

# Architetture dei Sistemi di Elaborazione

Sistemi basati su ARM T2.1 – 3 febbraio 2020

## Leggere con attenzione:

- 1) Durata: 2 ore.
- 2) Occorre sviluppare un progetto ARM usando l'IDE KEIL  $\mu$ Vision.
- 3) Effettuare login su propria area al LABINF ed usare il software disponibile per editare, compilare e debuggare il codice.
- 4) Utilizzare la scheda LANDTIGER per prototipare il progetto
- 5) L'accesso al portale della didattica è garantito dal LABINF (url: didattica.polito.it) mentre sono inibiti tutti gli altri accessi. Tentativi di accesso ad altri siti provocherà l'immediata segnalazione da parte dei tecnici del LABINF che monitorano le attività di rete durante la prova, e la conseguente espulsione della persona individuata.
- 6) Si possono utilizzare progetti esistenti, presenti su portale o su propria chiavetta USB, ed è possibile consultare materiale cartaceo.
- 7) Entro l'orario di consegna, occorre caricare sul portale, alla pagina elaborate, il progetto sviluppato. Creare un archivio chiamato 20200203\_T2\_1.zip che contenga l'intero progetto. Le consegne in ritardo 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.

Nome e Cognome \_\_\_\_\_

Matricola \_\_\_\_\_

Firma \_\_\_\_\_

Orario di consegna su portale: \_\_\_\_:\_\_\_\_

Bernardi [ ] – Sanchez [ ]

Code compiles: yes [ ] no [ ]

## Esercizio 1 (max 30 punti)

Sviluppare le seguenti funzionalità utilizzando la scheda LANDTIGER e il system-on-chip LPC1768.

- 1) Si vuole usare il TIMER 1 dell'LPC1768 come generatore di numeri. Per questo, si programmi il timer in modo ciclico senza interruzioni con un periodo di 2,43 minuti.
- 2) Si devono acquisire **N** numeri interi da inserire in un vettore **VETT**. I numeri da memorizzare sono senza segno e minori di 256 (unsigned char).
  - La pressione del pulsante KEY1 permette di acquisire un valore leggendo il registro TC dal timer predisposto
  - Dal valore letto sono estratti 8 bits (da indice 23 a indice 16) per essere scritti in una variabile unsigned char chiamata VAL.
- 3) Durante l'acquisizione dei numeri, i LED da 11 (MSB) fino a 4 (LSB) mostrano la codifica binaria dell'indice del valore da acquisire attualmente. Ad esempio, inizialmente i LED saranno tutti spenti, una volta acquisito il primo numero sarà mostrato 1, e così via fino a N-1.
- 4) Ogni volta che si acquisisce un valore, si deve lanciare la seguente funzione scritta in linguaggio ASSEMBLER che ordina il dato nuovo nel vettore VETT; i valori maggiori devono occupare le posizioni meno significative del vettore, alla posizione 0 risulta essere memorizzato il massimo attuale.

```
unsigned char get_and_sort(unsigned char* VETT, unsigned char VAL, int n);
```

```
/* dove VAL è il nuovo numero acquisito */
```

```
la funzione restituisce
```

```
○ Il valore VAL
```

- 5) Una volta acquisiti gli **N** valori in VETT, questi devono essere mostrati in sequenza sui LED con una frequenza di  $\frac{1}{2}$  Hz, seguita da LED tutti spenti.
- 6) Il processo ricomincia da 2).