

Master Synchronous Serial Port

Periférico responsável pela configuração do SPI e do I2C do microcontrolador.

Registadores

- SSPxSTAT → configurações gerais do MSSP
- SSPCON 1 e 2 → configurações gerais do MSSP
- SSPxADD → Este registrador é de endereçamento (modo slave) ou define o baudrate da comunicação (modo master)
- SSPxBUF → Buffer de envio ou recebimento de dado
- SSPxSR → Buffer de envio ou recebimento de dado

O modo SPI pode trabalhar com 3 pinos do microcontrolador no modo MASTER

- Serial Data Out (SDO) - RC5/SDO
- Serial Data In (SDI) - RC4/SDI/SDA
- Serial Clock (SCK) - RC3/SCK/SCL

Ou com 4 pinos no modo SLAVE, quando é acrescentado à função Slave Select (SS) - RA5.

As diferenças entre os protocolos de comunicação são:

Tecnologia	Barramento de comunicação	Taxa máxima	Fluxo de dados
UART (RS232)	2	115,2 kbps	Half ou Full Duplex
SPI	3 + número de Slaves	2 Mbps	Full duplex
I2C	2 (até 127 dispositivos)	400 kbps	Half duplex

Sobre o protocolo I2C

A comunicação I2C, que significa Inter-Integrated Circuit, é um protocolo de comunicação serial síncrona de dois fios que foi desenvolvido pela Philips (agora NXP Semiconductors). Ele é utilizado para interconectar vários dispositivos de hardware em sistemas embarcados e microcontroladores.

A comunicação I2C é caracterizada por sua simplicidade e eficiência em termos de utilização de pinos. Ela requer apenas dois fios para transmitir dados entre dispositivos: um fio para o clock serial (SCL) e outro para os dados (SDA).

O I2C é apropriado para periféricos onde simplicidade e baixo custo são mais importantes que velocidade. Algumas aplicações são:

- Descrever dispositivos conectados via pequenas tabelas de configuração ROM para habilitar operações plug and play, como detecção de presença serial de EEPROMs em módulos de memória dual in-line, e Dados de identificação de display estendido para monitores VGA, DVI e HDMI
- Monitoramento de sistema para PC via SMBus. SMBus é um barramento simples de dois fios de terminação única para fins de comunicação leve. É comumente

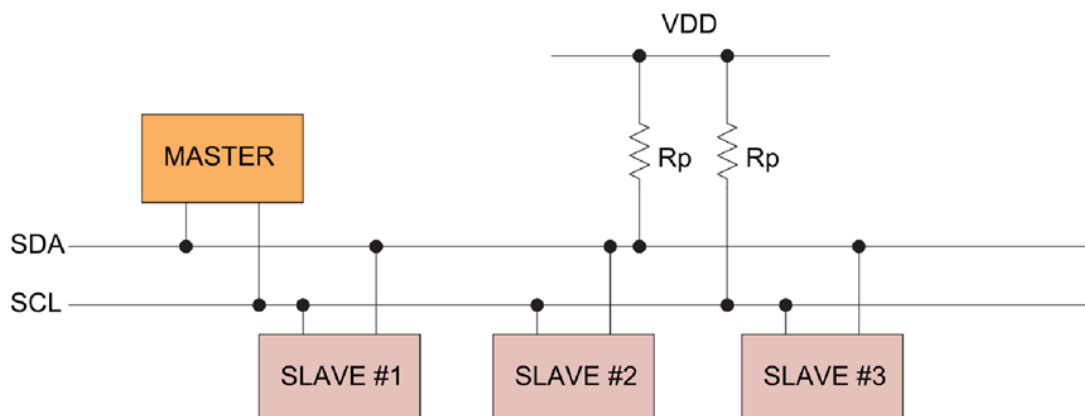
encontrado em placas-mãe de computador para comunicação com a fonte de alimentação para instruções ON/OFF.

- Acesso à clocks de tempo de real e chips de NVRAM que armazenam configurações
- Acessar DACs e ADC de baixa velocidade
- Configurações de monitores: luz de fundo; contraste, balanço de cor, tonalidade
- Volume em alto-falantes inteligentes
- Controlar pequenos displays LCD e OLED
- Sensores
- On/Off de fontes de componentes do sistema
- Comunicação com memórias, displays e outros microcontroladores

A quantidade de nodes em que podem compartilhar um canal de comunicação I2C é limitado ao número de endereçamento compatível e a capacitância máxima de 400pF, o que restringe a comunicação para poucos metros.

Como funciona o protocolo

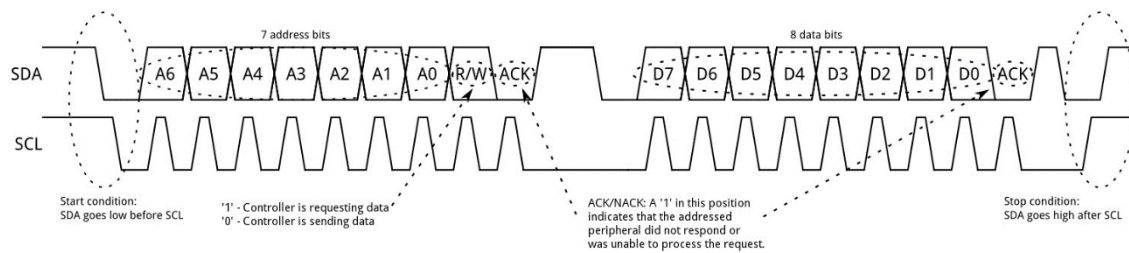
Na comunicação I2C, o byte é enviado em somente 1 canal. Este canal é chamado de **SDA** e está ligado em paralelo com todos os dispositivos da rede, ou seja, todos os dispositivos receberão o mesmo byte. Ainda, os dois canais devem possuir resistores de Pull-Up conectados, geralmente de 2.2K, (os valores são padronizados) para elevar o estado lógico do pino do master quando ele estiver configurado em alta impedância (entrada).



Fonte: <https://www.analog.com/en/technical-articles/i2c-primer-what-is-i2c-part-1.html>

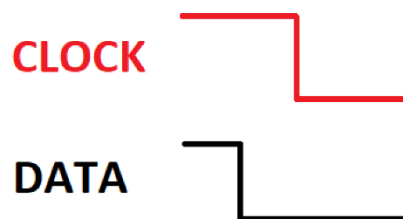
Para que o slave seja capaz de ler os bits corretamente, o clock é utilizado para sincronizar esta leitura. Sempre quando o clock estiver em alta, uma leitura é realizada. Se o clock subir com SDA em alta é lido 1, se o clock subir com SDA em baixa, é lido 0.

Além dos bytes, existem alguns bits que possuem funções específicas no protocolo como: início da comunicação; fim da comunicação; reconhecimento da comunicação. Estes bits serão detalhados melhor a frente. Veja abaixo o processo de comunicação completo:



Bits de sinalização

START bit → Indica o início da comunicação para o Slave. O Slave reconhece o bit de start quando recebe um sinal de Clock (SCL) em alta e um sinal de data (SDA) caindo. A partir daí, ele estará pronto para receber o primeiro byte com o endereçamento e o bit R/W.



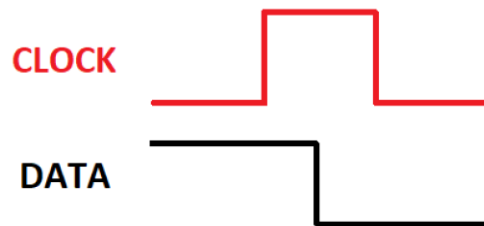
STOP bit → Indica o fim da comunicação para o Slave. Clock (SCL) estando em alta e data (SDA) subir:



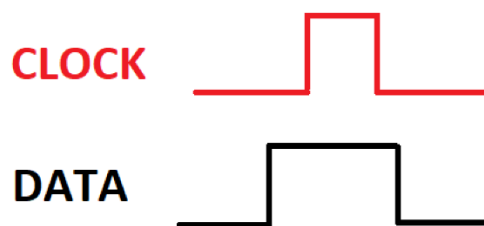
ACK bit → bit de confirmação. É o bit que indica ao Master que o Slave recebeu o dado e está pronto para receber mais um. Este bit está no 9º pulso de clock (SCL) da comunicação. Enquanto data (SDA) está em baixa, o clock (SCL) realiza um pulso completo.



Repeated start bit → Uma condição de stop seguida de outra de start. É utilizado no modo combinado no Repeated Start para leitura de um endereço específico de memória:

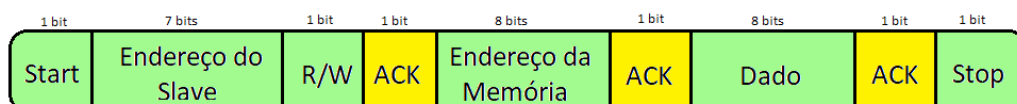


NACK bit → Negação, finalização. Este bit indica o fim do processo de comunicação, quando o slave recebe este bit ele para de incrementar a sua memória interina. Um pulso no SCL quando SDA estiver alto:



Exemplo - Escrita em uma memória EEPROM

O processo começa com um bit de start enviado pelo Master seguido pelo endereço do Slave + bit de Write. O Slave retorna ACK para dizer ao Master que entendeu e espera um endereço de memória. O Master envia este endereço de memória, o Slave retorna outro ACK, e o Master passa a enviar os dados. A comunicação continua até que o Master envie o bit de Stop.

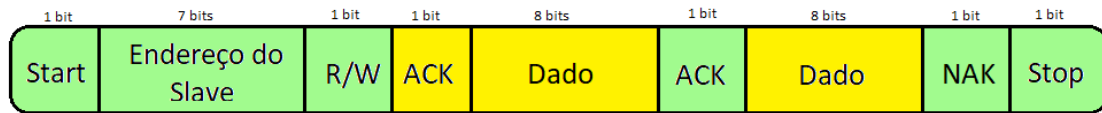


● Enviado pelo master

● Enviado pelo slave

Exemplo - Leitura de uma memória EEPROM

O modo de leitura padrão ocorre de forma semelhante ao de escrita. A diferença é que agora, após o primeiro ACK, é o Slave que passa a enviar dados e o Master responde com ACK. A cada ACK enviado pelo Master, o Slave incrementa seu endereço de memória e envia o byte correspondente. Este processo se repete até que o Master envie o NAK, informando que não quer mais receber dados.

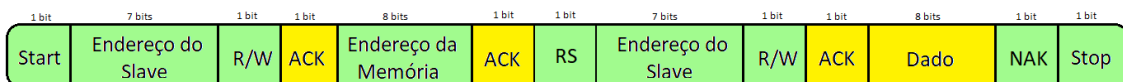


● Enviado pelo master

● Enviado pelo slave

O problema do modo acima é que o escravo não espera receber um endereço de memória, como ocorre no modo escrita, ele retorna os dados a partir do primeiro endereço. Para contornar esta situação, que em muitos casos pode ser um problema, existe o modo combinado.

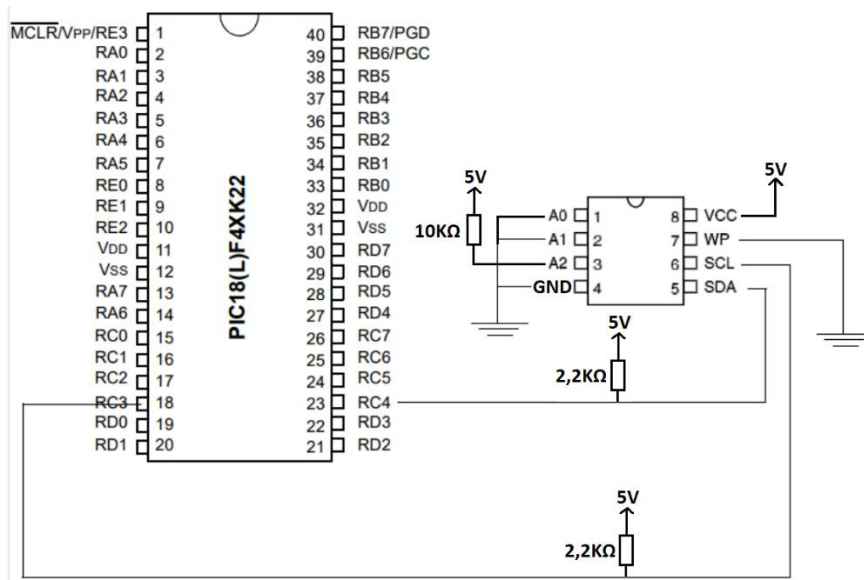
O modo combinado é chamado assim pois combina os processos de escrita e leitura. A comunicação inicia como um processo de escrita padrão, mas sofre um **Restart** logo após o envio do endereço de memória. O endereço fica gravado no Slave, mas ele não sabe mais se será escrito ou lido. A partir deste ponto, o processo de leitura pode ocorrer normalmente.



● Enviado pelo master

● Enviado pelo slave

Esquemático da conexão



Com esta memória, o próprio projetista pode definir qual será o endereço dela (pelo menos em parte). No caso de comunicação entre dois microcontroladores, você pode fazer este endereçamento via software, mas como não é possível gravar software dentro de uma EEPROM, criamos este endereçamento com os pinos A0, A1 e A2. Dos 8 bits de

endereçamento, os 4 bits mais significativos vem pré-definidos de fábrica como 1010, o pino menos significativo é quem indica modo Read(1) ou Write(0) R/W. Os últimos três são os pinos físicos A0, A1 e A2. O formato do endereçamento fica assim:



Mas para o modelo FT24C08A específico, somente o A2 funciona como endereçamento, enquanto os pinos restantes não são utilizados. No software, você deve utilizar:

- Escrita → 10101000;
- Leitura → 10101001;

Registrador SSPxSTAT

Registrador geral responsável por descrever o estado do MSSP:

- Bit 7 **SMP** → Bit de controle do Slew Rate. O slew rate é um parâmetro elétrico que descreve a taxa de variação de uma tensão ou corrente em um circuito em relação ao tempo.
- Bit 6 **CKE** → No modo I2C, habilita a compatibilidade com o SMBus. O SMBus é um protocolo de comunicação baseado no I2C mais confiável, porém mais lento.
- Bit 5 **D/A** → Flag que indica se o último byte recebido foi dado ou endereço.
- Bit 4 **P** → Flag de detecção do bit de stop.
- Bit 3 **S** → Flag de detecção do bit de start.
- Bit 2 **R/W** → Bit de leitura ou escrita. No modo master, indica se uma transmissão está em execução
- Bit 1 **UA** → Indica se o usuário precisa atualizar o endereço do registrador SSPxADD.
- Bit 0 **BF** → O registrador SSPBUF está cheio, tanto na escrita quanto no recebimento.

Registrador SSPxCON1

- Bit 7 **WCOL** → Colisão de escrita. Indica que houve uma tentativa de escrita no registrador SSPxBUF quando este estava cheio.
- Bit 6 **SSPOV** → Colisão de recebimento. Indica que uma informação chegou ao SSPxBUF enquanto este estava cheio.
- Bit 5 **SSPEN** → Habilita o módulo MSSP.
- Bit 4 **CKP** → O slave pausa o clock da rede para que possa processar o dado recebido.
- Bit <3:0> **SSPxM** → Seleciona o modo de operação do MSSP.

Registrador SSPxCON2

- Bit 7 **GCEN** → Gera uma interrupção quando uma chamada geral (0x00) é recebida.
- Bit 6 **ACKSTAT** → Indica recebimento do bit ACK.
- Bit 5 **ACKDT** → Indica se o bit é ACK ou NACK.
- Bit 4 **ACKEN** → Transmite o bit ACK na rede.
- Bit 3 **RCEN** → Habilita o modo de recepção do I2C.
- Bit 2 **PEN** → Transmite o bit de stop na rede.
- Bit 1 **RSEN** → Transmite o bit de Repeated Start na rede.
- Bit 0 **SEN** → Envia o bit de start (master) - Habilita o modo Clock stretching (slave).

Registrador SSPADD

No modo **Master**, este registrador armazena o BaudRate da comunicação:

```
SSP1ADD = (_XTAL_FREQ/1000000/4) - 1; // baudrate para 100KHz
```

No modo **slave**, este registrador indica o endereço do microcontrolador em uma rede I2C.

Interrupções do MSSP

Os bit de interrupção são os mesmos para I2C e para SPI. No entanto, a ocorrência deles se dá de forma diferente. No modo I2C, a sinalização da interrupção (SSPIF) se dá pelos seguintes motivos:

- Condição de Início
- Condição de Parada
- Transmissão ou recepção completada
- Transmissão do Acknowledge
- Condição de Repeat Start