

Timer per ventilatore a pale

Luca Valcavi

31 luglio 2016

Indice

1	Schemi a blocchi	1
1.1	loop	1
1.2	Orologio	2
1.3	chkTEMPO	3
1.4	setLEDs	3
1.5	chkPremuto	4
1.6	chkRilasciato	4
2	Schema elettrico	5
3	PCB	5

Modifiche versione 4:

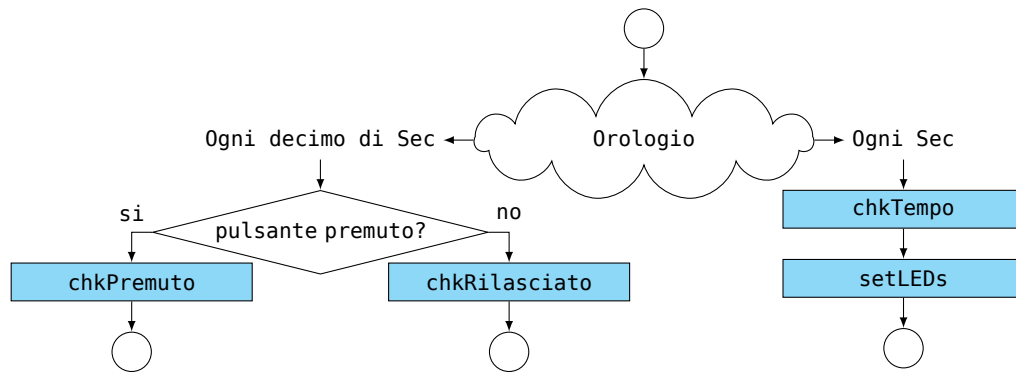
- correzione errori nella base dei tempi ($P_a=Q_a-X_a$ e $P_b=Q_b-X_b$)
- impostate ore 2,4,6,8
- spegnimento del led sulla porta 13 dopo un lampeggio iniziale quando si collega il circuito

Il circuito è un timer programmabile per l'accensione e lo spegnimento automatico della pala del ventilatore.

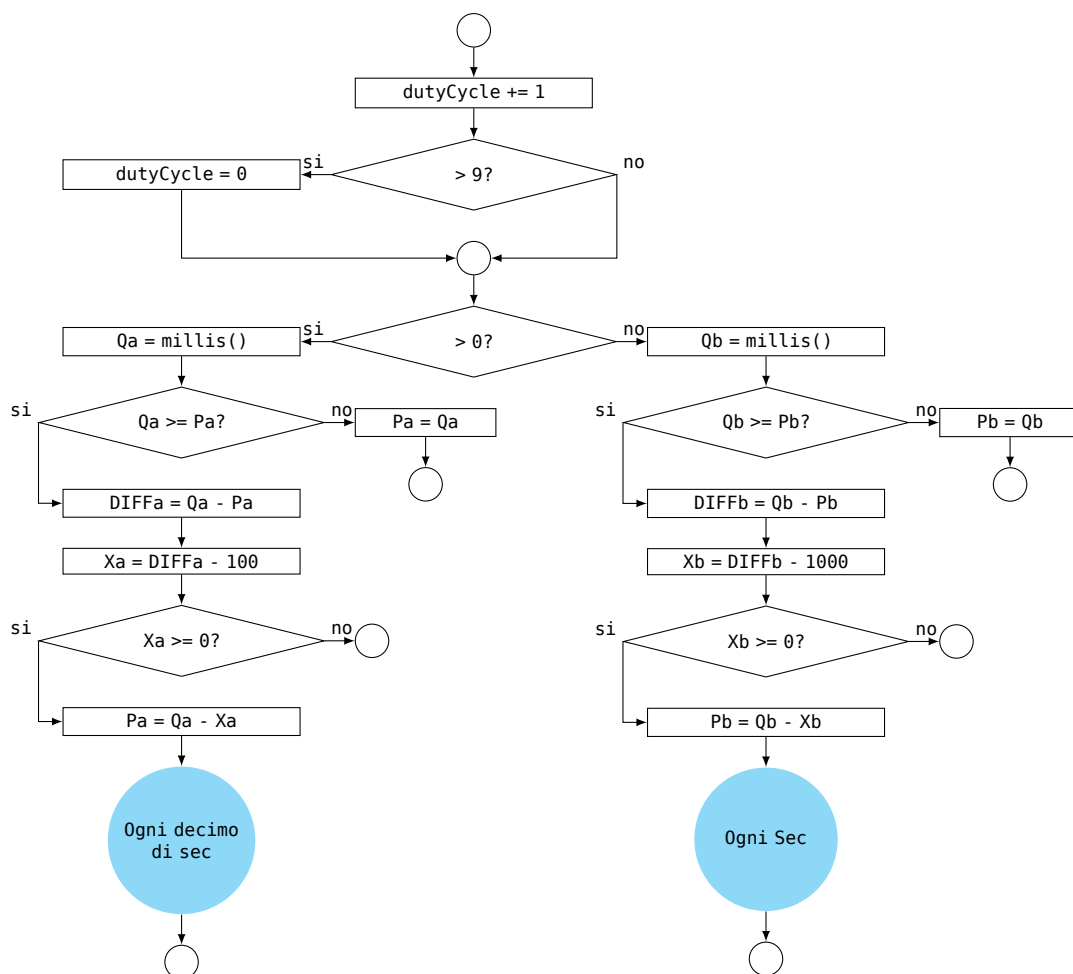
- Il timer si attiva premendo il pulsante.
- L'indicazione del tempo programmato è visualizzata da quattro LED luminosi che si accendono progressivamente ad ogni pressione del pulsante.
- Alla quarta attivazione il tempo impostato riparte da zero.
- Una pressione prolungata del pulsante permette di spegnere il timer se risulta già acceso. Se spento, la stessa prolungata accensione ne determina l'attivazione di un tempo predefinito pari a tre LED accesi.

1 Schemi a blocchi

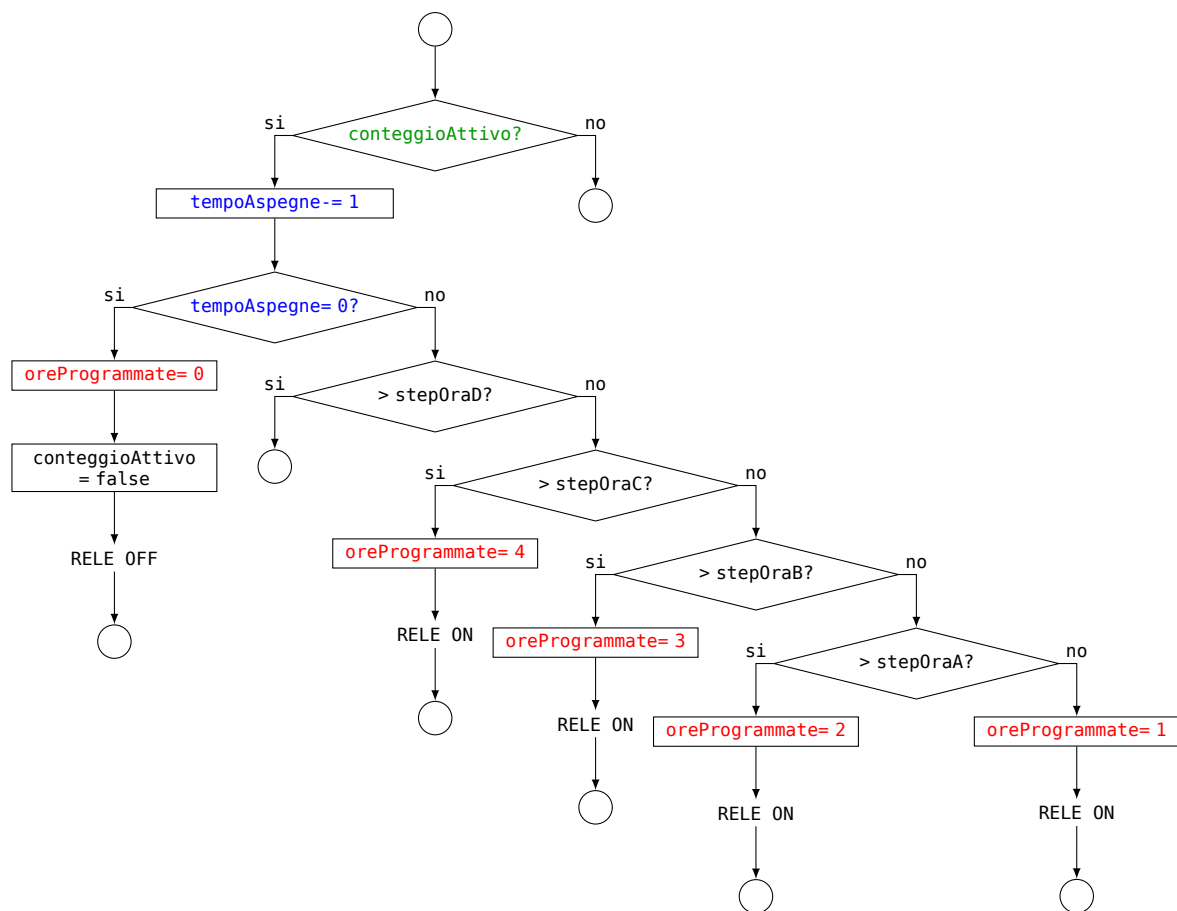
1.1 loop



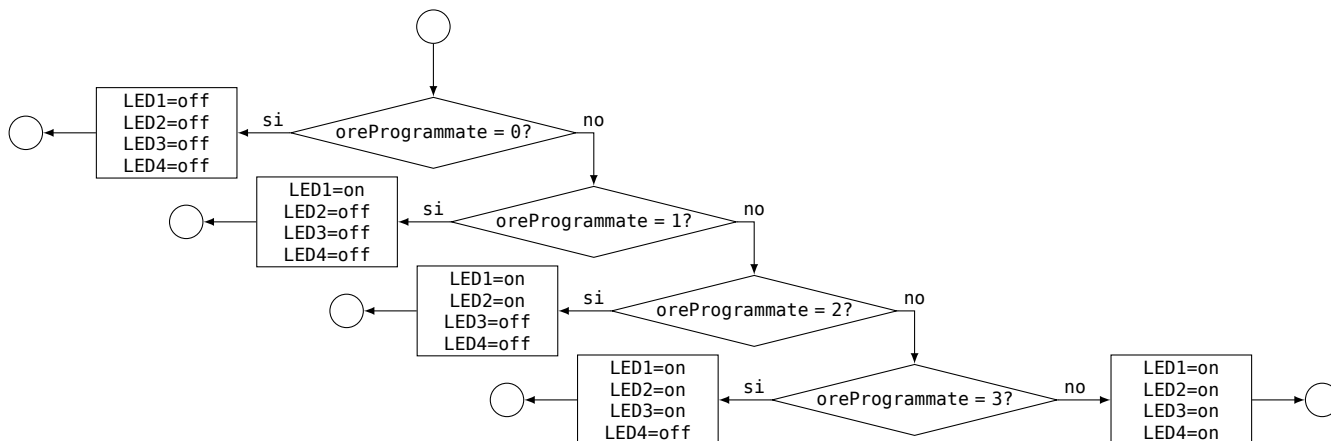
1.2 Orologio



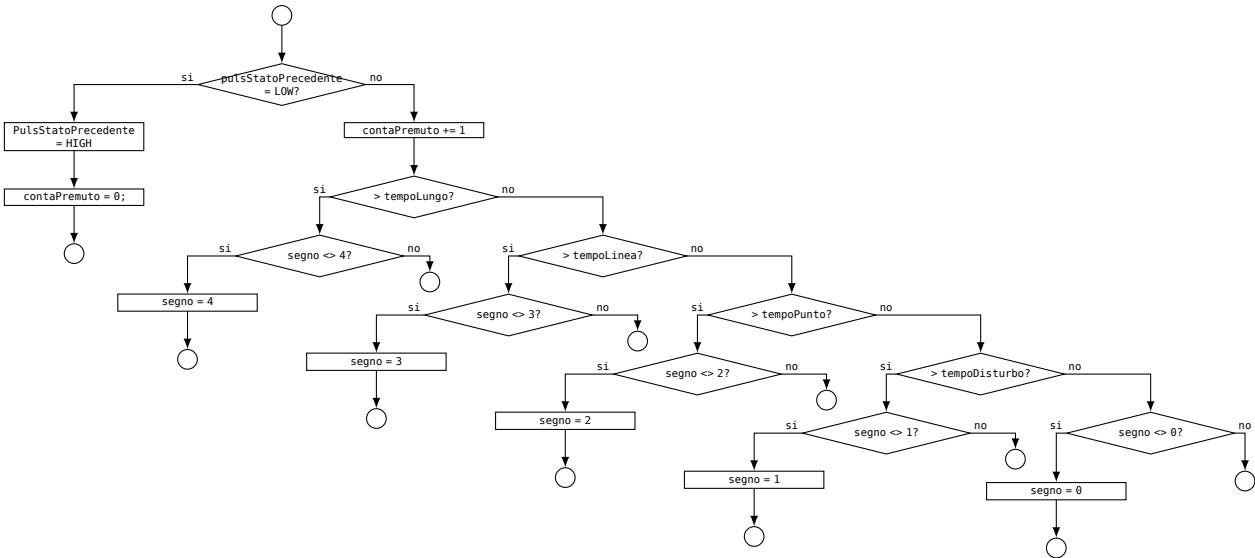
1.3 chkTEMPO



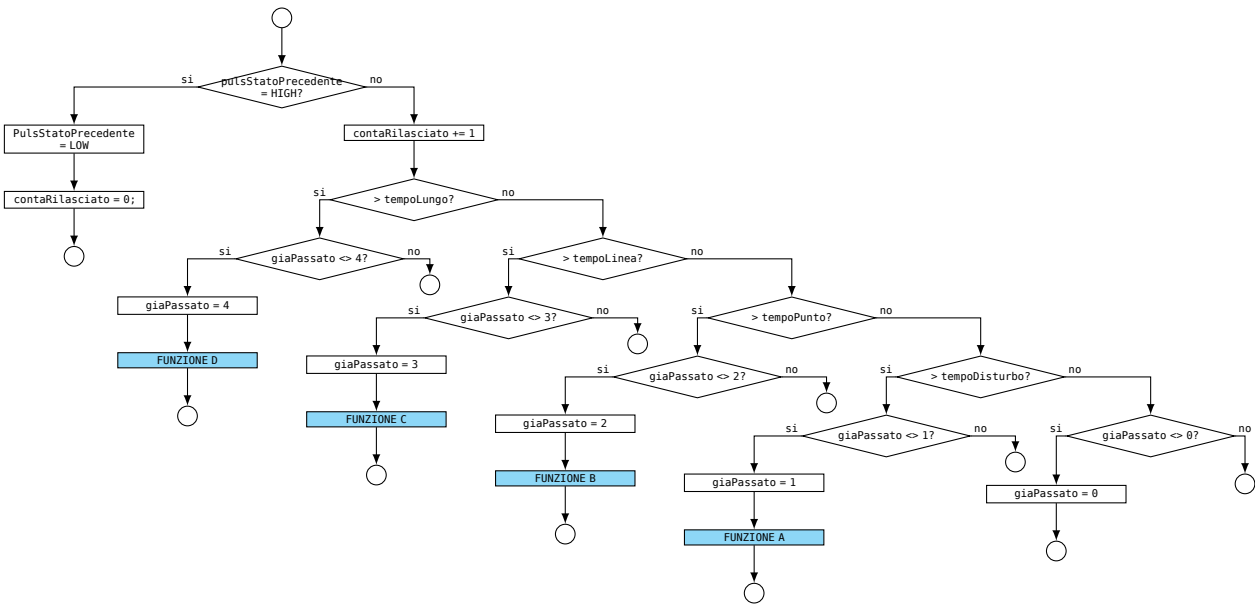
1.4 setLEDs



1.5 chkPremuto



1.6 chkRilasciato



1.6.1 FunzioneA

```
1 //=====
2 // programma le ore
3 //=====
4
5 void programmaLeOre(){
6     if (segno == 1){
7         //_____
8         // tempo corto (punto)
9         //_____
10        oreProgrammate += 1;
11        if (oreProgrammate > 4){
12            oreProgrammate = 0;
13        }
14    } else {
15        if (segno > 1 ){
16            //_____
17            // tempi piu lunghi (linea o tempo lungo)
18            //_____
19            if (oreProgrammate > 0){
```

```

19 oreProgrammate = 0; // spegne se acceso
   } else {
21 oreProgrammate = 3; // accende se spento
   }
23 }
   }
25 }
//=====
27 // impostazioni tempo
//=====
29 void impostaTempo(){
   switch (oreProgrammate){
31     case 0:
       tempoAspegne = 1; // fine
33     break;
       case 1:
35     tempoAspegne = step0raA;
       break;
       case 2:
37     tempoAspegne = step0raB;
       break;
39     case 3:
41     tempoAspegne = step0raC;
       break;
43     case 4:
45     tempoAspegne = step0raD;
       break;
   }
47 }

```

code/TimerVentilatore4/TimerVentilatore4.ino

1.6.2 FunzioneB

(non usata)

1.6.3 FunzioneC

```

1 //-----
   // attiva il conteggio del
3 // tempoAlloSpegnimento
   //-----
5 if (oreProgrammate > 0){
   conteggioAttivo = true;
7 }

```

code/TimerVentilatore4/TimerVentilatore4.ino

1.6.4 FunzioneD

(non usata)

2 Codice completo

```

1 //bool DEBUG =true;
const int inPulsante = 7;
3 const int LED1 = 2;
const int LED2 = 3;
5 const int LED3 = 4;
const int LED4 = 5;

```

```

7 const int RELE = 6;
bool pStatoPrec = LOW;
9 unsigned long int contaPremuto = 0;
unsigned long int contaRilasciato = 0;
11 //
// 0==|=====|=====|=====|=====|====.....
13 // |   |       |               |<- pausa tra parole
// |   |       |               >|<- linea
15 // |   |       >|<- punto
// |   >|<- disturbo
17 // >|<- 0
//
19 int tempoDisturbo = 2; // quarti di decimi di secondo ("disturbo")
int tempoPunto = 10;    // quarti di decimi di secondo ("punto")
21 int tempoLinea = 40;   // quarti di decimi di secondo ("linea")
int tempoLungo = 160;   // quarti di decimi di secondo ("pausa tra parole")
23 int giaPassato;
int step0raA = 7200;     //secondi in 2 ore
25 int step0raB = 14400;  //secondi in 4 ore
int step0raC = 21600;    //secondi in 6 ore
27 int step0raD = 28800;  //secondi in 8 ore
int tempoAspegne=0;
29 bool conteggioAttivo = false;
int dutyCycle = 0;
31 unsigned long int Pa;
unsigned long int Pb;
33 int oreProgrammate;
int segno;
35 //=====
// setup
37 //=====
void setup() {
39     pinMode(inPulsante,INPUT);
//
41     pinMode(LED1,OUTPUT);
digitalWrite(LED1,LOW);
43     //
pinMode(LED2,OUTPUT);
45     digitalWrite(LED2,LOW);
//
47     pinMode(LED3,OUTPUT);
digitalWrite(LED3,LOW);
49     //
pinMode(LED4,OUTPUT);
51     digitalWrite(LED4,LOW);
//
53     pinMode(RELE,OUTPUT);
digitalWrite(RELE,LOW);
55     // solo per test
// Serial.begin(115200);
57     //
// LAMPEGGIO DEL PIN 13
// poi lo spengo
59     //
pinMode(13,OUTPUT);
61     digitalWrite(13,LOW);
63     delay(500);
digitalWrite(13,HIGH);
65     delay(500);
digitalWrite(13,LOW);
67 }
//=====
69 // loop
//=====
void loop() {
71     unsigned long int Qa;
73     unsigned long int Qb;

```

```

75 int DIFFa;
76 int DIFFb;
77 int Xa;
78 int Xb;
79 //
80 dutyCycle += 1;
81 if (dutyCycle > 9){
82     dutyCycle = 0;
83 }
84 if (dutyCycle > 0){
85     Qa=millis();
86     if (Qa >= Pa){
87         DIFFa=Qa-Pa;
88         Xa = DIFFa - 25;
89         if (Xa >= 0){
90             Pa = Qa;
91             //-----
92             // da qui passa ogni 0.025 Sec
93             // quarto di decimo di secondo
94             //-----
95             if (digitalRead(inPulsante)){
96                 chkPremuto();
97             } else {
98                 chkRilasciato();
99             }
100         } else {
101             Pa = Qa - Xa;
102         }
103     } else {
104         Qb=millis();
105         if (Qb >= Pb){
106             DIFFb=Qb-Pb;
107             Xb = DIFFb - 1000;
108             if (Xb >= 0){
109                 Pb = Qb - Xb;
110                 //-----
111                 // da qui passa ogni secondo
112                 //-----
113                 chkTempo();
114                 setLEDs();
115             }
116         } else {
117             Pb = Qb;
118         }
119     }
120 }
121 void chkTempo(){
122     //=====
123     // se il conteggio e attivo
124     // imposta le ore programmate
125     // e lo stato del rele
126     // sulla base del tempoAlloSpegnimento
127     //=====
128     if (conteggioAttivo){
129         tempoAspegne -= 1;
130         if (tempoAspegne == 0){
131             oreProgrammate = 0;
132             digitalWrite(RELE,LOW);
133             //-----
134             // spegne il conteggio del
135             // tempoAlloSpegnimento
136             //-----
137             conteggioAttivo = false;
138         } else {
139             if (tempoAspegne > stepOraD){
140                 // non dovrebbe mai verificarsi

```

```

141 tempoAspegne = step0raD - 1;
    } else {
143     if (tempoAspegne > step0raC){
        oreProgrammate = 4;
145         digitalWrite(RELE,HIGH);
    } else {
147         if (tempoAspegne > step0raB){
            oreProgrammate = 3;
149             digitalWrite(RELE,HIGH);
        } else {
151             if (tempoAspegne > step0raA){
                oreProgrammate = 2;
153                 digitalWrite(RELE,HIGH);
            } else {
155                 oreProgrammate = 1;
                digitalWrite(RELE,HIGH);
157             }
        }
    }
159 }
161 }
163 }
//=====
165 // impostazioni led in base alle
// ore programmate
167 //=====
void setLEDs(){
169     if (oreProgrammate == 0){
        digitalWrite(LED1,LOW);
171         digitalWrite(LED2,LOW);
        digitalWrite(LED3,LOW);
173         digitalWrite(LED4,LOW);
    } else {
175         if (oreProgrammate == 1){
            digitalWrite(LED1,HIGH);
177             digitalWrite(LED2,LOW);
            digitalWrite(LED3,LOW);
179             digitalWrite(LED4,LOW);
        } else {
181             if (oreProgrammate == 2){
                digitalWrite(LED1,HIGH);
183                 digitalWrite(LED2,HIGH);
                digitalWrite(LED3,LOW);
185                 digitalWrite(LED4,LOW);
            } else {
187                 if (oreProgrammate == 3){
                    digitalWrite(LED1,HIGH);
189                     digitalWrite(LED2,HIGH);
                    digitalWrite(LED3,HIGH);
191                     digitalWrite(LED4,LOW);
                } else {
193                     digitalWrite(LED1,HIGH);
                    digitalWrite(LED2,HIGH);
195                     digitalWrite(LED3,HIGH);
                    digitalWrite(LED4,HIGH);
197                 }
            }
        }
199 }
201 }
//=====
203 // pulsante rilasciato
//=====
205 void chkRilasciato(){
    if (pStatoPrec == HIGH){
207         pStatoPrec = LOW;

```



```

    contaRilasciato =0;
209 } else {
    contaRilasciato +=1;
211 if (contaRilasciato > tempoLungo){
    if (giaPassato != 4){
213     giaPassato = 4;
        // funzione D
215     }
    } else {
217     if (contaRilasciato > tempoLinea){
        if (giaPassato != 3){
219     giaPassato = 3;
            // funzione C
221     //-----
        // attiva il conteggio del
223     // tempoAlloSpegnimento
        //-----
225     if (oreProgrammate > 0){
        conteggioAttivo = true;
227     }
        }
229     } else {
        if (contaRilasciato > tempoPunto){
231     if (giaPassato != 2){
        giaPassato = 2;
233     // funzione B
        }
        } else {
235     if (contaRilasciato > tempoDisturbo){
        if (giaPassato != 1){
237     giaPassato = 1;
            // funzione A
239     programmaLeOre();
            impostaTempo();
241     }
        } else {
243     if (giaPassato != 0){
        giaPassato = 0;
245     }
        }
247     }
        }
249     }
    }
251 }
}
253 //=====
// pulsante premuto
255 //=====
void chkPremuto(){
257 if (pStatoPrec == LOW){
    pStatoPrec = HIGH;
259 contaPremuto =0;
} else {
    contaPremuto +=1;
261 if (contaPremuto > tempoLungo){
    if (segno != 4){
263     segno = 4;
    }
265 } else {
    if (contaPremuto > tempoLinea){
        if (segno != 3){
267     segno = 3;
        }
269 } else {
        if (contaPremuto > tempoPunto){
            if (segno != 2){
271     segno = 2;
273

```

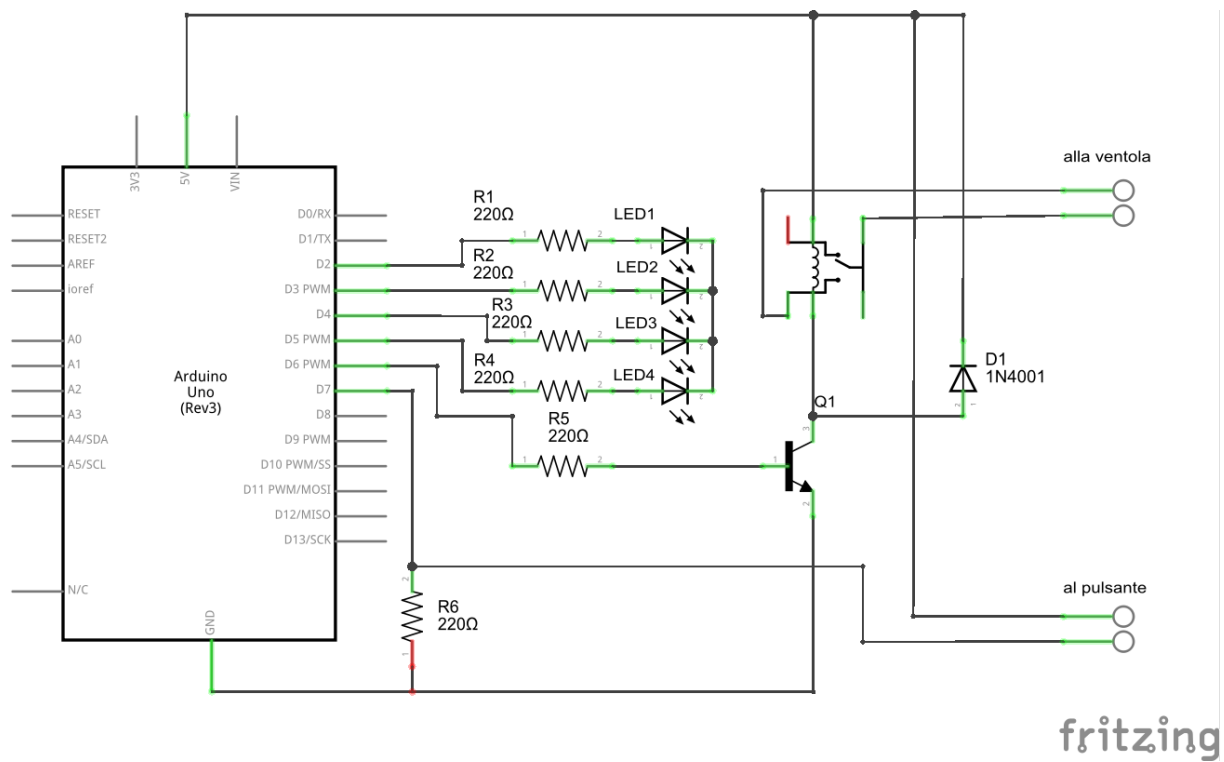
```

275     }
276     } else {
277         if (contaPremuto > tempoDisturbo){
278             if (segno != 1){
279                 segno = 1;
280             }
281         } else {
282             if (segno != 0){
283                 segno = 0;
284             }
285         }
286     }
287 }
288 }
289 }
290 }
291 //=====
292 // programma le ore
293 //=====
294 void programmaLeOre(){
295     if (segno == 1){
296         //=====
297         // tempo corto (punto)
298         //=====
299         oreProgrammate += 1;
300         if (oreProgrammate > 4){
301             oreProgrammate = 0;
302         }
303     } else {
304         if (segno > 1 ){
305             //=====
306             // tempi piu lunghi (linea o tempo lungo)
307             //=====
308             if (oreProgrammate > 0){
309                 oreProgrammate = 0; // spegne se acceso
310             } else {
311                 oreProgrammate = 3; // accende se spento
312             }
313         }
314     }
315 }
316 //=====
317 // impostazioni tempo
318 //=====
319 void impostaTempo(){
320     switch (oreProgrammate){
321         case 0:
322             tempoAspegne = 1; // fine
323             break;
324         case 1:
325             tempoAspegne = step0raA;
326             break;
327         case 2:
328             tempoAspegne = step0raB;
329             break;
330         case 3:
331             tempoAspegne = step0raC;
332             break;
333         case 4:
334             tempoAspegne = step0raD;
335             break;
336     }
337 }

```

code/TimerVentilatore4/TimerVentilatore4.ino

3 Schema elettrico



4 PCB

