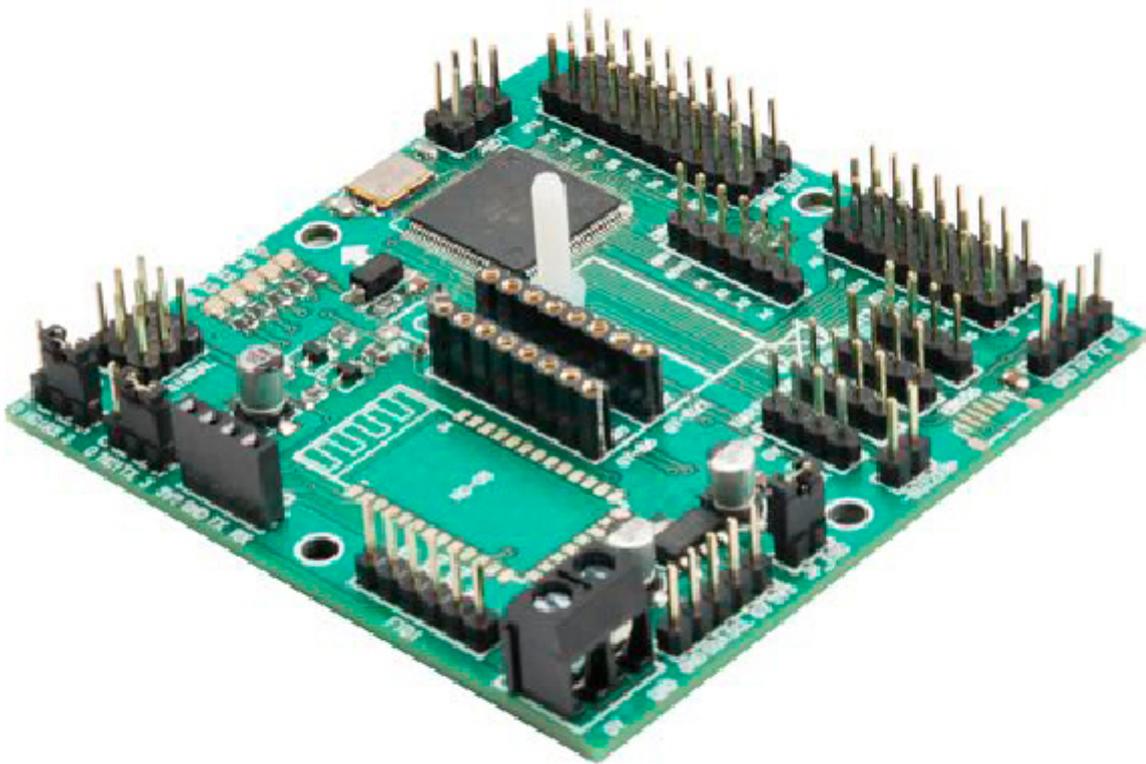


Board quadricottero Open Source

Prezzo: 28.69 €

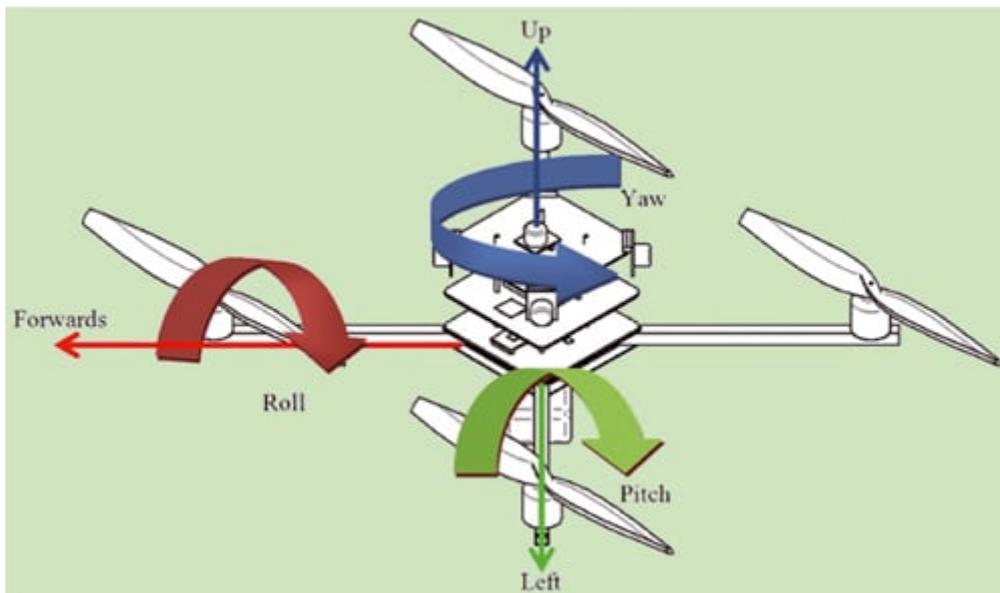
Tasse: 6.31 €

Prezzo totale (con tasse): 35.00 €



Scheda basata su Arduino Mega, che permette di gestire i motori di un quadricottero, ricevere i comandi da un radiocomando per aeromodelli e supportare la telemetria tramite smartphone. Oltre all'Atmega 2560, programmato con il firmware open source denominato MultiWii (permette di controllare multicotteri, elicotteri ed aerei), dispone 8 uscite per motori, 8 ingressi per ricevitori RC standard e compatibile con ricevitori PPM, 4 porte seriali per debug/ Bluetooth Module/OSD/GPS/telemetria, 3 uscite servo per il controllo di un sistema gimbal (pitch, roll, trigger), I²C-Bus per aggiungere sensori o dispositivi, connettore per modulo GPS, compatibile con moduli IMU GY-521, GY-86 ed esterni, connettore per convertitore USB/seriale FTDI, uno stadio di alimentazione a 3,3V per eventuali dispositivi aggiuntivi e LED di segnalazione. L'altro elemento di fondamentale importanza è il modulo IMU (Inertial Measurement Unit, ovvero la piattaforma che rileva accelerazione, movimenti sui tre assi ecc.) disponibile separatamente e inseribile sull'apposito connettore della scheda. Sia la scheda che il software supportano diverse configurazioni di volo: da semplici aeromodelli a multirotori. La scheda premontata (solo la sezione SMD) non comprende il modulo IMU, i motori, le eliche, ecc (vedere prodotti correlati). [Clicca qui per leggere l'articolo completo presentato su Elettronica In n. 205 - Maggio 2016 e n. 206 Giugno 2016](#)

Terminologia del Volo



- **PITCH:** rappresenta il beccheggio, ovvero l'oscillazione sull'asse trasversale.
- **ROLL:** detto anche rollio, è il termine usato per definire la rotazione del modello sul proprio asse longitudinale.
- **YAW:** rotazione di coda, cioè la rotazione del modello sul proprio asse verticale.
- **PASSO:** indica l'inclinazione della pala dell'elica rispetto al piano di rotazione.
- **CLASSE:** indica la categoria a cui appartiene il quadricottero, identificata dalla distanza in diagonale tra due motori in millimetri.
- **ESC:** acronimo di electronic speed controller ovvero il driver di potenza per i motori brushless.
- **BEC:** acronimo di battery eliminator circuit indica che l'ESC è fornito di regolatore di tensione interna.
- **PDB:** acronimo di power distribution board ovvero lo stampato usato per portare l'alimentazione della batteria a tutti gli ESC.
- **MODE2:** modalità di utilizzo del telecomando con gas e yaw a sinistra ed il controllo pitch e roll sulla destra.
- **MODE1:** modalità di utilizzo del telecomando con gas e yaw a destra e pitch e roll sulla sinistra.
- **BIND:** indica l'associazione tra il telecomando ed il ricevitore in apparati radio RC.
- **PID:** indica la terna dei parametri proporzionale, integrativo e derivativo di un controllo.

Caricamento del BOOTLOADER e Test della scheda

Se il microcontrollore della scheda è sprovvisto di bootloader è necessario caricarlo; allo scopo potete usare un apposito programmatore o più semplicemente una scheda Arduino facente funzioni di programmatore. In quest'ultimo caso consultate la Tabella 2 per conoscere le connessioni da effettuare tra la scheda programmatore e la nostra. Poi aprite l'IDE di Arduino e caricate lo sketch di esempio denominato ArduinoISP.ino sulla Arduino usata come programmatore, quindi accedendo al menu Strumenti>tipo di Arduino impostate come scheda target "Arduino Mega 2560 or Mega ADK"; fatto ciò impostate Arduino come programmatore (Strumenti>Programmatore>Arduino as ISP) quindi avviate la programmazione con il comando Strumenti>Scrivi il Bootloader. A procedura completata (sono richiesti alcuni minuti) sconnettete il programmatore ed interfacciate la scheda al PC tramite l'ausilio di un convertitore USB/Seriale connesso al connettore FDTI; la vostra scheda sarà vista dall'IDE di Arduino come una scheda Arduino Mega. Potrebbe succedere che a fine procedura compaia l'errore "avrdude: verification error, first mismatch at byte 0x1e000" accorso durante la rilettura dei fuses scritti, anche se la procedura di scrittura è andata buon fine.

Il Firmware

Il firmware che gira all'interno della scheda non è dei più semplici: oltre alla stabilizzazione del volo c'è da gestire gli allarmi, la telemetria, la posizione e tanto altro, per questo abbiamo preferito utilizzare un software già esistente, completamente open source e sviluppato negli ultimi anni anche col contributo degli appassionati in rete, denominato MultiWii. Esso permette di controllare multicotteri, elicotteri ed aerei; il sito di riferimento è www.multiwii.com (la versione italiana è www.multiwii.it). Il progetto è nato utilizzando il giroscopio di un Nintendo Wii Motion Plus (il più economico e facile da reperire all'epoca) e una scheda Arduino mini pro, ma ora supporta un gran numero di sensori e schede. Il firmware è molto leggero e può girare su un piccolo Atmega 328, ma con tutte le funzioni attivate (navigazione e GPS) è necessario utilizzare un Atmega 2560. Il software viene gestito nel repository <https://code.google.com/p/multiwii/>. Scaricando il pacchetto software si dispone del software per Arduino e la GUI, scritta in Processing, da far girare sul PC per gestire le impostazioni e la telemetria. Il software per Arduino usa solo librerie di sistema e quindi il suo utilizzo è immediato disponendo del solo IDE di Arduino. Per configurare la board e usare la telemetria e la pianificazione missioni, è disponibile anche il software WinGui e una applicazione per smartphone Android.