

Bài 1:

1. Đề bài

1. Curiosity Marsbot

Xe tự hành Curiosity 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 Marsbot như sau:

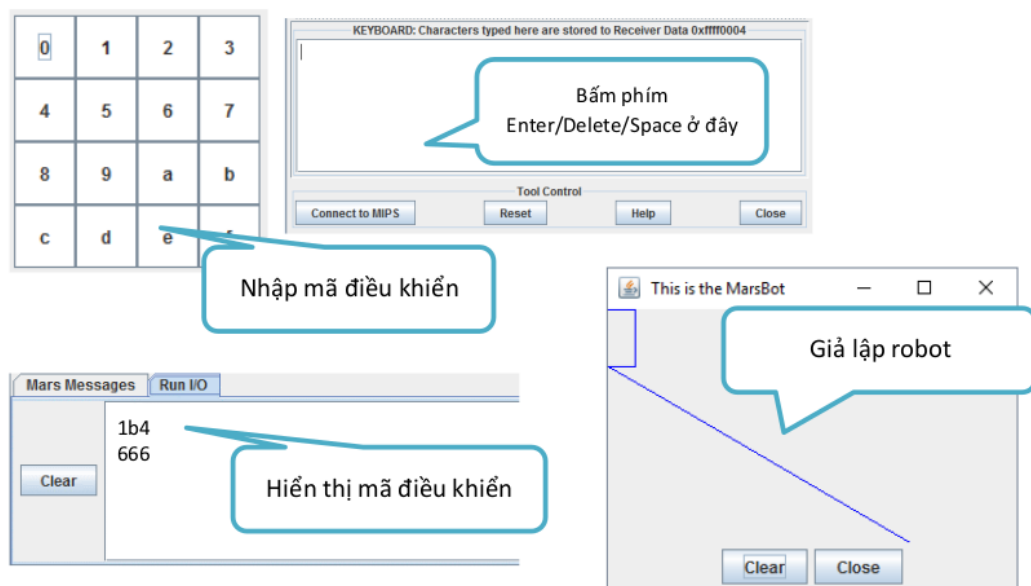
Mã điều khiển	Ý nghĩa
1b4	Marsbot bắt đầu chuyển động
c68	Marsbot đứng im
444	Rẽ trái 90 độ so với phương chuyển động gần nhất
666	Rẽ phải 90 độ so với phương chuyển động gần nhất
dad	Bắt đầu để lại vết trên đường
cbc	Chấm dứt để lại vết trên đường
999	Tự động đi theo lộ trình ngược lại. Không vẽ vết, 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, Curiosity 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	Lập 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.



2. Thuật toán:

- **Bước 1:** Mỗi khi người dùng nhập 1 ký tự từ Digital Lab Sim sẽ tạo ra interrupt để lưu ký tự được nhập vào bộ nhớ, tạo nên đoạn code điều khiển.
Ký tự được nhập vào sẽ được lưu trong chuỗi inputControlCode, tăng độ dài của chuỗi lên 1 và thêm ký tự kết thúc '\0' và cuối chuỗi.
- **Bước 2:** Kiểm tra liên tục xem các lệnh kích hoạt mã có được nhập ở Keyboard & Display MMIO Simulator hay không.
 - Khi ký tự Enter được nhập, sẽ kiểm tra xem đoạn code điều khiển có hợp lệ không (gồm 3 ký tự), nếu không sẽ thông báo code lỗi và sang bước 4. Nếu có thì chuyển sang bước 3.
 - Nếu ký tự Delete được nhập, tiến hành xóa toàn bộ mã điều khiển đang nhập
 - Nếu ký tự Space được nhập, lặp lại lệnh đã thực hiện trước đó.
- **Bước 3:** Lần lượt kiểm tra xem code điều khiển được nhập vào có trùng với các đoạn code điều khiển đã quy định sẵn. Nếu không thì thông báo đoạn code bị lỗi. Ngược lại thực hiện thao tác theo quy định sẵn:
 - Để marsbot có thể nhớ được được toàn bộ mã điều khiển. Chúng ta sử dụng mảng path để lưu trữ 1 structure bao gồm tọa độ của điểm (x, y) và hướng của marsbot tại thời điểm hành động thay đổi.
 - Khi marsbot đi theo lộ trình ngược lại chỉ cần duyệt mảng path từ phía cuối và đảo ngược hướng được lưu tại mỗi điểm.
 - Khi cho marsbot lặp lại lệnh thực hiện trước đó thì cũng cần phải lưu vào mảng path. Để có thể thực hiện lại lệnh trước đó chỉ cần xét hướng trong structure cuối cùng trong mảng path đồng thời sử dụng biến isRight để kiểm tra xem trước đó marsbot đang đi thẳng (2), rẽ phải (1) hay rẽ trái (0). Nếu

rẽ phải thì cộng thêm 90 vào hướng được lấy ra, rẽ trái trừ 90 vào hướng đó còn đi thẳng thì không thay đổi hướng.

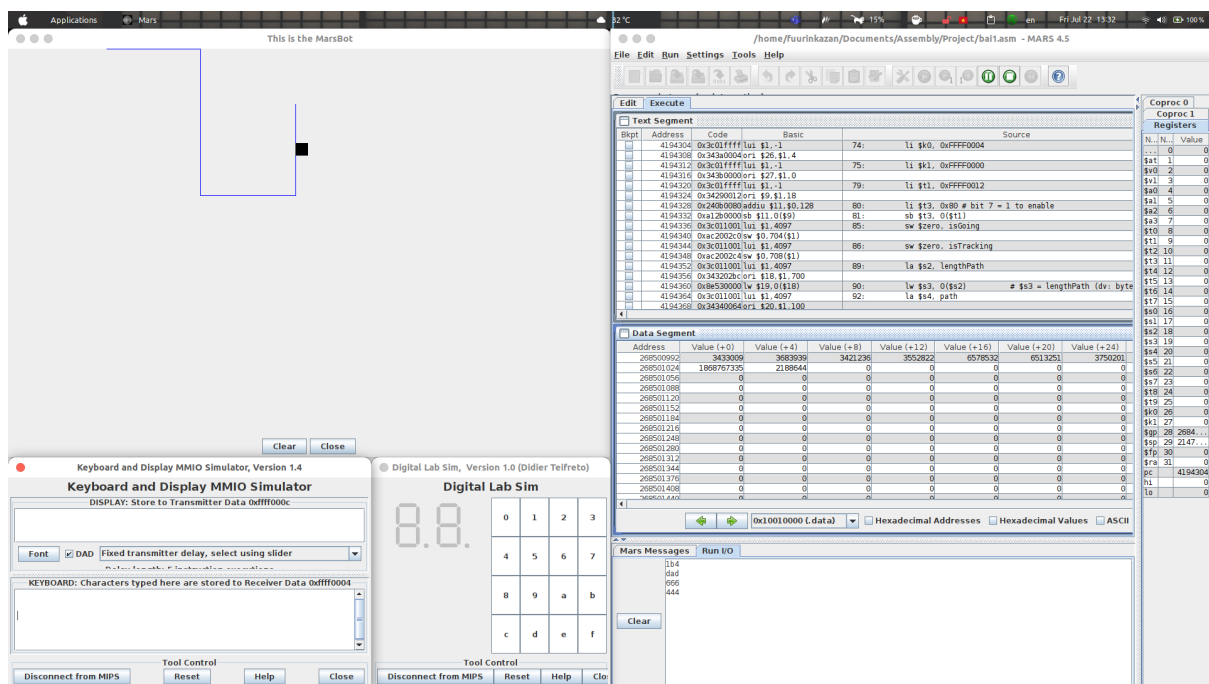
- **Bước 4:** In ra trên màn hình console code điều khiển đã nhập và xóa lưu trữ trong bộ nhớ.

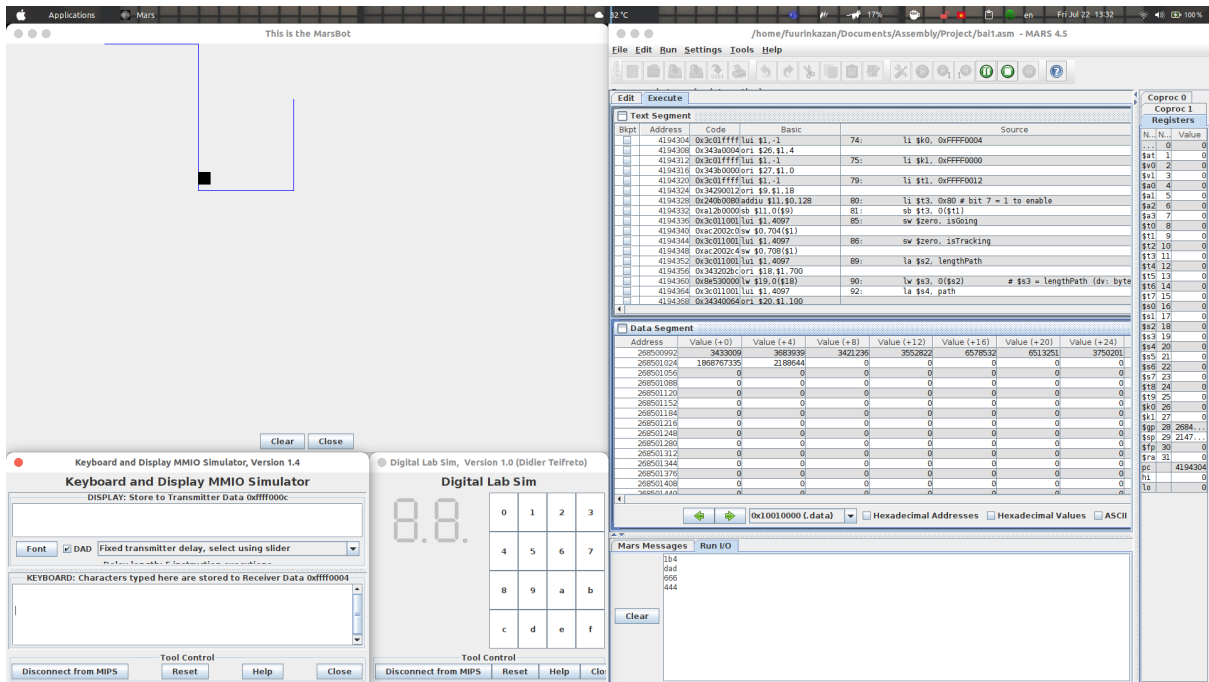
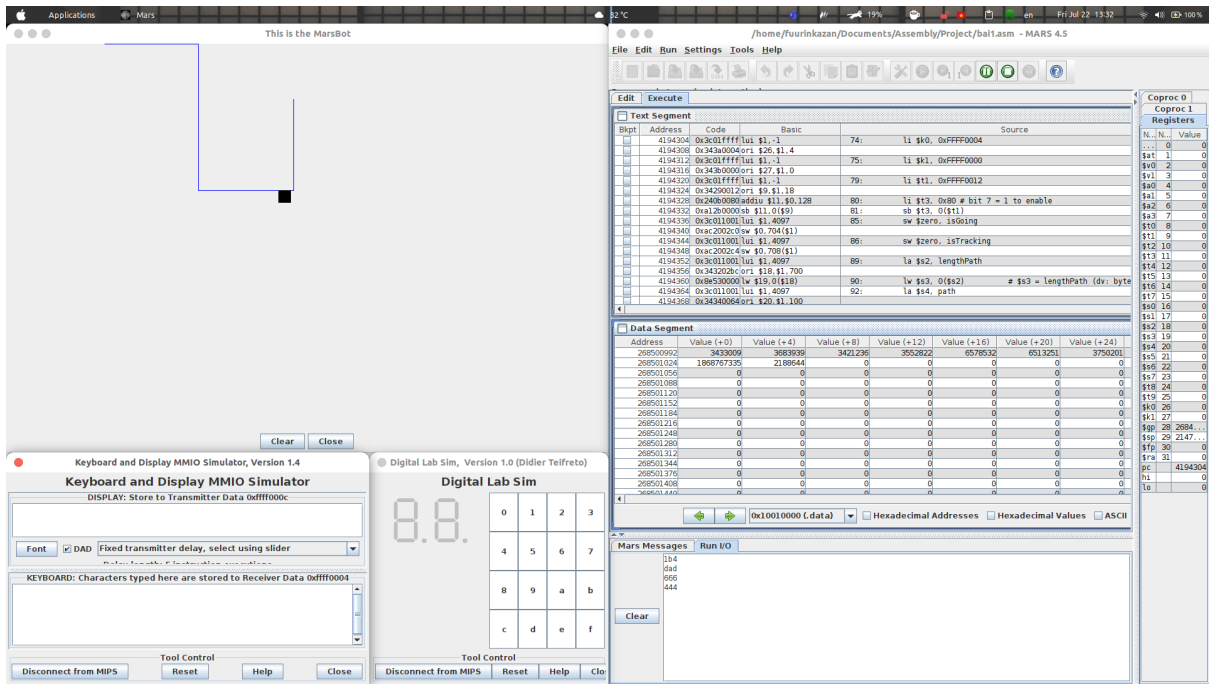
3. Giải thích mã nguồn

- Khởi tạo các mã code và gán vào các nhãn.
- **inputControlCode:** Chuỗi lưu trữ mã điều khiển.
lengthControlCode: Độ dài mảng chuỗi inputControlCode
nowHeading: Hướng (góc) hiện tại.
path: mảng lưu trữ vị trí mà marsbot thay đổi hành động. Vị trí là 1 structure (12 bytes) bao gồm tọa độ của marsbot (x, y) và hướng tại điểm hành động thay đổi (z).
lengthPath: Độ dài mảng path
isGoing(boolean): Kiểm tra marsbot có đang di chuyển hay không.
isTracking(boolean): Kiểm tra marsbot có đang để lại vết hay không.
isRight(int): Kiểm tra marsbot đang đi thẳng, rẽ trái hay rẽ phải..
- **Phần init(khởi tạo):** Khởi tạo gốc ban đầu của marsbot và quay marsbot sang phải để nó không di chuyển ra khỏi hộp dialog, đồng thời lưu trữ gốc đó vào mảng path và tăng chiều dài của mảng path thêm 12 để dùng cho lần lưu trữ tiếp theo.
- **check:** Kiểm tra mã điều khiển nhập vào có hợp lệ hay không. Hợp lệ ở đây là chiều dài đã bằng 3 hay chưa hoặc có nằm trong các mã điều khiển đã được quy định hay không. Nếu vi phạm 1 trong 2 điều kiện trên thì thông báo lỗi.
- **repeat:** Thực hiện lệnh trước đó của marsbot. Nếu marsbot đang đứng yên thì vẫn tiếp tục đứng yên, nếu đang di chuyển thẳng trước đó thì vẫn tiếp tục di chuyển thẳng. Lấy structure cuối của mảng path, sau đó trích xuất góc trong structure, kiểm tra xem trước đó marsbot có rẽ trái hay phải không bằng biến isRight. Nếu rẽ phải thì thêm 90 vào góc được lấy ra đó ngược lại thì giảm 90 nếu trước đó rẽ trái.
- **printControlCode:** In mã điều khiển ra console.

- **storePath**: Lưu trữ tọa độ hiện tại thông qua WHEREX, WHEREY và nowHeading vào mảng path. Tăng độ dài mảng path thêm 12 để sử dụng cho lần lưu trữ sau.
- **goBack**: Đi theo lộ trình ngược lại. Đầu tiên là tắt keypad ở DigitalLabSim và tắt track. Đọc mảng path từ cuối lên, lấy góc trong từng structure sau đó đảo ngược để có thể đi đúng hướng ngược lại. Cho marsbot di chuyển đồng thời kiểm tra tọa độ hiện tại với tọa độ được lưu trữ trong structure. Nếu trùng cả x và y thì quay lại begin để đọc tiếp structure tiếp theo.
- **finish**: Reset lại tọa độ ban đầu, bật interrupt của DigitalLabSim.
- **track**: Nhảy đến nhãn TRACK và in mã lệnh điều khiển.
- **untrack**: Nhảy đến nhãn UNTRACK và in mã lệnh điều khiển.
- **go**: Khi nhập mã điều khiển "1b4" thì cập nhật giá trị của isRight = 2 tức là đang đi thẳng, để dùng cho hàm repeat. Sau đó nhảy đến nhãn GO và in mã lệnh điều khiển.
- **stop**: Nhảy đến nhãn STOP và in mã lệnh điều khiển.
- **goRight**: Cho marsbot dừng, untrack. Cộng hướng hiện tại thêm 90 để marsbot rẽ phải và isRight = 1(repeat). Nếu đang track trước đó thì tiếp tục track.
- **goLeft**: Trừ hướng hiện tại đi 90 để marsbot rẽ trái và isRight = 0(repeat)
- **remove**: Xóa mã lệnh điều khiển trong chuỗi inputControlCode
- **isEqual**: Kiểm tra xem inputControlCode có trùng với các code điều khiển hay không, nếu trùng thì trả về 1 không thì trả về 0 (sử dụng cho hàm check)
- **error**: In mã lỗi và thông báo lỗi sau đó xóa mã lỗi trong inputControlCode để chuẩn bị cho lần nhập tiếp theo.
- **GO, STOP, TRACK, UNTRACK, ROTATE** -> thực hiện các hoạt động của marsbot.
- **get_code, get_code_in_char**: đọc keypad được nhập từ DigitalLabSim
- **store_code**: Lưu ký tự vừa được nhập vào chuỗi inputControlCode và tăng chiều dài lengthControlCode lên 1, thêm ký tự kết thúc chuỗi '\0' vào cuối chuỗi.

4. Demo:





This is the MarsBot

Keyboard and Display MMIO Simulator, Version 1.4

DISPLAY: Store to Transmitter Data 0xffff000c

Font: DAD Fixed transmitter delay, select using slider

KEYBOARD: Characters typed here are stored to Receiver Data 0xffff0004

Digital Lab Sim, Version 1.0 (Didier Telfreto)

8.8.

0 1 2 3
4 5 6 7
8 9 a b
c d e f

Tool Control: Disconnect from MIPS, Reset, Help, Close

File Edit Run Settings Tools Help

Text Segment

Bkpt	Address	Code	Basic	Source
	4194300	0x3c01ffff	lui \$1, -1	74: lui \$k0, 0xffff0004
	4194308	0x343a0004	ori \$26, \$1, 4	75: lui \$k1, 0xffff0000
	4194312	0x3c01ffff	lui \$1, -1	
	4194316	0x343a0004	ori \$27, \$1, 0	
	4194320	0x3c01ffff	lui \$1, -1	
	4194324	0x343a0004	ori \$28, \$1, 18	
	4194328	0x24b00000	addiu \$11, \$0, 128	80: lui \$13, 0x80 # bit 7 = 1 to enable
	4194332	0x12b0000b	lui \$11, 0(\$9)	81: sb \$13, 0(\$11)
	4194336	0x3c011001	lui \$1, 4097	85: sw \$zero, isTracking
	4194340	0x20020000	sw \$0, 704(\$1)	
	4194344	0x3c011001	lui \$1, 4097	
	4194348	0x20020000	sw \$0, 708(\$1)	
	4194352	0x3c011001	lui \$1, 4097	
	4194356	0x3432000c	ori \$18, \$1, 700	
	4194360	0x9b530000	lw \$19, 0(\$18)	90: lw \$s3, 0(\$s2) # \$s3 = LengthPath (dv: byte
	4194364	0x3c011001	lui \$1, 4097	92: lw \$s4, path
	4194368	0x343a0004	ori \$20, \$1, 100	

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)
26950092	3433009	368399	3421296	355282	6578532	6513251	3750201
26950104	1868787395	2188644	0	0	0	0	0
269501056	0	0	0	0	0	0	0
269501088	0	0	0	0	0	0	0
269501120	0	0	0	0	0	0	0
269501152	0	0	0	0	0	0	0
269501184	0	0	0	0	0	0	0
269501216	0	0	0	0	0	0	0
269501248	0	0	0	0	0	0	0
269501280	0	0	0	0	0	0	0
269501312	0	0	0	0	0	0	0
269501344	0	0	0	0	0	0	0
269501376	0	0	0	0	0	0	0
269501408	0	0	0	0	0	0	0
269501440	0	0	0	0	0	0	0

Mars Messages Run I/O

134
dad
666
444

Clear

This is the MarsBot

Keyboard and Display MMIO Simulator, Version 1.4

DISPLAY: Store to Transmitter Data 0xffff000c

Font: DAD Fixed transmitter delay, select using slider

KEYBOARD: Characters typed here are stored to Receiver Data 0xffff0004

Digital Lab Sim, Version 1.0 (Didier Telfreto)

8.8.

0 1 2 3
4 5 6 7
8 9 a b
c d e f

Tool Control: Disconnect from MIPS, Reset, Help, Close

File Edit Run Settings Tools Help

Text Segment

Bkpt	Address	Code	Basic	Source
	4194300	0x3c01ffff	lui \$1, -1	74: lui \$k0, 0xffff0004
	4194308	0x343a0004	ori \$26, \$1, 4	75: lui \$k1, 0xffff0000
	4194312	0x3c01ffff	lui \$1, -1	
	4194316	0x343a0004	ori \$27, \$1, 0	
	4194320	0x3c01ffff	lui \$1, -1	
	4194324	0x343a0004	ori \$28, \$1, 18	
	4194328	0x24b00000	addiu \$11, \$0, 128	80: lui \$13, 0x80 # bit 7 = 1 to enable
	4194332	0x12b0000b	lui \$11, 0(\$9)	81: sb \$13, 0(\$11)
	4194336	0x3c011001	lui \$1, 4097	85: sw \$zero, isTracking
	4194340	0x20020000	sw \$0, 704(\$1)	
	4194344	0x3c011001	lui \$1, 4097	
	4194348	0x20020000	sw \$0, 708(\$1)	
	4194352	0x3c011001	lui \$1, 4097	
	4194356	0x3432000c	ori \$18, \$1, 700	
	4194360	0x9b530000	lw \$19, 0(\$18)	90: lw \$s3, 0(\$s2) # \$s3 = LengthPath (dv: byte
	4194364	0x3c011001	lui \$1, 4097	92: lw \$s4, path
	4194368	0x343a0004	ori \$20, \$1, 100	

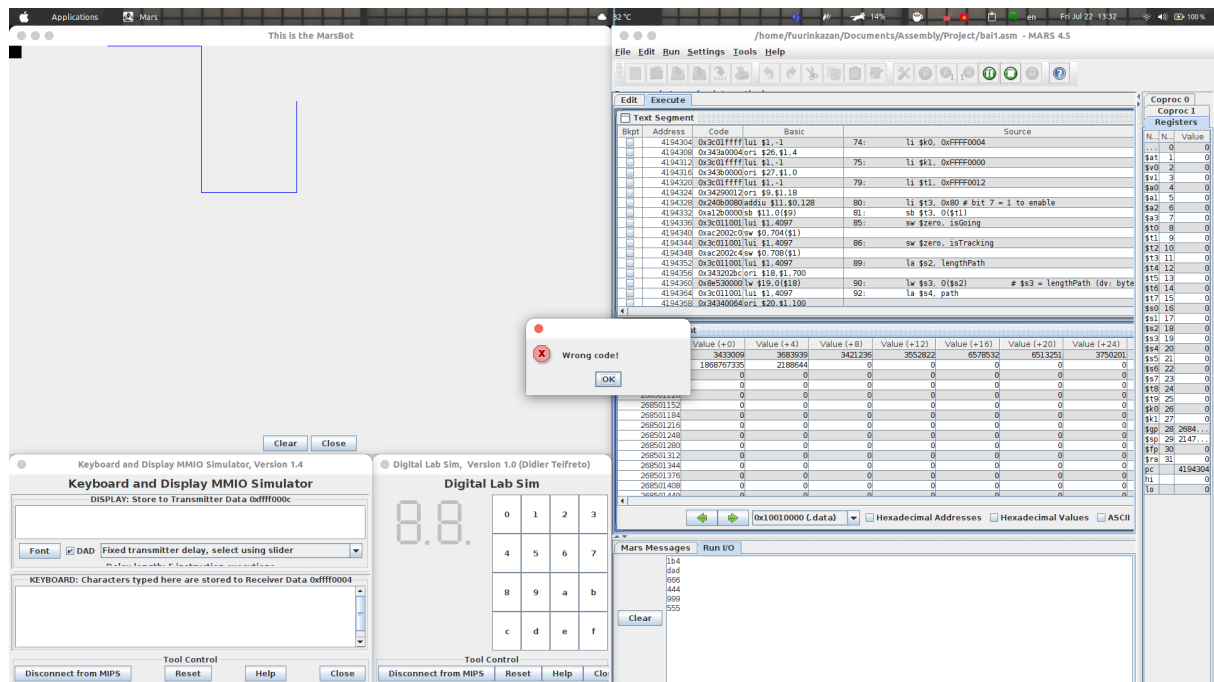
Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)
26950092	3433009	368399	3421296	355282	6578532	6513251	3750201
26950104	1868787395	2188644	0	0	0	0	0
269501056	0	0	0	0	0	0	0
269501088	0	0	0	0	0	0	0
269501120	0	0	0	0	0	0	0
269501152	0	0	0	0	0	0	0
269501184	0	0	0	0	0	0	0
269501216	0	0	0	0	0	0	0
269501248	0	0	0	0	0	0	0
269501280	0	0	0	0	0	0	0
269501312	0	0	0	0	0	0	0
269501344	0	0	0	0	0	0	0
269501376	0	0	0	0	0	0	0
269501408	0	0	0	0	0	0	0
269501440	0	0	0	0	0	0	0

Mars Messages Run I/O

134
dad
666
444

Clear



This is the MarsBot

Clear

Close

Keyboard and Display MMIO Simulator, Version 1.4

DISPLAY: Store to Transmitter Data 0xffff000c

Font:

Fixed transmitter delay, select using slider

KEYBOARD: Characters typed here are stored to Receiver Data 0xffff0004

Disconnect from MIPS

Reset

Help

Close

Digital Lab Sim, Version 1.0 (Didier Teifreto)

8.8.

0

1

2

3

4

5

6

7

8

9

a

b

c

d

e

f

Disconnect from MIPS

Reset

Help

Close

/home/fuurinkazan/Documents/Assembly/Project/bat1.asm - MARS 4.5

File

Edit

Run

Settings

Tools

Help

Text Segment

Blkpt

Address

Code

Basic

Source

4194304

0x3c01ffff

li \$1, -1

74:

li \$k0, 0xffff0004

Source

4194308

0x343e0004

ori \$26, \$1, 4

75:

li \$k1, 0xffff0000

4194312

0x3c01ffff

li \$1, -1

4194316

0x343e0000

ori \$27, \$1, 0

4194320

0x3c01ffff

li \$1, -1

79:

li \$t1, 0xffff0012

4194324

0x343e0001

ori \$9, \$1, 18

4194328

0x249b0000

addiu \$t1, \$0, 128

80:

li \$t3, 0x80 # bit 7 = 1 to enable

4194332

0xa12b0000

ori \$t1, 0, 169

81:

or \$t3, 0, (\$t1)

4194336

0x3c011000

li \$1, 4097

85:

sv \$zero, isGoing

4194340

0xac2002c0

sw \$0, 704(\$t1)

4194344

0x3c011000

li \$1, 4097

4194348

0xac2002c0

sw \$0, 708(\$t1)

4194352

0x3c011000

li \$1, 4097

4194356

0x343e000c

ori \$t8, \$1, 700

4194360

0x9e530000

lw \$t3, 0(\$t2) # \$t3 = lengthPath (dv: byte

4194364

0x3c011000

li \$1, 4097

4194368

0x34340004

ori \$20, \$1, 100

Data Segment

Address

Value (+0)

Value (+4)

Value (+8)

Value (+12)

Value (+16)

Value (+20)

Value (+24)

26950092

343309

368339

342136

362892

607632

601320

379000

269501024

1868767395

2188644

0

0

0

0

0

269501056

0

0

0

0

0

0

0

269501088

0

0

0

0

0

0

0

269501120

0

0

0

0

0

0

0

269501152

0

0

0

0

0

0

0

269501184

0

0

0

0

0

0

0

269501216

0

0

0

0

0

0

0

269501248

0

0

0

0

0

0

0

269501280

0

0

0

0

0

0

0

269501312

0

0

0

0

0

0

0

269501344

0

0

0

0

0

0

0

269501376

0

0

0

0

0

0

0

269501408

0

0

0

0

0

0

0

269501440

0

0

0

0

0

0

0

Mars Messages

Run I/O

Clear

1b4

9ad

666

444

999

555

1b4

666

c08

Coproc 0

Coproc 1

Registers

N...

Value

...

0

0

\$at

1

0

\$v0

2

0

\$v1

3

0

\$a0

4

0

\$a1

5

0

\$a2

6

0

\$a3

7

0

\$t0

8

0

\$t1

9

0

\$t2

10

0

\$t3

11

0

\$t4

12

0

\$t5

13

0

\$t6

14

0

\$t7

15

0

\$a0

16

0

\$t1

17

0

\$t2

18

0

\$t3

19

0

\$t4

20

0

\$t5

21

0

\$t6

22

0

\$t7

23

0

\$t8

24

0

\$t9

25

0

\$t0

26

0

\$t1

27

0

\$t2

28

2684...

\$t3

29

2147...

\$t4

30

0

\$t5

31

0

\$t6

32

4194364

\$t7

33

0