Florian LEBOUL Stage Interfaçage fluorimètre

10/04/2017

Premier rapport d’analyses

Problème :

Ancienne machine, ne disposant plus de logiciel PC pour exploiter les données.

Objet du stage :

Création d’un logiciel permettant de récupérer les données renvoyées par le bloc de commande et les photodiodes, pour les synthétiser et les synchroniser sous forme graphique en temps réel, avec possibilité de les exporter dans des formats réutilisables (XML, CSV, excel, HTML, …).

Problématique : Récupérer les informations, les synchroniser, et les envoyer au pc pour analyse.

* L’appareil de commande dispose d’une prise au format DB-25 communicant sous le protocole RS-232. Ce protocole et les prises qui lui sont associées sont rarement présentes sur les machines récentes du fait de leur âge. Les photodiodes (postérieurement désignées PD) quant à elles, fournissent simplement une tension entre 0 et 7.5V max théorique (valeur théorique car risque de détérioration si soumis à une luminosité correspondant à ce niveau de tension).
* L’idée est donc de permettre à n’importe quel ordinateur de pouvoir récupérer ces informations. Pour cela, l’utilisation du standard USB semble idéale, de par son universalité. Un simple adaptateur RS-232/USB pourrait suffire pour exploiter les données de l’appareil de commande, mais il faudrait utiliser une seconde prise du PC pour connecter les PD.
* La solution proposée consiste à utiliser une carte type Arduino, associée à un MAX232. Cet ensemble communiquerait avec le PC.
  + Le MAX232 est un circuit intégré, permettant de convertir un signal RS232 au format TTL (simple liaison Rx-Tx), exploitable via l’Arduino. Ce circuit doit être alimenté en 5V, donc en théorie une alimentation par le port USB du PC serait possible. En pratique, un PC voit ses ports bridés à 500mA, donc l’idéal serait d’ajouter au système un transformateur délivrant du 5V, avec une intensité max de 1,5A (disons « Pour la marge », la carte Arduino consommerait 50mA max en fonction du type de carte, le MAX232 200mA max, donc 1,5A laisse une très bonne marge). Un autre avantage du MAX232 est qu’il est bon marché (2€ en magasin), et ne nécessite pas beaucoup de composants connexes (seulement 4 condensateurs polarisés de 1/0.1µF à 10cts pièce).
  + Concernant les PD, les cartes Arduino ont l’avantage de disposer d’entrées analogiques, fournissant une valeur sur 10 bits, correspondant au résultat de la tension appliqué sur l’entrée divisée par la tension d’alimentation, le tout multiplié par 1024 (par exemple, si on applique 2.5V, (2.5/5)\*1024 = 512). L’ « inconvénient » de ces entrés est qu’elles n’acceptent pas plus de 5V, au risque d’endommager le microcontrôleur. Pour parer ce problème, on peut utiliser un pont diviseur de tension, qui nous permet de ramener la valeur théorique maximale de 7.5V à 5V. On peut donc mesurer une plage plus large, mais en contrepartie on perd un peu en précision (les 1024 valeurs ne correspondent plus à [0, 5] mais à [0, 7.5]).
  + L’Arduino servira donc à récupérer les informations, et les enverra à l’ordinateur via un câble USB. Le PC émulant un port COM (communication série) à la connexion de la carte, il reste simplement à afficher les informations sur l’écran avec la forme la plus adaptée (graphique, en fonction de l’avancement, possibilité de rajouter d’autres formes). L’un des principaux avantages à l’utilisation de ce type de cartes est la synchronisation. Le délai sera réglable via le logiciel (par exemple, si on règle sur une transmission par seconde, toutes les secondes la carte va récupérer les informations au près de la machine et des PD, pour les transmettre). Le choix de type de carte proposé est l’Arduino Nano, à base d’atmel atmega328 (microcontrôleur le plus courant sur les arduinos, puissant et économe en énergie, sans superflu), pour son faible encombrement, son petit prix , et la présence d’un convertisseur d’origine, permettant de communiquer avec la carte à partir du PC, sans avoir à passer par un circuit externe.
  + Enfin, le logiciel développé pour PC devant fonctionner sur Windows, l’utilisation de C++ semble plus adaptée que JAVA (nécessite des installations annexes pour fonctionner), tout en disposant d’une puissance de calcul suffisante. Il devra être simple d’utilisation, contenir une rubrique d’aide pour les utilisateurs, la possibilité d’afficher les informations (sa fonction principale), et de les sauvegarder dans plusieurs formats, dont certains universels.
* La solution proposée permet également de penser à des améliorations futures, par exemple, en connectant un module bluetooth au microcontrôleur, n’importe quel ordinateur, muni du logiciel et d’un dongle bluetooth pourrait utiliser la machine sans fils. Idem pour un smartphone. (même si l’utilité d’un tel appareil en bluetooth n’est pas forcément évidente, elle permet simplement de montrer l’étendue des améliorations possibles).
* Travail effectué :
* Recherche de la tension maximum des photodiodes puis vérification à l’oscillo.
* Recherche et étude des différents composants utilisables pour la réalisation.
* Édition de la liste des composants en annexe.
* Calcul du pont diviseur de tension nécessaire à la protection du microcontrôleur lors de la mesure des photodiodes.
* Repérage des branchements de la prise DB-25 en vue d’un montage expérimental pour la récupération d’infos à base de transistors et résistances.
* Démarrage du code Arduino. Création de la base du programme, avec la fonction de configuration du taux de rafraichissement par envoi série, récupération des informations des PD, et envoi de toutes les valeurs au PC.
* Création des schémas de montage du système.
* Création d’un GIT pour faciliter les accès multiples, et pour simplifier la vérification de l’avancement du projet

Liste du matériel :

