

# EIR - Elektor Intern

## Puces haut de gamme pour écouter la

Harald Kipp et Thomas Scherer

**Il fut un temps où il fallait moduler, en analogique, les sons sur une porteuse haute-fréquence pour pouvoir ensuite les capter et en dériver quelque chose de plus ou moins audible ! Il en va tout autrement aujourd'hui : comprimer un signal audio, le mettre en paquets IP, « streamer », capter, tamponner, décompresser et vous voici le maître des radios Internet du monde. Tout cela grâce à un matériel ultra-haut de gamme que nous allons vous présenter ici...**

Une Webradio (Internet-Radio) est quelque chose de très spécial. Aucun récepteur Ondes Courtes, aussi sensible soit-il, est en mesure d'approcher, même de très loin, ce qu'elle offre en musique et informations sans même parler de la qualité. Comme les « Webradio » n'ont pas besoin de « cracher », dans l'éther, des centaines de kW de signaux HF (de l'électrosmog), la mise en oeuvre d'une station de ce type est à la portée (financière) de petits groupes d'utilisateurs. Il y a beaucoup de choses à dire au sujet de ce nouveau genre de radio (cf. l'encadré), mais plus important notre réponse à la question :

### Pourquoi pas 100% logiciel ?

Il faut commencer par dire qu'il existe les programmes les plus divers (WinAmp, iTunes, VLC, etc.) et cela pour tous les systèmes d'exploitation imaginables, permettant de se mettre à l'écoute des Netradios. Tout le monde a, en ce début de 21<sup>ème</sup> siècle, qui un PC, un Mac ou encore une machine tournant sous Linux chez soi. Pourquoi dépenserait-on alors de l'argent pour un appareil physique non virtuel, voire pour le réaliser de ses propres mains ?

Pour commencer, le matériel de base d'une radio logicielle consomme du courant et pas peu vu ce que l'on en attend. Passer des heures à écouter la radio via Internet sur son PC est un comportement on ne peut plus anti-écologique. L'approche proposée se débrouille

avec à peine 1 W d'énergie. À raison de 10 heures par jour, l'EIR s'est remboursé en un an à peine par les économies d'électricité qu'il permet par rapport à un PC haut de gamme (*Gamer*) utilisé pour écouter les Webradios.

Secundo, il existe des applications pour lesquelles un PC ne convient pas : lors de la connexion à une installation stéréo par exemple. Au contraire, une Internet-Radio à base d'*Open Source* pourra facilement être modifiée pour mieux répondre à des souhaits particuliers et, last but not least comme disent nos amis d'outre-Manche, EIR continue de fonctionner lorsque le PC s'est planté, voire crashé ;-)

### Le principe

S'agissant, dans le cas d'EIR, d'un projet complexe faisant appel aux technologies les plus modernes, il est impossible d'en englober tous les aspects dans un seul article. Nous allons donc, ici, passer en revue les aspects matériel, réalisation et utilisation. Vous trouverez d'autres informations dans des documents disponibles sur notre site (au fur et à mesure de leur disponibilité), [www.elektor.fr](http://www.elektor.fr) et sur la page du projet [1] (en anglais) voire dans de possibles articles publiés ultérieurement.

Nous n'allons pas vous apprendre qu'il faut capter un flux de données Internet-Radio, le tamponner et le décoder. Ceci requiert un microcontrôleur d'un certain « poids ». Comme nous le disions dans l'article du mois dernier [3],

c'est un ARM7 [4] qui a ici été mis à contribution.

Le schéma de la **figure 1** décrit l'environnement : la CPU trône en haut au centre ayant à sa disposition pas moins de 64 Moctets de SD-RAM, plus qu'il n'en faut pour le tampon et bien d'autres « extras ». Le progiciel (*firmware*) est implanté dans la CPU, les 4 Moctets de mémoire Flash permettant eux un stockage stable des données. Une horloge en temps réel (RTC) à sauvegarde par Supercap permet de réaliser un radio-réveil voire une autre application nécessitant une information d'heure exacte. Pour éviter de pousser l'ARM7 dans ses derniers retranchements, l'aspect décodage audio est confié à une puce spécialisée du type VS1053 [5].

En ce qui concerne les interfaces, EIR en est richement doté : on y découvre, outre l'indispensable Ethernet, il faut bien qu'EIR arrive, d'une façon ou d'une autre, sur Internet, une interface de programmation via USB, une interface sérielle ainsi qu'une interface JTAG (utile en cas de débogage) sans oublier 3 embases d'extension remontant jusqu'aux ports.

Il est en outre prévu un connecteur pour carte SD/MMC au cas où l'on voudrait enregistrer les émissions.

### Généralités

Les flux de données entrants sont, en règle générale, des données stéréo comprimées de façon à avoir, ty-



# et Radio

## radio



piquement, une résolution de 16 bits et une fréquence d'échantillonnage de 44,1 kHz, de façon qu'elles se contentent du dixième et avec, dans le meilleur des cas à 192 Kbits/s voire moins au lieu de 1,4 Mbits/s. On a donc besoin, pour créer un tampon d'une capacité de 10 s, de quelque 256 Koctets de RAM. Cela peut, de nos jours, paraître peu, mais pour un microcontrôleur cela reste une quantité « impressionnante ». Si l'on veut éviter de prendre des risques et que l'on souhaite disposer de possibilités d'extension et d'« extras » on a vite besoin de 512 Koctets voire plus. Le processeur ARM7 utilisé supporte la RAM SD de sorte qu'avec les 64 Moctets disponibles EIR n'a pas le moindre souci d'espace mémoire.

Le système d'exploitation choisi est Nut/OS, moins encombrant que Linux et se contentant de 40 Koctets. Tout compris, il faut au logiciel de l'ordre de 200 Koctets. En ce qui concerne les données, 1 Moctet fait largement l'affaire. La CPU disposant elle-même de 512 Koctets de Flash pour le logiciel et qu'il y a de la RAM à ne plus savoir qu'en faire, il ne doit pas y avoir de problème. L'ensemble du logiciel est Open Source à l'exception du programme Flash d'Atmel.

À noter que le contrôleur est suffisamment puissant pour utiliser la carte SD en parallèle pour l'enregistrement d'un second *stream* audio. Il ne se passera sans doute pas très longtemps avant qu'un auteur de la communauté Open

Source ajoute cette fonction et d'autres extras.

Pour éviter de forcer les extensions dans une direction ou dans une autre, la carte ne comporte pas d'organe de commande explicite tels que touches ou affichage. Les embases d'extension présentes permettent, sans le moindre problème, de telles adjonctions. Dès le départ, EIR a été conçu pour se voir connecter des extensions propres à son réalisateur, raison pour laquelle le progiciel est prévu pour être piloté depuis une page Web intégrée. Le progiciel étant totalement ouvert, il n'y a aucune raison pour que cela ne change pas...

Source ajoute fonction et d'autres extras.

### Détails

Un coup d'oeil au schéma de la **figure 2** suffit pour se convaincre de la complexité du projet. Pour cette raison, la description qui suit s'oriente aux différents blocs fonctionnels :

#### • Ethernet

Internet aborde la carte au travers d'une embase Ethernet à transformateur intégré et 2 LED. La LED verte d'allume lors du transfert de données, la LED jaune signalant l'existence d'une connexion. C'est une puce spécialisée, un DM9000E (IC10), qui se charge du trafic Ethernet. Le tampon IC9 permet l'utilisation de l'entrée WAIT de la CPU par d'éventuelles extensions.

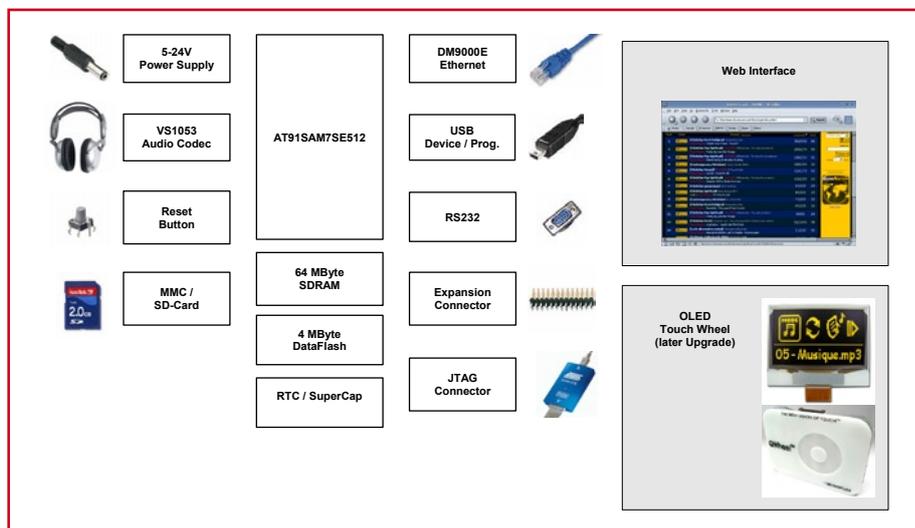
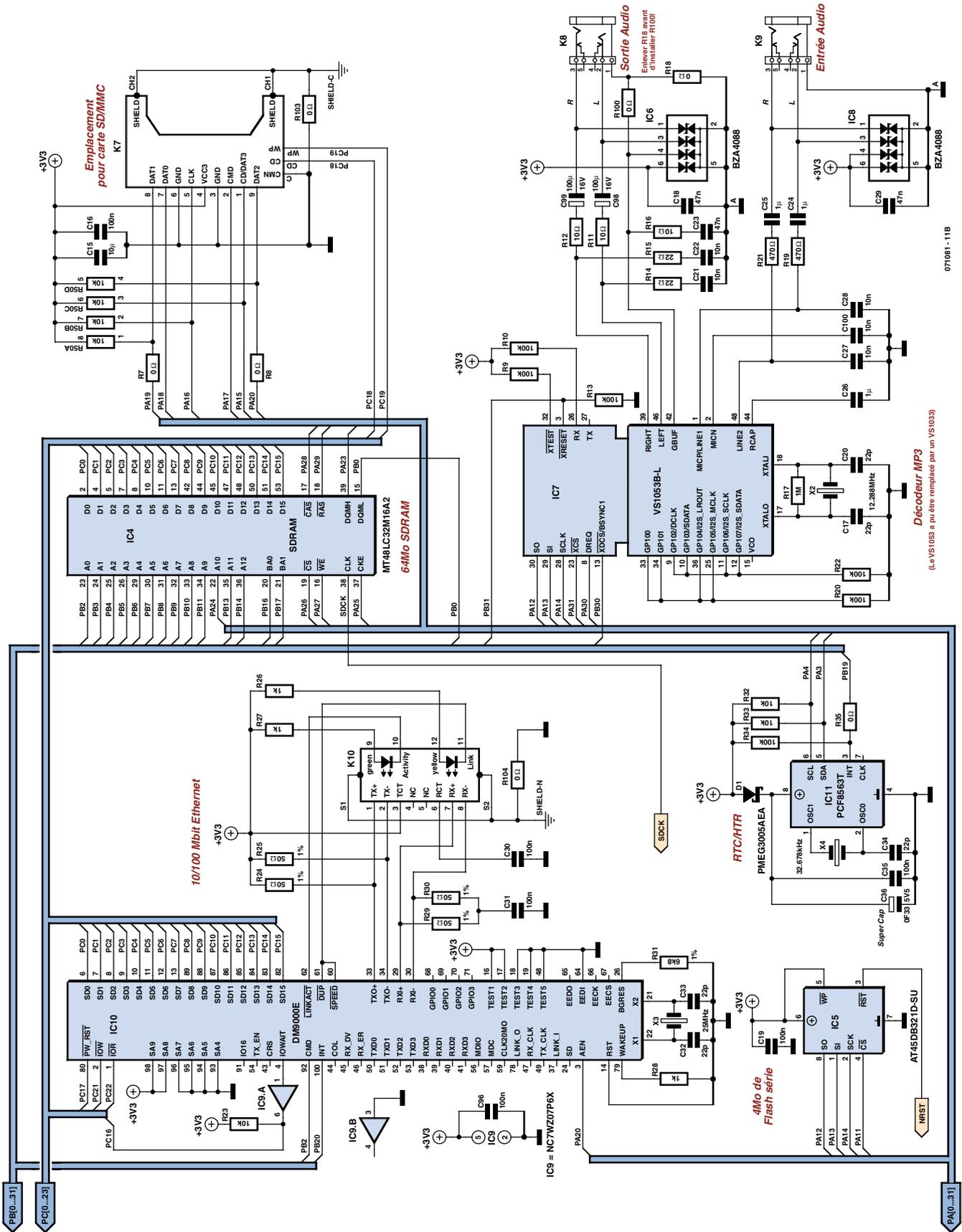


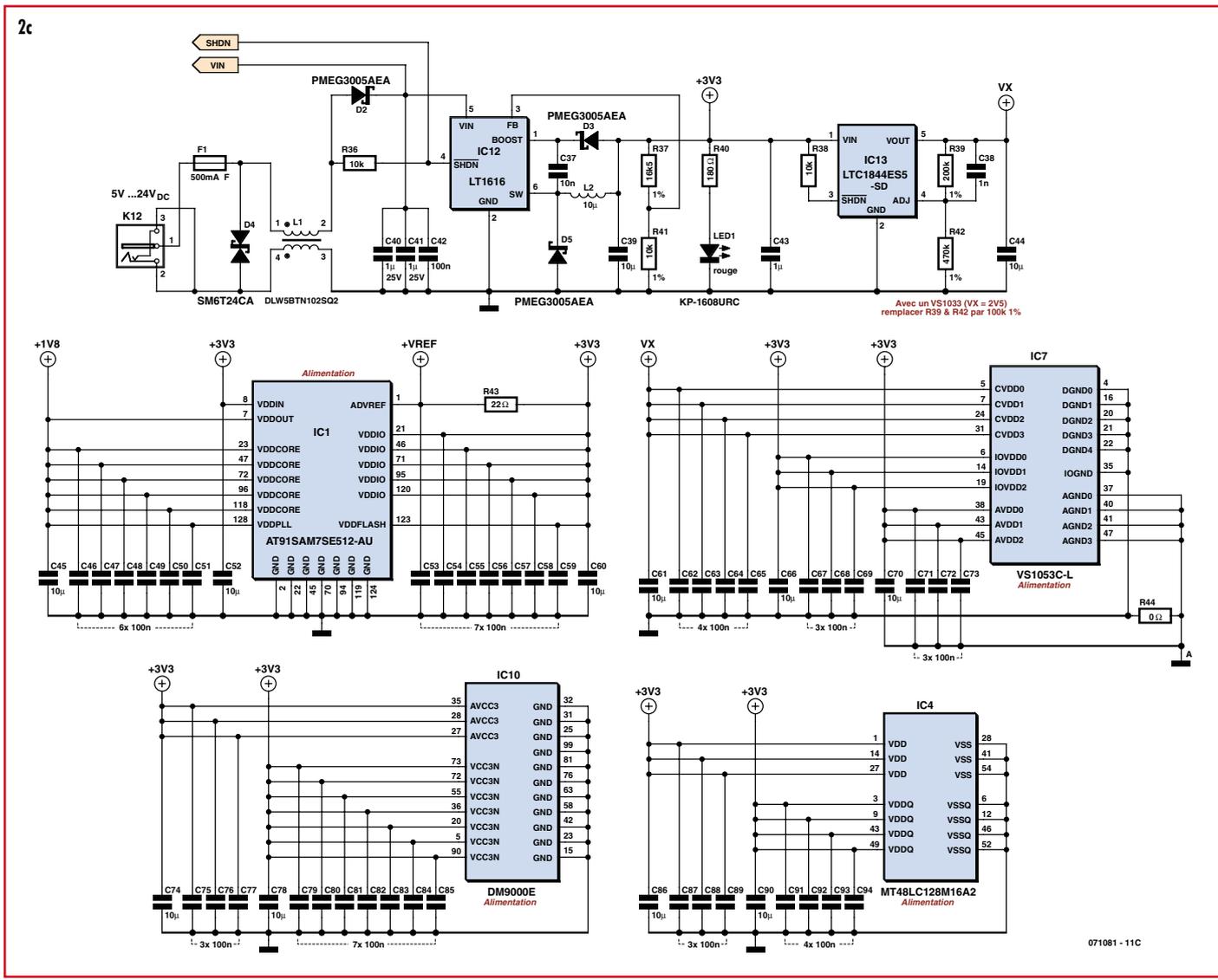
Figure 1. Synoptique d'EIR (Elektor Internet Radio).





077081 - 118

2c



● **Décodeur audio**

S'il est vrai qu'un ARM7 devrait être en mesure (tout juste) d'effectuer un décodage logiciel de données MP3 ou AAC, un circuit dédié tel que IC7 soulage fortement la CPU sans parler de sa capacité de s'accommoder, outre des variantes MP3 classiques, également de données HE-AAC voire même Ogg-Vorbis. Cela permet en outre de simplifier le logiciel requis. Sur nos prototypes nous avons utilisé un échantillon Vorab de VLSI (VS1053). En cas de problèmes d'approvisionnement il serait possible d'utiliser la variante VS1033 (sans Ogg-Vorbis). Bien que la CPU soit dotée d'une sortie 1,8 V utilisable pour l'alimentation de périphériques, nous avons préféré, pour des raisons de stabilité, doter IC7 de son propre régulateur de tension. La variante VS1033 nécessitant une tension de 2,5 V, cela impliquera de faire passer les résistances R39 et R42 à 100 kΩ.

● **Flash additionnelle**

La mise en oeuvre de la radio requiert la mémorisation de nombreux paramètres dont on doit disposer même après une coupure de courant - la liste de stations en particulier. Il aurait été possible d'utiliser la mémoire Flash interne de la CPU, mais son écriture est délicate. Pour nous simplifier la vie, nous avons ajouté une mémoire Flash sérielle, IC5, dont les 4 Mottets suffisent largement au stockage de longues listes de stations et à bien d'autres choses.

● **Alimentation**

La réduction de la consommation d'énergie de EIR au strict minimum nous a amené à la doter d'un régulateur à découpage centré sur IC12. On dispose, pour une tension d'entrée comprise entre 5 et 24 V, d'une puissance de 5 W sous 3,3 V. EIR ne consommant que 1 W, il nous reste suffisamment d'énergie pour d'éventuelles extensions de matériel.

● **Soudures**

Nous trouvant en présence d'une platine multi-couche équipée de CMS minuscules (cf. figures 3 et 4) et dont certains circuits intégrés ont un écartement de broches de 0,5 mm seulement, Elektor a choisi de vous proposer une platine dotée de tous les composants CMS (avec un VS1053). Il reste donc à implanter les composants de taille normale. On évitera ainsi des erreurs difficiles à identifier. Rien ne vous interdit cependant, en utilisant les dessins de pistes proposés, de réaliser ce projet totalement vous-même.

**Test de fonctionnement**

Il faudra, pour un premier test de l'alimentation, charger le côté 3,3 V de quelques mA - et donc ne pas la laisser tourner à vide. À partir de 4 V en entrée, le régulateur devrait se mettre au travail et consommer, en fonction de la charge, entre 50 et 150 mA. Dans le cas

d'une tension de 24 V en entrée, le courant ne sera plus que de 30 à 50 mA. Si tout est OK, la LED LED1 s'allume. Une fois les circuits intégrés mis en place, on pourra vérifier à l'oscilloscope le fonctionnement du quartz. Si X1 oscille, la CPU devrait être accessible. La CPU est dotée, à l'origine, d'une amorce (*bootloader*) permettant, outre le transfert d'un nouveau progiciel, la communication bidirectionnelle avec la RAM et la mémoire Flash. Atmel propose à l'adresse [6], le fichier « AT91-SIP.exe » intégrant, après décompaction, le programme Windows « SAM-BA ». Après avoir installé ce dernier on connecte EIR par USB à un PC. Dès la mise sous tension d'EIR, Windows devrait activer automatiquement le pilote correspondant. On peut alors démarrer SAM-BA. Les paramètres à choisir sont alors USB (type de liaison) et « AT91SAM7SE512-EK » pour le type de périphérique, EIR lui étant très proche.

On trouvera, sur le site Elektor, à la page dédiée à cet article, un progiciel simple utilisable à des fins de test. Il permet, si tant est que la CPU et l'interface série fonctionnent, de s'assurer du fonctionnement d'autres périphériques tels qu'Ethernet et le décodeur audio. Après transfert du progiciel

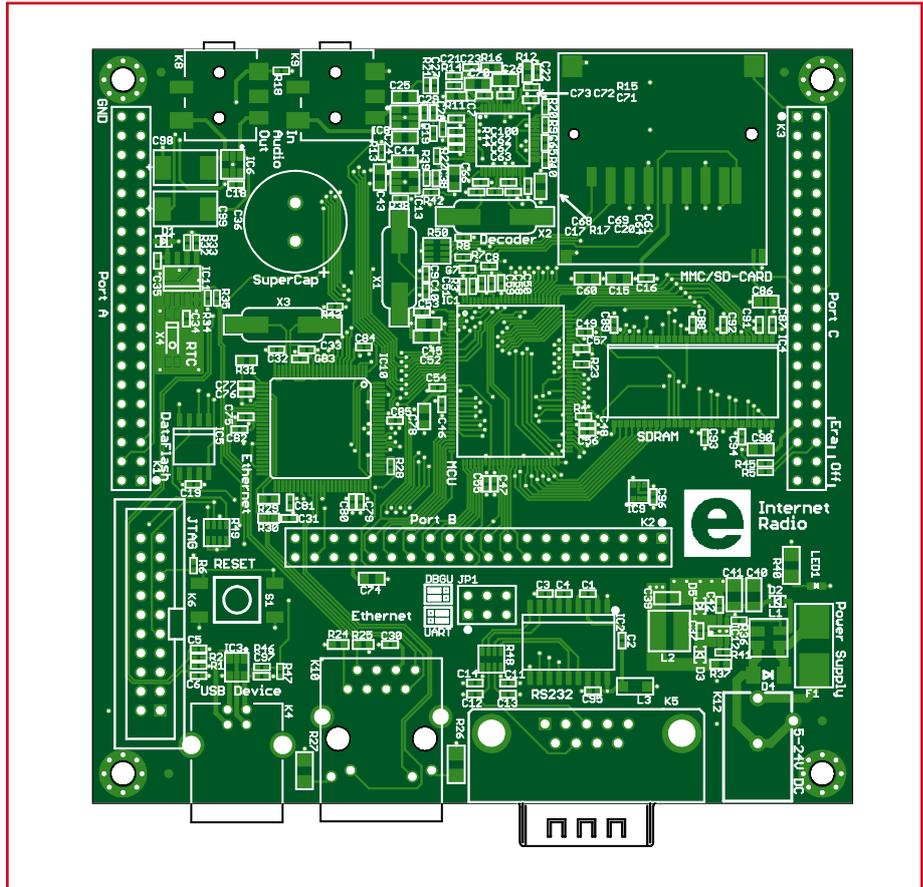


Figure 3. Implantation des composants d'EIR. Il vous est proposé une platine dotée de ses composants CMS.

## Liste des composants

### Résistances :

- R1, R2 = 27 Ω CMS 0402
- R3 = 1kΩ5 CMS 0402
- R4, R5, R28, R45 = 1 kΩ CMS 0402
- R6, R23, R32, R33, R36, R38 = 10 kΩ CMS 0402
- R7, R8, R18, R35, R44 = 0 Ω CMS 0402
- R9, R10, R13, R20, R22, R34 = 100 kΩ CMS 0402
- R11, R12, R16 = 10 Ω CMS 0603
- R14, R15, R43 = 22 Ω CMS 0402
- R17 = 1 MΩ CMS 0402
- R19, R21 = 470 Ω CMS 0402
- R24, R25, R29, R30 = 50 Ω/1% CMS 0402
- R26, R27 = 1 kΩ CMS 1206
- R31 = 6kΩ8/1% CMS 0603
- R37 = 16kΩ5/1% CMS 0603
- R39 = 200 kΩ\*/1% CMS 0402
- R40 = 180 Ω CMS 1206
- R41 = 10 kΩ/1% CMS 0402
- R42 = 470 kΩ\*/1% CMS 0402
- R46 = 15 kΩ CMS 0402
- R47 = 22 kΩ CMS 0402
- R48 = 22 Ω réseau CAY16
- R49 = 100 kΩ, réseau CAY16
- R50 = 10 kΩ, réseau CAY16
- R100 à R106 = 0 Ω\* CMS 1206 (non requis)
- \* Cf. texte

### Condensateurs :

- (CMS céramique 6,3 V sauf mention contraire)

- C1 à C4, C16, C19, C30, C31, C35, C42, C46 à C51, C53 à C59, C62 à C65, C67 à C69, C71 à C73, C75 à C77, C79 à C85, C87 à C89, C91 à C97 = 100 nF (CMS 0402)
- C5, C6, C9, C10, C17, C20, C32, C33, C34 = 22 pF (CMS 0402)
- C7, C38 = 1 nF (CMS 0402)
- C8, C21, C22, C27, C28, C37, C100 = 10 nF (CMS 0402)
- C11 à C14 = 220 pF (CMS 0402)
- C15, C39, C44, C45, C52, C60, C61, C66, C70, C74, C78, C86, C90 = 10 μF (CMS 0805)
- C18, C23, C29 = 47 nF (CMS 0402)
- C24 à C26, C43 = 1 μF (CMS 0805)
- C36 = 0,1 μF capacité double couche FGOH104Z135
- C40, C41 = 1 μF/25 V (CMS 1206)
- C98, C99 = 100 μF/16 V tantale (CMS)

### Inductances :

- L1 = DLW5BTN102SQ2 (Murata)
- L2 = 10 μH MSS5131 (Coilcraft)
- L3 = BLM31A (Murata)

### Semi-conducteurs :

- D1 à D3, D5 = PMEG3005AEA (Philips)
- D4 = SM6T24CA (STM)
- IC1 = AT91SAM7SE512-AU (Atmel)
- IC2 = MAX3222ECWN (Maxim)
- IC3, IC6, IC8 = réseau de diodes BZA408B
- IC4 = MT48LC32M16A2
- IC5 = AT45DB321D-SU (Atmel)
- IC7 = VS1053C-L (VLSI)\*
- IC9 = NC7WZ07P6X (Fairchild)
- IC10 = DM9000E (Davicom)

- IC11 = PCF8563T (Philips)
- IC12 = LT1616 (Linear Technology)
- IC13 = LTC1844ES5-SD (Linear Technology)
- LED1 = LED rouge KP-1608URC (CMS 0603, Kingbright)

### Divers :

- X1 = quartz 18,432 MHz (CMS HC49SM)
- X2 = quartz 12,288 MHz (CMS HC49SM)
- X3 = quartz 25,000 MHz (CMS HC49SM)
- X4 = quartz 32,678 kHz (CMS MC-146)
- F1 = porte-fusible + fusible 0,5 A rapide CMS OMNI-BLOK (Littelfuse)
- K1 à K3 = embase autosécable à 2 rangées de 20 contacts au pas de 2,54 mm
- K4 = embase USB-B (AMP-787780)
- K5 = embase Sub-D en équerre à 9 contacts aux normes US
- K6 = embase HE10 à 2 rangées de 10 contacts au pas de 2,54 mm
- K7 = connecteur pour carte SD CMS FPS009-2700 (Yamaichi)
- K8, K9 = embase jack 3,5 mm stéréo CMS SJ1-3515 (CUI)
- K10 = embase RJ-45 avec transformateur Ethernet et LED (CMS RJLD-043TC, Taimag)
- K12 = embase CC avec fiche 2 mm TDC-002-3
- JP1 = embase autosécable à 2 rangées de 3 contacts au pas de 2,54 mm + 2 cavaliers
- S1 = bouton-poussoir (CMS LSH, Schurter) platine vierge (EPS071081-1) ou platine dotée des CMS (EPS071081-1) logiciel téléchargeable depuis site Elektor



Figure 4. Cet exemplaire terminé prouve sans ambiguïté qu'une réalisation personnelle est loin d'être une sinécure.

ciel il reste à informer EIR qu'il lui faudra, au prochain démarrage, s'amorcer sous ce progiciel. Il faudra pour cela, sous « *Scripts* » choisir la routine « *Boot from Flash*

(*GPNVM2*) » et cliquer sur « *Execute* » - clore SAM-BA et appuyer sur la touche de réinitialisation (*Reset*). EIR est alors reliée à un PC pas son interface série et un câble modem nul (bro-

Tableau 1. Erweiterungsstecker K1					
Broche	Signal	Fonction	Broche	Signal	Fonction
1	PA0	Libre	2	PA1	Libre
3	PA2	Libre	4	PA3	SDA TWI
5	PA4	SCL TWI	6	PA5	RxD UART0 via JP1
7	PA6	TxD UART0 via JP1	8	PA7	RTS UART0
9	PA8	CTS UART0	10	PA9	RxD DEBUG via JP1
11	PA10	DEBUG TxD via JP1	12	PA11	Chip Select DataFlash
13	PA12	SPI MISO	14	PA13	SPI MOSI
15	PA14	SPI SPCK	16	PA15	Chip Select MMC
17	PA16	Horloge MMC	18	PA17	Commande MMC
19	PA18	DAT0 MMC	20	PA19	MMC DAT1 via R7
21	PA20	DAT2 MMC via R8	22	PA21	Libre
23	PA22	Libre	24	PA23	DQMH SDRAM
25	PA24	A10 SDRAM	26	PA25	CKE SDRAM
27	PA26	Chip Select SDRAM	28	PA27	WE SDRAM
29	PA28	CAS SDRAM	30	PA29	RAS SDRAM
31	PA30	IRQ1 - Interruption MP3	32	PA31	Command Select MP3
33	Vref	Référence CAN	34	3.3V	Alimentation
35	AD4	Entrée analogique libre	36	AD5	Entrée analogique libre
37	AD6	Entrée analogique libre	38	AD7	Entrée analogique libre
39	GND	Masse	40	GND	Masse



Figure 5. Recopie d'écran de SAM-BA sous Windows 2000.

ches 2 et 3 interconnectées); un émulateur de terminal (nous recommandons **TeraTerm** [7] pour Windows, **Miniterm** pour Linux, un Mac disposant de toutes façons d'un Terminal) permet alors, à 115,2 kbaud et un paramétrage de 8/0/1 (bits de données, parité et d'arrêt), de suivre ce que fait EIR.

### Écouter le radio

Il faut, avant de pouvoir le faire, débarasser EIR du progiciel de test et le doter du progiciel de radio. Pour permettre le chargement du progiciel il faut commencer par ponter par cavalier les contacts 34 et 36 de K3, appuyer sur le bouton de réinitialisation et enlever le cavalier ensuite. EIR redémarre avec l'amorce et SAM-BA permet d'y transférer le progiciel de radio.

Il reste alors, par Ethernet, à relier EIR au réseau local (par *hub/switch* ou routeur Internet multi-ports) et de connecter un casque d'écoute ou un amplificateur à sa sortie audio.

### Bild 6, Bild 7

Si le réseau local (LAN) ou le routeur disponible dispose d'un serveur DHCP activé, EIR en obtient une adresse valide et se met à l'écoute des stations radio pré-paramétrées. En cas de préférence d'adresses IP fixes : lors de l'installation de Nut/OS il y a eu installation, sur le PC, d'un petit outil, « **Discover** » qui permet dans tous les cas de figures de trouver EIR (**figure 6**) et de paramétrer l'adresse IP souhaitée. Comme le montre le **figure 7**, on entre dans le champ Gateway l'adresse du routeur. Ensuite, l'écoute de la radio (**figure 8**) devrait fonctionner même avec des adresses IP fixes.

### Perspectives



Figure 6. Ce programme (tournant ici sous Linux KDE) permet de retrouver EIR même en cas d'adresse IP inconnue.

**Tableau 2. Embase d'extension K2**

Broche	Signal	Fonction	Broche	Signal	Fonction
1	PB0	DQML SDRAM	2	PB1	Libre
3	PB2	Bus d'adresse A2	4	PB3	Bus d'adresse A3
5	PB4	Bus d'adresse A4	6	PB5	Bus d'adresse A5
7	PB6	Bus d'adresse A6	8	PB7	Bus d'adresse A7
9	PB8	Bus d'adresse A8	10	PB9	Bus d'adresse A9
11	PB10	Bus d'adresse A10	12	PB11	Bus d'adresse A11
13	PB12	Libre	14	PB13	Bus d'adresse A13
15	PB14	Bus d'adresse A14	16	PB15	Libre
17	PB16	SDRAM BAO	18	PB17	SDRAM BA1
19	PB18	Libre	20	PB19	FIQ - Interruption RTC
21	PB20	IRQ0 - Interruption Ethernet	22	PB21	Libre
23	PB22	Chip Select DataFlash	24	PB23	Moniteur USB
25	PB24	Libre	26	PB25	Libre
27	PB26	Libre	28	PB27	Libre
29	PB28	Libre	30	PB29	Libre
31	PB30	Sélection Données MP3	32	PB31	Reset Hardware MP3
33	3,3 V	Alimentation	34	3,3 V	Alimentation
35		NC	36		NC
37		NC	38	NRST	Reset Matériel
39	GND	Masse	40	GND	Masse

EIR est, nous l'avons mentionné, un concept totalement ouvert. Il est possible de modifier le logiciel ou de doter le matériel d'extensions (grâce aux embases d'extension). Il est fort probable que vous retrouverez, dans le futur, l'une ou l'autre mention à son sujet.

On pourra, pour s'informer de l'existence d'outils logiciels ou de développements ultérieurs, jeter un coup d'oeil à la page consacrée à ce projet [1] chez egnite. On y trouvera le fichier-source et les installations pour Windows, Linux et OSX. On y trouvera également des liens d'environnements de développement et de projets Open Source en évolution.

L'imagination est reine. On pourrait penser à quelques touches et un affichage LCD de sorte que l'on pourrait utiliser EIR non seulement en browser Internet, mais même en appareil auto-

## Internet-Radio

Un petit tour sur Internet a de quoi époustoufler : à l'entrée du mot-clé Internetradio dans Google se traduit par plus de 2,3 millions de hits - indubitablement un sujet d'actualité. Les premiers essais dans ce domaine, « émissions » par paquets, datent de 1993, quasi-parallèlement à l'apparition du premier browser opérationnel, Mosaic de NCSA et pour ainsi dire à l'heure zéro du commerce sur Internet. Il y déjà longtemps que les « vraies » stations de radio avaient choisi de diffuser, par le biais d'Internet aussi, sous forme de « streaming », leurs émissions limitées jusqu'à présent à une diffusion via les éthers. Aujourd'hui, une connexion Internet standard permet la réception de dizaines de milliers de programmes radio. Il est possible actuellement, parallèlement à une quantité industrielle de programmes thématiques, de se mettre à l'écoute de la quasi-totalité des stations de radio commerciales.

La dénomination de « streaming » recouvre un flux de données de contenu audio ou vidéo le plus continu possible et émis en temps réel, le côté émetteur devant fournir un stream par client, ce qui pourrait

entraîner un trafic important et pourrait, en cas de forte écoute, se traduire par un coût (assez) élevé. De manière à garder les taux de transfert dans des limites acceptables, on procède, avant émission, à une compression des données (le plus souvent avec un minimum de pertes) pour les décompresser à l'arrivée côté récepteur. Une radio Internet devra donc - qu'elle soit purement logicielle, ou prenne la forme de matériel dédié - disposer de l'un ou l'autre décodeur de streaming, tel que soit MP3, Ogg Vorbis voire Real Audio.

Vu d'autre part que sur Internet les protocoles classiques HTTP et FTP ne peuvent pas garantir des durées stables des différents paquets de données, le récepteur doit disposer d'un tampon de données de capacité suffisante, ce qui a pour effet de retarder la réception de quelques secondes, la réception n'étant plus que « quasi-live ». Il devient difficile ainsi de « zapper » d'une émission à l'autre. La numérisation présente cependant les avantages d'une qualité de son stable, d'une portée mondiale et comparé à la radiodiffusion classique, d'un nombre, quasiment infini de programmes. À cela s'ajoute, en principe, la possibilité de réception de « conserves » (= émissions passées) sous forme de « Audio on Demand » ce que ne peut aucune radio conventionnelle.

**Tableau 3. Embase d'extension K3**

Broche	Signal	Fonction	Broche	Signal	Fonction
1	PC0	Bus de données D0	2	PC1	Bus de données D1
3	PC2	Bus de données D2	4	PC3	Bus de données D3
5	PC4	Bus de données D4	6	PC5	Bus de données D5
7	PC6	Bus de données D6	8	PC7	Bus de données D7
9	PC8	Bus de données D8	10	PC9	Bus de données D9
11	PC10	Bus de données D10	12	PC11	Bus de données D11
13	PC12	Bus de données D12	14	PC13	Bus de données D13
15	PC14	Bus de données D14	16	PC15	Bus de données D15
17	PC16	NWAIT bus, Collecteur ouvert	18	PC17	Reset Hardware Ethernet
19	PC18	Détection Carte MMC	20	PC19	Write Protect MMC
21	PC20	Libre	22	PC21	NWE bus d'adresses/de données
23	PC22	NRD bus d'adresses/de données	24	PC23	Chip Select Ethernet
25		NC	26		NC
27		NC	28		NC
29		NC	30		NC
31		NC	32		NC
33	3,3 V	Alimentation	34	3,3 V	Alimentation
35	JTAGSEL	Validation Boundary Scan	36	ERASE	Effacement Firmware
37	VIN	5-24V non régulé via R106	38	SHDN	Coupure Alimentation
39	GND	Masse	40	GND	Masse

nome de luxe. L'existence du connecteur pour carte SD appelle à s'en servir en lecteur MP3.

(071081-1)

## Bibliographie et Liens :

[1] Page Web de projets de egnite : [www.ethernet.de/en/hardware/eir/index.html](http://www.ethernet.de/en/hardware/eir/index.html)

[2] Article Wikipedia : <http://fr.wikipedia.org/wiki/Webradio>

[3] Ethernet et la famille Kipp : Elektor N° 357, page 28 et suivantes

[4] Informations au sujet de l'ARM7 : [www.atmel.com/products/at91/](http://www.atmel.com/products/at91/)

[5] Informations concernant le VS1053 : [www.vlsi.fi/en/products/vs1053.pdf](http://www.vlsi.fi/en/products/vs1053.pdf)

[6] Lien vers AT91-ISP.exe : [www.atmel.com/dyn/resources/prod\\_documents/Install%20AT91-ISP%20v1.10.exe](http://www.atmel.com/dyn/resources/prod_documents/Install%20AT91-ISP%20v1.10.exe)

[7] Terminal Windows : [tssh2.sourceforge.jp/](http://tssh2.sourceforge.jp/)