Assignment 3: MARSBOT RIDER

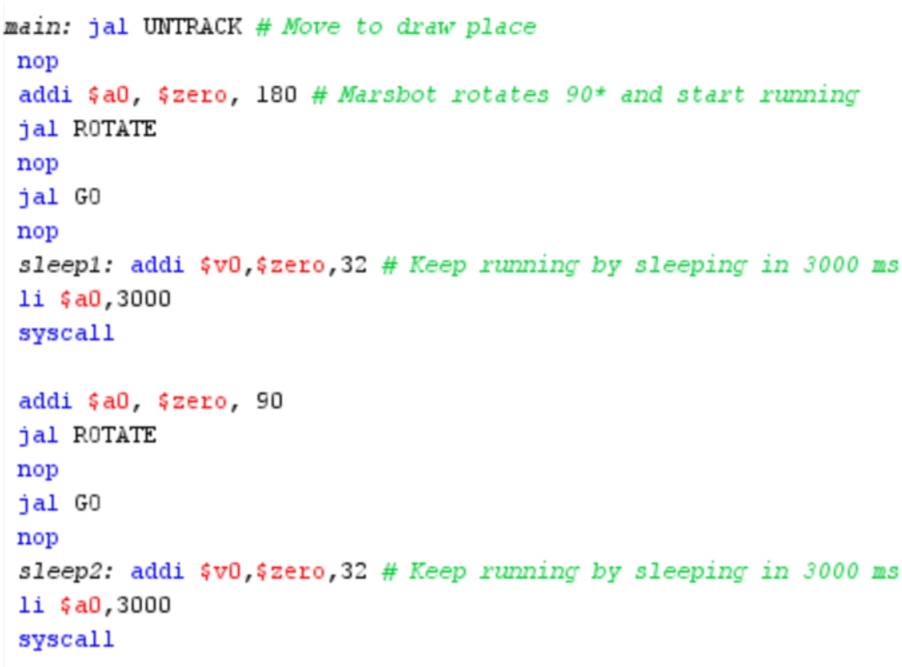
Mã nguồn:

```
.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: jal UNTRACK # Move to draw place
 nop
 addi $a0, $zero, 180 # Marsbot rotates 90* and start running
 jal ROTATE
 nop
 jal GO
 nop
 sleep1: addi $v0,$zero,32 # Keep running by sleeping in 3000 ms
 li $a0,3000
 syscall

 addi $a0, $zero, 90
 jal ROTATE
 nop
 jal GO
 nop
 sleep2: addi $v0,$zero,32 # Keep running by sleeping in 3000 ms
 li $a0,3000
 syscall

 startdraw:
 b1: jal TRACK
 nop
 addi $a0, $zero, 180
 jal ROTATE
 nop
 jal GO
 nop
 sleep3: addi $v0,$zero,32 # Keep running by sleeping in 6000 ms
 li $a0,7000
 syscall
 jal UNTRACK # keep old track
 nop
 jal TRACK # and draw new track line
 nop

b2: addi $a0, $zero, 90
 jal ROTATE
 nop
 jal GO
 nop
 sleep4: addi $v0,$zero,32 # Keep running by sleeping in 3000 ms
 li $a0,3000
 syscall
 jal UNTRACK # keep old track
 nop
 jal TRACK # and draw new track line
 nop

 b3: addi $a0, $zero, 0
 jal ROTATE
 nop
 jal GO
 nop
 sleep5: addi $v0,$zero,32 # Keep running by sleeping in 3000 ms
 li $a0,3000
 syscall
 jal UNTRACK # keep old track
 nop
 jal TRACK # and draw new track line
 nop

 b4: addi $a0, $zero, 270
 jal ROTATE
 nop
 jal GO
 nop
 sleep6: addi $v0,$zero,32 # Keep running by sleeping in 3000 ms
 li $a0,3000
 syscall
 jal UNTRACK # keep old track
 nop
 jal TRACK # and draw new track line
 nop

 b5: addi $a0, $zero, 0
 jal ROTATE
 nop
 jal GO
 nop
 sleep7: addi $v0,$zero,32 # Keep running by sleeping in 3000 ms
 li $a0,1000
 syscall
 jal UNTRACK # keep old track
 nop
 jal TRACK # and draw new track line
 nop

##########
b6: addi $a0, $zero, 90
 jal ROTATE
 nop
 jal GO
 nop
 sleep8: addi $v0,$zero,32 # Keep running by sleeping in 3000 ms
 li $a0,3000
 syscall
 jal UNTRACK # keep old track
 nop
 jal TRACK # and draw new track line
 nop

 b7: addi $a0, $zero, 0
 jal ROTATE
 nop
 jal GO
 nop
 sleep9: addi $v0,$zero,32 # Keep running by sleeping in 3000 ms
 li $a0,3000
 syscall
 jal UNTRACK # keep old track
 nop
 jal TRACK # and draw new track line
 nop

 b8: addi $a0, $zero, 270
 jal ROTATE
 nop
 jal GO
 nop
 sleep10: addi $v0,$zero,32 # Keep running by sleeping in 3000 ms
 li $a0,3000
 syscall
 jal UNTRACK # keep old track
 nop
end_main:

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

Giải thích:

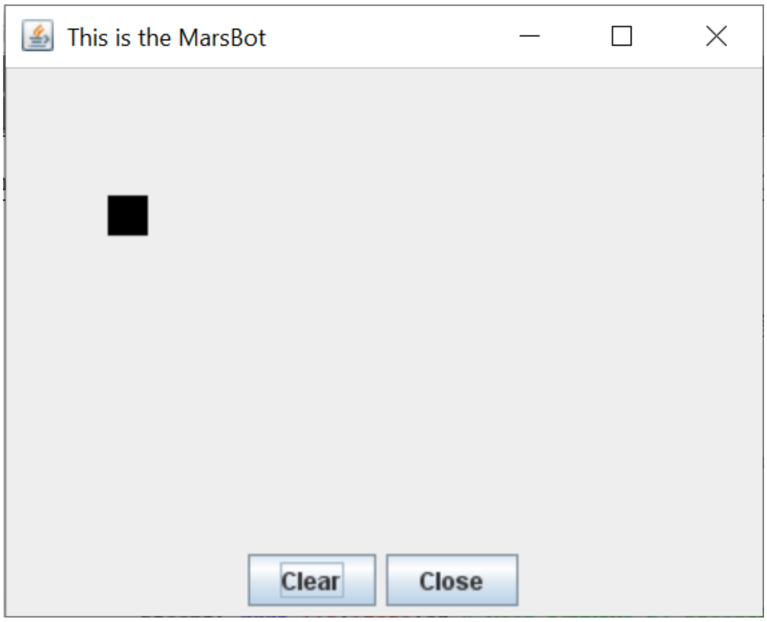
Mục tiêu: Ta cần vẽ chữ B.

- Đầu tiên, chúng ta dùng các lệnh sau để di chuyển Marsbot về địa điểm cần vẽ.

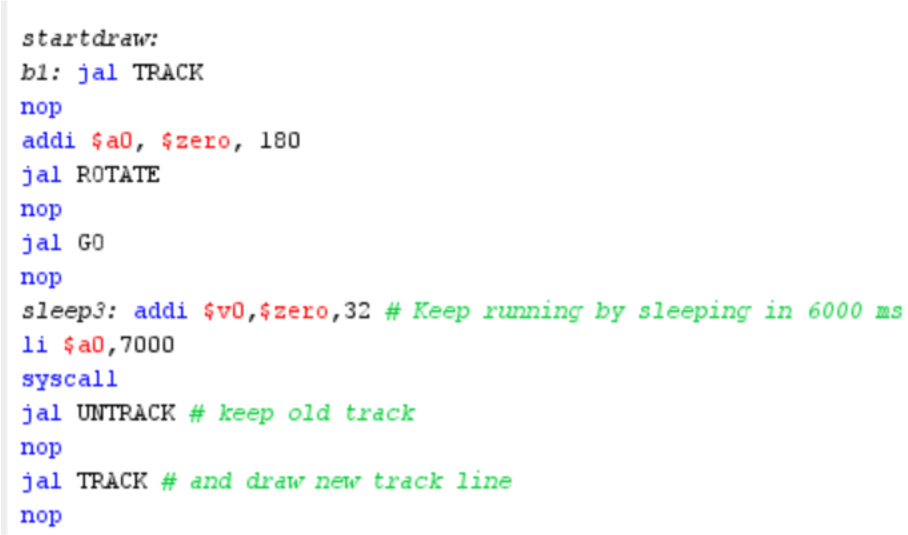


```
main: jal UNTRACK # Move to draw place
 nop
 addi $a0, $zero, 180 # Marsbot rotates 90* and start running
 jal ROTATE
 nop
 jal GO
 nop
 sleep1: addi $v0,$zero,32 # Keep running by sleeping in 3000 ms
 li $a0,3000
 syscall

 addi $a0, $zero, 90
 jal ROTATE
 nop
 jal GO
 nop
 sleep2: addi $v0,$zero,32 # Keep running by sleeping in 3000 ms
 li $a0,3000
 syscall
```

Kết quả thực hiện:

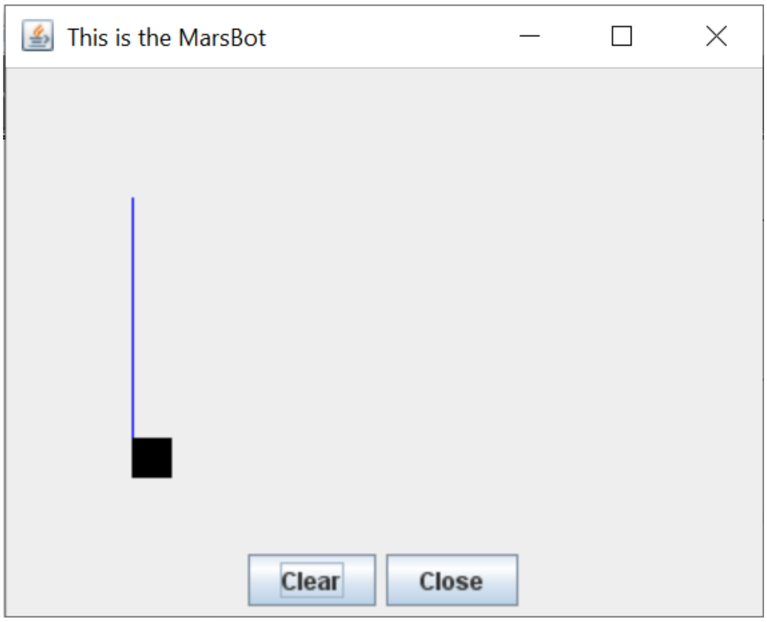


- Tiếp theo, ta dùng các lệnh sau để vẽ nét đầu tiên của chữ B.

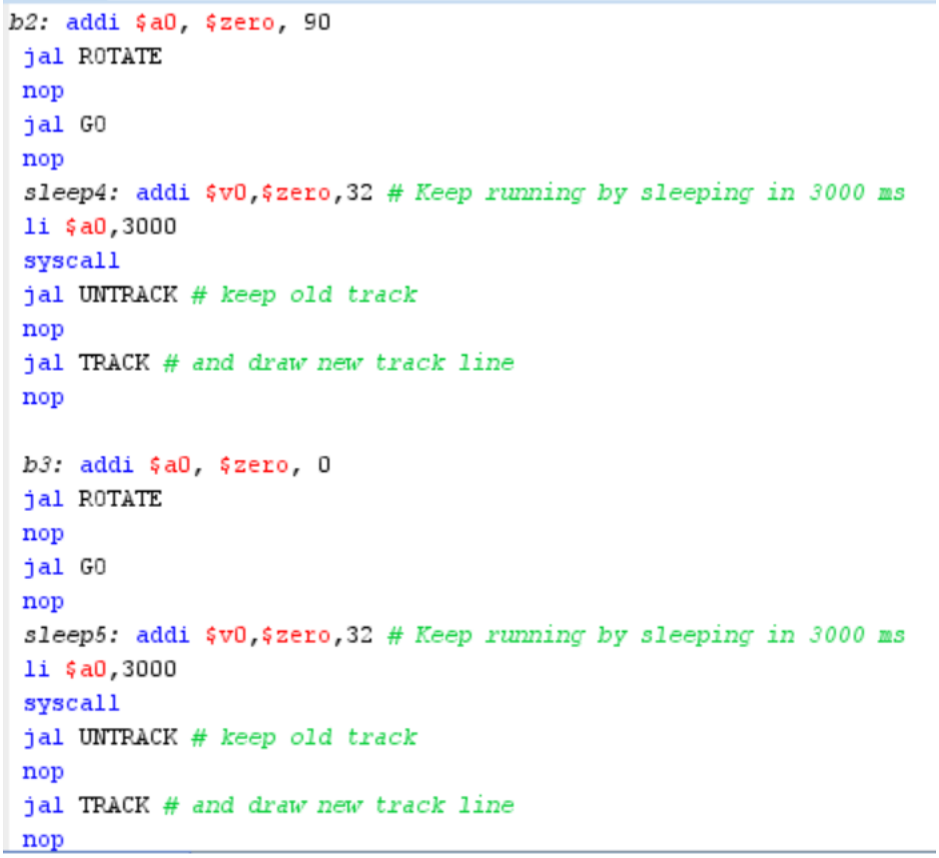


```
startdraw:
b1: jal TRACK
nop
addi $a0, $zero, 180
jal ROTATE
nop
jal GO
nop
sleep3: addi $v0,$zero,32 # Keep running by sleeping in 6000 ms
li $a0,7000
syscall
jal UNTRACK # keep old track
nop
jal TRACK # and draw new track line
nop
```

Kết quả thực hiện:

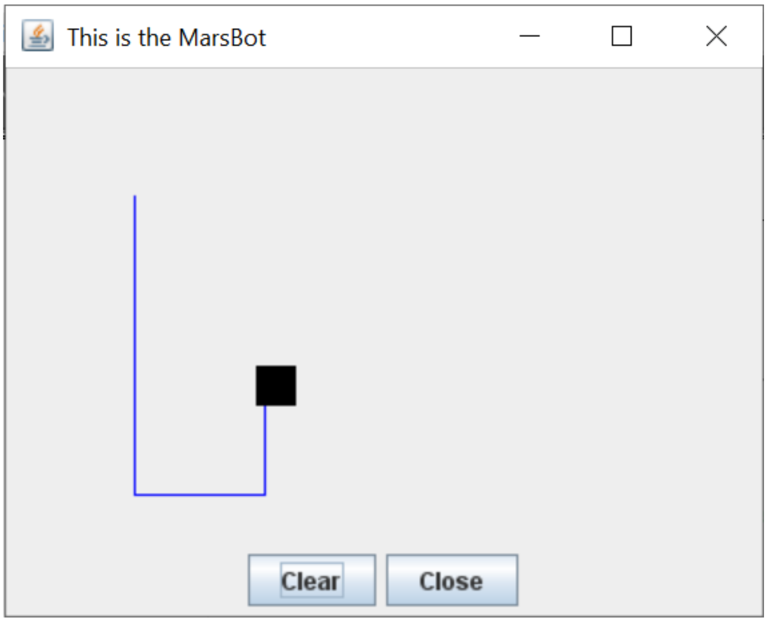


- Tương tự cho các nét sau



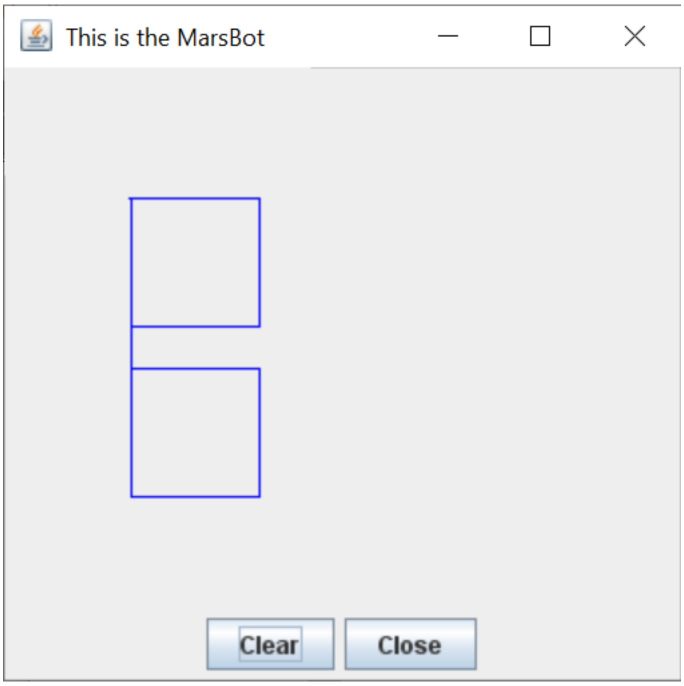
```
b2: addi $a0, $zero, 90
 jal ROTATE
 nop
 jal GO
 nop
 sleep4: addi $v0,$zero,32 # Keep running by sleeping in 3000 ms
 li $a0,3000
 syscall
 jal UNTRACK # keep old track
 nop
 jal TRACK # and draw new track line
 nop

 b3: addi $a0, $zero, 0
 jal ROTATE
 nop
 jal GO
 nop
 sleep5: addi $v0,$zero,32 # Keep running by sleeping in 3000 ms
 li $a0,3000
 syscall
 jal UNTRACK # keep old track
 nop
 jal TRACK # and draw new track line
 nop
```

Kết quả thực hiện:



- Các bước sau tương tự.

Kết quả thực hiện:



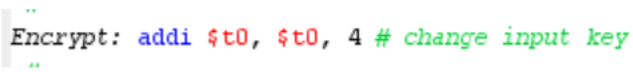
Assignment 4: KEYBOARD and DISPLAY MMIO

Mã nguồn:

```
.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
 nop
 beq $t1, $zero, WaitForKey # if $t1 == 0 then Polling
 nop
 #-----------------------------------------------------
ReadKey: lw $t0, 0($k0) # $t0 = [$k0] = KEY_CODE
 nop
QuitKey:
 beq $t0, 'b', quit
 nop
 #-----------------------------------------------------
WaitForDis: lw $t2, 0($s1) # $t2 = [$s1] = DISPLAY_READY
 nop
 beq $t2, $zero, WaitForDis # if $t2 == 0 then Polling
 nop
 #-----------------------------------------------------
Encrypt: addi $t0, $t0, 4 # change input key
 #-----------------------------------------------------
ShowKey: sw $t0, 0($s0) # show key
 nop
 #-----------------------------------------------------
 j loop
 nop
 quit:
```

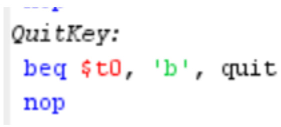
Giải thích:

- So với mã nguồn gốc, ta cần thay đổi giá trị mã hoá từ 1 sang chữ số cuối MSSV *(4)*.



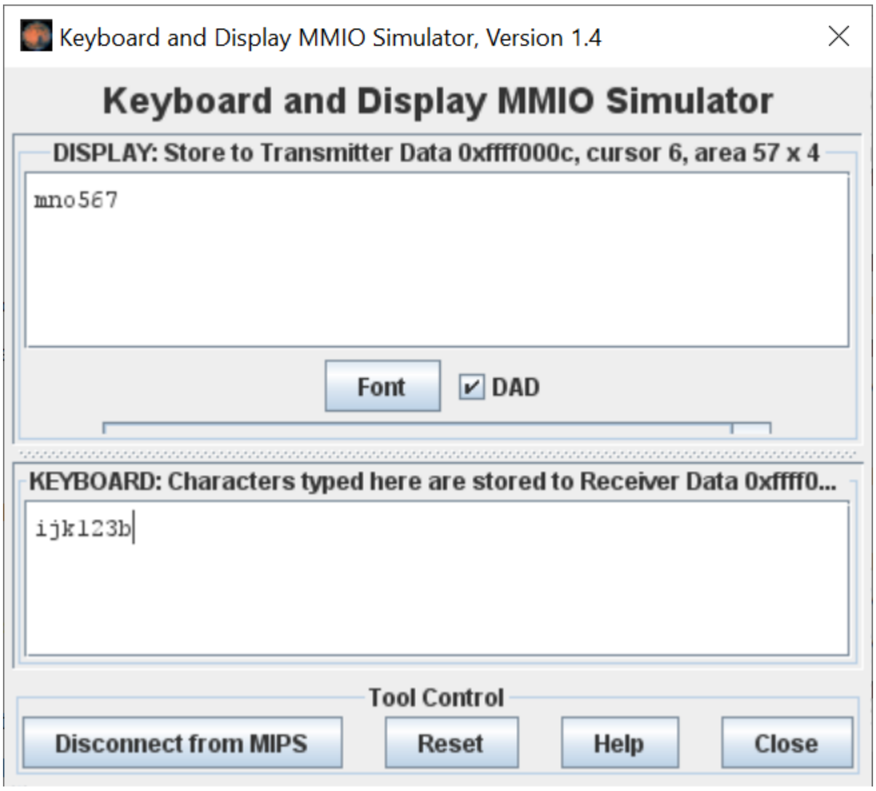
```
Encrypt: addi $t0, $t0, 4 # change input key
```

- Và bổ sung thêm đoạn mã dừng chương trình khi gặp kí tự đầu tiên trong tên *(b)*.



```
QuitKey:
 beq $t0, 'b', quit
 nop
```

Kết quả thực hiện:


Keyboard and Display MMIO Simulator, Version 1.4
Keyboard and Display MMIO Simulator
DISPLAY: Store to Transmitter Data 0xffff000c, cursor 6, area 57 x 4
Font
KEYBOARD: Characters typed here are stored to Receiver Data 0xffff0...
Tool Control
Disconnect from MIPS
Reset
Help
Close

☑ DAD

```
mno567
```

```
ijk123b
```