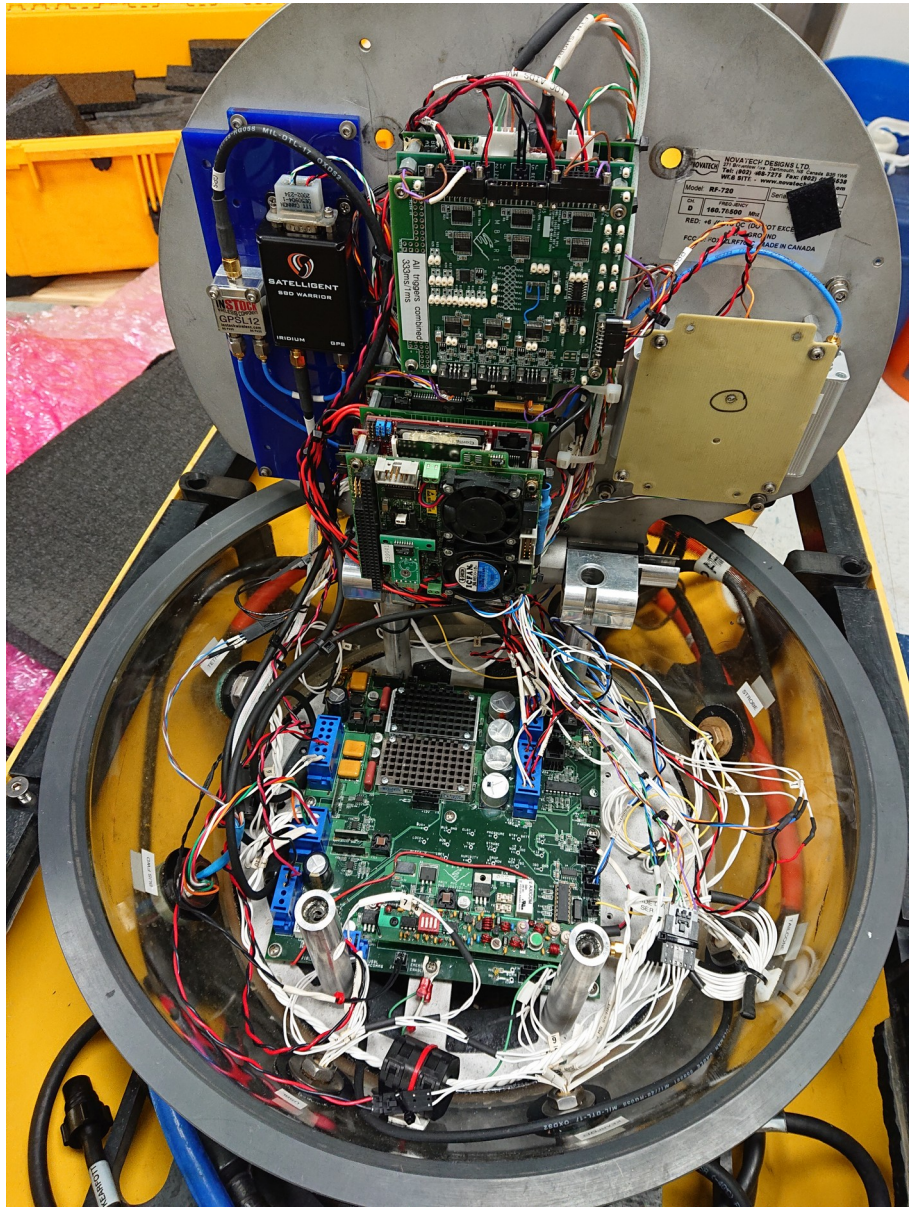
Dorado MVC design

MBARI Monterey Bay Aquarium Research Institute

Current MVC

Power board with PC104 stack

- mix of commercial and custom PC104 cards
- Most commercial cards unavailable
- Drop weight circuit on power board
- All loads always on
- Ratsnest of wires to connect PC104 cards
- Simple and relatively easy to reconfigure
- CPU is 500MHz AMD Geode LX800 (x86)
- Operating System: QNX 4.25



Existing “New” MVC

Design:

- Consists of CPU board and Power board
- Uses Technologic TS4100 SOM with i.MX6UL (528MHz/512MB RAM)
- 8 native UARTs plus 8 Exar SPI UARTs, all isolated
- Flexible UART output, each can be RS232/485
- Freewave, Iridium and GPS mounted straight to CPU board
- USB and 2x Ethernet

Issues:

- Monolithic design: there is an issue with one component, whole board needs changing/fixing
- CPU SOM tightly integrated with board, makes switching to new CPU SOM hard
- No consistent load control architecture, different loads get turned on by different means (GPIOs, I2C IO expanders)
- No consistent isolation scheme, seemingly all outputs are always isolated which burns tons of power (board uses 10W idling)
- Uses custom supercap holdover that works against the supercap management the SOM provides
- Current board rev full of countless blue wires and more issues constantly popping up

New approach

Learn from existing projects and adjust as necessary!

- Load control architecture works well on LRAUV, let's go with that but adjust some of the drawbacks
  - SPI requires a chip select for every LCB or complicated multiplexing, maybe go with I2C
  - Don't need SPI speed if UART IC moves off the LCB, however, this requires more pins at the LCB interface
  - Relays sub-optimal for loads >5A, try using solid state isolation scheme
  - Need more IO pins than just one (PPS, trigger)
  - Form factor and connector is not suitable, need more pins and wide design not optimal for mounting in sphere

- Ground fault sensing can be adapted from LRAUV
  - Try to get away with just one GF test voltage instead of >5, make it negative so it works on every channel
  - Need to use better op-amp where bias currents are not larger than GF currents to measure
  - !!consider RS232 using -12V line voltage, maybe use different metal for electrode (zinc)

- Contemplate using CPLD emergency circuit from LRAUV
  - Depends on adaptability to different requirements and if CPLD still has advantage over familiar uC designs !!!maybe we can get away without any firmware

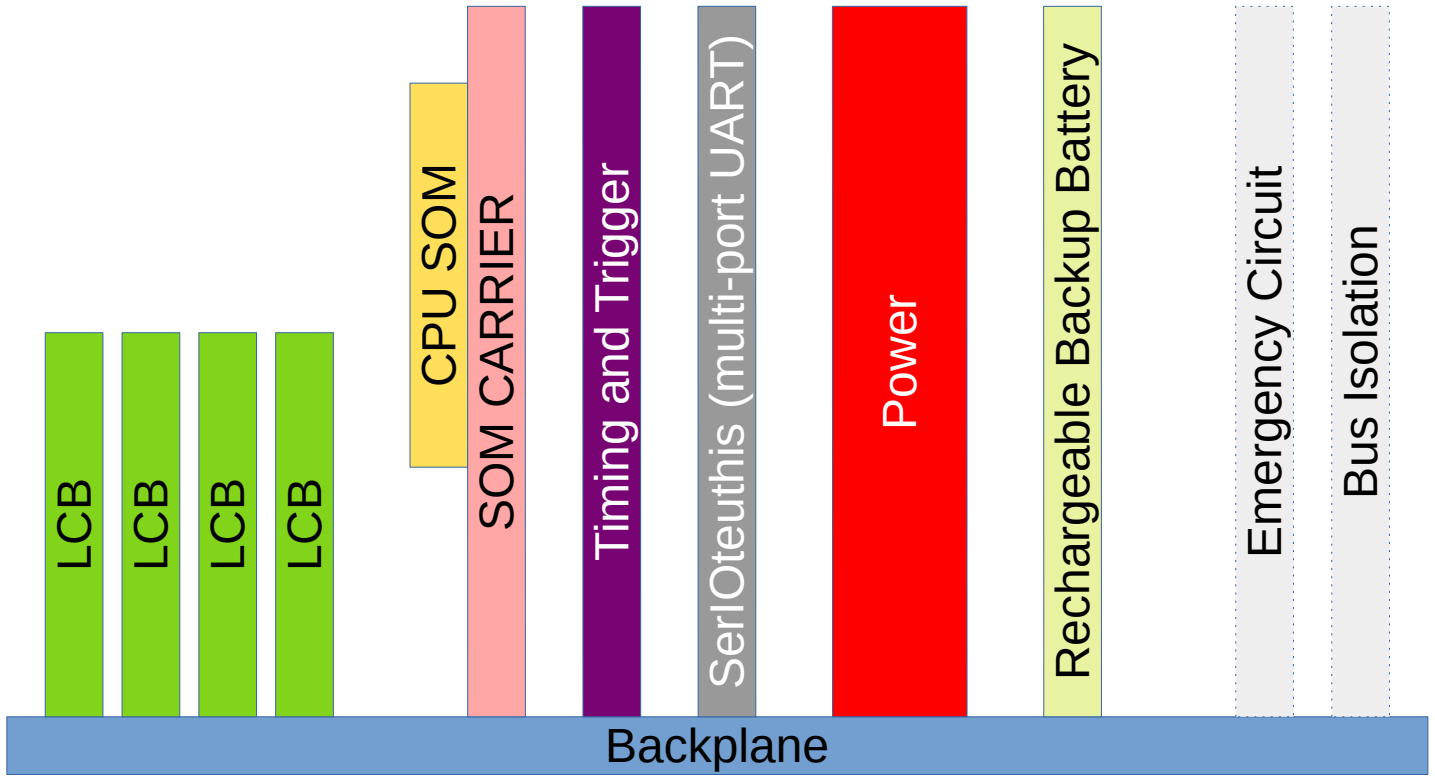
General architecture and design concept

- Instead of trying to get one enormous project right the first try: Move fast, fail fast.
- Instead of one monolithic board, try the modular approach:

  Cons:
  - Reliability? This seems to have been the prevalent concept of the past
  - More connectors, more points of failure

  Pros:
  - Sub-component can be redesigned independently
  - Much easier to fix in the field even by less experienced operators
  - Much more flexible to incorporate evolving hardware needs

General architecture and design concept



LCB thoughts

- Prototype exists
- Basic tasks:
  - Turn on/off loads, isolation
  - Allow isolation of UART outputs and IOs
  - Limit current, allow measuring output voltage and current
  - Can be configured for different voltages and IO modes

- Notes
  - Need pins for I2C address strapping, 4 should be enough (16 LCBs), maybe 5
  - Need to find proper connector, want to be able to switch up to 10A for 24V and bus
  - Supported output voltages: 5, 12, 24, bus, the few things that need 3.3V can have reg on LCB
  - Allow multiple LCB circuits that just share a common interface spec (don’t try to make everything work just by changing configuration)
  - How to handle devices that have a common power/signal ground? When joining them at LCB we get loops. Consider perm iso.
  - RTS/CTS for every UART? (needed for freewave and RS485 instruments)
  - Keep distinction between isolate and turn on/off even though we don’t have relays that wear out, it allows to do GF scan on devices that are turned off
  - !!!consider multiple channels per LCB

CPU board thoughts

- Keep CPU connections generic so CPU board can be replaced easily
  - Keep direct GPIO connections to minimum (use expanders that live on LCBs/backplane/etc)
  - Just use a few bus connections to interact with rest of system (SPI plus a few CS, I2C, USB routed on board?)
  - Don’t rely on CPU SOM having tons of UARTs, use multi-port UART instead

- GPIOs that should go directly into CPU board:
  - 1PPS
  - Trigger?

- Having less IO requirements allows to swap in almost any SBC by just making a new SOM adapter board

- Current CPU SOM Technologic TS4100 with i.MX6UL
  - Need to investigate if still best solution
  - Some necessary IOs are routed through an FPGA which is painful to deal with and makes debugging more diffiucult
  - Has supercap holdover but we have a pre-production release that might not have all the necessary features to make this work correctly
  -
  - !!!Look at comexpress standard

UART board thoughts

- Eventually use Serioteuthis multi-port SPI UART
- Until then, can use multiple FTDI quad UARTs connected by USB
  - not ideal for power but would be temporary
  - Will require board design but no software

- CPU board could have different number of native UARTs so there is a need to mux the UARTs depending on which ones come from the CPU and which from the UART board
  - Probably done on back plane
  - Could route all UARTs from CPU to UART board first for muxing → requires tons of pins

- Pin count will be significant: 12-16 UARTS, 4 pins each (TX, RX, RTS, CTS)
- Maybe not all LCB ports have RTS/CTS?

Rechargeable Backup Battery

- Allows CPU to keep running when bus is dead
  - Keep beacons alive when batteries empty after mission
  - Allows CPU board to shut down cleanly if power turns off
  - Don’t need to open housing if empty, just recharge
- Recharge anytime power is on bus (so would get charged by main batteries if deployed not all the way full)
- 300Wh is probably enough (3x Inspired Energy packs)
- Allows disconnecting from bus for GF scan
- Needs charge controller
- Uses a lot of space inside MVC
- What to power from this battery:
  - CPU board
  - Load control system?
  - Instruments?
  - Needs eval of what makes sense, thinking at least Iridium and Freewave modem should stay on so comms stay up

Bus Isolation

- Necessary if we want to do GF scans, otherwise GF on bus could not be detected
- Can use SolidState ISO circuit
- Probably should just be a LCB, needs to be able to handle 20A

- Tether isolation:
  - Should self isolate if no power is provided from tether
  - Could also be a LCB, so one could monitor voltage/current
  - While we are here, tether still has 2 free wires, could use for serial console (but would require some thought since we would need 3 wires or have to use a RS485 scheme or similar

List of instruments run through LCB system

Instrument	Voltage	Comms	Notes	
Freewave	3.3/5	RS232 + RTS/CTS		
GPS/Iridium Xeos	12/24	RS232	Second RS232 for GPS	
Compass VN100	5	RS232	No need for iso	
INS	24	Ethernet		
AvTrak	24	RS232		
Paro	12	RS232		
CTD	12	RS232		
DVL	bus	RS232		
Gulper	12	RS485		
Tailcone	bus	RS485/CAN/RS232		
Emergency box	24			
Cell modem	?	Ethernet?	Not currently spec'd	
Acoustic modem	?	RS232	Not currently spec'd	

Emergency Circuit

Problem: we have many beacons/items that all need at least power (some need serial)

- Argos (Xeos)
- VHF (Xeos)
- Iridium/GPS (Xeos)
- Pressure switch
- Burn wire
- External Backup battery(?)
- Reed switch

Putting them all on separate penetrators doesn’t work
- Spider cable → have to throw away whole assembly when one connector is broken
- Junction box → have messy oil filled housing or potted housing

External primary battery
- Will keep beacons alive if MVC is flooded
- Needs extra pressure housing for batteries
- Not sure if worth the effort but makes replacing backup batteries easy

Emergency circuit truth table

When should emergency beacon be on

Vehicle power	Reed switch/magnet	Pressure switch		VHF	Iridium	Argos
OFF	OFF	OFF		OFF	OFF	OFF
OFF	OFF	ON		OFF	OFF	OFF
OFF	ON	OFF		OFF	OFF	OFF
OFF	ON	ON		ON	ON	ON
ON	OFF	OFF		OFF	ON	OFF
ON	OFF	ON		OFF	ON	OFF
ON	ON	OFF		OFF	ON	ON
ON	ON	ON		ON	ON	ON

Note: Iridium beacon is also our primary GPS source

What to change about LRAUV
More modular motherboard
LC UARTs
More GPIO
More modern linux
Add datalogger

Things to think about
Backseat can talk to instruments directly

Power systems

Power supplies:

- 24V
  - always on (provides power to emergency circuits so should not be controllable)

- 12V
  - only non-essential instruments on there
  - can be turned off by I2C (CPU)
  - turned off if in emergency mode and backup is low

- 5V
  - powers CPU so not switchable by CPU
  - turned off if in emergency mode and backup is low (survival mode)
  - Should always be on when we are not in survival mode

- 3.3V
  - provides power for emergency logic (and other logic stuff across the backplane)
  - Can be turned off in survival mode, just have to ensure emergency beacons stay on when it goes away

Power modes

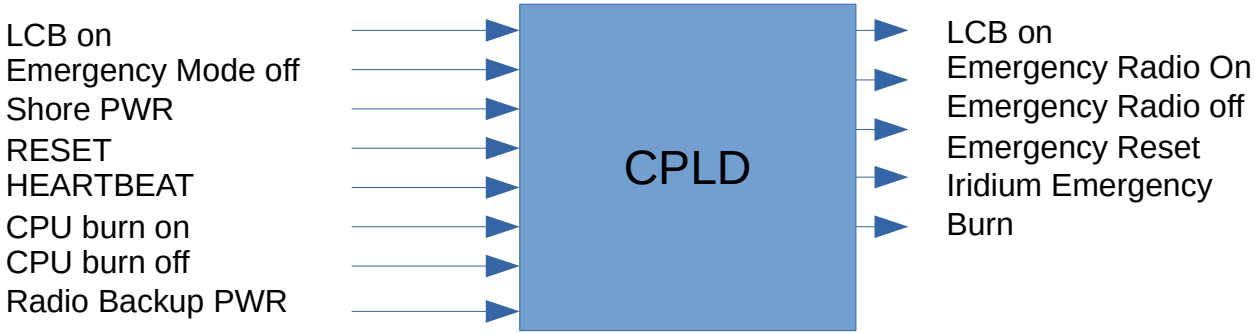
- bus on (from tether or main battery):
  - 5V on to power CPU
  - 24V and 3.3V on to power emergency logic and beacons
  - 12V can be controlled by CPU as needed

- bus off, magnet in:
  - vehicle is off, no power to anything

- bus off, magnet out, backup battery full:
  - 5V on to power CPU, CPU still controls vehicle
  - 24V and 3.3V on to power emergency logic and beacons
  - CPU manages loads based on situation

- bus off, magnet out, backup low (survival mode):
  - 12V, 5V off, CPU down and not in control anymore
  - 3.3V off, emergency logic down, can't get out of this mode unless somebody provided bus power
  - 24V on to power emergency beacons
    (how to manage burn wire in this mode? Maybe have separate 3.3V for emergency logic)

LRAUV Emergency CPLD



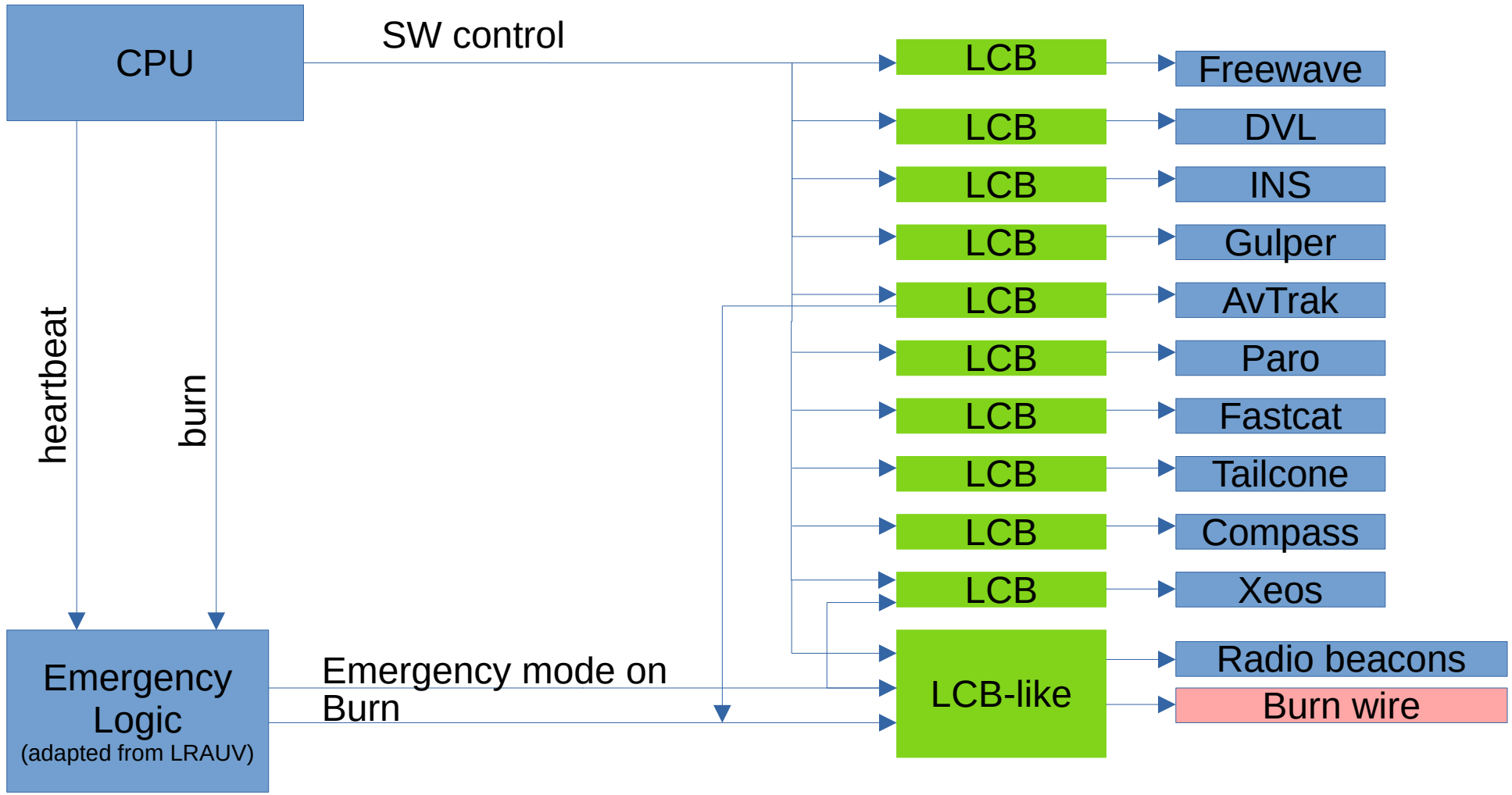
LCB on	Enable LCB backplane power	Don't have sep. LCB backplane, could control power to LCBs though
Emergency Mode off	Take CPLD out of Emerg Mode	
Shore PWR	We have external power	Not used
RESET	Global reset input	
heartbeat	From CPU to keep running in normal mode	
CPU burn on/off	Burnwire control from CPU	
Radio Backup PWR	Bypass LCB to turn Radio beacons on	Will not use it this way but can use this output as a generic emergency mode indicator

LCB on	Enable LCB backplane power	Don't have sep. LCB backplane, could control power to LCBs though
Emergency Radio on	Enable Radio LCB bypass	See above
Emergency Radio off	Disable Radio LCB bypass	See above
Emergency Reset	Global Reset if hearbeat fails	
Iridium Emergency	Sets NAL9602 into emergency mode	Don't have that, ours can fall into beacon mode automatically if they stop receiving cmds
Burn	Burnwire output	

Emergency enabled LCBs

- LRAUV uses bypass power to turn on radio beacons when in emergency mode and LCBs are off, this power comes straight from the MoBo
- I think a better approach is having LCBs that have an emergency input when the load is required for an emergency situation (Iridium beacon, VHF, Argos, etc)
- In emergency mode, LCB deisolatetes and turns on load in hardware
- LCB uC should probably be put in reset or depowered in this mode
- for burnwire and beacons we probably have a non-standard LCB that has the same LCB interface but isn’t just a data passthrough, it will generate burn wire voltage and read things like the pressure switch to turn on/off beacons as necessary
- CPU should be in charge of this LCB when in normal mode which allows to isolate beacons to do GF scans, however, when in emergency mode, these loads get turned on right away

LCB Emergency Control overview



Radio beacon control

- Originally thought of complex logic in hardware to decide when beacons are on (have to consider state of magnet, pressure switch, backup battery status, etc.)
  One such solution is shown here:
  smb://atlas.shore.mbari.org/projectlibrary/368000_DMO_DORADO/Project.Documents/2024_MVC_Design/dorado_emergency_system_overview
- Now I'm thinking we should simplify like this:
  - Normal mode (CPU running): beacons are solely controlled by SW which allows to turn beacons on and off on deck easily and status of magnet or pressure switch doesn't matter
  - Emergency mode: 2 options:
    - Always on (probably don't have anything that falls in that category, maybe if we add a modem again)
    - Only on at surface (using pressure switch to decide when on surface)
  - This makes the hardware much simpler but requires change of thinking what the magnet and pressure switch does. We wouldn't use the pressure switch anymore to turn the VHF on and off when having 2 vehicles on deck and would probably need a check-out test if the pressure switch is working

Ideas, comments:

Argos charged by solar?
Think about emergency mode when batteries are still on.
Make 24V supply redundant
Read back some of the actuated signals (dwout, 24V on beacons)
Think of pressure switch redundancy

Backplane discussion:
Look at single pair ethernet, can

Project Hyperion

New working title for this. Hyperion is Tethys' brother.
Hyperion's children were Helios, Eos and Selene
(which make awesome vehicle names), unfortunately he only had 3

Progress Update

- use existing interface spec from last meetings and start implementing prototypes of the essential modules
- this allows identifying issues that only show up when thinking about all the small details
- then go back to the back plane and integrate these differences and make all modules compatible with each other
- Essential module prototype list:
    - generic LCB (95% finished)
    - CPU SOM carrier (finished, needs review)
    - emergency LCB (finished, needs review)
    - power supply (no schematic started yet, working on fundamental design)
    - bus/tether entry (no schematic yet, but fundamental design exists)
    - battery backup controller (some basic ideas exist)

LCB interface refinement

Things to keep in mind:
- keep firmware the same for all load controllers → keep LCB SW interface generic

Specs:
- every load controller has 2 switchable power channels
- isolate mode will isolate all outputs
- emergency LCB will turn themselves on as needed (no SW involved)
- every LCB can monitor 2 voltages and 2 currents
    - they don’t need to be for the power channels
    - can implement analog LCB using this scheme
- every LCB has 2 discrete outputs and inputs
    - not all might be enabled in hardware on all LCBs
    - LCB will have a matrix where different signals can be connected to the actual out/inputs, planning on having 4 inputs and 4 outputs

- only one RS232 interface per LCB

LCB interface refinement (cont.)

Some changes to LCB interface:
- add shore_pwr and battery_flag (this allows making bus/tether-entry, power supply and backup battery controllers LCBs too)
- maybe remove DIVE flag (only really needed for emergency LCB, when CPU is running we can determine if we are under from pressure sensor, maybe there is a need to turn something on by HW when diving?)

Questions:
- can multiple LRAUV components access/control the same LCB?
  (this might allow us to separate drivers for beacons from the drop weight even though both systems hang on one emergency LCB, having just one Emergency component would also make sense)
- any concerns with moving some board level GPIO lines to LCB IOs?
  (e.g. Dwsense and dw_buf_out could be handled as IO on the emergency LCB)

LCB interface refinement (cont.)

Ideas for next iteration:

- For now we are planning to use I2C for LCBs where every LCB has the same address and we use muxes to address a particular LCBs. If we have muxed I2C buses, we could put I2C power and/or energy monitors on them which then could be read out directly by their Linux drivers
- Problem with that is that the drivers are not hotplug compatible, so we would most likely have to write Our own drivers, which is easy but time consuming
- If we are writing our own drivers, we can use some better energy monitors that are not supported by Linux yet and support things like fuel gauging. Values could be read by just reading a file in sysfs

Power considerations

Using hierarchical power supply structure:
- 12V converter sourced by 24V output, 5V converter sourced by 12V output, etc.
- will improve efficiency
- would mean that if higher level converter dies, all children go out too
- could think about smart switching

Thank you!

Questions?

www.mbari.org

MBARI_news MBARInews MBARIvideo