2016 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
VEHICLE ELECTRONICS AND ARCHITECTURE (VEA) TECHNICAL SESSION
AUGUST 2-4, 2016 - NOVI, MICHIGAN

A COST-EFFECTIVE APPROACH TO ADAPTING CURRENT-FORCE EQUIPMENT TO VICTORY STANDARD IN-VEHICLE NETWORKS

Michael S. Moore, Ph.D.
Southwest Research Institute
San Antonio, TX

Kase J. Saylor, PMP
Southwest Research Institute
San Antonio, TX

Joshua Klein
Southwest Research Institute
San Antonio, TX

ABSTRACT

This paper describes a novel, cost-effective method of adapting existing, non-networked equipment to interoperate with the Vehicular Integration for Command, Control, Communications, and Computers (C4), Intelligence Surveillance and Reconnaissance (ISR) Electronic Warfare (EW) Interoperability (VICTORY) standards. It briefly introduces the VICTORY In-Vehicle Network (IVN) concept, explains why adaptation of existing equipment is necessary, and discusses the different patterns for adapting current-force equipment to VICTORY standards. It introduces a new approach we call the VICTORY Smart Cable, and describes its costs and benefits, including size, weight, and power (SWaP), recurring costs, and flexibility. The conclusion is that the VICTORY Smart Cable is a cost-effective transitional technology that can bridge the gap between several types of current-force, non-networked equipment, and future VICTORY-enabled devices, and that the adaptation does not require modification to existing devices, or to vehicle hardware or software.

INTRODUCTION

A major challenge in deploying open architectures such as VICTORY, Integrated Sensor Architecture (ISA), and Future Airborne Capability Environment, is that existing systems and components do not have compliant interfaces. A strategy must be created for leveraging current-force equipment: to either update this equipment or adapt it to the standard to integrate it with the new architecture. This is a general difficulty in open systems programs, and is perhaps the most common reason for "failure to launch": the new architecture will work, and the benefits are clear, but there is no tenable plan for leveraging all of the existing non-compliant equipment. Assuming a "big bang" approach, wiping the slate clean, assuming that all systems, sub-systems, and components will have compliant interfaces, is not a plan for success for deploying open standards when there has been significant previous investment and deployment. Success will be more likely if a strategy is developed up-front, during the architecture definition phase, based on analysis of the domain and the business model.

There are several possible strategies for integrating existing equipment with newly emerging open architectures. Clearly understanding the tradeoffs between these approaches, and defining a strategy for adaptation for how current-force equipment is to be integrated is important. Determining which strategy is appropriate for a particular environment requires understanding of the architectural approach, the drivers for how it was selected for the domain, and the nature of the current-force equipment.

BACKGROUND

Modularity and Interoperability

The overarching goals that drive the government to take open systems approaches in developing, integrating, and deploying complex systems and systems of systems are to reduce life-cycle costs, reduce the cycle times (time necessary to deploy a new technology or to reconfigure systems to meet mission needs), and to reduce the growth of SWaP in the electronics systems as capabilities are added. From an architectural perspective, each of these goals has at its core the need for modularity and interoperability between modules. Modules that interoperate with well-defined, open interfaces are less costly to integrate and are more likely to be able to be reused in different systems. Modularity requires interfaces to be defined clearly and for the behaviors related to the interfaces to be understood so that they can interoperate. Interoperation is based on the concepts of sharing: sharing data, sharing resources, and sharing services, and commonality: common data syntax and semantic, common mechanisms for configuration, control, and health management, and sometimes common physical/mechanical interfaces.

The concept of interoperating modules is relatively generic in that it applies to different aspects of interfaces depending upon the context of the system and what the modules are. Modules may have many types of interfaces, including:

- *Mechanical* (e.g. common interfaces for attaching modules of a space station)
- *Electrical* (e.g. connector and electrical interfaces for a common household power infrastructure)
- *Signaling* (e.g. common frequency, modulation scheme, and media access control method for sharing a radio frequency data link)
- *Networking* (e.g. common mechanism for addressing, prioritizing, and delivering messages between elements on a routed network)
- *Messaging* (e.g. common data semantic, syntax, encoding, encapsulation)
- *Interacting* (e.g. a protocol defines a common sequence of interface actions required for modules to interact)
- *Software* (e.g. common application program interfaces (API) for accessing underlying platform resources or to higher-layer software layers)

Examples of Modularity and Interoperability

The types of modules and interfaces that best serve the needs of a particular application domain depend strongly on the cost and schedule drivers of that domain.

For instance, in avionics and flight control systems, shared networking transport, processing resources, and physical backplane resources are likely assumed. The schedule and cost drivers are related to designing, developing, and integrating the flight control and human machine interface software. To reduce life-cycle costs, FACE defines a layered middleware architecture with software modules that interoperate with shared services and resources in the layers below and above via a combination of software API, messaging, and protocol style interfaces. This software-centric architectural approach is appropriate for the FACE environment, as the costs being addressed are related to software development and integration.

As a second example, when integrating electronic systems with military ground vehicles, the business model is different. There is not an existing environment that provides shared processing resources or network transport. Electronic systems have in the past been designed and procured mostly separately and have not been designed to interoperate or share resources. However, the need to reduce the growth of SWaP of the electronics systems and to support more advanced capabilities has driven the definition of a modular open electronics architecture. The VICTORY context is within the ground vehicle, so wired network links can be assumed. Because of the lack of a common, shared transport, VICTORY first concentrated on defining the signaling, networking, and messaging interfaces for a core network infrastructure. The In-Vehicle Network (IVN) establishes a managed, wired network transport with Quality of Service (QoS) and Information Assurance (IA) controls. The IVN also defines a shared processing units (SPUs) concept, which provides shared processing resources, and includes very light APIs to reduce the cost of porting applications between SPU platforms. VICTORY then defines a large set of loosely coupled messaging and protocol interfaces to shared data services, sensors of various types, vehicle and power systems, electronic warfare (EW) devices, and weapon systems. These interfaces support integration of many types of applications including: Situational Awareness (SA), vehicle management, navigation, communications, ISR, EW, intelligence, and logistics applications. The driver for the loose coupling approach (messaging instead of software APIs) is that in VICTORY, modules can be boxes, services, sensors, or subsystems. These messaging and networking interfaces could assume a relatively dependable, wired network infrastructure due to the internal ground vehicle context. In the case of VICTORY, a different business model resulted in a different approach.

For one additional data point, consider the problem space faced by ISA, which aims to integrate SA, ISR, and other sensing related applications across a diverse and geographically distributed context. The system of systems environment supported by ISA requires interoperation between sensors and applications located at Command Posts (CPs), Forward Operating Bases (FOBs), manned and unmanned aircraft, unmanned aerial sensor (AUS) platforms, unmanned ground sensor (UGS) platforms, and manned and unmanned ground vehicle platforms. ISA modules can be sensors, sensor platforms, complete systems, or software applications. With such a highly diverse and distributed context, it is necessary for the module interfaces to not only be loosely coupled, but also to be designed to continue operating or degrade gracefully as the quality of wireless network links and quality of service (QoS) varies. As with

VICTORY, ISA defines messaging and protocol module interfaces. However, ISA messages are more compact (bit efficient) to reduce use of wireless network resources, and the transport and protocols are designed to be more dynamic and resilient to changes in the wireless networks. This requires the development of custom methods of encapsulation, link management, service discovery, which in turn reduces the opportunity to use of standard, ubiquitous protocols and software packages. This may increase the cost of software development, but this is necessary due to the widely distributed and dynamic application space.

Each of these widely varying architectural approaches, with different types of modules and interfaces, is tailored to the business model and technical realities of its application space. That said, there are common threads between them that allow for a level of commonality and development of best practices. These include the need for a common set of terms and definitions, and a common formal description of the syntax and semantics of data (sometimes called a data model). Interoperability also requires a method of defining, executing, and documenting the tests that interfaces must pass to be compliant, and systems must pass to be conformant to specifications. It is also necessary to define a method for integrating equipment that does not yet (and may never) support the common interface standard. Research is being performed with the aim of creating languages and tools for data modeling, interface specification, and automated acceptance testing. Those will not be discussed here. The following will delve further into the later aspect: how best to adapt so-called current-force equipment to a modular open architecture, specifically the VICTORY architecture. The reasoning for the discussion of other related modular open architectures will become more clear in the conclusions.

ARCHITECTURAL PATTERNS FOR ADAPTERS

From an architectural standpoint, there are several alternative patterns for adapting current-force equipment for a modular open architecture such as VICTORY. Clearly the approach that is simplest architecturally is to modify the equipment to include a native VICTORY-compliant interface. However, this approach is naïve when considering current-force equipment, as it is not cost-effective, or in some cases even possible in the acquisition environment. Although future equipment can be procured with compliant interfaces, that approach can be dismissed when considering adaptation of current-force equipment. The two approaches that have been used most up to now are 1) a stand-alone hardware (HW) adapter and 2) a software (SW) adapter that runs on a shared processor. These are identified as HW and SW, although clearly in each case both hardware and software are involved.

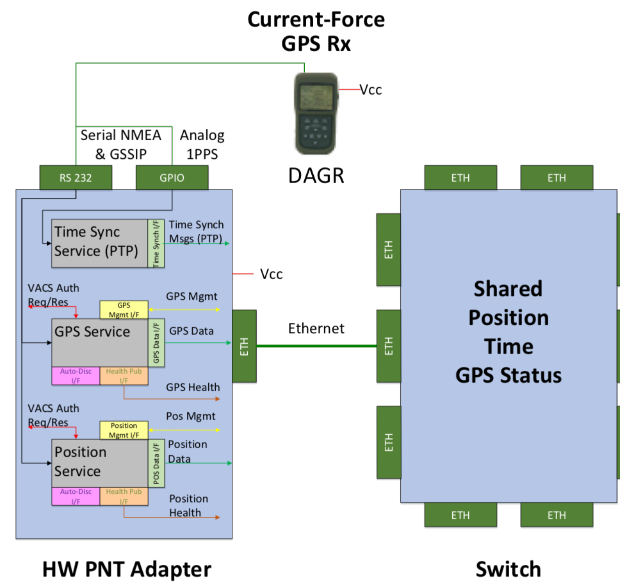

Figure 1. Example VICTORY Hardware Adapter for PNT

These two options are illustrated in Figure 1 and Figure 2. Note that these patterns are functionally equivalent from the VICTORY standpoint. Neither approach is more compliant with the interface standards than the other. However, the costs and benefits are different, as will be discussed.

The Hardware Adapter Pattern

An HW adapter is a "box" with a port and connector that matches the native device interface, a VICTORY standard Ethernet port, and a power port to plug into vehicle power. It provides internal processing to perform the conversion between the native and VICTORY interfaces, and to support the other required interfaces such as management and health reporting. A HW adapter represents an additional Line Replaceable Unit (LRU) that is added to the design.

An example of the HW adapter pattern is shown in Figure 1. The example is a HW LRU designed to adapt a current-force GPS receiver, a Defense Advanced GPS Receiver (DAGR), to VICTORY network interfaces.

The DAGR has two serial ports that provide position, time and GPS status data in standard formats. In the example, the adapter receives the GPS data on an RS232 serial port on the LRU and also receives a 1 Pulse Per Second (1PPS) signal on a General Purpose Input/Output (GPIO) port. The HW adapter runs software that adapts these native device interfaces to VICTORY standard interfaces. The time synchronization service converts the 1PPS signal to Precision Time Protocol (PTP), which was adopted by VICTORY for network time synchronization. The position service receives the serial messages from the GPS receiver and implements the VICTORY position service which publishes position data and implements management and health reporting interfaces. The GPS receiver service receives status messages from the GPS receiver and publishes GPS status data, such as the Time Figure of Merit (TFOM) and the Signal to Noise Ratio (SNR) received from each GPS satellite. It also implements management and health reporting interfaces.

The benefit of the HW adapter pattern is that there is no modification necessary to any existing processing units to adapt a non-networked device. The drawbacks of the HW adapter pattern include:

1) A new LRU is added to the design to adapt each non-networked device. Although it is also possible to create HW adapters that adapt multiple devices in a single LRU, each HW adapter configuration must be managed by a program as an LRU.

2) The adapter LRU must include connectors for each input signal, the output Ethernet interface, and for vehicle power (labeled *Vcc* in the figure). That means that at least six connectors are required for non-networked device that is adapted (two for the power connector, at least two for the connection from the device to the adapter, and two for the connection from the adapter to the network switch). For comparison, if the device implemented the VICTORY interfaces natively, only two connectors would be required (one native and one Ethernet for the cable connecting the device to the switch).

The Software Adapter Pattern

A SW adapter consists of software that runs on a processor that has the appropriate interfaces and connectors. Software services run on the processor to adapt the native device interfaces to the VICTORY standards.

Figure 2 illustrates a SW adapter with equivalent functionally to the previous example, as it adapts a DAGR to VICTORY network interfaces. Note that the HW adapter pattern can be implemented using a processor that does not complying with the VICTORY SPU interfaces, but hosting the services on an SPU, as is shown in the figure, may reduce software porting costs. The fact that the processor hosting the adaptation services is a VICTORY SPU is evidenced by the presence of the SPU Service. The SPU also runs the VICTORY Data Bus (VDB) Management Service and VICTORY Access Control Services, although these do not play a direct role in the adaptation.

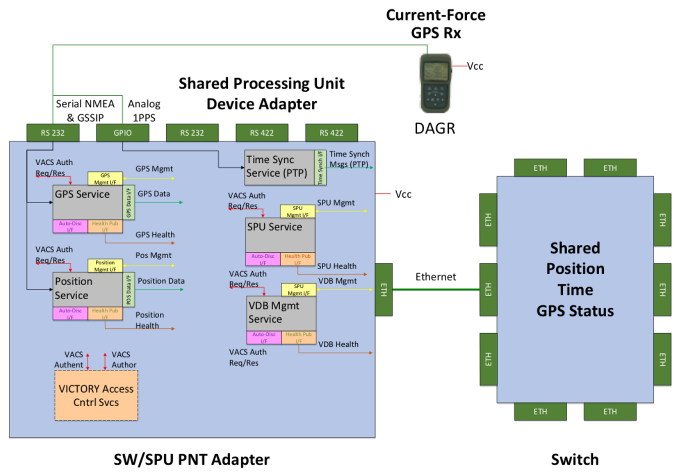

Figure 2. Example VICTORY Software Adapter for PNT

In the SW adapter pattern, the native port on the current-force equipment is cabled to a port of the same type on the SPU, and the adapter software interfaces that port through the operating system. In the case shown, the SPU API provides a standard operating system interface method, reducing the dependence of the software on the underlying operating system and hardware platform. A SW adapter does not represent an additional LRU if the processing unit already exists, which is the case when a VICTORY IVN is already implemented in the vehicle. SW adaptation only requires addition of cable to connect the native port of the device to a port on the processor.

This adaptation pattern is far superior to the HW adapter pattern, as the processor can be used to adapt several different non-networked devices simultaneously without implying additional LRUs. The main benefits include: 1) The number of devices adapted is scalable and is accomplished by adding only a cable and a software module. 2) Scaling is accomplished without adding additional LRUs for a program to manage. 3) If the SPU is already present in the design, and the correct type of port and connector is available, then adapting a non-networked device only requires two additional connectors (two native connectors for the cable between the device and the processor).

This is an extremely attractive approach to adaptation. Multiple products are being developed that can support the SW adaptation pattern. However, there are considerations that must be evaluated before it is seen as the most tenable and cost-effective approach for all situations. These considerations include:

1) The processor must provide a port and connector corresponding to the native port of each device it adapts. Since connectors drive the SWaP and cost (SWaP-C) of

devices, it is very important to balance the number and type of connectors to be included on the adaptation processor (SPU) with the overall cost across the set of configurations that are expected. Limiting the number and type of devices that the processor is expected to support simultaneously reduces the number of connectors, and thus reduces the SWaP-C of the processor LRU. Supporting adaptation of more devices increases the number of connectors and the SWaP-C of the SPU. Because it is burdensome and expensive for a program to manage several product configurations, particularly for a computer, there will likely be a very small number of SPU configurations available for vehicle programs to leverage. This encourages SPUs to be specified to include with far more types and numbers of connectors than will be necessary in most configurations, which drives up the SWaP-C of the SPU. On the other hand, if few ports are specified, in larger scale implementations, additional SPUs may be needed merely to support the adaptation of the current-force devices, not because additional processing is actually required. Striking a balance between the number of configurations and connectors requires analysis of the range of configurations that are likely to be supported by the SPU.

2) When non-networked devices are added to a system, the software build on the processor must be updated. This may trigger a round of operational testing information assurance analysis to re-certify the configuration. This drives both cost and schedule.

3) The last consideration is subtler, but perhaps most important of all. Not all vehicle programs will implement VICTORY IVNs at the same time. Within programs different variants will receive different equipment configurations, and some will not include IVNs. It must be assumed that there will be a significant number of vehicles that will not have VICTORY IVNs at least in the near term, so those vehicles will not include a VICTORY SPU or other processor that can host the adaptation software. This may not seem to be a problem at first blush, as it may seem needless to adapt equipment when there will not be a full VICTORY IVN. However, as will be explained, there are use cases in which adapting devices to the VICTORY interface standards will be highly beneficial, even in vehicles without a full IVN implementation. Without an SPU, the SW adaptation pattern is not possible, and fixed HW adaptation remains too costly.

The VICTORY Smart Cable Adaptation Pattern

Southwest Research Institute (SwRI) has conceptualized and prototyped an implementation of a third pattern, which may be more cost-effective in many situations. The concept is called the VICTORY Smart Cable, because adaptation is performed by a cable instead of by a fixed HW LRU or by software running on an SPU. As is illustrated in Figure 3, the VICTORY Smart Cable implements the logic necessary to

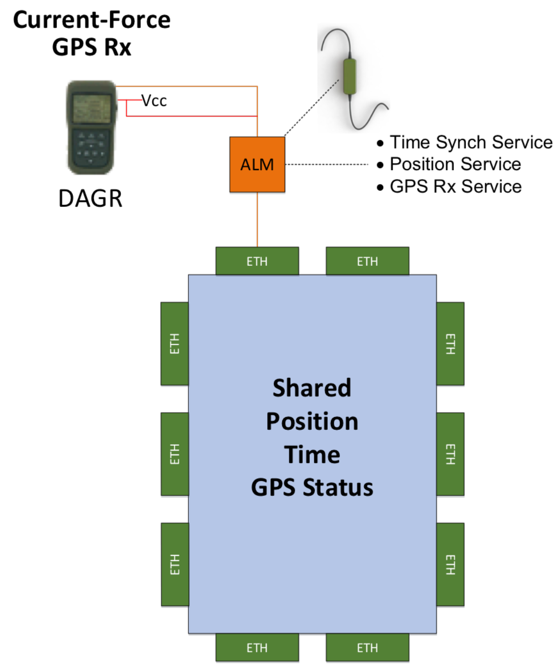

Figure 3. Example VICTORY Smart Cable for PNT

adapt the native protocol of the non-networked device to the VICTORY standard interfaces in the Adapter Logic Module (ALM), which is built into the cable. The ALM itself does not have removable connectors, and is considered part of the cable.

Logically, The VICTORY Smart Cable is almost equivalent to the HW adapter, but for a few important distinctions, which are beneficial. The benefits include:

1) The VICTORY Smart Cable can be managed as a cable, as opposed to as an LRU. ALM units with the cabling can be provided to integrators, and the cable lengths and connectors will be applied during integration.

2) It requires fewer MIL-style connectors than the HW adapter (the lower bound is three per adapted device, as opposed to six), which decreases the cost. The three connectors that are required include one native device connector, one Ethernet connector, and one power connector. Note that the SPU is sinking the 1PPS signal from the DAGR, as was shown in the earlier examples. Although the VICTORY Smart Cable requires one connector more per adapted device than for the SW adapter pattern, the cost scales better than the HW adapter pattern, and almost as well as the SW pattern, and it scales down for very small systems.

3) The VICTORY Smart Cable plugs directly into the IVN switch, not into the SPU, so it does not consume a non-Ethernet port (serial, CAN, etc.) on the SPU. This means that

SPUs may have very few native device ports, while the overall number of devices that can be adapted by a system scales to the total number of Ethernet ports in the system. As Ethernet switches with twelve to eighteen ports are commonly fielded, this is not seen as a restriction.

4) The VICTORY Smart Cable pattern can be applied to systems that do not include an SPU, or a full IVN.

5) The VICTORY Smart Cable provides a transitional capability, adapting non-networked equipment to VICTORY standards in the interim until vehicle programs implement the IVN capability, and product programs being implementing native VICTORY interfaces.

TECHNICAL APPROACH

VICTORY PNT Smart Cable Prototype

SwRI has developed a functional proof-of-concept prototype of the VICTORY Smart Cable, instantiated as a Position Navigation and Timing (PNT) device adapter. The cable implements and is compliant with the VICTORY Time Synchronization, Position, and GPS Receiver component type specifications. The VICTORY PNT Smart Cable includes RS232 and Ethernet ports, and is powered from the existing GPS receiver power cable. It has been demonstrated adapting DAGR devices and DAGR Distributed Devices (D3). Testing of the prototype demonstrated that the start-up time (from power-off to fully operational) as less than the DAGR acquisition time, and significantly less than the boot time of most SPUs. This means that the cable will be fully functional before the rest of the system requires its services. The SWaP of the current prototype (V1) is shown in Table 1.

VICTORY Serial Smart Cable V2

VICTORY Serial Smart Cable V2 is currently in preliminary design. The main goals of the update are to further reduce the size, to mature ALM electronics design, and to make the packaging more rugged. The ALM electronics platform will be capable of adapting devices with native serial interfaces to VICTORY standard interfaces. The estimated SWaP for the V2 ALM are shown in Table 1. The V2 ALM design is less than half the size of the current prototype, and is also lighter. Figure 4 illustrates the (draft) profile of the Version 2 ALM, based on current plans. The recurring engineering cost of the V2 ALM is estimated to be below $200 for production of 1,000+ units.

It is not yet clear which type of devices will be adapted in the V2 demonstration. Candidates include GPS Receiver, Inertial Measurement Unit (IMU), legacy voice radio (management interface), or EW device (Duke V3).

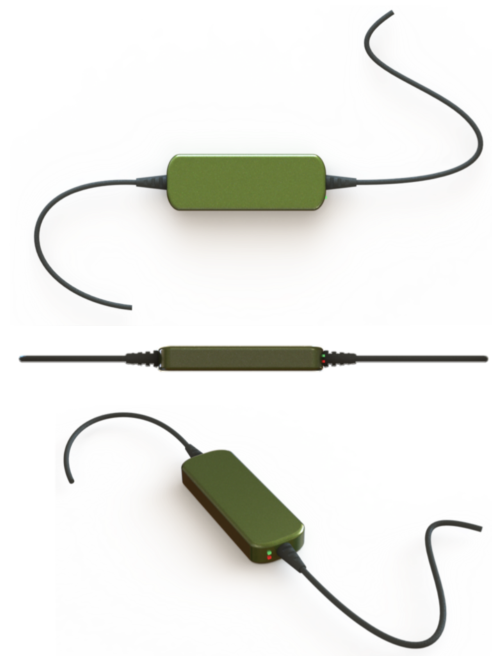

Figure 4. VICTORY Serial Smart Cable V2 Renderings

Table 1. SWaP of the VICTORY Smart Cable ALM Prototypes

Version	Size (Volume)	Weight	Power Consumption
V1 (current)	~6 1/2 in^3 (~107 cm^3)	~2.7 oz (~77 g)	~1 W
V2 (predicted)	~2 5/8 in^3 (~43 cm^3)	~2 oz (~57 g)	~1 W

Another possible demonstration scenario is to update the ALM platform to do “reverse adaptation” and emulate the interfaces required by another device. For instance, one demonstration target is to implement a VICTORY GPS Emulator, which adapts VICTORY Position, Time Synchronization, and GPS Receiver message interfaces to the serial data and 1PPS signals required by current devices. The potential benefit of the VICTORY GPS Emulator is to provide GPS-like interfaces to devices that depend upon a direct connection to a receiver. This would allow the GPS receivers that are currently embedded in or connected to these devices to be removed, which would greatly reduce the SWaP-C of integrated systems, independent of whether there will be a VICTORY IVN in the vehicle. Devices that are candidates for VICTORY GPS Emulator experimentation include Blue Force Tracker (BFT), Duke V3, the Manpack radio, and Mid-tier Networking Vehicular Radio (MNVR).

SwRI plans to garner interest and input from the government to direct the development and demonstration plans for the VICTORY Serial Smart Cable V2.

CONCLUSIONS

This paper has provided an overview of the main design patterns used in adapting current-force equipment to modular open network-based architectures such as VICTORY. SwRI has conceptualized and developed a technical demonstration of a new adapter technology called VICTORY Smart Cable. The technology demonstrator adapts GPS interfaces to VICTORY Time Synchronization, Position, and GPS Receiver compliant interfaces, and has been demonstrated with military GPS receivers in a laboratory environment.

The uniqueness of the PNT Smart Cable that sets it apart from the other approaches is that the cable *is* the adapter. The logic necessary to implement the VICTORY interfaces is embedded directly into the cable using the ALM electronics platform, as opposed to being implemented in software on an SPU, or requiring an additional Line Replaceable Unit (LRU). With this approach, one end of the cable connects to the native device port, and the other end connects to the IVN switch. As a result, the device can be treated as a cable, as opposed to an LRU, and that adaptation does not require any modifications to other LRUs or their software builds.

Benefits of the VICTORY Smart Cable technology include:

- Provides a complete adapter technology
- It is a cable, not an LRU
- It adapts without modifications to vehicle HW or SW
- It does not use SPU ports or processing resources
- It has a very low SWaP-C impact
- It starts up quickly and automatically
- The ALM platform itself is modular
- VICTORY Smart Cables could be assembled and integrated by the government laboratories, by third-parties, or by vehicle integrators

The SW and VICTORY Smart Cable adaptation patterns are both potentially cost-effective adaptation solutions. The relative cost-effectiveness depends upon the structure and requirements of the program. The VICTORY Smart Cable supports beneficial use cases in the interim until full IVN and products with native interfaces are developed.

The PNT Smart Cable is complimentary to ongoing VICTORY product and system development efforts, such as the Mounted Family of Computer Systems (MFoCS), PNT Hub, VICTORY-in-a-Box, and VICTORY Enabled Company Transformation (VECTOR).

The ALM hardware platform, with modifications to the adapter logic, can be used to adapt any devices with serial interfaces, such as voice radios and electronic warfare devices. The modular ALM platform can be extended to adapt non-serial device interfaces, such as analog video sources Controller Area Network (CAN) busses.

The VICTORY Smart Cable concept is not only applicable serial-based devices. The basic ALM electronics can be repurposed to adapt current-force devices with other types of interfaces, such as Ethernet (Boomerang, networked CREW devices, data radio management), RS170 video (Check Six, Driver Vision Enhancer [DVE], and Remote Weapon Station [RWS] video), and potentially CAN. Note that the ALM SWaP may increase depending upon the interface technology.

The VICTORY Serial Smart Cable could also be extended to support data filtering and data fusion. For instance, it could data from the VICTORY automotive system interfaces, and use the vehicle speed, gear, and engine speed as inputs to a PNT fusion algorithm that improves the certainty of position and direction of travel data.

And finally, the VICTORY Smart Cable concept and technology is generally applicable to the class of networking message-based modular open architectures. For instance, with logic modifications, Smart Cables could be developed to adapt the same types of current-force equipment discussed in the VICTORY context to interoperate with the ISA and FACE systems.

Please do not hesitate to contact the authors for more information or ideas for applications of the VICTORY Smart Cable technology.

REFERENCES

[1] M. Moore, et. al., "Modular Open System Architectures (MOSA) for Military Systems: Addressing Challenges of Complex Systems", Interoperable Open Architecture 2016 Conference, London, UK, April 27, 2016.

[2] S. Cole, “Vetronics Designers Focus on Reduced SWaP-C and Enhanced C4ISR”, Military Embedded Systems, July/August 2014.

[3] M. Moore, et. al., “Deploying VICTORY Architecture and Specifications”, 2012 NDIA Ground Vehicle Systems Engineering and Technology Symposium (GVSETS), Dearborn, MI, August 10, 2012.