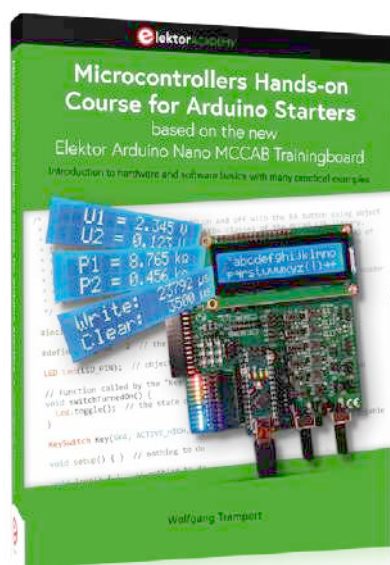
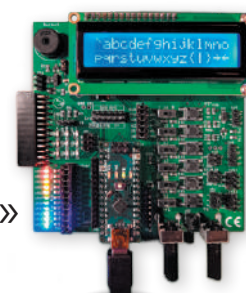




la carte d'apprentissage MCCAB pour Arduino Nano

plateforme pour le cours
« Microcontrollers Hands-On Course »



Wolfgang Trampert (Allemagne) et Jan Buiting (Elektor)

L'assemblage de petits circuits d'extension pour Arduino consiste à placer des composants sur une plaque d'essai et de les connecter avec quelques fils de connexion colorés. Cependant, toutes ces connexions prennent souvent plus de temps que l'écriture du programme. De plus, la disposition des composants, par exemple, pour un feu de circulation à 11 LED et autant de résistances en série est sujette à des erreurs et à des heures précieuses consacrées au débogage du matériel ! Dans de tels cas, une carte d'apprentissage dédiée, comme la nouvelle « MCCAB » d'Elektor, est plus pratique, surtout qu'elle est livrée avec un excellent manuel utilisateur.

Note de la rédaction. cet article est un extrait du livre *Microcontrollers Hands-on Course for Arduino Starters* (Elektor, 2023), formaté et légèrement modifié pour correspondre aux normes éditoriales et à la mise en page du magazine Elektor. L'auteur et l'éditeur ont fait de leur mieux pour l'éviter et seront heureux de répondre aux questions – pour les contacter, voir l'encadré « Des questions, des commentaires ? »

La carte d'apprentissage pour Arduino Nano « MCCAB » (*Microcontroller Crash Course for Arduino Beginners*) a été spécialement conçue par l'auteur et fabriquée pour accompagner le guide d'Elektor *Microcontrollers Hands-on Course for Arduino Starters*. La carte MCCAB, le guide (en anglais ou en allemand) et une carte Arduino Nano sont disponibles en offre groupée dans l'e-choppe Elektor [1].

Sur la carte d'apprentissage – appelée MCCAB dans la suite de cet article – de nombreux composants sont déjà connectés au module de microcontrôleur branché ou peuvent être connectés avec de simples fils de connexion. Ces composants incluent des interrupteurs/boutons, des LED, des potentiomètres, des buzzers, un écran LCD, des interfaces et des convertisseurs de niveau pour les connexions séries, ainsi que des étages pilotes pour les appareils externes.

Il est possible de connecter des modules externes au MCCAB via un connecteur femelle, ou vous pouvez les relier au microcontrôleur de la carte via les interfaces série. Cela permet d'éviter l'assemblage fastidieux de circuits expérimentaux et de se concentrer sur l'essentiel, c'est-à-dire sur le logiciel, ou « le programme ». Nous chargerons donc les programmes créés dans nos exercices dans le microcontrôleur ATmega328P du MCCAB, où ils seront exécutés (comme code de programme).

Certaines des 32 broches du microcontrôleur ATmega328P sont utilisées pour son alimentation – un quartz est connecté à l'oscillateur intégré pour la génération du signal d'horloge – ou comme entrée

RESET pour le bouton-poussoir du module de microcontrôleur connecté au MCCAB (l'appui sur ce bouton réinitialise le microcontrôleur à l'état initial défini et redémarre le programme). Cependant, la majorité des broches sont des entrées/sorties à usage général (GPIO), qui peuvent être utilisées pour connecter le microcontrôleur au monde extérieur. Sur le MCCAB, ces broches – sauf quelques exceptions réservées à des fins internes – sont accessibles grâce à des barrettes de connecteurs.

La **figure 1** montre une vue du MCCAB avec ses blocs fonctionnels codés en couleur. Le mode d'emploi du MCCAB, avec une description détaillée de tous les composants, est téléchargeable gratuitement sur le site web d'Elektor [2].

Blocs fonctionnels du MCCAB

Sur la figure 1, on peut distinguer les blocs fonctionnels et les modules suivants sur le MCCAB.

(1) **Module microcontrôleur Arduino Nano** avec bouton RESET (flèche 1a), LED (flèche 1b) et prise mini USB pour la connexion au PC de l'utilisateur.

(2) Connecteurs SV5 et SV6 pour **les entrées/sorties du microcontrôleur**. En utilisant des câbles Dupont, il est possible de connecter les broches d'E/S (GPIO) du microcontrôleur aux composants internes de la carte d'apprentissage ou à des modules externes via ces deux connecteurs.

(3) **11 LED**, LD10-LD20 (indicateurs d'état pour les entrées/sorties D2-D12 du microcontrôleur). On peut connecter chaque LED aux broches affectées, D2-D12, via un cavalier sur le connecteur JP6.

(4) **Matrice LED 3×3** LD1-LD9 (9 LED rouges). Les colonnes sont connectées en permanence aux sorties D6, D7 et D8 du microcontrôleur, et les lignes peuvent être connectées aux broches D3, D4 et D5 avec des cavaliers sur le connecteur JP1.



Si les lignes de la matrice LED 3×3 sont connectées aux broches D3, D4 et D5 via les cavaliers sur le connecteur JP1 ou à d'autres broches du microcontrôleur avec des câbles Dupont, ces lignes, ainsi que les colonnes D6, D7 et D8, ne doivent jamais être utilisées pour d'autres tâches dans un croquis.

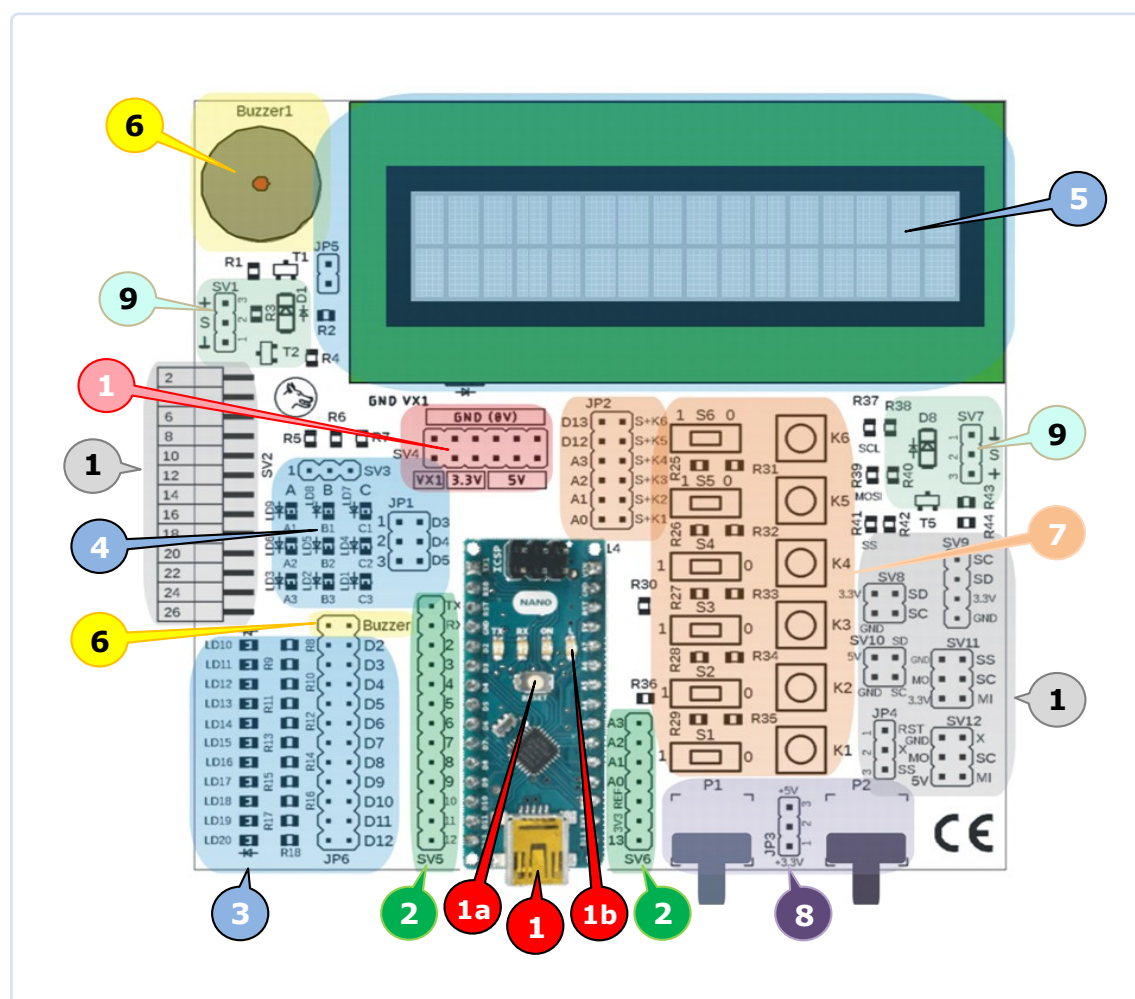
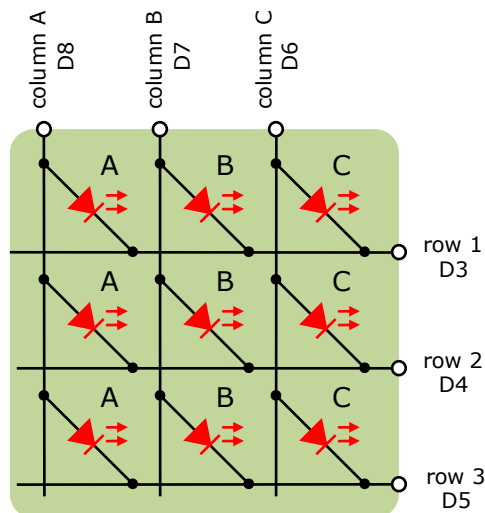


Figure 1. Schéma fonctionnel de la carte d'apprentissage pour Arduino Nano « MCCAB » d'Elektor.

Figure 2. Matrice LED 3x3.



Une double affectation des broches de la matrice risque d'entraîner des dysfonctionnements, voire d'endommager le MCCAB !

(5) **Écran LCD** de 2x16 caractères, connecté via le bus I²C aux broches A4 et A5 du microcontrôleur. Il est possible de couper la tension qui alimente l'écran LCD en tirant le cavalier JP5, pour les essais ou les tests où l'écran n'est pas utilisé.

(6) Il est possible de connecter **le buzzer piézoélectrique** Buzzer1 à la broche D9 avec un cavalier sur la position « Buzzer » du connecteur JP6.

(7) **6 interrupteurs à glissière, S1-S6**, connectés en parallèle à 6 boutons, **K1-K6**. Il est possible de les connecter aux entrées A0-A3 et D12-D13 du microcontrôleur via des cavaliers sur le connecteur JP2.

(8) **Potentiomètres P1 et P2**, dont les curseurs sont connectés aux broches d'entrée analogique du microcontrôleur, A6 et A7. L'alimentation 3,3 V ou 5 V peut être appliquée aux potentiomètres via le connecteur JP3.



Attention : les broches A6 et A7 de l'ATmega328P sont définies comme des entrées analogiques par l'architecture interne de la puce. Les configurer avec la fonction `pinMode()` n'est pas autorisé et peut conduire à un comportement erroné du programme.

(9) Les broches SV1 et SV7 sont **des sorties de commutation pour les appareils externes.**

(10) Connecteurs pour la liaison série des modules **SPI et I²C** externes.

(11) Barrette de connexion SV2 avec 2x13 broches pour **la connexion de modules externes.**

(12) Le connecteur SV4 est le **distributeur des tensions de fonctionnement de la carte.** Ces tensions peuvent alimenter des composants internes de la carte d'apprentissage ou des modules externes avec des câbles Dupont.

Les deux unités fonctionnelles un peu plus complexes, la matrice 3x3 LED et l'écran LCD (voir **figure 1**) seront

décrites plus en détail dans la suite. Pour les parties restantes, veuillez vous référer à la description détaillée dans le document *MCCAB Operating Instructions* [2].



La bibliothèque *MCCAB_Lib* est disponible pour piloter la matrice LED 3x3, les LED LD10-LD20, les boutons K1-K6 et les interrupteurs S1-S6, ainsi que Buzzer1. La bibliothèque est disponible gratuitement par les propriétaires de MCCAB et intégrée dans l'EDI Arduino.

Pour contrôler l'écran LCD, nous utilisons la bibliothèque *LiquidCrystal_I2C*, que vous pouvez télécharger gratuitement et ajouter à l'EDI Arduino.

Matrice LED 3x3

Le MCCAB contient neuf diodes électroluminescentes placées dans une matrice (voir **figure 1**). La **figure 2** montre leur circuit de base.

La matrice se compose de trois colonnes et de trois lignes. Les colonnes sont nommées A, B et C, tandis que les lignes sont numérotées 1, 2 et 3. À chacune des neuf intersections colonne/ligne, une LED est connectée – son anode à la colonne et sa cathode à la ligne. Les neuf LED sont étiquetées en fonction de leurs positions colonne/ligne respectives, par exemple, « B2 » est connectée à la colonne B et à la ligne 2. Pour qu'une LED s'allume, sa colonne doit être au niveau logique 1 (+5 V) et sa ligne au niveau logique 0 (0 V). Le manuel d'utilisation du MCCAB indique que les colonnes sont connectées en permanence aux broches D6-D8 du microcontrôleur. Vous pouvez connecter les lignes aux broches D3-D5 avec des cavaliers sur le connecteur JP1.

Si les lignes de la matrice LED 3x3 LED sont connectées aux broches D3, D4 et D5 via les cavaliers sur le connecteur JP1 ou aux autres broches du microcontrôleur avec des câbles Dupont, les lignes et les colonnes D6-D8 ne doivent pas être utilisées pour d'autres tâches dans votre croquis. Une telle connexion des broches de la matrice entraînerait des dysfonctionnements ou, dans le pire des cas, endommagerait le MCCAB !

Si la matrice n'est pas utilisée dans un croquis, vous devez enlever les cavaliers sur le connecteur JP1 du MCCAB.

Avantages de l'arrangement matriciel

Les matrices LED sont souvent utilisées, par exemple dans les stades, avec des LED colorées de forte puissance pour générer des images animées.

Si nous devons piloter 9 LED individuellement, nous aurions besoin de 9 broches du microcontrôleur. En organisant les LED dans une matrice, nous aurons besoin de six broches seulement. Plus il y a de LED à piloter, plus l'avantage de l'utilisation d'une matrice est grand : avec une matrice composée de huit colonnes et



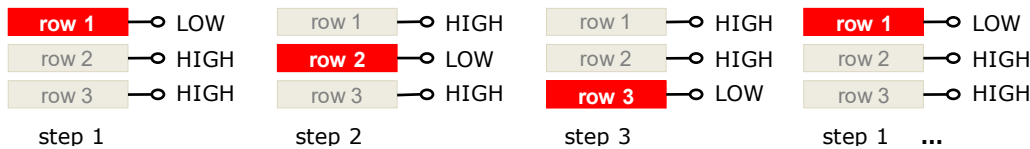


Figure 3. Activation progressive et cyclique des trois lignes qui composent la matrice.

huit lignes, nous pouvons piloter 64 LED, alors qu'avec des lignes de contrôle individuelles, 48 broches supplémentaires seraient nécessaires !

Pilotage de la matrice en mode multiplex

La **figure 2** montre que chaque colonne et chaque ligne de la matrice sur le MCCAB est connectée à trois LED. Par conséquent, le contrôle simultané de toutes les lignes et colonnes ne fonctionnerait pas, car les LED qui devraient en fait être éteintes seraient allumées involontairement. Au lieu de cela, comme le montre la **figure 3**, une seule ligne peut être activée à la fois et les colonnes doivent appliquer la séquence de bits de la ligne activée. Les deux autres lignes doivent être ouvertes pendant ce temps ou désactivées avec un niveau logique haut afin qu'aucun courant ne puisse les traverser.

Si les trois lignes sont rapidement activées les unes après les autres comme dans la **figure 3**, et à cause de la persistance rétinienne, on voit une image statique des neuf LED. Le programme utilisateur contrôle la matrice avec une boucle infinie, dans laquelle l'une des trois lignes, 1, 2 ou 3, est mise à un niveau logique bas cycliquement, tandis que les deux autres lignes sont mises à un niveau haut. Les connexions de colonne de toutes les LED à allumer dans la ligne active sont mises au niveau haut. Les connexions de colonne des LED qui doivent être éteintes dans la ligne active sont mises à un niveau logique bas.

Par exemple, pour allumer deux LED, A3 et C3, la ligne 3 doit être au niveau bas et les colonnes A et C au niveau haut, tandis que les deux lignes de la rangée, 1 et 2, sont au niveau haut et la colonne B au niveau bas.

LCD

Le MCCAB d'Elektor est équipé d'un afficheur LCD qui permet d'afficher du texte, des valeurs numériques

et même des caractères spéciaux définis par l'utilisateur. L'afficheur utilisé comporte deux lignes de 16 colonnes. Il est possible d'afficher un caractère dans chaque colonne. Chaque caractère est formé des points d'une matrice 5x8, comme le montre la **figure 4**. Les séquences de bits de la matrice 5x8 points nécessaires pour chaque caractère individuel sont stockées à l'intérieur du contrôleur LCD selon la table ASCII.

L'ASCII est un code de 7 bits, alors que notre microcontrôleur ATmega328P est une « unité de 8 bits », ce qui signifie qu'il peut traiter un octet à la fois et stocker ses données sous la forme d'un octet. Par conséquent, le huitième bit est généralement mis à 0 lors du stockage des codes ASCII. Cependant, les développeurs du contrôleur d'affichage HD44780 (qui est standard dans cette catégorie et que l'on trouve sur presque tous les modules LCD) n'ont pas voulu laisser ce huitième bit inutilisé, et ont étendu le jeu de caractères ASCII de 128 codes de caractères supplémentaires (où le huitième bit est à la valeur logique 1). Pour cette raison, notre LCD utilise également les codes de caractères 128 à 255 (de manière plus ou moins contiguë). Outre les 95 caractères imprimables correspondant aux codes 32-126, l'ASCII compte également 33 caractères de contrôle, à savoir 0-31 et 127, qui sont « non imprimables ». Comme ces caractères sont destinés à des fins de contrôle, l'écran ne les affiche pas. Le contrôleur d'affichage HD44780 est disponible en deux versions : soit avec le code ROM A00, soit avec le code ROM A02. La version A00 affiche des caractères vides pour les codes 106-31 ainsi que 128-159. La version A02 affiche des caractères spéciaux pour ces espaces d'adresses (voir **tableau 1**). Il est impossible de prévoir laquelle des deux versions de ROM est installée dans l'écran LCD du MCCAB, car c'est le fabricant du module LCD qui fait ce choix.

L'utilisateur peut définir jusqu'à huit caractères

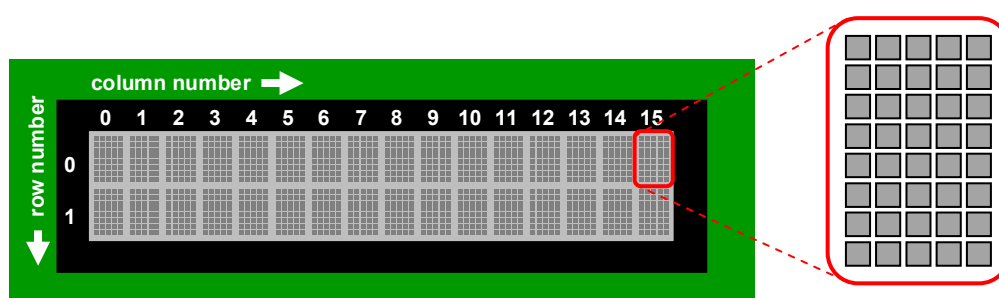


Figure 4. Illustration de l'afficheur LCD du MCCAB avec ses 16x2 caractères affichables.

Table 1. Allocation de mémoire de caractères pour les deux variantes de HD44780

Code de caractère	Code ROM A00	Code ROM A02
0-7	caractères définis par l'utilisateur	caractères définis par l'utilisateur
8-15	caractères définis par l'utilisateur	caractères définis par l'utilisateur
16-31	(sans affichage)	caractères visibles
32-127	caractères visibles	caractères visibles
128-159	(sans affichage)	caractères visibles
160-255	caractères visibles	caractères visibles

spéciaux et les afficher sur l'écran LCD. Les codes ASCII 0 à 7 sont attribués à ces huit caractères spéciaux. En option, il est possible d'adresser ces mêmes huit caractères spéciaux par l'intermédiaire des codes ASCII 8 à 15.

La **figure 4** montre également la numérotation des lignes et des colonnes de l'afficheur, qui, dans les deux cas, commence par 0. En spécifiant ces données, on peut écrire un caractère à une certaine position. À cette fin, l'écran LCD dispose d'un curseur qui détermine cette position. La bibliothèque LCD décrite dans le guide contient des méthodes pour positionner ce curseur.

Bien que l'afficheur ne puisse afficher que 16 caractères dans chaque ligne, la mémoire dans laquelle les caractères sont stockés dans le contrôleur d'affichage dispose en fait de 40 emplacements mémoire pour chaque ligne. Il y a un écart de 24 emplacements mémoire entre la dernière adresse d'affichage, 39, de la première ligne et l'adresse de début, 64, de la deuxième ligne (voir **figure 5**).

Grâce aux fonctions de décalage (*shift*), qui sont

incluses dans la bibliothèque LCD, la zone de mémoire visible à l'afficheur peut être déplacée sur l'ensemble de la zone de mémoire.

La figure 5 comporte trois parties :

- en haut : la zone visible de l'afficheur dans le réglage de base, sans opérations de décalage préalables.
- au milieu : le contenu de l'afficheur par défaut après décalage à gauche.
- en bas : le contenu de l'afficheur par défaut après un décalage à droite.

Réglage du contraste de l'écran LCD

Le MCCAB ne dispose pas d'un réglage de contraste pour l'afficheur LCD. Pour cette raison, et parce que le contraste de l'écran peut varier en fonction des conditions environnementales (telles que la température) ou du vieillissement, le LCD est doté d'un potentiomètre de réglage qui permet d'ajuster le contraste. Ce petit potentiomètre est accessible par le dessous de la carte et est marqué d'une flèche. Réglez le contraste avec un petit tournevis lors de la première utilisation ou lorsque vous en avez besoin.

Astuce : Si aucun caractère n'est affiché lors de la première utilisation de l'afficheur LCD, c'est probablement parce que le réglage du contraste n'a pas été effectué correctement !

Transmission de données du microcontrôleur à l'écran LCD

Chaque module LCD est équipé d'un contrôleur d'affichage HD44780, qui reçoit les codes (ASCII) des caractères envoyés par le microcontrôleur via une interface, génère les caractères correspondants de la matrice à

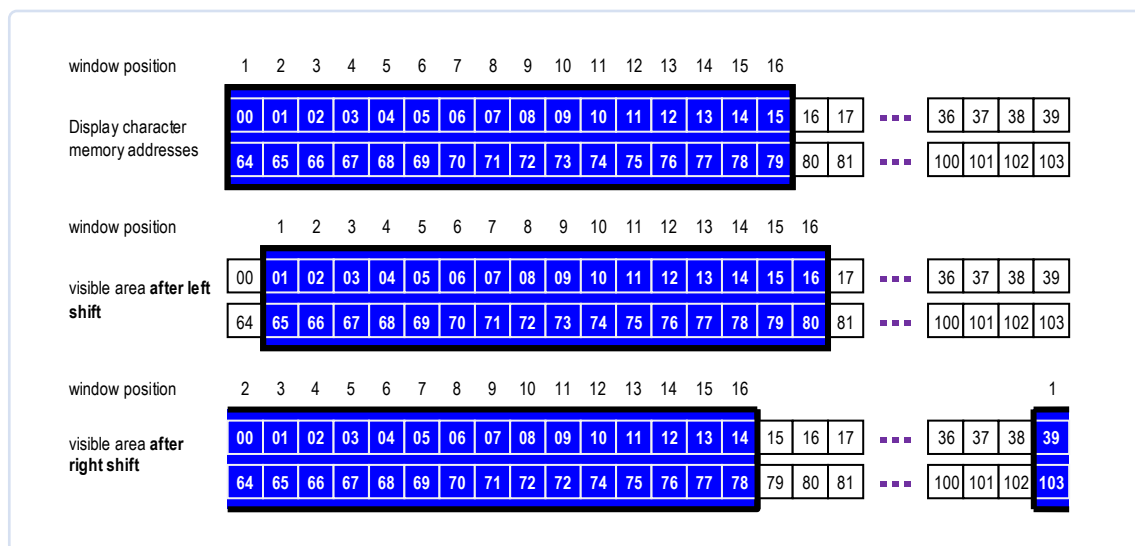


Figure 5. Contenu de l'afficheur avant et après le décalage.

points 5x8 mentionnée ci-dessus à partir des codes et les affiche.

Le contrôleur LCD HD44780 ne dispose que d'une interface parallèle pour recevoir les données à afficher, c'est-à-dire que le microcontrôleur écrit un octet composé de huit bits, D0-D7, dans le registre d'entrée du contrôleur HD44780, puis sur le LCD avec trois signaux de commande : RS, RW et E, comme le montre la **figure 6**. Cette méthode de transmission de données en parallèle limiterait considérablement les capacités du MCCAB car l'écran LCD occuperait déjà 11 des 16 broches disponibles du microcontrôleur ATmega328P !

La deuxième option pour la transmission des données au contrôleur HD44780, dans laquelle les huit bits de données sont transmis en deux paquets de quatre bits chacun, n'est pas réellement utile, car, même dans ce cas, sept broches du microcontrôleur seraient encore utilisées par le LCD, puisque les trois signaux de contrôle, RS, RW, E, sont également nécessaires dans ce cas.

Pour cette raison, le module LCD sur le MCCAB est contrôlé via le bus I²C, un bus série synchrone qui consiste en une seule ligne de données (SDA), une ligne d'horloge (SCL), et il transmet les données bit par bit. Le trafic de données via l'interface I²C est réalisé grâce à deux lignes du microcontrôleur ATmega328P, A4 (SDA) et A5 (SCL) (voir **figure 7**).

Un adaptateur supplémentaire situé sous le module LCD convertit les signaux I²C en signaux parallèles. Cet adaptateur facilite la méthode mentionnée ci-dessus. Il permet de transmettre deux ensembles de quatre bits à la suite via les lignes de données D4-D7, c'est-à-dire que les bits de données D4-D7, puis les bits de données D0-D3 sont transmis sur les mêmes lignes, D4-D7.

Étant donné que les lignes A4 et A5 du microcontrôleur sur le MCCAB sont de toute façon réservées à l'interface I²C, aucune ressource n'est perdue avec ce type de transmission à l'écran LCD car, en principe, plusieurs participant peuvent être connectés au bus I²C en même temps. Puisque chaque participant connecté au bus occupe sa propre adresse I²C, il ne « voit » qu'il est adressé que lorsqu'un paquet de données arrive avec l'adresse de ce participant. Le LCD du MCCAB occupe généralement l'adresse I²C 0x27. Si l'adresse diffère en raison du fabricant, cela est indiqué sur l'écran. ◀

230215-04

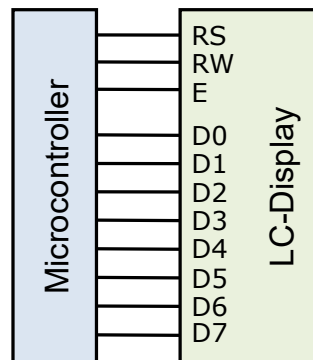


Figure 6. Pilotage de l'afficheur LCD avec huit bits de données et trois lignes de commande.

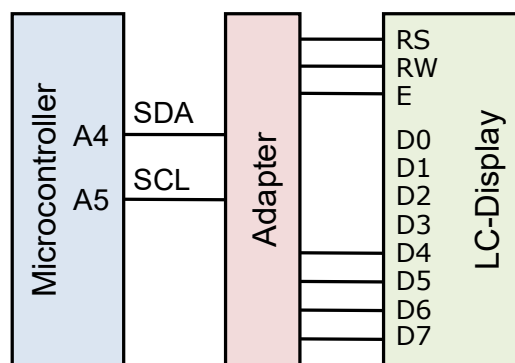


Figure 7. Pilotage de l'afficheur LCD via le bus I²C plutôt qu'en parallèle (voir figure 6).

Des questions, des commentaires?

Contactez Elektor (redaction@elektor.fr).



Produits

> Microcontrollers Hands-on Course for Arduino Starters (offre groupée) (SKU 20440)

- 1 × Guide : *Microcontrollers Hands-On Course for Arduino Starters*
 - 1 × MCCAB Arduino Training Board
 - 1 × Arduino Nano
- <https://elektor.fr/20440>

LIENS

- [1] Elektor Arduino Nano Training Board MCCAB and Companion Guide: <https://elektor.fr/20440>
 [2] Manuel d'utilisation de la carte MCCAB d'Elektor : <https://elektor.fr/20440>