

18

Encore une interface LCD à un seul fil

un microcontrôleur est nécessaire pour le faire

Rob van Hest (Pays-bas)

Il existe de nombreuses interfaces permettant d'envoyer des données à un écran LCD alphanumérique standard via un fil, grâce à un registre à décalage.

L'interface présentée ici utilise un microcontrôleur qui reçoit les données de l'hôte au moyen d'une UART.

J'avais besoin d'un écran LCD pour l'un de mes projets, mais sur le système hôte, une seule broche était disponible pour le piloter. Pas de souci : il existe de nombreux exemples proposés notamment par *Elektor* pour construire une telle interface. La plupart d'entre eux utilisent un registre à décalage, mais je n'en avais pas, dans mon stock privé de composants. Il a donc été nécessaire de réaliser la fonction de décalage par logiciel à l'aide d'un petit microcontrôleur. Mais réfléchissons. Pourquoi implémenter une nouvelle interface des deux côtés alors que des interfaces standards existent déjà ? J'ai donc décidé d'utiliser une UART pour la connexion. Les couches inférieures de l'interface sont déjà mises en œuvre dans le microcontrôleur, ce qui rend le reste du projet très simple.

Matériel

La **figure 1** montre le schéma de cette interface LCD série. J'ai utilisé un microcontrôleur PIC16F1825 à 14 broches de Microchip Technology, mais le PIC16F1823, plus abordable, fera également l'affaire. La plupart des broches sont utilisées pour contrôler l'afficheur, mais une broche reste disponible et peut servir pour une fonction supplémentaire (ou même deux si la programmation basse tension est désactivée). Deux entrées série sont disponibles : une interface compatible RS-232, qui peut par exemple être connectée à un PC (je l'ai utilisée pour tester le micrologiciel), et une interface UART-TTL, qui peut être utilisée pour connecter le circuit au microcontrôleur du système hôte. Dans ce dernier cas, il est possible d'omettre le T2 et les composants environnants. Le débit n'est que de 9600 bauds ; il est donc possible d'utiliser

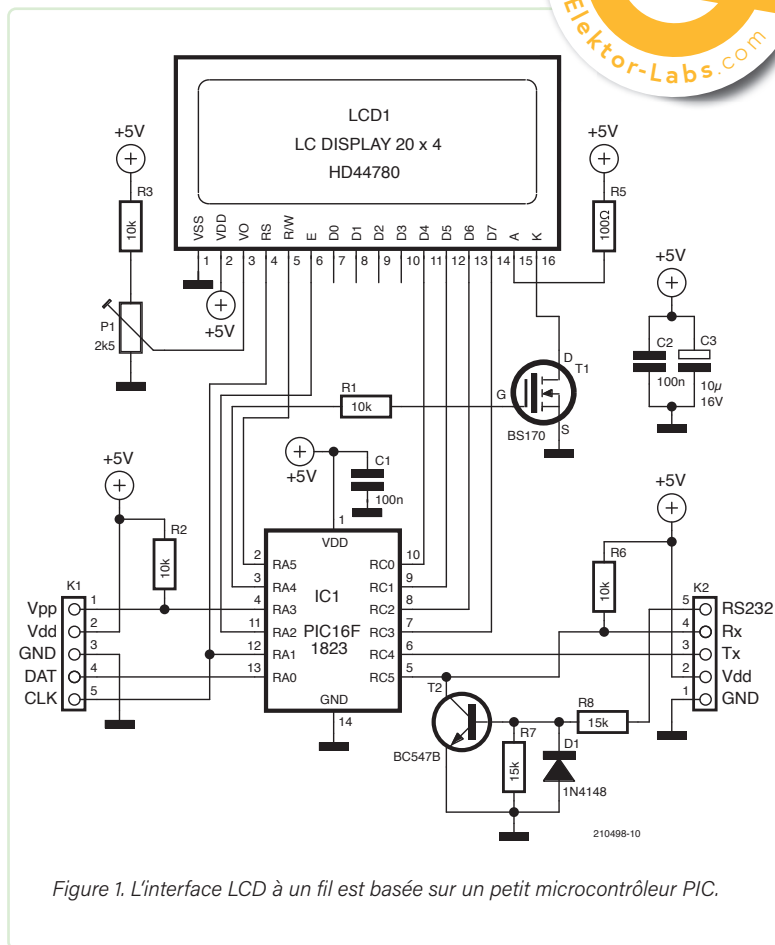


Figure 1. L'interface LCD à un fil est basée sur un petit microcontrôleur PIC.

liser le circuit simple avec T2 pour convertir un signal RS-232 au niveau logique de l'entrée RC5 du microcontrôleur. On choisit également un débit en bauds faible, le système hôte n'ayant ainsi pas besoin d'inclure des cycles d'attente après certaines commandes.

T1 est inclus pour commuter le rétroéclairage de l'écran LCD, et le trimmer P1 règle le contraste de l'écran. K1 fournit la connexion pour une interface de programmation Microchip Pickit, mais ce connecteur peut être omis dans la réalisation finale. Bien que j'aie conçu un circuit imprimé, la version finale a été réalisée sur Veroboard (**figure 2**), car je n'avais besoin que d'une seule pièce. Le circuit imprimé n'a donc jamais été produit et testé. Néanmoins, les fichiers Eagle peuvent être téléchargés sur la page Elektor Labs de ce projet [1] ; vous pouvez les utiliser à vos propres risques.

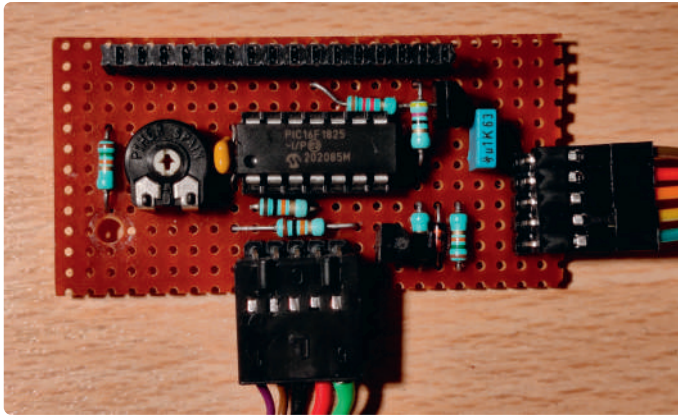


Figure 2. Circuit construit sur une plaque Veroboard.

Logiciel

Le programme est compilé avec le compilateur CC5X de B. Knudsen [2]. Le code source de cette interface et le fichier HEX sont disponibles pour téléchargement sur la page Elektor Labs de ce projet [1].

Pour comprendre les étapes suivantes, vous pouvez effectuer une recherche en ligne sur le contrôleur LCD HD44780 [3]. S'il s'agit d'envoyer un octet à l'écran, nous devons placer deux octets successivement sur les quatre fils D4... D7.

La première idée était de réaliser l'écran sous la forme d'un petit terminal VT52 ou VT100, mais cela demandait beaucoup trop d'opérations pour un appareil aussi simple. La version finale n'est donc trop sophistiquée.

Il y a trois modes implémentés dans mon micrologiciel pour envoyer des données à l'écran : 4 bits, 8 bits de données et 8 bits de contrôle. En mode 4 bits, le programme place simplement les 4 bits de poids faible (0... 3) de chaque caractère reçu sur les quatre fils du LCD, en utilisant le bit 4 pour contrôler l'entrée RS (0 = contrôle, 1 = données). Il est possible de contrôler entièrement l'affichage avec les caractères ASCII 0x20 à 0x3f (« » à « ? », voir les exemples). Pour envoyer un octet de données à l'écran, nous avons besoin de deux quartets, donc de deux caractères transmis au microprocesseur via l'UART. Le microprocesseur attend la réception du deuxième caractère avant de placer le premier quartet sur les fils. Pour éviter que l'interface ne « se bloque », il y a un délai d'attente avant la réception de ce deuxième caractère : l'interface est réinitialisée si elle ne le reçoit pas au bout d'une seconde.

Dans les modes 8 bits, le microcontrôleur divise les données entrantes en deux quartets pour le LCD. Comme aucun bit supplémentaire n'est disponible pour l'entrée RS, des modes 8 bits de contrôle et 8 bits de données distincts sont nécessaires. On peut simplement envoyer une chaîne de caractères en mettant le LCD en mode données sur 8 bits et en envoyant la chaîne sous forme de caractères ASCII. L'utilisation du contrôle sur 8 bits est moins utile, car souvent, il n'est pas plus efficace que le contrôle sur 4 bits. Les codes de contrôle suivants sont disponibles (n'utilisez pas ces codes pour des caractères en mode données sur 8 bits !)

- > CTRL-A (0x01) – rétroéclairage éteint
- > CTRL-B (0x02) – rétroéclairage allumé
- > CTRL-C (0x03) – envoi de 8 bits de contrôle
- > CTRL-D (0x04) – envoi de 8 bits de données (ASCII)
- > CTRL-E (0x05) – effacement de l'écran
- > Échappement (0x1b) – envoi de 4 bits de contrôle et de données

En l'absence de fonctions intelligentes, il est possible d'utiliser tous les types d'afficheurs basés sur le contrôleur LCD HD44780 et autres versions compatibles. Il appartient au système hôte de gérer les informations relatives au nombre de lignes et de caractères. ◀

210498-04

Des questions, des commentaires ?

Envoyez un courriel à l'auteur (trainer99@ziggo.nl) ou contactez Elektor (redaction@elektor.fr).

Exemples de contrôle de l'écran LCD

effacer l'écran, allumer le rétroéclairage et imprimer « test 1 » sur la première ligne en utilisant l'interface 8 bits :

```
0x05,0x02,0x04,'test 1' (ctrl-e,ctrl-b,ctrl-d,'test 1')
```

même séquence en utilisant une interface à 4 bits :

```
0x1b,0x20,0x21,0x02,0x37,0x34,0x36,0x35,0x37,0x33,0x37,0x34,0x32,0x30,0x33,0x31 (Esc,'!',ctrl-b,'746573742031')
```

imprimer « line 2 » sur la deuxième ligne de l'écran en utilisant une interface à 8 bits :

```
0x03,0xc0,0x04,'line 2'
```

même chose en utilisant une interface à 4 bits :

```
0x1b,0x2c,0x20,0x36,0x3c,0x36,0x39,0x36,0x3e,0x36,0x35,0x32,0x30,0x33,0x32 (Esc,', 6<696.652032')
```

↓ Télécharger le projet

www.elektormagazine.fr/summer-circuits-22



Produits

- > IIC/I²C Serial Interface Adapter Module (SKU 19332)
www.elektor.fr/19332
- > 2x16 Character LCD blue/white (SKU 17757)
www.elektor.fr/17757

LIENS

[1] B Knudsen Data, compilateur CC5X : <https://www.bknd.com/cc5x/>

[2] Ce projet sur Elektor Labs : <https://www.elektormagazine.fr/labs/yet-another-1-wire-lcd>

[3] « Contrôleur LCD HD44780 de Hitachi », Wikipedia : <https://fr.wikipedia.org/wiki/HD44780>