**BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH**

**Lab 10 : Giao tiếp với các thiết bị ngoại vi**

|  |  |
| --- | --- |
| **Họ tên** | **MSSV** |
| Phạm Minh Hiển | 20235705 |

**Assignment 1:**

Tạo project để thực hiện Home Assignment 1. Thay đổi các giá trị hiển thị trên LED 7 đoạn để hiển thị 2 chữ số cuối của MSSV.

**Source Code:**

.eqv SEVENSEG\_LEFT 0xFFFF0011

.eqv SEVENSEG\_RIGHT 0xFFFF0010

.text

main:

li a0, 0x06

# li a0, 0x3F # Hien thi so 0

jal SHOW\_7SEG\_LEFT

li a0, 0x3F

# li a0, 0x6D # Hien thi so 5

jal SHOW\_7SEG\_RIGHT

exit:

li a7, 10

ecall

end\_main:

SHOW\_7SEG\_LEFT:

li t0, SEVENSEG\_LEFT

sb a0, 0(t0)

jr ra

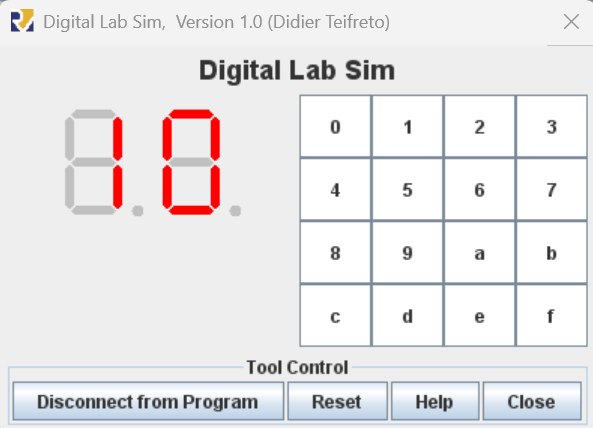
SHOW\_7SEG\_RIGHT:

li t0, SEVENSEG\_RIGHT

sb a0, 0(t0)

jr ra

* Kết quả chạy chương trình Home Assignment 1:



+ **li** a0, 0x06 : thanh **b** và **c** sáng những thanh còn lại tắt.

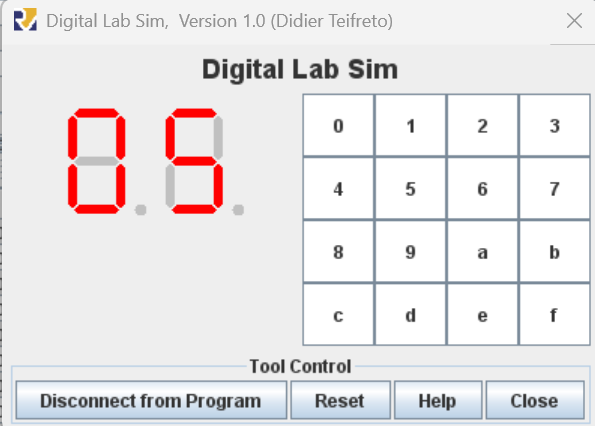
+ **li** a0, 0x3F : thanh **a, b, c, d, e, f** sáng.

* Thay đổi giá trị hiển thị trên đèn LED thành 05 (hai chữ số cuối của MSSV):

+ **li** a0, 0x06 thành **li** a0, 0x3F để hiển thị số 0 ở LED trái.

+ **li** a0, 0x3F thành **li** a0, 0x6D để hiển thị số 5 ở LED phải.

+ Kết quả:



**Assignment 2:**

Tạo project để hiển thị trên LED 7 đoạn 2 chữ số cuối của mã ASCII (ở hệ cơ số 10) của ký tự được nhập từ bàn phím.

**Source Code:**

.eqv SEVENSEG\_LEFT 0xFFFF0011

.eqv SEVENSEG\_RIGHT 0xFFFF0010

.data

SEG7\_TABLE: .byte 0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F

.text

li a7, 12 # Nhap ky tu

ecall

add t1, zero, a0 # Sao chep ma ASCII vao t1

addi t2, zero, 100 # Lay 2 chu so cuoi

rem t1, t1, t2 # t1 = ASCII % 100

addi t2, zero, 10 # Tach chu so hang chuc va don vi

div t3, t1, t2 # s2 = s1 / 10

rem t4, t1, t2 # s3 = s1 % 10

la t5, SEG7\_TABLE

lb a0, 0(t5)

slli t3, t3, 0 # Hien thi LED trai (hang chuc)

add t6, t5, t3

lb a0, 0(t6)

jal SHOW\_7SEG\_LEFT

slli t4, t4, 0 # Hien thi LED phai (hang don vi)

add t6, t5, t4

lb a0, 0(t6)

jal SHOW\_7SEG\_RIGHT

exit:

li a7, 10

ecall

SHOW\_7SEG\_LEFT:

li t0, SEVENSEG\_LEFT

sb a0, 0(t0)

jr ra

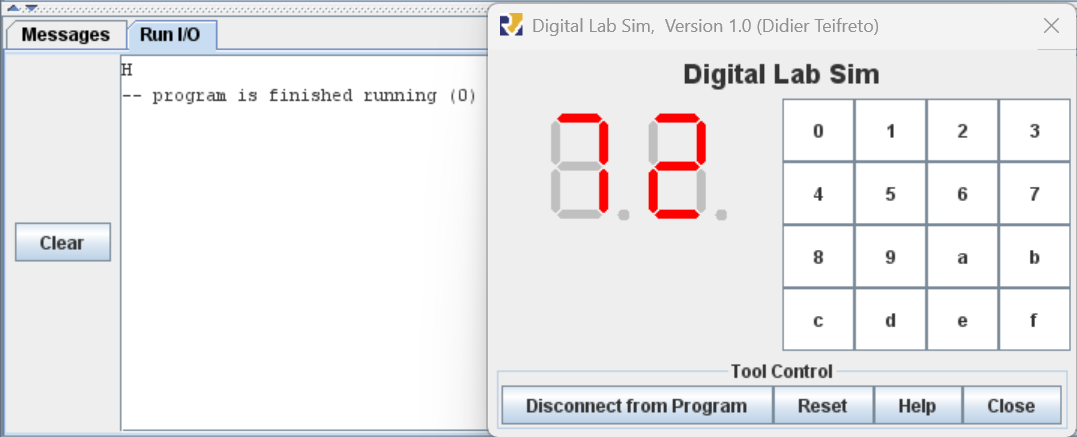
SHOW\_7SEG\_RIGHT:

li t0, SEVENSEG\_RIGHT

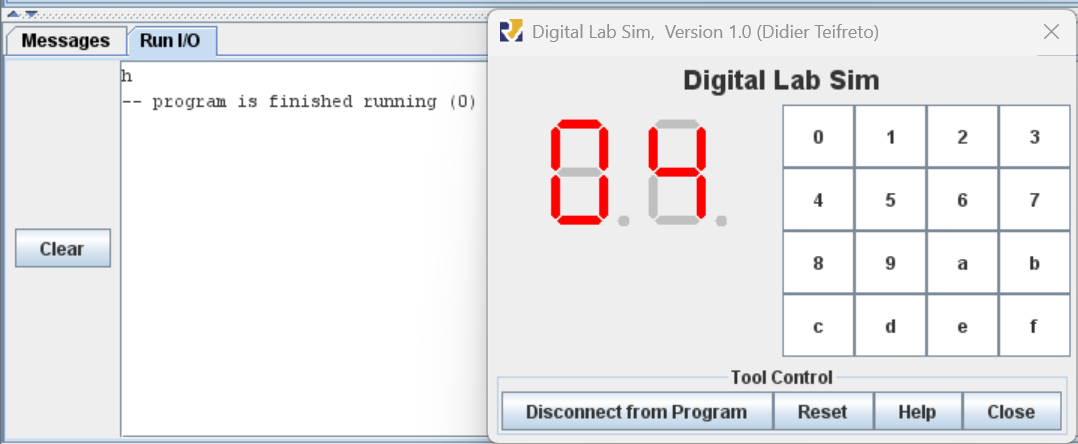
sb a0, 0(t0)

jr ra

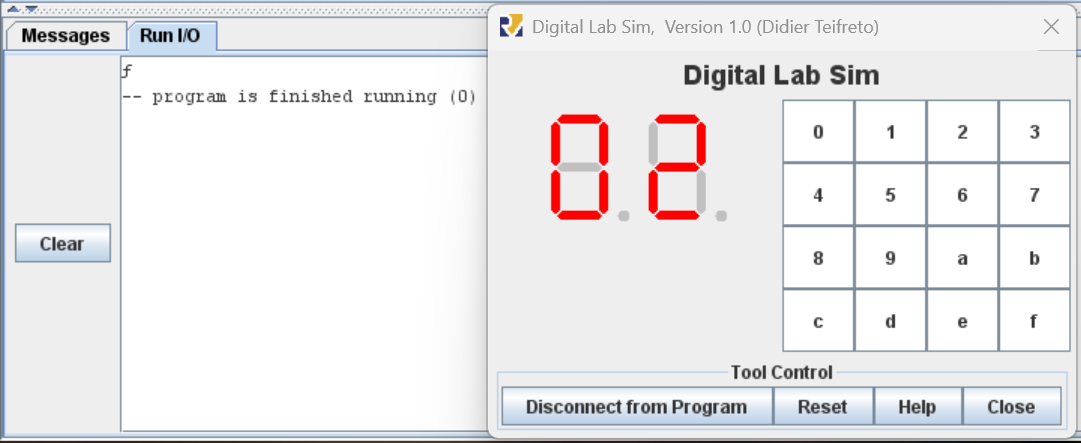
* Kết quả chạy:



+ Thử với ký tự ‘H’ trong bảng mã ASCII có giá trị 72.



+ Thử với ký tự ‘h’ có giá trị trong bảng mã ASCII là 104.

+ Với những ký tự không thuộc 128 ký tự cơ bản của ASCII thì sẽ hiển thị không chính xác:

* Chương trình hoạt động như ý.

**Assignment 3:**

Tạo project để thực hiện Home Assignment 2. Cập nhật mã nguồn để vẽ bàn cờ vua trên màn hình với 2 màu bất kỳ (khác màu đen).

**Source Code:**

.eqv MONITOR\_SCREEN 0x10010000

.eqv BLUE 0x000000FF

.eqv WHITE 0x00FFFFFF

.eqv WIDTH 8 # Ban co 8x8

.text

li a0, MONITOR\_SCREEN

li t1, WIDTH # So hang/cot

li t2, 0 # i : dem hang

row:

li t3, 0 # j : dem cot

andi t4, t2, 1 # Kiem tra chan le, 0 neu chan, 1 neu le

col:

add t5, t4, t3 # j + i%2

andi t5, t5, 1 # Kiem tra chan le de chon mau

slli t6, t2, 5 # i \* 32 (8 cot 4 byte)

slli a1, t3, 2 # j \* 4

add a1, a1, t6 # offset

add a1, a1, a0 # Dia chi o co

beq t5, zero, draw\_white

li a2, BLUE

j store\_color

draw\_white:

li a2, WHITE

store\_color:

sw a2, 0(a1) # Luu mau vao o

addi t3, t3, 1 # Tang j

blt t3, t1, col

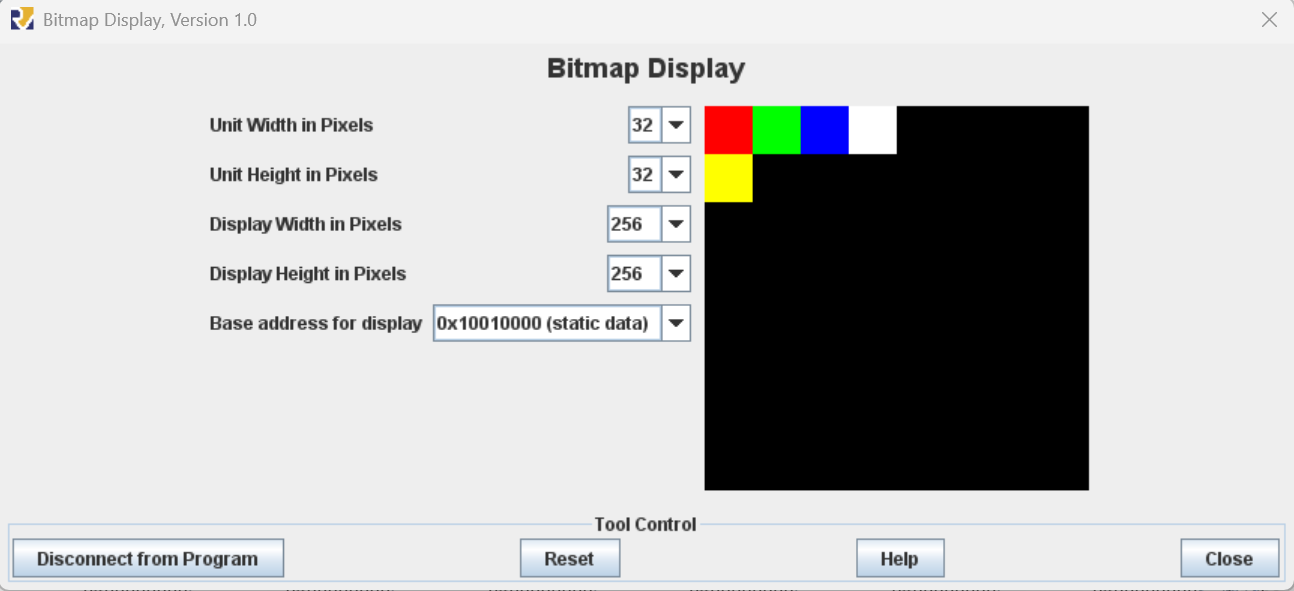
addi t2, t2, 1 # Tang i

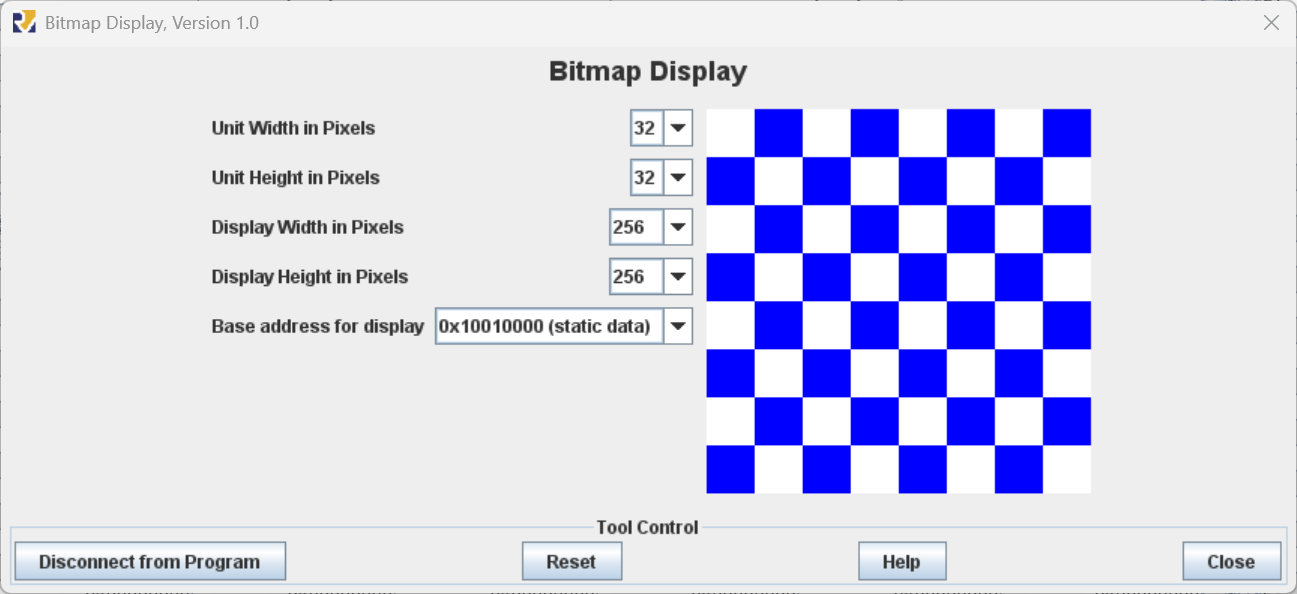
blt t2, t1, row

exit:

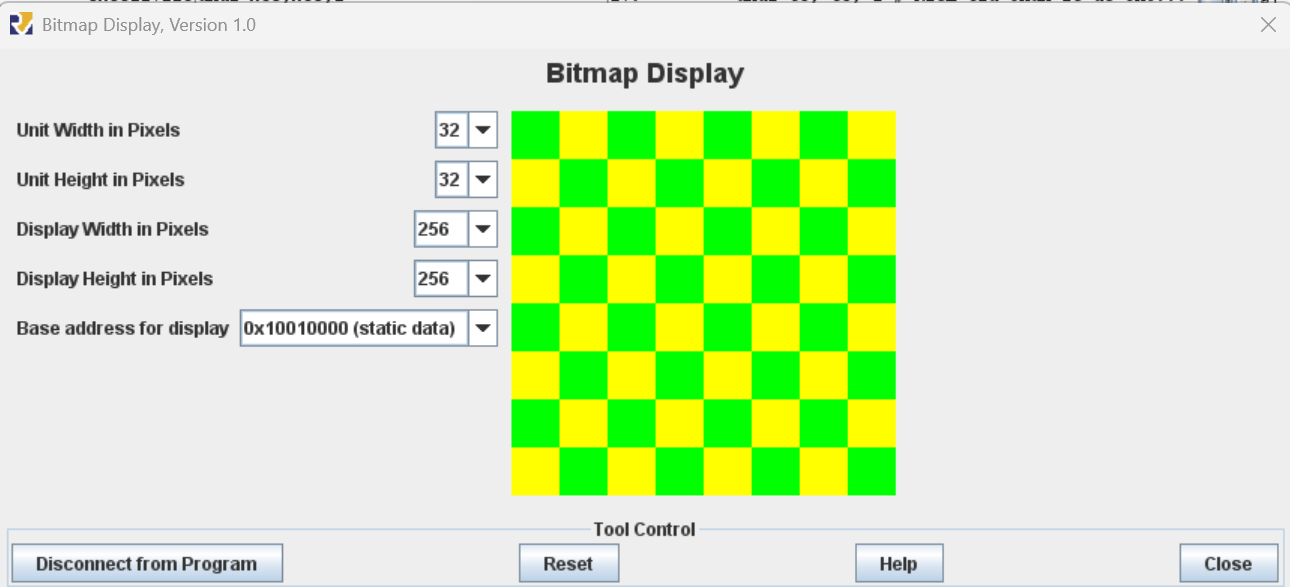
li a7, 10

ecall

* Khi chạy Home Assignment 2:
* Kết quả chạy thử code vẽ bàn cờ vua:

****

* Thử với các màu khác:

****

* Chương trình hoạt động.

**Assignment 4:**

Tạo project để thực hiện Home Assignment 3. Cập nhật mã nguồn để hoàn thành yêu cầu sau: Nhập ký tự thường => hiển thị ký tự hoa tương ứng, nhập ký tự hoa => hiển thị ký tự thường tương ứng, nhập ký tự số thì giữ nguyên, nhập ký tự khác => hiển thị ký tự \*. Khi nhập chuỗi ký tự "exit" thì kết thúc chương trình.

**Source Code:**

.eqv KEY\_CODE 0xFFFF0004

.eqv KEY\_READY 0xFFFF0000

.eqv DISPLAY\_CODE 0xFFFF000C

.eqv DISPLAY\_READY 0xFFFF0008

.data

exit\_str: .ascii "EXIT"

buffer: .byte 0, 0, 0, 0

.text

li a0, KEY\_CODE

li a1, KEY\_READY

li s0, DISPLAY\_CODE

li s1, DISPLAY\_READY

la s2, buffer # Dia chi buffer

li s3, 0 # Bien dem vi tri buffer

la s4, exit\_str # Dia chi chuoi "exit"

li s5, 4 # Do dai chuoi "exit"

loop:

WaitForKey:

lw t1, 0(a1)

beq t1, zero, WaitForKey

ReadKey:

lw t0, 0(a0)

li t2, 'a' # Kiem tra va xu ly ky tu

blt t0, t2, CheckUpper # Neu < 'a' kiem tra hoa

li t2, 'z'

bgt t0, t2, Other # Neu > 'z' kiem tra ky tu khac

addi t0, t0, -32 # Neu chu thuong thi viet hoa

j Char

CheckUpper:

li t2, 'A' # Kiem tra va xu ly ky tu

blt t0, t2, CheckDigit # Neu < 'A' kiem tra so

li t2, 'Z'

bgt t0, t2, Other # Neu > 'Z' kiem tra ky tu khac

addi t0, t0, 32 # Neu chu hoa thi viet thuong

j Char

CheckDigit:

li t2, '0'

blt t0, t2, Other # Neu < '0' kiem tra ky tu khac

li t2, '9'

bgt t0, t2, Other # Neu > '9' kiem tra ky tu khac

j Char

Other:

li t0, '\*' # Thay ky tu khac bang '\*'

Char:

bne s3, s5, SkipShift # Meu buffer chua du thi skip shift

lb t3, 1(s2) # load byte 1 vao byte 0

sb t3, 0(s2)

lb t3, 2(s2)

sb t3, 1(s2)

lb t3, 3(s2)

sb t3, 2(s2)

addi s3, s3, -1 # i -= 1

SkipShift:

add t3, s2, s3 # Dia chi o trong buffer

sb t0, 0(t3) # Luu ky tu vao buffer

addi s3, s3, 1 # i += 1

blt s3, s5, ExitCheck # Neu buffer chua day thi bo qua

li s3, 4

ExitCheck:

li t3, 0 # j

blt s3, s5, Display # Neu chua du ky tu thi skip check

ExitLoop:

bge t3, s5, Exit # Kiem tra du 4 ky tu

add t4, s2, t3 # Dia chi ky tu trong buffer

lb t5, 0(t4) # Lay ky tu

add t6, s4, t3 # Dia chi ky tu "exit"

lb t6, 0(t6)

bne t5, t6, Display # Neu khac thi hien thi

addi t3, t3, 1 # j += 1

j ExitLoop

Display:

WaitForDis:

lw t2, 0(s1)

beq t2, zero, WaitForDis

ShowKey:

sw t0, 0(s0)

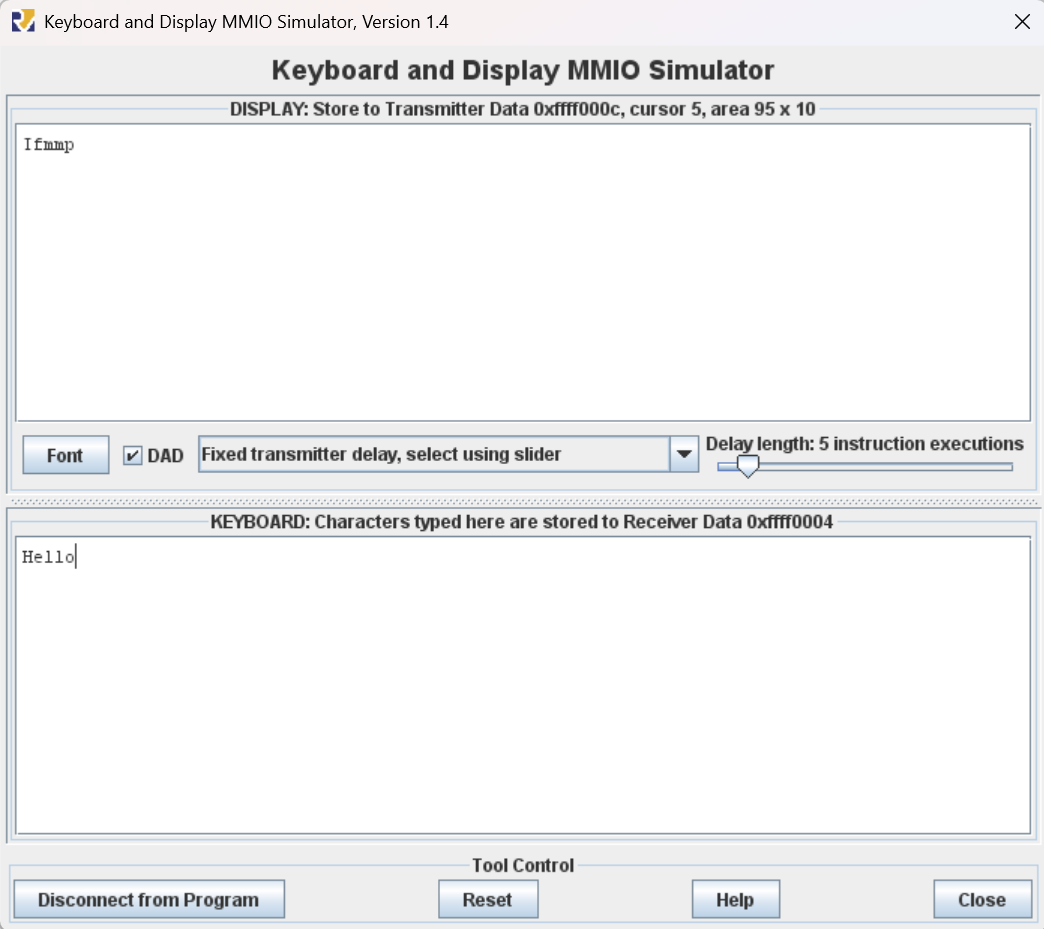
j loop

Exit:

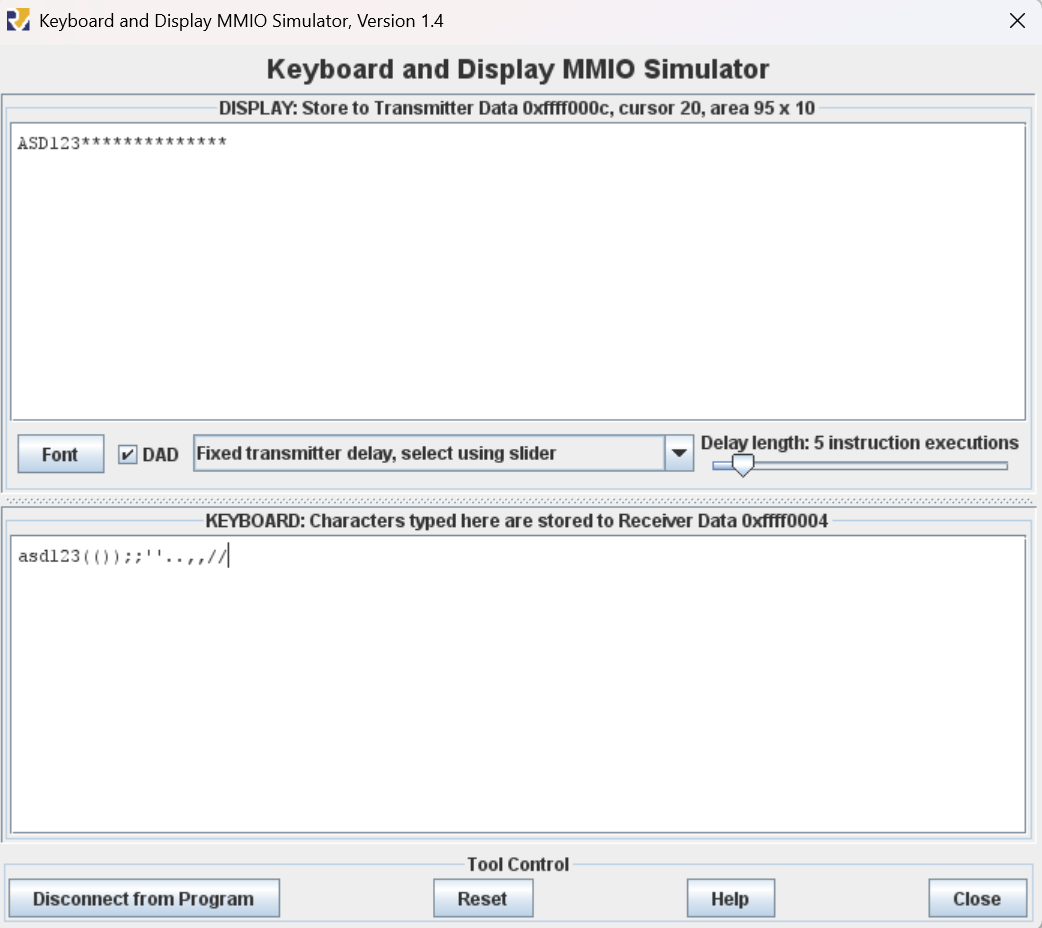
li a7, 10

ecall

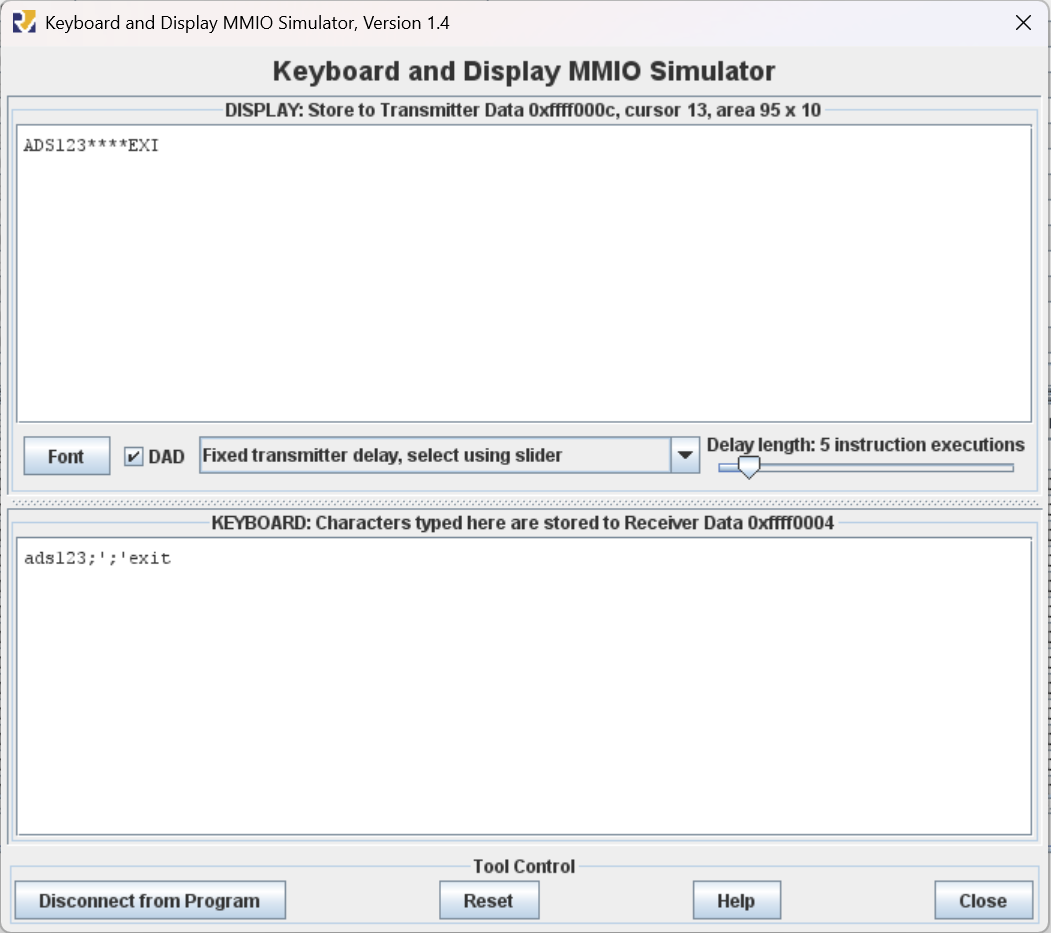
* Kết quả chạy Home Assignment 3:



* Chạy code sửa theo yêu cầu :



* Khi nhập chuỗi “exit” từ keyboard (display: “EXIT”) thì chương trình sẽ dừng:



* Chương trình hoạt động như yêu cầu.

**Additional Assignment:**

Vẽ hình ảnh cờ đỏ sao vàng lên bitmap display với kích thước hình vuông (512x512 hoặc 256x256).

A screenshot of a computer

AI-generated content may be incorrect.