A Multi-Sensor Array for Atmospheric Science

**Part 1: Introduction to Main Subsystems**

**Introduction**

This two part series describes the design and construction of a Multi-Sensor Array (MSA) for studying atmospheric pollution in an urbanized mountain basin; specifically, the region above Salt Lake City, Utah (USA). The MSA is flown on research balloons by HARBOR, an undergraduate research group located at Weber State University in Ogden, Utah. The MSA produces a column of measurements from ground level (approx. 1km ASL) to the lower stratosphere (approx. 35 km ASL).

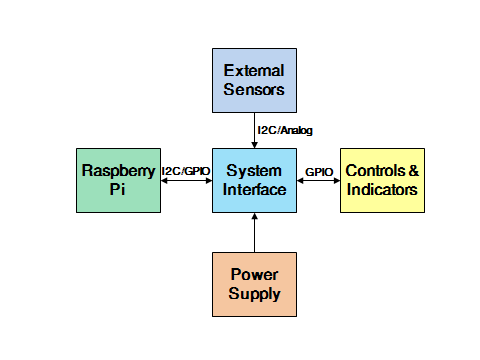
During flight, the system is exposed to pressures as low as 0.75 mmHg, where heat exchange becomes difficult even at temperatures of -50°C. Jet stream winds can reach speeds of 160 km/h, applying punishing shock and vibration forces to our electronic equipment. In this extreme environment, the MSA must continue to gather scientific data for a minimum of 4 hours.

The first part of this series focuses on the hardware design of the MSA system. Part 2 will describe how the software was designed and implemented.

**Theory of Operation**

The MSA is a data logging computer with a standard set of built-in sensors and an expanded set of inputs to allow guest packages to be connected. External devices such as LEDs, buzzers, pumps, and heaters are controlled via existing GPIO pins. The MSA is comprised of six major subsystems:

1. Raspberry Pi (Version 1 - Model B)
2. System Interface Daughterboard
3. External Sensor Board
4. External Controls and Indicators
5. Power Supply
6. Enclosures



The Raspberry Pi serves as the primary platform for the MSA system. It monitors inputs, communicates with sensors, processes and stores flight data, and controls LEDs, motors, and heaters.

**System Interface Daughterboard**

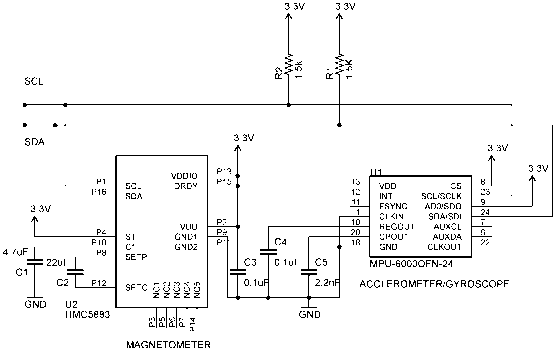
The System Interface Daughterboard (SID) links the Raspberry Pi to everything else in the system. It also provides regulated power. The SID is where analog and digital sensors are mounted or connected. It is equipped with the following components:

* MPU6000 (3-axis accel/gyro)
* HMC5883L (3-axis magnetometer)
* MPX2102A (pressure sensor)
* MAX4208 (instrument amplifier)
* HIH5030 (humidity sensor)
* TMP112 (temperature sensor)
* COM-08720 (pushbutton)
* MIC2937A (3.3V regulator)
* MIC2937A (5.0V regulator)
* LTC2495 (16 Channel ADC)
* DS3231M (Real-Time Clock)

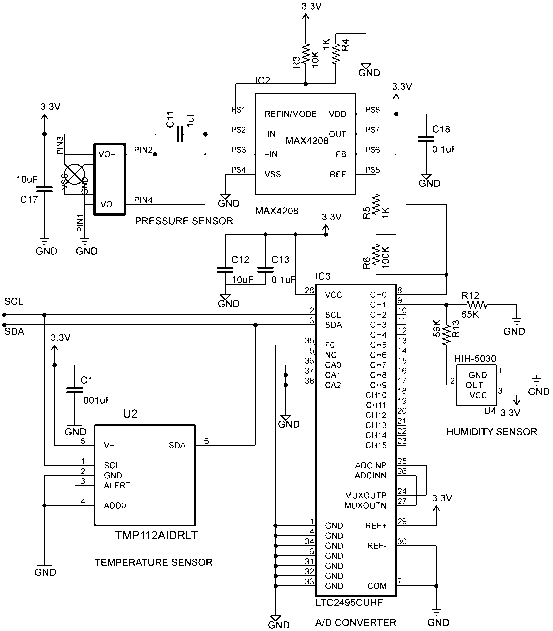
The sensors are used to calibrate data and help eliminate bad data which may have been recorded during an event; such as an impact, extreme temperature, or high humidity.

All sensors communicate over the available I2C bus, with analog sensors routed to an I2C capable ADC.

There are two primary sensor circuits built into the SID. The first is a flight dynamic circuit which consists of an I2C accelerometer/gyroscope sensor and magnetometer.

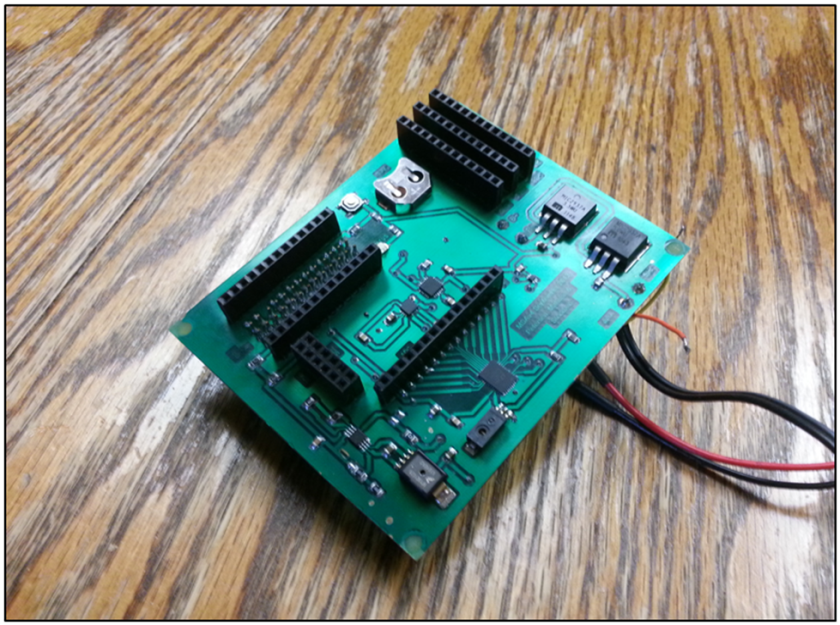


The second is an environmental circuit composed of temperature, humidity, and pressure sensors.



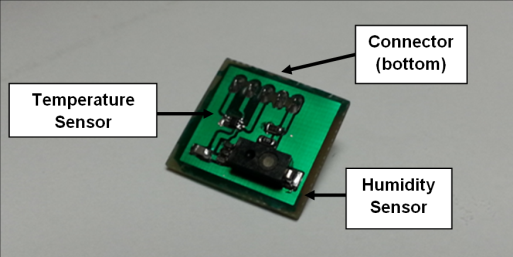
The ADC can support up to sixteen single-ended analog sensors, not including an internal temperature sensor. The I2C bus can support many more devices.

We designed our circuit boards using a free version of the Eagle PCB Design Software by CadSoft. We manufactured the boards in-house using commercially available chemicals and photosensitive PCBs. The top view of a completed SID board is shown below. The assembled board mounts to the Raspberry Pi via the 2x13 pin header.



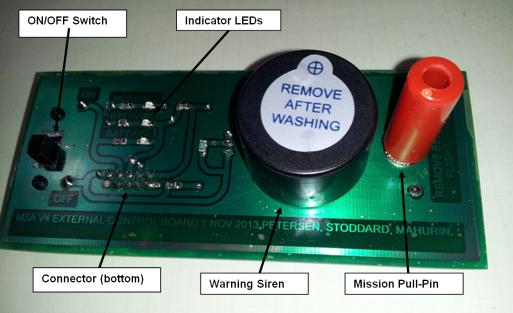
**External Sensors**

The external sensor board consists of a TMP112 temperature sensor and an HIH5030 humidity sensor mounted to a small PCB embedded in the exterior wall of the external enclosure and connected with a short harness.



**External Controls & Indicators**

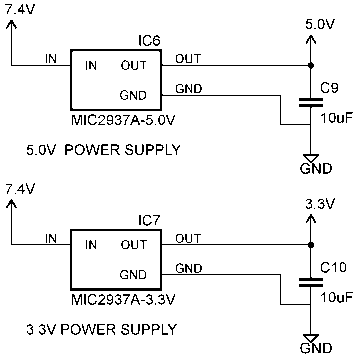
External controls and indicators allow the user to power up and shutdown the MSA from outside its insulated enclosure. LEDs and buzzers allow the user to know the status of the MSA, whether it is in standby or data logging mode, and also alerts unaware people on the ground when it is descending near the surface. A pull-pin starts a mission timer to allow synchronization of data between various instruments at the start of a flight. It can also be re-inserted to save data files and initiate a safe-shutdown.



**Power Supply**

The power supply subsystem is a combination of software and hardware to provide power and initiate a safe shutdown in the event of battery failure.

The hardware component is basically 7.4V LiPO battery and two linear voltage regulators (5V and 3.3V). The 5V regulator supplies power to the Raspberry Pi and has a couple hundred milliamps available for other devices. The 3.3V regulator provides voltage for the I2C bus and all of the onboard sensors.



Battery power is a precious commodity on balloon flights, as capacity per gram quickly eats into mass budgets. Federal regulations limit balloon packages to 5.4 kg, which must be distributed amongst various flight packages; including radio transmitters, cameras, and other scientific instruments. The MSA system was limited to just 408g which made battery selection crucial.

The Raspberry Pi draws between 310mA and 390mA under normal operation; this includes the current draw of the HDMI output chip and the USB/Ethernet chips, which account for nearly one half of the total current consumption. To prolong battery life during flight, these chips must be disabled. The chips are disabled in the MSA code by calling two Linux commands.

The C code to shut down the HDMI video chip is:

system("/opt/vc/bin/tvservice -off");

The next line of code suspends the USB/Ethernet controller:

system("sh -c \"echo 1 > /sys/devices/platform/bcm2708/usb/bussuspend\" ");

Both of these chips are re-enabled when the MSA is rebooted or powered up for the first time.

After the chips are disabled, the average current draw of the Raspberry Pi drops to 180mA.

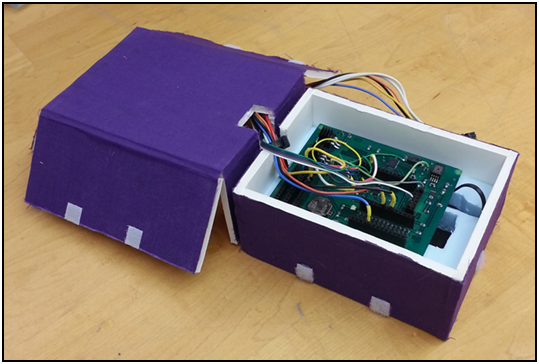
Factoring in the additional power consumption of the sensors and ICs, a total of 14.42 mA, and the required battery life of 4 hours, a minimum battery capacity of 778mAh is required.

4hrs x (180mA + 14.42 mA) = 778mAh

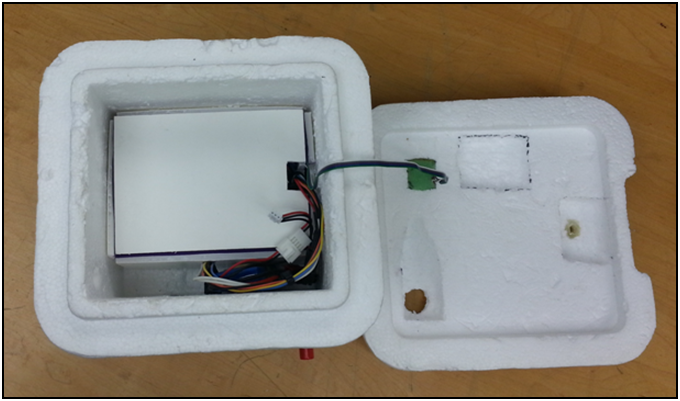
We selected a 7.4V 1100mAh LiPO battery, which provides a modest amount of headroom and only costs us 72g of our 408g mass budget.

**Enclosures**

An internal enclosure was designed to protect the Raspberry Pi and SID from shock and vibrations as the MSA makes its way through the jet stream and upon landing. It is made of high-density foam wrapped in a lightweight fabric. The fabric acts as a hinge and allows the lid to wrap around the sides, which are secured with Velcro strips.



The internal enclosure fits snuggly into a 30cmx30cmx30cm Styrofoam external enclosure. The external enclosure provides structural rigidness for the external controls, and insulates sensitive components from extreme environmental conditions.



Completely assembled, the whole unit weighs 406.9 grams. It has been successfully tested on four flights and has reached an altitude of 34 km. It has survived landing in rugged desert terrain and also a reservoir.



**Coming Up**

In part 2 of this series, I will discuss how the MSA initiates the data logging program at startup, communicates with some of the sensors, stores data, manages sampling rates, interfaces with the user, and responds to conditional events; such as low battery voltage. The code for the MSA is written completely in C and is facilitated with the aid of the BCM2835 C Library, maintained by Mike McCauley at www.airspayce.com. The code also takes advantage of semaphores and parallel programming to create a timer without the use of interrupts.

