**Introduction aux liaisons séries filaires UART, I2C, et SPI (et détail de leurs protocoles de communication)**



UART, I2C, SPI, … ces acronymes vous parlent sûrement, car on les retrouve souvent, en électronique ! Comme vous le savez certainement déjà, ce sont **différents types de communication série filaire**. Mais en pratique, savez-vous comment ils se câblent ? comment ils fonctionnent ? et à quel débit on peut les utiliser ?

C’est en tout cas ce que je vous propose de découvrir aujourd’hui, au travers de cet **article sur les différents types de liaisons séries filaires**. Par contre, pour rester digeste, je me limiterai ici à présenter uniquement les « principaux » protocoles de communication série, que nous rencontrons couramment de nos jours, sur nos petites cartes à microcontrôleur (comme l’Arduino, par exemple). Alors en avant !

**Sommaire**

[**1. Généralités**](https://passionelectronique.fr/liaisons-series-uart-i2c-spi/#generalites)

[**2. La liaison série UART**](https://passionelectronique.fr/liaisons-series-uart-i2c-spi/#la-liaison-serie-uart)

[**3. La liaison série I2C**](https://passionelectronique.fr/liaisons-series-uart-i2c-spi/#la-liaison-serie-i2c)

[**4. La liaison série SPI**](https://passionelectronique.fr/liaisons-series-uart-i2c-spi/#la-liaison-serie-spi)

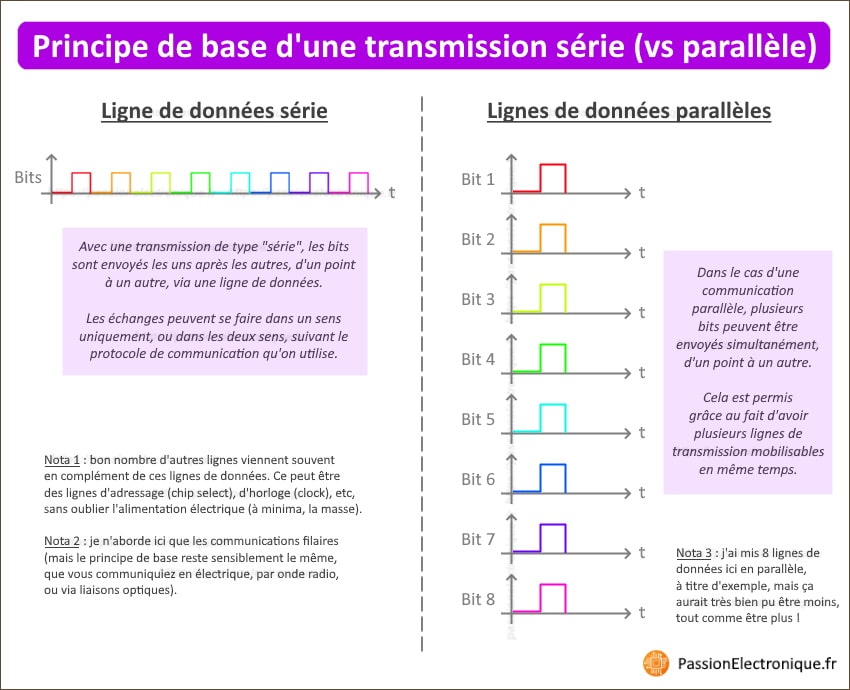
[**5. Les autres protocoles de communication série filaires**](https://passionelectronique.fr/liaisons-series-uart-i2c-spi/#les-autres-protocoles-de-communication-serie-filaires)

[**6. Tableau comparatif UART / I2C / SPI (synthèse)**](https://passionelectronique.fr/liaisons-series-uart-i2c-spi/#tableau-comparatif-uart-i2c-spi-synthese)

[**7. Conclusion**](https://passionelectronique.fr/liaisons-series-uart-i2c-spi/#conclusion)

**Généralités**

Par définition, **une liaison série est un moyen de relier deux éléments (ou plus), afin que ceux-ci puissent échanger des informations, et ce, en utilisant un minimum de fils**.



**La liaison série UART**

**Le protocol UART** (de l’anglais « Universal Asynchronous Receiver Transmitter »). Il permet à deux éléments (pas plus), de communiquer ensemble, via une liaison série filaire.

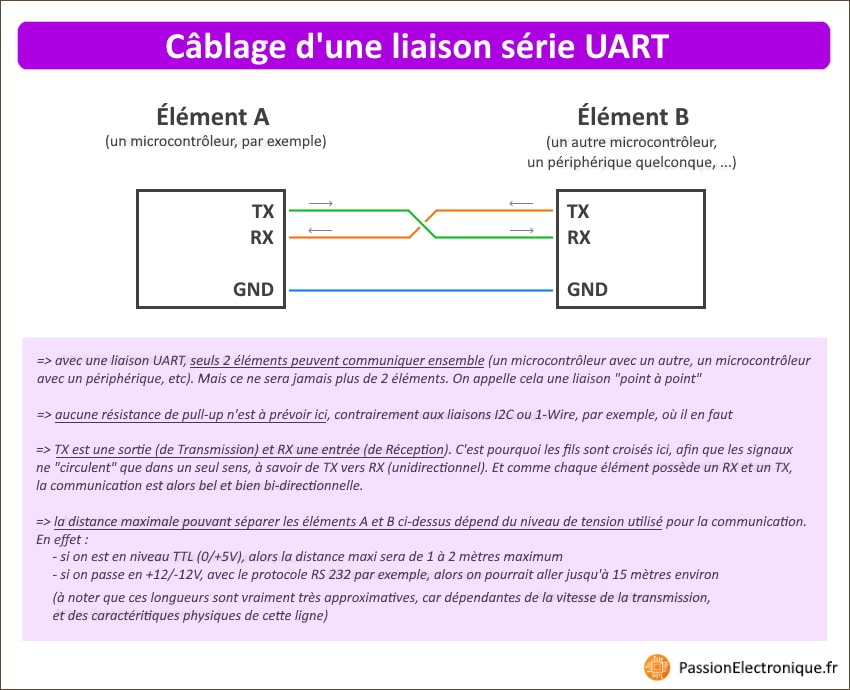
**La liaison UART est ce qu’on appelle une liaison point à point, en ce sens qu’elle ne permet de relier que 2 éléments, et deux éléments seulement**.

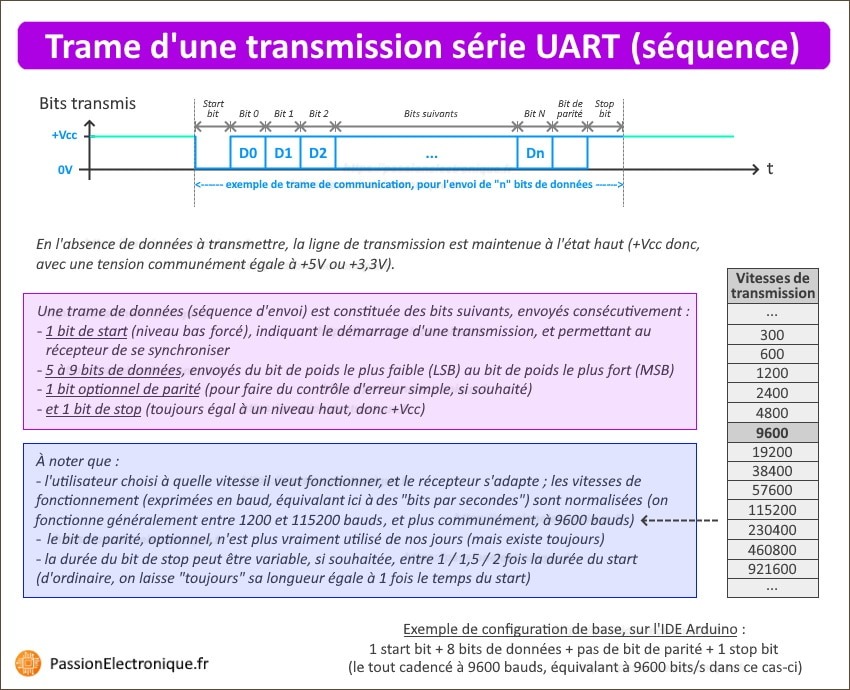
En fait, chaque élément dispose de :

* une entrée pour la réception de données, notée RX
* une sortie pour la transmission de données, notée TX

U**ne liaison filaire UART comporte 2 fils de communication, dont le premier permet de faire circuler les données dans un sens (TX → RX), et dont le second permet de faire circuler les données dans l’autre sens (RX ← TX)**.

En image, voici ce que cela donne :





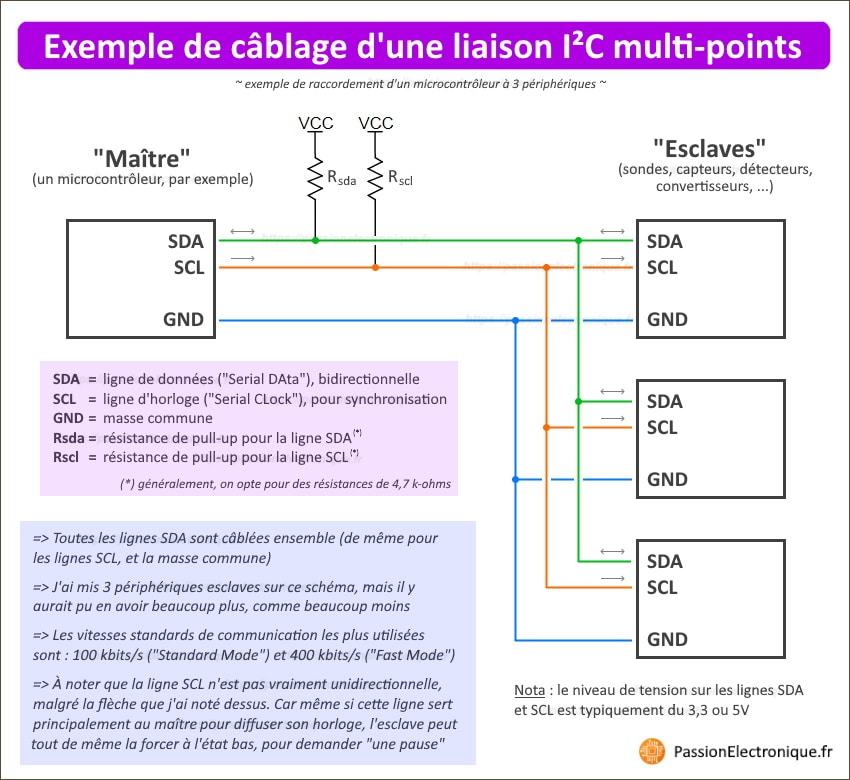
Par contre, **il est possible de communiquer sur de plus grandes distances, en utilisant par exemple le protocole RS-232 (« calqué » sur le UART)**. Ceci permet d’aller jusqu’à :

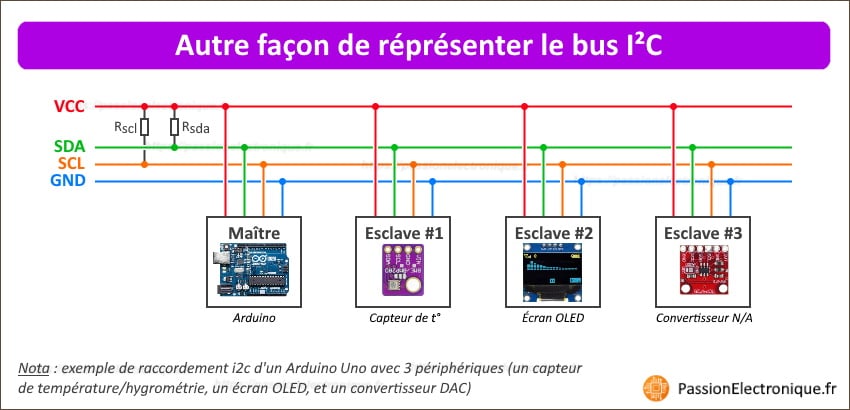
* 2 m environ, pour du 56 kbits/s
* 3,5 m environ, pour du 38400 bits/s
* 7,5 m environ, pour du 19200 bits/s
* 15 m environ, pour du 9600 bits/s (c’est la valeur la plus couramment usitée)
* 30 m environ, pour du 4800 bits/s
* 60 m environ, pour du 2400 bits/s

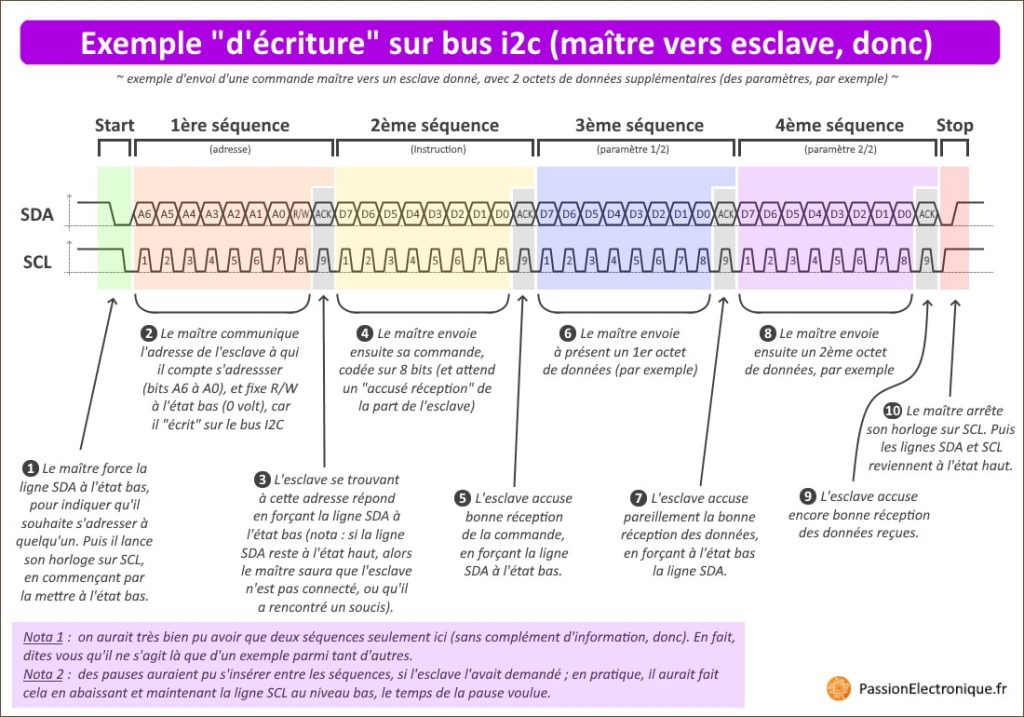
**La liaison série I2C**

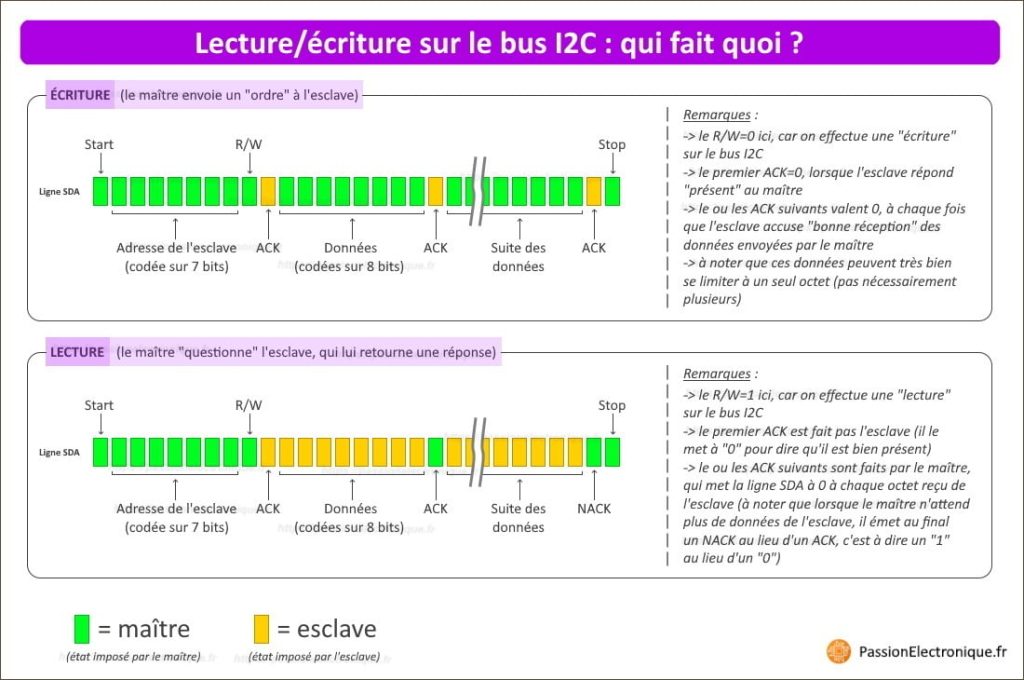
Ce type de communication série permet l’échange de données séries filaires, notamment entre un « maître » (un microcontrôleur, par exemple) et des « esclaves » (des périphériques). On appelle également « bus I2C » l’ensemble des liens constituant l’interconnexion du « maître » avec tous ses « esclaves ».

De nos jours, **le bus I2C est devenu un moyen de communication très prisé**, dès lors qu’on cherche à interfacer un microcontrôleur avec des périphériques, tels que des :





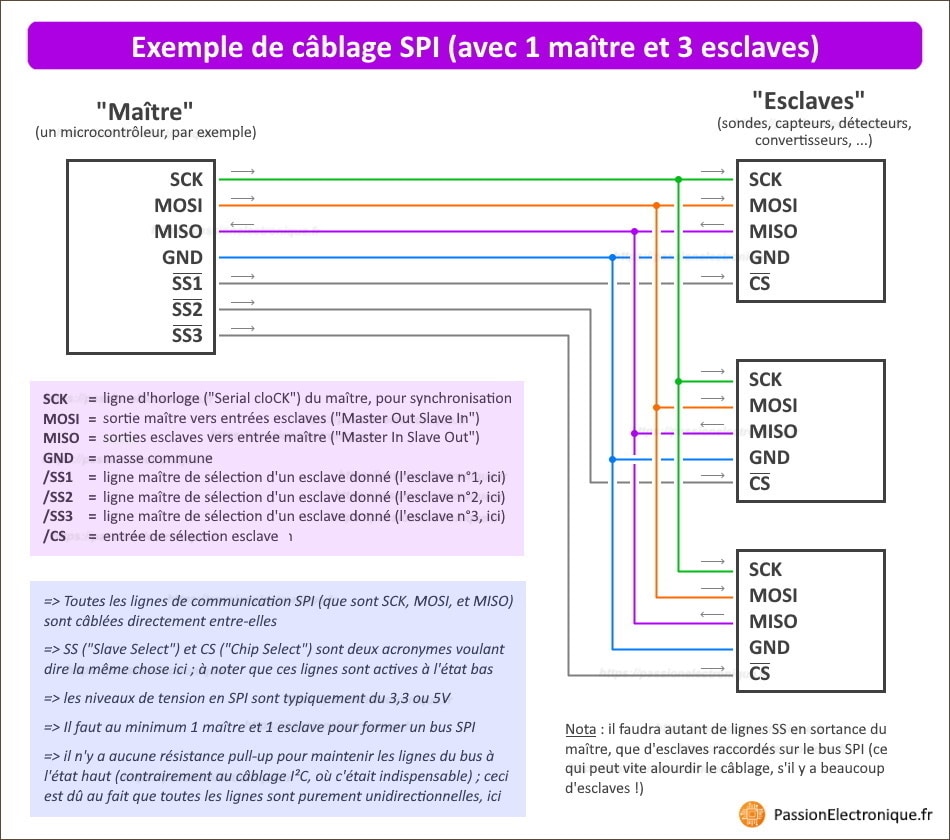


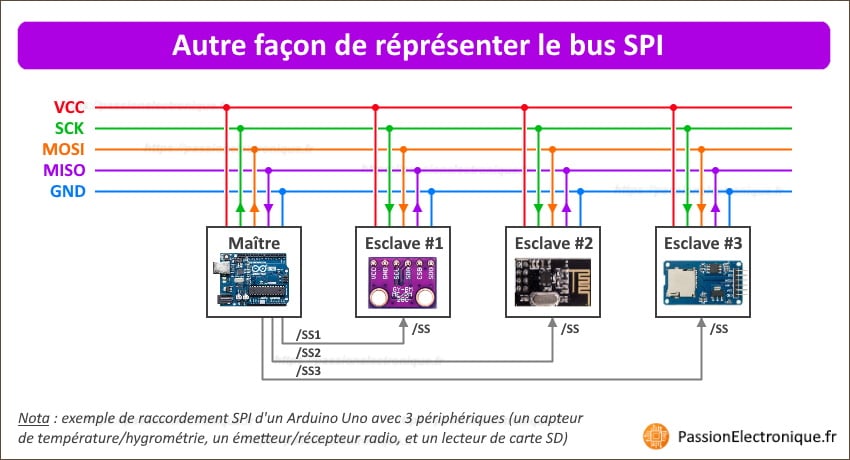


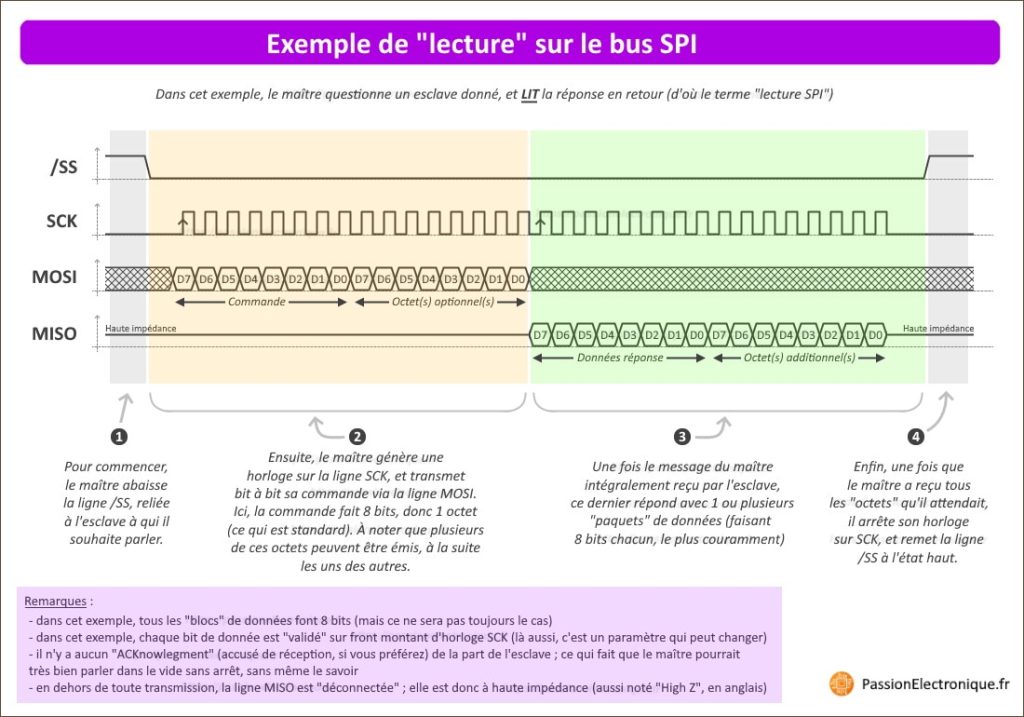
**La liaison série SPI**

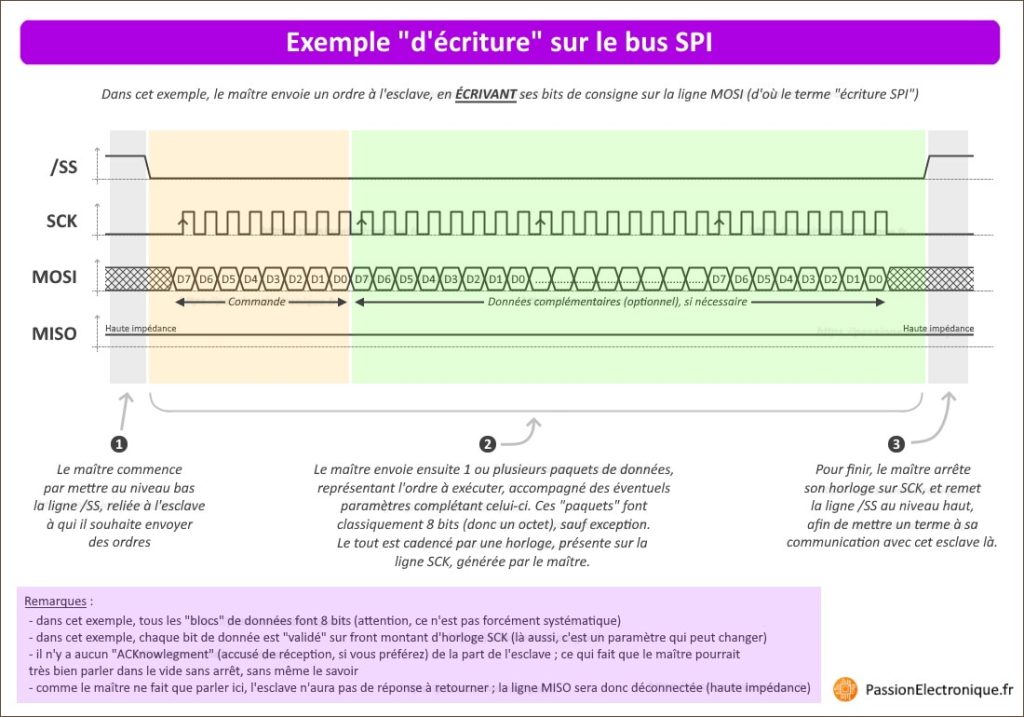
Il s’agit d’une liaison maître/esclaves, avec, généralement :

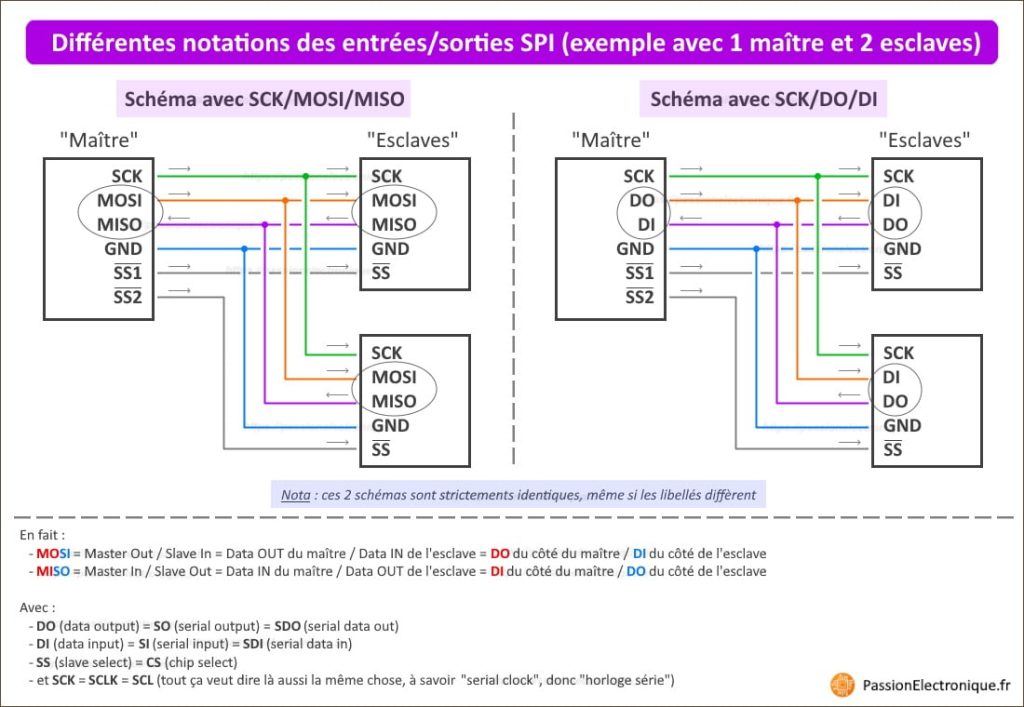
* un microcontrôleur comme maître
* et des périphériques (capteurs, sondes, convertisseurs, …) comme esclaves











**Les autres protocoles de communication série filaires**

* au bus CAN, très prisé dans le milieu automobile, qui permet avec 2 fils seulement (hors alimentation) de relier 20 à 30 éléments ensemble (selon si on fonctionne à 125 kbits/s, ou 1 Mbits/s) ; sans parler des longueurs de communication qui sont vraiment appréciables : environ 30m à 1 Mbits/s, et jusqu’à 5 km à 10 kbits/s ! Ce type de liaison filaire série est d’ailleurs assez facile à mettre à œuvre, notamment avec nos arduino, et aux mini shields « tout équipés » MCP2515 / TJA1050.
* au bus 1-Wire, popularisé avec les sondes de température DS18B20 de chez Dallas Semiconductor ; ici, un seul fil suffit, et la masse ! Difficile de faire plus simple, niveau câblage, bien que des fois, il faille rajouter le +Vcc ! Et c’est d’autant plus simple à mettre en œuvre qu’il existe de nombreuses librairies gérant cela (car à coder, ce serait fastidieux !). Du reste, une liaison 1-Wire est bidirectionnelle, fonctionne en semi-duplex (un seul sens à la fois, au niveau de la transmission), nécessite une résistance de pull-up, et fonctionne à 16 kbits/s environ, en standard (ce qui, au passage, permet d’aller sur « d’assez longues » distances)

**Tableau comparatif UART / I2C / SPI (synthèse)**

