Phạm Minh Thống -20205029

Báo cáo thực hành kiến trúc máy tính tuần 11

ASM 3:

```
.eqv HEADING 0xffff8010 # Integer: An angle between 0 and 359
 # 0 : North (up)
# 90: East (right)
 # 180: South (down)
 # 270: West (left)
.eqv MOVING 0xffff8050 # Boolean: whether or not to move
.eqv LEAVETRACK 0xffff8020 # Boolean (0 or non-0):
 # whether or not to leave a track
.eqv WHEREX 0xffff8030 # Integer: Current x-location of MarsBot
.eqv WHEREY 0xffff8040 # Integer: Current y-location of MarsBot

.text

main:
goRight_1:
        jal UNTRACK # draw track line
        nop
        addi $a0, $zero, 90 # Marsbot rotates 90* and start running
        jal ROTATE
        nop
        jal GO
        nop

sleep1:
        addi $v0,$zero,32 # Keep running by sleeping in 1000 ms
        li $a0,1000
        syscall


goDOWN:
        addi $a0, $zero, 180 # Marsbot rotates 180*
        jal ROTATE
        nop

sleep2:
        addi $v0,$zero,32 # Keep running by sleeping in 2000 ms
        li $a0,2000
        syscall

        jal UNTRACK # keep old track
        nop
        jal TRACK # and draw new track line
        nop

goRight_2:
        jal TRACK # draw track line
        nop
        addi $a0, $zero, 90 # Marsbot rotates 90* and start running
        jal ROTATE
        nop
        jal GO
        nop
```

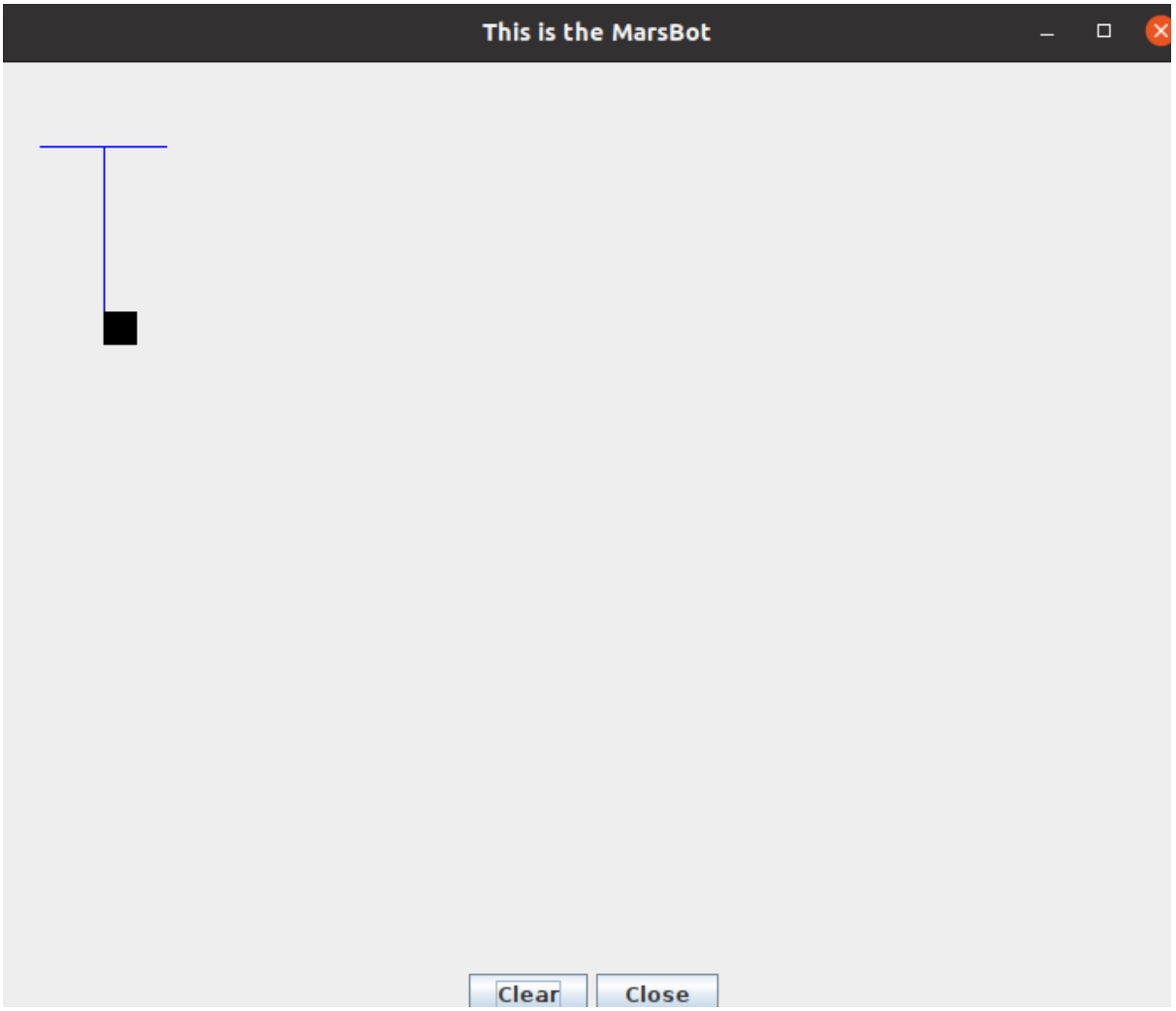
```
54  sleep3:
55          addi $v0,$zero,32 # Keep running by sleeping in 1000 ms
56          li $a0,3000
57          syscall
58
59          jal UNTRACK # keep old track
60          nop
61          jal TRACK # and draw new track line
62          nop
63
64
65  goLeft:
66          addi $a0, $zero, 270 # Marsbot rotates 210*
67          jal ROTATE
68          nop
69
70  sleep4:
71          addi $v0,$zero,32 # Keep running by sleeping in 1000 ms
72          li $a0,1500
73          syscall
74
75          jal UNTRACK # keep old track
76          nop
77          jal TRACK # and draw new track line
78          nop
79
80  goDOWN_2:
81          addi $a0, $zero, 180 # Marsbot rotates 180*
82          jal ROTATE
83          nop
84
85  sleep5:
86          addi $v0,$zero,32 # Keep running by sleeping in 2000 ms
87          li $a0,4000
88          syscall
89
90          jal UNTRACK # keep old track
91          nop
92          jal TRACK # and draw new track line
93          nop
94
95          jal STOP
96
97  end_main:
98          li $v0, 10
99          syscall
100
```

```
105 GO:
106         li $at, MOVING # change MOVING port
107         addi $k0, $zero,1 # to logic 1,
108         sb $k0, 0($at) # to start running
109         nop
110         jr $ra
111         nop
112 #---------------------------------------------------------
113 # STOP procedure, to stop running
114 # param[in] none
115 #---------------------------------------------------------
116 STOP:
117         li $at, MOVING # change MOVING port to 0
118         sb $zero, 0($at) # to stop
119         nop
120         jr $ra
121         nop
122 #---------------------------------------------------------
123 # TRACK procedure, to start drawing line
124 # param[in] none
125 #---------------------------------------------------------
126 TRACK:
127         li $at, LEAVETRACK # change LEAVETRACK port
128         addi $k0, $zero,1 # to logic 1,
129         sb $k0, 0($at) # to start tracking
130         nop
131         jr $ra
132         nop
133 #---------------------------------------------------------
134 # UNTRACK procedure, to stop drawing line
135 # param[in] none
136 #---------------------------------------------------------
```

```
137 UNTRACK:
138         li $at, LEAVETRACK # change LEAVETRACK port to 0
139         sb $zero, 0($at) # to stop drawing tail
140         nop
141         jr $ra
142         nop
143 #---------------------------------------------------------
144 # ROTATE procedure, to rotate the robot
145 # param[in] $a0, An angle between 0 and 359
146 # 0 : North (up)
147 # 90: East (right)
148 # 180: South (down)
149 # 270: West (left)
150 #---------------------------------------------------------
151 ROTATE:
152         li $at, HEADING # change HEADING port
153         sw $a0, 0($at) # to rotate robot
154         nop
155         jr $ra
156         nop
```

- Em thực hiện vẽ sang phải (2s) và xuống (1s) không lưu vết
- Sau đó vẽ chữ T bằng 3 đường :
    + Vẽ sang phải (quay góc 90 độ) trong 3s
    + Vẽ sang trái 1,5 s (quay góc 270 đô)
    + Vẽ xuống 4s (quay góc 180 độ)

Đây là kết quả



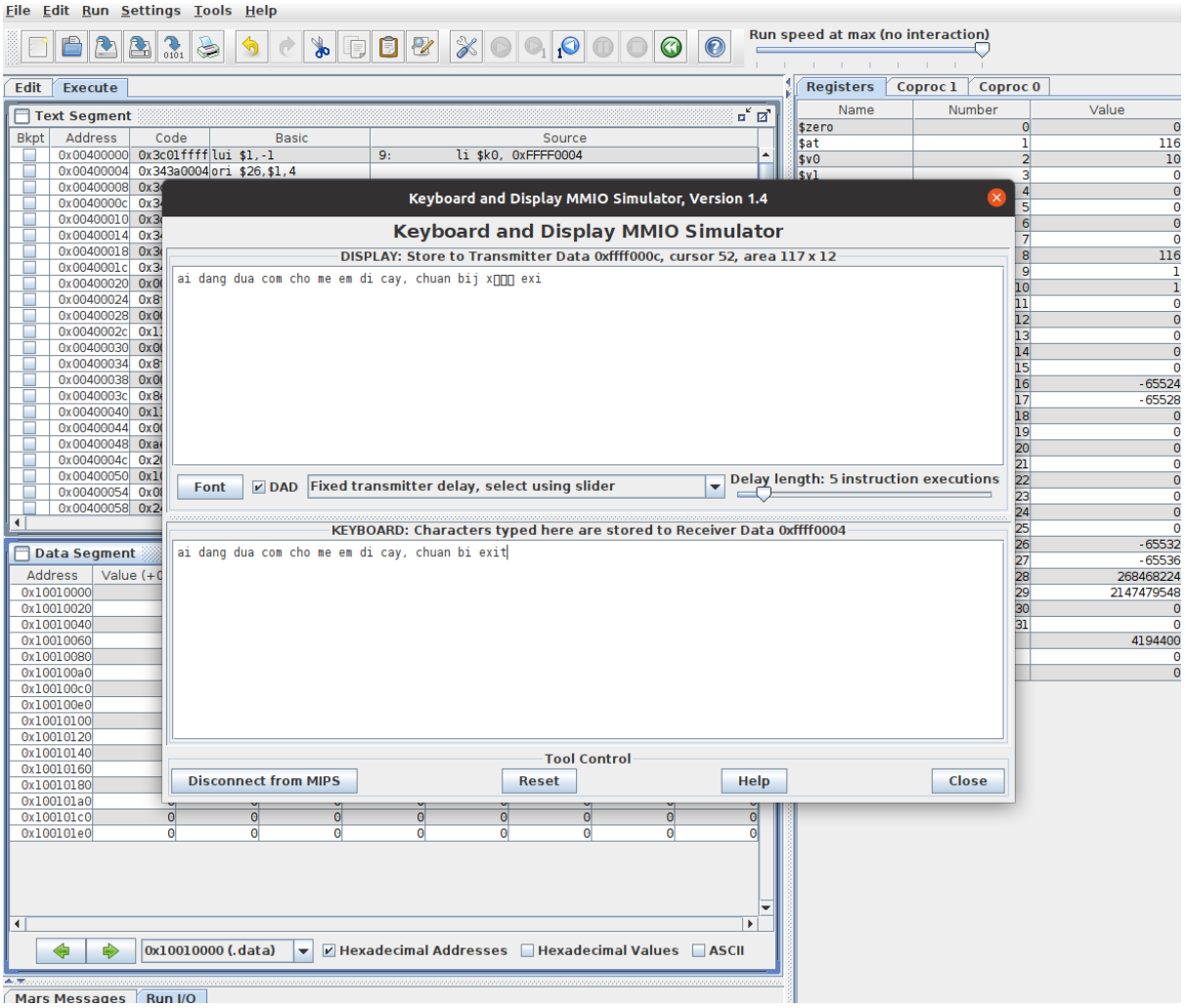
ASM 4:

```
.eqv KEY_CODE 0xFFFF0004 # ASCII code from keyboard, 1 byte
.eqv KEY_READY 0xFFFF0000 # =1 if has a new keycode ?
 # Auto clear after lw
.eqv DISPLAY_CODE 0xFFFF000C # ASCII code to show, 1 byte
.eqv DISPLAY_READY 0xFFFF0008 # =1 if the display has already to do
 # Auto clear after sw

.text
        li $k0, KEY_CODE
        li $k1, KEY_READY

        li $s0, DISPLAY_CODE
        li $s1, DISPLAY_READY


loop:
        nop

WaitForKey:
        lw $t1, 0($k1) # $t1 = [$k1] = KEY_READY
        nop
        beq $t1, $zero, WaitForKey # if $t1 == 0 then Polling
        nop

 #-----------------------------------------------------
ReadKey:
        lw $t0, 0($k0) # $t0 = [$k0] = KEY_CODE
        nop

 #-----------------------------------------------------
WaitForDis:
        lw $t2, 0($s1) # $t2 = [$s1] = DISPLAY_READY
        beq $t2, $zero, WaitForDis # if $t2 == 0 then Polling
        nop
#-----------------------------------------------------
ShowKey:
        sw $t0, 0($s0) # show key

# Kiem tra T
CheckT:
        beq $t0, 0x74, Exit_CL  # Neu la chu t thi dung lai
        beq $t0, 0x54, Exit_CL  # Neu la chu T thi dung lai
        j loop
 #-------------------------------------------------------------------------------------------------

# Nếu là exit thì thoát chương trình
Exit_CL:
        li $v0, 10
        syscall
```

=> Em thực hiện kiểm tra từng ký tự nhập vào

Các bước:

- In ra các kí tự được nhập vào
- Kiểm tra xem ký tự có phải chữ (t) hoặc (T) không
- Nếu phải thì dừng lại
- Nếu không phải thì tiếp tục


File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Text Segment
Bkpt Address Code Basic Source
0x00400000 0x3c01ffff lui $1,-1 9: li $k0, 0xFFFF0004
0x00400004 0x343a0004 ori $26,$1,4
Keyboard and Display MMIO Simulator, Version 1.4
Keyboard and Display MMIO Simulator
DISPLAY: Store to Transmitter Data 0xffff000c, cursor 52, area 117 x 12
ai dang dua com cho me em di cay, chuan bij x exi
Font
DAD
Fixed transmitter delay, select using slider
Delay length: 5 instruction executions
KEYBOARD: Characters typed here are stored to Receiver Data 0xffff0004
ai dang dua com cho me em di cay, chuan bi exit
Tool Control
Disconnect from MIPS
Reset
Help
Close
Registers Coproc 1 Coproc 0
Name Number Value
$zero 0 0
$at 1 116
$v0 2 10
Data Segment
Address Value (+0
0x10010000
0x10010020
0x10010040
0x10010060
0x10010080
0x100100a0
0x100100c0
0x100100e0
0x10010100
0x10010120
0x10010140
0x10010160
0x10010180
0x100101a0
0x100101c0 0 0 0 0 0 0 0 0
0x100101e0 0 0 0 0 0 0 0 0
0x10010000 (.data)
Hexadecimal Addresses
Hexadecimal Values
ASCII
Mars Messages
Run I/O