

TP N°2 COMMUNICATION ENTRE DEUX CARTES ARDUINO

I. Schéma

Pour mettre en réseau les 2 Arduino, nous allons utiliser l'interface I2C de l'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.

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 l'Arduino donnant la commande (qui sera le maître) avec l'Arduino allumant la LED (qui sera l'esclave).

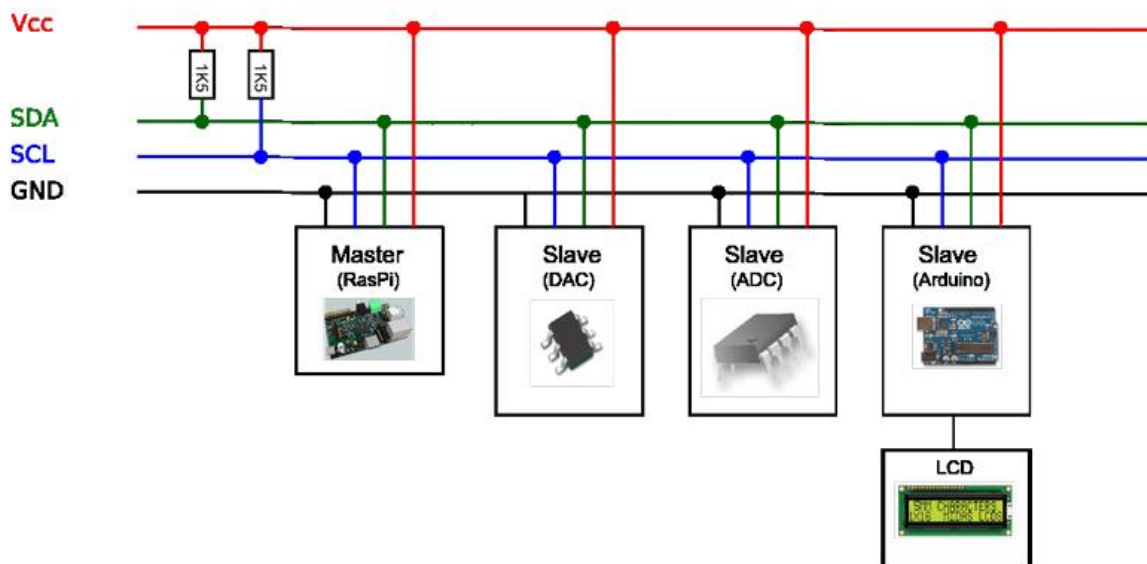


Figure 1

II. Principe du montage

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

- L'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 l'Arduino maître.
- L'Arduino maître : Le Arduino maître reçoit son alimentation de l'Arduino esclave sur son entrée Vin.

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

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

- Entre la borne Analog 4 (SDA) de l'Arduino maître et la borne Analog 4 de l'Arduino esclave
- Entre la borne Analog 5 (SCL) de l'Arduino maître et la borne Analog 5 de l'Arduino esclave

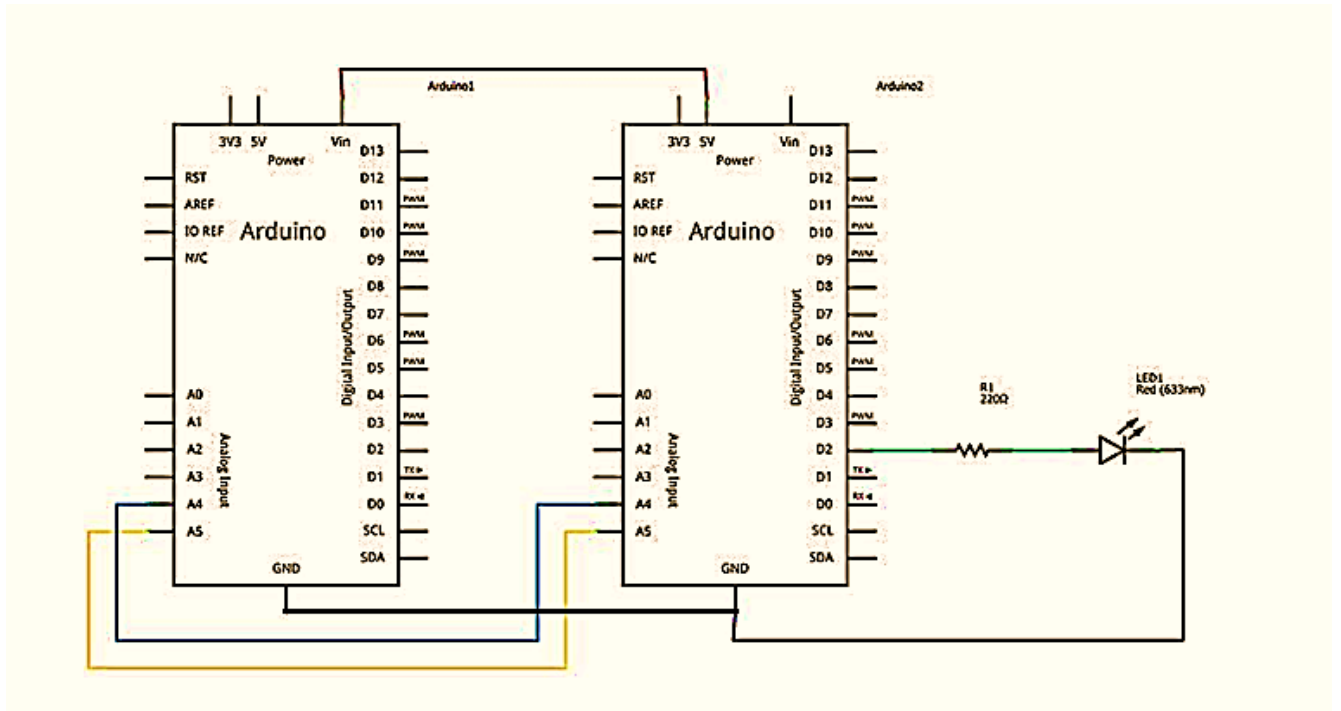


Figure 2

III. 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.

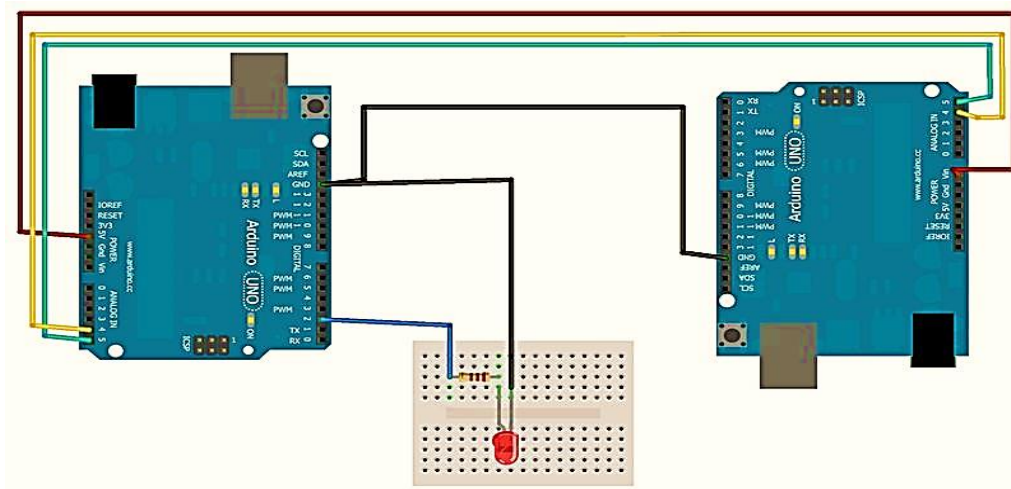


Figure 3

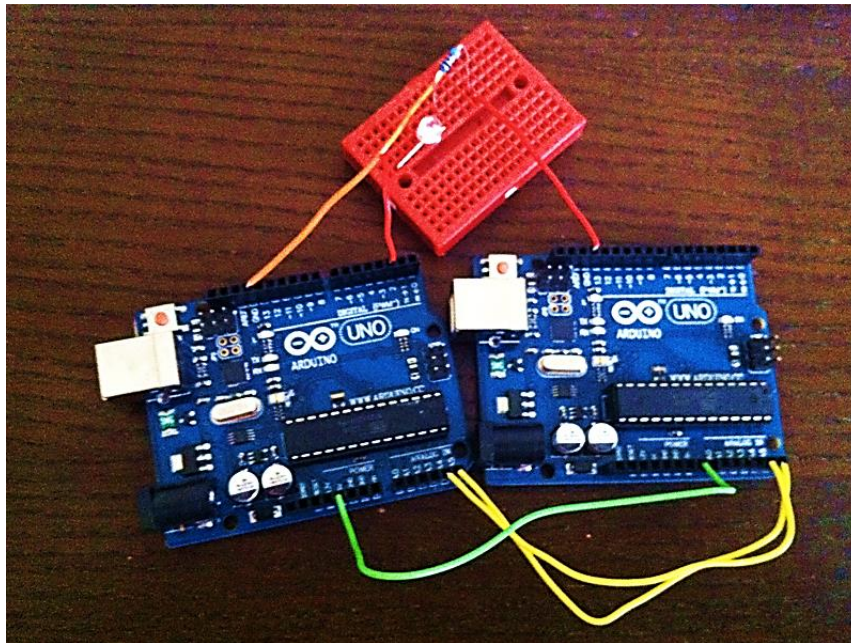


Figure 4

IV. 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.nomdela fonction» .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.
- **beginTransaction(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 messages 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 cartes Arduino durant la programmation !

V. MANIPULATION

1. Compléter le Programme d'Arduino Esclave

Le Arduino esclave définit qu'il a une sortie sur digital 2. Il dit que son adresse est le 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 microcontrôleur se nomme maintenant : L1

void setup( )
{
  Wire.begin(---);          // Rejoindre le bus à l'adresse #4
  Wire.onReceive(receiveEvent); // Préparer une fonction spécifique à la réception de donnée
  Serial.begin(9600);        // Démarrer la liaison série avec le PC
  pinMode(L1, -----);      // L1 est une broche de sortie
}

void loop()
{
  delay(100);
}

// Fonction qui s'exécute si quelque chose est présent sur l'interface

void receiveEvent(int howMany)
{
  int x = Wire.read();       // recevoir un chiffre
  Serial.println(x);         // afficher ce chiffre sur l'interface série
  if(x == 1)
  {
    digitalWrite(L1, -----); // allumer L1
  }
  if(x == 0)
  {
    digitalWrite(L1, -----); // éteindre L1
  }
}
```

2. Compléter le Programme d'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(----);           // Envoyer vers device #4
  Wire.write(---);                        // Envoi un 1
  Wire.endTransmission();                 // Arrêter la transmission
  delay(-----);                         // Attendre 1s
  Wire.beginTransmission(----);           // Envoyer vers device #4
  Wire.write(-----);                    // Envoi un 0
  Wire.endTransmission();                 // Arrêter la transmission
  delay(-----);                         // Attendre 2s
}

```

3. Brancher l'Arduino puis compiler le programme et le téléverser.