

NOTICE D'UTILISATION DU MICROLOGICIEL Open Bidouille Robot

Sommaire

NOTICE D'UTILISATION DU MICROLOGICIEL Open Bidouille Robot	1
Petite description :	3
1. Protocole de communication OBR	4
1.1. Principes de fonctionnement du protocole OBR.....	4
1.2. Connexion à OBR.....	5
1.3. Utilisation du protocole pour transmettre des commandes	6
1.4. Utilisation du watchdog pour détecter la déconnexion.....	6
2. Configuration d'un système OBR	6
2.1. Connexion en administrateur à OBR.....	6
2.2. Remise à 0 du mot de passe, et Blocage système.....	7
2.3. Utilisation des commandes simples	7
2.4. Utilisation des commandes plus avancées.....	7
2.5. Accès direct à la mémoire de la carte	7
3. Pilotage d'un système OBR	7
3.1. Connexion et initialisation du mode de pilotage	7
3.2. Utilisation des commandes de pilotage depuis un PC	7
3.3. Connexion d'une télécommande OBRoC Arduino	7
4. Configuration d'une télécommande OBRoC (Arduino).....	7
4.1. Connexion à la télécommande depuis le PC	7
4.2. Utilisation des commandes de configuration.....	7
5. Configuration d'une télécommande OBRoC (PC).....	8
5.1. Création d'un fichier d'initialisation .map.....	8
5.2. Prise en main du logiciel OBRoC PC.....	8
5.3. Pilotage d'une télécommande depuis le PC.....	8
5.4. Connecter OBRoC PC sur le robot, en liaison directe.....	8
6. Utilisation du logiciel TeC'OBiR pour la configuration simplifiée.....	8
6.1. Utilisation de Terminal Configuration for OBR.....	8
6.2. Les atouts de TeC'OBiR.....	9
6.3. Utilisation avancée avec TeC'OBiR	9
7. Utilisation des fichiers de configuration automatiques '.map'	9
7.1. Utilité d'un fichier .map.....	9
7.2. Contenu du fichier	9

7.3.	Créer un fichier de configuration pour votre robot	9
8.	Liste des commandes de configuration pour OBR	9
8.1.	Commandes en tant qu'utilisateur.....	9
8.2.	Commandes en tant qu'administrateur	9
9.	Liste des commandes de pilotage pour OBR.....	10
10.	Liste des commandes pour OBRoC Arduino.....	10
11.	Utilisation de la bibliothèque en C pour créer votre propre programme sur PC.....	10

Petite description :

Open Bidouille Robot est un micrologiciel pour arduino, servant à créer des robot, avec moteurs, servomoteurs, entrées analogiques, entrées digitales, et sorties digitales.

Un robot OBR est constitué d'un maître et d'une télécommande.

Le maître peut être épaulé par deux esclaves, pour augmenter le nombre d'entrée/sorties, et la télécommande peut être un arduino tournant sous OBRoC, ou un PC avec 'OBRoC PC'.

L'ensemble forme un réseau, et les cartes peuvent être connectées n'importe où sur le maître ou les esclaves, un système d'adresses et de passerelles permet la communication entre deux cartes, qu'elles soient reliées en direct ou via un autre arduino.

Une fois compilé et installé sur l'arduino, OBR ne sait rien faire. Il faut utiliser un PC pour configurer le robot, c'est-à-dire les E/S, ainsi que les liaisons entre Entrées et Sorties (ex : Un moteur peut tourner seulement si le capteur de fin de course n'est pas enclenché).

Il y a deux méthodes de configuration : manuelle en ligne de code, ou automatique via un fichier (.map).

Il y a deux types de programmes sur PC : un programme de configuration (TeC'OBiR par exemple), et un programme de pilotage (OBRoC PC par exemple).

Le second a besoin d'un fichier de configuration automatique pour fonctionner, car il contient la liste des capteurs, sorties, et autres, mais aussi les boutons à placer sur l'écran et leurs actions.

Les chapitres qui vont suivre vont vous permettre de mettre en fonction votre premier robot OBR.

1. Protocole de communication OBR

1.1. Principes de fonctionnement du protocole OBR

Le protocole OBR est assez strict dans les syntaxes à respecter. Voici une liste des syntaxes propre au protocole :

- Tous les caractères doivent être en MAJUSCULE
- Les commandes doivent commencer par « OBR_ », non obligatoire pour la réponse
- Toutes les commandes doivent contenir l'adresse de l'envoyeur et celui du destinataire
- Aucun espace n'est autorisé dans une commande ou une réponse
- Une commande peut prendre jusqu'à trois paramètres, séparés par des tirets
- Toutes les commandes retournent une réponse à la requête
- Une commande, adresse et paramètres inclus, ne doit pas dépasser 128 caractères
- Toutes les commandes doivent se terminer par une '\n', qui indique la fin de la ligne

La partie adresse se trouve comme ceci : [N° Envoyeur | N° Destinataire]

La partie commande se formule comme ceci : OBR_COMMANDE

Et enfin, la partie paramètres s'écrit comme ceci : -PARAMETRE

Il peut y avoir trois paramètres, ce qui donne ceci :

« [N° Envoyeur | N° Destinataire] OBR_COMMANDE-PARAMETRE1-PARAMETRE2-PARAMETRE3\n ».

Un exemple de commande, avec l'adresse de l'envoyeur à 1 (Le PC), et l'adresse du destinataire à 3 (La Carte OBR Maître). On demande une connexion, avec le mot de passé en paramètre :

« [1|3]OBR_CONNECTER_ADMIN-2560\n ». Réponse : « [3|1]OK\n »

Concernant les périphériques du réseau, il peut y avoir jusqu'à un maître, deux esclaves, un PC et une télécommande.

Le maître contrôle les esclaves, la télécommande contrôle le maître et les esclaves, et enfin, le PC a la main sur tout le monde.

Il peut y avoir jusqu'à deux esclaves, qui permettent d'augmenter le nombre d'E/S.

Les cartes Maître, esclave1 et esclave2 sont définis à la compilation : un Maître ne peut pas devenir esclave, et vice-versa.

ATTENTION : L'esclave1 et l'esclave2 sont définis à la compilation, et sont distincts : un esclave1 ne peut pas devenir un esclave2, et il ne peut pas y avoir deux esclave1 (ou esclave2) connectés.

Enfin, toutes les connexions se font via les ports série des cartes Arduino.

Les adresses sont :

- 1 = PC
- 2 = Télécommande
- 3 = Maître
- 4 = Esclave1
- 5 = Esclave2

1.2. Connexion à OBR

Pour se connecter à un arduino tournant sous Open Bidouille Robot, il faut savoir que la vitesse de communication série est réglé sur 57600 bauds, et n'est pas modifiable.

Dans le protocole OBR, c'est le maître qui attend les connexions, et les autres qui envoient des commandes de connexion.

Cependant, il est tout à fait possible de nous connecter à un esclave, ou une télécommande.

Exceptionnellement, la commande de demande de connexion ne nécessite aucune adresse, mais la réponse en contient une.

Il faut donc demander à se connecter, sans donner d'adresse car on ne sait pas encore à qui nous nous connectons, puis récupérer l'adresse contenue dans la réponse.

En effet, lorsque nous sommes connectés, la réponse envoyée contient l'adresse de l'envoyeur, nous savons donc maintenant à qui nous sommes connectés, et nous pouvons désormais ajouter les adresses aux commandes.

Lorsque nous sommes connecté à un esclave ou à la télécommande, et que le maître fait partie du réseau, si nous envoyons un message à ce dernier, il sera retransmis et acheminé jusqu'au maître. Pour cela, le système OBR prend en compte les connexions de toutes les cartes, et utilise le système de passerelle.

Exemple, le PC est connecté à la télécommande, qui est connectée à l'esclave2 qui lui-même est connecté à l'esclave1, relié au maître.

(PC <-> Télécommande <-> Esclave2 <-> Esclave1 <-> Maître)

Comme ceci, si le PC veut communiquer avec le maître, le message sera lu par la télécommande, il extrait l'adresse, et sait que le maître est connecté à l'esclave2. La télécommande transmet donc le message sur le port série de l'esclave2, en laissant l'adresse du maître. L'esclave2 fait de même vers l'esclave1, pour enfin arriver sur le maître.

Le maître ayant l'adresse de l'envoyeur, (ici le PC) retourne la réponse en l'envoyant sur le port série de l'esclave1, sur lequel le PC est connecté (indirectement).

Pour se connecter, il faut donc envoyer une commande de connexion, puis récupérer l'adresse de la réponse.

Les commandes, selon qui nous sommes :

- OBR_CONNECTER_PC
- OBR_CONNECTER_TELECOMMANDE
- OBR_CONNECTER_ESCLAVE1
- OBR_CONNECTER_ESCLAVE2

La réponse sera du type : « [AdresseDeCeluiàQuiNousSommesConnecté | NotreAdresse]OK\n »

Nous savons, en extrayant l'adresse, sur quelle carte nous sommes connectés.

ATTENTION : A partir de ce moment-là, il ne faut pas oublier d'insérer les adresses au début des commandes. Sinon, la commande ne sera tout simplement pas lue par le destinataire, et non transmise via les passerelles.

1.3. Utilisation du protocole pour transmettre des commandes

Une fois que nous sommes connectés au robot (via un esclave ou non), nous pouvons envoyer des commandes à celui qu'on veut, tant que le destinataire choisi n'est pas déconnecté.

Si le destinataire n'est pas connecté, ou que l'adresse est mauvaise, aucune réponse n'est envoyée.

Pour transmettre une commande à un périphérique, il suffit d'indiquer son adresse pour que le message arrive à bon port, et sa propre adresse pour avoir une réponse.

Il est inutile de se préoccuper du réseau, grâce au système de routage, la commande peut passer par plusieurs arduinos sans problème.

Il ne faut surtout pas oublier de commencer par l'adresse, puis la commande, et les paramètres, et enfin le retour à la ligne ('\n').

Enfin, toutes les commandes retournent quelque chose, même si ce n'est qu'un « OK ».

1.4. Utilisation du watchdog pour détecter la déconnexion

Le WatchDog présent sur OBR est relativement simple : S'il n'y a pas eu de message (même vide) depuis 5 secondes venant d'un périphérique, il le déconnecte.

Pour cela, le protocole OBR 0.4 inclus un système qui vérifie le délai depuis l'envoi d'une commande, et transmet un message vide contenant uniquement une adresse, si ça fait plus de deux secondes qu'on a rien envoyé.

Si le PC se déconnecte (manuellement ou via le WatchDog), l'administrateur est automatiquement déconnecté, laissant place au simple utilisateur.

2. Configuration d'un système OBR

2.1. Connexion en administrateur à OBR

Une fois connecté au maître ou à un esclave, on peut le configurer et lui envoyer des commandes. Cependant, certaines commandes ont un accès limité, qui les empêchent de s'exécuter en tant que simple utilisateur.

Pour cela, OBR contient une gestion d'administration, qui nécessite un mot de passe, et sécurise l'accès aux utilisateurs qui n'ont pas le mot de passe de connexion. Cela évite la modification du robot par une personne non autorisée.

En administrateur, nous pouvons par exemple ajouter et retirer des objets (moteurs, capteurs, etc...), accéder à l'EEPROM directement (mais des cases sont verrouillées par le système), déconnecter manuellement un périphérique, et plein d'autres choses.

En d'autres termes, l'utilisateur n'a pas accès à grand-chose, si ce n'est le nom, la version et d'autres petits trucs non modifiables, du robot.

Lorsque l'administrateur est connecté, la LED 13 s'allume pour indiquer que l'accès est ouvert à tout le monde. Lorsque l'administrateur se déconnecte, elle s'éteint.

Enfin, pour se connecter en administrateur, on utilise la commande « OBR_CONNECTER_ADMIN-[CODE] ». Pour se déconnecter : « OBR_DECONNECTER_ADMIN ».

2.2. Remise à 0 du mot de passe, et Blocage système

Lorsque le mot de passe de connexion est perdu, ou ne fonctionne plus (EEPROM modifié par un autre programme), il est possible de le remettre à « 2560 », le nom de l'atmega présent sur le MEGA.

Pour cela, il faut relier les PIN 12 et 13 au démarrage du système, pour indiquer que nous réinitialisons le mot de passe par défaut.

Une fois réinitialisé, le système se bloque pour obliger un redémarrage de la carte (éviter les incohérences en RAM).

Concernant le blocage système, lorsque que ça se produit, toutes les sorties sont arrêtées, et la LED 13 clignote indéfiniment.

2.3. Utilisation des commandes simples

2.4. Utilisation des commandes plus avancées

2.5. Accès direct à la mémoire de la carte

3. Pilotage d'un système OBR

3.1. Connexion et initialisation du mode de pilotage

3.2. Utilisation des commandes de pilotage depuis un PC

3.3. Connexion d'une télécommande OBRoC Arduino

4. Configuration d'une télécommande OBRoC (Arduino)

4.1. Connexion à la télécommande depuis le PC

4.2. Utilisation des commandes de configuration

5. Configuration d'une télécommande OBRoC (PC)

5.1. Création d'un fichier d'initialisation .map

5.2. Prise en main du logiciel OBRoC PC

5.3. Pilotage d'une télécommande depuis le PC

Le robot ne prend en compte qu'un périphérique d'une même famille. C'est-à-dire qu'il ne prend qu'une télécommande à la fois, il n'y en aura jamais deux en même temps de connectées. Cependant, le PC peut se faire passer pour une télécommande, et se connecter à OBRoC Arduino pour le piloter.

« PC <-> Télécommande Arduino <-> Robot »

De cette manière, le PC peut voir en temps réel ce qu'il se passe, tout en laissant la main à la télécommande arduino.

ATTENTION : Le PC peut lui aussi interagir avec le Robot, il y a donc des risques de conflits entre les commandes. Dans ce cas, c'est le PC qui est prioritaire, et la télécommande ne peut plus envoyer de commande tant que le PC n'a pas rendu la main (OBRoC Arduino continuera d'afficher les infos si dispo (LED, Ecran Digital...), mais ne réagira plus aux contrôles de l'utilisateur).

5.4. Connecter OBRoC PC sur le robot, en liaison directe

Il est possible de faire passer le PC pour une télécommande, et ainsi contrôler le robot directement depuis le PC (via OBRoC PC par exemple).

Cependant, le robot ne prend en compte qu'un périphérique d'une même famille. C'est-à-dire qu'il ne prend qu'une télécommande à la fois, il n'y en aura jamais deux en même temps de connectées.

S'il y a déjà une télécommande de connectée, le PC ne sera pas pris en compte, il n'y a aucune priorité au niveau des connexions.

6. Utilisation du logiciel TeC'OBiR pour la configuration simplifiée

6.1. Utilisation de Terminal Configuration for OBR

TeC'OBiR est très adapté à l'utilisation avec OBR, car la norme OBR est intégrée au programme. Ainsi, le logiciel se connecte tout seul dès qu'il trouve un arduino sous OBR, et se connecte en administrateur si possible, juste après la connexion.

De plus, le WatchDog est automatiquement désactivé pour éviter les déconnexions intempestives, qu'on peut trouver sur les autres consoles séries.

Pour fermer le programme, il suffit de taper la commande « **exit** », qui va réactiver le WatchDog sur la carte, se déconnecter de l'administrateur, fermer la connexion série, et fermer le programme proprement.

6.2. Les atouts de TeC'OBiR

Contrairement à une console série classique, TeC'OBiR simplifie énormément la tâche en mettant en norme la commande qu'on tape :

- Tous les caractères tapés sont mis en majuscules
- Tous les espaces sont remplacés par des tirets
- L'adresse est automatiquement ajoutée à la commande
- Le retour à la ligne est inséré automatiquement
- OBR_ est mis au début de la ligne si besoin

Exemple : « **connEcTer_admin 2560** » Pour se connecter en administrateur sur la carte sur laquelle nous sommes connecté.

6.3. Utilisation avancée avec TeC'OBiR

Nous pouvons définir le destinataire en commençant la ligne par « p:, m:, t:, e1: ou e2: », correspondant respectivement à : PC, Maître, Télécommande, Esclave1 et Esclave2.

Exemple : « **m:connecter_admin 2560** » Pour se connecter en administrateur sur le maître.

7. Utilisation des fichiers de configuration automatiques '.map'

7.1. Utilité d'un fichier .map

7.2. Contenu du fichier

7.3. Créer un fichier de configuration pour votre robot

8. Liste des commandes de configuration pour OBR

8.1. Commandes en tant qu'utilisateur

8.2. Commandes en tant qu'administrateur

9. Liste des commandes de pilotage pour OBR

10. Liste des commandes pour OBRoC Arduino

11. Utilisation de la bibliothèque en C pour créer votre propre programme sur PC

Une bibliothèque codée en C a été développée pour faciliter la création des programmes devant utiliser le protocole OBR.

La bibliothèque est extrêmement simple à utiliser, et permet de communiquer avec un arduino sous OBR ou OBRoC sans problème.

Cependant, il faut qu'une bibliothèque d'utilisation du port série, et agréementée pour être utilisée par OBR.h soit incluse (souvent fournie avec la bibliothèque)

Une structure est définie dans le header, permettant de voir le robot comme un objet avec un nom, un port série, etc...

La liste des fonctions de la bibliothèque n'est pas très grande, mais suffisante :

- void OBR_Commande_Console(char* Buffer, Commande_OBR *Commande); /// Récupère la commande tapée en console (inutile pour un programme graphique)
- int OBR_Recup_Commande(Commande_OBR *Commande); /// Récupère la dernière commande reçue par le PC, retourne l'adresse de l'envoyeur
- void OBR_Envoie_Commande(char* Message, Commande_OBR *Commande, int NoDestinataire); /// Envoie une commande à l'arduino
- int OBR_Arduino_Pret(Commande_OBR *Commande); /// Demande à l'arduino s'il est prêt, retourne 1 si connecté en tant qu'utilisateur, 2 si on est administrateur, et 0 si déconnecté du PC
- void OBR_Maintenir_Connexion(Commande_OBR *Commande); /// Envoie un message de watchdog pour maintenir la connexion active
- int OBR_Connecter(int NoPort, int TypePeripherique); /// Demande à l'arduino une connexion, en se faisant passer pour un PC (configuration) ou une télécommande (pilotage), retourne l'adresse de la Commande_OBR si connecté
- void OBR_Deconnecter(Commande_OBR *Commande); /// Déconnecte l'arduino du PC