

Báo cáo tuần 11

Bài 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 .eqv WHEREY 0xffff8040     # Integer: Current y-location of MarsBot
11 .text
12 main:    jal goDOWN
13          nop
14          jal ROTATE
15          nop
16          jal GO
17          nop
18          jal sleep8000
19          nop
20          jal goRIGHT
21          nop
22          jal sleep2000
23          nop
24          jal TRACK
25          nop
26          jal sleep2000
27          nop
28          jal goUP
29          nop
30          jal sleep1000
31          nop
32          jal goLEFT
33          nop
34          jal sleep1000
35          nop
36          jal goDOWN
37          nop
38          jal sleep8000
39          nop
40          jal goLEFT
41          nop
42          jal sleep1000
```

```

43      nop
44      jal goUP
45      nop
46      jal sleep4000
47      nop
48      jal goRIGHT
49      nop
50      jal sleep8000
51      nop
52      jal goUP
53      nop
54      jal sleep4000
55      nop
56      jal goLEFT
57      nop
58      jal sleep1000
59      nop
60      jal goDOWN
61      nop
62      jal sleep8000
63      nop

```

```

64      jal goRIGHT
65      nop
66      jal sleep1000
67      nop
68      jal STOP
69      nop
70  end_main:
71      addi $v0, $0, 10
72      syscall
73
74  goASKEW: addi $a0, $zero, 45      # Marsbot rotates 45*
75          add $s0, $0, $ra
76          jal ROTATE
77          nop
78          add $ra, $s0, $0
79          jr $ra
80          nop
81  goUP:   addi $a0, $zero, 0        # Marsbot rotates 0*
82          add $s0, $0, $ra
83          jal ROTATE
84          nop

```

```

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

```

```

127         nop
128         jal TRACK                # and draw new track line
129         nop
130         add $ra, $s0, $0
131         jr $ra
132         nop
133         sleep2000:
134         addi $v0,$zero,32        # Keep running by sleeping in 1000 ms
135         li $a0,2000
136         syscall
137         add $s0, $0, $ra
138         jal UNTRACK             # keep old track
139         nop
140         jal TRACK                # and draw new track line
141         nop
142         add $ra, $s0, $0
143         jr $ra
144         nop
145     sleep8000:        addi $v0,$zero,32        # Keep running by sleeping in 2000 ms
146         li $a0,8000
147         syscall
148         add $s0, $0, $ra
149         jal UNTRACK             # keep old track
150         nop
151         jal TRACK                # and draw new track line
152         nop
153         add $ra, $s0, $0
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,
162         sb $k0, 0($at)        # to start running
163         nop
164         jr $ra
165         nop
166     #-----
167     # STOP procedure, to stop running
168     # param[in] none

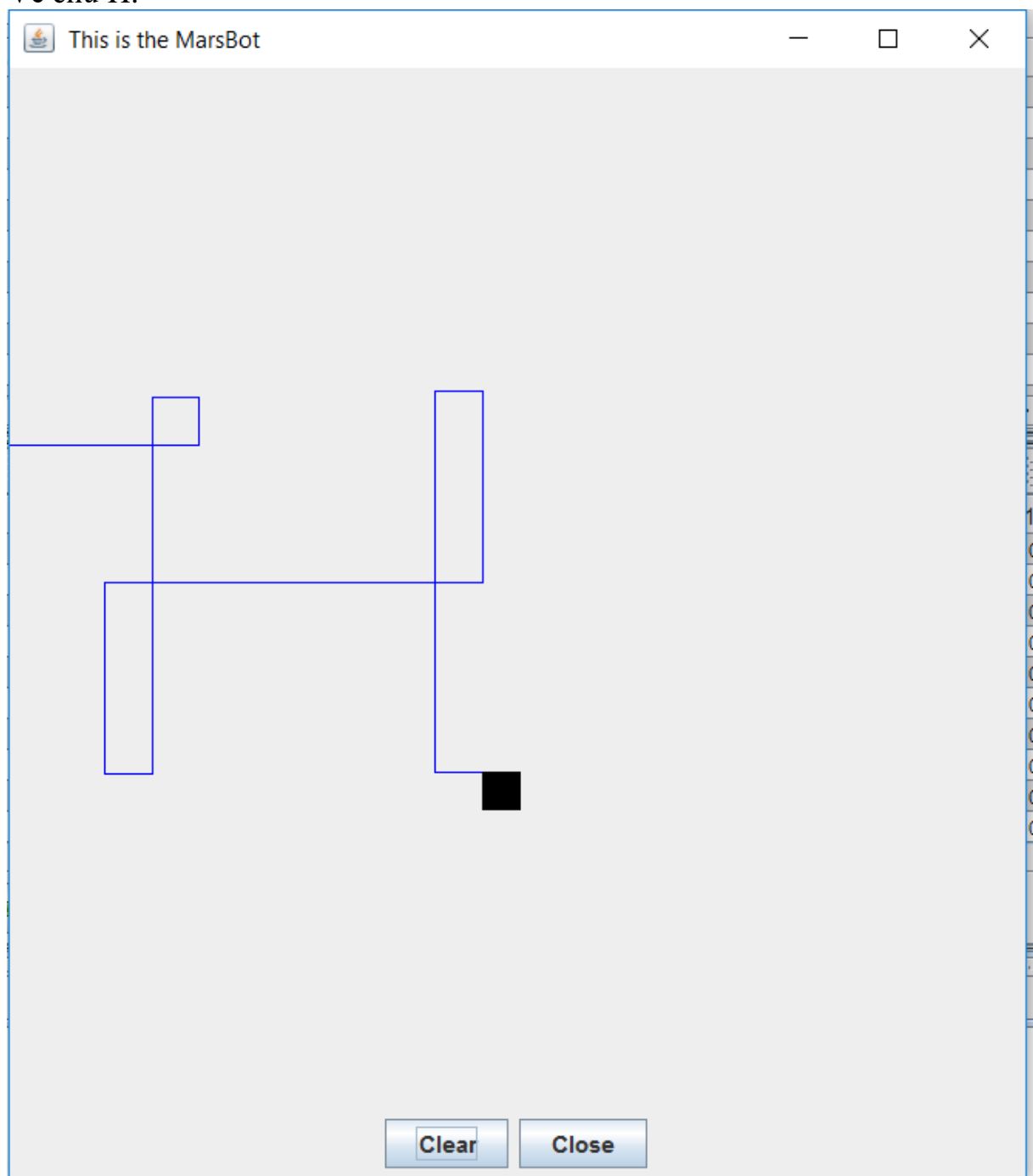
```

```

169 #-----
170 STOP:  li $at, MOVING      # change MOVING port to 0
171        sb $zero, 0($at)   # to stop
172        nop
173        jr $ra
174        nop
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
182        nop
183        jr $ra
184        nop
185 #-----
186 # UNTRACK procedure, to stop drawing line
187 # param[in] none
188 #-----
189 UNTRACK:li $at, LEAVETRACK  # change LEAVETRACK port to 0
190        sb $zero, 0($at)   # to stop drawing tail
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
205        jr $ra
206        nop
207

```

Vẽ chữ H:

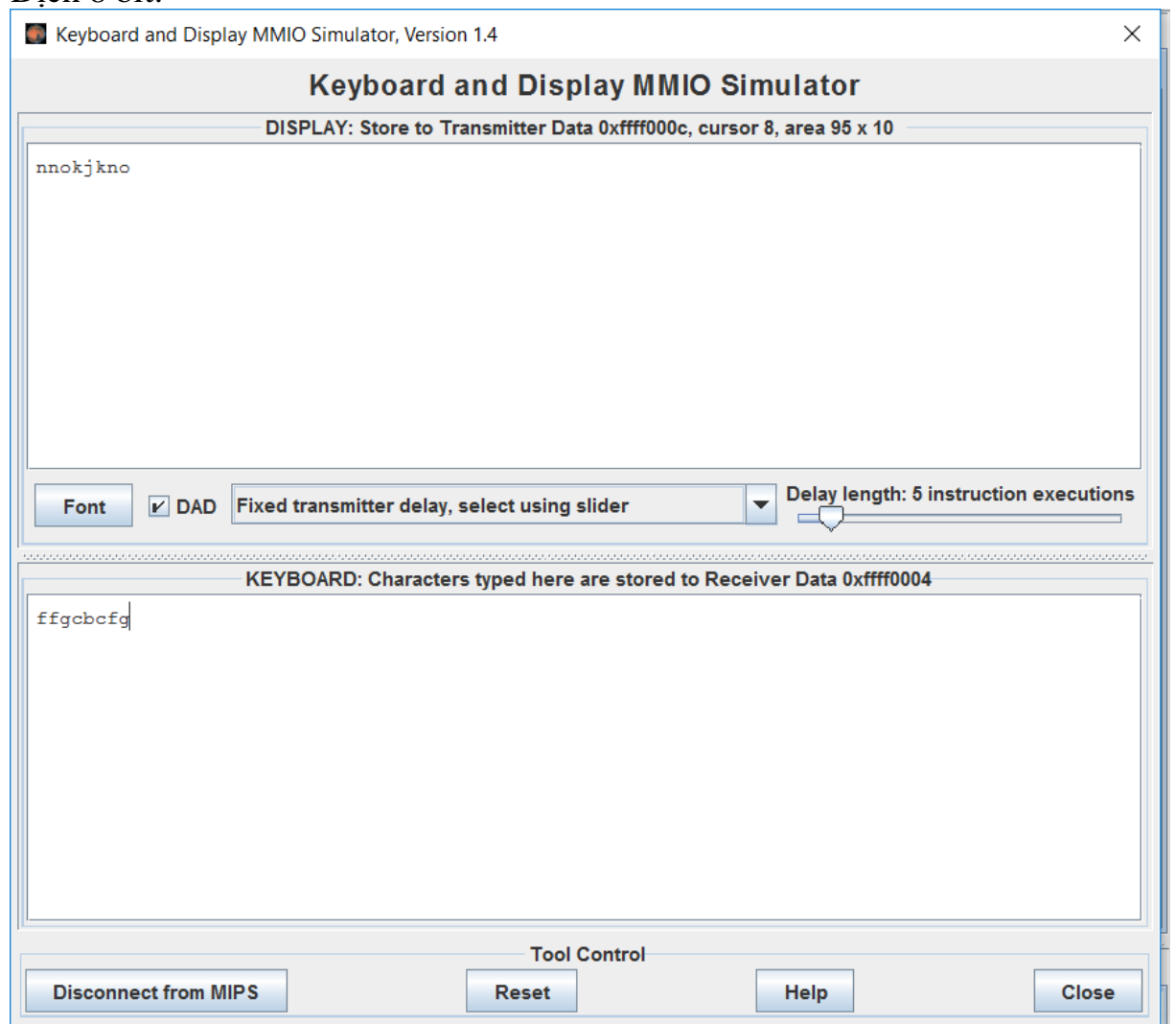


Bài 4:

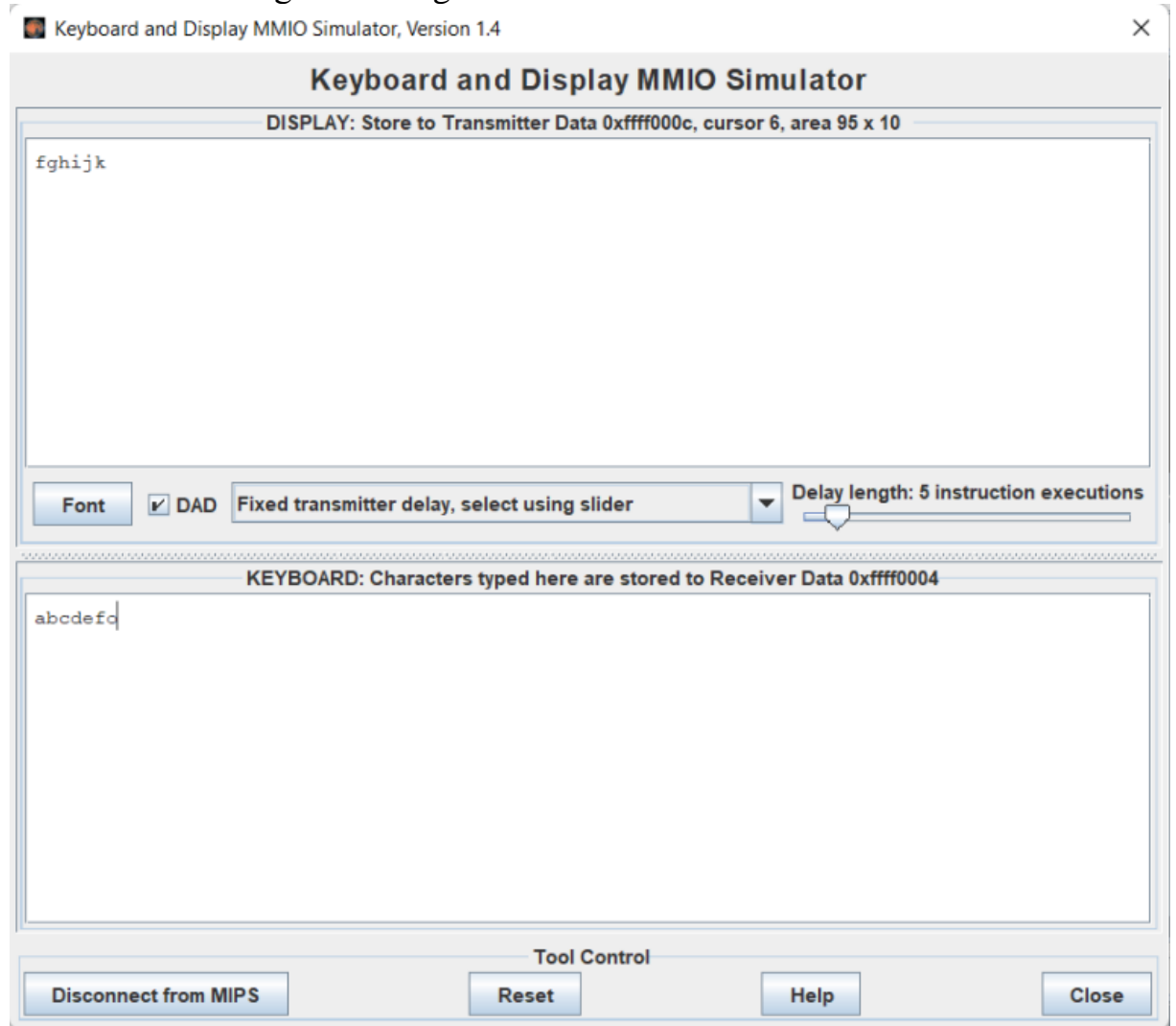
- Mã nguồn:

```
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
12  li $s1, DISPLAY_READY
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  beq $t2, $zero, WaitForDis # if $t2 == 0 then Polling
26  nop
27  #-----
28  Encrypt: addi $t0, $t0, 8 # change input key
29  beq $t0, 104, quit # go 'o' de thoat. 104 = 'h'
30  beq, $t0, 'H', quit
31  nop
32
33  #-----
34  ShowKey: sw $t0, 0($s0) # show key
35  nop
36  #-----
37  j loop
38  nop
39
40  quit:
41
```

- Kết quả:
Dịch 8 bit:



Bấm ‘o’ thì chương trình dừng:



- Giải thích:
 - Dòng 8: Gán k0 là key_code
 - Dòng 9: gán k1 là key_read
 - Dòng 11: gán s0 là display_code
 - Dòng 12: gán s1 là display_ready
 - Dòng 15: nhấn WaitForKey: đọc ký tự key_ready vào t1
 - Dòng 17: nếu t1 = 0 thì chờ ký tự tiếp theo
 - Dòng 20: nhấn ReadKey: đọc ký tự nhập từ bàn phím vào t0
 - Dòng 23->26: nhấn WaitForDis: hiển thị ra màn hình
 - Dòng 28->30: nhấn Encrypt: để mã hóa dịch. Ký tự nhập vào sẽ được dịch đi 5 . Ví dụ nhập ‘l’ 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.
 - Dòng 34->38: lưu t0 vào s0 để hiển thị
 - Dòng 39: nhấn quit: thoát chương trình