

Week 10

Assignment 1:

Create a new project, type in, and build the program of Home Assignment 1. Show different values on LED

Code

```

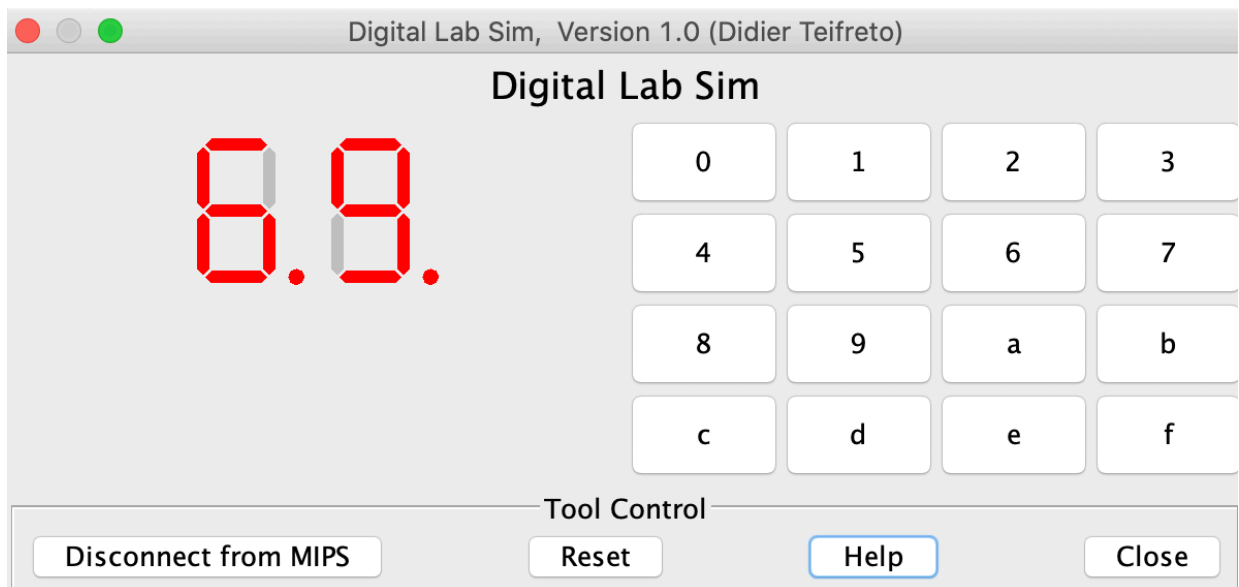
.eqv SEVENSEG_RIGHT 0xFFFF0010 # Dia chi cua den led 7 doan trai.
    # Bit0=doana;
    # Bit1=doanb;...
    # Bit7=dau.
    # Dia chi cua den led 7 doan phai
.eqv SEVENSEG_LEFT 0xFFFF0011
.text
main:
    li    $a0, 0xFD    # set value for segments
    jal   SHOW_7SEG_LEFT    # show
    li    $a0, 0xEF    # set value for segments
    jal   SHOW_7SEG_RIGHT   # show
exit:
    li    $v0, 10
    syscall
endmain:
#-----
# Function SHOW_7SEG_LEFT : turn on/off the 7seg
# param[in] $a0 value to shown
# remark $t0 changed
#-----
SHOW_7SEG_LEFT:
    li $t0, SEVENSEG_LEFT    # assign port's address
    sb  $a0, 0($t0)          # assign new value
    jr  $ra
#-----
# Function SHOW_7SEG_RIGHT : turn on/off the 7seg
# param[in] $a0 value to shown
# remark $t0 changed #-----
#-----
SHOW_7SEG_RIGHT:
    li $t0, SEVENSEG_RIGHT # assign port's address
    sb $a0, 0($t0)          # assign new value
    jr $ra

```

Nhận xét

- Ta cần hiển thị 2 chữ số cuối của mã sinh viên (20176869)
- Để hiển thị số 6 thì các thanh a, c, d, e, f, g và . của SEVENSEG_LEFT cần phải set bằng 1, khi đó mã hexa là: 0xFD
- Để hiển thị số 9 thì các thanh a, b, c, d, f, g và . của SEVENSEG_RIGHT cần phải set bằng 1, khi đó mã hexa là: 0xEF

Kết quả



Assignment 2:

Create a new project, type in, and build the program of Home Assignment 2. Draw something.

code

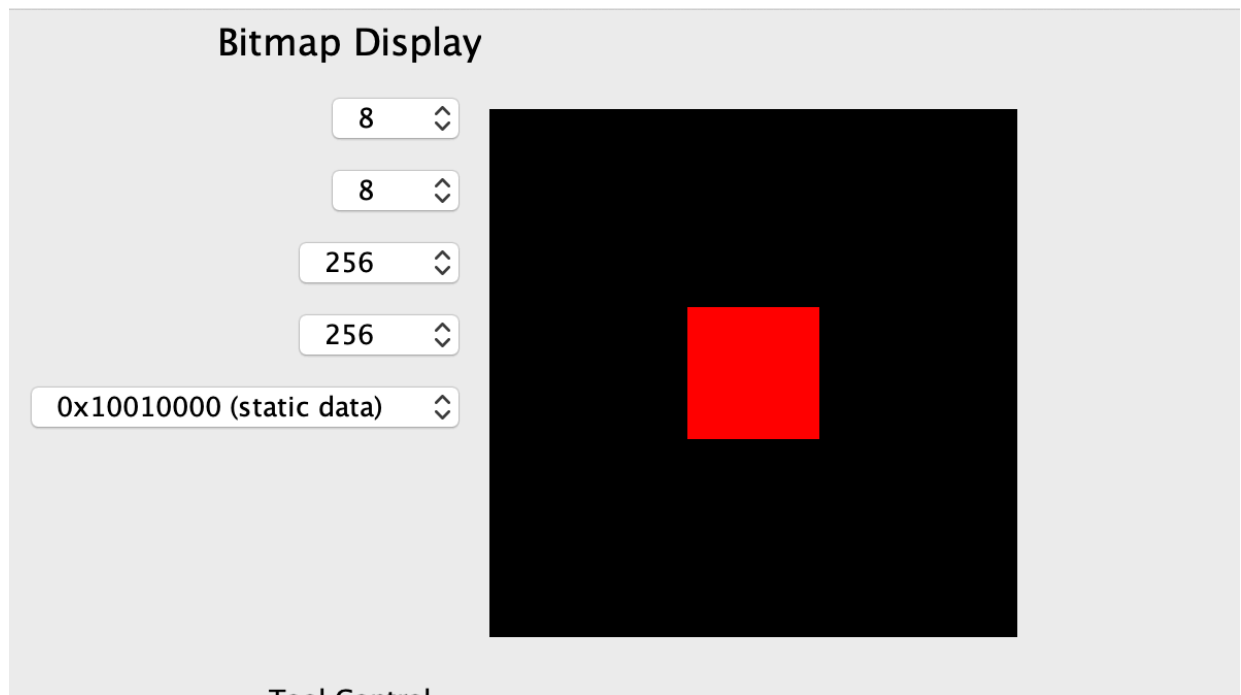
```

.eqv MONITOR_SCREEN    0x10010000
.eqv RED                0x00FF0000
.eqv GREEN             0x0000FF00
.eqv BLUE              0x000000FF
.eqv WHITE             0x00FFFFFF
.eqv YELLOW            0x00FFFF00
.text
# -----
# $k0 = $k0 + 32*12*4
# for( y = 12; y < 20; y++)
#   for( x = 0; x < 32; x++)
#       if(11 < x < 20) draw red
#       $k0 += 4
# -----
    li $k0, MONITOR_SCREEN

    addi $t0, $zero, 12 # $t0 = y = 12
    sll $t2, $t0, 2
    addi $k0, $k0, 1536 # $k0 = $k0 + 32*12*4
loop1:
    addi $t1, $zero, 0  # $t1 = x = 0
    slti $t2, $t0, 20
    beqz $t2, end_loop1
loop2:
    slti $t2, $t1, 32
    beqz $t2, end_loop2
if1:
    sgt $t2, $t1, 11
    beqz $t2, end
if2:
    slti $t2, $t1, 20
    beqz $t2, end
    li $t3, RED
    sw  $t3, 0($k0)
end:
    addi $t1, $t1, 1
    addi $k0, $k0, 4
    j loop2
end_loop2:
    addi $t0, $t0, 1
    j loop1
end_loop1:

```

Kết quả



Assignment 3:

Create a new project, type in, and build the program of Home Assignment 3. Make the Bot run and draw a triangle by tracking `Code`

```
.eqv    HEADING    0xffff8010  # Integer: An angle between 0 and 359
                                     # 0 : North (up)
                                     # 90: East (right)
                                     # 180: South (down)
                                     # 270: West (left)

.eqv    MOVING     0xffff8050  # Boolean: whether or not to move
.eqv    LEAVETRACK 0xffff8020  # Boolean (0 or non-0):
                                     # whether or not to leave a track

.eqv    WHEREX     0xffff8030  # Integer: Current x-location of MarsBot
.eqv    WHEREY     0xffff8040  # Integer: Current y-location of MarsBot

.text
main:
    addi $a0, $zero, 135        # Marsbot rotates 135° and start running
    jal ROTATE
    jal GO

sleep0: addi $v0,$zero,32        # Keep running by sleeping in 12000 ms
    li $a0,14000
    syscall
```

```

        jal STOP

goE1:   addi $a0, $zero, 90          # Marsbot rotates 90*
        jal ROTATE
        jal TRACK                   # draw first edge
        jal GO

sleep1: addi $v0,$zero,32           # Keep running by sleeping in 5000 ms
        li $a0,6000
        syscall

        jal STOP
        jal UNTRACK                # keep old track
goE2:   addi $a0, $zero, 210        # Marsbot rotates 210*
        jal ROTATE
        jal TRACK                   # draw second edge
        jal GO

sleep2: addi $v0,$zero,32           # Keep running by sleeping in 6000 ms
        li $a0,6000
        syscall

        jal STOP                  # keep old track
        jal UNTRACK
goE3:   addi $a0, $zero, 330        # Marsbot rotates 330*
        jal ROTATE
        jal TRACK                   # draw third edge
        jal GO

sleep3: addi $v0,$zero,32           # Keep running by sleeping in 6000 ms
        li $a0,6000
        syscall

        jal STOP                  # keep old track
        jal UNTRACK
end_main:
        li $v0, 10
        syscall

#-----
# GO procedure, to start running
# param[in] none
#-----
GO:     li $at, MOVING              # change MOVING port
        addi $k0, $zero,1          # to logic 1,
        sb $k0, 0($at)             # to start running
        jr $ra

#-----

```

```

# STOP procedure, to stop running
# param[in] none
#-----
STOP:  li $at, MOVING      # change MOVING port to 0
      sb $zero, 0($at)    # to stop
      jr $ra

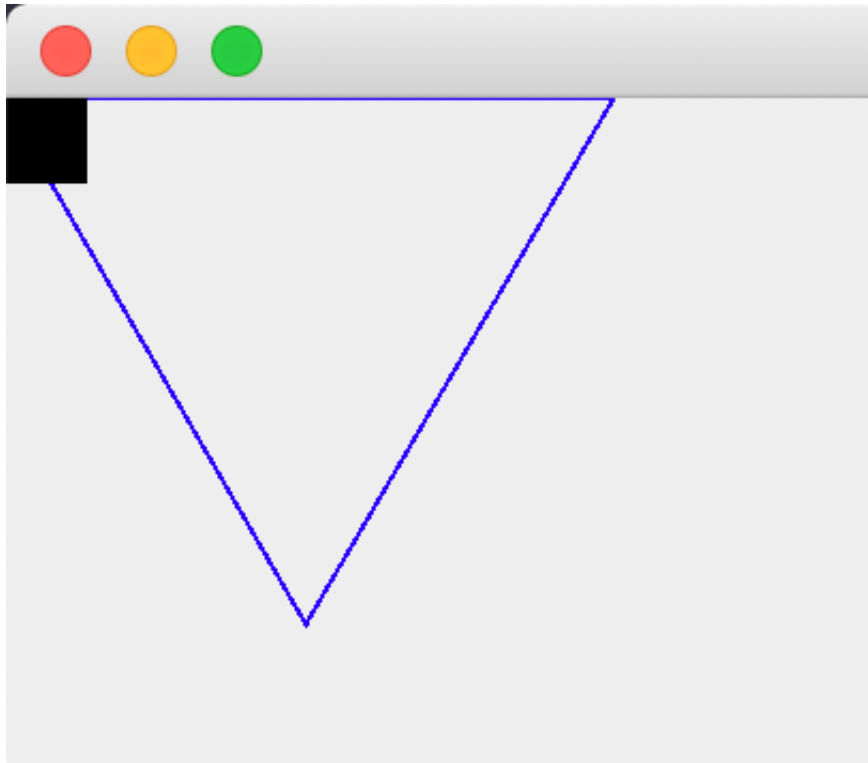
#-----
# TRACK procedure, to start drawing line
# param[in] none
#-----
TRACK: li $at, LEAVETRACK  # change LEAVETRACK port
      addi $k0, $zero, 1   # to logic 1,
      sb $k0, 0($at)      # to start tracking
      jr $ra

#-----
# UNTRACK procedure, to stop drawing line
# param[in] none
#-----
UNTRACK:li $at, LEAVETRACK # change LEAVETRACK port to 0
      sb $zero, 0($at)    # to stop drawing tail
      jr $ra

#-----
# ROTATE procedure, to rotate the robot
# param[in] $a0, An angle between 0 and 359
# 0 : North (up)
# 90: East (right)
# 180: South (down)
# 270: West (left)
#-----
ROTATE: li $at, HEADING    # change HEADING port
      sw $a0, 0($at)      # to rotate robot
      jr $ra

```

Kết quả



Assignment 4:

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.

Code

```
.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:  nop

WaitForKey:
    lw $t1, 0($k1)           # $t1 = [$k1] = KEY_READY
    beq $t1, $zero, WaitForKey # if $t1 == 0 then Polling
ReadKey:
    lw $t0, 0($k0)           # $t0 = [$k0] = KEY_CODE
WaitForDis:
    lw $t2, 0($s1)           # $t2 = [$s1] = DISPLAY_READY
    beq $t2, $zero, WaitForDis # if $t2 == 0 then Polling
```



```

    beq $t0, 'e', check_e
    beq $t0, 'x', check_x
    beq $t0, 'i', check_i
    beq $t0, 't', check_t
    j    skip_checking
Encrypt:
    addi $t0, $t0, 1          # change input key
ShowKey:
    sw $t0, 0($s0)           # show key
    nop
    j loop

check_e:
    bne $s2, $zero, skip_checking
    addi $s2, $zero, 1        #wait for x
    j    Encrypt

check_x:
    addi $t3, $zero, 1
    bne $s2, $t3, skip_checking
    addi $s2, $zero, 2        #wait fro i
    j    Encrypt

check_i:
    addi $t3, $zero, 2
    bne $s2, $t3, skip_checking
    addi $s2, $zero, 3        #wait fro t
    j    Encrypt

check_t:
    addi $t3, $zero, 3
    bne $s2, $t3, skip_checking
    li $v0, 10                #terminate the program if get 'exit'
    syscall

skip_checking:
    addi $s2, $zero, 0
    j    Encrypt
    nop

```

Kết quả

Keyboard and Display MMIO Simulator, Version 1.4

Keyboard and Display MMIO Simulator

DISPLAY: Store to Transmitter Data 0xffff000c, cursor 11, area 113 x 12

ifmmp!uiboh

Font ☒ DAD Fixed transmitter delay, select using slider Delay length: 5 instruction executions

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

hello thang

Tool Control

Disconnect from MIPS Reset Help Close

-- program is finished running --

Clear

Bkpt	Address	Code	Basic
<input type="checkbox"/>	0x00400000	0x3c01ffff lui \$1,0	
<input type="checkbox"/>	0x00400004	0x343a0004 ori \$26,	
<input type="checkbox"/>	0x00400008	0x3c01ffff lui \$1,0	
<input type="checkbox"/>	0x0040000c	0x343b0000 ori \$27,	
<input type="checkbox"/>	0x00400010	0x3c01ffff lui \$1,0	
<input type="checkbox"/>	0x00400014	0x3430000c ori \$16,	
<input type="checkbox"/>	0x00400018	0x3c01ffff lui \$1,0	
<input type="checkbox"/>	0x0040001c	0x34310008 ori \$17,	
<input type="checkbox"/>	0x00400020	0x00000000 nop	
<input type="checkbox"/>	0x00400024	0x8f690000 lw \$9,0x	
<input type="checkbox"/>	0x00400028	0x1120ffff beq \$9,\$	
<input type="checkbox"/>	0x0040002c	0x8f480000 lw \$8,0x	

Address	Value (+0)	Value (-)
0x10010000	0x00000000	
0x10010020	0x00000000	
0x10010040	0x00000000	
0x10010060	0x00000000	
0x10010080	0x00000000	
0x100100a0	0x00000000	
0x100100c0	0x00000000	
0x100100e0	0x00000000	
0x10010100	0x00000000	

Keyboard and Display MMIO Simulator, Version 1.4

Keyboard and Display MMIO Simulator

DISPLAY: Store to Transmitter Data 0xffff000c, cursor 15, area 113 x 12

ifmmp!uiboh fyj

Font ☒ DAD Fixed transmitter delay, select using slider Delay length: 5 instruction executions

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

hello thang
exit|

Tool Control

Disconnect from MIPS Reset Help Close

-- program is finished running --

Clear

-- program is finished running --

Bkpt	Address	Code	Basic
<input type="checkbox"/>	0x00400000	0x3c01ffff lui \$1,0	
<input type="checkbox"/>	0x00400004	0x343a0004 ori \$26,	
<input type="checkbox"/>	0x00400008	0x3c01ffff lui \$1,0	
<input type="checkbox"/>	0x0040000c	0x343b0000 ori \$27,	
<input type="checkbox"/>	0x00400010	0x3c01ffff lui \$1,0	
<input type="checkbox"/>	0x00400014	0x3430000c ori \$16,	
<input type="checkbox"/>	0x00400018	0x3c01ffff lui \$1,0	
<input type="checkbox"/>	0x0040001c	0x34310008 ori \$17,	
<input type="checkbox"/>	0x00400020	0x00000000 nop	
<input type="checkbox"/>	0x00400024	0x8f690000 lw \$9,0x	
<input type="checkbox"/>	0x00400028	0x1120ffff beq \$9,\$	
<input type="checkbox"/>	0x0040002c	0x8f480000 lw \$8,0x	

Address	Value (+0)	Value (-)
0x10010000	0x00000000	
0x10010020	0x00000000	
0x10010040	0x00000000	
0x10010060	0x00000000	
0x10010080	0x00000000	
0x100100a0	0x00000000	
0x100100c0	0x00000000	
0x100100e0	0x00000000	
0x10010100	0x00000000	

Registers	Coproc 1
Number	Value
addr)	8
status)	12
cause)	13
epc)	14