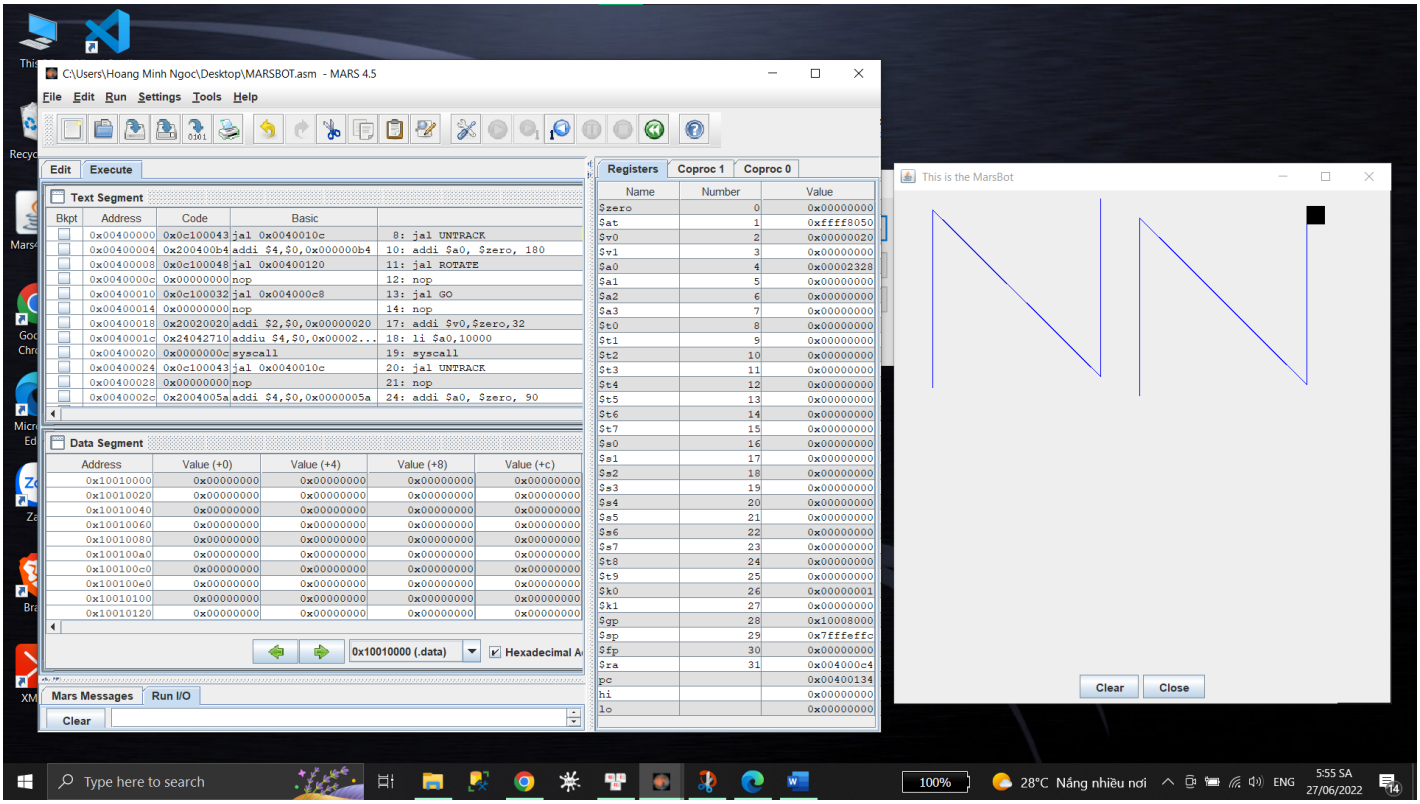
BÁO CÁO TUẦN 11

Bài 1:



Bài code trong MIPS

```
.eqv HEADING 0xffff8010
.eqv MOVING 0xffff8050
.eqv LEAVETRACK 0xffff8020
.eqv WHEREX 0xffff8030
.eqv WHEREY 0xffff8040
.text
main:
jal UNTRACK
rotate_1:
addi $a0, $zero, 180
jal ROTATE
nop
```

```
jal GO
nop

sleep_1:
addi $v0,$zero,32
li $a0,10000
syscall
jal UNTRACK
nop

rotate_2:
addi $a0, $zero, 90
jal ROTATE
nop

sleep_2:
addi $v0,$zero,32
li $a0,2000
syscall
nop

# Start Drawing
jal TRACK

rotate_3:
addi $a0, $zero, 0
jal ROTATE
nop

line_1:
```

```
addi $v0,$zero,32
li $a0,9000
syscall
jal UNTRACK
nop

rotate_4:
addi $a0, $zero, 135
jal ROTATE
nop

line_2:
jal TRACK
addi $v0,$zero,32
li $a0,12000
syscall
jal UNTRACK
nop

rotate_5:
addi $a0, $zero, 45
jal ROTATE
nop


rotate_6:
addi $a0, $zero, 0
jal ROTATE
nop
```

```
line_3:
jal TRACK
addi $v0,$zero,32
li $a0,9000
syscall
jal UNTRACK
nop

jal STOP
j end

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:
li $at, MOVING              # change MOVING port to 0
sb $zero, 0($at)# to stop
nop
jr $ra
nop

TRACK:
li $at, LEAVETRACK          # change LEAVETRACK port
addi $k0, $zero,1           # to logic 1,
sb $k0, 0($at)              # to start tracking
```

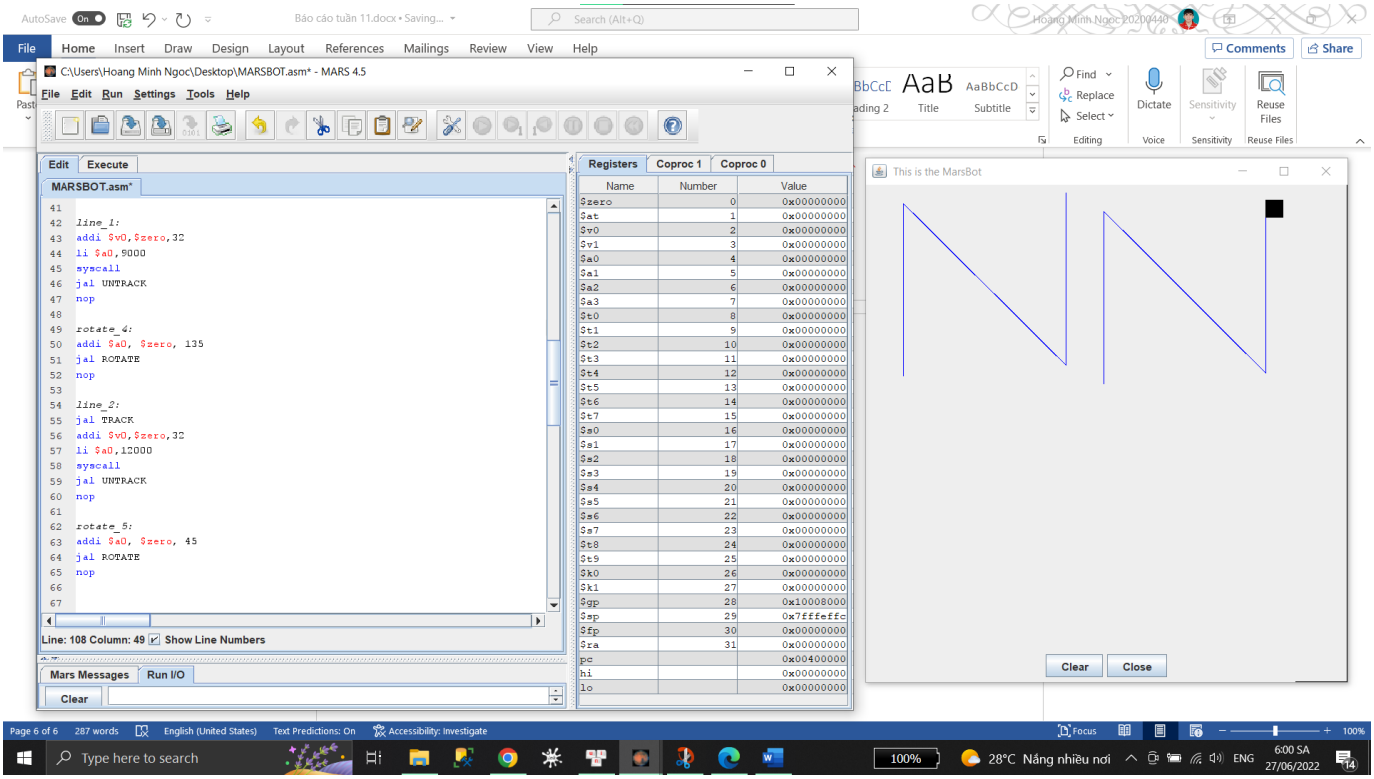
```
nop
jr $ra
nop

UNTRACK:
li $at, LEAVETRACK # change LEAVETRACK port to 0
sb $zero, 0($at)   # to stop drawing tail
nop
jr $ra
nop

ROTATE:
li $at, HEADING # change HEADING port
sw $a0, 0($at)  # to rotate robot
nop
jr $ra
nop
end:
```

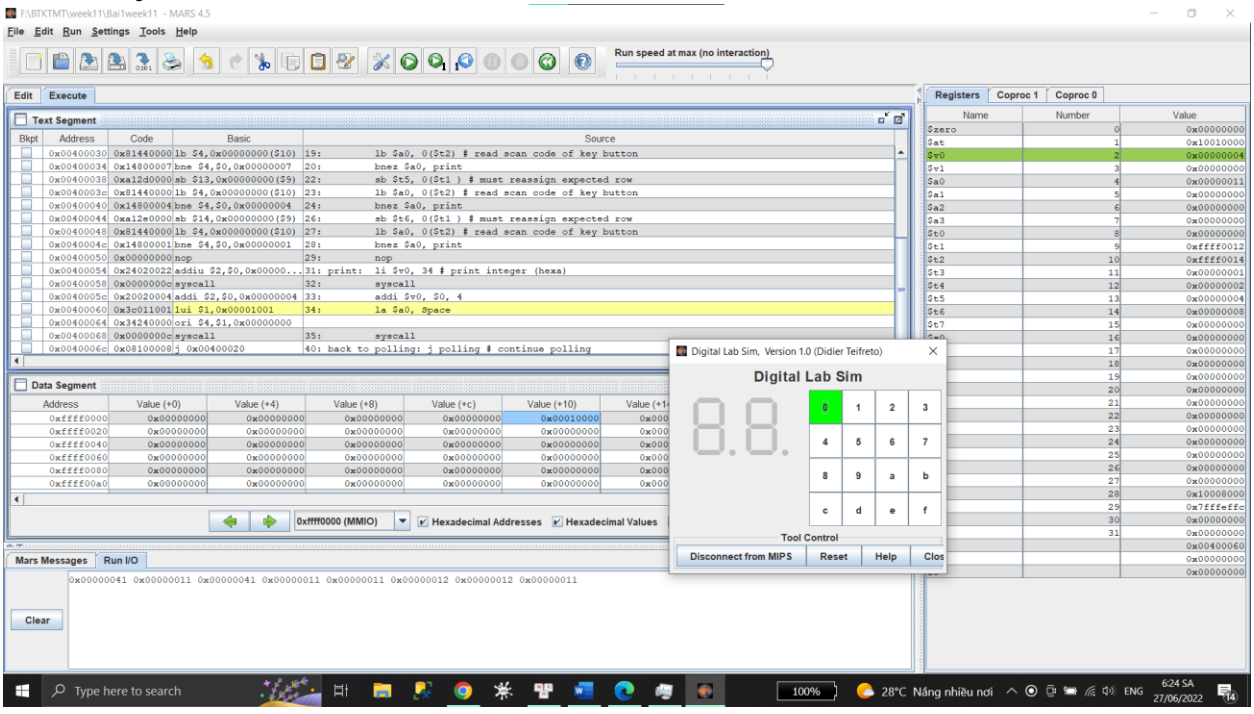
line là các đường để chạy, chữ N là kết hợp giữa 3 đường, 3 đường e đã tô ở trên, đường 1 và 3 xoay 0 độ và đường 2 xoay 135 độ

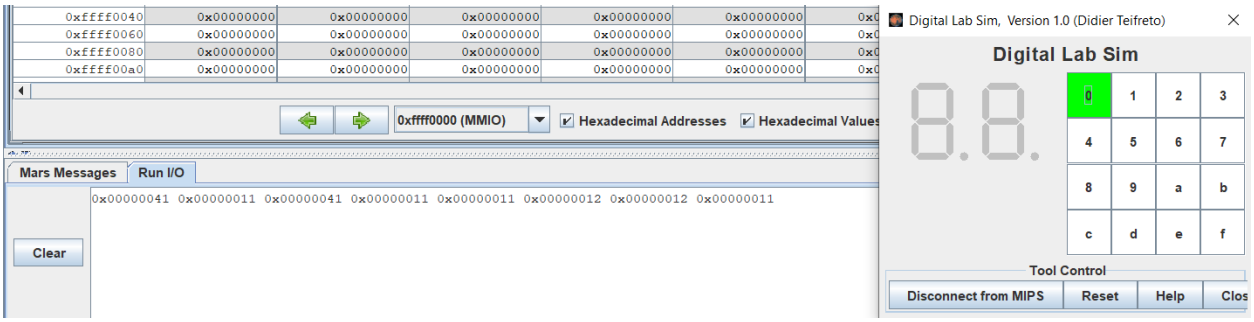
chạy 2 lần liên tục ạ



Bài 2: (Bài 1 week 11)

Run chạy mã số sinh viên 20200440





Code trong MIPS

```
.eqv IN_ADDRESS_HEXA_KEYBOARD 0xFFFF0012
.eqv OUT_ADDRESS_HEXA_KEYBOARD 0xFFFF0014
.data
Space: .asciiz " "
.text
main:
                li $t1, IN_ADDRESS_HEXA_KEYBOARD
                li $t2,
OUT_ADDRESS_HEXA_KEYBOARD
                li $t3, 0x1 # check row 1 with key 0,1,2,3
                li $t4, 0x2 # check row 2 with key 4,5,6,7
                li $t5, 0x4 # check row 3 with key 8,9,a,b
                li $t6, 0x8 # check row 4 with key c,d,e,f
polling:
                sb $t3, 0($t1 ) # must reassign expected row
                lb $a0, 0($t2) # read scan code of key button
                bnez $a0, print

                sb $t4, 0($t1 ) # must reassign expected row
                lb $a0, 0($t2) # read scan code of key button
```

```
                    bnez $a0, print

                    sb $t5, 0($t1 ) # must reassign expected row
                    lb $a0, 0($t2) # read scan code of key button
                    bnez $a0, print

                    sb $t6, 0($t1 ) # must reassign expected row
                    lb $a0, 0($t2) # read scan code of key button
                    bnez $a0, print
                    nop

print:              li $v0, 34 # print integer (hexa)
                    syscall
                    addi $v0, $0, 4
                    la $a0, Space
                    syscall

#sleep:             li $a0, 2500 # sleep 100ms
#                   li $v0, 32
#                   syscall
back_to_polling: j polling # continue polling
```

em có khóa sleep lại vì chạy theo thời gian e bị lag máy

```
7           li $t1, IN_ADDRESS_HEXA_KEYBOARD
8           li $t2, OUT_ADDRESS_HEXA_KEYBOARD
9           li $t3, 0x1 # check row 1 with key 0,1,2,3
10          li $t4, 0x2 # check row 2 with key 4,5,6,7
11          li $t5, 0x4 # check row 3 with key 8,9,a,b
12          li $t6, 0x8 # check row 4 with key c,d,e,f
```

Ta khai báo cả 4 dòng cần check

```
13  polling:
14          sb $t3, 0($t1 ) # must reassign expected row
15          lb $a0, 0($t2) # read scan code of key button
16          bnez $a0, print
17
18          sb $t4, 0($t1 ) # must reassign expected row
19          lb $a0, 0($t2) # read scan code of key button
20          bnez $a0, print
21
22          sb $t5, 0($t1 ) # must reassign expected row
23          lb $a0, 0($t2) # read scan code of key button
24          bnez $a0, print
25
26          sb $t6, 0($t1 ) # must reassign expected row
27          lb $a0, 0($t2) # read scan code of key button
28          bnez $a0, print
29          nop
30
```

E chia thành 4 khổ, từng cái check xem là có ở dòng đó hay không, nếu có nhảy tới print, do có 1 đầu vào và 1 trong 4 nên chắc chắn không có xung đột

```
31  print:  li $v0, 34 # print integer (hexa)
32          syscall
33          addi $v0, $0, 4
34          la $a0, Space
35          syscall
```

E chèn thêm khoảng trống giữa các số để cho dễ nhìn

```
36
37  #sleep:         li $a0, 2500 # sleep 100ms
38  #       li $v0, 32
39  #       syscall
40  back_to_polling: j polling # continue polling
```

quay lại chương trình lặp vô hạn

dòng 37-29 em cho thời gian để bấm nhưng nó hơi lag máy nên e đã khóa lại bấm chay :V