www.asap-firmware.com

Prefazione

Teoria della progettazione di una interfaccia utente
Interfacce utente grafiche e alfanumeriche
Interfacce utente alfanumeriche
Gestione della tastiera in un pannello di controllo
Gestione di display alfanumerici
Gestione della visualizzazione dei dati
Gestione di lingue diverse
La Soluzione ASAP
Esempi in linguaggio C

Titolo: FWS0001A

Interfaccia Uomo-Macchina

Teoria della progettazione di una interfaccia utente

Un'interfaccia utente dovrebbe avere le caratteristiche degli strumenti di uso quotidiano, come il martello, che non indirizzerà mai l'utente a formulare l'obiettivo di avvitare una vite!

Molte delle frustrazioni che gli utenti sperimentano derivano proprio dal fatto che l'interfaccia in qualche modo invita a formulare intenzioni incompatibili con i vincoli del sistema artificiale e quindi irrealizzabili.

L'analogia è una delle soluzioni più efficaci per aiutare gli utenti a pianificare le loro azioni; l'utente sviluppa in ogni caso un modello del sistema e più il modello è compatibile con il modello reale del sistema, più l'utente avrà successo nella sua interazione con il sistema.

Un altro aspetto da considerare è aiutare l'esecuzione delle azioni stesse, scegliendo modalità che facilitino l'azione motoria; una leva molto dura, una tastiera mal orientata, una scarsa visibilità dello schermo, impediscono o rendono difficile l'esecuzione delle azioni.

L'informazione deve essere presentata all'utente in modo che la sua comprensione dello stato del sistema sia la più adeguata possibile; l'uso di icone e di simboli quotidiani evita in parte all'utente di dover apprendere un linguaggio specifico, perché quei simboli fanno già parte del suo bagaglio culturale.

Nella progettazione di una interfaccia utente, l'idea centrale è che occorra partire da una valutazione delle difficoltà che l'utente incontra nell'interazione; quest'ultima è guidata-controllata sostanzialmente dalla *valutazione* di ciò che sta avvenendo e dall'*esecuzione* di programmi motori che rispondono alla domanda "cosa occorre fare?".

Per quanto riguarda la *valutazione*, l'interfaccia deve consentire all'operatore di avere una buona consapevolezza sullo stato del sistema e della situazione; per quanto riguarda l'*esecuzione*, è necessario ridurre il numero di passaggi per trasformare un'intenzione in un'azione e l'interfaccia deve poter evocare "cosa fare" e "come fare".

Alcuni studi sull'interazione uomo-macchina hanno, inoltre, messo in risalto altri principi base di cui occorre tener conto nella progettazione, come la consistenza e la dipendenza contestuale.

Riguardo alla consistenza ci deve essere uniformità nei comandi così che l'utente, dopo averne compreso la logica, possa utilizzarla nei diversi contesti in cui si trova ad operare; relativamente alla dipendenza contestuale, dovrebbero essere visualizzate solo le informazioni rilevanti allo stato in cui si trova l'applicazione.

Interfacce utente grafiche e alfanumeriche

Le interfacce utente grafiche (GUI) consistono in un sistema di windowing o sistema a finestre che controlla gli oggetti sullo schermo ed in un insieme di "regole" che determinano lo stile delle interfacce delle applicazioni.

Il noto sistema di windowing fornisce un ambiente a finestre in cui più applicazioni possono coesistere sullo stesso schermo; le interazioni sono guidate da eventi ("event-driven") che vengono comunicati dal sistema all'applicazione, come ad esempio la pressione di un tasto o l'invio di un messaggio da una finestra ad un'altra.

Le regole di stile determinano il "look&feel" di un'interfaccia grafica, specificando in quale modo gli oggetti grafici come menu, icone, finestre dovrebbero apparire e quale dovrà essere il loro comportamento.

Se i suggerimenti di stile non vengono seguiti si rischia di perdere parte dei benefici derivanti dalle GUI; è consigliabile pertanto utilizzare degli strumenti di sviluppo che assistano lo sviluppatore nella creazione di interfacce conformi alle linee guida.

In ambiente PC i linguaggi di programmazione ad oggetti (es. Visual Basic, C++) sono già conformi allo stile Windows; in ambiente embedded dove si utilizzano microcontrollori con core a 8-16bit e display LCD grafici di dimensioni contenute è necessario dotarsi di librerie dedicate e di driver software specifici per il tipo di LCD controller montato sul display (es. Samsung, Hitachi, Toshiba).

Interfacce utente alfanumeriche

Nel mondo embedded l'utilizzo di un display LCD grafico deve essere giustificato da ragioni tecnico/commerciali; questi prodotti negli ultimi tempi hanno visto diminuire il rapporto prezzo/prestazioni facendo aumentare la loro presenza sul mercato.

Nella maggior parte dei casi si utilizzano dei display LCD alfanumerici con un certo numero di caratteri per riga.

In questo caso l'interfaccia utente può risultare meno immediata da utilizzare, ed è per questo che deve essere progettata con maggior attenzione.

Si consiglia l'utilizzo di un sistema di menu ad albero, in cui si può scendere o salire di livello, e la scelta di pochi tasti a uso generico come ENTER, ESC e le frecce di direzione.

Non è possibile creare dei fonts di caratteri, ma si utilizzano i font fissi presenti nella memoria ROM del controller; sono comunque presenti in memoria RAM dei font configurabili per creare lettere o simboli diverse dallo standard ASCII (es. caratteri cirillici).

Gestione della tastiera in un pannello di controllo

Le tastiere utilizzate nei pannelli di controllo possono essere costituite da tradizionali tasti meccanici o da tasti a membrana.

Il firmware di gestione dei tasti deve essere realizzato in modo da facilitare l'acquisizione della pressione del tasto e filtrare pressioni e rimbalzi indesiderati.

Se è presente un avvisatore acustico e i tasti non forniscono l'effetto tattile, è consigliabile generare un "beep" per informare l'utente dell'avvenuta pressione.

E' consigliabile anche gestire la pressione continua per agevolare operazioni come l'incremento-decremento dei valori di grandezze tarabile da parte dell'utente.

Gestione di display alfanumerici

Il firmware deve essere in grado di automattizzare nel miglior modo possibile il movimento tra i vari menu utente.

Un menu è composto in genere da una descrizione fissa, e da una parte variabile che deve essere aggiornata come valori di grandezze o messaggi di stato del sistema.

E' importante che la parte fissa venga scritta solo al primo ingresso nel menu e che la parte variabile venga trasferita sul display a partire dalla sua posizione senza cancellare le altre parti per evitare un fastidioso effetto tremolante (flicker).

In genere i menu più utilizzati riguardano la visualizzazione delle grandezze relative al dispositivo (misure), la visualizzazione di messaggi di stato e allarme, l'esecuzione di comandi, la configurazione dei parametri.

Gestione della visualizzazione dei dati

Il tipo di dati che vengono visualizzati è molteplice e dipende dall'apparecchiatura: possiamo avere grandezze numeriche in vari formati (decimale-Hex-binario), stringhe di testo, storico degli eventi.

Il firmware deve essere strutturato in modo da conoscere quali sono le grandezze da visualizzare nel menu corrente, se in modo dinamico o statico e il loro formato esterno.

Molte volte si utilizzano sistemi a intelligenza distribuita dove più dispositivi sono collegati tra loro in una rete di comunicazione (es. RS485,CAN Bus), e quindi la logica di visualizzazione deve tener conto anche del fatto che il dispositivo che contiene la grandezza sia effettivamente connesso e comunicante.

Gestione di lingue diverse

La configurazione della lingua è ormai una consuetudine.

Nei display alfanumerici esistono dei limiti per il fatto che si devono utilizzare i caratteri standard ASCII; lingue con caratteri molto diversi (es. cirillico,cinese) possono essere implementate via hardware richiedendo al fornitore del display un controller custom con il set di caratteri dedicato o via software sfruttando i caratteri (in genere max otto) configurabili in memoria RAM.

La soluzione hardware è quella più efficace, ma in genere viene attivata dal fornitore del display solo per una quantità notevole di pezzi; la soluzione software richiede una maggiore complessità e lentezza di esecuzione, ma consente una maggiore flessibilità in risposta a cambiamenti futuri.

Un aspetto importante da considerare è la massima riduzione possibile dello spazio in memoria ROM necessario, e la possibilità di modificare il set di lingue a disposizione.

Per ridurre lo spazio necessario il firmware deve essere strutturato in modo da eliminare caratteri inutili, come ad esempio gli spazi che servono dopo la scrittura di un menu o di un messaggio per arrivare alla fine di una riga.

La modifica del set di lingue a disposizione da parte dell'utente finale è permessa dall'uso di memorie di tipo Flash divise in settori, e dalla capacità del firmware di gestire una organizzazione dei menu e dei messaggi aventi lunghezza diversa.

La Soluzione ASAP

ASAP realizza interfacce utente semplici ed evolute con display LCD di tipo grafico o alfanumerico.

Grazie all'esperienza maturata nel corso degli anni, sono state realizzate delle librerie che consentono di automatizzare la procedura completa dell'interfaccia utente, dalla rilevazione della pressione del tasto alla visualizzazione automatica di stati e allarmi, nonché alla gestione di uno storico eventi.

Vengono seguite delle regole che permettono la portabilità su sistemi diversi, basate sulla creazione di driver specifici per l'hardware in uso che dialogano con gli strati software di livello superiore.

In seguito vengono riportare parti di codice scritte in linguaggio C, utilizzate nelle nostre applicazioni.

Per ogni ulteriore informazione, ASAP è raggiungibile al seguente indirizzo di posta elettronica :
info@asap-firmware.com

Esempi in linguaggio C

Vettore dei testi relativi ai menu in lingua base:

```
const char *pt_menu_italiano[MAXMENU] =
{
/*  --------- display numero (MN_TITLE) ----------
----1   5    10   5    20   5    30   5    40   5 */
   "  *** Interfaccia uomo-macchina ***",
};
```

Vettore dei testi relativi ai menu in lingua 1:

```
const char *pt_menu_lingua1[MAXMENU] =
{
/*  --------- display numero (MN_TITLE) ----------
----1   5    10   5    20   5    30   5    40   5 */
   "  *** Man-Machine Interface ***",
};
```

Vettore dei testi relativi ai messaggi di allarme in lingua base:

```
const char *pt_allarmi_italiano[MAXALLARMI] =
{
/*  --------- display numero (ALL_COD00) ----------
----1   5    10   5    20   5    30   5    40   5 */
   "A00: BATTERIA SCARICA",
};
```

Vettore dei testi relativi ai messaggi di allarme in lingua 1:

```
const char *pt_allarmi_lingua1[MAXALLARMI] =
{
/*  --------- display numero (ALL_COD00) ----------
----1   5    10   5    20   5    30   5    40   5 */
   "A00: BATTERY DISCHARGE",
};
```

Vettore dei testi relativi ai messaggi generici lingua base:

```
const char *pt_messaggi_italiano[MAXMESSAGGI] =
{
/*  --------- display numero (MSG0) ----------
----1   5    10   5    20   5    30   5    40   5 */
   "abilitato",
};
```

Vettore dei testi relativi ai messaggi generici in lingua 1:

```
const char *pt_messaggi_lingua1[MAXMESSAGGI] =
{
/*  --------- display numero (MSG0) ----------
----1   5    10   5    20   5    30   5    40   5 */
   "enabled",
};
```

Vettore per la gestione della scelta della lingua corrente:

```
const char **menu_lingue[]=
{
 pt_menu_italiano,
 pt_menu_lingua1,
};
```