TP 3 : Bits, octets, trames et code

Consignes générales

- Un compte-rendu par trinôme
- Justifiez vos réponses mais soyez concis

Ojectifs

- Mécanismes de MAC

Consignes

- Voir TP précédent

Introduction

Après quelques séances de réseau, vous avez compris que les nœuds s'échangent des trains de bits sur un médium en introduisant des schémas spéciaux indiquant le début et la fin des messages. Ces trains de bits sont découpés logiquement en champs pouvant avoir des significations diverses. Par exemple, dans une trame Ethernet, vous pourrez identifier au moins les adresses de source et de destination ainsi que la charge utile. Dans une trame Bluetooth, l'adresse _____ est absente car _____.

Vous allez mettre en place un réseau de communication avec découverte de voisin. Après les exercices préparatoires, vous mettrez en œuvre les mécanismes de base :

- implémentation d'un format de trames
- génération d'un acquittement

Activités

I. Interaction avec les nœuds

Dans la suite de vos manipulations, vous aurez besoin d'envoyer des commandes vers votre nœud communicant (la carte TeensyWino) en passant par le clavier et le moniteur série. En effet, l'interface du moniteur série contient un champ vous permettant d'entrer des messages à destination du nœud. Le tableau suivant liste les différents ordres à implémenter et leurs effets.

Code	Résultat
a	Retourne l'adresse sur 16 bits du nœud au format hexadécimal
lon	Active le clignotement de la LED (ON pendant 1s et OFF pendant 1s)
loff	Stoppe le clignotement de la LED

En utilisant les méthodes de *Serial* (par exemple, *available()* et *readString()*), implémentez les comportements correspondants dans l'émetteur et le récepteur. Une piste vous est proposée dans le listing suivant.

```
int led = 13;

void setup() {
 Serial.begin(9600);
 delay(5000);
 Serial.println("On to the main loop...");
 pinMode(led, OUTPUT);
}

void loop() {
   int j;
   if (Serial.available())
   {
     String str = Serial.readString();

     if(str.equals("marco")){
       Serial.println("polo");
     }
   }
}
```

Listing 1: Exemple de détection de message sur le port série

Pour vérifier votre implémentation, définissez une adresse sur 16 bits et récupérez-la en envoyant la commande appropriée par le lien série.

II. Attente de manière efficace

Il est probable que, durant la séance précédente, vous ayez réalisé votre émetteur en implémentant la phase d'attente avec la fonction *delay*(...). Cette commande met en œuvre une attente bloquante : votre nœud ne peut rien faire tant que la durée passée en paramètre n'est pas écoulée. Si un événement ne produit durant cet intervalle, il est ignoré. Pour éviter ce type de situation, vous allez mettre en œuvre l'attente basée sur les *timers*. Le principe est le suivant :

1. Marquer l'heure de départ (T0)
2. Comparer régulièrement la différence « delta = heure courante - T0 » et la durée d'attente souhaitée
3. Lorsque delta est supérieur ou égal au délai, la prochaine action est exécutée.

Pour mettre en œuvre cet algorithme, vous utiliserez la fonction *millis*().

III. Formatage de messages

Il est temps de mettre en forme les messages échangés. La figure 1 présente le format général des messages.

figure 1: Format général de trame

Le champ *Frame Control* (FC) de 8bits est décrit par la figure 2 : il comporte 1 bit (ACK *request*) indiquant que la trame en cours doit être acquittée et un groupe de 3 bits permettant de spécifier le type de trames (*type*). Pour le moment, seul le type de trame « Données » sera considéré. Son code sera 000.

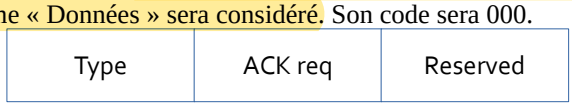


figure 2: Détail du champ Frame Control

Les bits du champs *Reserved* sont réservés à un usage futur et obligatoirement mis à 0.

Le numéro de séquence sera codé sur 8 bits et incrémenté à chaque trame envoyée et correctement reçue si un acquittement est requis. Autrement, le numéro de séquence est incrémenté systématiquement. Les adresses de destination seront codées sur 16 bits.

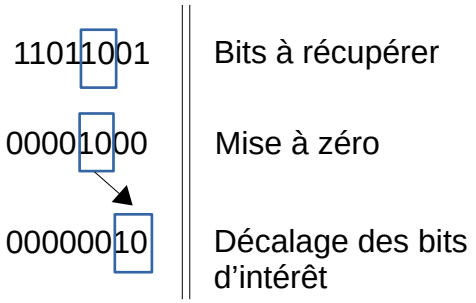
- Représentez sur votre compte-rendu la machine d'états de votre émetteur en tenant compte de l'influence du bit ACK req.

Considérerez une taille maximale de trames de 150 octets. Modifiez votre émetteur de manière à ce qu'il implémente ce format de trames : pour chaque partie de la trame, créez un en-tête de fonction permettant de modifier le champ sans affecter les champs adjacents. Créez également les en-têtes de fonction de lecture de champs.

- Où appellerez-vous ces fonctions ?

Pour déterminer le type d'un message, votre nœud devra être en mesure de ne considérer qu'un sous ensemble des bits de l'octet associé. Vous utiliserez pour isoler ces bits les opérateurs « et », « ou » et décalage binaire. Une manière commune de procéder pour la lecture est de :

i. mettre à zéro les bits non-importants,
ii. décaler à _____ le groupe de bits d'intérêt. Ce décalage fera correspondre le bit de poids faible de l'octet au bit de poids faible du groupe de bits,
iii. comparer le résultat aux différentes valeurs définies pour le groupe de bits. Ces valeurs seront souvent définies comme des constantes avec des lignes du type `#define CONSTANTE VALEUR`.



- Pour chacune des étapes, retrouver l'opérateur dans le langage de programmation.
- Proposez une méthode (étapes et opérateurs) pour modifier des bits ou groupes de bits dans un octet.

A partir de votre réponse, implémentez les prototypes spécifiés dans l'émetteur. Répercutez vos modifications sur le récepteur de manière à ce qu'il reconnaisse les différents champs de la trame : à la réception d'un message, faites analyser sa structure par le récepteur.

- Quel est l'intérêt de faire dépendre le traitement de tous les champs de la valeur du champ « adresse de destination » ?

IV. Data-Ack avec *Stop and Wait*

Puisque vos trames comportent un bit indiquant l'attente par la source d'un acquittement, il est temps pour vous de développer cet acquittement. Le format de cette trame sera simplifié pour ne contenir que le *Frame Control*, et le numéro de séquence.

Dessinez la machine d'états du récepteur.

Créez deux nouveaux sketches pour implémenter ce comportement. (A ce point du TP, vous avez 4 sketches à rendre)

Bonus : Exploitation des données

Vous disposez à présent de nœuds capables de s'échanger des messages, de les décoder suivant un format prédéterminé et de les acquitter. Vous allez exploiter ces fonctionnalités et les commandes définies à l'exercice 0 pour contrôler à distance l'activation/désactivation de la LED du récepteur.

Présentez votre démarche, implémentez les modifications appropriées et réalisez une démonstration.