“Development of an Adaptable Vent for Latex Weather Ballooning”
**Vent Flapper**
- Releases lifting gas

**Attachment Point**
- Secured to the balloon neck

**Resistor Cutters**
- Burn to release the balloon

**Servo Notch**
- Avionics attachment
Mechanism
Thin tubing with rings cut out of caps for reinforcement
Inward-opening flap pressed against the bottom ring
EPDM foam seal attached to the opening flap
Access port and filling hatch
Foam rings sealing the balloon neck and rigging
“Service-able” design
VENTING ACTION

VENT DOOR OPENS INWARD

SERVO AND ROD ATTACHMENT
ADJUSTABLE LENGTH
FOAM SEAL

THE ATTACHED SERVO CONTROLS THE “FLAPPER”, THEREBY RELEASING LIFTING GAS UPON COMMAND
VENT RIGGING

**Top Rigging** starts at the “Lasso” and runs through a polyethylene tube, with a notch for resistor burner. The line ends in a loop at the bottom, where payload stacks can be clipped to it. The rigging secures the balloon neck to the vent in a circular notch.
The primary resistor performs a controlled, pinpoint burn on the “waist line” rigging securing the balloon neck.

The rigging is cut loose, the balloon neck itself slips off of the vent, and the payload begins its free-fall.

Should this burn fail, the emergency cut resistor activates, and the vent assembly is lost.
Avionics
**FUNCTIONALITY**

- **ACCESSIBLE CASING**
- **HARDWARE PACKAGE**
- **AUTONOMOUS CONTROL**
- **LED INDICATORS**

```c
// Function to release the vent from the battery

void terminate()
{
    if(emulationCheck == true) { // used only for emulation
        Serial5.print("TERM");
    }
    currTime5 = millis() / 1000; // Resetting currTime5 just in case it's off by a couple seconds.
}

if (terminationBegan) {
    terminationStart5 = currTime5;
    terminationBegan = true;
    Serial.print("Termination has begun.
    
    if (terminationStart5 + 20 < currTime5) {
        if (!Res1Burned)
            if (!Res1on)
                burnTime1Start5 = currTime5;
            else if (!Res1Burned2)
                if (!Res1on)
                    burnTime1Start5 = currTime5;

        burnResistor();
        Serial.print("Burning waistline resistor.");
        if (currTime5 - burnTime1Start5 > 7.9) {
            stopBurn();
            Serial.print("Waistline resistor burning stopped.");
            Res1Burned = true;
        }
        else if (!Res1Burned2)
            if (!Res1on)
                if (!Res1on)
                    burnTime1Start5 = currTime5;
    }
```
MODULAR DESIGN
ONBOARD SENSOR SUITE

80 Female Headers | 8 Terminals | 2 H-Drivers | 1 H-Driver Socket

2 Resistor Cutters | 1 Thermistor | 1 Heater Relay | 1 Pressure Sensor

2 9V Battery Connectors | 1 5V Regulator | 1 Ublox GPS | 1 Teensy 3.5
-40°C
MINIMUM OPERATING TEMPERATURE

85°C
MAXIMUM OPERATING TEMPERATURE

3.5V
EXPECTED VOLTAGE

PRIMARILY USED FOR VENTING DECISIONS

GPS/ALTIMETER

Enables location and altitude specific data
HiLetgo MS5611

**MINIMUM OPERATING TEMPERATURE**
-40°C

**MAXIMUM OPERATING TEMPERATURE**
85°C

**EXPECTED VOLTAGE**
3.5V

**PRESSURE SENSOR**

Backup for GPS Venting Decisions
Calculates altitude from pressure data