Universal Flight Controller/Datalogger with Model Rocketry Applications

by
John S. DeMar
NAR 52094

August 1993

http://web.syr.edu/~jsdemar
jsdemar@syr.edu

Table of Contents

Abstract (R&D Report Summary)

The Universal Flight Controller/Data logger (UFC) is a small light-weight computer designed for multiple applications in the field of advanced model rocketry. This NAR research & development report describes the project from initial conception, through the design and prototyping stages of both the computer hardware and software. In addition, several possible applications and experiments are described and one example is used to gather actual results.

The primary goal of this project is to build a multi-purpose system that is easy to program and operate by the average person without the need for a computer or electronics background. Another important requirement is a good cost/performance ratio compared to using several commercially available systems (timers, dataloggers, altitude payloads, etc.). Secondary goals include: small size, low mass, expandability and reliability.

The use of electronic payloads in model rocketry has become more popular over the years with the availability of higher power motors. An early idea was a transmitting beacons which uses a radio signal to locate a model rocket after landing; a similar device uses a loud audible "beeper". The small size of these "passive" electronic payloads makes them easy to lift using small motors and simple models. However, more complex designs are needed to control aspects of a flight or to record data during the flight.

Two popular uses of "active" electronic payloads are the staging timer and the ejection timer; they typically use low-cost timer chips but some are based on more accurate crystal clock designs. These timer applications are most useful in larger model rockets which use composite rocket motors. Unlike the "black power" motors, composite motors need an active ignition system to ignite an upper stage at a pre-set time. Also, most composite motors are available in just three widely spaced ejection delays that may not be accurate enough for a safe ejection near apogee. A timer circuit is often used to trigger an ejection charge at a pre-set time that is different than the motor's ejection delay. Often they are used as a redundant backup for models with a great deal of money and time invested in them.

As advanced high power rocketry is becoming more popular and reusable technology is bringing down the price of larger motors, rocketry experimenters are finding it easier to lift more complex payloads. Recently, altimeters and dataloggers have become popular, especially with hobbyists who have a background in electronics. However, the typical rocket flyer may find it too difficult to work with a computerized payload which requires custom computer programming. Other than the simple one-purpose payloads, a project which requires multiple inputs, multiple outputs and moderate data logging speed is beyond the scope of the rocketry hobbyist.

At the present time, someone interested in designing a data logging experiment would first locate a microcontroller capable of handling the number of inputs and the sampling rate needed. The board would have to be reasonably small and light-weight, and operate from low-power batteries for a period long enough to prepare, fly and recover the model, then retrieve the data and permanently store the results. The system would need to be low-cost (under $200) and further reduce the cost and complexity by incorporating other electronics (staging timers, ejection timers, etc.) into the main computer system. Then, most difficult of all, the system would require no

computer programming beyond simple instructions and would require no special equipment (such as an EPROM programmer). Also, the experimenter would need a program capable of analyzing the raw data by converting it to an understandable format (graphs, charts, etc.).

As the preceding paragraph describes, most rocketry enthusiasts quickly realize that computerized data logging and controlling is beyond their capabilities. Unfortunately, many are missing out on one of the most challenging aspects of advanced model rocketry.

This R&D project describes the design a universal computerized payload which makes it easier for the average person to experiment with active electronics in model rocketry. The result is a hardware and software prototype which could be further refined and offered as a low-cost kit (through NARTS, for example).

Introduction

The Universal Flight Controller/Datalogger (UFC) is a small light-weight computer designed for multiple applications in the field of advanced model rocketry. This NAR research & development report describes the project from initial conception, through the design and prototyping stages of both the computer hardware and software. In addition, several possible applications and experiments are described and one example is used to gather actual results.

The primary goal of this project is to build a multi-purpose system that is easy to program and operate by the average person without the need for a computer or electronics background. Another important requirement is a good cost/performance ratio compared to using several commercially available systems (timers, dataloggers, altitude payloads, etc.). Secondary goals include: small size, low mass, expandability and reliability.

Since the ultimate purpose of this R&D report is its educational value to NAR members and the model rocketry community, I have organized the report to show the design process followed throughout the project. Although the report shows the details necessary to produce the hardware, a full understanding of the design requires an advanced background in electronics.

The first section, *Concept and Requirements*, describes the need for an on-board computerized system in model rocketry and the problems the average hobbyist may face when working with currently available hardware. This section concludes with a summary of the requirements for a system which would solve these problems. The second section, *System Description*, defines the basic components of the system and how they interact with each other.

Next, *Hardware Design*, details the electronic system; this section requires the reader to have an understanding of microprocessors, digital electronics and analog electronics. The section on *Software Design* covers both the system software which controls the UFC and the user program which runs on a personal computer. Again, to understand the details of this section, the reader will need a background in microprocessors and computer software design.

The next section, *System Integration*, describes how the components physically connect and work together. The *User's Guide* section explains how the software is used to "profile" a sample application and how to "retrieve" the logged data after a flight.

The final section, *Application Ideas*, suggests a few possible uses for the Universal Flight Controller/Datalogger. Some examples are: simple ejection timer; staging timer with feedback; altitude profile datalogger using pressure sensor; and a multi-axis acceleration datalogger.

Appendix A is the assembly language source for the UFC system software. Appendix B is the 'C' language source for the "Profiler/Retriever" software. Appendix C shows a cost estimate for producing small quantities of the system.

I. Concept and Requirements

The use of electronic payloads in model rocketry has become more popular over the years with the availability of higher power motors. An early idea was a transmitting beacons which uses a radio signal to locate a model rocket after landing; a similar device uses a loud audible "beeper". The small size of these "passive" electronic payloads makes them easy to lift using small motors and simple models. However, more complex designs are needed to control aspects of a flight or to record data during the flight.

Two popular uses of "active" electronic payloads are the staging timer and the ejection timer; they typically use low-cost timer chips but some are based on more accurate crystal clock designs. These timer applications are most useful in larger model rockets which use composite rocket motors. Unlike the "black power" motors, composite motors need an active ignition system to ignite an upper stage at a pre-set time. Also, most composite motors are available in just three widely spaced ejection delays that may not be accurate enough for a safe ejection near apogee. A timer circuit is often used to trigger an ejection charge at a pre-set time that is different than the motor's ejection delay. Often they are used as a redundant backup for models with a great deal of money and time invested in them.

As advanced high power rocketry is becoming more popular and reusable technology i s bringing down the price of larger motors, rocketry experimenters are finding it easier to lift more complex payloads. Recently, altimeters and dataloggers have become popular, especially with hobbyists who have a background in electronics. However, the typical rocket flyer may find it too difficult to work with a computerized payload which requires custom computer programming. Other than the simple one-purpose payloads, a project which requires multiple inputs, multiple outputs and moderate data logging speed is beyond the scope of the rocketry hobbyist.

At the present time, someone interested in designing a data logging experiment would first locate a microcontroller capable of handling the number of inputs and the sampling rate needed. The board would have to be reasonably small and light-weight, and operate from low-power batteries for a period long enough to prepare, fly and recover the model, then retrieve the data and permanently store the results. The system would need to be low-cost (under $200) and further reduce the cost and complexity by incorporating other electronics (staging timers, ejection timers, etc.) into the main computer system. Then, most difficult of all, the system would require no computer programming beyond simple instructions and would require no special equipment (such as an EPROM programmer). Also, the experimenter would need a program capable of analyzing the raw data by converting it to an understandable format (graphs, charts, etc.).

As the preceding paragraph describes, most rocketry enthusiasts quickly realize that computerized data logging and controlling is beyond their capabilities. Unfortunately, many are missing out on one of the most challenging aspects of advanced model rocketry.

The concept for this R&D project is to design a universal computerized payload which makes it easier for the average person to experiment with active electronics in model rocketry. The result will be hardware and software prototypes which could be further refined and offered as a low-cost kit (through NARTS, for example).

The main requirements are summarized as follows:

- The system must simplify the "programming" by hiding complexities from the user through a specially designed computer program (compatible with most personal computers, especially inexpensive portables for use in the field).
- The system must be capable of sampling up to eight analog inputs (sensor data) at a rate of no less than 500 samples per second and control up to four timed outputs with an accuracy of at least 1/100th of a second.
- Up to four external events (such as "start") must be sensed by the system and be capable of triggering other events (such as datalogging and timers).
- The physical circuit board must be less than 2.5 inches wide (to fit inside a moderately small high-powered model rocket) and weigh less than four ounces (for reasonably low-cost flights using at least D motors).
- The system must use low power, low weight rechargeable batteries capable of no less than a half hour of continuous operation.
- The main board must be expandable to accept various sensors and provide for memory expansion for more complex applications.

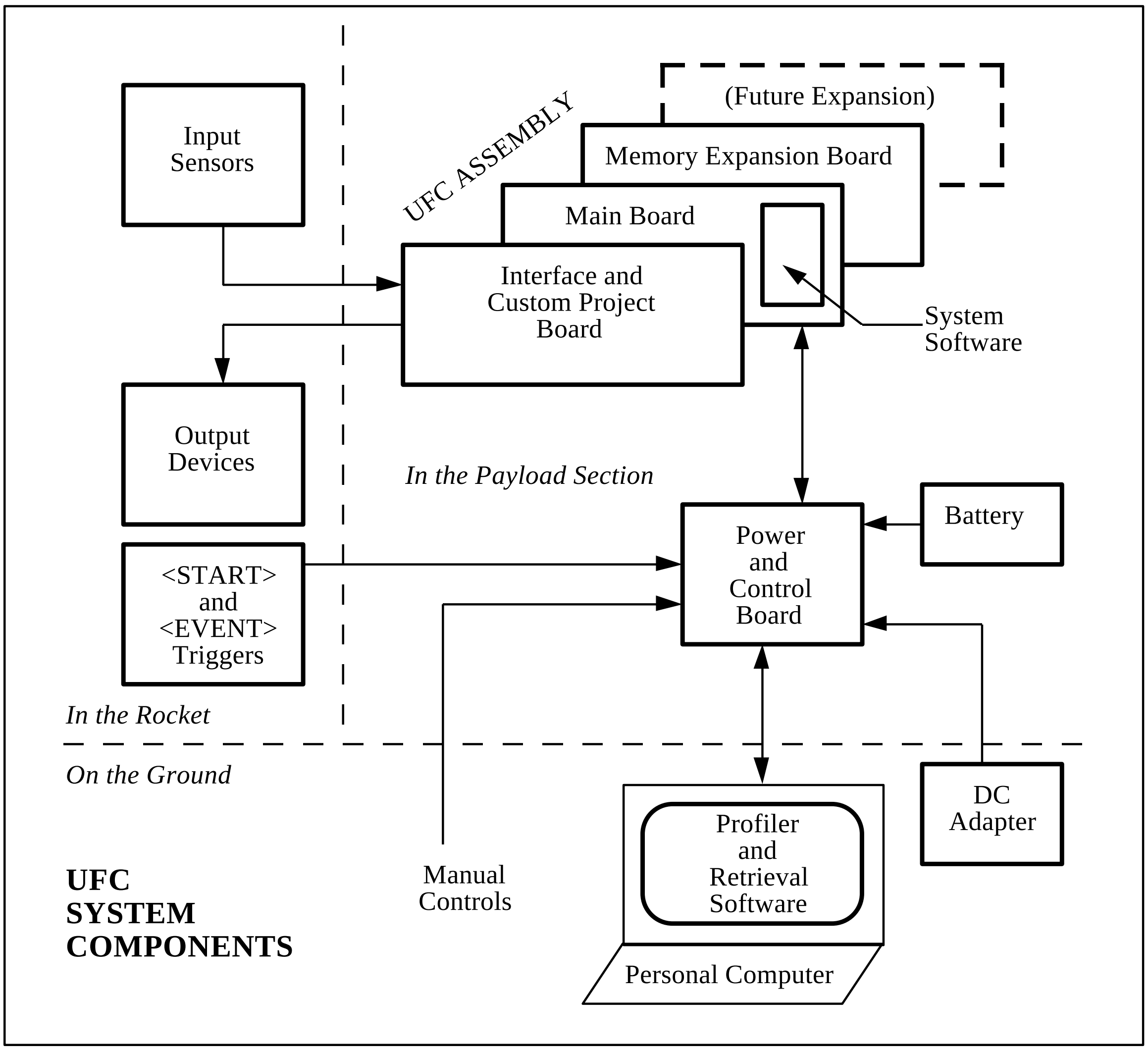
This project required the following equipment & supplies:

- Electronic prototyping tools such as a soldering iron and wire cutters.
- Electronic test equipment: minimum 20MHz dual-trace oscilloscope.
- Electronic parts catalogs to select and order components.
- The electronic components to build the prototype.
- A Personal Computer and compatible development software including:
  - Motorola 68HC11 microcontroller assembler and debugger, free from Motorola for PC or Macintosh (Macintosh was used for that part of the project);
  - a PC-compatible 'C' language compiler (Microsoft C);
  - programming tools for GEM (Graphics Environment Manager from DRI).
- EPROM programmer and EPROM eraser. (An EPROM is an Erasable Programmable Read-Only Memory chip which requires UV light to erase and a programming unit to store the microcontroller software in the chip's memory.).
- Software to produce design documents and drawings (Apple Macintosh with Microsoft Word and MacDraw was used).

II. System Description

After the initial requirements were defined, the project began with an overall drawing of the basic components needed for the final system. This type of description simplifies the project into groups which define the work ahead without the distraction of unneeded details. Another advantage is to have a "road map" to refer to as the project develops and the various components are designed to fit into the whole system.

The following diagram represents the complete Universal Flight Controller/Datalogger system. The top portion of the drawing contains the components which are physically mounted in the model rocket and operate during flight. This section is further divided into the parts which are contained in the payload section and other parts which may need to be placed elsewhere in the airframe (switches and sensors, for example). The bottom section of the drawings represents the components which are used on the ground to interact with the system before and after flight.



UFC Assembly

The heart of the UFC system is the circuit board assembly which includes: the Main Computer Board that controls the other components and contains the system software; the Interface Board that connects to external devices and has room for other parts which may be needed for custom applications; the Memory Expansion Board which may be added for more complex datalogging experiments; and an optional expansion board that will give the system future capabilities (LCD display, higher resolution A-to-D converters, servo controllers, etc.).

Power and Control Board

This separate circuit board holds the power supply circuit (with battery recharger), the manual switches and indicators needed for the user interaction, connections for the in-flight triggering mechanisms (at least one switch is needed to indicate first motion or clearing of the launch rod), and the connection to a personal computer (to "profile" the flight and to "retrieve" data). This board is not attached directly to the main assembly; this gives the user the option of mounting it in a more accessible location. Three manual switches are used for: RESET, START, and EVENT. Three low-power LED indicators show: Power On, Activity, and Battery Charging.

Battery and DC Adapter

The DC Adapter powers the system when it is not being used in flight. At the same time, the adapter supplies the power to recharge the battery.

A standard 7.2V NiCad battery (the size of a regular 9V battery) powers the UFC during remote use. Additional batteries may be needed for the output devices if they will be powering higher current loads (relay, igniter, etc.).

Input Sensors

Depending on the application, up to eight analog signals may be connected to the interface board and sampled by the UFC. The input devices may need to be mounted within the airframe in an area other than the payload section. Sensors are available to indicate physical parameters such as temperature, light, pressure and acceleration. Some sensors may need additional circuitry to adjust the voltage levels before the signal is sent to the main board; the custom project area may be used when this is necessary. Nominal input range will be 0-5 volts.

Other sensed inputs may be used to indicate simple on or off conditions. For this purpose, an additional four "digital" inputs are available.

Output Devices

Up to four outputs are available to drive output devices such as an up per stage ignition system or an ejection igniter. These output are timed by the flight profile which is set by the user before flight.

Personal Computer and User Software

Before the flight, the user needs to describe the flight "profile" using the Profiler software. Once entered, the sequence of events is kept in the UFC's memory even after the power is turned off. At the time of launch when the unit is powered on, it will wait for the <START> trigger and begin the pre-programmed flight profile.

After the flight, the data is held in memory until the user recovers the data using the Retrieval software. The main battery may be used to hold the data for periods of less than an hour. For longer times, battery-backed memory modules may be used but they are more expensive. These modules are especially useful for multiple flights, using a different memory module for each flight.

III. Hardware Design

The circuit design for the Universal Flight Controller is centered around a microcontroller integrated circuit from Motorola called the 68HC11. This "chip" is essentially an 8-bit microprocessor with several interface circuits built in. It was chosen because of its low cost, low power, and the ease of design (many automobile engines and other commercial products use the 68HC11 for a controller).

The circuit is separated into four segments which represent the hardware components as described in the system design. This approach makes it possible to keep the cost low for simple application which may need just the Main board, but allow for easy expansion to a set of boards which fit inside a moderately small airframe. In the case of the Power and Control segment, the separation of this part of the circuitry makes it easier to access the manual functions (switches, LED indicators, battery connection, and external computer connection).

A detailed explanation of the circuits and individual components is beyond the scope of this report. However, the following briefly describes each board.

The main board contains the microcontroller which connects to the support chips via the data bus, address bus and various control signals. The EPROM contains the UFC system software which takes control when power is turned on. The Profile RAM holds the user's flight profiles and any other information needed to be kept while power is off (this RAM chip contains its own battery). The clock circuit generates the 8MHz timing base for the circuit using a quartz crystal; all timing features are based off this accurate source. The rate of 8MHz is the maximum speed specified for the 68HC11 and gives ample processing power for the features required. The three connectors are used as follows: the lower right connects to the memory expansion board; the lower left connects to the interface/custom project board; and the left side connects to the power and control board.

The Power and Control board supplies power via the battery or the DC adapter. The 5V DC regulator converts any voltage from 6.5V DC to 12V DC and supplies a constant known voltage to the remainder of the hardware. This allows the system to use many battery types and various DC adapters. This board has three push-button switches which give the user manual control of the system: the <START> button begins the controlling/logging sequence and has the same function as the <START> trigger signal; the <EVENT> switch simulates the <EVENT> trigger for testing purposes; and the <RESET> button returns the system to the same state it was in when it was turned on. Three LED indicators represent the following: Power is ON (green), Battery is charging (orange), and System Activity (red). The Activity LED is used to show various modes by varying its blinking rate. The connectors at the bottom of the circuit diagram are used to connect to the trigger signals and to the personal computer.

The memory board may be used for applications which require larger amounts of data storage due to higher sampling rates of several channels. The board will hold from 32K bytes to 512K bytes of static RAM chips. The expansion connector on the main board connects to the memory board; individual units could be plugged in and out for each flight.

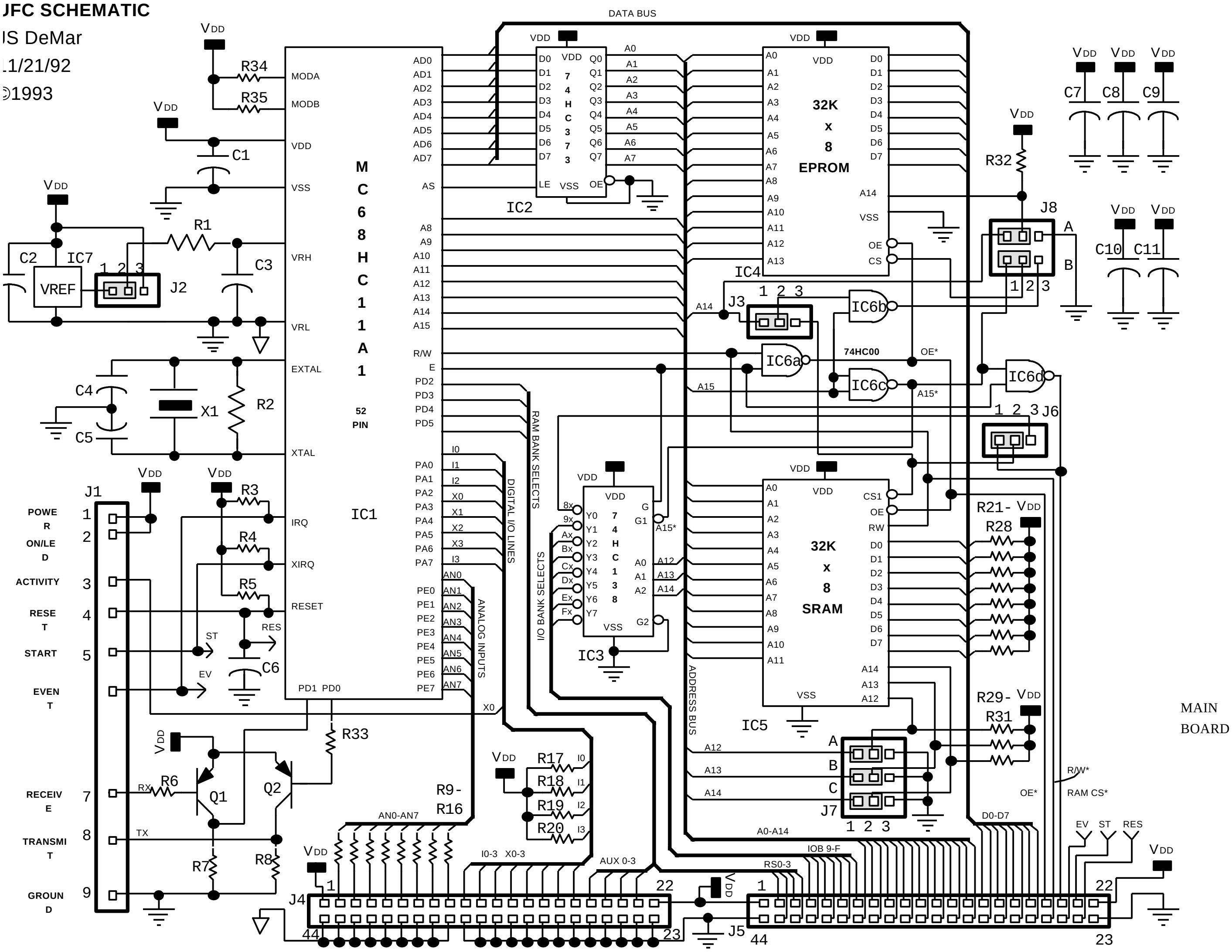
The total assembly is illustrated later in the section called System Integration.

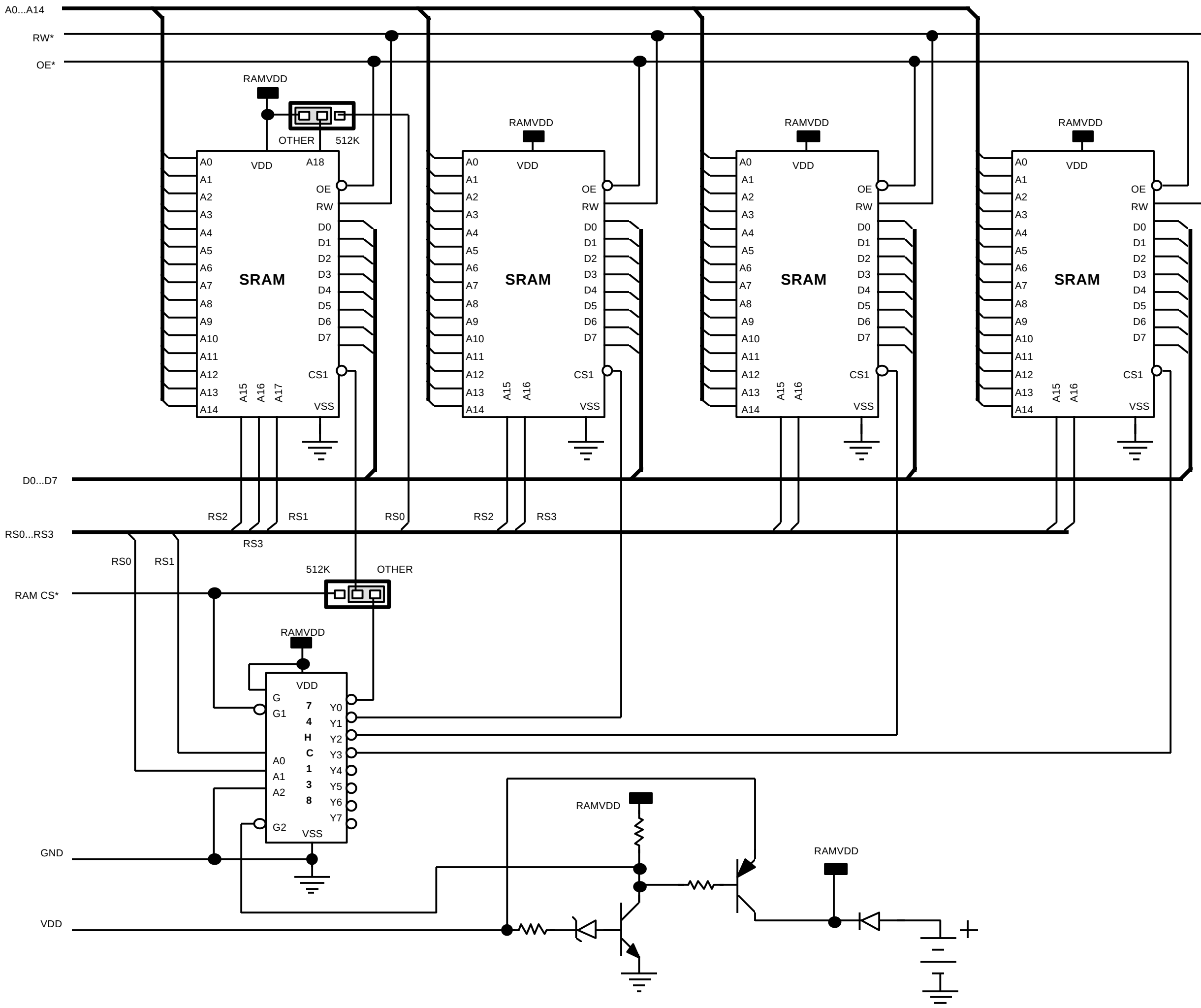
JFC SCHEMATIC

JS DeMar

11/21/92

©1993

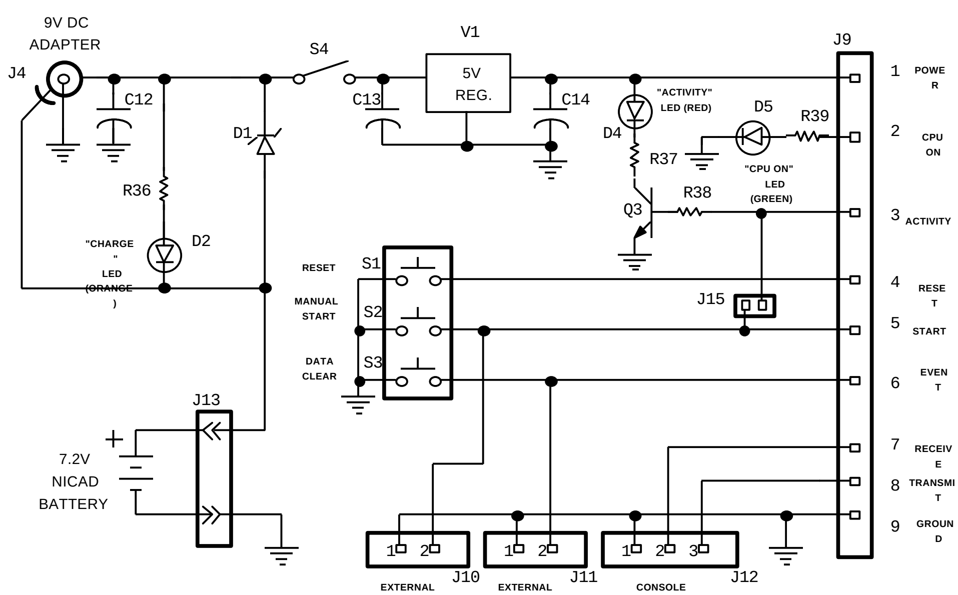




UFC Memory Expansion

JS DeMar
2/10/93

UFC SCHEMATIC
Power & Control Segment
JS DeMar 2/3/93



IV. Software Design

A significant portion of this project was the development of the system software. The programming was done in the "assembly language" or primitive instruction of the 68HC11 microcontroller, and required intricate coding of each function. A printed copy of the UFC system software is including in Appendix A of this report.

The system software operates in a multitasking scheme and is organized into four functional groups: Initialization (sets the system to a known state and waits for the <START> signal); Monitor (communicates with the user's personal computer through the serial port); Timer Task (runs every 1/100th of a second to control timed events and log digital inputs); and Data Sampler (runs every 1/1000th of a second to sample and store analog sensor data). A meaningful description of the program would require the reader to have sufficient knowledge of microprocessor programming and realtime software design. In any case, the operation of the software from the point of view of the user will be more meaningful to the purpose of this report.

The other half of the software design is the user software which runs on a personal computer. To make this part of the system easy to operate, the program is designed using a "graphical user interface" called GEM. This product is similar in functionality to the Apple Macintosh system and Microsoft Windows, but requires very little memory and will run on inexpensive portable PC's without a hard drive. Future plans include versions for Microsoft Windows and Apple Macintosh.

The Profiler/Retrieval software was written in the 'C' language using "Microsoft C" and development tools for the GEM system. A programming tool called the "Resource Construction Set" is an interactive graphics program used to design the menus and screens for the user software. The majority of the program involves interacting with the menus and screens as they are presented to the user. Another major portion of the software communicates with the UFC system via the serial port; a set of commands translates the instruction from the user into directives and data for the UFC. Other functions of the software include storing data to disk, viewing data and printing data. A printed copy of the Profiler/Retrieval source code is included in Appendix B.

The prototype developed for this report has some additional software to help in development and "debugging". The UFC monitor has a menu system which bypasses the need for the user program. However, it requires more knowledge of the inner working of the controller than does the Profiler software. This menu mode communicates with any personal computer via the serial port and a terminal emulator program (such as ProComm on a PC). Also, a 68HC11 debugger is stored in the system EPROM to aid in system development; this program was free from Motorola but required some modification to work with this design.

V. System Integration

The system is designed to assemble and fit within at least a 2.5-inch diameter model rocket payload section. Further redesign may allow down to a 1.6 inch diameter tube. The following drawings show how the three boards are intended to connect together in the final design. However, the first prototype for this project is slightly larger because most of the components are on one board. As of the date of this report, custom printed circuit boards are being built to prototype the system in the actual three-board configuration.

All of the main hardware components connect together using 0.1-inch (pin spacing) headers and connectors. This system is low-cost and easy to work with for the beginner. The serial connection was designed to use a small RJ-11 connector (phone jack) which requires an adapter to connect to the typical personal computer (this type was chosen to reduce size and weight).

At least one external trigger switch is needed for most applications to indicate a <START> signal to the UFC. Typically, a small "rocker switch" would be mounted so that the launch rod would close the switch until the rocket cleared the launch rod. Other methods may be used, such as a relay signaled from the launch controller, or a sensor which detects the first movement of the rocket. Other switches may be used to signal events such as successful staging or ejection. The <EVENT> trigger is meant for these purposes and may be sensed within the flight profile to control other actions.

VI. User's Guide

This section introduces the operation of the UFC by stepping through a simple example application. The intent of the report is to show feasibility of an easy-to-use system and, through the course of this work, I have defined many areas that could be further improved and simplified for the final product.

The following example will "profile" the controller to sample one Sensor Input at a rate of 100 samples per second. After 10 seconds, the controller will go into low power mode (to stop sampling and save on battery power). Then, the force transducer will be used to simulate an in-flight accelerometer experiment. Lastly, the results will be retrieved and viewed.

1) Required Equipment

- MSDOS PC Compatible with a serial mouse, VGA graphics, an extra serial port and a 9-pin modem cable.
- Terminal emulation software (such as ProComm).
- The UFC software disk.
- The Universal Flight Controller Prototype and DC adapter.
- A Motorola pressure transducer (included with the prototype for this report).

2) Hardware Connection

NOTE: The prototype is not as rugged as a production unit would be, so please be careful not to damage the wiring while handling it.

Connect the 9-pin serial cable from the COM2 port of the PC to the serial port on the UFC. The mouse should be connected to COM1.

Connect the DC Adapter to the power connector on the UFC then plug the adapter into an AC outlet. At this point, the Green power light should be ON and the Red activity light should be OFF.

3) Profile Session

Turn on the PC, change to drive A, and type PROFILER (then hit Enter). Wait for the main Profiler screen to appear.

Next, hold down the UFC's <START> button, press F1 on your PC's keyboard, then release the <START> button. This will put the UFC into Monitor mode and the Red activity light will be on continuously (non-blinking).

The Profiler will read the current profile from the UFC and display the information on your PC's screen. You are now ready to change the profile for this example.

In the Profiler program, the mouse may be used to point and click on the line you wish to change. Or, you can use the <TAB> key to jump to the next line and <Shift-Tab> to move to the previous line. Arrow keys and the delete key work as they would in normal typing. The main functions at the bottom of the screen work with a mouse click or by pressing the "F" number key associated with it.

Change the lines in the box labeled "A NALOG INPUTS" as follows: for the first line (AIN-1) enter the number 10 for a sample period of 10msec (100 samples per second); enter 0 for the remaining 7 lines (this turns off the sampling for these inputs).

Enter the following on the first line in the box labeled "TIMERS": for Type enter SS (stop system) and for Time enter 1000 (10 seconds, counts in increments of 1/100th of a second).

Now, press F2 to Store the profile in the UFC's profile memory. You may exit the Profiler program at this point.

4) Simulating a Flight

With the new profile stored, press <RESET> on the UFC to make it ready for simulated flight. Press the <START> button on the UFC and immediately apply a negative pressure (i.e. with your mouth) to the pressure transducer port. This will simulate the reduced pressure at altitude and give the UFC something meaningful to record. During the 10 seconds after pressing <START> the Activity LED (Red) will blink at a rate of once per second. At 10 seconds it will stop and the UFC will place itself into low power mode.

The above paragraph may be repeated to take multiple flight samples. The data will be stored as separate flights until memory is full. The Activity LED will blink at a faster rate when there is no room left to store samples. In this example, a total of 1000 samples is stored (100 per second for 10 seconds) and requires about 2K of memory (2 bytes is required per sample). About 16 "flights" can be stored in the main 32K RAM module.

You may keep the serial connector attached to the PC during the tests in order to watch the status sent out the port. Use a terminal emulator set at 9600 baud. This will also let you enter the monitor menus and view system parameters while the UFC is working. A demonstration of this mode will be part of the presentation for this report.

5) Retrieving the Data

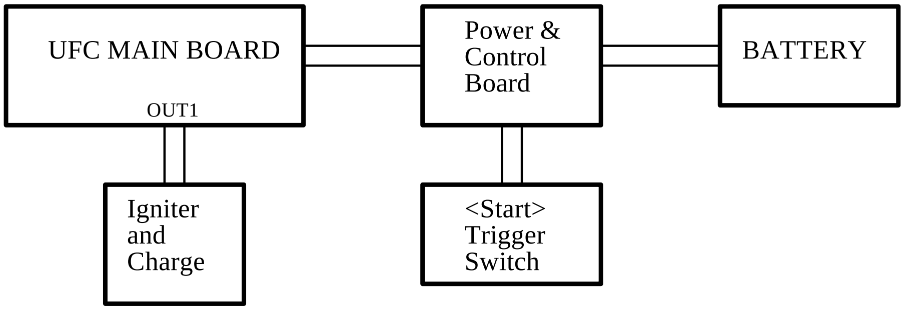
Run the program called "RETRIEVE" on the UFC program disk. When the main screen appears, press F1 to load in a list of datasets found in the UFC. Select a dataset and press F2 to load the data from the UFC to your PC. The data will appear as a simple graph versus time. Press F3 to store the data to disk.

[More time needs to be spent on the user software portion of the project. This example was done to show the "look and feel" of the final software.]

VII. Application Ideas

Example 1

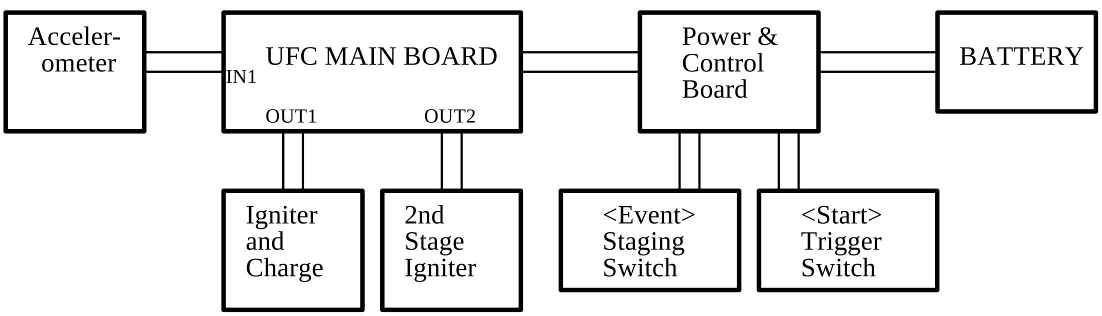
Simple Ejection Timer



Profile: One output timer set for desired elapsed time from launch to ejection.

Example 2

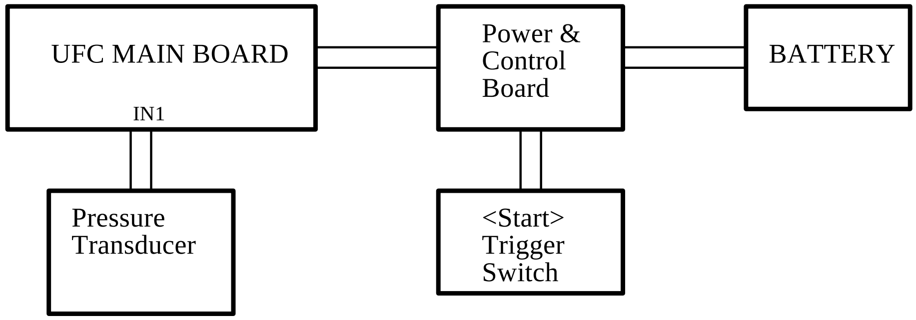
Staging Timer with Feedback



Profile: An output timer set for desired elapsed time from launch to 2nd stage ignition.
An event sample timer to ignite upper stage ejection if staging event does not take place.
An analog input set to 1000 samples per second to record staging performance.

Example 3

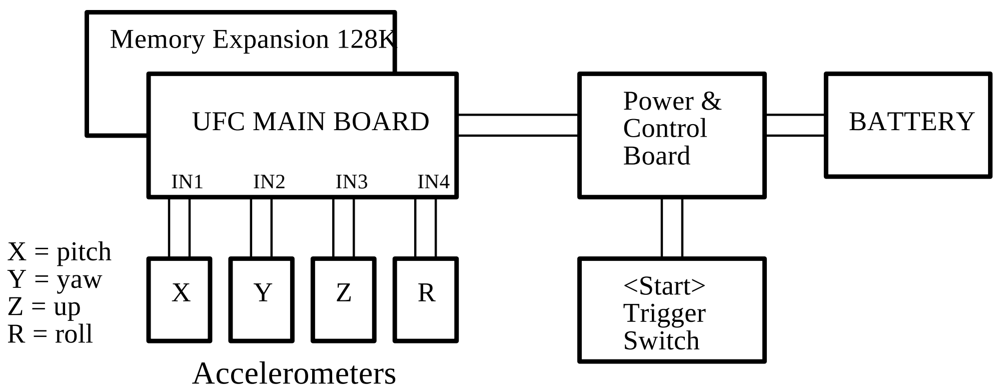
Altitude Profile Datalogger



Profile:
An analog input set to 1000 samples per second to record absolute atmospheric pressure every 1/1000th of a second. Data is retrieved and scaled to altitude versus time.

Example 4

Multi-axis Acceleration Datalogger



Profile:
Four analog inputs set to 1000 samples per second to record four axis of acceleration.

Appendix A: UFC Assembly Source Listing

(no longer available)

Appendix B: Profiler/Retriever Source Listing

(no longer available)