

Blue Tester

Premessa

Sul numero di marzo 2015 di PC Professionale esce un articolo che mi svela la nascente attenzione sulla la luce blu emessa dai LED (quindi potenzialmente da tutti gli schermi moderni di TV, tablet, smartphone, ecc) perchè pare che le quantità innaturalmente emesse da questi dispositivi possano creare danni agli occhi e non solo. La luce blu nell'illuminazione a LED è fondamentale per poter creare la luce bianca, e riuscire a realizzare LED su questa lunghezza d'onda è stata una sfida vinta solo in anni recenti, cosa che ha pure fruttato il Nobel ai suoi autori proprio pochi mesi prima, nel 2014 (http://www.lescienze.it/news/2014/10/07/news/nobel_fisica_2014-2321542/).

C'è del vero o si tratta dell'ennesima teoria complottista e/o fake virale? Di sicuro la "novità" è stata ben accolta dal settore commerciale: in breve tempo i produttori di ottica hanno cominciato a proporre lenti anti luce blu, generalmente indicandole per chi sta molte ore davanti al PC. Ma si sa, pur di far soldi i negozianti vendono di tutto, dalle armi agli omeopatici, quindi l'esistenza di tali prodotti di per sé non significherebbe nulla.

La certezza che qualcosa realmente di anomalo c'è, però, la si ritrova su PubMed -l'indice mondiale delle pubblicazioni scientifiche-, dove non è difficile trovare studi sull'argomento, peraltro molto recenti e piuttosto inquietanti come questo "[Retinal damage induced by commercial light emitting diodes \(LEDs\)](#)" oppure questo "[White light-emitting diodes \(LEDs\) at domestic lighting levels and retinal injury in a rat model](#)" che conclude con un preciso avvertimento: "Thus, we suggest a precautionary approach with regard to the use of blue-rich "white" LEDs for general lighting".

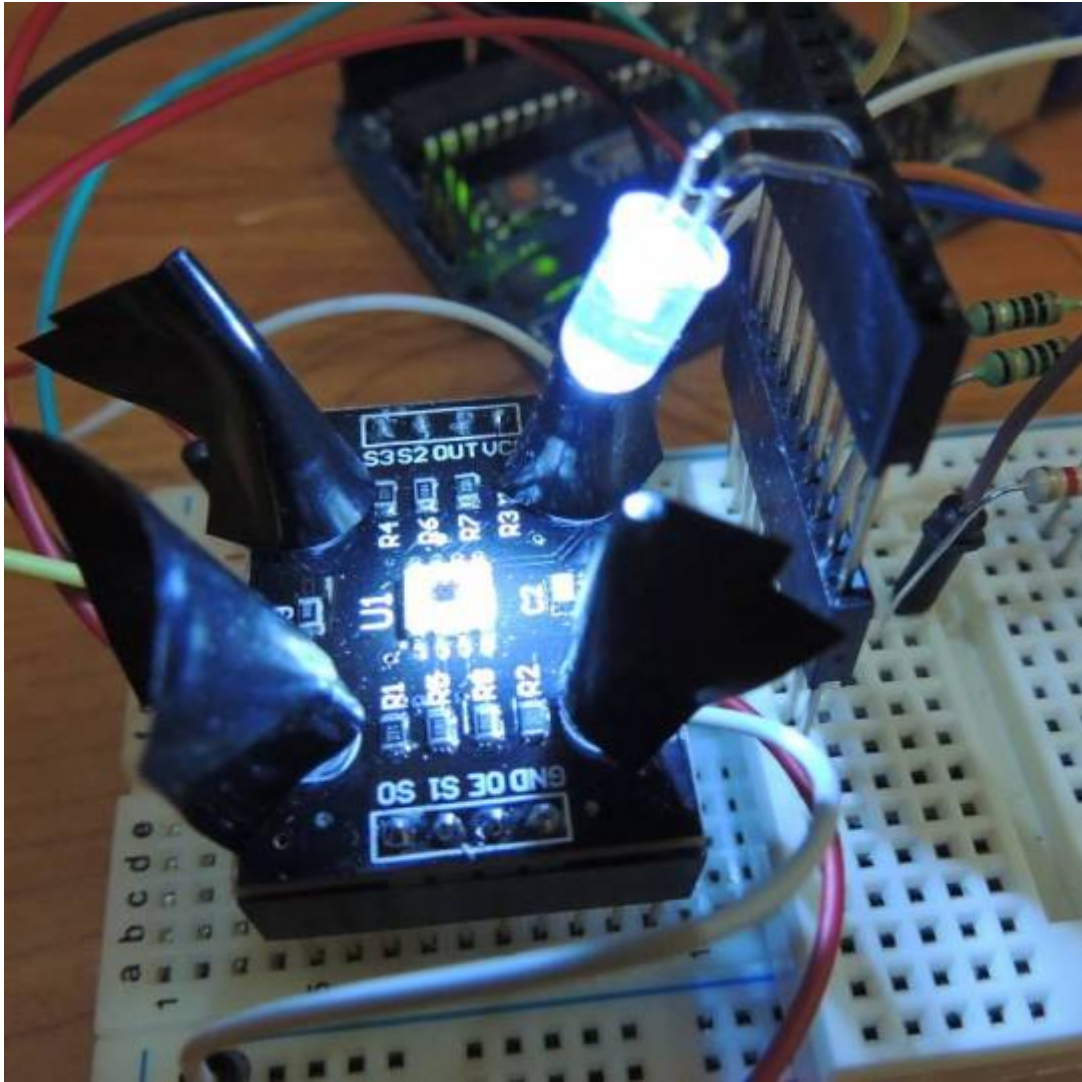
La soluzione

Al momento pare che i produttori di schermi stiano studiando nuove soluzioni circa l'illuminazione dei pannelli, con led in grado di emettere meno luce blu pur mantenendo una buona luminosità e contrasto. Nel frattempo l'unica soluzione appare quella di utilizzare i sopraccitati occhiali con trattamenti filtranti, e per quanto riguarda l'illuminazione casalinga optare per lampade "calde" (che infatti non a caso sono più simili alle vecchie lampadine ad incandescenza..) invece che "fredde" in quanto quelle fredde per ottenere l'effetto bianco puro emettono molta più luce blu.

Il blu tester

Nonostante gli studi letti rimango un po' scettico sulla cosa, e decido quindi di costruirmi un tester di luce blu per fare qualche semplice verifica casalinga (oltre che verificare l'efficacia delle eventuali soluzioni). Decido quindi di interfacciare un sensore TSC3200 con Arduino, di cui si trovano già in rete schemi e programmi demo. Non ho visto utilizzi molto intelligenti di questo sensore, probabilmente il mio tester potrebbe candidarsi a dargli finalmente uno scopo serio.

Trovo qui il datasheet: <http://www.mouser.com/catalog/specsheets/TCS3200-E11.pdf>



I programmi trovati in rete mi sembrano tutti uguali, e si basano sull'utilizzo del timer e degli interrupt, ma sono molto scarsi circa la manipolazione di dati letti. Decido quindi di farmelo da zero, in modo meno elegante ma più semplice. Creo una sola funzione che restituisce la frequenza in kHz campionando direttamente il pin di ingresso. In base al tempo trascorso per rilevare 1000 impulsi, viene calcolato il valore: i risultati sono molto precisi (confrontati in tempo reale con il mio tester che fa anche da frequenzimetro). Inizialmente avevo usato l'istruzione Arduino `pulseIn`, ma funzionava solo entro certi range, poi nelle reference ho letto che in effetti l'istruzione non funzionava per impulsi inferiori ai 10 microsecondi quindi impossibile fare letture sopra i 50Khz, e nella realtà si fermava già a poco più di 30. Anche l'istruzione `micros` ha dei grossi limiti (inattendibile oltre 16000) quindi alla fine ho riscritto tutto con il `digitalRead`, il `millis`, e impostando il prescaling sul sensore al 20% in modo che avere una frequenza nel range 1-100Khz. Questa la prima versione funzionante:

```
void setup() {  
  
#define S0      6  
#define S1      5  
#define S2      4  
#define S3      3  
#define OE      7  
#define OUT     2  
  
pinMode(S0, OUTPUT);
```

```
pinMode(S1, OUTPUT);
pinMode(S2, OUTPUT);
pinMode(S3, OUTPUT);
pinMode(OE, OUTPUT);
pinMode(OUT, INPUT);

// keep sensor off
digitalWrite(OE, HIGH);

// set 20% frequency scaling
digitalWrite(S0, HIGH);
digitalWrite(S1, LOW);

// activate serial output
Serial.begin(57600);

}

float Read_Freq() {

int start=millis();
boolean readpin = HIGH;
for (int count=0; count<1000; count++) {
    while (readpin == LOW) {
        readpin=digitalRead(OUT);
    }
    while (readpin == HIGH) {
        readpin=digitalRead(OUT);
    }
}

int stop=millis();
int duration=stop-start;
float period=duration/1000.0;
float freq=1.0/period;

return freq;
}

void loop() {

//activate sensor
digitalWrite(OE, LOW);

float freq;
// read clear light
digitalWrite(S2, HIGH);
digitalWrite(S3, LOW);
freq = Read_Freq();
Serial.print("Full light: ");Serial.print(freq);Serial.println("kHz");
```

```
// read red light
digitalWrite(S2, LOW);
digitalWrite(S3, LOW);
freq = Read_Freq();
Serial.print("Red light:  ");Serial.print(freq);Serial.println("kHz");

// read green light
digitalWrite(S2, HIGH);
digitalWrite(S3, HIGH);
freq = Read_Freq();
Serial.print("Green light: ");Serial.print(freq);Serial.println("kHz");

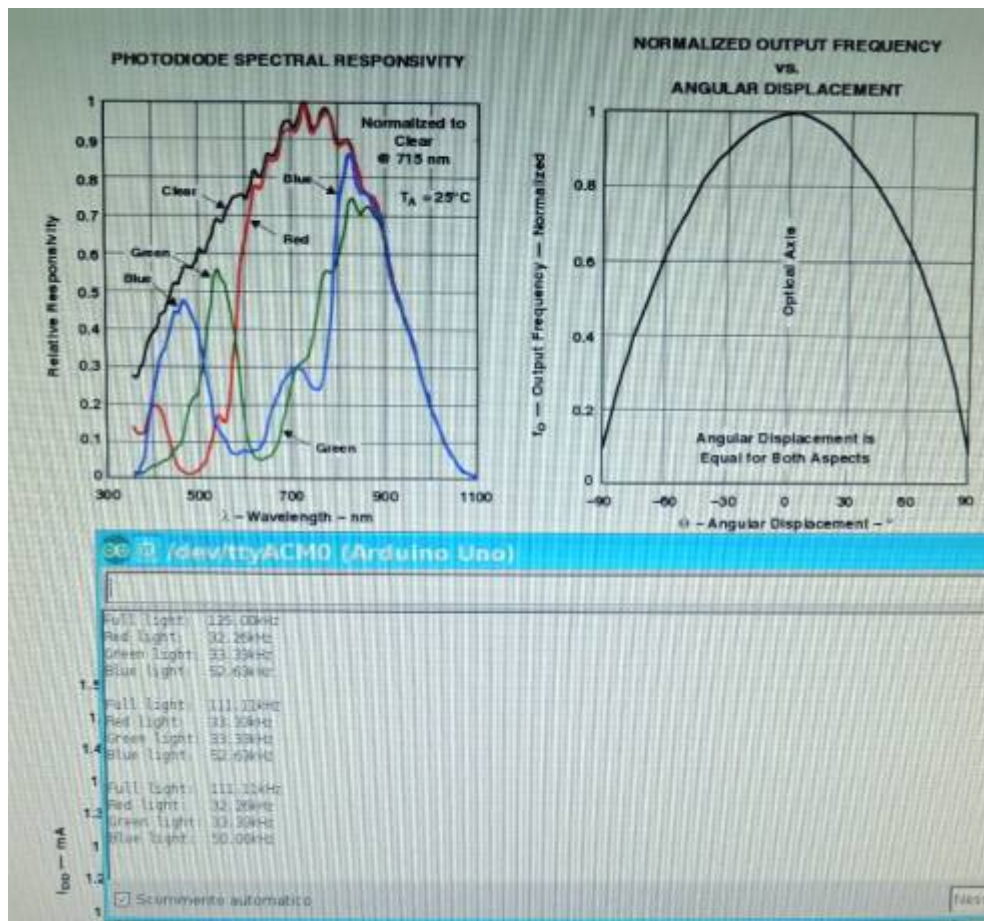
// read blue light
digitalWrite(S2, LOW);
digitalWrite(S3, HIGH);
freq = Read_Freq();
Serial.print("Blue light:  ");Serial.print(freq);Serial.println("kHz");

// keep sensor off
digitalWrite(OE, HIGH);

Serial.println(" ");
delay(2000);

}
```

Il sensore in mio possesso è su una schedina con 4 led bianchi che si accendono non appena si alimenta il circuito. Questi moduli infatti in ambiente maker sono usati per rilevare i colori, quindi i led servono a illuminare un soggetto non luminoso e il sensore legge la luce di ritorno. A me interessa invece testare una fonte luminosa, quindi i 4 led sono d'impiccio. Usando il circuito mostrato nell'immagine precedente, usando un semplice led bianco (e mascherando con il nastro i 4 led presenti su scheda) ho fatto i primi test visualizzando i dati sul monitor seriale dell'IDE di Arduino.



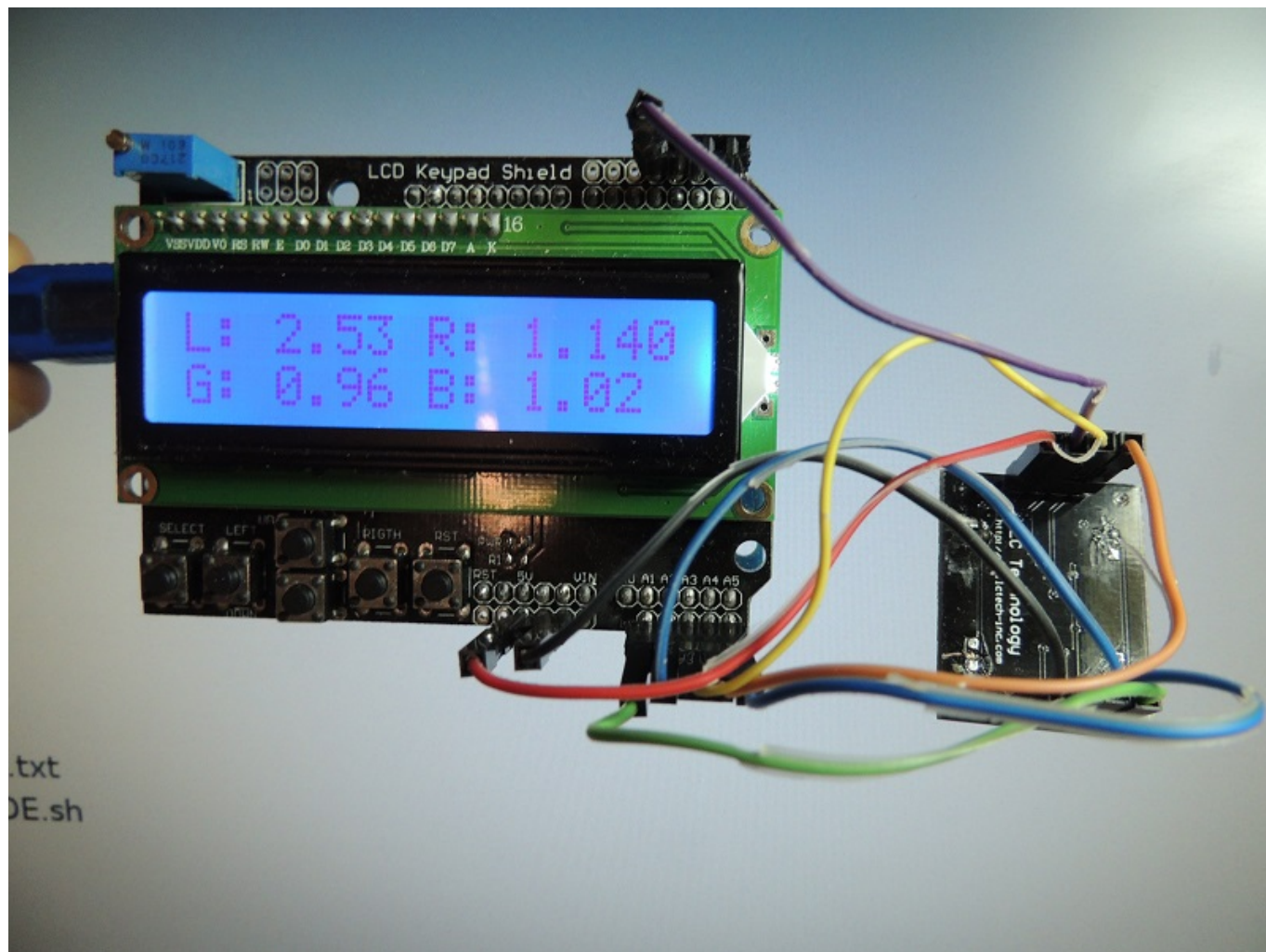
Le prime letture usando il led bianco puntato sul sensore mostrano come la frequenza prodotta dal sensore sia proporzionale all'intensità luminosa (o alla distanza, mantenendo costante l'intensità, come ho fatto io). Ecco qualche serie:

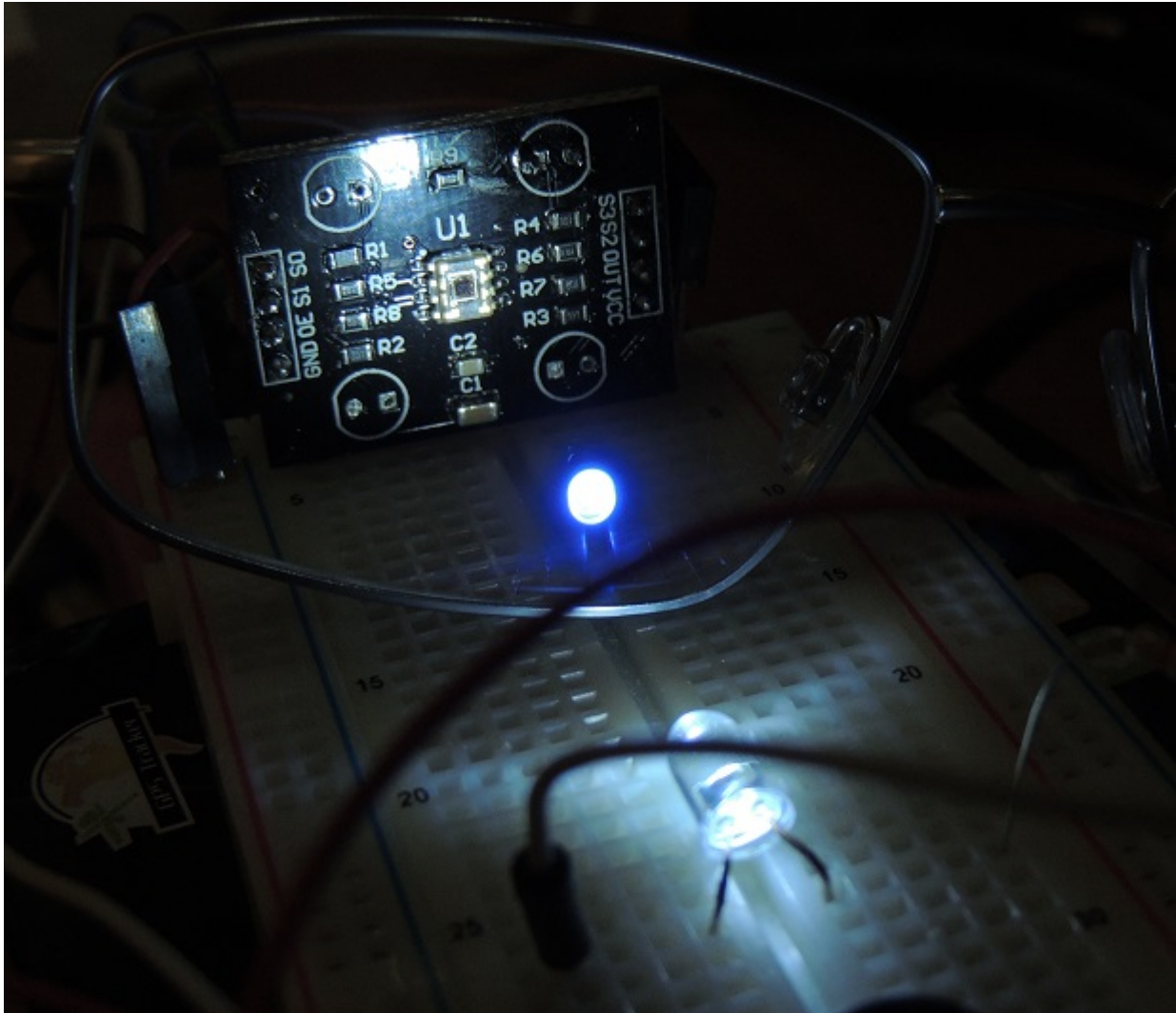
Light	Freq1	%1	Freq2	%2	Freq3	%3
Full	100.00	100	35.71	100	12.20	100
Red	32.26	32.3	11.90	33.3	4.18	34.3
Green	34.48	34.5	10.99	33.3	3.56	29.2
Blu	52.63	52.6	15.87	44.4	5.10	41,8

Si nota che in tutti i casi la lettura della componente blu è maggiore di quelle rossa e verde. In particolare nel primo caso, con il led più vicino al sensore, il blu è intorno al 50% mentre rosse e verde poco più del 30%. Da notare che la somma delle 3 componenti non dà il 100%, questo perchè i 3 colori hanno una risposta gaussiana con picco su determinate lunghezze d'onda (come si vede in figura precedente, tratta dal datasheet).

Test con sensore senza led

Light	TV LCD	%	TV LED	%	L1 Fredda	%	Neon	%	L2 Calda	%	L3 Fredda	%	THL W100	%	Brondi
Full	2.00	100	12.50	100	40.00	100	33.33	100	40.00	100	40.00	100	10.53		10.00
Red	0.84	42	5.00	40	20.00	50	16.80	50	28.50	71	18.10	45	3.51		3.51
Green	0.70	35	4.55	36	12.50	31	11.80	35	11.11	28	14.30	36	3.23		3.39
Blue	0.80	40	5.26	42	12.50	31	14.30	44	9.10	22	15.30	38	3.51		3.70





From:
<http://www.gaggini.net/notes/> - **Paolo Gaggini's Notes**

Permanent link:
<http://www.gaggini.net/notes/doku.php/bluetester1>

Last update: **16/01 /2021 10:19**

