Architetture dei Sistemi   
di Elaborazione

Nome e Cognome ………………………

Matricola …………………………

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

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

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

L’ambiente di debug è stato utilizzato : sì [] 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 sistema in grado di effettuare misure temporali ed elaborarle.

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

1. Il sistema deve misurare in centesimi di secondo il tempo che intercorre tra la pressione del pulsante KEY1 e la pressione del pusante KEY2. Tale valore deve essere memorizzato in una variabile VAR di tipo unsigned su 32 bit:
2. A valle della registrazione in VAR del tempo tra pressioni, i LED devono visualizzare i seguenti valori ad intervalli regolari (alternanza ogni 120ms)
   * il valore su 8 bit ottenuto tramite la seguente operazione logica tra i byte che compongono VAR:

LED0-7 = (VAR31-24 OR VAR23-16) XOR (VAR15-8 AND VAR7-0)

* + il valore inverso di LED0-7 ottenuto tramite l’operatore logico NOT

1. Ogni 5 secondi, il valore corrente di VAR viene copiato nella prima posizione libera di un vettore chiamato VETT, istanziato di tipo opportuno e composto di N elementi (usare N = 6 nella versione consegnata)
2. Quando il vettore VETT satura, allora deve essere lanciata la seguente funzione ASSEMBLER:

unsigned char contatore\_inrange(unsigned int VETT[], const unsigned int N);

tale funzione riceve come parametri il vettore VETT, N e serve a elaborare i valori in VETT. La funzione restituisce il numero dei valori registrati che cadono strettamente tra i valori memorizzati in VETT[0] e VETT[N-1] (che vanno esclusi dal conteggio). Tale valore è espresso su 8 bit.

1. Il risultato deve essere visualizzato tramite i led per una durata di 6 secondi. Durante il periodo di visualizzazione non sarà possibile acquisire altri valori. Una volta finito il tempo di visualizzazione del risultato, il sistema riprende dal punto 1 con VAR e VETT “svuotati” così come inizialmente.