Sicurezza e affidabilità nel progetto di sistemi embedded

Nella prima parte di questo articolo vengono descritte le metodologie di sviluppo e di collaudo necessarie per la realizzazione di applicazioni basate sui microcontrollori affidabili e discusse le regole e gli standard - emanati a livello sia governativo sia industriale - in vigore al fine di offrire l'opportunità di minimizzare le problematiche legate al soddisfacimento dei requisiti richiesti

Abdul Aleaf
Applications manager
National Semiconductor

I parte

Il progredire della tecnologia dei semiconduttori comporta una parallela diminuzione del costo delle memorie, il che comporta un aumento del grado di sofisticazione dei dispositivi consumer. Solo alcuni anni fa i progettisti dovevano effettuare un'attenta valutazione del costo delle memorie in rapporto alle prestazioni e alle funzionalità aggiuntive che potevano essere fornite dal software residente nelle memorie stesse. Al giorno d'oggi anche i più economici microcontrollori embedded dispongono on chip di memorie di ampie dimensioni grazie alle quali è possibile sviluppare qualsiasi funzionalità atta a garantire valore aggiunto all'utilizzatore finale. Per esempio, mediante la raccolta dei dati relativi alle prestazioni dei dispositivi, ai modelli di utilizzo, alle variazioni delle condizioni di funzionamento, alle preferenze dell'utilizzatore e ad altri elementi interenti sia il contesto sia le modalità d'utilizzo, l'"intelligenza" fornita dal software contribuisce a rendere auto-adattativi questi dispositivi nei confronti sia dell'utente sia dell'applicazione.
Una miriade di dispositivi, tra cui elettrodomestici, apparati elettronici audio/video, sistemi automotive e altri prodotti consumer sono controllati da microcontrollori sempre più ricchi di funzionalità. Non bisogna comunque dimenticare che tutti questi progressi, che hanno contribuito a semplificare la vita di ogni giorno, hanno per contro aperto la strada a nuovi potenziali rischi: ad esempio un dispositivo "innocuo" come un tostapane può convertire una quantità sufficiente di elettricità in calore da dar vita a un incendio.
Il software preposto al controllo di un elettrodomestico svolge il suo compito in maniera sicuramente adeguata, a meno che non si verifichi una situazione imprevista. Ad esempio la macchinetta del caffè mantiene il contenitore caldo finché il caffè non evapora. Ciascuno di noi potrebbe aspettarsi che i microcontrollori e le funzionalità offerte dai software più sofisticati non solo semplifichino la progettazione di un prodotto, ma contribuiscano a garantire maggiori livelli si affidabilità e sicurezza al prodotto finale. Non va comunque dimenticato che se fin dall'inizio non vengono adottate metodologie idonee in fase di progettazione e di collaudo, il prodotto finale potrebbe celare insidie, con conseguenze dannose sull'utilizzatore stesso.
L'intero processo di sviluppo - dalla definizione delle specifiche iniziali alla scelta delle metodologie di progettazione e dei tool di sviluppo, alla realizzazione, ai controlli qualità fino alla manutenzione - deve essere attentamente pianificato al fine di

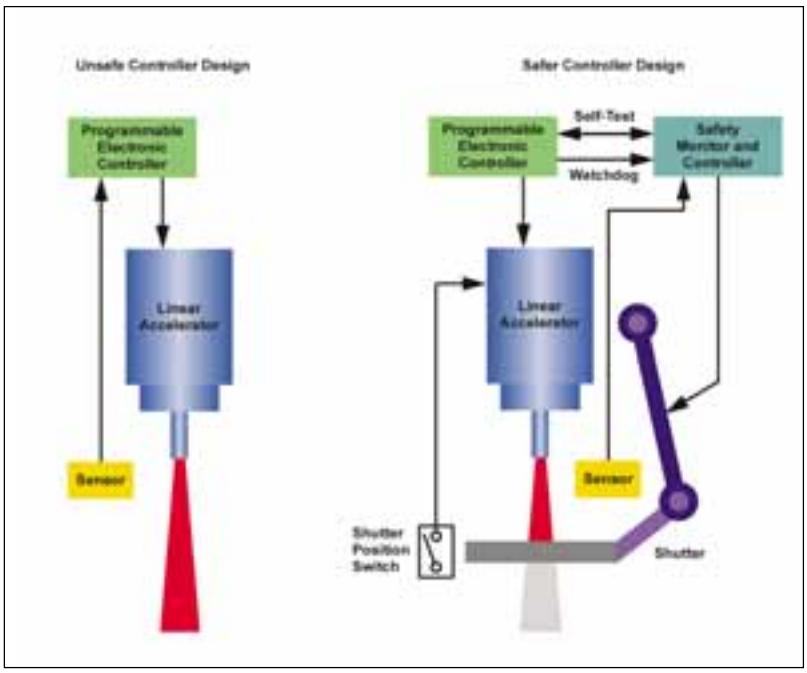


Fig. 1 - Controllore dotato di meccanismo di sicurezza indipendente (a destra) e privo di esso (a sinistra)

minimizzare il numero di guasti software e limitare quindi i danni ad essi imputabili. Nel corso di questo articolo vengono descritte le metodologie di sviluppo e di collaudo necessarie per la realizzazione di applicazioni affidabili basate sui microcontrollori: inoltre verranno discusse le regole e gli standard - emanati a livello sia governativo sia industriale - in vigore al fine di offrire l'opportunità di minimizzare le problematiche legate al soddisfacimento dei requisiti richiesti.

I pericoli dei sistemi controllati via software

Ogni nuova generazione di prodotti destinati al mercato consumer è caratterizzata da maggiori semplicità d'uso (si pensi ai telefonini dotati di memorie avanzate e migliori funzionalità in termini di composizione del numero), silenziosità (lavatrici che adottano motori in continua senza spazzole al posto di cinghie e frizioni), sicurezza (airbag e sistemi frenatura Abs a bordo degli autoveicoli) e migliore rendimento energetico (motori con controllo computerizzato dell'accensione e dell'iniezione): tutto ciò è reso possibile dalla presenza di microcontrollori embedded molto sofisticati e del relativo corredo software.

La crescente complessità di questi sistemi controllati via software sta mettendo a dura prova la capacità di progettazione, realizzazione, collaudo e manutenzione degli stessi. Ogni giorno emergono meccanismi di guasto sempre più complessi, che spaziano dai semplici inconvenienti (impossibilità di chiamare qualcuno dal proprio cellulare), ai danni economici (interruzioni di transazioni di commercio elettronico) agli infortuni più seri (guasto del sistema di frenatura di un'automobile).

Anche se gli errori software non possono essere totalmente eliminati, esistono procedure di sviluppo che permettono di minimizzare il loro verificarsi, limitare la portata dei loro effetti e aumentare la probabilità di intercettarli prima del rilascio del prodotto. Per esempio Underwriters Laboratories (Ul) ha redatto uno standard - UL 1998 - che può essere adottata come linea guida per assicurare l'affidabilità del software.

Lo sviluppo di software embedded affidabile

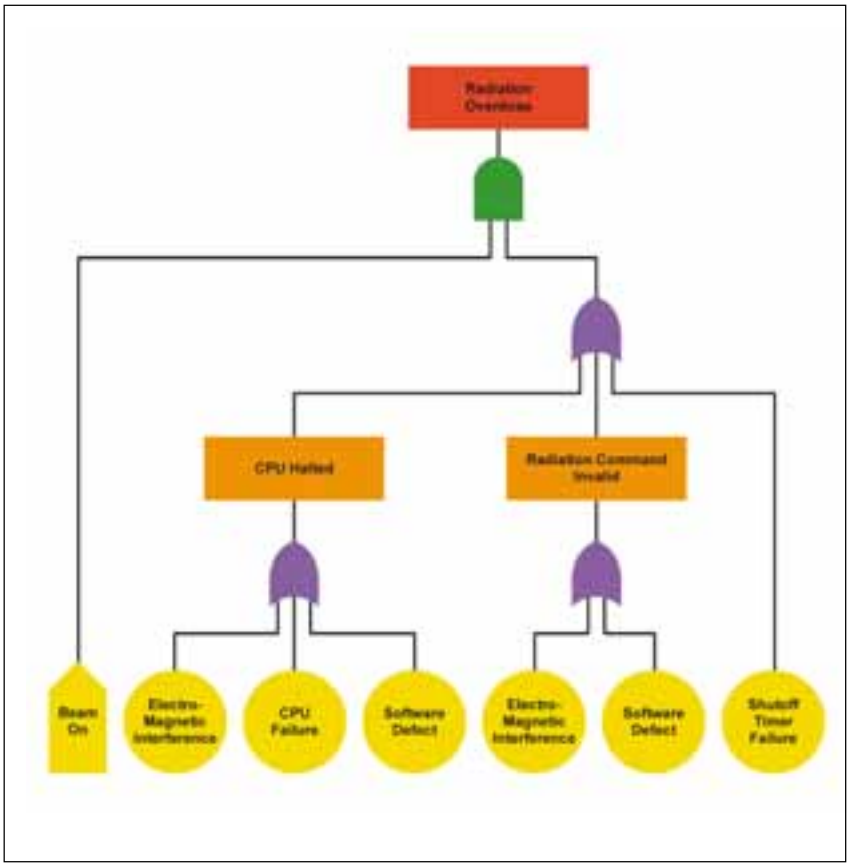
È errato credere che l'affidabilità del software si riferisca solamente alla scrittura del codice. Il concetto di affidabilità coinvolge infatti anche le procedure di sviluppo adottate per la progettazione e l'implementazione del software. Lo standard Ul 1988 fornisce un'esaustiva e dettagliata descrizione dei requisiti necessari per lo sviluppo di software affidabile e per la sua manutenzione. Esso contiene inoltre specifiche relative ai processi e ai criteri di progettazione richiesti per gestire le complesse interazioni tra le componenti hardware e quelle software che potrebbero dar luogo a meccanismi di guasto. Queste comprendono analisi di rischio, progettazione fail-safe (capacità di sostenere un guasto e terminare in sicurezza un'operazione) e faul-tolerant (capaci di gestire situazioni critiche legate alla presenza di guasti), analisi e metodologie di test, oltre all'etichettatura che permette di identificare l'interfaccia del

prodotto, la piattaforma hardware e la configurazione software. Anche se lo standard UL 1998 si riferisce principalmente agli aspetti legati alla sicurezza, un prodotto sviluppato in conformità a questo standard evidenzia caratteristiche qualitative migliori in aree non strettamente legate alla sicurezza.
Le procedure richieste dallo standard Ul 1998 sono già adottate in tutte quelle industrie che richiedono un'elevata affidabilità del software, come ad esempio quella aerospaziali, ma trascurate in altre realtà come ad esempio quella consumer. Per queste ultime, che spesso tendono a privilegiare il time-to-market a discapito della sicurezza, lo standard Ul 1998 fornisce utili indicazioni circa le procedure che devono essere adottate per far fronte ai rischi connessi alla realizzazione di sistemi sempre più complessi. Lo standard, per esempio, richiede l'analisi dei potenziali meccanismi di guasto e l'implementazione di procedure per la tracciabilità - attraverso il ciclo di sviluppo del software - delle azioni intraprese per gestire questi meccanismi di guasto. Ul 1998 richiede un ulteriore collaudo per i meccanismi di guasto e i test alle sollecitazioni nel corso del ciclo di sviluppo del software, oltre ai collaudi minimi che vengono eseguiti tradizionalmente. Per dimostrare la conformità allo standard Ul 1998 sono richiesti i programmi di collaudo formali e i risultati dei test.
Nel seguito viene riportato un elenco degli errori contemplati nello standard Ul 1998:
- Errori di progettazione come ad esempio inesattezze a livello di algoritmi o interfacce
- Errori di codifica, compresi quelli relativi alla sintassi, alla inesattezza dei segni, alla presenza di loop infiniti e così via
- Errori di temporizzazione che possono provocare un'esecuzione anticipata o ritardata del programma
- Errori indotti provocati da guasti hardware
- Errori latenti non rilevabili fino al momento in cui non si verifica un determinato insieme di condizioni
- Guasto del sistema che impedisce lo svolgimento di qualsiasi funzione.
Lo standard in questione si occupa anche dei requisiti di verifica e collaudo relativi a:
- Self test in fase di avviamento
- Built in test durante il run time (fase di esecuzione)
- Affidabilità del software/tolleranza ai guasti
- Sezioni critiche
- Memorie non volatili
- Memorie di blocco
- Procedure di scale down
- Procedure di shutdown
- Sezione di supervisione
Un obiettivo importante dello standard Ul 1998 è la tolleranza ai guasti (fault tolerance). Un sistema di questo tipo garantisce il livello di servizio più soddisfacente possibile nel momento in cui si verificano guasti a livello hardware e software, come accade ad esempio nel momento in cui un sensore non è più disponibile o vengono rilevate condizioni che richiedono l'intervento del software di ripristino. Le tecniche impiegate per garantire la tolleranza ai guasti comprendono:
- Sicurezza del codice - tecnica di controllo per mezzo della quale la protezione contro errori sistematici e/o casuali nelle informazioni in ingresso e in uscita viene garantita mediante l'uso di dati ridondanti e/o trasferimento di ridondanza
- Verifica dell'uscita - tecnica di controllo che prevede il confronto delle uscite con ingressi indipendenti
- Monitoraggio degli intervalli temporali - tecnica di controllo che prevede l'esecuzione di operazioni di trigger periodiche dei dispositivi di temporizzazione con una base tempi indipendente al fine di controllare sequenza e funzioni del programma
- Somme di controllo multiple - tecnica di controllo che contempla la generazione e il salvataggio di parole separate che

Tabella 1 - Potenziali pericoli legati all'utilizzo di un acceleratore lineare per la generazione di un fascio di elettroni

Tipo di pericolo	Livello di rischio	Tempo di tolleranza	Guasto	Probabilità	Tempo di rilevamento	Misure di controllo	Tempo di esposizione
Dose eccessiva di radiazioni	Serio	100 ms	Guasto della Cpu	Rara	50 ms	Il controllore di sicurezza verifica il segnale di watchdog	50 ms
Dose insufficiente di radiazione	Moderato	2 settimane	Impostazione errata dei dati	Abbastanza alta	10 ms	Verifica Crc sui dati ogni volta che avviene l'accesso	50 ms
Emissione di radiazioni in fase di reset	Grave	100 ms	Fascio attivato durante il reset	Abbastanza alta	n.d.	L'otturatore si chiude quando viene tolta l'alimentazione	0 ms

Fig. 2 - Esempio di un possibile albero di guasto



rappresentano i contenuti delle aree di memoria che devono essere collaudate. Nel corso del self test viene generata una somma di controllo (checksum) utilizzando il medesimo algoritmo che viene confrontata con quella memorizzata

- Ripristino del sistema - il sistema è in grado di garantire il ripristino in presenza di salti di istruzione involontari senza dar luogo alla possibilità che si verifichi alcun tipo di rischio.

Standard e certificazioni europee

Se Underwriters Laboratories si occupa di stabilire le metodologie atte a migliorare le procedure di sviluppo software, dall'altra sponda dell'Atlantico è TÜV (Technischer Überwachungs Verein - Technical Supervisory Association in inglese) a espletare un simile ruolo. TÜV è un gruppo di laboratori di certificazione che esegue collaudi indipendenti e certificazioni di sicurezza ma, a differenza di Ul, non sviluppa alcun tipo di standard, occupandosi solo della certificazione degli standard sviluppati da altre organizzazioni come ad esempio Iec (International Electrotechnical Commission) e Ul. Con l'introduzione nel 1990 degli standard Din 19250 (Basic Safety Evaluation of Measuring and Control Protective Equipment) e Din 0801 (Principles for Computers in Safety-Related Systems) Tuv ha avviato la certificazione della sicurezza funzionale di sistemi elettronici programmabili utilizzando le classi di sicurezza definite in Din 19250 e i requisiti di sicurezza definiti in DIN 0801. La sicurezza funzionale fa riferimento alle operazioni del sistema che devono essere eseguite correttamente per garantire la protezione contro i pericoli. Ad esempio un sensore di pressione potrebbe rilevare una pressione eccessiva in un serbatoio a reazione e il sistema potrebbe reagire a questa indicazione aprendo una valvola limitatrice di pressione al fine di prevenire un'esplosione. L'esplosione è in esempio di pericolo e il meccanismo atto a prevenirla è un esempio di funzione di sicurezza.

La sicurezza funzionale non è sinonimo di affidabilità. Un sistema realizzato a partire da componenti non affidabili può essere certificato in conformità a uno standard di sicurezza funzionale e nel momento in cui si verifica un guasto viene garantita la possibilità di entrare in una modalità sicura. Riprendendo l'esempio precedente, nel caso il sensore di pressione si guasti e il sistema non sia più in grado di garantire l'affidabilità al verificarsi di una condizione di pressione eccessiva, esso potrebbe arrestarsi in un modo sicuro. Per contro, un sistema affidabile potrebbe garantire un uptime del 99,9% ma subire eventi catastrofici in caso di guasto.

Lo standard Din 19250 definisce otto classi - da AK1 a AK8 - per le funzioni di sicurezza, in relazione all'estensione del guasto che potrebbe essere causato dal pericolo, alla probabilità che il pericolo si presenti, al tempo di esposizione al pericolo e alle azioni preventive che possono essere intraprese al fine di attenuare il pericolo. Gli standard Din 19250 e Din 0801 sono stati rimpiazzati da Iec 61508, che integra i

concetti chiave di questi due standard comprese le classi di sicurezza da Sil1 a Sil4. TÜV offre la certificazione Iec 61058, oltre a quella relativa agli standard Din e Ul 1998.
Sebbene concepito per i sistemi per il controllo di processo, lo standard Iec 61508 prevede numerose norme alternative per specifici segmenti e i futuri standard emanati da Iec faranno riferimento alle procedure base previste da Iec 61508. Attualmente si sta cercando di adeguare quest'ultimo con lo standard Do-178B in vigore nell'industria aerospaziale. In ultima analisi, le norme derivate dallo standard Iec 61508 potrebbero essere sviluppate per altri sistemi, come ad esempio quelli utilizzati in ambito automobilistico o, più in generale, quelli per i quali la sicurezza rappresenta un elemento critico.

Analisi dei rischi

L'analisi delle situazioni pericolose ha l'obbiettivo di anticipare ogni rischio insito in sistemi che devono garantire un adeguato grado di sicurezza. Per ciascuna situazione a rischio, è necessario prendere in considerazione i seguenti aspetti:
- Livello di rischio - entità delle conseguenze di un guasto nel momento in cui esso si verifica
- Probabilità - probabilità che si verifichi un guasto
- Tempo di esposizione - durata della finestra temporale in cui il guasto può verificarsi
- Tempo di tolleranza - periodo di tempo durante il quale il sistema può sopportare un guasto nel caso in cui questi si verifichi
- Tempo di rilevamento - tempo che intercorre dal momento in cui il guasto si verifica e quello in cui viene rilevato
- Misure di controllo - presenza di meccanismi che permettano di prevenire il guasto o perlomeno minimizzare i suoi effetti.

Si prenda ad esempio un acceleratore lineare utilizzato per generare un fascio di elettroni o una radiazione fotonica nel corso di un trattamento medico. Nella tabella 1 vengono riportati i potenziali pericoli.
Un sistema di sicurezza indipendente potrebbe contribuire ad attenuare i due pericoli maggiori bloccando l'uscita del fascio nel momento in cui questo viene attivato in maniera non corretta. Nella figura 1 viene riportato un confronto tra un controllore dotato di un sistema di sicurezza e uno privo di esso. In quest'ultimo caso, un controllore programmabile decide quando deve iniziare l'emissione del fascio. Esso è dotato di un sensore di radiazione che può essere usato per rilevare quando il fascio è in azione. Ciò dà origine a una situazione di pericolo in quanto la stessa Cpu viene impiegata contemporaneamente per implementare le funzioni di controllo e quelle di sicurezza. Nel caso l'emissione del fascio non sia corretta perché il funzionamento della Cpu si arresta per qualsivoglia motivo (compresi guasti di natura hardware e software), essa non sarà in grado di gestire le funzioni di sicurezza o avviare il ripristino da un meccanismo di guasto.
Nel secondo caso, un sistema di sicurezza indipendente esegue il monitoraggio sia del processore di controllo sia della durata e dell'intensità del fascio. Nel caso venga rilevato un guasto, essa ha la capacità di inserire un apposito otturatore nel fascio per bloccare la radiazione, indipendentemente dal fatto che il processore di controllo abbia ordinato all'acceleratore lineare di interrompere l'emissione del fascio. Entrambi i processori comunicano in fase di accensione quando eseguono i programmi di autodiagnostica. Nel caso l'inizializzazione di uno dei due processori non sia corretta, l'altro porrà il sistema in uno stato di sicurezza. Dopo il reset, un segnale di watchdog proveniente dal processore di controllo indica al processore di sicurezza che il software di controllo sembra stia girando in maniera corretta. Se il segnale di watchdog non cambia stato alla velocità prevista, il processore di sicurezza si preoccuperà di porre il sistema in una modalità di sicurezza.

Utilizzo di un modello ad albero

Un albero dei guasti fornisce un metodo grafico efficace per rappresentare i percorsi che possono favorire il presentarsi di un guasto. I simbolo logici vengono utilizzati per rappresentare un guasto che può essere prodotto da più cause (porta OR) o da più elementi che si verificano in contemporanea (porta AND). Una possibile rappresentazione ad albero relativa all'esempio appena descritto viene riportata nella figura 2. Alla sommità dell'albero vi è una porta AND, che sta ad indicare che un guasto prodotto da una dose eccessiva di radiazioni si verifica solamente nel momento in cui il segnale di controllo relativo al fascio è attivo. Immediatamente sotto vi è una porta OR, che indica che ciascuna delle tre sorgenti può in modo spurio far entrare in funzione il fascio. Una causa può essere una Cpu bloccata (ad esempio a causa di interferenze elettromagnetiche), oppure un comando non valido (provocato ancora da un'interferenza oppure da un difetto software), mentre una terza causa è il guasto del timer che arresta il fascio nel momento in cui la dose prevista deve essere erogata.

National Semiconductor
readerservice.it n. 45