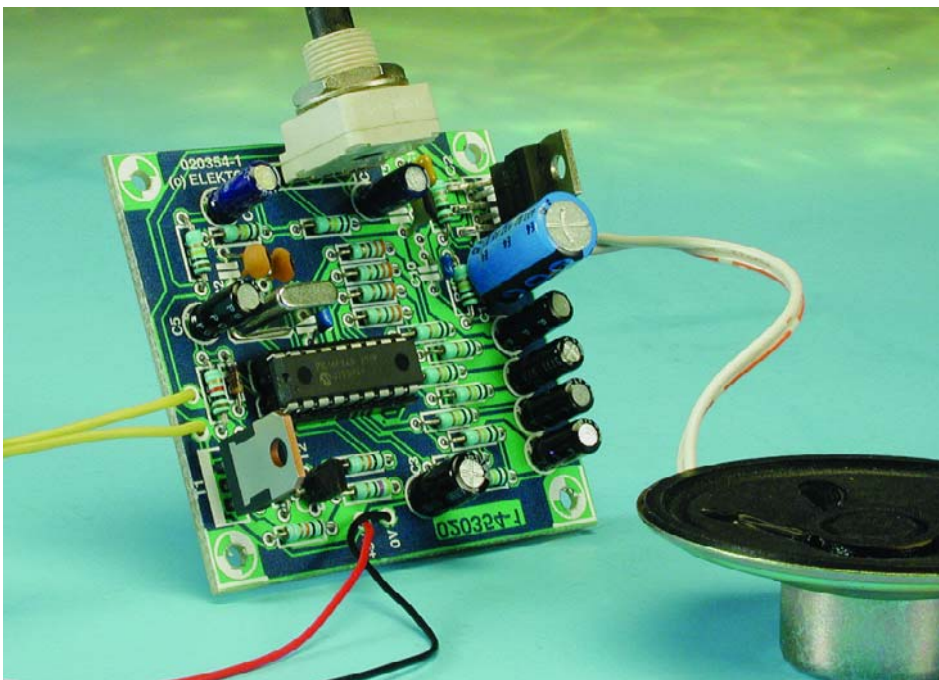


Carillon de porte

Polyphonique et multi-mélodies à PIC 16F84

Daniel Manier

« La musique est l'art de combiner des sons de manière agréable à l'oreille ... » a dit quelqu'un de célèbre. En ce début de nouveau millénaire, nous allons tenter de confier cette tâche délicate à, vous l'avez deviné, au microcontrôleur le plus populaire de l'écurie de Microchip, un PIC 16F84.



Voici un des rares montages générateurs de musique à base de microcontrôleur configurable par l'utilisateur. Il vous étonnera par ses performances et la qualité du son (polyphonique) qu'il fournit. Bien sûr il existe sur le marché divers carillons de porte mais celui-ci, en plus de l'intérêt d'une réalisation propre, possède des qualités non négligeables :

- Très bonne sonorité
- 8 mélodies différentes
- consommation nulle au repos
- musique personnalisable
- coût de réalisation très abordable

Le concept

Cette électronique musicale devra remplir 4 fonctions distinctes :

1. Gérer sa propre alimentation de manière à être sous tension dès le moindre appui sur le bouton « sonnette » et à ce qu'à l'état de repos (à la fin de la mélodie), la consommation de l'a circuiterie électronique redevienne nulle.
2. Générer des notes de musique

(jusque quatre à la fois), produire les enveloppes de ces notes et les mixer, ceci à partir de la partition musicale se trouvant en mémoire.

3. Mémoriser le numéro du dernier morceau de musique interprété afin d'en changer à la prochaine sonnerie, mémorisation se faisant en EEPROM afin de conserver l'information même en l'absence de tension d'alimentation.
4. Amplifier le signal pour le transmettre au haut-parleur.

Les trois premières fonctions (gestion de l'alimentation, musique, mémorisation) sont confiées au microcontrôleur PIC 16F84. L'amplification est réalisée par un circuit intégré spécialisé, un TDA2006. Le réglage du paramétrage de cet amplificateur se limite en fait au positionnement du potentiomètre de manière à éviter la saturation. Le haut-parleur connecté en sortie aura une impédance comprise entre 4 et 8 ohms.

Notions de base

Quelques notions de musique: lorsqu'un musicien interprète un morceau de musique, il lit une partition. C'est sur cette partition que sont écrites les notes à jouer ainsi que leurs durées respectives.

La position de la note sur la portée (nos fameuses 5 lignes) détermine le

nom et partant la hauteur) de la note. Ici les 4 notes de la **figure 1** sont, successivement, un LA, un SOL, un DO et un FA.

La représentation de la note (pleine ou évidée, avec ou sans queue, avec ou sans crochet), détermine sa durée. Ici nous avons, dans l'ordre, une noire, une croche, une ronde et une blanche.

En durée, une ronde dure deux fois plus longtemps qu'une blanche, une blanche dure deux fois plus longtemps qu'une noire et une noire deux fois plus longtemps qu'une croche.

Il existe bien d'autres durées et bien d'autres notions de musique mais ce paragraphe n'est qu'une approche condensée abordant les notions importantes dans le cadre de ce montage, à savoir celles de note et de durée.

Les partitions des morceaux de musique correspondant aux différentes sonneries sont mémorisées dans le PIC programmé sous la forme de matrices (tableaux) associant note et durée.

Une gamme chromatique tempérée se compose de douze notes (DO, DO dièse, RÉ, RÉ dièse, MI, FA, FA dièse, SOL, SOL dièse, LA, LA dièse, SI). Ces notes ont des fréquences allant en progression géométrique

d'une raison de racine douzième de 2 (1,05962...). Pour générer ces notes on pourra utiliser dans le programme des décompteurs que l'on chargera avec une valeur correspondant à la fréquence à obtenir. À chaque fois qu'un décompteur atteindra la valeur zéro il suffira d'inverser l'état d'une ligne de port (RB0 à RB3 pour 4 notes) pour générer un signal carré tel celui représenté en **figure 2**.

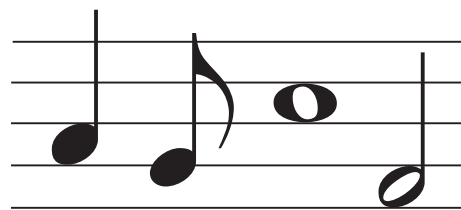
Synoptique

Nous allons prendre dans l'ordre les blocs présents sur le synoptique de ce montage représenté en **figure 3**.

Le montage se subdivise en 4 sous-ensembles majeurs : l'alimentation, le cerveau et la mémoire, sous la forme du PIC16F84, qui prend en fait en compte la totalité des fonctions actives, un mélangeur (*mixer*) et l'amplificateur audio chargé d'attaquer le haut-parleur reproduisant la mélodie.

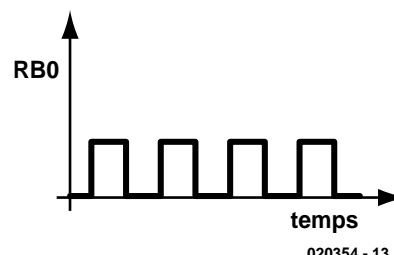
L'alimentation

La mise en route de la sonnerie s'effectue en appuyant sur le bouton poussoir S1. Le transistor T2 devient alors conducteur et rend le transistor T1 conducteur à son tour. T1 étant passant, une tension de 5,1 volts apparaît aux bornes de la diode zener D1. Cette tension filtrée par C4



020354 - 12

Figure 1. Le ba-a-ba de la musique : une portée avec les 4 types de notes les plus courants.



020354 - 13

Figure 2. Signal carré disponible sur la ligne de port RB0.

et C5 alimente alors le microcontrôleur PIC 16F84. D'autre part la tension disponible sur le collecteur de T1 est envoyée vers l'amplificateur audio, le TDA2006.

Dès le démarrage le PIC met au niveau haut (« 1 ») la sortie RA3 envoyant ainsi une tension de 5 volts sur la base de T2 par l'intermédiaire de R6 et autorisant alors le relâchement du bouton-poussoir. C'est à la fin de la sonnerie que le PIC fait retomber au niveau bas (« 0 ») la sortie RA3 et si le bouton-poussoir est relâché, T1 et T2 se bloquent provoquant la mise hors-tension de la carte.

Le générateur de notes

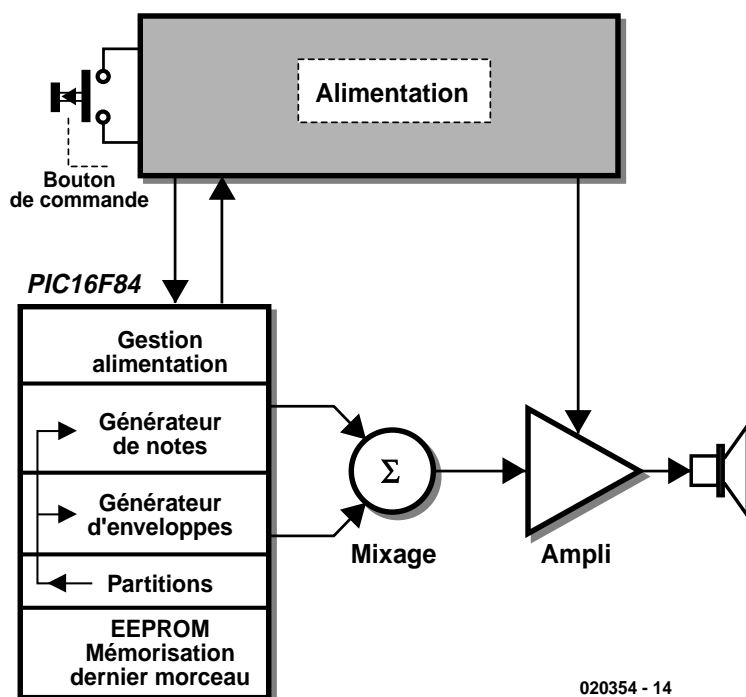
Le programme est capable ici de générer 4 notes simultanément, ceci s'appelle la polyphonie.

La génération des notes se fait dans le programme principal afin d'avoir une vitesse d'exécution maximum.

Pour ce qui est de la durée, c'est le temporisateur (*timer*) du PIC qui rythme l'interprétation. Un décompteur chargé avec une valeur correspondant à la durée de la note est décrémenté à chaque interruption.

Lorsque ce décompteur arrive à la valeur zéro, il y a passage à la note suivante et à la durée suivante.

Le son d'un signal carré est riche en harmoniques, ce qui améliore sa portée, mais pour donner à notre carillon de porte un son plus agréable à l'oreille et moins monotone, nous l'avons doté d'un dispositif baptisé



020354 - 14

Figure 3. Synoptique du carillon de porte polyphonique.

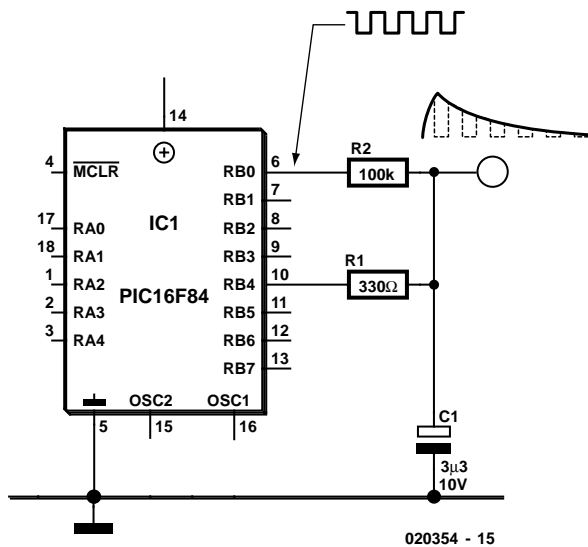


Figure 4. Le principe du générateur d'enveloppe.

Générateur d'enveloppe

Le principe de ce générateur est représenté dans la **figure 4**.

À chaque début de note, le condensateur C8 est chargé durant une durée très brève à travers une résistance de faible valeur (330 Ω) reliée à une ligne de port, ce dernier étant configuré en sortie (RB4 à RB7), puis cette ligne de port est immédiatement basculée en entrée c'est-à-dire mise à haute impédance

empêchant ainsi le condensateur de se décharger.

Les générateurs de notes (RB0 à RB3) sont reliés aux générateurs d'enveloppes par des résistances de forçage de valeur élevée (100 kΩ). Pour obtenir la courbe représentée, il aurait fallu que les sorties RB0 à RB3 soient des sorties à collecteur ouvert (style RA4), or ce n'est pas le cas pour RB0 à RB3 ;

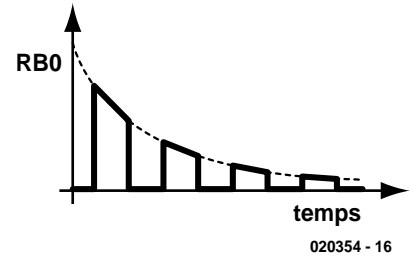


Figure 5. Le générateur d'enveloppe donne du « corps » au signal carré.

ce mode de fonctionnement a été obtenu en faisant fonctionner RB0 à RB3 de la manière suivante :

- Port en sortie s'il est à « 0 »
- Port en entrée s'il est à « 1 »

Une ligne paramétrée en sortie et présentant un niveau bas correspond bien à un transistor à collecteur ouvert « sortant » un zéro. En entrée, la ligne correspondante est à haute impédance et c'est la résistance de forçage de la ligne de port qui impose un état « 1 ». C'est à travers ces résistances de tirage de forte valeur (100 kΩ) que les condensateurs du générateur d'enveloppes vont de décharger. Ici

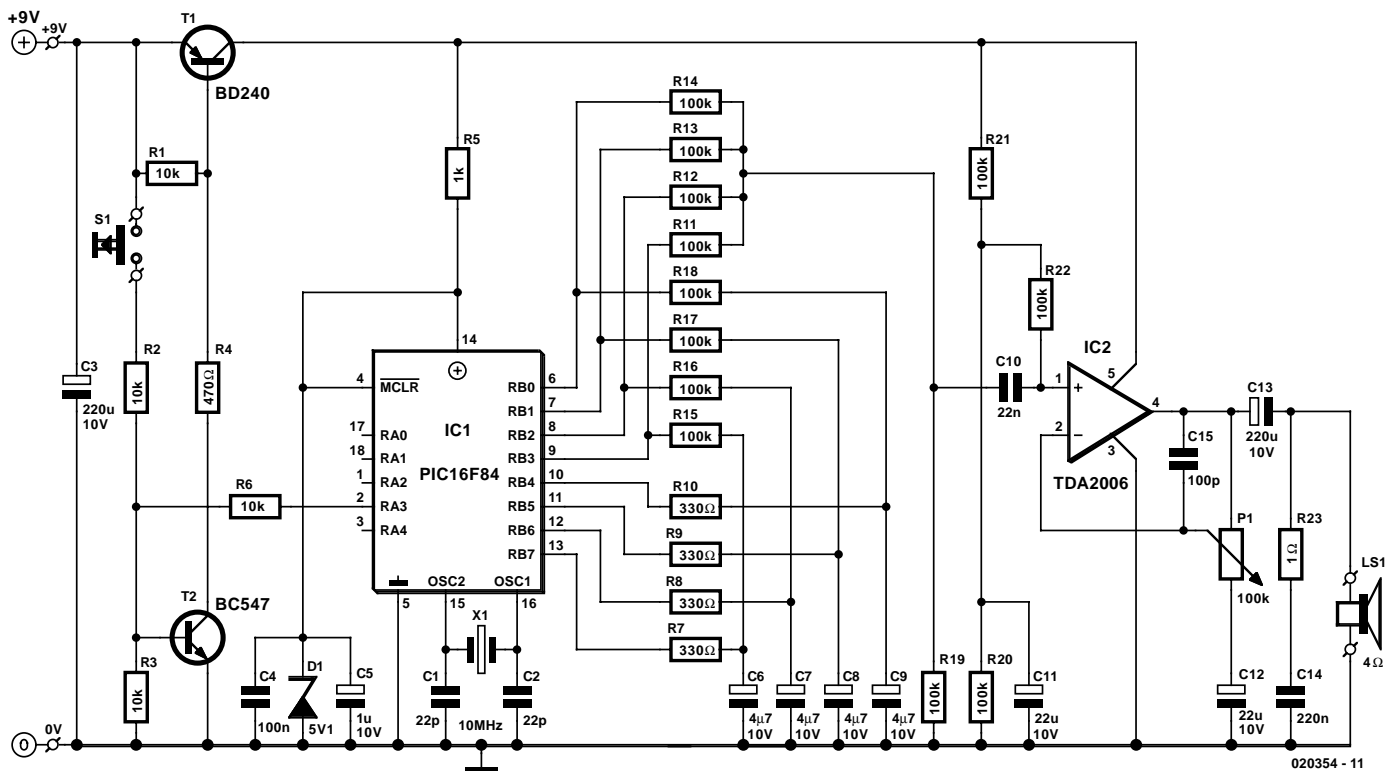


Figure 6. L'électronique se résume à une paire de circuits intégrés enrobés de composants passifs.

la constante de temps avoisine les 0,3 secondes.

La forme d'onde disponible sur une sortie RB0 à RB3 aura donc la forme représentée en **figure 5**.

Cette forme d'onde, attaque brutale et évanouissement exponentiel donne aux notes générées un timbre se rapprochant du piano (pardon aux puristes !).

Les 4 sorties « note » RB0 à RB3 sont mixées par les résistances R15 à R18. Le son global est disponible aux bornes de la résistance R19.

Ce son global est envoyé à l'amplificateur audio par l'intermédiaire du condensateur C10. Le gain de cet amplificateur est réglé par le biais du potentiomètre P1. Celui-ci devra être mis dans la position donnant la puissance de sortie maximale sans qu'il n'y ait de distorsion.

il est possible, pour augmenter la puissance sonore, d'alimenter le montage jusqu'à une tension de 13,5 volts (3 piles de 4,5 volts prises en série).

Si l'on veut pouvoir interrompre la mélodie avant que celle-ci ne soit arrivée « au bout du rouleau » (cela nous rappelle les tout premiers phonographes) lorsque l'on est sur le point d'ouvrir la porte il suffit de connecter un bouton-poussoir aux bornes de R4.

Le schéma

Un coup d'œil au schéma de l'électronique représentée en **figure 6** nous apprend que nous y retrouvons les sous-ensembles du synoptique. Examinons-le d'un peu plus près.

Le schéma semble encore plus simple que le synoptique. La puissance des microcontrôleurs actuels est impressionnante.

Au centre du montage on retrouve le PIC16F84 doté de toute une bardée de résistances sur ses lignes de port. La partie droite du schéma concerne l'amplificateur audio basé sur le TDA2006.

La réalisation

Grâce au dessin de platine représenté en **figure 7**, la réalisation est vraiment un jeu d'enfant. On commencera par l'implantation des composants de petite taille, pour terminer par les composants à fort « déve-

Liste des composants

Résistances :

R1 à R3, R6 = 10 k Ω
 R4 = 470 Ω
 R5 = 1 k Ω
 R7 à R10 = 330 Ω
 R11 à R22 = 100 k Ω
 R23 = 1 Ω
 P1 = potentiomètre 100 k Ω log

Condensateurs :

C1, C2 = 22 pF
 C3 = 220 μ F/10 V radial
 C4 = 100 nF
 C5 = 1 μ F/10 V radial
 C6 à C9 = 4 μ F/10 V radial
 C10 = 22 nF
 C11, C12 = 22 μ F/10 V radial
 C13 = 220 μ F/10 V radial
 C14 = 220 nF
 C15 = 100 pF

Semi-conducteurs :

D1 = diode zener 5V1/500 mW
 IC1 = PIC 16F84A-10/P
 (programme **EPS 020354-1**)
 IC2 = TDA2006 (Philips)
 T1 = BD240
 T2 = BC547

Divers :

LS1 = haut-parleur 4 Ω
 S1 = bouton-poussoir unipolaire à contact travail
 X1 = quartz 10 MHz
 connecteur à pression pour pile 9 V
 pile 9 V

loppement vertical » comme T1, IC2 et P1. La position de IC1 sera prise par un support si vous envisagez de modifier de temps à autre le « répertoire » de votre carillon. Sinon, si vous décidez de ne plus jamais y toucher, autant le souder directement sur la platine.

Après avoir vu l'électronique sous ses différents aspects, théorique et pratique, il est intéressant de passer à l'aspect logiciel vu que l'on a la possibilité de programmer ses propres mélodies dans le PIC.

Le programme

Le programme principal « **carillon.asm** » (disponible avec le reste des codes au téléchargement sur notre site, adresse www.elektor.fr, sous la dénomination

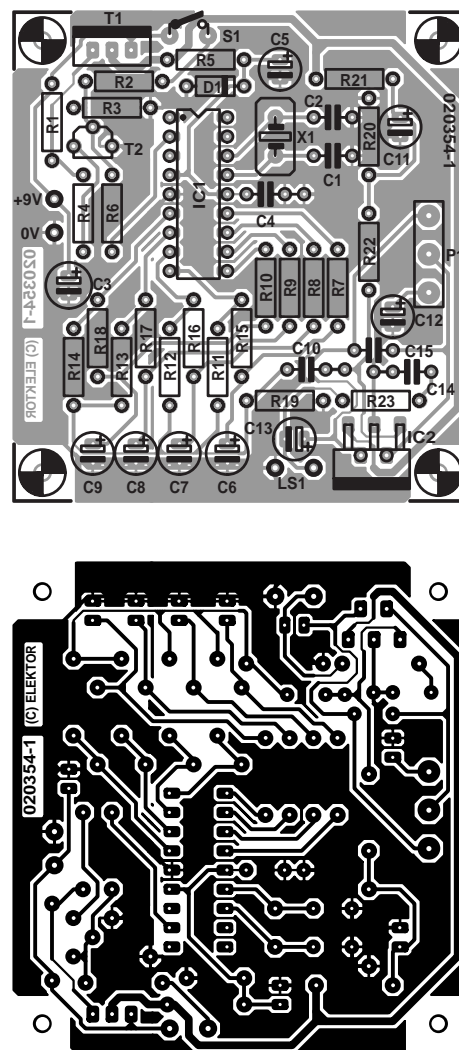


Figure 7. Dessin des pistes et sérigraphie de la platine du carillon de porte.

EPS020354-11 ou sous la forme d'une disquette disponible auprès des adresses habituelles, ses commentaires sont en anglais) a été développé sous MPLAB de Microchip. Il utilise trois routines : l'initialisation, la procédure « play » et la routine d'interruption qui fait appel à la procédure « Morceau ».

Le fichier « **def84.asm** » contient les déclarations des SFR (*Special File Register*) et « **macro84.asm** » contient des macros permettant d'utiliser des structures de programme « pseudo-Pascalienne » (IF...REPEAT...BEGIN...END...).

Enfin « **morceau.asm** » contient les morceaux de musique (partitions) et la définition des valeurs de décompteurs pour les générateurs de notes.

Nous vous proposons, en **figure 8** un synoptique du fonctionnement du programme.

Lors d'une action sur le bouton-poussoir S1, le microcontrôleur est mis sous tension.

Dans la procédure d'initialisation, après avoir

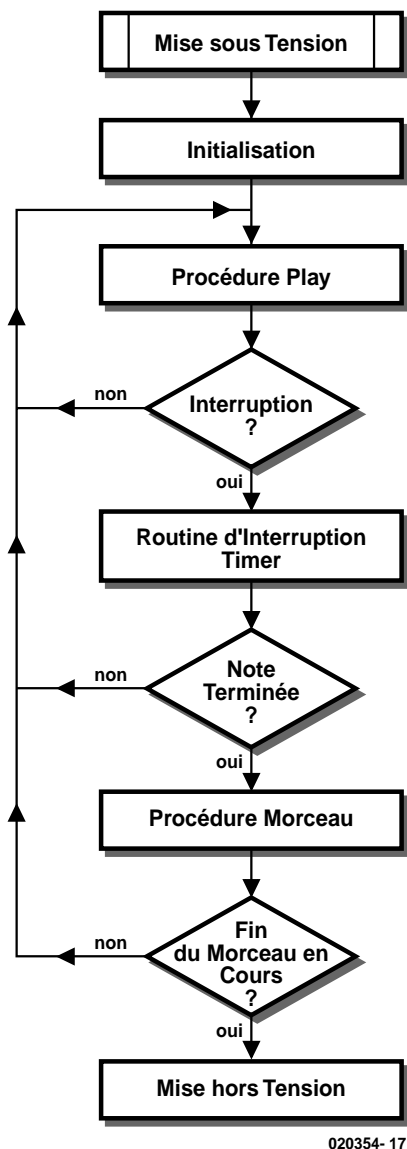


Figure 8. Ordinarogramme du programme.

configuré les différents registres de direction des ports et les registres concernant les interruptions (*timer*), la sortie RA3 est mise à « 1 » afin d'alimenter le transistor T2 pour maintenir l'alimentation de la carte.

Toujours dans cette procédure, le programme va lire le numéro du dernier morceau de musique interprété (NUM0) à l'adresse EEPROM « 0x00 » et son complément à l'adresse « 0x01 ». Si NUM0 et son complément ne concordent pas, c'est qu'il y a eu une erreur lors de la dernière sauvegarde de ces deux valeurs. NUM0 est alors remis à « 0x00 », son complément à « 0xFF » et ces valeurs sont à nouveau sauvegardées en EEPROM.

Si NUM0 et son complément sont corrects (par exemple « 0x02 » et « 0xFD »), NUM0 est incrémenté sans dépasser le dernier numéro

possible (au quel cas il est remis à zéro) et il est sauvegardé avec son complément pour la prochaine mise sous tension.

Pour finir la procédure initialisation autorise encore les interruptions timer.

Le programme principal appelle alors la routine « play ». Cette routine est la routine principale qui génère les 4 notes de musique. Cette routine devant obligatoirement « tourner » à la vitesse maximum, les instructions ont été optimisées à cet effet.

Au début de cette routine, on autorise globalement les interruptions afin de permettre à l'interruption timer de s'exécuter et on prépositionne le masque à 0. 4 décompteurs sont décrémentés à chaque exécution de la routine. Lorsqu'un décompteur arrive à zéro, le masque est mis à jour pour positionner RB0 à RB3 en sortie à « 0 » ou en entrée (haute impédance) et ce décompteur est rechargé par la valeur correspondant à la note à générer. À la fin de la routine, le masque est appliqué au registre de direction du port B par un ou exclusif qui a pour effet d'inverser les sorties concernées (signal carré). La routine d'interruption intervient à chaque fois que le timer passe de 0xFF à 0x0. Dans cette routine, le timer est rechargé avec la valeur « vitesse », le décompteur « durée » est décrémenté et lorsqu'il arrive à zéro la routine « morceau » est appelée. D'autre part les sorties RB3 à RB7 correspondant aux générateurs d'enveloppes sont positionnées en entrée (haute impédance).

La routine « morceau » commence par prépositionner le masque1 qui servira à piloter les sorties RB3 à RB7 pour charger les condensateurs des générateurs d'enveloppe.

Un appel est alors effectué au tableau « Morceau0 » qui est le début de la partition musicale. Les pointeurs « ptrmorceau' » et « ptrmorceau » qui ont été calculés dans l'initialisation permettent de sauter au début du morceau désiré et de récupérer les 4 notes à interpréter, la durée et l'accident. Cet accident peut prendre la valeur « tempo » pour régler la « vitesse » (valeur à recharger dans le timer) d'interprétation ou la valeur 'fin' pour dire que le mor-

Liens Internet

MPLAB IDE 5.70de Microchip
www.microchip.com/1010/pline/tools/picmicro/devenv/mplab/mplab5x/index.htm

ceau en cours d'interprétation est terminé. La sortie RA3 est dans ce cas remise à zéro, ce qui a pour effet de couper l'alimentation.

En guise de conclusion

On pourrait imaginer de nombreuses applications pour ce montage outre celui d'un carillon de porte, boîte jouet pour enfant, boîte à bijoux musicale, musique d'attente au téléphone, etc.

On peut fort bien envisager de changer les morceaux programmés dans le PIC, mais cela implique la reprise de tout le processus d'assemblage et de compilation nécessaire à la génération du code hexadécimal à mettre dans le processeur. Un défi aux amateurs de ce processeur ô combien populaire, l'environnement nécessaire, MPLAB (c'est la version 5.70 que nous avons utilisée ici, la version 6.13 requérant un format de projet différent) est disponible gratuitement au téléchargement. Le processus est simple, charger le projet (assurez-vous que tous les fichiers se trouvent bien dans le répertoire prévu, c:/software/ en l'occurrence) effectuer un assemblage + compilation par l'option Build All pour disposer du fichier .hex à mettre dans le PIC. Par le biais d'un programmeur de PIC adéquat.

(020354)

Contenu de la disquette 020354-I I

Fichiers assembleur : carillon.asm, def84.asm, morceau.asm, macro84.asm
 Fichier hexadécimal : carillon.hex
 Fichier de projet : carillon.pjt