P1N Control Protocol

For Routing Switcher Control Over a Network Interface

81906206100 Rev C
V1.1 Jan. 14, 2015

Proprietary and Confidential to PESA

Revision History

Revision	Date	Author	Notes
V1.0	10/14/2008	Duncan Swords	P1N Control Protocol
V1.0, Rev B	07/22/2014	Bob Sykes	Added Section 1.6, Checksum Calculation, to document. No change to protocol
V1.1, Rev C	01/14/2015	Duncan Swords	Added command for active/standby status. Reformatted Table of Contents

TABLE OF CONTENTS

Introduction

This document describes the PESA P1N control protocol (81906206100) for use by GUI or controller software developers wanting to develop an interface to the PESA routing switcher products.

The PESA PERC2000 controller has a means of being controlled through a TCP connection while providing source and destination information

The P1 Network protocol (P1N) is the protocol deployed in the PERC2000 controller, which can be used by making a TCP connection to port 12000 of the PERC2000. It is an ASCII based protocol that uses numeric entries or names to reference sources and destinations.

The protocol uses the following format:

<SOH><Protocol>,<MP Flag>,<Msg Type>,<Command>[,<Data>],<checksum><EOT>

<SOH> ASCII SOH (0x01)

<Protocol> This is the protocol type. Three ASCII characters "P1N"

<MP Flag> This is the Multiple Packet Flag. This is an ASCII "0" if it is the last or only packet in the sequence. Any other ASCII value indicates more packets will follow to complete the data portion of the message.

<Msg Type> This is the Message Type. It is an ASCII "C" (0x43) if it is a command and an ASCII "R" (0x52) if it is a response.

<Command> P1N Command. This is echoed back in the Response.

[,<Data>] Any data or parameters which may be associated with the P1N Command or Response.

<checksum> The sum mod 256 of all previous bytes (excluding SOH). This one byte value is sent as two ASCII characters ("0" - "9", "A" - "F"), high nibble first.

<EOT> ASCII EOT (0x04)

1 Protocol #1 Network (P1N) 81906206100

1.1 Control Commands:

1.1.1 KCI - Controller ID

Identifies the type of controller and the current version of software that is executing.

Command:

KCI - Identify controller

Controller responses:

(N) - Unknown Data block or unable to process command.

(E) - Command not received correctly.

[<Controller information>]

Where:

The controller information consists of the following form:

(<Controller Name>, <Version Number>)

Controller Name is an ASCII character string (without comma) describing the controller.

Version Number is a character string. The version format is 'Vx.x.x' where x is a one or two digit numeric value. Words such as "beta" may be added to better delineate the version.

Example Command:

<SOH>P1N,0,C,KCI,<checksum><EOT>

Example Response:

<SOH>P1N,0,R,KCI,(PERC2000,V1.99.10),<checksum><EOT>

Where <SOH>, <checksum>, and <EOT> would be replaced with the appropriate ASCII values. Note that only the response has a <Data> payload.

1.1.2 KCM - Controller Mode of Operation

Identifies whether the operating mode of the controller is "Active" mode of operation or "Standby" mode of operation.

Command:

KCM – Controller Operating Mode

Controller responses:

(N) - Unknown Data block or unable to process command.

(E) - Command not received correctly.

[<Controller operating mode information>]

Where:

The controller operating mode information consists of the following:

(<Controller Operating Mode>)

Controller Operating Mode is the ASCII character string “Active” (without the quotes) if the controller is the active controller, or the ASCII string “Standby” (without the quotes) if the controller is the standby controller.

Example Command:

<SOH>P1N,0,C,KCM,<checksum><EOT>

Example Response:

<SOH>P1N,0,R,KCM,(Active),<checksum><EOT>

Where <SOH>, <checksum>, and <EOT> would be replaced with the appropriate ASCII values. Note that only the response has a <Data> payload.

1.2 Name Query Commands:

The query commands generate a data block that contains the requested definitions extracted from the controller database. The format of the definitions is described in each of the appropriate sections of this document.

NOTE: The PERC2000 control system maintains multiple names for sources, destinations, and reentries. The command responses will contain two names for these items called “name” and “panel name”. The “name” is the one that should be used in all status and switch commands. This is the master name in the PERC2000’s configuration by which everything name oriented is keyed. The “panel name” is an alternate name that is used on the remote control panel displays. Command responses will contain both names so that a user can choose which to use in their application.

1.2.1 KQ - Initiate Configuration Query

Message to initiate the request configuration information from the controller. The query can request a portion of the controller’s configuration or it can request the full configuration.

Command:

KQ<Data Request>

Data Request specifies the specific data definition requested. The data request “ALL” sends back all configured levels, sources, destinations and reentries.

Controller Response:

(N) - Unknown Data block or unable to process command.

(E) - Command not received correctly.

[<Data Block>] - Data block consisting of the definitions defining all controller configuration items queried in the KQ command. A data block can contain a portion of one or more configuration definitions. If all requested definitions cannot be sent in one data block, the definition may be split among multiple data blocks. In this case, the next KQ response's data block resumes where the last data block left off. If the controller configuration has none of the item being queried, the response from the controller is the null set.

Example Command:

<SOH>P1N,0,C,KQALL,<checksum><EOT>

Where <SOH>, <checksum>, and <EOT> would be replaced with the appropriate ASCII values. Note that only the response has a <Data> payload.

1.2.2 Level - KQLEV

Levels of control are the base element that the user can access. The level defines signals to their lowest level of use such as video, audio, or timecode. For example, a signal such RGB video would reside on one control level.

(LEV, <Name>,<Number>, <# of inputs>,<# of outputs>[,<Properties>,...])

Level # indicates where the level is mapped to for panel breakaway operation.

Properties specify those properties pertaining to the particular level

C - Level can be placed into Chop

Properties are optional. Lack of a property definition indicates that the property does not hold.

Example Command:

<SOH>P1N,0,C,KQLEV,<checksum><EOT>

Example Response:

<SOH>P1N,0,R,KQLEV,(LEV,HD VIDEO,1,128,128)(LEV,SD VIDEO,2,128,128,C),<checksum><EOT>

Where <SOH>, <checksum>, and <EOT> would be replaced with the appropriate ASCII values. Note that only the response has a <Data> payload. The above response would define a two level system of 128x128.

1.2.3 Source - KQSRC

A source is a grouping of level inputs that are used to logically represent a complex signal incoming to the router. Sources may have one input per level defined. It is not required that each level have an input defined. A level's input may be shared among any number of source groups.

For example, a source group may consist of video, audio, and timecode signals that represent a signal coming from a studio.

(SRC, <Name>, <Panel Name>, <Source Number>, [<Input #>][, [<Input #>],...])

Rules:

The level ordering implies the ordering of input numbers. The first input # refers to level 1, the second input number refers to level 2, etc. If a source is not defined on a particular level, that level's definition is left blank. (e.g. SRC, SRC 1, SRC 1, 1, 1, , 1 is defined on levels 1 and 3 but not level 2.)

Sources definitions with fewer levels defined than there are configured levels will have the added levels defined with no inputs on those levels. (e.g. a source with three levels defined on a four level system will be defined with no input assigned to level 4.)

All inputs specifications must be numeric with the number in range for the given level.

Example Command:

<SOH>P1N,0,C,KQSRC,<checksum><EOT>

Example Response:

<SOH>P1N,0,R,KQSRC,(SRC,SRC 1,SRC 1,1,1,1)(SRC,SRC 2,SRC 2,2,,2)…(SRC,SRC 128,MY SRC,128,128,2),<checksum><EOT>

Where <SOH>, <checksum>, and <EOT> would be replaced with the appropriate ASCII values. Note that only the response has a <Data> payload. The above response shows that source number 128 has a master name of "SRC 128" and a panel name of "MY SRC".

1.2.4 Destinations - KQDST

A destination is a grouping of level outputs that are used to logically represent a complex signal leaving the router. Destinations may have one output per level defined. It is not required that each level have an output defined. A level's output may be assigned to only one destination group at any one time. A destination may not use an output that has been assigned to a tieline.

For example, a destination group may consist of video, audio, and timecode signals that represent a signal going to an edit booth.

(DST, <Name>, <Panel Name>, <Destination Number>,[<Output #>][, [<Output #>],...])

Rules:

The level ordering implies the ordering of output numbers. The first output # refers to level 1, the second output number refers to level 2, etc. If a destination is not defined on a particular level, that level's definition is left blank. (e.g. DST,DST 1,DST 1,1,1,,1 is defined on levels 1 and 3 but not level 2.)

Destination definitions with fewer levels defined than there are configured levels will have the added levels defined with no outputs on those levels. (e.g. a destination with three levels defined on a four level system will be defined with no output assigned to level 4.)

All outputs specifications must be numeric with the number in range for the given level. Any other destination group or tieline may not use the output number on a level.

Example Command:

<SOH>P1N,0,C,KQDST,<checksum><EOT>

Example Response:

<SOH>P1N,0,R,KQDST,(DST,DST 1,DST 1,1,1,1)(DST,DST 2,MY DST,2,,2)…(DST,DST 128,DST 128,128,128,128),<checksum><EOT>

Where <SOH>, <checksum>, and <EOT> would be replaced with the appropriate ASCII values. Note that only the response has a <Data> payload. The above response shows that destination number 2 has a master name of “DST 2” and a panel name of “MY DST”.

1.2.5 Reentry - KQRTY

Reentries are a virtual connection within the router that allows a single source group to be routed to many destinations in a single action. A reentry appears to the user as both a source and destination. The user connects a reentry to multiple destinations using the reentry as a source. The user then can connect a source to the reentry using the reentry as a destination. This causes the source to be connected to all destinations previously assigned to the reentry.

(RTY, <Name>, <Panel Name>, <Source Number>, <Destination Number>)

Example Command:

<SOH>P1N,0,C,KQRTY,<checksum><EOT>.

Example Response:

<SOH>P1N,0,R,KQDST(RTY,RENT1,RENT1,2118,2400)(RTY,MY RENT,RENT2,2119,1494)),<checksum><EOT>

Where <SOH>, <checksum>, and <EOT> would be replaced with the appropriate ASCII values. Note that only the response has a <Data> payload. The above response shows that reentry number 2 has a master name of “RENT 2” and a panel name of “MY RENT”. It’s Source Number is 2119 and the Destination Number is 1494.

1.2.6 Salvo Group – KQSVG

A Salvo group is a grouping of salvo entries. These entries are activated when the user executes the salvo group specified.

(SVG,<Name>,<Panel Name>,<Number>,<Salvo Type>)

The Salvo type must be:

SOG - Switch Only Salvo

Example Command:

<SOH>P1N,0,C,KQSVG,<checksum><EOT>

Example Response:

<SOH>P1N,0,R,KQSVG,(SVG,SALVO 1,MY SALVO,1,SOG)(SVG,SALVO 2,SALVO 2,2,SOG),<checksum><EOT>

Where <SOH>, <checksum>, and <EOT> would be replaced with the appropriate ASCII values. Note that only the response has a <Data> payload. The above

response shows that salvo group number 1 has a master name of “SALVO 1” and a panel name of “MY SALVO”.

1.2.7 Salvos Entry – KQSVE

Salvo entries are switch specifications used in the execution of salvos. A salvo entry is assigned to a salvo group. When the salvo group is executed, all the salvo entries associated with the group are executed.

(SVE, <Salvo Group Name>, <Salvo Group Panel Name>,SOE, <Destination Name|Reentry Name>, [<Source Name|Reentry Name>] [, [<Source Name|Reentry Name>]...])

The level ordering implies the ordering of sources. The first source refers to level 1, the second source refers to level 2, etc. A level may have no source specified by simply omitting any source name i.e. place two commas together in the definition.

Example Command:

<SOH>P1N,0,C,KQSVE,<checksum><EOT>

Example Response:

<SOH>P1N,0,R,KQSVE,(SVE,SALVO 1,SALVO 1,SOE,DST 1,SRC 1,,SRC 1,SRC 1) (SVE,SALVO 1,SALVO 1,SOE,DST 2,SRC 2,,SRC 3,SRC 3)(SVE,SALVO 2,SALVO 2,SOE,DST 1,SRC 2,SRC 2,SRC 4,SRC 4),<checksum><EOT>

Where <SOH>, <checksum>, and <EOT> would be replaced with the appropriate ASCII values. Note that only the response has a <Data> payload.

1.3 Status Commands

The following are status commands that output information on delta basis. This command set provides a mechanism so the P1N device can provide status for only those destinations that have changed since the P1N device last sent out status. The command allows the P1N client to determine when update status is to be received but eliminates the constant need for receiving repeat information.

The following commands sends router status back for those destinations that have had their status change since the last tracking update command.

For this command, the controller marks each destination with a dirty bit that is set when the destination’s status changes. (e.g. the destination is locked or switched.) The bit is reset when the destination’s status has been sent in response to a tracked update command. Only those destinations with set dirty bits have their status sent in response to an update tracking command.

1.3.1 UDN - Request All Destination Update Command (Name Based)

This command requests that information be sent about all destinations whose status has changed since the last time a request All Destination Update command (numbered or named) has been made. All information is sent back in a name based format. (If the command has not been previously sent since the P1N device has come on-line, all destinations are sent.)

Command:

UDN

Controller Response:

(E) - Error in Transmittal

(N) - Format Error (i.e. extra characters in command).

(R) - A configuration file has been received. The number of destinations and the names may have changed.

[<Destination information>]

Where:

The destination information consists of 0 to the number of defined destinations' status of the following form:

(<Destination Name>, <Destination's Panel Name>, [<destination status codes>] [,[<Level 1 Source Name>],[<Level 1 Source's Panel Name>],[<Level 1 source status codes>]...])

- Commas are used to delimit the individual destination and source fields.
- If all defined levels are not specified in the response, it is assumed those levels have no defined status on them.
- The destination status codes indicate
  L - The destination is Locked

  P - The destination is Protected
- The level source status codes indicate:
  R - The source is reporting a read back error

  C - The Level source is reporting a confidence error

Example Command:

<SOH>P1N,0,C,UDN,<checksum><EOT>

Example Response:

<SOH>P1N,0,R,UDN,(DST 2,MY DST,L,SRC 2,SRC 2,,SRC 128,MY SRC,),<checksum><EOT>

Where <SOH>, <checksum>, and <EOT> would be replaced with the appropriate ASCII values. Note that only the response has a <Data> payload.

Example Response when no changes have occurred.

<SOH>P1N,0,R,UDN,<checksum><EOT>

Where <SOH>, <checksum>, and <EOT> would be replaced with the appropriate ASCII values.

1.3.2 UD1 - Request All Destination Update Command (Number Based)

This command requests that information be sent about all destinations whose status has changed since the last time a request All Destination Update command (numbered or named) has been made. All information is sent back in a number format using the same numbers used in protocol #1 commands. The numbers are variable lengths using only the number of bytes required to express the number. (e.g. 001 is sent as 1, 025 is sent as 25, 112 is sent as 112, etc.) (If the command has not been previously sent since the P1N device has come on-line, all destinations are sent.)

Command:

UD1

Controller Response:

(E) - Error in Transmittal

(N) - Format Error (i.e. extra characters in command).

(R) - A configuration file has been received. The number of destinations and the names may have changed.

[<Destination information>]

Where:

The destination information consists of 0 to the number of defined destinations' status of the following form:

(<Destination Number>, [<destination status codes>] [,[<Level 1 Source Number>],[<Level 1 source status codes>]...])

- Commas are used to delimit the individual destination and source fields.
- If all defined levels are not specified in the response, it is assumed those levels have no defined status on them.
- The destination status codes indicate
  L - The destination is Locked

  P - The destination is Protected
- The level source status codes indicate:
  R - The source is reporting a read back error

  C - The Level source is reporting a confidence error

Example Command:

<SOH>P1N,0,C,UD1,<checksum><EOT>

Example Response:

<SOH>P1N,0,R,UD1,(2,L,2,,128,),<checksum><EOT>

Where <SOH>, <checksum>, and <EOT> would be replaced with the appropriate ASCII values. Note that only the response has a <Data> payload.

Example Response when no changes have occurred.

<SOH>P1N,0,R,UD1,<checksum><EOT>

Where <SOH>, <checksum>, and <EOT> would be replaced with the appropriate ASCII values.

1.3.3 URD - Reset Destination Update Information Command

This command causes the controller to mark all destinations in the system as needing status update. The next time that an Update request is requested, all destinations will be sent.

Command:

URD

Controller Response:

(N) - Format Error (i.e. extra characters in command).

(E) - Error in Transmittal

(G) - Pending Status has been reset

Example Command:

<SOH>P1N,0,C,URD,<checksum><EOT>

Example Response:

<SOH>P1N,0,R,URD,(G),<checksum><EOT>

Where <SOH>, <checksum>, and <EOT> would be replaced with the appropriate ASCII values. Note that only the response has a <Data> payload. The above response indicates that the status has been reset.

1.3.4 UDO - Request Specific Destination Update Command (Name Based)

This command requests that information be sent about one or more specified destinations. The status is sent regardless if the destination's status has been recently modified. All information is sent back in a name based format. Using this command causes the specified destination's status to be excluded from the all destination update command until the destination's status changes.

Command:

UDO,(<Destination Name>[,<Destination Name>…])

Controller Response:

(E) - Error in Transmittal

(N) - Format Error (i.e. extra characters in command).

(R) - A configuration file has been received. The number of destinations and the names may have changed.

[<Destination information>]

Where:

The destination information is of the following form:

(<Destination Name>, <Destination's Panel Name>, [<destination status codes>] [,[<Level 1 Source Name>],[<Level 1 Source's Panel Name>],[<Level 1 source status codes>]...])

- Commas are used to delimit the individual destination and source fields.
- If an error occurs with a destination, that destination will not be included in the returned destination information. It is the user's responsibility to verify that a response has been included for each destination. An "N" response will only be returned if an error occurs with all of the destinations.
- The maximum number of destinations which can be requested in a command is 128.
- If all defined levels are not specified in the response, it is assumed those levels have no defined status on them.
- The destination status codes indicate
  L - The destination is Locked

  N - The destination was Not found.

  P - The destination is Protected
- The level source status codes indicate:
  R - The source is reporting a read back error

  C - The Level source is reporting a confidence error

Example Command:

<SOH>P1N,0,C,UDO,(DST 2,DST 3,DST 5),<checksum><EOT>

Example Response:

<SOH>P1N,0,R,UDO,(DST 2,MY DST2,L,SRC 2,SRC 2,,SRC 128,MY SRC1,)(DST 3,MY DST3,L,SRC 3,SRC 3,,SRC 129,MY SRC2,)(DST5,,N),<checksum><EOT>

Where <SOH>, <checksum>, and <EOT> would be replaced with the appropriate ASCII values.

1.3.5 UD2 - Request Specific Destination Update Command (Number Based)

This command requests that information be sent about one or more specified destinations. The status is sent regardless if the destination's status has been recently modified. All information is sent back in a number based format. Using this command causes the specified destination's status to be excluded from the all destination update command until the destination's status changes.

Command:

UD2,(<Destination Number>[,<Destination Number>…])

Controller Response:

(E) - Error in Transmittal

(N) - Format Error (i.e. extra characters in command).

(R) - A configuration file has been received. The number of destinations and the names may have changed.

[<Destination information>]

Where:

The destination information is of the following form:

(<Destination Number>, [<destination status codes>] [,[<Level 1 Source Number>],[<Level 1 source status codes>]...])

- Commas are used to delimit the individual destination and source fields.
- If an error occurs with a destination, that destination will not be included in the returned destination information. It is the user's responsibility to verify that a response has been included for each destination. An "N" response will only be returned if an error occurs with all of the destinations.
- The maximum number of destinations which can be requested in a command is 128.
- If all defined levels are not specified in the response, it is assumed those levels have no defined status on them.
- The destination status codes indicate
  L - The destination is Locked

  N - The destination was Not found.

  P - The destination is Protected
- The level source status codes indicate:
  R - The source is reporting a read back error

  C - The Level source is reporting a confidence error

Example Command:

<SOH>P1N,0,C,UD2,(2,3,4000),<checksum><EOT>

Example Response:

<SOH>P1N,0,R,UD2,(2,L,2,,128,)(3,L,3,,129,)(4000,N),<checksum><EOT>

Where <SOH>, <checksum>, and <EOT> would be replaced with the appropriate ASCII values.

1.4 Named Take/Lock P1N Commands

This command set performs actions on the router such as taking switches and locking destinations. The command set is name based. The Commands are performed using the "I" command.

1.4.1 IS - Take Named Switch

This command takes a switch on a single destination for all levels specified

Command:

IS,(<Destination Name>,[<Level 1 Source Name>],[<Level 2 Source Name>],...[>])

Where:

- Commas are used to delimit the individual destination and source level fields.
- The ">" indicator indicates that the last level specification applies for all following levels.
- If the command specifies less than the number defined levels, the command is accepted and unspecified levels are not switched.
- Omitting a source definition from a level means that no switch is taken on that level.

Controller Response:

(E) - Error in Transmittal

(N) - Illegal specification of the switch. (E.g. invalid destination, invalid source specification, too many levels specified)

(G) - Attempt to make full switch was performed. (Confidence or readback errors may indicate that an error occurred during taking of the switch.)

(L) - Destination was Locked

(B) - Part or all of the requested switch was blocked/prevented due to controller configuration. Reasons for preventing a switch may include that the source to destination is blocked, the switch specified a reentry switched to a reentry, or a reentry switch specified a source to destination block. (Portions of the switch request that did not have any configuration conflicts are attempted.)

(T) - Part or all of the requested switch was blocked/prevented due to a Tie Line Block.

Example Command:

<SOH>P1N,0,C,IS,(DST 2,SRC 1,SRC 2),<checksum><EOT>

Example Response:

<SOH>P1N,0,R,IS,(L),<checksum><EOT>

Where <SOH>, <checksum>, and <EOT> would be replaced with the appropriate ASCII values.

1.4.2 IS1 - Take Numbered Switch

This command takes a switch on a single destination for all levels specified

Command:

IS1,(<Destination Number>,[<Level 1 Source Number>],[<Level 2 Source Number>],...[>])

Where:

- Commas are used to delimit the individual destination and source level fields.
- The ">" indicator indicates that the last level specification applies for all following levels.
- If the command specifies less than the number defined levels, the command is accepted and unspecified levels are not switched.
- Omitting a source definition from a level means that no switch is taken on that level.

Controller Response:

(E) - Error in Transmittal

(N) - Illegal specification of the switch. (E.g. invalid destination, invalid source specification, too many levels specified)

(G) - Attempt to make full switch was performed. (Confidence or readback errors may indicate that an error occurred during taking of the switch.)

(L) - Destination was Locked

(B) - Part or all of the requested switch was blocked/prevented due to controller configuration. Reasons for preventing a switch may include that the source to destination is blocked, the switch specified a reentry switched to a reentry, or a reentry switch specified a source to destination block. (Portions of the switch request that did not have any configuration conflicts are attempted.)

(T) - Part or all of the requested switch was blocked/prevented due to a Tie Line Block.

Example Command:

<SOH>P1N,0,C,IS1,(2,1,2),<checksum><EOT>

Example Response:

<SOH>P1N,0,R,IS1,(G),<checksum><EOT>

Where <SOH>, <checksum>, and <EOT> would be replaced with the appropriate ASCII values.

1.4.3 IL - Lock/Protect Named Destination

This command causes the specified destination to be locked or unlocked by the P1N device. A lock protect action will occur if the destination is presently not locked/protected or if the P1N device's lock/protect priority is greater than the destination's existing lock/protect status or the P1N device has Master lock/protect priority.

Command:

IL,(<Destination Name>,<L|P|N>)

Where:

- L causes the destination to be locked.
- P causes the destination to be protected.
- N causes the destination's lock/protect status to be cleared.

Controller Response:

(E) - Error in Transmittal

(N) - Invalid destination specification

(G) - Action was taken

(L) - Higher or Equal Priority Lock is in place

(P) - Higher or Equal Priority Protect is in place

Example Command:

<SOH>P1N,0,C,IL,(DST 2,N),<checksum><EOT>

Example Response:

<SOH>P1N,0,R,IL,(G),<checksum><EOT>

Where <SOH>, <checksum>, and <EOT> would be replaced with the appropriate ASCII values.

1.4.4 IL1 - Lock/Protect Numbered Destination

This command causes the specified destination to be locked or unlocked by the P1N device. A lock protect action will occur if the destination is presently not locked/protected or if the P1N

device's lock/protect priority is greater than the destination's existing lock/protect status or the P1N device has Master lock/protect priority.

Command:

IL1,(<Destination Number>,<L|P|N>)

Where:

- L causes the destination to be locked.
- P causes the destination to be protected.
- N causes the destination's lock/protect status to be cleared.

Controller Response:

(E) - Error in Transmittal

(N) - Invalid destination specification

(G) - Action was taken

(L) - Higher or Equal Priority Lock is in place

(P) - Higher or Equal Priority Protect is in place

Example Command:

<SOH>P1N,0,C,IL1,(2,N),<checksum><EOT>

Example Response:

<SOH>P1N,0,R,IL1,(L),<checksum><EOT>

Where <SOH>, <checksum>, and <EOT> would be replaced with the appropriate ASCII values.

1.4.5 IV - Take Named Salvo

This command causes a salvo to be fired.

Command:

I V,(<Salvo Name>)

Controller Response:

(E) - Error in Transmittal

(N) - Invalid salvo specified

(G) - Salvo was fired

Example Command:

<SOH>P1N,0,C,IV,(SALVO 1),<checksum><EOT>

Example Response:

<SOH>P1N,0,R,IV,(G),<checksum><EOT>

Where <SOH>, <checksum>, and <EOT> would be replaced with the appropriate ASCII values.

1.4.6 IV1 - Take Numbered Salvo

This command causes a salvo to be fired.

Command:

I V1,(<Salvo Number>)

Controller Response:

(E) - Error in Transmittal

(N) - Invalid salvo specified

(G) - Salvo was fired

Example Command:

<SOH>P1N,0,C,IV1,(1),<checksum><EOT>

Example Response:

<SOH>P1N,0,R,IV1,(G),<checksum><EOT>

Where <SOH>, <checksum>, and <EOT> would be replaced with the appropriate ASCII values.

1.5 Unrecognized Command Response

This response is sent when there is a message is received and is not recognized by the protocol.

Response:

ERR, (<Error type>,<Incorrect data>)

Where Error Type is:

P – Protocol Error. The first three bytes were not “P1N”.

H – Header Error. Either the header data is incomplete or no command was found.

Incorrect data shows the bytes which were received to assist the user with correcting the problem. Up to 4 bytes will be sent for a protocol error and the entire header will be sent for a header error,

Examples:

Bad Protocol (No P1N prefix):

<SOH>UDN,<checksum><EOT>

Response:

<SOH>P1N,0,R,ERR,(P,UDN),<checksum><EOT>

Bad Header (Missing “,<cmd>”):

<SOH>P1N,0,C<checksum><EOT>

Response:

<SOH>P1N,0,R,ERR,(H,P1N,0,R),<checksum><EOT>

1.6 Checksum Computation

The checksum is a number derived from each data byte for the purpose of verifying data transmission on both sides of the transmission link. A data stream being transmitted computes a checksum that is sent with the data and the termination characters. The receiving equipment generates a checksum from the received data and compares the two checksums. The checksum is calculated as follows:

1. Add the ASCII decimal character codes for each character (excluding SOH) that precedes the checksum (see Figure 1 for ASCII character codes).
2. Take the sum calculated in Step 1, divide by 256 and retain the remainder.
3. Convert the remainder value to its hexadecimal value.
4. The checksum characters are the two ASCII decimal characters of the hex value, high nibble first.

Row	b8 b7 b6 b5 b4 b3 b2 b1	0	1	2	3	4	5	6	7
0	0 0 0 0	NUL 0 0 0	DLE 20 16 10	SP 40 32 20	0 60 48 30	@ 100 64 40	P 120 80 50	` 140 96 60	p 160 112 70
1	0 0 0 1	SOH 1 1 1	DC1 (XON) 21 17 11	! 41 33 21	1 61 49 31	A 101 65 41	Q 121 81 51	a 141 97 61	q 161 113 71
2	0 0 1 0	STX 2 2 2	DC2 22 18 12	" 42 34 22	2 62 50 32	B 102 66 42	R 122 82 52	b 142 98 62	r 162 114 72
3	0 0 1 1	ETX 3 3 3	DC3 (XOFF) 23 19 13	# 43 35 23	3 63 51 33	C 103 67 43	S 123 83 53	c 143 99 63	s 163 115 73
4	0 1 0 0	EOT 4 4 4	DC4 24 20 14	$ 44 36 24	4 64 52 34	D 104 68 44	T 124 84 54	d 144 100 64	t 164 116 74
5	0 1 0 1	ENQ 5 5 5	NAK 25 21 15	% 45 37 25	5 65 53 35	E 105 69 45	U 125 85 55	e 145 101 65	u 165 117 75
6	0 1 1 0	ACK 6 6 6	SYN 26 22 16	& 46 38 26	6 66 54 36	F 106 70 46	V 126 86 56	f 146 102 66	v 166 118 76
7	0 1 1 1	BEL 7 7 7	ETB 27 23 17	' 47 39 27	7 67 55 37	G 107 71 47	W 127 87 57	g 147 103 67	w 167 119 77
8	1 0 0 0	BS 10 8 8	CAN 30 24 18	(50 40 28	8 70 56 38	H 110 72 48	X 130 88 58	h 150 104 68	x 170 120 78
9	1 0 0 1	HT 11 9 9	EM 31 25 19	) 51 41 29	9 71 57 39	I 111 73 49	Y 131 89 59	i 151 105 69	y 171 121 79
10	1 0 1 0	LF 12 10 A	SUB 32 26 1A	* 52 42 2A	: 72 58 3A	J 112 74 4A	Z 132 90 5A	j 152 106 6A	z 172 122 7A
11	1 0 1 1	VT 13 11 B	ESC 33 27 1B	+ 53 43 2B	; 73 59 3B	K 113 75 4B	[133 91 5B	k 153 107 6B	{ 173 123 7B
12	1 1 0 0	FF 14 12 C	FS 34 28 1C	, 54 44 2C	< 74 60 3C	L 114 76 4C	\ 134 92 5C	l 154 108 6C	| 174 124 7C
13	1 1 0 1	CR 15 13 D	GS 35 29 1D	- 56 45 2D	= 75 61 3D	M 115 77 4D	] 135 93 5D	m 155 109 6D	} 175 125 7D
14	1 1 1 0	SO 16 14 E	RS 36 30 1E	. 56 46 2E	> 76 62 3E	N 116 78 4E	^ 136 94 5E	n 156 110 6E	~ 176 126 7E
15	1 1 1 1	SI 17 15 F	US 37 31 1F	/ 57 47 2F	? 77 63 3F	O 117 79 4F	_ 137 95 5F	o 157 111 6F	DEL 177 127 7F

Key

Character | ESC | 33 (Octal) 27 (Decimal) 1B (Hex)

Figure 1. ASCII Character Codes

Sample Checksum Calculation

Protocol format for the Controller ID command is:

<SOH>P1N,0,C,KCI,<checksum><EOT>

<checksum> is the sum mod 256 of all command character bytes (excluding SOH). This one byte value is sent as two ASCII characters ("0" - "9", "A" - "F"), high nibble first.

Checksum calculation starts by adding the ASCII decimal character codes of the command characters:

Character	ASCII Decimal Character Code
P	80
1	49
N	78
,	44
0	48
,	44
C	67
,	44
K	75
C	67
I	73
,	44
Total:	713

The result sum is divided by 256:

- 713 / 256 = 2 with a remainder of 201

The remainder value is converted to a hexadecimal value as follows:

- 201 (Dec) = C9 (Hex)

The two nibbles of the hex value are the checksum characters inserted in the command string.

Calculated checksum for the command string P1N,0,C,KCI, is C9

The command string including checksum would be:

<SOH>P1N,0,C,KCI,C9<EOT>