

1222·2022
800
ANNI



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Miglioramento di un sistema di acquisizione per la caratterizzazione di motori per droni

Amadi Fabio, Fanchin Giuseppe, Righetto Giacomo

Relatore: Dr. Davide Biadene

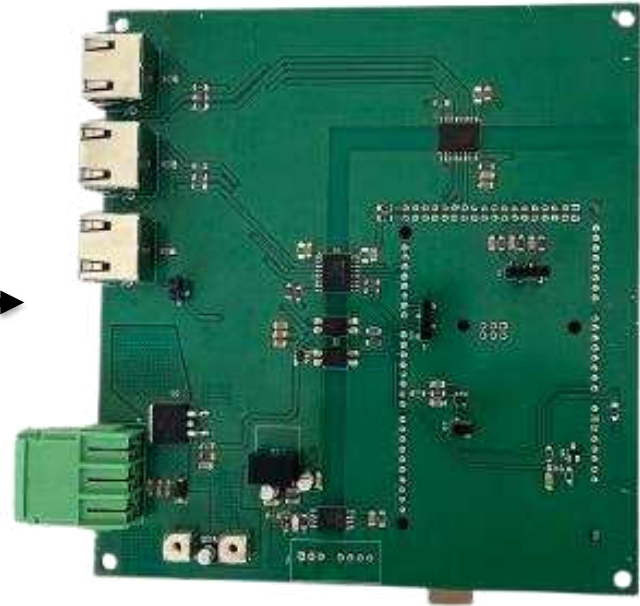
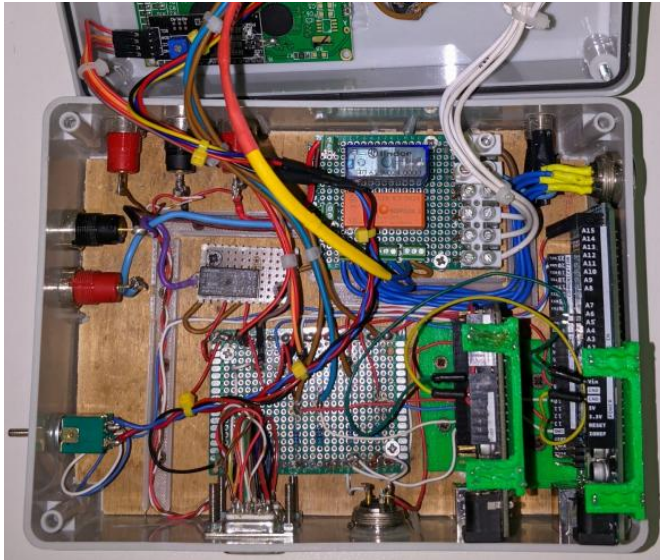
Co-relatori: Dr.ssa Giulia Michieletto, Dr. Massimiliano Bertoni

Department of Management and Engineering (DTG) – Vicenza

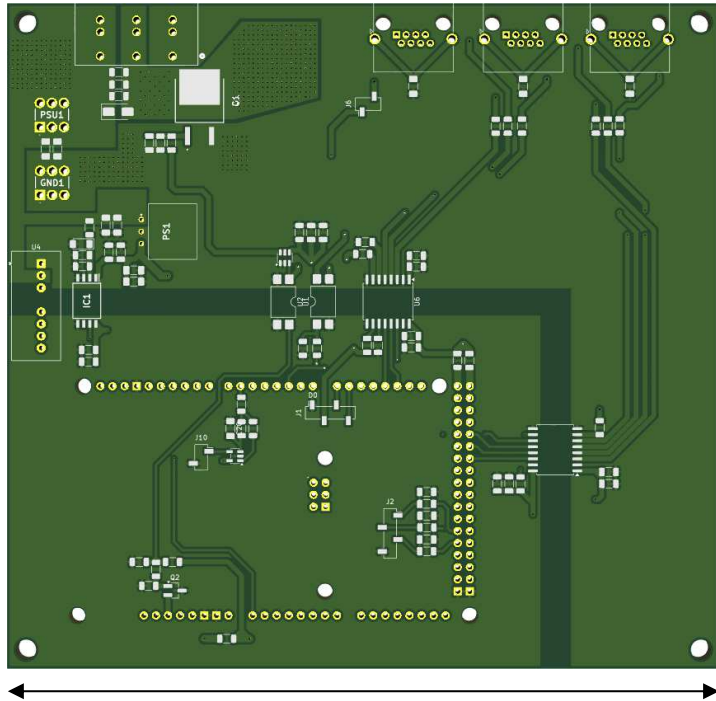


LA SCHEDA

Rinnovo e riprogettazione del sistema di controllo e sicurezza



La Scheda

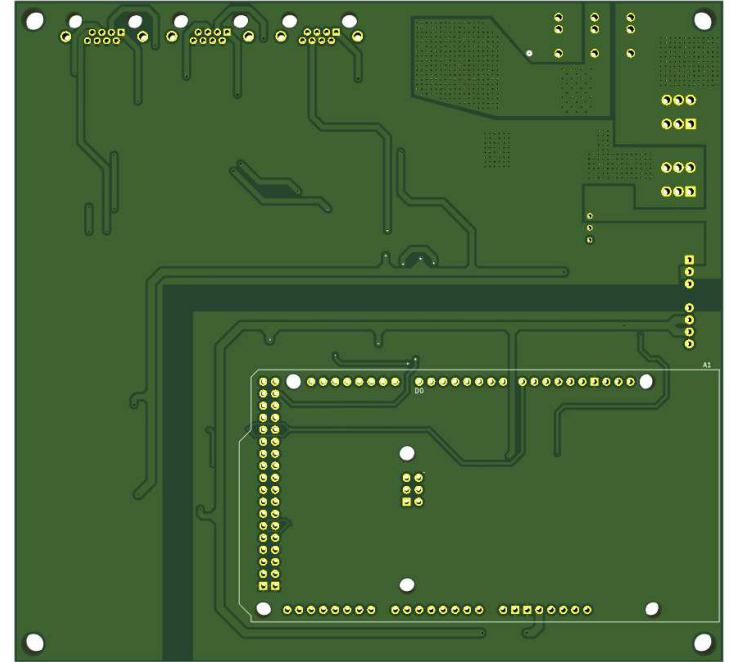


140 mm

140 mm
TOP

$2 \frac{oz}{ft^2}$

2 layer PCB

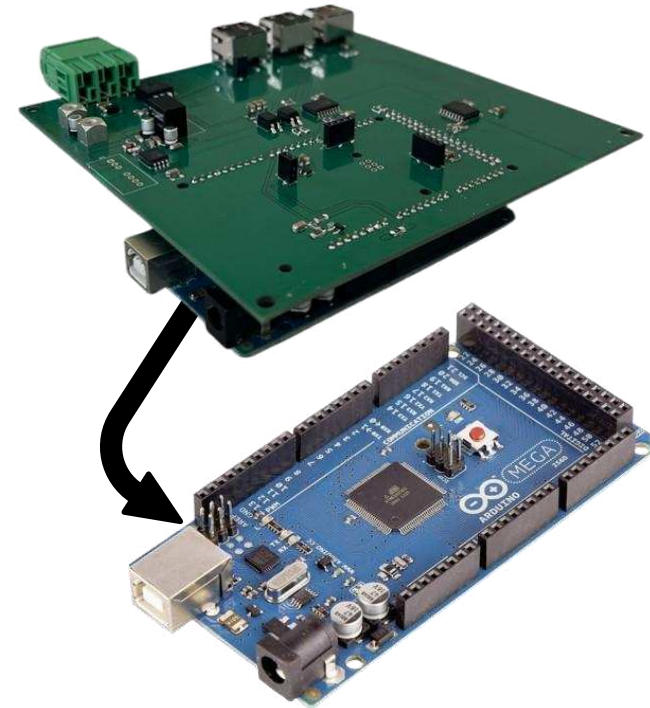
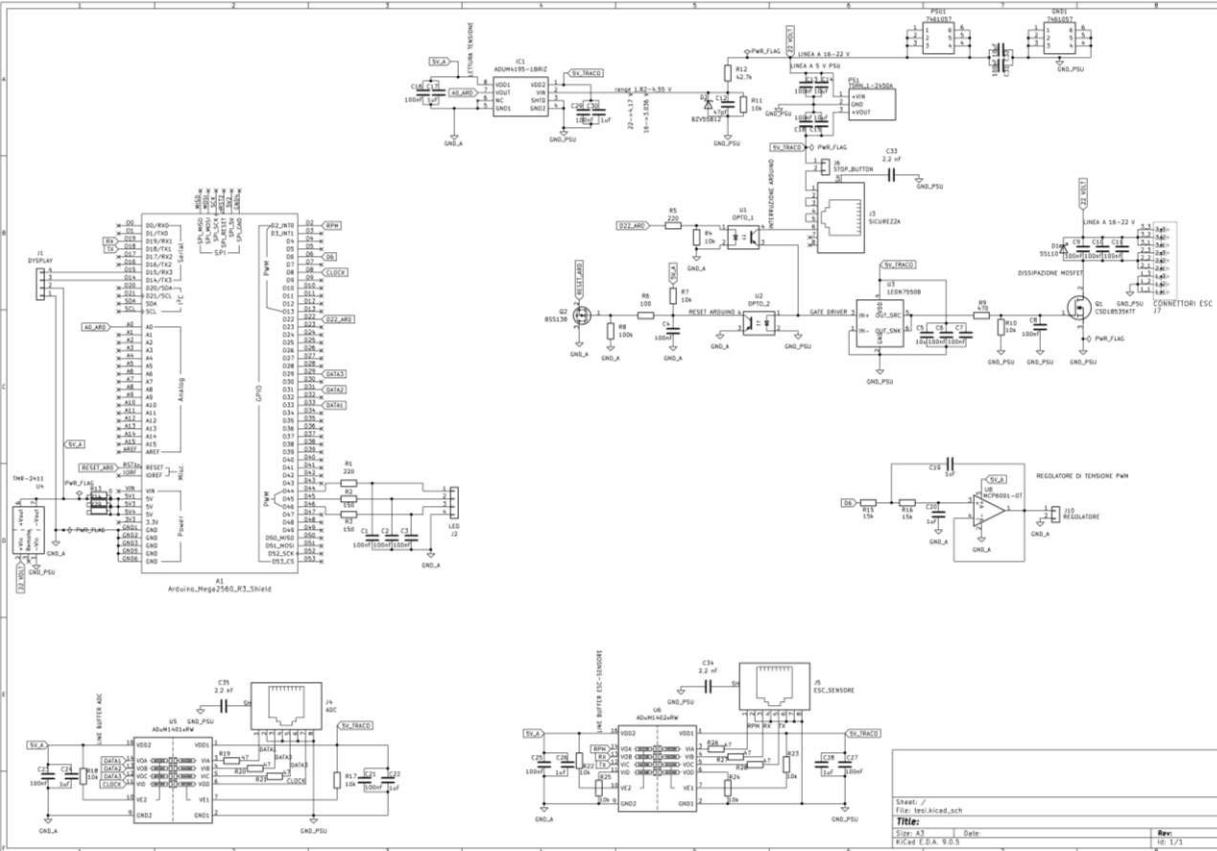


BOTTOM

Lo Schematico



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



Arduino Mega 2560 Rev 3

Alimentazione



Sulla scheda è presente una suddivisione della zona di alimentazione in 3 piani separati:

- 16-22 Volt
- Gnd Motore
- Gnd scheda

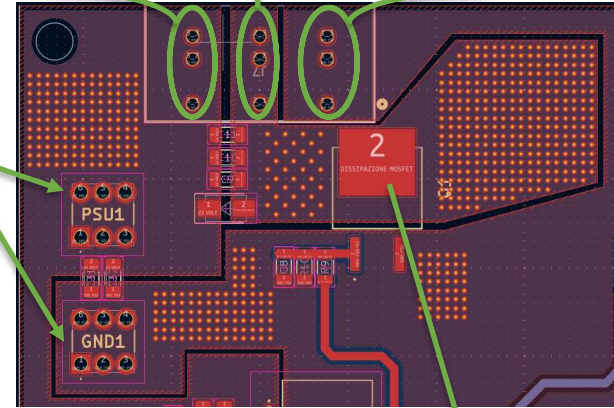
La potenza erogata dall'alimentatore viene distribuita alla scheda attraverso 2 terminali a 6 pin.

La trasmissione della potenza al motore e alla scheda di acquisizione dati avviene per mezzo di un connettore 1720479 (Phoenix Contact).

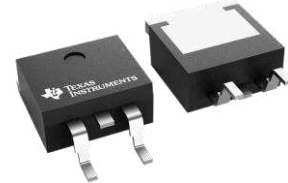


Wurth Elektronik

Alimentazione Massa motore Massa scheda

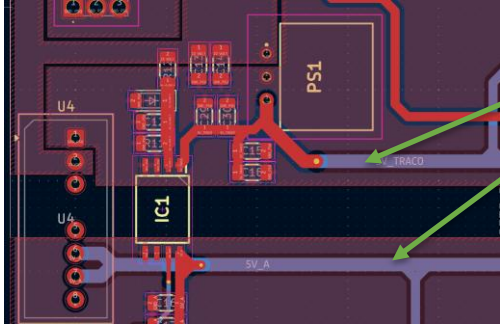


CSD18535KTT



Questo sistema garantisce l'alimentazione di entrambe le schede anche a sicurezza aperta, con motore fermo

Gestione delle alimentazioni



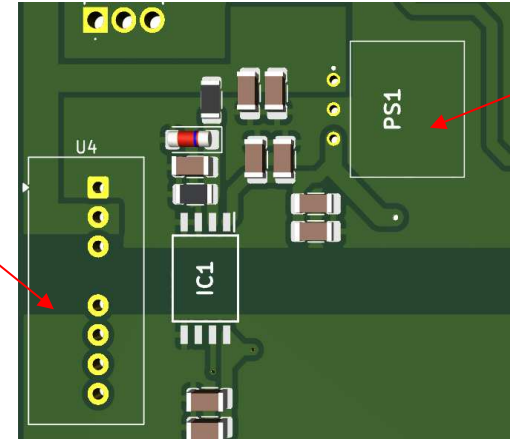
- Il circuito di controllo a 5 Volt è alimentato separatamente da
- DC-DC converter → TSNR_1-2450A
 - Arduino

I due canali di alimentazione sono separati per garantire l'isolamento tra Arduino e il resto della scheda.

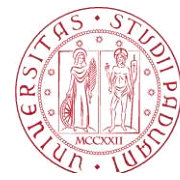
Se la scheda Arduino non fosse in grado di erogare la potenza richiesta è prevista un'alimentazione ausiliaria tramite TMR 2411, DC-DC converter isolato



In serie alle uscite a 5 V di Arduino sono inserite resistenze da 0 Ω . Qualora non si desideri alimentare il circuito tramite Arduino, è necessario dissaldare tali resistenze per interrompere il collegamento.

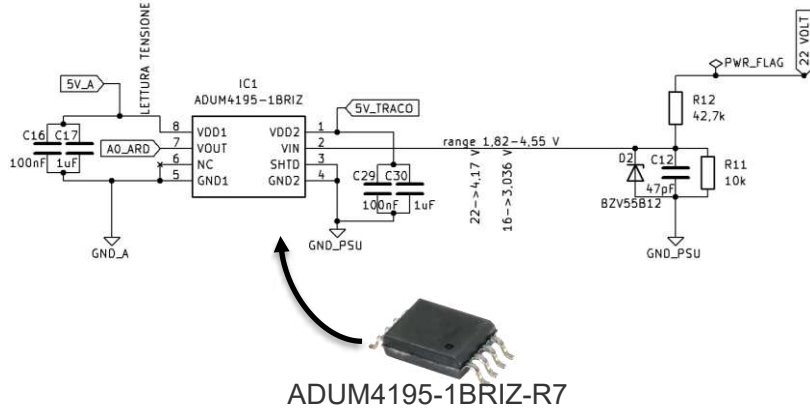


Lettura della Tensione



La tensione in uscita dall'alimentatore viene inizialmente applicata a un partitore resistivo, che ne riduce l'ampiezza riportandola entro i limiti di ingresso dell'operazionale isolato.

Il segnale condizionato viene quindi acquisito da Arduino tramite il pin analogico A0. Il microcontrollore elabora il valore letto (tenendo conto del fattore di scala del partitore) e ricostruisce così la tensione reale erogata dal power supply.



ADUM4195-1BRIZ-R7

$$\text{Tensione in ingresso } V_{in} = V_{psu} \cdot \frac{R_{11}}{R_{11} + R_{12}}$$

V_{PSU} [V]	9,6	16	22	24
V_{IN} [V]	1,8113	3,0189	4,1509	4,5283



V_{in}

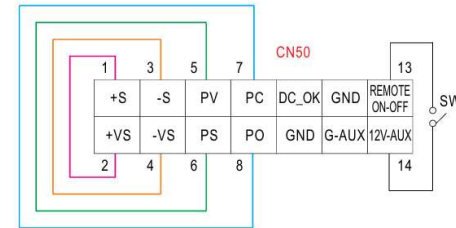
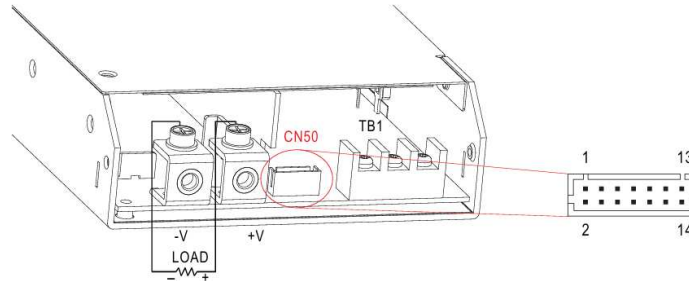
```

1  const int feedbackPin = A0;           // Pin analogico collegato al partitore
2  const float feedbackRatio = 5.273;   // Rapporto del partitore
3  float usbVoltage = 5.0;              // Tensione di riferimento ADC (Vcc)
4
5  // Funzione che legge e calcola la tensione reale
6  float readSupplyVoltage() {
7
8      int adcValue = analogRead(feedbackPin); // 1 Legge il valore ADC (0-1023)
9
10     float voltageOnPin = adcValue * usbVoltage / 1023.0;
11     // 2 Converte il valore ADC in tensione reale presente su A0
12
13     float realVoltage = voltageOnPin * feedbackRatio;
14     // 3 Moltiplica per il rapporto del partitore -> tensione reale alimentatore
15
16     return realVoltage;
17 }
    
```

Alimentazione



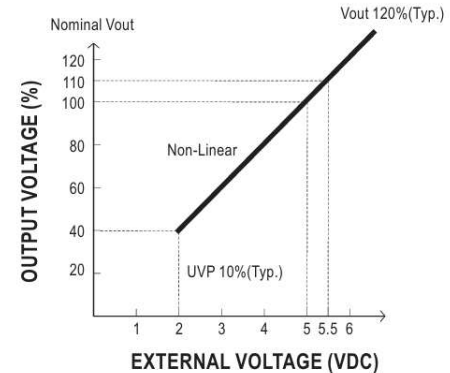
- Esigenze del sistema
- Motivazione della scelta di un Alimentatore switching
 - versatilità
 - dimensioni
 - controllo



- Aree di funzionamento RSP-750

- Oltre alla regolazione tramite il potenziometro integrato, la tensione di uscita può essere regolata dal 40% al 110% del valore nominale.

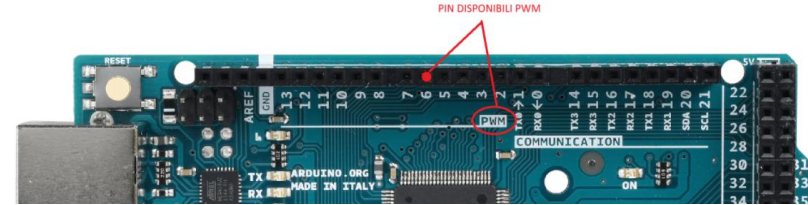
INPUT \ MODEL	24V
100~264VAC	751.2W 31.3A



Controllo Alimentatore in Tensione



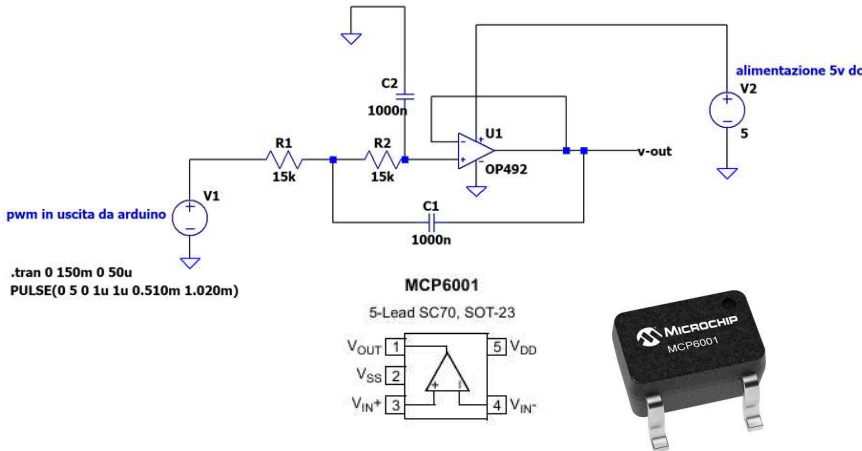
- Requisiti del costruttore
- Ripple accettato alimentatore 10mV p.p (datasheet)
- Opzioni di Generazione del segnale di controllo
 - Dac dedicato
 - Filtro del secondo ordine



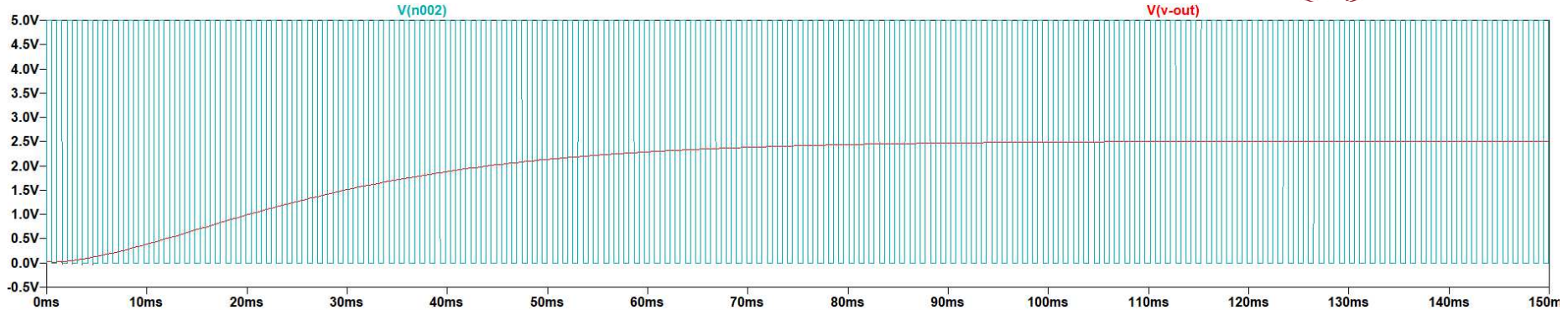
Scelta e dettagli del filtro $D = 50\%$

- $V_1 = \frac{4A}{\pi}$ • $V_{pp} = 2V_1$
- $|H(j\omega)| = \frac{0.01}{6.37}$ • $|H(j\omega)| \approx \left(\frac{f_c}{f}\right)^2$
- $f_c = f_{PWM} \cdot \sqrt{1.57 \times 10^{-3}}$
- $R = 15\text{ k}\Omega$, $f_c = 38\text{ Hz}$
- $C = \frac{1}{2\pi R f_c} = \frac{1}{2\pi \cdot 15000 \cdot 38}$

$$C \approx 2.79 \times 10^{-7} F = 279\text{ nF}$$

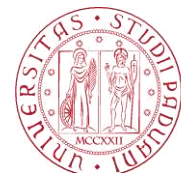


Simulazione Spice



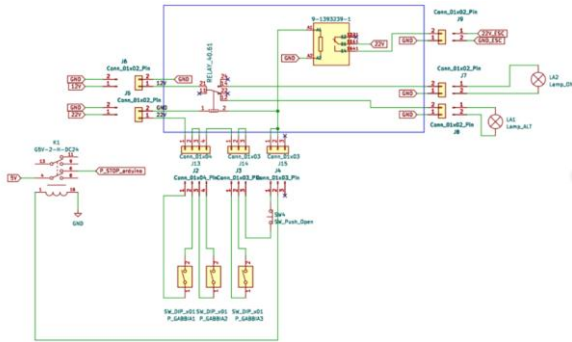
- Transitorio elevato
-pro e contro
- Attenuazione del ripple fino a 4mV
- Scelte progettuali alternative

Led di Sicurezza

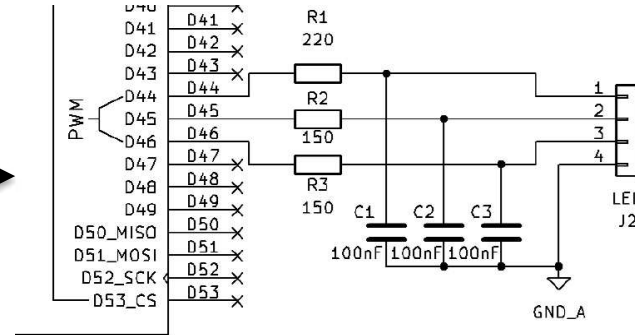


- Collegamento: pin D (44,45,46)
- Controllo

```
1 void loop() {  
2  
3   if (digitalRead(SENSE_PIN) == LOW) {  
4     // Continuità presente  
5     setRGB(0, 255, 0); // VERDE  
6   } else {  
7     // Nessuna continuità  
8     setRGB(255, 0, 0); // ROSSO  
9   }  
10  
11 }
```



Schematico precedente



Schematico aggiornato

Dimensionamento Heatsink



R _{DS(on)} Drain-to-source on-resistance	V _{GS} = 4.5V, I _D = 100A	2.3	2.9	mΩ
	V _{GS} = 10V, I _D = 100A	1.6	2.0	mΩ

- Potenza

$$P = R \cdot I^2 \longrightarrow 2,9 \cdot 10^{-3} \cdot 30^2 = 2,6w$$

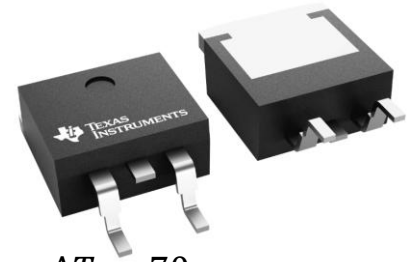
- Temperatura

Op temp (°C)
-55 to 175

$$T_{(box)} = 40^{\circ}C$$

$$T_{(junction)} = 110^{\circ}C$$

$$\Delta T = T_j - T_b = 110 - 40 = 70^{\circ}C \longrightarrow R_T = \frac{\Delta T}{P} = \frac{70}{2.6} \approx 26^{\circ}C/W$$

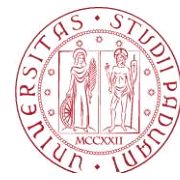


- Scelta del dissipatore e collegamento sulla PCB



- ABL components 14x14x10 mm \rightarrow 26,5 °C/W
- Tipologia di collegamento
- Alternative per mantenere una temperatura interna più bassa

Mosfet e relativo pilotaggio



mosfet : Sezionamento di potenza (CSD18535KTT)

- Condizione operativa (mosfet ON)
Il mosfet conduce e collega la massa dell'ESC alla massa del PSU, chiudendo il percorso di ritorno

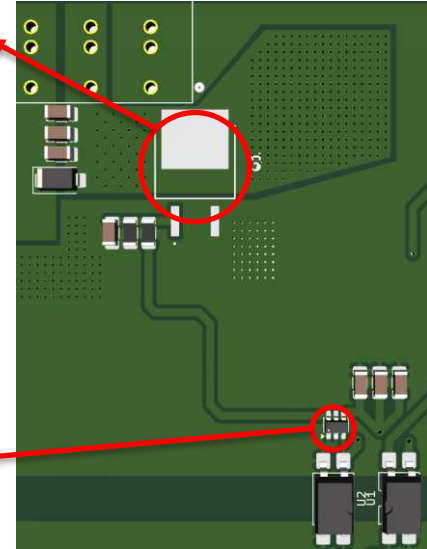
$$V_{ESC} \rightarrow 16 - 22 V$$

- Condizione di sicurezza (mosfet OFF)
Il mosfet è spento e interrompe il collegamento della massa dell'ESC

$$V_{GS} \rightarrow 0 V$$

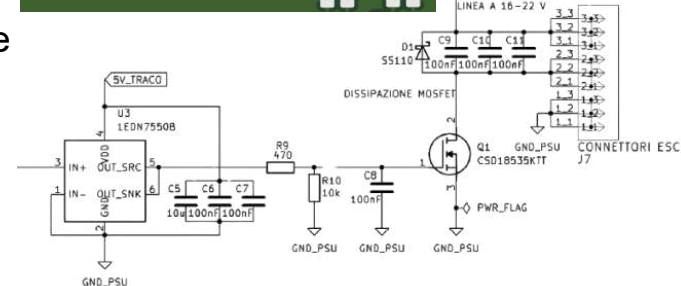


1edN7550B



Gate driver : stadio di pilotaggio che fornisce la corretta V_{GS} e la corrente di picco per caricare/scaricare rapidamente il mosfet, garantendo :

- Commutazioni rapide
- Minori perdite
- Maggiore immunità ai disturbi



Optoisolatori e circuito di sicurezza



Rj45 e microswitch:

Tramite una porta Rj45 la scheda è collegata a un circuito di sicurezza composto da 3 microswitch installati in serie sulla gabbia; l'apertura di uno qualunque genera il fault e disabilita il sistema.

Stop Button:

Interruttore normalmente chiuso che se premuto apre il circuito di sicurezza e disabilita la potenza

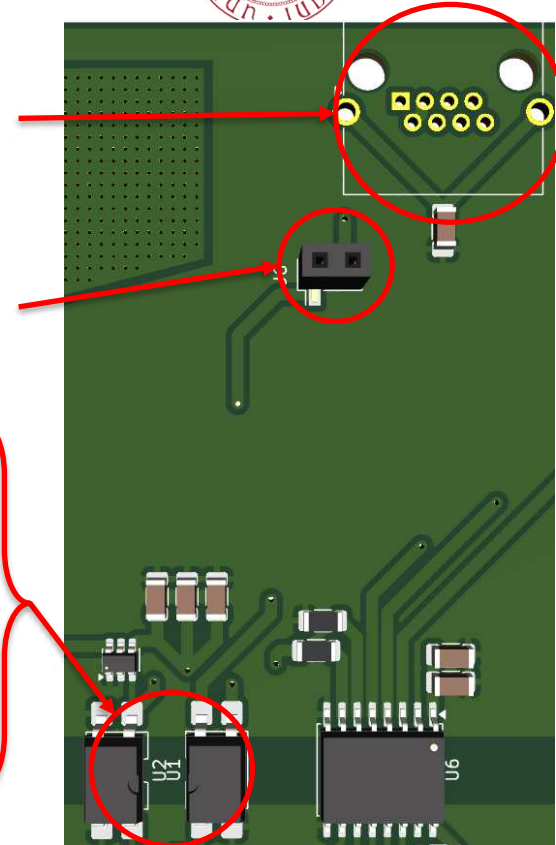
Optoisolatori (LTV-817S) :

- Optoisolatore 1

Utilizzato come interruttore software attraverso il pilotaggio del pin d22.

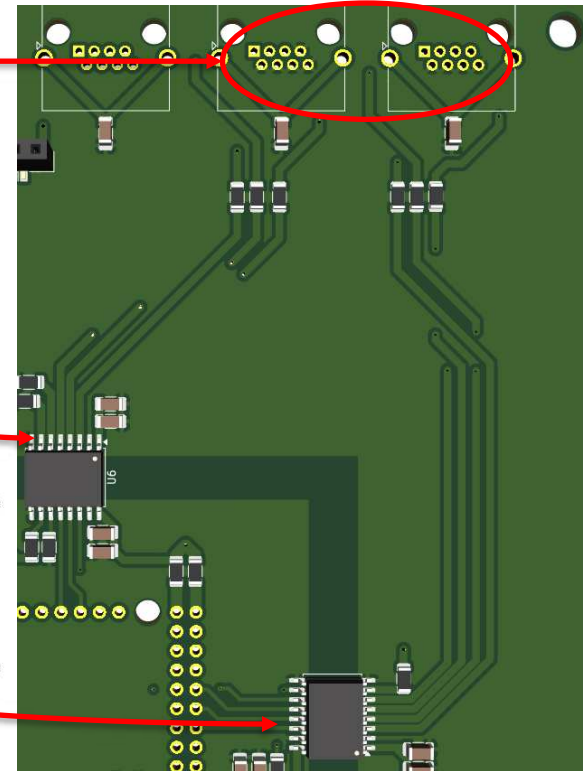
- Optoisolatore 2

Implementa un reset hardware in caso di fault: Quando la sicurezza si apre e/o l'optoisolatore precedente si spegne la sua uscita forza la linea di reset.



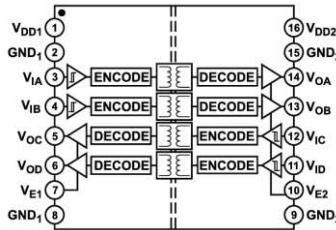
Comunicazione dati

- Porte Rj45 : consentono la trasmissione di dati tra la scheda di controllo e quella di interfaccia al banco prova.
- Isolatori Digitali



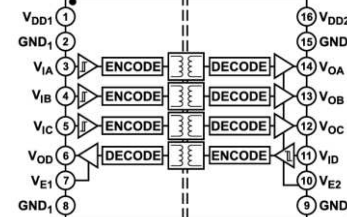
ADUM1402 : isola e garantisce l'integrità dei segnali da ESC e sensore di velocità verso Arduino e viceversa.

- →TX : trasmissione dati esc
- ←RX : ricezione dati esc
- ←RPM : sensore di velocità



ADUM1401 : isola galvanicamente i segnali tra acquisizione delle celle di carico e Arduino.

- ←DATA 1 : cella di carico #1
- ← DATA 2 : cella di carico #2
- ← DATA 3 : cella di carico #3
- →CLOCK : segnale che temporizza trasmissione dati



Codice regolazione alimentatore



```
2 // --- Main Control Logic ---
3
4 // Convert desired voltage to PWM duty cycle
5 float pwmValue = (currentVoltage / maxSupplyVoltage) * 255;
6 analogWrite(pwmPin, pwmValue);
7
8
9 // --- Voltage Ramp Control ---
10
11 while (abs(currentVoltage - targetVoltage) > 0.01) {
12     // Increase or decrease voltage gradually
13     if (targetVoltage > currentVoltage)
14         currentVoltage += 1.0;
15     else
16         currentVoltage -= 1.0;
17
18     // Convert voltage to PWM signal
19     float pwmValue = (currentVoltage / maxSupplyVoltage) * 255;
20
21     // Limit PWM between 0 and 255
22     pwmValue = constrain(pwmValue, 0, 255);
23
24     // Apply PWM to control the power supply
25     analogWrite(pwmPin, (int)pwmValue);
26
27     delay(200); // Time between ramp steps
28 }
29
```

Conversione Tensione PWM

Rampa controllata in tensione

Incremento/Decremento graduale

Ricalcolo PWM per ogni step

Impostazione limiti di saturazione

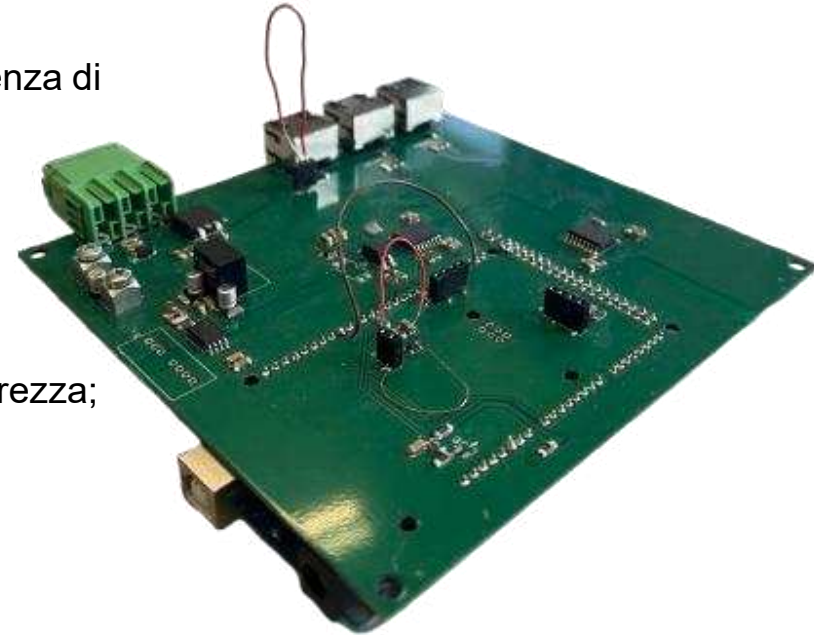
Applicazione del comando al filtro

Delay e slew rate tra gli step

Risultati test



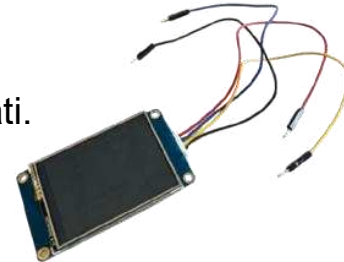
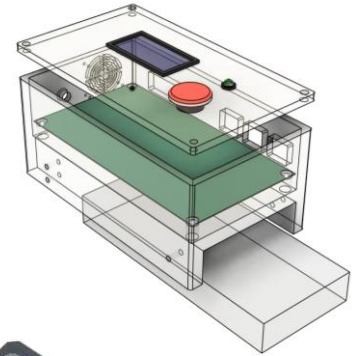
- Corretto funzionamento del controllo dell'alimentatore e assenza di rumore;
- Alimentazione ausiliaria di Arduino non necessaria;
- Corretto funzionamento del sistema elettromeccanico di sicurezza;
- Modificato il pin di TX di uno dei 2 line buffer da D18 a D9.
- Necessaria analisi della linea del pin d22 di Arduino;



Sviluppi futuri



- Scatola: progettazione e realizzazione di una scatola comprensiva di scheda, alimentatore e dispositivi dedicati.
- Sensore di velocità: montaggio, taratura e calibrazione del sensore di velocità, per poter analizzare il segnale rpm seguendo la predisposizione circuitale.
- Display : implementazione del codice per poter visualizzare i dati desiderati.



1222·2022
800
ANNI



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Thanks for your attention





UNIVERSITÀ
DEGLI STUDI
DI PADOVA