

MATRICE de LEDS (La tête à Toto)

La matrice de leds est gérée par le circuit max 7219 qui permet de piloter jusqu'à 8 afficheurs 8 segments. Pour tout renseignement complémentaire se référer au datasheet sur le web.
Une façon (ce n'est pas la seule) de communiquer avec le max 7219 est d'utiliser la liaison série SPI. La bibliothèque est disponible par défaut dans l'IDE Arduino. On l'importe avec :

```
#include <SPI.h>
```

La liaison SPI utilise 4 fils :

MISO (master in slave out)	broche 12 Arduino
MOSI (master out slave in)	broche 11 Arduino
SCK (serial clock), horloge de synchronisation	broche 13 Arduino
SS sélection du mode maître esclave	
SS est à 1 au repos et passe à 0 pour la transmission	

Remarque : pour la matrice de leds Miso ne sert à rien car la matrice ne renvoie rien à Arduino

Le principe de transmission au max7219 est donc le suivant :

On met SS à 0 pour commencer

On envoie une donnée (octet) à une adresse (octet) du max dans l'ordre **adresse puis donnée** :

On remet SS à 1.

Adresses et octets peuvent être envoyés aussi bien en décimal, qu'en hexa ou en binaire.

Exemple 255, 0xff ou B11111111 reviennent au même.

Il est indispensable de créer une fonction pour cela !

Dans le programme fourni en page 2 elle est nommée « ENVOI »

Un certain nombre de registres du max7219 servent aux réglages (voir datasheet) :

0x09 mode de décodage

(transforme par exemple 5 en code binaire pour allumer les bons digits sur un afficheur 8 segments)

0x0A intensité

0x0B nombre d'afficheurs ...etc.

0x0C Shutdown qu'il faut mettre à 1

Les lignes(ou afficheurs) ont pour adresse 1,2....(0x01, 0x02, en hexa)

Voici donc un programme qui permet d'afficher un octet sur la ligne choisie (A, B, C,...).

Par le moniteur série, si on envoie « A125 », on affiche le code correspondant à 125 sur la ligne 1.

« B255 » va allumer toutes les leds de la ligne 2...

On lit le premier caractère avec « read » qui renvoie le code ASCII,

on lit l'entier qui suit avec « parseInt() » et on envoie le tout par SPI

Ensuite, à vous de l'adapter pour gérer la tête à toto et faire la communication avec Python.

Dans le programme ci-après des infos ont été volontairement effacées par toto et remplacées par sa signature. Réparer d'abord cette malversation !

```

//Exemple de contrôle de la matrice de leds par le moniteur série
//MOSI 11 de arduino connecté à DIN (max7219)
//SCK 13 connecté à CLK
//7 (load) connecté à CS, broche déclarée pour SS

# toto <SPI.h>

#define load 7

byte caractere; //contient le caractère de départ de l'envoi
byte nombre; //contient le nombre envoyé

//Fonction qui transfère une ligne(8 bits) à une adresse (ligne de l'afficheur)
void envoi(byte adresse, byte octet) {
    digitalWrite(load, toto);
    SPI.transfer(toto);
    SPI.transfer(toto);
    digitalWrite(load, toto);
}

void setup() {
    Serial.toto (9600); //Démarrage com série
    pinMode(load, toto);
    SPI.begin(); //Démarrage SPI
    digitalWrite(load, toto); //On ne communique pas au départ !
    envoi(0x09, toto); // mode de décodage (chiffres..)
    envoi(0x0A, toto); // Intensité de 0 à f
    envoi(0x0B, toto); // Nombre de digits
    envoi(toto, 0x01); // Shutdown à 1
}

void loop() {
    if ( Serial.available() ) {
        caractere = Serial.read();
        if ( caractere toto & caractere toto ){
            nombre=Serial.parseInt();
            envoi(toto, toto);
        }
        else {Serial.flush();} //pour vider le tampon série si bêtise
    }
}

```