

EA201 - EXERCÍCIO PRÁTICO 3

Vinícius Esperança Mantovani, 247395

Introdução:

Neste exercício, foi observado o comportamento dos sinais enviados por uma MCU a um display LCD, por meio de interface SPI. Para tanto, foi analisado o conjunto de sinais de cada um dos pinos, enumerados no enunciado do presente exercício. Além disso, foi analisado o datasheet do display de 84x48 pontos, como forma de entender os sinais e reconhecer seu papel na configuração do display usado.

Item 1:

Analisando o datasheet do display em suas páginas 20 e 21, nota-se que o mínimo tempo em que o sinal reset deve estar em estado baixo é de 100 ns, como se vê no último elemento da tabela da imagem abaixo:

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
f _{OSC}	oscillator frequency		20	34	65	kHz
f _{clk(ext)}	external clock frequency		10	32	100	kHz
f _{frame}	frame frequency	f _{OSC} or f _{clk(ext)} = 32 kHz; note 1	–	67	–	Hz
t _{VHRL}	V _{DD} to $\overline{\text{RES}}$ LOW	Fig.16	0 ⁽²⁾	–	30	ms
t _{WL(RES)}	$\overline{\text{RES}}$ LOW pulse width	Fig.16	100	–	–	ns

Imagem 1

Conforme se pode ver na imagem a seguir, o tempo aproximado em que RST fica em low é de 101 ms:

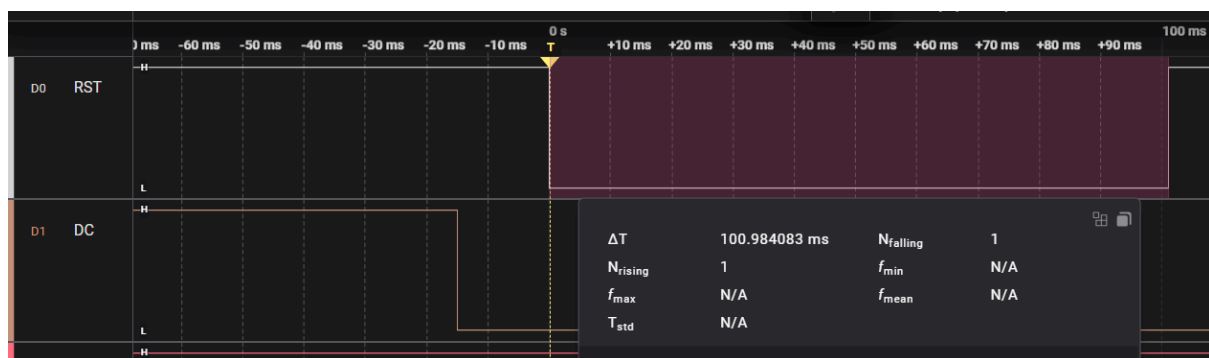


Imagem 2

Portanto, como 101 ms \gg 100 ns, temos que o pulso enviado pela MCU é certamente suficiente.

Item 2:

Após a subida do sinal *RST*, a primeira sequência de dados enviados é a que se apresenta na imagem abaixo (sinal *DI*):

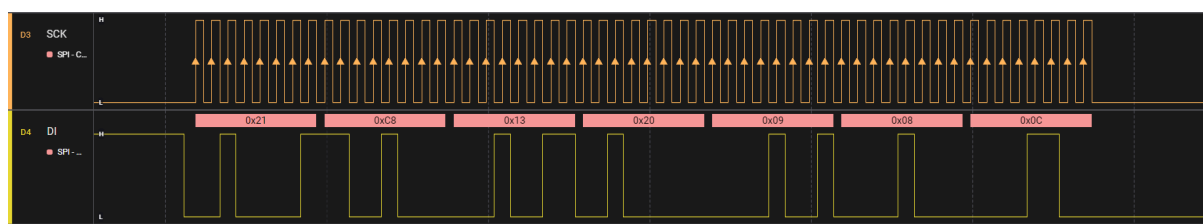


Imagem 3

Observando os dados acima e a tabela 1 da página 14 do datasheet, que se apresenta abaixo, pode-se montar a tabela desejada para este item, a qual é também apresentada mais adiante (usa-se ainda a tabela 2 da mesma página para entender o que os bits *PD*, *H*, *V*... ocasionam no comportamento do display de acordo com o nível lógico que estão assumindo no momento).

Table 1 Instruction set

INSTRUCTION	D/C	COMMAND BYTE								DESCRIPTION
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
(H = 0 or 1)										
NOP	0	0	0	0	0	0	0	0	0	no operation
Function set	0	0	0	1	0	0	PD	V	H	power down control; entry mode; extended instruction set control (H)
Write data	1	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	writes data to display RAM
(H = 0)										
Reserved	0	0	0	0	0	0	1	X	X	do not use
Display control	0	0	0	0	0	1	D	0	E	sets display configuration
Reserved	0	0	0	0	1	X	X	X	X	do not use
Set Y address of RAM	0	0	1	0	0	0	Y ₂	Y ₁	Y ₀	sets Y-address of RAM; 0 ≤ Y ≤ 5
Set X address of RAM	0	1	X ₆	X ₅	X ₄	X ₃	X ₂	X ₁	X ₀	sets X-address part of RAM; 0 ≤ X ≤ 83
(H = 1)										
Reserved	0	0	0	0	0	0	0	0	1	do not use
	0	0	0	0	0	0	0	1	X	do not use
Temperature control	0	0	0	0	0	0	1	TC ₁	TC ₀	set Temperature Coefficient (TC _x)
Reserved	0	0	0	0	0	1	X	X	X	do not use
Bias system	0	0	0	0	1	0	BS ₂	BS ₁	BS ₀	set Bias System (BS _x)
Reserved	0	0	1	X	X	X	X	X	X	do not use
Set V _{OP}	0	1	V _{OP6}	V _{OP5}	V _{OP4}	V _{OP3}	V _{OP2}	V _{OP1}	V _{OP0}	write V _{OP} to register

Imagem 4

Table 2 Explanations of symbols in Table 1

BIT	0	1
PD	chip is active	chip is in Power-down mode
V	horizontal addressing	vertical addressing
H	use basic instruction set	use extended instruction set
D and E		
00	display blank	
10	normal mode	
01	all display segments on	
11	inverse video mode	
TC ₁ and TC ₀		
00	V _{LCD} temperature coefficient 0	
01	V _{LCD} temperature coefficient 1	
10	V _{LCD} temperature coefficient 2	
11	V _{LCD} temperature coefficient 3	

Imagem 5

Sob comparação entre os valores passados por meio do sinal *DI* e os valores apresentados na tabela, encontram-se os comandos referentes aos conjuntos de bits transmitidos, do modo como se observa a seguir:

Valores dos conjuntos de 8 bits (amostrados na imagem 3) e seus respectivos comandos associados:

1. 0x21 = 0b00100001 → *Function set*, dita os valores $PD = 0$ (chip ativo), $V = 0$ (endereço horizontal) e $H = 1$ (uso do conjunto expandido de instruções);
2. 0xC8 = 0b11001000 → *Set V_{OP}*, determina os valores $V_{OP6} = 1$, $V_{OP5} = 0$, $V_{OP4} = 0$, $V_{OP3} = 1$, $V_{OP2} = 0$, $V_{OP1} = 0$, $V_{OP0} = 0$. Ditando a voltagem de operação do display;
3. 0x13 = 0b00010011 → *Bias System*, determina os valores $BS_2 = 0$, $BS_1 = 1$, $BS_0 = 1$, determinando o Bias System;
4. 0x20 = 0b00100000 → *Function set*, dita os valores $PD = 0$, $V = 0$, $H = 0$ (uso do conjunto básico de instruções);
5. 0x09 = 0b00001001 → *Display Control*, dita os valores $D = 0$ e $E = 1$, determinando que todos os segmentos do display sejam ligados;
6. 0x08 = 0b00001000 → *Display Control*, dita os valores $D = 0$ e $E = 0$, determinando que o display fique vazio (apagado por completo);
7. 0x0C = 0b00001100 → *Display Control*, dita os valores $D = 1$ e $E = 0$, determinando que o display atue em modo normal (não invertido);

Após esses 7 conjuntos que são enviados no mesmo intervalo de *CE* em nível lógico alto, no próximo intervalo de *CE* em alto, tem-se mais três comandos conforme se segue:

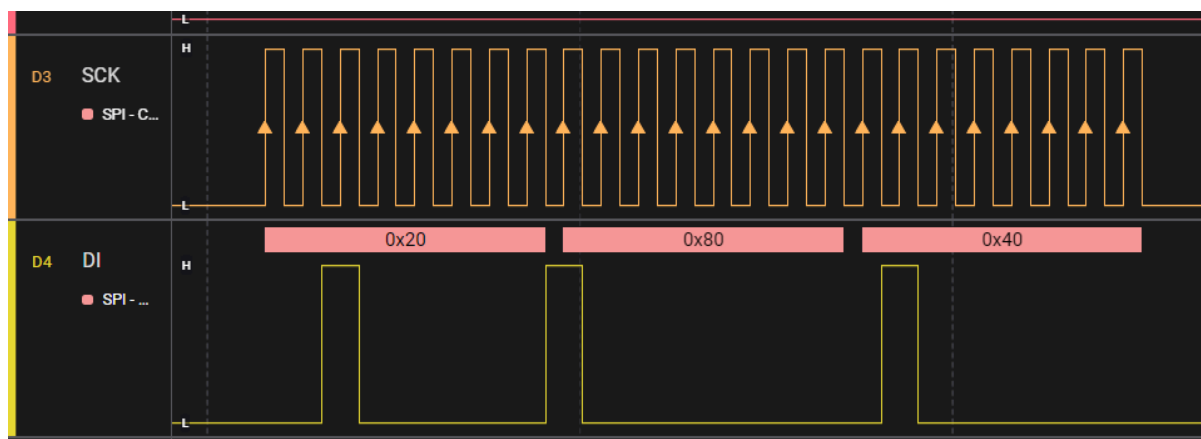


Imagem 6

1. $0x20 = 0b00100000 \rightarrow$ *Function set*, dita os valores $PD = 0$, $V = 0$, $H = 0$ (uso do conjunto básico de instruções);
2. $0x80 = 0b10000000 \rightarrow$ *Set X Address of RAM*, determina a posição em 0 no eixo X, definindo cada um dos bits referentes à posição X na RAM (apresentados na imagem 4);
3. $0x40 = 0b01000000 \rightarrow$ *Set Y Address of RAM*, determina a posição em 0 no eixo Y, definindo cada um dos bits referentes à posição Y na RAM (apresentados na imagem 4).

Levando-se em conta o que se apresenta acima, montou-se a tabela abaixo, com os comandos nos moldes da tabela feita no exercício anterior, com exceção da coluna que tratava da resposta do módulo, pois não temos resposta do display para apresentar:

Número	Comando MCU	Significado
1	Function set	Dita os valores $PD = 0$ (chip ativo), $V = 0$ (endereço horizontal) e $H = 1$ (uso do conjunto expandido de instruções)
2	Set Vop	Determina os valores $V_{OP6} = 1$, $V_{OP5} = 0$, $V_{OP4} = 0$, $V_{OP3} = 1$, $V_{OP2} = 0$, $V_{OP1} = 0$, $V_{OP0} = 0$. Ditando a voltagem de operação do display
3	Bias System	Determina os valores $BS2 = 0$, $BS1 = 1$, $BS0 = 1$, determinando o Bias System
4	Function set	Dita os valores $PD = 0$, $V = 0$, $H = 0$ (uso do conjunto básico de instruções)
5	Display Control	Dita os valores $D = 0$ e $E = 1$, Determinando que todos os

		segmentos do display sejam ligados
6	Display Control	Dita os valores $D = 0$ e $E = 0$, determinando que o display fique vazio (não escrito)
7	Display Control	Dita os valores $D = 1$ e $E = 0$, determinando que o display atue em modo normal (não invertido)
8	Function set	Dita os valores $PD = 0$, $V = 0$, $H = 0$ (uso do conjunto básico de instruções)
9	Set X Address of RAM	Determina a posição em 0 no eixo X, definindo cada um dos bits referentes à posição X na RAM (apresentados na imagem 4)
10	Set Y Address of RAM	Determina a posição em 0 no eixo Y, definindo cada um dos bits referentes à posição Y na RAM (apresentados na imagem 4)

Por fim, sintetizando o que se organiza acima tanto em forma de tabela como de texto, pode-se explicar esses comandos como uma sequência procedural responsável por definir o contexto de “inicialização” da operação do display. Tal “inicialização” abrange desde o controle de voltagem do aparelho até a imposição dos valores nulos à posição nos eixos X e Y do display, de maneira a passar pela determinação do modo de operação do dispositivo (modo normal) e pelo esvaziamento de qualquer dado que pudesse estar escrito na tela. Nesse sentido, os comandos acima apresentados são os responsáveis por permitir a operação correta do display dentro das condições em que a desejamos.

Item 3:

Conforme se nota ao observar a imagem 4, a única instrução que ocorre sob *DC* é a de escrita (“Write data”), que escreve dados na RAM do display, mostrando-os na tela. Partindo disso e da análise da imagem que se segue, pode-se concluir o que significam as rajadas de dados de bytes em 0 para o display.

Table 6 Programming example

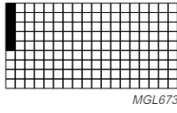
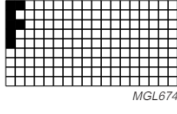

STEP	SERIAL BUS BYTE									DISPLAY	OPERATION
	D/C	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
1	start										\overline{SCE} is going LOW
2	0	0	0	1	0	0	0	0	1		function set PD = 0 and V = 0, select extended instruction set (H = 1 mode)
3	0	1	0	0	1	0	0	0	0		set V_{OP} ; V_{OP} is set to a $+16 \times b$ [V]
4	0	0	0	1	0	0	0	0	0		function set PD = 0 and V = 0, select normal instruction set (H = 0 mode)
5	0	0	0	0	0	1	1	0	0		display control set normal mode (D = 1 and E = 0)
6	1	0	0	0	1	1	1	1	1	 MGL673	data write Y and X are initialized to 0 by default, so they are not set here
7	1	0	0	0	0	0	1	0	1	 MGL674	data write
8	1	0	0	0	0	0	1	1	1	 MGL675	data write

Imagem 7

Conforme se pode observar, na imagem 7, tem-se uma breve inicialização do display e, em sequência, a escrita de três colunas que compõem uma letra “P”. Além disso, se bem repararmos, o modo de endereçamento do dispositivo neste exemplo também está “setado” como horizontal. Assim, faz-se notável que cada uma das escritas, no modo de endereçamento presente, é responsável por preencher uma coluna e “empurrar” o “cursor” à direita (incrementar o valor da posição no X no endereço referente a ele na RAM). Por fim, cabe ressaltar que ao final de uma linha, o “cursor” seria empurrado não só para o lado direito, mas também para baixo (incremento do valor da posição no Y no endereço referente a ele na RAM), de maneira a permitir a composição da próxima linha.

Portanto, na sequência do que se discutia, pode-se afirmar que a grande sequência de dados em zero enviada ao display à qual se refere (seis conjuntos de 84 subconjuntos de 8 bits em zero) no enunciado é responsável por zerar as seis linhas de altura de 8 bits (8 pixels) do display. Dessa forma, tem-se o display garantidamente “limpo” para se escrever o que é desejado.

Item 4:

Seguindo o mesmo procedimento aplicado no item 2 em análise da imagem 8 abaixo, tem-se o que segue essa imagem:

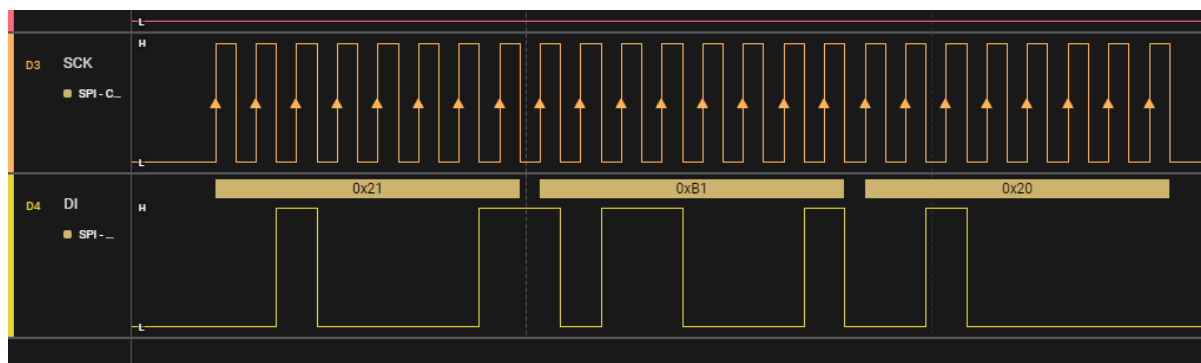


Imagem 8

Valores dos conjuntos de 8 bits (amostrados na imagem 3) e seus respectivos comandos associados:

1. $0x21 = 0b00100001 \rightarrow$ *Function set*, dita os valores $PD = 0$ (chip ativo), $V = 0$ (endereço horizontal) e $H = 1$ (uso do conjunto expandido de instruções);
2. $0xB1 = 0b10110001 \rightarrow$ *Set V_{OP}* , determina os valores $V_{OP6} = 0$, $V_{OP5} = 1$, $V_{OP4} = 1$, $V_{OP3} = 0$, $V_{OP2} = 0$, $V_{OP1} = 0$, $V_{OP0} = 1$. Alterando a voltagem de operação do display;
3. $0x20 = 0b00100000 \rightarrow$ *Function set*, dita os valores $PD = 0$, $V = 0$, $H = 0$ (uso do conjunto básico de instruções);

Disso, tira-se a tabela:

Número	Comando MCU	Significado
1	Function set	Dita os valores $PD = 0$ (chip ativo), $V = 0$ (endereço horizontal) e $H = 1$ (uso do conjunto expandido de instruções)
2	Set V_{OP}	Determina os valores $V_{OP6} = 0$, $V_{OP5} = 1$, $V_{OP4} = 1$, $V_{OP3} = 0$, $V_{OP2} = 0$, $V_{OP1} = 0$, $V_{OP0} = 1$. Alterando a voltagem de operação do display
3	Function set	Dita os valores $PD = 0$, $V = 0$, $H = 0$ (uso do conjunto básico de instruções)

Do que se exhibe acima, é possível concluir que esses últimos comandos são responsáveis por alterar a voltagem de operação do display.

Item 5:

A sequência de Bytes escritos no display para composição do caractere “0” pode ser vista na imagem abaixo:

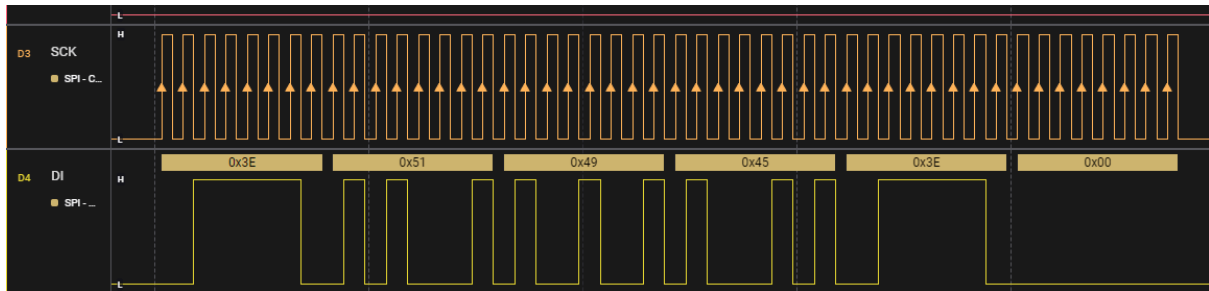


Imagem 9

Considerando o fato de que as colunas são escritas de baixo para cima, ou seja, os bits mais significativos estão associados aos pixels inferiores da linha de 8 bits do display, temos da imagem acima:

1. $0x3E = 0b00111110 \rightarrow$



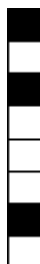
2. $0x51 = 0b01010001 \rightarrow$

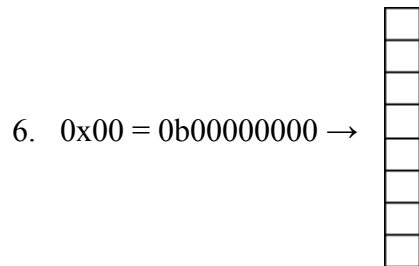
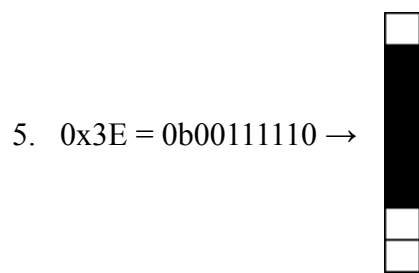


3. $0x49 = 0b01001001 \rightarrow$

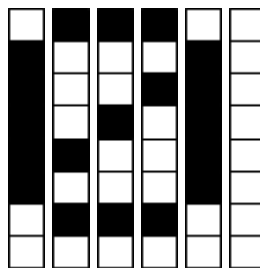


4. $0x45 = 0b01000101 \rightarrow$





Observando o acima exposto, por fim, temos o zero no começo da primeira linha do display conforme o exibido abaixo:



Item 6:

Neste último item, a imagem utilizada foi a disposta abaixo:

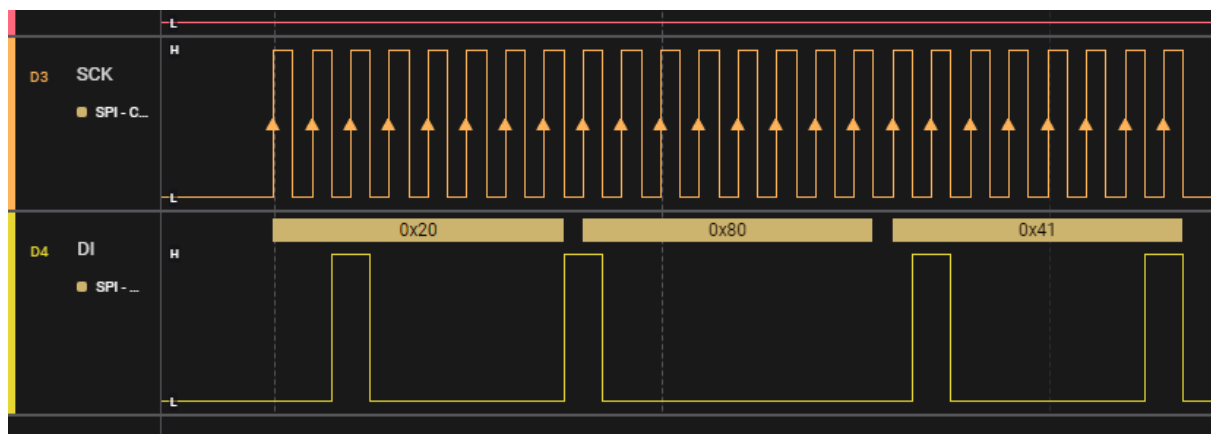


Imagem 10

Dessa imagem, conseguimos retirar os comandos que sucedem a composição do caractere “0”, após a descida de *DC*. Tais comandos estão explicados abaixo, tanto em texto como em tabela:

1. $0x20 = 0b00100000 \rightarrow$ *Function set*, dita os valores $PD = 0$, $V = 0$, $H = 0$ (uso do conjunto básico de instruções);
2. $0x80 = 0b10000000 \rightarrow$ *Set X Address of RAM*, determina a posição em 0 no eixo X, definindo cada um dos bits referentes à posição X na RAM (apresentados na imagem 4);
3. $0x41 = 0b01000001 \rightarrow$ *Set Y Address of RAM*, determina a posição em 1 no eixo Y, definindo cada um dos bits referentes à posição Y na RAM (apresentados na imagem 4).

Desses dados, extraiu-se a tabela a seguir:

Número	Comando MCU	Significado
1	Function set	Dita os valores $PD = 0$, $V = 0$, $H = 0$ (uso do conjunto básico de instruções)
2	Set X Address of RAM	Determina a posição em 0 no eixo X, definindo cada um dos bits referentes à posição X na RAM (apresentados na imagem 4)
3	Set Y Address of RAM	Determina a posição em 1 no eixo Y, definindo cada um dos bits referentes à posição Y na RAM (apresentados na imagem 4)

Seguindo, temos a imagem 11 abaixo, que apresenta os Bytes enviados ao display para serem escritas as colunas componentes do caractere “1”:

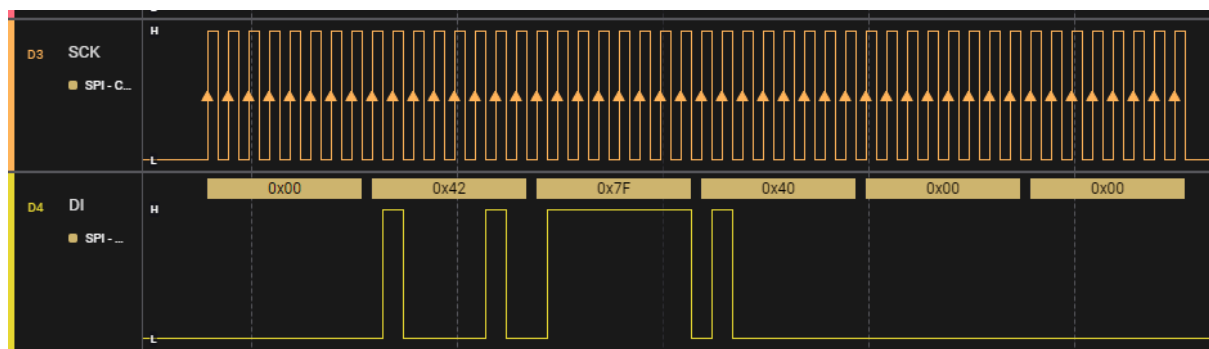


Imagem 11

Abaixo está demonstrado o processo de composição da imagem final da segunda linha do display, o caractere “1”:

1. $0x00 = 0b00000000 \rightarrow$



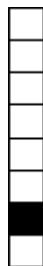
2. $0x42 = 0b01000010 \rightarrow$



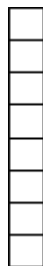
3. $0x7F = 0b01111111 \rightarrow$



4. $0x40 = 0b01000000 \rightarrow$



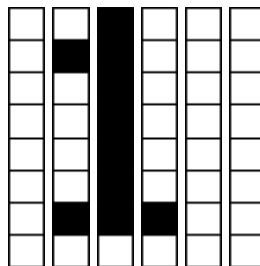
5. $0x00 = 0b00000000 \rightarrow$



6. $0x00 = 0b00000000 \rightarrow$



Seguindo, então, o procedimento já usado para compor o caractere “0”, do que se apresenta acima, é composto o caractere “1”, conforme se expõe logo abaixo:



Finalmente, pode-se compor a completude da imagem formada (com exceção das colunas em zero mais a direita nas duas linha):

