VISTEK DESIGNERS HANDBOOK FOR V3.0 DARTnet & DARTbus SYSTEMS

CONTENTS

1. INTRODUCTION

This document defines the **V3.0 DARTnet** protocol that is used to exchange data between nodes on a DARTnet network and the **V3.0 DARTbus** protocol that is used to extend communication down to module level in modular rack frames. The document is intended to provide guidance to Vistek and 3rd party personnel in the design of DART modules, controllers and systems.

2. VERSIONS

V3.00 of the DARTnet protocol is the first version to include Class 7 modules and the first version to have features introduced by Vistek. The new features are included in rack controller V6081, although not all are available in the first release. V6081 FRS2 is identified in a rack as having revision **`r20`** or higher on the 3 digit display. The first release has a revision lower than this. Data in the node heartbeat indicates the available feature set.

V3.00 provides backwards compatibility with Avitel version **V2.3** as far as support for Class 1, 2, 3 & 4 modules is concerned. Avitel Class 5, which in Vistek networks has only been present in previous versions of the rack controller, is not supported. However, a new Vistek Class 5 is created which is the same as Class 4 but is used only for rack controllers. This is to maintain compatibility with some existing programs that expect the module in slot 15 to report back as Class 5. Future references in this document to Class 4 will therefor be deemed to include the Class 5 V6081 rack controller. Class 6, which has recently appeared in a document for **V2.4**, is not supported. Any future module classes introduced by Avitel will not be the same as future Vistek module classes from Class 7 upwards.

3. DARTNET

DARTnet uses the CAN (Controller Area Network) standard as defined by the CAN 2.0B specification which includes extensive error checking, non-destructive arbitration and automatic re-transmission. It provides efficient network communications between module rack frames and controlling devices by use of node-to-node and broadcast messages. Messages use standard CAN data frames, the 11-bit identifier providing a 6-bit node address, a 4-bit module address and another bit which is always 0.

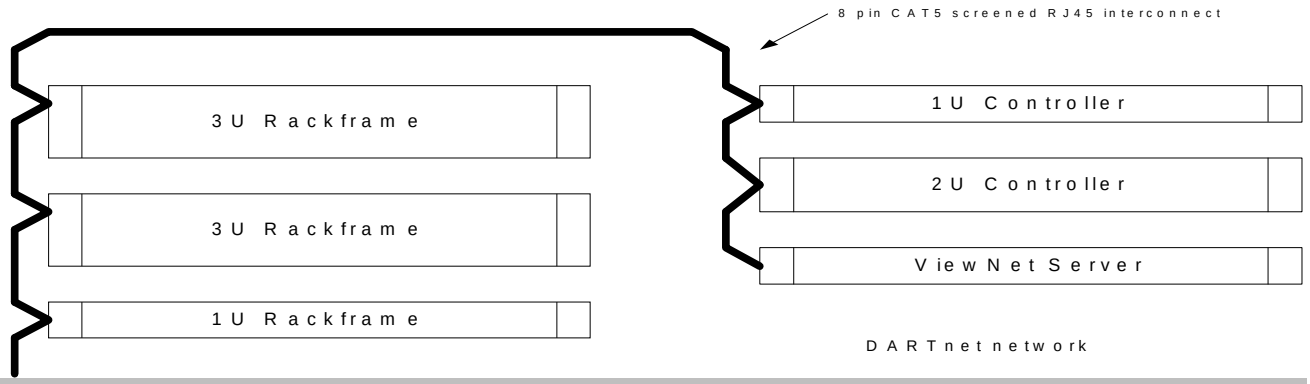
The Vistek V606 rack has 14 module slots numbered 1 – 14 plus slot 15 occupied by the rack controller which is the interface between DARTnet and the module interface, DARTbus. Slot 0 is a notional module slot in the rack controller, which provides PSU and Fan monitoring.

Node address 63 is used to identify a Broadcast message which may be accepted by any node, so each network may include up to 63 nodes, 0-62.

While rack frames are designed to accept messages to just their own node address, DARTnet controllers will accept broadcast messages as well. Broadcast messages are issued by rack controllers to inform DARTnet controllers of status changes within the modules.

3.1. DARTNET PHYSICAL CONNECTION

DARTNet uses 8 pin RJ45 connectors with a four-pair CAT5 screened cable. At 250kbps the maximum overall cable length is 200m, at 125kbpd this is extended to 500m, although, in practice, the slower speed is never used.



While rack controllers provide direct access onto the DARTnet, most DARTnet controllers use a CAN/RS232 interface device such as a DIN341 or V6041, which provides access to the DARTnet at up to 115.2kbps. The DIN341 filters out unwanted messages so that the lower RS232 speed compared to the CAN speed is not usually an issue. Speed must be set the same for all nodes on the network.

RJ45 pin	1	2	3	4	5	6	7	8
Function	CAN-H	CAN-L	0v	0v	0v	V+	V+	V+
EIA 568B Cable	Wh/Or	Or/Wh	Wh/Gn	Bl/Wh	Wh/Bl	Gn/Wh	Wh/Bn	Bn/Wh

One twisted pair carries CAN-H and CAN-L data and the remaining six wires carry power (0v and V+, nominally 24V). On Vistek networks, rack frames donate power to the network, powering any DIN341 interface devices used for connecting DARTnet controllers to the network.

DARTnet end nodes must have a termination of 120 ohm across pins 1 & 2 of the RJ45 connector. In most cases this is switch selectable. An RJ45 socket doubler or an equivalent way of avoiding stubs must be used for intermediate nodes.

3.2. DARTNET MESSAGES

The first two bytes form a message header with the 'payload' of 1 to 8 data bytes following.

3.2.1. Message header

Header 1								Header 2							
IDENTIFIER											RTR	CONTROL			
id10	id9	id8	id7	id6	id5	id4	id3	id2	id1	id0	0	dlc3	dlc2	dlc1	dlc0
0	6 Bit DARTnet destination node address for message							4 Bit slot number			0	Data Length Code (1 - 8) No. of payload data bytes			

Note that the maximum of 8 data bytes is defined by the CAN 2.0B specification even though the data length code could be set to 15.

The identifier determines the priority of the message, the lower the identifier, the higher the priority. If two sources try to transmit using the same identifier, the rest of the message determines the one that gets priority. The DARTnet protocol forms the identifier from a priority bit (10), a node/rack address and a slot address.

A **node** message has a node address range of **0-62** and transfers data to **that specific node**. The slot number is the **destination** slot.

A **broadcast** message has node address **63** and transfers data to **all interested nodes**. The slot number is the **source** slot. The source node, if applicable, is included in the message payload.

Id10 **must be** set to 0. However, because some DARTnet controllers set Id10 to a '1', all rack controllers must ignore this bit.

3.2.2. Message payload types

- **C (control) bits** – Class 3 or 4 Modules. Normally used to send controls that are single or small numbers of bits. These are stored in rack controller NV RAM (non-volatile RAM) and sent to the module when required. They are read from the rack controller rather than directly from the module. Designers should ensure that C bit changes use Modify C bits messages so that in multi-controller networks, one controller does not overwrite changes made by another. In new module designs, all C-bit controls should have the corresponding bits in the S-register reserved for their status read-back. A maximum of 32 C-bits is available.
- **S (status) bits** – Class 3 or 4 Modules. Normally used to send status codes that are single or small numbers of bits. Some carry the status of C-bit controls. S bits are saved in the rack controller's NV RAM and changes may generate a Broadcast message on the DARTnet, depending on the setting of a broadcast mask. A maximum of 32 S-bits is available.

- **I²C Refreshed messages** – Class 3 or 4 Modules. Up to 4 messages per slot of up to 6 data bytes are available, all of which are stored in the rack controller's NV RAM. Refreshed messages are used to read or write data from or to a module on a regular basis.
- Read refreshed messages are loaded via DARTnet messages (Class 3)or, by the rack controller when the module is first detected, from the Reference Data EEPROM (Class 4). The messages regularly read status from a module via the DARTbus and, if a change is detected, a Broadcast message is generated on the DARTnet.
- Write refreshed messages are loaded via DARTnet messages (Class 3 and 4) and thus remain intact in NV RAM even when a module is replaced, thereby setting a replacement module to the same control settings as the previous module.
- Different refreshed read messages may have different I²C addresses or share the same I²C address. In the latter case an offset byte distinguishes the messages. Different refreshed write messages may also share a single I²C address, in which case all or part of the first byte is reserved as a message identifier.
- If a Class 4 module requires no more than 12 bytes of control and status in addition to that provided by the C and S bits, pairs of refreshed read and write messages should be used. The refreshed read messages should occupy the first message locations in NV RAM, the refreshed write messages should immediately follow the refreshed read messages.
- If a Class 4 module requires more than 12 bytes of control and status in addition to that provided by the C and S bits, the writes should be implemented by refreshed write messages, so that data is still stored in the rack controller's NV RAM. The reads should be implemented by I²C Miscellaneous read messages, preferably using broadcast change bits in the S-register to initiate the read from the DARTnet controller.
- DARTnet controllers need to ensure that status messages are used to update their corresponding control buffers because there are separate read and write buffers in the rack controller NV RAM.
- **I²C Data registers** – Class 7 Modules. Up to 1024 registers per slot of up to 5 data bytes are available. Of these, 192 registers per slot are stored in the rack controller's NVRAM.
- Read data - Unlike Class 4 modules, detection of status change in Class 7 modules is done in the module to obviate the need for the rack controller to read a potentially huge amount of data on a regular basis. Instead, the rack controller reads flags, which indicate the need to read further data and generate Broadcast messages on the DARTnet.
- Write data – Unlike Class 4 modules, data is not written to the module on a regular basis regardless of whether or not it has changed. Instead, the flags read back from the module indicate whether the data needs to be written because, for instance, the module has been replaced or the control switch has just been changed over to Remote.
- All 1024 registers have unique Broadcasts associated with them regardless of whether they are among those (0-191) that are stored in the rack controller's NV RAM.
- The module designer determines, if the module requires more than 960 bytes of control and status, which controls to place in registers 192 upwards. These controls will not be updated automatically when a module is replaced since they are not stored in the rack controller's NV RAM. Ideally they should be used for read-only (status) data.
- All I²C Data registers share the same read or write I²C address. Different registers are differentiated by the first 2 bytes of the message payload.
- **I²C Miscellaneous messages** – Class 4 Modules. These are messages that access the module directly, as opposed to refreshed messages (Class 4) that access the module indirectly via the rack controller's NV RAM. They are not used for Class 7 modules.

- If a Class 4 module requires more than 12 bytes of control and status in addition to that provided by the C and S bits, the writes should be implemented by refreshed write messages and the reads by I^2C miscellaneous reads. The module should detect that a message block has changed and set a change bit in the S-register to trigger an S-register Broadcast. A DARTnet controller, on receiving this broadcast, should initiate an I^2C miscellaneous read to read the status from the module. Ideally a separate change bit should be available for every message block which could change. In addition, the S-register should contain a bit that toggles so that rapidly occurring successive changes also cause S-register broadcasts even when a change bit remains set.
- Writing using I^2C miscellaneous message should be avoided because a corresponding read message is also required plus a means of generating a broadcast on a change. Also, since the rack controller doesn't store miscellaneous write messages, data is lost when power is removed or the module is changed. If miscellaneous write/miscellaneous reads must be used, the module should store remote control data in addition to local control data. Messages that write but have no means of reading back should not be used because there is then no means of initialising the controls in a screen when that screen is first opened.
- I^2C miscellaneous read messages do provide a way of reading large quantities of status information from a module as long as it is not necessary to do this automatically when the information changes. Usually a single I^2C address will be used in the request followed by one or two bytes to indicate what data is being requested. Up to 7 bytes of data may be returned using an I^2C miscellaneous read message. A particular use of this is to read a time code which is changing almost continuously and which would otherwise generate almost continuous broadcasts. A Class 7 module would use one of its I^2C Data messages for this purpose and control the change flag to prevent too frequent a broadcast.
- In Vistek systems, miscellaneous reads do not occur unless the module is currently selected on a DARTnet controller. For this reason, bits defined as alarm-generating bits must be included in a broadcast message, either S or read refreshed.
- **Reference data messages** – These messages are used primarily for reading fixed information from a Class 4 module's EEPROM. The fact that this EEPROM is available distinguishes a Class 4 module from other Classes. Some older modules use reference data write messages to change controls that are held in locations 237-248 of the reference data EEPROM. These messages should write only 2 bytes at a time and 50ms should be left between messages to the same slot. These controls are known as shared memory controls and should not be used in new designs. Reference data messages are also used for retrieving fixed information from Class 7 modules, however, this is held in the module's program memory not in EEPROM.
- **NV RAM messages** – These messages are used for accessing the non-volatile RAM for a particular slot. Many of these messages are included for compatibility with legacy equipment since new messages now exist to do the same thing.
- **Heartbeat messages** – Broadcast regularly by all nodes to confirm their continued presence and report certain system values. Absence of heartbeat indicates a node being switched off or disconnected.
- **Status messages** – Requests for module status and broadcast messages indicating alarm conditions or major status changes.
- **Other messages** – Special messages for specific jobs. For instance there is a message to reset the rack controller.

3.2.3. Commands, replies and broadcasts

The first byte in the payload of every message determines the purpose of the message. The meaning of this command byte is different depending on whether it is a broadcast or if its destination is a rack controller or if it originates from a rack controller.

Messages sent to a rack controller either write data or request data. Commands 0-127 write data, commands 128-255 request data.

Messages received by a DARTnet controller are either responses to data requests or broadcasts. Broadcasts may originate from rack controllers or other nodes in the DARTnet. Command bytes 32-103 indicate that the message is a reply to a request. Other values indicate that the message is a broadcast. Only a small number of these have been defined, most are reserved for future broadcasts.

A particular type of broadcast is called a unicast. This has been introduced in the new rack controller and is only ever sent by the rack controller in response to an Initiate Broadcast command. It differs from a broadcast in the destination node address, a broadcast using address 63 whereas a unicast uses the address of a specific DARTnet controller. It is therefore rejected by the CAN filtering of all other nodes.

When a DARTnet controller requests data, the message includes a return header and a parameter number (P#) which has the value 32-103. The return header is exactly what it's name suggests and ensures the message is returned to the correct node and contains the correct amount of data. The P# is returned in the first byte of the return message payload and ensures that the DARTnet controller can relate the returned data with the requesting message. The DARTnet controller should save details of the request message with the P# so that it knows what to do with the returned message when it arrives.

3.2.4. Using the DARTnet messages

The introduction of the V6081 rack controller has opened up the possibility of making the DARTnet system more efficient and influencing the way the system designer used the DARTnet messages in the design of a DARTnet controller. Previously there has been little guidance in how the DARTnet protocol is best used in a system and, while this hasn't been important in relatively small systems, as the complexity of the system has increased, bus loading, particularly when the system is first turned on, has become a critical issue.

The first version of the V6081 was really just a copy, in terms of functionality, of the original V606 and V1608 rack controllers, plus a few enhancements. The second version (FRS2), however, has features that, if used correctly, should dramatically reduce the amount of traffic on the DARTnet.

The main traffic on a DARTnet is that involved in a DARTnet controller getting information from a rack or a module. This information can either be broadcast by the rack or polled (requested) by the controller. It is more efficient if the information can be broadcast automatically when it changes, rather than looking for a change by regular polling.

A DARTnet controller has to…

1. detect the presence of racks, modules and controllers in a network
2. detect changes in status from the modules in a network
3. make changes to controls in the modules in a network

3.2.4.1. Detecting the presence of racks, modules and controllers in a network

Racks and controllers issue heartbeats which are broadcast messages (5 or 9) occurring at regular intervals. Detecting that a heartbeat has not occurred when expected indicates that a rack or controller has been disconnected or powered off. The physical insertion or removal of a module causes a variant of broadcast message 6. With the V6081 FRS2, this type of broadcast does not occur when the rack is first turned on because most DARTnet controllers don't use them but request information on what module is in which slot when the first heartbeat is seen. This method works regardless of the order of turning on the controller and the racks.

3.2.4.2. Detecting changes in status from the modules in a network

A rack controller will issue a broadcast for every module register that has a broadcast associated with it, whenever a change is detected in that register. In an effort to extend the data capacity of Class 4 modules, some 3rd party module designers have used methods which do not involve broadcasts for signalling changes. This means that polling is necessary and this dramatically increases the traffic on the DARTnet. Even when the Class 4 refreshed messages have been used as intended, in a multi-controller network it has been necessary to poll the C-register in the rack controller's NV RAM. The V6081 FRS 2 implements a C-register broadcast that makes this unnecessary. All the registers in a Class 7 module have broadcasts associated with them so no polling is necessary.

3.2.4.3. Making changes to controls in the modules in a network

Commands that write to module registers are used to change controls in modules. Ideally a status broadcast feeds back to the controller that the change has occurred. In some instances, the module will limit the range of a control so that the read back status updates the control to the actual achieved value in the DARTnet controller.

In the case of Class 4 modules, in some DARTnet controllers, the controls in the C-Register are have been changed using Read-Modify-Write commands so that operation is correct in a multi-controller environment when different controls in the same module are being changed via different controllers. Partly due to the complication of doing so, this has not usually been extended to the other Class 4 registers. Although commands are available for doing Read-Modify-Write changes to Class 7 module registers, it is considered unnecessary because of the rapid status read back that is a feature of the V6081 rack controller. With the new C-register broadcast in the V6081 FRS2, the use of RMW for C-register controls is probably unnecessary as well.

3.2.5. Typical command sequence

- Broadcast 8 (Node Booting) received from Rack. Set Rack and its Modules off-line. This occurs only if the rack is powered up after the controller
- Broadcast 5 (Node Heartbeat) received from Rack. Scan rack for modules by
    - Requesting module status (NVRAM 0-3) from each slot OR
    - Initiating a Broadcast 6 (New Module/Module Removed) from all slots (V6081 FRS2)
- Broadcast 6 or requested Module Status received. Set each slot on-line or off-line according to information received.
- For each on-line module, get control and status data from module by
    - Requesting data from the module OR
    - Initiating Unicasts 0-4 (Data or Status) from the module (V6081 FRS2)
- Broadcast 0-3, 4 or 17 received. Update appropriate control/status registers for the module.
- Control changed. Send appropriate data write (0-3, 112) or RMW (8-11, 113) to DARTnet. Some Class 4 modules may use Miscellaneous I2C write (6).

The amount of data requested from the module initially will depend on the characteristics of the DARTnet controller (in particular, does it need to maintain the full status of the network) and whether the module is of current interest to the operator. Some controllers will request just those registers that contain status and alarm information if the module is not currently being controlled by the operator. In a Class 4 module, it is important to ensure that any status required for an alarm is in a broadcast register (S or Refreshed Read). All registers in a Class 7 module are broadcast but alarm/status bits (including module Options and Remote/Local) should be concentrated in a contiguous range of registers at the end of the register block.

3.2.6. Optimisation of DARTnet traffic

To reduce traffic on the DARTnet, improvements have been added to the new V6081 rack controller (FRS2 or higher). Ideally, a DARTnet controller should make use of these improvements when accessing nodes controlled by a V6081. This is particularly important in networks containing Class 7 modules because of their greater data capacity.

Information in the rack controller heartbeat is used to inform the DARTnet controller that certain features are supported. RC Type in the OpMode byte will be set to 01 and Flags bit 0 will be set to 1 to indicate that the following features are supported.

- C bits broadcast – tells other DARTnet controllers in the network when C bits have been changed. This makes a regular read of the C bits unnecessary in a multi-controller network.
- S mask is set to FFFFFFF when a module is detected. This makes an S-mask write unnecessary when a new module is detected unless a different value mask is required.
- Initiate broadcast for a particular slot or for all slots. Can be instant or deferred. If deferred, the broadcast is delayed until the next rack scan so if several DARTnet controllers have requested the same broadcast, only one broadcast message will be sent. Used instead of requesting data for a module slot. A broadcast can also be specified with an option bit set or clear, which is currently ignored except for broadcasts 0-3 from a Class 7 module or broadcast 4 from a Class 4 module. Note the following…
    - Requests for broadcasts other than 0-3, 4, 5, 6, 13 and 17 are ignored.
    - A request for broadcast 0-3 from a Class 7 module slot results in all read registers used by the module in that 256 register group being broadcast. a control screen is first opened. While this creates less traffic on the DARTnet, the resulting broadcasts are accepted by all DARTnet controllers and must be processed by them. The replies from individual requests for data, while creating more traffic, are rejected by the CAN address filtering on all other DARTnet controllers and therefore don't require processing.
    - However, if the 'single' bit of the initiate message is set, the broadcast will be sent to a single destination by replacing the Node Address 63 in the header with the DARTnet controller address. This is known as a **Unicast** and can only ever occur as a result of an initiate broadcast command.
    - If the 'option' bit of the initiate message is set, the broadcasts will be limited to those registers in the group that contain alarm (and other) status (as specified in the module's reference data). This is used for detecting pre-existing module alarm conditions from a module that comes on-line. Since every DARTnet controller needs this information, it is worthwhile doing it this way. The module designer should ensure that status specifying Local/Remote control and module options and any other status items that affect the control screen for that module when it first comes on-line are included in these registers.

- A request for broadcast 0-3 from, a Class 3 or 4 module initiates an S-register broadcast (4) plus all available refreshed read broadcasts (0-3) from the module. If the 'option' bit is set, it initiates just the requested broadcast.
- A request for a broadcast 6 will cause a New Module or Module Removed broadcast depending on whether a slot is occupied or not. If 'All' is specified, broadcasts are initiated for all slots.
- A request for broadcast 13, Module Identify, can be used instead of broadcast 6 as it carries the same information. Module status will be 0 if a slot is empty.
- A request for broadcast 5 (heartbeat) from a particular rack could be used in a situation where a heartbeat from a rack has been missed to detect whether the rack really has gone off-line or whether it was just a lost message. Frequent requests for broadcast 5 should be avoided, otherwise they could trigger the duplicate node detection built into any V6041 DARTnet Gateways fitted in the network.

3.2.6.1. Broadcasts v Data Requests

The use of the 'Initiate Broadcast' command with the 'All' and 'Deferred' options reduces traffic on the DARTnet significantly compared with each DARTnet controller making individual requests for data registers. However, there are pros and cons for either method and the table below summarises them.

	Initiate Broadcast	Request Data Registers
DARTnet traffic reduced (1)	Yes	No
Loading on other DARTnet controllers increased (2)	Not if Unicast specified	No
Retries possible If messages lost (3)	Not easy	Yes

- Especially if the 'deferred' option is used so that a single set of broadcasts occurs for all DARTnet controllers in many situations.
- If other DARTnet controllers have no interest in Broadcast data, they still have to accept the messages and process them. This could happen when a control screen for a module is first opened. However, if a Unicast is specified when opening a module control screen, the message will be filtered out by the CAN address filtering on other controllers. When a rack comes on-line or a module comes on-line, all controllers will have an interest in the data so a Broadcast should be specified.
- Since every request and corresponding reply is tagged with an identifying P number, the DARTnet controller can implement a generalised retry system if data requested has not been received after a certain time. More specific retry systems would be required for each situation where an initiate broadcast was used in order to check the correct broadcasts were received. This might not be considered worthwhile particularly since the reduction in traffic should make lost messages a thing of the past, especially in conjunction with the new V6041 DARTnet Gateway.

3.2.6.2. Comparison of operation

The following table shows how important it is to make use of the new initiate broadcast command, especially when working with racks of class 7 modules in multi-controller networks. It is assumed that the 'deferred' option is used and that all controllers send their initiate broadcast command about the same time. The numbers are total numbers of messages on the DARTnet.

	Without new features		With new features	
First rack heartbeat detected	16 status requests & 16 status replies (each controller).	32	1 initiate broadcast from each controller. 16 broadcasts	17
As above – 2 DARTnet controllers		64		18
As above – 6 DARTnet controllers		192		22
Module detected (class 7) with 10 status registers containing alarms, LOC/REM, options.	Request status.	20	1 initiate broadcast from each controller. 10 broadcasts.	11
	Write status mask.	1	Write status mask.	0
As above – 2 DARTnet controllers		42		12
As above – 6 DARTnet controllers		126		16
14 modules (as above) detected in a rack – 1 DARTnet controller	10 status requests & 10 status replies (each controller). Write S mask.	294	14 initiate broadcast from each controller. 10 broadcasts per module.	154
As above – 2 DARTnet controllers		588		168
As above – 6 DARTnet controllers		1764		224
Open control screen for module (class 7) with 50 control & status registers	50 data requests & 50 data replies.	100	1 initiate broadcast. 50 broadcasts.	51

The greatest reductions in traffic are on initial detections. Strategies for spreading the network load are no longer required when initiate broadcasts are used.

3.2.7. Typical command sequence for V6081 FRS2 node

- Broadcast 5 (Node Heartbeat) received from Rack.
- Scan rack for modules by initiating a Broadcast 6 (Deferred, All slots)
- Broadcast 6/3 (Module Removed) received – set slot to off-line
- Broadcast 6/2 (New Module) received from Class 4 module
    - If full module control and status registers required, request all registers
    - If just alarm/status registers required, Initiate Broadcast 0 (Deferred, Single slot) to trigger broadcasts of all broadcast registers (e.g. S-register and Refreshed Read registers) at the next refresh (rack scan).
- Broadcast 6/2 (New Module) received from Class 7 module
    - If full module control and status registers required, Initiate Unicast 0 (Single destination, Instant, Single slot) to trigger broadcast of all registers to a single controller address. Also Unicast 1, 2 and 3 if more than 256, 512 or 768 registers.
    - If just alarm/status registers required, Initiate Broadcast 0 (Option, Deferred, Single slot) to trigger broadcasts of all alarm/status registers at the next refresh (rack scan). Also Broadcast 1, 2 and 3 if more than 256, 512 or 768 registers.

Note that the data that is of interest to other controllers is broadcast if possible and the broadcasts are deferred so that they are consolidated with those triggered by other controllers.

Data of interest only to a single controller is unicast so that it is rejected by the CAN identifier filtering of other controllers and the unicasts are instant because there is no advantage in deferring them.

3.2.8. Message details

The following sections show each DARTnet message in detail, first grouped by Write, Read or Broadcast, then by Function.

	Commands new to DARTnet version 3.0 and FRS 1 version of the V6081 rack controller.
	New commands included in FRS 2 version of V6081 rack controller, some of which are required for operating with Class 7 modules.
	Proposed new command not yet implemented

3.2.9. Data Write Messages 0 - 30

Command / Data byte **à**	0	1	2	3	4	5	6	7
Write ref. msg. #0-3 Class 3/4	0-3	I2C address W	data 1/offset	data 2	data 3	data 4	data 5	data 6
(set up 'Read') Class 3/4	0-3	I2C address R	offset					
Write data register Class 7	0-3	0-255	data 1	data 2	data 3	data 4	data 5	
Write to NVR – note [1]	4	Start address	Protection	data 1	data 2	data 3	data 4	data 5
Refresh Slot	5							
Write misc. I2C Class 3/4	6	I2C address	data 1/offset	data 2	data 3	data 4	data 5	data 6
Write user reference data – unprotected [2]	7	Start address	data 1	data 2	data 3	data 4	data 5	data 6
Write user reference data – protected [3]	7	Start address	Protection	data 1	data 2	data 3	data 4	data 5
Mod ref. msg. #0-3 Class 3/4	8-11	I2C address	B0mask	B0data	B1mask	B1data	B2mask	B2data
Modify data register Class 7	8-11	0-255	offset 0-4	B0mask	B0data	B1mask	B1data	
Modify NVR – note [4]	12	Start address	Protection	B0mask	B0data	B1mask	B1data	
Write S broadcast mask	13	7:0	15:8	23:16	31:24			
Store Module Preset (in module)	14	1[5] Preset No						
Recall Module Preset (from module)	14	0 Preset No						
	15-30							

[1] (Start address XOR Protection) = FF
[2] Addresses 33 – 88 or 237 – 248 (Class 4)
[3] All addresses. (Start address XOR Protection) = FF (Class 4)
[4] (Start address XOR Protection) = FF

[5] Bit determines whether store (1) or recall (0)

3.2.10. Data Write Messages 31 - 127

Command / Data byte →	0	1	2	3	4	5	6	7
			7 6 5 4 3 2 1 0					
Initiate broadcast [1]	31[2]	B'cast No [3]	0=All, 1=Single destination [4] Option [5] 0=Deferred, 1=Instant [6] 0=This, 1=All slots	Controller address [7]				
Modify broadcast flags (slot 15)	32	Mask 7:0	Data 7:0	Mask 14:8	Data 14:8			
Modify refresh flags (slot 15)	33	Mask 7:0	Data 7:0	Mask 14:8	Data 14:8			
Modify Local/Remote flags (slot 15) Class 7	34	Mask 7:0	Data 7:0	Mask 14:8	Data 14:8			
	35-62							
Soft Reset	63	0F	B4	01				
	64-111							
Write C	112	7:0	15:8	23:16	31:24			
Modify C	113	Start Byte #	B0mask	B0data	B1mask	B1data	B2mask	B2data
	114-127							

[1] An initiated data broadcast (0-4) will not occur if the data is not yet valid. This could occur if the module has just been plugged in. Broadcasts will occur naturally when data is read from the module.
[2] Changed from Avitel proposed 137 command because it doesn't send a return header and P#.
[3] Requests for broadcasts other than 0-3, 4, 5, 6, 13 and 17 are ignored. Request for broadcast 6 gives a New Module or Module Removed broadcast(s) depending on whether slot(s) occupied.
[4] Normally broadcasts, as their name suggests, go to all nodes (ID node 63) but sometimes, when requested by initiate broadcast command, they can be directed to a single destination node so that uninterested nodes don't have to process them. This is useful when requesting data for a new module control screen. Only available for instant broadcast, not available for deferred broadcast.
[5] Option is used when a reduced set of broadcasts is available.

Broadcast	Option = 0	Option = 1
0-3	Class 4: S register broadcast plus all available refreshed read (0-3) broadcasts	Class 4: Individual broadcast as requested
0-3	Class 7: All available broadcasts in 256 register group	Class 7: All available alarm/status register broadcasts in 256 register group
4	Class 4: S register broadcast	Class 4: S register broadcast

A naturally occurring broadcast occurring before a deferred initiated broadcast will prevent the deferred broadcast happening.
[6] Deferred means the broadcast occurs at end of next module refresh if it hasn't already occurred during the module refresh. Useful when several controllers are likely to initiate the same broadcast (such as when a rack sends its first heartbeat) to prevent multiple identical broadcasts occurring.

[7] Node address of destination controller when specifying single destination. Ignored otherwise.

3.2.11. Data Request Messages [1]

Command / Data byte à	0	1	2	3	4	5	6	7
Req. ref. msg. #0-3 Class 3/4	128-131	Ret'n hdr1	Ret'n hdr2	Ret'n P#				
Returned Data	*P#*	*data 1*	*data 2*	*data 3*	*data 4*	*data 5*	*data 6*	
Req. data register Class 7	128-131	Ret'n hdr1	Ret'n hdr2	Ret'n P#	0-255			
Returned Data	*P#*	*data 1*	*data 2*	*data 3*	*data 4*	*data 5*		
Req. NVR data (Layout as for V2.3)	132	Ret'n hdr1	Ret'n hdr2	Ret'n P#	NVR address			
Returned Data	*P#*	*data 1*	*data 2*	*data 3*	*data 4*	*data 5*	*data 6*	*data 7*
Req. reference data	133	Ret'n hdr1	Ret'n hdr2	Ret'n P#	start address			
Returned Data	*P#*	*data 1*	*data 2*	*data 3*	*data 4*	*data 5*	*data 6*	*data 7*
Req. misc. I2C – Class 3/4	134	Ret'n hdr1	Ret'n hdr2	Ret'n P#	I2C address	(setup1)	(setup2)	
Returned Data	*P#*	*data 1*	*data 2*	*data 3*	*data 4*	*data 5*	*data 6*	data 7
Req. S broadcast mask	135	Ret'n hdr1	Ret'n hdr2	Ret'n P#				
Returned Data	*P#*	*7:0*	*15:8*	*23:16*	*31:24*			
Req. C	136	Ret'n hdr1	Ret'n hdr2	Ret'n P#				
Returned Data	*P#*	*7:0*	*15:8*	*23:16*	*31:24*			
	137							
Req. S	138	Ret'n hdr1	Ret'n hdr2	Ret'n P#				
Returned Data	*P#*	*7:0*	*15:8*	*23:16*	*31:24*			
Req. Module ID, FRS & Status	139	Ret'n hdr1	Ret'n hdr2	Ret'n P#				
Returned Data	*P#*	*ID MS*	*ID LS*	*FRS*	*Mod. Status*			
Request Slot NVRAM (V3.0 Layout)	140	Ret'n hdr1	Ret'n hdr2	Ret'n P#	Offset MS	Offset LS		
Returned Data	*P#*	*data 1*	*data 2*	*data 3*	*data 4*	*data 5*	*data 6*	*data 7*
	141-149							
Req. Revision Class 7	150	Ret'n hdr1	Ret'n hdr2	Ret'n P#				
Returned Data	*P#*	*FRS*	*S/W*	*F/W*	*0000* | *H/W*			
	151-159							
Req. broadcast flags (slot 15)	160	Ret'n hdr1	Ret'n hdr2	Ret'n P#				
Req. refresh flags (slot 15)	161	Ret'n hdr1	Ret'n hdr2	Ret'n P#				
Req. Local/Remote flags (slot 15)	162	Ret'n hdr1	Ret'n hdr2	Ret'n P#				
Returned Data	*P#*	*7:0*	*15:8*					
	163-255							
Failed returned data (DLC = 1)	*P#*							

[1] All data request messages include a return header and parameter number so that returned data is accepted by the correct controller and is related to the correct command.

3.2.12. Broadcast Messages 0 - 6

Broadcast / Data byte à	0	1	2	3	4	5	6	7
B'cast ref. msg. #0-3 Class 3/4	0-3 *	Node Address	data 1	data 2	data 3	data 4	data 5	data 6
Broadcast data register Class 7	0-3 *	Node Address	0-255	data 1	data 2	data 3	data 4	data 5
Broadcast S bits	4 *	Node Address	7:0	15:8	23:16	31:24		
Node heartbeat - 1608 rack controller	5	Node Address	1st +V	2nd +V	Bus +V	Temp	0 0 Op Mode	Err Flags
Node heartbeat - 606 rack controller	5	Node Address	PSU2 Volts	0	PSU1 Volts	Temp	0 0 Op Mode	Err Flags
Node heartbeat - 6081 rack controller	5 *	Node Address	PSU2 Volts	Bus Volts	PSU1 Volts	Temp	0 1 Op Mode [1]; Type ident; spare; I^2C retries off; Heartbeat rate; Refresh rate	Flags [2] = 00000001
Node DARTbus fail (slot 15)	6	Node Address	1	TBD				
New module	6 *	Node Address	2	ID MS	ID LS	FRS	Module status	
Module removed	6 *	Node Address	3	Old status				
Initialisation failed	6	Node Address	4	Old status	Reason [3]			
Slot NVRAM checksum error	6	Node Address	5	Required	Actual	ß Slot checksum on power up		
Module DARTBus fail	6	Node Address	6	Old status	Reason [4]			
NVRAM Initialised	6	Node Address	7	Indicates NVRAM was blank or invalid on power up/reset or function button held in for 5 seconds				

* Indicates the broadcast may be initiated by an Initiate Broadcast command.

[1] Heartbeat rate: 0=10s/1=5s/2=30s. Refresh rate: 0=2s/1=1s/2=500ms/3=250ms.

[2] Flags – used to indicate features supported.

- 00 No additional feautures
- 01 C-broadcast, request broadcast command (31), S-mask defaults to FFFFFFFF.

[3] Reason for failure: 0=any, 1=i2c errors, 2=write checksum errors, 3=read checksum errors.

[4] Reason for failure: 0=any, 1=i2c errors, 2=write checksum errors, 3=read checksum errors.

3.2.13. Broadcast Messages 7 - 31, 104 - 255

Broadcast Data byte à	0	1	2	3	4	5	6	7
Node PSU fail	7	Node Address	(PSU) 1 or 2					
Node booting	8	Node Address	22	ß For compatibility with old rack controllers (boot A only)				
Gateway heartbeat	9	0 Config [1]	ID MS	ID LS	FRS	Minor rev [2]	Release	
DARTNet Controller heartbeat	9	1 Config [3]	ID MS	ID LS	FRS	G'way ID MS [4]	G'way ID LS	G'way FRS
Lockout request	10	Node Address	on/off	1st node	1st slot	2nd node	2nd slot	
Help me	11	Node Address	ID MS	ID LS	Data 1	Data 2	Data 3	Data 4
To Gateway	12							
Module Identify [5]	13 *	Node Address	Module status [6]	ID MS	ID LS	FRS		
Mains Fail	14	Node Address						
Fan Status	15	Node Address	0, 1 or 2	ß 0 = Fail, 1 = OK, 2 = Fast				
DARTnet transmit error	16	Node Address	0	CAN Transmit error, probably indicating the CAN bus has been disconnected.				
DARTnet receive error	16	Node Address	1	CAN Receive error, probably indicating the CAN bus termination is incorrect.				
DARTnet overflow	16	Node Address	2	CAN Receive buffer has overflowed and a message has been missed.				
Broadcast C bits	17 *	Node Address	7:0	15:8	23:16	31:24		
	18-31							
Return parameter P#	32-103							
	104-255							

* Indicates the broadcast may be initiated by an Initiate Broadcast command.

[1] V6041, this normally shows the actual node address. DIN341, this shows the configuration switch setting (related to the node group)
[2] V6041 only. Gateway's software minor revision and release (FRS is major revision).
[3] Controller node address x 2
[4] Gateway details optional.
[5] Only ever occurs if initiated by command 31.

[6] Module status is reported as 0 if the slot is empty.

3.2.14. Messages Grouped by Function

Command/Broadcast	Data byte **à**	0	1	2	3	4	5	6	7

3.2.14.1. Class 3 or 4 I²C Refreshed messages

Write ref. msg. #0-3 Class 3/4	0-3	I2C address W	data 1/offset	data 2	data 3	data 4	data 5	data 6
(set up 'Read') Class 3/4	0-3	I2C address R	offset					
Mod ref. msg. #0-3 Class 3/4	8-11	I2C address	B0mask	B0data	B1mask	B1data	B2mask	B2data
Req. ref. msg. #0-3 Class 3/4	128-131	Ret'n hdr1	Ret'n hdr2	Ret'n P#				
Returned Data	*P#*	*data 1*	*data 2*	*data 3*	*data 4*	*data 5*	*data 6*	
B'cast ref. msg. #0-3 Class 3/4	B 0-3	Node Address	data 1	data 2	data 3	data 4	data 5	data 6

3.2.14.2. Class 7 I²C Data Register messages

Write data register Class 7	0-3	0-255	data 1	data 2	data 3	data 4	data 5	
Modify data register Class 7	8-11	0-255	offset 0-4	B0mask	B0data	B1mask	B1data	
Req. data register Class 7	128-131	Ret'n hdr1	Ret'n hdr2	Ret'n P#	0-255			
Returned Data	*P#*	*data 1*	*data 2*	*data 3*	*data 4*	*data 5*		
Broadcast data register Class 7	B 0-3	Node Address	0-255	data 1	data 2	data 3	data 4	data 5

3.2.14.3. C-bit messages

Write C	112	7:0	15:8	23:16	31:24			
Modify C	113	Start Byte #	B0mask	B0data	B1mask	B1data	B2mask	B2data
Req. C	136	Ret'n hdr1	Ret'n hdr2	Ret'n P#				
Returned Data	*P#*	*7:0*	*15:8*	*23:16*	*31:24*			
Req. NVR data [1]	132	Ret'n hdr1	Ret'n hdr2	Ret'n P#	12			
Returned Data	*P#*	*7:0*	*15:8*	*23:16*	*31:24*			
Broadcast C bits	B 17	Node Address	7:0	15:8	23:16	31:24		

3.2.14.4. S-bit messages

Write S broadcast mask	13	7:0	15:8	23:16	31:24			
Req. S broadcast mask	135	Ret'n hdr1	Ret'n hdr2	Ret'n P#				
Returned Data	*P#*	*7:0*	*15:8*	*23:16*	*31:24*			
Req. S	138	Ret'n hdr1	Ret'n hdr2	Ret'n P#				
Returned Data	*P#*	*7:0*	*15:8*	*23:16*	*31:24*			
Req. NVR data [2]	132	Ret'n hdr1	Ret'n hdr2	Ret'n P#	4			
Returned Data	P#	7:0	15:8	23:16	31:24			
Broadcast S bits	B 4	Node Address	7:0	15:8	23:16	31:24		

[1] Use in preference to Request C command to work with old and new rack controllers

[2] Use in preference to Request S command to work with old and new rack controllers

3.2.14.5. I²C Miscellaneous messages [1]

Write misc. I2C [2]	6	I2C address	data 1/offset	data 2	data 3	data 4	data 5	data 6
Req. misc. I2C	134	Ret'n hdr1	Ret'n hdr2	Ret'n P#	I2C address	(setup1)	(setup2)	
Returned Data	*P#*	*data 1*	*data 2*	*data 3*	*data 4*	*data 5*	*data 6*	data 7

3.2.14.6. Module Status messages

See also DARTbus Status messages

Req. Module ID, FRS & Status	139	Ret'n hdr1	Ret'n hdr2	Ret'n P#				
Returned Data	*P#*	*ID MS*	*ID LS*	*FRS*	*Mod. Status*			
Req. NVR data	132	Ret'n hdr1	Ret'n hdr2	Ret'n P#	0			
Returned Data	*P#*	*Mod. Status*	*ID MS*	*ID LS*	*FRS*			
Module Identify broadcast	B 13	Node Address	Module status	ID MS	ID LS	FRS		
New module broadcast	B 6	Node Address	2	ID MS	ID LS	FRS	Module status	
Module removed broadcast	B 6	Node Address	3	Old status				
Slot NVRAM checksum error broadcast	B 6	Node Address	5	Required	Actual	ß Slot checksum on power up		
Req. Revision Class 7	150	Ret'n hdr1	Ret'n hdr2	Ret'n P#				
Returned Data	*P#*	*FRS*	*S/W*	*F/W*	*0000* *H/W*			

3.2.14.7. Rack Node Status messages

See also DARTbus Status and DARTnet Status messages

*Node heartbeat - 6081 rack controller [3]	B 5	Node Address	1st +V	2nd +V	Bus +V	Temp	00	Op Mode	Err Flags
*Node heartbeat - 606 rack controller	B 5	Node Address	PSU2 Volts	0	PSU1 Volts	Temp	00	Op Mode	Err Flags
Node heartbeat - 6081 rack controller	B 5	Node Address	PSU2 Volts	Bus Volts	PSU1 Volts	Temp	01	Op Mode	Err Flags
NVRAM Initialised	B 6	Node Address	7	Indicates NVRAM was blank or invalid on power up/reset or function button held in for 10 seconds					
Node PSU fail	B 7	Node Address	(PSU) 1 or 2						
Node booting	B 8	Node Address	22	ß For compatibility with old rack controllers (boot A only)					
Mains Fail	B 14	Node Address							
Fan Status	B 15	Node Address	0, 1 or 2	ß 0 = Fail, 1 = OK, 2 = Fast					

[1] Not used for Class 7 modules.
[2] The first one or two data bytes may be used to distinguish between multiple data blocks.
[3] Old rack controller heartbeats are shown for comparison.

3.2.14.8. Other Node Status messages

Gateway heartbeat	B 9	0	Config [1]	ID MS	ID LS	FRS		
DARTNet Controller heartbeat	B 9	1	Config [2]	ID MS	ID LS	FRS		

3.2.14.9. DARTbus Status messages

Module Initialisation failed broadcast	B 6	Node Address	4	Old status	Reason			
Module DARTBus fail broadcast	B 6	Node Address	6	Old status	Reason			
Node DARTbus fail (slot 15) broadcast	B 6	Node Address	1	TBD				

3.2.14.10. DARTnet Status messages

DARTnet connection	B 16	Node Address	0	CAN Transmit error, probably indicating the CAN bus has been disconnected.
DARTnet termination	B 16	Node Address	1	CAN Receive error, probably indicating the CAN bus termination is incorrect.
DARTnet overflow	B 16	Node Address	2	CAN Receive buffer has overflowed and a message has been missed.

3.2.14.11. Module Preset messages – Class 7

Store Module Preset (in module)	14	0	Preset No						
Recall Module Preset (from module)	15	0	Preset No						

Other messages – not required for normal operation – use with care!

3.2.14.12. Old NV RAM messages [3]

Write to NVR – note [4]	4	Start address	Protection	data 1	data 2	data 3	data 4	data 5
Modify NVR – note [5]	12	Start address	Protection	B0mask	B0data	B1mask	B1data	
Req. NVR data [6]	132	Ret'n hdr1	Ret'n hdr2	Ret'n P#	NVR address			
Returned Data	*P#*	*data 1*	*data 2*	*data 3*	*data 4*	*data 5*	*data 6*	*data 7*

[1] 'Config' in a Gateway (DIN341) heartbeat is the setting of 8 DIL switches, which determine the message filtering.
[2] 'Config' in a DARTnet Controller heartbeat is its Node address x 2.
[3] Uses V2.3 NV RAM layout
[4] Start address XOR Protection = FF (Hex)
[5] Start address XOR Protection = FF (Hex)

[6] Instances with specific addresses already included above. Use in preference to new command to work with old as well as new rack controllers.

3.2.14.13. New NV RAM messages [1]

Request Slot NVRAM (V3.0 Layout)	140	Ret'n hdr1	Ret'n hdr2	Ret'n P#	Offset MS	Offset LS		
Returned Data	*P#*	*data 1*	*data 2*	*data 3*	*data 4*	*data 5*	*data 6*	*data 7*

3.2.14.14. Rack controller control messages

Soft Reset [2]	63	0F	B4	01				
Initiate broadcast	31	B'cast No	Control bits	Controller address				
Modify broadcast flags (slot 15) [3]	32	Mask 7:0	Data 7:0	Mask 14:8	Data 14:8			
Modify refresh flags (slot 15) [4]	33	Mask 7:0	Data 7:0	Mask 14:8	Data 14:8			
Modify Local/Remote flags (slot 15) [5]	34	Mask 7:0	Data 7:0	Mask 14:8	Data 14:8			
Req. broadcast flags (slot 15)	160	Ret'n hdr1	Ret'n hdr2	Ret'n P#				
Req. refresh flags (slot 15)	161	Ret'n hdr1	Ret'n hdr2	Ret'n P#				
Req. Local/Remote flags (slot 15)	162	Ret'n hdr1	Ret'n hdr2	Ret'n P#				
Returned Data	*P#*	*7:0*	*15:8*					

3.2.14.15. Broadcasts from other sources

Lockout request [6]	10	Node Address	on/off	1st node	1st slot	2nd node	2nd slot	
Help me [7]	11	Node Address	ID MS	ID LS	Data 1	Data 2	Data 3	Data 4
To Gateway [8]	12							

[1] Uses V3.0 NV RAM layout
[2] Resets the rack controller microprocessor.
[3] Controls the flags that determine whether broadcasts are enabled for each slot.
[4] Controls the flags that determine whether refresh writing of controls occurs regularly. Also available via the V6081's C-bit and S-bit registers. Always ON for Class 3 or 4 modules – normally OFF for Class 7
[5] Controls the flags that determine the way local and remote control modes work in Class 7 modules. Also available via the V6081's C-bit and S-bit registers.
[6] Not currently used in Vistek networks.
[7] Used in Vistek networks for Router Switching and Virtual Module Presets.
[8] Not currently used in Vistek networks.

4. DARTBUS

DARTbus uses the Philips I^2C-bus specification at a speed of approximately 100kbps to provide communications in a rack frame between the rack controller and the modules. A V1606 rack frame has slots numbered 1-14 running from left to right plus a special horizontal slot over the PSUs on the right for the rack controller. This occupies slots 0 and 15, slot 15 carrying the main functions of the rack controller, slot 0 providing monitoring of the PSUs and Fan. A PCB bus carries the I^2C bus (SCL and SDA) and power to each module slot plus an individual Module Select (SEL) line for each of the 14 slots. A thermistor is mounted on the DARTbus to provide a means of measuring rack temperature.

4.1. MODULE CLASSES

The Class of a module defines what resources it can access and permits modules of different complexity and cost to be accommodated in the same rack frame. A module's Class is detected by the rack controller by a combination of the voltage it reads on a Low Level Scan and the I^2C device addresses to which it responds. The voltage returned on a low-level scan is determined by the values in a resistor network in the module. The values of these have, until recently, been regarded as critical. However, unless the module is a Class 1 module, the voltage returned merely indicates whether a module is present or not. If the module also acknowledges the address of an 8582 EEPROM, it is Class 4 regardless of the precise low-level scan voltage. (See 8 LOW LEVEL SCAN RESISTORS).

Resource Type	Class 1	Class 3	Class 4	Class 7
C-bits in NV RAM	-	8 bits	32 bits	-
S-bits in NV RAM	-	8 bits	32 bits	-
I^2C Refreshed messages in NV RAM	-	4x6 bytes	4x6 bytes	-
Data registers in NV RAM	-	-	-	192x5 bytes
Data registers not in NV RAM	-	-	-	832x5 bytes
I^2C Miscellaneous read messages	-	⁂	256x7 bytes	-
I^2C Miscellaneous write messages	-	⁂	256x5 bytes	-
Identifier	via Resistor 16, 32 or 48	via Expander 0 - 255	from EEPROM 0 - 65535	from Module 0 - 65535
Reference Data	-	-	from EEPROM	from Module
Primary device (for class detection)	Resistor	I^2C ADDR 48	I^2C ADDR A6	I^2C ADDR 06

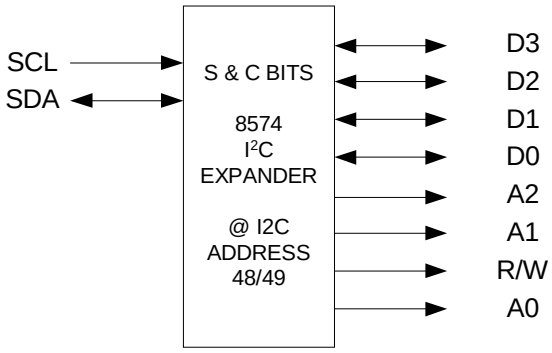
New module designs should be Class 4 or Class 7 (including Class 7A).

4.2. FAILED REQUESTS ON DARTBUS

The V6081 rack controller handles failed DARTbus requests by returning a message on the DARTnet with a data length (DLC) of 1, the data returned being just the P#. With previous rack controllers, the data returned in such circumstances was indeterminate. A DARTnet controller receiving such a reply can choose to re-send the message and/or inform the operator of the error.

4.3. CLASS 4 MODULES

In Class 4 modules, communication on the DARTbus with the rack controller is limited to I^2C data messages (implementing refreshed messages and miscellaneous I^2C messages), the 32-bit S and C registers and certain areas in the I^2C EEPROM.



4.3.1. S and C Bits

The sizes of the S and C registers in Class 4 modules are defined in the reference data read from the EEPROM at I^2C address A7. The S and C registers are implemented using an I^2C expander, data being read or written 4 bits (a nibble) at a time.

A carefully defined sequence of reading and writing nibbles alternately is used, starting with the top S nibble and top C nibble and finishing with nibble 0. If there are different numbers of S and C

nibbles, reads or writes of nibble 0 will be repeated at the end but the sequence always finishes with C nibble 0.

This sequence ensures that the R/W bit (bit 1) of the I2C expander in the module toggles and can be used in simple modules to clock the data in or out. The bit is left in the low state (always ending with a write) and starts off by going high (always beginning with a read).

While this may not be the most efficient way of transferring data, the method is always used by all rack controllers for all Class 3 and 4 modules, so the module designer must design their DART interface to take this into account.

Function	Address	Data
Select S nibble 7	48	11111111
Read S nibble	49	ddddxxxx
Write C nibble 7	48	dddd1101
Select S nibble 6	48	11111110
Read S nibble	49	ddddxxxx
Write C nibble 6	48	dddd1100
Select S nibble 5	48	11111011
Read S nibble	49	ddddxxxx
Write C nibble 5	48	dddd1001
etc…		

4.3.2. I^2C Refreshed messages

Up to 4 refreshed messages are held in the rack controller's NV RAM. Each message occupies 8 bytes.

Byte	Refreshed Write Messages	Refreshed Read Messages
0	Number of data bytes	Start address offset for device
1	I^2C write address (even)	I^2C read address (odd)
2	First data byte	First data byte
3-7	Up to 5 more data bytes as required	Always 5 more data bytes

Refresh messages occupy up to 4 consecutive message blocks in NV RAM. Refreshed read messages must always come first in memory. They are read from the reference data EEPROM at I^2C address A7. Refreshes write messages are entered into NVRAM when data is changed by a Write Refreshed message on DARTnet.

Refreshed messages are executed on every module refresh, sometimes called a 'high level rack scan', which occurs approximately every 2 seconds. If a V6081 rack controller is fitted, this may be adjusted to occur more often.

Modules that share a common rear connector should have different I^2C addresses from each other. This is to overcome a problem in old rack controllers which left old refreshed write messages in memory when a different module was inserted. The V6081 rack controller corrects this problem, but this principle should be retained if compatibility with old rack controllers is required. Modules may have several I^2C addresses or just one for every read or write message. The simplest case is when a different I^2C address is used for each message. The module can then distinguish between messages on the DARTbus by the address. However, many modules can only support a single I^2C address for writes and a single I^2C address for reads. In the case of reads, the start address offset is changed for each read refreshed message. In the case of writes, part or all of the first data byte of the message must be used as a message identifier.

4.3.2.1. Special Refreshed Read messages

Three special cases of refreshed read messages are available for devices that do not have a start address, that is, they can only be read from the start of the device. By specifying a special address offset byte, these allow different blocks of 5 bytes to be stored in NV RAM by reading more bytes than required from the device and discarding earlier bytes.

Offset byte	Bytes read	Bytes stored
253	0-4	0-4
254	0-9	5-9
255	0-14	10-14

4.3.3. I^2C Miscellaneous messages

Theoretically, 65535 miscellaneous messages are available by using both the setup1 and the setup2 bytes to define the message. However, because most DARTnet controllers need to reserve memory for each of these 'registers', it is really only practical to use setup1 to define 256 messages. Using setup2 also reduces the write payload by a further byte. Using setup2 to extend the number of miscellaneous registers is not supported by Vistek's DARTnet controllers.

4.3.4. DARTbus Messages for Class 4 modules

The following table shows the sequence of I^2C messages on the DARTbus for the different operations. The sequence for reading and writing the S and C bits has already been shown above.

Function	Class	0	1	2	3	4	5	6	7
Write I^2C refreshed #0-3	4	I^2C W Add	D1	D2	D3	D4	D5	D6	
Request I^2C refreshed #0-3	4	I^2C W Add	0						
		I^2C R Add	D1	D2	D3	D4	D5	D6	
I^2C Miscellaneous write	4	I^2C W Add	offset	(offset)	D1	D2	D3	D4	D5
I^2C Miscellaneous read	4	I^2C W Add	offset	(offset)					
		I^2C R Add	D1	D2	D3	D4	D5	D6	D7
Read Reference Data	4	A6	offset	D1	D2	D3	D4	D5	D6

4.3.4.1. Refreshed and Miscellaneous message usage notes

There are several ways of using I^2C refreshed messages and I^2C miscellaneous messages in DART modules. These methods are analogous to the X-Classes in Vistek's MDX Document System.

1. Single 48-bit read/write register. Use refreshed read message 0 (R/R #0) to read it, refreshed write message 1 (R/W #1) to write to it.
2. 2 x 48-bit read/write registers. Use R/R #0 to read register #0, R/R #1 to read register #1, R/W #2 to write register #0, R/W #3 to write register 1.
3. 4 x 48-bit read/write registers. Use R/W #0..#3 to write registers #0..#3. Set individual change flag bits in S register when a register changes. Any changes to these bits causes an S-register broadcast message to the DARTnet controller which can be programmed to initiate a I^2C miscellaneous read message to read each of the 4 registers. Note that if the data is changing rapidly, there should be an extra bit set in the S-register which toggles between successive rack scans when any of the flags is set. This ensures the rack controller always sees a change to the S-register and therefore initiates a broadcast on every register change. Note also that if registers share an I^2C address, bits in the first byte define the register, so the data capacity is reduced.
4. Single 48-bit read-only register AND 3 x 48-bit read/write registers. Use R/R #0 to read register #0. Use R/W #1..#3 to write registers #1..#3. Then use 3 change flag bits in the S-register as described above.

4.3.5. DARTbus Errors

If an error occurs while reading or writing data on the DARTbus, the operation is re-tried up to a total of 5 times and, if unsuccessful, the module slot's I^2C error count is incremented. The count is decremented on all successful operations. If the error count reaches a predefined threshold (128), the module status is set to failed and an Initialisation Failed or DARTbus Failed broadcast message is sent on the DARTnet. The status is set to $F1_{16}$ if the I^2C error count was exceeded during the module detection and initialisation phase or to $F2_{16}$ if during normal operation.

The module is then ignored by the rack controller until its status is cleared by a message from a DARTnet controller or as a result of being removed and re-inserted.

4.4. CLASS 7 MODULES

In addition to transferring data on the DARTbus between the rack controller and a module, Class 7 modules have the capability to pass other types of information by coding the first and sometimes the second data byte. Class 7 modules use just a single pair of I^2C addresses, 06 (write) and 07 (read). The type of message on the DARTbus to the module is defined by the first data byte following the I^2C address.

All messages written and read on the DARTbus to and from a Class 7 module have a checksum appended to them. Instead of the rack controller having to regularly read and write every data register as is the case with Class 3 and 4, the module decides which registers have changed or need to be written and instructs the rack controller accordingly through flags, which are read regularly instead. Requests for data normally consist of 2 bytes plus checksum after the I^2C address; read REF data has an offset and a count byte as well so is 4 bytes plus checksum.

In addition to Full Class 7, the sub-class of the module is used to define alternative or additional operation. Sub-class 1 defines the class as Class 7A (Simple Class 7) and sub-class 2 defines the class as Class 7B (Class 7 with protected write registers). No other sub-classes are currently defined.

4.4.1. Data Register messages

All data register messages carry checksums that are generated by the rack controller when writing to the module and generated by the module when reading. Automatic retries occur when the receiving end detects a checksum error. Write data register messages always write 5 data bytes plus a checksum; read data register messages always read 5 data bytes plus a checksum.

Function	Class	0	1	2	3	4	5	6	7	8
Write data register	7	06	0-3	0-255	D1	D2	D3	D4	D5	Check
Request data register	7	06	0-3	0-255	Check					
		07	D1	D2	D3	D4	D5	Check		

4.4.2. Local/Remote control mode

DART modules have, historically, had two distinct sets of controls. (1) Those when set to Local control which are stored in the module and adjusted via the front panel and (2) those when set to Remote control which are stored in the rack controller and adjusted via DARTnet. This is called Normal L/R Mode. In Alternative L/R Mode the module has a single set of controls which are adjusted from either the front panel or DART depending on how the Remote/Local switch is set on the module. These controls are stored in the module and in the rack controller. Modules may be designed to work in just one of these modes or either. If either, the mode may be changed through the module front panel or via DART (or possibly either). The L/R Mode is indicated via the module flags (see below) and may be changed using a DARTbus control message. If the mode cannot be changed in this way, this will be indicated by the module flag bit not changing. The L/R Mode controls what happens on power up and when the control source is changed.

No module to date has been designed to use the Alternative L/R Mode.

4.4.3. Flag messages

A hierarchy of message flags enables the rack controller to detect a change in the module without always having to read every data block and is also used to signal checksum errors in messages received by the module.

Function	Class	0	1	2	3	4	5	6	7	8
Read Master Flags	7	06	4	0	Check					
		07	Flags	Flags	Check					
Read Message Flags 0 – groups of 32 data blocks	7	06	4	1	Check					
		07	Flags	Check						
Read Message Flags 1 – groups of 32 data blocks	7	06	4	2	Check					
		07	Flags	Check						
Read Message Flags 2 – groups of 32 data blocks	7	06	4	3	Check					
		07	Flags	Check						
Read Message Flags 3 – groups of 32 data blocks	7	06	4	4	Check					
		07	Flags	Check						
Read Message Flags 00-07 1 set per 32 block group	7	06	4	5-12	Check					
		07	Flags	Flags	Flags	Flags	Check			
Read Message Flags 10-17 1 set per 32 block group	7	06	4	13-20	Check					
		07	Flags	Flags	Flags	Flags	Check			
Read Message Flags 20-27 1 set per 32 block group	7	06	4	21-28	Check					
		07	Flags	Flags	Flags	Flags	Check			
Read Message Flags 30-37 1 set per 32 block group	7	06	4	29-36	Check					
		07	Flags	Flags	Flags	Flags	Check			
Read Check Flags 0 – groups of 32 data blocks	7	06	4	37	Check					
		07	Flags	Check						
Read Check Flags 1 – groups of 32 data blocks	7	06	4	38	Check					
		07	Flags	Check						
Read Check Flags 2 – groups of 32 data blocks	7	06	4	39	Check					
		07	Flags	Check						
Read Check Flags 3 – groups of 32 data blocks	7	06	4	40	Check					
		07	Flags	Check						
Read Check Flags 00-07 1 set per 32 block group	7	06	4	41-48	Check					
		07	Flags	Flags	Flags	Flags	Check			
Read Check Flags 10-17 1 set per 32 block group	7	06	4	49-56	Check					
		07	Flags	Flags	Flags	Flags	Check			
Read Check Flags 20-27 1 set per 32 block group	7	06	4	57-64	Check					
		07	Flags	Flags	Flags	Flags	Check			
Read Check Flags 30-37 1 set per 32 block group	7	06	4	65-72	Check					
		07	Flags	Flags	Flags	Flags	Check			
Clear Flags	7	06	4	73	Check					

4.4.3.1. Master flags

The master flags are read every module refresh (high level rack scan). In most cases they indicate that no action is necessary. The detection of a block change flag set, for instance, triggers a read of the next level of flags followed by the next level and, finally, the data registers themselves that have changed. These data blocks can then be included in a DARTnet Data register broadcast message.

1st byte

7	6	5	4	3	2	1	0
Data change flags				Control flags			
Block 768-1023	Block 512-767	Block 255-511	Block 0-255	Module reset	Alternative L/R mode	Rem/Local changed	Remote control

2nd byte

15	14	13	12	11	10	9	8
Checksum error flags				SPARE			
Block 768-1023	Block 512-767	Block 255-511	Block 0-255				

Other bits are set when a checksum error is detected in a data message received by the module or when the control source is changed or when the module is reset or first powers up.

If a module uses 32 registers or fewer, the 2nd level of change or checksum flags may be omitted because only one bit is used at this level and the rack controller goes straight to the 3rd level.

4.4.4. Module Information and Control messages

These messages are used for reading module information and changing the way the module works. The Revision and REF data replace that read from the EEPROM in a Class 4 module. Note that software revision is included in REF data and therefore not included in the Read Revision command. The module designer should decide which firmware (FW) revision should be used if there is a choice of devices on the module. The Options are for future features that are currently undefined. A future rack controller, having checked that an option is supported by the module, could then enable it. The Local/Remote mode instructs the module how to work with regards Local and Remote control. Normal operation means a separate set of controls is maintained for Remote and Local operation, the Remote controls being stored in the rack controller's NV RAM and the Local controls being stored in the module's NV memory. Alternative operation means that nothing changes when control is switched between Remote and Local. On a Remote to Local change, the module retains its previous settings from the rack controller and on a Local to Remote change, controls are copied from the module to the rack controller's NV RAM. Controls are stored in the module's NV memory in both Remote and Local. When a module powers up or is inserted in a rack, the position of the Remote/Local switch determines whether the initial control data is taken from the module or the rack controller. The Clear Flags command is sent once the flags have been successfully read by the rack controller. On receipt of this command the module should clear all of those flags which have been read since the last Clear Flags command.

Function	Class	0	1	2	3	4	5	6	7	8
Read Revision	7	06	4	74	Check					
		07	FRS	FW	HW	Check				
Upload REF data	7	06	4	75	Offset	Count	Check			
		07	REF	REF	REF	etc.	Check			
Set Local/Remote mode	7	06	4	76	0/1	Check				
Read Options	7	06	4	77	Check					
		07	Opt	Opt	Opt	etc	Check			
Enable Options	7	06	4	78	Opt	Opt	Opt	etc	Check	

4.4.5. Module Preset messages

Module presets may be stored in the module or defined permanently in the module. These messages enable the transfer of data between the rack controller's NV RAM and preset memory in the module. NOT YET IMPLEMENTED.

Function	Class	0	1	2	3	4	5	6	7	8
Recall Module Preset	7	06	4	79	P No.	Check				
Store Module Preset	7	06	4	80	P No.	Check				

4.4.6. Activate Class 7 message

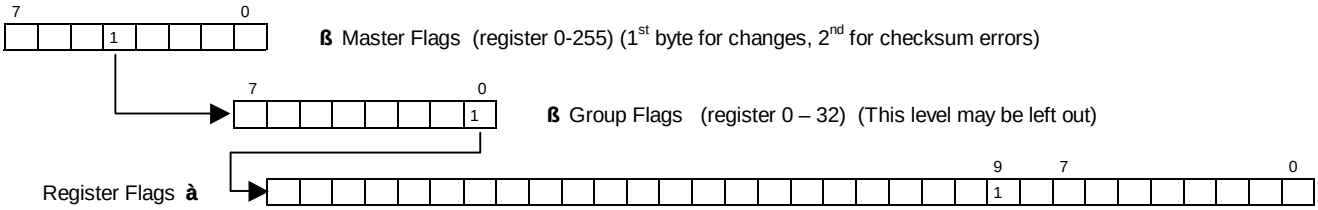
This is a special message used to switch a Class 4/7 module from Class 4 to Class 7. Class 4/7 modules always start up in Class 4 and may be switched by the rack controller to Class 7. All Class 7 modules, whether or not they are Class 4/7 must accept an Activate Class 7 message although, currently no Class 4/7 modules exist and none are likely to be designed in the future.

Function	Class	0	1	2	3	4	5	6	7	8
Activate Class 7	7A	06	255	Check						

4.4.7. Examples of Flag usage

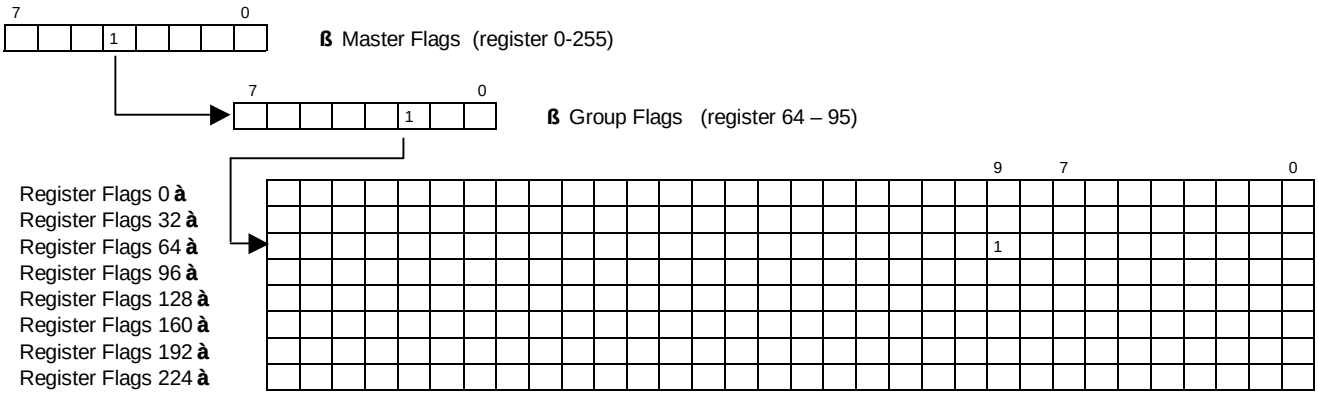
4.4.7.1. Example 1 – module with up to 32 registers

Change or checksum error in register 9.



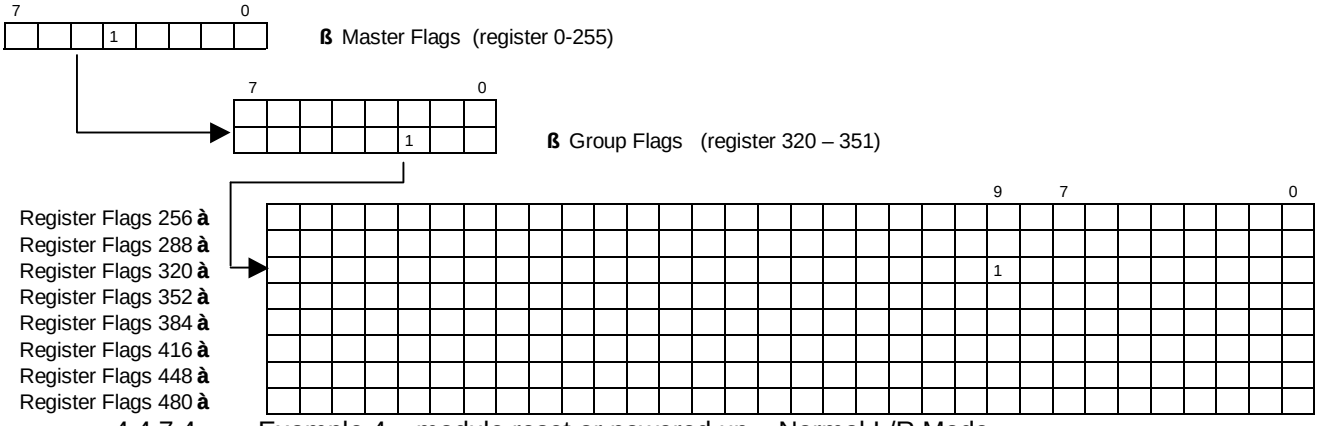
4.4.7.2. Example 2 – module with up to 256 registers

Change or checksum error in register 73.



4.4.7.3. Example 3 – module with up to 512 registers

Change or checksum error in register 329.



4.4.7.4. Example 4 – module reset or powered up – Normal L/R Mode

Module set to Local

The rack controller reads all of the registers from the module (except those outside NVRAM) as defined by the register number information in the REF data.

Module set to Remote

7	6	5	4	3	2	1	0	
				1	0	x	1	ß Master Flags (Reset + L/R Normal + Remote control)

The rack controller writes all of the write registers in NVRAM to the module (once they are valid). If the module is Class 7B, protected write registers are not written.

4.4.7.5. Example 5 – module reset or powered up – Alternative L/R Mode

Module set to Local – initial settings from module

7	6	5	4	3	2	1	0	
				1	1	x	0	ß Master Flags (Reset + L/R Alternative + Local control)

The rack controller reads all of the registers from the module (except those outside NVRAM) as defined by the register number information in the REF data and copies the data to the write registers in NVRAM.

Module set to Remote – initial settings from DART

7	6	5	4	3	2	1	0	
				1	1	x	1	ß Master Flags (Reset + L/R Alternative + Remote control)

The rack controller writes all of the write registers in NVRAM to the module (once they are valid). The module should save these settings in its local non-volatile memory (EEPROM). If the module is Class 7B, protected write registers are not written.

4.4.7.6. Example 6 – module control source changed – Normal L/R Mode

Module switched from Local to Remote control

7	6	5	4	3	2	1	0	
				0	0	1	1	ß Master Flags (L/R Normal + Control source changed + Remote control)

The rack controller writes all of the write registers in NVRAM to the module (once they are valid). If the module is Class 7B, protected write registers are not written.

Module switched from Remote to Local control

7	6	5	4	3	2	1	0	
				0	0	1	0	ß Master Flags (L/R Normal + Control source changed + Local control)

The rack controller reads all of the registers from the module (except those outside NVRAM) as defined by the register number information in the REF data.

4.4.7.7. Example 7 – module control source changed – Alternative L/R Mode

Module switched from Local to Remote control

7	6	5	4	3	2	1	0	
				0	1	1	1	ß Master Flags (L/R Alternative + Control source changed + Remote control)

The rack controller reads all of the registers from the module (except those outside NVRAM) as defined by the register number information in the REF data and copies the data to the write registers in NVRAM.

Module switched from Remote to Local control

7	6	5	4	3	2	1	0	
				0	1	1	0	ß Master Flags (L/R Alternative + Control source changed + Local control)

The rack controller will already have written all of the write registers in NVRAM to the module (once they were valid). The module should save these settings in its local non-volatile memory (EEPROM). If the module is Class 7B, protected write registers are not written.

4.4.8. DARTbus Errors in Class 7 Modules

In addition to I^2C errors, which are dealt with in the same way as in Class 4, DARTbus errors can also be due to read or write checksum errors. As with the I^2C errors, counts are decremented on successful operations and incremented on errors. If either reaches a predefined threshold (128), the module status is set to $F3_{16}$ (write checksum failure) or $F4_{16}$ (read checksum failure) and an Initialisation Failed or DARTbus Failed broadcast message is sent on the DARTnet. An extra byte in the message indicates whether the prime reason for failure is I^2C error, read checksum error or write checksum error. The module is then ignored by the rack controller until its status is cleared by a message from a DARTnet controller or as a result of being removed and re-inserted.

Checksums are added to all messages that are sent on the DARTbus. If the module detects a checksum error, this is signalled to the rack controller via the checksum flags (see above). However, these are available only for data register messages. Other types of message on the DARTbus have a checksum appended but this merely prevents malfunction if the message is corrupted; there is no means of signalling this back to the rack controller. However, if a checksum error is detected on the written request command that precedes a read on the DARTbus, the module should respond with a message that includes a checksum error (e.g. data bytes all zero and checksum set to FF_{16}) so that the rack controller has another try.

All messages read from the DARTbus have a checksum error, the whole sequence (written request followed by read data) is repeated. If after 5 attempts the data still cannot be read without a checksum error, the read checksum error count is incremented (see above) and the operation is abandoned. As indicated above, the error might be due to a checksum error having occurred when the request command was written to the module.

4.4.9. Class 7 or Class 4/7

A Class 4/7 module operates as Class 4 until the rack controller instructs it to switch to Class 7 by an Activate Class 7 command. It can therefore operate (possibly at reduced capacity) in a rack with an old rack controller. A new rack controller will detect a Class 7 module because it responds to I^2C address 06. However, it should have an EEPROM fitted at address A6, so that an old rack controller will detect it as Class 4. A true Class 7 module (not Class 4/7), must have its EEPROM loaded with reference data including a non-supported FRS number or one that references a blank control screen, in order to work sensibly in a rack with an old rack controller.

Note that it is unlikely that a Class 4/7 module will ever be designed because of the difficulties involced in designing two sets of capabilities in a single module.

4.4.10. Class 7A – Simple Class 7

A Class 7A module operates as Class 7 but does not have the complication of having to maintain the set of flags that is inherent in the operation of full Class 7. Instead the rack controller operates the module in a similar way to Class 4 by reading all read registers and writing all write registers every module refresh (high level scan). The rack controller checks whether any status has changed and if so sends a broadcast message. A sub-class byte in the module's reference data tells the rack controller that it is Class 7A. To be Class 7A, the module must…

- use a small number of registers to prevent overloading the rack controller – max recommended is 8.
- respond to Write data register, Read data register, Upload REF data and Read Revision DARTbus commands
- accept and ignore Activate Class 7 DARTbus command
- check checksum on data received
- append checksum on data written
- Class 7A modules have no provision for Module Presets, Options or Alternative Local/Remote operation

4.4.11. Class 7B – Class 7 with Protected Write Registers

A Class 7B module operates at full Class 7 but specifies in its reference data a range of registers which are not written to the module (when their data is valid) in circumstances where the rack controller would normally write all registers to the module. This normally happens when…

- The module powers up in Remote mode
- The rack controller is reset when the module is in Remote mode
- The module is switched from Local to Remote mode.

This is required in modules that derive some of their data from local storage regardless of whether set to Remote or Local, but allow that data to be modified remotely via DARTnet. Typical examples are local Presets or Video ID sets selectable remotely or locally.

5. REFERENCE DATA

Class 4 module reference data is held in an EEPROM at I^2C address A7. That for class 7 modules is read over the DARTbus by a special command and resides in the module's program memory.

5.1. CLASS 4 REFERENCE DATA

Currently programmed into the module's EEPROM using a special program from Avitel, this data is read by the rack controller to define the resources used by the module.

Address	Contents	Notes
0 – 1	Serial/batch number	0-65535
2	Week of manufacture	1-52
3	Year of manufacture	0-99 (1900-1999) 100-199 (2000-2099)
4 – 11	Option codes	Not used
12 – 32	Equipment Title	
33 – 88	User text	Not used
89 – 96	Module storage	Not used
97 – 98	Module identifier	e.g. 1656
99	FRS	Functional revision state
100	No. of S nibbles	
101	No. of C nibbles	
102 – 108	Blank check	"Avitel\0"
109 – 111	(not used)	
112	No. of read refreshed messages.	Only read refreshed messages are stored in EEPROM
113 – 120	Refreshed message #0	
121 – 128	Refreshed message #1	
129 – 136	Refreshed message #2	
137 – 144	Refreshed message #3	
145	No. of initialisation messages	Used mainly for initialising EDH chip and in older modules
146 – 236	Initialisation message headers and data	
237 – 248	Module shared memory	Used by some older class 4 modules
249 – 255	Not available	

It should be noted that the **V6081 Rack Controller** is defined as a Class 5 module so that the Avitel RefData program operates correctly with it. However, since it uses S and C bits and refreshed messages it **actually operates as a Class 4 module.**

5.2. CLASS 7 REFERENCE DATA

Replacing the data held in EEPROM in a Class 4 module, this reference data is held in the module's program memory. The following is the order of the reference bytes when uploaded by DARTbus command.

Byte	Contents	Notes
0	Ref. Data organisation version	0 – for future use if the layout of the ref. data changes
1 – 9	Software version	SS.SS.SS e.g. 01.02.03 (null terminated)
10 – 21	Date of building software	MMM DD YYYY e.g. Nov 12 2004 (null terminated)
22 – 30	Time of building software	HH:MM:SS e.g. 10.05.32 (null terminated)
31 – 51	Manufacturer	Null terminated string – max 20 characters + null
52 – 72	Equipment Title	Null terminated string – max 20 characters + null
73 – 74	Module identifier	e.g. 6302, MS byte first
75	FRS	Functional revision state
76	Sub-class	Normally 0. 1 indicates Class 7A, 2 Class 7B
77 – 78	No. of data registers used in total	0 – 1024, MS byte first
79 – 80	No. of data registers that are R/W	0 – 1024, MS byte first (Starts at register 0)
81 – 82	First alarm/status register	0 – 1023, MS byte first (1^{st} register containing an alarm)
83 – 84	Last alarm/status registers	0 – 1023, MS byte first
85 – 116	User text	Null terminated string – max 31 characters + null
117 – 118	First protected write register	0 – 1023, MS byte first – Class 7B only
119 – 120	Last protected write register	0 – 1023, MS byte first – Class 7B only

Unlike the refreshed read messages in Class 4 modules, the read and write data register messages are not defined in the REF data, just the number of registers that are used. The rack controller constructs the DARTbus messages for accessing these data registers automatically since all registers use the same I^2C address (06/07) and are the same length (5 data bytes).

The information on alarm/status registers allows a DARTnet controller to read the minimum number of registers when a module comes on-line in order to establish whether any alarm conditions exist. These registers should also contain the local/remote status and any module options that define a module variant so that basic module status can also be established before opening a module control screen.

The protected write registers specified for Class 7B modules defines a range of registers which are not written by the rack controller in circumstances when all would normally be written if they contained valid data.

6. NON-VOLATILE RAM

6.1. LAYOUT FOR NV RAM COMMANDS (V2.3 LAYOUT)

Bytes	Offset		Contents	Comments
	decimal	hex		
1	0	000	Module Status	All classes
2	1-2	001	Module ID (MS byte first)	
1	3	003	Functional Revision State (FRS)	
4	4-7	004	Status bits 31:0	LS byte first – Classes 3 & 4 only
4	8-11	008	Status broadcast mask bits 31:0	
4	12-15	00C	Control bits 31:0	
8	16-23	010	Refreshed message #0	Classes 3 & 4 only
8	24-31	018	Refreshed message #1	
8	32-39	020	Refreshed message #2	
8	40-47	028	Refreshed message #3	

6.2. LAYOUT FOR NV RAM (V3.0 LAYOUT)

Bytes	Offset		Contents	Comments	
	decimal	hex			
2	0-1	000	Module ID	MS byte first	Common – Layout version 0
1	2	002	Functional Revision State (FRS)		
1	3	003	Module Status		
1	4	004	Layout – Set to 0	**Original NVRAM layout**	
1	5	005	Class 3 & 4 data valid status	Bits: 4 - S register, 3..0 – refreshed reads *	
2	6-7	006	Number of class 7 registers	MS byte first	
	6	006	No. of C and S nibbles – class 4	[7..4] S nibbles [3..0] C nibbles	
	7	007	No. of R/R messages – class 4	Refreshed read messages, not write	
2	8-9	008	Number of class 7 R/W registers	MS byte first	
2	10-11	00A	First status register – class 7	MS byte first. Used to read alarms and other status when a module comes on-line	
2	12-13	00C	No. of status registers – class 7		
1	14	00E	Sub-class – class 7	Normally 0, 1 indicates Class 7A	
4	15	00F		4 spare bytes	
1	19	013	Checksum	Over previous 19 bytes	
4	20	014	Control bits 31:0	LS byte first – Classes 3 & 4	
4	24	018	Status bits 31:0		
4	28	01C	Status broadcast mask bits 31:0		
32					

Bytes	Offset		Contents	Comments	
	decimal	hex			
2	0-1	000	Module ID	MS byte first	Common – Layout version 1
1	2	002	Functional Revision State (FRS)		
1	3	003	Module Status		
1	4	004	Layout – Set to 1	**From Revision 2.5 when Class 7B added**	
1	5	005	Class 3 & 4 data valid status	Bits: 4 - S register, 3..0 – refreshed reads *	
2	6-7	006	Number of class 7 registers	MS byte first	
	6	006	No. of C and S nibbles – class 4	[7..4] S nibbles [3..0] C nibbles	
	7	007	No. of R/R messages – class 4	Refreshed read messages, not write	
2	8-9	008	Number of class 7 R/W registers	MS byte first	
2	10-11	00A	First status register – class 7	MS byte first. Used to read alarms and other status when a module comes on-line	
2	12-13	00C	No. of status registers – class 7		
2	14-15	00E	First protected write register	MS byte first – class 7B	
2	16-17	010	Last protected write register	MS byte first – class 7B	
1	18	012	Sub-class – class 7	0= Class 7, 1 = Class 7A, 2 = Class 7B	
1	19	013	Checksum	Over previous 19 bytes	
4	20	014	Control bits 31:0	LS byte first – Classes 3 & 4	
4	24	018	Status bits 31:0		
4	28	01C	Status broadcast mask bits 31:0		
32					

* Class 3 & 4 read data valid status is not included in the checksum.

8	32-39	020	Refreshed message #0	Classes 3 & 4 only	Class 3 & 4
8	40-47	028	Refreshed message #1		
8	48-55	030	Refreshed message #2		
8	56-63	038	Refreshed message #3		

8	32-39	020	Refreshed message #0		V6081 Rack controller (slot 15)
8	40-47	028	Refreshed message #1		
8	48-55	030	Refreshed message #2		
8	56-63	038	Refreshed message #3		
2	64-65		Broadcast flags	Slots 15..8 then 7..0	
2	66-67	03A	Refresh flags	Slots 15..8 then 7..0	
2	68-69	03C	Local/Remote operation flags	Slots 15..8 then 7..0. Used for class 7 only.	
2	70-71	03E	Refresh time (MS byte first)	Controls refresh (rack scan) rate.	
2	72-73	040	Heartbeat time (MS byte first)	Controls heartbeat rate.	
1	74	042	Error rate select	All on, all off or selected on.	
23	75-97	043	NVRAM Check string	“Vistek Electronics Ltd” (null terminated)	
1	98	062	Reset command flag	Set to distinguish between reset command and watchdog timeout	
960	32-991	020	Read registers #0 to #191	LS bytes first – all 5 bytes long	Class 7
960	992-1951	3E0	Write registers #0 to #191	LS bytes first – all 5 bytes long	
192	1952-2144	7A0	Register status #0 to #191	Status for registers in NVRAM	

Note that status for registers #192 - #1023 is elsewhere in volatile RAM.

With new messages available it is really no longer necessary to know the layout of a module slot’s NV RAM. However, existing DARTnet controllers need to work with old and new rack controllers without modification other than to make use of new features, and therefore need to continue accessing the NVRAM using command 132. This command still accesses common and class 3 & 4 NVRAM using the V2.3 layout. A new command (140) accesses the V3.0 layout NVRAM directly.

6.3. CLASS 7 REGISTER BUFFERS

The first 192 read registers and the first 192 write registers are held in buffers in NV RAM in the rack controller. Each read buffer has a corresponding write buffer that may or may not be used. The reference data defines how many registers there are in total and how many write registers are used. Registers are always allocated from 0 upwards. There is a status byte associated with each read/write register pair in NVRAM. The status for registers #192 - #1023 is elsewhere in volatile RAM

Register	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Offset	
Read 0	D0	D1	D2	D3	D4	000	0x000
Read 1	D0	D1	D2	D3	D4	005	0x005
….							
Read 190	D0	D1	D2	D3	D4		
Read 191	D0	D1	D2	D3	D4		
Write 0	D0	D1	D2	D3	D4	960	0x3C0
Write 1	D0	D1	D2	D3	D4	965	0x3C5
….							
Write 190	D0	D1	D2	D3	D4		
Write 191	D0	D1	D2	D3	D4		
Status 0	Status for registers stored in NVRAM					1920	0x780
Status 1							
…							
Status 190						2110	0x83E
Status 191						2111	0x83F

Status	
7	Read Copy
6	Read Data Valid
5	
4	
3	
2	Write to Module
1	Write Checksum
0	Write Data Valid

The 2nd and 3rd byte of the message on the DARTbus (the 1st has the I²C address 06) determines which data buffer is being written or read.

Data Register Message	D1	D2	
1st	0	00	IN NVRAM
2nd	0	01	
3rd	0	02	
…			
192nd	0	BF	
…			
256th	0	FF	
257th	1	00	
…			
512th	1	FF	
513th	2	00	
…			
768th	2	FF	
769th	3	00	
…			
1024th	3	FF	

Note that the data blocks in NVRAM are those numbered 0:00 up to 0:192 (0:00 to 0:BF hex)

7. GATEWAYS

To extend control outside the DARTnet domain, a gateway device is required. Examples of this are the Avitel DIN341 and the Vistek V6041. Their function is to translate DARTnet packets into RS232 packets and vice-versa. The DARTnet packet format (2 header bytes + up to 8 data bytes) is identical in the RS232 domain, except that an extra byte with a value of 254 (FE_{16}) is appended to the packet. The RS232 interface uses one stop bit, 8 data bits and no parity. It requires hardware handshaking (RTS, CTS) and the DIN341 is held in a reset state until DSR is asserted. The V6041 ignores DSR.

7.1. NODE ADDRESS FILTERING

The Gateway includes address filtering in the DARTnet **à** RS232 direction that can be set to pass traffic for selected addresses. In the case of the DIN341, this can be 1, 2, 4, 8, 16, 32 or 64 addresses and is set up by configuration switches on the device. When used with a DARTnet controller, the gateway must pass broadcasts (address 63) as well as traffic to that particular node. On a DIN341 with a function revision state (FRS) of less than 20, the address filtering also determines the baud rate, and this should ideally be set to 115.2kbps. On a DIN341 with FRS 20 or higher the baud rate is fixed at 115.2kbps.

The V6041 uses BCD switches, when fitted, to set up an individual address from 0 – 62. Broadcast address 63 is always included as well. Setting 63 on the BCD switches allows all addresses to be passed. If the BCD switches are not fitted, the address can be set by Gateway command.

7.2. GATEWAY COMMANDS

In addition to translating DARTnet packets to RS232 message packets, the gateway accepts special commands which are identified by a first byte with a value of 255 (FF_{16}) which is illegal as the first header byte of a CAN message. The format of these messages is…

Byte 1:	Gateway message marker 255 (FF)
Byte 2:	Data packet length (bytes 3 to n-1)
Byte 3:	Gateway command
Bytes 4 to n-1:	Data (if any)
Byte n:	Terminator 254 (FE)

The following table shows the commands supported and examples of responses. Those commands marked with an asterisk (*) are new and are not available on the DIN341.

Command	Description	TO gateway	FROM gateway	NOTES	
26 (1A)	Request special identifier 1	FF 01 1A FE	FF 07 1A “VISTEK” FE	1	
32 (20)	Request special identifier 2	FF 01 20 FE	FF 07 20 “tH9ap4” FE	1	
40 (28)	Request special identifier 3	FF 01 28 FE	FF 07 28 “&88Das” FE	1	
90 (5A)	Request manufacturer string	FF 01 5A FE	FF 07 5A “Vistek” FE		
91 (5B)	Request ID string	FF 01 5B FE	FF 06 5B “V6041” FE	2	
92 (5C)	Request ID number	FF 01 5C FE	FF 03 5C 17 99 FE	3	
93 (5D)	Request FRS	FF 01 5D FE	FF 02 5D 01 FE	4	
94 (5E)	Request option code 1	FF 01 5E FE	FF 03 5E “00” FE		
95 (5F)	Request option code 2	FF 01 5F FE	FF 03 5F “00” FE		
96 (60)	Request option code 3	FF 01 60 FE	FF 03 60 “00” FE		
97 (61)	Request configuration	FF 01 61 FE	FF 05 61 “Din” xx FE	5	
98 (62)	Set address	FF 02 62 xx FE		6	*
99 (63)	Reset V6041	FF 01 63 FE			*
100 (64)	Echo back this packet	FF 01 64 FE	FF 01 64 FE	7	
101 (65)	Error messages		FF nn 65 xx … xx FE	8	*
	DARTnet CAN error corrected		FF 02 65 00 FE		
	DARTnet CAN error detected		FF 02 65 01 FE		
	Duplicate node corrected		FF 03 65 02 nn FE	9	
	Duplicate node detected		FF 03 65 03 nn FE	9	
	All duplicate nodes corrected		FF 03 65 02 FF FE	10	

Notes:

1. These special identifiers are used to identify the V6041 as being capable of working with Avitel’s RefData program. Not available on a standard DIN341.
2. If the V6041 is set to a DIN341 compatible mode, this command returns “DIN341”
3. If the V6041 is set to a DIN341 compatible mode, this command returns 01 55 (decimal 341)
4. If the V6041 is set to a DIN341 compatible mode, this command returns 64 (decimal 100)
5. See V6401 manual for details of configuration byte xx. This Gateway response is also sent automatically shortly after the first 100 command received after reset. This allows the DARTnet controller to either update its address from the V6041 or send an address to the V6041 depending on the configuration setting received.
6. Address byte xx may be 00 to 3E (decimal 0 – 62)
7. Used for regular “are you there” check
8. Error is identified by first xx byte.; there may be more data to follow
9. nn is duplicate node address
10. Sent when a DARTnet error has been corrected to clear what could be false duplicate node alarms.

8. LOW LEVEL SCAN RESISTORS

A resistor fitted in part of the DART select input circuit of a module defines the Low Level Scan voltage. It is this that determines whether a module is present, if so, whether it is a Class 1 module or not and, if so, what type.

Value	Class	Module identifier
100	1	16 - Analogue video
6K2	1	32 - Analogue audio
16K	1	48 - Miscellaneous
31K5	3, 4, 7	Determined by other means

(Please contact Vistek for further information on this input circuit).