

[Bus I2C] Arbitraje

- Este proceso solo se da en los sistemas multimasters. Si varios masters intentan tomar el control del bus a la vez, un reloj resultante sincronizado va a permitir un arbitraje que dará el control a un solo master pues no puede haber más de uno a la vez gobernando el bus.
- Un master puede iniciar una transmisión solo si el bus esta libre. Dos o mas master pueden generar una condición de Inicio en el bus al mismo tiempo lo que da como resultado una condición de Inicio general en Bus.
- El sistema de arbitraje actúa sobre la línea de datos SDA, mientras la línea SCL esta a nivel 1, de una manera tal que el master que intente transmitir un nivel 1, pierde el arbitraje (prioridad) sobre otro master que al mismo tiempo envía un nivel 0 a SDA. Por la constitución de colector abierto podemos comprender por qué es así.
- El proceso podría ser el siguiente:
 - Cada Master debe comprobar si el bit de datos que transmite junto a su pulso de reloj, coincide con el nivel lógico en la línea de datos SDA. Por ello, cuando un master pone un 1 en SDA y, tras una espera prudencial para que dicha transición tenga lugar, lee dicha línea y ve que hay un 0 es que se está produciendo una emisión simultánea de otro master, por lo que debe dejar de transmitir y:
 - a) Si el byte transmitido era el de dirección, **seguirá escuchando** al otro master por si dicha dirección es la suya y debe colocarse como slave. Si un master pierde la prioridad durante el direccionamiento (envío byte 1º tras START) debe pasar inmediatamente a slave receptor, pues puede ser él quien esté siendo direccionado.
 - b) Si el byte transmitido era de datos deberá liberar (desbloquear) las líneas del bus (SDA=SCL=1) para permitir que el otro master continúe su transferencia al slave direccionado. Este segundo caso ocurre cuando las direcciones comparadas de los dos masters son idénticas (es decir los dos quieren comunicar con el mismo slave), continuando el arbitraje con los bits de datos.
 - En ambos casos se ve que la transferencia de información no es retardada por el arbitraje, ya que los datos transferidos inicialmente por el master ganador no se pierden ni hay que repetirlos. Un master que pierde el arbitraje puede generar pulsos de reloj hasta el fin de byte en el cual él pierde el arbitraje.
- Como se ve, el proceso de arbitraje continúa hasta que se dé la circunstancia de control del bus por uno de los master. Especial atención debe ponerse si durante una transferencia de datos el procedimiento de arbitraje está todavía en proceso justo en el momento en el que se envía al bus una condición de Stop. Es posible que esta situación pueda ocurrir, en este caso el master afectado debe mandar códigos de Inicio o Stop.
- En la figura adjunta se ve el arbitraje entre dos master, aunque pueden estar involucrados más. En el momento en que hay una diferencia entre el nivel interno de la línea de datos del master DATA1 y el actual nivel de la línea de datos SDA, su salida de datos es interrumpida, lo cual significa que un nivel 0 esta dominando en el Bus. Esto no afecta los datos transferidos inicialmente por el master ganador.



- En el momento que un master toma el control, solo este master toma las decisiones y genera los códigos de dirección, no existen master centrales, ni existen órdenes prioritarias en el bus.

From:
<https://www.euloxio.myds.me/dokuwiki/> - **Euloxio wiki**

Permanent link:
https://www.euloxio.myds.me/dokuwiki/doku.php/doc:tec:lab:bus_i2c:0_arbitraje:inicio?rev=1774719262

Last update: **2026/03/28 18:34**

