Báo cáo thực hành KTMT tuần 10.2

Họ và tên: Đỗ Gia Huy

MSSV: 20215060

Assignment 1

1. Vẽ hình tam giác đều

1.1. Code

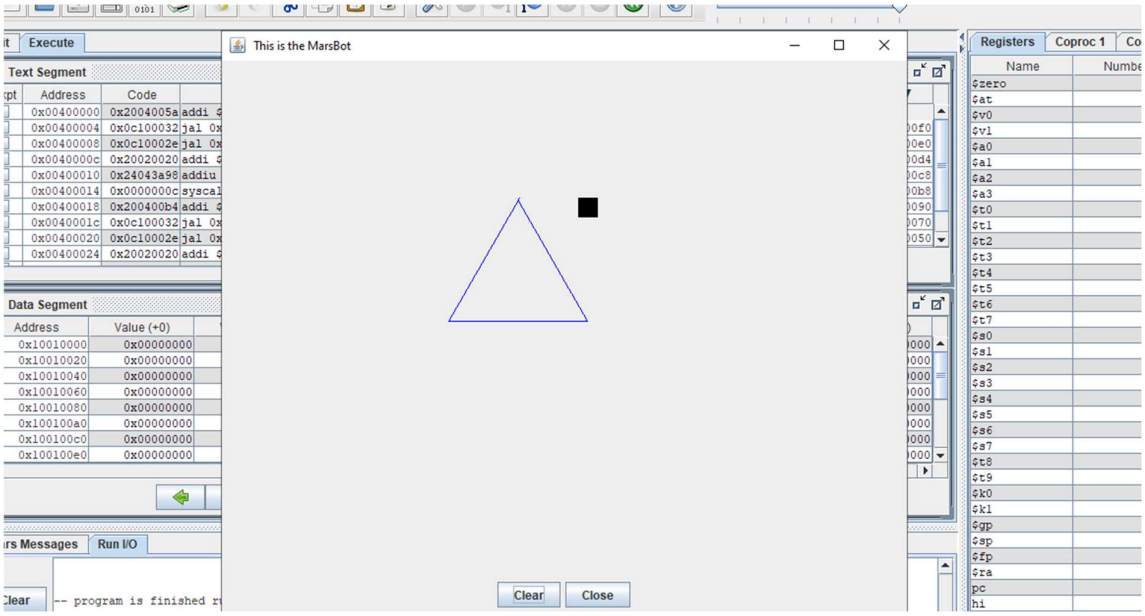
```
.eqv   HEADING      0xffff8010
.eqv  MOVING     0xffff8050
.eqv  LEAVETRACK 0xffff8020
.eqv  WHEREX     0xffff8030
.eqv   WHEREY     0xffff8040

.text
main:
        addi   $a0, $zero, 90
      jal     ROTATE
      jal     GO

        addi $v0,$zero,32    # Keep running by sleeping in 1000 ms
        li     $a0,15000
        syscall

        addi   $a0, $zero, 180
      jal     ROTATE
      jal     GO

      addi    $v0,$zero,32    # Keep running by sleeping in 1000 ms
```

```
        li     $a0,7000
        syscall
 #_____________________________________________________
 #Mio sleep la 1 doan, ve hay khong tuy nguoi lap trinh
 #_____________________________________________________
sleep1:
        addi    $a0, $zero, 150
      jal     ROTATE
      jal     GO
      jal     UNTRACK        # keep old track
      jal     TRACK          # and draw new track line
        addi    $v0,$zero,32    # Keep running by sleeping in 1000 ms
        li     $a0,7000
        syscall
sleep2:
        addi    $a0, $zero, 270
      jal     ROTATE
      jal     GO
      jal     UNTRACK        # keep old track
      jal     TRACK          # and draw new track line
      addi  $v0,$zero,32    # Keep running by sleeping in 1000 ms
        li     $a0,7000
        syscall
sleep3:
        addi    $a0, $zero, 30
      jal     ROTATE
      jal     GO
      jal     UNTRACK        # keep old track
```

```
        jal     TRACK           # and draw new track line
        addi    $v0,$zero,32    # Keep running by sleeping in 1000 ms
          li      $a0,7000
          syscall
end_main:
          jal     UNTRACK         # keep old track
          addi  $a0, $zero, 90
        jal     ROTATE
        jal     GO
          addi  $v0,$zero,32    # Keep running by sleeping in 1000 ms
          li      $a0,3000
          syscall
          jal       STOP
          li        $v0, 10
          syscall



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
```

1.2. Kết quả



2. Vẽ hình vuông

2.1. Code

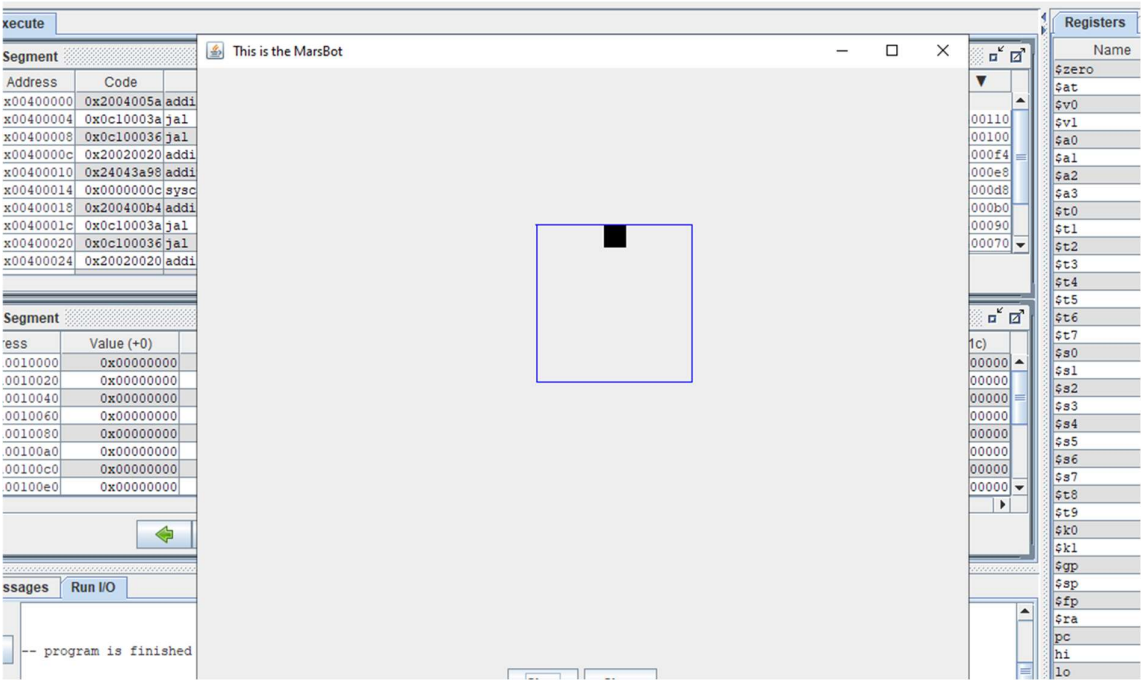
```
.eqv   HEADING      0xffff8010
.eqv   MOVING     0xffff8050
.eqv   LEAVETRACK 0xffff8020
.eqv   WHEREX     0xffff8030
.eqv   WHEREY     0xffff8040

.text
main:
        addi   $a0, $zero, 90
      jal     ROTATE
      jal     GO

        addi $v0,$zero,32    # Keep running by sleeping in 1000 ms
        li     $a0,15000
        syscall

        addi   $a0, $zero, 180
      jal     ROTATE
      jal     GO

      addi    $v0,$zero,32    # Keep running by sleeping in 1000 ms
        li     $a0,7000
        syscall

sleep1:
        addi    $a0, $zero, 90
      jal     ROTATE
      jal     GO
```

```
        jal     UNTRACK       # keep old track
        jal     TRACK         # and draw new track line
          addi    $v0,$zero,32    # Keep running by sleeping in 1000 ms
          li      $a0,7000
          syscall
sleep2:
          addi    $a0, $zero, 180
        jal     ROTATE
        jal     GO
        jal     UNTRACK       # keep old track
        jal     TRACK         # and draw new track line
        addi  $v0,$zero,32    # Keep running by sleeping in 1000 ms
          li      $a0,7000
          syscall
sleep3:
          addi    $a0, $zero, 270
        jal     ROTATE
        jal     GO
        jal     UNTRACK       # keep old track
        jal     TRACK         # and draw new track line
        addi    $v0,$zero,32    # Keep running by sleeping in 1000 ms
          li      $a0,7000
          syscall
sleep4:
          addi    $a0, $zero, 0
        jal     ROTATE
        jal     GO
        jal     UNTRACK       # keep old track
```

```
    jal     TRACK          # and draw new track line
    addi  $v0,$zero,32    # Keep running by sleeping in 1000 ms
      li        $a0,7000
      syscall
end_main:
      jal     UNTRACK        # keep old track
      addi  $a0, $zero, 90
    jal     ROTATE
    jal     GO
      addi  $v0,$zero,32    # Keep running by sleeping in 1000 ms
      li      $a0,3000
      syscall
      jal       STOP
      li        $v0, 10
      syscall



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
```

2.2. Kết quả



3. Vẽ ngôi sao 5 cánh

3.1. Code

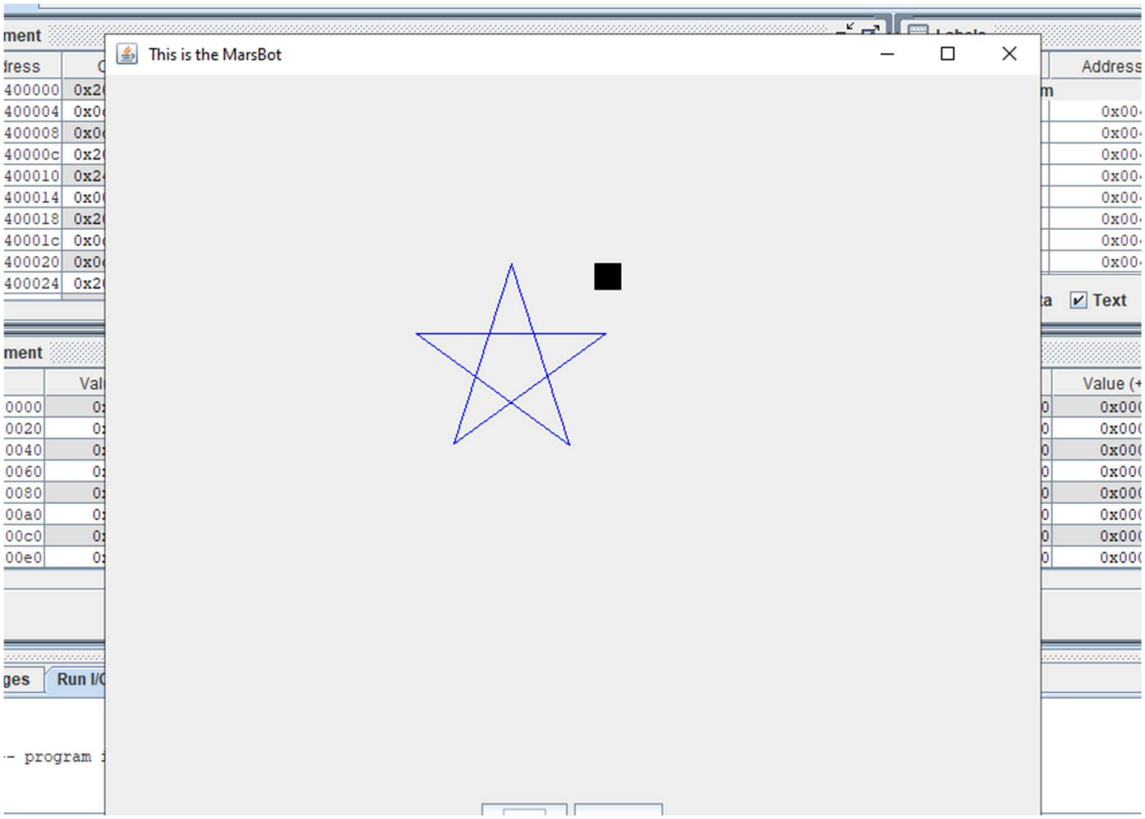
```
.eqv   HEADING      0xffff8010
.eqv  MOVING     0xffff8050
.eqv  LEAVETRACK 0xffff8020
.eqv  WHEREX     0xffff8030
.eqv   WHEREY     0xffff8040

.text
main:
        addi   $a0, $zero, 90
      jal     ROTATE
      jal     GO

        addi $v0,$zero,32    # Keep running by sleeping in 1000 ms
        li     $a0,15000
        syscall

        addi   $a0, $zero, 180
      jal     ROTATE
      jal     GO

      addi    $v0,$zero,32    # Keep running by sleeping in 1000 ms
        li     $a0,7000
        syscall

sleep1:
        addi    $a0, $zero, 162
      jal     ROTATE
```

```
        jal     GO
        jal     UNTRACK         # keep old track
        jal     TRACK           # and draw new track line
          addi    $v0,$zero,32    # Keep running by sleeping in 1000 ms
          li      $a0,7000
          syscall
sleep2:
          addi    $a0, $zero, 306
        jal     ROTATE
        jal     GO
        jal     UNTRACK         # keep old track
        jal     TRACK           # and draw new track line
        addi  $v0,$zero,32    # Keep running by sleeping in 1000 ms
          li      $a0,7000
          syscall
sleep3:
          addi    $a0, $zero, 90
        jal     ROTATE
        jal     GO
        jal     UNTRACK         # keep old track
        jal     TRACK           # and draw new track line
        addi    $v0,$zero,32    # Keep running by sleeping in 1000 ms
          li      $a0,7000
          syscall
sleep4:
          addi    $a0, $zero, 234
        jal     ROTATE
        jal     GO
```

```
        jal     UNTRACK        # keep old track
        jal     TRACK          # and draw new track line
        addi  $v0,$zero,32     # Keep running by sleeping in 1000 ms
          li        $a0,7000
          syscall
sleep5:
          addi    $a0, $zero, 18
        jal     ROTATE
        jal     GO
        jal     UNTRACK        # keep old track
        jal     TRACK          # and draw new track line
        addi    $v0,$zero,32    # Keep running by sleeping in 1000 ms
          li        $a0,7000
          syscall
end_main:
          jal     UNTRACK        # keep old track
          addi  $a0, $zero, 90
        jal     ROTATE
        jal     GO
          addi  $v0,$zero,32    # Keep running by sleeping in 1000 ms
          li     $a0,3000
          syscall
          jal      STOP
          li        $v0, 10
          syscall



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
```

3.2. Kết quả



Assignment 2

1. Code

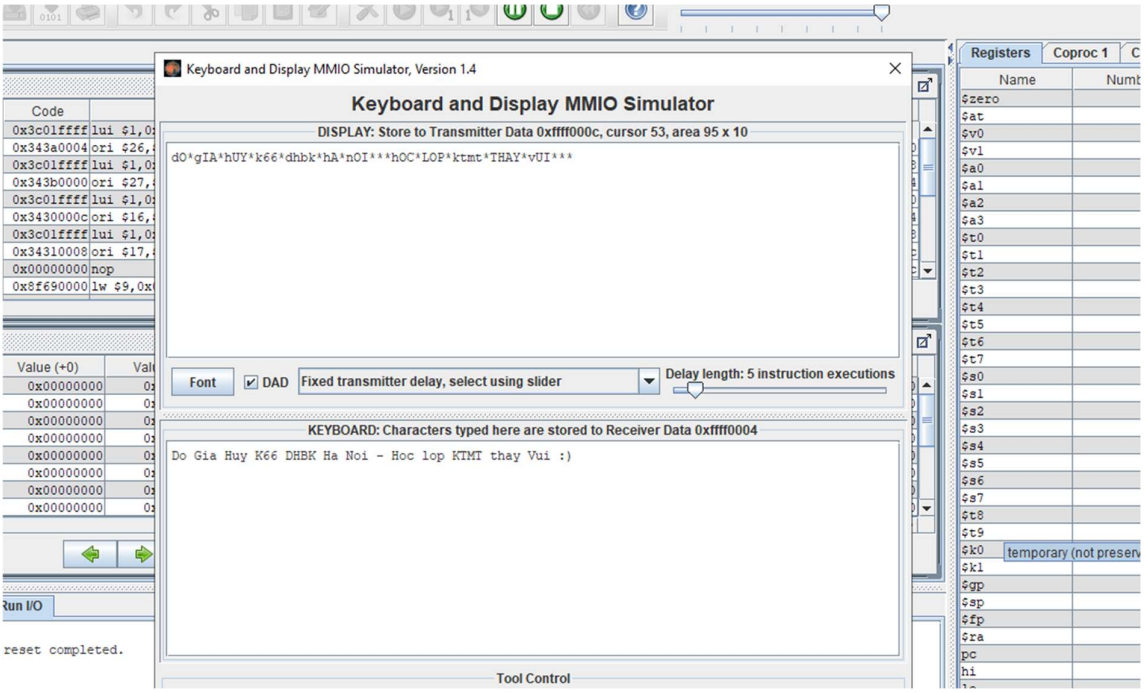
```
.eqv KEY_CODE   0xFFFF0004          # ASCII code from keyboard, 1 byte
.eqv KEY_READY  0xFFFF0000          # =1 if has a new keycode ?
                                    # Auto clear after lw
.eqv DISPLAY_CODE   0xFFFF000C   # ASCII code to show, 1 byte
.eqv DISPLAY_READY  0xFFFF0008   # =1 if the 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:
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:
ChuHoa:
        bgt    $t0, 90, ChuThuong
        blt    $t0, 65, ChuThuong
        addi   $t0, $t0, 32
        j      ShowKey
ChuThuong:
        bgt    $t0, 122, ChuSo
        blt    $t0, 97, ChuSo
        addi   $t0, $t0, -32
        j      ShowKey
ChuSo:
        bgt    $t0, 57, Khac
        blt    $t0, 48, Khac
        addi   $t0, $t0, 0
        j      ShowKey
Khac:
        addi   $t0, $zero, 42

ShowKey:
        sw     $t0, 0($s0)          # show key
        nop
        beq    $t3, 4, Exit
      j loop
Co:
        addi   $t3, $t3, 1
```

```
        j       Encrypt2
Exit:
        li      $v0, 10
        syscall
```

2. Kết quả



Assignment 3

```
.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 track
```

```
        jal     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
```