

# Atelier IoT-Lab

## Programmer son premier objet connecté

### Etape 1: Installation

- Installer Arduino
- Installer ESP8266 core

Aller dans fichier, préférences, URL de gestionnaire de cartes supplémentaires et copier/coller ceci :

[https://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](https://arduino.esp8266.com/stable/package_esp8266com_index.json)

- Aller dans outils, Type de carte, gestionnaire de carte et sélectionner *esp8266 Community* afin de l'installer
- Configurer NodeMCU comme plateforme par défaut

### Etape 2: Prise en main de l'interface Arduino

- Création d'un nouveau sketch
- Ecriture d'un programme « HelloWorld + LED Blink » écrivant sur le port série
- Vérification à l'aide d'un programme lisant le port série

### Etape 3 : Contrôle du Servo

- Ecriture d'un programme utilisant la librairie « servo » ; la commande est envoyée via le port série (brun – GND, Rouge – VCC, Orange – S)
- Utilisation d'une interface vers le port série pour envoyer la commande

### Etape 4 : Connexion au réseau WiFi – version DHCP

- Ecriture d'un programme permettant de se connecter à un réseau WiFi à l'aide de la librairie ESP8266WIFI :
  - o Configurer le microcontrôleur en mode station
  - o SSID : \*\*\*\*\*
  - o Password : \*\*\*\*\*
  - o Mode : DHCP (=mode par défaut)
- Récupérer, via le port série, la valeur
  - o De « Hostname »
  - o De l'adresse MAC
  - o Des paramètres IP (votre IP, IP Gateway, netmask)

### Etape 5 : Connexion au réseau WiFi – version IP statique

- Ecriture d'un programme permettant de se connecter à un réseau WiFi à l'aide de la librairie ESP8266WIFI :
  - o Configurer le microcontrôleur en mode station
  - o SSID : \*\*\*\*\*
  - o Password : \*\*\*\*\*
  - o Mode : statique
    - Pour l'IP, choisir une IP entre 192.168.2.201 à 192.168.2.230
- NB : pour cela, imposer les paramètres « votre IP », « IP Gateway », « netmask »
- Pour s'assurer que le device est bien connecté, un paquet est envoyé automatiquement après connexion par le microcontrôleur vers un serveur de gestion

### Etape 6 : Contrôle du Servo via WiFi via un socket UDP

- Utilisation d'un programme, sur votre PC, permettant de contrôler le servo via un socket UDP
  - o Soit via l'utilisation du software « PacketSender » pour envoyer la commande
  - o Soit en écrivant un programme (ex : en python) de type « programmation socket »
- Ecriture d'un programme, sur l'ESP, permettant de contrôler le servo en utilisant le wifi et en utilisant la librairie WIFIUDP

# Atelier IoT-Lab

## Ressources:

- Arduino software: <https://www.arduino.cc/en/Main/Software>
- ESP8266 core for Arduino: <https://github.com/esp8266/Arduino>
- Programme de gestion du port série (Windows):  
<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>
- Servo SG90 datasheet: <http://www.micropik.com/PDF/SG90Servo.pdf>
- Arduino Servo Library: <https://www.arduino.cc/en/reference/servo>
- Arduino Serial Library: <https://www.arduino.cc/en/Reference/Serial>
- ESP8266 WiFi Library: <https://github.com/esp8266/Arduino/tree/master/doc/esp8266wifi>
- Programme d'envoi de paquets: <https://packetsender.com/download>

## Pour aller plus loin...

- FUN-MOOC : Objets connectés : des radiofréquences aux réseaux  
<https://www.fun-mooc.fr/courses/course-v1:univ-toulouse+101003+session02/info>
- FUN-MOOC : Fabriquer un objet connecté  
<https://www.fun-mooc.fr/courses/MinesTelecom/04018S02/session02/info>
- Youtube Channel : Andreas Spiess - My ESP32 videos  
[https://www.youtube.com/channel/UCu7\\_D0o48KbfhpEohoP7YSQ](https://www.youtube.com/channel/UCu7_D0o48KbfhpEohoP7YSQ)