Vu Duc An – 20215174

Lab Report

1. Assignment 1:

A screenshot of a computer

Description automatically generated

1. Assignment 2:

A screenshot of a computer

Description automatically generated

# Define constants for monitor screen and colors

.eqv MONITOR\_SCREEN 0x10010000

.eqv RED 0x00FF0000

.eqv GREEN 0x0000FF00

.eqv BLUE 0x000000FF

.eqv WHITE 0x00FFFFFF

.eqv BLACK 0x00000000

.text

li $k0, MONITOR\_SCREEN

sub $k0, $k0, 4 # starting address

addi $k1, $k0, 256 # ending address

# Initialize loop counters

addi $t1, $0, 0 # bits in row

addi $t2, $0, 32

addi $t3, $0, 0 # column num

loop:

beq $k1, $k0, end

row\_loop:

# Check if $t1 is equal to 32

beq $t1, $t2, end\_row\_loop

addi $t1, $t1, 4

li $t0, BLACK

addi $k0, $k0, 4

sw $t0, ($k0)

beq $t3, 4, check\_row\_1

beq $t3, 8, check\_row\_2

beq $t3, 12, check\_row\_3

beq $t3, 16, check\_row\_4

beq $t3, 20, check\_row\_5

beq $t3, 24, check\_row\_6

# Repeat Loop1

j row\_loop

# End of Loop1

end\_row\_loop:

# Reset loop counters

addi $t1, $0, 0

addi $t2, $0, 32

addi $t3, $t3, 4

check\_row\_1:

beq $t1, 16, place

beq $t1, 20, place

j loop

check\_row\_2:

beq $t1, 12, place

beq $t1, 24, place

j loop

check\_row\_3:

beq $t1, 8, place

beq $t1, 28, place

j loop

check\_row\_4:

beq $t1, 8, place

beq $t1, 12, place

beq $t1, 16, place

beq $t1, 20, place

beq $t1, 24, place

beq $t1, 28, place

j loop

check\_row\_5:

beq $t1, 8, place

beq $t1, 28, place

j loop

check\_row\_6:

beq $t1, 8, place

beq $t1, 28, place

j loop

j end

place:

li $t0, WHITE

sw $t0, ($k0)

j loop

end:

1. Assignment 3:

A screen shot of a computer

Description automatically generated

.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

.text

main:

prepare:

li $a0, 180

jal ROTATE

jal TRACK

jal GO

li $v0, 32

li $a0, 6500

syscall

jal UNTRACK

li $a0, 90

jal ROTATE

jal TRACK

jal GO

li $v0, 32

li $a0, 1000

syscall

jal UNTRACK

first\_edge:

li $a0, 30

jal ROTATE

jal TRACK

jal GO

li $v0, 32

li $a0, 6000

syscall

jal UNTRACK

second\_edge:

li $a0, 150

jal ROTATE

jal TRACK

jal GO

li $v0, 32

li $a0, 6000

syscall

jal UNTRACK

third\_edge:

li $a0, 330

jal ROTATE

jal TRACK

jal GO

li $v0, 32

li $a0, 3000

syscall

jal UNTRACK

fourth\_edge:

li $a0, 270

jal ROTATE

jal TRACK

jal GO

li $v0, 32

li $a0, 3000

syscall

jal UNTRACK

done:

li $a0, 90

jal ROTATE

jal TRACK

jal GO

li $v0, 32

li $a0, 3000

syscall

jal UNTRACK

quit: # quit program

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