

Báo cáo thực hành số 10

Nguyễn Thị Minh Châu - 20214997

1. Bài 1

- Mã nguồn:

```
1 .eqv SEVENSEG_LEFT 0xFFFF0011 # Địa chỉ của đèn led 7 đoạn trái. # Bit 0 = đoạn a;  
2  
3 .eqv SEVENSEG_RIGHT 0xFFFF0010 # Địa chỉ của đèn led 7 đoạn phải .text  
4 main:  
5 li $a0, 0x6F # set value for segments jal SHOW_7SEG_LEFT # show  
6 nop  
7 li $a0, 0x7 # set value for segments jal SHOW_7SEG_RIGHT # show  
8 nop  
9 exit: li $v0, 10  
10 syscall  
11 endmain: #----- # Function  
12 # param[in] $a0 value to shown  
13 # remark $t0 changed #-----  
14 sb $a0, 0($t0) # assign new value  
15 nop  
16 jr $ra  
17 nop #-----  
18 # Function SHOW_7SEG_RIGHT : turn on/off the 7seg  
19 # param[in] $a0 value to shown  
20 # remark $t0 changed #-----  
21 nop  
22 jr $ra  
23 nop  
~
```

- Giải thích:

SEVENSEG_LEFT và SEVENSEG_RIGHT đều đã có địa chỉ đặt sẵn tùy theo nhà sản xuất chip

SHOW_7SEG_LEFT và SHOW_7SEG_RIGHT là các hàm hiển thị số ra theo hiển thị 7 thanh.

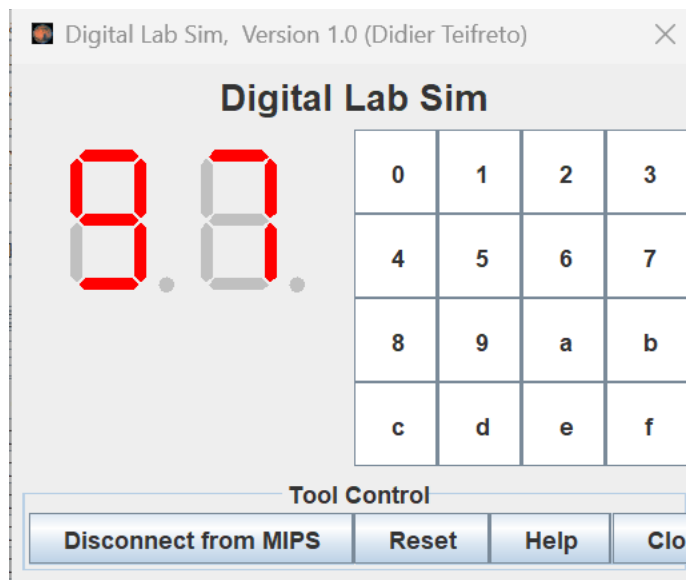
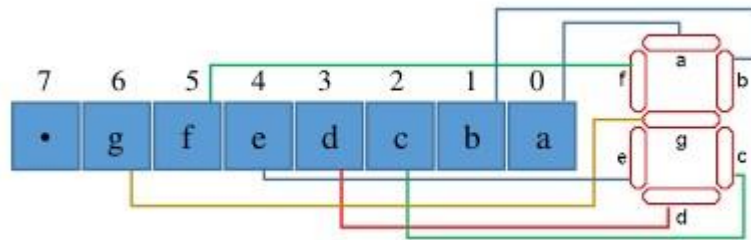
\$a0 là mã nhị phân của số mình muốn hiển thị 7 thanh

Để hiển thị số 9 thì thanh e không được sáng

.	g	f	e	d	c	b	a	
0	1	1	0	1	1	1	1	-> 0x6F

Để hiển thị số 7 thì thanh f, g, e, d không được sáng

.	g	f	e	d	c	b	a	
0	0	0	0	0	1	1	1	-> 0x7

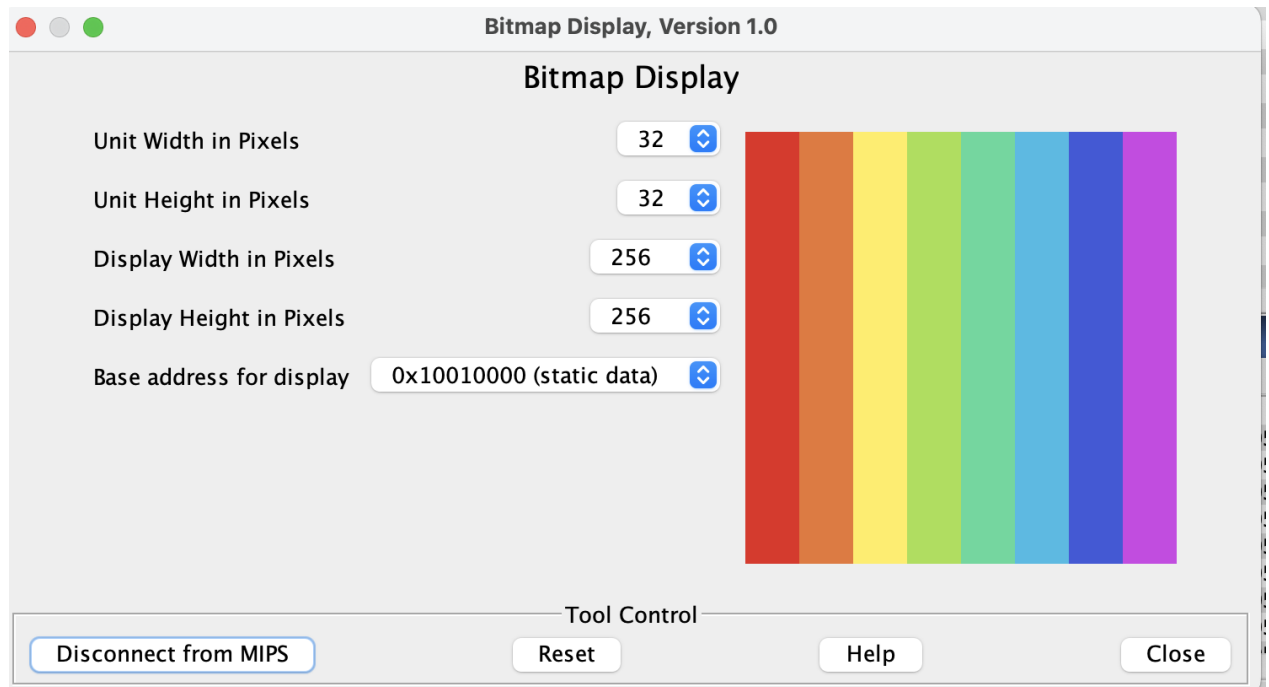


2. Bài 2

- Mã nguồn:

```
1  .eqv MONITOR_SCREEN 0x10010000
2  .eqv RED 0x00E6261F
3  .eqv ORANGE 0x00EB7532
4  .eqv YELLOW 0x00FFEE58
5  .eqv GREEN 0x00A3E048
6  .eqv MINT 0x0049DA9A
7  .eqv BLUE 0x0034BBE6
8  .eqv INDIGO 0x004355DB
9  .eqv VIOLET 0x00d23BE7
10 .eqv WHITE 0x00FFFFFF
11 .text
12 li $k0, MONITOR_SCREEN
13 addi $k1, $k0, 4096
14 li $s0, 0
15 paint:
16 li $t0, RED
17 sw $t0, 0($k0)
18 nop
19 #Nap dia chi bat dau cua man hinh #Nap dia chi ket thuc cua man hinh
20 li $t0, ORANGE
21 sw $t0, 4($k0)
22 nop
23 li $t0, YELLOW
24 sw $t0, 8($k0)
25 nop
26 li $t0, GREEN
27 sw $t0, 12($k0)
28 nop
29 li $t0, MINT
30 sw $t0, 16($k0)
31 nop
32 li $t0, BLUE
33 sw $t0, 20($k0)
34 nop
35 li $t0, INDIGO
36 sw $t0, 24($k0)
37 nop
38
39 li $t0, VIOLET
40 sw $t0, 28($k0)
41 nop
42 addi $k0, $k0, 32
43 blt $k0, $k1, paint
```

- Kết quả chạy:



- Giải thích:

Hiển thị theo pixel trên màn hình 256x256 và mỗi pixel là 32x32 nên ta sẽ có 8 hàng và 8 cột

Chạy vòng lặp và cho vào từng giá trị tại địa chỉ của từng unit một giá trị mã màu.

3. Bài 3

- Mã nguồn:

```

1  .eqv HEADING 0xffff8010
2  .eqv MOVING 0xffff8050
3  .eqv LEAVETRACK 0xffff8020
4  .eqv WHEREX 0xffff8030
5  .eqv WHEREY 0xffff8040
6  .text
7  main:
8      jal TRACK
9      nop
10     addi $a0, $zero, 90
11     jal ROTATE
12     nop
13     jal GO
14     nop
15  sleep1:
16     addi $v0,$zero,32
17     li $a0,3000
18     syscall
19     jal UNTRACK
20     nop
21     jal TRACK
22     nop
23  goDOWN:
24     addi $a0, $zero, 180
25     jal ROTATE
26     nop
27  sleep2:
28     addi $v0,$zero,32

```

```

28     addi $v0,$zero,32
29     li $a0,3000
30     syscall
31     jal UNTRACK
32     nop
33     jal TRACK
34     nop
35  goLEFT:
36     addi $a0, $zero, 315
37     jal ROTATE
38     nop
39
40  sleep3:
41     addi $v0,$zero,32
42     li $a0,4000
43     syscall
44     jal UNTRACK
45     nop
46
47
48
49  end_main:
50  GO: li $at, MOVING
51     addi $k0, $zero,1
52     sb $k0, 0($at)
53     nop
54     jr $ra
55     nop

```

```

52  sb $k0, 0($at)
53  nop
54  jr $ra
55  nop
56
57  STOP: li $at, MOVING
58  sb $zero, 0($at)
59  nop
60  jr $ra
61  nop
62
63  TRACK: li $at, LEAVETRACK
64  addi $k0, $zero, 1
65  sb $k0, 0($at)
66  nop
67  jr $ra
68  nop
69  UNTRACK: li $at, LEAVETRACK
70  sb $zero, 0($at)
71  nop
72  jr $ra
73  nop
74  ROTATE: li $at, HEADING
75  sw $a0, 0($at)
76  nop
77  jr $ra
78  nop
79

```

- Kết quả



- Giải thích:

Hàm ROTATE-> chỉnh góc của con trỏ để đi tiếp

Hàm STOP -> set giá trị của MOVING về 0 để dừng con trỏ

Hàm GO -> set MOVING về 1 và con trỏ sẽ di chuyển theo góc được chỉ định qua hàm ROTATE

Hàm TRACK -> set LEAVETRACK về 1 và bắt đầu vẽ

Hàm UNTRACK -> ngược lại với TRACK, set LEAVETRACK về 0 và dừng vẽ

Các nhãn loop để set thời gian con trỏ di chuyển theo hướng vừa được set up bằng hàm ROTATE để vẽ các đoạn theo ý muốn

4. Bài 4

- Mã nguồn

```

1  .eqv KEY_CODE 0xFFFF0004 # ASCII code from keyboard, 1 byte
2  .eqv KEY_READY 0xFFFF0000 # =1 if has a new keycode ?
3  # Auto clear after lw
4  .eqv DISPLAY_CODE 0xFFFF000C # ASCII code to show, 1 byte
5  .eqv DISPLAY_READY 0xFFFF0008 # =1 if the display has already to do #
6  .data
7  key:      .space 5
8  string:   .space 10
9  .text
10         li $k0, KEY_CODE
11         li $k1, KEY_READY
12         li $s0, DISPLAY_CODE
13         li $s1, DISPLAY_READY
14         la $a0, string
15 main:
16 char1:
17         jal      loop
18         beq      $t0, 101, char2
19         j        char1
20 char2:
21         jal      loop
22         beq      $t0, 120, char3
23         beq      $t0, 101, char2
24         j        char1
25 char3:
26         jal      loop
27         beq      $t0, 105, char4
28         beq      $t0, 101, char2

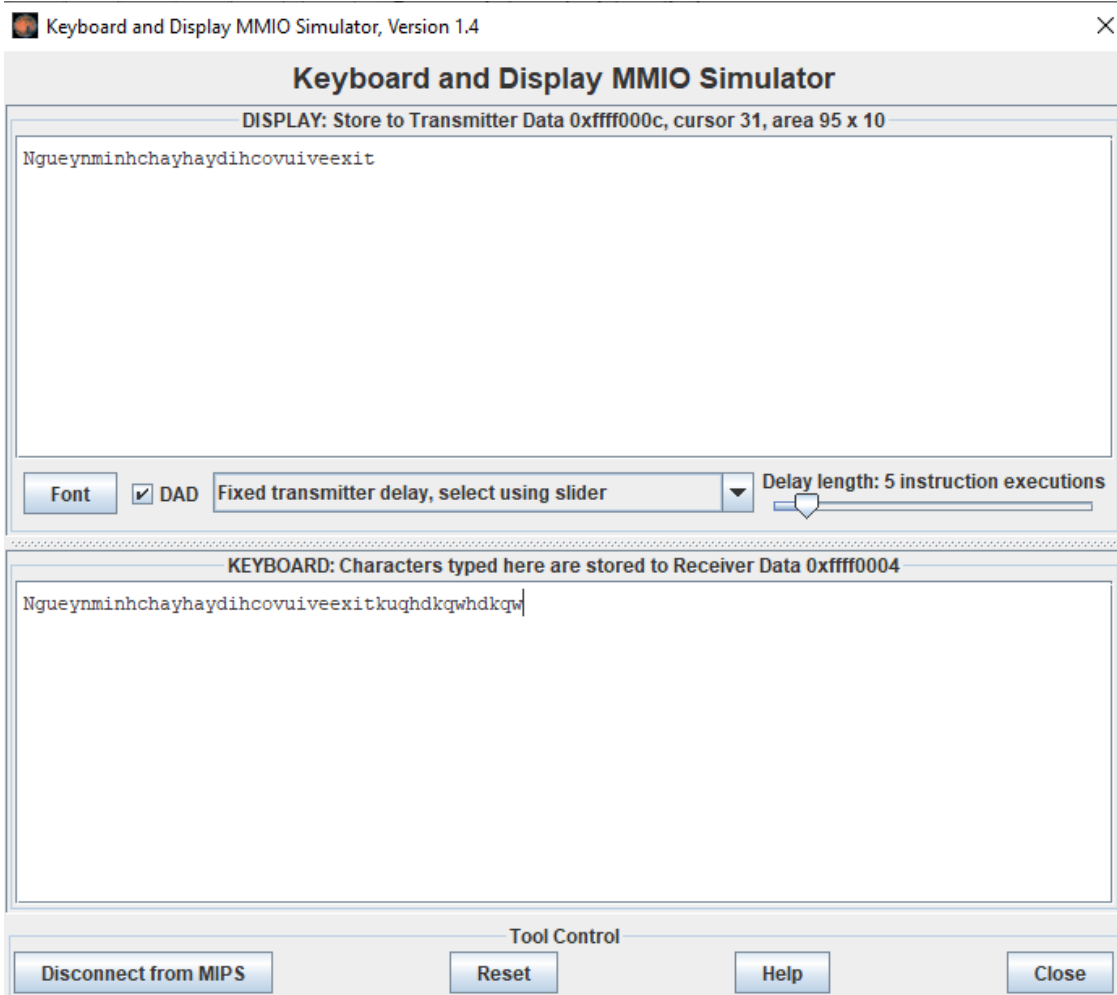
```



```

28     beq     $t0,101,char2
29     j       char1
30 char4:
31     jal     loop
32     beq     $t0,116,end_main
33     beq     $t0,101,char2
34     j       char1
35 end_main:
36 exit:
37     li      $v0,10
38     syscall
39 loop:
40     nop
41 WaitForKey:
42     lw      $t1, 0($k1) # $t1 = [$k1] = KEY_READY nop
43     beq     $t1, $zero, WaitForKey # if $t1 == 0 then Polling nop
44     #-----
45 ReadKey:
46     lw      $t0, 0($k0) # $t0 = [$k0] = KEY_CODE nop
47     #-----
48 WaitForDis:
49     lw      $t2, 0($s1) # $t2 = [$s1] = DISPLAY_READY nop
50     beq     $t2, $zero, WaitForDis # if $t2 == 0 then Polling nop
51     #----- #-----
52     sw      $t0, 0($s0)          # show key
53     nop
54     #-----
55     jr      $ra

```



- Giải thích:

Chương trình có 1 vòng loop nhận các kí tự đọc vào và hiển thị