

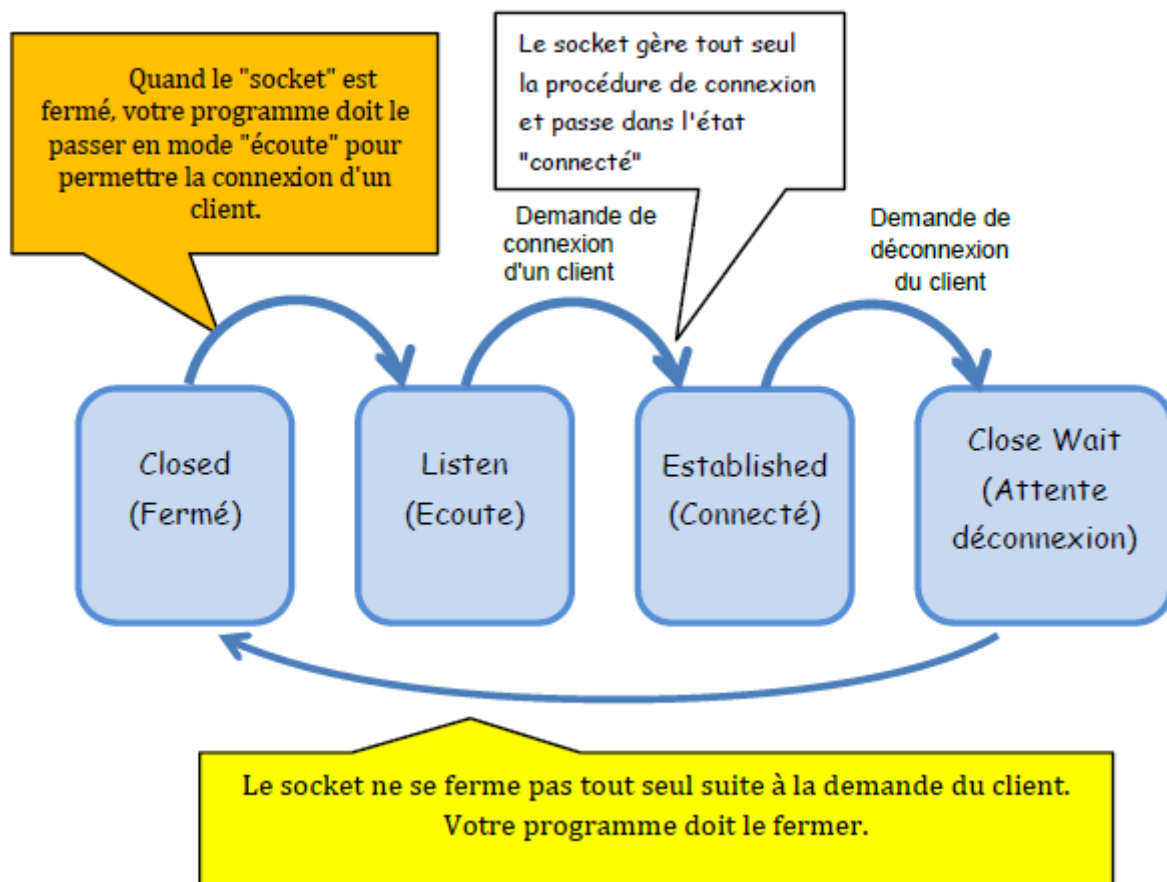
## SEQ : TCP & Arduino

Le protocole TCP est un protocole "orienté connexion" avec la notion de Client/Serveur. Dans cette étude l'arduino sera utilisé en Serveur.

Cela implique quelques différences d'utilisation du "socket" TCP par rapport au "socket" UDP.

En mode "TCP serveur" le résumé du fonctionnement est le suivant :

- Il faut bien sûr commencer par créer un "socket" TCP (Par défaut le socket est "fermé" : Aucun client ne peut se "connecter")
- Ensuite il faut placer le "socket" en mode "écoute". (A partir de cet instant, un client peut se connecter)
- Quand un client est connecté, on peut échanger des données avec lui.
- Quand le client ne désire plus échanger de données, il fait une demande de déconnexion, à ce moment-là le serveur doit "fermer" le "socket" (pour terminer le processus de déconnexion) et ensuite replacer le socket en mode "écoute" pour permettre à nouveau la connexion d'un client...



1. Le programme Arduino complet est dans le dossier Ressources – Arduino – TCPserveur.

```
void setup()
{
  Ethernet.begin(mac, ip_shield);
  server.begin();
  lcd.init();
  lcd.backlight();
  lcd.print("Server started");
  delay(3000);
}

void loop()
{
  lcd.home();
  lcd.print("Attente client");

  EthernetClient client = server.available();

  if (client)
  {
    while (client.connected()==true)
    {
      client.flush();
      message="";
      change=false;
      while (client.available() > 0)
      {
        char c = client.read();
        message+=c;
        change=true;
      }

      if (change == true) {
        lcd.clear();
        lcd.print("Client connecte");
        lcd.setCursor(0,1);
        lcd.print(message);
        client.print("Message recu :"+message);
      }
    }
    lcd.clear();
  }
}
```

le socket est créé, le serveur est démarré ;

Le serveur est en mode « écoute » ;

Le client est connecté suite à l'envoi de données, on les affiche sur l'écran LCD;

### 2. Programmez l'Arduino.

- Ouvrez sur votre ordinateur le logiciel Hercule onglet TCP Client et configurez-le.
- Dans la zone *Send* saisissez un message et cliquez sur le bouton *Send* ; le message doit s'afficher sur l'écran LCD.
- Fermez puis ré-ouvrez la connexion dans Hercule et constatez que l'émission reste possible.

### 3. Côté B4A vous pouvez reprendre le TCP avec les librairies.

```

36
37 Sub Activity_Create(FirstTime As Boolean)
38     Activity.LoadLayout("layout1")
39     Label1.text="Non connecté"
40     Connect.Visible=True
41     deconnect.Visible=False
42     Send.Visible=False
43     EditText1.Visible=False
44 End Sub
45 Sub TCP_client_Connected(Succes As Boolean)
46     Label1.text="Connecté"
47     Connect.Visible=False
48     deconnect.Visible=True
49     Send.Visible=True
50     EditText1.Visible=True
51 End Sub
52
53 Sub Send_Click
54     Dim emission() As Byte
55     Dim bc As ByteConverter
56     Dim message As String
57     message=EditText1.Text
58     emission=bc.StringToBytes(message,"ASCII")
59     TCP_client.emission(emission)
60 End Sub
61
62 Sub Connect_Click
63     TCP_client.connexion("192.168.1.205",5500)
64 End Sub
65
66 Sub deconnect_Click
67     TCP_client.fin_connexion
68     Label1.text="Non connecté"
69     Connect.Visible=True
70     deconnect.Visible=False
71     Send.Visible=False
72     EditText1.Visible=False
73 End Sub
74
75 Sub TCP_client_newdata(buffer() As Byte)
76     Dim msg As String
77     msg=BytesToString(buffer,0,buffer.length,"ascii")
78     Label2.Text=msg
79 End Sub
80
81

```