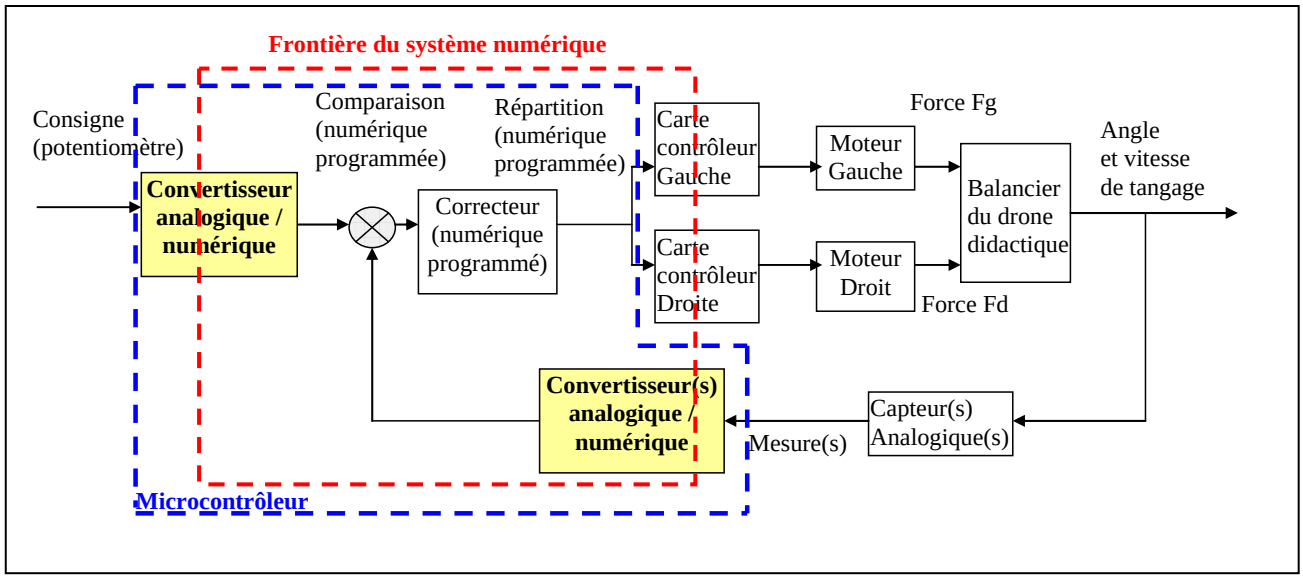
Fiche TP « traitement numérique des signaux par (pour) le microcontrôleur »

1- Présentation

Les calculs (traitement « numérique ») réalisés par le microcontrôleur sont des traitements programmés qui portent sur des grandeurs numériques, par exemple : comparaison des signaux, mise en place des correcteurs, répartition des signaux vers les deux cartes « contrôleur » des moteurs.

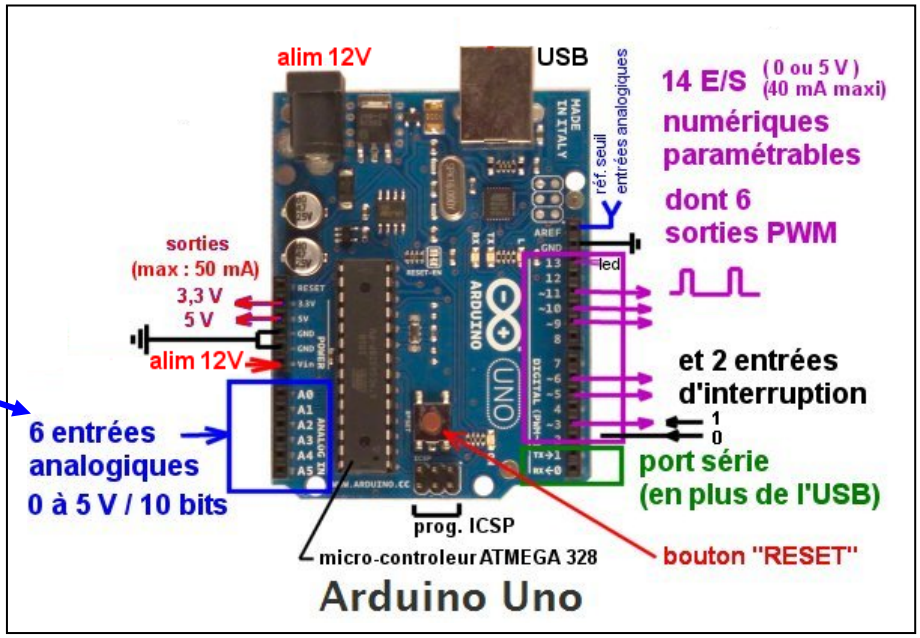
Le schéma ci-dessous montre que les grandeurs « consigne » et « mesure(s) » sont des grandeurs « analogiques », car fournies par des composants électroniques à signaux continus (potentiomètre et capteurs analogiques).



Pour pouvoir être traitées par programmation dans le microcontrôleur, ces grandeurs analogiques sont transformées par les convertisseurs numériques/analogiques présents sur les entrées dites « analogiques » ;

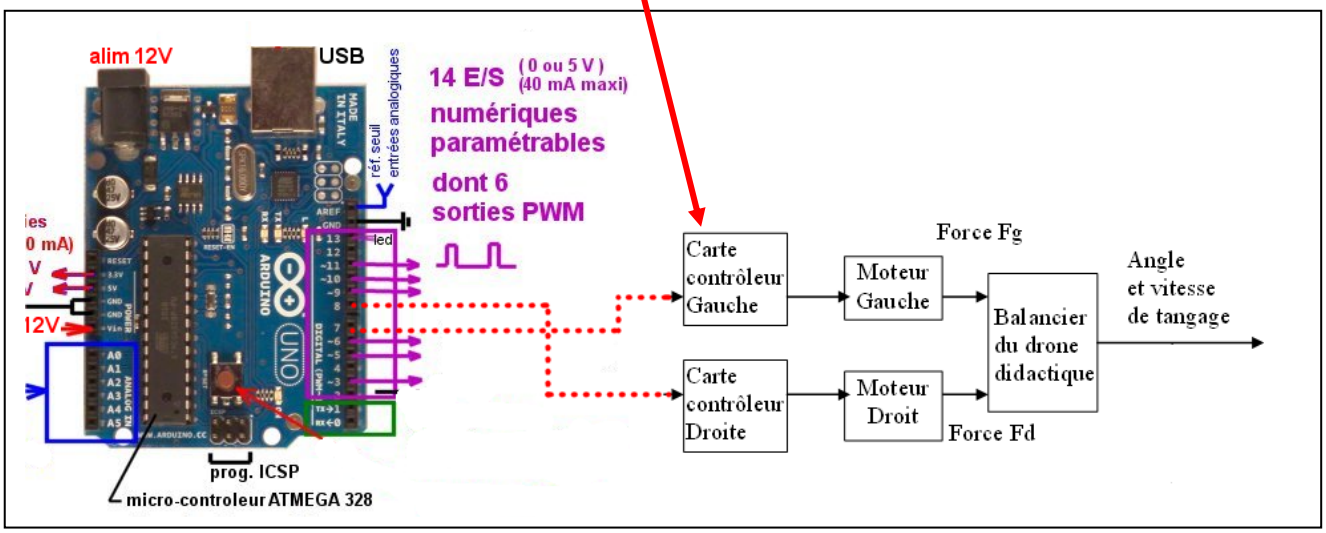
On repère à gauche de l'image ci-contre, les 6 entrées analogiques sur la carte à microcontrôleur Arduino (numérotées A0 à A5)

Les signaux numériques qui résultent de la conversion sont fournis sur 10 bits.



Société DMS
Aéroparc Saint Martin – 12 rue de Caulet – 31300 TOULOUSE – ☎ : + 33 (0)5 62 88 72 72 🕮 : + 33 (0)5 62 88 72 79
Site internet : www.dmseducation.com Email : info@dmseducation.com

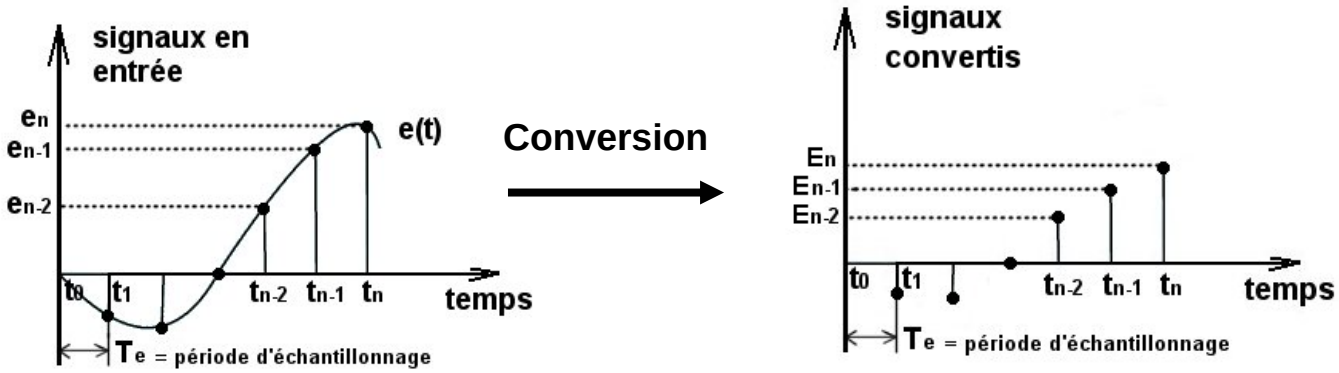
Pour information, on notera que les commandes des moteurs fournies par le microcontrôleur sont dirigées vers les « cartes contrôleur » gauche et droite sous forme de signaux numériques (binaires : 0 ou 5 volts) ; ceux-ci seront fournis par les sorties numériques de la carte à microcontrôleur Arduino :



2- L'acquisition des signaux analogiques par le microcontrôleur

2-1 L'échantillonnage :

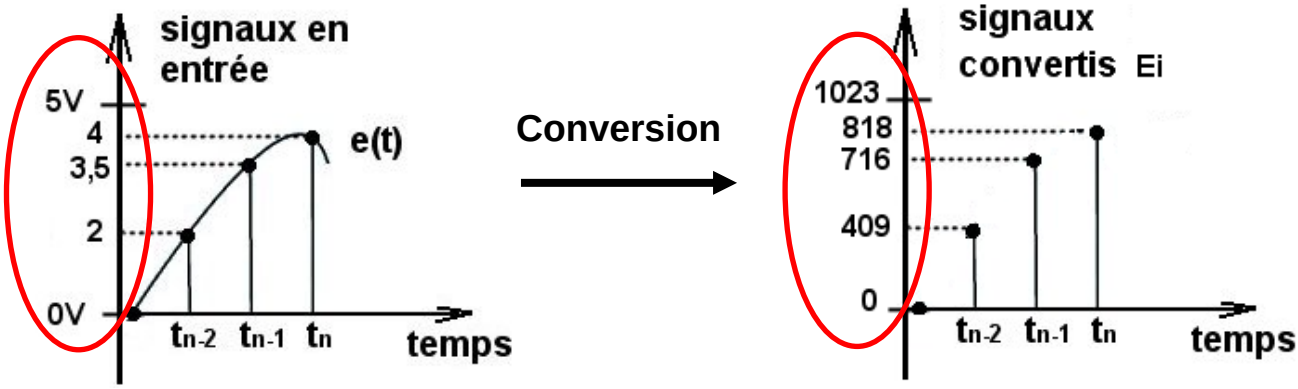
Le « micro-contrôleur » qui doit effectuer les opérations sur les grandeurs de la boucle d'asservissement (comparaison, correction …) fonctionne dans le domaine « échantillonné » ;
Concernant l'acquisition de données, la prise des informations d'entrée « e(t) » (consignes, mesures) ainsi que la délivrance des résultats de conversion « Ei » , s'effectuent de manière discontinue à des instants successifs définis par la « période d'échantillonnage » Te.



2-2 La quantification :

Pour le microcontrôleur de la carte Arduino, la **plage d'entrée des convertisseurs est définie de 0 à 5 Volts**.
Pour que la conversion se déroule correctement, il est nécessaire que le signal d'entrée ne soit pas à l'extérieur de cette plage, sinon le signal d'entrée est tronqué.

Par ailleurs, le micro-contrôleur de l'Arduino effectue l'acquisition des signaux analogiques à l'aide d'un convertisseur analogique/numérique qui **quantifie le signal sur 10 bits** ;
Les grandeurs attribuées aux signaux convertis « Ei » seront donc fournies sur au maximum $2^{10} = 1024$ valeurs (ou « points ») du calculateur (donc de 0 à 1023).



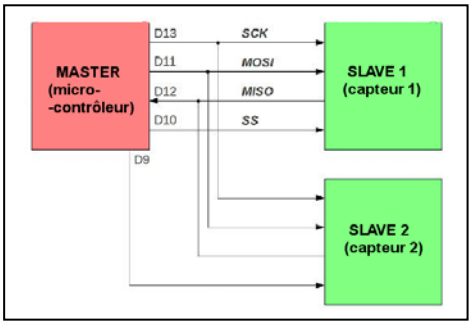
On notera que ceci donne un « pas de quantification » ou « quantum » de 5 / 1023 = 0,00489 Volt / point (environ 5 millivolts par point)

3- Le transfert des signaux numériques du gyromètre par le bus « SPI »

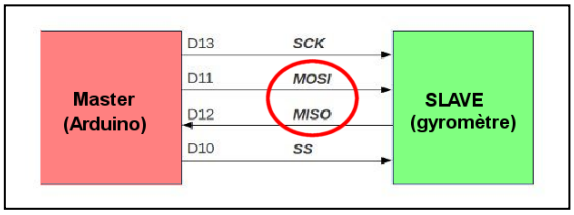
Le gyromètre MLX 90609 possède une particularité : il peut délivrer son information soit sous forme d'un signal analogique, soit sous forme numérique. Les deux types de sorties sont générées simultanément du fait que la sortie numérique est distincte de la sortie analogique.

3-1 connectique de la sortie numérique du gyromètre (bus « SPI »)

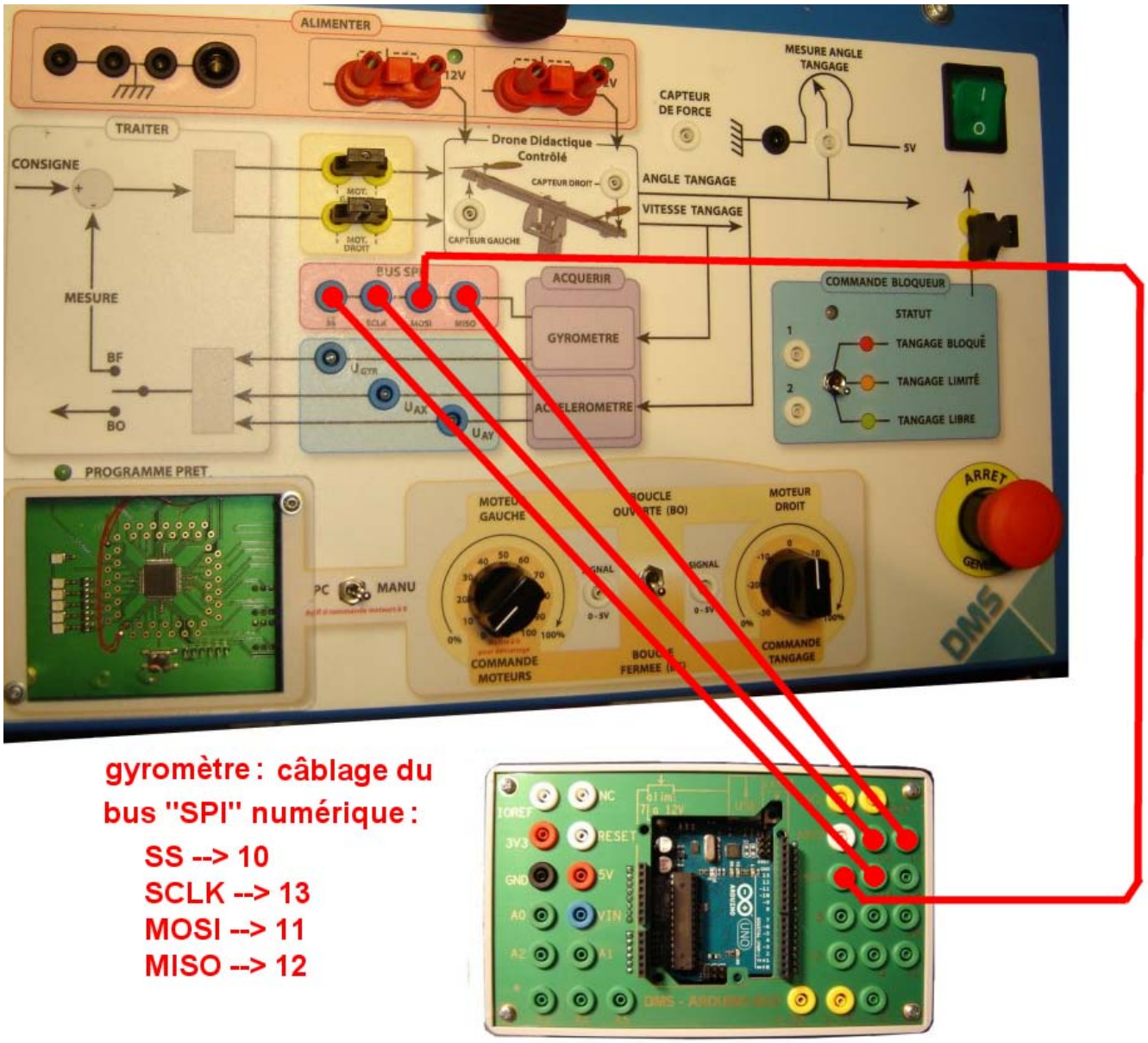
La sortie numérique s'effectue avec le protocole de transmission série « SPI », mis au point par les industriels pour converser simultanément avec de nombreux capteurs.
En général, le microcontrôleur et appelé « Master », les capteurs sont appelés « Slave » ; 4 fils sont nécessaires.
- SCK est un fil pour le signal d'horloge (synchronisation) ;
- SS est un fil qui permet de sélectionner le capteur auquel la transmission est destinée (il faut donc une ligne SS différente pour chaque capteur) ;



Les informations peuvent circuler simultanément dans les deux sens :
- sens « Master → Slave » : sur la ligne nommée « MOSI » (abréviation de : Master Out Slave In) ;
- sens "Slave → Master" : sur la ligne nommée « MISO » (abréviation de : Master In Slave Out).



Mise en place de la connectique pour l'utilisation du bus SPI avec l'arduino-box :

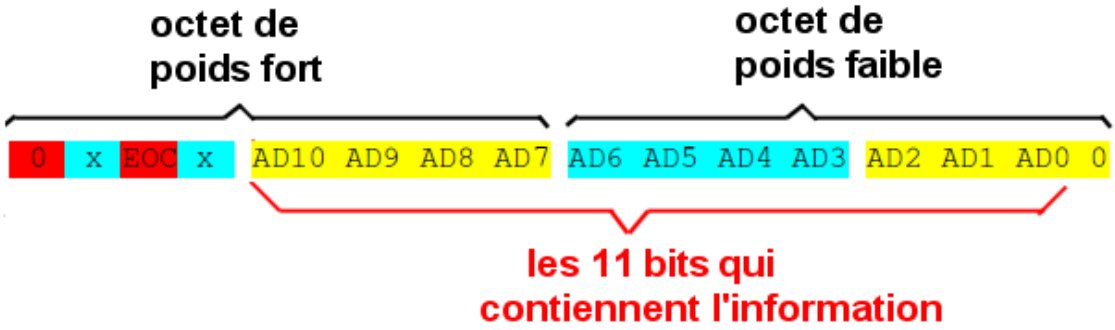


3-2 codage de l'information pour commander la lecture du gyromètre

On donne ci-dessous un extrait simplifié de la fiche technique du gyromètre MLX90609 (p 13) :
- Les instructions qui permettent de faire fonctionner la transmission de données sont envoyées au gyromètre sous forme d'octets (mot de 8 bits), avec les valeurs suivantes :

Instruction	Octet à envoyer au gyromètre (x = 0 ou 1)
Activer la ligne	1001x100 (et attendre 115 microsecondes)
Lancer la conversion numérique de la vitesse gyro	10010100 (et attendre 115 microsecondes)
Envoyer les valeurs sur le bus SPI vers le master	10000000

- Les valeurs renvoyées par le capteur sur le bus SPI sont constituées de 2 octets (= 16 bits au total) ;



Néanmoins, les 11 bits qui contiennent l'information de vitesse doivent être extraits à partir de ces deux octets comme le montre la figure ci-dessus.

3-3 protocole simplifié de transmission et le décodage

On donne ci-dessous un extrait simplifié de la fiche technique du gyromètre MLX90609 (p 14) :

Transmission (les ordres sont envoyés depuis le « Master » arduino :
- étape 1 : Activer le convertisseur analogique numérique du gyromètre
- étape 2 : Lancer la conversion numérique de la mesure de vitesse sur le gyromètre
- étape 3 : Faire envoyer les valeurs sur le bus SPI (les deux octets qui contiennent l'information ne peuvent être lus que l'un après l'autre, après avoir reçu préalablement 3 octets)

Décodage :
- étape 4 : recevoir les deux octets utiles ;
  premier `0 x EOC x AD10 AD9 AD8 AD7`
  deuxième `AD6 AD5 AD4 AD3 AD2 AD1 AD0 0`
- étape 5 : assembler dans le bon ordre les deux octets pour composer un mot de 16 bits
  `0 x EOC x AD10 AD9 AD8 AD7 AD6 AD5 AD4 AD3 AD2 AD1 AD0 0`
- étape 6 : décaler les 15 bits de poids fort vers la droite pour éliminer le bit de poids le plus faible
  `0 0 x EOC x AD10 AD9 AD8 AD7 AD6 AD5 AD4 AD3 AD2 AD1 AD0`
- étape 7 : récupérer les 11 bits de poids faible qui résultent de ce décalage ;
  `AD10 AD9 AD8 AD7 AD6 AD5 AD4 AD3 AD2 AD1 AD0`
- étape 8 : exprimer cette valeur en nombre décimal

Programmation : le programme ***« _0_Mesure_Gyro_SPI.ino »*** à téléverser dans Arduino réalise ces étapes, pour afficher le résultat sur le port série.