AptiLoop Protocol

Introduction

The structure is master-slave where a slave is a power module (AD) and the master is the interrogating device (CC) – such as a computer serial screen for manual use or as phone app for automatic use.

The protocol is based on the NMEA 0183 style message format

Message format from CC to attached device AD:

$<command type>,<address>,<data><CR>

The address field is a single ASCII digit, range '0' (address 0) to 'F' (address 15), as the AptiLoop hardware supports a maximum of 15 devices. No checksum is required for CC to AD commands.

Message format from AD to CC:

$<response type>,<address>,<data1>,<data 2>.....<data n>*<checksum><CR>

The checksum is calculated using the same algorithm as NMEA0183 and results in two printable ASCII digits.

Command types:

1. APA: device debug control.

Command format:

$APA,<address>,<data><CR>

The data field is ASCII-coded HEX (i.e. range '0' to 'F') and is allocated bitwise as follows:

bit0 - 1 = enable analogue inputs debug info
bit1 - 1 = enable state change debug info
bit2 - 1 = enable button changes debug info
bit3 - 1 = disable output sentence checksum

e.g. if the analogue inputs debug info option is enable and the output sentence checksum is disabled the HEX code is 0x1001; the ASCII equivalent of this is '9'.

Device response:

$APA,<address>,<data><CR>

e.g. to enable the full analogue input debug data output format on the AD with address 3 the following command is used:

$APA,3,1<CR>

The AD will return the following response:

$APA,3,<status>*<checksum>

where <status> = 1 if the command is successful and 0 if the command fails.

2. APE: device status control

Command format:

$APE,<address>,<data><CR>

The data field is allocated follows:

0 - no action
1 = reset attached device
2 = turn off attached device
3 = turn on attached device
4 = remove attached device from node

All other ASCII values are invalid and will be discarded.

Response format:

$APE,<status>*<checksum>

e.g. to turn on device 12 the following message is sent:

$APE,C,3<CR>

The AD will return the following response:

$APE,C,<status>*<checksum>

where <status> = 1 if the command is successful and 0 if the command fails.

3. APF: get device fault code(s)

Command format:

$APF,<address><CR>

Device response:

$APF,<address>,<fault status>,<fault code>*<checksum><CR>

where <fault status> = 0 indicates no faults and the <fault code> field is not sent, <fault status> = 1 indicates that there is a fault with the device and the <fault code> field contains extended information about the fault type.

Extended fault codes:

TBA

4. API - get device information

Command format:

$API,<address><CR>

Device response:

$API,<address>,<device type>,<serial number>,<firmware version>,<hardware revision>*<checksum><CR>

where:

<address> - device address, one ASCII character, range '0' to 'F'
<device type> - five ASCII characters e.g. VAS10
<device serial number> - eight ASCII characters e.g. 12345678
<firmware revision> - two ASCII characters e.g. 01
<hardware revision> - one ASCII character e.g. A

5. APQ: get device data

Command format:

$APQ,<address><CR>

Device response:

$APQ,<address>,<data0>,data1>,.....,<datan>*<checksum><CR>

The following data field types are supported:

<address> - device address, one ASCII character, range '0' to 'F'
<status> - device status, TBA
<device type> - five ASCII characters e.g. VAR00
<VINA> - three ASCII characters, range 00.0 to 50.0V
<VINB> - three ASCII characters, range 00.0 to 50.0V
<VOUT> - three ASCII characters, range 00.0 to 50.0V
<IINA> - three ASCII characters, range 00.0 to 49.9A
<IINB> - three ASCII characters, range 00.0 to 49.9A
<IOUT> - three ASCII characters, range 00.0 to 19.9A
<TBOX> - three ASCII characters, range -40 to (+)125
<TBATT> - three ASCII characters, range -40 to (+)125

Note that if the AD does not support one or more of the above analogue signals then a null field will be sent e.g. ,,. No leading zero suppression will be used on the voltage, current and temperature fields e.g. TBATT = 5.1 degrees will be sent as ,051,.

6. APS: device address setup

NOTE that this command has to be issued with the loop through one module only.

Command format:

$APS,<current_address>,<new_address><CR>

Device response:

$APS,<new_address>,<status>*<checksum><CR>

where:

<current_address> - current device address (range '0' to 'F')
<new_address> - new device address (range '1' to 'F')
<status> - 1 if the command is successful and 0 if the command fails.

Note that loop address '0' is a 'wildcard address i.e. all devices will respond to it. This allows devices to be interrogated and configured outside of a configured Aptiloop environment i.e. when connected via an Aptiloop interface to a P.C. running a terminal emulation application. When the device is attached to a multiple device Aptiloop system address '0' should not be used by the loop cluster controller.

End

Autonnic Research Ltd www.autonnic.com
Tollesbury
CM9 8SE UK