# THỰC HÀNH VI XỬ LÝ – VI ĐIỀU KHIỂN

GVHD: Trần Hoàng Lộc

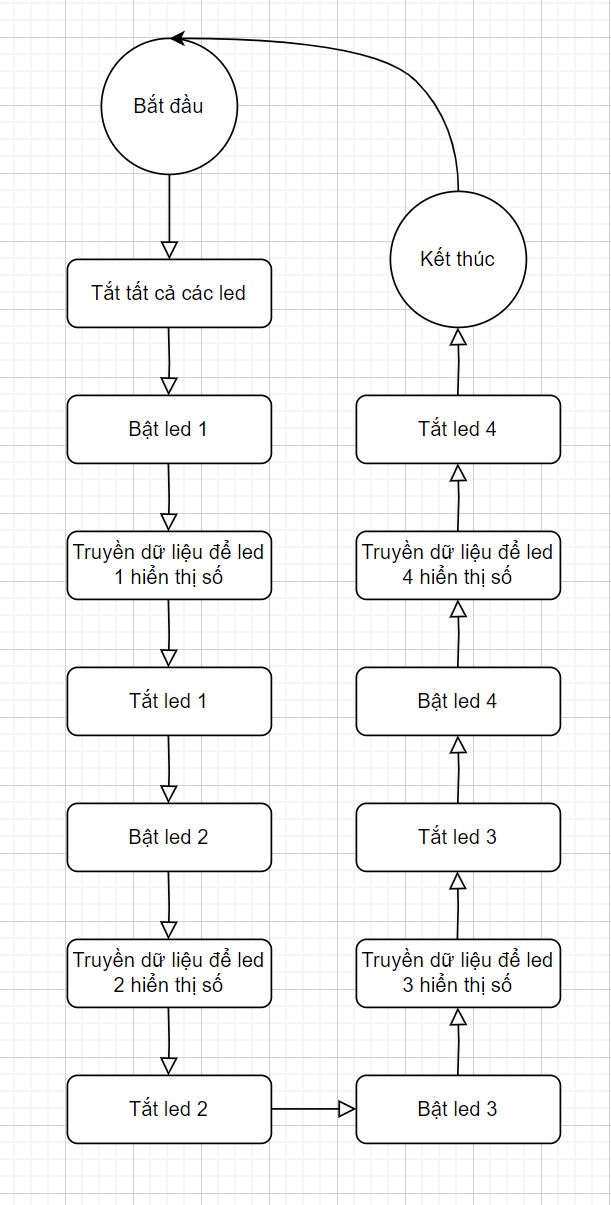
Họ và tên sinh viên thực hiện: Nhâm bảo Minh

Mã số sinh viên: 22520881

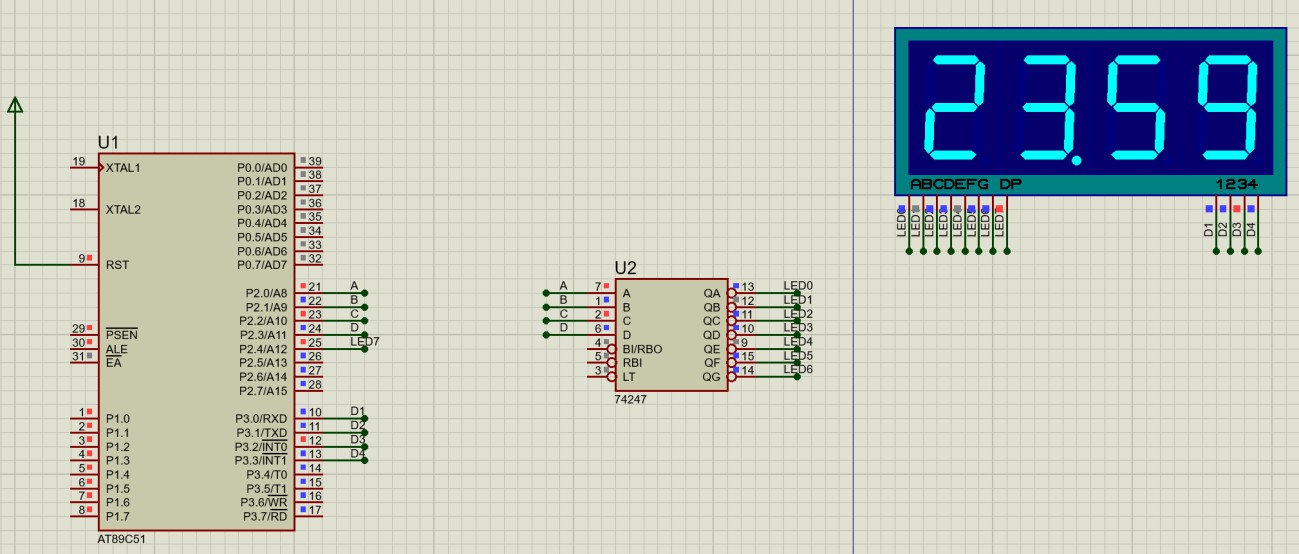
# BÁO CÁO THỰC HÀNH SỐ 4

<GIAO TIẾP VỚI 7-SEGMENT LED VÀ TIMER>

1. **Nội dung I**
2. **Nội dung II**
3. **Bài tập**
4. **Tài liệu tham khảo**
5. **Báo cáo**
   * Thuật toán quét LED áp dụng cho màn hình LED 7 đoạn:
     + Tất cả cả các LED 7 đều chung 1 đường dữ liệu, trong 1 thời điểm chỉ có 1 LED sáng.
     + Đầu tiên, Led 1 bật, các led khác đều tắt, lúc này ta truyền dữ liệu hiển thị 1 số nào đó để led 1 hiện số.
     + Thời điểm sau, led 2 bật, các led khác tắt, lúc này truyền dữ liệu để led 2 hiển thị số.
     + Thời điểm tiếp theo, led 3 bật, các led khác tắt, truyền dữ liệu để led 3 hiển thị số.
     + Cuối cùng, led 4 bật và các led còn lại tắt, lúc nay truyền dữ liệu để led 4 hiển thị đèn.
       - Như vậy mỗi đèn hiển thị 1 số khác nhau, nếu quá trình trên diễn ra với tốc độ đủ nhanh, mắt chúng ta không còn phân biệt được đèn có tắt hay không (hiện tượng lưu ảnh trên võng mạc) thì ta sẽ thấy số trên 4 led sáng liên tục không gián đoạn (mặc dù trong thực tế trong 1 thời điểm chỉ có 1 đèn sáng.
   * Sơ đồ thuật toán:



* + Shematic Capture:



* + Mạch bao gồm chip 8051, LED 7 đoạn dương chung và IC 74247.
  + Chân P3.0, P3.1, P3.2, P3.3 của 8051 sẽ nối chân 1, 2, 3, 4 của LED 7 đoạn.
  + Chân P2.0, P2.1, P2.2, P2.3 của 8051 sẽ nối với chân A, B, C, D của IC 74247; chân P2.4 sẽ nối với cân DP của LED 7 đoạn.
  + Chân QA, QB, QC, QD, QE, QF, QG của IC 74247 nối với chân A, B, C, D, E,

F, G của LED 7 đoạn.

* + Truyền giá trị vào cổng P3 để bật các LED của LED 7 đoạn.
  + IC 74247 giúp truyền dữ liệu vào LED dễ dàng hơn
  + Chương trình tạo đồng hồ đếm 24h, sử dụng Timer trong 8051:

;=================================================

===================

; Main.asm file generated by New Project wizard

;

; Created: Wed Mar 20 2024

; Processor: AT89C51

; Compiler: ASEM-51 (Proteus)

;=================================================

===================

$NOMOD51

$INCLUDE (8051.MCU)

;=================================================

===================

; DEFINITIONS

;=================================================

===================

;=================================================

===================

; VARIABLES

;=================================================

===================

;=================================================

===================

; RESET and INTERRUPT VECTORS

;=================================================

===================

; Reset Vector org 0000h

jmp Start

;=================================================

===================

; CODE SEGMENT

;=================================================

===================

org 0100h

;=================================================

=================== DELAY: ;{

mov R5, #185 Delay\_Loop1: ;{

nop

djnz R5, Delay\_Loop1

;}

ret

;}

;=================================================

===================

;=================================================

===================

Delay\_5ms: ;{

mov TMOD, #01H mov TH0, #0EEH mov TL0, #00H

setb TR0 Wait\_loop: ;{

jnb TF0, Wait\_loop clr TF0

clr TR0 ret

;}

;}

;=================================================

===================

;=================================================

===================

Delay\_1s: ;{

mov R7, #100

Loop1\_Delay\_1s: ;{

call SetUp

call Delay\_5ms call SetUp

djnz R7, Loop1\_Delay\_1s

;}

ret

;}

;=================================================

===================

;=================================================

===================

SetUp: ;{

;set up hour led ;{ mov A, R0

mov B, #10 div AB

mov P3, #1 mov P2, A setb P2.4 call DELAY

mov P3, #0

mov P3, #2 mov P2, B call DELAY

;}

;set up minute led ;{ mov A, R1

mov B, #10 div AB

mov P3, #0

mov P3, #4 mov P2, A setb P2.4 call DELAY

mov P3, #0

mov P3, #8 mov P2, B setb P2.4 call DELAY

;}

ret

;}

;=================================================

===================

;=================================================

===================

Start: ;{

mov R0, #23 ;R0: hour mov R1, #59 ;R1: minute mov R2, #00 ;R2: second

Loop1\_Start: ;{

call SetUp call Delay\_1s

inc R2

cjne R2, #60, Continue

;second = 60 ;{

mov R2, #0 inc R1

cjne R1, #60, Continue

;minute = 60 ;{

mov R1, #0 inc R0

cjne R0, #24, Continue

;hour = 24 ;{

mov R0, #0

;}

;}

;}

Continue:

sjmp Loop1\_Start

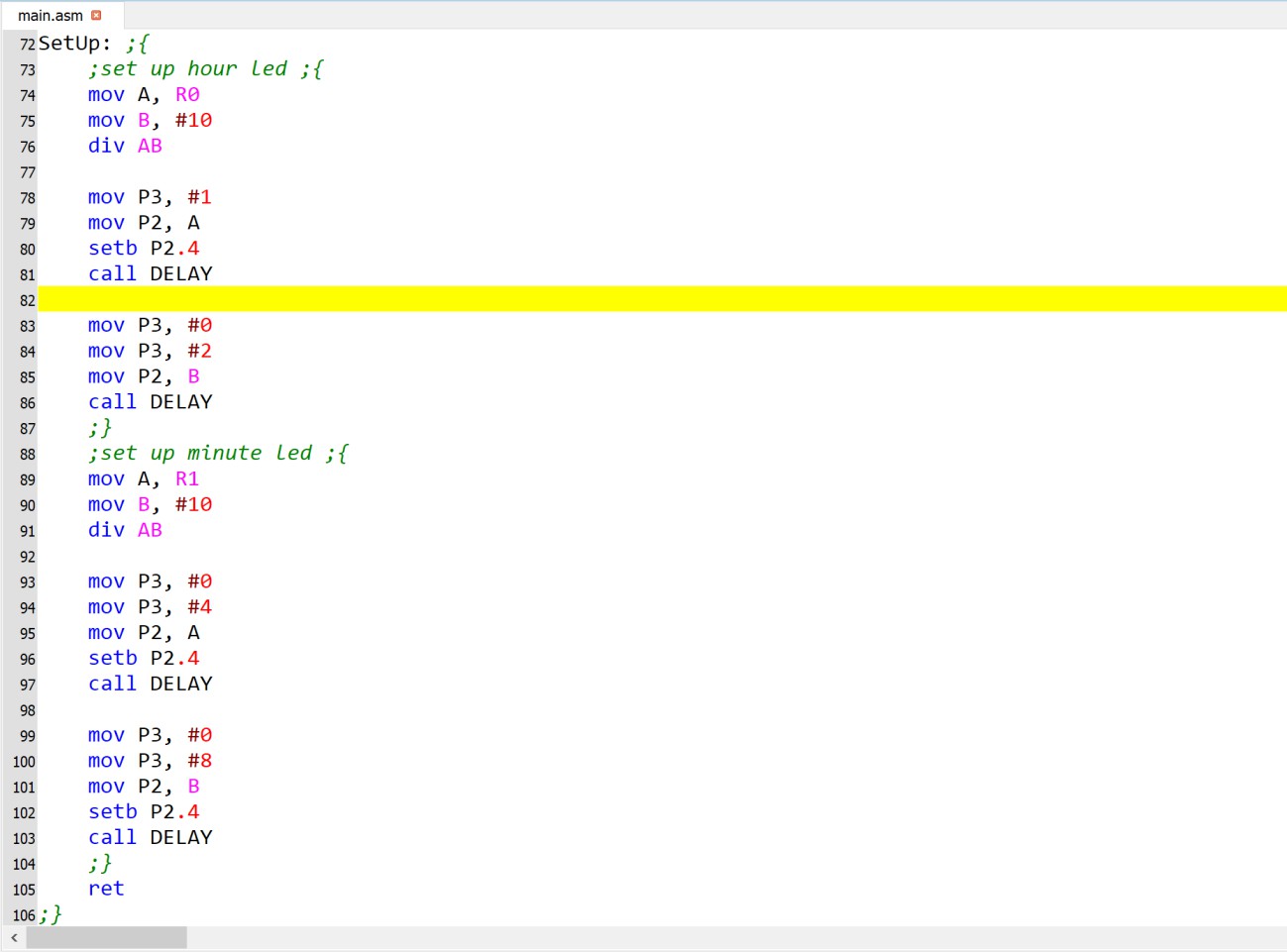
;}

;}

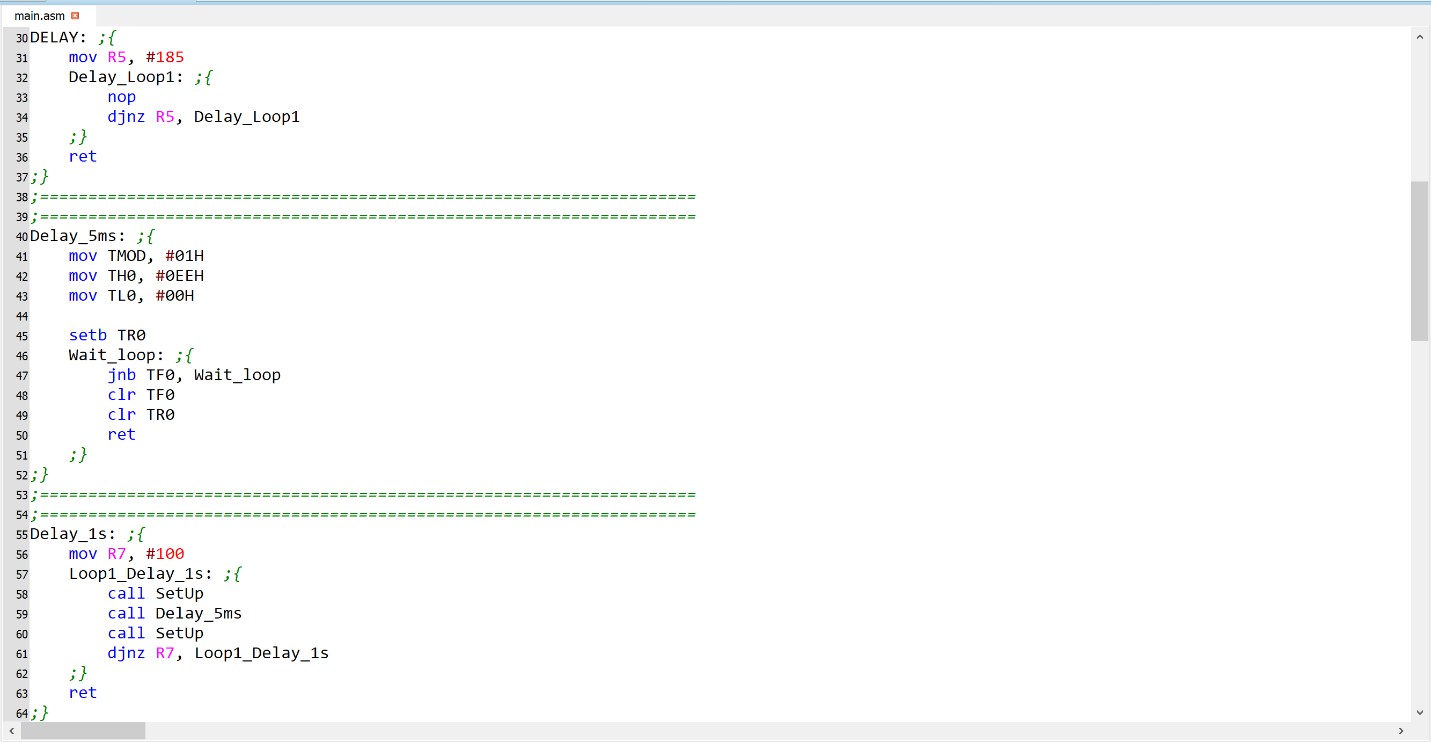
;=================================================

=================== END

* Ý tưởng chương trình:
  + Sử dụng thuật toán quét LED, kết hợp với Timer của 8051
  + Thanh ghi R0 lưu giờ hiện tại, R1 lưu giá phút hiện tại và R2 lưu giây hiện tại.
  + Cứ mỗi giây trôi qua thì tăng R2 lên 1, nếu R2 bằng 60 thì gán 0 vào R2, tăng R1 lên 1; R1 bằng 60 thì gán 0 vào R1, tăng R0 lên 1; R0 bằng 24 thì gán 0 cho R0.
  + Dựa vào giá trị R0, R1, R2 để hiện thời gian hiện tại ra LED.
  + LED 1 của LED 7 đoạn sẽ hiện giá trị của (R0/10), LED 2 sẽ hiện giá trị của (R0%10); LED 3 sẽ hiện giá trị (R1/10) và LED 4 sẽ hiện giá trị (R1%10).



* SetUp sẽ thực hiện quét LED, sử dụng các thanh ghi R0, R1 để hiện thời gian hiện tại



* Hàm DELAY để tạo độ trễ cho quét LED.
* Delay\_5ms sử dụng Timer của 8051 để tạo độ trễ 5ms.
* Trong Delay\_1s sẽ quét LED, kết hợp với Delay\_5ms.



* Trong Start sẽ có 1 vòng lặp để thự hiện đếm và hiện thời gian.
* Chương trình tạo đồng hồ đếm 24h, sử dụng vòng lặp thay vì Timer của 8051:

;=================================================

===================

; Main.asm file generated by New Project wizard

;

; Created: Wed Mar 20 2024

; Processor: AT89C51

; Compiler: ASEM-51 (Proteus)

;=================================================

===================

$NOMOD51

$INCLUDE (8051.MCU)

;=================================================

===================

; DEFINITIONS

;=================================================

===================

;=================================================

===================

; VARIABLES

;=================================================

===================

;=================================================

===================

; RESET and INTERRUPT VECTORS

;=================================================

===================

; Reset Vector org 0000h

jmp Start

;=================================================

===================

; CODE SEGMENT

;=================================================

===================

org 0100h

;=================================================

=================== DELAY: ;{

mov R5, #185 Delay\_Loop1: ;{

nop

djnz R5, Delay\_Loop1

;}

ret

;}

;=================================================

===================

;=================================================

===================

Delay\_5ms: ;{

mov R6, #255 Delay\_5ms\_Loop1:

mov R7, #184 Delay\_5ms\_Loop2:

djnz R7, Delay\_5ms\_Loop2 djnz R6, Delay\_5ms\_Loop1 ret

;}

;=================================================

===================

;=================================================

===================

Delay\_1s: ;{

mov R7, #100 Loop1\_Delay\_1s: ;{

call SetUp

call Delay\_5ms call SetUp

djnz R7, Loop1\_Delay\_1s

;}

ret

;}

;=================================================

===================

;=================================================

===================

SetUp: ;{

;set up hour led ;{ mov A, R0

mov B, #10 div AB

mov P3, #1 mov P2, A setb P2.4 call DELAY

mov P3, #0

mov P3, #2 mov P2, B call DELAY

;}

;set up minute led ;{ mov A, R1

mov B, #10 div AB

mov P3, #0

mov P3, #4 mov P2, A setb P2.4 call DELAY

mov P3, #0

mov P3, #8 mov P2, B setb P2.4 call DELAY

;}

ret

;}

;=================================================

===================

;=================================================

===================

Start: ;{

mov R0, #23 ;R0: hour mov R1, #59 ;R1: minute mov R2, #00 ;R2: second

Loop1\_Start: ;{

call SetUp call Delay\_1s

inc R2

cjne R2, #60, Continue

;second = 60 ;{

mov R2, #0 inc R1

cjne R1, #60, Continue

;minute = 60 ;{

mov R1, #0 inc R0

cjne R0, #24, Continue

;hour = 24 ;{

mov R0, #0

;}

;}

;}

Continue:

sjmp Loop1\_Start

;}

;}

;=================================================

=================== END

* Với cách sử dụng Timer của 8051:
  + Timer trong 8051 dễ sử dụng, sử dụng công thức tính toán làm giảm đi thời gian tính tay phải chạy bao nhiêu chu kỳ để đủ 1 khoảng thời gian. Code ngắn gọn, dễ hiểu hơn nhiều, có nhiều mode sử dụng linh hoạt, dễ Debug.
  + Tuy nhiên phải điều chỉnh tần số thạch anh chính xác, nhiều lúc có thể sử dụng Timer không đúng cách.
* Với cách sử dụng vòng lặp thay cho Timer:
  + Cách làm dễ hiểu, cũng dễ code và Debug, code mạch lạc, dễ dàng tái sử dụng cho những lần sau.
  + Dễ code nhưng khá dài dòng, phải tính tay nhiều để tìm ra số chu kỳ để chạy trong 1 khoảng thời gian. Thời gian Delay nhiều thì code dài dòng, làm tốn sức.