TECHNIQUES FOR ACQUIRING DIGITAL DATA FROM INTELLIGENT SUBSYSTEMS

Harold B. Gilje
Vice President, Systems
Aydin Vector Division
Friends Lane
Newtown, Pennsylvania 18940

Raymond F. Nicolais
Senior Engineer, Digital Products
Aydin Vector Division
Friends Lane
Newtown, Pennsylvania 18940

ABSTRACT

There is an ever-increasing need for intelligent sub-systems onboard airborne and ground vehicles. With this intelligence comes asynchronous bus communications under the control of various industry standards and specifications. As a result, two telecommunications needs have developed; from a telemetering viewpoint, that of acquiring data from the various intelligent sources and time-division multiplexing that data with classic analog instrumentation, and, from an intravehicle communications viewpoint, that of providing a means for transfer of information between two dissimilar sub-systems. A systematic breakdown of the necessary elements to link, synchronize, sort, interpret, store, merge and control such data is examined.

Proven approaches to handling data from both standard (ARINC, MIL-STD-1553, IEEE-488, etc.) and specialized subsystems is overviewed from an instrumentation perspective.

INTRODUCTION

The electronic intelligence provided by today's advanced technologies is a welcomed and necessary step in achieving the growth demanded by the evolution of vehicle requirements. As various electronic subsystems become more intelligent, the need for higher speed, more complex, intrasystem communication becomes more evident. Based on system and subsystem parameters, equipment designers have developed communications protocols for this interactive transfer of data between subsystem components. Such protocols are selected based upon parameters such as volume of data transferred, operational speed of the subsystems, number of subsystem equipments involved, etcetera. To simplify the designers task, monitoring organizations such as ARINC, IEEE, EIA, CCITT and the military have all developed "standards" which well define the rules whereby the communications process will take place. Regretfully, each of these standards are unique to

themselves, as they have each been designed for specific requirements and/or applications. To further complicate the issue, numerous non-standard protocols exist, being designed for custom interface with very specific operational equipment.

As a result of all this evolution, we have been exposed to two requirements which will be discussed herein. First, the monitoring of the communicated data for telemetric, data gathering and, second, the requirement for transfer of information between subsystem equipments that do not have compatible communications protocols.

THE TELEMETRY PROBLEM

Digital telemetry transmission is governed by IRIG Standard 106-80. The format, i.e., word and frame structure, is well defined by this standard and, as one would predict, is not compatible with the above mentioned protocols. Additionally, the rates of data transfer are not only unequal, but are also generally non-synchronous.

To provide for transfer of information from the communications bus into the telemetry network, requires a multi-functioned interface. This interface must first handle the input/output protocol (sometimes handled by available chip sets), read and/or respond to addressing and subaddressing, acquire and store the data intended for telemetry transmission and, then, output the data to the telemetry system in a format compatible with the PCM system. Although dedicated hardware can be designed to perform these functions, it is often times advantageous to accomplish these tasks using general hardware interfaces under control of a microprocessor.

The addition of a microprocessor provides a secondary capability, that of data processing beyond simple I/O reformatting. We have now developed a terminal capable of interpretation, selective data transfer and data compression.

THE INTRACOMMUNICATIONS PROBLEM

As an extension of that addressed above, a similar microprocessor controlled device can provide interactive data transfer between two or more communications busses. Typical of such a situation would involve an airborne vehicle with a 1553 bus. How could one interface a small computer with only RS-232 I/O capability? How would one communicate with ruggedized commercial test equipment designed only to converse via an IEEE-488 bus? Because of the speed and intelligence required by these interfaces, a microprocessor controlled translator is required to interface both busses and provide the interaction necessary to intelligently converse from original sender to ultimate receiver.

THE INTERFACES

The repertoire of interfaces available to the systems designer are depicted in Table I. The MIL-STD-1553 and IEEE-488 both provide moderately high speed data transfers and are well defined (yet flexible) interfaces. ARINC and EIA RS-232/422 serial interfaces are very mature and their speed is considerably slower than 1553 or 488. Non-standard interfaces may be found on specific applications such as computers for guidance or flight control.

MIL-STD-1553B is a I MBPS serial command/response bus which has become a standard for high performance military aircraft. A bus controller initiates all transactions (messages) by issuing commands to remote terminals. Well defined message formats and extensive use of status words provide a highly reliabile link, ideal for avionics or stores management. The standard defines the monitor function as a receive only, information extraction device. The monitor serves as a transparent “wire tap” on the bus to allow the bus controller to perform its real time tasks unencumbered by the data acquisition needs.

IEEE-488 is a parallel bus traditionally used for lab instrument communication. As with 1553, it utilizes a central bus controller which commands all message transactions. The eight-bit data bus is supported by a reliable, three-wire handshake and five management lines to allow remote/local control, switchover and service request interupts. The transfer rate is not fixed, but rather adapts to the needs of each remote unit. While typically much slower than the 1553, selected interfaces do provide rates greater than 1553 in limited applications. Major MPU manufactures are offering interface chips with military temperature ranges. As a result, the IEEE-488 bus is finding airborne applications due to its low cost.

Both ARINC and EIA serial interfaces may be found in military aircraft. The ARINC serial bus is a broadcast only interface with each transmitter maintaining its own bus. Embedded in each 32-bit word is an eight-bit label which defines the data content. EIA RS-232 and RS-422, while very mature and relatively slow, still provide reliable linking between computers and peripherals. It remains the most common interface between manned terminals and computers.

Non-Standard interfaces are typically found on specific applications where guidance of flight control computers cannot afford (technically or financially) the impact of supporting the high speed bus controller overhead (hardware and software) needed by 1553 or 488.

THE HARDWARE SOLUTIONS

In response to the need for interface hardware, component manufacturers have developed "chip sets" for each of the standard busses. Each chip set approximates the functions found in the classical UART (Universal Asynchronous Receiver-Transmitter). Linking to 1553 requires transformer coupling, a 1553 transceiver and a 1553 encoder/decoder. These devices have typically been hybrid modules. Since most 1553 applications also require redundant interfaces, both cost and part count are relatively high. Recent introductions of LSI devices should lower both cost and part count.

Linking to IEEE-488 typically requires two octal transceivers and an interface chip. Most MPU manufacturers offer interface chips which directly interconnect with their MPU products. Exclusive of software development costs, an entire intelligent 488 interface is less costly than a 1553 interface.

ARINC interfacing is via a single chip containing both transmitter and receiver. Both Harris and Western digital provide ARINC interface devices. RS232/422 interfacing is accomplished with a UART and, optionally, a programmable rate generator for full flexibility. Non-Standard interfaces may use either differential receivers, single ended receivers, or simple logic buffers depending on cable length and noise considerations.

Bit and word synchronization are fully supported on the standard interface chip sets. Non-Standard interfacing, however, requires the selection of one of three possible approaches. The first approach involves asynchronous strobing into data buffers within a PCM encoder. The approach requires an intensive liason between the system designer, the encoder manufacturer, and the computer's hardware and software designers to insure a reliable interface. The approach may incur development costs for the encoder's manufacturer, but should unburden the computer development task. The second approach involves slaving the computer interface to the PCM. Most encoders provide a standard option which flags format time slots for digital data sampling. This eliminates the need for data buffering within the encoder and allows virtually off-the-shelf encoders to be utilized.

A third approach, that of extracting clock signals, is a variant on the first approach (strobing). Bit clock may be extracted from Bi-phase, data-bit encoding. Word clocks are recoverable from a flywheel counter (synchronized by a frame strobe) or by gap times between data words. The approach minimizes signal lines between units and the intensive liasons required to insure a reliable interface.

Message/frame synchronization, data sorting, and data storage are best overviewed as a set of interactive elements for each interface. Monitoring MIL-STD-1553 presents a unique set of challenges. Bus addressing permits up to 31 terminal addresses times 30

subaddresses at rates, when merged with the classic analog instrumentation, could readily overload a telemetry link. The intelligence of the Bus Controller to dynamically adapt to mission needs virtually prevents a monitor design which depends on cyclic frames of data.

The concept adapted by most bus monitor manufacturers is that of a highly specialized type of logic analyzer. Data is acquired from the bus on a message-by-message basis. A random access “Trigger List” in PROM, checks the command word at the start of each message and stores the selected message words in RAM. The PCM encoder may access the RAM during any word time. The PROM also partitions the RAM so that each word of each discernable message has its own unique RAM address. Enhancements such as the time tagging of data and flag bits for data “staleness” and “overflow” have become standard features of bus monitors.

The IEEE-488 bus provides an opportunity for cost sensitive projects to not only acquire data, but also provide interactive communications between telemetry and aircraft systems. While still a command/response bus like 1553, its informal message protocol allows messages of any size. Controllers may dynamically adapt messages for telemetering, and telemetry may report to the controller on selected parameters collected by the encoder. This is all readily accomplishable since the interface chips available are designed as I/O devices to support a product line of peripherals for or by the MPU manufacturers.

Management of multiple buffers by the MPU permits coherent messages over telemetry. The MPU may simply output data to a second port connected to an encoder digital input. Speeds comparable to 1553 are achievable with the addition of a DMA controller to the processor’s architecture. The features discussed are not intended to replace MIL-STD-1553, but to provide an opportunity to take advantage of a standard bus in cost sensitive projects.

ARINC data acquisition provides a ready means of sorting data. Each data word (32 bits) contains a unique eight-bit label which may be used for addressing multi-port RAM memory during write cycles. The encoder may address the data in memory by label for insertion into the PCM stream via a digital input port. Multiple ARINC interfaces may share the same memory since units are specified by unique labels as to their data content.

Data from EIA Rs-232/422 interfaces may be buffered by First-In First-Out (FIFO) memory.

Non-Standard interfaces require a careful selection of the buffering technique. Simple FIFO buffering may be applied if the input data rate is lower than the encoder’s sampling rate. The source should supply its own frame synchronization code, and the ground station needs the intelligence to provide software synchronization. Should this not be practical,

then two FIFO arrays may be ping-ponged at the encoders frame rate. This requires that the source and encoder maintain identical frame times and utilize a common clock source.

For large amounts of data, or when data is a cyclic “refresh” data, Random Access Memory (RAM) should be used. Dual port RAM with two address generators (counters) can emulate large FIFO arrays. The cyclic refresh also utilizes two address generators, one slaved to the source of data and the other to the encoder. This approach is simple to implement in cases where time correlation and message coherence is not necessary.

IMPLEMENTING THE INTELLIGENT PROCESSOR

Functionally, it would be advantageous to implement the multibus processor in a stacked, building block concept, such that the unit could be quickly configured to suit the particular application. This is achievable, providing the equipment is configured in blocks as indicated in Figure 1. The basic building block set would be the MPU with its random-accessed and read-only memories, implemented with its direct-memory-access (DMA) controller and DMA support circuitry. One could then configure a shopping list of interface circuits including IEEE-488, MIL-STD-1553B, RS-232, RS-422, ARINC, and so on. Custom interface designs would be simply added to the internal processor bus as required.

The limitation to such customization is two-fold. Each implementation, by nature of redefinition of I/O parameters, requires new firmware. By proper structuring (modularity) of the program, this non-recurring effort should be minimized. The second, and more important limitation, is time. All the data transfer is achieved via the MPU or its DMA controller. In either case, data transfer uses up “clock cycles”. Additionally, any message initialization and termination, reading of addresses or subaddresses, interpretation of data, etcetera, requires substantial routines which take time. Typically, messages are received and stored (in RAM) before retransmission, often times with triple buffering to insure message continuity. Transferring in and out of memory takes time. Each implementation or configuration must be considered individually with a time study, to insure that under worst-case conditions, the program is capable of keeping up with the data transfer requirements.

THE SP-900 FAMILY

The SP-900 family is one of a series of intelligent processing equipments developed by Aydin Vector. It was specifically designed as a solution to the system problems addressed above. The heart of the SP-900 family is a powerful Motorola 6809 microprocessor fully integrated with its programmed ROM, high-speed RAM and dedicated DMA controller. The DMA controller is programmed on a cycle-steal basis, so as to impact processor operation only on an “as required” basis during data transfer.

The SP-900 is implemented as a stacked array of printed circuit boards, as shown in the photograph in Figure 2. There are 5 boards in this particular implementation; functionally, from top to bottom:

- IEEE-488 Interface
- Microprocessor With DMA
- PCM System Interface
- Up-Link Decoder
- Power Supply

As an advantage of the parallel-bus structure, other functional boards can be integrated to provide a customized solution at relatively little cost. Also, various versions of similar configurations can be supplied.

As an additional feature, the ROM program can be replaced with a small, ROM-contained, program-load routine. In this configuration, the SP-900 RAM can be loaded from a remote processor, tape or disk, so that the user can software-reconfigure the system parameters before an exercise.

THE PBM-1553 PROGRAMMABLE BUS MONITOR

The Programmable Data Bus Monitor, PBM-1553, is a versatile airborne unit that provides access to selected information from a MIL-STD-1553 data bus for input to the onboard flight test system. The PBM-1553, pictured in Figure 3, will interface with a dual redundant 1553 bus and operate as a monitor for data acquisitions without interference to the vehicle data bus operation. The unit is compatible with MIL-STD-1553A, MIL-STD-1553B and the 1553 systems implemented in the F-16, F/A-18, B-52 OAS, AV-8B, A-10 and other, modern military aircraft.

TABLE I. PARAMETER MATRIX FOR DIGITAL DATA ACQUISITION

INTERFACE	INTENDED APPLICATION	CONCEPT	BIT SYNCHRONIZATION	WORD SYNCHRONIZATION	MESSAGE SYNCHRONIZATION	TYPICAL MINIMUM INTERPRETATION	SORTING/ STORAGE
MIL STD 1553	MILITARY AIRCRAFT	TYPICALLY MONITOR, I/O OPTIONAL	CHIP SET	CHIP SET	COMMAND WORD COMPARES TO A LIST OF TRIGGERS	FLAG ERRORS	EXTENSIVE MULTI-LEVEL SORT RAM STORE
IEEE 488 (GPIB) (HPIB)	LABORATORY INSTRUMENTATION	TYPICALLY I/O, MONITOR OPTIONAL	N.A. (BYTE SERIAL)	CHIP SET	COMMAND WORDS INTERPRETED BY CPU AND/OR CHIP SET	FLAG ERRORS	SUB-ADDRESS SORT RAM STORE
ARINC	AIRCRAFT	RECEIVE ONLY	CHIP SET	CHIP SET	USES LABEL FIELD OF INCOMING WORDS	FLAG ERRORS	LABEL SORT RAM STORE
EIA RS-232 RS-422	COMMERCIAL COMPUTER	TYPICALLY I/O	CHIP SET	CHIP SET	OPTIONAL, DESIGNATED DATA CODES (CONTROL CHARACTERS)	FLAG ERRORS	FIFO STORE
NON-STD SERIAL	SPECIAL COMPUTER	TYPICALLY SIMPLEX	1. STROBE IN 2. SLAVED 3. EXTRACTED FROM BI-Ø DATA	1. STROBE IN 2. SLAVED 3. FLYWHEEL BIT COUNTER	1. STROBE IN 2. SLAVED 3. DATA BUFFER	A.R.	RAM/FIFO STORE
NON-STD BYTE SERIAL	SPECIAL COMPUTER	TYPICALLY SIMPLEX	N.A.	1. STROBE IN 2. SLAVED	1. STROBE IN 2. SLAVED 3. DATA BUFFER	A.R.	RAM/FIFO STORE

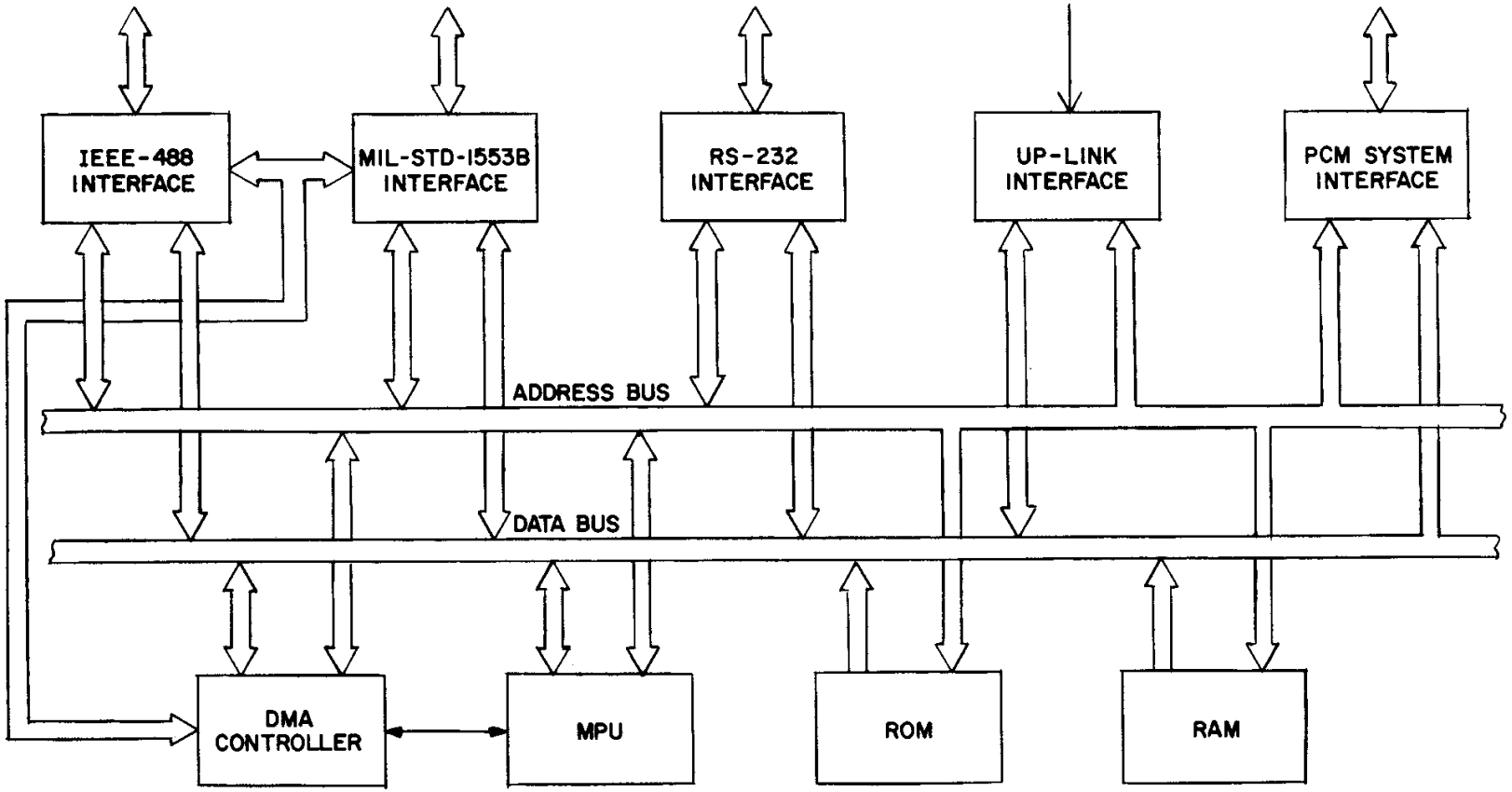

FIGURE I FUNCTIONAL BLOCK DIAGRAM

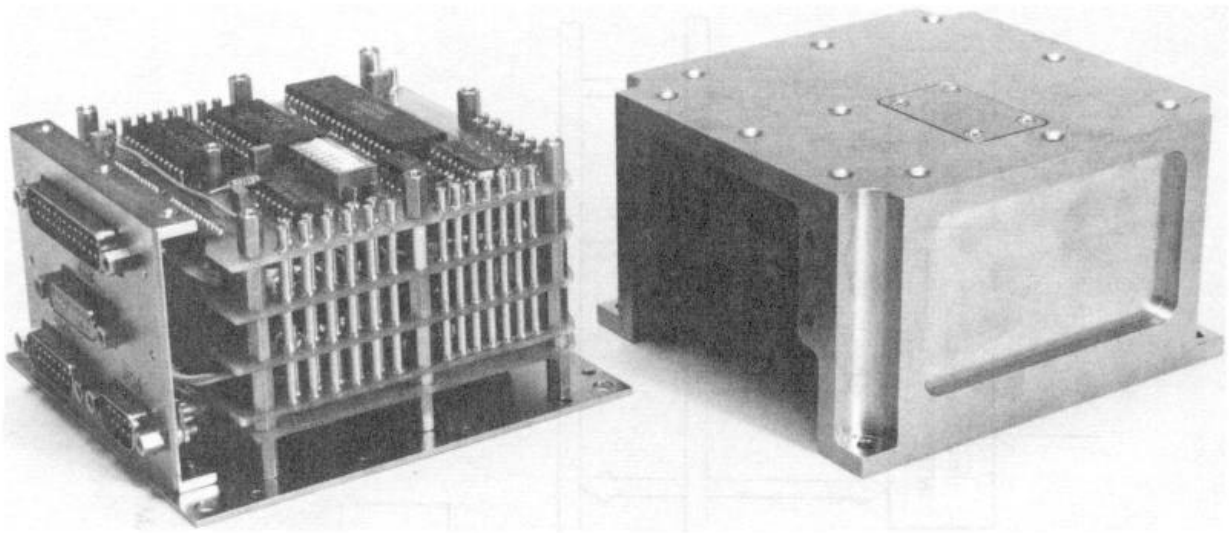

FIGURE 2. THE SP-900

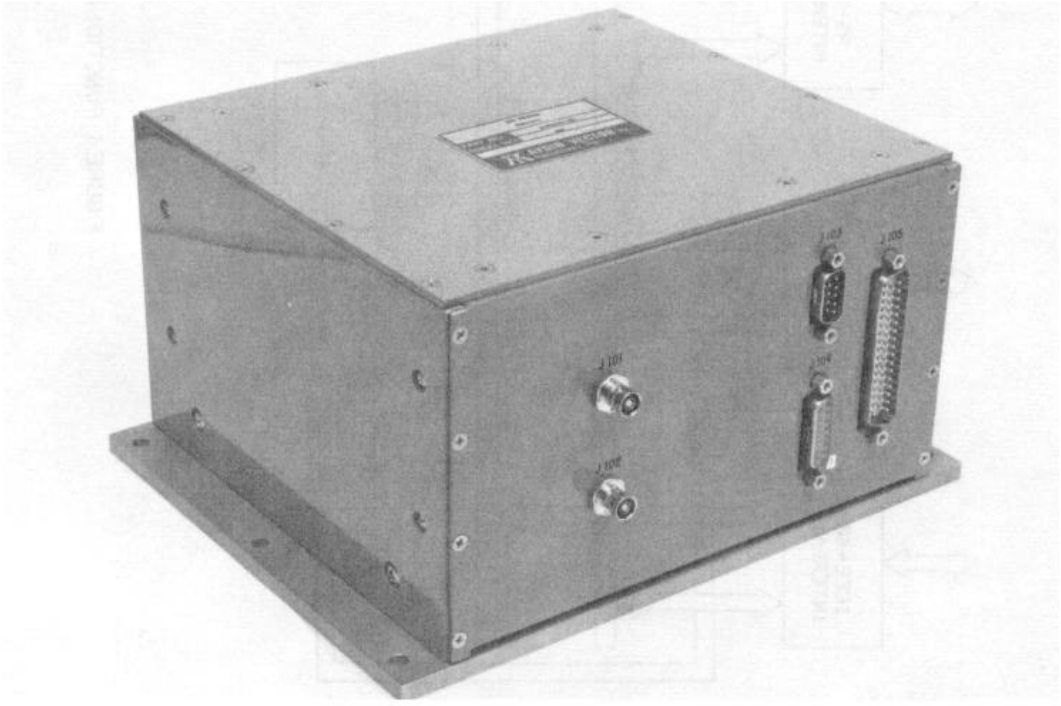

FIGURE 3. THE PBM-1553