

Laboratório 6: Caracteres Porta RS232:

Professor: Felipe Calliari | Monitor: Cristiano Nascimento

Aluno: Pedro Gabriel Serodio Sales e Thiago Levis

Matrícula: 2211911 e 1812899

1 Introdução

O objetivo deste relatório é apresentar as soluções para o laboratório 6 da disciplina ENG1448 assim como apresentações do laboratório em vídeo.

2 Resolução

2.1 : Associacao SSegmentos

O primeiro passo foi deixar pronto a selecao do display de sete segmentos para fazer a associacao:

```
with hex
  select sseg(6 downto 0) <=
    "1111110" when "0000", -- 0
    "0110000" when "0001", -- 1
    "1101101" when "0010", -- 2
    "1111001" when "0011", -- 3
    "0110011" when "0100", -- 4
    "1011011" when "0101", -- 5
    "1011111" when "0110", -- 6
    "1110000" when "0111", -- 7
    "1111111" when "1000", -- 8
    "1111011" when "1001", -- 9
    "1110111" when "1010", -- A
    "0011111" when "1011", -- b
    "1001110" when "1100", -- C
    "0111101" when "1101", -- d
    "1001111" when "1110", -- E
    "1000111" when "1111", -- F
    "0110111" when others; -- H
  -- decimal point
```

end Behavioral;

(1)

Figura 1: Selecionando segmento

2.2 : Rising Edge

Fizemos a nossa logica do rising edge junto com estados Idle, Half-Startbit, rx-data e tx-data para fazer o controle da melhor forma. Os casos detalhados seguem abaixo

```
when idle =>
    bit_counter <= (others => '0');
    clk_counter_enable <= '0';
    if (rx = '0') then -- recebi um 0, começar a contar!! K/2
        clk_counter_enable <= '1';
        clk_counter_limit <= to_unsigned(216, clk_counter_limit'length);
        curr_state <= half_startbit;
    end if;

when half_startbit =>
    if (clk_counter_flag = '1') then -- contei k/2
        clk_counter_limit <= to_unsigned(433, clk_counter_limit'length);
        curr_state <= rx_data;
    end if;

when rx_data =>
    if (clk_counter_flag = '1') then -- contei k/2
        if (bit_counter /= 8) then -- considera a leitura de dados com 8 bits
            bit_counter <= bit_counter + 1;
            tx_reg <= rx & tx_reg(7 downto 1);
        else -- if (bit_counter = 8) then
            bit_counter <= (others => '0');
            curr_state <= tx_data;
        end if;
    end if;

when tx_data =>
    if tx_reg /= x"0D" then
        curr_state <= idle;
        tx <= tx_reg;
    end if;
```

(2)

Figura 2: Casos

2.3 : Emulador

Obtivemos os resultados esperados e usamos o emulador com a ajuda do professor Felipe e fomos verificando tecla por tecla no emulador, e como iam aparecendo no display de acordo com a tabela abaixo:

Dec	Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
0	0	000	NULL	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	Start of Header	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	Start of Text	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	End of Text	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	End of Transmission	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	Enquiry	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	Acknowledgment	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	Bell	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	Backspace	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	Horizontal Tab	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	Line feed	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	Vertical Tab	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	Form feed	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	Carriage return	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	Shift Out	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	Shift In	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	Data Link Escape	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	Device Control 1	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	Device Control 2	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	Device Control 3	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	Device Control 4	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	Negative Ack.	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	Synchronous idle	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	End of Trans. Block	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	Cancel	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	End of Medium	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	Substitute	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	Escape	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	File Separator	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	Group Separator	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	Record Separator	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	Unit Separator	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		Del

asciicharstable.com

(3)

Figura 3: Associacao das Letras ao respectivo Codigo no display SSeg

3 Vídeo de Apresentação

O vídeo de apresentação segue ao lado: **Lab 6: Caracteres Porta RS232**

ou acessar pelo url:<https://youtu.be/AQ6e85-1eQU>