

Enseignes et afficheurs à LED

Afficheurs matriciels multiplexés

Afficheurs matriciels multiplexés

Pierre-Yves Rochat

Afficheurs matriciels multiplexés

Pierre-Yves Rochat

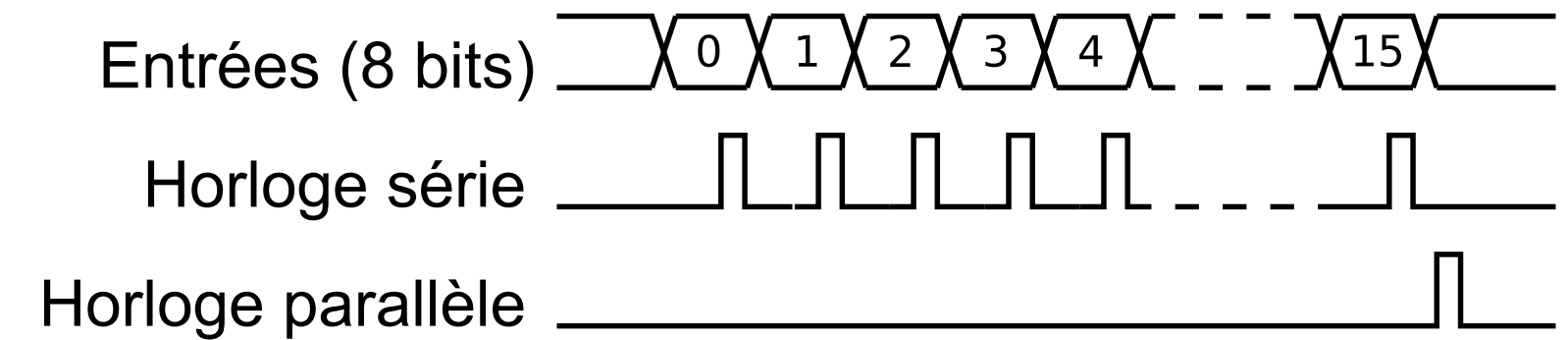
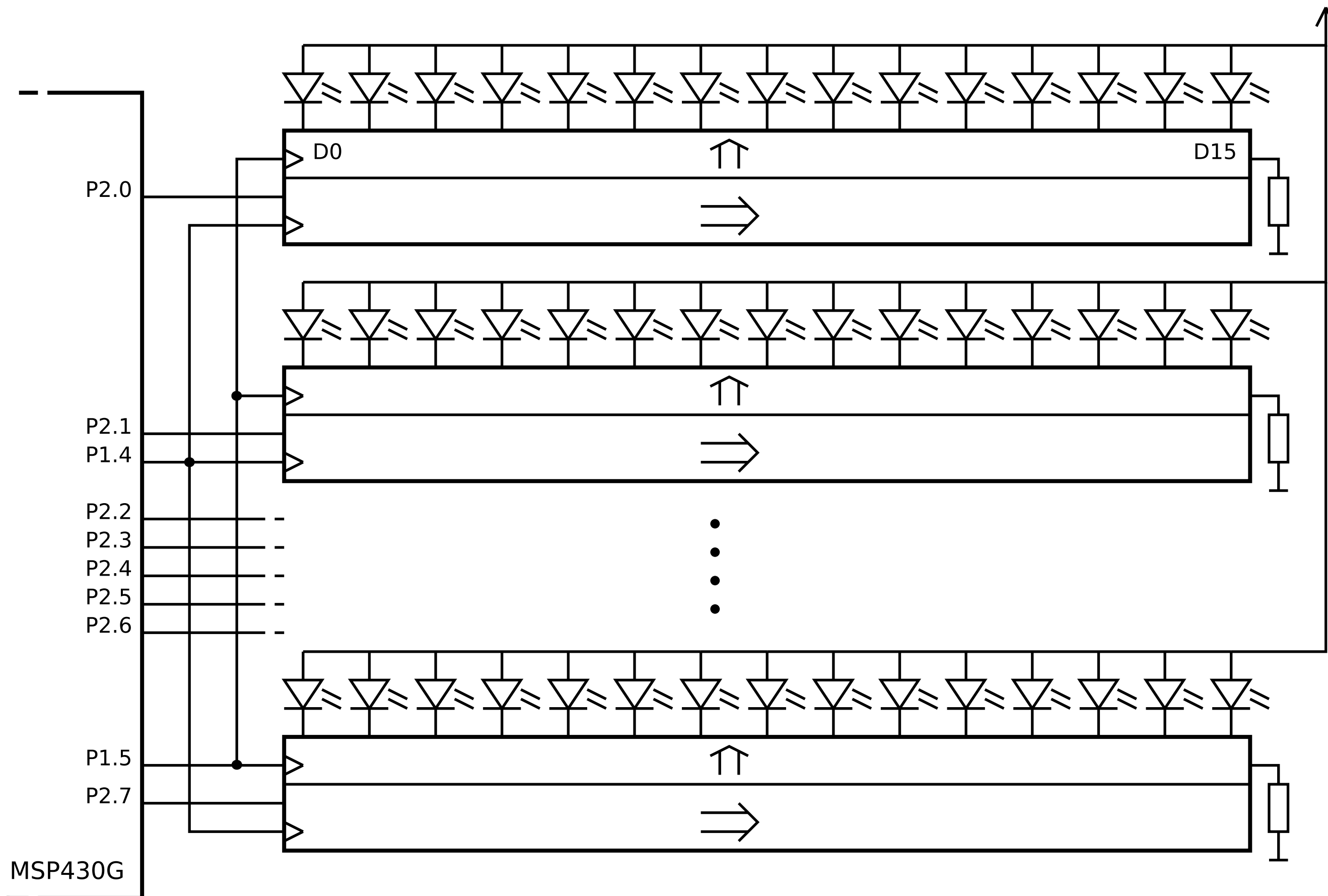
- Augmentation du nombre de LED
- Usage du multiplexage temporel
- Conséquence sur la luminosité
- Programmation

Afficheurs matriciels multiplexés

Pierre-Yves Rochat

- Augmentation du nombre de LED
- Usage du multiplexage temporel
- Conséquence sur la luminosité
- Programmation

Schéma d'un afficheur matriciel

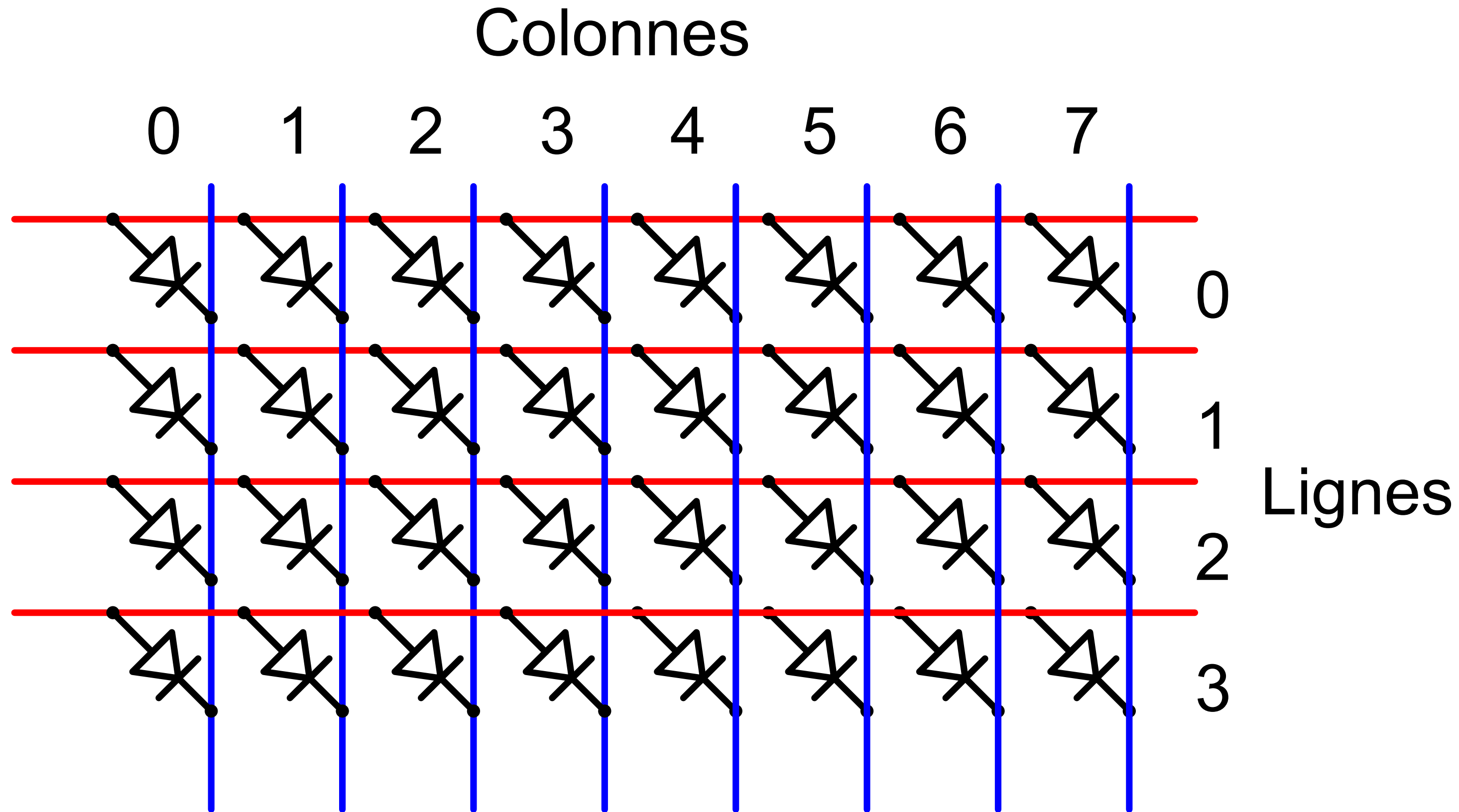


Conséquence de l'augmentation du nombre de LED

- 32 x 32 pixels => 1024 sorties de registre
- 128 registres 8 bits 74HC595 et 1024 résistances
- Trois fois plus pour une matrice RGB !
- Registres 16 bits à sortie à courant constant SUM2016
- 192 circuits intégrés et 192 résistances

- Peut-on obtenir des schémas plus simples ?
- Avec le multiplexage temporel !

Regroupement des anodes et des cathodes par direction



Multiplexage temporel

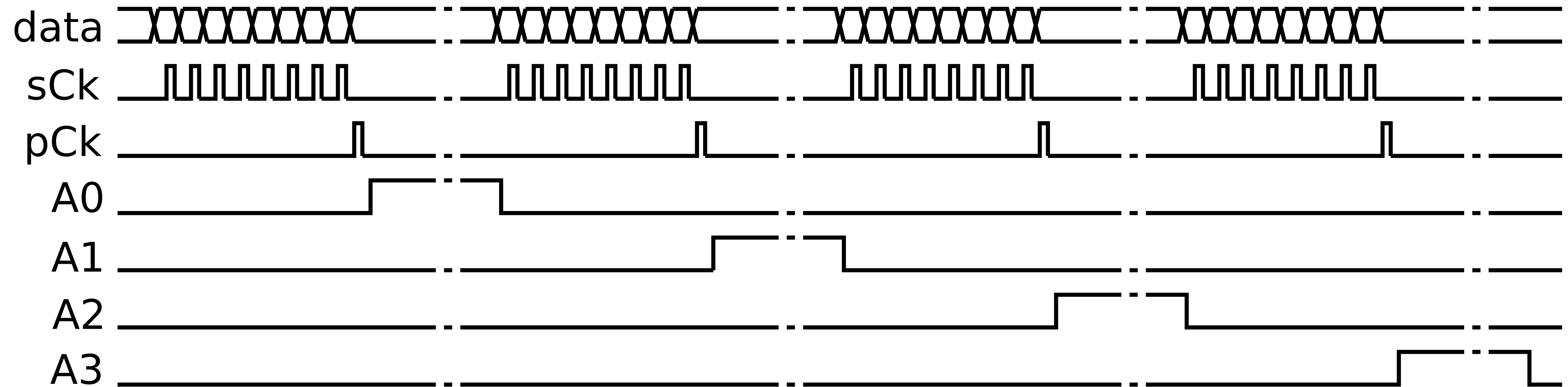
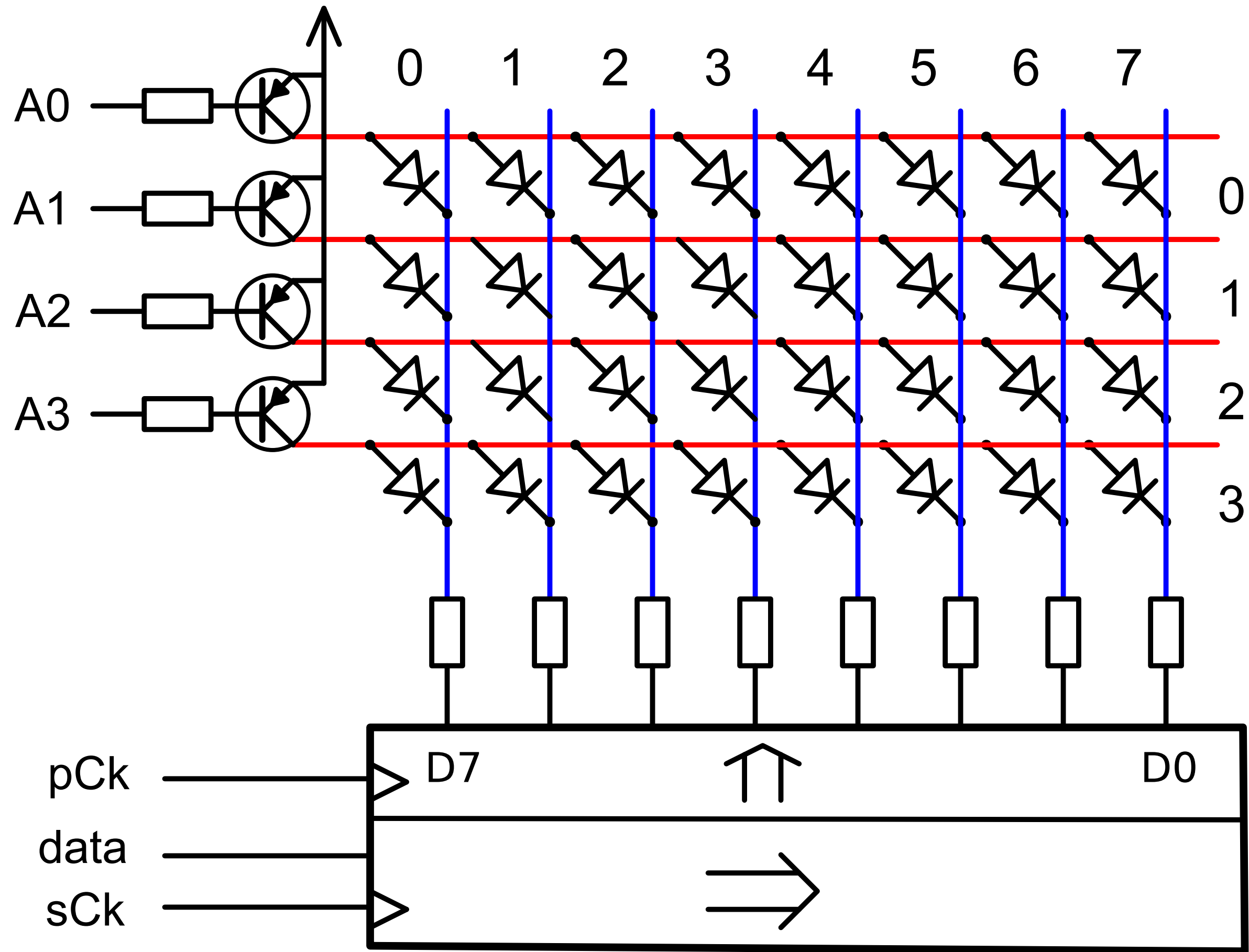


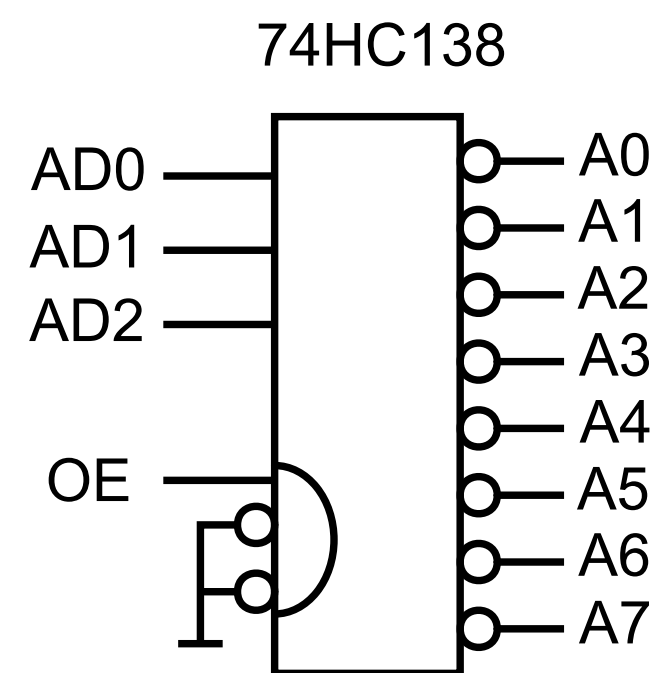
Schéma d'un afficheur matriciel multiplexé



Courants ?

- n anodes
- 1 cathode

Démultiplexeur ou décodeur



- 74HC138

Courant nominal et courant maximal

- Un courant **nominal** est donné par le fabricant
- Le courant **maximal** est supérieur, mais ne peut pas être permanent

- Souvent : 150 % du courant nominal

Comparaisons des architectures

- Multiplexage par 2 :
- Multiplexage par 4 : compromis intéressant
- Multiplexage par 8 et 16 : afficheurs d'intérieur

- Facteurs de multiplexage plus importants : trop peu de luminosité

- Panne d'une LED peut entraîner des perturbations sur les LED voisines

Programme de commande

- Similaire à un afficheur non multiplexé
- Procédure pour un cycle d'affichage
- Base de temps donné par cette procédure

Cycle d'affichage

```

1 uint8_t matrice[4];
2
3 void CyclesMatrice(uint16_t nbCycles) {
4     uint16_t n, x, y;
5     for (n=0; n<nbCycles; n++) {
6         for (y=0; y<4; y++) { // envoi et affichage des 4 lignes
7             for (x=0; x<8; x++) { // envoi des 8 bits d'une ligne
8                 if (matrice[y] & (1<<x) {
9                     DataClear;
10                } else {
11                    DataSet;
12                }
13                SerClockSet; SerClockClear; // envoie un coup d'horloge série
14            }
15            ParClockSet; ParClockClear; // envoie un coup d'horloge parallèle
16            AttenteLigne(); // affichage de la ligne durant 25 ms
17        }
18    }
19 }

```

Modification du programme de commande

```

1 void Ping() {
2   int16_t x=0;
3   int16_t y=0;
4   int8_t sensX=1;
5   int8_t sensY=1;
6   do {
7     AllumePoint(x,y);
8     AfficheMatrice();
9     Attente(DELAI);
10    EteintPoint(x,y);
11    x+=sensX;
12    if(x==(MaxX-1)) sensX=(-1);
13    if(x==0) sensX=1;
14    y+=sensY;
15    if(y==(MaxY-1)) sensY=(-1);
16    if(y==0) sensY=1;
17  } while (!(x==0)&&(y==0));
18 }

```

La procédure de cycle donne la base de temps

```

1 void Ping() {
2   int16_t x=0;
3   int16_t y=0;
4   int8_t sensX=1;
5   int8_t sensY=1;
6   do {
7     AllumePoint(x,y);
8     CyclesMatrice(DELAI); // l'affichage fait office de délai
9
10    EteintPoint(x,y);
11    x+=sensX;
12    if(x==(MaxX-1)) { sensX=(-1); }
13    if(x==0) { sensX=1; }
14    y+=sensY;
15    if(y==(MaxY-1)) { sensY=(-1); }
16    if(y==0) { sensY=1; }
17  } while (!(x==0)&&(y==0));
18 }

```


Afficheurs matriciels multiplexés

- Augmentation du nombre de LED
- Usage du multiplexage temporel
- Conséquence sur la luminosité
- Programmation