

MSX Article

MARMSX

Joynet Básica

Resumo

O objetivo deste artigo é demonstrar como é possível construir uma rede alternativa para MSX, através dos conectores de *joystick* do micro. Será apresentado desde a construção do cabo até o funcionamento lógico da rede, em três artigos.

1- Introdução

A comunicação entre computadores envolve diversos benefícios. Com ela, é possível trocar arquivos entre máquinas, compartilhar programas, jogar o mesmo jogo com outras pessoas etc.

O MSX possui diversos dispositivos de comunicação, com o a interface RS-232, Modem e diversos projetos de rede em planejamento. Porém, todos eles apresentam problemas ao usuário do padrão MSX, devido a diversos problemas. A interface RS-232 e o modem não são mais fabricados e as placas de rede planejadas sairão a um preço alto.

A rede *joynet* surgiu a partir de uma lista da Internet, a lista de discussão internacional de MSX.

A montagem da rede foi padronizada, seguindo o modelo de rede criado pela Konami, para o jogo F1 Sibir it 3D, para MSX 2+. Por outro lado, não foi definido nenhum protocolo de comunicação ou divulgado qualquer trabalho para que pudesse servir de referência para a rede *joynet*.

A proposta deste artigo é mostrar a confecção de cabos *joynet* MSX-MSX e MSX-PC, previamente desenhados, como se controlar a porta de *joystick*, alertar sobre os perigos envolventes no projeto e apresentar um protocolo básico para comunicação entre dois computadores.

2- O cabo do sistema Joynet

Existem dois cabos distintos, mas com a mesma finalidade: um cabo para ligar dois MSX, via porta de *joystick* e outro cabo para ligar um MSX ao PC, via *joystick* e porta paralela.

A construção do cabo *joynet* é extremamente simples e barata. O custo total dos componentes do cabo fica em torno de R\$ 10,00.

2.1 - O cabo MSX-MSX

O material necessário para se construir um cabo MSX-MSX:

- 2 conectores DB-9 fêmeas.
- 2 caixas protetoras para os DB-9.
- 7 fios de telefone (diâmetro aproximado de 25 mm) com 1,5 m de comprimento.
- Ferro para soldar.

A numeração de 1 a 9 normalmente vem impressa no conector DB-9. Mas a numeração vista pelo lado da solda é o mostrado pela figura 2.1:

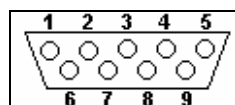


Fig. 2.1

Terá que ser feita a correlação entre os pinos do conector 1 com o conector 2, de acordo com a tabela 2.1. Por exemplo, um fio deve ser soldado com uma ponta no pino 1 do conector 1 e com outra ponta no pino 6 do conector 2.

Conector 1	Conector 2
1	6
2	7
3	8
6	1
7	2
8	3
9	9

Tabela 2.1

O mapa de fiação relativo à tabela 2.1, é apresentado na figura 2.2.

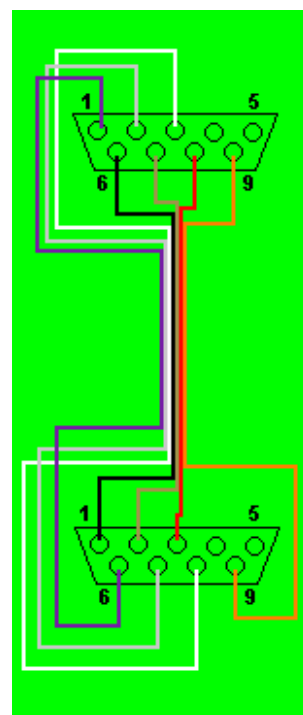


Fig. 2.2

2.2 - O cabo MSX-PC

O material necessário para se construir um cabo MSX-PC:

- 1 conector DB-9 fêmea.
- 1 conector DB-25 macho.
- 1 caixa protetora para o DB-9 e outra para o DB-25.
- 7 fios de telefone (diâmetro aproximado de 25 mm) com 1,5 m de comprimento.
- Ferro para soldar.

O DB-9 foi apresentado na seção anterior. O DB-25 é apresentado na figura 2.3. Assim como o DB-9, a imagem representa o lado da solda.

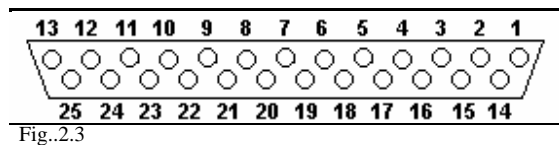


Fig..2.3

Terá que ser feita a correlação entre os pinos do conector do MSX com o conector do PC, de acordo com a tabela 2.2. O esquema de fios, é apresentado na figura 2.4.

Con. MSX	Con. PC
1	2
2	3
3	4
6	13
7	12
8	10
9	18 ~ 25

Tabela 2.2

O Pino 9 do MSX deverá ser ligado aos pinos 18, 19, 20, 21, 22, 23, 24 e/ou 25.

Estes pinos são o terra do PC.

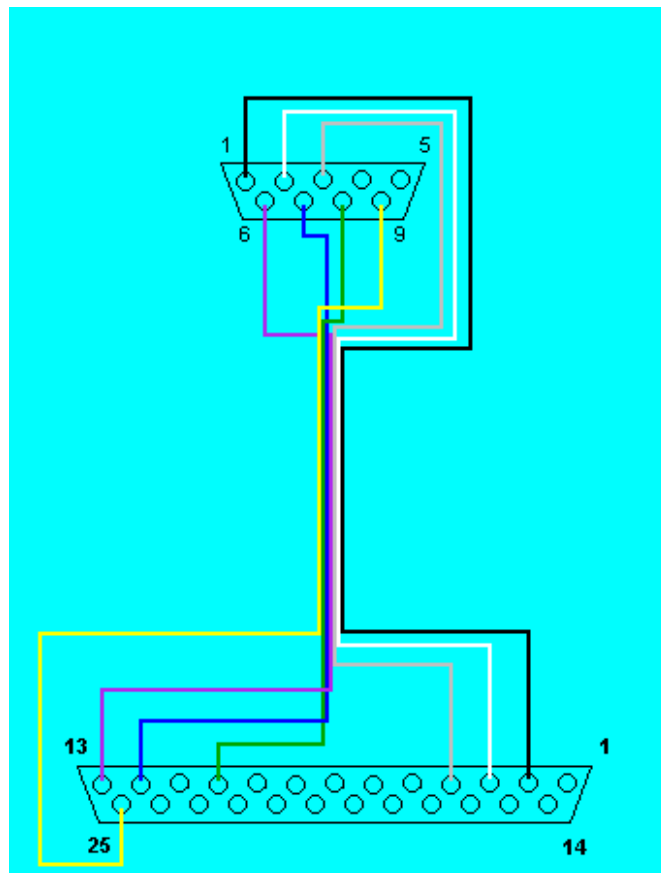


Figura 2.4

2.3 - A Porta de Joystick do MSX

O MSX contém, normalmente, duas portas de *joystick*. As portas são compostas de conectores padrão DB-9 macho, onde a configuração de pinos, vistos de frente (lado do encaixe), é a mesma da figura 2.1.

Quando um *joystick* é conectado a esta porta, os pinos delas se comportam conforme a tabela 2.3.

Pino	Função
1	Para cima
2	Para baixo
3	Esquerda
4	Direita
5	+ 5 Volts CC
6	Botão 1 de tiro
7	Botão 2 de tiro
8	Saída
9	GND (terra)

Tabela 2.3

Os pinos 1, 2, 3 e 4 somente RECEBEM sinais, o pino 8 somente ENVIA sinal e por fim, os pinos 6 e 7 fazem ambas funções.

Os sinais transitados são os mesmos 5 Volts CC. Portanto, podemos imaginar que o circuito de *joystick* é um simples circuito, com chaves que interrompem seus botões.

O circuito integrado responsável pelo controle das portas de *joystick* é o PSG AY-3-8910-A. PSG vem de *Programmable Sound Generator* ou gerador de sons programável. Com isso, concluímos que o processador de sons do MSX é o responsável por controlar as portas de *joystick*.

O PSG tem 15 registradores e se comunica com o processador Z80-A através de portas. Somente 2 registradores são utilizados para este fim: o 14 e o 15.

As portas de acesso ao PSG são as portas A0H, A1H e A2H.

A porta A0H é a responsável por escolher o registro atual do PSG, onde dados serão escritos pela porta A1H e lidos pela porta A2H.

As tabelas 2.3 e 2.4 descrevem a configuração de bits para os registradores 14 e 15, respectivamente.

Bit	7	6	5	4	3	2	1	0
Descrição	E. Cas.	M. Tec.	BTN2	BTN1	DIR	ESQ	BAIXO	CIMA

Tabela 2.3

Bit	7	6	5	4	3	2	1	0
Descrição	LED	SEL.	PULSO2	PULSO1	1	1	1	1

Tabela 2.4

As regiões sombreadas nas tabelas 2.3 e 2.4 indicam que estes bits não serão utilizados pelo sistema, por não serem relativos às portas de *joystick*.

2.4 - Descrição dos Registradores 14 e 15 do PSG

O registrador 14 tem apenas 6 bits que interessam a porta de *joystick*. São os bits 0, 1, 2, 3, 4 e 5. Todos estes bits estão relacionados ao estado dos botões do *joystick*. O valor 1 indica que o respectivo botão está solto, ou seja, não há corrente chegando ao PSG, enquanto que o valor 0 indica que o respectivo botão está sendo pressionado. O registrador 14 manipula a porta A do PSG, que é a entrada de dados.

O registrador 15 manipula a porta B do PSG, que é a porta de saída de dados. Ele tem 7 bits de interesse. O bit 6 seleciona a porta de *joystick* em uso. Infelizmente só poderá ser usada uma porta de *joystick* de cada vez. O valor 0 indica porta 1 e o valor 1 indica porta 2. Os outros bits enviam sinais para os pinos 6, 7 e 8 da porta de *joystick*, como será descrito a seguir.

Os bits de pulso (4 e 5) enviam um curto pulso através do pino 8. Se o bit 4 estiver em 0, nada é enviado. Se estiver em 1, o pulso é enviado. Os bits 0 e 1 indicam qual pino deve ter seu sinal elevado. O bit 0 corresponde ao pino 6 e o bit 1 ao pino 7. Ambos podem enviar dados ao mesmo tempo. De acordo com a tabela 2.4, observa-se as diversas possibilidades de enviar sinal pela porta de *joystick*. Leia-se os bits de 0 a 7 da direita para a esquerda do registrador 15.

REG 15	PINO 6	PINO 7	PINO 8
XX001111	0	0	VCC
XX011111	0	0	PULSO
XX001110	VCC	0	VCC
XX011110	VCC	0	PULSO
XX001101	0	VCC	VCC
XX011101	0	VCC	PULSO
XX001100	VCC	VCC	VCC
XX011100	VCC	VCC	PULSO

Tabela 2.4

O valor XX001111 é devolvido ao registrador 15, após cada operação. Portanto, para se manter algum pino aceso (6 ou 7), é necessário criar um “loop” que ajuste constantemente o registrador 15 para o valor desejado, de acordo com a tabela 2.4.

O pino 8, ao contrário dos pinos 6 e 7, mantém o nível de tensão alto.

2.5 - Manipulando dados nos Registradores do PSG

O primeiro passo para se manipular um registrador do PSG, seja na leitura ou na escrita, é indicá-lo à porta A0H. Então, o seguinte código em linguagem de máquina deve ser feito:

```
LD A,&H0F
OUT (&HA0),A ; Para escolher o registrador. 15
```

Ou

```
LD A,&H0E
```

OUT (&HA0),A ; Se o registrador a ser manipulado é o 14.

Sempre quando se for manipular um registrador diferente do atual, os comandos LD A,<reg> e OUT (&HA0),A deverão ser dados.

O passo seguinte é ler ou escrever. Para escrever, será usada a porta A1H e para ler, a porta A2H.

Deve-se ter em mente que, quando desejar alterar o valor de algum registrador, alguns bits não deverão ser alterados. Para isto, deve utilizar alguns artifícios como as operações lógicas AND e OR.

A operação AND é utilizada para forçar zero em alguns bits e manter o resultado anterior em outros. Onde o numero auxiliar tiver bit 0, o bit relativo do numero de dado será ajustado para 0, e aonde tiver bit 1 no auxiliar, o bit relativo no dado será mantido. O seguinte exemplo mostra como é isto de fato:

```
DADO = &B 10101101
AUX =  &B 00001111
(AND &B00001111 )
DADO= &B 00001101
```

A operação OR é utilizada para forçar o um em alguns bits e manter o resultado anterior em outros. Faz o mesmo que o AND, só que com 1. Seguindo o exemplo anterior:

```
OR &B00001111
DADO=&B10101111
```

Estes cuidados são indispensáveis para a manipulação dos registradores. Se quisermos alterar o valor de algum registrador, teremos que ler ele, usar a operação lógica adequada e verificar qual numero binário auxiliar que alterará o bit(s) desejado(s), sem alterar os demais.

Para ilustrar, suponha-se que deseja saber o estado da porta 1 do joystick. O seguinte código em LM deverá ser digitado:

```
LD A,&H0F
OUT (&HA0),A      ; Escolhe o registrador 15
IN A,(&HA2)       ; Lê o registrador 15, passando-o para A
AND &HBF         ; Faz operação A AND x0xxxxxx
OUT (&HA1),A     ; Escreve no registrador 15
                  ; Aqui foi selecionado o joystick 1

LD A,&H0E
OUT (&HA0),A     ; Escolhe registrador 14
IN A,(&HA2)      ; A contém os dados da porta 1 do joystick
```

2.6 - Testando se o cabo está ligado a outro MSX

Este é um teste básico, porém necessário para fazer verificações periódicas se o cabo está conectado a outro MSX.

Para este teste, basta apenas verificar se o estado do pino 8 está alto. Como o pino 8 gera pulsos, ou seja, estados alto e baixo alternadamente, é necessário fazer a verificação em um certo período de tempo. Então, temos:

```
LD A, &H0E
OUT (&HA0), A
LD B, &HFF
LD A, 1
LD (&HC110), A
LOOP: IN A, (&HA2)
BIT 2, A
JR Z, FLG
DJNZ LOOP
LD A, 0
LD (&HC110), A
FLG: RET
```

A seguinte função em pascal poderia ser criada:

```
function cabo_ligado:boolean
var res : integer absolute $C100
begin
  inline ( $3E / $0E / $D3 / $A0 / $06 / $FF / $3E / $01 / $32 / $4E / $2F / $DB / $A2 /
$CB / $57 / $28 / $07 / $10 / $ F8 / $ 3E / $00 / $32 / $4E / $2F / $C9 );
  if (res = 0) then
    return false;
  else
    return true;
  end;
```

3 - Créditos

Este artigo foi escrito por Marcelo Teixeira Silveira, engenheiro de sistemas e computação, formado pela UERJ.

Este é o primeiro dos três artigos sobre a Joynet:

- Joynet Básica
- Joynet Comunicação
- Joynet Protocolos