

ADVANCED FLIGHT ELECTRONICS FOR HIGH POWER ROCKETRY

Part I: Flight Electronics Basics

In the four decades since the inception of the Tripoli Rocketry Association (TRA) amateur high power rocketry has grown in both performance and complexity as has the electronics to support it. Any Level 3 certified flyer is well familiar with flight electronics, as are most Level 2 flyers. High end Research (or Experimental) rocketeers are pushing the limits of performance with motor thrusts of hundreds or even thousands of pounds and altitudes well over 100,000'. In 2009 the Federal Aviation Administration (FAA) increased the maximum ceiling these flyers can pursue to 150 kilometers (93 miles) and total system energy to 200,000 pound-seconds.

For most amateur flyers the demands on the electronics are fairly simple and basic:

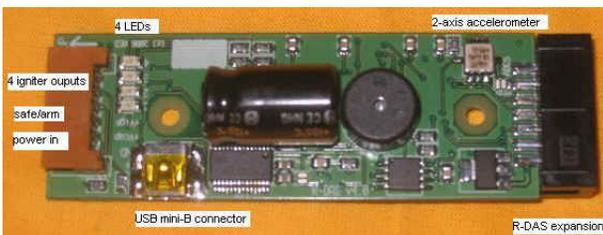
- Record and report maximum altitude attained
- Provide for parachute recovery through a pyrotechnic device triggered at apogee and/or a specific altitude during descent

These systems are generally based on 8 or 16 bit single chip micro-computers operating at low speed for power savings and using generally available and low cost sensors.

The advanced flyer will want more control and data on their flight including:

- Time/Altitude plot
- Time/Acceleration plot
- Dual deployment recovery (Drogue/Main Chute)
- Additional pyro channels for motor air-starts
- One-way telemetry
- On-board GPS

The later not only provides a three dimensional plot of the rocket's course but the last GPS coordinate received by telemetry provides an accurate location for recovery.



A popular computer at this level is the AED RDAS Tiny selling for \$300 for the basic unit. It has four pyro outputs, 2.9-15.2 psi barometric altimeter, $\pm 50g$ accelerometer (0.1g resolution) and samples data at up to 200 times per second for 30 minutes.

But the demands of advanced high performance rocketry are exceeding the capabilities of most flight electronics systems available today. In part it is a matter of cost as the electronics can become quite expensive. However when large projects take months or years to design and build and cost thousands of dollars an investment to improve the reliability, safety and data recovered from flights makes it worthwhile to spend more in this area.

Another limiting factor is the flexibility of most currently available hardware and its supporting software. Most flight computers are single board designs with limited options and configurations. A user may have to buy more than they want to get what they need, or may simply not be able to buy the features they need off-the-shelf. Rocketry has become so complex it is not practical for a single individual to perform all forms of design needed themselves, especially complex electronics hardware and software. This has unfortunately placed serious restrictions and limitations on what kind of research and development the private researcher can attempt.

There are many serious flyers that develop their own high performance motors, operate static test stands and need high performance (altitude/speed) instrumentation and control systems. Achieving FAA Level 3 flight qualification places much higher demands on system functionality and performance, especially in multi-stage configurations.

Other researchers are interested in measuring and characterizing flight dynamics, loads and performance. Still others are involved in developing active flight stabilization controls. There are even a few courageous and ingenious people developing complex liquid fuel rocket motors.

Based on its specific and sometimes unique needs Black Magic Missile Works (BMMW) has developed three generations of Universal Flight Controllers (UFC). The need arose from the development of computer controlled hybrid rocket motors designed to increase performance, reliability and safety. By managing the flow of the self pressurized Nitrous Oxide injected into the combustion chamber, the computer can optimize motor performance for the flight envelop and atmospheric conditions. It can also optimize fuel efficiency by compensating for the wide pressure variations (and therefore flow rate) in NOS due to temperature, maintaining optimal oxidizer to fuel ratio. Motor faults, such as over-pressurization or stability failure can be protected against or minimized by shutting off oxidizer flow, something not achievable in common solid rocket motors.

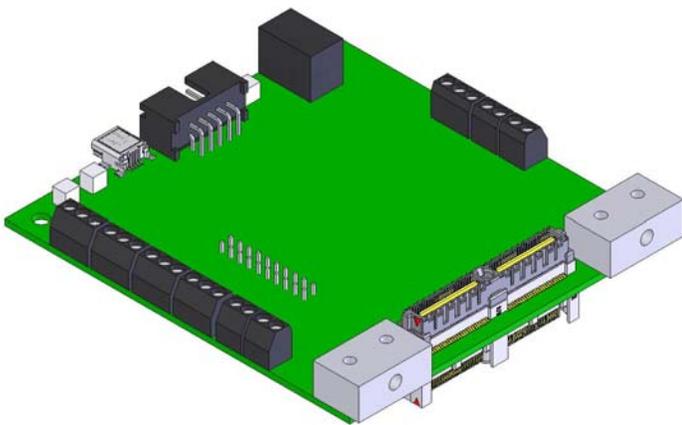
The first two generations of UFCs were single board computers with a few accessories; Two-way long range telemetry, 5 fix per second GPS and later a 3-Axis Roll/Pitch/Yaw Inertial Sensor (all external modules attached by cables).

The second generation UFC-2 was made available to the amateur market two years ago. Because it was more advanced than anything else available, it was also more expensive and more difficult to use. It has additional features that not all users can make effective use of including numerous analog input channels and digital I/O ports, many with power output drivers. It has improved but still limited configurability to meet unique needs of individual users because it was designed to meet BMMW's specific needs.

The third generation UFC-3 was designed from the bottom up to address these issues and provide unequaled flexibility and configurability. The system software also saw major redesign to simplify the user interface and increase flexibility.

Modular Design

By basing the UFC-3 on a modular design it is possible to create a single-board system that provides all the basic features required for common flight systems in a single Base Module at minimal cost, yet allow the original purchased configuration to grow in size and complexity as the user's needs (and budget) grow. Up to seven identical size modules can be stacked on the Base (CPU) Module to extend performance and features into a highly complex system. Furthermore, module stacks (nodes) can be connected over a RS-485 network to provide cooperative redundancy or distributed processing of up to 9 nodes (that's 72 active modules in a single system!).



The Base Module provides:

- System processor: low power, 32-bit, 200 MHz ARM9 SOC (System On a Chip)
- Ground Support serial port (laptop connection)

- 256 Mbytes of Flight Log Memory (Error Correcting NAND Flash, optional 1 Gbyte SD Card may be used)
- 512 Kbytes SRAM
- High performance 0.15-16.2 psi 15-bit Barometric Altimeter
- Dual ($\pm 8/\pm 50g$) Accelerometers
- Two constant current pyrotechnic channels for recovery
- Eight user definable digital I/O ports
- 104 pin stacking connectors for module expansion (top and bottom)
- Power conditioning and monitoring for the entire module stack

All of this at a 25% lower cost than the base UFC-2 configuration. Using modules, additional functions can be easily added, often offering different levels of performance or meeting different legal requirements to further reduce costs.

For instance, most flyers do not fly over 60,000' and many don't exceed mach 1.5. Common GPS receivers are limited to these maximums by international COCOM agreements, however removal of these limits is available using an alternative GPS module (requires application for use and additional costs and fees). UFC-3 offers two different modules to meet these needs.

Likewise North American and European requirements for unlicensed wireless communications differ. 900 MHz is the standard North American frequency for long range wireless, whereas Europe standardizes on 868 MHz. Once again UFC-3 offers two different modules to meet these needs, but from the perspective of the user they are functionally identical.

Additional modules to support high precision analog sensors (up to 32 channels of 16-bits each with thermocouple support), Roll/Pitch/Yaw Inertial Sensor and Six Channel Pyrotechnic Modules (up to three modules per node) are options. Additional modules provide for cooperative redundant control of critical flight functions (as opposed to simple passive systems where two entirely independent computers attempt to control the same resources separately, such as recovery systems).

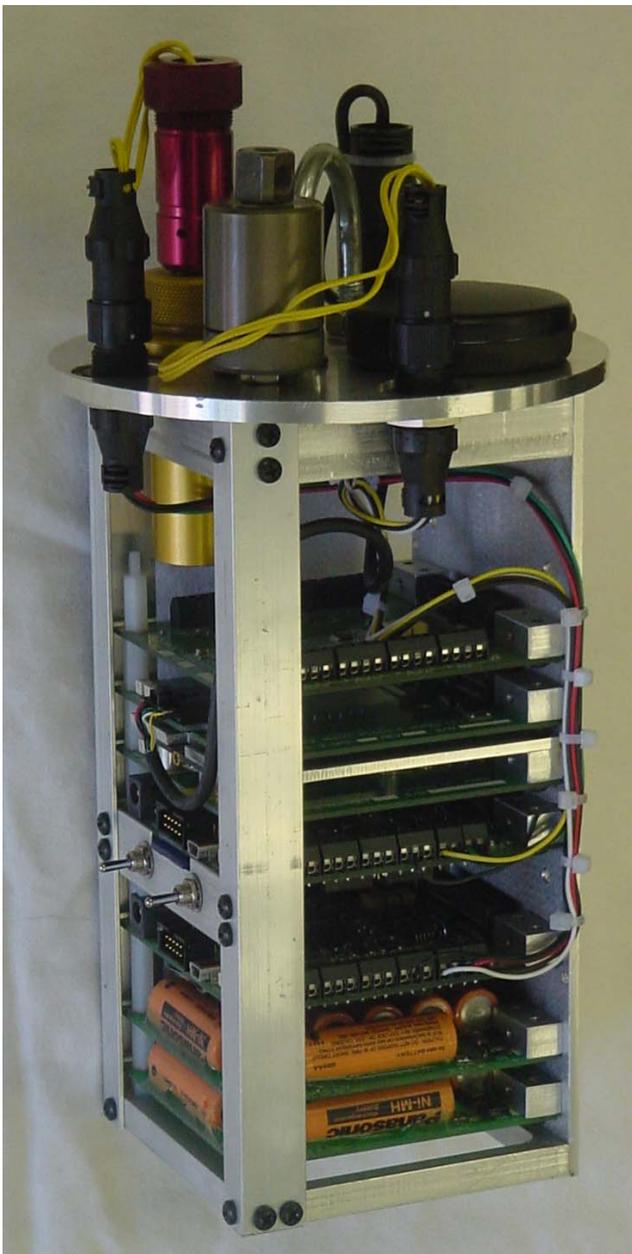
Finally there are two distinctly different power modules available (not part of the eight module per node maximum); one using industrial 1.2V Nickel Metal Hydride batteries (one bank of 9.6V @ 2,000mAh) with a temperature range of -10°C to $+65^{\circ}\text{C}$; the other being a more expensive one using 3.6V Organic Lithium aerospace cells (configured as two banks of 14.4V @ 1,600mAh or four banks of 7.2V @ 800mAh, redundant operation available) with a temperature range of -40°C to $+85^{\circ}\text{C}$.

Summary

Although a more expensive solution for the entry level user, the modular design allows minimal investment for relatively simple needs yet the ability to scale up the system as needs grow without having to replace substantial pieces of hardware or learn new systems. Ultimate performance and functionality far exceeds anything else available. Additional modules are under development and potential users are encouraged to submit ideas for new modules. Custom modules can be built for a reasonable cost.

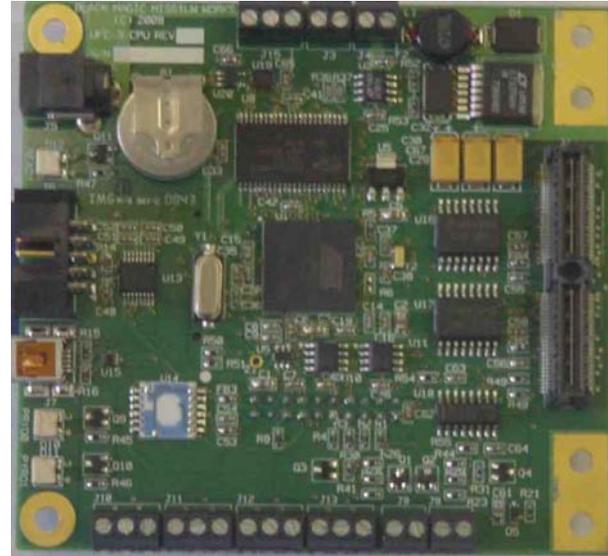
Part II: Hardware Design and Component Selection

The basic Module is a 3.55" by 3.77" Printed Circuit Board (PCB) with a vertical spacing of 0.625" between modules (the PCB adds an additional 0.062"). This requires a minimum 6" inner diameter airframe. Ruggedized 104 pin stacking connectors with an integral ground plane provide signal transfer between modules. All modules use immersion gold plating and are conformal coated as required. Each module has mounting for two optional thermal blocks along the back edge corners for heat transfer to a common heat plate. Almost all components are rated -40°C to +85°C (except for the low cost industrial Battery Module option).



Each module, including the Base Module, has a Serial Electrically Erasable Programmable Read Only Memory (SEEPROM) that identifies its function and carries its user defined configuration. Each module picks the signals it needs from the stacking connectors and passes the remaining signals on to the modules above it, making each module position independent.

Base (CPU) Module



The \$600 Base Module provides all the basic functionality for flight logging and dual deployment recovery. Although it is twice the cost of the RDAS, it has substantially higher performance (more than 5x), memory capacity (256x) and much greater expandability and configurability.

The Atmel AT91SAM9XE processor provides 200 MHz performance with minimal power consumption. It has several on-chip memories:

- 512Kbytes Program Flash
- 32Kbytes High Speed SRAM
- 16Kbytes Instruction Cache (Icache)
- 8Kbytes Data Cache (Dcache)

This is a "System On a Chip" or SOC, meaning it has a number of peripheral devices and several off-chip busses:

- External Memory Bus (SRAM and NAND Flash)
- Error Correction Code (ECC) protection for the NAND Flash
- I2C Serial Bus (Module SEEPROMs and I/O)
- SD Memory Card Interface

- Five Serial Ports
- Two Serial Peripheral Interface (SPI) Busses (4 devices each)
- Four 10-bit ADC Channels (for the accelerometers, battery voltage/current monitoring)
- Two Host USB Ports
- Device USB Port
- 45 Digital I/O ports (26 bussed in the stack)
- Coin cell battery backed Time Of Day clock

The on-board 256Mbyte NAND flash provides high speed, shock and vibration proof Flight Log storage. It can absorb all sensor data and other flight information at a rate of up to 1,024 samples per second. Alternatively an SD Memory Card of 1Gbyte capacity can be used, but it is slower (limits maximum sample rate) and is subject to shock and vibration issues.

There are also 512Kbytes of on-board SRAM to buffer the SD Card data and provide a full buffer for firmware update downloads. It also has the potential future use of storing the program code (loaded from the NAND Flash or SD Card) when a 400 MHz processor (AT91SAM9G20 which does not have on-chip Flash) is installed (this version is not available as yet because the 200 MHz processor has proven itself capable of sufficient performance).

Altitude	Resolution
0	2.6
1,000	2.7
5,000	3.0
10,000	3.5
15,000	4.1
20,000	4.9
25,000	5.8
30,000	7.0
40,000	10.2
50,000	15.6
60,000	25.0
70,000	42.5
80,000	80.0
90,000	150.0
100,000	368.0

The Intersema MS5534CM Barometric Sensor has a wide range (0.15 psi to 16.2 psi) usable to 100,000'. The internal 15-bit ADC is temperature compensated and factory calibrated, with the calibration data stored in internal memory and read out during system initialization, providing the best possible accuracy (this is the most expensive single component on the Base Module). The temperature and pressure are measured 8 times per second.

The dual accelerometers ($\pm 8/\pm 50g$) provide wide range and improved accuracy and resolution (0.019g) at relatively low acceleration. Each device has a status pin that is tested to verify proper operation and a self test pin that is used to internally activate the sensor to a known simulated gravity (sensor deflection), further assuring proper operation. These devices are connected to two of the internal 10-bit ADC channels of the SOC.

The pyrotechnic outputs are current limited to 1 Amp regardless of input voltage to protect the battery and system from excessive currents while firing igniters. The on-time is

limited to one half a second to assure that the drivers cannot overheat into a dead short. The pulse circuit providing this timing is disabled during power up to assure the channels do not accidentally fire. A continuity circuit provides feedback to the SOC so the connection status of the igniter can be verified and yellow LEDs provide a visual indication.

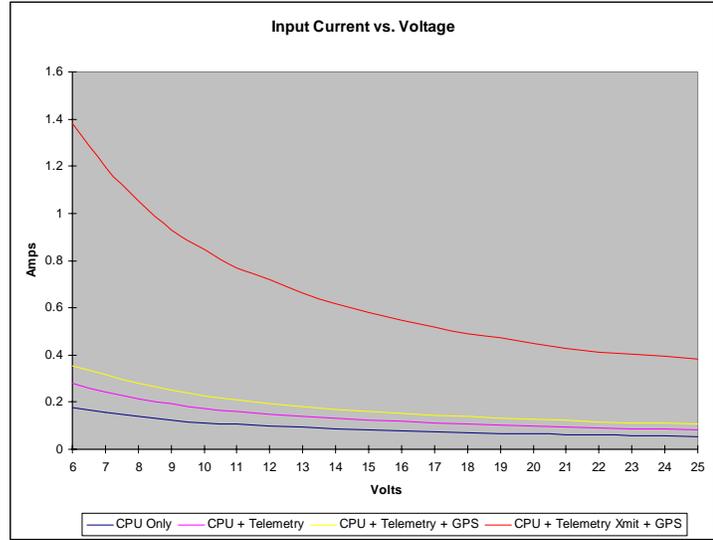
The on-board serial port is buffered by an RS-232 transceiver to connect into a standard PC serial port. Because many laptops no longer have serial ports, a USB port is also available, or the user can connect a Serial to USB adapter cable.

Eight of the Digital I/O ports are available for user connection to external logic. They are 3.3V compliant and have protection diodes to 3.3V and ground. The eighth port also has a Green LED driven through a buffer. The user can use the port for their own purpose, or configure the system to use the port/LED as a status indicator; solid on for power applied, slow blinking for normal non-flight operation and fast blinking when the flight sequencer is operational.

There are two additional digital signals that go out the stacking connectors; DEFAULT and MASTER. These along with the fourth serial port provide a mechanism for cooperative redundancy and fail-over. The Isolation Module connects two stacks together allowing the two Base Modules to communicate by passing a Heart Beat between them over the serial port. The static DEFAULT signal is high for the upper stack and low for the lower stack. The MASTER signal is normally high. Circuitry on the Isolation Module and some accessory modules monitors these two signals. When they are of equal value the controlled function is enabled. By default this is the upper stack. If the lower Base Module detects a failure of the upper Base Module it pulls the MASTER signal low, giving it exclusive control over these functions. This allows only one computer at a time to control a function and passing control over on failure.

The entire stack may be powered from a single power source (which might be redundant batteries) with a wide input range of 6 to 25 Volts. A connector on the front of the module provides a standard 2mm battery charger connection. Current from the battery is monitored by a circuit that allows an SOC ADC channel to monitor both current into the system and going back out to the battery when charging. An additional connector allows additional loads to be connected for monitoring. Maximum currents are limited to 4 Amps and reverse polarity protection is provided by diodes. Another SOC ADC channel monitors the battery voltage. An on-board 5V regulator provides regulated power to the rest of the stack (the battery power is also available to all the modules through the stack). There are also on-board 3.3V and 1.8V regulators for the CPU circuitry. A 3V Lithium coin cell battery and 1.8V regulator

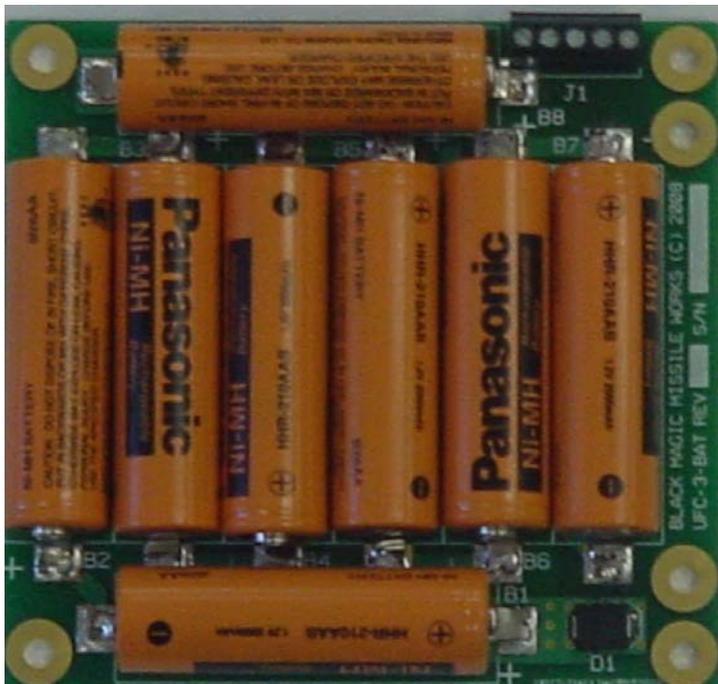
provide a separate power source to keep a Time Of Day (TOD) clock running when main power is removed.



Battery Modules

One or more Battery Modules can be added to the stack. These modules do not count as part of the eight modules because they do not tie into the stacking connectors. They merely have the same form factor and mounting, with somewhat greater module spacing due to the AA battery diameter.

Two types of modules are available; a single bank 9.6V Nickel Metal Hydride Industrial battery (shown) providing 2,000 mAh of current and a two or four bank, 14.4V or 7.2V Organic Lithium Aerospace batteries providing 800 mAh of current each..

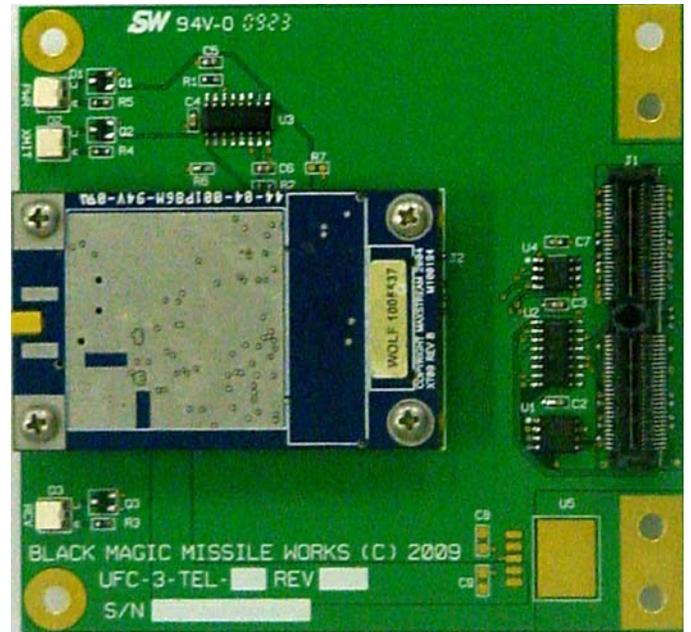


The \$125 Industrial Battery Module has a temperature range of -10°C to +65°C. Continuous current rating is 4 Amps.

The Aerospace Battery Module is available with up to four banks of batteries for \$500, but because there can be four banks the cost per bank is about the same. However the temperature range is -40°C to +85°C, the continuous current rating is 4 Amps (but a lower total capacity) and the cells have a peak rating of 15 Amps for one second, supporting the non-Super Capacitor version of the Six Channel Pyro Module. Overall the cost can be less in these configurations and the level of redundancy and greater temperature range make it an attractive option.

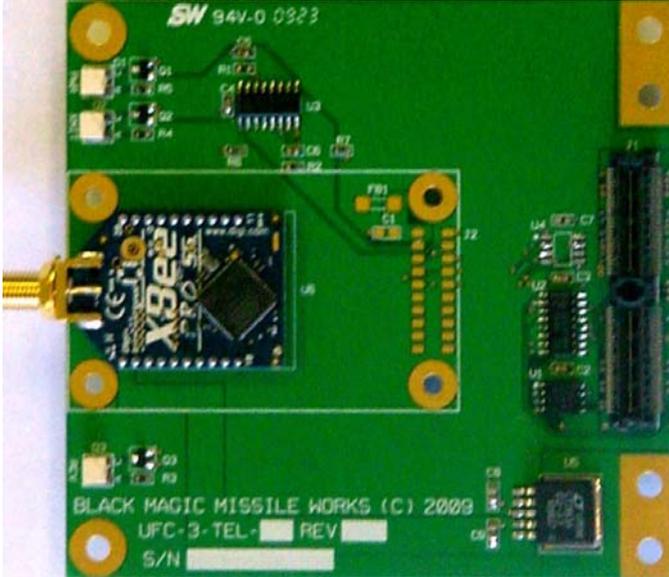
Telemetry Modules

Two Telemetry (RF Modem) modules are available providing complete two-way long range communications. The modems provide transparent serial connections between the Base Module and ground station, however special character sequences are used to communicate directly with the modem for setup and obtaining status. Both modems have programmable power and channel assignments. Both modules support Power LEDs (and power control by the Base Module), and Transmit/Receive LEDs. Functionally they operate in a nearly identical manner.



The \$825 (complete) 900 MHz North American module and base station provides up to 1 Watt of transmitter power, with a line-of-sight range of 40 miles or more depending on the antenna gain. Telemetry data rate can be set to 9,600 baud (about 1,000 characters per second) for the greatest range or 115,200 baud (over 10,000 characters per second) for about half the range.

When operating at full power and at high data rates (duty cycle) the internal temperature of the modem can increase by 40°C or more. When added to the already high temperature of a rocket waiting at the launch pad in the desert, the 85°C rating of the module can be exceeded. An optional heat plate that connects to the module heat transfer system reduces the temperature rise to 5°C.



The functionally identical \$600 (complete) 868 MHz European module and base station provides up to 315 mw of transmitter power, with a line-of-sight range of up to 25 miles. The telemetry data rate is 24,000 baud, however for unlicensed operation the modem is limited to a 10% duty cycle over 1 hour. This provides an effective maximum of about 800,000 characters TOTAL.

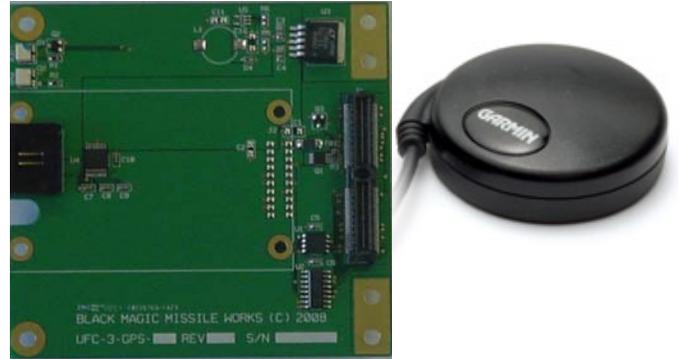
The software that samples data and sends it over the telemetry link allows the user to define what data and at what sampling rate it is to be sent, allowing the user to assure that essential data (system state messages, GPS location, Altimeter, etc) are always transmitted. Data for on-board logging is independently defined and can be at the maximum sensor data rate.

Ground station modems are supplied and various optional high gain antenna systems (omni-directional, dipole and parabolic) are available. In addition, power output can be increased with licensed external amplifiers (5, 10 and 25 Watt models, www.l-com.com).

GPS Modules

There are two GPS Modules available; a 5 fix per second unit (using an external Garmin GPS 18x Receiver) and a 20 fix per second unit (using a NovAtel OEMV-1 internal Receiver and external Antenna). Both modules have Base Module power control, Power and Lock LEDs.

These units provide not only Longitude and Latitude, but Altitude, Course and Horizontal Velocity information. The Horizontal velocity can be used as an "Event" by the software to qualify user specified actions. This could be used to prevent a motor air-start or staging if the horizontal velocity were too great.



The \$450 External GPS Module has an on-board RS-232 transceiver to communicate with the GPS receiver. Resolution and accuracy are good, but the unit is limited to 60,000' and approximately 1,000 mph by international COCOM limits. This is typical of all amateur rocket GPS systems.



The \$2,375 20 fps Internal GPS Module has an on-board 3.3V power supply for the receiver and 8V power supply for the external antenna. It uses the stack TTL serial port directly without buffering. Resolution and accuracy are excellent and there is even a GLONASS version, but the cost is high. By submitting an application justifying the use to NovAtel and paying an additional fee, the COCOM limits can be removed from a specific unit. It is up to the purchaser to make these arrangements directly with NovAtel (www.novatel.com) although BMMW can provide assistance.

3-Axis Inertial Sensor

There are two 3-Axis Inertial Sensor Modules, although one that uses an external MicroStrain 3DM sensor board is not recommended for new users (it is supported for UFC-2 users who wish to upgrade to UFC-3) and will not be discussed here.

The \$1,300 Internal Inertial Sensor Module connects to SPI port 1 and uses one digital I/O port for Base Module Power Control, with a Power LED. Due to the height of the internal sensor, this module must be at the top of the stack (the Isolation Module has a cutout to allow it to protrude through when it is installed in the lower stack of a redundant pair).



This sensor provides Roll, Pitch and Yaw rate (up to 300 degrees per second) and linear acceleration. It has on board monitors for power supply and gyro temperatures.

The system software will calibrate the unit to remove gyro drift (error) by recording the data at maximum data rate for a user specified period (30 to 150 seconds recommended) and then calculating a correction offset for each axis. This yields excellent system stability. It can be used for monitoring attitude. Yaw and Pitch position, as well as Roll rate can be "Event" inputs to the user defined scripts that control flight operations. For instance the user may qualify an air-start or staging based upon being within a certain angle of vertical.

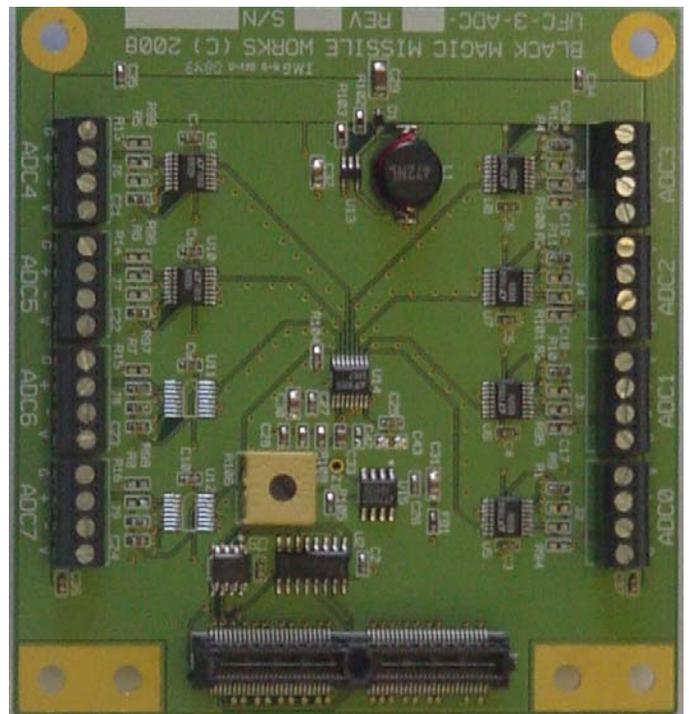
Analog-Digital Converter Module

The \$425 ADC Module (cost approximate according to configuration) provides 8 channels of 16-bit analog conversion for reading various user supplied sensors. Up to four of these modules are supported in a stack for a total of 32 channels of user data acquisition. Maximum sampling rate is 1,024 samples per second, all channels running. The 8-channel ADC is supplied a precision 0.05% reference which can be trimmed to a tighter tolerance. Each module occupies one decode space on SPI bus 0.

Each channel can be configured in a variety of ways:

- Direct input with a default range of 0-5V (higher input ranges available on special request) suitable for reading various voltages and voltage type sensors
- Differential Programmable Gain Amplifier (PGA) input with a full scale input range from 1 millivolt to 4.096 Volts
- Thermocouple input (type K) with a temperature range of -180°C to +1,200°C

In addition, each module has its own 8V precision regulated power supply for powering external sensors or providing the excitation voltage for millivolt bridge type sensors (pressure sensors and load cells). Additionally, the user may specify that the last channel on each module be used to monitor this voltage so that the exact excitation voltage can be automatically factored in the output calculation.



One very important thing to note that will be described in greater detail later is that each input is fully user definable. That is the user is free to connect the appropriate device or sensor to each input and define the scale and offset for each input, along with its name and unit of measure.

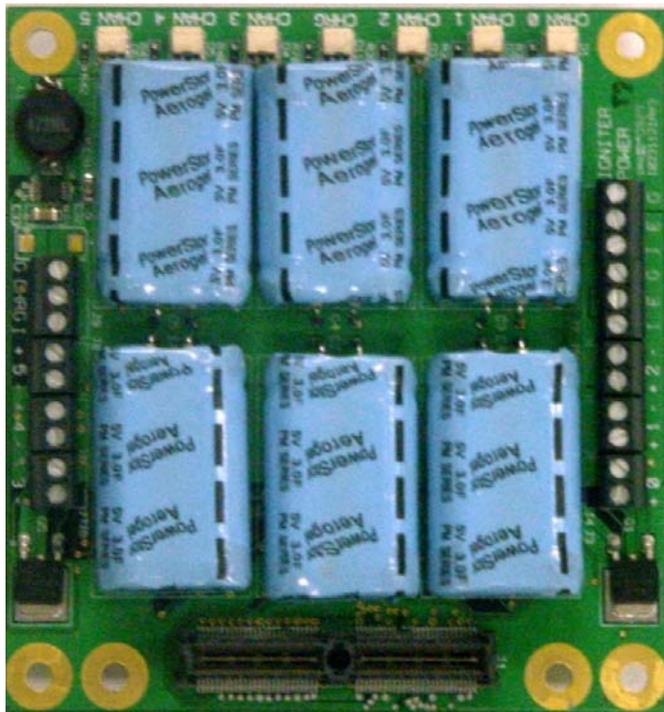
In this way a millivolt type pressure sensor with an output of 2 mv/volt, driven by the on-board 8V excitation supply and a capacity of 2,000 psi for measurement of oxidizer pressure could be defined as "Oxidizer Pressure (psi)" with a full scale output of 2,000 psi. The sensor is referred to by its name "Oxidizer Pressure" and the data presented in the appropriate format. Another thermocouple input could be defined as "Oxidizer Temperature (C)" and its output displayed

appropriately. This configuration information is stored in the module's EEPROM so that the module can be moved from one system to another or one location in a stack to another and it retains its personality.

6-Channel Pyro Module

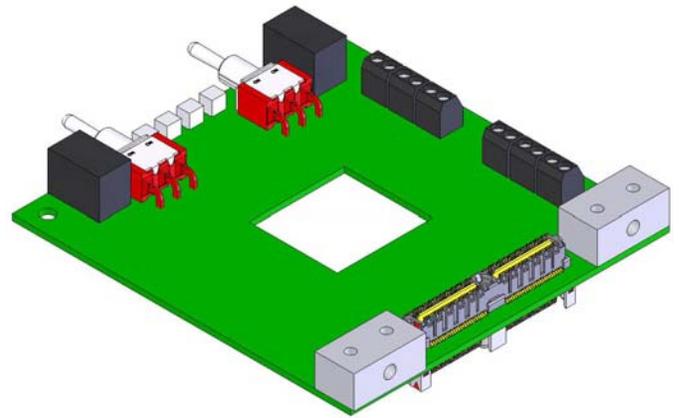
The 6-Channel Pyrotechnic Driver Module is available in two forms. The \$300 basic unit draws igniter power directly from an external battery (not the stack battery power controlled by the Base Module). Its capacity to drive multiple, 4 Amp (maximum) igniters is limited by this external battery (the Organic Lithium Battery Module is a suitable source).

The \$525 Energy Storage unit charges a set of on-board Super-Capacitors (18 Farads) with a maximum total energy of 190 joules. In this case system battery current draw is less than 1.5 Amps during charging. This version has a Green LED indicating full charge.



In either version the channel continuity can be read via the I2C bus I/O port and is visually indicated by a yellow LED for each channel. Each channel is independent, but may be connected in parallel to increase current supply, up to 24 Amps per module (up to three modules per stack).

Isolation Module



As previously mentioned, the \$125 Isolation Module sits between two stacks (nodes) to provide cooperative redundant control of shared components. The only signals that pass through the module from the top stack to the bottom stack are the fourth serial port ("Heart Beat") that is used by the two base modules to verify the other is functioning properly; and the DEFAULT and MASTER signals that automatically control which Base Module has sole control over a shared resource. For instance the 6-Channel Pyro Module has the option to use this mechanism to assure only one functioning computer can fire the pyrotechnic devices (driven by separate modules on each stack). If the Default Master Stack fails, it is prevented from attempting to fire its igniters, possibly erroneously.

The module also has active low and active high open drain power drivers for these signals to drive external control components. For instance a single antenna/power amplifier might support Telemetry Modules on each side of the redundant stack. Only the current Master could utilize this resource.

The module also has indicator LEDs for the "Heart Beat" in each direction and a Green LED to indicate when the primary stack is Master and a yellow LED to indicate when the secondary stack has taken over control.

In addition there are separate power switches and battery charger inputs for each stack. The charger inputs may be necessary because a redundant battery system will have isolation diodes in series that prevents the Base Module charger connector from driving the Battery Module.

Summary

The modular hardware architecture is designed to:

- Minimize cost of an entry level system
- Provide the highest performance available
- Create a building block "pay as you go" approach to system design

- Provide maximum flexibility and expandability to support what ever requirement the user may have
- Provide a range of devices and performance capabilities to meet the user's needs
- Support cooperative redundancy and control
- Allow additional modules, including custom user modules

Part III – System Software

The organization and architecture of the system software will be described here, along with the basic supporting software functionality. The Flight Control portion of the system will be described in the final part.

The system software can be divided into several functional parts:

- Module Enumeration and Initialization
- System Background Tasks
- System Configuration Control and Setup
- System Support Functions
- Flight and Post Flight Operations (discussed primarily in the next part)

Module Enumeration and Initialization

Upon power up or reset (control R from the keyboard) the system enters the Module Enumeration Process. This process interrogates the individual module SEEPROMs, starting with the Base Module and working its way consecutively up the stack to the last module. For each module it verifies the SEEPROM validity (it will reject an invalid module), performs initialization and self test functions and loads the user configuration information into the system configuration tables in SRAM.

Because each module deducts the resources it needs from the stack (serial ports, digital I/O ports, SPI chip selects, etc) and passes on the remaining resources to the next module, the system can map the local devices on each module to the system resources during this process, creating a plug-and-play architecture. If a module is inserted into a different place in the stack, it uses a different physical port or ports for its functions, but this is transparent to the rest of the system and the user. During enumeration, each module's location and name are printed followed by the results of its specific self tests.

The **CPU Module** first initializes its local (none stack resource) serial port (for ground support), setting the baud rate specified by the user in the SEEPROM. At this point it prints the first message "Scanning Modules...". It then performs a crc16 checksum on the program Flash contents to verify program integrity. It loads other control variables, including Time/Date base, the number of files the NAND Flash is to be partitioned into, node network ID, the use of the RS-485 network and the enabling of the Status LED. It then loads its portion of the Analog and PIO (digital port) control tables with the port definitions and initializes those ports to the proper state. Once this is completed it calibrates the Altimeter with the factory settings and then starts self tests in the following order; SRAM (multi-pattern), SD Card presence and size, NAND Flash validity

and bad block map, Accelerometer Self-Test and finally displays the Configured Ports on this module.

The **Telemetry Module** process cycles power to the modem, copies its port configuration to the PIO table then sets the connected serial port parameters and places the modem into its direct communication mode for setup and enquiry. This involves a special character sequence/timing to place the modem into the AT command mode. If the modem fails to respond properly to this process it is marked as a Modem Fault and skipped. Once communication has been established, the channel, RF Data Rate (900 MHz modem only), and transmit power as defined by its SEEPROM are set. Finally the modem firmware version is captured and displayed and the AT command mode is exited. Once this is completed the modem will begin duplicating the ground support serial port output (and respond to user input). The telemetry link can be used for any setup and command function ground support can.

The **GPS Module** process cycles power to the GPS receiver, copies its port configuration to the PIO tables and initializes its serial port. It then waits for the GPS to send its Sensor Status Information sentence. If not received within the 60 second timeout it is marked as a GPS timeout. If the sentence is received it is checked to verify proper sensor self test completion. At this point the sensor NEMA GPGGA (Global Positioning System Fix Data) and GPVTG (Track Made Good and Ground Speed) sentences are tracked by a background task.

The **Inertial Sensor Module** process cycles power to the sensor, copies its port configuration the PIO tables and initializes its SPI port. As a user option, it will then calibrate the sensor to correct for drift. The sensor output is sampled for the user specified period of seconds and the values are then averaged as a floating point value in degrees per second per axis offset. This offset is applied to the accumulated sensor position data in the sensor background task. Note that because the sensor is "cold" at this time the calibration effort is not optimal. The user has commands and/or the pre-launch options to perform the operation after the sensor has had time to temperature stabilize, generally 5-10 minutes. At this point the sensor is sampled 1,024 times per second by the Periodic Timer Interrupt.

The **ADC Module** copies its ADC channel properties to the ADC table in SRAM. At this point the channels are sampled 1,024 times per second by the background Periodic Timer Interrupt.

The **6-Channel Pyro Module** copies its portion of the PIO tables into SRAM and probes the I2C bus I/O port for the channel continuity status, displaying it on the ground support console. Note that the optional Super Capacitor Energy Storage bank is

not charged at this time. Doing so consumes a significant amount of battery power and the charge will not hold indefinitely, so the charging operation is generally performed by user command as part of the pre-launch preparation.

Once the Enumeration and Initialization process completes for all detected modules, control is passed to the main program that enables the Instruction Cache (Icache), sets the Status LED state to operating and displays the main menu. At this time all normal background tasks are executing.

System Background Tasks

The system background tasks:

- Handle Serial Port Interrupts
- Synchronize GPS Data
- Handle SD Card Interface Interrupts
- Perform periodic scans (1,024 times per second) of the ADC channels and Inertial Sensor registers
- Provide the Time Slice timing for the Flight Operations Sequencer
- “Heart Beat” communications between redundant nodes and failover logic

Serial Port Interrupts/GPS

The three general purpose stack serial ports and Base Module serial port (Ground Support) are all interrupt driven and largely identical. The receive buffers are 256 bytes (the UART hardware does not have a FIFO) and the transmit buffers are 1,024 bytes. In most cases the buffers are circular. Receive/transmit errors are counted but not reported. Receive characters are tested for a carriage return (end of line). At this point either the line count is incremented (allowing programs to determine if a complete input line is available) or if the port is the GPS port, the line is decoded as to type and if one of the three sentences of interest, it is transferred to a Sentence Buffer. In this case only the input character count is reset so the buffer is not circular. Applications that use the GPS data will clear the first character in the Sentence Buffer after use so the application knows when new data is available. This allows the GPS logging process to only record new data, regardless of the user specified logging rate.

SD Card Interrupts

SD Card Data is buffered in the 512Kbyte SRAM. When a complete 512 byte sector is ready, the DMA channel is triggered for the sector write. Completion of the write operation and cleanup is performed by an interrupt.

Periodic Scans/Time Slice Interrupt

The processor’s Periodic Interrupt is set to a Time Slice of 1,024 per second. The interrupt increments the Time Slice Tick counter which is used to initiate the next state sequence during Flight Operations. Each time the interrupt occurs it kicks off a series of 9 consecutive 10 microsecond timer interrupts. For eight of these interrupts, one ADC channel on each of the four potential ADC modules is read and the data stored, reading all 32 ports each time slice (the Base Module ADCs are all read during the Periodic Interrupt). Similarly, each of the first eight Inertial Sensor registers (sensor voltage, yaw rate, pitch rate, roll rate, yaw acceleration, pitch acceleration, roll acceleration, temperature) is read but only stored if the reading is new. The ninth timer interrupt is used to read the Inertial Sensor Status.

Heart Beat

The redundant hardware configuration (with Isolation Module) is supported by each of the two base modules sending a status message to the other every 8 time slices (128 times per second). If this heart beat message is not received within 9 time slices of the last, or the message indicates a node failure, the process will force the MASTER state to disable the failed processor.

System Configuration Control and Setup

The essential function of the UFC is to gather sensor data and act on it according to user instructions. When the data is sampled (up to 1,024 times per second) the Flight Sequencer checks the relevant Event queues for condition met (more information on this later in Flight Operations).

There are many system defined sensors that can be recorded and initiate actions:

- Altimeter
- Accelerometers (2)
- Battery Voltage
- Battery Current
- Pyro 0/1 Continuity (and Pyro Module Continuity)
- Inertial Sensor Position/Rate/Acceleration
- GPS data

The Altimeter actually can generate several different kinds of events based on past state(s). Flight operations tracks pre-launch, ascent, apogee, descent and landed state based on the Altimeter data.

The eight Base Module digital I/O ports, ADC Modules and future sensors can provide user defined data for recording and act as Events invoking Action during Flight Operations. Defining these sensors is one of the primary functions of System Configuration Control and Setup. These functions are:

- Configure Modules
- Calibrate Accelerometer
- Calibrate Altimeter
- Calibrate Inertial Sensor
- Report Devices
- Set Time/Date
- Port Connect (for use with outside setup software)

Configure Modules

The Configure Modules main menu item allows the user to configure each detected module in the stack. Note that when a new module is detected during factory setup that the SEEPROM is blank and the module is undefined. This forces a different Configure Module (Initial) process to select the module type, serial number, manufacture date, warrantee date and other module specific information. This configuration setup is only done once at the factory and is not available to the user.

This tool lists all detected modules and prompts the user to select the module to configure. The “escape” key can be used at any time to abort the process and leave the data in the module’s SEEPROM unchanged. Likewise the user is given a last chance to not save before making the changes permanent. Entering a return at any prompt leaves the parameter unchanged from its current value.

CPU Module

The CPU Module has the following user configurable parameters:

- Number of Files in NAND Flash (the flash is partitioned into identical sized files for flight logging, default 16 which is 16 Mbytes per file). Note that changing this parameter may make data currently stored in the log unavailable.
- Ground Support Serial Port baud rate (default 38,400)
- Enable UFC Network (default No, single node configuration)
- Enable Status LED (default No, use digital port 8 for user defined function)
- Modify Digital I/O Port Assignments

The eight user defined ports and the Base Module pyro ports (control output and continuity input) may be modified by the user. Only the name of the port and its state can be changed for the pyro ports, their functionality cannot be changed. After selecting the port for configuration the user enters the new port name (or leaves it unchanged). The important thing to remember is that the port name is used in Flight Operations to refer to the port and select it for Events (input and outputs) and Actions (outputs only). This is why the Pyro ports are available for renaming. A logical name for the port and its function will

help the user create clear and concise Flight Operations Scripts. For instance, the names might be changed as follows:

```
Pyro 0 Continuity: Drogue Continuity
    1 (Connected) State: Drogue Ready
    0 (Open) State: Drogue Open
Pyro 0 Output: Drogue Release
    1 (Ready) State: Drogue Off
    0 (Fired) State: Deploy Drogue
```

With this naming convention the Drogue (assuming it is connected to Pyro 0) igniter state can be check for continuity (Event) by “Drogue Ready” or “Drogue Open” and the channel fired (Action) by “Deploy Drogue”. The channel could just as well be the Main Chute or some other function.

The other eight ports are fully user definable. In addition to port and state name are:

- Input or Output: How is the port to be used?
- Default High or Low (output only): What is its initial output state to be?
- Open Collector (output only): If an output, can it be shared (Open Collector)?
- Pull-up: If an input or an Open Collector output, should a pull-up resistor be enabled to set its open state to high?

With these configuration controls the inputs might connect to switches that provide meaningful feedback as to the state of the system; break wire (on launch pad or attached to lower stage), hatch closed, chute in place (a break wire or switch indicating when the drogue or main chute has been attached or it has been deployed), etc.

Outputs, with the appropriate buffers, can be used to initiate user functions such as turn on or off cameras, experiments, release chute tethers after landing, ignite a flare or smoke source, etc. Remember this outputs can be controlled by any Event or combination of user defined Events including remote commands.

Telemetry Module

The Telemetry Module setup is largely the same between the 900 MHz North American Version and the 868 MHz European version:

- Low Speed/Long Range Mode (900 MHz Only): 9,600 or 115,200 baud (the lower speed extends the range at the sacrifice of less data transmitted)
- Low Power Mode: from 1mw to the maximum output of the transmitter (1Watt for 900 MHz, 315 mw for 868 MHz). Lower power will reduce range but also power consumption and modem temperature rise

- Channel (900 MHz) or Destination Address High/Low (868 MHz): the 900 MHz modem supports 10 frequency hopping channels to avoid conflict with nearby units. The 868 MHz modem uses a network type destination address (64 bits) to direct its end point (which will have this modem as its destination)

GPS Module

The GPS modules do not have any user defined parameters.

Inertial Sensor Module

The Inertial Sensor can have automatic calibration on power up (enumeration) enabled. This samples the output for the user specified number of seconds and creates an offset value to be used to correct for drift. It is a good idea to do a short calibration to provide an initial attitude reference, but for Flight Operations the system should be allowed several minutes to come to a stable temperature and the calibration should be run for 30 to 150 seconds to get optimal performance. Calibration can also be done by the Calibrate Inertial Sensor menu item and also automatically as part of launch preparation.

ADC Module

This ADC Module configuration allows each of the eight analog input channels to be defined according to the attached sensor. In all cases the channel name is used as the identifier for logging (Action) or testing the output of the sensor (Event). The user should use a clear and useful name for each channel and can include in parenthesis the unit of measure (text within the parenthesis is ignored as part of the name but is included for display purposes). There are three types of sensors defined:

The **standard input** may or may not have an amplifier on the front end. The user provides a scale factor and offset to present the data in the appropriate manner. For instance if a voltage type pressure sensor with an input range of 0-2,000 psi and an output range of 1-5V is connected, the scale can be set to 500 psi per volt. Therefore the scale factor should be:

$$(4 / 5.32738) * (2000 / 65535)$$

That is the output range of the sensor (4V) divided by the full scale input range of the channel times the sensor input range (2,000) divided by the full scale count of a 16-bit channel (65,535). The offset applied after scaling is -500 because the 1V starting output would otherwise be interpreted as 500 psi.

Scaled inputs are similar to standard inputs except that their value is related to the excitation voltage supplied by the module (or other source). These are proportional bridge type sensors found in lower cost pressure sensors, load cells and strain

gauges. In this case the sensor output will rise proportionally with an increase in excitation voltage. The user specifies a scale factor (usually no offset) to represent the 1 Volt excitation voltage input range and the system will automatically account for the true excitation voltage (channel 8 configured to monitor this voltage, which could be an outside source).

Thermocouple inputs support type K sensors over a range of -180°C to +1,200°C. The system knows the correct scale factor and the user only has to enter a current reference temperature for the system to automatically calculate the offset.

6-Channel Pyro Module

This module does not have any user defined parameters.

Calibrate Accelerometer

This main menu item assumes the rocket is in the launch attitude (pure vertical) and sets the accelerometer references to -1g. This information is saved in the SEEPROM so it will be used when powered up.

Calibrate Altimeter

This main menu item allows the user to enter a known current altitude, or calibrates the current altitude according to the GPS, which must have a valid set of satellites before this operation can succeed. This information is saved in the SEEPROM so it will be used when powered up.

Calibrate Inertial Sensor

This main menu item samples the output for the user specified number of seconds and creates an offset value to be used to correct for drift. It is a good idea to allow several minutes for the system to come to a stable temperature and the calibration should be run for 30 to 150 seconds to get optimal performance. Calibration can also be done automatically as part of launch preparation.

Report Devices

This main menu item scans the attached modules and allows the user to select the module (or all modules) to report on. The report in general will display the module location (number), name, revision, serial number, warranty start date, and any other module dependant information.

Base Module

The Base Module reports:

- Firmware revision and network ID

- Internal Flash and SRAM size
- External NAND Flash and SRAM size
- SD Card size if present
- Internal Battery (coin cell) Age
- Barometric Altitude and Temperature
- 8g and 50g Accelerometer values
- Primary Battery Current and Voltage values
- All defined digital I/O port states, plus the Base Module Pyro ports

Telemetry Module

The Telemetry Module reports:

- Modem Enable state: Enabled or Disabled
- Firmware version and date
- Modem internal voltage, temperature, last received signal strength and transmit power
- Received Good packets, Receive Errors and Transmit Errors
- Channel and RF Data Rate (900 MHz only) or Destination Address (868 MHz only)

GPS Module

The GPS Module reports:

- GPS Enable state: Enabled or Disabled
- GPS Lock LED state: On or Off (reflects the state of the LED as it blinks, not the state of Lock)
- Firmware version and checksum test result
- Coordinated Universal Time (UTC), Satellites in use and Lock state
- Longitude, Latitude and Altitude

Inertial Sensor Module

The Inertial Sensor Module reports:

- Sensor Enable state: Enabled or Disabled
- Sensor Power Supply, Temperature and Status
- Yaw Position, Rate and Acceleration
- Pitch Position, Rate and Acceleration
- Roll Position, Rate and Acceleration

ADC Module

The ADC Module reports:

- The current sensor value for each channel. It is displayed according to the user defined name/units.

6-Channel Pyro Module

The Pyro Module reports:

- Charge State of the Super-Capacitors (if present)

- Channel and current continuity state name (status)

Set Time/Date

The Set Time/Date main menu item is used to set the current date and time. It is battery backed up and is used to display the current time at the top of the main menu. It is logged when the Flight Operations goes to the Ready state (launch sequencer started) which is used as the file identifier in the NAND Flash or SD Card Flight Log.

Port Connect

The Port Connect main menu item is used to transparently connect the ground support serial port to any of the three stack serial ports (i.e., Telemetry Module, GPS Module, etc.). This allows the module manufacturer's setup software to communicate with the module to perform setup operations that are not support directly by the UFC software.

System Support Functions

The primary system support function is the **Download Firmware Update** main menu item. It reads the update file from the ground support serial port and saves it in the SRAM, checking each line for errors. If there is an error it sends a negative acknowledge back to the connected host for retransmittal. The file format is Motorola S-Records. Once the file download is completed a final CRC check is performed to make sure the new software has no errors. It then takes just a few seconds for the system to "Flash" the new firmware, after which the system automatically resets.

Flight and Post Flight Operations

Details on the Flight scripts and their function in providing user control of the Flight Operations are provided in the last part. This section will provide insight into the basic operation and underlying structures used to:

- Load the User Script
- Simulate Flight Operations
- Perform actual Flight Operations
- Record and recover Flight Data and perform post flight data reduction and analysis

Load Script

The Load Script main menu item is used to download a text file (script) that will control Flight Operations, both actual and simulated. The script is created using any text editor that saves the file in plain text format. The script is stored in the Base Module's EEPROM and thus can be loaded at any time prior to use. When a new script is to be loaded the current script is

displayed. The user has the option of keeping the current script (abort the download) or continuing. When a new script is to be downloaded the terminal emulator being used to connect to the ground support serial port must have a text file transfer utility to send the file. After the file is sent the user enters the escape key to terminate the transfer.

The script has several functions that will be detailed in the next part, but the general functions are:

- Perform pre-launch system checks
- Perform pre-launch sensor calibration
- Select the Altimeter source for flight (barometric sensor or GPS). Note that GPS is not useable above 60,000' unless the COCOM limits have been removed (available on Internal GPS Module only) and the barometric sensor is limited to 100,000',
- Define Events, both simple and complex based on Flight state, sensor inputs, user defined flags, remote commands, etc.
- Define Actions resulting from Events

Simulate Launch

The Simulate Launch main menu item executes the launch sequence as described next in a special mode. It uses the same software and user script, but the script may have special simulation actions (directives) that substitute for real world Events. That is Actions to simulate acceleration or constant velocity provide a means to simulate the launch, ascent, apogee, descent and landing. All other flight Events are recognized and Actions executed just as if this were an actual flight. These directives are ignored when the actual flight is in progress, so there is no need to have a different script for test and actual operations.

This mode also ignores certain error conditions, such as lack of pyro channel continuity. This allows the simulation to occur without having to enforce the actual launch environment. However at the user's option the igniters could be connected to verify proper operation of the circuits or the charge size. Syntax errors are never ignored and simulation is a good way to make a first pass test of the script construction.

Prepare for Launch

This main menu item initiates the launch process and flight sequence. It reads the script in from SEEPROM, performing proscribed system checks, calibration and pre-launch preparation commands immediately. All other script elements define flight states (Event/Action sequences) that are interpreted and stored in a series of Event Queues and Action elements. There is a queue for each general class of event:

- ReadyQ (after successful completion of launch preparation and approval of the user to proceed with the launch sequence)
- TimerQ for any Event that occurs that then needs a time delay applied (the timer Events decrement each event timer at the beginning of each time slice and when the timer is decremented to zero, the event is now recognized)
- StageQ is an Event that signifies the nth acceleration occurrence starting with the launch. Launch is the first Event (Stage 1) and the next acceleration, presumably an air-start, would be the second (Stage 2) or subsequent Event
- AscentQ is an Event that passes through the specified altitude during the Ascent phase of the flight
- ApogeeQ is the Event of achieving the peak altitude and beginning the Descent phase of the flight
- DescentQ is an Event that passes through the specified altitude during the Descent phase of the flight
- LandQ is an Event defined as no motion at the end of the Descent phase of flight for 10 seconds
- RemoteQ is a match of a predefined string (part of the script) with a character sequence received from the telemetry uplink or another node in the network
- InputQ is the change to a defined state of any of the 40 user accessible digital I/O ports
- ADCQ is the value of a sensor going above (>) or below (<) the specified value. The value is in the form of that specified by the sensor setup (Configure Modules or system defined)
- TempQ is the value of the temperature sensor on the Base Module going above (>) or below (<) the specified value
- SpeedQ is the Altimeter vertical speed (in feet per second) going above (>) or below (<) the specified value. The value can be positive (Ascent phase) or negative (Descent phase)
- RollQ is an absolute Roll rate in degrees per second (unsigned) going above (>) or below (<) the specified value
- PitchQ is an absolute Pitch position, relative to the position at the time of calibration, in degrees (unsigned) going above (>) or below (<) the specified value
- YawQ is an absolute Yaw position, relative to the position at the time of calibration, in degrees (unsigned) going above (>) or below (<) the specified value
- IfQ is the set (IF) or cleared (IFNOT) state of the user defined flag that was set or cleared by a previous Event definition being satisfied (Action)
- KnotsQ is the GPS horizontal speed (in knots) going above (>) or below (<) the specified value

Each queue points to a linked list of elements (zero if no more queue elements). The Event elements consist of:

- Event Link: link to next Event in the queue if any
- Argument: the user specified value that qualifies the Event
- Timer: the timer value in Ticks (1,024 per second) that must pass after the Event before the associated Action can take place
- Action Link: points to the Action to be performed once all associated Events have occurred
- SimA Action: specifies a new simulated acceleration constant to be used during Simulate Launch. It remains in effect until a new SimA or SimV action
- SimV Action: specifies a new simulated vertical velocity constant to be used during Simulate Launch. It overrides SimA
- Set Action: sets one of 26 user defined flags (A-Z) for later reference
- Clear Acton: clears one of 26 user defined flags (A-Z) for later reference

The Action elements contain the following information:

- Event Count (total): the number of Event elements linked to this Action
- Current Events: counts down as each Event linked to this Action is satisfied in a time slice. The Action is taken when this number is decremented to zero (last Event has simultaneously occurred in the same time slice).
- Action: is the link to the Action function to be executed
- Parameter 1: a value used by the Action subroutine
- Parameter 2: optional second value used by the Action subroutine

At the beginning of each time slice, each Event queue is scanned to determine which linked Actions have now been completed (Current Events equals zero). If they are complete, the Event element is removed from the queue. If not all events were satisfied in the previous time slice, the Event Count is copied back into the Current Event count, restarting the qualification process for the new time slice. In this manner simple and complex user defined Events control the execution of user defined Actions during flight operations.

There are many Actions that can be specified:

- ADC Action: set the current sample rate for logging and telemetry for an individual Analog channel
- Input Action: set the current sample rate for logging and telemetry for all digital I/O ports
- Temperature Action: set the current sample rate for logging and telemetry for the Altimeter Temperature
- Altimeter Action: set the current sample rate for logging and telemetry for the Altimeter
- Accelerometer Action: set the current sample rate for logging and telemetry for the Accelerometers
- GPS Action: set the current sample rate for logging and telemetry for GPS
- RPY (Roll/Pitch/Yaw) Action: set the current sample rate for logging and telemetry for the Inertial Sensor
- Output Action: specifies the new state for the selected digital output port (this includes the Pyro channels)
- End Action: performs flight operations cleanup (flushing data to the flight log) before shutting down operations (including all logging)

Once the Ready state is entered, Flight Operations are in control of the system until the specified End Action terminates it. At the start of the first time slice, the Ready Queue is exhausted, with the timed Actions moving over to the TimerQ. It is never scanned again. After that and all subsequent time slices the system formats the log buffer with the current processor ID and time stamp. This buffer is then used for logging all messages to the on-board flight log and telemetry. After the Queue check refreshes the Action Event counts or removes the remaining related completed Action's associated Event elements, the system scans each possible sensor in sequence, checks for sensor related Events and logging currently enabled sensor data or Event/Action messages.

Log or Read Stored Flight Data

During Flight Operations (between Ready and End) most Events, some Actions and all user specified sensor data are logged to the on-board Flight Log and transmitted over the Telemetry system. The data stored/send is identical in format and differs only in the extent of user specified data, which is independently controlled. The first part of the data always starts with the Ready record specifying ground level and time/date of flight (or simulation). Following this first packet are additional packets that contain the user or system configuration for all sensors. This information is essential for programs interpreting the data to know the port name, state names, scale factor, offset, etc., of the data herein. Thereafter follows the flight operations generated messages and ending with the final End message.

The flight telemetry data can be captured real-time with the terminal emulator's capture text file function which should be started before the user gives the go ahead for Flight Operations (in order to capture the configuration information). Generally this data will consist of the most essential flight data, particularly that essential to monitoring flight progress, recovery and if necessary failure analysis. At a later date a Windows application will provide users with a real time display of this data.

Post flight, the user can invoke the Read Stored Flight Data main menu item to select the required data from the file list (by

time/date) and upload it to the host PC the same was as is done with telemetry, by enabling the terminal emulator's capture text file function.

In either case, the data is formatted such that it is easily loaded into an Excel template that is designed to interpret all the data (using the sensor configuration information at the front of the data). Once the data has been presented in processed form in Excel, Excel's charting facilities can be used to present the selected data in the way most meaningful to the user (template graphs are provided for the most common areas of interest). In this way the user is free to present their data in what every way is appropriate and relate the data as it applies to their configuration.

The data packets (lines) are comma delimited and contain (in order):

- Processor ID (when data from multiple systems are to be combined)
- Time Stamp (seconds, ticks)
- Packet Type (ADC, Digital I/O, GPS, Inertial Sensor, etc.)
- Type dependant data

Part IV – Flight Operations

As we have seen previously, the UFC-3 has a complete and complex set of high performance modules to support various flight operations and data acquisition requirements. The supporting software is designed to allow complete freedom in defining the use and meaning of user defined sensors and to get the most out of system sensors and resources. All of this requires active participation by the user in configuring and specifying the control functions of these devices. Without that user input, the system does nothing.

The functionality of the software and systems it controls has already been thoroughly discussed. What will be discussed

here are the various ways this capability can enhance the safety and reliability of the flight and maximize the return of data. It is up to the user to create innovative new ways to use this system to provide useful research data from their successful flight(s).

At the heart of all this is the user defined launch script. What follows are examples of these scripts or portions of them that provide insight in how to effectively use them.

Script Example supporting Simulation

Don't be intimidated! This script is meant to show many of the possible uses, but the user can start with a much less complex scenario (these scripts assume a full set of modules):

```
;  
;  
; UFC-3 LAUNCH/SIMULATION SCRIPT  
; (assumes CPU, Telemetry, GPS and Inertial Sensor are present)  
;  
; Calibrate Functions  
;  
cal alt ;Calibrate Altimeter  
cal accel ;Calibrate Accelerometer  
cal 3-axis 120 ;Calibrate 3-Axis Sensor  
;  
; Check Functions (Check failure will abort launch but not simulation, provides diagnostic message)  
;  
check accel ;Check Accelerometer Functionality  
check altimeter > 1500 ;Check Altimeter (reasonable reading?)  
check altimeter < 2500  
check temperature < 50 ;Check Temperature  
check pyro 0 ;Check Recovery Pyro Continuity/Ready  
check pyro 1  
check GPS ;Check GPS for Lock  
check primary battery voltage > 9; Check battery voltage  
check drogue ready ;Redundant for check pyro 0 but is an example of user I/O test  
check drogue off  
;  
; Sample Rates (most data sent once per second over Telemetry to limit data rate), starting at Ready (user could reduce or  
; eliminate non-essential readings during descent by repeating directives at apogee)  
;  
ready alt 8 8 ;Log Altitude/Acceleration 8 times per second  
ready temp 1 ;Log Temperature  
ready accelerometer 256 8 ;Log Acceleration  
ready 8 g accelerometer 1 ;Redundant for ready accelerometer, example of user analog input  
ready 50 g accelerometer 1  
ready primary battery voltage 1;Log Battery Voltage/Current  
ready primary battery current 256 1 ; Note that current monitoring can indicate pyro function so we use increased resolution  
ready input 4 1 ;Log Digital Inputs  
ready gps 5 5 ;Log GPS  
ready 3-axis 820 8 ;Log Inertial Sensor data (internally at maximum)  
;  
; Simulate 2 Stage Launch (ignored by Prepare for Launch)  
;  
ready sima 321 ;Launch Booster at 10G Simulated for 2 seconds  
ready plus 2048 sima 0  
if b sima 321 ;Sustainer at 10G for 2 seconds
```

```

if b plus 2048 sima 0
apogee simv -80 ;Simulate Drogue at Apogee
descent 800 simv -20 ;Simulate Main at 800'
;
; Flight State Machine
;
Ready booster video on ;Turn on launch camera
stage 1 set s ;Staging Flags
stage 2 set t
if t booster video off ;Change camera when staging
if t sustainer video on
fps > 500 set a ;Flag speed greater the 500 fps
ascent 4000, if a, fps < 400, roll < 15, pitch < 15, knots < 25 set b ;Condition for Air-start
if b airstart ;Air-start is a user defined output channel to fire the second stage
apogee deploy drogue ;Release Drogue at Apogee
apogee sustainer video off; and change cameras
apogee chute video on
remote drog deploy drogue ;Backup remote Drogue release
fps < -100 deploy main ;Backup action if Drogue doesn't deploy, try Main
descent 800 deploy main ;Release Main at 800'
remote main deploy main ;Backup remote Main release
land end ;End when Landed
land chute video off

```

Note that the case does not matter (everything is internally converted to upper case). The various keywords can be abbreviated so long as they do not conflict with other keywords. Commands have been grouped to help distinguish one type of command from another. The order of the commands is largely irrelevant, the events determine the timing. Comments can (and should) be entered anywhere to document the intent. Some of the sensor names are not the default ones, but rather examples of logical names for the actual functions performed.

Notice the use of flags “a”, “b”, “s” and “t”. “s” is set on launch and “t” is set when the air-start is detected by the accelerometer. “a” is set when vertical velocity exceeds 500 fps and “b” is set to indicate conditions have been met for the air-start. In this case “a” must be set (500 fps or greater vertical velocity) before “b” can be set based in part on velocity falling below 400 fps. This assures we don’t start looking for “fps < 400” before “fps > 500” has occurred, which would otherwise occur immediately. The purpose of this sequence is to perform the air-start once the vehicle slows down to a certain point, and has the correct attitude, before the air-start. If any of these conditions do not all simultaneously occur, the air-start will not occur. Setting the “b” flag as the Event Action simplifies its use in more than one instance (it is also used to control the simulation). The “t” flag is another example of this (the Events could be repeated with the additional Actions to achieve the same result).

Notice that the statement to set “b” (air-start) has both a condition for attitude (“roll < 15, pitch < 15”) but also one for horizontal velocity (“knots < 25”). This is a backup (or if the Inertial Sensor is not present an alternative) method of assuring the vehicle is in the proper state for the air-start.

The sequence “fps < -100 deploy main” is an example of how the user might provide alternate actions to handle failure scenarios. If the drogue chute should keep the descent rate below -100 fps, then the statement is not invoked. However if the drogue indeed failed to deploy properly a reasonable (although normally undesirable at this point) action would be to deploy the main chute.

Another way to cover deployment failure(s) is to use the remote commands. The telemetry system is two-way and a 1-4 character message received can initiate an action, such as deployment override.

There are many other fault checks or backup actions that might be specified depending on the system design. The goal of this presentation is to expose the reader to these possibilities and inspire them to create new and innovative uses for these advanced functions, enriching their experience, increasing their return on investment and helping to assure a successful venture.

Summary

This series has been presented to achieve several objectives:

- Help readers understand the fundamentals of Flight Electronics
- Help readers understand the underlying hardware components and their importance in providing a reliable system with the performance needed to maximize data return
- Help readers understand the underlying software architectural features that create the flexibility to allow them to configure the system how they want, rather than being restricted by a rigid set of rules and capabilities designed in by the manufacturer
- Expose a pathway the user might utilize to create whole new applications and research areas of interest previously outside their ability. Amateur rocketry is all about creativity and exploration!

For more information email

Tom@BlackMagicMissileWorks.com