Comunicacions

GND Connectat

Pq. tots els disp. de la comunicació tinguin mateixa ref del qual sig. Vss. hem de connectar tots els GND. ▽▽ IMPO

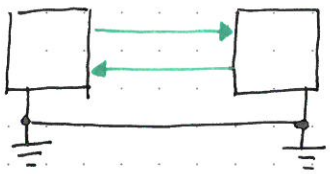
Classificació

Asíncron / Síncron

Asíncron

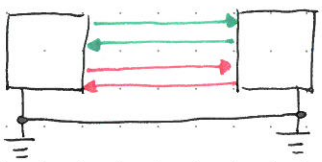
Cada disp. té el seu propi clock.

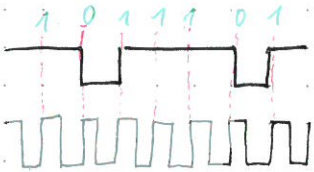
Han d'estar a mateixa freq per no malinterpretar.



Síncron

Per cada cable de dades hi ha d'haver un cable de clock.

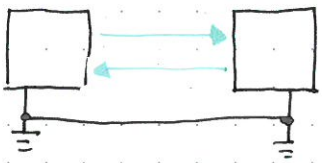




Full / Half Duplex

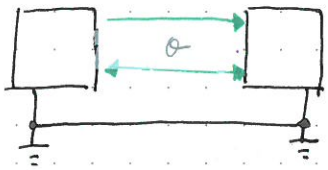
Full Duplex

Dades es poden enviar dos direccions en el mateix moment.



Half Duplex

Dades només es poden en 1 direcció o una direcció cada moment.



Tarden el doble

Port Sèrie

1 cable per direcció

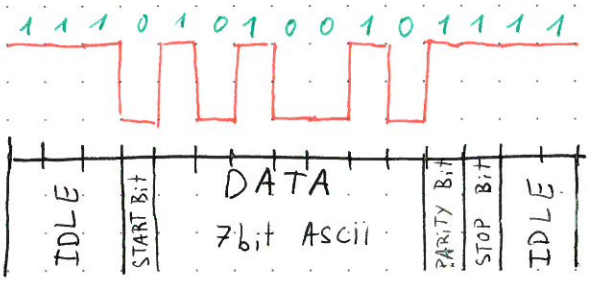
Només 2 disp

Qualsevol pot iniciar comunicació.

És Serial, Asíncron, Full-Duplex, Point-2-Point, Multimaster.

Idea principal:

- Per defecte entàs Idle ⇒ Emetent 1.
- Algo indica que ja has acabat o comences ⇒ Normalment 0
- Bits s'envien 1 a 1 en un temps determinat. (Prefixat)



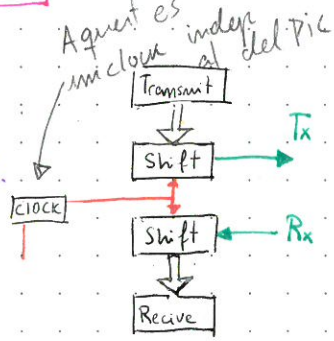
- START Bit: Inici del valor de Idle.
- PARITY Bit: (Opcional) Indica quantitat de 1s que hi ha → Parell / Senar
- STOP Bit: Permet agafar dades i guardar-ho.

Total = 1 (START) + n (Data) + 3 0, 1 & (PARITY) + 3 1, 1½, 2 & (STOP)

USART: Dispositiu que permet comunicar-se → Síncron. / Asíncron.

Permet enviar i rebre dades en ferm ús del protocol RS-232.

Shift-Register: Passen dades a format per enviar-se.



Errors Comuns

- Framming error: Error de sincronització de clocks.
  "Trobo un bit STOP on no deuria".
- Reciver overrun: CPU no llegeix dades que li envien o li envien massa.
  S'omple RCREGx (FIFO) i port sèrie deixa de funcionar.
- Parity Error: No es correcte el bit de paritat amb les dades rebudes.
  #Pot ser degut a soroll electromagnètic

Implementacions

- Bit-Banging: Tot per software (Delay, Start, ...) Van fent data & 01 i enviant i després data >> 1
- Hardware: Microchip té HW dedicat. 1 bit STOP, 8 bits DATA

REG Tx (Transmissor) TXEN

- TXREGx: Registre on es guarden dades que s'han d'enviar. // Pre càrrega pel TSR
- TXxIF: Indica si TXREGx està disp per ficar dades (1) o no (0) // Sobreescriure
- TSR: Shift Register. Envia dades pel cable.
- TRMT: Indica si s'estàn enviant dades (0) o ja ha acabat (1)
- Baud Rate Generator: Indica el clock que s'ha de fer servir
  - BRG16: Indica si necesitem 16 bits per indicar el n (1) o no (0).
  - SPBRG: Indicar el valor de velocitat de Baud. (El n en la fórmula)
  - BRGH: Volem que sigui High (1) o Low (0) speed.
  - SYNC: Seleccionar Asyncron (0) o Syncron (1).

REG Rx (Receptor) RCEN

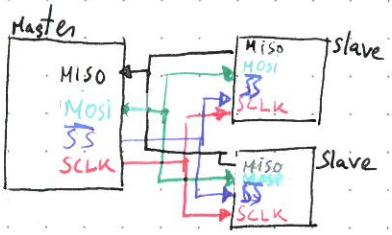
- FERR: Detectar Framming Error
- OERR: Overrun Error. ▽ Tot i que hi hagi OERR si que entram ISR ✓ El podem gestionar aquí dins
- RCREGx: FIFO de 2 pos on estan dades rebudes. Si arriba tercer ⇒ Overrun.
- RCxIF: Indica que hi ha dada a llegir (1) o no (0).

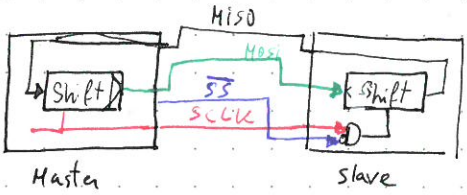
SPI (Serial Peripherial Interface)

És Serial, Síncron (Clock del Master), Full-Duplex, Multipoint, Master/Slave.

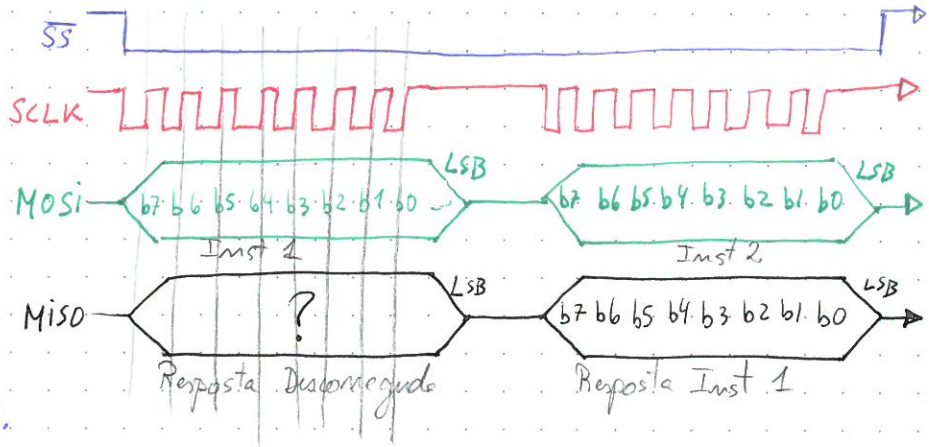
Connexions:

- MISO: Master Input
- MOSI: Master Output
- SCLK: Source Clock
- $\overline{SS}$: Slave Select





Hem de llegir el datasheet del Slave per veure quina inst. envien pel MOSI i el format de la resposta rebuda pel MISO.



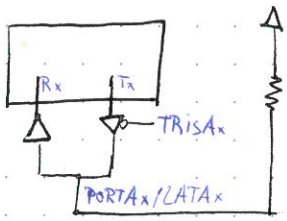
I2C (Inter Integrated Circuit)

És Serie, Síncron (Clock Del Master), Half-Duplex, Multipoint, Master/Slave.

Canal SDA (Dades) i SCL (Clock) que estan connectats en un Pull-Up (Sempre 1)

Procediment:

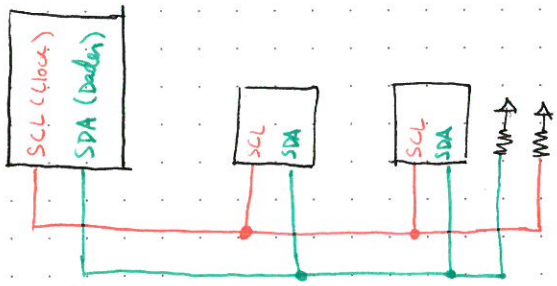
1. S'envia START BIT
2. S'envia @7b (Marcat fabricant)
3. Tipus d'operació R/W (1 bit)
4. Resposta ACK del Rx. ACK=0 ⇒ OK
5. Dades s'envien/reben (8b)
6. Resposta ACK del Tx.
7. S'envia STOP BIT.

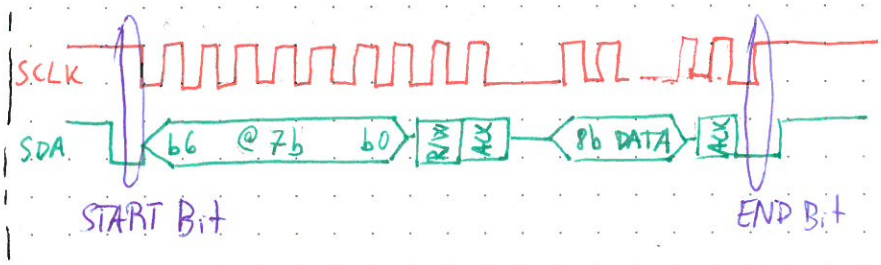


Obs: Quan M ha enviat dades ha de fixar-se a escoltar el ACK del Slave.

Aquest (Slave) fica a 0 el seu TRISx i envia en LATAx un 0 per fer que SDA caigui a 0V. (Resistència fa caigudes)

Master (que estava llegint amb TRISx=1) "rebé" del SDA un 0 i sig. que OK.





1-Wire

Serie, Asíncron, Half-Duplex, Multipoint, M/S. Només hi ha un cable i GND.

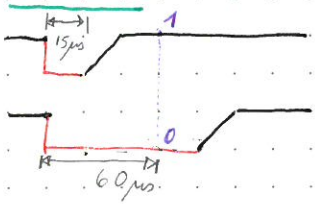
Idea Principal:

1. Host ha de descobrir si hi ha algú.
2. Device avisa de l'existència.
3. Host ho detecta.
4. A partir d'ara (Host) pot ↔ enviar / rebre dades.

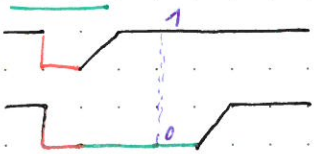
- Cada disp. té un ID únic.
- Dins dels disp. hi ha un Capacitador que permet "fer foto" del medi.
- Sempre Pull-Up.
- És lent i s'ha de fer Bit-Banging.

Per exemple. Anem a algo

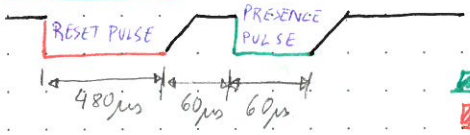
Enviar



Rebre

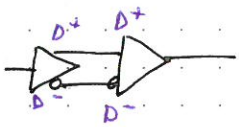


Descobrir

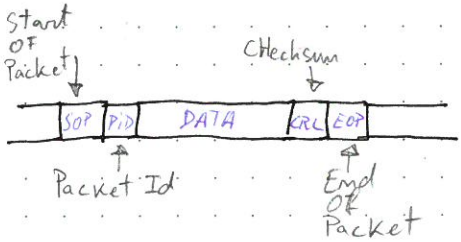


USB

Half-Duplex (1-2)
És Serial, Asíncron, Full-Duplex (3), M/S, Té bus diferencial



- Signaling: Defineix com representen físicament dades. Interpretacions, Velocitat, ...
- Codificació: → NRZI: El 0 fa switch level
  → Stuffed bit: Bit extra (0) que s'afegeix després del 6è 1.
- — Start: Per defecte $D^+ = 1 \Rightarrow D^+ = 0$ per iniciar.
- — Stop: Ficar $D^+ = D^- = 0$.
- Packing: Dades s'escriuen seguint estàndard



PID	Nom	Resum
Token	OUT IN SOF SETUP	Iniciar transmissió de dades
Data	DATA 0 DATA 1	Enviar dades s'ha d'anar intercanviant
Handshake	ACK NAK	Ha rebut OK No està dispo (Rx Tx)