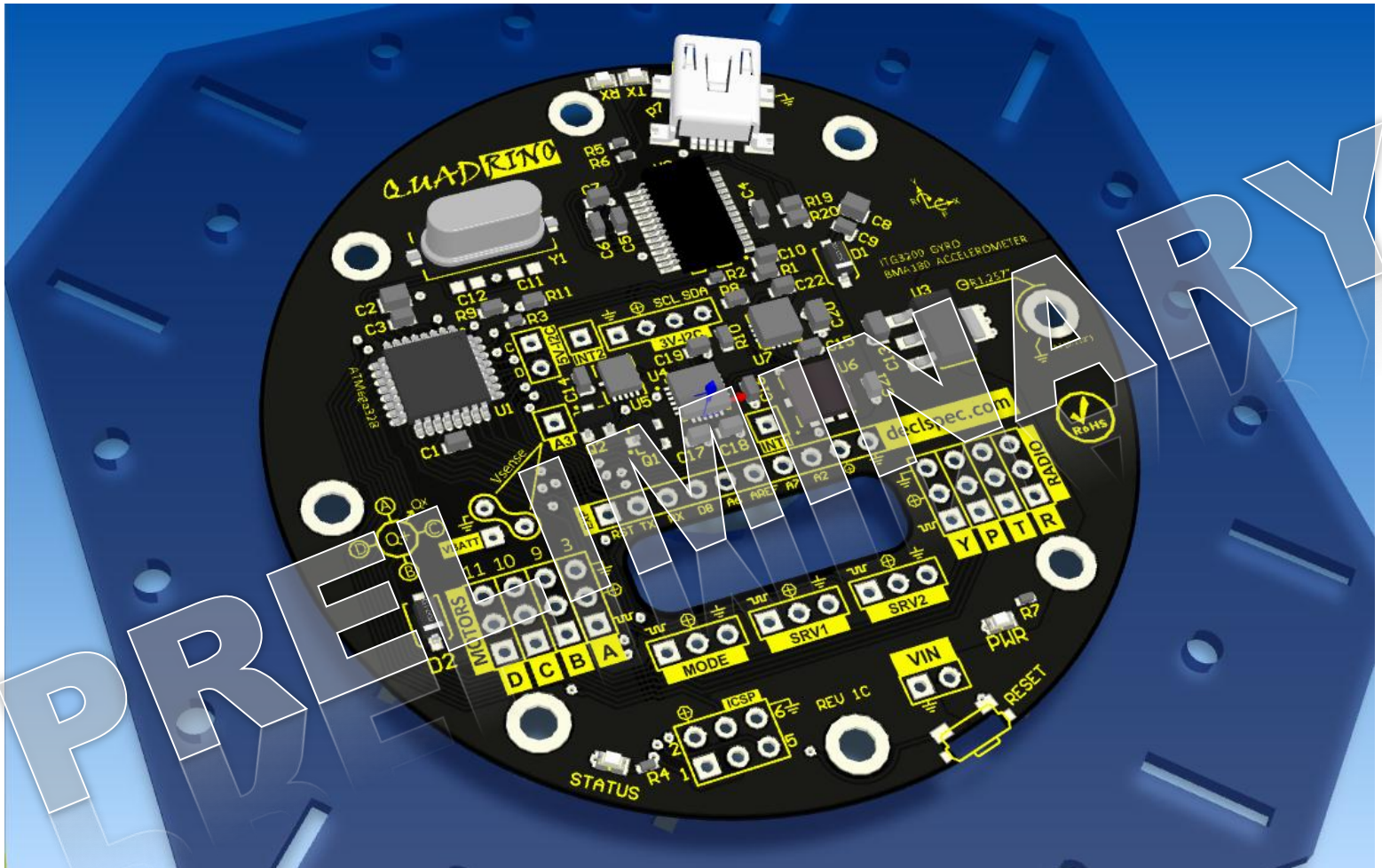




DECLSPEC.COM

Ready-to-Fly Flight Controller  
Arduino Compatible with 10-DOF IMU



## Quadrino Data sheet

Release date	April 10 <sup>th</sup> , 2011
Data sheet revision	1.0
Document number	QUADRINO-DS01-01

# Quadrino

Ready-to-Fly Flight Controller  
Arduino Compatible with 10-DOF IMU

DECLSPEC.COM

## Gyro

The Quadrino Flight Controller is a ready-to-fly alternative to the complicated process of gathering the many separate pieces required to make a MultiWii multi-rotor copter.

The Quadrino includes the Arduino micro-controller, Gyro, Accelerometer, Barometer, Magnetometer and on-board usb interface. Plugin your radio receiver channels and ESC motor wires and you are ready-to-fly<sup>(1)</sup>.

The Quadrino supports all MultiWii features such as auto-level mode (acc mode), heading hold (mag mode), and altitude hold (baro). There is a cheaper alternative that removes the magnetometer and barometer.

## Accelerometer

## Magnetometer

## Barometer



## Features

- **100% fully arduino compatible. (based on Arduino Nano v3)**
- Uses **stock MultiWii source code** requiring only a few #define parameters to be set based on your setup.
- Contains **ITG3200 Gyro** and **BMA180 Accelerometer**.
- ADV version has **HMC5883 Mag** and **BMP085 Baro**.
- Solder points for 5v and 3.3v I2C bus.
- Mosfet logic level translators for 3.3v I2C bus.
- 400KHz I2C bus speed supported.
- Servo ports for camera mounts (SRV1 & SRV2).
- Expansion header breaks out unused Arduino pins.
- Works in Quad+, QuadX and also Tri or Hex copters.
- Mini-USB connector on board - direct connection!
- Voltage monitoring - add two resistors to enable battery voltage monitoring, then use expansion header pin to trigger low-batt alarm.
- Test radio receiver inputs within MultiWii config. (Protection diode prevents USB from powering ESCs/motors, but powers receiver.)

## Benefits




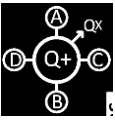



- Average setup time is 20mins, from wiring to upgrading firmware, verifying in MultiWii Config, to first flight!
- All sensors supported by stock MultiWii source.
- Forget the datasheet at home, all headers, signals, and diagrams are labeled on the PCB!
- 62mil PCB is strong and sturdy to survive hard crashes or rough landings.
- 8 mounting holes support most standard mounts for Quad-X, Quad+, and Tri copters.
- Lower control-loop cycle time than popular Wii/nunchuk units translates to more stable flight characteristics. (avg. 2500 versus 4500.)
- Wiring can be pulled up through the center of the PCB for less wire mess.
- **Easily add other I2C sensors using 4pin header. (Sparkfun breakouts highly recommended.)**

(1) Firmware upload may be required if your setup differs from the default settings.

# WIRING THE QUADRINO

## PCB ICONIC DESCRIPTIONS

The following icons are used on the PCB to indicate the type of signal. These icons make wiring in the field simple even without this datasheet.

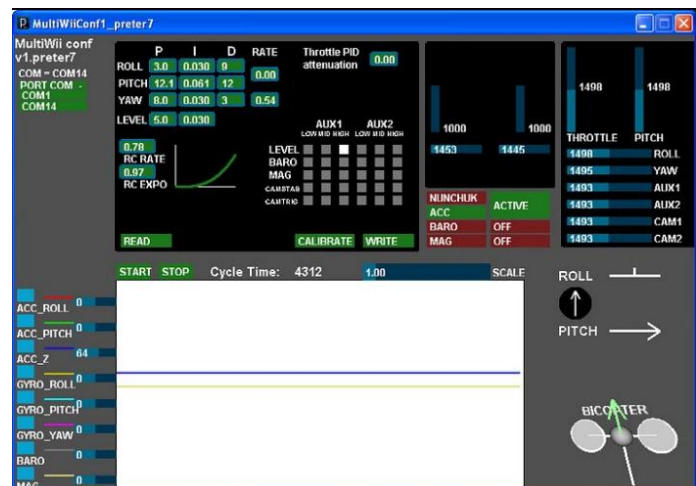
Icon	Name	Description
	Power	5V Power
	Ground	Power ground plane. The indicated pin or PCB pad goes to the PCB ground plane.
	Signal	Used in the motor, server and receiver headers to indicate the pin that carries the PWM signal.
	Motor Diagram	This diagram indicates the placement and naming of the motors relative to the Quadrino board. In Quad+ mode, the front of the Quadrino points directly to motor A. In Quad-X mode, the front of the Quadrino points between motor A and C.
	Radius Dimension	This specified a radius dimension. In this case, the distance from any mounting hole to the center of the board.
	Coordinate System	This icon indicates the coordinate system the Quadrino uses. All the sensors are positioned to match this coordinate system. This is also the default coordinate system for the MultiWii firmware.
	RoHS	The Quadrino PCB is RoHS approved. This means a higher level of environmentally safe processes were used in it's manufacture.

## USING THE MULTIWII CONFIG PROGRAM

You should get familiar with the MultiWii config program before wiring up the Quadrino. Download the latest software and MultiWii source code from the MultiWii web site.

The web site is found at <http://www.multiwii.com/>, but the source code and config program is found on the google code site.

### VERIFY YOUR QUADRINO



Before mounting and wiring the Quadrino, you should plug the usb cable directly into the bare and unconnected Quadrino and run the MultiWii config program. You will be able to communicate with the Quadrino and you should see the sensor output change as you move the Quadrino PCB in your hand.



You **should** take care to eliminate any static charge buildup by grounding yourself before handling the Quadrino. Usually it suffices to touch a metal surface on your desktop computer as the case is connected to the earth ground on the power plug. You may also use a static discharge strap available at your local Radio Shack. Other than dangerous static discharges, the Quadrino is safe to handle in this un-mounted state.

## CHOOSING YOUR POWER STRATEGY

There are a few different power strategies when wiring the Quadrino, the speed controllers (ESCs), and the radio receiver. The various wiring methods may change certain power behaviors of the overall copter, specifically during MultiWii config (when the Quadrino is plugged into USB), during normal flight, and during certain control failure situations.

The goals in any of our power strategies are to:

1. Prevent motors from being powered by USB when connected.
2. Provide the safest mode during control failure situations for both nearby people and copter. The most common control failure is loss of radio.
3. Limit electrical interference between the Quadrino, the motive system (ESCs/motors), and receiver.
4. Optionally, we would like the receiver to be powered when using MultiWii config (when USB is connected).

### POWER STRATEGY OVERVIEW

	<b>Strategy</b>	<b>Behavioral Effect</b>	
		Normal <b>flight</b> mode, <b>USB</b> connected, or <b>FailSafe</b> during control failure	
<b>1</b>	ESC powers Quadrino through motor pin header, Quadrino powers radio.	<i>Flight</i>	ESCs power Quadrino, Quadrino powers Receiver.
		<i>USB</i>	USB powers Quadrino and Receiver, ESCs isolated/off.
		<i>FailSafe</i>	If receiver fails, Quadrino continues to fly the copter. Risk of fly-away copter and to nearby people from spinning props.
<b>2</b>	ESC powers radio, radio powers Quadrino through radio pin header.	<i>Flight</i>	Radio powers Quadrino, ESC powers radio.
		<i>USB</i>	USB powers Quadrino and Radio, ESCs isolated/off.
		<i>FailSafe</i>	If receiver fails, power fails on the Quadrino thus ESCs cut out. Copter free falls. Prevents fly-away copter loss! Though opinions differ, this is probably the safest failsafe mode to nearby people as there are no spinning props.
<b>3</b>	Battery powers Quadrino through VIN port, Quadrino powers radio.	<i>Flight</i>	Battery powers Quadrino. Quadrino powers radio.
		<i>USB</i>	USB powers Quadrino and radio. ESCs isolated/off
		<i>FailSafe</i>	If receiver fails, Quadrino continues to fly the copter. Risk of fly-away copter and to nearby people from spinning props.
<b>4</b>	Second battery powers Quadrino through VIN port.	<i>Flight</i>	Lowest electrical interference between ESCs and Quadrino controller as power systems are well isolated.
		<i>USB</i>	USB or battery powers Quadrino and Radio. ESCs isolated/off.
		<i>FailSafe</i>	If receiver fails, Quadrino continues to fly the copter. Risk of fly-away copter and to nearby people from spinning props. If second battery fails, copter free falls.

There are other combinations on these strategies but generally you will choose one of the first 3. Using a second (much smaller) battery to power the Quadrino separately from the motors is ideal for the lowest electrical interference but is generally not needed when good power distribution planning is done. Use strategy 4 to troubleshoot a misbehaving copter and you want to eliminate electrical interference as a possibility.



*MultiWii version 1.7 now contains FAILSAFE features. You should enable this to provide a safe mode of operation during control failure!*

#### Power distribution Do's and Don'ts:

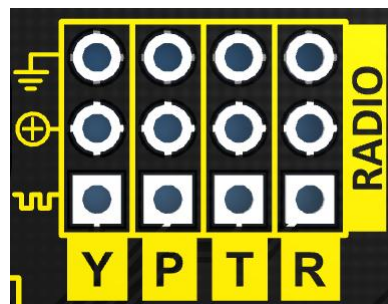
1. **DO** connect **only one** power wire and one ground wire from the power source – be it from the ESC or receiver. Multiple ground wires cause ground loops and multiple power pins are redundant and just add weight.
2. **DO** power the Quadrino from **only one** ESC, or the single radio receiver (depending on your power strategy). **DO NOT** connect all your ESC's power pins to the Quadrino. The Quadrino does not use much current and a single power source will do.
3. **DO** use large gauge wire as appropriate for powering your ESCs. Normal RC header gauge wire (about 22Gauge) is fine for the Quadrino as it uses very little current.
4. **DO** connect each ESC power using separate large gauge wire. **DO NOT** wire your ESCs like a daisy chain (i.e. from A -> B -> C -> D.) Your wiring should look like a star. If your frame is made of metal you may choose to make your frame GND.
5. **DO** use a large capacitor (300-1000mF) across input power pins if you suspect electrical interference is a problem. Try this if the Quadrino unexpectedly resets. With proper wiring a capacitor is not needed.
6. **DO** not wire the Quadrino in such a way that the USB port will power the ESCs. If you notice your motors twitching when USB is connected immediately disconnect and review your power wiring. There is a polarity diode that protects the USB power from going to the ESCs through the motor pin header. However, USB power will go through the radio pin header, if there is a path to the ESCs through the radio receiver your computer's USB port may become overloaded.
7. **DO NOT** connect VBATT directly to either the motor or pin header power pins. Your Quadrino **WILL BE DESTROYED**. The Quadrino core components can only handle up to 5.25 volts. If you are powering the Quadrino from a battery, connect VBATT to the VIN header where the on-board voltage regulator will be used. (The VBATT header for voltage monitoring is also safe for battery voltages.)

#### THE RECEIVER

The Quadrino YPTR header is used to connect to the radio receiver.

If a PPM decoder is used, the THROTTLE pin become the PPM input.

Pin	Name
Y	YAW
P	PITCH
T	THROTTLE
R	ROLL



## THE MODE INPUT

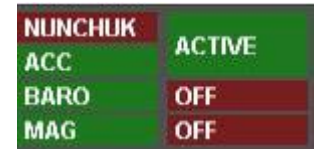
If you have a 5 or more channel receiver you should wire one AUX channel to the mode pin. This allows the control of flying mode via the transmitter. You can use a 2 or 3 position toggle on the transmitter to switch between **acro**batic mode **accelerometer** mode, and enabling of the barometer and magnetometer sensors for heading and/or altitude hold.



**IMPORTANT!** A common step people miss is not configuring the MODE selection through the MultiWii config program. Without this step, only **acro**batic mode will be enabled and accessible during flight and the accelerometer, mag and baro sensors will be unused.

MultiWii copters can change into various flight modes. The activation of each sensor determines how the MultiWii will fly.

**ACCELEROMETER** Provides auto-level flight. The copter will maintain a level pitch and roll when the transmitter controls are in their default position. Before use, the accelerometer must be properly calibrated.



**MAGNETOMETER** Maintains the same heading counteracting any yaw drift. This is a nice feature for beginners who are busy concentrating on controlling the pitch and roll of the copter.

The mag sensor will keep the copter pointed in the same direction from takeoff to landing.

**BAROMETER** When activated this sensor will keep the copter at the same altitude within a small amount of error. This works best in low wind conditions.

Activating the barometer close to the ground can cause greater error as the backwash from the rotors contribute to the pressure on the sensor.

## CONFIGURING THE MODE SELECTOR

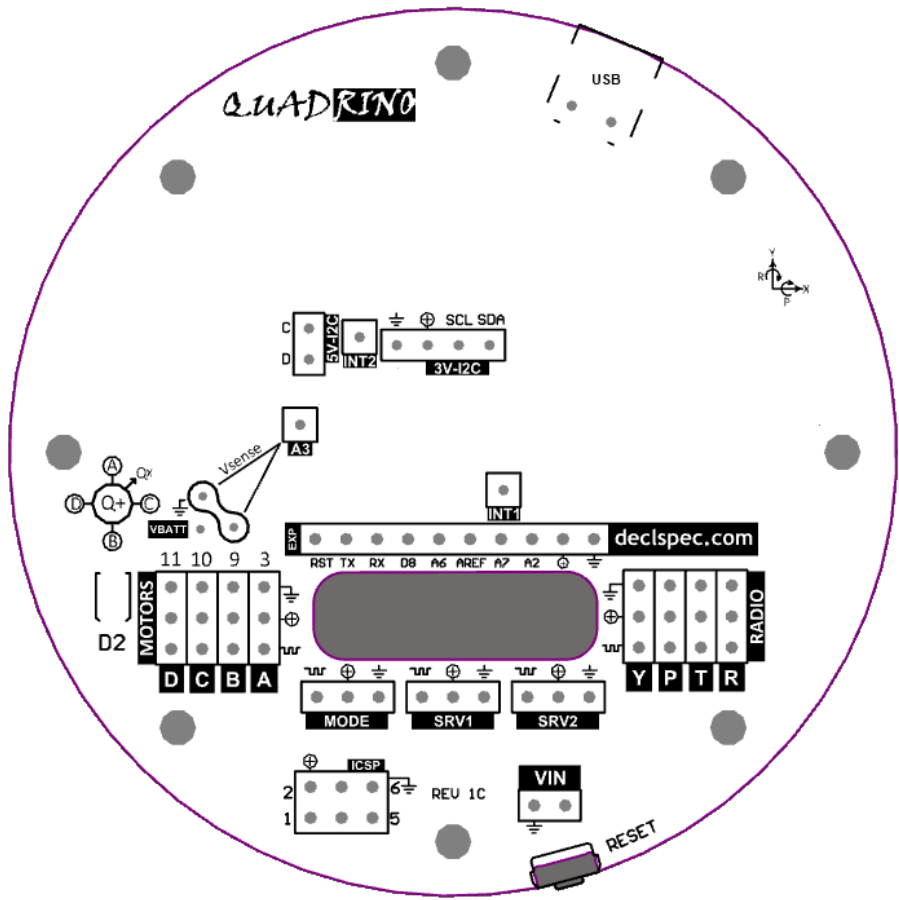
The picture on the right shows the controls within the MultiWii config program that configure how the mode switch (the AUX1 or AUX2 channels) map to the various flight modes. By toggling the state of the white squares you can map the state of the AUX pin (Low, Mid, or High) to which sensors are made active.

As the MultiWii program changes often, you should visit the MultiWii web site for more detailed information (near the bottom of this page):

[http://www.multiwii.com/?page\\_id=104](http://www.multiwii.com/?page_id=104)



# PIN HEADER DESCRIPTIONS



## VIN PIN HEADER

If the onboard voltage regulator (5V) is to be utilized, the battery input should be wired here. If you are powering the Quadrino using power from **one** of the ESCs, you should not use this VIN port. You may optionally unsolder the 5V voltage regulator if desired.

Pin	Name	Description
1	GND	Power Ground.
2	VCC	5V Power.

## EXPANSION PIN HEADER

*This header supplies unused Arduino pins and can be used to expand the Quadrino flight capabilities.*

Pin	Name	Description
1	RESET	Active Low signal, resets Arduino and sensors.
2	TX	Arduino serial TX pin. (Can be connected while USB is connected.)
3	RX	Arduino serial RX pin. ( <b>Do not</b> connect while USB is connected.)

4	D8	D8 pin on the Arduino.
5	A6	A6 pin on the Arduino.
6	AREF	Arduino AREF pin. Analog-to-digital reference signal.
7	A7	A7 pin on the Arduino.
8	A2	A2 pin on the Arduino.
9	VCC	5V output. (same as VCC on motor or receiver headers)
10	GND	Power ground.

### 3V-I2C PIN HEADER

*Most I2C sensors now use a 3.3v I2C bus. The Quadrino uses mosfet level converters to provide bus translation between the ATmega328 (Arduino) and the sensors. You can expand the number of sensors by wiring more sensors through this header.*

Pin	Name	Description
1	GND	Power ground.
2	3V3	3V3 Power. (Can only supply 25mA max. Use a 3V3 regulator if more power is required.)
3	SCL	I2C serial Clock signal.
4	SDA	I2C serial Data signal.

### 5V-I2C PIN HEADER

*Some sensors or devices may still use a 5V bus, although lately less common. Use this header and the expansion header (for power) for these sensors.*

Pin	Name	Description
1	SCL	I2C serial Clock signal.
2	SDA	I2C serial Data signal.

### ICSP PIN HEADER

*This header is used to reprogram or unbrick the Quadrino in case of bootloader corruption. It can also be used as an SPI port for any SPI based sensors.*

Pin	Name	Description
1	MISO	Master Input / Slave Output
2	VCC	5V Power.
3	SCK	SPI CLK
4	MOSI	Master Output / Slave Input

5	RESET	Arduino RESET pin.
6	GND	Power ground.

### SINGLE PIN HEADERS

Name	Description
A3	A3 pin on Arduino. Also used for battery monitoring when enabled. Add appropriate resistors from A3 to Vsense pins depending on your battery parameters.
INT1	Interrupt output on gyro. Not typically used for MultiWii but available for custom implementations. (See ITG3200 datasheet.)
INT2	Interrupt output on accelerometer. Not typically used for MultiWii but available for custom implementations. (See BMA180 datasheet.)

### BATTERY MONITORING PIN HEADERS

*If battery monitoring is enabled, and the proper user-installed resistors are present, an alarm can be wired to sound or light when the battery voltage drops too low.*

Name	Description
A3	A3 pin on Arduino. Add appropriate resistors from here to Vsense pins depending on your battery parameters.
Vsense (1&2)	One is GND (as noted on PCB), the other is attached directly to the VBATT pin. Add appropriate resistors from A3 to Vsense (GND) and A3 to Vsense (VBATT) depending on your battery parameters. (See <i>Battery Monitoring</i> .)
VBATT	Connect this pin directly to the POS+ battery terminal.