Architetture dei Sistemi   
di Elaborazione

Nome e Cognome …Pierpaolo…Bene……

Matricola ……..……S319841……

Il codice compila senza errori: sì [x] no [ ]

Ho provato il progetto in emulazione: sì [ ] no [x]

Ho provato il progetto su board: sì [x] no [ ]

L’ambiente di debug è stato utilizzato : sì [x] no [ ]

Desidero ritirarmi [ ]

Sistemi basati su ARM –8 febbraio 2024 – T1

Leggere con attenzione:

1. Occorre sviluppare un progetto ARM usando l’IDE KEIL µVision.
2. Effettuare login su propria area al LABINF ed usare il software disponibile per editare, compilare e debuggare il codice.
3. Utilizzare l’area desktop sul computer del LABINF per creare il vostro progetto.
4. Utilizzare la scheda LANDTIGER o l’emulatore con tutte le non-idealità abilitate per debuggare il progetto
5. Sono inibiti tutti gli accessi ad internet.
6. Si possono utilizzare progetti esistenti, prelevati dalla propria chiavetta USB, ed è possibile consultare materiale cartaceo.
7. Entro l’orario di consegna, occorre finalizzare il salvataggio di tutti i file (valido anche per la parte di modern architecture) e **copiarli nella propria area personale Z:/ all’interno della cartella che contiene le tracce**. Le consegne in ritardo (con file salvati oltre l’orario massimo di consegna) non vengono considerate valide e conducono in ogni caso all’insufficienza.
8. In caso non sia possibile compilare con successo il progetto consegnato, la prova sarà considerata insufficiente. Si richiede di predisporre l’ambiente di debug con le watch che permettono di seguire il flusso del programma.

Esercizio 1 (max 30 punti)

Si vuole realizzare un gioco che chieda all’utente di premere un pulsante per una serie di periodi di tempo e misurarne l’accuratezza. I periodi di attesa sono codificati in un vettore inizializzato a tempo di compilazione, come da esempio:

* + unsigned char VETT[N] = [35, 20, 80, 46]; // N = 4 in questo esempio

Sviluppare le seguenti funzionalità per la scheda LANDTIGER ed il system-on-chip LPC1768.

1. Il sistema deve visualizzare in binario sui LED il tempo in decimi di secondo per cui l’utente deve tenere premuto il pulsante KEY1 (as esempio il valore 35 corrisponde a 3,5 secondi).
2. Alla pressione di KEY1, l’attuale valore da conteggiare verrà visualizzato sui LED e inizia un conto alla rovescia non visualizzato all’utente.
   * Al rilascio del pulsante i LED si spengono
     + L’obiettivo del gioco è quello di rilasciare il pulsante il più vicino possibile alla scadenza del conto alla rovescia
     + È possibile che l’utente rilasci il pulsante prima o dopo lo scadere del conto alla rovescia
     + Il programma deve depositare il tempo di pressione in un secondo vettore VETT2[N] di tipo unsigned char su 8 bit inizialmente vuoto.
3. Dopo aver rilasciato KEY1 e salvato il tempo di pressione su VETT2[i], il gioco ricomincia dal punto 1 con un nuovo valore di conteggio a ritroso.
4. Quando il vettore VETT2 satura, allora deve essere lanciata la seguente funzione ASSEMBLER, comprensiva di istruzioni condizionali:

unsigned char analisi\_accuratezza(unsigned char VETT[], unsigned char VETT2[],   
 const unsigned int N, char RES[]));

tale funzione riceve come parametri i vettori VETT e VETT2, N, e RES, con N lunghezza massima dei vettori. La funzione serve a calcolare l’accuratezza nelle pressioni da parte dell’utente. La funzione restituisce in RES[i] la distanza temporale (in decimi di secondo e in valore assoluto) tra VETT[i] e VETT2[i], cioè la differenza tra il tempo di pressione richiesto e il tempo di pressione effettivo. Inoltre, la funzione deve ritornare come valore di ritorno la media aritmetica dei valori contenuti in RES.

1. Il contenuto di RES deve essere visualizzato tramite i led:
   * Se RES[i] > media calcolata dalla funzione asm: il valore deve essere visualizzato per una durata di 0,5 secondi sui LED con un periodo di 0,100 s.
   * Se RES[i] <= media calcolata dalla funzione asm: il valore deve essere visualizzato per una durata di 1 secondi sui LED con un periodo di 0,250 s.
   * Durante il periodo di visualizzazione non sarà possibile acquisire altri valori né chiamare la funzione assembly.
2. Una volta conclusa la visualizzazione del risultato, il sistema riprende in modo ciclico dal punto 1 con VETT2 e RES opportunamente gestiti.