

EXPERIÊNCIA 3:

FAMILIARIZAÇÃO COM PERIFÉRICOS DE ENTRADA E SAÍDA: INTERFACE COM DISPLAY

Autores: Prof. Dr. Carlos Eduardo Cugnasca e Prof. Dr. Ricardo Costa Zerbini

*Adaptação para o display alfanumérico: Daniel Ziviani Cassettari, Luiz Paulo Rocha Yanai,
Vitor Pereira Spachi, Rodrigo Gioia Guimarães, Erika Midori Kato, Paula Franchi Cruz*

Agosto – 2004

Revisão: Profs. Dr. André Riyuiti Hirakawa e Dr. Carlos Eduardo Cugnasca

Março – 2009

1. OBJETIVO

Nesta experiência será apresentado e estudado o funcionamento do dispositivo de saída denominado “display”. A título de exemplo de aplicação, será implementada e testada a interface entre um display de cristal líquido da Alfacom e a Placa Experimental MC68000.

2. DISPLAYS

2.1. Principais Tecnologias

Um dos displays mais antigos é o de tecnologia desenvolvida pela “Borroughs Corporation” denominado display “NIXIE”. Este elemento era formado por um tubo com 10 números sobrepostos em planos diferentes, de maneira que quando um deles se 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 um certo desconforto na leitura de um número com vários dígitos diferentes. Outro problema era que o acendimento dos números exigia altas tensões de operação, e o chaveamento rápido de altas tensões, necessário na implementação da técnica de multiplexação (essencial na redução de tamanho e custos dos displays), sempre foi problemático. Mais tarde surgiram os displays de sete segmentos, que tiveram bastante aceitação e custos bem menores, estando disponíveis até hoje em grande variedade de formatos, cores e tipos.

As principais tecnologias de displays de sete segmentos são:

- *Displays incandescentes*: operando com base em filamentos incandescentes, são disponíveis em várias cores e tamanhos, são os que podem oferecer maior intensidade de luz.
- *Displays de neon ou display de 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.
- *Displays fluorescentes*: são displays azuis ou verdes que exigem tensões um pouco menores que as exigidas pelo tipo anterior, simplificando a multiplexação.

- *Display de LEDs ("Light Emitting Diode")*: é uma tecnologia do início da era do semicondutor, sendo assim um dispositivo de estado sólido. Aproveita-se aqui o 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 e o verde.
- *Display de cristal líquido (LCD)*: Nestes dispositivos algumas impurezas polares 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 polares), 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 o transmissivo, que apresenta uma iluminação na sua parte traseira. Estes dispositivos, por exigirem apenas tensão para mover as impurezas, consomem muito 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 maior sensibilidade à radiação ultra-violeta encontrada em ambientes abertos (origem solar).

3. DISPLAY ALFANUMÉRICO DE CRISTAL LÍQUIDO

Nos últimos anos, os módulos de display alfanumérico têm se tornado extremamente populares, devido à grande escala de utilização que vem ocorrendo nos diversos produtos, equipamentos e instrumentos. Dentre as aplicações do display podemos citar calculadoras, terminais de consulta de bancos, painéis informativos, painéis de equipamentos diversos, etc. Essa popularização do display se deve às vantagens oferecidas, que são: conexão facilitada com equipamentos diversos, fácil operação, baixo consumo e baixo custo.

Atualmente, existe muitos fabricantes produzindo LCDs, que funcionalmente mantêm as mesmas características. As diferenças são quanto à conexão, número de linhas e colunas, dimensões físicas e consumo. Assim, seqüências de comandos criados para um display podem ser aplicadas em outros sem necessidade de maiores modificações.

Um display normal (padrão de mercado) possui dois registradores: um registrador de comandos e outro de dados.

Um display recebe comandos no seu registrador de comandos para:

- a sua inicialização;
- mover o cursor para a direita;
- deslocar todos os caracteres para a esquerda;
- limpar a tela do display;
- etc.

4. CÓDIGOS DE PROGRAMAÇÃO DO DISPLAY

O display a ser utilizado aceita códigos para a sua programação de 8 bits, que são direcionados para o registrador de instrução quando os sinais $RS = 0$ e $R/\overline{W} = 0$ [1][6]. Durante o desenvolvimento de programas, 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 display em formato hexadecimal são listados na Tabela 1. Um exemplo de montagem de caracteres a partir da matriz de pontos do display é apresentada na Figura 1 [1][6].

Tabela 1 – Principais Códigos de Programação do Display.

DESCRIÇÃO DO COMANDO	CÓDIGO DE INSTRUÇÃO (HEX)	DESCRIÇÃO DO COMANDO	CÓDIGO DE INSTRUÇÃO (HEX)
<i>Controle de Display Habilitado</i>	0C	<i>Posição de Entrada de Dado</i>	
<i>Limpar (Conteúdo retido na memória)</i>	0A, 08	Esquerda	04
<i>Limpar Display e Cursor</i>	01	Direita	06
<i>Home Cursor</i>	02	<i>Deslocamento de Cursor com entrada de dado</i>	
<i>Cursor</i>		Esquerda	07
<i>ON</i>	0E	Direita	05
<i>OFF</i>	0C	<i>Deslocamento de Cursor sem entrada de dado</i>	
<i>Piscar</i>	0D	Esquerda	18
<i>Deslocar a Esquerda</i>	10	Direita	1C
<i>Deslocar a Direita</i>	14	<i>Endereçamento dos dados do Display</i>	
<i>Home</i>	02	1ª Linha	80
		2ª Linha	C0

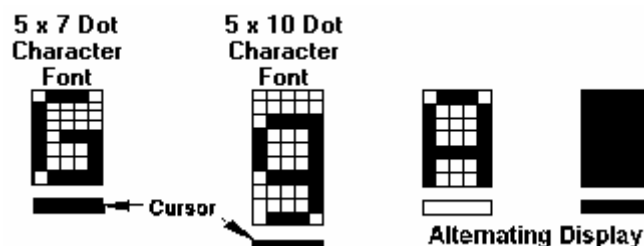


Figura 1 – Montagem dos Caracteres do Display.

Um display recebe dados que são os caracteres alfanuméricos inseridos no display. O ciclo de envio de comando e dado para o display é mostrado na seguinte Figura 2.

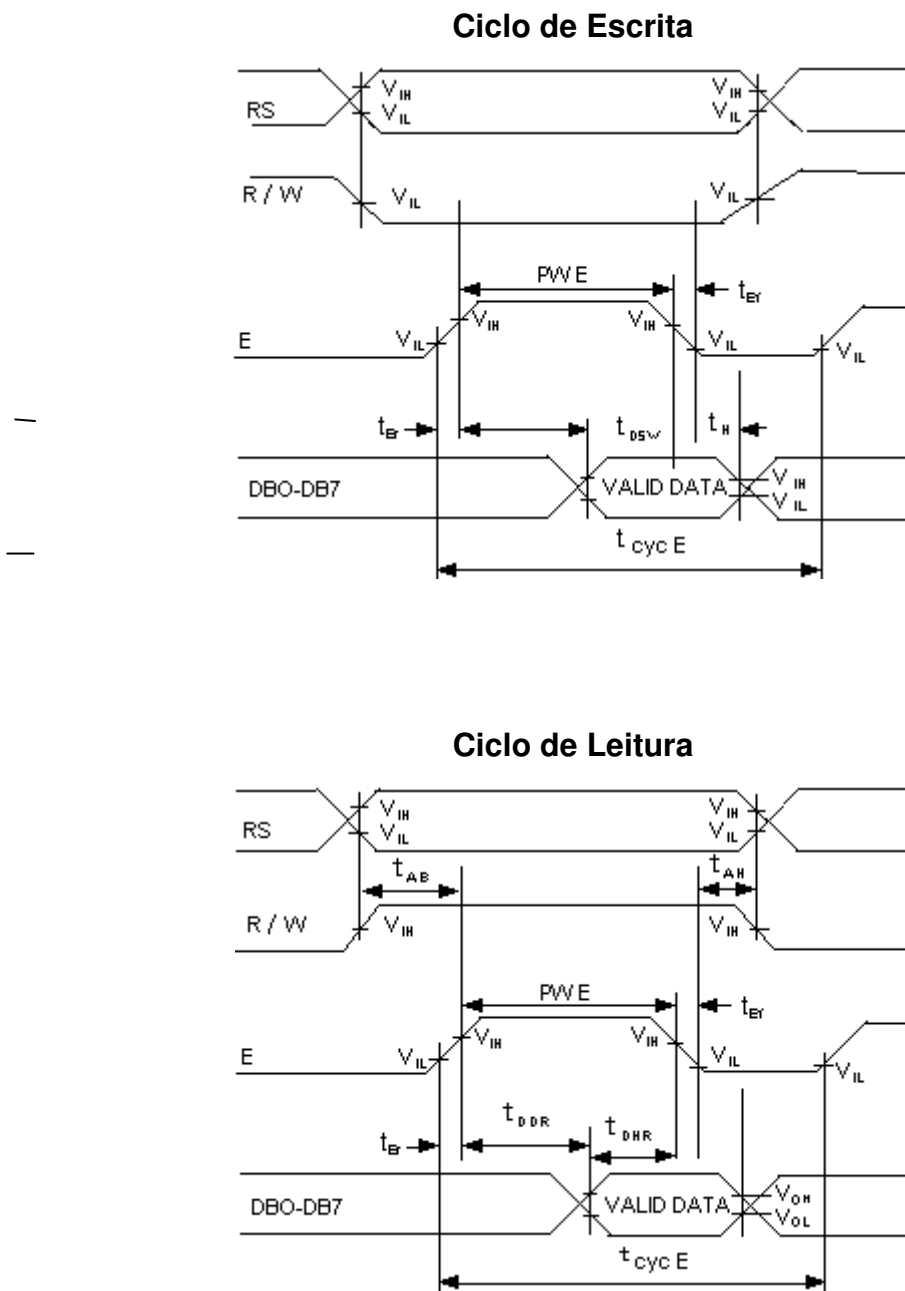


Figura 2 – Ciclos de Escrita e de Leitura do Display

Na Figura 2, as seguintes condições devem ser satisfeitas para a borda de descida do sinal "E", indicando o instante da escrita:

- $R/\overline{W}=0$ (indicando write);
- dados estáveis na via de dados;

- RS=0 indica um **comando** para o display (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 display recomenda-se a seguinte seqüência:

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

Para se escrever um byte de dados no display recomenda-se a seguinte seqüência:

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

Entre cada escrita na porta paralela, deve-se colocar um **atraso** da ordem de milissegundos.

4.1 Inicialização do display

Toda vez que o display é alimentado, um processo de inicialização deve ser executado. Este procedimento consiste no envio de um grupo de instruções para configurar o display no modo de execução normal. Para inicializá-lo pode-se enviar a seguinte seqüência de bytes de controle típica:

0x38, 0x38, 0x38, 0x06, 0x0C, 0x01.

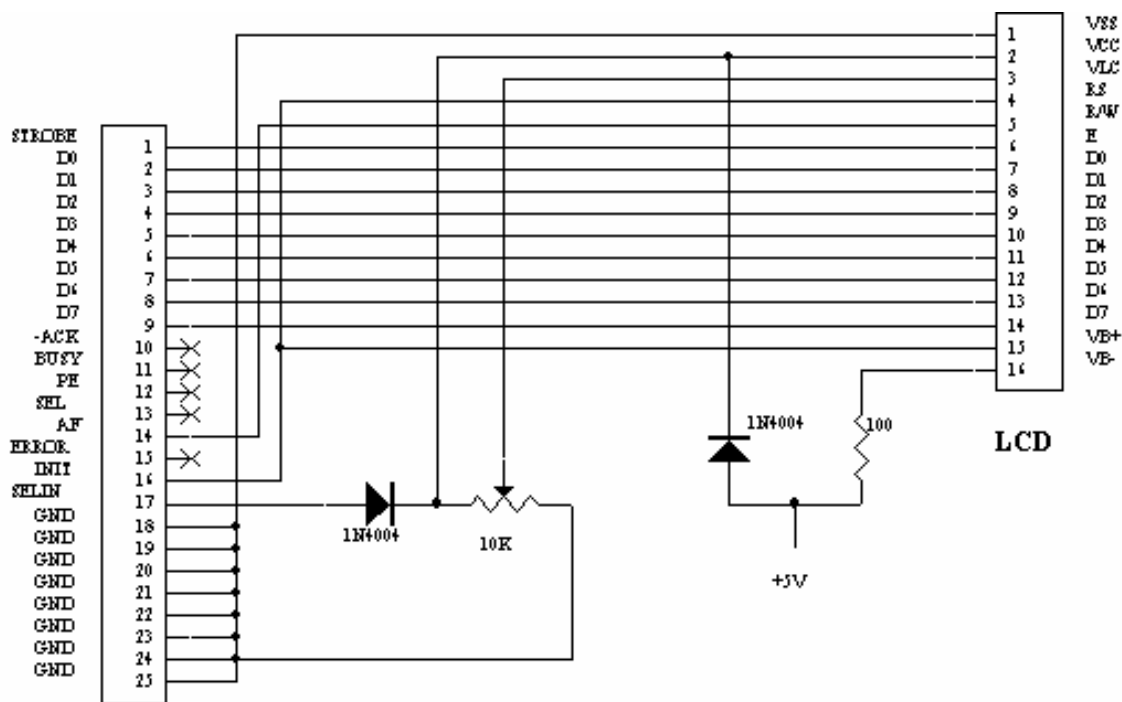
Todo o processo de inicialização é realizado com RS e $\overline{R/\overline{W}}$ ambos em nível 0.

O código hexadecimal inicial “38” é enviado mais de uma vez para garantir que o display assuma o modo de operação de tamanho 8 bits. O controle “0x01” limpa o display.

Uma vez que o display esteja inicializado, pode-se escrever um dado, ex: enviando o byte 0x41 o display apresenta o caractere "A" (0x41 é o seu código ASCII).

5. DISPLAY NA PORTA PARALELA DO MC68000

O hardware necessário à experiência (MC68000, interface paralela e display) será fornecido já montado, pronto para ser utilizado. O esquema elétrico da ligação do display na interface paralela é mostrado na Figura 3.



PRINTER PORT

Figura 3 – Ligação do Display a uma Porta Paralela

A conexão dos principais sinais é apresentada na Tabela 2.

Tabela 2 – Conexão dos Principais Sinais do Display.

J7	sinal	pino na porta paralela	conexão com o display
17	data strobe	1	E
	auto line feed	14	$\overline{R/W}$
	initialize printer	16	RS

6. PARTE EXPERIMENTAL

A parte experimental consiste em interligar um display alfanumérico ao microprocessador da Placa Experimental. Esta ligação será feita por meio da porta paralela do periférico MC68230, conforme mostra a Figura 3.

O programa de inicialização, controle e apresentação de dados no display, a ser elaborado em Linguagem *Assembly*, deverá obedecer à seguinte especificação:

- a) O display deverá exibir o conteúdo de oito posições consecutivas de memória, “PCS-2497”, previamente carregadas com valores compatíveis com o padrão da tabela do Apêndice II.
- b) Implementar um programa que apresenta a seqüência de caracteres “PCS-2497” da memória RAM de forma circulante no display, em uma de suas linhas.
- c) Modificar o item “b” de modo que o vetor de caracteres seja mostrado de forma circulante nas duas linhas (percorrer toda a primeira linha e continuar na segunda linha e depois voltar para a primeira linha).
- d) Modificar o item “b” de modo que o vetor de caracteres percorra a primeira linha, deslocando-se para a direita e, ao chegar ao final voltar, deslocando-se para a esquerda.

7. PLANEJAMENTO

Elaborar o planejamento conforme modelo e trazer impresso para a aula. Estruture o programa a ser elaborado de acordo com o sugerido no Apêndice I

8. BIBLIOGRAFIA

- [1] ALFACOM. Módulos Multi-Matrix - Manual de Utilização.
- [2] AZTEC C68k/ROM - Cross Development System - Version 3.4, November 1987, Manx Software Systems, Inc.
- [3] CUGNASCA, C.E. & ZERBINI, R.C. Experiência nº 3 - Familiarização com Periféricos de Entrada e Saída: Interface com Display. EPUSP, 1989.
- [4] INTEL. Especificações Técnicas do Componente 8279.
- [5] MOTOROLA. MC68230 - Peripheral Interface and Timer (PI/T). Motorola Inc.
- [6] SHELLY, INC. Site da empresa: <http://www.shellyinc.com/products/DotMatrix.htm>.

APÊNDICE I – ESTRUTURA BÁSICA DO PROGRAMA

/* ENDEREÇOS DE REGISTRADORES DO MC68230

```
/* PADR - 0xFE8011 */
/* PACR - 0xFE800D */
/* PBCR - 0xFE800F */
/* PADDR - 0xFE8005 */
/* PBDDR - 0xFE8007 */
```

/*

Hardware:

- 1) O conector JP7 deve estar fechado com um jumper para que os sinais de J7 sejam passados para J8 = conector da porta paralela.
- 2) NÃO SERÁ UTILIZADO O SINAL DE R/W/ NESTA EXPERIÊNCIA. ASSIM, ELE SERÁ CONECTADO AO PINO 21 (TERRA) DO CONECTOR J7 DA PLACA.

Software de interface com o Hardware

// 1) Criar uma rotina para inicializar a porta paralela (configura as portas A e B do MC68230)

INICPIA:

```
MOVE.B #0x30,0xFE8001
MOVE.B #0xA0,0xFE800D
MOVE.B #0xA0,0xFE800F
MOVE.B #0xFF,0xFE8005
MOVE.B #0xFF,0xFE8007
RTS
```

// 2) CRIAR UMA ROTINA QUE ENVIA UMA SEQUENCIA DE BYTES PARA GERAR OS SINAIS DE CONTROLE DESCRITOS NA FIGURA 2, NO MODO ENVIAR DADO. O BYTE DE DADO A SER ENVIADO DEVE ESTAR GUARDADO EM UM REGISTRADOR OU EM UMA POSIÇÃO DA MEMORIA.

DADO:

// 2) CRIAR UMA ROTINA QUE ENVIA UMA SEQUENCIA BYTES PARA GERAR OS SINAIS DE CONTROLE DESCRITOS NA FIGURA 2 NO MODO ENVIAR CONTROLE. O BYTE DE CONTROLE A SER ENVIADO DEVE ESTAR GUARDADO EM UM REGISTRADOR OU EM UMA POSIÇÃO DA MEMORIA.

CONTROLE:

Para fazer as rotinas DADO e CONTROLE, deve-se mexer nos sinais RS e E, como explicado no item 4. R/W/ = 0 por hardware.

PARA ENVIAR DADOS PARA A VIA DE DADOS DEVE-SE UTILIZAR:

```
MOVE.B dado,PADR
```

PARA MODIFICAR O ENABLE (usado na rotina para enviar um byte de controle ou de dados para o display)

E = 0 -> PACR = 0xA8

E = 1 -> PACR = 0xA0

PARA MODIFICAR O RS (usado na rotina para enviar um byte de controle ou de dados para o display)

RS = 0 -> PBCR = 0xA8

RS = 1 -> PBCR = 0xA0

```
/* CRIAR UMA ROTINA DE INICIALIZACAO DO DISPLAY, QUE MANDE OS COMANDOS DE CONTROLE INDICADOS NO ITEM 4. */
```

```
INIC:
```

```
*/
```

```
.TEXT
```

```
.GLOBL MAIN
```

```
.MAIN:
```

```
/* 1. inicializar a porta paralela */
```

```
JSR INICPIA
```

```
/* 2. APOS A PORTA PARALELA TER SIDO CONFIGURADA, A ROTINA DE INICIALIZACAO DO DISPLAY É CHAMADA – ver item 4.*/
```

```
JSR INIC
```

```
/* A PARTIR DAQUI, PODE-SE COMECAR A ENVIAR DADOS OU CONTROLE PARA O DISPLAY. EXEMPLO:
```

```
MOVE.B #0x41,local
```

```
JSR DADO ou
```

```
JSR CONTR
```

```
COM ESTE COMANDO, A ROTINA QUE ENVIA DADOS DEVE BUSCAR O BYTE #0x41 NO LOCAL QUE FOI DEFINIDO E ENVIA-LO PARA O DISPLAY, SEGUNDO A SEQUENCIA DEFINIDA NA APOSTILA. */
```

```
/* ESTA ROTINA DE LOOP SERVE PARA FINALIZAR O PROGRAMA, APÓS TEREM SIDO ENVIADOS TODOS OS DADOS/CONTROLE. */
```

```
LOOP: BRA LOOP
```

```
/* CRIAR UMA ROTINA DE DELAY
```

```
ESSA ROTINA É IMPORTANTE, POIS DEVE HAVER UM TEMPO MÍNIMO ENTRE ALGUNS SINAIS SEGUNDO A CARTA DE TEMPO PARA LEITURA E ESCRITA DE DADOS.*/
```

```
DELAY:
```

APÊNDICE II – CARACTERÍSTICAS DO DISPLAY

O display utilizado na experiência tem 2 linhas de 20 caracteres cada. A sequência de inicialização que está na apostila limpa o display e desloca o cursor para o início da 1ª. linha. Cada caractere pode ser endereçado individualmente. A tabela a seguir mostra o endereço de cada um deles.

	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

Para endereçá-los é preciso enviar um byte de controle contendo o valor do caractere. Por exemplo, para se endereçar o 7º. caractere da 1ª linha, envia-se o byte 0x86 como controle. Após receber este caractere de controle, o registrador de endereço do display é incrementado automaticamente para receber o próximo caractere (após receber o caractere, portanto, o registrador estará com o valor 0x87 nesse exemplo).

Vários caracteres podem ser enviados para serem exibidos no display. Os mais utilizados estão na tabela a seguir:

	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 0x48 como dado. A tabela completa está no manual do display Alfacom [1].