

# Utiliser 2 Arduinos en série

Allumer une LED connectée à un Arduino depuis un autre Arduino.

 Difficulté Très facile

 Durée 1 heure(s)

 Catégories Électronique

 Coût 20 EUR (€)

## Sommaire

Introduction

Étape 1 - Schéma

Étape 2 - Principe du montage

Étape 3 - Réalisation du montage

Étape 4 - Programmation du montage

Étape 5 - Programme du Arduino Esclave

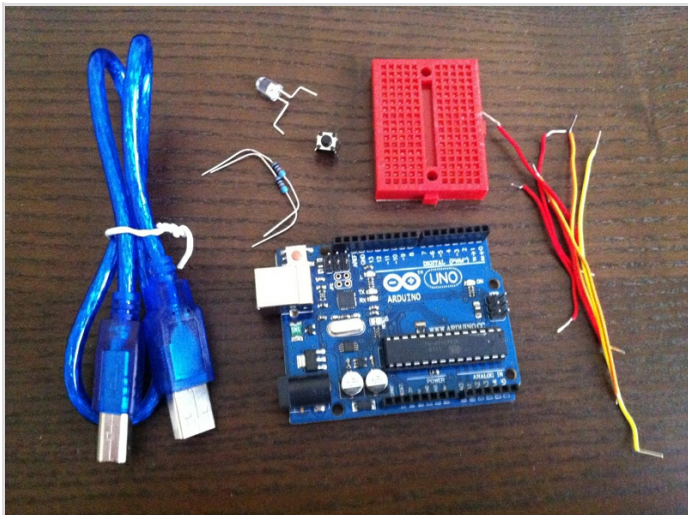
Étape 6 - Programme du Arduino Maître

Étape 7 - Fini !

Commentaires

## Introduction

Ce montage va utiliser le réseau I2C du Arduino.



## Matériaux

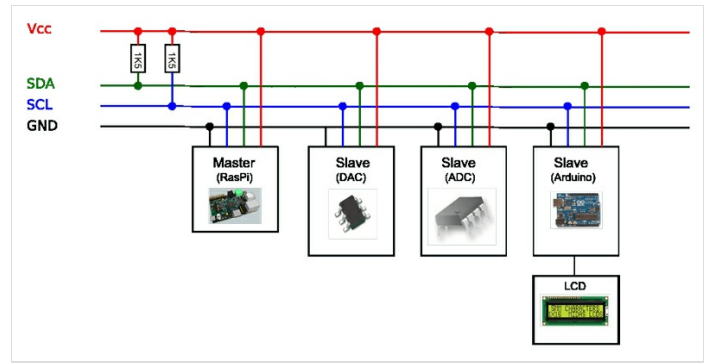
- 2 Arduino
- 1 LED
- 1 résistance (Optionnel)
- Une Breadboard
- Des fils de connexion

## Outils

# Étape 1 - Schéma

Pour mettre en réseau les 2 Arduinos, nous allons utiliser l'interface I2C du Arduino. Le réseau I2C fonctionne sur le principe Maître / Esclave. Il peut y avoir un seul maître par réseau et plusieurs esclaves (chaque esclave a un identifiant). Cette interface nécessite 2 fils le SDA (Broche Analog A4) et SCL (Broche Analog A5). Grâce au réseau I2C, il est possible de connecter différentes cartes électroniques ou simple circuit intégré (« puce ») facilement comme le montre le schéma ci-contre.

**i** On voit sur ce schéma que pour fonctionner les éléments du réseau doit être alimenté par la même alimentation. Pour simplifier ceci, nous allons alimenter le Arduino donnant la commande (qui sera le maître) avec le Arduino allumant la LED (qui sera l'esclave).



# Étape 2 - Principe du montage

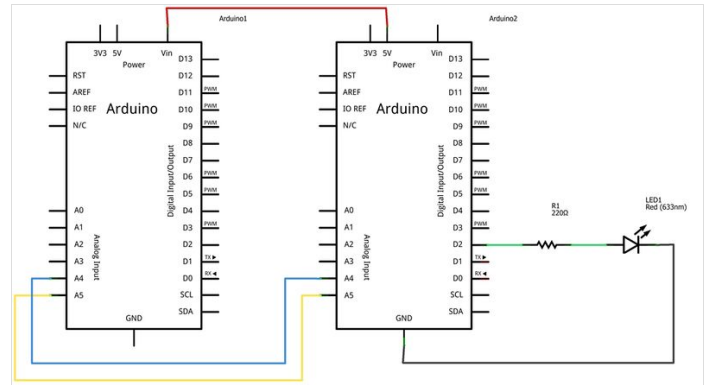
Le montage va cette fois-ci se décomposer en 2 parties :

- Le Arduino esclave: Le Arduino esclave aura une LED et sa résistance câblée entre sa sortie digital 2 et le GND. Il utilisera sa sortie 5V pour alimenter le Arduino maître.
- Le Arduino maître : Le Arduino maître reçoit son alimentation du Arduino esclave sur son entrée Vin.

Pour terminer la bonne connexion de l'alimentation, il reste à connecter entre eux les GND des 2 Arduinos.

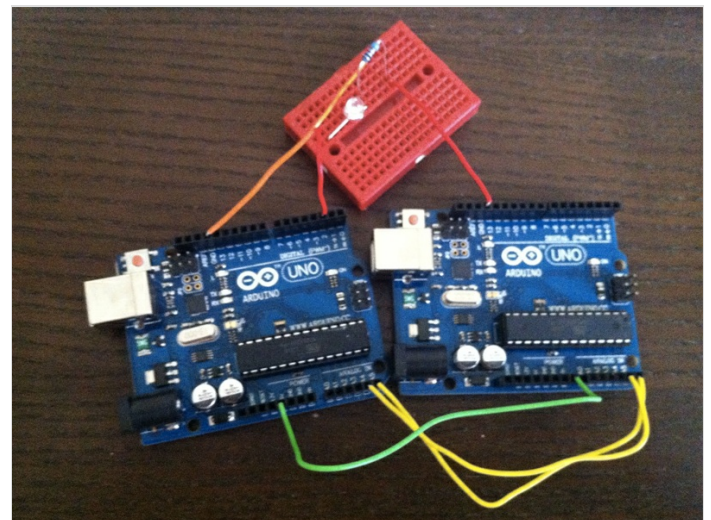
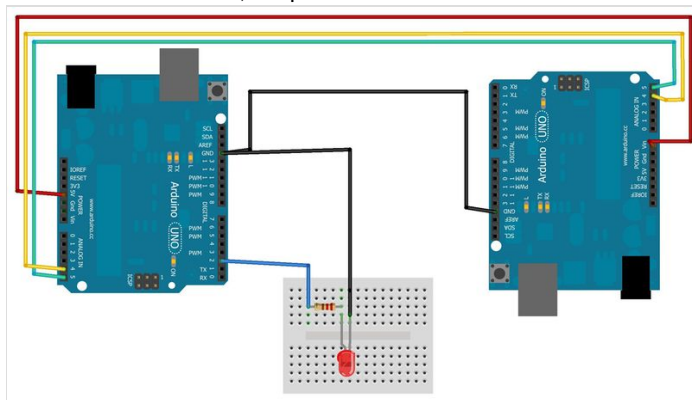
Pour établir la connexion entre ces 2 Arduinos, il va être nécessaire de raccorder les signaux SDA/SCL:

- Entre la borne Analog 4 (SDA) du Arduino maître et la borne Analog 4 du Arduino esclave
- Entre la borne Analog 5 (SCL) du Arduino maître et la borne Analog 5 du Arduino esclave



# Étape 3 - Réalisation du montage

Pour le Arduino esclave, ce montage est le même que le montage « Allumer une LED »  
Pour le Arduino maître, uniquement des connexions avec le Arduino esclave.



## Étape 4 - Programmation du montage

La programmation de ce montage s'appuie sur la librairie Wire. Comme c'est une librairie, pour utiliser ces fonctions, il faudra écrire «Wire.nomdelafonction». Elle contient les fonctions suivantes :

- **begin(X)** : Initialise la liaison I2C. Se place dans la section « setup ». Le paramètre X est l'adresse de la carte esclave. Pour le maître le paramètre reste vide.
- **RequestFrom(X, Y)** : Commande utilisée par le maître pour demander des informations aux esclaves (requête). Le paramètre X est l'adresse de l'esclave. Le paramètre Y est le nombre d'octets demandé.
- **beginTransmission(X)** : Commande utilisée par le maître pour démarrer une transmission avec l'esclave de l'adresse X.
- **endTransmission()** : Commande utilisée par le maître pour clore une transmission.
- **Write(X)** : Commande utilisée par l'esclave pour répondre le message X à une demande du maître.
- **Available()** : Renvoie le nombre d'octet disponible pour la lecture. C'est une fonction très intéressante pour savoir s'il y a des messages en attente de lecture.
- **Read()** : Lit les message en attente sur le bus. Attention : La fonction read lit caractère par caractère.
- **OnReceive(X)** : C'est une commande destinée uniquement à l'esclave qui va être appelée lorsqu'il y a des messages à lire la fonction X.
- **onRequest()** : C'est une commande destinée uniquement à l'esclave qui va être appelée lorsqu'il y a une requête du maître.

Le programme va lui aussi être coupé en 2, un pour chaque Arduino.



ATTENTION : Il faut déconnecter les Arduinos durant la programmation !

## Étape 5 - Programme du Arduino Esclave

Le Arduino esclave définit qu'il a une sortie sur digital 2. Il dit que son adresse est la numéro 4. Il va attendre de recevoir le chiffre 1 sur l'interface I2C pour allumer la LED. Quand il reçoit 0 il éteint la LED.

```
#include <Wire.h> // Librairie pour la communication I2C

const int L1 = 2; // broche 2 du micro-contrôleur se nomme maintenant : L1

void setup()
{
  Wire.begin(4); // Rejoindre le bus à l'adresse #4
  Wire.onReceive(receiveEvent); // Preparer une fonction spécifique a la reception de donnee
  Serial.begin(9600); // Demarrer la liaison serie avec le PC
  pinMode(L1, OUTPUT); // L1 est une broche de sortie
}

void loop()
{
  delay(100);
}

// Fonction qui s execute si quelque chose est present sur l interface
void receiveEvent(int howMany)
{
  int x = Wire.read(); // recevoir un chiffre
  Serial.println(x); // afficher ce chiffre sur l'interface serie
  if(x == 1)
  {
    digitalWrite(L1, HIGH); // allumer L1
  }
  if(x == 0)
  {
    digitalWrite(L1, LOW); // eteindre L1
  }
}
```

## Étape 6 - Programme du Arduino Maître

Le Arduino maître envoie un 1 sur l'interface I2C, attend 1 seconde, envoi un 0 sur l'interface I2C, attend 2 secondes et recommence à l'infini.

```
#include <Wire.h>

void setup()
{
  Wire.begin(); // Rejoindre le bus I2C (Pas besoin d adresse pour le maitre)
}

void loop()
{
  //contenu du programme
  Wire.beginTransmission(4); // Envoyer vers device #4
  Wire.write(1); // Envoi un 1
  Wire.endTransmission(); // Arreter la transmission
  delay(1000); // Attendre 1s
  Wire.beginTransmission(4); // Envoyer vers device #4
  Wire.write(0); // Envoi un 0
  Wire.endTransmission(); // Arreter la transmission
  delay(2000); // Attendre 2s
}
```

---



# Étape 7 - Fini !

Il reste à brancher le Arduino pour compiler le programme et le téléverser.

Une fois terminé cela nous donne :

- Photo 1 : LED éteinte
- Photo 2 : LED allumée

**Pour aller plus loin :** Il est possible de rendre autonome le montage en lui ajoutant un coupleur de pile. Il est aussi possible de modifier les temps allumés et éteints. On peut remplacer les temps par un bouton poussoir ce qui ferait un système de bouton déporté.

