HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

------\*\*\*------

****

ASSEMBLY LANGUAGE AND COMPUTER ARCHITECTURE LAB

BÁO CÁO BÀI TẬP LỚN IT3280E -131102

Giảng viên hướng dẫn: Đỗ Công Thuần

Nhóm 04

| Name | Student ID | Task |
| --- | --- | --- |
| Trần Chí Thành | 20194845 | 3 |
| Đào Quang Dương | 20194747 | 4 |

Hanoi, 2022

# Table of contents

[Project 3: Kiểm tra tốc độ và độ chính xác khi gõ văn bản](#_x0snpghcwjs5) 3

[Problem](#_bczldcp7aj6e) 3

[Solution](#_dso97gva8kc5) 3

[Project 4: Postscript CNC Marsbot](#_5rdg6g9ty7vp) 9

[About my project](#_8rh3gc85icp7) 9

[Problems and solutions](#_rerb4i85ixc) 10

[Key Matrix](#_6sp1nqb9qtgh) 10

[Postscript](#_pnxd5su6hwm6) 10

[Marsbot](#_4h2w2tpr7jhb) 11

[Result](#_f5p8lny0qbf5) 13

[Differences](#_jcdoioxsnpc5) 14

# 

# Project 3: Kiểm tra tốc độ và độ chính xác khi gõ văn bản

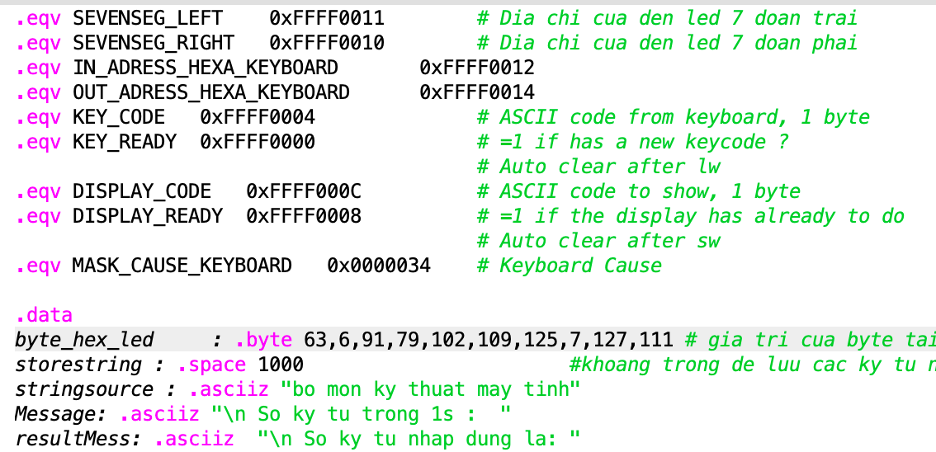
Người chịu trách nhiệm chính: Trần Chí Thành

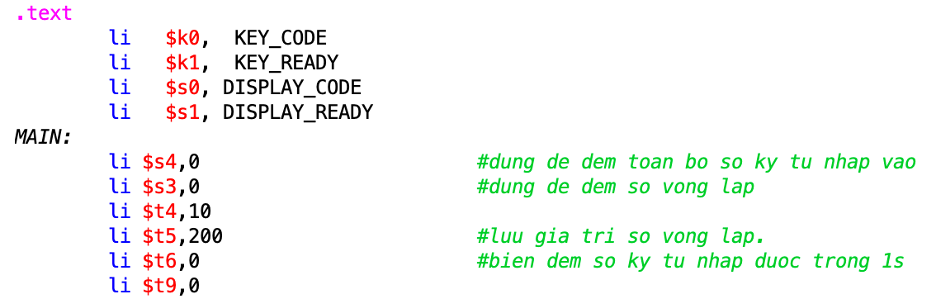
## 1. Problem

## **2. Solution**

We have splited the program into 5 parts:

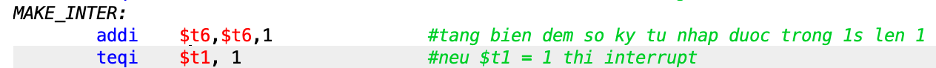
2.1. Declaration:

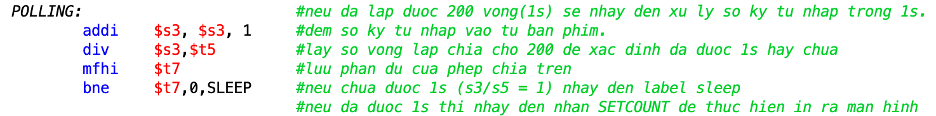
Declare load variable:

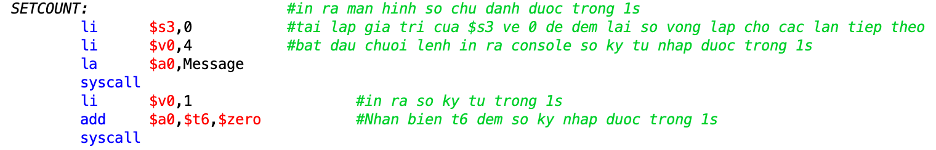
2.2. Counting the numbers of letter typed in 1s

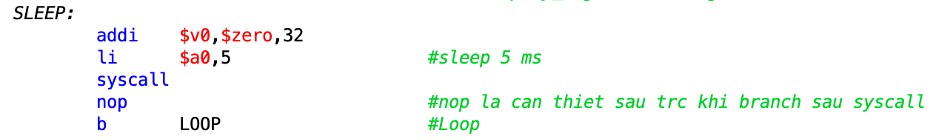
- Polling, check input control bit:

- If user type a letter then counter += 1 and call interrupt:

- If not, check whether the recursive reach 200 times(~1s):

- If it reach then print the numbers of letter typed in 1s:

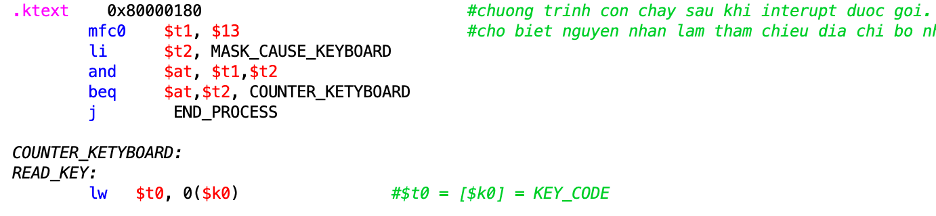
- If not, do sleep for 1 time(~5ms) then loop:

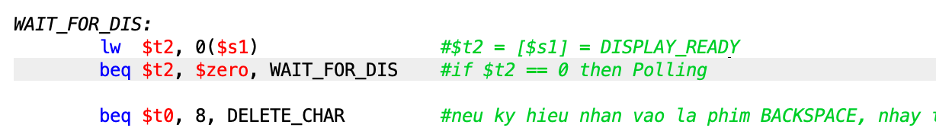
- Finally, reset counter and loop.

# 

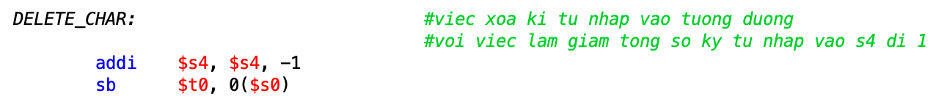
2.3. Soft interrupt

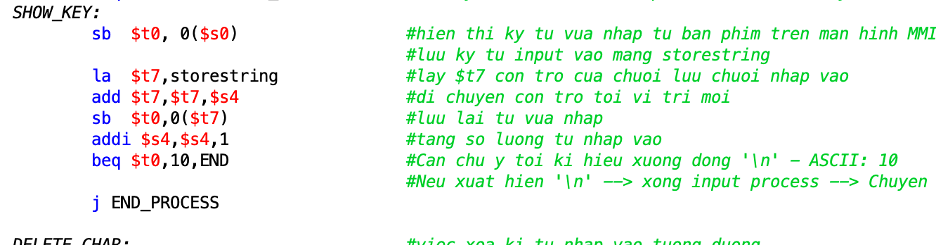
- Read data from “receiver data”:

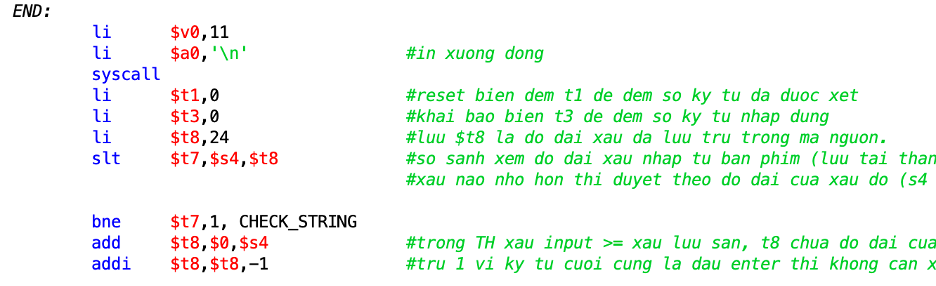
- If the value of transmitter control is 0(not lw), continue to loop:

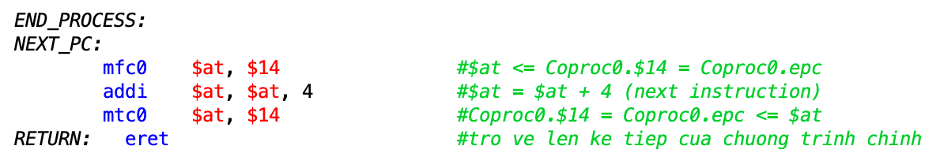
- Check whether the typing letter is BACKSPACE:

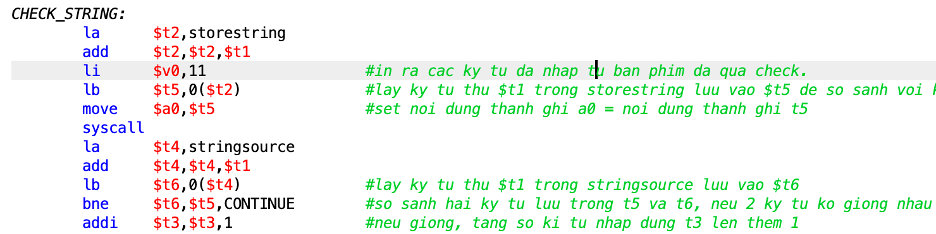
- Delete the previous letter:

- If the typing letter is not BACKSPACE and value of display\_ready is 1, then display the typed letter in MMIO display and store in array:

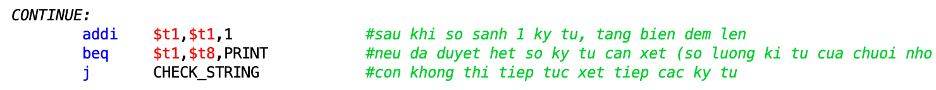
- If the typing letter is ‘\n’ branch to end of program:

- End the interrupt, reset location of register to the main and go to the next process:

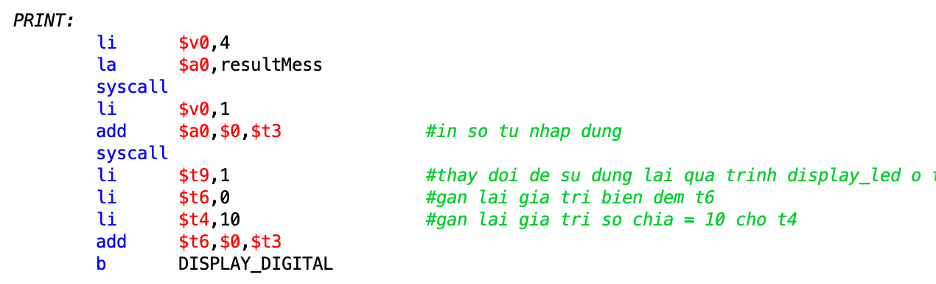
2.4. Typing accuracy check

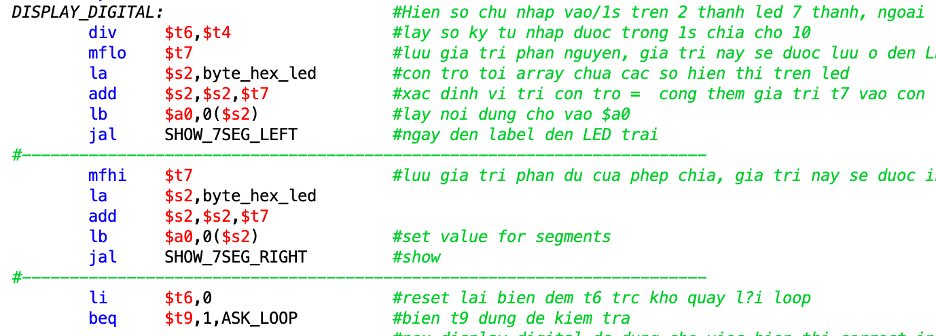
- Compare two strings: typing string(stored in storestring) and stringsource.

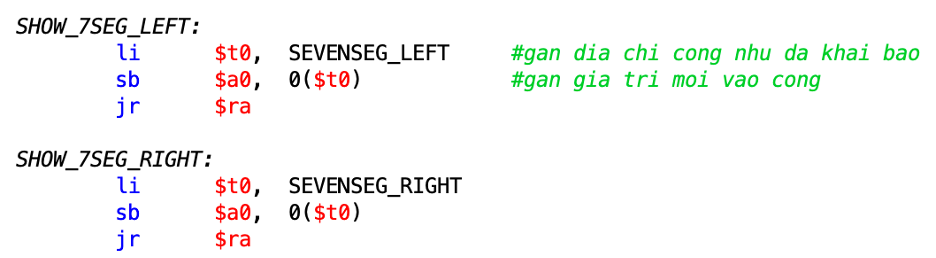
- Use two pointers to compare two corresponding two letter from two strings. If two letters are the same, then make addition to counter or change the pointers if not.

- Continue comparing until the end of shorter string.

- Print the number of correct letter to LED:

2.5. Print function to LED:





# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

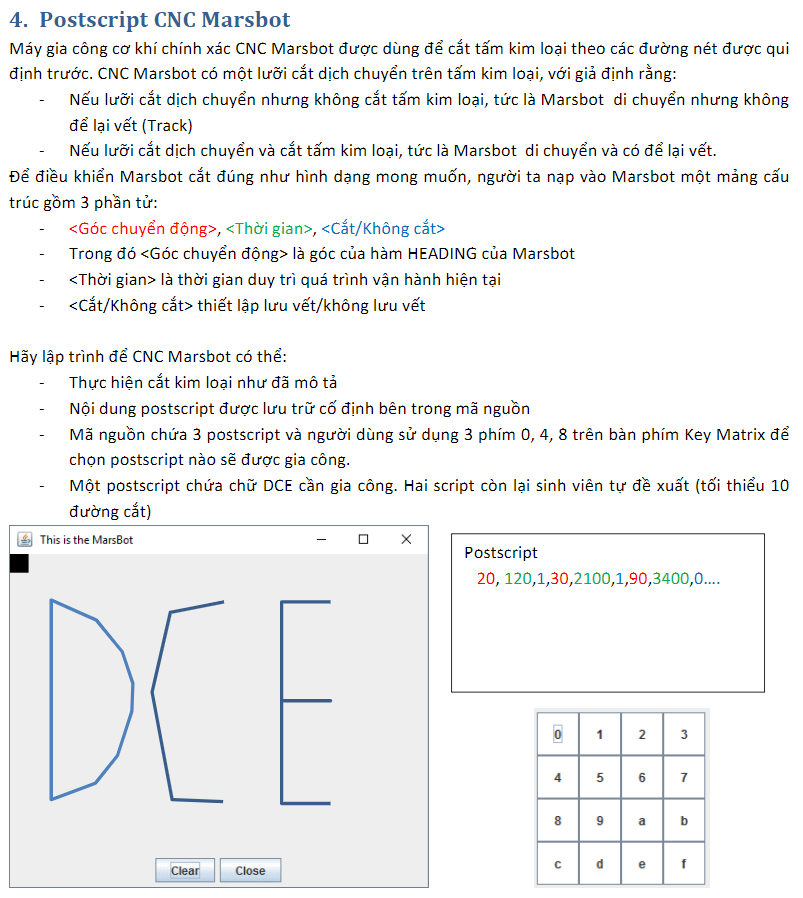
# 

# 

# Project 4: Postscript CNC Marsbot

Người chịu trách nhiệm chính: Đào Quang Dương

## About my project



## Problems and solutions

### Key Matrix

Digital Lab Sim is one of MARS tools for Controlling I/O devices.

The Hex keyboard of Digital Lab Sim is Memory-mapped I/O. It includes:

Byte value @0xFFFF0012 → used to command row number of hexadecimal keyboard.

Byte value @0xFFFF0014 → used to receive row and column number of the key pressed

(0 = no key pressed).



The program have to scan, one by one, each row and then observe if a key is pressed.

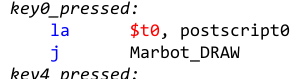
.text:

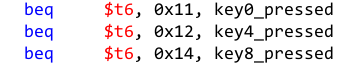
load I/O address



polling: example for row 1, if pressed, the keycode will store to $t6



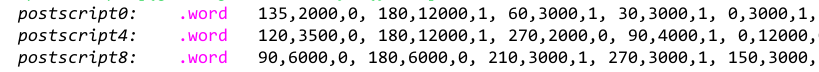
and then check the value to get the corresponding postscript



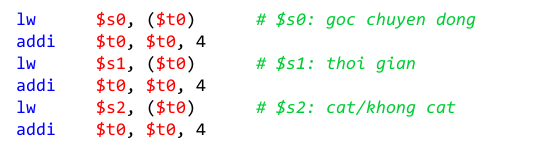
### Postscript

Postscripts is stored in a 3 arrays of word, separated by “,”

Postscript is the set of structs (góc chuyển động, thời gian, cắt hay không cắt)



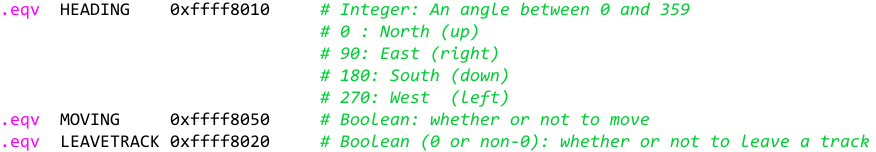
read\_postscript loops and read all the structs one by one.



### Marsbot

The MarsBot is a virtual robot in the MARS. The MarsBot can travel in a 2D space, optionally leaving a trail (or track).

This program uses the HEADING,LEAVETRACK, and MOVE to control the movement of the MarsBot.



STOP and GO change the value in the MOVING location to 0 or 1

TRACK and UNTRACK change the value in the LEAVETRACK location to 0 or 1

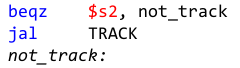
ROTATE change the value in the HEADING location to an angle between 0° and 359°

After reading 3 components of a step from the postscript, the Marsbot will be triggered to run.

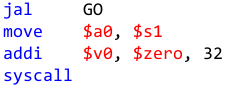
* ROTATE



* Check whether or not to leave a TRACK



* start GO for a time by sleep system



* STOP and UNTRACK to keep old track

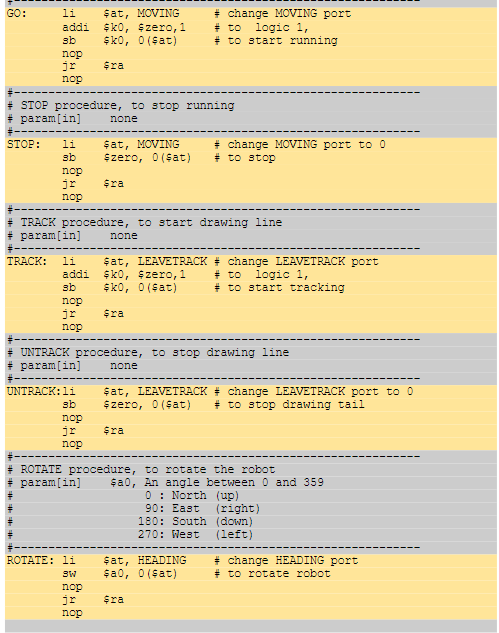


figure: Marsbot procedures

## 

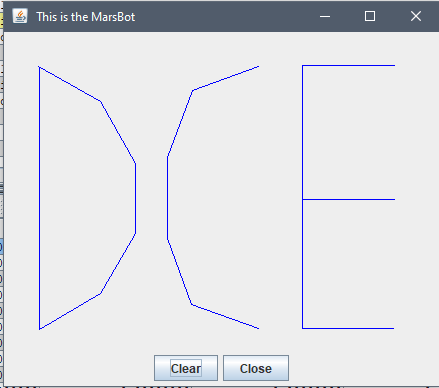
## Result

Run speed 30 inst/sec

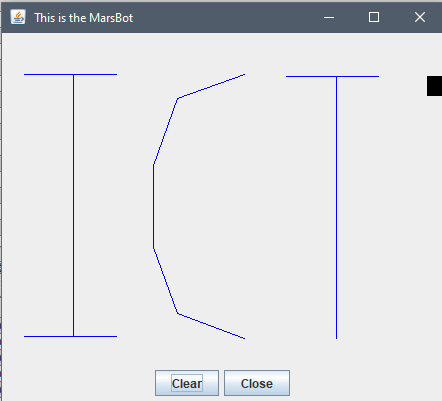
Open Digital Lab Sim and Connect to MIPS

Open MarsBot (clear after run a new postscript)

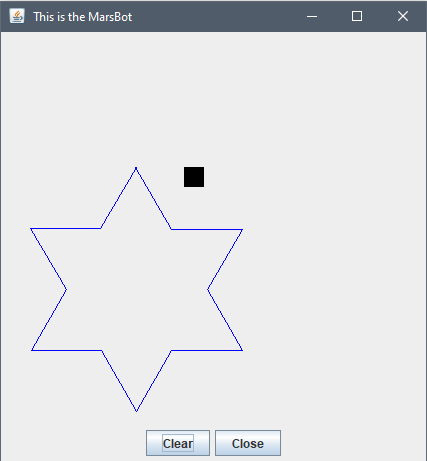
postscript 1 - key 0 - DCE - 15 tracks



postscript 2 - key 4 - ICT - 10 tracks



postscript 3 - key 8 - star - 12 tracks



## 

## Differences

1. Postscript

The number “-1” is added at the end of the array. It makes checking the last element of the array easier and faster while it doesn't affect anything. Easy to create new postscript without worrying about its length.

1. Key Scan





When polling, the previously pressed key is stored and used to check whether or not it is different from the currently pressed key. If not, the program will back\_to\_polling. That will save the marsbot from having to redraw the previous postscript when the user forgets to reset the hex keyboard.