

Figura 1: Controlo de um reservatório

## 1. Descrição do Controlo do Reservatório

Tendo em vista consolidar os conhecimentos adquiridos nas aulas teóricas sobre comunicações RS232, pretende-se adquirir, controlar e monitorizar grandezas digitais de um processo industrial. Este processo é controlado por um autómato programável (PLC) equipado com cartas de entradas e saídas digitais, uma carta analógica e uma carta de comunicações RS232. É também utilizado um computador com uma porta série RS232.

Pretende-se controlar (enviar) e receber no computador informações relativas a um reservatório de água (Figura 1). A ideia é implementar a comunicação entre o PC e o PLC, de modo a, por um lado, controlar o reservatório, enviando ordens do PC para o PLC, por outro lado, monitorizar o reservatório, recebendo no PC o estado do equipamento vindo do PLC. Segue uma descrição detalhada das entradas e saídas do PLC que serão utilizadas.

A saída digital **Y0** do PLC ativa um motor (Motor) que permite encher o reservatório.

A saída digital **Y1** permite ativar a electroválvula de entrada de água do reservatório (EV\_in) e por isso, tanto o motor como a electroválvula de entrada **Y0** devem ser ativados em simultâneo para que o reservatório possa ser cheio.

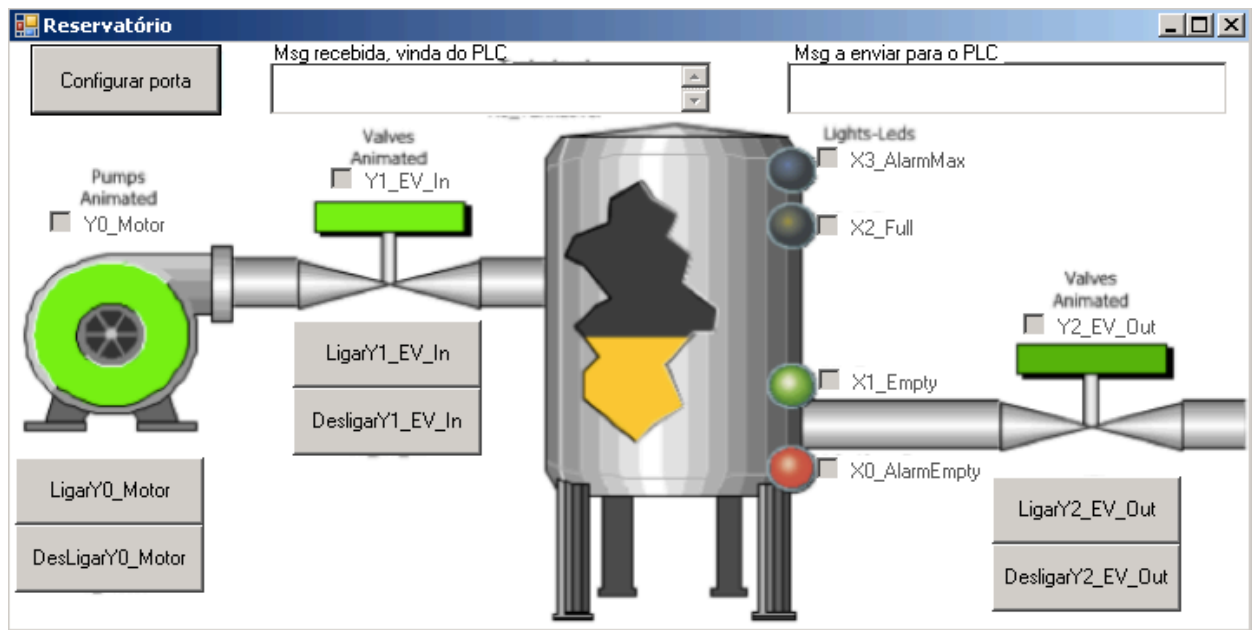
Existe também uma electroválvula que permite a saída da água do reservatório (EV\_out), ativada pela saída digital **Y2** do PLC.

O Reservatório tem 4 sensores de nível que são ativados quando detectam água, nessa altura aplicam 24 volt nas entradas digitais do PLC, nas entradas: X0, X1, X2 e X3.

Em síntese, as entradas e saídas do PLC acima descritas são:

- Y0 - Controla o Motor (Motor)
- Y1 – Controla a ElectroValvula de entrada (EV\_In)
- Y2 – Controla a ElectroVálvula de saída (EV\_Out)
- X0 – Detecta a água no nível mais baixo (AlarmEmpty)
- X1 – Detecta água no nível mínimo (Empty)
- X2 – Detecta água quando o reservatório está cheio (Full)
- X3 – Detecta água no nível de alarme mais alto do reservatório (AlarmMax)

A figura seguinte mostra um exemplo de um programa em Visual Basic para controlo do reservatório.



**Figura 2: Interface em Visual Basic**

A imagem de fundo utilizada para desenvolver a aplicação em cima está disponível no elearning.

## 2. Mecanismos de Controlo do Reservatório

Pretende-se que o controlo do reservatório funcione da forma descrita a seguir. Note-se que o valor analógico do nível não é usado no controlador, mas apenas para visualização.

- a) Quando a água fica abaixo do nível X1 ( $X1 = \text{False}$ ), a água deve ser bombeada para o tanque ( $Y0 = \text{True}$  e  $Y1 = \text{True}$ ) até que a água fique acima do nível X2 ( $X2 = \text{True}$ );
- b) Sempre que o nível de água esteja acima de X1 ( $X1 = \text{True}$ ) deve poder ser consumida água do reservatório ( $Y2 = \text{True}$ );
- c) Sempre que o nível de água esteja abaixo de X0 ( $X0 = \text{False}$ ) deve ser lançado um alarme a indicar que o reservatório está vazio (por exemplo numa caixa de texto);
- d) Sempre que o nível de água esteja acima de X3 ( $X3 = \text{True}$ ) deve ser lançado um alarme a indicar que o reservatório está cheio (por exemplo numa caixa de texto);

## 3. Protocolo de Comunicação

Esta secção visa definir as mensagens que circulam do PC para o PLC bem como no sentido contrário.

## Mensagem do PC para o PLC

A mensagem enviada pelo PC para o PLC serve para fazer o controlo do sistema, ou seja, para envio de comandos para ligar e desligar as saídas do PLC, quer dizer, o computador deve enviar a ordem para ligar ou desligar a(s) electroválvulas Y0, Y1, Y2 do PLC.

A mensagem é definida da seguinte forma:

**s\_ToPLC[Y0][Y1][Y2]\_e**

Em que **s** é o carácter de início de mensagem, **e** o carácter de final de mensagem, **ToPLC** indica a direção da mensagem (neste caso, enviada do PC para o PLC). Finalmente, **[Y0]** deve ser um carácter “1” ou “0” (ver na tabela ASCII os valores decimais correspondentes), indicando o estado desejado das saída correspondente do PLC.

A mensagem tem 12 bytes/caracteres. Por exemplo, se se pretender que o PLC tenha a saída Y0 desligada e as saídas Y1 e Y2 ligadas, a mensagem a enviar deve ser:

**s\_ToPLC011\_e**

consultando a tabela ascii (ver em baixo), a mensagem em cima seria, em decimal:

Decimal	115	95	84	111	80	76	67	48	49	49	95	101
Carácter	s	_	T	o	P	L	C	0	1	1	_	e

Em visual basic, depois de declarar um objeto SerialPort1, para enviar a mensagem indicada atrás deve fazer-se:

SerialPort1.Write(“s\_ToPLC011\_e”)

## Mensagem do PLC para o PC

O PLC deve enviar periodicamente na sua PORT3 (ficha DB9) mensagens para o PC informando sobre o estado do sistema. A mensagem é definida da seguinte forma:

**s\_ToPC[Y0][Y1][Y2][X0][X1][X2][X3] [Nível]\_e**

Em que **s** é o carácter de início de mensagem, **e** o caractere de final de mensagem, **ToPC** indica a direção da mensagem (neste caso, enviada do PLC para o PC). **[Y0]** deve ser um carácter “1” ou “0” (ver na tabela ASCII os valores decimais correspondentes), indicando o estado desejado das saída correspondente do PLC. **[Nível]** é um valor na gama de 0 a 100, que deve ser compreendido como um decimal e não um caracter como no caso de todos os outros bytes da mensagem, que indica em percentagem o nível de água no reservatório.

A mensagem contém 14 bytes/caracteres. Um exemplo, se o estado interno do PLC for Y0=0, Y1=0, Y2=1, X0=1, X1=1, X2=0, X3=1, o nível do reservatório a 67% (caracter “C”), a mensagem enviada deve ser:

# s\_ToPC0011101[67]\_e

consultando a tabela ascii (ver em baixo), a mensagem em cima seria em decimal:

Decimal	115	95	84	111	80	67	48	48	49	49	49	48	49	67	95	101
Caractér	s	_	T	o	P	C	0	0	1	1	1	0	1	C	_	e

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	<b>NUL</b> (null)	32	20	040	&#32;	<b>Space</b>	64	40	100	&#64;	<b>@</b>	96	60	140	&#96;	<b>`</b>
1	1	001	<b>SOH</b> (start of heading)	33	21	041	&#33;	<b>!</b>	65	41	101	&#65;	<b>A</b>	97	61	141	&#97;	<b>a</b>
2	2	002	<b>STX</b> (start of text)	34	22	042	&#34;	<b>"</b>	66	42	102	&#66;	<b>B</b>	98	62	142	&#98;	<b>b</b>
3	3	003	<b>ETX</b> (end of text)	35	23	043	&#35;	<b>#</b>	67	43	103	&#67;	<b>C</b>	99	63	143	&#99;	<b>c</b>
4	4	004	<b>EOT</b> (end of transmission)	36	24	044	&#36;	<b>\$</b>	68	44	104	&#68;	<b>D</b>	100	64	144	&#100;	<b>d</b>
5	5	005	<b>ENQ</b> (enquiry)	37	25	045	&#37;	<b>%</b>	69	45	105	&#69;	<b>E</b>	101	65	145	&#101;	<b>e</b>
6	6	006	<b>ACK</b> (acknowledge)	38	26	046	&#38;	<b>&amp;</b>	70	46	106	&#70;	<b>F</b>	102	66	146	&#102;	<b>f</b>
7	7	007	<b>BEL</b> (bell)	39	27	047	&#39;	<b>'</b>	71	47	107	&#71;	<b>G</b>	103	67	147	&#103;	<b>g</b>
8	8	010	<b>BS</b> (backspace)	40	28	050	&#40;	<b>(</b>	72	48	110	&#72;	<b>H</b>	104	68	150	&#104;	<b>h</b>
9	9	011	<b>TAB</b> (horizontal tab)	41	29	051	&#41;	<b>)</b>	73	49	111	&#73;	<b>I</b>	105	69	151	&#105;	<b>i</b>
10	A	012	<b>LF</b> (NL line feed, new line)	42	2A	052	&#42;	<b>*</b>	74	4A	112	&#74;	<b>J</b>	106	6A	152	&#106;	<b>j</b>
11	B	013	<b>VT</b> (vertical tab)	43	2B	053	&#43;	<b>+</b>	75	4B	113	&#75;	<b>K</b>	107	6B	153	&#107;	<b>k</b>
12	C	014	<b>FF</b> (NP form feed, new page)	44	2C	054	&#44;	<b>,</b>	76	4C	114	&#76;	<b>L</b>	108	6C	154	&#108;	<b>l</b>
13	D	015	<b>CR</b> (carriage return)	45	2D	055	&#45;	<b>-</b>	77	4D	115	&#77;	<b>M</b>	109	6D	155	&#109;	<b>m</b>
14	E	016	<b>SO</b> (shift out)	46	2E	056	&#46;	<b>.</b>	78	4E	116	&#78;	<b>N</b>	110	6E	156	&#110;	<b>n</b>
15	F	017	<b>SI</b> (shift in)	47	2F	057	&#47;	<b>/</b>	79	4F	117	&#79;	<b>O</b>	111	6F	157	&#111;	<b>o</b>
16	10	020	<b>DLE</b> (data link escape)	48	30	060	&#48;	<b>0</b>	80	50	120	&#80;	<b>P</b>	112	70	160	&#112;	<b>p</b>
17	11	021	<b>DC1</b> (device control 1)	49	31	061	&#49;	<b>1</b>	81	51	121	&#81;	<b>Q</b>	113	71	161	&#113;	<b>q</b>
18	12	022	<b>DC2</b> (device control 2)	50	32	062	&#50;	<b>2</b>	82	52	122	&#82;	<b>R</b>	114	72	162	&#114;	<b>r</b>
19	13	023	<b>DC3</b> (device control 3)	51	33	063	&#51;	<b>3</b>	83	53	123	&#83;	<b>S</b>	115	73	163	&#115;	<b>s</b>
20	14	024	<b>DC4</b> (device control 4)	52	34	064	&#52;	<b>4</b>	84	54	124	&#84;	<b>T</b>	116	74	164	&#116;	<b>t</b>
21	15	025	<b>NAK</b> (negative acknowledge)	53	35	065	&#53;	<b>5</b>	85	55	125	&#85;	<b>U</b>	117	75	165	&#117;	<b>u</b>
22	16	026	<b>SYN</b> (synchronous idle)	54	36	066	&#54;	<b>6</b>	86	56	126	&#86;	<b>V</b>	118	76	166	&#118;	<b>v</b>
23	17	027	<b>ETB</b> (end of trans. block)	55	37	067	&#55;	<b>7</b>	87	57	127	&#87;	<b>W</b>	119	77	167	&#119;	<b>w</b>
24	18	030	<b>CAN</b> (cancel)	56	38	070	&#56;	<b>8</b>	88	58	130	&#88;	<b>X</b>	120	78	170	&#120;	<b>x</b>
25	19	031	<b>EM</b> (end of medium)	57	39	071	&#57;	<b>9</b>	89	59	131	&#89;	<b>Y</b>	121	79	171	&#121;	<b>y</b>
26	1A	032	<b>SUB</b> (substitute)	58	3A	072	&#58;	<b>:</b>	90	5A	132	&#90;	<b>Z</b>	122	7A	172	&#122;	<b>z</b>
27	1B	033	<b>ESC</b> (escape)	59	3B	073	&#59;	<b>;</b>	91	5B	133	&#91;	<b>[</b>	123	7B	173	&#123;	<b>{</b>
28	1C	034	<b>FS</b> (file separator)	60	3C	074	&#60;	<b>&lt;</b>	92	5C	134	&#92;	<b>\</b>	124	7C	174	&#124;	<b> </b>
29	1D	035	<b>GS</b> (group separator)	61	3D	075	&#61;	<b>=</b>	93	5D	135	&#93;	<b>]</b>	125	7D	175	&#125;	<b>}</b>
30	1E	036	<b>RS</b> (record separator)	62	3E	076	&#62;	<b>&gt;</b>	94	5E	136	&#94;	<b>^</b>	126	7E	176	&#126;	<b>~</b>
31	1F	037	<b>US</b> (unit separator)	63	3F	077	&#63;	<b>?</b>	95	5F	137	&#95;	<b>_</b>	127	7F	177	&#127;	<b>DEL</b>

Source: [www.LookupTables.com](http://www.LookupTables.com)