Implementation of MilCAN on a Main Battle Tank

Abdul Qabaz, BAE Systems Land Systems

This paper details the first implementation of MilCAN on a UK military vehicle. The UK Challenger 2 (CR2) Main Battle Tank manufactured by BAE Systems Land Systems has been updated to include improved Commander's Situational Awareness (SA) with new technologies communicating over MilCAN.

Situational Awareness is critical to any fighting force allowing Battlefield Commanders to make better tactical decisions in a reduced timeframe, whilst providing the Commander with mission planning capabilities such as generation and monitoring of orders, reports, overlays etc.

Currently, command and control on the battlefield is based largely on manual processes for the monitoring and planning of operations. The introduction of the Platform Battlefield Information System Application (PBISA) System for CR2 interfaces with both new and legacy vehicle sub-systems and allows automation of many of the currently manual tasks. The MilCAN data bus provides the communication between several processing resources on the vehicle and presents the vehicle Commander with essential information. The MilCAN data bus also distributes accurate and up to date position data to both Commander and Driver at their respective displays in a simple and easy to read format.

Introduction

Critical to any fighting force is the concept of Situational Awareness (SA), where, information such as, '*Where am I*?' or '*Where is the enemy*?' would allow Battlefield Commanders to make better tactical decisions in a reduced timeframe. In addition to SA is the mission planning capability, Command and Control (C2), which provides the Commander with the capability to generate and monitor orders, reports and overlays.

The current mechanism for command and control on the battlefield is based largely on manual processes for the monitoring and planning of operations. It relies on the use of hand-written logs, manual map boards and hand-drawn overlays. Furthermore, at the tactical level, there is no automated command and control support for fighting vehicle crews.

The Platform Battlefield Information System Application (PBISA) System for CR2 interfaces with both new and legacy platform sub-systems, capturing platform derived sensor information that augments the SA picture and other Battlefield Management System (BMS) reporting functions. Thus improving both the Commander's SA and mission planning/execution, whilst reducing his workload.

The Commander is provided with a visual display, which uses graphics images to display SA data on a digital map, as illustrated in Figure 1.

Figure 1 – Commander's Display

The SA data displayed includes own vehicle position and bearing, positions of enemy/friendly objects and installations, and positions of other battlefield obstacles.

Assistance with the navigation of the vehicle is also provided via the driver's display. This includes automatic update of own vehicle position, route planning and waypoint entry, as well as automatic update and transmission of 'steer-to' and 'distance-to' for the driver, as seen in Figure 2.

The MilCAN data bus is used as the primary communication network for the PBISA system.

Figure 2 – Driver's Display

The MilCAN Protocol

MilCAN A uses the 29-bit, extended, identifier format defined in ISO 11898-1. As with any CANbus implementation, data is broadcast, and bits 0 to 7 of the identifier include the physical address of the device that actually transmits a frame, rather than a destination address. Thus enabling multiple remote nodes to determine where a message originated and distinguish between similar messages from different devices. The MilCAN A protocol also defines message type and a priority level using bits 26-28 of the frame identifier in order to allow the system designer to allocate priorities on a message specific basis as part of a latency guarantee within the deterministic message transmission protocol.

The MilCAN application layer adopts a segmented message assignment scheme and flexible deterministic protocol in order to accommodate both vehicle and application dependent changes. The MilCAN 29-bit identifier is illustrated in Figure 3. Bits 28-26 define the priority of an individual message and therefore the priority of the associated data frame, as shown in Table 1. Bits 23-16 define the 256 message primary types and bits 15-8 the 256 message sub-types for each primary type.

Messages are grouped by means of their primary types e.g. Navigation, Power Management, Data acquisition etc. Efficient message allocation is vital, but it is important to recognise that when the initial allocation is made, the number of physical instances of that function can not be predicted. For example, consider a message designed to control a particular camera function. At the outset it is not known how many cameras will need to be supported, and indeed the number and location may vary from vehicle to vehicle. To support this requirement MilCAN defines a multi-instance addressing scheme that is independent of message type.

If applicable to a particular system function message, then the MilCAN A physical instance element is carried within one byte of the data payload rather than the source address field. MilCAN provides a network data flow structure to accommodate such real-time needs without requiring non-time-critical devices to incur the overhead of complicated time-slice transmission. Devices connected to a MilCAN network will vary greatly in their capabilities, hence support for deterministic message transmission must be provided by both sophisticated and simple devices.

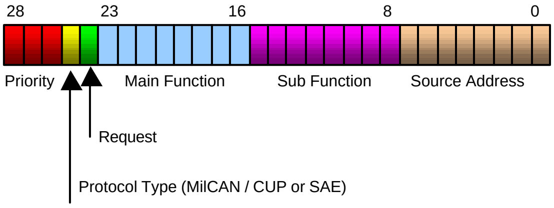

Figure 3 – MilCAN Frame Identifier Format

MilCAN uses a prioritised bus access with bounded throughput protocol. This supports determinism for those devices that require it while providing sufficient flexibility for those devices that do not.

Put simply, a number of time unit levels are defined; each has a particular latency guarantee and individual nodes are only allowed to transmit one message of each type within their allocated period.

Assuming the largest possible data field (8 bytes), the CAN protocol guarantees the delivery of 15 messages within each Primary Time Unit (PTU). In an extreme case this could equate to 15 Level 1 messages. In practice each PTU is more likely to support some level 1 messages with the remainder of the 15 slots being spare or made up of a combination lower priority Level 2, Level 3 or Level 4 messages. I.e. Hard Real Time (HRT), where timing and latency are very critical to system performance; Soft Real Time (SRT), where timing is still important, but absolute latency is not critical and Non Real Time (NRT), where there is no latency requirement.

Priority	Latency Guarantee
0 (highest)	Protocol Operation Messages (e.g. SYNC) and low jitter messages.
1	HRT Level 1 – 2ms at 1Mbit/s
2	HRT Level 2 – 16ms at 1Mbit/s
3	HRT Level 3 – 128ms at 1Mbit/s
4	SRT Level 1 – 16ms at 1Mbit/s
5	SRT Level 2 – 128ms at 1Mbit/s
6	SRT Level 3 – 2048ms at 1Mbit/s
7 (lowest)	NRT – use any available space

Table 1 – Message Priority Assignment

This bus allocation method achieves a number of goals:

- Support for HRT, SRT and NRT messages.
- Support for both event driven and periodic messages.
- Limitation of the maximum trigger rate of each message to no more than one message per unit time in order to provide bus capacity for low priority messages.
- Support for the inclusion of a “sync” message once per unit time for those nodes which require it.
- Support for fault recovery, jitter and other errors in message trigger timing.

If a message’s data payload exceeds eight bytes then it will have to be distributed across more than one CAN data frame, which, depending upon the nature of the data, can be handled in one of two ways. In order to guarantee delivery performance, time-critical or safety-critical data will normally be transmitted as a group of single frame messages each with unique function identifiers. If, however, the data is not critical, then it can be transmitted by means of a number of linked data frames. This is termed a multi-frame message. A dedicated handler is required in order to manage individual frames within a multi-frame message and issue them to the data link layer. The mechanism makes use of the normal frame format but uses the first data byte of the payload to pass a code used to guarantee the chronology of the data. At the start of a multi-frame message this is set to 0, incrementing up to 249 with each succeeding frame. At 249 the leading data byte value rolls over to 1. A value of 250 is reserved to indicate the end of a multi-frame message regardless of the number of individual frames.

Military vetronic architectures are normally comprised of multiple distributed real-time sub-systems, and as a result the communication protocol employed must support both determinism and co-ordination. MilCAN achieves this by employing one of the nodes as a Sync Frame message generator (or Sync Master). It is this sync generator which broadcasts a “sync frame” every PTU in order to provide a means by which nodes can co-ordinate actions.

Resilience is prerequisite to any military communications architecture, and since a single Sync Frame message generator makes the system vulnerable to its failure, MilCAN allows other nodes to assume this role in the event of a failure of the current Sync Master.

The generation of Sync Frame messages is the responsibility of the node that has won the arbitration for the role of Sync Master that takes place at system start up and in the event of a failure in the currently elected Sync Master. The arbitration process ensures that the potential Sync Master with the highest priority becomes the system Sync Master. If the Sync Master with the highest priority is non-functional then the next highest priority potential Sync Master will become the system Sync Master.

PBISA

A crucial requirement of the PBISA system was the capability for future growth, since the CR2 has a service life of over 25 years, it will potentially have to be upgraded with new technologies/features during its life to maximise the use of emerging technologies and software developments.

The CR2 vehicle architecture was originally designed with growth in mind and as such already employs the Mil-Std 1553B dual redundant data bus, however, a more flexible and cheaper option was required.

With CANbus already well established in both the commercial automotive world and the process industry, it is now gaining widespread support in the defence community. Its robustness, reliability, cost effectiveness, and the large following from the semiconductor industry are some of the benefits offered by CAN bus networks. The need for deterministic communication between certain devices on the bus was to be the overriding factor in the selection of the CANbus solution. Furthermore, a solution that could support both cyclic and event driven messages would enable high communication performance to be achieved at relatively low baud rates by reducing the bus loading to a minimum.

CANopen provides a mechanism that allows some level of determinism to message transfer. Permitting devices to operate either asynchronously or synchronously to the bus. However, synchronisation of devices works on a master/slave basis, where one node is allocated as a permanent sync master. The sync messages generated in CANopen are dummy messages that contain zero data in the payload, e.g. no sync message count. CANopen therefore only supports one master node and does not provide a mechanism for election of a new master in the event of a failure. In addition, CANopen is based on the 11-bit frame identifier and as such imposes significant restrictions on systems requiring large message sets.

The J1939 standard is based on the 29-bit frame identifier however it is not deterministic.

MilCAN A enhances the synchronisation capability provided by CANopen by operating on a multi-master basis. With MilCAN A, multiple master nodes arbitrate for the right to generate the master sync message. This method provides a level of redundancy to the system that ensures that a new master is elected after the failure of the current sync master is detected. In addition to this, MilCAN A further time-slices the sync message by providing a count in the data payload that identifies the sync slot number (range 0 to 1023). This mechanism facilitates the scheduling of message transfers.

The PBISA system is a distributed real-time system that requires certain messages to be delivered between devices with predictable performance, e.g. own vehicle position (every 500ms), alive status (every 1s). To guarantee message delivery to meet maximum latency requirements the selected solution had to provide some mechanism for deterministic communication. Since MilCAN was an Open protocol and based on standard, low cost CAN hardware, it was investigated and proposed as the medium to be used to enable the PBISA solution.

The architecture of the system was strongly influenced by the physical characteristics of the vehicle. The system required Line Replaceable Units (LRUs) to be located in both the turret and the hull, primarily due to the fact that information had to be passed between the Commander's and Driver's stations. Although the severe space constraints within the turret also led to this decision.

The PBISA system shown in Figure 4 features two computing platforms working together to optimise the integration into the vehicle. The PBISA Processor Unit (PBPU) provides moving map and situational awareness data to the vehicle Commander, whilst the PBISA Digitisation Processor Unit (PDPU) processes navigation system and weapon system information in real-time and passes this data to the PBPU. It provides co-ordinate conversion calculations for tasks such as far target locations, assisted lay, automatic waypoint switching and other vehicle specific functions. The PDPU also acts as a firewall between PBISA and the real-time Challenger 2 weapons system.

The PDPU allows accurate positional and heading data to be combined with weapon system sensor information and passed over the MilCAN bus. This data is then displayed at the Commander's display showing information from both the PBPU and the PDPU battle management data and vehicle level systems data.

The Inertial Reference Unit (IRU) passes navigation data to the Driver's display. The driver uses this information as an aide to reaching the designated waypoint with minimum intervention from the vehicle Commander.

The Driver's display depicts the current waypoint number, own vehicle position (as a 6 figure grid reference), distance to waypoint in Km, bearing to waypoint (relative to North) and 'steer to' as a number of arrows graduated to appear left or right depending on the amount of steer required.

To allow the PBISA system to be utilised whilst static and on the move, a number of Man Machine Interface (MMI) mechanisms are available i.e. a pointing device, 20 fixed function keys and a stowed keyboard to enter free text whilst the vehicle is stationary.

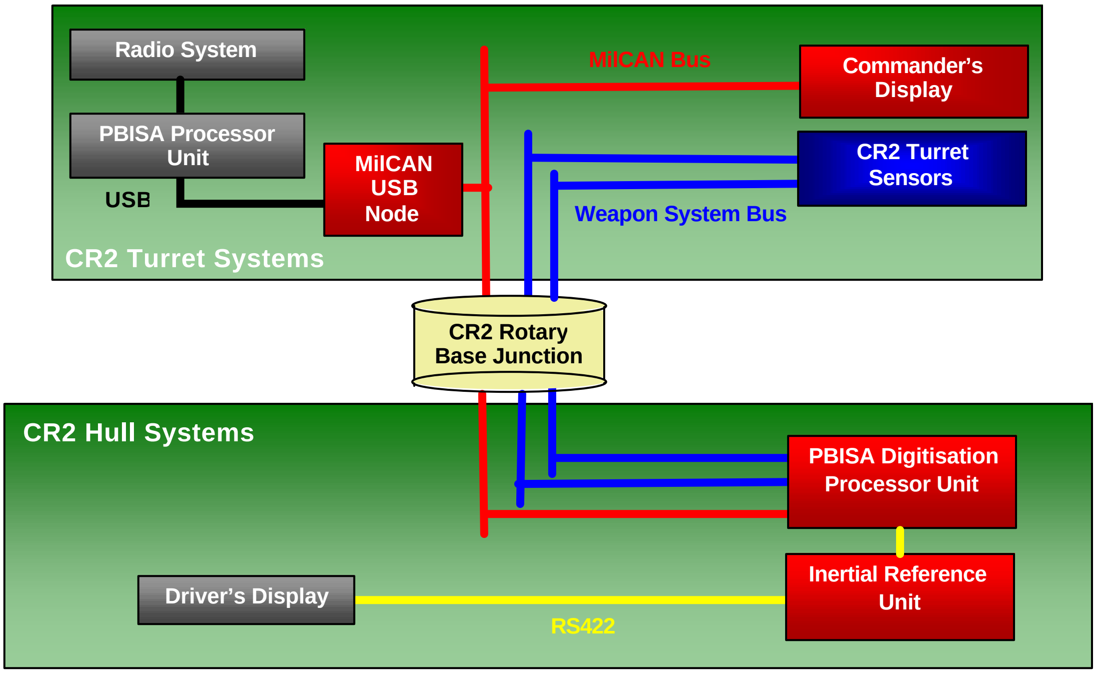

Figure 4 - Platform Battlefield Information System Application (PBISA)

PBISA Digitisation Bus Protocol

The digitisation bus is based on the MilCAN A protocol and supports both periodic and event driven messages. If messages are inherently point-to-point they are still treated as broadcast messages with a single recipient. For event driven messages requiring acknowledgement, this is achieved in the application layer either directly or by inference.

Synchronous messages are used wherever possible as this provides low latency low loading risk for the bus. Large event driven data messages were avoided, however, where they were necessary, e.g. waypoint upload/ download, attention was given to prevent short-term bus overload. Some typical PBISA messages are identified in Table 2.

All devices connected to the bus transmit an Alive message to enable the system to determine the respective node’s functional state i.e. healthy/unhealthy.

The bus operates at 250kbps to allow compatibility with J1939 nodes.

PBISA Modes of operation

When powered on, devices connected to the PBISA bus can be in either one of three modes of operation as illustrated in Figure 5.

Pre-Operational Mode – Following the application of power, a reset, loss of Sync Frames or upon exiting the system configuration mode, devices connected to the PBISA bus automatically enter this mode. In this mode of operation message transmission is restricted to Sync Frame and enter/exit configuration mode messages only. Having entered pre-operational mode, devices designated as potential Sync Masters arbitrate for the role of Sync Master.

Once devices have entered pre-operational mode, they remain in this mode until they have received a valid Sync Frame message or a valid enter system configuration mode message sequence. Following receipt of a valid Sync Frame, all devices enter the operational mode and start normal message transmission. Following receipt of a valid enter system configuration mode message sequence, all devices enter the system configuration mode.

Operational Mode – Devices suspend pre-operational mode and enter operational mode following receipt of a valid Sync Frame. In operational mode, normal message transmission is allowed.

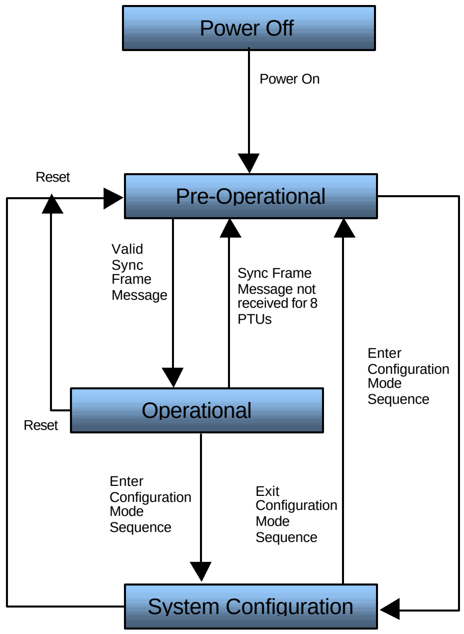

Figure 5 – Modes of Operation

All devices suspend operational mode and enter pre-operational mode following a reset or in the event that Sync Frame messages cease to be generated on the PBISA bus within 8 PTUs of the last received Sync Frame.

System Configuration Mode – All devices suspend pre-operational mode and enter system configuration mode on request by a Configuration Master Node. Whilst in system configuration mode, devices only respond to system configuration mode messages.

System configuration mode is invoked by the Configuration Master Node, attached to the bus, transmitting a sequence of three messages with the same ID but with a different defined payload for each message. On receipt of the correct sequence of messages all devices attached to the bus suspend operational mode and enter system configuration mode.

In system configuration mode the Configuration Master Node continuously transmits the enter system configuration mode message sequence at the system Primary Time Unit (PTU) rate. Ensuring that any devices that come on-line when the bus is in system configuration mode also enter into system configuration mode.

Design and Implementation

The PBISA software was written in Tornado C and ran under the VxWorks Real Time Operating System (RTOS).

A structured design methodology was adopted using Select Yourdon. Detailed functional models of the system were developed defining data flow diagrams and state transitional diagrams to model dynamic behaviour.

CAN bus operation was monitored using the CANoe CAN bus development software.

During development, software based emulators were used to simulate the operation of the PBISA systems and subsystems.

Results

Figure 6 shows the loading of the bus over time. When the system is first initialised the bus loading is relatively high (20%), after five seconds, it goes down to a 9% average.

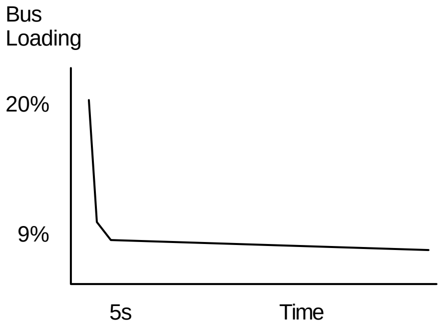

Figure 6 – PBISA Bus Loading

Message	Content	Message Group	Type	Nominal Freq.	Bytes
Sync Frame	Sync counter	System Management	Periodic	15.625ms	2
Target Location Report	Target location (range source, target latitude, longitude and altitude, raw range and bearing)	STA	Event	-	15
Commanders Primary Sight Field of View	Magnification and orientation of Commanders primary sight	STA	Periodic	500ms	3
PBISA Assisted Lay	Target location to enable alignment of Commanders primary sight (target latitude, longitude and altitude)	STA	Event	-	11
Own Vehicle Latitude and Longitude	Own vehicle position (latitude and longitude)	Navigation	Periodic	500ms	8
Route Download	Download route with a maximum of 20 waypoints (route number, route name, number of waypoints in route, current waypoint number, waypoint data (number, name, grid position, proximity boundary))	Navigation	Event	-	664 (max)
Ammunition Status	CR2 ammunition state	Comms/BMS	Periodic	500ms	7
PBISA ‘Alive’ Status	P-BISA healthy/not healthy status	Diagnostics	Periodic	1s	1
CCS Display Key Status	CCS display mode and key state	Generic MMI	Periodic	62.5ms	7

Table 2 – Typical PBISA Messages

Conclusions

The PBISA system has been the subject of rigorous user trials carried out by the British Army. As a result the system has proven itself as a powerful tool to the user, greatly enhancing the Commander's situational awareness and easing the critical command and control decisions with a greatly increased up to date battlefield knowledge base.

Three hundred and thirty six of the British Army's fleet of CR2s are being integrated with PBISA, easing the British Army's transition towards the digitised battlefield.

Commercial Off-The-Shelf (COTS) and Military Off-The-Shelf (MOTS) units have been utilised extensively by the PBISA system, minimising the high costs associated with bespoke hardware.

Since a MilCAN network now resides on the CR2, it is possible to accommodate future vehicle systems integration with minimum integration effort and cost.

The automotive control system for CR2 has recently been replaced with one that is CAN enabled. Thus providing further opportunity to share some of the data over the vehicle network.

Future aspirations for the Cr2 are to embody a Health and Usage Monitoring System (HUMS). This would interface to the vehicle Electronics Control Unit (ECU), sensors and other units, to measure and record the vehicle usage and operational state.

References

1. MilCAN Specifications, http://www.MilCAN.org/

2. Steven T. Majoewsky, Colin Davies, "MilCAN: Adapting COTS CANbus to Military Vetronics", 8th International CAN Conference, 2002, USA.

Abdul Qabaz
BAE Systems Land Systems
Leeds Valley Park
Leeds
United Kingdom
LS10 1AB

Tel: +44 (0) 113 2724290
Fax: +44 (0) 113 2724201
Abdul.Qabaz@BAESystems.com
www.BAESystems.com