BÁO CÁO THỰC HÀNH TUẦN 11

Lương Thị Tâm-20194663

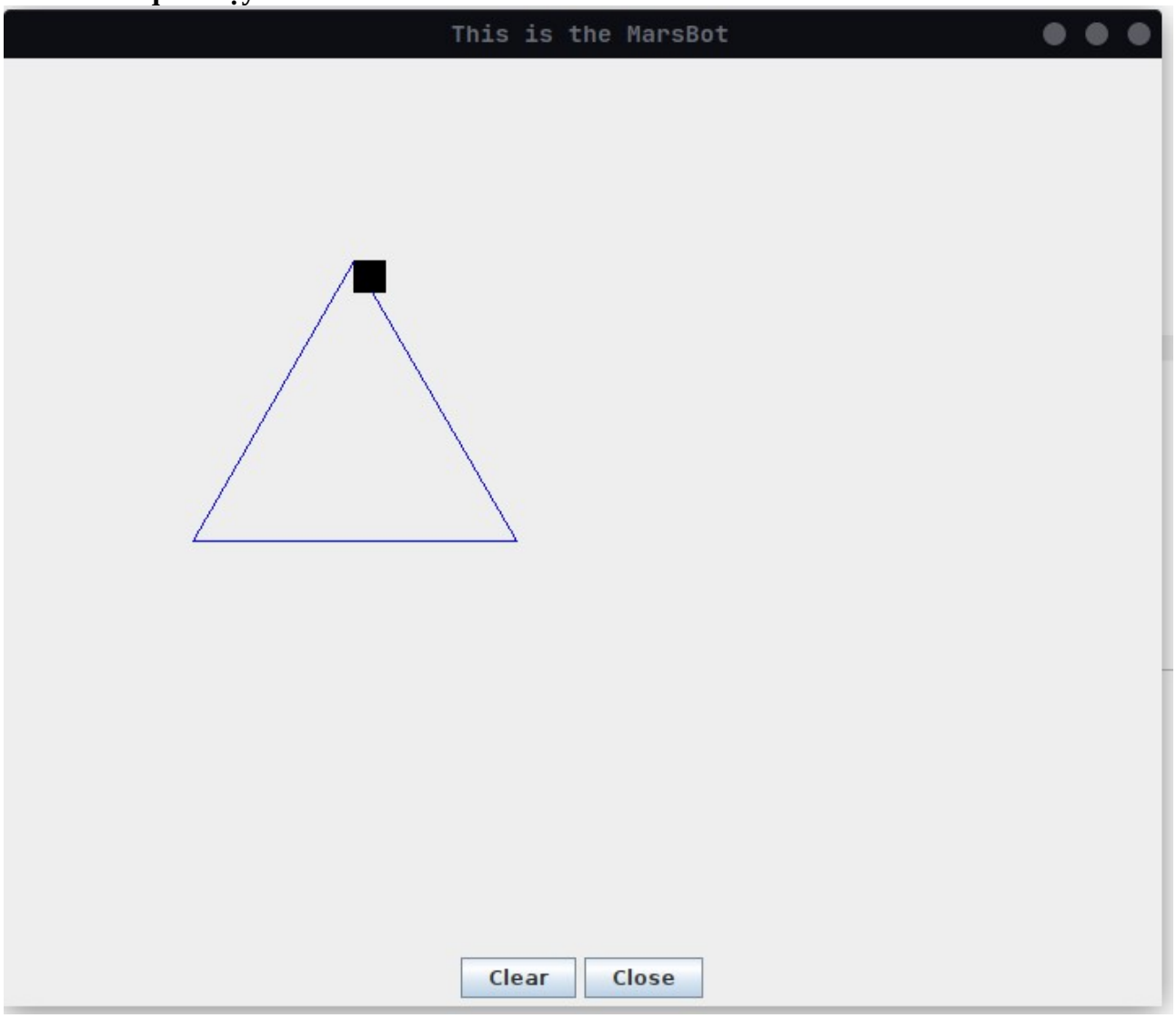
Bài 1:

- **Vẽ tam giác cân**

  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, $zero, 120# Marsbot rotates 120* and start running
     jal   ROTATE
     jal   GO
sleep1: addi    $v0,$zero,32    # Keep running by sleeping in10000 ms
     li  $a0,10000
     syscall
     jal     TRACK# and draw new track line
go1: addi    $a0, $zero, 150# Marsbot rotates 150*
     jal     ROTATE
sleep2: addi $v0,$zero,32  # Keep running by sleeping in 8000 ms
     li      $a0,8000
     syscall
     jal     UNTRACK         # keep old track
     jal     TRACK           # and draw new track line
go2: addi  $a0, $zero, 270# Marsbotrotates 270*
     jal   ROTATE
sleep3: addi    $v0,$zero,32    # Keep running by sleeping in 1000 ms
     li      $a0,8000
     syscall
     jal     UNTRACK         # keep old track
     jal     TRACK           # and draw new track line
go3:addi  $a0, $zero, 30 # Marsbot rotates 30*
     jal   ROTATE
sleep4: addi    $v0,$zero,32    # Keep running by sleeping in 2000 ms
     li      $a0,8000
     syscall
     jal     UNTRACK         # keep old track
     #jal     TRACK           # and draw new track line
     jal STOP
     li $v0, 10
     syscall
end_main:
#---------------------------------------------------------
# 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
```

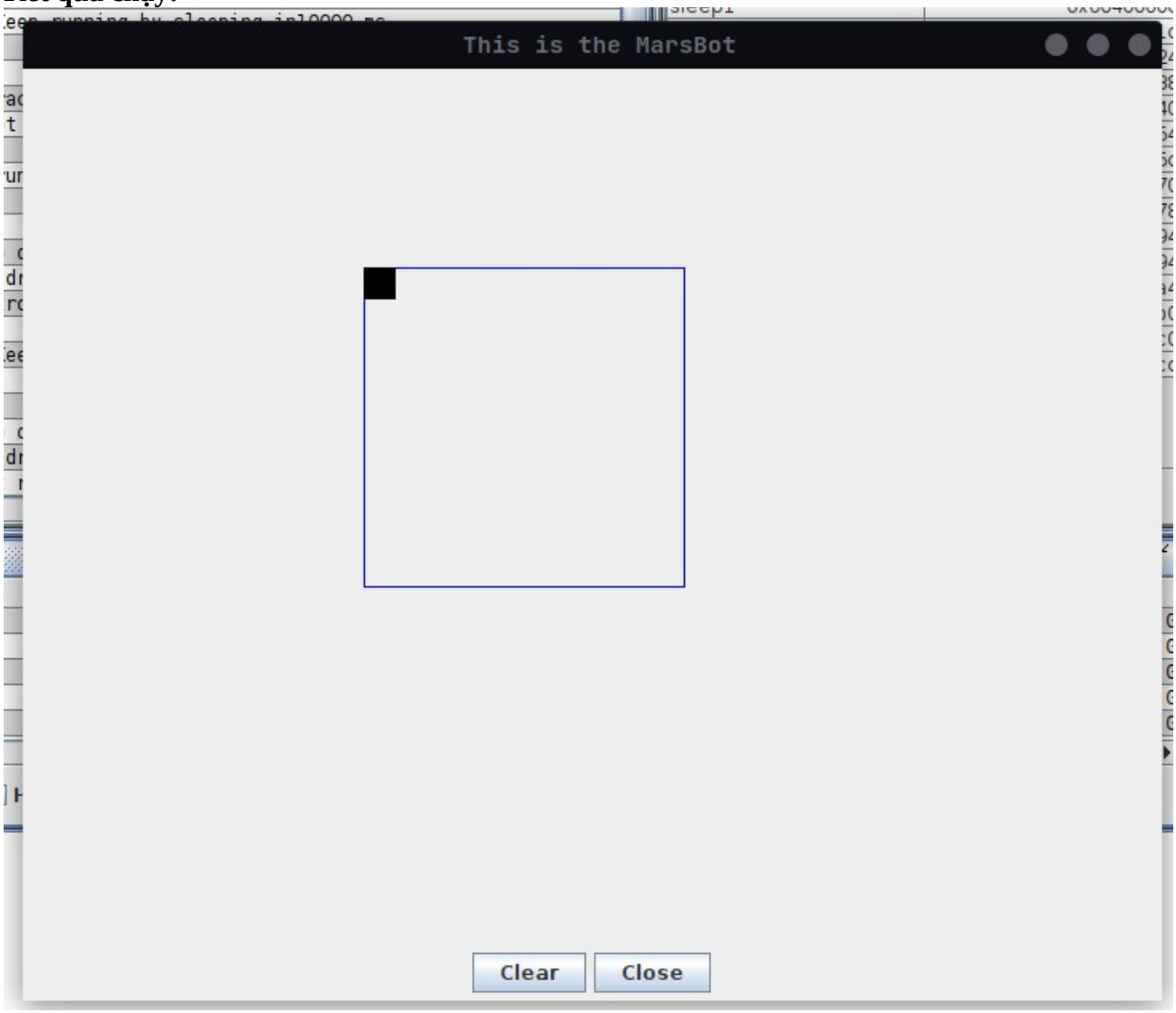
Kết quả chạy:



- **Vẽ hình vuông**
  Mã nguồn:

```
.text
main:  #jal  TRACK# draw trackline
     addi  $a0, $zero, 120# Marsbot rotates 120* and start running
     jal   ROTATE
     jal   GO
sleep1: addi    $v0,$zero,32    # Keep running by sleeping in10000 ms
     li  $a0,10000
     syscall
     #jal     UNTRACK# keep old track
     jal     TRACK# and draw new track line
go1: addi    $a0, $zero, 90# Marsbot rotates 90*
     jal     ROTATE
sleep2: addi $v0,$zero,32  # Keep running by sleeping in 8000 ms
     li      $a0,8000
     syscall
     jal     UNTRACK         # keep old track
     jal     TRACK           # and draw new track line
go2: addi  $a0, $zero, 180# Marsbotrotates 180*
     jal   ROTATE
sleep3: addi    $v0,$zero,32    # Keep running by sleeping in 8000 ms
     li      $a0,8000
     syscall
     jal     UNTRACK         # keep old track
     jal     TRACK           # and draw new track line
go3:addi  $a0, $zero, 270 # Marsbot rotates 270*
     jal   ROTATE
sleep4: addi    $v0,$zero,32    # Keep running by sleeping in 8000 ms
     li      $a0,8000
     syscall
     jal     UNTRACK         # keep old track
     jal     TRACK           # and draw new track line
go4:addi  $a0, $zero, 0 # Marsbot rotates 0*
     jal   ROTATE
sleep5: addi    $v0,$zero,32    # Keep running by sleeping in 8000 ms
     li      $a0,8000
     syscall
     jal     UNTRACK         # keep old track
     jal STOP
     li $v0, 10
     syscall
end_main:
```

Kết quả chạy:

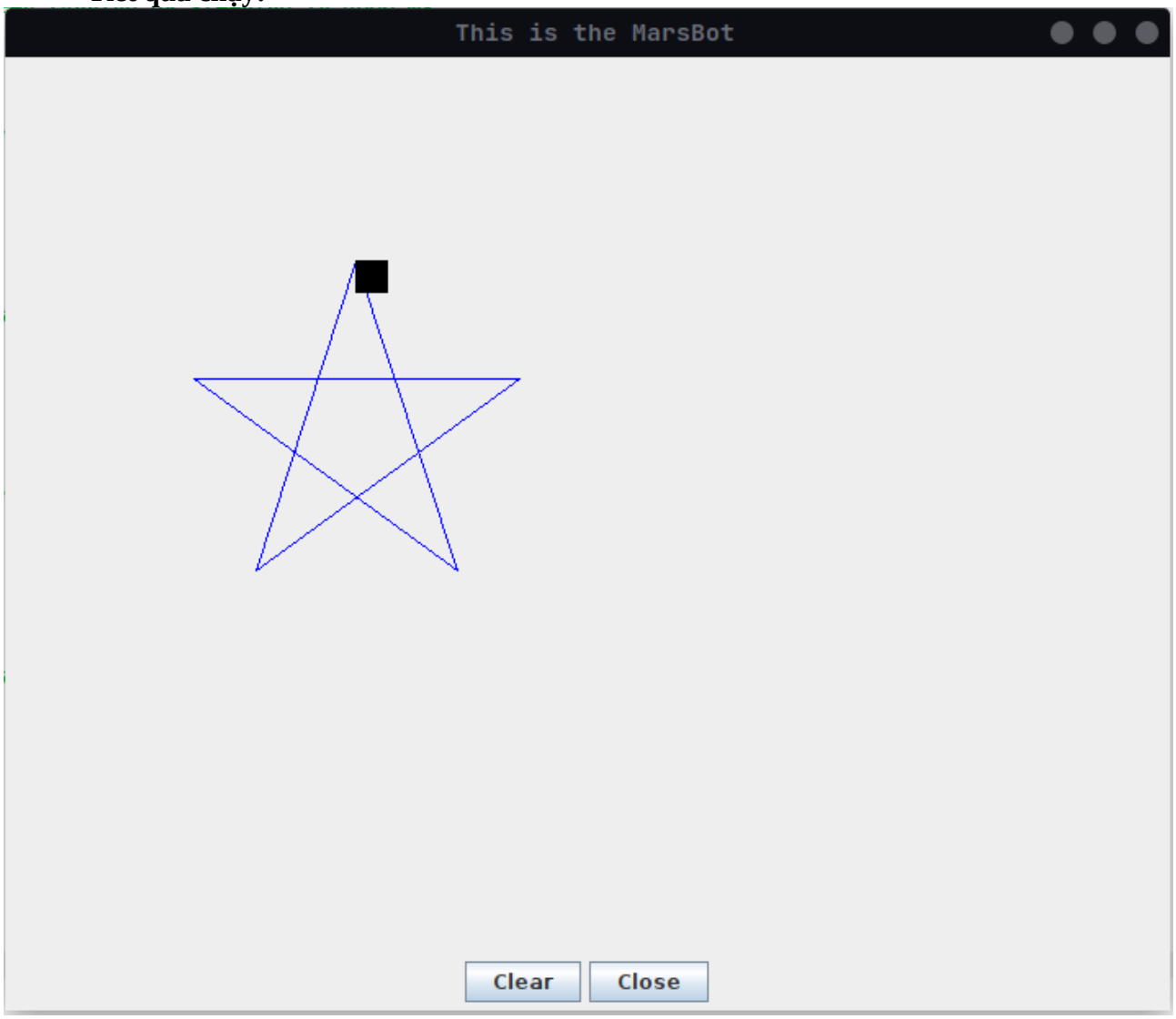


- **Vẽ ngôi sao năm cánh**
  Mã nguồn:

```
.text
main:addi  $a0, $zero, 120# Marsbot rotates 120* and start running
     jal   ROTATE
     jal   GO
sleep1: addi    $v0,$zero,32    # Keep running by sleeping in10000 ms
     li  $a0,10000
     syscall
     jal     TRACK# and draw new track line
go1: addi    $a0, $zero, 162# Marsbot rotates 162*
     jal     ROTATE
sleep2: addi $v0,$zero,32  # Keep running by sleeping in 8000 ms
     li      $a0,8000
     syscall
     jal     UNTRACK         # keep old track
     jal     TRACK           # and draw new track line
go2: addi  $a0, $zero, 306# Marsbotrotates 306*
     jal   ROTATE
sleep3: addi    $v0,$zero,32    # Keep running by sleeping in 8000 ms
     li      $a0,8000
     syscall
     jal     UNTRACK         # keep old track
     jal     TRACK           # and draw new track line
go3:addi  $a0, $zero, 90 # Marsbot rotates 90*
     jal   ROTATE
sleep4: addi    $v0,$zero,32    # Keep running by sleeping in 8000 ms
     li      $a0,8000
     syscall
     jal     UNTRACK         # keep old track
     jal     TRACK           # and draw new track line
go4:addi  $a0, $zero, 234 # Marsbot rotates 234*
     jal   ROTATE
sleep5: addi    $v0,$zero,32    # Keep running by sleeping in 8000 ms
     li      $a0,8000
     syscall
     jal     UNTRACK         # keep old track
          jal     TRACK           # and draw new track line
go5:addi  $a0, $zero, 18 # Marsbot rotates 18*
     jal   ROTATE
sleep6: addi    $v0,$zero,32    # Keep running by sleeping in 8000 ms
     li      $a0,8000
```

```
     li      $a0,8000
     syscall
     jal     UNTRACK         # keep old track
     jal STOP
     li $v0, 10
     syscall
end_main:
```

Kết quả chạy:



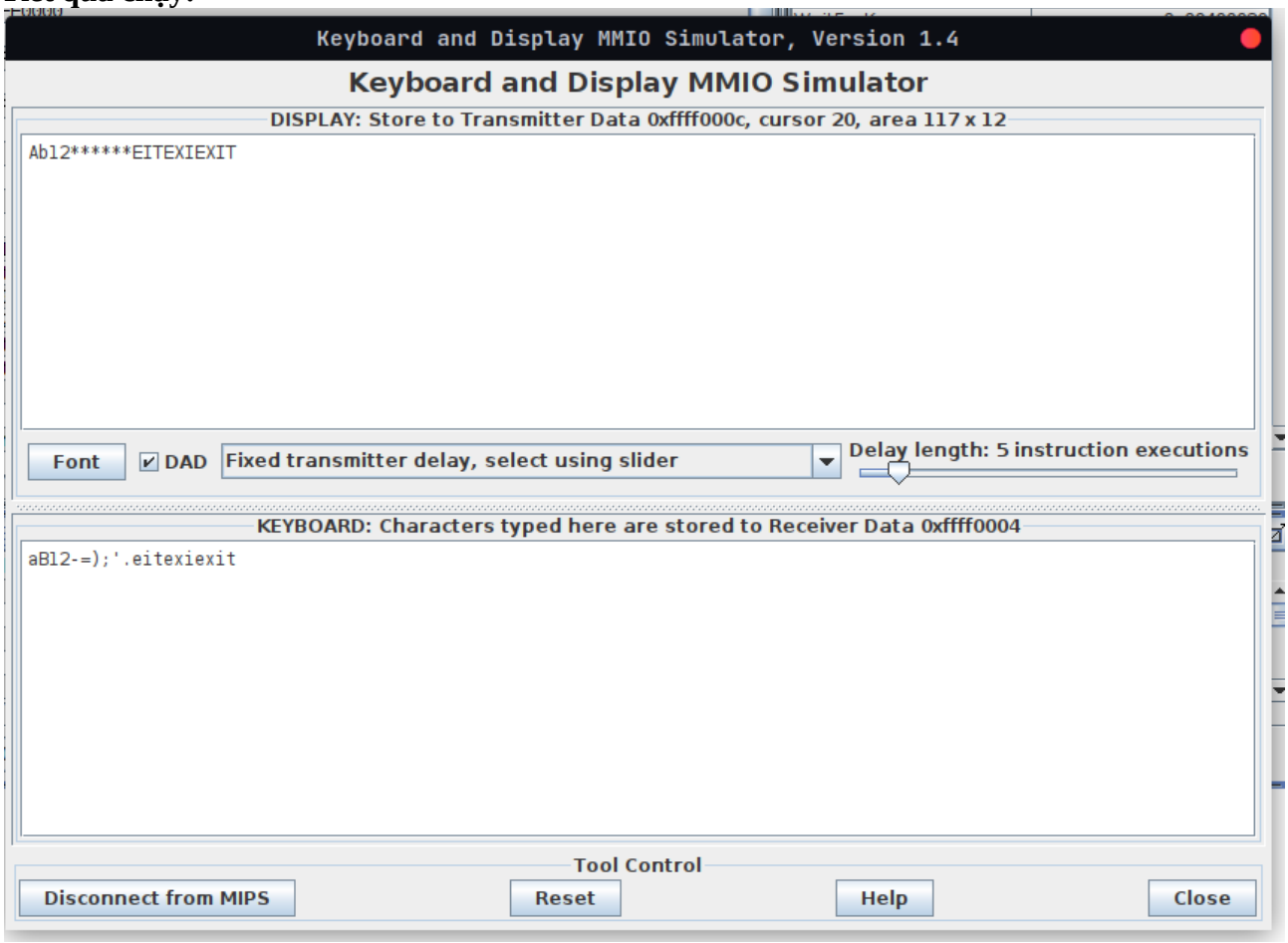
Bài 2:

Mã nguồn:

```
.eqv KEY_CODE   0xFFFF0004       # ASCII code from key
.eqv KEY_READY  0xFFFF0000       # =1 if has a new keycode ?
                                 #Auto clear after lwboard, 1 byte
.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
.data
exit: .asciiz "exit"
.text
     li $k0,  KEY_CODE
     li $k1,  KEY_READY
     li $s0, DISPLAY_CODE
     li $s1, DISPLAY_READY
     li $t3, 0
     la $s3, exit
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
             bgt $t0, 47, check1
             j Encrypt
             nop
check1:      bgt $t0, 57, check2
             addi $t0, $t0, 0
             j ShowKey
             nop
check2:      bgt $t0, 64, check3
             j Encrypt
             nop
check3:      bgt $t0, 90, check4
             addi $t0, $t0, 32
             j ShowKey
             nop
check4:      bgt $t0, 96, check5
             j Encrypt
             nop
```

```
check5:      bgt $t0, 122, Encrypt
             add $s4, $t3, $s3
             lb $t4, 0($s4)
             beq $t4, $t0, next
             beq $t0, 101, next2
             addi $t3, $t3, 0
             addi $t0, $t0, -32
             j ShowKey
             nop
next:        addi $t3, $t3, 1
             addi $t0, $t0, -32
             j ShowKey
             nop
next2:       addi $t3, $0, 1
             addi $t0, $t0, -32
             j ShowKey
             nop
Encrypt:     addi $t0, $0, 42 # change input key

ShowKey:     sw $t0, 0($s0)# show key
             nop
             nop
             beq $t3, 4, end_main
             j loop
end_main:
```

Kết quả chạy:


Keyboard and Display MMIO Simulator, Version 1.4
Keyboard and Display MMIO Simulator
DISPLAY: Store to Transmitter Data 0xffff000c, cursor 20, area 117 x 12
Ab12******EITEXIEXIT
Font
DAD
Fixed transmitter delay, select using slider
Delay length: 5 instruction executions
KEYBOARD: Characters typed here are stored to Receiver Data 0xffff0004
aB12-=);'.eitexiexit
Tool Control
Disconnect from MIPS
Reset
Help
Close

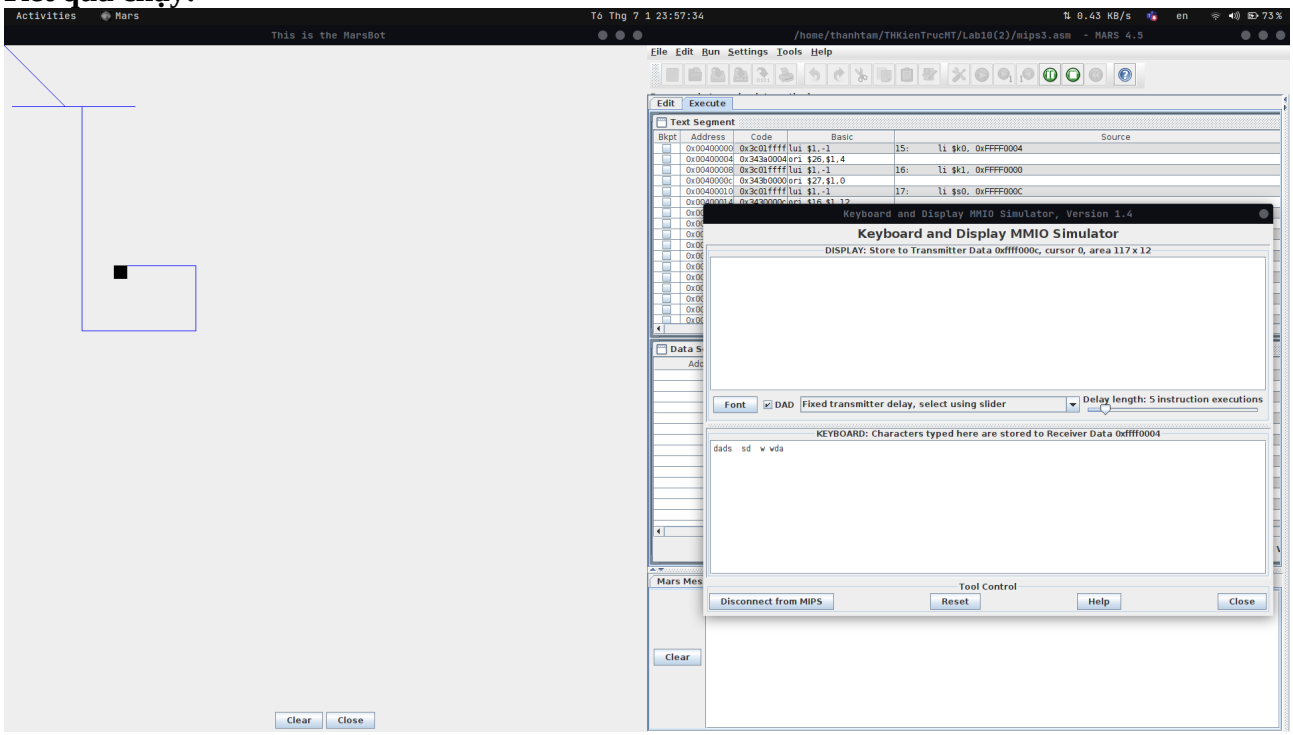
Bài 3:

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
.eqv HEADING 0xffff8010 # Integer: An angle between 0 and 359
.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
.data
.text
     li $k0, KEY_CODE
     li $k1, KEY_READY
     li $s0, DISPLAY_CODE
     li $s1, DISPLAY_READY

     addi $a0, $zero, 135          #Marsbot rotates 135* and start
     jal ROTATE
     jal TRACK           #Start draw
     jal GO
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
          addi $v0, $0, 1
          addi $a0, $t0, 0
          syscall
Control:
          jal UNTRACK         # keep old track
          li   $a0, MOVING
          lb   $a0, 0($a0)
SPACE:    bne  $t0, ' ', UP
          bne  $a0, $0, stop
go:       jal  GO
          j    continue
stop:     jal  STOP
```

```
40              j    continue
41  UP:         bne  $t0, 'w', DOWN
42              addi $a0, $zero, 0
43              j    rotate
44  DOWN:       bne  $t0, 's', LEFT
45              addi $a0, $zero, 180
46              j    rotate
47  LEFT:       bne  $t0, 'a', RIGHT
48              addi $a0, $zero, 270
49              j    rotate
50  RIGHT:      bne  $t0, 'd', continue
51              addi $a0, $zero, 90
52              j    rotate
53  rotate:
54              jal ROTATE
55
56              j    continue
57
58  continue:   jal TRACK           # and draw new track line
59              j loop
60  end_main:
61       addi $v0, $0, 10
62       syscall
63
64  GO:   li $at, MOVING # change MOVING port
65            addi $a0, $zero,1 # to logic 1,
66            sb $a0, 0($at) # to start running
67            jr $ra
68  STOP:       li $at, MOVING # change MOVING port to 0
69            sb $zero, 0($at) # to stop
70            jr $ra
71  TRACK:      li $at, LEAVETRACK # change LEAVETRACK port
72            addi $a0, $zero,1 # to logic 1,
73            sb $a0, 0($at) # to start tracking
74            jr $ra
75  UNTRACK:li $at, LEAVETRACK # change LEAVETRACK port to 0
76            sb $zero, 0($at) # to stop drawing tail
77            jr $ra
78  ROTATE: li $at, HEADING # change HEADING port
79            sw $a0, 0($at) # to rotate robot
80            jr $ra
81
```

Kết quả chạy:


Activities
Mars
Tô Thg 7 1 23:57:34
0.43 KB/s
en
73 %
This is the MarsBot
/home/thanhtam/THKienTrucMT/Lab10(2)/mips3.asm - MARS 4.5
File Edit Run Settings Tools Help
Edit Execute
Text Segment
Bkpt Address Code Basic Source
Keyboard and Display MMIO Simulator, Version 1.4
Keyboard and Display MMIO Simulator
DISPLAY: Store to Transmitter Data 0xffff000c, cursor 0, area 117 x 12
Font
DAD
Fixed transmitter delay, select using slider
Delay length: 5 instruction executions
KEYBOARD: Characters typed here are stored to Receiver Data 0xffff0004
dads sd w wda
Tool Control
Disconnect from MIPS
Reset
Help
Close
Clear
Clear
Close