Báo cáo Project cuối kì Thực hành Kiến trúc máy tính

GVHD: Nguyễn Thành Trung

Nhóm 7: - Nguyễn Quốc Huy

- Nguyễn Hà Phú Thịnh

I, Đề bài: Postscript CNC Marsbot

Máy gia công cơ khí chính xác CNC Marsbot được dùng để cắt tấm kim loại theo các đường nét được qui định trước. CNC Marsbot có một lưỡi cắt dịch chuyển trên tấm kim loại, với giả định rằng:

- Nếu lưỡi cắt dịch chuyển nhưng không cắt tấm kim loại, tức là Marsbot di chuyển nhưng không để lại vết (Track)

- Nếu lưỡi cắt dịch chuyển và cắt tấm kim loại, tức là Marsbot di chuyển và có để lại vết.

Để điều khiển Marsbot cắt đúng như hình dạng mong muốn, người ta nạp vào Marsbot một mảng cấu trúc gồm 3 phần tử:

- <Góc chuyển động>, <Thời gian>, <Cắt/Không cắt>

- Trong đó <Góc chuyển động> là góc của hàm HEADING của Marsbot

- <Thời gian> là thời gian duy trì quá trình vận hành hiện tại

- <Cắt/Không cắt> thiết lập lưu vết/không lưu vết

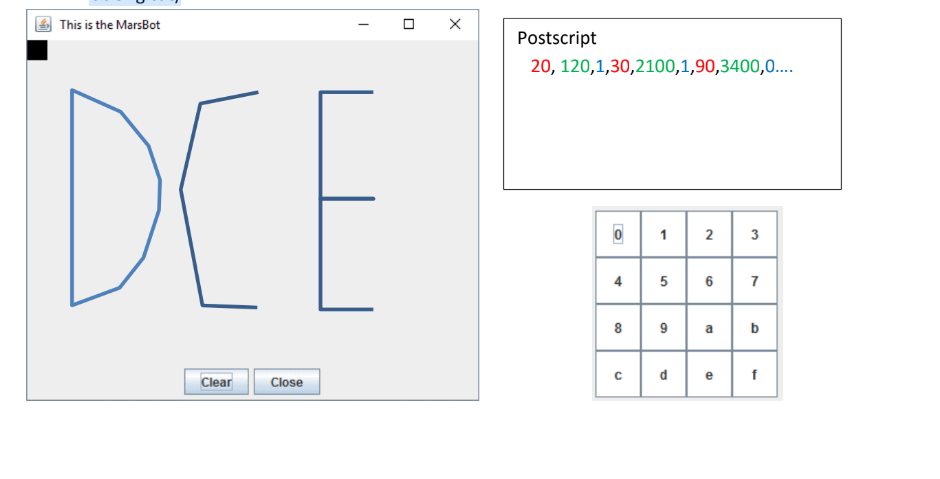
Hãy lập trình để CNC Marsbot có thể:

- Thực hiện cắt kim loại như đã mô tả

- Nội dung postscript được lưu trữ cố định bên trong mã nguồn

- Mã nguồn chứa 3 postscript và người dùng sử dụng 3 phím 0, 4, 8 trên bàn phím Key Matrix để chọn postscript nào sẽ được gia công.

- Một postscript chứa chữ DCE cần gia công. Hai script còn lại sinh viên tự đề xuất (tối thiểu 10 đường cắt)



II, Thực hiện bài toán:

1, Các công cụ và các hàm sử dụng trong chương trình:

a, Các công cụ:

- Digital Lab Sim: có trong phần mềm mô phỏng Mars.

- Mars Bot: có trong phần mềm mô phỏng Mars.

b, Các hàm:

- Polling: sử dụng để thăm dò bỏ phiếu xem postscript nào được chọn để gia công.

- Read\_rotate, Read\_time, Read\_track: lần lượt sử dụng để lưu các giá trị góc xoay của hàm Reading, thời gian di chuyển và trạng thái lưu vết của Mars Bot

- Go, Stop: lần lượt sử dụng để thay đổi trạng thái di chuyển hay dừng lại của Mars Bot

- Track, Untrack: lần lượt sử dụng để thay đổi trạng lưu vết hay không lưu vết của Mars Bot

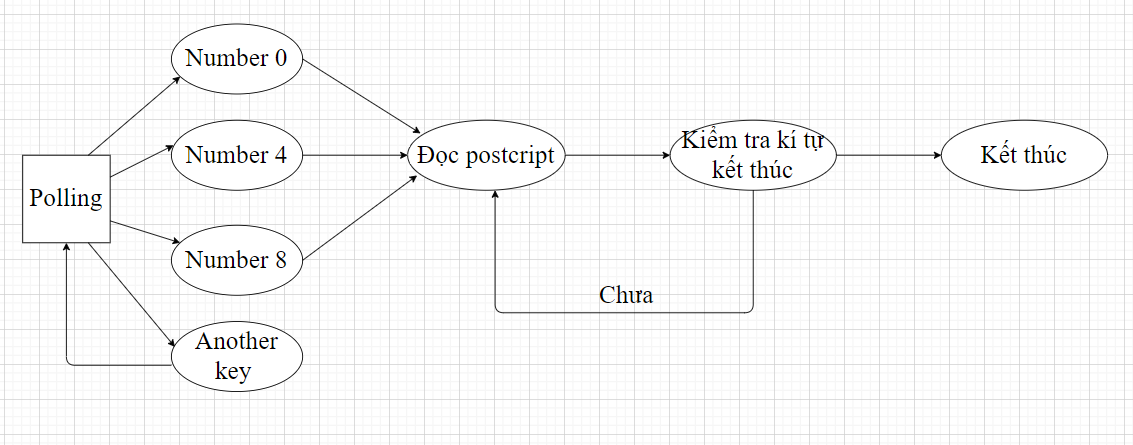
- Rotate: sử dụng để thay đổi góc của hàm Reading của Mars Bot.

2, Ý tưởng thực hiện:

- Đầu tiên sử dụng hàm polling để thăm dò ý kiến, xác định postsript nào sẽ được lựa chọn để gia công.

- Sau khi chọn được postscript để gia công, đọc mảng cấu trúc chứa các giá trị được lưu giữ trong mảng cấu trúc và thực hiện gia công.

- Sơ đồ thực hiện:



3, Lời giải sử dụng MIPS:

# Mars bot

.eqv HEADING 0xffff8010 # Interger: 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

# Key matrix

.eqv OUT\_ADRESS\_HEXA\_KEYBOARD 0xFFFF0014

.eqv IN\_ADRESS\_HEXA\_KEYBOARD 0xFFFF0012

.data

# (rotate, time, 0 = untrack | 1 = track)

# postscript0: DCE => numpad 0

pscript1: .asciiz "90,2000,0;180,3000,0;180,5790,1;80,500,1;70,500,1;60,500,1;50,500,1;40,500,1;30,500,1;20,500,1;10,500,1;0,500,1;350,500,1;340,500,1;330,500,1;320,500,1;310,500,1;300,500,1;290,500,1;280,490,1;90,8000,0;270,500,1;260,500,1;250,500,1;240,500,1;230,500,1;220,500,1;210,500,1;200,500,1;190,500,1;180,500,1;170,500,1;160,500,1;150,500,1;140,500,1;130,500,1;120,500,1;110,500,1;100,500,1;90,1000,1;90,5000,0;270,2000,1;0,5800,1;90,2000,1;180,2900,0;270,2000,1;90,3000,0;"

# postscript1 - HUY => numpad 4

pscript2: .asciiz "90,2000,0;180,3000,0;180,6000,1;0,3000,0;90,3000,1;0,3000,0;180,6000,1;90,1500,0;0,6000,0;180,4800,1;170,200,1;160,200,1;150,200,1;140,200,1;130,200,1;120,200,1;110,200,1;100,200,1;90,200,1;90,500,1;80,200,1;70,200,1;60,200,1;50,200,1;40,200,1;30,200,1;20,200,1;10,200,1;0,4800,1;90,1500,0;155,3900,1;25,3900,1;205,3900,0;180,2500,1;"

# postscript2 - THINH => numpad 8

pscript3: .asciiz "90,2000,0;180,3000,0;90,4000,1;270,2000,0;180,6000,1;90,3000,0;0,6000,1;180,3000,0;90,2500,1;0,3000,0;180,6000,1;90,1000,0;0,6000,1;90,1000,0;180,6000,1;0,6000,0;155,6500,1;0,6000,1;90,1000,0;180,6000,1;0,3000,0;90,2500,1;0,3000,0;180,6000,1;"

.text

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

# col 0x1 col 0x2 col 0x4 col 0x8

#

# row 0x1

# 0 1 2 3

# 0x11 0x21 0x41 0x81

#

# row 0x2

# 4 5 6 7

# 0x12 0x22 0x42 0x82

#

# row 0x4

# 8 9 a b

# 0x14 0x24 0x44 0x84

#

# row 0x8

# c d e f

# 0x18 0x28 0x48 0x88

#

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

# Keymatrix

# command row number of hexadecimal keyboard (bit 0 to 3)

# Eg. assign 0x1, to get key button 0,1,2,3

# assign 0x2, to get key button 4,5,6,7

# NOTE must reassign value for this address before reading,

# eventhough you only want to scan 1 row

li $t3, IN\_ADRESS\_HEXA\_KEYBOARD

# receive row and column of the key pressed, 0 if not key pressed

# Eg. equal 0x11, means that key button 0 pressed.

# Eg. equal 0x28, means that key button D pressed.

li $t4, OUT\_ADRESS\_HEXA\_KEYBOARD

polling:

li $t5, 0x1 # row-1 of key matrix

sb $t5, 0($t3) # must reassign expected row

lb $a0, 0($t4) # read scan code of key button

bne $a0, 0x11, NOT\_NUMPAD\_0 #if $a0 != 0x11 branch NOT\_NUMPAD\_0

la $a1, pscript1

j START

NOT\_NUMPAD\_0:

li $t5, 0x2 # row-2 of key matrix

sb $t5, 0($t3)

lb $a0, 0($t4)

bne $a0, 0x12, NOT\_NUMPAD\_4 #if $a0 != 0x11 branch NOT\_NUMPAD\_4

la $a1, pscript2

j START

NOT\_NUMPAD\_4:

li $t5, 0x4 # row-3 of key matrix

sb $t5, 0($t3)

lb $a0, 0($t4)

bne $a0, 0x14, COME\_BACK #if $a0 != 0x11 branch COME\_BACk

la $a1, pscript3

j START

COME\_BACK: j polling # khi cac so 0,4,8 khong duoc chon -> quay lai doc tiep

# <!--end-->

# MARS BOT

START:

jal GO

nop

READ\_PSCRIPT:

addi $t0, $zero, 0 # bien t0 dung de luu gia tri rotate

addi $t1, $zero, 0 # bien t1 dung de luu gia tri thoi gian

READ\_ROTATE:

add $t7, $a1, $t6 # Dich dia chi de doc gia tri tiep theo trong postscript

lb $t5, 0($t7) # luu tung ki tu vao $t5

beq $t5, 0, END # xet xen chuoi da ket thuc chua

beq $t5, 44, READ\_TIME # branch READ\_TIME neu $t5 = 44 "ma ACSCII cua ',' la 44"

mul $t0, $t0, 10 # nhan $t0 = $t0 \* 10;

addi $t5, $t5, -48 # ma ASCII cua so 0 la 48.

add $t0, $t0, $t5 # cong cac chu so lai voi nhau.

addi $t6, $t6, 1 # tang so ky tu can dich chuyen len 1

j READ\_ROTATE # quay lai doc tiep den khi gap dau ','

READ\_TIME: # doc thoi gian chuyen dong.

bne $a0, $t0, ACTIVITY # kiem tra $a0 chua gia tri rotate hay chua, chua thi re nhanh activity de thuc hien luu tru

j CONTINUE\_READTIME # neu da luu gia tri vao $a0 thi branch de tiep tuc doc thoi gian

ACTIVITY:

add $a0, $t0, $zero # $ao = $t0 + $0

jal ROTATE #branch ROTATE

nop

CONTINUE\_READTIME:

addi $t6, $t6, 1 # dich bit 1

add $t7, $a1, $t6 # ($a1 luu dia chi cua postscript)

lb $t5, 0($t7) # load value $t7 vao $t5

beq $t5, 44, READ\_TRACK # branch READ\_TRACK neu $t5 = 44 "ma ACSCII cua ',' la 44"

mul $t1, $t1, 10

addi $t5, $t5, -48

add $t1, $t1, $t5

j READ\_TIME # quay lai doc tiep den khi gap dau ','

READ\_TRACK:

addi $v0, $zero, 32 # Keep mars bot running by sleeping with time = $t1

add $a0, $zero, $t1

addi $t6, $t6, 1

add $t7, $a1, $t6

lb $t5, 0($t7)

addi $t5, $t5, -48

beq $t5, $zero, CHECK\_UNTRACK # 1 = track | 0 = untrack

jal UNTRACK

nop

jal TRACK

nop

j INCREMENT

CHECK\_UNTRACK:

jal UNTRACK

nop

INCREMENT:

syscall

addi $t6, $t6, 2 # bo qua dau ';'

j READ\_PSCRIPT

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

# 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

nop

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

# STOP procedure, to stop running

# param[in] none

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

STOP:

li $at, MOVING

sb $zero, 0($at)

jr $ra

nop

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

# TRACK procedure, to start drawing line

# param[in] none

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

TRACK:

li $at, LEAVETRACK

addi $k0, $zero,1

sb $k0, 0($at)

jr $ra

nop

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

# UNTRACK oricedure, to stop drawing line

# param[in] none

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

UNTRACK:

li $at, LEAVETRACK

sb $zero, 0($at)

jr $ra

nop

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

# 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

sw $a0, 0($at)

jr $ra

nop

END:

jal STOP

nop

li $v0, 10

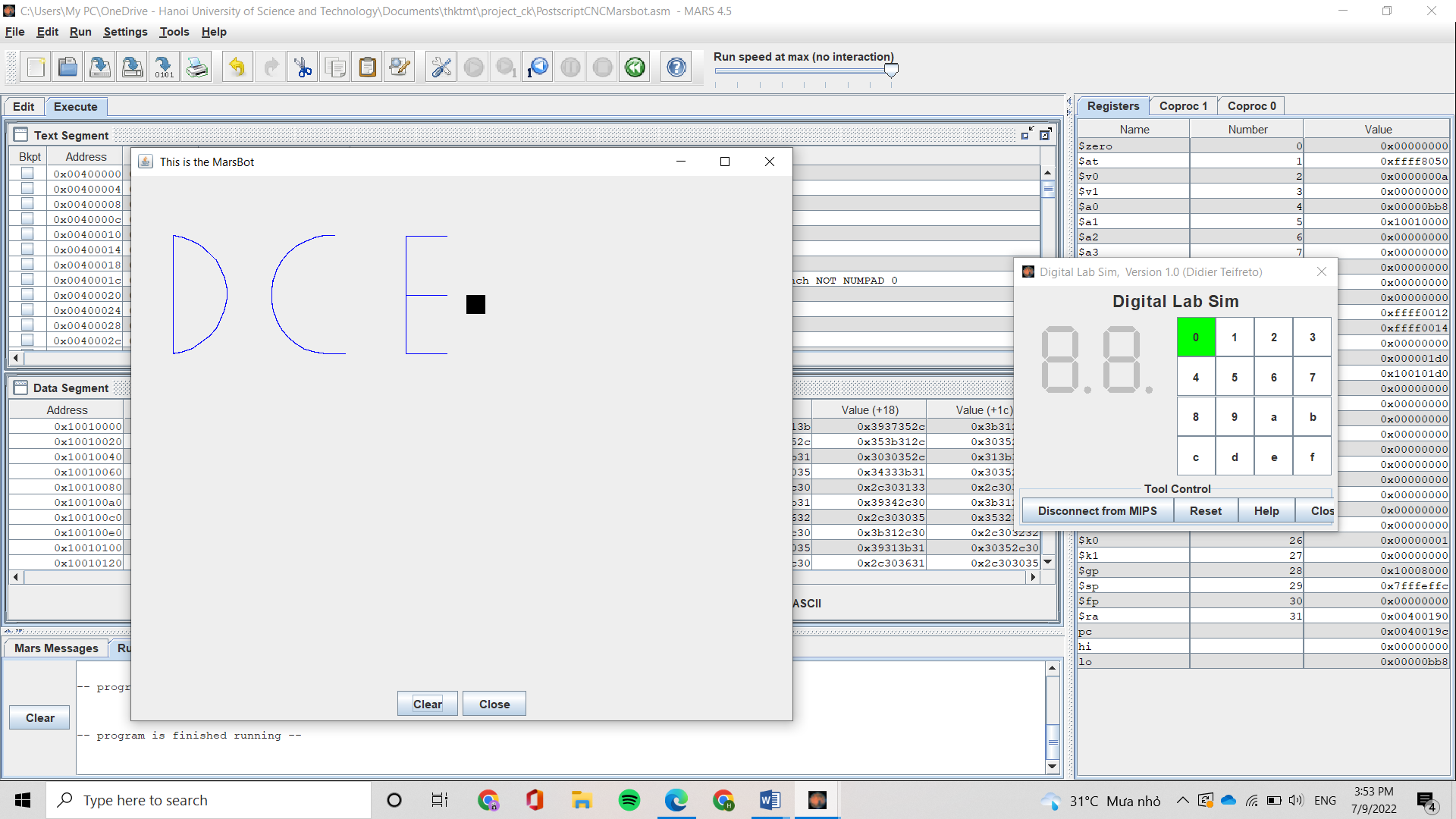
syscall

j polling

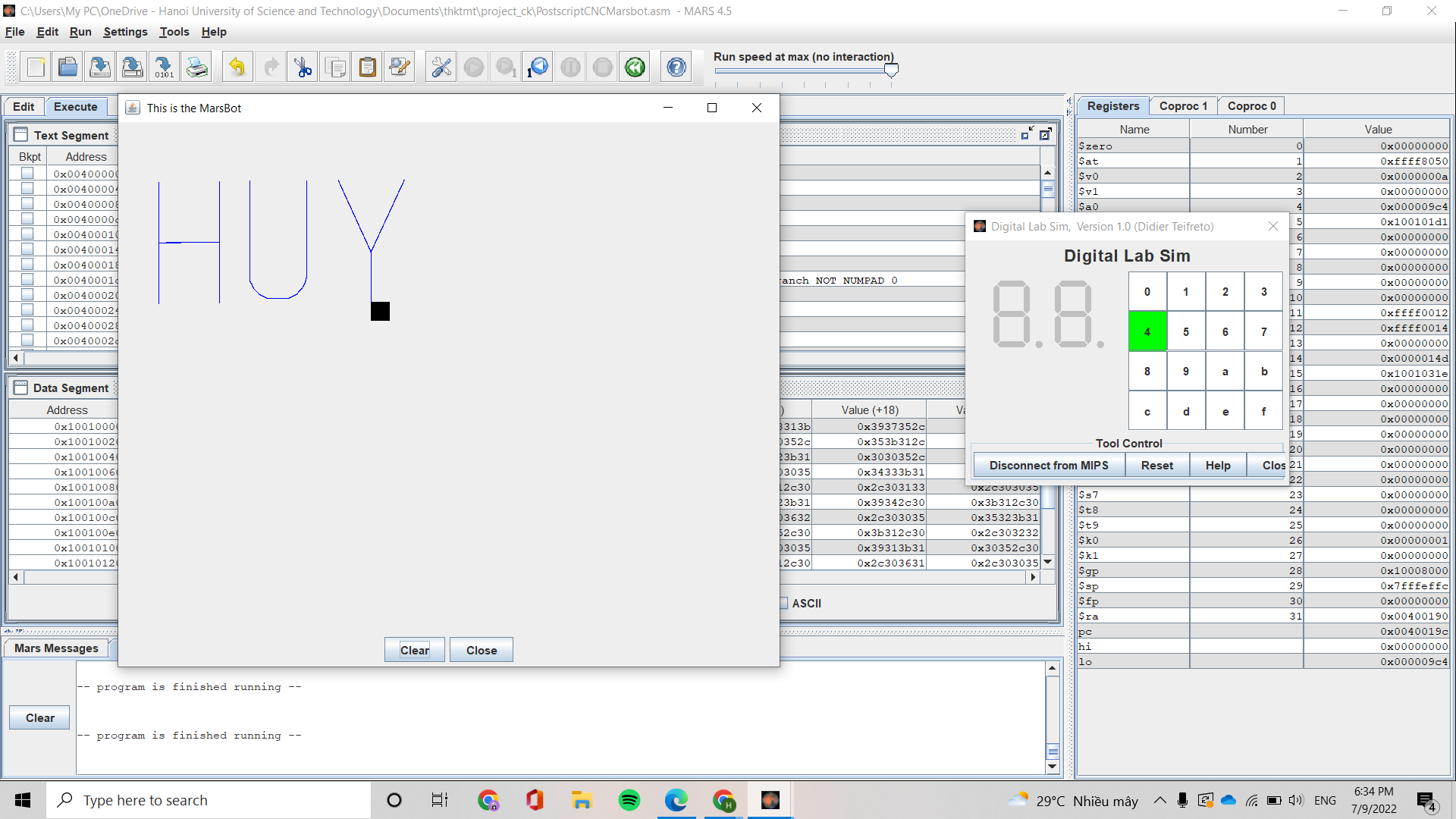
# <!--end-->

4, Một số kết quả khi chạy chương trình:

a, Key matrix = 0: Gia công chữ DCE:



b, Key matrix = 4: Gia công chữ HUY:



c, Key matrix = 8: Gia công chữ THINH:

