

BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH

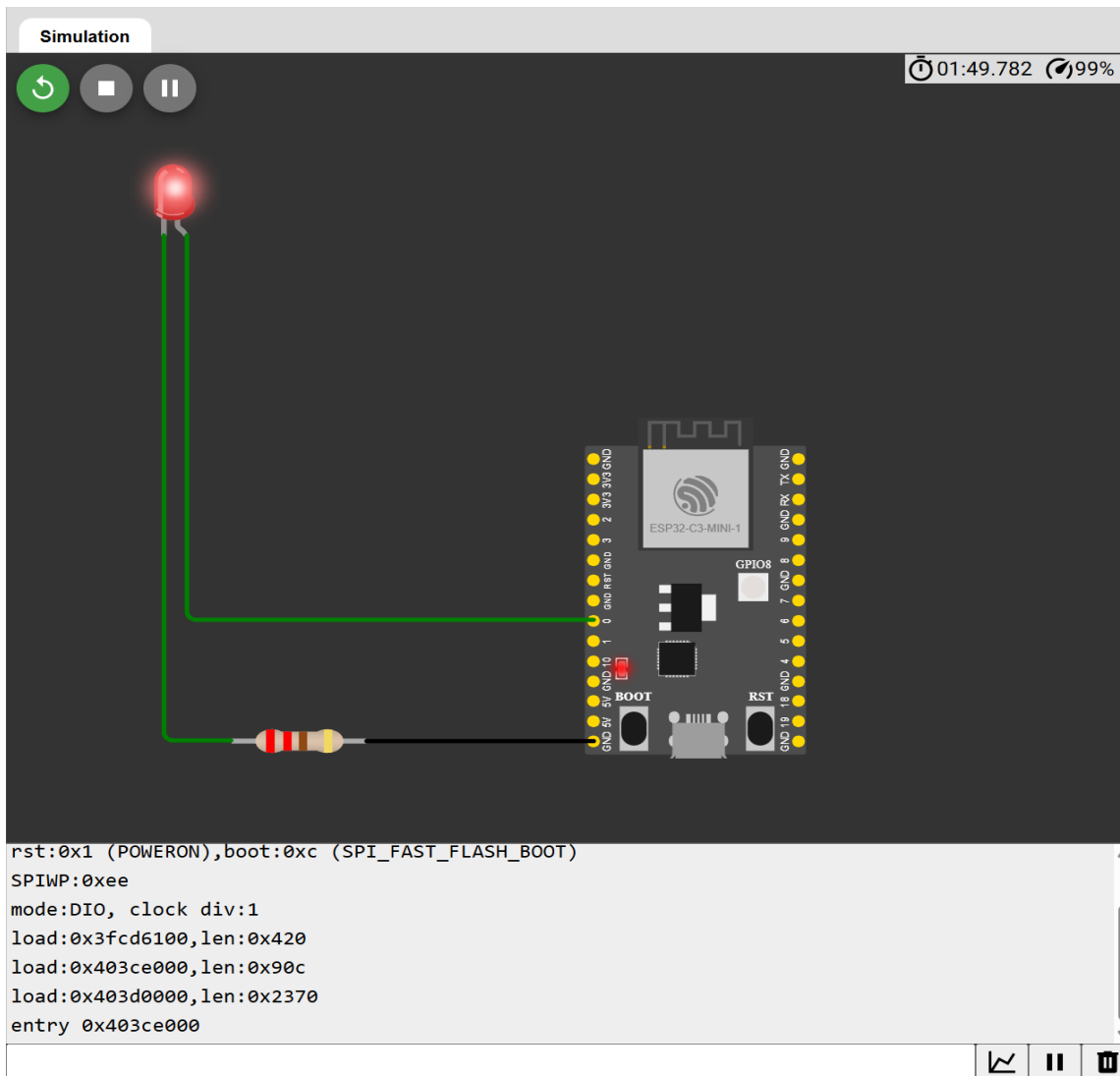
Lab 13 : Lập trình hợp ngữ với ESP32-C3 – Mô phỏng bằng Wokwi (Tham khảo)

Họ tên	MSSV
Phạm Minh Hiền	20235705

Assignment 1:

Tạo project để thực hiện và thử nghiệm Home Assignment 1. Cập nhật mã nguồn để thử nghiệm với các cổng GPIO khác (GPIO2, GPIO3, GPIO4).

- Kết quả chạy thử:



- Cập nhật mã nguồn:

```
.global init
```

```
.eqv GPIO_ENABLE_REG, 0x60004020 # Thanh ghi cấu hình vào/ra các chân GPIO
```

```
.eqv GPIO_OUT_W1TS_REG, 0x60004008 # Thanh ghi thiết lập chân GPIO
```

```
init:
```

```
li a1, GPIO_ENABLE_REG # Thiết lập chân GPIO0 là chân xuất tín hiệu
```

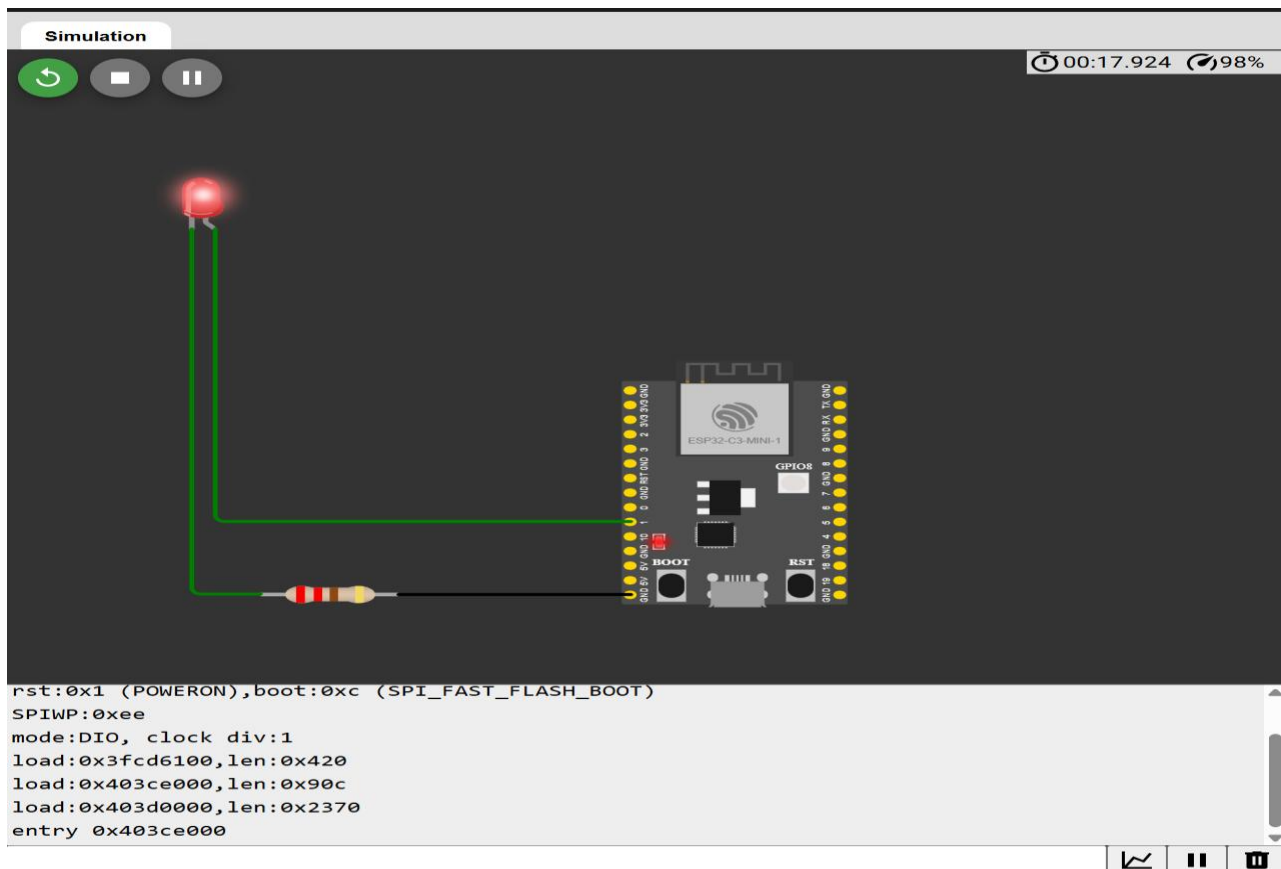
```
li a2, 0x02 # Nạp mặt nạ GPIO 1, 2, 3, 4 lần lượt là 0x02, 0x04, 0x08, 0x10
```

```
sw a2, 0(a1)
```

```
li a1, GPIO_OUT_W1TS_REG # Thiết lập chân GPIO0 ở mức cao, bật LED
```

```
li a2, 0x02 # Nạp mặt nạ GPIO 1, 2, 3, 4 lần lượt là 0x02, 0x04, 0x08, 0x10
```

```
sw a2, 0(a1)
```

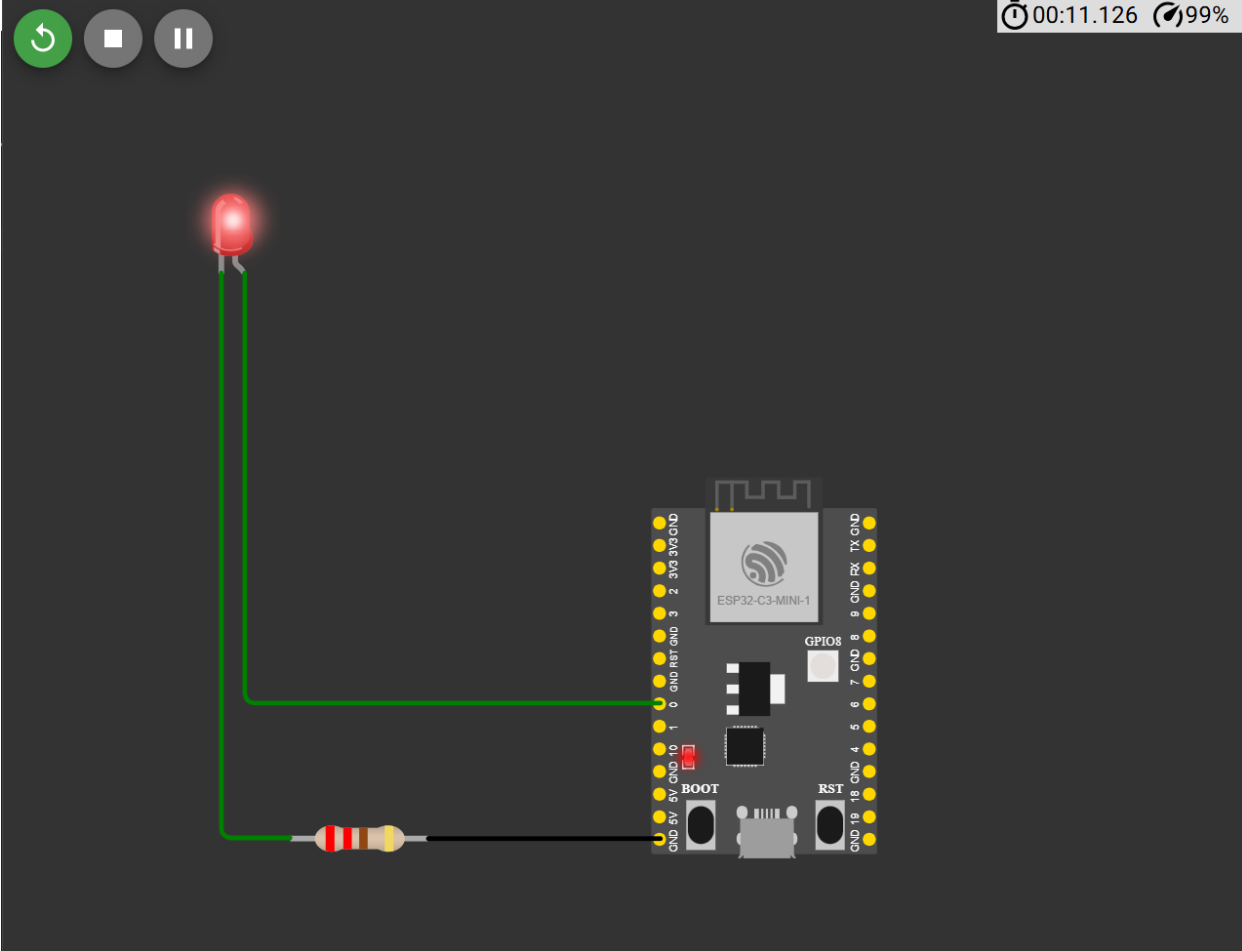


Assignment 2:

Tạo project để thực hiện và thử nghiệm Home Assignment 2. Cập nhật mã nguồn để thử nghiệm với các cổng GPIO khác (GPIO2, GPIO3, GPIO4) và thay đổi thời gian nhấp nháy đèn LED.

Simulation

00:11.126 99%



```
rst:0x1 (POWERON),boot:0xc (SPI_FAST_FLASH_BOOT)
SPIWP:0xee
mode:DIO, clock div:1
load:0x3fcd6100,len:0x420
load:0x403ce000,len:0x90c
load:0x403d0000,len:0x2370
entry 0x403ce000
```

Simulation

00:29.623 99%

```

rst:0x1 (POWERON),boot:0xc (SPI_FAST_FLASH_BOOT)
SPIWP:0xee
mode:DIO, clock div:1
load:0x3fcd6100,len:0x420
load:0x403ce000,len:0x90c
load:0x403d0000,len:0x2370
entry 0x403ce000

```

- Cập nhật mã nguồn:
.global init

```

.equ GPIO_ENABLE_REG, 0x60004020
.equ GPIO_OUT_W1TS_REG, 0x60004008
.equ GPIO_OUT_W1TC_REG, 0x6000400C

```

init:

```
li a1, GPIO_ENABLE_REG # Thiết lập GPIO0 là chân output
```

```
li a2, 0x02    # Nạp mặt nạ GPIO 1, 2, 3, 4 lần lượt là 0x02, 0x04, 0x08, 0x10
sw a2, 0(a1)
```

```
main_loop:
```

```
li a1, GPIO_OUT_W1TS_REG # Thiết lập GPIO0 ở mức cao
li a2, 0x02    # Nạp mặt nạ GPIO 1, 2, 3, 4 lần lượt là 0x02, 0x04, 0x08, 0x10
sw a2, 0(a1)
call delay_asm      # Delay
```

```
li a1, GPIO_OUT_W1TC_REG # Xóa GPIO0 về mức thấp
li a2, 0x02    # Nạp mặt nạ GPIO 1, 2, 3, 4 lần lượt là 0x02, 0x04, 0x08, 0x10
sw a2, 0(a1)
call delay_asm      # Delay
```

```
j main_loop        # Loop
```

```
# Chương trình con delay, chờ một khoảng thời gian
```

```
delay_asm:
```

```
li a3, 0          # Giá trị biến đếm
li a4, 2000000     # Thời gian chờ (số lần đếm)
```

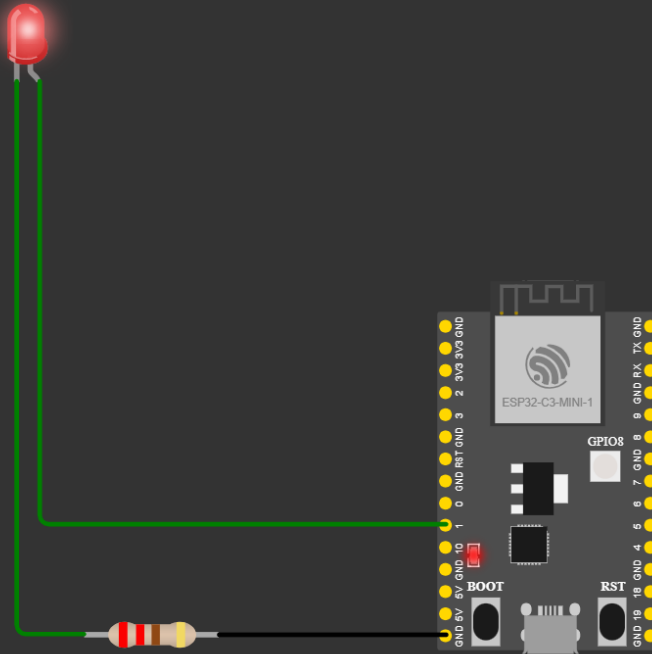
```
loop_delay:
```

```
addi a3, a3, 1
blt a3, a4, loop_delay
ret
```

Simulation



00:06.659 101%



rst:0x1 (POWERON),boot:0xc (SPI_FAST_FLASH_BOOT)

SPIWP:0xee

mode:DIO, clock div:1

load:0x3fcd6100,len:0x420

load:0x403ce000,len:0x90c

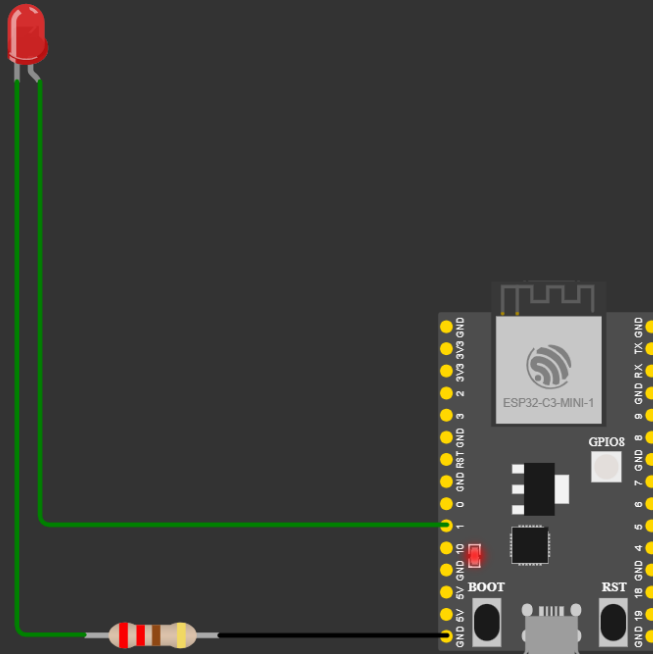
load:0x403d0000,len:0x2370

entry 0x403ce000



Simulation

00:15.546 100%



rst:0x1 (POWERON),boot:0xc (SPI_FAST_FLASH_BOOT)

SPIWP:0xee

mode:DIO, clock div:1

load:0x3fcd6100,len:0x420

load:0x403ce000,len:0x90c

load:0x403d0000,len:0x2370

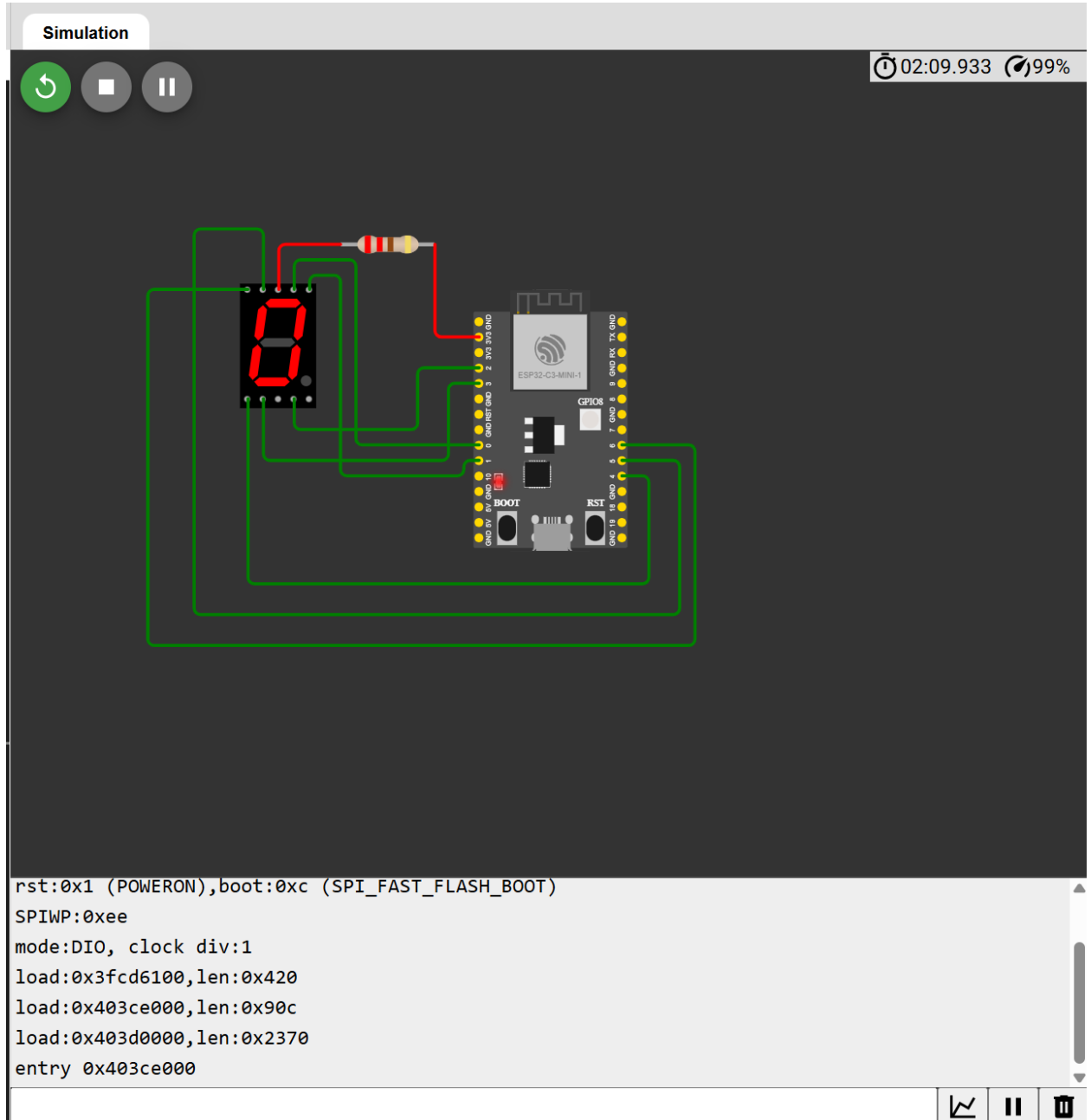
entry 0x403ce000



Assignment 3:

Tạo project để thực hiện và thử nghiệm Home Assignment 3. Cập nhật mã nguồn để hiển thị các chữ số khác nhau (từ 0 đến 9).

- Kết quả chạy thử:



- Sửa lại mã nguồn:

```
.global init
.equv GPIO_ENABLE_REG, 0x60004020  # Cho phép xuất tín hiệu các chân
GPIO
.equv GPIO_OUT_REG, 0x60004004      # Thiết lập mức logic đầu ra
.equv IO_MUX_GPIO4_REG, 0x60009014  # Cấu hình chức năng GPIO4
.equv IO_MUX_GPIO5_REG, 0x60009018  # Cấu hình chức năng GPIO5
.equv IO_MUX_GPIO6_REG, 0x6000901C  # Cấu hình chức năng GPIO6
.equv IO_MUX_GPIO7_REG, 0x60009020  # Cấu hình chức năng GPIO7 (nếu
dùng DP)
```

```
.text
```

```
init:
```

```
    # Kích hoạt các chân GPIO0-GPIO7 để xuất tín hiệu
    li a1, GPIO_ENABLE_REG
    li a2, 0xFF                # 11111111: Kích hoạt GPIO0-GPIO7 (a-g + dp)
    sw a2, 0(a1)
```

```
    # Thiết lập chức năng GPIO cho các chân GPIO4-GPIO7
```

```
    li a2, 0x1000              # Cấu hình chức năng GPIO
```

```
    li a1, IO_MUX_GPIO4_REG
```

```
    sw a2, 0(a1)
```

```
    li a1, IO_MUX_GPIO5_REG
```

```
    sw a2, 0(a1)
```

```
    li a1, IO_MUX_GPIO6_REG
```

```
    sw a2, 0(a1)
```

```
    li a1, IO_MUX_GPIO7_REG
```

```
    sw a2, 0(a1)
```

```
    li a1, GPIO_OUT_REG
```

```
    li a2, 0x92                # (0,1,2,3,4,5,6,7,8,9) = (0xC0, 0xF9, 0xA4, 0xB0, 0x99,
0x92, 0x82, 0xF8, 0x80, 0x90)
```

```
    sw a2, 0(a1)              # Xuất ra LED 7 đoạn
```

```
    # Lặp vô hạn để giữ trạng thái
```

```
loop:
```

```
    j loop
```

Simulation

01:21.406 102%

```
rst:0x1 (POWERON),boot:0xc (SPI_FAST_FLASH_BOOT)
SPIWP:0xee
mode:DIO, clock div:1
load:0x3fcd6100,len:0x420
load:0x403ce000,len:0x90c
load:0x403d0000,len:0x2370
entry 0x403ce000
```

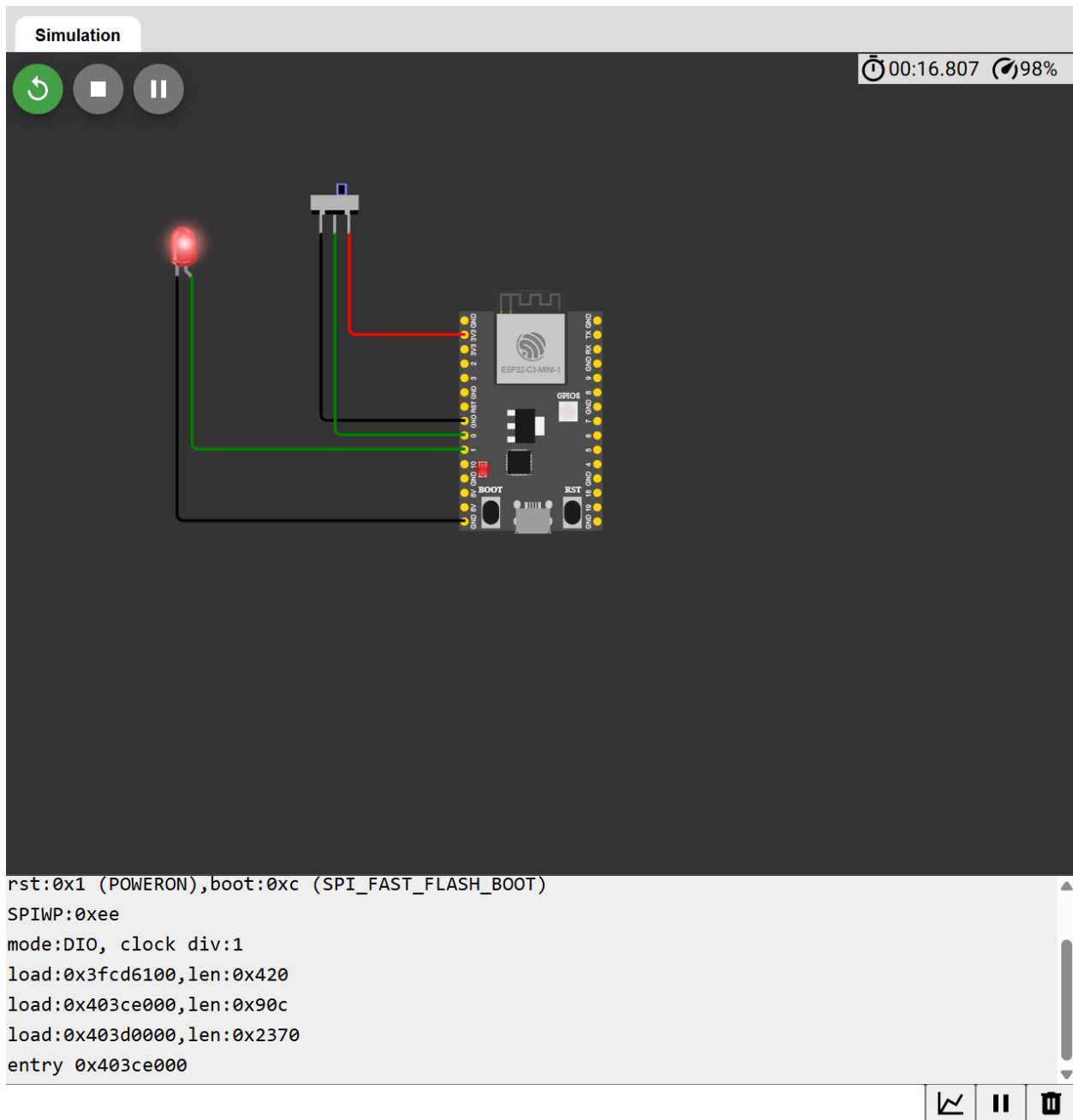
Assignment 4:

Tạo project để thực hiện và thử nghiệm Home Assignment 4. Cập nhật mã nguồn để sử dụng các cổng GPIO khác làm chân nhận tín hiệu (GPIO2, GPIO3, GPIO4).

Simulation

00:10.366 98%

```
rst:0x1 (POWERON),boot:0xc (SPI_FAST_FLASH_BOOT)
SPIWP:0xee
mode:DIO, clock div:1
load:0x3fcd6100,len:0x420
load:0x403ce000,len:0x90c
load:0x403d0000,len:0x2370
entry 0x403ce000
```



- Sửa lại mã nguồn:

```
.global init
```

```
.eqv GPIO_OUT_W1TS_REG, 0x60004008 # Thanh ghi thiết lập
```

```
.eqv GPIO_OUT_W1TC_REG, 0x6000400C # Thanh ghi xóa
```

```
.eqv GPIO_ENABLE_REG, 0x60004020 # Thanh ghi cho phép xuất tín hiệu
```

```
.eqv GPIO_IN_REG, 0x6000403C # Thanh ghi đọc trạng thái GPIO
```

```
.eqv IO_MUX_GPIO0_REG, 0x60009004 # Thanh ghi thiết lập chức năng
GPIO0
.eqv IO_MUX_GPIO2_REG, 0x6000900C # Thanh ghi thiết lập chức năng
GPIO2
.eqv IO_MUX_GPIO3_REG, 0x60009010 # Thanh ghi thiết lập chức năng
GPIO3
.eqv IO_MUX_GPIO4_REG, 0x60009014 # Thanh ghi thiết lập chức năng
GPIO4
```

init:

```
li a1, GPIO_ENABLE_REG # Thiết lập GPIO1 là chân xuất tín hiệu
li a2, 0x02
sw a2, 0(a1)
```

li a1, IO_MUX_GPIO2_REG # Thiết lập cho phép GPIOx (x = 2, 3, 4 thay thế
tùy theo ý định) nhận tín hiệu

```
lw a2, 0(a1)
ori a2, a2, 0x200 # Thiết lập bit IO_MUX_GPIO0_FUN_IE
sw a2, 0(a1)
```

loop:

```
li a1, GPIO_IN_REG # Đọc trạng thái các chân GPIOx
lw a2, 0(a1)
andi a3, a2, 0x04 # Kiểm tra mức tín hiệu GPIOx (0x04, 0x08, 0x10)
beq a3, zero, clear # Nếu GPIO0 = 0 => Tắt LED
```

set:

```
li a1, GPIO_OUT_W1TS_REG # Bật LED: Thiết lập GPIO1 = 1
li a2, 0x02
sw a2, 0(a1)
j next
```

clear:

```
li a1, GPIO_OUT_W1TC_REG # Tắt LED: Xóa GPIO1 = 0
li a2, 0x02
sw a2, 0(a1)
```

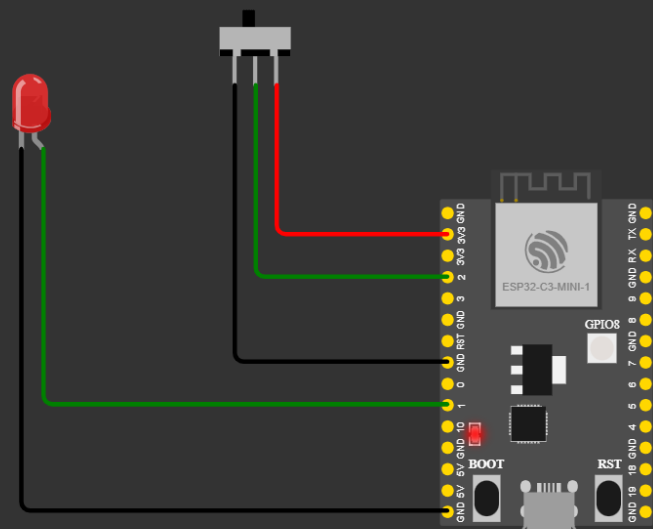
next:

```
j loop # Loop
```

- Kết quả:

Simulation

00:11.275 101%



The diagram shows an ESP32-C3-MINI-1 microcontroller board with the following connections:

- A red LED is connected to pin 1 (GPIO 1) via a green wire and to pin 10 (5V) via a black wire.
- A 3-pin header is connected to pins 2 (GPIO 2), 3 (GPIO 3), and 0 (GND) via green, red, and black wires respectively.

```
rst:0x1 (POWERON),boot:0xc (SPI_FAST_FLASH_BOOT)
SPIWP:0xee
mode:DIO, clock div:1
load:0x3fcd6100,len:0x420
load:0x403ce000,len:0x90c
load:0x403d0000,len:0x2370
entry 0x403ce000
```

Simulation

00:17.802 86%

```
rst:0x1 (POWERON),boot:0xc (SPI_FAST_FLASH_BOOT)
SPIWP:0xee
mode:DIO, clock div:1
load:0x3fcd6100,len:0x420
load:0x403ce000,len:0x90c
load:0x403d0000,len:0x2370
entry 0x403ce000
```

Assignment 5:

```
.global init
.equ GPIO_ENABLE_REG, 0x60004020  # Cho phép xuất tín hiệu các chân
GPIO
.equ GPIO_OUT_REG, 0x60004004      # Thiết lập mức logic đầu ra
.equ IO_MUX_GPIO4_REG, 0x60009014  # Thiết lập chức năng chân GPIO4
.equ IO_MUX_GPIO5_REG, 0x60009018  # Thiết lập chức năng chân GPIO5
.equ IO_MUX_GPIO6_REG, 0x6000901C  # Thiết lập chức năng chân GPIO6
.equ DELAY_COUNT, 5000000          # Hằng số cho độ trễ

.data
# Common Anode: 0 = bật đoạn, 1 = tắt đoạn
digit_patterns:
.word 0xC0 # 0: 11000000 (a,b,c,d,e,f bật - active low)
.word 0xF9 # 1: 11111001 (b,c bật)
.word 0xA4 # 2: 10100100 (a,b,d,e,g bật)
.word 0xB0 # 3: 10110000 (a,b,c,d,g bật)
.word 0x99 # 4: 10011001 (b,c,f,g bật)
.word 0x92 # 5: 10010010 (a,c,d,f,g bật)
.word 0x82 # 6: 10000010 (a,c,d,e,f,g bật)
.word 0xF8 # 7: 11111000 (a,b,c bật)
.word 0x80 # 8: 10000000 (tắt cả đoạn bật)
.word 0x90 # 9: 10010000 (a,b,c,d,f,g bật)

.text
init:
    # Kích hoạt các chân GPIO0-GPIO6 để xuất tín hiệu
    li a1, GPIO_ENABLE_REG
    li a2, 0x7F          # 01111111: Kích hoạt GPIO0-GPIO6 (a-g)
    sw a2, 0(a1)

    # Thiết lập chức năng GPIO cho các chân GPIO4-GPIO6 (e,f,g)
    li a2, 0x1000        # Cấu hình chức năng GPIO
    li a1, IO_MUX_GPIO4_REG
    sw a2, 0(a1)
    li a1, IO_MUX_GPIO5_REG
    sw a2, 0(a1)
```



```
li a1, IO_MUX_GPIO6_REG
sw a2, 0(a1)
```

Chuẩn bị hiển thị các chữ số

```
la a3, digit_patterns    # Địa chỉ mảng mẫu bit
li a4, 10                # Số lượng chữ số (0-9)
```

display_loop:

```
li a5, 0                # Chỉ số chữ số (0-9)
```

digit_loop:

```
slli t0, a5, 2          # Nhân chỉ số với 4 (kích thước word)
add t0, a3, t0           # Địa chỉ mẫu bit cho chữ số hiện tại
lw a2, 0(t0)            # Lấy mẫu bit
li a1, GPIO_OUT_REG
sw a2, 0(a1)            # Xuất mẫu bit ra GPIO0-GPIO6
```

Độ trễ để hiển thị chữ số

```
li t1, DELAY_COUNT
```

delay:

```
addi t1, t1, -1
bnez t1, delay
```

Tăng chỉ số chữ số

```
addi a5, a5, 1
```

```
blt a5, a4, digit_loop  # Lặp lại cho đến khi hiển thị hết 0-9
```

```
j display_loop          # Lặp lại vô hạn
```

- Kết quả:

Simulation

01:50.835 99%

Stop the simulation

```
rst:0x1 (POWERON),boot:0xc (SPI_FAST_FLASH_BOOT)
SPIWP:0xee
mode:DIO, clock div:1
load:0x3fcd6100,len:0x420
load:0x403ce000,len:0x90c
load:0x403d0000,len:0x2370
entry 0x403ce000
```

Simulation

01:46.569 100%

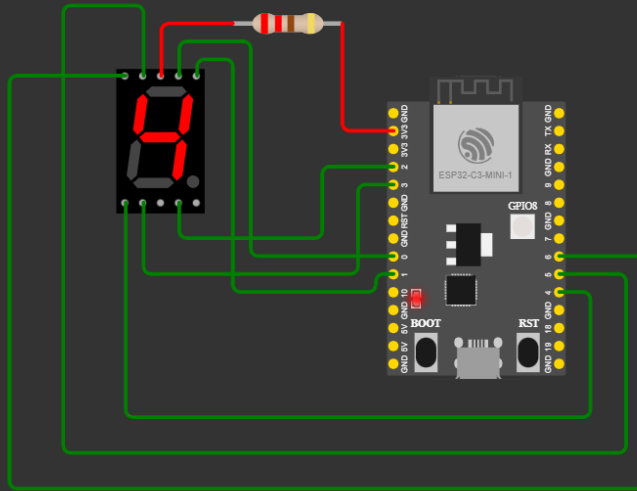
Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Markup and share

Simulation

01:46.569 100%



rst:0x1 (POWERON),boot:0xc (SPI_FAST_FLASH_BOOT)

SPIWP:0xee

mode:DIO, clock div:1

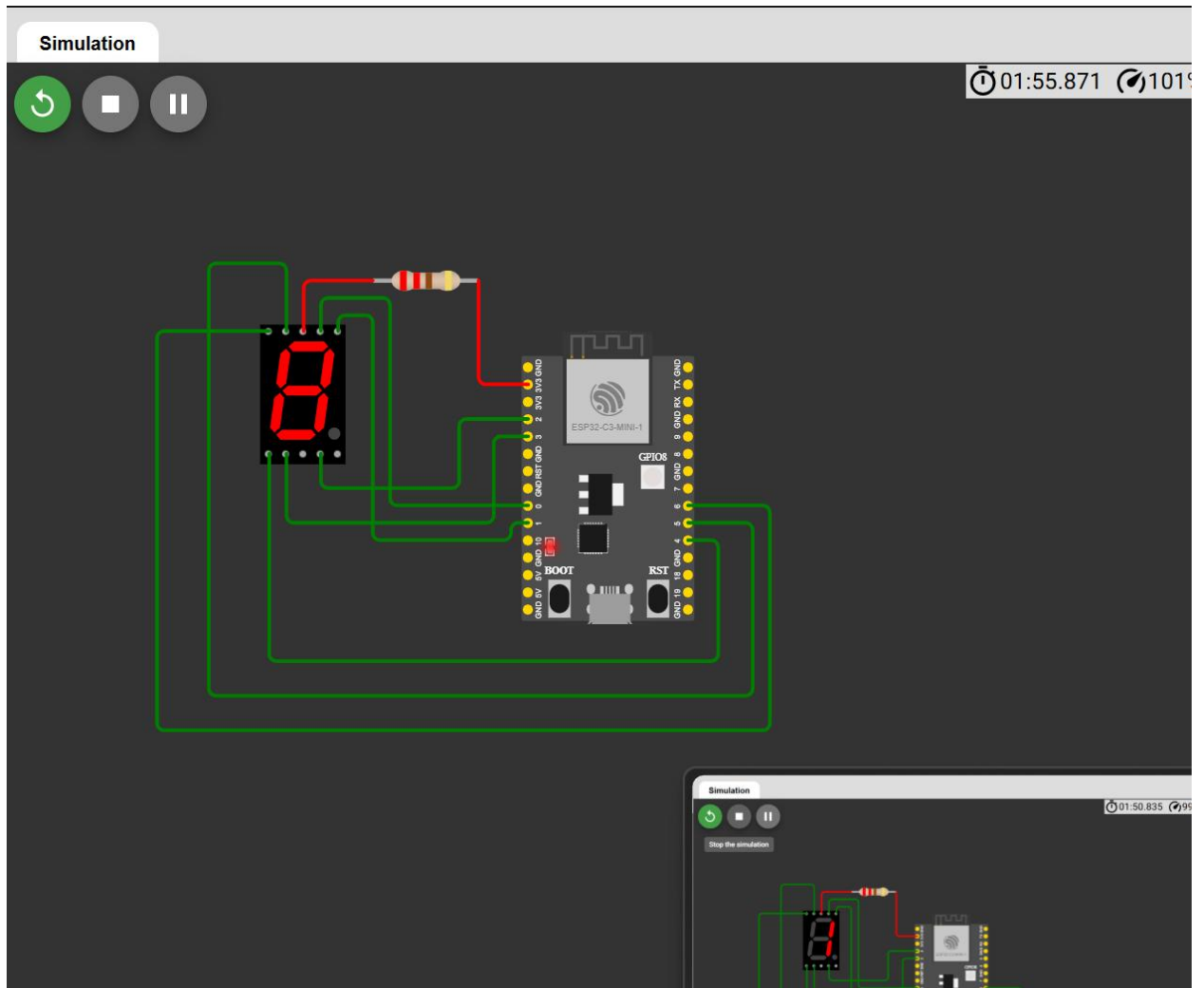
load:0x3fcd6100,len:0x420

load:0x403ce000,len:0x90c

load:0x403d0000,len:0x2370

entry 0x403ce000





rst:0x1 (POWERON),boot:0xc (SPI_FAST_FLASH_BOOT)

SPIWP:0xee

mode:DIO, clock div:1

load:0x3fcd6100,len:0x420

load:0x403ce000,len:0x90c

load:0x403d0000,len:0x2370

entry 0x403ce000

