

Conception et réalisation d'un convertisseur de protocoles angulaires

Ing. S. RICHARD,
Ir H. COPPENS d'EECKENBRUGGE
PIERRARD – Virton

Pour repérer des positions angulaires, on utilise des encodeurs reliés à un système de lecture. La communication entre ces deux interlocuteurs exige un protocole de transmission de données, majoritairement de type SSI ou ENDAT, avec une résolution comprise entre 13 et 25 bits. On désire dans ce projet réaliser un adaptateur électronique qui rende compatible des interlocuteurs de marques différentes, avec des protocoles différents et une résolution variable.

Mots-clefs : encodeur, positions angulaires, SSI, ENDAT.

To locate angular positions we use encoders linked to a reading system. The communication between these two interlocutors requires a data transmission protocol, mainly of type SSI or ENDAT, with a resolution between 13 and 25 bits. With this project, we want to make an electronic adapter which makes interlocutors of different brands, with different protocols and a variable resolution compatible.

Keywords : encoder, angular positions, SSI, ENDAT.

1. Présentation du travail proposé

Le but du travail est de construire un convertisseur de protocole pour différents encodeurs absolus.

Ces encodeurs sont utilisés chez HITEC sur les antennes paraboliques pour mesurer leur position. Il y en a généralement trois par antenne, un pour chaque axe de rotation.

Chaque marque utilise un protocole de transmission de données entre ses encodeurs et systèmes de lecture. Ces systèmes sont principalement des automates ou autre interface d'acquisition de données propre à chaque marque.

Le problème est que si l'on possède un encodeur d'un type, on est obligé d'avoir le système de lecture correspondant. Par contre, si l'on veut modifier l'un ou l'autre pour des raisons quelconques, il faut changer les deux.

Grâce à ce convertisseur, on pourra connecter plusieurs types d'encodeur sur le système de contrôle ou sur plusieurs types de système de contrôle.

Les protocoles de transmission de données des encodeurs absolus sont majoritairement de type SSI (Serial Synchronous interface) ou ENDAT (Encodeur Data) et la précision est variable de 13 à 25 bits.

Il va falloir concevoir un circuit électronique qui permet de recevoir les données venant de l'encodeur. Il faudra analyser et convertir ces données et ensuite les renvoyer au bon format et à la bonne vitesse. L'appareil doit en plus avoir une interface utilisateur avec afficheur à cristaux liquides, un clavier et une liaison RS232 pour interfaçage avec un ordinateur (figure 1).

Au départ du travail, HITEC a proposé un encodeur Siemens SSI 25 bits. Sur cette base, a été conçu et réalisé un prototype électronique qui lit les 25 bits transmis en série venant de l'encodeur. Ce circuit transfère les 25 bits série à un microcontrôleur Intel 8051 via un registre tampon et un port parallèle 8bits. Il faudra quatre transferts pour récupérer les 25 bits.

Le prototype de ce premier circuit a été réalisé en technique wire-wrap et on a utilisé un terminal déjà réalisé par HITEC. Ce terminal est un système complet qui permet d'utiliser le microcontrôleur avec toutes ses

potentialités. Il comprend de la mémoire ROM, RAM, un circuit pour port série, des boutons de communication, un afficheur LCD et des entrées sorties adressables.

Pour programmer le microcontrôleur, c'est le programme VISUAL C++ qui a été utilisé et compilé avec le compilateur C51. Ce compilateur permet de convertir le langage C en langage d'assemblage.

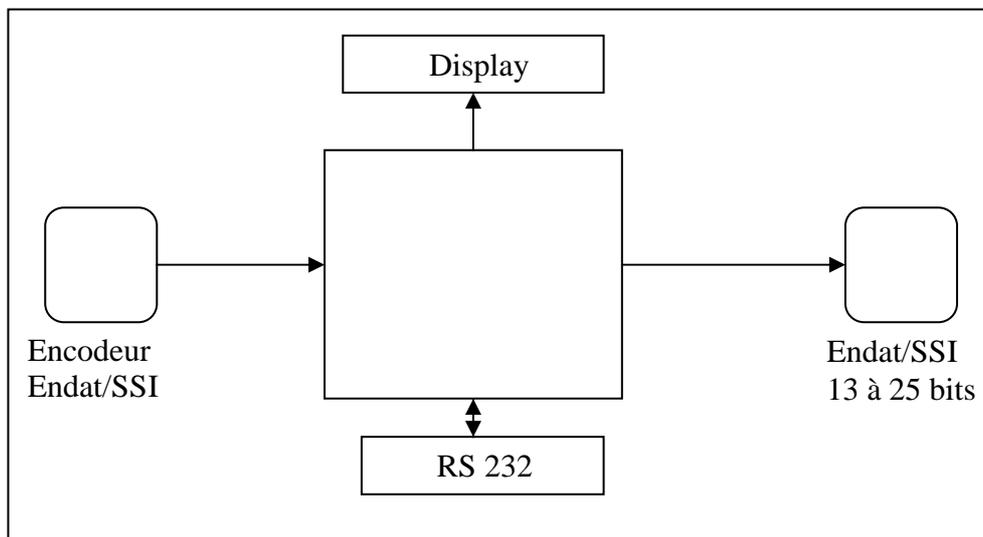


Figure 1 : Convertisseur de protocoles : schéma général

2. Description des différents protocoles de transmissions

2.1 Le protocole SSI (serial synchronous interface)

Le principe du protocole SSI est que le système de lecture envoie le signal d'horloge à l'encodeur à une fréquence de 125 kHz et qu'à chaque coup, l'encodeur lui transmet un bit de données. Une fois toute la trame transmise, l'horloge laisse un temps d'arrêt puis envoie la trame suivante. Les données sont transmises en binaire pur ou en code GRAY sur certains modèles.

L'avantage de code GRAY est d'avoir une meilleure immunité aux bruits parce qu'on ne change qu'un bit à la fois lorsqu'on passe au chiffre suivant.

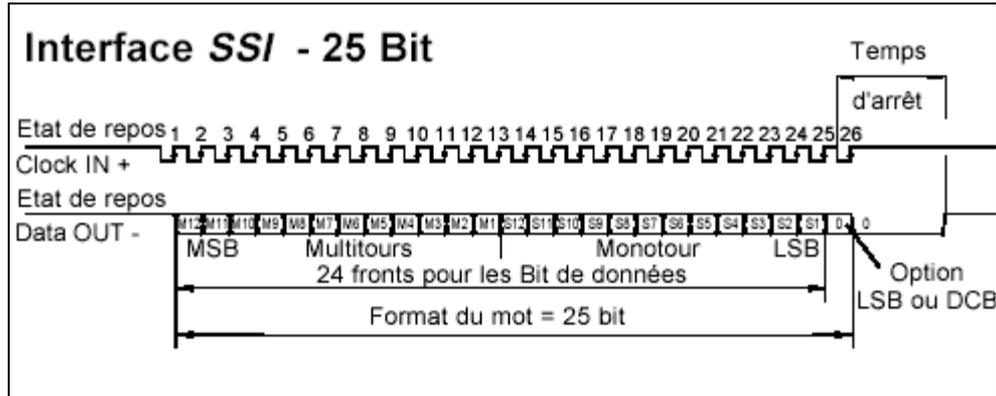


Figure 2 : Protocole SSI

Cet encodeur a une résolution de 4096 positions (12bits) par tour et peut mesurer 4096 tours (12bits). Cela fait en tout 24 bits plus 1 bit dont on ne tient pas compte. Ce dernier bit est un bit de parité qui est présent sur certains encodeurs.

Les données d'horloge et de position sont transmises en mode différentiel suivant le protocole RS 485 entre l'encodeur et l'interface de lecture.

2.2 Le protocole ENDAT (encoder data)

Interface ENDAT

Par rapport au protocole SSI, l'électronique de mesure envoie également des modes de commande à l'encodeur pour choisir si des valeurs de position ou des paramètres doivent être transmis.

Transfert de données

Deux types de transferts de données sont possibles, transfert de valeur de position et transfert de paramètre.

Cycles opératoires pour le transfert des valeurs de position

Le signal d'horloge est transmis par l'électronique de mesure pour synchroniser l'échange de données avec l'encodeur. Après deux impulsions

d'horloge ($2T$), l'électronique de mesure envoie à l'encodeur le mode commande pour transmettre la valeur de position.

Après que l'encodeur ait accompli le calcul de la valeur absolue de position (T_{cal}), il transmet le bit de départ à l'électronique de mesure.

Le bit suivant, d'alarme, est un signal commun pour toutes les fonctions surveillées et sert à la surveillance d'échec. Il devient actif s'il y a un défaut de fonctionnement dans l'encodeur qui pourrait avoir comme conséquence des valeurs incorrectes de position. La cause exacte de l'alarme est sauvée dans la zone de mémoire operating-status où elle peut être interrogée par lecture de cette zone.

La valeur absolue de position est alors transmise en commençant par LSB (bit le moins significatif). La longueur dépend de la résolution de l'encodeur. Elle est sauvée dans la zone mémoire du fabricant d'encodeur. On conclut la transmission de données avec le contrôle par redondance cyclique (CRC).

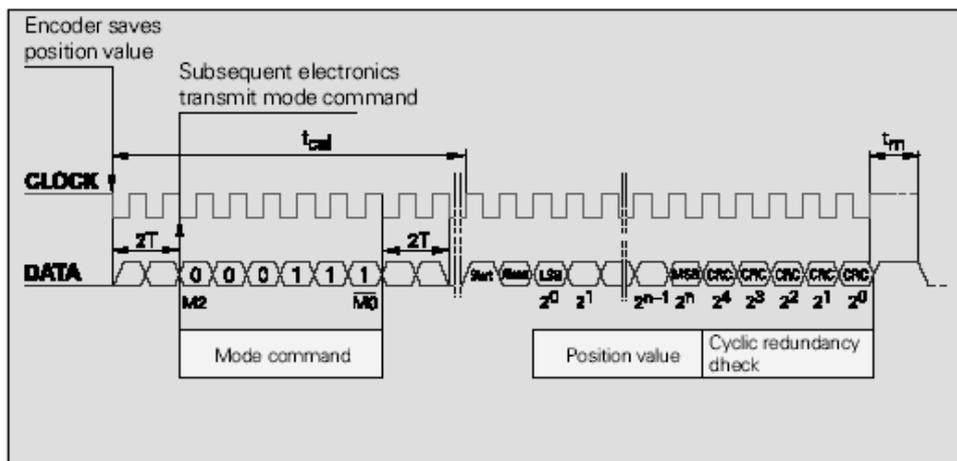


Figure 3 : Protocole ENDAT

Horloge interrompue

À la fin du message contenant des données, le signal d'horloge est placé au niveau haut. Après le temps T_m (10 à 30 μ s) qui est le temps d'interruption

entre deux positions, il y a retour de la ligne de données à l'état bas et ensuite, commence une nouvelle transmission suivant le signal d'horloge.

3. Conception du circuit pour le format SSI:

3.1 Principe général du convertisseur

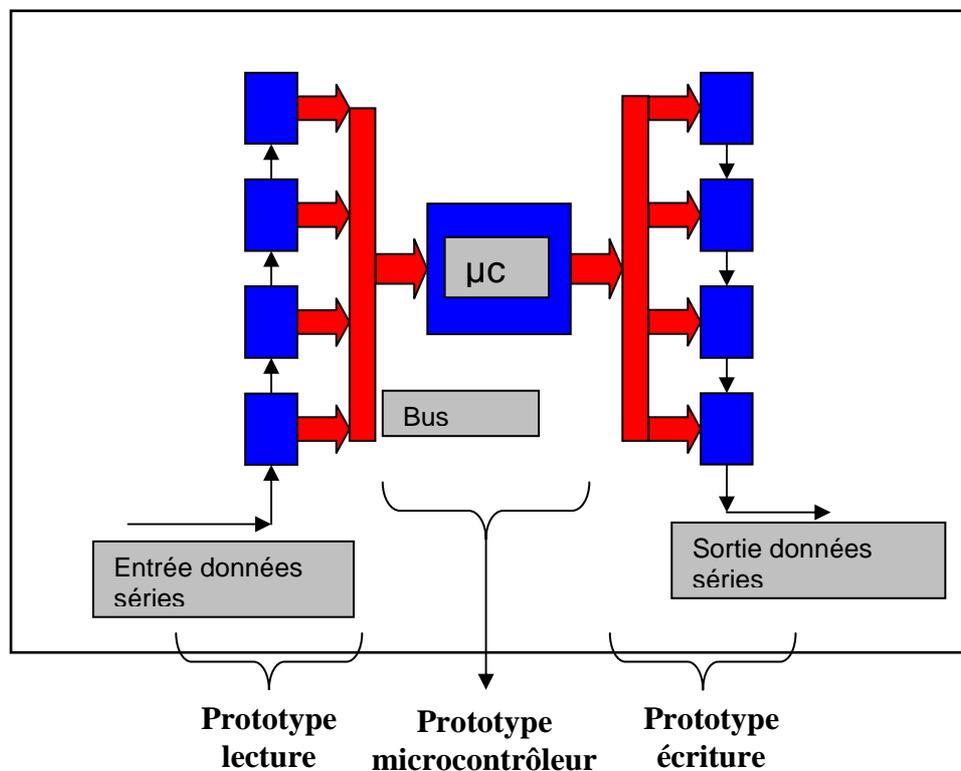


Figure 4 : Principe général

Pour réaliser ce convertisseur, on utilise un microcontrôleur 8 bits. Ce microcontrôleur devra pouvoir lire les données venant de l'encodeur, ensuite décoder la position de ce dernier. Une fois la position de l'encodeur calculée, le microcontrôleur recalculera la position suivant le format de transmission choisi et transfèrera celle-ci à l'interface de lecture. Ce convertisseur sera totalement transparent pour le système de lecture.

Pour pouvoir transformer les données séries en parallèles, on utilise des registres à décalages. Ce sont des registres 8 bits et on en raccorde 4 pour avoir au total 32 bits en entrée. On en utilise aussi quatre en sortie.

Il est certain que le microcontrôleur pourrait faire seul ces opérations. Cependant, l'électronique permet de gagner le temps de passage de série à parallèle ou de parallèle à série.

3.2 Description du prototype de lecture

Ce prototype permet de lire des valeurs de position d'un encodeur SSI 25 bits.

Le circuit est constitué en partie de quatre registres 8 bits à décalages et d'un générateur d'horloge. Pour réaliser une lecture de position, on envoie 25 impulsions d'horloge à l'encodeur et on reçoit en synchronisme les données. Pour commander l'envoi et l'arrêt de l'horloge, une bascule est actionnée par le programme.

On compte le nombre de coups d'horloge que l'on envoie grâce à un temporisateur compteur du microcontrôleur. Ce dernier est programmé en mode compteur, et lorsqu'il arrive à 25, il génère une interruption. Cette interruption provoque la lecture des registres par le microcontrôleur.

3.3 Réalisation du prototype d'envoi de données

Le circuit sera capable de recevoir les données venant du microcontrôleur, 8 bits par 8, et de les envoyer au format SSI 25 bits suivant le protocole. Ce prototype devra être transparent pour l'interface de lecture et donc quand on raccordera le prototype d'envoi de données à l'interface de lecture, le prototype devra réagir comme si on avait un encodeur raccordé.

Conception du circuit

Il va falloir réaliser un circuit qui transforme des données 8 bits parallèles en données 25 bits séries.

Ce circuit devra, par l'intermédiaire de registres à décalage, mémoriser les bits de données venant du microcontrôleur et, à chaque coup d'horloge envoyé par le système de lecture, les faire sortir l'un derrière l'autre du 1^{er} au 25^{ème}. Pendant le décalage des registres, la valeur de la position suivante

est calculée et transférée du μc dans chaque latch de chaque registre. Ceci permet d'avoir un temps très court entre deux transmissions.

Dès qu'il y aura une impulsion sur le signal d'horloge d'une longueur différente, ce qui correspondra au temps recherché pour la synchronisation, on aura en sortie une impulsion qui commandera les registres. Cette commande permettra de transférer les données qui sont dans les bascules des registres vers le registre à décalage même (entrée PL des registres). L'entrée du circuit de détection fonctionnera pour un temps d'arrêt à l'état haut ou à l'état bas.

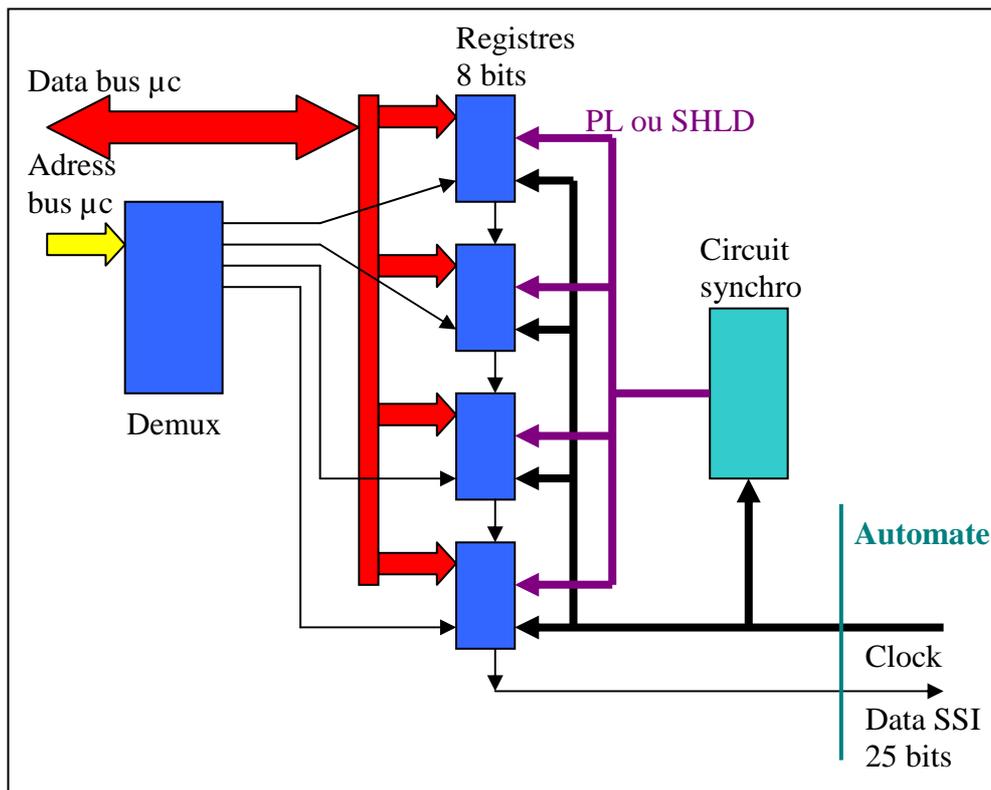


Figure 5 : Schéma de principe du circuit d'écriture de l'encodeur

Le principe de fonctionnement de la carte est que l'on copie les données venant du μc dans les registres (bascules) à n'importe quel moment par rapport à l'horloge. Une fois la position copiée, c'est le circuit de synchronisation qui se charge de transférer celle-ci au bon moment des bascules mémoires dans les registres pour être transmise par décalage juste après.

3.4 Réalisation du prototype microcontrôleur

On va concevoir un circuit qui va remplacer l'utilisation du terminal HITEC. Ce circuit doit être constitué du microcontrôleur, de son alimentation, d'un port de communication avec les deux autres cartes, d'une interface RS232 pour communiquer avec un ordinateur. Il faudra aussi pouvoir choisir le mode de conversion de données manuellement.

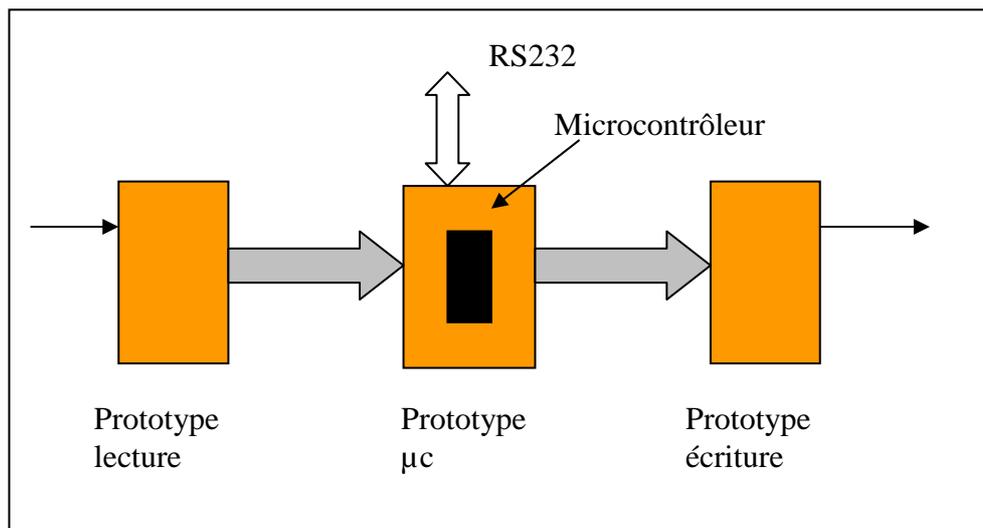


Figure 6 : Raccordement des trois cartes

On a utilisé un microcontrôleur ATMEL AT89C52. En plus du microcontrôleur, il y aura sur le prototype une alimentation, une communication RS 232 et des cavaliers pour faire des choix de conversion.

Programme des trois prototypes ensemble

L'essai réalisé est le raccordement des trois prototypes car ils fonctionnent séparément mais il faut créer un programme pour qu'ils communiquent tous les trois ensemble. Comme pour le test du prototype de lecture et d'écriture avec le terminal, on va faire une boucle de données. On choisit une position avec un ordinateur via la liaison RS 232. Cette position sera convertie au format SSI et envoyée au prototype de lecture. Le prototype de lecture enverra la donnée au microcontrôleur et celui-ci la transfèrera au PC pour l'affichage de la valeur de position. Ce test permettra de voir si le convertisseur SSI 25 bits / SSI 25 bits utilisé fonctionne correctement.

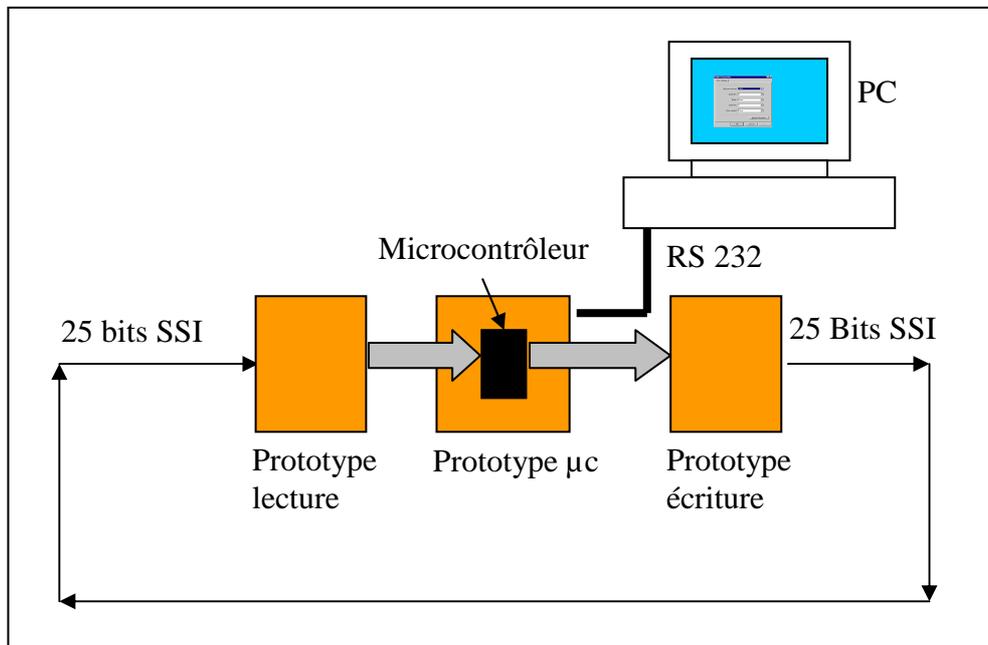


Figure 7 : Essais SSI

4. Modification et conception du circuit pour le format ENDAT

Comme il est expliqué dans la description du protocole ENDAT, cette communication est bidirectionnelle. L'encodeur envoie des données à

l'interface de lecture et celle-ci est capable d'envoyer aussi des données à l'encodeur. Comme le système qui fonctionne pour le format SSI ne possède qu'un sens de communication, de l'encodeur vers le système de lecture, on doit modifier le circuit. Il va falloir revoir le prototype de lecture pour recevoir les valeurs de position de l'encodeur. On devra aussi transformer le prototype d'écriture pour que l'interface de lecture soit capable d'utiliser les données que l'on envoie.

Pour cette partie du travail, on utilise un encodeur SIEMENS ENDAT 13 bits. C'est un encodeur mono tour qui a une résolution de 13 bits, ce qui correspond à une précision de 0.04° .

4.1 Circuit de lecture

Le principe que l'on va utiliser pour envoyer le mode de commande va utiliser une bascule. Pour cette opération, la bascule B2 sera à l'état bas les 5 premières périodes et ensuite à l'état haut les 5 périodes suivantes. Ceci simulera l'envoi du mode de commande à l'encodeur. Pour commander la bascule, on utilise des compteurs qui actionneront la bascule au bon moment.

Toute la partie principale du prototype est restée comme pour le format SSI. Le fonctionnement général est identique, sauf le début d'une lecture de position.

Le microcontrôleur a deux choix pour faire une lecture : soit ENDAT et dans ce cas, il commence par START, soit SSI, et il commence par SET_FF.

La lecture des registres suit le même principe qu'une lecture en SSI sauf que les données arrivent inversées dans les registres (MSB à la place du LSB) et il faudra donc, dans le programme, retourner les données à la lecture des registres.

4.2 Circuit d'écriture

Il s'agit maintenant de simuler un encodeur ENDAT avec le prototype. Pour simuler l'encodeur, on va se limiter au mode transfert de valeurs de position du protocole. En effet, on ne fait que des mesures de position et on

ne s'occupe pas des autres possibilités des encodeurs. Une autre raison est que si l'on fait une conversion depuis SSI, il n'y a que le mode transfert de position possible en SSI.

Conception du circuit

Le fonctionnement du circuit sera indépendant de l'interface de lecture. Ceci veut dire que l'interface pourra demander n'importe quoi à l'encodeur, le circuit qui le simule transmettra la valeur de position. On ne va donc pas lire le mode commande demandé, mais simplement envoyer les données de position juste après le temps nécessaire à la lecture du mode de commande. Ainsi, l'interface pourra les utiliser.

Pour transférer une valeur de position, on va attendre les dix premiers coups d'horloge, et ensuite on enverra les données suivant le protocole. Au circuit existant, on ajoute un compteur et une bascule qui permettra de choisir entre SSI ou ENDAT. Ce choix se fait dans le programme de conversion.

Calcul du CRC (Cyclic Redundancy Check)

Afin de détecter des erreurs résultant d'interférences pendant la transmission de données, un code de 5-bit CRC est assigné à chaque mot contenant des données. Le code de CRC est produit par l'électronique dans l'encodeur. Les différentes erreurs de transmission de données peuvent être détectées au moyen du CRC.

4.3 Raccordement convertisseur ENDAT / ENDAT

Configuration de test

Pour tester nos deux nouveaux prototypes, on refait le test en boucle.

Après plusieurs essais et modifications au niveau du programme, le circuit fonctionne correctement. Il est capable d'envoyer et recevoir les données en protocole ENDAT. Il est aussi capable d'envoyer et recevoir les données en SSI sans modification hardware.

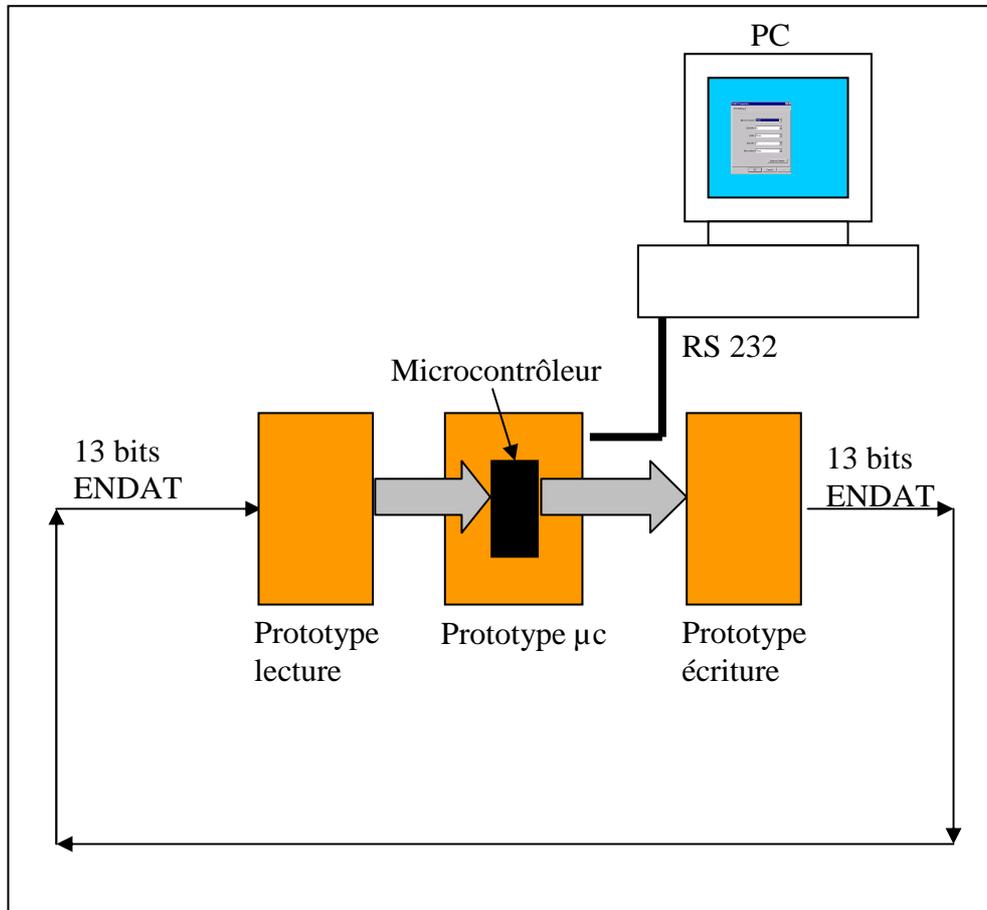


Figure 8 : Essais ENDAT

5. Convertisseur ENDAT/SSI

Pour réaliser une conversion, on utilise l'encodeur ENDAT 13 bits et l'interface de lecture sera un automate SIEMENS.

A l'automate, un module de positionnement est raccordé. Ce module est une interface d'axe qui est capable de faire de l'asservissement de position. Il a une entrée pour encodeur SSI et une sortie qui commande un variateur de

vitesse. Ce module permet de calculer la commande à envoyer au variateur en fonction de la consigne et de la position actuelle mesurée par l'encodeur.

Dans le cas considéré, on utilise l'API et le module d'asservissement pour relire les positions de l'encodeur, sans faire de boucle de régulation. Ce test permettra de voir si le circuit est capable de réaliser une conversion de protocole.

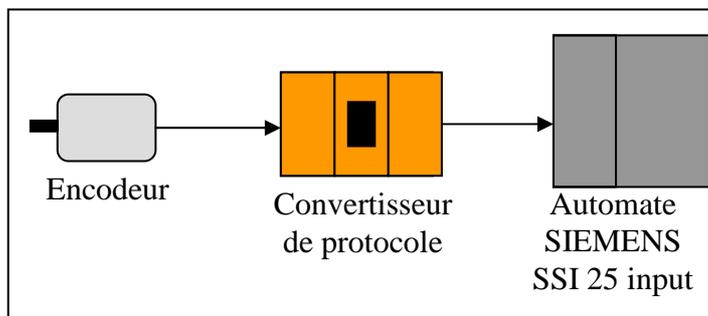
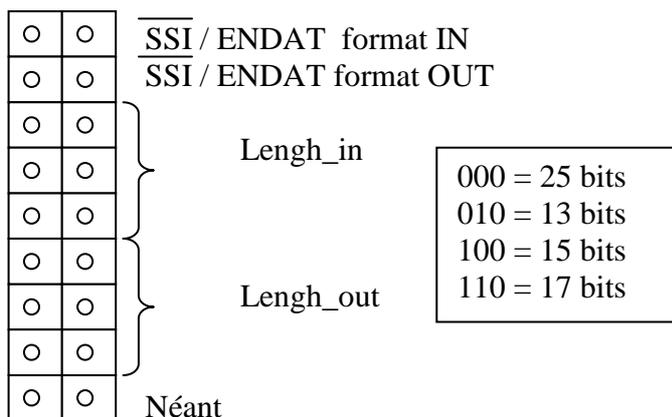


Figure 9 : Schéma de la conversion

Utilisation du convertisseur

Pour faire les choix de conversion, on utilise les petits cavaliers sur le prototype microcontrôleur. On fera des choix de protocoles en entrée et en sortie. Il faudra aussi sélectionner le nombre de bits des transmissions de position.



Pour le choix des longueurs, il n'y a que quatre possibilités utilisées sur les 8 disponibles.

6. Conclusions

Le but du travail était de pouvoir convertir des protocoles de transmission de données d'encodeur angulaire. Pour convertir ces données, il a fallu concevoir un circuit électronique capable de lire les données venant de l'encodeur et capable de les renvoyer à l'interface de lecture. Ce circuit devait être transparent pour l'interface de lecture, et donc, réagir comme un encodeur.

Le circuit conçu et réalisé répond aux attentes. Il est capable de recevoir et d'envoyer des données au format SSI. Il est aussi capable de lire et d'envoyer des données au format ENDAT. Le circuit permet d'utiliser les deux protocoles, lire SSI et envoyer ENDAT, mais également lire ENDAT et envoyer SSI.

Si le circuit est opérationnel pour les deux formats, il l'est aussi pour les différentes longueurs de données. Au niveau de la longueur dans les protocoles, seul le nombre de bits de données change, mais le principe de transmission reste toujours le même et donc, le circuit est capable de fonctionner avec des longueurs de 13, 15, 17, 19, 25 bits pour ENDAT et SSI.

Dans le programme réalisé, toutes les longueurs de transmission ne sont pas implémentées. La mémoire du microcontrôleur étant de 8 Ko, on a été limité dans la taille du programme. Dans un premier temps, on n'a implémenté que les formats directement intéressants. Par après, il faudra utiliser un microcontrôleur avec plus de mémoire comme le ATMEL AT89C55WD qui possède 20 Ko de mémoire. Ce nouveau microcontrôleur sera parfaitement capable de couvrir toutes les longueurs de format.

Le circuit possède aussi une communication série ce qui lui permet d'être interfaçable avec un PC pour une utilisation quelconque.

L'objet principal de ce prototype est son utilisation pour un futur projet chez HITEC. Les interfaces de contrôle de position de l'antenne ont une entrée pour encodeur codée sur 19 bits, or la précision de 17, 15 voire 13 bits, est suffisante.

Donc si l'on peut connecter un encodeur de moindre précision, et donc de moindre coût, à l'interface de positionnement, l'utilisation du convertisseur est justifiée. Sachant qu'un encodeur 19 bits coûte aux alentours de 2500 € et un encodeur 13 bits environ 250 €, il est fondé d'utiliser ce convertisseur.

Ayant réalisé le prototype complet en wire-wrap et sachant que le prix d'un circuit imprimé est très variable suivant la taille du circuit et le nombre à produire, il n'a pas été possible de calculer le prix total du convertisseur. Mais par approximation, le convertisseur ne devrait pas coûter plus de 250 € et l'utilisation reste toujours justifiable.

7. Références bibliographiques

www.hitec.lu

www.philips.com

www.ti.com (Texas Instruments)

www.siemens.com

www.machinetoolinspection.com (HEIDENHAIN CORPORATION)

The Linear Control Circuits Data Book, Texas Instruments.

Microsoft Visual C++ Programmer's Guide, Microsoft Corporation.

ODANT, Bernard, *Microcontrôleur 8051 et 8052, Description et mise en œuvre*, DUNOD.