



I2C bus

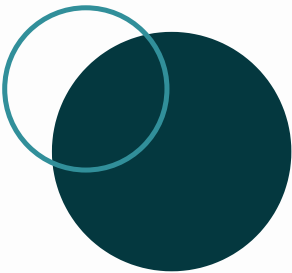
Inter-Integrated Circuit

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería Electrónica
Diseño de Sistemas Digitales
Ing. Juan Scott Chaves M.S.E.E

I2C

CARACTERÍSTICAS

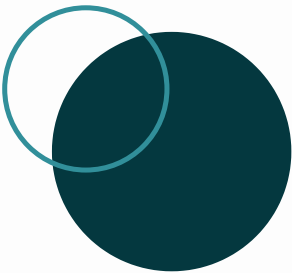
- The I2C es un bus multi-master bus.
- The I2C bus físicamente consiste de 2 cables activos y una conexión a GND.
- Los cables activos se llaman SDA y SCL y son bi-direccionales.
- SDA es la Serial Data line, y SCL es la Serial Clock line.
- Cada dispositivo conectado al bus tiene una dirección única.



I2C

FUNCIONAMIENTO

1. El MCU dará la condición de START. Esta actúa como una señal de "ATENCIÓN" para todos los dispositivos conectados.
2. Todos los ICs conectados al Bus van a "ESCUCHAR" del bus los datos que envió.
3. El MCU envía la DIRECCIÓN del dispositivo que quiere accesar, con un indicador de si el acceso es una operación de READ o WRITE.



I2C

FUNCIONAMIENTO

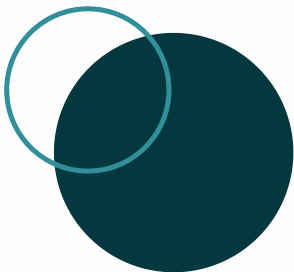
4. Cuando reciben la dirección los dispositivos la comparan con la propia:

Si no hace match: El IC espera que el bus se desocupe.

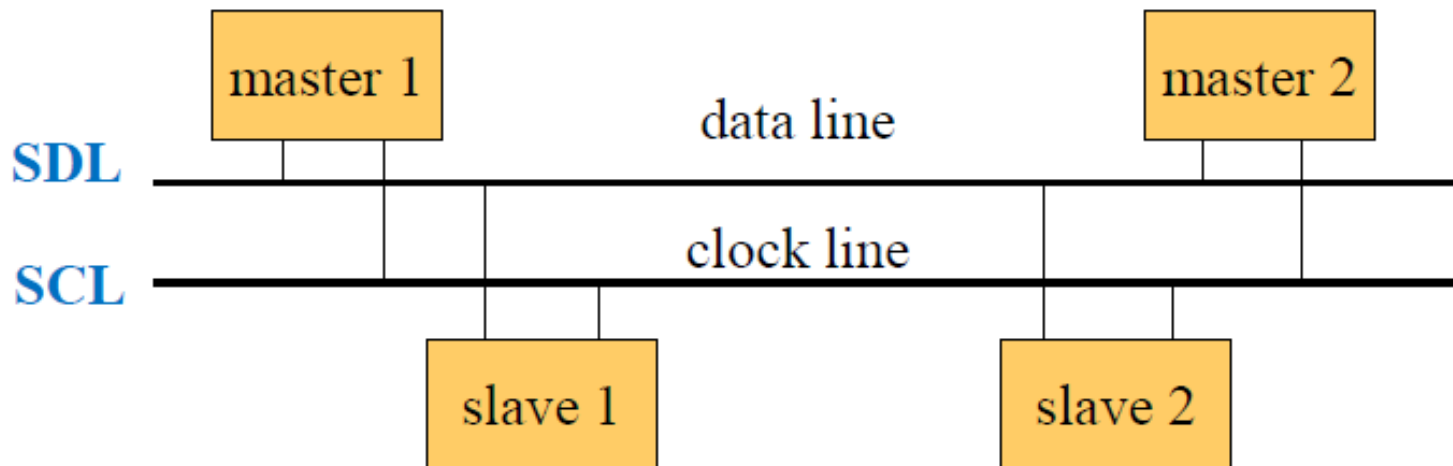
Si hace match: El IC va a producir una respuesta llamada Señal de ACKNOWLEDGE.

5. Cuando el MCU recibe el ACKNOWLEDGE, puede empezar a transmitir o recibir datos.

6. Cuando termina el proceso el MCU envía la condición de STOP.



I2C CAPA FÍSICA





I2C LINK LAYER

DIRECCIÓN DE LLAMADA GENERAL

(0000000) para broadcast.

CADA DISPOSITIVO TIENE UNA DIRECCIÓN

- Configurada por el dispositivo y/o el diseñador del sistema.
- 7 bits en standard (10 bits en extension).
- El bit 8 de las señales de dirección lee (1) o escribe (0).

TRANSACCIÓN DE BUS = SERIE DE TRANSMISIONES DE UN BYTE

El Master envía la dirección del Slave seguido de datos hacia o desde el Slave. Bueno para la programación de "inserción de datos" (data-push).



SEÑALIZACIÓN I2C

EXCEPCIONES:

SDA 1→ 0 mientras SCL = 1 señala la condición de INICIO

SDA 0→ 1 mientras SCL = 1 señala la condición de PARADA

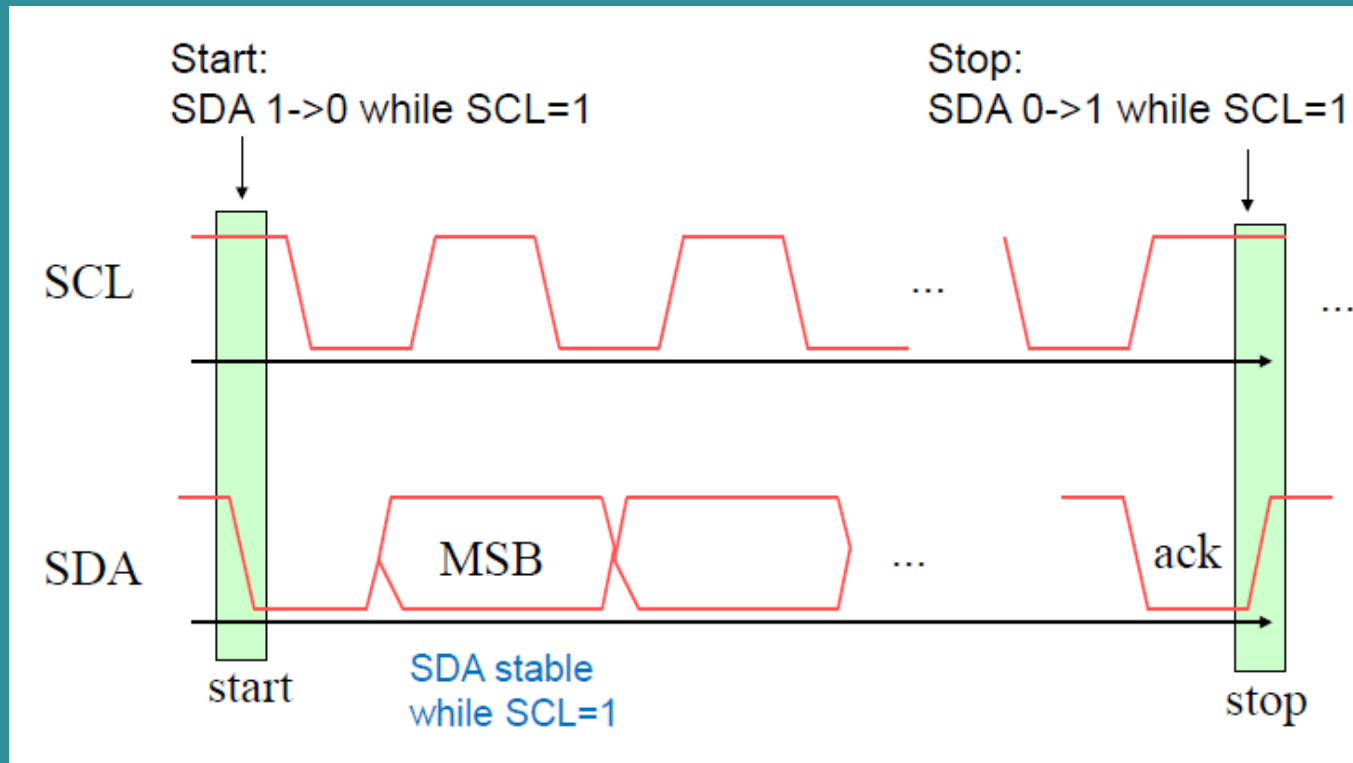
BUS = CONFIGURACIÓN "WIRED-AND"

- Colector abierto / controladores de drenaje en SDA y SCL
- La resistencia tira del bus a la lógica 1.
- Cualquier remitente puede bajar el bus a 0, incluso si otros remitentes intentan conducir el bus a 1.
- El remitente "libera" SDA al deshabilitar su controlador, permitiendo que SDA sea llevado a la lógica 1

LOS DATOS SOBRE SDA DEBEN SER ESTABLES MIENTRAS SCL HIGH

- Los datos en SDA se muestrean mientras que SCL es HIGH
- SDA puede cambiar solo mientras SCL LOW

I2C DATA FORMAT

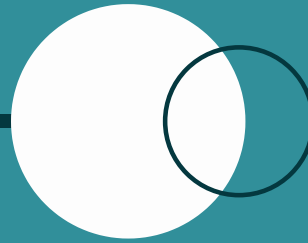


Sincronización de reloj

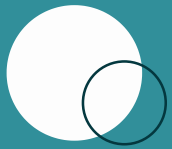
Master genera su propio reloj en SCL durante la transferencia de datos

La sincronización del reloj utiliza "wired-and"

- Conducción "low pull" tira SCL LOW, reiniciando todos los contadores de reloj
- SCL permanece LOW mientras cualquier conductor lo baja
- SCL low time= reloj más lento (otros en estado de espera)
- Primer dispositivo en terminar
- High State pulls tira SCL LOW



I2C modos de operación



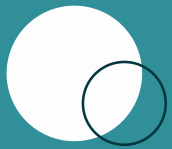
MASTER-SENDER

El módulo emite INICIO y DIRECCIÓN, y luego transmite datos al dispositivo esclavo direccionado



SLAVE-SENDER

Otro Master emite INICIO y la DIRECCIÓN de este módulo, que luego envía datos al maestro



MASTER-RECEIVER

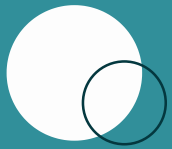
El módulo emite INICIO y DIRECCION, y recibe datos del dispositivo esclavo direccionado



SLAVE-RECEIVER

Otro Master emite INICIO y la DIRECCIÓN de este módulo, que luego recibe datos del maestro.

Arbitraje de bus I2C



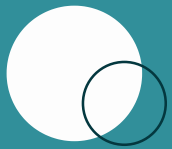
**EL MASTER PUEDE
COMENZAR A ENVIAR SI
EL BUS ESTÁ LIBRE**

2 o más pueden generar
START al mismo tiempo



**EL REMITENTE DEJA DE
TRANSMITIR SI SE PIERDE
EL ARBITRAJE**

Transmite 1 y escuche 0 en
SDA.



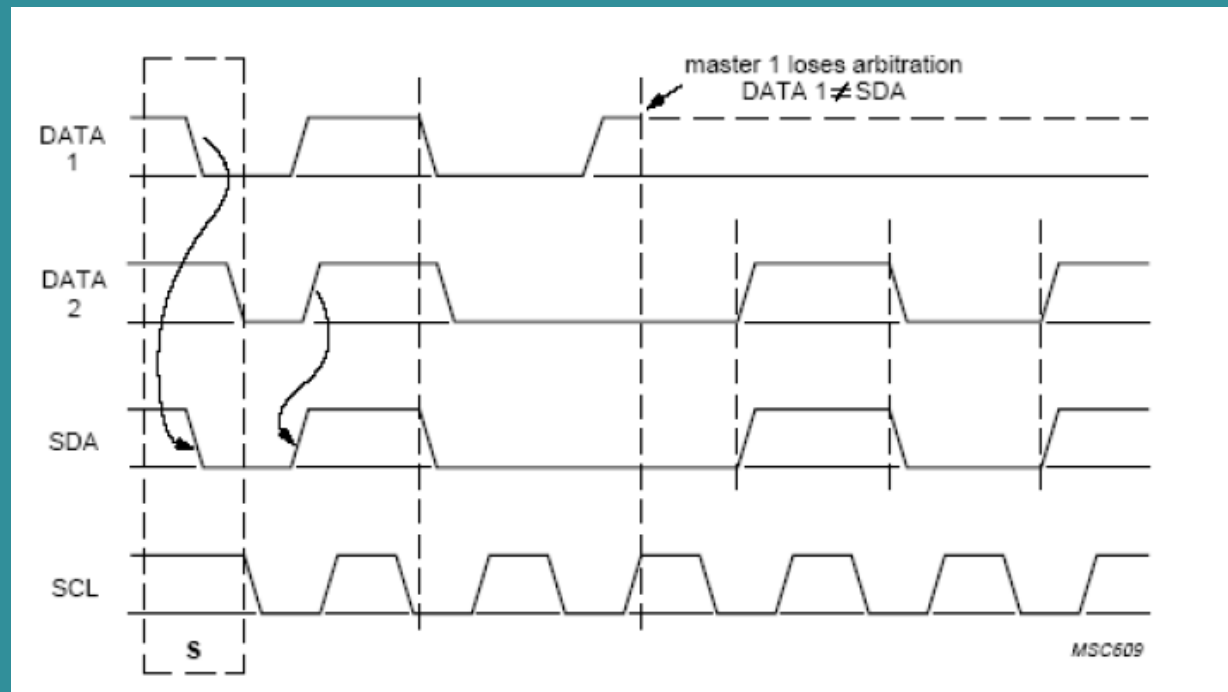
**EL REMITENTE ESCUCHA
MIENTRAS SE ENVÍA**

Prueba SDA mientras SCL HIGH



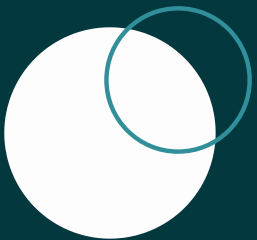
**EL ARBITRAJE CONTINÚA
A TRAVÉS DE
DIRECCIONES Y ACKBITS,
Y LUEGO DATOS Y
ACKBITS SI ES NECESARIO**

EJEMPLO DE ARBITRAJE



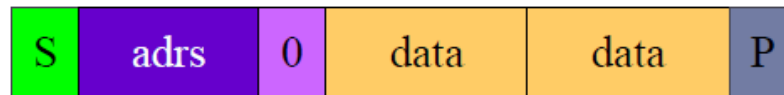
TRANSFERENCIA DE DATOS

- Enviar byte de 8 bits (MSB primero)
- Cada byte seguido de bit de reconocimiento (acknowledge)
 - Master lanza la línea SDA (alta) durante el reloj
 - Slave debe tirar SDA low para reconocimiento adecuado
 - Si SDA se dejó High, el master puede detenerse o repetir el inicio
 - Si el master está recibiendo del slave, el slave libera SDA para permitir que el master tire de SDA low por un golpe
- El slave puede mantener SCL low para forzar el tiempo de espera entre bytes



Transmisiones I2C (no se muestran los ACK)

multi-byte write



read from slave



write, then read



Re-start without giving up the bus

FIN

