Communication avec le bus SPI

Objectifs :

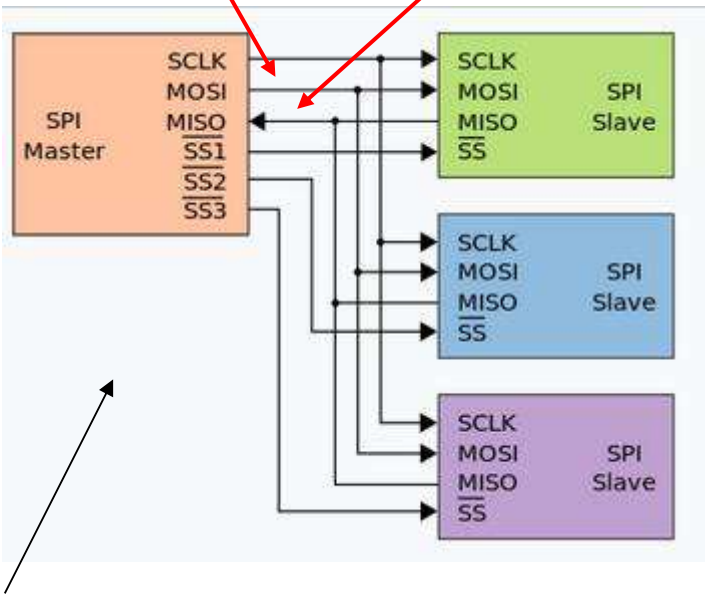
Comprendre le bus SPI et mettre en place une communication entre deux matériels.

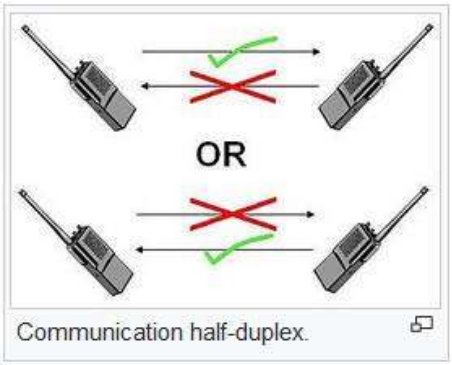
PRESENTATION DU PROTOCOLE SPI (*Serial Peripheral Interface)*

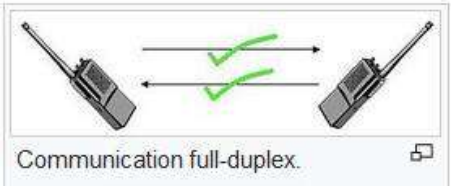
Le bus SPI est un bus de données série synchrone de type full duplex.
La communication est de type maître esclave donc pas de perte de données.

Le maître écrit sur cette ligne (MOSI) pour l'esclave sélectionné

Le maître lit sur cette ligne (MISO) les données fournis par l'esclave sélectionné







Dans le schéma ci-dessus, il y a un maître et trois esclaves.
Le maître parle un seul esclave à la fois.

Principe de la sélection d'un esclave par le maître par les trois sorties /SS3, /SS2 et /SS1.

/SS3	/SS2	/SS1		SPI vert	SPI bleu	SPI Violet
1	1	0		Non sélectionné	Non sélectionné	Sélectionné
1	0	1		Non sélectionné	Sélectionné	Non sélectionné
0	1	1		Sélectionné	Non sélectionné	Non sélectionné
1	1	1		Non sélectionné	Non sélectionné	Non sélectionné

Résumé : pour la liaison série synchrone

Les données (octets) sont envoyées sur la ligne MOSI en série en principe de poids fort (B7) vers poids faible (B0) et lus Par un esclave sur un front d'horloge.

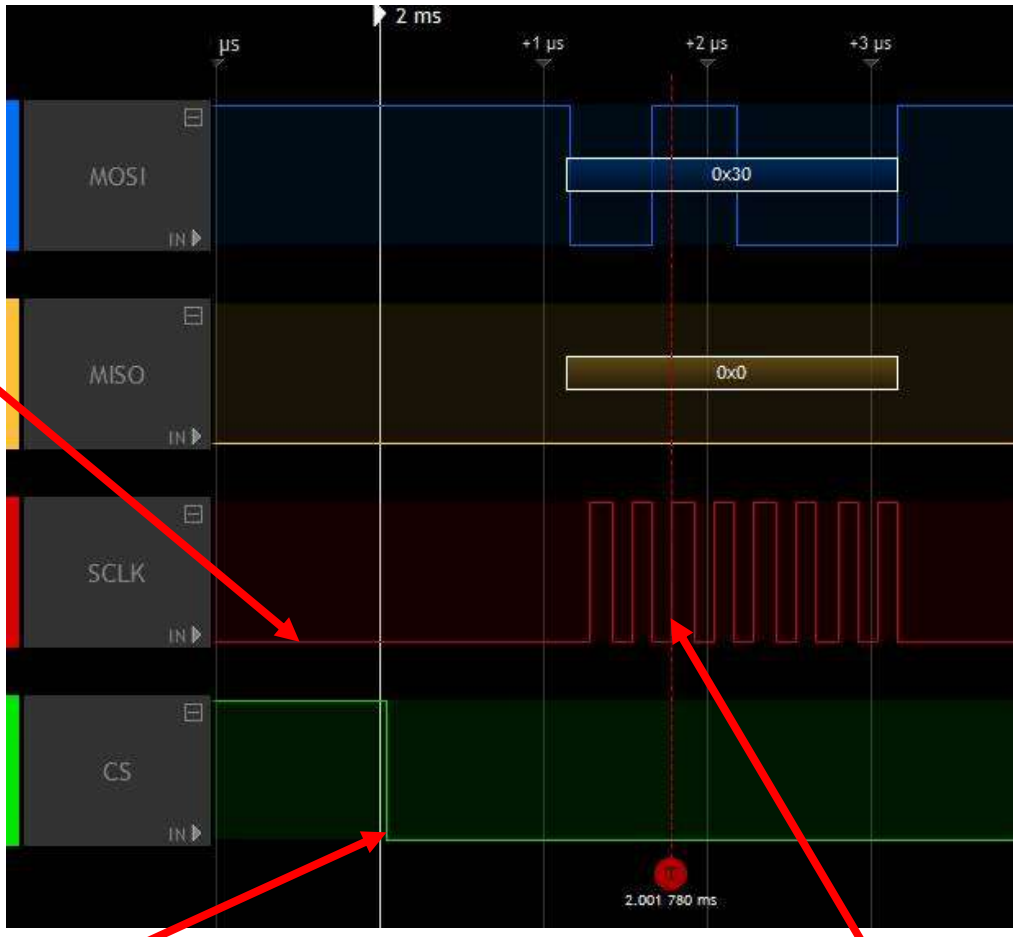
Remarque :

Il existe quatre modes pour la transmission des données : **SPI_MODE0 à SPI_MODE3**

Mode	Clock Polarity (CPOL)	Clock Phase (CPHA)	Output Edge	Data Capture
SPI_MODE0	0	0	Falling	Rising
SPI_MODE1	0	1	Rising	Falling
SPI_MODE2	1	0	Rising	Falling
SPI_MODE3	1	1	Falling	Rising

Exemple :

SCLK = 0 (CPOL = 0) avant la sélection du boîtier.



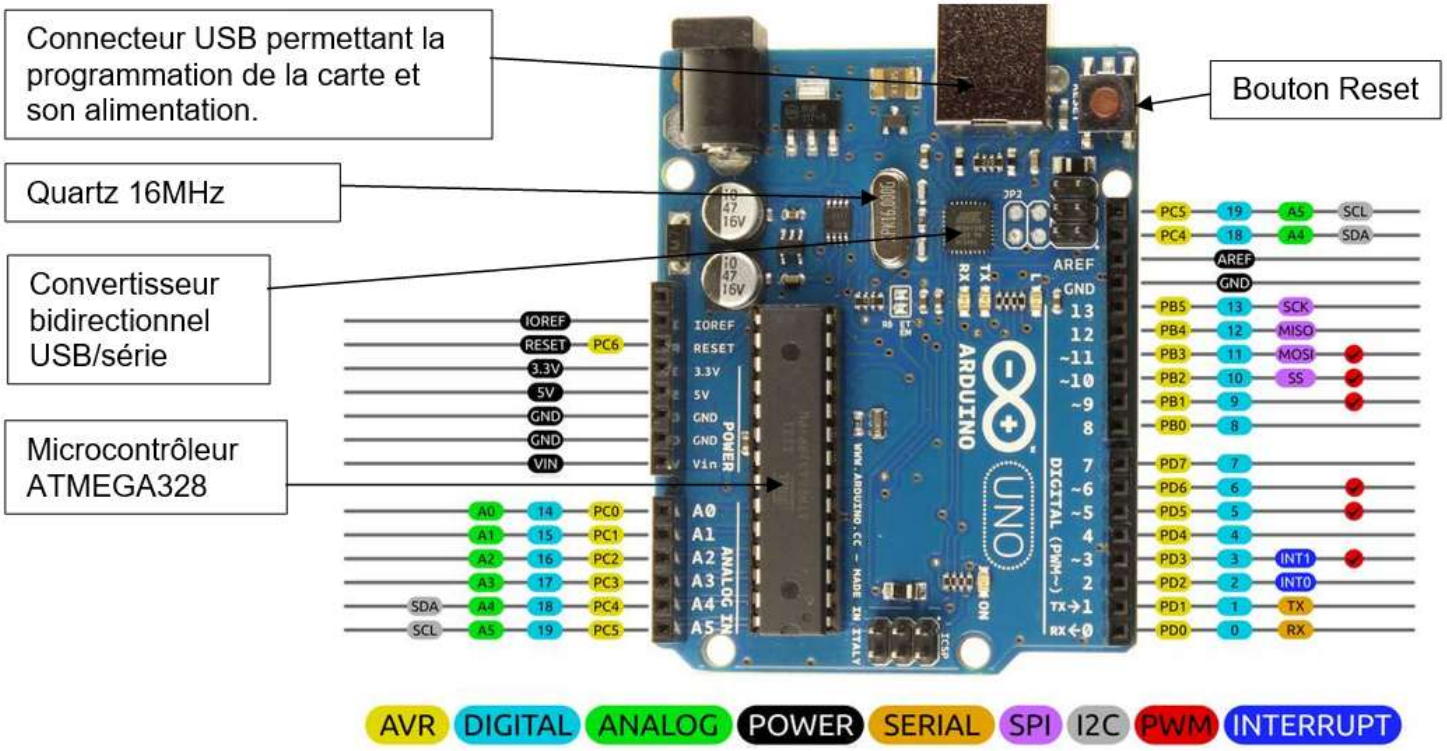
Sélection du circuit : mise à 0 de la broche CS (chip select)

Les données sont lues sur le front montant de SCLK

Le chronogramme ci-dessus correspond au mode : **SPI_MODE0**

ETUDE DE LA BIBLIOTHEQUE SPI pour Arduino

1 Aspect matériel:



La communication SPI utilise les broches :

11 pour MOSI
12 pour MISO
13 pour SCK (SCLK)
10 pour SS

2 Aspect Logiciel :

1 Il faut faire un #include<SPI.h>
2 Les différentes fonctions possibles avec Arduino1.6

- SPISettings
- begin()
- end()
- beginTransaction()
- endTransaction()
- setBitOrder()
- setClockDivider()
- setDataMode()
- transfer()

Les plus utiles sont :

SPI.begin() ; // en maître seulement
SPI.end() ;
SPI.transfert(unsigned char) ;

Syntax

receivedVal = SPI.transfer(val)
receivedVal16 = SPI.transfer16(val16)
SPI.transfer(buffer, size)

Parameters

val: the byte to send out over the bus
val16: the two bytes variable to send out over the bus
buffer: the array of data to be transferred

Returns

the received data

3 Exemple : Transmission de données à un esclave.

```
#include <SPI.h>
unsigned char tab[6]={'0','1','2','A','B','C'};

void setup(){
  SPI.begin(); // f= 4MHz SPI_mode0
}

void loop() {
  digitalWrite(10,LOW);  // selection esclave sur broche 10
  for(int i=0;i<=5;i++)
  SPI.transfer(tab[i]);  // écriture des données
  digitalWrite(10,HIGH); // devalidation de l'esclave
  delay(2);
}
```