

800 ANNI UNIVERSITÀ DEGLI STUDI DI PADOVA

# Design di un sistema di caratterizzazione degli attuatori di un drone

Laurea triennale Ingegneria Meccatronica

Nicholas Busetti Denis Calovi Thomas Robusti

800 ANNI UNIVERSITÀ DEGLI STUDI DI PADOVA

Il nome d'arte:  
*Coleottero*

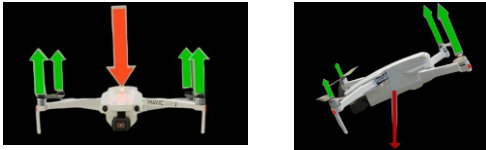


**OBBIETTIVO:** FORNIRE DATI PER LA PROGETTAZIONE DEL SISTEMA DI CONTROLLO DEL DRONE:

- Quale velocità impostare al motore affinché sviluppi una determinata forza verticale?
- Quale velocità impostare al motore affinché sviluppi una determinata coppia di rotazione?

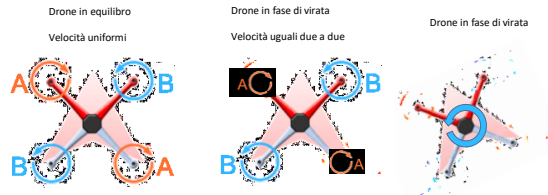
800 ANNI UNIVERSITÀ DEGLI STUDI DI PADOVA

**ANALISI DELLE FORZE**



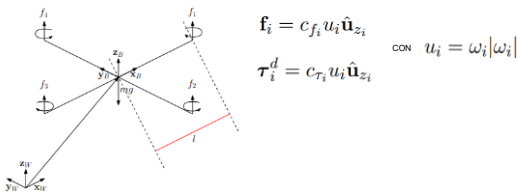
800 ANNI UNIVERSITÀ DEGLI STUDI DI PADOVA

**ANALISI DEI MOMENTI**



800 ANNI UNIVERSITÀ DEGLI STUDI DI PADOVA

**DINAMICA DEI DRONI**



800 ANNI UNIVERSITÀ DEGLI STUDI DI PADOVA

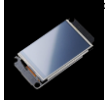
**PRESENTAZIONE DEL NUOVO BANCO PROVA**





**DISPOSITIVI PRINCIPALI**

- **MOTORE TAROT** : motore brushless;
- **ESC**: L'Electronic Speed Controller
- **CELLA DI CARICO** TAL220
- **ARDUINO UNO e MEGA**
- **DISPLAY NEXTION**

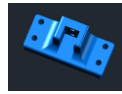


**PROGRAMMI PRINCIPALI**

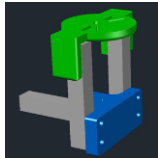
- **IDE ARDUINO**
- **MATLAB-SIMULINK**
- **AUTOCAD, SOLID EDGE, INVENTOR**
- **CURA, PRUSA SLICER**
- **NEXTION EDITOR**



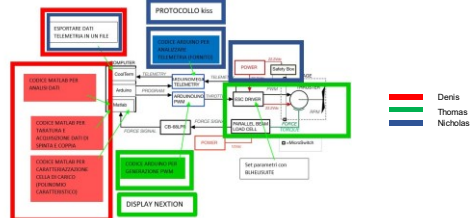
**SUPPORTI IN 3D PER CELLE DI CARICO**



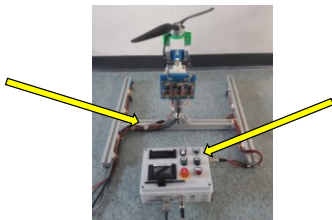
**SUPPORTI IN 3D PER CELLE DI CARICO**



**SCHEMA A BLOCCHI DEL PROGETTO**



**IMPIANTO ELETTRICO**



**QUADRO ELETTRICO: evoluzione**





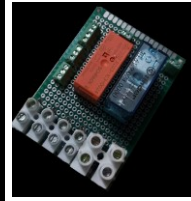
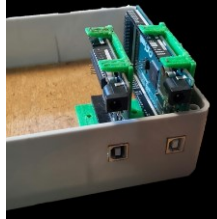
QUADRO ELETTRICO: evoluzione



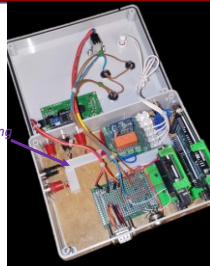
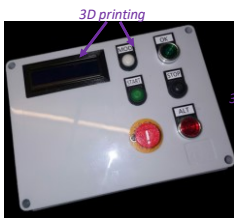
Tagliato e inserito legno



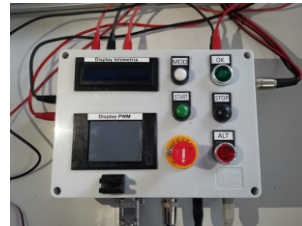
QUADRO ELETTRICO: evoluzione



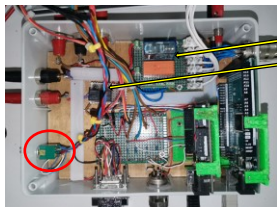
QUADRO ELETTRICO : prima versione



QUADRO ELETTRICO



QUADRO ELETTRICO



Circuito di sicurezza

- Relè anti ripartenze improvise
- Contatto NC del relé

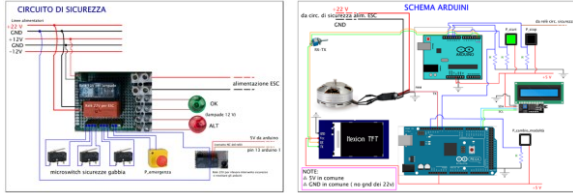


CIRCUITO DI SICUREZZA





SCHEMI ELETTRICI DEL QUADRO



PROGRAMMA ARDUINO PWM



```

//////////////////////////////////////
//PARAMETRI DISPLAY//////////////////////////////////////
HexButton bOn = HexButton(1, 4, "b1");
HexButton bOff = HexButton(1, 5, "b2");

HexText letturaCase = HexText(1, 8, "c3");
HexText letturaDuty = HexText(1, 9, "c4");
HexText scritturaRun = HexText(1, 12, "c6");

HexTouch *hex_listen_list [] = {
  &bOn, &bOff, NULL
};
//////////////////////////////////////

```

HOME PAGE



```

//FUNZIONI PER DISPLAY
void bton( void *ptr) {
  digitalWrite(13, HIGH);
  statoPulsanteMotoreStart = 1;
}

void btoff( void *ptr) {
  digitalWrite(13, LOW);
  statoPulsanteMotoreStart = 0;
  statoPulsanteMotoreStop = 1;
  Reset_AVR();
}

```



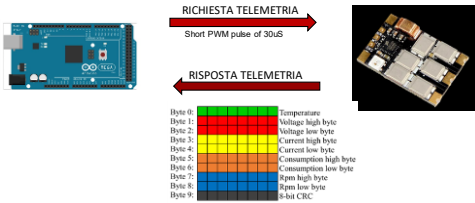
```

//VERIFICA DEL FUNZIONAMENTO DEL MOTORE CON DUTY
//950-1000-050-050-100-150-200-250-300
case 4:
  comando_telemetria = 1;
  if (comando_telemetria == 1) {
    digitalWrite(Pin_comando_telemetria, HIGH);
  }
  sprintf(label, "b100", "Running...");
  hexTouch(hexTouchObtack);
  void btoff(void);
  funzione_messa_action();
  if (d == run_pwm_pos) {
    void btoff(void);
  }
  case stop (1):
    sprintf(label, "b01", "0");
    letturaCase.hexText(lectura);
  }
  duty = duty + d_pwm_pos *
  value = duty * 200;
  case stop (1):
    sprintf(label, "b01", "00 + 0");
    letturaDuty.hexText(lectura);
    void btoff(void);
  while ((bOnPressed) / 1000 - x) == (d + d_pwm_pos) {
    digitalWrite(Pin_comando_telemetria);
    void btoff(void);
    funzione_messa_action();
    digitalWrite(Pin_PWM, value);
  }
}

```



Comunicazione ESC- arduino : protocollo KISS telemetry



Programma arduino telemetria: punti salienti

```

int i; computedCRC = get_crc(0x01, 3);
if (i computedCRC == buff(0)) {
  Controllo CRC

  buff(0) = ((buff(0) & 0x0F) | 0x01);
  Salvataggio in variabili dei dati contenuti nel buffer di ricezione

  temp = temp;
  tempVoltage = ((buff(0) < 0x0F) < 0x0F) / 100.0;
  tempCurrent = ((buff(0) < 0x0F) < 0x0F) / 100.0;
  tempConsumption = ((buff(0) < 0x0F) < 0x0F) / 100.0;
  tempRPM = ((buff(0) < 0x0F) < 0x0F) / (rpm / (rpm / 2));
  Conversioni per unità di misura

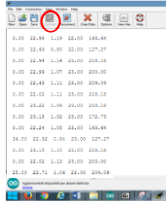
  Velocità meccanica
  Velocità elettrica
}

```



Salvataggio dati da seriale 1.9.1.964

```
Serial.print(trueTemp);
Serial.print(" ");
Serial.print(trueVoltage);
Serial.print(" ");
Serial.print(trueCurrent);
Serial.print(" ");
Serial.print(trueConsumption);
Serial.print(" ");
Serial.println(trueRPM);
Serial.println("");
```



File .txt

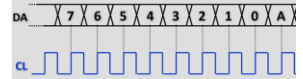
2022-09-01	01:00	22.49	1.07	28.00	185.84
2022-09-01	01:00	22.44	1.00	28.00	127.27
2022-09-01	01:00	22.49	1.04	28.00	208.18
2022-09-01	01:00	22.49	1.07	28.00	200.00
2022-09-01	01:00	22.44	1.00	28.00	208.18
2022-09-01	01:00	22.50	1.11	28.00	228.18
2022-09-01	01:00	22.52	1.08	28.00	208.18
2022-09-01	01:00	22.51	1.00	28.00	178.79
2022-09-01	01:00	22.54	1.00	28.00	168.89
2022-09-01	01:00	22.50	0.95	28.00	127.27
2022-09-01	01:00	22.51	1.00	28.00	208.18
2022-09-01	01:00	22.51	1.00	28.00	200.00
2022-09-01	01:00	22.71	1.04	28.00	208.18



Visione telemetria in tempo reale su display LCD

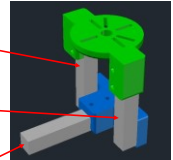
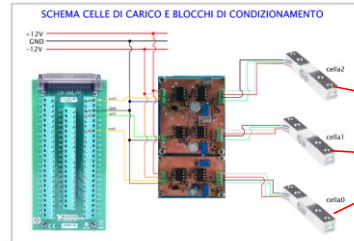
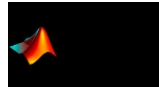
```
switch (modalita)
{
case 0:
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("temperatura");
  lcd.setCursor(0, 1);
  lcd.print(trueTemp);
  lcd.setCursor(11, 1);
  lcd.print("gradi");
  break;
case 1:
```

Implementazione semplice grazie allo standard seriale i2c

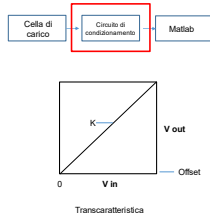
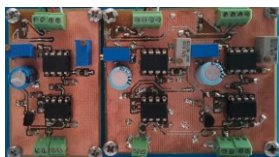


ACQUISIZIONE DATI

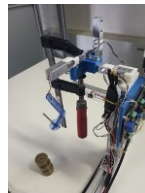
- Sistema di acquisizione dati
- Calibrazione celle di carico
- Analisi dati
- Risultati



CIRCUITO DI CONDIZIONAMENTO

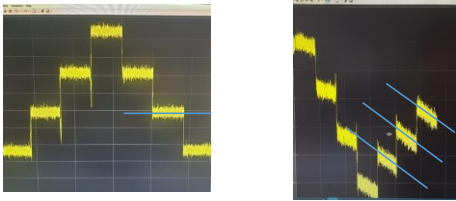


CALIBRAZIONE CELLE

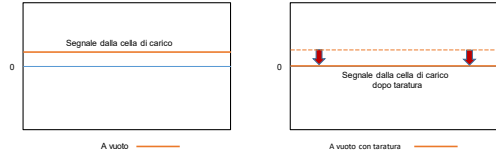




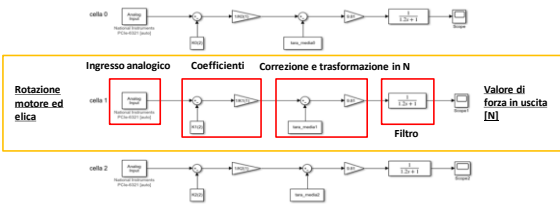
**Problema riscontrato**



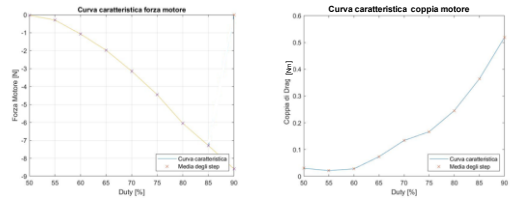
**IMPLEMENTAZIONE ALGORITMO TARATURA**



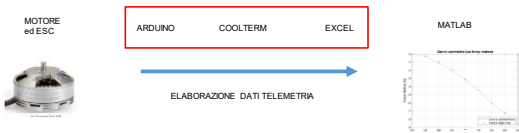
**ACQUISIZIONE DATI**



**RELAZIONE DUTY-SPINTA MOTORE**



**VELOCITA' DI ROTAZIONE DEL MOTORE**



**DATI TELEMETRIA**

COOLTERM				EXCEL			
15:48:28	0,00	22,73	0,73	7,00	163,64		
15:48:28	0,00	22,18	0,80	7,00	163,64	15	48
15:48:28	0,00	22,50	0,79	7,00	209,09	15	48
15:48:28	0,00	22,22	0,76	7,00	218,18	15	48
15:48:28	0,00	22,44	0,77	7,00	218,18	15	48
15:48:28	0,00	22,25	0,82	7,00	218,18	15	48



ANALISI DATI IN MATLAB



TELEMETRIA MOTORE

15	40	28	385,84
15	40	28	385,84
15	40	28	385,84
15	40	28	385,84
15	40	28	385,84
15	40	28	385,84
15	40	28	385,84
15	40	28	385,84
15	40	28	385,84
15	40	28	385,84

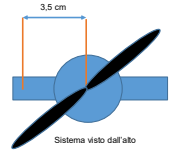
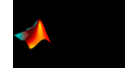
Dati di FORZA dalle CELLE DI CARICO



coefficiente\_coppia  
coefficiente\_forza



ANALISI DATI IN MATLAB

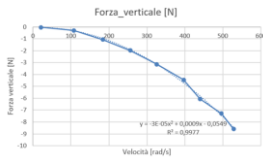


- Media velocità telemetria
- conversione velocità in rad/s
- Inserimento **braccio** per il calcolo della coppia
- Determinazione dei coefficienti tra velocità e forza/coppia

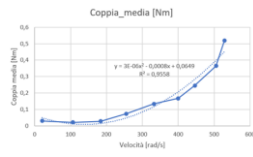
coefficiente\_forza -3.0597e-05  
coefficiente\_coppia 2.9053e-06



RISULTATI



coefficiente\_forza -3.0597e-05

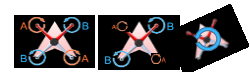
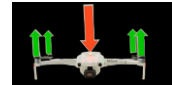


coefficiente\_coppia 2.9053e-06



CONCLUSIONI

- Scelta della **forza/coppia** che deve sviluppare il motore
- La relazione forza/coppia – velocità di rotazione è nota
- Impostazione della **velocità di rotazione corretta**
- Il controllo risulta OTTIMIZZATO!



COMPONENTI INTEGRATIVE

- Video tutorial su:
  - Cablaggio
  - Procedura di funzionamento
- Manuale di istruzioni
- Schemi Elettrici
- Qr-code



Obiettivo:  
facilitare utilizzo e sviluppi futuri.



RINGRAZIAMENTI

Giulia Michieletto  
Stefano Michieletto  
Massimiliano Bertoni

Nicholas Busetti

Denis Calovi

Thomas Rebutti