Họ và Tên: Hà Trung Chiến
MSSV: 20225794

Báo cáo Lab 10(2)

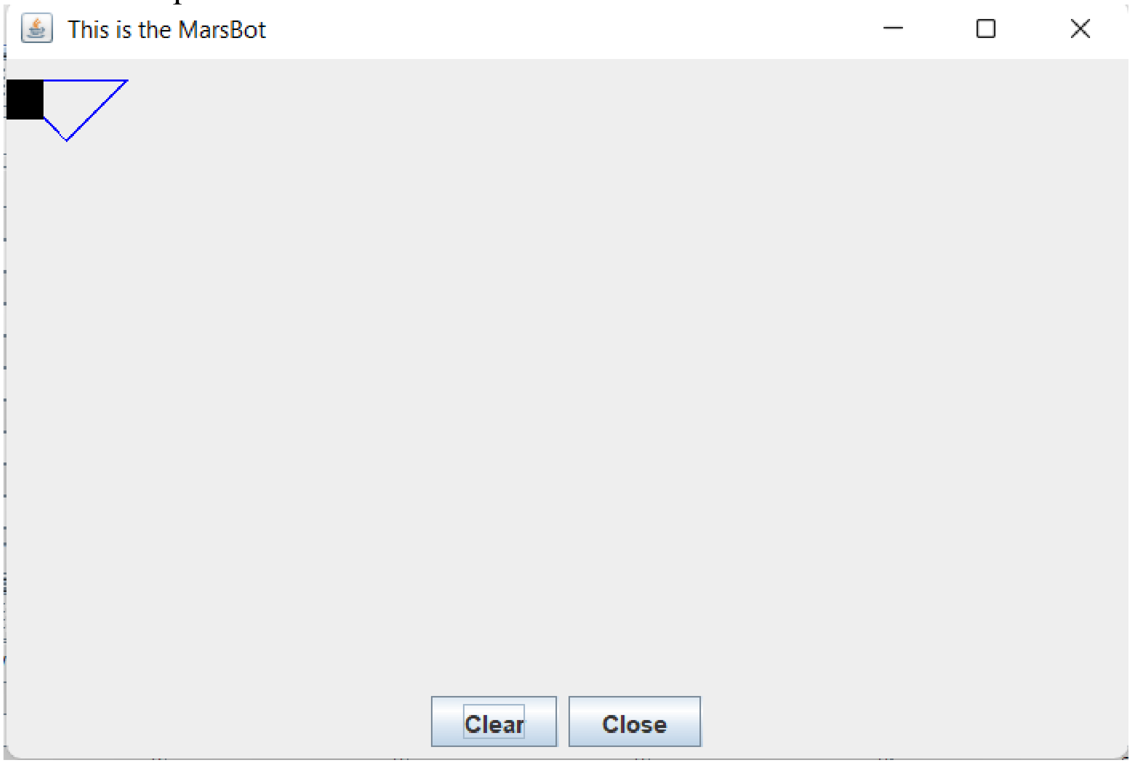
Assignment 3:

```
.eqv HEADING 0xffff8010
.eqv MOVING 0Xffff8050
.eqv LEAVETRACK 0xffff8020
.eqv WHEREX 0xffff8030
.eqv WHEREY 0xffff8040
.text
main:
	li $a0, 180
	jal ROTATE
	jal GO
	li $v0, 32
	li $a0, 500
	syscall
	li $t0, 2
drawTriangle:
	beq $t0, $zero, end_main
goDown:
	jal TRACK
	li $a0, 135
	jal ROTATE
	li $v0, 32
	li $a0, 2000
	syscall
	jal UNTRACK
goUp:
	jal TRACK
	li $a0, 45
	jal ROTATE
	li $v0, 32
	li $a0, 2000
	syscall
	jal UNTRACK
goLeft:
	jal TRACK
	li $a0, 270
```

```
        jal ROTATE
        li $v0, 32
        li $a0, 2850
        syscall
        jal UNTRACK
        addi $t0, $t0, -1
        j drawTriangle
end_main:
        jal STOP
        li $v0, 10
syscall
GO:
        li $at, MOVING # change MOVING port
        addi $k0, $zero,1 # to logic 1,
        sb $k0, 0($at) # to start running
        jr $ra
STOP:
        li $at, MOVING # change MOVING port
        sb $zero, 0($at) # to start running
        jr $ra
TRACK:
        li $at, LEAVETRACK # change LEAVETRACK port
        addi $k0, $zero,1 # to logic 1,
        sb $k0, 0($at) # to start tracking
        jr $ra
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:



Assigment 4:

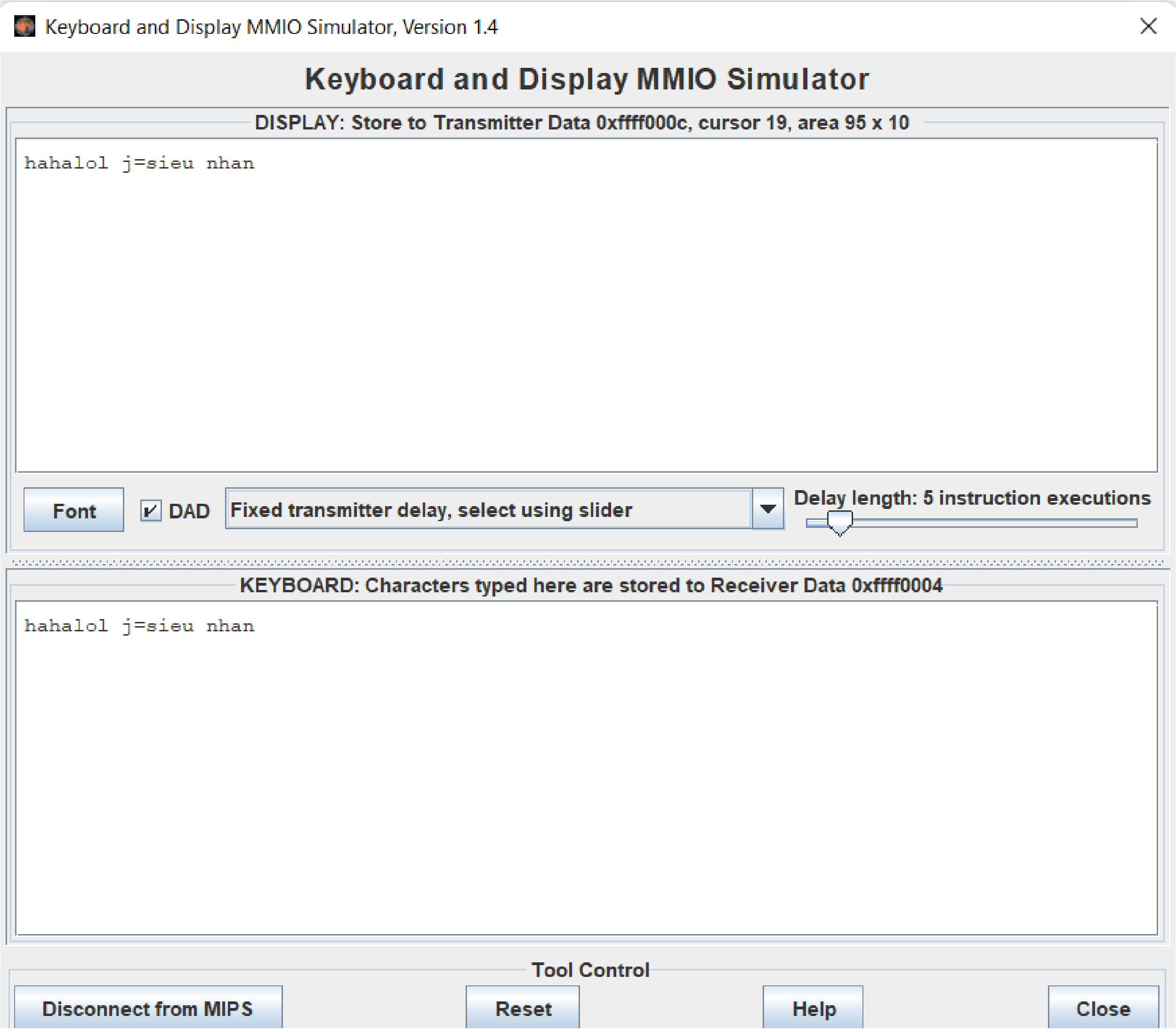
```
.eqv KEY_CODE 0xFFFF0004 # ASCII code from keyboard, 1 byte
.eqv KEY_READY 0xFFFF0000 # =1 if has a new keycode ?
# Auto clear after lw
.eqv DISPLAY_CODE 0xFFFF000C # ASCII code to show, 1 byte
.eqv DISPLAY_READY 0xFFFF0008 # =1 if the display has already to do
# Auto clear after sw
.text
	li $k0, 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
beq $t0, 116, exit
#---------------------------------------------------
WaitForDis:
        lw $t2, 0($s1) # $t2 = [$s1] = DISPLAY_READY
        nop
        beq $t2, $zero, WaitForDis # if $t2 == 0 then Polling
nop
#---------------------------------------------------
Encrypt:
addi $t0, $t0, 0 # change input key
#---------------------------------------------------
ShowKey:
sw $t0, 0($s0) # show key
nop
#---------------------------------------------------
j loop
nop
```

exit:

li $v0,10

syscall

- Kết quả:
- Trước khi gặp “exit”:


Keyboard and Display MMIO Simulator, Version 1.4
Keyboard and Display MMIO Simulator
Font
Tool Control
Disconnect from MIPS
Reset
Help
Close

DISPLAY: Store to Transmitter Data 0xffff000c, cursor 19, area 95 x 10

```
hahalol j=sieu nhan
```

☑ DAD

Fixed transmitter delay, select using slider

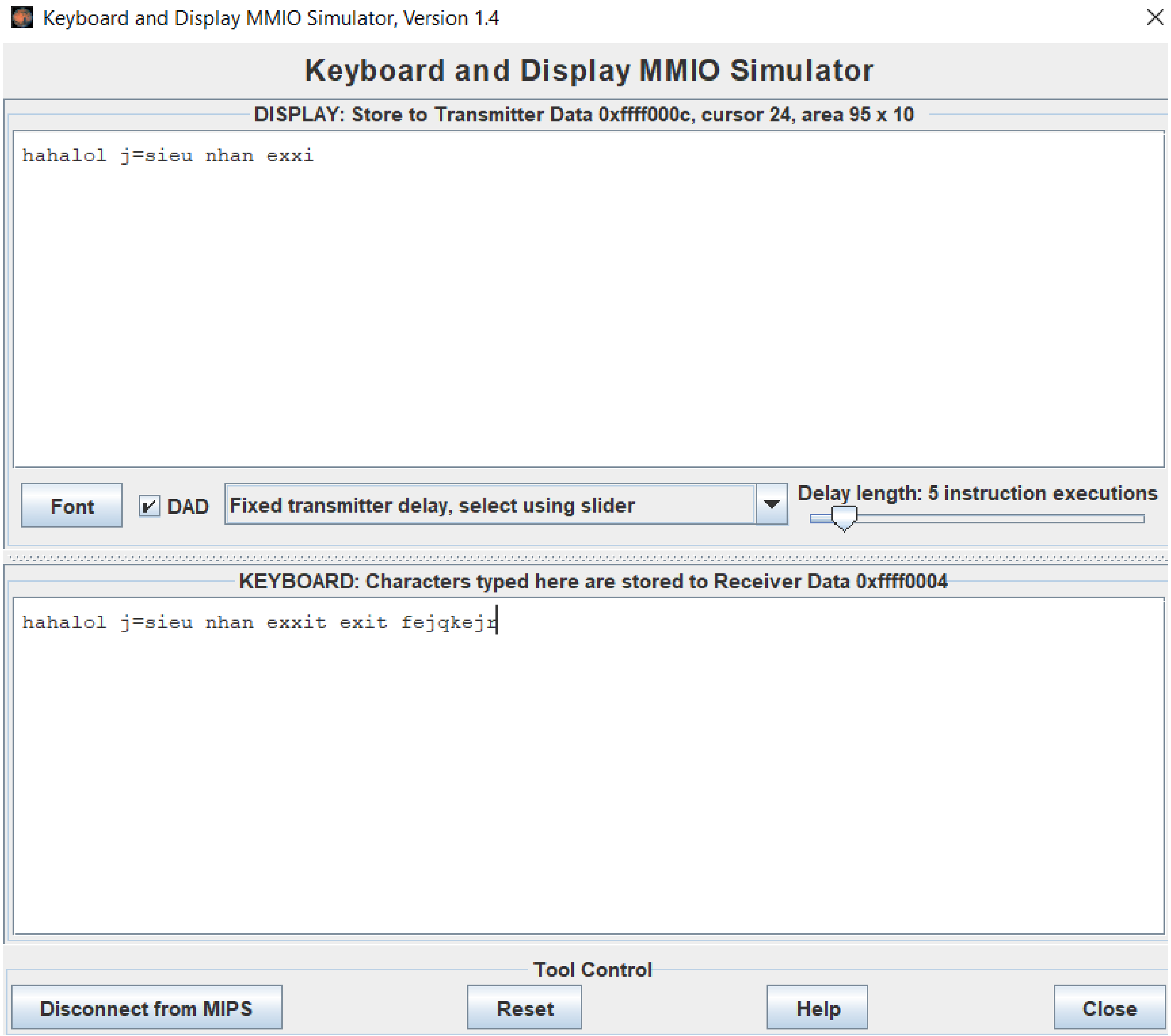
Delay length: 5 instruction executions

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

```
hahalol j=sieu nhan
```

→ Chương trình chạy bình thường và đưa kết quả nhập từ bàn phím hiện ở mục DISPLAY

- Sau khi gặp exit:


Keyboard and Display MMIO Simulator, Version 1.4
Keyboard and Display MMIO Simulator
Font
Tool Control
Disconnect from MIPS
Reset
Help
Close

DISPLAY: Store to Transmitter Data 0xffff000c, cursor 24, area 95 x 10

```
hahalol j=sieu nhan exxi
```

☑ DAD

Fixed transmitter delay, select using slider

Delay length: 5 instruction executions

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

```
hahalol j=sieu nhan exxit exit fejqkejr
```

→ Chương trình tự động ngắt khi gặp “exit” và không nhận input được nhập sau lúc này.