Découverte de la carte Flickr

Demande #12270 : https://forge.clermont-universite.fr/issues/12770

Présentation :

L'objectif du projet est de contrôler l'affichage des leds en utilisant le capteur flick zero et une raspberry pi puis de réaliser un PCB pour les mettre en commun.

Lien carte CAO : https://forge.clermont-universite.fr/projects/cao/repository/revisions/3033

Il manque le circuit imprimé de la carte tactile.

I Listes des composants :

a) capteur Flick zero :

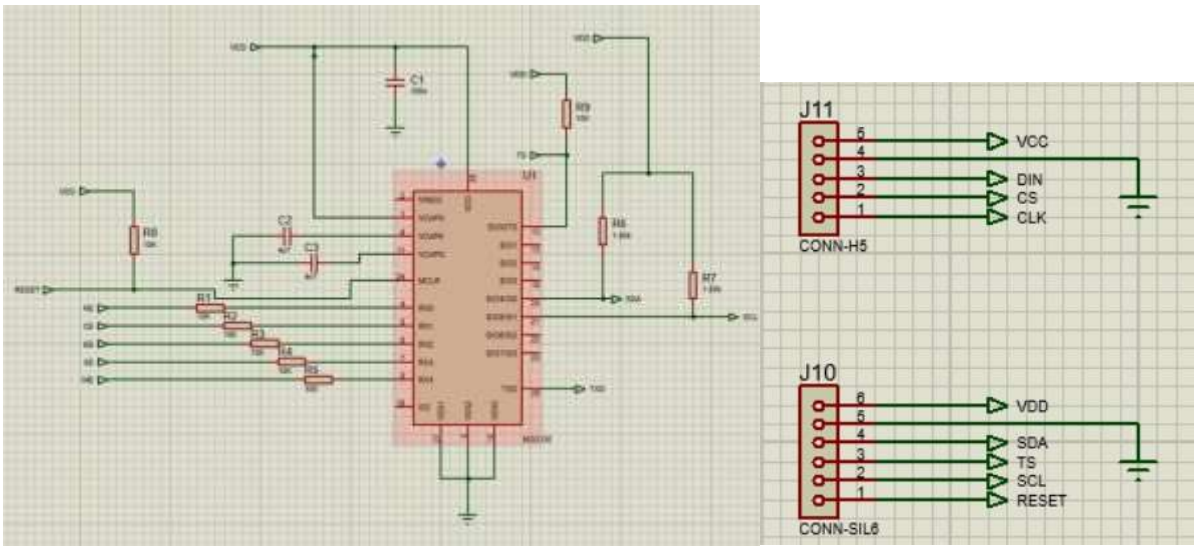
Principe de mesure : Le capteur de proximité émet des champs électromagnétiques ou des faisceaux de rayonnement électromagnétique et observe les changements de champs électriques ou de signaux de retour pour remplir leurs fonctions. L'objet détectable est appelé la cible du capteur de proximité. Le capteur a une "portée nominale de 15 cm. Il peut ajuster et de détecter les rapports de classification de distance dans la plage nominale.

Par conséquent, le capteur de proximité n'a pas de pièces mécaniques et il n'y a pas de contact physique entre le capteur et l'objet détecté, il a donc une **fiabilité élevée** et une **longue durée de vie**.

Flick software setup : https://github.com/PiSupply/Flick

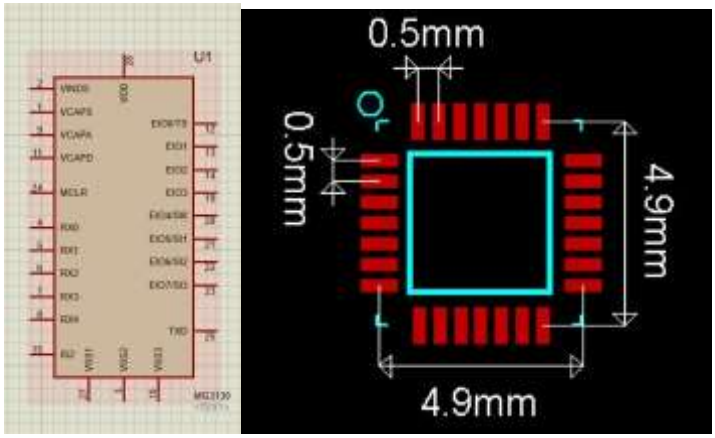
Lien achat : https://fr.farnell.com/pi-supply/ps-flick-zero/flick-zero-gesture-sensor/dp/2687145

Empreinte :



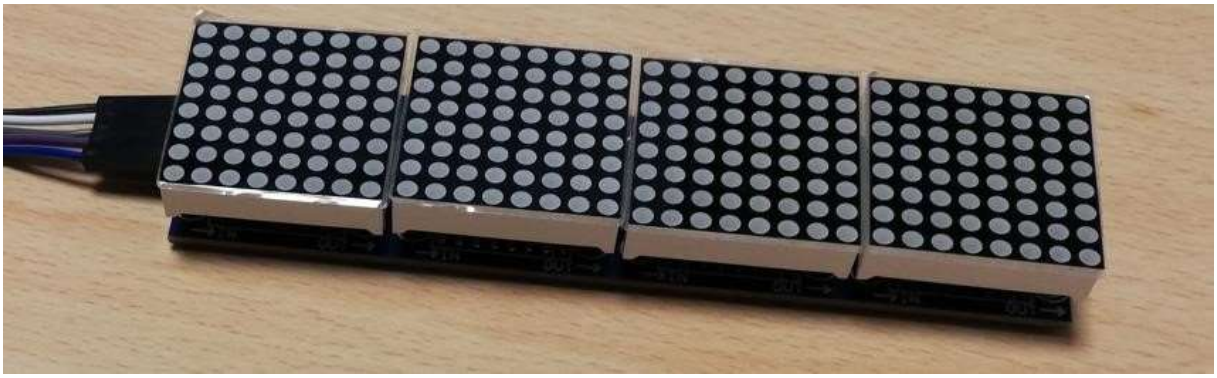
Connections entre flick zero et Raspberry :

Chip Mg3130



Datasheet : https://ww1.microchip.com/downloads/en/DeviceDoc/40001662B.pdf

b) panneau d'affichage : il se compose de 4 matrices de 8x8 leds commandées en courant par 4 Max7219.



Max7219

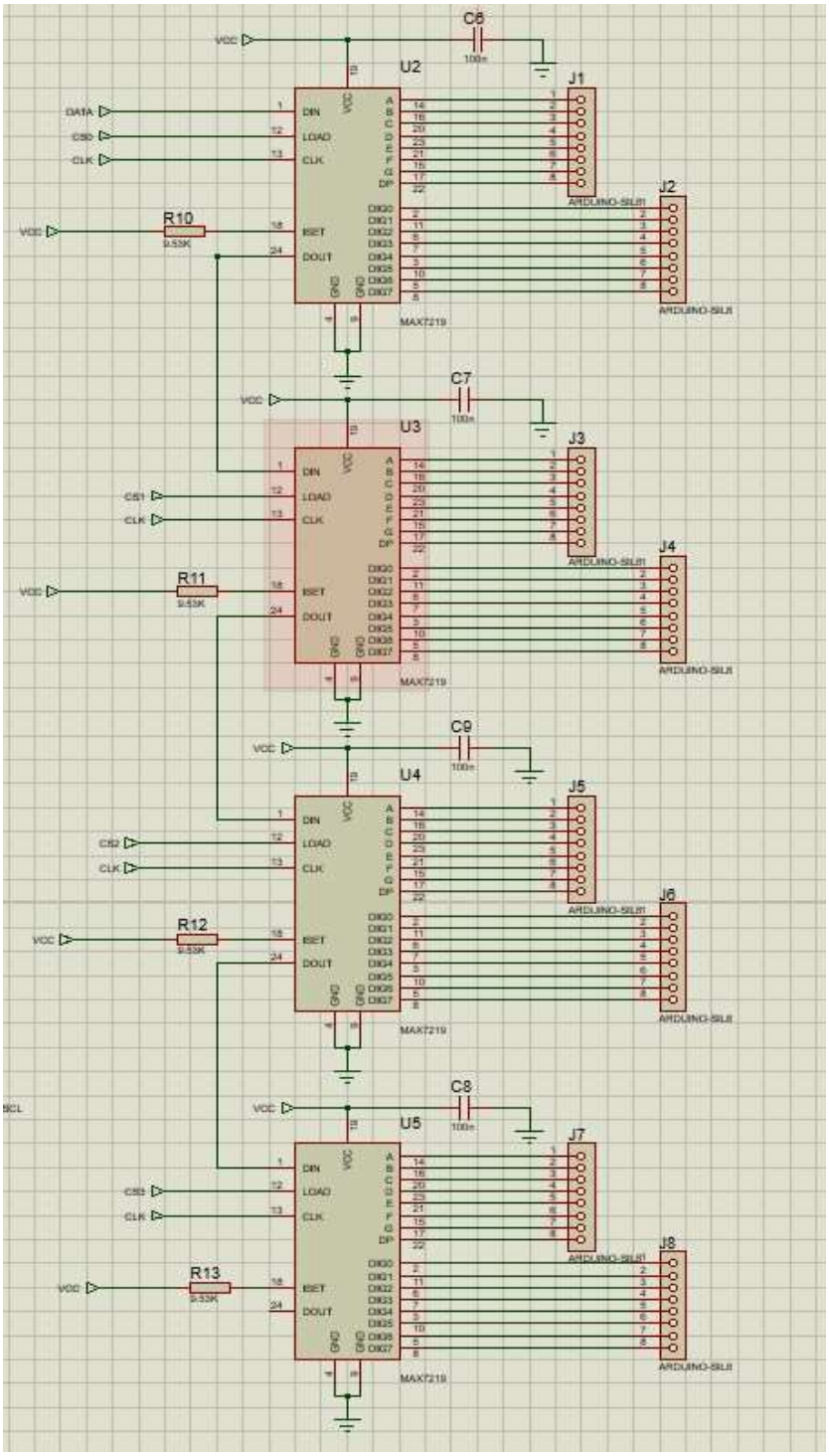
Datasheet : https://datasheets.maximintegrated.com/en/ds/MAX7219-MAX7221.pdf

Driver : https://github.com/rm-hull/max7219.git

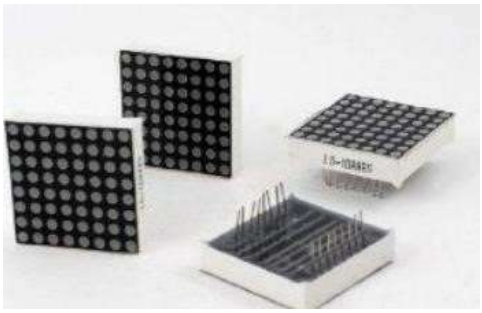
Non disponible inventaire

Lien achat : https://fr.farnell.com/maxim-integrated-products/max7219cng/driver-de-led-8-digit-cc7219/dp/2519433?st=max7219

Empreinte : https://forge.clermont-universite.fr/projects/cao/repository/revisions/3030/entry/trunk/proteus/librairies/etudiants_sch/MAX7219.cmp

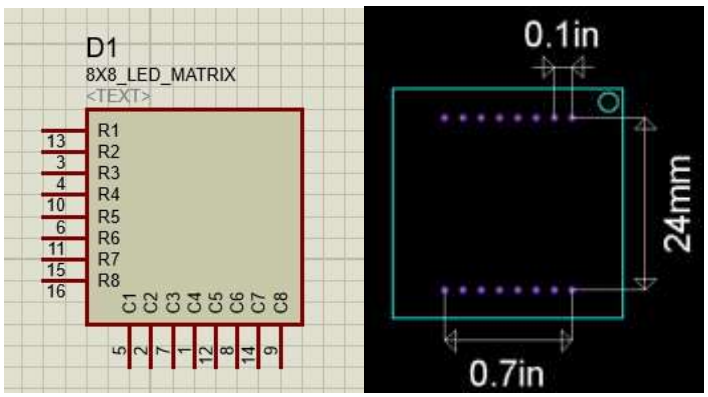


8x8 leds matrice :



Datasheet : http://www.ledtoplite.com/uploadfile/2017/1088/TOP-CC-1088AS.pdf

Lien empreinte : https://forge.clermont-universite.fr/projects/cao/repository/revisions/3038/entry/trunk/proteus/librairies/etudiants_sch/8X8_LED_MATRIX.cmp



Non disponible inventaire

Lien achat : https://www.aliexpress.com/item/32426680871.html?src=google

II Programmation :

a) Principe commande en tension matrice 8x8 leds :

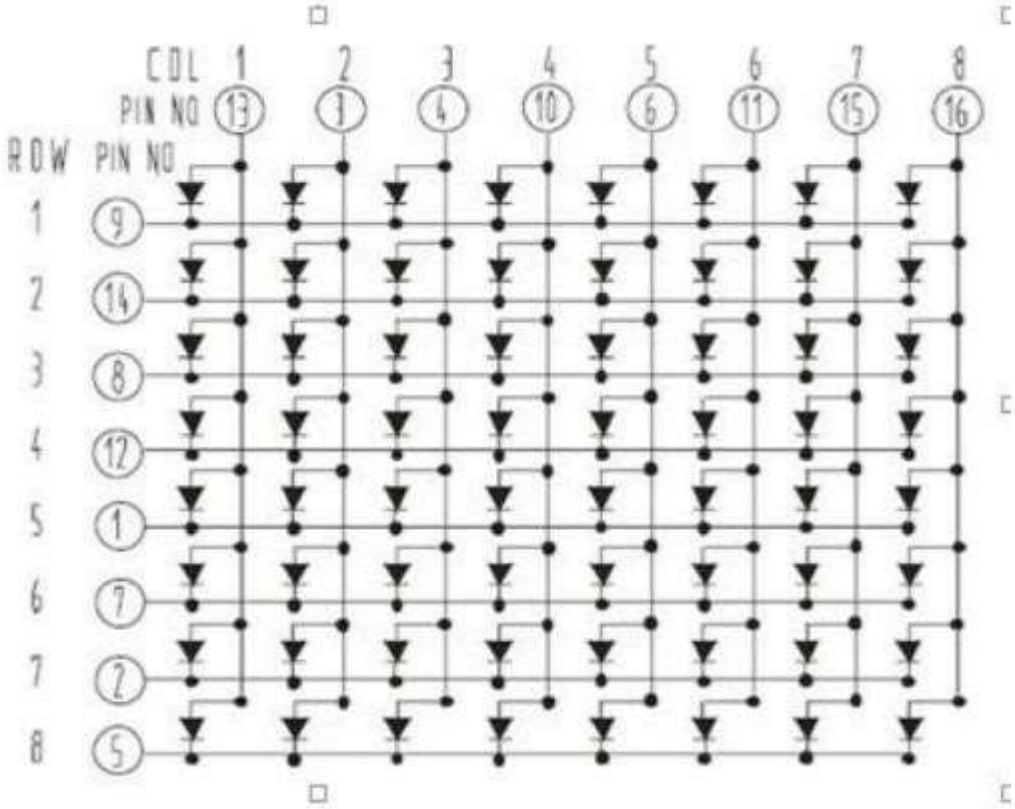
On dispose de 8 lignes et 8 colonnes que l'on commande en tension avec soit les lignes à l'état haut ou bas et inversement pour les colonnes pour déterminer l'adresse d'une led.

Le module utilisé est basé sur MAX7219 avec une liaison série pour les données. Le bus SPI est constitué de trois fils :

CS : Chip Select pour l'activation du modèle

CLK : l'horloge synchrone pour les données. L'horloge définie la vitesse de transmission des données série

DIN : La donnée série sur 8 bits



Intuitivement on peut commander chaque led en mettant sa ligne à l'état bas et sa colonne à l'état haut. Seulement dans notre cas cela revient à commander 16 broches si on veut commander chaque leds ce qui n'est pas optimal.

Pour contourner ce problème, on contrôle donc une unique ligne à chaque fois, en changeant de ligne régulièrement pour tromper la persistance rétienne de l'œil humain. Ce nombre de lignes que l'on rafraichit régulièrement pour créer une image est la « scanline ».

Ce principe c'est le « scanning notamment utilisés dans les vieux écrans cathodiques.

Il faut alors choisir le nombre de ligne à contrôler à tout instant t : pour réellement contrôler 1 ligne à tout instant t dans notre cas il faudrait en contrôler une toutes les 1/8 t. Pour être plus rapide on peut en contrôler 2 toutes les 1/8 t pour passer à ¼ de t.

Exemple de code python pour commander en tension ligne par ligne d'une matrice 6x6 de leds à adapter pour 8x8 puis 4x8 et 4x8.

```
int row[6] = {0,1,2,3,4,5};//板子上把两组线反过来，判断就接到0-5上，行就接到6-12上
int coloumn[6] = {8,9,10,11,12,13};
int appear[6][6] = {{0,1,1,1,1,0},
                    {0,1,0,0,1,0},
                    {0,1,0,0,1,0},
                    {0,1,0,0,1,0},
                    {0,1,1,1,1,0},
                    {0,0,0,0,0,0}};
void setup() {
    for(int i = 0;i < 6;i++){
      pinMode(row[i],OUTPUT);
      pinMode(coloumn[i],OUTPUT);
      digitalWrite(row[i],LOW);//把行设为低电压，把列设为高电压，最终能通就亮
      digitalWrite(coloumn[i],HIGH);
      }
}
void loop() {
    for(int i = 0;i < 6;i++){
      for(int j = 0;j < 6;j++){
        if(appear[i][j] == 1){//遍历数组查是否等于1，
          digitalWrite(row[i],HIGH);
          digitalWrite(coloumn[j ],LOW);//通过这两行代码就通，点亮
          delay(1);
          digitalWrite(row[i],LOW);
          digitalWrite(coloumn[j],HIGH);//最后这两行没太明白，咋又关了呢？高手指点下吧
          }
        }
      }
}
```

fonctions disponibles dans la libraire LedControl.h du Max7219 :

https://www.electronique-mixte.fr/projet-electronique-gestion-dune-matrice-des-led-avec-arduino/

Sources : https://max7219.readthedocs.io/en/stable/install.html

https://skyduino.wordpress.com/2014/01/11/tutoinfo-matrice-de-leds-rgb-partie-1/

https://www.best-microcontroller-projects.com/max7219.html

Auteurs :

-Leng Jiyan

-Huang Shizhi