EDOC021

EUROGAM PROJECT

NSF DATA ACQUISITION SYSTEM

Network VME System

Edition 1.1
Apr 1991

NSF Software Systems Group
UK Science and Engineering Research Council
Daresbury Laboratory

Introduction

This document describes the Network VME System (NVS) protocol specification used to provide access to the distributed array of VME and VXI crates which form part of the NSF data acquisition system.

It is intended that this protocol be available over a number of communications media and hence the relevant *Communications Protocol* specification document should be consulted for detail in this area.

This protocol is an application carried by the RPC protocol which is described in the Eurogam document *Remote Procedure Call.*

The general structure of the protocol is described in the Eurogam document *RPC Application Programs.*

Document History

Version 1.0 July 1990 - Initial version

Version 1.1 April 1991 - add failing address in error responses

Use

Throughout this document *capability* is an 8 byte (64 bit) opaque field generated by the server and used by the client as an access token. At present this service places no access restrictions by clients to the VME address space. The capability field is provided for possible future use and at present should be set to nulls.

It is the responsibility of the client to supply valid VME addresses.

Program Specification

RPC Information

The NVS service uses only the Null authentication

NVS is supported on the UDP/IP transport only

NVS is RPC Program 28000210

This is Version 1 of the protocol

NVS uses UDP port number 10210

Procedure 0. Null procedure

This procedure does no work. There are no input parameters and no returned results. It is made available in all RPC services to allow server response testing and timing.

Procedure 1. Write

REQUEST

Writes the supplied data items to the specified VME addresses.

capability
mode
items
$address_1$
$data_1$
$address_2$
$data_2$
$address_n$
$data_n$

Mode=	1	write byte - ie 8 bits
	2	write short - ie 16 bits
	3	write long - ie 32 bits

Items is the number of address/data item pairs following. The data items are right justified in 32 bits.
Each data item is written in the specified mode to the corresponding VME address.

RESPONSE

procedure status
address

Procedure status is a 4 byte field which indicates the result of the request.

Procedure status=	0	accepted
	1	invalid VME address
	2	VME bus error occurred
	3	VME parity occurred

Address is a 4 byte field which is present only if *procedure status* is non zero and contains the VME address at which the error occurred.

Procedure 2. Write incremental

REQUEST

Writes the supplied data items starting at the specified VME address.

capability
mode
increment
items
address
$data_1$
$data_2$
$data_n$

Mode=	1	write byte - ie 8 bits
	2	write short - ie 16 bits
	3	write long - ie 32 bits

Increment is the amount by which the VME address is incremented following each write operation.
Items is the number of data items following. The data items are right justified in 32 bits.
Each data item is written in the specified mode to the current VME address.

RESPONSE

procedure status
address

Procedure status is a 4 byte field which indicates the result of the request.

Procedure status=	0	accepted
	1	invalid VME address
	2	VME bus error occurred
	3	VME parity occurred

Address is a 4 byte field which is present only if *procedure status* is non zero and contains the VME address at which the error occurred.

Procedure 3. Write incremental packed

REQUEST

Writes the supplied data starting at the specified VME address.

capability
mode
increment
items
address
data

Mode=	1	write byte - ie 8 bits
	2	write short - ie 16 bits
	3	write long - ie 32 bits

Increment is the amount by which the VME address is incremented following each write operation.
Items is the number of data items following. The data items are packed into *data* and each occupies the next 8 bits (mode=1), 16 bits (mode=2) or 32 bits (mode=3). Note that in mode=3 write incremental and write incremental packed are equivalent.

RESPONSE

procedure status
address

Procedure status is a 4 byte field which indicates the result of the request.

Procedure status=	0	accepted
	1	invalid VME address
	2	VME bus error occurred
	3	VME parity occurred

Address is a 4 byte field which is present only if *procedure status* is non zero and contains the VME address at which the error occurred.

Procedure 4. Read

REQUEST

Reads data items from the specified VME addresses.

capability
mode
items
$address_1$
$data_1$
$address_2$
$data_2$
$address_n$
$data_n$

Mode=	1	read byte - ie 8 bits
	2	read short - ie 16 bits
	3	read long - ie 32 bits

Items is the number of address/data item pairs following. The contents of the data items is undefined in the request.

RESPONSE - success

The data items read from the corresponding VME addresses are returned.

success = 0
items
$address_1$
$data_1$
$address_2$
$data_2$
$address_n$
$data_n$

The data items are right justified in 32 bits.

RESPONSE - failure

procedure status
address

Procedure status is a 4 byte field which indicates the result of the request.

Procedure status=	1	invalid VME address
	2	VME bus error occurred
	3	VME parity occurred

Address is a 4 byte field which contains the VME address at which the error occurred.

Procedure 5. Read incremental

REQUEST

Reads the required data items starting at the specified VME address.

capability
mode
increment
items
address

Mode=	1	write byte - ie 8 bits
	2	write short - ie 16 bits
	3	write long - ie 32 bits

Increment is the amount by which the VME address is incremented following each read operation.
Items is the number of data items to be returned. The data items are right justified in 32 bits.

RESPONSE - success

The data items read from the corresponding VME addresses are returned.

success = 0
items
address
$data_1$
$data_2$
$data_n$

The data items are right justified in 32 bits.

RESPONSE - failure

procedure status
address

Procedure status is a 4 byte field which indicates the result of the request.

Procedure status=	1	invalid VME address
	2	VME bus error occurred
	3	VME parity occurred

Address is a 4 byte field which contains the VME address at which the error occurred.

Procedure 6. Read incremental packed

REQUEST

Reads the required data starting at the specified VME address.

capability
mode
increment
items
address

Mode=	1	write byte - ie 8 bits
	2	write short - ie 16 bits
	3	write long - ie 32 bits

Increment is the amount by which the VME address is incremented following each read operation.

RESPONSE - success

The data items read from the corresponding VME addresses are returned.

success = 0
items
address
data

The data items are packed into *data* and each occupies the next 8 bits (mode=1), 16 bits (mode=2) or 32 bits (mode=3). Note that in mode=3 read incremental and read incremental packed are equivalent.

RESPONSE - failure

procedure status
address

Procedure status is a 4 byte field which indicates the result of the request.

Procedure status=	1	invalid VME address
	2	VME bus error occurred
	3	VME parity occurred

Address is a 4 byte field which contains the VME address at which the error occurred.