

## Objectifs :

- comprendre l'architecture des cartes Arduino
- comprendre l'architecture des microcontrôleurs de la famille ATmega 328
- être capable de mettre en place un environnement de développement pour les kits Arduino
- être capable de développer des applications en langage C pour les kits Arduino

Durée : 4 jours

Public : développeurs

## Prérequis :

- connaissance du langage C

## Démarche pédagogique :

- présentation des concepts, suivi d'exercices
- manipulation du kit Arduino avec platine d'essais
- 60 % d'exercices pratiques et manipulations

## Programme détaillé :

- Présentation du projet
  - l'open source dans l'électronique
  - le langage Arduino et le projet Wiring
- Kits Arduino
  - les différents kits Arduino
    - nombre d' E/S, mémoire, compatibilité
  - les shields : les cartes d'extensions et expérimentation
    - cartes afficheurs, LED, communication, support carte SIM, ...
  -
- Rappels d'électronique logique
  - les opérateurs logiques : AND, OR, NOT, XOR
  - représentation des circuits
  - niveaux logiques, courant de sortie
  - résistances et capacités
  - polarisation des transistors à l'état logique
  - utilisation des diodes et LED
- ATmega 328
  - architecture du microcontrôleur
  - le modèle mémoire
  - les entrées-sorties
    - choix des modes de fonctionnement
  - les registres spécifiques
  - les interruptions

- gestion de la consommation électrique
- Les outils de développement
  - l'EDI Arduino
  - l'EDI Atmel Studio 6
  - la chaîne de compilation
  - les bibliothèques Arduino
- Le "Hello, world" Arduino
  - coder le clignotement d'une LED
    - analyse du matériel
      - choix de la sortie
      - caractéristiques courant / tension
      - câblage
    - écriture du code
      - création d'un projet
      - analyse du code
      - chargement sur le kit
      - exécution
      - mode debug
- Quelques bibliothèques pour l'Arduino
  - langage Arduino
    - core functions
  - bibliothèques standards Arduino
    - EEPROM, Ethernet, LiquidCrystal, SD cards reading, ...
  - bibliothèques tiers
  - utiliser les bibliothèques
    - les adapter... pour tenir en mémoire
- Les entrées sorties numériques
  - les niveaux logiques pour le TOR (Tout Ou Rien)
  - les entrées TOR
  - les sorties TOR
  - les canaux PWM
- les entrées sorties analogiques
  - principes de l'échantillonnage
  - la conversion analogique numérique
  - utilisation d'un canal PWM pour créer un signal analogique
- Les interruptions
  - concepts d'interruption matérielle et logicielle
  - vecteurs d'interruption
  - interruptions externes
  - interruptions internes
- Les timers
  - horloge de référence
  - compteurs
  - timer et PWM
- Les interfaces de communication

- les interfaces séries synchrones et asynchrones
  - bus I2C, SPI
- Un peu d'assembleur
  - faut-il optimiser ?
  - ajouter de l'assembleur au code C
- Le bootloader du kit Arduino
  - principes du bootloader
  - enlever le bootloader
  - charger un bootloader
- Créer ses propres cartes
  - créer des shields Arduino
  - partir d'un projet Arduino pour créer une implémentation matérielle spécifique