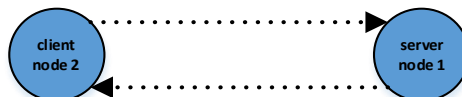


Déroulement (2h) :

- Deux programmes à développer et à déposer sur « 100-DepotTE\nom_de_famille » à la fin du test (inclus le fichier Makefile) ;
- Commenter votre code ;
- **Utiliser des variables globales uniquement si c'est absolument nécessaire.**

Énoncé :

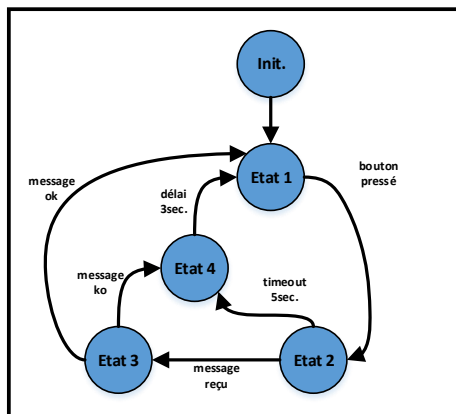
Réaliser une communication *unicast* entre deux *nodes* permettant à un *node* (*client*) de demander le temps de fonctionnement à un autre *node* (*server*).



Vous devez partir des canevas mis à disposition (ucastClient_TPL.c et ucastServer_TPL.c) et remplacer les trois dernières lettres avec vos initiales.

Le fonctionnement du « Node Client » est le suivant :

- Init. : toutes les LEDs sont allumées pendant 2 secondes et le message « Node Client : start » est envoyé sur la console ;
- Etat 1 : la LED verte clignote (fréquence de 1Hz avec l'utilisation d'un *etimer*) ; le *node* attend la pression du bouton pour envoyer, en mode *unicast*, au *node server* le message « Time? » et passe à l'état 2, sinon reste à l'état 1 ;
- Etat 2 : la LED verte clignote (fréquence de 2Hz) ; un *timer* « timeout » de 5 secondes est enclenché (utiliser un *ctimer*) ; si un *event* est reçu de la fonction *receiver*, on passe à l'état 3 ; si le *timer* « timeout » arrive à échéance, afficher le message « timeout » et passer à l'état 4 ;
- Etat 3 : la LED verte clignote (fréquence de 4Hz) ; le *node* vérifie si le message reçu est au bon format « ServerID :yy ;Time: xxxx » (tester s'il commence par ServerID) ; si oui on l'affiche sur la console et on passe à l'état 1, sinon on affiche le message « error » et on passe à l'état 4 ;
- Etat 4 : la LED rouge clignote (fréquence de 2Hz) pendant 3 secondes (utiliser un *etimer*), ensuite affichez le message « wait » et passez à l'état 1 ;



Le fonctionnement du « Node Server » est le suivant :

- Init. : toutes les LEDs sont allumées pendant 2 secondes et le message « Node Server : start » est envoyé sur la console ;
- Etat 1 : la LED bleue clignote (fréquence de 2Hz avec l'utilisation d'un *etimer*) et le *node* attend la réception d'un message (via un *event*) pour passer à l'état 2 ;
- Etat 2 : la LED bleue clignote (fréquence de 4Hz) et le *node* vérifie le message reçu ; si le message est correct, on passe à l'état 3 sinon on passe à l'état 4 ;
- Etat 3 : la LED bleue clignote (fréquence de 4Hz) ; le *node* attend 3 secondes et ensuite envoie la réponse « ServerID :yy ;Time: xxxx » à l'adresse du client (**sachant qu'il peut y avoir plusieurs clients**) ;
Les lettres 'yy' représentent le numéro de la réponse et les lettres 'xxxx' représentent le temps de fonctionnement du *node Server* ;
Après l'envoi on passe à l'état 1 ;
- Etat 4 : on affiche le message « error » et la LED rouge clignote (fréquence de 2Hz avec l'utilisation d'un *etimer*) pendant 4 secondes, ensuite passer à l'état 1.

Note 1 : quand le server traite la requête d'un client, il n'écoute plus les éventuelles requêtes des autres clients.

