Thực hành kiến trúc máy tính
Week 10.2

Bùi Quang Hưng – 20225849

Assignment 1

a.Vẽ tam giác đều

- Code:

```
.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:
	#jal TRACK # draw track line
	addi $a0, $zero, 90 # Marsbot rotates 90* and start running
	jal ROTATE
	jal GO
sleep1:
	addi $v0,$zero,32 # Keep running by sleeping in 1000 ms
	li $a0,16000
	syscall
	jal UNTRACK # keep old track
	#jal TRACK # and draw new track line
goDOWN:
        addi $a0, $zero, 180 # Marsbot rotates 180*
        jal ROTATE
sleep2:
        addi $v0,$zero,32 # Keep running by sleeping in 2000 ms
        li $a0,5000
        syscall
        jal UNTRACK # keep old track
        jal TRACK # and draw new track line
go135:
        addi $a0, $zero, 150 # Marsbot rotates 270*
        jal ROTATE
sleep3:
        addi $v0,$zero,32 # Keep running by sleeping in 1000 ms
        li $a0,10000
        syscall
        jal UNTRACK # keep old track
        jal TRACK # and draw new track line
go270:
        addi $a0, $zero, 270 # Marsbot rotates 120*
        jal ROTATE
sleep4:
        addi $v0,$zero,32 # Keep running by sleeping in 2000 ms
        li $a0,10000
        syscall
        jal UNTRACK # keep old track
        jal TRACK # and draw new track line
go30:
        addi $a0, $zero, 30 # Marsbot rotates 120*
        jal ROTATE
```

```
sleep5:
        addi $v0,$zero,32 # Keep running by sleeping in 2000 ms
        li $a0,10000
        syscall
        jal UNTRACK # keep old track
goUP:
        addi $a0, $zero, 0 #Marsbot rotates 0*
        jal ROTATE
sleep6:
        addi $v0,$zero,32 # Keep running by sleeping in 2000 ms
        li $a0,4000
        syscall
        jal UNTRACK # keep old track
        #jal TRACK # and draw new track line

end_main:
        jal STOP
        li $v0,10
        syscall
#-----------------------------------------------------------
# GO procedure, to start running
# param[in] none
#-----------------------------------------------------------
GO:
        li $at, MOVING # change MOVING port
        addi $k0, $zero,1 # to logic 1,
        sb $k0, 0($at) # to start running
        jr $ra
#-----------------------------------------------------------
```

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

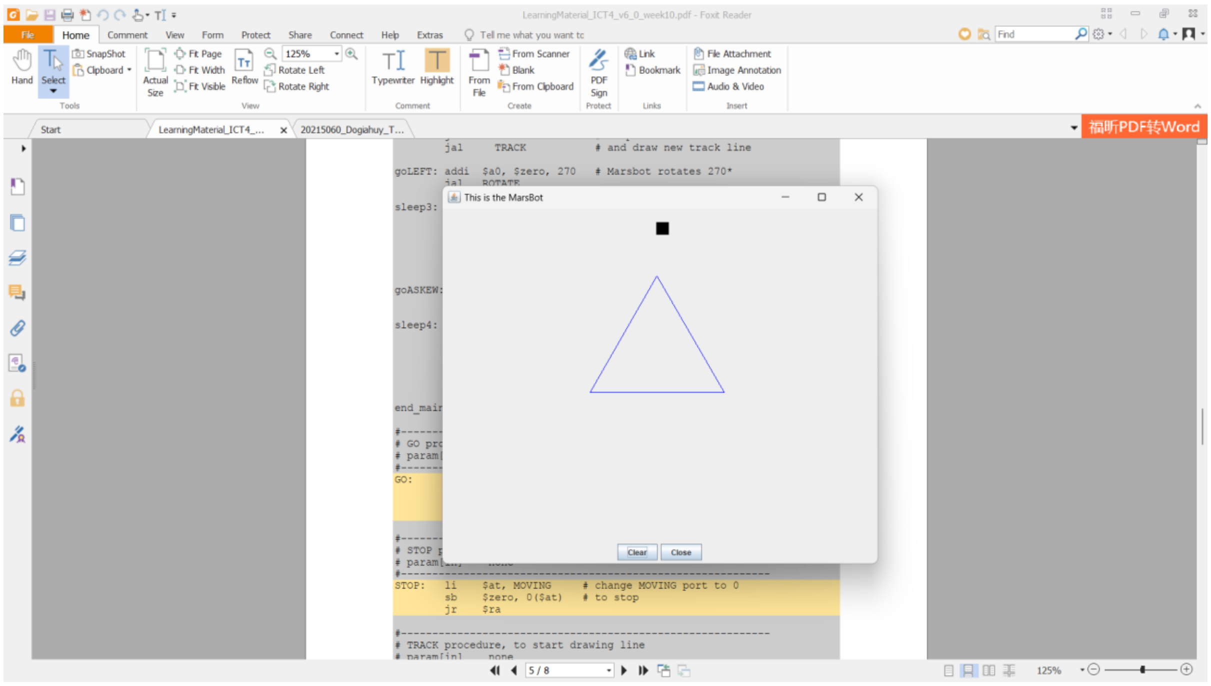
#---

ROTATE: li $at, HEADING # change HEADING port

sw $a0, 0($at) # to rotate robot

jr $ra

- Kết quả:



```
         jal     TRACK              # and draw new track line

goLEFT:  addi    $a0, $zero, 270    # Marsbot rotates 270*
         jal     ROTATE

sleep3:

goASKEW:

sleep4:

end_mair

#-------
# GO prc
# param[
#-------
GO:

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

#---------------------------------------------------------
# TRACK procedure, to start drawing line
# param[in]    none
```

➔ Kết quả đúng với lí thuyết

b.Vẽ hình vuông

- Code

.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):

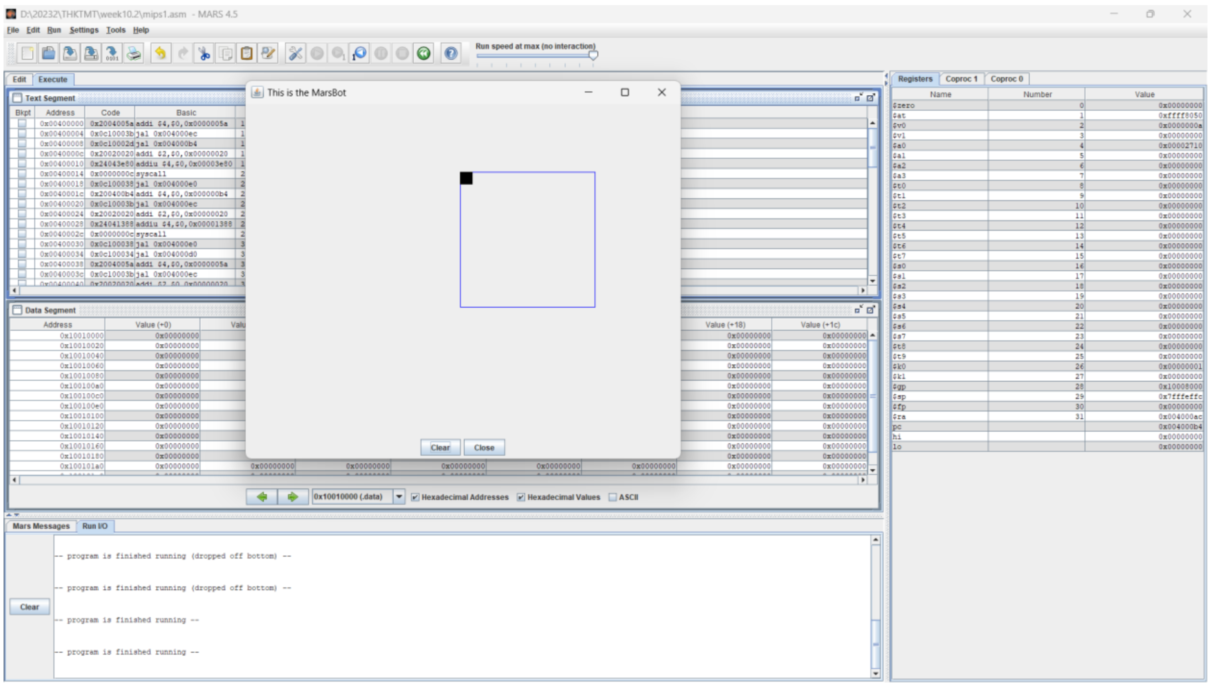
# whether or not to leave a track

```
.eqv WHEREX 0xffff8030 # Integer: Current x-location of MarsBot
.eqv WHEREY 0xffff8040 # Integer: Current y-location of MarsBot
.text
main:
        #jal TRACK # draw track line
        addi $a0, $zero, 90 # Marsbot rotates 90* and start running
        jal ROTATE
        jal GO
sleep1:
        addi $v0,$zero,32 # Keep running by sleeping in 1000 ms
        li $a0,16000
        syscall
        jal UNTRACK # keep old track
        #jal TRACK # and draw new track line
goDOWN:
        addi $a0, $zero, 180 # Marsbot rotates 180*
        jal ROTATE
sleep2:
        addi $v0,$zero,32 # Keep running by sleeping in 2000 ms
        li $a0,5000
        syscall
        jal UNTRACK # keep old track
        jal TRACK # and draw new track line
go90:
        addi $a0, $zero, 90 # Marsbot rotates 270*
        jal ROTATE
sleep3:
        addi $v0,$zero,32 # Keep running by sleeping in 1000 ms
        li $a0,10000
        syscall
        jal UNTRACK # keep old track
        jal TRACK # and draw new track line
go180:
        addi $a0, $zero, 180 # Marsbot rotates 120*
        jal ROTATE
sleep4:
        addi $v0,$zero,32 # Keep running by sleeping in 2000 ms
        li $a0,10000
        syscall
        jal UNTRACK # keep old track
        jal TRACK # and draw new track line
go270:
        addi $a0, $zero, 270 # Marsbot rotates 120*
        jal ROTATE
sleep5:
        addi $v0,$zero,32 # Keep running by sleeping in 2000 ms
        li $a0,10000
        syscall
        jal UNTRACK # keep old track
        jal TRACK # and draw new track line
goUP:
        addi $a0, $zero, 0 #Marsbot rotates 0*
        jal ROTATE
sleep6:
        addi $v0,$zero,32 # Keep running by sleeping in 2000 ms
        li $a0,10000
        syscall
        jal UNTRACK # keep old track
        jal TRACK # and draw new track line



end_main:
        jal STOP
        li $v0,10
        syscall
#----------------------------------------------------------
# GO procedure, to start running
# param[in] none
#----------------------------------------------------------
GO:
        li $at, MOVING # change MOVING port
        addi $k0, $zero,1 # to logic 1,
        sb $k0, 0($at) # to start running
        jr $ra
#----------------------------------------------------------
# STOP procedure, to stop running
# param[in] none
#----------------------------------------------------------
STOP:
        li $at, MOVING # change MOVING port to 0
        sb $zero, 0($at) # to stop
        jr $ra
#----------------------------------------------------------
# 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
	jr $ra
#-----------------------------------------------------------
# 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
	jr $ra
#-----------------------------------------------------------
# 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
jr $ra
```

- Kết quả



Text Segment

Bkpt	Address	Code	Basic
	0x00400000	0x2004005a	addi $4,$0,0x0000005a
	0x00400004	0x0c10003b	jal 0x004000ec
	0x00400008	0x0c10002d	jal 0x004000b4
	0x0040000c	0x20020020	addi $2,$0,0x00000020
	0x00400010	0x24043e80	addiu $4,$0,0x00003e80
	0x00400014	0x0000000c	syscall
	0x00400018	0x0c100038	jal 0x004000e0
	0x0040001c	0x200400b4	addi $4,$0,0x000000b4
	0x00400020	0x0c10003b	jal 0x004000ec
	0x00400024	0x20020020	addi $2,$0,0x00000020
	0x00400028	0x24041388	addiu $4,$0,0x00001388
	0x0040002c	0x0000000c	syscall
	0x00400030	0x0c100038	jal 0x004000e0
	0x00400034	0x0c100034	jal 0x004000d0
	0x00400038	0x2004005a	addi $4,$0,0x0000005a
	0x0040003c	0x0c10003b	jal 0x004000ec

Registers

Name	Number	Value
$zero	0	0x00000000
$at	1	0xffff8050
$v0	2	0x0000000a
$v1	3	0x00000000
$a0	4	0x00002710
$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	0x00000001
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x004000ac
pc		0x004000b4
hi		0x00000000
lo		0x00000000

```
-- program is finished running (dropped off bottom) --

-- program is finished running (dropped off bottom) --

-- program is finished running --

-- program is finished running --
```

➔ Kết quả đúng với lí thuyết

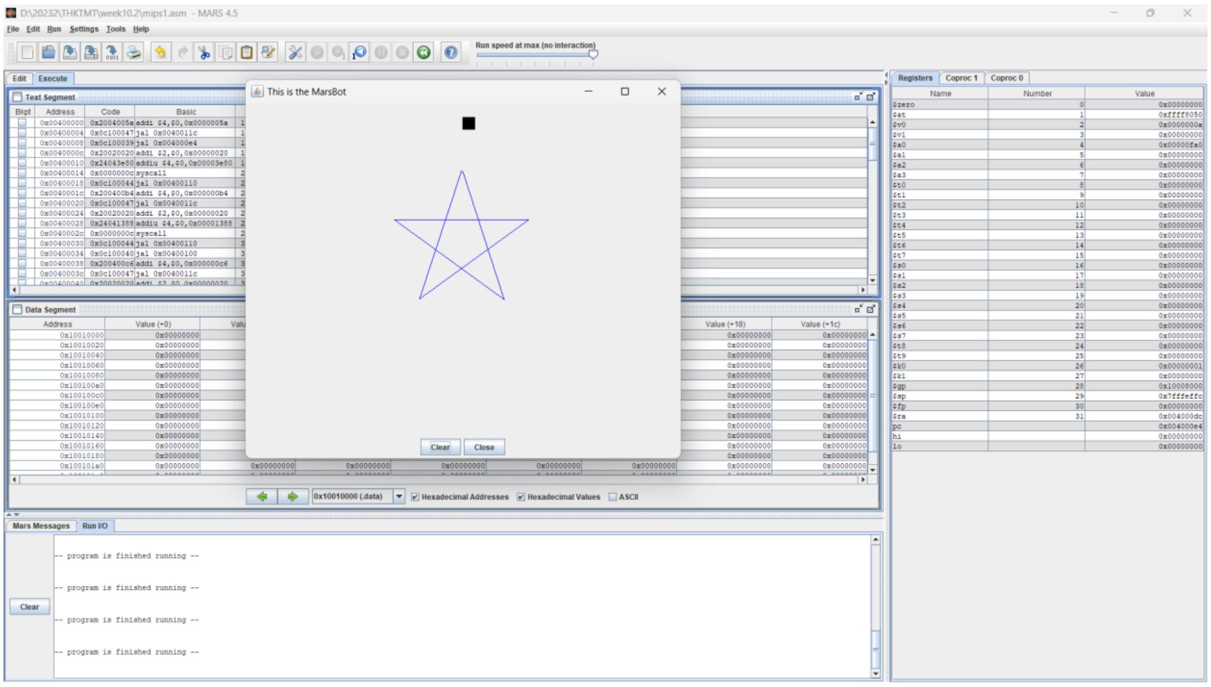
c.Vẽ ngôi sao năm cánh

- Code

```
.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:
        #jal TRACK # draw track line
        addi $a0, $zero, 90 # Marsbot rotates 90* and start running
        jal ROTATE
        jal GO
sleep1:
        addi $v0,$zero,32 # Keep running by sleeping in 1000 ms
        li $a0,16000
        syscall
        jal UNTRACK # keep old track
        #jal TRACK # and draw new track line
goDOWN:
        addi $a0, $zero, 180 # Marsbot rotates 180*
        jal ROTATE
sleep2:
        addi $v0,$zero,32 # Keep running by sleeping in 2000 ms
        li $a0,5000
        syscall
        jal UNTRACK # keep old track
        jal TRACK # and draw new track line
canh1:
        addi $a0, $zero, 198 # Marsbot rotates 198*
        jal ROTATE
sleep3:
        addi $v0,$zero,32 # Keep running by sleeping in 1000 ms
        li $a0,10000
        syscall
        jal UNTRACK # keep old track
        jal TRACK # and draw new track line
canh2:
        addi $a0, $zero, 54 # Marsbot rotates 54*
        jal ROTATE
sleep4:
        addi $v0,$zero,32 # Keep running by sleeping in 2000 ms
        li $a0,10000
        syscall
        jal UNTRACK # keep old track
        jal TRACK # and draw new track line
canh3:
        addi $a0, $zero, 270 # Marsbot rotates 270*
        jal ROTATE
sleep5:
        addi $v0,$zero,32 # Keep running by sleeping in 2000 ms
        li $a0,10000
        syscall
        jal UNTRACK # keep old track
        jal TRACK # and draw new track line
canh4:
        addi $a0, $zero, 126 #Marsbot rotates 126*
        jal ROTATE
sleep6:
        addi $v0,$zero,32 # Keep running by sleeping in 2000 ms
        li $a0,10000
        syscall
        jal UNTRACK # keep old track
        jal TRACK # and draw new track line
canh5:
        addi $a0, $zero, 342 #Marbot rotate 342*
        jal ROTATE
sleep7:
        addi $v0,$zero,32 # Keep running by sleeping in 2000 ms
        li $a0,10000
        syscall
        jal UNTRACK # keep old track
goUp:
        addi $a0, $zero, 0 #Marbot rotate 0*
        jal ROTATE
sleep8:
        addi $v0,$zero,32 # Keep running by sleeping in 2000 ms
        li $a0,4000
        syscall
        jal UNTRACK # keep old track
end_main:
        jal STOP
        li $v0,10
        syscall
#-----------------------------------------------------------
# GO procedure, to start running
# param[in] none
#-----------------------------------------------------------
GO:
        li $at, MOVING # change MOVING port
        addi $k0, $zero,1 # to logic 1,
        sb $k0, 0($at) # to start running
        jr $ra
#-----------------------------------------------------------
# STOP procedure, to stop running
# param[in] none
#-----------------------------------------------------------
STOP:
        li $at, MOVING # change MOVING port to 0
        sb $zero, 0($at) # to stop
        jr $ra
#-----------------------------------------------------------
# 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
        jr $ra
#-----------------------------------------------------------
# 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
        jr $ra
#-----------------------------------------------------------
# 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
```

jr $ra

- Kết quả



```
Bkpt  Address     Code        Basic
      0x00400000  0x2004005a  addi $4,$0,0x0000005a
      0x00400004  0x0c100047  jal 0x0040011c
      0x00400008  0x0c100039  jal 0x004000e4
      0x0040000c  0x20020020  addi $2,$0,0x00000020
      0x00400010  0x24043e80  addiu $4,$0,0x00003e80
      0x00400014  0x0000000c  syscall
      0x00400018  0x0c100044  jal 0x00400110
      0x0040001c  0x200400b4  addi $4,$0,0x000000b4
      0x00400020  0x0c100047  jal 0x0040011c
      0x00400024  0x20020020  addi $2,$0,0x00000020
      0x00400028  0x24041388  addiu $4,$0,0x00001388
      0x0040002c  0x0000000c  syscall
      0x00400030  0x0c100044  jal 0x00400110
      0x00400034  0x0c100040  jal 0x00400100
      0x00400038  0x200400c6  addi $4,$0,0x000000c6
      0x0040003c  0x0c100047  jal 0x0040011c
```

```
Address     Value (+0)
0x10010000  0x00000000
0x10010020  0x00000000
0x10010040  0x00000000
0x10010060  0x00000000
0x10010080  0x00000000
0x100100a0  0x00000000
0x100100c0  0x00000000
0x100100e0  0x00000000
0x10010100  0x00000000
0x10010120  0x00000000
0x10010140  0x00000000
0x10010160  0x00000000
0x10010180  0x00000000
0x100101a0  0x00000000
```

```
-- program is finished running --

-- program is finished running --

-- program is finished running --

-- program is finished running --
```

Name	Number	Value
$zero	0	0x00000000
$at	1	0xffff8050
$v0	2	0x0000000a
$v1	3	0x00000000
$a0	4	0x00000fa0
$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	0x00000001
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x004000dc
pc		0x004000e4
hi		0x00000000
lo		0x00000000

➔ Kết quả đúng với lí thuyết

Assignment 2

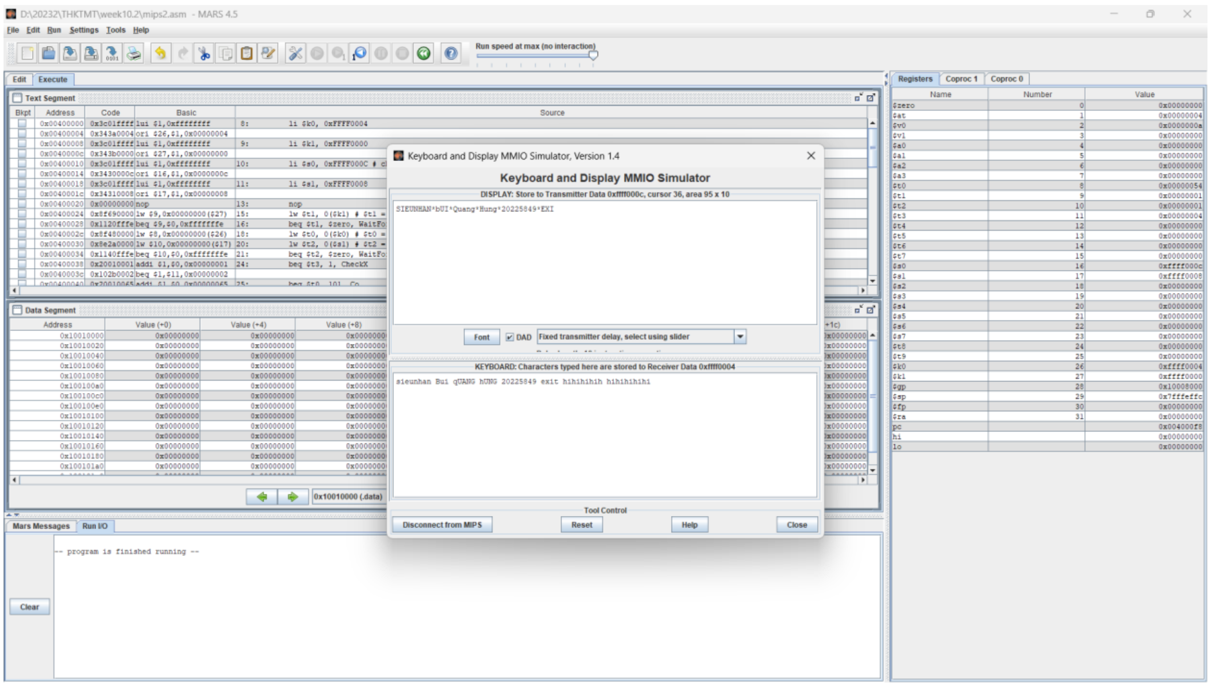
- Code:

```
.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 # chua ky tu can in ra man hinh
        li $s1, DISPLAY_READY
loop:
        nop
```

```
WaitForKey:
        lw $t1, 0($k1) # $t1 = [$k1] = KEY_READY
        beq $t1, $zero, WaitForKey # if $t1 == 0 then Polling
ReadKey:
        lw $t0, 0($k0) # $t0 = [$k0] = KEY_CODE
WaitForDis:
        lw $t2, 0($s1) # $t2 = [$s1] = DISPLAY_READY
        beq $t2, $zero, WaitForDis # if $t2 == 0 then Polling
Kiemtra:
CheckE:
        beq $t3, 1, CheckX
        beq $t0, 101, Dem
CheckX:
        beq $t3, 2, CheckI
        beq $t0, 120, Dem
CheckI:
        beq $t3, 3, CheckT
        beq $t0, 105, Dem
CheckT:
        beq $t3, 4, Encrypt2
        beq $t0, 116, Dem
Encrypt1:
        addi $t3, $zero, 0
Encrypt2:
Upper:
        bgt $t0, 90, Lower
        blt $t0, 65, Lower
        addi $t0, $t0, 32
        j ShowKey
```

```
Lower:
        bgt $t0, 122, ChuSo
        blt $t0, 97, ChuSo
        addi $t0, $t0, -32
        j ShowKey
ChuSo:
        bgt $t0, 57, Otherwise
        blt $t0, 48, Otherwise
        addi $t0, $t0, 0
        j ShowKey
Otherwise:
        addi $t0, $zero, 42
ShowKey:
        sw $t0, 0($s0) # show key
        nop
        beq $t3, 4, Exit
        j loop
Dem:
        addi $t3, $t3, 1
        j Encrypt2
Exit:
        li $v0, 10
        syscall
```

- Kết quả:


D:\20232\THKTMT\week10.2\mips2.asm - MARS 4.5

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x3c01ffff	lui $1,0xffffffff	8: li $k0, 0xFFFF0004
	0x00400004	0x343a0004	ori $26,$1,0x00000004	
	0x00400008	0x3c01ffff	lui $1,0xffffffff	9: li $k1, 0xFFFF0000
	0x0040000c	0x343b0000	ori $27,$1,0x00000000	
	0x00400010	0x3c01ffff	lui $1,0xffffffff	10: li $s0, 0xFFFF000C # c
	0x00400014	0x3430000c	ori $16,$1,0x0000000c	
	0x00400018	0x3c01ffff	lui $1,0xffffffff	11: li $s1, 0xFFFF0008
	0x0040001c	0x34310008	ori $17,$1,0x00000008	
	0x00400020	0x00000000	nop	13: nop
	0x00400024	0x8f690000	lw $9,0x00000000($27)	15: lw $t1, 0($k1) # $t1 =
	0x00400028	0x1120fffe	beq $9,$0,0xfffffffe	16: beq $t1, $zero, WaitFo
	0x0040002c	0x8f480000	lw $8,0x00000000($26)	18: lw $t0, 0($k0) # $t0 =
	0x00400030	0x8e2a0000	lw $10,0x00000000($17)	20: lw $t2, 0($s1) # $t2 =
	0x00400034	0x1140fffe	beq $10,$0,0xfffffffe	21: beq $t2, $zero, WaitFo
	0x00400038	0x20010001	addi $1,$0,0x00000001	24: beq $t3, 1, CheckX
	0x0040003c	0x102b0002	beq $1,$11,0x00000002	
	0x00400040	0x20010065	addi $1,$0,0x00000065	25: beq $t0, 101, Co

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)
0x10010000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000

0x10010000 (.data)

Keyboard and Display MMIO Simulator, Version 1.4

Keyboard and Display MMIO Simulator

DISPLAY: Store to Transmitter Data 0xffff000c, cursor 36, area 95 x 10

```
SIEUNHAN*bUI*Quang*Hung*20225849*EXI
```

DAD: Fixed transmitter delay, select using slider

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

```
sieunhan Bui qUANG hUNG 20225849 exit hihihihih hihihihihi
```

Registers

Name	Number	Value
$zero	0	0x00000000
$at	1	0x00000004
$v0	2	0x0000000a
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000054
$t1	9	0x00000001
$t2	10	0x00000001
$t3	11	0x00000004
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0xffff000c
$s1	17	0xffff0008
$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	0xffff0004
$k1	27	0xffff0000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x004000f8
hi		0x00000000
lo		0x00000000

Mars Messages / Run I/O

```
-- program is finished running --
```

➔ **Kết quả đúng với lí thuyết**

Assignment 3

- Code

.eqv HEADING 0xffff8010

.eqv MOVING 0xffff8050

.eqv LEAVETRACK 0xffff8020

.eqv WHEREX 0xffff8030

.eqv WHEREY 0xffff8040

.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

main:

li $t8, KEY_CODE

li $t9, KEY_READY

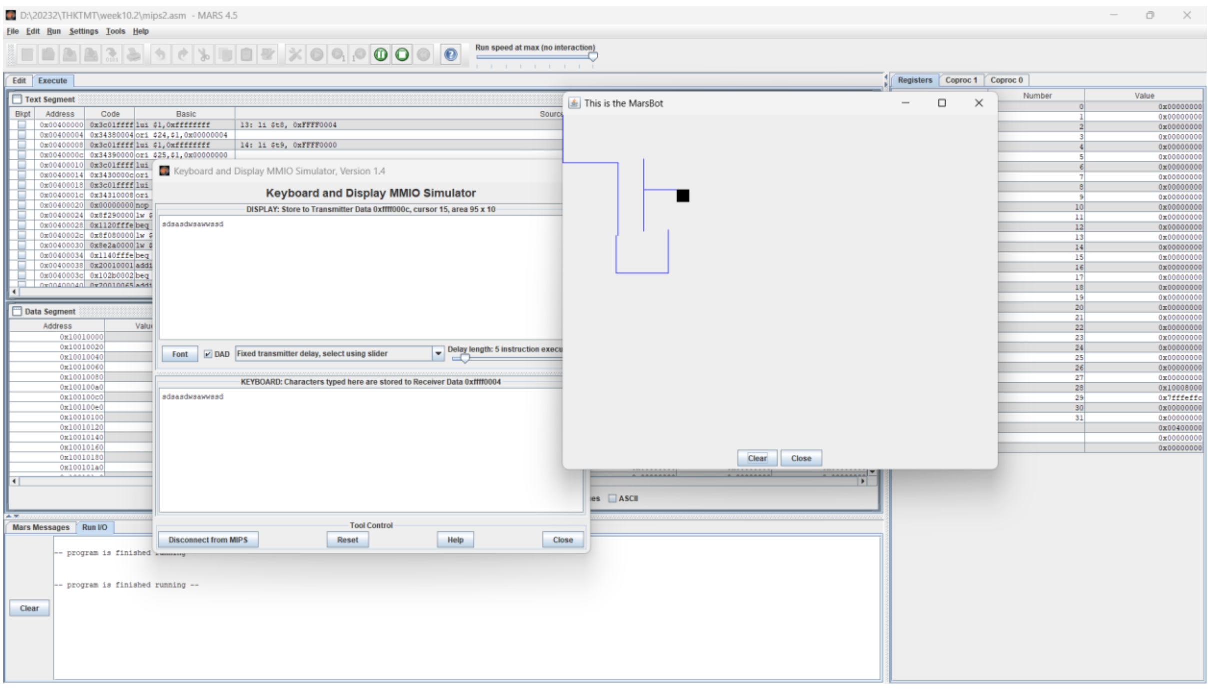
li $s0, DISPLAY_CODE # chua ky tu can in ra man hinh

```
li $s1, DISPLAY_READY
loop: nop
WaitForKey:
lw $t1, 0($t9) # $t1 = [$k1] = KEY_READY
beq $t1, $zero, WaitForKey # if $t1 == 0 then Polling
ReadKey:
lw $t0, 0($t8) # $t0 = [$k0] = KEY_CODE
WaitForDis:
lw $t2, 0($s1) # $t2 = [$s1] = DISPLAY_READY
beq $t2, $zero, WaitForDis # if $t2 == 0 then Polling
Kiemtra:
KiemTraE:
beq $t3, 1, KiemTraX
beq $t0, 101, Co
KiemTraX:
beq $t3, 2, KiemTraI
beq $t0, 120, Co
KiemTraI:
beq $t3, 3, KiemTraT
beq $t0, 105, Co
KiemTraT:
beq $t3, 4, Encrypt2
beq $t0, 116, Co
Encrypt:
addi $t3, $zero, 0
Encrypt2:
beq $t0, 65, sleepA
beq $t0, 97, sleepA
beq $t0, 87, sleepW
beq $t0, 119, sleepW
beq $t0, 68, sleepD
beq $t0, 100, sleepD
beq $t0, 83, sleepS
beq $t0, 115, sleepS
beq $t0, 32, Nghiem
beq $t0, 67, Ditiep
beq $t0, 99, Ditiep
ShowKey:
sw $t0, 0($s0) # show key
nop
j loop
Co:addi $t3, $t3, 1
j Encrypt2
sleepW:
addi $a0, $zero, 0
jal ROTATE
jal GO
jal UNTRACK # keep old track
jal TRACK # and draw new track line
j ShowKey
sleepS:
addi $a0, $zero, 180
jal ROTATE
jal GO
jal UNTRACK # keep old track
jal TRACK # and draw new track line
j ShowKey
sleepD:
```

```
addi $a0, $zero, 90
jal ROTATE
jal GO
jal UNTRACK # keep old track
jal TRACK # and draw new track line
j ShowKey
sleepA:
addi $a0, $zero, 270
jal ROTATE
jal GO
jal UNTRACK # keep old trackjal TRACK # and draw new track line
j ShowKey
Nghiem:
jal STOP
j ShowKey
Ditiep:
jal GO
j ShowKey
end_main:
GO:
li $at, MOVING # change MOVING port
addi $k0, $zero,1 # to logic 1,
sb $k0, 0($at) # to start running
jr $ra
ROTATE:
li $at, HEADING # change HEADING port
sw $a0, 0($at) # to rotate robot
jr $ra
STOP:
```

```
li $at, MOVING # change MOVING port to 0
sb $zero, 0($at) # to stop
jr $ra
TRACK:
li $at, LEAVETRACK # change LEAVETRACK port
addi $k0, $zero,1 # to logic 1,
sb $k0, 0($at) # to start tracking
jr $ra
UNTRACK:
li $at, LEAVETRACK # change LEAVETRACK port to 0
sb $zero, 0($at) # to stop drawing tail
jr $ra
```

- Kết quả



D:\20232\THKTMT\week10.2\mips2.asm - MARS 4.5

Run speed at max (no interaction)

Bkpt	Address	Code	Basic	Source
	0x00400000	0x3c01ffff	lui $1,0xffffffff	13: li $t8, 0xFFFF0004
	0x00400004	0x34380004	ori $24,$1,0x00000004	
	0x00400008	0x3c01ffff	lui $1,0xffffffff	14: li $t9, 0xFFFF0000
	0x0040000c	0x34390000	ori $25,$1,0x00000000	
	0x00400010	0x3c01ffff	lui	
	0x00400014	0x3430000c	ori	
	0x00400018	0x3c01ffff	lui	
	0x0040001c	0x34310008	ori	
	0x00400020	0x00000000	nop	
	0x00400024	0x8f290000	lw $	
	0x00400028	0x1120fffe	beq	
	0x0040002c	0x8f080000	lw $	
	0x00400030	0x8e2a0000	lw $	
	0x00400034	0x1140fffe	beq	
	0x00400038	0x20010001	addi	
	0x0040003c	0x102b0002	beq	

Keyboard and Display MMIO Simulator, Version 1.4

Keyboard and Display MMIO Simulator

DISPLAY: Store to Transmitter Data 0xffff000c, cursor 15, area 95 x 10

```
sdsasdwsawwssd
```

DAD: Fixed transmitter delay, select using slider — Delay length: 5 instruction execu

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

```
sdsasdwsawwssd
```

This is the MarsBot

Mars Messages — Run I/O

```
-- program is finished running --

-- program is finished running --
```

➔ Kết quả đúng với lí thuyết