



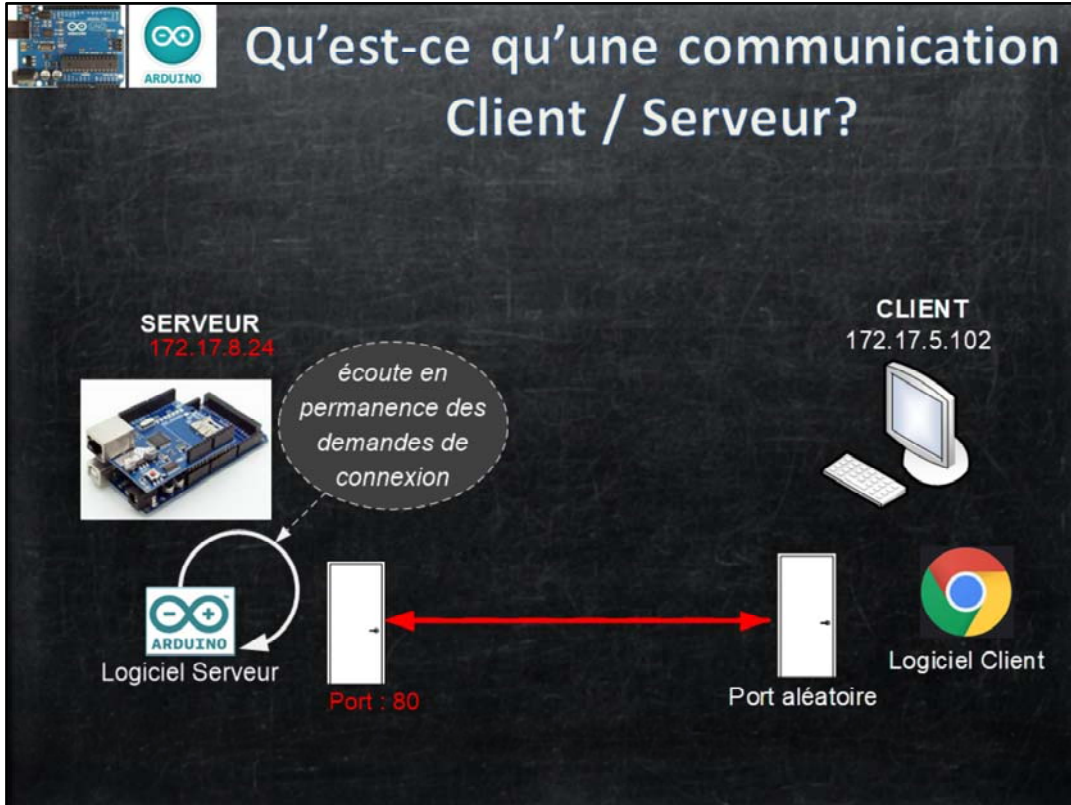
SÉANCE 3 – Activité 3



Communication entre systèmes Par TCP/IP



Dans cette nouvelle capsule, nous allons voir comment mettre en place une communication Client/Serveur par TCP entre une carte Arduino et une interface Web

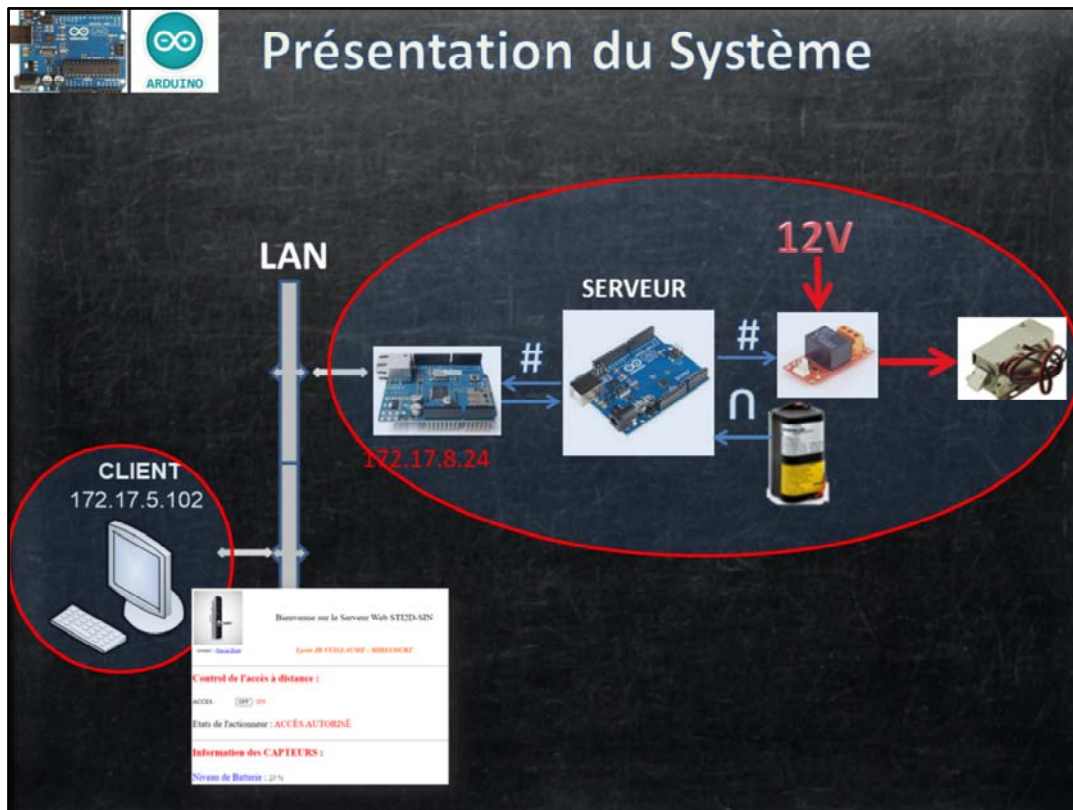


Petit rappel concernant la communication Client/Serveur (**CLICK**)

LA communication Client/Serveur nécessite la création d'un Socket sur le Serveur. Le serveur se met alors en écoute permanente de connexion sur ce socket (**CLICK**).

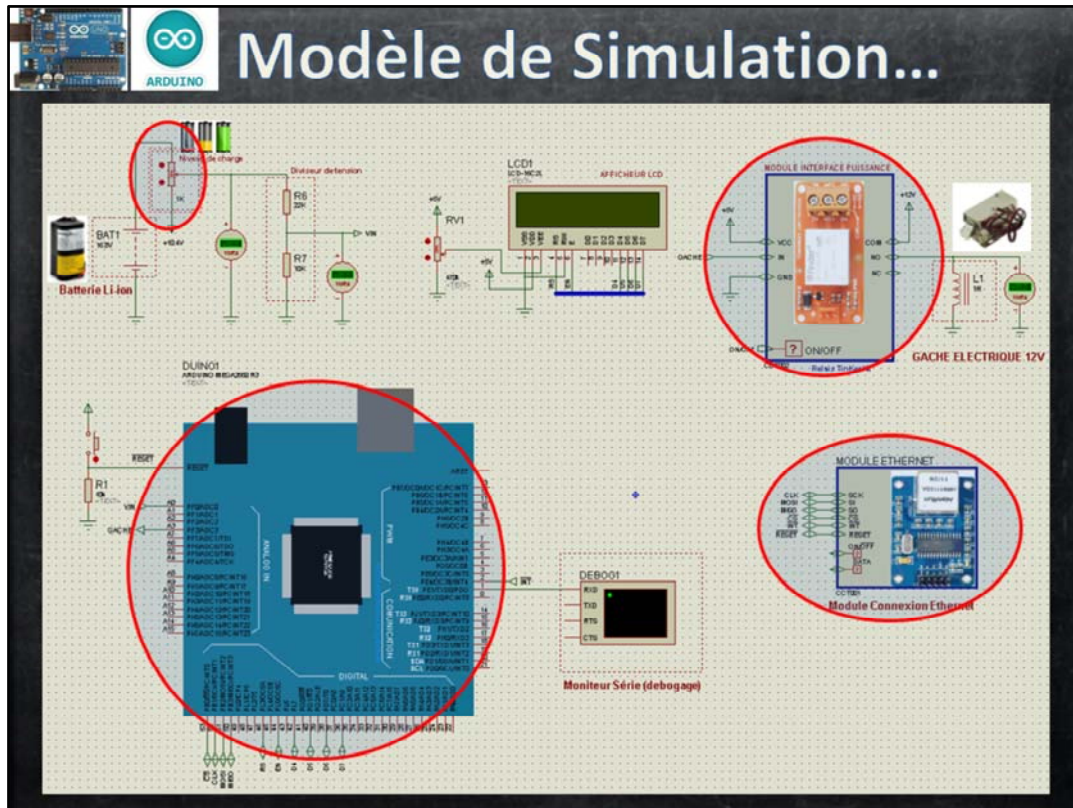
Lors qu'un Client veut se connecter au serveur (**CLICK**), il lui faudra donc se connecter à cet socket en indiquant l'adresse IP et le port qui est ouvert (**CLICK**)

Une fois la REQUETE de connexion acceptée par le serveur, la communication entre le Client et le serveur est alors active et bidirectionnelle (**CLICK**)

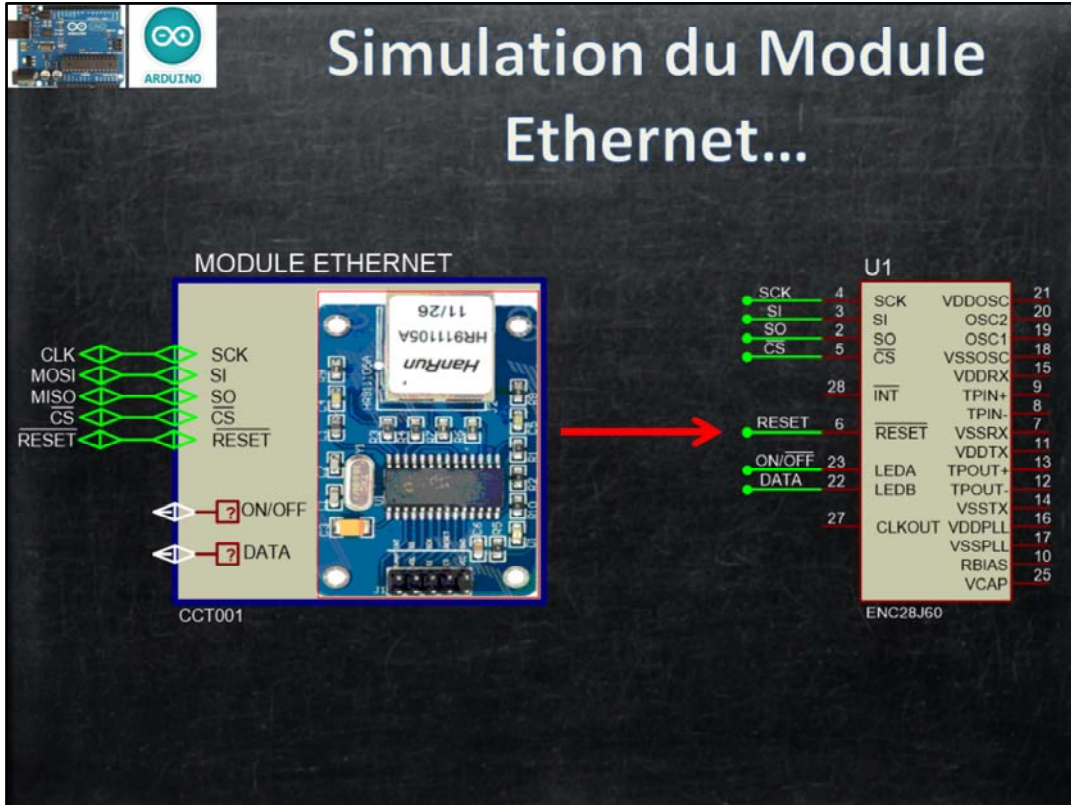


Dans cette activité (**CLICK**), la partie Serveur sur Arduino permettra la commande de la gâche électrique via une interface de puissance (RELAIS), ainsi que la lecture du niveau de batterie.

Le Client (**CLICK**) doit pouvoir commander à distance l'accès à l'habitation, et visualiser le niveau de batterie restante en % à partir d'une interface de communication Web sur un navigateur (**CLICK**)

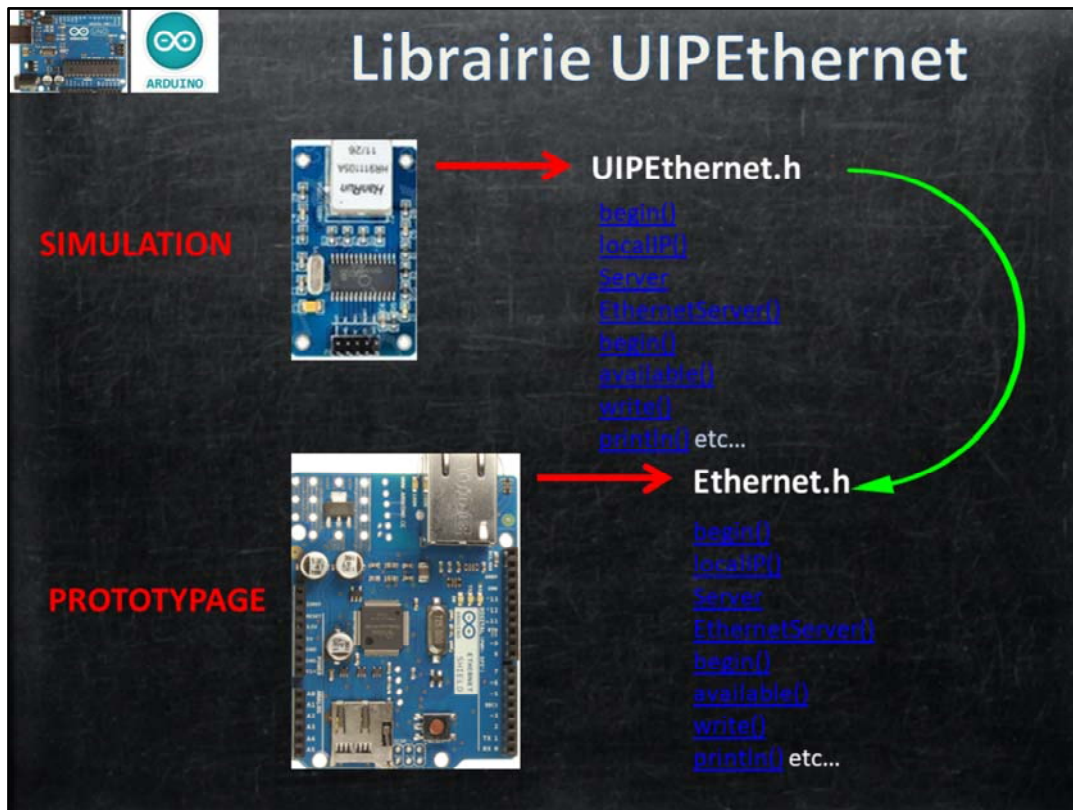


Voici le modèle qui sera utilisé pour la simulation : **(CLICK)** une carte Arduino Mega pour la programmation du Serveur, **(CLICK)** un module Relais pour la commande de la GACHE, **(CLICK)** un POTENTIOMETRE pour la modification de la tension batterie et enfin **(CLICK)** le module Ethernet pour la communication TCP/IP via la carte réseau de votre PC grâce au module ENC28J60



En fait derrière le Module Ethernet (**CLICK**) se cache un circuit électronique : l'ENC28J60 (**CLICK**) . Celui-ci permet d'utiliser la carte réseau de la machine comme interface Réseau pour le simulateur ISIS

Ce circuit utilise tout comme le module Ethernet le bus SPI pour communiquer avec Arduino.



La librairie UIPEthernet permet de commander le module Ethernet à base de l'ENC28J60 (**CLICK**). L'intérêt majeur étant que les fonctions utilisées (**CLICK**) sont exactement les mêmes que celles utilisées par la librairie Ethernet Arduino (**CLICK**) des modules Shield Ethernet qui utilisent le contrôleur W5100.

Ce qui permet de simuler puis de prototyper (**CLICK**) directement la solution simplement en changeant la librairie du programme



Connexions utilisées sur Arduino...

Broche	SIGNAL	RÔLE
0	RXD0	Réception données en série sur Port USB
1	TXD0	Transmission des données en série sur port USB (débugage)
A3	GACHE	Sortie numérique vers le Module Relais (commande de l'ACCES)
53	CS	Module Ethernet (Sélection du module)
52	CLK	Module Ethernet (Signal d'horloge)
51	MOSI	Module Ethernet (Données Maître -> Esclave)
50	MISO	Module Ethernet (Données Esclave -> Maître)
A0	VIN	Entrée analogique : mesure de la tension de la BATTERIE

Voici les différentes broches utilisées pour le module Arduino :

(CLICK) La commande de la GACHE se fera par la broche A3.

(CLICK) Les broches 2 et 50 à 53 seront utilisées par le module Ethernet (Bus SPI)

(CLICK) Et enfin A0 sera l'entrée Analogique permettant la mesure de la tension BATTERIE



Les fonctions Arduino...

Fonction	RÔLE
<code>EthernetServer server(PORT)</code>	Instancie l'objet serveur (objet qui écoute le PORT)
<code>Ethernet.begin(mac, ip, DNS, gateway, subnet)</code>	Initialise les paramètres de la carte réseau
<code>server.begin();</code>	Démarre le serveur
<code>EthernetClient client = server.available();</code>	Instancie l'objet client à partir du serveur (true ou false)
<code>if (client)</code>	Permet de tester si un client est présent (true)
<code>while (client.connected())</code>	Tant que le client est connecté
<code>char c = client.read();</code>	Lecture du caractère transmis par le client
<code>readString = readString + c;</code>	Concaténation du caractère reçu dans la chaîne readString
<code>client.println("<head>");</code>	Transmission du code HTML au client
<code>client.stop();</code>	Arrêt de la communication avec le client (libère l'objet client)

Voici différentes commandes utilisées pour la gestion d'un client/serveur sur Arduino