



UNIVERSITÀ DEGLI STUDI DI PALERMO
DIPARTIMENTO DI INGEGNERIA
CORSI DI LAUREA IN INGEGNERIA CIBERNETICA ED ELETTRONICA

ELETTRONICA DEI SISTEMI DIGITALI - ELETTRONICA DEI SISTEMI EMBEDDED
Prof. C.G. GIACONIA A.A. 2019/2020 14 Settembre 2020

Prova Finale riservata agli studenti del Nuovo Ordinamento

Durata della Prova: 100 minuti

Candidato (scrivere a stampatello):

Nome: _____ Cognome: _____ Consegnato: ____ (SI - NO)

Corso di Laurea: _____ Matricola: _____ Macchina: _____

Riportare i dati personali e riconsegnare al docente alla fine della prova.

IMPORTANTE

1. I programmi devono essere realizzati in **linguaggio C** ed essere eseguibili su:

➤ **ATMEGA328P-XMINI** (con l'ausilio della **ESDPLab EDUBOARD**)

NOTE RELATIVE ALL'ARCHIVIAZIONE DEI RISULTATI DELLA PROVA

Per la corretta archiviazione bisogna seguire i passi seguenti:

Il quesito va risolto con l'uso dell'ambiente di sviluppo **ATMEL STUDIO 6.2**, installato sulle macchine dell'Aula Informatica.

Indicazioni per il corretto salvataggio:

1. Formare una cartella denominata con il proprio cognome_numero_di_matricola sul Desktop (es.: **C:\Utenti\f100\Desktop\cognome0123456**);

2. Alla consegna salvare l'intero progetto in un unico file **.zip** chiamato con il proprio numero di matricola (es.: **cognome0123456.zip**) e copiarlo sul **DESKTOP**

**IL NON ATTENERSI A QUESTA PROCEDURA COMPORTA L'IMPOSSIBILITÀ,
PER IL DOCENTE, DI LEGGERE I CONTENUTI DELLA PROVA ED IL CONSEGUENTE**

ESITO NEGATIVO DELLA STESSA

QUESITO:

Scrivere un programma che sia in grado di controllare il funzionamento di un frullatore che può erogare 4 velocità possibili (minima, media, alta e max).

All'accensione il frullatore sarà fermo (LED spenti) e la velocità voluta sarà selezionabile tramite i tasti **B2** e **B3**.

In particolare premendo il pulsante **B3** si incrementa la velocità di uno step alla volta fino alla massima; mentre con **B2** si rallenta la velocità di uno step alla volta fino allo stop completo.

La velocità corrente verrà visualizzata tramite l'intensità luminosa emessa dai LED **L1**, **L2**, **L3** ed **L4**. In particolare dovrà essere rispettata la seguente tabella di specifiche:

Velocità Frullatore	Intensità Luminosa L1	Intensità Luminosa L2	Intensità Luminosa L3	Intensità Luminosa L4
Spento	0% (OFF)	0% (OFF)	0% (OFF)	0% (OFF)
Minima	25% del max	0% (OFF)	0% (OFF)	0% (OFF)
Media	25% del max	50% del max	0% (OFF)	0% (OFF)
Alta	25% del max	50% del max	75% del max	0% (OFF)
Max	25% del max	50% del max	75% del max	100% = max

Il tempo totale del ciclo di funzionamento del frullatore è determinato in **10 secondi** che partono da quando non si digita alcun pulsante.

Se invece, in un istante qualsiasi, viene premuto il tasto **B7**, esso produce lo spegnimento immediato del frullatore (LED tutti spenti) e porterà la macchina allo stato di pronto per una nuova attivazione.



Risultati Prova Finale del 14 Settembre 2020

Elettronica dei Sistemi Digitali

Elettronica dei Sistemi Embedded

(Prof. Costantino Giaconia)

#	Matricola	Data Prenotazione	Macchina	Valutazione	Esito	Orale S/N
1	0627713	02 set 2020, 11:29	***	***	ASSENTE	
2	0628257	02 set 2020, 10:03	01	30	AMMESSO ALL'ORALE	
3	0628268	28 ago 2020, 12:43	***	***	ASSENTE	
4	0628498	05 set 2020, 03:10	***	***	ASSENTE	
5	0640264	25 ago 2020, 00:19	***	***	ASSENTE	
6	0643986	23 ago 2020, 12:28	***	***	ASSENTE	
7	0650847	07 set 2020, 10:07	13	insuff	NON AMMESSO ALL'ORALE	
8	0650926	09 set 2020, 10:52	***	***	ASSENTE	
9	0650972	07 set 2020, 17:25	08	insuff	NON AMMESSO ALL'ORALE	
10	0651081	07 set 2020, 10:08	23	insuff	NON AMMESSO ALL'ORALE	
11	0651224	31 ago 2020, 08:51	41	22	AMMESSO ALL'ORALE	
12	0651340	04 set 2020, 11:41	***	***	ASSENTE	
13	0655359	14 ago 2020, 13:42	06	24	AMMESSO ALL'ORALE	
14	0655794	04 set 2020, 13:11	43	29	AMMESSO ALL'ORALE	
15	0656132	02 set 2020, 09:03	***	***	ASSENTE	
16	0656657	09 set 2020, 12:11	27	insuff	NON AMMESSO ALL'ORALE	
17	0660252	07 set 2020, 09:48	47	insuff	NON AMMESSO ALL'ORALE	
18	0660396	09 set 2020, 12:10	15	insuff	NON AMMESSO ALL'ORALE	
19	0664079	10 set 2020, 19:58	39	insuff	NON AMMESSO ALL'ORALE	
20	0664209	19 ago 2020, 15:08	07	***	NON CONSEGNA	
21	0664648	29 ago 2020, 13:11	31	***	NON CONSEGNA	
22	0665012	09 set 2020, 14:22	03	30	AMMESSO ALL'ORALE	
23	0665139	29 ago 2020, 09:38	29	22	AMMESSO ALL'ORALE	
24	0665264	07 set 2020, 20:36	53	insuff	NON AMMESSO ALL'ORALE	
25	0665649	29 ago 2020, 11:00	25	insuff	NON AMMESSO ALL'ORALE	
26	0665799	09 set 2020, 12:11	21	***	NON CONSEGNA	
27	0665817	08 set 2020, 09:06	49	insuff	NON AMMESSO ALL'ORALE	
28	0666050	03 set 2020, 14:57	***	***	ASSENTE	
29	0667294	08 set 2020, 12:19	63	24	AMMESSO ALL'ORALE	
30	0667544	07 set 2020, 13:09	19	28	AMMESSO ALL'ORALE	
31	0667799	20 ago 2020, 22:45	09	29	AMMESSO ALL'ORALE	
32	0667991	06 set 2020, 16:02	37	***	NON CONSEGNA	
33	0668208	08 set 2020, 12:19	67	insuff	NON AMMESSO ALL'ORALE	
34	0669039	20 ago 2020, 11:36	***	***	ASSENTE	
35	0669040	29 ago 2020, 17:19	35	insuff	NON AMMESSO ALL'ORALE	
36	0669698	10 set 2020, 11:53	17	28	AMMESSO ALL'ORALE	
37	0669804	08 set 2020, 12:19	55	insuff	NON AMMESSO ALL'ORALE	
38	0704357	02 set 2020, 10:11	***	***	ASSENTE	

```
1: /*
2:  * student.c
3:  * with comments
4:  * Created: 14/09/2020 09:06:06
5:  * Author: f100
6:  */
7: #include <avr/io.h>
8: #include <avr/interrupt.h>
9: // Definizione dei LED come uscite Porta C
10: // #define L0 (1<<PORTC0)
11: #define L1 (1<<PORTC1)
12: #define L2 (1<<PORTC2)
13: #define L3 (1<<PORTC3)
14: #define L4 (1<<PORTC4)
15: // #define L5 (1<<PORTC5)
16: // Definizione dei SW come ingressi Porta D
17: #define B2 (1<<PORTD2)
18: #define B3 (1<<PORTD3)
19: // #define B4 (1<<PORTD4)
20: // #define B5 (1<<PORTD5)
21: // #define B6 (1<<PORTD6)
22: #define B7 (1<<PORTD7)
23: // definizione di una macchina a tre stati (forse ne bastano due...)
24: typedef enum {reset, start, inserimento} state;
25: state stato=reset; // stato fondamentale
26: // variabili globali per il tempo, un contatore ed il valore iniziale
27: int time=0; int tick=0; int contatore=0; uint8_t storia=255;
28: // vettori di dato definiscono rispettivamente periodo ed valore da dare ai LED
29: int vect[6]={0,5,10,15,20}; //tempi di accensione dei LED su periodo base i 20ms
30: int valor[5]={0,2,6,14,30}; // Vettore dei valori che devono assumere i LED
31: // esempio: 6 in binario è 00000110 cioè i bit corrispondenti ad L1 ed L2
32:
33: int main(void)
34: {
35:     PORTC=0; DDRC=255; // porta C definita come uscita
36:     PORTD=255; DDRD=0; // porta D definita come ingresso
37:     PCICR=4; PCMSK2=255; //registri di controllo emissione interrupt di gruppo 2
38:     // in effetti la maschera (255) mette tutti gli ingressi in condizioni di
39:     // generare interrupt ma basterebbero B7, B2,B3
40:
41:     TCCR0A=2; // configurazione timer
42:     TCCR0B=0;
43:     OCR0A=249; // valore contatore per produrre base dei tempi di 1 millisecondo
44:     TIMSK0=2;
45:
46:     sei(); //abilitazione generale interrupt
47:
48:     while(1) //ciclo infinito della macchina a stati
49:     {
50:         switch(stato)
51:         {
52:             case reset:
53:                 TCCR0B=0;
54:                 PORTC=0;
55:                 time=0;
56:                 tick=0;
57:                 contatore=1;
58:                 stato=inserimento;
59:                 break;
60:
61:             case inserimento: // ecco lo stato superfluo
62:                 break;
63:
64:             case start:
```

```
65:         TCCR0B=3;
66:         break;
67:     }
68: }
69: }
70:
71: ISR(PCINT2_vect)
72: {
73:     int changebit=0;
74:     changebit=PIND^storia;
75:     if(changebit & b2) // polling sul tasto B2
76:     {
77:         if(contatore>1)
78:         {
79:             time=0; //SE pigiamo un tasto il timer riparte
80:             contatore--;
81:         }
82:     }
83:
84:     if(changebit & b3) // polling sul tasto B3
85:     {
86:         if(contatore<5) // aumentiamo la potenza del frullatore solo se
87:             //non abbiamo raggiunto il massimo
88:             contatore++;
89:             time=0;
90:             stato=start;
91:     }
92: }
93: if(changebit & b7) // se interviene il reset (B7) riparto
94: {
95:     stato=reset;
96: }
97: }
98:
99: ISR(TIMER0_COMPA_vect)
100: {
101:     time++; //variabile dedicata al timeout di 10 secondi
102:     tick++; // variabile dedicata alla regolazione dell'intensità dei LED (PWM)
103:
104:     if(time<10000) // time out a 10 secondi
105:     {
106:         if(tick<vect[contatore-1]) // preleviamo il periodo di accensione LED
107:             { // sulla base del valore del contatore definito nella macchina a stati
108:                 PORTC=valor[contatore-1]; //assegniamo il valore che deve assumere
109:             } //la porta (i LED che vanno accesi)
110:         else
111:         {
112:             PORTC=0;
113:         }
114:         if(tick>20) // se raggiunge 20 millisecondi
115:         {
116:             tick=0; //ricomincia (periodo del PWM)
117:         }
118:     }
119:     else
120:     {
121:         stato=reset;
122:     }
123: }
124: //Buono (29): Assenza di commenti; Stato inserimento ridondante. Per il resto ok
```