

EA079 Laboratório de Micro e Minicomputadores: Hardware Laboratório III

Projeto de Interface com o Teclado

1 Objetivo do Laboratório III

1. Entender as características elétricas e funcionais do pino \overline{IRQ} .
2. Projetar e implementar uma interface entre o MC68HC11 e um teclado de 12 teclas.
3. Desenvolver três diferentes versões de um programa que capture teclas do teclado e identifique as teclas digitadas.

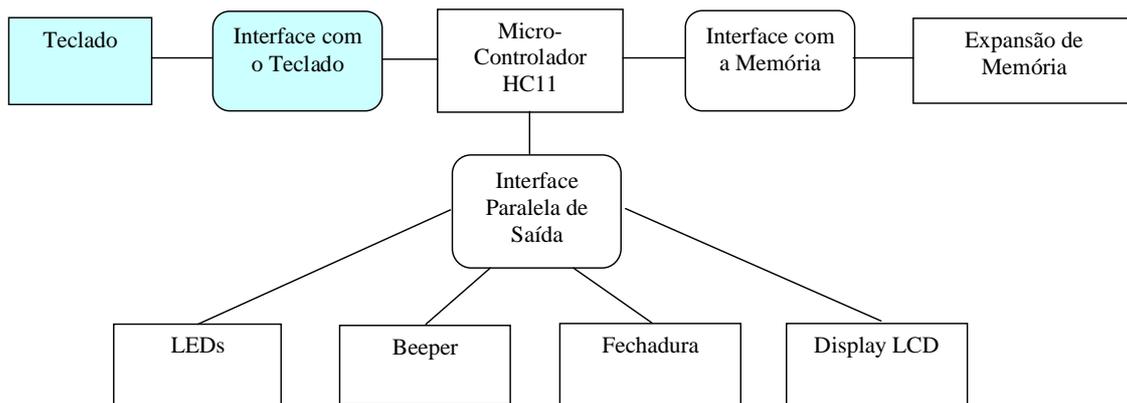


Figura 1 - Arquitetura Global do Sistema e Componentes sendo Desenvolvidos neste experimento

2 Material

- Placa PDHC11-FEEC
- Teclado.
- Placa de montagem wire-wrap
- Componentes disponíveis no almoxarifado da FEEC (p.ex.: soquetes, resistores, rolo de fio para wire-wrap, CI's)
- Ferramentas de montagem: kit de wire wrap
- Ferramentas de depuração: osciloscópio e multímetro

A disposição do teclado pode ser visualizado na figura 2 a seguir. Os conectores podem ser visualizados nas figuras 3 e 4 na sequência.

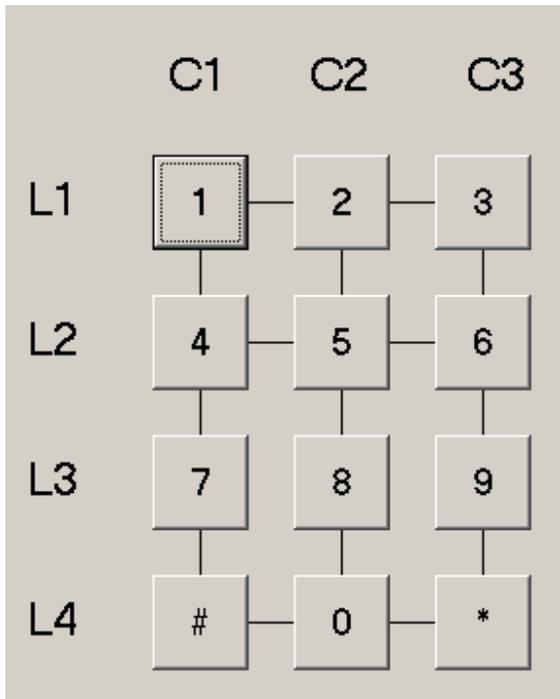


Figura 2 - Configuração do Teclado
Li = Linha - Ci = Coluna

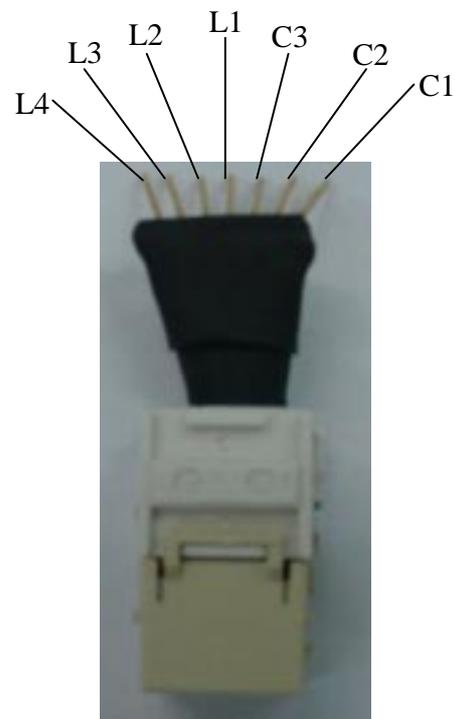


Figura 3 - Informações do Conector do Teclado
(Vista Superior)

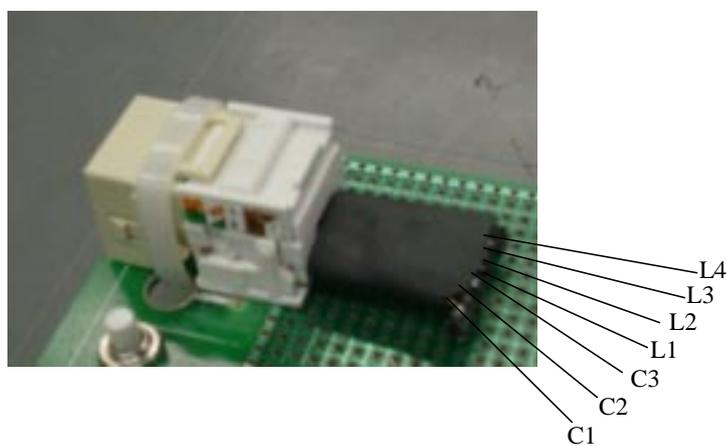


Figura 4 - Informações do Conector do Teclado (Vista Lateral)

3 Cronograma de Atividades

Este experimento será desenvolvido em três aulas. No início da primeira aula, cada grupo deve já ter um esboço dos esquemáticos do projeto das interfaces com o teclado e ter providenciado junto com o almoxarifado da FEEC os componentes necessários para a montagem.

- 1a aula: montagem/teste de interface com o teclado.
- 2a aula: programação da interface com o teclado.
- 3a aula: finalizar testes e demos.

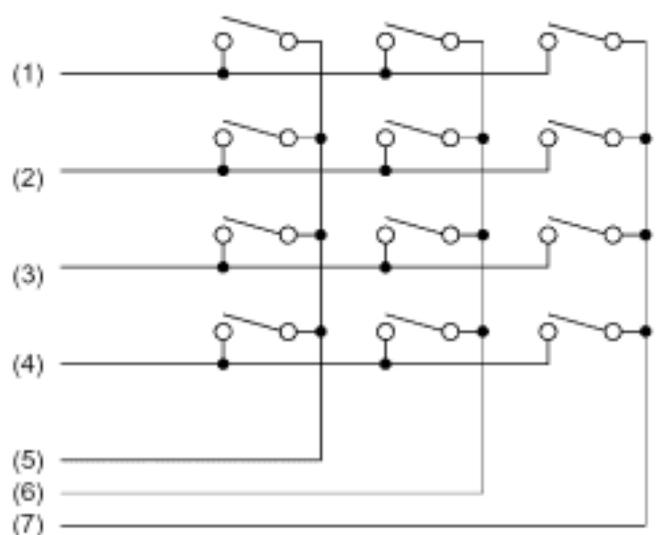
4 Projeto Conceitual

Para podermos obter informações do teclado, ao contrário dos demais dispositivos até agora implementados (LEDs, Display, Beeper e Controle da Fechadura), teremos que adotar uma configuração de hardware que promova a escrita e a leitura de dados. Isso exige que tomemos um certo cuidado na elaboração do projeto de hardware. Considere portanto atentamente as descrições a seguir:

4.1 Interface com o Teclado

Diferentemente de dispositivos meramente de saída, como o beeper e os LEDs, o teclado é um dispositivo de entrada/saída.

1. O teclado matricial de 12 teclas pode ser organizado de acordo com o seguinte esquema:



Como se consegue identificar quando uma tecla é acionada? Em que nível lógico fica uma linha, quando nenhuma tecla associada for acionada?

2. Um problema crítico do teclado é o comportamento de "vai-vem" (bounce) das suas teclas, podendo um mesmo sinal ser interpretado como vários eventos. A solução seria efetuar o chamado "debounce" das teclas por hardware ou por software. Qual seria uma solução por hardware? E uma solução por software?
3. Optaremos neste projeto pelo esquema de interrupção mascarável usando o pino \overline{IRQ} . Este pino pode ser sensível a borda de descida ou sensível a nível. O que você entende por sensível a borda e sensível a nível?
4. Enquanto MC68HC11 estiver atendendo uma solicitação de interrupção, a linha \overline{IRQ} deve ser mantido em nível alto ou em nível baixo?

5. O teclado deve ser controlado por dois registradores de 8 bits. Ambos registradores devem estar no mesmo endereço. Na leitura ele funciona como um registrador de estado (status register) e para escrita ele funciona como um registrador de controle.

Os bits do registrador de controle KEYCTRL devem ser:

Bits:	B7	B6	B5	B4	B3	B2	B1	B0
Significado:	CI	EI	X	X	L3	L2	L1	L0

cujo significado são:

- L3,L2,L1,L0 Linhas de acionamento das linhas do teclado
- EI (Interrupt enable) Habilita a interrupção do teclado
- CI (Interrupt clear) Limpa o pedido de interrupção

Os bits do registrador de status KEYSTAT são:

Bits:	B7	B6	B5	B4	B3	B2	B1	B0
Significado:	FI	X	X	X	X	C2	C1	C0

cujo significado são:

- C2,C1,C0 Leitura das colunas do teclado
- FI (Interrupt flag) Indica solicitação de interrupção

Princípio de funcionamento:

O funcionamento da interface de teclado é baseado numa solução intermediária entre software e hardware. Para esperar o apertado de uma tecla, programa-se as linhas L3-L0 com zero. Quando qualquer tecla for apertada, o flag deve ficar verdadeiro. Para verificar qual foi a tecla apertada, é necessário um procedimento de varredura: aciona-se apenas uma única linha de cada vez, L0, ou L1, ou L2 ou L3 e verifica-se em C2, C1 e C0 qual foi a tecla apertada. Neste instante a rotina de interface de teclado deve retornar o código ASCII da tecla apertada. Observe que a tecla pode ficar apertada por um certo tempo, isto não significa que uma nova tecla foi apertada. Para evitar que uma nova interrupção seja solicitada, recomenda-se neste instante desabilitar localmente a interrupção do teclado usando-se o sinal EI do registrador KEYCNTL. Quando a tecla for solta, limpa-se o pedido de interrupção, acionando-se o sinal CI e habilita-se novamente a interrupção. Para que a espera de se soltar a tecla fique consumindo CPU, sugere-se que uma outra subrotina acionada pelo temporizador de 1 ms seja usada para verificar o estado da rotina de leitura da tecla.

6. Esboce o esquemático do circuito de interface para o teclado de acordo com as convenções acima. O endereço dos registradores KEYCNTL e KEYSTAT é o que foi definido no experimento anterior (Interfaces de Saída).
7. Liste todos os componentes necessários para a implementação da interface com o teclado.
8. Quando se liga o HC11 com a interface de teclado, é importante que sua interface de interrupção não seja habilitada. Como se pode garantir isto no reset?

5 Hardware

Para a implementação deste projeto recomendamos o mesmo procedimento apresentado no roteiro do segundo experimento.

5.1 Montagem da Interface com o Teclado

1. Recomenda-se primeiro testar a interface de teclado usando o comando mm do BUFFALLO (Qual é a sequência de comandos mm para verificar se a interface de teclado está funcionando?)
2. Em seguida, recomenda-se testar a interface de teclado usando um programa simples, sem interrupção, que fica esperando os eventos acontecerem. Chamamos atenção de que nesta versão o ruído da tecla pode fazer com que uma tecla apareça apertada várias vezes. Faça duas subrotinas:
 - `int readkey();`
 - `int getkey();`

A subrotina `readkey` deve retornar 0 caso nenhuma tecla seja apertada ou, em caso contrário, deve retornar o código ASCII da tecla apertada. A subrotina `getkey` chama a subrotina `readkey` e fica esperando até que uma tecla seja apertada, retornando o seu valor ASCII. Observe que apesar do código ASCII ser de 7 bits, as subrotinas retornam um inteiro.

- Teste as suas subrotinas com um programa principal que fique ecoando na tela a tecla batida, usando a função `printf`.

Faça também a subrotina

- `keyinit();`

responsável pela inicialização do hardware do teclado.

3. Faça agora **uma segunda versão** da subrotina de leitura de teclado. Agora utilize uma interrupção temporizada de 1 ms. A cada interrupção, verifique se o estado do teclado esteja de acordo. Projete uma máquina de estado para tratar os eventos do teclado, porém varrendo-o a cada 1 ms. Nesta versão do programa, utilize o mesmo nome da subrotina `readkey` e teste com o mesmo programa principal anterior. Altere também a subrotina de inicialização do teclado, que deve conter também a inicialização do temporizador. A comunicação entre esta subrotina e a subrotina de serviço do temporizador deve ser feita por variáveis globais.
4. Finalmente, **a terceira versão** deve utilizar além da versão anterior, usando interrupção temporizada de 1ms, a interrupção do teclado. Nesta versão utilize o mesmo programa principal.

Referências

[1] Alan Clements, Microprocessor Systems Design: 6800 Hardware, Software, and Interfacing, 3rd edition, PWS Publishing Company, 1997 (ISBN: 0-534-94822-7)

[2] -. HC11 M68HC11A8 Technical Data. Motorola.

[3] -. HC11 M68HC11 Reference Manual. Motorola.