

Bài 1

```
.eqv IN_ADRESS_HEXА_KEYBOARD    0xFFFF0012
.eqv OUT_ADRESS_HEXА_KEYBOARD    0xFFFF0014

.data
    endlіne: .asciiz "\n"

.text
main:
    li $t1, IN_ADRESS_HEXА_KEYBOARD
    li $t2, OUT_ADRESS_HEXА_KEYBOARD
    li $t3, 0x01 # check row 4 with key C, D,E, F
    li $t4, 0x02 # check row 4 with key C, D,E, F
    li $t5, 0x04 # check row 4 with key C, D,E, F
    li $t6, 0x08 # check row 4 with key C, D,E, F


polling:


    sb $t4,0($t1 ) # must reassign expected row
    nop
    lb $a0,0($t2)  # read scan code of key button
    nop
    bne $a0, $zero,print
    sb $t5,0($t1 ) # must reassign expected row
```

```
    nop
    lb $a0,0($t2)  # read scan code of key button
    nop
    bne $a0, $zero, print
    sb $t6,0($t1 ) # must reassign expected row
    nop
    lb $a0,0($t2)  # read scan code of key button
    nop
    bne $a0, $zero, print
    sb $t3,0($t1 ) # must reassign expected row
    nop
    lb $a0,0($t2)  # read scan code of key button
    nop
```

```
j print
print:
li $v0, 34    # print integer (hexa)
syscall
nop
la $a0, endline
li $v0, 4
syscall
nop
sleep:
li $a0, 3000  # sleep 3000ms
```

```
li $v0, 32
syscall
nop
back_to_polling:
j polling    # continue polling
```

```
0x00000000
0x00000024
0xffffffff82
0x00000021
0x00000011
0x00000000
```

Output khi bấm các số khác và bằng 0x00000000 khi không bấm số nào

Bài 2

```
.eqv IN_ADRESS_HEX_KEYBOARD 0xFFFF0012
.data
Message: .asciiz "Oh my god. Someone's presed a button.\n"
endline: .asciiz "\n"
```

```
#~~~~~
# MAIN Procedure
#~~~~~
.text
main:
#-----
# Enable interrupts you expect
```

```
#-----
```

```
# Enable the interrupt of Keyboard matrix 4x4 of Digital Lab Sim
```

```
li $t1, IN_ADRESS_HEXa_KEYBOARD
```

```
li $t3, 0x80 # bit 7 of = 1 to enable interrupt
```

```
sb $t3, 0($t1)
```

```
#-----
```

```
# No-end loop, main program, to demo the effective of interrupt
```

```
#-----
```

```
Loop:
```

```
li $a0, 0x00000000
```

```
li $v0, 34 # print integer (hexa)
```

```
syscall
```

```
nop
```

```
la $a0, endlne
```

```
li $v0, 4
```

```
syscall
```

```
nop
```

```
li $a0, 3000 # sleep 100ms
```

```
li $v0, 32
```

```
syscall
```

```
b Loop # Wait for interrupt
```

```
end_main:
```

```
#~~~~~
```

```
# GENERAL INTERRUPT SERVED ROUTINE for all interrupts
```

```
#~~~~~
```

```
.ktext 0x80000180
```

```
#-----
```

```
# Processing
```

```
#-----
```

```
IntSR: addi $v0, $zero, 4 # show message
```

```
la $a0, Message
```

```
syscall
```

```
#-----
```

```
# Evaluate the return address of main routine
```

```
# epc <= epc + 4
```

```
#-----
```

```
next_pc:
```

```
mfc0 $at, $14 # $at <= Coproc0.$14 = Coproc0.epc
```

```
addi $at, $at, 4 # $at = $at + 4 (next instruction)
```

```
mtc0 $at, $14 # Coproc0.$14 = Coproc0.epc <= $at
```

```
return: eret # Return from exception
```

```
0x00000000
0x00000000
0x00000000
0x00000000
Oh my god. Someone's presed a button.
```

Code sẽ in ra 0x00000000 liên tục cho đến khi bấm nút nào đó nó sẽ hiện ra message

Bài 3

```
.eqv IN_ADRESS_HEXА_KEYBOARD 0xFFFF0012
```

```
.eqv OUT_ADRESS_HEXА_KEYBOARD 0xFFFF0014
```

```
.data
```

```
Message: .asciiz "Key scan code "
```

```
#~~~~~
```

```
# MAIN Procedure
```

```
#~~~~~
```

```
.text
```

```
main:
```

```
    #-----
```

```
    # Enable interrupts you expect
```

```
    #-----
```

```
    # Enable the interrupt of Keyboard matrix 4x4 of Digital Lab sim
```

```
    li $t1, IN_ADRESS_HEXА_KEYBOARD
```

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

```
    sb $t3, 0($t1)
```

```
    #-----
```

```
    # Loop an print sequence numbers
```

```
    #-----
```

```
    xor $s0, $s0, $s0 # count = $s0 = 0
```

```
Loop: addi $s0, $s0, 1    # count = count + 1
```

```
prn_seq:addi $v0,$zero,1
```

```
    add  $a0,$s0,$zero # print auto sequence number
```

```
    syscall
```

```
prn_eol:addi $v0,$zero,11
```

```
    li   $a0,'\n'  # print endofline
```

```
    syscall
```

```
sleep: addi $v0,$zero,32
```

```
    li   $a0,300    # sleep 300 ms
```

```
    syscall
```

```
    nop
```

```
    b    Loop      # Loop
```

```
end_main:
```

```
#~~~~~
```

```
# GENERAL INTERRUPT SERVED ROUTINE for all interrupts
```

```
#~~~~~
```

```
.ktext 0x80000180
```

```
    #-----
```

```
    # SAVE the current REG FILE to stack
```

```
    #-----
```

```
IntSR: addi $sp,$sp,4 # Save $ra because we may change it later
```

```
    sw  $ra,0($sp)
```

```
    addi $sp,$sp,4 # Save $ra because we may change it later
```

```
    sw  $at,0($sp)
```

```
    addi $sp,$sp,4 # Save $ra because we may change it later
```

```
    sw  $v0,0($sp)
```

```
addi $sp,$sp,4 # Save $a0, because we may change it later
```

```
sw $a0,0($sp)
```

```
addi $sp,$sp,4 # Save $t1, because we may change it later
```

```
sw $t1,0($sp)
```

```
addi $sp,$sp,4 # Save $t3, because we may change it later
```

```
sw $t3,0($sp)
```

```
#-----
```

```
# Processing
```

```
#-----
```

```
prn_msg:addi $v0,$zero, 4
```

```
la $a0, Message
```

```
syscall
```

```
get_cod:
```

```
li $t1, IN_ADRESS_HEX_KEYBOARD
```

```
li $t3, 0x82 # check row 4 and re-enable bit 7
```

```
sb $t3, 0($t1) # must reassign expected row
```

```
li $t1, OUT_ADRESS_HEX_KEYBOARD
```

```
lb $a0, 0($t1)
```

```
bne $a0,$zero, prn_cod
```

```
li $t1, IN_ADRESS_HEX_KEYBOARD
```

```
li $t3, 0x84 # check row 4 and re-enable bit 7
```

```
sb $t3, 0($t1) # must reassign expected row
```

```
li $t1, OUT_ADRESS_HEX_KEYBOARD
```



```
lb  $a0, 0($t1)
bne $a0, $zero, prn_cod
```

```
li  $t1, IN_ADRESS_HEXa_KEYBOARD
li  $t3, 0x88  # check row 4 and re-enable bit 7
sb  $t3, 0($t1) # must reassign expected row
li  $t1, OUT_ADRESS_HEXa_KEYBOARD
lb  $a0, 0($t1)
bne $a0, $zero, prn_cod
```

```
li  $t1, IN_ADRESS_HEXa_KEYBOARD
li  $t3, 0x81  # check row 4 and re-enable bit 7
sb  $t3, 0($t1) # must reassign expected row
li  $t1, OUT_ADRESS_HEXa_KEYBOARD
lb  $a0, 0($t1)
```

prn_cod:

```
li  $v0, 34
syscall
li  $v0, 11
li  $a0, '\n'  # print endofline
```

```
syscall
```

```
#-----
```

```
# Evaluate the return address of main routine
```

```
# epc <= epc + 4
```

```
#-----
```

```
next_pc:mfc0  $at, $14  # $at <= Coproc0.$14 = Coproc0.epc
```

```

addi $at, $at, 4  # $at = $at + 4 (next instruction)
mtc0 $at, $14    # Coproc0.$14 = Coproc0.epc <= $at

```

```

#-----

```

```

# RESTORE the REG FILE from STACK

```

```

#-----

```

```

restore:lw  $t3, 0($sp)  # Restore the registers from stack

```

```

addi $sp, $sp, -4

```

```

lw  $t1, 0($sp)  # Restore the registers from stack

```

```

addi $sp, $sp, -4

```

```

lw  $a0, 0($sp)  # Restore the registers from stack

```

```

addi $sp, $sp, -4

```

```

lw  $v0, 0($sp)  # Restore the registers from stack

```

```

addi $sp, $sp, -4

```

```

lw  $ra, 0($sp)  # Restore the registers from stack

```

```

addi $sp, $sp, -4

```

```

return:eret      # Return from exception

```

```

5
6
7
8
Key scan code 0x00000022
9
10
11
12Key scan code 0xffffffff84

```

Nhập các phím 5 và b sẽ hiện ra key scan code 0x22 và 0x84