

Architetture dei Sistemi di Elaborazione

Esempio preparato in vista dell'appello I, 2023

Leggere con attenzione:

- Occorre sviluppare un progetto ARM usando l'IDE KEIL μ Vision.
- Effettuare login su propria area al LABINF ed usare il software disponibile per editare, compilare e debuggare il codice.
- Utilizzare l'area desktop sul computer del LABINF per creare il vostro progetto.
- Utilizzare la scheda LANDTIGER o l'emulatore con tutte le non-idealità abilitate per debuggare il progetto.
- Sono inibiti tutti gli accessi ad internet.
- Si possono utilizzare progetti esistenti, prelevati dalla propria chiavetta USB.
- 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.
- 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 ed i breakpoint che permettano di seguire il flusso del programma.**

Nome e Cognome _____

Matricola _____

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

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

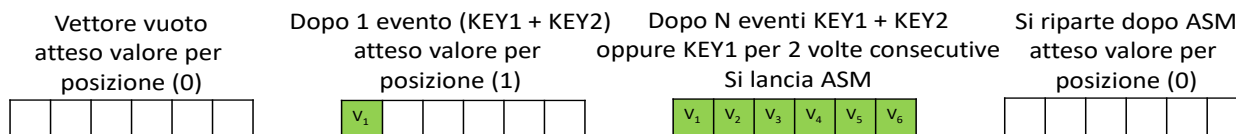
Ho provato il progetto su board: sì [] no []

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

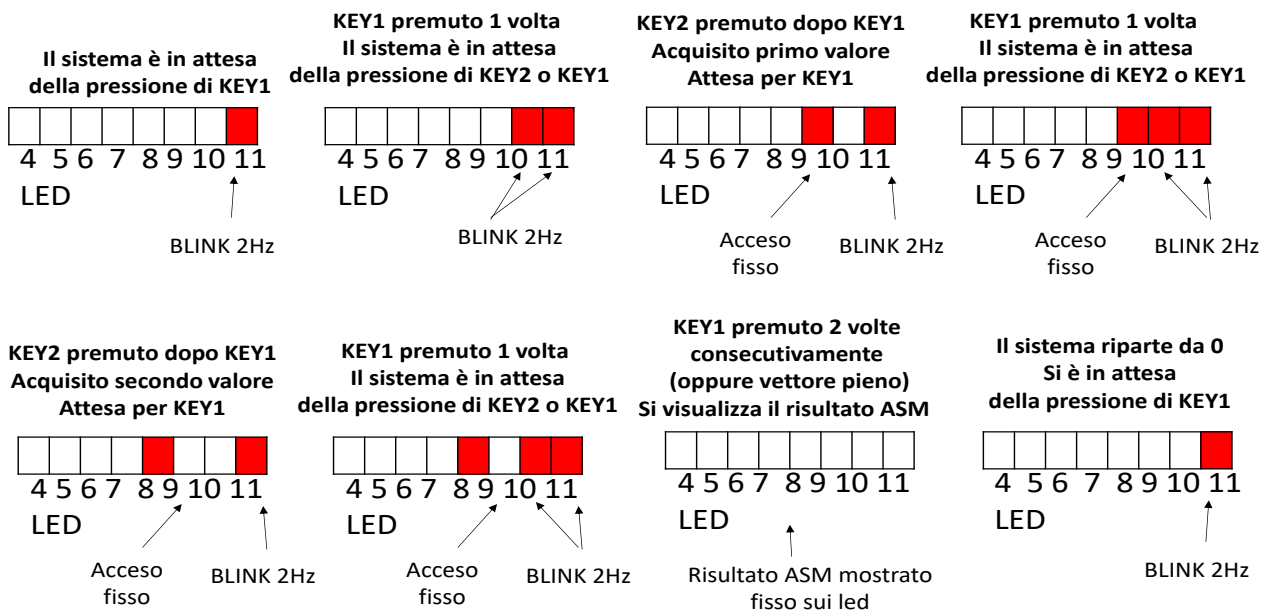
Desidero ritirarmi []

- 1) Si consideri un vettore **VETT** di **N** di valori interi espressi su 8 bit. **N** è una costante dichiarata a tempo di compilazione e il vettore sarà dichiarato come **unsigned char VETT[N]**.
- 2) In una prima fase, si devono acquisire le elementi del vettore, derivandoli dal Timer 1 programmato per innescare interruzioni con una frequenza di 5kHz. In particolare:
 - Ogni volta che Timer 1 raggiunge il suo conteggio, deve essere incrementata una variabile chiamata VAR (inizialmente pari a 0).
 - Alla pressione successiva di KEY1 e KEY2 (ovvero dopo che sia KEY1, sia KEY2 sono stati premuti in sequenza uno dopo l'altro) il valore di VAR deve essere scritto all'interno di VETT nella posizione attualmente libera, partendo dalla posizione 0.
 - Una volta pieno (dopo N inserimenti) oppure se si registra la doppia pressione di KEY1 (2 volte consecutive), il programma deve lanciare la funzione ASM descritta successivamente.

Scrittura del vettore VETT[]



- 3) Durante la fase di acquisizione dei valori, i led 10 e 11 mostrano se si è in attesa della prima pressione di KEY1 o KEY2, come mostrato in figura, lampeggiando alla frequenza di 2Hz, mentre i 6 rimanenti (LD4-9) mostrano il numero di valori attualmente memorizzati in VETT (ovviamente questo limita la dimensione del vettore a 2^6 elementi)



La funzione scritta in linguaggio ASSEMBLER da invocare alla saturazione del vettore o dopo la pressione di KEY1 due volte consecutivamente, deve dire se la sequenza è monotona crescente (non strettamente). Se sì, restituisce il valore 0xFF, altrimenti 0x55.

unsigned int monotono_crescente(unsigned char V[], unsigned int num);

- dove num indica il numero di valori inseriti nel vettore finora

- 4) Il valore risultato deve essere presentato fisso usando tutti i LED. Durante la visualizzazione non è possibile operare con KEY1 e KEY2 (le funzionalità sono disattivate).
- 5) Alla ulteriore pressione di INT0, la sequenza di visualizzazione viene terminata, ed il processo ricomincia da 2) dopo aver "svuotato" il vettore ed aspettando il valore da inserire alla posizione (0).