LV3.1 Electromechanical Recovery System (ERS) PCBA

Overview

This is the PCB that will control the recovery system of LV3. The basic function involves releasing the parachute, and verifying and actuating the separation of the nose cone.

Requirements

- Actuate nose cone separation using 12VDC Motor.
- Actuate a 12VDC Linear Actuator to release the parachute. This will include an H-Bridge control chip.
- Supply power to a STM32F0 which will receive I/O and make decisions
- Receive two incoming signals from main avionics
    - Release nose cone
    - Release parachute
- Read data from BPR-301 Light sensors to verify nose cone separation
- Run off LiPo batteries that will charge from the umbilical power

Parts

Not including resistors, capacitors, or inductors.

Description	Man. Part No.	Quantity	Distributer	Dis Part No.	Data Sheet
Light Sensor	BPR-301	2	Digikey	BPR-301-ND	http://americanbrightled.com/pdffiles/infrared/BPR-301.pdf

Linear Actuator	P16-50-64-12-P	1	N/A	N/A	https://s3.amazonaws.com/actuonix/Actuonix+P16+Datasheet.pdf
DC Motor	TS37GB60-BL362 LC5	1	eBay	N/A	https://www.ebay.com/itm/TSINY-37MM-12V-24V-DC-Geared-Brushless-Motor-With-Square-Wave-Output-Reversible/223514021185?hash=item340a794d41:m:mJEkE4EMU3dM-d8wEyesypA
3.3V SPS	TPS630702RNM R	1	Digikey	296-TPS630702RNMRCT-ND	http://www.ti.com/lit/ds/symlink/tps63070.pdf
LiPo Charger	MP26123DR-LF-Z	1	Digikey	1589-1609-1-ND	https://www.monolithicpower.com/en/documentview/productdocument/index/version/2/document_type/Datasheet/lang/en/sku/MP26123/document_id/1398
STM32F0	N/A	1	N/A	N/A	N/A
H-Bridge for the LA	BD62120AEFJ-E2	1	Digikey	BD62120AEFJ-E2CT-ND	https://d1d2qsbl8m0m72.cloudfront.net/en/products/databook/datasheet/ic/motor/dc/bd62120aefj-e.pdf

Buzzer	SMT-0440-S-R	1	Digikey	668-1488-1-ND	http://www.puiaudio.com/pdf/SMT-0440-S-R.pdf
NTC Thermistor, 0805 Series	NHQ103B375T10	1	Digikey	235-1109-1-ND	https://www.digikey.com/product-detail/en/amphenol-advanced-sensors/NHQ103B375T10/235-1109-1-ND/374827
OptoIsolator	H11F1SR2M	2	Digikey	H11F1SR2MCT-ND	https://www.onsemi.com/pub/Collateral/H11F3M-D.pdf
LED		3	Digikey		https://d1d2qsbl8m0m72.cloudfront.net/en/products/databook/datasheet/opto/led/chip_mono/sml-d12x1-e.pdf

System Level Design Notes

Designed Schematic, need to design board and assemble a proto board.

SubSystem Design Notes

States

We have 4 states before launch. Arm/Not Arm is controlled by a P-CH Mosfet that essentially kills power to the board, allowing manual locking.

Prelaunch States

Not Locked	Nose cone off
Locked	Nose cone on
Armed	DEFAULT. Waiting signal from Avionics, and kill the signal from locking switch. Do we want to have the arm switch interrupt the inputs physically
Not Armed	Board is off.

MicroController

- Added N-Channel MOSFET to turn sensors on and off for power conservation.

Pin Descriptions

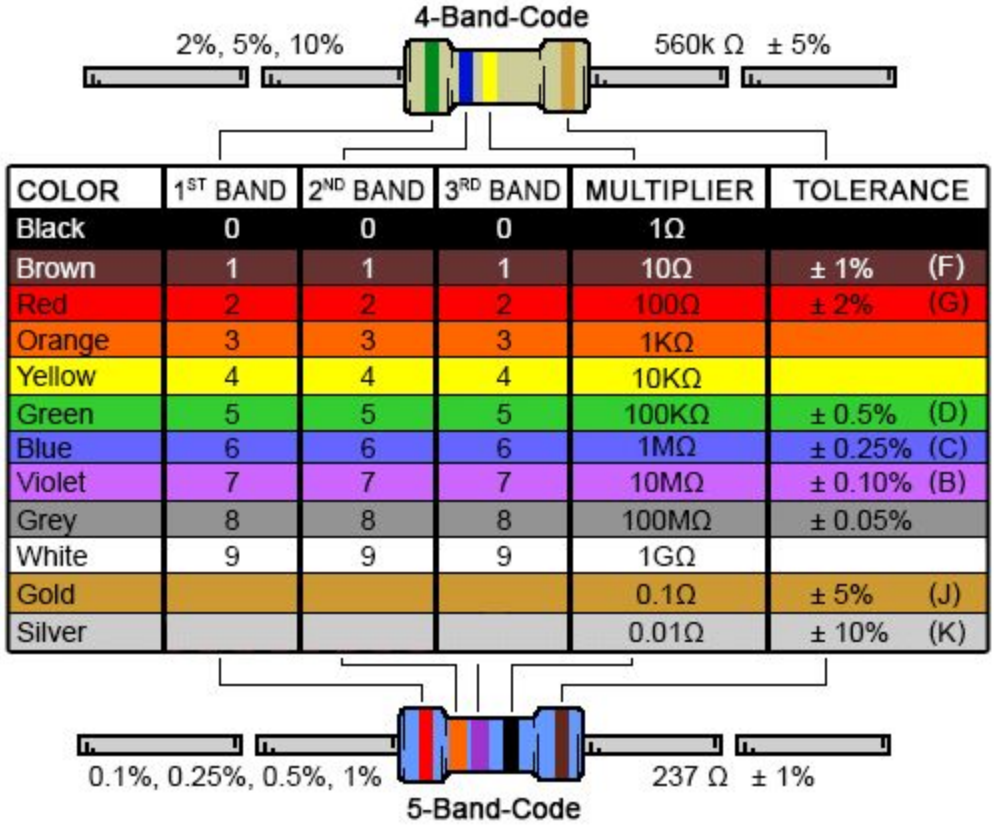
Purpose	**Pin Number**	**Schematic Name**	**STM32 Pin Name**	**Pin Mode**
DC motor speed	6	DCM_SPEED	PA0	PWM IN
Analog Input for potentiometer to read LA position	7	LA_POS	PA1	AI
USART protocol Tx	8	USART2_TX	PA2	AF
Chute Signal from OptoIsolator	9	ISO_CHUTE	PA3	DI
Drogue Signal from OptoIso	10	ISO_DROGUE	PA4	DI
Direction of DC Motor	11	DCM_DIR	PA5	DO

DC Motor PWM output	12	DCM_PWM	PA6	PWM OUT
Turn DC Motor on	13	DCM_ON	PA7	DO
Status LED	14	LED	PB0	DO
Read Battery Voltage	15	BATT_READ	PB1	AI
HBridge for LA Input 1	18	LA_IN1	PA8	DO
Hbridge Input 2	19	LA_IN2	PA9	DO
Free	20			
CAN Rx	21	N/A	PA11	CAN
CAN TX	22	N/A	PA12	CAN
SWDIO	23	SWDIO	PA13	SWDIO
SWCLK	24	SWCLK	PA14	SWCLK
USART Protocol Rx	25	USART_RX	PA15	AF
Turn Sensors on	26	SENSOR_ON	PB3	DO
Sensor 2	27	SENSOR2	PB4	DI
Sensor 1	28	SENSOR1	PB5	DI
Free	29			
Plugged into Umbilical?	30	ACOK	PB7	DI
Bootloader/CH1 PWM output for speaker	31	SPKR	PB8	PWM OUT

Light Sensor

So upon t esting, I discovered that the BPR301 gives me a voltage when I apply a reflective sheet of paper. The output voltage is less than 2.0V however, which is the threshold for the 3.3V logic level. So Andrew suggested that I use a smaller resistor value on the LED to allow more photons to hit the base of the transistor. I am testing a few different values

Resistor Value	**Diode Current**	**Output Voltage**
470	5mA	~1.8V
330	6.6mA	~2.3V
220	9.8mA	~3.0V
110	18.8	~3.1

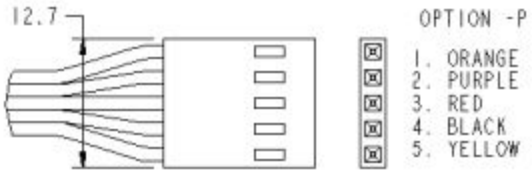


I chose to use the 220 Ohm resistor.

RTOS Based Firmware

Out of curiosity, I decided to write two versions of the firmware...one with FreeRTOS, and one on the bare metal. This will involve me reading the documentation. Right off the bat I can tell this probably won't work because my MCU only has 6kb of RAM which is not enough for FreeRTOS...oh well.

Board Debug Session 1 - 3/11/2020

- Right off the bat, we discovered that my **mosfet has the wrong orientation**. The gate pin is currently placed where a Source pin is expected. I mixed up the footprint on my schematic.
- Secondly, one of the sensors on the board (the one I had soldered to leads for testing) is broken. **A fresh sensor works**.
- The **speaker doesn't seem to work**.
- The LED works!
- The linear actuator pins are not aligned correctly
    - 1 - Orange – Feedback Potentiometer negative reference rail
    - 2 - Purple – Feedback Potentiometer wiper
    - 3 - Red – Motor V+ (12V)
    - 4 - Black – Motor V- (Ground)
    - 5 - Yellow – Feedback Potentiometer positive reference rail
    - 

- The Motor power works!
- The LA power works!
- Sensor 1 works!
- SPS is working!
- Add oblong annuli to the sensors

- Shorting the arming pins turns off the board
- **Probably want to add more test points**

Things to still test

1. The optoisolator signals from the Telemetrum.
2. The feedback signals from the linear actuator and the motor
3. The speaker still not working?
4. The full flight software

Board Debug Session 2 - 3/12/2020

- **Yeah speaker doesn't work**
- **Need to test the analog drivers...read HAL manual**
- **The analog drivers require me to switch some pins around...moving SPKR, the LA Position, and BATT_READ...Need to change those in the schematic**

Things to still test

5. The optoisolator signals from the Telemetrum.
6. The feedback signals from the linear actuator and the motor
7. The full flight software

Board Debug Session 3 - 3/13/2020

- Motor analog speed controller seems to be really bad?
    - It gives me numbers that alternate between 130-150 and 220-250 when the motor is running, but responds with lower numbers when I unplug the collector, it gives me values of 40-60. When I touch the collector to ground, it gives 0 and when I put it on rail it gives 4095 (don't do this again too much voltage to the MCU), which means the ADC is working.
    - Not sure if I want to rely on this motor's speed reading.
- LA analog position works very well! Max position is a consistent value of 2580 ± 5, and minimum position is 0. Very well made!

- Optoisolator signals work very well! They do their job of transforming a higher voltage signal to a lower voltage signal with two different grounds. The input side can be 5V, 7V, whatever, and the output side will always be 3.3V. Very nice
- I soldered and tested sensors on the board, and they work great!
    - **Sensor 1 dims the LED more than Sensor 2**...this is weird.
- Might want to add test points all over the board to make the next Rev debugging much easier.

Things to still test

- Flight software inside the nose coupling. **Write the test software and create cables to test board in the cone**

Board Debug Session 4 - 3/15/2020

- Boot0 neeeds to be pulled down for the board to boot from flash
- Speaker needs to be driven with a 4000Hz PWM signal and so I need to read the PWM manual. In the meantime though, we are using the PB8 boot0 pin as a PWM output to drive the speaker. We are board hacking by jumping the spkr to PB8, and so I need to change that on the second rev.
- To board hack...exacto knife a trace to seperate it and mod wire it to the correct place, soldering the proper locations. Andrew did this for the spkr pin, and the LA analog pin to correct those. POT_OUT_AN went to PA1. this leaves PA10 floating, and PB6 floating.
- To read multiple analog channels need to use a multiplexer. The chip has a multiplexer built in so I need to figure out how to read channels.
    - We also sliced off PB6 which was BATT_READ.
- I was completely wrong with the motor...it is not a potentiometer, it's a Frequency generator. This is a transistor that is pulling an input pin to ground, making a pulse. The pulses were created by something spinning in the motor. What we're doing is hacking again to put in a 10K pullup to 3.3V to be driven to ground by this transistor.
    - We discovered there is another resistor inside the motor that makes the frequency not pull all the way to ground. This is unwanted. It makes the low level be like 1.5V which is too close to the min of a HIGH reading. If we reduce the resistance of our pull up it will bring the lower level down to 0.9V which is fine for the STM32 reading a low.
    - Andrew found a 4.7K 0402 package resistor and modwired it to the board in place.
    - We did the math wrong and the voltage low went in the wrong direction.
    - Time the pulses with interrupt on change
    - Set interrupts at a at number of pulses

Things to still test

- Test the PWM pins