track
```

```
127
          nop
          jal TRACK
                            # and draw new track line
128
129
          nop
130
          add $ra, $s0, $0
131
          jr $ra
132
          nop
133
         sleep2000:
          addi $v0,$zero,32  # Keep running by sleeping in 1000 ms
134
135
         li $a0,2000
136
          syscall
          add $s0, $0, $ra
137
                            # keep old track
138
          jal UNTRACK
139
          nop
                            # and draw new track line
          jal TRACK
140
141
          nop
          add $ra, $s0, $0
142
143
          jr $ra
144
         nop
               addi $v0,$zero,32 # Keep running by sleeping in 2000 ms
145 sleep8000:
146
    li $a0,8000
147
         syscall
148
         add $s0, $0, $ra
          jal UNTRACK
                               # keep old track
149
150
          nop
          jal TRACK
                                # and draw new track line
151
152
          nop
          add $ra, $s0, $0
153
154
          jr $ra
155
          nop
156 #-----
157 # GO procedure, to start running
158 # param[in] none
159 #-----
160 GO: li $at, MOVING # change MOVING port
161 addi $k0, $zero,1 # to logic 1,

20 Show the Start running
          sb $k0, 0($at)
                               # to start running
162
          nop
163
164
          jr $ra
165
          nop
166 #-----
167 # STOP procedure, to stop running
168 # param[in] none
```

```
169 #-----
170 STOP: li $at, MOVING # change MOVING port to 0
171 sb $zero, 0($at) # to stop
172 nop
173
          jr $ra
174
175 #-----
176 # TRACK procedure, to start drawing line
177 # param[in] none
178 #-----
179 TRACK: li $at, LEAVETRACK  # change LEAVETRACK port
180 addi $k0, $zero, 1  # to logic 1,
181 sb $k0, 0($at)  # to start tracking
         sb $k0, 0($at)
         nop
jr $ra
nop
182
183
184
185 #-----
186 # UNTRACK procedure, to stop drawing line
187 # param[in] none
189 UNTRACK:li $at, LEAVETRACK # change LEAVETRACK port to 0
      sb $zero, 0($at) # to stop drawing tail
190
191
             nop
192
             jr $ra
193
            nop
194 #-----
195 # ROTATE procedure, to rotate the robot
196 # param[in] $a0, An angle between 0 and 359
197 # 0 : North (up)
198 # 90: East (right)
199 # 180: South (down)
200 # 270: West (left)
201 #----
202 ROTATE: li $at, HEADING # change HEADING port
203
            sw $a0, 0($at) # to rotate robot
204
             nop
            jr $ra
205
206
             nop
207
```

Vẽ chữ H:

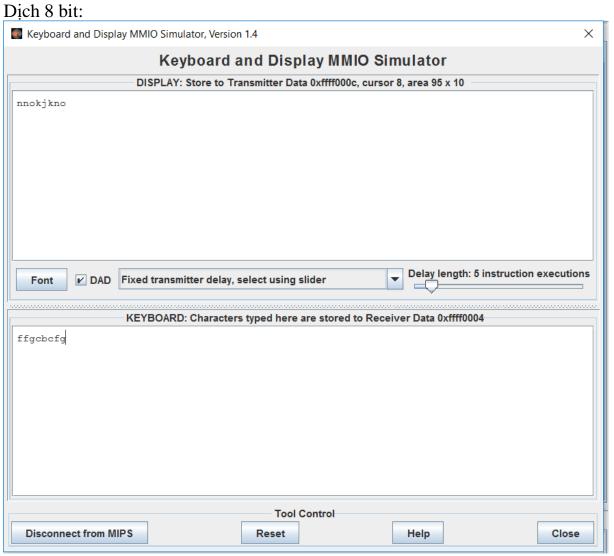


Bài 4:

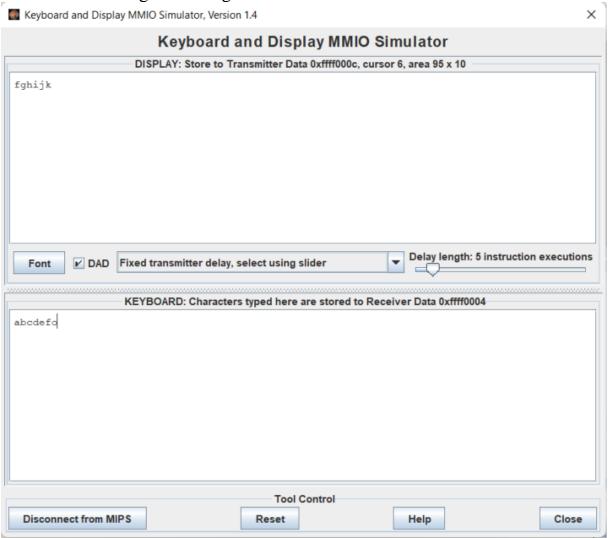
Mã nguồn:

```
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 .text
8 li $k0, KEY_CODE
 9
    li $k1, KEY READY
10
11 li $s0, DISPLAY CODE
   li $s1, DISPLAY_READY
12
13 loop: nop
14
15 WaitForKey: lw $t1, 0($k1) # $t1 = [$k1] = KEY READY
16 nop
17 beq $t1, $zero, WaitForKey # if $t1 == 0 then Polling
18 nop
19 #-----
20 ReadKey: lw $t0, 0($k0) # $t0 = [$k0] = KEY CODE
21 nop
22 #-----
23 WaitForDis: lw $t2, 0($s1) # $t2 = [$s1] = DISPLAY READY
24 nop
25 beg $t2, $zero, WaitForDis # if $t2 == 0 then Polling
26 nop
27
28 Encrypt: addi $t0, $t0, 8 # change input key
   beq $t0, 104, quit # go 'o' de thoat. 104 = 'h'
   beq, $t0, 'H', quit
31
32
33
34 ShowKey: sw $t0, 0($s0) # show key
35 nop
36 #-----
37 j loop
38 nop
39
40 quit:
41
```

• Kết quả:



Bấm 'o' thì chương trình dừng:



• Giải thích:

- o Dòng 8: Gán k0 là key_code
- o Dòng 9: gán k1 là key read
- o Dòng 11: gán s0 là display_code
- O Dòng 12: gán s1 là display ready
- o Dòng 15: nhãn WaitForKey: đọc ký tự key_ready vào t1
- o Dòng 17: nếu t1 = 0 thì chờ ký tự tiếp theo
- o Dòng 20: nhãn ReadKey: đọc ký tự nhập từ bàn phím vào t0
- O Dòng 23->26: nhãn WaitForDis: hiển thị ra màn hình
- Obng 28->30: nhãn Encrypt: để mã hóa dịch. Ký tự nhập vào sẽ được dịch đi 5. Ví dụ nhập '1' thành '6'. Khi nhập đến ký tự 'o' (mã ASCII là 116) hoặc 'O'. Tức sau khi mã hóa thành 't' hoặc 'T' thì nhảy đến 'quit' để thoát chương trình.
- O Dòng 34->38: lưu t0 vào s0 để hiện thị
- o Dòng 39: nhãn quit: thoát chương trình