

Báo cáo tuần 10 (P2)

Thực hành kiến trúc máy tính

Họ tên: Phan Minh Anh Tuấn
MSSV: 20205227

Mục lục

1	Assignment 3	2
1.1	Phân tích đề bài	2
1.2	Triển khai MIPS	2
1.2.1	GO	2
2	Assignment 4	6
2.1	Phân tích đề bài	6
2.2	Triển khai MIPS	6

1 Assignment 3

1.1 Phân tích đề bài

Đề bài: Create a new project, type in, and build the program of Home Assignment 3. Make the Bot run and draw a triangle by tracking
MarsBots là robot ảo di chuyển trong không gian hai chiều. Để điều khiển, có 5 thủ tục bao gồm: GO, STOP, TRACK, UNTRACK, ROTATE

1.2 Triển khai MIPS

1.2.1 GO

```
GO:      li $at, MOVING # change MOVING port
         addi $k0, $zero, 1 # to logic 1,
         sb $k0, 0($at) # to start running
         nop
         jr $ra
         nop
```

Hình 1: Hàm GO

Trong đó:

- Tác dụng: điều khiển Marsbots bắt đầu di chuyển
- Cách thực hiện: đổi giá trị logic của MOVING thành 1

```
STOP:    li $at, MOVING # change MOVING port to 0
         sb $zero, 0($at) # to stop
         nop
         jr $ra
         nop
```

Hình 2: Hàm STOP

Trong đó:

- Tác dụng: điều khiển Marsbots dừng lại
- Cách thực hiện: đổi giá trị logic của MOVING thành 0

```
TRACK:  li $at, LEAVETRACK # change LEAVETRACK port
        addi $k0, $zero, 1 # to logic 1,
        sb $k0, 0($at) # to start tracking
        nop
        jr $ra
        nop
```

Hình 3: Hàm TRACK

Trong đó:

- Tác dụng: điều khiển Marsbots vẽ đường thẳng
- Cách thực hiện: đổi giá trị logic của LEAVETRACK thành 1

```
UNTRACK:  li $at, LEAVETRACK # change LEAVETRACK port to 0
        sb $zero, 0($at) # to stop drawing tail
        nop
        jr $ra
        nop
```

Hình 4: Hàm UNTRACK

Trong đó:

- Tác dụng: điều khiển Marsbots dừng đường thẳng
- Cách thực hiện: đổi giá trị logic của LEAVETRACK thành 0

```
ROTATE: li $at, HEADING # change HEADING port
        sw $a0, 0($at) # to rotate robot
        nop
        jr $ra
        nop
```

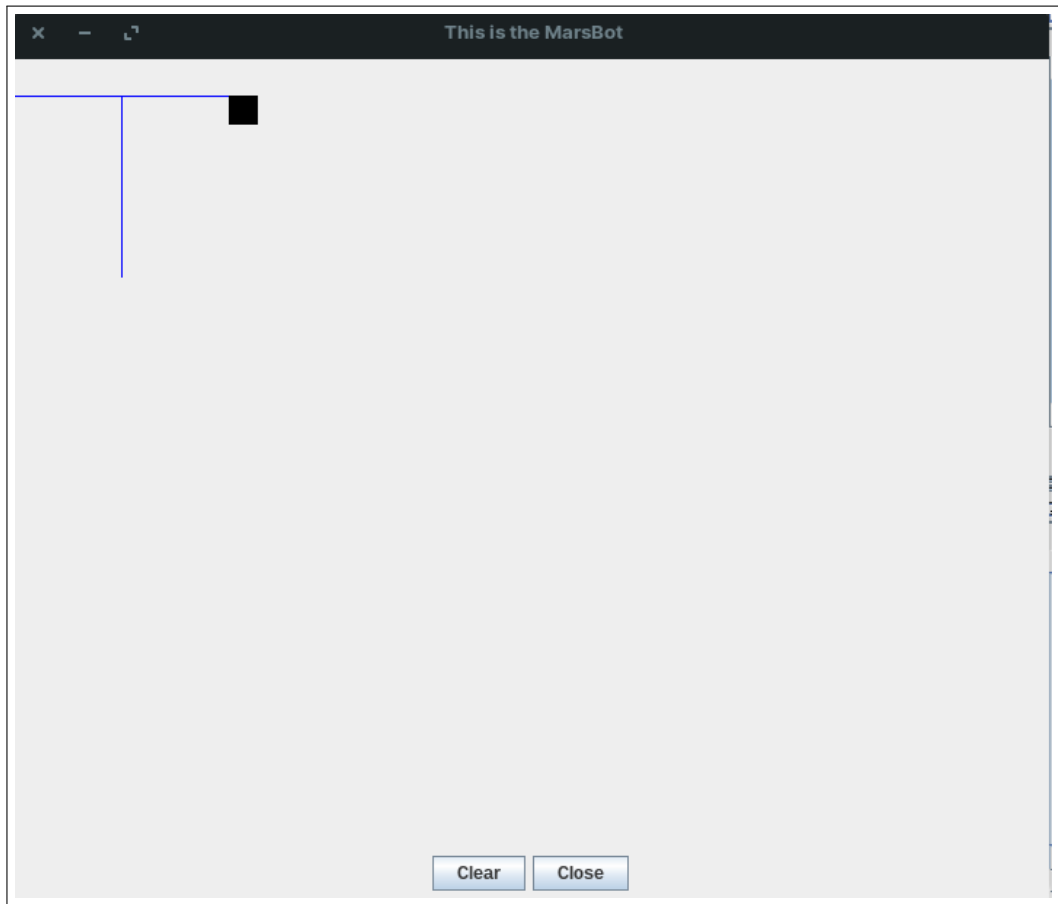
Hình 5: Hàm ROTATE

Trong đó:

- Tác dụng: điều khiển Marsbots quay từ 0 - 359
- Cách thực hiện: đổi giá trị logic của HEADING thành góc muốn quay.
Một số hướng đi phổ biến như:
 - Đi lên: 0 độ
 - Sang phải: 90 độ
 - Đi xuống: 180 độ
 - Sang trái: 270 độ

Yêu cầu: Vẽ chữ cái đầu tiên của tên: chữ T

- Cho bots đi xuống dưới (gán góc là 180 độ cho rotate)
- Cho bots đi sang trái (gán góc là 90 độ cho rotate)
- Cho bots đi xuống dưới (gán góc là 180 độ cho rotate)
- Đi lên, sau đó đi sang trái



Hình 6: Kết quả

2 Assignment 4

2.1 Phân tích đề bài

Đề bài: Create a new project, type in, and build the program of Home Assignment 4. Read key char and terminate the application when receiving “exit” command.

2.2 Triển khai MIPS

```
.eqv KEY_CODE    0xFFFF0004    # ASCII code from keyboard, 1 byte
.eqv KEY_READY   0xFFFF0000    # =1 if has a new keycode ?
                                # Auto clear after lw
.eqv DISPLAY_CODE 0xFFFF000C    # ASCII code to show, 1 byte
.eqv DISPLAY_READY 0xFFFF0008   # =1 if the display has already to do
                                # Auto clear after sw

.text
        li $k0, KEY_CODE
        li $k1, KEY_READY
        li $s0, DISPLAY_CODE
        li $s1, DISPLAY_READY

loop:
WaitForKey: lw $t1, 0($k1)          # $t1 = [$k1] = KEY_READY
            nop
            beq $t1, $zero, WaitForKey # if $t1 == 0 then Polling
            nop
            #-----
ReadKey:   lw $t0, 0($k0)          # $t0 = [$k0] = KEY_CODE
            nop
            #-----
WaitForDis: lw $t2, 0($s1)         # $t2 = [$s1] = DISPLAY_READY
            nop
            beq $t2, $zero, WaitForKey # if $t2 == 0 then Polling
            nop
            #-----
Encrypt:  addi $t0, $t0, 7 # change input key
            #-----
ShowKey:  sw $t0, 0($s0) # show key
            nop
            beq $t0, '{', quit
            #-----
            j loop
quit:     nop
```

Hình 7: Code bài tập 4

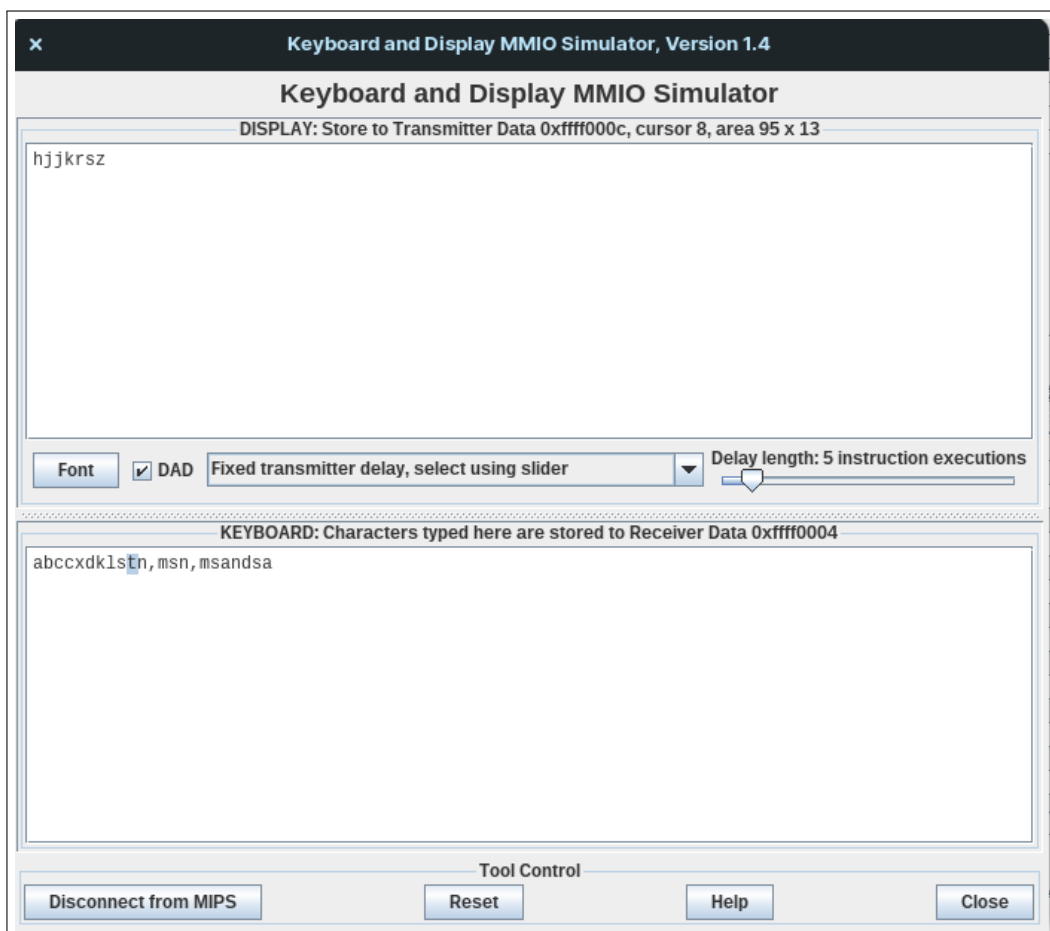
Chương trình cho phép người dùng nhập vào một ký tự sau đó mã hóa nó thành một ký tự khác và hiển thị ra màn hình. Chương trình dừng lại khi nhận được ký tự "t" (Do tên là Tuấn)

```

Encrypt:      addi $t0, $t0, 7 # change input key
              #-
ShowKey:      sw $t0, 0($s0) # show key
              nop
              beq $t0, '{', quit
              #-
              j loop
              nop

```

Khoảng cách mã hóa là 7 (do MSSV là 20205227), kí tự kết thúc là "t" tương ứng với "{" khi mã hóa



Hình 8: Kết quả

Nhận xét: Sau khi nhập chữ t, chương trình dừng lại không mã hóa nữa. Các kí tự được mã hóa cách 7 đơn vị trong bảng ASCII.