



**Aula 6**

Engenharia de Sistemas Embarcados

Prof. Abel Guilhermino

Tópico: Arquitetura e Ambiente de Desenvolvimento




### Interrupção

- É um evento externo ou interno que obriga o microprocessador a suspender suas atividades temporariamente, para atender a este evento que o interrompeu.
- Em outras palavras, é uma ocorrência que faz o microprocessador parar a sua rotina e se desviar para outro ponto do software, em que se localiza o serviço de interrupção que foi gerado pela ocorrência.
- Após o serviço de interrupção estar completo, o microprocessador desvia-se novamente e exatamente para onde estava antes de ter sido interrompido.

### Interrupções no 8051

- 5 maneiras:
  - Pela interrupção externa INT0
  - Pela interrupção externa INT1
  - Pelo interna gerada pelo TIMER0
  - Pelo interna gerada pelo TIMER1
  - Pelo canal de comunicação serial
- Prioridade
  - INT0 (maior)
  - Timer0
  - INT1
  - Timer1
  - Serial (menor)

T2 / P1.0	1	40	VCC
T2EX / P1.1	2	38	P0.0 / AD0
P1.2	3	39	P0.1 / AD1
P1.3	4	37	P0.2 / AD2
P1.4	5	36	P0.3 / AD3
P1.5	6	35	P0.4 / AD4
P1.6	7	34	P0.5 / AD5
P1.7	8	33	P0.6 / AD6
P1.8	9	32	P0.7 / AD7
P2.0 / P3.0	10	31	EA
TxD / P3.1	11	30	ALE
-INT0 / P3.2	12	29	RESN
-INT1 / P3.3	13	28	P2.7 / A15
T0 / P3.4	14	27	P2.6 / A14
T1 / P3.5	15	26	P2.5 / A13
-WR / P3.6	16	25	P2.4 / A12
-RD / P3.7	17	24	P2.3 / A11
XTAL2	18	23	P2.2 / A10
XTAL1	19	22	P2.1 / A9
GND	20	21	P2.0 / A8

### Interrupção 8051

- Endereços de desvio para cada interrupção:
  - INT0\ : 03H
  - Timer/Counter0 : 0BH
  - INT1\ : 13H
  - Timer/Counter1 : 1BH
  - Serial : 23H

Obs: INT0,Timer0,INT1 e Timer1 possuem apenas 8 bytes antes de sobrepôr com o próximo. O reset tem 3 bytes para poder ser deslocado para outra posição de memória (o início do programa).

### Interrupção 8051

- Registros de controle das interrupções
  - O 8051 possui dois registros de 8 bits cada um na sua memória RAM que servem para habilitação ou não das interrupções e indica para a CPU a prioridade de cada uma delas.
  - Sempre que uma interrupção é requisitada, um certo bit de controle dessa interrupção será setado e assim permanecerá até que essa interrupção seja atendida.
  - Exceção é a interrupção de canal serial, que deve ter seu bit de controle setado por software.

### Registro IE (Interrupt Enable)

- O registro IE (Interrupt Enable - Habilitador de Interrupções)
- Tem por função indicar qual das interrupções está ou não habilitada.

EA	X	X	ES	ET1	EX1	ET0	EX0
----	---	---	----	-----	-----	-----	-----

- EA: (Enable All), habilita todas as interrupções. Qdo 0, desabilita todas as interrupções. Qdo 1, permite habilitar outras interrupções
- ES: (Enable Serial), qdo 1 libera a interrupção pedida pelo canal serial, se EA=1. Qdo 0, inibe a interrupção pedida pelo canal serial, independente de EA.
- ET1: (Enable Timer1),qdo 1, libera a interrupção pedida pelo timer1, se EA=1. Qdo 0, inibe interrupção pedida pelo timer1 independente de EA

### Registro IE (Interrupt Enable)

EA	X	X	ES	ET1	EX1	ET0	EX0
----	---	---	----	-----	-----	-----	-----

- EX1 : (Interrupção externa 1), qdo 1, libera interrupção pedida pelo dispositivo ligado ao pino INT1\ se EA=1. Qdo 0, inibe a interrupção pedida pelo dispositivo externo ligado a INT1\, independente de EA.
- ET0: (Enable Timer 0), qdo 1, libera interrupção pedida pelo Timer 0, desde que EA=1. Qdo 0, inibe a interrupção pedida pelo Timer 0, independente de EA.
- EX0: (Interrupção externa 0), qdo 1, libera interrupção pedida pelo dispositivo ligado ao pino INT0\ se EA=1. Qdo 0, inibe a interrupção pedida pelo dispositivo externo ligado a INT0\, independente de EA.

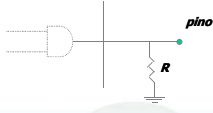
### Registro IP (Interrupt Priority)

X	X	X	PS	PT1	PX1	PT0	PX0
---	---	---	----	-----	-----	-----	-----

- PS: Prioridade serial e qdo em 1, indica prioridade alta à interrupção gerada pelo canal serial, se a mesma estiver habilitada. Qdo em 0 indica prioridade baixa.
- PT1: Prioridade no Timer1, qdo 1, indica prioridade alta para a interrupção pedida pelo Timer1. Qdo 0, indica prioridade baixa para esta interrupção.
- PX1: Prioridade de INT1, e quando 1, indica prioridade alta para interrupção pedida pelo dispositivo externo ligado ao pino INT1. Se 0 indica prioridade baixa para esta interrupção.
- PT0: idem a PT1 para o Timer0
- PX0: idem a PX1 para o INT0.

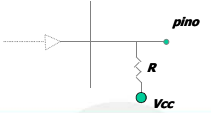
### Portas de I/O

- Observações Importantes
  - Pull-Down
    - Processo pelo qual garantimos que um certo ponto num circuito lógico ficará num nível lógico fixo, não flutuando aleatoriamente. Neste caso garantimos nível 0.



### Portas de I/O

- Observações Importantes
  - Pull-Up
    - Processo pelo qual garantimos que um certo ponto num circuito lógico ficará num nível lógico fixo, não flutuando aleatoriamente. Neste caso garantimos nível 1.



### Programação Modular

- Como a maioria das linguagens de alto nível, C é uma linguagem de programação modular (mas não uma linguagem orientada a objetos).
- Cada tarefa pode ser encapsulada como uma função

### Estrutura básica de um programa

- Diretivas do compilador
- Include file
- Declaração de funções
- Função Principal
- Sub-funções
- Rotinas de serviço de interrupção

## Variáveis

- Tipos
  - int (16-bits in our compiler)
  - char (8-bits)
  - short (16-bits)
  - long (32-bits)
  - sbit (1-bit) → not standard C – an 8051 extension
  - others that we will discuss later



## Tipos

- Alguns outros tipos:

signed char (8 bits) –128 to +127  
 signed short (16 bits) –32768 to +32767  
 signed int (16 bits) –32768 to +32767  
 signed long (32 bits) –2147483648 to +2147483648

unsigned char (8 bits) 0 to + 255  
 unsigned short (16 bits) 0 to + 65535  
 unsigned int (16 bits) 0 to + 65535  
 unsigned long (32 bits) 0 to + 4294967295



## Novos Tipos

- New data types:
- Exemplo:
 

```
bit    bit new_flag;           //stored in 20-2F
sbit   sbit LED = P1^6;
sfr    sfr SP = 0x81;         //stack pointer
sfr16  sfr16 DP = 0x82;      // data pointer
```



## Operadores

- Arithmetic: +, -, \*, /
- Relational: >, >=, <, <=
- Equality: ==, !=
- Logical: && (and), || (or)
- Increment and decrement: ++, --
- Example:
 

```
if (x != y) && (c == b)
{
  a=c + d*b;
  a++;
}
```



## Programação

- Definição de biblioteca para o 8051
  - #include <reg51.h>
    - Biblioteca onde encontram-se definida o endereçamento dos: SFR, bit register, TCON, IE, IP, P3 e SCON.




## Programação

- Atribuir um valor a uma porta:
  - P3=0xFF;



### Timer

- Usados para gerar periódicos e precisos pedidos de interrupção, medir larguras de pulsos externos, contagem de tempo, entre outras funções.
- O 8051 possui 2 timers internos, que são programados por software e podem operar completamente independente dos demais sistemas do chip.
- Dois registros são importantes para o controle do timer (TMOD e TCON)




Grupo de Engenharia da Computação - CIn / UFPE

### Registro TCON

TF1	TR1	TF0	TR0				
-----	-----	-----	-----	--	--	--	--

- TCON (Timer Control – Controle do Timer)
- TF1: sempre que ocorrer um overflow no Timer1, este bit será setado, gerando um pedido de interrupção do Timer1. É resetado pelo hardware ao fim da rotina de interrupção.
- TR1: Setado pelo software para ligar o Timer1. Resetado para desligar (parar a contagem)
- TF0: sempre que ocorrer um overflow no Timer0, este bit será setado, gerando um pedido de interrupção do Timer0. É resetado pelo hardware ao fim da rotina de interrupção.
- TR0: Setado pelo software para ligar o Timer0. Resetado para desligar (parar a contagem)




Grupo de Engenharia da Computação - CIn / UFPE

### Registro TCON

						IT0	IE0	IT1	IE1
--	--	--	--	--	--	-----	-----	-----	-----

- Obs: Os bits 3,2,1 e 0 do TCON são usados para o controle das interrupções externas.
- IT0: Indica qual o processo para a chamada da interrupção INTO. Se 1, INTO será aceita na transição de 1 para 0, devendo permanecer em 0 pelo menos durante 12 períodos de clock
- IE0: Bit para o hardware de controle da interrupção INTO. Este bit é setado pelo hardware interno quando for detectada uma transição de 1 para 0 no pino INTO. É resetado logo que a interrupção é atendida.
- IT1: Indica qual o processo para a chamada da interrupção INT1. Se 1, INT1 será aceita na transição de 1 para 0, devendo permanecer em 0 pelo menos durante 12 períodos de clock
- IE1: Bit para o hardware de controle da interrupção INT1. Este bit é setado pelo hardware interno quando for detectada uma transição de 1 para 0 no pino INT1. É resetado logo que a interrupção é atendida.




Grupo de Engenharia da Computação - CIn / UFPE

### Registro TMOD

G1	C1	M11	M01	G0	C0	M10	M00
----	----	-----	-----	----	----	-----	-----

- G1 e G0: tem a função de escolher como os timers serão habilitados. Ex: para o Timer0, se o G0 for 0, o Timer0 será habilitado fazendo o bit TR0 (no TCON) = 1.
- C1 e C0: estes dois bits selecionam a função de cada timer individualmente. Se o bit C0 ou C1 foi 0, a função é de temporização e o sinal será interno, e se for 1, a função será de contagem e o sinal será externo.
- (M11 e M01) e (M10 e M00) : estes dois bits combinados permite-nos obter modos diferentes de uso para cada timer.
  - Modo 0
  - Modo 1
  - Modo 2




Grupo de Engenharia da Computação - CIn / UFPE

### Registro TH0/TH1 e TL0/TL1

TH0 ou TH1								TL0 ou TL1							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
contagem								ignorar				escala			


- Modo 0
  - Contador ou temporizador de 8 bits com divisor de frequência de até 32 vezes.
  - Neste modo, o registro TH0 e TH1 receberá o valor inicial da contagem, e dessa forma podemos contar até FFH, com o valor inicial escrito por software.
  - Ao ocorrer o overflow nesse registro, o timer em questão gerará u pedido de interrupção, que será ou não aceito pela CPU.
  - O sinal de contagem(interno ou externo) será dividido pelo valor binário presente nos bits 0 a 4 do registro TL0 ou TL1, conforme o timer usado.
  - Os bits 5 a 7 devem ser ignorados em caso de leitura.



Grupo de Engenharia da Computação - CIn / UFPE

### Exemplo (Modo0)

- A cada 2000 pulsos na entrada gerar uma interrupção. Obs: Contagem de 9BH até FFH (100 contagens) ---- M1x=0 e M0x=0
- Obs: Contagem de 9BH até FFH (100 contagens)
- Obs: Pode-se usar um divisor de frequência de 20 (a cada 20 pulsos será contado 1)
  - Escreve-se em TH0 o valor inicial da contagem, ou seja, TH0=9BH
  - Depois escrevemos em TL0 um valor binário de certa forma que os pulsos de entrada sejam divididos por 20, valor este então 14H.
  - Quando o contador for acionado, a cada 20 contagens externas, o registro TH0 sofrerá UM incremento, começando de 9BH.
  - Ao chegar em FFH, ocorrerá o overflow na contagem seguinte, e teremos uma interrupção gerada pelo timer0.
  - Como podemos dividir no máximo por 32 e contar no máximo até 255 este modo permite-nos contar 8160 vezes.



Grupo de Engenharia da Computação - CIn / UFPE

### Modo 1

- Contador ou temporizador de 16 bits
- M1x=0 e M0x = 1
- Neste modo de funcionamento, temos um timer de 16 bits, que desta forma usa o par de registradores TH0 e TL0 ou TH1 e TL1 para efetuar a contagem.
- Da mesma forma que no modo 0, podemos contar até FFFFH (65535) com valor inicial programável por software.
- Aqui também ao ocorrer overflow, o sistema recebe um pedido de interrupção.

### Modo 2

- Contador ou temporizador de 8 bits com recarga automática
- M1x=1 e M0x=0
- Neste modo de funcionamento, temos um sistema no qual não precisamos reescrever por software o valor a ser contado.
- Aqui escrevemos no registro TL0 e TL1, o valor inicial da contagem, e em TH0 ou TH1 o valor inicial da PRÓXIMA contagem.
- Ao ocorrer o overflow em TLx, o sistema gera a interrupção e automaticamente recarrega em TLx o valor contido em THx, e prossegue incrementando.

<b>TH0 ou TH1</b>	<b>TL0 ou TL1</b>
Valor a recarregar	Registro onde ocorre a contagem

### Inicializando Timer0

**TMOD = 0x21**

<b>G1</b>	<b>C1</b>	<b>M11</b>	<b>M01</b>	<b>G0</b>	<b>C0</b>	<b>M10</b>	<b>M00</b>
0	0	0	0	0	0	0	1

```

void InitTimer0 (void)
{
    TMOD = 0x01; // timer 0 16-bit timer modo 1
    TH0 = 0xDB; // high(56319); // timer 0 is a 10ms timer
    TL0 = 0xFF; // low(DBFF=56319);
    TR1 = 0; // disable timer 1
    TR0 = 1; // enable timer 0
    EA=1; // enable all interrupts
    ET0=1; // enable timer0 interrupt
}
    
```

**Contagem do timer:**

- Um único ciclo de máquina consiste de 12 pulsos do cristal.
- Com isso uma execução do timer seria incrementado:  $11.059.200 / 12 = 921.600$  vezes/seg
- Se um timer tem contagem de DBFF até FFFF:  $(FFFF-DBFF)/921.600 = 0,01$  segundos

**Timer de 10ms**

### Tratando Interrupção Timer0

- Interrupção do timer0 usando o banco 2
- Sempre que for gerada a interrupção esta função é executada.

```

void timer0 (void) interrupt 1 using 2 {
    interruptcnt++;
    if (interruptcnt == 50) {
        /* count to 50 */
        /* second counter */
        interruptcnt = 0;
        flag = 1;
    }
    TH0=0xDB; TL0 = 0xFF;
}
    
```

### Exemplo usando Timer0

```

#include <reg51.h>

void InitTimer0 (void)
{
    TMOD = 0x01; TH0 = 0xDB;
    TL0 = 0xFF; TR1 = 0;
    TR0 = 1; EA=1;
    ET0=1;
}

void timer0 (void) interrupt 1 using 2 {
    interruptcnt++;
    if (interruptcnt == 50) {
        interruptcnt = 0;
        flag = 1;
    }
    TL0 = 0xFF; TH0 = 0xDB;
}

void Main(void){
    InitTimer0();
    while (1) {
        if (flag==1){
            Saida = 1;
        }
    }
}
    
```

### Serial

- Modos de comunicação Síncrona e Assíncrona
  - 1 síncrona (modo 0) e 3 assíncronas (modo 1,2,3)
- Transmite um byte, bit por bit em sequência preestabelecida e pré-programada.
- Tipicamente, costuma-se codificar o byte a ser transmitido em código ASCII, pois é padrão universal de comunicação.
- O periférico serial do microcontrolador 8051 apresenta 2 palavras de atuação: uma de controle (SCON) e outra de transmissão e recepção (SBUF).
  - O erro de baud-rate é derivado exclusivamente do Timer1, isto é, devemos programá-lo para gerar o baud-rate necessário.

### Registro SCON

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

- SM0 e SM1 combinados, programam os 4 modos possíveis de operação.
- O bit SM2 é usado para multiprocessamento.
- O bit REN permite iniciar a recepção de dados
- Os bits TB8 e RB8 permitem enviar um nono bit junto com cada byte transmitido.
- Os bits TI e RI são os flags de interrupção para transmissão TI e recepção RI de dados.

### Registro PCON

SMOD	*	*	*	GF1	GF0	PD	IDL
------	---	---	---	-----	-----	----	-----

- No caso da serial apenas interessa o bit **SMOD**.
- O bit SMOD dobra a relação de divisão de frequência na SERIAL
- GF0 e GF1: bit de uso geral
- PD: Bit de Power Down, modo especial de trabalho dos microcontroladores da série MOD-H em que o microcontrolador congela suas atividades.
- IDL: Bit de modo Idle, modo especial de trabalho dos microcontroladores da série MOS-H, em que o microcontrolador congela suas atividades.

### Condições dos modos da Serial

MOD0	SM0	SM1	Comunicação	Tamanho	Baud-rate
0	0	0	Síncrona	8bits	$f_{clk}/12$
1	0	1	Assíncrona	8bits	Dados por Timer1
2	1	0	Assíncrona	9bits	$f_{clk}/32$ ou $f_{clk}/64$
3	1	1	Assíncrona	9bits	Dados por Timer1

### Como gerar baud-rates no Timer1

Baud-Rate desejado	Baud-rate obtido	Erro obtido (%)	Clock usado (MHz)	Bit SMOD	Carag Timer 1 Modo 2
1200	1200	0	11,059	0	E8h
2400	2400	0	11,059	0	F4h
9600	9600	0	11,059	0	FDh
19200	19200	0	11,059	1	FDh
1200	1202	0,16	12,000	0	E6h
2400	2404	0,16	12,000	1	E6h
9600	8923	7*	12,000	1	F9h
19200	20833	8,5*	12,000	1	FDh

\* Representam erros que podem ser intoleráveis na transmissão serial (usual: até 5%)

### Programando a Serial

```

void init_serial() {
    EA = 1;
    ES = 0;
    SCON = 0x50;
    PCON = 0x80;
    ES = 1;
}

void serial_envia(char env) {
    SBUF=env;
}

void serial_recebe(char rec) {
    armazena=SBUF;
}

void serial_irq() interrupt 4{
    ES=0;
    if (RI) {
        serial_recebe(SBUF);
        RI = 0;
    } else{
        TI = 0;
    }
    ES = 1;
}

Programar Timer 1 (baud rate){
    TMOD |= 0x20;    --timer1 modo2
    TL1 = TH1 = 232; 1 --1200 baud 12MHz
    TR1 = 1;        --timer1 run
}
    
```

### Exercício 1

- Implementar um programa utilizando a ferramenta Keil para o 8051 que controle o seguinte sistema abaixo:

- A e B: entrada de 4 bits (botões)
- OP: Bit entrada. OP=1 (adição), OP=0 (subtração)
- Pino 2.0 = pino de ativação de operação
- C: saída de leds resultante da operação

### Exercício 2

- Implementar um programa utilizando a ferramenta Keil para o 8051 que controle o seguinte sistema abaixo:

- Fazer um led piscar de 1 em 1 segundo
- Início do processo: qdo for acionado a chave do pino P2.0

T2/P1.0	1	40	VCC
T2EX/P1.1	2	39	P0.0/AD0
P1.2	3	38	P0.1/AD1
P1.3	4	37	P0.2/AD2
P1.4	5	36	P0.3/AD3
P1.5	6	35	P0.4/AD4
P1.6	7	34	P0.5/AD5
P1.7	8	33	P0.6/AD6
P1.8	9	32	P0.7/AD7
RST	9	31	EA
RXD/P3.0	10	30	ALE
TXD/P3.1	11	29	PSEN
INT0/P3.2	12	28	P2.7/A15
INT1/P3.3	13	27	P2.6/A14
T0/P3.4	14	26	P2.5/A13
T1/P3.5	15	25	P2.4/A12
WR/P3.6	16	24	P2.3/A11
RD/P3.7	17	23	P2.2/A10
XTAL2	18	22	P2.1/AS
XTAL1	19	21	P2.0/AB
GND	20		

### Exercício 3

- Implementar um programa utilizando a ferramenta Keil para o 8051 que controle o seguinte sistema abaixo:

- Fazer um led piscar de 1 em 1 segundo ou de 2 em 2 segundos
- Início do processo: qdo for acionado a chave do pino P2.0
- Chave de seleção de frequência: P1.2

T2/P1.0	1	40	VCC
T2EX/P1.1	2	39	P0.0/AD0
P1.2	3	38	P0.1/AD1
P1.3	4	37	P0.2/AD2
P1.4	5	36	P0.3/AD3
P1.5	6	35	P0.4/AD4
P1.6	7	34	P0.5/AD5
P1.7	8	33	P0.6/AD6
P1.8	9	32	P0.7/AD7
RST	9	31	EA
RXD/P3.0	10	30	ALE
TXD/P3.1	11	29	PSEN
INT0/P3.2	12	28	P2.7/A15
INT1/P3.3	13	27	P2.6/A14
T0/P3.4	14	26	P2.5/A13
T1/P3.5	15	25	P2.4/A12
WR/P3.6	16	24	P2.3/A11
RD/P3.7	17	23	P2.2/A10
XTAL2	18	22	P2.1/AS
XTAL1	19	21	P2.0/AB
GND	20		

### Exercício 4

- Implementar um programa que fazem os 4 leds piscarem alternadamente de 0,5 em 0,5 segundo.

- Início do processo: qdo for acionado a chave do pino P2.0 e ao pressionar novamente o sistema pausa.

T2/P1.0	1	40	VCC
T2EX/P1.1	2	39	P0.0/AD0
P1.2	3	38	P0.1/AD1
P1.3	4	37	P0.2/AD2
P1.4	5	36	P0.3/AD3
P1.5	6	35	P0.4/AD4
P1.6	7	34	P0.5/AD5
P1.7	8	33	P0.6/AD6
P1.8	9	32	P0.7/AD7
RST	9	31	EA
RXD/P3.0	10	30	ALE
TXD/P3.1	11	29	PSEN
INT0/P3.2	12	28	P2.7/A15
INT1/P3.3	13	27	P2.6/A14
T0/P3.4	14	26	P2.5/A13
T1/P3.5	15	25	P2.4/A12
WR/P3.6	16	24	P2.3/A11
RD/P3.7	17	23	P2.2/A10
XTAL2	18	22	P2.1/AS
XTAL1	19	21	P2.0/AB
GND	20		