Protocole de communication du microcontrôleur – Eurobotruck 2018

# Généralités

La communication avec le microcontrôleur responsable du contrôle du robot se fait par l’intermédiaire d’une liaison Ethernet sur laquelle est utilisé le protocole TCP/IP.

Le microcontrôleur agit comme un serveur, il peut accepter jusqu’à 4 connexions simultanées.

Un serveur identique, utilisant le même protocole et permettant d’effectuer les mêmes actions écoute également sur le port série RS232 du microcontrôleur. Aucun mécanisme de contrôle d’intégrité de l’information, de correction d’erreurs, ni d’acquittement n’est mis en place dans le protocole que nous définissons ici. La liaison série étant un support n’offrant aucune garantie de ce type, cette dernière présente un moins bon niveau de fiabilité et n’est à utiliser que si la liaison Ethernet est défaillante.

# La trame

La communication entre le client et le serveur est asynchrone et se compose d’une succession de trames, formatées comme suit :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0xFF | ID | Length | Data 0 | … | Data n |

*Une case = un octet*

**0xFF** : identificateur de début de trame. Simplement un octet contenant la valeur 0xFF.

**ID** *[0x00 – 0xFF]* : identifiant de trame. Octet indiquant quelle commande doit être effectuée par le microcontrôleur, ou à l’inverse quelle commande vient d’être effectuée par le microcontrôleur.

**Length** *[0x00 – 0xFE]* : taille du champ *Data*, en nombre d’octets. La taille totale de la trame vaut donc *Length* + 3. Le champ *Data* peut contenir, au maximum, 254 octets.

**Data** : ensemble d’octets contenants les informations nécessaires à l’exécution de la commande. La signification de ces octets est spécifiée plus loin, pour chaque commande.

**Cas particulier : la trame d’information**

La trame d’information est une trame contenant une chaine de caractères destinée à être lue par un opérateur humain. Elle sera ignorée par le serveur si il en reçois une. Elle suit le même schéma que la trame normale, à deux détails près : Length vaut 0xFF (255) et il n’y a pas de limite de taille pour la trame (la fin de la trame est donnée par le caractère de fin de chaine).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0xFF | ID | 0xFF | Char 0 | … | Char n | End of string |

**ID** *[0x00 – 0x1F]* : il correspond nécessairement à celui d’un canal de données, et jamais à celui d’un ordre (voir chapitre suivant pour la définition de ces notions).

**Char** : succession de caractères ASCII destinés à être affichés, il s’agit du code ASCII NON étendu, ce qui interdit l’usage de caractères accentués.

Les caractères autorisés sont : \0 \t \n \r (space) ! “ # $ % & ‘ ( ) \* + , - . / 0123456789 : ; < = > ? @ ABCDEFGHIJKLMNOPQRSTUVWXYZ [ \ ] ^ \_ ` abcdefghijklmnopqrstuvwxyz { | } ~

Les valeurs décimales correspondantes sont : 0 ; 9 ; 10 ; 13 ; 32–126

**End of string** *[0x00]* : caractère ASCII « fin de chaine ». Octet valant zéro.

# Les commandes

L’espace des 256 commandes adressables est divisé en 3 parties :

|  |  |
| --- | --- |
| Plage d’ID | Catégorie |
| 0 – 31 | Inscription aux canaux de données |
| 32 – 127 | Ordres à exécution longue |
| 128 – 255 | Ordres immédiats |

## Inscription aux canaux de données

Les ordres permettent à un client de spécifier au serveur qu’il souhaite, ou non, recevoir par la suite les informations concernant un certain canal de données. Si le client souscrit à un canal de données, il recevra des trames ayant pour ID le numéro du canal de données et contenant des informations donc le format est spécifié plus bas.

Structure de la trame d’inscription/désinscription :

|  |  |  |  |
| --- | --- | --- | --- |
| 0xFF | ID du canal | 0x01 | Inscription |

**ID du canal** : valeur entre 0x00 (0) et 0x1F (31) désignant l’un des canaux de données (listés plus loin).

**Inscription** : 0x00 pour se désinscrire. 0x01 pour s’inscrire.

Aucune réponse ne sera envoyée par le serveur à la suite de cette commande.

Liste des canaux

|  |  |  |
| --- | --- | --- |
| Identifiant | Désignation | Format des données envoyées sur ce canal |
| 0x00 |  |  |
| 0x01 |  |  |
| 0x02 |  |  |
| 0x03 |  |  |
| 0x04 |  |  |
| 0x05 |  |  |
| 0x06 |  |  |
| … |  |  |
| 0x1F |  |  |

## Ordres à exécution longue

On regroupe ici toutes les commandes dont la réalisation nécessite un temps « long », ce qui correspond, en pratique, à tout ce qui nécessite plus d’une milliseconde. On y trouve donc notamment les commandes mettant en œuvre des composants mécaniques.

L’exécution de l’ordre long commence à la réception de la trame de commande et se termine au bout d’un temps à priori inconnu. Une trame sera envoyée par le serveur à la fin de l’exécution de l’ordre long afin d’en notifier le client. Cette trame utilise l’identifiant de la commande et peut contenir, au besoin, des informations relatives à la qualité d’exécution de la commande. Il est interdit d’exécuter un ordre long si un autre ordre long utilisant le même identifiant (i.e. : le même ordre) est encore en cours d’exécution.

Liste des ordres longs

|  |  |  |  |
| --- | --- | --- | --- |
| Identifiant | Description | Format des données de la commande | Format de données de la réponse |
| 0x20 |  |  |  |
| 0x21 |  |  |  |
| 0x22 |  |  |  |
| 0x23 |  |  |  |
| 0x24 |  |  |  |
| 0x25 |  |  |  |
| 0x26 |  |  |  |
| … |  |  |  |
| 0x7F |  |  |  |

## Ordres immédiats

On regroupe ici les commandes pouvant être exécutées en un temps très court devant les temps de transmission, typiquement toujours inférieur à une milliseconde. On y trouve principalement des commandes permettant de lire ou de modifier des variables d’état du système.

L’exécution d’un ordre immédiat se fait dès la réception de la trame associée, et peut faire, ou non, l’objet d’une réponse à l’aide d’une trame portant le même identifiant.

Liste des ordres immédiats

|  |  |  |  |
| --- | --- | --- | --- |
| Identifiant | Description | Format des données de la commande | Format des données de la réponse |
| 0x80 |  |  |  |
| 0x81 |  |  |  |
| 0x82 |  |  |  |
| 0x83 |  |  |  |
| 0x84 |  |  |  |
| 0x85 |  |  |  |
| 0x86 |  |  |  |
| … |  |  |  |
| 0xFF |  |  |  |