

PCS 3432 – LABORATÓRIO DE PROCESSADORES

EXPERIÊNCIA 6:

CONEXÃO DE MÓDULO DISPLAY DE CRISTAL LÍQUIDO NA PLACA EXPERIMENTAL

***Autores: Prof. Dr. André Riyuiti Hirakawa
Prof. Dr. Carlos Eduardo Cugnasca***

2018 – v.01

1. OBJETIVO

Nesta experiência será apresentado e estudado o funcionamento de um dispositivo de saída muito utilizado, produzido por muitas empresas e que se tornou um padrão de mercado: o módulo com display de cristal líquido (LCD). Um módulo LCD típico será incorporado em uma das portas da Placa Experimental SDC68k, requerendo a elaboração de um programa para a geração dos sinais elétricos requeridos ao seu funcionamento.

2. DISPLAYS

2.1. Principais Tecnologias

Um dos displays mais antigos é o de tecnologia desenvolvida pela Borroughs Corporation, denominado display Nixie (Figura 1). Tratava-se de um tubo de vidro preenchido com um gás a baixa pressão, e com um anodo em formato de uma malha de fio, e múltiplos catodos na forma de dígitos ou outros caracteres. A aplicação de energia em um dos catodos provoca uma descarga de brilho laranja em sua volta.

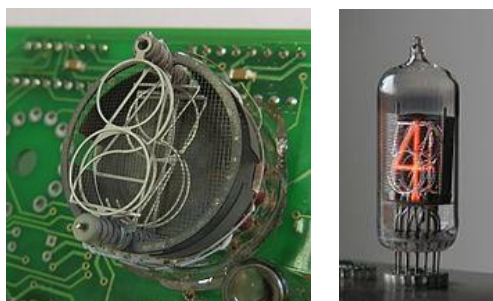


Figura 1 – Display Nixie

Fonte: extraído de https://en.wikipedia.org/wiki/Nixie_tube

Assim, quando um dos 10 números sobrepostos em planos diferentes acendia, os outros permaneciam apagados, permitindo que se visse através deles (a espessura destes números era bastante pequena), visualizando-se apenas o número aceso. Uma desvantagem deste sistema era que os números não ficavam à mesma distância do usuário (não estavam no mesmo plano), criando algum desconforto na leitura de um número com vários dígitos diferentes. Outro problema era que o acendimento dos números exigia altas tensões de operação com chaveamento rápido, algo que sempre foi problemático, embora necessário na implementação da técnica de multiplexação (essencial na redução de tamanho e custos dos *displays*).

Mais tarde surgiram os *displays* de sete segmentos, que tiveram bastante aceitação e apresentavam custos bem menores, estando disponíveis até hoje em grande variedade de formatos, cores e tipos (Figura 2).

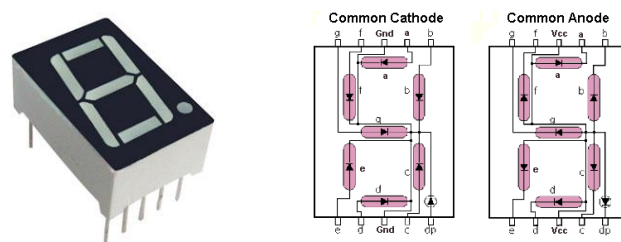


Figura 2 – Display de sete segmentos

Fonte: extraído de <https://arduinoexamples.wordpress.com/2012/10/06/display-de-7-segmentos-1-digito/>

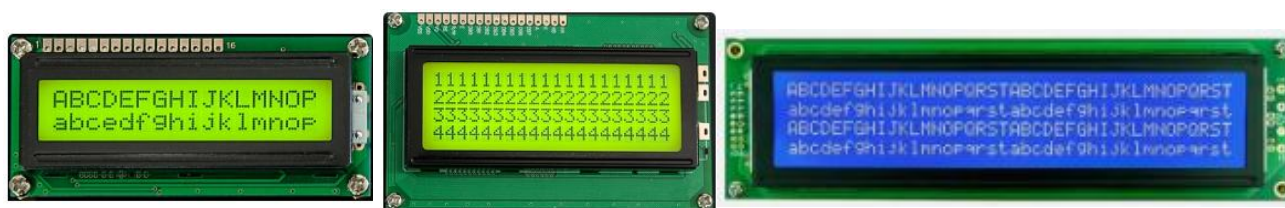
As principais tecnologias de *displays* de sete segmentos são:

- **Incandescentes:** operando com base em filamentos incandescentes, são disponíveis em várias cores e tamanhos. São os que podem oferecer maior intensidade de luz.
- **Neon ou plasma:** baseados na ionização de gases presos em bulbos, estão disponíveis na cor laranja-avermelhada (intrínseca aos gases utilizados). Exigem tensão alta para operação, o que dificulta sua multiplexação.
- **Fluorescentes:** são *displays* azuis ou verdes, que exigem tensões um pouco menores que as exigidas pelo tipo anterior, simplificando a multiplexação.
- **LEDs (*Light Emitting Diode*):** é uma tecnologia do início da era do semicondutor, sendo, assim, um dispositivo de estado sólido. Ele se aproveita do efeito de emissão de luz visível na região de junção dos semicondutores. A vantagem deste *display* reside no seu tamanho pequeno (embora hoje em dia já se disponha de *displays* de LEDs de tamanhos grandes), confiabilidade de operação em ambientes hostis e, principalmente, compatibilidade com tensões e correntes dos dispositivos da tecnologia de circuitos integrados disponíveis. Os LEDs são geralmente vermelhos, mas também são fabricados em outras cores como o amarelo, o verde e o azul.

- **Cristal Líquido (LCD):** nestes dispositivos algumas impurezas polarizadas são inseridas em um material que apresenta comportamento semelhante aos materiais em estado líquido, mas que consegue manter uma estrutura cristalina organizada em temperatura ambiente. Quando o material está com sua estrutura não perturbada, ele permite a passagem de luz pelo seu meio. Quando se aplica uma tensão de maneira a fazer com que as moléculas de impureza colocadas na substância se movam (elas são polarizadas), a estrutura cristalina é perturbada e as características ópticas do material se modificam, bloqueando a luz. Quando cessa o movimento das impurezas, a estrutura cristalina se recompõe, e o material volta a permitir a passagem de luz. Percebe-se, portanto, que podem existir dois tipos de *displays* de cristal líquido, um **reflexivo**, que trabalha com iluminação frontal (e possui uma superfície refletora por trás), e outro **transmissivo**, que apresenta uma iluminação na sua parte traseira (também conhecida como *backlight*). Estes dispositivos, por exigirem apenas tensão para mover as impurezas, consomem pouca energia do sistema de controle, mas apresentam, como desvantagem, um tempo de vida menor (por degeneração do cristal líquido), bem como menor confiabilidade e a maior sensibilidade à radiação ultravioleta encontrada em ambientes abertos (origem solar).

3. MÓDULOS DE DISPLAYS ALFANUMÉRICOS DE CRISTAL LÍQUIDO

Os módulos de *display* alfanuméricos (ou módulos LCD) têm se tornado extremamente populares, sendo utilizados em muitos produtos, como calculadoras, terminais de consulta de bancos, painéis informativos, sistemas de controle de acesso, painéis de equipamentos diversos, etc. Essa popularização do módulo LCD se deve às vantagens oferecidas, que são: conexão facilitada



com equipamentos diversos, fácil operação, baixo consumo e baixo custo (Figura 3).

Figura 3 – Módulos LCD 16 x 2, 20 x 4 e 40 x 4

Fonte: extraído de <http://agte.com.br/produtos/lcd-caracteres/modulo>

Atualmente existem muitos fabricantes que produzem módulos LCD que funcionalmente mantêm as mesmas características. As principais diferenças são quanto à conexão, número de linhas e colunas, dimensões físicas, cores, presença ou ausência de *backlight* e consumo. Assim, sequências de comandos e programas criados para um módulo LCD podem ser aplicadas em outros de uma mesma família, sem necessidade de maiores modificações. A Tabela 1 apresenta os tipos mais comuns desses módulos LCD que podem ser adquiridos no mercado.

Um módulo LCD padrão de mercado típico possui dois registradores: o **Registrador de Instruções** (também conhecido como registrador de comandos ou registrador de controle) e o **Registrador de Dados**. O **Registrador de Instruções** é utilizado, por exemplo, para:

- inicializar o módulo LCD e limpar a sua tela;
- mover o cursor (para a direita, esquerda, etc);
- deslocar todos os caracteres para a direita ou esquerda.

Já o **Registrador de Dados** permite que o valor de um caractere (normalmente representado em código ASCII) seja colocado na posição corrente do cursor.

Tabela 1 – Tipos Comuns de Módulos LCD

Número de Colunas	Número de Linhas	Código
8 Colunas	2 Linhas	8 x 2
16 Colunas	1 Linha	16 x 1
	2 Linhas	16 x 2 (*)
	4 Linhas	16 x 4
20 Colunas	2 Linhas	20 x 2 (*)
	4 Linhas	20 x 4
40 Colunas	2 Linhas	40 x 2
	4 Linhas	40 x 4

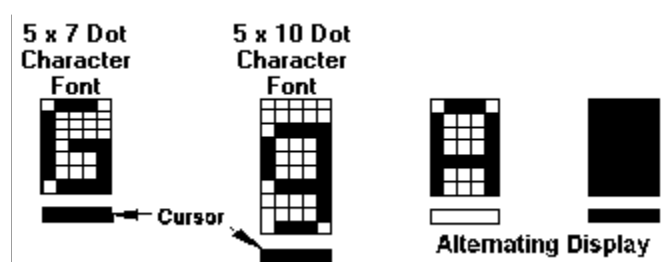
(*) Modelos utilizados na experiência

Fonte: <http://agte.com.br/produtos/lcd-caracteres/modulo>

3.1. Códigos de Programação do Módulo LCD

O módulo LCD a ser utilizado na presente experiência representa cada caractere exibido por meio de uma **matriz de pontos**. Um exemplo de montagem de caracteres a partir da matriz de pontos é apresentado na Figura 4 [16]. A sua programação é feita por meio de códigos de 8 bits (**comandos**) enviados ao seu **Registrador de Instruções**. Os módulos LCD permitem que o programador defina alguns caracteres, como caracteres com acentuação e outros símbolos gráficos.

Figura 4 – Montagem dos Caracteres do Display



Durante o desenvolvimento de programas para a programação do módulo LCD, a representação dos códigos no formato hexadecimal equivalente é mais simples de ser utilizada do

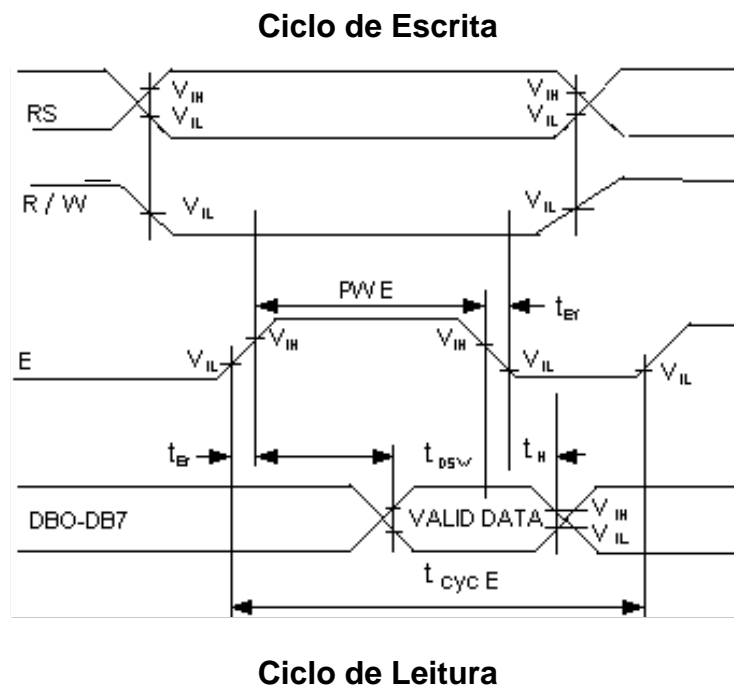
que a representação em código binário. Os principais códigos de programação do módulo LCD, em formato hexadecimal, são listados no [Apêndice I](#), enquanto que a criação de novos caracteres é apresentada no [Apêndice II](#).

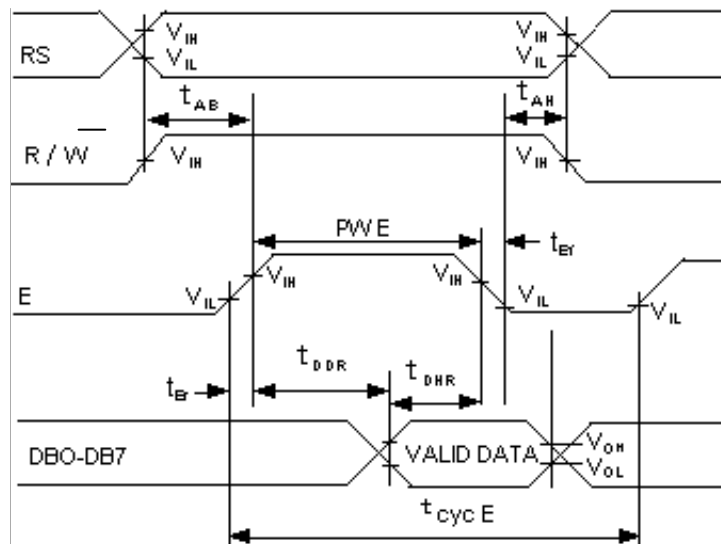
4. SINAIS DE INTERFACEAMENTO COM O MÓDULO LCD

Um *display* pode exibir caracteres alfanuméricos ou símbolos diversos, cujos códigos são escritos no seu Registrador de Dados para que a respectiva exibição seja efetuada. Entretanto, para que isso seja possível é necessário que ocorra uma correta manipulação dos sinais elétricos de interfaceamento com o módulo LCD.

Os ciclos de envio de comandos e dados para o módulo LCD são mostrados na Figura 5. O **Registrador de Instruções** é acessado quando os sinais são: **RS = 0** e **R/W = 0**, e o **Registrador de Dados**, quando os sinais são: **RS = 1** e **R/W = 0** [16].

Figura 5 – Ciclos de Escrita e de Leitura do Display





Conforme é mostrado na Figura 5, as seguintes condições devem ser satisfeitas para a **borda de descida** do sinal "E", indicando o instante da escrita:

- $\overline{R/W} = 0$, indicando **escrita**;
- dados estáveis na via de dados;
- **RS = 0**, indicando um **comando** para o *display* (p.ex: limpa o *display*) e **RS = 1** indica um **dado** a ser **escrito** no *display* (um caractere alfanumérico).

Assim, para se escrever um byte de **controle** no módulo LCD recomenda-se a seguinte sequência:

1. **RS = 0, $\overline{R/W} = 0$, E = 0**
2. **byte de comando na via de dados**
3. **RS = 0, $\overline{R/W} = 0$, E = 1**
4. **RS = 0, $\overline{R/W} = 0$, E = 0, gerando a borda de descida**
5. **RS = 0, $\overline{R/W} = 1$, E = 0**

Para se escrever um byte de **dados** no módulo LCD recomenda-se a seguinte sequência:

1. **RS = 1, $\overline{R/W} = 0$, E = 0**
2. **byte de dado na via de dados**
3. **RS = 1, $\overline{R/W} = 0$, E = 1**
4. **RS = 1, $\overline{R/W} = 0$, E = 0, gerando a borda de descida**
5. **RS = 1, $\overline{R/W} = 1$, E = 0**

Entre cada escrita na porta paralela, deve-se colocar um **atraso** da ordem de **milissegundos (em geral 15 ms)**, pois o módulo LCD é um dispositivo mais lento do que o microprocessador. Uma alternativa a esse atraso é a leitura do **Busy Flag** do módulo LCD, que indica quando uma operação foi concluída. Porém, o seu uso depende que o circuito de interligação do módulo LCD

seja implementado com a via de dados bidirecional, possibilitando a leitura de informações do módulo LCD. Essa implementação é mais robusta por não depender da velocidade de cada processador utilizado. Na presente experiência esse recurso **não será utilizado**, pois o módulo LCD foi conectado na interface para impressora da Placa Experimental (padrão Centronics), possibilitando apenas a escrita no módulo LCD. Outras informações podem ser obtidas em [16].

4.1 Inicialização do Display

Toda vez que o módulo LCD é alimentado, um processo de inicialização deve ser executado. Este procedimento consiste no envio de um grupo de instruções para configurá-lo no modo de execução normal. Para inicializá-lo, uma possível sequência de bytes de controle é:

\$38, \$38, \$38, \$06, \$0C, \$01

Todo o processo de inicialização é realizado com **RS** e **R/W** ambos em nível **0**.

O código hexadecimal inicial **\$38** é enviado mais de uma vez por recomendação dos fabricantes, colocando o módulo LCD no modo de operação 2 linhas, com caracteres 5 x 7, e comunicação na via de dados com 8 bits (via de dados de 4 bits também pode ser utilizada [16]). O controle **\$01** limpa o módulo LCD. Os demais códigos encontram-se descritos na Tabela 3 do Apêndice I.

Após o módulo LCD ser inicializado, pode-se escrever um **dado**. Por exemplo, após se enviar o byte **\$41**, será exibido o caractere "A" (**\$41** é o seu código ASCII).

Informações importantes sobre as opções de inicialização do módulo LCD se encontram no Apêndice I.

5. CONEXÃO DO DISPLAY NA PORTA PARALELA DO MC68000

O módulo LCD pode ser conectado de diversas formas a um microprocessador, como [16]:

- Mapeado em memória, conectando-se à via de dados e aos sinais de controle do processador. Nesse caso, o módulo LCD receberá um endereço, e os sinais de controle são gerados a partir dos sinais da via de controle.
- Conectado em uma porta paralela, utilizando-se 4 ou 8 bits da via de dados. Nesse caso, os sinais de controle devem ser gerados por software.

A presente experiência se utiliza da **segunda alternativa**.

O hardware necessário para a realização da experiência é fornecido já montado na Placa Experimental do MC 68000, pronto para ser utilizado, com o módulo LCD conectado na interface para impressora da Placa Experimental (padrão Centronics), possibilitando apenas a escrita no módulo LCD. Essa interface é baseada no dispositivo MC 68230 [18]. O esquema elétrico da

ligação do módulo LCD é mostrado na Figura 6, e a conexão dos principais sinais é apresentada na Tabela 2.

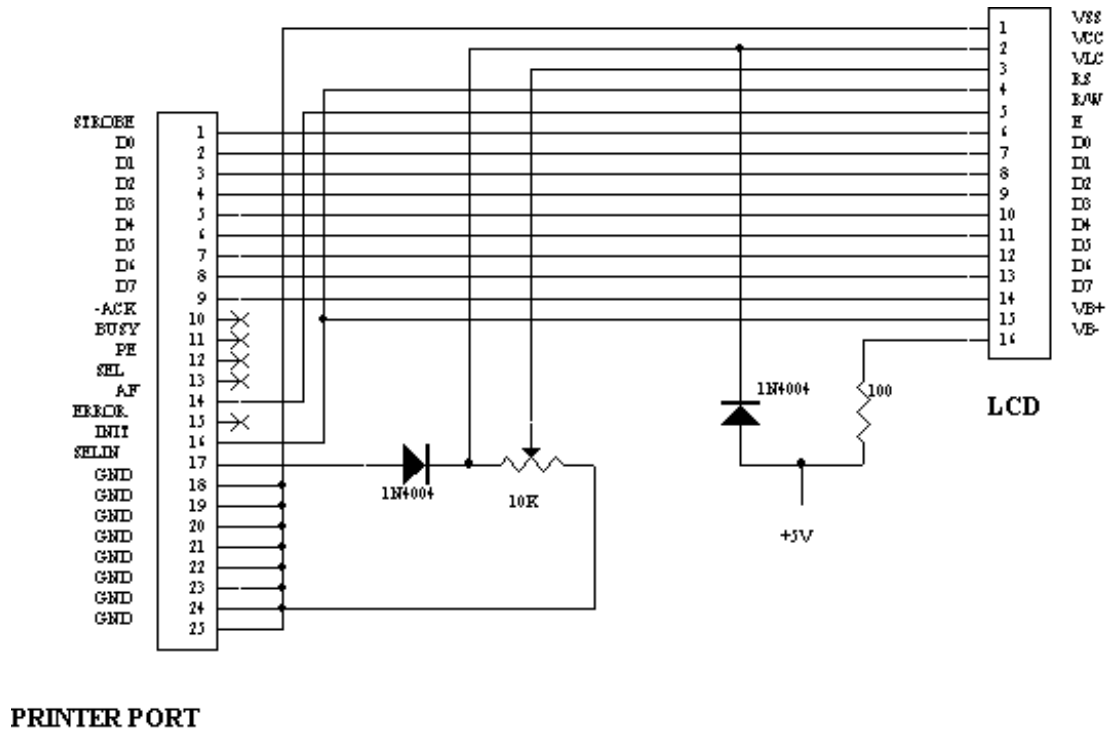
Na Placa Experimental, o seu **conector JP7** deve estar fechado com um **jumper** para que os sinais de **J7** sejam passados para **J8** (DB 25), que é o conector da porta paralela [06].

Tabela 2 – Conexão dos Principais Sinais do módulo LCD

Módulo LCD			Conector Impressora (DB 25)	MC 68230		Outros
Sinal	Descrição	Conector Pino		Porta A	Porta B	
				Conector J7 Pino (sinal)		
VSS (0 V)	Alimentação	1	18 ... 25			Cabo externo Fio Preto Terra da Fonte
VCC (+5 V)	Alimentação	2	---			Cabo externo Fio Vermelho + 5 V da Fonte
VLC	Ajuste de brilho (<i>trimpot</i>)	3	---			
RS	Seleção de Registrador (Instruções/Dados)	4	16		19 (H4)	
$\overline{R/W}$	Leitura/Escrita Mantido sempre em “0” (só escrita)	5	14 (vazio)			Fio externo Conectar no Terra: pino 1 de J2
E	Habilitação	6	1 (Strobe)	17 (H2)		
D0	Dados	7	2 (D0)	2 (PA0)		
D1		8	3 (D1)	4 (PA1)		
D2		9	4 (D2)	6 (PA2)		
D3		10	5 (D3)	8 (PA3)		
D4		11	6 (D4)	10 (PA4)		
D5		12	7 (D5)	12 (PA5)		
D6		13	8 (D6)	14 (PA6)		
D7		14	9 (D7)	16 (PA7)		
A	Anodo do <i>backligh</i> (*)	15	10 ...15 (vazio)			
K	Catodo do <i>backligh</i> (*)					

(*) se existir

Figura 6 – Ligação do Módulo LCD na Porta Paralela da Placa Experimental



5. REFERÊNCIAS

- [01] HIRAKAWA, A.R.; CUGNASCA, C.E. **PCS 3432 – Laboratório de Processadores, Experiência 1 - Familiarização com a Placa Experimental do Microprocessador MC 68000.** PCS - EPUSP, 2017. (*)
- [02] HIRAKAWA, A.R.; CUGNASCA, C.E. **PCS 3432 – Laboratório de Processadores, Experiência 2 - Programação em Linguagem Assembly para a Placa Experimental do Microprocessador MC 68000.** PCS - EPUSP, 2017. (*)
- [03] HIRAKAWA, A.R.; CUGNASCA, C.E. **PCS 3432 – Laboratório de Processadores, Experiência 3 - Ferramentas Para o Desenvolvimento de Software: Linguagem Assembly e Simulador.** PCS - EPUSP, 2017. (*)
- [04] HIRAKAWA, A.R.; CUGNASCA, C.E. **PCS 3432 – Laboratório de Processadores, Experiência 4: Periféricos de Entrada e Saída: Acionamento de Motor de Passo Utilizando a Linguagem C.** PCS - EPUSP, 2017. (*)
- [05] HIRAKAWA, A.R.; CUGNASCA, C.E. **PCS 3432 – Laboratório de Processadores, Experiência 5: Processamento de Exceções na Placa Experimental para o Uso de Rotinas Utilitárias.** PCS - EPUSP, 2017. (*)
- [06] ARNEWSH INC. **SBC68K-User's Manual**, Revision 1.1, Arnewsh Inc. C.O., USA, 1990. (*)
- [07] ARNEWSH INC. **SBC68K - 68000 Single Board Computer User Manual**, Arnewsh Inc. C.O., USA, 1990. (*)
- [08] MOTOROLA. **M68000 16/32-Bit Microprocessor Programmer's - Reference Manual.** N.J., EUA, Prentice Hall, 4a. edição, 1984. 218p. (*)
- [09] MOTOROLA. **M68000 Family Reference.** EUA, 1988. (*)

- [10] MOTOROLA. **M68000 8-/16-/32-Bit Microprocessors User's Manual**. N.J., EUA, 9ª edição, 1993. (*)
- [11] MOTOROLA. **M68000 Family Programmer's Reference Manual (Part 1)**, N.J., EUA, 1992. (*)
- [12] MOTOROLA. **M68000 Family Programmer's Reference Manual (Part 2)**, N.J., EUA, 1992. (*)
- [13] FONDSE, P. J. **68000/020 Integrated Development Environment**, v 3.30, 2015. <<http://home.kpn.nl/pj.fondse/ide68k/>>
- [14] FONDSE, P. J. **IDE68K-getting_started**, v 3.30, 2015. <<http://home.kpn.nl/pj.fondse/ide68k/>>
- [15] FONDSE, P. J. **IDE68k Help**, help do aplicativo, v 3.30, 2015.
- [16] ALFACOM. **Módulos Multi-Matrix**. Manual de Utilização. (*)
- [17] CUGNASCA, C.E.; ZERBINI, R.C. **Experiência nº 3 - Familiarização com Periféricos de Entrada e Saída: Interface com Display**. EPUSP, 1989.
- [18] MOTOROLA. **MC68230 - Peripheral Interface and Timer (PI/T)**. Motorola Inc. (*)

(*) Disponíveis no formato pdf no site da disciplina.

6. PARTE PRÁTICA

A parte prática prevê o teste e a modificação do programa proposto sobre o tema da Experiência.

Para tanto, o planejamento deve incluir a elaboração de todos os programas com a respectiva documentação, que deverá incluir:

- **Diagramas em Blocos** para programas elaborados em linguagem *Assembly*.
- **Diagramas de Nassi-Shneiderman (NSD)** para programas elaborados em linguagem C. Utilizar o Programa-Exemplo do [Apêndice III](#) como referência para o estilo dos demais programas a serem elaborados.
- Incluir **comentários** em todos os programas.

As etapas a seguir deverão ser seguidas para a elaboração dos programas.

6.1. 1ª Etapa: Programa-Exemplo do Apêndice III

- Estudar a interligação de um módulo LCD na Placa Experimental, explicando-a no relatório.
- Analisar e testar o programa-exemplo, que inicializa, controla e apresenta dados no módulo LCD. A mensagem a seguir é apresentada duas vezes, após o que o programa pergunta ao usuário se ele deseja repetir o teste ou retornar o controle ao Programa Tutor:

PCS 3432
1o.Semestre 2018

Avaliar o programa, identificando os tipos de controle encaminhados ao módulo LCD.

6.2. 2ª Etapa – Elaborar Programa de Teste do Módulo LCD

- Elaborar um programa que permita ao usuário escolher os **comandos** e os **dados** a serem enviados ao módulo LCD. Por exemplo, solicitar a seleção no **terminal** das opções (*loop* infinito):

1- Enviar Comando
2- Enviar Dado
Digite a opção:

Caso 1 seja digitado, enviar a mensagem:

Digite o código do Comando (em hexadecimal):

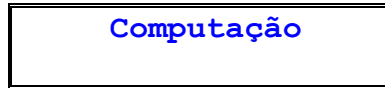
Caso 2 seja digitado, enviar a mensagem:

Digite o código do Caractere (em hexadecimal):

Após cada opção ser digitada e executada voltar ao início do programa (*loop* infinito). Avaliar o programa e anotar os resultados no relatório. Testar o programa com os principais comandos apresentados na Tabela 3 do [Apêndice I](#).

6.3. 3ª Etapa: Programação de Caracteres Especiais

- a) Consultar o [Apêndice II](#) e criar os caracteres especiais ç e ã. Elaborar um programa que exiba a mensagem no display:



6.4. 4ª Etapa - Itens Específicos para cada Turma

Turma 1: Fazer um programa que solicita a digitação de uma mensagem de **40 caracteres** e a envia ao módulo LCD de forma circulante, da direita para a esquerda (similar aos equipamentos existentes em muitas lojas).

Turma 2: Fazer um programa que solicita a digitação de duas mensagens de até **16 caracteres** cada, e as exibe uma em cada linha, de forma **piscante** à taxa de 1 segundo. Enquanto a primeira linha é exibida, a segunda é apagada e vice-versa.

APÊNDICE I – CARACTERÍSTICAS DO MÓDULO DISPLAY

O módulo LCD utilizado na experiência possui duas linhas de 16 caracteres cada. Contudo, internamente ele possui memória para representar 20 caracteres por linha, deixando de exibir os demais. Quando se utiliza esse módulo LCD por outro da mesma família, mas com 20 caracteres por coluna, todos os caracteres serão exibidos.

Toda vez que o módulo LCD é alimentado, um processo de inicialização deve ser executado, com as configurações de interesse a cada aplicação. Ao ser ligado, o display acende todos os pontos de todos os caracteres da primeira linha, deixando a segunda linha apagada, indicando que nenhuma configuração foi efetuada. O procedimento de inicialização consiste no envio de um grupo de instruções para configurar o módulo LCD no modo de execução desejado. Uma sequência de inicialização típica é a que limpa o *display* e desloca o cursor para o início da 1ª. Linha. Para tanto, a seguinte sequência de comandos, obtidos na Tabela 3, deve ser enviada:

\$38, \$38, \$38, \$06, \$0C, \$01

O código hexadecimal inicial **\$38** é enviado mais de uma vez para garantir que o módulo LCD assuma o modo de operação de tamanho 8 bits (recomendado em manuais de fabricantes). O controle **\$01** limpa o *display*.

Uma vez que o módulo LCD esteja inicializado, pode-se escrever um dado. Por exemplo, após se enviar o byte **\$41**, o módulo LCD apresentará o caractere "A" (**\$41** é o seu código ASCII).

Tabela 3 – Principais Códigos de Programação do Display

DESCRIÇÃO DO COMANDO		CÓDIGO (HEX)
Condições de Utilização	1 Linha 5 x 7	30
	2 Linhas 5 x 7	38
Controle do Display e Cursor	Limpa Display e Retorna o Cursor	01
	Coloca Cursor na Posição Inicial (Home)	02
	Display Apagado	08
	Desloca somente o Cursor para a Esquerda	10
	Desloca o Cursor e a Mensagem para a Esquerda	18
	Desloca somente o Cursor para a Direita	14
	Desloca o Cursor e a Mensagem para a Direita	1C
	Display Aceso sem Cursor	0C
	Display Aceso com Cursor Fixo	0E
	Display Aceso com Cursor Intermitente	0F
	Desloca o Cursor para a 1ª. linha, 1ª. posição	80
	Desloca o Cursor para a 2ª. linha, 1ª. posição	C0
Modos de Escrita	Escreve deslocando o Cursor para a Esquerda	04
	Escreve deslocando o Cursor para a Direita	06
	Escreve deslocando a Mensagem para a Esquerda (*)	07
	Escreve deslocando a Mensagem para a Direita (*)	05
	(*) Cursor fica Imóvel	

Cada caractere pode ser endereçado individualmente. A Tabela 4 mostra o endereço de cada um deles, para módulo LCD **16 x 2** e **20 x 2**. Para outros módulos deve-se consultar o manual [16].

Tabela 4 – Endereços dos Caracteres dos Módulos LCD 16 x 2 e 20 x 2 (em hexadecimal)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
LINHA 1	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	90	91	92	93
LINHA 2	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	D0	D1	D2	D3
Válido para módulos 16 x 2 e 20 x 2																Apenas para módulos 20 x 2				

Para endereçá-los é preciso enviar um byte de controle contendo o valor da posição do endereço. Por exemplo, para se endereçar o 7º. caractere da 1ª. linha, envia-se o byte **\$86** como controle. Após receber este caractere de controle, o Registrador de Endereço do módulo LCD é incrementado automaticamente para receber o próximo caractere (após receber o caractere, portanto, o Registrador de Endereço estará com o valor **\$87** nesse exemplo).

Vários caracteres podem ser enviados para serem exibidos no *display*. Os mais utilizados estão na Tabela 5.

Tabela 5 – Caracteres mais utilizados

	2	3	4	5	6	7
0	(espaço)	0	@	P	`	p
1	!	1	A	Q	a	q
2	"	2	B	R	b	r
3	#	3	C	S	c	s
4	\$	4	D	T	d	t
5	%	5	E	U	e	u
6	&	6	F	V	f	v
7	'	7	G	W	g	w
8	(8	H	X	h	x
9)	9	I	Y	i	y
A	*	:	J	Z	j	z
B	+	;	K	[k	{
C	,	<	L	¥	l	
D	-	=	M]	m	}
E	.	>	N	^	n	->
F	/	?	O	_	o	<-

Para se enviar a letra **H** ao *display*, por exemplo, deve-se enviar o valor **\$48** como dado. A tabela completa está no manual do *display* [16], reproduzida na Tabela 6. Pode-se constatar que a maior parte dos caracteres segue a codificação ASCII (por exemplo: **A** – **\$41**, **a** – **\$61**, etc). Contudo, outros caracteres podem ser exibidos, como letras gregas e até mesmo caracteres da língua japonesa.

Tabela 6 – Tabela com todos os Caracteres Disponíveis no Módulo LCD

Upper 4 Bits Lower 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)			0	1	P	`	P				—	タ	ミ	α	ρ
xxxx0001	(2)		!	1	A	Q	a	q			。	ア	チ	△	ã	q
xxxx0010	(3)		"	2	B	R	b	r			「	イ	ツ	×	β	θ
xxxx0011	(4)		#	3	C	S	c	s			」	ウ	テ	モ	ε	∞
xxxx0100	(5)		\$	4	D	T	d	t			、	エ	ト	ト	μ	Ω
xxxx0101	(6)		%	5	E	U	e	u			・	オ	ナ	ノ	σ	Ü
xxxx0110	(7)		&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)		'	7	G	W	g	w			ア	キ	ヌ	ラ	g	π
xxxx1000	(1)		(8	H	X	h	x			ィ	ク	ネ	リ	フ	Σ
xxxx1001	(2))	9	I	Y	i	y			ウ	ケ	ル	ル	‘	γ
xxxx1010	(3)		*	:	J	Z	j	z			エ	コ	ハ	レ	j	〒
xxxx1011	(4)		+	;	K	C	k	c			オ	サ	ヒ	ロ	*	π
xxxx1100	(5)		,	<	L	¥	l	l			カ	シ	フ	ワ	Φ	円
xxxx1101	(6)		—	=	M	J	m	}			ユ	ズ	ハ	ン	£	÷
xxxx1110	(7)		.	>	N	^	n	→			ヨ	セ	ホ	°	ñ	
xxxx1111	(8)		/	?	O	_	o	←			ッ	ソ	マ	°	ö	■

APÊNDICE II – CARACTERES DEFINIDOS PELO PROGRAMADOR

No módulo LCD utilizado cada caractere é construído a partir de uma matriz de pontos de 5 colunas por 8 linhas. Ele permite que o programador defina **até 8 caracteres**, por meio da construção dessa matriz para cada um deles. Durante o processo de inicialização do módulo LCD esses caracteres são definidos e posteriormente utilizados sempre que necessários.

Caracteres acentuados da língua portuguesa e outros símbolos não disponíveis na referida tabela podem ser criados, por exemplo: **á, Â, à, ã, Ã, ç, Ç, õ, Õ, ↓, ↑**. Caso se deseje mais de 8 caracteres diferentes em um dado instante pode-se, por software, alterar dinamicamente o conteúdo da memória desses caracteres, mantendo os necessários a cada instante.

Os novos caracteres devem ser gravados na memória **CG RAM** do módulo LCD, e podem ser utilizados da mesma forma que os caracteres da Tabela 6: o 1º. caractere definido possui valor **00**, o 2º., valor **01**, ..., o 8º., valor **07**. Contudo, essa memória é volátil, perdendo-se o seu conteúdo a cada desligamento. Os endereços-base para gravação dos novos caracteres são: **\$40, \$48, \$50, \$58, \$60, \$68, \$70 e \$78**. Cada um deles deve receber a configuração dos 5 pontos de uma das linhas do caractere, como mostra o exemplo da Figura 7.

Figura 7 – Exemplo de Construção de um Caractere

Set CG RAM Address														Data					
Adresse				Hex											Bit				Hex
					7	6	5	4	3	2	1	0							
0	1	0	0	0															

Nessa figura os pontos ativados significam valor 1, e se pode constar que se trata do primeiro caractere definido pelo programador (inicia-se no endereço **\$40**), e os valores das posições de memória de **\$40 a \$47** são, respectivamente: **\$04, \$04, \$04, \$04, \$15, \$0E, \$04 e \$00**.

Após o endereçamento da CG RAM, o cursor se desloca para a primeira posição da segunda linha.

APÊNDICE III – PROGRAMA EXEMPLO EM LINGUAGEM ASSEMBLY PARA TESTE DO MÓDULO LCD

Para que o Módulo LCD opere corretamente, a Placa Experimental deve estar convenientemente configurada:

- O conector **JP7** deve estar **fechado** com um *jumper* para que os sinais de **J7** sejam passados para **J8** (DB 25), que é o conector da porta paralela.
- **Não** será utilizado o sinal do *display* **R/W** nesta experiência. Assim, o respectivo fio será conectado ao **pino 1 (terra)** do conector **J2** da Placa Experimental.

O programa para controlar e operar o Módulo LCD deve incluir as seguintes funções:

- 1) Inicialização da porta paralela (configura as portas **A** e **B** do MC 68230).
- 2) Envio de uma sequência de bytes para gerar os sinais de controle descritos na Figura 5, no modo de enviar o **controle** do módulo LCD.
- 3) Envio de uma sequência de bytes para gerar os sinais de controle descritos na Figura 5, no modo de enviar um **dado** do módulo LCD.
- 4) Para o envio de controle e dado deve-se manipular os sinais **RS** e **E**, como explicado no item 4. O sinal **R/W** será sempre = **0** por hardware. Dessa forma, leituras não são permitidas.
- 5) Para modificar o sinal de habilitação **E** (usado na subrotina para enviar um byte de controle ou de dados para o módulo LCD) deve-se:

Para **E = 0** => **PACR = \$A8**

Para **E = 1** => **PACR = \$A0**

- 6) Para modificar o **RS** (usado na subrotina para enviar um byte de controle ou de dados ao módulo LCD) deve-se:

RS = 0 => **PBCR = \$A8**

RS = 1 => **PBCR = \$A0**

- 7) Criar uma subrotina de **espera de tempo**, pois deve haver um tempo mínimo entre alguns sinais segundo a carta de tempo para leitura e escrita de dados.
- 8) **Programa-Exemplo**: a seguir segue um programa-exemplo de configuração e utilização do Módulo LCD.

```

; Experiencia 06 - Exemplo: Teste do Módulo LCD
; Para executar na Placa Experimental
; Author: Carlos E. Cugnasca

; Este programa exibe uma mensagem em cada linha do Módulo LCD duas vezes e solicita ao
; usuário digitar 1 para repetir o teste, ou qualquer tecla para retornar ao Programa Tutor

        org     $1000

; Funções TRAP #14: número da função deve ser colocado em d7

TUTOR    equ     228      ; função que retorna ao Programa Tutor
INCHE    equ     247      ; funcao que pega um caractere do teclado
OUTCH    equ     248      ; funcao que envia um caractere para a tela
OUTPUT   equ     243      ; funcao que envia uma string para a tela
OUT1CR   equ     227      ; funcao que envia CR e LF ao terminal

; Endereços de Registradores do MC68230

PGCR     equ     $FE8001 ; registrador de controle geral
PACR     equ     $FE800D ; Porta A - registrador de controle
PBCR     equ     $FE800F ; Porta B - registrador de controle
PADDR    equ     $FE8005 ; Porta A - registrador de direção de dados
PBDDR    equ     $FE8007 ; Porta B - registrador de direção de dados
PADR     equ     $FE8011 ; Porta A - registrador de dados
PBDR     equ     $FE8013 ; Porta B - registrador de dados

; Códigos de configuração do módulo LCD

LCD2L5X7 equ     $38      ; LCD 2 linhas 5 X 7
LIMLCD   equ     $01      ; Limpa Display e Retorna o Cursor
HOME     equ     $02      ; Coloca Cursor na Posição Inicial (Home)
LCDOFF   equ     $08      ; Display Apagado
CESQ     equ     $10      ; Desloca somente o Cursor para a Esquerda
CMESQ    equ     $18      ; Desloca o Cursor e a Mensagem para a Esquerda
CDIR     equ     $14      ; Desloca somente o Cursor para a Direita
CMDIR    equ     $1C      ; Desloca o Cursor e a Mensagem para a Direita
LCDONSC  equ     $0C      ; Display Aceso sem Cursor
LCDONCF  equ     $0E      ; Display Aceso com Cursor Fixo
LCDONCF  equ     $0F      ; Display Aceso com Cursor Intermitente
C1L1P    equ     $80      ; Desloca o Cursor para a 1ª. linha, 1ª. posição
C2L1P    equ     $C0      ; Desloca o Cursor para a 2ª. linha, 1ª. posição
WDCESQ   equ     $04      ; Escreve deslocando o Cursor para a Esquerda
WDCDIR   equ     $06      ; Escreve deslocando o Cursor para a Direita
WDMESQ   equ     $07      ; Escreve deslocando a Mensagem para a Esquerda
WDMDIR   equ     $05      ; Escreve deslocando a Mensagem para a Direita

; Programa Principal

TLCD     move.l   #$8000,a7      ; inicia a pilha: última posição da RAM + 1
         bsr      INICPIA        ; Inicializa MC 68230
         bsr      DPINIT         ; Inicializa Modulo LCD

TLCD1    move.b   #C1L1P,d0      ; seleciona 1a. linha
         bsr      WRITEC        ; escreve caractere
         lea       (MEN01,pc),a0  ; (a0)<= end. inicial da mensagem
         bsr      PRINT          ; imprime mensagem

         move.b   #C2L1P,d0      ; seleciona 2a. linha
         bsr      WRITEC        ; escreve caractere
         lea       (MEN02,pc),a0  ; (a0)<= end. inicial da mensagem
         bsr      PRINT          ; imprime mensagem

         bsr      ESPERA2        ; espera tempo grande

         move.b   #LIMLCD,d0     ; apaga display
         bsr      WRITEC        ; escreve caractere

         bsr      ESPERA2        ; espera tempo grande

         move.b   #C1L1P,d0      ; seleciona 1a. linha
         bsr      WRITEC        ; escreve caractere
         lea       (MEN01,pc),a0  ; (a0)<= end. inicial da mensagem
         bsr      PRINT          ; imprime mensagem

```

```

    move.b    #C2L1P,d0      ; seleciona 2a. linha
    bsr       WRITEC
    lea       (MEN02,pc),a0   ; (a0)<= end. inicial da mensagem
    bsr       PRINT          ; imprime mensagem

    move.b    #OUT1CR,d7     ; envia CR e LF ao terminal
    trap      #14

    lea.l     MEN03,a5       ; a5 aponta para o inicio da mensagem
    lea.l     MEN04,a6       ; a6 aponta para o fim da mensagem +1
    move.b    #OUTPUT,d7     ; envia mensagem - repetir ou terminar o programa
    trap      #14

    move.b    #INCHE,d7      ; pega caractere em do.b
    trap      #14

    move.b    #OUTCH,d7      ; envia caractere recebido em d0.b para tela
    trap      #14

    cmpi.b    #'1',d0        ; digitou 1: repete teste
    beq       TLCD1

    move.b    #TUTOR,d7      ; termina o programa
    trap      #14           ; e retorna ao Programa Tutor

; ***** fim do programa *****

; Sub-rotina para inicializar a porta paralela (configura as portas A e B do MC 68230)
INICPIA move.b    #$30,PGCR   ; programa modo
        move.b    #$A0,PACR   ; programa submodo Porta A
        move.b    #$A0,PBCR   ; programa submodo Porta B
        move.b    #$FF,PADDR   ; programa direção da Porta A (saída)
        move.b    #$FF,PBDDR   ; programa direção da Porta B (saída)
        rts

; Sub-rotina para inicializar o Modulo LCD
; Códigos de programação: $38, $38, $38, $06, $0C, $01
DPINIT
    move.b    #LCD2L5X7,d0    ; LCD 2 linhas, caractere 5x7
    bsr       WRITEC          ; Envia programação 3 vezes
    move.b    #LCD2L5X7,d0    ; recomendação dos fabricantes
    bsr       WRITEC
    move.b    #LCD2L5X7,d0
    bsr       WRITEC
    move.b    #WDCDIR,d0      ; Escreve deslocando o Cursor para a Direita
    bsr       WRITEC
    move.b    #LCDONSC,d0     ; Display Aceso sem Cursor
    bsr       WRITEC
    move.b    #LIMLCD,d0      ; Limpa Display e Retorna o Cursor
    bsr       WRITEC
    rts

; Sub-rotina que envia sinais de controle ao Modulo LCD
; Entrada: d0.b = código de controle
WRITEC move.b    #$A8,PBCR    ; RS = 0
        move.b    #$A8,PACR    ; E = 0
        move.b    d0,PADR      ; Comando na via
        move.b    #$A0,PACR    ; E = 1
        move.b    #$A8,PACR    ; E = 0
        bsr       ESPERA       ; Espera exigida pelo sinal
        rts

; Sub-rotina que envia sinais de dados ao Modulo LCD
; Entrada: d0.b = dado em código ASCII
WRITED move.b    #$A0,PBCR    ; RS = 1
        move.b    #$A8,PACR    ; E = 0
        move.b    d0,PADR      ; Dado na via
        move.b    #$A0,PACR    ; E = 1
        move.b    #$A8,PACR    ; E = 0
        bsr       ESPERA       ; Espera exigida pelo sinal
        rts

```

```
; Sub-rotina para esperar um certo tempo para a geração dos sinais
; no Módulo LCD
; Registradores que utiliza: d1.w e flags

ESPERA  move.w    #20480,d1
ESP1    subq.w    #1,d1
        bne      ESP1
        rts

; Sub-rotina para esperar um tempo grande
; no Módulo LCD
; Registradores que utiliza: d1.w d2.l e flags
; chama n vezes ESPERA, d2.l = n

ESPERA2 move.l    #10,d2
ESP2     bsr      ESPERA
        subq.l    #1,d2
        bne      ESP2
        rts

; Sub-rotina imprime uma string no Módulo LCD
; Registradores que utiliza: d0.b, a0.l e flags
; Entrada: a0.l = endereço inicial da string

PRINT   move.b    (a0)+,d0      ; ao = endereço do 1o. caractere
        beq      PRL1          ; imprime ate encontrar 00
        bsr      WRITED
        bra      PRINT
PRL1     rts

; Área de mensagens

        org      $2000

MEN01   dc.b      "    PCS 3432    "
        dc.b      $00          ; fim de mensagem
MEN02   dc.b      "1o.Semestre 2018"
        dc.b      $00          ; fim de mensagem
MEN03   dc.b      'Digite 1 para repetir o teste ou outra tecla para sair: '
MEN04                                     ; endereço seguinte ao da string anterior
```

ANEXO I - MODELO DE RELATÓRIO E ITENS EXPERIMENTAIS

Esse anexo apresenta a parte experimental que deverá ser realizada por cada grupo, constituído por **dois alunos** (eventualmente um grupo possuirá três alunos, no caso de número ímpar de alunos na turma).

Apenas os nomes dos alunos presentes deverão constar do Relatório.

Diversas atividades previstas no **Modelo de Relatório** de cada experiência deverão ser realizadas **antes da aula**, por **todos** os componentes do grupo, que deverão se reunir e elaborar o **Planejamento da Experiência**, que é o Modelo de Relatório com os devidos itens preenchidos ou realizados (algoritmos, documentação, especificação e códigos de programas, etc). No caso de programas, arquivos específicos com **programas-fonte** deverão também ser trazidos.

O **Planejamento da Experiência** não precisa ser impresso, porém os arquivos correspondentes devem ser trazidos **por cada componente do grupo**. Cada um deverá saber explicar e justificar o que foi elaborado pelo grupo, **não sendo aceita a divisão de tarefas**, com cada um apenas conhecendo a parte que realizou. Essa explicação será exigida pelo professor.

Todos os arquivos deverão ser trazidos em **pen drives** (mais de uma cópia), além de cópias de segurança em espaços de armazenamento na nuvem (por exemplo, GDrive, Dropbox, OneNote, etc).

Grupos sem planejamento (por qualquer motivo) ou que não conseguirem explicá-lo ao professor inviabilizarão a realização da experiência, com prejuízo na respectiva nota.

Na página seguinte segue a cópia do **Modelo de Relatório**, cujo **arquivo .doc** se encontra no site da disciplina.

Cada grupo realizará a experiência em uma das bancadas do laboratório, numeradas de **0** a **7**. Esse número passará a ser o **número do grupo** até o final do semestre, e deverá ser colocado no cabeçalho do Relatório.

Os **nomes dos arquivos** necessariamente deverão seguir o padrão a seguir, bem como o **título dos emails**:

- nome do arquivo de cada Relatório:

pcs3432-Tx-E03.ext

onde

x é o número da Turma: 1 = segundas-feiras, 2 = terças-feiras e 3 = sextas-feiras

yy é o número da Experiência: 01, 02, ...

ext é a extensão do arquivo: doc, docx, pdf (preferido).

Caso arquivos adicionais sejam enviados (por exemplo, arquivos de programas), os respectivos nomes deverão ser:

pcs3432-Tx-E03-Azz.ext

onde

zz é o número do Anexo (01, 02, ...)

ext é a extensão do arquivo: dependerá do tipo de arquivo.

Os anexos necessariamente deverão ser referenciados no Relatório.

Por comodidade, todos os arquivos de um Relatório (incluindo os anexos) poderão ser compactados no padrão **zip**. Nesse caso, basta enviar um arquivo

pcs3432-Tx-E03.zip

que incluirá o Relatório e respectivos anexos.

PCS 3432 – LABORATÓRIO DE PROCESSADORES RELATÓRIO - EXPERIÊNCIA 06 - xx/xx/2018

Bancada b: 1º. Componente do Grupo
2º. Componente do Grupo
3º. Componente do Grupo (quando houver)

Professor:
Turma x

1) Seguir o item 6 - PARTE PRÁTICA da apostila:

- a) Para todas as turmas, incluir os **itens 6.1, 6.2 e 6.3, elaborando os diagramas.**
- b) Incluir a parte prevista para a sua turma apresentada no **item 6.4, elaborando o diagrama e o código do programa.**

Encaminhar, juntamente com este relatório, os arquivos relativos a todos os programas modificados e elaborados.

2) COMENTÁRIOS E CONCLUSÕES

Incluir comentários relevantes sobre a experiência, bem como problemas enfrentados e respectivas soluções, e itens não realizados da experiência, justificando.

**Colocar todos os documentos e arquivos em um único arquivo compactado:
pcs3432-Tx-E06.zip.**