

# **USING THE STD BUS FOR ENCODER/DECODER SYSTEMS ABOARD A HIGH-ALTITUDE BALLOON PLATFORM**

**Jim Willis**

**Program Manager**

**Stabilized High Altitude Research Platform Physical Science  
Laboratory New Mexico State University Las Cruces, NM 88003**

## **ABSTRACT**

The Physical Science Laboratory (PSL) is developing a balloon-launched platform that will support a wide range of high-altitude projects. The Stabilized High Altitude Research Platform (SHARP) will house several on-board microprocessor-based subsystems to provide telemetry, command, and navigation data. Support for a wide variety of scientific experiments requires versatile electronic subsystems aboard the platform. Using the STD bus as a building block component, the Laboratory has designed and fabricated such subsystems. The STD bus, a standard interface throughout the computer industry, was chosen for the design because of its availability, ruggedness, versatility, and wide selection of compatible, off-the-shelf components.

## **INTRODUCTION**

The Physical Science Laboratory at Las Cruces, NM, was contracted by the U.S. Air Force Geophysics Laboratory to develop a balloon-launched platform capable of supporting experiments that require three-axis, stabilized pointing. The Laboratory was asked to develop this platform to support normal balloon control functions, housekeeping tasks, and experiment requirements. The support functions included telemetry encoding, command encoding/decoding, platform navigation, and pointing. All of these support roles would be mission-dependent and, therefore, would require maximum flexibility in configuration and intelligence .

Because many of the experiments to be supported were under development, the support configurations of the platform were undefined. It was obvious by the requirements that the different tasks would require programmability. Off-the-shelf solutions would be required because the command system would be used on other platforms. The system chosen would have to be rugged, compact, and able to withstand long-duration flights at altitudes up to 125,000 feet. There was a strong desire that the various subsystems use the same type of

modules as much as possible to ensure interchangeability and maintainability of the various components.

With these requirements in mind, PSL decided to use a standard, off-the-shelf system that would allow the platform to be tailored to the requirements defined for each payload configuration. PSL chose the STD bus component line because of its ability to meet the wide range of requirements at a cost-effective level .

## **DECIDING ON SUBSYSTEM DESIGNS**

At the beginning of the project, there were many difficulties with the definitions of the various subsystems on the SHARP design. An example was the definition of the telemetry encoder. The usual questions arose when the use of the encoder was discussed: how many channels, what type of signal (analog, digital , etc. ), sampling rates of the channels, maximum size, and power consumption. These are classic questions that must be answered when any airborne telemetry system is being designed. The mix of analog and digital parameters is difficult to identify when a program is getting started. Even quantities of the measurements to be encoded is difficult to scope. However, one thing is certain: the number of measurements that are required to be encoded will grow to the defined limits of the encoder. In a development project, this rule is usually honored.

Another subsystem PSL had to design was the command and control subsystem. This subsystem had two distinct elements, the ground and airborne packages, that would have to establish a one-way communications link and control the various functions aboard the payload. The airborne unit would have to command payload functions in serial, parallel, digital, and relay language. The airborne unit also had to check all of its received commands for errors so that an erroneous command could not be executed. The rate of the commands and how they would be executed were also undefined at the start of the development.

A market search turned up several “off-the-shelf” solutions that would suffice for the telemetry encoder design; however, this was not the case for the command systems. The types of commands were basically defined but the details of the mix were not. As the overall program progressed, the format of the airborne command system changed drastically. With different experiments there would be different command types required. There were many obvious solutions to the command system problem. The initial consensus of the design team was to build special hardware that suited the defined requirement. PSL had used this technique successfully in the past, but the devices, like most special-design hardware, were always limited and difficult, if not impossible, to reconfigure. The final consensus of the design team was that the project needed a solution that would be versatile and still accommodate the foreseen expansion.

The third and most complex of the subsystems on the payload was the navigation and stabilization subsystem. The main feature of the SHARP payload is a three-axis, stabilized platform with the ability to “stare” at a target on the ground for prolonged periods of time. The stabilized platform has the capability to compensate for the motion and drift of the payload and the balloon. To accomplish this task it was obvious that an onboard computer would have to calculate the platform corrections in realtime. In the overall design this was the first real problem that required an airborne intelligence. Not only was an on-board computer required, but several interfaces were required to link the position sensors to the CPU. These sensors included gyros, encoders, and distance measuring units (DMEs). The position of the platform was also required to be telemetered to the ground. The platform itself was to have the capability for control from the ground in a “joy stick” mode of operation. This meant linkage to the command subsystem.

Stabilization and navigation were not simple problems, as altitude above the target was also a major consideration. An off-the-shelf guidance computer was considered, but the costs were prohibitive and, after all, we were not going to go to Mars. The design team decided to use an adapted version of an aircraft positioning system and conventional aircraft gyros. All of these units required interfacing to a computer; ergo, the design began to drift toward a bus-structured microprocessor system. As with any solution to a design problem, the answer seemed simple once it was decided. There were many bus-structured computer systems on the market with a seemingly infinite selection of interfaces and microprocessors. The question became which bus and what mix of components should be used.

## **SELECTING A BUS SYSTEM**

The market place has a wide variety of bus-structured systems including CAMAC, STD, S, and MULTIBUS. All of these systems have advantages and disadvantages in relation to each other. The problem for the design team was to determine the system best suited for our application. The design team set about developing guidelines for selection of the bus system to be used in the platform. These guidelines included:

- Small size of the card cage and circuit boards.
- Rugged construction of components.
- Ability to operate in harsh environments.
- Wide selection of interface modules.
- Choice of microprocessors and memory types.
- Industry acceptance.
- Many second sources for components.
- Future product availability.

With these guidelines in mind, the design team examined the specifications for each bus system and how the systems matched up.

The STD bus had the edge in size. Size, as well as weight, was important because of the size constraints of the payload structure.

For ruggedness and operation in harsh environments, the STD bus again seemed to have the edge. The STD bus components are designed to operate in the industrial environment and, although the industrial environment is not the same as the environment in the stratosphere, industrial environment tests did provide the temperature and handling specifications needed by the design team. Temperature is the main enemy of components on long-duration missions in the near-space regions of the atmosphere.

The choice of microprocessors available for the STD bus system was a parameter that left something to be desired. Off-the-shelf, STD microprocessors consisted entirely of 8-bit units and none of the new 16-bit “machines” that have considerably more computing power. The bus is structured basically to support an 8-bit processing environment and does not have the number of lines necessary to fully support the 16-bit microprocessors.

The choice of interfaces for the STD bus, however, seemed to be almost unlimited, with second sources available for almost every type of interface. Because of the project requirements, this wide selection of interfaces was a major plus in the favor of the STD bus. A decision made by the design team at the beginning of the project came into play during this stage of the selection process: minimize the use of custom-built components and move toward increased usage of “off-the-shelf” material. The selection of circuit modules for the STD bus was so good that it looked like many of the design problems of the three subsystems could be solved with off-the-shelf components.

Industry acceptance of the STD bus was wide; also, excellent documentation was available on the components and their applications. With this widespread use throughout the industry, the future availability of STD bus items seemed assured. This observation was supported by the announcements of new STD bus components almost each month in the trade journals.

The only negative factor in the STD bus examination was the available selection of microprocessors. All the other guidelines were met or exceeded by the STD bus. In short, the bus seemed to be made for our application.

The other bus systems that were evaluated met some of the criteria but, for one or more reasons, were eliminated during the selection process.

- MULTIBUS was eliminated because of its size and fragility.
- The S bus was disqualified because of its lack of component selections and the possibility of its obsolescence.
- CAMAC was simply too large for this application.

With all of these facts in favor of the STD bus, it was the natural choice of the design team.

## **CHOOSING THE APPROPRIATE MICROPROCESSOR**

The next decision point in the design process was to select a microprocessor module that would give the performance desired and have the software support necessary for the project. The key to the development of any microprocessor-based system is the firmware design and implementation of good, sound programming practices. The availability of applications software that can speed this process is invaluable to the success of a development project.

The unit chosen by the design team was the 4 MHz Z80 CPU with the supporting CPM operating system. Both of these elements are practically industry standards; the CPM operating system has extensive software packages available off-the-shelf. The only real programming that the development team wanted to do was the individual drivers and the English command translators. The CPM operating system also had several utility products written for it that allowed the use of standard file transfer routines and back-up procedures.

## **DETERMINING THE BUS MODULES**

After the bus system and microprocessor were chosen, the various subsystem designers went about the task of choosing the bus modules that met their requirements. It was at this point that the decision to use the STD bus system began to look like a wise decision.

### **Telemetry Subsystem Implementation (Ref. Figure 1)**

The designer for the telemetry encoder subsystem found that all the required analog interfaces were off-the-shelf items. The same was true for the parallel digital units and the serial interfaces that were used to communicate with the other subsystems. The encoder for telemetry looked to be almost entirely “off-the-shelf.” The generation of IRIG Bi-phase PCM required only a serial SDLC card with a slight modification and the protocol turned off. Because the interface cards were removable from the bus, a card mix could be selected that would support the various mission requirements expected in the future.

The use of the microprocessor to control the sampling scheme gave the system the versatility required in an evolving project. A baseline sampling scheme was developed to handle the general case. All sampling tables were stored in PROM to control the CPU; to change the sampling scheme, one merely changed the program in PROM. Nothing was “fixed”; sampling rates, types of measurements, bit rates, interfaces selected, and sampling schemes were not “locked in.” With upper frequency limitations, the system could perform the housekeeping mission and be adapted to monitor balloon control functions.

### **Command Subsystem Implementation** (Ref. Figure 2)

The designer of the command subsystem found similar success in using the available STD bus modules in his application. The availability of relay cards was excellent with the same choice in digital, parallel, and serial units. The “programmability” of the command subsystem made it very versatile. Reaction to a received command can be programmed into the command subsystem to provide a sequence of commands based on one input. Other PROM-programmable features include: timed commands for execution for fixed amount of time after reception; fail safe commands that do such things as restart the computer and initialize the system if not reset by a ground command within a certain time; and legal commands that the system will respond to. The execution of each command is a programmable function that also resides in PROM.

For the command subsystem, the best “fall out” from the STD bus design was the error checking capability that could be purchased. The error detection scheme chosen for the command system was the cyclic redundancy character (CRC) test. CRC is supported by several off-the-shelf units for the STD bus and is used in many computer applications throughout the industry. This type of error detection gives the command subsystem excellent protection from executing erroneous commands.

As with the other subsystems, the choice of which interface to use for the command subsystem can be defined before each mission and is not a fixed parameter. The subsystem is very versatile and almost all components are available commercially.

### **Navigation Subsystem Implementation** (Ref. Figure 3)

In the navigation subsystem, there was a mixture of PSL-developed interfaces and off-the-shelf units. This subsystem used the same computer module as the encoder and command subsystems. The navigation computer, however, would be doing much more computational work than the command and telemetry computers. To speed up the arithmetic performance of the Z80 module, a floating point unit was added. The floating point unit was another off-the-shelf module available for the STD bus. The interfaces to the gyros and the distance measuring units had to be custom-built to support the aircraft-type hardware. These

interfaces and some transmitter pre-modulation filters were the only custom-built cards that were necessary. Telemetry data from the navigation subsystem was sent to the encoder via a standard serial interface. The motor controllers were also available off-the-shelf.

The firmware programs for the three subsystems were developed under the same operating system, with many of the programs being interchangeable between subsystems.

## **TESTING THE SYSTEM**

Overall, the system (Figure 4) has performed very well. The hardware has been flown twice with excellent success. There also have been a number of high-vacuum chamber runs that have added to the reliability of the testing results. Actual balloon control functions (e.g., valving, ballast control, and flight termination) have been accomplished during a proof flight mission. During flight times, no anomalies have occurred with the STD units.

The dynamics of the system also were tested during the test flights. Each flight carried different instruments to be supported by the system. These different configurations caused changes in the telemetry sampling strategy and command formats. Last minute configuration changes were accommodated without too much agonizing.

PSL constructed a portable command unit so that a full size computer would not be necessary to operate the payload. This unit is very useful in readiness testing and remote support roles. Again, it was built entirely of STD bus components.

## **SUMMARY**

The STD bus exceeded the expectations of the design team in both performance and reliability. It is functional and versatile with new products appearing every day. In fact, the designers are looking at a new version of the Z80 module that carries more memory and runs at faster execution speeds. The overall system meets the design goals of being reflyable and reconfigurable, mission after mission, now and in the future.

The development of the Stabilized High Altitude Research Platform is sponsored by the Air Force Geophysics Laboratory, Air Force Systems Command.

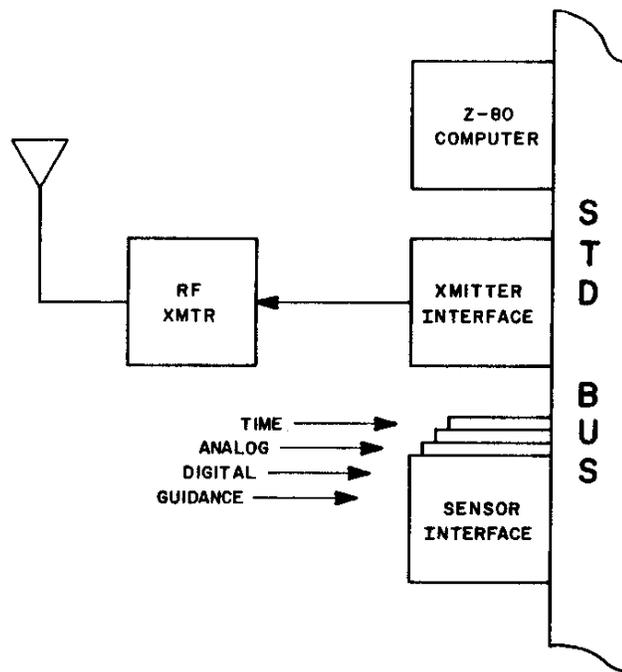


Figure 1 Telemetry Encoder Subsystem

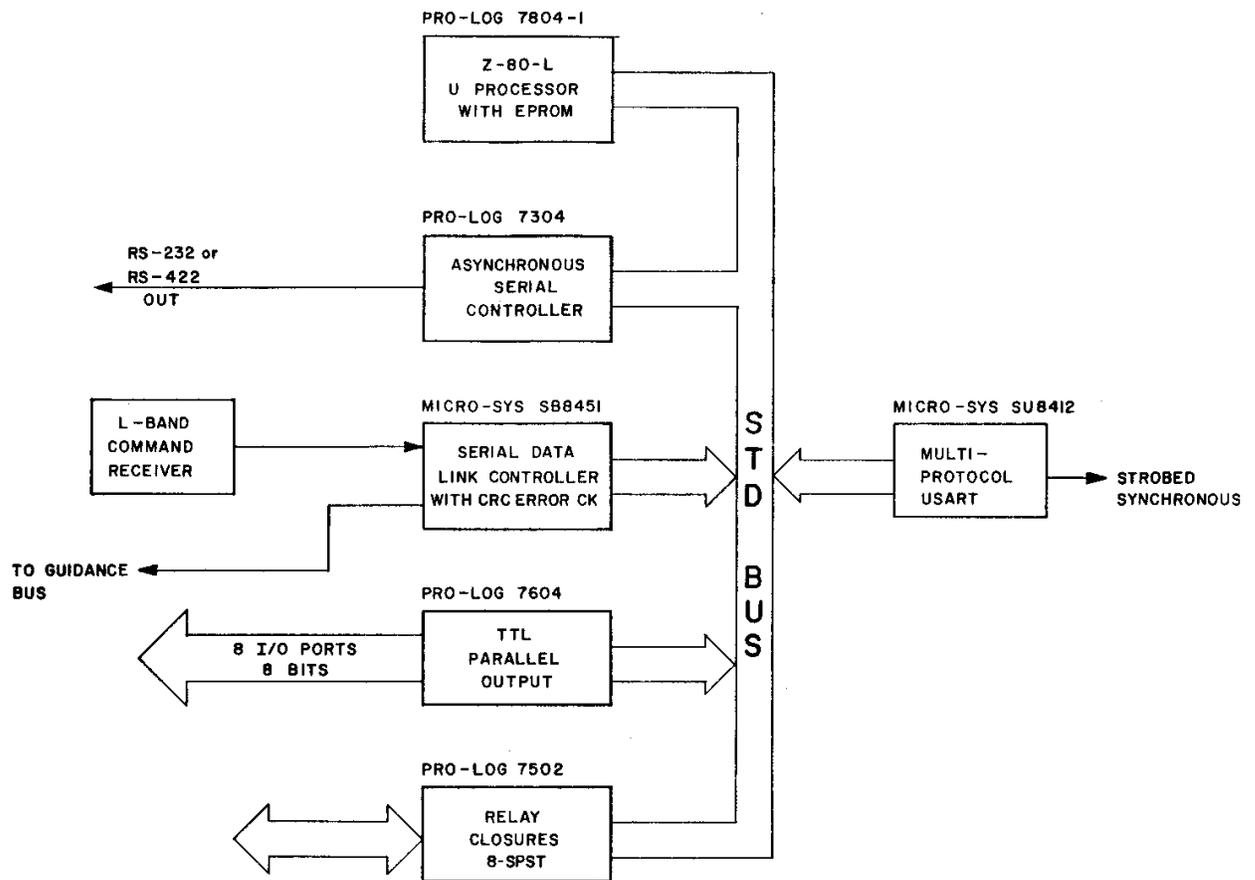


Figure 2 Command Subsystem

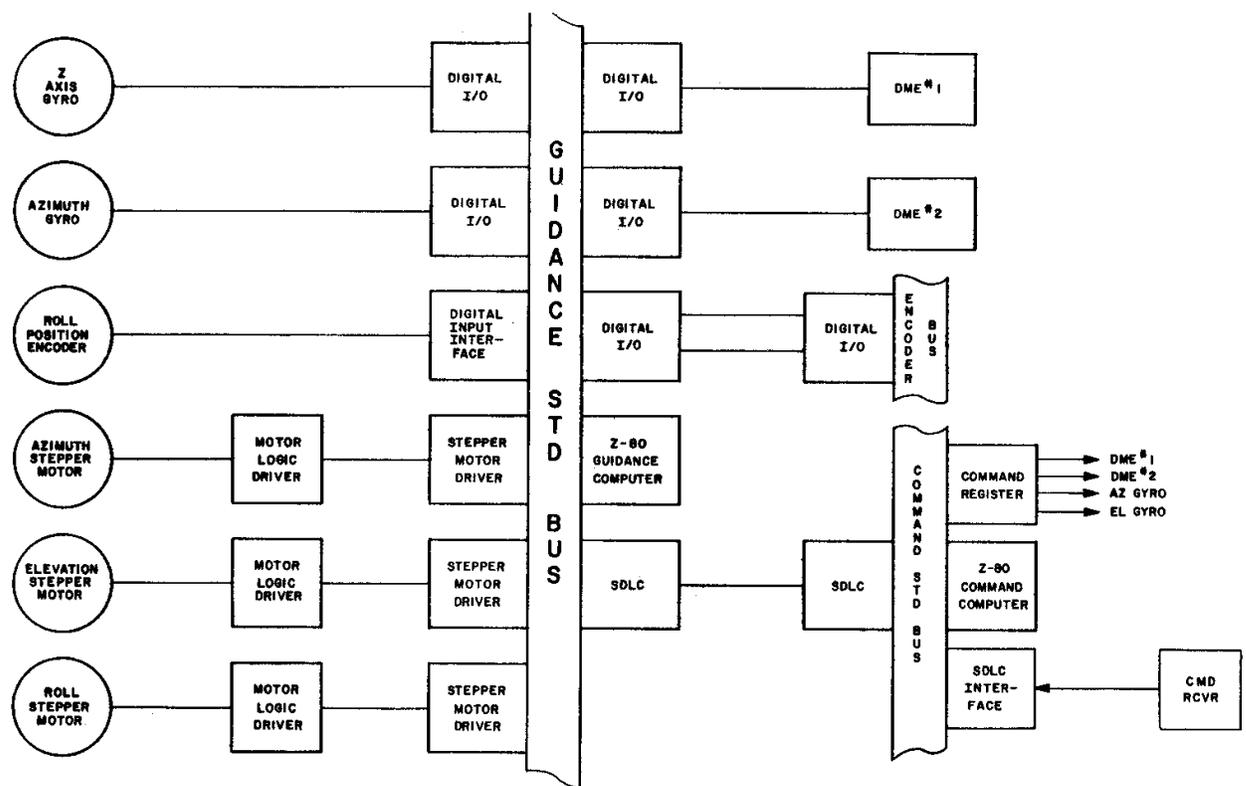


Figure 3 Navigation Subsystem

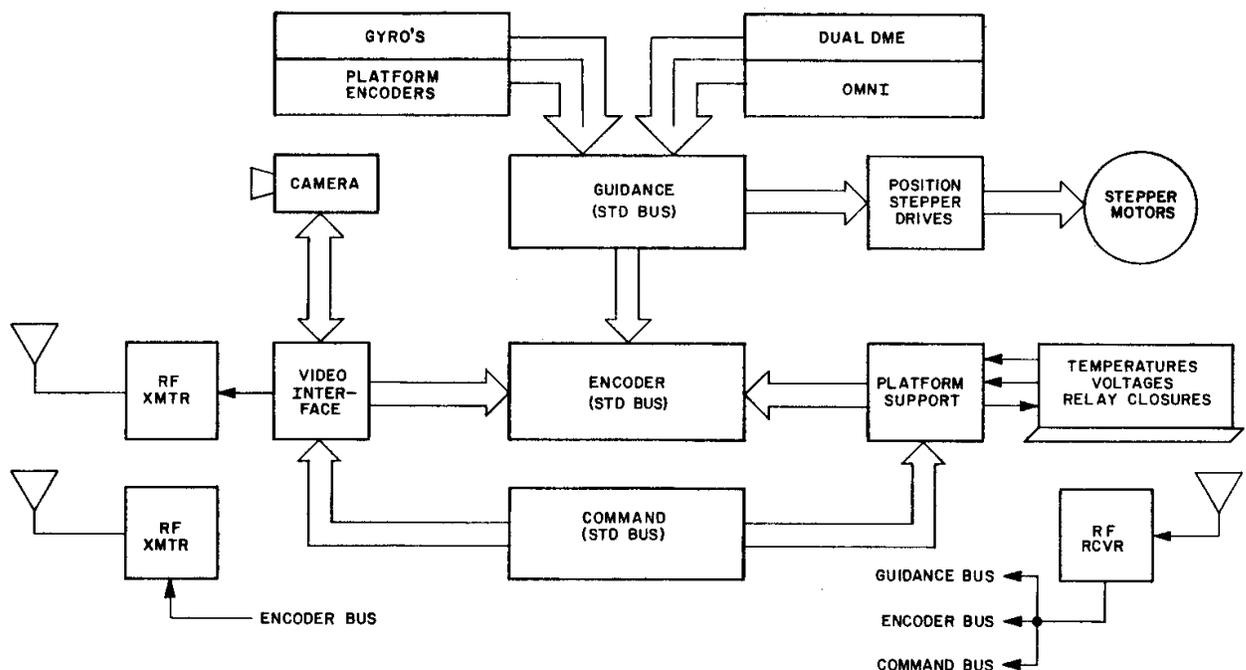


Figure 4 Airborne Electronic System