Bài thực hành số 10 phần 2

Nguyễn Văn Hiếu – 20225717

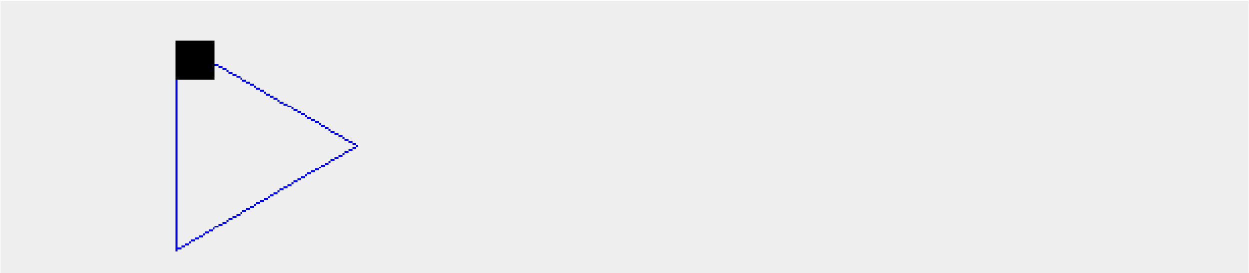
Assignment 1:

TH1: Vẽ tam giác đều

```
.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
.text
main: #jal TRACK # draw track line
 nop
 addi $a0, $zero, 120 # Marsbot rotates 90* and start
 jal ROTATE
 nop
 jal GO
 nop

sleep1: addi $v0,$zero,32 # Keep running by sleeping in 1000 ms
 li $a0,5000
 syscall

 jal UNTRACK # keep old track
 nop
 jal TRACK # and draw new track line
 nop
#goDOWN: addi $a0, $zero, 180 # Marsbot rotates 180*
 #jal ROTATE
 #nop

sleep2: addi $v0,$zero,32 # Keep running by sleeping in 2000 ms
 li $a0,5000
 syscall
 jal UNTRACK # keep old track
 nop
 jal TRACK # and draw new track line
 nop
goLEFT: addi $a0, $zero, 240 # Marsbot rotates 240
 jal ROTATE
 nop

sleep3: addi $v0,$zero,32 # Keep running by sleeping in 1000 ms
 li $a0,5000
 syscall
 jal UNTRACK # keep old track
 nop
 jal TRACK # and draw new track line
 nop

goASKEW:addi $a0, $zero, 360 # Marsbot rotates 360
 jal ROTATE
 nop

sleep4: addi $v0,$zero,32 # Keep running by sleeping in 2000 ms
 li $a0,5000
 syscall

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

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

Kết quả thu được:



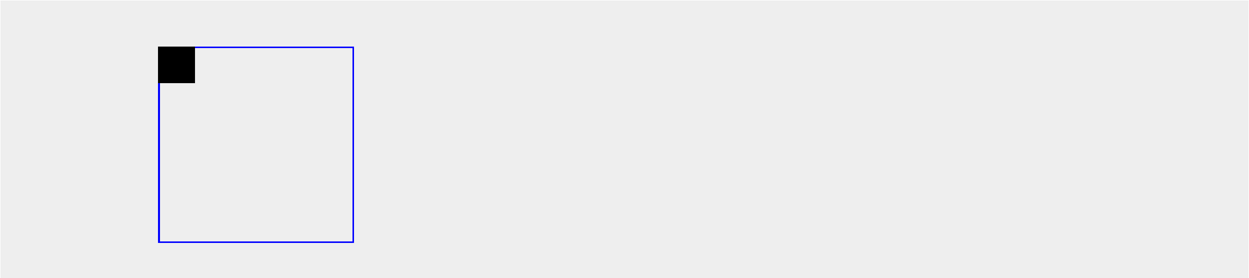
Trường hợp 2: Vẽ hình vuông

```
.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
.text
main: #jal TRACK # draw track li
 addi $a0, $zero, 120 # Marsbot rotates 90* and start
 jal ROTATE
 nop
 jal GO
 nop

sleep1: addi $v0,$zero,32 # Keep running by sleeping in 1000 ms
 li $a0,5000
 syscall

 jal UNTRACK # keep old track
 nop
 jal TRACK # and draw new track line
 nop
 addi $a0,$zero,90
 jal ROTATE
 nop
 jal GO
 nop
#goDOWN: addi $a0, $zero, 180 # Marsbot rotates 180*
 #jal ROTATE
 #nop

sleep2: addi $v0,$zero,32 # Keep running by sleeping in 2000 ms
 li $a0,5000
 syscall
 jal UNTRACK # keep old track
 nop
 jal TRACK # and draw new track line
 nop
goLEFT: addi $a0, $zero, 180 # Marsbot rotates 240
 jal ROTATE
 nop

sleep3: addi $v0,$zero,32 # Keep running by sleeping in 1000 ms
 li $a0,5000
 syscall
 jal UNTRACK # keep old track
 nop
 jal TRACK # and draw new track line
 nop

goASKEW:addi $a0, $zero, 270 # Marsbot rotates 360
 jal ROTATE
 nop

sleep4: addi $v0,$zero,32 # Keep running by sleeping in 2000 ms
 li $a0,5000
 syscall

 jal UNTRACK # keep old track
 nop
 jal TRACK # and draw new track line
 nop
 go:addi $a0,$zero,360
 jal ROTATE
 nop
 sleep5: addi $v0,$zero,32
 li $a0,5000
 syscall
 jal UNTRACK
 nop

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

Kết quả thu được:



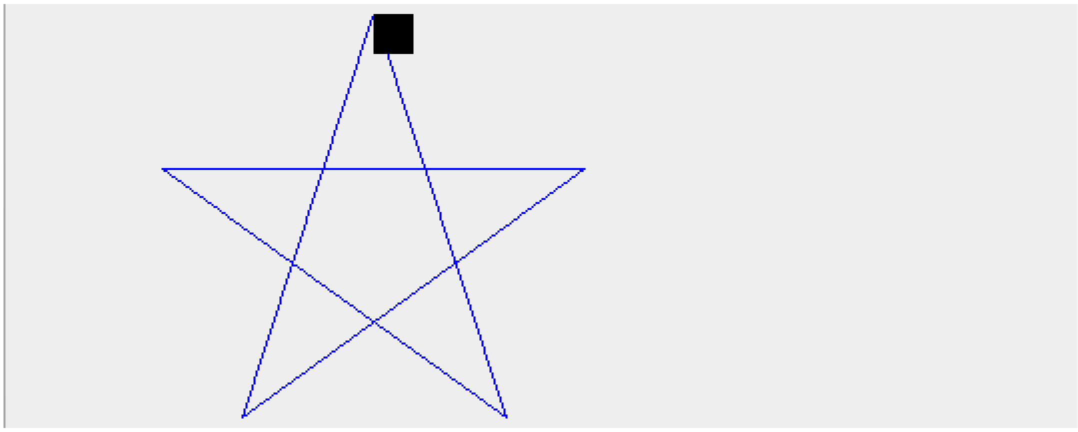
Trường hợp 3: Vẽ ngôi sao 5 cánh

```
.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
.text
main: #jal TRACK # draw track li
 addi $a0, $zero, 120 # Marsbot rotates 90* and start
 jal ROTATE
 nop
 jal GO
 nop

sleep1: addi $v0,$zero,32 # Keep running by sleeping in 1000 ms
 li $a0,10000
 syscall

 jal UNTRACK # keep old track
 nop
 jal TRACK # and draw new track line
 nop
 go:addi $a0,$zero,198
 jal ROTATE
 nop
 jal GO
 nop
#goDOWN: addi $a0, $zero, 180 # Marsbot rotates 180*
 #jal ROTATE
 #nop

sleep2: addi $v0,$zero,32 # Keep running by sleeping in 2000 ms
 li $a0,10000
 syscall
 jal UNTRACK # keep old track
 nop
 jal TRACK # and draw new track line
 nop
go1: addi $a0, $zero, 54 # Marsbot rotates 240
 jal ROTATE
 nop

sleep3: addi $v0,$zero,32 # Keep running by sleeping in 1000 ms
 li $a0,10000
 syscall
 jal UNTRACK # keep old track
 nop
 jal TRACK # and draw new track line
 nop

go2:addi $a0, $zero, 270 # Marsbot rotates 360
 jal ROTATE
 nop

sleep4: addi $v0,$zero,32 # Keep running by sleeping in 2000 ms
 li $a0,10000
 syscall

 jal UNTRACK # keep old track
 nop
 jal TRACK # and draw new track line
 nop
 go3:addi $a0,$zero,126
 jal ROTATE

 nop

 sleep5: addi $v0,$zero,32

 li $a0,10000

 syscall

 jal UNTRACK

 nop

 jal TRACK

 nop

 go4: addi $a0,$zero,342

 jal ROTATE

 nop

 sleep6: addi $v0,$zero,32

 li $a0,10000

 syscall

 jal UNTRACK

 nop


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

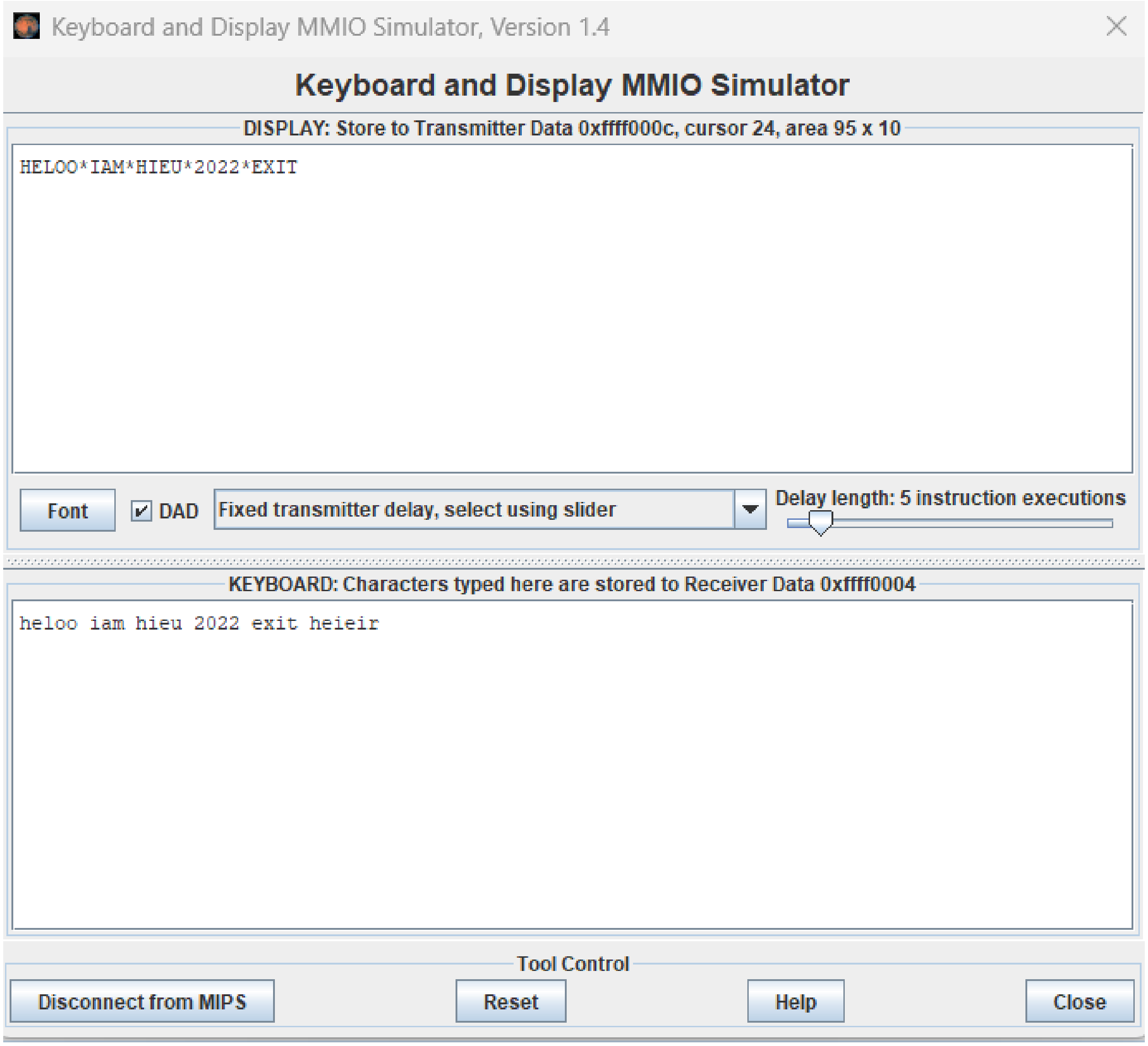
Kết quả thu được:



Assginment 2:

```
.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      # Địa chỉ của mã ASCII từ bàn phím

        li $k1, KEY_READY     # Địa chỉ của cờ báo hiệu có mã phím mới

        li $s0, DISPLAY_CODE   # Địa chỉ của mã ASCII để hiển thị

        li $s1, DISPLAY_READY  # Địa chỉ của cờ báo hiệu màn hình sẵn sàng

loop:

        nop

WaitForKey:

        lw $t1, 0($k1)       # Đọc giá trị từ KEY_READY vào $t1

        beq $t1, $zero, WaitForKey # Nếu $t1 == 0, chờ mã phím mới

ReadKey:
        lw $t0, 0($k0)      # Đọc mã phím từ KEY_CODE vào $t0
WaitForDis:
        lw $t2, 0($s1)      # Đọc giá trị từ DISPLAY_READY vào $t2
        beq $t2, $zero, WaitForDis # Nếu $t2 == 0, chờ màn hình sẵn sàng
Kiemtra:
CheckE:
        beq $t3, 1, CheckX   # Nếu $t3 == 1, kiểm tra ký tự 'X'
        beq $t0, 101, Co     # Nếu $t0 == 'e' (ASCII 101), đi đến Co
CheckX:
        beq $t3, 2, CheckI   # Nếu $t3 == 2, kiểm tra ký tự 'I'
        beq $t0, 120, Co     # Nếu $t0 == 'x' (ASCII 120), đi đến Co
CheckI:
        beq $t3, 3, CheckT   # Nếu $t3 == 3, kiểm tra ký tự 'T'
        beq $t0, 105, Co     # Nếu $t0 == 'i' (ASCII 105), đi đến Co
CheckT:
        beq $t3, 4, Encrypt2  # Nếu $t3 == 4, đi đến Encrypt2
        beq $t0, 116, Co     # Nếu $t0 == 't' (ASCII 116), đi đến Co
Encrypt:
        addi $t3, $zero, 0   # Đặt $t3 về 0
Encrypt2:
ChuHoa:
        bgt $t0, 90, ChuThuong # Nếu $t0 > 'Z' (ASCII 90), đi đến ChuThuong
        blt $t0, 65, ChuThuong # Nếu $t0 < 'A' (ASCII 65), đi đến ChuThuong
        addi $t0, $t0, 32    # Chuyển ký tự hoa thành thường
        j ShowKey
ChuThuong:
        bgt $t0, 122, ChuSo    # Nếu $t0 > 'z' (ASCII 122), đi đến ChuSo
        blt $t0, 97, ChuSo     # Nếu $t0 < 'a' (ASCII 97), đi đến ChuSo
        addi $t0, $t0, -32     # Chuyển ký tự thường thành hoa
        j ShowKey
ChuSo:
        bgt $t0, 57, Khac      # Nếu $t0 > '9' (ASCII 57), đi đến Khac
        blt $t0, 48, Khac      # Nếu $t0 < '0' (ASCII 48), đi đến Khac
        j ShowKey
Khac:
        addi $t0, $zero, 42    # Chuyển ký tự khác thành '*'
ShowKey:
        sw $t0, 0($s0)         # Hiển thị ký tự
        nop
        beq $t3, 4, Exit       # Nếu $t3 == 4, kết thúc chương trình
        j loop
Co:
        addi $t3, $t3, 1       # Tăng giá trị của $t3
        j Encrypt2
Exit:
        li $v0, 10             # Kết thúc chương trình
        syscall
```

Kết quả thu được:


Keyboard and Display MMIO Simulator, Version 1.4

Keyboard and Display MMIO Simulator

DISPLAY: Store to Transmitter Data 0xffff000c, cursor 24, area 95 x 10

```
HELOO*IAM*HIEU*2022*EXIT
```

☑ DAD

Fixed transmitter delay, select using slider

Delay length: 5 instruction executions

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

```
heloo iam hieu 2022 exit heieir
```

Assignment 3:

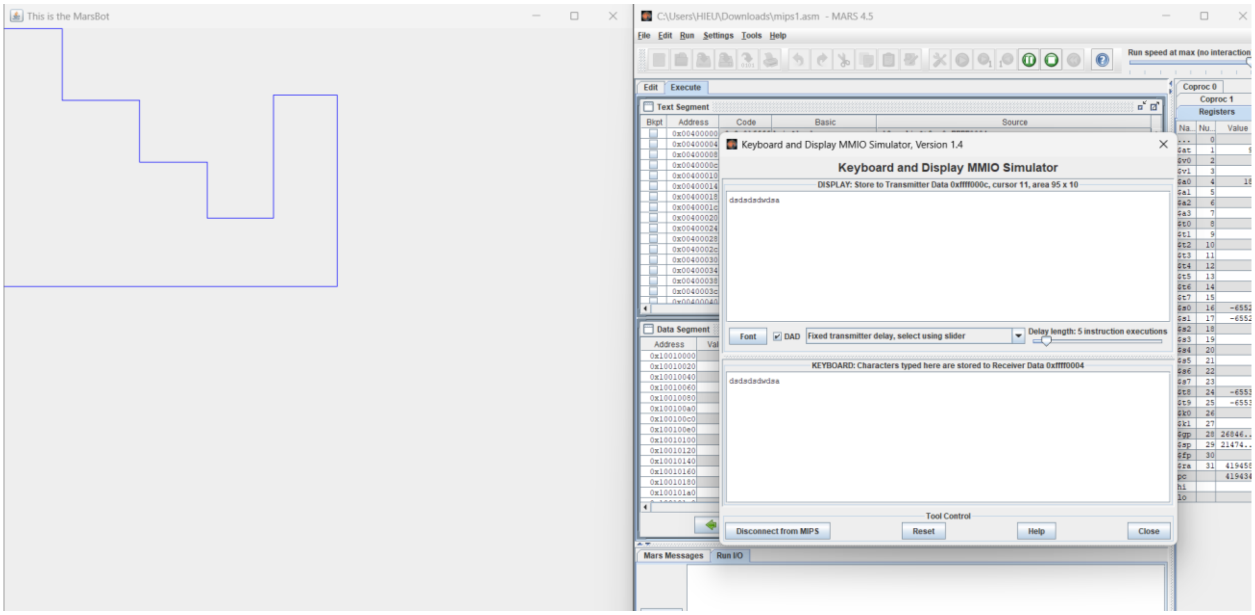
GO:

```
li $at, MOVING            # Địa chỉ của cổng MOVING
addi $k0, $zero, 1        # Đặt giá trị 1 vào $k0 (chạy MarsBot)
sb $k0, 0($at)            # Bắt đầu di chuyển MarsBot
jr $ra
```

ROTATE:

```
 li $at, HEADING            # Địa chỉ của cổng HEADING
 sw $a0, 0($at)             # Xoay MarsBot theo góc $a0
 jr $ra

STOP:
 li $at, MOVING             # Địa chỉ của cổng MOVING
 sb $zero, 0($at)           # Dừng MarsBot
 jr $ra

TRACK:
 li $at, LEAVETRACK         # Địa chỉ của cổng LEAVETRACK
 addi $k0, $zero, 1         # Đặt giá trị 1 vào $k0 (vẽ dấu vết)
 sb $k0, 0($at)             # Bắt đầu vẽ dấu vết
 jr $ra

UNTRACK:
 li $at, LEAVETRACK         # Địa chỉ của cổng LEAVETRACK
 sb $zero, 0($at)           # Ngừng vẽ dấu vết
 jr $ra
```

Kết quả thu được:


This is the MarsBot
C:\Users\HIEU\Downloads\mips1.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction
Edit Execute
Text Segment
Bkpt Address Code Basic Source
Data Segment
Address Val
Coproc 0
Coproc 1
Registers
Mars Messages Run I/O
Keyboard and Display MMIO Simulator, Version 1.4
Font
Disconnect from MIPS
Reset
Help
Close

Keyboard and Display MMIO Simulator

DISPLAY: Store to Transmitter Data 0xffff000c, cursor 11, area 95 x 10

```
dsdsdsdwdsa
```

Font ☑ DAD Fixed transmitter delay, select using slider

Delay length: 5 instruction executions

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

```
dsdsdsdwdsa
```

Tool Control