

JANKOWIAK Claire  
MOUNEU Olivier

### TP3 : Bits Octets Trames et Code

#### Partie 1 :

Les adresses suivantes ont été définies :

- Claire : AA11
- Olivier : AA00

En utilisant l'exemple (Listing 1), on a implémenté les comportements liés aux commandes *a*, *lon* et *loff*.

```
int led=13;
void setup() {
  Serial.begin(9600);
  Serial.println("On to the main loop...");
  pinMode(led, OUTPUT);
}
void loop() {
  if (Serial.available())
  {
    String ADDRESS = "AA11";

    String str = Serial.readString();

    if(str.equals("marco\n")){
      Serial.println("polo");
    }
    if(str.equals("a\n")){
      Serial.println(ADDRESS);
    }
    if(str.equals("lon\n")){
      digitalWrite(led, HIGH);
    }
    if(str.equals("loff\n")){
      digitalWrite(led, LOW);
    }
  }
}
```

## Partie 2 :

On met en place une attente non bloquante en utilisant la fonction *millis()*. Le code implémentant cette fonction est copié ci-dessous.

La variable *début* correspond à l'heure de départ T0. A l'exécution du logiciel, elle est définie sur 0. Dans la fonction *loop()* on vérifie si le temps utilisé pour réaliser une tâche a dépassé le délai défini (variable *délai* défini à 2000ms) grâce au calcul :

$$\text{delta} = \text{heure courante} - T0$$

Si le délai est dépassé on change d'état (de tâche) et on change T0 par l'heure actuelle.

```
#include <SPI.h>
#include <RH_RF22.h>
#define SET_TX 1
#define SEND 2
#define WAIT_SEND 3
#define REPLY 4
#define canal 4

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;

int led = 13;
String ADDRESS = "AA00";
int debut=0;
int delai=2000;

// the setup routine runs once when you press reset:
void setup() {

  pinMode(led, OUTPUT);
  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 = SEND;
  delay(3000);
```

```

        Serial.println("On to the main loop...");
    }
    // the loop routine runs over and over again forever:
    void loop() {

        switch(state){
            case SEND:
                send();
                break;
            default:
                break;
        }
        int delta = millis()-debut;

        if( delta > delai ) {
            Serial.println("Delai dépassé");
            debut = millis();
        }
    }

    void send() {
        Serial.println("Activating TX mode...");
        rf22.setModeTx();

        uint8_t data[5];
        data[0] = 0x99;
        data[1] = 0x34;
        data[2] = 0x54;
        data[3] = 0x76;
        data[4] = 0x94;
        rf22.send(data, sizeof(data));

        Serial.println("Sending...");
        digitalWrite(led, HIGH);

        rf22.waitPacketSent();
    }
}

```

### Partie 3 :

#### Machine à états de l'émetteur (utilisant les ACK)

