ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
-----ꕥ 🕮 ꕥ-----

BÁO CÁO

BÀI TẬP CUỐI KỲ

THỰC HÀNH KIẾN TRÚC MÁY TÍNH

Mã học phần: IT3280
Mã lớp: 147799

Giảng viên hướng dẫn : **Lê Bá Vui**
Trợ giảng : **Đỗ Gia Huy**
Nhóm : **17**
Sinh viên thực hiện : **Nguyễn Đức Quang - 20225913**
Nguyễn Mạnh Tùng - 20225682

Hà Nội, tháng 6 năm 2024

Mục lục:

A. Chủ đề 1: Curiosity Marsbot
I. Mô tả yêu cầu
II. Chạy chương trình
    1. Hướng dẫn
    2. Xử lý ngoại lệ
III. Giải thích các hàm
IV. Minh họa kết quả
B. Chủ đề 10: Máy tính bỏ túi
I. Mô tả yêu cầu
II. Chạy chương trình
    1. Hướng dẫn
    2. Xử lý ngoại lệ
III. Nguyên lý hoạt động
IV. Minh họa kết quả

A. Chủ đề 1: Curiosity Marsbot

I. Mô tả yêu cầu

Xe tự hành Curioisity Marsbot chạy trên sao Hỏa, được vận hành từ xa bởi các lập trình viên trên Trái Đất. Bằng cách gửi đi các mã điều khiển từ một bàn phím ma trận, lập trình viên điều khiển quá trình di chuyển của Marbot như sau:

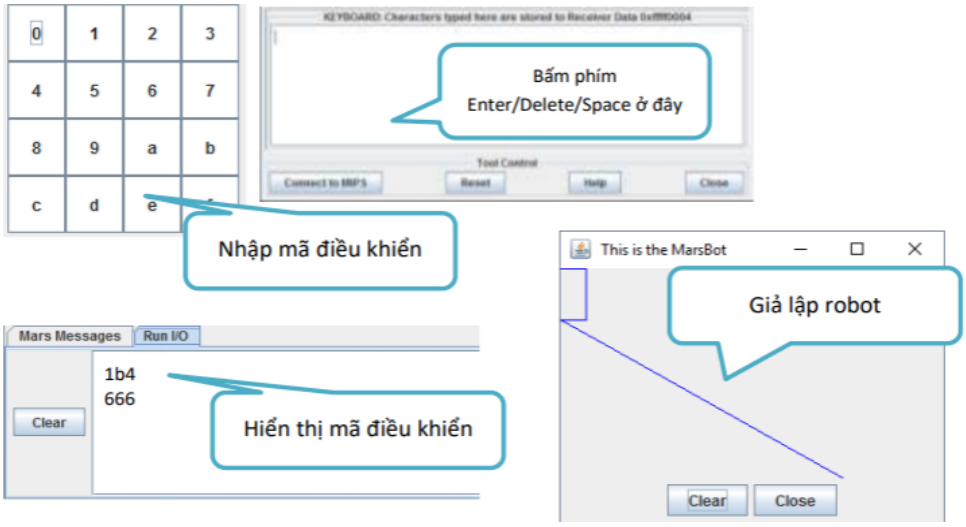
Mã điều khiển	Ý nghĩa
1b4	Marsbot bắt đầu chuyển động
c68	Marsbot đứng im
444	Rẽ trái 90 độ so với phương chuyển động gần nhất
666	Rẽ phải 90 độ so với phương chuyển động gần nhất
dad	Bắt đầu để lại vết trên đường
cbc	Chấm dứt để lại vết trên đường
999	Tự động đi theo lộ trình ngược lại. Không vẽ vết, không nhận mã khác cho tới khi kết thúc lộ trình ngược. Mô tả: Marsbot được lập trình để nhớ lại toàn bộ lịch sử các mã điều khiển và khoảng thời gian giữa các lần đổi mã. Vì vậy, nó có thể đảo ngược lại lộ trình để quay về điểm xuất phát

Sau khi nhận mã điều khiển, Curioisity Marsbot sẽ không xử lý ngay, mà phải đợi lệnh kích hoạt mã từ bàn phím Keyboard & Display MMIO Simulator. Có 3 lệnh như vậy:

Kích hoạt mã	Ý nghĩa
Phím Enter	Kết thúc nhập mã và yêu cầu Marsbot thực thi
Phím Delete	Xóa toàn bộ mã điều khiển đang nhập
Phím Space	Lặp lại lệnh đã thực hiện trước đó

Hãy lập trình để Marsbot có thể hoạt động như đã mô tả.

Đồng thời bổ sung thêm tính năng: mỗi khi gửi một mã điều khiển cho Marsbot, hiển thị mã đó lên màn hình console để người xem có thể giám sát lộ trình của xe.



II. Chạy chương trình

1. Hướng dẫn

Bước 1: Chạy chương trình, mở và kết nối Digital Lab Sim, Keyboard and Display MMIO, Marsbot

Bước 2: Nhập các mã điều khiển như bảng trên bằng Digital Lab Sim, nhấn các mã kích hoạt bằng Keyboard and Display

Bước 3: Quan sát Marsbot

Lưu ý:

Chú ý xem phím trên Digital Lab Sim đã chuyển sang màu xanh hay chưa (khi chuyển màu xanh tức là đã nhận giá trị).

2. Xử lý ngoại lệ

- Khi nhập sai mã điều khiển, chương trình sẽ thông báo lỗi "Ma khong hop le"
- Khi nhấn kí tự khác ngoài các mã kích hoạt enter/space/delete, chương trình sẽ bỏ qua và tiếp tục polling

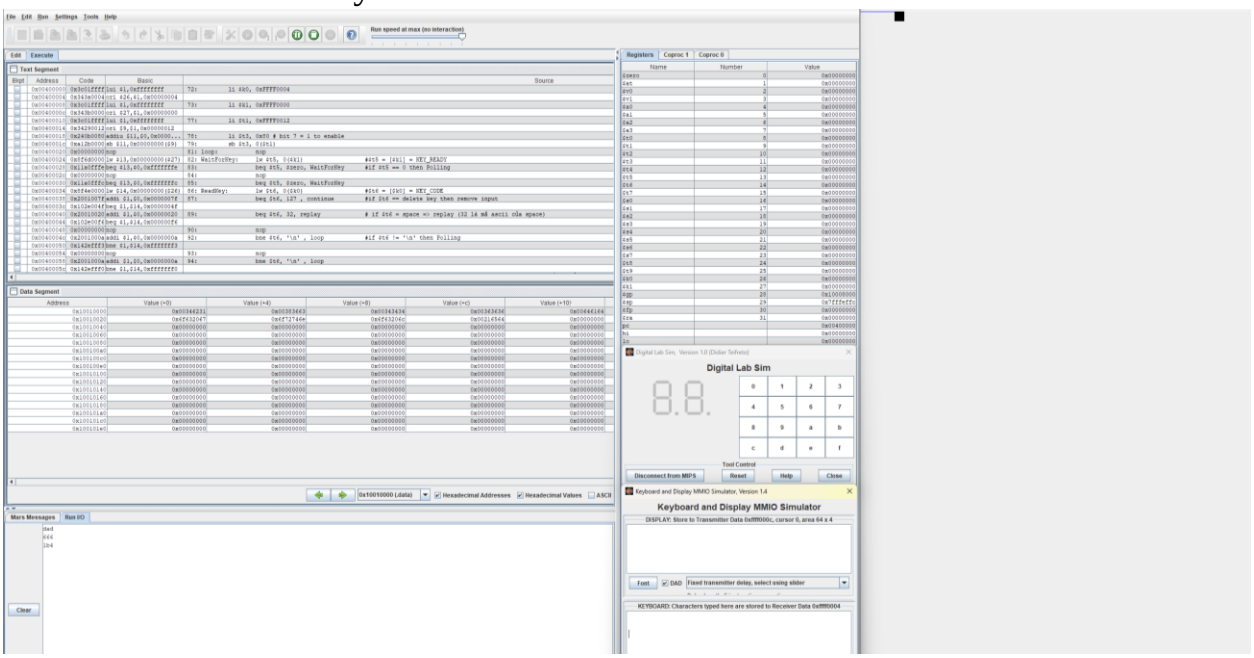
III. Giải thích các hàm

- CheckControlCode: kiểm tra xem input code có giống các mã điều khiển
- printError: in lỗi khi input code không hợp lệ
- printControlCode: in mã điều khiển vừa thực thi
- removeControlCode: xóa input code điều khiển sau khi đã thực thi
- storePath: lưu vết các lần rẽ của marsbot, mỗi lần rẽ sẽ lưu (x,y,z) với tọa độ x, y tại vị trí rẽ và z là hướng rẽ khi đó tương đương 12 bytes
- path: mảng lưu các cấu trúc (x,y,z) trên
- lengthOfPath: lưu độ dài của mảng path
- các lệnh turnleft, turnright, replay,turnback, track, untrack thực hiện các chức năng tương ứng
- Trước khi thực hiện các procedure, thực hiện backup dữ liệu của các thanh khi vào $sp sau khi thực hiện procedure thì restore dữ liệu tương ứng

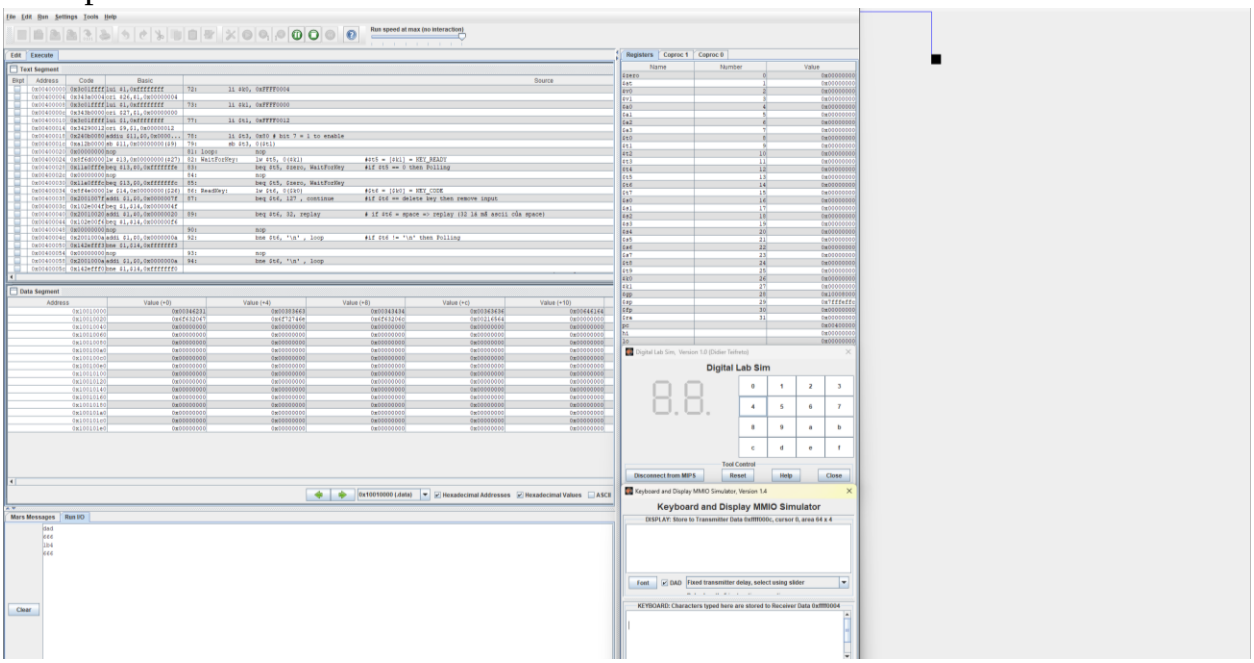
IV. Minh họa kết quả

- Chạy thử chương trình:

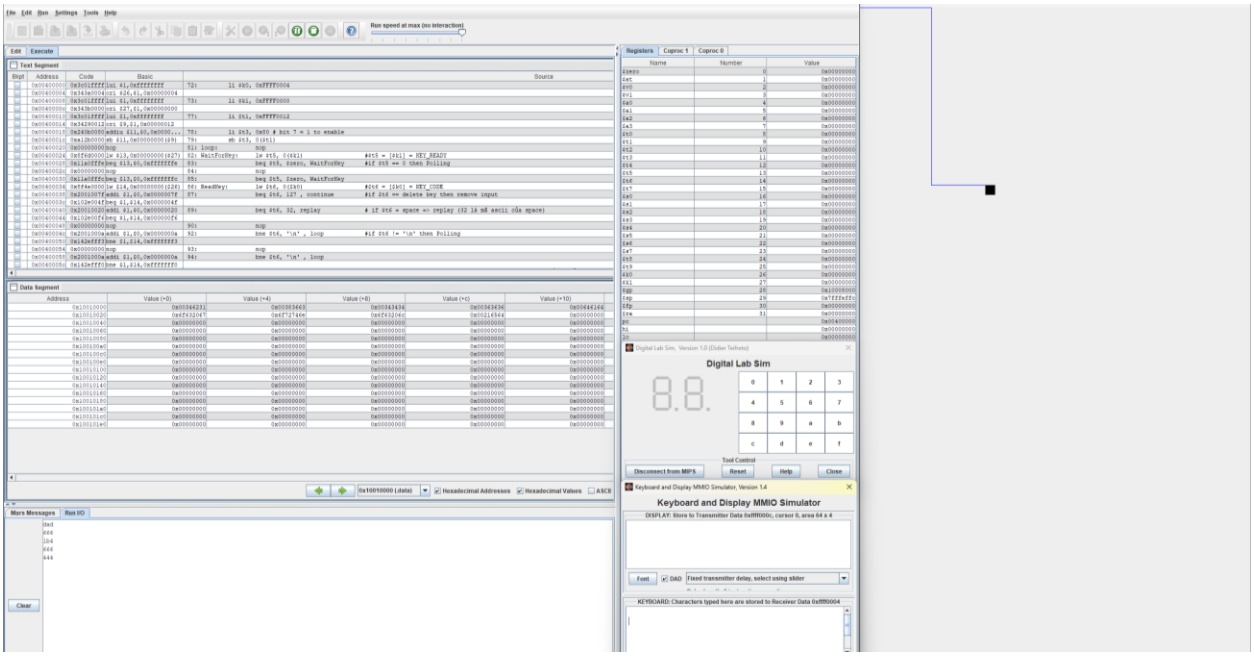
+ 2 lệnh dad và 666 đầu tiên để marsbot thực hiện lưu vết và đi ngang
+ 1b4 để bắt đầu di chuyển



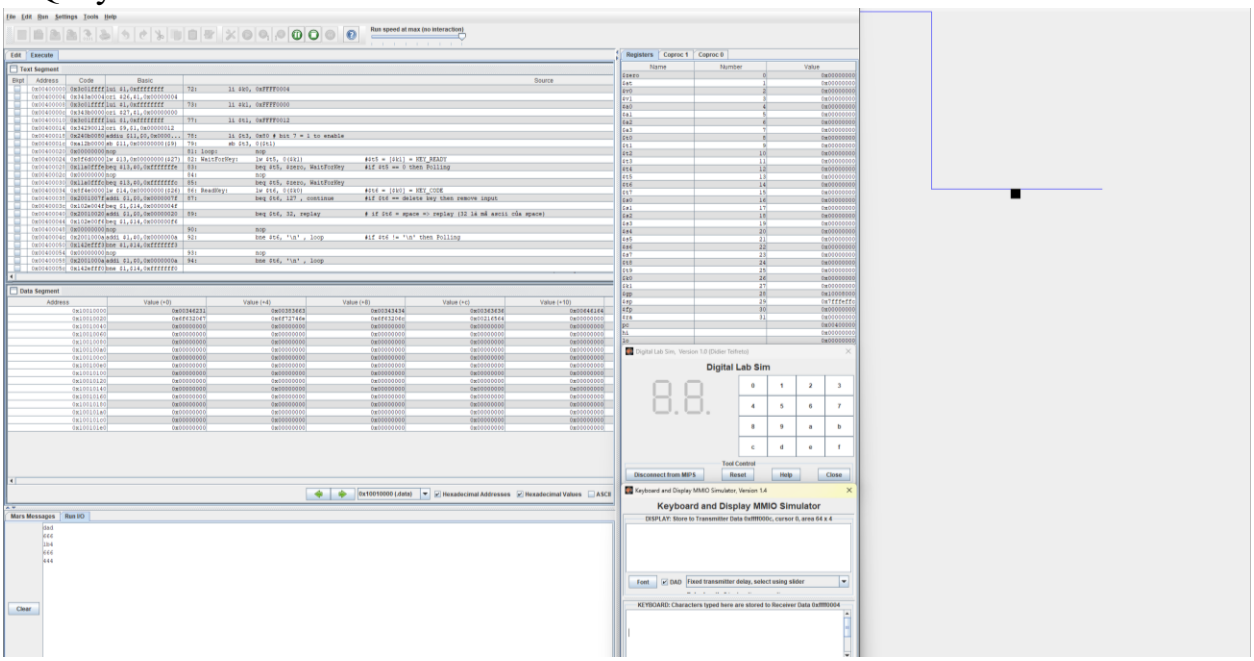
+ Rẽ phải:

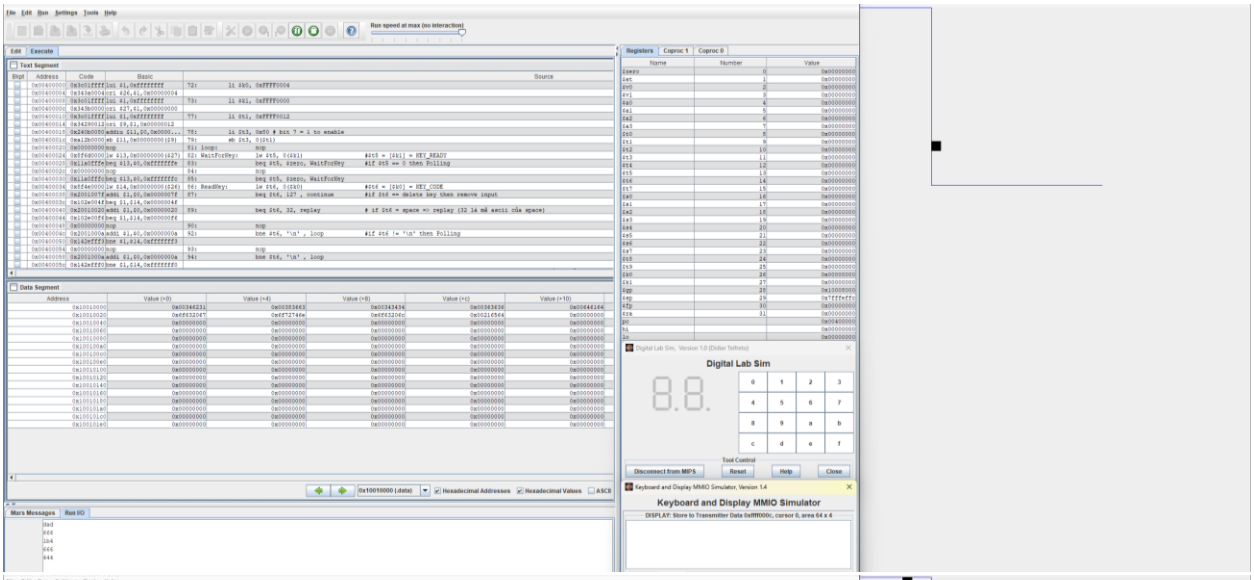


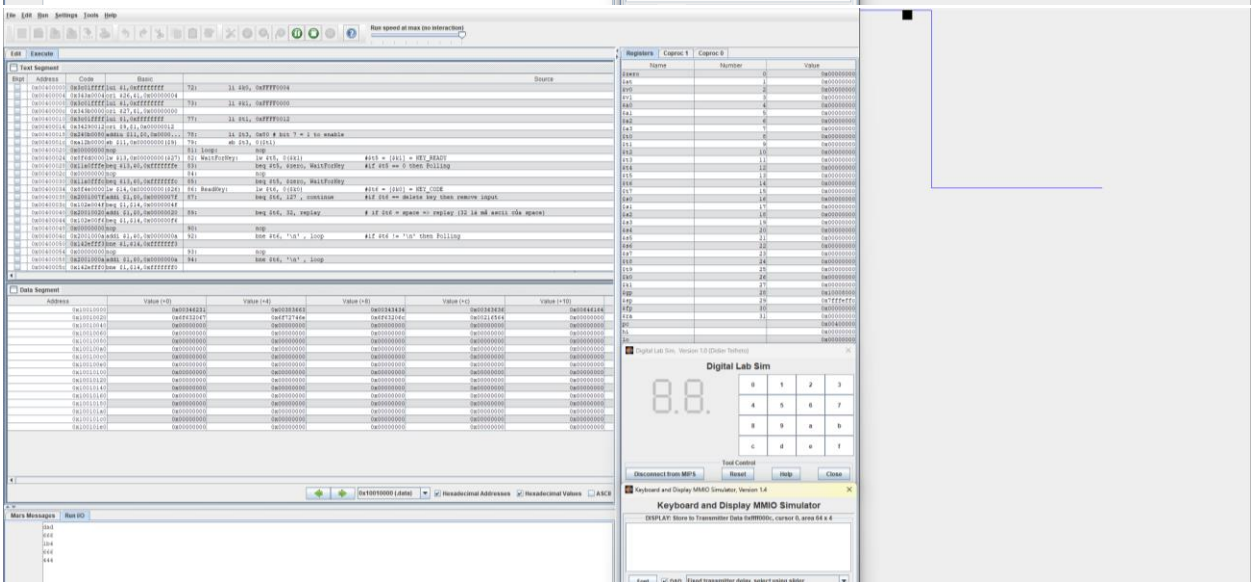
+ Rẽ trái:



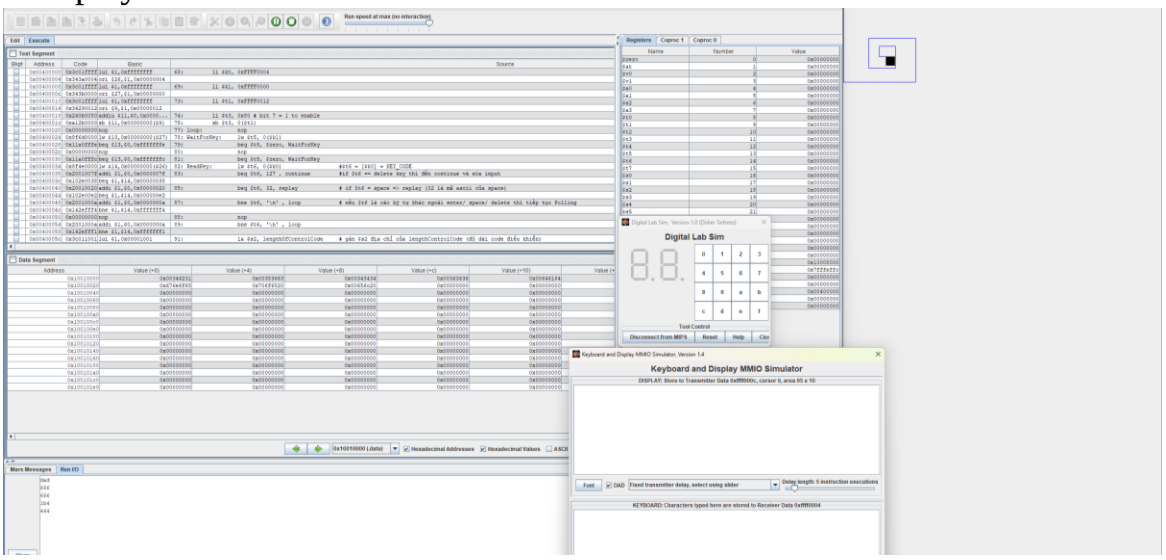
+ Quay đầu:







+ Replay:



B. Máy tính bỏ túi

I. Mô tả yêu cầu

- Sử dụng bàn phím KEYPAD và LED 7 thanh để xây dựng một máy tính bỏ túi đơn giản. Hỗ trợ thực hiện các phép toán +, -, *, /, % với các toán hạng là số nguyên. Với các phím chức năng:

+ Bấm phím ‘a’ để nhận phép tính ‘+’
+ Bấm phím ‘b’ để nhận phép tính ‘-’
+ Bấm phím ‘c’ để nhận phép tính ‘*’
+ Bấm phím ‘d’ để nhậnphép tính ‘/’
+ Bấm phím ‘e’ để nhận phép tính ‘%’
+ Bấm phím ‘f’ để nhận phép ‘=’

- Cụ thể:

+ Khi nhấn các phím số, hiển thị lên LED, do chỉ có 2 LED nên chỉ hiển thị hai số cuối cùng. Ví dụ khi nhấn phím 1 → hiển hị 01. Khi nhấn thêm phím 2 → hiển thị 12. Khi nhấn thêm phím 3 → hiển thị 23.
+ Sau khi nhập số sẽ nhập phép tính +, -, *, /, %
+ Sau khi nhấn phím f (dấu ‘=’), tính toán và hiển thị kết quả lên LED.
+ Có thể thực hiện các phép tính liên tiếp.

II. Chạy chương trình

1. Hướng dẫn

Bước 1: Chạy chương trình, mở Digital Lab Sim và kết nối với Mips
Bước 2: Nhập số thứ nhất → nhập toán tử → nhập số thứ hai
Bước 3: Nhập ‘f’ (dấu ‘=’)

Lưu ý:

Chú ý xem phím trên Digital Lab Sim đã chuyển sang màu xanh hay chưa (khi chuyển màu xanh tức là đã nhận giá trị).

2. Xử lý ngoại lệ

- Khi thực hiện chia cho ‘0’: Khi người dùng nhập số thứ 2 là ‘0’ và nhập toán tử là ‘/’ hoặc ‘%’, chương trình sẽ thông báo lỗi “khong chia duoc cho 0!”
- Khi người dùng nhập liên tiếp hai toán tử, chương trình sẽ coi là thực hiện phép tính với 0 và in ra kết quả
- Với kết quả là số âm (<0) sẽ hiện thị số đối của kết quả trên LED, kết quả đúng vẫn được hiển thị trên màn hình RUN I/O

3. Hạn chế

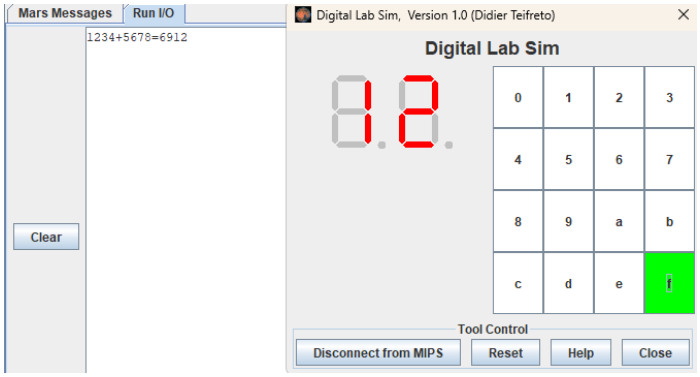
Chưa xây dựng được chức năng thực hiện liên tiếp các phép tính (chỉ thực hiện được với phép tính có hai toán tử và một toán hạng)

III. Nguyên lý hoạt động

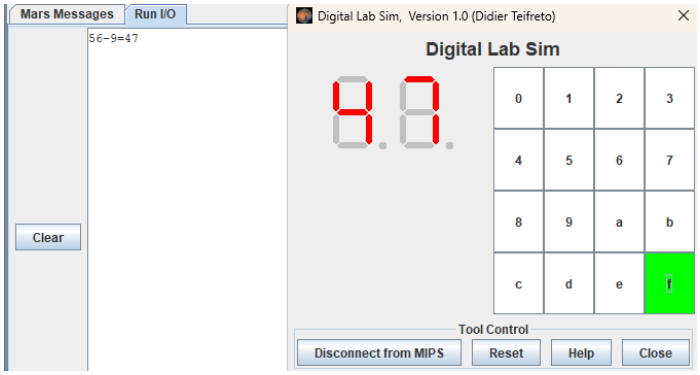
- Khởi tạo biến và giá trị:
- Sau khi khởi tạo biến và giá trị, chương trình đi vào vòng lặp vô tận chờ ngắt từ bàn phím:
- Xử lý ngắt:
+ Khi ngắt, chương trình sẽ kiểm tra theo hàng để xác định phím đã được nhấn
+ Giải mã phím đã nhấn
+ Thực hiện các hoạt động tiếp theo dựa trên phím đã giải mã
+ Hàm *check_row_X* thực hiện kiểm tra xem có nút được nhấn trên hàng *X* hay không và tiến hành gọi hàm *get_value* lấy giá trị
+ Hàm *convert_row_X* thực hiện hiển thị các giá trị là toán hạng, tiếp tục xử lý phép tính nếu là toán tử.
+ Hàm *set_first_number* và *set_second_number* được gọi khi nhập số thứ nhất và số thứ hai, sau đó sẽ thực hiện các phép toán
+ Sau khi thực hiện tính toán sẽ hiển thị kết quả lên LED và in phép tính lên màn hình RUN I/O

IV. Minh họa kết quả

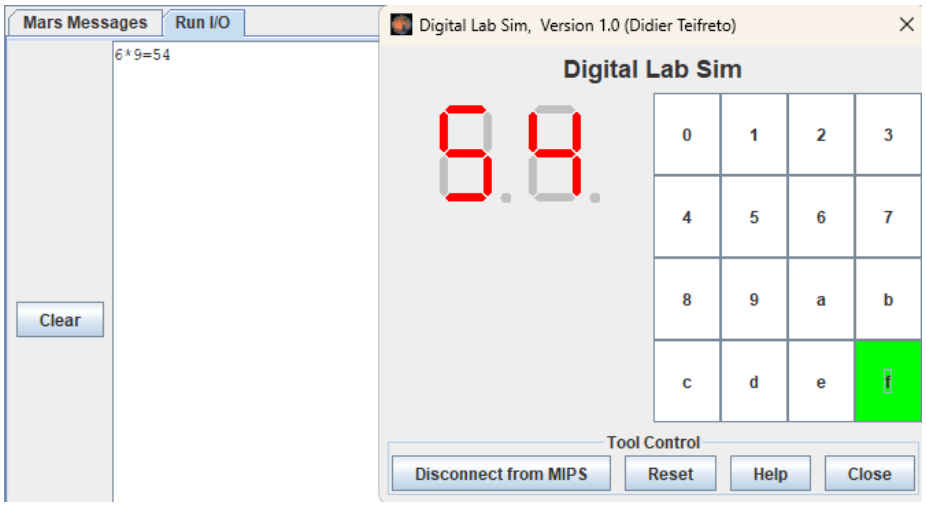
Các phép toán thông thường:
+ Phép cộng: 1234 + 5678 = 6912



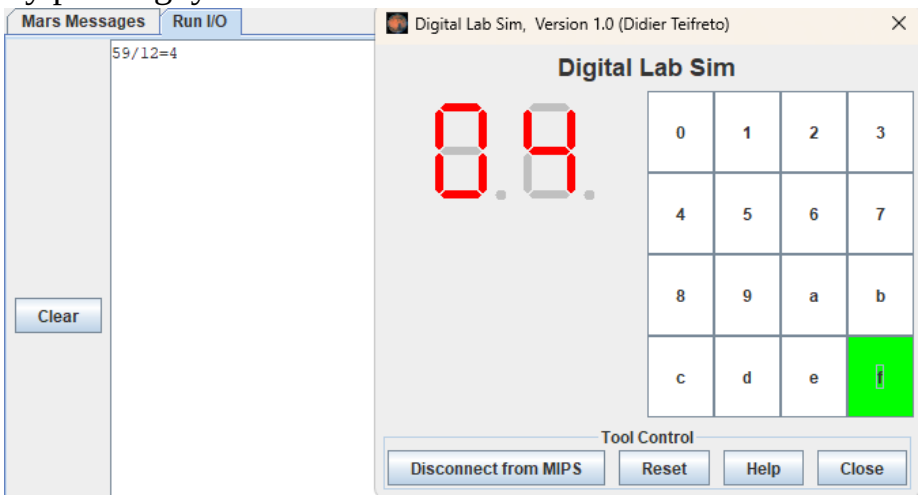
+ Phép trừ: 56 – 9 =47



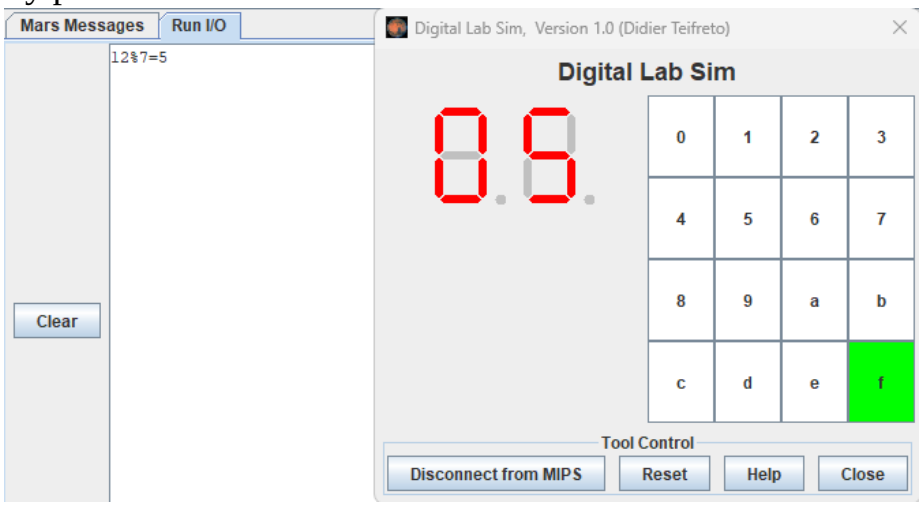
+ Phép nhân: 6 * 9 = 54



+ Phép chia lấy phần nguyên: 59 / 12 = 4

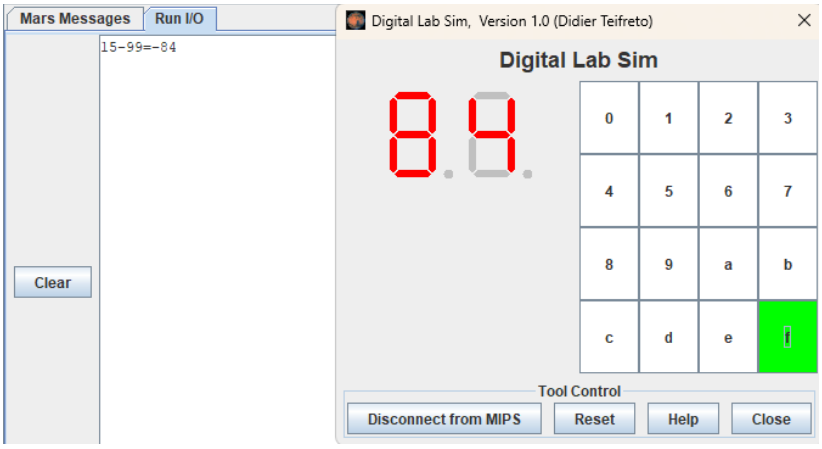


+ Phép chia lấy phần dư: 12 % 7 = 5

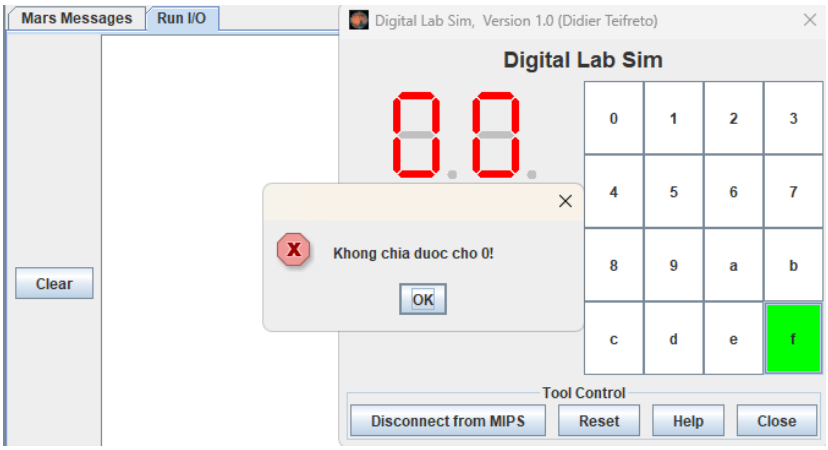


Các phép toán ngoại lệ:

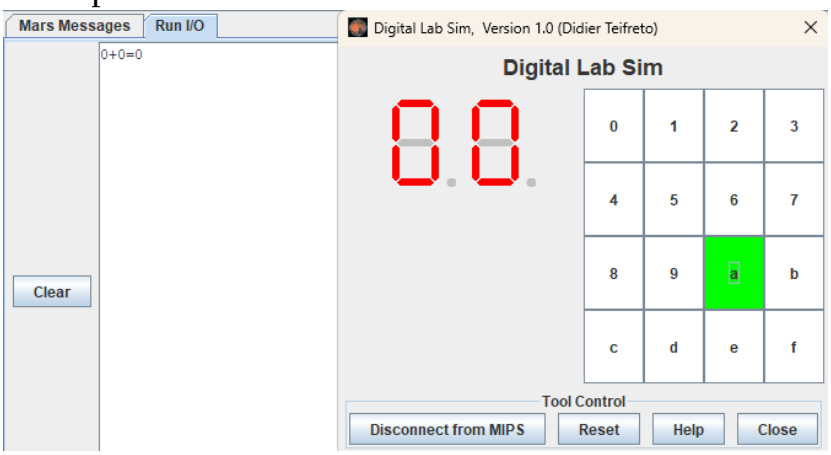
+ Kết quả phép trừ < 0: 15 – 99 = -84

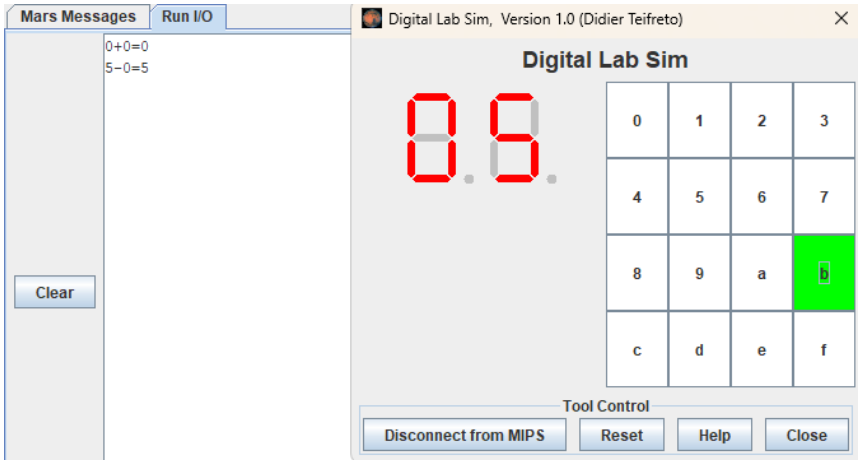


+ Khi chia cho ‘0’:



+ Khi nhấn liên tiếp toán tử:




Mars Messages
Run I/O
0+0=0
5-0=5
Clear
Digital Lab Sim, Version 1.0 (Didier Teifreto)
Digital Lab Sim
0 1 2 3
4 5 6 7
8 9 a b
c d e f
Tool Control
Disconnect from MIPS
Reset
Help
Close