

Luminaire 3D

Matrice programmable de 125 LED

Jerry Jacobs

Tout le monde a déjà vu une matrice de LED à deux dimensions, mais la version dont il est question ici est d'un tout autre calibre : elle se compose de cinq matrices superposées pour constituer un cube, donc une matrice tridimensionnelle, dont on peut commuter chaque LED séparément.

Caractéristiques techniques du matériel

- 125 LED en matrice 3D spéciale
- Microcontrôleur ATMEGA32 avec horloge interne à 1 MHz
- Connecteur ISP à 10 broches pour reprogrammer la puce
- 5 transistors BC337 pour commuter les couches
- 25 transistors BC547 pour la commutation des colonnes.

La plupart des gens sont fascinés par les LED qui clignotent, mais le plus souvent, il s'agit de quelques LED ou d'un afficheur. Le cube à LED est d'une toute autre essence, il jette un pont vers une autre dimension. C'est un véritable affichage 3D dont toutes les LED sont commandées individuellement.

Le cœur de ce cube scintillant est un microcontrôleur AVR.

On trouve aisément ces contrôleurs et de bons outils de développement pour eux existent en source libre, et pas seulement pour Windows, mais également pour les systèmes d'exploitation Linux et Mac.

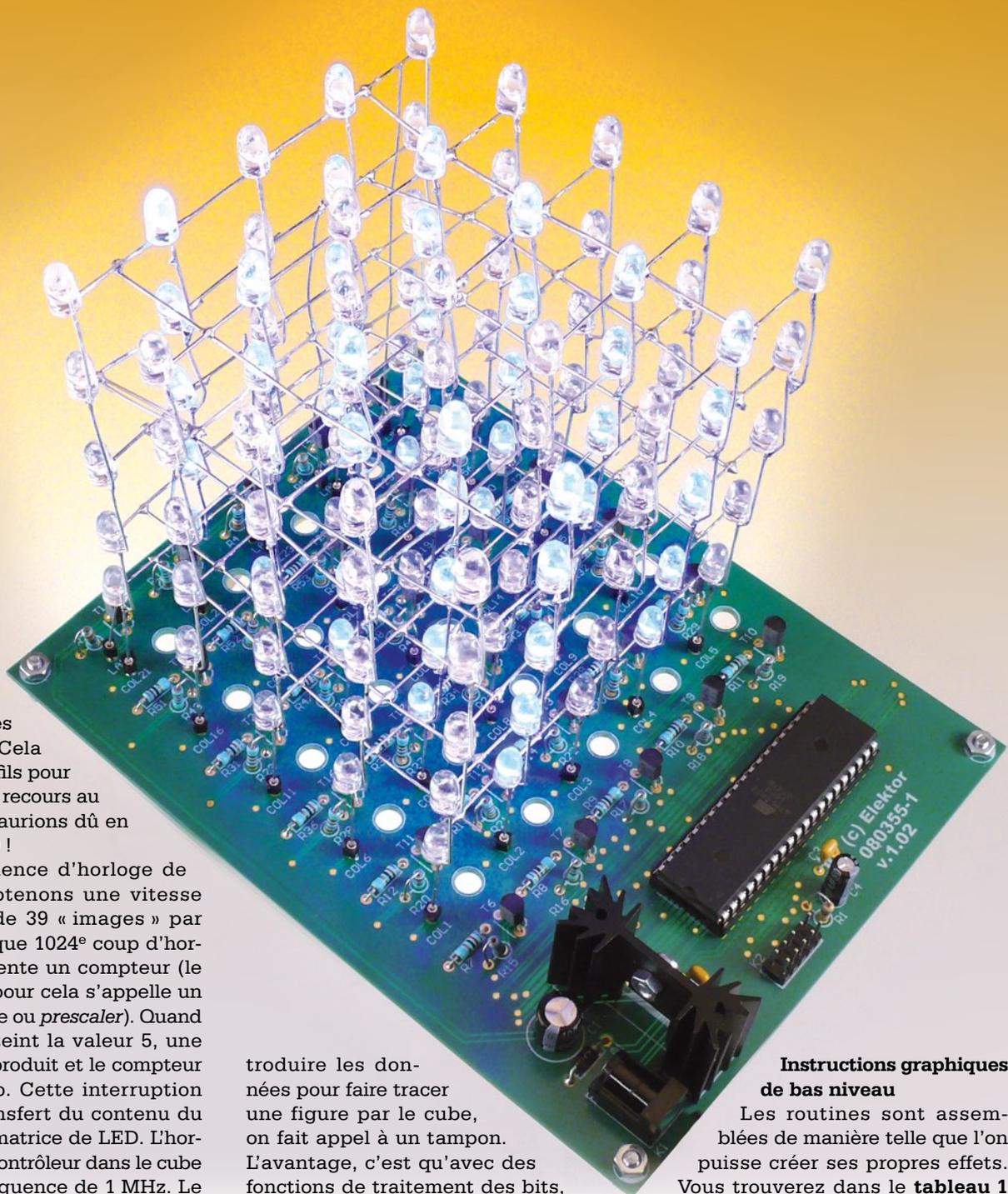
Fonctionnement

Vous imaginez qu'avec 125 LED, il faudra une quantité impressionnante de fils pour les allumer une par une ; or ce n'est pas nécessaire. Comme les signaux sont multiplexés, on épargne pas mal de conducteurs. Une « couche », c'est-à-dire l'ensemble des 25 LED qui se situent à la même hauteur dans chaque colonne, peut être commandée par une seule ligne. Cela donne donc 26 fils de signaux. S'il avait fallu les commander séparément, nous aurions dû mettre en service 50 lignes.

Pour faire s'allumer une LED, nous commutons la tension positive sur la couche voulue et nous sélectionnons

Tableau 1. Commande des couches et colonnes

PORT A							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Colonne 8	Colonne 7	Colonne 6	Colonne 5	Colonne 4	Colonne 3	Colonne 2	Colonne 1
PORT B							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Colonne 25	-	-	Couche 5	Couche 4	Couche 3	Couche 2	Couche 1
PORT C							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Colonne 16	Colonne 15	Colonne 14	Colonne 13	Colonne 12	Colonne 11	Colonne 10	Colonne 9
PORT D							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
Colonne 24	Colonne 23	Colonne 22	Colonne 21	Colonne 20	Colonne 19	Colonne 18	Colonne 17



une colonne. Notre cube compte 5 couches et 25 colonnes. Cela signifie donc 30 fils pour 125 LED. Sans le recours au multiplex, nous aurions dû en ordonnancer 250 !

Avec une fréquence d'horloge de 1 MHz, nous obtenons une vitesse de récurrence de 39 « images » par seconde. À chaque 1024^e coup d'horloge, on incrémente un compteur (le diviseur utilisé pour cela s'appelle un diviseur préalable ou *prescaler*). Quand ce compteur atteint la valeur 5, une interruption se produit et le compteur est remis à zéro. Cette interruption provoque le transfert du contenu du tampon vers la matrice de LED. L'horloge interne du contrôleur dans le cube tourne à une fréquence de 1 MHz. Le logiciel s'occupe de compter les 1024 coups d'horloge. Le diviseur préalable réduit la vitesse de l'horloge interne. Ceci conduit à une fréquence de rafraîchissement de 195 Hz pour l'ensemble du cube. Comme nous avons cinq couches, divisons cette vitesse par cinq et nous obtenons 39 Hz par couche.

Logiciel

Le logiciel, qui est en fait du micrologiciel, est rédigé en C et nous pouvons le compiler avec **avr-gcc** [1]. Consulter sa documentation est comparable à la visite d'un site Internet. C'est Doxygen [2] qui l'a rendu possible.

Tampon

Comme il est assez compliqué d'in-

roduire les données pour faire tracer une figure par le cube, on fait appel à un tampon. L'avantage, c'est qu'avec des fonctions de traitement des bits, on peut les manipuler sans avoir à les écrire en sortie, c'est la routine d'interruption qui s'en charge. Le tampon, exactement comme le cube, a plusieurs dimensions et vous pouvez « dessiner » la figure dans le tampon, la routine d'interruption fera le reste.

Interruption

L'interruption dans le code sert donc à rafraîchir l'image sur le cube 39 fois par seconde. Cette interruption, à l'aide d'une autre fonction, s'empare des valeurs que vous avez indiquées dans le tampon et les transfère à la bonne place en passant par les ports correspondants.

Nous avons fait usage de masques de bits qui permettent de ne considérer que les bits qui changent d'état.

Instructions graphiques de bas niveau

Les routines sont assemblées de manière telle que l'on puisse créer ses propres effets.

Vous trouverez dans le **tableau 1** sur quelles broches se situent les connexions vers le cube. Cela simplifie la vie quand on veut réaliser vite ses propres graphismes et évite aux débutants de commencer par emmagasiner la signification des masques de bits, des décalages de bits et autres fonctions délicates. Ces instructions de niveau inférieur sont définies dans `draw.h`, le fichier d'interface pour la commande de chaque colonne et chaque couche, par port et par bit. Ci-dessous, quelques exemples de la manière dont on peut utiliser les fonctions.

Pour commander une rangée sur une couche déterminée, nous utiliserons :

```
set_row(ROW_1, LAYER_1);
clear_row(ROW_1, LAYER_1);
toggle_row(ROW_1, LAYER_1);
```

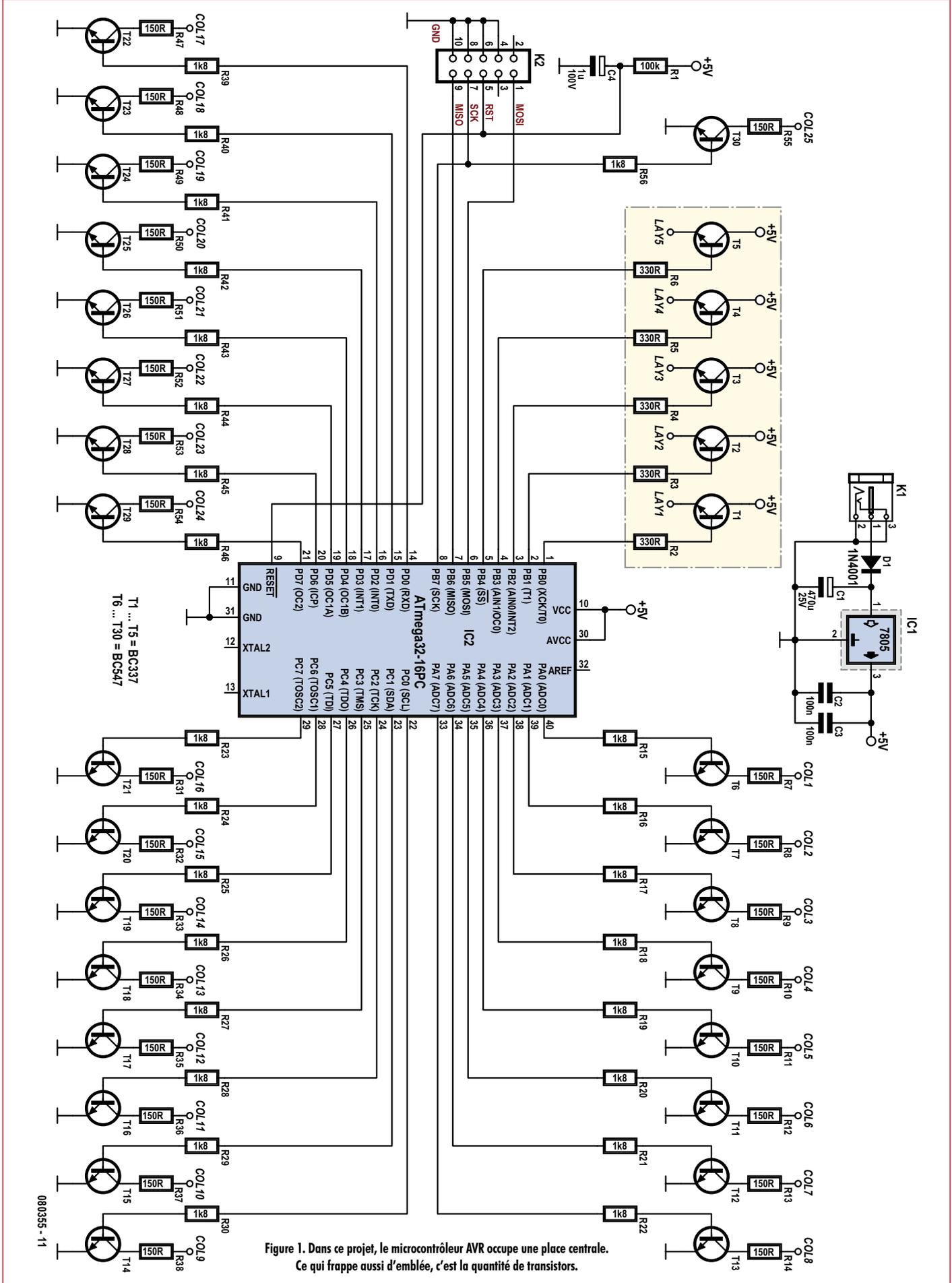


Figure 1. Dans ce projet, le microcontrôleur AVR occupe une place centrale. Ce qui frappe aussi d'emblée, c'est la quantité de transistors.

Pour allumer et éteindre une colonne, on peut employer les fonctions suivantes :

```
set_column(COLUMN_1, ON);
set_column(COLUMN_1, OFF);
```

Nous avons utilisé des dénominations pratiques telles que ON, OFF et COLUMN_1. Ce sont des noms définis, que l'on appelle aussi des macros, qui portent des valeurs fixes. Ainsi, ON a la valeur 1 et OFF, la valeur 0.

On peut faire se succéder toutes ces fonctions pour dessiner la figure souhaitée. Comme nous n'avons pas ici la place pour vous montrer des exemples de tous les codes sources, ils sont à votre disposition pour un téléchargement sur le site d'Elektor.

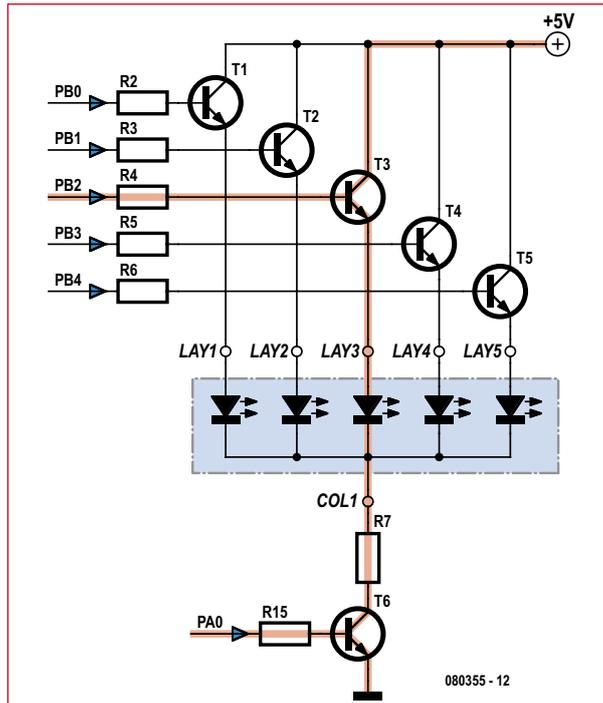


Figure 2. Cette figure décrit le trajet du courant à travers les LED de la première colonne.

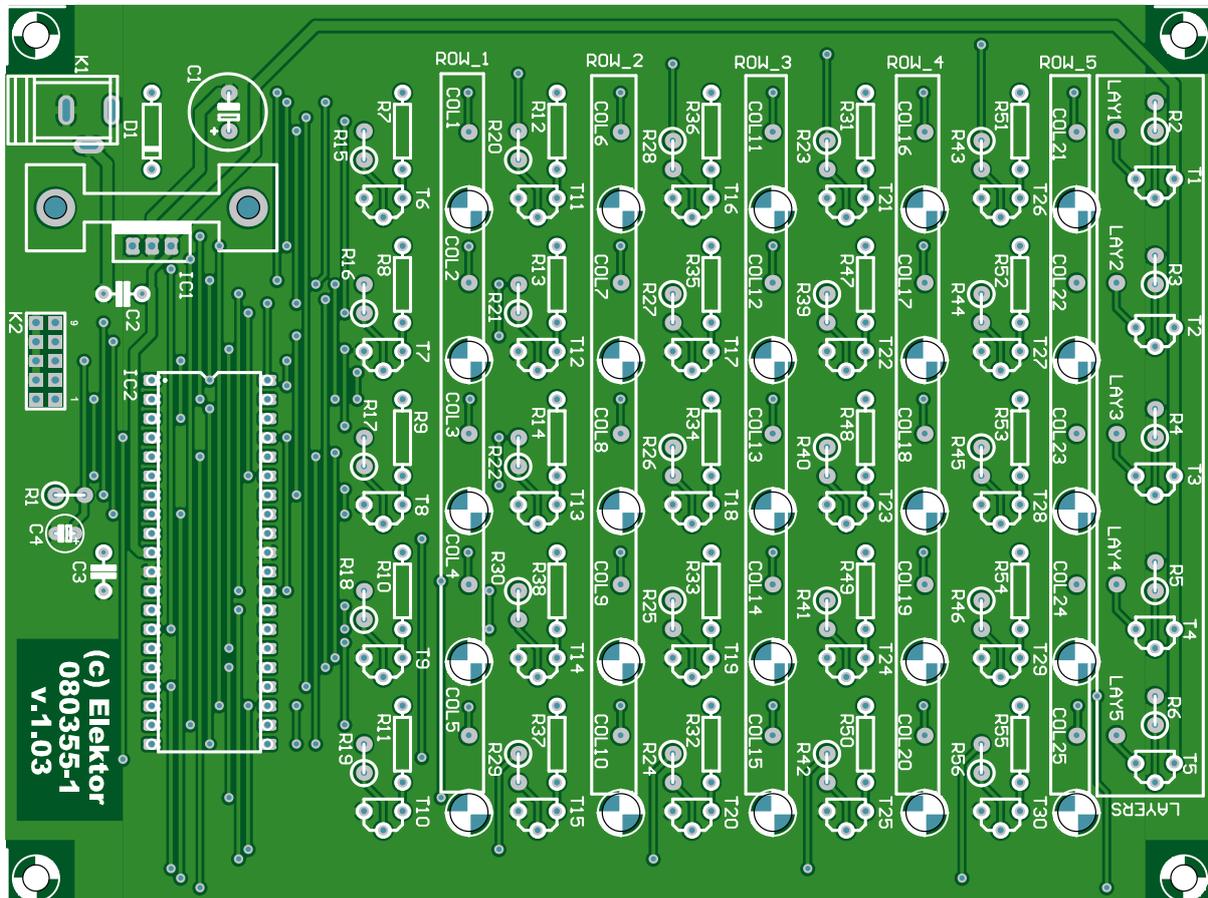
Matériel

La réalisation fait usage de composants « traversants » normaux. Malgré quoi, la platine reste parfaitement compacte. Pour l'alimenter, on peut se servir d'un adaptateur secteur d'au moins 9 V sous 600 mA. IC1 est un 7805 qui assure la stabilisation de tension et la diode D2 protège la carte contre une inversion de polarité.

Les transistors T1 à T5 permettent d'envoyer la tension d'alimentation sur les différentes couches. Ce sont les transistors T6 à T25 qui commutent les colonnes. Le courant traverse ces derniers transistors pour rejoindre la masse et le circuit se referme via les LED (cf. figures 1 et 2).

La valeur des résistances des colonnes dépend de la chute de

Figure 3. La sérigraphie des composants sur la face supérieure montre une disposition ordonnée de la platine.



Liste des composants

Résistances

R1 = 100 kΩ
 R2 à R6 = 330 Ω
 R7 à R14, R31 à R38, R47 à R55 = voir
 texte
 R15 à R30, R39 à R46, R56 = 1,8 kΩ

Condensateurs

C1 = 470 μF/25 V
 C2, C3 = 100 nF
 C4 = 1 μF/100 V

Semi-conducteurs

IC1 = 7805
 IC2 = Atmega32
 D1 = 1N4001
 T1 à T5 = BC337
 T6 à T30 = BC547

125 LED pour le cube

Divers

K2 = Embase 2x5, au pas de 2,54 mm
 Support pour IC2 (DIL40)
 Radiateur TO-220 5°C/W pour IC1
 4 boulons M3x5 avec entretoises hexagonales de 10 mm
 Prise 2,5 mm pour adaptateur secteur
 Platine 080355-1 disponible via www.elektor.fr

tension aux bornes des LED. Nous partons du principe qu'une LED demande 20 mA. C'est aussi le courant qui va circuler dans toute la colonne. La tension procurée à la couche est de 5 V. Voici comment calculer les résistances :

$$R = (5 V - U_{LED}) / 20 \text{ mA}$$

Pour reprogrammer le cube, on peut brancher l'interface ISP à K2. Le tableau 1 montre la correspondance entre colonnes, rangées et bits.

(080355-1)

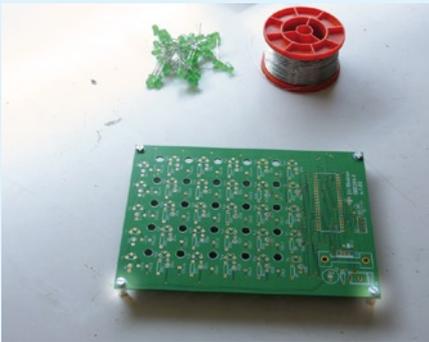
Liens :

[1] AVR-GCC Toolchain:

- Pour Windows: <http://winavr.sourceforge.net>
 - Pour Mac: <http://www.obdev.at/products/avrpack>

[2] Doxygen :

<http://www.doxygen.org>



Au travail !

1^{re} étape

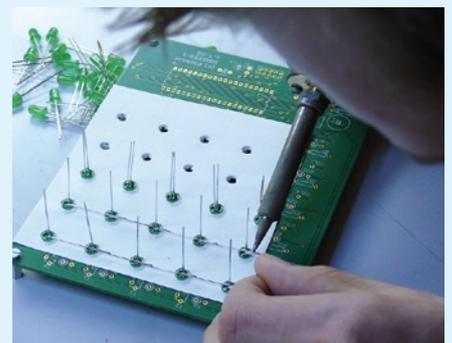
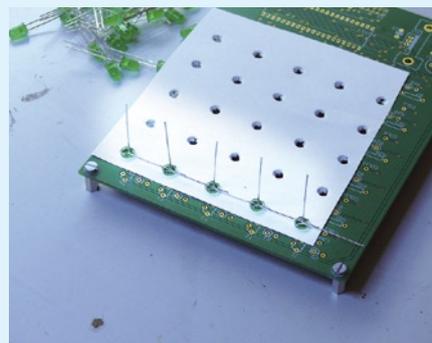
On commence par fixer les entretoises sous la platine puis on coupe une feuille de papier aux bonnes dimensions pour arrimer les LED au bon endroit.

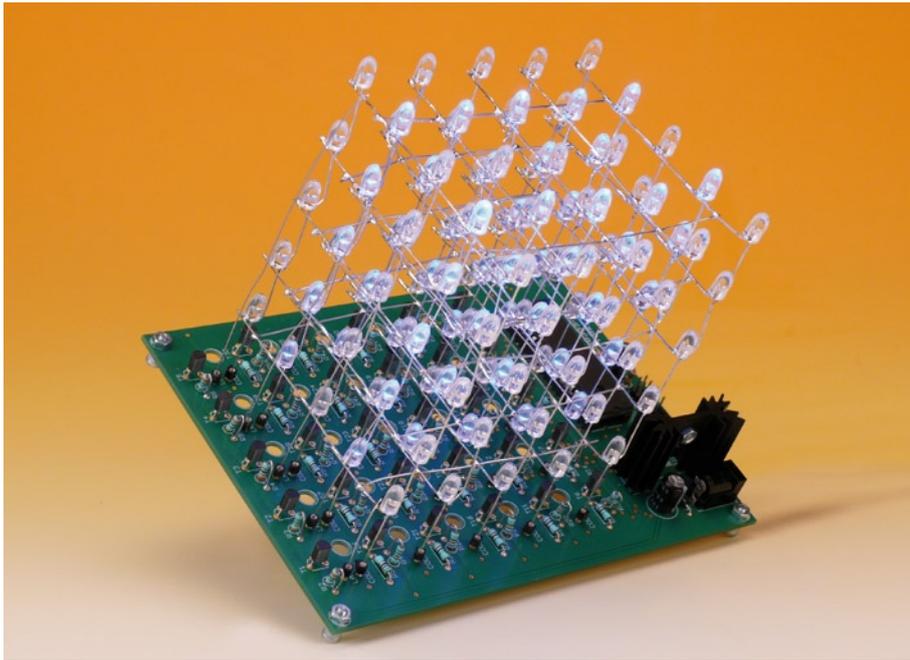
2^e étape

Nous plaçons ensuite cinq LED dans la rangée supérieure avec l'anode (la broche longue) vers le haut et la cathode (la broche courte) vers le bas. Nous plions alors la première LED vers la gauche et nous passons à la deuxième LED. Elle devra être soudée à la première. Nous répétons l'opération jusqu'à ce que la rangée soit finie. Il nous faut cinq LED pour chaque niveau. Nous répétons le même processus jusqu'à la cinquième rangée. Quand les cinq rangées sont faites, on relie ensemble à l'aide de deux fils verticaux les anodes courbées.

3^e étape

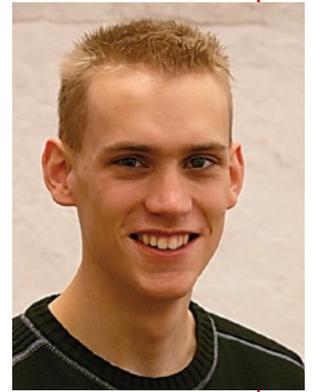
Quand les cinq couches sont soudées, il faut encore les souder les unes aux autres pour former un ensemble. Cela se fait en plaçant une couche sur la platine et en recourbant les 25 colonnes, environ 3 mm vers les anodes de la couche suivante. La deuxième couche se place sur la première et nous soudons ensemble les couches aux quatre coins, en réservant le même écartement entre les couches.



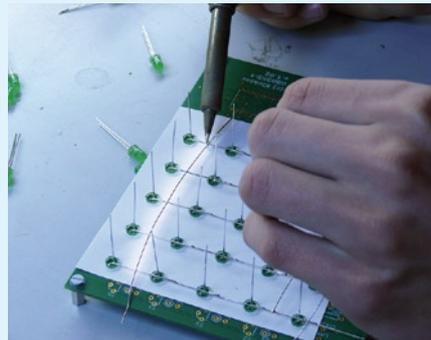
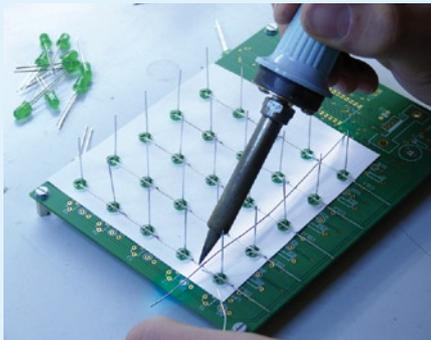


L'auteur

Jerry Jacobs, né à Heerlen (Pays-Bas) en 1989, est étudiant en troisième année de Télécommunication/ICT à l'université ROC Leeuwenborg de Sittard. Il



a toujours été fasciné par le fonctionnement des ordinateurs et par l'électronique. Il est aussi un grand fan de Linux. La réalisation décrite dans cet article est son projet de stage qu'il a achevé pendant sa période de stage chez Elektor.

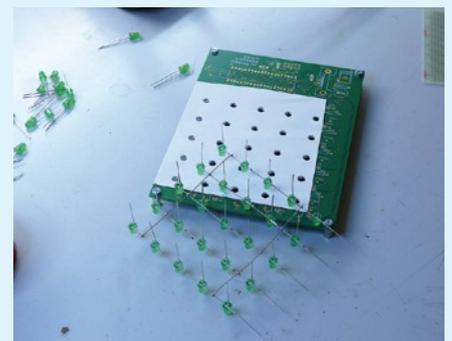


4^e étape

À présent, nous installons les composants sur la platine imprimée. Veillez à ne pas intervertir les BC337 et les BC547 ! En dernier lieu, montez le régulateur de tension avec son radiateur.

5^e étape

Il ne reste plus qu'à relier les couches aux transistors. T1 se branche à la couche inférieure et T5 à celle du haut. Nous employons pour cela des fils de cuivre étamés.



6^e étape

On programme alors le code de l'exemple dans la puce. Ce code exécutable ainsi que le code source sont disponibles au téléchargement sur le site d'Elektor, où vous pouvez aussi commander la platine.

