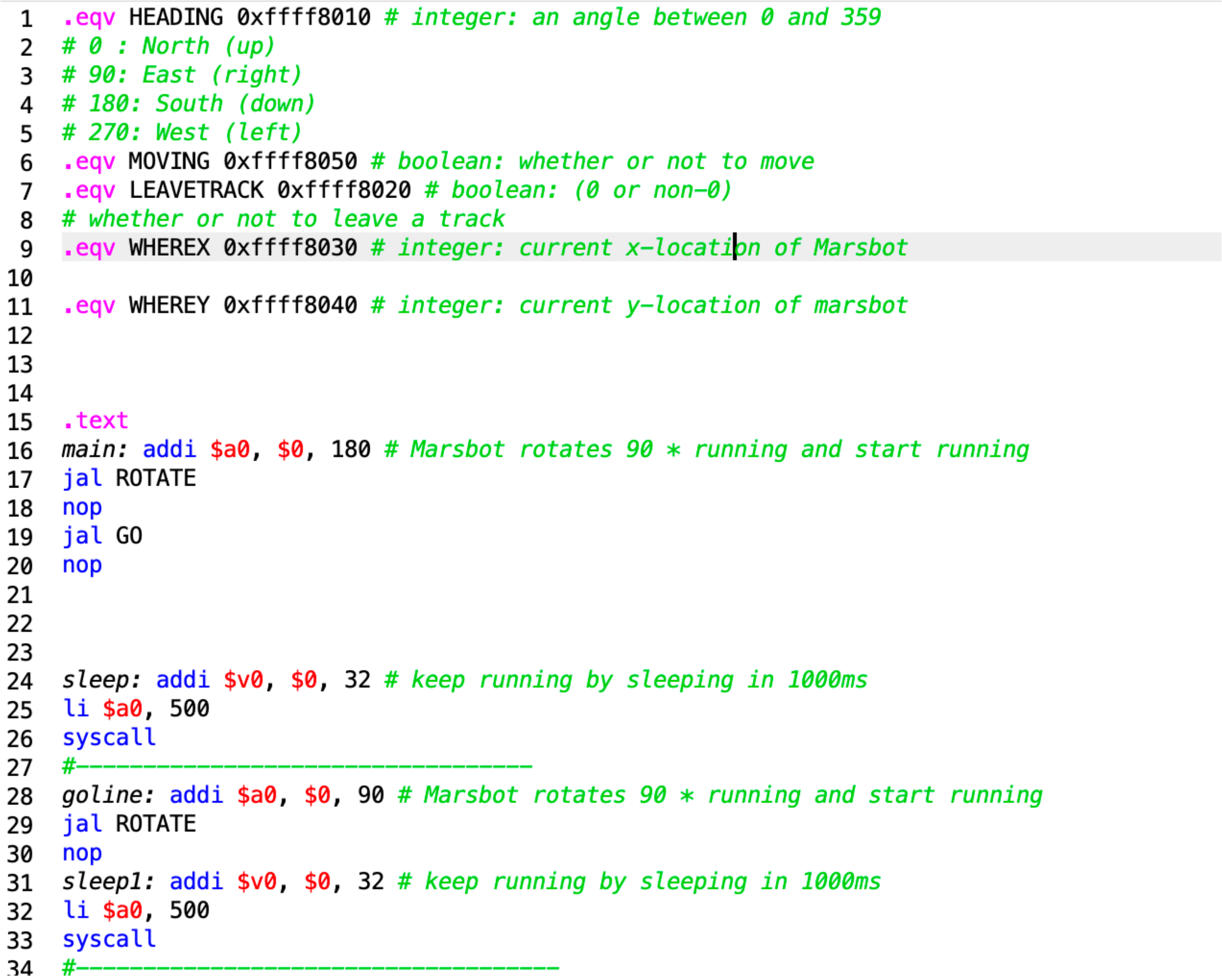
Laboratory Exercise 10.2 – Report:

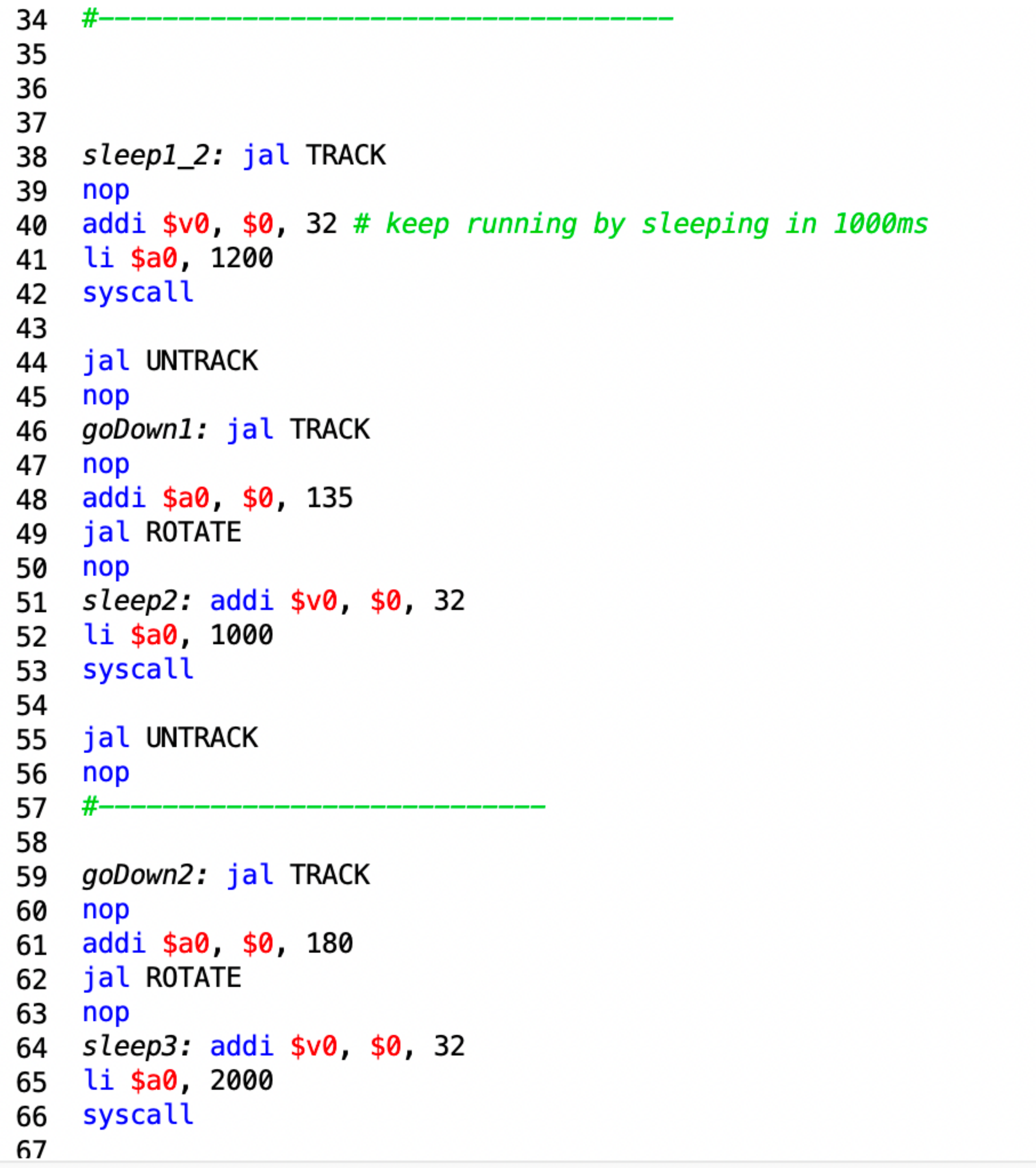
Nguyễn Hải Dương- 20194530

3. Assignment 3

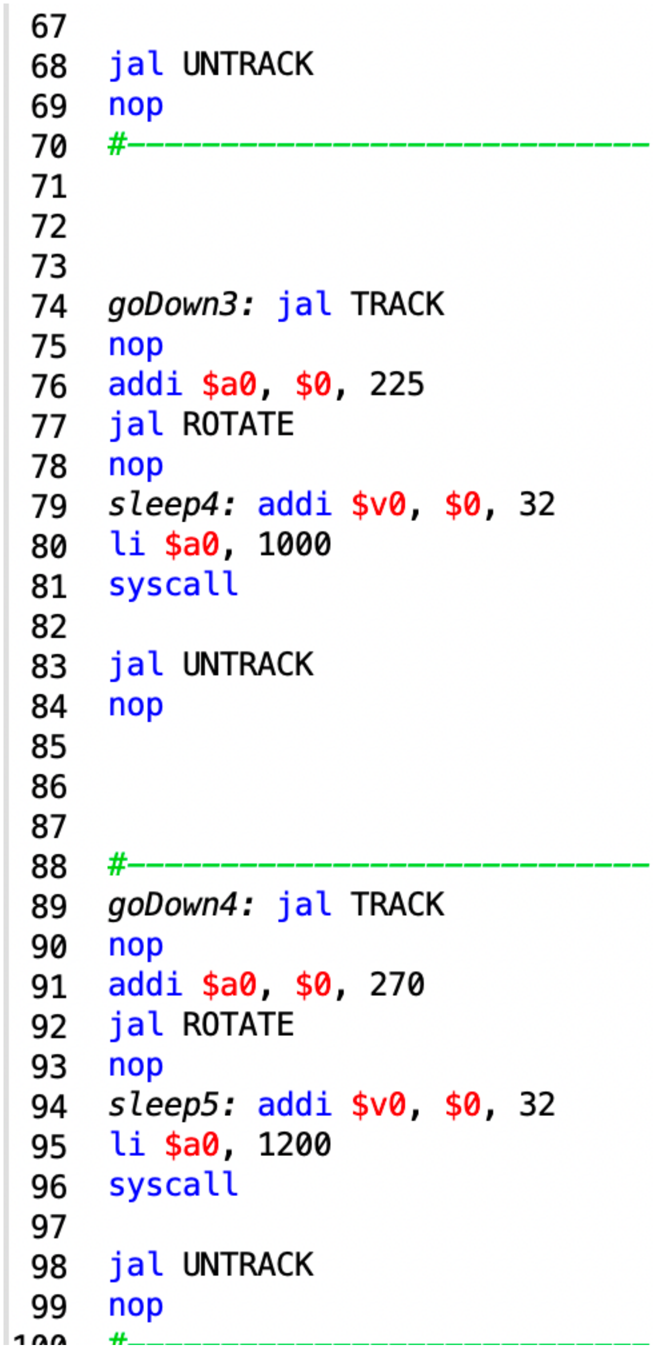
- Mã nguồn:



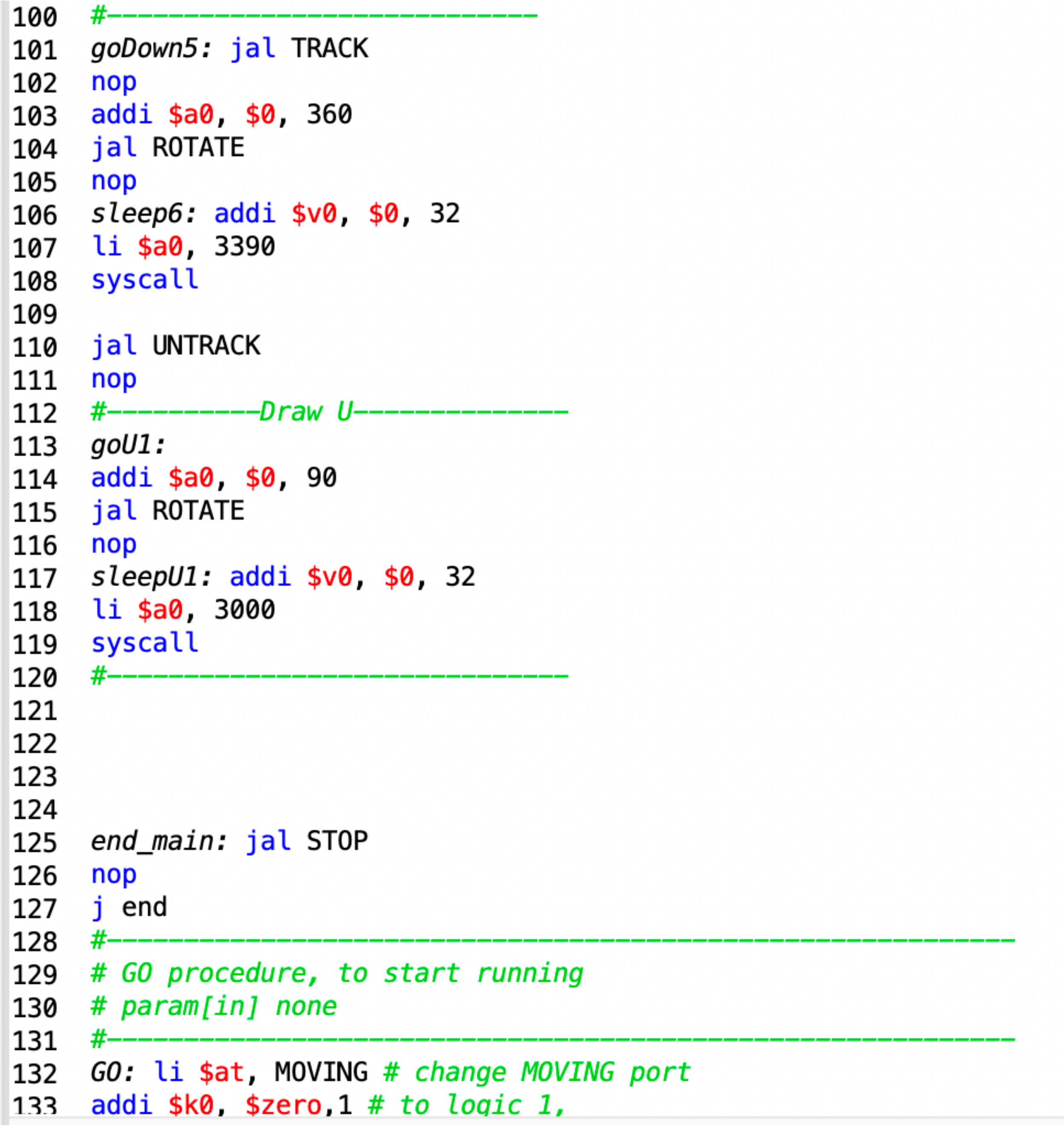
```
.eqv HEADING 0xffff8010 # integer: an angle between 0 and 359
# 0 : North (up)
# 90: East (right)
# 180: South (down)
# 270: West (left)
.eqv MOVING 0xffff8050 # boolean: whether or not to move
.eqv LEAVETRACK 0xffff8020 # boolean: (0 or non-0)
# whether or not to leave a track
.eqv WHEREX 0xffff8030 # integer: current x-location of Marsbot

.eqv WHEREY 0xffff8040 # integer: current y-location of marsbot



.text
main: addi $a0, $0, 180 # Marsbot rotates 90 * running and start running
jal ROTATE
nop
jal GO
nop



sleep: addi $v0, $0, 32 # keep running by sleeping in 1000ms
li $a0, 500
syscall
#-----------------------------------
goline: addi $a0, $0, 90 # Marsbot rotates 90 * running and start running
jal ROTATE
nop
sleep1: addi $v0, $0, 32 # keep running by sleeping in 1000ms
li $a0, 500
syscall
#-------------------------------------
```



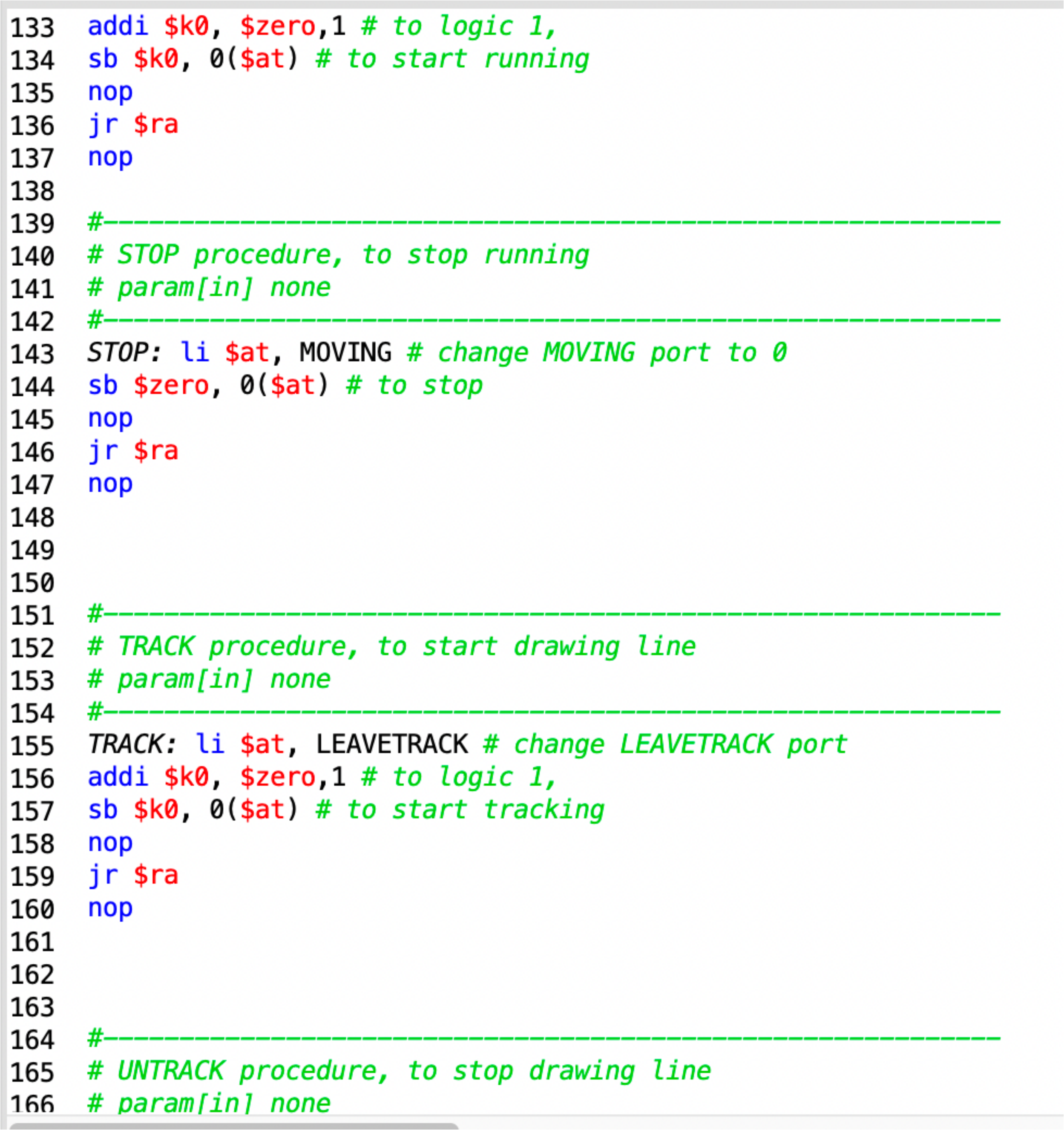
```
#-----------------------------------



sleep1_2: jal TRACK
nop
addi $v0, $0, 32 # keep running by sleeping in 1000ms
li $a0, 1200
syscall

jal UNTRACK
nop
goDown1: jal TRACK
nop
addi $a0, $0, 135
jal ROTATE
nop
sleep2: addi $v0, $0, 32
li $a0, 1000
syscall

jal UNTRACK
nop
#--------------------------

goDown2: jal TRACK
nop
addi $a0, $0, 180
jal ROTATE
nop
sleep3: addi $v0, $0, 32
li $a0, 2000
syscall

```



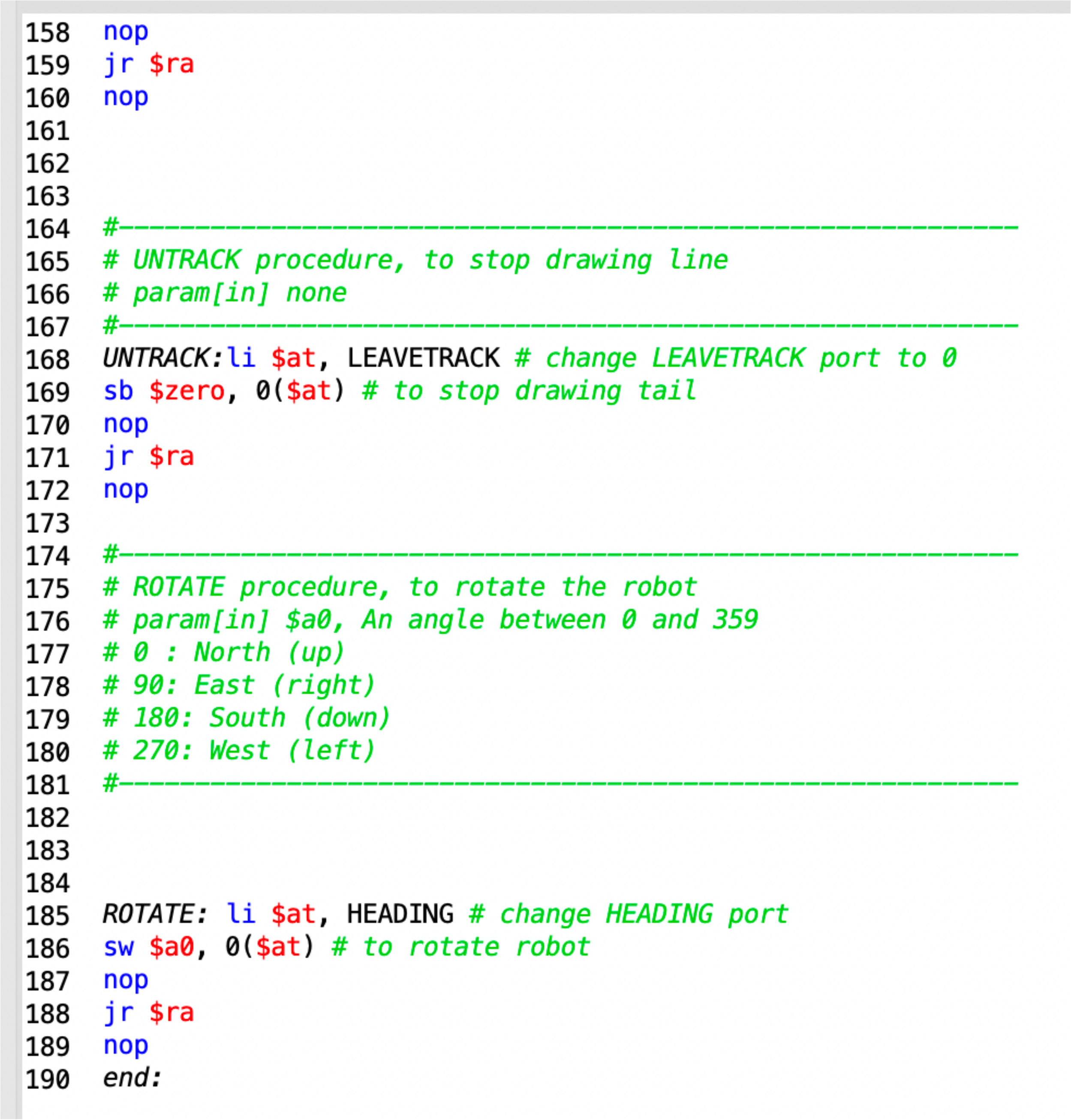
```

jal UNTRACK
nop
#----------------------------



goDown3: jal TRACK
nop
addi $a0, $0, 225
jal ROTATE
nop
sleep4: addi $v0, $0, 32
li $a0, 1000
syscall

jal UNTRACK
nop



#----------------------------
goDown4: jal TRACK
nop
addi $a0, $0, 270
jal ROTATE
nop
sleep5: addi $v0, $0, 32
li $a0, 1200
syscall

jal UNTRACK
nop
#----------------------------
```



```
#----------------------------
goDown5: jal TRACK
nop
addi $a0, $0, 360
jal ROTATE
nop
sleep6: addi $v0, $0, 32
li $a0, 3390
syscall

jal UNTRACK
nop
#----------Draw U--------------
goU1:
addi $a0, $0, 90
jal ROTATE
nop
sleepU1: addi $v0, $0, 32
li $a0, 3000
syscall
#------------------------------




end_main: jal STOP
nop
j end
#-----------------------------------------------------------
# GO procedure, to start running
# param[in] none
#-----------------------------------------------------------
GO: li $at, MOVING # change MOVING port
addi $k0, $zero,1 # to logic 1,
```

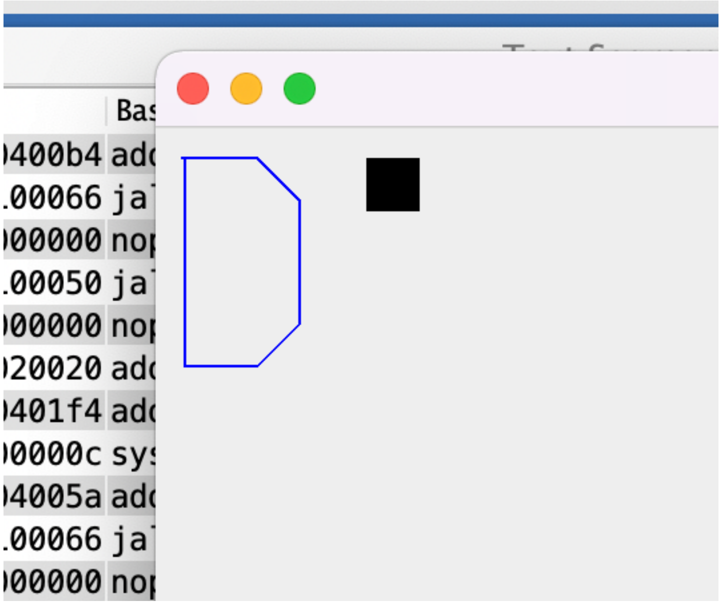


```
addi $k0, $zero,1 # to logic 1,
sb $k0, 0($at) # to start running
nop
jr $ra
nop

#-----------------------------------------------------------
# STOP procedure, to stop running
# param[in] none
#-----------------------------------------------------------
STOP: li $at, MOVING # change MOVING port to 0
sb $zero, 0($at) # to stop
nop
jr $ra
nop



#-----------------------------------------------------------
# TRACK procedure, to start drawing line
# param[in] none
#-----------------------------------------------------------
TRACK: li $at, LEAVETRACK # change LEAVETRACK port
addi $k0, $zero,1 # to logic 1,
sb $k0, 0($at) # to start tracking
nop
jr $ra
nop



#-----------------------------------------------------------
# UNTRACK procedure, to stop drawing line
# param[in] none
```



```
nop
jr $ra
nop



#-------------------------------------------------------------
# UNTRACK procedure, to stop drawing line
# param[in] none
#-------------------------------------------------------------
UNTRACK:li $at, LEAVETRACK # change LEAVETRACK port to 0
sb $zero, 0($at) # to stop drawing tail
nop
jr $ra
nop

#-------------------------------------------------------------
# ROTATE procedure, to rotate the robot
# param[in] $a0, An angle between 0 and 359
# 0 : North (up)
# 90: East (right)
# 180: South (down)
# 270: West (left)
#-------------------------------------------------------------



ROTATE: li $at, HEADING # change HEADING port
sw $a0, 0($at) # to rotate robot
nop
jr $ra
nop
end:
```

- Kết quả chạy mô phỏng:



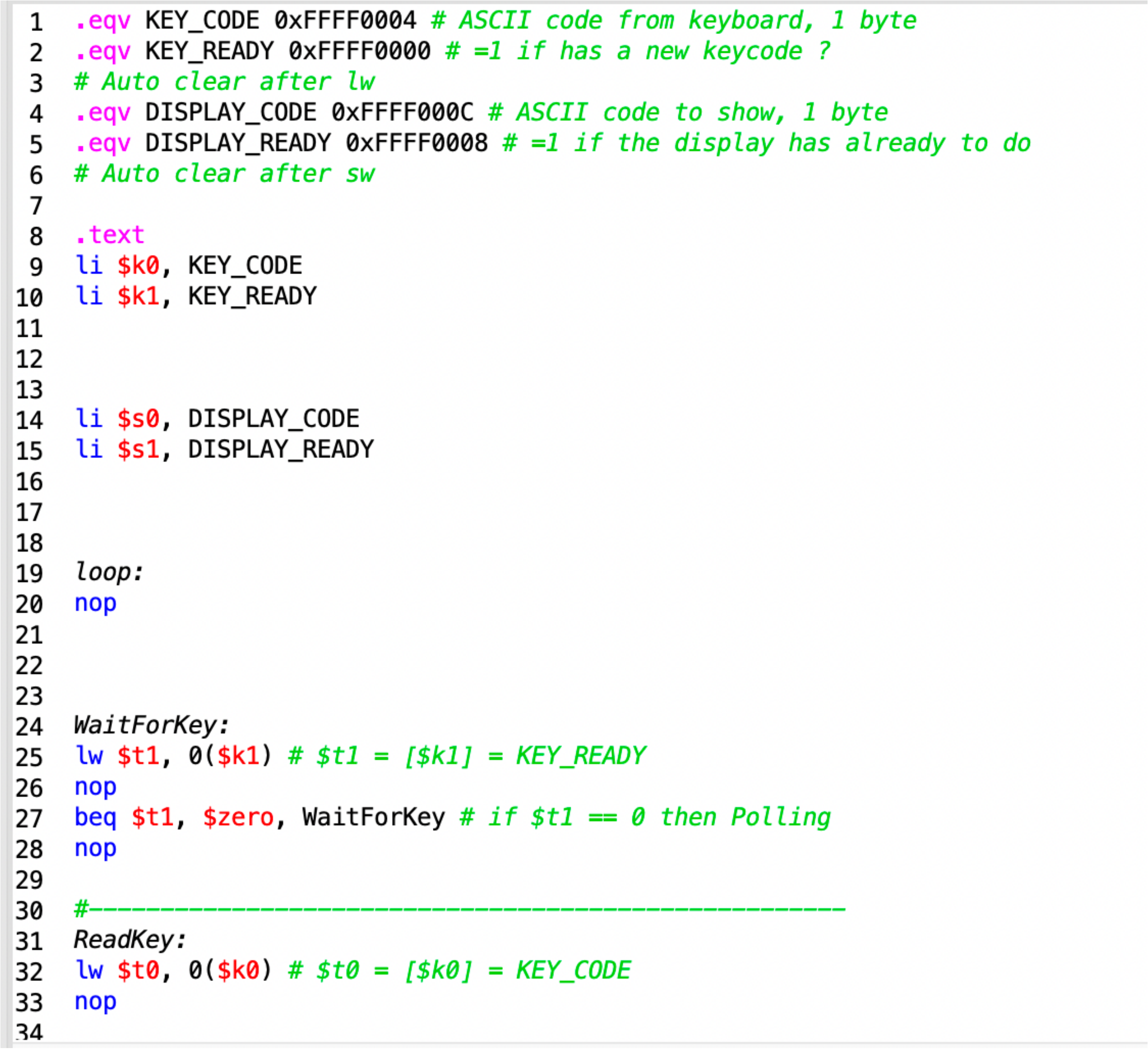
```
Bas
400b4 add
00066 jal
00000 nop
00050 jal
00000 nop
20020 add
401f4 add
0000c sys
4005a add
00066 jal
00000 nop
```

- Giải thích:

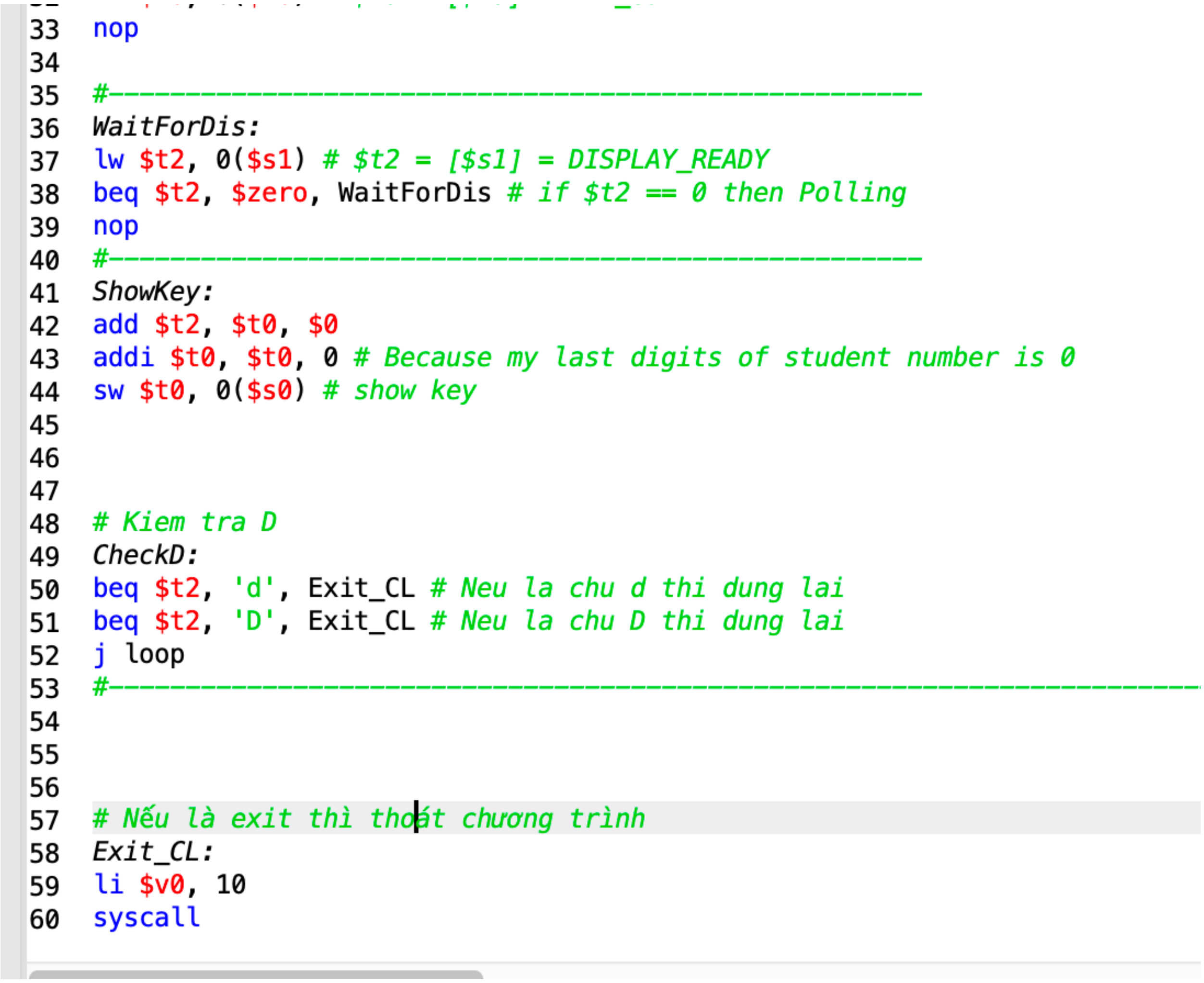
- B1: Đầu tiên, em vẽ sang phải (90 độ)
- B2: Vẽ sang góc 135 độ
- B3: Vẽ xuống dưới (180 độ)
- B4: Vẽ sang góc 225 độ
- B5: Vẽ ngang sang trái (270 độ)
- B6: Vẽ thẳng lên trên góc 360 độ

4. Assignment 4

- Mã nguồn:

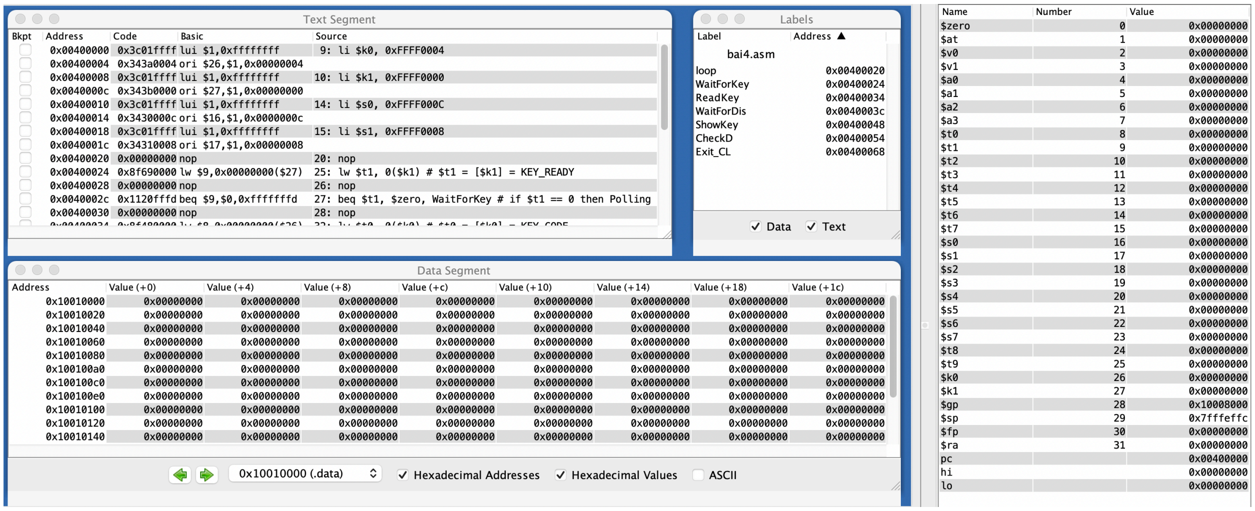


```
.eqv KEY_CODE 0xFFFF0004 # ASCII code from keyboard, 1 byte
.eqv KEY_READY 0xFFFF0000 # =1 if has a new keycode ?
# Auto clear after lw
.eqv DISPLAY_CODE 0xFFFF000C # ASCII code to show, 1 byte
.eqv DISPLAY_READY 0xFFFF0008 # =1 if the display has already to do
# Auto clear after sw

.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

```



```
nop

#-----------------------------------------------------
WaitForDis:
lw $t2, 0($s1) # $t2 = [$s1] = DISPLAY_READY
beq $t2, $zero, WaitForDis # if $t2 == 0 then Polling
nop
#-----------------------------------------------------
ShowKey:
add $t2, $t0, $0
addi $t0, $t0, 0 # Because my last digits of student number is 0
sw $t0, 0($s0) # show key



# Kiem tra D
CheckD:
beq $t2, 'd', Exit_CL # Neu la chu d thi dung lai
beq $t2, 'D', Exit_CL # Neu la chu D thi dung lai
j loop
#------------------------------------------------------------------------



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

- Kết quả chạy:



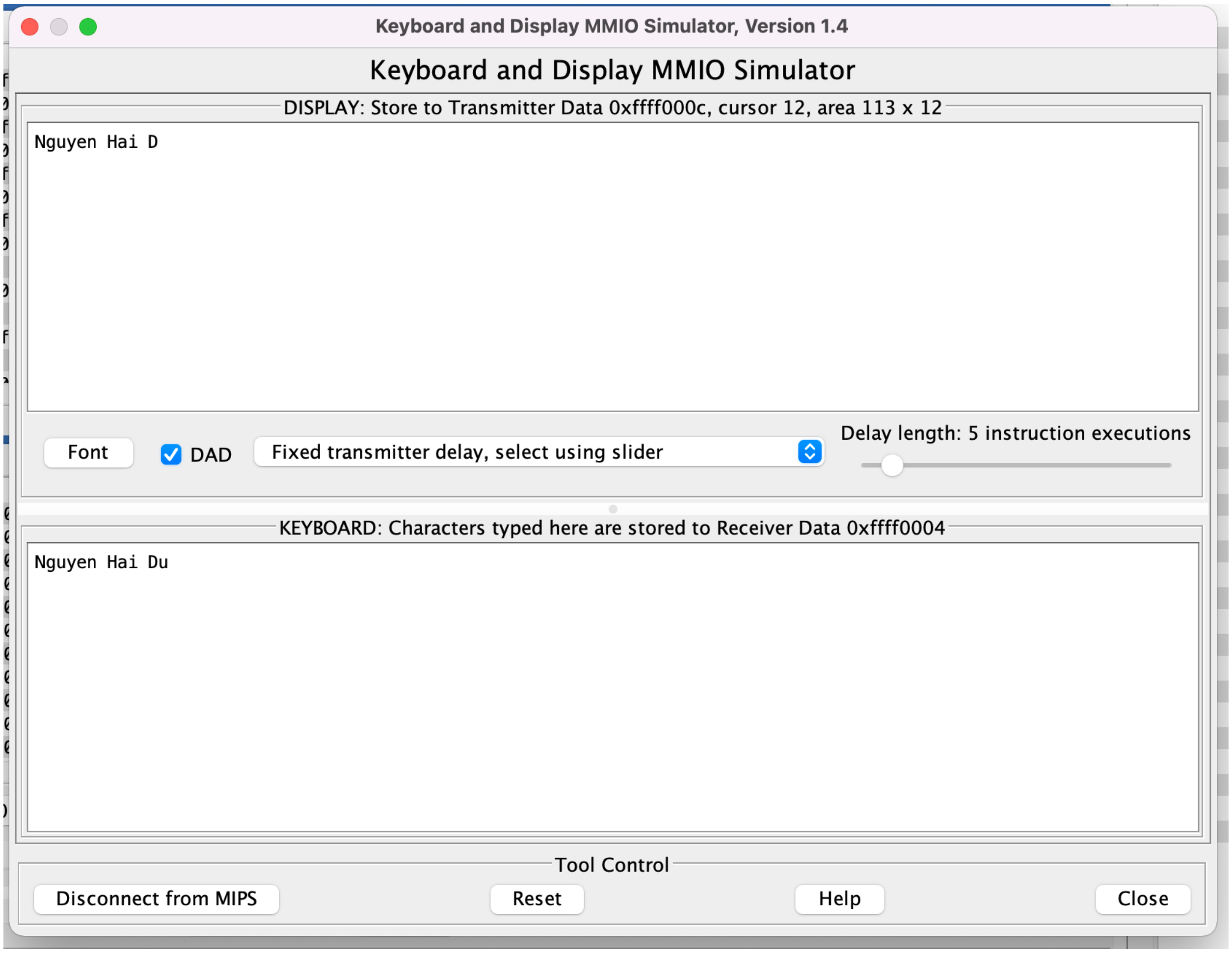
Bkpt	Address	Code	Basic	Source
□	0x00400000	0x3c01ffff	lui $1,0xffffffff	9: li $k0, 0xFFFF0004
□	0x00400004	0x343a0004	ori $26,$1,0x00000004	
□	0x00400008	0x3c01ffff	lui $1,0xffffffff	10: li $k1, 0xFFFF0000
□	0x0040000c	0x343b0000	ori $27,$1,0x00000000	
□	0x00400010	0x3c01ffff	lui $1,0xffffffff	14: li $s0, 0xFFFF000C
□	0x00400014	0x3430000c	ori $16,$1,0x0000000c	
□	0x00400018	0x3c01ffff	lui $1,0xffffffff	15: li $s1, 0xFFFF0008
□	0x0040001c	0x34310008	ori $17,$1,0x00000008	
□	0x00400020	0x00000000	nop	20: nop
□	0x00400024	0x8f690000	lw $9,0x00000000($27)	25: lw $t1, 0($k1) # $t1 = [$k1] = KEY_READY
□	0x00400028	0x00000000	nop	26: nop
□	0x0040002c	0x1120fffd	beq $9,$0,0xfffffffd	27: beq $t1, $zero, WaitForKey # if $t1 == 0 then Polling
□	0x00400030	0x00000000	nop	28: nop
□	0x00400034	0x8f480000	lw $8,0x00000000($26)	32: lw $t0, 0($k0) # $t0 = [$k0] = KEY_CODE

Label	Address ▲
bai4.asm	
loop	0x00400020
WaitForKey	0x00400024
ReadKey	0x00400034
WaitForDis	0x0040003c
ShowKey	0x00400048
CheckD	0x00400054
Exit_CL	0x00400068

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hexadecimal Values □ ASCII

Name	Number	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000


Keyboard and Display MMIO Simulator, Version 1.4

Keyboard and Display MMIO Simulator

DISPLAY: Store to Transmitter Data 0xffff000c, cursor 12, area 113 x 12

```
Nguyen Hai D
```

Font ☑ DAD Fixed transmitter delay, select using slider

Delay length: 5 instruction executions

KEYBOARD: Characters typed here are stored to Receiver Data 0xffff0004

```
Nguyen Hai Du
```

- Giải thích:

- Thực hiện kiểm tra từng kí tự nhập vào
- Các bước:
    - In ra ký tự được nhập vào
    - Kiểm tra xem ký tự có phải ‘D’ hoặc ‘d’ không
    - Nếu là ‘D’ hoặc ‘d’ thì dừng lại, không thì tiếp tục