Báo cáo Tuần 10 (phần 2)

Phùng Ngọc Vinh – 20194719

Bài 1:

Điều khiển marsbot di chuyển theo:

1. Hình tam giác đều

.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 trackline

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

jal ROTATE

jal GO

sleep0: addi $v0,$zero,32 # Keep running by sleeping in1000 ms

li $a0,8000

syscall

jal UNTRACK # keep old track

#jal TRACK # and draw new track line

#jal TRACK # draw trackline

addi $a0, $zero, 180 # Marsbot rotates 180\* and start running

jal ROTATE

jal GO

sleep1: addi $v0,$zero,32 # Keep running by sleeping in3000 ms

li $a0,3000

syscall

jal UNTRACK # keep old track

jal TRACK # and draw new track line

goASKEWRIGHT: addi $a0, $zero, 150 # Marsbot rotates 150\*

jal ROTATE

sleep2: addi $v0,$zero,32 # Keep running by sleeping in 5000 ms

li $a0,5000

syscall

jal UNTRACK # keep old track

jal TRACK # and draw new track line

goLEFT: addi $a0, $zero, 270 # Marsbotrotates 270\*

jal ROTATE

sleep3: addi $v0,$zero,32 # Keep running by sleeping in 5000 ms

li $a0,5000

syscall

jal UNTRACK # keep old track

jal TRACK # anddraw new track line

goASKEWLEFT:addi $a0, $zero, 30 # Marsbot rotates 30\*

jal ROTATE

sleep4: addi $v0,$zero,32 # Keep running by sleeping in 5000 ms

li $a0,5000

syscall

jal UNTRACK # keep old track

#jal TRACK # and draw new track line

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)

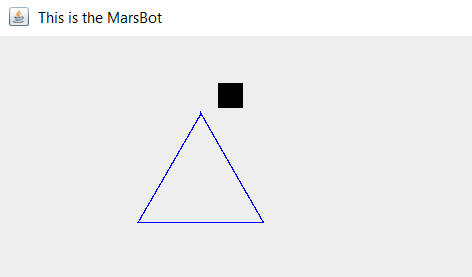
#-----------------------------------------------------------

ROTATE: li $at, HEADING # change HEADING port

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

jr $ra

Kết quả:



2.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:

addi $a0, $zero, 135 # Marsbot rotates 90\* and start running

jal ROTATE

jal GO

sleep1: addi $v0,$zero,32 # Keep running by sleeping in1000 ms

li $a0,2000

syscall

jal TRACK# and draw new track line

goRIGHT: addi $a0, $zero, 90# 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

goDOWN: addi $a0, $zero, 180# Marsbotrotates 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 # anddraw new track line

goLEFT: addi $a0, $zero, 270# Marsbotrotates 270\*

jal ROTATE

sleep4: addi $v0,$zero,32 # Keep running by sleeping in 1000 ms

li $a0,5000

syscall

jal UNTRACK # keep old track

jal TRACK # anddraw new track line

goUP: addi $a0, $zero, 0# Marsbotrotates 270\*

jal ROTATE

sleep5: addi $v0,$zero,32 # Keep running by sleeping in 1000 ms

li $a0,5000

syscall

jal UNTRACK # keep old track

end\_main:

li $v0, 10#exit

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)

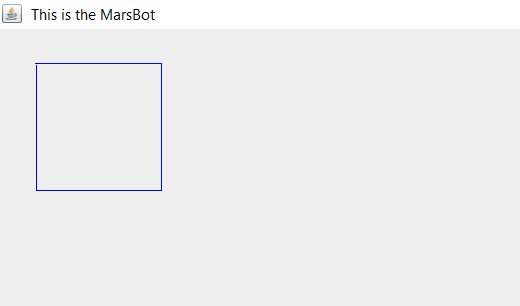
#-----------------------------------------------------------

ROTATE: li $at, HEADING # change HEADING port

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

jr $ra

Kết quả:



3.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:

addi $a0, $zero, 135 # Marsbot rotates 135\* and start running

jal ROTATE

jal GO

sleep1: addi $v0,$zero,32 # Keep running by sleeping in4000 ms

li $a0,4000

syscall

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 4000 ms

li $a0,4000

syscall

jal UNTRACK # keep old track

jal TRACK # and draw new track line

go2: addi $a0, $zero, 234# Marsbotrotates 234\*

jal ROTATE

sleep3: addi $v0,$zero,32 # Keep running by sleeping in 4000 ms

li $a0,4000

syscall

jal UNTRACK # keep old track

jal TRACK # anddraw new track line

go3: addi $a0, $zero, 18 # Marsbotrotates 18\*

jal ROTATE

sleep4: addi $v0,$zero,32 # Keep running by sleeping in 4000 ms

li $a0,4000

syscall

jal UNTRACK # keep old track

jal TRACK # anddraw new track line

go4: addi $a0, $zero, 162# Marsbotrotates 162\*

jal ROTATE

sleep5: addi $v0,$zero,32 # Keep running by sleeping in 4000 ms

li $a0,4000

syscall

jal UNTRACK # keep old track

jal TRACK # anddraw new track line

go5: addi $a0, $zero, 306# Marsbotrotates 306\*

jal ROTATE

sleep6: addi $v0,$zero,32 # Keep running by sleeping in 4000 ms

li $a0,4000

syscall

jal UNTRACK # keep old track

end\_main:

li $v0, 10#exit

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)

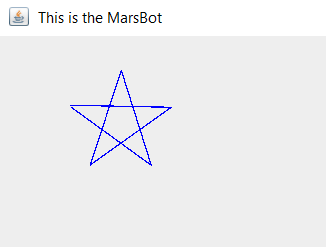
#-----------------------------------------------------------

ROTATE: li $at, HEADING # change HEADING port

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

jr $ra

Kết quả:



Bài 2:

.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:

toUpper:

li $t1, 'a'

blt $t0,$t1,toLower

bge $t0,123, defautl

addi $t0,$t0,-32

j ShowKey

toLower:

li $t4,91

bge $t0,$t4,defautl

li $t2,'A'

blt $t0,$t2, number

addi $t0,$t0,32

j ShowKey

number:

li $t3,58

bge $t0,$t3,defautl

li $t2,'0'

blt $t0,$t2,defautl

j ShowKey

defautl:

li $t0,'\*'

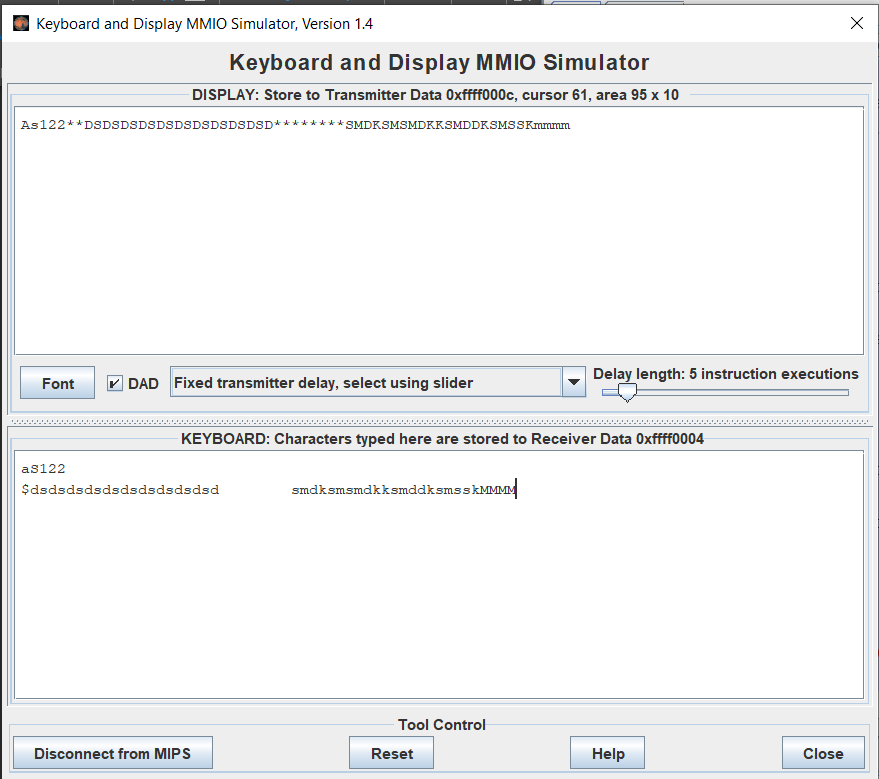
j ShowKey

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

nop

j loop

Kết quả:



Bài 3:

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 ofMarsBot

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

main:

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

jal ROTATE

jal TRACK

jal GO

li $s1,'w' #w

li $s2,'a' #a

li $s3,'d' #d

li $s4,'s' #s

li $s5, 32

toStart: nop

li $v0,12

syscall

add $a1,$zero,$v0

beq $a1,$s5,pause

lw $t8, MOVING

beq $t8,$zero,toStart

beq $a1,$s1, toUp

beq $a1,$s2,toLeft

beq $a1,$s3,toRight

beq $a1,$s4,toDown

jal TRACK # draw track line

toUp:

addi $a0, $zero, 0 # Marsbot rotates 0\* and startrunning

jal ROTATE

jal UNTRACK # keep old track

jal TRACK # and draw new track line

j toStart

toDown:

addi $a0, $zero, 180 # Marsbot rotates 180\* and startrunning

jal ROTATE

jal UNTRACK # keep old track

jal TRACK # and draw new track line

j toStart

toRight:

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

jal ROTATE

jal UNTRACK # keep old track

jal TRACK # and draw new track line

j toStart

toLeft:

addi $a0, $zero, 270 # Marsbot rotates 270\* and startrunning

jal ROTATE

jal UNTRACK # keep old track

jal TRACK # and draw new track line

j toStart

pause:

lw $t1, MOVING

beq $t1,$zero,GO

li $t2,1

beq $t1,$t2,STOP

j toStart

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)

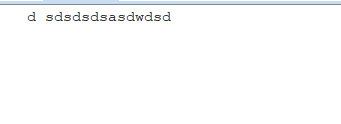
#-----------------------------------------------------------

ROTATE: li $at, HEADING # change HEADING port

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

jr $ra

\*Kết quả:

Nhập dãy: 

Hiển thị: 