INNOVATIONS... MONTAGES FIABLES... ÉTUDES DÉTAILLÉES... ASSISTANCE LECTEUR

ELECTRONIQUE

ET LOISIRS magazine

http://www.electronique-magazine.com

n°86 SEPTEMBRE 2006

L'ELECTRONIQUE POUR TOUS ELECTRONIQUE n°86

Un répéteur HF pour télécommande

SOMMAIRE DÉTAILLÉ PAGE 3



Un convertisseur DMX-ETHERNET

Un serveur Web GPRS

Une interface Client FTP avec PIC et SD-Card

Une alimentation double symétrique professionnelle

Une nouvelle platine d'expérimentation pour PIC

etc...

Imprimé en France / Printed in France

M 04662 - 86 - F: 5,00 €

France 5,00 € - DOM 5,00 € - CE 5,00 € - Suisse 7,00 FS - MARD 50 DH - Canada 7,50 $C

N° 86 - SEPTEMBRE 2006

elc
la qualité au sommet

«Toujours moins d'échauffement et plus de légèreté avec les nouvelles alimentations»

- Tension continue et alternatives simultanées
- Générateur de courant
- Sorties protégées

- Ventilation contrôlée
- Véritable troisième voie
- Série ou parallèle avec lecture directe

AL 936N — Transfo torique



Voies principales
2 x 0 à 30V / 2 x 0 à 3A séparé
ou 1 x ±0 à 30V / 0 à 3A tracking
ou 1 x 0 à 30V / 0 à 6A parallèle
ou 1 x 0 à 60V / 0 à 3A série

Sortie auxilliaire
2 à 5,5V / 3A
5,5V à 15V / 1A
lecture U ou I 592,02 €

NOUVEAU

ALR3002M



0 à 5, 6, 12 ou 30V
0-25mA, 250mA ou 2,5A =
6 ou 12 ou 24V 5A ~ 227,24 €

NOUVEAU

ALR3003 — Transfo torique



0 à 30V / 0 à 3A 161,46 €

ALR3003D — Transfo torique



2 x 0 à 30V / 2 x 0 à 3A séparé
ou 1 x ±0 à 30V / 0 à 3A tracking
ou 1 x 0 à 60V / 0 à 3A série
ou 1 x 0 à 30V / 0 à 6A *parallèle

(*mise en parallèle extérieure possible par l'utilisateur)
478,40 €

- Trois voies simultanées
- Mémorisation des réglages
- Logiciel fourni

AL 991S — Interface RS 232



±0 à 15V / 1A ou 0 à 30V / 1A
2 à 5,5V / 3A
-15 à +15V / 200mA 238,00 €

AL 843A



6V ou 12V / 10A = et ~
ou 24V / 5A = et ~ 238,00 €

AL 924A



0 à 30V / 0 à 10A 416,21 €

AL925



6 ou 12V / 5A = et ~ 133,95 €

AL901A



1 à 15V / 4A à 15V et 1A à 1V 106,44 €

ALF1205M



6V et 12V / 5A 155,48 €

AL923A



1,5V à 30V / 5A à 30V et
1,5A à 1,5V 157,87 €

AL 781NX



0 à 30V / 0 à 5A 321,72 €

ALF1201M



6V et 12V / 1A 83,72 €

AL890NX



+ et - 15V / 500mA
51,43 €

AL841B



3V 4,5V 6V 7,5V 9V
12V / 1A 43,06 €

Prix TTC

elc
59, avenue des Romains - 74000 Annecy
Tel +33 (0)4 50 57 30 46 Fax +33 (0)4 50 57 45 19
http:// www.elc.fr courriel commercial@elc.fr

En vente chez votre fournisseur de composants électroniques ou les spécialistes en Instrumentation.

Je souhaite recevoir une documentation sur :

Nom

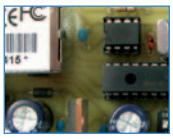
Adresse

Ville Code postal

FRANÇOISE BAUDOUX CRÉATION GRAPHIQUE 25 / 11 / 2005

86 SOMMAIRE 86

Un convertisseur DMX-ETHERNET 5



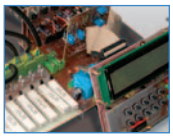
Nous mettons à profit les LAN désormais présents dans les théâtres et les discothèques pour transférer les flux de données standard DMX512 de la régie aux portiques où sont fixés les projecteurs et autres appareils de puissance. Le circuit peut fonctionner comme convertisseur DMX512/Ethernet et comme décodeur Ethernet/DMX512.

Un serveur Web GPRS 16



Appelé par un téléphone mobile, ce serveur se connecte au réseau GPRS pour nous permettre de consulter les données lues localement par deux sondes et accède à Internet par un navigateur. L'adresse IP à spécifier dans la barre des adresses pour accéder à la page web du dispositif est l'IP publique que le modem a obtenu durant la connexion et qu'il nous a communiquée par SMS.

Une alimentation double symétrique professionnelle 29
Troisième partie: la fin de la réalisation pratique des platines modulaires



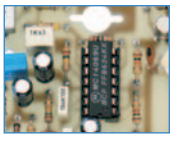
Alimentation professionnelle de laboratoire, ETALI est entièrement gérée par microcontrôleur ; elle fournit deux tensions continues stabilisées, symétriques par rapport à la masse et réglables de +1/0/–1 V à +36/0/–36 V. C'est l'outil idéal pour faire fonctionner des circuits à alimenter sous tension simple ou double symétrique ; elle peut fournir un courant de 3 A par branche. Les valeurs sont réglées par poussoirs et visualisées sur afficheur LCD. Dans cette troisième et dernière partie, nous terminons le montage du dernier module et mettons la dernière main à l'ensemble.

Une interface Client FTP* avec PIC et SD-Card 37
Première partie: analyse théorique et réalisation



Nous utilisons un microcontrôleur Microchip pour publier via FTP* des données sur Internet. Pour la première fois nous adoptons une interface réseau entièrement construite par nous à partir d'une des plus populaires puces Ethernet : la RTL8019 de Realtek. Avec un programme résident spécifique, ce circuit peut également servir de serveur Web.

Un répéteur HF pour télécommande 51



Ce répéteur VHF pour télécommande vous permet d'augmenter considérablement la portée de votre télécommande TV ou Hi-Fi, etc. et même de commander un appareil situé dans une autre pièce. Vous pourrez ainsi atteindre tous les dispositifs dotés de commande infra-rouge comme le décodeur, le lecteur de DVD, le magnétoscope, la chaîne AV...

Une nouvelle platine d'expérimentation pour PIC 60



Cette nouvelle platine d'expérimentation est un outil (ludique !) indispensable pour quiconque souhaite se perfectionner dans la mise en œuvre des microcontrôleurs PIC. Elle permet d'utiliser un clavier numérique (comme on en trouve sur les téléphones) et un afficheur LCD (tout aussi numérique) –composant désormais incontournable au labo d'électronique pour effectuer toutes les mesures et tous les contrôles voulus.

À la découverte du BUS CAN 67
Quatrième partie:



Conçu comme protocole de communication série pour faire communiquer entre eux tous les systèmes électroniques présents à bord d'une voiture, le bus CAN gagne aussi du terrain dans les domaines de l'automatisation industrielle (robotique) et de la domotique. Dans cette série d'articles, ou de Leçons (comme vous voudrez), nous allons aborder la théorie de son fonctionnement et nous prendrons de nombreux exemples dans le domaine domotique (c'est-à-dire des automatismes dédiés à la maison). Dans cette quatrième partie, nous analyserons comment un module peut acquérir des données et les rendre disponibles sur le bus.

L'index des annonceurs 76

Les Petites Annonces 76

Le bon d'abonnement 77

Ce numéro a été envoyé à nos abonnés le 24 août 2006

Crédits Photos : Corel, Futura, Nuova, JMJ

ABONNEZ-VOUS À

ELECTRONIQUE ET LOISIRS magazine

LE MENSUEL DE L'ÉLECTRONIQUE POUR TOUS

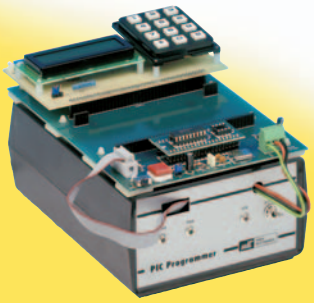
Notre nouveau site - www.electronique-magazine.com - est en ligne

Articles, Revues et CD téléchargeables au format PDF

Abonnements et anciens numéros papier en ligne

LES KITS DU MOIS... LES KITS DU MOIS

UNE NOUVELLE PLATINE D'EXPÉRIMENTATION POUR PIC



Cette nouvelle platine d'expérimentation est un outil (ludique!) indispensable pour quiconque souhaite se perfectionner dans la mise en œuvre des microcontrôleurs PIC. Elle permet d'utiliser un clavier numérique (comme on en trouve sur les téléphones) et un afficheur LCD (tout aussi numérique) –composant désormais incontournable au labo d'électronique pour effectuer toutes les mesures et tous les contrôles voulus. Le kit est livré avec un CD-ROM contenant **MPLAB IDE** version 7.20, **IC-Prog** version 1.05D, le nouveau système intégré **Proton DS Lite** version 1.036 et des programmes de test pour la platine EN1585.

EN1585..... Kit platine clavier + afficheur.....42,00 €

UN RÉPÉTEUR HF POUR TÉLÉCOMMANDE



Ce répéteur VHF pour télécommande vous permet d'augmenter considérablement la portée de votre télécommande TV ou Hi-Fi, etc. et même de commander un appareil situé dans une autre pièce. Vous pourrez ainsi atteindre tous les dispositifs dotés de commande infra-rouge comme le décodeur, le lecteur de DVD, le magnétoscope, la chaîne AV...L'émetteur comme le récepteur sont alimentés par pile de 9 Volts.

EN1628..... Kit émetteur avec coffret .....16,80 €
EN1629..... Kit récepteur avec coffret .....22,40 €

UN GÉNÉRATEUR DE FIGURES DE LISSAJOUS



Quand le physicien français Jules Antoine LISSAJOUS (1822-1880) fabrique un appareil mécanique, constitué de deux diapasons et de deux miroirs, grâce auquel il réussit à rendre visible la composition géométrique de deux mouvements harmoniques de fréquences identiques ou différentes, il ne pensait certainement pas que son nom serait indissolublement lié à un instrument de mesure, n'existant pas alors, que nous connaissons aujourd'hui sous le nom d'oscilloscope. Ce kit vous permet de générer

EN1612..... Kit complet avec coffret .....39,00 €

UN ÉMETTEUR TV AUDIO ET VIDÉO UHF



Les étonnantes prestations fournies par le modulateur audio et vidéo en CMS utilisé par le Générateur de mire EN1630 (voir les numéros 82 et 83 d'ELM) nous ont encouragés à concevoir et à réaliser un émetteur TV (audio et vidéo). Cet émetteur fonctionne dans la gamme de fréquences UHF mais ne nécessite ni self ni condensateur ajustable d'accord.

CARACTÉRISTIQUES TECHNIQUES : Tension d'alimentation 6 V - Courant consommé : 180 à 190 mA - Emission en UHF : canaux CH21 à CH69 - Puissance de sortie : 70 mW environ - Portée moyenne : 50 mètres - Signal entrée vidéo : 500 mVpp

KM1635..... Kit complet monté et testé ..... 125,00 €

UN CONVERTISSEUR DMX-ETHERNET



Nous mettons à profit les LAN désormais présents dans les théâtres et les discothèques pour transférer les flux de données standard DMX512 de la régie aux portiques où sont fixés les projecteurs et autres appareils de puissance. Le circuit peut fonctionner comme convertisseur DMX512/Ethernet et comme décodeur Ethernet/DMX512. Alimentation 12 Vdc. Dim : 10 cm x 5 cm. Le kit est livré avec le module Tibbo EM200, et2 prises DMX.

ET600..Kit convertisseur DMX-ETHERNET.....128,00 €

UN SERVEUR WEB GPRS



Appelé par un téléphone mobile, ce serveur se connecte au réseau GPRS pour nous permettre de consulter les données lues localement par deux sondes et accède à Internet par un navigateur. L'adresse IP à spécifier dans la barre des adresses pour accéder à la page web du dispositif est l'IP publique que le modem a obtenu durant la connexion et qu'il nous a communiquée par SMS. Le dessin du circuit imprimé ainsi que le programme du microcontrôleur sont téléchargeables sur **www.electronique-magazine.com** Le module GR47 est vendu séparément.

GR47 ..... Module GSM GR47 seul.....230,00 €

L'AUDIO-METRE OU LABO BF INTÉGRÉ



Tout amateur éclairé qui se lance dans la réalisation d'un montage BF s'aperçoit tout de suite que, pour effectuer les mesures requises, il devrait disposer d'une nombreuse instrumentation très coûteuse...qu'il n'a pas, bien sûr, puisqu'il n'est pas un professionnel ! Pour sortir de cette impasse, nous vous proposons de construire un instrument de mesure simple mais universel, dédié aux basses fréquences (BF), donc à l'audio et contenant, dans un seul et unique boîtier : un générateur BF, un fréquencemètre numérique et un voltmètre électronique mesurant les tensions, même en dB. Alimentation 230 Vac.

EN1600K ..... Kit complet avec boîtier .....210,00 €

AMPLI. STÉRÉO 2 X 55 W HYBRIDE LAMPES/MOSFET



Notre amplificateur stéréo Hi-Fi utilise en entrée deux tubes montés en cascode et comme étage final deux MOSFET de puissance capables de produire 2 x 55 WRMS, ce qui fait tout de même 2 x 110 W musicaux.

CARACTÉRISTIQUES TECHNIQUES : - Tension pour les lampes V1-V2: 340V - Tension pour les MOSFET finaux: 2 x 35 V - Courant de repos : 100 à 120 mA par canal - Courant à la puissance maximale: 1,5 A par canal - Amplitude maximale du signal d'entrée: 2 Vpp - Puissance maximale sur 8 ohms: 55 WRMS par canal - Distorsion harmonique maximale: 0,08%- Réponse en fréquence: 8 Hz à 40 kHz

EN1615..... Kit avec tubes et MOSFET mais sans le coffret ..... 269,00 €
MO1615..... Coffret percé et sérigraphié..... 48,00 €

Photos non contractuelles. Publicité valable pour le mois de parution. Prix exprimés en euro toutes taxes comprises. Sauf erreurs typographiques ou omissions.

PASSEZ VOS COMMANDES DIRECTEMENT SUR NOTRE SITE : www.comelec.fr

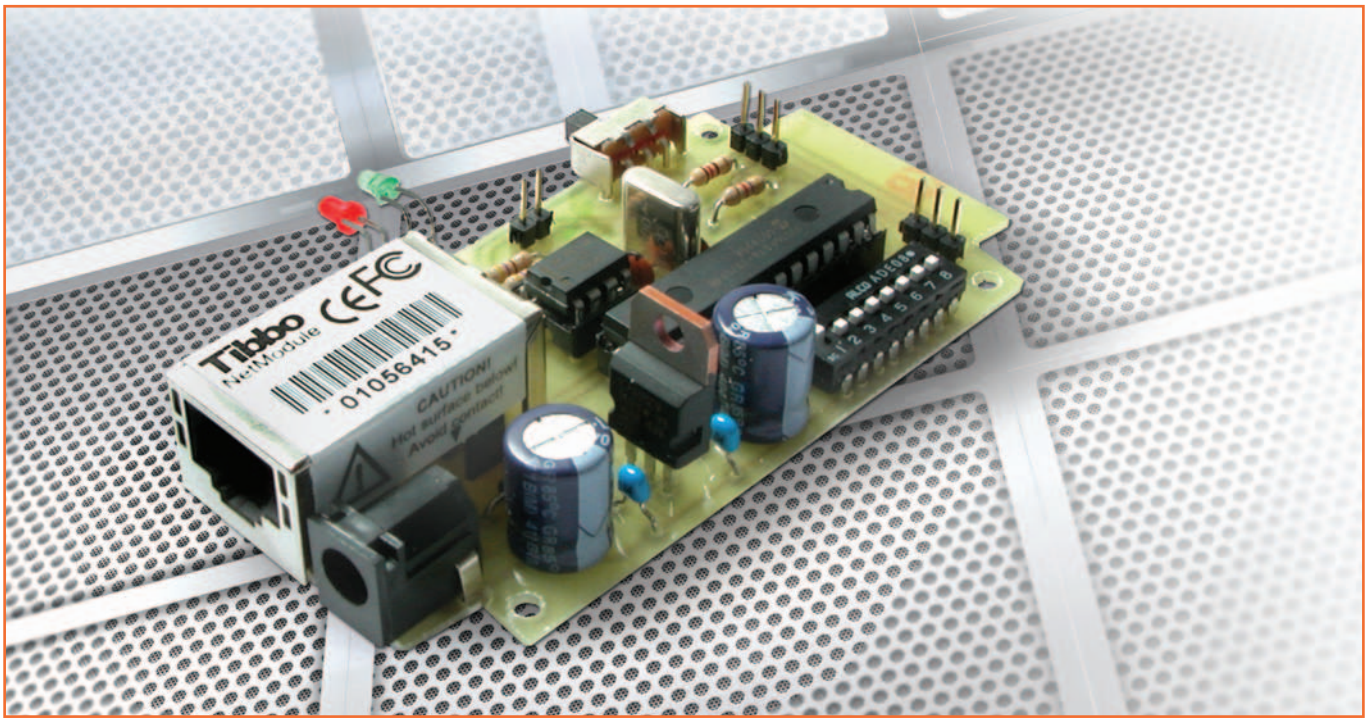
COMELEC CD 908 - 13720 BELCODENE www.comelec.fr
Tél.: 04 42 70 63 90 Fax: 04 42 70 63 95

DEMANDEZ NOTRE CATALOGUE 96 PAGES ILLUSTRÉES AVEC LES CARACTÉRISTIQUES DE TOUS LES KITS

Epéditions dans toute la France.Moins de 5 Kg : port 8,40 €. Règlement à la commande par chèque, mandat ou CB. Bons administratifs acceptés.
De nombreux kits sont disponibles, envoyez nous votre adresse et cinq timbres, nous vous ferons parvenir notre catalogue général de 96 pages.

Un convertisseur DMX-ETHERNET

Nous mettons à profit les LAN désormais présents dans les théâtres et les discothèques pour transférer les flux de données standard DMX512 de la régie aux portiques où sont fixés les projecteurs et autres appareils de puissance. Le circuit peut fonctionner comme convertisseur DMX512/Ethernet et comme décodeur Ethernet/DMX512.

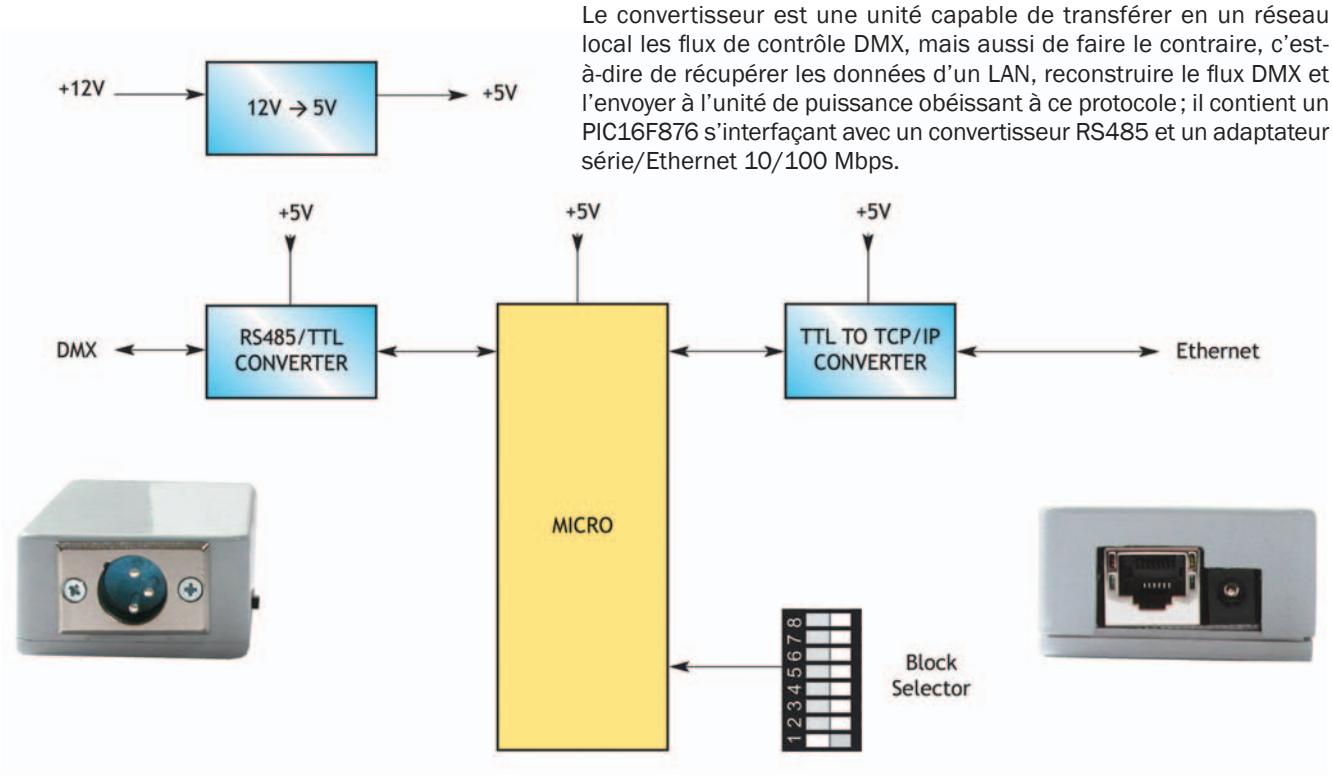


Les installations d'éclairage de spectacles (théâtre, concert, etc.) sont depuis des années gérées au moyen d'appareils (variateur, mélangeur, console ou centrale de commande) répondant au standard DMX512 : cette convention a permis d'uniformiser les câblages et les périphériques puisque, tous fonctionnant de la même manière, on peut étendre à volonté une installation existante en étant certain de l'entière compatibilité des éléments ajoutés. Le DMX512 prévoit que, afin d'éviter les pertes le long des interminables câbles d'alimentation des divers groupes d'éclairage et de simplifier le câblage des portiques, la console envoie à chaque projecteur, au format numérique (sur 8 bits), les informations du moment et de la quantité de son illumination. Localement, un décodeur convertit l'information numérique en une tension continue commandant le variateur de chaque luminaire (lequel reçoit son alimentation directement du secteur). Voir figure 6.

Notre réalisation

Ce n'est pas la première fois que nous abordons avec vous une réalisation DMX, mais dans cet article nous vous proposerons un convertisseur original permettant d'utiliser un LAN pour transférer les données du protocole DMX512. Il s'agit d'une interface permettant à la console d'atteindre les unités de puissance (variateurs situés près des projecteurs) à l'aide d'un convertisseur qui transforme les données DMX512 produites par la régie en signaux Ethernet et les envoie le long du réseau 10/100 Mbps et d'un autre (identique, en effet notre circuit est bidirectionnel) qui fait l'inverse, c'est-à-dire qu'il intercepte les données circulant sur le LAN et les transforme en un flux DMX512 propre à être décodé et à actionner le variateur. Cette idée, apparemment rarement applicable dans des situations pratiques, n'en a pas moins de réelles possibilités d'application : ce

Figure 1: Organigramme du système.



Le convertisseur est une unité capable de transférer en un réseau local les flux de contrôle DMX, mais aussi de faire le contraire, c'est-à-dire de récupérer les données d'un LAN, reconstruire le flux DMX et l'envoyer à l'unité de puissance obéissant à ce protocole ; il contient un PIC16F876 s'interfaçant avec un convertisseur RS485 et un adaptateur série/Ethernet 10/100 Mbps.

n'est d'ailleurs pas pour rien que dans le commerce on vous propose de tels convertisseurs en rack 19"...et à un prix en rapport avec ce boîtier professionnel ! Voir figures 1 et 2.

A l'origine de ce concept il y a le fait que les salles de spectacles modernes sont désormais équipées d'un réseau local ; alors, quand on utilise ce réseau existant pour véhiculer des données DMX512, on économise le traditionnel câblage RS485 en le limitant à la liaison entre la console et un ou plusieurs convertisseurs et entre les convertisseurs et les unités à variateurs.

Avec notre appareil, on peut même envisager d'éclairer, par exemple, tout un édifice ou un vaste local, etc., au moyen d'un contrôleur DMX512, en faisant passer les signaux à destination des luminaires du local ou des divers étages de l'édifice par le réseau Ethernet.

Songez aux centaines de mètres de câble "hard use" (apte à supporter le piétinement, l'usure et l'écrasement) économisées : il suffit alors d'alimenter les unités de puissance à la prise secteur la plus proche et de les commander par des câbles ne véhiculant que des courants faibles, ceux déjà installés du réseau LAN.

Le schéma électrique

Le circuit, comme le montre le schéma électrique de la figure 3, est fort simple, du moins en apparence : il y a en effet derrière une extrême complexité de fonctionnement. Cet appareil est réversible : il peut faire office de convertisseur RS485 (DMX)/Ethernet ou l'inverse. L'interface avec le bus sur lequel transitent les données DMX512 est U3, un MAX485 contenant un "transceiver" (émetteur-récepteur) RS485. Il s'agit en fait d'un émetteur et d'un récepteur gérés par les niveaux logiques des lignes RE/DE : la première est active au zéro logique et la seconde au niveau logique haut ; bien qu'elles puissent être contrôlées indépendamment l'une de l'autre (et pour des applications particulières il faut le faire : par exemple pour bloquer le TX ou le RX), pour nous ici le microcontrôleur les gère toutes deux ensemble. Quand la broche 12 de ce dernier est au niveau logique haut, il inhibe le récepteur de U3 et, à partir de sa broche 17, il envoie à l'entrée de l'émetteur (broche 4) les données à transmettre. Inversement, quand il ne doit pas émettre il est prêt à recevoir : il active le récepteur avec le même niveau logique bas qui a servi à inhiber l'émetteur ; ainsi, les données converties par la boucle de courant RS485 en TTL atteignent la broche 17 du micro.

Ici, un convertisseur fonctionne toujours et dans une seule direction : s'il est relié à la console et doit envoyer sur le LAN les données de l'unité à commander, il reçoit les signaux RS485 et les envoie au PIC ; s'il est en revanche utilisé pour commander un variateur de puissance en partant des données DMX512 prélevées par Ethernet, le micro envoie les données à U3 pour qu'il les transmette. Comme vous le voyez, la gestion simplifiée des broches RE/DE est au point et le micro, bien secondé, n'a pas à vérifier les données sur le bus RS485 pour décider du moment de la transmission.

C'est l'état logique de la ligne RA0 qui décide du mode de fonctionnement du convertisseur : le micro lit cette ligne après initialisation des E / S ; ou plutôt le paramétrage s'effectue grâce au cavalier J1 (fermé, le circuit est un récepteur RS485 et émet sur le LAN ; ouvert, le circuit lit l'Ethernet et émet en DMX512-RS485). Le cavalier est paramétré toujours avant l'alimentation du circuit : les éventuelles modifications ultérieures seraient ignorées.

Nous avons dit que le circuit est plus complexe que le schéma électrique ne le laisse apparaître : en effet, il en va bien ainsi car toute la gestion du protocole Ethernet et donc la conversion de LAN IEEE 802.3 à niveau TTL acceptable

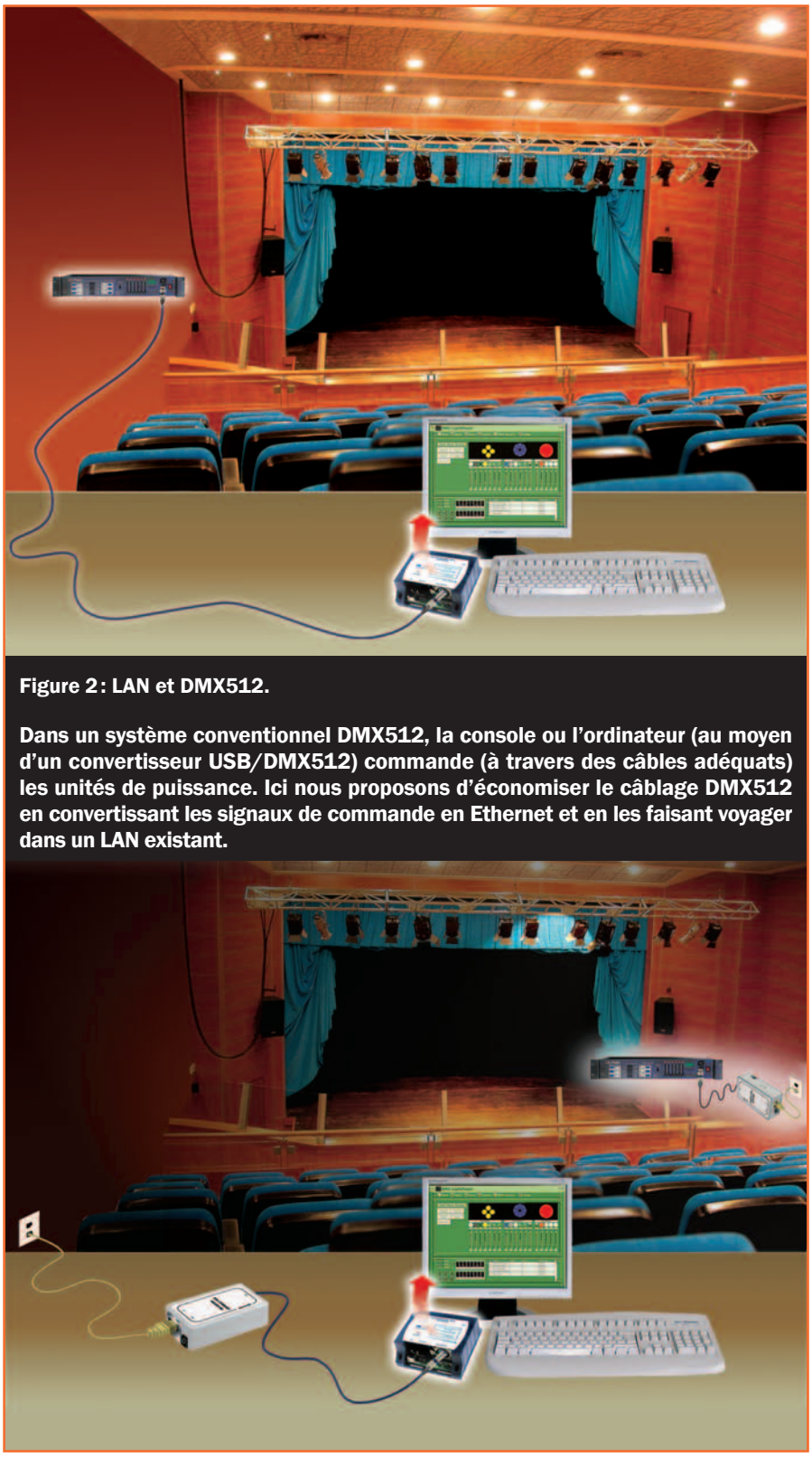


Figure 2 : LAN et DMX512.

Dans un système conventionnel DMX512, la console ou l'ordinateur (au moyen d'un convertisseur USB/DMX512) commande (à travers des câbles adéquats) les unités de puissance. Ici nous proposons d'économiser le câblage DMX512 en convertissant les signaux de commande en Ethernet et en les faisant voyager dans un LAN existant.

pour le PIC et de celui-ci à Ethernet est confiée au module U1 : il s'agit d'une interface LAN bidirectionnelle conçue par Tibbo et contenant un CPU plus un "buffer" en mesure d'élaborer les données afin d'interfacer un réseau Ethernet standard (selon le protocole TCP/IP) avec un dispositif série travaillant avec des signaux compatibles TTL.

C'est en fait une carte réseau comme il y en a dans les ordinateurs ; la différence est que les cartes des PC disposent, en plus du convertisseur (semblable au notre) d'un autre convertisseur qui transfère les données sur le bus ISA, EISA ou PCI.

Notre EM202 est donc une demi carte réseau. Si nous le relions au microcontrôleur, l'affaire est dans le sac : le convertisseur RS485/TTL d'un côté, le convertisseur Ethernet de l'autre et le PIC au milieu.

Pour approfondir un peu, rappelons que le protocole DMX512 peut gérer, à partir d'une console, 512 canaux en paramétrant pour chacun, avec un retard de quelques millièmes de seconde, les conditions d'illumination : chaque fois qu'un changement est voulu, le mélangeur émet un flux contenant jusqu'à 512 octets dont chacun décrit l'état que doit prendre chaque canal (aboutissant à chaque éclairage). Mais cela implique l'utilisation d'organes de communication très rapides ! En effet, mettre à jour 512 canaux en quelques fractions de seconde (et cela est nécessaire pour réaliser les effets de lumières souhaités en temps réel) requiert des délais de propagation très réduits ; ce n'est pas pour rien que le DMX512 (mettant à jour l'état de 512 canaux 40 fois par seconde) est caractérisé par une vitesse de communication de 250 kbps.

Et c'est là qu'apparaît un premier problème : le microcontrôleur utilise deux UART, un matériel et un logiciel ; le premier correspond aux lignes TX et RX et est utilisé pour dialoguer avec la ligne RS485. Cela parce qu'il est le plus rapide (plus de 250 kbps) et qu'il est en mesure de régir la vitesse de communication des dispositifs DMX512 (250 kpbs). Le second UART est émulé par un programme logiciel ("routine") qui initialise les lignes RC2 et RC3 respectivement en émission et réception des données ; il ne permet qu'une vitesse de communication de 38,4 kbps bien que le module Ethernet ait une interface série TTL à 115 kbps (car conçue pour travailler avec les UART "rapides" des ports COM des ordinateurs). Pour contourner cet obstacle, nous avons fait en sorte que l'élaboration d'une partie seulement du flux DMX512 soit confiée au convertisseur ; bref, en conversion RS485/Ethernet le micro, bien qu'il acquière en temps réel la totalité des flux DMX512, n'extrait les données que de 64 canaux seulement. Nous obtenons ainsi une division par huit de la vitesse de communication et 32 kbps suffisent.

Pour permettre au circuit de profiter de toutes les possibilités du protocole et de travailler, 64 à la fois, avec les 512 canaux possibles, nous avons prévu de définir à l'aide d'un dip-switch le groupe géré : si vous jetez un coup d'œil au schéma électrique de la figure 3, vous voyez que DS1, dont les huit micro-interrupteurs sont lus par le port RB du microcontrôleur, a des résistances de tirage internes

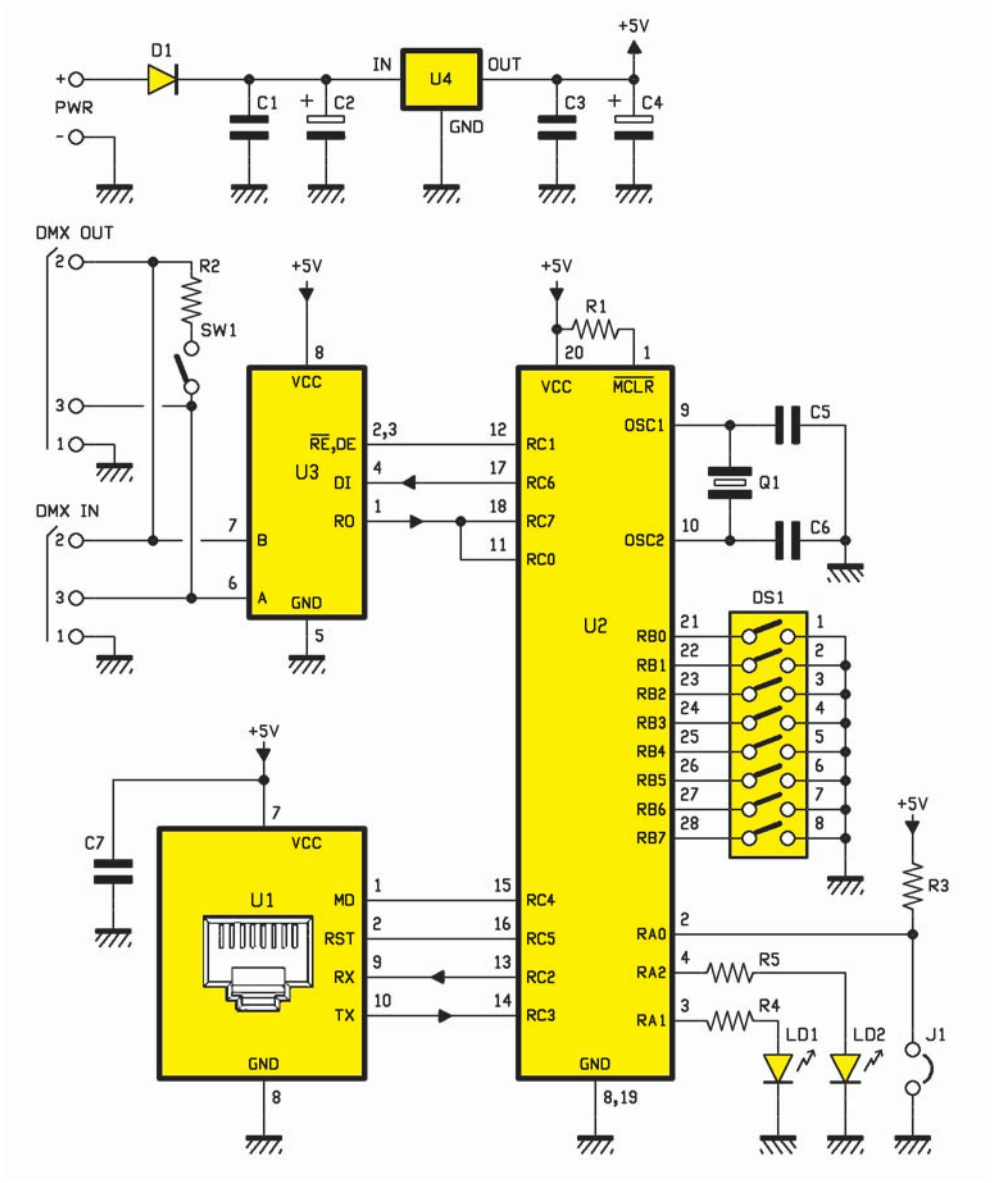


Figure 3 : Schéma électrique du convertisseur DMX-ETHERNET.

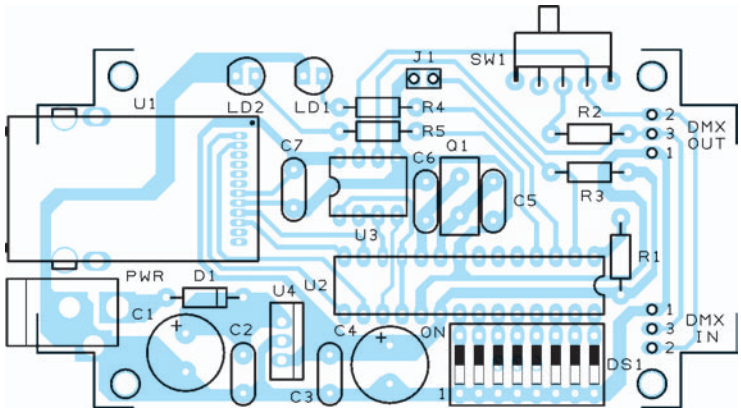
(d'ailleurs vous ne les voyez pas, justement parce qu'elles sont internes !) ; eh bien, le réglage de ce dip-switch dit au PIC16F876 quel groupe de 64 canaux il doit prendre en compte dans le flux standard DMX512 intéressant notre bus RS485.

Par exemple, si on ferme mint1 (broche 21 de U2 au zéro logique) le circuit s'occupe du groupe 1 à 64 ; en mode convertisseur RS485/Ethernet cela implique que, des données DMX512 reçues depuis la console, notre appareil ne prend en examen que les canaux du premier au soixante-quatrième ; en mode LAN/DMX512, le micro ne s'occupe et n'envoie les informations que des canaux 1 à 64. Toujours en termes de gestion des données côté RS485, précisons que le programme résidant dans le PIC sait reconnaître le commencement de chaque flux grâce à l'en-tête ("header") d'identification : chaque série de données au standard DMX commence par un "break" d'au moins 88 µs au niveau logique bas (tension négative sur le fil A par rapport au B et à la masse).

L'UART utilisée pour lire le canal des données arrivant du mélangeur ou de la console est configuré pour distinguer les différents octets : chacun est séparé du précédent par un bit de start et du suivant par deux bits de stop (le format des données en protocole DMX512 est du type 8, N, 2). A cause de son mode de fonctionnement, l'UART ne pourrait détecter correctement le commencement d'un flux sans, ensuite, perdre des transitions concernant les données des canaux.

C'est pourquoi le programme résident utilise le canal RX pour la seule réception des données, le TX pour l'envoi (quand le circuit fonctionne en mode convertisseur Ethernet/DMX) et qu'il initialise la ligne RC0 comme entrée utilisée pour détecter ledit "break". Donc, quand la broche 11 du micro passe au zéro logique et y reste pendant au moins 88 µs, le programme commence la lecture et le décodage des données, lesquelles sont ensuite répercutées le long du second canal série obtenu, cette fois, avec les E / S RC2, RC3, RC4, RC5. Cette dernière est une interface de niveau TTL servant au dialogue avec le convertisseur EM202, dont le canal des données veut, en plus des signaux TX et RX, deux signaux de contrôle nommés MD et RST ; le premier (en utilisation normale) est maintenu au niveau logique haut et le second est le "reset" (après chaque mise sous tension il est mis par le micro d'abord à l'état logique un, de façon à réinitialiser le module, puis au zéro logique). Dans la création des flux de données reçues par le bus

Figure 4a : Schéma d'implantation des composants du convertisseur DMX-ETHERNET.



Le module EM202 Tibbo est inséré directement dans les trous du circuit imprimé puis ses broches sont soudées avec un petit fer à panne fine et du tinol de 0,75 millimètre de diamètre (soudez aussi les languettes de fixation de son boîtier, à défaut de quoi lors d'une extraction de la fiche RJ45 vous risqueriez de l'arracher). Les connecteurs barrettes mâles servent à relier l'appareil (au moyen des connecteurs femelles au pas de 2,54 mm correspondants) aux connecteurs XLR, du moins si vous désirez que l'unité serve de pont vers d'autres. Si en revanche vous voulez un module à connecter exclusivement à un élément DMX, il vous suffit de disposer d'un seul connecteur XLR femelle.

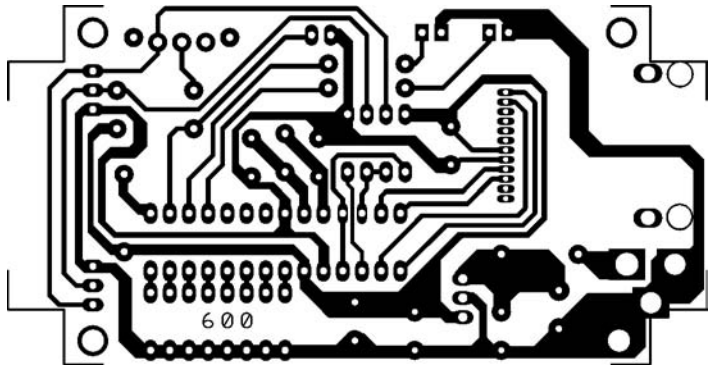


Figure 4b : Dessin, à l'échelle 1, du circuit imprimé du convertisseur DMX-ETHERNET.

Liste des composants

R1 ..... 4,7 k
R2 120
R3 ..... 4,7 k
R4 470
R5 470

C1..... 100 nF multicouche
C2..... 470 µF 25 V électrolytique
C3..... 100 nF multicouche
C4..... 470 µF 25 V électrolytique
C5..... 10 pF céramique
C6..... 10 pF céramique
C7..... 100 nF multicouche

D1 ..... 1N4007
LD1 ... LED 3 mm verte
LD2 ... LED 3 mm rouge

U1..... EM202
U2..... PIC16F876-EF600 déjà programmé en usine
U3..... MAX485
U4..... 7805

Q1 ..... quartz 20 MHz

DS1 ... dip-switch à 8 micro-interrupteurs
SW1 .. inverseur à glissière 90°

Divers :

1 prise d'alimentation
1 clip batterie 6F22 9 V
1 support 2 x 4
1 support 2 x 14
1 cavalier
2 connecteurs barrettes mâles à 3 broches
2 connecteurs XLR3 M/F
2 nappes à 3 fils 5 centimètres
1 boulon 3MA 8 millimètres

Sauf spécification contraire, toutes les résistances sont des 1/4 W à 5 %.

DMX et à transmettre sur le réseau, le PIC16F876 "économise" les données afin d'accélérer la communication en épargnant de la bande sur le réseau ; cela est obtenu en transmettant les informations strictement nécessaires. Cette optimisation est obtenue par un procédé que l'on peut comprendre en considérant que chaque unité répond aux instructions concernant le groupe des 64 canaux pour lequel elle est configurée par DS1. Le protocole DMX implique qu'il y ait dans les flux transmis les octets du nombre de canaux servant à parvenir à l'adresse du dernier : pratiquement, si l'on ne doit opérer que sur les canaux du 50 eme au 60 eme , le flux doit contenir de toute façon (même si tous sont au zéro logique) les octets du premier au 50 eme (en plus de ceux des 50 eme au 60 eme). Ces derniers avec des valeurs de 0 à 255 en fonction de l'intensité lumineuse voulue.

Un peu de syntaxe Ethernet

Puisqu'il doit répéter les données du DMX, le protocole Ethernet prévoit que le micro envoie à la broche RX de U1 un flux commençant par */n, où n est le numéro du plus haut canal à gérer, jusqu'à 64 donc.

Par exemple, si sur un groupe de 64 canaux on ne doit gérer que 10 unités, de la cinquième à la quinzième, le flux commence par */15 ; les données suivantes sont quinze octets, au zéro logique pour ce qui regarde les canaux non utilisés (du premier au cinquième) et complets pour les canaux à paramétrer (6 à 15). Puis le module Ethernet n'envoie que quinze octets en réseau et non tous les 64, car cela n'aurait pas de sens, occupant inutilement le LAN et ralentissant la communication. De même, pour alléger le travail du PIC et rendre la réponse aux commandes le plus rapide possible, quand le circuit fonctionne en mode Ethernet/RS485 on lit le numéro situé au début du flux afin de savoir quand cesser de déchiffrer les données.

En fait, quand il trouve */15 le micro de l'unité fonctionnant en récepteur LAN

n'attend pas la réception des 64 octets mais achève l'acquisition au bout des quinze, puis lance tout de suite après le "task" de conversion DMX512 en restituant, à travers l'UART interne, les données qui en découlent au RS485, à travers la ligne TX (RC6). Bien sûr, dans la conversion Ethernet/DMX, le micro lit aussi les micro-interrupteurs de DS1 et sait avec précision où sont insérées les données provenant du LAN.

Précisons à ce propos que, les décodeurs DMX identifiant les données après le "break", le convertisseur est obligé de reconstruire un flux contenant un nombre d'octets égal à celui du plus haut canal à retransmettre ; bref, si on travaille avec le groupe 65 à 128 et si l'on doit recevoir les données du canal 100, bien qu'il soit paramétré pour reconnaître 64 canaux, en transmission DMX le microcontrôleur doit engendrer au moins cent flux. Il est bien entendu que, si l'on doit agir sur le canal 100, après le "break" il produira 99 octets au zéro logique et un avec de vraies données.

L'artifice utilisé pour réduire à l'essentiel le flux des données dans le LAN permet de réaliser des applications particulières et de transférer les adresses des canaux, par exemple pour commander à partir d'un mélangeur ayant moins de 64 voies, une partie inutilisée d'une unité de puissance ayant justement ces mêmes canaux de libres.

Par exemple, si nous avons installé une unité variateur avec décodage dans laquelle, sur les 64 canaux disponibles, les 32 derniers sont gérés par une console paramétrée de l'adresse 481 à la 512 ; nous disposons également d'un mélangeur à 16 canaux (adresse 17 à 32) que nous voulons utiliser pour commander seize éclairages sans avoir à installer une seconde unité de puissance. Donc, le mélangeur appartient au premier groupe de 64 canaux et l'unité de puissance au dernier.

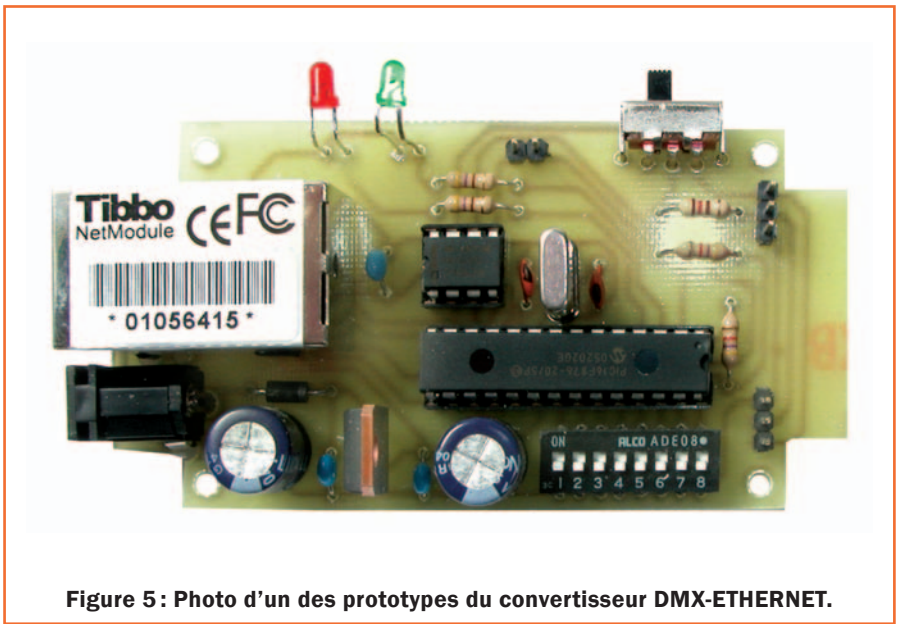
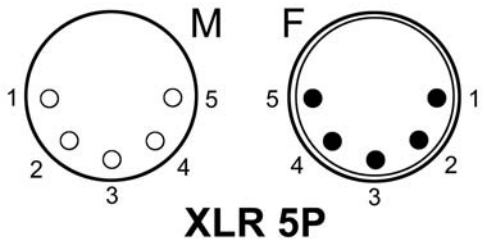


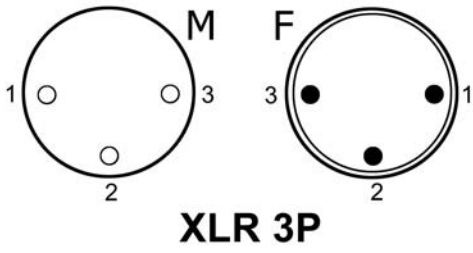
Figure 5 : Photo d'un des prototypes du convertisseur DMX-ETHERNET.





Figure 6 : Le standard DMX512.





C'est un protocole conçu pour faire fonctionner une régie de lumières : la console envoie des flux de données série contenant l'état des canaux gérés ; la particularité de ces flux est que le système donne en une seule séquence l'état d'un maximum de 512 canaux (d'où le nom de DMX512) reliés aux éléments montés en parallèle le long d'un bus RS-485. La console envoie les informations sur l'état de chaque éclairage, rotor ou autre accessoire (comme une machine fumigène) au format numérique ; localement le dispositif destinataire convertit le signal en analogique, c'est-à-dire en une tension continue de 0 à 10 V commandant le variateur de lumière associé au projecteur.

Le flux définissant la condition de chaque canal contient 8 bits avec lesquels on exprime une valeur binaire de 0 à 255 : zéro correspond à l'extinction et 255 à la pleine puissance lumineuse. Du moins si l'on commande des éclairages ou des éléments en mouvement, mais s'il s'agit de dispositifs à allumer ou éteindre, 0 correspond à OFF et 255 à ON. Le principal problème avec le DMX512 est la vitesse de communication du bus : à la limite, il doit envoyer 512 flux à une fréquence (40 fois par seconde) telle que l'exécution soit pratiquement immédiate ; sans quoi, avec des séquences lumineuses particulièrement rapides, des retards inadmissibles pourraient se produire. Chaque commande et son flux contiennent les données d'états de tous les canaux gérés ; les octets ne sont pas codés avec une adresse mais par position et chaque séquence DMX512 commence par un "break" (période au niveau logique bas pendant au moins 88 µs.) suivi d'un octet vide, d'un bit de stop puis, un après l'autre, les octets des différents canaux avec comme intervalle chaque fois un bit de start et un de stop. Les décodeurs des dispositifs DMX512 comptent les octets pour savoir quand arrive celui qui, en fonction de son paramétrage, contient l'état à prendre. L'envoi des flux est séquentiel et respecte l'ordre 1 à 512 : si on a par exemple 23 canaux à gérer, les octets sont transmis à partir du premier, puis le deuxième suit et ainsi de suite jusqu'au vingt-troisième. En outre le standard DMX512 prévoit que même pour faire varier la condition d'un canal on doit transmettre les flux de tous les canaux, toujours dans l'ordre arithmétique croissant : si, par exemple, on ne veut modifier à partir de la console que le canal 45, on doit tout de même envoyer un flux contenant les données des canaux du premier jusqu'au quarante-cinquième au moins ; il n'est en revanche pas obligatoire de reconstruire la totalité de la séquence de 1 à 512 !

La communication utilise un bus RS485 à ligne équilibrée (ou symétrique) dans lequel les données voyagent de manière symétrique (justement) le long de deux connecteurs chauds par rapport à la masse de référence : la condition logique 1 correspond à une impulsion positive sur le fil A et, en même temps, une négative de même amplitude sur le B (polarité référée à la masse). Inversement, le zéro est obtenu par une impulsion négative en A envoyée en même temps qu'une positive en B. Le récepteur possède un étage différentiel qui ne retient qu'une impulsion sur les deux, ce qui permet de supprimer les interférences lesquelles, on le suppose, se propagent uniformément sur les deux conducteurs chauds et s'annulent donc dans le différentiel. La RS485 est une sorte de boucle de courant différentiel caractérisée par une haute immunité aux perturbations, ce qui est idéal pour des liaisons sur des centaines de mètres dans un environnement saturé d'interférences comme celui où on a installé des éclairages à haute tension. La connexion se fait avec des câbles à deux fils plus tresse de blindage au bout desquels on a monté des XLR3 ou 5 (c'est le nombre de broches). Les appareils sont souvent équipés de deux connecteurs mâles, afin de pouvoir en monter plusieurs en parallèle et de constituer ainsi une chaîne (la sortie de l'un à l'entrée de l'autre et le dernier doit être doté d'un "terminator" ou bouchon constitué d'une résistance de 120 ohms entre les fils A et B).

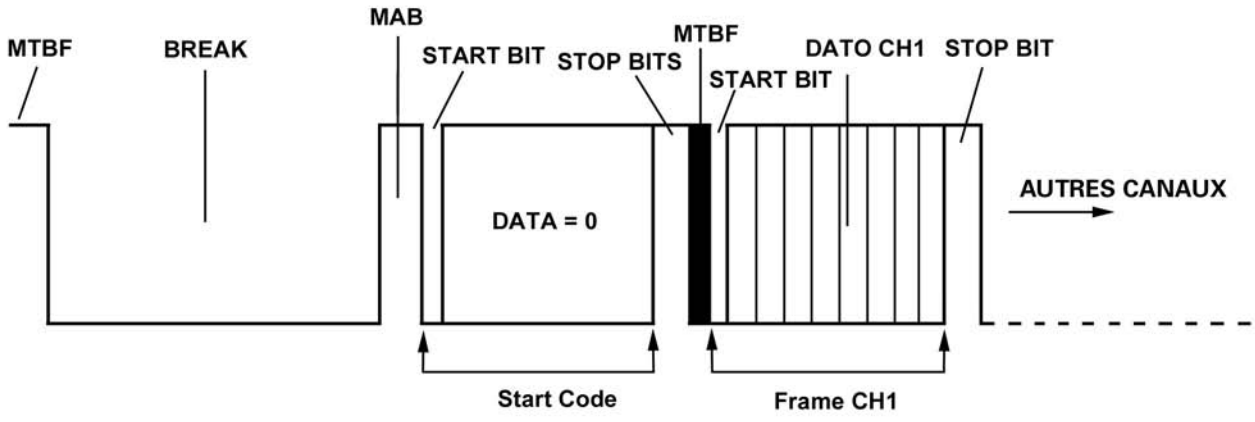


Figure 7 : L'adressage du convertisseur.

Pour dire au circuit quel groupe de données des 64 sur les 512 canaux possibles il doit lire dans le LAN ou lui envoyer, on utilise les huit micro-interrupteurs du dip-switch DS1 que l'on paramètre en fonction du faisceau désiré : pour savoir combien en mettre à ON, il suffit de diviser par 64 le numéro du canal le plus haut du groupe voulu (par exemple, pour le 321 à 384 il faut fermer 384 : 64 = 6 micro-interrupteurs, toujours à partir du 1). Le tableau ci-contre vous ôtera tout doute de l'esprit.

SELECTION DES CANAUX								
GROUPE	MINT1	MINT2	MINT3	MINT4	MINT5	MINT6	MINT7	MINT8
1 à 64	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF
65 à 128	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF
129 à 192	ON	ON	ON	OFF	OFF	OFF	OFF	OFF
193 à 256	ON	ON	ON	ON	OFF	OFF	OFF	OFF
257 à 320	ON	ON	ON	ON	ON	OFF	OFF	OFF
321 à 384	ON	ON	ON	ON	ON	ON	OFF	OFF
385 à 448	ON	ON	ON	ON	ON	ON	ON	OFF
449 à 512	ON	ON	ON	ON	ON	ON	ON	ON

Le problème est résolu en "trompant" le second convertisseur, celui qui prend les données du LAN et les convertit en DMX512 ; cela se fait en paramétrant l'adresse du premier convertisseur sur le groupe 1 à 64 (seul le mint1 est fermé) de manière que dans la conversion DMX512/LAN l'unité reconnaisse les données du mélangeur à 16 canaux. L'autre convertisseur (Ethernet/DMX512) doit en revanche avoir la même adresse que l'unité de puissance, afin de pouvoir engendrer un flux contenant les 64 derniers canaux.

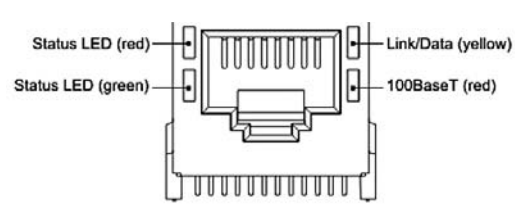
Dans la conversion des données en RS485, le microcontrôleur prend les informations arrivant du mélangeur DMX512 et les envoie au module Ethernet qui les reconstruit avec l'habituelle syntaxe */n en envoyant sur le réseau les octets des 32 premiers canaux : 16 au zéro logique et 16 avec les informations issues des curseurs du mélangeur. L'autre convertisseur prend les informations du LAN et en reconstruit le flux DMX512. Bien sûr, pour le convertisseur LAN/RS485 les données lues sur le LAN n'ont pas de position spécifique, car les octets arrivant sont toujours de toute façon au nombre maximum de 64, de celui du premier canal (utilisé ou non) à celui du dernier effectivement utilisé. Dans la conversion LAN/DMX512, c'est aux micro-interrupteurs de DS1 de dire au microcontrôleur comment localiser les octets reçus dans le flux DMX512 qu'il doit reconstruire et envoyer à l'unité de puissance ; donc, si avec DS1 on paramètre le groupe d'adresses des 64 derniers canaux (tous les micro-interrupteurs sur ON), le micro engendre un flux de 512 canaux lesquels, selon notre exemple, contiennent les données significatives des seules positions 481 à 512, soit des 16 dernières du groupe de 64 canaux dont elles font originairement partie. En d'autres termes, en définissant différemment les adresses des deux convertisseurs, les données synthétisées par le LAN/DMX512 se maintiennent à la même position à l'intérieur du groupe de 64 canaux, mais sont transférées dans le groupe défini par DS1.

Bon, eh bien puisque nous avons éclairci le fonctionnement du module, en récepteur DMX et en émetteur LAN comme en RX Ethernet et TX DMX, il ne nous reste qu'à expliquer quelques détails, comme l'intervention des LED de signalisation. LD1 est le voyant de fonctionnement et elle est allumée par le microcontrôleur après l'initialisation des E / S. LD2 s'allume au rythme des transitions sur le canal des données et

Figure 8 : Le module Tibbo EM202.

Le composant utilisé pour la réception et l'envoi des données à partir du et vers le LAN est un adaptateur Ethernet 10/100 Mbps doté d'une interface série TTL à 115,2 kbps ; le module n'est guère plus grand qu'un connecteur RJ45, mais il contient un CPU, une puce Ethernet et l'interface TTL. Aux côtés de la RJ45, il dispose de quatre LED : celles de droite sont Link/Data et 100 Mbits ; les autres peuvent être gérées par voie logicielle. Le port série interne peut travailler en "half-duplex" ou "full-duplex", à 7 ou 8 bits de données et accepte cinq modes de parité (aucune, paire, impaire, mark, space) ; il implémente en outre le contrôle de flux à travers RTS et CTS, ce qui en autorise la connexion aux ports RS232 de l'ordinateur. Côté Ethernet, le EM202 accepte les protocoles UDP, TCP-IP, ARP, ICMP (PING) et DHCP. Pour les liaisons avec l'extérieur, voici le brochage :

1 **Mode Selection ;** se fixe normalement au 1 logique ;
2 **Reset ;** actif au 1 logique, normalement se maintient à zéro ;
3 **DTR ;** ligne configurable comme E / S pour d'autres applications ;
4 **DSR ;** ligne configurable comme E / S pour d'autres applications ;
9 **RX ;** canal données en réception ;
10 **TX ;** canal données en émission ;
11 **CTS ;** sélectionne aussi le mode full (niveau logique haut) ou half-duplex (zero) ;
12 **RTS** en full-duplex ou Data Direction Control en half-duplex.



1 = TX +
2 = TX -
3 = RX +
4 = non utilisé
5 = non utilisé
6 = RX -
7 = non utilisé
8 = non utilisé

Disposition des LED de signalisation et brochage du connecteur RJ45 du module Tibbo EM202.

Figure 9 : Le programme résident.

```
DEFINE OSC 20

@               DEVICE HS_OSC

'Configuration usart 250000
DEFINE HSER_BITS 9
DEFINE HSER_RCSTA 208
DEFINE HSER_TXSTA 101
DEFINE HSER_BAUD 250000
DEFINE HSER_CLROERR 1 'Hser clear overflow automati-
cally



SYMBOL  RES      =PORTC.5
SYMBOL  MD       =PORTC.4
SYMBOL  LEDV     =PORTA.1
SYMBOL  LEDR     =PORTA.2
SYMBOL  JUMP     =PORTA.0
SYMBOL  TX       =PORTC.2
SYMBOL  RX       =PORTC.3
SYMBOL  DMX      =PORTC.1
SYMBOL  IN       =PORTC.0
SYMBOL  TXDMX    =PORTC.6

TMP             VAR     WORD
TMP1            VAR     WORD
VALORE          VAR     byte[65]
BREAK           VAR     WORD
NDMX            VAR     WORD
DATI            VAR     WORD

ADCON0=0
ADCON1=7
OPTION_REG.7=0

HIGH RES
HIGH MD
PAUSE 500
LOW RES


 CLEAR
LOW LEDR
High LEDV
For TMP=0 TO 5
        Toggle LEDV
        Pause 500
Next TMP
HIGH LEDV
NDMX=0

MAIN:

NDMX= 512-(64*(    1+
1*PORTB.1+1*PORTB.2+1*PORTB.3+1*PORTB.4+1*PORTB.5+1*PORTB.6+1*PORTB.7))

    IF JUMP=0 THEN
        LOW DMX
        LOW LEDR
        DATI=0
        PULSIN in,0,break

        IF BREAK>=40 THEN

            HSERIN 2000,MAIN,[tmp,tmp1]
            if tmp<>0 and tmp1<>0 then
                goto main
            endif

            FOR TMP=1 TO ndmx
                HSERIN 10,main,[tmp1]
            NEXT TMP

            FOR tmp=1 TO 64
                hserin 10,EXITTX,[VALORE[TMP]]
                DATI=DATI+1
            NEXT TMP

        EXITTX:
                    PAUSE 3
                    high ledR
                    SEROUT2 TX,6, ["*/",DATI]
                    FOR TMP=1 TO DATI
                        SEROUT2 TX,6,
[VALORE[tmp]]
                    NEXT TMP

            ENDIF

        ELSE
            HIGH DMX
            LOW LEDR

            SERIN2 RX,6,1000,MAIN,[WAIT ("*/"),DATI]

            FOR tmp=1 TO DATI
                SERIN2 RX,6,200,EXITRX,[VALORE[TMP]]
            NEXT TMP

            EXITRX:
                    HIGH LEDR
                    RCSTA.7=0
                    OUTPUT TXDMX
                    low TXDMX

                    PAUSEUS 128
                    RCSTA.7=1
                    PAUSEUS 14

                    FOR TMP=1 TO NDMX
                        HSEROUT [0]
                    NEXT TMP

                    FOR TMP=1 TO DATI
                        HSEROUT [VALORE[TMP]]
                    NEXT TMP
        ENDIF

    GOTO MAIN
```

Figure 10 : Directement depuis l'ordinateur.

Le convertisseur réversible décrit dans cet article peut être utilisé pour commander des unités de puissance DMX512 directement à partir d'un PC, à travers la carte réseau ou un "hub" ou un "switch" ; pour cela il suffit d'implémenter une "routine" (un programme) en Basic ou une "utility" (utilité) pour Microsoft Windows en Visual Basic ou Delphi incorporant la syntaxe des instructions utilisée pour faire voyager les données DMX512 en un LAN. Cette syntaxe prévoit que chaque flux converti par le circuit RS485/Ethernet commence par les caractères */n, où * et / sont fixes et n indique au convertisseur s'occupant de la transformation en DMX512 de combien d'octets sera composé le flux ; plus exactement, étant donné que le protocole DMX512 prévoit qu'un flux de commande contienne un nombre d'octets égal à celui du plus haut canal à gérer, n correspondra au numéro du dernier canal du mélangeur ou de la console de régie lumière. Après le "header" (en-tête) qu'on a décrit (*/n) le programme du PC doit envoyer, un après l'autre, les octets définissant l'état des différents canaux entre le premier et le dernier et non tous, les octets relatifs aux canaux non utilisés devant tous être à zéro ; par exemple, pour intervenir seulement sur le canal 15, les octets de 1 à 14 doivent avoir pour valeur zéro.

donne de ce fait un aperçu visuel de la communication.

Le bus du et vers les dispositifs DMX est un RS485 : les fils A et B acheminent des impulsions de courant en opposition de phase reconstruites, par le récepteur de U3, au moyen d'un étage différentiel qui extrait les niveaux logiques 1 et 0 TTL ; les impulsions sont référées à un troisième pôle, commun, représenté comme masse. Le zéro logique TTL correspond (en réception et en émission) au fil A positif et B négatif par rapport à la référence ; pour le un logique c'est le contraire.

L'interrupteur en série avec R2 sert, conformément à ce qu'impose le standard DMX512, à clore le bus (c'est un "terminator" ou bouchon) : le dernier dispositif doit avoir ce bouchon, constitué d'une résistance de 120 ohms montée entre les fils A et B.

Enfin, notez que notre appareil est déjà disposé à être inséré dans un bus : le connecteur à trois fils à relier à la prise socle de face avant est en parallèle avec son jumeau lequel sert à passer les signaux à une unité suivante montée en cascade.

Tout le fonctionnement du convertisseur est dû au programme résident du microcontrôleur U2 écrit en Basic et dont le "listing" complet est visible figure 8.

La réalisation pratique

La réalisation pratique de ce convertisseur DMX/Ethernet est des plus simples (la simplicité apparente paie !). Voir figure 5. La platine est constituée d'un petit circuit imprimé simple face, dont la figure 4b donne le dessin à l'échelle 1. Une fois le circuit imprimé réalisé ou procuré, commencez par insérer les deux supports de circuits intégrés 2 x 4 et 2 x 14, le dip-switch DS1 (attention, les chiffres de 1 à 8 vers le bas), le cavalier J1 et les deux barrettes mâles à 3 broches. Vérifiez attentivement vos soudures (ni court-circuit entre pistes ou pastilles ni soudure froide collée). Insérez et soudez ensuite tous les composants (comme le montrent les figures 4a et 5), en poursuivant par les résistances, condensateurs, diode, LED, quartz et régulateur (ces derniers montés debout) et en terminant par les "périphériques" : la prise d'alimentation, l'inverseur à glissière SW1 et enfin le module U1 Tibbo (pour celui-ci, utilisez un petit fer à panne fine et du tinol de 0,75 millimètre et n'oubliez pas de souder aussi les languettes de fixation). Attention à l'orientation des quelques composants polarisés : circuits intégrés (repère-détrompeurs

Figure 11 : Lancement du programme.

Avant d'utiliser le circuit, il faut configurer l'adresse IP du module EM202 : après en avoir connecté la prise RJ45 à la carte réseau du PC au moyen d'un câble croisé, lancez le programme et, dans la fenêtre de dialogue, écrivez l'adresse à laquelle vous voulez que le module réponde.

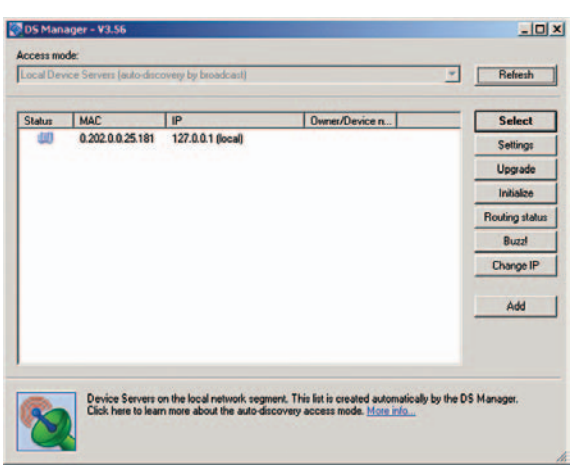


Figure 12 : Comment changer ou attribuer une IP.

Le programme de configuration reconnaît les dispositifs reliés au réseau et permet de définir les IP avec un clic sur le bouton Change IP. Le passage visible ici sert pour changer une IP existante et pour définir l'adresse d'un module quand il y en a deux sur le réseau.

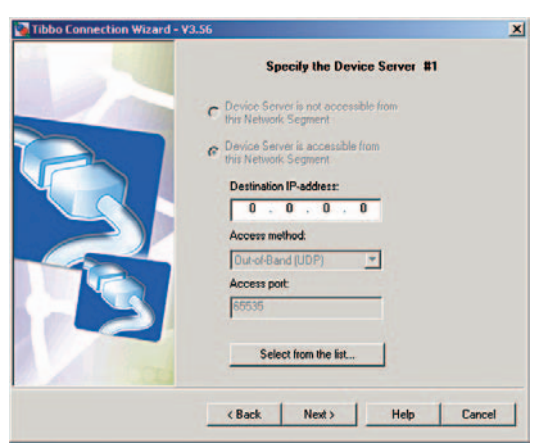
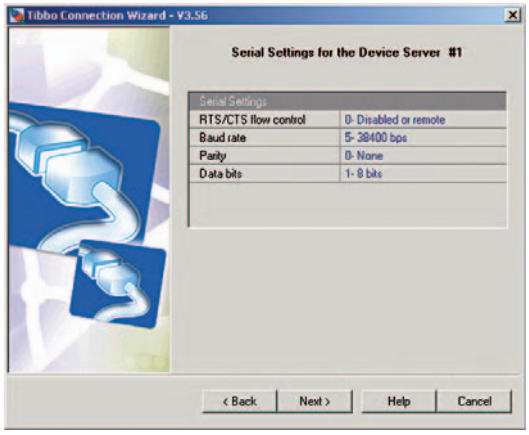


Figure 13 : Fenêtre de récapitulation des paramètres acceptés lors du “setup”.



A partir d’une fenêtre de dialogue du menu de configuration, il est possible de définir les caractéristiques de fonctionnement de l’interface série du module EM202, parmi lesquelles l’utilisation des signaux RTS et CTS. Vous voyez ici le récapitulatif des paramètres du premier module (ou le seul s’il n’y en a qu’un) ; il s’agit des valeurs prédéfinies proposées et acceptées lors des phases précédentes de “setup” (réglages de base ou configuration initiale).

en U bien orientés vers R1 et C7, mais insérez-les à la toute fin), diode (bague vers U4), LED (méplats vers la gauche), régulateur (semelle vers C3) et électrolytiques (+ vers l’intérieur du ci). Vérifiez bien toutes les polarités et (encore une fois) la qualité des soudures. Insérez les deux circuits intégrés en les orientant correctement.

Vous pouvez installer la platine dans un boîtier plastique de dimensions appropriées, comme le montrent les figures 1 et 2 : le grand côté sera percé pour le passage des LED, l’un des petits pour celui de la RJ45 et du jack d’alimentation et l’autre pour la XLR. Relier cette dernière à la platine par fils.

Le logiciel Tibbo

Quand le montage est terminé, connectez-vous au site de **www.tibbo.com** et, dans la section “download” (téléchargement), téléchargez le logiciel de gestion Device Server Toolkit (DST) for Windows si vous avez un PC doté du SE Microsoft ou Virtual Serial Port Driver (VSPDL) si vous tournez sous Linux. Le téléchargement est gratuit (il suffit de bien vouloir vous enregistrer).

Une fois lancé le fichier .exe, on accède au menu d’installation : dans les deux premières fenêtres de dialogue, acceptez le paramétrage proposé en cliquant sur Next ; dans la troisième fenêtre qui apparaît, vous devez définir l’adresse IP, indispensable pour que le module Tibbo fonctionne selon le protocole TCP/IP. Un clic sur le poussoir d’option Device Server is accessible from this Network Segment et, dans la case Destination IP address, écrivez l’adresse à donner au module. Cette adresse doit être différente de celle déjà attribuée à d’autres appareils opérant sur le réseau, sans quoi ni le module ni le périphérique ayant la même adresse que lui ne pourraient communiquer. Voir figure 11.

Par un clic sur le poussoir Select from the list on peut choisir un des EM202 trouvés sur le LAN : cela est très utile, par exemple, quand on configure deux modules présents sur le même réseau, auquel cas le cadre montre les adresses MAC et les éventuelles IP. Là, avec un clic sur le bouton Change IP on accède à une fenêtre de dialogue dans laquelle on peut écrire la nouvelle adresse, que l’on confirme par OK, puis par Add, ce qui nous ramène à la troisième fenêtre du menu. Voir figure 12.

Avec un clic sur Next on passe à la quatrième, où il est possible de configurer un éventuel second module sur le réseau (Device Server #2). Si vous voulez paramétrer un dispositif à la fois, cliquez sur Next le nombre de fois qu’il faut pour arriver à la fenêtre de dialogue récapitulative dans laquelle, avec un clic sur Finish, les paramètres deviennent opérationnels et l’on sort du menu et du programme Setup. Voir figure 13.

Quand on a configuré les unités à utiliser avec le système (une si l’on veut gérer l’interface directement à partir d’un ordinateur du réseau au moyen d’un logiciel spécifique ; deux au cas où on se sert du LAN pour véhiculer des données DMX à travers un mélangeur et un bloc de variateurs) il ne reste qu’à les connecter ; le câblage le plus simple est celui où deux unités sont interconnectées avec un **câble réseau croisé** mais, dans la majeure partie des cas, les convertisseurs utilisant le LAN comme moyen de communication, on reliera le connecteur RJ45 de chacune à une prise Ethernet, c’est-à-dire à un port du “hub” ou du “switch” interconnectant les périphériques du réseau.

L’unité servant de convertisseur RS485/LAN (côté mélangeur ou interface DMX512 depuis PC) doit avoir le cavalier J1 fermé et doit être reliée, en plus du réseau local, avec un câble standard au mélangeur ou à l’interface depuis l’ordinateur. L’autre (convertisseur Ethernet/DMX512), en plus du câblage réseau, doit avoir une connexion vers le bloc de variateurs, réalisée avec des câbles à deux fils plus tresse de blindage et connecteurs XLR.

Avant de mettre sous tension il faut, dans les unités convertisseur, paramétrer (en se servant des huit micro-interrupteurs de DS1) le groupe des 64 adresses ; pour connaître le groupe, il suffit de vérifier sur quelle adresse travaille la console (ou une partie : par exemple un groupe de curseurs) envoyant les commandes. Alimentez les unités avec des adaptateurs secteur 230 V (sortie basse tension 9 à 15 Vcc 300 mA au moins). A lui seul le module EM202 consomme 230 mA ! La LED verte doit s’allumer. Vérifiez que tout fonctionne alors comme indiqué ci-dessus. Pour cela, lancez une commande à partir du mélangeur (manuelle ou à partir d’un programme de régie lumière de l’ordinateur dédié) et contrôlez que les projecteurs répondent convenablement.

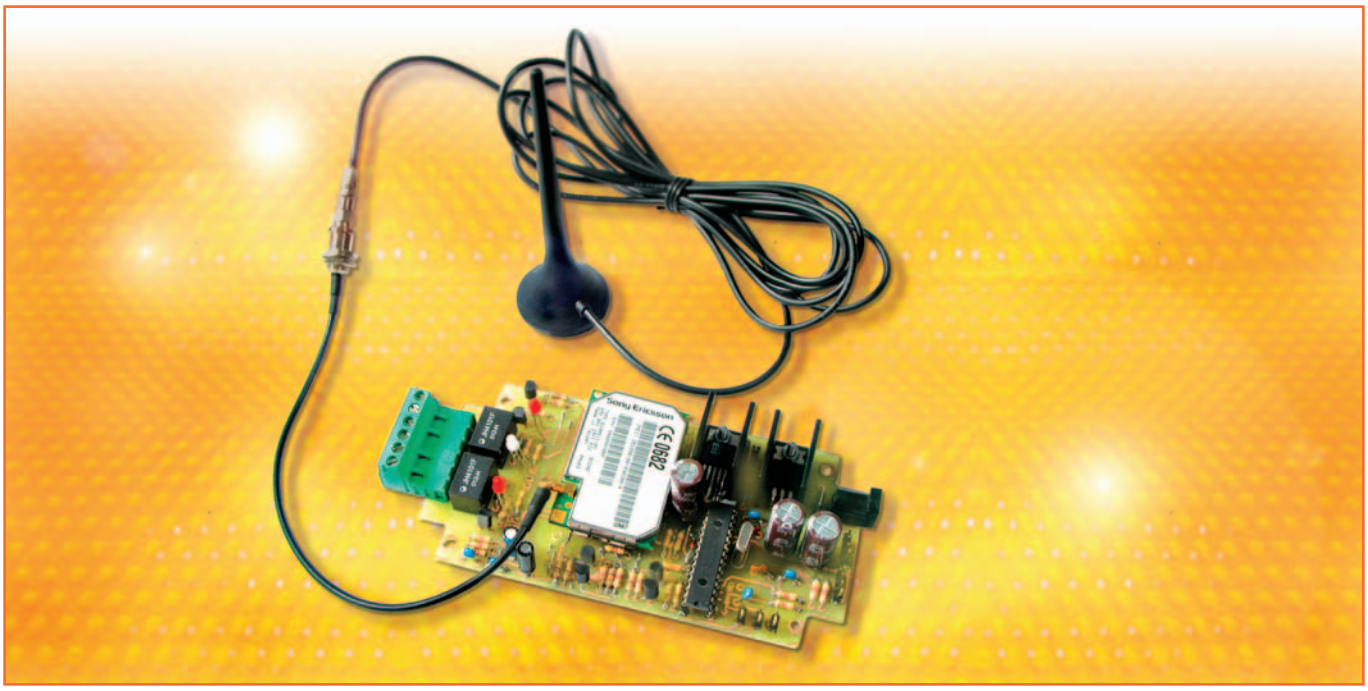
Comment construire ce montage ?

Tout le matériel nécessaire pour construire ce convertisseur DMX-ETHERNET ET600 est disponible chez certains de nos annonceurs. Voir les publicités dans la revue.

Les typons des circuits imprimés et les programmes disponibles sont téléchargeables à l’adresse suivante :
http://www.electronique-magazine.com/circuitrevue/086.zip. ◆

Un serveur Web GPRS

Appelé par un téléphone mobile, ce serveur se connecte au réseau GPRS pour nous permettre de consulter les données lues localement par deux sondes et accède à Internet par un navigateur. L'adresse IP à spécifier dans la barre des adresses pour accéder à la page web du dispositif est l'IP publique que le modem a obtenu durant la connexion et qu'il nous a communiquée par SMS.



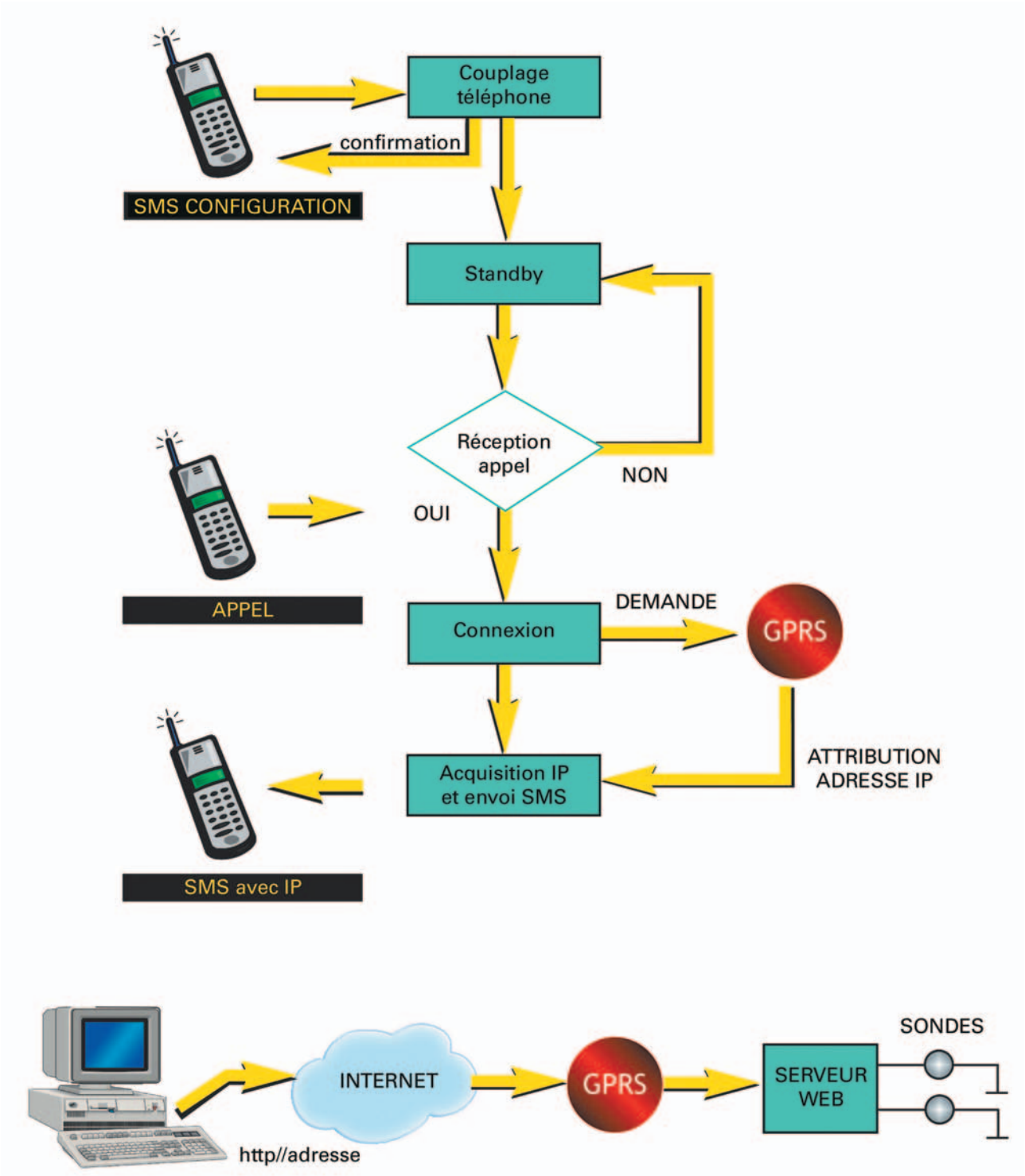
Internet, le grand réseau reliant des millions d'usagers à travers le vaste monde, étend chaque jour un peu plus sa Toile ("Web") ; son accès est de plus en plus facile, il devient donc populaire, grâce entre autres au développement de technologies permettant même à des personnes privées (et plus seulement à de riches sociétés) de se connecter et de travailler à des vitesses naguère inimaginables.

Non contents des résultats obtenus avec le réseau câblé, nous avons assisté ces dernières années à une montée en puissance fulgurante de nombreuses techniques "wireless" (sans fil) parmi lesquelles on trouve celles rendant le Web accessible à partir d'un téléphone mobile : après une période WAP (sur les lignes GSM) pas très convaincante, nous sommes parvenus au système GPRS, soit l'accès à Internet par mobile à la vitesse respectable de 171 kbps, sans appel à effectuer (un peu comme avec l'ADSL), puisqu'il suffit de spécifier les paramètres du serveur, la connexion à celui-ci étant instantanée à partir du moment où le téléphone la demande.

Notre réalisation

L'importance du système GPRS et notre exigence croissante de rendre accessible par le Web des données ayant une certaine utilité, nous a poussé à réaliser le serveur Web décrit dans cet article. Cet appareil, à la demande d'un usager habilité, se connecte au réseau GPRS et, une fois obtenue l'adresse IP publique que le service attribue à son modem, la communique de façon à rendre l'accès à Internet possible.

Figure 1: L'accès au serveur Web GPRS.



Pour accéder au serveur Web, il faut procéder selon trois phases : la première est le couplage au téléphone utilisé pour activer la connexion ; il faut envoyer un SMS au serveur Web avec le téléphone mobile (ce SMS contient le mot de passe, c'est-à-dire les cinq derniers chiffres de l'IMEI du module GPRS) et attendre un SMS de confirmation. A la suite de l'appel, le serveur Web se connecte au réseau GPRS et, quand l'IP est obtenue, il la communique au mobile avec un message. Il est alors possible d'accéder à la page Web depuis n'importe quel PC relié à Internet : il suffit pour cela de taper l'IP contenue dans le SMS dans la barre des adresses.

Dans cette première phase, nous vous proposons une application, disons didactique (notre vocation jamais abjurée), consistant à accéder par Internet, à l'aide d'un modem doté de son IP et d'un PC disposant d'un navigateur (peu importe lequel), à la page Web de notre système et à visualiser les températures (appelons-les interne et externe par convention) relevées par deux capteurs à semiconducteurs placées dans le même circuit.

Une fois la logique de fonctionnement apprise et les notions de base assimilées (vous trouverez une partie du logiciel de gestion du micro et du modem GSM/GPRS GR47 figures 5 et 6), vous disposerez des moyens vous permettant

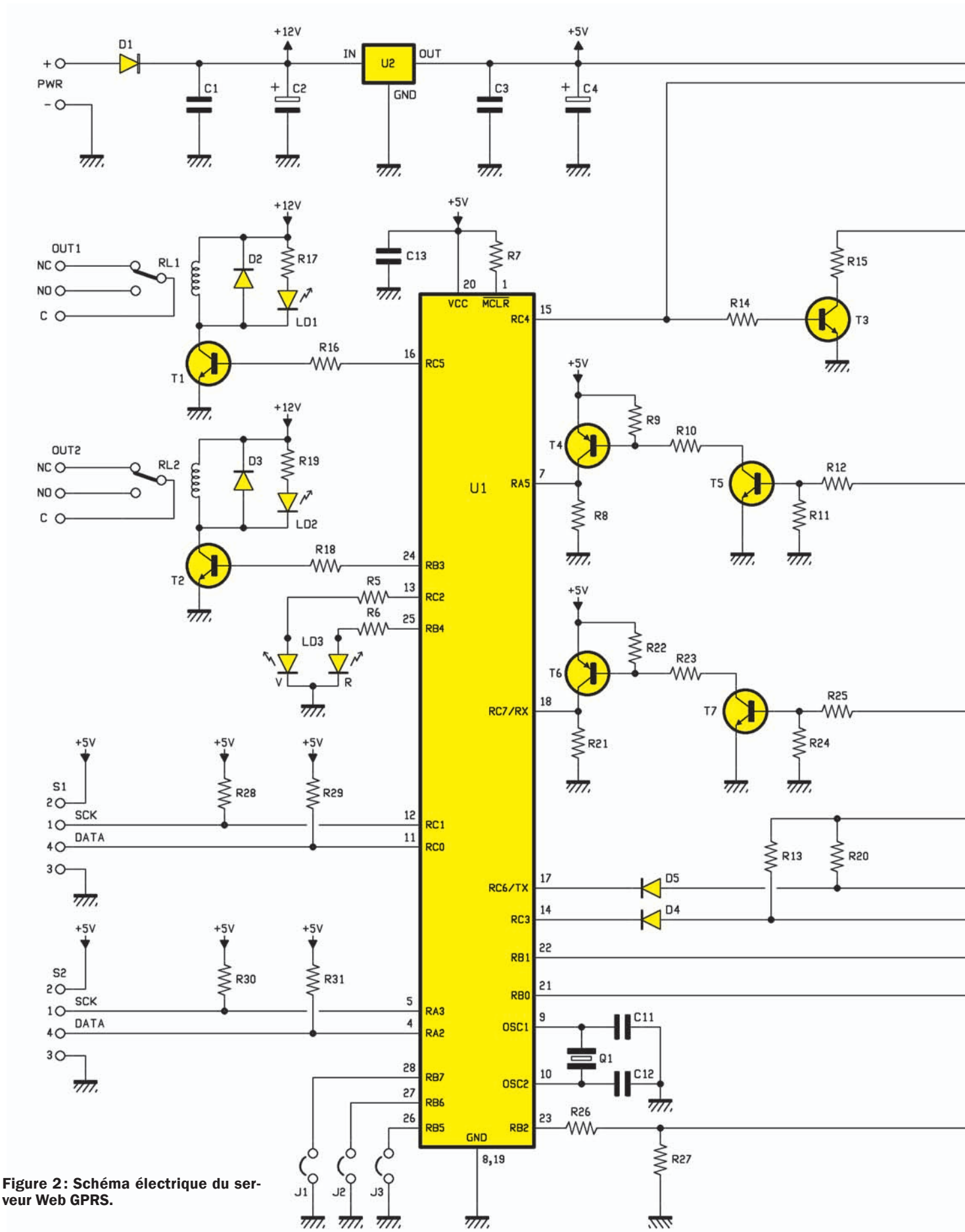
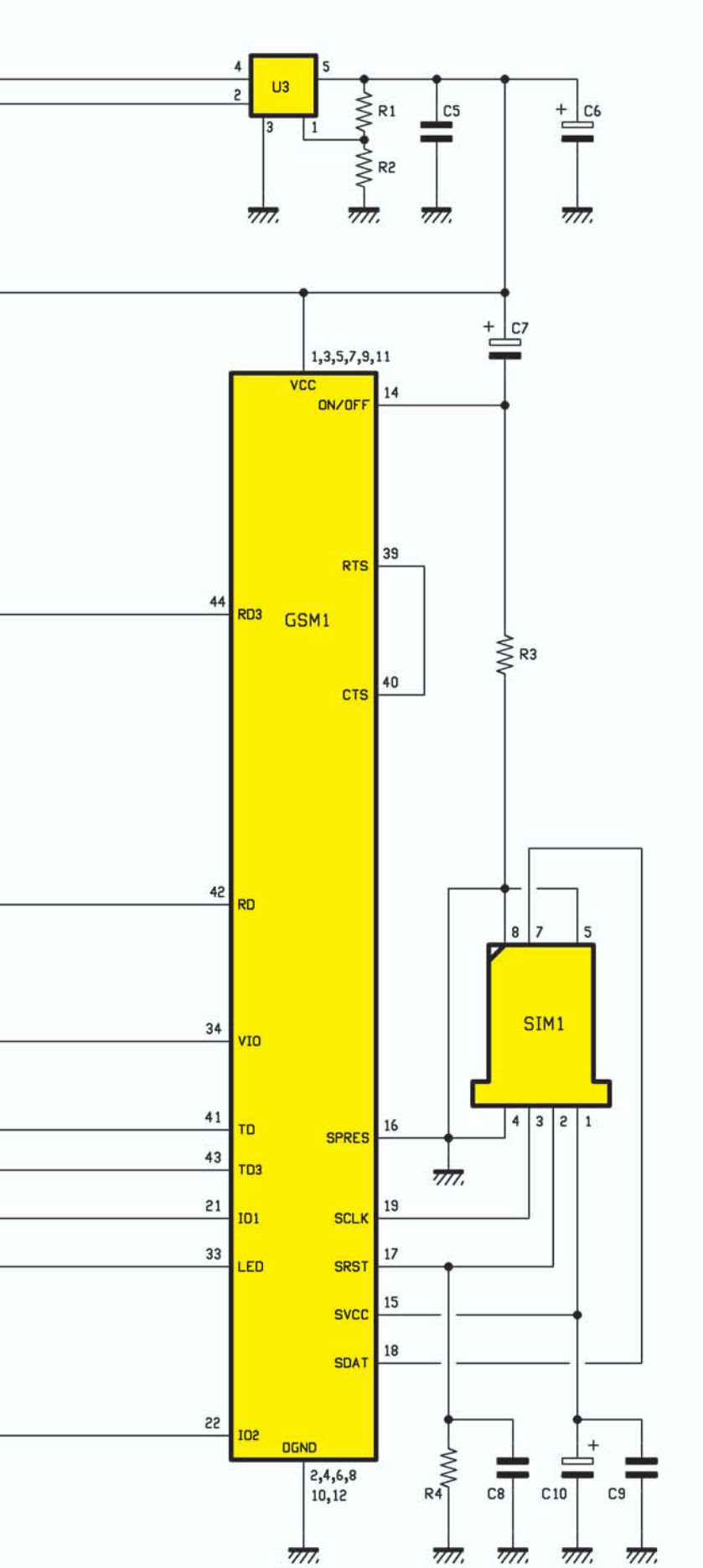


Figure 2 : Schéma électrique du serveur Web GPRS.



de réaliser des applications beaucoup plus substantielles et mieux adaptées à vos attentes que notre exemple ne le comporte.

Le schéma électrique

Notre serveur Web, dont le schéma électrique est visible figure 2, se compose d'un mobile GSM/GPRS Sony Ericsson, bien connu de nos plus fidèles lecteurs

Le fonctionnement de l'appareil

Il est géré par un microcontrôleur Microchip PIC16F876, dont les fonctions sont : lire les températures relevées par deux sondes intelligentes et les publier dans une page Web statique, sur demande, le module Sony Ericsson rend disponible pour qui y accède via Internet. L'accès est subordonné à la connexion au réseau GPRS, connexion qui, afin d'éviter des problèmes de "time-out" (délai dépassé), n'a lieu que sur demande. En fait le serveur Web fonctionne ainsi : quand le module reçoit un appel d'un numéro de téléphone préalablement mémorisé et habilité, il se connecte au réseau GPRS et attend que celui-ci lui donne une IP publique pour la connexion ; quand il l'obtient et la reconnaît, il prépare un message de texte et l'envoie au numéro qui l'a appelé pour activation. L'usager lit le SMS et, connaissant alors l'IP, il peut (par Internet) visualiser la page de données de températures relevées ; il lui suffit pour cela d'ouvrir le navigateur installé sur son PC et de taper l'IP dans la barre des adresses, avec la syntaxe de commande bien connue :

http://ipaddress:4000/index.htm

commande dans laquelle ipaddress est l'adresse et 4000 indique que l'accès se fait par le port 4000.

Précisons à ce propos qu'habituellement les connexions à Internet passent par le port 80 (celui défini par le paramétrage de réseau du système d'exploitation mais qui, malheureusement, ne permet pas le passage à notre application. Pour indiquer au serveur Internet que l'on désire utiliser le port 4000, on pourrait aussi ne pas l'écrire dans la barre des adresses et modifier le paramétrage de réseau, toutefois nous vous le déconseillons car cela exclut l'accès à la plupart des pages Web ! Mieux vaut donc pour cela définir ce paramètre dans la barre des adresses.

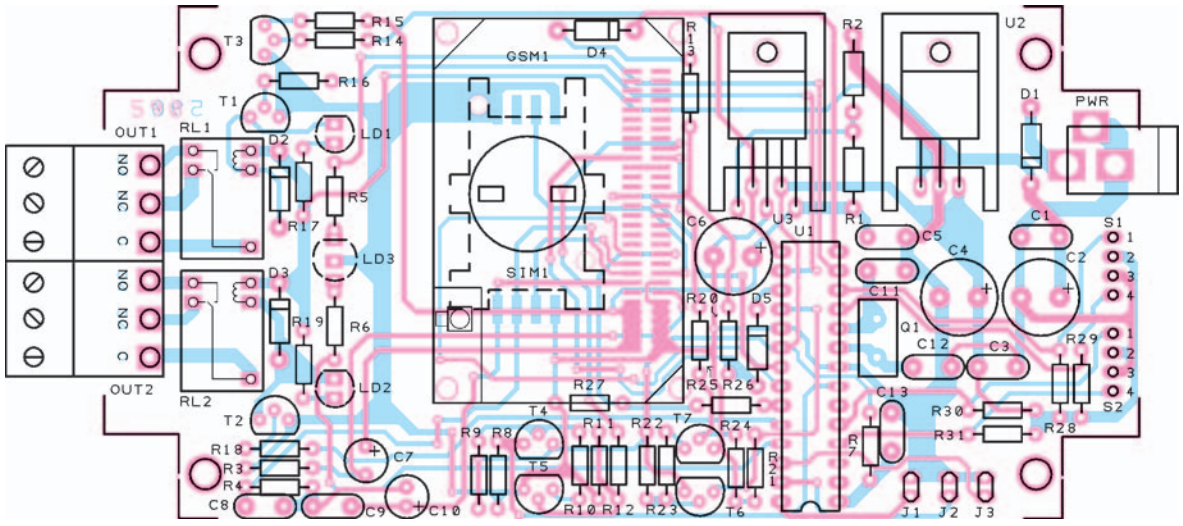


Figure 3a : Schéma d'implantation des composants du serveur Web GPRS.

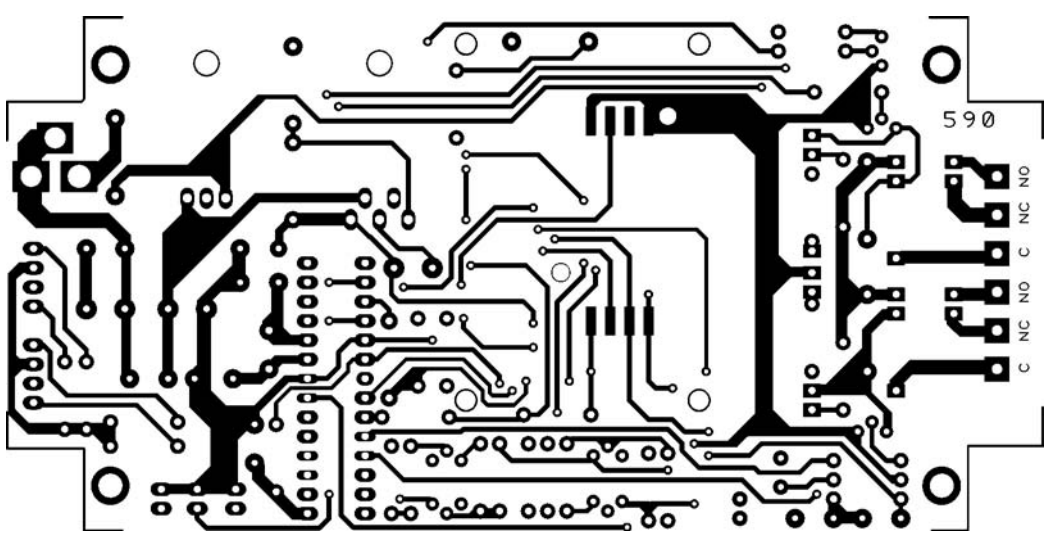


Figure 3b-1 : Dessin, à l'échelle 1, du circuit imprimé double face du serveur Web GPRS, côté soudures.

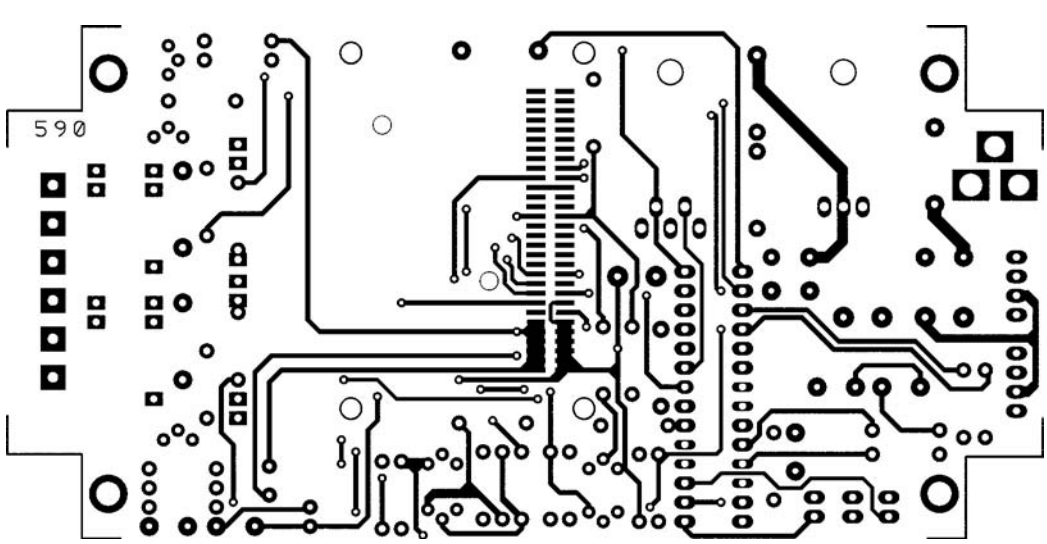


Figure 3b-2 : Dessin, à l'échelle 1, du circuit imprimé double face du serveur Web GPRS, côté composants.

Quand la connexion a été faite, le navigateur affiche la page html des températures relevées par les sondes ; la page visualisée est statique, en ce sens que les valeurs sont référées au moment où le module GPRS du serveur Web les acquiert du microcontrôleur. Nous avons décidé de ne pas les mettre à jour en temps réel afin de ne pas appesantir le logiciel de gestion du module Sony Ericsson (notre application a, nous l'avons dit, une visée didactique) et de réduire le coût d'exploitation (en effet, le service GPRS se paie en fonction de la quantité de données transmises et non en fonction du temps). Si vous vous sentez déjà de taille à modifier ce logiciel et à développer une application perso, vous pourrez, si vous préférez, obtenir une page Web mise à jour en permanence en temps réel.

Après l'exploration de la page ou, à défaut d'accès par Internet, après un délai, le serveur Web se déconnecte ; pour visualiser à nouveau la page, il faut rappeler le dispositif, attendre qu'il donne le signal occupé ; quand le SMS contenant l'IP arrive, on peut reprendre la connexion en tapant l'adresse dans la barre du navigateur comme ci-dessus (bref, bis repetita). Notez que ce passage est obligé parce que, lorsqu'on se déconnecte et se reconnecte, normalement le serveur Web reçoit une IP publique différente de la précédente ; alors, si on tape la même adresse, le navigateur risque d'afficher un message d'erreur ou bien de vous connecter à un autre usager étranger.

Bon, nous avons dit que le module Sony Ericsson se connecte au réseau GPRS seulement quand il reçoit un appel du numéro habilité ; mais comment enregistrer ce numéro de téléphone ? Facile : avec le mobile que vous utiliserez pour gérer le serveur Web, envoyez au module un message de texte contenant seulement les cinq derniers chiffres du code IMEI dudit module (il est facilement lisible sur son étiquette, comme le montre la figure 8) sans espace ni tiret. Quand vous l'aurez fait, vous serez sûr que le serveur Web ne s'activera que quand il recevra un appel de ce téléphone (à l'exclusion de tout autre). Par contre, si d'un autre mobile vous envoyez un nouveau message contenant les cinq derniers chiffres de l'IMEI, le numéro de ce mobile remplacera le numéro précédemment enregistré (habilité, donc) ; en effet, le serveur Web n'admet qu'un seul numéro habilité à la fois et donc chaque nouveau SMS de commande, s'il est correct, désactive le précédent numéro en mémoire et le remplace par le dernier numéro ayant envoyé une commande valide.

Voilà synthétiquement comment fonctionne l'appareil, du moins pour ce qu'on peut en voir de l'extérieur.

Le fonctionnement vu de plus près

Mais à l'intérieur il se passe beaucoup plus de choses, essentiellement orchestrées par le microcontrôleur PIC16F876.

Après la mise sous tension et l'initialisation des lignes d'E / S, il fait tourner son programme principal ("main program") dont le rôle est, notamment, de recevoir les données du module GSM/GPRS, d'interroger les deux sondes de température et d'envoyer à son tour des données traitées au module, afin de préparer le message de texte. Le PIC continue de vérifier si le module GSM qu'il gère détecte l'arrivée d'un nouveau message de texte ou d'un appel.

Dans le premier cas, il analyse le texte et, si les cinq derniers chiffres de l'IMEI s'y trouvent, il mémorise le numéro du mobile ayant envoyé le SMS.

Dans le second, il vérifie le numéro de téléphone qui appelle puis interrompt la connexion en faisant en sorte que l'appelant reçoive la tonalité occupé (ainsi, l'interrogation ne coûte rien à l'appelant).

Liste des composants

R1 ..... 200 k 1%
R2 ..... 100 k 1%
R3 ..... 4,7 k
R4 ..... 1 k
R5 470
R6 470
R7 ..... 4,7 k
R8 ..... 10 k
R9 ..... 10 k
R10 .... 4,7 k
R11 .... 10 k
R12 .... 4,7 k
R13 .... 4,7 k
R14 .... 4,7 k
R15 470
R16 .... 4,7 k
R17 .... 4,7 k
R18 .... 4,7 k
R19 .... 4,7 k
R20 .... 4,7 k
R21 .... 10 k
R22 .... 10 k
R23 .... 4,7 k
R24 .... 10 k
R25 .... 4,7 k
R26 .... 2,2 k
R27 .... 4,7 k
R28 .... 4,7 k
R29 .... 4,7 k
R30 .... 4,7 k
R31 .... 4,7 k

C1 ..... 100 nF multicouche
C2 ..... 1 000 µF 25 V électrolytique
C3 ..... 100 nF multicouche
C4 ..... 1 000 µF 25 V électrolytique
C5 ..... 100 nF multicouche
C6 ..... 1 000 µF 25 V électrolytique
C7 ..... 1 µF 100 V électrolytique
C8 ..... 100 nF multicouche
C9 ..... 100 nF multicouche
C10 .... 1 µF 100 V électrolytique
C11 .... 10 pF céramique
C12 .... 10 pF céramique
C13 .... 100 nF multicouche

D1 ..... 1N4007
D2 ..... 1N4007
D3 ..... 1N4007
D4 ..... BAT85
D5 ..... BAT85
LD1 .... LED 3 mm rouge
LD2 .... LED 3 mm rouge
LD3 .... LED 3 mm bicolore

T1 ..... BC547
T2 ..... BC547
T3 ..... BC547
T4 ..... BC557
T5 ..... BC547
T6 ..... BC557
T7 ..... BC547

U1 ..... PIC16F876-EF590A déjà programmé en usine
U2 7805
U3 ..... MIC2941

GS1 .... GR47-EF590B déjà programmé en usine
Q1 ..... quartz 20 MHz

RL1 .... relais 12 V
RL2 .... relais 12 V
S1 ..... sonde SHT71
S2 ..... sonde SHT71

Divers :

1 prise d'alimentation
2 borniers 3 pôles 90°
2 barrettes mâles verticales 4 broches
3 cavaliers 2 broches
1 support 2 x 14
2 boulons 3 MA 12 mm
2 dissipateurs ML26
2 entretoises 5 mm
1 lecteur de SIM-Card

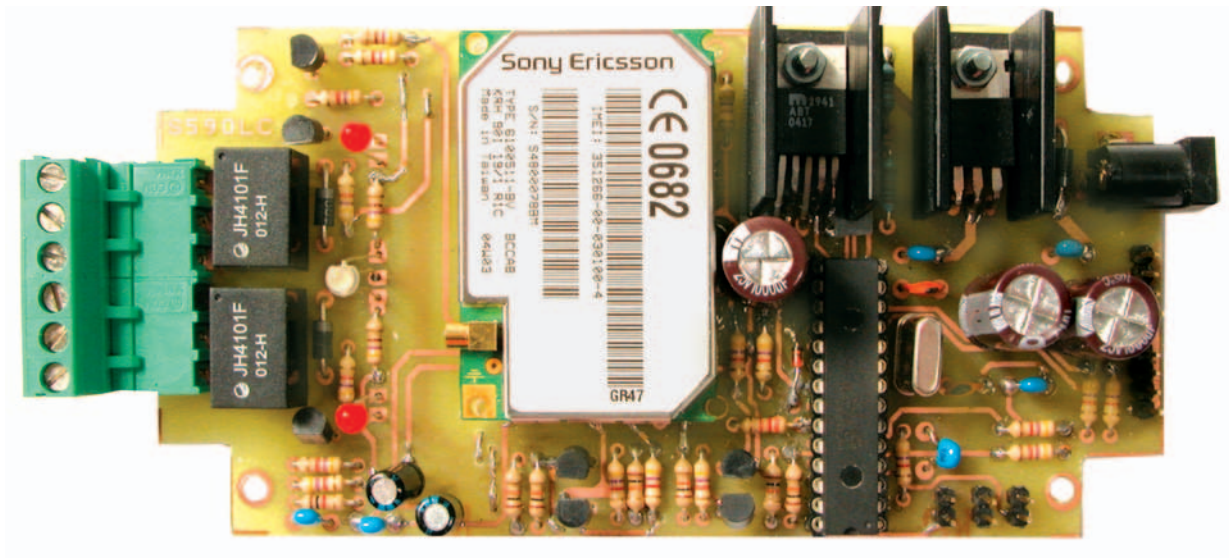


Figure 4a : Photo d'un des prototypes du serveur Web GPRS, côté composants.

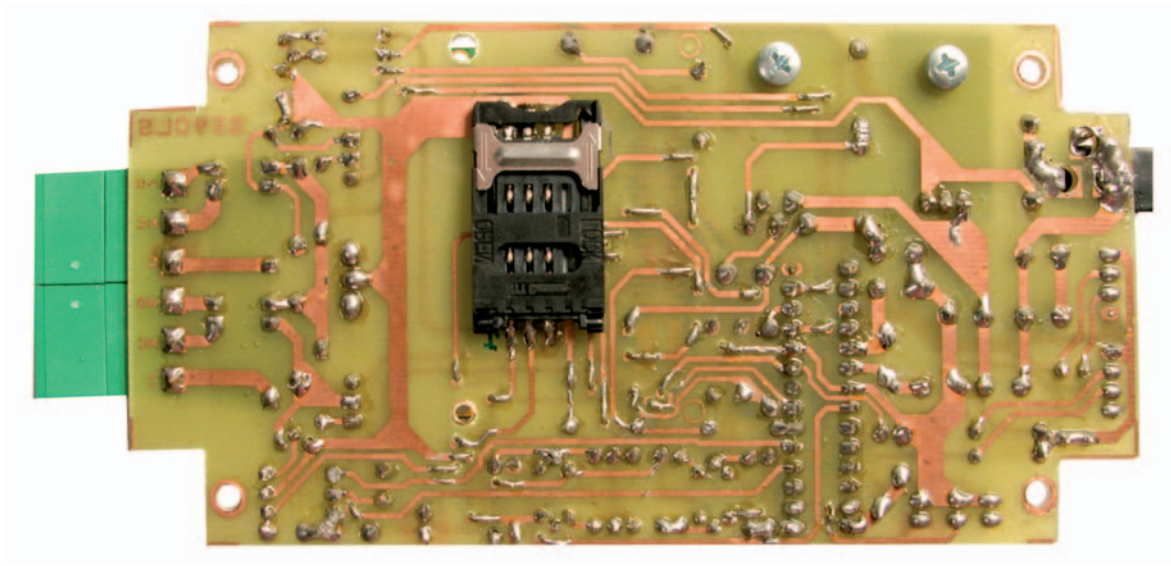


Figure 4b : Photo d'un des prototypes du serveur Web GPRS, côté soudures (où l'on monte le lecteur de carte SIM).

Deux choses peuvent alors se produire : la première, le numéro de téléphone de l'appelant est différent de celui ayant envoyé le dernier SMS de configuration reçu ; dans ce cas le PIC ignore l'opération et se dispose à attendre des appels ou des messages. La seconde, le numéro est bien celui mémorisé, habilité donc à commander la connexion GPRS ; dans ce cas le module GSM ne répond pas et raccroche, pendant que le micro attend que la connexion avec le service GPRS soit établie ; quand celle-ci est établie, il demande au module quelle est l'IP actuelle et, quand il la connaît, il prépare un message de texte qu'il envoie au numéro de téléphone dont il a reçu l'appel. Le dialogue entre le PIC et le module Sony Ericsson se fait par une ligne de communication série dotée d'un adaptateur de niveau (car le GSM travaille en 3,6 V et le micro avec des signaux TTL 0-5 V). D4 et D5, quand les lignes du micro sont à zéro volt, contraignent au niveau logique bas TD et TD3 du module ; quand ces lignes sont à 5 V, les diodes étant bloquées, les résistances de tirage R13 et R20 portent à 3,6 V (niveau logique haut) les lignes correspondantes. Voilà ce qu'il en est en transmission de micro à module.

La réception se fait à travers les interfaces réalisées avec les transistors T4/T5 et T6/T7 ; voyons la première, quand la sortie RD3 du GSM est au niveau logique haut (3,6 V), T5 est saturé et contraint R10 à presque 0 V, ce qui détermine la polarisation de la base de T4 lequel, étant un PNP, entre lui aussi en saturation (ce qui met à environ 5 V la ligne RA5 du PIC).

En revanche, avec RD3 au niveau logique zéro , les transistors T4 et T5 restent bloqués, ce qui laisse à zéro volt la broche 7 de l'intégré U1. Toujours à propos d'interface, avec l'UART interne (broches RC6/TX et RC7/RX) le micro envoie au module GSM les commandes et en reçoit les éventuelles réponses ; les données du module (caractères des messages, numéros appelant ou envoyant des SMS) transitent par la ligne RA5 (RD3 du module GSM) initialisée comme entrée.

Enfin, les informations sur les numéros à appeler et sur le texte des messages à envoyer aux mobiles sont transmises par le microcontrôleur à travers l'UART (contenues dans les commandes Hayes) et entrent dans le GSM par la broche TD3.

Figure 5 : “Listing” du lecteur de sonde.

```
SENSOR :

             ' mesure la temperature
             GoSub Idle1
             GoSub ResetComm1
             GoSub TransStart1
             OByte = shtMT
             GoSub WriteByte1
             Pause 300
             GoSub ReadByte1           ' lit l'octet haut
             SOt.Byte1 = IByte
             GoSub ReadByte1           ' lit l'octet bas
             SOt.Byte0 = IByte
             IF ERRORE1=1 THEN
                     RETURN
               ENDIF


            SOt1=SOt/100
             SOt2=SOt-(SOt1*100)
             if(SOt1>=40) then     '  temp pos
                    SOt1=SOt1-40
            else
                    SOt1=-(SOt1-40)    'temp neg
            endif

             ' measure the humidity

             GoSub Idle1
             GoSub ResetComm1
             GoSub TransStart1
             OByte = shtMH             ' mesure l'humidité
             GoSub WriteByte1
             Pause 400          ' achève la conversion
             GoSub ReadByte1
             SOrh.Byte1 = IByte
             GoSub ReadByte1
             SOrh.Byte0 = IByte

            'lecture registre
             GoSub Idle1
             GoSub ResetComm1
             GoSub TransStart1
             OByte = shtsrr            ' mesure l'humidité
             GoSub WriteByte1
             Pause 400          ' achève la conversion
             GoSub ReadByte1
             SOreg = IByte


            Sorh1=Sorh*4/100
            Sorh2=(Sorh*4)-(Sorh1*100)
            Sorh1=sorh1-4



              'calcule humidité avec compensation de la température
              'c1 =0.01
              'c2=0.00008
              if (SOt1 >=25) then                        'se temp > 25°C
                 SOrhtrue=t2/10
                 rhtrue=(sot1-25)*(1/100+sorhtrue)+(sorh1)
                 rhtrue1=rhtrue/100
                 rhtrue2=rhtrue-(rhtrue*100)

                 DEBUG "HUMY = ", dec rhtrue,",",DEC rhtrue2," %RH", 13,10
               else

                DEBUG "HUMY   = ", dec Sorh1,",",DEC Sorh2," %RH", 13,10

              endif


RETURNTesto...
```

Figure 6 : “Listing” de la page Web.

```
scpy(comandoat,»AT+CGDCONT=1»);
scat(comandoat,»\,\»IP»);
scat(comandoat,»\»,»);
scat(comandoat,»\»»);
scat(comandoat,apn);
       scat(comandoat,»»);
scat(comandoat,»\»»);


err = atsnd (comandoat,  dmyRes, slen(comandoat) , 100, &dmySize);
dlys(1);
if(err==0)
{
prtf(«\n context creato»);
break;
}
else prtf(«\n ERREUR!!!CREATION CONTESTO,code:%d»,err);

aterr = atsnd («AT*E2OTR?»,tempMicro, 10, 100, &tempMicroSize);

for (i=14;i<16; i++)
       {temp[i-14]=tempMicro[i];
       }
scpy(home,»<html><head><body><center><h1>gr 47 Webserver                    </h1></center>»);
scat(home,»<hr></h1>»);
scat(home,»<h2>maintenant tu pourras contrôler les températures de ton environnement</h2>»);
scat(home,»<h1>température interne module :»);
scat(home,temp);
scat(home,»  °C</h1>»);
scat(home,»</body></head></html>\r\n»);

if(!err)
       {
prtf(«\n pdp context written»);
        err = pdpa(1,1);
       }
       if(!err)
          {prtf(«\n pdp context activé»);
       val = ipi(0);
       }
else
{
prtf(«\n ERREUR!!!PDP CPNTEXT-code= %d»,err);
return(err);
}
inviosms(destinataire,»ip»);

        err=ipo(1,&Sck);
if(err==0)
             err=tcpl(Sck,PORTNUM);
}

        if(err==0)
{
int Port = 0;
       int Addr = 0;

err=tcpa(Sck,Port,Addr);
       prtf(«\n Accept error = %d»,err);

       }
   else prtf(«\erreur fonction tcpl!!!»);

while(errorricezione!=0)
{
if(gtf(ip_data))
{
datalen=500;

errorricezione=tcpr(Sck,datareceve,&datalen);
```

```
prtf(«\n ricevuti %d  bytes di dati»,datalen);
datareceve[datalen]=0;
prtf(«\n Risposta = %s», datareceve);
}
else
{prtf(«\n wait for send response»);
}

size=slen(home);
         err = tcps(Sck,home,&size);
   prtf(«\n Sent %d bytes, err = %d»,size,err);
                                dlys(2);
               prtf(«\n waiting for data!»);

dlys(1);
       prtf(«\n SERVER Finished!»);
```

Le micro détecte l'arrivée des messages par la condition de la ligne IO1 du module, qu'il lit à travers RB1 ; la sortie pour la LED est lue par le PIC au moyen de RB0, afin que l'on sache si le GSM1 est en réseau ou non.

Le contrôle des deux relais qui gèrent les sorties OUT1 e OUT2 est confié aux lignes RC5 et RB3 ; à ce propos, dans cette application, conçue avec une forte intention didactique (c'est pour cela qu'on le répète inlassablement !), le logiciel ne s'occupe pas des relais. Les deux se trouvent dans le circuit pour ceux d'entre vous qui, après s'être familiarisés avec le système, souhaiteraient réaliser un programme résident en mesure de commander RL1 et RL2 directement à partir de la page Web ou au moyen de commandes locales ou via SMS, par exemple quand un certain seuil programmable de température est dépassé

Le lignes RB4 et RC2 produisent les signalisations de la LED bicolore LD3, en se mettant au niveau logique haut, une à la fois pour produire le rouge ou le vert, ou ensemble pour produire l'orange.

Les sondes de température sont lues au moyen de deux canaux indépendants, dont chacun comporte une ligne d'horloge et une sur laquelle transitent les données proprement dites ; pour la sonde S1, le micro utilise RC1 comme sortie pour l'envoi de l'horloge série et RC0, initialisée comme ligne bidirectionnelle sur laquelle transitent les commandes de demande des données relevées (du micro vers la sonde) et les données proprement dites (de la sonde vers le micro).

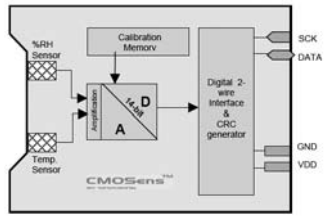
Pour S2 ce sont les lignes RA3 (horloge) et RA2 (E / S bidirectionnelles du canal données) ; comme prévu par le constructeur, toutes les lignes destinées à la communication avec les sondes sont pourvues de résistances de tirage reliées au positif 5 V, indispensables pour assurer les niveaux logiques hauts.

Le protocole de communication avec les capteurs de température prévoit l'interrogation de la part du PIC au moyen d'une instruction qui commence en mettant au niveau logique bas la ligne DATA quand SCK est au niveau logique haut ; donc en envoyant, DATA se portant et se maintenant au niveau logique haut, une impulsion au zéro logique quand SCK a repris elle aussi le niveau logique haut.

Après la mise sous tension et l'initialisation des lignes d'E / S, le programme résident du PIC réinitialise l'interface de communication avec les sondes en envoyant à chacune d'elles une série de

Figure 7 : La sonde de température.







Pour détecter les températures à publier sur la page Web, le microcontrôleur se base sur les deux capteurs Sensirion (www.sensirion.com) SHT71, chacun contenant une sonde thermique et un détecteur d'humidité relative ; les informations sont échantillonnées (au moyen d'un convertisseur A/N à 14 bits) et envoyées, à travers une logique adéquate, au circuit qui en fait la demande. Pour la communication, le capteur utilise une ligne à deux fils assimilable à un bus I2C : par sa broche DATA il reçoit les demandes et envoie les données de température et d'humidité détectées ; la communication est établie par le signal d'horloge reçu de la broche SCK. Les données envoyées par le micro utilisé pour gérer le composant et celles que l'unité logique de ce dernier envoie au micro sont lues au moment de chaque front de montée du signal d'horloge. En ce qui concerne la température, le SHT71 peut mesurer entre –40 et +123 °C ; quant à l'humidité, la gamme de mesure va de 0 à 100%. Le protocole de communication avec la sonde prévoit que le microcontrôleur commence la demande en mettant au niveau logique bas la ligne DATA quand l'horloge est au 1 logique ; donc, quand DATA est au niveau logique haut, en donnant une impulsion à zéro sur la ligne SCK. La structure de la réponse, contenant humidité et température, est celle qui apparaît ci-dessous ; elle prévoit des flux dont les données sont exprimées dans l'ordre du bit le plus significatif à celui le moins significatif (d'abord le bit 7, puis le bit 0) suivis d'une somme de contrôle servant au PIC pour en vérifier l'exactitude.

Figure 8 : L'habilitation du mobile.



Pour habiliter un téléphone mobile à l'activation du serveur Web et à la réception du SMS avec les données de l'IP, il est nécessaire d'envoyer avec le même mobile un SMS contenant le code IMEI du module GPRS (donnée présente sur l'étiquette du dispositif).

neuf impulsions d'horloge (le long de la ligne SCK) pendant qu'il maintient DATA au niveau logique haut.

La commande de "reset" de la ligne de communication et de l'unité logique du capteur peut avoir lieu à tout moment, quand des problèmes d'échange de données peuvent surgir ; pas de risque de perte d'informations échantillonnées (température et humidité), car elle ne concerne que la section de communication et non le "buffer" où les données résident.

Il ne nous reste qu'à dire trois mots sur l'alimentation : l'alimentation principale 12 Vcc est appliquée aux points + et –PWR ; le régulateur U2 fournit 5 V au microcontrôleur et U3, un régulateur à rétroaction (par R1 et R2), fournit 3,6 V entre la masse et la broche 5 du module Sony Ericsson.

La réalisation pratique

La réalisation pratique de ce serveur Web est assez simple. La platine est constituée d'un circuit imprimé double face, dont les figures 3b-1 et 2 donnent les dessins à l'échelle 1. N'oubliez surtout pas de réaliser les liaisons entre les deux faces : d'abord avec de petits morceaux de fil de cuivre (des chutes de queues de composants) soudés des deux côtés et ensuite en soudant également sur les deux faces les fils et broches des composants, chaque fois que cela est possible.

Commencez par insérer (côté composants) les supports du PIC et du module GR47, les "straps" et les trois cavaliers à deux broches J1, J2 et J3. Attention, en particulier, au support à 60 broches (2 x 30) du module GSM/GPRS, car c'est un support CMS : utilisez un fer à panne fine et du tinol de 0,5 mm. Vérifiez attentivement vos soudures (ni court-circuit entre pistes ou pastilles ni soudure froide collée).

Insérez et soudez ensuite tous les composants (comme le montrent les figures 3a et 4a), en poursuivant par les résistances, condensateurs, diodes, LED, transistors boitiers TO 92, régulateurs, quartz et relais et en terminant par les "périphériques" : la prise d'alimentation, les borniers à 3 pôles à 90°.

Attention à l'orientation des composants polarisés : PIC (repère-détrompeur en U vers le bas, mais insérez-le à la toute fin), diodes, LED, transistors, régulateurs et électrolytiques. Le quartz est monté debout. Les deux régulateurs sont montés couchés dans leurs dissipateurs ML26 et maintenus par des boulons 3 MA (comme il y a des pistes sous ces dissipateurs, il est préférable de les isoler avec du carton ou du Téflon gris). Le module sera inséré à la toute fin. Voilà pour le côté composants.

Retournez alors la platine côté soudures (voir figures 3a et 4b) et montez le lecteur de SIM-Card. Vérifiez bien toutes les polarités et (encore une fois) la qualité des soudures sur les deux faces de la platine, puis insérez le PIC et le module dans leurs supports (attention à la polarité du PIC).

Vous pouvez maintenant installer la platine dans un boîtier plastique de dimensions appropriées : le couvercle sera percé de trois trous pour le passage des LED et les deux petits côtés seront évidés, l'un pour la prise d'alimentation jack 12 V et l'autre pour le câble de l'antenne GSM et les deux borniers à 3 pôles.

Les essais et le paramétrage

Avant de refermer le couvercle du boîtier, insérez une carte SIM valide dans son lecteur et paramétrez les trois cavaliers en fonction des températures que vous souhaitez voir apparaître sur la page Web. Souvenez-vous que J1 fermé habilite la publication de la valeur relevée par la sonde S1 et que J2 fermé permet la consultation de la lecture de S2 ; enfin, en fermant J3, on publie aussi la température lue par le capteur dont est doté le module GSM/GPRS GR47 et qui a été prévu pour détecter toute surchauffe due à un endommagement du circuit ou à une température ambiante excessive.

A propos du module, n'oubliez pas de le relier à une antenne 900/1 800 MHz adéquate, à l'aide d'un connecteur adaptateur MMCX/FME.

L'alimentation à brancher à la prise jack latérale doit fournir 12 Vcc pour un courant de 800 à 1 000 mA, attention, le + est au centre.

Quand vous alimentez le circuit, attendez que la LED bicolore s'allume en vert : cette couleur signale que l'on est en phase de mise sous tension (!) et d'initialisation. Avec un téléphone mobile, envoyez au serveur Web le message de configuration (adressez-le au numéro de la SIM insérée dans le lecteur au verso de la platine du circuit). Quand le message est reçu, le circuit fait clignoter en jaune la LED bicolore et, si tout est OK, il envoie au téléphone

Figure 9 : Comment fonctionne le GPRS.



Le GPRS (General Packet Radio Service) est le maillon entre le GSM et l'UMTS, le système actuellement utilisé pour la vidéotéléphonie radiomobile : il est supérieur au GSM car il optimise la bande utilisée ; toutefois, comme il fonctionne sur le réseau GSM, il est limité en vitesse de communication. Il est basé sur une méthode dite "à communication de paquet" et ses ressources radio ne sont effectivement mobilisées qu'au moment d'échanger des données et non pendant toute la durée de la connexion. Les données à transmettre sont divisées en paquets, chacun contenant les informations concernant sa propre destination. Les paquets sont ensuite insérés dans le réseau et ils doivent donc trouver leur route jusqu'au destinataire de manière autonome ; ce qui implique que l'envoyeur et le destinataire n'ont pas à être forcément reliés l'un à l'autre et que les données peuvent être reçues par une série de plusieurs envoyeurs/destinataires au cours d'un même appel. Il est ainsi possible pour plusieurs usagers d'utiliser en même temps des ressources du réseau, comme cela se fait dans une LAN ou sur Internet : sur chaque canal de trafic, dans des trames différentes, des paquets de divers usagers peuvent voyager.

La différence avec la technologie GSM est que ce dernier se base sur un système "à commutation de circuit", ce qui signifie qu'à chaque usager est attribué en exclusivité un canal de trafic demeurant à sa disposition pendant toute la durée de la session, ce qui mobilise les ressources radio jusqu'à une demande explicite de "raccrochage" ; la ligne est ouverte et les données voyagent librement entre l'envoyeur et le destinataire. Si l'application demande une attente pour la réponse du destinataire, la ligne doit rester ouverte même si aucune donnée n'est transmise. En tant que solution pour le transport des données, le GSM est évidemment inefficace et extrêmement coûteux.

Avec la transmission par paquets garantie par le GPRS, les coûts de service pour le client sont optimisés car elle autorise l'accès aux services en continu et ne mobilise les ressources du réseau que pour la quantité d'informations effectivement transférées. Le téléphone reste constamment relié au réseau et l'usager peut simultanément effectuer les appels traditionnels et recevoir des informations (prévisions météo, cours de la bourse, etc.). En outre le GPRS permet de relier l'ordinateur portable à Internet à travers le téléphone mobile, sans les lenteurs subies jusqu'à aujourd'hui.

Si on souhaite approfondir l'argument, on peut ajouter que les téléphones et les modems GPRS communiquent avec la station GSM au moyen d'un Serving GPRS Support Node (SGSN) représentant la liaison avec l'infrastructure GSM qui envoie et reçoit les données par et pour l'appareillage mobile. Le SGSN communique avec le Gateway GPRS Support Node (GGSN), un système assurant les communications avec les autres réseaux comme Internet, ou les réseaux X.25 ou LAN privées. Un réseau GPRS peut utiliser plusieurs SGSN, mais requiert un GGSN unique pour la connexion à des réseaux externes. Quand l'appareillage mobile envoie des paquets de données, ces derniers passent du SGSN au GGSN qui les convertit au format compatible avec le réseau de destination. De la même manière, les paquets venant d'Internet pour l'appareillage mobile sont initialement reçus par le GGSN, puis passés au SGSN et enfin transmis à l'appareillage. La vitesse de transmission maximale du GPRS est théoriquement de 171,2 kbps (une amélioration non négligeable par rapport aux 9,6 ou 19,2 kbps du réseau GSM) et on peut l'obtenir tous les 8 "timeslots" disponibles et sans encourir d'erreur de protection. Bref, il est très improbable qu'un opérateur permette à un seul usager l'utilisation de tous les "timeslots" : la vitesse effective dépend donc des limitations propres au réseau et aux terminaux. Grâce à la vitesse qu'il permet, le GPRS rend possible l'accès à Internet et le chargement de fichiers du Web ; mais il autorise également une gestion efficace de la poste électronique, la consultation des banques de données et des services d'information disponibles en ligne ; le commerce électronique par téléphonie mobile peut ainsi se développer. Et tout cela sans oublier bien sûr le coup de téléphone traditionnel (après tout, c'est ce qui a fait la fortune de ces téléphones qu'on appela d'abord portables).

Les limites du système GPRS dépendent étroitement de sa structure : comme les paquets au moyen desquels les données sont transmises voyagent indépendamment les uns des autres, ils peuvent suivre des parcours divers et arriver au destinataire pas forcément dans l'ordre de leur envoi ; chaque paquet doit donc contenir toutes les informations pour pouvoir donner au destinataire la possibilité de reconstruire la séquence de transmission exacte. Comme la durée du voyage et donc le retard de leur arrivée ne sont pas prévisibles avec une bonne précision, il peut arriver que, pour reconstruire une information en partant des différents paquets, un délai assez long se produise, non cohérent avec le "data transfer" promis, lequel se réfère à un seul paquet : une limite pour les applications nécessitant un "data rate" constant. Autre inconvénient : quand tous les usagers utilisant le même canal GPRS veulent communiquer au même moment, une congestion du système risque de se produire, avec pour conséquence un retard dû aux codes de transmission. En outre, on l'a dit, quand les quantités de données à échanger deviennent significatives, le système de transfert de données par paquet n'est plus la solution optimale. Le principal problème est donc la charge effective du réseau en termes de "data transfer" et de nombre d'usagers connectés.

mobile un message de confirmation "mobile habilité au système".

Ah, dernier point : cela ne marche pas avec Vodaphone, car ce dernier insère un filtre empêchant d'accéder à l'unité mobile par le réseau Internet GPRS.

Comment construire ce montage ?

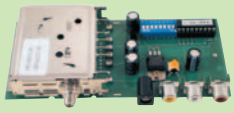
Tout le matériel nécessaire pour construire ce serveur Web GPRS ET590 est disponible chez certains de nos annonceurs. Voir les publicités dans la revue.

Les typons des circuits imprimés ainsi que l'ensemble des programmes disponibles sont téléchargeables sur votre site à l'adresse suivante :

http://www.electronique-magazine.com/circuitrevue/086.zip. ◆

ÉMETTEUR 1,2 & 2,4 GHz

RÉCEPTEUR 1,2 & 2,4 GHz



Nouveau 1.2 GHz
1.255 GHz 1 Watt



EMETTEUR 1.2 & 2,4 GHz 20, 200 et 1000 mW

Alimentation :13,6 VDC. 4 fréquences en **2.4 GHz** :**2,4 - 2,427 - 2,454 - 2,481 GHz** ou 8 fréquences en **1.2 GHz 20 mW: 1,112 - 1,139 - 1,193 - 1,220 - 1,247 - 1,264 - 1,300 GHz** ou 4 fréquences en **1.2 GHz 1 W: 1,120 - 1,150 - 1,180 - 1,255 GHz**. Sélection des fréquences : dip-switch. Stéréo : audio 1 et 2 (6,5 et 6,0 MHz). Livré sans alim ni antenne.

TX2-4G ..... Emetteur 2,4 GHz 4 c monté 20 mW ..... 39,00 €
TX2-4G-2-... Emetteur monté 4 canaux 200 mW ..... 99,00 €
TX1-2G ..... Emetteur 1,2 GHz 20 mW monté 4 canaux ..... 38,00 €
TX1-2G-2-... Emetteur 1,2 GHz monté 1 W 4 canaux ..... 99,00 €

VERSION 256 CANAUX

Ce petit kit se monte sur les emetteurs TX2.4G et TX1.2G et permet d'augmenter leur nombre de canaux à 256. Le pas est de 1 MHz et la sélection des canaux se fait par dip-switch. Fréquences de départ : 2,3 pour les versions TX2,4G et 1,2 pour les TX 1,2G Cette extension est vendue sans l'emetteur.

TEX1.2 ..... Kit extension 1,2 à 1,456 GHz..... Promo .....19,80 €
TEX2.3 ..... Kit extension 2,3 à 2,556 GHz..... Promo .....19,80 €

RÉCEPTEUR 4 CANAUX 1,2 & 2,4 GHz

Alimentation : 13,6VDC. 4 fréquences en **2.4 GHz** :**2,4 - 2,427 - 2,454 - 2,481 GHz** ou 8 fréquences en **1.2 GHz** : **1,112 - 1,139 - 1,193 - 1,220 - 1,247 - 1,264 - 1,300 GHz**. Sélection des fréquences : dip-switch pour le 1,2 GHz et par poussoir pour les versions 2,4 GHz. Stéréo : audio 1 et 2 (6,5 et 6,0 MHz). Fonction scanner pour la version 1.2 GHz. Livré sans alimentation ni antenne.

RX2-4G.....Récepteur monté 2.4 GHz 4 canaux..... 39,00 €
RX1-2G.....Récepteur monté 1.2 GHz 4 canaux..... 38,00 €

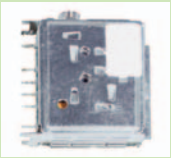
VERSION 256 CANAUX



Ce petit kit se monte sur les récepteurs RX2.4G et RX1.2G et permet d'augmenter leur nombre de canaux à 256. Le pas est de 1 MHz et la sélection des canaux se fait par dip-switch. Fréquences de départ au choix: 2,3 pour les versions RX2,4G et 1,2 pour les RX 1,2G Cette extension est vendue sans l'emetteur.

REX1.2..... Kit extension 1,2 à 1,456 GHz..... Promo .....19,80 €
REX2.3..... Kit extension 2,3 à 2,556 GHz..... Promo .....19,80 €

MODULES RX 2,4 GHz & MODULES TX 2,4 GHz



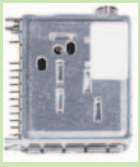
Module RX programmable en I2C-BUS entre 2 et 2,7 GHz ou 1.1 et 1.6 selon la version; alimentation 12 V.

RX24MOD Module 2.4 G.....~~30,00 €~~ Promo.....25,00 €.

Module TX d'environ 20 mW programmable en I2C-BUS entre 2 et 2,7 GHz ou 1.1 et 1.6 selon la version; alimentation 12 V.

TX24MOD Module 2.4 G 20 mW.....~~27,00 €~~ Promo.....22,00 €

TX24MOD2 Module 2.4 G 200 mW.....~~87,00 €~~ Promo.....72,00 €



Cette antenne directive patch offre un gain de 8,5 dB. Elle s'utilise en réception aussi bien qu'en émission et permet d'augmenter considérablement la portée des dispositifs RTX travaillant sur des fréquences. Ouverture angulaire: 70° (horizontale), 65° (verticale). Gain: 8,5 dB. Connecteur de sortie: SMA femelle. Impédance: 50 Ω. Dim.:90x 120 x 20 mm. Poids: 130 g. Puissance max.: 100 Watts

ANT-8080N..... Antenne patch ..... 52,00 €
CORDON/C ..... Câble SMA Male / SMA Male ..... 9,90 €



ANTENNE GP24001 POUR 2.4 GHz

OMNI. POLAR. VERTICALE, GAIN 8 DBI, HAUTEUR 39 CM.
99,50 €



ANTENNES "BOUDIN" 2,4 GHZ & 1,2



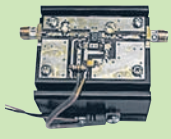
ANT-STR..... Ant. droite 2.4 GHz.. 6,00 €
ANT-2G4 ..... Ant. coudée 2.4 GHz 7,00 €
ANT-STR12 Ant. droite 1.2 GHz... 7,00 €

PARABOLES GRILLAGÉES 2,4 GHZ,



acier inoxydable, connecteur N mâle, puissance max. 50 W, impédance 50 Ω.
ANT SD15, gain 13 dBi, dim. : 46 x 25 cm, 2,5 kg .....**37,00 €**
ANT SD27, gain 24 dBi, dim. : 91 x 91 cm, 5 kg .....**69,00 €**

AMPLI 1,3 W 1,8 à 2,5 GHz



Alimentation: 9 à 12 V. Gain: 12 dB. P. max.: 1,3 W. F. in: 1 800 à 2 500 MHz.
AMP2-4G-1W…Livré monté et testé ..... 135,70 €

TX/RX 2.4 GHZ AVEC CAMERA COULEUR

Ensemble émetteur récépteur audio/vidéo offrant la possibilité (à l'aide d'un cavalier) de travailler sur 4 fréquences différentes dans la bande des 2,4 GHz . Portée en champs libre: 200 à 300 mètres. Entrée audio : 2 Vpp max. antenne. Existe en trois versions différentes pour la partie emettrice. L'émetteur miniature intégre une caméra CCD couleur Chaque modèle est livré complet avec un émetteur, un recepteur, les antennes et les alimentations









ER245..... Dim TX (44 x 56 mm); Alim 5 à 8 V Poids 200 g puissance 10 mW .....125,00 €
ER242..... Modèle ultra léger: Dim TX (23x23x23 mm),alim 5 à 8 V et poids 10 g, puissance 10 mW.....125,00 €
ER226..... Moniteur 5.6"LCD PAL/NTNC,Télécommande, alim 12VDC ou 230 AC.....175,00 €
ER124..... Moniteur 7"LCD PAL/NTNC/SECAM,Télécommande, alim 12VDC ou 230 AC .....250,00 €

COMELEC **CD 908 - 13720 BELCODENE** **WWW.comelec.fr**
Tél. : 04 42 70 63 90 Fax : 04 42 70 63 95

Expéditions dans toute la France. Moins de 5 kg : Port 8,40 €. Règlement à la commande par chèque, mandat ou carte bancaire. Bons administratifs acceptés. Le port est en supplément. De nombreux kits sont disponibles, envoyez votre adresse et cinq timbres, nous vous ferons parvenir notre catalogue général.

Photos non contractuelles. Publicité valable pour le mois de parution. Prix exprimés en euro toutes taxes comprises. Sauf erreurs typographiques ou omissions.

Création COMELEC 06/2005

Une alimentation double symétrique professionnelle

Troisième partie : la fin de la réalisation pratique des platines modulaires

Alimentation professionnelle de laboratoire, ETALI est entièrement gérée par microcontrôleur ; elle fournit deux tensions continues stabilisées, symétriques par rapport à la masse et réglables de +1/0/–1 V à +36/0/–36 V. C'est l'outil idéal pour faire fonctionner des circuits à alimenter sous tension simple ou double symétrique ; elle peut fournir un courant de 3 A par branche. Les valeurs sont réglées par poussoirs et visualisées sur afficheur LCD. Dans cette troisième et dernière partie, nous terminons le montage du dernier module et mettons la dernière main à l'ensemble.

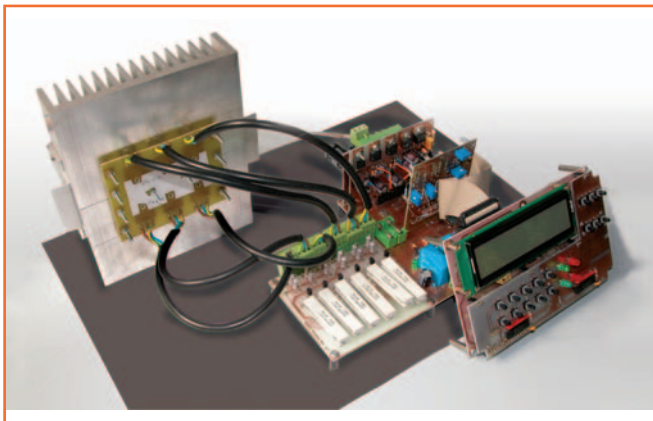


Schéma synoptique de l'alimentation modulaire

ETALI = MF + MA + MP (carte-mère) + MD + MM (avec afficheur LCD) + MPSR ... et simulateur de température.
MF = Module de filtrage
MA = Module des alimentations
MP = Module de puissance (platine de base recevant les autres modules)
MD = Module DAC
MM = Module microcontrôleur (comporte l'afficheur LCD et reçoit le MPSR)
MPSR = Module des poussoirs (fixé sur MM)

Les caractéristiques techniques et les fonctions

Chaque alimentation de laboratoire se distingue des autres par ses caractéristiques techniques et par le nombre de fonctions dont elle est dotée. Voici les caractéristiques techniques et les fonctions de notre ETALI :

- Gestion numérique des tensions de sortie par microprocesseur.
- Convertisseur DAC : à 10 bits.
- Résolution tension de sortie : + ou –50 mV.
- Tension stabilisée maximale canal positif : +36 V.
- Tension stabilisée minimale canal positif : +1 V.
- Tension stabilisée maximale canal négatif : –36 V.
- Tension stabilisée minimale canal négatif : –1 V.
- Protection électronique des transistors et fusibles pour le courant max.
- Afficheur LCD alphanumérique pour la gestion des menus de contrôle : à 2 lignes de 16 caractères.
- Protection par microprocesseur pour le courant max de seuil : si le seuil est franchi, la tension de sortie est mise à zéro par déclenchement du relais correspondant.
- Deux sondes de température pour la surveillance des darlingtons de puissance sur les deux canaux.
- Ventilateur tachymétrique à 3 fils : la vitesse de rotation dépend de la température du dissipateur ; le ventilateur est alimenté par une tension PWM produite par le microcontrôleur ; trois gammes de vitesses correspondant à trois valeurs de température sont prévues.
- Courant max pouvant fournir chaque canal : 3 A.

Nous voici enfin arrivés à la dernière phase d'analyse et de réalisation de notre alimentation de labo professionnelle ETALI. Avant de passer à l'étude des platines qui manquent encore (module des poussoirs MPSR et simulateur de température), jetez un coup d'œil à l'encadré ci-dessus et vous aurez un récapitulatif des caractéristiques techniques de cette alimentation double symétrique réglable, protégée et gérée par microcontrôleur (contrôle par afficheur LCD) et de son organisation modulaire.

Il nous reste maintenant à nous pencher sur la platine interface usager, soit le module des poussoirs MPSR. Mais auparavant, ajoutons un mot à propos du module microcontrôleur MM : il comporte un connecteur CN4, utilisé pour la connexion

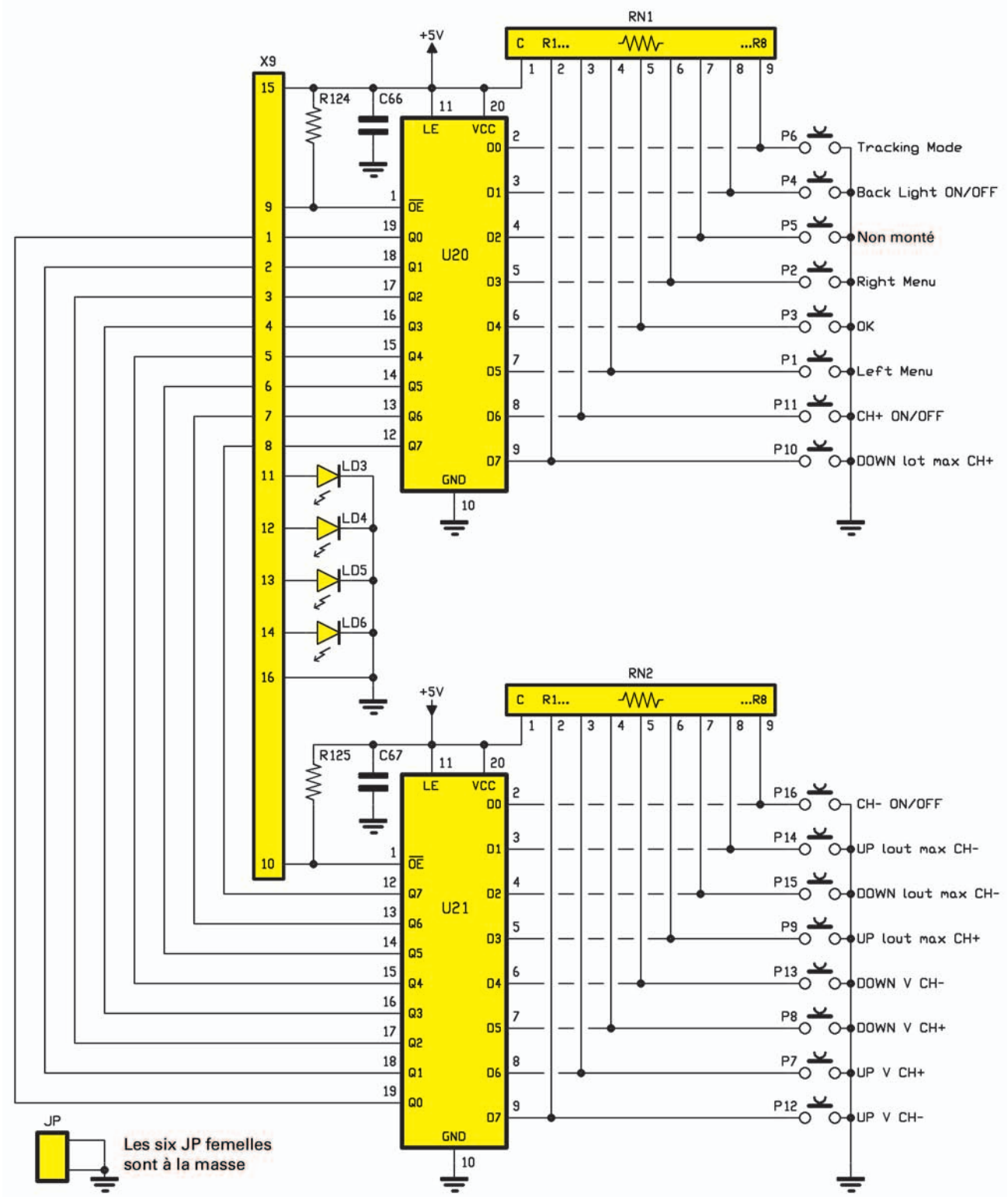


Figure 1: Schéma électrique du Module des poussoirs de l'alimentation de laboratoire professionnelle.

avec l'ICD2 ; cela permet de se servir de ce dispositif comme émulateur pour écrire le programme résident du PIC et exécuter le débogage (en effet, la possibilité d'insérer un "break point", ou point d'arrêt, a été prévue) et comme programmateur. ICD2 est très commode pour la programmation "in-circuit". Le programme résident occupe les quatre bancs de mémoire "flash" disponibles ; il est divisé en blocs fonctionnels que nous résumons ci-dessous et que nous appelons "pages".

Page 0 :

- Routine de gestion E / S (lit et écrit les ports du PIC) ;
- Routine de gestion poussoirs (interprète les commandes à exécuter en fonction du poussoir pressé) ;
- Routine d'initialisation du PIC ;
- Routine gestion SPI ;
- Routine gestion sondes NTC ;
- Routine lecture courant consommé par la charge ;
- Routine gestion interruptions.

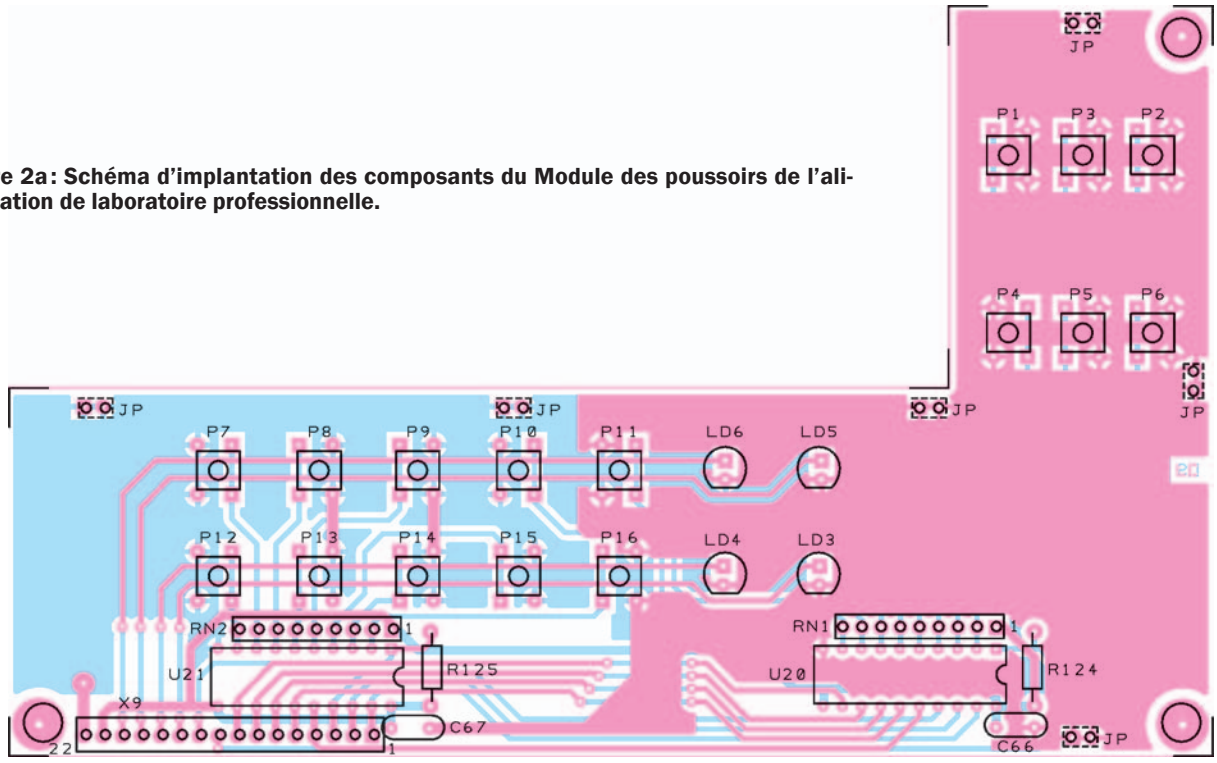


Figure 2a : Schéma d'implantation des composants du Module des poussoirs de l'alimentation de laboratoire professionnelle.

Page 1 :

- Routine gestion hardware afficheur LCD ;
- Routine gestion menu afficheur LCD
- Routine mathématique.

Page 2 :

- Non utilisée (2 ko libres).

Page 3 :

- Tableau linéarisation température lue par les sondes NTC ;
- Tableau linéarisation courant consommé par la charge ;
- Tests à visualiser sur afficheur LCD ;
- Les 256 derniers octets doivent être laissés libres pour le débogage ; ils servent au fonctionnement correct de l'ICD2.

Le Module des poussoirs

Voyons le matériel du bloc contenant l'interface pour la réception des commandes. Il s'agit d'une platine comportant les seize poussoirs grâce auxquels nous pouvons paramétrer le fonctionnement du système tout en utilisant l'afficheur LCD comme moniteur. Dans ce MPSR on trouve deux "latches" (verrous)

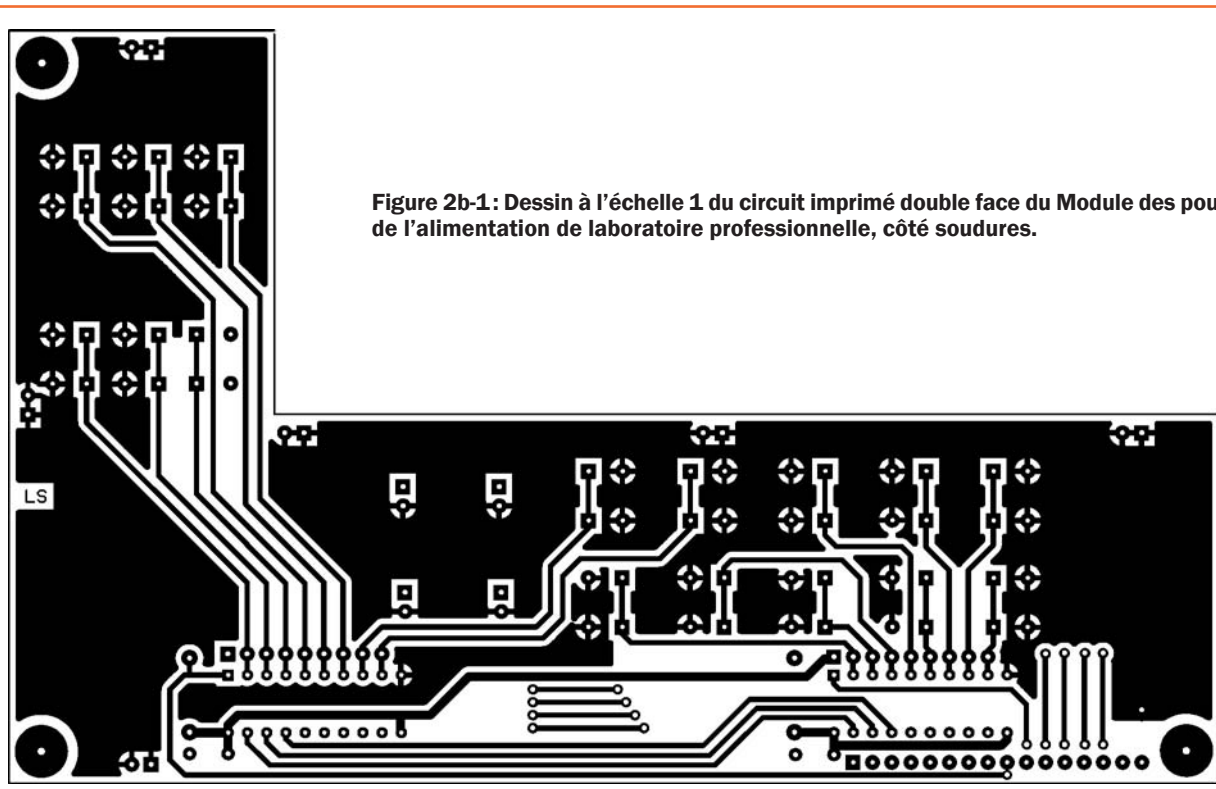


Figure 2b-1 : Dessin à l'échelle 1 du circuit imprimé double face du Module des poussoirs de l'alimentation de laboratoire professionnelle, côté soudures.

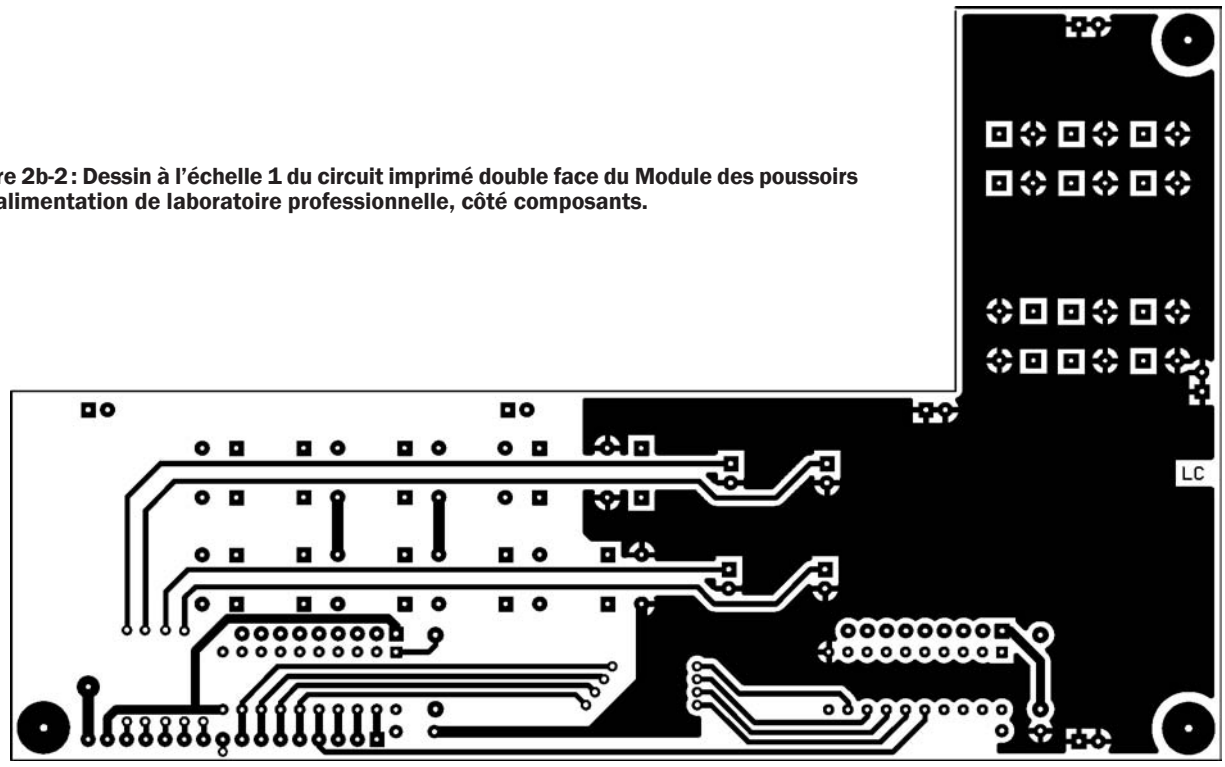


Figure 2b-2 : Dessin à l'échelle 1 du circuit imprimé double face du Module des poussoirs de l'alimentation de laboratoire professionnelle, côté composants.

74HCT573 permettant au micro de lire, avec seulement huit lignes d'E / S, l'état de ces seize poussoirs ; les verrous servent essentiellement à charger les états logiques déterminés en chacun par le groupe correspondant des huit poussoirs, puis à les transférer au bus du micro quand le logiciel le demande. La lecture se fait périodiquement et assez rapidement pour rendre cette opération transparente pour l'usager (c'est-à-dire plus rapidement que les doigts de l'usager sur les poussoirs) ; le transfert des données lues au micro se fait par contre alternativement, selon un cycle de trois phases : au cours de la première, le 74HCT573 situé sur le MM et utilisé pour gérer les LED et les relais locaux est activé, au cours de la deuxième, le PIC lit les données à travers le premier verrou correspondant aux huit premiers poussoirs et au cours de la troisième, il active seulement le troisième verrou pour prélever les états des huit derniers poussoirs. Puis le cycle recommence depuis le début.

Le schéma électrique

Jetons un coup d'œil au schéma électrique de la figure 1 : le verrou situé sur le MM est piloté par la broche RB3 ; quand cette dernière est au niveau logique haut, ce qui est présent sur le bus à huit bits est reporté sur les sorties du verrou (Q0÷Q7). Naturellement, le programme résident aura au préalable préparé la donnée à transférer à la sortie, avant d'activer le verrou, sans quoi on risquerait de retrouver sur les sorties des données inadaptées à la gestion des LED et des relais.

Dans le programme résident, des registres de mémoire ont été prévus pour écrire les états des sorties et des entrées ; le micro travaille sur ces registres et non directement sur le PORTD (quand il doit les envoyer aux LED et aux relais, il les prend dans la zone de mémoire correspondante et les transfère sur le port, puis il active le verrou correspondant pour les lui faire charger). Avec les broches RB4 et RB5, le PIC pilote les verrous d'entrée pour la lecture des poussoirs.

Les deux signaux de contrôle pour les 74HCT573 sont actifs au niveau logique bas et donc si on met l'un d'eux à zéro, il est possible de lire l'état des poussoirs reliés au verrou correspondant. Les broches RB3, RB4 et RB5 ne sont donc jamais au zéro logique en même temps, sans quoi les deux verrous seraient activés ensemble et les sorties de ceux utilisés pour lire les poussoirs pourraient être endommagés ; on imagine en effet ce qui se passerait si l'un prenait le niveau logique haut et l'autre, en parallèle, le niveau logique bas (hue et dia ne font pas bon ménage) ! Afin d'éviter tout dommage, non seulement les verrous sont activés un à la fois, mais en plus ceux qui sont actuellement inhibés peuvent donner à leurs sorties un troisième état logique dit "open ou high impedance" (haute impédance) ; en effet, les verrous comme le 74CHT573 (mais aussi les vieux 74374 ou les TTL 74244 et 74245) étant conçus pour coupler sur un bus standard de huit bits seulement un nombre de périphérique bien plus élevé, ont la possibilité

Liste des composants

R124 . 4,7 k 1/4 W 5%
R125 . 4,7 k 1/4 W 5%
RN1 ... réseau résistif 8 x 4,7 k
RN2 ... réseau résistif 8 x 4,7 k

C66 ... 100 nF 100 V céramique
C67 ... 100 nF 100 V céramique

P1..... micropoussoir pour ci NO
[...]
P16 ... micropoussoir pour ci NO

LD3 ... LED 5 mm rouge
LD4 ... LED 5 mm verte
LD5 ... LED 5 mm rouge
LD6 ... LED 5 mm verte

U20 ... 74HC573
U21 ... 74HC573

X8..... barrette mâle 16 broches
JP..... 6 connecteurs femelles 2 broches

Divers :

2 supports 2 x 10

Toutes les résistances sans indication contraire sont des 1/4 W.

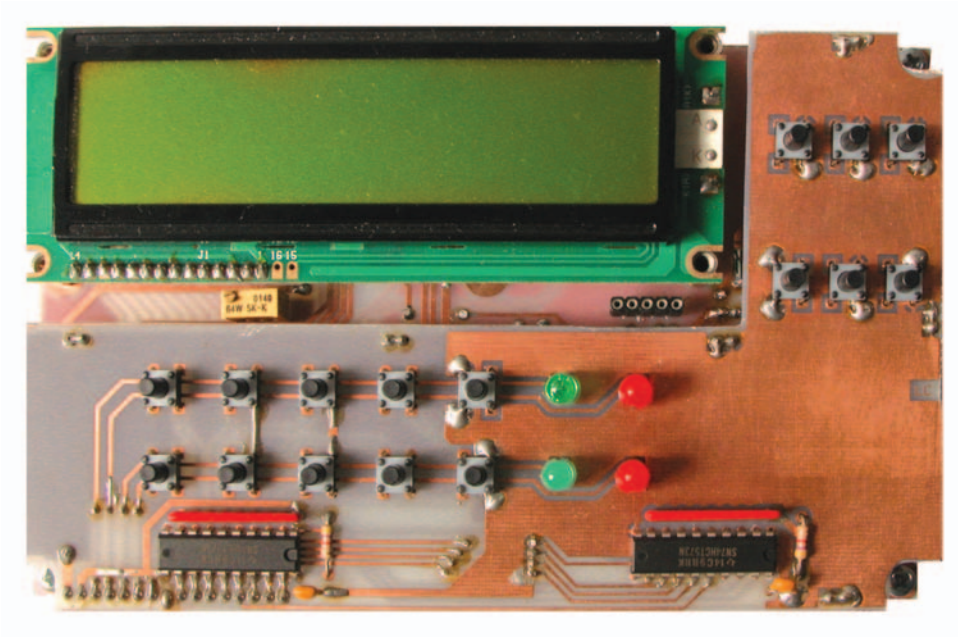


Figure 3 : Photo d'un des prototypes du Module des poussoirs de l'alimentation de laboratoire professionnelle. Cette platine en L est fixée sur la MM à l'aide de X9 et des 6 JP. On le voit, la découpe en L permet de faire apparaître l'afficheur LCD en face avant.

de connecter en parallèle les sorties de l'un avec celles des autres et c'est justement pour cela qu'ils sont construits de manière à pouvoir insensibiliser les lignes de sortie quand un autre dispositif doit imposer son propre état logique. Par conséquent, lorsque les sorties doivent transférer la donnée emmagasinée, il suffit d'activer les puces en fournissant à leurs broches "enable" (habilitation) une condition logique (normalement zéro TTL) ; quand en revanche il faut ignorer les sorties, on fournit le un logique, ce qui met les sorties en haute impédance et simule la déconnexion du circuit. C'est là le principe de base de la logique "three states" (trois états) dont nos verrous relèvent : la possibilité pour les sorties de prendre non seulement les états de la logique binaire (1/0) mais encore un troisième état.

Les commandes

Toutes les fonctions de l'alimentation sont accessibles par poussoirs, selon un menu qui s'affiche sur le LCD ; avant d'aborder ce menu de gestion du système, il est nécessaire de décrire les fonctions remplies par chaque poussoir :

P1 : exécute un défilement à gauche du menu.
P2 : exécute un défilement à droite du menu.
P3 : poussoir OK ; utilisé en diverses occasions :

- pour sauvegarder les valeurs de tension et courant maximales paramétrées pour la branche positive comme pour la négative ; ces valeurs sont mémorisées en EEPROM et lues à la mise sous tension du système (la sauvegarde en EEPROM est soulignée par un éclair des quatre LED du MPSR) ;

- pour le "reset" les conditions de court-circuit (éteint les LED5 et LED3) ;

- pour confirmer les pas de configuration de l'algorithme de gestion des sondes de température NTC ; pratiquement on paramètre les valeurs minimale et maximale en confirmant pas à pas à l'aide de P3 (le logiciel guide au fur et à mesure l'usager au cours des opérations à exécuter pour le réglage de l'instrument).

P4 : active/désactive le rétro-éclairage de l'afficheur LCD.
P5 : actuellement non utilisé.
P6 : active/désactive le "tracking mode", mode dans lequel la valeur de tension de la sortie positive est égale à celle de la tension de sortie négative. La tension de sortie sera ensuite paramétrée en agissant seulement sur P7 et P8 (P21 et P13 seront désactivés).
P7 : augmente la tension positive de sortie de l'alimentation par pas de 50 mV ; s'il est maintenu pressé pendant plus de deux secondes, il produit une augmentation rapide et continue de la tension de sortie jusqu'à ce qu'on le relâche.
P8 : diminue la tension positive de sortie de l'alimentation par pas de 50 mV ; s'il est maintenu pressé pendant plus de deux secondes, il produit la diminution rapide et continue de la tension de sortie jusqu'à ce qu'on le relâche.
P9 : augmente par pas de 5 mA la valeur maximale du courant autorisé sur la branche positive ; si le seuil est dépassé, la sortie du canal positif est déconnectée et il devient nécessaire d'exécuter un "reset" logiciel de la condition d'anomalie. L'anomalie est éliminée en pressant P3 (poussoir OK). Si P9 est pressé et maintenu pendant plus de deux secondes, une augmentation rapide et continue du seuil de courant a lieu tant que ce poussoir n'est pas relâché.
P10 : diminue par pas de 5 mA la valeur maximale du courant autorisé sur la branche positive ; s'il est maintenu pendant plus de deux secondes, une diminution rapide et continue du seuil de courant a lieu tant qu'il n'est pas relâché.
P11 : active/désactive la sortie positive de l'alimentation ; la condition de sortie active est signalée par l'allumage de LED 6.
P12 : augmente la tension négative de sortie de l'alimentation par pas de 50 mV ; s'il est maintenu pressé pendant plus de deux secondes, il produit une augmentation rapide et continue de la tension de sortie jusqu'à ce qu'on le relâche.
P13 : diminue la tension négative de sortie de l'alimentation par pas de 50 mV ; s'il est maintenu pressé pendant plus de deux secondes, il produit une diminution rapide et continue de la tension de sortie jusqu'à ce qu'on le relâche.

P14 : augmente par pas de 5 mA la valeur maximale du courant autorisé sur la branche négative ; si le seuil est dépassé, la sortie du canal négatif est déconnectée et il devient nécessaire d'exécuter un "reset" logiciel de la condition d'anomalie. L'anomalie est éliminée en pressant P3 (poussoir OK). Si P14 est pressé et maintenu pendant plus de deux secondes, une augmentation rapide et continue du seuil de courant a lieu tant que ce poussoir n'est pas relâché.
P15 : diminue par pas de 5 mA la valeur maximale du courant autorisé sur la branche négative ; s'il est maintenu pendant plus de deux secondes, une diminution rapide et continue du seuil de courant a lieu tant qu'il n'est pas relâché.
P16 : active/désactive la sortie négative de l'alimentation ; la condition de sortie active est signalée par l'allumage de LED 4.

Bon, maintenant que nous connaissons la fonction de chaque poussoir, voyons comment on utilise l'alimentation au moyen du menu, avec visualisation sur l'afficheur LCD. Le menu principal (voir figure 6 l'organigramme ou "flow-chart") prévoit six pages plus une de présentation (cette dernière n'apparaît qu'à la mise sous tension). Il est possible de passer d'une page à l'autre avec P1 et P2, comme décrit dans les paragraphes précédents. Voici un aperçu des pages disponibles :

Page 1 : Gestion tension de sortie pour la branche positive comme pour la négative.
Page 2 : Visualisation courant consommé pour la branche positive comme pour la négative.
Page 3 : Paramétrage courant de seuil pour la branche positive comme pour la négative.
Page 4 : Visualisation température atteinte par les transistors de puissance de la branche positive.
Page 5 : Visualisation température atteinte par les transistors de puissance de la branche négative.
Page 6 : Visualisation nombre de tours du ventilateur tachymétrique.

Si des anomalies concernant les sondes de température ou le ventilateur se produisent, les pages 4, 5 et 6 s'affichent.

En dehors du menu principal que nous venons de décrire, il existe un second menu, dit "menu caché" et implémenté pour le réglage des valeurs minimale et maximale de température pour les sondes NTC (ne peut être appelé qu'à la mise sous tension). Pour accéder à ce menu, il faut presser et maintenir P1, P2 et P3 puis allumer l'alimentation. Le menu caché apparaît. Pour régler les sondes NTC il est nécessaire d'avoir au préalable monté le simulateur de température faisant l'objet du prochain chapitre. L'organigramme de ce menu caché est visible figure 7 ; le logiciel aide l'usager à régler l'instrument.

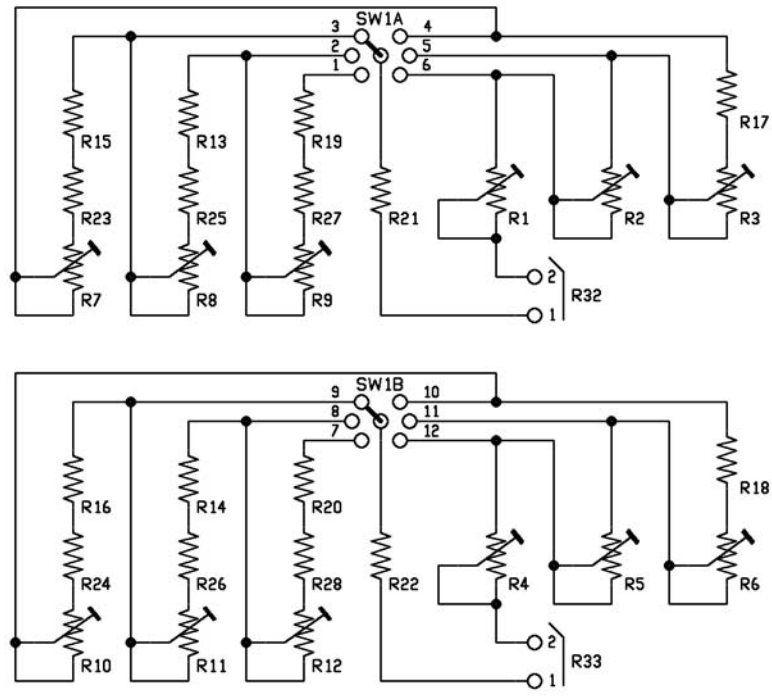


Figure 4 : Schéma électrique du simulateur de températures.

La réalisation pratique

Le circuit imprimé en L du Module des poussoirs MPSR est un double face (réalisez-le à partir des dessins à l'échelle 1:1 de la figure 2b-1 et 2b-2, sans oublier les connexions entre les deux faces). Quand vous l'avez devant vous (voir figures 2a et 3), côté composants, commencez par enfoncer puis souder les deux supports de circuits intégrés et les deux réseaux de résistances RN2 et RN1, puis vérifiez soigneusement vos soudures (ni court-circuit entre pistes ou pastilles ni soudure froide collée). Montez ensuite les rares composants bas (résistances et condensateurs) et terminez par les seize poussoirs et les quatre LED. Côté soudures, montez les six connecteurs femelles à deux broches JP (en pointillés) et la barrette mâle X9. Si vous observez bien les figures 2a et 3 et la liste des composants, vous n'aurez aucune difficulté pour le faire. Attention à l'orientation des composants polarisés (LED méplat vers le bas ; U21 et U22 repère-détrompeurs vers la droite). Les broches 1 (point noir repère-détrompeur) des réseaux résistifs RN2 et RN1 doivent être du même côté que les repère-détrompeurs en U des deux circuits intégrés verrous U21 et U20.

Pour cette platine comme pour toutes les autres, n'oubliez pas de souder des deux côtés du circuit imprimé les queues ou broches des composants qui comportent des pastilles communes côté composants et côté soudures.

Bien sûr la barrette X9 mâle et les six connecteurs à deux broches femelles vont servir à connecter et fixer cette platine en L sur la platine du MM (voir figure 3 de ce numéro et figure 8b page 30 dans le numéro 85 du mois dernier). Tout ce sous ensemble (MM + MPSR) sera fixé derrière la face avant (dûment percée pour le passage des axes des seize poussoirs, des quatre LED et de la fenêtre de l'afficheur LCD) ; un petit trou sera percé en face avant au-dessous du LCD pour pouvoir agir sur la petite vis du trimmer de réglage du contraste. Ce sous ensemble sera relié au MD par nappe (voir ci-après).

Le simulateur NTC

Pour le réglage, nous avons imaginé simuler le comportement des thermistances NTC à l'aide d'un simulateur



Figure 5 : Le simulateur (ici monté dans son boîtier plastique) sert à régler les capteurs de température NTC qui ont pour rôle de protéger en température les jonctions des six darlingtons de puissance montés sur le gros dissipateur. Ce simulateur comporte un certain nombre de résistances de précision lesquelles, insérées à l'aide du commutateur rotatif à deux galettes et six positions, fournissent des valeurs résistives correspondantes à celles prises par les NTC soumises à des températures déterminées.

Liste des composants

R1 ..... trimmer multitour 100 ohms
R2 ..... trimmer multitour 100 ohms
R3 ..... trimmer multitour 100 ohms
R4 ..... trimmer multitour 100 ohms
R5 ..... trimmer multitour 100 ohms
R6 ..... trimmer multitour 100 ohms
R7 ..... trimmer multitour 100 ohms
R8 ..... trimmer multitour 100 ohms
R9 ..... trimmer multitour 100 ohms
R10 ... trimmer multitour 100 ohms
R11 ... trimmer multitour 100 ohms
R12 ... trimmer multitour 100 ohms
R13 ... 27 k 1%
R14 ... 27 k 1%
R15 ... 2,2 k 1%
R16 ... 2,2 k 1%
R17 ... 470 1%
R18 ... 470 1%
R19 ... 62 k 1%
R20 ... 62 k 1%
R21 ... 110 1%
R22 ... 110 1%
R23 ... 680 1%
R24 ... 680 1%
R25 ... 1,2 k 1%
R26 ... 1,2 k 1%
R27 ... 2,4 k 1%
R28 ... 2,4 k 1%
R32 ... NTC 10 k à 25 °C
R33 ... NTC 10 k à 25 °C
SW1 .. commutateur rotatif deux circuits six positions

Divers :

2 borniers 2 pôles
1 circuit imprimé à pastilles ou à bandes

constitué de résistances sélectionnables au moyen d'un commutateur rotatif. Nous pouvons ainsi obtenir par commutation les conditions correspondant à six valeurs de températures possibles : –20, 0, 50, 100, 150 et 170 °C. Les températures –20 °C et +170 °C simulent les conditions de sonde respectivement déconnectée et en court-circuit.

Le schéma électrique visible figure 4, montre que l'appareil est formé de deux canaux symétriques, ce qui permet de l'utiliser pour régler les deux capteurs de température. Le commutateur est un modèle rotatif à deux voies (deux "galettes") et six positions : quand on tourne l'axe (muni d'un bouton, comme le montre la figure 5), la valeur de résistance insérée entre les bornes de l'appareil change (les deux canaux ayant chaque fois les mêmes valeurs ohmiques).

Soudez les résistances et les trimmers multitours sur un circuit imprimé à pastilles ou à bandes et reliez ce dernier au double commutateur rotatif, puis montez l'ensemble dans un boîtier plastique adéquat dont la figure 5 vous donne une idée. Le commutateur est fixé directement sur la face avant par le canon fileté de son axe et par ses écrous plats. Il est ensuite assorti d'un bouton à index. Sur une étiquette on inscrira les six positions. L'un des côtés sera percé de deux trous pour le passage des deux paires allant à l'ohm-mètre d'abord (lire ci-après) puis aux borniers des NTC qu'il s'agit de simuler.

Le réglage du simulateur doit commencer en mettant le commutateur en position 6 et en reliant les fils de sortie à un multimètre réglé en position ohmmètre ; on agit alors sur le trimmer multitour R1 pour obtenir une résistance de 119 ohms (ce qui correspond en NTC à une température de +170 °C).

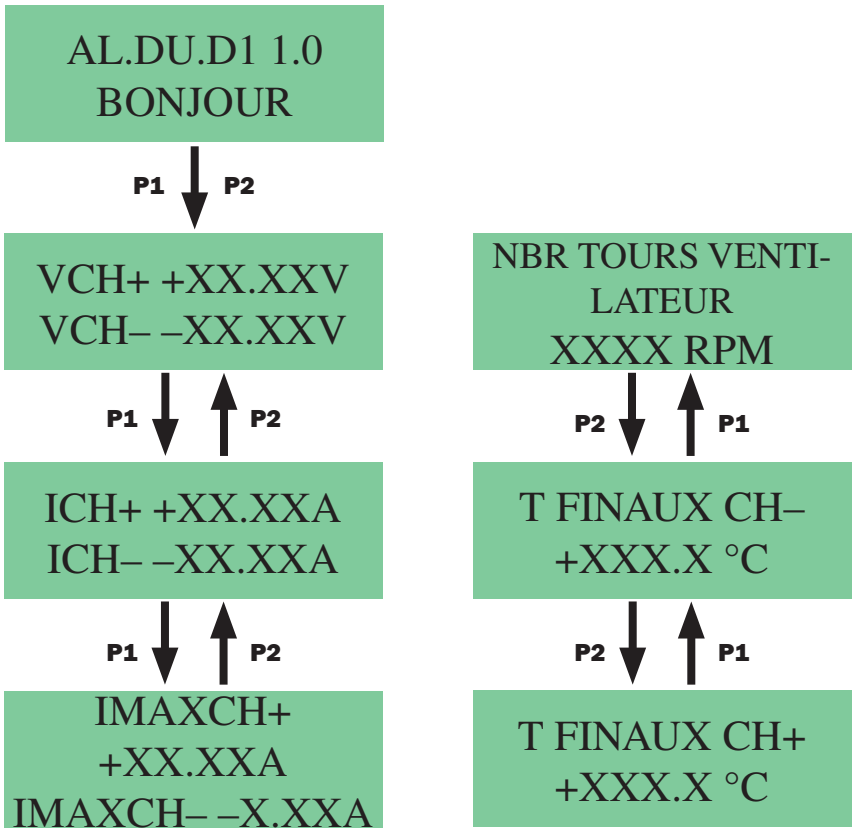
Tournez ensuite l'axe vers la position 5 et agissez sur le trimmer R2 pour obtenir 185 ohms (ce qui correspond à +150 °C). Puis mettez le bouton en position 4 et agissez sur R3 pour obtenir 680 ohms (ce qui correspond à +100 °C).

Mettez le bouton en position 3 et agissez sur R7 pour obtenir 360 ohms (ce qui correspond à +50 °C). Puis mettez le bouton en position 2 et agissez sur R8 pour obtenir 32,65 ohms (ce qui correspond à 0 °C). Enfin, mettez le bouton en position 1 et agissez sur R9 pour obtenir 97,07 ohms (ce qui correspond à –20 °C ou sonde coupée ou débranchée).

Ceci vaut bien sûr pour un canal ; pour le second canal, il faut répéter ces opérations identiquement, les positions du bouton étant les mêmes et seuls les trimmers intéressés changeant de position. Voir le tableau suivant :

Position bouton	Trimmer can 1/2	Valeur ohmique mesurée	Température équivalente NTC
6	R1/R4	119 ohms	+170 °C
5	R2/R5	185 ohms	+150 °C
4	R3/R6	680 ohms	+100 °C
3	R7/R10	360 ohms	+50 °C
2	R8/R11	32,65 ohms	0 °C
1	R9/R12	97,07 ohms	–20 °C

Figure 6 : Organigramme du menu principal. On va en avant ou en arrière avec P1 ou P2.

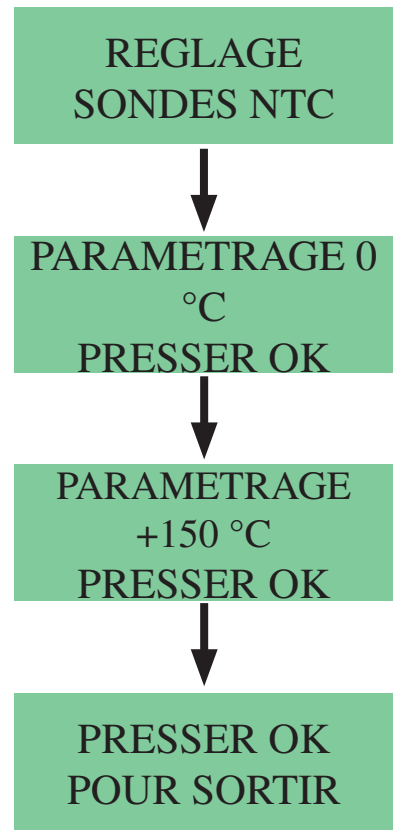


L'assemblage de l'alimentation

Il va maintenant falloir installer tout ce beau monde (voir les numéros 84 et 85 d'ELM) dans un boîtier adéquat, mais avant tout il convient d'assembler et d'interconnecter les modules réalisés. Sur le MP ("carte-mère"), insérez le MD, le MA et le MF. Les borniers du MF doivent être reliés au pont redresseur et au secondaire du transformateur d'alimentation (en fonction du schéma publié dans le numéro 84 d'ELM) ; à ce propos, notez que les points COM correspondent à la prise centrale du transformateur torique. Connectez le transformateur 2 x 9 V à X1. Avec une nappe à 26 fils, reliez le MD au MM (celui qui est accolé au MPSR) puis dotez les câbles arrivant des six darlingtons de puissance (montés sur la platine appliquée sur le dissipateur) de six connecteurs adaptés à ceux que vous avez montés sur le MP et branchez-les sur ce dernier. Pour cette liaison, les NPN peuvent être branchés sans avoir à respecter un ordre particulier, à condition qu'ils aillent bien aux connecteurs Q10, Q11, Q12 ; même remarque pour les PNP, à relier aux connecteurs Q17, Q18, Q19. Les points Vout+, Vout–, Sense+, Sense– de X7 sont à relier au MM ; aux borniers de sortie reliez en revanche Sense+ Load et Sense– Load qui arrivent du MM et la masse (à prendre directement sur le transformateur au point COM). Il ne manque que les deux NTC, à fixer au dos du dissipateur : une du côté où s'appuient les darlingtons NPN et l'autre où sont les PNP ; elles se connectent électriquement aux borniers du MM.

L'alimentation est alors prête à fonctionner : après un dernier contrôle de l'assemblage et des interconnexions, reliez le primaire du transformateur torique et celui du 2 x 9 V au cordon d'alimentation et branchez ce dernier sur le secteur 230 V. L'afficheur LCD doit s'allumer et, au bout d'un instant, afficher le message de salut. Avant de presser les poussoirs, avec un multimètre réglé pour les tensions continues (1 ou 2 V fond d'échelle), pointe négative noire à la masse et pointe positive rouge sur la broche 3 de U16, réglez le trimmer R59 pour obtenir exactement 0 V ; déplacez alors la pointe positive vers la broche 3 de U17 et réglez R68 pour 0 V également. Si vous souhaitez un réglage plus précis, réglez le multimètre numérique sur le calibre sur lequel il est le plus sensible (par exemple 200 mV). Si, pendant l'utilisation avec une charge égale sur les deux branches, vous notez une différence entre les tensions de sortie positive et négative, retouchez R67 et R71 jusqu'à ce que ces deux tensions soient égales en valeur absolue.

Figure 7 : Organigramme du menu caché. Ce menu sert à régler le champ de variation de température des thermistances NTC. On y accède lors de la mise sous tension de l'alimentation (on maintient pressés P1, P2 et P3 et on allume l'appareil). On se déplace d'un champ à l'autre à l'aide de P1. Quand le paramétrage est fait, on confirme en pressant P3 (OK) et cela nous ramène au menu principal.



Conclusion

Vous voilà maintenant en possession d'une véritable alimentation de laboratoire pour laquelle vous auriez dépensé dix ou vingt fois plus en choisissant un modèle identique (à supposer que vous le trouviez...) du commerce.

Comment construire ce montage ?

Tout le matériel nécessaire pour construire cette alimentation de laboratoire professionnelle ETALI est disponible chez certains de nos annonceurs. Voir les publicités dans la revue.

Les typons des circuits imprimés et les programmes disponibles sont téléchargeables à l'adresse suivante :

http://www.electronique-magazine.com/circuitrevue/086.zip. ◆

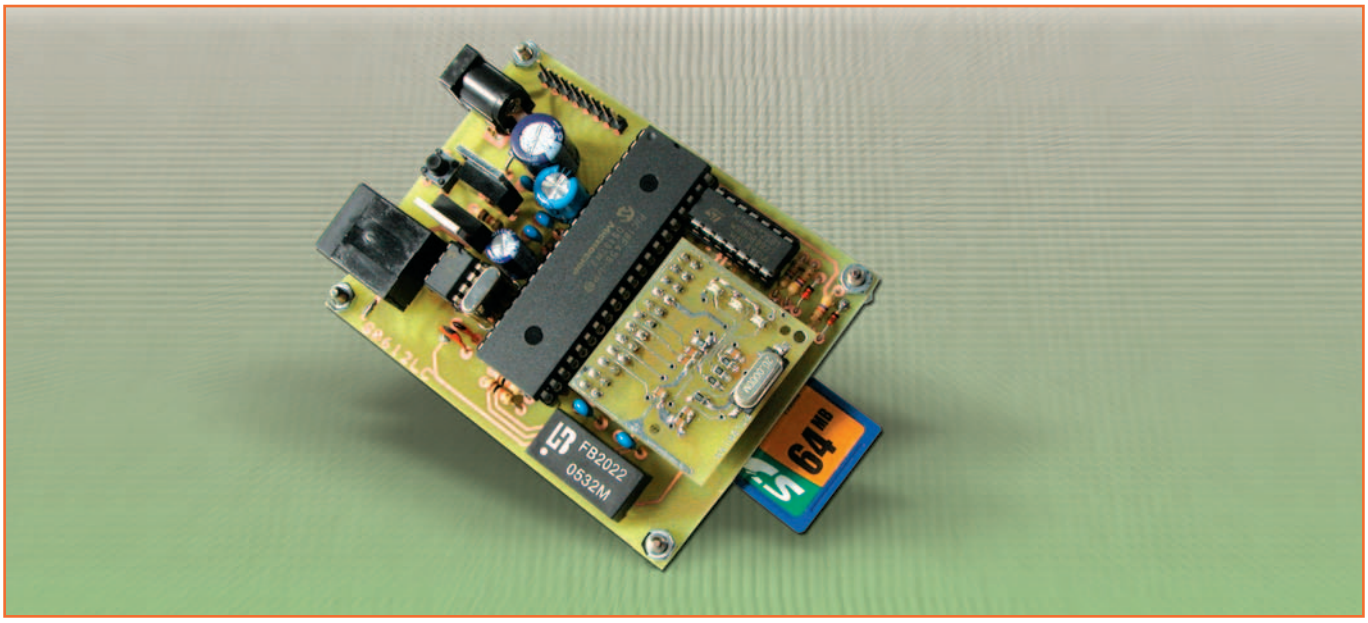
ET616-ET612-1

Une interface Client FTP*

avec PIC et SD-Card

Première partie : analyse théorique et réalisation

Nous utilisons un microcontrôleur Microchip pour publier via FTP* des données sur Internet. Pour la première fois nous adoptons une interface réseau entièrement construite par nous à partir d'une des plus populaires puces Ethernet : la RTL8019 de Realtek. Avec un programme résident spécifique, ce circuit peut également servir de serveur Web.



La publication de données sur Internet est, pour nos lecteurs, un argument du plus haut intérêt ; c'est pourquoi, après la présentation du serveur Web avec SitePlayer ou celui avec GPRS, nous le reprenons ici pour vous proposer un montage intégrant le "Stack" TCP/IP de Microchip [pour "stack", mon dictionnaire donne "pile (mémoire spéciale)", mais nous allons voir au cours de l'article de quoi il s'agit au juste]. Comme première application, nous l'utilisons en Client pour gérer (au moyen du programme résidant dans son PIC) la connexion avec un serveur FTP* et l'envoi de données ("upload") ; dans les prochains numéros d'ELM vous verrez comment avec ce même circuit on peut réaliser un véritable serveur Web.

Bien que les cahiers des charges des clients soient multiples, nous décrirons ce qui intéresse la plupart des passionnés qui nous sont fidèles (et qui nous le font savoir par courriel) : la publication par Internet des données acquises par des capteurs reliés au PIC. Beaucoup d'entre vous ont suivi avec curiosité l'apparition sur le marché de plusieurs dispositifs en mesure de transformer un microcontrôleur en un petit serveur Web : on pense bien sûr tout de suite au fameux SitePlayer (lire et relire notre Cours dédié dans les anciens numéros d'ELM) et ensuite peut-être au IPic ou au PICDem.Net de Microchip. Il s'agit là d'appareils de bon niveau, mais qui comportent cependant deux limites : faible capacité de mémorisation et difficulté de visualisation pour les usagers d'un réseau plus

étendu qu'un LAN** (WAN ***ou Internet). Pour le premier grief, la solution paraît fort simple (à tel point que nous nous préparons à publier un article vous proposant un serveur Web à mémoire "flash" de grande capacité) ; pour le second par contre c'est plus complexe, car entre notre système et le reste des usagers du réseau un troisième obstacle se dresse : le "Service-Provider" (fournisseur d'accès Internet).

La manière la plus simple de rendre notre PIC-Server visible au reste du monde est de lui attribuer une adresse IP fixe ; il est d'aiilleurs fort dommage que cette possibilité ne soit pas actuellement donnée à tous les usagers d'Internet et que, quand elle est obtenue, elle implique un coût que les passionnés peuvent difficilement assumer dans le domaine des loisirs. L'utilisation, par exemple, d'un module GPRS comme le fameux GR47 de Sony-Ericsson (là encore, lire et relire dans les anciens numéros le Cours dédié) comporte l'inconvénient de requérir l'attribution d'une IP par le fournisseur d'accès, or cette IP n'est pas permanente : elle change à chaque nouvelle connexion ! D'autre part, l'attribution de l'adresse à travers un serveur DHCP (donc de manière dynamique) est une pratique courante de nos chers (!) FAI. Sans parler des réseaux de téléphonie mobile, réseaux privés où bien souvent, pour des motifs de sécurité, l'utilisateur reste hors d'atteinte des usagers d'Internet. Le contrôle du trafic, dans ce cas, est effectué de manière restrictive et il n'est pas possible de connaître l'adresse réelle avec laquelle nous entrons sur le réseau. En fait, le nœud utilisateur est un client pur et il ne peut satisfaire à aucune demande provenant de l'extérieur, puisque cette demande est bloquée par le FAI. Une telle procédure, visant, répétons-le, notre sécurité, n'en annule pas moins nos efforts pour réaliser un programme résident publiant des données accessibles à tous.

***FTP** = File Transfer Protocol, protocole de transfert de fichier.
****LAN** = Local Area Network, réseau local → un LAN.
*****WAN** = Wide Area Network, réseau étendu.

Notre réalisation

Pour résoudre ces problèmes, nous avons pris un chemin de traverse : puisque se faire attribuer une IP° fixe est si difficile, il est beaucoup plus simple d'insérer notre site dans un des nombreux espaces gérés par un FAI°° et de le mettre à jour à distance en transférant les fichiers correspondants à travers le protocole FTP. Ainsi tous les usagers pourront accéder au site directement à partir de leur navigateur (Internet Explorer ou Netscape, etc.), simplement en tapant l'adresse attribuée. Nous pourrons de ce fait connecter notre circuit au réseau domestique de façon à pouvoir utiliser le routeur ADSL°°° pour le transfert, ou alors expérimenter le transfert des données directement sur notre ordinateur à travers la carte réseau et un serveur FTP gratuit.

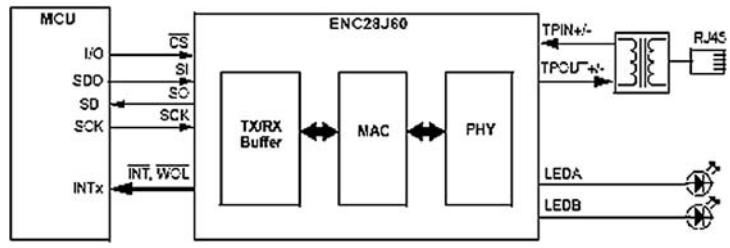


Figure 1 : Organigramme du module d'interface Ethernet utilisé pour ce montage Client FTP.

°**IP** = Internet Protocol (abréviation de adresse IP, c'est pourquoi j'écris "une").
°°**FAI** = Fournisseur d'Accès Internet ou "Provider".
°°°**ADSL** = Asymetric Digital Subscriber Line (accès à Internet à moyen et haut débit par ligne téléphonique).

Le circuit

Pour la réalisation, nous nous sommes inspirés d'une platine d'expérimentation ("demoboard") commercialisée par Microchip sous le nom de PICDEM.net, opportunément modifiée par ajout d'un "slot" (emplacement) pour SD-Card et élimination de tout ce qui ne nous est pas nécessaire. Les lignes ainsi libérées ont été regroupées en un bus ("strip") pouvant être utilisé pour la connexion à des sondes de différents types : le microcontrôleur de la platine d'expérimentation peut ainsi acquérir les valeurs relevées par des capteurs de température, d'humidité ... ou par des dispositifs météorologiques, etc.

Pour lire la température, on s'est servi d'une DS18B20 (déjà rencontrée dans le montage SD-Thermo) : il s'agit d'une excellente sonde thermique gérable par le protocole "one-wire" (monofil) et pouvant mesurer des températures entre –55 et +125 °C (avec une précision de ±0,5 °C entre –10 et +85 °C). Sa résolution peut être configurée jusqu'à 12 bits, avec des délais de réponse inférieurs à la seconde (exactement 750 ms).

En ce qui concerne l'interface réseau, nous avons maintenu fixe la configuration du PIC de manière à devoir intervenir le moins possible sur la source du "stack" TCP/IP` Microchip. La puce en question (RTL8019) est produite par

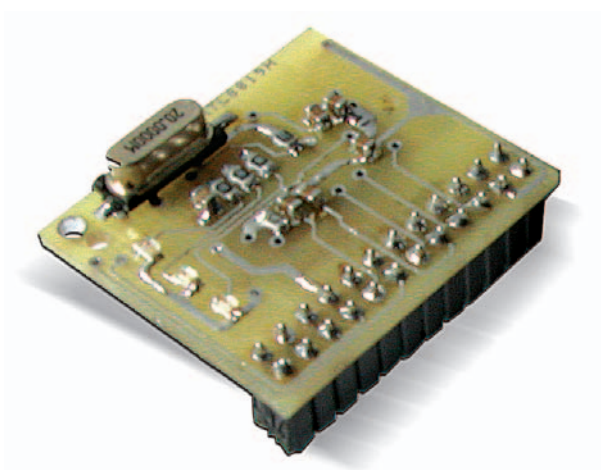


Figure 2 : Module Ethernet où sont montées les LED de signalisation CMS.

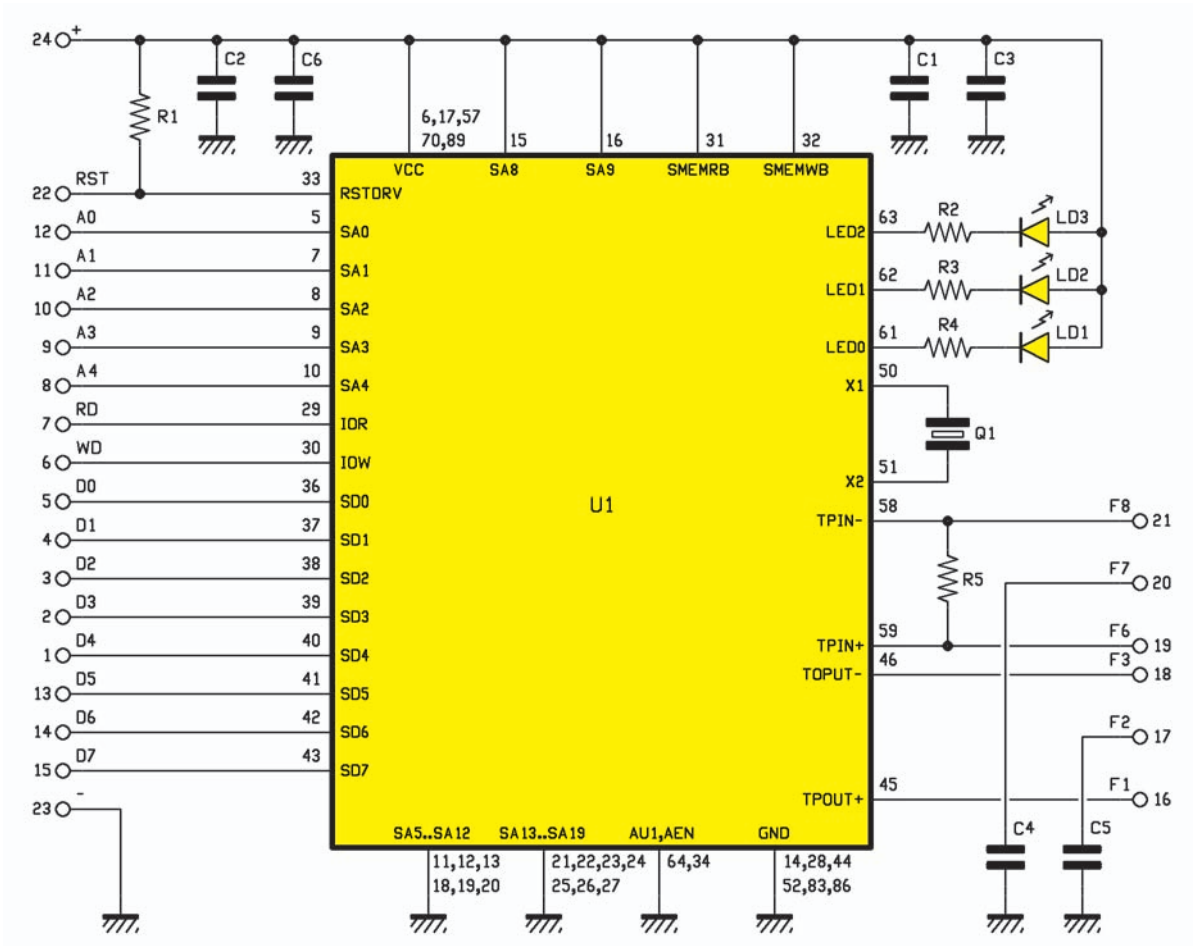


Figure 3 : Schéma électrique du module Ethernet RTL8019 ET616.

la société taiwanaise Realtek ; elle a été beaucoup montée sur les cartes réseau de nos ordinateurs de bureau.

Il s'agit d'une version à 10 Mbps qui n'est plus utilisée sur les cartes de nouvelle génération mais qui reste une référence à cause de sa simplicité d'emploi et de sa stabilité dans le domaine des serveurs Web embarqués ; ça peut paraître étrange, mais elle va très bien à notre application et elle y fonctionne comme une 100 Mbps : en effet, nous avons à publier des données sur Internet et donc passer par une ligne téléphonique, or même avec l'ADSL le plus rapide nous n'arriverons jamais à une vitesse de communication de 10 Mbps.

Pour dialoguer avec une ligne ne dépassant pas 6 Mbps (c'est l'ADSL le plus rapide ... à condition de ne pas habiter trop loin d'une mégapole !) les 10 Méga de l'adaptateur réseau suffisent amplement et ont même une sérieuse avance. Si le serveur Web s'interface avec un LAN avant d'atteindre le modem`` ou le routeur, la limitation de la puce Realtek n'est pas un problème et c'est pourquoi la norme est à 10 Mbps ou, en double vitesse, à 10/100 Mbps. Vous voyez que l'efficacité de notre montage ne pâtira pas le moins du monde du choix de cette puce, certes un peu déclassée en comparaison de ses concurrentes récentes bien plus aguichantes (en termes de connectivité Ethernet pour micro).

En tout cas, nos choix et solution (manifestés dans cet article) feront une fois de plus pâlir les appareils du commerce (en particulier pour des raisons de coût, mais pas seulement). La plupart d'entre eux ont des problèmes de mémoire car ils utilisent des EEPROM``` de quelques ko seulement, alors que nous utiliserons une carte SD de 64 Mo !

Pour la conservation des données nous utiliserons en revanche cette EEPROM : elle permet à l'usager de paramétrer de manière extrêmement simple le fonctionnement de l'appareil au moyen d'un programme que nous vous fournissons. Ce dernier produit des fichiers binaires à insérer dans l'EEPROM afin d'établir l'adresse IP, le masque de sous réseau, la "gateway" (passerelle), le nom de l'usager, etc.

La puce RTL8019 a été montée sur une petite platine CMS (composants à montage en surface, en anglais SMD) double face, les composants se montant sur les deux faces, comme le montrent les **figures 4 et 5**. Les lignes d'interface aboutissent à un connecteur barrette femelle à 2 x 12 = 24 trous, ce qui permet ensuite d'intégrer facilement le module ainsi constitué dans diverses platines.

Ce module est le ET616. La puce RTL8019 a été conçue pour un interfaçage compatible avec le standard IEEE```` 802.3 (10Base5, 10Base2, 10BaseT) vers un bus ISA'. La reproductibilité facile des mécanismes de communication implémentés pour ce bus permet une intégration plus directe avec le PIC. Voyons concrètement la configuration de base mise en œuvre, en nous référant au schéma d'interfaçage Realtek RTL8019AS et à la **figure 1**. Le système est relié au PIC à travers une séquence de seize lignes dont le **Tableau 1** synthétise les fonctions.

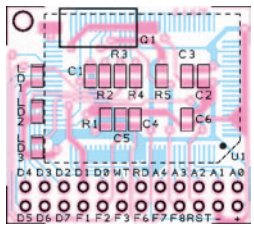


Figure 4a : Schéma d'implantation des composants du module Ethernet RTL8019 ET616.

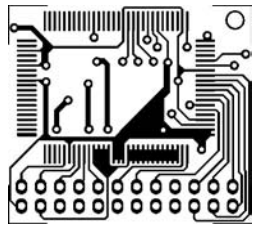


Figure 4b-1 : Dessin, à l'échelle 1, du circuit imprimé double face pour CMS du module Ethernet ET616, côté puce RTL8019 et connecteur.

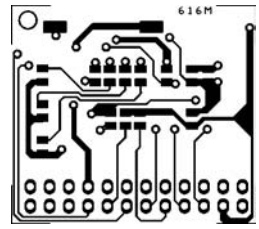


Figure 4b-2 : Dessin, à l'échelle 1, du circuit imprimé double face pour CMS du module Ethernet RTL8019 ET616, côté des autres composants.

Le bus d'adressage est en réalité constitué de 20 broches, mais comme elles ne sont pas nécessaires, nous relions à la masse (GND) les autres lignes, ce qui force au zéro logique les bits correspondants. Les paires TPIN+/- et TPOUT+/- représentent respectivement les lignes différentielles d'entrée et de sortie pour les séquences de données en codage Manchester. Elles sont reliées directement au filtre Ethernet et donc à l'habituelle RJ45. Par commodité, le filtre a été placé sur la platine du PIC et la puce RTL (nous l'avons dit) sur la petite platine CMS en compagnie de quelques composants passifs. La connexion entre les deux platines se fait au moyen d'un connecteur double à 24 trous (mais cela aussi nous l'avons dit).

Comme le montre la figure 2, les trois LED de signalisation sont montées sur cette platine CMS : elles indiquent l'état de la puce. Selon la valeur des bits de configuration, trois situations différentes peuvent avoir lieu : si le bit LEDS0 du registre CONFIG3 est égal à 0, LED0 sert de signalisation de collisions, sinon elle sert d'indicateur de présence du signal de "link" (liaison).

Quant à LED1, si le bit LEDS1 du registre CONFIG3 est égal à 0, elle sert d'indicateur de réception des données (LED_RX) et sinon d'indicateur de porteuse (LED_CRS) ; dans ce dernier cas elle clignote pour chaque paquet en transit (entrant ou sortant). Enfin, si le bit LEDS1 du registre CONFIG3 est égal à 0, LED2 sert d'indicateur d'émission des données (LED_TX) et sinon elle signale l'entrée en "standby" (attente) du "buffer" (tampon) situé dans la SRAM'' pour obtenir le mode d'économie d'énergie. Le **Tableau 2** récapitule tout cela.

Par ailleurs les **figures 2** à **5** et la **6** vous présentent ce module du schéma électrique au brochage des E / S en passant par la réalisation pratique. Le brochage met en évidence les broches servant à la communication avec le microcontrôleur et la séquence des LED de signalisation. La série F1-F8 +/- est en revanche nécessaire pour la connexion avec le filtre de ligne (coupleur magnétique Ethernet) et l'alimentation.

En ce qui concerne le micro, une fois de plus véritable cœur de tout le système, notre choix s'est porté sur un PIC18F458 ; en le faisant travailler à 20 MHz il est en mesure de gérer toutes les fonctions que nous lui avons confiées. Son interfaçage avec la SD-Card nous a imposé une certaine configuration particulière, permettant d'adapter les niveaux logiques TTL'' avec lesquels le micro fonctionne et ceux de la SD (3,3 V) ; il s'agit d'un schéma éprouvé dans des montages où le micro ne pouvait fonctionner sous des tensions inférieures à 5 V.

Les lignes du PIC utilisées pour le dialogue avec la carte, sont toutes "open-drain" et ont chacune une diode schottky et une résistance de tirage ("pull-up") ; ainsi, quand l'une est à l'état ouvert (état logique haut), le contact correspondant de la SD est maintenu à une tension d'environ 3,3 V (car la schottky est bloqué) et dès que sur la broche du micro le zéro logique est présent, la diode conduit et la ligne de la carte se trouve à un potentiel égal à la chute de tension (environ 0,3 V). Quant à la connexion dans la direction inverse, soit de carte à PIC, les choses sont légèrement différentes : afin de rendre la traduction des niveaux logiques simple tout en restant efficace et précise, nous avons utilisé un "buffer/line driver" (pilote de ligne/tampon) en technologie HCT ; il s'agit d'un circuit intégré très économique mais

Liste des composants CMS

R1 ..... 20 k
R2 ..... 1 k
R3 ..... 1 k
R4 ..... 1 k
R5 220

C1..... 100 nF multicouche
C2..... 100 nF multicouche
C3..... 100 nF multicouche
C4..... 100 nF multicouche
C5..... 100 µF 35 V électrolytique
C6..... 100 nF multicouche

LD1 ... LED rouge
LD2 ... LED verte
LD3 ... LED jaune

U1..... RTL8019

Q1 ..... quartz 20 MHz

Divers :

2 barrettes femelles à 12 trous

Sauf spécification contraire, toutes les résistances sont des 1/4 W à 5 %.

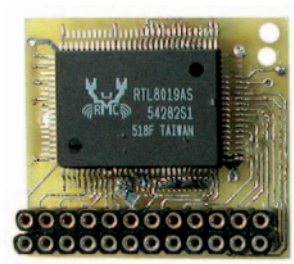


Figure 5a : Photo d'un des prototypes du module Ethernet ET616, côté puce RTL8019 et connecteur.

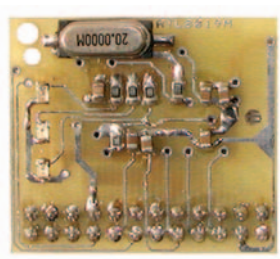


Figure 5b : Photo d'un des prototypes du module Ethernet ET616, côté des autres composants.

Tableau 1.

BROCHE RTL	BROCHE PIC	DESCRIPTION
SA0:SA4	RB0:RB4	Bus d'adressage. Permet l'accès aux divers registres de la puce.
IORB	RE0	Ligne pour commandes de lecture.
IOWB	RE1	Ligne pour commandes d'écriture.
RSTDRV	RE2	Quand elle est mise au niveau logique haut pendant au moins 800 ns, permet d'effectuer le reset matériel de la puce.
SD0:SD7	RD0:RD7	Bus données: permet l'échange de données bidirectionnelles entre PIC et RTL.

Tableau 2.

LED0	Si le bit LEDS0 du registre CONFIG3 est égal à 0, la LED signale une collision, sinon elle joue le rôle d'indicateur de présence du signal de liaison.
LED1	Si le bit LEDS1 du registre CONFIG3 est égal à 0, la LED signale une réception de données (LED_RX), sinon elle joue le rôle d'indicateur de porteuse (LED_CRS); dans ce dernier cas, elle clignote pour chaque paquet en transit, aussi bien entrant que sortant.
LED2	Si le bit LEDS1 du registre CONFIG3 est égal à 0, la LED opère comme indicateur d'émission des données (LED_TX), sinon elle signale l'entrée en "standby" du "buffer" situé dans la SRAM pour économiser l'énergie.

qui permet de résoudre le problème de manière élégante. Nous en avons utilisé la version la plus courante, le 74HCT125, commandant quatre lignes. Son rôle est de transformer en TTL (0/5 V) les niveaux 0/3,3 V produits par la ligne DATI USCITA de la SD-Card. Pour habiliter les sorties de U4, on utilise les quatre broches OE1 à OE4 (Output Enable); en fait le signal d'entrée est présenté en sortie quand la ligne OE est en condition LOW. Comme nous voulons que le passage E vers S se fasse sans temps mort, nous avons relié les broches OE correspondant à la section utilisée directement à GND ("ground", masse). Les lignes d'entrée sont pleinement compatibles avec les signaux provenant des cartes SD, car les circuits intégrés basés sur la logique ACT/HCT acceptent en entrées des niveaux inférieurs aux TTL (par exemple 0/3V) et présentent en sortie des niveaux 0/5 V; quand ils sont alimentés en 5 V, ils "voient" un niveau de 3 V comme si c'était un niveau logique TTL 5 V normal et fournisent donc en sortie 5 V, tension idéale pour commander directement les lignes d'entrée du PIC18F458. Pour communiquer avec l'EEPROM, nous utilisons le module MSSP du PIC et donc la broche RC3 pour le signal d'horloge et la RC4 pour la ligne des données. A propos de l'EEPROM, bien qu'elle ait plus de capacité qu'il ne nous en faut, nous avons inséré dans le circuit une 24LC256. Voilà bien de la redondance et de l'abondance, foi d'écureuil!

En effet, les données de configuration occupent moins de 100 octets; toutefois, si nous considérons qu'il s'agit là d'un schéma de base appelé, une fois encore, à d'ultérieurs remaniements, extensions et plus si affinités, cette énorme "marge" ou "réserve" pourrait bien se révéler utile un de ces jours... Dans le serveur Web que nous publierons, vous verrez combien il est utile de gérer un nombre élevé de fichiers (et notre EEPROM servira alors de table de localisation pour conserver les pointeurs aux secteurs de la SD relatifs à chaque page Web).

`**TCP/IP** = Transmission Control Protocol/Internet Protocol.

``**MODEM** = MOdulateur/DEModulateur (appareil permettant d'accéder en émission et en réception à une ligne téléphonique ou à un réseau sans fil).

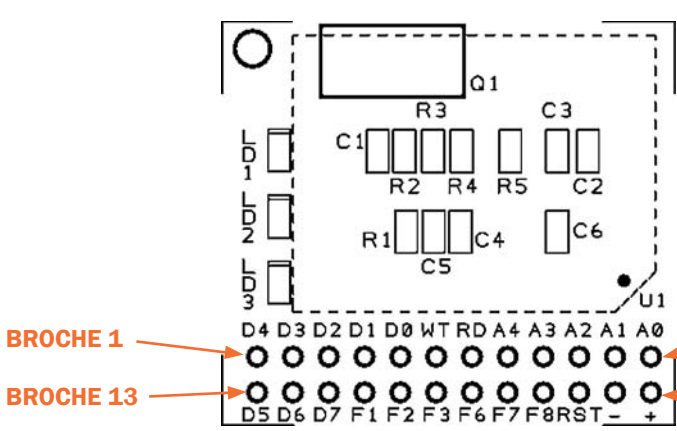
```**EEPROM** = Electrically Erasable Programmable Read Only Memory, mémoire morte modifiable électriquement.

````**IEEE** = Institute of Electrical and Electronics Engineers (institut américain ayant créé ce nouveau standard connectique).

'ISA = Industrial Standard Architecture.

''SRAM = Static Random Access Memory (type de mémoire vive dont les temps d'accès ont représenté

Figure 6: Brochage du module Ethernet RTL8019 ET616.



BROCHE 1: Data 4
BROCHE 2: Data 3
BROCHE 3: Data 2
BROCHE 4: Data 1
BROCHE 5: Data 0
BROCHE 6: WT
BROCHE 7: RD
BROCHE 8: Adresse 4
BROCHE 9: Adresse 3

BROCHE 10: Adresse 2
BROCHE 11: Adresse 1
BROCHE 12: Adresse 0
BROCHE 13: Data 5
BROCHE 14: Data 6
BROCHE 15: Data 7
BROCHE 16: F1
BROCHE 17: F2
BROCHE 18: F3
BROCHE 19: F6
BROCHE 20: F7
BROCHE 21: F8
BROCHE 22: Reset
BROCHE 23: GND
BROCHE 24: Vcc

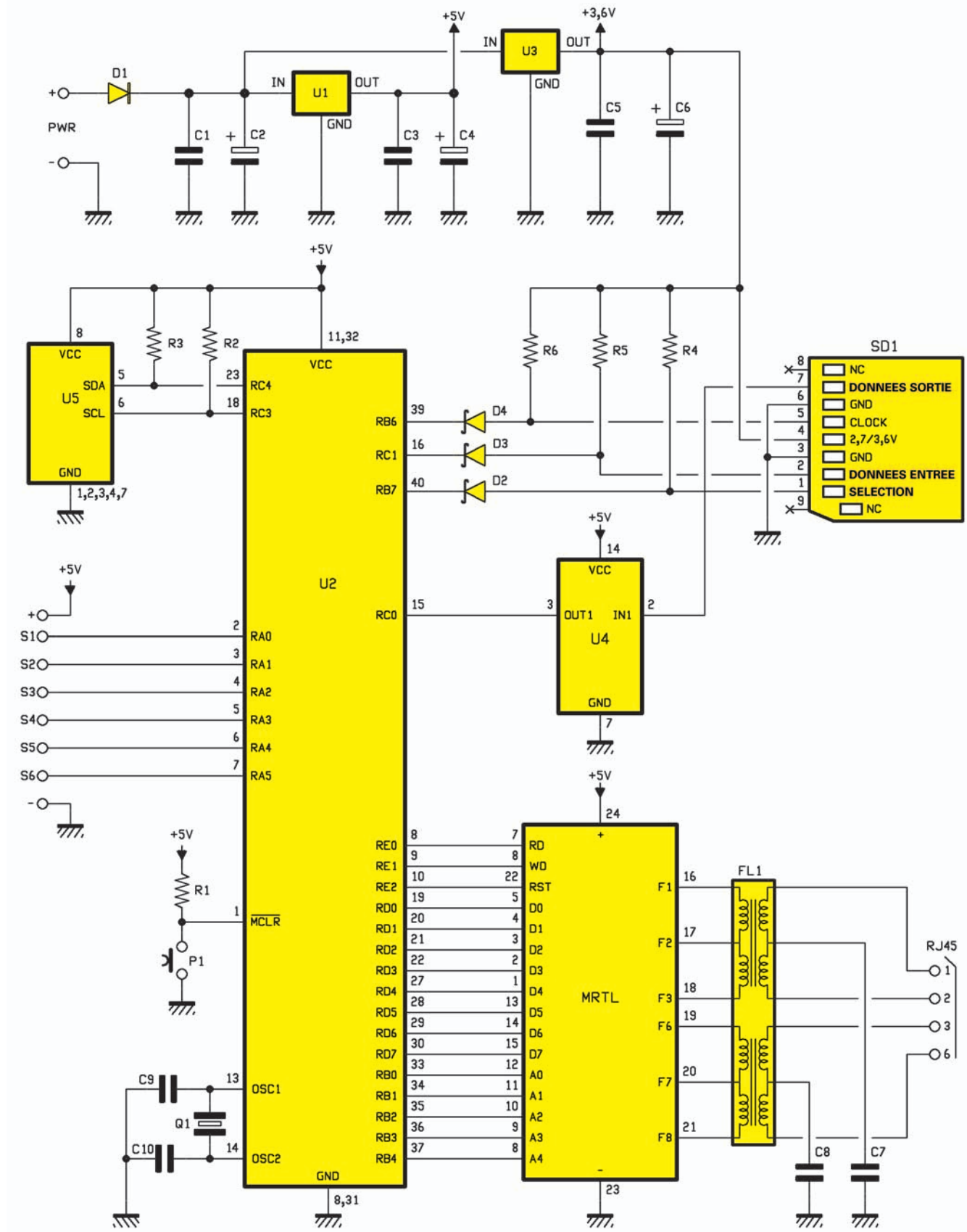


Figure 7 : Schéma électrique de l'interface réseau-platine d'expérimentation ET612.

autrefois pour l'informatique une avancée décisive).

'''TTL = Transistor Transistor Logic, logique à transistor multiémetteur.

Le stack TCP/IP

Après avoir jeté un coup d'œil au matériel, commençons à analyser le programme résident chargé de l'animer. Laissons pour le moment de côté la théorie (nous y reviendrons à propos de ce serveur Web à PIC que nous vous avons annoncé...) au

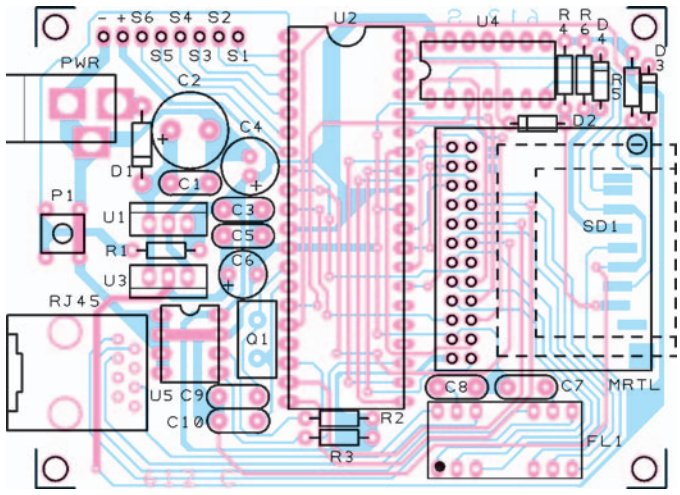


Figure 8a : Schéma d'implantation des composants de l'interface réseau-platine d'expérimentation ET612.

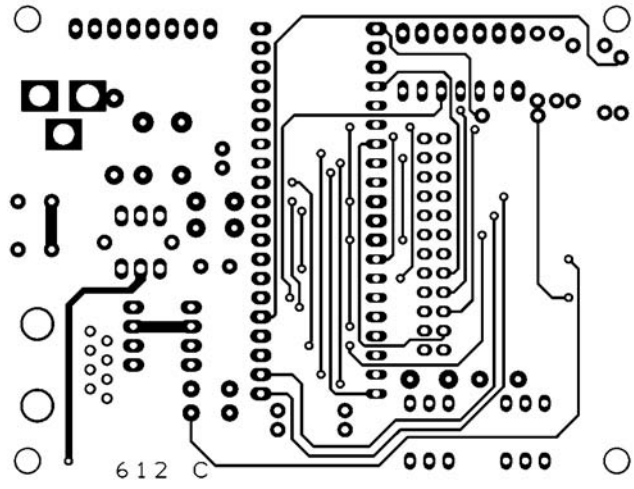


Figure 8b-1 : Dessin, à l'échelle 1, du circuit imprimé double face de l'interface réseau-platine d'expérimentation ET612, côté composants.

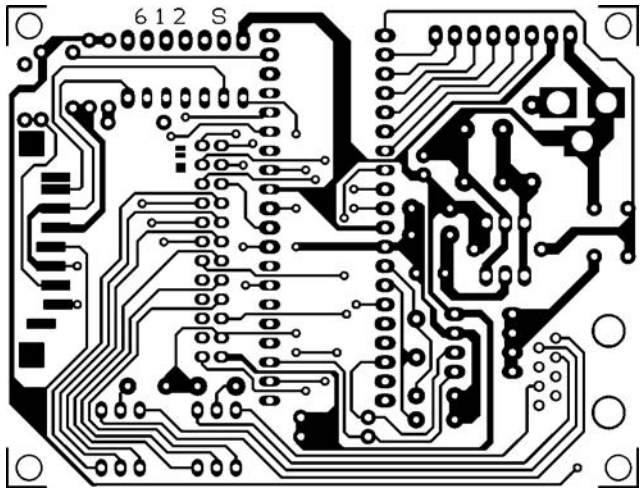


Figure 8b-2 : Dessin, à l'échelle 1, du circuit imprimé double face de l'interface réseau-platine d'expérimentation ET612, côté soudures.

Liste des composants

R1 ..... 10 k
R2 ..... 10 k
R3 ..... 10 k
R4 ..... 4,7 k
R5 ..... 4,7 k
R6 ..... 4,7 k

C1..... 100 nF multicouche
C2..... 470 µF 25 V électrolytique
C3..... 100 nF multicouche
C4..... 470 µF 16 V électrolytique
C5..... 100 nF multicouche
C6..... 47 µF 16 V électrolytique
C7..... 100 nF multicouche
C8..... 100 nF multicouche
C9..... 15 pF céramique
C10 ... 15 pF céramique

D1 ..... 1N4007
D2 ..... BAT85
D3 ..... BAT85
D4 ..... BAT85

MRTL. module serveur Web avec RTL8019-ET616
U1..... 7805
U2..... PIC16F458
U3..... LM1086-3.3
U4..... 74HC125
U5..... 24LC256

Q1 ..... quartz 20 MHz
FL1.... filtre FB2022

P1..... micropoussoir

Divers :

1 support 2 x 4
1 support 2 x 7
1 support 2 x 20 double pas
1 prise d'alimentation
1 barrette mâle à 8 broches
1 lecteur pour carte SD
1 connecteur RJ45
1 connecteur barrette mâle double (2 x 12 = 24 broches)

Sauf spécification contraire, toutes les résistances sont des 1/4 W à 5 %.

profit d'une approche plus concrète : en effet, développer un programme résident utilisant le "stack" TCP/IP n'implique pas nécessairement que l'on connaisse tous les détails de son fonctionnement. Si nous focalisons notre attention sur le modèle de référence utilisé dans le développement TCP/IP nous voyons qu'il est constitué d'une série de niveaux organisés hiérarchiquement de telle manière que chacun fournit à celui immédiatement supérieur des services, tout en cachant les détails d'implémentation. C'est cette

particularité qui nous permet de l'intégrer facilement en agissant au niveau le plus élevé. Si nous ajoutons un nouveau protocole en maintenant fixes les règles de communication avec le niveau inférieur, nous n'aurons pas à intervenir ultérieurement : tous les autres niveaux se comporteront en conséquence. Ce qui est important à comprendre, c'est le mode selon lequel le "stack" TCP/IP Microchip a été développé. L'implémentation de ce dernier à l'intérieur d'un système d'exploitation multitâche comme Windows et sur un dispositif avec ressources pratiquement illimitées comme un ordinateur d'aujourd'hui est chose relativement simple ; les choses sont nettement plus compliquées en revanche à partir du moment où nous tentons de faire la même chose avec un microcontrôleur à huit bits, avec peu de RAM et peu d'espace à notre disposition pour la programmation.

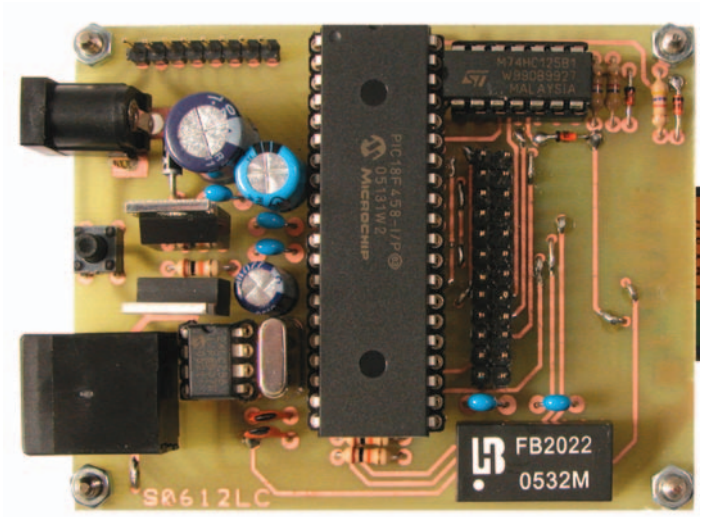


Figure 9 : Photo d'un des prototypes de l'interface réseau-platine d'expérimentation ET612.

Pour que ce soit faisable, on a utilisé le C18 et un mode d'implémentation nommé "multitasking (multitâche) coopératif". L'élaboration avance au fil de tâches ("tasks") suffisamment petites dont chacune accomplit son propre travail, à la fin duquel elle restitue le contrôle de son exécution à la tâche suivante. Chacune coopère donc avec l'autre pour mener à bien l'élaboration. Ainsi, le système est organisé comme une machine à états définis ou FSM (Finite State Machine) dont les transitions se font face à des événements déterminés. Il est clair que le programme résident d'intégration doit lui aussi être développé dans cet esprit. Si nous ajoutons que nous nous servons d'un langage facilitant la modularité en permettant le développement de fonctions autonomes, vous comprendrez comment va se dérouler l'implémentation de notre programme résident : nous prendrons la source du "stack" TCP/IP, laquelle est librement modifiable, nous éliminerons les parties qui ne nous servent pas et nous ajouterons une série de fonctions qui implémentent le protocole FTP selon les RFC standard. Enfin, nous insèrerons dans le programme principal le code relatif à l'échantillonnage des données de température, à leur enregistrement sur SD et à leur téléchargement via FTP sur un serveur adéquat. Nous travaillerons principalement sur le dernier niveau du modèle de référence. Pour avoir une idée de l'organisation de l'implémentation du "stack" par rapport au modèle de référence, voir la **figure 10**.

Dans ce développement nous maintiendrons la possibilité de gérer des paquets ICMP, de telle façon que l'usager puisse vérifier la présence du dispositif dans son réseau local avec un simple "Ping". En second lieu, nous utiliserons deux fonctions implémentées dans le module ARP pour contacter le serveur FTP et recevoir son adresse MAC afin de pouvoir continuer la communication correctement. Attention, à la différence de ce que proposent directement les Notes d'application de la platine d'expérimentation de Microchip, l'implémentation du protocole FTP que nous proposons –nous–, permet l'utilisation du PIC comme client. Ceci est très important pour saisir les différentes approches possibles du développement des deux fonctions : dans le premier cas, nous attendons une demande de l'extérieur et nous répondons en conséquence, dans l'autre en revanche on interagit avec le serveur en première personne et la transition dépend de la réponse que l'on reçoit de lui.

Le protocole FTP

Ce protocole a été décrit pour la première fois par la publication de la RFC959 en 1985 : ce document contient toutes les règles auxquelles doit se tenir n'importe quel logiciel développé pour gérer le FTP. Naturellement, au cours des ans, d'autres publications ont eu lieu et elles ont intégré les fonctions de base ; les efforts se sont concentrés en particulier sur l'amélioration de la sécurité dans l'interaction client/serveur. En effet, dans le FTP l'accès est géré par deux commandes (USER et PASS) qui envoient le nom de l'usager et le mot de passe en clair (cela expose l'usager au "sniffing" des desperados qui écument le Net). Les RFC 2228 et 2577 font face à ce problème : une session de transfert est initialisée par le client puis validée par une série de réponses ("replies") du serveur, chaque réponse ("reply") étant identifiée par un

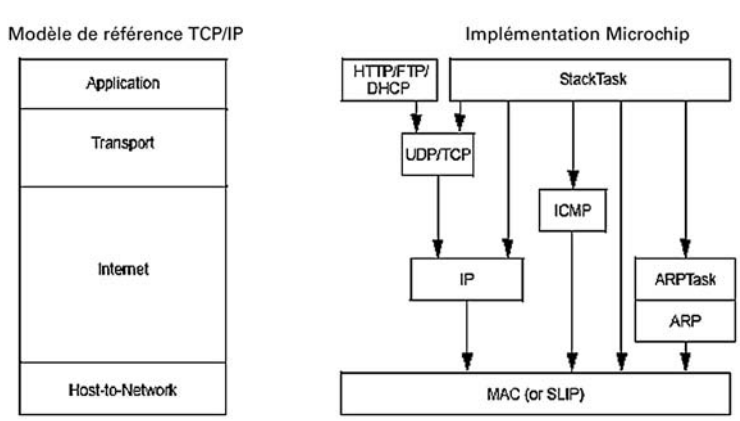


Figure 10 : Organigramme montrant comment est implémenté le "stack" par rapport au modèle de référence.

Tableau 3.

Valeur	DESCRIPTION
1xx	La réponse est préliminaire ; cela signifie que le serveur n'a pas encore terminé l'opération requise et qu'il suivra une autre signalisation.
2xx	Réponse avec issue positive : l'opération s'est correctement terminée.
3xx	Une erreur s'est produite durant le processus ; l'usager doit revoir la demande et la retransmettre.
4xx	Réponse avec issue négative : la demande n'a pas abouti.
5xx	Une erreur grave s'est produite.

"Listing" 1.

```
---------------------------- Frame ID: 0 -----------------------------
Data Link Control   (DLC):
        Destination: FFFFFFFFFFFF   [BROADCAST]
        Source: 0080C8F8E7F4
        EtherType: 0x0806   (Address Resolution Protocol (ARP))
Address Resolution Protocol   (ARP):
        Hardware Type: 1   (Ethernet)
        Protocol Type: 0x0800   (IP)
        Hardware Addr Length: 6 bytes
        Protocol Addr Length: 4 bytes
        Operation: 1   (Request)
        Sender Ethernet Addr: 0080C8F8E7F4
        Sender IP Address: 192.168.0.7
        Target Ethernet Addr: 000000000000
        Target IP Address: 192.168.0.10
Data/FCS:
        Data/Padding: [18 bytes]
        Frame Check Sequence: 0xB2351EAB (Correct)

0000:   FF FF FF FF  FF FF 00 80  C8 F8 E7 F4  08 06 00 01       ........ÈøçôÈøçô....
0010:   08 00 06 04  00 01 00 80  C8 F8 E7 F4  C0 A8 00 07       ........ÈøçôÀ¨..
0020:   00 00 00 00  00 00 C0 A8  00 0A 20 20  20 20 20 20       ......À¨..
0030:   20 20 20 20  20 20 20 20  20 20 20 20  B2 35 1E AB                  _5.»

---------------------------- Frame ID: 1 -----------------------------
Data Link Control   (DLC):
        Destination: 0080C8F8E7F4
        Source: 00E0182DB501
        EtherType: 0x0806   (Address Resolution Protocol (ARP))
Address Resolution Protocol   (ARP):
        Hardware Type: 1   (Ethernet)
        Protocol Type: 0x0800   (IP)
        Hardware Addr Length: 6 bytes
        Protocol Addr Length: 4 bytes
        Operation: 2   (Reply)
        Sender Ethernet Addr: 00E0182DB501
        Sender IP Address: 192.168.0.10
        Target Ethernet Addr: 0080C8F8E7F4
        Target IP Address: 192.168.0.7
Data/FCS:
0000:   00 80 C8 F8  E7 F4 00 E0  18 2D B5 01  08 06 00 01       ..ÈøçôÈøçô.....
0010:   08 00 06 04  00 02 00 E0  18 2D B5 01  C0 A8 00 0A       .......à.-µ.À¨..
0020:   00 80 C8 F8  E7 F4 C0 A8  00 07 00 00  00 00 00 00       ..ÈøçôÀ¨........
0030:   00 00 00 00  00 00 00 00  00 00 00 00  98 75 79 D2       ............uyÒ
```

Dans cette première phase, le client envoie un paquet de "broadcast" (diffusion) destiné, par conséquent, à tous les nœuds du réseau avec une demande ARP. C'est comme si la platine envoyait une demande de type : "Qui a l'adresse IP 192.168.0.10 ?". En fait le client demande l'adresse de niveau Data Link (MAC) nécessaire pour l'envoi des trames et qui encapsulera les paquets suivants. Le MAC identifie univoquement l'interface réseau correspondante sur le serveur. On note, en effet, que le champ "Target Ethernet Address" est nul. C'est un pas préliminaire fondamental puisque notre platine utilisera un cache adéquat pour sauvegarder durant chaque session les données d'adressage du nœud avec lequel elle sera mise en communication.

C'est seulement l'hôte ("host") ayant l'adresse 192.168.0.10 qui répond au paquet en insérant dans la réponse son adresse Data Link (00:E0:18:2D:B5:01) dans le champ qui était d'abord nul. Rappelons que cette valeur se compose de 48 bits divisés en deux groupes de 3 octets. Le premier identifie le constructeur de l'interface et le deuxième est un numéro de série. Dans la puce RTL8019AS les trois derniers octets sont mis à zéro à la mise en marche et ils peuvent donc être attribués librement.

nombre à trois chiffres (FTP reply-code). Le premier chiffre définit l'état général de la commande, comme le montre le **Tableau 4**.

L'interaction avec un serveur FTP se fait à travers l'ouverture de deux canaux de connexion séparés : l'un est utilisé pour envoyer des commandes et recevoir les réponses ; l'autre pour le transfert des données. Le premier canal est réalisé au début d'une session par l'ouverture par le client d'un "socket" (au sens propre : douille ou prise femelle) sur le port 21 ; le second, par l'ouverture par le serveur, avant le transfert de données, d'un "socket" sur le port 20. Ce double rôle du client, d'abord actif puis passif, est essentiel pour comprendre comment se fait réellement l'interaction de notre platine avec un serveur FTP. Dans le programme résident, nous devrons d'abord créer le canal de données, effectuer le login en transmettant notre accréditation

Tableau 4.

Valeur	DESCRIPTION
x0x	Le message est relatif à la syntaxe de la commande envoyée.
x1x	Le message est de type informatif. Il concerne typiquement une commande d'aide ou d'état.
x2x	Le message concerne l'état de la connexion.
x3x	Le message concerne l'authentification de l'usager. Dans le cas où l'on tape, par exemple, un mot de passe erroné, le code numérique est "530".
x4x	Non spécifié.
x5x	Le message concerne le fichier qui est transféré.

"Listing" 2.

```
---------------------------- Frame ID: 9 ----------------------------
Data Link Control   (DLC):
        Destination: 00E0182DB501
        Source: 0080C8F8E7F4
        EtherType: 0x0800   (Internet Protocol (IP))
Internet Protocol   (IP):
        ....
        Source Address: 192.168.0.7
        Destination Address: 192.168.0.10
Transmission Control Protocol   (TCP):
        Source Port: 1031
        Destination Port: 21   (File Transfer [Control])
        Sequence Number: 655610    (next expected sequence number: 655622)
        Acknowledgement Number: 3572774138
        ....
File Transfer Protocol Control   (FTP):
        Line 1: USER charles<0D><0A>
Data/FCS:
0000:   00 E0 18 2D  B5 01 00 80  C8 F8 E7 F4  08 00 45 00       .à.-µ...ÈøçÔ..E.
0010:   00 34 44 00  40 00 80 06  35 62 C0 A8  00 07 C0 A8       .4D.@...5bÀ¨..À¨
0020:   00 0A 04 07  00 15 00 0A  00 FA D4 F4  34 FA 50 18       .........úÔô4úP.
0030:   21 A4 67 B7  00 00 55 53  45 52 20 63  61 72 6C 6F       !€g·..USER charles
0040:   0D 0A 50 35  F6 9E                                      ..P5ö.

---------------------------- Frame ID: 10 ----------------------------
Data Link Control   (DLC):
        Destination: 0080C8F8E7F4
        Source: 00E0182DB501
        EtherType: 0x0800   (Internet Protocol (IP))
Internet Protocol   (IP):
        ....
        Source Address: 192.168.0.10
        Destination Address: 192.168.0.7
Transmission Control Protocol   (TCP):
        Source Port: 21   (File Transfer [Control])
        Destination Port: 1031
        Sequence Number: 3572774138    (next expected sequence number: 3572774171)
        Acknowledgement Number: 655622
        ....
File Transfer Protocol Control   (FTP):
        Line 1: 331 Password required for charles<0D><0A>
Data/FCS:
0000:   00 80 C8 F8  E7 F4 00 E0  18 2D B5 01  08 00 45 00       ..ÈøçÔ.à.-µ...E.
0010:   00 49 01 11  40 00 80 06  78 3C C0 A8  00 0A C0 A8       .I..@...x<À¨..À¨
0020:   00 07 00 15  04 07 D4 F4  34 FA 00 0A  01 06 50 18       ......ÔÔ4ú....P.
0030:   44 64 CF 90  00 00 33 33  31 20 50 61  73 73 77 6F       DdÏ...331 Passwo
0040:   72 64 20 72  65 71 75 69  72 65 64 20  66 6F 72 20       rd required for
0050:   63 61 72 6C  6F 0D 0A 77  BE 92 C0                      charles..w_.À

---------------------------- Frame ID: 11 ----------------------------
ACK da parte del Client
---------------------------- Frame ID: 12 ----------------------------
```

Le client envoie le nom de l'usager à travers la commande USER. On remarque que le port de destination est celui défini par le canal commandes FTP. Attention, chaque commande doit se terminer par une paire "new line-carriage return" (à la ligne-retour chariot) et les paramètres sont passés en clair.

Réponse de la part du serveur qui accepte le login de la part de l'usager "charles" et demande la frappe d'un mot de passe à travers une réponse 331. Ensuite le client envoie un paquet de confirmation (ACK).

suite “Listing” 2.

```
Data Link Control   (DLC):
        Destination: 00E0182DB501
        Source: 0080C8F8E7F4
        EtherType: 0x0800   (Internet Protocol (IP))
Internet Protocol   (IP):
        ....
        Source Address: 192.168.0.7
        Destination Address: 192.168.0.10
Transmission Control Protocol   (TCP):
        Source Port: 1031
        Destination Port: 21   (File Transfer [Control])
        Sequence Number: 655622    (next expected sequence number: 655634)
        Acknowledgement Number: 3572774171
        ....
File Transfer Protocol Control   (FTP):
        Line 1: PASS prova<0D><0A>
Data/FCS:
0000:   00 E0 18 2D  B5 01 00 80  C8 F8 E7 F4  08 00 45 00       .à.-µ...ÈøçÔ..E.
0010:   00 34 46 00  40 00 80 06  33 62 C0 A8  00 07 C0 A8       .4F.@...3bÀ¨..À¨
0020:   00 0A 04 07  00 15 00 0A  01 06 D4 F4  35 1B 50 18       ..........ÔôõP.
0030:   21 83 43 C0  00 00 50 41  53 53 20 70  72 6F 76 61       !.CÀ..PASS prova
0040:   0D 0A 56 28  B1 FF                                       ..V(±.

---------------------------- Frame ID: 13 -----------------------------
Data Link Control (DLC):
        Destination: 0080C8F8E7F4
        Source: 00E0182DB501
        EtherType: 0x0800   (Internet Protocol (IP))
Internet Protocol   (IP):
        ....
        Source Address: 192.168.0.10
        Destination Address: 192.168.0.7
Transmission Control Protocol (TCP):
        Source Port: 21   (File Transfer [Control])
        Destination Port: 1031
        Sequence Number: 3572774171    (next expected sequence number: 3572774186)
        Acknowledgement Number: 655634
File Transfer Protocol Control   (FTP):
        Line 1: 230 Logged on<0D><0A>
Data/FCS:
0000:   00 80 C8 F8  E7 F4 00 E0  18 2D B5 01  08 00 45 00       ..ÈøçÔ.à.-µ...E.
0010:   00 37 01 1C  40 00 80 06  78 43 C0 A8  00 0A C0 A8       .7..@...xCÀ¨..À¨
0020:   00 07 00 15  04 07 D4 F4  35 1B 00 0A  01 12 50 18       ......ÔôõP.....P.
0030:   44 58 C6 B0  00 00 32 33  30 20 4C 6F  67 67 65 64       DXÆ°..230 Logged
0040:   20 6F 6E 0D  0A 5A E8 F2  41                             on..ZèòA
---------------------------- Frame ID: 14 -----------------------------
ACK de la part du Client
```

On en arrive ainsi à l’envoi de la part du client du mot de passe d’authentification. Si vous regardez bien le paquet, le flux correspondant se voit clairement. Vous vous rappelez? Avoir la possibilité de surveiller le trafic du réseau peut être une source sûre d’informations!

Voici donc la réponse ponctuelle de la part du serveur. La réponse 230 précise que le client est correctement relié avec le serveur à travers le canal commandes FTP. Ensuite un paquet de confirmation est envoyé.

et demander le transfert du fichier. Il sera alors nécessaire de se mettre en réception sur le port 21 pour réaliser le transfert des données proprement dites. La possibilité de modifier le port des données par rapport au standard est intéressante.

Prenons un exemple de session de transfert en analysant le log d’un serveur FTP en logiciel libre (“freeware”) qui nous servira de référence pour notre application (voyez la **figure 11**: log sur la console de FileZilla). On identifie facilement trois phases fondamentales: dans la première (1), ou Connexion, le client ouvre un canal de communication avec le serveur pour l’envoi des commandes; l’ouverture est signalée par la réponse “220” et l’envoi du message de bienvenue. En 2, ou Authentification, le client envoie le nom de l’usager avec la commande “USER <nome utente>”. A la réponse “331” réclamant le mot de passe, le client envoie la commande “PASS <password>”. Si cette deuxième phase est menée à bien, la réponse “230” est reçue. Dans la troisième et dernière, ou Ouverture Canal Données, le client envoie d’abord une commande “PORT IP4,IP3,IP2,IP1,-PORTA1,PORTA0” à travers laquelle le port sur lequel il restera en réception est indiqué au serveur. IPn est l’octet de l’adresse IP. Dans notre exemple, le client a l’IP 192.168.0.7. Les deux octets PORTAn représentent la valeur à 16 bits correspondant au port de données que l’on souhaite utiliser. Dans l’exemple, on utilise le port 1031 (10000000111b). Ensuite, le canal est ouvert après la commande “STOR termo.txt” par laquelle le client demande d’envoyer le fichier termo.txt. L’ouverture correcte du canal est signalée par la réponse “150”. A la fin du transfert, le serveur envoie la réponse “226”. La quatrième et dernière phase

"Listing" 3.

```
---------------------------- Frame ID: 22 -----------------------------
Data Link Control   (DLC):
        Destination: 00E0182DB501
        Source: 0080C8F8E7F4
        EtherType: 0x0800   (Internet Protocol (IP))
Internet Protocol   (IP):
        Source Address: 192.168.0.7
        Destination Address: 192.168.0.10
Transmission Control Protocol   (TCP):
        Source Port: 1032
        Destination Port: 20   (File Transfer [Default Data])
        Sequence Number: 667468    (next expected sequence number: 668928)
        Acknowledgement Number: 3575768854
File Transfer Protocol Data   (FTPDATA):
        Data: [1460 bytes]
Data/FCS:
0000:   00 E0 18 2D  B5 01 00 80  C8 F8 E7 F4  08 00 45 00       .à.-µ...ÈøçÔ..E.
0010:   05 DC 4B 00  40 00 80 06  28 BA C0 A8  00 07 C0 A8       .ÜK.@...(°À¨..À¨
0020:   00 0A 04 08  00 14 00 0A  2F 4C D5 21  E7 16 50 10       ......../LÕ!ç.P.
0030:   22 38 3C D3  00 00 45 6C  65 74 74 72  6F 6E 69 63       «8<Ó..Electronique
0040:   61 20 49 6E  20 45 6C 65  74 74 72 6F  6E 69 63 61        LM Electronique
0050:   20 49 6E 20  45 6C 65 74  74 72 6F 6E  69 63 61 20        LM Electronique
0060:   49 6E 20 45  6C 65 74 74  72 6F 6E 69  63 61 20 49       LM Electronique
0070:   6E 20 45 6C  65 74 74 72  6F 6E 69 63  61 20 49 6E       LM Electronique LM
.....
05C0:   69 63 61 20  49 6E 20 45  6C 65 74 74  72 6F 6E 69       ique LM Electronique
05D0:   63 61 20 49  6E 20 45 6C  65 74 74 72  6F 6E 69 63       LM Electronique
05E0:   61 20 49 6E  20 45 6C 65  74 74 4D D2  2D 9C                          LM ElectMÒ-.
```

Dans notre exemple le fichier contient le nom de la revue répété plusieurs fois. Comme on le voit clairement, la séquence d'octets est transférée à un "socket" (au sens propre : support) différent du précédent et qui est identifié par la paire IP: PORTA 192.168.0.10:20. Découpé en plusieurs paquets, le fichier est transféré séquentiellement dans le serveur.

est la Fermeture Session : le client clot la communication avec la commande "QUIT".

Côté client, la séquence entière apparaît comme le montre la **figure 12**, laquelle illustre une session FTP vue, justement, du côté du client.

Voir derrière les écrans

Avant d'aborder l'analyse des sections clés de notre programme résident, nous voulons vous montrer ce qui se passe réellement sur un réseau pendant une session FTP.

Pour effectuer un développement correct et stable, nous avons surveillé les paquets transitant sur le segment du réseau sur lequel se fait la communication entre client et serveur. Il existe divers outils qui permettent ce type d'opération et représentent une instrumentation optimale pour voir exactement la séquence des trames échangées. Les applications professionnelles, en particulier, permettent d'analyser les paquets en transit en offrant une vision détaillée des informations de compétence des divers niveaux (PCI Protocol Control Information) du modèle de référence TCP/IP. Il est donc possible d'effectuer un test approfondi du comportement de notre programme résident en temps réel.

Disons tout d'abord que, dans la petite expérimentation que cet article vous propose de réaliser (voir ci-après **La réalisation pratique**), l'adresse IP du client est 192.168.0.7, celle du serveur 192.168.0.10. Le "tracking" (suivi de variation, pistage) de la session tout entière est visible en une séquence de 36 trames, parmi lesquelles, dans le **"Listing" 1**, vous pouvez voir les fondamentales. Dans les deux premières, on voir clairement comment se fait un premier échange d'informations entre client et serveur.

Au moment où nous ouvrons une session FTP (dans Windows il suffit de taper la commande ftp <numero IP serverFTP>), notre carte réseau envoie un paquet ARP pour recevoir l'adresse MAC qui identifie l'interface du serveur. Ensuite, l'ouverture d'un "socket" sur l'IP 192.168.0.10, port 21 pour le canal commande est demandée.

La séquence consiste en le fameux "three-way handshaking". Le client envoie un segment TCP avec le bit SYN au un logique et le bit ACK au zéro logique, pour proposer un "sequence number" x. Le segment arrive à l'interface réseau du serveur. Alors, si le processus arrivant sur le port accepte la demande de connexion, il envoie en réponse un segment de confirmation avec SYN et ACK tous deux au un logique, "acknowledgement number" égal à x+1 et un nouveau "sequence number" y.

L'activation de la connexion se termine alors par l'envoi d'un dernier segment de confirmation de la part du client avec "sequence number" égal à x+1 et "acknowledgement number" égal à y+1. Cette procédure permet une synchronisation précise entre les deux dispositifs. A partir de là, le canal de commande est ouvert et l'échange des premières séquences FTP (comme le message de bienvenue du serveur) peuvent commencer.

Dans le **"Listing" 2** nous voyons les trames suivantes (2 à 14) concernant l'authentification ; pour des raisons de place, nous avons exclu certains champs peu significatifs. Dans la séquence, on peut facilement comprendre comment se fait la communication entre deux dispositifs : notre client envoie une commande et le serveur l'exécute en répondant par un code numérique.

Quand il est nécessaire d'ouvrir un canal de communication, les trois paquets du "three-way-handshaking" sont échangés. La chose se répète avec la commande PORT et pour l'initialisation du canal des données avec l'instruction STOR.

Pour conclure, dans le **"Listing" 3** nous voyons de quelle manière un paquet

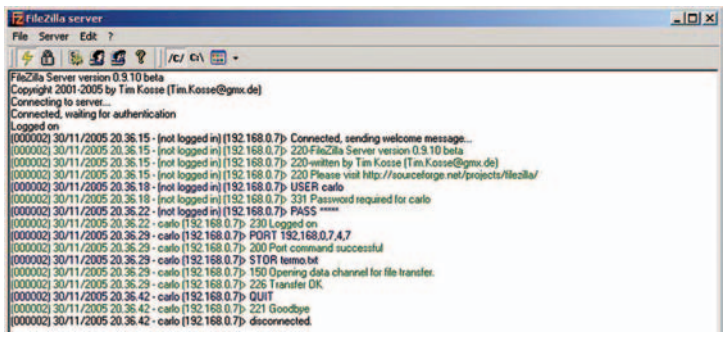


Figure 11 : Log sur la console de FileZilla (serveur FTP en logiciel libre) utilisé comme référence pour notre application.

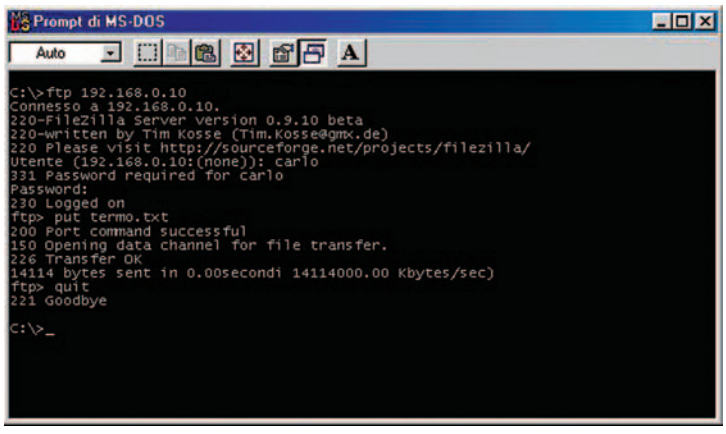


Figure 12 : Une session FTP vue côté client.

est envoyé par le client au serveur Web durant le transfert réel des données.

La réalisation pratique

La réalisation pratique de cette interface Client FTP avec PIC et SD-Card comporte deux platines, la petite ET616 et la grande ET612 qui intègre la précédente en tant que module.

La première platine ET616 (**figures 2, 4, 5 et 6**) est constituée d'un circuit imprimé double face pour composants à montage en surface (CMS). Réalisez ce circuit imprimé à l'aide de la **figure 4b-1 et 2**. Seul le connecteur barrette femelle à 2 x 12 = 24 trous est traversant et relie les deux faces. Sur la face a, montez le circuit intégré RTL8019 (aux innombrables broches disposées sur les quatre côtés) en le soudant avec un petit fer à panne très fine et du tinol de 0,5 mm. Vérifiez bien ces nombreuses soudures (ni court-circuit entre pistes ou pastilles ni soudure froide collée). Montez la double barrette traversante, vérifiez bien ces nouvelles soudures. Puisque vous voilà de l'autre côté, montez-y les composants CMS restants (toujours avec un petit fer à panne très fine et du tinol le plus fin que vous trouvez) : ce sont des résistances et des condensateurs puis un quartz et enfin, seuls éléments polarisés, trois LED. Après de multiples vérifications, mettez cette platine en réserve : vous l'insèrerez dans la suivante quand vous l'aurez réalisée. Mais, nous l'avons dit, ce module servira aussi à des montages futurs.

La seconde platine ET612 (**figures 8, 9 et photo de première page**) est également constituée d'un circuit imprimé double face, mais cette fois pour composants classiques traversants (sauf lecteur de carte SD, monté côté soudures du circuit imprimé). Réalisez ce circuit imprimé, dont la **figure 8b-1 et 2** donne les dessins à l'échelle 1. Commencez par insérer les trois supports de circuits intégrés, la double barrette mâle 2 x 12 broches (elle recevra à la toute fin le module ET616 que vous venez de réaliser) et la barrette mâle à huit broches. Vérifiez attentivement vos soudures (ni court-circuit entre pistes ou pastilles ni soudure froide collée). Insérez et soudez ensuite tous les composants (comme le montrent les **figures 8a et 9**), en poursuivant par les résistances, condensateurs, diodes, LED, selfs, quartz, régulateurs (debout, sans dissipateur) et en terminant par les "périphériques" : la prise d'alimentation, le poussoir P1, la RJ45 et le filtre FB2022. Attention à l'orientation des composants polarisés : circuits intégrés (insérez-les à la fin), diodes, LED, régulateurs (semelles métalliques vers C1 et U5), filtre (point repère-détrompeur en bas à gauche) et électrolytiques. Voilà pour le côté composants.

Retournez alors la platine côté soudures et montez le connecteur pour SD-Card. Vérifiez bien toutes les polarités et (encore une fois) la qualité des soudures sur les deux faces de la platine, puis insérez les circuits intégrés dans leurs supports (attention à leur orientation : repère-détrompeurs en U bien orientés, tous vers la gauche ou vers le haut).

Vous pouvez maintenant insérer la platine module ET616 sur la platine ET612 à l'aide de la paire M/F de doubles barrettes 2 x 12 = 24 pôles (voir **photo de première page**). Insérez une carte SD de 64 Mo dans son lecteur. Prévoyez une petite alimentation bloc secteur de 9 Vcc. Votre interface-platine d'expérimentation est prête à être utilisée. Pour cela, rendez-vous au prochain numéro d'ELM pour la deuxième partie de cet article, dont la visée didactique ne vous aura pas échappée.

Conclusion et à suivre

Après avoir un peu détaillé l'analyse du fonctionnement du protocole FTP et certains aspects des paquets échangés entre client et serveur durant une session de communication, dans la seconde partie nous commencerons à éclaircir les points fondamentaux du développement du programme résident. Pour finir, nous expliquerons comment configurer notre nœud en reliant une sonde thermique à l'une des cinq entrées dispositifs : nous verrons comment les données acquises sont transférées sur un serveur Web pour y être figurées au sein d'une page Web accessible par le réseau.

Comment construire ce montage ?

Tout le matériel nécessaire pour construire cette interface-platine d'expérimentation Client FTP avec PIC et SD-Card ET616 et ET612 est disponible chez certains de nos annonceurs. Voir les publicités dans la revue.

Les typons des circuits imprimés et les programmes disponibles sont téléchargeables à l'adresse suivante :
http://www.electronique-magazine.com/circuitrevue/086.zip. ◆

KIT DE RÉGLAGE DE VOLUME - 6 CANAUX

Compatible avec tout processeur numérique 2x3 voies ou décodeur numérique DOLBY 5:1

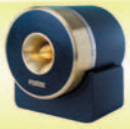
- Niveau de chaque canal ajustable pour harmonisation en fonction de la sensibilité des amplis utilisés
- Technique audiophile **exclusivité Selectronic** totalement neutre et transparente (buffers d'E/S à FETs)
- Potentiomètre ALPS motorisé
- Condensateurs de filtrage TFRS, etc.
- Circuit imprimé double-face avec vernis épargne
- Le grand luxe habituel



Kit de base (sans coffret ni télécommande) **753.4310-1** 199,00 €TTC
Kit avec TÉLÉCOMMANDE (sans coffret) **7533.4310-2** 239,00 €TTC
MONTÉ, en ordre de marche, en coffret avec télécommande 753.4310-3M 399,00 €TTC
(Garantie: 1 an)



FOSTEX



➜ Toute la gamme **en stock chez Selectronic**

KIT DE COMMANDE POUR MOTEUR PAS À PAS

- A base de L297 et L6203 ce kit permet de piloter tout moteur pas à pas bipolaire jusque 4A sous 36V
- Signaux de cde sur connecteur 10pts
- Bobinages et alimentation de puissance sur 3 borniers 2pts
- CI double face 75x75mm "renforcé" 70µm vernis et sérigraphié
- ***Documentation disponible sur le site www.selectronic.fr*** *(rubrique Téléchargement")*



Le kit 753.6950-1 49,90 €TTC
Version monté (en état de marche)
753.6950-1M 79,00 €TTC

ELM07/26-2 Photos non contractuelles

Selectronic
L'UNIVERS ELECTRONIQUE

B.P 10050 59891 LILLE Cedex 9
Tél. 0 328 550 328 - Fax : 0 328 550 329
www.selectronic.fr



NOS MAGASINS :

PARIS :
11 Place de la Nation 75011 *(Métro Nation)*
Tél. 01.55.25.88.00 • Fax : 01.55.25.88.01



LILLE (Ronchin) :
ZAC de l'Orée du Golf
16, rue Jules Verne 59790 RONCHIN

Conditions générales de vente : Réglement à la commande : frais de port et d'emballage 5,00€, **FRANCO** à partir de 130,00€. Contre-remboursement : +10,00€. **Livraison par transporteur :** supplément de port de 13,00€. Tous nos prix sont TTC



Selectronic
L'UNIVERS ELECTRONIQUE

Catalogue **Général**

Commandez-le dès maintenant !

Plus de 800 pages en couleur

Coupon à retourner à: **Selectronic B.P 10050 • 59891 LILLE Cedex 9**

❑ OUI, je désire recevoir le **Catalogue Général 2007 Selectronic** ELM
à l'adresse suivante (ci-joint 10 timbres-poste au tarif "lettre" en vigueur ou 5,50€ par chèque):

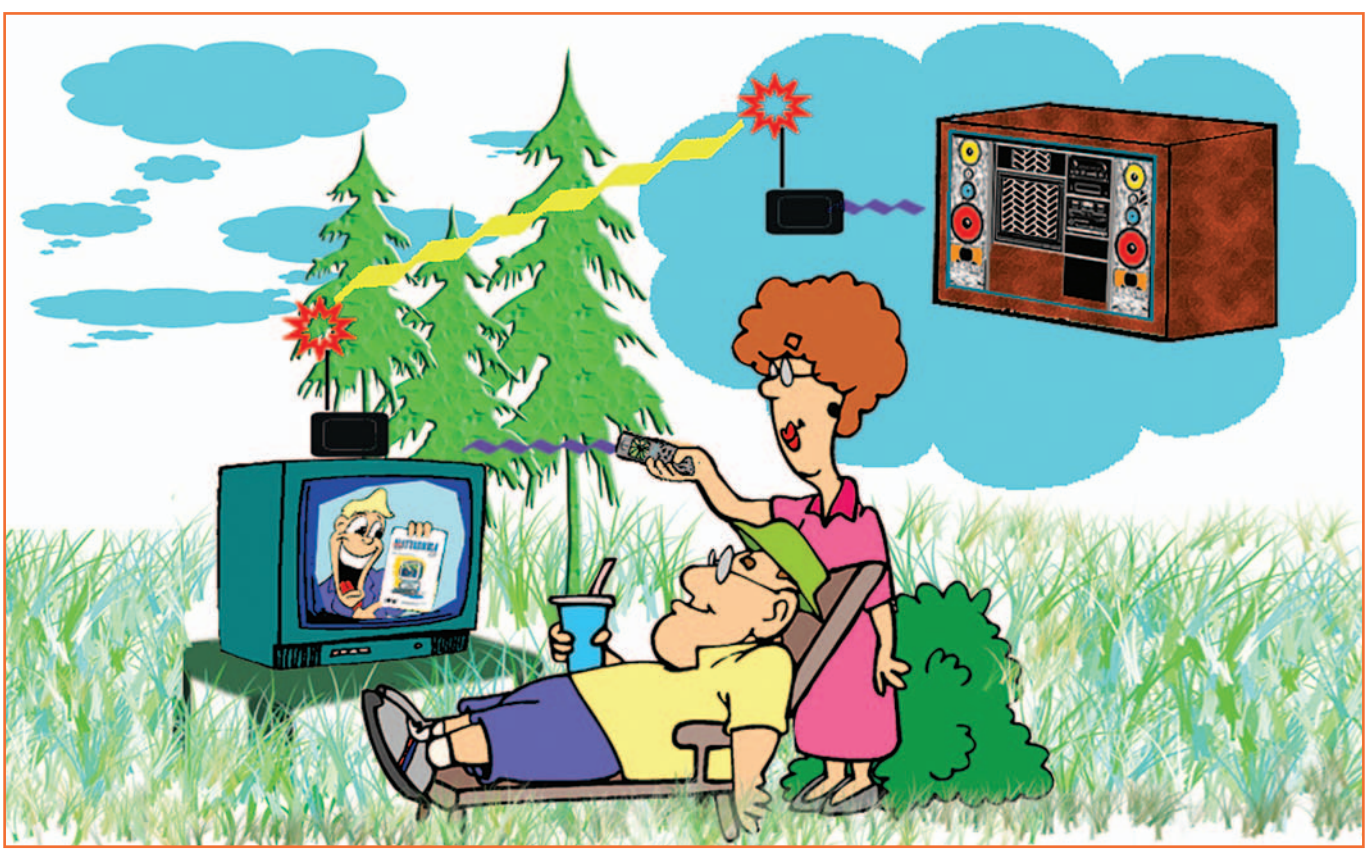
Mr. / Mme : ..... **Tél :** .....
N° : ..... **Rue :** .....
Ville : ..... **Code postal :** .....

"Conformément à la loi informatique et libertés n° 78.17 du 6 janvier 1978, Vous disposez d'un droit d'accès et de rectification aux données vous concernant"

EN1628-1629

Un répéteur HF pour télécommande

Ce répéteur VHF pour télécommande vous permet d'augmenter considérablement la portée de votre télécommande TV ou Hi-Fi, etc. et même de commander un appareil situé dans une autre pièce. Vous pourrez ainsi atteindre tous les dispositifs dotés de commande infra-rouge comme le décodeur, le lecteur de DVD, le magnétoscope, la chaîne AV…



Ceux d'entre vous qui ont construit le TX/RX AV 2,4 GHz EN1557-1558 seront certainement intéressés par le répéteur HF de télécommande infra-rouge que vous propose cet article. En effet, quand nous avions publié l'analyse et la construction de cet émetteur/récepteur audio/vidéo, nous avions prévu une application permettant de transmettre (sans fil bien sûr) les signaux vidéo prélevés sur la PERITEL du décodeur TV ; cette application peut être fort utile si l'on souhaite transmettre le signal de sortie d'un décodeur relié au téléviseur du salon à un second téléviseur situé dans une autre pièce (chambre à coucher ou cuisine…). Ainsi, il est possible de suivre les émissions de télévision où que l'on se trouve dans la maison, sans avoir à débrancher-déplacer-rebrancher le décodeur (ou le magnétoscope ou le lecteur de DVD…) et sans fil !

Mais, dans ce cas, votre télécommande IR (celle qui commande le décodeur ou le magnétoscope ou le lecteur de DVD…) serait inopérante (puisque ces télécommandes supposent une courte distance et surtout une liaison "en vision directe" sans obstacle) ; vous seriez donc obligé de vous déplacer pour commander l'appareil (décodeur…) se trouvant dans une autre pièce.

Eh bien, le montage que cet article vous propose est destiné à palier ce dernier inconvénient. Il est en effet destiné, non seulement à augmenter la portée de votre télécommande IR, mais encore à la rendre insensible aux obstacles interposés, telles les cloisons ou les portes de l'habitation. Cette insensibilité aux obstacles est due au fait que ce répéteur jette un "pont" radio (VHF) entre l'émetteur et le récepteur



Figure 1: Le répéteur pour télécommande se compose d'un module émetteur TX qui reçoit le signal IR de la télécommande et le transforme en un signal radio VHF et d'un module récepteur RX qui le reconvertit en un signal IR capable de commander n'importe quel appareil doté, bien sûr, d'un système de télécommande infra-rouge.

de télécommande IR. Ce répéteur ou pont est constitué d'un TX à placer à 1 ou 2 mètres de la télécommande (le petit boîtier manuel, qui est en fait un émetteur IR) et d'un RX à placer près de l'appareil à commander (ce dernier contient le récepteur IR dissimulé). Le rôle du TX est de recevoir la modulation provenant de la télécommande et d'en acheminer le signal, par radio (donc à travers les obstacles domestiques), vers un RX restituant ce signal au récepteur de télécommande destiné à commander l'appareil distant. Le signal radio réalisant ce pont est en VHF à 350 MHz et son rayon d'action est de quelque 50 mètres en espace libre (soit de quoi agir du jardin vers la maison). Nous avons surtout évoqué le décodeur télévision, mais ce répéteur reste valable pour commander à distance et en dépit des obstacles n'importe quel appareil à télécommande IR (voir figures 1 à 4).

Les schémas électriques

Le schéma électrique du TX est visible figure 7 et celui du RX figure 10.

L'émetteur TX

Chaque fois que l'on presse un bouton de la télécommande IR pour envoyer une commande à l'appareil (par exemple le bouton rouge de mise en marche du décodeur), un signal codé est émis (séquence de niveaux logiques 0 et 1, variable selon la fonction que l'on veut activer, par exemple mise en marche ou choix de la chaîne 2...). Pour plus de fiabilité, le code est modulé en amplitude sur une porteuse de fréquence 50 kHz environ, comme le montre la figure 5 (environ, car cette fréquence change un peu en fonction du constructeur et du type de télécommande).

Ce signal émis par la télécommande IR est capté par la diode détectrice (ou réceptrice) IR BPW41 (DRX1, voir schéma électrique figure 7) située à l'entrée du TX radio EN1628 que nous analysons. Cette diode est reliée à la broche 13 de IC1/A, un inverseur monté avec R1 en amplificateur de gain 50 environ. Le signal de sortie broche 12 est acheminé par C1 au filtre passe-bas JAF1+C2/C3 qui élimine la porteuse à 50 kHz présente sur le signal provenant de la télécommande de façon à ne conserver que le code. A son tour l'inverseur IC1/B amplifie modérément le signal qui est ensuite envoyé à deux inverseurs IC1/C et IC1/D lesquels mettent le signal parfaitement en quadrature. A la broche 6 de IC1/D est reliée DL1 qui, en clignotant, indique la réception correcte du signal.

Le signal est acheminé de cette broche 6 vers l'entrée de IC1/E et IC1/F montés en "buffers" (tampons) afin d'augmenter le courant fourni à l'étage suivant: un oscillateur constitué par TR1, par la demi spire imprimée L1, par C8 et l'ajustable C9.

Quand le signal à la sortie de IC1/F, broche 4, est au niveau logique bas, l'étage oscillateur n'oscille pas et donc aucun signal ne parvient à l'antenne; dès que le signal passe au niveau logique haut, l'étage oscillateur commence à osciller sur une fréquence d'environ 350 MHz.

A la sortie de l'antenne, nous aurons donc un signal reproduisant exactement le code reçu depuis la télécommande, mais modulé cette fois sur une porteuse à 350 MHz, comme le montre la figure 6.

Le circuit est tout simplement alimenté par une pile 6F22 de 9 V dont la tension parvient à IC1 et à l'étage oscillateur à travers JAF2 dont la fonction est d'éliminer tout retour de HF vers l'alimentation.

Le récepteur RX

Le signal capté par l'antenne est envoyé au récepteur à super-réaction constitué par le transistor TR1, par C3, par la demi spire imprimée L1, par C4, C5, C6 et JAF1 qui détecte le signal en éliminant la porteuse à 350 MHz.

Ce RX EN1629 à super-réaction caractérisé par une sensibilité élevée et une faible sélectivité, caractéristiques qui permettent au circuit de recevoir au mieux le signal envoyé par le TX EN1628 décrit ci-dessus (même dans des conditions non optimales).

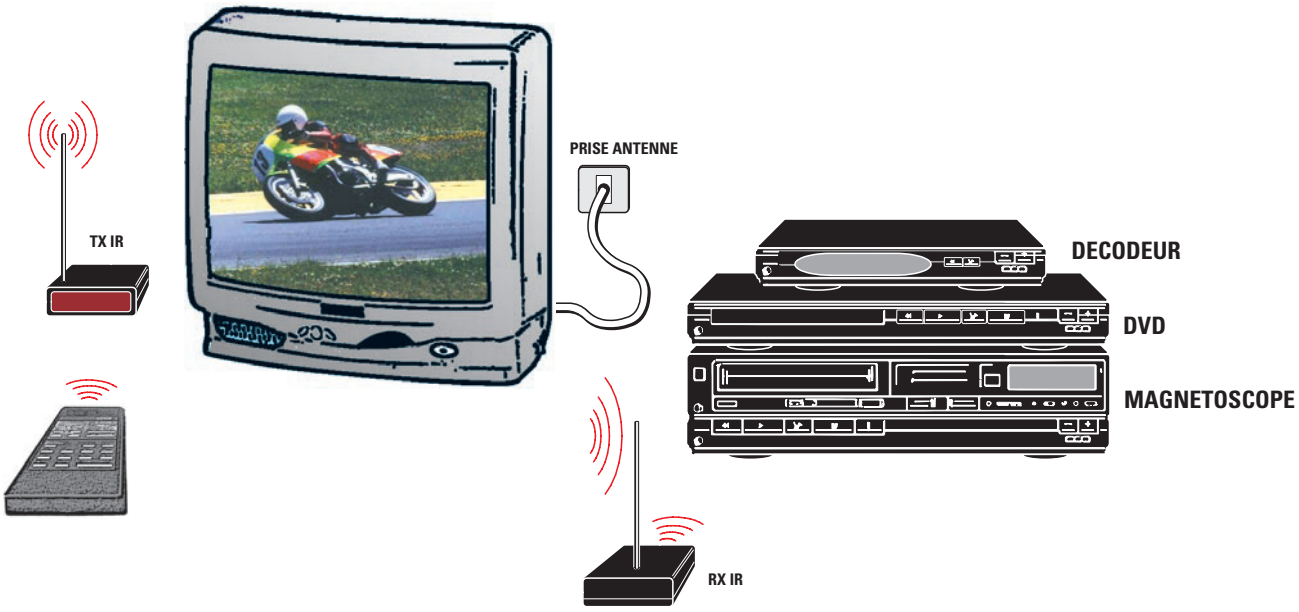


Figure 2 : Le répéteur HF pour télécommande vous permet d'actionner à distance le décodeur, le magnétoscope ou le lecteur de DVD situés dans une autre pièce.

Ainsi, si vous disposez d'une prise d'antenne satellite supplémentaire, vous pourrez regarder vos programmes favoris tranquillement installé dans le jardin.

Après avoir été détecté, le signal est acheminé à l'entrée non-inverseuse de l'étage amplificateur IC2/A qui l'amplifie environ 200 fois et le débarrasse d'éventuels résidus HF.

La broche 7 de IC2/A est reliée à l'entrée non-inverseuse de IC2/B qui constitue le circuit de quadrature. DL1, reliée à la sortie de IC2/B, indique, par son clignotement, que le signal est reçu.

Sa cathode est reliée à la broche 4 de IC3, un NE555 monté en multivibrateur astable, capable d'osciller à une fréquence réglable entre 12 et 68 kHz au moyen du trimmer R16.

Quand le signal à la sortie de IC2/B est au niveau logique haut, IC3 oscille et dès que ce signal passe au niveau logique bas, il n'oscille plus.

La fréquence produite sur la broche 3 de IC3 est acheminée vers la diode émettrice IR CQX89 (DTX1, voir schéma électrique figure 10) qui recrée ainsi en sortie exactement le même code que celui émis par la télécommande et modulé sur une fréquence variable entre 12 et 68 kHz (en mesure d'actionner le décodeur ou autre appareil muni d'un tel système de télécommande infra-rouge).
Si l'émission du signal par le TX et la réception de ce même signal par le RX se font correctement, en pointant la télécommande IR vers le TX et en pressant n'importe quel bouton, la DL1 du TX et la DL1 du RX doivent clignoter en parfaite synchronisation.

Ce circuit aussi est alimenté par une pile 6F22 de 9 V dont la tension parvient à IC2 et IC3 ; l'étage récepteur à super-

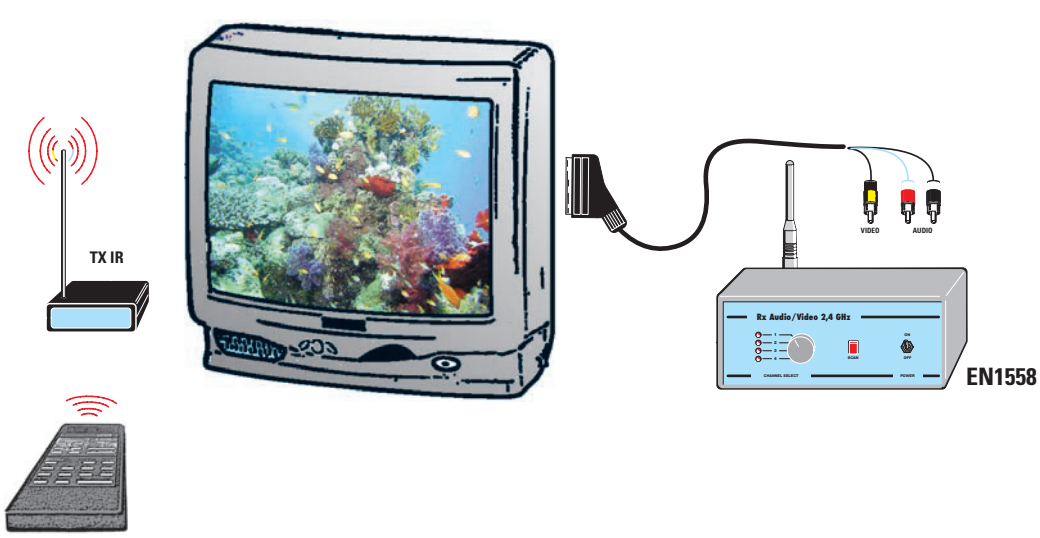


Figure 3 : Si vous n'avez pas de prise d'antenne supplémentaire à laquelle relier le téléviseur, vous pourrez également regarder la télévision hors de votre habitation en utilisant notre émetteur/récepteur audio-vidéo EN1557-1558 ; il vous suffira de relier votre téléviseur au récepteur audio-vidéo EN1558.

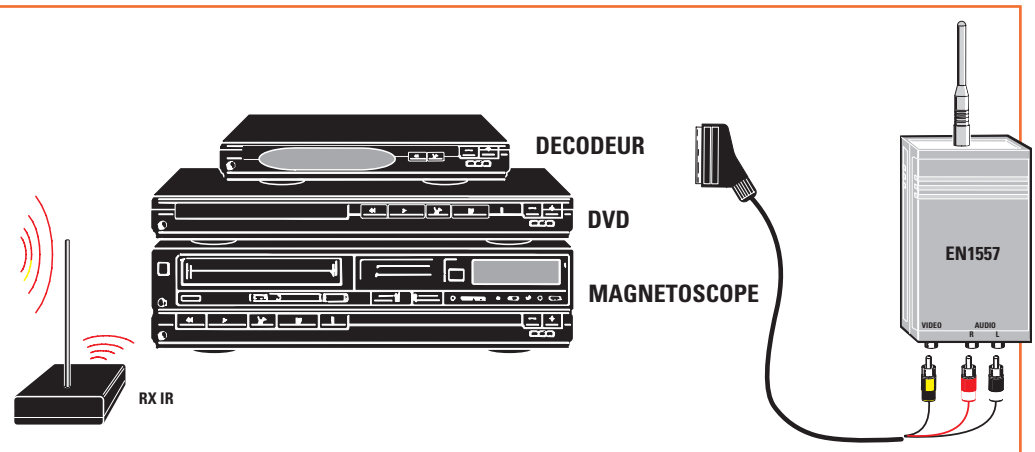


Figure 4 : Si vous désirez suivre un programme télévisé, ou bien regarder un DVD ou une cassette vidéo, vous devez relier notre émetteur audio-vidéo EN1557 à la prise PERITEL du décodeur, du lecteur DVD ou du magnétoscope, comme le montre la figure.

réaction reçoit le 5 V dont il a besoin de la sortie du régulateur IC1 78L05.

La réalisation pratique

Rien de plus facile que de mener à bien la construction de ces deux modules HF puisque, aussi bien le TX que le RX ont leur self déjà imprimée sur le circuit (L1).

Les deux platines se ressemblent assez, comme le montre la figure 17 et vous pouvez envisager de les monter de

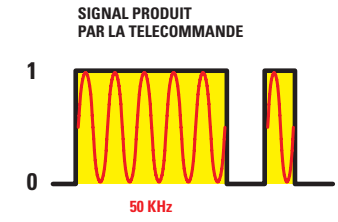


Figure 5 : La télécommande du téléviseur engendre un signal infra-rouge (IR) codé en une forme numérique (on module une porteuse d'environ 50 kHz).

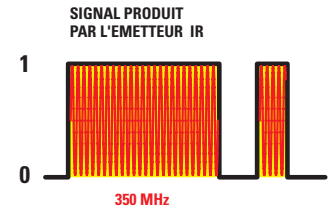
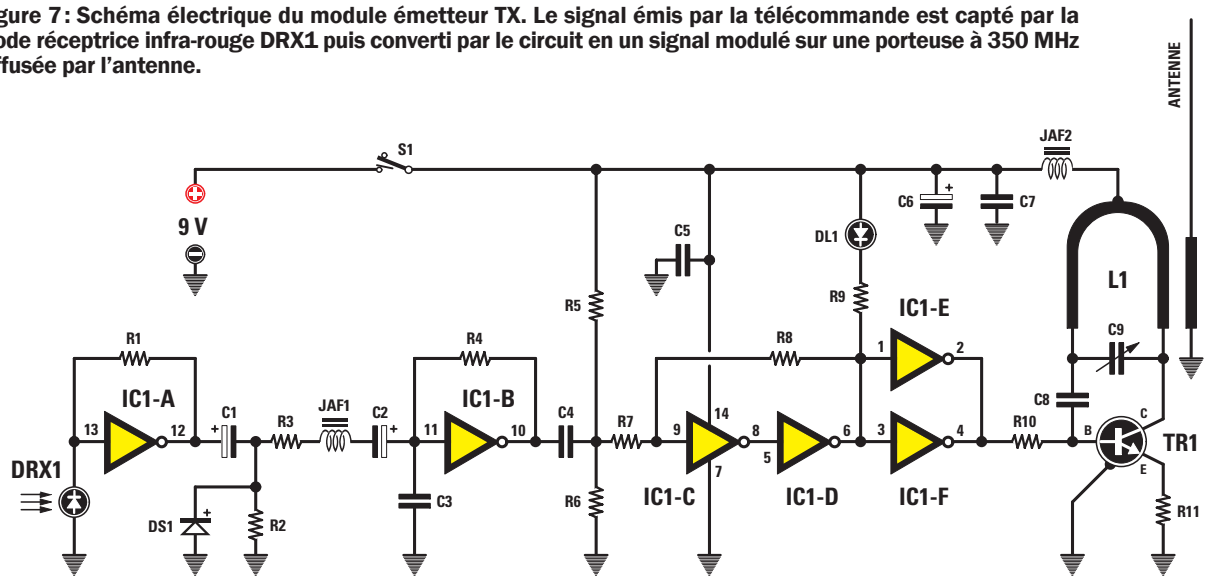


Figure 6 : Le signal engendré par la télécommande est converti par le module émetteur TX en un signal identique modulant cette fois une porteuse radio VHF de 350 MHz.

arquié composants
Rue de écoles 82600 Saint-Sardos France
Tél. 05 63 64 46 91 Fax 05 63 64 38 39
SUR INTERNET http://www.arquie.fr/
e-mail : arquie-composants@wanadoo.fr
Catalogue N°63
Afficheurs.
Alimentations.
Caméras. Capteurs.
Cartes à puces.
Circuits imprimés.
Circuits intégrés.
Coffrets. Condensateurs.
Cellules solaires
Connectique.
Diodes. Fers à souder.
Interrupteurs.
Kits. LEDs.
Microcontrôleurs.
Multimètres.
Oscilloscopes. Outillage.
Programmateurs.
Quartz. Relais.
Résistances. Transformateurs.
Transistors. Visserie.
Etc...
COMPOSANTS ELECTRONIQUES
Nouveau catalogue N°63
BON pour CATALOGUE FRANCE: GRATUIT (3.00 €.pour: DOM, TOM, UE et autres pays)
Nom:.....Prénom:.....
Adresse:.....
.....
Code Postal:..... Ville:.....
ELM

Déjà un nouveau standard !
la chaîne complète de CAO 100% français
Winschem
Saisie de schémas
WinECAD
Simulateur
Wintypon
Routage de la carte
Essayez les !
NOUVEAU
Tygra
Usinage de la carte
■ Import direct du typon fait avec Wintypon
■ Pilotage direct des fraiseuses UPA
■ Choix des méthodes d'approximation des arrondis
■ Génération ISO G-Code optimum
démo téléchargeable sur : www.micrelec.fr/cao
MICRELEC
4, place Abel Leblanc - 77120 Coulommiers
tel : 01 64 65 04 50 - Fax : 01 64 03 41 47

Figure 7 : Schéma électrique du module émetteur TX. Le signal émis par la télécommande est capté par la diode réceptrice infra-rouge DRX1 puis converti par le circuit en un signal modulé sur une porteuse à 350 MHz diffusée par l'antenne.



concert...Nous allons cependant quant à nous en décrire le montage une platine après l'autre.

L'émetteur TX

Quand vous vous êtes procuré –ou que vous avez réalisé– le petit circuit imprimé simple face dont la figure 13b donne le dessin à l'échelle 1: 1, montez tout d'abord le support de circuit intégré et vérifiez vos soudures (ni court-circuit entre pistes ou pastilles ni soudure froide collée) et poursuivez par les résistances, la diode (bague vers C1), la self bobinée JAF2, la self moulée JAF1, les condensateurs céramiques, les polyesters et les électrolytiques (attention à la polarité de ces derniers), le transistor (ergot repère-détrompeur vers R11), le

Liste des composants

R1 ..... 150 k
R2 ..... 10 k
R3 ..... 10 k
R4 ..... 100 k
R5 ..... 180 k
R6 ..... 330 k
R7 ..... 47 k
R8 ..... 1 M
R9 ..... 1 k
R10 ... 10 k
R11 ... 39

C1..... 10 µF électrolytique
C2..... 10 µF électrolytique
C3..... 10 nF polyester
C4..... 1 µF polyester
C5..... 100 nF polyester
C6..... 10 µF électrolytique
C7..... 10 nF céramique
C8..... 4,7 pF céramique
C9..... ajustable 1,2-6 pF

JAF1 .. self 47 mH
JAF2 .. self de choc 10 µH
L1..... self "strip-line" (ligne imprimée)

XTAL .. quartz 4 MHz

DS1 ... 1N4148
DL1 ... LED
DRX1. diode IR RX BPW41

TR1.... NPN 2N918

IC1..... CD4069

S1..... interrupteur

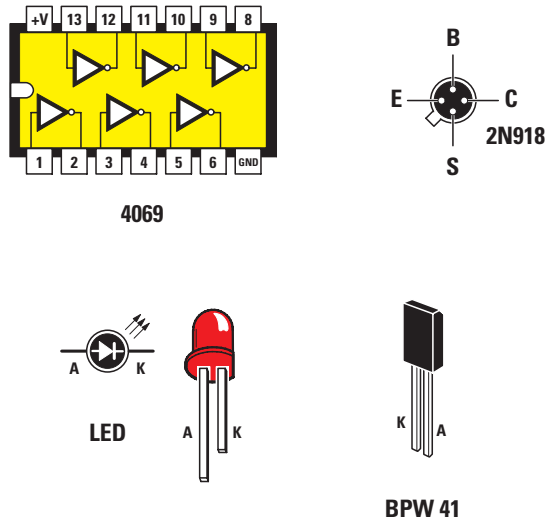


Figure 8 : Brochages du CD4069 vu de dessus, du 2N918 vu de dessous, de la LED vue de face et de la diode réceptrice IR vue de la face photosensible (celle qui ne comporte pas de marquage).

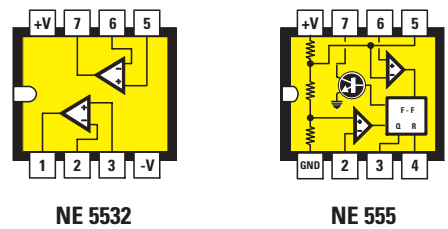


Figure 9 : Brochages du NE5532 et du NE555 vus de dessus et repère-détrompeurs en U vers la gauche.

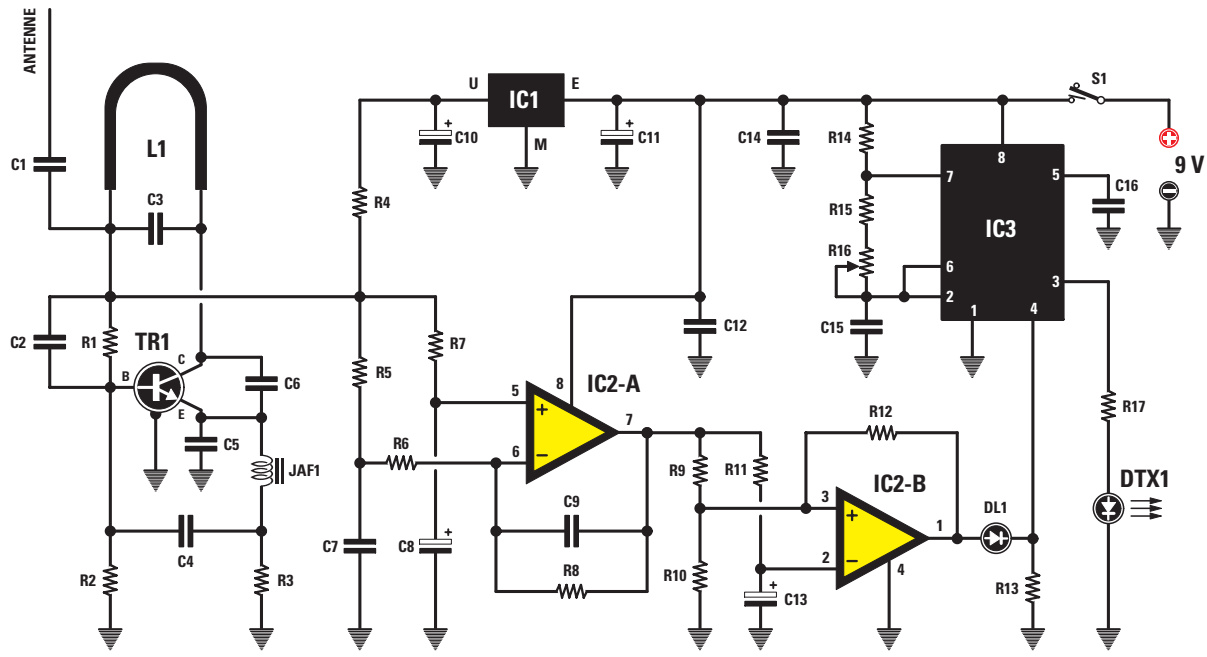


Figure 10 : Schéma électrique du module récepteur RX. Le signal à 350 MHz est détecté par le circuit à super-réaction formé de TR1, L1 et C2, C3, C5 et C6 puis concerti en un signal IR modulé sur une porteuse comprise entre 12 et 68 kHz, capable d'actionner tout appareil équipé d'un système de télécommande infra-rouge.

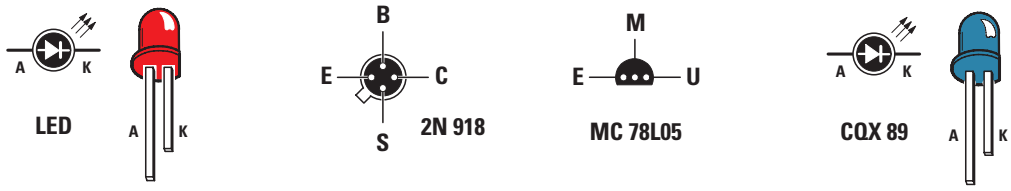


Figure 11 : Brochages de la LED vue de face, du NPN 2N918 et du régulateur vus de dessous et de la diode émettrice IR vue de face comme la LED.

condensateur ajustable C9, la diode IR DRX1 (face non marquée vers l'extérieur de la platine), la LED DL1 (anode –broche la plus longue– vers C7), les fils de la prise de pile 6F22 (rouge au +, noir au –9 V) et enfin l'interrupteur à glissière S1. L'antenne se soude côté cuivre, comme vous pouvez le voirsur la figure 13a.

Une fois la dernière soudure effectuée, insérez IC1 dans son support, repère-détrompeur en U vers C6.

Avec l'aide des figures 13a et 14 (plus la liste des composants) vous vous en sortirez très bien.

Le récepteur TX

Quand vous vous êtes procuré –ou que vous avez réalisé– le circuit imprimé simple face (un peu plus grand que le précédent) dont la figure 15b donne le dessin à l'échelle 1:1, montez tout d'abord les deux supports de circuits

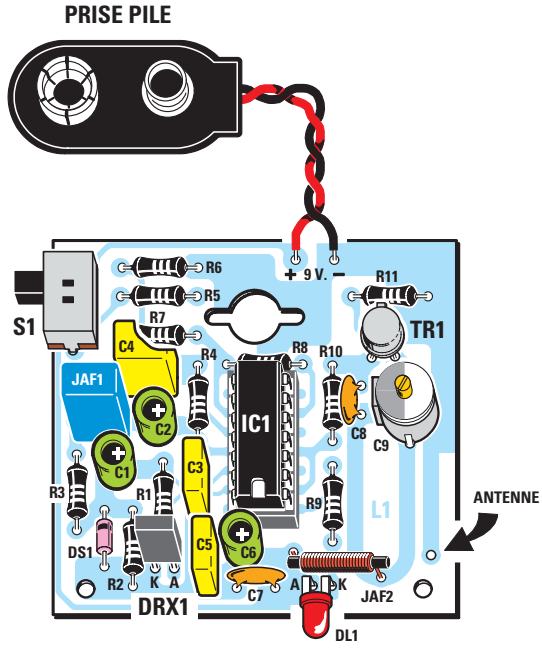


Figure 12 : Au moment de monter la diode réceptrice IR BPW41, prenez soin de tourner vers l'extérieur sa face photosensible (c'est-à-dire celle qui n'est pas marquée).

Liste des composants

R1 ..... 10 k
R2 ..... 47 k
R3 ..... 2,2 k
R4 ..... 1 k
R5 ..... 10 k
R6 ..... 12 k
R7 ..... 22 k
R8 ..... 4,7 M
R9 ..... 10 k
R10 ... 470 k
R11 ... 10 k
R12 ... 2,2 M
R13 ... 1 k
R14 ... 1 k
R15 ... 10 k
R16 ... 50 k
R17 ... 100

C1..... 1,5 pF céramique
C2..... 1 nF céramique
C3..... 4,7 pF céramique
C4..... 1 nF céramique
C5..... 4,7 pF céramique
C6..... 1,5 pF céramique
C7..... 1 nF céramique
C8..... 10 µF électrolytique
C9..... 2,2 pF céramique
C10 ... 10 µF électrolytique
C11 ... 10 µF électrolytique
C12 ... 100 nF polyester
C13 ... 10 µF électrolytique
C14 ... 100 nF polyester
C15 ... 1 nF polyester
C16 ... 100 nF polyester

JAF1 .. self 1 µH
L1 ..... self "strip-line" (ligne imprimée)

DL1 ... LED
DTX1 . diode IR TX CQX89

TR1.... NPN 2N918

IC1..... 78L05
IC2..... NE5532
IC3..... NE555

S1..... interrupteur

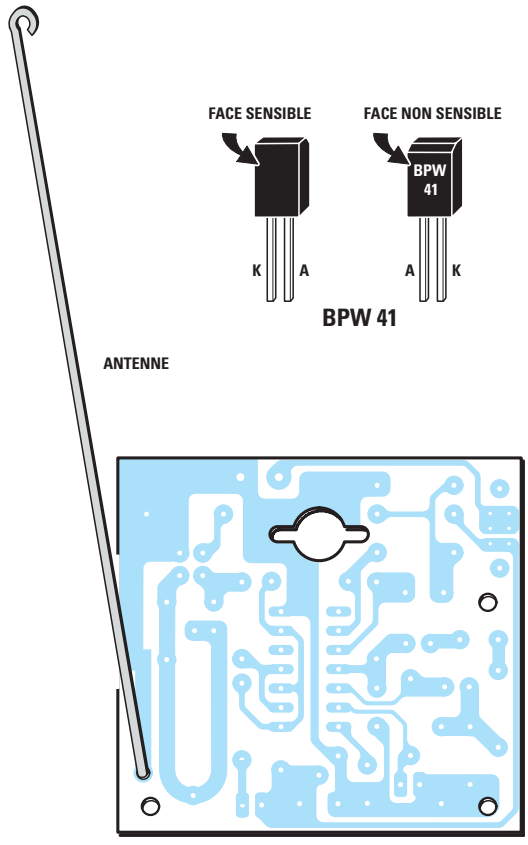


Figure 13a : Schéma d'implantation des composants du module émetteur TX vu du côté des composants et du côté des soudures. Notez, sur ce dernier côté, le point d'insertion de l'antenne.

intégrés et vérifiez vos soudures (ni court-circuit entre pistes ou pastilles ni soudure froide collée).

Poursuivez par les résistances, la self moulée JAF1, les condensateurs céramiques, les polyesters et les électrolytiques (attention à la polarité de ces derniers), le transistor (ergot repère-détrompeur vers C5), le régulateur IC1 (méplat repère-détrompeur vers C11), le trimmer R16, la diode IR DTX1 (anode –broche la plus longue– vers la gauche), les fils de la prise de pile 6F22 (rouge au +, noir au –9 V) et enfin l'interrupteur à glissière S1.

L'antenne se soude côté cuivre et la LED DL1 également (anode –broche la plus longue– vers l'antenne), comme le montre la figure 15a.

Une fois la dernière soudure effectuée, insérez IC2 et IC3 dans leurs

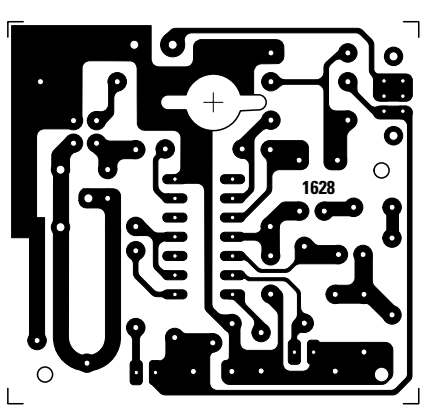


Figure 13b : Dessin, à l'échelle 1:1, du circuit imprimé du module émetteur TX.

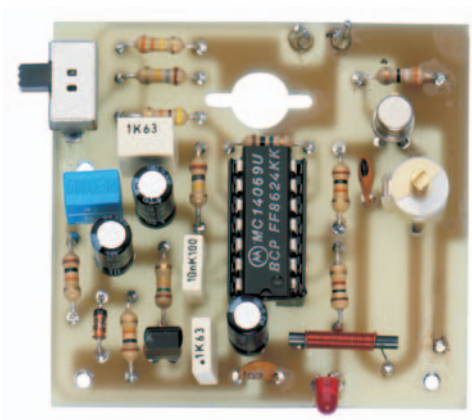


Figure 14 : Photo d'un des prototypes du module émetteur TX. On voit le condensateur ajustable C9 et la ligne inductive (self imprimée) L1.

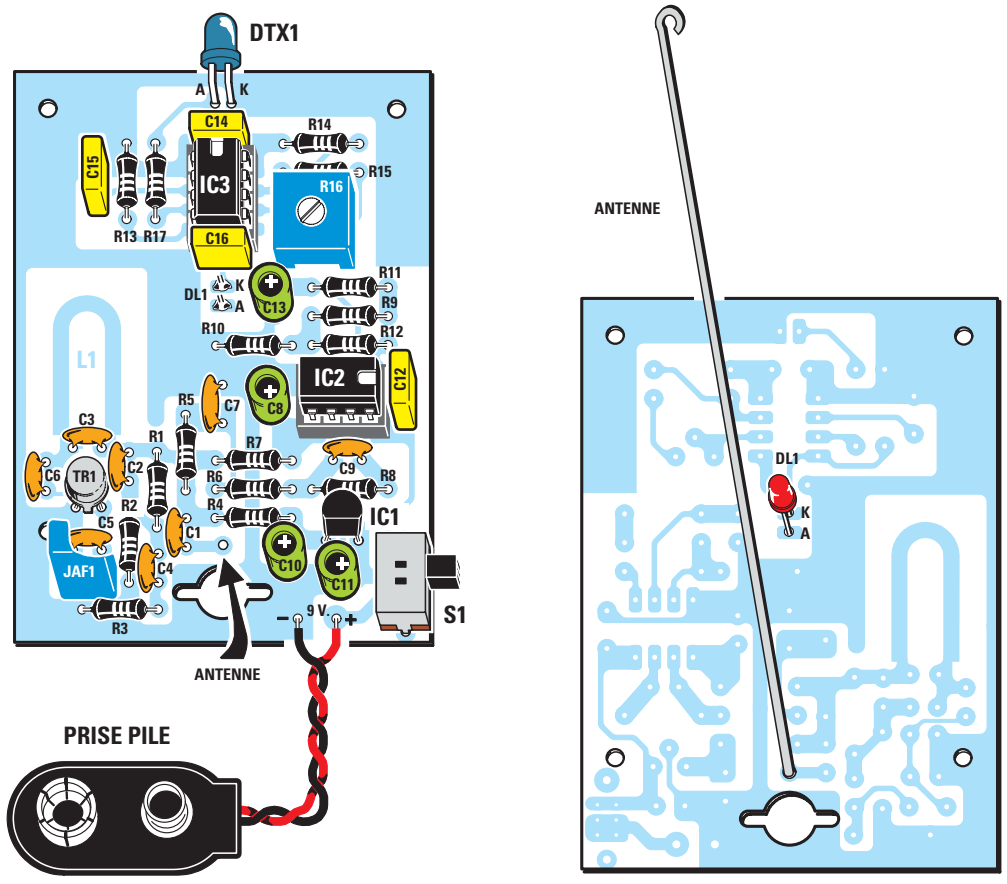


Figure 15a : Schéma d'implantation des composants du module récepteur RX vu du côté des composants et du côté des soudures. Notez, sur ce dernier côté, le point d'insertion de l'antenne.

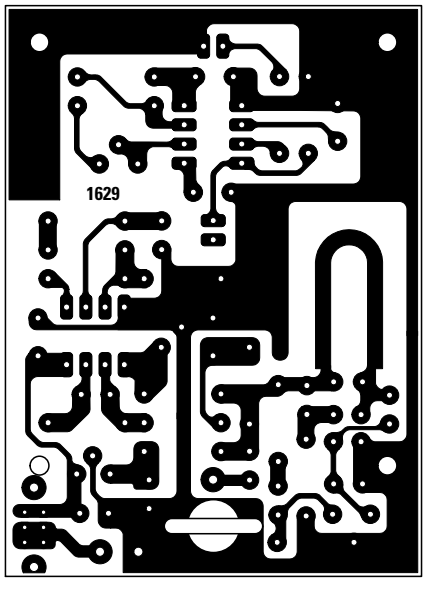


Figure 15b : Dessin, à l'échelle 1:1, du circuit imprimé du module récepteur RX.

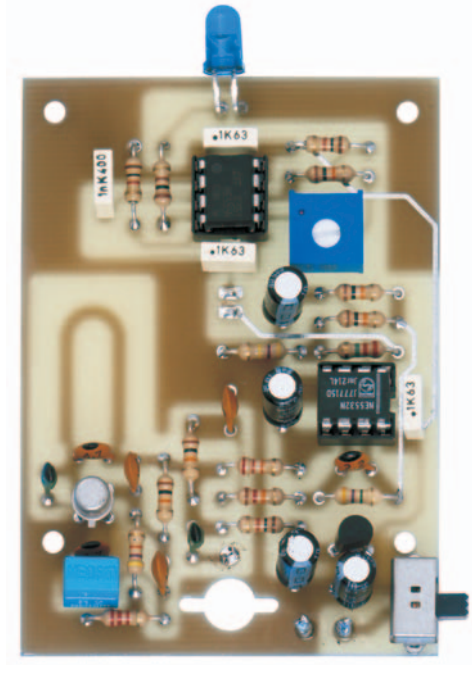


Figure 16 : Photo d'un des prototypes du module récepteur RX. On voit le potentiomètre R16 servant à régler la fréquence du signal émis par la diode IR DTX1 et la ligne inductive (self imprimée) L1.

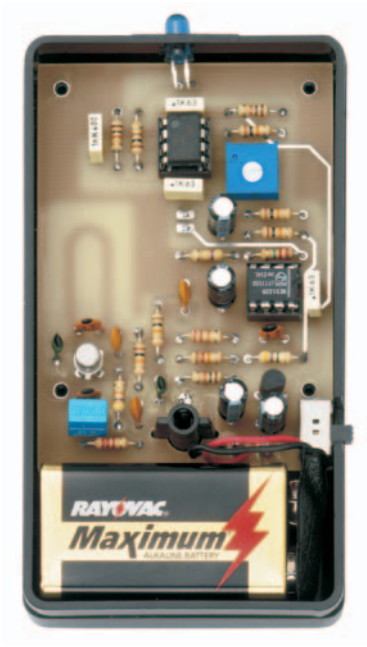




Figure 17 : Installation dans leur boîtier plastique respectifs du module émetteur TX (c'est la plus petit) et du module récepteur RX, chacun étant alimenté par une pile 6F22 de 9 V.

supports, repère-détrompeurs en U respectivement vers C12 et C14. Avec l'aide des figures 15a et 16 (plus la liste des composants) vous vous en sortirez très bien.

L'installation dans les boîtiers

Là encore, avec la figure 17, vous aurez du mal à vous tromper ! Dans les deux boîtiers plastiques spécifiques les platines se fixent au fond au moyen d'un axe en saillie.
Les côtés sont percés pour le passage des LED de signalisation et des interrupteurs (TX et RX), pour celui de la diode IR réceptrice (TX) et de la diode IR émettrice (RX). Pour les deux modules, un emplacement est prévu pour la pile 6F22 de 9 V.

Les essais et les réglages

Pour régler vos deux modules, TX et RX, suivez la procédure qui suit :
- Réglez le trimmer R16 du RX à mi course.
- Placez les modules TX et RX côte à côte.
- Prenez la télécommande IR que vous utilisez, par exemple, pour actionner le décodeur et pointez-la vers le module TX. Pressez un bouton (par exemple celui de mise en marche). La LED du module TX clignote pour confirmer que le signal de la télécommande IR est bien reçu par le TX.
- Continuez à actionner la télécommande de manière répétitive et en même temps tournez le condensateur ajustable C9 du TX jusqu'à voir clignoter de façon parfaitement synchrone les LED DL1 du TX et du RX.
Cela confirme la réception correcte du signal par le RX
- Placez les modules à environ 1 m de l'appareil à commander (par exemple le décodeur), en ayant soin de placer le RX en face de l'appareil pour que le rayon infra-rouge du RX atteigne bien cet appareil.
- Actionnez à nouveau la télécommande et vérifiez que l'appareil est bien commandé (par exemple que le décodeur s'allume).
Faites des essais avec plusieurs boutons de la télécommande et vérifiez que chaque fois la commande correspondante a bien été reçue par l'appareil. Sinon, retouchez R16.
- Laissez le RX où il se trouve et éloignez-vous progressivement avec la télécommande et le TX, tout en continuant à contrôler le bon fonctionnement du système.
- Enfin, réglez la fréquence de la porteuse du module RX et, pour cela, mettez-vous dans une position plus critique, comme le montre la figure 18 : désalignez quelque peu le RX et l'appareil à commander.
- Rapprochez le module TX du module RX et actionnez la télécommande. Vérifiez que la commande continue à être reçue par l'appareil.
Si ce n'est pas le cas, retouchez R16 jusqu'à l'amélioration de l'action sur l'appareil commandé.Quand tout cela est fait, remettez le RX face à l'appareil à commander.

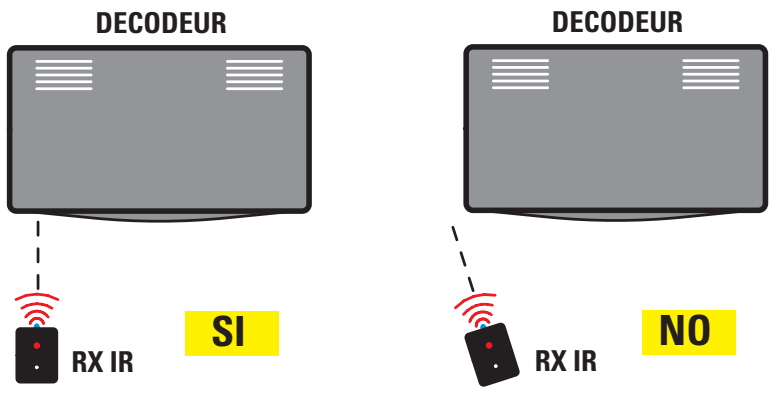


Figure 18 : Pour régler la fréquence de la porteuse du module récepteur, vous devez désaligner légèrement ce module par rapport à l'appareil à commander (ici par exemple un décodeur) puis régler le trimmer R16 jusqu'à obtenir, même dans cette condition critique, une commande correcte de l'appareil.

Conclusion

Si vous souhaitez actionner aussi d'autres appareils, comme un lecteur de DVD ou un magnétoscope, vous devrez les empiler comme le montrent les figures 2 et 4.

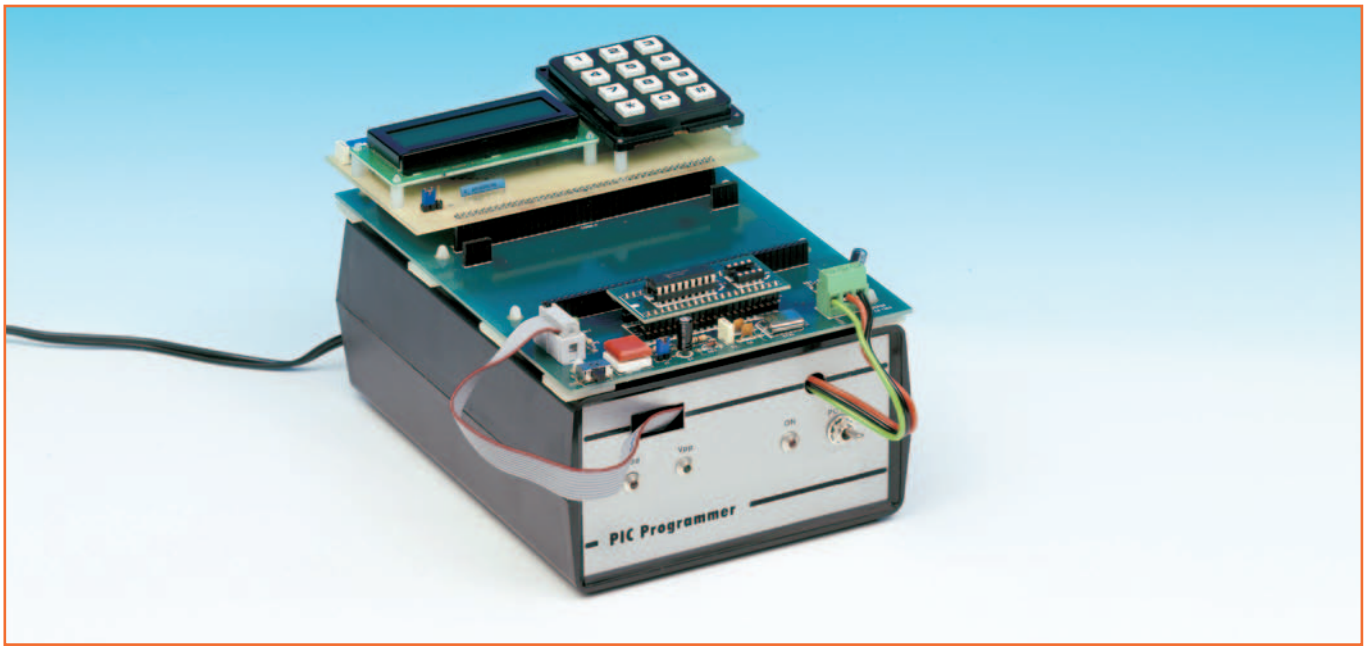
Placez bien le RX en face de cet empilement pour que le faisceau IR atteigne facilement les appareils.

Comment construire ce montage ?

Tout le matériel nécessaire pour construire ce répéteur HF pour télécommande TX et RX EN1628-1629 est disponible chez certains de nos annonceurs. Voir les publicités dans la revue.
Les typons des circuits imprimés et les programmes disponibles sont téléchargeables à l'adresse suivante :
http://www.electronique-magazine.com/circuitrevue/086.zip. ◆

Une nouvelle platine d'expérimentation pour PIC

Cette nouvelle platine d'expérimentation est un outil (ludique !) indispensable pour quiconque souhaite se perfectionner dans la mise en œuvre des microcontrôleurs PIC. Elle permet d'utiliser un clavier numérique (comme on en trouve sur les téléphones) et un afficheur LCD (tout aussi numérique) –composant désormais incontournable au labo d'électronique pour effectuer toutes les mesures et tous les contrôles voulus.



Tout d'abord un rappel : notre programmateur de PIC, en mesure de programmer les microcontrôleurs PIC 12F675, 16F628, 16F876 et 16F877, se compose du programmateur EN1580 proprement dit, du bus EN1581 et de l'alimentation EN1203. S'y ajoutent les platines d'expérimentation EN1582 (c'est la première proposée, elle fait partie intégrante du programmateur), EN1583 (platine à quatre relais), EN1584 (platine à quatre TRIAC et un darlington)... et voici aujourd'hui la toute dernière : la platine à clavier et afficheur LCD EN1585.

Le clavier est à douze touches et l'afficheur est un LCD alphanumérique. Nous avons choisi cette fois de vous proposer une platine d'expérimentation à clavier et afficheur parce que c'est le mode de communication le plus utilisé entre l'homme et les appareils électroniques. Il est en effet devenu habituel de se servir d'un LCD pour visualiser les données résultant d'une mesure (nous pensons par exemple aux instruments de mesure ou aux appareils électromédicaux –comme le générateur d'ultrasons– ou aux convertisseurs de signaux ou aux exciteurs FM ou à tant d'autres instruments encore, proposés par votre revue et que vous avez peut-être construits).

Mais il est vrai que nous pensons surtout aux dispositifs dont l'utilisation ne requiert aucune préparation professionnelle de la part de l'usager : les téléphones fixes ou mobiles, les guichets DAB, ou les machines à boisson, ainsi qu'une foule innombrable d'autres appareils privés ou publics ayant envahi notre quotidien, sont dotés de claviers et d'afficheurs. Il est donc important, pour qui veut se former en électronique –et c'est le cas de nos lecteurs, même les plus chevronnés– d'apprendre à conjuguer les microcontrôleurs et leurs programmes résidents avec ces claviers et afficheurs LCD. Ceci dit, nous pouvons maintenant passer au schéma éléctrique

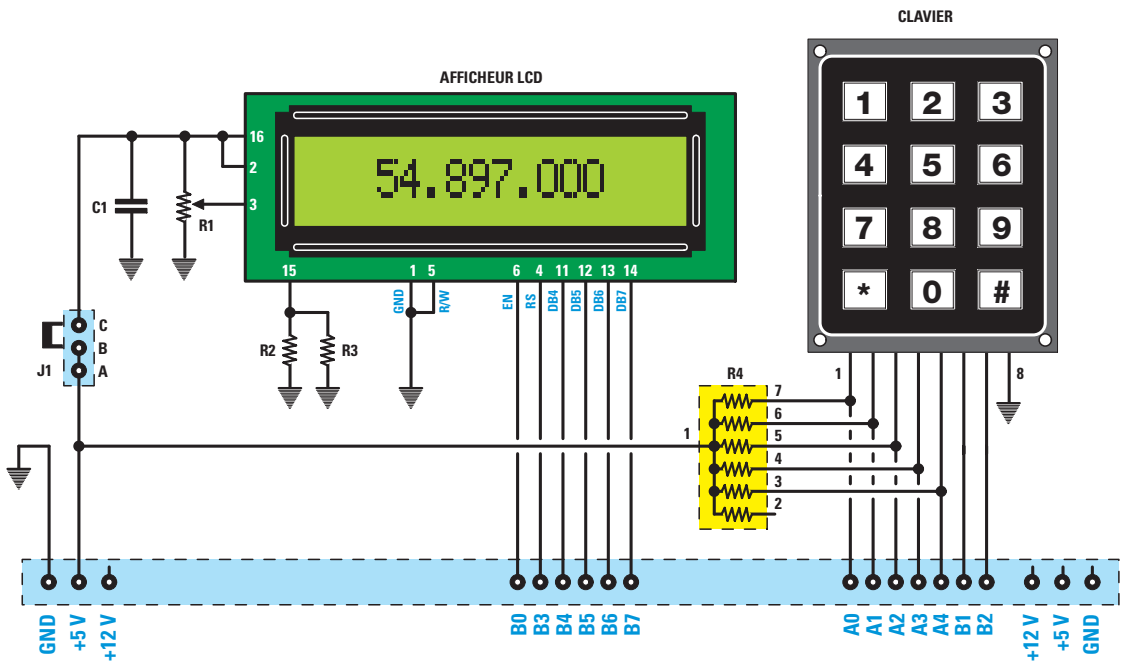


Figure 1 : Schéma électrique de la platine d'expérimentation à clavier et afficheur LCD pour PIC EN1585. Avec le trimmer R1 relié à la broche 3 du LCD, on peut régler le contraste de façon à optimiser l'affichage pour faciliter la lecture. Le réseau de résistances R4 constitue les résistances de tirage des broches du port A du PIC (A0 à A4).

Le schéma électrique

L'afficheur LCD

Nous avons choisi un afficheur LCD modèle CMC116L01 de la marque CCT : nous le connaissons bien et vous le retrouvez dans plusieurs de nos montages, comme ce fut le cas avec le Fréquencemètre 2,2 GHz EN1572. Cet afficheur a un rapport qualité/prix optimal. Parmi les composants externes nécessaires à son fonctionnement, vous avez le condensateur polyester C1 inséré dans la ligne d'alimentation (entre les broches 2-16 et la masse) qui sert de filtre en acheminant à la masse d'éventuelles perturbations (voir figure 1, le schéma électrique); le trimmer R1 dont le curseur est relié à la broche 3 permettant de régler le contraste du LCD afin d'améliorer l'affichage ; enfin R2 et R3 (15 ohms 1/2 W chacune) reliées à la broche 15 et servant à limiter le courant arrivant à la barre de LED qui fournit le rétro-éclairage.

Sur les huit broches concernant les données à envoyer au LCD (DB0 à DB7, figure 2), nous n'en utilisons que quatre : les quatre autres ne sont pas connectées. Cette configuration nous permet d'économiser quatre lignes d'E / S du PIC.

Le mode à quatre bits est indiqué par tous les manuels, bien que ceux-ci oublient de préciser que, pour ce modèle en tout cas, les quatre broches restantes ne doivent pas être reliées à la masse mais demeurer non connectées. Ce choix est presque obligatoire quand on monte un PIC à peu de broches et nous avons souhaité vous montrer comment mettre en œuvre cette solution (permettant, répétons-le, de "gagner" des lignes d'E / S)

Le clavier

Sur la platine nous avons en outre monté un clavier à matrice relié au PIC au moyen du connecteur CONNA. Le réseau résistif R4, relié entre le 5 V et les broches A0 à A4 du microcontrôleur PIC16F628, constitue la résistance de tirage ("pull-up" = maintien du niveau logique haut). Si vous consultez la table des caractéristiques ("datasheet") de ce PIC, vous verrez que les broches du port A n'ont pas de résistance de tirage interne ; il faut donc en prévoir une externe (une par broche, d'où le réseau R4 figure 1) et on peut ainsi utiliser ces broches comme entrées "tirées".

Note : comme le montre la figure 1, une des résistances du réseau, celle aboutissant à la sortie 2, n'est pas utilisée.

Le cavalier **J1**, relié entre le 5 V et les broches 2-16 du LCD, est normalement fermé (sur CB). Si on souhaite n'utiliser que le clavier, il faut le laisser ouvert (positionné sur AB) afin de couper l'alimentation de l'afficheur.

Notes sur l'afficheur LCD

La figure 2 donne le brochage du LCD CMC116L01 et la description de la fonction de chaque broche. Les LCD comportent un circuit intégré interne servant de contrôleur et de pilote pour gérer chaque caractère : ce circuit intégré ne fait pratiquement rien d'autre que de lire les données et les commandes que le PIC lui envoie, les interpréter, sélectionner les symboles à visualiser et modifier le mode de fonctionnement de l'afficheur.

Il possède une table de référence interne (voir le Tableau des caractères) avec laquelle on détermine, en fonction de la valeur binaire envoyée, l'écriture d'un caractère à la place d'un autre. Dans cette table, 248 éléments contiennent des caractères déjà définis constituant la DDRAM (Data Display RAM) ; 8 éléments sont en revanche personnalisables et peuvent contenir des caractères produits par l'usager. Ce sont cette fois des CGRAM (Character Generator RAM). Ces codes se trouvent dans les caractéristiques de l'afficheur LCD sous forme d'une grosse matrice : la première partie du nombre binaire est en haut et les quatre autres bits restants sont reportés sur un côté de la matrice. Comme le montre le Tableau des caractères, la lettre majuscule L, par exemple, est visualisée si nous envoyons au circuit intégré la combinaison binaire 0100 1100.

TABLEAU DES CARACTERES

0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111	
32	48	64	80	96	112	160	176	192	208	224	240	xxxx0000
33	49	65	81	97	113	161	177	193	209	225	241	xxxx0001
34	50	66	82	98	114	162	178	194	210	226	242	xxxx0010
35	51	67	83	99	115	163	179	195	211	227	243	xxxx0011
36	52	68	84	100	116	164	180	196	212	228	244	xxxx0100
37	53	69	85	101	117	165	181	197	213	229	245	xxxx0101
38	54	70	86	102	118	166	182	198	214	230	246	xxxx0110
39	55	71	87	103	119	167	183	199	215	231	247	xxxx0111
40	56	72	88	104	120	168	184	200	216	232	248	xxxx1000
41	57	73	89	105	121	169	185	201	217	233	249	xxxx1001
42	58	74	90	106	122	170	186	202	218	234	250	xxxx1010
43	59	75	91	107	123	171	187	203	219	235	251	xxxx1011
44	60	76	92	108	124	172	188	204	220	236	252	xxxx1100
45	61	77	93	109	125	173	189	205	221	237	253	xxxx1101
46	62	78	94	110	126	174	190	206	222	238	254	xxxx1110
47	63	79	95	111	127	175	191	207	223	239	255	xxxx1111

Afin d'éviter toute confusion, plusieurs constructeurs s'accordent pour standardiser ce codage des afficheurs alphanumériques.

Des contrôleurs compatibles ont donc été conçus : le HD44780 de Hitachi, le KS0066 de Samsung et le MSM6222 de OKI en font partie. Les pilotes HD44780 et KS0066 (ce dernier utilisé dans le LCD CMC116) sont à peu près identiques et les petites différences entre eux ne concernent que les temporisations des signaux de contrôle et des données.

L'afficheur choisi comporte une seule ligne de 16 caractères ; mais on doit le piloter comme un afficheur à deux lignes de 8 caractères chacune (voir figure 2).
Si on observe cet afficheur à la loupe lorsqu'il est illuminé, on voit que chaque caractère est dessiné à l'intérieur d'un groupe de carreaux : chacun de ces groupes est une matrice et chaque carreau est un pixel. Les pixels, constitués d'une substance devenant sombre au passage d'un courant, sont disposés en une matrice de 5 x 8, ce qui fait 40 pixels par caractère. Sur les 40, seuls 35 (soit 5 x 7) sont utilisés pour visualiser les caractères alphanumériques ; les 5 restants de la dernière ligne sont destinés à la gestion et à la visualisation du curseur. Regardez bien le Tableau : vous voyez qu'en effet chaque caractère est constitué exclusivement d'une matrice de 5 x 7 pixels. Si vous envoyez en même temps un nombre de caractères supérieur à seize, ils seront perdus : donc, faites bien attention à la longueur du texte, lequel ne doit pas dépasser huit caractères par secteur.

Description du clavier à matrice

Le clavier utilisé est semblable à celui d'un téléphone et sa gestion se fait par une matrice interne qui lui donne son nom de clavier à matrice. Ces claviers à matrice sont surtout utilisés en raison de la simplicité de leur construction et de leur contrôle (leur coût est de plus assez faible).

Mais pourquoi clavier à "matrice" ? Une matrice est une table d'éléments disposés sur des lignes horizontales et des colonnes verticales. Pour nous ici la matrice comporte douze carreaux soit quatre lignes et trois colonnes ; chaque carreau est un poussoir. Comme le montre la figure 4, chaque poussoir est relié à deux des huit broches du clavier.

Quand les poussoirs ne sont pas pressés, les niveaux logiques des broches 1 à 7 de la matrice sont au un logique sous l'effet des résistances de tirage ; seule la 8 (à laquelle sont reliés tous les poussoirs) est à la masse.

Quand, au contraire on presse un poussoir, le contact acheminant la masse se ferme sur les deux extrémités de chaque poussoir. Le PIC effectue la lecture et, en fonction de la combinaison binaire, il détecte quelles broches sont à la masse et par conséquent quel poussoir a été pressé. Un fonctionnement, somme toute, assez simple et très efficace.

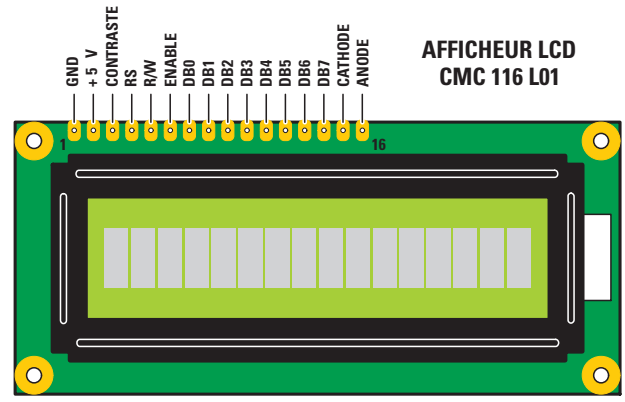
Le clavier adopté est le plus élémentaire du point de vue, justement, du fonctionnement ; cela permet une grande souplesse d'adaptation à des besoins de tous types. Avec ce modèle en effet, on a en sortie une donnée différente pour chaque poussoir pressé (voir le tableau de la figure 4).

La réalisation pratique

Pour réaliser ce programmateur, il vous faut le circuit imprimé EN1585 : c'est un double face à trous métallisés dont la figure 5b-1 et 2 donne les dessins à l'échelle 1. Le cuivre du côté composants se limitant aux pastilles du connecteur CONNA, si vous réalisez le circuit imprimé vous-même, vous pouvez ne graver qu'un simple face (et ne pas souder, faute de pastilles, les broches de CONNA côté composants, mais seulement du côté soudures).

Quoi qu'il en soit, quand vous avez le circuit imprimé devant vous, montez

BROCHE	DESCRIPTION	
1	masse module	
2	alimentation module 5V	
3	contraste	
4	sél. registres	0 instruction
		1 données
5	Read-write	1 read
	(lis-écrit)	0 write
6	enable	
7-14	bus data DB0 - DB7	
15	rétro éclairage cathode –	
16	rétro éclairage anode +	



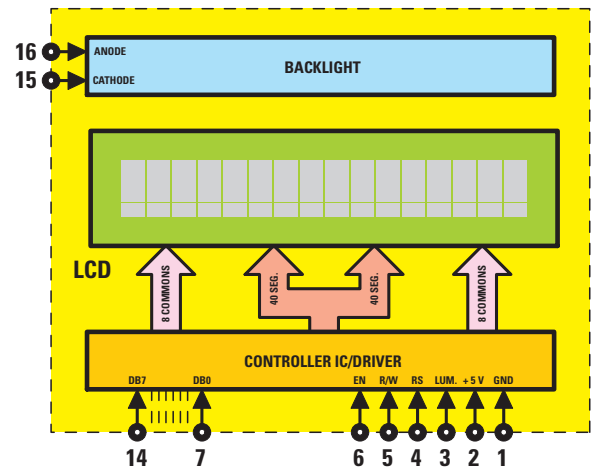


Figure 2: Brochage et schéma synoptique de l'afficheur LCD CMC116L01. Avec le programme SIM.asm contenu dans le CDR1585 en dotation, on visualise le jeu complet de caractères DDRAM et CGRAM du pilote visible dans le tableau.

Liste des composants

R1 ..... 10 k trimmer
R2 ..... 15 1/2 W
R3 ..... 15 1/2 W
R4 ..... réseau résistif 6 x 1 k

C1..... 100 nF polyester

AF ..... LCD CMC116L01
CLV ..... clavier 12 touches
CONNA..... connecteur 40 pôles mâle
CONN AF.... connecteur 16 pôles femelle
CONN CLV .. connecteur 8 pôles femelle
J1..... cavalier 3 pôles

Divers:

2 connecteurs barrettes à 4 pôles
6 entretoises plastiques

Sauf spécification contraire, toutes les résistances sont des 1/4 W à 5 %.

d'abord (côté soudures) le grand connecteur CONNA puis (toujours côté soudures) les deux barrettes à 4 broches (elles n'ont qu'un rôle mécanique, ne servant qu'à positionner cette platine sur le bus).

Le grand CONNA à 40 broches est destiné, comme le montre la photo de la première page, à insérer cette platine d'expérimentation dans la platine bus EN1581.

Ensuite, côté composants, montez les CONN AF et CONN CLV et le cavalier J1 (à trois broches).

Si vous observez bien les figures 5a et 6 et la liste des composants, vous n'aurez aucune difficulté à monter cette platine.

Accordez beaucoup d'attention aux soudures de ces différents connecteurs (ni court-circuit entre pistes ou pastilles ni soudure froide collée). Soudez alors le réseau de résistances (point repère-détrompeur en bas à gauche vers J1), les deux résistances, le trimmer et le condensateur polyester (pas de polarité). Mettez tout de suite le jumper de J1 sur CB (afin de ne pas oublier). Fixez sur cette platine les six entretoises plastiques.

Placez maintenant le clavier (à l'aide du connecteur Mâle/Femelle à 8 pôles et des deux entretoises métalliques.)

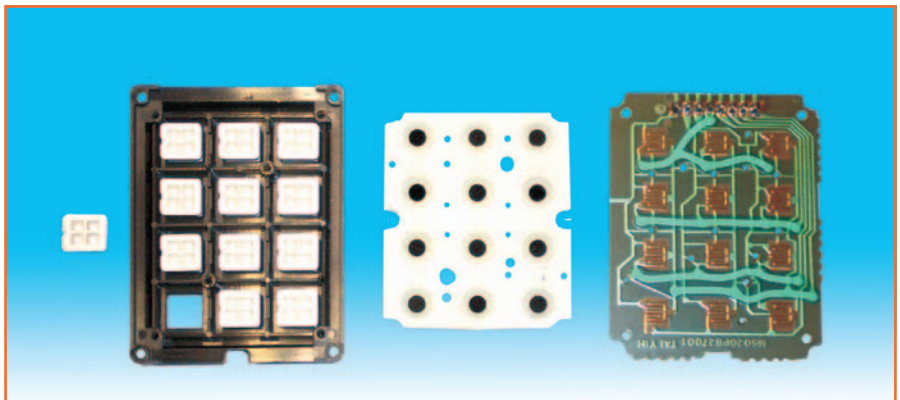
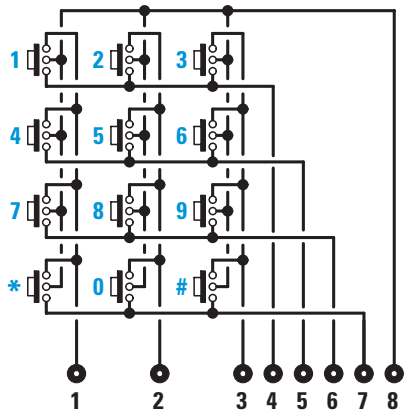


Figure 3: Afin de satisfaire votre légitime curiosité, nous avons ouvert le boîtier du clavier à matrice (comme le montre la figure 4, chaque poussoir ou touche est relié à deux des broches du clavier).



TOUCHE	BR 1	BR 2	BR 3	BR 4	BR 5	BR 6	BR 7
1	0	1	1	0	1	1	1
2	1	0	1	0	1	1	1
3	1	1	0	0	1	1	1
4	0	1	1	1	0	1	1
5	1	0	1	1	0	1	1
6	1	1	0	1	0	1	1
7	0	1	1	1	1	0	1
8	1	0	1	1	1	0	1
9	1	1	0	1	1	0	1
*	0	1	1	1	1	1	0
0	1	0	1	1	1	1	0
#	1	1	0	1	1	1	0

Figure 4 : Chaque poussoir (touche) du clavier à matrice est relié à deux broches. Quand une touche est pressée, seules deux des broches du clavier se portent au niveau logique bas (voir tableau ci-contre) et, en fonction de la combinaison binaire ainsi déterminée, le PIC détecte quelles broches sont à la masse et par conséquent quelle touche a été pressée ; il peut alors commander l'affichage sur le LCD du symbole correspondant.

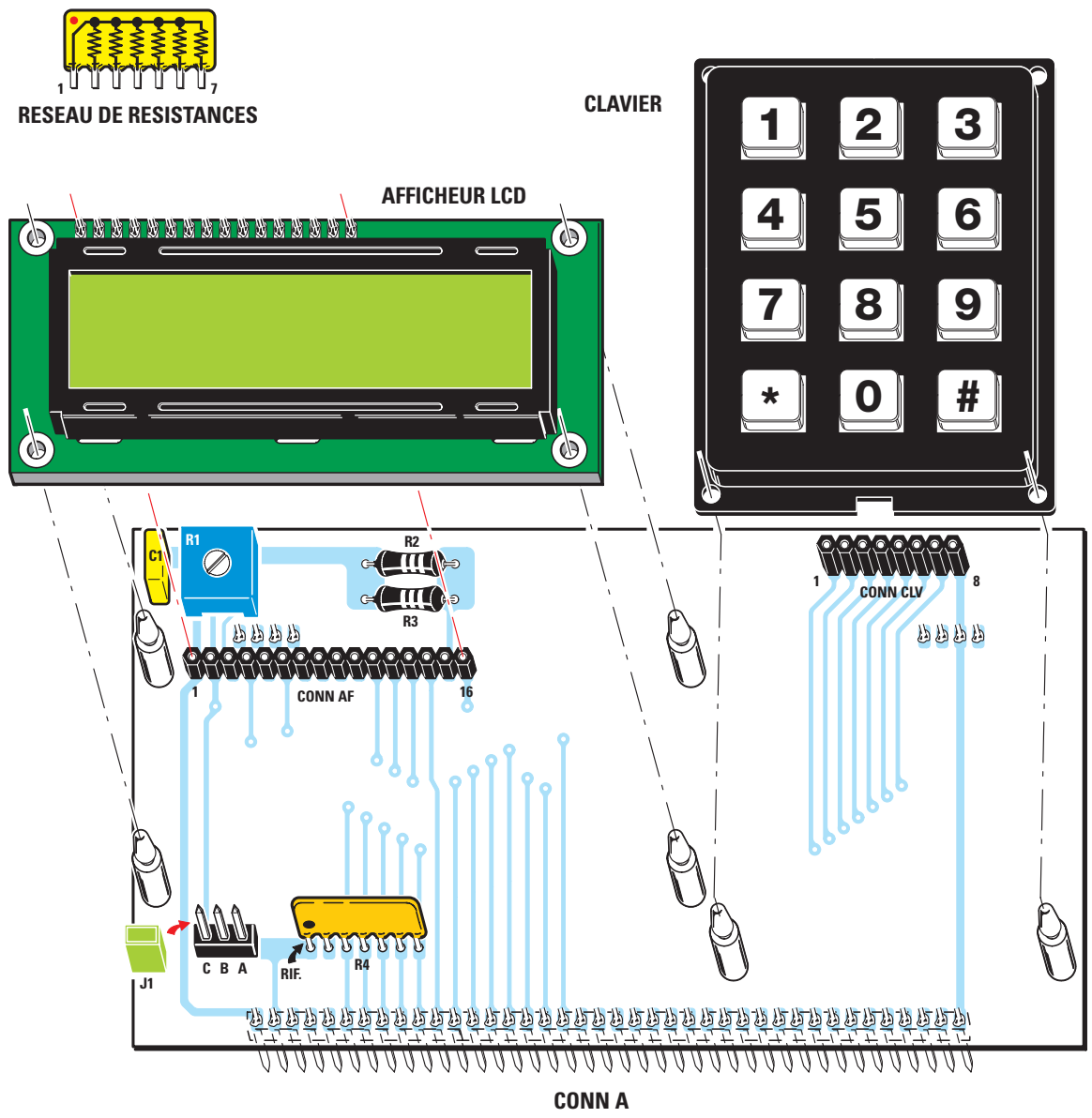


Figure 5a : Schéma d'implantation des composants de la platine d'expérimentation à clavier et afficheur LCD pour PIC EN1585. Seul composant à orienter correctement : le réseau de résistances R4 (point repère-détrompeur en bas à gauche vers J1). Dès que vous avez soudé le cavalier J1, mettez son jumper sur CB.

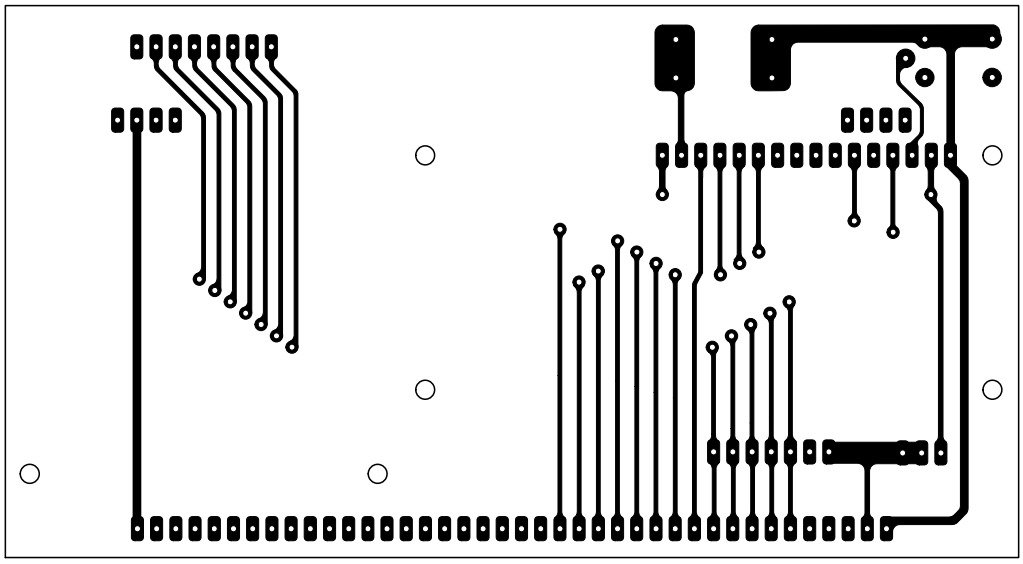


Figure 5b-1 : Dessin, à l'échelle 1, du circuit imprimé double face à trous métallisés de la platine d'expérimentation à clavier et afficheur LCD pour PIC EN1585, côté soudures.

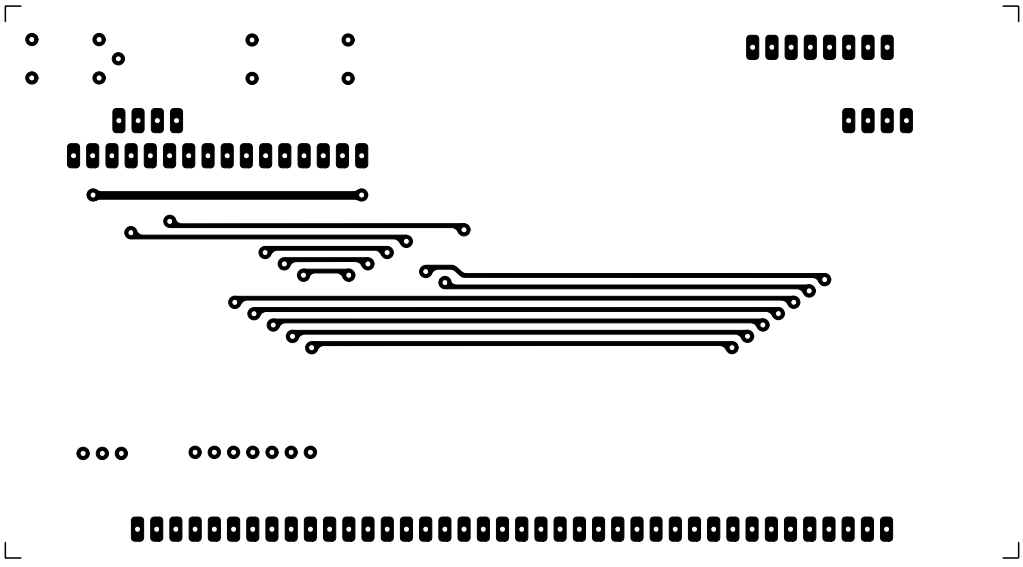


Figure 5b-2 : Dessin, à l'échelle 1, du circuit imprimé double face à trous métallisés de la platine d'expérimentation à clavier et afficheur LCD pour PIC EN1585, côté composants.

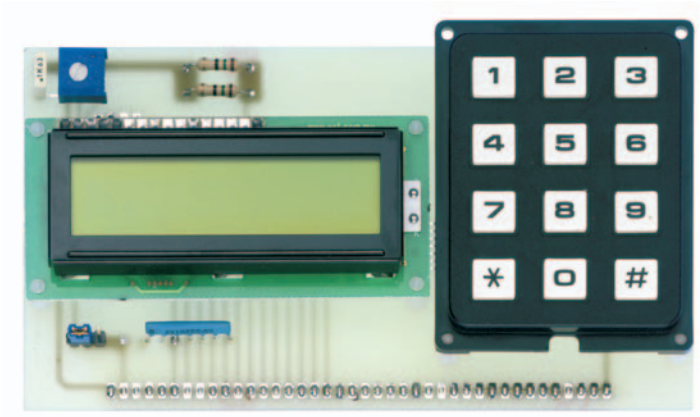


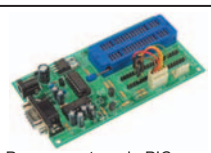
Figure 6 : Photo d'un des prototypes de la platine d'expérimentation à clavier et afficheur LCD pour PIC EN1585.

GO TRONIC

35ter, Route Nationale - B.P. 45
F-08110 BLAGNY (FRANCE)
E-mail: contacts@gotronic.fr
Tél.: 03.24.27.93.42
Fax: 03.24.27.93.50



Chargeur alimenté par USB pour 1 ou 2 R3 ou R6 NiMh.
Code: 09562 Prix : 19.95 €



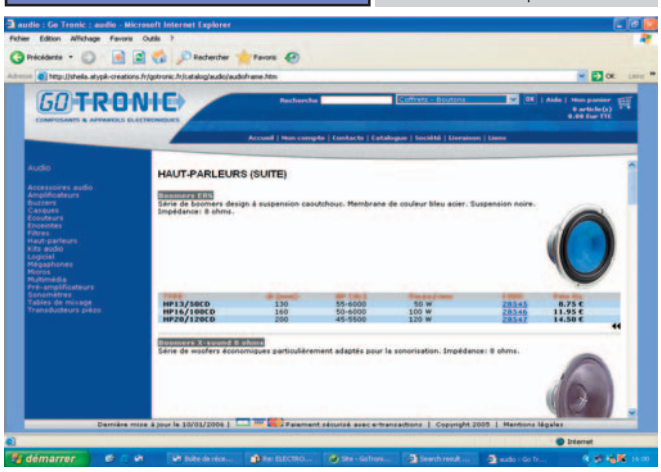
Programmateur de PIC avec support ZIF. Kit à souder. (nécessite alim 15Vcc/300mA)
Code: 24238 Prix : 39.95 €



Radar de recul à 4 capteurs étanches, indication sonore réglable et affichage LCD. La fréquence des bips augmente quand la distance entre le véhicule et l'obstacle diminue.
Code: 26707 Prix : 69.50 €

Consultez notre nouveau site
www.gotronic.fr

Port: 4.60 (ordinaire) ou 7.50 (colissimo)
Paiement: CB ou chèque à la commande



COMMENT FABRIQUER FACILEMENT VOS CIRCUITS IMPRIMES ?

Nouveau produit qui arrive tout droit des États-Unis et qui a révolutionné les méthodes de préparation des circuits imprimés réalisés en petites séries :

plus de sérigraphie grâce à une pellicule sur laquelle il suffit de photocopier ou d'imprimer le master...



ET-PNP5
Lot de 5 feuilles au format A4
18,75€

12/2003

COMELEC • CD908 • 13720 BELCODENE • Tél.: 04 42 70 63 90 Fax : 04 42 70 63 95

Prenez le petit circuit imprimé de l'afficheur et soudez (côté opposé au LCD) le connecteur double mâle/mâle à 16 broches ; vous pouvez alors placer cet afficheur sur la platine à l'aide du connecteur Mâle/Femelle à 16 pôles et des 4 entretoises métalliques.

Seul et unique réglage : après avoir monté et alimenté la platine, réglez le contraste du LCD avec un petit tournevis en agissant sur le curseur du trimmer de luminosité R1.

Les programmes en Assembleur pour PIC

Les trois nouveaux programmes que nous vous proposons pour cette platine sont le **LCD.ASM**, le **SIM.ASM** et le **TST.ASM**.

Au cours de l'installation vous les sauvegarderez dans le répertoire :

C:\IC-PROG\PRG DEMO.

Le programme LCD.ASM visualise sur le LCD :

*****ElectroniqueLM Cours de PIC Microchip*****

Le programme **SIM.ASM** visualise, lui, en séquence, les sept caractères **DDRAM** et **CGRAM** contenus dans le pilote de l'afficheur.

Avec ce programme, les sept caractères ASCII et quelques symboles de l'alphabet chinois s'affichent.

Le programme **TST.ASM** quant à lui est un simple exemple d'utilisation du clavier : à la pression de chaque touche correspond sur l'afficheur LCD la visualisation du chiffre ou du symbole choisi.

Le contenu du CDR1585

Un CDROM contenant les programmes **MPLAB IDE** version **7.20**, **IC-PROG** version **1.05D** et le tout nouveau **Proton DS Lite** version **1.036** est disponible.

En outre, dans le fichier d'installation de **IC-PROG**, vous trouverez une série de programmes d'exemples pour l'essai et l'utilisation de nos platines d'expérimentation dont celle faisant l'objet du présent article.

Quand vous installez **IC-PROG**, deux répertoires se créent automatiquement :

- le répertoire **PRG DEMO** de **IC-PROG** contient des programmes d'exemple en Assembleur pour PIC aux formats .asm et .hex.
- le répertoire **PRG DEMO BASIC** de **IC-PROG** contient les exemples Assembleur récrits en Proton Basic.

Note : pour le mode et les notes d'installation de ces programmes, lisez le fichier INDEX.HTML présent sur le CDR1585 et consultable avec un navigateur normal (Internet Explorer par exemple).

Comment construire ce montage ?

Tout le matériel nécessaire pour la réalisation de cette platine d'expérimentation à clavier et afficheur LCD pour microcontrôleurs PIC EN1585 ainsi que les composants programmés sont disponibles chez certains de nos annonceurs.

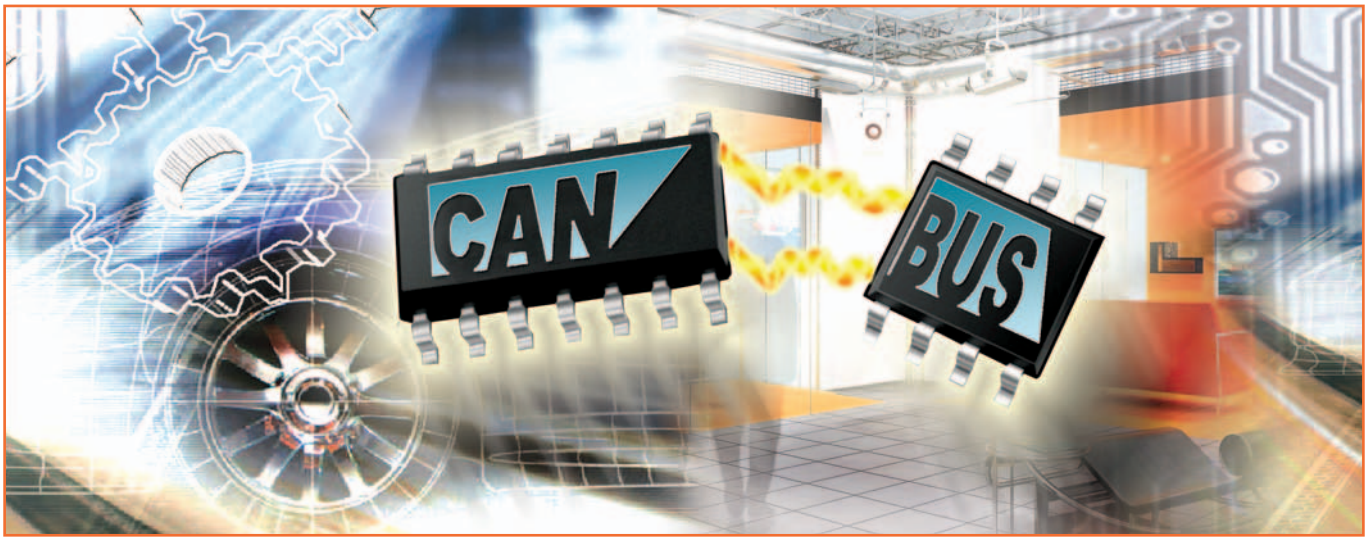
Voir les publicités dans la revue.

Les typons des circuits imprimés et les programmes disponibles sont téléchargeables à l'adresse suivante :
http://www.electronique-magazine.com/circuitrevue/086.zip. ◆

À la découverte du BUS CAN

Quatrième partie

Conçu comme protocole de communication série pour faire communiquer entre eux tous les systèmes électroniques présents à bord d'une voiture, le bus CAN gagne aussi du terrain dans les domaines d'automatisation industrielle (robotique) et de la domotique. Dans cette série d'articles, ou de Leçons (comme vous voudrez), nous allons aborder la théorie de son fonctionnement et nous prendrons de nombreux exemples dans le domaine domotique (c'est-à-dire des automatismes dédiés à la maison). Dans cette quatrième partie, nous analyserons comment un module peut acquérir des données et les rendre disponibles sur le bus.



Le mois dernier, en effet, nous nous sommes quittés après avoir examiné le matériel constituant le module CAN, en nous promettant de voir à la rentrée comment il peut acquérir des données extérieures et les rendre disponibles sur le bus pour les autres modules.

Première expérimentation : Sonde Thermo CAN

Commençons à mettre en pratique les concepts théoriques vus au cours des parties précédentes de cette série, avec une première expérimentation qui nous permettra de voir de près la communication standard entre deux nœuds CAN. Notre premier réseau, en effet, sera constitué d'une paire de nœuds : l'un aura pour tâche de relever la température ambiante au moyen d'une sonde DS18B20 et d'envoyer les données sur un bus CAN ; l'autre de mémoriser ces données dans une EEPROM 24LC256. Nous ne nous préoccuperons pas de formater les données reçues : nous les enregistrerons telles qu'elles nous parviennent de la sonde. Nous nous concentrerons, en effet, plutôt sur la communication CAN et analyserons deux fonctions fondamentales de la librairie ECAN : "ECANSendMessage" et "ECANReceiveMessage".

Elles constituent la base de tout programme résident gérant la communication sur bus CAN. Nous entendons, en outre, décrire deux aspects typiques de l'environnement C18 : l'intégration et la réutilisation du code.

En fait, le développement se fera en réutilisant des librairies du domaine public dont l'intégration et la personnalisation dépendront des objectifs que nous nous sommes fixés. Cette façon de faire permet d'augmenter notre productivité et de réaliser des programmes résidents facilement

Tableau 1.

Fonction	Fichier	Description
Protocole onewire	onewire.c onewire.h	Regroupe toutes les fonctions permettant d'effectuer le "reset" d'un dispositif onewire, de lui envoyer et d'en recevoir un octet. C'est une récriture des instructions OWIN, OWOUT du PICBasic.
CAN bus	ecan.c ecan.h ecan.def	C'est la librairie de base que nous utiliserons pour tous les projets logiciels sur bus CAN. Conçu pour la famille PIC18 à 64, 68, 80 broches mais, comme nous le verrons ultérieurement, elle peut s'adapter très facilement aux plus modestes 18F458 et, de toute façon, à tous les PIC18 dotés de module CAN.
EEPROM	xeeprom.c xeeprom.h	Il s'agit d'une librairie utilisée pour l'accès, la lecture et l'écriture des EEPROM à adressage 16 bits. Optimisée pour la 24LC256 montée sur nos nœuds CAN.
RS232	usart.h	C'est une librairie standard distribuée avec le compilateur C18. Permet de gérer tous les aspects de la communication série selon le standard RS232. Nous l'utiliserons pour les messages de contrôle envoyés au PC par le nœud émetteur.

personnalisables. Notre programme doit pouvoir dialoguer avec une sonde thermique au moyen d'un protocole "one-wire" (monofil), communiquer les données sur bus CAN et les écrire dans une EEPROM à travers l'interface I²C.

Enfin, nous implémenterons encore des messages dans le nœud émetteur, à des fins de diagnostic et de contrôle (ces messages travailleront sur RS232). Toutes ces fonctions sont regroupées dans des librairies. La structure générale du projet logiciel est résumée dans le **Tableau 1**. Nous utiliserons en outre deux fichiers indispensables à différentes occasions :

1) **C18cfg.asm** : contient les valeurs pour les bits de configuration du PIC, comme celles relatives à la désactivation du temporisateur du chien de garde ("watchdog timer") ou du type d'horloge utilisée ; nous y insérons aussi le fichier **.inc** à propos du microcontrôleur actuellement utilisé (dans notre cas **p18f458.inc**).
2) **compilatore.h** : dans ce fichier, nous regroupons toutes les définitions nécessaires à la reprise des registres du microcontrôleur et des valeurs intéressant la totalité du code de notre projet logiciel. Par exemple, nous redéfinissons la broche correspondant à la ligne données de la sonde thermique (RB5) comme PORTB_RB5 : le but de l'opération est d'en faciliter l'utilisation.

Afin de rendre plus clair le développement, analysons séparément les deux nœuds. Dans le paquet téléchargeable sur le site de la revue vous trouverez deux dossiers, le firmNODOTX et le firmNODORX, la finalité étant de distinguer les deux sous-projets. Commençons par le nœud émetteur.

Nœud TX

Sur le nœud émetteur nous relions une sonde thermique DS18B20. Nous utilisons la broche RB5 du PIC18F458 comme ligne de données. Les deux autres broches de la sonde doivent être reliées l'une à la masse (GND) et l'autre à la ligne +5 V (Vdd). Il est nécessaire de monter une résistance de tirage de 4,7 k sur la broche de sortie du capteur Dallas ; pour que cela soit plus simple, utilisons directement les barrettes latérales de la platine d'expérimentation (voir **figure 1**).

Une fois la connexion de la sonde effectuée, préparons notre programme résident. Nous avons surtout à réaliser un cycle de lecture des registres de la sonde concernant la température et leur envoi sur le bus CAN. Pour faciliter les actions de l'usager, faisons en sorte que l'échantillonnage se fasse dès que l'on presse le poussoir SW2 de la platine d'expérimentation, soit celui qui est relié à la broche RB0 du PIC. L'envoi des lectures s'arrête dès que ce poussoir est maintenu pressé. Nous avons décidé d'effectuer une mesure toutes les secondes environ. L'état du nœud est signalé par trois LED de couleurs, en particulier au démarrage la LED verte s'allume afin de signaler que le nœud est prêt à recevoir la commande de départ avec SW2.

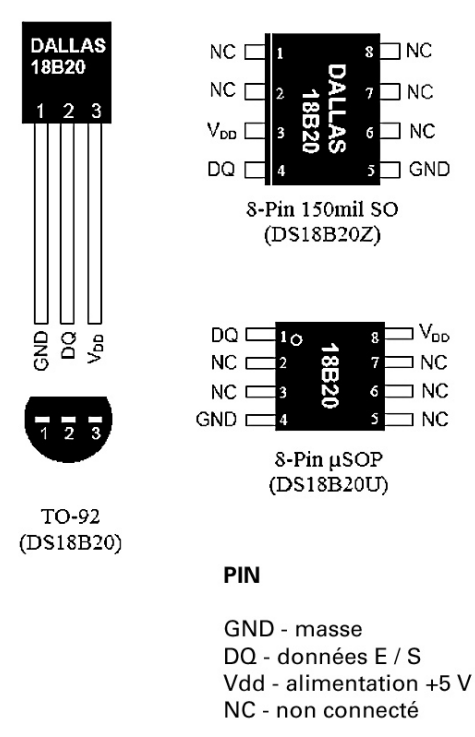


Figure 1 : Montage d'une sonde thermique DS18B20 sur le nœud émetteur et brochages de celle-ci (monter une résistance de tirage de 4,7 k sur la broche de sortie du capteur Dallas en utilisant les barrettes latérales de la platine d'expérimentation).

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
LS Byte	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
MS Byte	S	S	S	S	S	2^6	2^5	2^4

Figure 2 : À chaque demande la sonde répond par deux octets dont les bits sont disponibles dans deux registres, selon le format représenté ici.

Durant l'échantillonnage la LED rouge clignote. Enfin, après la pression suivante du poussoir, le nœud termine les opérations de transmission et la LED jaune s'allume. Toutes les phases d'élaboration peuvent être suivies à travers une session d'HyperTerminal ouverte directement sur le port série (celui-là même où nous avons relié la platine d'expérimentation) ; la COM devra être configurée selon les paramètres suivants : vitesse de transmission 19 200 bps, 8 bits de données, aucune parité, 1 bit de stop (19200-8-N-1).

Le premier problème rencontré a été celui consistant à écrire une petite librairie qui permet d'exécuter toutes les opérations de communication avec la sonde ; on a pu le résoudre facilement en nous référant au "datasheet" (banque de données) du composant, dans lequel on nous explique en détail comment le DS18B20 est interrogé, quelles réponses il donne et dans quel format il exprime la température.

Naturellement, les fonctions ainsi construites seront éminemment utiles chaque fois que nous devrons dialoguer avec un composant monofil ("onewire"). La sonde en question nous permet de mesurer la température en °C avec une précision d'un demi degré entre –10 °C et +85 °C. La valeur est exprimée par deux octets dont les bits représentent le module et le signe ; il est possible de définir combien de bits utiliser pour l'un et combien pour l'autre : nous avons opté pour la formule 11 bits de valeur absolue (module) et 5 de signe.

Donc, à chaque demande la sonde répond par deux octets dont les bits sont disponibles dans deux registres, selon le format illustré **figure 2**. Les bits S permettent d'établir le signe de la température relevée (S=0 pour les valeurs positives, S=1 pour les négatives). Par exemple, avec une valeur égale à 00A2h nous aurons une température égale à 10,125 °C et avec FFF8h une température de –0,5 °C.

Le protocole "OneWire"

Le système de communication prévu pour le DS18B20 réclame l'utilisation d'un protocole particulier mais très efficace. N'importe quelle opération commence par le dispositif "master" (maître), représenté ici par le PIC.

La première phase à considérer est le "reset" (réinitialisation) du dispositif, consistant en une impulsion provenant du "master", auquel la sonde répond par l'envoi d'un signal de présence. Pour vous éclaircir les idées à propos de la phase de réinitialisation, jetez un coup d'œil au schéma de la **figure 3**. Durant cette phase, le microcontrôleur met au niveau logique bas la ligne de données pendant au minimum 480 µs. Ensuite, il se met en réception et, pour cela, il se détache du bus ; par conséquent la résistance de tirage met la ligne au niveau logique haut. Dès que la sonde détecte ce changement, elle attend pendant un délai allant de 15 à 60 µs, puis elle envoie une impulsion de présence en mettant la ligne au niveau logique bas pendant un délai de 60 à 240 µs.

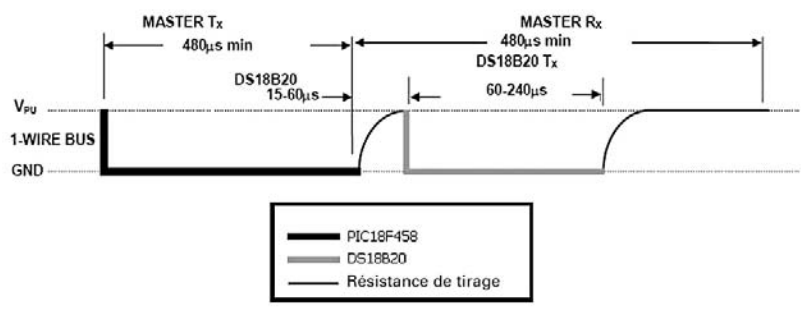


Figure 3 : Phase de réinitialisation du microcontrôleur.

A la fin elle se détache à nouveau du bus et la résistance de tirage remet la ligne de données au niveau logique haut. Toute cette procédure peut être facilement développée en C18. Nous avons regroupé les instructions sous la fonction du **Listing 1**.

La fonction Delay10TCYx(n) fait partie de la librairie standard C18 et permet d'insérer un retard égal à 10n cycles d'horloge. Il est clair que la temporisation dépend de la fréquence d'horloge de l'oscillateur utilisé (pour nous 20 MHz). Pour la mettre en œuvre, il suffit d'insérer les instructions suivantes :

```
#include <delays.h>
```

La "OWReset" renvoie une valeur égale à 1 si le PIC reçoit l'impulsion de présence de la part de la sonde et elle peut donc être utilisée à l'intérieur d'une expression logique. De même, nous avons préparé la fonction "OWTX" pour transmettre un octet au dispositif et la "OWRX" pour recevoir toujours un octet de celui-ci. Pour des raisons d'espace, nous n'analyserons pas leur "listing". Les plus curieux de nos lecteurs peuvent satisfaire leur "vilain défaut" en ouvrant le fichier du projet CANTX.mcp et en faisant un double clic sur le fichier "onewire.c". Toutes les déclarations relatives à chaque fonction on été recueillies dans le fichier onewire.h. Voici une liste qui fait la synthèse des diverses procédures :

- unsigned char **OWReset** (void) : exécute l'initialisation de la sonde en en détectant la présence sur le bus ; renvoie la valeur 1 si la sonde est présente ;

- void **OWTX** (unsigned char) : envoie au dispositif relié l'octet passé comme paramètre ;
- unsigned char **OWRX** (void) : reçoit un octet du dispositif relié et le passe à travers le paramètre de sortie ;
- unsigned char OWRX1 (void) : reçoit un unique bit du dispositif ; est utilisé pour vérifier si le relevé de la température est terminé ou non.

En ce qui concerne le relevé de la température, il est nécessaire d'utiliser des séquences spécifiques afin de commander le DS18B20. En envoyant la paire d'octets **CCh-44h** on lance une conversion.

Durant l'opération, le dispositif envoie un 0 et, dès qu'il a terminé, un 1 logique. Il est alors possible d'envoyer la séquence **CCh-BEh** pour lire les registres de la sonde, structurés comme le montre le **Tableau 2**.

Pour nous, il suffit d'extraire les deux premiers octets ; en effet, nous utilisons la configuration prédéfinie de la sonde et, pendant la lecture, nous écarterons les 7 octets restants.

Le code C18, qui permet de repérer les informations que nous devrons ensuite transférer via le bus CAN, est celui du Listing 2. On voit que les deux octets de la valeur de la température relevée sont transférés dans deux champs de 8 bits composant la variable **tempera**. Pour effectuer cette attribution de manière correcte, il a fallu définir une structure adéquate, comme le montre le code du Listing 3.

La communication RS232

Pour envoyer un message à un PC tout en permettant à l'usager de suivre les diverses phases de l'élaboration, nous devons utiliser une autre librairie distribuée avec le compilateur C18.

Là encore, nous mettons en œuvre une instruction d'inclusion ("include") relative au fichier usart.h contenant la déclaration des diverses fonctions utilisables ; voyons concrètement celles qui sont nécessaires à notre projet. Avant tout nous devons initialiser le port de communication selon les paramètres définis durant l'analyse initiale du projet logiciel. La fonction correspondante est la suivante :

void OpenUSART (unsigned char ***config***, unsigned int ***spbrg***);

Le premier paramètre est créé à travers une opération de AND entre une série de valeurs définies dans le fichier usart.h et qui permettent d'établir le fonctionnement précis du module série du PIC18F458, module correspondant aux broches RC6 et RC7 (**Tableau 3**). Le second précise la vitesse de transmission. Dans le cas d'une réception/transmission de type asynchrone, la valeur est calculée avec la formule suivante :

FOSC/(16* (nbr de bit/seconde + 1))

en prenant comme FOSC la valeur de la fréquence d'oscillation du quartz utilisé par le circuit. Pour nous, avec un FOSC de 20 000 000 (20 MHz) et une vitesse de 19 200 bps, la valeur du paramètre est :

((20 000 000 : 19 200) : 16) –1 = 64,104

Nous utilisons donc le 64. Ce résultat présuppose l'utilisation d'un prédiviseur ("prescaler") par 16. L'instruction utilisée pour l'initialisation du port est la suivante :

OpenUSART(USART_TX_INT_OFF&USART_RX_INT_OFF&USART_ASYNCH_MODE&USART_EIGHT_BIT&USART_CONT_RX&USART_BRGH_HIGH, 64);

Pour envoyer une quelconque phrase à travers le port série, nous pouvons nous servir de la **putrsUSART**.

Elle prend en entrée le pointeur à un flux de caractères ; en général on se sert de la **putrsUSART** pour des flux enregistrés dans la mémoire de données et de la **putrsUSART** pour des flux de la mémoire de programme. La déclaration que nous trouvons à l'intérieur de la usart.h est la suivante :

void putrsUSART (const rom char *data);

L'utilisation est très simple : il suffit de passer comme argument la phrase entre guillemets, comme le montre l'instruction suivante :

putrsUSART("Electronique Loisir Magazine \n\r");

Nous utilisons **\n** et **\r** pour les caractères, respectivement de "**new line**"

Listing 1

```
/*********************************************************************
 * Fonction:        unsigned char OWReset(void)
 * Input:           Aucun
 * Output:          1 = Dispositif présent
 *                  0 = Dispositif non présent
*********************************************************************/
unsigned char OWReset(void)
{
    unsigned char pd;

    pd = 0;
    TRISB_RB5 = 0;
    PORTB_RB5 = 0;
    Delay10TCYx(240);
    TRISB_RB5 = 1;
    Delay10TCYx(40);
    if (PORTB_RB5 ==0)
    pd = 1;
    Delay10TCYx(200);
    return pd;
}
```

Le PIC met au niveau logique bas la ligne DQ pendant 480 µs.

Le PIC se détache du bus, se met en réception et attend 80 µs de manière à détecter l'impulsion de présence pendant ce délai.

Si le niveau logique bas est détecté, la sonde a répondu correctement et donc le paramètre de sortie est valorisé à 1.

Tableau 2.

Octet 0	Température LSB
Octet 1	Température MSB
Octet 2	Registre THEORIE
Octet 3	Registre TL
Octet 4	Registre Configuration
Octet 5	Réservé FFh
Octet 6	Réservé 0Ch
Octet 7	Réservé 10h
Octet 8	CRC

(à la ligne) et “carriage return” (retour chariot) nécessaires pour faire revenir le curseur au début.

Les définitions pour le bus CAN

Après avoir vu ces quelques codes préliminaires, analysons le cœur du programme résident et par conséquent la partie nécessaire pour effectuer la communication sur le bus CAN. Avant tout, nous devons considérer le fichier **ECAN.def** contenant tous les paramètres nécessaires pour configurer le module CAN du PIC18F458. Par commodité, nous utilisons l’interface graphique de Microchip Application Maestro ; toutefois, pour effectuer les modifications adéquates, vous êtes libres d’utiliser n’importe quel éditeur de texte. Le **Tableau 4** représente la séquence entière des paramètres utilisés dans nos nœuds ; pour la description des champs, nous vous renvoyons à la troisième partie, soit au numéro précédent d’ELM.

On utilise des points de suspension là où les valeurs se répètent pour plusieurs paramètres à la suite ; par exemple, la valeur 0x0L pour le paramètre ECAN_RXF2_VAL se répète aussi pour ECAN_RXF3_VAL, ECAN_RXF4_VAL, ECAN_RXF5_VAL.

Attention, nous utiliserons un mode de fonctionnement constant pour toute

Listing 2.

```
OWReset();
OWTX(0xCC);
OWTX(0x44);
while (OWRX1());
OWReset();
OWTX(0xCC);
OWTX(0xBE);
tempera.bytes.LSB = OWRX();
tempera.bytes.MSB = OWRX();
for (indice=1;indice<=7;indice++)
conv=OWRX();
```

Reset de la sonde et lancement de la détection de la température.

Attente jusqu’à la fin de la détection.

Demande des 9 octets contenus dans les registres de la sonde.

Enregistrement des deux premiers octets.

Les 7 derniers octets écartés.

Listing 3.

```
union
    {
        unsigned short Val;
        struct
        {
            unsigned char LSB; /*Last significant byte*/
            unsigned char MSB; /*Most significant byte*/
        } bytes;
    } tempera;      /*Température divisée en deux champs*/
```

Tableau 3.

Valeur	Description
USART_TX_INT_ON USART_TX_INT_OFF	Active/désactive le signal d’interruption en transmission
USART_RX_INT_ON USART_RX_INT_OFF	Active/désactive le signal d’interruption en réception
USART_ASYNCH_MODE USART_SYNCH_MODE	Active le mode de transmission/réception asynchrone ou synchrone
USART_EIGHT_BIT USART_NINE_BIT	Transmet/reçoit données à 8 bits ou à 9 bits
USART_SYNC_SLAVE USART_SYNC_MASTER	En mode de transmission/réception synchrone configure le module comme Esclave ou comme Maître
USART_SINGLE_RX USART_CONT_RX	Etablit si la réception doit se faire de manière continue ou pour un seul paquet
USART_BRGH_HIGH USART_BRGH_LOW	Etablit si le module doit être initialisé pour une vitesse de transmission élevée ou non

Tableau 4.

Paramètre	Valeur
ECAN_LIB_MODE_VAL	ECAN_LIB_MODE_FIXED
ECAN_FUNC_MODE_VAL	ECAN_MODE_0
ECAN_INIT_MODE	ECAN_INIT_NORMAL
ECAN_SJW_VAL	2
ECAN_BRP_VAL	4
ECAN_PHSEG1_VAL	8
ECAN_PHSEG2_VAL	8
ECAN_PROPDEG_VAL	8
ECAN_PHSEG2_MODE_VAL	ECAN_PHSEG2_MODE_PROGRAMMABLE
ECAN_BUS_SAMPLE_MODE_VAL	ECAN_BUS_SAMPLE_MODE_THRICE
ECAN_WAKEUP_MODE_VAL	ECAN_WAKEUP_MODE_ENABLE
ECAN_FILTER_MODE_VAL	ECAN_FILTER_MODE_DISABLE
ECAN_TXDRIVE_MODE_VAL	ECAN_TXDRIVE_MODE_VDD
ECAN_TX2_SOURCE_VAL	ECAN_TX2_SOURCE_COMP
ECAN_CAPTURE_MODE_VAL	ECAN_CAPTURE_MODE_DISABLE
ECAN_RXB0_MODE_VAL	ECAN_RECEIVE_ALL_VALID
ECAN_RXB0_DBL_BUFFER_MODE_VAL	ECAN_DBL_BUFFER_MODE_DISABLE
ECAN_RXB1_MODE_VAL	ECAN_RECEIVE_ALL_VALID
ECAN_B1_TXRX_MODE_VAL	ECAN_BUFFER_TX
ECAN_B1_MODE_VAL	ECAN_RECEIVE_ALL_VALID
ECAN_B1_AUTORTR_MODE	ECAN_AUTORTR_MODE_DISABLE
...	
ECAN_B5_TXRX_MODE_VAL	ECAN_BUFFER_TX
ECAN_B5_MODE_VAL	ECAN_RECEIVE_ALL_VALID
ECAN_B5_AUTORTR	ECAN_AUTORTR_MODE_DISABLE
ECAN_RXF0_MODE_VAL	ECAN_RXFn_ENABLE
ECAN_RXF0_MSG_TYPE_VAL	ECAN_MSG_STD
ECAN_RXF0_VAL	0x0L
ECAN_RXF0_BUFFER_VAL	RXB0
ECAN_RXF0_MASK_VAL	ECAN_RXM0
ECAN_RXF1_MODE_VAL	ECAN_RXFn_ENABLE
ECAN_RXF1_MSG_TYPE_VAL	ECAN_MSG_STD
ECAN_RXF1_VAL	0x0L
ECAN_RXF2_MODE_VAL	ECAN_RXFn_ENABLE
ECAN_RXF2_MSG_TYPE_VAL	ECAN_MSG_STD
ECAN_RXF2_VAL	0x0L
ECAN_RXF2_BUFFER_VAL	RXB1
ECAN_RXF2_MASK_VAL	ECAN_RXM1
...	
ECAN_RXF5_MODE_VAL	ECAN_RXFn_ENABLE
ECAN_RXF5_MSG_TYPE_VAL	ECAN_MSG_STD
ECAN_RXF5_VAL	0x0L
ECAN_RXF5_BUFFER_VAL	RXB1
ECAN_RXF5_MASK_VAL	ECAN_RXM1
ECAN_RXF6_MODE_VAL	ECAN_RXFn_ENABLE
ECAN_RXF6_MSG_TYPE_VAL	ECAN_MSG_STD
ECAN_RXF6_VAL	0x0L
ECAN_RXF6_BUFFER_VAL	RXB0
ECAN_RXF6_MASK_VAL	ECAN_RXM0
...	
ECAN_RXF15_MODE_VAL	ECAN_RXFn_ENABLE
ECAN_RXF15_MSG_TYPE_VAL	ECAN_MSG_STD
ECAN_RXF15_VAL	0x0L
ECAN_RXF15_BUFFER_VAL	RXB0
ECAN_RXF15_MASK_VAL	ECAN_RXM0
ECAN_RXM0_MSG_TYPE_VAL	ECAN_MSG_STD
ECAN_RXM0_VAL	0x0L
ECAN_RXM1_MSG_TYPE_VAL	ECAN_MSG_STD
ECAN_RXM1_VAL	0x0L

l'élaboration (ECAN_LIB_MODE_FIXED) avec le protocole standard (ECAN_MODE_0) et la gestion des opérations de transmission comme de réception (ECAN_

```
//          ECANCON_MDSEL1 = ECAN_FUNC_MODE_VAL >> 7;
//          ECANCON_MDSEL0 = ECAN_FUNC_MODE_VAL >> 6;
```

ECANCON: ENHANCED CAN CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
MDSEL1(1, 2)	MDSEL0(1, 2)	FIFOWM	EWIN4	EWIN3	EWIN2	EWIN1	EWIN0
bit 7							bit 0

Figure 4 : La structure du registre ECANCON.

RXMnSIDL: RECEIVE ACCEPTANCE MASK n STANDARD IDENTIFIER MASK REGISTERS, LOW BYTE [0 ≤ n ≤ 1]

R/W-x	R/W-x	R/W-x	U-0	R/W-0	U-0	R/W-x	R/W-x
SID2	SID1	SID0	—	EXIDEN(1)	—	EID17	EID16
bit 7							bit 0

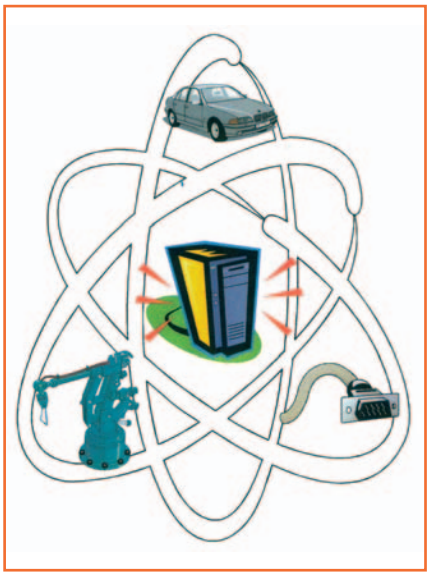
RXMnSIDL: RECEIVE ACCEPTANCE MASK n STANDARD IDENTIFIER MASK, LOW BYTE REGISTERS

R/W-x	R/W-x	R/W-x	U-0	U-0	U-0	R/W-x	R/W-x
SID2	SID1	SID0	—	—	—	EID17	EID16
bit 7							bit 0

Figure 5 : Comparons le registre RXM0SIDL dans les deux types de PIC.

INIT_NORMAL). Si l'on considère que le "bit time" (durée du bit) est égal à 25 TQ (le maximum selon le standard CAN) et que nous utilisons une fréquence de 20 MHz, la vitesse de transmission est égale à 80 kbps.

La vitesse en question est certainement suffisante, du moins si l'on considère que nos paquets auront une partie de données longue de deux octets tout au plus, ceux relatifs à la température relevée. Si vous observez les autres paramètres, vous remarquerez que toutes les fonctions de filtrage des messages (filtres et masques sont à zéro) sont désactivés ; tous ceux considérés comme valides seront acceptés. Bien que ce soit une simplification, cela nous permettra d'apprécier la différence de fonctionnement dans le cas où l'un des filtres du module serait activé. Pour notre premier projet, nous laissons une possibilité de dialogue maximale entre les deux nœuds.



Modifications d'implémentation

La librairie que nous utilisons est une version modifiée de celle que l'on trouve sur le site de Microchip. Les différences sont en partie formelles et en partie substantielles. Dans le premier cas, on a principalement éliminé les définitions et les structures relatives à l'utilisation des autres compilateurs ; dans l'autre, nous avons éliminé certaines instructions spécifiques de la classe supérieure des PIC (64, 68, 80 broches) pour lesquels la librairie a été initialement développée.

Deux modifications seulement, mais il faudra bien les prendre en considération si vous voulez réaliser un projet avec ce groupe de microcontrôleurs. Toutes les deux concernent la fonction ECANInitialize utilisée pour la configuration et le lancement du module.

Ces deux instructions font référence à deux bits du registre ECANCON, lequel est implémenté dans les PIC 18F6585, 18F8585, 18F6680, 18F8680, mais pas dans le PIC18F458. La structure du registre est celle de la **figure 4**.

Les deux bits en question permettent d'établir le mode de fonctionnement du module CAN. Pour le PIC18F458 on utilise exclusivement le MODO0, qui correspond à MDSEL0=0 et MDSEL1=0.

Précisons que la présence de ces deux instructions comporte une erreur de compilation, car le symbole ECANCON est inconnu. C'est pourquoi dans notre version les deux instructions ont été éliminées.

De même, on a effacé toute référence au registre substrat. La seconde modification concerne la valorisation des deux bits de configuration relatifs à l'identifiant de message que le nœud peut filtrer (Standard ou "Extended", étendu). Les deux bits ne sont utilisables qu'en modes MODO1 et 2. Là encore, nous nous trouvons confrontés à une structure implémentée dans la classe supérieure et non dans l'inférieure.

Comparons le registre **RXM0SIDL** dans les deux types de PIC (**figure 5**). Comme on peut le voir, le bit **EXIDEN** n'est pas utilisé avec le PIC18F458 et donc nous pouvons éliminer le code correspondant et ne maintenir que la partie relative à la valorisation du masque. D'où le code suivant :

Listing 4.

```
    ECANSetOperationMode(ECAN_OP_MODE_CONFIG);

    BRGCON1 = ((ECAN_SJW_VAL-1) << 6) | (ECAN_BRP_VAL-1);
    BRGCON2 = (ECAN_PHSEG2_MODE_VAL << 7)       | \
              (ECAN_BUS_SAMPLE_MODE_VAL << 6)   | \
              ((ECAN_PHSEG1_VAL-1) << 3)        | \
              (ECAN_PROPSEG_VAL-1);
    BRGCON3 = (ECAN_WAKEUP_MODE_VAL << 7)       |
              (ECAN_FILTER_MODE_VAL << 6)       |
              (ECAN_PHSEG2_VAL-1);
    CIOCON = ECAN_TX2_SOURCE_VAL << 7           | \
             ECAN_TX2_MODE_VAL << 6         | \
             ECAN_TXDRIVE_MODE_VAL << 5     | \
             ECAN_CAPTURE_MODE_VAL;

 #if ( ECAN_FUNC_MODE_VAL == ECAN_MODE_0 )
    RXB0CON = (ECAN_RXB0_MODE_VAL << 5) | (ECAN_RXB0_DBL_BUFFER_MODE_VAL << 2);
    RXB1CON = ECAN_RXB1_MODE_VAL << 5;
#endif

#if ( (ECAN_RXF0_MODE_VAL == ECAN_RXFn_ENABLE) || (ECAN_FUNC_MODE_VAL == ECAN_MODE_0) )
    // Set Standard or Extended value.
    #if ( ECAN_RXF0_MSG_TYPE_VAL == ECAN_MSG_STD )
        _SetStdRXFnValue(RXF0, ECAN_RXF0_VAL);
    #else
        _SetXtdRXFnValue(RXF0, ECAN_RXF0_VAL);
    #endif
#endif

#if ( (ECAN_RXF1_MODE_VAL == ECAN_RXFn_ENABLE) || (ECAN_FUNC_MODE_VAL == ECAN_MODE_0) )
    #if ( ECAN_RXF1_MSG_TYPE_VAL == ECAN_MSG_STD )
        _SetStdRXFnValue(RXF1, ECAN_RXF1_VAL);
    #else
        _SetXtdRXFnValue(RXF1, ECAN_RXF1_VAL);
    #endif
#endif

..............

#if ( (ECAN_RXF5_MODE_VAL == ECAN_RXFn_ENABLE) || (ECAN_FUNC_MODE_VAL == ECAN_MODE_0) )
    #if ( ECAN_RXF5_MSG_TYPE_VAL == ECAN_MSG_STD )
        _SetStdRXFnValue(RXF5, ECAN_RXF4_VAL);
    #else
        _SetXtdRXFnValue(RXF5, ECAN_RXF5_VAL);
    #endif
#endif

#if ( ECAN_RXM0_MSG_TYPE == ECAN_MSG_STD )

    _SetStdRXMnValue(0, ECAN_RXM0_VAL);
//    RXM0SIDL_EXIDEN = 0;
#else
{
    _SetXtdRXMnValue(0, ECAN_RXM0_VAL);
//    RXM0SIDL_EXIDEN = 1;
}
#endif

#if ( ECAN_RXM1_MSG_TYPE == ECAN_MSG_STD )
    _SetStdRXMnValue(1, ECAN_RXM1_VAL);
//    RXM1SIDL_EXIDEN = 0;
#else
    _SetXtdRXMnValue(1, ECAN_RXM1_VAL);
//    RXM1SIDL_EXIDEN = 1;
#endif

#if ( ECAN_INIT_MODE != ECAN_INIT_CONFIGURATION )
    ECANSetOperationMode(ECAN_INIT_MODE);
#endif
}
```

Met le module en “configuration mode”

Précise le taux de transfert (“bit rate”) en fonction des paramètres contenus en ECAN.def ainsi que le fonctionnement de la ligne TX2. Pour nous les bits du registre CIOCON sont désactivés.

Mode de fonctionnement des “buffers”.

Mode de fonctionnement des filtres.

Précise le mode de fonctionnement des masques et établit par exemple s’il faut activer ou non la possibilité de filtrer les messages avec identifiant standard ou étendu.

A la fin, commute le module dans le mode établi avec le paramètre ECAN_INIT_NORMAL qui active le nœud en émission comme en réception.

Listing 5.

```
void ECANSetOperationMode(ECAN_OP_MODE mode)
{
    CANCON &= 0x1F;       // efface le mode précédent
    CANCON |= mode;       // valorise les bits du nouveau mode

    while( ECANGetOperationMode() != mode ); //  Attend que le module change d'état
}
```

CANCON: CAN CONTROL REGISTER

R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
REQOP2	REQOP1	REQOP0	ABAT	WIN2	WIN1	WIN0	—
bit 7							bit 0

Figure 6 : Structure du registre CANCON.

```
#if ( ECAN_RXM0_MSG_TYPE == ECAN_MSG_STD )
    _SetStdRXMnValue(0, ECAN_RXM0_VAL);
//     RXM0SIDL_EXIDEN = 0;
#else
    _SetXtdRXMnValue(0, ECAN_RXM0_VAL);
 //    RXM0SIDL_EXIDEN = 1;
#endif
```

La librairie **ECAN** est assez universelle et elle permet le développement de codes pour les deux classes de PIC. Naturellement, il est nécessaire de prêter un peu d'attention aux diverses fonctions que nous voulons activer.

En fait, avec la famille inférieure, l'utilisation de la broche TX2 du module n'est pas possible, car elle n'existe tout bonnement pas.

La routine d'initialisation

La première fonction de la librairie **ECAN** que nous allons analyser est celle qui permet la configuration du module CAN et son lancement.

Il s'agit en substance d'une procédure qui met le module CAN en mode Configuration, puis exécute toute une série d'initialisations sur la base de paramètres contenus dans le fichier **ECAN.def** (par suite sont précisés la vitesse de transmission, les "buffers" de transmission et réception, les éventuels filtres, etc.).

Toutes ces choses étant faites, le module est mis en mode de fonctionnement établi dans le paramètre **ECAN_INIT_MODE** (voir les dernières lignes du **Listing 4**). La fonction en question n'a pas de paramètres en entrée ni en sortie ; ci-dessous nous donnons sa déclaration (elle est contenue dans le fichier ECAN.def) :

void ECANInitialize(void);

Si nous allons éplucher ce "listing", nous identifions facilement les diverses phases de cette procédure ; référez-vous bien au Listing 4, dans lequel vous aurez sans doute noté l'utilisation d'une sous-procédure appelée **ECANSetOperationMode**. Le code de cette dernière est fort simple et il est visible dans le **Listing 5**.

La chose devient encore plus claire (!) si nous jetons un coup d'œil à la structure du registre CANCON (**figure 6**).

Avec un peu d'attention, nous voyons que la première opération de AND ne fait pas autre chose que mettre à zéro les trois bits les plus significatifs et maintenir les cinq autres tels quels. Selon la valeur donnée à ces bits, le mode de fonctionnement change :

1xx: Configuration Mode
011: Listen Only
010: Loopback
001: Désactivé
000: Normal

Ensuite, à travers le OR, le registre est valorisé par le paramètre Mode communiqué à la fonction. Dans le dernier "while" on ne fait que vérifier que le module atteint bien l'état correspondant.

L'instruction correspondante est un AND logique et elle est insérée à travers une déclaration que nous trouvons dans le fichier **ECAN.h**, comme le montre la ligne suivante :

#define ECANGetOperationMode() (CANCON & ECAN_OP_MODE_BITS)

L'initialisation du module CAN se fait simplement en appelant la **ECANInitialize**, comme nous le verrons dans la cinquième partie le mois prochain.

Nous reviendrons sur la conclusion de ce galop d'essai consacré à notre première expérimentation. Nous y analyserons en outre l'envoi des messages standard sur le bus CAN et la structure du nœud récepteur.

Enfin, nous relierons adéquatement les deux circuits, nous les alimenterons et nous suivrons les phases de communication entre les deux nœuds CAN directement sur l'écran de notre PC.

Rendez-vous, donc, en octobre pour la cinquième partie de cette série (ou de ce Cours) consacrée au bus CAN.

Comment construire ce montage ?

Tout le matériel nécessaire pour construire la platine d'expérimentation bus CAN est d'ores et déjà disponible chez certains de nos annonceurs. Voir les publicités dans la revue.

Les typons des circuits imprimés et les programmes disponibles sont téléchargeables à l'adresse suivante :
http://www.electronique-magazine.com/circuitrevue/086.zip. ◆

PETITES ANNONCES

Ingénieur informatique donne cours de programmation VB C C++ DLL. Active COM Tous niveaux tarif 20 € de l'heure Paris 35€ Banlieue IDF. Tél. : 06.20.70.89.71

Vends oscilloscope professionnel de marque Schlumberger 2 x 50 MHz au prix de 100€ Tél. : 04.66.67.14.09

Achète ordinateur Amiga 4000 processeur 68040 à 25 MHz et 32 bits contrôleur SCSI2 ; + disque dur. Faire offre, description Tél. : 06.33.15.90.07

Cherche programmateur de PROM ou schéma SN74S188 Tél. : 04.90.86.95.31

Vends lot de 50 appareils de mesure (oscillos, génés, alim) dont 38 en état de fonctionnement et 12 à réviser ou à réparer 750 € + port. Photocopieurs récents (1998 / 2001) révisés bon état 95 € en A4 110 € en A4/B4 180 € tous formats réductions Zooms etc.... Tél. 02.48.64.68.48.

Recherche professeur d'électronique pour des cours particuliers dans le Gard. Tél. : 04.66.67.14.09

Vends oscillo telequipement D61a 2 x 10 MHz, 120 € + port, multimètre d'atelier AOIP (unie-2 b), 50 € + port, postes de radio et électrophones à lampes. Cherche magnétophone Hencot pour pièces et livre (pannes radio) de sorokine, éditions radio, des années 60 : faire offre. Tél. : 06.14.98.31.05

Vends lampes radio TV anciennes et récentes. Liste sur demande. Tél. :03 25 87 11 90

Recherche professeur d'électronique pour des cours particuliers dans le Gard. Tél. : 04.66.67.14.09

Vends Rucal Dana 9303, voltmètre, wattmètre pro à microprocesseur 2 GHZ en bon état avec notice complète 300 € Tél. : 01.69.30.64.50 le soi ou laisser un message.

Vends transfo torique ARABEL 630 VA 2 x 33 V. Vends OSCILLO à lampe METRIX monotrace. Lot de condos en C038 et C039. Vends transistors 2N3055 et son complémentaire. Tél. : 06.63.97.22.48.

INDEX DES ANNONCEURS

ELC – Alimentation	2
COMELEC – Kits du mois	4
PCB POOL – Réalisation de prototypes	10
MULTIPOWER – Autoformation et CAO	10
COMELEC – Mesure	28
SELECTRONIC – Catalogue et site	50
ARQUIÉ– Catalogue N°63	54
MICRELEC –Chaîne CAO	54
GO TRONIC – Catalogue et site	66
COMELEC – PNP	66
JMJ –Bon d'abonnement	77
REF UNION – Salon HAMEXPO	78
JMJ – CD-Rom anciens numéros ELM	79
COMELEC – Kits Santé	80

ANNONCEZ-VOUS !

VOTRE ANNONCE POUR SEULEMENT 2 TIMBRES* À 0,53 € !

LIGNES	TEXTE : 30 CARACTÈRES PAR LIGNE. VEUILLEZ RÉDIGER VOTRE PA EN MAJUSCULES. LAISSEZ UN BLANC ENTRE LES MOTS.
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

*Particuliers : 2 timbres à 0,53 € - Professionnels : La grille : 90,00 € TTC - PA avec photo : + 30,00 € - PA encadrée : + 8,00 €

Nom ..... **Prénom** .....

Adresse .....

Code postal ..... **Ville** .....

Toute annonce professionnelle doit être accompagnée de son règlement libellé à l'ordre de JMJ éditions. Envoyez la grille, avant le 10 précédent le mois de parution, accompagnée de votre règlement à l'adresse:

JMJ/ELECTRONIQUE • Service PA • BP 20025 • 13720 LA BOUILLADISSE

Directeur de Publication
Rédacteur en chef
J-M MOSCATI
redaction@electronique-magazine.com

Direction - Administration
JMJ éditions
B.P. 20025
13720 LA BOUILLADISSE
Tél. : 0820 820 534
Fax : 0820 820 722

Secrétariat - Abonnements
Petites-annonces - Ventes
A la revue

Vente au numéro
A la revue

Publicité
A la revue

Maquette - Illustration
Composition - Photogravure
JMJ éditions sarl

Impression
SAJIC VIEIRA - Angoulême
Imprimé en France / Printed in France

Distribution
NMPP

Hot Line Technique
0820 000 787*
du lundi au vendredi de 16 h à 18 h

Web
www.electronique-magazine.com

e-mail
info@electronique-magazine.com

* N° INDIGO : 0,12 € / MN

ELECTRONIQUE ET LOISIRS magazine
LE MENSUEL DE L'ÉLECTRONIQUE POUR TOUS

EST RÉALISÉ
EN COLLABORATION AVEC :

NUOVA ELETTRONICA
Elettronica In

JMJ éditions
Sarl au capital social de 7800 €
RCS MARSEILLE : 421 860 925
APE 221E
Commission paritaire: 1000T79056
ISSN: 1295-9693
Dépôt légal à parution

I M P O R T A N T

Reproduction, totale ou partielle, par tous moyens et sur tous supports, y compris l'internet, interdite sans accord écrit de l'Editeur. Toute utilisation des articles de ce magazine à des fins de notice ou à des fins commerciales est soumise à autorisation écrite de l'Editeur. Toute utilisation non autorisée fera l'objet de poursuites. Les opinions exprimées ainsi que les articles n'engagent que la responsabilité de leurs auteurs et ne reflètent pas obligatoirement l'opinion de la rédaction. L'Editeur décline toute responsabilité quant à la teneur des annonces de publicités insérées dans le magazine et des transactions qui en découlent. L'Editeur se réserve le droit de refuser les annonces et publicités sans avoir à justifier ce refus. Les noms, prénoms et adresses de nos abonnés ne sont communiqués qu'aux services internes de la société, ainsi qu'aux organismes liés contractuellement pour le routage. Les informations peuvent faire l'objet d'un droit d'accès et de rectification dans le cadre légal.



ABONNEZ VOUS à ELECTRONIQUE

ET LOISIRS magazine

LE MENSUEL DE L'ÉLECTRONIQUE POUR TOUS











et profitez de vos privilèges !

RECEVOIR votre revue directement dans votre boîte aux lettres près d'une semaine avant sa sortie en kiosques

BÉNÉFICIER de 50% de remise** sur les CD-Rom des anciens numéros

voir page 79 de ce numéro.

ASSURANCE de ne manquer aucun numéro

RECEVOIR un cadeau* !

* Pour un abonnement de 24 numéros uniquement (délai de livraison : 4 semaines environ). ** Réservé aux abonnés 12 et 24 numéros.

OUI, Je m'abonne à ELECTRONIQUE ET LOISIRS magazine — LE MENSUEL DE L'ÉLECTRONIQUE POUR TOUS

A PARTIR DU N° 87 ou supérieur

E086

Ci-joint mon règlement de _____ € correspondant à l'abonnement de mon choix.

Adresser mon abonnement à : Nom _____ Prénom _____

Adresse _____

Code postal _____ Ville _____

Tél. _____ e-mail _____

❑ chèque bancaire ❑ chèque postal ❑ mandat

❑ Je désire payer avec une carte bancaire
Mastercard – Eurocard – Visa

Date d'expiration :

Cryptogramme visuel :
(3 derniers chiffres du n° au dos de la carte)

*Date, le*_____

Signature obligatoire ▷

Avec votre carte bancaire, vous pouvez vous abonner par téléphone.

TARIFS CEE/EUROPE

❑ **12 numéros** — **49,00 €**

TARIFS FRANCE

❑ **6 numéros** — **22,00 €**
au lieu de 27,00 € en kiosque, soit **5,00 € d'économie**

❑ **12 numéros** — **41,00 €**
au lieu de 54,00 € en kiosque, soit **13,00 € d'économie**

❑ **24 numéros** — **79,00 €**
au lieu de 108,00 € en kiosque, soit **29,00 € d'économie**

Pour un abonnement 24 numéros, cochez la case du cadeau désiré.

DOM-TOM/HORS CEE OU EUROPE : NOUS CONSULTER

1 CADEAU au choix parmi les 5

POUR UN ABONNEMENT DE 24 numéros



Gratuit :

❑ **Un money-tester**
❑ **Une radio FM / lampe**
❑ **Un multimètre**
❑ **Un réveil à quartz**
❑ **Une revue supplémentaire**



Avec 4,00 € uniquement en timbres :

❑ **Un alcootest électronique**



Photos non contractuelles

délai de livraison : 4 semaines dans la limite des stocks disponibles

POUR TOUT CHANGEMENT D'ADRESSE, N'OUBLIEZ PAS DE NOUS INDIQUER VOTRE NUMÉRO D'ABONNÉ (INSCRIT SUR L'EMBALLAGE)

Bulletin à retourner à : JMJ – Abo. ELM

B.P. 20025 - 13720 LA BOUILLADISSE – Tél. 0820 820 534 – Fax 0820 820 722

HAMEXPO

28 ème Salon International Radioamateur

Techniques de radiocommunication et informatique

7-8 Octobre 2006



Réseau des Emetteurs Français - Union Française des Radioamateurs
REF-UNION 32, rue de Suède BP 77429 - 37074 TOURS cedex 2
Tél: 02 47 41 88 73 - www.ref-union.org

CD-ROM ENTIÈREMENT IMPRIMABLE

LISEZ ET IMPRIMEZ VOTRE REVUE SUR VOTRE ORDINATEUR PC OU MACINTOSH

50 € Les 3 CD du Cours d'Électronique en Partant de Zéro







COURS NIVEAU 3

SOMMAIRE INTERACTIF

ENTIÈREMENT IMPRIMABLE







5.50 € LE CD

SUPER AVANTAGE POUR LES ABONNÉS DE 1 OU 2 ANS

- 50 % SUR TOUS LES CD DES ANCIENS NUMÉROS CI - DESSOUS



LE CD 6 NUMÉROS

24€



LE CD 12 NUMÉROS

43€

FRAIS DE PORT INCLUS POUR LA FRANCE (DOM-TOM ET AUTRES PAYS: NOUS CONSULTER.)

adressez votre commande à :

JMJ/ELECTRONIQUE - B.P. 20025 - 13720 LA BOUILLADISSE *avec un règlement par Chèque à l'ordre de* ***JMJ***

Par téléphone : 0820 820 534 ou par fax : 0820 820 722 *avec un règlement par Carte Bancaire*

Vous pouvez également commander par l'Internet : www.electronique-magazine.com/anc_num.asp

FORME

UN GÉNÉRATEUR D'ULTRASONS À USAGE MÉDICAL



La capacité de pénétration des ultrasons dans les tissus du corps humain a révolutionné l'imagerie médicale (avec l'échographie) et donc la fiabilité du diagnostique. Cette propriété des ultrasons les fait également utiliser en physiothérapie avec un succès qui n'est plus à démontrer. L'appareil que nous vous proposons de construire est un générateur d'ultrasons à usage médical : il vous rendra de grands services pour de nombreuses affections (comme Arthropathie, Arthrose, Arthrite, Névrite, Périarthrite, Tendinite, Epicondylite, Traumatisme par contusion, Retard de consolidation osseuse, Adiposité localisée, Ostéite, Myalgie, Bursite, Lombalgie, Rigidité et douleur articulaire) qu'il vous aidera à soigner. Le diffuseur professionnel SE1.6 est livré monté est étalonné avec son cordon.

EN1627K.. Kit complet avec coffret et 1 diffuseur SE1.6 ..... 290,00 €
SE1.6..... diffuseur ultrassons ..... 139,00 €

UN ÉLECTROSTIMULATEUR BIPHASIQUE ABDOMINAL



Cet électrostimulateur neuromusculaire a été conçu spécialement pour faire travailler les abdominaux en entraînement passif (allongé sur son lit !) ou en mixte (en faisant du footing... ou la cuisine !) puisqu'il est portatif. Il comporte quatre programmes correspondant à quatre traitements : idéal pour se maintenir en forme ou pour entretenir son esthétique quand on n'a pas trop de temps.

ET447 ..... Kit complet avec batterie et électrodes .....120,00 €

UN APPAREIL DE MAGNÉTOTHÉRAPIE À MICROCONTRÔLEUR ST7



Beaucoup de médecins et de praticiens de santé, comme les kinésithérapeutes, utilisent la magnétothérapie : certains ont découvert qu'en faisant varier de manière continue la fréquence des impulsions on accélère la guérison et on élimine plus rapidement la douleur. Les maladies que l'on peut traiter avec cet appareil de magnétothérapie sont très nombreuses. Vous trouverez ci-dessous la liste des plus communes, suggérées par le corps médical et le personnel paramédical, : arthrose, arthrite, sciatique, lombalgie, tendinite, talalgie, déchirure et douleur musculaires, luxation, fractures ect.

EN1610 .... Kit complet avec boitier mais sans nappe ..... 79,00 €
PC1293 .... Nappe dimensions 22 x 42 cm ..... 31,00 €
PC1325 .... Nappe dimensions 13 x 85 cm ..... 31,50 €

STIMULATEUR ANALGESIQUE



Cet appareil permet de soulager des douleurs tels l'arthrose et les céphalées. De faible encombrement, ce kit est alimenté par piles incorporées de 9 volts. Tension électrode maximum : –30 V - +100 V. Courant électrode maximum : 10 mA. Fréquences : 2 à 130 Hz.

EN1003 .... Kit complet avec boitier ..... 36,30 €

MAGNETOTHERAPIE VERSION VOITURE



La magnétothéraphie est très souvent utilisée pour soigner les maladies de notre organisme (rhumatismes, douleurs musculaires, arthroses lombaires et dorsales) et ne nécessite aucun médicament, c'est pour cela que tout le monde peut la pratiquer sans contre indication. (Interdit uniquement pour les porteurs de Pace-Maker.

EN1324 .... Kit complet avec boîtier et une nappe version voiture ..... 66,50 €
PC1324 .... Nappe supplémentaire ..... 27,50 €

UN GÉNÉRATEUR D'ONDES DE KOTZ POUR SPORTIFS ET KINÉS



Le générateur d'ondes de Kotz est utilisé en médecine pour la récupération musculaire des personnes ayant eu un accident ou une maladie et qui sont donc restées longtemps inactives, comme pour le sport ou l'esthétique corporelle afin de tonifier et raffermir les muscles sains.

EN1520-1521 .... Kit complet avec boîtier, plaques et bat. .... 220,00 €

STIMULATEUR MUSCULAIRE



Tonifier ses muscles sans effort grâce à l'électronique. Tonifie et renforce les muscles (4 électrodes).
Le kit est livré complet avec son coffret sérigraphié mais sans sa batterie et sans électrode.

EN1408 ..... Kit avec boîtier ..... 96,35 €
Bat. 12 V 1.2 A ..... Batterie 12 V / 1,2 A ..... 15,10 €
PC1.5 ..... 4 électrodes + attaches ..... 28,00 €

LA IONOTHERAPIE: TRAITER ELECTRONIQUEMENT LES AFFECTIONS DE LA PEAU



Pour combattre efficacement les affections de la peau, sans aucune aide chimique, il suffit d'approcher la pointe de cet appareil à environ 1 cm de distance de la zone infectée. En quelques secondes, son "souffle" germicide détruira les bactéries, les champignons ou les germes qui sont éventuellement présents.

EN1480 .... Kit étage alimentation avec boîtier ..... 80,00 €
EN1480B . Kit étage voltmètre ..... 24,00 €
PIL12.1 .... Batterie 12 volts 1,3 A/h ..... 15,10 €

MAGNETOTHERAPIE BF (DIFFUSEUR MP90) A HAUT RENDEMENT



Très complet, ce kit permet d'apporter tous les "bienfaits" de la magnétothérapie BF. Par exemple, il apporte de l'oxygène aux cellules de l'organisme, élimine la cellulite, les toxines, les états inflammatoires, principales causes de douleurs musculaires et osseuses.
Fréquences sélectionnables : 6.25 - 12.5 - 25 - 50 - 100 Hz. Puissance du champ magnétique : 20 - 30 - 40 Gauss. Alimentation : 220 VAC.

EN1146 .... Kit complet avec boîtier et diffuseur... .....165,60 €
MP90 ..... Diffuseur supplémentaire. ....22,15 €

DIFFUSEUR POUR LA IONOPHORÈSE



Ce kit paramédical, à microcontrôleur, permet de soigner l'arthrite, l'arthrose, la sciatique et les crampes musculaires. De nombreux thérapeutes préfèrent utiliser la ionophorese pour inoculer dans l'organisme les produits pharmaceutiques à travers l'épiderme plutôt qu'à travers l'estomac, le foie ou les reins. La ionophorèse est aussi utilisée en esthétique pour combattre certaines affections cutannées comme la cellulite par exemple.

EN1365 .... Kit avec boîtier, hors batterie et électrodes ..... 95,60 €
PIL12.1 .... Batterie 12 V 1,3 A/h ..... 15,10 €
PC2.33x ... 2 plaques conduct. avec diffuseurs ..... 13,70 €

COMELEC
Tél. :04.42.70.63.90
Fax :04.42.70.63.95
www.comelec.fr
CD 908 - 13720 BELCODENE

DEMANDEZ NOTRE CATALOGUE 96 PAGES ILLUSTRÉES AVEC LES CARACTÉRISTIQUES DE TOUS LES KITS