Báo cáo thực hành KTMT WEEK 11

Họ và tên: Lê Anh Dũng

MSSV: 20194522

Bài 1:

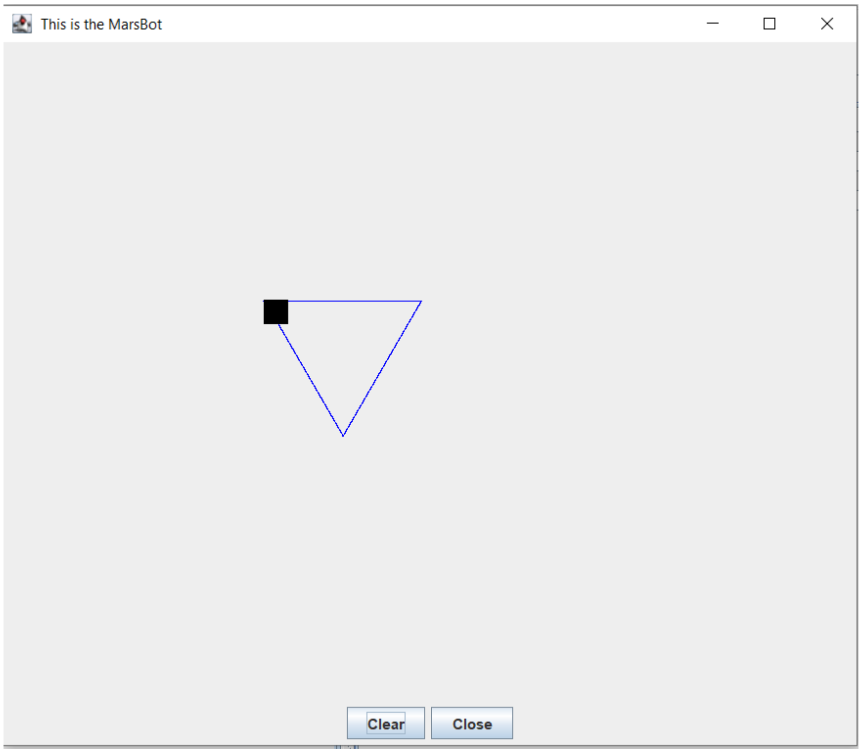
a) **Tam giác**

```
.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, $zero, 135            # Marsbot rotates 135* and start running ( move to
center of display)
        jal ROTATE
        jal GO

sleep0: addi $v0,$zero,32               # Keep running by sleeping in 12000 ms
        li $a0,14000
        syscall
        jal STOP

goE1:   addi $a0, $zero, 90             # Marsbot rotates 90*
        jal ROTATE
        jal TRACK                       # draw first edge
        jal GO

sleep1: addi $v0,$zero,32               # Keep running by sleeping in 6000 ms
        li $a0,6000
        syscall

        jal STOP
        jal UNTRACK                     # keep old track
goE2:   addi $a0, $zero, 210            # Marsbot rotates 210*
        jal ROTATE
        jal TRACK                       # draw second edge
        jal GO
```

```
sleep2: addi $v0,$zero,32               # Keep running by sleeping in 6000 ms
        li $a0,6000
        syscall

        jal STOP                        # keep old track
        jal UNTRACK
goE3:   addi $a0, $zero, 330            # Marsbot rotates 330*
        jal ROTATE
        jal TRACK                       # draw third edge
        jal GO
sleep3: addi $v0,$zero,32               # Keep running by sleeping in 6000 ms
        li $a0,6000
        syscall
        jal UNTRACK
        jal STOP                        # keep old track
end_main:
        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
```

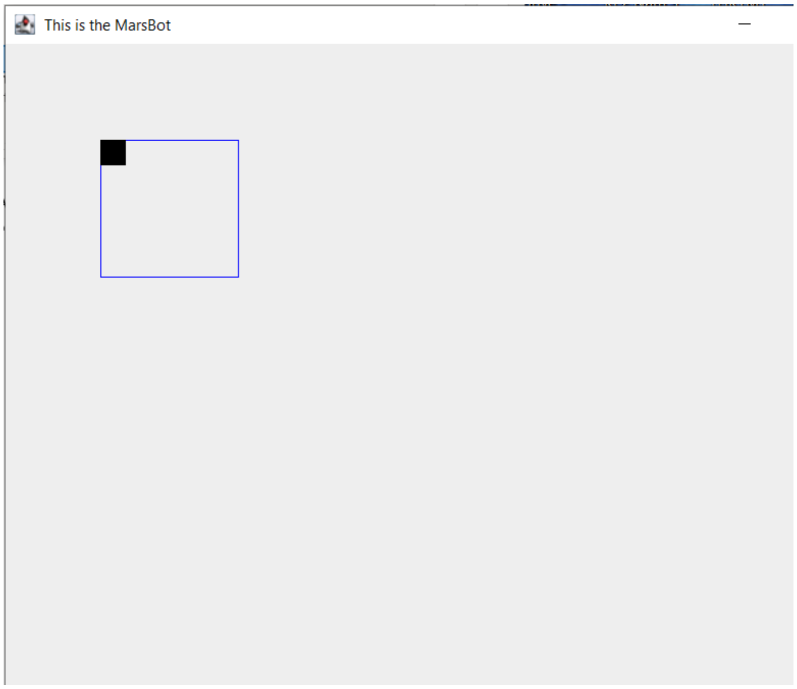
Kết quả:



b) 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 MarsBot
.text
main:
        #jal TRACK                       # draw track line
        addi $a0, $zero, 135    # Marsbot rotates 90* and start running
        jal ROTATE
        jal GO
        addi $v0,$zero,32               # Keep running by sleeping in 1000 ms
        li $a0,5000
        syscall

goRIGHT:
        addi $a0, $zero, 90     # Marsbot rotates 180*
        jal ROTATE
sleep1:
        jal TRACK
        addi $v0,$zero,32               # Keep running by sleeping in 1000 ms
        li $a0,5000
        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
goLEFT:
        addi $a0, $zero, 270    # Marsbot rotates 270*
        jal ROTATE
sleep3:
        addi $v0,$zero,32               # Keep running by sleeping in 1000 ms
        li $a0,5000
        syscall
        jal UNTRACK             # keep old track
        jal TRACK                       # and draw new track line
goUP:
        addi $a0, $zero, 0      # Marsbot rotates 120*
        jal ROTATE

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

        jal STOP

end_main:
        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:



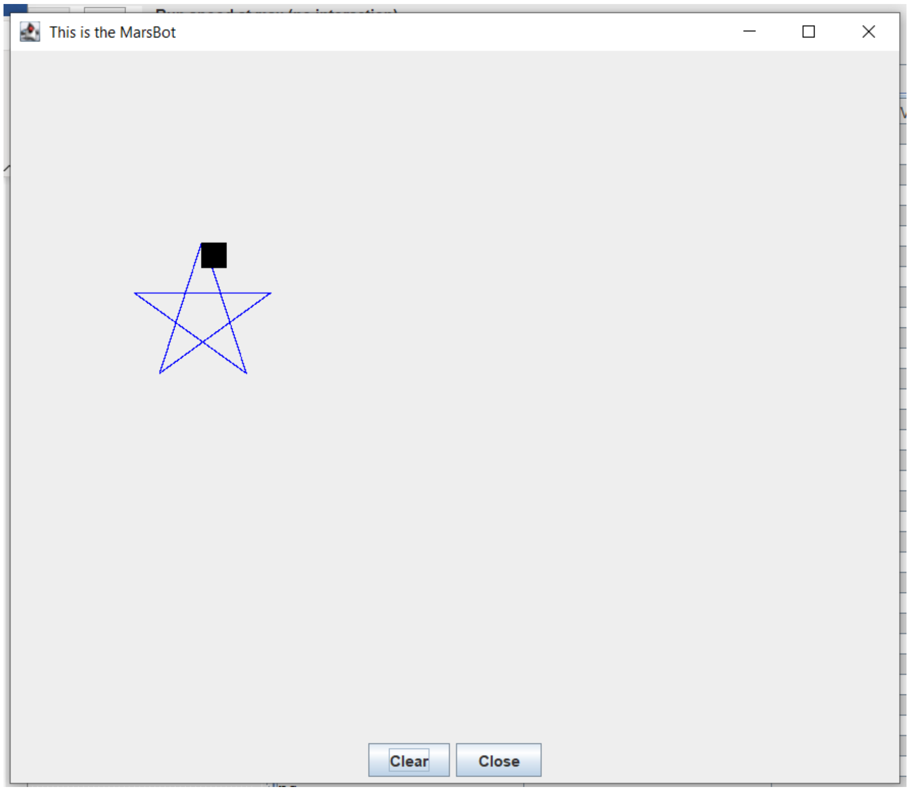
c) Hình sao

```
.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, 135    # Marsbot rotates 90* and start running
        jal ROTATE
        jal GO
        addi $v0,$zero,32               # Keep running by sleeping in 1000 ms
        li $a0,10000
        syscall

goDiaDownRight:
        addi $a0, $zero, 162    # Marsbot rotates 180*
        jal ROTATE
sleep1:
        jal TRACK
        addi $v0,$zero,32               # Keep running by sleeping in 1000 ms
        li $a0,5000
        syscall
        jal UNTRACK             # keep old track
        jal TRACK                       # and draw new track line
goDiaUpLeft:
        addi $a0, $zero, 306    # 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
goRight:
        addi $a0, $zero, 90     # Marsbot rotates 270*
        jal ROTATE
sleep3:
        addi $v0,$zero,32               # Keep running by sleeping in 1000 ms
        li $a0,5000
        syscall
        jal UNTRACK             # keep old track
        jal TRACK                       # and draw new track line
goDiaDownLeft:
        addi $a0, $zero, 234    # Marsbot rotates 120*
        jal ROTATE

sleep4:
        addi $v0,$zero,32               # Keep running by sleeping in 2000 ms
        li $a0,5000
        syscall
        jal UNTRACK             # keep old track
        jal TRACK
goDiaUPRight:
        addi $a0, $zero, 18     # Marsbot rotates 270*
        jal ROTATE
sleep5:
        addi $v0,$zero,32               # Keep running by sleeping in 1000 ms
        li $a0,5000
        syscall
        jal UNTRACK             # keep old track



        jal STOP
        jal TRACK
end_main:
        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:



Bài 2: Nhập ký tự ở Keyboard và hiển thị ở Display: nhập ký tự thường => hiển thị ký tự hoa tương ứng, nhập ký tự hoa => hiển thị ký tự thường tương ứng, nhập ký tự số thì giữ nguyên, nhập ký tự khác => hiển thị ký tự *

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

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

Encrypt:

li $t1 , 'a'

blt $t0 , $t1,      case2

bge $t0 , 123 , case4

sub $t0,$t0,32

j ShowKey

case2:

li $t4 , 91

bge $t0 , $t4 , case4

li $t2 , 'A'          #get value of 'A'

blt $t0 ,$t2,case3

addi $t0,$t0,32

j ShowKey

case3:

li $t3 , 58

bge $t0 , $t3 ,case4

li $t2 , '0'          #get value of '0'

blt $t0 ,$t2,case4

addi $t0,$t0,0

j ShowKey

case4:

li $t0, '*'

j ShowKey
```

```
ShowKey: sw $t0, 0($s0) # show key

nop

j loop
```

Kết quả:

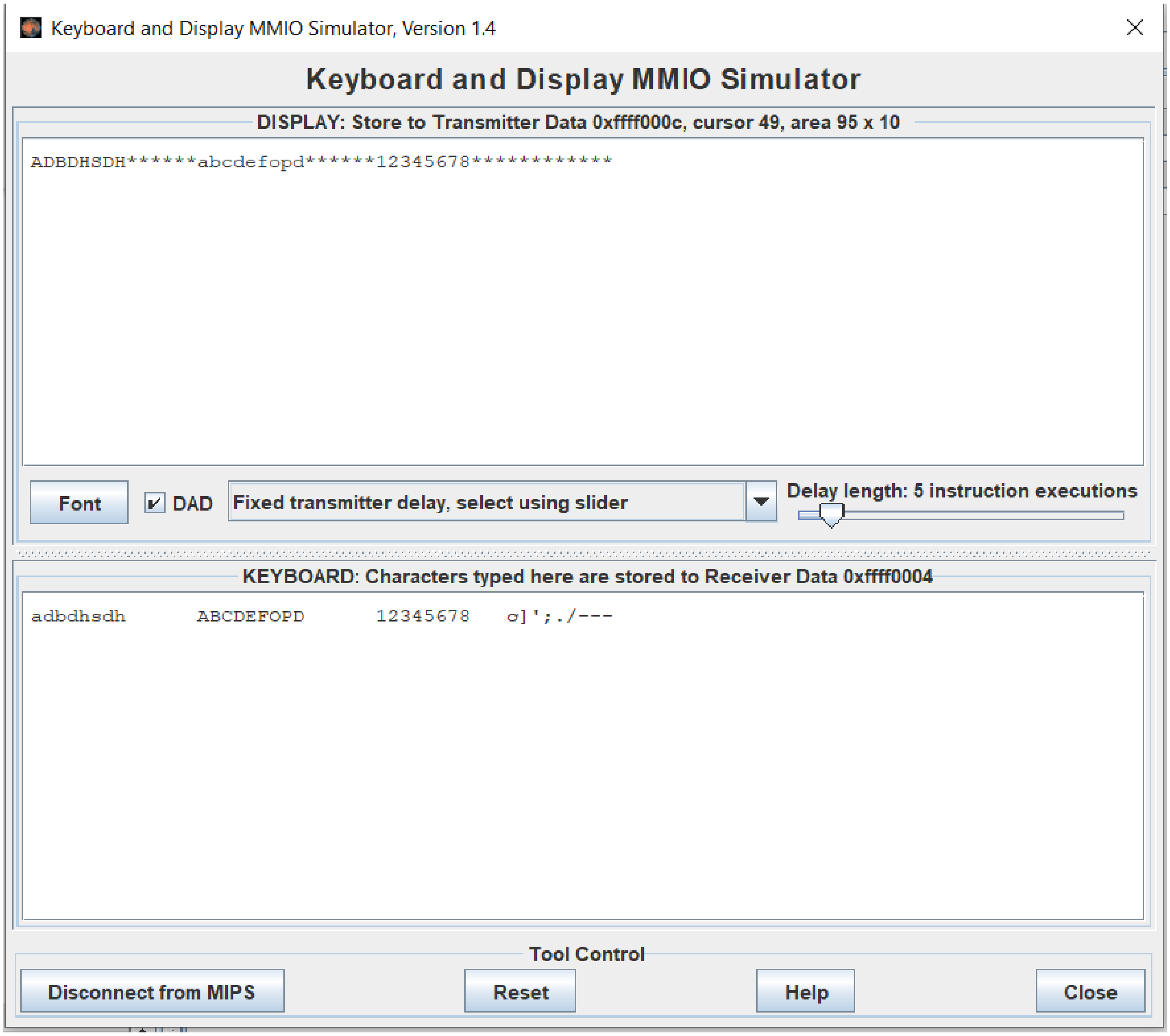
Bảng mã ascii :

Chữ thường: 97 -> 122, trừ đi 32 thành chữ hoa

Chữ hoa: 65-> 90, cộng thêm 32 thành chữ thường

Chữ số: 48-> 57, giữ nguyên

Còn các kí tự còn lại đổi thành * = ‘48’



Keyboard and Display MMIO Simulator, Version 1.4
Keyboard and Display MMIO Simulator
Font
Tool Control
Disconnect from MIPS
Reset
Help
Close

DISPLAY: Store to Transmitter Data 0xffff000c, cursor 49, area 95 x 10

```
ADBDHSDH******abcdefopd******12345678************
```

☑ DAD

Fixed transmitter delay, select using slider

Delay length: 5 instruction executions

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

```
adbdhsdh      ABCDEFOPD      12345678   σ]';./---
```

Bài 3: Dùng keyboard điều khiển marsbot

+ Space: bắt đầu / dừng di chuyển

+ W: đi lên, S: đi xuống, A: sang trái, D: sang phải

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 ofMarsBot
.eqv WHEREY 0xffff8040 # Integer: Current y-location ofMarsBot
.text

main:
addi $a0, $zero, 120 # Marsbot rotates 90* and startrunning
jal ROTATE
jal GO
jal TRACK # draw track line
sleep0: addi $v0,$zero,32 # Keep running by sleeping in 1000 ms
li $a0,1000
syscall
li $s1,'w' #w
li $s2,'a' #a
li $s3,'d' #d
li $s4,'s' #s
li $s5,' '
li $s6,'e'
li $t6, 1
start:
        #la $a0,mess
        #li $v0,4
        #syscall
        #lw $a1, 0xffff0004

        li $v0,12
        syscall
        add $a1,$zero,$v0

        beq $a1,$s5,caseSpace
        li $t8,MOVING
        beq $t8,$zero,start
        beq $a1,$s1,caseW
        beq $a1,$s2,caseA
        beq $a1,$s3,caseD
        beq $a1,$s4,caseS
        beq $a1,$s6,exit
        j start
caseW:
        addi $a0, $zero, 0 # Marsbot rotates 90* and startrunning
        jal ROTATE

sleep1:
        jal UNTRACK           # keep old track
        jal TRACK                     # and draw new track line
        addi $v0,$zero,32 # Keep running by sleeping in 1000 ms
        li $a0,100
        syscall

        j start
caseS:
        addi $a0, $zero, 180 # Marsbot rotates 90* and startrunning
        jal ROTATE

sleep2:
        jal UNTRACK # keep old track
        jal TRACK # and draw new track line
        addi $v0,$zero,32 # Keep running by sleeping in 1000 ms
        li $a0,100
        syscall

        j start
caseD:
        addi $a0, $zero, 90 # Marsbot rotates 90* and startrunning
        jal ROTATE

sleep3:
        jal UNTRACK # keep old track
        jal TRACK # and draw new track line
        addi $v0,$zero,32 # Keep running by sleeping in 1000 ms
        li $a0,100
        syscall
        j start
caseA:
        addi $a0, $zero, 270 # Marsbot rotates 90* and startrunning
        jal ROTATE

sleep4:
        jal UNTRACK # keep old track
        jal TRACK # and draw new track line
        addi $v0,$zero,32 # Keep running by sleeping in 1000 ms
        li $a0,100
        syscall
```

```
        j start
caseSpace:
        beq $t6, $zero, goto
        jal UNTRACK # keep old track
        li $t6,0
        jal STOP
        j start

goto:
        li $t6, 1
        jal TRACK # and draw new track line
        jal GO
        j start
end_main:
exit:
        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
#---------------------------------------------------------
```

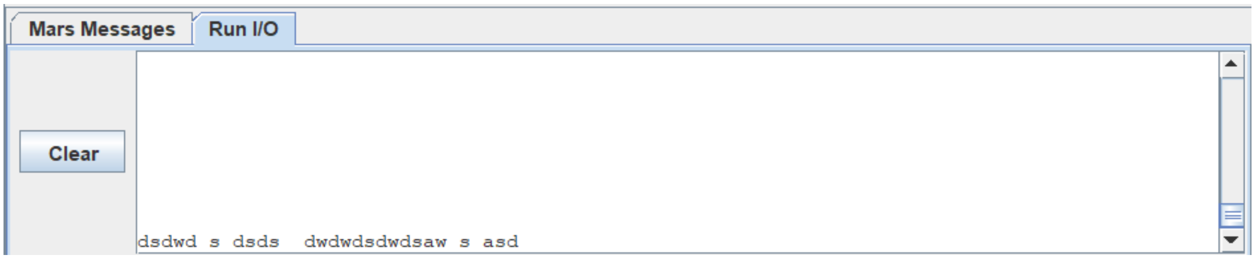
```
STOP: li $at, MOVING # change MOVING port to 0
sb $zero, 0($at) # to stop
jr $ra
#---------------------------------------------------------
# TRACK procedure, to start drawing line
# param[in] none
#---------------------------------------------------------
TRACK: li $at, LEAVETRACK # change LEAVETRACK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

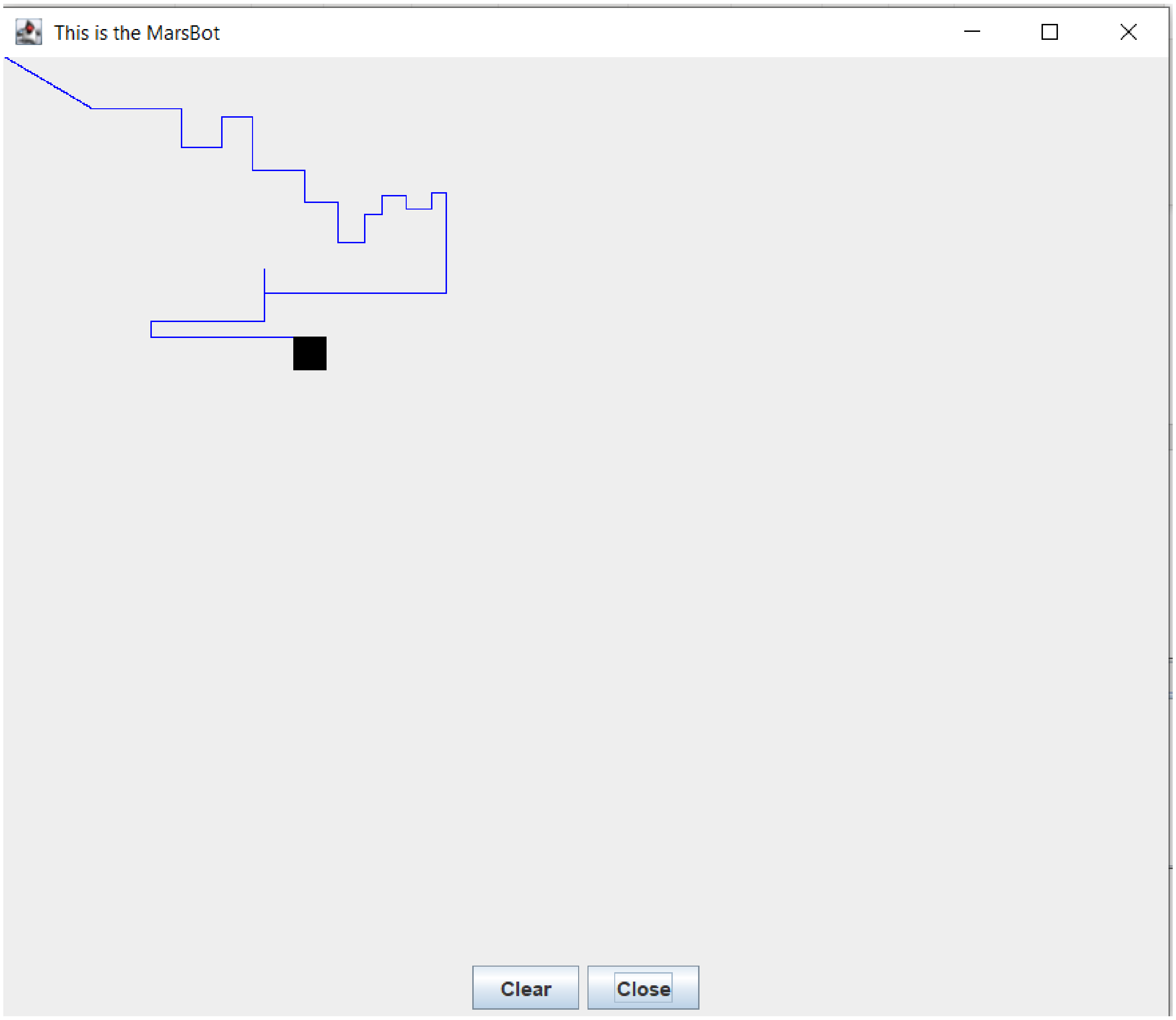
+ Space: bắt đầu / dừng di chuyển

+ w: đi lên, s: đi xuống, a: sang trái, d: sang phải

+ e: để thoát khỏi vòng lặp chương trình



```
dsdwd s dsds  dwdwdsdwdsaw s asd
```


This is the MarsBot
Clear
Close