**BÁO CÁO TUẦN 10(2)**

**Bài 1:**

a, Tam giác đều:

* *Chương trình:*

.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 in1000 ms

li $a0,10000

syscall

jal TRACK # And draw new track line

edge1: addi $a0, $zero, 150 # Marsbot rotates 150\*

jal ROTATE

sleep2: 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

edge2: 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

edge3: 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 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:*

Graphical user interface, application

Description automatically generated

b, Hình vuông:

* *Chương trình:*

.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 135\* and start running

jal ROTATE

jal GO

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

li $a0,10000

syscall

jal TRACK # And draw new track line

edge1: addi $a0, $zero, 90 # Marsbot rotates 90\*

jal ROTATE

sleep2: 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

edge2: addi $a0, $zero, 180 # Marsbotrotates 180\*

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

edge3: addi $a0, $zero, 270 # Marsbot rotates 270\*

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

edge4: addi $a0, $zero, 0 # Marsbot rotates 0\*

jal ROTATE

sleep5: 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:*

Graphical user interface, application

Description automatically generated

C, Hình sao

* *Chương trình:*

.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 135\* and start running

jal ROTATE

jal GO

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

li $a0,10000

syscall

jal TRACK # And draw new track line

edge1: addi $a0, $zero, 162 # Marsbot rotates 162\*

jal ROTATE

sleep2: 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

edge2: addi $a0, $zero, 306 # Marsbotrotates 306\*

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

edge3: addi $a0, $zero, 90 # Marsbot rotates 90\*

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

edge4: addi $a0, $zero, 234 # Marsbot rotates 234\*

jal ROTATE

sleep5: 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

edge5: addi $a0, $zero, 18 # Marsbot rotates 18\*

jal ROTATE

sleep6: 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:

Graphical user interface, application

Description automatically generated

**Bài 2:**

* Chương trình:

Text

Description automatically generated

* Kết quả:

A screenshot of a computer

Description automatically generated with medium confidence

**Bài 3:**

* Chương trình:

.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

j continue

UP: bne $t0, 'w', DOWN

addi $a0, $zero, 0

j rotate

DOWN: bne $t0, 's', LEFT

addi $a0, $zero, 180

j rotate

LEFT: bne $t0, 'a', RIGHT

addi $a0, $zero, 270

j rotate

RIGHT: bne $t0, 'd', continue

addi $a0, $zero, 90

j rotate

rotate:

jal ROTATE

jal TRACK # and draw new track line

j continue

continue: j loop

end\_main:

addi $v0, $0, 10

syscall

GO: li $at, MOVING # change MOVING port

addi $a0, $zero,1 # to logic 1,

sb $a0, 0($at) # to start running

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 $a0, $zero,1 # to logic 1,

sb $a0, 0($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

ROTATE: li $at, HEADING # change HEADING port

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

jr $ra

* Kết quả:

Graphical user interface, application

Description automatically generated