

## Bico64c – 64 LED de 2 couleurs pour s'amuser

BicoKit name is also used for our new educational kit (May 2018).

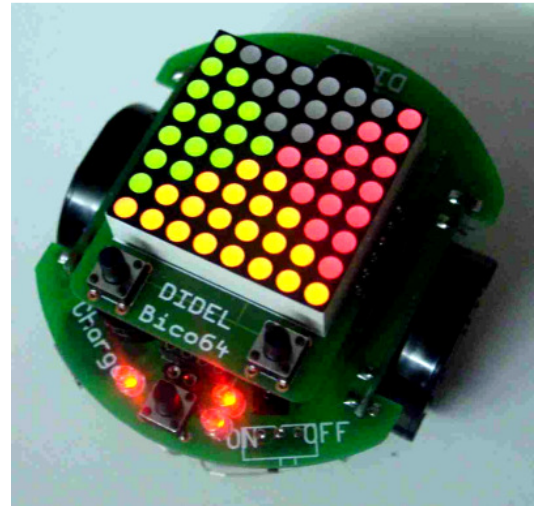
See [www.didel.com/BicoKitEn.pdf](http://www.didel.com/BicoKitEn.pdf) and <https://www.didel.com/BicoKitAssembly.pdf>

En français <https://www.didel.com/BicoKit.pdf> et <https://www.didel.com/BicoKitMontage.pdf>

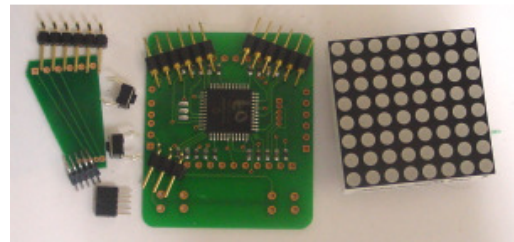
Le circuit **Bico64** affiche en vert et rouge, avec du jaune en mélangeant.

Il a été développé pour le Wellbot, mais il a un processeur propre et peut s'utiliser seul et afficher des motifs fixes ou animés. La programmation du microcontrôleur PIC 16F884 se fait avec un compilateur qui supporte ce processeur. Didel propose un ensemble de routines de base et des programmes de démo écrits en assembleur CALM avec l'environnement de développement SmileNG, qui est gratuit.

Le microcontrôleur rafraîchit l'écran en copiant une table en mémoire. 8 à 32 tables peuvent être mémorisées selon la complexité du programme.

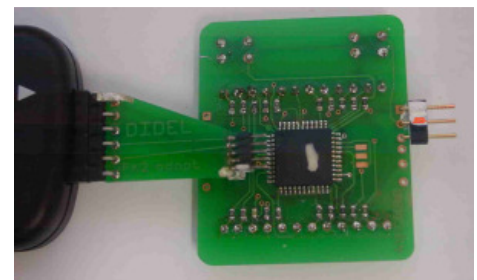


Le kit est facile à souder, puisque les composants SMD sont en place. L'adaptateur pour programmer avec le Picket2 est fourni. Un adaptateur pour MP-Lab existe. Le processeur est programmé pour afficher le motif de la photo qui montre comment le vert et rouge se mélangent. A vous de faire plus joli !



Le circuit se programme avec le Picket2 et l'adaptateur. Les étapes de programmation, assemblage-charge dans le PIC se font sans temps perdu à se connecter-déconnecter.

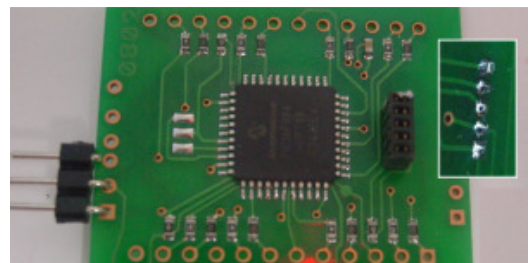
Quant le logiciel est au point, une alimentation 4-6V est branchées sur la prise B. La consommation est de 10-20mA, maximum 30 mA. Avec un courant si faible, l'utilisation ne peut pas se faire dans un local très éclairé. Etre plus lumineux implique quantité de transistors ou des circuits spéciaux qui augmentent le prix.



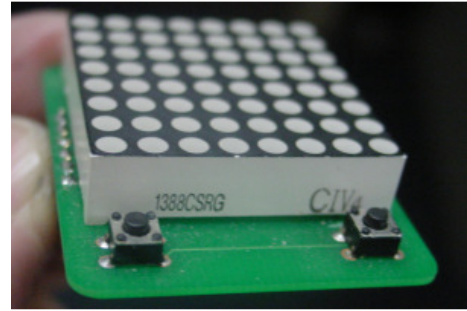
La programmation dans une première étape consiste à inventer des motifs et les appeler avec le poussoir, ou créer des enchaînements décoratifs. En coupler plusieurs qui se synchronisent est envisageables puisque trois lignes sont accessibles sur le connecteur.

### Assemblage

Pour le montage comme unité à mettre sur un WellBot, voir [www.didel.com/wellbot/WeBico.pdf](http://www.didel.com/wellbot/WeBico.pdf)  
Comme unité indépendante, il faut souder d'abord le connecteur de programmation et la prise B pour l'alimentation. Couper les pattes du connecteur de programmation à raz (0.5mm) pour que cela ne gêne pas le bloc de LEDs.



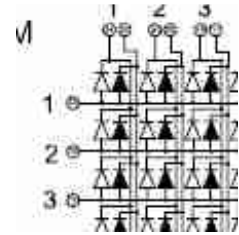
Le connecteur d'alimentation a une patte non connectée. Voir <http://www.bricobot.ch/docs/PriseB.pdf>  
 Insérer et souder l'affichage et les poussoirs.  
 L'inscription sur le bord de l'affichage donne l'orientation.



Les barrettes mâles à 6 pôles sont prévues si vous voulez une fois vous brancher sur un WellBot. Il faudra alors dessouder la priseB.

### Organisation des LEDs

Les leds sont câblées en x-y, avec les cathodes communes (type CSR). Attention, il existe aussi le modèle avec des anodes communes (type ASR). Le processeur peut heureusement être programmé pour les deux, mais il faut savoir lequel.  
 Pour allumer une diode, il faut sélectionner une ligne et une colonne.  
 On peut sélectionner les deux couleurs en même temps. Pour avoir la même intensité lumineuse, la LED verte a besoin de plus de courant que la rouge, d'où une résistance série plus faible.

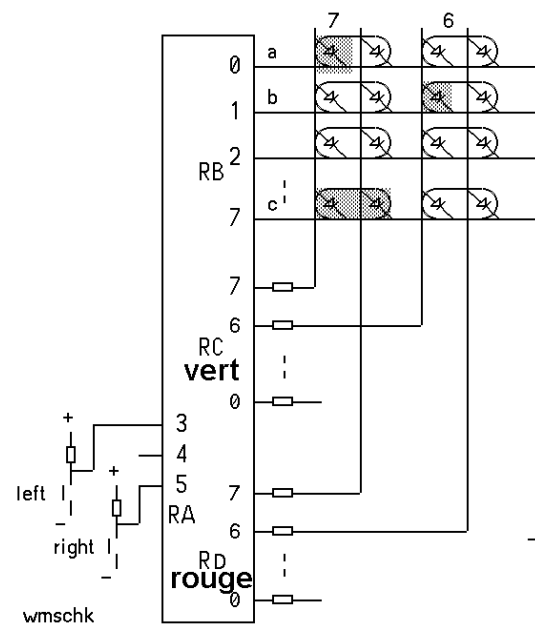


Le processeur choisi a 3 ports 8 bits complets, plus des bits sur le portA qui permettent de communiquer avec les poussoirs sur la carte, et avec le robot ou d'autres afficheurs . On ne peut allumer ensemble que les LEDs d'une rangée.

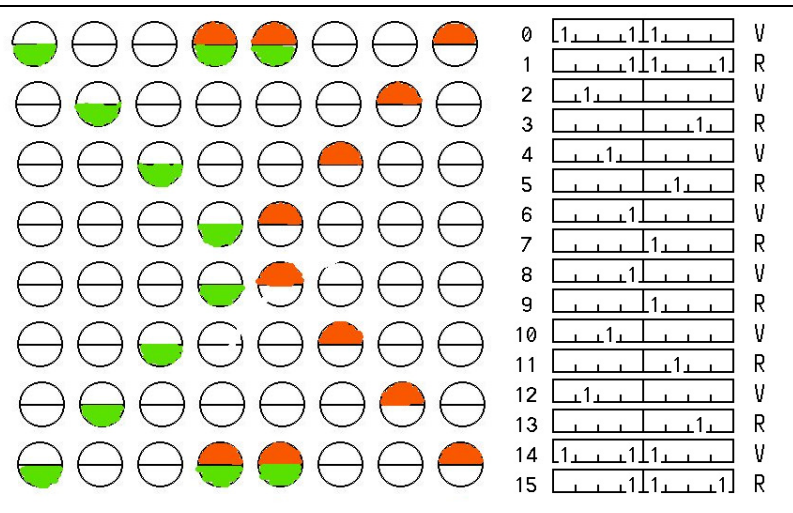
Le port RB est actif à zéro. Les ports C et D sont actifs à 1. On peut tout allumer en mettant RB à zéro et RC RD à un. Pour un motif quelconque, il faut balayer.

Le processeur sélectionne la ligne et active les 2x8 bits correspondant aux LEDs de la ligne pour allumer ce qui est voulu. 2ms plus tard, on éteint, passe à la ligne suivante et allume pour 2ms. On recommence donc après 16ms, et avec la persistance rétinienne, on voit une image complète est stable.

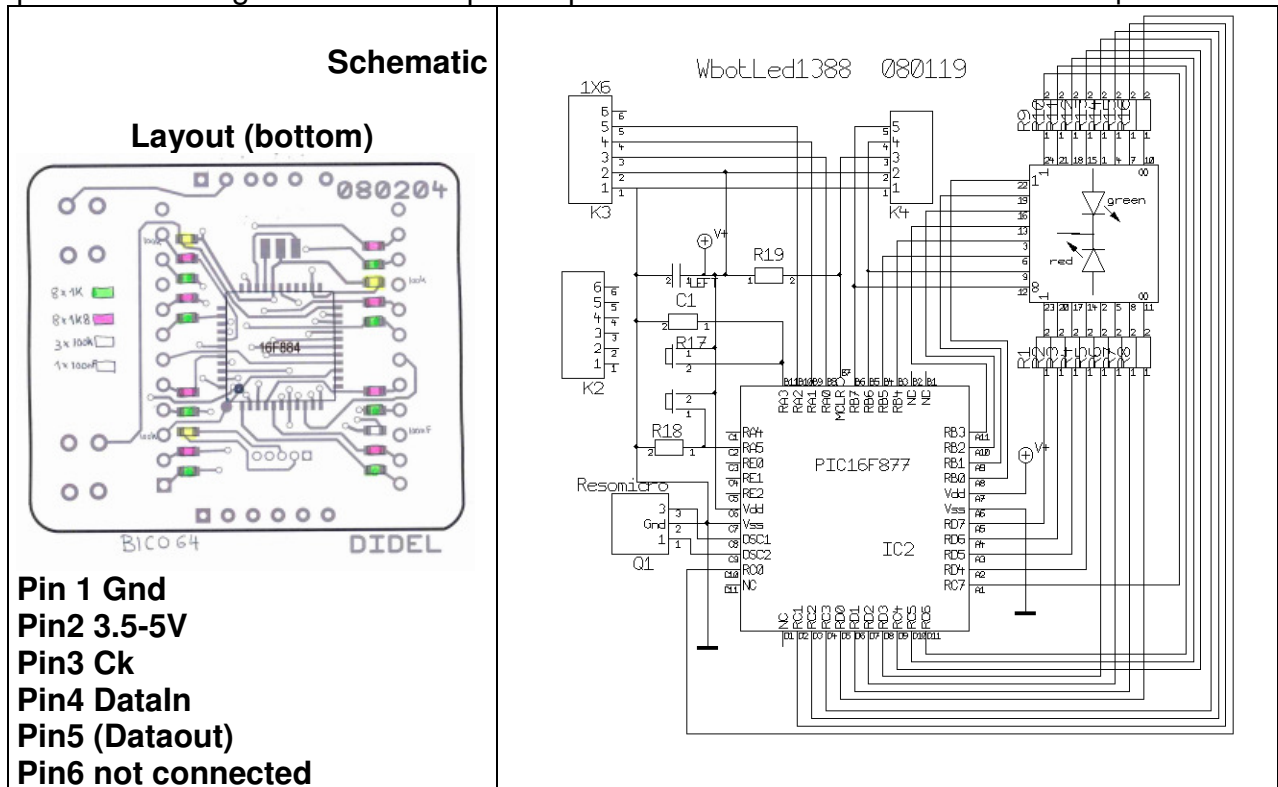
L'image peut être calculée en permanence, pour avoir un affichage dynamique. par exemple pour un jeu de Pong.



En général, l'image est préparée en mémoire et une routine transfère en continu ce bloc mémoire sur l'affichage.  
 Le format utilisé par les routines de Didel est donné ci-contre, avec un exemple de contenu (le 0 ne sont pas représentés).  
 Les Leds rouge sont aux adresses impaire, les verts aux adresses impaires.



Le but ici est de programmer des motifs et animation. Connecté à un robot, un protocole permet de charger les motifs depuis le processeur maître. Voir [www.didel.com/pic/wellbot](http://www.didel.com/pic/wellbot)



### Exemples de programmes

Les programmes se trouvent sous [www.didel.com/wellbot/BicoSoft.zip](http://www.didel.com/wellbot/BicoSoft.zip)

Ils s'adressent à des personnes qui connaissent bien la programmation en CALM, voir [www.didel.com/pic/Programmer.pdf](http://www.didel.com/pic/Programmer.pdf)

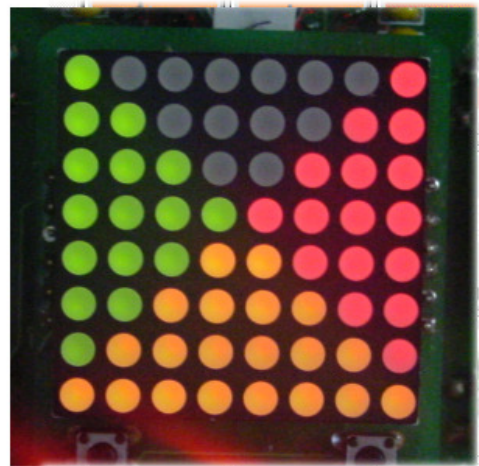
- WMA01.asm** Allumages simples – faites des variantes
- WMA02.asm** Diagonales
- WMA03.asm** Copie d'un motif – testez vos propres motifs
- WMA04.asm** Place pour 16 bitmaps
- WMA05.asm** 1-16 actions sur le poussoir droite pour sélectionner



```

00FF s: ; A copier en ram pour le jouer
00FF 8207 gne: Add W,PCL
0100 1:
0100 8034 RetMove #2 10000000,W
0101 0134 RetMove #2 00000001,W
0102 C034 RetMove #2 11000000,W
0103 0FC 3FFF 3FFF 3FFF 0782 2. 0011,W
0104 100 3480 3401 34C0 3403 4. 0000,W
0105 104 34E0 3407 34F0 340F 4. 0111,W
0106 108 34F8 341F 34FC 343F 4. 0000,W
0107 10C 34FF 340E 34FF 34FF 4. 0000,W
0107 0F34 RetMove #2 00001111,W
0108 1034 RetMove #2 11110000,W
0109 1074 RetMove #2 11111000,W
010A EC34 RetMove #2 11111100,W
010B 3F34 RetMove #2 00111111,W
010C FE34 RetMove #2 11111110,W
010D 7F34 RetMove #2 01111111,W
010E FF34 RetMove #2 11111111,W
010F FF34 RetMove #2 11111111,W
0110
9/19 13:28:59

```

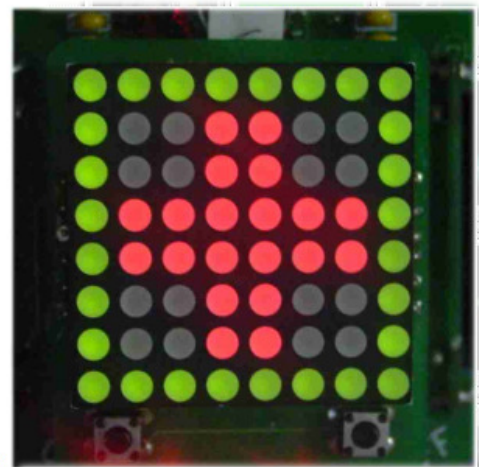


On peut modifier ou créer les motifs directement dans la mémoire du Pickit2, après lecture du binaire WMA05 sur le 664

```

0110 2:
0110 FF34 RetMove #2 11111111,W
0111 0034 RetMove #2 00000000,W
0112 8134 RetMove #2 10000001,W
0113 1834 RetMove #2 00011000,W
0114 8134 RetMove #2 10000001,W
0115 1834 RetMove #2 00011000,W
0116 8134 RetMove #2 10000001,W
0117 7E34 RetMove #2 01111110,W
0118 8134 RetMove #2 10000001,W
0119 7E34 RetMove #2 01111110,W
011A 8134 RetMove #2 10000001,W
011B 1834 RetMove #2 00011000,W
011C 8134 RetMove #2 10000001,W
011D 1834 RetMove #2 00011000,W
011E FF34 RetMove #2 11111111,W
011F 0034 RetMove #2 00000000,W
0120 ; ajouter les motifs 2 .. 14 ici
0120

```



Les routines de communication en série avec le processeur d'un robot (le Wellbot a un connecteur compatible) existent. Le protocole, documenté sous [www.didel.com/wellbot/WeBico.pdf](http://www.didel.com/wellbot/WeBico.pdf) permet de transférer un motif complet ou d'agir sur des bits.

Le Dauphin a un écran graphique. Regardez comment ont été définies les routines et programmez des routines équivalentes <http://www.epsitec.ch/dauphin/>