Họ tên: Kiều Đăng Nam

MSSV: 20176830

**Báo cáo thực hành tuần 10**

**Asignment 1**

Hiển thị 2 số cuối MSSV.

1. Source code:

#Laboratory Exercise 10, Assignment 1

.eqv SEVENSEG\_LEFT 0xFFFF0010 # Dia chi cua den led 7 doan trai.

# Bit 0 = doan a;

# Bit 1 = doan b; ...

# Bit 7 = dau .

.eqv SEVENSEG\_RIGHT 0xFFFF0011 # Dia chi cua den led 7 doan phai

.text

main:

li $a0, 0x3F # set value for segments

jal SHOW\_7SEG\_LEFT # show

li $a0, 0x4F # set value for segments

jal SHOW\_7SEG\_RIGHT # show

exit:

li $v0, 10

syscall

endmain:

#---------------------------------------------------------------

# Function SHOW\_7SEG\_LEFT : turn on/off the 7seg

# param[in] $a0 value to shown

# remark $t0 changed

#---------------------------------------------------------------

SHOW\_7SEG\_LEFT:

li $t0, SEVENSEG\_LEFT # assign port's address

sb $a0, 0($t0) # assign new value

jr $ra

#---------------------------------------------------------------

# Function SHOW\_7SEG\_RIGHT : turn on/off the 7seg

# param[in] $a0 value to shown

# remark $t0 changed

#---------------------------------------------------------------

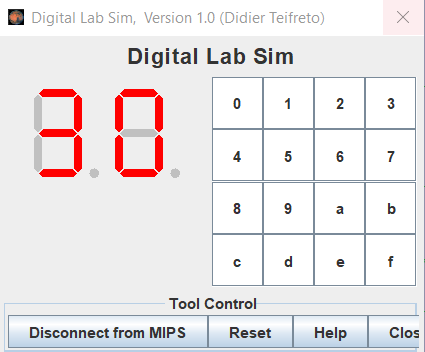
SHOW\_7SEG\_RIGHT:

li $t0, SEVENSEG\_RIGHT # assign port's address

sb $a0, 0($t0) # assign new value

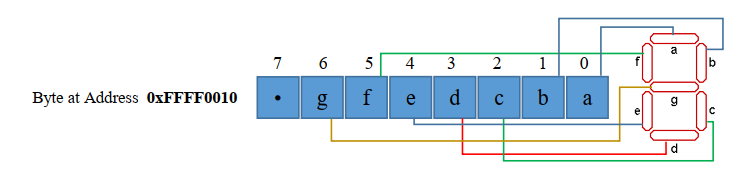
jr $ra

1. Kết quả chạy chương trình



Giải thích: Nhiệm vụ là hiển thị số 3 và số 0:

Từ lý thuyết:



* Hiển thi số 3:

Các thanh a, b, c, d, g sáng 🡪 01001111 = 4F

li $a0, 0x4F # set value for segments

jal SHOW\_7SEG\_RIGHT # show

* Hiển thi số 0:

Các thanh a, b, c, d, e, f sáng 🡪 00111111 = 3F

li $a0, 0x3F # set value for segments

jal SHOW\_7SEG\_LEFT # show

**Asignment 2:**

1. Source code

#Laboratory Exercise 10, Assignment 2

.eqv MONITOR\_SCREEN 0x10010000

.eqv RED 0x00FF0000

.text

li $k0, MONITOR\_SCREEN

addi $t0, $zero, 1584 #$t0 = i = 1584 = (12\*32 + 12)\*4

#start point of start row)

addi $t5, $zero, 2608 #$t5 = 2608 = (20\*32 + 12)\*4

#end point of end row

draw:

addi $t4, $t0, 32 #t4 = i + 32 : start point of row

add $t1, $zero, $t0 #t1 = j = i

draw\_row:

add $t2, $k0, $t1 #address of k[j]

li $t3, RED #draw for address j

sw $t3, ($t2)

add $t1, $t1, 4 #j = j+ 4

beq $t1, $t4, end\_of\_draw\_row

j draw\_row

end\_of\_draw\_row:

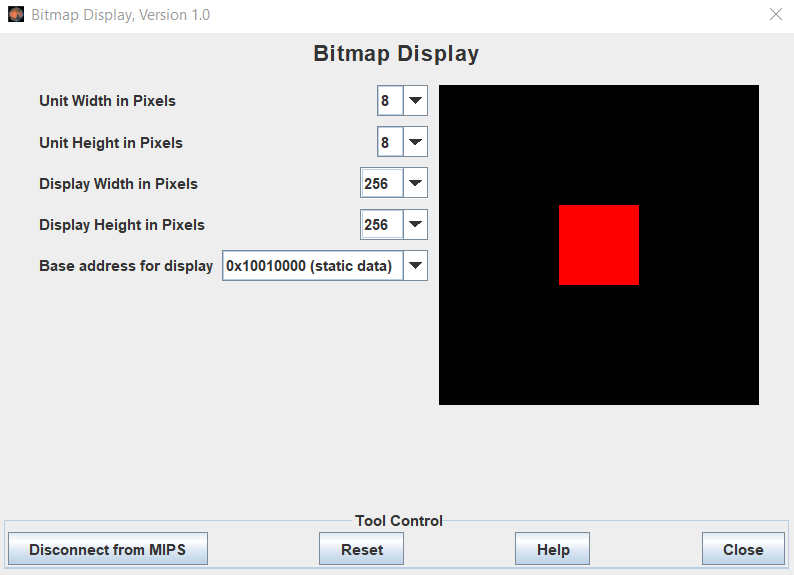
addi $t0, $t0, 128 #i = start point of next row

beq $t0, $t5, end\_of\_draw

j draw

end\_of\_draw:

1. Kết quả :



**Assignment 3**

1. Source 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 of MarsBot

.eqv WHEREY 0xffff8040 # Integer: Current y-location of MarsBot

.text

main:

jal TRACK # draw track line

addi $a0, $zero, 90 # Marsbot rotates 90\* and start

start:

jal ROTATE

jal GO

sleep1:

addi $v0, $zero, 32 # Keep running by sleeping in 3000 ms

li $a0, 10000

syscall

jal UNTRACK # keep old track

jal TRACK # and draw new track line

goDown:

addi $a0, $zero, 210 # Marsbot rotates 210\*

jal ROTATE

sleep2:

addi $v0, $zero, 32 # Keep running by sleeping in 3000 ms

li $a0, 10000

syscall

jal UNTRACK # keep old track

jal TRACK # and draw new track line

goUp:

addi $a0, $zero, 330 # Marsbot rotates 330\*

jal ROTATE

sleep3:

addi $v0, $zero, 32 # Keep running by sleeping in 3000 ms

li $a0, 10000

syscall

jal UNTRACK # keep old track

jal TRACK # and draw new track line

stop:

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)

#-----------------------------------------------------------

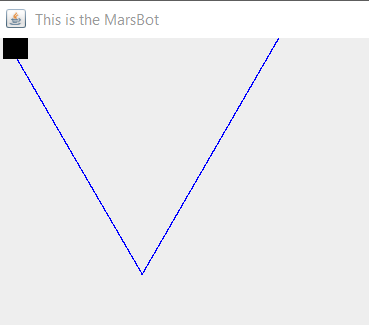
ROTATE:

li $at, HEADING # change HEADING port

sw $a0, 0($at) # to rotate robot

jr $ra

1. Kết quả:



**Asignment 4**

1. Source 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

.eqv e\_Char 0x65

.eqv x\_Char 0x78

.eqv i\_Char 0x69

.eqv t\_Char 0x74

.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

j checkE

WaitForDis:

lw $t2, 0($s1) # $t2 = [$s1] = DISPLAY\_READY

beq $t2, $zero, WaitForDis # if $t2 == 0 then Polling

ShowKey:

sw $t0, 0($s0) # show key

nop

j loop

checkE:

beq $t3, e\_Char, checkX # check if exist e in queue, checkX

bne $t0, e\_Char, WaitForDis # if current char is not e, continue

add $t3, $t0, $zero # save 'e' to $t3

j WaitForDis

checkX:

beq $t4, x\_Char, checkI # check if exist x in queue, checkI

bne $t0, x\_Char, reset # if current char is not x, reset then continue

add $t4, $t0, $zero # save 'x' to $t4

j WaitForDis

checkI:

beq $t5, i\_Char, checkT # check if exist i in queue, checkT

bne $t0, i\_Char, reset # if current char is not i, reset then continue

add $t5, $t0, $zero # save 'i' to $t5

j WaitForDis

checkT:

beq $t0, t\_Char, terminate # check if meet t, terminate (exit word complete)

j reset # if current char is not t, reset then continue

reset: li $t3, 0 # set 'e' to unspecified

li $t4, 0 # set 'x' to unspecified

li $t5, 0 # set 'i' to unspecified

j WaitForDis

terminate: li $v0, 10

syscall

1. Kết quả

