Proceedings of the 2010 NDIA Ground Vehicle Systems Engineering and Technology Symposium (GVSETS)
Vehicle Electronics and Architecture (VEA) Mini-Symposium
August 17-19 Dearborn, Michigan

Update on the Mini-VCS (Vehicle Computer System) – Leveraging Vehicle Networks & User Interfaces for CBM+ (Condition Based Maintenance Plus) Diagnostics, Prognostics, and Sensor Integration

Mark Zachos (mark@dgtech.com)
President & CEO
DG Technologies
SAE Board of Directors &
SAE J1939-84 HD-OBD Task Force Chairman

Kenneth DeGrant (kdegrant@dgtech.com)
Field Applications Engineering Manager
DG Technologies
TMC RP1210 Task Force Chairman

ABSTRACT

This paper is a technology update of the continued leveraging of using the newest vehicle diagnostics system, the Smart Wireless Internal Combustion Engine (SWICE) interface as the Mini-VCS (Vehicle Computer System). The objective is to further enhance Conditioned Based Maintenance Plus (CBM+) secure diagnostics, data logging, prognostics and sensor integration to support improvement of the US military ground vehicle fleet's uptime to enhance operational readiness.

Evolving advancements of the SWICE initiative will be presented, including how the SWICE "At Platform" Test System can readily be deployed as a multiple-use Mini-VCS. The application of the Mini-VCS integrates the best practices of diagnostics and prognostics, coupled with specialized sensor integration, into a solution that optimally benefits the military ground vehicle fleet. These benefits include increased readiness and operational availability, reduced maintenance costs, lower repair part inventory levels, reduced cost of consumables, and an overall reduction in maintenance process errors.

INTRODUCTION

In the 1980's, the US Army developed a test platform called an ICE (Internal Combustion Engine) kit for the maintenance of their wheeled vehicle fleet. At that time, most of these vehicle's on-going operational issues were diagnosed through add-on analog sensors and transducers, with a few vehicle components being diagnosed through data bus technology.

DG Technologies, then known as Dearborn Group Technology, was included in these ICE kits in the form of a vehicle data link adapter, called the Dearborn Protocol Adapter (DPA), for the diagnosis of those early vehicles with data bus technology. Just as the ICE kit has evolved since then to provide for new programs and abilities described in this paper, the DPA has evolved in lock-step with ICE and overall advancements in vehicle network technology.

The ICE system served its purpose well. As the fleet of vehicles grew, the DPA was being used more extensively in conjunction with the growing number of IETMs (Integrated Electronic Technical Manuals) used by fleet technicians to diagnose and troubleshoot their analog and data bus, now vehicle network technology, equipment.

Although very expensive and requiring long lead times to develop, the IETMs worked great, especially for technicians new to multiplexed vehicles. At the same time, the natural progression in the commercial world was taking the ability to access vehicle data to the next level in what has been dubbed "sensor-based diagnostics", "prognostics" or, in general, "condition based maintenance plus" (CBM+). While the + symbolizes the next wave of applications and technology in this discipline, CBM in its simplest form can best be compared to the advancements in crankcase oil replacement. The first steps were to replace the oil "every X thousand miles", whereas now the oil is sampled and tested and replaced only when necessary. This provides the longest oil life, and a fleet's optimal return on investment, all while effectively providing engine protection.

In the commercial fleet, CBM+ is accomplished by downloading vehicle data captured by a data logging system (covered in detail later) such as hours, miles, faults, min/max temps and pressures, etc. The download can be somewhat expensive when using GPS/cellular systems (near-real-time-data) or essentially free by waiting for that vehicle to reach

one of the fleet terminal locations so that a technician can physically connect a vehicle diagnostic adapter to it. Over hundreds or thousands of vehicles, and over time, the CBM+ database can be "mined" to determine specific trends and patterns as to vehicle and individual component life. This process is sometimes called "Pattern and Trend Analysis (PTA)", and saves fleets thousands of dollars every year from expensive on-highway downtime incidents and even more importantly, preemptive parts replacement based mostly on general human observations.

In the military, preemptive parts replacement is colorfully referred to as "PM'ing a vehicle to death", and the troops and the technicians know that the parts they are replacing have nothing "major" wrong with them are really capable of a longer life. The problem is that the military does not have a way of determining what the "extra part life" is, and due to the nature of a potential combat scenario vehicle failure, must err on the side of caution.

Taking their own as well as commercial fleets' operational challenges into consideration, the US Army has charted a new path for the maintenance solutions that opened a whole new world to the development of CBM+ and other prognostic systems by adopting a concept and developing solutions called the Wireless ICE (WICE). Available today, WICE greatly reduces the size and cost of the vehicle maintenance kits as it introduces something significantly more important, the ability to wirelessly obtain and interact with vehicle data using a device that has a small footprint, small enough to always remain on the vehicle or "At Platform".

Building on the successful WICE program, the US Army has begun development of solutions called Smart Wireless ICE, or SWICE. The SWICE program provides the next piece of the puzzle in getting the US Army to their desired goal of CBM+.

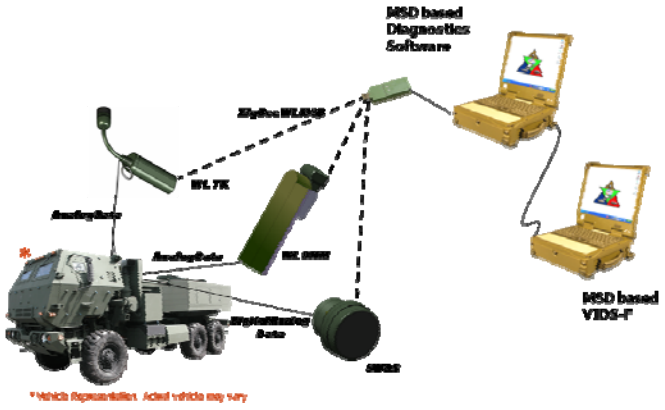

Figure 1 – SWICE Program Overview

SWICE (see Figure 1) includes a Smart Wireless Diagnostic Sensor (SWDS), initially providing a wireless vehicle datalink protocol adapter supporting several critical vehicle network protocols, including the most common (CAN/J1939 and J1708/J1587). SWICE also has an analog measurement component, the wireless transducer kit (WL TK), that provides access to data available via the analog data port in the vehicles (ICE), along with a Wireless Digital Multi-Meter (WL DMM) developed based upon a wired COTS DMM. These devices all communicate wirelessly via a USB device that plugs into a PC, called the "Interrogator".

The importance of the SWICE initiative is that the SWDS can either be used as needed, or mounted semi-permanently in the vehicle through interchangeable end-plates that mate to the various in-vehicle network port connectors.

Keeping the SWDS mounted to the vehicle data port allows the vehicle not only to share data (VIDS-F, or Vehicle Integrated Diagnostics Software-Fleet, described later) when it gets within range of a friendly vehicle, but also offload data via lower-cost Telematics platforms (as opposed to full-blown GPS/cellular based Telematics systems) including via "guard post" radio antenna. The current wireless protocol used is called Zigbee (802.15); however the SWDS was designed in a modular platform where longer range wireless communications (i.e. secure 802.11 – Wi-Fi, emerging WiMax) can be readily implemented.

Radio communications are not always practical or tactical, so an Ethernet communications cable is used as a backup communications channel. Security considerations, discussed next, surround the wireless Zigbee standards-based communication between the SWICE modules (SWDS, WL TK, and WL DMM) and the "Interrogator", the connected to the PC via a USB port.

WIRELESS SECURITY

Security is a primary and critical component of SWICE, whether dealing with wireless or wired communication with the wireless devices or modules: SWDS, WL TK, or the WL DMM. The general security architecture for any wireless PC access is via the wireless access interface on the PC. Often this interface is built-in to the PC, such as in the case of Wi-Fi or Bluetooth, but in the SWICE configuration, Zigbee wireless access interaction is via the Interrogator, which plugs into the PC's USB port and communicates using this same standard to the modules. While the SWICE security solution has been implemented, only high-level architectural considerations of this security can be described here.

The cornerstone this security architecture are appropriate cryptographic module(s), coupled with Federal Information Processing Standard (FIPS) Publication 140-2 “Security Requirements for Cryptographic Modules”. Available in its current format since late 2002, it was developed under the auspices of the U.S. Department of Commerce by the National Institute of Standards and Technology with participation and input from industry.

Basically, FIPS 140-2 is a U.S. government computer security standard used to accredit cryptographic modules produced by private sector vendors who seek to have their products certified for use in government departments and regulated industries (e.g. such as financial institutions) that collect, store, transfer, share and disseminate sensitive but unclassified information.

FIPS 140-2 establishes the Cryptographic Module Validation Program (CMVP) as a joint effort by the NIST and the Communications Security Establishment (CSE) for the Canadian government. In the CMVP, vendors of cryptographic modules use independent, accredited testing laboratories to have their modules tested. National Voluntary Laboratory Accreditation Program (NVLAP) - accredited laboratories perform cryptographic module compliance/conformance testing. All of the tests under the CMVP are handled by third-party laboratories that are accredited as Cryptographic Module Testing laboratories by the National Voluntary Laboratory Accreditation Program. Vendors interested in validation testing may select from about a dozen or so accredited labs.

FIPS 140-2 Security Levels

FIPS 140-2 defines four levels of security, simply named "Level 1" to "Level 4". It does not specify in detail what level of security is required by any particular application. The standard provides four increasing, qualitative levels of security: Level 1, Level 2, Level 3, and Level 4. These levels are intended to cover the wide range of potential applications and environments in which cryptographic modules may be employed.

Level 1

Security Level 1 provides the lowest level of FIPS 140-2 security, providing basic yet essential security requirements specified for a cryptographic module. No specific physical security mechanisms are required in a Security Level 1 cryptographic module beyond the basic requirement for production-grade components.

Level 2

Security Level 2 adds to Level 1 by requiring features that show evidence of tampering, including tamper-evident coatings or seals that must be broken to attain physical access to the plaintext cryptographic keys and critical security parameters (CSPs) within the module. Thus, Level 2 adds essential protection against unauthorized physical access.

Level 3

Security Level 3 attempts to prevent gaining access to CSPs held within the cryptographic module, and is intended to have a high probability of detecting and responding to attempts at physical access, use or modification of the cryptographic module. Examples of physical security mechanisms may include the use of strong enclosures and tamper detection/response circuitry that deletes data, resets data, etc. upon tampering

Level 4

Security Level 4, the highest level of security, provides a complete envelope of protection around the cryptographic module with the intent of detecting and responding to all unauthorized attempts at physical access. Penetration of the cryptographic module enclosure from any direction has a very high probability of being detected, resulting in the immediate “reset to zero” of all plaintext CSPs.

FIPS Software State Machine & Block Diagram

The FIPS Software State Machine (see Figure 2) provides an overview of all of the numerous states that a device may be in, and how this state moves toward resolution in one way or another.

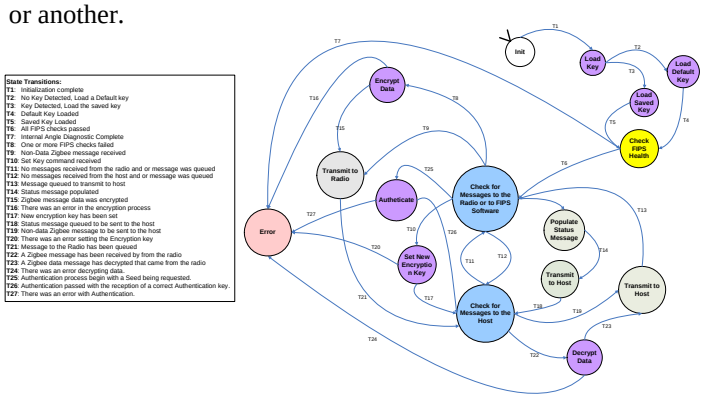

Figure 2 – FIPS Software State Machine

While this process, written down, looks complex, it is actually substantially similar to other many other and common software design maps, and is essentially a logic diagram.

The FIPS High Level Block Diagram (see Figure 3) applies specifically to software. There is now an encryption layer that is called upon to resolve key handling functions, as well as a message handling layer, used to encrypt and decrypt messages, actually better thought of as streams of information instances.

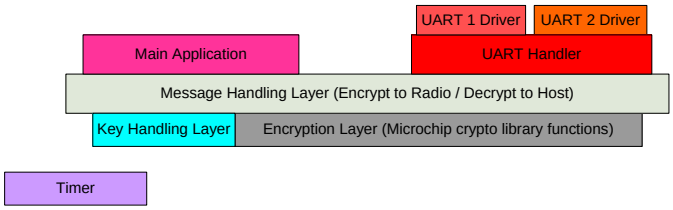

Figure 3 – FIPS Block Diagram – High Level

FIPS 140-2 for SWICE Overview

At this point, a summary description of how this information about security applies to the SWICE, and specifically wireless devices: SWDS, WL TK, or the WL DMM, is in order. Recall that the SWICE Kit uses a wireless network protocol, Zigbee, to communicate with these various wireless devices of modules, and each instance of connection is required to be FIPS 140-2 compliant.

Now it is evident that, for example, SWICE at a Level 2 certification requires additional security, including authentication of the encryption device as well as evidence of tampering or attempted probing of the encryption device.

An encryption device can be used not only to encrypt/decrypt data with a certifiable set of algorithms, but also allows for a single or harmonized encryption solution for all of the SWICE kit modules or any other module that require encryption with Zigbee.

Security Authentication Requirements

In this example, there are real-world requirements of FIPS 140-2 level 2 certification. At every power cycle, the FIPS chip must be authenticated. Such authentication requirements include that for each attempt to use the authentication mechanism, the probability shall be less than one in 1,000,000 that a random attempt will succeed or a false acceptance will occur. There are also requirements for multiple attempts & random attempts to use the authentication system that must be met.

To handle these requirements, typical encryption chips use a Seed/Key Exchange between the Host Processor and the FIPS chip that limit authentication attempts.

In summary, the SWICE platform components described in the previous section, coupled with proper security, are preparing the US Army for the future of vehicle diagnosis and CBM+, providing a solid foundation for future advancements, including leveraging the SWICE SWDS as a Miniature Vehicle Computer System (Mini-VCS).

BUILDING ON SWICE – INTRODUCING SWDS, AN AFFORDABLE/SCALABLE MINI-VCS

The decades spent by DG Technologies in continuous improvement of the mainstream DPA product along with development in keeping up-to-date with the US Army and Marine Corps with their ICE, WICE, SWICE and VADS projects have made DG a leader in Vehicle Diagnostic Adapter (VDA) technology on the hardware, driver, and software fronts.

When dealing with capabilities past the generic “plugging-in” or “connecting wirelessly” into the vehicle diagnostic port and performing diagnostics, (1) near real-time vehicle information transfer via wide area networks, and (2) CBM+ are at the top of the list.

While commercial providers of (1) include Qualcomm, PeopleNet, and Xata, properly rolling out (2) CBM+ requires a very large infrastructure to be in place at all levels within a commercial or military fleet. This includes computer servers with large disc capacity, high speed database systems, as well as the software and reports to mine the database for patterns and trends. One of the toughest aspects is to get that information, in an understandable form, back to fleet maintenance managers.

For the purpose of this paper, we shall spare the reader the detailed work effort involved in the planning and implementation of a CBM+ back-end system, and will instead focus on getting a leveraged hardware computer platform into the vehicle that is able to collect and process the on-vehicle data. Besides this proper hardware platform, the best solution will include an API to support functions of the prognostics client "plug-in" module and integrated support for the Common Logistic Operating Environment (CLOE) implementation.

This best solution is the SWICE's SWDS serving as the Mini-VCS!

LINUX "IS ALL THAT" TO SUPPORT CBM+ ON-VEHICLE HARDWARE

Proprietary and single-purpose electronic platforms were "all the rage" in the 1970's and 1980's until the arrival of the first personal computers and multi-game arcade systems. These computer systems introduced us to a concept of a "standard" hardware platform running a standard Operating System (OS) on top of a generic Basic Input Output System (BIOS). Software engineers could then design software that could be mass distributed with a great chance that their software would run on any generic system. Our OS for this article will be focused around Linux.

The Linux Operating System (OS), a publicly developed and supported operating system, running on standard PCs has been a very rapidly growing market, and a host of application programs (such as Open Office) are making it very competitive with Windows™ and Mac™ (another UNIX-based OS). Now, the latest "rage" is the embedded, micro-PC market with Linux at the heart. One of the greatest attributes about the Linux platform is that a UNIX® platform is generally what software engineers learn on and develop for in college, providing a great annual crop of engineers with the skills to readily develop applications using the Linux OS. Since Linux is also free, it makes it very attractive for companies looking to employ a real time embedded operating system (RTOS).

The backbone of the SWICE is the SWDS device, which is a Linux-based personal computer platform. It forms a leveraged solution as the Vehicle Diagnostic Adapter and the heart of the CBM+ system as the Mini-VCS. As CBM+ applications are developed and ported to the Mini-VCS, the solution will have great longevity, because one of the other great Linux attributes is that the operating system has been ported to a wide variety of processors. In the event that the processor in the SWDS ever becomes outdated, or no longer produced by the manufacturer, the core functionality can easily be ported to another processor.

THE MINI-VCS: YOUR LEVERAGED ALLY

The SWDS as the Mini-VCS (see Figure 4) has the horsepower and onboard functionality to easily support CBM+ today, tomorrow, and for the future. Some of the attributes are:

- Processing power rivaling that of a PC.
- Standard data storage capability of 16GB upgradeable using larger flash devices.
- Communication capabilities including USB, Ethernet, Zigbee, and secure Wi-Fi.

Figure 4 – SWICE's SWDS Unit – The Mini-VCS

Since the Mini-VCS is connected to the vehicle diagnostic port, is has access to information flow of almost any vehicle, allowing the creation of a seamless prognostic or CBM+ application.

Although, the SWDS lacks a monitor and a keyboard, it makes it even more attractive and affordable for more on-vehicle applications. It frees up space, can leverage existing monitors and keyboard/mouse types of devices, and helps reduce gross vehicle weight (GVW) and vehicle device interface complexity.

Command and Control (C2) type vehicles will always have the need for a fully functional PC (and sometimes more than one); however, the SWDS as the Mini-VCS can enhance these and other vehicles by being able to:

- Monitor the vehicle network parameters and faults, along with analog vehicle data.
- Provide visual indication of vehicle faults through a simple LED, without the need for a PC.
- Provide data to leveraged display units and/or PC(s).
- Process and archive the data to flash or disc.
- Forward the data to appropriate CBM+ servers whenever entering a friendly radio tower.
- Notify on-site maintenance personnel that the vehicle is now due for maintenance.

By providing all or just a subset of the aforementioned functionality, the Mini-VCS becomes a great asset in those vehicles that would, by budget, not normally have any type of onboard/off board computing device.

As a side note, the Mini-VCS could transmit encrypted (SWDS is FIPS 140.2 Level 2 compliant) Identification Friend Foe (IFF) data to other US military vehicles in the vicinity, increasing the chance they will be recognized and decreasing the chance of friendly fire incidents. This would be an asset to those vehicles that, by budget, have not been IFF equipped. Other outstanding advanced features designed into the architecture are concepts like smart hand-off, prognostics, and convoy mode operations.

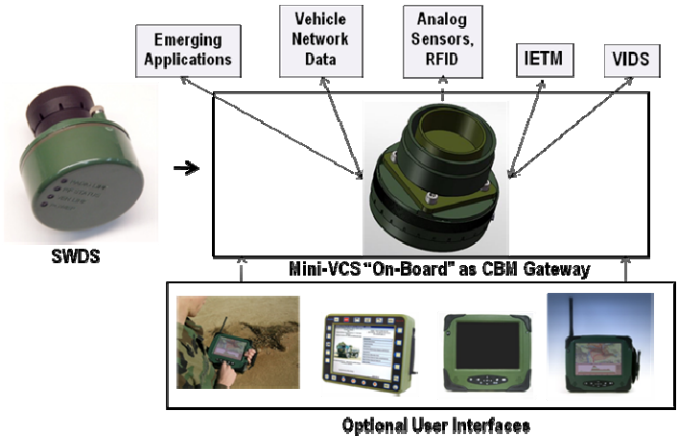

Figure 5 – Existing & Emerging CBM+ Applications Hosted on the Mini-VCS

We will now step through using the SWDS as the Mini-VCS (see Figure 5), supporting IETM & OEM Software, Diagnostic Software, VIDS-F, Analog Data/Emerging Applications, Data Logging with Examples, and Prognostics. Finally, the importance of Security using Data Encryption is discussed.

MINI-VCS: RUNS IETM & OEM SOFTWARE

The Mini-VCS can be the leveraged as the mechanism to store and provide IETM information for the vehicle. By using particular data about a single vehicle variant, we can minimize the extraneous information and offer very specific and relevant IETM data. As IETM provides a portal to manage technical documentation, IETM software packages are very useful and complete packages. Providing this data for developed IETM platforms and variants is great leverage.

The IETM software packages are very good, but they are very expensive and take a long period to develop, sometimes having vehicles being fielded before the IETM can be finished. Also, a separate IETM must be developed for every vehicle variant. Outside of the IETM, we have mentioned the generic DS diagnostic software and the OEM specific software, however have not mentioned how OEM software fits into the Mini-VCS picture.

The US military recognizes the issues with fielding IETMs and has started slowly integrating OEM diagnostic software. More and more staff and non-commissioned officers are receiving OEM specific software training who then train their subordinates. OEM software is typically written to the Technology and Maintenance Council (TMC) standard called RP1210. The Mini-VCS comes standard with a Windows 2000/XP/Vista/Windows 7 RP1210A-compliant set of drivers, therefore the majority of OEM diagnostic applications will work wirelessly or wired through the Mini-VCS.

TMC RP1210, SAE J1708/J1587 and SAE J1939 have been recognized by the US military as vital standards to be followed in future designs, and the US Marine Corps enlisted DG Technologies to author a document that has wording designed to be placed in future vehicle contracts ensuring that a vehicle is completely "diagnostically compatible" with these standards. The document may be of interest to the US Army and other branches of the service in order to ensure End-To-End (E2E) vehicle diagnostic compatibility.

THE MINI-VCS AND DIAGNOSTIC SOFTWARE

The SWDS, and therefore the Mini-VCS, is capable of running non-OEM-specific Diagnostic Software (DS). It implements simple features commonly found in OEM diagnostic applications, such as:

- Vehicle and component identification (i.e. make, model, serial number, software version, etc.)
- Vehicle fault detection and analysis (i.e. generic J1939 and J1587 fault codes)
- Vehicle data monitoring (digital and analog)
- Vehicle data archiving & Special test routines
- Data forwarding for fleet maintenance purposes.

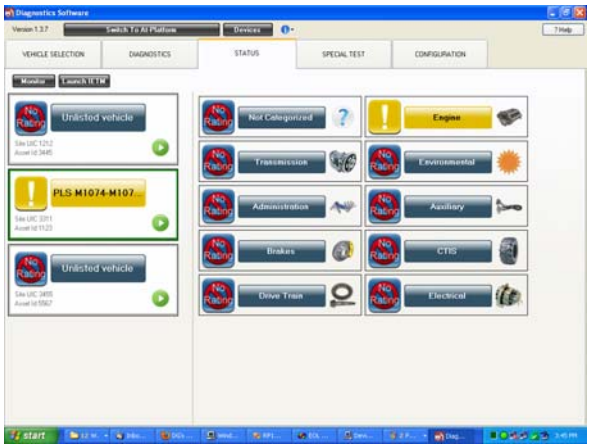

Figure 6 – DS Vehicle Status Overview

The DS provides very good triage functionality (see Figure 6). It has been noted that the majority of fault codes observed in the commercial fleets can be solved without needing to enter into that specific OEMs diagnostic software. It has also been noted that maintenance personnel tend to memorize the most common fault codes for a particular vehicle type, giving them the ability to solve the vehicle's problem without the need to enter the OEM software.

The best part of the DS is that it is architected in a client-server model (see Figure 7) where a high degree of decoupling has been achieved between the user interface and data processing layers. This is done knowing that the application will most likely be needed to be ported to alternate operating systems and software environments.

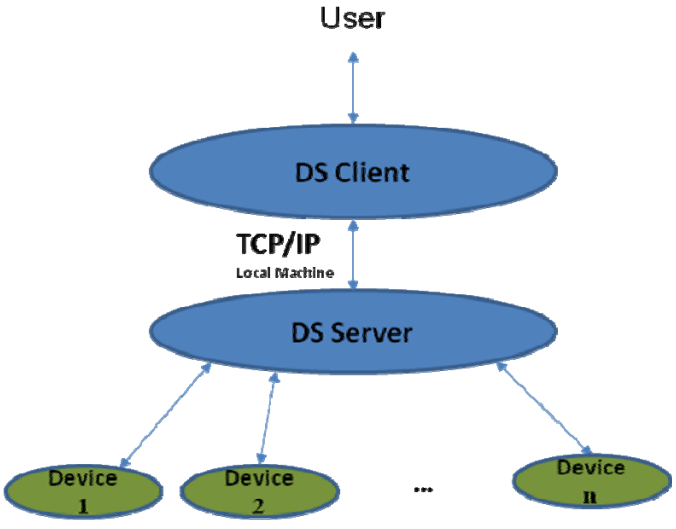

Figure 7 – DS High Level Architecture

Finally, an example of the DS in action using the client-server model and providing vehicle signal monitoring is shown (see Figure 8).

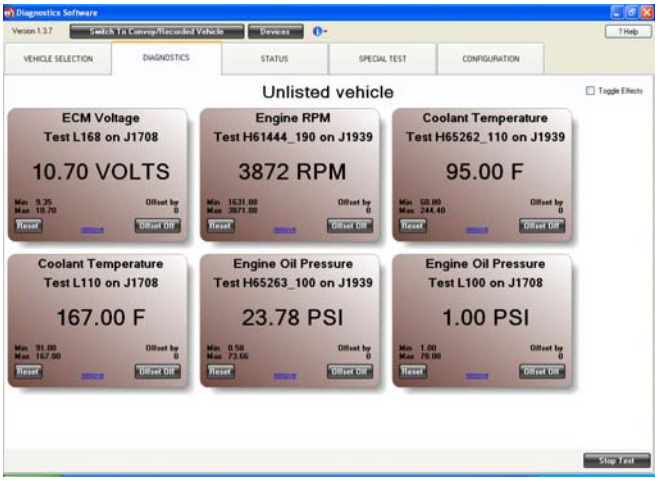

Figure 8 – DS Vehicle Signal Monitoring

MINI-VCS: RUNS VIDS-F FLEET MANAGEMENT APPLICATION

A fleet management application (VIDS-F) is a companion application to the DS that will be used by vehicle fleet managers to track and maintain the health of all the vehicles in their fleet. VIDS-F closely integrates with the DS data and proves additional features like:

- Vehicle fleet health monitoring
- Vehicle maintenance scheduling and tracking (see Figure 9)
- Prognostics
- Fleet reports (see Figure 9)
- Interface to Army order processing applications

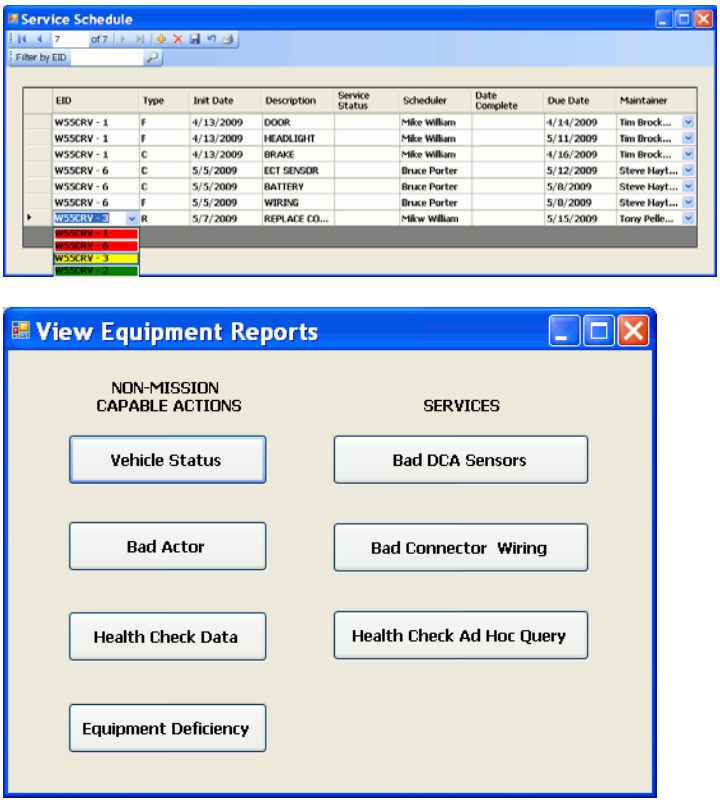

Figure 9 – VIDS-F Showing a Service Schedule Form and Fleet Report Menu

The VIDS-F and DS use MIMOSA data standards extensively for data exchanges between themselves as well as with other Army systems. In the authors' opinions, one of the greatest money and time saving capabilities would be to incorporate SWDS data, through the VIDS-F platform, into the current equipment repair order (ERO) system.

MINI-VCS: SUPPORTS VECHICLE ANALOG DATA & EMERGING APPLICATIONS

There are many analog data elements on a vehicle that are not part of a vehicle databus or vehicle network, and the Mini-VCS can easily be included in capturing these increasingly important data points.

A very high-end feature of the SWDS platform is that it has analog input signal capability. This allows the SWDS to be configured for, monitor, store, and report on various sensors, transducers, thermocouples, and pulse-type signals that are not otherwise found on the databus. Although what sounds like a very "common" type of functionality is actually not that common due to the added expense of adding protection to the circuits and software to do signal conditioning. In the commercial world, there are various type of "I/O Modules" (input/output), but they are generally very expensive, highly specialized, and are usually meant for engineering applications.

Being able to monitor analog parameters gives the fleet manager an additional tool to fight "pre-emptive parts replacement". For example, being able to monitor hydraulic oil temperature and pressure from a Mine Resistant Ambush Protected (MRAP) could help the fleet manager decide when to replace filters and/or rebuild the hydraulic motor as opposed to "every X months/hours". Another example includes a vehicle's wet air tanks, which are generally not connected to the databus. A lot of money and downtime could be saved by patterning and trending the air system to detect when valves, lines and fixtures start leaking, and when the air drying desiccant needs replacement.

Not only does this analog input help the fleet manager in the maintenance bay, but it could help commanders on the battlefield as well. Something as simple as a low-round-count off/on "electric eye", or a weight transducer in the 25mm ammunition box of a US Army Bradley Fighting Vehicle could alert battlefield commanders as to the fighting readiness of that vehicle. A company commander could make platoon level adjustments to have vehicles with higher round counts engage while lower round count platoons disengage and re-link more ammunition to the existing belt.

In general, each type of military vehicle has its own "special" types of issues that are not being monitored by an electronic controller on that vehicle and could benefit by being connected and monitored by the SWDS. For a list of these existing and/or emerging applications, one needs to go no further than a company or battalion maintenance chief for ten to twenty more suggestions on what could be monitored that would assist in keeping their vehicles online.

An example of existing data that is seldom used includes capturing real-time data for appropriate use and analysis, so called **"data logging"**. Finally, while we have given some examples of existing data use, emerging applications include supporting off-vehicle network data, such as inventory levels via RFID tags, and reporting this information to the Mini-VCS for storage, logistics, archival, and backup.

MINI-VCS: SUPPORTS DATA LOGGING

Amongst the many Mini-VCS functional attributes, the Mini-VCS also includes the ability to function as a real-time data logger which is important to future CBM+ and prognostics efforts. A data logger in the short-term can assist fleet operators in enhancing predictive maintenance

schedules, with the long-term end-goal of platform-level CBM+ and prognostics. The Mini-VCS provides a powerful and sophisticated, yet easy to configure platform for data logging.

Since the Mini-VCS has the ability to connect and communicate on any of the vehicle networks, it can also store copies of any of the messages it finds on a network in to a disk "file" in Mini-VCS's memory. This file can then be uploaded for pattern and trend analysis (prognostics) or it can be used to play back a scenario seen in the field by an technician or engineer investigating vehicle performance issues, for example, "rough shifting".

Filters and Triggers

One negative aspect of data logging, the most commonly seen, is that of information overload. Too much data can be overwhelming, and very time consuming to sift through. Eventually, the data gets pared down to meet the exact need. To pare the information down to the right level, two tools can be deployed in the Mini-VCS, filtering and triggers.

Simple filtering allows only specific messages of interest (possibly carrying more than one data item of interest) to be recorded at an appropriate time interval. This requires more post-processing work on an external PC to extract the data points from the messages. In complex filtering, data value(s) inside a message are used when selecting what information to store. This is done internally by the Mini-VCS, making for less post-processing work, and is the preferred method.

As filters tend to be based on values or time intervals, the Mini-VCS also employs a more powerful feature called "triggers". A trigger can best be described as an "event-based" reason for logging of data. Triggers differ from filters in that when a specific data value meets a trigger condition (i.e. the "event"), a pre-described action takes place. Actions may include starting/stopping logging, logging x minutes before or after the trigger, applying a set of filters, sending a message, or just writing an event to the data log. Triggers can also activate other triggers, a term called "stacked triggers". A very popular trigger condition in commercial fleets is one for "hard braking" (an x mph/second deceleration of the vehicle).

A PC-based graphical user interface (GUI) makes the Mini-VCS easy to configure, including filters and triggers. To simplify the process, and to allow for more detailed hands-on configuration, filters and triggers are defined in an XML language file that can be hand edited, and can be deployed to other Mini-VCS systems. Once the file has been created, the file is downloaded to the Mini-VCS and from there, the logging and triggering operations are automatic.

Data Storage & Processing

Once filtering and triggering has been activated, the next consideration is how to store the data. There are two camps when it comes to data storage, simple ASCII and binary. The Mini-VCS team chose not to implement the data storage in binary form, as any file captured and transferred would mandate post-processing to make it human readable. Instead, the Mini-VCS captures and stores the data log in a human readable ASCII format, which can be immediately interpreted. The file is also marked (i.e. timestamp, data bus, message header, etc) in such a way that it would be easy to write a post-processing PC-based application to parse and store the data in a database for later predictive maintenance or prognostics.

The files stored on the Mini-VCS are on a removable Flash memory card. When the Mini-VCS is configured to perform data logging functions, one of the parameters is how much memory to set aside for storage of these data log files. When the end of storage nears, the Mini-VCS can be configured to overwrite the oldest data, or to stop collecting data entirely. When the memory is full, a trigger can be set to perform other actions.

With consideration for the "too much information" syndrome and the amount of memory available, the Mini-VCS can be deployed for months before the memory allocated for data logging has been exhausted.

For example, this log-file format, which is overly verbose, is about 185 bytes for a simple J1939 message:

Rx J1939 TS=[76419] Chan=[0] EB=[off] PGN=[61450|0xF00A] PF=[0xF0|240] PS=[PDU2-GE|0x0A| 10] HOW=[N/A] P=[6] SRC=[0] DST=[255] DL=[8] DATA-HEX[FF][FF][00][00][FF][FF][FF][FF]

If a computer was configured with 8 GB ($8 * 2^{30}$ bytes) of logging space, this message, logged once every second could be recorded for about 537 days.

8,589,934,592 bytes / (185 bytes * 60 seconds * 60 minutes * 24 hours)

8,589,934,592 / 15,984,000 = ~ 537 days

Off-board Processing

Once the data files are written, they may be retrieved from the Mini-VCS by simply uploading them to a PC using any wireless technology or a wired Ethernet connection. The most cost-effective means of download would be done wirelessly as there is no need for human intervention. Any type wireless communications strategy (i.e. Wi-Fi, Zigbee, and Bluetooth) could be used, and encryption could be employed to establish proper and required data security.

Configuring a Data Logger

When configuring a data logger (see Figure 10), we must select which from among Active Channels to monitor. With a proper Graphical User Interface (GUI), we first need to specify a Protocol (here, High Speed (HS) CAN), and set up various criteria. In this example, we have set up a Filter to monitor messages (hexadecimal values) $110, $124, and $308. Finally, we need to specify Actions, such as Logging the data, and optional advanced features such as formatting messages to be sent upon particular criteria being met, and/or schedules or times/dates to run the various data logging functions.

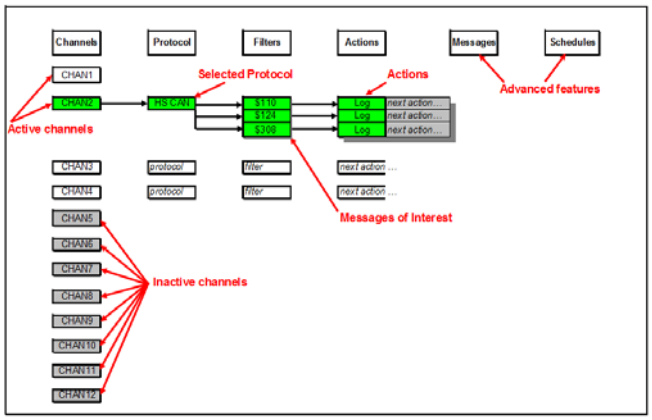

Figure 10 – A GUI to Configure a Data Logger

Data Logger Output File Explanation

Using the above example, and assuming that we were monitoring channels, 1 and 2, Figure 11 is shows what we would expect to see as output.

The Initial Trigger Time Stamp appears on the top line, and at the first list of the logged CAN messages. The Initial Trigger Time Stamp serves to give the data log file a unique file name as well as a reference point for beginning post processing of the logged messages.

Make no mistake, post processing of information is a significant task, as data is being sent from many sources in sometimes 1000 or more per second. This is why we must determine how often to capture data, as well as various conditions that need to be met; to isolate or minimize the amount and type of data captured, or to sort it all out with a post-process specialized software program that has a reasonable amount of data to parse.

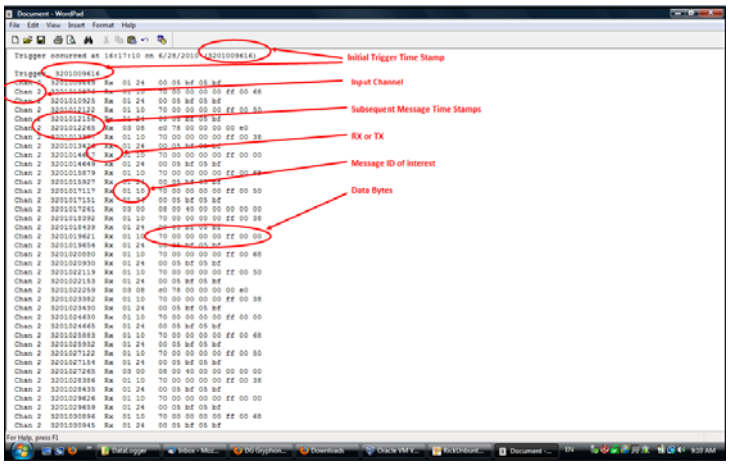

Figure 11 –Data Logger Output File Example

The Input Channel indicates the source of the particular logged message. A data log file may contain messages from many different channels and protocols. By indicating the source channel in the data log file, using the aforementioned specialized post processing software or other means, it would be a simple matter to sort the file by the Input Channel to group all the messages from Channel 1, Channel 2 and so on.

The Subsequent Message Time Stamps indicate when each of these additional messages were read on the input channel. The Mini-VCS provides a granularity of 10 uS. The message stamped “3201012265” was captured 1090 uS (or 1.09 ms) after the previous message, (above it in the table) “3201012156”. This is found by subtracting the two numbers, resulting in the value 109 units of 10 uS, or 1090 uS. Again, this data logging lends itself to easy formatting and manipulation during post processing.

The RX or TX column indicates whether the Mini-VCS sent (Transmitted Tx) or received (Rx) the message. In this example, the Mini-VCS is acting strictly as a data logger, so all values are Rx. There may be instances where the Mini-VCS must transmit certain messages to prevent the system from going into a particular state, for example, a “sleep mode”. Another example would have the Mini-VCS transmit

a message to initiate a certain sequence of events. We could capture all of those Tx transmission "commands" and note the responses associated Rx responses in the data log as well.

The next column shows the Message ID or Message Identifier that was logged. For simplicity, in this example, we've only captured 3 Message IDs. We could once again sort by Message ID and examine their time stamps or other associated data.

The Data Bytes are the data that accompanies each message. Each bit or groups of bits among the data for each specific Message ID has its own meaning. By applying a data base which maps Signal Names and Conversions to the data, we can determine the actual engineering units for each signal. This would be yet another operation that could be undertaken during post processing.

There exist many powerful and versatile software tools for the post processing and management of data log files. From the above explanation and example, it is hoped that you can clearly see that they are extremely important, and one should not be short-sighted in the proper selection of the appropriate software tool(s) used in conjunction with the Mini-VCS used as a data logger. This software for post-processing, of data, therefore, holds great value.

MINI-VCS: SUPPORTS PROGNOSTICS

The Mini-VCS will implement a Prognostics Host with a published specification for designing prognostics modules that can then be hosted on the Mini-VCS. At a minimum, the Prognostics Host will marshal both digital and analog data and make it available to the prognostics modules for processing, data mining, and for interaction with end users (see Figure 12). All data will conform to the MIMOSA standard.

With the approach taken by the Mini-VCS team, third-party prognostics applications can easily be written for the Linux OS and hosted on the Mini-VCS. This will allow vehicle experts to develop data logging and prognostics algorithms along with the transmittal of that data to the VIDS-F application without being bogged down by details regarding vehicle networks.

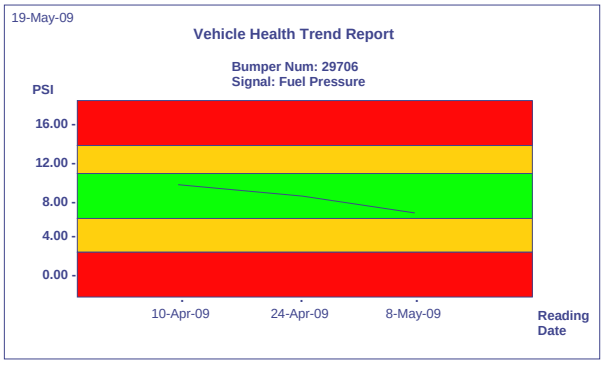

Figure 12 – A Prognostics Report (Using VIDS-F)

The DS software compliments this by providing facilities for upload/download of algorithms, enable/disable algorithms, the upload of CBM+ related data and forwarding to VIDS-F for fleet analysis.

CONCLUSION

It should now be evident from the information and positions stated in our paper that the SWICE SWDS, leveraged as the vehicle's Mini-VCS, further enhances economic and logistical viability of Conditioned Based Maintenance Plus (CBM+), and supports improvement of the US military ground vehicle fleet's uptime to enhance operational readiness.

Furthermore, with plenty of space coupled with sophisticated filtering, triggering, and security, the Mini-VCS is a mission-ready data collection device to assist the military fleet meet their predictive and prognostic goals as a short-term leverage to begin immediate cost-savings of CBM+.

REFERENCES

[1] Simplified Test Equipment/Internal Combustion Engine (STE/ICE).

[2] SAE J1708/J1587 and J1939 standards.

[3] MIL-STD-810F, Environmental Engineering Considerations and Laboratory Tests.

[4] MIL-STD-461E, Requirements for the Control of Electromagnetic Interference Characteristics of Subsystems and Equipment.

[5] MIL-STD-464A, Electromagnetic Environmental Effects, Requirements for system

[6] ATA/TMC RP1210A, RP1210B standards.

[7] FIPS PUB 140-2 “Security Requirements for Cryptographic Modules” U.S. Department of Commerce, National Institute of Standards and Technology May 25, 2001 with Change Notices 12-03-2002.