Eurogam Project: DT32-bus Specification

J. Alexander

Version 2 April 1991

Version 2 updates version 1 by the addition of the DTSTOP and PAUSE signals, as agreed at the Eurogam Meeting in Strasbourg, 17 October 1990; and by the addition of the Eurogam event format agreed at the same meeting.

Introduction

The DT32-bus is a cable bus, used in Eurogam to transmit data from the VXI readout controllers to the event builder. Other data sources such as readout controllers in NIM, VME or Camac crates can be added when it becomes necessary to read data from ancillary detectors which use these standards for their data acquisition. Other units, such as the Eurogam histogrammer, can spy on the data passing on the bus; but these usually take no part in controlling the data flow.

The bus carries 32 bits of data in parallel at a transfer rate of up to 10 MHz. It uses FERA[1]-style data transfer control: the active data source being selected by a REN (Readout ENable) daisy-chain.

This specification covers the following:

1. Definition of the bus signals and bus topology.
2. The bus protocol and signals used to transmit data.
3. The bus protocol and signals used by a readout controller to indicate that data acquisition has been terminated (the “Stop” protocol).
4. The format and sequence of data words used for transmitting Eurogam events.
5. Physical aspects:
   electrical and mechanical specifications of the bus interface, connectors and cable;
   signal levels;
   timing.

1 Bus signals and bus topology

The DT32-bus signals are listed in Table 1.
DTREN/PASS is a “daisy-chain” signal, connected from one unit to the next around the bus. It is called DTREN at a module’s input and DTPASS at its output. DTPASS from one module is connected to DTREN of the next module in the direction indicated in Figure 1.

---

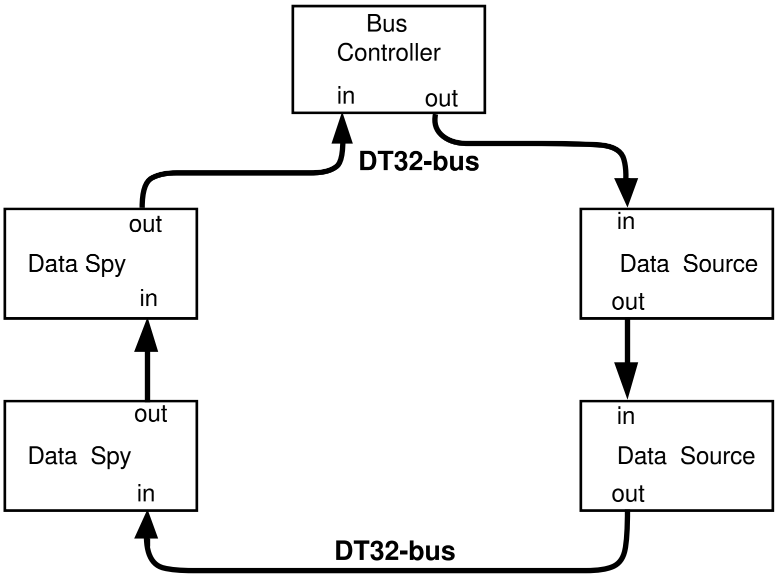
[1] 'Fast Encoding and Readout of ADCs' bus, defined by LeCroy Corp.

The remaining signals function as bussed signals but, for reasons of transmission characteristics, are also transmitted from point to point in the same direction. These signals start and end at the bus controller. All other units, apart from the currently selected data source, pass these signals from input to output unmodified. Thus all units 'downstream' of the currently selected data source will see the signals which it transmits.

Table 1.

Signal	Type	From	Use
DTREN/PASS	Daisy-chain		Readout ENable
D00-D31	Bus	Source	Data (D00 is the least significant bit)
WST	Bus	Source	Write STrobe
PAUSE	Bus	Bus Controller	Flow control signal
DTSTOP	Bus	Source	Data-acquisition termination signal

Fig 1: Illustration of the DT32-bus topology



2 Data Transfer Protocol

The data transfer protocol uses all of the DT32-bus signals apart from DTSTOP. The protocol is illustrated in Figure 2, which shows the signal sequence for a bus with two data sources, the first of which transmits three data words and the second two words. Use of the PAUSE signal (during transmission of the second data word) is also illustrated.

The bus controller (the event builder in Eurogam) initiates a data transfer cycle by asserting its DTPASS output. The DTREN/PASS daisy-chain is used to select one data source at a time: a source being selected when its DTREN input is asserted and it is not asserting DTPASS. Each source, when it is selected, transmits its data (as described below) and then asserts DTPASS. It may transmit as many data words as is necessary, and it may delay doing so for as long as is necessary. For Eurogam, the number of words is determined by the length of subevent being transmitted, and a readout controller will delay transmission until it has a subevent to transmit.

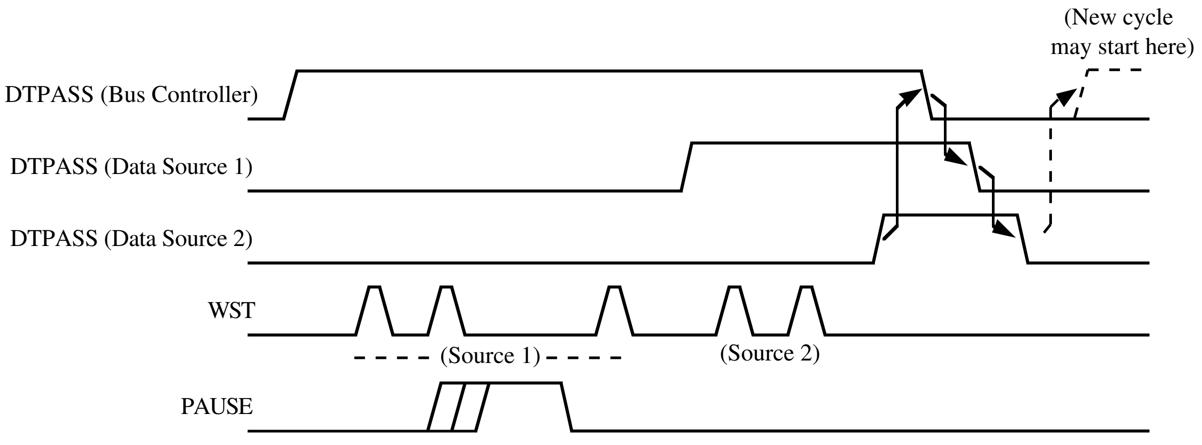
A data spy will usually assert DTPASS immediately upon receipt of DTREN; but may delay this action if necessary (for example, to delay the initiation of the *next* DT32-bus cycle while analysing the data collected during the current cycle).

When the bus controller receives the assertion of DTREN, it deasserts its DTPASS output to terminate the cycle. All other units then deassert their DTPASS outputs as they receive the deassertion of DTREN. Finally the bus controller receives DTREN deasserted, and may then initiate a new DT32-bus cycle.

A data source transmits its data, when selected, by placing each word in turn on D00-D31 and then pulsing WST – the timing of which is specified in section 5. The maximum data rate is 10 MHz, and there is no minimum rate. The bus controller transmits logic 0's on the data and WST lines so that the selected data source can logically 'OR' its signals onto the lines (this minimises the complexity of DT32-bus interface circuitry). All unselected data sources and all data spies pass the signals unchanged from input to output. Therefore, upstream of the active data source, these signals will have the value 0 (as transmitted by the bus controller), downstream they will have the values transmitted by the active data source.

The bus controller will usually be able to accept data at the maximum rate, but the PAUSE signal is provided to enable it to temporarily halt the flow of data if necessary. This signal is also passed from input to output unchanged, and when received by the selected data source causes it to stop transmitting WST pulses until PAUSE is deasserted again. However, once the bus controller has asserted PAUSE it may receive a few more data words before transmission halts, due to delays around the signal paths. This should be allowed for in the design of the controller, for example by the use of a FIFO at its input.

Fig 2: Example of a DT32-bus Data Transfer Cycle



3 Stop Protocol

This protocol is used by a data source to inform the bus controller that data acquisition has been stopped. In Eurogam, a readout controller uses it when:

a) the VXI line "TTL Trig4" (designated GO*/STOP for Eurogam) is high, and
b) the readout controller has transmitted all the data for events accepted by the trigger while data acquisition was enabled.

The protocol operates as follows (it is a standard handshake between the bus controller and the data source, using the DTREN/PASS and DTSTOP signals):

1 When the data source receives DTREN asserted (and it has *not* asserted DTPASS) it asserts DTSTOP (instead of data and WST).
2 When the bus controller receives DTSTOP asserted it deasserts DTPASS.
3 When the data source receives DTREN deasserted it deasserts DTSTOP.

All other units transmit DTSTOP from input to output unaltered (in the same way as for other 'bussed' signals).

The Stop sequence will recur whenever the bus controller attempts to initiate a DT32-bus cycle while data acquisition is stopped.

In Eurogam, it will be the first readout controller in the DTREN/PASS daisy-chain which signals Stop in this way *if* the readout controllers have not got out of step in their transmission of subevent blocks (see next section). If the readout controllers *have* got out of step then it is possible that the first (and possibly some other) readout controllers will transmit data and then a down-stream readout controller will signal Stop within the *same* DT32-bus cycle. The event builder can detect this by the absence of one or more subevent blocks from the event's data.

4 Eurogam Event Format

The Eurogam event format specifies:

1) the structure of "data words" and "tokens" carried by the DT32-bus, defined in Figure 3
2) the sequence of data words and tokens used to transmit a Eurogam event, defined in Figure 4.

Data words consist of a 16-bit data field, a 14-bit address field identifying the source of the data, and a 2-bit qualifier field carrying status information. The address field is sub-divided into 'i' and 'g' fields by software; but this has no significance for the hardware, which treats them as a single 14-bit field.

Data words in this format are generated by Eurogam data acquisition modules, and transmitted unaltered over the DT32-bus by VXI readout controllers. Readout controllers in other bus systems, however, may need to build data words from appropriate sources of information within those systems.

Tokens are DT32-bus words used to delimit events and subevents (a subevent is the contribution to an event from an individual readout controller). There are two types of token: StartEvent tokens and EndSubevent tokens. There is *no* explicit EndEvent type of token: but the identification code 2F (hex) is reserved for the last readout controller on the bus, and thus the last EndSubevent token will contain this code.

Figure 3: Eurogam DT32-bus Word Formats

Field Size:	*2*	*6*	*8*	*16*
	D31			D00
Data Word:	q q	i	g	Data
StartEvent token:	3	3F	FF	Event Sequence Number

	2	*6*	*8*	*12*	*4*
EndSubevent token:	e e	ROCO #	FF	Subevent Word-count	EN4

Key:

q	=	data qualifier bit
i	=	item number (within a group)
g	=	group number
e	=	error bit
ROCO #	=	readout controller identification code
EN4	=	Event Validation code (last 4 bits of Event Number)

Note 1: Data Words from the Master Trigger unit have: g = FF
i = 30 - 3E

Note 2: ROCO #'s are in the range 00 to 2F; with 2F being assigned to the last readout controller on the DT32-bus.

A **StartEvent token** is used to mark the beginning of an event block and to transmit the sequence number of that event. It is generated by the Master Trigger unit, which is implemented as a VXI module and thus generates data words, including the StartEvent token, for readout by a VXI readout controller. To ensure that the StartEvent token is the first word of an event:

a) the readout controller which reads the Master Trigger module must be the first in the DTREN/PASS daisy-chain, and

b) the Master Trigger module must be the first in the readout chain within that VXI crate.

This also ensures that all the readout controllers can see the StartEvent token, and therefore can capture the four least significant bits of the event sequence number (for the purpose described below).

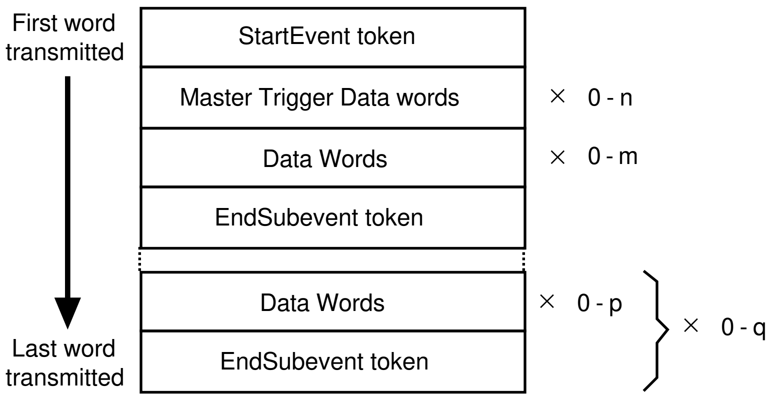
An **EndSubevent token** is generated by each readout controller to mark the end of its contribution to an event, and to transmit its identification code, word count, event validation code and error bits.

The identification codes are programmed into readout controllers when the system is initialised. The word count is the number of words, *including* the EndSubevent token, in a particular subevent. The event validation code is the 4-bit code which the Master Trigger module broadcast over the VXI backplanes together with the Event Validation signal; it is the four least significant bits of the event sequence number.

The most significant error bit (D31) is used to indicate an “event number fault”. During read-out of an event over the DT32-bus, each readout controller compares the event validation code (stored in its EndSubevent token) with the four least significant bits of the event sequence number which it captured from the StartEvent token. If these are different, then it sets (to 1) the event number fault bit to indicate that event readout has got out of synchronism in some way. Synchronism will probably have to be re-established by software action to halt data acquisition, clear sub-event buffers in the data acquisition modules and readout controllers, and then re-start data acquisition. (Of course, an event number fault can be detected by comparing the validation codes of subevents in the event builder; but doing it in hardware saves event builder processing time.)

The second error bit is currently not used, and will remain at 0.

Figure 4: Eurogam DT32-bus Event Format



Note: “× 0 - n” etc — These annotations indicate that zero or more of these words and subevent blocks can be present in an event.

5 Physical Specifications

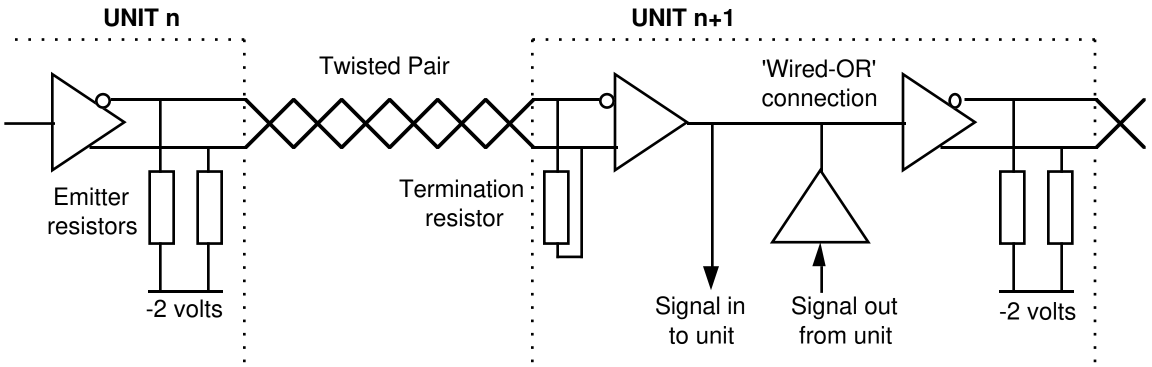
Differential ECL signals are used, preferably interfaced with 100K ECL integrated circuits.

Signal polarity: when the Signal+ wire is more positive than the Signal– wire, the signal is ‘true’, ‘asserted’ or at ‘logical 1’.

Connection topology: point-to-point, i.e. each wire-pair has a single driver and a single receiver.

Terminators: Each wire-pair is terminated at the receiver with 100 Ω between the wires. The ECL output (emitter) resistors are located in the driving module, not at the receiver (see Figure 5).

Fig 5: Positions of interface resistors; and suggested interface circuitry



Cable type: Twisted pairs, characteristic impedance of about 100 Ω, with flat (parallel wire) sections if necessary for mounting the connectors. Example: 3M Scotchflex type 1700.

Connectors: 40-way and 34-way "IDC" connectors (2 row, 0·1 inch spacing between pins) with centre-bump polarisation. Socket to be mounted on the cable, plug on the unit.
All units have two sets of connectors: one set for the input signals (labelled 'In 1' and 'In 2'), and one for the output signals (labelled 'Out 1' and 'Out 2'). Pin allocations for these connectors are specified in Table 2.

Note that there is no provision for a signal reference (earth) connection within the cables. The units connected to the DT32-bus are assumed to be at sufficiently close earth potentials to make this unnecessary.

Table 2: DT32-bus Connectors: Pin Allocations

In/Out 1			
Pin	Signal	Pin	Signal
1	DTREN/PASS+	2	DTREN/PASS –
3	D00+	4	D00 –
5	D01+	6	D01 –
•	•	•	•
•	•	•	•
33	D15+	34	D15 –
35	PAUSE+	36	PAUSE –
37	WST+	38	WST –
39	DTSTOP+	40	DTSTOP –

In/Out 2			
Pin	Signal	Pin	Signal
1	D16+	2	D16 –
3	D17+	4	D17 –
•	•	•	•
•	•	•	•
31	D31+	32	D31 –
33	Spare	34	Spare

5·1 Signal Timing Specifications

The **minimum times** between signal edges on the DT32-bus, generated by a selected data source, are defined in Figure 6. The receipt of the PAUSE signal must not cause the data source to violate any of these timings. For example, if PAUSE is received while WST is high, the WST pulse must not be shortened.

The **maximum *differential* delay** between data and WST signals which any (one) other unit may introduce, when propagating these signals from input to output, is 5nS.

Note: the use of interface circuitry as illustrated in Figure 5 will keep the differential delay between signals to a minimum (by using the minimum number of circuits in each data path). Also, the use of 100K ECL integrated circuits should result in differential delays which are considerably less than 5 nS.

Fig 6: Timing Specifications