Developing a Stable Architecture to Interface Aircraft to Commercial PCs

Dan W. Christenson and Lynn Silver
Ogden Air Logistics Center

This article introduces a new concept of utilizing commercial personal computer products to interface with military and commercial aircraft regardless of their product life-cycle mismatch. Existing products are described, architecture for each implementation is derived, and their strengths and weaknesses are explored. An attempt to define the root causes for the problem of implementing interface architectures in this environment is presented. In addition, a new developmental architecture is introduced that is designed to maintain the strengths of the traditional architectures and eliminate some of the weaknesses and inefficiencies. A series of hardware/software co-development projects are described to demonstrate this new architecture. The relative performance of the architecture has been evaluated and refined by multiple implementations. These are described and future implementations examined.

In response to a growing need within the U.S. Air Force (USAF) to lower operating costs, a significant amount of research has been initiated to find logical, supportable opportunities to utilize commercial products and to replace products that have historically been custom-designed. Multiple development efforts have helped to refine concepts that have proven their utility at lowering acquisition costs while other efforts have proven to have lower sustainment costs. Recent efforts have shown that by managing the architecture of the developed equipment, it is possible to lower the overall life-cycle cost, while providing long-term, technically viable, and user-friendly equipment.

The current USAF loader/verifier of choice eliminates the obsolescence of computer hardware and software used on traditional loader/verifiers by being nonproprietary in its design. This allows the USAF to upgrade to newer personal computer (PC) platforms without a lot of expense.

Background

For the past decade, the Department of Defense (DoD) has faced budget cuts that have translated into lost personnel, slashed weapon systems development budgets, curtailed maintenance budgets, and extended weapon systems lifetimes. In this era of doing more with less, one of the easiest implemented directives was to use, to the greatest extent possible, commercial off-the-shelf (COTS) products.

However, since the DoD does not command a significant part of the electronics market share, it has little ability to affect the direction of the overall marketplace and thus COTS products. This has been further underscored by the cancellation of most military standards, because, among other reasons, the standards themselves could not be updated quickly enough to allow military product developers the opportunity to use current technology before it became obsolete.

Some of the first uses of COTS included applying commercial PCs to aircraft back-shops and flight lines. Several generations of COTS PC products have been in use by the USAF.

> *"It has only been since the late 1970s that any significant communication standards have existed for aircraft. If a designer is to interface with many different aircraft interface types, flexible interface techniques must be developed."*

Each of the PC products introduced into the DoD has faced a similar set of initial requirements that could not be updated quickly enough to utilize current technology. Each has taken a similar path to implement the requirements, and each has had similar problems at the end of the short product lifetime. These PCs are used for a variety of uses, including user interfaces for embedded computers, memory loader/verifiers for embedded computers, test equipment, test equipment controllers, technical order delivery systems, and digital communications equipment.

A recent picture that ran in many aerospace and local publications highlights the problem that the USAF is facing. The picture was a family photo of a B-52 commander, his father, and grandfather – all three had served on the same aircraft. As weapon systems lifetimes are extended, the opportunities to update the systems becomes more challenging. While every electronic system in the B-52 may have been updated, remnants of the original infrastructure remain. Much like today's railroads that have track separation distances based on the wheelbase of Roman chariots, the avionics systems of the USAF have interface specifications that have outlived their authors.

When the F-4 left the USAF inventory in the 1990s, it retained interface descriptions that reflected its Resistor Transistor Logic (RTL) roots from the 1950s. The F-16 discrete signal interface descriptions that were written in the late 1970s have specifications most easily implemented by switches and relays.

Only in the last 10 years has the AIM-9 missile provided an interface that did not reflect its original servo-type interface developed in the 1950s. For many years, both the aircraft and the weapon implemented the original archaic analog interface, using a mixture of analog and digital circuitry only because it was the easiest way to make sure the weapon and weapon systems would be interoperable. Ultimately, just as in the case of the AIM-9 missile interface, the designer of today's computer-to-aircraft interface is forced to be compliant with the existing weapon systems interfaces.

Traditionally with each new embedded computer, a new method to communicate with and to control it was introduced. This is extremely unfortunate for today's interface design engineer. When you examine the historical rationale, there were at least four significant reasons to create a unique protocol:

1. Aircraft-unique throughput or communications needs forced unique solutions.
2. Standards that would meet the needs were not well known.
3. It was chosen because of management concerns.
4. It was chosen because of convenience.

In many cases this uniqueness extends to voltage levels, drive current, timing, and data protocol. The number of electrical interface *standards* now exceeds 100; the number of data protocol *standards* is several times that number. This means that in most cases, each interface that the designer approaches is different from the last. It has only been since the late 1970s that any significant communication standards have existed for aircraft. If a designer is to interface with many different aircraft interface types, flexible interface techniques must be developed.

As this set of issues was evaluated, it became apparent that addressing this problem was the central issue for the architecture. Because aircraft and commercial PC life cycles are so far out of synchronization, and because that gap is growing, providing a long-term, supportable method to *buffer* the two environments is the essential artifact of this architecture. A graphical representation of the concept is shown in Figure 1.

With each new interface type, the adapter between the aircraft and the PC had to be addressed as part of the design; the logical place to build a standard was at that point, named the Aircraft Adapter Group (AAG). The name was chosen because the predecessors of this architecture used AAGs to interface their standards loader/verifier to the weapon systems. Therefore, this historical name was chosen because it is somewhat analogous to the function.

Examples of PC-Based Support Equipment

Three USAF examples of aircraft support equipment based on PC technology include the Automatic Test Systems/Product Group Manager's Digital Computer System (ATS/PGM's DCS), the F-16 Enhanced Diagnostics Aid (EDNA), and the F-15 Programmable Loader/Verifier – NT version (PLV-NT). Each of these PC-based systems was introduced into the USAF inventory in the last 15 years.

Each was acquired with similar standards that have traditionally been levied on all support equipment. Virtually none of the PC products utilized for aircraft support equipment has been used without modification. The necessity to modify the interface was driven by additional requirements associated with the unique environment of use. These environmental requirements fall into three basic categories:

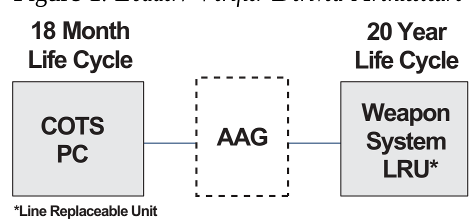
1. Security environment to avoid compromise of classified data when the equipment is operated in the close proximity of those who had no need-to-know.
2. A physical environment requiring that all input/output (I/O) is installed inside the PC.
3. The PC was required to be environmentally compatible with a USAF flight line.

Forcing the PC to comply with these requirements has at least three negative effects:

1. They drive up the acquisition cost because the COTS PC selected is virtually a custom product.
2. They drive up the re-host cost because the new PC has to be re-procured from the original source because of proprietary data issues.
3. Compliance drives down the overall performance because the custom computer market lags behind standard PCs by as much as two years.

As these pieces of equipment were introduced into the inventory, they were initially well received, but quickly were considered archaic when compared to traditional PC equipment. Their lag-behind technology, the inability to use current hardware and software products, and the cost to acquire and maintain the equipment made them unpopular. Within a

Figure 1: *Loader/Verifier Derived Architecture*



Definitions

- **Ethernet —** A high-speed serial data bus generally used to implement Local Area Networks. Ethernet was not designed to power peripherals; it is therefore required that a separate power cable/supply be used.
- **Firewire —** A high-speed serial data bus generally used for video/audio processing peripherals. Firewire was designed to provide a limited amount of power to peripherals. Firewire has the liability that it is not as widely accepted in the marketplace as a Universal Serial Bus (USB) is and that with a few exceptions (like Sony), it is not generally implemented in laptops.
- **IEEE 488 —** An eight-bit parallel bus common on test equipment. The IEEE 488 standard was proposed by Hewlett-Packard in the late 1970s and has undergone a couple of revisions. It allows up to 15 intelligent devices to share a single bus with a maximum data rate of one megabit per second.
- **MIL-STD-1553 —** A military standard that is slower than most modern serial busses and does not provide power to any peripherals. Because of the complexity of the protocol, expensive integrated circuits are required to implement the interface. Its redundancy and noise immunity have made it a popular interface for aircraft use.
- **Parallel Bus —** A bus consisting of multiple signal lines that simultaneously transfer data in a parallel method.
- **RS 232 —** A simple, universal low-power serial bus that can be found in many different applications from modems to PCs, where the length and quality of the cable depends on the data speed.
- **RS 422 —** A differential serial bus designed for greater distances and higher baud rates than the RS 232. Data rates of up to 100,000 bits/second and distances up to 4,000 feet can be accommodated with the RS 422.
- **SCSI —** A Small Computer System Interface designed originally to communicate between a computer and disk drives has been used when high-speed communication is necessary. Because of the number of wires required, SCSI cables are generally bulky. SCSI was never designed to power peripherals; it is therefore required that a separate power cable/supply be used.
- **Serial Bus —** A bus consisting of a limited number of signal lines (usually one or two) that transfer data in a serial (one bit at a time) fashion.
- **Universal Serial Bus —** A USB is a high-speed serial bus universally available on all PC products. It provides a minimal amount of power, allowing implementation of simple peripherals that do not require additional power supplies.

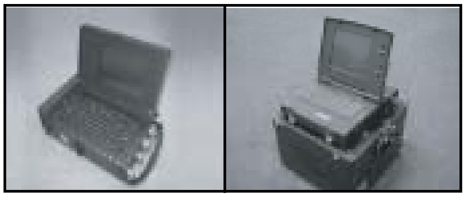


Figure 2: *F-16 EDNA and F-15 PLV-NT*

fraction of the traditional USAF product lifetime, each was considered obsolete and in need of update and/or replacement.

The USAF has not ignored this problem and as early as seven years ago, efforts were made to begin creation of support equipment standards. Efforts have also been made on the weapon systems acquisition system to drive standardization. At this point, the electrical interface *standards* fall into the following standards: IEEE 488, RS 232, RS 422, MIL-STD-1553, and *unique*. Depending on the weapon system type, about 60 percent of the interfaces are MIL-STD-1553, and 20 percent are RS 422, RS2 32, and IEEE 488. The remaining 20 percent are unique; many are simply the electrical interface between the microprocessor and its memory.

Techniques to address these interfaces have been developed. There are interface devices that implement all standard interfaces; with the development of the Field Programmable Gate Array (FPGA), interfacing to unique interface standards was greatly simplified. The unique interface timing as a minimum can be fully implemented. In the case of Transistor-Transistor Logic-based standards, the electrical portion of the interface can be addressed as well. These technologies were incorporated into F-16 EDNA and the F-15 PLV-NT, driving the acquisition costs to one-third of the traditional loader/verifier. These systems are illustrated in Figure 2.

This was considered a great feat until it was recognized that although the new devices were designed to last 10 years, the products became obsolete in less than three years. This made the products no cheaper than their predecessors did. The lesson learned is that not only must the acquisition cost be lower, but also the product acquired must be an *add on* so that neither internal installation nor modification to the PC is required. This allows the PC, weapon systems, and the AAG to be independently modified to accommodate updates.

"At the core of this development effort is the requirement to address a problem that plagues the entire DoD: commercial product lifetimes are becoming shorter while DoD product lifetimes are becoming longer."

Requirements Development

In 1997, a team of users (Next Generation Loader/Verifier users group), program managers, and engineers were convened to begin looking at the growing problem. The support equipment requirements that forced modification to the PC were addressed.

At the core of this development effort is the requirement to address a problem that plagues the entire DoD: commercial product lifetimes are becoming shorter while DoD product lifetimes are becoming longer. As the evaluation proceeded, the basic requirement for developing a stable architecture emerged: Develop a suite of standards-based tools and products that can be functionally implemented with many technologies.

The significant driving function, as mentioned earlier, is the rapid development cycle of the commercial PC. The PC must be allowed to change to utilize current technology. For this reason and for this application, the USAF has adopted a nontraditional approach, allowing functional configuration instead of traditional physical configuration. This means no effort is to be taken with the new equipment to physically configure the PC. Any PC that complies with the functional configuration document is acceptable for use.

The need to modify the PC to accommodate security requirements was replaced with tests and procedures that accomplished the intent of the modification. These tests must be accomplished on a relatively small sample lot and are the basis for the technical data that describe additional security protocols that are necessary to protect the classified information being processed.

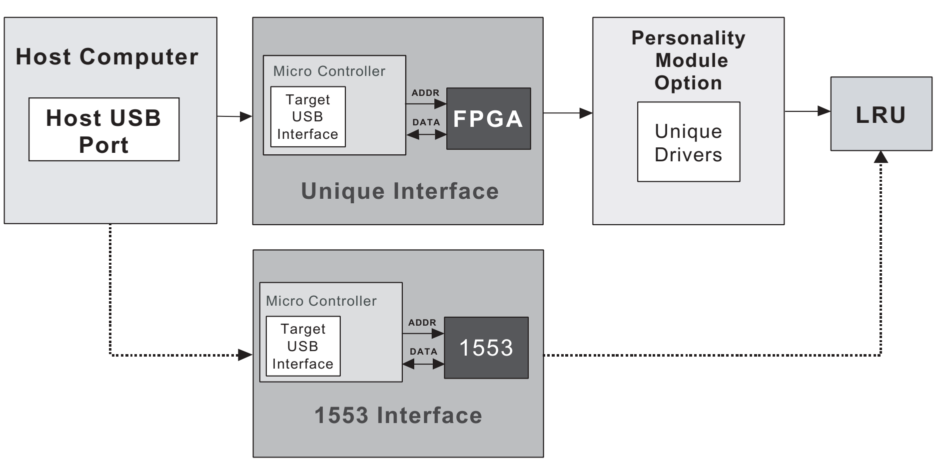
The need to have an integrated PC that addressed all the interface requirements was replaced by a lightweight PC with a small number of external interfaces that address each type of aircraft interface.

To accommodate MIL-STD-1553 remote terminal and monitor functions, without any significant local buffer in the AAG, any external interface must exceed one megabit/second (mbit/sec). Realistically the interface should be at least 10 mbit/sec to allow time for data processing and calculation of responses. This limited the commercial standards that were examined to accommodate the needs of Small Computer System Interface, Ethernet, firewire, MIL-STD-1553, and Universal Serial Bus (USB).

Since the external interface would have to be powered, an interface that could provide the needed power was desirable for two reasons. First, if the interface standard did not provide power, as in the case with MIL-STD-1553, then an external power supply would have to be used. Second, this power supply would also have to be ground-isolated to allow the system to interface with the ground reference of the aircraft. These requirements limited the selections to firewire and USB.

USB was chosen because it is part of the commercial PC standard and is back-

Figure 3: *Detailed Architecture System Diagram*



ward compatible (i.e., USB 1.0 devices will work with USB 2.0).

As implementations were examined, it became obvious that MIL-STD-1553 would be best accommodated in a separate interface, and because of the availability of commercial USB-IEEE 488, initial implementations were procured commercially. Figure 3 represents the block diagrams of the interface elements that currently implement the AAG architecture.

The strengths of prior developments include utilization of FPGAs to implement most of the weapon systems interfaces and PC-based user interfaces to lower the recurring costs. Those strengths are preserved with this architecture. Additionally, by limiting the host PC interface to one type, in this case USB, and by limiting the dependence to only two points, if the interface type becomes obsolete, the amount of re-host required to update to current technology is minimized. While USB is current in the PC environment, the host PC can be updated independently of the AAG hardware.

Possibly, the most exciting aspect of this is the blending of hardware techniques into the software arena. All of the *customization* of the hardware to accomplish the needed AAG functions is done with data that is stored on the PC. The FPGA data, the micro-controller firmware, and the 1553 configuration data are all stored in the PC as data and are effectively *executed* on the I/O elements. Any additional functions needed to implement the AAG are put in a Dynamic Link Library. A Computer Program Identification Number manages these data.

Prior Implementations

Variants of this architecture have been successfully utilized with minor variation to implement USAF subsystems in the following areas:

1. The Ogden Data Device (ODD). This device is used to interface with data transfer cartridges on F-16 and A-10 aircraft. The ODD has been utilized for mission planning purposes for more than seven years with only minor modifications and updates to the driver software.
2. The Personal Computer Memory Loader Verifier. The USAF used this device for F-4 reprogramming during Desert Shield. It has been in continuous use by allied countries for more than eight years, and has performed flawlessly.

Additional Benefits

Because the development tools can be hosted on the computer that is being utilized to host the interface, a simple, quick, and mobile development environment can be established. This allows the development environment to be taken to the integration environment during integration. This is extremely convenient when no local hot bench or integration facility is co-located with the AAG development environment. Many times the equipment to be interfaced is in remote, otherwise inaccessible locations.

The equipment is usable in multiple environments: flight line, back-shop, development, and integration facility. Because the equipment is based on prior implementations, the development costs can be lowered by reuse of development artifacts.

The ideas presented have been implemented in the Common Aircraft Portable Reprogramming Equipment (CAPRE). The CAPRE has been chosen by the USAF to be the next generation loader/verifier as shown in Figure 4.

Figure 4: *The CAPRE System*

Conclusion

The benefits of this implementation include the following:

1. Long-term supportability.
2. Simple re-host.
3. Supports shorter PC life cycle and longer weapon systems life cycle.
4. Supports hosting of the development tools on target platform.
5. Mature and stable technology.
6. Useful in multiple environments: development, back-shop, as well as flight line.

This program is being implemented under program direction of Warner Robins Air Logistics Center with technical implementation accomplished by Ogden Air Logistics Center/MASMD.◆

About the Authors

Dan W. Christenson is branch chief for the Weapon Systems Software Engineering Branch, Maintenance Directorate, Ogden Air Logistics, and supervises a 146-person organization that specializes in automatic test equipment development and maintenance. He began his career as an electronic design engineer and has continued to act as a systems design engineer for major projects, including avionics design for the F-16 and F-22. Christenson has a bachelor's degree in electrical engineering from Brigham Young University and a Master of Science in computer science from Utah State University.

Ogden Air Logistics Center
Software Engineering Division
(OO-ALC/MASM)
7278 4th St., Bldg. 100
Hill AFB, UT 84056-5205
Phone: (801) 777-9863
E-mail: dan.christenson@hill.af.mil

Lynn Silver is Section Chief for the Automatic Test Equipment and Weapons Systems Interface Engineering Section, Maintenance Directorate, Ogden Air Logistics where he directs a 50-person organization that specializes in automatic test equipment. He has previously worked as an electronic design engineer and was part of the Quality Engineering Support Team that established process quality assurance for the Software Engineering Division and later earned Capability Maturity Model Level 5 recognition. Silver is a former publisher of CrossTalk. He has bachelor's and master's degrees in electrical engineering from the University of Utah.

Ogden Air Logistics Center
Software Engineering Division
(OO-ALC/MASM)
7278 4th St., Bldg. 100
Hill AFB, UT 84056-5205
Phone: (801) 777-3823
E-mail: lynn.silver@hill.af.mil