PSAS Launch Tower Upgrade System Overview

Created By: Richard Witherspoon
Created On: April 19th, 2012

Last Updated By: Richard Witherspoon
Last Updated: June 18th, 2012

Launch Tower Goals

- Create system that can have any Linux computer put as controller.
- I/O management through USB external to control computer.
- Launch via relay interlock from rocket and ground control.
- Disconnect power to rocket
- Manage visual and auditory signaling.
- System should NOT function with loss of computer capabilities. Fail to Safety!
- Control computer physical access if lost through Wi-Fi
- Ample cooling and dust management

- Trigger video acquisition of takeoff.

TO DO

- FULL PUNCHLIST
- ----------
- - Ignition Board -
- Select & Purchase the LIPO chargers
- Get dimensions of LIPO batteries
- Adapt Ignition board for LIPO on-board
- Select diodes and LEDs

- - Power Board -
- Add circuitry for umbillical on power board
- Select and purchase 5v and 19v DC/DC converters
- Modify power board for DC/DC converters
- Select diodes and LEDs

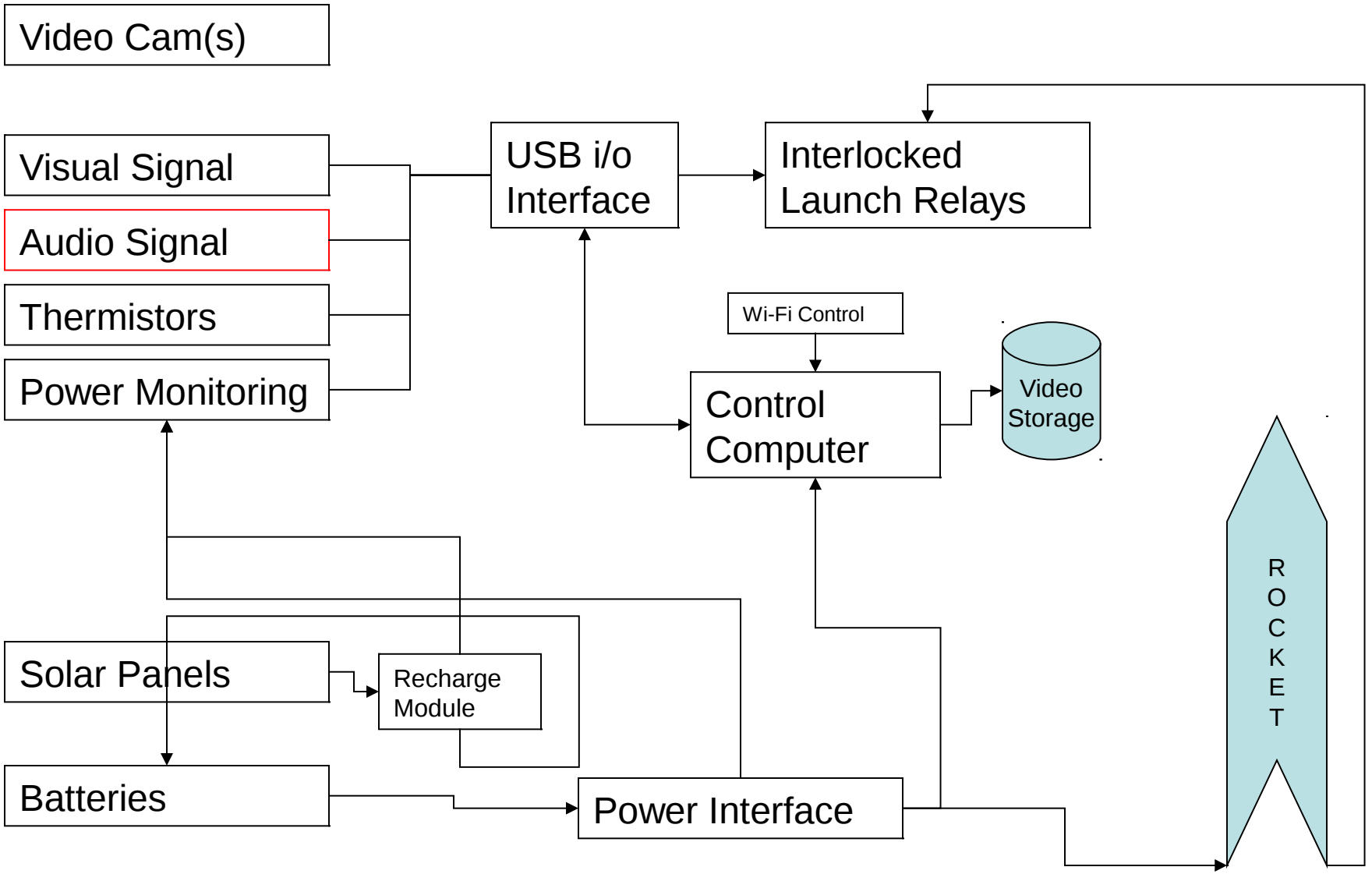
- - LTC Box -
- Dimension parts and figure layout in box
- Get dimensions for solar charger
- Figure out exact connections needed
- Figure out how to route wires
- Find a box that meets the dimension requirements

- - BeagleBoard -
- BeagleBoard - make script to launch webservice at boot - instead of manual start
- BeagleBoard - Remove apache

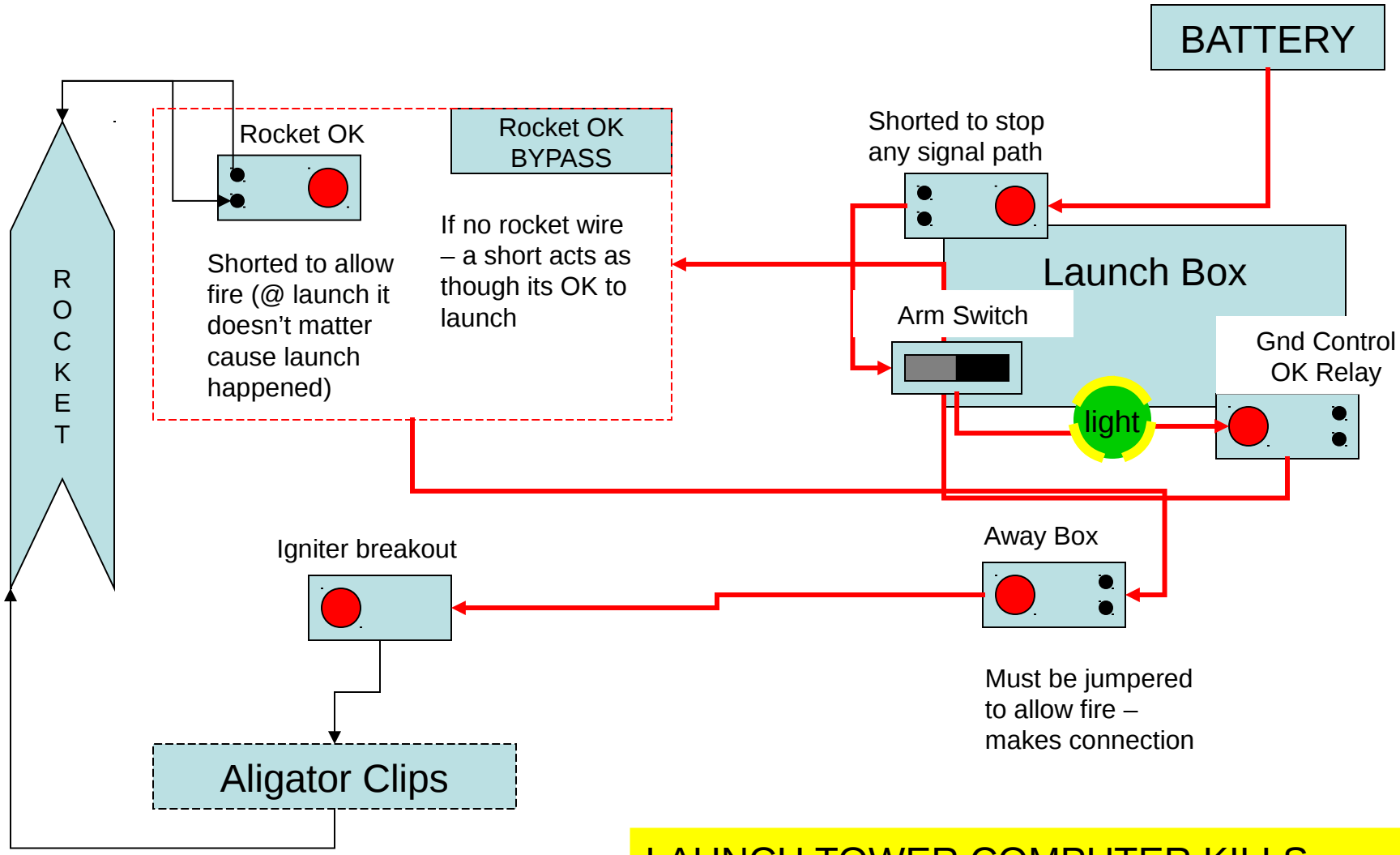
- - Phidgets -
- Application - add error handling and logging
- Get annemometer and wind vane to setup the environmental tacking
- Figure out wire lengths and mountings for sensors on the tower (wind and temp)

- - Other -
- Wiki The setup of everything - in progress

Block Level Design

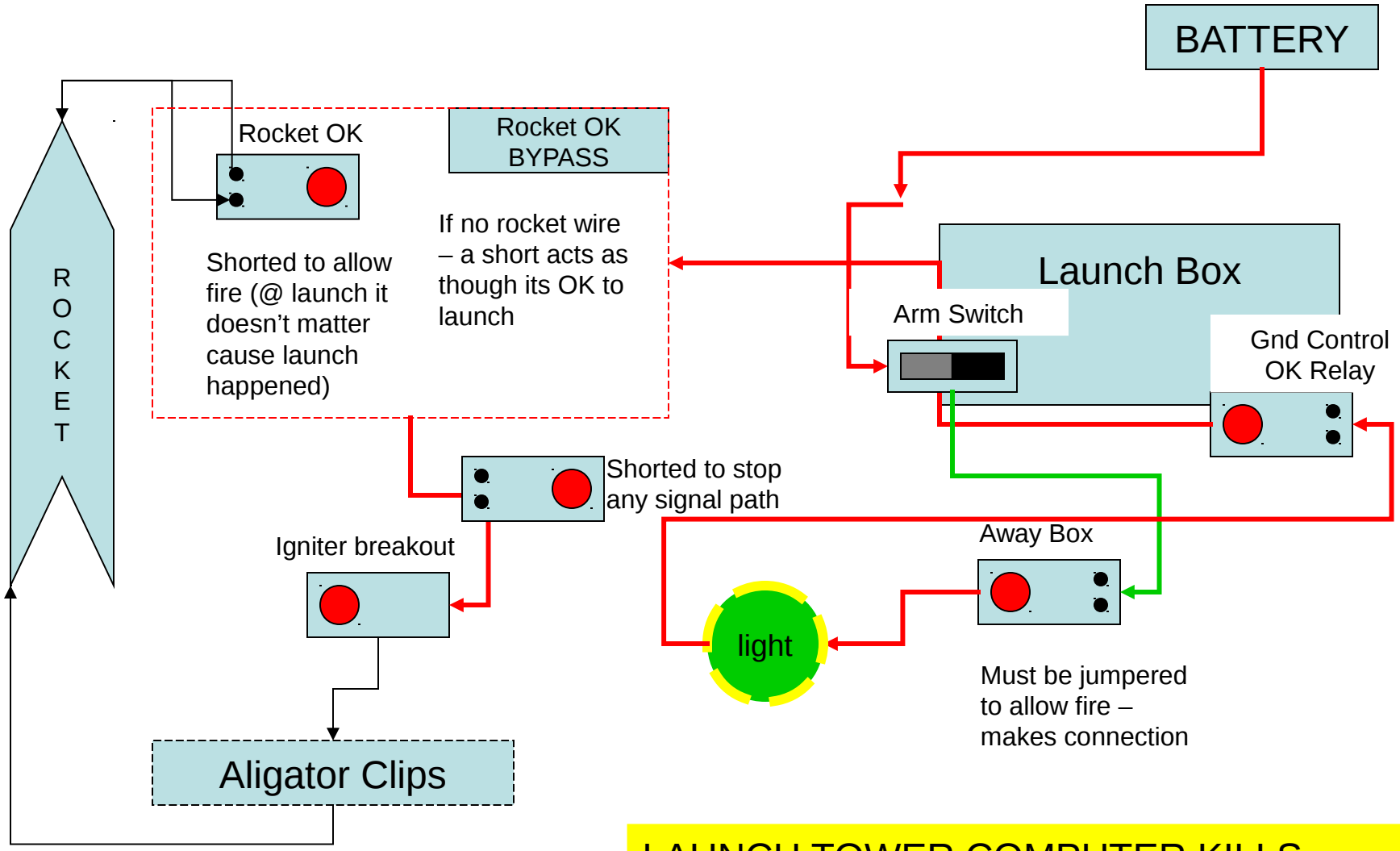


Interlocks - CURRENT



LAUNCH TOWER COMPUTER KILLS POWER TO ROCKET BEFORE LAUNCH

Interlocks - NEW



LAUNCH TOWER COMPUTER KILLS POWER TO ROCKET BEFORE LAUNCH

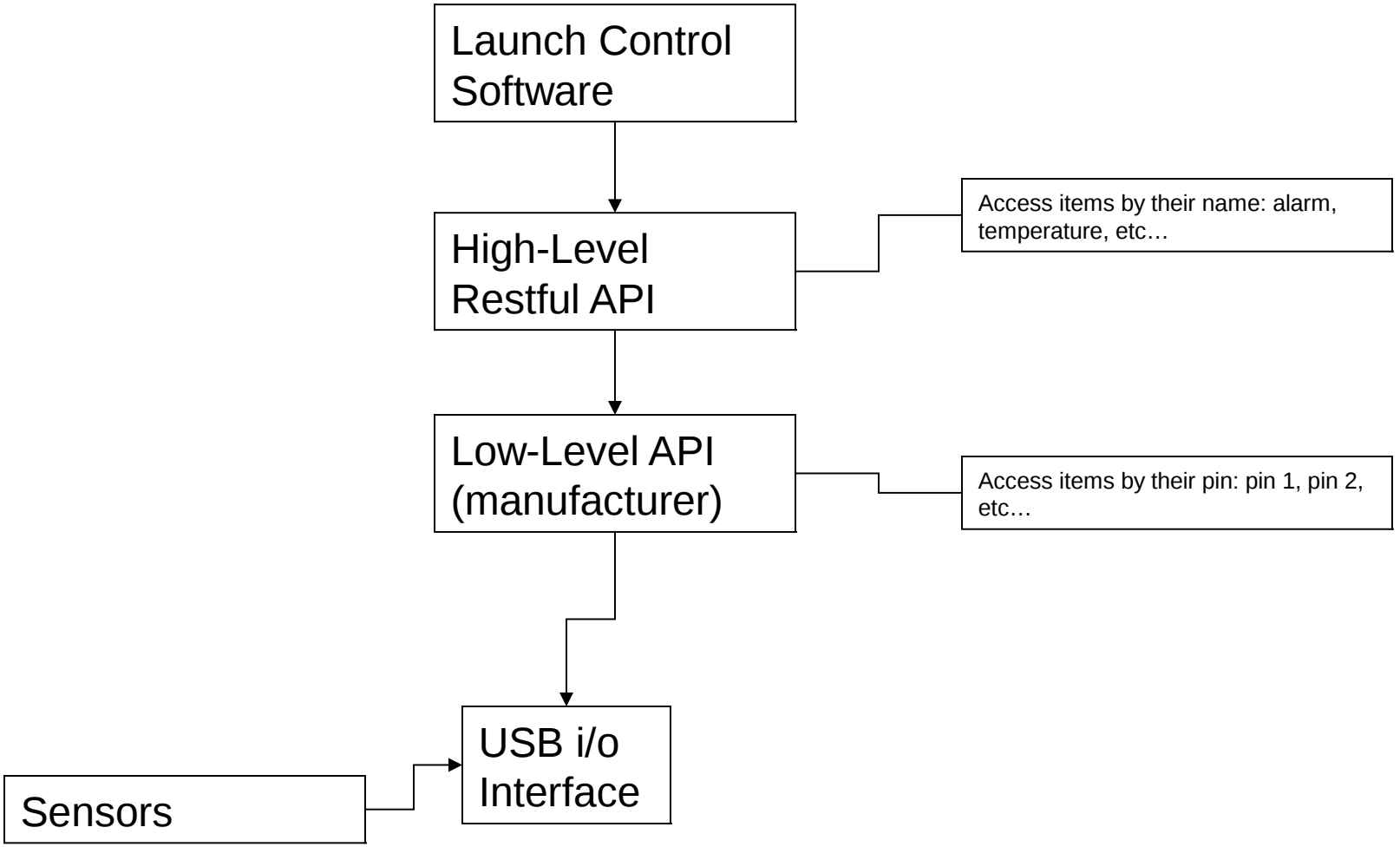
Computer Steps

- Boots Up
- Connects to USB i/o interface
- Starts web server and SSH access
- Starts launch management software
- Starts drivers for webcam access
- Login from ground control (via SSH or web access)
- View program interface for status and ability to launch.

Software Goals

- Take input to begin countdown
- Allow halt/cancel of countdown
- Report on system status
  - Batteries
  - Solar Cell
  - Relays
  - Cameras
- Launch sequence
  - Start video collection – via GPIO to 3rd-party board (and some form of validation back in)
  - Set relays
  - Sound horns
  - Visual alert
  - Send fire signal – This last step must be done again by ground control – NOT automatically in the loop
  - Send signal to disconnect shore power to rocket

Software Structure



Timeline

	May					June				July
	1 - 5	6 - 12	13 - 19	20 - 26	28 - 31	3 - 9	10 - 16	17 - 23	24 - 30	COMPLETE
I/O	Order Interface									
	Order Relays									
	Order Sensors									
		Shipping								
			Assembly							
			Programming	Programming	Programming	Programming				
					Test	Test				
CONTROL INTERFACE					Design Layout					
						Coding	Coding			
								Test		
VIDEO	Research Web Camera									
		Order								
			Shipping							
				Programming	Programming					
					Test	Test				
COOLING						Design				
							Build			
								Test	Test	
SYSTEM TESTING					Design Procedure					
						Test	Test	Test	Test	

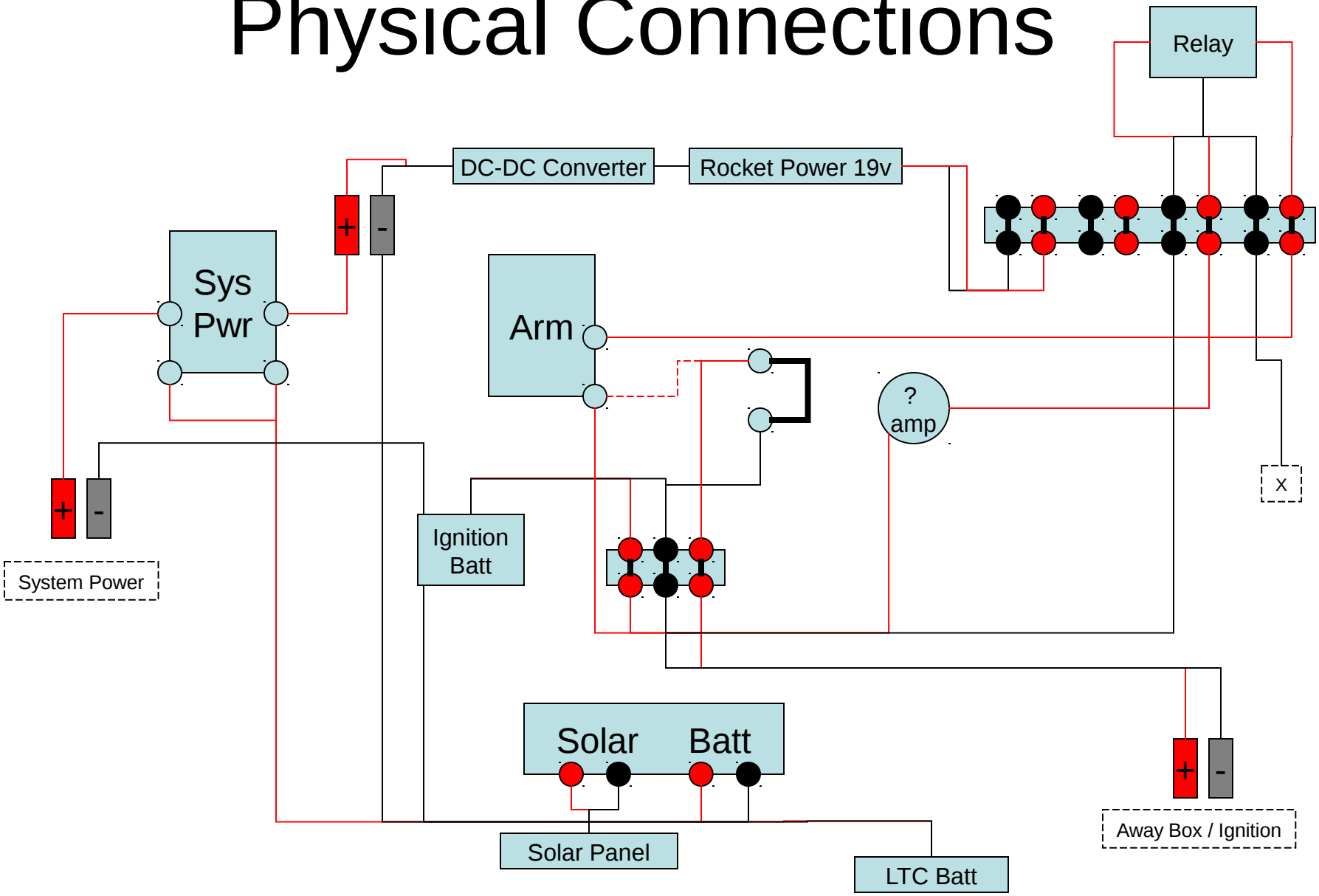
Possible I/O Interfaces

- Phigets
  - Website: http://www.phidgets.com/
  - Pros
    - Lots of libraries in many languages
    - Relays, IO, and webcams all with libraries
    - Lots of sensors with libraries (immersive heat up to 800*C – in the plume)
    - Debian board for integrated development
- Electronics-DIY
  - Website: http://electronics-diy.com/USB_IO_Board.php
  - Pros
    - Super low-cost - $23
    - Access control/logging via PHP
    - 16 individual microcontroller I/O pins
    - 500mA for electronic projects
- Audrino
  - Website: http://www.trossenrobotics.com/p/arduino-duemilanove.aspx

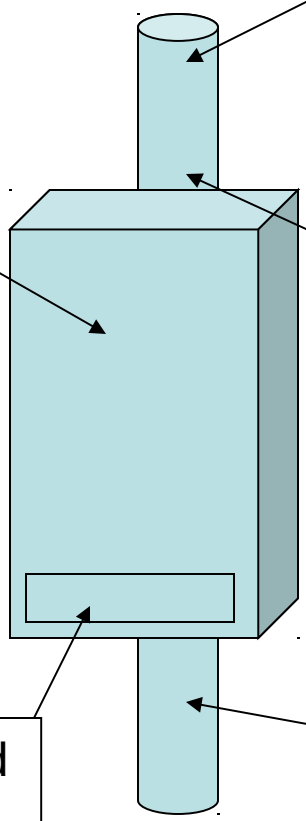
Budget - $500

- USB i/o interface
- Web Cameras
  - Under plume
  - Wider take-off shot
- Sensors
  - Plume Temperature
  - Power status
  - Thermal Status – Electronics
  - Thermal Status – Ambient Air
- Siren
- Lights
- Computer system
  - NetPC with cooling
  - Thermally hardened computer – Ideal Range (30*F - 150*F)
- Cooling
  - Some form of cooling for electronics
- Dust mitigation

Physical Connections



Box Mounting



1.25" diameter steel tubing with heavy galvanization

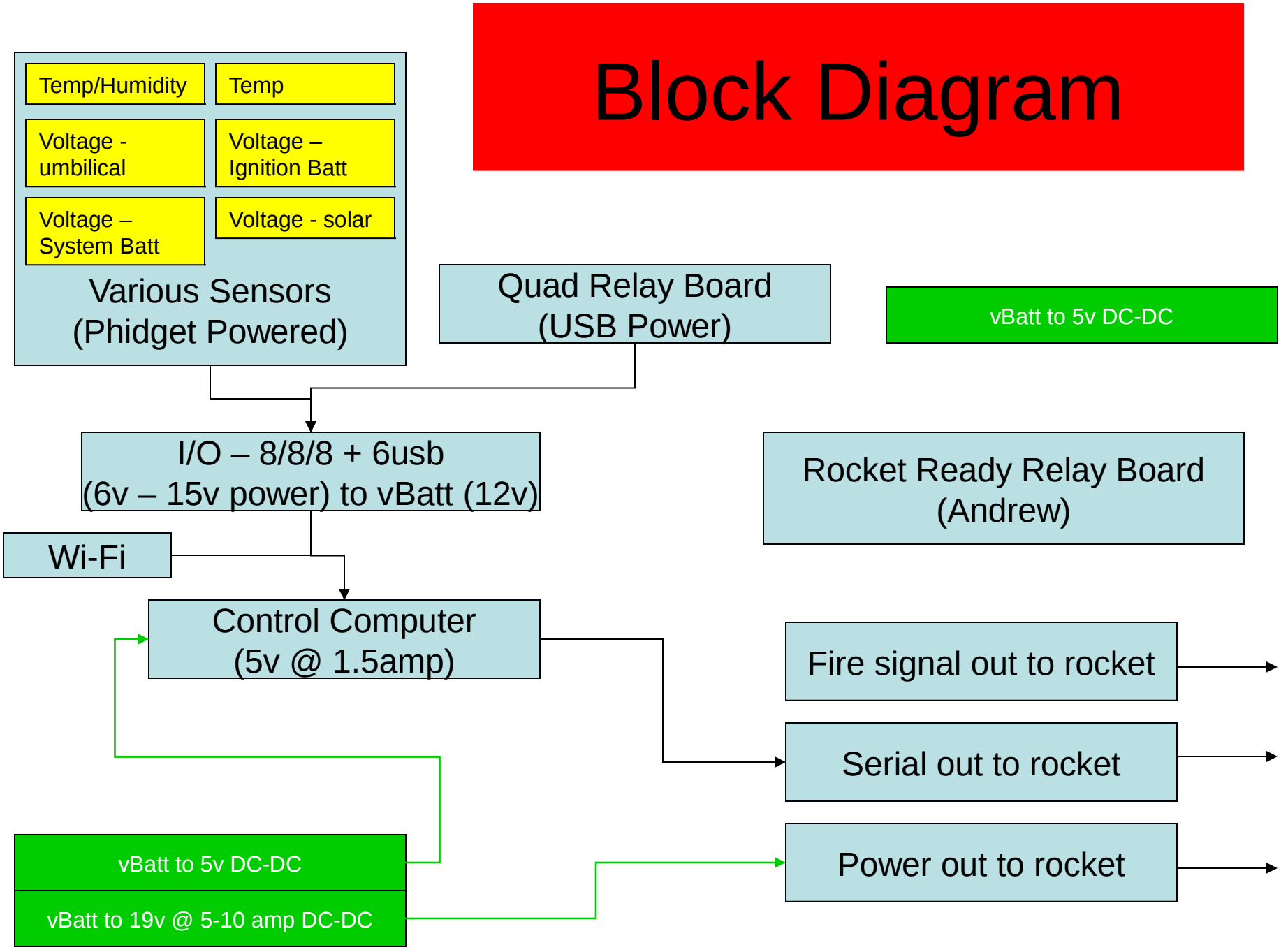
NEMA plastic box with rain resistance and dirt protection.

Mount with U-bolt and bolt saddle so box edge not tight to tube.

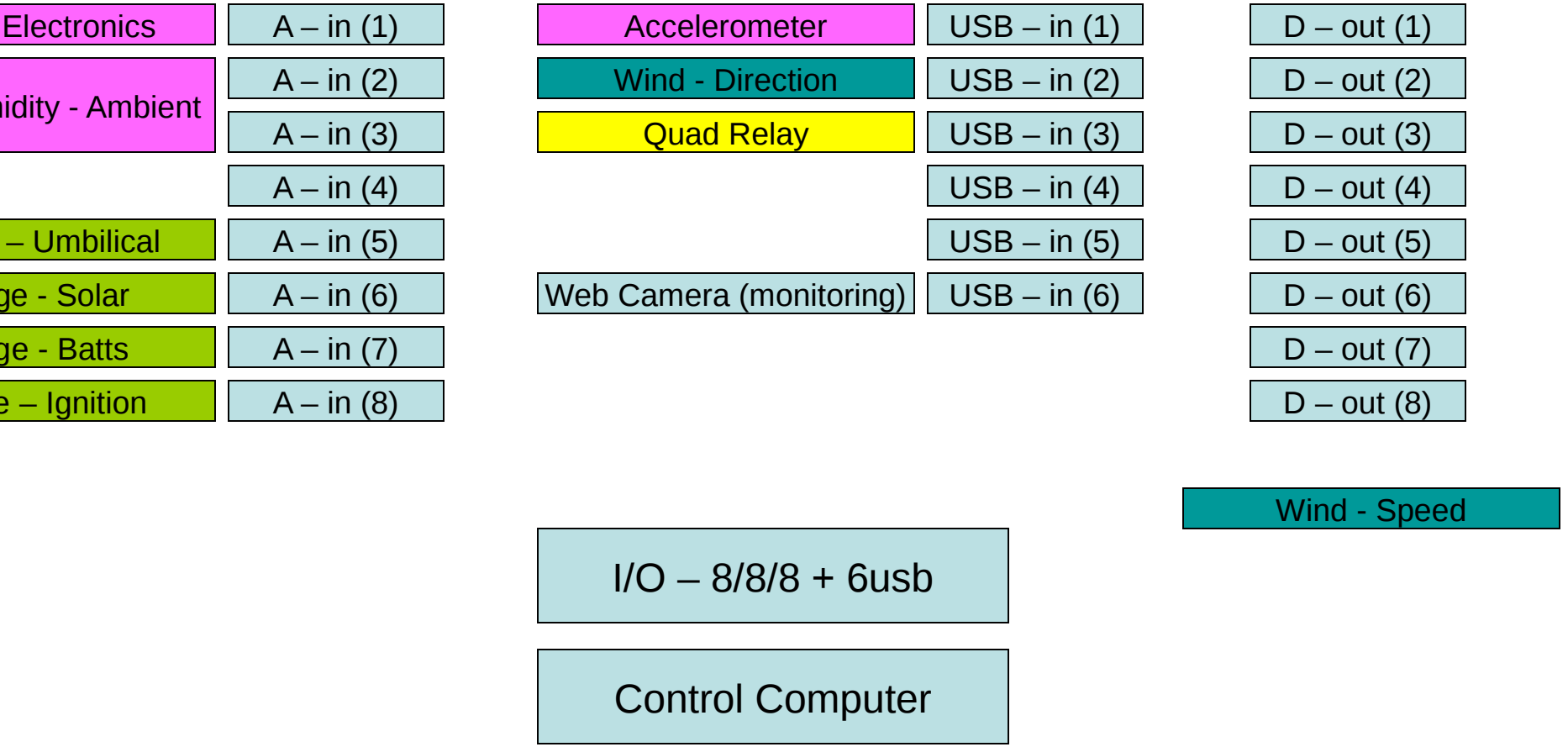
All connections panel mounted through bottom – and through rubber seals (including wi-fi)

Need some form of strain relief on the cables hanging

Block Diagram

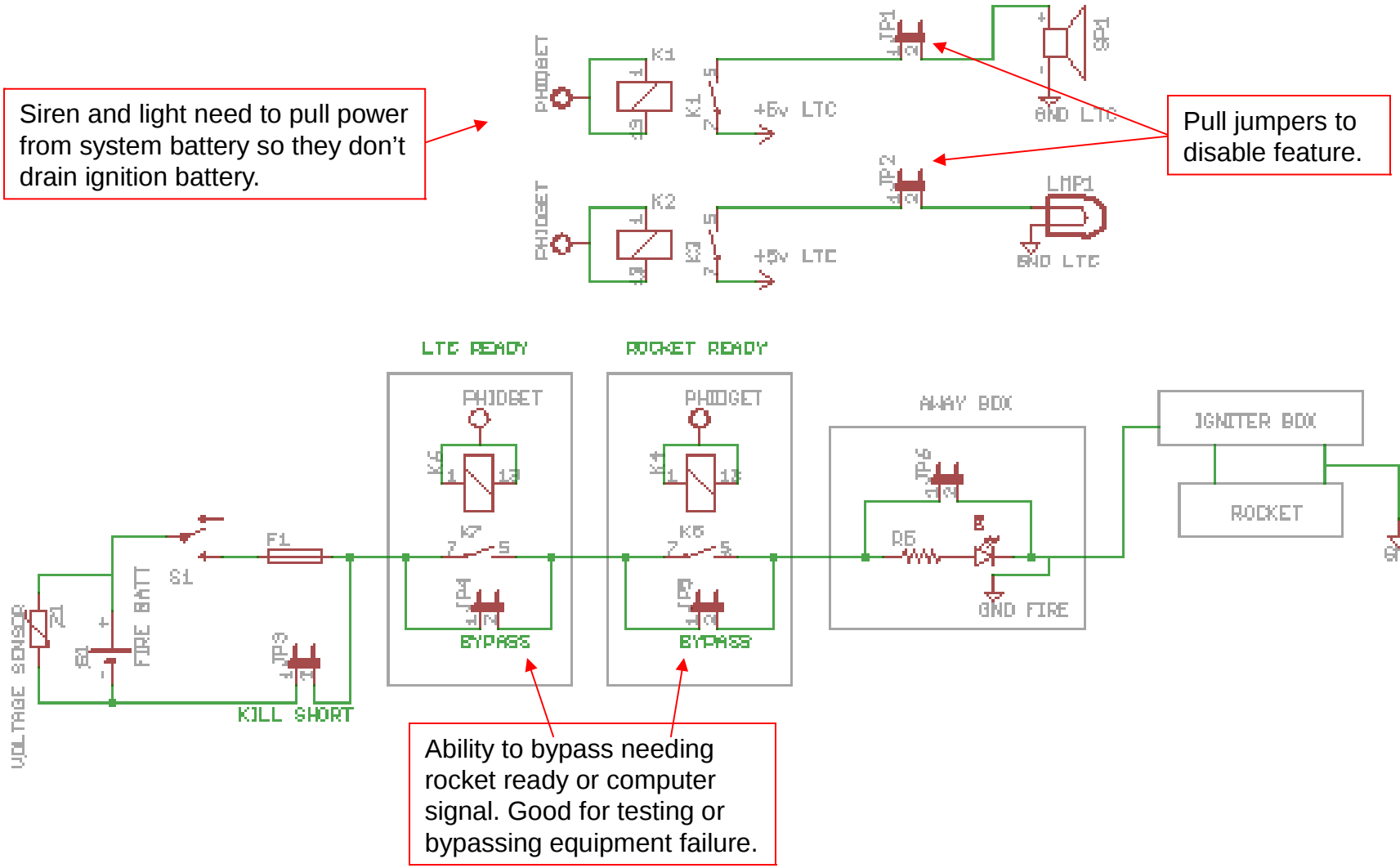


I/O Design - Phidgets



IGNITION PATH & FEATURES

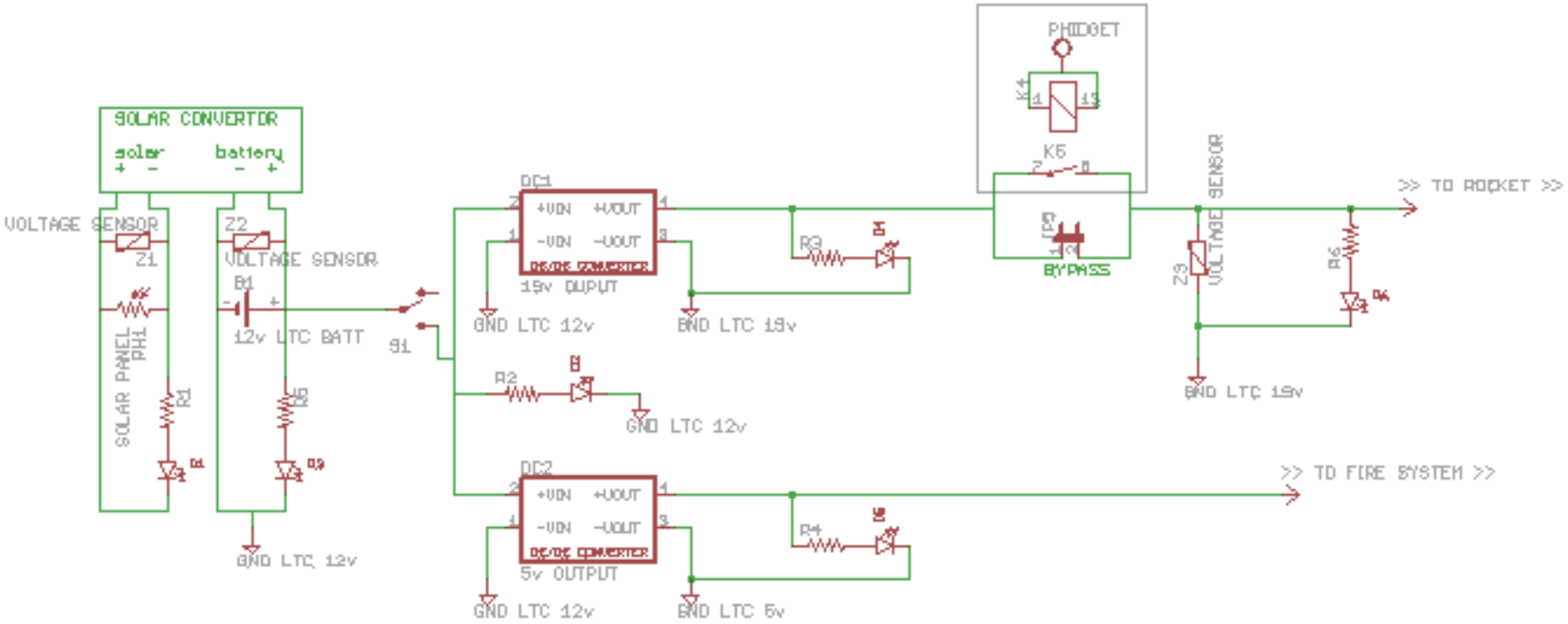
Siren and light need to pull power from system battery so they don't drain ignition battery.



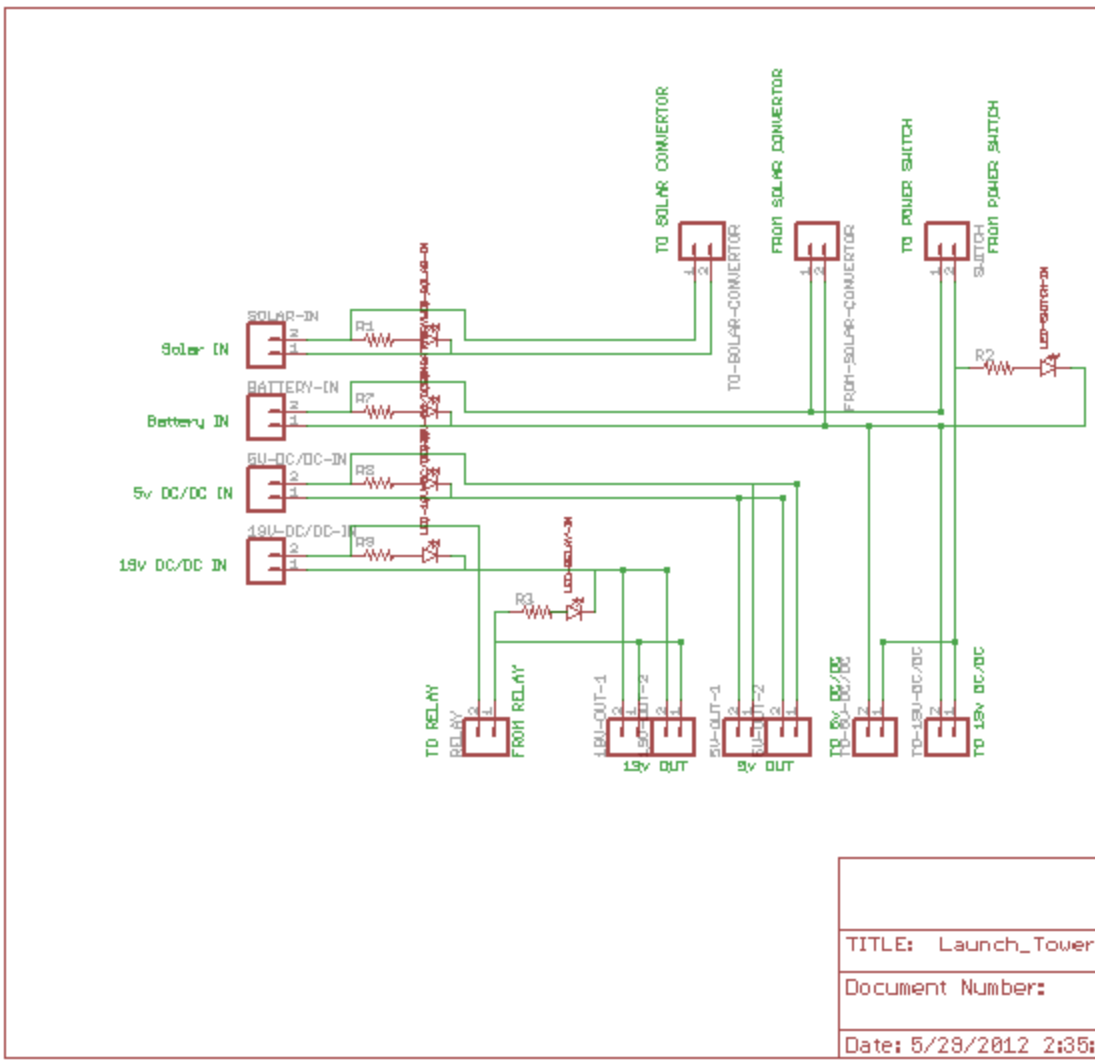
Pull jumpers to disable feature.

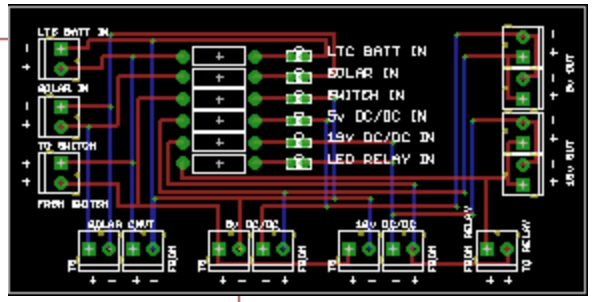
Ability to bypass needing rocket ready or computer signal. Good for testing or bypassing equipment failure.

SYSTEM POWER



Possible Power Interface Card

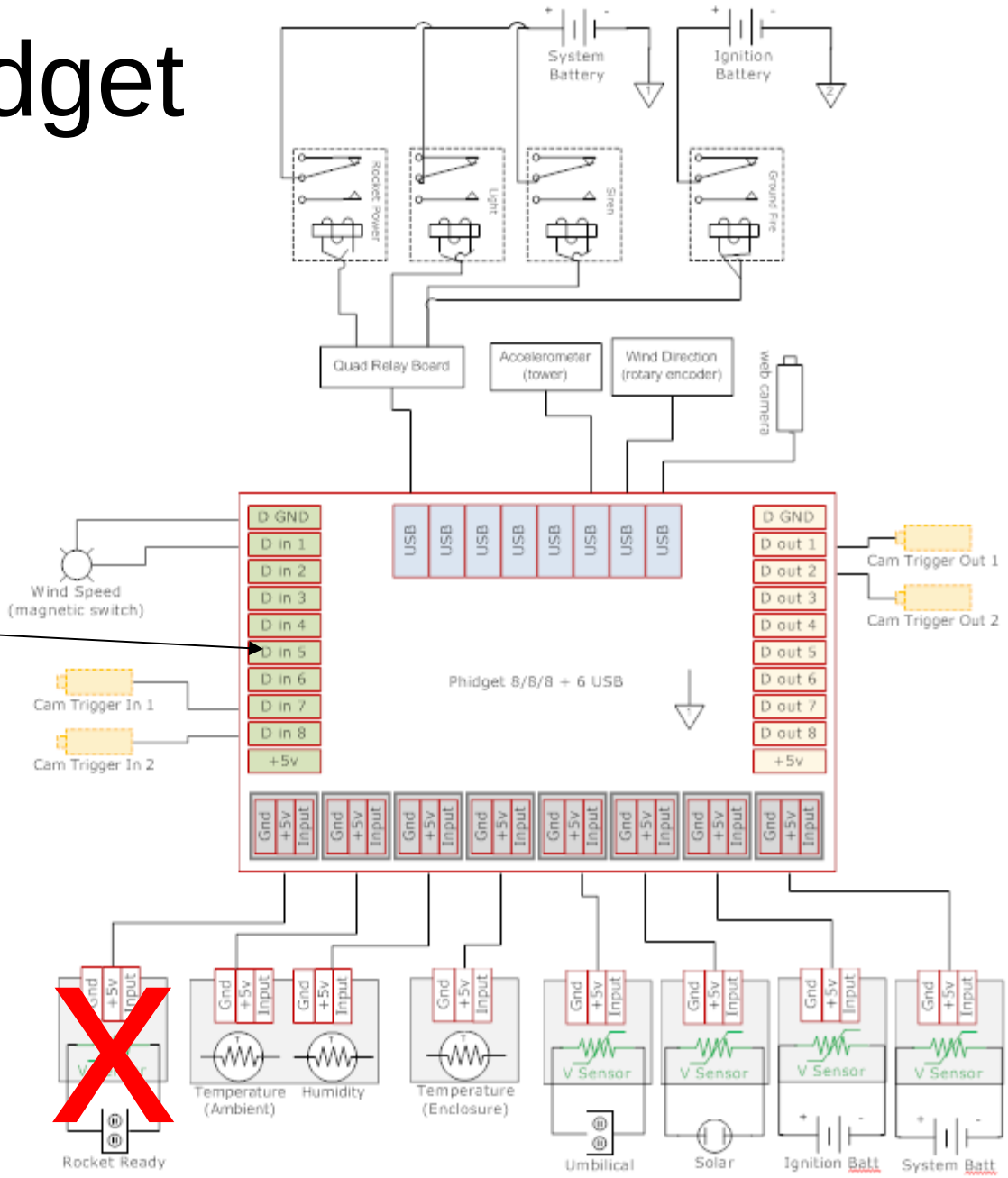


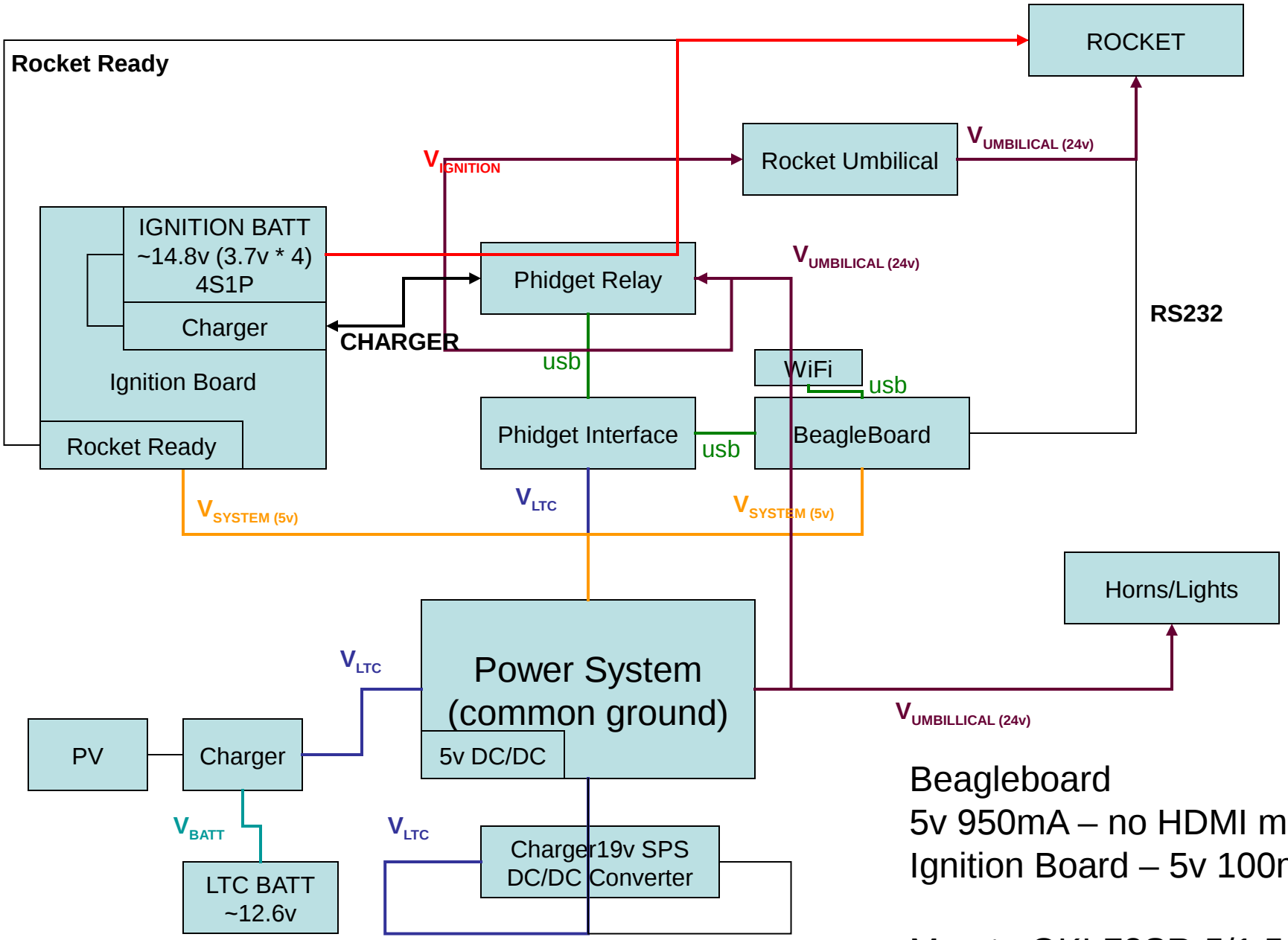


Puts all power connections onto one monitoring card with LEDs so you can see if everything is functional right after hook-up.

Circuit - Phidget

DPDT on rocket ready to Digital i/o





Beagleboard
5v 950mA – no HDMI monitor a
Ignition Board – 5v 100mA (?)

Murata OKI-78SR-5/1.5-W36-C