I2C

Friday, October 22, 2021 18:52

- Sistema de comunicación síncrona desarrollado como protocolo de comunicación entre distintos circuitos inter-integrados en una motherboard
  - Las placas tienen muchos circuitos y se dificultaba la comunicaciones entre los mismos por medio de muchos buses, por lo que se desarrolló este protocolo
- Reemplazo del RS232
- Sólo se conforma por dos líneas de comunicación
  - Una envía la señal del clock para lograr sincronía (línea SCL)
  - La otra (bidireccional) transmite información (línea SDA)
    - Estas líneas requieren resistencias de pull - up
    - Estas líneas siempre están en estado lógico alto (1)
    - Como siempre están en uno, los dispositivos conectados sólo podrán mandar ceros por el bus (los unos se hacen automáticamente por las resistencias de pull - up al "soltar" la línea
- Frecuencia de transmisión de 100[KHz] a 400[KHz] (las nuevas versiones de I2C llegan hasta los 5[MHz])
  - Depende del fabricante y el dispositivo
Categoría	Frecuencia de transmisión máxima	Dirección de bus
Standard mode (SM)	100[KHz]	Bidireccional
Fast mode (FM)	400[KHz]	Bidireccional
Fast mode plus (FM+)	1[MHz]	Bidireccional
High speed mode (HSM)	3.4[MHz]	Bidireccional
Ultra fast mode (UFM)	5[MHz]	Unidireccional
- Las transmisiones son de 1[B] a la vez (8[b])
- Soporta de 3.3[V] a 5[V]
- Las resistencias de pull - up van de 1[KΩ] a 47[KΩ]
  - El valor preferido es de 10[KΩ]
- Para organizar las transmisiones en un bus bidireccional, se usa una jerarquía master / slave (relación 1:N)
  - Master:
    - Maneja el clock
    - Comienza la transmisión de datos
    - Puede haber uno o más de uno
  - Slave:
    - "Oyen" cuando el master lo solicita
    - "Hablan" cuando el master lo permite
    - Debe haber al menos uno
    - Cada slave tiene un ID único de 7 ó 10 bits (especificado por el fabricante)
      - Si el ID es de 7 bits, podemos tener hasta 128 dispositivos conectados para comunicarse entre sí
      - Si el ID es de 10 bits, podemos tener hasta 1024 dispositivos conectados para comunicarse entre sí
- Las transferencias por la línea SDA se conforman de:
  - Datos (8[b])
  - Bit que indica el modo de lectura / escritura
  - Bit de reconocimiento que indica si se recibió / terminó de enviar la información transmitida (dependiendo de si es TX o RX)
- La secuencia start / stop generalmente sigue esta serie de pasos:
  - Master envía la señal de que va a transmitir algo
    - Cuando el clock (SCL) está en alto, el master baja la línea de SDA
  - Master termina la transmisión de datos cuando levanta la línea de SDA cuando SCL está en alto
    - En este punto los slaves pueden dejar de "escuchar" y siguen haciendo sus tareas
  - En resumen:
    - Cuando SCL está en alto y SDA va de alto a bajo, tenemos un start
    - Cuando SCL está en alto y SDA va de bajo a alto, tenemos un stop
    - Cualquier otra combinación de estados entre SCL y SDA no se considera como señal de sincronización para la transmisión
- Cuando se completa una transmisión, el receptor envía una señal de reconocimiento (acknowledgement - ACK)
- La secuencia de señalización de reconocimiento generalmente sigue esta serie de pasos:
  - TX transmite 1[B] durante 8 ciclos de reloj y libera la línea SDA para recibir el pulso de ACK
  - RX debe enviar un pulso de ACK luego de cada byte recibido exitosamente para notificar la correcta recepción
  - Si TX no recibe el pulso de ACK de parte de RX, TX decide si reenviar la secuencia o no
- Finalmente, la secuencia completa de comunicación I2C entre TX y RX es como sigue:
  - Se genera la señal de start entre SCL y SDA
  - TX envía el byte de control:
    - TX envía el ID del slave al que se quiere comunicar (7 ó 10 bits)
    - TX envía el bit de lectura / escritura (R(1) / W(0)) dependiendo de si master quiere leer datos del slave o si quiere enviarle información
  - Se genera la señal de ACK (cada 9 bits se envía una señal de ACK)
    - El dispositivo RX (master o slave) es responsable de generar este bit de ACK (0)
    - Se puede generar un bit de NACK (1) para indicar el final de la comunicación
    - Si ACK viene de RX, indica que el byte se recibió correctamente
  - TX envía el byte (o bytes) de información a transmitir a RX
  - Se genera la señal de stop entre SCL y SDA