

EN QUÊTE DU BON MICROCONTRÔLEUR

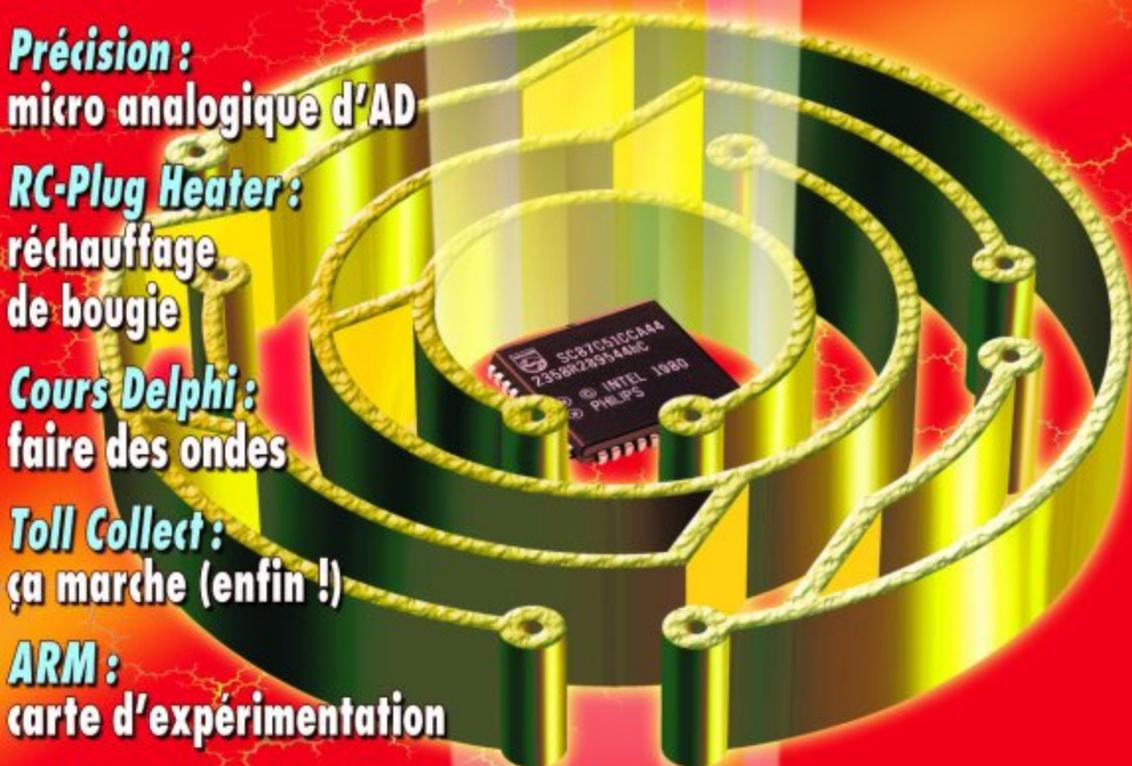
Précision :
micro analogique d'AD

RC-Plug Heater :
réchauffage
de bougie

Cours Delphi :
faire des ondes

Toll Collect :
ça marche (enfin !)

ARM :
carte d'expérimentation



SC-analyser 2005 :
testeur de semi-conducteurs



**Reversed
engineering :**
au cœur du
mini Mac



Nouvelle génération

Les microcontrôleurs, un mot chargé s'il en est dans la communauté des lecteurs de magazines d'électronique. Pour les uns, une bénédiction, un tel composant remplaçant plusieurs dizaines de circuits intégrés classiques, mal à prendre en patience pour d'autres qui n'ont pas eu l'occasion de suivre l'évolution de ce composant tant universel, une « horreur » pour nombre de nos lecteurs qui nous lisent depuis près de 30 ans.

Ce numéro se fait leur champion, vu que près de la moitié de son contenu a un lien direct avec le monde des microcontrôleurs.

Elektor n'a jamais craint de prendre en main les rênes de l'histoire, comme le montre le premier projet à base de μC , un SC/MP de National Semiconductor (cf. la page Rétronique), mais n'a jamais été sectaire non plus, n'hésitant pas à adopter un 2650 de Philips pour l'ordinateur pour jeux TV, voire un 6502, dans notre premier fréquencemètre à μP dont les performances pouvaient aisément de mesurer aux ténors de l'époque, Hewlett-Packard ou Philips. Notre première horloge DCF « intelligente » utilisait elle aussi un 6502. Vers le milieu des années 80 les microcontrôleurs prirent la relève des microprocesseurs, l'une des réalisations de pointe de l'époque, une alimentation de laboratoire à interface RS-232 faisant appel à un 8751. Vinrent les années 90 qui virent l'arrivée d'une nouvelle génération de contrôleurs performants, les DSP, que nous avons utilisé pour la première fois dans notre LCR-mètre « high-tech ». À quoi pouvons-nous nous attendre en ce début de nouveau siècle ? Aux microcontrôleurs 32 bits (et 64 plus tard ???) comme le montre la première réalisation à base d'ARM...



Guy Raedersdorf Rédacteur en Chef



**28^{ème} année, N° 322,
avril 2005**

ISSN 0181-7450
Commission paritaire N° 1004U8313

SEGMENT B.V. / ELEKTOR

Chez W.W.S. International
38, rue la Condamine
75017 PARIS
Téléphone : 01.42.61.18.75
Télécopieur : 01.42.61.18.78
Internet : www.elektor.fr

Banque: CA Ile de France à Paris
Numéro : 18206 00043 41519320001
42 CRCA PARIS OPERA

LIBELLE à « WWS International »

Elektor désire être une source d'inspiration pour ses lecteurs, les intéresser à l'électronique, par la description de projets à faire soi-même, et les tenir au courant des développements en électronique et en micro-informatique.

Elektor paraît 11 fois, le numéro de juillet/août est un numéro double.

Il existe, sous le nom Elektor, des éditions anglaises, allemande et française, et sous celui d'Elektuur, une édition néerlandaise. Elektor est vendu dans plus de 50 pays.

Conformément à la loi "Informatique et Liberté", vous bénéficiez d'un droit d'accès et de rectification des données vous concernant. Sauf refus écrit de votre part auprès du service abonnement, ces informations pourront être utilisées par des tiers.

Rédacteur en chef international :
Mat Heffels

Rédacteur en chef France :
Guy Raedersdorf
(redaction@elektor.fr)

Rédactions :
Harry Baggen, Thijs Beckers,
Jan Buiting, Rolf Gerstendorf,
Ernst Krempelsauer.

Secrétariat de rédaction :
Hedwig Hennekens

Rédaction technique :
Karel Walraven (chef),
Ton Giesberts (concepteur)
Paul Goossens (concepteur),
Luc Lemmens (concepteur)

Maquette et graphisme :
Ton Gulikers, Giel Dols

Directeur/éditeur :
Paul Snackers

Responsable marketing :
Margriet Debeij

Administration des ventes :
(ventes@elektor.fr)

Publicité :
SL Régie - Sophie Lallonder
(sophie.lallonder@wanadoo.fr)

Abonnements Suisse :
Sono Light Import
Champs-Montants 16b
CH-2074-Marin-Epagnier
Tél: 032-710.16.60
Fax: 032-710.16.63
Email: admin@sonolight.ch

DROITS D'AUTEUR :
© 2005 Segment B.V.
Toute reproduction ou représentation intégrale ou partielle, par quelque procédé que ce soit, des pages publiées dans la présente publication, faite sans l'autorisation de l'éditeur est illicite et constitue une contrefaçon. Seules sont autorisées, d'une part, les reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective, et, d'autre part, les analyses et courtes citations justifiées par le caractère scientifique ou d'information de l'oeuvre dans laquelle elles sont incorporées (Loi du 11 mars 1957 -

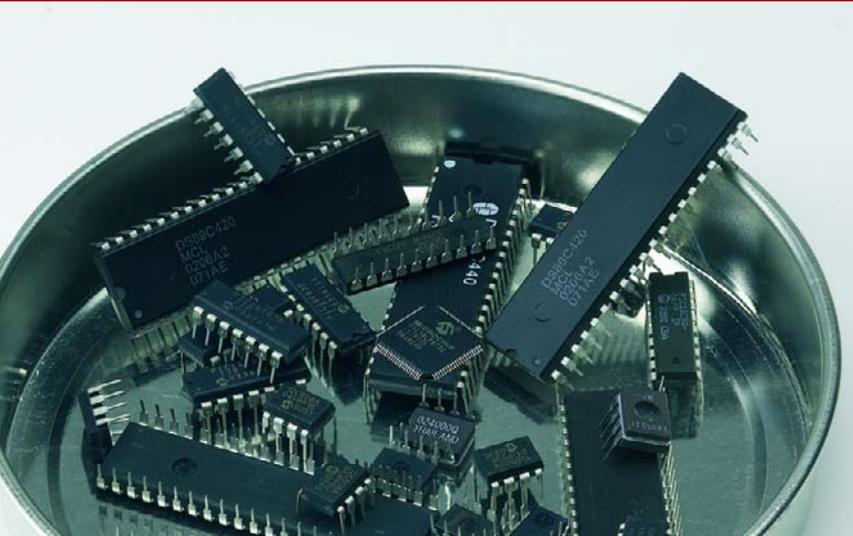
art. 40 et 41 et Code Pénal art. 425). Certains circuits, dispositifs, composants, etc. décrits dans cette revue peuvent bénéficier de droits propres aux brevets; la Société éditrice n'accepte aucune responsabilité du fait de l'absence de mention à ce sujet. Conformément à l'art. 30 de la Loi sur les Brevets, les circuits et schémas publiés dans Elektor ne peuvent être réalisés que dans des buts privés ou scientifiques et non commerciaux. L'utilisation des schémas n'implique aucune responsabilité de la part de la Société éditrice. La Société éditrice n'est pas tenue de renvoyer des articles qui lui parviennent sans demande de sa part et qu'elle n'accepte pas pour publication. Si la Société éditrice accepte pour publication un article qui lui est envoyé, elle est en droit de l'amender et/ou de le faire amender à ses frais; la Société éditrice est de même en droit de traduire et/ou de faire traduire un article et de l'utiliser pour ses autres éditions et activités, contre la rémunération en usage chez elle.

Elektor est édité par Segment B.V.
Siège social : Peter Treckpoelstraat 2-4
6190 VK Beek (L), Pays-Bas
RC Heerlen, nr. 35306

Imprimé aux Pays-Bas par
Tijl-Offset - Zwolle

Distribué en France par M.L.P. et en
Belgique par A.M.P.

20



Au coeur du silicium

Dès que l'on souhaite réaliser une application à base de microcontrôleur, que l'on soit simple amateur ou professionnel chevronné, la première question qui se pose est celle du choix du microcontrôleur. Nous voudrions, par le biais de cet article, vous proposer une sorte de guide insistant sur les facteurs importants lors de ce choix.

28



SC-Analyser 2005

Les transistors sont des composants que l'on rencontre pratiquement dans tous les montages électroniques. L'amateur dispose souvent dans ses tiroirs un certain nombre de ces composants récupérés sur d'anciennes cartes d'origines diverses et qui n'attendent que d'être réutilisés.

Infos & Actualités

- 08 Infos & Actualités
- 12 Courrier Lecteurs
- 14 Toll Collect fonctionne !
- 18 Mac mini
- 20 Au coeur du silicium
- 36 μ C analogique de précision
- 54 Delphi pour les électroniciens (IV)
- 68 Détecteur de brachions

Rubriques

- 05 Colonne-Ours
- 42 Electronique en ligne
- 70 Technique de mesure USB
- 73 Rétronique
- 74 Kit de développement ARM7
- 76 TRICK.e du Prof. OSSMANN
- 77 Secrets du concepteur
- 84 Avant-Première

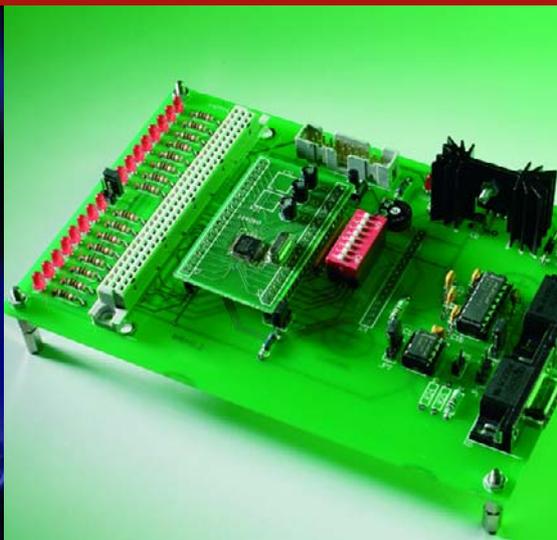
36



µC analogique de précision

Toujours en recherche de puissance de travail et de perfectionnements, les concepteurs ont abouti à la série des microcontrôleurs en 16/32 bits à structure RISC basés sur le concept ARM7TDMI precision analog microcontroller.

44



Carte de développement LPC210x « ARMÉE »

Nous avons abordé, dans l'article du mois dernier, l'ARM, une famille des microcontrôleurs 32 bits. Passons aux choses sérieuses: voici « ARMÉE », une carte de développement ARM puissante à réaliser et programmer soi-même.

56



RC-Plug Heater

RC-Plug Heater - cela fait quand même plus chic que « réchauffeur » de bougie RC- a été conçu pour permettre le chauffage des bougies des modèles réduits radiocommandés (R/C).

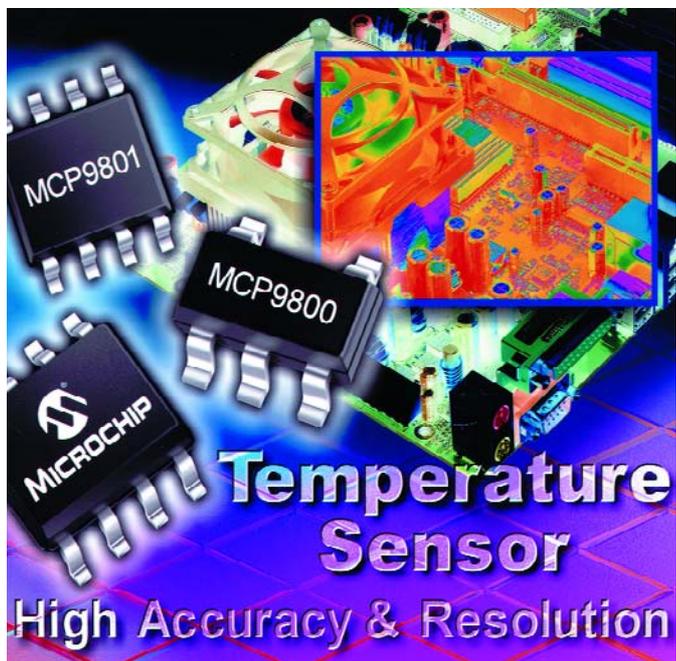
Projets & Réalisations

- 28 SC-Analyser 2005
Testeur de semi-conducteurs
- 44 Carte de développement
LPC210x « ARMÉE » (2)
- 56 RC-Plug Heater
Chauffage de bougie pour modèles R/C
- 66 Télé-enregistreur de température
pour la carte PIC18F

www.elektor.fr tout neuf...

- **Forum** dans lequel vous pourrez bavarder avec d'autres lecteurs,
- **Petites Annonces**, tout un programme,
- **e-CHOPPE**, pour vos achats
- **moteur de recherche**

Nous sommes impatients de vous retrouver...



MCP980X

Capteurs thermiques deux fils

Microchip a annoncé aujourd'hui de nouveaux capteurs numériques de température en boîtiers miniatures SOT-23. Avec une erreur de température maximum de ± 1 °C, ces capteurs fournissent une information de température précise pour la protection et/ou la calibration, en utilisant une surface de carte minimum et aucun composant externe.

Les capteurs MCP980X de Microchip convertissent et communiquent les données de température en environ 30 milli-secondes (pour une résolution de 9 bits) via une interface standard de l'industrie I²C™ ou SMBus. Le temps de conversion peut être augmenté pour obtenir une plus grande résolution jusqu'à 12 bits en fonction des besoins du système.

La temporisation système SMBus supplémentaire disponible sur les capteurs MCP980213 améliore la fiabilité du système en évitant le blocage du bus de communications.

Ces capteurs numériques de température ne consomment que 200 microampères et présentent

une intensité au repos de 1 microampère. En outre, un mode de mesure de la température mono coup peut éveiller le capteur pour effectuer une seule mesure de température avant de revenir au mode veille.

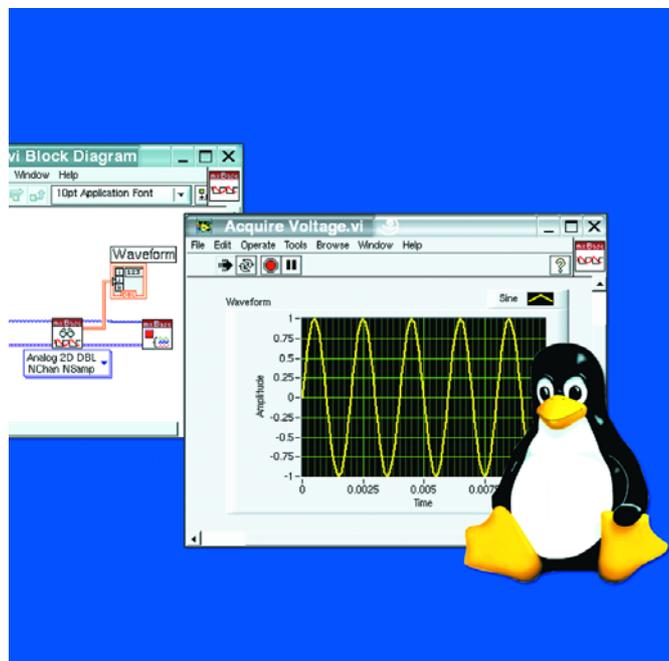
Les MCP9800/2 sont disponibles en boîtier SOT-23 et comportent des adresses esclaves établies en interne. Les MCP9801/3 sont disponibles en boîtiers MSOP-8 et SQIC-8 et comportent des adresses esclaves programmables par l'utilisateur. Le MCP9802/3 comporte une temporisation SMBus supplémentaire. Les échantillons et la production de masse sont actuellement disponibles.

Pour plus d'informations, visiter le site Web de Microchip à www.microchip.com/MCP9800

Points clés:

- Résolution sélectionnable par l'utilisateur jusqu'à 12 bits
- Erreur maximum ± 1 °C (-10 à +85 °C)
- Interface I²C™ ou SMBus
- Boîtiers miniatures SOT-23

(047235-1)



Driver de matériels d'acquisition de données pour Linux

National Instruments annonce NI-DAQmx Base, interface de programmation et driver Linux ouvert pour le contrôle des matériels d'acquisition de données (DAQ) multifonctions de la Série E, des matériels d'E/S numériques NI DIO-96 et des boîtiers USB DAQ de NI. Ce nouveau driver permet aux utilisateurs de Linux d'accéder à la technologie innovante de services de mesures de NI-DAQmx et ainsi d'intégrer facilement les matériels DAQ hautes performances de NI dans leurs applications.

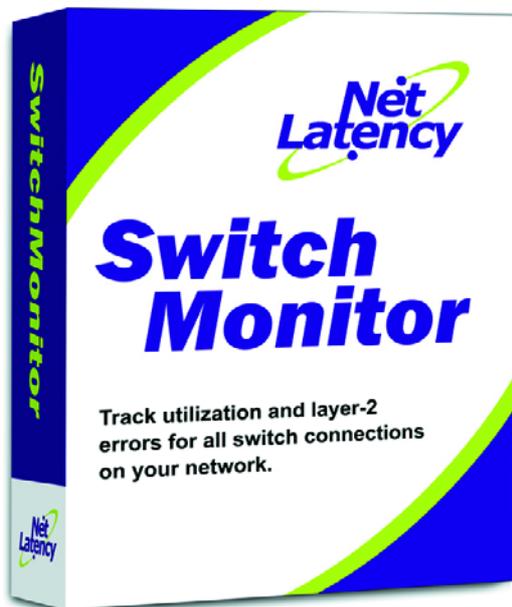
Plus qu'un driver, le logiciel de services de mesures NI-DAQmx Base offre une interface de programmation concise et claire, la possibilité de créer des voies et des tâches par programmation et une intégration étroite avec plus de 70 fonctions d'analyse de LabVIEW; des caractéristiques uniquement disponibles jusqu'à présent avec NI-DAQmx pour Windows. Il inclut également des fonctions en C et des instruments virtuels LabVIEW prêts-à-l'emploi similaires à ceux disponibles dans NI-DAQmx.

Le driver NI-DAQmx Base pour Linux a été construit avec la boîte à outils NI Measurement Hardware DDK (Driver Development Kit) et développé presque entièrement dans l'environnement de développement graphique LabVIEW. Les ingénieurs utilisant actuellement des OS comme VxWorks, QNX et Mac OS X peuvent exploiter les interfaces de programmation de bas niveau incluses dans NI Measurement Hardware DDK afin de créer des drivers pour un large éventail de matériels de mesure NI.

NI et l'acquisition de données

En intégrant les technologies du monde du PC et en investissant sans relâche dans la recherche et le développement, NI continue de réduire les coûts des systèmes de mesure et de contrôle tout en augmentant leurs performances, leur précision et leur fiabilité. Pour en apprendre davantage sur les matériels d'acquisition de données et les avantages de NI-DAQmx, visitez www.ni.com/france/daq.

(047246-1)



SwitchMonitorPro

Pour le contrôle du fonctionnement de votre réseau

Afin de superviser le fonctionnement des réseaux locaux, distants et WI FI, GMI Databox présente le logiciel « SwitchMonitorPro ». Ce nouveau logiciel de supervision des réseaux LAN, WAN et MAN, permet de fournir des indications sur :

- Le trafic réseau en général, sur chaque matériel « *device* » réseau et sur chaque interface réseau.
- Les erreurs sur le réseau en général, sur chaque matériel « *device* » réseau et sur chaque interface réseau.
- La bande passante occupée en temps réel sur chaque interface réseau et bien d'autres éléments d'analyses.

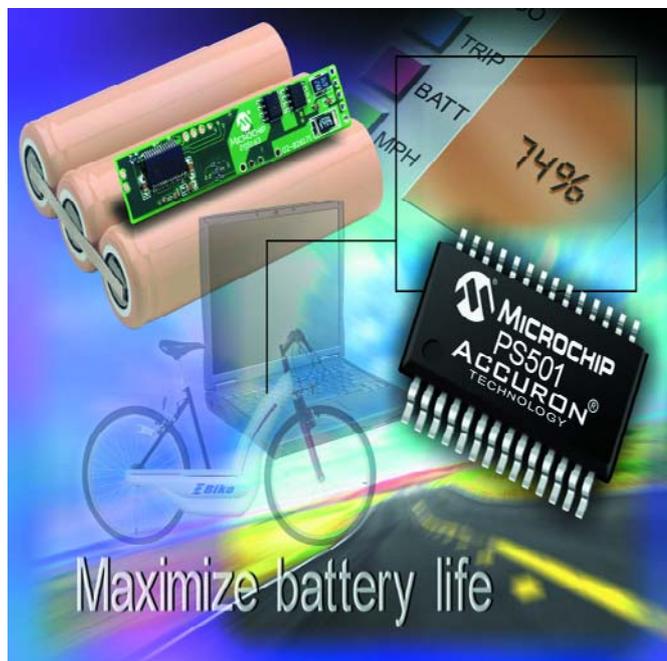
Cet outil de supervision est com-

plet et permet une véritable réactivité face aux anomalies de fonctionnement du réseau.

Son interprétation simple et parfaitement compréhensive en font vite un outil indispensable pour la supervision des réseaux de l'entreprise.

SwitchMonitorPro est idéal pour les réseaux locaux ou multi-sites, il peut superviser le fonctionnement des interfaces, des Switchs (commutateurs), des routeurs, des bornes WI-FI, des serveurs, des caméras, des imprimantes et plus généralement tous les matériels SNMP. Il convient aussi bien à des réseaux de 25 interfaces qu'à des réseaux de 25 000 interfaces.

(057029-1)



PS501, circuit de gestion d'accu

Pour applications à base de Li-Ion, NiMH ou CdNiH

Le PS501 de Microchip est en fait une gauge de niveau d'énergie (les américains appellent cela *fuel-gauge*) intégré, reprogrammable, tout un programme, pour 2, 3, ou 4 séries de cellules Lithium-ion (Li-Ion) et lithium-polymère (LiPo), ou encore 6 à 12 cellules NiMH (nickel métal hydrure) ou cadmium-nickel (CdNi). Ce système de suivi d'accumulateur donne de très bonnes prévisions quant à la capacité du système d'alimentation, la « pile ». Il monitor les cellules quant aux conditions de sécurité tout au long de la décharge, maximisant de ce fait la durée de vie de l'accumulateur et sa durée d'utilisation.

Ce circuit combine un microcontrôleur PIC18 de Microchip avec

l'algorithme Accuron de Microchip stocké dans 16 Koctets de mémoire Flash embarquée. Il travaille à 150 μ A. Lorsqu'il ne travaille pas, le circuit passe en mode sommeil au cours duquel il consomme moins de 1 μ A, optimisant la capacité et évitant une décharge profonde de l'accumulateur. Le circuit possède, entre autres périphériques, un CAN Δ/Σ 16 bits et une circuiterie signal mixte optimisée application, permettant des mesures précises du courant de l'accumulateur, de ses température et tension ainsi qu'un branchement direct d'un maximum de 4 packs de cellules Li-Ion ou Li-Po prises en série.

(057019-1)

Publicité



TOPAS86FS49

Starter Kit pour le TMP86FS49

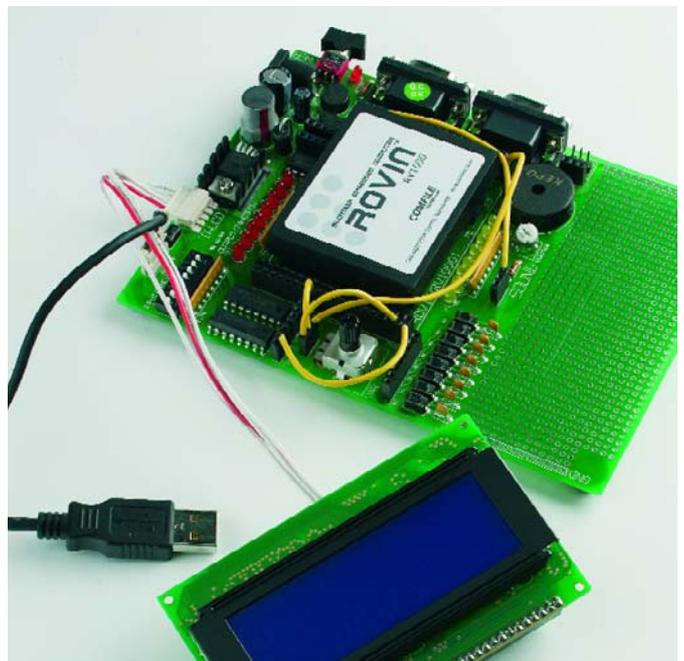
Toshiba Europe vient de mettre sur le marché une plate-forme de développement mettant à disposition tout le matériel et les logiciels nécessaires à la conception, à la réalisation de prototype, au test et au débogage d'applications enfouies (*embedded*) ayant pour cible le marché des appareils domestique tant industriel que grand public. Cet outil repose sur le nouveau microcontrôleur 8 bits doté de 60 Koctets de mémoire Flash de Toshiba, le TOPAS86FS49 Starter Kit peut permettre de se passer de tout outil de développement et de débogage additionnel souvent coûteux.

Le microcontrôleur TMP86FS49 incorpore de la mémoire SuperFlash™ de SST et des périphériques ciblés tels que bus I²C, des CAN 10 bits et des canaux UART standard. Le TOPAS86FS49 Starter Kit combine une carte matérielle de développement à du logiciel de développement et de débogage, un programmeur de mémoire Flash, des bibliothèques logicielles étoffées, des exemples et des

notes d'application.

Outre le microcontrôleur TMP86FS49, la carte de développement faisant partie du Starter Kit incorpore un module LCD, un clavier, des boutons-poussoirs de démarrage de processus, des LED d'état, une EEPROM, un résonateur piézo-électrique, une embase à laquelle pourra être connecté un haut-parleur, sans parler d'un certain nombre d'interfaces dont RS-232, I²C et USB. En tout état de cause, la clef de la simplification du développement de programmes est l'IDE (*Integrated Development Environment*) dans lequel Toshiba a emmaillotté son matériel. Ce cadre omniprésent et omnipotent met à la disposition de l'ingénieur de conception tout ce dont il a besoin pour déboguer, créer, éditer et programmer un programme, sans même parler de la possibilité de le simuler. On trouve ainsi, par exemple, un outil très puissant rarement disponible dans le cas d'une solution 8 bits, à savoir un moniteur en ROM.

(057024-1)



ROVIN™

À base d'ARM7DMI™

ROVIN™ est un module microcontrôlé multi-tâches programmable de chez Comfile Technology, société coréenne dont la notoriété en France est beaucoup due à ses produits PICBASIC distribués et supportés dans l'Hexagone par Lextronic.

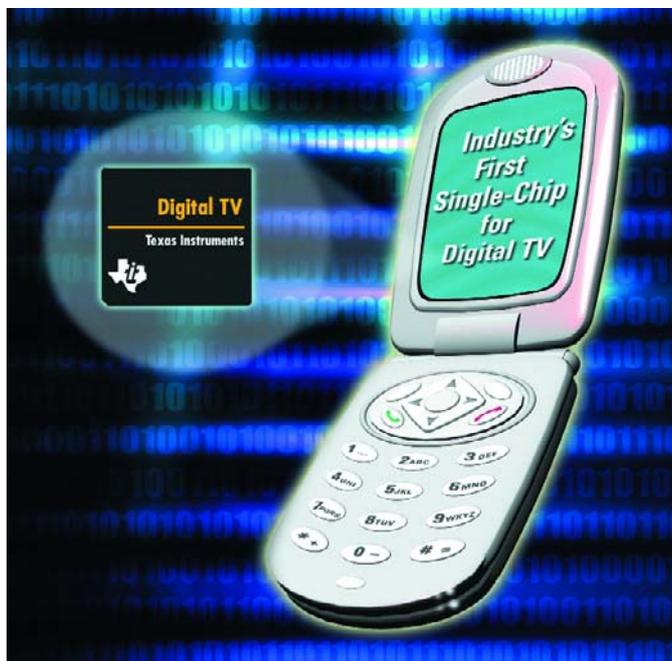
Le kit de développement ROVIN que nous a fourni cette société de compose du module ROVIN™ proprement dit, à base d'ARM7DMI™ d'ARM™, un puissant processeur 32 bits cadencé à 83 MHz et d'une carte de support, dans laquelle il vient s'enficher, carte baptisée Quick.Start.Board. À cela s'ajoutent un câble spécial et un CD-ROM. Comme on le voit bien sur la photo, cette carte dotée de son ROVIN™, ce petit module capoté de la taille d'une demi-boîte d'allumettes, permet d'effectuer des manipulations dès qu'elle est mise sous tension.

Il existe sur cette carte un champ d'expérimentation à pastilles sur lequel l'utilisateur peut réaliser sa propre électronique d'extension, mais il est préférable, pour éviter d'abîmer cette partie de la carte

de support, d'envisager l'utilisation d'une ou plusieurs plaquettes d'expérimentation rapide que l'on fixera à l'aide d'adhésif double face sur le champ à pastilles. L'intérêt majeur de ROVIN™ est qu'il se laisse programmer très simplement en langage C. Autre aspect important, la documentation, toute en français, tant le manuel que l'interface d'environnement (IDE) de même que les exemples. Un travail de titans. Pourquoi ROVIN™ ? Il s'agit d'un module très puissant, disposant de 2 ports série, étant capable d'effectuer des calculs sur 64 bits, possédant 60 lignes d'E/S de sorte que certaines de ses lignes assurent la communication I²C, d'autres la SPI, le comptage, les opérations MLI, le traitement d'interruption, la génération de sons. Tout un programme.

À noter qu'il existe un joli manuel consacré au ROVIN™ à télécharger depuis le site de Lextronic : www.lextronic.fr Nous aurons sans doute l'occasion de nous y intéresser de plus près dans un prochain numéro.

(057018-1)



« Hollywood », puce de télévision numérique

La télévision numérique directement sur votre télé- phone portable !

La première télévision numérique sur une puce unique : Texas Instruments annonce en exclusivité la puce « Hollywood » qui permettra bientôt de recevoir et visionner vos programmes télé directement sur votre téléphone portable...

Vous allez bientôt pouvoir regarder vos programmes favoris sans avoir à rester assis sur votre canapé, grâce à une nouvelle puce que Texas Instruments met actuellement au point. Texas Instruments annonce la conception, sur une puce unique, de la première télévision numérique pour téléphones portables capable de capter le signal de télévision et de permettre ainsi aux utilisateurs de téléphones portables de recevoir des programmes en direct, qu'il s'agisse d'un « reality show », d'une rencontre sportive ou d'un flash d'informations. La puce nécessaire à l'opération, baptisée « Hollywood », permettra la réception

d'émissions de télévision numérique en direct grâce à une nouvelle infrastructure de télévision destinée à la téléphonie cellulaire en cours de développement. Véritable révolution technologique, « Hollywood » est à la téléphonie ce que la TVHD est à la télévision.

« Hollywood » a été conçu sur la base du savoir-faire de TI dans les domaines convergents de la téléphonie sans fil et de l'électronique grand public, notamment l'exploitation de contenus vidéos haute qualité en streaming sur des combinés 2,5 G et 3 G grâce à ses processeurs multimédia OMAP™. Ce dispositif complète utilement la gamme actuelle de TI en électronique de consommation, qui compte notamment la technologie DLPT™ et une technologie analogique et de traitement du signal éprouvée destinée à des produits tels que les appareils photos numériques, les lecteurs audio ou la radio numérique. (Voir : www.ti.com/hollywoodpr).

(047250-1)



PIC® Flash

Avec connectivité USB 2.0 pleine vitesse

Microchip a annoncé aujourd'hui les premiers microcontrôleurs PIC® Flash avec connectivité USB 2.0 pleine vitesse et fonctionnant à 48 MHz pour des débits de transfert de 12 mégabits/seconde. Associées à de nombreux périphériques et à la technologie nanoWatt de gestion de la consommation, ces caractéristiques fournissent une solution complète pour les concepteurs qui travaillent avec USB dans des applications industrielles, médicales et d'autres applications embarquées.

La plupart des microcontrôleurs disposant d'USB sont optimisés pour les ordinateurs personnels et les applications grand public plutôt que pour des applications embarquées. La nouvelle famille de microcontrôleurs PIC USB de Microchip apporte les avantages d'USB pleine vitesse à une gamme plus large d'applications qui peuvent fonctionner dans des environnements difficiles et se connecter moins souvent à des ordinateurs personnels. Les applications-cibles de ces

produits se trouvent dans les marchés industriel, médical, et automobile, ainsi que dans des applications grand public avancées et d'autres appareils alimentés sur batterie et connectés à un PC. L'interface USB 2.0 pleine vitesse comporte un émetteur/récepteur intégré et un port de flux parallèle pour les transferts directs de données vers des périphériques externes avec une charge minimum pour l'unité centrale. Une autre caractéristique essentielle de cette famille est sa mémoire Flash améliorée auto programmable de 24 ou 32 Koctets, permettant une mise à jour sur le terrain via le port USB. Utilisant une technologie Flash PEEC (PMOS à effacement électrique) de Microchip, ces circuits présentent une endurance allant jusqu'à 100 000 cycles d'effacement/écriture et une longue rétention de données de plus de 40 ans. Pour plus d'informations, visitez le site Web de Microchip à www.microchip-usb.

(047236-1)

Wireless Connectivity

Bonjour, sur le magazine Elektor (bien sûr!) du mois de Février 2005, concernant l'article "Wireless Connectivity", sur le "tableau 1. Bandes ISM usuelles" en page 27, on a un texte en allemand dans la colonne "5 - Bluetooth" / "Inconvénients", et moi, j' comprend pas l'allemand...

Un copier/coller qui n'a pas traduit à 100% l'article d'origine, sans doute... Voilà,

Greg (via E-mail)

Il ne s'agit pas même d'un copier/coller raté, mais d'une adjonction a posteriori de la rédaction allemande au dit tableau, de sorte que cela a échappé à mon attention. Voilà ce que dit le texte...

Propagation problématique (quasi-optique), parasitage par fours micro-ondes

Mise en ligne d'un nouveau site

Je viens de mettre en ligne uni-son.com, un Site spécialisé dans l'audio sur lequel je travaille depuis plusieurs années.

Il contient déjà pas mal d'articles, projets et montages dont de multiples informations introuvables ailleurs. Puisque votre numéro de mars traitera plus particulièrement de ce sujet, votre lecteur pourrait être intéressé d'apprendre sa création. Bien fidèlement.

Marc Basquin

Je suis désolé, mais comme vous le savez sans doute, la publication d'un magazine ne se fait pas au jour le jour.

Le numéro de mars est, à l'heure où vous lirez ces lignes, déjà chez nos lecteurs... mais cela ne m'empêche pas d'en faire mention dans le courrier des lecteurs...

BOUQUET DE FLEURS

Cher Monsieur, J'ai trouvé votre adresse dans le numéro de décembre. Comme à l'accoutumée, ce numéro est passionnant et il couvre un domaine que j'affectionne tout particulièrement, à savoir domotique et microcontrôleurs PIC.

En lisant la liste des adresses internet de la page 69 du mois de décembre, je me suis dit qu'il y avait une adresse qui manquait, qui serait pourtant utile à vos lecteurs. Il s'agit de www.intellihome.be.

Je suis client de ce distributeur on-line de produits domotiques depuis plus d'un an je crois, et je pense qu'il est intéressant pour vos collaborateurs et à faire connaître à vos lecteurs. J'ai déjà passé plusieurs commandes, et j'en ai toujours été très satisfait. Voici pourquoi :

- Leur site est bien fait, suffisamment attractif sans tomber dans l'excès.
- Il est multilingue (personnellement ça ne me dérangerait pas en anglais, mais je suis sûr que ce n'est pas le cas pour tous les lecteurs).
- Les prix, bien que plus élevés qu'aux US, sont toutefois attractifs pour l'Europe et sans concurrence en Europe à ma connaissance.
- Il annonce les prix TTC.
- Il dispose d'une aide en ligne courte et efficace, sérieux "plus" pour le client qui se pose des questions à la commande et qui lui permet de s'en sortir sans problème.
- Les délais de livraison sont impeccables (24 h chrono, départ Belgique livré Toulouse c'est exceptionnel dans le monde d'aujourd'hui... toute commande passée avant 14 h part le jour même, j'ai essayé et c'est vrai...)
- La facturation n'a lieu qu'au départ de la livraison (certains facturent dès la commande et font parfois partir des colis vides, ou acceptent la commande sur des stocks épuisés! Si, si, et - dans le pire des cas - j'ai mis 8 mois à récupérer l'avoir que l'on m'avait généreusement attribué sur 2/2 commandes insatisfaites...)
- Ils respectent leur client et tiennent leurs promesses...

(Un tout petit "bémol" du côté des informations sur les produits sur le site. Celles-ci sont quelquefois soit insuffisantes, soit manquent de précision, surtout pour le néophyte. Mais cela n'est qu'un détail mineur au regard du reste, surtout lorsque l'aide en ligne est là pour résoudre le problème).

Ayant passé une nouvelle commande ce matin, je leur ai parlé de votre revue et me permettez de les joindre en copie de ce message, afin qu'ils sachent que leur politique de satisfaction du client est reconnue...

Quand à Elektor, je tiens à vous témoigner également de ma satisfaction. Je suis un lecteur occasionnel depuis environ 20 ans, je me suis abonné lorsque vous avez annoncé (septembre 04?) que l'année allait être riche en articles sur le PIC. A propos, www.microchip.com, voilà encore un exemple à citer, mais vous l'avez déjà fait dans votre revue. Je tiens à vous dire que là aussi le respect du client, le petit comme les gros, est à la clef!

Je n'ai pas l'habitude de faire des courriers aux rédactions, mais je me suis dit que cette fois-ci ça valait la peine, pour un petit coup de coeur.

Bravo pour votre revue...

Michel Charpentier

Il ne nous arrive pas souvent de publier ce genre de courrier, c'est même la première fois. Mais votre plaidoirie est si convaincante.



À nos lecteurs

Téléchargements gratuits !

Sur le nouveau site d'Elektor tous les articles peuvent être téléchargés moyennant une modique contribution de 1,35 € par téléchargement. Lors de la construction du site, les dessins des platines qui n'avaient délibérément pas été imprimés dans la revue (le plus souvent par manque de place) et étaient de ce fait disponibles gratuitement, se sont malencontreusement retrouvés dans la catégorie des téléchargements payants du nouveau site. Nous avons rétabli la gratuité de ces téléchargements afin que les lecteurs du magazine imprimé puissent disposer de la publication complète sans frais supplémentaires, et présentons nos sincères excuses à nos lecteurs lésés par cette erreur.

L'Éditeur

vez télécharger, entre autres, depuis :

www.modul-bus.de/mbnews/mbnews02/isp.htm

Ouvrez le menu « Parameter » dans le coin supérieur gauche de l'écran; choisissez ensuite le port COM et optez pour DK7JD. (Mr Vooy nous a signalé depuis, par E-mail, que cette solution fonctionnait parfaitement, la Rédaction).

qui eu seront mis au courant de l'existence de ce nouveau site dans le numéro d'avril.

Carte 89S8252 Flash vs PC rapide

Chère équipe d'Elektor, j'ai utilisé la carte 89S8252 Flash (décembre 2001) pendant près de 2 ans avec grand plaisir, ayant réussi à y connecter nombre d'applications telles que CAN, CNA et autres systèmes de commande. Cependant, depuis le début du projet, en 2002, je rencontre des problèmes lors du téléchargement de programmes. Après quelques expériences, il apparut que mon PC tournant à 2,8 GHz, ne fonctionnait pas correctement avec l'utilitaire MicroFlash, tandis que ma vieille machine tournant à 433 MHz fonctionnait comme un charme. Il y a quelques jours, j'ai refait le test avec les mêmes résultats : problèmes de communication lors du téléchargement de programmes avec le PC rapide. Existe-t-il une solution ?

Bram Vooy

Nous avons été surpris d'apprendre que vous aviez des problèmes sachant que plusieurs milliers de ce type de carte ayant été vendus depuis décembre 2001. Il existe une solution : utiliser un utilitaire baptisé ATMELISP que vous pou-

ATMELISP amélioré

Ayant lu une question concernant la lecture de micros AT. J'aimerais ajouter que j'ai créé une version de Atmelisp offrant de nouvelles fonctionnalités et qui permet de télécharger des programmes en BASIC et BASIC-52, tant dans le sens montant (upload) et que descendant (download). Le programme utilise soit 2 ports sériels conventionnels soit un port USB et un port RS-232, le port USB étant utilisé pour la communication de données par le biais d'un convertisseur USB->RS-232. Mon programme a été testé sous Windows 98, ME et XP Home Edition. Je suis disposé à proposer les fichiers binaires aux lecteurs intéressés.

Piet van der Wal

Les lecteurs que cette proposition intéresse peuvent demander à la rédaction les détails de contact avec Mer van der Wal en envoyant un E-mail à la rédaction redaction@elektor.fr en mettant comme sujet atmelisp.

MISES AU POINT

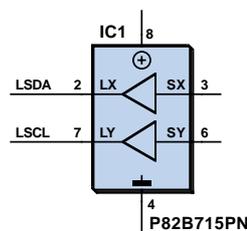
Interface USB/I2C (040334-1)

Elektor n° 318, décembre 2004, page 24 et suivantes

Bis repetita placent

Il y a contradiction entre les 2 schémas.

Ce n'est pas celui de la page 25 qui est faux, au contraire, il est OK lui, mais ce sont malheureusement ceux des pages 19 (figure 2) et 20 (figure 4) sur lesquels le schéma IC1 est « monté à l'envers ». Sur les 2 schémas, il faudra inverser la dénomination des broches 2 et 3 qui deviennent respectivement 3 et 2, et celle des broches 7 et 6 qui deviennent 6 et 7 (Cf. le dessin ci-dessous qu'il vous faudra photocopier 2 fois avant de les coller par-dessus les



IC1 des schémas d'origine.) Il s'agit, pour IC1, d'un composant bidirectionnel, la section à impédance plus faible devant se trouver reliée à la sortie de l'amplificateur. On voit en outre sur la figure 1 que la broche la plus à l'extérieur du jack destiné à l'alimentation est reliée à la masse, alors que sur la platine c'est précisément le contraire.

Extension de commande IR rustique, (030103-1) Elektor juillet/août 2004, page 123

L'article mentionne le transistor T1 type BD240 comme un transistor de "postcombustion" ...

Le transistor BD240 est un transistor de type PNP.

Sur le schéma on retrouve pour T1 le symbole d'un transistor NPN...

1° Est-ce la référence du composant qui est correcte ou est-ce le schéma ? Merci de me m'éclaircir !

2° Comment fait-on pour mesurer le courant dans les LED IR ? C'est plutôt difficile étant donné que le rapport cyclique des signaux IR est faible. Y-a-t-il un "truc" pratique ?

Salutations, John Roose

Le schéma de la page 123 comporte une petite erreur de dénomination : T1 n'est pas un BD240 qui est un PNP, mais un BD241, un NPN tel que le représente le schéma...

Un double merci à Gilbert Deloge et John Roose.

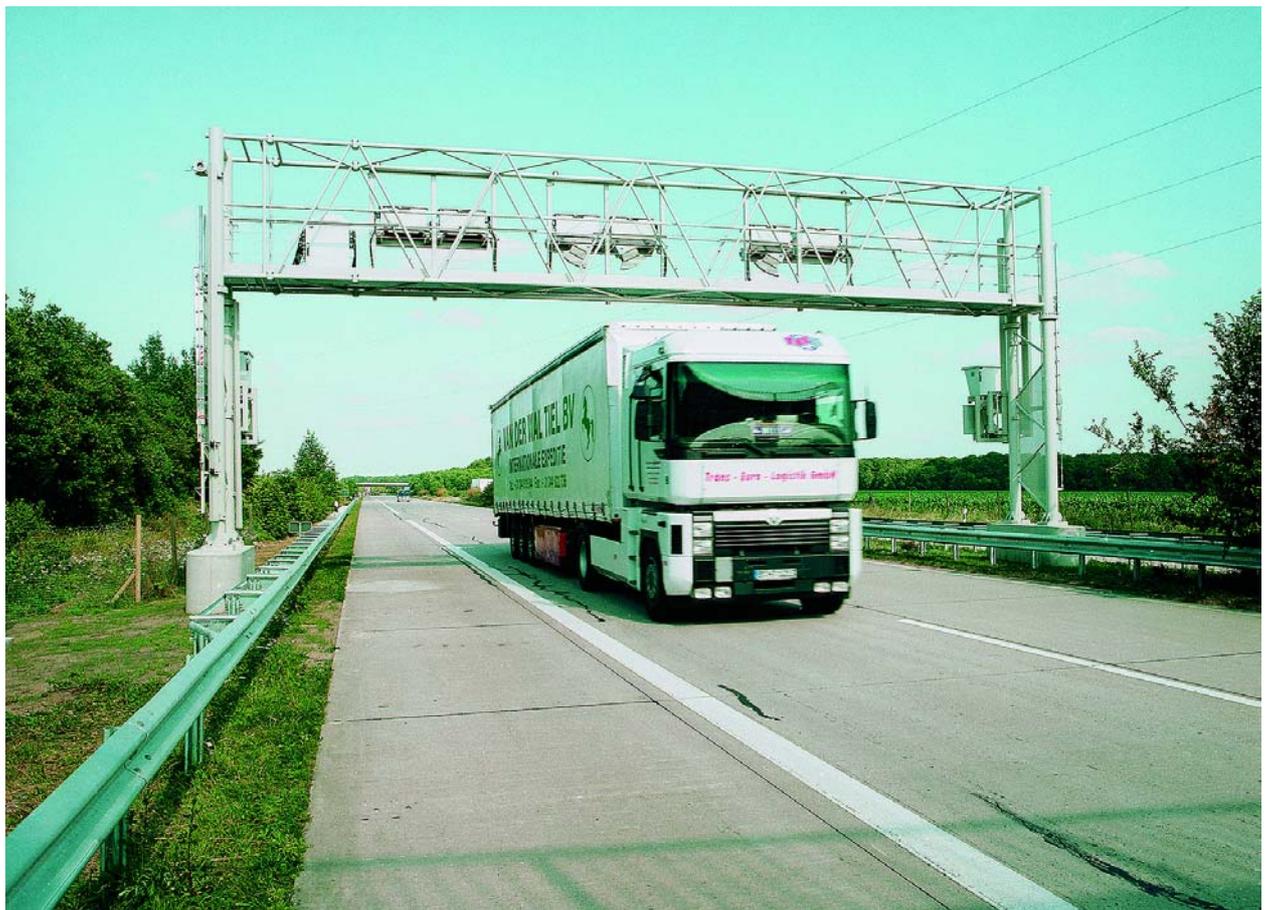
Il faut mesurer le courant aux bornes de la résistance R7 (56 kΩ) et prendre la valeur de crête avec un multimètre qui puisse le faire...

Règles du jeu

- Publication de la correspondance de lecteurs à la discrétion du Rédacteur en chef
 - Les points de vue et opinions exprimées par les correspondants ne sont pas nécessairement ceux du Rédacteur en chef ou de l'Éditeur.
 - La correspondance pourra, le cas échéant, être traduite ou éditée en longueur, clarté et style.
 - En cas de réponse à COURRIER, veuillez S.V.P. indiquer le numéro concerné.
 - Veuillez S.V.P. adresser votre correspondance : redaction@elektor.fr ou
- Rédacteur en chef Elektor
 Chez W.W.S.
 38, Rue la Condamine
 75017 Paris - France

TOLL COLLECT

Mais comment ?



Ekkehard Scholz

Le système allemand de collecte du péage pour poids lourds est célèbre pour sa complexité technique. Il s'appuie sur 3 technologies : GPS, GSM et l'infrarouge.

Il est enfin en service après des difficultés de fonctionnement prolongées. Nous vous révéleront tout sur son fonctionnement.

FONCTIONNNE!

Le système de Toll Collect destiné à collecter les frais de péage pour poids lourds de plus de 12 tonnes est enfin en service après avoir connu des problèmes de fonctionnement aussi spectaculaires que coûteux. On a craint le pire, mais le tout – ou plus exactement une version légèrement simplifiée – fonctionne de manière satisfaisante après une ultime « répétition générale » en décembre 2004. Le chaos aux terminaux de péage prévu pour le début de l'année n'était pas au rendez-vous. La barre a été placée très haut lors du développement du système allemand de péage pour poids lourds :

- Le calcul de la taxe tient compte aussi bien de la catégorie de pollution que du nombre d'essieux.
- Il doit être possible techniquement de créer de nouvelles classes de taxe incluant par exemple l'heure et le lieu du trajet.
- Pour permettre à un poids lourd soumis au péage de ne pas s'arrêter (*free flow system*), il est possible de s'enregistrer automatiquement au moyen d'un appareil embarqué (*On-Board Unit, OBU*).
- Il est aussi possible de s'enregistrer manuellement à un terminal de péage de l'autoroute ou par Internet.

Le système de péage devra être techniquement (euro)compatible avec des systèmes similaires. Selon l'exploitant Toll Collect et le Ministère Fédéral de la Circulation, ce système de péage remplit toutes les conditions nécessaires pour rencontrer un grand succès à l'exportation.

Enregistrement par OBU, restez assis

L'enregistrement automatique pour des routes à péage nécessite le montage d'un appareil de la taille d'un autoradio dénommé *On-Board Unit (figure 1)*. L'enregistrement automatique constitue le « moindre effort » pour le chauffeur de poids lourd : pas besoin d'enregistrer soi-même son trajet ni de procéder à d'éventuelles modifications ou suppressions. Chaque appareil OBU est attribué à un certain poids lourd avant même d'être monté. Les données principales du véhicule sont mémorisées à demeure dans l'appareil.

L'appareil se met en marche lors du démarrage du véhicule et reprend toutes les données préenregistrées. Si certaines d'entre elles ont changé depuis le trajet précédent, le conducteur peut indiquer



Figure 1. Appareil du véhicule (*On-Board Unit, OBU*) pour l'enregistrement automatique sur les tronçons à péage. Il contient les données propres au véhicule que le conducteur peut mettre à jour (par exemple lorsqu'il roule sans remorque). (Cliché : Grundig)

- si le poids lourd pèse plus ou moins de 12 t (si le véhicule roule sans semi-remorque ou sans remorque et échappe donc au péage),
- le nombre d'essieux du train routier (varie selon la combinaison de remorques et détermine le montant du péage),
- s'il s'agit d'une nouvelle tournée ou s'il faut continuer à payer à partir du montant de la dernière tournée,

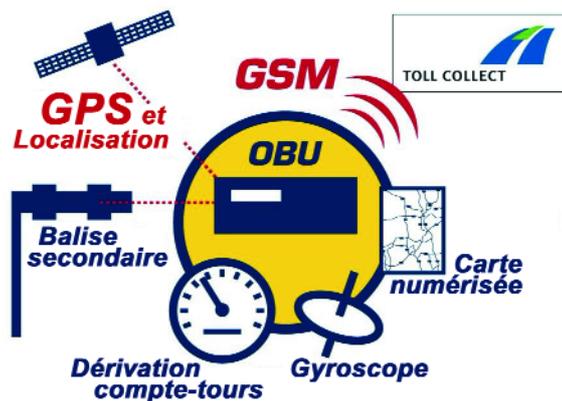


Figure 2. Les données du système GPS (Global Positioning System), du tachymètre et du gyroscope permettent à l'OBU de déterminer la position et le trajet à 10 m près sur une carte enregistrée numériquement. Le péage calculé est envoyé au point de collecte par GSM (téléphone mobile). (Figure : Toll Collect)

– quel centre de frais doit être enregistré (cette saisie facultative permet à l'entreprise de transport d'identifier la tournée et le véhicule lors du décompte mensuel du péage).

OBU est activé après la saisie des données. Il détermine si le véhicule se trouve sur un trajet à péage : le signal GPS (système de navigation par satellite) et les données d'autres capteurs de repérage (figure 2) sont comparés au réseau autoroutier enregistré numériquement. Le repérage fonctionne comme celui du système de navigation d'une voiture particulière. La précision du capteur GPS est améliorée par les données d'autres capteurs. Le repérage « à l'estime » est effectué par un gyroscope (avec capteurs piézoélectriques). Le déplacement et la direction du véhicule sont fournis par le signal tachymétrique. Le repérage à l'estime est indépendant du GPS. Les résultats du repérage par satellite et du repérage à l'estime sont constamment comparés. Le repérage à l'estime

permet au système automatique de se passer provisoirement du GPS.

L'OBU détermine le montant du péage à partir des données du trajet et de la taxe ainsi que des données du véhicule saisies et enregistrées. Il mémorise cette information et la transmet par téléphonie mobile (GSM) à la centrale de collecte du péage (figure 3). Cette opération est toujours effectuée lorsque le péage a atteint un certain montant ou après un certain laps de temps.

300 po(i)nts de contrôle

Depuis un certain temps, des constructions en forme de pont évoquant la surveillance par radar et caméras surplombent les autoroutes allemandes. Tout véhicule s'approchant d'un tel pont de contrôle est identifié de loin par une unité de détection et de repérage (Tracking) DSRC (Dedicated Short Range Communications) mise au point par DaimlerChrysler.

La voie où se trouve le véhicule et l'instant précis de son passage sous le pont sont déterminés immédiatement pour permettre sa classification exacte. Le véhicule repéré est redevable d'un péage si son poids total admissible atteint ou dépasse 12 tonnes. Ce critère est déterminé par scrutation des 3 dimensions de chaque véhicule. Une technique d'évaluation des contours du véhicule indique s'il est soumis au péage. Dans le cas contraire, l'exploitant s'engage à détruire immédiatement toutes les données.

Le pont vérifie par une liaison infrarouge à faible portée (DSRC) si le véhicule participe au système automatique au moyen d'un appareil embarqué. Si un véhicule soumis au péage n'émet pas de signal infrarouge, on se trouve en présence d'un fraudeur ou d'un cas d'enregistrement manuel. Pour lever le doute, l'immatriculation de chaque véhicule est photographiée par une caméra infrarouge et comparée immédiatement aux numéros enregistrés par la centrale de Toll Collect. Un enregistrement manuel comporte en effet le numéro de plaque d'immatriculation. Les données sont détruites si l'enregistrement manuel a bien été effectué. Les données ne sont conservées qu'en cas de fraude. Elles sont transmises à l'Office fédéral pour la circulation des marchandises (BundesAmt für Güterver-

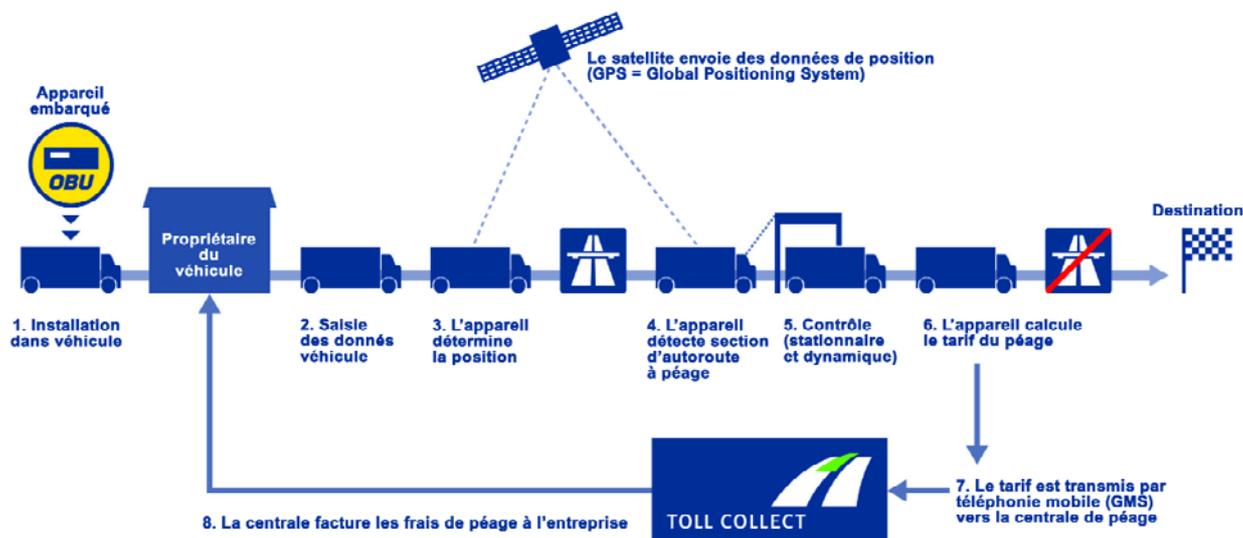


Figure 3. Déroulement de l'enregistrement automatique. Le poids lourd doit être équipé d'un appareil OBU. (Figure : Toll Collect)



Font déjà partie de la vie quotidienne sur l'autoroute : ponts de contrôle avec détecteurs destinés à saisir les données de péage. (Cliché : Toll Collect)

kehr, BAG) qui infligera une contravention au fraudeur.

Contrôleurs

Les véhicules dont les conducteurs sont suspectés de fraude peuvent être arrêtés immédiatement après un pont de contrôle (s'applique à la moitié des ponts environ) par des membres de l'Office fédéral pour la circulation des marchandises (BAG). L'équipement de ces ponts de contrôle comporte aussi une unité émettrice-réceptrice. Elle permet de communiquer avec les agents de l'office BAG effectuant les contrôles sur une aire de stationnement ou à une sortie suivant immédiatement le pont. Cela se passe quelques secondes seulement après que le poids lourd a franchi le pont de contrôle.

Des équipes volantes de l'office BAG disposant d'environ 300 véhicules contrôlent le versement du péage 24 heures sur 24. Ces véhicules, qui participent à la circulation générale, complètent les contrôles des ponts. Un contrôle global et flexible sur chaque tronçon du réseau autoroutier est donc assuré. Le personnel de contrôle détermine d'abord si le véhicule est soumis au péage. Si tel est le cas, il vérifie l'intégration au système automatique et l'indication correcte du nombre d'essieux. Le poids lourd n'est pas intégré au système automatique ? L'équipe fait alors appel à un PC de contrôle pour comparer l'immatriculation du véhicule aux données d'enregistrement manuelles de la base de données centrale. Le véhicule est arrêté si un doute existe sur le versement du péage.

Si le soupçon d'un délit de fraude se confirme, le personnel de contrôle inflige immédiatement une contravention au conducteur du véhicule. En résumé, les poids lourds sont soumis au contrôle automatique et mobile pendant leur trajet. La circulation n'est pas entravée. Le système de contrôle vérifie que le péage a été acquitté. Il ne sert ni à saisir ni à calculer le péage.

La position de chaque poids lourd sur les quelque 12 000 kilomètres d'autoroute formant environ 5 200 tronçons est déterminée et cataloguée. Le système fonctionne encore dans une version simplifiée. Les poids lourds seront équipés l'an prochain de la variante « dynamique » 2.0 de l'appareil OBU. Il devrait être alors possible de prélever un péage routier (*Road Pricing*) pour le détour par des routes nationales destiné à éviter... le péage autoroutier.

(045121-1)

Mac mini

Unix-in-a-Box

Dr. Thomas Scherer

Apple vient de réaliser une « compression » du Mac telle que cela a donné une version subminiaturisée incroyablement petite qui n'en est pas moins un vrai ordinateur. L'important est de savoir ce que permet de faire ce véritable « micro »-ordinateur et quels sont les domaines d'applications qui sortent des capacités du Mini. Cet émulateur matériel pour Mac OS X est en effet proposé à un prix tel que même des inconditionnels du PC pourraient être tentés de l'acheter...

Ce qui pourrait, à première vue, sembler être un porte-pot de fleur ou un mini-grille-pain au look design, se révèle être, en fait, un PC complet au format 16, 5 x 16, 5 x 5 cm (l x P x H). Soit à peine plus que le volume de 5 coffrets de CD empilés. Le poids : 1,3 kg, pas plus. En ce qui concerne le grille-pain, le Mac mini n'a rien d'une chauffe-rette, même à plein régime il ne chauffe pas. En fonctionnement normal, en l'entend à peine, en mode standby il est silencieux.

Au dehors ?

Sur le dos (figure 2) on découvre un certain nombre d'interfaces. La position de l'interrupteur en particulier n'est pas très pratique. Une série d'évents sert de sortie au ventilateur qui aspire de l'air sur le dessous, à gauche, à droite et sur le devant. Partant il est important de ne pas placer le mini Mac sur un support mou qui pourrait empêcher la circulation de l'air.

Le coffret est étonnant non seulement en raison de sa taille très compacte mais par son aspect plaisant et bien fini. Il comporte 2 parties : une structure en alu de 4 mm d'épaisseur, avec un couvercle de plastique blanc et une carcasse de support comportant les interfaces sur laquelle est montée la totalité de l'électronique (figure 3).

Les 2 parties sont reliées l'une à l'autre par nombre de crochets en plastique fixés sur le fond du coffret et qui viennent s'encaster dans des encoches de la structure en alu. Rien ne sert d'utiliser un tournevis pour tenter d'ouvrir le coffret. Apple n'a pas conçu le mini-Mac pour les *case-modder* ou les personnes tenant à expérimenter, mais comme une solution toute faite, il suffit de brancher la prise et c'est parti, une sorte de système de chargement pour iPod ou d'outil multimédia pour une utilisation domestique. L'appareil répond parfaitement à ce cahier des charges mais c'est tout. Dans ces conditions, toute modification est du ressort du revendeur mais il vaut mieux définir à la commande la configuration souhaitée, RAM, graveur de DVD, disque dur, intégration de WLAN et Bluetooth...

Ceci n'a pas de quoi effrayer un lecteur d'Elektor. Il est en effet possible d'économiser quelques euros en ajoutant soi-même de la RAM. L'ouverture du coffret requiert une sorte de spatule à lame fine. On trouvera des informations plus détaillées sur Internet.

Et à l'intérieur ?

Comme le montre la figure 3, ce n'est pas de l'air en boîte que vend Apple. Les composants de cet ordinateur ultra-compact



Figure 1. Mac mini vu du devant. Un design alu typiquement Apple avec une LED de contrôle, une fente pour média optiques et le logo d'Apple.

son, le cas échéant, du type *notebook*. L'alimentation externe (1/4 du volume du Mac mini approximativement) fournit bien jusqu'à 85 W, mais en fonctionnement normal il n'en consomme pas même, (mesuré !) 20 W ! On se trouve ici en présence d'un ordinateur très silencieux et bien « vert ».

La CPU de chez Freescale Semiconductor (ex-Motorola) n'y est pas pour peu. Sa consommation à pleine charge n'est que de 18 W. Comparée aux horloges d'Intel, la fréquence de 1,25 GHz peut paraître modeste. Mais c'est trompeur : La CPU possède, par exemple, un pipeline étonnamment court de 7 étages. Ceci la rend, à l'image des CPU du type Pentium-M, très efficace si on la compare aux versions actuelles de la P4 avec leur pipeline à 31 étages. À cela s'ajoute l'unité SIMD intégrée baptisée Altivec, qu'il est difficile de battre lorsqu'il s'agit du traitement de données multimédia. D'autre part, un concept appelé Quartz-Extreme accélère le tout en mettant autant que possible le processeur graphique à contribution pour la construction du GUI (l'image écran). On peut dire, grossièrement, que la puissance, étalonnée à la AMD, se situe, en fonction de l'application, entre 2,2 et 2,8 GHz en « unités normées à la P4 ».

On ne se trouve pas aux premiers rangs, mais l'industrie est incapable de coincer plus dans un coffret aussi petit.

À cela s'ajoute, outre *iLife*, un set comportant les programmes *iPhoto* (optimisation d'images et base de données), *iMovieHD* (montage vidéo au format HD), *iDVD* (création et gravure de ses propres films vidéos), *Garageband* (création et traitement de musique très universel), sans oublier *iTunes* gratuit (téléchargement et traitement de fichiers musique de formats MP3 et AAC) ni bien entendu *AppleWorks* (traitement de texte, tableurs et dessin vectorisé).

Quoi d'autre ?

Le prix demandé par Apple est étonnant, compte tenu des coûts du système d'exploitation (€ 149) et de celui du set multimédia *iLife* (€ 79), car cela implique qu'Apple propose le matériel de la version 1,25 GHz au prix incroyable de € 261 ! Pas question de proposer, dans ce cadre financier, de clavier ou de souris. Mais comme il est possible d'utiliser tout clavier et toute souris du type USB. Il est préférable, pour le clavier uniquement, de préférer un modèle Apple en raison d'une disposition quelque peu spécifique des touches.

Outre une utilisation en bureautique personnelle, ou en



Figure 2. Mac mini vu de l'arrière. De la gauche vers la droite : interrupteur et alimentation, embase réseau, embase modem, embase DVI-I, 2 embases USB 2.0, une embase Firewire à 6 contacts, sortie audio 3,5 mm et orifice allongé pour verrou Kensington.



Figure 3. Mac min ouvert. On voit sur le dessus le lecteur optique ultra-plat. Il est superposé à un disque dur 2,5". On voit une partie du ventilateur de CPU noir. En haut à droite on trouve la mémoire : un classique module DDR-DIMM du type PC2700 à 184 contacts.

machine Internet stable et sans virus (il n'existe pas, pour le moment, de virus pour OS X), on peut penser à d'autres applications : système *Home-Entertainment* chic et silencieux (on trouve sur Internet des solutions en *Media-Center* très attrayantes) voire petit serveur à tout faire : comme le système repose sur Unix à base du kernel BSD, on dispose, avec Apache, SMB, etc. tous les programmes et protocoles d'excellente facture requis pour faire tourner un petit serveur Web, FTP, AFP et

fichiers. À l'aide du « *Apple Remote Desktop* » ou VNC, l'administration peut se faire sans souris ni clavier. Face à un système Mini-ITX à base de CPU Epia de VIA plus encombrant, le Mac-mini offre plus de confort et de puissance, tout en coûtant moins cher, sans parler du design.

(050033-1)

Liens :

www.apple.com/fr/
www.apple.com/fr/hotnews

Caractéristiques techniques & prix :

- CPU MPC 7447 alias G4 PowerPC ; 512 Koctets de cache L2 ; 1,25 ou 1,42 GHz
- 256 Moctets de RAM DDR PC2700; jusqu'à 1 Goctet max.
- Lecteur/graveur DVD±R/CD-RW Combo ou DVD±RW
- Disque dur IDE 2,5" de 40 ou 80 Goctets
- Interfaces Firewire ; 2 X USB 2.0, Modem ; Audio-Out; DVI-I (numérique et analogique via adaptateur)
- Software OS X 10.3.7; MacOS Classic; iLife-05-Paket
- En outre : bloc d'alimentation externe ; ni clavier ni souris
- € 489 pour 1,25 GHz/40 GB; € 589 pour 1,42 GHz/80 GB;
- DVD±RW 4X € 100 de plus; WLAN/Bluetooth € 100 de plus

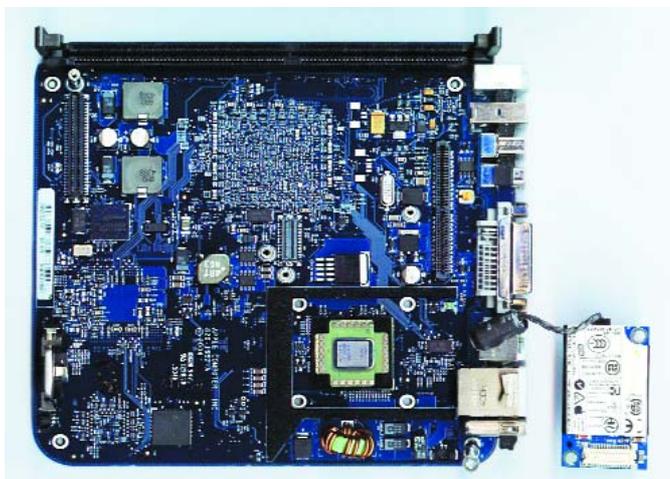


Figure 4. Carte-mère vue du dessus. Le connecteur de RAM occupe ici le bord supérieur. À gauche le connecteur pour la carte WLAN/Bluetooth optionnelle. À droite l'interface IDE. La CPU sans ventilateur se trouve en bas à droite (Photo : Dr. Thomas Scherer).

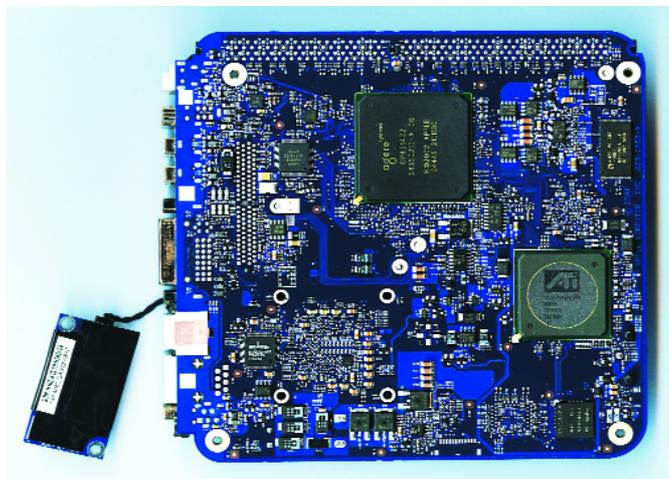


Figure 5. Carte-mère vue du dessous. La grosse puce en haut est d'origine Apple; elle remplit les fonction de la puce North/South-Bridge d'un PC. À droite, on voit la GPU à base d'ATI 9200 dotée de 32 Moctets de mémoire graphique propre. (Photo : Dr. Thomas Scherer).

LICIUM **Comment choisir son microcontrôleur**

Lorsque l'auteur de ces lignes a commencé à travailler avec ces circuits, il y a de cela près de 25 ans, répondre à une telle question était très facile. Il n'y avait en effet que deux ou trois fabricants actifs en ce domaine et chacun ne proposait que deux ou trois circuits. Il était donc facile de les comparer et de voir celui qui convenait le mieux (ou plutôt, compte tenu des performances de l'époque, celui qui convenait le moins mal !).

Aujourd'hui, des dizaines de fabricants proposent une multitude de circuits, répartis au sein de familles distinctes qui sont parfois concurrentes chez un même fabricant. Il est donc impossible de procéder par comparaison directe des possibilités des différents circuits car cela nécessiterait la consultation de plusieurs milliers de fiches techniques.

Il faut donc procéder de façon différente, et cet article est là pour vous aider en vous montrant quelles sont les questions à se poser, quelles sont les réponses que l'on peut en attendre et, à terme, quelles sont les familles, voire les circuits qui seront le mieux adaptés à vos besoins. Mais, ne nous leurrions pas ; nous n'allons pas vous donner la recette miracle du style : tel microprocesseur convient à telle application, pour la simple et bonne raison que c'est impossible. En effet, et c'est ce qui fait la force de ces circuits, tous peuvent presque tout faire et ce ne sont donc pas seulement les performances pures d'un microcontrôleur qui feront qu'une application sera réussie ou non mais également la façon dont il aura été mis en œuvre.

Un sondage

Selon une récente étude américaine, réalisée auprès des principaux développeurs d'applications à base de microcontrôleurs, leurs critères de choix se répartissent de la façon suivante :

- disponibilité des outils de développement logiciels : 67% ;
- prix : 51% ;
- richesses en entrées/sorties et périphériques internes : 40% ;
- disponibilité des outils de développement matériels : 35% ;
- et enfin
- compatibilité du code produit : 27%.

Bien sûr nous direz-vous, cela fait largement plus de 100%, mais ce sondage autorisait les réponses multiples ; réponses multiples qui montrent d'ailleurs bien les problèmes que pose un tel choix et la difficulté que l'on a pour définir un critère prépondérant.

Vos habitudes de travail

La première considération à prendre en compte concerne le fait que vous connaissiez déjà une famille de circuits. Si tel est le cas, et sauf si cette dernière ne vous donne vraiment pas satisfaction, vous aurez tout intérêt à vous orienter vers des microcontrôleurs de la même famille. En effet, en procédant ainsi vous connaîtrez déjà les « habitudes » du fabricant et vous n'aurez pas tout à redécouvrir : jeux d'instructions, modes d'adressage, noms des registres, méthode de traitement des entrées/sorties, etc. En outre, si vous avez déjà investi dans différents outils de développement tels que : environnement de développement, programmeur et à plus forte raison émulateur, vous pourrez très certainement les utiliser, quitte à les faire évoluer (à les « upgrader » comme l'on dit en paraphrasant nos amis anglo-saxons).

Par contre, si vous n'avez aucune connaissance préalable de la moindre famille de circuits, vous êtes totalement libre et pouvez donc aborder les critères suivants.

Langage machine ou langage évolué

Paradoxalement, et alors que cette question est purement logicielle, sa réponse est directement conditionnée par le matériel ou, selon le point de vue que l'on adopte, conditionne directement le choix de ce dernier.

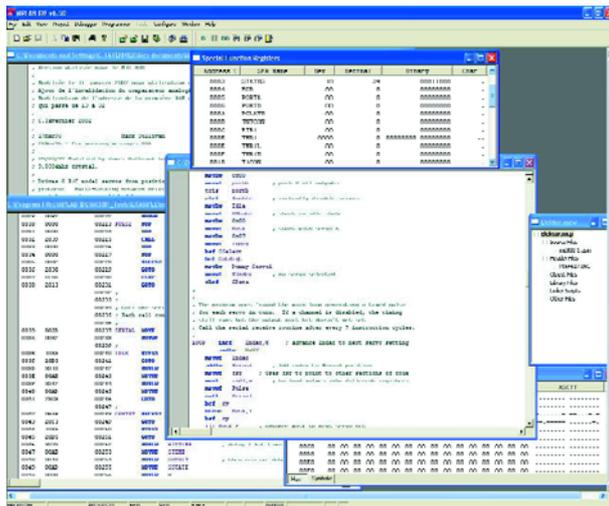
Il y a encore quelques années, cette question ne se posait même pas car la taille de la mémoire de programme des microcontrôleurs était tellement faible qu'il n'était possible de les programmer qu'en langage machine (en assembleur pour adopter un raccourci qui, même s'il est incorrect, est largement utilisé). Il faut savoir en effet que, lorsque l'on programme en langage évolué, ce dernier est traduit en langage machine par le compilateur et que chaque ligne de langage évolué donne naissance à de nombreuses lignes de langage machine. Le recours au langage évolué impose donc de disposer de microcontrôleurs ayant une mémoire de programme de capacité importante.

Même si les compilateurs actuels ont fait d'énormes progrès et sont capables d'optimiser le code produit, ils n'atteignent pas les performances du langage machine, au moins pour ce qui est des microcontrôleurs dont la principale occupation est de manipuler des bits dans des registres ; opérations pour laquelle les compilateurs de langage évolué sont très mal adaptés. Ainsi par exemple :

```
output_D(i);
```

qui a pour but de faire sortir la valeur de l'entier *i* codé sur 8 bits via le port D d'un microcontrôleur, est-il traduit de la façon suivante par un des meilleurs compilateurs C pour microcontrôleurs PIC actuels :

Figure 1. Malgré sa gratuité, MPLAB est un véritable environnement de développement avec éditeur, assembleur et même simulateur ; le tout étant parfaitement interactif.



```

MOVLW 00
BSF 03.5
MOVWF 08
BCF 03.5
MOVF 20,W
MOVWF 08
    
```

Un programmeur travaillant en langage machine ferait la même chose en deux instructions seulement ! Le recours à un compilateur a donc, dans cet exemple précis et réel, multiplié l'occupation de la mémoire de programme par trois par rapport au langage machine.

Ceci étant, si votre application doit réaliser des calculs faisant intervenir des nombres décimaux ou représentés en notation scientifique, ou bien encore si elle doit recourir à des fonctions mathématiques complexes telles que les fonctions trigonométriques par exemple, le recours au langage évolué est alors quasiment inévitable. Tous ceux d'entre-vous qui ont déjà calculé un cosinus en langage machine en sont sans doute déjà convaincus !

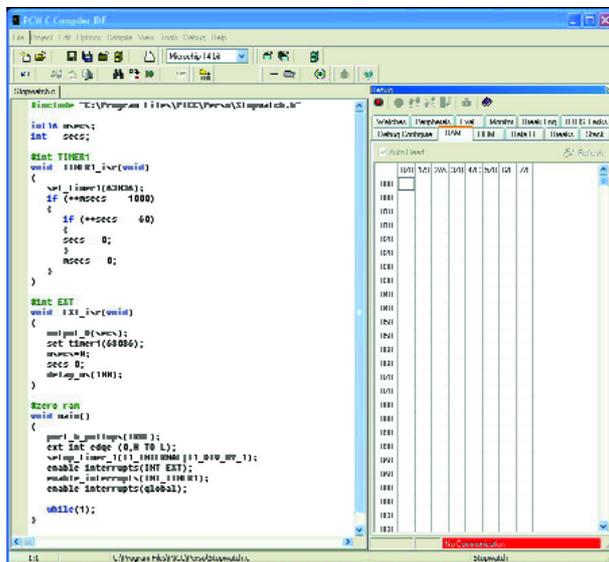
Le choix du langage de travail dépendra donc avant tout des traitements logiciels réalisés par votre application : si elle ne fait que de la manipulation d'entrées/sorties et des calculs simples, le langage machine est le mieux adapté.

Si elle réalise de nombreux calculs ou fait appel à des fonctions mathématiques complexes, le langage évolué devra être choisi.

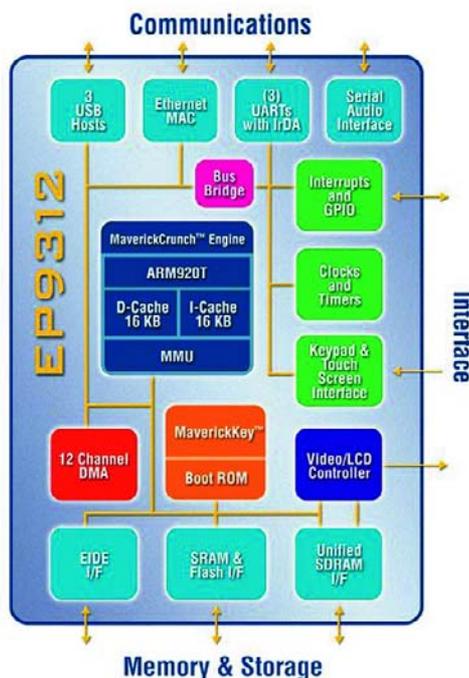
Cette méthode de choix admet une exception qui est relative à la connaissance préalable que vous pouvez avoir de tel ou tel langage de programmation. En effet, si vous maîtrisez un langage évolué vous aurez intérêt à l'utiliser, même s'il n'est pas le mieux adapté, car cela vous évitera de devoir apprendre à programmer en langage machine, sauf bien sûr si cela devait conduire à des pertes de performances rédhibitoires de votre application. Avant d'en terminer avec cette question, sachez que, à de rares exceptions près, les seuls langages évolués actuellement disponibles pour les microcontrôleurs sont le Basic et le C.

Ceci étant, le choix du langage de programmation ne nous a pas encore permis de choisir telle ou telle famille de microcontrôleur mais nous savons déjà que si nous voulons travailler en langage évolué, il nous faudra une mémoire de programme de taille relativement importante. C'est un premier élément.

Figure 2 : L'environnement de développement PCWH du compilateur C pour microcontrôleurs PIC de CCS est également un modèle d'intégration réussie.



Comme le montre ce synoptique du EP9307, les microcontrôleurs deviennent de plus en plus complexes. (source Cirrus Logic)



Les outils de développement

Sous ce vocable on regroupe habituellement un certain nombre de programmes et d'interfaces matérielles avec :

- un éditeur de texte pour l'écriture des programmes ;
- un assembleur si l'on travaille en langage machine ou un compilateur si l'on travaille en langage évolué ; la présence du compilateur impliquant quant à elle toujours la présence d'un assembleur associé ;
- un simulateur, qui est une solution purement logiciel permettant de simuler le programme de l'application afin de le tester ;
- un émulateur, qui est une solution mixte matériel-logiciel permettant de tester le programme alors même qu'il est déjà implanté dans l'application ;
- un programmeur, qui permet de programmer le microcontrôleur avec le programme que l'on a développé.

Hormis le simulateur et l'émulateur dont la présence est facultative, les autres éléments sont évidemment indispen-

Ne programmez plus, « bootloadez » !

Avec l'arrivée sur le marché de microcontrôleurs dont la mémoire de programme est de taille importante, une nouvelle méthode de travail est en train de se développer et fait appel à ce que l'on appelle un « bootloader » (désolé, il n'y a aucun mot français qui convienne pour le remplacer).

Qu'est ce qu'un « bootloader » ? C'est est un programme, le plus court possible, programmé une fois pour toutes en mémoire haute ou en mémoire basse du microcontrôleur monté sur l'application dont le programme est en cours de développement. Ce programme est capable de communiquer, généralement au moyen d'un port série (mais d'autres approches existent via l'USB, l'I2C ou même le bus CAN), avec l'outil de développement utilisé pour écrire le programme de l'application en cours de développement.

Ce programme est en outre capable d'interpréter un certain nombre de commandes dont celles de lecture, de programmation et d'effacement de la mémoire de programme du circuit dans lequel il se trouve implanté. Dès lors, son principe d'utilisation est facile à comprendre.

Lorsque le programme en cours de développement est prêt à être essayé, la communication avec le bootloader est établie et celui-ci télécharge et programme ce dernier dans la mémoire de programme du microcontrôleur sans qu'il soit nécessaire de faire appel au moindre programmeur externe.

Une fois cette opération terminée, le bootloader transfère le contrôle au programme ainsi téléchargé que l'on peut alors exécuter et tester étant entendu que ces enchaînements d'opé-

rations (téléchargement puis transfert d'exécution) peuvent être réalisés autant de fois que nécessaire et, en tout état de cause, jusqu'à ce que le programme en cours de développement soit au point.

Le gain en vitesse d'exécution de ces opérations, par rapport à une programmation classique, atteint à peu près un rapport de dix ce qui se ressent évidemment au niveau de la vitesse et du confort de développement de l'application.

À la lecture de ce court exposé, vous aurez compris que l'utilisation d'un bootloader impose tout de même un certain nombre de contraintes au niveau du microcontrôleur choisi. Il faut en effet :

qu'il dispose d'une mémoire de programme de taille suffisante pour y loger le bootloader en plus du programme de l'application en cours de développement.

Qu'il supporte l'effacement et la programmation « en interne » de sa mémoire de programme.

Qu'il dispose d'un port série, ou USB ou CAN dans le cas des autres approches évoquées ci-dessus, et que ce dernier soit accessible au bootloader sans perturber l'application.

Sous réserve que ces contraintes soient respectées, le recours à un « bootloader » est aujourd'hui très facile car de nombreux programmes de ce type sont mis à votre disposition gratuitement sur Internet pour les différentes familles de microcontrôleurs qui supportent ce mode de travail.

sables. Ils sont très souvent regroupés aujourd'hui dans ce que l'on appelle un environnement de développement ou IDE (acronyme de *Integrated Development Environment*) qui, sous Windows, se présente comme un logiciel à fenêtre offrant un grand confort d'emploi en permettant de passer immédiatement de la fenêtre de l'éditeur à celle de l'assembleur ou du compilateur, voire même à celle du simulateur ou de l'émulateur s'ils sont intégrés. Les recopies d'écran qui illustrent cet article vous montrent quelques exemples avec l'excellent MPLAB de Microchip (**figure 1**) ou bien encore l'environnement PCWH pour le compilateur C de CCS (**figure 2**).

Si l'on excepte l'émulateur et le programmeur, qui nécessitent du matériel et sont donc nécessairement payants, les autres outils ne sont constitués que par du logiciel. Certains fabricants de microcontrôleurs ont donc fait le pari commercial de donner ces outils. Pour surprenante qu'elle puisse paraître, cette façon de faire est en fait très logique car elle incite ainsi les développeurs à y faire appel et donc à utiliser les microcontrôleurs du fabricant concerné.

Malgré cette gratuité, ces outils de développement ne sont pas des produits de seconde zone, et loin s'en faut, et si nous devons les classer nous mettrions sans hésiter en tête MPLAB de Microchip. En effet, outre le fait que cet environnement soit complet pour travailler en langage machine (éditeur, assembleur et simulateur) il s'interface aussi quasi automatiquement avec des compilateurs et émulateurs tiers, c'est-à-dire non produits par Microchip, et les intègre à l'environnement comme s'ils en avaient toujours fait partie.

Cette gratuité des outils de développement ne va toutefois pas au-delà du travail en langage machine et, chez quasiment tous les fabricants actuels, les compilateurs pour langage évolué sont payants. Ceci étant, un certain

nombre d'éditeurs de compilateurs (Basic ou C) proposent souvent des versions de démonstration fonctionnelles qui peuvent donner satisfaction à un amateur et peuvent permettre à un professionnel d'essayer le produit avant d'investir.

Le tableau de synthèse que nous vous proposons en guise de conclusion à cet article, mais que vous pouvez dès à présent consulter, dresse la liste de tous les environnements de développement gratuits proposés par les fabricants de microcontrôleurs. N'hésitez pas les télécharger afin de les essayer puisque c'est possible même sans avoir le moindre circuit sous la main. Vous pourrez alors apprécier leur ergonomie. Si vous voulez ou devez travailler en langage évolué, vérifiez bien l'intégration du compilateur à ces outils, d'où l'intérêt, là encore, des versions de démonstration.

Reste le problème du programmeur puisqu'il vous faudra bien, un jour ou l'autre, programmer votre micro. Sachez donc qu'aujourd'hui la tendance est à la programmation en circuit appelée ISP (*In System Programming*) ou ICSP (*In Circuit Serial Programming*). Cette façon de faire permet de programmer le circuit alors qu'il est déjà en place sur l'application au moyen d'une liaison ne comportant que trois ou quatre signaux actifs. Le programmeur nécessaire s'en trouve donc très simplifié et Elektor vous a déjà proposé à de nombreuses reprises des réalisations en ce domaine pour les familles de microcontrôleurs les plus diverses. La disponibilité, les prix et/ou la facilité de réalisation de ce programmeur seront donc des éléments à prendre en compte pour le choix de votre famille de microcontrôleurs. Une autre tendance est actuellement en train de se faire jour, c'est le recours à ce que l'on appelle un « *bootloader* », au moins en phase de développement de pro-

Fabricant	Site Internet	Taille	Famille	Architecture	Noyau	IDE gratuit
Analog Device	www.analog.com	8 bits	ADUC8xx	CISC	8051	–
		32 bits	ADUC7xx	RISC	ARM7	–
Atmel	www.atmel.com	8 bits	AT89xxx	CISC	8051	Programmes indépendants
		8 bits	TS87xxx	CISC	–	Programmes indépendants
		8 bits	AVR	RISC	–	AVR studio
		16 bits	AT91xxx	RISC	ARM7/9	–
Cirrus Logic	www.cirrus.com	32 bits	EP73xxx	RISC	ARM7	–
		32 bits	EP93xxx	RISC	ARM9	–
Cygnal	www.silabs.com	8 bits	C8051F	CISC	8051	–
Freescale (ex. Motorola)	www.freescale.com	8 bits	HC05	CISC	6800	–
		8 bits	HC08	CISC	6809	Code Warrior
		8 bits	HC11	CISC	6809	–
		16 bits	HC12	CISC	–	–
		16 bits	HCS12	CISC	–	Code Warrior
		16 bits	HC16	CISC	–	–
		16 bits	56800	CISC	–	–
		32 bits	68K	CISC	68000	–
		32 bits	ColdFire	CISC	–	–
		32 bits	MAC7100	RISC	ARM7	–
Fujitsu	www.fujitsu.com	8 bits	F2MC-8	CISC	–	–
		16 bits	F2MC-16	CISC	–	–
		32 bits	FR	RISC	–	–
Infineon	www.infineon.com	8 bits	C5xxx, C8xxx	CISC	8051	–
		16 bits	C16xxx	CISC	–	–
		16 bits	XC16xxx	CISC	–	–
		32 bits	TCxxx	CISC	–	–
Intel	www.intel.com	8 bits	MCS251	CISC	8051	–
		16 bits	MCS96/296	CISC	–	–
Maxim (Dallas)	www.maxim-ic.com	8 bits	DS80Cxxx	CISC	8051	–
		8 bits	DS83Cxxx	CISC	8051	–
		8 bits	DS89Cxxx	CISC	8051	–
		16 bits	MAXQ	RISC	–	–
Microchip	www.microchip.com	8 bits	PIC 10,12,16,18	RISC	–	MPLAB
		16 bits	dsPIC	RISC	–	MPLAB

Tableau de synthèse : Les différentes familles des principaux fabricants de microcontrôleurs. La colonne « noyau » indique sur quelle architecture ou sur quelle unité centrale est bâti le circuit. La colonne « IDE gratuit » donne le nom de l'environnement de développement, téléchargeable sur le site du fabricant concerné.

gramme (cf. l'encadré consacré à ce sujet). Si vous ne savez pas encore ce que c'est, consultez l'encadré que nous avons rédigé à ce sujet, mais sachez dès à présent que cette technique n'est utilisable qu'avec les circuits qui sont capables de programmer eux-mêmes leur mémoire

de programme. Les fabricants y viennent peu à peu, mais tous n'ont pas encore sauté le pas.

Pour conclure cette partie relative aux outils de développement on peut dire que, si vous êtes un amateur ou si vous travaillez dans une petite entreprise qui ne peut investir des sommes trop importantes ; la disponibilité d'un programmeur facile à réaliser et peu coûteux associé à un outil de développement gratuit seront des éléments déterminants pour le choix de votre famille de circuits (**figure 3**). Une fois encore, notre tableau de synthèse devrait vous aider en ce domaine.

8, 16 ou 32 bits ?

À l'instar des microprocesseurs, dont ils reprennent parfois tout ou partie des unités centrales, les microcontrôleurs ne cessent de grossir et, alors que les modèles 4 bits ont quasiment disparu ; on trouve aujourd'hui sur le marché des circuits manipulant des mots de données sur 8, 16 voire même 32 bits. Il ne faudrait cependant pas en conclure que les circuits les plus puissants sont nécessairement les modèles 32 bits et que vous êtes « ringard » si vous utilisez un microcontrôleur 8 bits. En effet, dès lors que votre application ne nécessite que la manipulation d'entrées/sorties simples et ne fait pas appel à des calculs complexes sur des données de

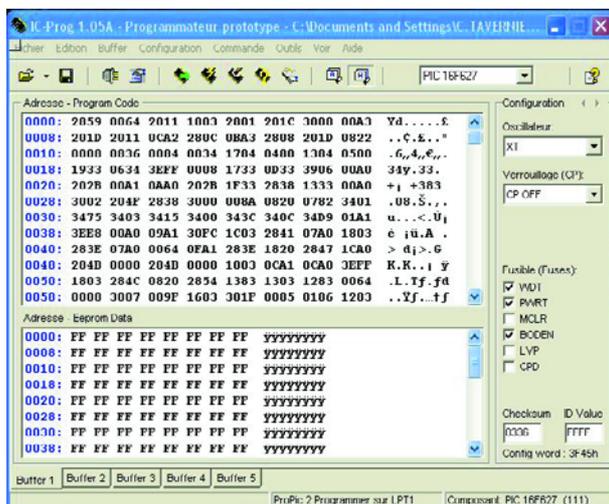


Figure 3. Le « freeware » bien connu IcProg (www.ic-prog.com) permet de programmer de nombreux microcontrôleurs avec des programmeurs aux schémas fort simples et plusieurs fois décrits dans Elektor.

Fabricant	Site Internet	Taille	Famille	Architecture	Noyau	IDE gratuit
NS	www.national.com	8 bits	COP8xxx	CISC	—	Webench
		16 bits	CR16Cxxx	CISC	—	—
		16 bits	CP3000	RISC	—	—
Philips	www.semiconductors.philips.com	8 bits	P8xxx	CISC	8051	—
		16 bits	Xaxxx	CISC	—	—
		32 bits	LPC2000	RISC	ARM7	—
Rabbit Semiconductor	www.rabbitsemiconductor.com	8 bits	Rabbit2000	CISC	—	—
		8 bits	Rabbit3000	CISC	—	—
Renesas	www.renesas.com	8 bits	740	CISC	—	—
		16 bits	H8	CISC	—	HEW
		16 bits	H8S	CISC	—	HEW
		16 bits	M16C	CISC	—	—
		16 bits	7700	CISC	—	—
		32 bits	H8SX	CISC	—	—
		32 bits	Super H	CISC	—	HEW
ST	www.stm.com	8 bits	ST5	CISC	—	Visual FIVE
		8 bits	ST6	CISC	—	—
		8 bits	ST7	CISC	—	STVD 7
		8 bits	ST9	CISC	—	STVD 9
		16 bits	ST9	CISC	—	STVD 9
		16 bits	ST10	CISC	—	—
		32 bits	ARM7	RISC	ARM7	—
Texas Instruments	www.ti.com	8 bits	MSC12xxx	CISC	8051	—
		16 bits	MSP430	CISC	—	Eclipse
		32 bits	TMS470	RISC	ARM7	—
Toshiba	chips.toshiba.com	8 bits	870	CISC	—	—
		16 bits	900/900H	CISC	—	—
		32 bits	900/900H	CISC	—	—
Uicom (ex. Scenix)	www.uicom.com	8 bits	SXxx	RISC	—	—
Zilog	www.zilog.com	8 bits	Z8xxx	CISC	Z80	—
		8 bits	Z8Encore !	CISC	Z80	—
		8 bits	eZ80Aclaim	CISC	Z80	—

grande taille, un microcontrôleur 8 bits est mieux adapté qu'un modèle 16 ou 32 bits. Cet apparent paradoxe est facile à comprendre au moyen de l'exemple que voici. Si votre application exploite une liaison série RS-232, elle doit nécessairement manipuler des mots de 8 bits puisque les caractères habituellement transmis sur une telle liaison le sont en code ASCII qui n'occupe, comme on le sait, que 7 ou 8 bits (selon qu'il est standard ou étendu). Un microcontrôleur 32 bits est donc handicapé dans une telle situation car, pour traiter les caractères reçus, il lui faut les manipuler dans ses mots de 32 bits dont 24 bits restent inutiles et doivent être masqués.

Par contre bien entendu, si votre application doit réaliser du calcul mathématique ou scientifique, les microcontrôleurs 16 voire 32 bits s'avèreront être les plus performants car ils permettront de manipuler des nombres codés en virgule flottante beaucoup plus facilement que les circuits 8 bits.

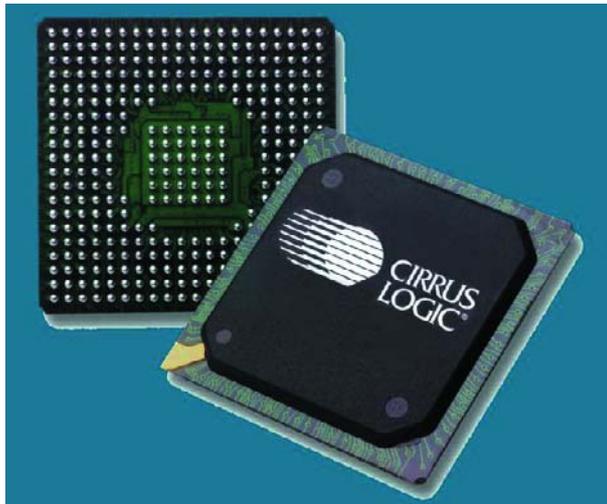
Pour couper la poire en deux on aurait donc tendance à choisir des microcontrôleurs 16 bits. Ce serait à notre avis une erreur compte tenu de la conjoncture actuelle. En effet les microcontrôleurs 32 bits sont de plus en plus présents sur le marché à des prix abordables alors que les microcontrôleurs 8 bits sont de plus en plus performants et de moins en moins coûteux. Les 16 bits sont donc quelque peu coincés entre les deux et faire appel à

eux ne se justifie plus vraiment. En effet, quitte à abandonner un microcontrôleur 8 bits parce qu'il n'est pas assez puissant pour une application, mieux vaut alors passer directement à du 32 bits qui sera aujourd'hui à peine plus coûteux que du 16 bits.



Infineon propose une palette de plus en plus étoffée de microcontrôleurs 32 bits tels que les TC1912 et TC1910 destinés à l'audio. (source Infineon)

Figure 4.
Il n'est bien évidemment plus question de souder ce type de composants qui requièrent un support spécial.
(source Cirrus Logic)



ROM, OTPROM, EEPROM, Flash ?

Tous ces sigles qualifient la mémoire de programme, et parfois une zone de données spécifique, du microcontrôleur. Il va de soi que pour un usage unitaire, les circuits avec de la mémoire EEPROM ou Flash devront être choisis puisque ces mémoires sont les seules que l'on puisse programmer et effacer électriquement, jusqu'à plusieurs milliers de fois si nécessaire.

Pour une production en petite série, et une fois que la maquette aura été mise au point avec un circuit à mémoire EEPROM ou Flash, il pourra s'avérer intéressant financièrement de faire appel à des circuits à mémoire OTPROM, c'est-à-dire à mémoire programmable une seule fois. Rappelons en effet que de tels circuits se programment avec les mêmes outils que les microcontrôleurs à mémoire Flash ou EEPROM mais ne peuvent ensuite plus s'effacer. Chez certains fabricants, ils sont moins coûteux, à référence identique, que leurs homologues à mémoire Flash ou EEPROM que l'on réservera alors pour la phase de développement.

Attention !

Tous les fabricants ne produisent pas tous leurs circuits dans les différentes technologies de mémoires existantes. Il est donc essentiel de bien vérifier sur le site Internet du fabricant la disponibilité des types de mémoires que l'on souhaite. Notez aussi que certains microcontrôleurs, destinés essentiellement à de la production en très grande série, n'existent qu'avec de la ROM programmable par masque. Vous pouvez donc les oublier s'il ne vous faut pas plusieurs milliers de pièces identiques !

Entrées/sorties et périphériques internes

L'intérêt essentiel d'un microcontrôleur est de permettre de faire appel à un minimum de circuits externes, voire à pas de circuits externes du tout. Connaissant les besoins en ressources de votre application, il va donc falloir en dresser la liste en mettant en évidence les points suivants :

- nombre de lignes d'entrées/sorties parallèles ;
- nombre et taille (8, 16, 32 bits) des timers nécessaires ;
- présence d'un ou plusieurs ports PWM ;
- présence d'un convertisseur analogique/digital, avec quelle résolution et combien d'entrées ;
- présence d'un port série synchrone ou SPI ;

- présence d'interfaces pour bus spécialisés : CAN, I²C, USB, etc.
- présence d'interfaces spécifiques telles qu'interface pour afficheurs LCD sans décodeur intégré par exemple.

Une fois un tel tableau dressé, il ne reste plus qu'à le comparer aux tableaux de sélection des principaux fabricants de microcontrôleurs qui restent en lice après exploitation des critères que nous avons vus jusqu'à présent pour pouvoir procéder généralement à de sévères éliminations.

Vu la richesse des microcontrôleurs actuels, et sauf si vous avez des besoins démesurés, vous devriez trouver votre bonheur chez de multiples fabricants. Si vous avez besoin de nombreuses ressources présentes simultanément dans un même circuit, il se peut toutefois que cela vous conduise à choisir un « gros » microcontrôleur, trop cher pour votre application. N'oubliez pas alors qu'il est toujours possible de faire des compromis. Ainsi par exemple si vous avez besoin d'un convertisseur analogique/digital mais qu'il n'existe pas dans la famille qui vous conviendrait, peut-être est-il possible de faire appel à un modèle externe à interface I²C ou SPI. De même, si un port USB vous est nécessaire, vous constaterez vite qu'il n'est présent que sur les plus récents et les plus richement dotés des circuits. Il existe par contre d'excellents contrôleurs externes qui s'interfacent ensuite facilement avec le microcontrôleur.

Vitesse d'horloge, vitesse « tout court » et consommation

La fréquence d'horloge des microcontrôleurs ne cesse d'augmenter et, alors que les premiers circuits travaillaient tranquillement à 1 MHz, on trouve aujourd'hui des circuits dont l'horloge atteint 100 MHz. Dans une éternelle recherche de performances et surtout de rapidité, on est donc souvent tenté de choisir systématiquement le circuit ayant l'horloge à la fréquence la plus élevée. Même si, en principe, plus l'horloge va vite et plus le microcontrôleur est rapide ; ce paramètre doit être relativisé.

La fréquence d'horloge ne suffit pas en effet à qualifier la vitesse de fonctionnement d'un microprocesseur pour au moins deux raisons :

elle est très souvent divisée en interne et la fréquence réelle de fonctionnement n'est donc que rarement celle marquée sur le quartz qui la pilote.

Selon que le microcontrôleur est à architecture RISC ou CISC, il exécute toutes ses instructions en un cycle d'horloge (RISC) ou en plusieurs cycles (CISC) ; le terme plusieurs pouvant parfois signifier 10 ou plus.

Pour vous donner un exemple concret, sachez donc qu'un 8051 qui fonctionne à 30 MHz d'horloge est moins performant (à programme identique bien sûr) qu'un Atmel AVR qui ne fonctionne qu'à 10 MHz. Le premier a une architecture CISC et certaines instructions consomment jusqu'à 12 cycles machine alors que le second est à architecture RISC et exécute quasiment tout en un seul cycle. Sachez aussi que, plus la fréquence d'horloge est élevée, plus le risque de rayonnement de perturbations électromagnétiques est important et plus celles-ci sont difficiles à éliminer. Lorsque l'on sait qu'aujourd'hui tous les produits électroniques doivent vérifier la directive européenne de CEM, c'est un paramètre à ne pas négliger.

Enfin, plus un microcontrôleur est rapide et plus il consomme. Les « bêtes de course » seront donc à réserver aux applications dont l'autonomie ne pose pas de

problème. Bien sûr, tous les circuits sont aujourd'hui réalisés en technologie MOS mais cela ne suffit pas à assurer une faible consommation. Rappelons en effet qu'en technologie CMOS, les circuits ne consomment du courant que lorsqu'ils changent d'état. Et justement, plus l'horloge est rapide, plus ils changent d'état souvent !

Si la consommation est un facteur important de votre application, vous veillerez donc à ne pas exagérer sur la fréquence d'horloge mais vous veillerez aussi à choisir un circuit qui dispose d'un mode veille (*standby*) ou sommeil (*sleep*). Dans un tel mode, le microcontrôleur ne fait rien mais consomme très peu, parfois même jusqu'à mille fois moins qu'en fonctionnement normal. À quoi sert un circuit qui ne fait rien direz-vous ? Tout simplement à attendre qu'on le réveille pour qu'il fasse quelque chose. C'est le cas de très nombreuses applications qui dialoguent avec un être humain. Ainsi par exemple un thermostat programmable passe son temps à ne rien faire puisqu'il se contente de mesurer la température à des intervalles de temps de quelques minutes et d'attendre une éventuelle action de l'utilisateur sur son clavier de programmation. Il peut donc être en sommeil quasiment tout le temps (c'est d'ailleurs le cas de la majorité d'entre eux ce qui explique pourquoi ils fonctionnent plus d'un an avec une vulgaire pile alcaline).

Un paramètre que l'on oublie souvent

Ca y est, vous avez trouvé le circuit de vos rêves ; vous êtes prêt à dessiner le circuit imprimé et vous cherchez donc fébrilement dans la fiche technique le brochage du circuit que vous aviez délibérément ignoré jusqu'à présent ... et c'est là que les choses risquent de se gâter. En effet, si vous avez choisi un microcontrôleur récent et/ou puissant, ou bien encore très richement doté en ressources internes, vous pouvez généralement dire adieu au bon vieux boîtier DIL, jugé bien trop encombrant aujourd'hui et de taille insuffisante pour faire passer la centaine de pattes de certains circuits.

Votre circuit a donc de grandes chances de n'exister qu'en boîtier PLCC ou PGA, ce qui est un moindre mal car il existe des supports. Mais il peut aussi n'exister que dans des formats de boîtiers moins faciles à manipuler par un amateur ou même par une petite entreprise non équipée. En effet, si les CMS de première génération dont les pattes sont au pas de 1,27 mm sont encore soudables avec du soin et un fer à panne fine ; les boîtiers CMS les plus récents, avec un pas de 0,64 mm, ne se prêtent quasiment plus qu'à un soudage à la machine. Et, si l'on peut encore envisager de tenter tout de même leur soudure, cela devient carrément impossible avec les boîtiers BGA qui commencent à pointer le nez à l'horizon. Si vous ne savez pas ce que c'est, imaginez un carré de céramique sous lequel on a mis plein de petits plots dorés en forme de demi sphères (**figure 4**) et vous comprendrez l'ampleur du problème...

Avant d'entériner votre choix, nous vous conseillons donc de bien vérifier dans le tableau de sélection du fabricant quels sont les boîtiers proposés pour le circuit que vous avez retenu car, selon le cas, vous pourrez être amené à le reconsidérer. En ce début 2005, ce problème ne se pose cependant avec acuité que pour les circuits les plus récents.

Les microcontrôleurs fantômes

Contrairement à ce que peut laisser croire ce titre, ce ne sont pas des circuits produits dans les vieux manoirs de

Mini biographie

Ingénieur conseil en électronique, Professeur des Universités Associé et Expert Judiciaire en Electronique et Informatique, Christian Tavernier, dont les nombreux ouvrages et articles font autorité, est aussi et surtout un spécialiste incontesté des microprocesseurs et microcontrôleurs depuis plus de vingt-cinq ans, ce qui explique que nous ayons fait appel à lui pour la rédaction de cet article de fond.



www.tavernier-c.com

contact@tavernier-c.com

nos amis écossais mais tout simplement des microcontrôleurs qui n'existent ... que sur papier. À une certaine époque, Motorola (qui s'appelle aujourd'hui Freescale) s'était fait le champion de la catégorie avec sa famille 68HC05, pléthorique sur le papier mais dont on n'arrivait à se procurer réellement qu'un ou deux membres. Ce problème n'est pas crucial pour une grande entreprise car si elle souhaite commander des milliers de pièces présentes au catalogue, le fabricant finira toujours par les lui fournir. Par contre, si vous ne voulez que quelques pièces d'un modèle, voire même une seule, il est indispensable de vous assurer, avant de commencer votre étude, qu'au moins un distributeur la tient en stock, dans le type de boîtier qu'il vous faut de surcroît.

En guise de conclusion

Nous vous l'avons dit en guise d'introduction, nous ne terminons pas cet article en vous disant « voila le microcontrôleur qu'il vous faut ». Nous espérons cependant que l'exposé qui précède vous permettra de vous poser les bonnes questions et, pour vous aider à y répondre, nous vous proposons un tableau de synthèse qui devrait vous permettre de faire un premier choix.

(050003-1)



Avec ses quelque 128 broches, le RABBIT 3000 constitue un beau défi de soudure pour l'amateur de réalisations personnelles. (source Rabbit Semiconductor)

SC-Analyser 2005

Testeur/Identificateur de semi-conducteurs

Michel Waleczek

Les transistors sont des composants que l'on rencontre pratiquement dans tous les montages électroniques. L'amateur dispose souvent dans ses tiroirs un certain nombre de ces composants récupérés sur d'anciennes cartes d'origines diverses et qui n'attendent que d'être réutilisés.



Les caractéristiques de ceux dont le marquage est connu peuvent être retrouvées dans les fameux recueils de fiches de caractéristiques (*data book*). Les autres transistors resteront bien sagement dans leur tiroir en attendant la délivrance. Leur délivrance arrivera peut-être avec ce montage qui permet l'identification rapide des semi-conducteurs les plus courants (bipolaire, JFET, FETMOS, diodes et aussi les CMS non identifiables), l'identification de leur brochage ainsi que le calcul de plusieurs paramètres les caractérisant (H_{FE} pour les bipolaires, $V_{TH}/I_{DSS}/R_{DSON}$ pour les JFET, tension de seuil pour les FETMOS, courant/tension directs et courant de fuite pour les diodes). Toutes ces informations sont visualisées sur un écran à cristaux liquides (LCD).

Principe de fonctionnement

Chacune des 3 bornes du semi-conducteur inconnu peut être reliée, à travers une résistance déterminée, soit à la masse soit à une tension de +5 V. Chacune des résistances peut prendre l'une des valeurs suivantes : 100 Ω , 1 k Ω , 5,6 k Ω ou 100 k Ω . On obtient ainsi pour une configuration donnée 3 tensions qui sont mesurées par un microcontrôleur PIC16F876. Le microcontrôleur va dans un premier temps effectuer un filtrage grossier mais rapide dans le but de reconnaître un transistor bipolaire. On va pour ce faire utiliser le principe suivant : 2 parmi les 3 bornes du transistor sont mises à la masse, la 3^{ème} étant reliée au +5 V à travers une résistance de 5k Ω .

Le microcontrôleur mesure la tension sur cette dernière et mémorise la valeur. Deux autres essais sont réalisés en changeant l'ordre des bornes, mais **la mesure se faisant toujours au même point**, à savoir en aval de la résistance de 5,6 k Ω vers le transistor. On obtient de cette façon 3 valeurs qui correspondent en quelque sorte à la signature du composant. Le **tableau 1** résume les niveaux théoriques que l'on devrait obtenir pour un transistor de type N et pour un transistor de type P. Le symbole - correspond à la connexion de l'électrode considérée à la masse par une résistance de 100 Ω . Le symbole + correspond à une connexion au +5 V à travers 5,6 k Ω . Un transistor NPN donnera deux niveaux proches de 5V et un autre proche de 0,7 V, tandis qu'un transistor PNP se

Tableau 1. Table de détermination primaire				
Borne	E	B	C	Mesure
NPN	-	-	+	5 V
	+	-	-	5 V
	-	+	-	0,7 V
PNP	-	-	+	0,7 V
	+	-	-	0,7 V
	-	+	-	5 V

verra attribué un seul niveau proche de 5 V et deux niveaux proches de 0,7 V.

Nous pouvons d'ores et déjà remarquer que notre premier test permet de déterminer la position de la base du transistor, c'est la borne qui présente un niveau différent des deux autres. Le transistor ainsi reconnu passera un examen d'évaluation de son gain en courant. Comme les positions de l'émetteur et du collecteur ne sont pas connues, une mesure du gain en courant sera effectuée pour chacune des deux combinaisons possibles. La valeur finale correspondra à la plus grande des deux. Si les tensions mesurées ne correspondent à aucune combinaison du tableau 1 le composant sera soumis à des tests spécifiques pour les composants particuliers (transistor FETMOS, diode et transistor à effet de champ). Une tentative de détection des FETMOS sera réalisée en faisant passer le test du gain en courant comme pour les bipolaires pour les six combinaisons possibles comme nous le verrons un peu plus loin. De

même certains transistors bipolaires ont une signature différente de celles indiquées sur le tableau 1. C'est notamment le cas des transistors possédant une diode de protection entre collecteur et émetteur. Le transistor va alors passer le test de mesure du gain en courant pour les six combinaisons de brochage possibles.

Limitations

Autant le signaler tout de suite : SC Analyser 2005 n'est pas en mesure de tester les thyristors ni les transistors Darlington.

Détermination du gain en courant des transistors bipolaires

La base du transistor étant déterminée lors du crible précédant les deux autres bornes seront identifiées en configurant le transistor en collecteur commun ou en source suivieuse s'il s'agit d'un FETMOS (**figure 1**).

Le gain du transistor est déterminé par la mesure de V_B et de V_E . On a en effet :

$$V_E = R_E \times (\beta + 1) \times V_B / R_B$$

$$\beta = [(V_E \times R_B) / (V_B \times R_E)] - 1$$

L'étendue du gain susceptible d'être mesuré par notre montage s'étend de 5 à 999. Dans le cas d'un transistor FETMOS canal N (**figure 1c**) le courant de grille est proche de zéro, ce qui permet de le distinguer du transistor bipolaire. La tension de seuil correspond

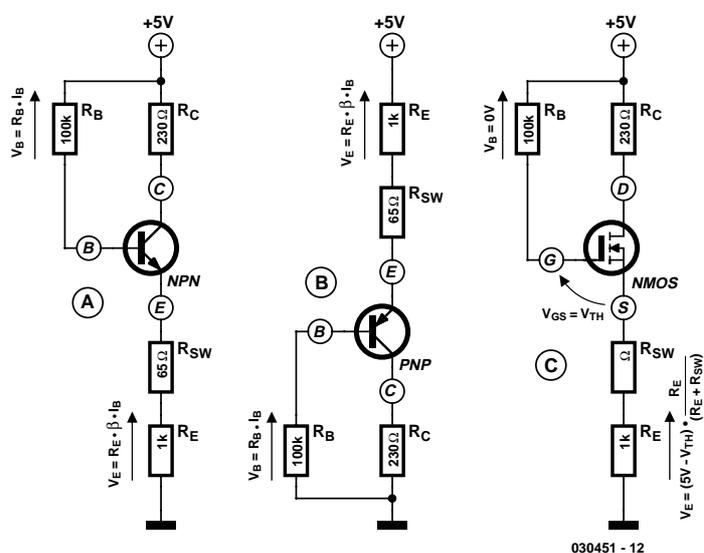


Figure 1. Configuration pour la mesure du β d'un transistor bipolaire et de la tension de seuil d'un FETMOS

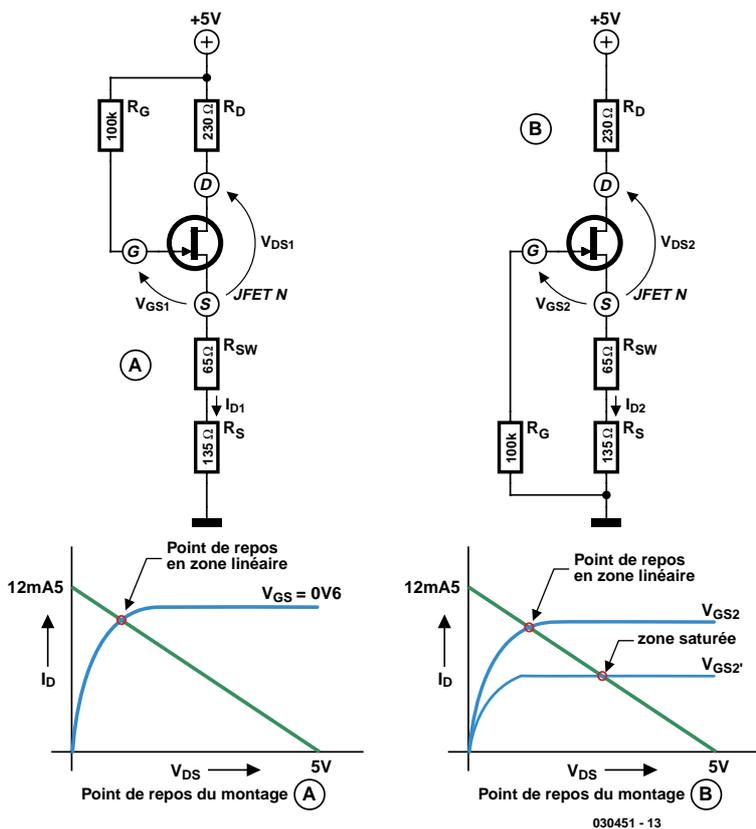


Figure 2. Configuration pour la mesure des paramètres des JFET.

alors à la tension $V_{CC}-V_E$ (pour un canal N). Pour qu'un transistor FET-MOS soit reconnu par notre montage, sa tension de seuil doit être inférieure à 4,5V et doit être du type à enrichissement, ce qui est le cas de la quasi-totalité de ces derniers. Le second type de FETMOS dit à appauvrissement n'est plus utilisé de nos jours.

Mesure des paramètres des transistors à effet de champ

Le transistor à effet de champ possède un certain nombre de paramètres le caractérisant mais nous nous intéresserons plus particulièrement à trois

d'entre eux: V_{TH} (tension de seuil), I_{DSS} (courant de saturation) et la résistance R_{DSON} . La mesure de ces paramètres est plus complexe que la simple mesure du gain en courant d'un transistor bipolaire. La topologie du schéma ne se prête guère à une mesure directe, c'est la raison pour laquelle nous allons utiliser une méthode inédite utilisant une modélisation mathématique du transistor (le modèle de Schichman-Hodges) afin de déterminer par calcul les trois paramètres précités. Ce modèle statique utilisé notamment dans les simulateurs Spice comprend trois paramètres principaux mais nous n'en conserverons que deux au final, le troisième (λ) n'apportant pas d'erreur sensible sur le calcul final (voir l'encadré). La détermination des deux paramètres restants nous amène à considérer deux points de repos distincts afin d'obtenir deux systèmes d'équations. Le transistor sera dans un premier temps polarisé dans sa région linéaire (figure 2a) en forçant la tension V_{GS1} à 0,6V environ par l'intermédiaire de la résistance R_G . Certains transistors à faible courant de saturation peuvent être polarisés dans leur zone de saturation, dans ce cas une variante du schéma de la figure 2 sera utilisée que nous ne détaillerons pas. Cette première configuration nous donne une première série de valeurs V_{DS1} , V_{GS1} et I_{D1} . Le deuxième point de repos sera obtenu en faisant fonctionner le transistor en source de courant (figure 2b), d'où une deuxième série de mesures V_{DS2} , V_{GS2} et I_{D2} . Les choses se compliquent quelque peu car le deuxième point de repos peut se trouver tout aussi bien en zone linéaire qu'en zone saturée. La zone de fonctionnement ne peut être connue que si la tension de seuil V_{TH} est elle-même connue. Nous n'avons pas d'autre choix que de calculer pour chacune des 2 zones de fonctionnement possibles la tension V_{TH} associée, puis de sélectionner parmi les deux résultats la valeur correcte en vérifiant à *posteriori* les zones de fonctionnement respectives.

Sur le schéma de la figure 3 nous avons représenté les résistances R_{SW} des commutateurs analogiques. Celles-ci sont de l'ordre de 65 Ω , cependant leurs valeurs exactes sont déterminées précisément par une procédure de calibrage automatique. La résistance de drain d'environ 230 Ω comprend la résistance de 100 Ω , la résistance du commutateur analogique (65 Ω) ainsi que la résistance de sortie

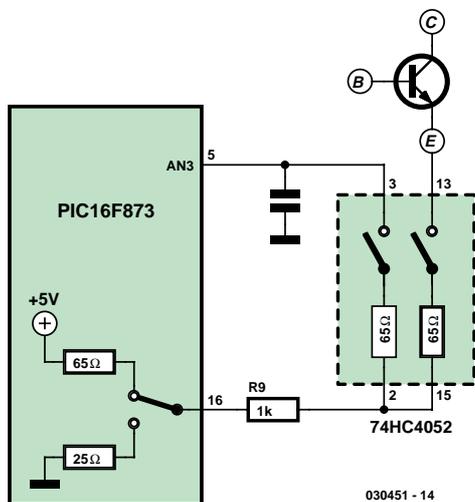


Figure 3. Le montage de type Kelvin permet de s'affranchir des résistances internes du 74HC4052.

du microcontrôleur (65 Ω). Notez par ailleurs que la grande majorité des transistors à effet de champ sont symétriques, c'est-à-dire que le drain et la source jouent le même rôle. Il n'est donc pas possible de déterminer ces deux électrodes. Ainsi seule la grille sera réellement identifiée par notre appareil. Cependant le drain et la source seront affichés conformément à la configuration utilisée lors du calcul des paramètres du transistor. La permutation du drain et de la source du transistor ne changera pas le marquage sur l'afficheur du SC-Analy-

ser 2005, mais les résultats des calculs seront toujours en correspondance avec le brochage indiqué.

Schéma électrique

Le schéma complet du montage est donné à la **figure 4**. Le montage consommant environ 6 mA (hors rétro-éclairage, ce dernier consommant de l'ordre de 20 mA) est alimenté par une pile 9 V. Un régulateur 78L05 associé à 3 condensateurs de découplage permet d'abaisser la tension à 5 V précisément, permettant d'alimenter le

PIC16F876, l'afficheur à cristaux liquides (à rétro-éclairage) ainsi que les 3 commutateurs analogiques 74HC4052. L'afficheur communique avec le microcontrôleur en mode 4 bits via 5 broches du Port C et par la broche RA5 du port A. Le PIC16F876 « tourne » à 1 MHz environ grâce au réseau R10/C4. Il dispose par rapport au classique 16F84 d'une unité de conversion analogique-numérique que nous mettons à contribution dans ce montage. La borne RESET est connectée directement au +5 V, le microcontrôleur disposant d'un circuit de reset

Calcul des paramètres des transistors à effet de champ avec le modèle de Schichman-Hodges

La caractéristique statique d'un transistor à effet de champ à jonction peut être modélisée par des équations donnant le courant de drain fonction des tensions V_{GS} et V_{DS} du transistor. Il y a 2 équations car le modèle distingue la zone linéaire ($0 \leq V_{DS} \leq V_{GS} - V_{TH}$) et la zone saturée ($V_{DS} \geq V_{GS} - V_{TH}$), V_{TH} étant la tension de seuil du transistor. Le modèle complet utilise 3 paramètres β , λ et V_{TH} . Le paramètre β est lié au courant de saturation et à la tension de seuil V_{TH} , le paramètre λ modélise la modulation de la largeur du canal responsable de la résistance interne dans la zone saturée, et sera négligé dans nos calculs ($\lambda=0$). Le modèle mathématique que nous utiliserons sera donc le suivant :

Pour $0 \leq V_{DS} \leq V_{GS} - V_{TH}$ (zone linéaire)

$$I_D = \beta \cdot V_{DS} \cdot [2 \cdot (V_{GS} - V_{TH}) - V_{DS}] \cdot (1 + \lambda \cdot V_{DS})$$

Et pour $V_{DS} \geq V_{GS} - V_{TH}$ (zone saturée)

$$I_D = \beta \cdot (V_{GS} - V_{TH})^2 \cdot (1 + \lambda \cdot V_{DS})$$

Nous avons vu que le transistor dans la première configuration était toujours en zone linéaire pour une valeur I_{DSS} suffisante, on obtient donc une première équation :

$$I_{D1} = \beta \cdot V_{DS1} \cdot [2 \cdot (V_{GS1} - V_{TH}) - V_{DS1}] \quad (1)$$

Le transistor dans la seconde configuration peut se trouver en mode linéaire

$$I_{D2} = \beta \cdot V_{DS2} \cdot [2 \cdot (V_{GS2} - V_{TH}) - V_{DS2}] \quad (2)$$

ou en mode saturé :

$$I_{D2} = \beta \cdot (V_{GS2} - V_{TH})^2 \quad (3)$$

On obtient un système de 2 équations dont les inconnues sont V_{TH} et β . Selon la zone de travail du transistor on utilisera l'équation (2) ou l'équation (3).

Si on suppose que le transistor travaille en zone linéaire, on obtient grâce aux équations (1) et (2) :

$$V_{THLIN} = C - (V_{GS2} - V_{DS2})^2 / (V_{DS2} - V_K) \quad (4)$$

Avec $V_K = (I_{D2} / I_{D1}) \cdot V_{DS1}$

$$\text{Et } C = V_{GS2}^2 + V_K \cdot (V_{DS1} - 2 \cdot V_{GS1}) \quad (5)$$

Introduisons pour la suite des calculs la constante B :

$$B = 2 \cdot (V_K - V_{GS2}) \quad (6)$$

Dans le cas où le point de repos du transistor se trouve dans la zone saturée on obtient grâce aux équations (1) et (3) une équation du second degré :

$$V_{TH}^2 + B \cdot V_{TH} + C = 0$$

On obtient deux solutions V_{TH1} et V_{TH2} :

$$V_{TH1} = [-B + \sqrt{(B^2 - 4 \cdot C)}] / 2 \quad (7)$$

Et

$$V_{TH2} = [-B - \sqrt{(B^2 - 4 \cdot C)}] / 2 \quad (8)$$

Seule l'une des deux solutions est physiquement possible. Matheux à vos plumes! Il ne reste plus qu'à évaluer le signe des quantités V_1 et V_2 suivantes afin de déterminer la zone de fonctionnement du transistor dans la deuxième configuration :

$$V_1 = V_{GS2} - V_{THLIN} - V_{DS2}$$

$$V_2 = V_{GS2} - V_{THSAT} - V_{DS2}$$

Si $V_1 > 0$ et $V_2 < 0$ le transistor fonctionne en mode linéaire et $V_{TH} = V_{THLIN}$

Si $V_1 < 0$ et $V_2 > 0$ le transistor fonctionne en mode saturé et $V_{TH} = V_{THSAT}$

Le paramètre I_{DSS} correspond au courant de drain lorsque $V_{GS} = 0$ et lorsque le transistor se trouve dans la zone saturée. Nous prendrons pour V_{DS} la valeur correspondant à la limite de la zone de saturation soit $V_{DS} = V_{GS} - V_{TH} = -V_{TH}$.

On obtient en substituant V_{DS} dans l'équation (2) ou (3)

$$I_{DSS} = \beta \cdot V_{TH}^2$$

La valeur de β se calcule facilement grâce à l'équation (1) :

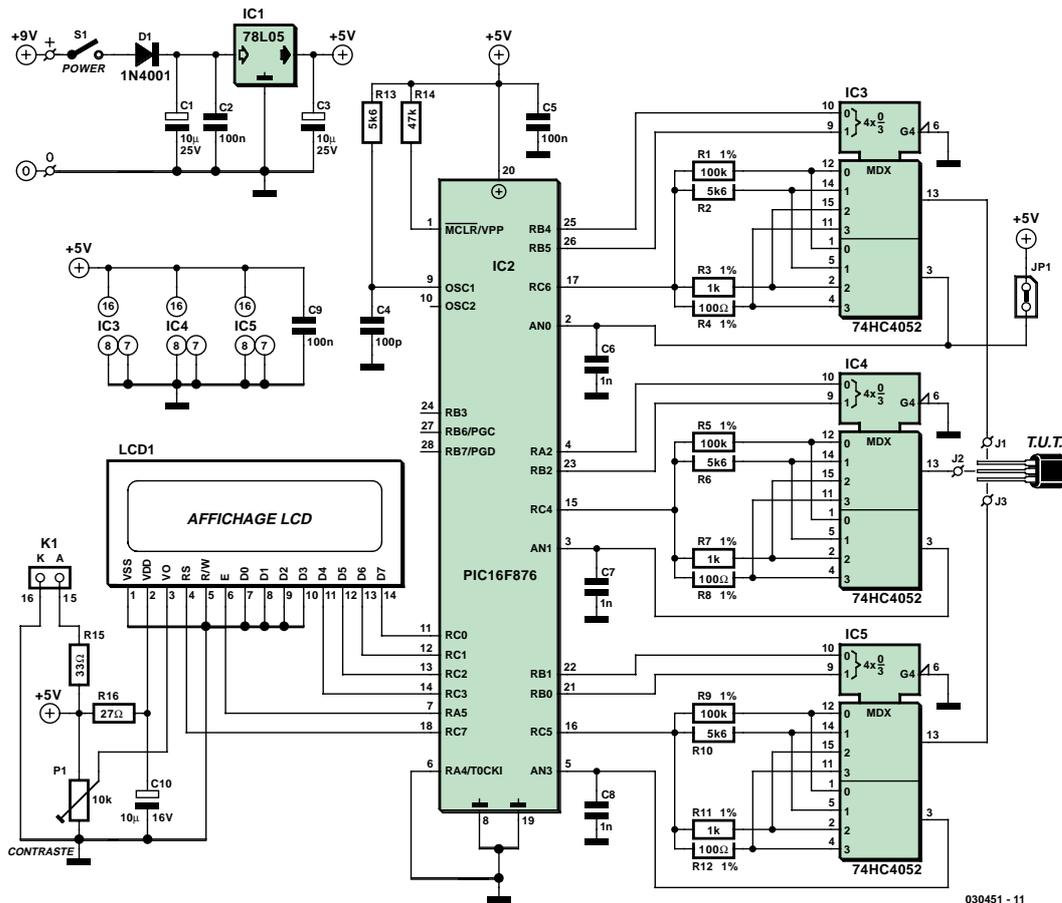
$$\beta = I_{D1} / V_{DS1} \cdot [2 \cdot (V_{GS1} - V_{TH}) - V_{DS1}]$$

La résistance R_{DSON} correspond à la pente à l'origine de la caractéristique $V_{DS} = f(I_D)$ lorsque $V_{GS} = 0$.

Pour V_{DS} proche de 0 V l'équation (1) donne en négligeant le terme du second ordre :

$$I_D = -2 \cdot \beta \cdot V_{DS} \cdot V_{TH}$$

$$R_{DSON} = dV_{DS} / dI_D = -1/2 \cdot \beta \cdot V_{TH}$$



030451 - 11

Figure 4. Schéma détaillé du SC-Analyser 2005.

automatique. Les 3 signaux de commande du transistor inconnu sont issus du Port C, RC4, RC5 et RC6. Ces 3 signaux qui peuvent prendre la valeur 0 V ou 5 V sont aiguillés vers les trois bornes de test via 3 multiplexeurs analogiques dont le rôle est d'intercaler une résistance entre les sorties du microcontrôleur et chacune des trois bornes. La sélection des différentes résistances est effectuée par la combinaison des signaux RB4/RB5 pour le signal de droite (J1), RA2/RB2 pour le signal du centre (J2), et les signaux RB1/RB0 pour le signal de gauche (J3, ceci vaut pour la platine vue de face). Les tensions présentes sur les trois bornes sont mesurées par le PIC16F876 grâce aux entrées analogiques AN0, AN1 et AN3. Afin de s'affranchir des résistances internes des commutateurs lors de la mesure des courants, la mesure des tensions n'est pas effectuée directement sur les bornes du transistor inconnu, mais aux bornes des résistances en utilisant le second multiplexeur du 74HC4052. La figure 3 schématise le fonctionnement lorsque la résistance R9 est com-

mutée. Seules interviennent les résistances du microcontrôleur qui sont de l'ordre d'une trentaine d'ohms. Pour finir les condensateurs de 1 nF effectuent un léger filtrage des signaux.

Le programme

Le programme a été écrit entièrement en assembleur et utilise une bonne partie de l'espace code du PIC16F876 dont près de la moitié est utilisée pour le seul calcul des paramètres des transistors à effet de champ. Seul le code .hex servant à la programmation du microcontrôleur est à votre disposition sur le site Elektor : www.elektor.fr (disquette **EPS030451-11**). Vous pouvez également vous procurer le microcontrôleur programmé (**EPS030451-41**) auprès des adresses habituelles.

Réalisation

Le tracé du circuit imprimé et l'implantation des composants sont donnés en **figure 5**. On veillera à respecter la bonne orientation des supports de circuits intégrés, des deux condensateurs

chimiques, du régulateur 5 V et des 4 circuits intégrés. Les résistances à 1% peuvent être remplacées par des résistances à couche métallique à 5% triées au multimètre. Les commutateurs analogiques seront obligatoirement du type 74HC, les versions CMOS ayant une résistance interne trop élevée pour notre application. L'afficheur à cristaux liquides sera fixé sur le circuit imprimé « côté pistes ». Afin d'être démontable, l'afficheur sera muni d'une barrette sécable mâle-mâle. Le circuit imprimé sera doté quant à lui d'une barrette de 16 contacts femelle. Nous avons opté pour une version moderne dite à P-LED (cf. l'encadré consacré aux P-LED et O-LED), mais il est également possible d'utiliser un afficheur LCD quelconque à base de contrôleur Hitachi HD44780 (ou compatible; à noter que cela se traduira quasi-inévitablement par un brochage différent du connecteur qui forcera à adopter la solution câble en nappe soudé en fonction du brochage du connecteur de l'afficheur). Attention, le brochage du superbe afficheur utilisé ici est très

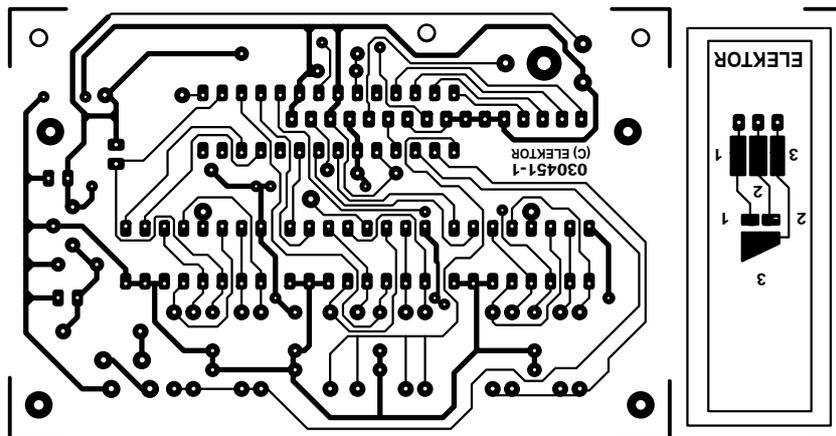
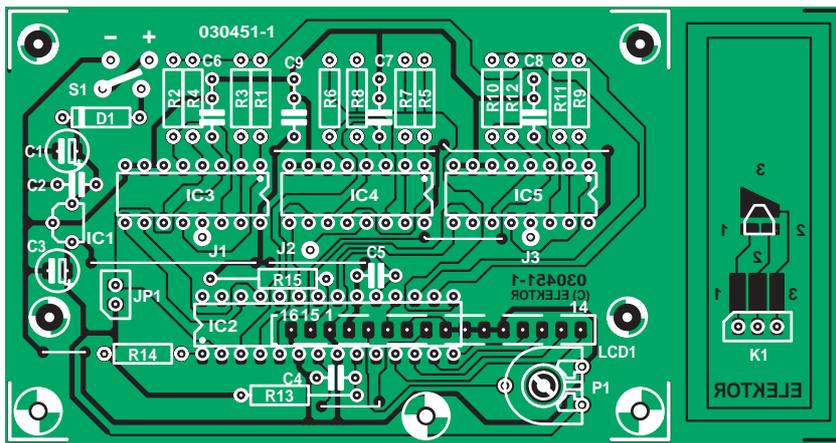


Figure 5. Dessin des pistes sérigraphie de l'implantation des composants. La petite platine permet aussi de tester des composants CMS.

particulier, commençant par la broche 14 pour aller vers la broche 1 et terminer par les broches 15 et 16. En ce qui concerne le support de test du transistor, nous avons dessiné une platine spéciale qui sera reliée à la pla-

tine principale. Cette extension permet également le test des semi-conducteurs (diodes et transistors) de type CMS (**M**ontage en **S**urface). On connectera ce support au circuit imprimé principal par 3 fils souples, en veillant à

Liste des composants

Résistances :

R1, R5, R9 = 100 k Ω /1%
 R2, R6, R10, R13 = 5k Ω
 R3, R7, R11 = 1 k Ω /1%
 R4, R8, R12 = 100 Ω /1%
 R14 = 47 k Ω
 R15 = 33 Ω
 R16 = 27 Ω
 P1 = ajustable 10 k Ω

Condensateurs :

C1, C3 = 10 μ F/25 V radial
 C2, C5, C9 = 100 nF
 C4 = 100 pF
 C6 à C8 = 1 nF MKT
 C10 = 10 μ F/16 V radial

Semi-conducteurs :

D1 = 1N4001
 IC1 = 78L05
 IC2 = PIC16F876-20/SP (programmé **PS040409-41**)
 IC3 à IC5 = 74HC4052 (HC uniquement !)

Divers :

S1 = interrupteur unipolaire à glissière
 LCD1 = affichage LCD 2x16 avec rétro-éclairage ASI-G-162FS-GF-EWS/W
 Embase + connecteur 1 x 16 ou câble plat pour connexion de l'affichage
 JP1 = embase à 1 rangée de 2 contacts + cavalier
 3 mini-pinces crocodile*
 supports 3 x 16 et 1 x 28
 boîtier Hammond 1591BTBU
 pile 9 V + connecteur à pression
 5 ponts de câblage
 * cf. texte

Le code .hex pour la programmation du contrôleur est disponible sur : www.elektor.fr

respecter l'ordre naturel du circuit imprimé (vu du dessus) : contact 1 sur J1, contact 2 (centre) sur J2 et le dernier, contact 3, sur J3. En fait, cela n'a guère d'importance pour la mesure, mais bien pour l'identification des broches, si tant est que le **point central soit relié à J2**. On pourra aussi utiliser des fils souples munis de grip-fils. Le circuit est prévu pour prendre place dans un boîtier standard doté d'un compartiment pour pile 9 V.

Étalonnage

Après mise sous tension de l'appareil on voit apparaître un message d'invite donnant la version du logiciel (*SC-Analyser 2005 Elektor Rev. 1.0e*). Si tel n'était pas le cas, il faut voir si une rotation du potentiomètre de réglage de contraste P1 change quelque chose à la situation.

Avant toute chose, il faut procéder à un calibrage des résistances internes des commutateurs analogiques. La connaissance exacte de celles-ci permet d'accroître la précision de la mesure des différents paramètres du semi-conducteur à tester. Dans le cas où le calibrage ne serait pas effectué, la valeur 65 Ω est prise par défaut pour chaque commutateur. Lors de chaque mise en marche, le logiciel va s'assurer que le calibrage a été effectué et le cas échéant affiche le message « Cal error » pendant une seconde environ. La procédure d'étalonnage est simple et automatique. Il faut pour cela de mettre le testeur sous tension en ayant pris soin de positionner un cavalier sur JP1 et de connecter ensemble les

Programmation du μ C

Le microcontrôleur devra être programmé avec les options suivantes :

- **Oscillateur en mode RC**
- **Chien de garde (WDT) non validé**
- **Timer à la mise sous tension validé**
- **Brown-out reset non validé**
- **Protection EEPROM non validé**
- **Autorisation d'écriture en mémoire flash non validé**
- **Mode debug non validé**
- **Protection du code non validé**

3 bornes de test. Le testeur affiche alors le message « *Cal Remove jumper* ». Le cavalier de court-circuit doit alors être retiré et la procédure de calibrage démarre. 3 résistances sont mesurées successivement, avec l'affichage successif de leurs valeurs. On voit ensuite apparaître le message « *Short R_{SH} XX Ω* ». Libérer ensuite les 3 points de mesure et le testeur entre de lui-même dans le mode de test « *No component *-*-** ».

Notez qu'il est possible de refaire un calibrage à tout moment (couper l'alimentation, implanter le cavalier, remettre l'alimentation).

Que faire en cas de problème : vérifier que l'on a bien mis en place et soudé les 7 ponts de câblage. Voir si les supports véhiculent bien la tension d'alimentation aux broches prévues.

Utilisation

Après un message d'invite donnant la version du logiciel (*SC-Analyser 2005 Elektor Rev 1.0e*), l'afficheur doit, en l'absence de composant, se manifester comme le montre la photo 1.

La première ligne donne les indications sur le type de transistor et la valeur d'un des paramètres le caractérisant. La seconde ligne donne le brochage du transistor et la valeur d'un second paramètre si applicable.

Transistor bipolaire

Sur la première ligne de l'écran LCD sont affichés la polarité du transistor (PNP ou NPN), le type de semi-conducteur (silicium ou germanium) et bien entendu le paramètre H_{FE} . Suivra le brochage du transistor sur la seconde

ligne ainsi que le courant de collecteur. Ce dernier courant se trouvera dans une fourchette de 1,5 mA à 4 mA environ selon la valeur du gain en courant. Selon la valeur du V_{BE} mesurée, SC-Analyser 2005 indique le type de semi-conducteur (germanium, Ge ou silicium, Si)

Au cas où le transistor est partiellement ou totalement en court-circuit SC-Analyser 2005 marque les bornes en court-circuit par un « X ». Un court-circuit est détecté lorsque la résistance mesurée entre deux bornes est inférieure à 50 Ω pour les deux sens du courant (photo 3).

Transistor à effet de champ

Les paramètres V_{TH} , I_{DSS} et $R_{DS(ON)}$ s'affichent successivement toutes les deux

O-LED et P-LED

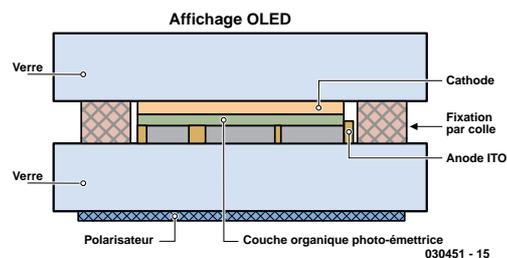
Les O LED (Organic LED) et P-LED (Polymeric-LED) sont des développements « récents » des LED. L'effet O-LED a été découvert au début des années 80 par Eastman Kodak, mais ce n'est que récemment que des matériels l'utilisant ont trouvé leur chemin dans des applications commerciales (PDA, lecteurs MP3). L'appareil photo numérique de Kodak LS633 fut l'un des premiers matériels à être doté d'un écran O-LED. Certains baladeurs MP3 de Packard Bell sont eux aussi dotés d'un affichage O-LED du plus bel effet.

O-LED repose sur la technologie du LCD, un sandwich de plusieurs couches de films au carbone coincés entre 2 électrodes chargées, une cathode métallique et une anode transparente. Les films organiques sont au nombre de 4 : une couche à injection de trous, une couche de transport de trous, une couche d'émission et une couche de transport d'électrons. Lors de l'application d'une tension, les charges positives et négatives injectées se recombinent dans la couche d'émission produisant ainsi de la lumière électroluminescente. Contrairement aux LCD qui requièrent un rétro-éclairage, les affichages OLED sont actifs émettant de la lumière.

P-LED, plus récent qu'O-LED, est une technologie d'affichage, d'illumination et de rétro-éclairage obtenue par prise en sandwich d'un polymère non-dopé entre 2 électrodes. Le polymère s'illumine lorsqu'il est soumis à une tension. Ces produits permettent l'affichage de toutes les couleurs et sont relativement bon marché comparés à d'autres technologies d'affichage telles que le LCD et O-LED.

Leurs avantages : ils ne requièrent que peu de puissance pour être très lumineux, sont plus légers, plus solides, ont une réponse plus rapide et une plage de températures plus étendue.

Comme vous l'avez compris, nous sommes extrêmement enthousiastes quant aux caractéristiques optiques de l'affichage PLED utilisé. Nous avons cependant de très grandes réserves quant à l'aspect « conception ». Il génère en effet d'énormes crêtes de courant allant jusqu'à plusieurs centaines de mA, ce qui se traduit par de très gros parasites sur la ligne 5 V. Cela tient, à notre avis, à une mauvaise conception du matériel (version A) où l'on n'a pas suffisamment tenu compte des temps de commutation des circuits de commande de l'afficheur. Il nous a fallu, pour contrer ce problème, ajou-



Coupe d'un affichage O-LED.
Le principe du P-LED est similaire.

ter un réseau RC sur la ligne 5 V. Nous ne savons pas si les afficheurs PLED d'autres fabricants présentent le même défaut, mais nous vous recommandons, si tant est que cela soit possible, d'opter pour un affichage PLED d'un autre fabricant, ceci vous évitera bien des soucis.



secondes environ. Les positions drain et de la source sont fixées arbitrairement par le SC-Analyser 2005 mais correspondent à la configuration réelle utilisée pour la mesure des paramètres. La permutation du drain et la source du transistor sous test permet d'obtenir la valeur des paramètres pour la configuration inverse, mais ne devrait pas donner de variation sensible du fait de la symétrie des transistors à effet de champ. La précision des valeurs données par SC-Analyser 2005 est assez bonne compte-tenu de la méthode employée. Elle se situe aux environs de $\pm 5\%$ pour le courant I_{DSS} et de $\pm 5 \Omega$ pour la résistance $R_{DS(on)}$. La tension de seuil quant à elle peut s'écarter de la réalité de 0,5 V pour certains transistors, la précision généralement constatée étant de l'ordre de 50 mV. Un transistor X de marque M1 pourra donner une valeur très proche de la réalité, alors que le même transistor sous une marque M2 donnera une valeur s'écartant de quelques centaines de millivolts. La précision est liée à la justesse du modèle de Schichman-Hodges face à la réalité et montre par là même les limites de celui-ci (cf. le cadre consacré au sujet).

Affichage de la tension de seuil

La tension de seuil est négative pour les transistors à canal N et positive pour ceux à canal P. La limite maximale est fixée à ± 20 V. La résolution est de 10 mV pour les valeurs de 0 à $\pm 9,99$ V et de 100 mV au-delà (photo 4).

Affichage du courant de saturation

Le courant de saturation est obtenu pour une tension (calculée) grille-source nulle. La valeur du courant de saturation s'étend de 0 à 99,9 mA. La résolution est de 10 μ A pour les courants jusqu'à 10mA et 100 μ A jusqu'à 100mA (photo 5).

Affichage de la résistance drain-source

La résistance affichée correspond à la pente à l'origine de la caractéristique $V_{DS}(I_D)$ pour une tension V_{GS} nulle. La mesure s'étend de 0 à 999 Ω (photo 6).

Transistor FETMOS

La tension indiquée correspond à la tension de seuil du transistor pour un courant drain de l'ordre de 2 mA. La mesure s'étend de 0 à 4,50 V avec une résolution de 10mV. Le courant de fuite de grille ne doit pas dépasser la valeur de 0,5 μ A pour que le transistor soit détecté (photo 7).

Diodes

La diode doit être insérée entre les bornes gauche et droite du testeur. Le SC-Analyser 2005 indique la position de l'anode et de la cathode. Le testeur donne 3 affichages successifs, toutes les 2 secondes environ. Le premier donne la tension directe ainsi que le courant traversant le semi-conducteur pour une résistance de limitation de 400 Ω environ. Le courant maximal se situe alors aux alentours de 12 mA (photo 8).

Le second indique la même chose mais avec une résistance de limitation de 5,9 k Ω environ, laquelle fixe le courant maximum à 800 μ A environ (photo 9).

Le dernier indique le courant inverse ainsi que la tension de test en valeur absolue. La résolution est de 100 nA (photo 10).

Les composants CMS

Vous vous êtes sans doute demandé comment utiliser la section CMS présente sur la platine auxiliaire. Il suffit de faire en sorte que les pattes du composant concerné soient bien en contact avec les surfaces cuivrées correspondantes. La diode CMS doit être posée entre les surfaces 1 et 3. On pourra utiliser un outil en plastique pour faire en sorte que les broches fassent bien contact.

En conclusion

SC-Analyser 2005 est un petit instrument simple à réaliser et doté de grandes possibilités qui rendra de bons et loyaux services tant pour l'amateur que pour le professionnel. Citons pêle-mêle: la recherche d'équivalence, le tri, l'appareillage, la détermination de transistors dont le marquage est inconnu, ou tout simplement la recherche de brochage sans avoir à consulter de *data book*...

À vos fers à souder!

(030451-1)

Liens Internet :

Site de l'auteur : www.mwinstruments.com

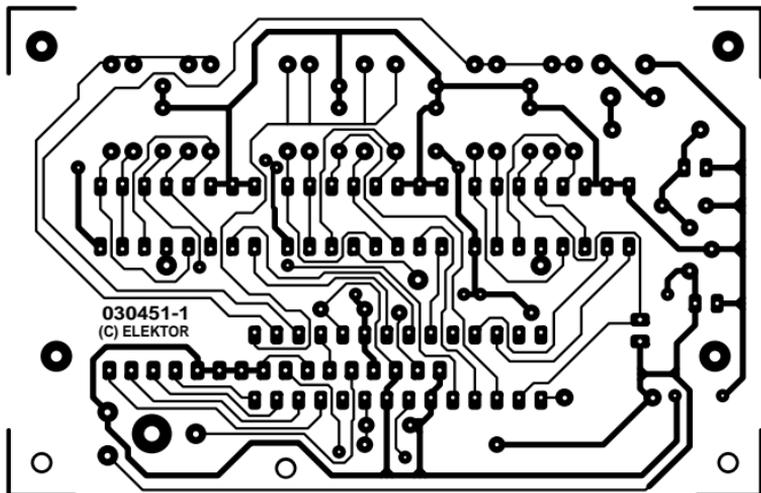
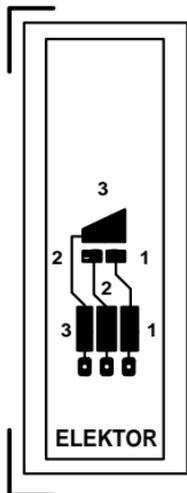
Section FAQ : www.mwinstruments.com/SCA2005/sca2005.html

Microchip : www.microchip.com/download/lit/pline/picmicro/families/16f87x/30292c.pdf

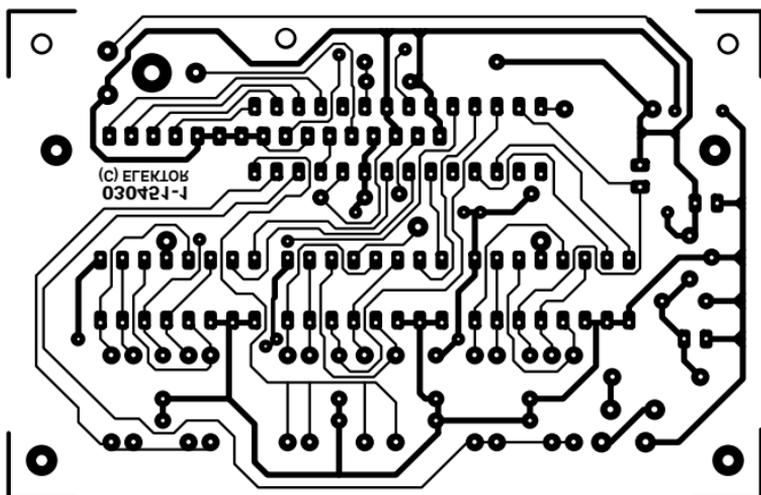
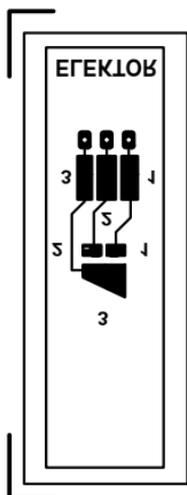


L'auteur

L'auteur est né en 1965, est diplômé de l'ENSERG (Ecole Nationale Supérieure d'Electronique et de Radioélectricité de Grenoble). Il a travaillé durant sept ans dans une grande société de pesage française en tant qu'ingénieur d'études avant d'intégrer la filiale française de la société Hameg où il occupe le poste de directeur des études. Entre deux voyages au bout du monde il trouve un peu de temps pour réaliser des petits instruments de mesure destinés aux amateurs.



non reflected

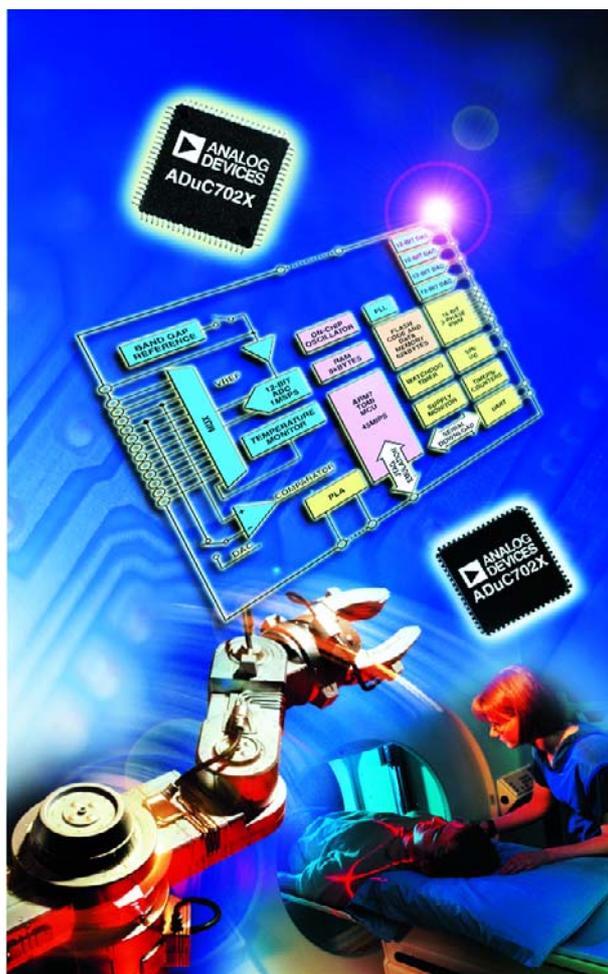


reflected

µC analogique de précision

DU NOUVEAU EN

Michael Müller-Aulmann



Dès 1998, le MicroConverter® ADuC812, doté du cœur des microcontrôleurs à 8 bits de la famille 8051, a fait son apparition sur le marché. Toujours en recherche de puissance de travail et de perfectionnements, les concepteurs ont abouti à la série des microcontrôleurs en 16/32 bits à structure RISC basés sur le concept ARM7TDMI *precision analog microcontroller*.

En intégrant un convertisseur A/N de 12 bits à huit canaux et deux convertisseurs N/A de 12 bits en complément d'une mémoire de programme Flash de 8 Koctets, d'une mémoire de données Flash/EE de 640 octets, autour d'un processeur 8052 classique, Analog Devices fondait en 1998 la série MicroConverter®. D'ailleurs, Elektor vous a présenté fin 2001 une série d'articles intitulés « ISAC contrôleur capteur/actionneur intelligent » à propos de cet ADuC812. Au fil du temps, la famille s'est développée et s'est imposée sur le marché.

Dans de nombreuses applications, ces microcontrôleurs constituent, avec l'infatigable piocheur de la gamme des 8 bits dont ils sont équipés, une excellente solution à

de calcul et surtout les performances obtenues en analogique vont sans arrêt de l'avant. Les microcontrôleurs de 16 et 32 bits sont à présent disponibles, on les met à l'ouvrage. Et ce n'est pas sans raison :

- ils permettent de réaliser la totalité des étapes, routines d'interruption comprises, en langage évolué, généralement en C ou C++,
- ils accroissent les possibilités de calcul,
- ils mettent à disposition de nombreuses fonctions supplémentaires.

Cette évolution a conduit Analog Devices à étendre la famille des MicroConverter® par la branche « Precision

MICROCONVERTER

Tableau 1. Variante de microcontrôleurs prévues au programme

Dénomination	Canaux A/N	Canaux N/A	Flash (octets)	SRAM (octets)	GPIO (max)	MLI (PWM) 3 phases	Boîtier	Outils d'évaluation
ADuC7020 BCP62	5	4	62k	8 K	14	–	LFCSF 40 broches 6 mm x 6 mm	EVAL- ADuC7020QS
ADuC7021 BCP62	8	2	62k	8 K	13	–	LFCSF 40 broches 6 mm x 6 mm	
ADuC7021 BCP32	8	2	32k	4 K	13	–		
ADuC7021 ACP32	8	2	32k	4 K	13	–		
ADuC7022 BCP62	10	–	62k	8 K	13	–	LFCSF 40 broches 6 mm x 6 mm	
ADuC7022 BCP32	10	–	32k	4 K	13	–		
ADuC7022 ACP32	10	–	32k	4 K	13	–		
ADuC7024 BST62	10	2	62k	8 K	30	ja	LQFP 64 broches	EVAL- ADuC7024QS
ADuC7024 BCP62	10	2	62k	8 K	30	ja	LFCSF 64 broches 9 mm x 9 mm	
ADuC7025 BCP62	12	–	62k	8 K	30	ja		
ADuC7025 BCP32	12	–	32k	4 K	30	ja		
ADuC7026 BST62	12	4	62k (+Ext)	8 K (+Ext)	40	ja	LQFP 80 broches 14 mm x 14 mm	EVAL-ADuC7026QS
ADuC7027 BST62	16	–	62k (+Ext)	8 K (+Ext)	40	ja		
ADuC7027 AST62	16	–	62k (+Ext)	8 K (+Ext)	40	ja		

Analog Microcontroller » ADuC702x, fondée sur le concept ARM7TDMI et équipée du processeur RISC à 16/32 bits. Comme c'était déjà le cas pour le noyau du 8052, il s'agit là du cœur d'un microcontrôleur qui a trouvé sa place dans le marché et propose quantité d'outils de développement.

Le coeur ARM

On peut scander le processeur ARM7TDMI jusqu'à 45 MHz. On trouve l'ARM7, comme contrôleur RISC à 32 bits, dans nombre d'applications telles que le PDA (Personal Digital Assistant), mais avec ses quatre désinences TDMI, il s'octroie encore d'autres offices dans les applications à microcontrôleur embarqué, tels que le GameBoy Advance de Nintendo. Mais d'où sortent ces quatre lettres capitales et que signifient-elles, accolées à la dénomination du cœur de ce processeur ?

T = mode Thumb

Indépendamment de ses instructions à 32 bits, le microcontrôleur connaît aussi un jeu d'instructions à 16 bits, plus économes de la mémoire. Elles sont particulièrement avantageuses pour les systèmes embarqués si, comme sur les ADuC702x, la mémoire flash utilisée principalement pour stocker le logiciel s'organise sur 16 bits de largeur. Il ne faut alors au processeur qu'un seul cycle d'horloge pour appeler en mémoire une instruction.

D = interface à Déboguier

Il y a une interface JTAG intégrée au cœur du microcontrôleur pour l'accès aux outils de développement. Nul besoin donc de connecteur supplémentaire pour émulateur. Sur cinq broches du composant, on peut brancher non seulement un émulateur, mais réaliser la commande

complète de la puce en cours de développement, y compris la programmation de la mémoire flash intégrée.

M = Multiplier

Intégré aussi, on trouve un accumulateur multiplicateur qui calcule un produit de 64 bits à partir de deux opérandes de 32 bits.

I = émulateur Interne

À côté de l'interface JTAG sont intégrés deux autres blocs qui permettent l'introduction très souple de points d'arrêt pour faciliter la conception.

Une explication détaillée du fonctionnement de l'ARM7TDMI sortirait du cadre de cet article. Mais une des raisons de la renommée de l'ARM7 réside dans le fait que toutes les informations sur le processeur d'ARM Ltd. sont disponibles dans les manuels et les feuillets de caractéristiques (cf. ci-dessous), lesquels totalisent environ 1 200 pages qui détaillent tout au point de permettre le développement d'un émulateur du processeur. Vous pouvez également vous les procurer gratuitement sur le site Internet de ARM Ltd ainsi qu'un complément d'information sur le cœur du microcontrôleur ARM sous la forme d'un CD-ROM.

Les ADuC702x

Depuis l'été dernier, les ADuC702x sont en effet disponibles en 14 versions différentes (cf. **tableau 1**). Lors de leur conception, l'accent a bien entendu été mis sur la périphérie analogique, mais dans la partie numérique aussi, on a intégré des blocs très séduisants pour l'utilisateur. Comme on peut l'apercevoir dans le bloc-diagramme de l'un des plus puissants composants de la famille, l'A-

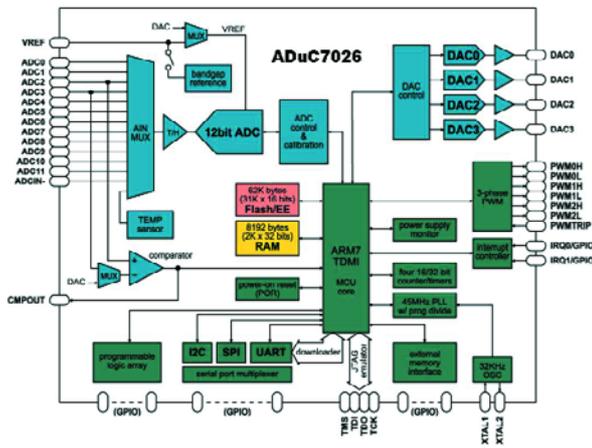


Figure 1.
La structure interne de l'ADuC7026.

DuC7026 (figure 1), outre le microcontrôleur central que l'on peut scander jusqu'à 45 MHz, il y a 62 Koctets de mémoire Flash (31 K x 16) et 8 Koctets de RAM (2K x 32 bits). On peut utiliser ces deux blocs aussi bien comme mémoire de programme que comme mémoire de données, grâce au fait que l'ARM7TDMI dispose d'un adressage linéaire unique de 2^{32} octets (4 Go). Les interfaces de commande des périphériques forment la partie supérieure de 64 Koctets du domaine d'adresse (0xFFFF0000

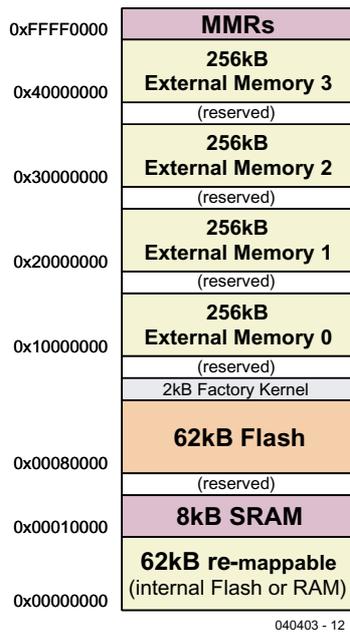


Figure 2.
La répartition de la mémoire linéaire.

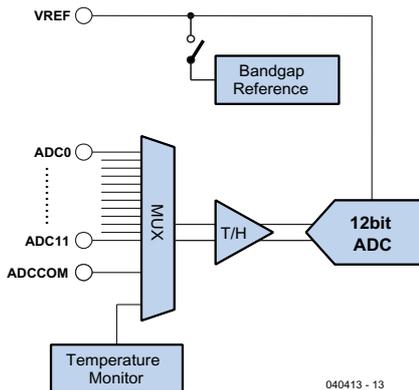


Figure 3.
Le convertisseur A/N et son multiplexeur.

– 0xFFFFFFFF), auquel on accède par des MMR (Memory Mapped Register) réservés, exactement comme pour une mémoire de données. Sur les deux modèles actuellement les plus grands, les ADuC7026 et ADuC7027, on peut, en plus, utiliser un domaine d'adresses externe de 1 Mo (figure 2).

Comme générateur de signaux d'horloge aussi rapides que 45 MHz, pour l'ARM7TDMI lui-même et sa périphérie, un oscillateur RC à 32 kHz et un circuit à PLL a été incorporé à la puce. Il est ainsi possible de réduire la vitesse par l'intermédiaire d'un MMR, en binaire, en huit pas, pour diminuer la consommation, indépendamment des modes Idle et Power Down, jusqu'à 330 kHz. Comme l'oscillateur RC est précis à 2%, dans la plupart des cas le contrôleur peut se passer de circuit d'horloge externe.

La périphérie analogique

Tout l'attrait du microcontrôleur *precision analog* réside évidemment dans sa périphérie analogique, même si le domaine numérique n'est pas en reste. Le noyau de la périphérie analogique est constitué d'un convertisseur A/N qui accomplit un million de conversions par seconde (1 Msps). On apprécie encore davantage la performance quand on exprime qu'il ne lui faut qu'une microseconde pour sortir, de la grandeur analogique qu'on lui applique, un échantillon numérisé de 12 bits de précision. Sur un ADuC7026 (figure 3), on peut brancher jusqu'à douze entrées analogiques en multiplex sur le convertisseur A/N et sur un ADuC7027, le nombre d'entrées monte à 16, sous trois configurations possibles (simple asymétrique, pseudo-différentiel ou symétrique). On peut utiliser comme référence de tension la source interne (10 ppm, 2,5 V) ou une externe (1 à 2,5 V). Le multiplexeur est doté d'un capteur de température intégré.

En parallèle sur les deux entrées analogiques canal 2 et canal 3 (figure 4), il y a un comparateur capable de confronter, au choix, les deux valeurs présentes sur ces entrées ou l'une d'elles à la référence ou encore par rapport à la sortie de l'un des quatre convertisseurs N/A à 12 bits présents sur la puce. On peut le programmer pour réagir au flanc montant ou descendant, avec éventuellement une hystérésis de 7,5 mV imposée par logiciel ou avec un retard (réglable entre 0,5 et 10 μ s) pour filtrer les parasites. La sortie du comparateur est exploitable directement ou peut servir en interne, comme signal d'interruption par exemple.

Les convertisseurs N/A embarqués affichent une résolution de 12 bits et sont dotés d'étages tampons amplificateurs. Comme pour les convertisseurs A/N, ils peuvent utiliser en guise de référence, pour la tension de sortie répartie sur 4 096 pas, la tension d'alimentation, entre 2,7 et 3,6 V (d'habitude 3,3 V), ou une source extérieure dans la même gamme.

La sortie des signaux analogiques se produit selon une chronologie précise. D'habitude, c'est une petite routine d'interruption qui se déclenche à l'issue d'un décompte. Mais comme chaque microcontrôleur a besoin d'un certain temps pour appeler cette routine (la latence d'interruption), il se peut qu'il tombe sur la transition (*jitter*) vers une autre valeur de sortie. C'est pourquoi les nouveaux microcontrôleurs de la série *Precision Analog* sont pourvus d'un tampon supplémentaire, qui peut être déclenché directement par un compteur pour délivrer un signal de sortie bien net, sans recours au filtrage.

La périphérie numérique

Les microcontrôleurs *precision analog* n'ont besoin que de 2,7 à 3,6 V de tension d'alimentation, les processeurs

ARM7TDMI fonctionnent en interne sous 2,5 V seulement, mais les entrées et sorties restent compatibles avec la logique TTL. Le **tableau 1** reprend la liste des constituants de la périphérie numérique. Mentionnons ici que tous les signaux numériques sont programmables bit par bit en entrée ou sortie, multiplexés de manière à ne pas encombrer de broches par des signaux inutilisés. Toutes les E/S de la périphérie numérique sont utilisables pour la commande ou la réception de signaux. En plus des quatre temporisateurs, vous y trouverez une interface asynchrone (UART), une SPI et deux I₂C. Naturellement, il y a aussi des entrées d'interruption, une mise à zéro du contrôleur et une surveillance de l'alimentation. Sur les versions les plus étendues, à partir des ADuC7024, il y a encore une unité très pratique de PWM (modulation en largeur d'impulsion) sous trois phases pour la commande de moteurs électriques ou de composants du même genre.

Mais venons-en à ces petits plaisirs, déjà signalés, que nous réservons au domaine de la périphérie numérique : le réseau logique programmable (PLA). Dans de nombreuses applications, c'est un signal extérieur qui provoque l'interruption. Mais ce signal doit, la plupart du temps, subir une transformation et réclame des fonctions logiques supplémentaires, telles qu'inverseur ou bascule ; ici, *system on chip* s'en occupe.

Pour résoudre simplement ces difficultés et bien d'autres encore, Analog Device a intégré un PLA (**figure 5**), un petit bloc de logique réparti en deux segments et composé de 16 éléments logiques, programmables par MMR. Ces éléments peuvent combiner des signaux d'entrée/sortie, un registre à 16 bits et des signaux internes. La sortie réapparaît sur une borne de sortie logique ou dans un registre de 16 bits du processeur. Outre les quatre entrées d'interruption prévues, il est possible de générer facilement d'autres interruptions par le PLA ou de déclencher le convertisseur A/N. La sortie du comparateur peut également participer aux combinaisons logiques dans le PLA. Les MMR permettent de façonner chaque élément du PLA pour traiter de seize manières différentes les signaux d'entrée.

Les outils de développement

Comme nous avons affaire, avec le système ARM7TDMI, à un processeur déjà largement répandu, il va de soi que de nombreux outils de développement sont disponibles. On peut commencer par se procurer gratuitement sur Internet des outils tels que GNUARM, passer à des environnements de conception de Keil, IAR ou Hitex et aller jusqu'aux outils les plus chers comme Green Hills et ARM lui-même, bien entendu. Si vous cherchez un peu sur Internet, vous y trouverez, à part ceux de ARM, de nombreuses autres possibilités.

Dans le but d'accélérer la diffusion des microcontrôleurs *Precision Analog* et de faciliter leur mise en œuvre, Analog Devices propose à prix intéressant des ensembles complets d'évaluation (**tableau 2**). Ces kits contiennent un CD-ROM avec des versions préparées, faciles à installer, des environnements de développement de Keil et IAR. Le CD-ROM propose aussi de courtes présentations de ces deux environnements. Vous pouvez obtenir gratuitement le CD-ROM complet via le serveur ftp d'Analog Devices.

En plus de ces deux outils complets de développement en version d'évaluation, vous recevrez deux outils supplémentaires :

- un ARMWSD (*ARM Windows Serial Downloader*) pour charger le code objet directement du PC dans la mémoire flash intégrée.

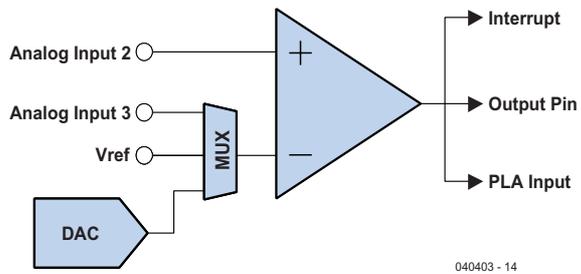


Figure 4. Le comparateur destiné à la surveillance et au déclenchement.

Histoire d'ARM

Il est étonnant de constater que le processeur du contrôleur ARM a déjà plus de vingt ans ! Dans les années 80, aux jeunes années des microprocesseurs comme le Z80, la firme ACORN existait déjà en Angleterre. Elle produisait, tout comme Apple (et Elektor !), un ordinateur personnel basé sur le 6502. Il apparut rapidement que pour des applications graphiques ou en audio, huit bits ne fournissaient pas une puissance de calcul suffisante même en entourant le processeur d'autres puces rapides. C'est ce qui déclencha chez ACORN, en 1983, le développement d'un processeur RISC à 32 bits, qui déjà s'appelait ARM, ce qui signifiait à l'époque ACORN RISC Machine. Il est vrai que, peu après, ACORN a été rachetée par d'autres fabricants de semi-conducteurs, mais en 1985, comme premier concerné par le silicium, ARM1 s'est largement étoffée. Son nom est devenu Advanced RISC Machine, comme la firme actuelle, ARM Ltd. qui propose le processeur en différentes versions pour l'intégrer sous licence à des composants semi-conducteurs.

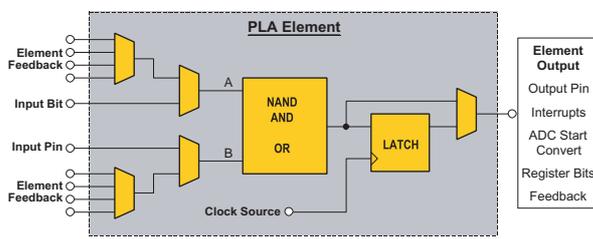
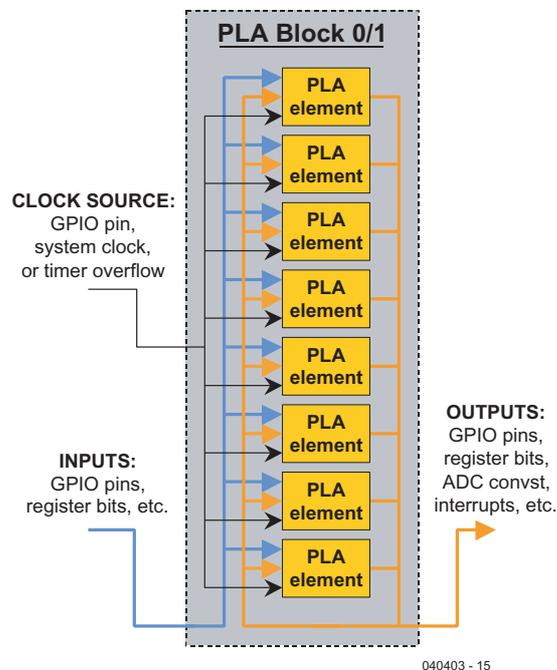


Figure 5. L'un des deux blocs de PLA (a) composés chacun de huit éléments logiques (b).

Tableau 2. Panorama des kits d'évaluation disponibles			
Dénomination	Evaluation-QuickStart-Kit	Evaluation-Board	MiniKit-ADuC7020
Preis	\$ 249	\$ 75	\$ 20
Inhalt	Carte d'évaluation, alim., câble adaptateur RS-232, CD Software et émulateur USB/JTAG	Carte d'évaluation, alim., câble adaptateur RS-232 et CD Software	Platine à 40 broches au format DIL, câble adaptateur RS-232 et CD Software
Versions disponibles	EVAL-ADuC7020QS EVAL-ADuC7024QS EVAL-ADuC7026QS	EVAL-ADuC7020EB EVAL-ADuC7024EB EVAL-ADuC7026EB	MiniKit-ADuC7020
* Disponible auprès des distributeurs autorisés d'Analog Devices ou directement auprès d'Analog Devices via Internet			

- le PLATool, un petit programme graphique de saisie pour réaliser la configuration du PLA. Un assembleur ou un programme C permet avec lui d'initialiser ou de générer aisément les valeurs à introduire dans les MMR du PLA.

(040403-1)

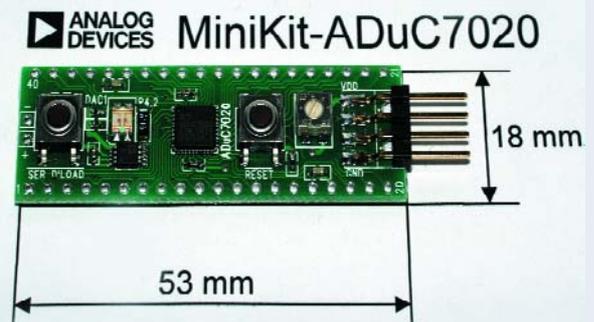
Adresses Internet et littérature :

Documents concernant le noyau de l'ARMTDMI :
 ARM Architecture Reference Manual [ARM DDI0100E]
 ARM7TDMI Datasheet [DDI0029E Part 1 to 3]
 ARM7TDMI Technical Reference Manual
 [DDI00210B_7TDMI_R4_trm]

Precision Analog Microcontroller sur Internet :
www.analog.com/MicroConverter/ARM7/
 Téléchargement CD-ROM d'évaluation :
ftp://ftp.analog.com/pub/MicroConverter/ADuC_Beta702xV05
 ISAC, Elektor n° 280, octobre 2001 et suivants
 GNU ARM Tools pour Cygwin, Linux et MacOS :
www.gnuarm.org
 IDE mVision3 de Keil : www.keil.com
 Embedded Workbench IDE de IAR : www.iar.com
 HITEX : www.hitex.de
 Green Hills Tools : www.ghs.com
 Macraigor Systems LLC : www.macraigor.com/wiggler.htm
 Questions à l'auteur (en anglais) : Michael.Mueller-Aulmann@analog.com

Le MiniKit ADuC7020

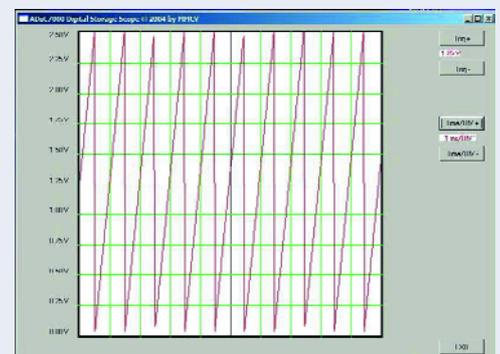
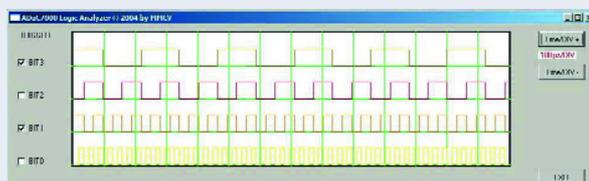
Une introduction dans le monde des ARM7TDMI à 16/32 bits, voilà ce que propose Analog Devices pour 20 \$ avec le MiniKit ADuC7020. La petite platine est organisée de manière à permettre le branchement de ses deux rangées de connecteurs à 20 bornes sur un support DIL (Dual in Line) de 40 broches. On peut raccorder une tension d'alimentation comprise entre 3,5 et 12 V aux deux bornes du côté gauche. Après connexion au PC et installation du logiciel on peut déjà se lancer dans les premiers exemples. Comme il n'y a pas d'émulateur JTAG joint et que tous les lecteurs d'Elektor n'en possèdent pas nécessairement à l'heure actuelle, il est possible, avec l'outil de travail embarqué (Embedded Workbench) et le moniteur de ROM, de passer au développement confortablement par le truchement du câble d'interface RS232 du PC.



Applications

L'amalgame de convertisseurs A/N et N/A précis, d'un processeur rapide et d'une logique programmable sur une seule puce vise essentiellement à simplifier les réalisations dans les domaines de la mesure, de la commande et de la régulation. On pense spécialement au rôle qu'il peut jouer dans l'automatisation de la production ainsi qu'en électronique automobile. Dans la technique de la communication, les nouveaux MicroConverters d'Analog Devices sont utilisés sur les réseaux optiques, où ils servent à transmettre les mesures de niveau des signaux de réception et à commander en conséquence ceux d'émission.

Leur capacité à appréhender les signaux de capteurs sous haute résolution, de les traiter vite et d'en assurer la régie avec précision en font aussi des auxiliaires de choix dans le monde de la robotique.



L'auteur a développé, à l'aide du Minikit ADuC7020, un projet en technique de mesure. Il l'a transformé en oscilloscope numérique à mémoire avec une fréquence d'échantillonnage de 1 Msps sous une résolution de 12 bits, combiné à la fonction d'analyseur logique. Vous en apprendrez davantage sur cette réalisation bientôt dans ce magazine.

Microcontrôleurs

« SHOPPING » VIA INTERNET

Thijs Beckers

Que peut-on faire avec un microcontrôleur ? Quel type faut-il utiliser pour une application donnée ? Où puis-je le trouver ? Des questions et encore des questions auxquelles il est difficile de donner une réponse. Nombre de fabricants sont prêts à donner un coup de pouce.

The screenshot shows the National Semiconductor website's parametric product selection tool. The page title is "Parametric Product Selection Table". It features a search bar at the top and a table of microcontroller products. The table has columns for "Device", "Title", "Price", "Package", and "Pin". The "Device" column includes part numbers like C18494A313, C18494A32, C18494A33, etc. The "Title" column describes the product families, such as "Family of 10-bit CAN enabled CompactRISC Microcontrollers". The "Price" column shows values like \$5.0, \$11.0, \$10.0, etc. The "Package" column lists types like PQFP, PLCC, and MLP. The "Pin" column shows pin counts like 80, 181, 14, etc. There is also a "Filter Results By:" section on the left with various criteria like "Supply Max", "Supply Min", "Power on Reset", "Upgrading Frequency", "I/O", "IO", "IO", "IO", "IO".

Device	Title	Price	Package	Pin
C18494A313	Family of 10-bit CAN enabled CompactRISC Microcontrollers	\$5.0	PQFP	80
C18494A32	Family of 10-bit CAN enabled CompactRISC Microcontrollers	\$11.0	PQFP	80
C18494A33	Family of 10-bit CAN enabled CompactRISC Microcontrollers	\$10.0	MLP	181
C18494A34	Family of CompactRISC 10-Bit Microcontrollers	\$10.0	PLCC	14
C18494A35	Family of CompactRISC 10-Bit Microcontrollers	\$11.0	PLCC	14
C18494A36	Family of CompactRISC 10-Bit Microcontrollers	\$5.0	PQFP	80
C18494A37	Family of CompactRISC 10-Bit Microcontrollers	\$11.0	PLCC	14
C18494A38	Family of CompactRISC 10-Bit Microcontrollers	\$10.0	MLP	44
C18494A39	Family of CompactRISC 10-Bit Microcontrollers	\$10.0	PLCC	14
C18494A40	8-Bit CMOS Flash Microcontroller with 8k Memory Dual Op Amps Virtual EEPROM Temperature Sensor 10-Bit A/D and 1.17 to 1.5V Brownout Reset	\$4.0	SOIC-WR8	24
C18494A41	8-Bit CMOS Flash Microcontroller with 8k Memory Virtual EEPROM 10-Bit A/D and 2.7V to 2.0V Brownout Reset	\$4.0 - \$4.45	MLP PLCC TSSOP	44 41
C18494A42	1408 130C11 1.5k Microcontroller with 32k Memory Virtual 11981M 11450 A/D and 2.7V to 2.0V Brownout	\$4 - \$4.5	MLP PLCC TSSOP	14 18 20 20 20
C18494A43	8-Bit CMOS Flash Microcontroller with 8k Memory Virtual EEPROM 10-Bit A/D and 1.17V to 1.5V Brownout Reset	\$7.1 - \$6.0	MLP PLCC TSSOP	44 41
C18494A44	1408 130C11 1.5k Microcontroller with 32k Memory Virtual 11981M 11450 A/D and 4.3V to 4.0V Brownout	\$4 - \$4.5	MLP PLCC TSSOP	14 18 20 20

« Tout ce que l'on peut imaginer », « Au choix, cela dépend de ce que vous voulez faire », « Commander en ligne, échantillons, dans le magasin du coin... » sont quelques-unes des réponses possibles d'une personne sachant de quoi il retourne dans ce monde. Mais elles ne

vous sont que d'un piètre secours. Et maintenant ? Ailleurs dans ce numéro, un article de Mr Tavernier mentionne les aspects auxquels il faut veiller lors du choix d'un microcontrôleur. Nous vous proposons ici quelques adresses Internet pratique à utiliser pour vous aider dans le

choix du bon microcontrôleur. Il n'existe pas encore de site (gratuit) où vous pouvez entrer tous les paramètres requis par une application spécifique et qui vous donne alors le type de processeur, son fabricant et son distributeur... Chez **EEM ICmaster** [1] on peut, sous l'onglet « *micro-controller* » effectuer des recherches paramétriques par fabricant. Il faut vous inscrire, mais c'est gratuit et nous n'avons pas encore eu de Pourriel.

National Semiconductor [2] offre une possibilité sur Internet de choisir un microcontrôleur dans un tableau de produits. Il faut ensuite choisir l'application ce qui se traduit par l'apparition d'une page sur laquelle on peut faire un choix sur la base des paramètres du composant souhaité. On a la possibilité, sur le site de **Keil Software** [3], de faire une sélection paramétrique des composants supportés par les logiciels de cette société. On pourra y trouver, par exemple, un ARM de spécifications données.

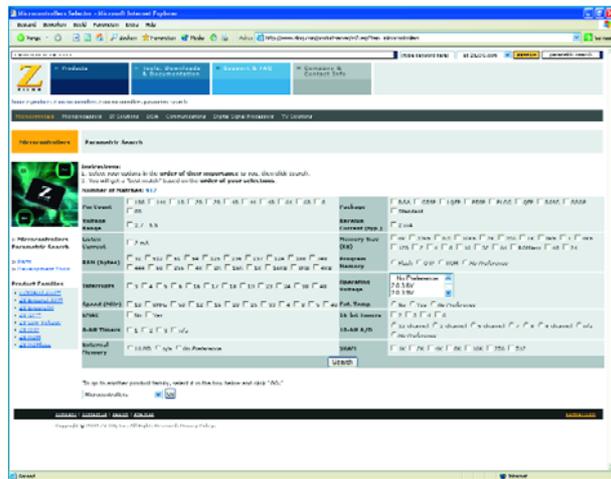
Le site de **Philips** [4] propose une présentation sous forme de catalogue. La recherche peut se faire sur les spécifications par type de composant, de chez Philips bien entendu. Il n'y a malheureusement pas de recherche paramétrique.

Renesas [5], société créée par Hitachi et Mitsubishi Electric, offre une possibilité de recherche paramétrique. En haut à droite de la page on voit un onglet « *parametric search* ». Par son intermédiaire on arrive à une page de sélection de catégorie de produit. Nous optons ici pour la catégorie de produits MPU/MCU. Il est alors proposé un champ dans lequel on peut entrer tous les paramètres souhaités.

Zilog [6] aussi offre une possibilité de choix paramétrique dans sa base de données produits, de même d'ailleurs que **Freescale Semiconductor** [7], une société faisant partie de Motorola. Chez Freescale, la possibilité de recherche paramétrique se trouve également en haut à droite. Nous optons ici aussi, bien évidemment, pour les microcontrôleurs (8, 16 ou 32 bits). On voit apparaître une liste dans laquelle il suffit de cocher les options de recherche souhaitées. Si l'on coche toutes les cases on obtient bien évidemment le plus de réponses possibles. Il est possible (enfin), sur la page qui apparaît ensuite de remplir toutes les options souhaitées pour découvrir un numéro de type d'un contrôleur répondant au cahier des charges.

Infineon [8], une société connue des amateurs d'ordinateurs propose aussi des microcontrôleurs dans son assortiment. Après une présélection de microcontrôleurs à 8, 16 ou 32 bits, il apparaît une page sur laquelle on peut choisir nombre de paramètres.

Texas Instruments [9] propose aussi un choix riche de microcontrôleurs. On opère, par le biais de « *Microcontrollers* » (dans la case bleue « *Products* »), un choix de plusieurs familles de microcontrôleurs. Sur la page qui s'ouvre ensuite il est possible d'obtenir plus d'informations sur diverses familles et de rechercher un microcontrôleur donné. Il vaut sans doute la peine de faire un petit tour dans la section « *Technical documents* » (à gauche). Nous vous proposons, pour finir, un lien vers le site Internet de **Wichit Sirichote** [10], sur lequel sont décrits un certain nombre de réalisations intéressantes qui pourraient servir de déclencheur pour de nouveaux projets.



Adresses Internet

- [1] EEM ICmaster: www.icmaster.com/Expertsearch.aspx
- [2] National Semiconductor: www.national.com/catalog
- [3] Keil Software: www.keil.com/dd/param_search.asp
- [4] Philips: www.semiconductors.philips.com/markets/mms/products/microcontrollers/index.html
- [5] Renesas: www.renesas.com/homepage.jsp
- [6] Zilog: www.zilog.com/productserver/ss2.asp?fam=microcontrollers
- [7] Freescale: www.freescale.com
- [8] Infineon: www.infineon.com/cgi/ecrm.dll/ecrm/scripts/search/parametric_search_overview.jsp
- [9] Texas Instruments: www.ti.com/
- [10] Wichit Sirichote: www.kmitl.ac.th/~kswichit/

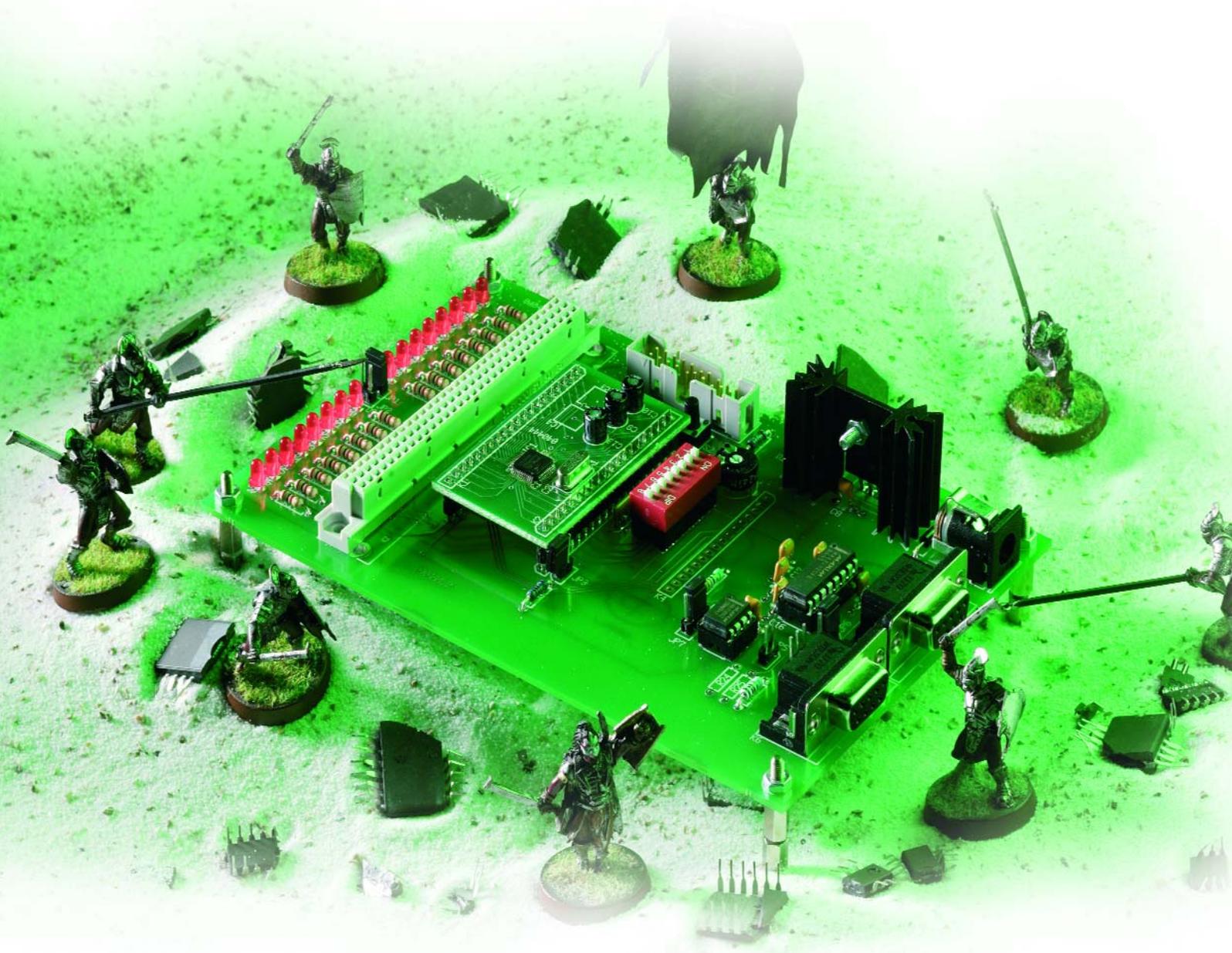
(045118-1)

Carte de développement

2^{ème} partie :

Réalisation et programmation d'« ARMÉE

Tony Dixon



ent LPC210x 'ARMee'



En avez-vous assez de ces PIC, AVR et autres 8051 utilisés par Mr-Tout-Le-Monde ? Nous avons abordé, dans l'article du mois dernier, l'ARM, une famille des microcontrôleurs 32 bits; le joyau des composants passés en revue est le LPC210x de Philips. Passons aux choses sérieuses: voici « **ARMÉE** », une carte de développement ARM puissante à réaliser et programmer soi-même.

Au cours des ans, Elektor a décrit dans ses colonnes un certain nombre de cartes de développement à microcontrôleur. Le présent article est le dernier d'une longue lignée de tels articles. La majeure différence du présent article et du projet qu'il décrit est que cette carte repose non pas sur un **microcontrôleur** 8 bits tels qu'un PIC [1], un AVR [2] ou un 8051 [3] mais sur un **32 bits** : un membre de la famille **ARM**, le LPC210x de Philips Semiconductor. Ce composant intègre tout ce dont peut rêver un microcontrôleur (cf. l'encadré « **Caractéristiques principales de ARMÉE & LPC210x** »). Mettons immédiatement les pieds dans le plat en signalant qu'il ne possède pas d'interface de conversion Analogique/Numérique (CAN) ni d'interface de bus externe. Le LPC210x est un vrai microcontrôleur sans interface de bus externe pour extension de la mémoire par puces Flash ou SRAM externe ou l'adjonction de périphériques externes.

Le matériel

On retrouve en **figure 1** l'électronique de notre carte de développement. Son alimentation pourra se faire à l'aide d'un adaptateur secteur fournissant entre 9 et 15 V continus. On pourra se contenter d'un courant de quelque 500 mA pour alimenter la carte proprement dite et tous autres périphériques

connectés à la carte.

Sur la carte-gigogne du processeur, des régulateurs de tension fournissent les tensions nécessaires au LPC210x : 1,8 V pour son noyau CPU et 3,3 V pour ses périphériques et les Entrées/Sorties (E/S). Le LPC210x a des E/S supportant 5 V et capables d'attaquer directement de la logique TTL 5 V.

Processeur moderne, le LPC210x possède un circuit de réinitialisation interne (Reset), de sorte qu'il ne faut rien de plus qu'une résistance de forçage au niveau haut (pull up) externe pour terminer le schéma. Le bouton-poussoir S2 permet un reset manuel le cas échéant.

K5 représente la communication sérielle RS-232, l'interface correspondante reposant sur IC1, un MAX3232, cet émetteur/récepteur mettant à disposition 2 étages de commande et 2 récepteurs. Le MAX3232 est alimenté sous 3,3 V, des pompes de charge embarquées fournissant les + et -12 V requis par l'interface RS-232. En option, la carte peut être dotée d'une seconde interface RS-232 au niveau de l'embase K1, IC1 fournissant les seconds étage de commande et récepteur. La validation du second port requiert l'implantation en position correcte des cavaliers JP6 et JP7.

Si l'on n'a pas besoin de 2 interfaces RS-232, l'embase K6 peut fournir une interface RS-485 compatible broche à

broche Modbus. L'interface RS-485 est pilotée par IC2, un MAX3082, cet émetteur/récepteur RS-485 de Maxim étant lui aussi alimenté en 3,3 V. Les cavaliers JP6 et JP7 seront positionnés comme nécessaire pour une validation du port RS-485 au lieu du second port RS-232.

Nous le disions, l'interface RS-485 est compatible Modbus. Cependant, Modbus utilise la broche 5 de l'embase sub-D 9 points en tant que ligne de signal alors qu'en RS-232, cette même broche est forcée à la masse. Il faudra, pour contourner ce problème, implanter le cavalier JP4 pour que la broche 5 fasse office soit de signal RS-485 soit de masse de signal en RS-232.

L'embase K4 offre la possibilité de connecter un affichage LCD alphanumérique standard à la carte. L'interface LCD partage certaines des broches d'E/S du LPC210x avec un octuple interrupteur DIL. Pour pouvoir utiliser l'affichage, tous les contacts de cet interrupteur 8 points doivent être ouverts (Off). L'ajustable P1 permet de régler le contraste de l'affichage. Le signal LCD R/W en broche 5 de K4 est relié au 0 V pour éviter que le programme ne prenne en compte quelque donnée que ce soit en provenance de l'affichage. Pour cela, il faut qu'un programme imbrique une temporisation rudimentaire pour permettre à l'affichage de traiter la dernière instruction

Caractéristiques principales de ARMÉE & LPC210x

LPC210x (2104/2105/2106) :

- noyau processeur 16/32 bits ARM7TDMI-S
- 128 Koctets de mémoire de programme Flash
- 16/32/64 Koctets de mémoire de données SRAM
- ISP et IAP
- E/S numériques; 2 UART
- Interfaces SPI et I²C
- timer capture et sorties MLI (PWM)
- Interface de débogage JTAG

Système de développement ARMÉE :

- En lignée directe avec d'autres cartes de développement publiés précédemment dans Elektor

- Présence d'un module processeur détachable doté dès le départ d'un LPC2106
- Visualisation par LED de l'état des ports
- Octuple interrupteur DIL pour commutation des entrées
- Connecteur pour branchement d'un affichage LCD alpha-numérique standard
- Connexion au monde extérieur par une paire d'embases sub-D 9 points : 2 x RS-232 ou 1 x RS-232 et 1 x RS-485 compatible broche à broche Modbus
- Connecteur DIN41612 d'E/S sur la carte-mère
- Format Eurocard 160 x 100 mm
- Compilateurs « C » GNU GCC multi-plate-formes gratuits
- Sélection étoffée de logiciels de développement ARM gratuits et commerciaux

reçue. Les lignes d'E/S numériques P0.22 et P0.23 commandent respectivement les signaux RS et E de l'affichage. Si l'affichage est doté d'un rétro-éclairage intégré, on pourra l'alimenter par le biais de la résistance-talon R22. La carte comporte une série de 16 LED, D1 à D16. Les LED sont reliées aux 16 premières lignes d'E/S numériques du LPC210x et peuvent servir d'indicateur d'état voire de moniteur d'activités périphériques telles que trafic au niveau des UART. La LED D1 est reliée à P0.0, D2 à P0.1 et ainsi de suite. À noter en outre que les LED partagent les broches d'E/S du LPC210x avec d'autres fonctions telles qu'UART0 et UART1.

Les LED sont subdivisées en 2 groupes de 8. Le premier groupe, D1 à D8, est validé par le cavalier JP1 qui les relie au 0 V; JP2 assure la même fonction pour le second groupe, D9 à D16, les mettant également au 0 V.

Comme mentionné, l'octuple interrupteur DIL S1 partage les mêmes broches d'E/S du PLC210x que l'embase de l'affichage LCD; il faudra partant ne l'utiliser que s'il n'y a pas de module LCD connecté au système. Vous pouvez utiliser ce bloc interrupteur à des fins de débogage ou en tant que dispositif de configuration.

La carte-mère est dotée d'un connecteur DIN41612 A+C, K2, qui a pour fonction de permettre la connexion au système de développement d'une circuiterie externe. Ce connecteur véhicule les lignes d'alimentation +5, +3,3 V, l'horloge, les signaux de reset et la totalité des 32 signaux d'E/S du LPC210x. Le connecteur véhicule également les

signaux pour des interfaces I²C et SPI distinctes. Il est à souligner que dans le cas du LPC210x, une broche d'E/S peut avoir plusieurs fonction, ici la broche d'E/S peut se trouver partagée avec l'UART, I²C, SPI voire d'autres signaux. À remarquer que les LED embarquées, l'interrupteur S1 et l'em-

base LCD partagent également des signaux d'E/S au niveau du connecteur DIN41612.

Les platines

La figure 2 montre le dessin des pistes de la carte-mère et de la carte-gigogne.

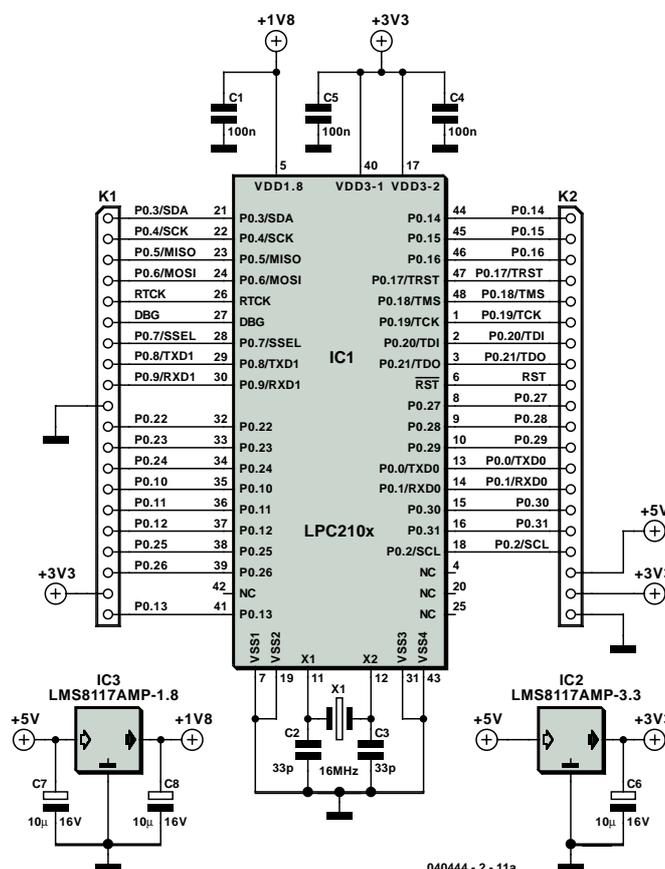


Figure 1. Schéma du système de développement ARMÉE. À remarquer la possibilité de sortir le module du processeur !

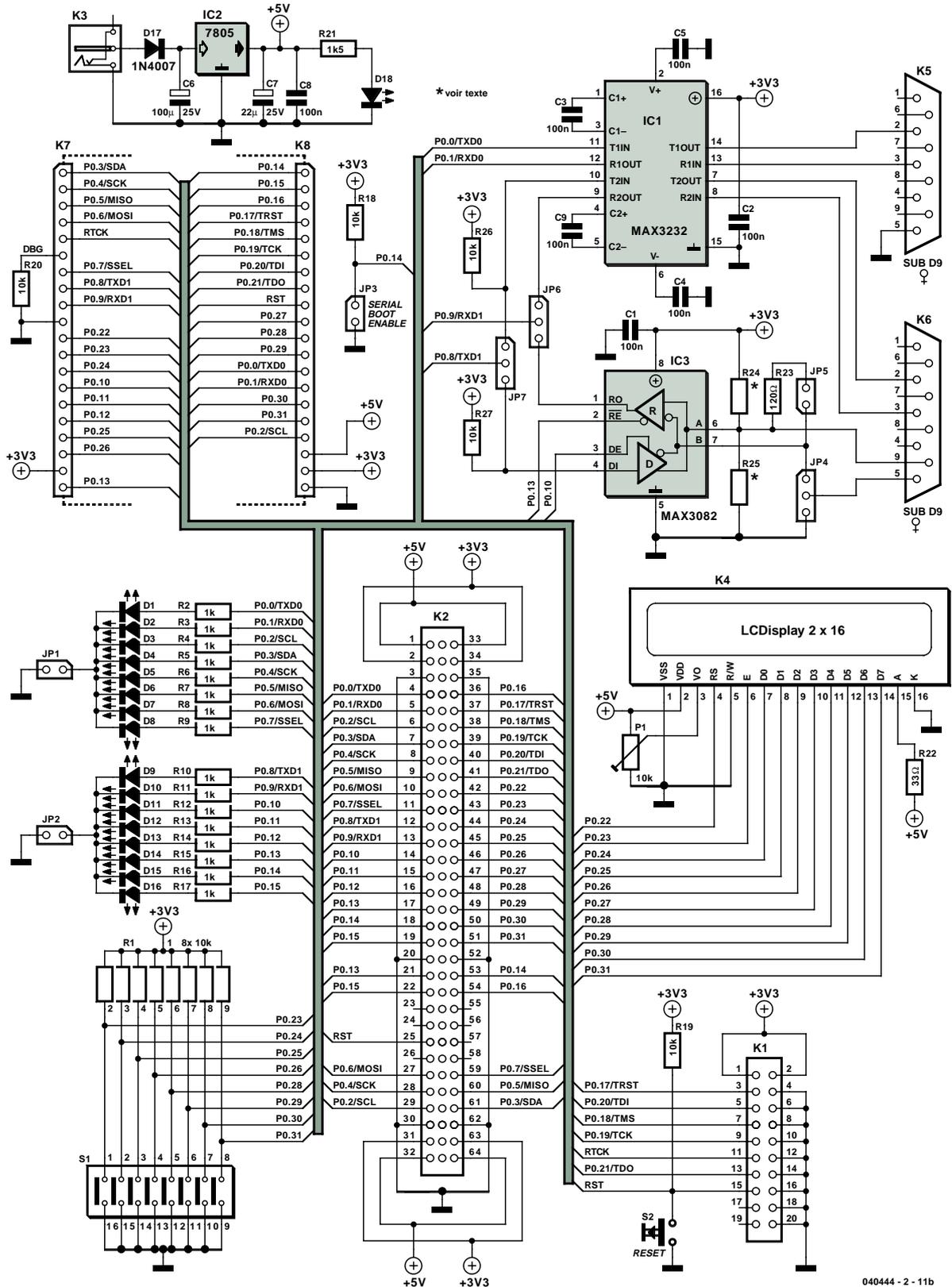
La carte-gigogne est disponible soit montée, **EPS040444-91** (avec PLC2106 monté) soit vierge, **EPS040444-1** auprès des adresses habituelles (et notre site).

Le LPC210x est un composant CMS à 48 broches à souder à même la platine. La platine **040444-1** vierge est pré-étamée. Positionner soigneusement IC1

sur les îlots CMS et souder 2 pattes diamétralement opposées à l'aide d'un fer à soude à pointe fine. Si vous êtes satisfait du positionnement du processeur, il restera à souder le reste de ses pattes. Veillez à effectuer de bonnes soudures. Ne vous inquiétez pas trop d'un excès de soudure qui court-circuiterait des broches. Utilisez de la tresse

à dessouder pour supprimer par absorption de la soudure les court-circuits de soudure créés lors de l'opération précédente. Une fois le pont éliminé, ressoudez avec soin avec votre fer à souder à pointe fine.

La carte-mère vierge est disponible sous la dénomination **EPS040444-2**.



040444 - 2 - 11b

Attention, lors de l'implantation des composants, à la polarité des diodes, des condensateurs électrochimiques et des LED. IC1 et IC3 de la carte-mère pourront être montés sur supports. Une fois tous les composants montés, une ultime inspection visuelle permettra de comparer le fruit de vos efforts avec notre prototype fonctionnel représenté en **figure 3**; on pourra mettre la carte sous tension.

Outils de programmation

Le LPC210x est aisément programmable en langage C. Il existe dans le com-

merce plusieurs compilateurs C pour l'architecture ARM (cf. liens Internet). Si vous ne voulez pas acheter de compilateur C du commerce vous pouvez utiliser un compilateur C GNU. Il s'agit d'un compilateur C open-source que nombre d'utilisateurs considèrent comme étant du niveau de la plupart de ses concurrents commerciaux. Le

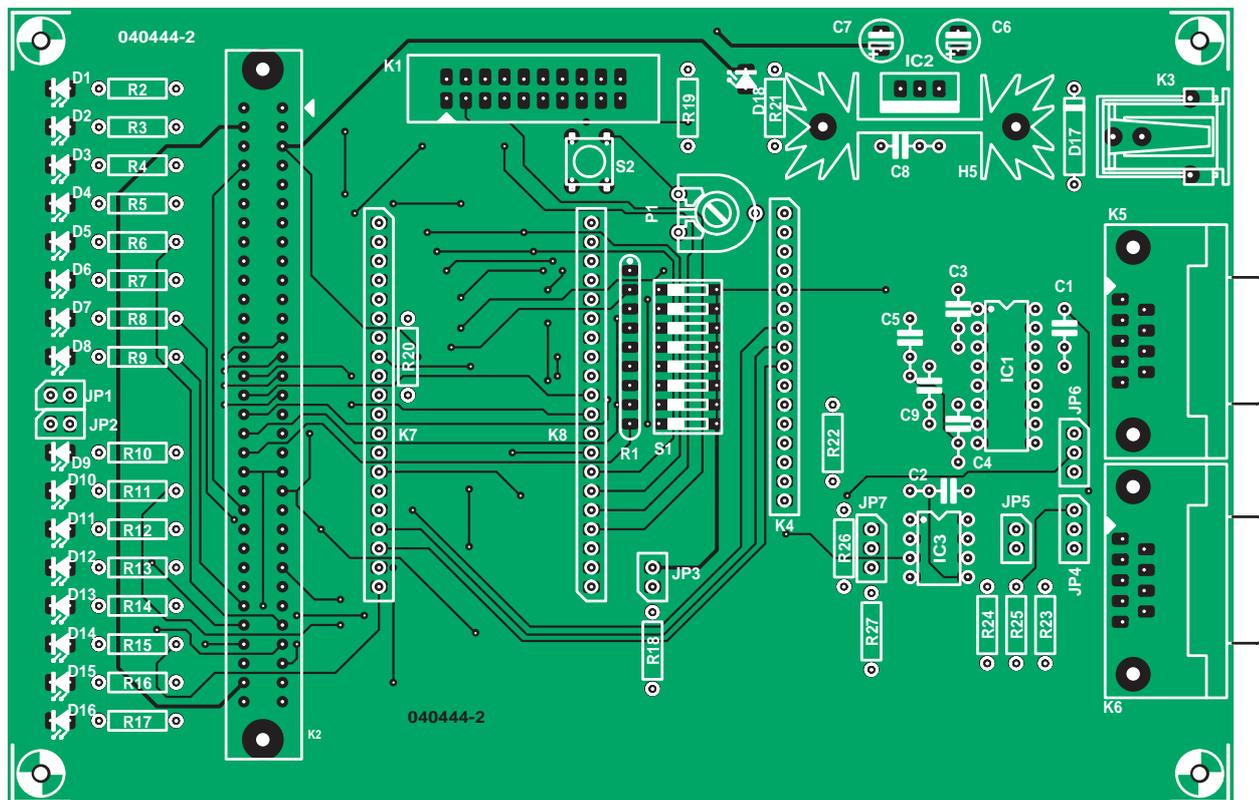


Figure 2a. Dessin recto-verso de la carte-mère, une platine de taille Eurocard.

Liste des composants

de la carte-mère d'« ARMÉE »

Résistances :

R1 = réseau SIL de 8 résistances de 10 k Ω
 R2 à R17 = 1 k Ω
 R18 à R20, R26, R27 = 10 k Ω
 R21 = 1 k Ω
 R22 = 33 Ω
 R23 = 120 Ω
 R24, R25 = cf. texte
 P1 = ajustable 10 k Ω

Condensateurs :

C1 à C5, C8, C9 = 100 nF
 C6 = 100 μ F/25 V radial

C7 = 10 μ F/25 V radial

Semi-conducteurs :

D1 à D16, D18 = LED 3 mm faible courant
 D17 = 1N4007
 IC1 = MAX3232CPE (Maxim-Dallas)
 IC2 = 7805
 IC3 = MAX3082CP (Maxim-Dallas)

Divers :

JP1, JP2, JP3, JP5 = embase autosécable à 1 rangée de 2 contacts + cavalier
 JP4, JP6, JP7 = embase autosécable à 1 rangée de 3 contacts + cavalier
 K1 = embase HE-10 à 2 rangées de 10 contacts
 K2 = connecteur DIN41612AC @a 2 rangées de 32 contacts

K3 = embase jack d'alimentation encartable

K4 = module LCD à 2 x 16 caractères

K5, K6 = embase sub-D 9-femelle en équerre encartable

K7, K8 = connecteur SIL à 1 rangée de 20 contacts

S1 = octuple interrupteur DIP

S2 = bouton/poussoir encartable tel que, par exemple, DTS6

Radiateur SK104 25,4 STC (Fischer)

Platine (**EPS040444-2**)

Disquette avec programme (LEDTest),

(**EPS040444-11**) ou téléchargement gratuit

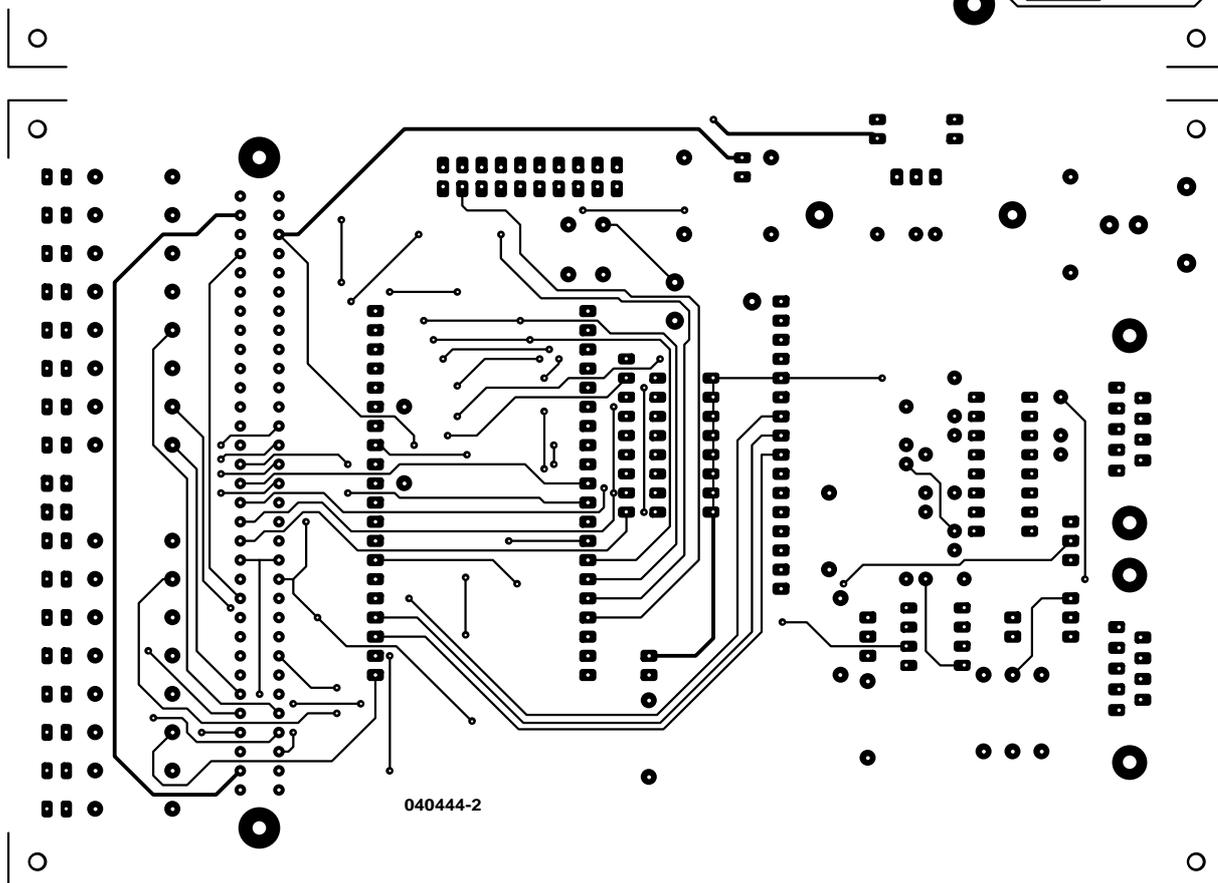
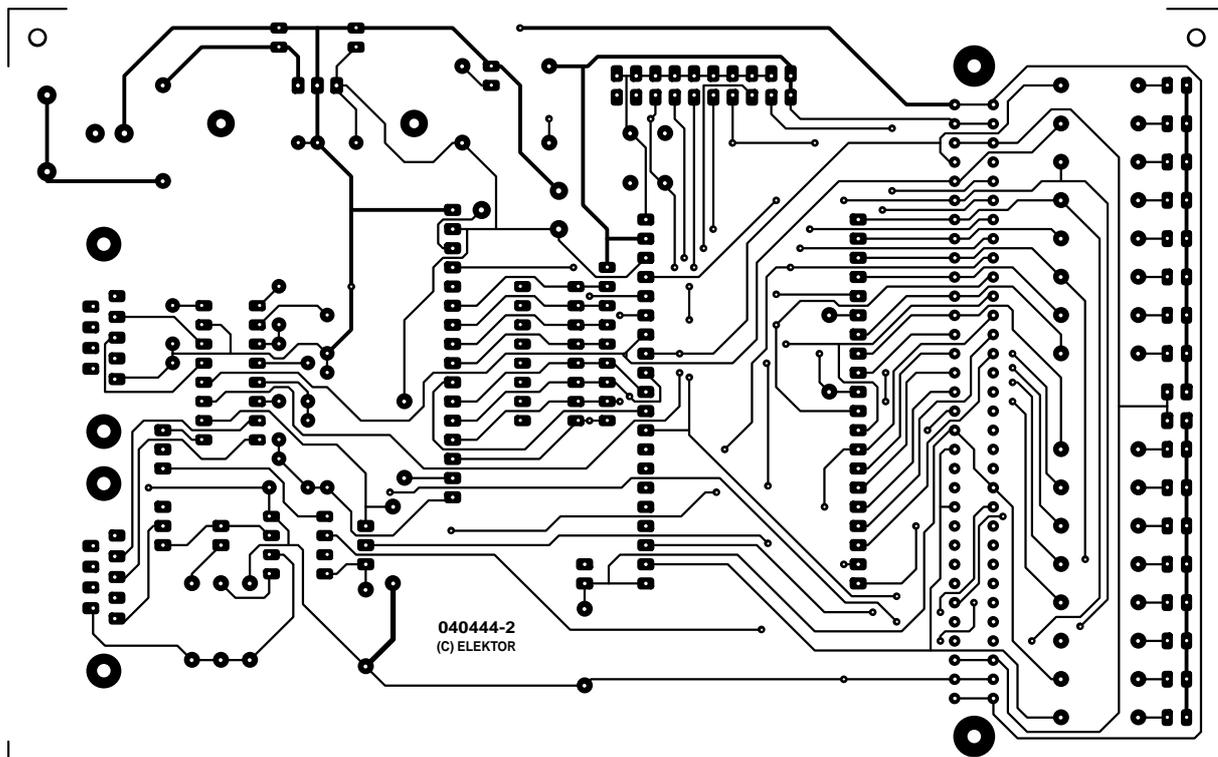
revers de la médaille est que l'on ne dispose pas, avec le compilateur GNU, d'un processus d'installation aussi peaufiné que celui d'un compilateur du commerce.

Le processeur LPC210x lui-même se laisse programmer par le biais de l'interface JTAG embarquée ou le SBL (*Serial Boot Loader*) interne.

Go JTAG !

L'interface JTAG peut servir à bien autre chose qu'à uniquement programmer le composant. Elle peut être utilisée pour déboguer un programme en cours d'exécution sur le LPC210x. Il faut, pour utiliser l'interface JTAG, intercaler un module de débogage

JTAG entre le système de développement à LPC210x et le PC-hôte. Il existe dans le commerce différents modules d'interface JTAG à prix abordable, de même que des modèles non commerciaux. Une petite recherche sur Internet, ou un clic sur les liens Internet donnés en fin d'article devraient vous aider dans cette quête.



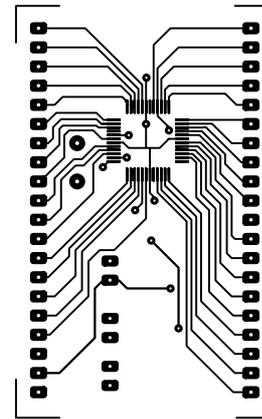
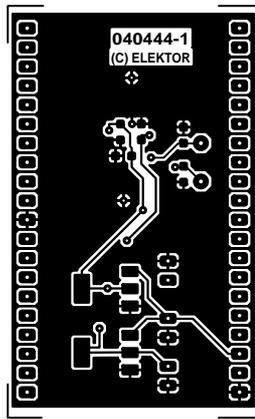
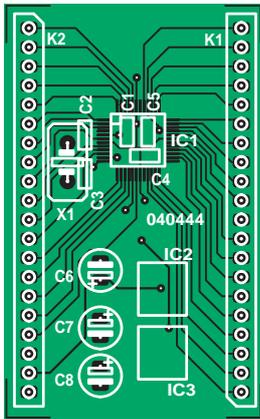


Figure 2b. Dessin des pistes et sérigraphie de la carte processeur gigogne du processeur LPC210x.

Liste des composants

de la carte gigogne à LPC210x

Condensateurs :

C1, C4, C5 = 100 nF, forme 0805
 C2, C3 = 33 pF, shape 0805
 C6 à C8 = 10 µF/16 V radial

Semi-conducteurs :

IC1 = LPC210x (2104/2105/2106)
 IC2 = LMS8117AMP-3.3
 IC3 = LMS8117AMP-1.8

Divers :

K1, K2 = connecteur SIL à 1 rangée de 20 contacts
 X1 = quartz 16 MHz
 PCB (vierge = **EPS 040444-1**)
 PCB (dotée de son LPC2106 = **EPS 040444-91**)

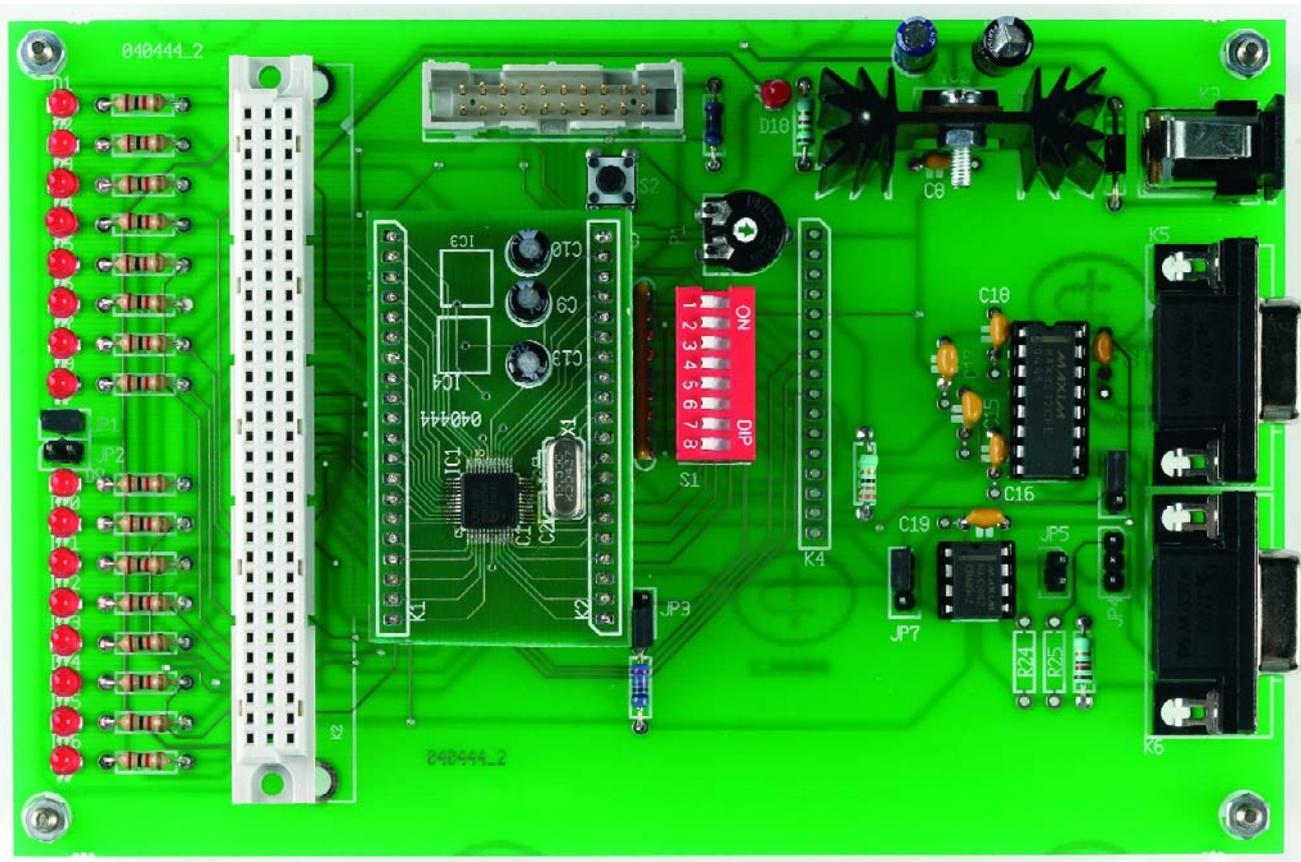


Figure 3. Exemple terminé et testé de notre carte de développement ARMÉE. Le module processeur peut être transféré vers l'application-cible !

Si l'on n'a que faire du débogage JTAG, on pourra programmer le LPC210x par le biais du SBL embarqué. Celui-ci permet d'utiliser à d'autres fins les 5 lignes d'E/S numérique requises par l'interface JTAG.

Serial Boot Loader

L'utilisation du SBL requiert l'implantation du cavalier JP3. Le cavalier JP3 force la ligne P0.14 à la masse et la libération du LPC210x de l'état reset le force à exécuter un programme SBL interne depuis les 8 Koctets supérieurs de sa mémoire Flash. L'utilisation d'un programme PC gratuit mis à disposition par Philips Semiconductor (**fig-**

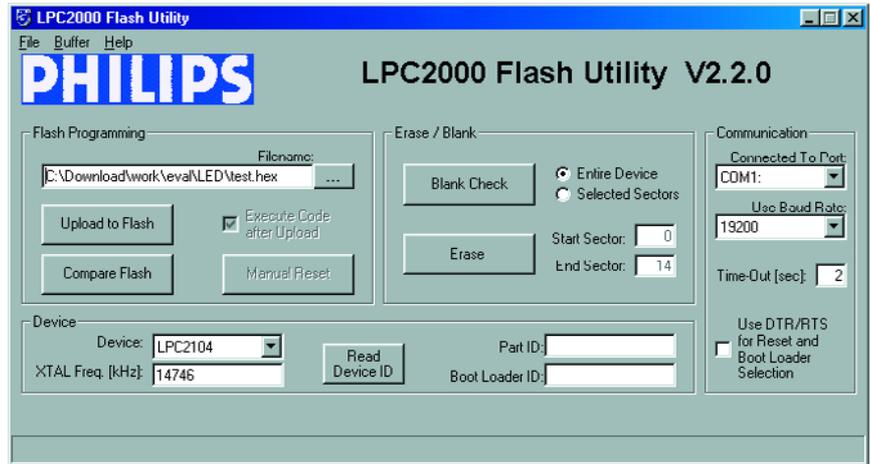


Figure 4. Flash Downloader de Philips.

SIOB - Simple Input Output Bus

Comme le laisse supposer son nom, SIOB est un bus d'extension le plus simple que l'on puisse imaginer. 32 signaux d'E/S numériques sont portés vers un connecteur DIN41612 embarqué et mis ainsi à disposition en tant que bus d'extension. Les lignes d'E/S numériques peuvent être des entrées ou des sorties selon le type d'interface d'extension requis. Le programmeur de la carte de développement est responsable de la configuration du registre de sens de données (DD, Data Direction) du micro-contrôleur en fonction de ses besoins en E/S numériques.

Outre les signaux d'E/S numériques, SIOB offre une possibilité d'extension sérielle sous la forme d'une interface SPI et I²C. On trouve également sur le bus quelques signaux SPI de sélection de puce (CS). De par la présence des interfaces SPI et I²C, il devient possible d'ouvrir le système de développement vers l'extérieur. On pourrait, par exemple, ajouter des convertisseurs A/N et N/A à base I²C ou des E/S numériques pilotées SPI.

Le bus comporte, pour finir, les lignes d'alimentation +5 et +3,3 V, le signal d'horloge, les signaux RESET et $\overline{\text{RESET}}$.

Une dernière note concernant SIOB : les broches de signal du LPC210x ne servent pas uniquement d'E/S numériques mais remplissent également d'autres fonctions telles que UART, I²C, SPI, MLI (PWM), Timer Capture, débogage JTAG. Si l'une des broches de signal sert à piloter une interface UART, elle n'est plus utilisable en tant que signal d'E/S numérique et partant devient indisponible pour le SIOB. Si un système requiert l'utilisation des interfaces UART, du débogage JTAG et de l'interface LCD, il ne restera que très peu de signaux d'E/S de disponible pour le SIOB. Dans ces conditions, on utilisera de préférence les interfaces I²C ou SPI présentes sur le SIOB pour étendre le système.

ure 4) et d'un câble sériel pris entre un PC-hôte et l'UART0 du PLC210x (port RS-232 n°1 de l'« ARMÉE ») permet d'effectuer la programmation du LPC210x.

Utiliser GNU GCC avec les processeurs ARM

« Pourquoi utiliser un compilateur GCC GNU ? » pouvez-vous vous demander ? Ces outils sont, par essence et la méthode de suivi communautaire, fournis gratuitement sur le site Internet de FSF *Free Software Foundation*. Ces outils sont mis à disposition par le projet GNU selon les termes de la licence *GNU general public*. Historiquement, les logiciels de GNU/FSF étaient écrits pour le système d'exploitation UNIX et GNU est l'acronyme de « GNU's Not UNIC », GCC signifiant « GNU Compiler Collection ».

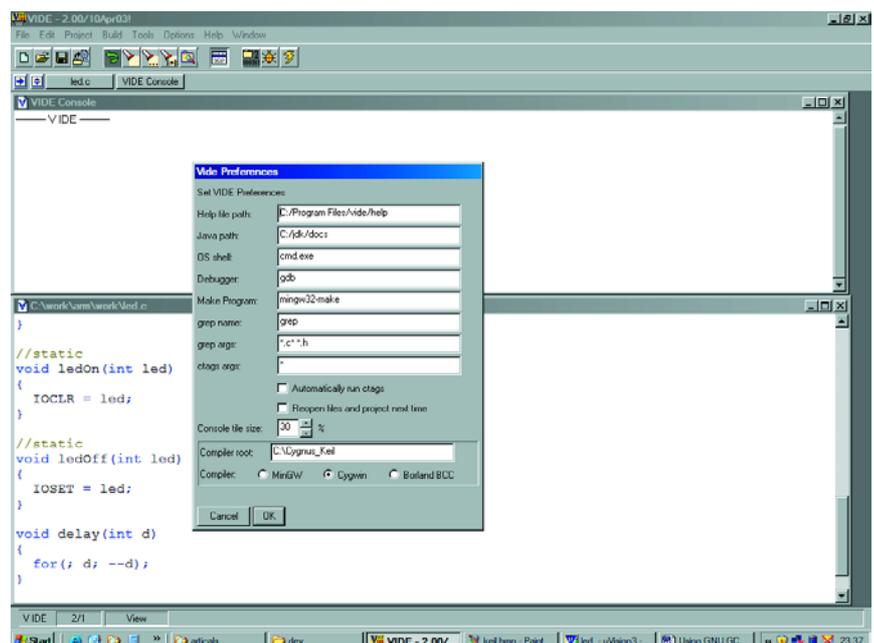


Figure 5. L'Editeur du programmeur VIDE.

Autres binutils pour ARM.

addr2line	Convertit les adresses en noms de fichier et numéros de lignes.
ar	Utilitaire d'archivage (création, modification et extraction d'archives)
gprof	Affiche information de profil.
nlmconv	Convertit code objet en un NLM.
nm	Liste les symboles des fichiers objet.
ranlib	Génère un index vers le contenu d'une archive.
readelf	Affiche information de n'importe quel fichier objet au format ELF.
size	Liste les tailles de section d'un fichier objet ou d'un fichier archive.
strings	Liste les chaînes imprimables de fichiers.
strip	Ne tient pas compte des symboles.

On peut trouver GNU GCC pour des dizaines de plate-formes-hôte différentes pour Linux, Microsoft Windows, UNIX, Solaris et bien d'autres. GNU GCC a été ciblé pour générer des exécutables pour de nombreuses architectures de microprocesseur différentes, allant du x86 d'Intel/AMD x86, au 68x00 de Motorola en passant par les PowerPC, ARM7 et ARM9. GNU GCC

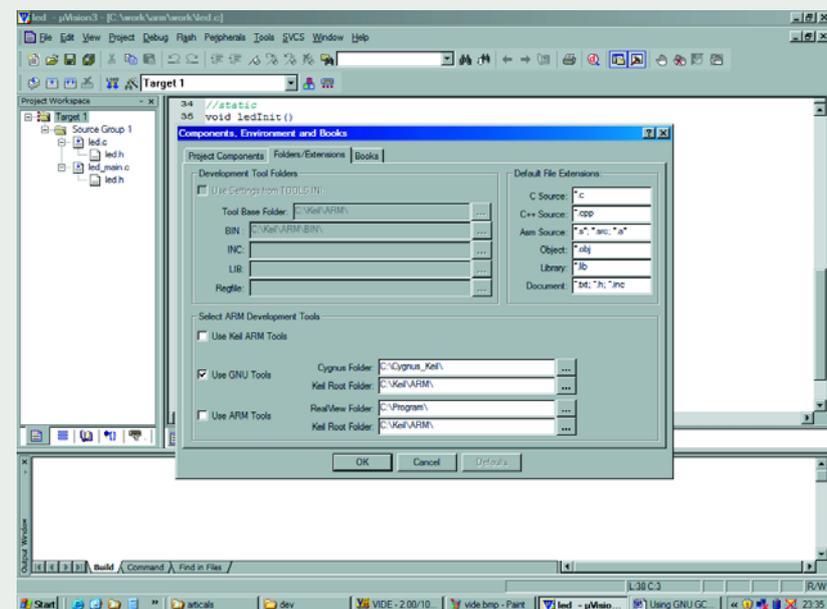
recouvre également plusieurs architectures de microcontrôleurs, dont les MSP430 de TI, les AVR d'Atmel et le 68HC11 de Motorola.

Autre bonne raison, les bugs, logiciels en particulier. La chaîne d'outils GNU est remise à jour régulièrement, ce qui signifie que tout problème signalé est résolu relativement vite.

Offres commerciales

Si vous ne voyez pas d'inconvénient à donner à Keil quelques informations personnelles, vous pouvez télécharger gratuitement leur IDE uVision et une version « préfabriquée » des outils GNU. Ces outils ne connaissent pas la moindre restriction pour la génération de code, à la seule limitation près (à ce jour) que le débogueur est limité à une taille de programme de 16 K. Les outils Keil reposent sur leur IDE uVision 3, la copie d'écran vous montrant cet IDE configuré pour GNU GCC.

Redhat et Microcross proposent une chaîne d'outils GNU « préfabriqués » pour la cible et la plate-forme-hôte de votre choix. Rowley Associates propose également un environnement de développement commercial basé sur la chaîne d'outils GNU mais reposant sur leurs propres bibliothèques et IDE.



Configuration de l'IDE uVision de Keil pour GNU GCC.

Il se pourrait que votre question suivante soit « quels est l'envers de l'utilisation de GNU GCC ? ». Tout d'abord l'incroyable palette d'hôtes et de cibles existante. Il vous faut commencer par voir si GCC existe pour votre plate-forme-hôte, voir ensuite si votre processeur-cible est supporté et pour finir décider si vous voulez créer (compiler en fait) le GCC vous-même ou si vous voulez en avoir la version binaire pré-compilée. Cependant, le plus souvent vous trouverez les architectures-cible les plus courantes tant pour des hôtes Win32 que Linux, et cela sous la double forme de code source et binaire. Ne vous attendez cependant pas à trouver un programme d'installation utilisant ou un IDE (*Integrated Development Environment*), nous nous trouvons ici dans le vrai monde de DIY (*Do It Yourself*). S'il vous faut un bel « installateur » et un IDE, jetez un coup d'oeil du côté des sociétés fournissant des versions préconfigurées et à mise en oeuvre instantanée des outils GNU. GNU-X tools de Microcross et GNUPro de Redhat sont 2 exemples d'outils GNU commerciaux.

Si vous voulez créer vos propres fichiers binaires, téléchargez-les du site Web de FS/GNU et suivez les instructions données. Nous allons pour notre part utiliser l'un des nombreux binaires « préfabriqués » et l'utiliser pour compiler notre code C.

Nous allons utiliser, pour nos exemples, les **binaires pre-built GNU GCC version 3.4 Win32** disponibles sur www.gnuarm.com. Vous pouvez, si vous le désirez, opter pour un hôte Linux. Il existe d'autres binaires *pre-built* sur le Web, un petit tour avec Google à base des mots-clés GNU, ARM et *binaires* et vous serez surpris du nombre de hits. Téléchargeons les binaires de GNUARM et installons-les sur notre PC.

Compilateur C GNU

Allez dans le sous-dossier GNUARM que vous avez installé, examinez-en les fichiers de programme et vous découvrirez un programme baptisé *arm-elf-gcc.exe*. Dans le même ordre d'idées, si vous utilisiez un microprocesseur à architecture SH3, vous trouveriez un *sh3-elfs-gcc.exe* dans le sous-répertoire.

arm-elf-gcc est un compilateur GCC GNU ayant pour cible l'architecture ARM et utilisant le format de fichier objet ELF (*Extended Linker Format*). Si vous utilisiez le format COFF (*Common*

Object File Format) vous y trouveriez un *arm-coff-gcc.exe*. ELF et COFF sont 2 formats de structure de fichier différents; le GCC peut enregistrer le produit objet du compilateur dans un format ou l'autre.

Notre première tâche est de nous assurer que le compilateur GCC a été installé correctement. Un test rapide consiste à faire apparaître le message de version du compilateur GCC. Ouvrez une fenêtre Command sous Windows ou une fenêtre MS-DOS et entrez l'instruction *arm-elf-gcc -v*. Si vous ne voyez pas le message de version GCC il est fort probable que vous ayez à ajouter le répertoire GCC à votre environnement chemin (*path*). Pour utiliser le compilateur, ouvrez une fenêtre DOS et entrez :

```
arm-elf-gcc -c ledtst.c
```

Vous trouverez le vrai programme dans l'encadré « *LEDTest - Un exemple* ». Nous mettons à votre disposition un fichier archive (*EPS0404444-11*) téléchargeable comportant le code assembleur, éditeur de liens (linker) et les fichiers compilateur résultants.

Le compilateur GCC possède une palette étoffée d'options, trop nombreuses pour les énumérer ici ! Voici quelques-unes de options à mettre dans la ligne de commande principale :

-c	pour compilation uniquement et sans liage du fichier objet;
-Wall	pour valider toutes les conditions d'avertissement et d'erreur que le compilateur pourrait produire;
-o nomdefichier	pour modifier le nom de fichier du fichier objet généré;
-S	pour produire un listing assembleur interlacé du code C origine;
-g	pour inclusion de l'information de débogage.

La commande

```
arm-elf-gcc -Wall ledtst.c -o ledtst
produira une compilation de ledtst.c avec validation de tous les avertissements compilateur et générer un fichier objet baptisé ledtst.o.
```

Lors de l'appel du compilateur GCC, ce dernier assure normalement le pré-traitement, la compilation, l'assemblage et le liage. Nous pouvons demander au compilateur de ne lier aucun des fichiers objet par adjonction de l'option -c. Si cela est le cas, nous pouvons utiliser le lieur GNU pour combiner nos fichiers objet en un seul et unique fichier de sortie.

Éditeur de liens GNU

Les fichiers objet produits en cours de compilation ne sont pas complets en eux-mêmes. Il nous reste à résoudre les variables internes et les références de fonction des fichiers objet et c'est là la tâche du lieur. Il combine ces fichiers objets et résout tout symbole et références de fonction non résolus.

La ligne de commande GNU linker *ld* (ou *arm-elf-ld*) prend comme argument les noms des fichiers objets à interlier ainsi que toute option et génère un unique fichier objet relocalisable. Si nous avons plusieurs fichiers à compiler et lier, nous pourrions entrer :

```
arm-elf-gcc -c module1.c
arm-elf-gcc -c module2.c
arm-elf-gcc -c module3.c
```

```
arm-elf-ld module1.o module2.o
module3.o -o prog1
```

Il n'y a plus, dans le fichier objet relocalisable, d'adresses de mémoire attribuées au code et aux sections de données. Il nous faut à cet effet un programme de localisation qui attribuera des adresses de mémoire physiques aux sections tant de code que de données.

Le lieur GNU intègre une fonction de localisation capable d'assigner les sections de code et de données du programme à des portions spécifiques de la mémoire ce qui se traduit par une image binaire à localisation absolue. Nous pouvons maintenant télécharger cette image binaire vers notre système de développement.

Le lieur GNU comporte un langage de script que l'on peut utiliser pour piloter le processus de liage et l'ordre exact des sections de code et de données à l'intérieur du programme relocalisable. Nous pouvons également assigner des adresses de mémoire aux sections de code et de données. Nous appelons un fichier script en mettant dans la ligne de commande l'option -T suivie du nom du fichier script :

```
arm-elf-ld -Tlpc2106-rom.ld
```

par exemple.

GNU Make

Le fichier make est utilisé pour automatiser le processus de compilation. Cet utilitaire suit en fait un set de règles définies par l'utilisateur qu'il exécute, les règles pas l'utilisateur, séquentiellement. Ainsi, au lieu d'entrer 4 lignes de commande de l'exemple de lieur plus haut, nous entrerions et enregistrerions ces instructions dans des fichiers de texte *makefile* ou tout autre nom de fichier utilisable séparés. Nous entrerions, pour compiler cet exemple :

```
make makefile
```

Fichier Start-up

Il faut, pour utiliser notre programme C compilé, commencer par initialiser les pointeurs de pile du processeur et les paramètres mémoire. Un fichier objet spécifique, dit fichier start-up, est chargé de cette tâche. Typiquement, le nom d'un tel fichier est *startup.s*, *boot.s* ou *crt0.s*. Le fichier start-up est lié au reste des fichiers objet compilés au cours de l'étape de liage du programme. Jetez un coup d'oeil aux fichiers d'exemple proposés par le groupe LPC2000 sur Yahoo, *crt0.s* ou *boot.s* qui sont des exemples types de fichiers start-up pour l'ARM.

Voici les fonction du code de start-up :

1. Inhiber toutes les interruptions.
2. Recopier les variables initialisés de la ROM vers la RAM.
3. Mettre à zéro les variables de données non initialisées.
4. Attribuer de l'espace à la zone de pile et l'initialiser.
5. Initialiser la pile.
6. Créer et initialiser la zone heap.
7. Valider les interruptions.
8. Sauter à to main ().

L'utilitaire Make détermine quels sont les fichiers de code-source ayant été modifiés depuis sa dernière exécution et ne compilera ainsi que les fichiers modifiés. Vous pouvez également inclure des options de compilation conditionnelles en utilisant des fonctions de prétraitement de Make.

Débogage

La chaîne d'outils GNU GCC comporte un outil de débogage au niveau de ligne de commande appelé GDB.

GNU GDB est un de ces outils dont le mode opératoire et le grand nombre d'options suffirait à remplir un ouvrage complet. Nous n'avons pas l'intention d'utiliser GDB avec la carte de développement ARMÉE, mais plutôt le Flash downloader proposé par Philips (figure 4). C'est pourquoi nous survolerons ici la grande majorité des fonctionnalités de GDB; ceux d'entre nos lecteurs qui voudraient en savoir plus au sujet de GNU GDB, pourront faire un tour sur Internet où ils trouveront une excellente introduction fournie par Bill Gatliff.

GDB est, en essence, un débogueur source. Par une liaison sérielle, il est connecté à une carte-cible avec laquelle il communique et interagit par le biais d'un programme moniteur sur la carte-cible. Pour lancer GDB entrez :

```
arm-elf-gdb ledtst
```

Une fois que GDB tourne, vous travaillerez sous la houlette de la console GDB par le biais de laquelle seront entrées les instructions de débogage. Vous pouvez ensuite charger le programme vers la carte-cible en utilisant l'instruction

```
(gdb) load
```

Une fois votre code chargé, vous pouvez le parcourir pas à pas, placer des points d'arrêt, voir quelles sont les fonctions appelées pour arriver à la ligne de source examinée, voire utiliser l'une des autres options de débogage offertes par GDB.

Autres outils

La chaîne d'outils GNU inclut quelques autres outils, les *binutils*, que nous pourrions utiliser en cours de développement. Objcopy et objdump sont 2 des outils les plus utiles.

L'utilitaire objcopy (ou *arm-elf-objcopy*) sert à convertir des fichier objet d'un format vers l'autre. Nous pouvons, par

LEDTest – un exemple

Dans le monde du développement enfoui, le premier programme normalement écrit avec un nouveau compilateur ou une nouvelle plate-forme-cible est un clignoteur à LED. Ceci est aussi évident que le fameux « hello world » en cours de développement sur PC. Nous n'allons pas déroger à la tradition. Notre programme d'exemple prend la forme d'un programme C qui fera clignoter une LED sur la carte de développement « ARMÉE ». Cf. listage !

Mettez le fichier C Philips ARM LPC2100 d'entête (header), le fichier start-up ARM (boot.o) et le script de liage GCC (lpc2106-rom.ld) dans le même répertoire que notre programme de test de LED. Depuis l'invite de ligne de commande entrez les instructions GCC suivantes :

```
arm-elf-gcc -c ledtst.c
```

```
arm-elf-ld -Tlpc2106-rom.ld -nostartfiles, -nostdlib -s -o led boot.o ledtst.o
```

```
arm-elf-objcopy --output-target ihex test test.hex
```

Nous devrions maintenant disposer d'un fichier .hex compilé et lié pouvant être envoyé vers la carte « ARMÉE ». Pour vous éviter d'avoir à retaper ces instructions GCC, nous pourrions les mettre dans un éditeur de texte et les enregistrer (save) comme fichier batch ledtst.bat ou encore créer un Makefile.

Pour finir, nous utiliserons, à la place de l'outil de débogage inclus dans GNU GCC, un petit programme de transfert Flash d4 Philips Semiconductors, cf. figure 4. Sélectionnez le nom de fichier de notre fichier .hex LED et transférez-le dans la carte-cible.

Une fois cette opération effectuée, appuyez sur le bouton Reset de la carte « ARMÉE » et vous devriez voir la LED D1 se mettre à clignoter.

Félicitations ! Vous venez de compiler, lier et transférer avec succès notre programme de test en utilisant le compilateur et les outils GNU GCC.

```
#include "LPC2000.h"

void Delay (unsigned long a)
{
    while (--a!=0);
}

int main (void) {

    /* System Init */
    /* Init PLL */
    SCB_PLLCFG = 0x23;
    SCB_PLLFEED = 0xAA; SCB_PLLFEED = 0x55;

    SCB_PLLCON = 0x01;          /* Enable PLL */
    SCB_PLLFEED = 0xAA; SCB_PLLFEED = 0x55;

    while ( !(SCB_PLLSTAT & PLOCK) ); /* Wait for PLL to lock */

    SCB_PLLCON = 0x03;          /* Connect PLL as clock source */
    SCB_PLLFEED = 0xAA; SCB_PLLFEED = 0x55;

    /* Init MAM & Flash memory fetch */
    MAM_MAMCR=0x2;             /* MAM = flash */
    MAM_MAMTIM=0x4;
    SCB_VPBDIV=0x1;           /* PCLK = CCLK */

    /* Init GPIO */
    GPIO_IODIR = 0x00000001;    /* P0.0 as output */
    GPIO_IOSET |= 0x00000001;   /* LED off */

    /* main loop */
    while (1) {

        GPIO_IOSET |= 0x00000001; /* LED off */
        Delay (1000000);

        GPIO_IOCLR |= 0x00000001; /* LED on */
        Delay (2000000);

    }
} /* End of main () */
```

exemple, traduire un fichier d'une image binaire vers un format s-record.
arm-elf-objcopy -O srec ledtst.o ledtst.s19

l'utilitaire objdump (ou *arm-elf-objdump*) peut être utilisée pour désassembler des fichiers objet. Une fois le fichier désassemblé, vous pourrez découvrir où le lieu à placé les différents textes, données et sections Bss de votre fichier objet.

```
arm-elf-objdump.    —disassemble ledtst.o
```

Vous pouvez de plus ajouter du code-source au fichier de sortie désassemblé en ajoutant l'option `-S` dans la ligne de commande. Si vous voulez voir le tableau symbolique de votre fichier objet, il suffit d'ajouter l'option `--syms` à la ligne de commande.

Le **tableau 1** énumère quelques autres *binutils* de GNU.

IDE

Une fois installé, GNU ne nous fournit

pas le moindre environnement IDE. Pour palier à cette absence, nous pouvons installer VIDE (**figure 5**), l'un des IDE les plus populaires pouvant être utilisés avec la chaîne d'outils GNU.

VIDE met à notre disposition un éditeur de texte et une interface graphique (GUI) sous laquelle tournent les outils GNU. Il existe d'autres IDE, gratuits et commerciaux, utilisables avec nos outils GNU. Un petit tour sur Internet pour voir ce qui existe, ou encore un passage dans le Forum de notre site pour entrer en contact avec d'autres « entortilleurs » d'ARM.

(040444-2)

Références :

1. Système de développement PICÉE Elektor n° 296, février 2003
2. AVRÉE - Système de développement pour AT90S2313, Elektor n° 297, mars 2003.
3. Carte 89S8252 Flash, Elektor n° 282, décembre 2001.

Autres lectures utiles

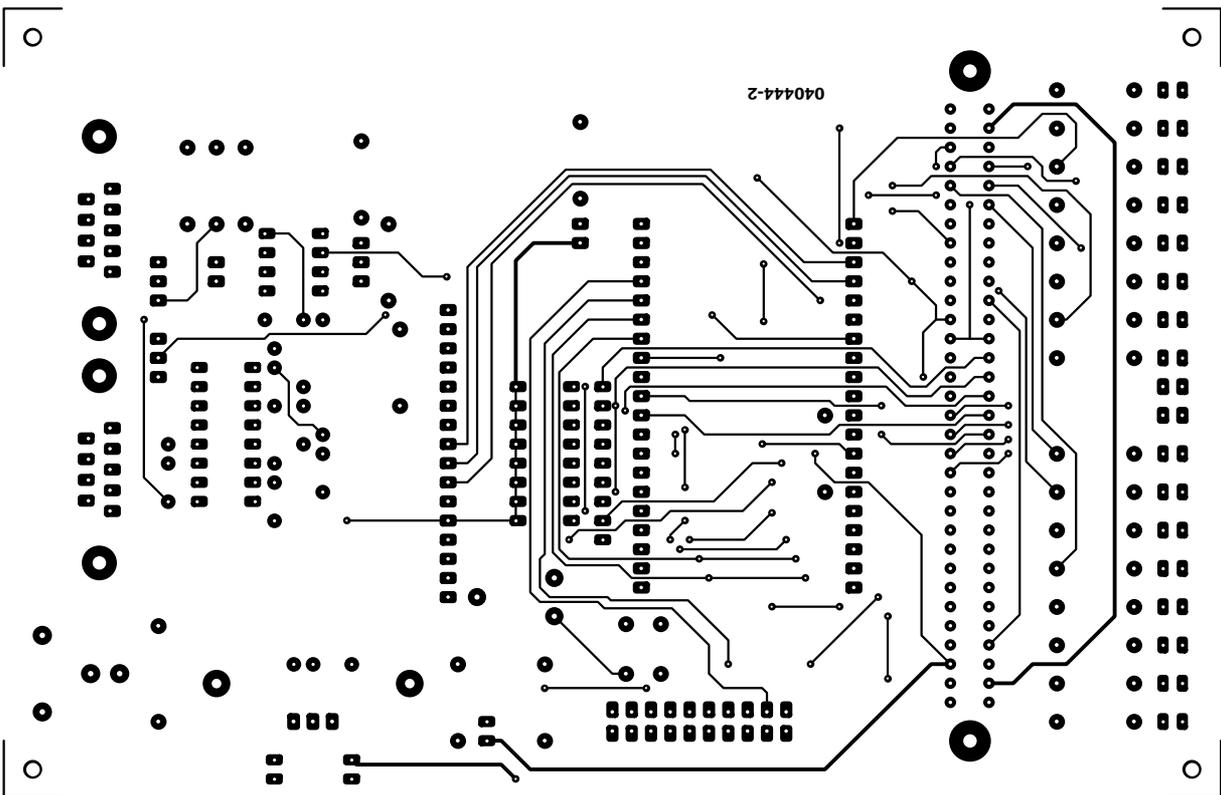
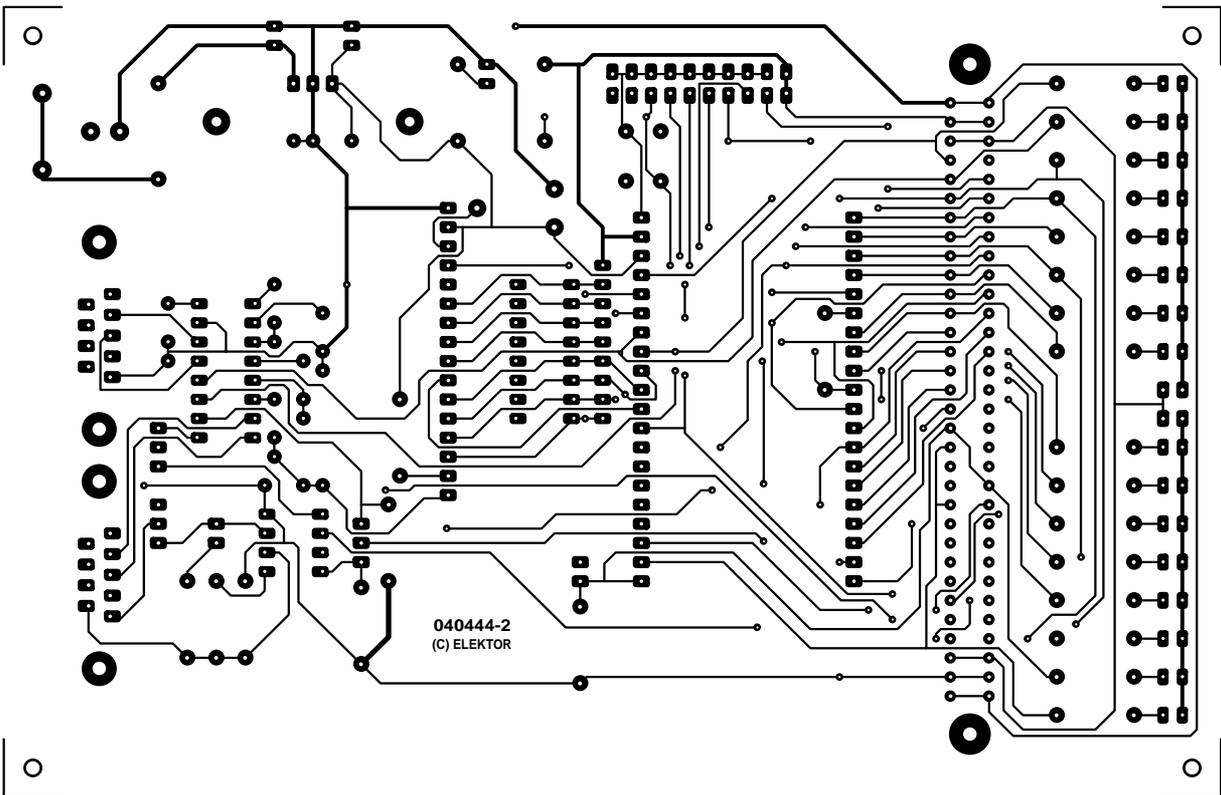
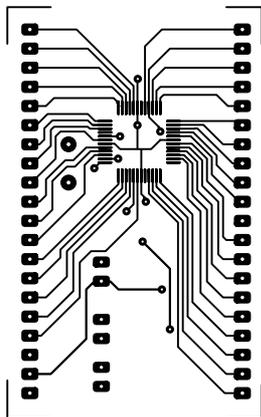
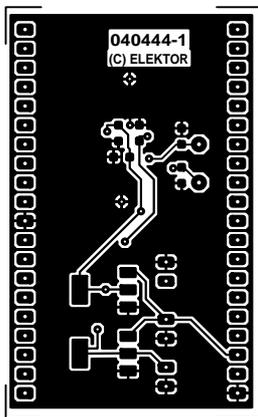
1. Programming Embedded Systems in C and C++, Michael Ball; O'Reilly (www.oreilly.fr)

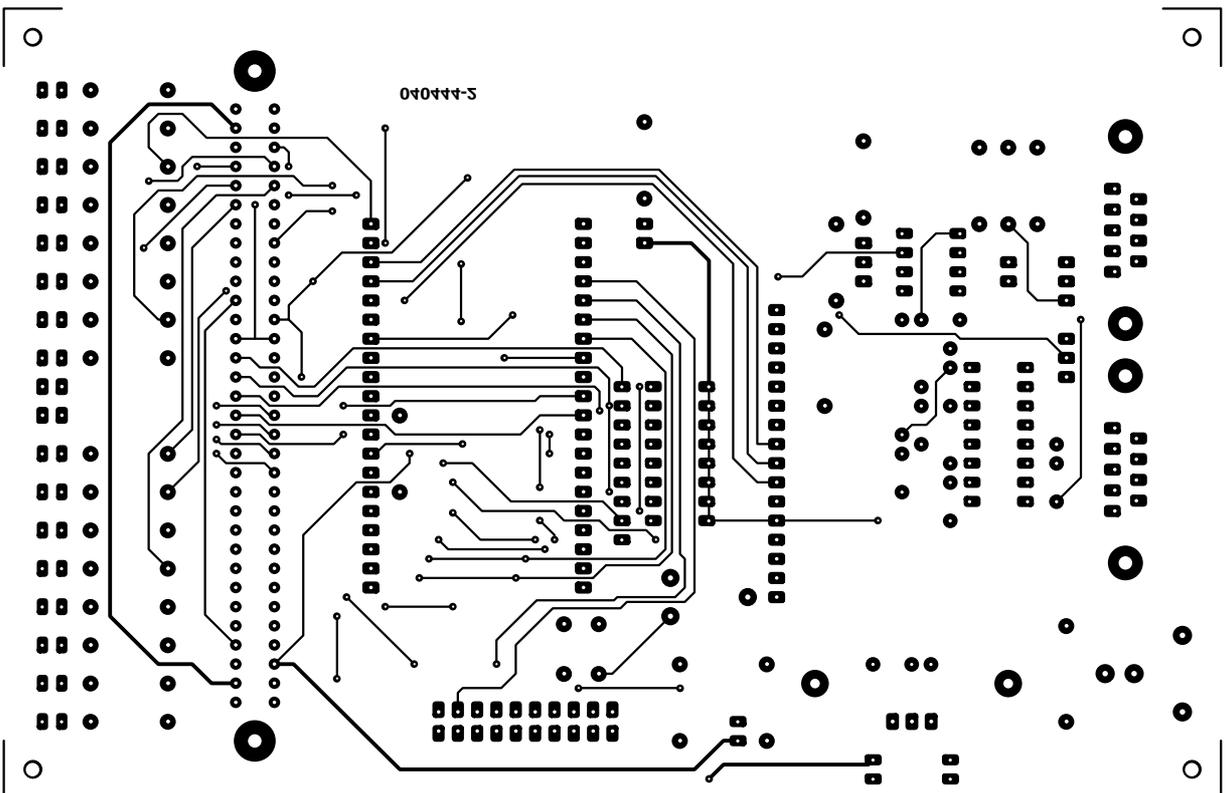
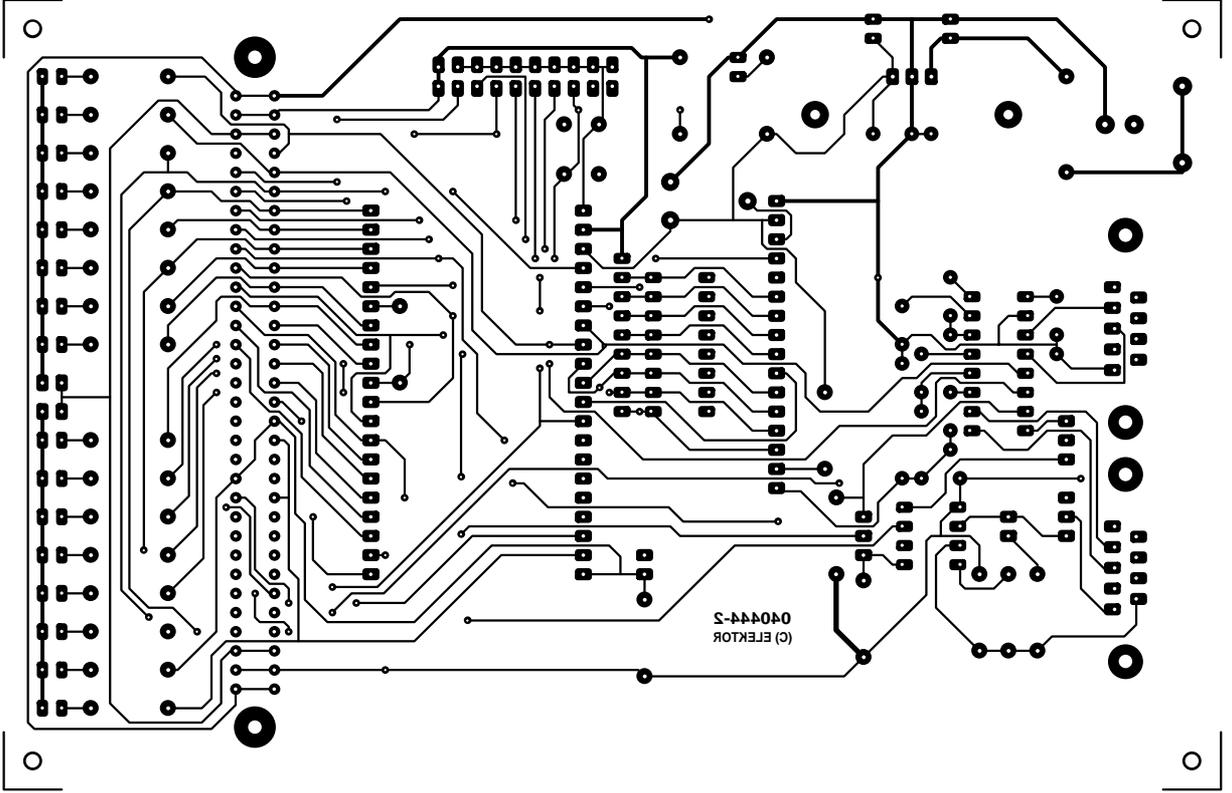
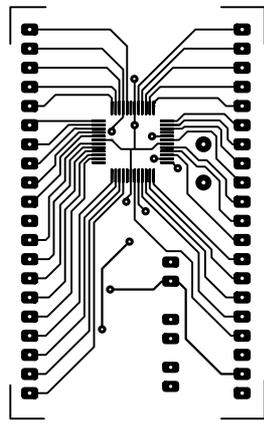
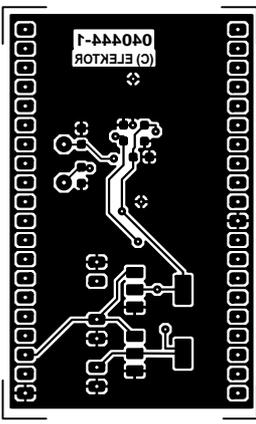
2. An Introduction to GCC, Brian J. Gough, préface de Richard M. Stallman; copie format .pdf payante sur www.network-theory.co.uk/gcc/intro/

Sites Internet :

- www.geocities.com/tonydixon2k1/index.html (site de support d'ARMÉE)
- www.gnu.org (Site Internet Principal de GNU)
- www.fsf.org (Site Internet Free Software Foundation)
- www.gnuarm.com (GNU Binaries pour processeurs ARM)
- <http://groups.yahoo.com/group/lpc2000/> (Groupe de discussion Philips LPC2000 ARM)
- <http://groups.yahoo.com/group/gnuarm/> (Groupe de discussion GNUARM)
- www.objectcentral.com/videl.htm (IDE VIDE pour Windows)
- www.billgatliff.com (excellents didacticiels GNU par Bill Gatliff)
- www.dreamislife.com/arm/ (Didacticiels LPC2106 par Senz)
- www.keil.com (GNU GCC et Compileurs C commerciaux)
- www.rowley.co.uk (GNU GCC et Compileurs C commerciaux)
- www.redhat.co.uk (Chaînes d'outils GNU GCC commerciales)
- www.microcross.co.uk (Chaînes d'outils GNU GCC commerciales)

Publicité

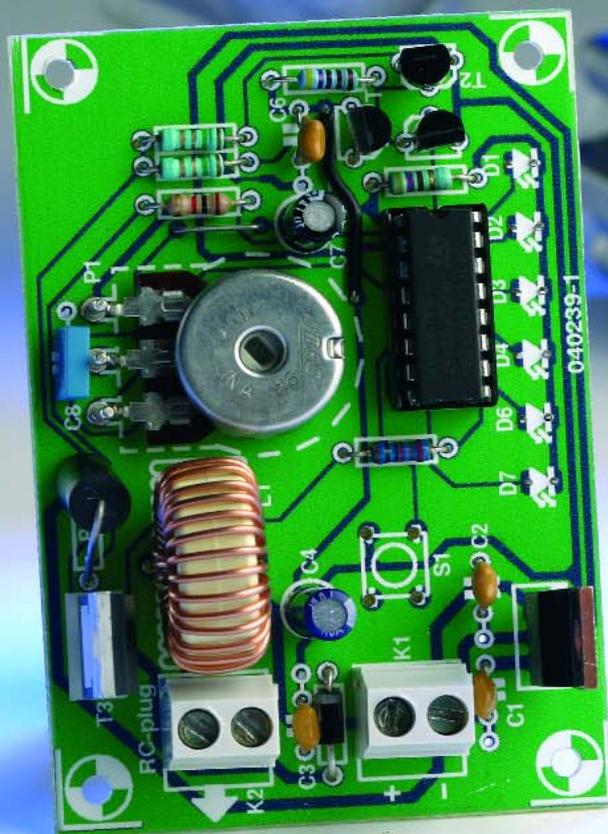




Démarrage garanti

RC-PLUG HEATER

Florent Coste



RC-Plug Heater -cela fait quand même plus chic que « réchauffeur » de bougie RC- a été conçu pour permettre le chauffage des bougies des modèles réduits radiocommandés (R/C).

Le lectorat auquel est destiné cet article sont les réalisateurs de tous types de maquettes embarquant un (des) moteur(s) thermique(s), telles qu'avions, voitures, bateaux et autres mobiles... et qui désirent pouvoir régler le courant de chauffage de la bougie et d'avoir une indication de son état physique.

Moteurs thermiques

Le présent montage est conçu spécifiquement pour les modélistes qui utilisent des modèles pourvus de moteurs thermiques, fonctionnant communément à partir d'un mélange d'huile, de méthanol et de nitro-méthane. La combustion de ce mélange permet de maintenir le filament de la bougie à une température suffisante pour assurer un fonctionnement correct du moteur sans besoin d'alimentation externe. Bien entendu, lors du démarrage, un apport énergétique initial est nécessaire, afin de provoquer une combustion interne qui viendra, par la suite, auto-entretenir l'incandescence de la bougie.

Généralement, une petite batterie externe (cellule 1,2 V de faible ampérage) est utilisée pour assurer le chauffage de la bougie, ne permettant aucunement un réglage du courant de chauffe. Qui plus est, combien de fois cette batterie n'a pas été rechargée la veille (oubli !) ou est tombée tout simplement à plat suite à une utilisation prolongée sur un moteur récalcitrant. Quant aux caisses de terrains pourvus d'un « *power panel* » pouvant assurer cette fonction, les méthodes employées par les fabricants sont quelque peu obsolètes (mais nous imaginons peu coûteuses...), puisqu'elles reposent le plus souvent sur une simple résistance intercalée entre l'alimentation (batterie de voiture) et la bougie. La chaleur dégagée par un tel dispositif nous rappelle souvent à l'ordre ! Une solution « active » devait s'imposer....

RC-Plug Heater

Le coeur de ce montage repose sur un microcontrôleur de l'écurie ST, le ST7Flite05. On en trouvera la fiche de caractéristiques à l'adresse : <http://www.st.com/stonline/books/ascii/docs/8348.htm>.

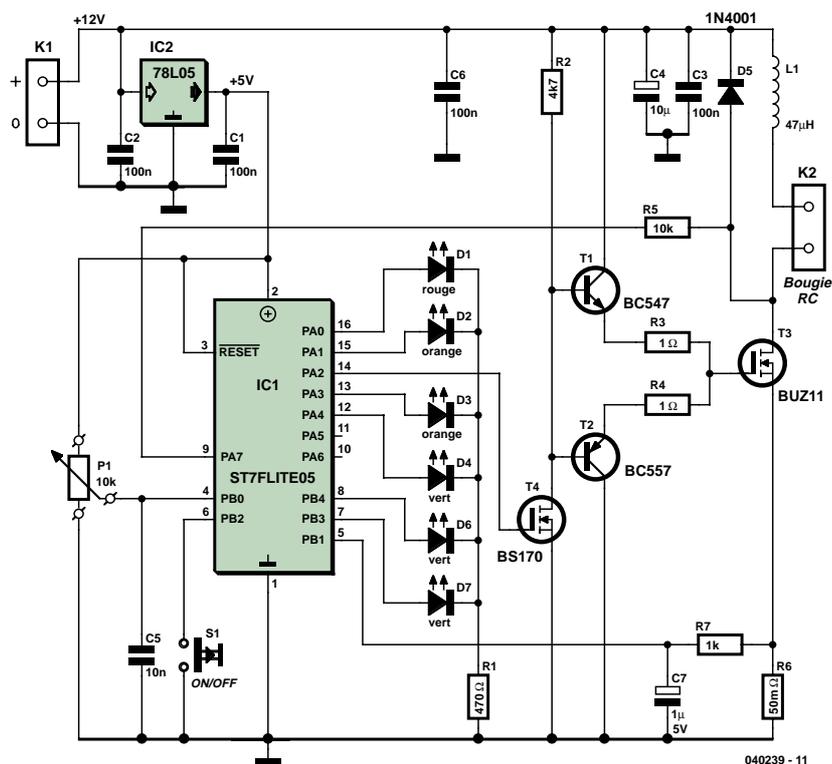


Figure 1. Le schéma de notre RC-Plug Heater brille par sa simplicité, un microcontrôleur moderne épaulé par quelques rares composants externes.

Ce petit boîtier n'a en fait de « LITE » que son nom : en effet il intègre pas moins de 1,5 Koctets de mémoire Flash, 128 octets de RAM, ainsi que de nombreux périphériques qui le rendent extrêmement performant et peu gourmand en composants externes. Pour ce montage, nous mettrons volontiers à contribution son oscillateur RC interne (calibré à 1%), la PLL intégrée assurant un cadencage à 8 MHz du coeur, ses multiples sources d'interruptions (externe : bouton-poussoir, interne : compteur RTC), le convertisseur Analogique/Numérique (CAN), ainsi que le mode MLI (Modulation en Largeur d'Impulsion = PWM pour *Pulse Width Modulation* en anglais) du compteur 12 bits. Le fonctionnement repose sur un découpage de la tension d'alimentation pour générer des impulsions de courant dans le filament de la bougie afin de la porter à bonne température.

Le schéma

Un coup d'oeil rapide au schéma représenté en **figure 1** suffit pour se rendre

compte du faible nombre de composants nécessaires : ce microcontrôleur est parfaitement taillé pour cette application, tout est déjà intégré ! Nul besoin de quartz externe, de résistance de forçage pour le contacteur, ou d'amplificateur pour la détection en courant.

Un petit régulateur 5 V alimente le ST7Flite05, ce microcontrôleur remplissant plusieurs fonctions :

- 1) La génération du signal MLI à une fréquence de 10 kHz sur sa broche PA2.
- 2) La lecture du courant désiré via le potentiomètre P1 et du convertisseur analogique/numérique (CAN).
- 3) La lecture du courant de chauffage toujours à l'aide du CAN et de l'ampli opérationnel interne.
- 4) L'affichage du courant et de l'état de la bougie sur la rangée de LED multicolores.
- 5) La validation/inhibition du chauffage à l'aide du bouton/poussoir S1.

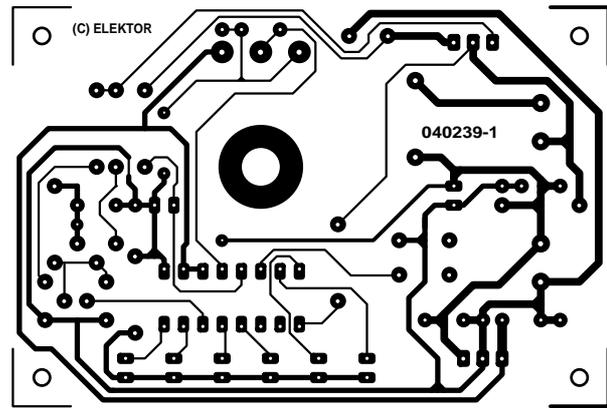
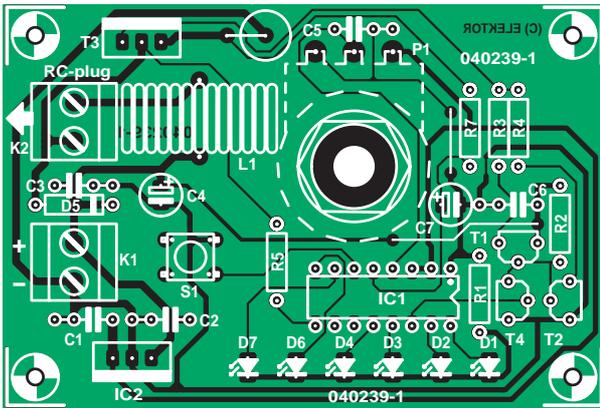


Figure 2. Dessin des pistes et implantation des composants de la platine dessinée à l'intention de cette réalisation.

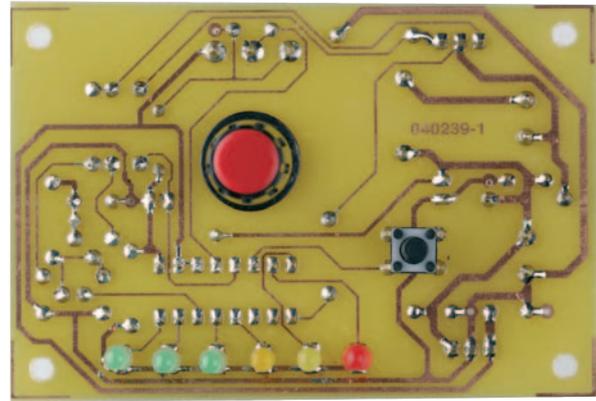
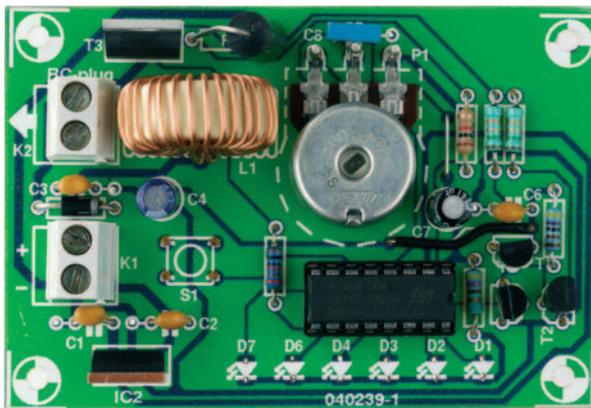


Figure 3. Exemple terminé du RC-Plug Heater.

L'étage de sortie

Un filament de bougie consomme un courant non négligeable, pouvant aller jusqu'à presque une dizaine d'ampères sous 1 volt environ. La partie puis-

sance sera dès lors assurée par un classique MOSFET BUZ11 (et généralement traînant dans les tiroirs de tout électronicien). On aurait pu choisir un FETMOS de type *Logic Level* et se passer alors d'un pré-étage (R2, T4)

pour la mise à niveau du signal de grille (5 V vers 12 V), mais ce type de composant est moins courant et souvent plus coûteux que ses homologues 12 V. R2 est la résistance de forçage au 12 V permettant d'attaquer l'entrée

Liste des composants

Résistances :

R1 = 470 Ω
 R2 = 4kΩ7
 R3,R4 = 1Ω5
 R5 = 10 kΩ
 R6 = 0Ω05 (50 mΩ)
 R7 = 1 kΩ
 P1 = 10 k10 μF/25 V log

Condensateurs :

C1 à C3,c6 = 100 nF
 C4 = 10 μF/25 V
 C5 = absent
 C7 = 1 μF/25 V
 C8 = 10 nF

Semi-conducteurs :

T1 = BC547
 T2 = BC557
 T3 = BUZ11
 T4 = BS170
 IC1 = ST7FLITE05 (programmé **EPS040239-41**)
 IC2 = 78L05

D1 = LED rouge 3 mm haute luminosité
 D2,D3 = LED orange 3 mm haute luminosité
 D4 à D7 = LED verte 3 mm haute luminosité
 D5 = 1N5820

Selfs :

L1 = 47 μH

Divers :

S1 = bouton-poussoir unipolaire à contact travail
 K1,K2 = bornier encartable à 2 contacts au pas de 5 mm

d'un étage push-pull constitué par T1 et T2. Ces 2 transistors permettent à leur tour d'assurer une charge et décharge très rapide de la grille du BUZ11, et ce malgré la valeur élevée de la résistance R2.

Pour ceux d'entre vous qui voudraient utiliser un autre type de MOSFET (canal N), ils peuvent éventuellement décroître la valeur de R2 afin de diminuer le temps de commutation si nécessaire. Attention toutefois à ne pas descendre trop bas puisque cette résistance dissipe constamment de l'énergie.

La détection du niveau de courant est assurée par 1 résistance de 50 mΩ, R6, et une cellule R/C (R7/C7) venant délivrer une tension de quelques dizaines de millivolts sur la broche PB1. La diode D5 protège des crêtes de tension parasites lors des commutations. La résistance R5 permet de vérifier la présence d'une bougie et de s'assurer qu'elle n'est pas grillée (présence d'un état bas sur la broche PA7 de IC1 dans ce cas).

Il nous reste à évoquer la self L1.

De par la nature de la charge (assimilable à une résistance de très faible valeur), le filament de tungstène restitue efficacement l'énergie emmagasinée lors de l'impulsion de courant, de sorte que l'on observe une intensité moyennée du courant sans qu'il ne soit besoin, théoriquement, d'ajouter de self dans l'étage de sortie. Nous avons cependant opté pour la sécurité tant au niveau de notre bougie que de celle des modèles réduits évoluant à proximité (le câble reliant le montage à la bougie aurait alors fait office d'antenne et de source de parasites potentiellement dangereux) et avons ajouté la self L1. Nous n'allons pas nous formaliser des quelques cm² de platine supplémentaires requis par cette self capable de supporter plusieurs dizaines d'ampères.

Construction et fonctionnement

Le circuit simple face dont on retrouve le dessin des pistes et la sérigraphie de l'implantation des composants en **figure 2** n'appelle pas de commentaires particuliers. On commencera par la mise en place du pont de câblage, puis

des résistances, des condensateurs, de la self et des circuits intégrés (attention aux polarités, IC1 étant de préférence monté sur un support de bonne qualité), pour finir par les connecteurs, K1 destiné à se voir appliquer la tension d'alimentation (+12 V) fournie par un adaptateur secteur ou une alimentation à batterie de voiture classique dans le monde du modélisme, K2 étant lui connecté à la bougie, pour terminer par le potentiomètre. Un petit radiateur sera nécessaire pour refroidir le FET-MOS et la diode vu les courants qu'ils ont à supporter.

Avant de monter le microcontrôleur sur son support, mieux vaut s'assurer de la présence du +5 V sur sa broche 2 après avoir branché le circuit (sans charge) sur une source 12 V. Ensuite, après avoir coupée l'alimentation, monter le ST7Flite05 et tourner le potentiomètre à fond dans le sens antihoraire pour s'assurer que le courant de chauffage est réglé à son minimum. Mettre sous tension. Comme aucune bougie n'est branchée, les LED de différentes couleurs (1 rouge, 2 oranges et 3 vertes) devraient s'allumer successivement (chenillard) pour signaler que rien n'est détecté (ou bien que la bougie est grillée... si elle est connectée au montage !).

Ensuite, ajuster le courant à l'aide du potentiomètre. Attention à ne pas trop solliciter la bougie sous peine de griller le filament (un contrôle visuel est parfois souhaitable, surtout pour des bougies dites « chaudes », ayant des filaments plus fins que les bougies « froides »). Un filament légèrement porté au rouge est bien souvent suffisant.

Il ne reste plus ensuite qu'à appuyer sur le bouton-poussoir S1 pour provoquer le chauffage du filament de la bougie.

Le montage consomme environ 11 mA à vide.

En guise de conclusion

Le programme implanté dans le ST7Flite05 a été écrit en C et compilé à l'aide du compilateur METROWERKS (www.metrowerks.com). Il existe par ailleurs un autre compilateur C disponible chez COSMIC (www.cosmic-software.com). Des versions gratuites d'évaluation et « light » limitées à

quelques kilo-octets permettent de travailler sans aucune restriction sur ce type de micro. Le programme (disponible au téléchargement gratuit sur notre site (ou sous la forme d'une disquette payante disponible auprès des adresses habituelles pour ceux de nos lecteurs qui n'auraient pas accès à Internet) sous la dénomination **EPS040236-11**) peut ainsi être modifié à votre guise ; une connaissance minimale en langage C et un programmeur seront nécessaires. Des outils de programmations bon-marché et utilisables pour toute la famille des microcontrôleurs Flash ST7 sont aussi disponibles chez ST Microelectronics (« *stick programmer* ») et Softec-microsystems (Indart STX/D, qui permet aussi de déboguer l'application).

(040239-1)

Liens Internet

Metrowerks : www.metrowerks.com

Cosmic : www.cosmic-software.com

ST : www.st.com

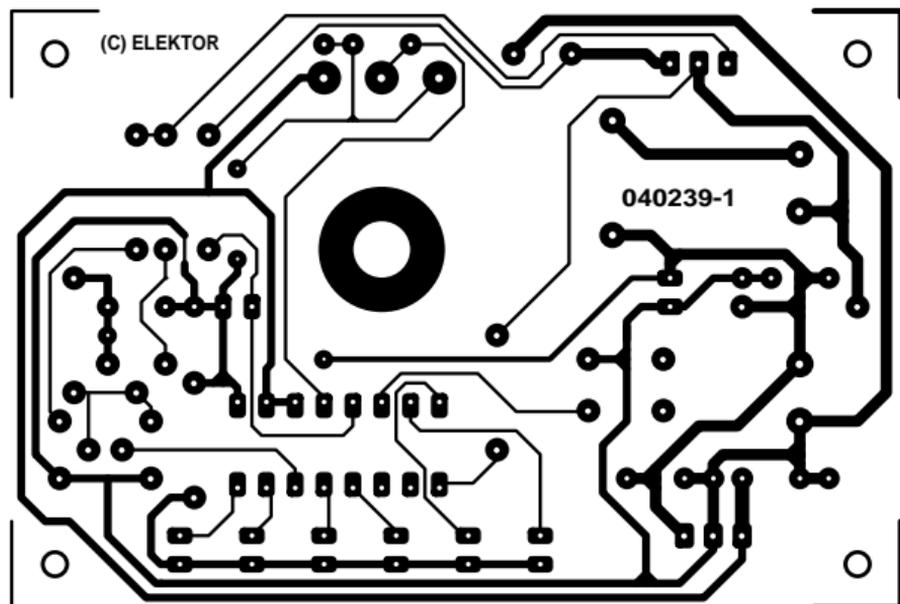
L'auteur

Florent COSTE, Diplômé Ingénieur Micro-Electronicien en l'an 2000 (Institut Charles Fabry, Marseille, France).

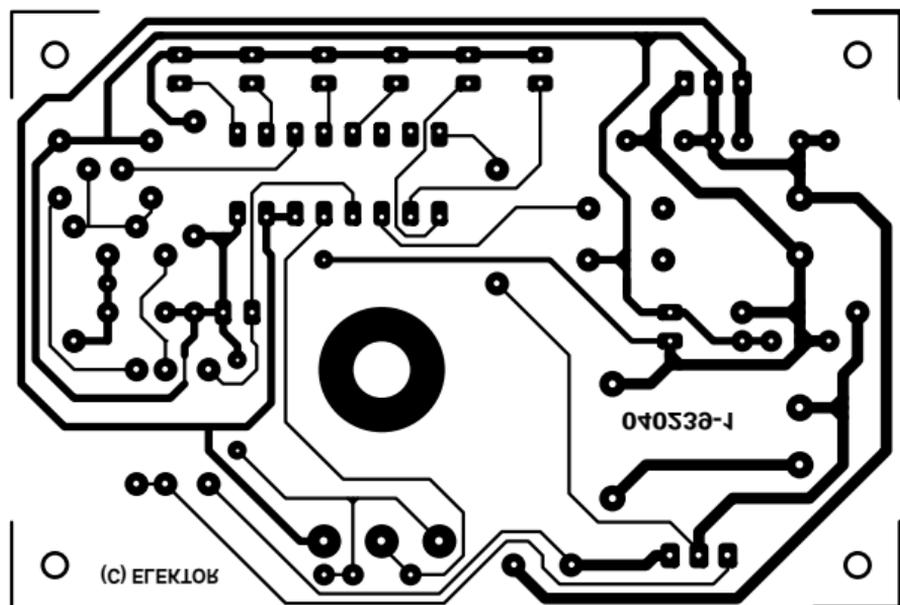
Employé par STmicroelectronics depuis, et basé à Hong-Kong en tant qu'ingénieur application & support.

Spécialisé dans les logiciels pour microcontrôleurs, il a, pendant 2 ans, travaillé en étroite collaboration avec la clientèle asiatique pour la mise en oeuvre de plate-formes car-multimedia. Il s'est par la suite spécialisé dans les applications « contrôle moteur », l'amenant à développer des projets autour de micros (« ST7MC » pour parler du tout dernier micro ST en « vogue ») dédiés au pilotage de moteurs synchrones (*brushless*, air conditionné par exemple) et asynchrone (induction).

Passionné d'aéromodélisme et d'électronique, il met très volontiers à profit ces 2 disciplines pour remplir ses temps libres.



non reflected



reflected

DELPHI

POUR ÉLECTRONICIENS

Partie 4 - Utilisation créative de la carte-son



Herman Bulle

La majorité des ordinateurs modernes comporte une carte-son, soit embarquée sur la carte-mère soit enfoncée dans un connecteur. Cette carte permet la conversion de données numériques en données analogiques. Peut-on utiliser une telle carte en générateur de fonctions se demanderont sans doute les électroniciens. Cela est bien évidemment possible à condition de tenir compte des limitations de la carte-son. Il existe à cet effet des programmes sur le marché mais quoi de plus instructif et marrant que d'écrire son propre programme Delphi pour la génération des formes d'ondes souhaitées.

Nous allons dans cet article construire, un générateur de fonctions logiciel générant un sinus, une dent de scie ou un signal rectangulaire sous forme logicielle envoyé vers la carte-son. Ce générateur a en outre la possibilité de lire un fichier du disque dur pour le reproduire sous forme d'onde.

La bande passante d'une carte-son est de 20 kHz environ. Si l'on veut s'en servir pour générer des formes d'ondes, on pourra réaliser un générateur sinus qui va jusqu'à 20 kHz. Il nous faut, pour les autres formes d'onde, au minimum une harmonique du 3^e ordre. La fré-

quence maximale se situe partant à 7 kHz environ. N'allez pas croire qu'il est possible de réaliser un générateur de fonctions allant jusqu'à 1 MHz.

On pourra, comme avec quasiment tous les projets plus ou moins complexes, le subdiviser en blocs d'un ordiogramme (*flowchart*) logique.

Le premier bloc sert à créer une période du signal. Il faut ensuite traiter ce signal rectangulaire pour pouvoir l'appliquer à la carte-son et faire en sorte que la carte-son reproduise ce signal en boucle jusqu'à ce qu'il y ait action sur le bouton « Stop ».

Génération de la fonction

Nous avons, pour les fonctions simples à réaliser, déclaré une matrice (*array*) du type **TDataArray** valide sur l'ensemble du programme (*global*). Le nombre d'échantillons que nous mettons dans cette matrice est rendu par la constante globale **NumOfSamples**.

Pour un sinus, on remplit cette matrice avec une période. On déclare, dans la fonction reliée au bouton « Sinus », une matrice **SineAr**, matrice du type TDataArray. Celle-ci est remplie à l'aide d'une période du sinus. Vu que les données à envoyer à la carte-son doivent être des entiers de 2 octets, il nous faudra d'abord arrondir ces valeurs. Pour le calcul du sinus nous faisons appel à une fonction spéciale. Delphi connaît en effet une fonction **sincos(phi, sin, cos)** beaucoup plus rapide qu'un simple calcul de sinus ou de cosinus vu que Borland l'a écrite en assembleur pur. Nous allons bien évidemment l'utiliser :

```
FOR index := 0 TO NumOfSamples DO
  BEGIN
    Sincos(2 * pi * index / NumOfSamples, sin,
cos);
    SineAr[index] := round(amplitude * sin);
  END;
```

La prise en compte d'une forme d'onde externe requiert une approche quelque peu différente.

```
FOR Index := 0 TO NumOfSamples - 1 DO
  SawToothAr[Index] := round(2 * Index * Amplitude / NumOfSamples) - amplitude
```

```
FOR Index := 0 TO NumOfSamples DIV 2 - 1 DO
  BEGIN
    SquareAr[Index] := amplitude;
    SquareAr[NumOfSamples - 1 - index] := -
amplitude;
  END;
```

Il est relativement simple de constituer une forme d'onde sous Excel. Dans l'exemple Excel Signals.xls (cf. le logiciel associé à ce projet **(040240-4-11)** sur le site Elektor) on a créé une forme d'onde de 100 périodes à amplitude maximale, suivies de 100 périodes à une amplitude de 10%. Les amateurs d'audio apprécient ce type de signaux pour tester le comportement de leurs amplificateurs face aux transitoires. La forme d'onde est exportée sous forme de fichier CVS.

Ce fichier est lu dans une variable du type **TStringlist**. Il s'agit d'un bloc de mémoire dont la taille n'est pas fixée d'avance. On parle de variable dynamique. Nous retrouverons ce type de variable plus loin. Comme, lors d'une déclaration, on ne définit que le type, il nous faudra définir l'adresse de début vers laquelle nous pouvons pointer lors de l'utilisation de cette variable. Nous le faisons par un appel de **Create**. Nous avons dans ce cas déclaré une variable **DataIn** de type **TStringlist**:

```
Var DataIn: TStringlist;
```

Dans le corps de la procédure on ajoute la ligne suivante :

```
DataIn := TStringlist.Create;
```

Cette variable est ainsi prête à être utilisée. Comme le suggère son nom, elle est destinée au stockage de chaînes. Comme les données du fichier Excel sont stockées dans un fichier .CVS (texte lisible), il nous est possible de les

stocker dans la variable pour traitement ultérieur. Les têtes pensantes de Borland ont imaginé une méthode (lire procédure) permettant une lecture facile de ce type de fichier depuis le disque dur :

```
DataIn.LoadFromFile('signals.csv')
```

Pour accroître la flexibilité du programme on emprunte un dialogue fichier (File, Open) à Windows, ce qui permet une sélection de fichier facile. Il s'agit d'un composant se trouvant dans l'onglet Dialogues et qui s'appelle *OpenDialog*. La construction la plus utilisée est la suivante :

```
IF OpenDialog1.Execute THEN
  DataIn.LoadFromFile(OpenDialog1.FileName)
ELSE
  BEGIN
    //generate an error message
    MessageDlg('File is not correct', mtError,
[mbOK], 0);
    //exit the procedure
    exit;
  END;
```

Nous avons pris les signaux en compte ou les avons créés. Il nous reste à convertir ces données en un format que la carte-son puisse comprendre.

Préparation pour la carte-son

Il est temps de faire croire à la carte que nous allons lui faire reproduire une pièce sonore.

Un type de fichier que la carte-son puisse traiter est doté de l'extension .WAV. Un fichier .wav comporte un en-tête de 44 octets dans laquelle se trouve des informations concernant le fichier, les données proprement dites venant ensuite. Ces données prennent la forme des valeurs du signal à des instants successifs (signal échantillonné). Il est possible, pour peu que l'on prenne des échantillons à intervalles suffisamment rapprochés de produire un son acceptable voire très bon. Nous n'allons pas entrer dans la théorie sur laquelle repose la prise d'échantillons. La **figure 1** reproduit un exemple d'échantillonnage simple.

Le type .wav fait partie d'une catégorie plus générale dans laquelle sont également définis les fichiers vidéo (tels qu'avi). Nous nous ne nous intéressons ici qu'à la forme la plus simple. Il est possible ainsi de décrire l'en-tête plus simplement que ne le font les documents de spécifications et on a l'impression que l'en-tête comporte des données en double. Toutes les données de l'en-tête sont regroupées pour constituer un enregistrement (record). C'est un bloc de données dont les parties (*fields*) peuvent être constituées de différentes sortes de variables telles que chaînes, matrices, entiers, etc. En spécifiant un enregistrement de la sortie nous avons défini à combien d'octets après le début de l'enregistrement se trouve une variable donnée et sa taille. En cas d'erreur à ce niveau, le compilateur signale une erreur de type, l'exécution du programme se faisant toujours sans problème.

De manière à améliorer le temps d'accès, on arrondit souvent, dans le cas de données complexes (comme les enregistrements) les champs vers le haut, à 2, 4 ou 8 octets. De ce fait, par exemple, une variable telle que **ChunkID** (3 octets) pourra se trouver arrondie à 4 octets. Toutes les autres variables se trouvent décalées d'un octet, de sorte que la carte-son ne s'y retrouve plus. Nous pouvons éviter ce problème en déclarant l'enregistrement **PACKED**.

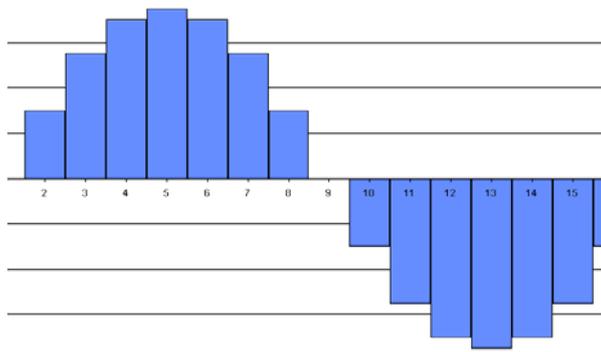


Figure 1. Distribution en 16 pas d'un signal échantillonné.

Nous avons gardé, pour les noms des champs, autant que possible les noms originaux de la spécification du fichier .wav.

Type

```
TFormat = PACKED RECORD
  ChunkID: ARRAY[0..3] OF Char;
  ChunkSize: longword;
  Format: ARRAY[0..3] OF Char
  SubChunk1ID: ARRAY[0..3] OF Char;
  SubChunk1Size: longword;
  AudioFormat: word;
  NumOfChannels: word;
  SampleRate: longword;
  ByteRate: longword;
  BytesperSample: word;
  BitsPerSample: word;
  SubChunk2ID: ARRAY[0..3] OF Char;
  SubChunk2Size: longword;
```

END;

Nombre des données de l'en-tête ne changent jamais. Nous faisons appel à la procédure **PrepareHeader** pour remplir de ces données l'en-tête wave. Les valeurs ci-après sont attribuées aux champs de l'en-tête :

```
WITH header DO
BEGIN
  ChunkID := 'RIFF';
  Format := 'WAVE';
  SubChunk1ID := 'fmt ';
  SubChunk1Size := $10;
  AudioFormat := $01;
  SubChunk2ID := 'data';
  NumOfChannels := 1;
  BytesperSample := 2;
  BitsPerSample := 16;
END;
```

Comme nous nous limitons pour le moment à un signal mono, **NumOfChannels** se voit attribuer la valeur 1 (stéréo = 2). Nous fixons également la taille d'échantillon à 16 bits (*smallinteger*).

SampleRate est la fréquence de lecture des échantillons. Comme on lit, pour les signaux « ordinaires », une période entière, le taux d'échantillonnage est égal à la fréquence de récurrence multipliée par le nombre d'échantillons dans une période.

ByteRate représente le nombre d'octets lus par seconde et vaut partant $\text{SampleRate} \times \text{NumOfChannels} \times \text{taille d'un échantillon (mono)}$. Cela devient, dans notre cas $2 \times \text{SampleRate}$.

Les champs restants concernent la taille du bloc de données :

Chunksize := taille des données (en octets) +36
SubChunk2Size := taille d'un bloc de données

Nous avons pratiquement terminé la construction du fichier .wav complet. Nous disposons de l'en-tête et des données. Il nous suffit de les accoler et de les envoyer vers la carte-son.

Delphi offre une méthode très flexible de réservation d'un bloc de mémoire et y stocker des données. Elle utilise la « famille » de **TStream**. Cette famille comporte plusieurs membres; nous utiliserons le flux mémoire (type **TMemoryStream**). TMemorystream a hérité toutes les propriétés de TStream et en possède quelques-unes qui lui sont propres. Les streams étant des variables dynamiques, on ne connaît en début de programme que la définition de type. Il restera à créer la variable proprement dite. Comme nous ne pouvons envoyer qu'un fichier .wav vers la carte-son, il ne nous faut qu'une seule variable de ce type que nous créons, en début de programme, dans la méthode OnCreate :

```
Wavestream:=TMemoryStream.Create
```

Tout comme dans le cas d TStringlist, la taille de la variable n'est pas fixée. C'est cette propriété qui donne tant de flexibilité à un stream.

Pour recopier l'en-tête vers le début de Wavestream, nous commençons par vider ce dernier :

```
WITH WaveStream DO
BEGIN
  Clear;
  WriteBuffer(Waveheader, sizeof(Waveheader));
  WriteBuffer(ar, sizeof(ar));
END;
```

Nous utilisons ensuite, comme on le voit ci-dessus, la méthode **WriteBuffer** :

```
WriteBuffer(Waveheader, sizeof(Waveheader));
```

Le premier paramètre est la variable à écrire vers le stream, le second donnant le nombre d'octets à écrire. Nous avons mis auparavant les données dans une matrice. Celles-ci sont écrites selon la même technique vers le stream. Le stream mémorise à l'aide d'un curseur la fin du stream et commence à écrire à partir de là. On place ainsi les données sans risque directement proprement en aval de l'en-tête.

```
WriteBuffer(ar, sizeof(ar));
```

Il nous est possible de lire le curseur évoqué plus haut et de le modifier à l'aide de la propriété **Position**. Les données sont maintenant prêtes à être envoyées vers la carte-son.

Comme la création d'un fichier .wav fait appel à 3 fonctions préprogrammées, nous avons combiné PrepareHeader et WriteBuffer dans la procédure :

```
Data2Wav(Sender: TObject; VAR ar: TDataArray)
```

Pour la visualisation, la fréquence d'échantillonnage n'est pas très pratique. Nous préférierions pouvoir paramétrer et lire la fréquence de récurrence du signal. Comme nous connaissons le nombre d'échantillons de données, la relation entre la fréquence de récurrence et le taux d'échantillonnage est simple :

```
WaveHeader.SampleRate = Freq * NumOfSamples;
```

Nous pouvons, à l'aide d'un ascenseur, ajuster la fré-

quence de récurrence pour calculer ensuite le SampleRate et le visualiser sur l'écran.
Après lancement d'une fonction d'onde on a lecture de FreqTrackBar et paramétrage de SampleRate.
Une fois ceci fait, nous pouvons, par le biais de **FreqTrackBar**, jouer sur la fréquence de récurrence. On a lecture de la position de Trackbar et calcul de SampleRate. On coupe ensuite brièvement le son (nous y reviendrons). Le curseur du stream est mis à 0, au début de l'en-tête, et à l'aide de WriteBuffer on recopie l'en-tête modifié par-dessus l'ancienne. Le son est ensuite rebranché. L'affichage de fréquence est bien entendu mis à jour après tout changement de fréquence.
Ces actions sont exécutées lors de chaque déplacement de l'ascenseur de FreqTrackBar. Il est logique par conséquent de faire démarrer ces action par l'événement **OnChange**. Sélectionnez l'ascenseur et double-cliquez sur l'événement OnChange sous l'onglet Événements de l'Inspecteur d'Objets. Nous pouvons entrer les instructions suivantes dans la procédure OnChange :

```
PROCEDURE TForm1.FreqTrackBarChange(Sender:
TObject);
BEGIN
  WITH FreqTrackBar DO
    BEGIN
      Waveheader.SampleRate :=
FreqTrackBar.Position * NumOfSamples;
      //zet geluid uit
      PlaySound(NIL, 0, 0);
      WaveStream.Position := 0;
      WaveStream.WriteBuffer(Waveheader,
sizeof(Waveheader));
      //met '%5d' worden de getallen
weergegeven met max 5 posities
      FreqLabel.Caption := Format('%5d', [Freq-
TrackBar.Position]) + ' Hz';
      //zet geluid weer aan
      PlaySound(MStream.Memory, 0, SND_MEMORY OR
SND_LOOP OR SND_ASYNC);
    END;
  END;
```

Il nous reste à convertir la forme d'onde importée en fichier .wav. Ce dernier avait, auparavant, été lu dans une StringList **DataIn**, un stream optimisé pour les chaînes. La StringList possède un compteur (property **Count**) qui indique le nombre de chaînes lues. On connaît partant le nombre d'échantillons. La forme d'onde ayant été créée à l'extérieur, nous allons supposer que le taux d'échantillonnage est de 50 kHz. Le taux d'échantillonnage doit être stocké dans 2 octets de sorte qu'il ne peut pas dépasser 64 kHz. 50 kHz est proche de la fréquence maximale et facilite les calculs. Lors de la construction dans Excel il a déjà été tenu compte de l'amplitude maximale. Encore quelque chose de réglé ! Nous pouvons, à l'aide de ces données, créer l'en-tête wave. Nous pouvons ensuite lire la StringList échantillon après échantillon, le convertir vers un *smallinterger* et l'ajouter au WaveStream :

```
FOR index := 0 TO SampleCount - 1 DO
  BEGIN
    sample := StrToInt(DataIn.Strings[index]);
    WaveStream.WriteBuffer(sample, sizeof(sample));
  END;
```

Une fois ceci fait, la carte-son peut se voir attaquer par le WaveStream, la reproduction pouvant alors commencer.

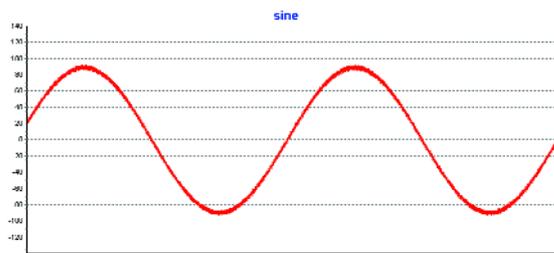


Figure 2. Forme d'onde du signal sinus à la sortie de la carte-son (2 kHz)



Figure 3. Voici le signal en dents de scie (2 kHz lui aussi).

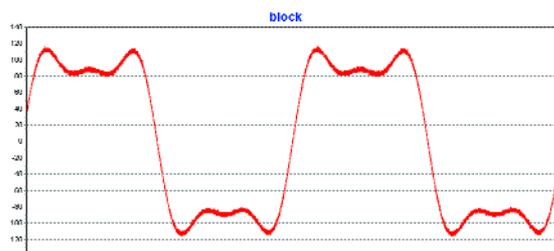


Figure 4. Et, pour finir, le signal rectangulaire mesurée (2 kHz également).

Mode d'emploi de la carte-son

L'unité *MMSystem* comporte un certain nombre de fonctions dont nous avons besoin pour activer la carte-son. La fonction la plus importante est une instruction servant à l'exécution d'un fichier :

```
PlaySound(<param1>,<param2>,<param3>)
```

Dans le cas le plus simple, param1 est le nom de fichier d'un fichier .wav, c:\winnt\media\chimes.wav par exemple.

Il est également possible de prendre ici l'adresse mémoire (*pointer*) d'un fichier .wav se trouvant en mémoire. C'est l'option que nous utilisons dans ce programme. Un *MemoryStream* possède une propriété **memory** du type **pointer**. C'est la raison du choix de ce type de stream. Le premier paramètre prend alors la forme suivante :

```
PlaySound(WaveStream.memory,<param2>,<param3>)
```

Param2 est 0, à moins que le son ne provienne d'une

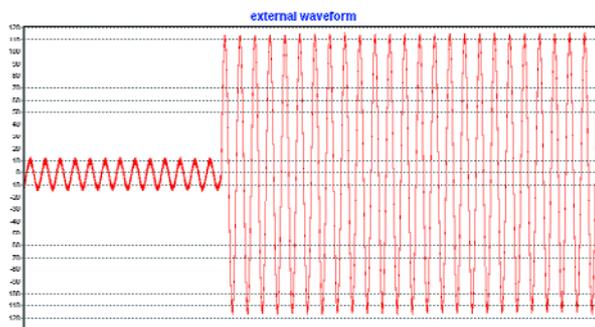


Figure 5. Partie d'une forme d'onde créée sous Excel. La période entière comporte un signal de 1 Hz, à une amplitude de 100% pendant 50 ms suivi d'une amplitude de 10% pour les 50 ms suivantes. La fréquence d'échantillonnage est de 50 kHz.



Figure 6. Structure binaire d'un SpeedButton.

ressource. Nous n'entrons pas dans le détail ici. Par un choix judicieux de param3 nous indiquons à la carte-son que param1 n'est pas un fichier mais un pointeur vers une adresse en mémoire. Param3 requiert une combinaison de 3 variables définies dans l'unité MMSys-tem. SND_MEMORY indique que param1 est une adresse en mémoire, SND_LOOP force une reproduction sans fin du fichier et SND_ASYNC indique que la reproduction doit se faire de façon autonome. Ce dernier point est requis en combinaison avec SND_LOOP. À chaque fois que nous voulons envoyer un flux de données déguisé en fichier .wav nous utilisons la fonction :

```
PlaySound(WaveStream.Memory, 0, SND_MEMORY
OR SND_LOOP OR SND_ASYNC);
```

Pour stopper la fonction nous utilisons playSound épaulé par d'autres paramètres :

```
PlaySound(NIL, 0, 0)
```

Le premier paramètre étant un pointeur, nous ne pouvons pas mettre un 0 sans autre forme de procès. Il existe à cet effet une sorte de pointeur zéro baptisé NIL. Il reste, pour finir, à paramétrer l'amplitude du signal. Nous le faisons par le biais d'une fonction couplée à la Wave Amplitude Control de Windows. Il nous faut pour cela une variable de 4 octets qui paramètre le niveau du signal. Les 2 octets de poids fort définissent la taille du signal du canal gauche, les 2 autres octets celle du canal droit.

On a défini 2 fonctions pour pouvoir paramétrer les 2 canaux indépendamment. On commence, à l'aide de **waveoutgetvolume**, par lire l'ancien paramétrage de volume. On remplace ensuite les 2 octets de poids fort ou faible, cette variable étant ensuite restockée avec **waveoutsetvolume**.

Normalement, le réglage de volume est logarithmique. Nous n'avons pas tenu compte de ce point lors de la présentation de l'amplitude en pourcents du signal maximum.

Maquette

Nous avons apporté une attention particulière à la présentation. Nous utilisons, pour la première fois, des boutons dotés d'images. Nous avons opté ici pour des speedbuttons en raison de certaines de leurs propriétés intéressantes. Il est possible, en fonction de l'état du bouton, de visualiser une image. Il est en outre possible, par action sur un bouton, de se « substituer » à un autre. Si l'on veut désactiver d'autres boutons il faut qu'ils soient conscients de leurs existences respectives, ce que nous pouvons faire à l'aide de la propriété **groupindex**. En cas d'action sur un bouton, les autres boutons du même groupindex sont relâchés. La solution idéale si nous ne voulons activer qu'un seul bouton, comme lors du choix d'une forme d'onde (groupindex >0).

Il est possible d'ajouter des images à l'aide la propriété Glyph. Un glyph (caractère ou symbole sculpté) est un bitmap comportant un maximum de 4 images, de taille identique et adjacentes sur une rangée horizontale. Le SpeedButton montre l'une de ces images en fonction de l'état du bouton (cf. **figure 5**).

Image 1 Bouton relâché

Image 2 Bouton désactivé (disabled)

Image 3 Le bouton a été cliqué. Si groupindex=0 on voit réapparaître ici l'image 1.

Image 4 Le bouton 1 reste activé.

Il nous reste à informer le speedbutton qu'il y a 4 images dans le glyph, ce qui est fait par la propriété **Num-Glyphs=4**.

Conclusion et résultats

Il est possible, en tenant compte des limitations dues à la définition du fichier .wav (fréquence d'échantillonnage de 64 kHz max.), de réaliser un joli générateur de fonctions. Comme il est possible de créer les fichiers-source en externe dans un tableur, on dispose d'un nombre illimité de formes d'ondes, ce qui rendent ce générateur de formes d'ondes quasiment unique.

Nous pourrions étendre le générateur par l'utilisation des 2 canaux (stéréo). Les canaux gauche et droit sont alors programmés individuellement. Cela offre la possibilité, par exemple, dans le cas de signaux complexes, de programmer le signal sur l'un des canaux, l'autre canal véhiculant un signal de déclenchement à temporisation programmable.

L'analyse du signal sinusoïdal à l'aide d'un analyseur de spectre (Agilent 4395A) s'est traduite par des résultats étonnamment bons :

Rapport S/B :	>95 dB
Harmonique d'ordre 2 :	<-65 dB
Harmonique d'ordre 3 :	<-80 dB

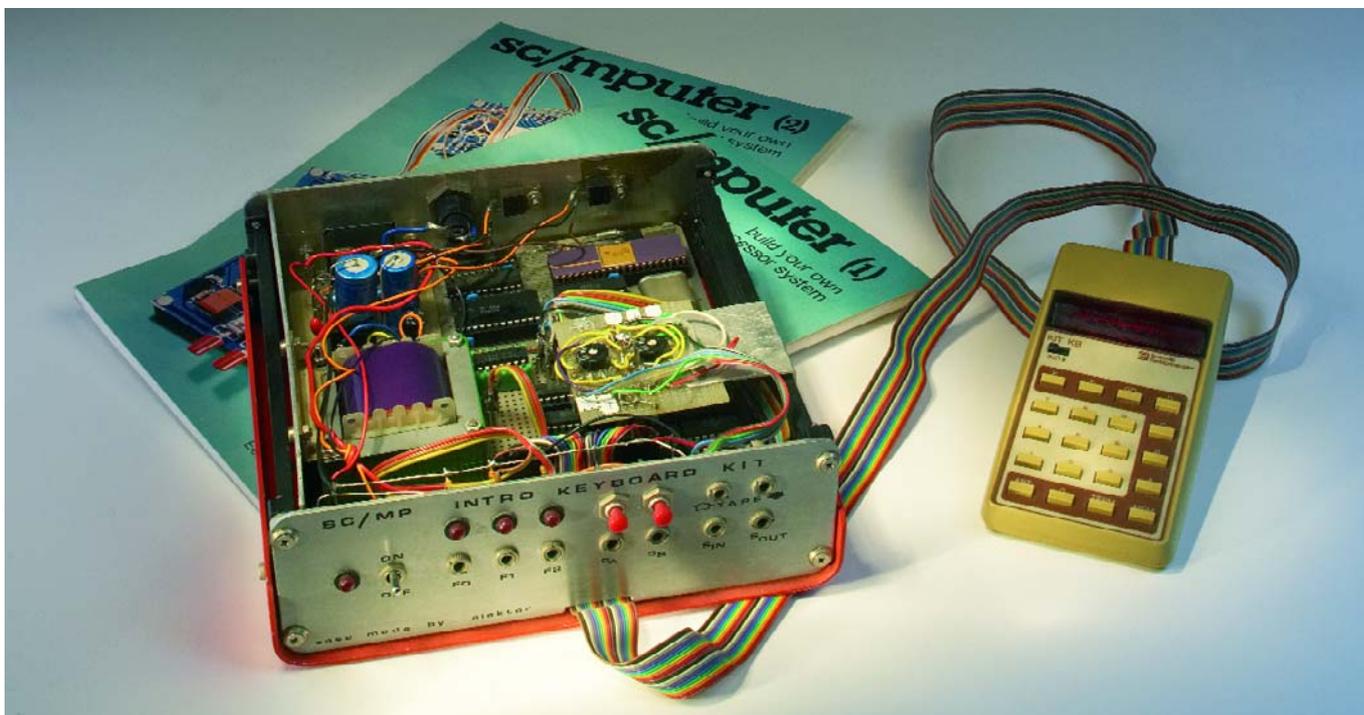
Mesuré sur un signal de 1 kHz à 0 dBm.

Ces chiffres concernent un ordinateur pris au hasard et doté d'une carte-son intégrée à la carte-mère. D'autres cartes peuvent produire des résultats très différents.

À noter que certaines cartes-son déforment beaucoup à modulation maximale. Il faudra y faire attention si la déformation joue un rôle. On pourra, le cas échéant, fixer la modulation maximale à une valeur plus faible. Les formes d'ondes illustrant cet article ont été prises à l'aide d'un oscilloscope numérique (leCroy WaveRunner LT344L).

Vous pouvez télécharger le code-programme complet de cet article depuis notre site www.elektor.fr (avril 2005, n° **040240-4-11**).

(040240-4)



SC/MP Computer d'Elektor (1978)

Jan Buiting

Il n'y a pas le moindre risque à affirmer que toute personne tenant à savoir ce qui se passe dans le monde de l'électronique se doit impérativement de s'intéresser aux microprocesseurs. Les implications de la révolution entraînée par les microcontrôleurs est telle que si l'on ne le fait pas, on sera inévitablement dépassé très rapidement. [...]

La préface d'un nouveau magazine parlant des microcontrôleurs 512 bits tournant à 300 GHz les plus récents ? Une pub agressive de Microchip ou d'Atmel ?

Non, vous venez juste de lire ce qui aurait pu être la préface à une série d'articles démarrant dans le n° 1 d'Elektor (mai/juin 1978) et décrivant le système à microcontrôleur SC/MP.

L'impact de cette série fut énorme, rare furent en effet les magazines concurrents ayant pris le risque de s'attaquer aux

µP. La controverse alla de bon train, les inconditionnels de l'approche analogique et numérique déclarant que mettre le microprocesseur dans les mains de Mr-tout-le-monde serait la fin de l'électronique amateur.

L'acronyme SC/MP signifie *Simple Cost-effective MicroProcessor*. La dénomination officielle de ce produit de National Semiconductor était ISP-8A/500D. Une version ultérieure de ce composant, en technologie NMOS, fut baptisée ISP-8A/600D.

Le SC/MP, habillé d'un boîtier DIP à 40 broches fait partie de la catégorie des tous premiers microprocesseurs, bâtis selon la structure traditionnelle de registres, un accumulateur et la capacité d'adresser de la mémoire externe. C'était là l'atout de l'époque que National Semiconductor pour s'assurer une partie du marché. Le succès ne fut pas au rendez-vous, d'autres acteurs tels qu'Intel, Motorola, Signetic et même RCA s'étant présentés dans cette arène quasiment vide à ce moment-là.

Les bruits courent que le Rédacteur en Chef tout puissant d'Elektor de l'époque décida d'opter pour le SC/MP plutôt que

pour le 6800 de Motorola, pour la (bonne ? et) simple raison que le produit de NS était meilleur marché !

Dès le départ, Le SC/MP d'Elektor était un système encombré, en dépit des efforts de l'époque d'en ordonner les plans de câble du mieux possible. Une photo du prototype montre une « paire de cartes interconnectées par une jungle de connexions ». Pour le lecteur lambda de l'époque les choses devaient être encore plus complexes.

Quoi qu'il en soit, il fonctionnait, les lecteurs gravaient et perceaient avec ferveur leurs propres circuits imprimés, entraient à la main les longs listages hexadécimaux proposés par pages complètes dans le magazine et se mirent à envoyer leurs propres programmes et leurs extensions-maison à la rédaction d'Elektor. En quelques mois, le SC/MP devint célèbre et de temps à autre un lecteur d'un certain âge nous fit savoir que le SC/MP était le premier ordinateur qu'il ait construit pour impressionner ses amis et autres membres de sa famille...

La publication du système SC/MP de base fut suivie de nombreux

articles « ni chair ni poisson » parlant de sujets aussi disparates que BASIC, un désassembleur, SC/MP chante Noël, un modulateur HF, une interface d'enregistrement 300 bauds pour magnétophone, une carte de 4 Koctets de RAM, sans oublier le fameux moniteur Elbug. La photographie d'illustration montre, il me semble, un système construit autour d'une carte de développement SC/MP originale fournie par National Semiconductor à Elektor. Le système semble avoir été conçu spécifiquement pour la lecture et l'écriture de cassettes de magnétophone et il se peut qu'il soit le système de production des bandes Master pour le Service ESS (Elektor Software Service) qui à l'époque proposait les programmes pour le SC/MP sur des disques vinyle 45 tours. Quelqu'un a-t-il encore des copies de l'ESS 001 et 002 ?

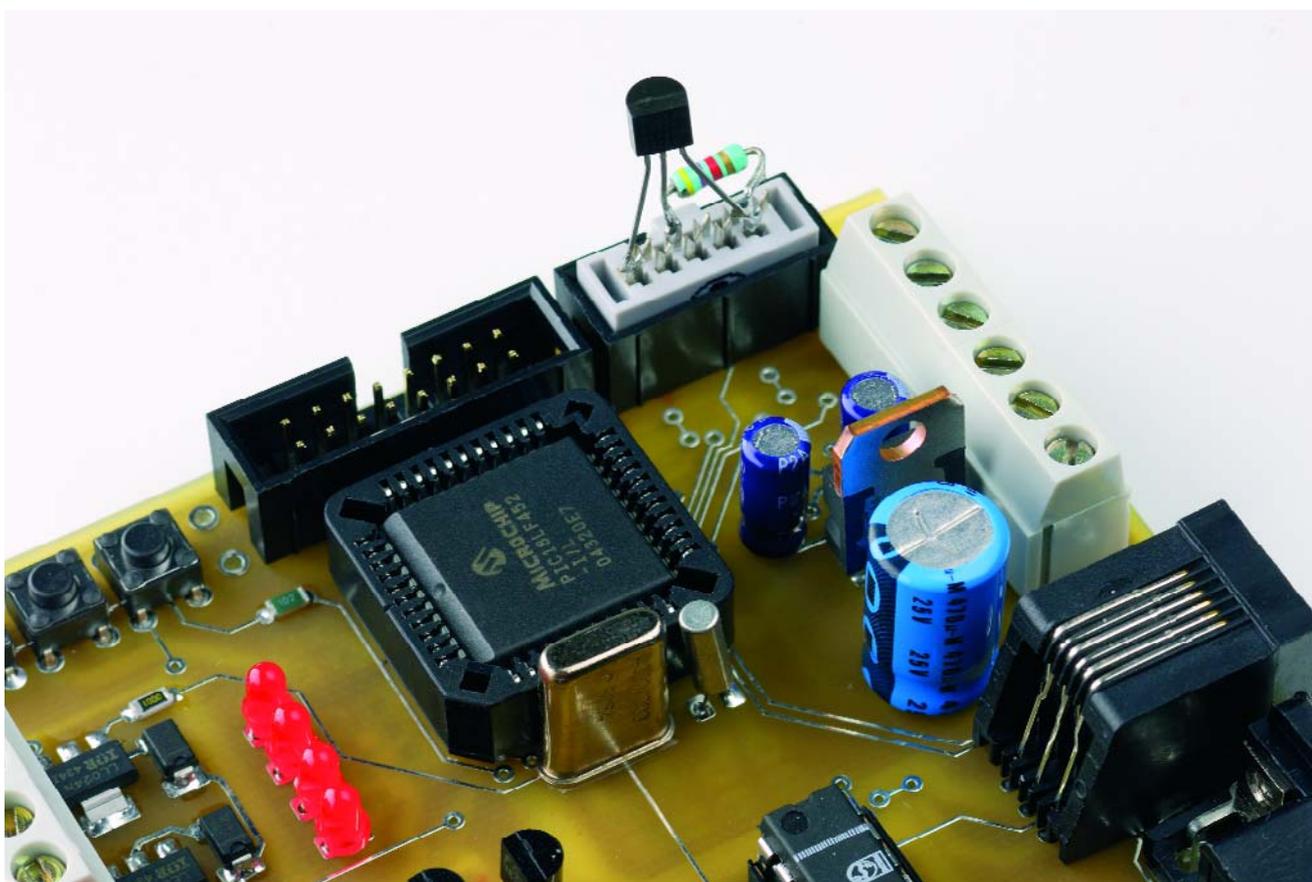
(045113-1)

Rétronique est une colonne mensuelle s'intéressant à de l'électronique du siècle dernier y compris des montages de légende décrits dans Elektor. Nous sommes ouverts à toutes les contributions et suggestions. N'hésitez pas à nous envoyer un E-mail à redaction@elektor.fr, sujet : Rétronique Elektor

TÉLÉ-ENREGISTREUR DE TEMPÉRATURE POUR LA CARTE PIC18F

À base de capteur « 1 fil » DS1820

Peter Moreton



Ce bref article, basé sur la carte de développement PIC18Flash décrite dans le numéro de janvier 2005, indique comment manipuler un capteur DS18B20 ou DS18S20 au niveau du bit par microprogramme PIC en « C » et enregistrer les mesures de température sur un PC par le port RS-232.

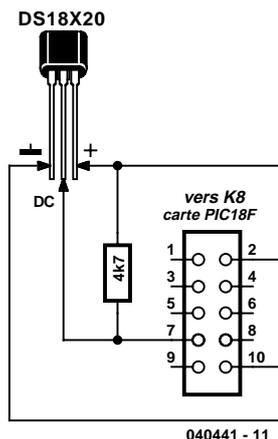


Figure 1. Schéma du matériel de l'enregistreur de température à distance – tous les éléments sont raccordés à K8, sur la carte de développement du PIC18F Flash.

Le connecteur « K8 » de la carte de développement PIC18Flash décrite dans Elektor de janvier 2005 comporte les broches processeur nécessaires pour communiquer par SPI/Microwire, I2C et « 1 fil » avec des appareils externes. Des modules matériels dédiés du PIC gèrent les protocoles SPI/Microwire et I2C. La bibliothèque C18 offre des fonctions standard d'accès au matériel qui simplifient grandement la tâche. Le PIC ne comporte pas de module matériel dédié pour le bus « 1 fil » Dallas/Maxim. Il faut recourir à des « manipulations au niveau du bit » (bit-banging) en positionnant la broche appropriée du microcontrôleur à l'état haut ou bas selon le protocole « 1 fil ».

Le bus « 1 fil »

Le bus « 1 fil » Dallas/Maxim n'utilise qu'une seule ligne de données (d'où « 1 fil ») pour envoyer et recevoir les données. Chaque unité « 1 fil » doit être aussi raccordée à une masse commune et, si nécessaire, à une alimentation. L'unité « 1 fil » peut en effet se passer de celle-ci en « empruntant » la puissance nécessaire à la ligne de données. Le bus « 1 fil » est normalement maintenu à l'état haut par une résistance de 4kΩ7 raccordée à V_{dd}. Les appareils envoient les bits de données en plaçant le bus à l'état bas puis en le libérant pour qu'il retourne à l'état haut. La broche RA4 à drain ouvert du PIC18F452 est particulièrement bien adaptée au protocole « 1 fil ». Elle per-

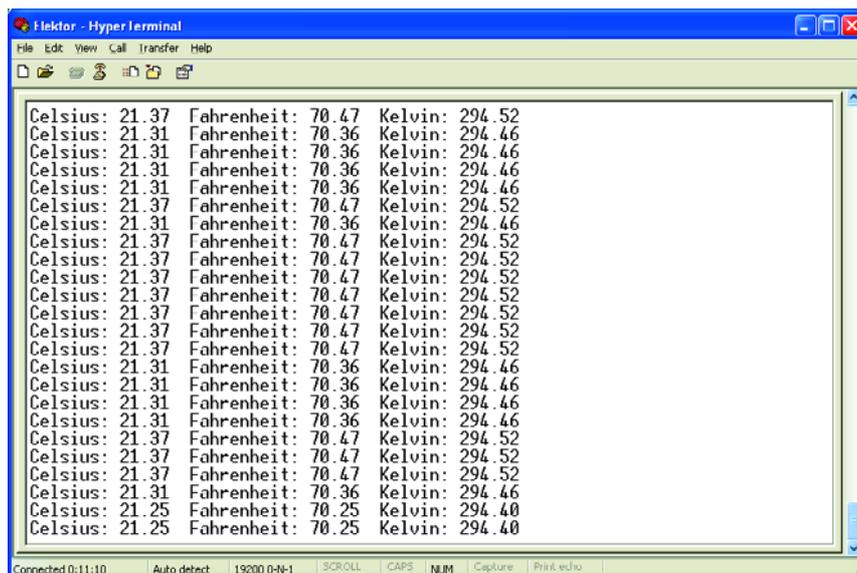


Figure 2. Saisie d'écran montrant le logiciel de l'enregistreur de température en fonctionnement.

met de « libérer » le bus et de le positionner à l'état haut par la résistance de rappel 4kΩ7.

Le protocole du bus est basé sur un certain nombre de types de signaux : une impulsion de réinitialisation, une impulsion de présence, l'écriture ou la lecture d'un 0, l'écriture ou la lecture d'un 1. Tous les systèmes « 1 fil » peuvent engendrer des intervalles de temps (time-slots) d'une grande exactitude. Chaque signal est en effet soumis à une synchronisation temporelle rigoureuse. Les lecteurs désireux d'avoir un aperçu détaillé de la signalisation « 1 fil » consulteront avec profit la documentation exhaustive du site Web de Dallas/Maxim.

Capteurs de température DS18x20

La série DS1820 comporte plusieurs composants similaires ; nous avons réalisé un microprogramme PIC pour le DS18S20 dont la résolution vaut 9 bits/0,5 °C et le DS18B20 dont la résolution atteint 12 bits/0,0625 °C. Ces 2 composants sont précis à 0,5 °C près. Pour utiliser l'un ou l'autre, appeler la fonction appropriée en éditant le fichier source DS1820.C si nécessaire :

```
Celsius =
Read_Temperature_DS18S20();
// use DS18S20 device
```

ou

```
Celsius =
Read_Temperature_DS18B20();
// use DS18B20 device
```

Le fichier HEX déjà compilé inclus dans le téléchargement gratuit concernant le présent article (EPS040441-11.zip) est prévu pour le composant DS18B20 à la résolution la plus élevée.

Description du circuit

Le capteur de température Dallas DS18S20 ou DS18B20 en boîtier TO92, représenté dans la figure 1 avec une résistance 4kΩ7 de rappel vers le niveau haut, est raccordé comme suit au connecteur K8 :

- broche 2 de K8 à la broche 3 du DS1820, V_{dd} ;
- broche 10 de K8 à la broche 1 du DS1820, masse ;
- broche 7 de K8 à la broche 2 du DS1820, DQ ;
- monter une résistance de rappel vers le niveau haut de 4kΩ7 entre DQ et V_{dd}.

Le câble d'interconnexion, qui peut atteindre plusieurs mètres, permet d'effectuer des mesures à distance. Une gaine thermorétractable sur un matériau imperméable à l'eau recouvrant un DS1820 utilisé à l'extérieur lui permettront de supporter l'assaut des éléments.

(040441-1)

DÉTECTEUR D

La boisson rend-elle violent ?

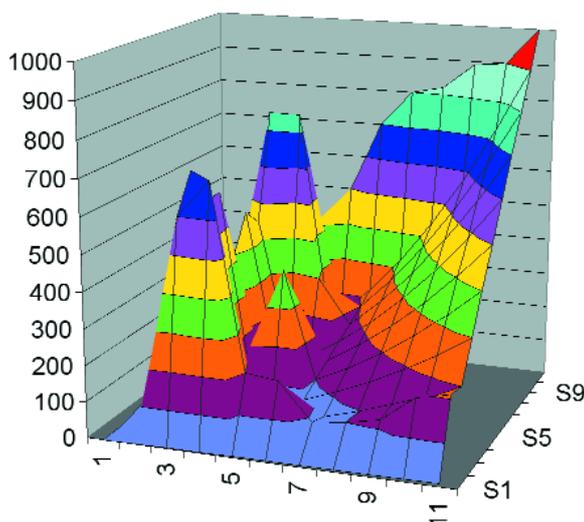
Peter van Herpen

L'université de Heidelberg a découvert tout récemment un nouveau type de particules qu'ils ont baptisé brachions, nom dérivé du latin « brachius » qui signifie avant-bras. L'Allemagne connaît une expression idiomatique, « *Brachiale Gewalt* » qui signifie avec force violence. On en déduit qu'à l'évidence les brachions sont liés à la violence.

Les scientifiques ont prouvé qu'une concentration importante de brachions dans le sang augmentait le risque de voir une personne devenir violente. C'est là que se pose la question de savoir comment les brachions finissent dans le sang. De nouvelles recherches ont démontré que les brachions se trouvaient à concentration importante plus particulièrement (mais pas uniquement) dans les boissons alcoolisées.

Ce qui surprend est que la relation entre la concentration de brachions et le taux d'alcool est loin d'être linéaire. Il apparaît en outre que l'effet des brachions dépend de facteurs tels qu'âge, index grasseux du corps (BMI = *Body Mass Index*) et sexe.

La relation entre ces facteurs est illustrée par le graphique 1 (pour les personnes de sexe féminin avec un BMI compris entre 20 et 30).



L'université a mis au point un montage intéressant permettant de mesurer la concentration de brachion dans un liquide quelconque et visualisant leur effet potentiel sur un afficheur, électronique plus particulièrement destinée aux cliniques et aux prisons.

La mesure de la concentration de brachions fait appel à 2 capteurs, un capteur infrarouge (IR) mesurant l'absorption de la lumière IR, un second capteur déterminant la conductivité. À partir de ces 2 facteurs, on peut calculer, à l'aide d'une formule complexe, la concentration de brachions. Cette formule a été convertie en programme.

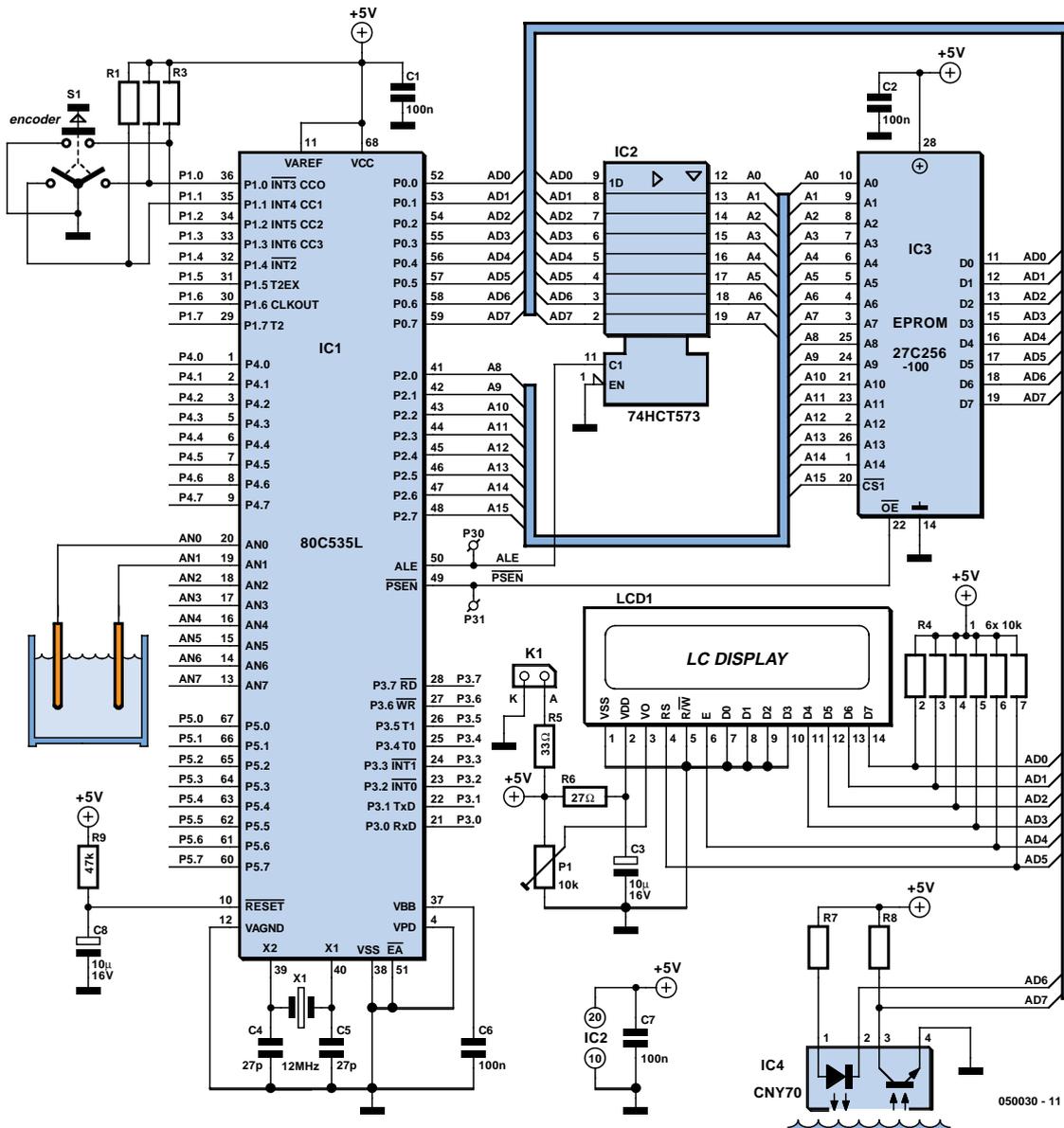
L'électronique

Nous avons conçu le présent montage à l'intention de ceux d'entre nos lecteurs qui voudraient expérimenter l'effet des brachions. Le cœur de l'électronique prend la forme d'un 80C535, un processeur de chez Siemens tournant à 12 MHz. Les capteurs, une barrière lumineuse à réflexion et un capteur standard, l'attaquent directement. Le schéma n'a rien de bien sorcier. L'affichage est connecté directement au port 0 au travers de résistances pull-up, vu que l'on se trouve en présence de sorties à drain ouvert, le bouton-poussoir/rotacteur étant lui relié à quelques entrées du port 1. On pourra mettre le progiciel dans une EEPROM; en raison du caractère expérimental de cette réalisation nous ne pouvons pas proposer d'EEPROM toute programmée.

Mode d'emploi

On commence, à l'aide du bouton-poussoir/rotacteur, par choisir le sexe de la personne à « examiner », puis on entre sa taille, son poids et son âge. À partir de ces éléments, le programme calcule le BMI et sélectionne les paramètres corrects pour le graphique. Ce qu'il ne faut pas confier au « cobaye » est que dans la position « Female » l'âge ne va pas au-delà de 35, vu qu'il semblerait que les

E BRACHIONS



femmes ne puissent pas être plus âgées que cela, et que le poids est « plombé » de 15% vu qu'il est connu que les femmes ne disent jamais la vérité à ce sujet. On plonge ensuite les capteurs dans le liquide à examiner, les mesures pouvant alors débuter. Il faut de l'ordre de 10 s avant que ne commencent les mesures. Ces processus sont visualisés sur l'affichage. Le résultat est exprimé en brachions par litre. Une valeur typique pour du vin rouge (français et plus jeune que 1995) est de $4,37 \cdot 10^{-12}$ brachions/litre. Une nouvelle action sur le bouton affiche l'effet de cette concentration après consommation de 100 ml. Avec chaque nouvelle action sur le bouton, cette quantité est incrémentée par pas de 0,1 litre. Pour les femmes ces étapes vont de « n'a aucune » à « n'a plus aucune » en passant par « a bien envie », pour les hommes de « n'a aucune » à

« hors d'état de nuire » en passant par « paraît plus grand que la réalité », « ne veut vraiment plus » et « violent ». Le programme connaît bien nombre d'autres pas intermédiaires. Il va sans dire que le test de ce programme a constitué un défi indéniable, et nous tenons à remercier la société Ricard pour son aide « bien venue ». Les lecteurs intéressés par ce programme peuvent envoyer un E-mail à redaction@elektor.fr en mettant comme sujet : « détecteur de brachions - numéro d'avril ».

(050030-1)

Littérature :

Brachions : nouvelles particules élémentaires et leurs effets, Prof. Friedrich Von Witzstein, 2004, Heidelberg (en allemand)

TECHNIQUE DE MES

Ralf-P. Hazebrouck Ing. Dipl.

Le bus USB était-il encore trop lent pour la technique de mesure dans la version 1.0 ou 1.1 ? La largeur de bande 40 fois plus élevée de la version 2.0 permet de capturer signaux et données à 480 Mbits/s.

L'acquisition de données de mesure par USB facilite le nomadisme des applications et améliore la capacité d'utilisation immédiate sans interruption du système en fonctionnement (*Hot Plug&Play*). Les travaux de mesure, considérablement simplifiés, gagnent énormément en flexibilité. La compatibilité amont avec les versions précédentes du bus USB est complète : seule la vitesse de transfert est affectée. Les convertisseurs USB/GPIB permettent même

de raccorder des appareils de mesure basés sur GPIB. Presque tous les ordinateurs portatifs ne sont pas équipés que de ports USB, mais encore d'un ou même de 2 logements PCMCIA permettant d'insérer des cartes CompactFlash (CF) de type I ou II. Les cartes CF peuvent être utilisées dans les ordinateurs bloc-notes au moyen d'adaptateurs PCMCIA très bon marché. Les CF, conçues initialement comme mémoire de masse pour caméras, PDA, etc., peuvent aussi servir de plate-forme de mesure nomade sur les ordinateurs bloc-notes et les PDA. Le raccordement direct d'appareils de mesure au port USB ou à un logement PCMCIA est très apprécié des techniciens chargés des mesures. L'assemblage bien plus simple du dispositif de mesure offre une extrême flexibilité. Un avantage de plus : les appareils de mesure sont directement alimentés par le logement PCMCIA ou le bus USB (jusqu'à 100 mA environ). Pas besoin d'une autre source de courant. Entre-temps, un nombre considérable d'appareils de mesure (multimètres numériques (DMM), oscilloscopes numériques à mémoire (DSO), analyseurs logiques et systèmes d'acquisition des données de mesure) ont vu le jour. Ils permettent au technicien d'effectuer des mesures test sur place avec son ordinateur bloc-notes ou portatif. Pour ce qui est des performances, certains de ces appareils n'ont pas grand chose à envier aux appareils autonomes.

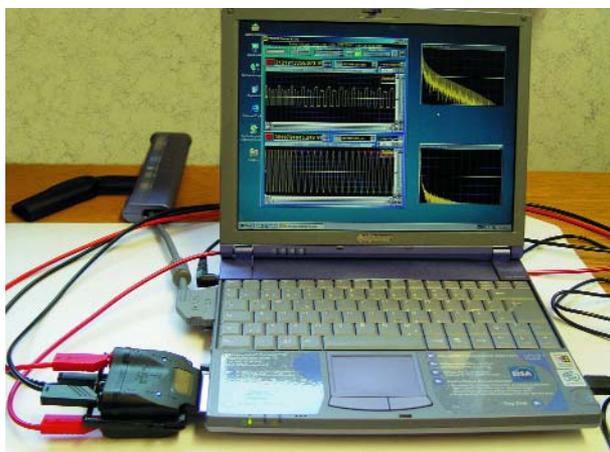


Figure 1.
DMM à 5 chiffres $\frac{1}{2}$
au format
CompactFlash.



Figure 2.
Le Penscope n'est
pas plus gros qu'un
stylo-bille mais
mesure jusqu'à
40 MHz.

DMM miniature en format CompactFlash

Les cartes CompactFlash (cartes CF) ont une taille plus petite qu'une demi-carte bancaire (43 x 36 mm²) et sont épaisses de 3,3 mm (type I) ou 5 mm (type II). Les cartes sont munies d'un connecteur 50 broches assurant le transfert des données et la connectivité. Les cartes CompactFlash standard peuvent fonctionner à 3,3 ou 5 V, donc avec presque tous les PDA ou ordinateurs portatifs. Le multimètre numérique CFD200 à 5 chiffres $\frac{1}{2}$ en format type 2 de Meilhaus Electronic permet de mesurer le courant, la tension et la résistance (**figure 1**). Utilisé avec des PDA ou un adaptateur CF dans le logement PCMCIA des ordinateurs bloc-notes/portatifs, il offre une solution idéale pour des tâches de mesure. Guère plus gros qu'une souris d'ordinateur, il est aussi à l'aise dans les mesures de tension que dans celles de courant et de résistance. Les

URE USB

Dispositifs de mesure flexibles par USB et PCMCIA

2 canaux indépendants permettent de mesurer des tensions jusqu'à 668 V DC ou AC (véritable valeur efficace) dans 4 plages. La bande passante pour les mesures AC atteint 6 kHz et la résistance interne pour les tensions DC et AC s'étend de $>10\text{ M}\Omega$ à $100\text{ M}\Omega$. La valeur mesurée du courant continu et alternatif ne doit pas dépasser 350 mA. Il est possible de programmer un pré/post déclenchement pour effectuer ces mesures et de mémoriser jusqu'à 16 384 valeurs de 24 bits chacune dans une mémoire tampon. Les canaux sont galvaniquement isolés l'un de l'autre et par rapport au PDA/PC jusqu'à 2,5 kV pour les protéger des pointes de surtension. La mesure de résistance de $1\ \Omega$ à $>40\text{ M}\Omega$ est divisée en 4 plages. Les lignes de mesure sont raccordées à quatre prises bananes de 4 mm. L'instrument est protégé par un boîtier en polycarbonate à l'épreuve des chocs. Il permet d'effectuer des mesures à température ambiante de -10 à $+45\text{ }^\circ\text{C}$.

DSO 40 MHz

Penscope, un oscilloscope numérique à mémoire 40 MHz est le plus petit instrument de mesure USB de la gamme de Meilhaus Electronic. Il s'agit d'un mini-oscilloscope supercompact à 1 canal. Sa résolution de 10 bits lui permet de mesurer des tensions atteignant $\pm 50\text{ V}$ (**figure 2**). La plage d'entrée atteignant 10 V/division comporte 7 pas d'ajustement. La base de temps est ajustable de $0,1\ \mu\text{s}/\text{division}$ (40 Méchantillons/s) à $50\text{ ms}/\text{division}$ (800 échantillons/s) en 19 pas. Le *Penscope* possède une pointe de mesure amovible à ressort. La pointe peut être remplacée par un adaptateur BNC (inclus dans la livraison). Des sondes standard en option $\times 1$ et $\times 10$ pour les niveaux plus élevés sont disponibles en option. Les déclenchements (*trigger*) hardware incorporé comme Auto, flanc positif et négatif ou *Largeur d'impulsion plus grande ou plus petite* que facilitent l'acquisition du signal. Le logiciel Windows offre les fonctions oscilloscope, voltmètre et fréquencemètre.

Analyse de signaux 150 MHz par USB

Le boîtier DSO USB de la famille M52x, à peine plus gros que 2 paquets de cigarettes (**figure 3**), permet de transformer un ordinateur portable en un DSO portable 2 canaux atteignant 150 MHz. Le boîtier USB est disponible en 2 versions (60 MHz ou 150 MHz) avec des fréquences d'échantillonnage de 50 Méchantillons/s ou 100 Méchantillons/s. Chaque canal possède une capacité de mémorisation de 8 Koctets. D'autres modèles jusqu'à 200 MHz et 200 Méchantillons/s sont à l'étude. Un CAN (ADC) 8 bits numérise les signaux saisis avec une erreur maximale de $\pm 2\%$ de la valeur traitée. Il est possible de mesurer directement $\pm 200\text{ V}$ au maximum avec une résistance de $200\text{ M}\Omega$. Les coefficients de déviation de $10\text{ mV}/\text{division}$ à $5\text{ V}/\text{division}$ peuvent être

ajustés selon la séquence 1-2-5. La base de temps peut être ajustée de $500\text{ ps}/\text{division}$ à $800\text{ ms}/\text{division}$. Le boîtier métallique robuste comporte un connecteur USB ainsi que 2 prises BNC pour le raccordement du signal et une entrée de déclenchement externe (de -10 à $+13\text{ V}$, 20 kHz). Ce petit DSO offre de nombreuses possibilités de déclenchement : outre le réglage d'un déclenchement à 2 niveaux (*Dual Level Trigger*), il est possible de choisir une source de déclenchement de niveau primaire, secondaire ou externe. Le déclenchement est provoqué par un flanc ascendant ou descendant avec l'option pente (*Slope*). Il est provoqué par la longueur d'impulsion (période de l'impulsion $>5\text{ ns}$) en choisissant l'impulsion de déclenchement. Le réglage continu de la position de déclenchement permet de définir le début de la saisie avant ou après le déclenchement. L'alimentation du DSO est assurée par le bus USB. La faible consommation (USB 1.1 : 350 mA, USB 2.0 : 450 mA) est idéale pour une utilisation itinérante.



Figure 3.
Un signal de 150 MHz ? Aucun problème pour le DSO USB 2 canaux de la famille M2x.



Figure 4.
Les analyseurs logiques miniatures 8 et 16 canaux sont petits mais recèlent de nombreuses possibilités de déclenchement.

Analyseur logique miniature

La recherche des erreurs dans les circuits numériques et la représentation du comportement temporel de divers signaux numériques est facilitée par les analyseurs logiques miniatures USB ANT8 (8 canaux) et ANT16 (16 canaux) format poche (dimensions 65 x 35 x 15 mm) (**figure 4**). L'analyseur logique miniature ANT8 possède 8 canaux et une profondeur de mémoire de 3 072 échantillons. Il consomme 1,5 W. Sa vitesse d'échantillonnage est de 500 MHz. L'analyseur logique miniature ANT16 saisit simultanément les signaux sur 16 canaux. Sa profondeur de mémoire atteint 2 048 échantillons par canal. Sa vitesse d'échantillonnage est aussi de 500 MHz. Il consomme 2,2 W. Un câble plat à codage couleur muni de 8 pointes de mesure rouges (canaux logiques) et d'une pointe de mesure noire (masse) avec mini crochet est disponible pour le ANT8.

Le câble équivalent du ANT16 possède 16 connexions rouges et 2 connexions noires ainsi qu'une connexion jaune et une connexion bleue pour Trigger-In/Clock-In et Trigger-Out. Les 2 modules permettent d'effectuer des déclenchements simples ou complexes.

Types de déclenchement disponibles :

- Déclenchement basé sur la forme (*Pattern*)

- Déclenchement basé sur une configuration de bits prédéterminée aux entrées (fonction logique ET)
- Déclenchement basé sur les flancs (*Edge*)
- Déclenchement basé sur des changements de flanc prédéterminés des signaux d'entrée (fonction logique OU)
- Déclenchement complexe (*Advanced*)
- Déclenchement combiné basé sur la forme et les flancs, complété par des fonctions de comptage et de temporisation.

Ces types de déclenchement sont rendus possibles par une logique de déclenchement multi-états (*multi-state*) basée sur les flancs, la forme et des conditions complexes. Le déclenchement complexe comporte un comptage d'événements et peut mesurer la durée minimale et maximale. Les analyseurs miniatures sont prêts à être branchés (*Plug&Play*). Le logiciel tourne sous les versions actuelles de Windows XP, 2000 et 98/Me. Un logiciel simulant le panneau frontal d'un analyseur logique est inclus dans la livraison. Il permet d'enregistrer les données sous forme de fichiers texte ou CSV et d'imprimer les formes d'ondes.

(040454-1)

Ralf-P. Hazebrouck Ing.-Dipl.

Meilhaus Electronic – www.meilhaus.com



Améliorez vos fusées à eau

Ivan Lanoë et Michel Mollo

L'objectif de cet ouvrage est d'aller plus loin que la maîtrise de la construction et du lancement, en améliorant les fusées à eau. La première étape passe, avant tout, par l'observation et la caractérisation de chaque vol.

Pour vivre cette nouvelle aventure, plusieurs pistes sont présentées, de la simple observation visuelle, chronomètre en main, à la caméra couleur embarquée en passant par un système d'acquisition électronique géré par microcontrôleur...

Altitude, trajectoire, vitesse, humidité et température dans la fusée, accélération, déclenchement d'un parachute... seront bientôt à la portée de chacun grâce à cet ouvrage guidant le lecteur, étape par étape, dans la réalisation de tous ces systèmes embarqués et dans une aventure qui se révélera aussi enrichissante que ludique.

Sommaire

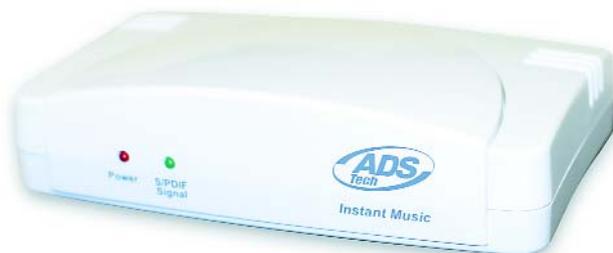
- Caractériser le vol de la fusée sans système embarqué
- Chronologie détaillée de la réalisation et de l'utilisation de notre système embarqué
- Développer un système de récupé-

ration fiable en prévision de l'embarquement de matériels coûteux et fragiles

- Accéder simplement aux conditions de l'intérieur de la fusée
 - Doter la fusée d'un système d'acquisition embarqué autonome
 - Mesurer l'accélération de $-2g$ à $+2g$, les vibrations et l'inclinaison de la fusée
 - Mesurer la température, la pression dynamique et estimer l'altitude et la vitesse
 - Déclencher l'acquisition des données à l'aide d'une télécommande
 - Un système de programmation des microcontrôleurs
 - Utiliser l'interface série pour relier la fusée à l'ordinateur
 - Exploiter les données enregistrées au cours du vol
 - Embarquer des caméras couleur.
- Ivan Lanoë est professeur de Technologie en collège, il est également l'auteur de « Construisez et lancez des fusées à eau ».

Michel Mollo est ingénieur et professeur agrégé en électronique. Ouvrage recommandé par le Centre National d'Études Spatiales (CNES).

(047226-1)



Instant Music

Le gardien de vos souvenirs

Ne jetez plus vos vinyles, ni vos cassettes audio ! Grâce à Instant Music, il vous est désormais possible d'enregistrer et de bonifier sur votre ordinateur vos enregistrements d'une génération précédente.

Offrez une nouvelle vie à vos cassettes audio et disques vinyles... et ce en toute simplicité.

Finis les petites craintes injustifiées de tous ceux qui appréhendent la technologie moderne !

Instant Music offre à tous les mélomanes la possibilité d'enregistrer simplement et rapidement les différents types d'enregistrement de leur discothèque sur leur ordinateur.

Pour cela, il suffit de connecter Instant Music au port USB 1.1 ou USB 2.0 de l'ordinateur et de relier ce périphérique à une table de lecture pour disques vinyle, à une platine cassette voire à toute autre source audio.

Pour procéder à l'enregistrement rien de plus facile :

- Choisir le format d'enregistrement : MP3, WMA, WAV ou ACC
- Démarrer la lecture du disque vinyle ou de la cassette audio

- Enregistrer les données en temps réel sur votre ordinateur sans pour autant que cela ne vous empêche de surfer sur Internet ou de travailler sur une autre application.

Instant Music est ainsi une solution idéale pour optimiser la gestion de l'espace mémoire de l'ordinateur et transporter avec soi tous ses meilleurs albums que cela soit sur CD, clé USB ou directement sur votre baladeur MP3.

Instant Music est fourni avec le Pack Nero Audio Suite conçu pour archiver et restaurer les données audio via des fonctions simples et rapides d'utilisation pour :

- Créer des fondus enchaînés, mixer les titres et ajouter des effets audio
- Créer des fichiers audio grâce à des méthodes de filtrage et d'optimisation sonores
- Lire, graver les fichiers audio et les convertir d'un format d'encodage à un autre.

Disponibilité : Janvier 2005
Prix Public Conseillé : 59 € TTC
Garantie : 2 ans

(057010-1)

Développer en 32 bits

Kit de développement ARM7 de Keil

Paul Goossens

La grande majorité des réalisations décrites dans Elektor reposent sur des microcontrôleurs 8 bits. Ces derniers sont dans la plupart des cas suffisamment puissants pour la tâche à remplir. Mais il arrive qu'un 8 bits ne suffise pas à la tâche. Il est temps alors d'élargir ses horizons. Nous allons nous intéresser ici à un kit de développement basé sur un contrôleur 16/32 bits.

Nombre d'appareils électroniques intègrent un contrôleur 8 bits. En dépit de toutes les prévisions optimistes, les 16 bits ne se sont pas (encore ?) substitués aux 8 bits. Mais la génération de contrôleurs 32 bits trouve de plus en plus sa place dans les appareils électroniques modernes. Ces contrôleurs puissants se sont « enfouis » dans les appareils les plus complexes tels que lecteurs de DVD, caméscopes, etc.

Bien souvent on a besoin, sur ce type d'appareils, de l'un ou l'autre système d'exploitation de sorte que le cahier des charges du contrôleur devient de plus en plus lourd. Il existe bien des systèmes d'exploitation pour processeurs 8 bits, mais ils arrivent, en général, bien vite à leurs limites tant en gestion qu'en taille de mémoire.

Les microcontrôleurs 32 bits embarquent souvent nombre de périphériques requis par les systèmes d'exploitation modernes. Ceci tient beaucoup au fait que la tendance dans le monde des appareils grand public est

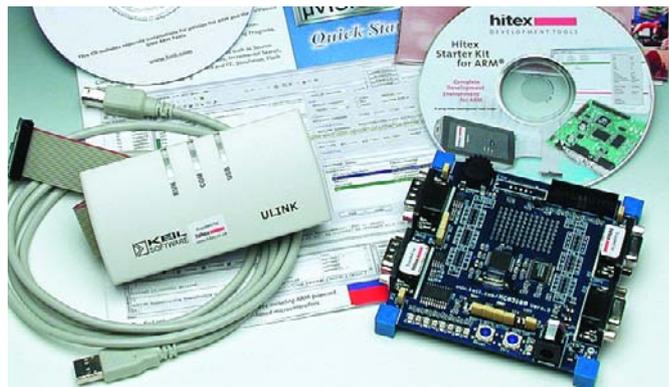
de dériver en direction du PC et d'avoir à remplir nombre de fonctions diverses.

Un bon début

Les personnes intéressées par l'électronique et voulant mettre la main à la pâte (raison pour laquelle elles lisent Elektor), apprécient souvent de pouvoir aborder facilement une nouvelle technologie. Un Starter Kit ou un kit de développement est une bonne base de départ surtout s'il est doté et d'électronique et de logiciels de support.

En ce qui concerne ce dernier point, pas de problème avec la société Keil dont la réputation dans le monde des environnements de développement pour différentes familles de microcontrôleurs n'est plus à faire. Outre ses programmes, elle propose également des cartes de développement permettant à l'utilisateur potentiel de se faire la main rapidement.

Le kit objet de cet article est le MCB2100 à base de microcontrôleur LPC2129. Cette carte est accompagnée d'une interface JTAG à connexion via USB. La partie logicielle



est complète elle aussi : compilateur, éditeur de liens (linker), simulateur et débogueur, tout y est sous forme de version d'évaluation. Ceci implique que le débogueur est limité à 16 Koctets de code et qu'il est interdit d'utiliser le programme créé par le compilateur dans un appareil commercialisé.

Installation

Il est préférable de commencer par installer le logiciel qui accompagne le kit et de ne brancher qu'ensuite l'interface JTAG. De cette manière, Windows XP recon-

naît l'interface automatiquement et l'installation du matériel ne devrait pas poser de problème. Nous avons installé successivement DK-ARM et le compilateur GNU. Une fois Windows redémarré, le logiciel est « bon pour le service ».

Un premier exemple

Encouragés par ces premiers pas prometteurs, nous avons décidé de compiler notre premier exemple pour le tester dans la foulée sur la carte du système. Il y eut quelques problèmes. Le logiciel signale qu'il n'a pas trouvé un cer-

Caractéristiques principales du LPC2129

- Noyau 16/32 bits
- 256 Koctets de mémoire Flash à bord
- 16 Koctets de mémoire RAM à bord
- Programmable ISP (In-System-Programmable)
- Différentes possibilités de débogage intégrées
- 2 interfaces CAN
- 2 ports sériels
- Interface I²C
- 2 interfaces ISP
- 2 temporisateurs (timer) 32 bits
- 6 sorties MLI (PWM)
- 4 convertisseurs A/N 10 bits
- 46 broches d'E/S
- Présence d'une PLL pour réglage de la fréquence d'horloge

Interfaces à bord

- 2 interfaces sérielles
- 2 interfaces CAN
- 8 LED
- Interface JTAG
- Potentiomètre analogique
- Divers signaux disponibles pour applications personnelles

tain programme. Il s'avéra finalement que nous avons choisi un exemple utilisant le compilateur de Keil et que visiblement ce dernier n'est pas, tel que fourni, complet. L'exemple pour le compilateur GNU fonctionna du premier coup. S'il est vrai que la compilation s'était faite correctement, il ne nous fut pas possible de télécharger le programme vers la carte. Pour y arriver il faut commencer par configurer l'utilitaire Flash. On le trouvera sous l'onglet *Flash/Configure Flash Tools/Settings*. Il faut ensuite ajouter dans la nouvelle fenêtre (cf. **figure 2**) le bon algorithme.

Ceci fait, il devint possible de télécharger le programme, mais il ne passa rien. Le processeur paraissait se trouver en état de réinitialisation (*reset*). Il devint possible de démarrer le processeur par le biais du débogueur. Cela nous paraît relativement compliqué. Après quelques recherches (les détails ont leur importance) il apparut que la fenêtre évoquée plus haut comportait une autre option, baptisée *Reset and Run*, qui était désactivée. Il semblerait que ce soit le paramétrage par défaut. Une fois la case correspondante cochée, le programme démarrait immédiatement une fois le téléchargement effectué.

Les autres expériences effectuées se passèrent sans le moindre problème. Il nous fallut constater cependant qu'il fallait reconfigurer les Flash-tools lors de chaque projet. Il suffit de le savoir !

Qu'en pense le jury ?

Le kit de développement comporte tous les ingrédients pour se mettre à « cuisiner » à base de processeur

32 bits. Nous ne pouvons pas, vu les remarques faites plus haut, parler d'un départ en trombe, mais ne fois ces obstacles franchis, à nous les petites... applications personnelles.

Ce qu'il en est vraiment du compilateur de Keil accompagnant ce kit reste flou, car nous n'avons pas réussi à compiler des exemples à l'aide du dit compilateur. Ceci n'est pas un problème insurmontable vu que le compilateur GNU travaille parfaitement lui et que l'environnement de développement de Keil, programmeur, simulateur et débogueur compris, fonctionne comme il faut. Cet environnement soutient même le compilateur GNU, de sorte qu'il n'est pas nécessaire de le quitter pour compiler un programme.

Le matériel est assez spartiate, car nous nous serions attendus, vu les performances du microcontrôleur, à quelques E/S supplémentaires. Certains de signaux en provenance du processeur sont tirés vers l'extérieur de sorte que l'on peut y connecter ses propres extensions. Malheureusement, le circuit imprimé n'est pas doté des dits connecteur qu'il faudra partant commencer par y souder soi-même.

Dans l'ensemble, ce kit fait l'affaire lorsque l'on cherche un bel environnement de développement pour processeurs 32 bits. L'interface JTAG en particulier est très pratique, mais à notre avis légèrement limitée. Surtout si l'on sait que le prix de ce kit est de 249 £ (€ 360 environ) !

(050004-1)

Ce kit a été mis à notre disposition par Hitex (www.hitex.co.uk).

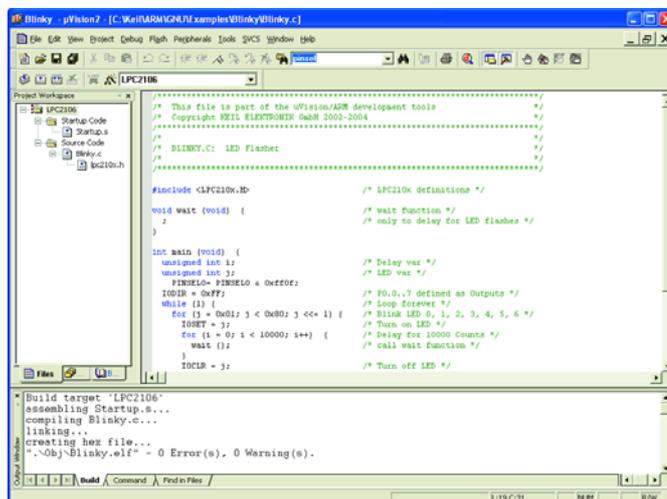


Figure 1. L'environnement de développement.

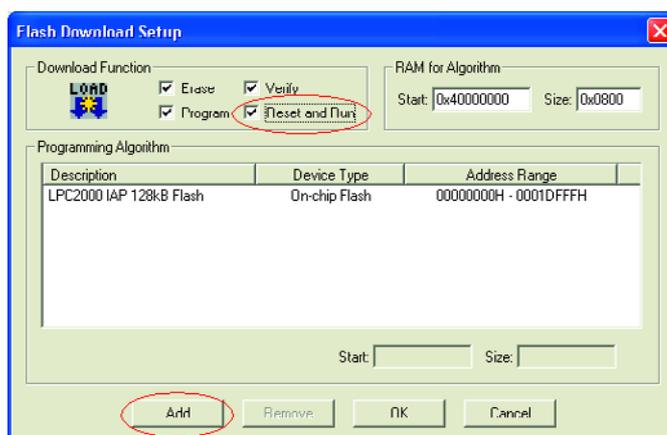


Figure 2. N'oubliez pas de bien configurer le Flash-tool.

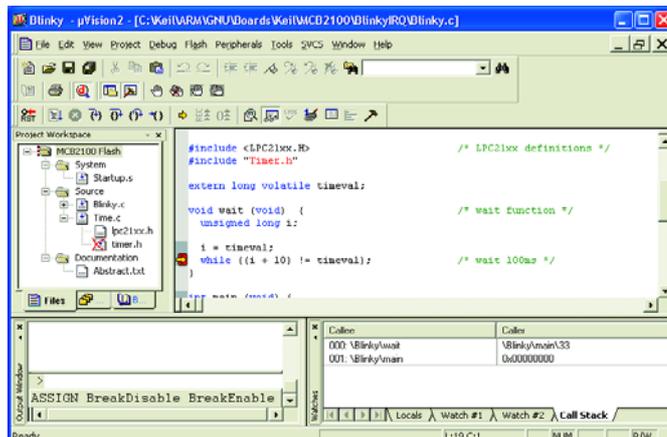


Figure 3. Exemple dans le débogueur.

TRICK.e DU PROF OSSMANN



Martin Oßmann est professeur à la FH d'Aix-la-Chapelle et depuis des années l'un de nos auteurs les plus prolifiques. Par sa colonne il aimerait faire réfléchir, réaliser, simuler et spéculer et naître des questions intéressantes.

Gagner avec TRICK.e !

La meilleure réponse à la question de ce TRICK.e sera récompensée cette fois-ci par un

Kit Carte 89S8252 Flash

d'Elektor d'un prix de € 99.

Ce kit comporte une platine dotée de ses composants et testée, avec bloc d'alimentation, câble et logiciel sur CD-ROM (cf. www.elektor.fr).

L'évaluation des réponses sera effectuée sous l'égide du Professeur Oßmann en collaboration active avec les rédactions d'Elektor; tout recours légal contre cette opération est exclu. En cas de réponses identiques, le vainqueur sera tiré au sort.

Nous allons ce mois-ci nous intéresser à l'euro. L'introduction de l'euro a été discutée en long et en large. Si cette nouvelle monnaie a exercé une attraction quasi-magique sur certains pays, elle s'est sentie plutôt rejetée par d'autres.

On connaît ce genre de forces dans le monde du magnétisme. Et c'est là la question du jour (ou du mois devrions-nous dire). Comment un euro exposé à un champ magnétique se comporte-t-il ? Pour le savoir, nous suspendons un euro pouvant se mouvoir librement devant un électro-aimant (cf. les images). Nous envoyons ensuite un courant puissant (entre 8 et 10 A efficaces) à travers la bobine. NOUS commençons par l'envoi d'un courant alternatif à une fréquence de 50 Hz. Nous recommençons l'expérience à une fréquence de 20 kHz.

La question de ce mois-ci : L'euro est-il attiré ou repoussé ?

Information :

À l'extérieur, une pièce de 1 euro est constituée de nickel et de laiton, à l'intérieur est dotée de 3 couches :

Cuivre-nickel, nickel, cuivre-nickel (info : www.bundesbank.de).

On pourra utiliser une pièce de 1 ou de 2 euro(s). La bobine des photos est constituée d'un corps P36x22 (Epcos) doté de 50 spires de tresse HF 90 x 0,1 mm de cuivre émaillé. Le corps de la bobine repose sur un demi-noyau ETD 44/22/15 de ferrite. L'inductivité de la bobine est de 240 µH environ. On a besoin, pour faire circuler dans la bobine, à 20 kHz, un courant de valeur efficace de 8 A, de l'ordre de 240 volts, tension qui n'a rien d'innocent, sachant que l'on a une puissance réactive de 2 kW environ. **Ndlr : il y a danger de mort ! en cas d'entrée en contact avec des parties ou des conducteurs non isolés véhiculant cette tension ! Il ne s'agit pas ici d'une expérience à faire dans son laboratoire personnel !**

Il est possible de répondre à la question sans avoir à faire de gros calculs pour peu que l'on réfléchisse bien. Il faut, si l'on veut vérifier par la pratique, avoir un minimum d'expérience dans le domaine des alimentations de puissance et disposer d'un laboratoire équipé en conséquence.



Un euro face à l'électro-aimant.
La pièce suspendue peut bouger librement.

Pour les lecteurs intéressés :

Il est facile de générer un courant 50 Hz à l'aide d'un transformateur torique sur lequel on bobine un enroulement secondaire de caractéristiques adéquates ne comportant que peu de spires. On peut essayer d'autres techniques d'embobinage, mais il faut veiller au facteur Q correct à 20 kHz pour disposer du courant requis. Il est possible d'obtenir l'intensité de courant élevée de la manière suivante : on utilise un pont complet à base de FETMOS de qualité ($R_{DS(on)}$ inférieur à 0,0...Ω), alimentés sous une tension continue entre 12 et 20 V et au courant correspondant. Le pilotage se fait à l'aide d'un signal carré de 20 kHz. La bobine est connectée au pont au travers d'un condensateur pris en série. Le dimensionnement du condensateur est tel qu'il constitue, en combinaison avec l'inductivité de la bobine, un réseau série résonnant syntonisé à 20 kHz environ. Il doit s'agir d'un très bon condensateur MKP prévu pour 600 V au minimum, capable de supporter 20 A sous 200 V (on en trouve dans les alimentations à découpage). La fréquence du signal carré est ensuite ajustée à la résonance du circuit et si tout est OK, on doit avoir les 8 A à travers la bobine à cette fréquence...



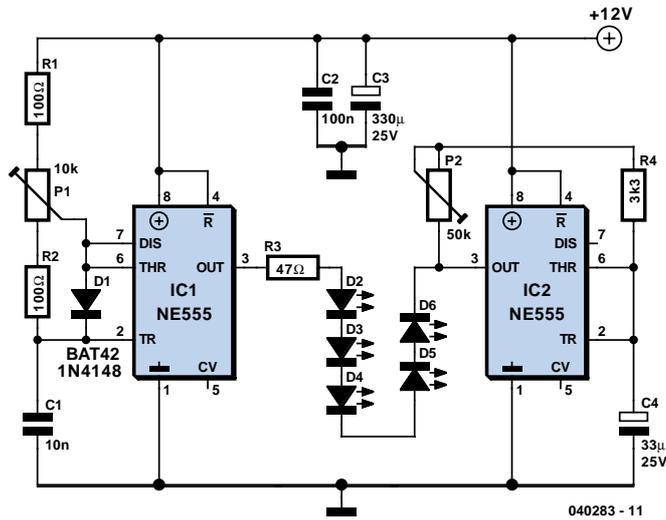
Règles de participation :

Envoyez votre réponse à la question Trick.e par Courriel, Télécopie ou Courrier à

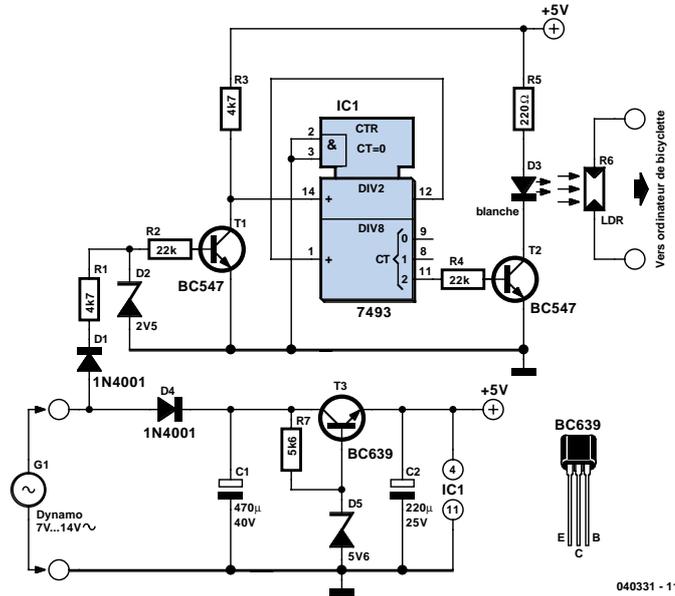
Rédaction Elektor, SEGMENT B.V./ELEKTOR
chez WWS, 38, rue la Condamine,
75017 Paris,
Télécopie 01.42.61.18.78,
Courriel : redaction@elektor.fr

**Date limite de réception des envois :
le 30 avril 2005.**

Tout recours légal est exclu. Cette opération est ouverte à tous nos lecteurs, exception faite des personnels de SEGMENT B.V., société dont fait partie Elektor.



040283 - 11



040331 - 11

Clignotant à gradateur

Myo Min

La plupart des éclairages de secours fixés ici et là utilisent des ampoules de verre pour leur lumière. Le filament de ces ampoules est fragile, ne vit que peu de temps et requiert une tension relativement élevée. Pour contourner ces inconvénients, l'auteur a conçu ce petit montage.

Dans le cadre d'un éclairage de secours, les LED à très haute intensité présentent des avantages évidents par rapport aux LED classiques et aussi aux ampoules de verre bien évidemment. IC1, épaulé par R1, R2, P1 et C1, fournit le signal modulé en largeur d'impulsion (MLI, PWM pour *Pulse Width Modulation* en anglais) qui attaque les U-LED à fréquence constante. P1 permet de jouer sur la luminosité. Il n'est nécessaire de prévoir d'éta- tage de commande additionnel, le NE555 étant en mesure de fournir jusqu'à 200 mA de courant. Il suffit, s'il vous une luminosité plus importante, d'ajouter des chaînes de LED additionnelles (R3, D2 à D6); on pourra prendre jusqu'à 8 chaînes en parallèle sur la chaîne du schéma.

IC2, constituée, avec P2, R4 et C4, un autre multivibrateur astable qui draine le courant de la chaîne de LED au rythme de la fréquence de clignotement. L'ajustable P2 sert à régler cette fréquence. C2 et C3 sont des condensateurs de découplage garantissant un fonctionnement stable aux fréquences faibles et élevées.

L'alimentation du circuit se fera à l'aide d'un adaptateur secteur 12 V continus à sortie régulée voire via un circuit convertisseur-rehausseur à base de MAX761 par exemple, l'énergie étant fournie par quelques piles AA. La valeur de la résistance de limitation R3 est fonction du type de LED bleues ou blanches et du nombre de LED prises dans la chaîne. En ce qui concerne IC1, la fréquence MLI répond à la formule suivante :

$$f = (1,44 / (PC1C1)).$$

Pour éviter des scintillements, la fréquence doit être supérieure à 5 kHz.

Au niveau de IC2, l'équation de conception est la même (on utilisera les valeurs de P2 et C2 cette fois). On ajustera la fréquence de clignotement entre 0,5 et 5 Hz.

(040238-1)

Compte-tours pour motocyclette

Peter van der Maarel

Nombre de motocyclettes (d'un certain âge) ne possèdent pas de compte-tours, accessoire intéressant s'il en est. Vu que la réalisation d'un système d'affichage est souvent le problème majeur lorsque l'on veut fabriquer un compte-tours pour motocyclette de son cru, l'auteur a trouvé une solution pratique en choisissant d'utiliser un « ordinateur » (bon marché) pour vélo. Cet accessoire est facile à monter sur le guidon et son écran est bien lisible.

La source du signal permettant de connaître le régime de rotation du moteur de la motocyclette est ici la dynamo. Celle-ci se trouve directement sur le vilebrequin et fournit une tension alternative au réseau de bord dont la fréquence correspond à la vitesse de rotation du moteur. Cette fréquence est trop élevée cependant pour être traitée telle quelle par l'ordinateur de bord. La solution consiste à commencer par la diviser par 16 à l'aide d'un compteur binaire 4 bits du type 7493 avant de l'appliquer à

l'ordinateur de bord.

On commence par un redressement et un écrêtage à 2,5 V, via D1, R1 et D2, du signal fourni par la dynamo, T1 convertissant le signal résultant en un signal logique. Le compteur IC1 comporte 4 bascules bistables (*flip-flop*) montées en cascade, de sorte que le signal subit une division par 16. Ce signal attaque, par le biais de T2, la LED blanche D3. La LDR R6 réagit à l'allumage de la LED et c'est elle qui attaque l'ordinateur et non pas le capteur de roue fourni d'origine.

Le signal de la dynamo fournit également la tension d'alimentation du circuit. La paire D4/C1 assure le redressement et le lissage, T3 et D4 régulant ensuite à 5 V la tension obtenue.

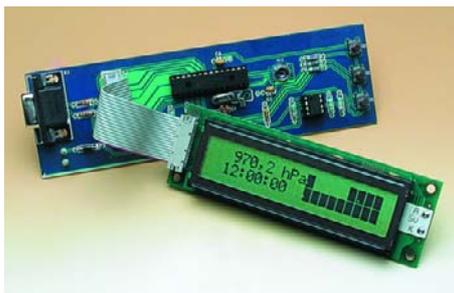
Pour un affichage correct (le vrai régime) le paramétrage de l'ordinateur doit être fait sur un développement de roue de 889 mm (ou 89 cm).

Il faudra veiller, lors de la réalisation du montage, à le mettre à l'abri des vibrations et de l'humidité. La LED et la LDR seront montées l'une en face de l'autre et être bien mise à l'abri de toute lumière ambiante incidente.

(040331-1)

Baro-altimètre de précision

Basé sur le capteur MS5534 de la société suisse INTERSEMA (spécialisée dans la mesure de pressions), ce montage vous fournira l'altitude au mètre près, mais aussi bien sûr la pression absolue, la température, et même l'heure. Le microcontrôleur utilisé ne vous est sans doute pas inconnu, puisqu'il s'agit d'un 16F876. Il est ici assez proche de ses limites, car la mesure de la pression, et surtout de l'altitude, représente un bon nombre d'opérations dont certaines en virgule flottante...



Enregistreur de températures

Pour l'extérieur et l'intérieur

Notre enregistreur (*logger*) de température est un appareil permettant la saisie continue de 2 températures en vue d'une analyse ultérieure sur PC. De par la présence d'un bus I²C, rien n'interdit d'intégrer d'autres types de capteurs.



Transparence - Émulateur 27C512 évolutif

Il nous a semblé intéressant, en partant d'un projet concret, celui d'un émulateur pour EPROM 27C512, utilisant un EPM7064 d'Altera et un AT90S8515 d'At-

mel d'entrer un peu plus dans le vif du sujet de manière à illustrer le déroulement du processus amenant à réaliser un tel montage.

Et quoi d'autre encore

Nous vous en laissons la surprise... mais nous ne pouvons pas nous empêcher d'attirer votre attention sur notre site qui vient de subir une cure de jouvence ; Forum, Petites Annonces, e-CHOPPE, autant de nouveautés qui méritent le détour, la part la plus importante du site restant cependant le magazine.



Il est possible de faire démarrer un abonnement à tout moment. Nous vous rappellerons en temps utile l'approche de la fin de votre abonnement. La méthode la plus rapide et la moins chère de vous abonner est de le faire par le biais de notre site Internet www.elektor.fr/abo/abo.htm, mais vous pouvez également le faire à l'aide du bon de commande se trouvant en fin de magazine. Il est possible de commander d'anciens numéros dans la limite de leur disponibilité (cf. le Bon de commande, leur prix est celui d'un numéro à l'unité).

Veuillez SVP nous fournir un changement d'adresse au moins 3 semaines auparavant en mentionnant votre numéro d'abonné (cf. le label accompagnant votre magazine), l'ancienne et la nouvelle adresse.

Le département Clients est accessible les jours ouvrables de 10h00 à 12h30 et de 13h30 à 16h30.

Si vous avez des questions concernant votre abonnement, vous pouvez appeler ce département au numéro 01.42.61.18.75

Pour le traitement de votre abonnement, WWS vous demande des données personnelles. Conformément à la loi « Informatique et Liberté », vous bénéficiez d'un droit d'accès à ces données et vous pouvez en demander la rectification. Sauf refus écrit de votre part auprès du service Abonnement, ces informations pourront être utilisées par des tiers.

Prix au numéro

France	5,75 €
DOM Surface	6,80 €
DOM Avion	8,50 €
Belgique	6,35 €
Suisse	10,90 FS
Canada	8.10 \$Can

Abonnement d'un an standard

France	59 €
Belgique	65 €
Suisse	110 FS
DOM Surface	77,50 €
DOM Avion	101 €

Étudiant

France	47,20 €
Belgique	52 €

Abonnement de 2 ans standard

France	106,20 €
Belgique	117 €
Suisse	198 FS
DOM Surface	139,50 €
DOM Avion	181,80 €

Étudiant

France	84,96 €
Belgique	93,60 €

Abonnements : WWS

E-mail : ventes@elektor.fr

Commandes/Ventes

E-mail : ventes@elektor.fr

Abonnement PLUS d'un an

France	68,50 €
Belgique	74,50 €
Suisse	130 FS
DOM Surface	87 €
DOM Avion	110,50 €

Étudiant

France	56,70 €
Belgique	61,50 €

Abonnement PLUS de 2 ans

France	125,20 €
Belgique	136 €
Suisse	238 FS
DOM Surface	158,50 €
DOM Avion	200,80 €

Étudiant

France	103,96 €
Belgique	112,60 €