



Paolo Rognoni (Paolino)

LO HAI MAI REALIZZATO CON UN PIC? UNA SORPRESA MUSICALE PER BABBO NATALE!

20 December 2009

Complici i miei figli, per quest'anno ho intenzione di fare una bella sorpresa a **Babbo Natale**. Tutti gli anni, a Natale, ci porta i doni e quest'anno abbiamo deciso di **fargli una sorpresa**. Però, per piacere: sssstt! Non ditelo in giro, altrimenti rischiamo di rovinare tutto.

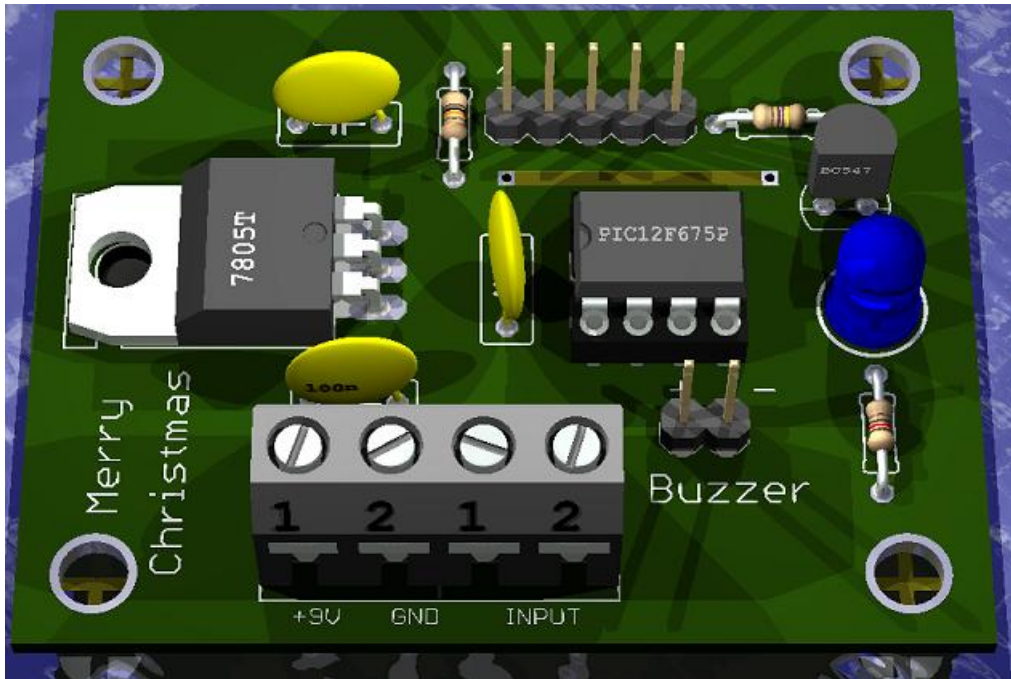
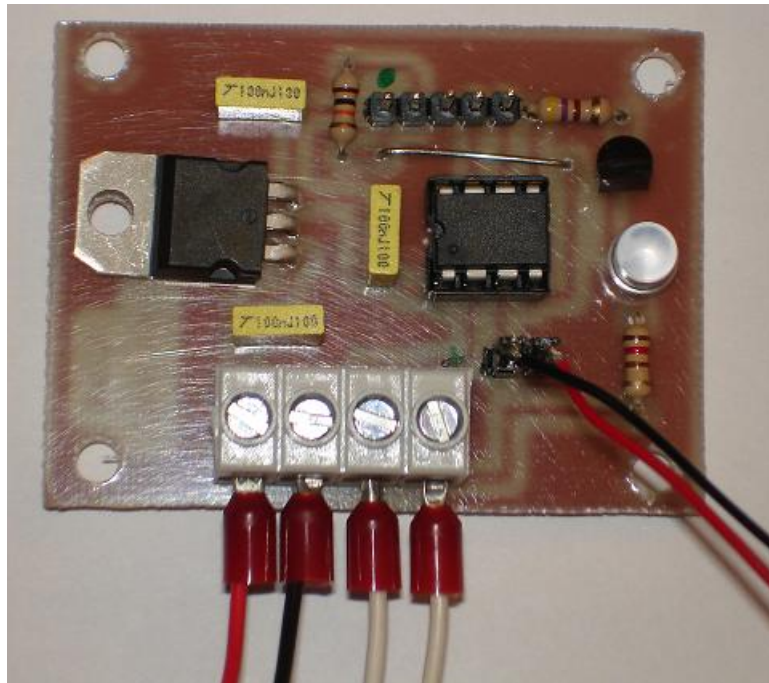
L'idea.

Abbiamo pensato di allestire un **piccolo circuito** e di lasciare disponibile sul tavolo, accanto al latte e ai biscotti, anche un pulsantino con un cartello: **PREMI QUI**. In questo modo vorremmo che Babbo Natale possa ascoltare alcuni **brani tipici natalizi** "suonati" (se così si può dire) da un circuito embedded a microcontrollore. **Con un PIC**, ovviamente! E allora ecco qui il circuito (per scaricare il file PDF proseguire alla sezione "Progetto"). Si osservi che anche per questo progetto si è pensato di impiegare un **PIC** della **serie 12F6xx** alimentato da uno stadio stabilizzatore a 5V. Al PIC sono connessi un pulsante normalmente aperto, un buzzer piezo elettrico ed un LED pilotato da un transistor.

viene pilotato da un piccolo transistor per evitare di portare il microcontrollore alla massima corrente disponibile (25mA); si può infatti pensare di utilizzare i LED ad alta luminosità che hanno una caduta di tensione piuttosto significativa e corrente superiori a quella massima erogabile dal PIC. Per questa sezione del circuito si raccomanda di valutare in modo opportuno i valori dei resistori R2 ed R3; sostituendo il transistor con un altro modello può darsi che R2 debba subire una variazione. Il valore di R3 è invece da calcolare sulla base del LED scelto. La lista dei componenti è la seguente:

- C1 100n
- C2 100n
- C3 100n
- IC1 PIC12F675P oppure PIC12F683
- IC2 7805
- J1 Pin strip a 5 poli, passo 2.54mm (programmazione ICSP del PIC)
- J2 Buzzer di tipo piezoelettrico privo di oscillatore
- LED1 BLU
- R1 10k
- R2 470
- R3 120
- T1 BC547
- X1 Connettore passo 5.08 mm
- X2 Connettore passo 5.08 mm

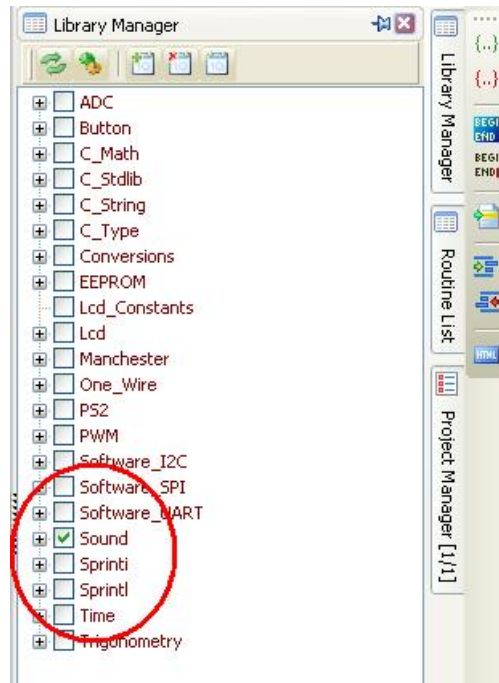
Si osservi che il progetto può funzionare tanto con PIC12F675 quanto con PIC12F683; i due microcontrollori presentano la medesima piedinatura e PIC12F683, oltre a disporre di funzionalità in più rispetto a PIC12F675, ha anche una flash/rom di programma (program memory) di dimensioni maggiori. Questo aspetto permetterà di poter far suonare brani più lunghi o un numero maggiore di brani. Per quanto concerne la realizzazione pratica, voglio illustrare il rendering al computer (ottenuto mediante Povray a partire dal circuito stampato disegnato con EAGLE) ed il prototipo che verrà utilizzato per allietare Babbo Natale:

*MerryChristmas.JPG**Prototipo_MerryChristmas.JPG*

Il progetto è veramente semplice pertanto non è richiesta la realizzazione del circuito stampato, una basetta millefori è senz'altro adatta allo scopo.

Il firmware

La stesura del codice è stata fatta con MikroC PRO di Mikroelektronica. Questo compilatore dispone di una vasta gamma di funzioni di libreria, già pronte; tra queste si cita la Sound Library che è attivabile dallo strumento Library Manager ponendo una spunta accanto al suo nome.



LibraryManager_SoundLibrary.JPG

Questa libreria mette a disposizione due semplici funzioni, una per la inizializzazione della porta che verrà utilizzata per la generazione dei suoni ed un'altra per la riproduzione acustica vera e propria; queste funzioni sono rispettivamente Sound_Init() e Sound_Play(). La funzione Sound_Play è in grado di generare un'onda quadra alla frequenza desiderata per la durata desiderata, con alcune limitazioni ben specificate. La libreria che si va ad impiegare sfrutta l'oscillatore di sistema per la generazione dell'onda quadra; per questa ragione, le frequenze minima e massima riproducibili sono vincolate da queste espressioni:

$$F_{min} = F_{osc} / (80 * 3) \quad (1)$$

$$F_{max} = F_{osc} / (80 * 255) \quad (2)$$

Questo aspetto è da non trascurare in quanto non è possibile riprodurre note musicali che vadano a coprire ottave molto distanti tra di loro. La frequenza delle note musicali è illustrata qui nel seguito (fonte <http://www.febat.com>): Il numero accanto

alla nota indica il numero dell'ottava alla quale la nota fa riferimento. Ora consideriamo il brano natalizio **Jingle Bells**, il cui spartito è riportato nel seguito:

Considerando di impiegare l'oscillatore interno al PIC con una frequenza pari a 4MHz, mediante la (1) e la (2) si stabiliscono le frequenze minima e massima delle note che possiamo far suonare:

$$F_{min} = 4000000 / 240 = 16666Hz$$

$$F_{max} = 4000000 / (20400) = 196Hz$$

Lo spartito mostra come il brano sia a cavallo di due ottave, pertanto si possono scegliere l'ottava 6 e la 7.

Jingle Bells



Spartito_Jingle_Bells.jpg

Tradotto in linguaggio C, lo spartito diventa:

```
unsigned int const JingleBells [102]={  
    SI6,QUARTO,  
    SI6,QUARTO,  
    SI6,META,  
  
    SI6,QUARTO,  
    SI6,QUARTO,  
    SI6,META,  
  
    SI6,QUARTO,  
    RE7,QUARTO,  
    SOL6,QUARTO,  
    LA6,QUARTO,  
  
    SI6,INTERO,  
  
    D07,QUARTO,  
    D07,QUARTO,  
    D07,QUARTO,  
    D07,QUARTO,  
  
    D07,QUARTO,  
    SI6,QUARTO,  
    SI6,QUARTO,  
    SI6,OTTAVO,  
    SI6,OTTAVO,  
  
    SI6,QUARTO,  
    LA6,QUARTO,  
    LA6,QUARTO,  
    SI6,QUARTO,  
  
    LA6,META,  
    RE7,META,  
  
    SI6,QUARTO,  
    SI6,QUARTO,  
    SI6,META,  
  
    SI6,QUARTO,  
    SI6,QUARTO,  
    SI6,META,
```

SI6,QUARTO,
RE7,QUARTO,
SOL6,QUARTO,
LA6,QUARTO,

SI6,INTERO,

D07,QUARTO,
D07,QUARTO,
D07,QUARTO,
D07,QUARTO,

D07,QUARTO,
SI6,QUARTO,
SI6,QUARTO,
SI6,OTTAVO,
SI6,OTTAVO,

RE7,QUARTO,
RE7,QUARTO,
D07,QUARTO,
LA6,QUARTO,

SOL6,INTERO
};

Le frequenze delle note ed i valori delle durate (INTERO, META, QUARTO, OTTAVO) sono stati così definiti:

```
// Frequenze note musicali [Hz]
#define D06  1046
#define RE6  1175
#define MI6   1318
#define FA6  1397
#define SOL6 1568
#define LA6  1760
#define SI6  1976
#define D07  2093
#define RE7  2349
#define MI7  2637
#define FA7  2794
#define SOL7 3136
#define LA7  3520
```

```
#define SI7 3951
#define D08 4186
#define PAUSA 0

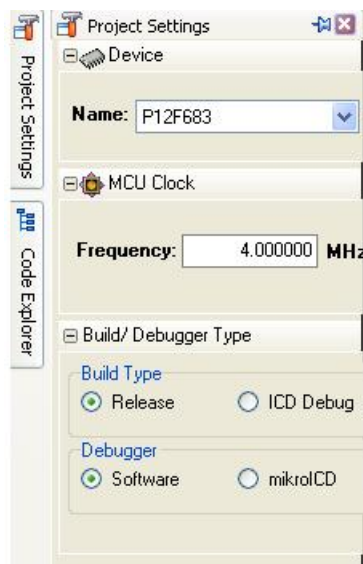
// Durata delle note [ms]
#define INTERO 800 // 1
#define META 400 // 1/2
#define QUARTO 200 // 1/4
#define OTTAVO 100 // 1/8
```

Per quanto concerne la durata, va detto che se i valori possono essere aumentati o ridotti (cum grano salis) ricordando che vanno rispettati i rapporti tra le durate! Se INTERO venisse portato a 1000 ms, OTTAVO non potrà che assumere valore 125 ms (rapporto 1:8).

Come detto in precedenza, il PIC12F675 non ha una memoria flash/rom molto ampia; per questa ragione è possibile sostituirlo con un PIC12F683 che ha un pin-out compatibile ma anche alcune funzionalità in più. Ecco perché prima di eseguire la compilazione è necessario attivare il flag relativo al PIC che si desidera utilizzare:

```
//#define PER_PIC12F675
#define PER_PIC12F683
```

Inoltre, sulla base del PIC che si vuole impiegare, va effettuata la scelta nell'ambiente di sviluppo di MikroC PRO; nell'immagine seguente è mostrata la scelta del PIC12F683:



ProjectSettings.JPG

Ma ciò non è ancora sufficiente: in funzione del PIC usato, vanno selezionati i Configuration bits secondo quanto illustrato nelle seguenti figure:

Configuration Bits	
Oscillator	Internal RC Clockout
Watchdog Timer	Off
Power Up Timer	Off
Master Clear Enable	Enabled
Brown Out Detect	On
Code Protect	Off
Data EE Read Protect	Off

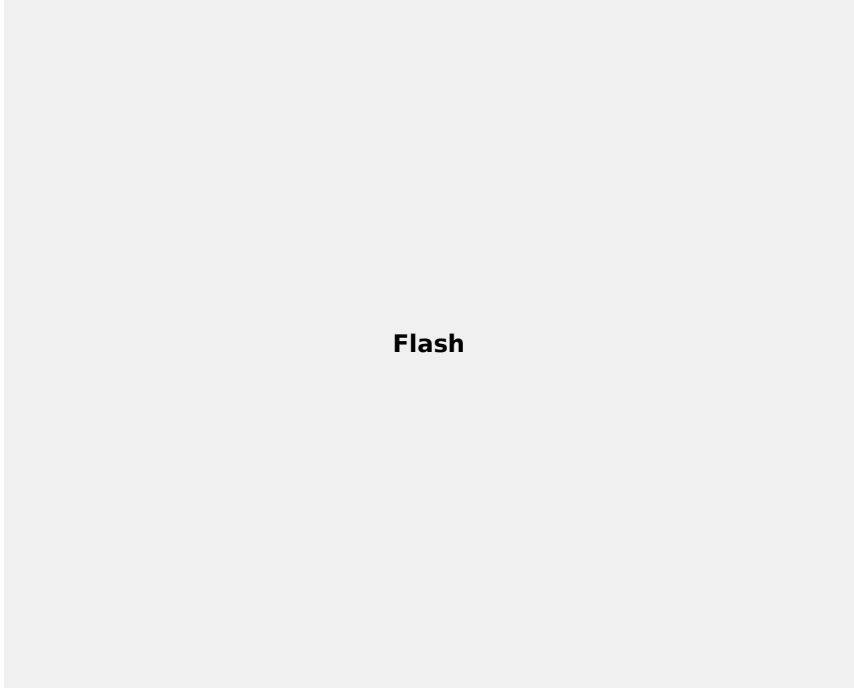
ConfBits_PIC12F675.JPG

Configuration Bits	
Oscillator	Internal RC No Clock
Watchdog Timer	Off
Power Up Timer	Off
Master Clear Enable	Enabled
Code Protect	Off
Data EE Read Protect	Off
Brown Out Detect	BOD Enabled, SBOREN Disabled
Internal External Switch Over Mode	Enabled
Monitor Clock Fail-safe	Enabled

ConfBits_PIC12F683.JPG

Come funziona

Una volta alimentato, il circuito esegue un breve test acustico e ottico, emettendo due note ad intermittenza ed un breve lampeggio del LED. Questo ci permette di capire che il circuito è acceso ed è in attesa di essere avviato. Il PIC avvia la riproduzione delle melodie, con la pressione del pulsante; al termine, il PIC si riporta nuovamente in stato di attesa che il pulsante venga azionato nuovamente (magari Babbo Natale vuole ascoltare il/i brano/i più di una volta). Ecco cosa dovrebbe accadere quando Babbo Natale premerà il pulsante:



Flash

Sperimentare ancora

Come sempre, al lettore è lasciata ampia libertà di ampliare il progetto; un interessante modifica potrebbe essere rappresentata dall'attivazione del brano musicale impiegando una fotoresistenza che, illuminata, dà il via alla esecuzione del brano. Così modificato, il progetto potrebbe essere utilizzato come biglietto elettronico di auguri di compleanno nascosto dentro ad un pacchetto. Il video proposto mostra come il PIC suoni due brani: **Jingle Bells** ed **Adeste Fideles**. Per quest'ultimo allego uno stralcio di spartito utilizzato come traccia per la riproduzione audio:

Adeste Fideles / O Come All Ye Faithful

John Francis Wade (1711-1786)
Arranged William Wallace
Copyright © 2004



Spartito_Adeste_Fideles.jpg

Progetto

Il progetto, composto dallo schema elettrico e dal firmware scritto in C con MikroC PRO, è scaricabile a lseguente indirizzo:

<http://www.electroportal.net/users/files/MerryChristmas.zip>

Riferimenti

Datasheet PIC12F675: <http://ww1.microchip.com/downloads/en/DeviceDoc/41190F.pdf>

Data sheet PIC12F683: http://ww1.microchip.com/downloads/en/DeviceDoc/41211D_.pdf

MikroC PRO: <http://www.mikroe.com/en/compilers/mikroc/pro/pic/>

Frequenza note musicali: http://www.febat.com/Musica/Musica_frequenze_musicali.html

CAD EAGLE: <http://www.cadsoftusa.com>

Pillole di microcontrollori PIC: <http://www.inwardizioni.it/pic2/>

Estratto da "<http://www.electroyou.it/mediawiki/index.php?title=UsersPages:Paolino:merrychristmas>"