

Trabalhando com memórias I2C 24xxxx

Por Renie S. Marquet em 01.02.2005 (<http://reniemarquet.sites.uol.com.br>)

Este documento descreve os procedimentos para utilização de memórias 24xxxx nos barramentos I2C, porém, o material exposto poderá ser aproveitado como base para utilização de outros dispositivos compatíveis com o padrão I2C.

Para trabalhar com memórias I2C é necessário compreender os seguintes itens:

- Endereço do dispositivo.
- Linhas de comunicação.
- Bit de início (Start bit).
- Bit de final (Stop bit).
- Sinal de recebimento (Acknowledge).

Endereço do dispositivo

Devido a comunicação I2C dar-se com o uso de apenas 2 linhas e todos os dispositivos estarem conectados nestas linhas, é necessário um meio de identificar cada dispositivo de maneira unívoca. A identificação dos dispositivos I2C é feita através dos endereços atribuídos a cada um deles. Alguns dispositivos I2C já vem com seu endereço fixo, ou uma faixa de endereços selecionáveis através de pinos de controle, no caso das memórias 24xxxx, geralmente existem até 3 linhas de controle (A0 a A2), que conforme o nível lógico aplicado em cada uma, irão compor os endereços das mesmas (poderá ser de 0 a 7).

O barramento I2C tem capacidade de trabalhar com até 256 dispositivos, porém este montante foi dividido em faixas cabendo a cada tipo uma faixa. As memórias 24xxxx, ocupam a faixa de endereços de 'A0'h até 'AF'h. Como as memórias são dispositivos de escrita e leitura, o bit 0 do endereço foi utilizado para identificar o tipo de operação desejado, sobrando assim 3 bits para selecionar a memória dentre as 8 possíveis.

No caso de memórias, o usuário deve prestar a atenção para a existência de dois tipos de endereços, o que identifica o dispositivo no barramento e o endereço dos dados internos da memória a serem acessados.

Algumas memórias de baixa capacidade dispõem apenas de 1 ou 2 linhas externas para definir o endereço do dispositivo, utilizando os bits restantes do byte de controle (endereço no barramento) para compor o endereço interno a ser acessado.

Para identificar o tratamento a ser dado ao endereço do dispositivo e aos endereços de acesso aos dados, o usuário deverá ter em mãos o datasheet correspondente ao chip de memória que deseja utilizar.

Obs: Os pinos de configuração de endereço do dispositivo podem variar de fabricante para fabricante.

Linhas de comunicação

A comunicação I2C dá-se através de apenas 2 linhas, clock e dados (SCL e SDA).

O dispositivo mestre (master) é responsável pela geração do sinal de clock, enviar os comandos com o endereço do dispositivo de destino e verificar o sinal de reconhecimento gerado pelo dispositivo.

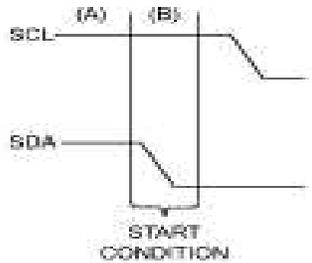
A linha de clock é responsável pela sincronização do envio/recebimento entre os dispositivos mestre e escravo (master e slave). O clock só estará presente durante a comunicação, quando o dispositivo mestre não estiver se comunicando com algum escravo a linha de clock, bem como a linha de dados, terão que ficar em nível alto.

Bit de início (Start bit)

O bit de início é responsável por avisar aos dispositivos escravos que o dispositivo mestre irá enviar um comando(+endereço), após o bit de início o dispositivo mestre envia um byte de controle que identificará o endereço do dispositivo escravo a que se destina o comando, e o tipo de comando (escrita / leitura no caso das 24xxxx).

Para gerar o bit de início, a partir do estado de repouso (linhas de clock e dados em nível alto), o dispositivo mestre mantém a linha de clock em

nível alto e leva a linha de dados a nível baixo, após um período de 1 semi-ciclo, leva a linha de clock a nível baixo para começar a enviar os bits do comando.

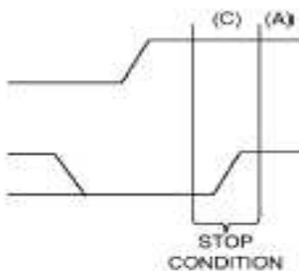


Cada bit a ser transmitido deverá ser colocado na linha de dados enquanto a linha de clock estiver em nível baixo. O(s) dispositivo(s) escravo(s) irá(rão) receber(em) os bits durante o semi-ciclo alto do sinal de clock.

Bit de final (Stop bit)

O bit de final é o aviso que o dispositivo mestre envia ao dispositivo escravo indicando o fim do comando ou da comunicação.

Para gerar o bit de final, o dispositivo mestre mantém a linha de clock em nível alto e leva a linha de dados também para nível alto. Enquanto não for necessário outro envio/recebimento de dados dos dispositivos do barramento I2C, as linhas de clock e sinal tem que serem mantidas em nível alto (repouso - standby).



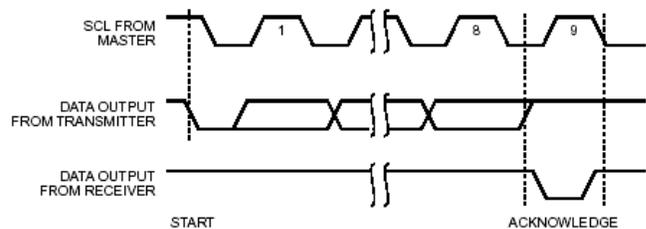
Sinal de recebimento (Acknowledge).

O sinal de recebimento deve ser fornecido após o envio de um byte pelo dispositivo que o está recebendo, ou seja, quando o dispositivo mestre envia um byte (comando/endereço/dado), o dispositivo escravo deverá fornecer o sinal de recebimento no pulso de clock seguinte; quando o dispositivo escravo envia um dado solicitado pelo

dispositivo mestre, ao final do recebimento o dispositivo mestre deverá fornecer o sinal de recebimento (para leitura seqüencial).

O dispositivo mestre é sempre o responsável pela geração do sinal de clock, portanto, para que o sinal de recebimento seja enviado ou recebido pelo escravo, o mestre deverá gerar um ciclo de clock extra após cada byte enviado/recebido.

O sinal de recebimento é gerado quando o sinal de clock estiver em nível baixo colocando a linha de sinal também em nível baixo; a linha de dados deverá ser lida pelo dispositivo que estiver aguardando a confirmação quando o clock voltar ao nível alto.



O sinal de recebimento tem diversas funções:

- Quando o dispositivo mestre envia um comando, somente o dispositivo escravo identificado pelo endereço deverá gerar o sinal de recebimento, caso não seja gerado um sinal de recebimento, significa que não existe um dispositivo escravo com o endereço indicado presente no barramento I2C ou o mesmo encontra-se ocupado (provavelmente executando o ciclo interno de gravação).
- Quando o mestre está enviando um dado ou endereço interno (no caso das 24xxx) ao escravo, o aviso de recebimento informa ao mestre que o dispositivo escravo está preparado para receber um novo dado ou o byte menor do endereço interno. No caso do mestre estar enviando dados, o escravo ao enviar o sinal de recebimento, incrementará o endereço interno do buffer de dados (veja o tópico Byte x Página).
- Quando o mestre está recebendo um dado solicitado ao escravo, o aviso de recebimento dispara o incremento do endereço de dados interno da memória escrava, possibilitando assim uma leitura seqüencial sem a necessidade de informar ao dispositivo endereço por endereço (leitura seqüencial).

Ordem de transmissão dos dados.

Os dados são enviados através do barramento I2C começando do bit de mais alta ordem (bit 7) até o de mais baixa ordem (bit 0).

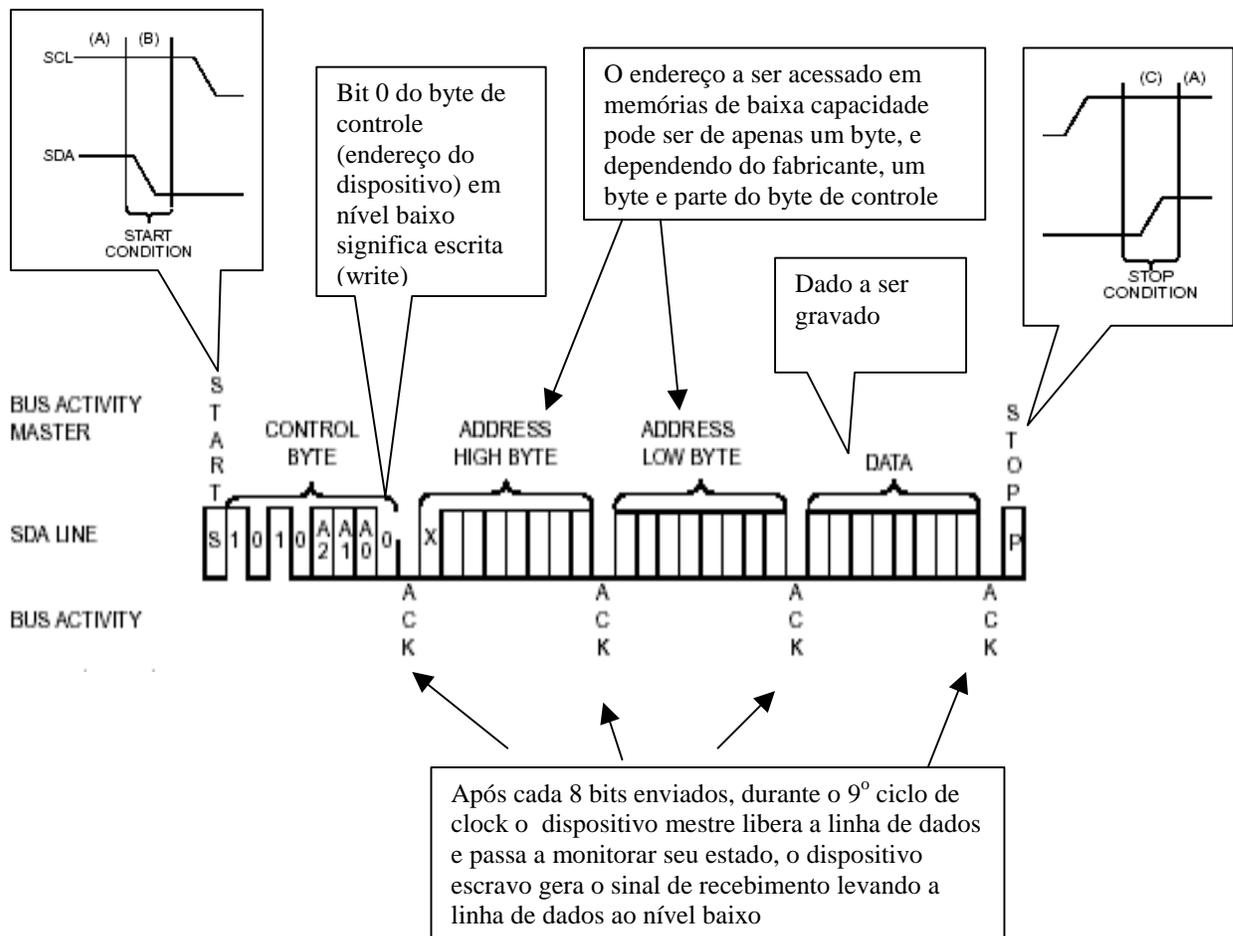
Nas memórias de capacidade acima de 256 x 8 é necessário utilizar dois bytes para indicar o

endereço interno da memória a ser acessado. O byte de maior ordem deverá ser enviado primeiro, seguido do de menor ordem.

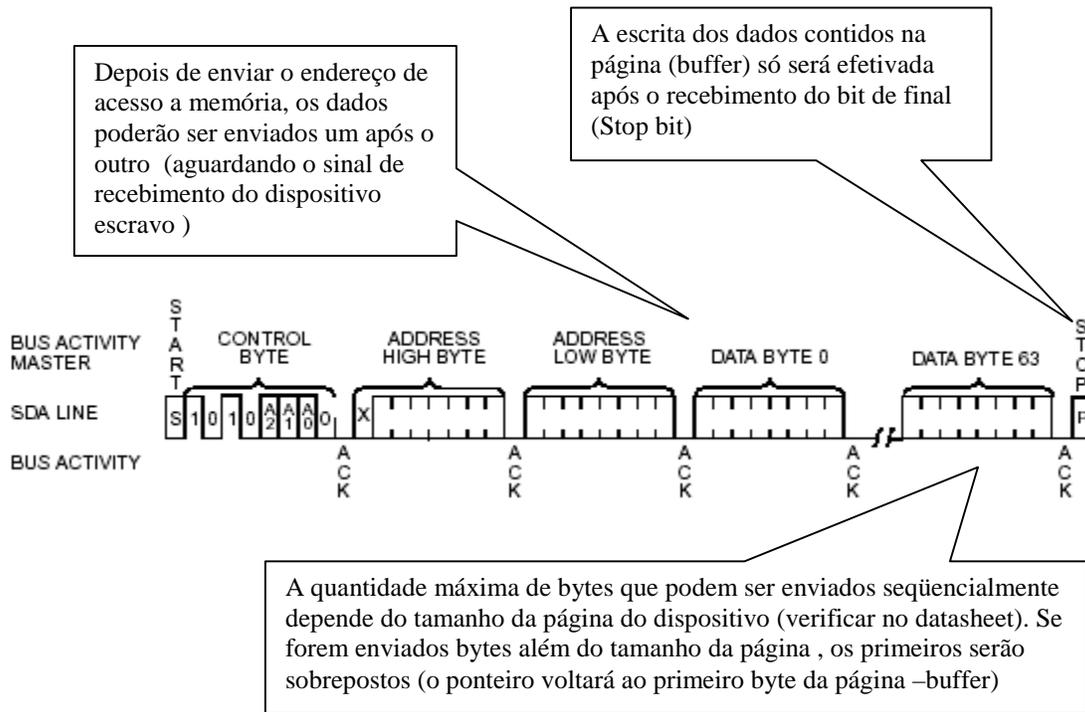
Tipos de acesso.

Lembrando, excluindo o bit de início e o bit de final, todos os demais bits devem ter seu valor definido na linha de dados (DAS) enquanto o sinal de clock (SCL) estiver em nível baixo.

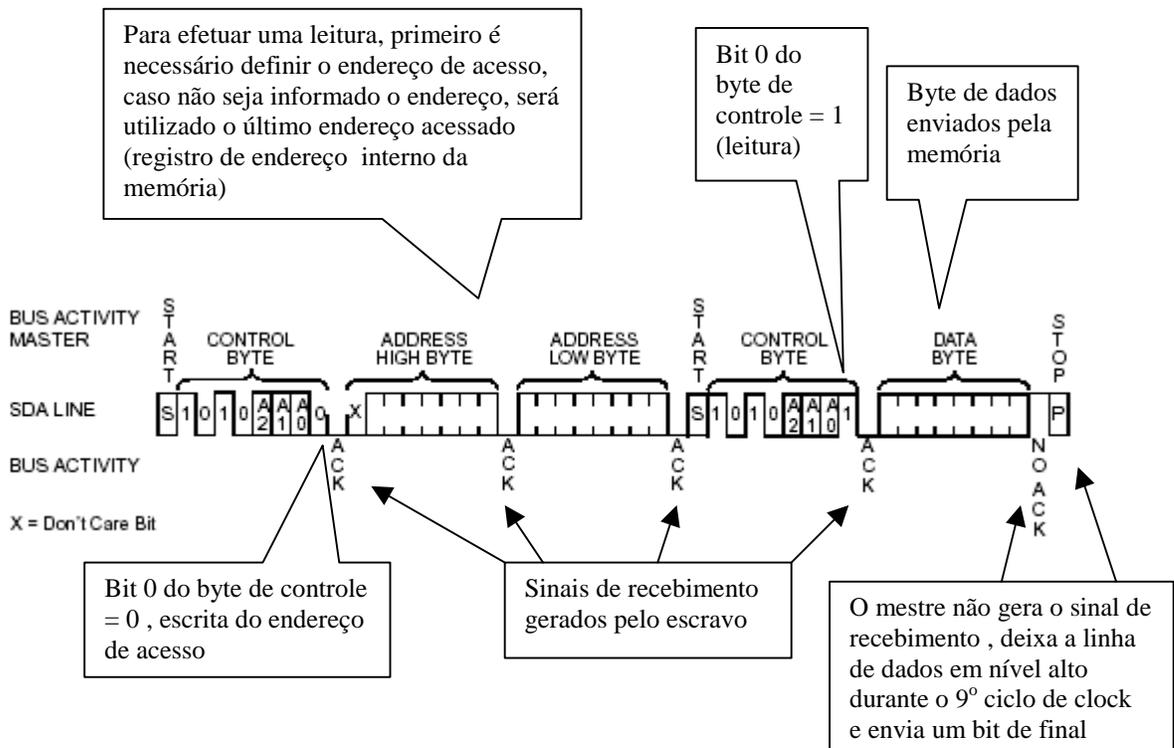
Escrevendo um único byte:



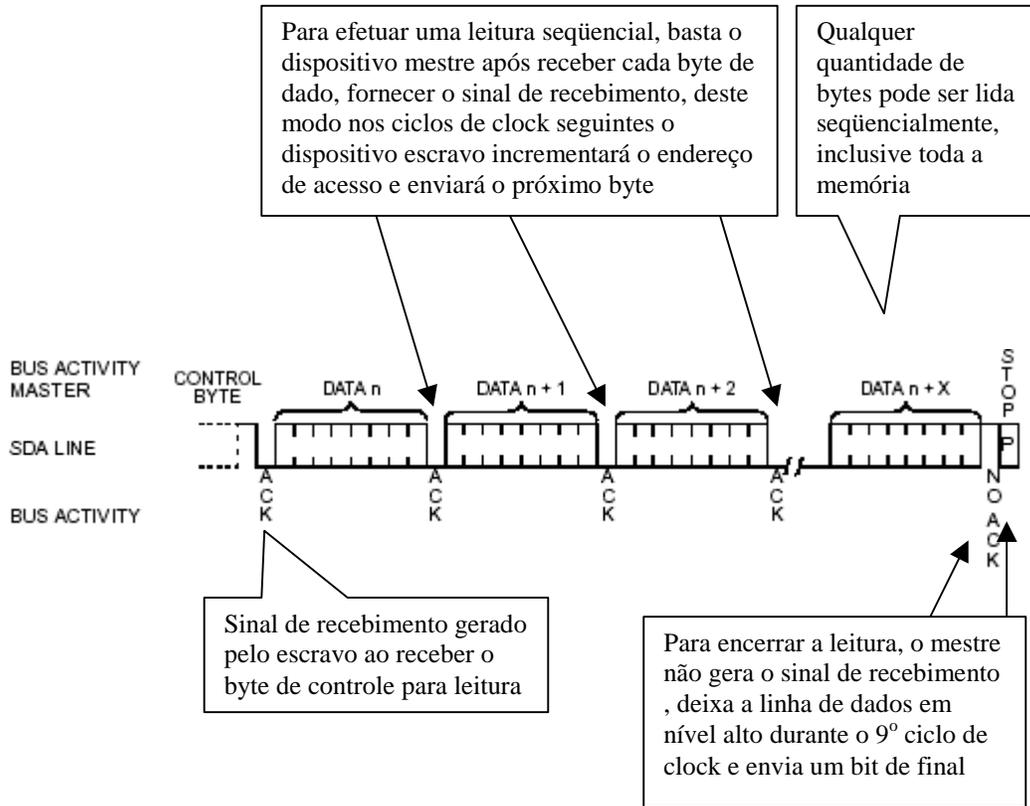
Escrita por página(sequencial):



Leitura aleatória:



Leitura seqüencial:



Considerações finais.

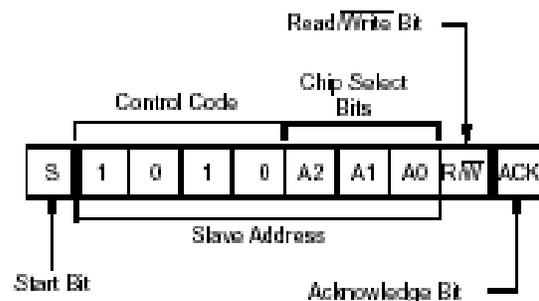
É muito importante o usuário verificar no datasheet exato da memória que vai utilizar as configurações disponíveis de endereço para o dispositivo.

Apesar da nomenclatura de algumas memórias serem muito semelhante e possuírem o mesmo total de bytes, o método de endereçamento de acesso interno e de identificação do dispositivo pode variar muito dependendo do fabricante.

Exemplo:

A 24WC16 da Catalyst ® não tem nenhum pino para configurar o nr. do dispositivo externamente, só pode ser utilizada uma em um barramento I2C, os 3 bits do byte de controle são utilizados para completar o endereço de acesso interno (+ 1 byte para o endereço de acesso).

A 24LC16 da Microchip ® possui 3 pinos que conforme a voltagem aplicada em cada um, determinará o endereço do dispositivo, podendo deste modo serem usados até 8 chips em um barramento I2C; para o endereço de acesso interno são utilizados 2 bytes após o byte de controle.



CAT24WC16	1	0	1	0	a10	a9	a8	R/W
-----------	---	---	---	---	-----	----	----	-----