

17 août 2017

Trouver le bon moyen de communication grâce à un protocole adapté

Pour que deux entités puissent communiquer, deux choses sont primordiales. Il faut tout d'abord que le vecteur de transmission soit adapté. Par exemple, il est plus judicieux pour deux personnes séparées par une vitre et désireuses de se saluer de se faire un signe de la main (lumière) plutôt que de se parler (son). Ensuite, il faut que les deux entités spécifient des règles qui permettent la communication. Par exemple, les interlocuteurs doivent parler la même langue, s'exprimer à tour de rôle, une question attend généralement une réponse, etc.

En informatique et microélectronique, les vecteurs de transmissions sont variés (câble, ondes radio, fibre optique, etc.) et la spécification des règles de communication est appelée un protocole.

Comment choisir un protocole ?

Les ingénieurs, lors du développement d'un système de communication pour une application spécifique, doivent prêter une attention particulière aux points suivants :

- **Débit:** Cette valeur définit la quantité de données (bits) envoyée par seconde.
- **Latence :** La latence représente le temps qui s'écoule entre une demande et une réponse. Cette quantité est importante pour définir la réactivité d'un système.
- **Intégrité :** Cette notion permet de qualifier un message et ses éventuelles altérations durant le transfert d'un paquet.
- **Sécurité :** Ce facteur tient compte de l'importance de la confidentialité des données transmises dans un message.

Ces différents points vont favoriser une architecture matérielle sérielle ou parallèle mais également le choix du protocole. Dans le cas TCP/UDP par exemple, UDP ne garantit pas l'intégrité du message (il n'y a pas d'acquittements et la somme de contrôle n'est pas obligatoire en IPv4) mais le débit et la latence sont bonnes ce qui fait de l'UDP un bon protocole pour des solutions où la perte de quelques paquets n'est pas importante (voix sur IP par exemple). Comme autre exemple, dans le cas de transmissions de vidéos, le débit influence le choix du protocole et du vecteur de transmission. En effet, le SDI (Serial Digital Interface) utilise des câbles coaxiaux (guides d'onde) pour transmettre des données jusqu'à 2,97 Gbit/s.

Les protocoles de communication entre microcontrôleurs

Le développement d'un système électronique embarqué et autonome est souvent dépendant des capacités de communication limitées des microcontrôleurs ainsi que des exigences spécifiques à ce type d'application:

Débit	Non critique	En principe les données sont de faible taille (Commandes, mesures de capteurs, etc.).
Latence	Critique	Une commande doit généralement être traitée le plus rapidement possible.
Intégrité	Critique	Le message doit être correctement transmis sous peine de graves conséquences. (Capteur dans un système de sécurité par exemple).
Sécurité	Non critique	Les données transmises ne sont en général pas sensibles (peu intéressantes et difficilement piratables) et ne nécessitent donc pas d'être protégées.
Puissance CPU	Critique	La puissance de calcul des microcontrôleurs est limitée.

1/3

Afin de respecter ces exigences, les ingénieurs de FiveCo travaillent très fréquemment avec des bus à deux lignes I²C (Inter-IC bus) qui est la liaison sérielle la plus connue entre les microcontrôleurs et les systèmes électroniques.

Cependant, lorsqu'il s'agit de communiquer entre des systèmes qui ne sont pas sur la même carte électronique, des considérations supplémentaires sont à prendre en compte et l'I²C n'est plus adapté. Les lignes de communications doivent être protégées contre les décharges électrostatiques (ESD). Les capacités parasites le long des lignes peuvent affecter le bus et le taux d'erreur lors de transmissions peut alors devenir inacceptable.

Pour des applications de systèmes embarqués qui sont interconnectées sur des distances importantes et pour lesquelles des infrastructures réseaux (réseaux TCP/IP) ne sont pas nécessaires, comme un groupe de modules synchronisées ou comme des périphériques qui se connectent ponctuellement à un système central, les ingénieurs de FiveCo ont développé un protocole dédié, le **RAP** qui peut fonctionner sur différents bus physiques comme, par exemple, des lignes en mode différentielles.

Le RAP « Register Access Protocol »

Le RAP a été développé dans le but d'accéder à des registres en lecture ou en écriture sur des microcontrôleurs distants. Il est utilisable sur plusieurs bus comme l'UART, le RS485, le RS422 et l'USB. Il peut également être utilisé en surcouche d'autres protocoles (IP, etc...).

Le protocole se base sur le principe d'une requête suivie d'une réponse éventuelle. Les requêtes et réponses sont composées de la structure de trame suivante :

1 Byte Adresse de destination	1 Byte Adresse de la source	N Bytes Taille des Data qui vont suivre	x Byte DATA
---	---------------------------------------	---	-----------------------

Les données (data) sont une série de messages qui peuvent être: des demandes/réponses de lecture, des demandes d'écriture, une demande d'acquiescement, une demande de routage, des messages d'erreurs, un message de fin de trame et une somme de contrôle.

Grâce à ces fonctionnalités, le protocole permet donc de :

- Faire de la communication point-à-point mais également multipoints (système d'adressage)
- Identifier l'origine et la destination de chaque requête.
- Accéder en lecture et en écriture aux registres accessibles de microcontrôleur depuis une interface sérielle (RS485, RS422, UART)
- Garantir l'intégrité et la réception des messages grâce à une somme de contrôle et un système d'acquiescement (y compris gestion des éventuelles collisions).
- Mettre à jour le code (firmware)
- Router un message sur une autre interface du périphérique.

Ces caractéristiques sont extrêmement intéressantes dans le cas de communications bas-niveau entre des systèmes distants. Il permet aux ingénieurs de développer et de garantir à leurs clients des systèmes interconnectés robustes, fiables et très réactifs.

En conclusion, le choix et la bonne mise en application de protocoles sont primordiaux dans le design de systèmes interconnectés. Cette problématique est traitée depuis de nombreuses années par FiveCo dans ses divers développements.

Philippe Dreyer
Firmware Engineer