

A09

151

Stefano Salvatori
Introduzione
alla progettazione
con gli LPC2000



Copyright © MMXI
ARACNE editrice S.r.l.

www.aracneeditrice.it
info@aracneeditrice.it

via Raffaele Garofalo, 133/A-B
00173 Roma
(06) 93781065

ISBN 978-88-548-4342-4

*I diritti di traduzione, di memorizzazione elettronica,
di riproduzione e di adattamento anche parziale,
con qualsiasi mezzo, sono riservati per tutti i Paesi.*

*Non sono assolutamente consentite le fotocopie
senza il permesso scritto dell'Editore.*

I edizione: ottobre 2011

*A mia madre,
che amava scrivere
e a mio padre,
un artista*

Se ascolto dimentico,
se vedo ricordo,
se faccio capisco.

Confucio

Indice

- v *Presentazione*
- 1 *Premessa*
- 9 **Capitolo I**
LPC2000 e ARM7TDMI in breve
- 1.1. Introduzione, 9 – 1.2. Il processore ARM7TDMI, 10 – 1.3. Caratteristiche degli LPC2000, 26 – 1.4. Note su alcuni μ C della famiglia LPC2000, 39 – 1.5. Osservazioni conclusive, 42 – Riferimenti, 44
- 45 **Capitolo II**
Ambiente per lo sviluppo dei progetti
- 2.1. Introduzione, 45 – 2.2. Sezione hardware del sistema di sviluppo, 45 – 2.3. Preparazione della sezione software dell'ambiente di sviluppo, 57 – 2.4. Stesura e verifica sperimentale di un progetto, 80 – 2.5. Esercizi, 102 – Riferimenti, 104
- 105 **Capitolo III**
Interfacciamento digitale
- 3.1. Introduzione, 105 – 3.2. Primo esempio di input e output, 106 – 3.3. Note sulle tecniche di manipolazione dei bit di un registro, 115 – 3.4. Ingressi e uscite multiple, 117 – 3.5. Gestione di una tastiera non codificata, 125 – 3.6. Interfacciamento con display alfanumerici a cristalli liquidi, 130 – 3.7. Connessione di tipo wired-or, 157 – 3.8. Fast-GPIO, 169 – 3.9. Esercizi, 173 – Riferimenti, 174

175 **Capitolo IV**
Gestione delle interruzioni

4.1. Introduzione, 175 – 4.2. Note sulla gestione delle interruzioni nell'ARM, 177 – 4.3. Impostazioni per la gestione dei segnali esterni di interrupt, 180 – 4.4. Software per la gestione delle IRQ vettorizzate, 185 – 4.5. Gestione di eventi, 195 – 4.6. Priorità per le interruzioni vettorizzate, 199 – 4.7. Uso delle interruzioni non vettorizzate, 204 – 4.8. Gestione delle FIQ, 206 – 4.9. Osservazioni conclusive, 209 – 4.10. Esercizi, 210 – Riferimenti, 214

215 **Capitolo V**
Interfacciamento analogico

5.1. Introduzione, 215 – 5.2. Programma di test per l'ADC, 216 – 5.3. Misura della temperatura ambiente, 224 – 5.4. Un circuito per le misure di intensità luminosa, 227 – 5.5. Interfaccia analogica per tastiera a matrice, 238 – 5.6. Pilotaggio del DAC MCP4921, 243 – 5.7. Un esempio di sistema di controllo, 250 – 5.8. Esercizi, 256 – Riferimenti, 262

263 **Capitolo VI**
Uso dei Timer

6.1. Introduzione, 263 – 6.2. Timer d'uso generale, 264 – 6.3. Uso dei timer per la misura di intervalli di tempo, 266 – 6.4. Interruzione per evento di *match*, 284 – 6.5. Generazione di segnali con i timer, 290 – 6.6. Ingressi esterni di clock, 299 – 6.7. Watchdog timer, 308 – 6.8. Real Time Clock, 315 – 6.9. Esercizi, 322 – Riferimenti, 328

329 **Capitolo VII**
Progetto di macchine con numero finito di stati

7.1. Introduzione, 329 – 7.2. Rappresentazione di una macchina a stati, 330 – 7.3. Descrizione di una FSM in linguaggio C, 339 – 7.4. Progetto di un orologio con funzione di *set*, 351 – 7.5. Modifica del progetto “*thermostat*”, 362 – 7.6. Esercizi, 371 – Riferimenti, 373

375 Capitolo VIII

Alcune tecniche di comunicazione seriale

8.1. Introduzione, 375 – 8.2. SPI, *Serial Peripheral Interface*, 378 – 8.3. Bus I²C, 396 – 8.4. Osservazioni e confronto tra bus SPI e I²C, 402 – 8.5. Interfacce I²C integrate negli LPC2000, 404 – 8.6. Funzioni per la gestione delle interfacce I²C, 410 – 8.7. Verifica con I/O Expander PCF8574, 424 – 8.8. Verifica con EEPROM 24AA515, 427 – 8.9. Standard RS-232, 432 – 8.10. UART standard '550' integrata negli LPC2000, 437 – 8.11. Verifica della comunicazione attraverso l'interfaccia RS-232, 441 – 8.12. Esercizi, 453 – Riferimenti, 456

Presentazione

I microcontrollori (μC), al pari dei microprocessori, rappresentano oggi la tipologia di dispositivi elettronici più diffusa in ambito sia industriale che dell'elettronica di consumo. L'importanza del ruolo rivestito da tali componenti nelle odierne applicazioni è in gran parte dovuta all'incremento del numero di applicazioni di controllo e l'espandersi, in modo sempre più capillare, dei cosiddetti sistemi *embedded*.

I μC , in sintesi, estendono il concetto di microprocessore (a cui si associa un'unità centrale di elaborazione) integrando su un medesimo chip una vasta gamma di dispositivi dedicati (ADC, DAC, Timer, unità di comunicazione, ecc.) che consentono un agevole ed efficiente interfacciamento verso l'esterno in cui va attuato un controllo o una acquisizione. In quest'ottica, la peculiarità maggiore dei μC non è solo quella di essere dispositivi d'uso generale di tipo programmabile, in grado cioè di eseguire un codice specifico, ma anche quella di interagire con l'ambiente in cui si trovano grazie alla disponibilità intrinseca di un gran numero di periferiche

Questo testo nasce soprattutto dall'esperienza didattica maturata in diversi anni presso la Facoltà di Ingegneria dell'Università degli Studi di Roma Tre, e dall'attività professionale di progettazione di sistemi elettronici di controllo e acquisizione a μC . Con esso si intende dare un'introduzione approfondita della famiglia di μC LPC2000 (NXP), molto diffusi in diversi ambiti dell'elettronica, anche *consumer*, per il loro impiego nei sistemi di controllo digitali e basati su uno dei più diffusi microprocessori a 32 bit: l'ARM venduto, a oggi, in decine di unità al secondo.

Rispettando un approccio pratico senza tuttavia tralasciare l'impor-

tanza degli aspetti teorici coinvolti, i diversi capitoli trattano le tematiche fondamentali utili allo sviluppo dei progetti descritti nel testo. Con essi il lettore acquisisce una buona autonomia e familiarità nell'impiegare gli strumenti di lavoro coinvolti nello studio. I capitoli si fondano sullo sviluppo di attività pratiche con cui guidare la realizzazione e la verifica di sistemi, via via più complessi, basati su μC .

L'idea fondamentale è quella di istruire il lettore con esercitazioni sperimentali che possono essere agevolmente svolte in un qualunque ambiente (come l'aula utilizzata per tenere le lezioni frontali e/o esercitazioni di tipo teorico). Questo approccio consente di superare efficacemente le difficoltà legate non solo all'allestimento di un vero e proprio laboratorio (didattico) ma, soprattutto, al suo mantenimento. La postazione, costituita da un hardware dal costo realmente contenuto e un computer (anche portatile), permette comunque di lavorare in un ambiente non troppo dissimile da quello reale, consentendo di sviluppare il progetto fino alle sue fasi conclusive di diagnostica sul campo (*debug in circuit*). Il software impiegato, "Eclipse", di tipo Open Source, è ampiamente diffuso e utilizzato anche in diversi ambiti professionali di programmazione. Per l'esiguo assorbimento di corrente, l'alimentazione di tutto l'hardware viene fornita da una presa USB del PC. Non da ultimo, allora, il lavoro sperimentale avviene in regime di assoluta sicurezza essendo assente qualunque connessione verso la rete elettrica di distribuzione. Si potrà verificare che viene acquisita una certa autonomia già dopo le prime esercitazioni, a partire da quella dedicata alla semplice descrizione degli strumenti di lavoro. Come ovvio, un tale traguardo rende lo svolgimento del lavoro complessivo molto agevole grazie all'entusiasmo e alla fiducia in sé che lo studente, o il lettore in particolare, può guadagnare. Inoltre, la possibilità di verificare sperimentalmente il funzionamento di un sistema permette di valutare "dal vivo" le problematiche insite nell'organizzazione e nello sviluppo dell'esperimento stesso e di apprezzare meglio le soluzioni circuitali e i contenuti teorici più nascosti. Quanto enunciato vuole rappresentare un indubbio punto di forza, cercando di superare le difficoltà tipiche che si hanno nell'insegnamento di discipline scientifiche attraverso il più "classico" metodo di trasmissione del sapere fondato soprattutto nell'apprendimento di nozioni teoriche e, al limite, alla simulazione "virtuale" del funzionamento del sistema.

Si sottolinea che la postazione di lavoro descritta fornisce allo studente un efficace strumento utile allo sviluppo di progetti personalizzati nel suo futuro di probabile professionista, mentre al lettore più esperto, o all'ingegnere già formato, lo strumento descritto e, ovviamente, il contenuto dell'opera possono permettere di acquisire le conoscenze e competenze adeguate alla tecnologia impiegata. Con tale aspetto il libro si rivolge anche a tecnici, ingegneri, educatori o professionisti del settore che intendano migrare verso una più recente tecnologia o anche semplicemente approfondire gli argomenti che il testo illustra (interruzioni, gestione di periferiche, realizzazione di macchine a stati anche complesse, bus e standard di comunicazione, per citarne alcuni) senza la necessità, forse, di seguire un corso accademico o professionalizzante. La trattazione non è limitata all'arida descrizione di uno schema o di un sistema, ma più proficuamente alla sua ideazione e alla sua verifica sperimentale di funzionamento sul campo secondo un'ottica profondamente progettuale piuttosto che di analisi. Anche se in modo specifico i diversi capitoli trattano il caso particolare di μC LPC2103 basato su ARM7TDMI (che costituisce un ottimo esempio di moderno processore RISC), non viene per nulla inficiata la generalità della trattazione. Le nozioni apprese, infatti, consentiranno di migrare agevolmente verso sistemi diversi sia come taglia che come architettura. L'opera si sviluppa attraverso la risoluzione di problematiche d'interfacciamento relativamente semplici lasciando lo spazio di sviluppare personalmente quelle più complesse legate a sistemi reali di controllo che sono però composte dai diversi elementi fondamentali appresi. Il lettore potrà far tesoro delle conoscenze e competenze acquisite nei riguardi dei moderni ed evoluti μC su cui il testo si basa per avviare un lavoro di approfondimento del tutto autonomo, personalizzato e adeguato all'attività di progettista elettronico.

L'opera può essere adottata come libro di testo in un corso di "Progettazione basata sui microcontrollori" dei Corsi di Laurea Triennale o Magistrale in Ingegneria Elettronica (6-9 crediti) o anche come riferimento per corsi di Elettronica presso le Facoltà di Ingegneria Meccanica o Elettrotecnica e Automazione in cui, solitamente, viene dato particolare risalto ai sistemi di controllo in ambito industriale. Per un efficace apprendimento del contenuto, è prevista una minima conoscenza di programmazione (preferibilmente in linguaggio C) anche se

si sottolinea che, in tal senso, pochi sono i costrutti fondamentali impiegati nello sviluppo dei progetti.

Con questo libro, si spera di poter dare un contributo nell'ambito della formazione soprattutto stimolando quella necessaria curiosità nel lettore con la quale poter approfondire una tematica realmente vasta, ma interessante per l'attività dell'ingegnere elettronico o tecnico in genere, appassionato all'argomento e sensibile all'approccio creativo che il lavoro di progettazione inevitabilmente prevede.

Contenuti

Dopo una breve premessa in cui sono indicati gli elementi salienti che possono spingere alla scelta di una particolare soluzione hardware, il primo capitolo è dedicato alla descrizione sommaria sia del processore ARM7TDMI che della famiglia di μC LPC2000, enunciando le caratteristiche salienti che li contraddistinguono. L'approfondimento dell'argomento, per arricchire la personale preparazione culturale di base, potrà essere affrontato con lo studio dei manuali tecnici e dei data-sheet indicati come riferimento e disponibili, gratuitamente, sui siti delle aziende costruttrici dei dispositivi stessi.

Il secondo capitolo è dedicato all'attenta descrizione dei passi utili alla realizzazione dell'ideale postazione di lavoro composta da una sezione hardware e una software di base completa. L'hardware può essere costituito da una delle numerosissime *development board* disponibili commercialmente o una scheda *custom* realizzata personalmente a cui affiancare altri dispositivi anch'essi facilmente reperibili sul mercato. Nella seconda parte del capitolo viene descritta l'installazione del software e la sua messa a punto. Viene altresì indicato un primo esempio di progetto con cui osservare le diverse fasi di lavoro, che vanno dalla stesura del codice al *debug* finale.

Il terzo capitolo tratta il tema (ovvio) d'interfacciamento attraverso le linee digitali di ingresso e uscita del μC . I progetti descritti vanno dal pilotaggio di un LED e l'acquisizione dello stato di un tasto, fino al pilotaggio di display a più caratteri, a LED o LCD (intelligenti) e, come esempio di acquisizione da più canali di ingresso, la cattura dello stato di un dip-switch o di una tastiera a matrice. L'ultima sezione è dedicata alla descrizione della comunicazione su bus singolo per evi-

denziare l'importanza di disporre di linee di tipo open-drain. Infine, sono anche mostrate le prestazioni in termini di velocità dovute all'accesso di tipo *fast* di porte digitali del μC .

Il quarto capitolo è interamente dedicato alla gestione delle interruzioni, argomento di fondamentale importanza nei sistemi di controllo a microprocessore. In quest'ambito viene descritto l'uso del *controller* di cui la famiglia LPC2000 dispone e che consente di gestire interruzioni, sia di tipo vettorizzato, sia veloci e a elevata priorità.

La maggioranza dei sistemi di controllo a microprocessore prevede la necessità di rilevare o fornire informazioni di tipo analogico verso l'esterno. Per la loro natura digitale, tali sistemi dovranno allora essere dotati di elementi utili alla conversione analogico-digitale e digitale-analogica. Nel capitolo quinto, anche per sottolineare l'importanza del tema, sono sviluppate diverse voci che riguardano l'acquisizione e l'attuazione di segnali di natura analogica. I progetti descritti sono relativi a sistemi per la misura di temperatura e di intensità luminosa discutendo nel dettaglio le problematiche relative all'elaborazione e al calcolo limitato alla dimensione intera, fissa a 32 bit.

Nei sistemi *embedded* è spesso richiesta la presenza di dispositivi in grado di misurare intervalli di tempo e di gestire segnali con una temporizzazione prefissata. Il capitolo sesto è allora dedicato all'uso dei timer, sia di uso generale che speciale, come il Watchdog o il Real Time Clock. Applicazioni tipiche di tali dispositivi riguardano la generazione di ritardi di precisione, la trasmissione o ricezione di segnali digitali modulati nella larghezza dell'impulso (PWM), la cattura di particolari transizioni di un segnale digitale con conseguente memorizzazione del valore di conteggio di un timer.

Nel settimo capitolo viene delineata la metodologia di progetto di sistemi che prevedano la sintesi di macchine con numero finito di stati (FSM). Le tecniche descritte si rivelano particolarmente utili nel caso di sintesi di apparati relativamente complessi. Il metodo si applica in pratica a qualunque problema e si mostrerà che lo sviluppo del progetto si limita essenzialmente solo alla definizione attenta del diagramma degli stati della FSM.

L'ottavo capitolo è dedicato all'approfondimento della comunicazione seriale su bus standard. In particolare, sono descritte semplici verifiche della funzionalità delle periferiche hardware integrate nel di-

spositivo LPC2103: UART, SPI, I²C. Valutandone i vantaggi sia in termini di velocità che di dimensione del codice, sono illustrate le modifiche di progetti già svolti e risolti, nel secondo capitolo, mediante tecnica software di *bit-banging*. È infine descritto un semplice sistema di comunicazione verso un computer esterno su cui basare eventuali altri progetti che prevedano un trasferimento dati tra apparati.

CD-ROM

Al testo è allegato un CD-ROM che contiene l'intera *toolchain* utile all'installazione del software "Eclipse", ver. 3.2 (www.eclipse.org, da cui è possibile scaricare, eventualmente, versioni più aggiornate) e tutta l'infrastruttura necessaria alla stesura, compilazione e debug del codice di ciascun progetto. Inoltre, esso contiene i file sorgente dei diversi esempi riportati nel testo, nonché figure e materiale di riferimento di ciascun capitolo. Il contenuto è suddiviso nelle seguenti cartelle:

- **software:** contiene i file relativi alla toolchain;
- **datasheet:** contiene diversi datasheet dei componenti più importanti usati nel testo;
- **user manuals:** manuali relativi ai μ C descritti nel testo;
- **progetti:** contiene i file sorgente dei diversi progetti descritti nel testo e suddivisi nei capitoli;
- **figure:** divise nei diversi capitoli, la cartella contiene le figure salienti dei diversi temi trattati (schemi, diagrammi di flusso, diagrammi di stato, ecc.).

Note tipografiche

Per il loro uso diffuso nel testo, i termini anglosassoni sono, al limite, riportati in corsivo solo nella loro prima citazione. Il corsivo è usato, invece, soprattutto per sottolineare e rafforzare il contenuto di parole e concetti in base al contesto della frase.

Un carattere diverso è usato quando si vogliono specificare azioni particolari (come durante le fasi di installazione del software) o per indicare la codifica di un'istruzione.

Per i prefissi dei multipli usati per esprimere le dimensioni di memoria o in generale grandezze binarie, si è adottata la notazione definita nel 1998 dalla IEC, *International Electrotechnical Commission*, e ormai approvata come standard mondiale (<http://physics.nist.gov/cuu/Units/binary.html>).

Ringraziamenti

Il primo ringraziamento è rivolto a mia moglie e mio figlio per la pazienza che hanno dimostrato nelle tante giornate loro rubate per portare a compimento il testo.

Il debito maggiore lo devo esprimere al collega ing. Nicola Masarone col quale ho avuto modo di collaborare in questi anni. Senza il suo supporto quest'opera non avrebbe mai potuto svilupparsi.

Un sentito ringraziamento deve essere rivolto a Laurent Gauch (Amontec, Gauch) per la gentile concessione del software di sviluppo che è stato incluso nel CD. Lo stesso ringraziamento è anche rivolto alla NXP Semiconductors, nella persona di Kenneth Dwyer, per aver autorizzato la pubblicazione delle figure nel testo nonché dei documenti tecnici presenti nel CD.

Al prof. e amico Gennaro (Rino) Conte devo la lucidità con cui è riuscito a indicare l'istituzione di nuovi corsi di elettronica presso la Facoltà di Ingegneria Elettronica dell'Università degli Studi di Roma. Tre grazie ai quali questo testo ha avuto modo di essere elaborato e sperimentato.

Al vecchio amico ing. Marco Barbetti devo molto. I commenti, le osservazioni, i consigli e le correzioni che egli ha prodotto hanno sicuramente reso il testo migliore in molte parti. Invidiabili la sua capacità di sintesi e la visione ad ampio spettro, non limitata al solo campo dell'elettronica, di cui egli è dotato.

Devo ringraziare l'ormai ing. Marco Pacilli per le critiche e i numerosi commenti, assolutamente preziosi e pertinenti. Degna di nota l'attenzione dedicata al lavoro.

L'amico ingegnere, Alessandro Ballini, malgrado i tanti impegni, è riuscito comunque a dare il suo contributo. Un vivo ringraziamento deve essergli espresso anche per il tempo che dedica in altre occasioni.

Ringrazio Roberto Tanfani per la pazienza dimostrata sia nello stu-

diare attentamente il testo ma soprattutto per il collaudo delle diverse sezioni software contenute nel testo.

Inoltre, un vivo ringraziamento deve essere espresso verso tutti gli studenti che, in questi anni, hanno seguito il corso di “Microprocessori e Microcontrollori”. Delle loro numerose osservazioni, critiche e consigli il contenuto di questo libro si è certamente arricchito.

Infine, vorrei ringraziare tutti coloro che segnaleranno errori e imprecisioni che, inevitabilmente, sono sfuggite all'autore.

Roma, giugno 2011