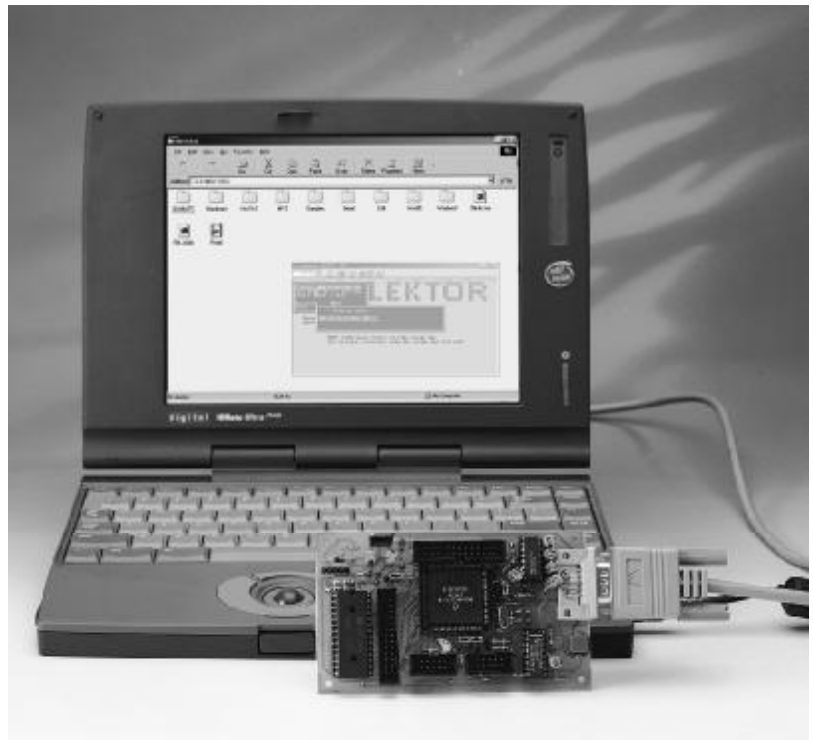


systeme de développement pour 68HC11F1

s'accommode de 8 boîtiers de mémoires différents

Le microcontrôleur 68HC11 de Motorola est sans le moindre doute l'un des plus appréciés dans l'univers du développement d'applications à base de microcontrôleurs. Voici l'une des solutions les moins chères, pour développer des applications à base de 68HC11. Cette approche se distingue en outre par une quantité de mémoire disponible relativement importante.



À quoi peut bien servir un système de développement ? Nombreux sont ceux d'entre nos lecteurs qui essaient de mettre au point leur propre montage à base de microcontrôleur et font appel, pour ce faire, à un outil de développement prenant la forme, selon le cas, de logiciel ou de matériel. Et c'est dans ce cas-là qu'entre en jeu un système de développement. Comme il apparaît que depuis quelque temps, c'est plutôt la version F1 de la famille 68HC11 qui semble être à la mode dans les revues d'électronique, nous vous proposons un système prévu pour ce composant en particulier. En effet, malgré ses 68 broches, le HC11F1 est (en France

du moins) pratiquement le moins cher de la gamme, et on le trouve assez facilement. Ce HC11, plutôt conçu pour les modes étendus, intègre un décodage d'adresse et un bus complet (non multiplexé). On peut donc lui adjoindre une mémoire RAM ou EPROM sans aucune circuiterie supplémentaire. Par ailleurs, le fonctionnement "single chip" reste possible si l'on peut se limiter à 1 Koctet de RAM et 512 octets d'EEPROM. Le HC11F1 accepte un quartz d'une fréquence allant jusqu'à de l'ordre de 16 à 20 MHz. À 16 MHz, il réalise un accès mémoire en 250 ns. On peut, bien sûr, travailler plus lentement si on le sou-

Projet : Jean Gonzalez &
Albéric Amaranthe

haite et les accès périphériques peuvent être plus lents que les accès mémoires (voir "clock stretching" dans la documentation du HC11F1, réf. [1]).

L'ÉLECTRONIQUE

L'architecture de base de la carte est très simple, mais suffisante pour ceux qui se limitent à l'assembleur, et nous verrons qu'elle peut être étendue.

La carte s'articule autour de 5 circuits intégrés : le HC11F1, un MAX232, un HC139, un circuit de mémoire et un régulateur.

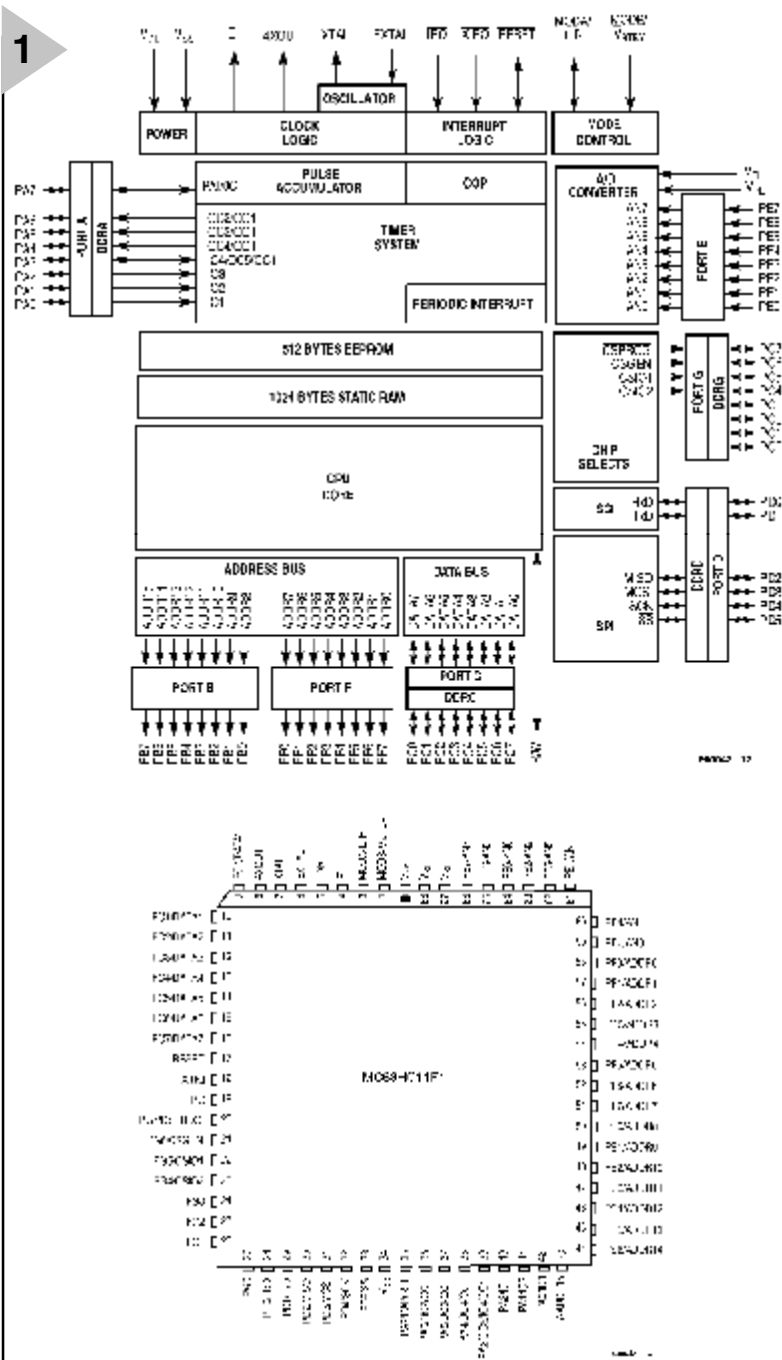
Le HC11F1 vient d'être présenté dans le paragraphe précédent, mais pourquoi ne pas en parler un peu pour donner envie à ceux qui ne le connaissent pas de l'essayer. Le 68HC11, dont on retrouve le brochage et la structure interne en **figure 1**, est sans doute le microcontrôleur de Motorola le plus vendu aux amateurs. La famille M68HC11 comporte pas moins, nous dit Motorola, de 17 membres. Un coup d'oeil à la structure interne vous montre qu'il s'agit d'un composant relativement complexe. Le 11F1 est l'un des exemplaires les plus récents (bien que datant d'il y a près de 10 ans) et des plus puissants de la famille, puisque non multiplexé et doté d'une RAM de 1 Koctet et de 512 octets d'EEPROM; il est proposé sous la forme d'un pavé à 68 broches. Il ne dispose ni de ROM ni d'EPROM interne, mais peut s'accommoder d'une variété étonnante et d'une capacité impressionnante de mémoire, ce qui ne fait qu'augmenter son intérêt. Il comporte pas moins de 4 ports, dont certains sont bi-directionnels.

Le second composant important du schéma représenté en **figure 2** est le MAX232; ce circuit intégré permet de connecter la carte au port série d'un PC; il est ainsi possible, via le logiciel M11 que nous vous présenterons plus loin, d'injecter du code et de le tester en disposant des points d'arrêts, des commandes de pas à pas, en visualisant le contenu des variables du programme, des registres...

Le troisième circuit, un décodeur d'adresses du type 74HC139, dont seule une moitié d'ailleurs est utilisée, fournit les signaux \overline{KL} et \overline{WR} qui donnent au bus la compatibilité avec des composants typés "INTEL". Ces signaux de lecture (*Read*) et d'écriture (*Write*) indépendants sont nécessaires pour les boîtiers EEPROM du type 2864.

Le support du boîtier mémoire, que nous avons, dans le schéma, doté, à titre d'exemple et pour nous simplifier l'existence, d'une RAM de 32 Koctets du type 62256, peut recevoir 8 boîtiers mémoires différents et ce n'est pas là la caractéristique la moins intéressante

Figure 1. Brochage et structure interne du 68HC11F1.
(Documents Motorola)



de ce montage ! En effet, le boîtier mémoire peut être n'importe quelle RAM, EPROM, EEPROM dès lors que ledit circuit possède 28 broches. Dans cette catégorie, on trouve donc les EPROM de 8, 16, 32, 64 Koctets, les RAM de 8 et 32 K ainsi que les EEPROM de 8 et 32 K. Il faudra disposer, en fonction du type de mémoire implanté dans le support destiné à IC2, jusqu'à 2 cavaliers sur le connecteur K7 (voir le **tableau 1** donné en annexe). Le logiciel M11, que ceux d'entre nos lecteurs qui ont réalisé le montage évoqué en référence [2] connaissent

bien, a été modifié pour permettre de programmer les EEPROM externes, ainsi ceux d'entre vous qui n'ont pas les moyens d'avoir un programmeur d'EPROM utiliseront des EEPROM, une 28C64 par exemple. Ceci ouvre un nouveau domaine d'utilisation de cette réalisation : on peut en effet même utiliser cette carte pour programmer des EEPROM pour d'autres applications. Une fois l'EEPROM externe programmée, on peut la protéger en écriture en enlevant le cavalier connecté sur \overline{WR} (connecteur K7). Il reste une autre EEPROM, intégrée dans le HC11, pour la sauvegarde des données de configuration.

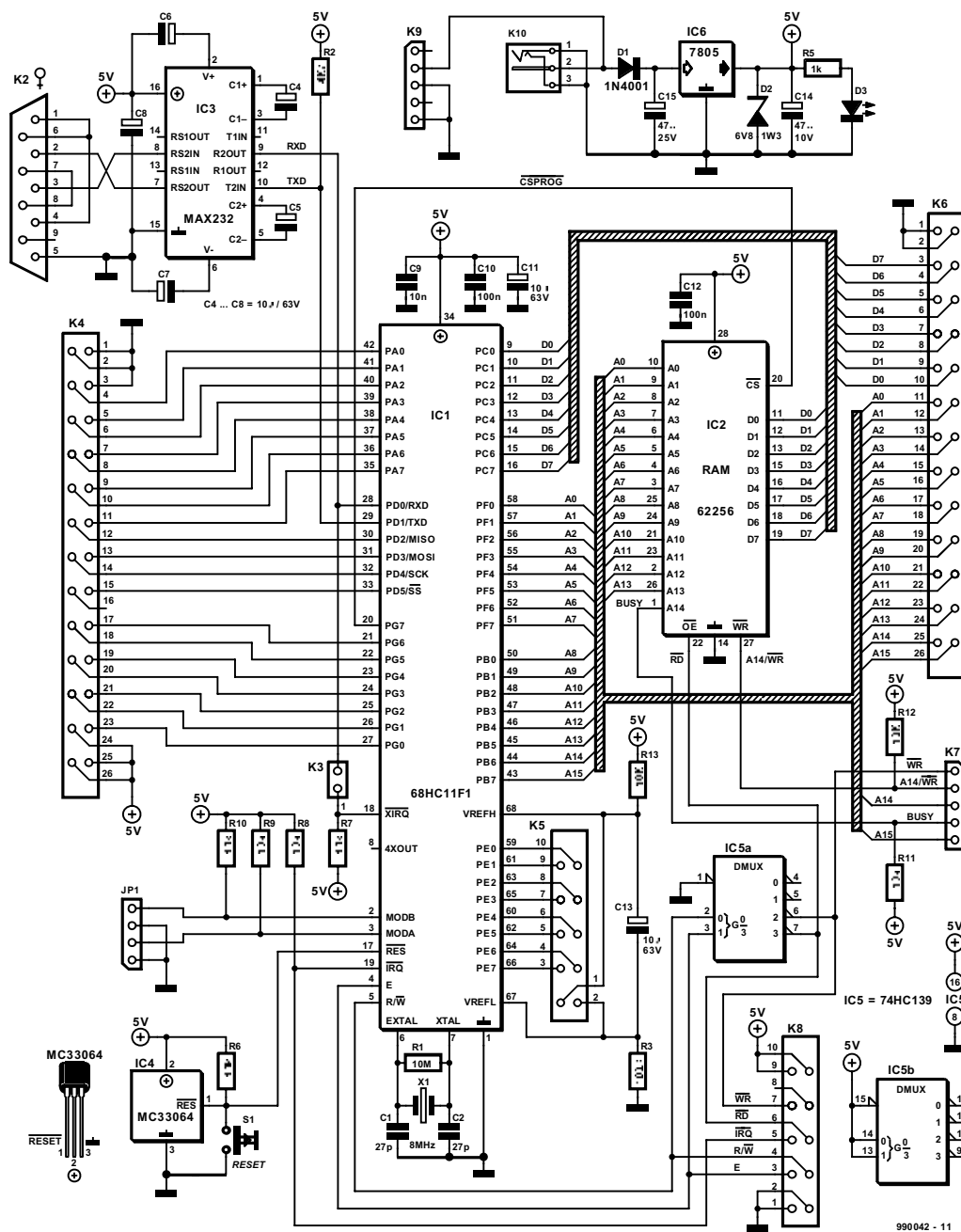


Figure 2. L'électronique du système de développement pour 68HC11F1 est écrasée par un microcontrôleur à 68 broches, le 11F1.

Remarque : Pour utiliser une EPROM externe, il faut posséder un programmeur d'EPROM, par contre M11 sait programmer les EPROM intégrées aux HC11 (versions autres que F1).

Vu que le HC11F1 possède une RAM interne de 1 K, sur une application terminée, IC2 sera une EPROM ou une EEPROM. IC2 sera en phase de mise au point, il faut pouvoir charger de multiples versions du programme, installer des points d'arrêts,.... Ceci impose pour IC2 une RAM ou une EEPROM. En mise au point, la RAM permet un chargement beaucoup plus rapide si le code dépasse quelques Koctets. Finalement, l'EEPROM est le type de

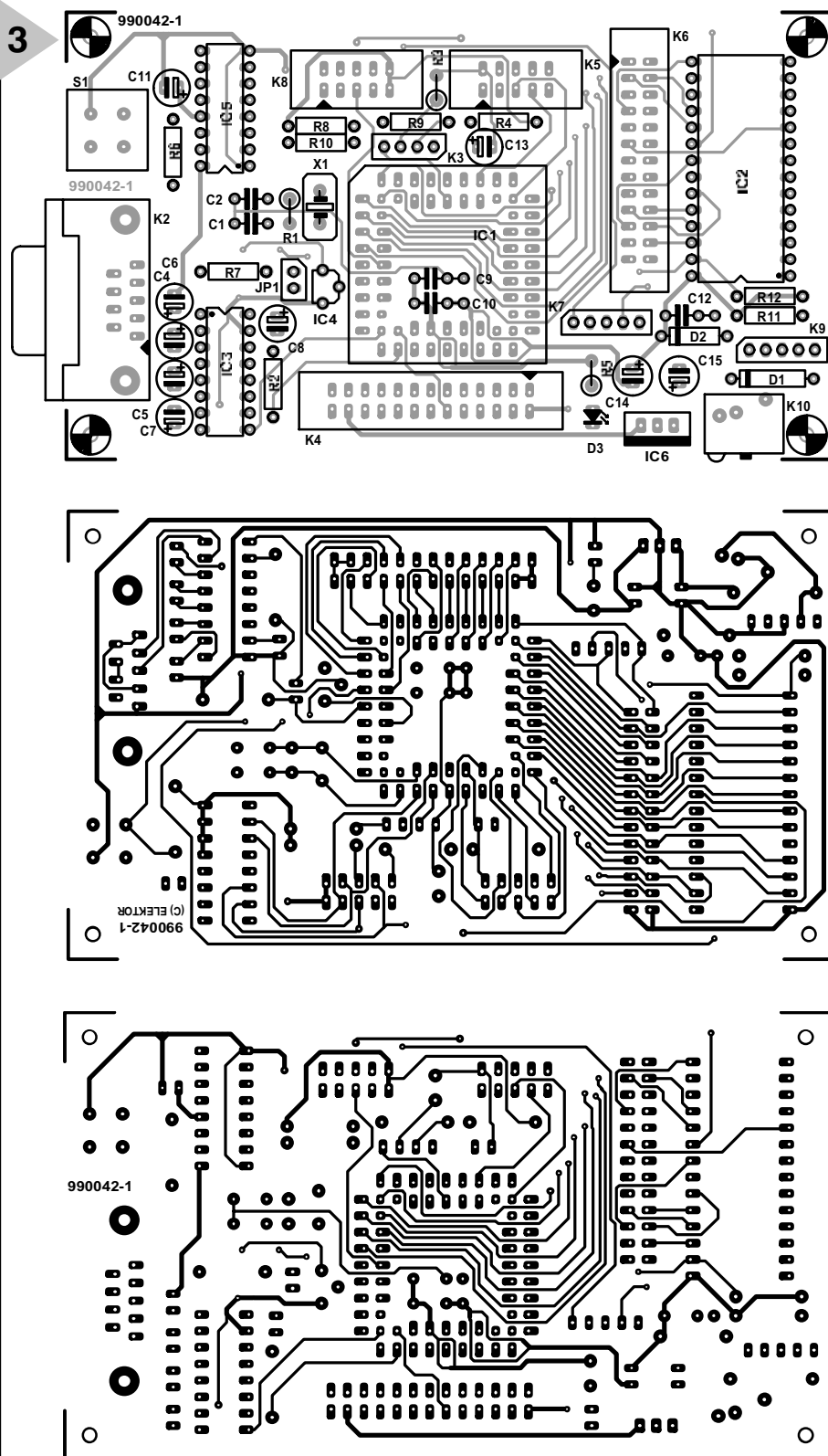
mémoire la plus cher pour une capacité donnée mais c'est la mémoire la plus universelle, et pour beaucoup d'utilisateurs amateurs, l'EEPROM de 8 K devrait suffire.

Le boîtier mémoire est décodé par le signal CS_{PROG} (PG7). Après un reset CS_{PROG} couvre par défaut 64 Koctets. Ceci ne perturbe pas la carte car les autres mémoires, étant internes au HC11, sont prioritaires. On peut donc tout à fait travailler sans reconfigurer CS_{PROG} à condition de se limiter à une zone mémoire de même taille que la mémoire utilisée. Par contre,

CS_{PROG} doit être configuré si on installe une deuxième mémoire externe sur un connecteur d'extension.

Lorsqu'on travaille avec M11, on ne s'occupe pas de la technologie de la mémoire. M11 génère automatiquement le type d'écriture qui convient en fonction de la cartographie de mémoire déclarée dans le menu "hc11_set". Il peut cependant y avoir un problème de frontière si le code est à cheval sur deux mémoires de technologies différentes et dont les adresses sont voisines (rare). Il faudra, dans ce cas, laisser une petite zone libre (environ 40 octets) avant la frontière pour forcer l'assembleur à séparer les blocs

3



Liste des composants

Résistances :

R1 = 10 M Ω

R2 = 4k Ω 7

R3 = 100 Ω

R4 = réseau de 8 résistances de 10 k Ω

R5 = 1 k Ω

Condensateurs :

C1, C2 = 27 pF

C4 à C8, C11, C13 = 10 μ F/63 V

C9 = 10 nF

C10, C12 = 100 nF

C14 = 47 μ F/10 V

C15 = 47 μ F/25 V

Semi-conducteurs :

D1 = 1N4001

D2 = diode zener 6V8/1W3

D3 = LED

IC1 = 68HC11F1 (Motorola)

IC2 = 62256 (RAM)

IC3 = MAX232 (Maxim)

IC4 = MC33064 (Motorola)

IC5 = 74HC139

IC6 = 7805

Divers :

JP1 = embase autosécable mâle à 4 contacts

K2 = embase Sub D femelle à 9 contacts

K3 = embase autoécable mâle à 2 contacts + cavalier

K4, K6 = embase mâle HE-10 à 2 rangées de 13 contacts

K5, K8 = embase mâle HE-10 à 2 rangées de 5 contacts

K7, K9 = embase autosécable mâle à 5 contacts

K9 = jack mâle d'alimentation encartable

S1 = bouton-poussoir

à contact travail

X1 = quartz 8 MHz

support PLCC à 68 contacts

Figure 3. Dessin des pistes et sérigraphie de l'implantation des composants de la platine double face à trous métallisés du système de développement pour 68HC11F1.

dans le fichier S19. Ceci peut paraître un défaut mais il y a peu d'outils qui acceptent de programmer plusieurs mémoires de technologie différentes en même temps.

Remarque : les zones mémoires non déclarées dans le menu "hc11_set" sont considérées comme de la RAM par M11.

LES CONNECTEURS.

La carte est équipée de 4 connecteurs d'extension K4, K5, K6, K8. Dans la majorité des cas, seuls K4 et K5 seront utilisés.

K4 correspond aux entrées/sorties binaires libres des ports A, D et G, K5 correspond au port E. Ces entrées, qui

peuvent éventuellement devenir analogiques, sont routées à part (K5 très proche du port E). Les références du convertisseur analogique/numérique (CAN) sont aussi ramenées sur K5, leur filtrage doit correspondre à des connections courtes.

Chaque port du HC11 peut être utilisé

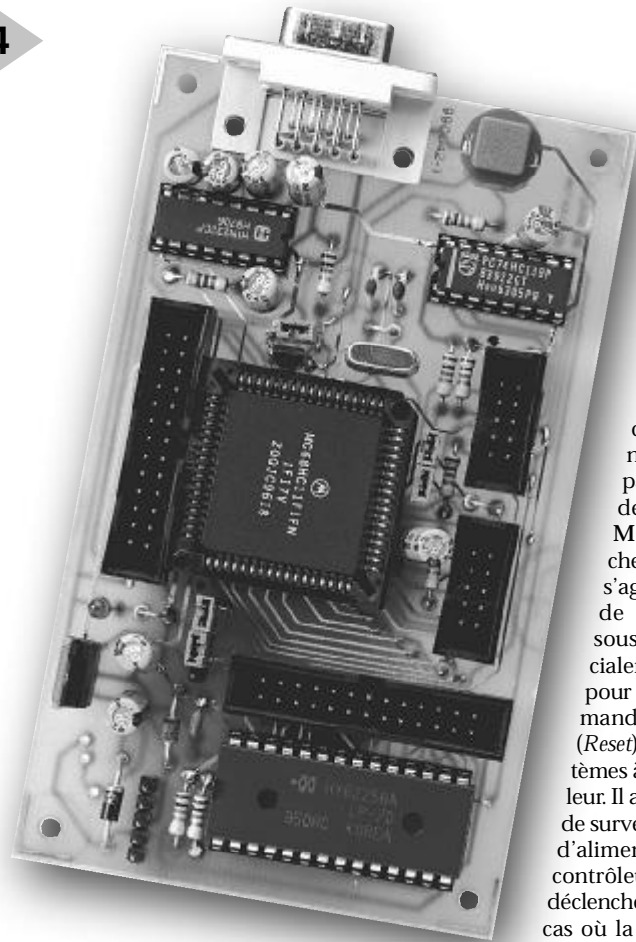


Figure 4. Exemple terminé de notre système de développement pour 68HC11F1.

Les plus attentifs d'entre vous n'auront pas manqué de constater que nous n'avons pas soufflé mot de IC4, un MC33064 de chez Motorola. Il s'agit d'un circuit de détection de sous-tension spécialement conçu pour servir de commande de RAZ (Reset) sur les systèmes à microcontrôleur. Il a pour fonction de surveiller la tension d'alimentation du contrôleur et de déclencher une RAZ au cas où la tension d'alimentation tomberait en-deçà de 4,6 V.

Il est temps maintenant de mettre la main à la pâte et de passer à...

LA RÉALISATION

Nous avons dessiné une jolie platine compacte au coeur de laquelle trône le 68HC11 et dont la figure 3 vous propose le dessin des pistes et la sérigraphie de l'implantation des composants. La mise en place des composants se fera dans l'ordre habituel, en commençant cependant cette fois par le support carré destiné à IC1. On veillera à bien respecter l'orientation prévue, identifiée par l'angle tronqué du support. Toute erreur à ce niveau a des conséquences dramatiques (mauvais positionnement du 68HC11 et partant non fonctionnement du montage, l'extraction du support sans destruction du circuit imprimé étant extrêmement délicate, la seule solution consistant dans ce cas-là à détruire le support à l'aide d'une pince et à extraire une à une les broches comme de vieux chicots). Les condensateurs C9 et C10 seront placés de préférence sur le dessous de la platine. On implantera ensuite les composants de petite taille pour finir par la mise en place des supports destinés à IC2, IC3 et IC5. Pour quoi ne pas utiliser un support tulipe de qualité, voire, pour IC2, un support FIN (à Force

d'Insertion Nulle ou ZIF en anglais) si l'on envisage de changer souvent de type de mémoire ou de programmer fré-

quemment l'une ou l'autre EEPROM. On terminera cette étape de la réalisation par l'implantation des embases et connecteurs. Attention à la polarité des composants en ayant une. Permettez-nous encore quelques remarques d'ordre général :

La soudure

Une carte bien soudée peut durer 20 ans. Si vous n'êtes pas un câbleur professionnel il est fort probable que vous mettiez trop de soudure, si si nous en sommes persuadés. Il faut un fer bien chaud (30 watts) mais fin. Bien chauffer avant de mettre la soudure et chauffer encore 2 à 3 secondes après avoir retiré la soudure.

LE CÂBLE DE LIAISON

Le câble servant à relier le système de développement au PC est du type neutre (absence totale de croisement), on peut donc mettre 2 connecteurs DB9, l'un mâle, l'autre femelle, à sertir sur un câble en nappe. On peut les sertir avec une pince multiprise ou un étai si on est attentif (inutile d'acheter la pince). Il faut faire attention aux numéros de broche (écrits en tout petit sur le connecteur) car il y a une symétrie entre le modèle mâle et femelle.

Nous voici arrivé au moment crucial, à la fois attendu et redouté, par tous ceux qui mettent encore un point d'honneur à réaliser l'un ou l'autre montage eux-mêmes.

PREMIÈRE MISE SOUS TENSION

Ce processus se fera en plusieurs étapes à exécuter dans un ordre précis :

- ✓ Alimenter la carte, sans les composants chers (HC11, mémoire) mais avec le MAX232, (IC3). Vérifier la présence du +5V sur les supports DIL, et la présence des tensions RS232 (environ 10 V sur la broche 2 du MAX et environ -10 V sur la broche 6 du MAX232. Evidemment, la LED de mise sous tension doit être allumée.
- ✓ Éteindre la carte, rajouter le HC11 et éventuellement la mémoire. Mettre 2 cavaliers sur l'embase JP1, les lignes MODA et MODB sont ainsi forcées à la masse, pour avoir le mode *bootstrap* au démarrage (le talker fera logiquement la commutation en mode test).
- ✓ Mettre un cavalier sur K3, ainsi un talker utilisant XIRK est possible (plus intéressant pour les interruptions).
- ✓ Établir la connexion entre l'ordina-

en entrées/sorties binaires ou/et être associé à une interface spécialisée : temporisateur (*timer*) pour le port A, liaisons séries pour le port D, CAN pour le port E.

Pour certaines applications, on peut avoir besoin d'ajouter un boîtier mémoire externe ou un composant d'interface conçu pour être connecté sur un bus microprocesseur. Dans ce cas, il est possible de reprendre les bus sur le connecteur K6 et les signaux de contrôle des bus sur le connecteur K8. Ensuite, il y a encore deux solutions : décodage d'adresse à l'ancienne ou bien récupérer sur le port G les broches qui peuvent devenir des "chip select". Cette dernière solution est meilleure si les broches du port G ne sont pas déjà prises : moins de câblage (mais un peu plus de logiciel) et possibilité de ralentir certaines broches pour accéder à des composants lents (par exemple, un afficheur LCD sur bus).

Il nous reste à parler du régulateur, IC6, ce qui nous amène à évoquer l'alimentation. Elle fait appel à un adaptateur secteur fournissant entre 9 et 12 V. L'embase K9 située à proximité immédiate du jack d'alimentation permet une alimentation directe sous une tension de l'ordre de +8 V depuis un montage-hôte si tant est que celui-ci dispose d'une telle tension régulée. L'allumage de la LED D3 signale la présence de la tension d'alimentation.

teur et la carte (connecteur DB9) et alimenter la carte (7,5 à 12 V).

- ✓ Lancer le logiciel M11.
- ✓ Ajuster les deux menus set :
- ✓ Régler le menu *serial_set* en fonction de votre quartz HC11, pour le reste les valeurs par défaut devraient convenir.
- ✓ Dans le menu *hc11_set* choisir de FE00 à FFFF pour l'EEPROM HC11, FF0 à FF0 (adresse inutile) pour EPROM HC11 car vous n'en avez pas.

Enfin 8000 à 9FFF pour 28CXX, par exemple, si IC2 est une 28C64 ou bien laisser 8000 à 8000 (valeur par défaut) si IC2 est une RAM.

- ✓ Faire ensuite menu (re)start => file.BOO -> HC11.
- ✓ Ecrire
c:\m11disk\m11\TKF1XTST.BOO (c'est un talker pour version F1 utilisant $\text{M}11$ et qui commute en mode test) puis valider. Faire une RAZ (Reset par le bouton-poussoir S1) comme le logiciel vous le demande, puis valider.

Si M11 affiche "Waits echo" ou "bad echo", lire les aides locales (ce menu et *serial_set*)

Si M11 affiche 1 ligne de "OK" tout va bien, (le HC11 a renvoyé en écho ce qu'il reçoit du PC).

Le talker est maintenant en RAM.

- ✓ Démarrer la communication avec le talker (9 600 bauds si le quartz du HC11 est de 8 MHz).

Si tout se passe bien, il apparaît 5 zones mémoires (en haut à gauche) et les registres du HC11 en haut à droite. Le HC11 fonctionne normalement.

Si la mémoire externe est absente, mettre hors-tension, insérer la mémoire sur son support, et redémarrer. Comme M11 conserve le contexte dans le fichier M11.TXT, c'est assez rapide.

- ✓ Vérifiez maintenant les accès en mémoire externe. Afficher \$8000 dans une zone adresse mémoire (en haut à gauche), cliquer dans la zone de données correspondante, et écrire 40 41 42 43 44 45 46 47. Si c'est bon elles s'affichent à l'écran et les équivalents ASCII @ ABCDEFG s'affichent à droite.

- ✓ Cliquer sur '*' à côté de 8000 et vérifier que cette séquence ne réapparaît pas plus loin. Assurez-vous aussi, en utilisant une 2^{ème} zone, l'absence de cette séquence aux adresses 8100 8200 8400 8800 9000 A000 C000 et pour chacune de ces valeurs faire un test d'écriture de valeurs toujours différentes dont au moins une fois \$55 suivi de \$AA.

Test d'un grand nombre d'accès en mémoire externe

- ✓ Charger le programme *Seconde.S19* (chemin : C:\m11disk\samples\F1) et

son fichier symboles *Seconde.LST*.

- ✓ Afficher les variables du programme en écrivant 100 101 103 105 107 dans les zones adresse mémoires. Si vous avez chargé le fichier symboles, le nom des variables doit apparaître.
- ✓ Cliquez sur PC, mettez 8000 cliquez sur '*' à côté de PC (désassemblage). On vérifie ainsi la présence du programme en mémoire externe (sortie par ESC).
- ✓ Déclenchez un chronomètre ou regardez une montre et faire go. Puis, de temps en temps presser la touche 'R' pour rafraîchir l'affichage sans arrêter le programme. Vous pouvez alors vérifier que les variables SEC et SEC_BCD évoluent comme des secondes.

Test des accès en écriture aux PORT A et PORT G

- ✓ Charger le programme *LedF1.S19* (chemin : C:\m11disk\samples\F1) et son fichier symboles *LedF1.LST*.

- ✓ Afficher les valeurs 1000 et 1002 dans la zone d'adresse, les symboles *porta* et *portg* s'affichent. Ensuite procéder à la même séquence que dans le cas du programme précédent. L'observation des sorties sur les ports peut se faire à pleine vitesse si on dispose d'un oscilloscope, ou d'un fréquencemètre. Sinon, mettre un point d'arrêt dans la boucle et observer au voltmètre.

Nous ne pouvons pas passer sous silence, l'un des éléments les plus importants de cette réalisation, vu qu'en son absence, le système n'est qu'un « ramassis » d'électronique sans intérêt, à savoir...

LE LOGICIEL M11

Ce programme vous est proposé, en compagnie des programmes d'exemple évoqués plus haut, sur une disquette, disponible auprès des adresses habituelles, et dont la dénomination est *EPS996005-I*. Voyons un peu ce qu'il sous-entend.

Bien qu'étant sous DOS ce logiciel est assez visuel : fenêtrage, souris,... Pour ceux qui ont une bonne maîtrise de l'Explorateur de Windows (*Windows Explorer*), il est possible d'associer les applications assembleur et éditeur aux fichiers source, éditeur aux fichiers listings, et un raccourci M11 dans le répertoire de travail. On arrive ainsi à travailler très rapidement : assemblage/édition par un simple click (droit) de la souris.

Voici comment procéder pour la Configuration du Win Explorer :

Après un double click sur un fichier *LST*, l'Explorateur Windows vous demande avec quelle application il faut l'ouvrir. Choisir le bloc note (*notepad*) dans la liste. L'Explorateur crée une association qui sera mémorisée et utilisée ensuite pour tous les fichiers por-

tant l'extension *LST*. Faire la même chose pour les fichiers *ASC* et même *S19* si vous savez les lire. Ensuite, il faut faire apparaître l'assembleur en plus du bloc-note pour les fichiers de type *ASC* (le choix entre éditeur et assembleur se fera en cliquant **souris droit** -> **menu local**).

Ceci est un peu compliqué : menu affichage -> option -> types de fichier retrouver le type *ASC* dans la liste marquez le -> modifier -> nouveau -> écrire assembleur dans la ligne titre (en haut) et c:\m11disk\asmhc11\asmhc11.EXE %1;x dans la ligne de commande (c'est tout simple !).

Reste à demander un raccourci pour M11 dans le répertoire de travail (par exemple *samples*). Pour ceci, il existe une méthode très simple quand on est dans l'Explorateur : cliquer sur « m11.EXE » et faites-le glisser dans le répertoire (dossier) *samples*. Puis cliquer souris droit pour l'accès aux propriétés de M11, on choisira parmi les différents onglets les options : fermer en quittant, affichage plein écran, faible sensibilité à l'attente, pas d'écran de veille...

Lune ou l'autre petite remarque en passant : lorsqu'on définit un raccourci pour M11 dans le répertoire de travail, il vaut mieux effacer la rubrique répertoire de travail dans les propriétés programme. Ainsi le répertoire de travail est par défaut, celui où on est avec le Win Explorer.

Une fois configuré, tout ceci fonctionne aussi à partir du poste de travail, on peut donc utiliser l'outil que l'on préfère. (Voir le répertoire \M11DISK\Win95 pour plus d'informations).

Les fichiers .LST contiennent la localisation des erreurs d'assemblage quand il y en a ainsi que la table des symboles.

Pour que l'assembleur fonctionne, il faut que les deux fichiers *offset.ASC* et *codes.ASC* soient présents dans le répertoire (dossier) courant (là où l'on travaille).

M11 ne fonctionnera pas sur un ordinateur en réseau et il doit être utilisé avec l'option plein écran dans Windows 95.

Remarque : le mode plein écran ne vous interdit pas de revenir dans Windows (avec Alt Tab).

Sur des PC lents ou avec des vitesses de transfert élevées (38 400 bauds par exemple), si malgré la propriété faible sensibilité à l'attente, il y a trop de problèmes de surcharge sur la liaison série, il faudra travailler uniquement sous DOS (redémarrer l'ordinateur en mode DOS). De même le fonctionnement un peu spécial (expériences avec le chien de garde, *watchdog*) qui impose une grosse charge de travail au PC ne fonctionne pas bien avec Windows 95. M11 télécharge dans le HC11 en mode *bootstrap* un petit utilitaire de commu-

nication appelé *talker*. La liaison série est utilisée pour cela. Ensuite, le *talker* permet l'accès aux mémoires et registres, ce qui permet à M11.EXE de fabriquer les fonctions de "debug" et de programmation de toutes les mémoires internes du HC11 (RAM, EEPROM, ou EPROM).

En principe, M11 peut s'adapter à tous les HC11 (sauf les versions D qui n'ont pas assez de mémoire interne) et dans tous les modes de fonctionnement avec très peu de contraintes de câblage. Pour d'autres informations sur le logiciel M11 il est possible de relire l'article consacré à l'émulateur 68HC11 (cf. référence [2]).

Mise à jour de M11 à partir d'une ancienne version.

Pour ceux qui possèdent la version de M11 qui a été vendue avec l'émulateur, les nouveautés principales sont : Talkers des versions K1/K4. Ces HC11 à 84/80 broches adressent jusqu'à 1 Mottet de mémoire et intègrent des PWM.

POSSIBILITÉ DE PROGRAMMER LES EEPROM EXTERNES

Quelques pages d'aide ont été modifiées suite à l'article de février 97 (référence [2]).

Quelques possibilités très limitées d'expérimentation sous chien de garde.

Lors de la mise à jour d'une version antérieure, M11 enverra un message d'erreur "problème de taille pour le fichier M11.TXT". En effet ce fichier de contexte contient maintenant les adresses de l'EEPROM externe, il est donc plus long. M11 proposera de réécrire une version par défaut de ce fichier. Il faut accepter cela et refaire votre configuration dans les deux menus SET.

Remarque : chaque projet dans un répertoire différent possède un fichier M11.TXT différent. Vous aurez donc peut-être à refaire ceci plusieurs fois.

Pour cette carte le *talker* de base est TKF1XTSTBOO (cavalier présent entre X1K1 et X1I1). Ce *talker* commute en mode test pour avoir l'accès aux mémoires externes mais sans déplacer les vecteurs qui restent ceux du mode bootstrap.

La vitesse de communication du port série avec ce *talker* est 19 200 bauds avec un quartz 16 MHz (9 600 avec 8 MHz, 14 400 avec 12 MHz).

Vu son plan mémoire, le HC11F1 acceptera de démarrer avec la plupart des *talkers* des versions A1 ou E1. Donc, s'il vous faut un *talker* un peu particulier qui n'est pas référencé en F1, il est possible de piocher dans ceux des versions A1 ou E1 et de rajouter en début de votre programme un LDS #3FF.

DIVERS

Choix d'une valeur de quartz pour le HC11F1

À 16 MHz, la fréquence des accès mémoire atteint 4 MHz (période : 250 ns). En simplifiant beaucoup on peut dire que le temps d'accès spécifié pour la mémoire doit être d'environ la moitié de la période des accès mémoires. Donc 125 ns environ pour un quartz 16 MHz et 250 ns maxi pour un quartz 8 MHz. La valeur 8 MHz permet donc d'utiliser des mémoires courantes.

Remarque : les mémoires fonctionnent souvent très bien à des vitesses un peu supérieures à leur spécification mais elles peuvent réserver des surprises en dehors des cas courants par exemple si la température change.

Utilisation de l'assembleur pour programmer des EEPROM destinées à d'autres applications

On peut générer un fichier S19 contenant des codes quelconques en utilisant les directives FCB, FDB ou FCC de l'assembleur à la suite d'un ORG décalé de l'adresse de base du boîtier mémoire sur la carte HC11.

Il y aurait sans doute encore beaucoup à dire au sujet d'une réalisation aux implications aussi multiples, mais rien de tel que de s'y essayer soi-même...

(990042-1)

Références de documentation :

[1] Motorola : Microcontrollers 68HC11, 68HC12, 68HC16 & MPC500 Families (CD-ROM:)

[2] Emulateur de 68HC11, Elektor n°224, février 1997, pages 22 et suivantes.

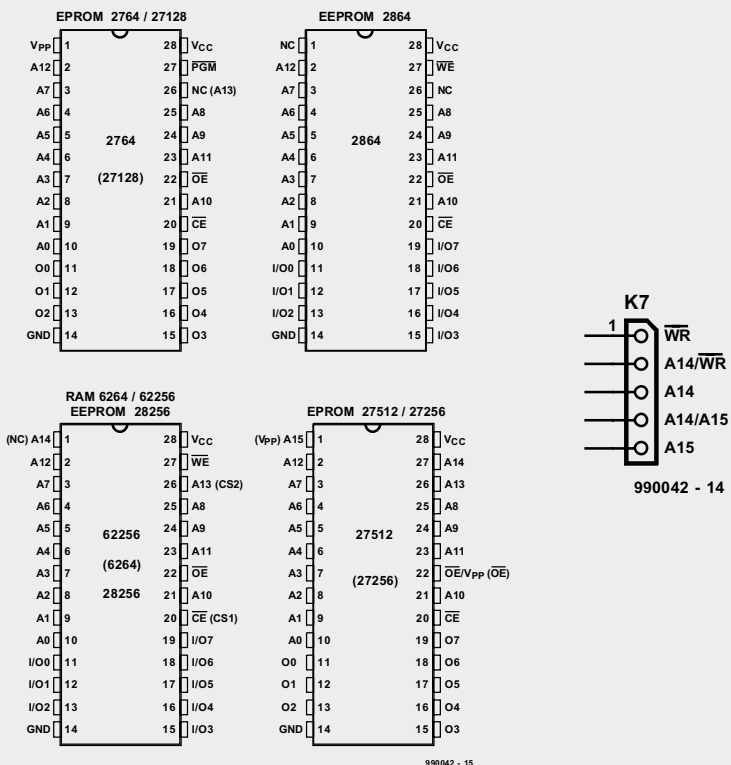
Tableau 1. Positionnement des cavaliers sur le connecteur K7.

Ce tableau récapitule les cavaliers à mettre en place sur l'embase à 5 contacts K7 en fonction du type de mémoire implanté dans le support destiné à IC2. Notons que la broche 1 de cette embase est identifiée par un tronçage.

	1-2	2-3	3-4	4-5
EPROM 8 K (2764)	-	-	-	-
EPROM 16 K (27128)	-	-	-	-
EPROM 32 K (27256)	-	X	-	-
EPROM 64 K (27512)	-	X	-	X
EEPROM 8 K (28C64)	X	-	-	-
EEPROM 32 K (28256)	X	-	X	-
RAM 8 K (6264)	X	-	X	-
RAM 32 K (62256)	X	-	X	-

X = cavalier présent

- = pas de cavalier



**systeme de developpement
pour 68HC11F1,
n°252, juin 1999, page 14
et suivantes**

L'article comporte l'une ou l'autre imprécision en raison d'une disparité entre le schéma, la liste des composants et le dessin de la platine : les dénominations des embases Jp1 et K3 ont été interverties sur le schéma et la liste des composants. JP1 de la platine est une embase à 2 contacts pour cavalier que l'on retrouve sous la dénomination de K3 dans le schéma et la liste des composants. K3 de la platine est une embase à 1 rangée de 4 contacts. La R4 de la liste des composants n'est pas un réseau de 8 résistances, mais 8 résistances distinctes numérotées respectivement R4 et R6 à R12.