1 SYSTEM OVERVIEW

To be detailed:

Power supply and p/s fan, main fan, sensors and actuators, fuses, battery connections, switch connections, RPI / Display / sound, backup alarm.

2 ELECTRONICS BOARD

2.1 PROCESSOR

The on-board processor is an Arduino Mega2560.

P1 through P7, connect to all Arduino connectors except the ICSP header (which is not needed). See schematic for details. It may not be necessary to populate all connector pins, but this may be simplest.

2.2 POWER MANAGEMENT

2.2.1 Connectors

2.2.1.1 Main Power Connector J18

Pin	Name	Function
1	RECT A POS	+ connection to rectifier module
2	RECT A NEG	GND connection to rectifier module
3	RECT B POS	+ connection to second/redundant rectifier module
4	RECT B NEG	GND connection to second/redundant rectifier module
5	INT BAT	+ connection to internal battery, fused on board
6	EXT BAT	+ connection to external battery, not fused on board
7	EXT BAT	+ connection to external battery, not fused on board
8	GND	For batteries or other purposes
9	GND	For batteries or other purposes
10	GND	For batteries or other purposes
11	GND	For batteries or other purposes
12	GND	For batteries or other purposes
13	+24V	+24V for alternate ESC supply
14	Spare	
15	BATT-ONOFF-A	Contact closure to enable/disable backup alarm
16	BATT-ONOFF-B	Contact closure to enable/disable backup alarm

2.2.1.2 Raspberry Pi Power Connector U1

USB type A connector, with only power pins connected, to provide +5V from the clean supply. The cable shield is left open.

We can expect a draw of 2A from the Raspberry Pi.

2.2.1.3 Buck Converters J19, J20 (bottom side)

4 pins matched to Abra buck converter (+IN, -IN, +OUT, -OUT), 39mm x 16.75 mm rectangle.

J19 supplies +5V (clean), J20 supplies +5VP (dirty, for servo)

2.2.2 Circuit Protection

The internal battery is fused (F3). A 10A fuse is specified.

External batteries are required to be fused at the battery.

The ESC power is fused (F1), current TBD.

2.2.3 Sensing

2.2.3.1 BATT_CURRENT_SENSE

Function: monitors internal + external battery current.

Arduino: pin A6 (P3-7); AREF mode: AVCC

Transfer function:$V_{OUT} = (I_{BATT})\ (R_S)\ (R_L) / 5k\Omega + V_{CC}\ R_1 / (R_1 + R_{27})$

Component values: $R_S = 10m\Omega$, $R_1 = R_{27} = 100k$, $R_L = 50k\ (R_1 // R_{27})$; $V_{CC} = 5V$

So: $V_{OUT} = 0.1\Omega \bullet I_{BATT}$

From a 10 bit unsigned ADC count, with AREF = AVCC this gives:

I_{BATT} = count * 50A / 1024

Note: on first prototype, the 10mΩ resistor is paralleled with another, so is 5mΩ. Also, only R27 is present, so RL = 100K. Therefore, the above expression is still correct.

2.2.3.2 BATT_VOLT_SENSE

Function: monitors (bidirectional) internal + external battery current.

Arduino: pin A7 (P3-8); AREF mode: AVCC

Transfer function: $V_{OUT} = (V_{BATT})\ R_{25} / (R_{24}+R_{25})$

Component values: $R_{25} = 10k$, $R_{24} = 22k$

So: $V_{OUT} = V_{BATT} \bullet 0.3125$

From a 10 bit unsigned ADC count, with AREF = AVCC this gives:

V_{BATT} = count• 16V / 1024

Reading accuracy is affected by VCC = 5V accuracy. A calibration may be done and stored in EEPROM as a permanent calibration for the unit.

2.2.4 Battery Current Control

Function: sets V_{GS} (gate-to-source voltage) of pass transistor Q2, which affects charging (and discharging) current. See 2.2.4.3 and 2.2.4.4 for details.

Arduino: D4 in PWM mode

2.2.4.1 Original Transfer Function

V_{OA} = 5V (duty-cycle) (1 + R_{31} / R_{30})

If R_{33} = 0, R_{32} = DNP: V_{GS} = (V_{+12V} – V_{OA})

Else, V_{GS} = (V_{+12V} – V_{OA}) (R_{32} / (R_{32} + R_{33}))

2.2.4.2 Improved Transfer Function

See **Error! Reference source not found.** for details of the required modification

Transfer function: V_{GS} = 5V (duty-cycle) R32 / R30

Setting R_{33} = 220Ω, R_{32} = R_{30} = 10k: Transfer function: V_{GS} = 5V (duty-cycle)

A slightly higher gain may be desirable, but the transistor should be mostly on with V_{GS} = 5V.

Technically the transfer function is also multiplied by the transistor alpha ≈ 1 but this effect should be negligible.

2.2.4.3 Charging Control

Q2 V_{GS}affects the charging current (if mains power is connected) as well as the voltage drop across Q2 in the discharge path. The higher the gate-source voltage, the lower the channel resistance and the more current will flow (at a given V_{DS}).

The improved transfer function is preferred, as it reduces control complexity by removing the dependence on V_{+12V}.

Unfortunately, the exact transfer function of the transistor is unknown, and may vary from unit to unit. Characterization may be required. If the channel resistance R_{Q2} turns out to be fairly constant over some range of source-to-drain voltage drops, charging current can be roughly predicted as:

I_{CHARGE} = (V_{+12V}. – V_{BATT}) / R_{Q2}

2.2.4.4 Mains Power Failure and Discharge Control

Mains power failure may be detected as follows:

- The battery current changes from some positive value to a negative value
- The battery voltage drops from (typically, if fully charged) above 13V to below 12.5V

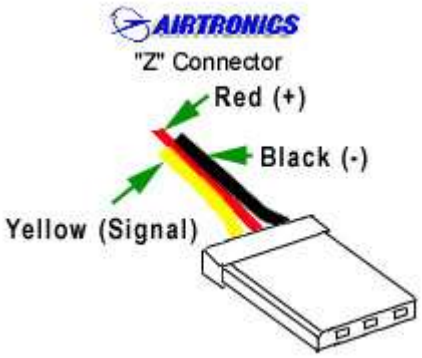
Note that if Q2 resistance is set relatively high (in a typical trickle charge situation with a full battery), the MOSFET body diode will initially conduct, dropping approx. 0.6V when the power is removed and resulting in V_{+12V} ≈ 12.0V (or a bit lower). When this mains power failure is detected, VGS should be set to the maximum. This will turn on the transistor in low resistance mode, bypassing the body diode and burning less power in the transistor, and transferring more power to the load.

2.3 ACTUATOR CONTROLS

2.3.1 Servo

The servo interface allows power to be switched on and off, and conveys a PWM signal for position control.

Arduino: SERVO_OFF D10 (high = off, low = on), SERVO_PWM1 D8 (activation level, PWM mode)



Runs off “dirty” 5V (on its own buck converter), 3A max, 1A continuous.

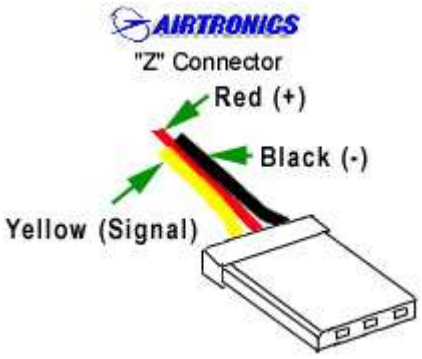
Servo Connector J5

Pin	Name	Function / Notes
1	GND	
2	+5V	
3	PWM control	See ESC data sheet for details

2.3.2 ESC (Speed Control)

12V power on a separate connector

Probably makes sense to fuse this one individually.



Ard-PWM9

Note: positive is NC

ESC Control Connector J3

Pin	Name	Function / Notes
1	GND	
2	+5V	
3	PWM control	See ESC data sheet for details

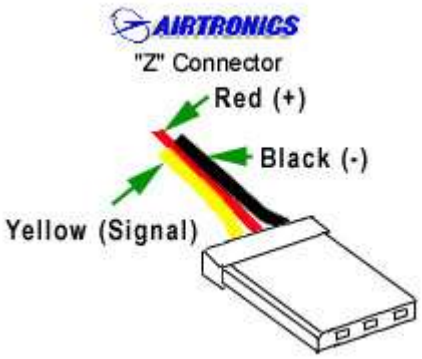
ESC Power Connector J10

Pin	Name	Function / Notes
1	+5V	
2	GND	

2.3.3 Fan

Power +12V on this connector

Tach Ard-D40, with pullup



Fan Connector J14

Pin	Name	Function / Notes
1	GND	
2	+5V	
3	Tachometer	Low-going pulses, contact closure (pullup provided)

2.3.4 Stepper Motor

Stepper Motor Connector J13

Pin	Name	Function / Notes
1	OUT1	TBD
2	OUT2	TBD
3	OUT3	TBD
4	OUT4	TBD

Arduino connections:

IN1: Ard-D45
IN2: Ard-D44
EnA: Ard-PWM2

IN3: Ard-D43
IN4: Ard-D41
EnB: Ard-PWM3

2.4 SENSORS

2.4.1 Flow Sensor

The flow sensor is a Sensirion SFM3300-D. For details see its data sheet. It connects on J17 as follows. Note that the pinout corresponds to that on the flow sensor itself; allowing easy use of a ribbon cable.

Pin	Name	Function / Notes
1	Heater power	**Mistake – this is 12V but should be +5V.** See 2.7.2.
2	SDA	I2C
3	GND	For the interface
4	SCL	I2C
5	Interface power	+5V or connected to two Arduino pins "FLOWSENSOR_POWER"
6	GND	For the heater

Connector J17

Arduino I2C pins D20 and D21 are used for I2C communications.

Jumper JP2 allows selection of +5V for the flow sensor power, or Arduino D24 and D22 ganged together. If the Arduino pins are used, the sensor can be power cycled to reset it if communications fail. **Take care to switch the pins together, as they are shorted to each other.**

O2 Sensor Connector J16

Pin	Name	Function / Notes
1	Sensor +	Positive voltage developed by sensor with respect to GND (25±2mV)
2	Sensor GND	

2.4.2 O2 Sensor

The O2 sensor, connected on J16, generates a DC voltage in the approximate range 25mV – 100mV, which is amplified and conveyed to Arduino pin A5.

2.4.3 Pressure Sensor

Provision exists for two redundant pressure sensors U2 and U3.

Type Honeywell HSCDLND001PGAA5, output range 0.5V – 4.5V (10% - 90% of VCC). Directly interfaced to Arduino A0 and A1 respectively.

2.5 ALARM CIRCUIT

Currently:

Arduino-D26 and D28 (ganged together) energize a relay. **Note that care should be taken to set both pins to the same value since they are connected together.** The buzzer is connected to the NC contacts, so is kept off. The buzzer is turned on by lowering the pins.

Buzzer on when power off, unless turned off

I really think this has to be redesigned!

2.6 OTHER

Flow sensor heater voltage, effect on calibration?

2.7 MODIFICATIONS

2.7.1 Trace Capacity

Type: board mod, ECO

Reason: +12V trace is not wide enough to adequately carry full current for ESC

Place a wire from the J18-1 or -3 to the ESC fuse (16AWG), on the bottom of the board

We may also want to shore up the battery circuit connection.

2.7.2 Flow Sensor Connector J17

Type: board mod, ECO

Reason: the flow sensor heater needs 5V, not 12V

Cut the trace to J17-1

Jump from JP2-3 to J17-1

2.7.3 O2 Sensor Offset Accuracy, add Voltage Reference

Type: ECO

To be described

2.7.4 Decouple AREF

If needed to lower noise.

2.7.5 Fix Unconnected R1 pin

Type: board mod, ECO

Connect R1-2 to R27-1.

FIX NOT NEEDED since U8 doesn't work for negative currents anyway ☹

2.7.6 Fix fuse footprint

Type: ECO

The footprint is too short (F1, F3). Various hacks are in place to accommodate a fuse on the v1.0 board.

2.7.7 Check and possibly upgrade hex diode package

Type: ECO, board mod if urgent

Diodes D2, D3 probably need a higher peak current rating.

2.7.8 Fix RPi USB Connector Pinout

Type: board mod, ECO

Power pins reversed!

2.7.9 Add Diode to Alarm Relay Coil

Type: board mod, ECO

From pin 12 to pin 1.

2.7.10 Increase Shunt Capacity

Type: board mod, ECO

Shunt should be 5W not 500mW, for the case where the battery has to power the ventilator. On the first prototype, two of them were doubled up (giving, perhaps, a 1 watt capacity).

2.7.11 Fix Battery Charge Control

Type: ECO

Measurement is unipolar; it should be bipolar. More generally, we may want to implement a proper charge management circuit. Or, use a current monitor that works bidirectionally.

2.7.12 Stepper Driver Heatsink

Type: board mod, ECO

Stepper driver needs a heat sink. The vertical mount one is probably best; this requires a footprint change.

2.7.13 ESD Protection on Battery Connection

Type: board mod, ECO

This maybe only really needs to be on the external battery connection (although a full analysis of ESD susceptibility – along with testing – should of course be carried out, and might reveal other paths).

On the trace from the external battery to the shunt, a 1uF ceramic capacitor was place (to GND). Additionally, 1k resistors were placed in series with the Arduino analog input pins measuring battery voltage and current.

3 ISSUES AND RESOLUTIONS

3.1 ELECTRICAL SAFETY FOR BATTERY BACKUP AND BACKUP ALARM

The currently designed circuit, with the main power disconnect also turning off the battery and the backup alarm, may pose electrical safety issues since the switch may not be safety rated between its sets of contacts, particularly not for medical standards.

If this poses a problem, the following scheme will replace it:

The mains switch is replaced by a single pole switch, in series only with the rectifier AC input.

The electronics are designed to power down to the microamp level with a soft on/off switch. The battery circuit is never switched off, thus avoiding the necessity to mix high and low voltage circuits on the same switch. The Arduino is kept powered, even when the system is switched off on battery backup. A primary battery provides backup power to the Arduino, which sounds the backup alarm only if operating power is cut while the ventilator is active.