

# C5 Les Bus série

## RS232, SPI et I2C

Yann DOUZE  
Polytech Sorbonne

# Les interfaces séries

- Liaison série asynchrone
- Liaison série synchrone
- Full Duplex
- Half Duplex
- Point à Point
- Multi-Point

Liaison asynchrone

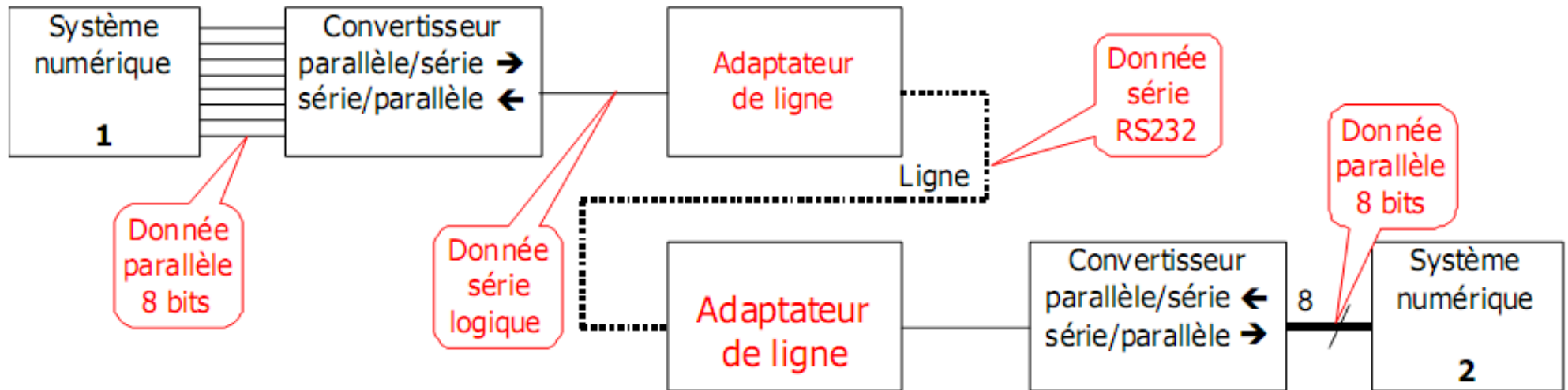
Full Duplex

2 fils (Tx /Rx)

Liaison point à point

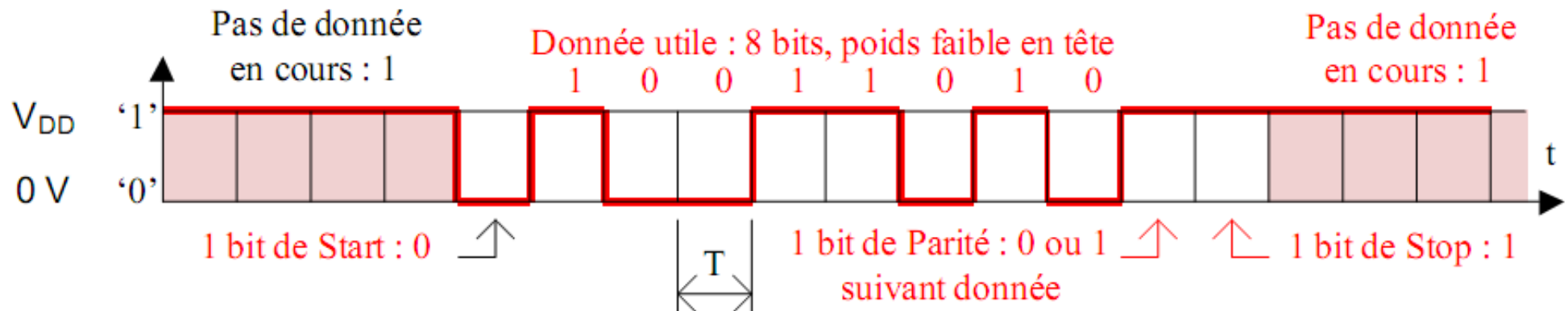
# **LIAISON RS232 ET RS485**

# RS232 : Organisation matérielle



# RS232 : Protocole

- un bit de Start,
- les 5 à 8 bits de données, poids faible en tête,
- éventuellement, un bit de vérification de Parité (paire ou impaire) qui permet de détecter des erreurs de transmission des 8 bits de donnée sur la ligne,
- un, un et demi, ou 2 bits de Stop.

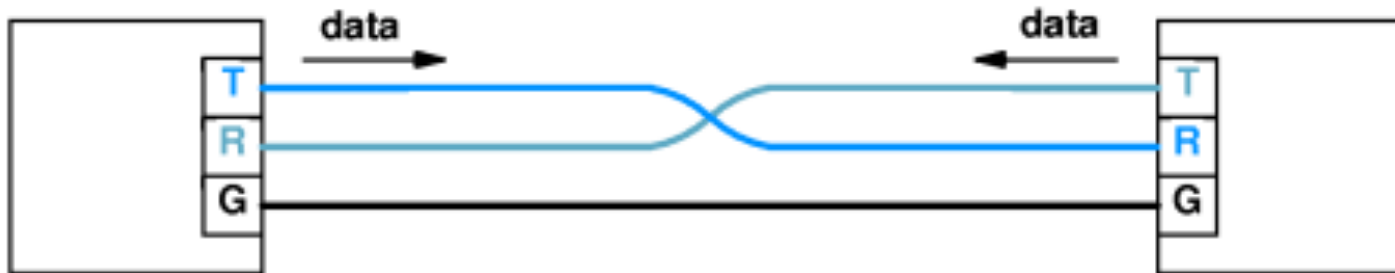


# Vitesse de transmission

- BAUDS : nombre de variations de fréquence (ou de phase) par seconde.
- Pour le RS232 -> BAUDS = BPS (Bits Par Seconde)
- Les principales vitesses de transmission normalisé :
  - 75, 110, 300, 600, 1200, 2400, 4800, **9600**, 14400, **19200**, 28800, 31200, 33600, 38400, **56000**, 57600, **115200**, 128000, 256000 bits/s

# Transmission Full Duplex

- Full Duplex : l'émission et la réception peuvent se faire en même temps.



# Les signaux

Tx	Transmit	conducteur d'émission des données
Rx	Receive	conducteur de réception des données
Gnd	Ground	conducteur de masse du signal

- Niveaux des signaux RS232 :

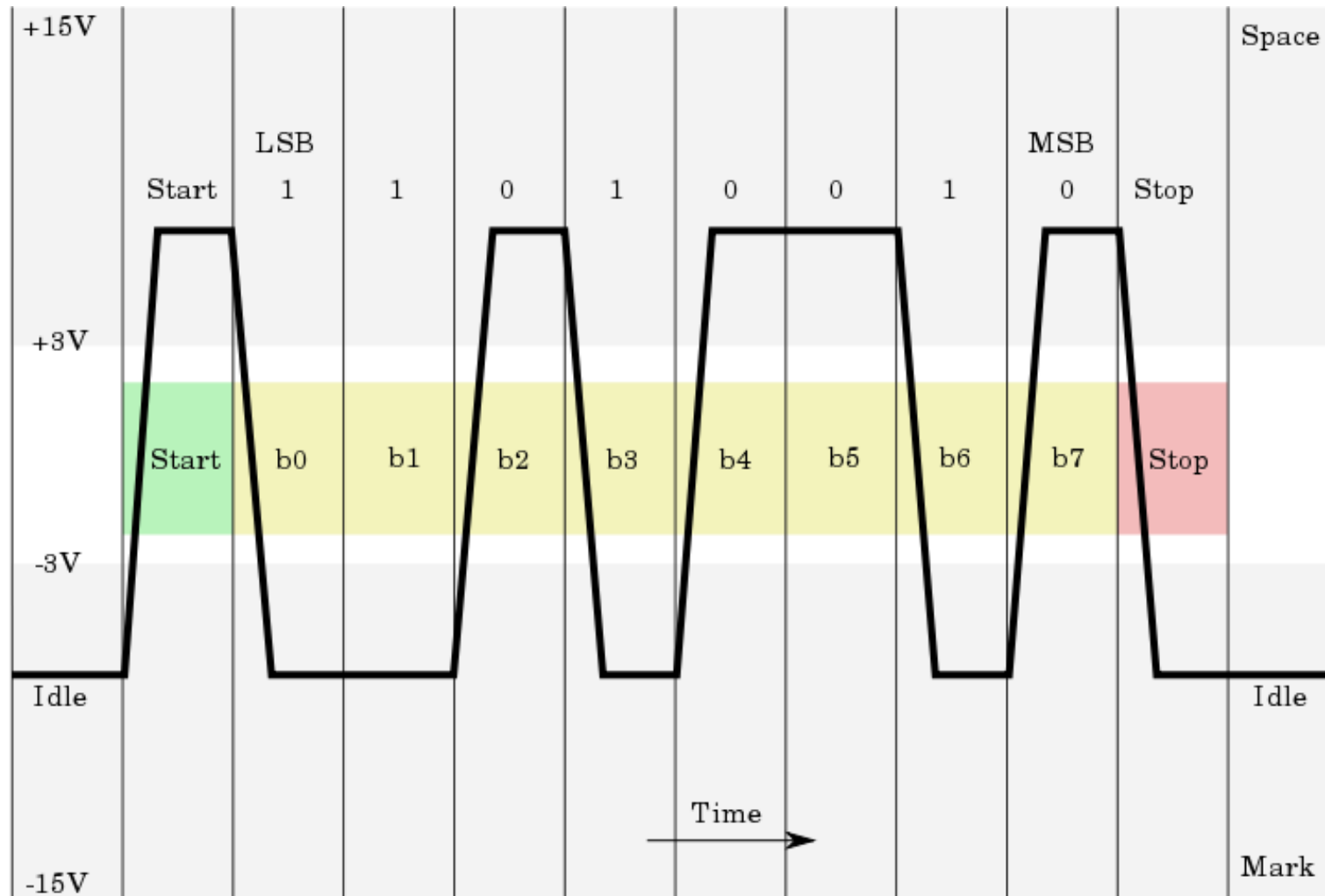
Niveau logique	Polarité	Intervalle de niveau électrique	Typique
'1'	Basse	entre -3V et -15 V	-12V
'0'	Haute	entre +3V et +15 V	+ 12V

**Versions de la norme :**

- RS232 → 48V
- RS232A → 25V
- RS232B → 12V
- RS232C → 5V

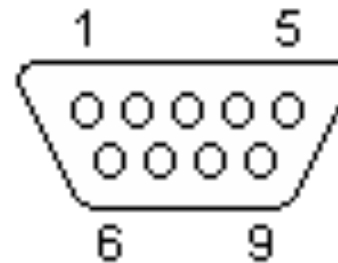


# Trace RS232 (obsolète)



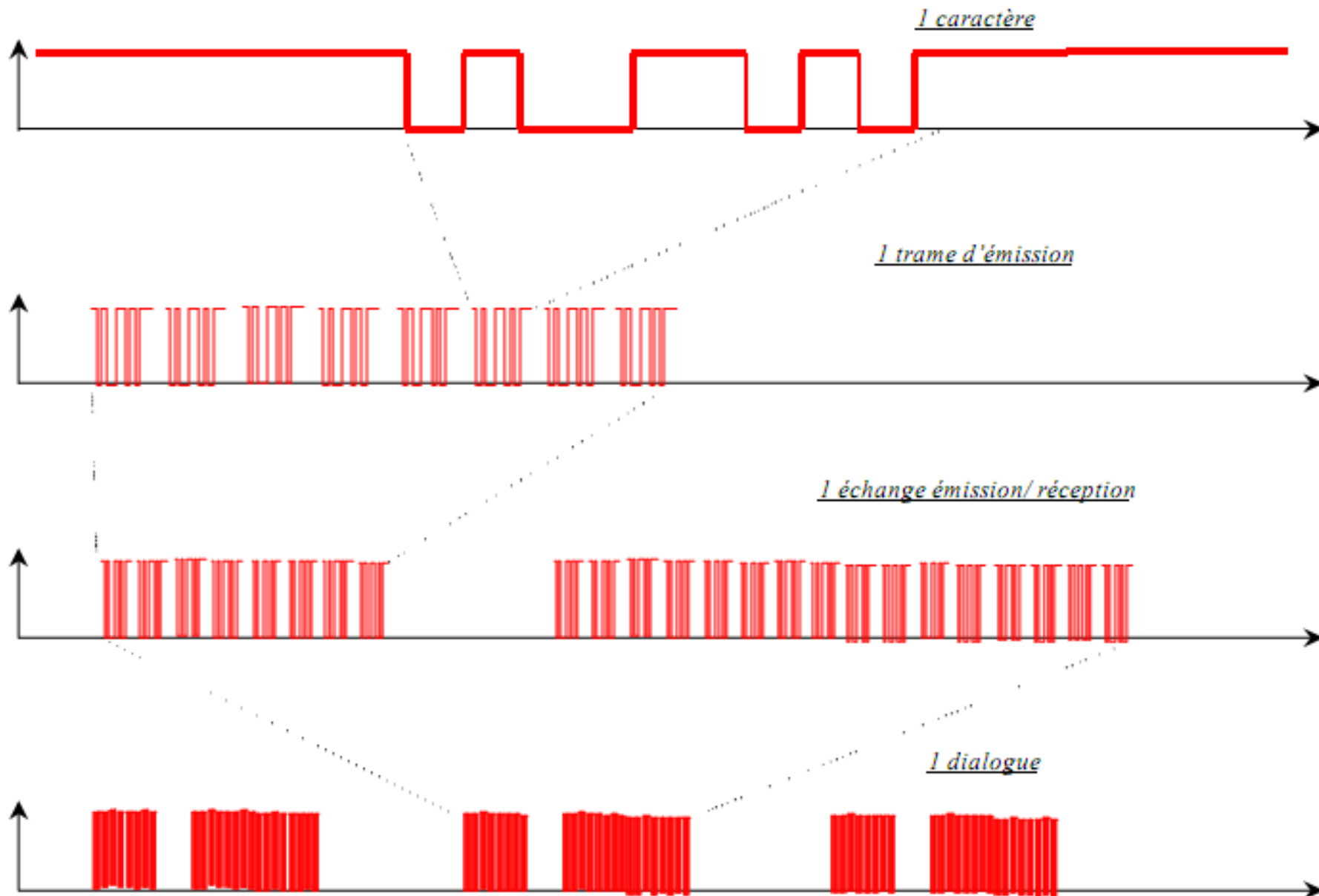
# Connectiques RS232 : DB9 (obsolète)

DB-9M	Function	Abbreviation
Pin #1	Data Carrier Detect	CD
Pin #2	Receive Data	RD or RX or RXD
Pin #3	Transmitted Data	TD or TX or TXD
Pin #4	Data Terminal Ready	DTR
Pin #5	Signal Ground	GND
Pin #6	Data Set Ready	DSR
Pin #7	Request To Send	RTS
Pin #8	Clear To Send	CTS
Pin #9	Ring Indicator	RI



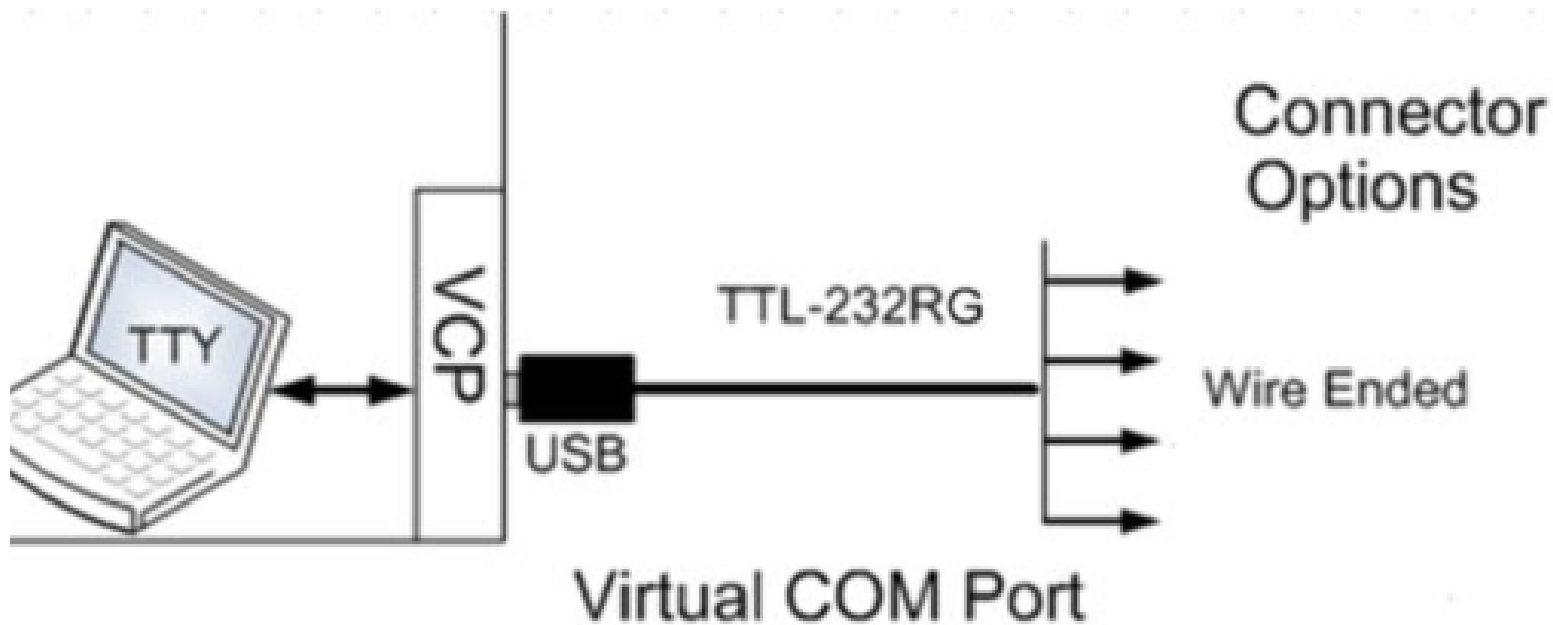
- Computer Terminal Equipment (CTE)
  - Tx sur la pin 2 et Rx sur la pin 3.
- Data Terminal Equipment (DTE)
  - Tx sur la pin 3 et Rx sur la pin 2.

# Communications



# Convertisseur série / USB

- Chip FTDI TTL-232RG



# Convertisseur USB-Série DFR0065



Bus Synchrone

2 signaux (SCL / SDA)

Half Duplex

Liaison Multipoint



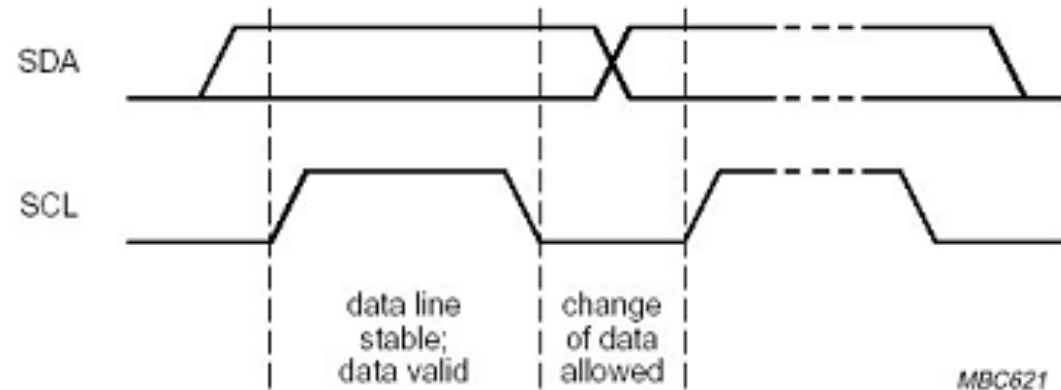
# **BUS I2C (INTER INTEGRATED CIRCUIT)**

# Bus I2C

- two-wired bus
- speeds:
  - 100 kbps (standard mode)
  - 400 kbps (fast mode)
  - 3.4 Mbps (high-speed mode)
- data transfers: serial, 8-bit oriented, bi-directional
- master/slave relationships with multi-master option (arbitration)
- master can operate as transmitter or receiver
- addressing: 7bit or 10bit unique addresses

# Wires and Signal

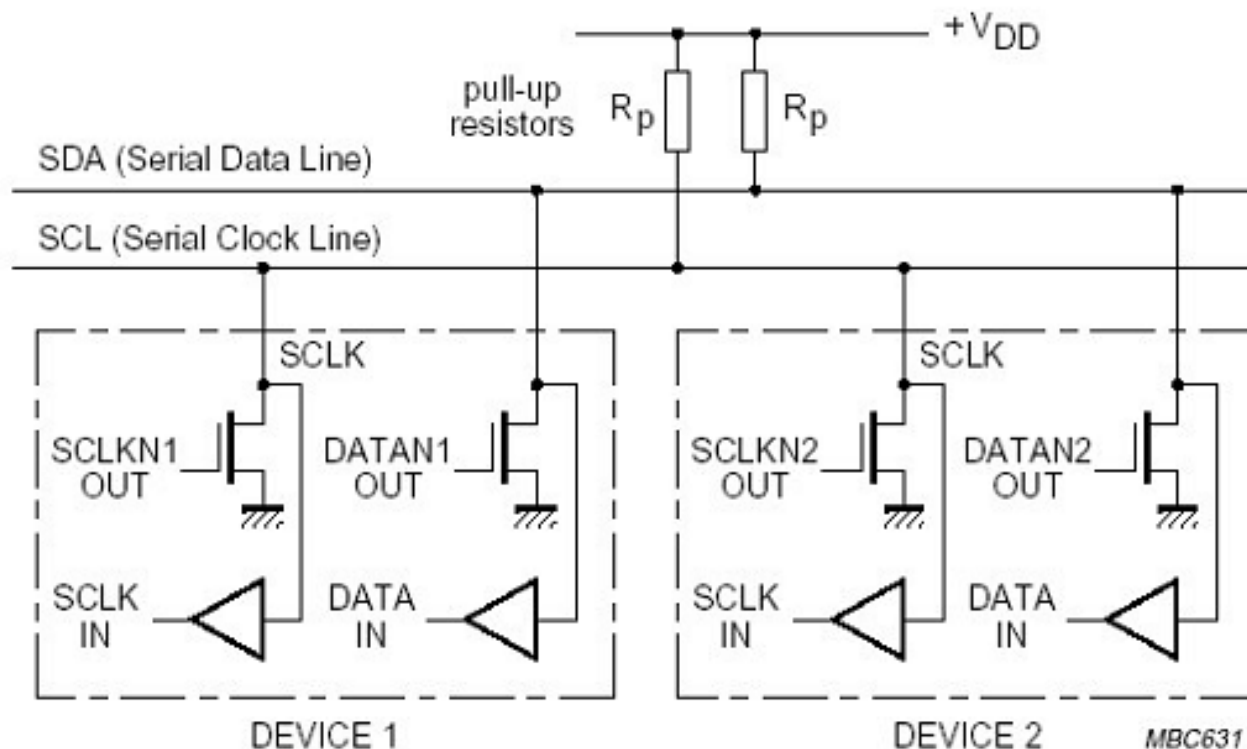
- two-wired bus
  - serial data line (SDA)
  - serial clock line (SCL)
- voltage levels
  - HIGH 1
  - LOW 0
- bit transfer (level triggered)
  - $SCL = 1 \rightarrow SDA = \text{valid data}$
  - one clock pulse per data bit
  - stable data during high clocks
  - data change during low clocks





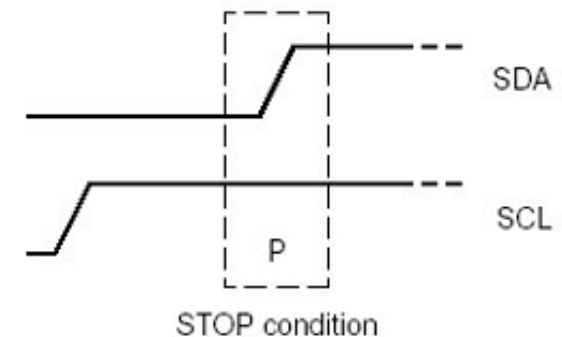
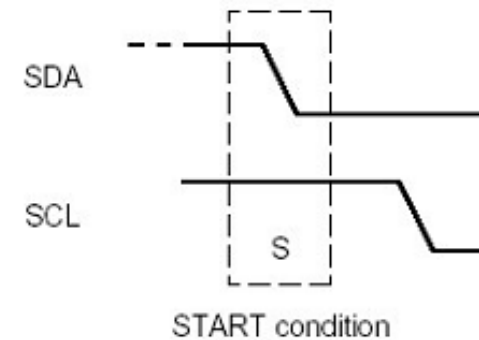
# Wired-AND connection

- bus is free  $\rightarrow$  SDA and SCL are high
  - by pull-up resistors
- device output is ANDed with signal on bus



# Frame

- start condition (S)
  - SDA 1  $\rightarrow$  0 transition when SCL = 1
- stop condition (P)
  - SDA 0  $\rightarrow$  1 transition when SCL = 1
- repeated start (Sr)
  - start is generated instead of stop
- bus state
  - busy ... after S and before next P
  - free ... after P and before next S



# Masters and Slaves

- Master device
  - controls the SCL
  - starts and stops data transfer
  - controls addressing of other devices
- Slave device
  - device addressed by master
- Transmitter/Receiver
  - master or slave
  - master-transmitter sends data to slave-receiver
  - master-receiver requires data from slave-transmitter

# Data Transfer

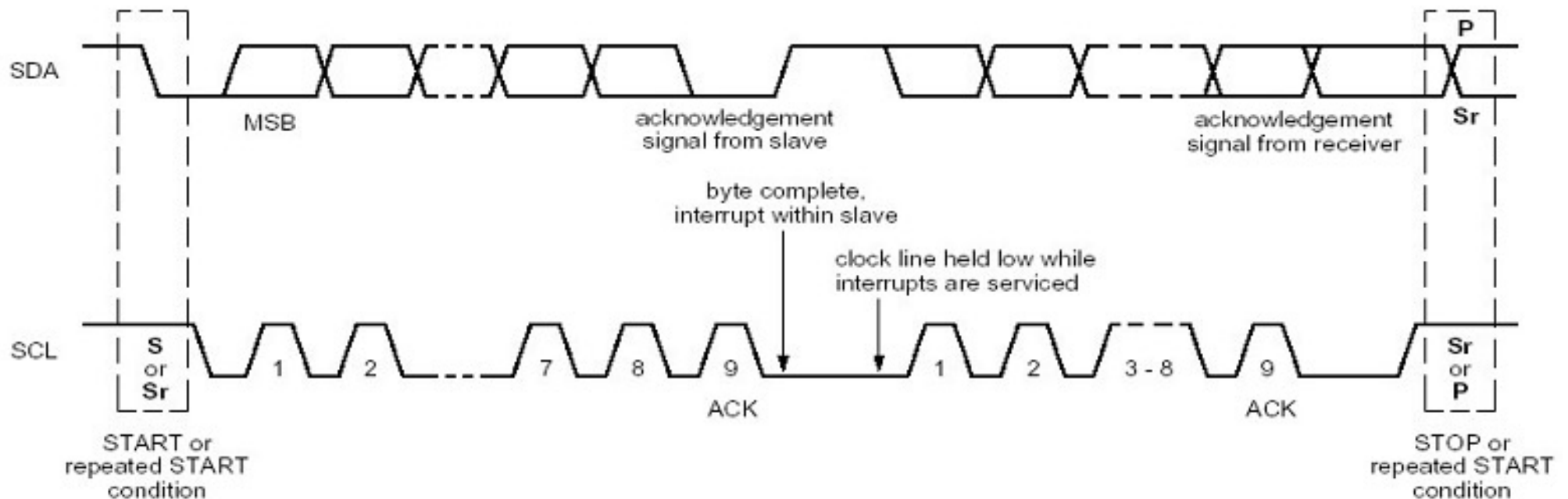
- data bits are transferred after start condition
- transmission is byte oriented
- byte = 8 bits + one acknowledge bit
- most significant bit (MSB) first
- Data transfer example :
  - first byte transferred
  - during the first byte transfer:
    - master is transmitter
    - addressed slave is receiver
  - next bytes: depends on the last bit in address byte

# Addressing by 7 bits

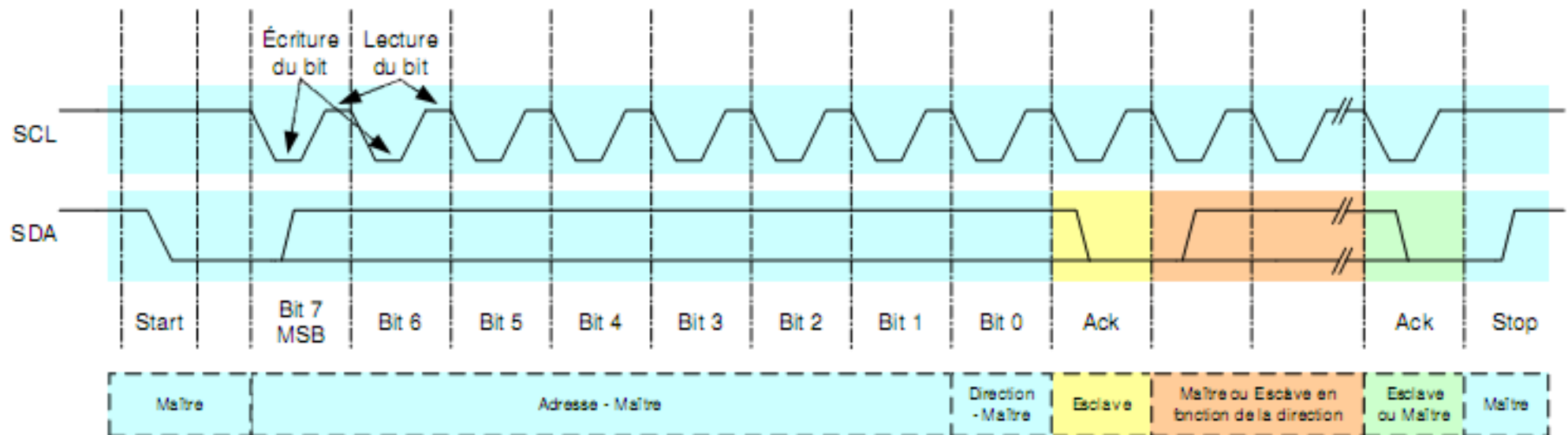
- the first byte transmitted by master:
  - 7 bits: address
  - 1 bit: direction (R/W)
    - 0 ... master writes data (W), becomes transmitter
    - 1 ... master reads data (R), becomes receiver
- data transfer terminated by stop condition
- master may generate repeated start and address another device
- each device listens to address
  - address matches its own → device switches state according to R/W bit
- address = fixed part + programmable part
  - fixed part assigned by I2C committee

# Data Transfer - SDA

- data bits are generated by transmitter as SCL pulses
- 9-th pulse:
  - transmitter releases SDA
  - receiver must hold SDA low in order to ack. received data
  - slave must release SDA after ack. bit (allows master to end frame)

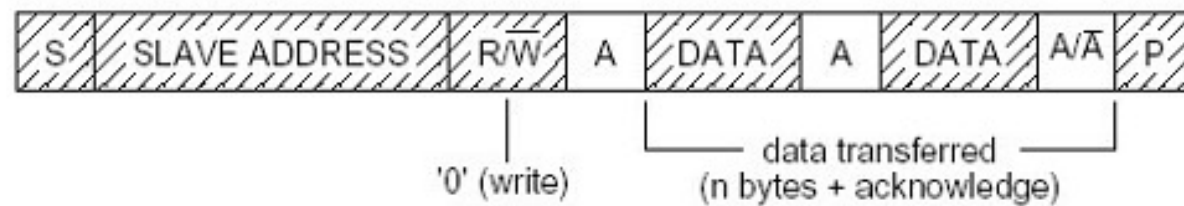


# Diagramme Temporel

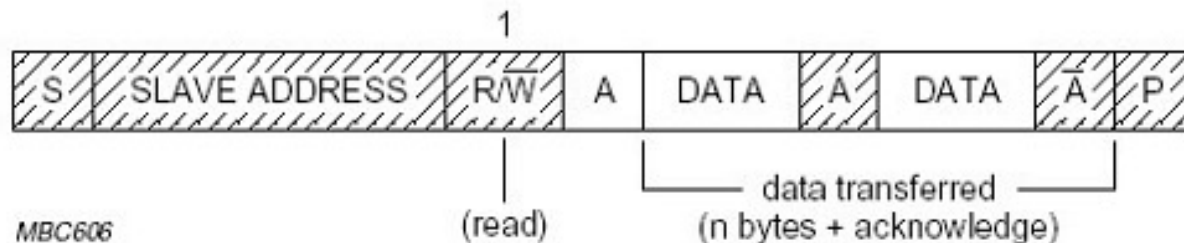


# Frame Formats

## master-transmitter



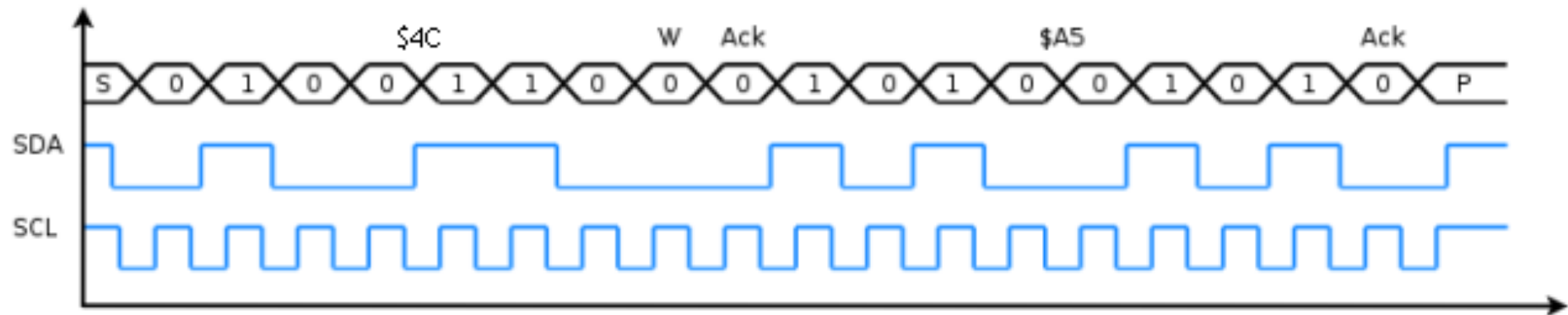
## master-receiver (since second byte)



MBC606

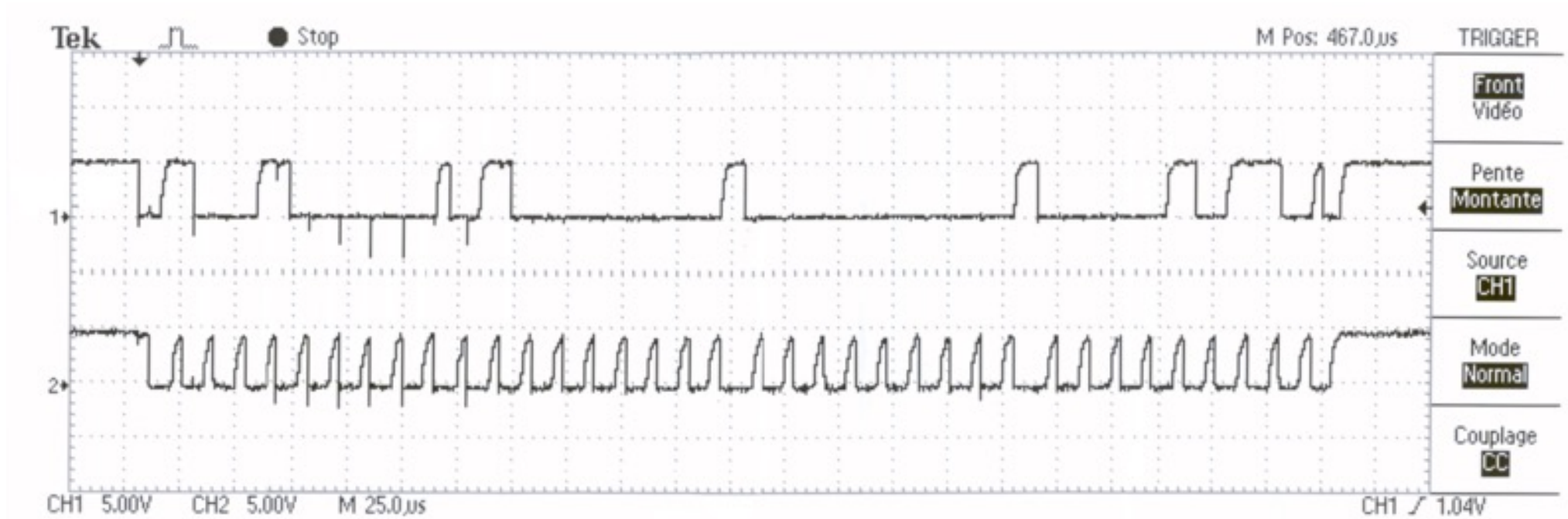


# Exemple



La communication commence par le StartBit  
puis l'adresse, (sur 8bits \$4C) avec bit de read/write à 0  
l'acknowledge (Ack)  
un octet de données (\$A5)  
de nouveau l'acknowledge (Ack)  
et enfin le StopBit.

# Exercice



Quel signal représente SCLK ?

Quel signal représente SDAT ?

Quels sont les messages qui sont transférés ?

Bus synchrone

4 fils (SCLK, SS, MISO, MOSI)

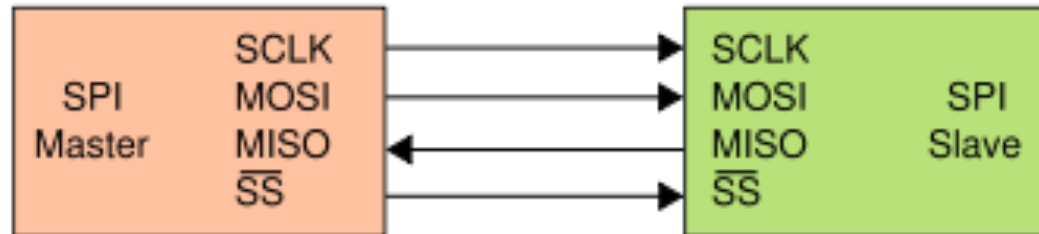
Liaison multipoint

Full duplex

## **BUS SPI (SERIAL PERIPHERAL INTERFACE)**

# Propriétés du SPI

- SPI : Serial Peripheral Interface
- Full Duplex
- Bus Synchrone

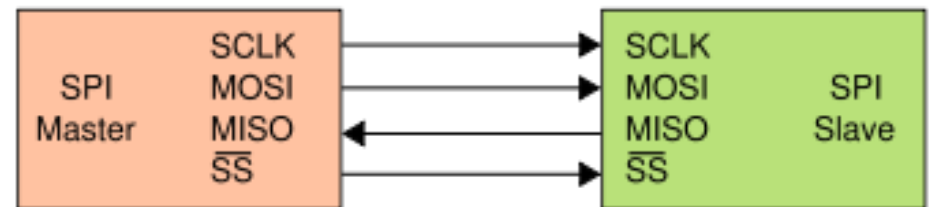
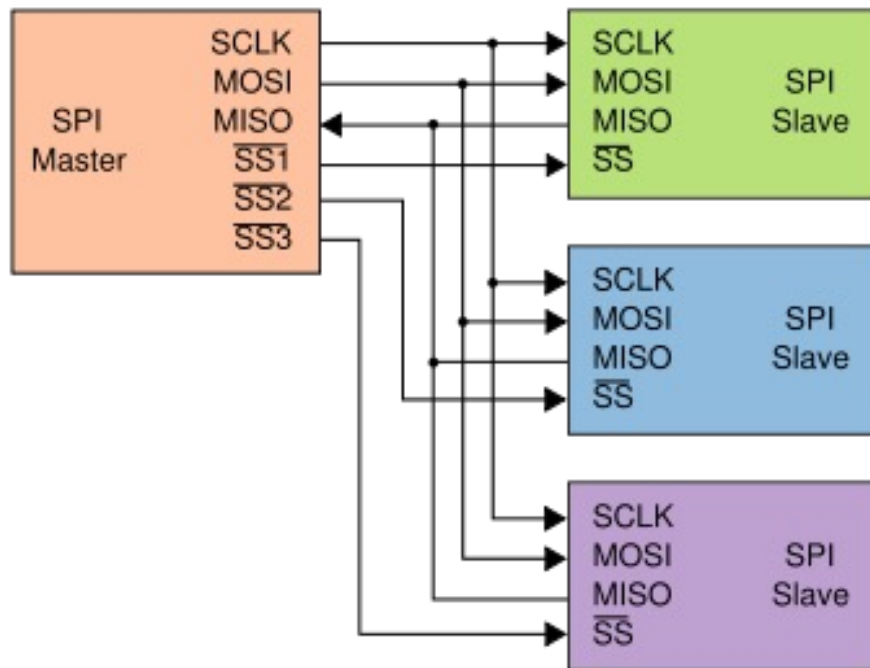


# Signaux du bus SPI

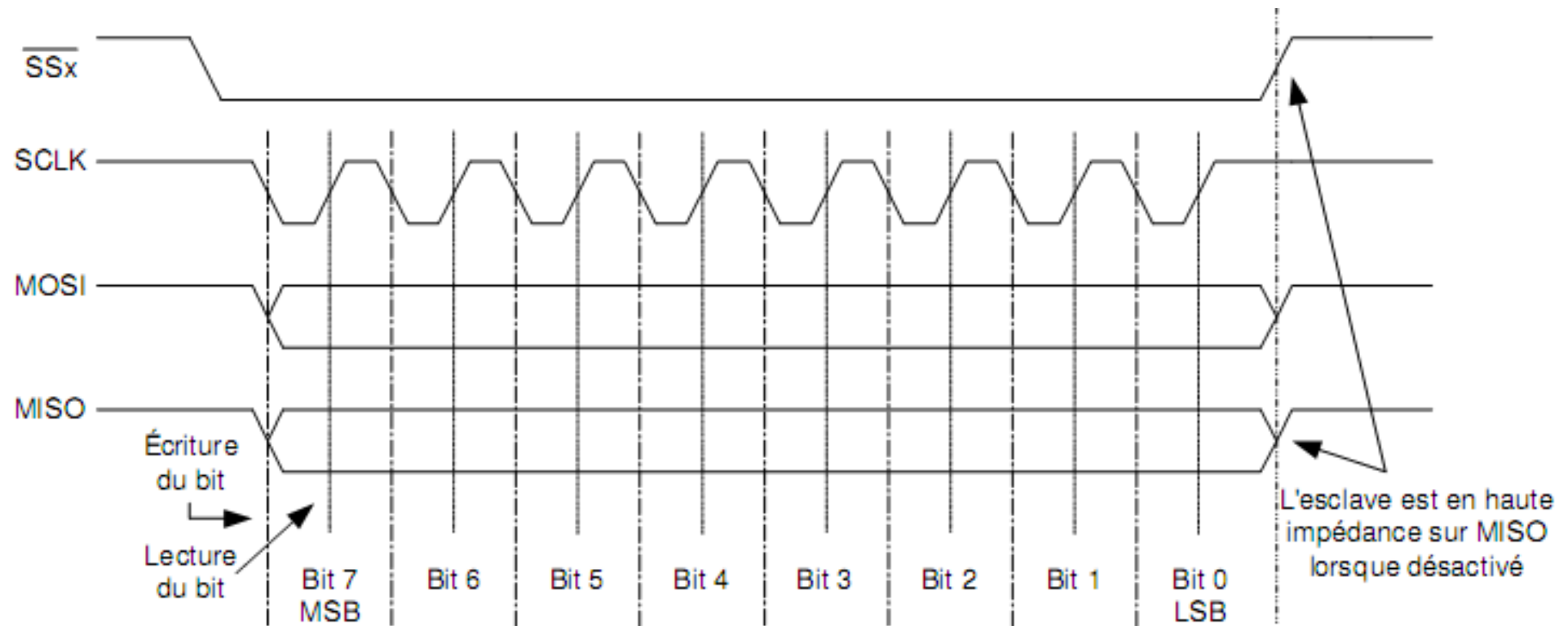
- Le bus SPI contient 4 signaux logiques
  - **SCLK** : Serial Clock Horloge (généré par le maître)
  - **MOSI** : Master Output, Slave Input (généré par le maître)
  - **MISO** : Master Input, Slave Output (généré par l'esclave)
  - **SS** : Slave Select, Actif à l'état bas, (généré par le maître)
- Il existe d'autres noms qui sont souvent utilisés.
  - **SCK** : Horloge (généré par le maître)
  - **SDI,DI,SI** : Serial Data IN
  - **SDO,DO,SO** : Serial Data OUT
  - **nCS, CS, nSS, STE** : Slave Select

# Maitre / Esclave

- Bus SPI : 1 maitre et un ou plusieurs esclaves.



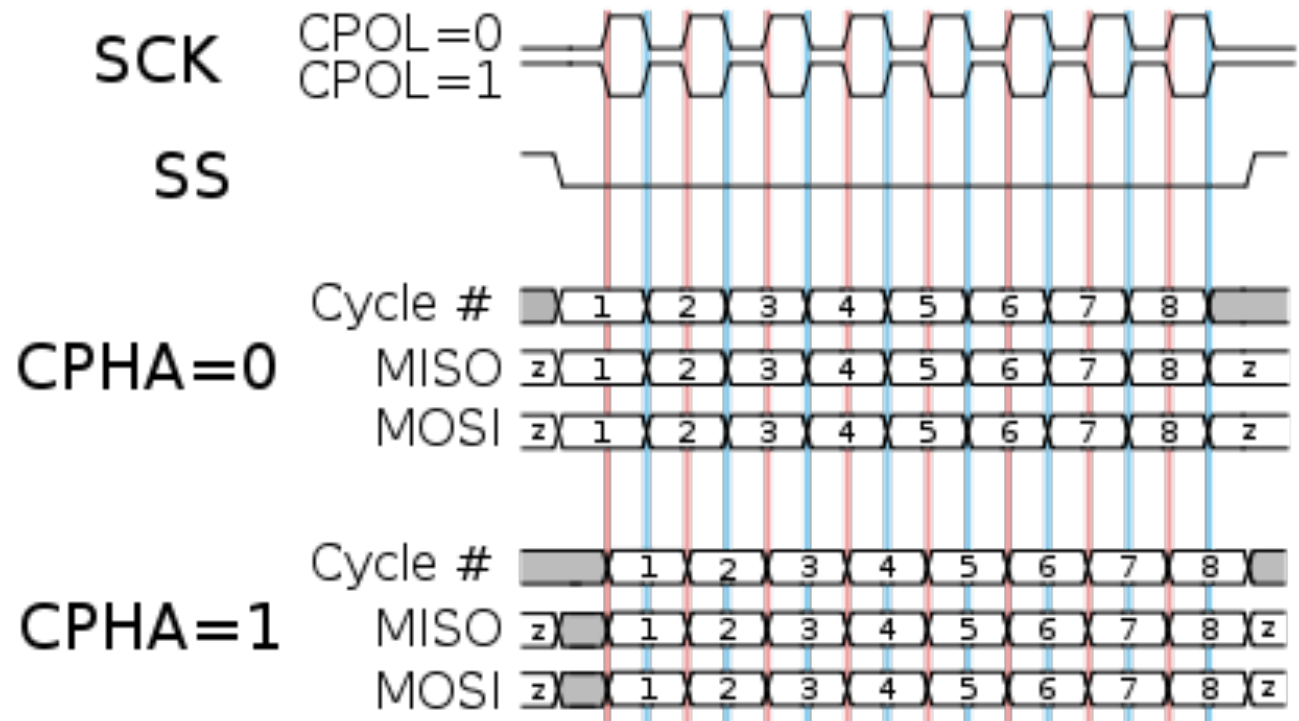
# Diagramme temporel d'une liaison SPI



# Clock Polarity / Clock Phase

CPOL (Clock Polarity)  
CPHA (Clock Phase)

Mode	CPOL	CPHA
0 (0,0)	0	0
1 (0,1)	0	1
2 (1,0)	1	0
3 (1,1)	1	1





# Avantages

- **Avantages**

- Communication Full duplex
- Débit plus important que I<sup>2</sup>C
- Flexibilité du nombre de bits à transmettre
- Simplicité de l'interface matérielle
  - Aucun arbitre nécessaire car aucune collision possible
  - Les esclaves utilisent l'horloge du maître et n'ont donc pas besoin d'oscillateur de précision
  - Pas de phy nécessaire
- Partage d'un bus commun pour l'horloge, MISO et MOSI entre les périphériques

# Inconvénients

- **Inconvénients**

- Monopolise plus de pattes d'un boîtier que l'[I2C](#) ou un [UART](#) qui en utilisent seulement deux.
- Aucun adressage possible, il faut une ligne de sélection par esclave en mode non chaîné.
- Le protocole n'a pas d'acquittement. Le maître peut parler dans le vide sans le savoir.
- Il ne peut y avoir qu'un seul maître sur le bus.
- Ne s'utilise que sur de courtes distances contrairement aux protocoles [RS-232](#), [RS-485](#) ou [bus CAN](#)

# Example

