



# COLIBRI: DRIVER DI POTENZA PER RGBW

Permette di pilotare i nuovissimi diodi luminosi a 4 colori, variandone agevolmente la luminosità e la colorazione.

di MASSIMO DEL FEDELE

**D**i recente sono apparsi sul mercato alcuni LED multicolore di potenza che, a differenza dei più comuni RGB, incorporano in più un LED bianco: vengono chiamati RGBW (Red, Green, Blu, White) proprio per questo. Sono stati creati perché sebbene per ottenere una luce bianca sia sufficiente accendere insieme LED rossi, verdi e blu (la proporzione tra la luminosità di rosso, verde e blu determina la tonalità del bianco, che può essere calda o fredda) l'aggiunta di un LED bianco permette di ottenere dei bianchi molto più "puliti" rispetto a quanto si può ottenere dalla miscelazione dei tre colori base; è inoltre possibile utilizzare tutti e quattro i insieme per ottenere una potenza luminosa maggiore.

I LED di tipo RGBW sono reperibili tipicamente in potenze come  $4 \times 1$  watt (4 watt totali) e  $4 \times 3$  watt (12 watt totali). C'è da dire che le potenze reali sono effettivamente inferiori rispetto a quanto dichiarato dai produttori, ed è sempre meglio guardare la massima corrente di pilotaggio piuttosto che la potenza, se non altro perché i LED

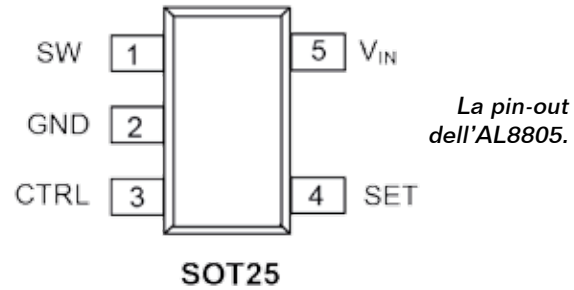
si comandano in corrente e perché i costruttori forniscono i grafici della luminosità in funzione della corrente diretta ( $I_F$ ).

Vista la disponibilità di questi nuovi LED a quattro colori, abbiamo progettato un driver di potenza per alimentarli a partire da segnali PWM di controllo che possono essere forniti, ad esempio, da un microcontrollore, ma anche da una scheda Arduino. In queste pagine descriveremo il progetto, proponendo altresì uno sketch per la gestione da Fishino Uno, che è la nostra scheda Arduino Uno compatibile.

Prima di analizzare lo schema elettrico del circuito, torniamo un attimo sulle considerazioni fatte pocanzi circa la potenza e la corrente, per spiegare come mai ai fini della valutazione di un LED RGBW si debba prendere in considerazione la corrente diretta. Prendiamo un LED quadricoloro da 4 watt, potenza distribuita tra quattro LED che usualmente hanno una corrente di pilotaggio di 300 mA (corrente diretta); siccome la caduta di tensione sui LED varia in base al colore, abbiamo

## CARATTERISTICHE TECNICHE

- Tensione di alimentazione: 6÷36 Vcc
- Funzionamento switching
- Corrente massima in uscita: 1 A per canale
- Controllo di luminosità (dimming):  
sia analogico che digitale
- Numero di canali per modulo: 4
- Dimensioni: 27,6 x 37,3 mm



potenze differenti per i vari LED, ossia:

Proso =  $(2,2+2,5V) \cdot 300 \text{ mA} = 0,66+0,75 \text{ W}$   
 Pverde =  $(2,5+2,8v) \cdot 300 \text{ mA} = 0,75+0,84 \text{ W}$   
 Pblu, bianco =  $(3,2v+3,4v) \cdot 300 \text{ mA} = 0,96+1,02 \text{ W}$

dove tra parentesi sono indicate le cadute di tensione minima e massima sui diodi. Considerando i valori massimi, abbiamo una potenza totale pari a:

$$P_{tot} = 0,75 + 0,84 + 2,04 \text{ W} = 3,63 \text{ W}$$

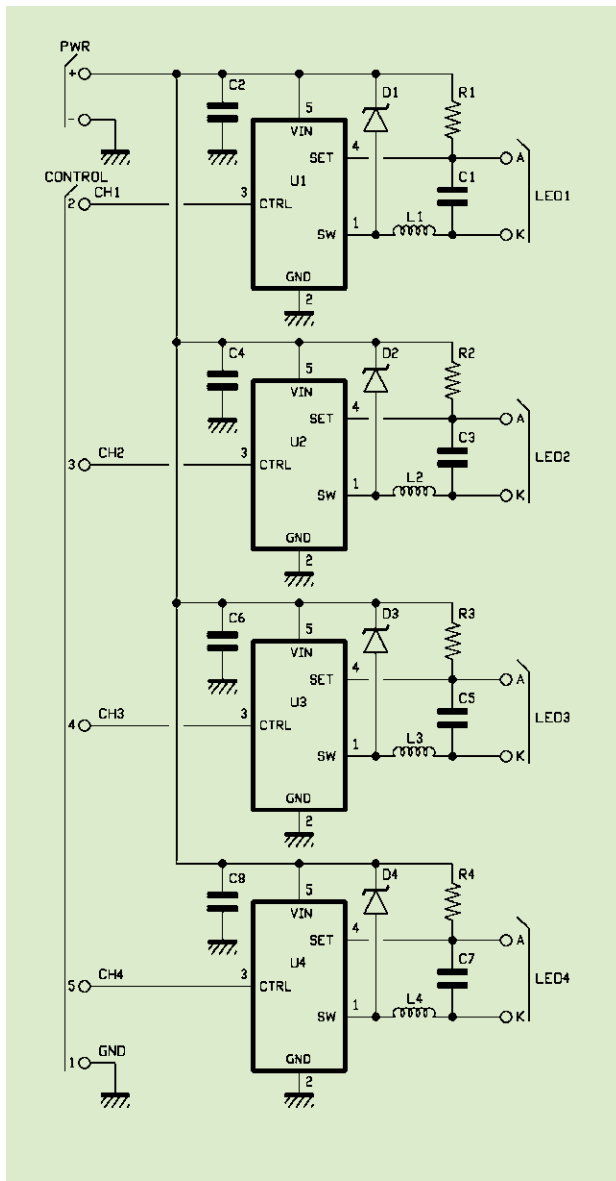
quindi abbastanza vicina a quanto dichiarato. Per i LED da 12 watt le cose cambiano un po', visto che la corrente di funzionamento dichiarata ammonta a 700 mA; ora le potenze diventano 1,75W per il LED rosso, 1,96W per il LED verde e 2,38W per bianco e blu. Con questi valori, la potenza massima complessiva diviene:

$$P_{tot} = 8,47 \text{ W}$$

Questo valore è ben inferiore a quanto dichiarato, anche se rimane una potenza notevole. La potenza dichiarata per i LED RGBW è quindi da intendersi quella massima pilotando ciascun LED alla massima potenza, perciò con correnti di valore differenti.

### IL PROGETTO

Come ben noto, i LED vanno pilotati a corrente costante e non a tensione costante, per ottenerne il massimo rendimento ed evitare di "bruciarli" (rammentate che la variazione della corrente nei diodi è esponenziale, quindi piccole variazioni di tensione determinano ampie variazioni di corrente). La tecnica principe per modularne la corrente e quindi la luminosità consiste nell'alimentarli con un segnale PWM, vale a dire una tensione di forma d'onda rettangolare di cui si modula il duty-cycle. Questa tecnica è prediletta perché consente di ottenere rendimenti elevatissimi, dato che per fornire l'alimentazione ai LED -sotto forma di impulsi- è sufficiente utilizzare transistor funzionanti in modo on-off, beneficiando delle minime resistenze serie dei moderni MOSFET e ottenendo così perdite di potenza irrisorie. Se i LED venissero pilotati da regolatori di tensione lineari, quindi con transistor funzionanti da amplificatori, il rendimento scenderebbe tanto più



[schema ELETTRICO]

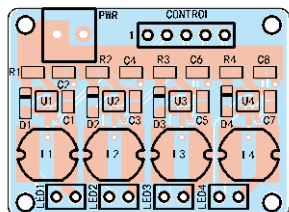
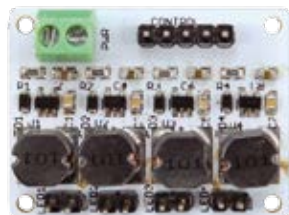
# [piano di MONTAGGIO]

## Elenco Componenti:

- R1 ÷ R4: 0,3 ohm (0805)
- C1: 1 µF ceramico (0805)
- C2: 2,2 µF ceramico (0805)
- C3: 1 µF ceramico (0805)
- C4: 2,2 µF ceramico (0805)
- C5: 1 µF ceramico (0805)
- C6: 2,2 µF ceramico (0805)
- C7: 1 µF ceramico (0805)
- C8: 2,2 µF ceramico (0805)
- D1 ÷ D4: BAT20JFILM
- U1 ÷ U4: AL8805W5-7
- L1 ÷ L4: Induttanza 33 µH

### Varie:

- Strip maschio 2 vie (4 pz.)
- Strip maschio 5 vie (1 pz.)
- Plug alimentazione SMD
- Circuito stampato S1242



quanto la tensione fornita ai diodi sarebbe più bassa di quella di alimentazione, dato che aumenterebbe la caduta di tensione sui transistor a parità di corrente e con essa la potenza dissipata. Alla perdita di rendimento si affiancherebbe quindi la necessità di adottare dei dissipatori, con l'inevitabile incremento di costi e dimensioni dei circuiti driver.

In questo articolo presentiamo quindi un driver per LED a 4 canali con ottime caratteristiche e in grado di pilotare elementi con un assorbimento massimo di 1 Ampere per canale; il nostro circuito è predisposto per ricevere segnali di comando PWM (o anche digitali, a livello logico TTL) ed è in grado di accettare una tensione in ingresso da 6 a 36 volt.

L'elevata tensione di alimentazione sopportabile permette inoltre di pilotare più LED in serie, così da realizzare lampade e faretto; per esempio, con una tensione di 36 volt è possibile pilotare un massimo di  $36V/3,4V = 10$  LED bianchi in serie per ogni canale.

## SCHEMA ELETTRICO

Come si nota, il driver è composto da quattro sezioni perfettamente identiche tra loro, quindi ne descriveremo una sola, fermo restando che quanto detto per essa varrà per le altre.

Il tutto ruota attorno ad un circuito integrato driver creato appositamente per il controllo di LED di potenza, IC1, che è un AL8805 prodotto dalla Diodes. Questo è in pratica un convertitore step-down (buck converter) realizzato in modo da fornire un'uscita a corrente costante, di valore regolabile, perché i LED si controllano in corrente, adeguando

la tensione fornita ai LED al fine di mantenere il valore di intensità corrispondente alla luminosità desiderata: più corrente viene richiesta, maggiore è la tensione fornita e viceversa.

All'interno dell'integrato, lo stadio di uscita che alimenta il LED è composto da MOSFET che fa capo al piedino SW (switch), mentre il piedino SET viene utilizzato per il controllo della corrente in uscita, impostata tramite la resistenza R1 secondo la formula seguente:

$$I_{LED} = \frac{V_{THD}}{R_1}$$

dove  $I$  è la corrente che si vuol far circolare nei LED e  $V_{THD}$  è una tensione di riferimento interna all'integrato, pari a 100 mV.

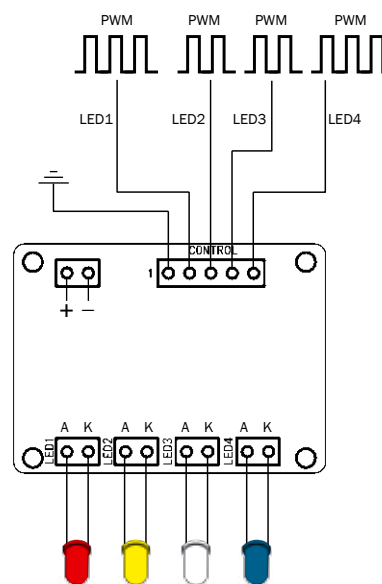
Quindi, volendo far circolare 300 mA nei LED, adatta ai LED da 4 watt di cui abbiamo parlato, occorre inserire una resistenza pari a:

$$R_1 = \frac{V_{THD}}{I_{LED}} = \frac{0.1V}{0.3A} = 0.33\ ohm$$

che è il valore da noi inserito nella scheda, arrotondandolo a 0,3 ohm. Per ottenere una corrente di 700 mA, adatta ai LED da 12 watt, occorre inserire una resistenza pari a:

$$R_1 = \frac{V_{THD}}{I_{LED}} = \frac{0.1V}{0.7A} = 0.14\ ohm$$

Questo valore è facilmente approssimabile tramite



Schema di collegamento del driver e comando tramite i segnali PWM.

## Listato 1 MAIN LOOP

```
void loop(void)
{
    // ciclo per luminosità del bianco crescente
    for(int w = 0; w < 256; w++)
    {
        // ciclo per la luminosità (VALUE) crescente
        for(int v = 100; v < 256; v++)
        {
            // ciclo per la tonalità (HUE)
            for(int h = 0; h < 256; h++)
            {
                showColor(h, 255, v, w);
                delay(del);
            }
        }

        // ciclo per la luminosità (VALUE) decrescente
        for(int v = 254; v >= 100; v--)
        {
            // ciclo per la tonalità (HUE)
            for(int h = 0; h < 256; h++)
            {
                showColor(h, 255, v, w);
                delay(del);
            }
        }
    }

    // ciclo per luminosità del bianco decrescente
    for(int w = 254; w >= 0; w--)
    {
        // ciclo per la luminosità (VALUE) crescente
        for(int v = 100; v < 256; v++)
        {
            // ciclo per la tonalità (HUE)
            for(int h = 0; h < 256; h++)
            {
                showColor(h, 255, v, w);
                delay(del);
            }
        }

        // ciclo per la luminosità (VALUE) decrescente
        for(int v = 254; v >= 100; v--)
        {
            // ciclo per la tonalità (HUE)
            for(int h = 0; h < 256; h++)
            {
                showColor(h, 255, v, w);
                delay(del);
            }
        }
    }
}
```

una seconda resistenza da 0,3 ohm saldata sopra quella presente sul PCB, in modo da realizzare un parallelo da 0,15 ohm totali. L'induttanza L1 e il diodo Schottky D1 costituiscono la rimanente parte del convertitore buck contenuto nell'integrato. Il converter lavora ad una frequenza massima di 1 MHz e consente quindi l'utilizzo di componenti di valori contenuti e piccole dimensioni, sia per l'induttanza che per il condensatore di filtro C1, che serve unicamente a ridurre il ripple in uscita (l'ondulazione della tensione continua fornita al LED) e ad attenuare eventuali interferenze prodotte dal convertitore, per quanto i LED a stretto rigore non ne abbiano bisogno.

Per ultimo, il condensatore C2 costituisce un'indispensabile filtro/disaccoppiamento in ingresso, in modo da far "vedere" all'integrato una tensione continua priva di disturbi ed evitare malfunzionamenti che possono derivare dalla presenza di spike. Parlando dell'impostazione della corrente in uscita effettuata tramite R1, abbiamo ommesso di dire che questa è da intendersi come la corrente massima, quando il chip è impostato per la massima luminosità. La corrente nel LED è in realtà modulabile tramite l'ingresso CTRL, il quale, grazie ad un paio di meccanismi interessanti, permette il controllo completo della luminosità sia tramite un valore di tensione analogico che con un segnale rettangolare in PWM.



## Listato 2 CONVERSIONE DA COLORE HSV A RGB

```
RgbColor HsvToRgb(HsvColor hsv) // conversione da colore HSV a colore RGB
{
    RgbColor rgb;
    unsigned char region, p, q, t;
    unsigned int h, s, v, remainder;
    if (hsv.s == 0)
    {
        rgb.r = hsv.v;
        rgb.g = hsv.v;
        rgb.b = hsv.v;
        return rgb;
    }
    h = hsv.h;
    s = hsv.s;
    v = hsv.v;
    region = h / 43;
    remainder = (h - (region * 43)) * 6;
    p = (v * (255 - s)) >> 8;
    q = (v * (255 - ((s * remainder) >> 8))) >> 8;
    t = (v * (255 - ((s * (255 - remainder)) >> 8))) >> 8;
    switch (region)
    {
        case 0:
            rgb.r = v;
            rgb.g = t;
            rgb.b = p;
            break;
        case 1:
            rgb.r = q;
            rgb.g = v;
            rgb.b = p;
            break;
        case 2:
            rgb.r = p;
            rgb.g = v;
            rgb.b = t;
            break;
        case 3:
            rgb.r = p;
            rgb.g = q;
            rgb.b = v;
            break;
        case 4:
            rgb.r = t;
            rgb.g = p;
            rgb.b = v;
            break;
        default:
            rgb.r = v;
            rgb.g = p;
            rgb.b = q;
            break;
    }
    return rgb;
}
```

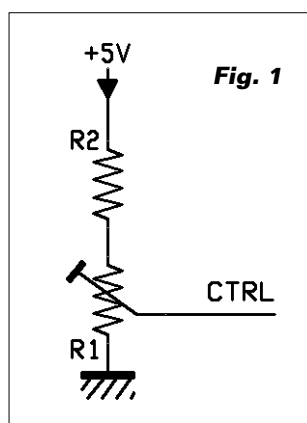
In pratica, il controllo si può effettuare così:

- quando al piedino CTRL è applicata una tensione continua inferiore a 0,4 volt, il convertitore si spegne e l'uscita va a zero volt;
- applicando a CTRL una tensione continua di valore compreso tra 0,4 e 2,5 volt, il convertitore varia la corrente in uscita da zero al massimo impostato tramite la R1, proporzionalmente alla tensione in ingresso;
- se la tensione continua applicata a CTRL supera i 2,5 volt, il convertitore mantiene l'uscita al massimo preimpostato tramite R1;
- applicando, invece, un segnale PWM sull'ingresso CTRL, con l'accortezza che il valore basso del

medesimo sia minore di 0,4 volt ed il segnale alto maggiore di 2,5 volt (ottenibile con logiche sia a 3,3 volt che a 5 V) l'uscita viene impostata secondo il duty-cycle del segnale stesso, con una risoluzione di 1.024 passi.

Risulta quindi semplicissimo realizzare un controllo analogico tramite un potenziometro, come mostrato in Fig. 1. Ruotando il potenziometro (siglato R1 nella predetta figura) è possibile variare la luminosità dal minimo al massimo, in maniera molto semplice e senza bisogno di circuiti di pilotaggio basati su microcontrollore; ciò permette di implementare la funzione dimmer con il minimo

hardware possibile. In questo schema semplificato non si tiene conto della soglia inferiore di 0,4 volt, per cui il potenziometro avrà una zona "morta" all'inizio nella quale il LED risulterà sempre spento. In alternativa è possibile inserire un'ulteriore resistenza tra il potenziometro e la massa in modo



da avere una tensione minima su CTRL pari a 0,4 volt. Con i valori nello schema è sufficiente inserire una resistenza di circa 1,5-1,8 kohm. Passando dall'analogico al digitale, con lo sketch del listato 2 possiamo ottenere tutta la gamma di colori sul LED, variabili con continuità.

La variazione non viene eseguita come di consueto nello spazio colore RGB (ovvero facendo variare i singoli colori in 3 cicli nidificati), cosa poco gradevole cromaticamente, ma lavorando nello spazio HSV (o HSL), ovvero **Hue-Saturation-Value, Tonalità-Saturazione-Luminosità**.

Per evitare una durata troppo elevata del ciclo è stata impostata la **Saturazione (Saturation)** al massimo valore (e quindi i colori saranno molto "carichi") e vengono fatte variare solo **Tonalità (Hue)** e **Luminosità (Value)**, con in più un ciclo esterno che aggiunge una componente bianca all'effetto.

Lo sketch è abbastanza semplice; tramite 3 cicli nidificati vengono fatti variare i valori del **Bianco (ciclo esterno)**, della **Luminosità (ciclo intermedio)** e della **Tonalità (ciclo interno)**.

I cicli aggiuntivi sono dovuti al fatto che si vogliono evitare "sbalzi" nei valori passando dal massimo (255) direttamente allo zero; i valori di **Bianco** e **Luminosità** vengono quindi fatti variare *da 0 a 255* e *da 255 a 0*, gradualmente. Il valore della **Tinta (Hue)**, trovandosi su uno spazio circolare, non ha bisogno di questo accorgimento.

La variabile 'del' controlla, come detto sopra, il ritardo nel ciclo più interno e quindi la velocità delle variazioni cromatiche.

Un'ultima nota: il controllo tramite Arduino prevede l'utilizzo delle uscite in PWM che sono piuttosto poche; con un Arduino UNO è possibile controllare solo un LED RGBW in questo modo, cosa piuttosto limitativa.

Per questo ed altri motivi, in un successivo articolo presenteremo uno shield d'espansione dotato di

16 canali PWM, controllabile tramite interfaccia I<sup>2</sup>C-Bus, che permetterà di utilizzare ben 4 driver in contemporanea, pilotando così quattro LED RGBW o ben 16 LED monocromatici contemporaneamente.

### REALIZZAZIONE PRATICA

Chiudiamo spiegando come costruire il driver per LED RGBW, che richiede un circuito stampato a doppia ramatura e fori metallizzati e che fa uso in buona parte di componenti SMD; per questa ragione, la sua costruzione richiede un minimo di manualità con le tecniche di montaggio superficiale e un po' di attrezzatura specifica che comprende un saldatore da non più di 20 watt di potenza, a punta fine, del filo di lega saldante da non più di 0,5 mm di diametro, della pasta fluxante, una lente d'ingrandimento e una pinzetta per posizionare i componenti sulla basetta.

Il montaggio si inizia con le resistenze e i condensatori in formato 0805 e prosegue con i diodi e gli integrati, quindi con le induttanze; per agevolare la saldatura ed evitare che la lega saldante cortocircuiti i contatti molto vicini degli AL8805, conviene spalmare della pasta fluxante sulle piazzole prima di saldare detti componenti. Il fluxante è comunque utile per tutti gli elementi del circuito. La morsettiera è a montaggio tradizionale, come i pjn-strip che utilizziamo per le connessioni ai segnali di controllo (o potenziometri di dimming) e ai LED. Completato il montaggio, date uno sguardo alla lente di ingrandimento per vedere se non ci sono cortocircuiti e nel caso rimuoveteli. Ora il driver è pronto per essere utilizzato, dato che non richiede alcuna operazione di taratura. ■



### per il MATERIALE

Tutti i componenti utilizzati in questo progetto sono si possono acquistare presso Futura Elettronica. Il master del circuito stampato può essere scaricato dal sito della rivista così come il listato completo. Il driver di potenza per RGBW (cod. COLIBRI) è disponibile a Euro 19,90. Il modulo LED RGBW 4x1W (cod. RGBWLED1) è in vendita a Euro 2,50, mentre il modulo LED RGBW 4x3W (cod. RGBWLED3) costa Euro 4,80.

Il materiale va richiesto a: Futura Elettronica,  
Via Adige 11, 21013 Gallarate (VA)  
Tel: 0331-799775 - Fax: 0331-792287  
<http://www.futurashop.it>