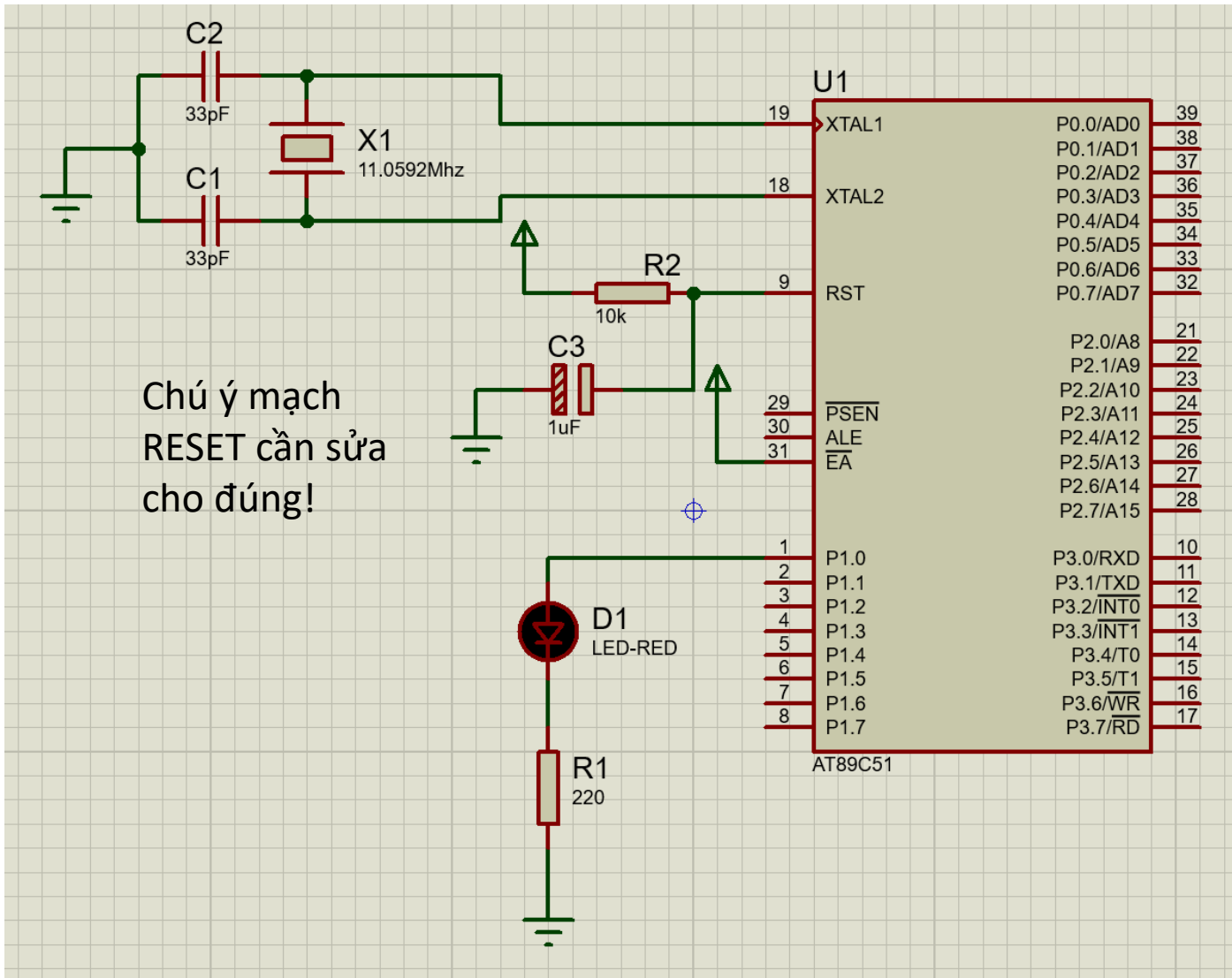


Lập trình 8051 (89S51/52)

# HelloWorld



**ORG 000**

**AGAIN:**

**SETB P1.0**

**ACALL DELAY**

**CLR P1.0**

**ACALL DELAY**

**SJMP AGAIN**

**DELAY:**

**MOV R1,#255**

**LOOP1:**

**MOV R2,#255**

**LOOP2:**

**MOV R3,#2**

**LOOP3:**

**DJNZ R3,LOOP3**

**DJNZ R2,LOOP2**

**DJNZ R1,LOOP1**

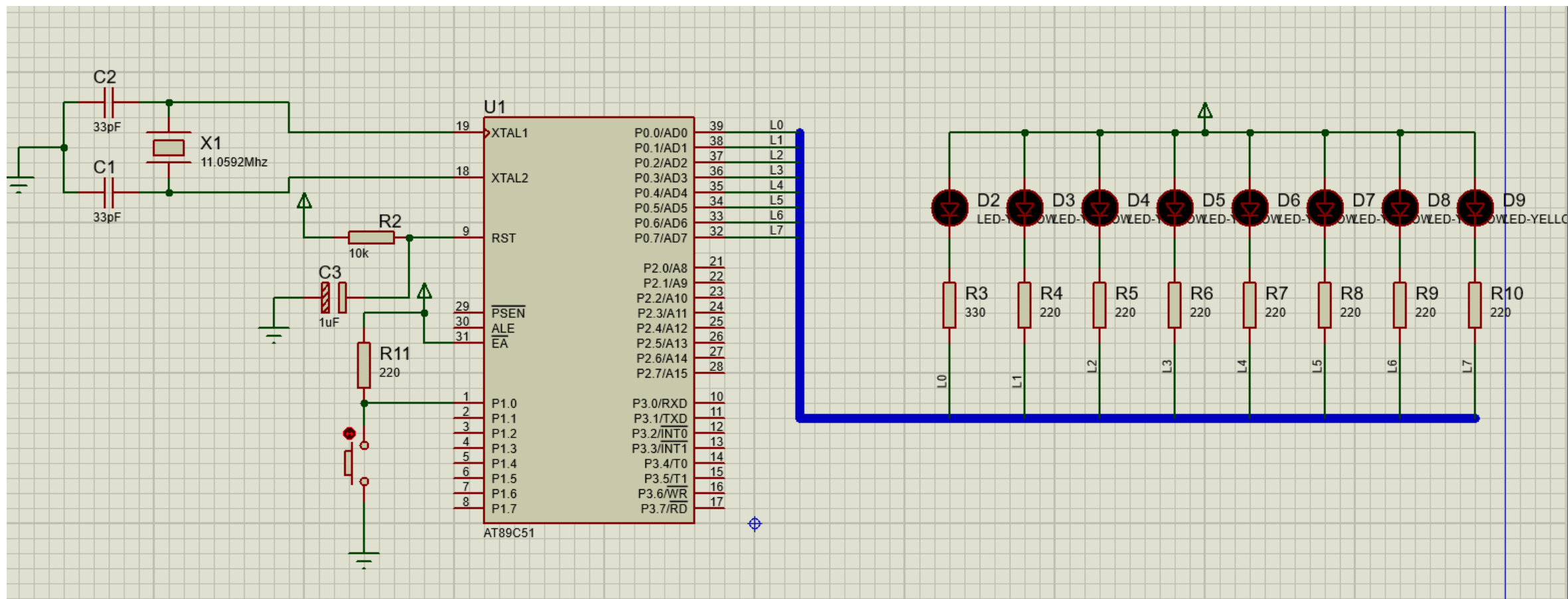
**RET**

**END**

# VD2: LED + button

- Thiết kế + lập trình mạch
  - 1 nút bấm nối với P1.0 và 8 đèn LED nối với P0
  - Khi bấm nút thì đèn sáng luân phiên 4 chẵn 4 lẻ, nhả nút thì ngừng hoạt động

# Sơ đồ mạch



# Code

```
#include<reg51.h>

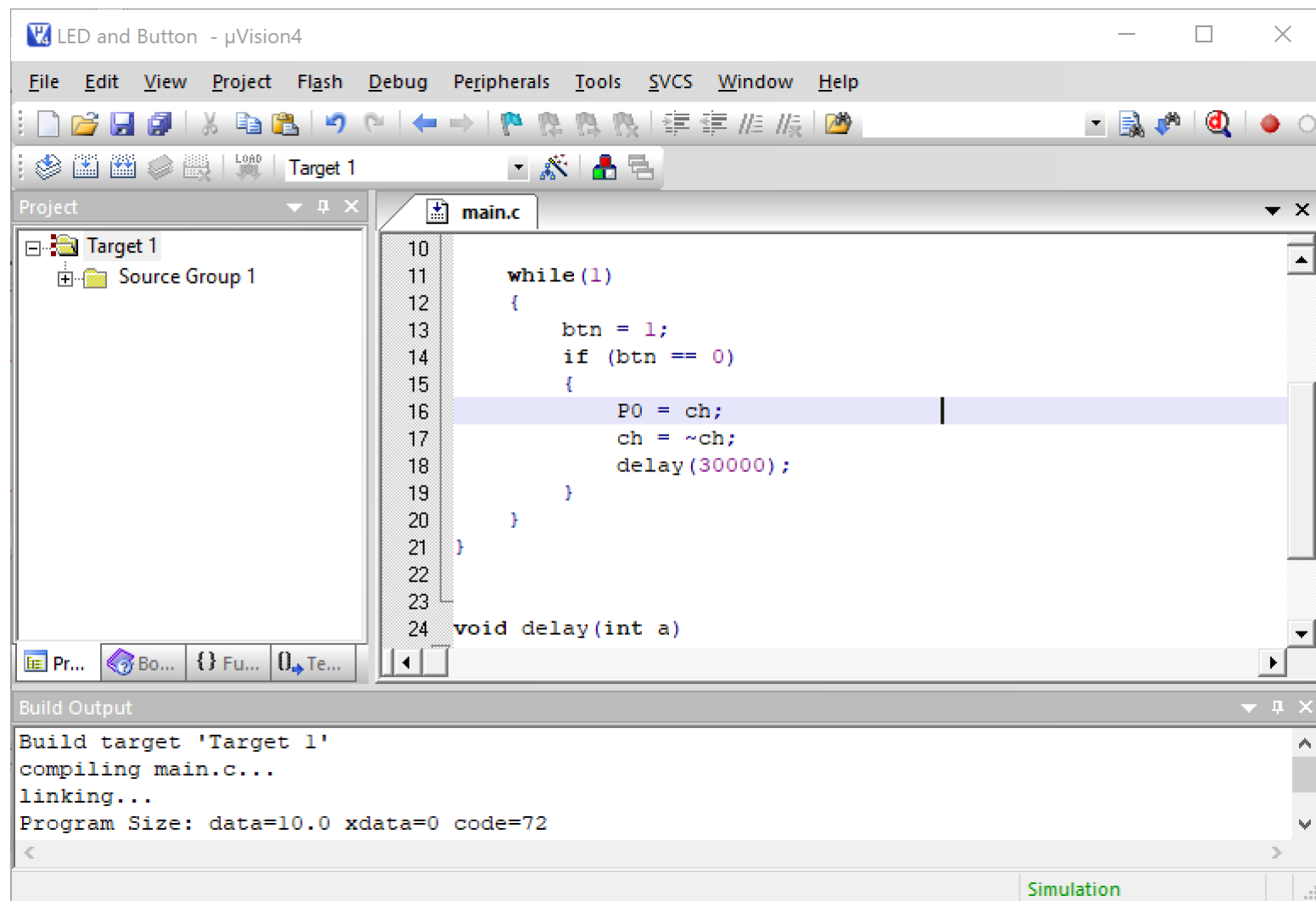
void delay(int);
sbit btn = P1^0;

int main(void)
{
    char ch = 0x55;
    P0 = 0x00;

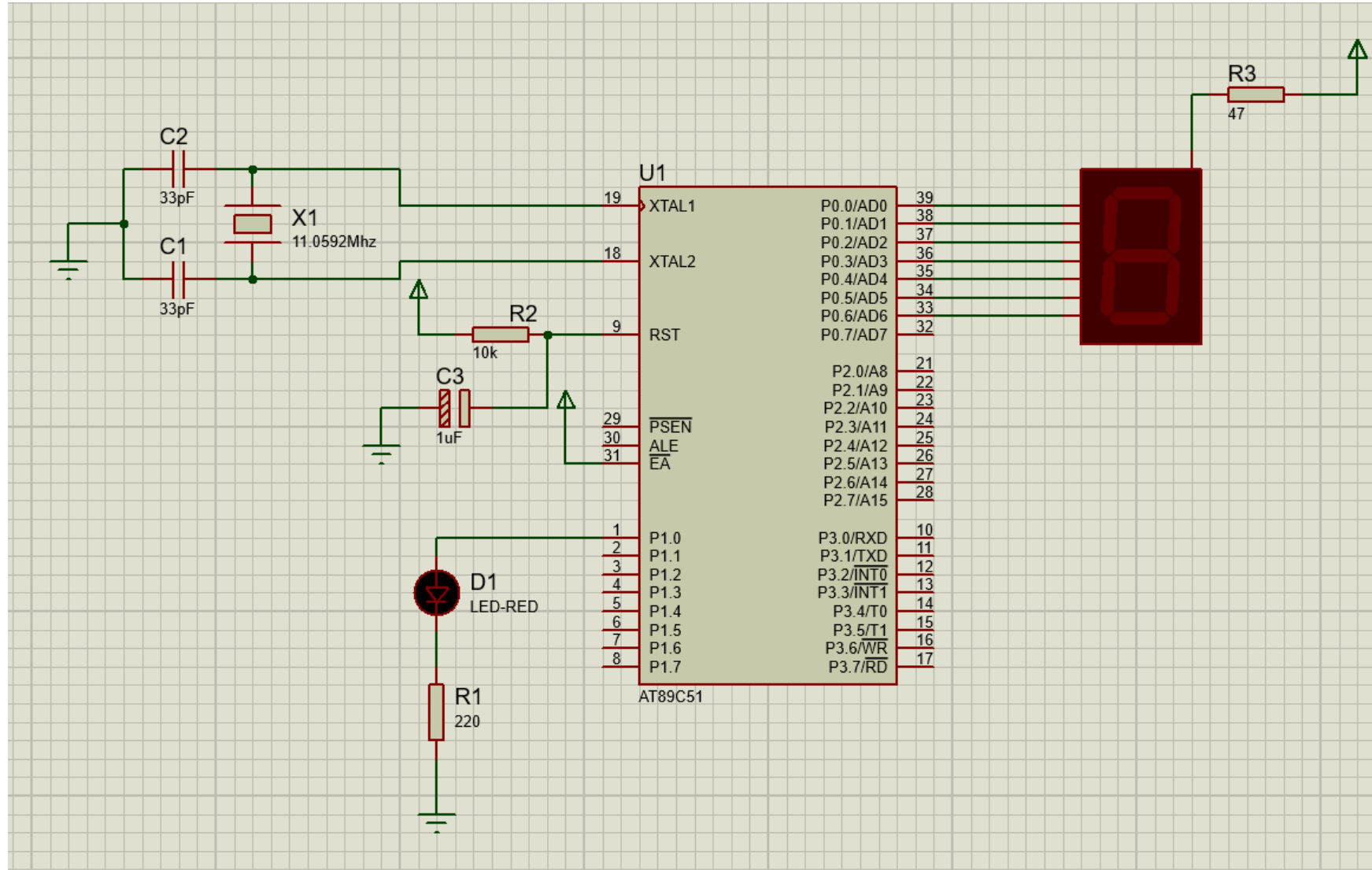
    while(1)
    {
        btn = 1;
        if (btn == 0)
        {
            P0 = ch;
            ch = ~ch;
            delay(30000);
        }
    }
}
```

```
void delay(int a)
{
    int i;
    for(i=0;i<a;i++);
}
```

# Sử dụng Keil C51



# 7-seg LED



# Code

```
void Display(char ch)
{
    switch(ch)
    {
        case '0':    P0 = ~0x3F;    break;
        case '1':    P0 = ~0x06;    break;
        case '2':    P0 = ~0x5B;    break;
        case '3':    P0 = ~0x4F;    break;
        case '4':    P0 = ~0x66;    break;
        case '5':    P0 = ~0x6D;    break;
        case '6':    P0 = ~0x7D;    break;
        case '7':    P0 = ~0x07;    break;
        case '8':    P0 = ~0x7F;    break;
        case '9':    P0 = ~0x6F;    break;

        default: P2 = 0x3F;    break;
    }
}
```

```
int main(void)
{
    char ch = '0';
    while(1)
    {
        Display(ch);
        delay(30000);

        ch++;
        if (ch==0x3A)
            ch = '0';
    }
}
```



# Lập trình ngắn

- 89S52 có 6 nguồn ngắt: 2 ngắt ngoài, 3 ngắt của timer, 1 ngắt của UART

(MSB)		(LSB)					
EA	–	ET2	ES	ET1	EX1	ET0	EX0
Enable Bit = 1 enables the interrupt.							
Enable Bit = 0 disables the interrupt.							

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
–	IE.6	Reserved.
ET2	IE.5	Timer 2 interrupt enable bit.
ES	IE.4	Serial Port interrupt enable bit.
ET1	IE.3	Timer 1 interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.

User software should never write 1s to reserved bits, because they may be used in future AT89 products.

# Keil C51 interrupt number

No.	Source	Address
0	EXTERNAL INT 0	0003h
1	TIMER/COUNTER 0	000Bh
2	EXTERNAL INT 1	0013h
3	TIMER/COUNTER 1	001Bh
4	SERIAL PORT	0023h
5	TIMER/COUNTER 2 (8052)	002Bh

# Lập trình ngắt với Keil 8051

```
void main()
{
    init();
    ...
}
```

```
void init()
{
    EA=1;
    EX0=1;
}
```

```
void EX0_ISR() interrupt 0
{
    ...
}
```

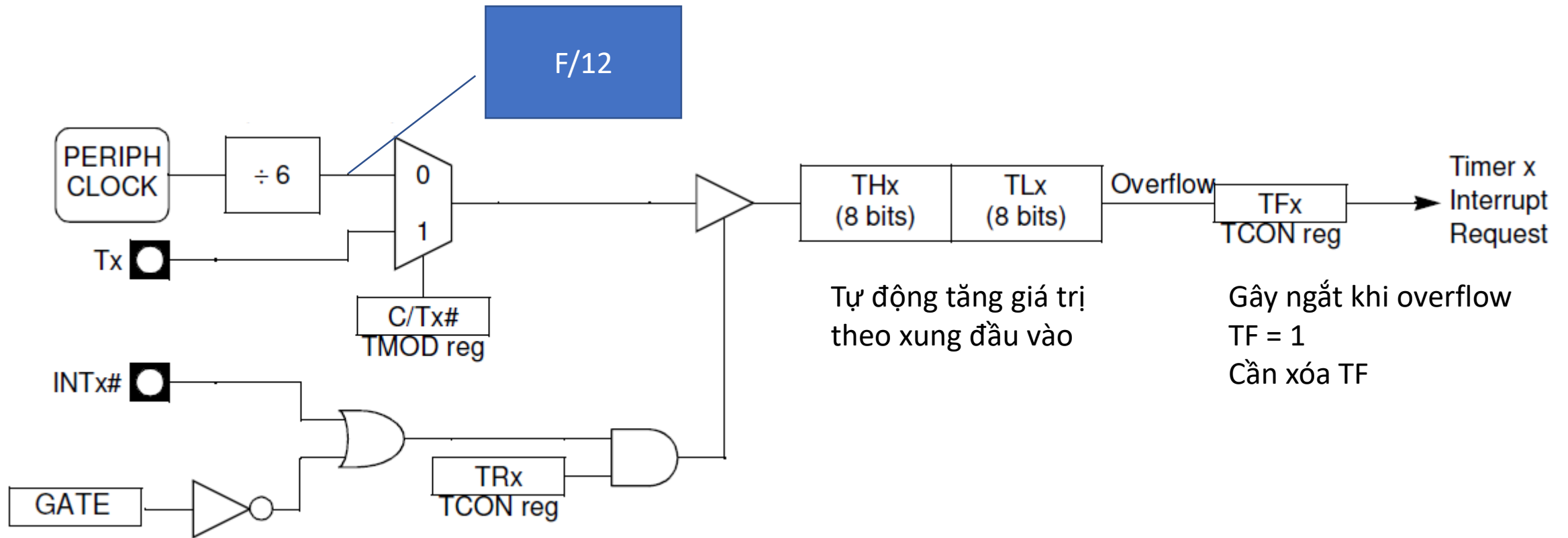
# Ví dụ: Ngắt ngoài

- Viết chương trình
- Hiển thị số trên bảng LED 7 thanh
- Xử lý ngắt ngoài 0 và 1 để tăng/giảm giá trị trên bảng LED

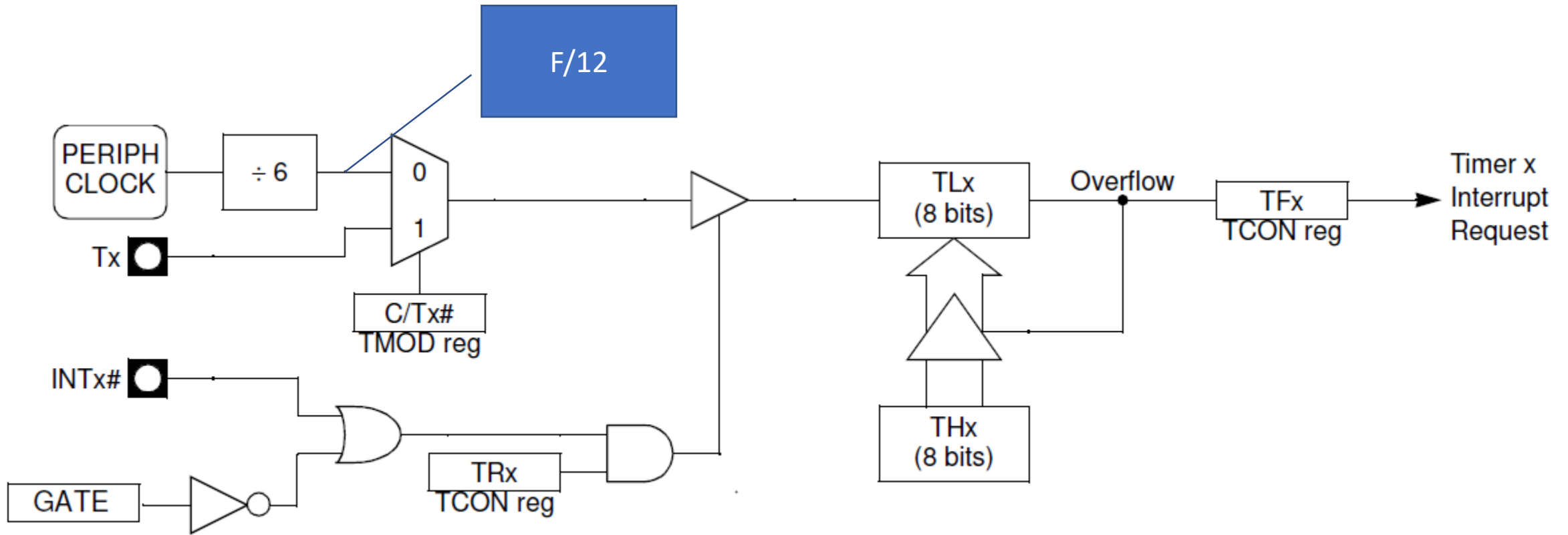
# Timer/counter

- 80x51: timer 0 + timer 1
- 80x52: timer 0 + timer 1 + timer 2
- Chức năng:
  - Đếm sự kiện (counter): đếm số xung trên 1 chân chip
  - Capture mode: đếm số xung + giữ giá trị đếm khi có sự kiện trên chân T2EX
  - Định thời (timer): gây ngắt sau một khoảng thời gian lập trình được. Timer 1 còn được dùng tạo xung cho mạch UART.

# Sơ đồ hoạt động: Mode 1 (16 bit)



# Sơ đồ hoạt động: Mode 2 (8 bit auto-reload)



# Ví dụ:

```
#include "reg51.h"
```

```
void main()
```

```
{
```

```
    TMOD = 0x01;           // timer  
0 to 16-bit mode
```

```
    TH0 = timer0_high_byte; // set  
timer 0 high byte
```

```
    TL0 = timer0_low_byte;  // set  
timer 0 low byte
```

```
    ET0 = 1;               //  
enable timer 0 interrupt
```

```
    EA = 1;                //  
enable global interrupt flag
```

```
    TR0 = 1;               //  
timer 0 run
```

```
}
```

```
void timer0_ISR (void) interrupt 1
```

```
{
```

```
    // stop timer
```

```
    TR0 = 0;                // timer 0 off
```

```
    // event processing here
```

```
    //
```

```
    // continue timer
```

```
    TH0 = timer0_high_byte; // reload timer 0  
high byte
```

```
    TL0 = timer0_low_byte;  // reload timer 0  
low byte
```

```
    TR0 = 1;                // timer 0 run
```

```
}
```



# UART: vào ra nối tiếp không đồng bộ

- Có 4 chế độ hoạt động
- Thường dùng Mode 1:
  - 8 bit dữ liệu,
  - Không có parity,
  - Baud rate lập trình được,
    - Dùng timer 1 (mặc định), hoặc timer 2
    - Dùng bộ tạo baud rate có sẵn (cho chip Atmel)
- Tính toán giá trị baud rate

<http://www.keil.com/products/c51/baudrate.asp>

# UART

```
#include "reg51.h"
```

```
char uart_data;
```

```
void main (void)
```

```
{
```

```
    SCON = 0x50;          /* uart in mode 1 (8 bit), REN=1 */
```

```
    TMOD = TMOD | 0x20 ; /* Timer 1 in mode 2 */
```

```
    TH1 = 0xFD;           /* 9600 Bds at 11.0592MHz */
```

```
    TL1 = 0xFD;           /* 9600 Bds at 11.0592MHz */
```

```
    ES = 1;               /* Enable serial interrupt*/
```

```
    EA = 1;               /* Enable global interrupt */
```

```
    TR1 = 1;              /* Timer 1 run */
```

```
    while(1);             /* endless */
```

```
}
```

```
void serial_IT(void) interrupt 4
```

```
{
```

```
    if (RI == 1)
```

```
    {
```

```
        RI = 0;           /* prepare for next reception */
```

```
        uart_data = SBUF;  /* Read receive data */
```

```
        SBUF = uart_data;  /* Send back same data on uart*/
```

```
    }
```

```
    else
```

```
        TI = 0;           /* if emission occur */
```

```
}
```

