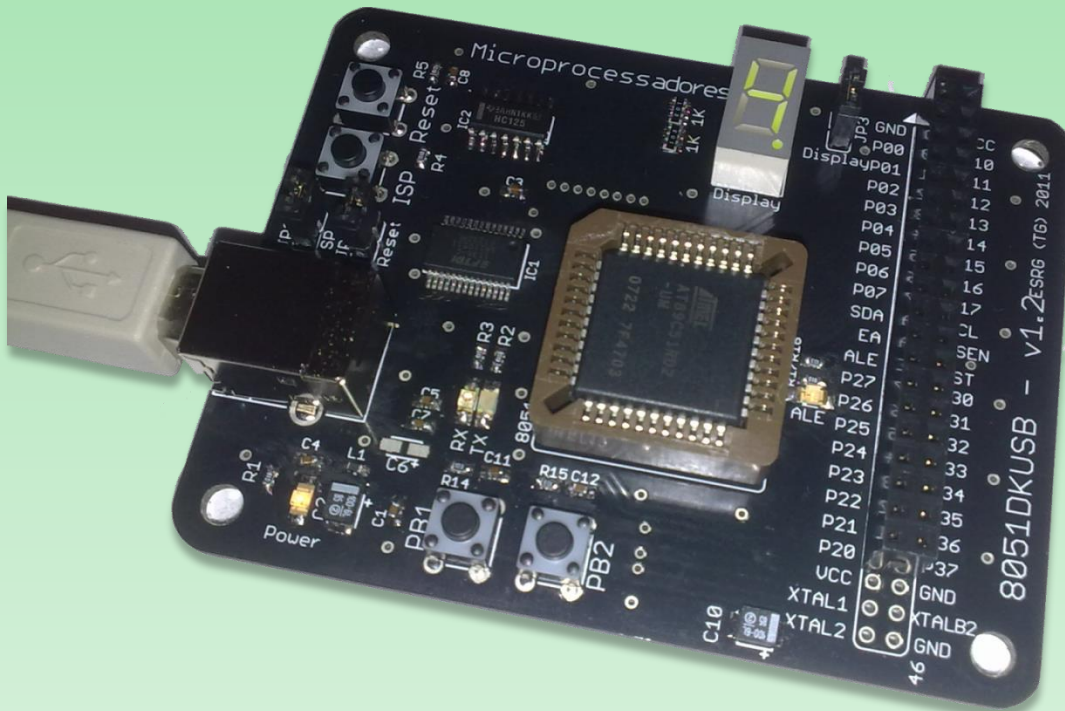
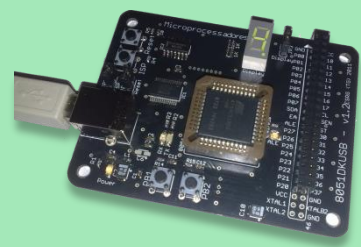


Mestrado Integrado em Engenharia de Telecomunicações e Informática



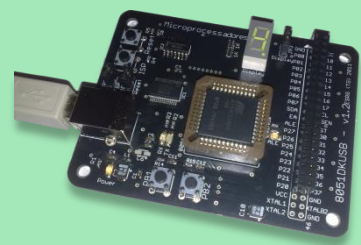
Interface ao exterior

Microcontroladores
2º Ano – A08



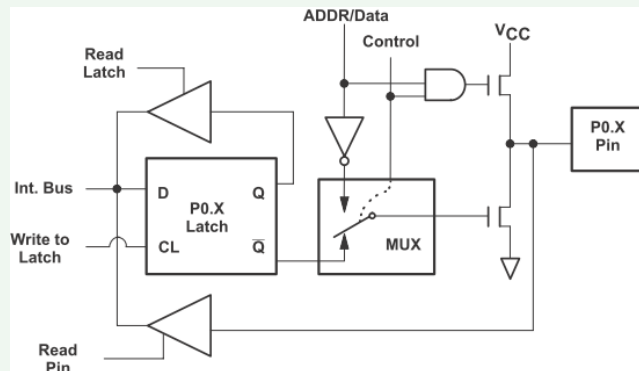
Interface

- Uma das características principais de qualquer microcontrolador é o conjunto de pinos para I/O genérico (Generic Purpose I/O – GPIO)
 - O 8051 contém 4 portas de 8 bits cada
 - Do ponto de vista do software, cada porta é um SFR
 - Porto 0 (80h), porto 1(90h) porto 2(A0h) e porto 3(B0h)
 - Cada pino é endereçável ao *bit*
 - Podem ser usados como entradas/saídas digitais
 - Alguns destes portas podem ter uma função dupla, desde barramentos de dados e endereços, porta série, etc.
 - A leitura de pu pino de um dos portas pode ocorrer em dois pontos diferentes
 - “read latch”: é lido o *flip-flop* do pino. Usado pelas as instruções que leem e alteram o conteúdo do registo: INC, DEC, CPL, JBC, DJNZ, ANL, ORL, XRL
 - “read pin”: é lido diretamente o pino. Usado pelas restantes instruções

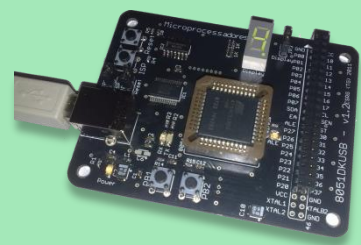


Interface

- Porto 0

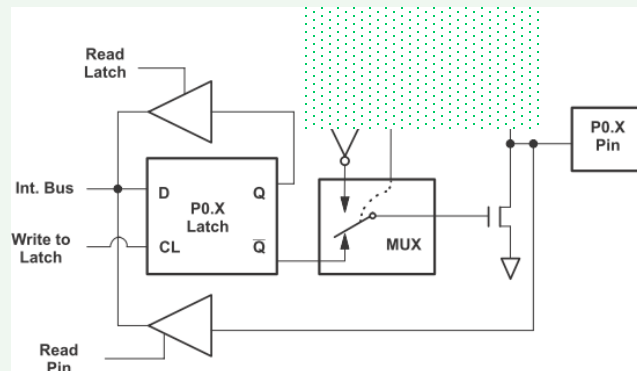


- A figura representa um pino do porto 0
 - A *latch* é a célula de memória para a qual escrevemos no bit do SFR
 - De notar que a leitura da *latch* e a leitura do pino são coisas diferentes. Algumas instruções que “leem” um porto ativam o sinal “read latch” (na verdade não estão a ler o pino) enquanto outras ativam o sinal “read pin” (idêntico em todos os portos)
- O porto 0 é também usado para aceder à memória externa
 - Neste modo, põe na saída o *byte* menos significativo do endereço, multiplexado com o *byte* de dados a ler/escrever na memória
- O porto 0 não contém uma resistência de *pull-up* de apoio à saída

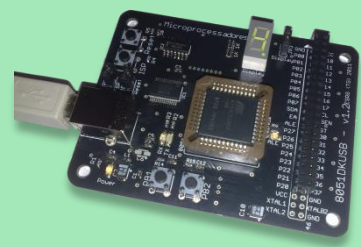


Interface

- Porto 0

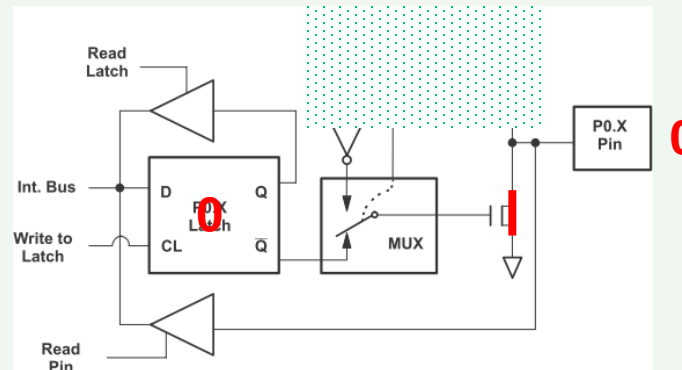


– Escrita/leitura de dados

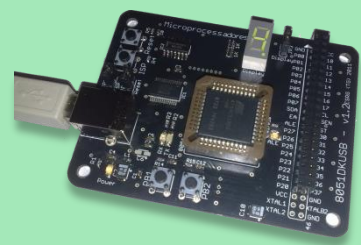


Interface

- Porto 0

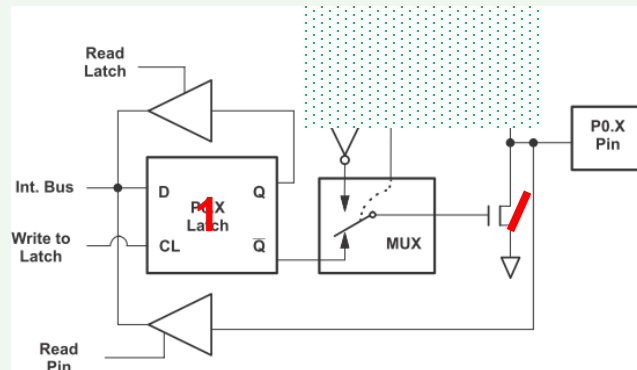


- Escrita/leitura de dados
- Escrita de **0** no pino (na verdade é na *latch*)
 - No pino vai aparecer 0
 - Se de seguida tentarmos ler do pino, vamos ler sempre 0, independentemente do que o hardware externo (se houver) lá colocar!

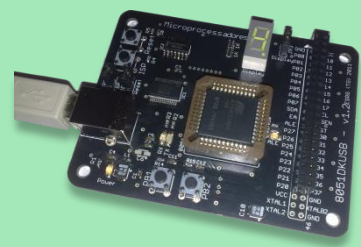


Interface

- Porto 0

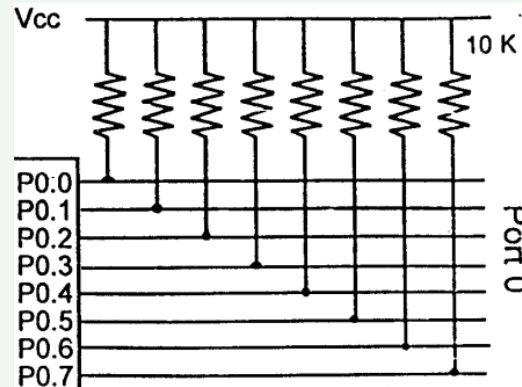


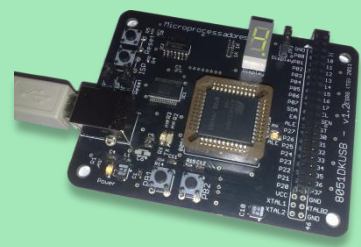
- Escrita/leitura de dados
- Escrita de 1 no pino (na verdade é na *latch*)
 - No pino vai aparecer 0 O pino vai ficar “no ar” (O porto 0 tem saída em coletor aberto!) Será necessário uma resistência de *pull-up* externa para da o nível 1
 - Se de seguida tentarmos ler do pino, vamos ler o que o hardware externo lá colocar! Daí que para ler o pino, antes temos de escrever 1
 - Durante o acesso à memória é escrito 1 na *latch*



Interface

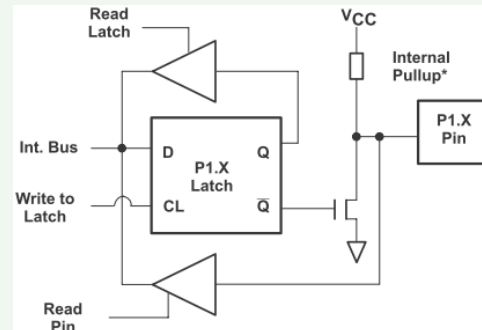
- Porto 0
 - Para usar o porto 1 como entrada deve ser usadas pull-ups externos





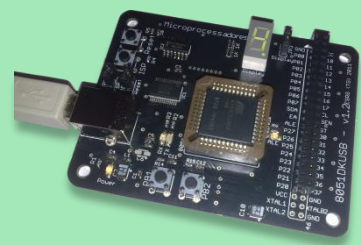
Interface

- Porto 1



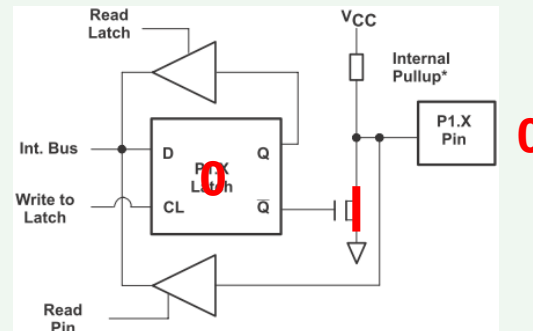
- A figura representa um pino do porto 1

- A *latch* é a célula de memória para a qual escrevemos no bit do SFR
- De notar que a leitura da *latch* e a leitura do pino são coisas diferentes. Algumas instruções que “leem” um porto ativam o sinal “read latch” (na verdade não estão a ler o pino) enquanto outras ativam o sinal “read pin”
- A grande diferença deste porto, tal como os portos 2 e 3, é a existência de *pull-up* de saída

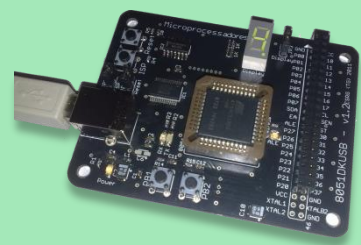


Interface

- Porto 1

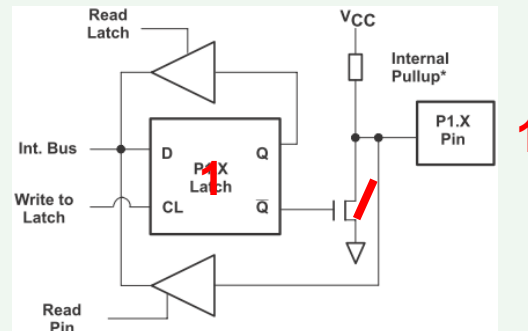


- Escrita/leitura de dados
- Escrita de **0** no pino (na verdade é na *latch*)
 - No pino vai aparecer 0
 - Se de seguida tentarmos ler do pino, vamos ler 0

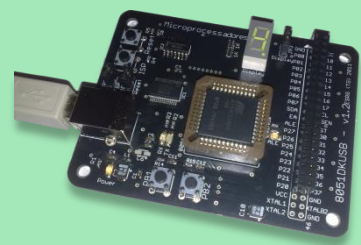


Interface

- Porto 1

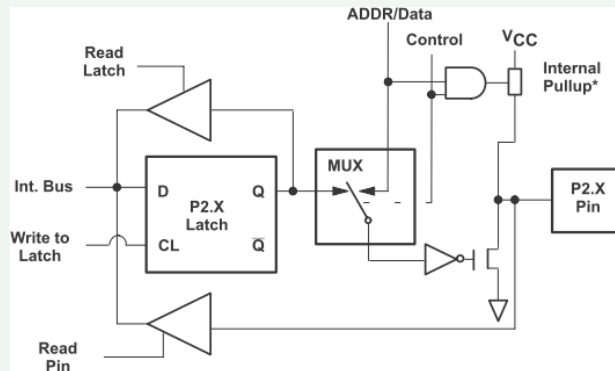


- Escrita/leitura de dados
- Escrita de 1 no pino (na verdade é na *latch*)
 - Na saída vai aparecer 1. A resistência de *pull-up* interna garante isso!
 - Se de seguida tentarmos ler do pino, vamos ler o que o hardware externo lá colocar! Daí que para ler o pino, antes temos de escrever 1 (se o *hardware* externo colocar 0, vamos ler 0, se colocar 1, vamos ler 1)

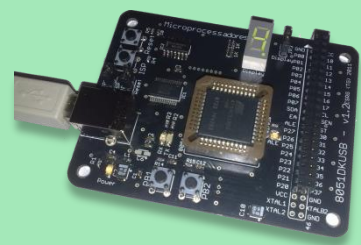


Interface

- Porto 2

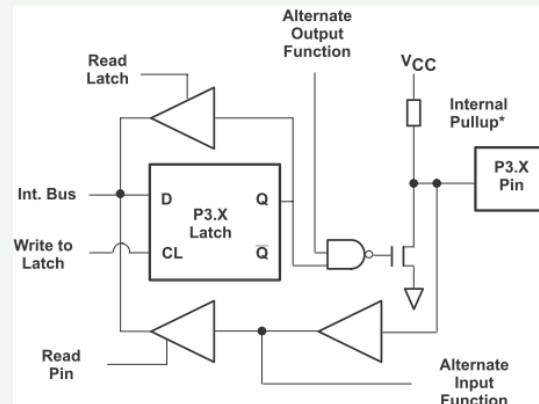


- Funcionamento idêntico ao do porto 1, do ponto de visto do I/O
- O porto 0 é também usado para aceder à memória externa
 - Neste modo, põe na saída o byte mais significativo do endereço
 - Durante o acesso à memória o conteúdo da *latch* permanece inalterado



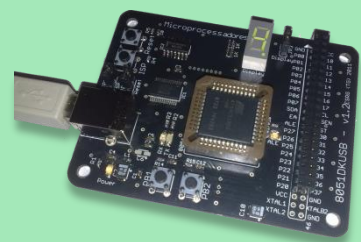
Interface

Porto 3



- Funcionamento idêntico ao do porto 1, do ponto de visto do I/O
- Funções alternativas do porto
 - Estas só podem ser ativadas se a *latch* contiver 1 (implica escrever 1 para o pino)

Port Pin	Alternate Function
P3.0	RxD (serial input port)
P3.1	TxD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt)
P3.3	$\overline{\text{INT1}}$ (external interrupt)
P3.4	T0 (Timer/Counter 0 external input)
P3.5	T1 (Timer/Counter 1 external input)
P3.6	$\overline{\text{WR}}$ (external Data Memory write strobe)
P3.7	$\overline{\text{RD}}$ (external Data Memory read strobe)



Características eléctricas

- Portos de Entrada/Saída

V_{OL} – tensão de saída nível lógico baixo

Portos 1,2,3

$V_{CC} = 4.5V \text{ to } 5.5V$	
0.3 V	$I_{OL} = 100 \mu A^{(4)}$
0.45 V	$I_{OL} = 1.6 \text{ mA}^{(4)}$
1.0 V	$I_{OL} = 3.5 \text{ mA}^{(4)}$

Porto 0

$V_{CC} = 4.5V \text{ to } 5.5V$	
	$I_{OL} = 200 \mu A^{(4)}$
	$I_{OL} = 3.2 \text{ mA}^{(4)}$
	$I_{OL} = 7.0 \text{ mA}^{(4)}$

V_{OH} – tensão de saída nível lógico alto

$V_{CC} = 5V \pm 10\%$	
$V_{CC} - 0.3$ V	$I_{OH} = -10 \mu A$
$V_{CC} - 0.7$ V	$I_{OH} = -30 \mu A$
$V_{CC} - 1.5$ V	$I_{OH} = -60 \mu A$

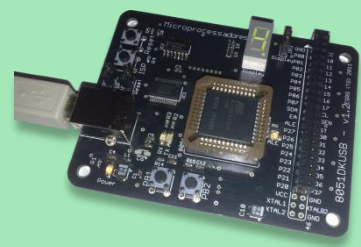
$V_{CC} = 5V \pm 10\%$	
	$I_{OH} = -200 \mu A$
	$I_{OH} = -3.2 \text{ mA}$
	$I_{OH} = -7.0 \text{ mA}$

Em condições de regime permanente (não transitórias), I_{OL} deve ser externamente limitada de modo a garantir:

I_{OL} máxima por pino de porto: 10mA

I_{OL} máxima por porto (8-bit): 15mA (P1, P2 e P3) e 26mA (P0)

I_{OL} total máxima para todos os pinos de saída: 71mA



Interface

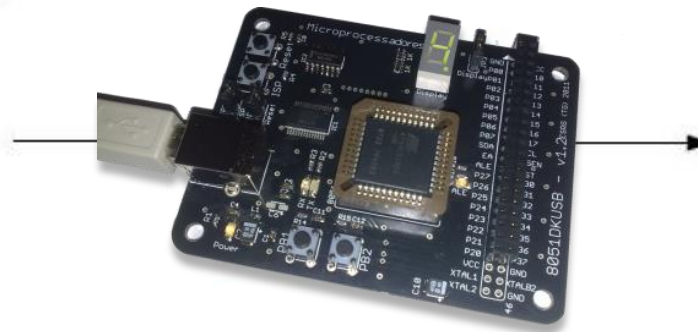
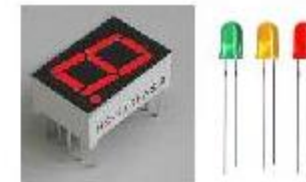
- Os microcontroladores permitem interagir com o mundo exterior e portanto são capazes de comunicar com outros dispositivos, como sensores e actuadores: motores, interruptores, teclados, *displays*, memórias e mesmo outros microcontroladores;
- Vários métodos de interface foram desenvolvidos tendo em consideração os vários critérios de desenho de circuitos, tais como as características pretendidas, custo, tamanho, peso, consumo de energia, fiabilidade, etc.
- A maioria dos desenhos com microcontroladores mistura vários métodos de interface. De um ponto de vista simples, um sistema com microcontrolador pode ser visto como um sistema que lê entradas, realiza operações de processamento e escreve para (controla) saídas.

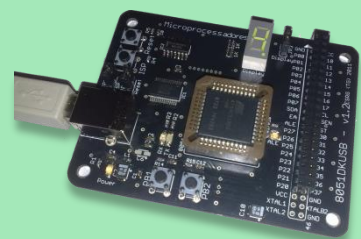
Interface

Dispositivos de Entrada

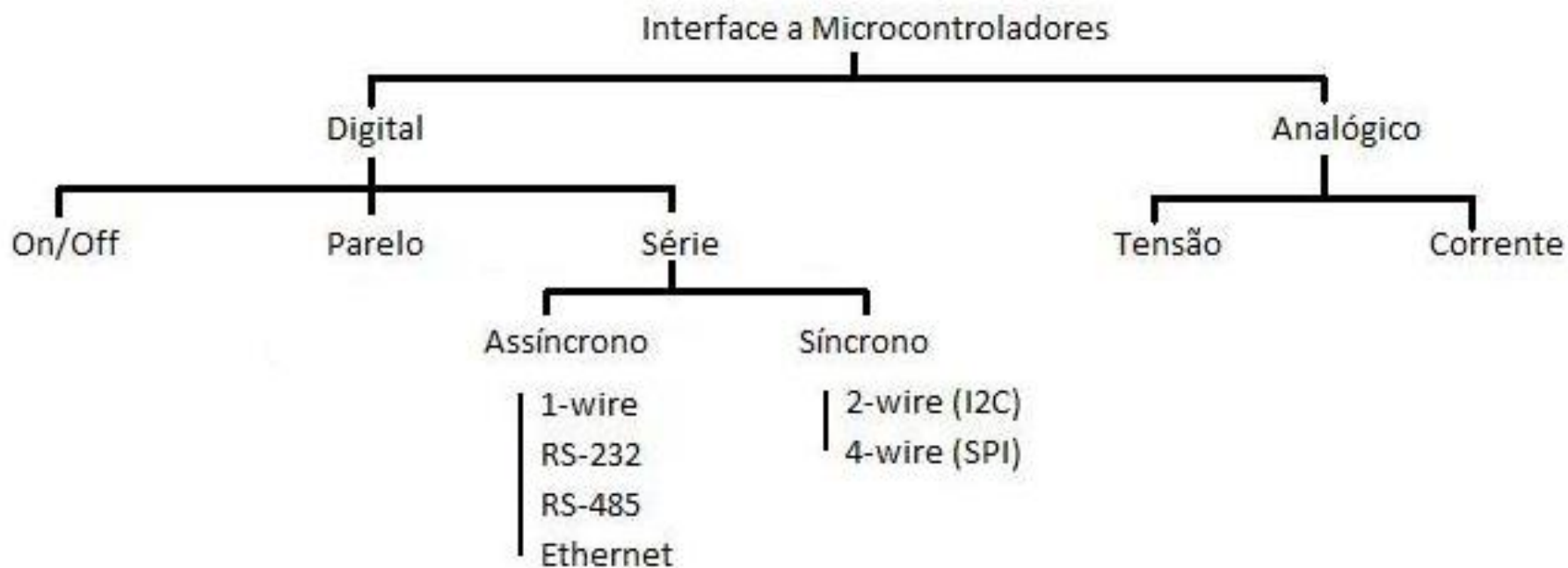


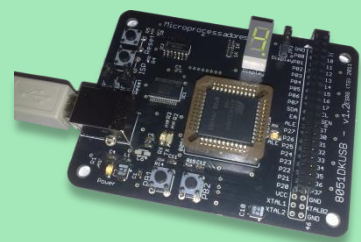
Dispositivos de Saída





Interface

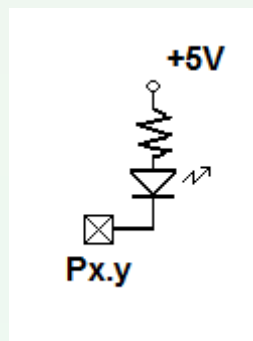
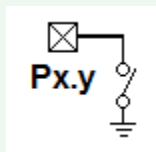
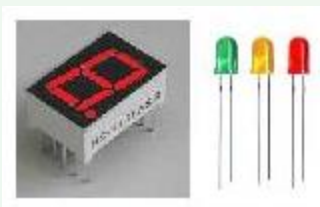




Leitura/Escrita Digital

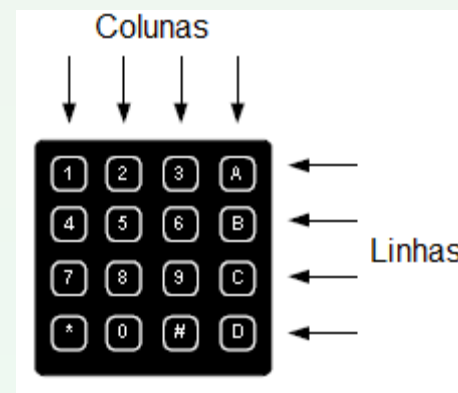
Vantagens:

- Interface muito simples;
- Custo reduzido (implementado no microcontrolador);
- Velocidades rápidas;
- Simplicidade de programação.

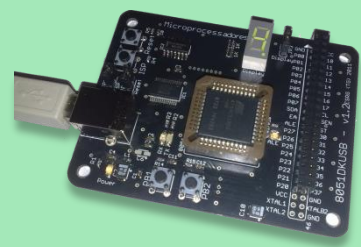


Desvantagens:

- Monitorização/Controlo digital (liga/desliga);
- Baixa distância: <50cm.
- Apenas um dispositivo pode ser monitorizado/controlado.

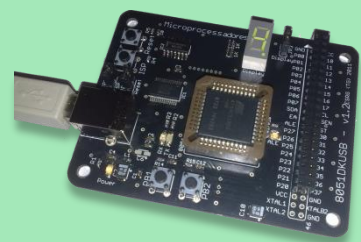


Para definir um pino como sendo de entrada é necessário escrever um “1” para a latch do porto respectivo.



Interface

- Uma forma muito eficaz de realizar o interface com sensores resistivos e capacitivos de baixa precisão é utilizar o LM555 (NE555 ou SA555);
- O objectivo é converter a grandeza física numa onda quadrada em que a frequência e/ou o *duty-cycle* são proporcionais ao valor dessa grandeza;
- O LM555 é um controlador de alta estabilidade e precisão capaz de produzir pulsos temporizados. Na operação “*monostable*”, o tempo de atraso é controlado por uma resistência externa e por um condensador a partir de um sinal de disparo. Na operação “*astable*”, a frequência e o *duty-cycle* são controlados por duas resistências externas e um condensador.



Interface

LM555

Nickname: "The IC Time Machine"

$$f = \frac{1.44}{(R_A + 2R_B)C}$$

$$t_{ON} = 0.693 \times (R_A + R_B) \times C$$

Aplicações:

- Temporizador de alta precisão
- Gerador de pulsos
- Gerador de atrasos (tempo)
- Temporizador sequencial

2. Astable Operation

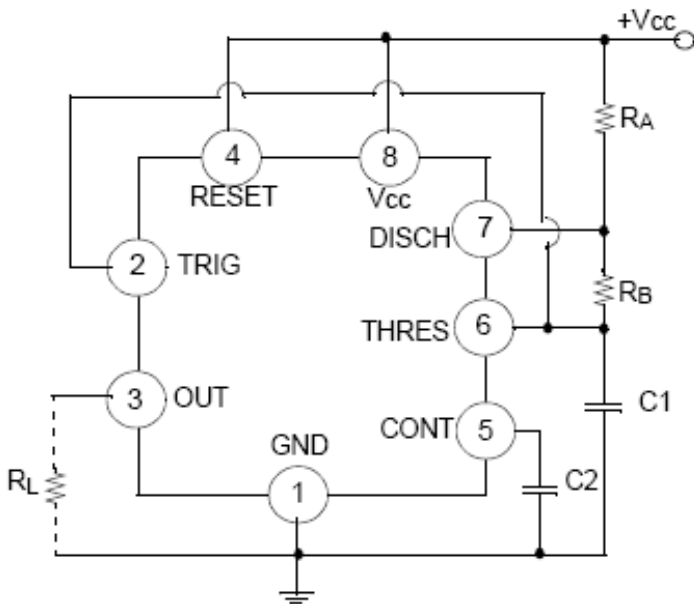


Figure 5. Astable Circuit

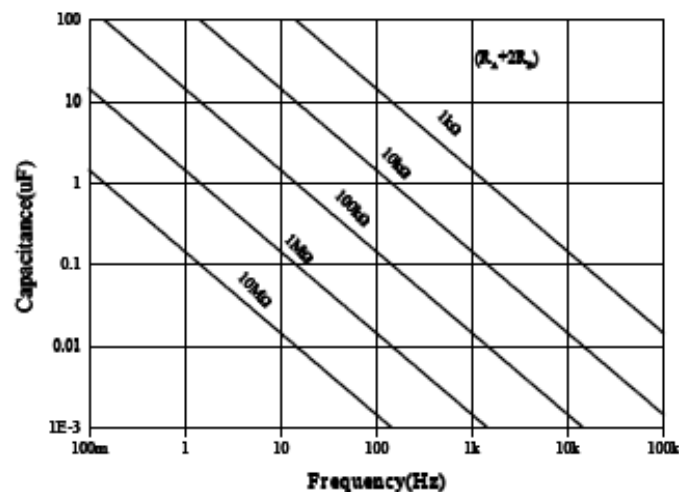
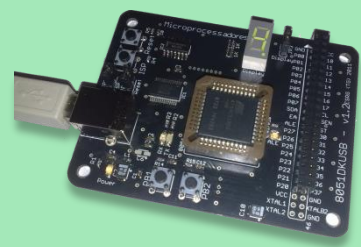
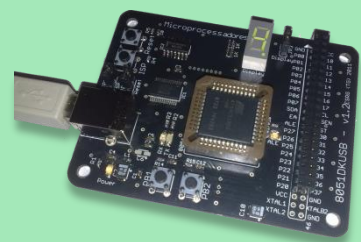


Figure 6. Capacitance and Resistance vs. Frequency



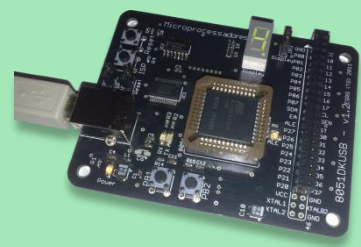
Interface

- Utilizamos um temporizador, ou o módulo de captura (P89C51Rx2), para determinar a frequência e/ou o *duty-cycle* da onda gerada pelo LM555;
- Quanto menor a frequência da onda maior será a precisão da leitura;
- Trata-se de um interface de 1-wire a um sensor, lento mas extremamente preciso e barato – pequenas distâncias;
- Quando discutirmos os modos de Captura e as Interrupções iremos implementar interfaces aos seguintes sensores:
 - **LDR** – Sensor de luminosidade - Resistência Dependente da Luz
 - **NTC/PTC** – Sensor de temperatura – Resistência Dependente da Temperatura
 - **Humidade** – Capacidade proporcional ao valor da humidade relativa



Interface

- Para interface directo com sinais analógicos (entrada ou saída) o microcontrolador utiliza ADCs (conversor Analógico para Digital) ou DACs (conversor Digital para Analógico);
- Há microcontroladores que têm estes conversores incorporados;
- O preço dos ADCs/DACs depende da sua resolução e sobretudo da sua velocidade de conversão;
- Para aplicações onde a velocidade não é um parâmetro chave (temperatura, luminosidade, distância, intrusão, humidade, etc) podemos encontrar ADCs/DACs muito baratos ($\ll 2\text{€}$);
- A característica principal destes é que o interface é série, regra geral SPI ou I2C;
- Recordar apenas que um ADC converte uma tensão de referência (normalmente 5V) para um equivalente digital, pelo que para realizar o interface com a maioria dos sensores é necessário acondicionar, filtrar, amplificar e amostrar o sinal proveniente do sensor.



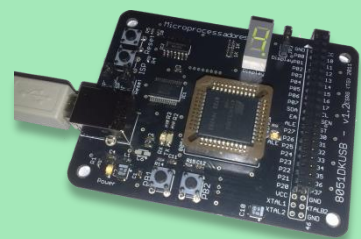
Exercício

Acender um de três LEDs (encarnado, amarelo ou vermelho). O controlo de qual o LED a acender é efectuado por um botão de pressão. Por defeito é o LED encarnado que se encontra aceso. Quando o botão é premido apaga o LED e acende o seguinte.

- Três saídas digitais – P2.0 a P2.2 – LEDs (encarnado, amarelo e verde)
- Uma entrada digital – P2.3 – botão de pressão

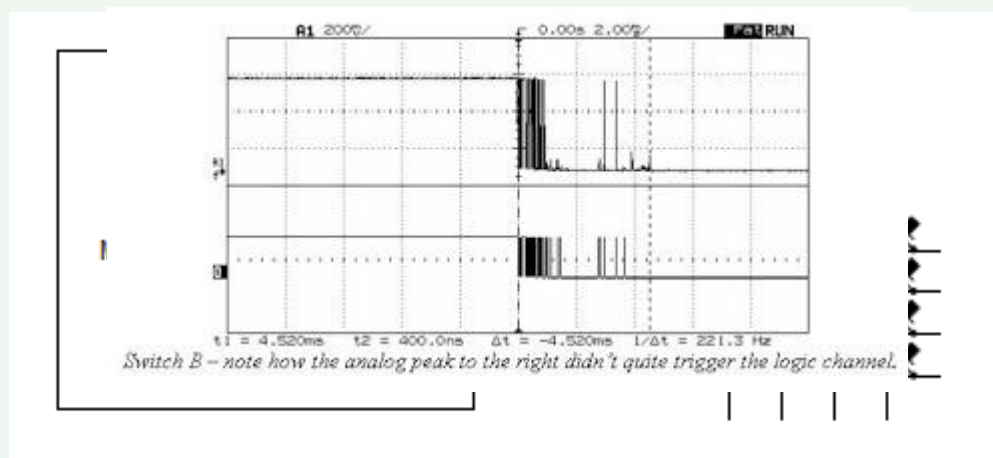
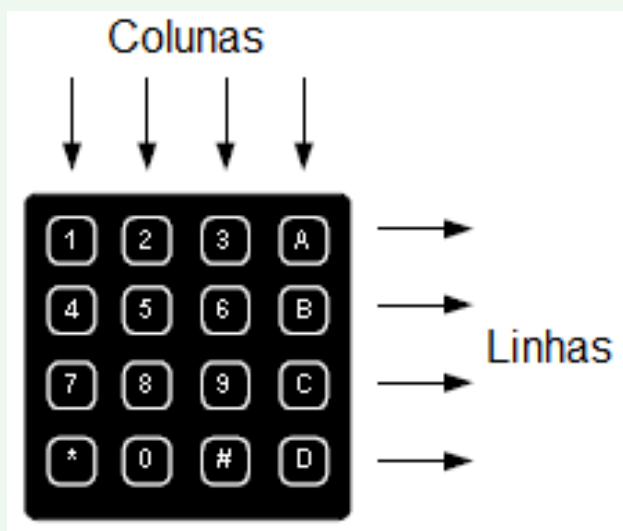
Porque ligar o hardware externo em lógica “inversa”?

- LEDs:
 - V_F – Tensão directa (2.1V);
 - I_F – Corrente directa (10mA);
 - $5-R \cdot I_F - V_F = 0 \Rightarrow R = 290\Omega$
- P89C51RD2
 - $I_{OH} = -30\mu A (P1:P3) / -3.2mA (P0)$
 - $I_{OL} = 1.6mA \text{ (máx. } 15mA) / 3.2mA$
- Interruptor (Switch)/Botão de Pressão
 - É um contacto mecânico \Rightarrow **bouncing**
 - O contacto móvel tem massa e uma constante de elasticidade. Oscilam várias vezes quando são fechados ou abertos.
 - O **debouncing** pode ser realizado por hardware (*latch* ou rede RC de atraso) ou por software. O tempo de **bouncing** varia. No caso dos interruptores e botões simples é estimado em 100ms.



Exercício

Implementar o interface com um teclado alfanumérico de 16 teclas. A tecla pressionada (0h a Fh) deve ser enviada para o *display* de 7-segmentos.



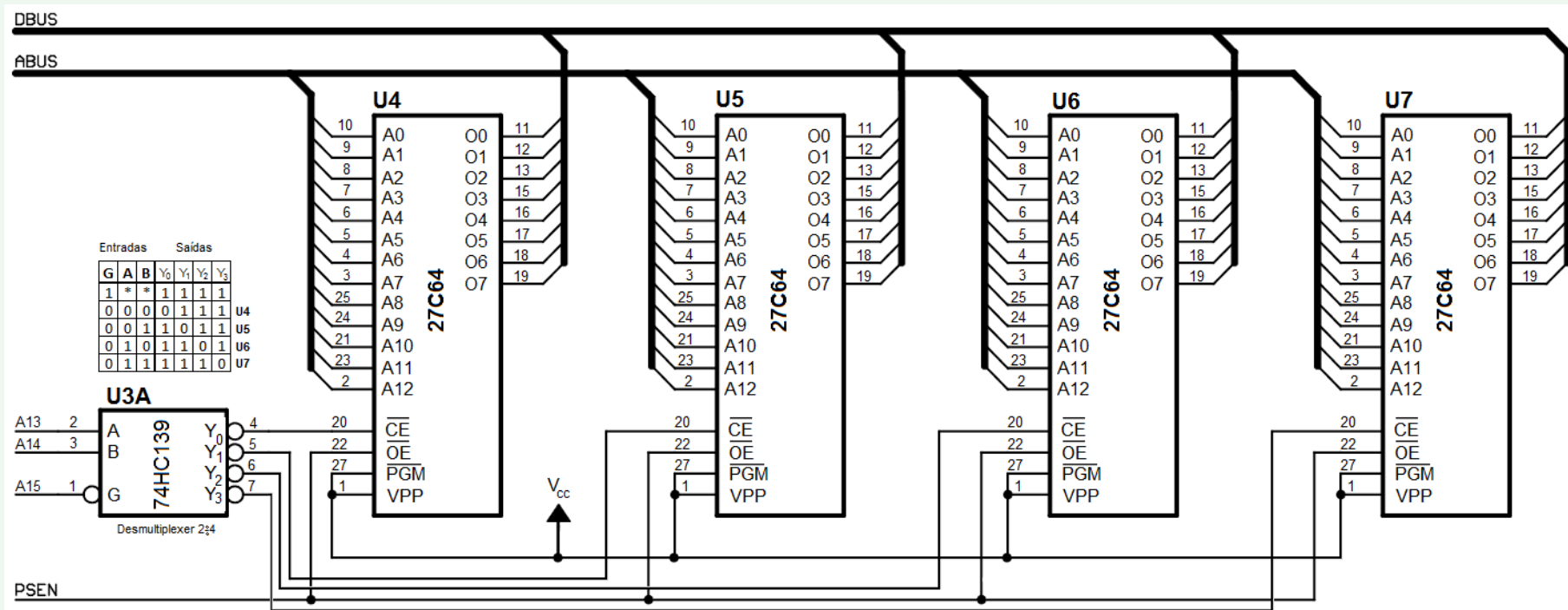
Utilizando a lógica “negativa” injecta-se um “0” numa das colunas (as restantes são mantidas a “1”) e lêem-se as quatro linhas. Se uma das linhas estiver a “0”, foi premida uma tecla dessa coluna, caso contrário procede-se para a coluna seguinte.

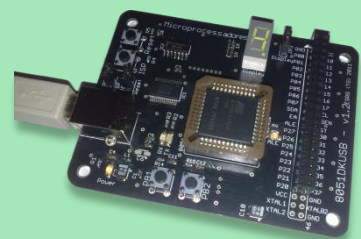
Como se trata de um contacto mecânico, é necessário efectuar o **debouncing** por software.

[illegible]

Interface Externo

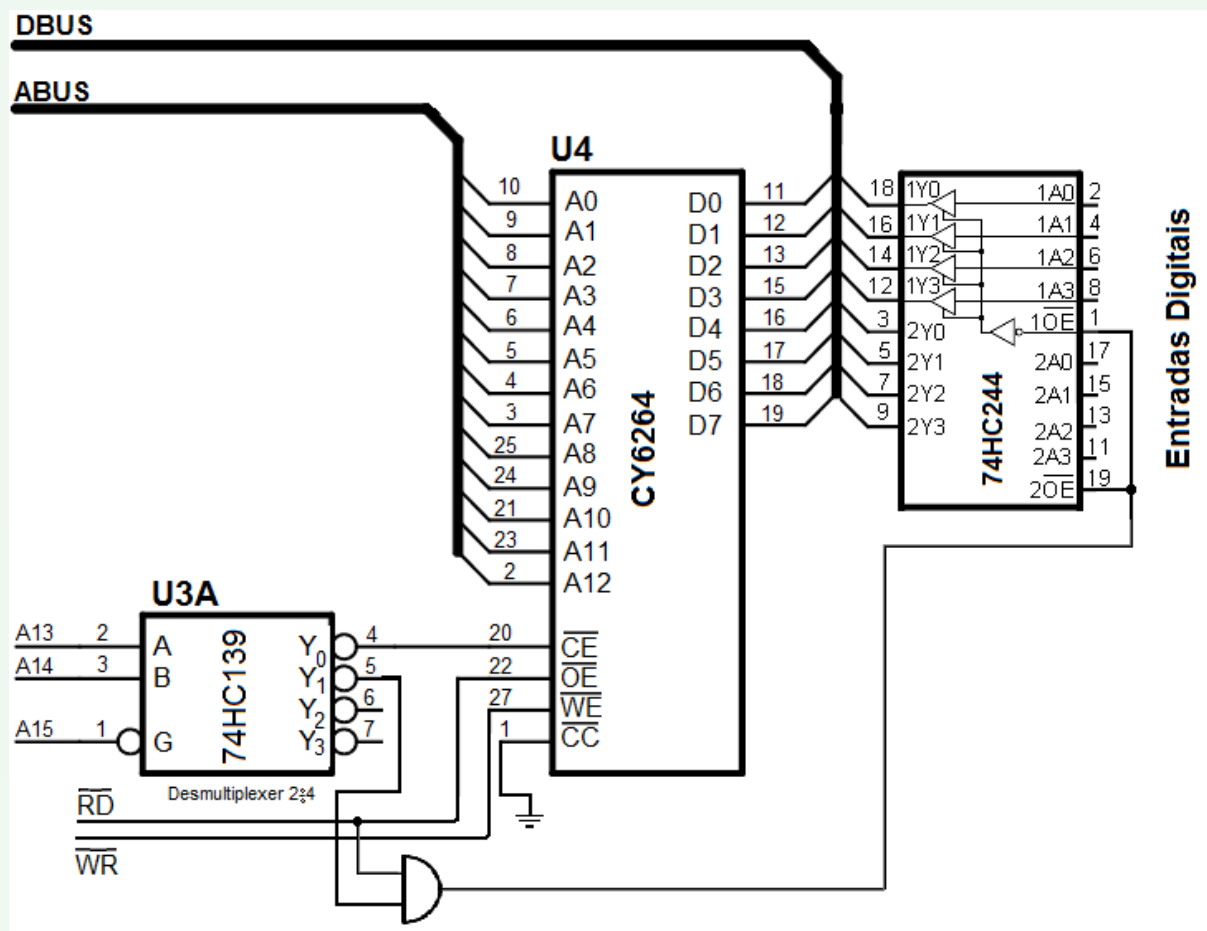
- Mapear memória

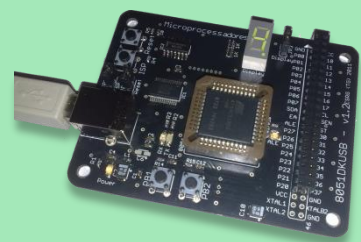




Interface Externo

- Mapear outros dispositivos





Interface LCD 16x2

- Interface com utilizador



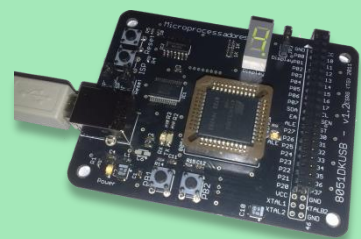
Todos os displays LCD têm um controlador, que permite controlar todas as operações do LCD.

O controlador mais utilizado é o HD44780 da Hitachi.

A ligação ao controlador é feita por:

- Um barramento de dados de 8 linhas (podemos também usar 4 linhas): D0..D7*
- Um barramento de controlo de 3 linhas RS, R/W e E (Register Select, R/W e sinal de Enable)*
- Led backlight – iluminação de fundo. Tensão entre os pinos 15 e 16 (I entre 100mA e 200mA)*

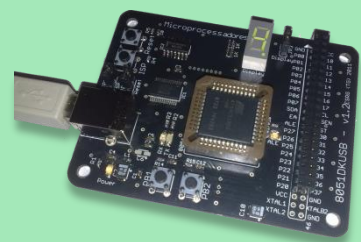
NOTA: Há programas (Proteus e Multisim por exemplo) que permitem simular o funcionamento de um 8051 ligado a diversos dispositivos e módulos de hardware típicos (LEDs, displays, LCDs, circuitos integrados, transístores bipolares, MOSFETs, etc).



Interface LCD 16x2

- Pinos

Pino	Função	Descrição
1	Alimentação	Terra ou GND
2	Alimentação	VCC ou +5V
3	V0	Tensão para ajuste de contraste
4	RS Seleção:	1 - Dado, 0 - Instrução
5	R/W Seleção:	1 - Leitura, 0 - Escrita
6	E Chip select	1 ou (1 → 0) - Habilita, 0 - Desabilitado
7	B0 LSB	Barramento de Dados
8	B1	
9	B2	
10	B3	
11	B4	
12	B5	
13	B6	
14	B7 MSB	
15	A (qdo existir)	Anodo p/ <i>LED backlight</i>
16	K (qdo existir)	Catodo p/ <i>LED backlight</i>



Interface LCD 16x2

- EN "Enable"

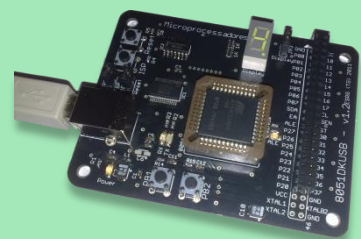
Esta linha de controlo serve para indicar ao LCD que lhe vão ser enviados dados. Para isso antes de os dados serem enviados o EN deve estar em 0, e só depois das outras linhas de controlo e dados estarem prontas o EN deve ser posto em 1, mantido nesse valor durante um tempo que depende do LCD e colocado de novo em 0. Este sinal indica ao controlador do LCD que pode ler o barramento de dados e de controlo.

- RS "Register Select"

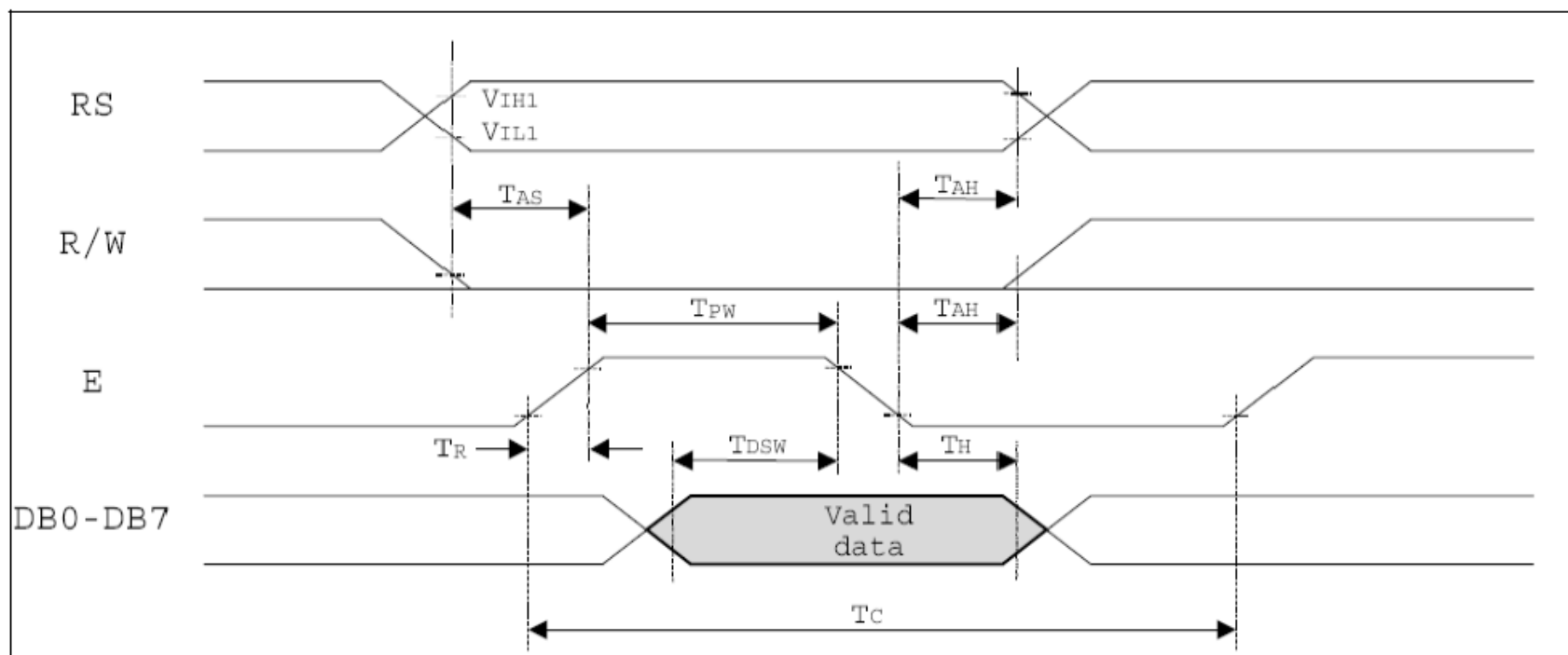
Quando RS=0, os dados são um comando ou uma instrução especial (limpa visor, posiciona cursor, etc.). Quando RS=1, os dados enviados são os caracteres de texto que devem ser visualizados no visor.

- RW "Read/Write"

Quando RW=0, a informação no barramento de dados é para ser escrita no LCD. Quando RW =1, pretende-se ler informação do LCD. De notar que só há uma instrução de leitura "Get LCD status". Todos os outros comandos são de escrita.



Interface LCD (Write)

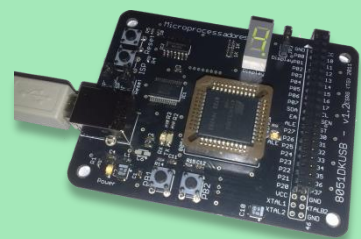


Write mode

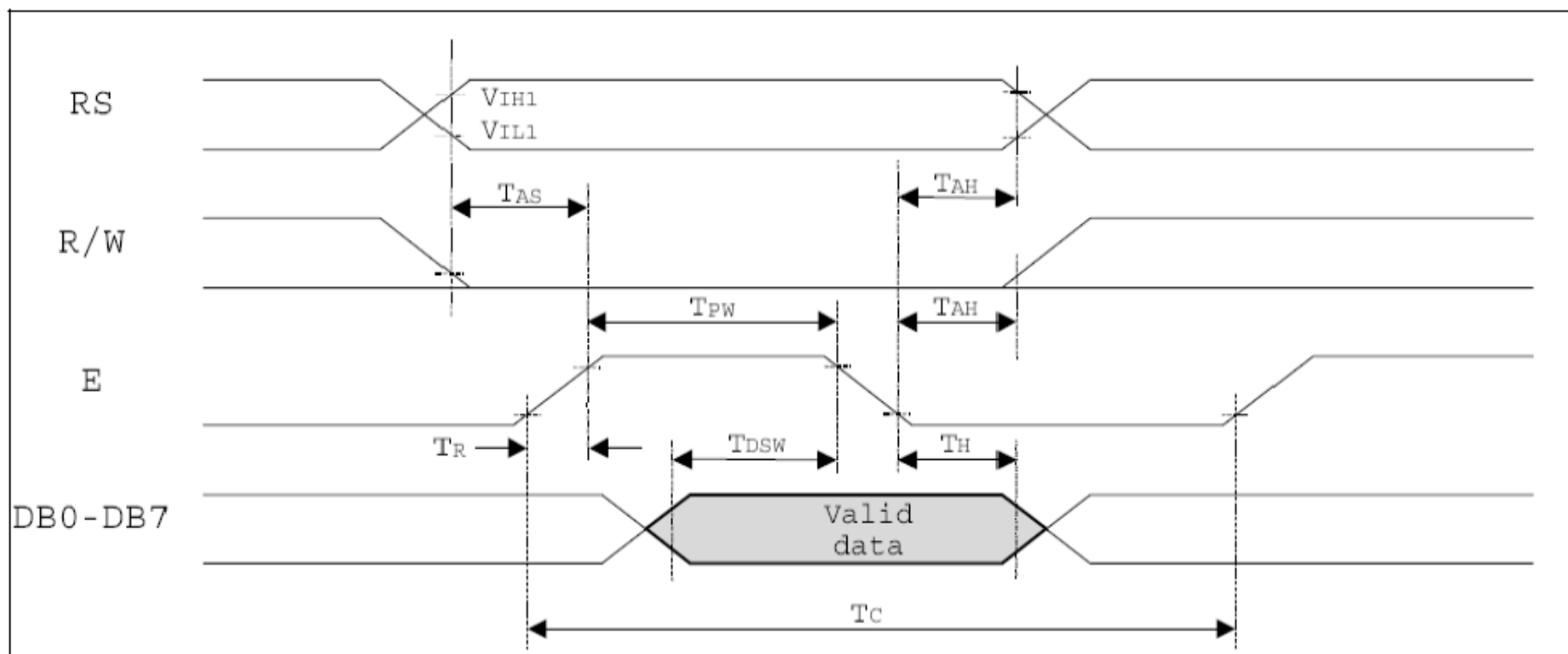
$$T_{PW}=140\text{ns}$$

$$T_{AS}=0\text{ns}$$

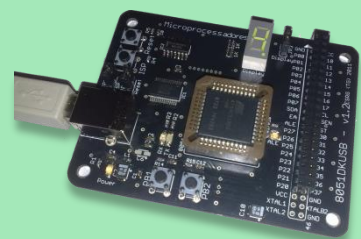
$$T_{DSW}=40\text{ns}$$



Interface LCD (Read)



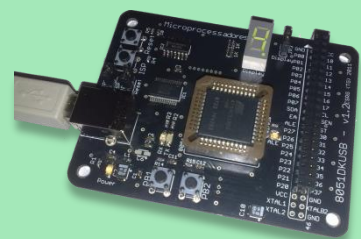
Read mode



Interface LCD 16x2

• Instruções/Comandos

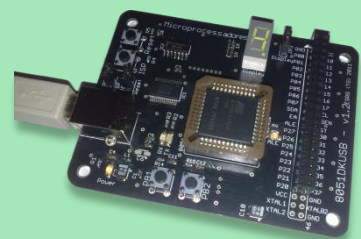
Instrução	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	Descrição e tempo de execução	t
Limpa Display	0	0	0	0	0	0	0	0	0	1	-Limpa todo o display e retorna o cursor para a primeira posição da primeira linha	1.6 mS
Home p/ Cursor	0	0	0	0	0	0	0	0	1	*	-Retorna o cursor para a 1. coluna da 1. Linha -Retorna a mensagem previamente deslocada à sua posição original	1.6 mS
Fixa o modo de funcionamento	0	0	0	0	0	0	0	1	X	S	-Estabelece o sentido de deslocamento do cursor (X=0 p/ esquerda, X=1 p/ direita) -Estabelece se a mensagem deve ou não ser deslocada com a entrada de um novo caracter (S=1 SIM, X=1 p/ direita) -Esta instrução tem efeito somente durante a leitura e escrita de dados.	40 uS
Controle do Display	0	0	0	0	0	0	1	D	C	B	-Liga (D=1) ou desliga display (D=0) -Liga(C=1) ou desliga cursor (C=0) -Cursor Pisca (B=1) se C=1	40 uS
Desloca cursor ou mensagem	0	0	0	0	0	1	C	R	*	*	-Desloca o cursor (C=0) ou a mensagem (C=1) para a Direita se (R=1) ou esquerda se (R=0) - Desloca sem alterar o conteúdo da DDRAM	40 uS



Interface LCD 16x2

• Instruções/Comandos

Instrução	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	Descrição e tempo de execução	t
Fixa o modo de utilização do módulo LCD	0	0	0	0	1	Y	N	F	*	*	-Comunicação do módulo com 8 bits(Y=1) ou 4 bits(Y=0) -Número de linhas: 1 (N=0) e 2 ou mais (N=1) -Matriz do caracter: 5x7(F=0) ou 5x10(F=1) - Esta instrução deve ser ativada durante a inicialização	40 uS
Posiciona no endereço da CGRAM	0	0	0	1	Endereço da CGRAM						-Fixa o endereço na CGRAM para posteriormente enviar ou ler o dado (byte)	40 uS
Posiciona no endereço da DDRAM	0	0	1		Endereço da DDRAM						-Fixa o endereço na DDRAM para posteriormente enviar ou ler o dado (byte)	40 uS
Leitura do Flag Busy	0	1	<i>B</i> <i>F</i>		AC						-Lê o conteúdo do contador de endereços (AC) e o BF . O BF (bit 7) indica se a última operação foi concluída (<i>BF=0 concluída</i>) ou está em execução (<i>BF=1</i>).	0
Escreve dado na CGRAM / DDRAM	0	1			Dado a ser gravado no LCD						- Grava o byte presente nos pinos de dados no local apontado pelo contador de endereços (<i>posição do cursor</i>)	40 uS
Lê Dado na CGRAM / DDRAM	1	1			Dado lido do módulo						- Lê o byte no local apontado pelo contador de endereços (<i>posição do cursor</i>)	40 uS

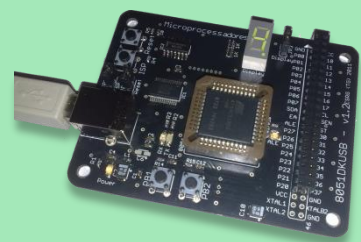


Interface LCD 16x2



Display	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
Line 1	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10 11 12 13 14 15 ...
Line 2	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50 51 52 53 54 55 ...

- O LCD pode ser visto como uma memória, nessa memória há endereços que aparecem no LCD (os das duas linhas visíveis) e outros que podemos utilizar para armazenamento.
- Para se posicionar o cursor na primeira coluna da 2ª linha
Instrução (80h + 40h) = **C0h** (80h – instrução para estabelecer o endereço da DDRAM +40h que é o 1º endereço da 2ª linha)



Interface LCD 16x2

- Sempre que é feito o reset ao módulo LCD deve ser executado o procedimento de inicialização, que consiste no envio de uma sequência de cinco instruções para configurar o modo de funcionamento do LCD.
- No caso de um visualizador de 2 linhas o procedimento é:
 1. Instrução 38h
 2. (aguardar 15ms)
 3. Instrução 38h
 4. (aguardar 15ms)
 5. Instrução 06h
 6. (aguardar “Busy=0” ou tempo para completar a instrução)
 7. Instrução 0Eh
 8. (aguardar “Busy=0” ou tempo para completar a instrução)
 9. Comando 01h
 10. (aguardar “Busy=0” ou tempo para completar a instrução)