Time-Tagging Issues Relating to Networked Data Acquisition Systems

Tara Owens, TRW
Alfredo Berard, 46TW/TSI
Tim Boolos, TRW
Common-Event Network Test-Instrumentation System (CENTS)
46th Test Wing Eglin Air Force Base/TRW

ABSTRACT

The CENTS Program is a Central Test and Evaluation Investment Program (CTEIP) effort conducted by the 46th Test Wing at Eglin Air Force Base, Florida. The project uses advanced internetworking technology to collect data unobtrusively from multiple sensors located throughout the aircraft without the time and expense of installing new wires. The sensors are used to unobtrusively extract data from several Line Replaceable Units (LRU's). The harvested data is then transported to a master network controller using the existing aircraft power lines.

A critical aspect of networked data acquisition is time-tagging the data so that the data timeline can be reconstructed to a specified resolution at the conclusion of the tests. This paper will discuss the time-tagging issues that arise when developing a networked data acquisition system, especially how they relate to the current effort to develop a power line based data acquisition network. In addition this paper will detail the scheme currently being tested to time tag data in the Common Event Network Test-Instrumentation System (CENTS) developed by the 46 TW/TSI Flight Test Division, Air Armament Center, Eglin AFB, FL.

INTRODUCTION

Over the past several years, more demands than ever have been forced upon the platforms of many DoD vehicles. The Common Event Network Test-Instrumentation System (CENTS) modernizes the core T&E infrastructure capability by providing data acquisition capability for advanced fighter aircraft where mounting volumes are severely limited or quick reaction instrumentation capability is required. The need for complex munitions platforms, more responsive avionics, and the increasing presence of modeling and simulation has pushed current data acquisition methods to near their limits. The next logical step in the evolution of instrumentation systems is to implement a network backbone to accommodate the expansive growth that will accompany these needs. Networked Data Acquisition Systems allow for overall system control of sensors and instrumentation that currently exists on a military vehicle. CENTS is an open architecture system utilizing variable length event driven data packet protocols that implements reduced spectrum modulation telemetry. The three building blocks of CENTS are the CENTS Local Area Network (CLAN), the CENTS Smart Instrumented Coupled Connectors (CSICC) and data Decoder as depicted in Figure 1. The CLAN is a "virtual network" that is implemented by superimposing data on the existing Aircraft/UAV power lines. At the lowest layer the CLAN provides for utilization of existing TCP/IP protocols and allows for remote wireless applications. CSICC's are the second building block of CENTS providing for direct interface to all "data producing" hardware. CSICC's connect directly to the existing Aircraft/UAV Input/Output (I/O) connectors and only transfer data in the event of a change of state or trip condition. CSICC's provide for built-in test capability and are automatically configured for the data type and expected values over the CLAN. The third component of the CENTS is a Decoder which is used to demodulate the superimposed data for onboard recording/telemetering.

By introducing an open-architecture approach built on a common smart transducer interface, CENTS strives to conform to the commercial standard that is widely taking hold of the sensor and data acquisition community. CENTS is an event-oriented system built on a publisher-subscriber model that monitors data from items under test via "instrumented coupled" connector interfaces. These "smart" connectors eliminate the need to add the traditional data acquisition boxes to a severely space limited air test vehicle. Figure 1 shows an example of smart sensor nodes implemented on a vehicle.

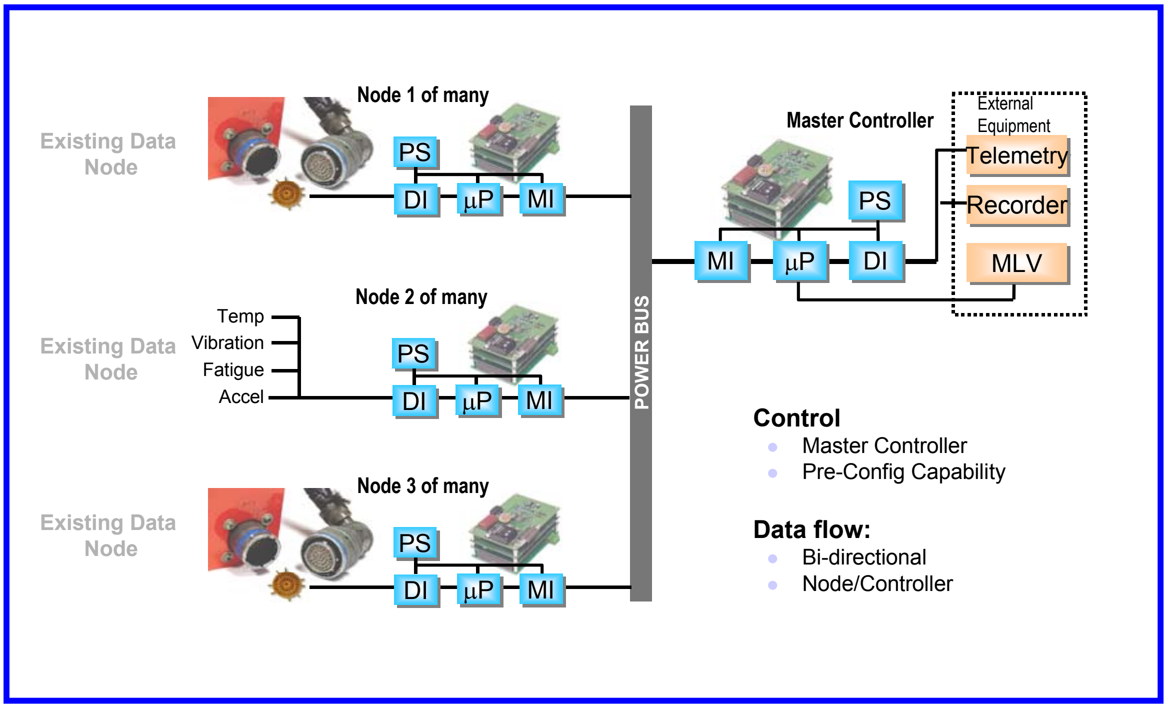


Figure 1: Networked Smart Connector Data Nodes

Given that the array of smart sensors are networked via the vehicle's power lines, there is a certain amount of latency that would accompany the transport of the data from the smart connector to the master network controller for demodulation and download into an onboard recording/telemetry device. The solution to the latency between registering the data and sending it to a master network controller is to time-tag the event as the smart connector registers it. This solution in itself poses some inherent issues.

The first issue to be addressed involves the synchronization of the smart connector to the Universal Time Code (UTC) employed by the testing equipment. Once a common network time has been established, the smart connector must have a way of determining the time between an interrupt telling the hardware that an event occurred, and the actual time that the event was tagged. Once all this data has been recorded, and the time tag with the associated interrupt lag is sent to the network master where it is decoded and attached to the data in UTC format to comply with test standards.

TIME TAGGING SCHEMA

In order for the time-tagging scheme to work, a 1 pulse per-second (pps) heartbeat is needed to keep the smart connector nodes located at the sensors in the vehicle in sync with the master controller. At the onset of a test, the master network controller broadcasts the network time, and tells all the smart connectors to synchronize to that time upon the next 1pps heartbeat. The heartbeat can be supplied by a variety of means, one being by an external single wire carrying a Global Positioning System (GPS) 1pps signal. The second method takes advantage of the frequency range outside of the Orthogonal Frequency Division Multiplexing (OFDM) carrier signals being used to transport the captured data from the smart connector to the master controller. In this instance, a simple oscillating circuit with time-interval reset circuitry can be filtered through a band-pass filter to create a heartbeat that can be registered by the smart connector.

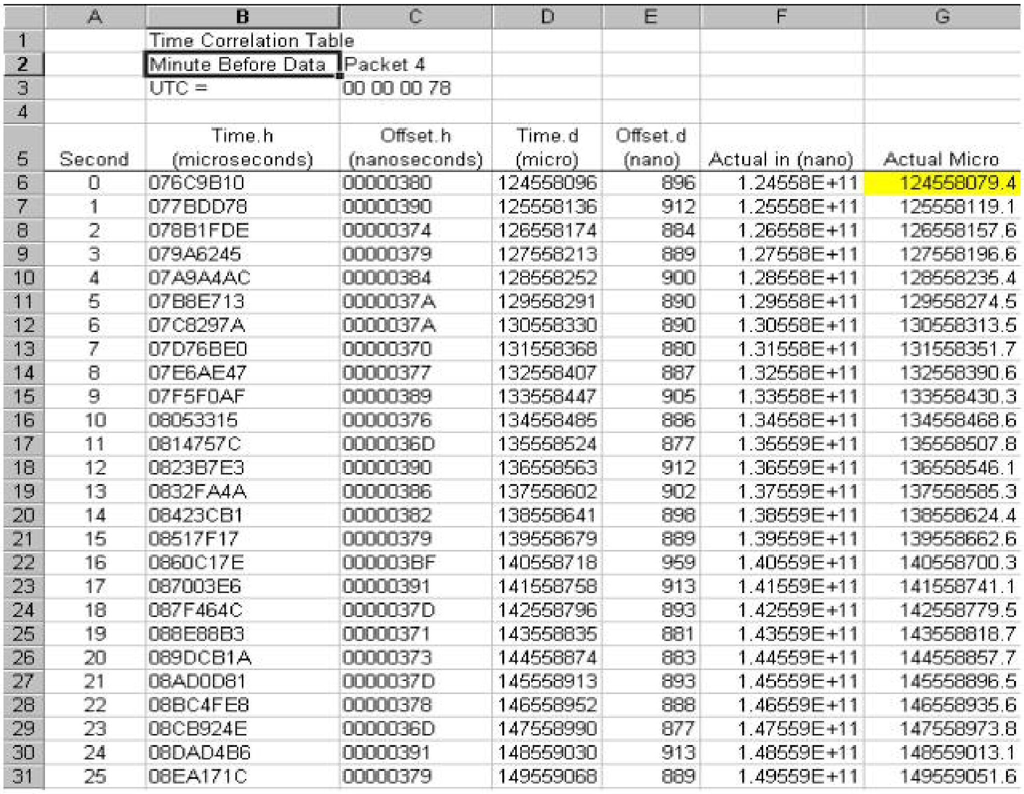
At this point, all the smart connectors will have established a reference point on the free-running internal counter that allows them to determine time from that synchronization pulse. This technique does not require that the crystal in the smart connector is accurate, just stable. By choosing this route for time-tagging issues, the system becomes more robust and less intolerant of changes in the surrounding environment, which ideally will help to determine the system's flightworthiness.

The internal free-running counters can be chosen to give any desired resolution depending on the application. This counter will receive an interrupt each second to write its count to a microprocessor also located within the smart connector. The microprocessor, which also has a free-running counter, is able to track the latency between when the interrupt was given and when the count was received. This count (usually in nanoseconds) is attached to the end of the free-running counter data. All this information is stored within the RAM of the smart connector to be sent at pre-determined time intervals to the network master as a correlation table for data that is received with a time-tag.

DATA CORRELATION

In order to differentiate between smart connector nodes on the network, a preamble is used to designate each node and to allow for recognition of the data type. Data types that are currently being monitored by smart sensors under the CENTS contract are: analog sensors, discrete sensors, and digital data bus sensors. The message sets from each smart connector node each have a header that the master controller can decode to determine the message type and origin. Each packet of data from any single smart connector is numbered, and thus can be sequentially reconstructed in the event of collisions on the power line or if priority had been given to another node at that time.

The master receives a time-correlation table from a smart connector at some pre-determined time on a cyclical basis. For example, if the smart connector sends a data correlation table at the end of each minute, that table will contain the readings of the smart connector free running timer each second. The correlation table will resemble Figure 2:



	A	B	C	D	E	F	G
1		Time Correlation Table					
2		Minute Before Data	Packet 4				
3		UTC =	00 00 00 78				
4							
5	Second	Time.h (microseconds)	Offset.h (nanoseconds)	Time.d (micro)	Offset.d (nano)	Actual in (nano)	Actual Micro
6	0	076C9B10	00000380	124558096	896	1.24558E+11	124558079.4
7	1	077BDD78	00000390	125558136	912	1.25558E+11	125558119.1
8	2	078B1FDE	00000374	126558174	884	1.26558E+11	126558157.6
9	3	079A6245	00000379	127558213	889	1.27558E+11	127558196.6
10	4	07A9A4AC	00000384	128558252	900	1.28558E+11	128558235.4
11	5	07B8E713	0000037A	129558291	890	1.29558E+11	129558274.5
12	6	07C8297A	0000037A	130558330	890	1.30558E+11	130558313.5
13	7	07D76BE0	00000370	131558368	880	1.31558E+11	131558351.7
14	8	07E6AE47	00000377	132558407	887	1.32558E+11	132558390.6
15	9	07F5F0AF	00000389	133558447	905	1.33558E+11	133558430.3
16	10	08053315	00000376	134558485	886	1.34558E+11	134558468.6
17	11	0814757C	0000036D	135558524	877	1.35559E+11	135558507.8
18	12	0823B7E3	00000390	136558563	912	1.36559E+11	136558546.1
19	13	0832FA4A	00000386	137558602	902	1.37559E+11	137558585.3
20	14	08423CB1	00000382	138558641	898	1.38559E+11	138558624.4
21	15	08517F17	00000379	139558679	889	1.39559E+11	139558662.6
22	16	0860C17E	000003BF	140558718	959	1.40559E+11	140558700.3
23	17	087003E6	00000391	141558758	913	1.41559E+11	141558741.1
24	18	087F464C	0000037D	142558796	893	1.42559E+11	142558779.5
25	19	088E88B3	00000371	143558835	881	1.43559E+11	143558818.7
26	20	089DCB1A	00000373	144558874	883	1.44559E+11	144558857.7
27	21	08AD0D81	0000037D	145558913	893	1.45559E+11	145558896.5
28	22	08BC4FE8	00000378	146558952	888	1.46559E+11	146558935.6
29	23	08CB924E	0000036D	147558990	877	1.47559E+11	147558973.8
30	24	08DAD4B6	00000391	148559030	913	1.48559E+11	148559013.1
31	25	08EA171C	00000379	149559068	889	1.49559E+11	149559051.6

Figure 2: Example Data Correlation Table

The microprocessor in the smart connector will read the 32-bit free-running timer each second and that count is represented in Column B in hexadecimal. Using a simple Hex-to-Dec conversion, the actual time in microseconds is shown in Column D.

The number in Column C is the offset reading in hexadecimal that the microprocessor's internal free-running counter registered while waiting for the response to the 1pps interrupts to the free-running counter. In the example in Figure 2, the oscillator for the offset counter was running at 54MHz, thus giving the counter value of 18.5ns per clock cycle. Once the time has been corrected for the interrupt delay, an accurate time correlation table can be constructed.

Each data packet received by the master controller contains the event-driven data captured by the smart connector. The UTC time at which the data was captured by the smart connector can be derived from a simple interpolative process at the master controller.

Figure 3 below shows the variance between time tag readings in microseconds. This was an important statistic to observe since it would determine the needs of the timing circuitry eventually utilized in the master controller of the networked data acquisition system.

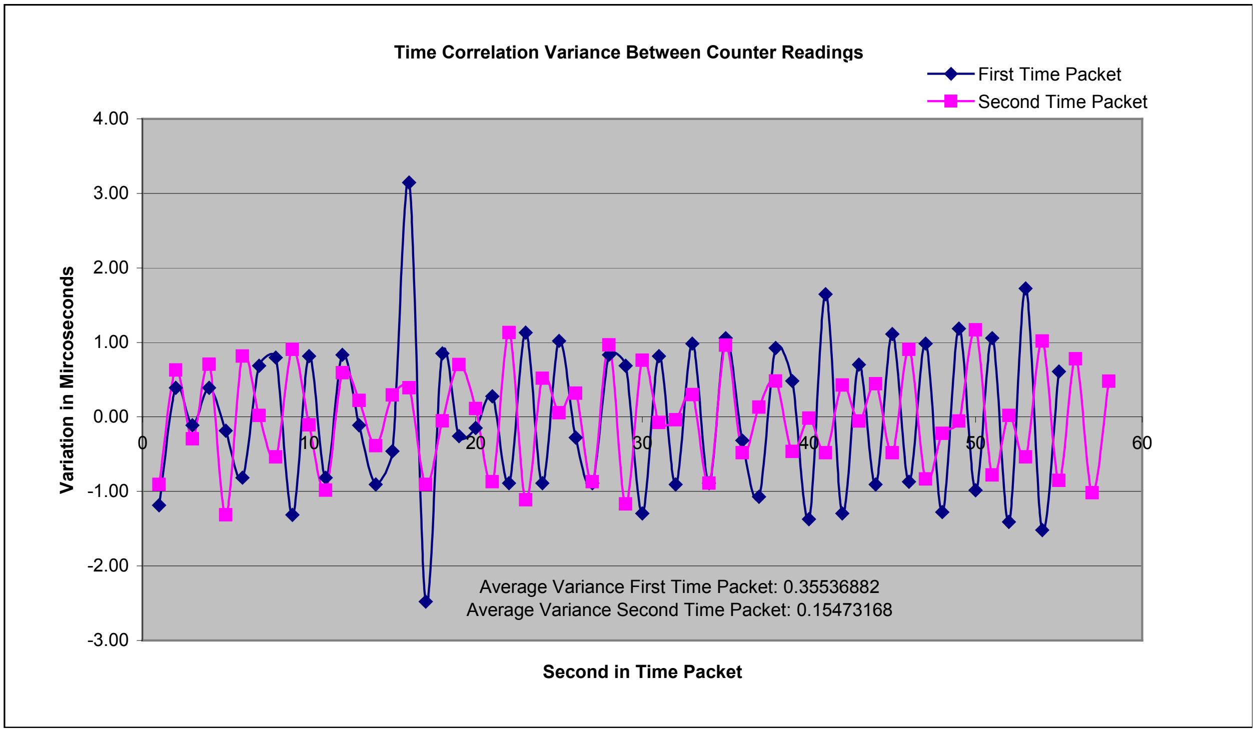


Figure 3: Variance Between Time Correlation Data Points

The oscillatory nature of the chart in Figure 3 had been expected. In the test environment in which this data was captured, a real-time 1pps generator was not used. Instead, a timing circuit that utilized a temperature-sensitive crystal had been constructed to determine if the heartbeat circuitry to be used in the final network configuration would be temperature dependent. By observing the trend in the variance between time correlation table entries, it was determined that a temperature-stabilized crystal would be needed to accurately depict the real-time sync pulse.

Once an accurate time correlation table is constructed and stored in the master controller, all data that was received from that particular smart sensor with a time tag during the previous minute can be reconstructed with a UTC time tag. This process is repeated every minute and the time correlation table updated.

Since each smart connector will have a free-running counter, it is required that each one construct and send a time correlation table to the master along with an address that distinguishes that smart connector from the other data correlation tables arriving at the master. For this reason, a TCP/IP or similar protocol can be utilized.

CONCLUSION

The 46TW at Eglin AFB is currently adapting networked data acquisition systems to the power lines of test aircraft. In order to achieve this, engineers have developed the time correlation schema described in this paper. Current and future integration testing will yield further data on achievable accuracies and required resolutions.