SEL-433 APLICAÇÕES DE MICROPROCESSADORES I

8051

AULA 8

Técnicas de Interface

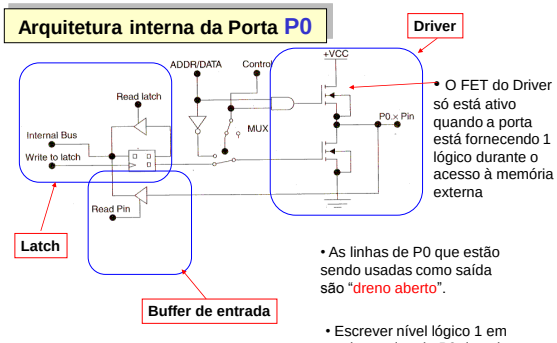
Prof. Evandro L. L. Rodrigues

Técnicas de Interface com Microcontroladores (8051)

Estrutura das Portas do 8051:

- Todas as quatro Portas (P0,P1,P2,P3) da família de Microcontroladores MCS-51 são bi-direcionais.
- Cada uma delas consiste de um **Latch**, um **Driver** de saída e um **Buffer** de entrada.
- Os Drivers de saída dos Buffers de entrada da Porta P3 são multifuncionais - podem também assumir funções alternativas.

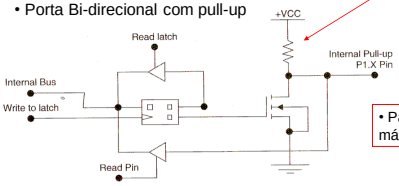
Arquitetura interna da Porta P0



- O FET do Driver só está ativo quando a porta está fornecendo 1 lógico durante o acesso à memória externa
- As linhas de P0 que estão sendo usadas como saída são "dreno aberto".
- Escrever nível lógico 1 em qualquer pino da P0 desativa os FETs de saída e assim o pino flutua.
- Porta Bi-direcional verdadeira

Arquitetura interna da Porta P1

- Porta Bi-direcional com pull-up



- O Resistor de Pull-up fixo, fornece corrente (**IIL = Corrente de Input Low**) quando externamente é aplicado um nível lógico 0.
- Para o 89S52 o valor máximo de **IIL = - 50 uA**
- Quando o Microcontrolador escreve nível lógico zero na saída, os Máximos Valores de **IOL (Corrente de Output Low)** para o 89S52 são:

Máxima IOL por pino: **10 mA**

Máxima IOL para 8-bits: P0 = **26 mA**

P1, P2 e P3: **15 mA**

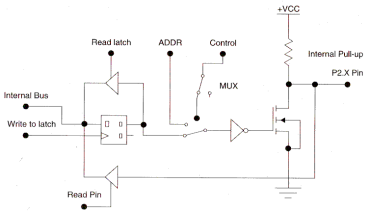
Máxima IOL - todos os pinos (P0, P1,P2,P3): **71 mA**

- A máxima corrente de saída em nível alto **(IOH = Corrente de Output High)** para o 89S52 é = **- 60 µA**

Arquitetura interna da Porta P2

• Os valores de IOL, IOH, IIL e IIH são os mesmos da Porta P1

• Porta Bi-direcional com pull-up



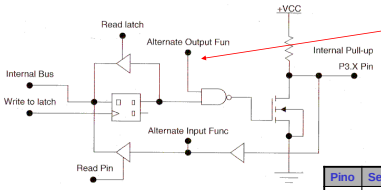
• Após um Reset, todas as saídas dos Latches vão para nível lógico 1, programando todas as Portas (P0, P1, P2, P3) como entradas.

• A Porta P2, como a P0, não pode ser usada como entrada/saída se estiver sendo usada como Barramento de Endereços.

Arquitetura interna da Porta P3

• Os valores de IOL, IOH, IIL e IIH são os mesmos da Porta P1

• Porta Bi-direcional com pull-up



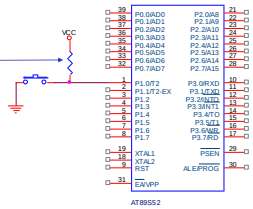
• Se o Latch da Porta P3 está em nível lógico 1, a saída é controlada pelo sinal de **Função de Saída Alternativa**

• Os Pinos da Porta P3 estão sempre disponíveis como Entrada das Funções Alternativas

Pino	Sentido	Função Alternativa
P3.0	Entrada	RxD (Entrada Serial)
P3.1	Saída	TxD (Saída Serial)
P3.2	Entrada	INT0 (Interrupção 0)
P3.3	Entrada	INT1 (Interrupção 1)
P3.4	Entrada	T0 (Contador 0)
P3.5	Entrada	T1 (Contador 1)
P3.6	Saída	WR (sinal de escrita)
P3.7	Saída	RD (Sinal de Escrita)

Conectando Chaves mecânicas às entradas do Microcontrolador

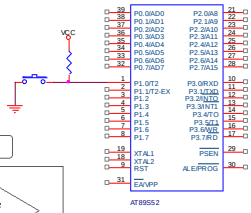
É possível utilizar, mas não obrigatoriamente, um Resistor de **10K ohms** que consumirá uma corrente da fonte de aproximadamente 500 uA quando a chave for acionada.



• É importante ressaltar que quando a chave for acionada, o resistor de Pull-up da porta fornece também no máximo IIL = 50 uA.

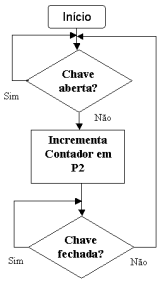
Exemplo de Aplicação

• Considerando o esquema, implementar um contador binário de 8 Bits cujo valor deve ser incrementado de uma unidade a cada toque na chave e enviado à Porta P2



• **Primeira Solução**

• Testando-se a chave duas vezes verifica-se se o usuário já completou a operação de um toque, ou seja, um pulso completo.

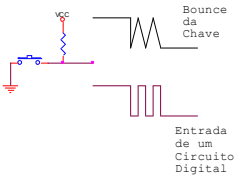


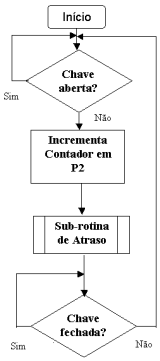
• Esta solução funcionaria na Prática?

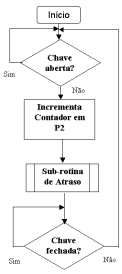
NÃO

• As chaves mecânicas possuem inércia ao fechar que geram ruídos que poderão ser detectados pelo software ao se proceder a leitura da mesma.

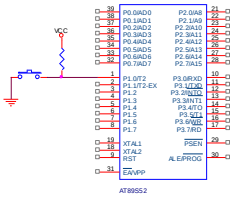
• Este ruído, chamado de Bounce da chave deve ser eliminado, ou por hardware (capacitor, FF tipo D) ou por software inserindo um atraso antes da leitura da chave fechada.





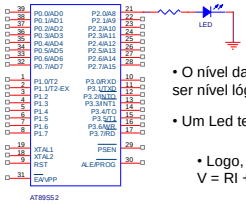



```
; ***********************************************************************
; Programa de debounce de chave mecânica
; Uma sub-rotina de atraso deve ser inserida para garantir o tempo de
; estabilização mecânica da chave. Para a maioria das chaves um atraso de
; 500 us é suficiente.
; ***********************************************************************

        ORG 0
        CLR A           ; Inicializa o contador em zero
LOOP:   JB P1.0,$       ; Testa se a chave está aberta
        INC A           ; Incrementa o contador
        MOV P2,A        ; Mostra o valor do contador na Porta P2
        LCALL ATRASO    ; Realiza o debounce da chave
        JNB P1.0,$      ; Verifica se a chave continua fechada
        SJMP LOOP       ; Retorna para nova contagem de pulso

; ***********************************************************************

ATRASO: MOV R0,#0FFH    ; Valor do atraso para debounce da chave de
        DJNZ R0,$       ; aproximadamente 500 us
        RET
; ***********************************************************************
```



Conectando Leds às saídas do Microcontrolador

• Qual o valor do Resistor e da corrente necessária para acender o Led no esquema abaixo?



• O nível da porta para acionar o Led deve ser nível lógico 1, ou seja = 5v

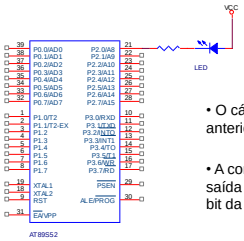
• Um Led tem uma queda de tensão de 1,4 v

• Logo,
V = RI + Vled → R = (5 – 1,4)/I

• Como um Led necessita de aproximadamente 10 mA para uma boa luminosidade → R = 3,6/10mA = 360 ohms

• Como a corrente de saída em nível alto IOH é no máximo 60 uA, esta ligação é inviável

• E se alterarmos o esquema de conexão, qual o valor do Resistor e da corrente necessária para acender o Led ?



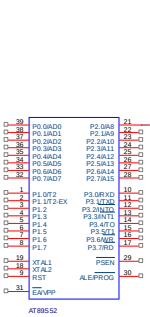
• O nível da porta para acionar o Led deve ser nível lógico 0, ou seja = 0v

• O cálculo do Resistor é o mesmo realizado anteriormente, ou seja, R = 360 ohms

• A corrente a ser avaliada agora é a de nível de saída baixo, ou seja, IOL é no máximo 10 mA por bit da Porta P2.

• O valor máximo de IOL para os 8 Bits da Porta P2 é 15 mA, ou seja, a solução é viável para um Bit mas não deve ser adotada para a porta toda, pois o nível de corrente ultrapassa o máximo recomendado.

Bufferizando as saídas do Microcontrolador



• Utilizando-se um Buffer/driver com transistores, um para cada bit da porta, fornece-se a corrente necessária à operação do Led sem afetar os valores máximos permitidos para o Microcontrolador.

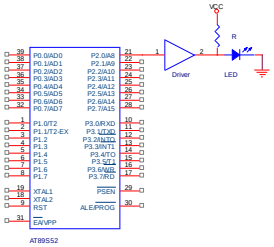
• Nível lógico 1 em P2.0 conduz o transistor acendendo o Led

• R = 360 Ohms fornece corrente de aproximadamente 10 mA ao Led.

• O resistor de base deve ser calculado conforme especificações do transistor com IOH máxima de 60 uA

Bufferizando as saídas do Microcontrolador

• A solução mais indicada é utilizar um Circuito "**Led Driver**" que fornecerá a corrente necessária aos Leds em cada um dos Bits da Porta não permitindo ultrapassar os valores máximos recomendados.



Alguns CI's de Led Driver

•7407 (4 por chip)

•74244 (8 por chip)

• ULN2803 (8 por chip)

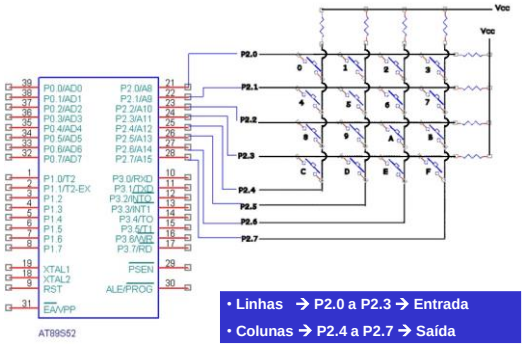
• outros……

Interface do Microcontrolador com Teclado Matricial

• Um Teclado Matricial de 4 linhas por 4 colunas permite a geração de 16 códigos independentes utilizando-se apenas 8 linhas de uma Porta do Microcontrolador.

• Dessa maneira é possível interfacear um Teclado Hexadecimal Matricial ao Microcontrolador para entrada de dados binários

Interface do Microcontrolador com Teclado Matricial

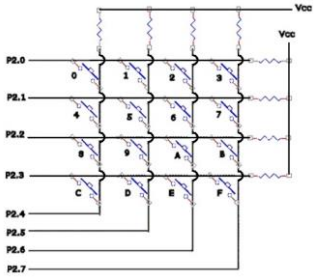


• **Linhas → P2.0 a P2.3 → Entrada**

• **Colunas → P2.4 a P2.7 → Saída**

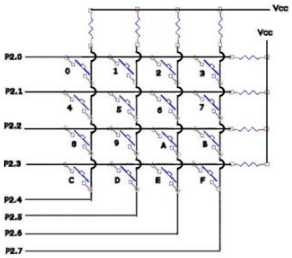
Valores aplicados às colunas(P2.4 a P2.7) e lidos nas linhas (P2.0 a P2.3), se for acionada uma tecla presente na coluna que contém zero.

P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	TECLA
1	1	1	0	1	1	1	0	0
1	1	0	1	1	1	1	0	1
1	0	1	1	1	1	1	0	2
0	1	1	1	1	1	1	0	3
1	1	0	1	1	1	0	1	4
1	1	0	1	1	1	0	1	5
1	0	1	1	1	1	0	1	6
0	1	1	1	1	1	0	1	7
1	1	1	0	1	0	1	1	8
1	1	0	1	1	0	1	1	9
1	0	1	1	1	0	1	1	A
0	1	1	1	1	0	1	1	B
1	1	1	0	0	1	1	1	C
1	1	0	1	0	1	1	1	D
1	0	1	1	0	1	1	1	E
0	1	1	1	0	1	1	1	F



Caso nenhuma tecla seja acionada, o valor das linhas é = F

```
                ORG 0
LOOP:           MOV P2,#0FFH        ; Garante nivel lógico 1 em todos os bits de P2
LOOP1:          CLR P2.4            ; Zera a coluna das chaves 0,4,8,C
                JNB P2.0,ZERO
                JNB P2.1,QUATRO
                JNB P2.2,OITO
                JNB P2.3,CCC
                MOV P2,#0FFH        ; Garante nivel lógico 1 em todos os bits de P2
LOOP2:          CLR P2.5            ; Zera a coluna das chaves 1,5,9,D
                JNB P2.0,UM
                JNB P2.1,CINCO
                JNB P2.2,NOVE
                JNB P2.3,DDD
                MOV P2,#0FFH        ; Garante nivel lógico 1 em todos os bits de P2
LOOP3:          CLR P2.6            ; Zera a coluna das chaves 2,6,A,E
                JNB P2.0,DOIS
                JNB P2.1,SEIS
                JNB P2.2,AAA
                JNB P2.3,EEE
                MOV P2,#0FFH        ; Garante nivel lógico 1 em todos os bits de P2
LOOP4:          CLR P2.7            ; Zera a coluna das chaves 3,7,B,F
                JNB P2.0,TRES
                JNB P2.1,SETE
                JNB P2.2,BBB
                JNB P2.3,FFF
                MOV A,#0FFH
                SJMP    LOOP
```




```
ZERO:       MOV A,#00H
            SJMP    LOOP1
UM:         MOV A,#01H
            SJMP    LOOP2
DOIS:       MOV A,#02H
            SJMP    LOOP3
TRES:       MOV A,#03H
            SJMP    LOOP4
QUATRO:     MOV A,#04H
            SJMP    LOOP1
CINCO:      MOV A,#05H
            SJMP    LOOP2
SEIS:       MOV A,#06H
            SJMP    LOOP3
SETE:       MOV A,#07H
            SJMP    LOOP4
OITO:       MOV A,#08H
            SJMP    LOOP1
NOVE:       MOV A,#09H
            SJMP    LOOP2
AAA:        MOV A,#0AH
            SJMP    LOOP3
BBB:        MOV A,#0BH
            SJMP    LOOP4
CCC:        MOV A,#0CH
            SJMP    LOOP1
DDD:        MOV A,#0DH
            SJMP    LOOP2
EEE:        MOV A,#0EH
            SJMP    LOOP3
FFF:        MOV A,#0FH
            SJMP    LOOP4
```

Exercício 5

•Entrega pelo Site do Curso até o dia 16/05/2016

Fazer um programa em Assembly do 8051 que interfaceado a um Teclado Matricial conforme esquema mostrado nessa aula, comande um Motor de Passo conectado à Porta P1 (P1.0 = Clock, P1.1 = Direção).

1. **O Número de passos (de 01 a 99) deve entrar pelo Teclado Matricial seguido da direção (A = Anti-horária e B = Horária). Após os dados de número de passos e direção terem sido inseridos o programa aguarda uma tecla de início (usar a letra F).**
2. **Quando o Motor executar os passos solicitados no item 1 deve parar e aguardar novos comandos de Número de Passos e Direção.**
3. **Simular no MCU8051 colocando um contador em paralelo com o clock do motor para verificar a contagem dos pulsos.**
4. **Letra C – corrige o campo atual**
5. **Letra D – repete sequência pré-gravada**
6. **Letra E – Limpa memória**