

TP 2 : Descente dans les couches basses

Consignes générales

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

Ojectifs

- Transmission et réception de messages radio

Matériel et logiciel

Vous disposerez pendant ce TP de 2 nœuds TeensyWiNo (<http://snootlab.com/shields-snootlab/1137-teensywin0-kit-accessoires-fr.html>). Quelques consignes de sécurité :

- Ne tentez pas de débrancher la tresse reliant l'antenne au shield : le connecteur supporte un nombre très limité d'insertions/retraits.
- Pour éviter les court-circuits, ne posez pas les nœuds sur une surface métallique.

Vous utiliserez vos ordinateurs personnels dans le cadre de ce TP. Il est de votre responsabilité de savoir réaliser les diverses manipulations demandées. Vous aurez bien entendu installé l'IDE Arduino et le module Teensyduino permettant de flasher des Teensy.

Consignes

- Les sketches que vous produirez devront contenir une bannière (code 1) indiquant la date, les noms et prénoms des membres du binôme et un court résumé du travail réalisé.

```
/**
 * \author   {Etudiant 1, Etudiant 2}
 * \date
 * \brief    Résumé
 */
```

Code 1: Modèle de bannière

- Commentez les blocs fonctionnels de votre code.
- Produisez un compte-rendu au format pdf par binôme. Le nom du fichier suivra le modèle TP1_etudiant1_etudiant2.pdf. Vous déposerez le fichier sur Moodle dans l'espace dédié.

Activités

I. Prise en main du matériel

Chaque binôme a reçu un package contenant deux nœuds communicants. Un nœud communicant se construit avec un µC Teensy, un *shield* radio, une antenne et une batterie.

Ne branchez pas la batterie si l'antenne n'est pas reliée à la carte : dans le cas contraire, l'énergie à dissiper peut détruire le modem radio.

Quelle version du Teensy est utilisée ? Vérifiez que vous pouvez choisir la plate-forme appropriée sur l'IDE Arduino

Quelle est la bande de fréquence supportée par le module radio ? Pourriez-vous utiliser une antenne WiFi dans ce cas ? Pourquoi ? Évitez de débrancher de manière répétitive l'antenne : le connecteur utilisé est fragile et ne supporte qu'un nombre limité d'insertions/retraits.

Assemblez votre nœud communicant. Si votre kit comporte des batteries, laissez-les de côté pour l'instant.

Afin de téléverser vos sketches sur votre Teensy, reliez-le par le câble USB fourni à votre machine. Sur l'IDE, retrouvez, parmi les *exemples/Basics*, le sketch *Blink*. Ce sketch basique a pour objectif de faire clignoter une LED sur la carte. Au moment de téléverser votre code pour la première fois, vous aurez à presser un bouton sur le Teensy pour lui permettre de basculer en mode programmation. Un orifice a été aménagé sur le *shield* pour vous donner accès à ce bouton. Utilisez un cure-dent épointé (**pas de tige métallique!**) pour réaliser l'opération.

II. Rx ON

```
#include <SPI.h>
#include <RH_RF22.h>

#define SET_RX 1
#define RECEPTION 2
#define PRINT_FRAME 3
#define REPLY 4
#define canal 0

RH_RF22 rf22(SS,9);

uint8_t state;
uint8_t rxbuf[RH_RF22_MAX_MESSAGE_LEN];
uint8_t rxbufLen = RH_RF22_MAX_MESSAGE_LEN;
uint8_t rxlen = RH_RF22_MAX_MESSAGE_LEN;
int rxf = 0;

// the setup routine runs once when you press reset:
void setup() {
  Serial.begin(11200);
  delay(5000);
  SPI.setSCK(14);

  if(!rf22.init()){
    Serial.println("init failed");
    while(1);
  }else
    Serial.println("init OK");
  rf22.setTxPower(RH_RF22_TXPOW_8DBM);
  rf22.setModemConfig(RH_RF22::GFSK_Rb125Fd125);
  rf22.setFrequency(433.1+canal*0.1, 0.05);

  state = SET_RX;
  delay(3000);
  Serial.println("On to the main loop...");
}

// the loop routine runs over and over again forever:
void loop() {
  int j;
  rxlen = sizeof(rxbuf);

  switch(state){
    case SET_RX:
      Serial.println("Activating RX mode...");
      rf22.setModeRx();
      state = RECEPTION;
      break;
    case RECEPTION:
      if(rf22.available()){
        state = PRINT_FRAME;
      }
      break;
    case PRINT_FRAME:
      if(rf22.recv(rxbuf, &rxlen)){
        rxf++;
        Serial.printf("[%d]", rxf);
        for(j=0; j<rxlen; j++){
          Serial.printf("%02x|", rxbuf[j]);
          Serial.println();
        }
        state = SET_RX;
        break;
      }
      default:
        break;
  }
}
```

Code 2: Sketch récepteur

L'encadré Code 2 présente le sketch d'un récepteur. Il s'agit de l'implémentation sous forme de switch/case de la machine d'état du nœud. Ce fonctionnement peut être représenté sous la forme d'un graphe des états où les états sont représentés par des cercles et les transitions par des flèches. Une transition peut être conditionnée, auquel cas le critère de test sera spécifié. Les états sont nommés (par convention, en majuscule) et numérotés.

- Dessinez sur votre CR l'automate correspondant au sketch proposé.
- Sur quel canal radio écoute votre nœud ?
- Téléchargez ce code sur l'un de vos nœuds : au besoin, utilisez à nouveau un cure-dent époiné pour faire basculer votre Teensy en mode programmation. Pour visualiser les chaînes de caractères produites, lancez le moniteur série en veillant à choisir le port série apparaissant après la première pression du bouton de programmation sur votre Teensy.

III. Génération de trafic

Vous allez à présent implémenter un émetteur radio. Cette source, une fois démarrée, va transmettre le même message toutes les 500ms. Le contenu de ce message sera 0x123456789A.

Dessinez le graphe d'état de votre émetteur. Donnez des noms explicites aux états que vous utiliserez dans votre code.

Implémentez votre émetteur : vous aurez à utiliser les deux méthodes suivantes :

- `rf22.send(données, taille de la donnée)`
- `rf22.waitPacketSent()`

A partir de la documentation de la classe, précisez le type des variables passées en paramètre et la signification des valeurs de retour si elles existent.

Testez le couple émetteur/récepteur : dans votre compte-rendu, présentez le code de l'émetteur, la trace produite par le récepteur et commentez-les.

Si vous avez rencontré un problème, décrivez-le et présentez dans le CR votre démarche de résolution.

IV. Génération de trafic II

Modifiez votre émetteur pour que le premier octet soit l'identifiant de groupe de TP, le second un identifiant unique de binôme (mettez-vous d'accord avec vos voisins !) et que le troisième soit le numéro de séquence de la trame. Interprétez la trace du récepteur.

Comment pouvez-vous filtrer physiquement les messages afin de ne recevoir que ceux de votre collègue ? Comment le faire logiciellement ?

Implémentez et vérifiez la stratégie physique.