Tempest TEMP Protocol

Developer Guide

Revision 1.0

October 2019

© Tempest Lighting, Inc.

Contents

Document Revisions

Rev	Details	Date
0.01	First draft version for review	30 APR 2015
0.02	Second draft for review and comment	01 MAY 2015
0.03	First revision after initial implementation. Change of Baud rate for 250K to 115.2K	12 MAY 2015
0.04	Inclusion of Error Codes, corrections from internal review.	12 MAY 2015
0.05	Revision of example to match valid mnemonics. Addition of Relay status for initialisation. Addition of status parameters for DEC 4	01 OCT 2016
0.06	Correction if inconsistent mnemonics for Heater and Fan Drive Levels	13 MAY 2017
1.00	Addition of Fan Speed reporting when PWM fans fitted	OCT 2019

1 Introduction

This document describes the protocol and command set for the remote control and status monitoring of enabled Tempest DEC3 controllers.

The protocol is a half duplex ASCII syntax, based on a client–server model.

2 Terms and Conventions

All commands are initiated by a TEMP Controller and are responded to by a TEMP Responder.
Commands and Responses are collectively referred to as packets.
A single Command and associated Response is referred to as an exchange.

2.1 TEMP Controller

A TEMP controller is not to be confused with a DEC or 'DEC Controller'. The TEMP Controller, referred to here on in as a Controller will be the system of software that is a client of the Tempest DECs.
All serial packets sent by the Controller will be referred to as a command.

2.2 TEMP Responder

A TEMP Responder, for the purpose of this document, refers to a Tempest DEC and from here on, is referred to as a Responder.
All serial packets sent by a Responder will be referred to as a response.

3 Connectivity

3.1 Physical Connection

The connection to the Tempest Equipment is via an RS485 Buss with Controller Idle State bias, compliant with ANSI E1.20 section 2.4.1. at 115.2K Baud, 8 data bits, two stop bits.

3.2 DEC Configuration

For a DEC to be operable with TEMP, the DEC must be configured to be in 'BASIC' mode. For further details on configuring a DEC, refer to the accompanying user manual.

4 Responder Addressing

The responder address will be the electronic UID of the responder. A responder UID is comprised of a PLASA manufacturer ID and a unique manufacturer serial number. For Tempest equipment, the vendor ID or VID is $544C. The remaining 8 characters are a hexadecimal extrapolation of the responder serial number, so for example, a serial number of 64035 would be represented by a hexadecimal value of $0000FA23. These two strings combined together form the responder address $544C0000FA23.

The responder serial number can be found on the Tempest electronics or by accessing the user display and scrolling the status information.

A GET command to address $FFFFFFFFFFFF will solicit a response from all responders, which having more than one responder on a single buss will cause a data conflict.

A SET command to address $FFFFFFFFFFFF will be acted on by a responder but not responded to.

5 Command Structure

Each packet is transmitted as an ASCII string, enclosed in the reserved characters '<' and '>'.
Any traffic before the < and after the > characters shall be disregarded.
On receipt of a < character before the final > character, all preceding data shall be disregarded.
Each packet is addressed to an associated Responder.
Each packet contains a reserved Command or Response Identifier character, being '?' or '!' respectively.

The responder address shall be sent immediately after the command or response Identifier.

There are two classes of exchange, **SET** to configure and **GET** to retrieve status data.

5.1 Get Command Example

`<?:$544C0000FA23;GET:STA>`

5.2 Get Response Example

`<!:$544C0000FA23;STA:ACV=#223,ACC=#5.3,ATM=#32,RHM=#62,RLY=AUT>`

5.3 Set Command Example

`<?:$544C0000FA23;SET:RLY=OFF>`

5.4 Set Response Example

`<!:$544C0000FA23;RLY=OFF>`

6 Packet Parsing

6.1 Order of Precedence

Packets are syntactically arranged for parsing in order of precedence, the precedence being a string literal, semi-colon, colon comma and vertical line.

6.2 Type Identifiers

- Decimal values are preceded by a hash character **#**

- Hexadecimal values are preceded by a dollar character **$**

- Text strings are enclosed by the double quote character **"**

- All other data implicitly represents and enumerated constant

6.3 Reserved Characters

The following character are reserved for packet delineation and text encapsulation:

**< > " **

Reserved characters used in strings will be preceded by the reserved escape character **** .

Character	Use	Description
\	\\	Backslash
<	\<	Start of packet
>	\>	Packet terminator
"	\"	String Literal Quote

7 Packet Parameters

Each Command packet will contain one parameter, optionally with an argument. Each Response packet will contain one or more parameters each with a parameter argument. Parameters with arguments will be in the format PPP=A, where PPP is the three letter Mnemonic or PID to identify the parameter and A being the argument value or data of the parameter. Parameters without arguments will be in the format PPP, where PPP is the three letter Mnemonic or PID to identify the parameter.

Get command packets may have one parameter argument.

Set command packets will have one parameter argument.

 Response packet parameters are cardinal with no pre-determination of the order.

7.1 Command and Response PIDs

7.1.1 STA

GET Command

This is used to return the current status of the DEC, no parameter arguments are sent with this request.

GET Response

Responses to the request include the following values:

- ACC   AC RMS Lamp current

- ACV   AC RMS Line Voltage

- ATE   Air Temperature (External) ℃

- ATM   Air temperature (Main) ℃

- DLH   Heater Drive Level %

- DLF   Fan Drive Level %

- Fxx   Fan Speed (xx) RPM *****

- LHR   Lamp hours

- PCT   PCB temperature ℃

- RHM   Relative Humidity (Main) %

- RLY   Relay Status (INI |OFF | SBY | RUN | TRP)

* Note: xx represents the fan index and will only be reported when the appropriate fan(s) are fitted and detected. For example, F01.

7.1.2 CFG

GET Command

This is used to return the current configuration of the DEC, no parameter arguments are sent with this request.

GET Response

Responses to the request include the following values:

- LOT   Lower Temperature Setting ℃

- UPT   Upper Temperature Setting ℃

- TRP   Trip Out Temperature Setting ℃

- RHM   Relative Humidity Maximum Setting %

- RUN   Lamp Current 'On' State Threshold AC Amps RMS

7.1.3 VER

GET Command

This is used to return version information associated with the DEC system, no parameter arguments are sent with this request.

GET Response

Responses to the request include the following values:

- HWR   Hardware Revision Text Description

- FWR   Firmware Revision Text Description

7.1.4 RLY

SET Command

This is used to change the operational mode of the relay in a DEC. valid arguments for this PID are:

- AUT   Automatic On, Relay Closed Unless Over Temperature.

- OFF   Manual Off, Relay Held Open

SET Response

Response to this parameter will be one of the following enumerated constants:

- INI   Relay Open, System Initialisation

- OFF   Manual Off state

- SBY   Relay Closed, Lamp Load Off

- RUN   Relay Closed, Lamp Load On

- TRP   Relay Automatically Tripped Open, Over Temperature

7.1.5 LBL

Labels are 7 bit ASCII strings up to 32 characters long, enclosed in " literal quotes. Reserved characters are escaped as detailed in section Reserved Characters

GET Command

This retrieves the current label string assigned to the DEC, no parameter arguments are sent with this request.

GET Response

- LBL   Quote Enclosed DEC Label

SET Command

This writes a new label to the responder, the new label is sent as a parameter argument, formatted as a quote enclosed string.

SET Response

- LBL   The Request String Echoed Back

8 Error Codes

In the case that a responder in not able to parse a Command packet, one of the following parameter and argument will be returned:

SET Response

- ERR   Hex Error Code Number

The table below describes the possible error codes

Code	Description
$01	Packet Format Error
$02	Unknown Command Class
$03	Unknown Command Parameter
$04	Parameter Argument Data Invalid
$FF	Un-handled Error