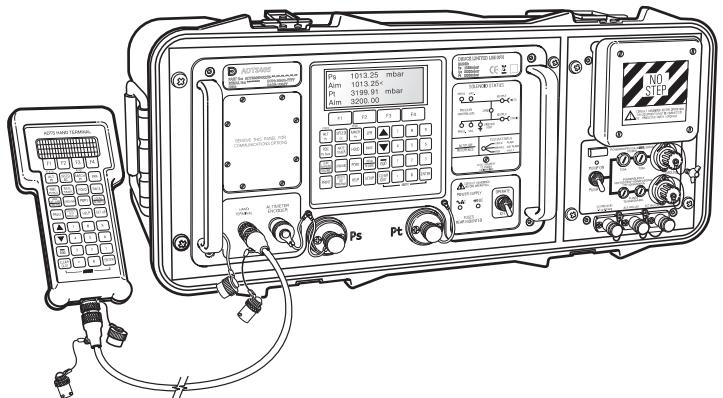
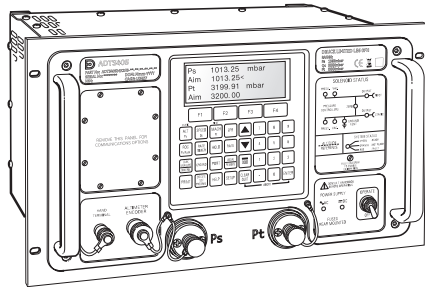


# ADTS405MK2

## Air Data Test Systems Communications Manual





# Introduction

This technical manual provides communication instructions for the Druck ADTS405MK2 Air Data Test System compatible with the requirements of first line operation.

## Scope

This technical manual contains the communications protocol for the operator of this equipment series.

## Software

This technical manual applies to software DK 127/DK 263 version 4.30 onwards or any DK 415.

## Safety

The manufacturer has designed this equipment to be safe when operated using the procedures detailed in this manual. Do not use this equipment for any other purpose than that stated.

This publication contains operating and safety instructions that must be followed to ensure safe operation and to maintain the equipment in a safe condition. The safety instructions are either warnings or cautions issued to protect the user and the equipment from injury or damage.

Use qualified<sup>1</sup> technicians and good engineering practice for all procedures in this publication.

## Pressure

Do not apply pressure greater than the maximum safe working pressure to the equipment.

## Maintenance

The equipment must be maintained using the manufacturer's procedures and should be carried out by authorized service agents or the manufacturer's service departments.

## Technical Advice

For technical advice contact the manufacturer or subsidiary.

---

1. A qualified technician must have the necessary technical knowledge, documentation, special test equipment and tools to carry out the required work on this equipment.

## Associated Documents

| Document | Title   |
|----------|---|
| K0114    | ADTS405 User Manual                                 |
| K0572    | ADTS405MK2 User Manual                              |
| K0154    | ADTS405 Series IEEE 488 Option Manual               |
| K0199    | ADTS405 Calibration Manual                          |
| K0230    | ADTS Test Program Manager Language Reference Manual |

## Abbreviations

The following abbreviations are used in this manual; the abbreviations are the same in the singular and plural.

| Abbreviation | Description  |
|--------------|--|
| A            | Ampere   |
| abs          | Absolute   |
| AC           | Alternating current  |
| ADTS         | Air data test system   |
| ALT          | Altitude   |
| ARINC        | Aeronautical Radio Incorporated                                    |
| ASCII        | American Standard Code for Information Interchange                 |
| ATE          | Automatic test equipment   |
| CAS          | Calibrated airspeed  |
| e.g.         | For example  |
| etc.         | And so on  |
| Fig.         | Figure   |
| ft           | Foot   |
| g            | Gauge  |
| hPa          | Hecto Pascal   |
| Hz           | Hertz  |
| i.e.         | That is  |
| IEEE 488     | Institute of Electrical and Electronic Engineers standard 488 data |
| in           | Inch   |
| inHg         | Inches of mercury  |
| inH2O4       | Inches of water at 4°C   |
| inH2O20      | Inches of water at 20°C  |
| kg           | Kilogram   |
| LSU          | Line Switching Unit  |

| Abbreviation | Description                                    |
|--------------|--|
| m            | Metre  |
| mA           | Milliampere                                    |
| max          | Maximum  |
| mbar         | Millibar                                       |
| min          | Minute or minimum                              |
| mm           | Millimetre                                     |
| mmHg         | Millimetre of mercury                          |
| mV           | Millivolts                                     |
| No.          | Number   |
| Pa           | Pascal   |
| PC           | Personnel computer                             |
| Ps           | Pressure static                                |
| psi          | Pounds per square inch                         |
| Pt           | Pressure Total (Pitot)                         |
| P/W          | Password                                       |
| Qc           | Differential pressure                          |
| SCPI         | Standard commands for programmable instruments |
| TAS          | True airspeed                                  |
| TPM          | Test program manager (Druck software package)  |
| V            | Volts  |
| +ve          | Positive                                       |
| -ve          | Negative                                       |
| °C           | Degrees Celsius                                |
| °F           | Degrees Fahrenheit                             |

## Glossary

The terminology used in this manual is specific and individual interpretation must not be introduced. The terms are defined as follows:

| Item     | Description  |
|----------|--|
| Adjust   | To bring to a more satisfactory state; to manipulate controls, levers, linkages, etc. to return equipment from an out-of-tolerance condition to an in-tolerance condition. |
| Align    | To bring into line; to line up; to bring into precise adjustment, correct relative position or coincidence.  |
| Assemble | To fit and secure together the several parts of; to make or form by combining parts.   |

| Item       | Description  |
|------------|--|
| Calibrate  | To determine accuracy, deviation or variation by special measurement or by comparison with a standard.   |
| Check      | Make a comparison of a measure of time, pressure, temperature, resistance, dimension or other quality with a known figure for that measurement.      |
| Disconnect | To detach the connection between; to separate keyed or matched equipment parts.  |
| Dismantle  | To take apart to the level of the next smaller unit or down to all removable parts.  |
| Examine    | To perform a critical visual observation or check for specific conditions; to test the condition of.   |
| Fit        | Correctly attach one item to another.  |
| Inspect    | Review the work carried out by Specialists to ensure it has been performed satisfactorily.   |
| Install    | To perform operations necessary to properly fit an equipment unit into the next larger assembly or system.   |
| Maintain   | To hold or keep in any particular state or condition especially in a state of efficiency or validity.  |
| Make sure  | To confirm that a proper condition exists; to find out with certainty.   |
| Operate    | Ensure that an item or system functions correctly as far as possible without the use of test equipment or reference to measurement.                  |
| Readjust   | To adjust again; to move back to a specified condition; to bring back to an in-tolerance condition.  |
| Reconnect  | To rejoin or refasten that which has been separated.   |
| Refit      | Fit an item which has previously been removed.   |
| Remove     | To perform operations necessary to take an equipment unit out of the next larger assembly or system. To take off or eliminate. To take or move away. |
| Repair     | To restore damaged, worn out or malfunctioning equipment to a serviceable, usable or operable condition.   |
| Replace    | Remove an item and fit a new or a serviced item.   |
| Reset      | To put back into a desired position, adjustment or condition.  |
| Service    | To perform such operations as cleaning, lubricating and replenishing to prepare for use.   |
| Test       | Ascertain by using the appropriate test equipment that a component or system functions correctly.  |

## Definitions

The following terms are used in this manual.

| Item     | Description   |
|----------|---|
| Bank     | A group of up to eight consecutive valves that can be shown on the screen at the same time. |
| Channel: | All the valves in the LSU that are connected to a common pressure output from the ADTS      |
| Source   | A pressure input to the LSU   |
| Valve    | A pressure output from the LSU  |





# Contents

|     |                                      |    |
|-----|--------------------------------------|----|
| 1.  | Introduction                         | 1  |
| 2.  | Specification                        | 3  |
| 3.  | Electrical Connection                | 5  |
| 4.  | Option Selection and Address Setting | 7  |
| 4.1 | Option Selection                     | 7  |
| 4.2 | IEEE Address Selection               | 7  |
| 5.  | Remote/Local Operation               | 9  |
| 5.1 | Operating Limits                     | 9  |
| 6.  | Command Syntax and Example Program   | 11 |
| 6.1 | Commands to ADTS                     | 11 |
| 6.2 | Command Response from ADTS           | 11 |
| 6.3 | SCPI Data Types                      | 12 |
| 6.4 | Example Program                      | 12 |
| 7.  | Status System                        | 15 |
| 7.1 | Questionable Status Group            | 16 |
| 7.2 | Output Queue                         | 16 |
| 7.3 | Standard Event Group                 | 17 |
| 7.4 | Operation Status Group               | 17 |
| 7.5 | Status Byte Group                    | 18 |
| 8.  | Command and Query Summary            | 21 |
| 8.1 | Command Structure                    | 21 |
| 8.2 | Password                             | 21 |
| 9.  | Command Reference                    | 27 |
| 9.1 | Introduction                         | 27 |
|     | CALCulate                            | 28 |
|     | CALC:ACOR                            | 28 |
|     | CALC:AZER                            | 29 |
|     | CALC:LIM:MAX:PRES                    | 30 |
|     | CALC:LIM:MAX:RATE                    | 31 |
|     | CALC:LIM:MIN:PRES                    | 32 |
|     | CALC:PTT                             | 33 |
|     | CALC:SALT                            | 34 |
|     | CALibrate                            | 35 |
|     | CAL:ABOR                             | 35 |
|     | CAL:ADJ                              | 36 |
|     | CAL:CHEC:CHAN                        | 37 |
|     | CAL:CHEC:ENDP                        | 38 |
|     | CAL:MAIN:ACC                         | 39 |

|                   |    |
|-------------------|----|
| CAL:MAIN:CHAN     | 40 |
| CAL:MAIN:RES      | 41 |
| CAL:MAIN:VAL      | 42 |
| CAL:RATE          | 43 |
| DIAGnostic        | 44 |
| DIAG:ERR          | 44 |
| DIAG:ERR:CLE      | 45 |
| DIAG:ERR:RES      | 46 |
| DIAG:SWIT:ERR     | 47 |
| DIAG:SWIT:ERR:CLE | 48 |
| DIAG:SWIT:ERR:RES | 49 |
| MEASure           | 50 |
| MEAS:AENC         | 50 |
| MEAS:PRES         | 51 |
| MEAS:QFE          | 52 |
| MEAS:QNH          | 53 |
| MEAS:RATE         | 54 |
| MEAS:TRAT         | 55 |
| OUTPput           | 56 |
| OUTP:ALL          | 56 |
| OUTP:CHAN         | 57 |
| OUTP:ISOL:STAT    | 58 |
| OUTP:RES          | 59 |
| OUTP:VALV:CONF    | 60 |
| OUTP:VALV:ERR     | 61 |
| OUTP:VALV:MAX     | 62 |
| OUTP:VALV:NAME    | 63 |
| OUTP:VALV:PRES    | 64 |
| OUTP:VALV:STAT    | 65 |
| SENSe             | 66 |
| SENS:TRAT         | 66 |
| SENS:TRAT:RES     | 67 |
| SENS:TRAT:STAR    | 68 |
| SENS:TRAT:TIME    | 69 |
| SENS:TRAT:WAIT    | 70 |
| SOURce            | 71 |
| SOUR:ARIN:SPEE    | 71 |
| SOUR:ARIN:TRAN    | 72 |
| SOUR:GTGR         | 73 |
| SOUR:LIM:ARINC    | 74 |
| SOUR:LIM:AIRC     | 75 |
| SOUR:LIM:MACH     | 76 |
| SOUR:LIM:MAX:PRES | 77 |
| SOUR:LIM:MAX:RATE | 78 |
| SOUR:LIM:MIN:PRES | 79 |
| SOUR:MODE:ALE     | 80 |
| SOUR:MODE:ALIM    | 81 |
| SOUR:MODE:ARAT    | 82 |
| SOUR:MODE:PSON    | 83 |
| SOUR:MODE:PTON    | 84 |
| SOUR:PRES         | 85 |
| SOUR:RATE         | 86 |
| SOUR:STAT         | 87 |

|  |     |
|--|-----|
| STATus   | 88  |
| STAT:OPER:CON  | 88  |
| STAT:OPER:ENAB   | 89  |
| STAT:OPER:EVEN   | 90  |
| STAT:QUES:CON  | 91  |
| STAT:QUES:ENAB   | 92  |
| STAT:QUES:EVEN   | 93  |
| SYSTem   | 94  |
| SYST:DATE  | 94  |
| SYST:ERR   | 95  |
| SYST:PASS:CDIS   | 96  |
| SYST:PASS:CEN  | 97  |
| SYST:PASS:NEW  | 98  |
| SYST:PASS:STAT   | 99  |
| SYST:PRES  | 100 |
| SYST:TIME  | 101 |
| SYST:VERS  | 102 |
| TEST   | 103 |
| TEST:SWIT:ACT  | 103 |
| TEST:SWIT:PASS   | 104 |
| UNITs  | 105 |
| UNIT:AER   | 105 |
| UNIT:PRES  | 106 |
| UNIT:TEMP  | 107 |
| *Standard Commands                                       | 108 |
| *CLS   | 108 |
| *DTM   | 109 |
| *ESE   | 110 |
| *ESR   | 111 |
| *ID2?  | 112 |
| *IDN   | 113 |
| *NET   | 114 |
| *OPC   | 115 |
| *OPT   | 116 |
| *RST   | 117 |
| *SRE   | 118 |
| *STB   | 119 |
| *TST   | 120 |
| *TTO   | 121 |
| *WAI   | 122 |
| <br>   |     |
| 10. Errors   | 123 |
| 10.1 Error Numbers -499 to -400 (Query Errors)           | 123 |
| 10.2 Error Numbers -399 to -300 (Device Specific Errors) | 123 |
| 10.3 Error Numbers -299 to -200 (Execution Errors)       | 123 |
| 10.4 Error Numbers -199 to -100 (Command Errors)         | 124 |
| <br>   |     |
| 11. Calibration  | 125 |
| 11.1 Calibration using SCPI                              | 125 |
| 11.2 Single Channel Calibration (PS or PT)               | 125 |
| 11.3 Dual Channel Calibration                            | 125 |



# Tables

| <b>Table</b> |                                 | <b>Page</b> |
|--------------|---------------------------------|-------------|
| 2-1          | IEEE 488 Codes                  | 3           |
| 3-1          | Connector Details               | 5           |
| 7-1          | Questionable Condition Register | 16          |
| 7-2          | Standard Event Register         | 17          |
| 7-3          | Operation Status Register       | 18          |
| 7-4          | Status Byte Register            | 19          |
| 10-1         | Query Errors                    | 123         |
| 10-2         | Device Specific Errors          | 123         |
| 10-3         | Execution Errors                | 123         |
| 10-4         | Command Errors                  | 124         |



# Figures

| <b>Figure</b> |                        | <b>Page</b> |
|---------------|------------------------|-------------|
| 1-1           | System Model           | 1           |
| 6-1           | General Command Syntax | 11          |
| 6-2           | Command Syntax         | 11          |
| 6-3           | Response Syntax        | 11          |
| 7-1           | Status System          | 15          |





# 1. Introduction

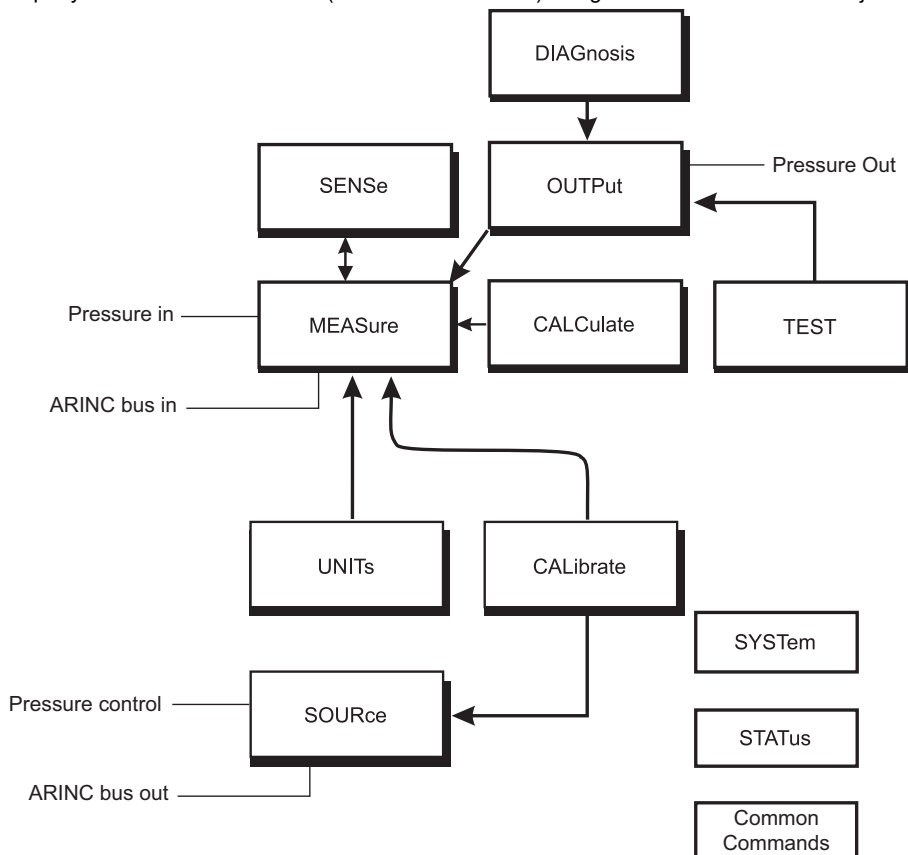
The SCPI IEEE 488 interface provides remote control of the ADTS from Automatic Test Equipment (ATE), using standardized commands.

SCPI (pronounced “skippy”) stands for Standard Commands for Programmable Instruments. The concept is that all instruments with a SCPI facility can be communicated with using the same commands. The number of instruments complying with the standard is continually growing although there are only a few pressure instruments.

The ADTS405MK2 implements the full SCPI command set with the exception of expressions and units suffixes. Although all the ADTS405MK2 commands at the first, or root, level are fully defined in the SCPI standard, many of the functions required have not been allowed for in the standard. The lower levels of commands have been created using the appropriate rules contained within the standard.

Information is given for each command where it specifically relates to the SCPI implementation of the command.

The book A beginners Guide To SCPI by Barry Eppler, Published by Addison-Wesley Publishing Company Inc. for Hewlett Packard (ISBN 0-201-56350-9) is a good introduction to the subject.



**Figure 1-1: System Model**



---

## 2. Specification

The SCPI IEEE 488 interface conforms to IEEE 488.1 with the following functions implemented.

**Table 2-1: IEEE 488 Codes**

| Function             | Code | Capability   |
|----------------------|------|--|
| Source Handshake     | SH1  | Complete   |
|                      | AH1  | Complete   |
| Talker Functions     | T6   | Basic talker + serial poll + unaddressed by MLA (no talk only mode). |
|                      | TE0  | No extended talk address.  |
| Listener Functions   | L4   | Basic listener + unaddressed by MTA (no listener only mode).         |
|                      | LE0  | No extended listen address.  |
| Service Request      | SR1  | Complete   |
| Remote Local         | RL1  | Complete   |
| Parallel Poll        | PP0  | No parallel poll.  |
| Device Clear         | DC2  | Universal device clear only (no selective device clear SDC).         |
| Device Trigger       | DT0  | No device trigger.   |
| Controller functions | C0   | No controller.   |



---

### 3. Electrical Connection



**CAUTION** The option connector cover must always be fitted when the equipment is used outdoors or in an environment where moisture may enter the unit via the option connector panel.

The IEEE 488 option allows connection to either the front or rear of the ADTS using the standard Amphenol connector. The front connector is behind the option connector cover plate.

The connector is as follows:

**Table 3-1: Connector Details**

| Standard Connector Pin Number | Function            | Standard Connector Pin Number | Function         |
|-------------------------------|---------------------|-------------------------------|------------------|
| 1                             | D101                | 13                            | D105             |
| 2                             | D102                | 14                            | D106             |
| 3                             | D103                | 15                            | D107             |
| 4                             | D104                | 16                            | D108             |
| 5                             | E01                 | 17                            | REN              |
| 6                             | DAV                 | 18                            | Gnd <sup>a</sup> |
| 7                             | NRFD                | 19                            | Gnd <sup>a</sup> |
| 8                             | NDAC                | 20                            | Gnd <sup>a</sup> |
| 9                             | IFC                 | 21                            | Gnd <sup>a</sup> |
| 10                            | SRQ                 | 22                            | Gnd <sup>a</sup> |
| 11                            | ATN                 | 23                            | Gnd <sup>a</sup> |
| 12                            | Shield <sup>b</sup> | 24                            | Gnd <sup>a</sup> |

- a. 0V return for signals (not chassis).
- b. Chassis connection.



## 4. Option Selection and Address Setting

The ADTS405MK2 IEEE 488 interface is available with alternative languages for communication. The low level IEEE 488 commands are the same in each case, it is the words or commands that are different. The SCPI language is to the latest standards and should be used for any new ATE systems being developed or for any existing systems based on SCPI.

The alternative languages offer compatibility with existing systems.

The same IEEE 488 option card is used for all of the options.

### 4.1 Option Selection

To select which option to use, first select CONFIG by holding down F1 while pressing SETUP.

The display should show:

```
(CONFIG)
F1 Units
F2 Limits
F3 More
```

Select [MORE], then [DISPLAY/OPTION], then [OPTIONS].

Select [IEEE 488], then [SCPI/OPT2] and then select [SCPI] if not already selected.

Following the selection, check that the IEEE address is correct

### 4.2 IEEE Address Selection

The IEEE device address can be set to any address from 0 to 30.

To set the address, select CONFIG, [MORE], [DISPLAYS/OPTIONS], [OPTIONS], [IEEE 488], [DEVICE ADDR]. Enter an address between 0 and 30.

When IEEE parameters have been changed, press QUIT. The ADTS will prompt for acceptance of the changes (and re-boot the system) or restore the previous values. If everything is set correctly, press ENTER, otherwise press QUIT.





## 5. Remote/Local Operation

Any command received over the SCPI interface puts the ADTS into REMOTE mode. In REMOTE mode, all local keys are disabled except LOCAL/REMOTE and ABORT. If the LOCAL/REMOTE key is pressed when in REMOTE, the ADTS returns to LOCAL. If the IEEE 488 command GTL (Go To Local) is sent to the ADTS, the ADTS returns to LOCAL.

If the IEEE 488 command LLO (Local Lock-out) is sent to the ADTS, the LOCAL/REMOTE key will also be disabled. This leaves the ADTS locked in REMOTE until power-down.

When the ADTS is in REMOTE, the word "REMOTE" appears on the display, the display shows the state of the ADTS and updates pressure readings. The parameters and units used for the display only change when a SOURce:PRESSure or SOURce:RATE command is received (new pressure or rate aim). Any parameters or units can be used over the SCPI interface, independent of the parameters and units shown on the display.

To prevent accidental selection of REMOTE and change the operating limits, the REMOTE LOCK must be disabled before selecting remote mode from the key-pad as follows:

```
[CONFIG]
└─ [REMOTE/LOCAL]
    └─ [ON]
        └─ [OFF]
```

ON Turns on remote lock.  
OFF Turns off remote lock.

The display shows an error message when in local control with remote lock enabled and the REMOTE/LOCAL key pressed.

If a SCPI command is received from the GPIB the unit goes into remote mode, even if remote lock is enabled. The remote/local key can be used to return to local control.

### 5.1 Operating Limits

When the unit goes into remote mode, a set of limits become current; these are the previously stored set or a new set of limits called \*SCPI\*. The values of these limits match the maximum operating limits of the unit. These values can be changed using the SOUR:LIM command, copied from a standard limit set with SOUR:LIM:AIRC. The CALC:LIM commands query the current limits.

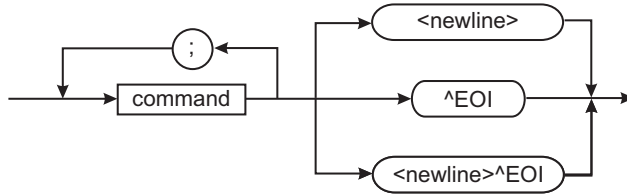
**Note:** \*SCPI\* limits are volatile, also if the unit returns to local control the limits can be replaced with a new set from local control.



## 6. Command Syntax and Example Program

### 6.1 Commands to ADTS

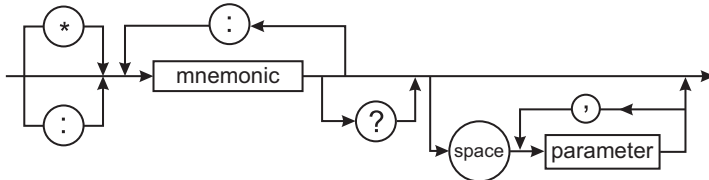
The general syntax for a complete message is shown below:



**Figure 6-1: General Command Syntax**

When sending more than one command in the same message, each command must be separated by a semicolon. One of the three terminators shown must be used to end the message i.e., either <newline> (ASCII char decimal 10), EOI asserted with last byte or EOI asserted with <newline>.

The syntax for each command is shown below:



**Figure 6-2: Command Syntax**

Common commands all start with a star (see list of commands).

Example: \*ESE

Each command word or mnemonic must be separated by a colon.

Example: SOURCE:MODE:PTONLY ON

There must be a space between the last command mnemonic and a parameter.

Example: UNIT:PRES mbar

All characters can be upper or lower case. Each command mnemonic can be either short form (shown in upper case in command list) or in long form, NOT in between.

If more than one parameter is sent, they must be separated by commas.

Example: SOUR:PRES alt,20000

If it is a query command, a question mark must follow the command, before any parameters.

Example: MEAS:PRES? alt

A maximum of sixteen, 100 character, messages may be sent at once. Each message may contain any number of semicolon (;) separated commands.

### 6.2 Command Response from ADTS

The syntax for response messages is shown below:



**Figure 6-3: Response Syntax**

## Chapter 6. Command Syntax and Example Program

---

When more than one value is returned, a comma is used as a separator.

**Note:** <newline> with EOI is always used to terminate the data.

### 6.3 SCPI Data Types

There are various types of data that can be sent to the ADTS as parameters or sent out from the ADTS as response data.

#### 6.3.1 Numeric

Numeric data may be:

1. Real
2. 8 bit Integer (0 to 255), or
3. 16 bit Integer (0 to 65535)

All commonly used decimal representations are accepted including optional signs, decimal points and scientific notation. The following are all valid:

- 100
- 100.
- -1.23
- 4.56e3
- -7.89E-01
- +256
- .5

If a real value is sent to the ADTS when an integer is expected, it will be rounded to an integer.

#### 6.3.2 Discrete

Discrete parameters can be used to select settings that have a finite number of values. They can be upper or lower case. Parameters used in the ADTS405MK2 must be specified exactly, there is no short form.

Parameters to be measured can be selected from ALT, CAS, MACH, PS, Qc etc.

Example: MEAS:PRES PS

#### 6.3.3 Boolean

Boolean parameters are similar to discrete but can only be ON or OFF.

Example: CALC:AZERO ON

Response from a query command will be ON or OFF (not 1 or 0).

##### 6.3.3.1 String

String parameters are only used in response to the query SYST:ERR? where text describing the error is appended to the error number.

Example: -222, "Out of Range".0

## 6.4 Example Program

The following program example is written in QUICK BASIC and illustrates the use of various commands and responses.

### 6.4.1 Program Overview

1. Initialize the IEEE 488 interface.
2. Initialize the ADTS405MK2 instrument and check for errors.
3. Set units as mbar and go to CONTROL.

4. Enter Ps and Qc rates of change as 200 mbar/min.
5. Enter Ps AIM of 800 mbar and Qc AIM of 220 mbar.
6. Wait for pressure to stabilize at AIM values.
7. Go to MEASURE mode.
8. Program a WAIT time of 1 minute for leak test.
9. Program a TIME of 30 sec for leak test.
10. Start leak test of WAIT 1 minute, time over 30 sec.
11. Wait for leak test to complete.
12. Display Ps leak rate.
13. Display final Ps value.
14. Go to CONTROL mode.
15. Go to ground.

Commands are sent out using a sub-routine that checks for errors after sending the command out.

Response data is obtained using a function that sends out the query command and returns the response. An error check is performed after obtaining the response.

The example program illustrates the use of both long and short form commands and upper and lower case.

The IEEE 488 commands vary, depending on which interface PCB is used in the PC or ATE system.

#### 6.4.2 Example Program

```

DECLARE FUNCTION GetResponse$ (a$)
DECLARE SUB SendCommand (a$)

'Open channels to the IEEE 488 interface for control and data.
OPEN "ieelectrl" FOR OUTPUT AS #1
OPEN "ieeedata" FOR OUTPUT AS #2

    'Set up interface as:
    'Address 1
    'Interface clear
    'Time out 10 secs
    'EOI with last byte out
    'Input terminated by EOI
IOCTL #1, "1 ABORT TO 13 REN END=ON EOS="

    'Initial set-up. Clear status register and check for errors.
IOCTL #2, "*CLS"
IOCTL #2, "SYST:ERR?"
PRINT "Error check reports - "; IOCTL$(2)

    'Set units and goto control mode.
SendCommand ("UNITS:PRESSURE mbar")
SendCommand ("SOURCE:STATE control")
Delay (10) 'Wait for ADTS to regain control.

    'Enter rate and pressure setpoints.
SendCommand ("SOURCE:RATE PS,200;RATE QC,500")
SendCommand ("SOUR:PRES ps,800;PRES QC,220")

    'Wait for pressures to stabilize at setpoint.
    'Look for bit 1 to be set in Status Operation Condition register.
DO
    Response$ = GetResponse$("STAT:OPERATION:CONDITION?")
    Status = VAL(response$)
LOOP UNTIL (status AND 2)

    'Go to measure mode.

```

## Chapter 6. Command Syntax and Example Program

---

```
SendCommand ("SOUR:STAT MEASURE")

    'Program WAIT and TIME times and start timing leak rate.
SendCommand ("SENSE:TRATE:WAIT 1,0")
SendCommand ("SENSE:TRATE:TIME 0,30")
SendCommand ("SENSE:TRATE:START")

    'Wait for timing to be completed.
DO
    Status$ = GetResponse$("SENSE:TRATE?")
LOOP UNTIL LEFT$(status$, 5) = "TIMED"

    'Get timed rate of change of Ps and display it.
Leakrate$ = GetResponse$("MEAS:TRATE? ps")
PRINT "Ps leak rate (mbar/min) = "; LeakRate$

    'Get Ps value and display it.
Response$ = GetResponse$("MEAS:PRES? ps")
PRINT "Final Ps pressure (mbar) = "; Response$

    'Go to control mode and then go to ground.
SendCommand ("SOURCE:STATE control")
Delay (3) 'Wait for ADTS to regain control
SendCommand ("SOUR:GTGR")

    'Wait until at ground. Controllers will be turned off.
DO
    Response$ = GetResponse$("STAT:OPER:CON?")
LOOP UNTIL (VAL(Response$) AND 4)

PRINT "END OF TEST"

FUNCTION GetResponse$ (a$)
    'This function sends the string passed to it (a$), and returns the response
    string.
    'A check is made for errors after getting the response.

    'Send string.
IOCTL #2, a$

    'Get response.
r$ = IOCTL$(2)

    'Check for errors and display if none zero.
    'Error 0 returned if "no errors"
IOCTL #2, "SYST:ERR?"
e$ = IOCTL$(2)
IF VAL(e$) <> 0 THEN
    PRINT "Error reported when sending "; a$; ""
    PRINT e$
END IF

    'Return response string.
GetResponse = r$
END FUNCTION

SUB SendCommand (a$)
    'This SUB sends the string passed to it (a$) and then checks for errors.

    'Send string.
IOCTL #2, a$

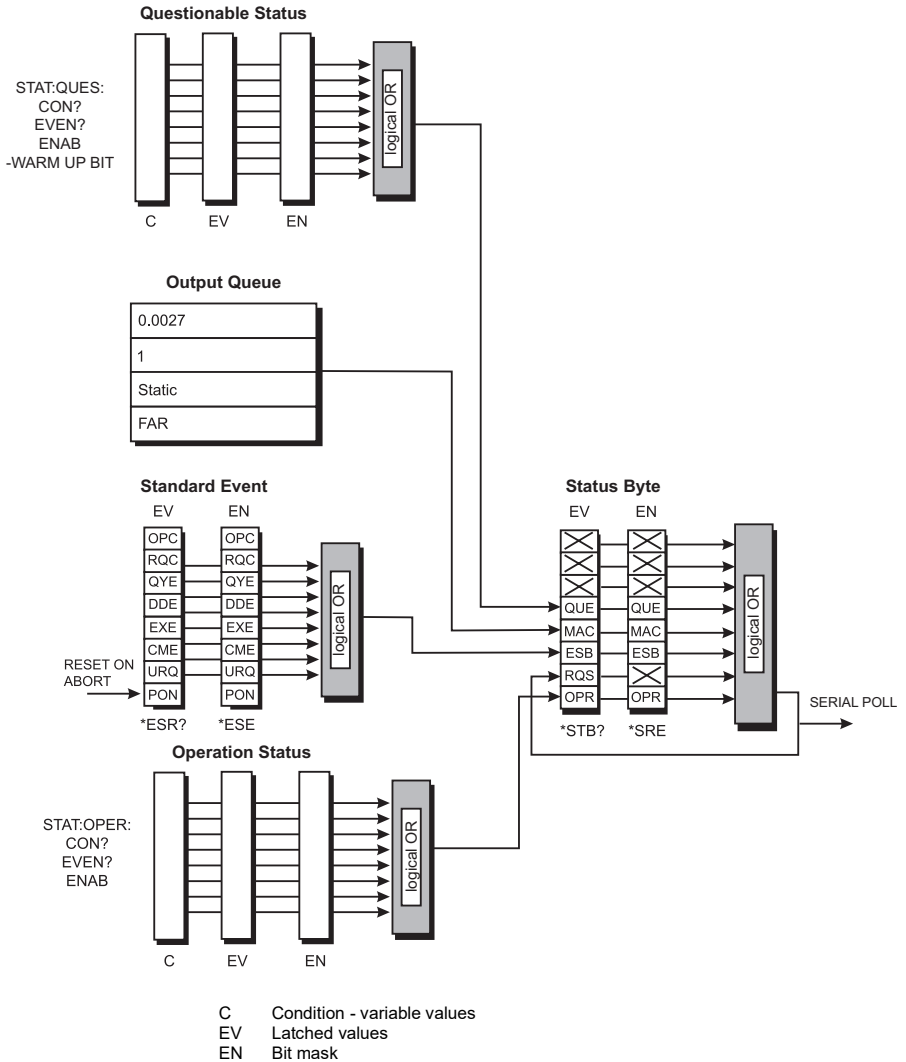
    'Check for errors and display if none-zero.
    'Error 0 is returned if "no error".
IOCTL #2, "SYST:ERR?"
e$ = IOCTL$(2)
IF VAL(e$) <> 0 THEN
    PRINT "Error reported, when sending "; a$
    PRINT "Error -"; e$
END IF

END SUB
```

# 7. Status System

The status system as implemented in the ADTS405MK2 is shown the following diagram:

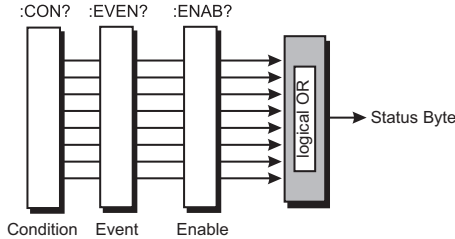
**Note:** Initial values of registers are 0, with the queues empty.



**Figure 7-1: Status System**

### 7.1 Questionable Status Group

The questionable data group are 16 bit registers read through the STAT:QUES commands. The event register is cleared by reading it; the event and enable registers are cleared by the \*CLS command.



When a questionable condition occurs an appropriate bit is set in the condition register (this clears when the condition no longer exists). The bit is then latched in the event register. If the associated bit in the enable register is set, the QUE bit in the status byte sets. The enable register may be set through the STAT:QUES:ENAB command so that only selected questionable events cause the QUE bit to set.

Problems can occur with some IEEE controllers reading 16 bit unsigned numbers. All registers in this group do not use bit 15. The enable bit cannot be set and when read returns 0.

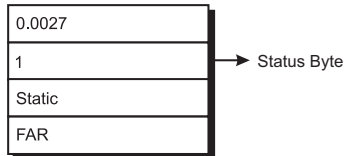
The questionable condition register is defined as follows:

**Table 7-1: Questionable Condition Register**

| Bit   | Description  |
|-------|--|
| 0-7   | Reserved   |
| 8     | Cal (Auto zero) in progress.   |
| 9     | Warm-up. Bit 9 is set to 1 during the warm-up period following power-on with the display showing "WARMUP". After the warm-up period bit 9 changes to 0 (zero). |
| 10-15 | Reserved   |

### 7.2 Output Queue

The output queue is a text readable data queue that is read via the IEEE 488 talk command. The queue is cleared by reading all elements in it or by the \*CLS command.

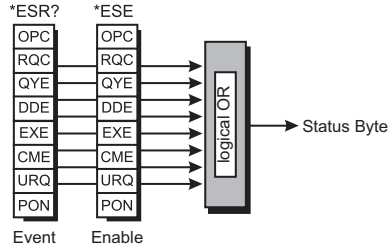


Every time a query has been successfully completed, the response, in a text readable format is placed at the end of the output queue. If the MAV bit in the "Status Byte" was previously cleared it will be set. The output queue can contain up to 256 characters. If there is not enough space in the output queue for a new message, the error -350, "Queue overflow" will be placed into the error queue and the most recent output message will be lost.



### 7.3 Standard Event Group

The standard event group are 8 bit registers that are read by the IEEE 488 standard commands. The event register is cleared by reading it; the event and enable registers are cleared by the \*CLS command.



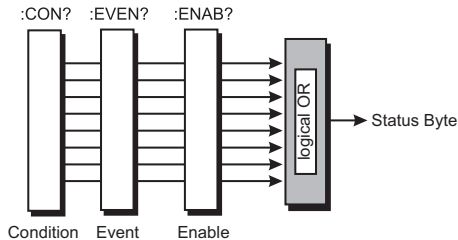
Bits within the standard event condition register are set by system errors and events. In addition to setting the status bits, a text message will be placed in the error/event queue. The ESB bit in the status byte sets if the associated bit in the event enable register is set. The enable register may be set via the \*ESE command so that selected standard events cause the ESB bit to be set. The system events that set each bit are as follows:

**Table 7-2: Standard Event Register**

| Bit | Name | Error/Event Number Range | Meaning/data                  |
|-----|------|--------------------------|-------------------------------|
| 0   | OPC  | Not used                 | Reserved currently returns 0. |
| 1   | RQC  | Not used                 | Reserved currently returns 0. |
| 2   | QYE  | -400 to -499             | Query errors.                 |
| 3   | DDE  | Not used                 | Reserved currently returns 0. |
| 4   | EXE  | -200 to -299             | Execution errors.             |
| 5   | CME  | -100 to -199             | Command errors.               |
| 6   | URQ  | Not used                 | Reserved currently returns 0. |
| 7   | PON  | -500 to -599             | Power on event.               |

### 7.4 Operation Status Group

The operation status group are 16 bit registers that are read by the STAT:OPER commands. The event register is cleared by reading it; the event and enable registers are cleared by the \*CLS command.



When a standard operation condition occurs an appropriate bit is set in the condition register (this clears when the condition no longer exists). The bit is then latched in the event register. If the associated bit in the enable register is set, the OPR bit in the status byte sets. The enable register

## Chapter 7. Status System

may be set through the STAT:OPER:ENAB command so that only selected standard operation events cause the OPR bit to set.

The condition register is defined as follows:

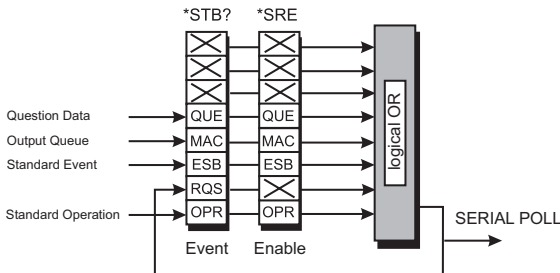
**Table 7-3: Operation Status Register**

| Bit | Data   |
|-----|--|
| 0   | Reserved - returns 0.  |
| 1   | Stable at aim value <sup>a</sup>   |
| 2   | Safe at ground.  |
| 3   | Ramping <sup>b</sup>   |
| 4   | Reserved - returns 0.  |
| 5   | Reserved - returns 0.  |
| 6   | Reserved - returns 0.  |
| 7   | Reserved - returns 0.  |
| 8   | Ps at set-point and in control mode <sup>c</sup>                         |
| 9   | Ps ramping and achieving rate.   |
| 10  | Pt at set-point and in control mode <sup>c</sup>                         |
| 11  | Pt ramping and achieving rate.   |
| 12  | Reserved - returns 0.  |
| 13  | Reserved - returns 0.  |
| 14  | Perform Calibrate - Set if calibrate in progress (DK 415 software only). |
| 15  | Calibrate Enable - Set if calibrate enabled (DK 415 software only).      |

- Set after 15 seconds (to give pressure time to settle) after both bits 10 and 8 (if not in Pt only mode) become set.
- This is set to 1 if bit 9 is set and bit 11 is set.
- Set immediately the aim set-point is reached. Bit 1 should generally be used to indicate that all pressures are stable before taking readings, rather than bits 8 or 10 as bit 1 includes stabilization time.

### 7.5 Status Byte Group

The status byte group are 8 bit registers that are read by the IEEE 488 standard commands. The event register is cleared by reading it; the event and enable registers are cleared by the \*CLS command.



Bits within the status byte are a summary of other data structures in the status system. These bits will become set if other parts of the status system indicates that they should do so (i.e., a message in the output or error queue or a condition and enable set in a register pair).

If the associated bit in the status enable register is set, a serial poll is generated and bit 6 is set. The enable register may be set via the \*SRE command so that only selected status bits cause a serial poll.

**Note:** Bit 6 of the enable register is always set to 0.

There are some small differences between \*STB? and serial polling. Either method can be used to read the state of bits 0-5 and bit 7. The reading method is different for bit 6 when using \*STB? and serial poll. In general, use serial polling inside interrupt service routines, not \*STB?. The status system data structure sets each bit as follows:

**Table 7-4: Status Byte Register**

| Bit | Name | Description                                 |
|-----|------|---|
| 0   | -    | Reserved - return 0.                        |
| 1   | -    | Reserved - return 0.                        |
| 2   | -    | Reserved - return 0.                        |
| 3   | QUE  | Summary bit from questionable data.         |
| 4   | MAV  | Messages available in output queue.         |
| 5   | ESB  | Summary bit from standard event.            |
| 6   | RQS  | Service request.                            |
| 7   | OPR  | Summary bit from standard operation status. |

Example commands using the Status Byte and Status Byte Enable registers:

| Command | Description  |
|---------|--|
| *SRE 16 | Generate an RQS interrupt when messages are available.       |
| *SRE?   | Find out what events are enabled to generate RQS interrupts. |
| *STB?   | Read and clear the Status Byte Event register.               |



## 8. Command and Query Summary

The following lists of all the SCPI commands and queries that apply to the ADTS405MK2.

### 8.1 Command Structure

Some of the following commands are enabled at specific times and conditions, most can be enabled at any time. The parameters column shows the states, values and data contained in a command. Some commands are further defined in the following with a '}' comment.

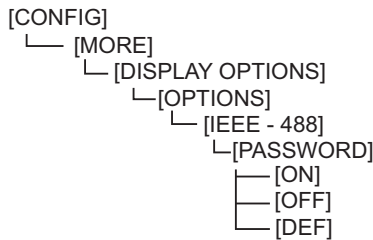
### 8.2 Password

Some of the commands are password protected, in the P/W column the "Y" symbol indicates password protection the "N" symbol indicates no password protection.

#### 8.2.1 Notes on Passwords

These notes apply to software DK 127/DK 263 version 4.30 onwards or any DK 415; before this software version password protection was not available. Some commands are password protected, i.e. if executed with password protection enabled, error -203 is produced.

- Queries are not password protected, for example:  
 SOUR:LIM:MAX:PRES alt? always returns MAX ALT LIMIT  
 SOUR:LIM:MAX:PRES alt, 3000 only sets ALT LIMIT if passwords are protected.
- The SCPI password and its enable/disable state are the only changeable SCPI values that are saved in an internal EEPROM these are stored after switching off the unit.
- Passwords are a maximum of 32 characters long and contain only alphanumeric characters (numbers 0 to 9 and letters A to Z). Letters are case sensitive.
- Factory defaults for the password are:  
 Password: "DEFAULT"  
 Enabled/disabled: DISABLED
- In accordance with SCPI protocols the password and protection status can be cleared locally:



ON Turn password protection on.  
 OFF Turn password protection off.  
 DEF Reset password to "DEFAULT".

| Command       | P/W | Parameters | Comments        |
|---------------|-----|------------|-----------------|
| CALCulate     |     |            |                 |
| :ACORrection  | N   | <value>    | } As ADTS405MK2 |
| :ACORrection? | N   |            |                 |
| :AZERo        | N   | <state>    |                 |

## Chapter 8. Command and Query Summary

| Command     | P/W | Parameters   | Comments                               |
|-------------|-----|--|--|
| :AZERo?     | N   |  |  |
| :LIMit      | N   |  |  |
| :MAXimum    |     |  |  |
| :PRESsure?  | N   | <parameter>  |  |
| :RATE?      | N   | <parameter>  |  |
| :MINimum    |     |  |  |
| :PRESsure?  | N   | <parameter>  |  |
| :PTTemp     | Y   | <value>  | } See user manual                      |
| :PTTemp?    | N   |  |  |
| :SALTitude  | Y   | <value>  | } See user manual                      |
| :SALTitude? | N   |  |  |
| CALibrate   |     |  |  |
| :ABORt      | Y   |  |  |
| :ADJust     | Y   | <channel>,<span>,<zero>,<res(0)>,<res(1)>,...<res(11)> |  |
| :ADJust?    | N   | <channel>  |  |
| :CHECK      |     |  |  |
| :CHANnel    | Y   | <channel>  |  |
| :ENDPs      | Y   |  |  |
| :MAIN       |     |  |  |
| :ACcept     | Y   | <state>  |  |
| :CHANnel    | Y   | <channel>  |  |
| :RESult?    | N   | <slope> <zero>   |  |
| :VALue      | Y   | <value>  | } Pressure range of ADTS405MK2 channel |
| :RATE       | Y   |  |  |
| :RATE?      | N   |  |  |
| DIAGnostic  |     |  |  |
| :ERRor?     | N   |  |  |
| :ERRor      |     |  |  |
| :CLEar      | Y   |  |  |
| :RESet      | N   |  |  |
| :SWITcher   |     |  |  |
| :ERRor?     | Y   |  |  |
| :ERRor      |     |  |  |
| :CLEar      | N   |  |  |
| :RESet      | N   |  |  |

| Command    | P/W | Parameters        | Comments     |
|------------|-----|-------------------|--------------|
| MEASure    |     |                   |              |
| :AENCoder? | N   |                   |              |
| :PRESSure? | N   | <parameter>       |              |
| :QFE?      | N   |                   |              |
| :QNH?      | N   |                   |              |
| :RATE?     | N   | <parameter>       |              |
| :TRATe?    | N   | <parameter>       |              |
| OUTPut     |     |                   |              |
| :ALL       | N   | <state>           |              |
| :CHANnel   | N   | <source>, <state> |              |
| :ISOLation |     |                   |              |
| :STATe     | N   | <state>           |              |
| :STATe?    | N   |                   |              |
| :RESet     | N   |                   |              |
| :VALVe     |     |                   |              |
| :CONFig?   | N   | <valve>           |              |
| :ERRor?    | N   | <valve>           |              |
| :MAXimum?  | N   |                   |              |
| :NAME?     | N   | <valve>           |              |
| :PRESSure? | N   | <value>           |              |
| :STATe     | N   | <valve>, <state>  |              |
| :STATe?    | N   | <valve>           |              |
| SENSe      |     |                   |              |
| :TRATe?    | N   |                   |              |
| :TRATe     | N   |                   |              |
| :RESet     | N   |                   |              |
| :START     | N   |                   |              |
| :TIME      | N   | <min>, <sec>      | } 0-59, 0-59 |
| :TIME?     | N   |                   | } 0-59, 0-59 |
| :WAIT      | N   | <min>, <sec>      | } 0-59, 0-59 |
| :WAIT?     | N   |                   | } 0-59, 0-59 |
| SOURce     |     |                   |              |
| :ARINc     |     |                   |              |
| :SPEEd     | Y   | <parameter>       |              |
| :SPEEd?    | N   |                   |              |
| :TRANsmit  | Y   | <parameter>       |              |
| :TRANsmit? | N   |                   |              |

## Chapter 8. Command and Query Summary

| Command     | P/W | Parameters           | Comments           |
|-------------|-----|----------------------|--------------------|
| :GTGRound   | N   |                      |                    |
| :GTGRound?  | N   |                      |                    |
| :LIMit      |     |                      |                    |
| :ARINc?     | N   |                      |                    |
| :AIRCraft   | Y   | <limits>             | STANDARD MAX ARINC |
| :MACH       | Y   | <parameter>          |                    |
| :MACH?      | N   |                      |                    |
| :MAXimum    |     |                      |                    |
| :PRESSure   | Y   | <parameter>, <value> |                    |
| :PRESSure?  | N   | <parameter>          |                    |
| :RATE       | Y   | <parameter>, <value> |                    |
| :RATE?      | N   | <parameter>          |                    |
| :MINimum    |     |                      |                    |
| :PRESSure   | Y   | <parameter>, <value> |                    |
| :PRESSure?  | N   | <parameter>          |                    |
| :MODE       |     |                      |                    |
| :ALEak      | Y   | <state>              |                    |
| :ALEak?     | N   |                      |                    |
| :ALIMit     | Y   | <state>              |                    |
| :ALIMit?    | N   |                      |                    |
| :ARATe      | Y   | <state>              |                    |
| :ARATe?     | N   |                      |                    |
| :PTONly     | Y   | <state>              |                    |
| :PTONly?    | N   |                      |                    |
| :PSONly     | Y   | <state>              |                    |
| :PSONly?    | N   |                      |                    |
| :PRESSure   | N   | <parameter>, <aim>   |                    |
| :PRESSure?  | N   | <parameter>          |                    |
| :RATE       | N   | <parameter>, <aim>   |                    |
| :RATE?      | N   | <parameter>          |                    |
| :STATe      | N   | <state>              |                    |
| :STATe?     | N   |                      |                    |
| STATus      |     |                      |                    |
| :OPERation  |     |                      |                    |
| :CONDition? | N   |                      |                    |
| :ENABle     | N   | <data>               |                    |
| :ENABle?    | N   |                      |                    |



| Command            | P/W | Parameters                 | Comments                    |
|--------------------|-----|----------------------------|-----------------------------|
| :EVENT?            | N   |                            |                             |
| :QUESTIONable      |     |                            |                             |
| :CONDITION?        | N   |                            |                             |
| :ENABLE            | N   | <data>                     |                             |
| :ENABLE?           | N   |                            |                             |
| :EVENT?            | N   |                            |                             |
| SYSTEM             |     |                            |                             |
| :DATE              | Y   | <YYYY>, <mm>, <dd>         | } 1993+, 1 to 12, 1 to 31   |
| :DATE?             | N   |                            |                             |
| :ERROR?            | N   | <error number>, <err test> |                             |
| :PASSWORD          |     |                            |                             |
| :CDISable          | N   | <password>                 |                             |
| :CENable           | N   | <password>                 |                             |
| :NEW               | N   | <current>, <new>           |                             |
| :STATE?            | N   |                            |                             |
| :PRES              | Y   |                            |                             |
| :TIME              | Y   | <hh>, <mm>, <ss>           | } 0 to 23, 0 to 59, 0 to 59 |
| :TIME?             | N   |                            |                             |
| :VERSION?          | N   | <version>                  |                             |
| TEST               |     |                            |                             |
| :SWITCher          |     |                            |                             |
| :ACTIVE?           | N   |                            | } active self-test          |
| :PASSive?          | N   |                            | } passive self-test         |
| UNITs              |     |                            |                             |
| :AERonautical      | N   | <units>                    |                             |
| :AERonautical?     | N   |                            |                             |
| :PRESsure          | N   | <units>                    |                             |
| :PRESsure?         | N   |                            |                             |
| :TEMPerature       | N   | <units>                    |                             |
| :TEMPerature?      | N   |                            |                             |
| *CLS               | N   |                            |                             |
| *DTM               | N   |                            |                             |
| *ESE               | N   | <data>                     |                             |
| *ESE?              | N   |                            |                             |
| *ESR?              | N   |                            |                             |
| *ID2? <sup>a</sup> | N   |                            |                             |
| *IDN?              | N   |                            |                             |

## Chapter 8. Command and Query Summary

---

| Command           | P/W | Parameters | Comments |
|-------------------|-----|------------|----------|
| *NET <sup>a</sup> | N   |            |          |
| *OPC              | N   |            |          |
| *OPC?             | N   |            |          |
| *OPT?             | N   |            |          |
| *RST              | N   |            |          |
| *SRE              | N   | <data>     |          |
| *SRE?             | N   |            |          |
| *STB?             | N   |            |          |
| *TST?             | N   |            |          |
| *TTO <sup>a</sup> | N   |            |          |
| *WAI              | N   |            |          |

a. DK 415 software only

## 9. Command Reference

### 9.1 Introduction

This section describes each command and query in detail including parameters used and response data returned. The general short form command is shown at the top of each page. The following information given:

| Command         |  |
|-----------------|--|
| Command Syntax  | The upper case represents the short form command.  |
| Parameter Type  | REAL, INTEGER, DISCRETE or STRING  |
| Parameter Range | Either the range of INTEGER/REAL numbers or the choice of DISCRETE or maximum STRING length. |
| Units           | The units used for some specified parameters.  |
| Password        | Password protects some commands; software DK 0127/DK 263 version 4.30 onwards or any DK 415. |
| Function        | Basic function of the command, see user manual for full description of the function.         |
| Conditions      | Any condition that limits the use of a command.  |

| Query               |   |
|---------------------|---|
| Query Syntax        | Syntax of query command includes parameters passed as part of the query.                      |
| Parameter Type      | As for command (above).   |
| Parameter Range     | As for command (above).   |
| Returned Data       | Data returned by the ADTS following a query command (when told to talk, refer to IEEE 488.1). |
| Returned Data Type  | As parameter type.  |
| Returned Data Range | As parameter range.   |
| Units               | Units of returned data.   |
| Function            | Basic function of query command.  |
| Conditions          | As for command.   |

## CALCulate

This group of commands is related to functions that modify the measured values.

### CALC:ACOR

#### Command Syntax

#### **CALCulate:ACORrection <value>**

|                  |   |
|------------------|---|
| Parameter Type:  | REAL                                      |
| Parameter Range: | As ADTS                                   |
| Units:           | As set by UNIT:AER                        |
| Password:        | No  |
| Function:        | To program the altitude correction value. |
| Conditions:      | Must be in measure mode.                  |

#### Query Syntax

#### **CALCulate:ACORrection?**

|                      |   |
|----------------------|---|
| Parameter Type:      | None                                    |
| Parameter Range:     |   |
| Returned Data:       | <value>                                 |
| Returned Data Type:  | REAL                                    |
| Returned Data Range: | As for command                          |
| Units:               | As set by UNIT:AER                      |
| Function:            | To query the altitude correction value. |
| Conditions:          | None.                                   |

---

## CALC:AZER

### Command Syntax

#### **CALCulate:AZERo <state>**

|                  |                                      |
|------------------|--------------------------------------|
| Parameter Type:  | Boolean                              |
| Parameter Range: | ON   OFF                             |
| Units:           |                                      |
| Password         | Yes                                  |
| Function:        | To switch auto zero on or off.       |
| Conditions:      | Controllers must be in measure mode. |

### Query Syntax

#### **CALCulate:AZERo?**

|                      |                               |
|----------------------|-------------------------------|
| Parameter Type:      | None                          |
| Parameter Range:     |                               |
| Returned Data:       | <state>                       |
| Returned Data Type:  | BOOLEAN                       |
| Returned Data Range: | ON   OFF                      |
| Units:               | As set by UNIT:AER            |
| Function:            | To query the auto zero state. |
| Conditions:          | None.                         |

## CALC:LIM:MAX:PRES

### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password

Function:

Conditions:

### Query Syntax

**CALCulate:LIMit:MAXimum:PRESSure? <parameter>**

Parameter Type: DISCRETE

Parameter Range: ALT | CAS | EPR | PS | QC | TAS | MACH | PT

Returned Data Type: REAL

Units: As set by UNIT:AER or UNIT PRES

Function: To query the maximum pressure limits of the ADTS405MK2.  
(These are the maximum limits that can be set by SOUR:LIM:MAX:PRES).

Conditions: None.

---

## CALC:LIM:MAX:RATE

### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password

Function:

Conditions:

### Query Syntax

**CALCulate:LIMit:MAXimum:RATE? <parameter>**

Parameter Type: DISCRETE

Parameter Range: ALT | CAS | EPR | PS | QC | TAS | MACH | PT

Returned Data Type: REAL

Units: As set by UNIT:AER or UNIT:PRES

Function: To query the maximum rate limits of the ADTS405MK2.  
(These are the maximum values that can be set by SOUR:LIM:MAX:RATE)

Conditions: None

## CALC:LIM:MIN:PRES

### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

### Query Syntax

**CALCulate:LIMit:MINimum:PRESSure? <parameter>**

Parameter Type: DISCRETE

Parameter Range: ALT | CAS | EPR | PS | QC | TAS | MACH | PT

Returned Data Type: REAL

Units: As set by UNIT:AER or UNIT:PRES

Function: To query the minimum pressure limits of the ADTS405MK2.  
(These are the minimum limits that can be set by SOUR:LIM:MIN:PRES).

Conditions: None



---

## CALC:PTT

### Command Syntax

#### **CALCulate:PTTemp <value>**

|                  |   |
|------------------|---|
| Parameter Type:  | REAL  |
| Parameter Range: | See user manual                                 |
| Units:           | As set by UNIT:TEMP                             |
| Password:        | Yes   |
| Function:        | To program the Pt temperature for true airspeed |
| Conditions:      | None.   |

### Query Syntax

#### **CALCulate:PTTemp?**

|                      |   |
|----------------------|---|
| Parameter Type:      | None  |
| Parameter Range:     |   |
| Returned Data:       | <value>                                       |
| Returned Data Type:  | REAL  |
| Returned Data Range: | As for command                                |
| Units:               | As set by UNIT:TEMP                           |
| Function:            | To query the Pt temperature for true airspeed |
| Conditions:          | None.   |

## CALC:SALT

### Command Syntax

#### **CALCulate:SALTitude <value>**

|                  |   |
|------------------|---|
| Parameter Type:  | REAL  |
| Parameter Range: | See ADTS405MK2 user manual                                    |
| Units:           | As set by UNIT:AER  |
| Password:        | Yes   |
| Function:        | To program the station (airfield) altitude value used for QNH |
| Conditions:      | None  |

### Query Syntax

#### **CALCulate:SALTitude?**

|                      |   |
|----------------------|---|
| Parameter Type:      | None  |
| Parameter Range:     |   |
| Returned Data:       | <value>   |
| Returned Data Type:  | REAL  |
| Returned Data Range: | As for command  |
| Units:               | As set by UNIT:AER  |
| Function:            | To query the station (airfield) altitude value used for QNH |
| Conditions:          | None.   |

---

## CALibrate

This group of commands is for calibrating the main transducers and the rate control system. See the calibration manual for full details.

## CAL:ABOR

### Command Syntax

#### CALibration:ABORt

Parameter Type: None

Parameter Range:

Units:

Password: Yes

Function: Exits calibration mode and returns to user mode.

Conditions: None

### Query Syntax

#### N/A

Parameter Type:

Parameter Range:

Returned Data:

Returned Data Type:

Returned Data Range:

Units:

Function:

Conditions:

## CAL:ADJ

### Command Syntax

**CALibration:ADJust** <channel>, <span>, <zero>, <res(0)>, <res(1)>, <res(2)>, <res(3)>, <res(4)>, <res(5)>, <res(6)>, <res(7)>, <res(8)>, <res(9)>, <res(10)>, <res(11)>

|                  |   |
|------------------|---|
| Parameter Type:  | DISCRETE, REAL, REAL, INTEGER (x12)   |
| Parameter Range: | PS   PT, 0.9-1.1, $\pm 100$ , $\pm 127$ (x12)   |
| Units:           | None, mbar/mbar, mbar, ppm FS (x12)   |
| Password:        | Yes   |
| Function:        | To manually adjust calibration. As well as zero and span, the curve fit can be adjusted using residual points. Refer to ADTS405MK2 User Manual for further information. |
| Conditions:      | Calibration enable switch must be active.   |

### Query Syntax

**CALibration:ADJust?** <channel>

|                      |  |
|----------------------|--|
| Parameter Type:      | DISCRETE   |
| Parameter Range:     | PS   PT  |
| Returned Data:       | <span>, <zero>, <res(0)>, <res(1)>, <res(2)>, <res(3)>, <res(4)>, <res(5)>, <res(6)>, <res(7)>, <res(8)>, <res(9)>, <res(10)>, <res(11)> |
| Returned Data Type:  | REAL,REAL,INTEGER(x12)   |
| Returned Data Range: | 0.9-1.1, $\pm 100$ , $\pm 127$ (x12)   |
| Units:               | mbar/mbar, mbar, ppm FS  |
| Function:            | To query the span, zero, and residual values for a specific channel.   |
| Conditions:          | This command requires up to 15 seconds before results are available.   |

**Note:** This query takes about 15 seconds to respond. To prevent controller time out, the following procedure is recommended:

|             |  |
|-------------|--|
| *CLS        | Clear output buffer.   |
| *SRE 16     | Request serial poll on output queue.   |
| CAL:ADJ? PS | Request static channel slope and zero waits serial poll.<br>Read output queue. |

---

## CAL:CHEC:CHAN

### Command Syntax

#### **CALibration:CHECK:CHANnel <channel>**

Parameter Type: DISCRETE

Parameter Range: PS | PT | PSPT

Units:

Password Yes

Function: Starts calibration check facility for selected channel(s).

Conditions: None

### Query Syntax

#### **N/A**

Parameter Type:

Parameter Range:

Returned Data:

Returned Data Type:

Returned Data Range:

Units:

Function:

Conditions:

## **CAL:CHEC:ENDP**

### **Command Syntax**

#### **CALibration:CHECK:ENDPs**

Parameter Type: None

Parameter Range:

Units:

Password Yes

Function: Closes the Qc zero solenoid to protect Ps when PsPt channel is selected for calibration check and all the lower pressures have been applied.

Conditions: Only applies to calibration check mode.

### **Query Syntax**

**N/A**

Parameter Type:

Parameter Range:

Returned Data:

Returned Data Type:

Returned Data Range:

Units:

Function:

Conditions:

---

## CAL:MAIN:ACC

### Command Syntax

#### **CALibration:MAIN:ACCept <state>**

Parameter Type: DISCRETE

Parameter Range: YES | NO

Units:

Password: Yes

Function: Yes - accepts calibration and stores slope and zero corrections.

No - rejects calibration. No adjustment is made.

Conditions: This must follow CAL:MAIN:RES

**Note:** The command stores the current system date as the calibration date and displays it at power-up. Make sure that this date is correct (see SYST:DATE?).

### Query Syntax

**N/A**

Parameter Type:

Parameter Range:

Returned Data:

Returned Data Type:

Returned Data Range:

Units:

Function:

Conditions:

## **CAL:MAIN:CHAN**

### **Command Syntax**

#### **CALibration:MAIN:CHANnel <channel>**

Parameter Type: DISCRETE

Parameter Range: PS | PT | PSPT

Units:

Password: Yes

Function: Starts main calibration for selected channel(s).

Conditions: Calibration switch must be enabled

### **Query Syntax**

#### **N/A**

Parameter Type:

Parameter Range:

Returned Data:

Returned Data Type:

Returned Data Range:

Units:

Function:

Conditions:



---

## CAL:MAIN:RES

### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password

Function:

Conditions:

### Query Syntax

#### **CALibration:MAIN:RESult?**

Parameter Type: None

Parameter Range:

Returned Data: <slope>,<zero>

Returned Data Type: REAL,REAL

Returned Data Range: Any real values.

Units: %reading, units as set by UNIT:PRES

Function: Completes calibration for the selected channel(s) and return the result in terms of zero and slope changes from the last calibration. When the selected channel is PSPT, the first use of CAL:MAIN:RES completes the calibration for the Ps channel and closes the Qc zero solenoid. The second CAL:MAIN:RES completes the Pt calibration.

Conditions: None

**Note:** This query takes about 15 seconds to respond. To prevent controller time out, the following procedure is recommended:

\*CLS Clear output buffer.

\*SRE 16 Request serial poll on output queue.

CAL:ADJ? PS Request static channel slope and zero waits serial poll.

Read output queue.

## CAL:MAIN:VAL

### Command Syntax

#### **CALibration:MAIN:VALue <value>**

|                  |  |
|------------------|--|
| Parameter Type:  | REAL                                   |
| Parameter Range: | Pressure range of ADTS channel.        |
| Units:           | As selected by UNIT:PRES               |
| Password:        | Yes                                    |
| Function:        | Value of actual applied pressure.      |
| Conditions:      | CAL:MAIN:CHAN must have been selected. |

### Query Syntax

#### **N/A**

|                      |
|----------------------|
| Parameter Type:      |
| Parameter Range:     |
| Returned Data:       |
| Returned Data Type:  |
| Returned Data Range: |
| Units:               |
| Function:            |
| Conditions:          |

---

## CAL:RATE

### Command Syntax

#### CALibration:RATE

|                  |   |
|------------------|---|
| Parameter Type:  | None                                      |
| Parameter Range: |   |
| Units:           |   |
| Password:        | Yes                                       |
| Function:        | Starts rate calibration of both channels. |
| Conditions:      | See user manual.                          |

### Query Syntax

#### CALibration:RATE?

|                      |  |
|----------------------|--|
| Parameter Type:      | None   |
| Parameter Range:     |  |
| Returned Data:       | <state>  |
| Returned Data Type:  | DISCRETE   |
| Returned Data Range: | ACTIVE   INACTIVE                                  |
| Units:               |  |
| Function:            | To query if rate calibration is still in progress. |
| Conditions:          | None.  |

## DIAGnostic

This group of commands is for maintenance purposes only.

### DIAG:ERR

#### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

#### Query Syntax

#### DIAGnostic:ERRors?

Parameter Type: None

Parameter Range:

Returned Data: 'nnn :hhhh yy/mm/dd hh:mm:ss <text>'

Returned Data Type: STRING

Returned Data Range:

Units:

Function: To query oldest logged ADTS405MK2 internal error (For SCPI command errors, see SYST:ERR?). Each query returns next error logged until "No Errors Remaining" is returned. Power-up or DIAG:ERR:RES resets pointer to start of error list. <error string> consists of "Error number, date, time, Error text".

Format of string:

nnn - error number

hhhh - hexadecimal code

yy/mm/dd - date

hh:mm:ss - time

<text> - error text

Conditions: None

---

## DIAG:ERR:CLE

### Command Syntax

#### DIAGnostic:ERRors:CLEar

Parameter Type: None

Parameter Range:

Units:

Password: Yes

Function: Clears all logged errors from ADTS.

**Note:** Ensure that all errors have been recorded before clearing.

Conditions: Requires 12 seconds to clear log.

[not required after software DK 127/DK 263 version 4.30 onwards or any DK 415]

### Query Syntax

**N/A**

Parameter Type:

Parameter Range:

Returned Data:

Returned Data Type:

Returned Data Range:

Units:

Function:

Conditions:

## DIAG:ERR:RES

### Command Syntax

#### DIAGnostic:ERRors:RESet

Parameter Type: None

Parameter Range:

Units:

Password: No

Function: Resets pointer to oldest error for use by DIAG:ERR?

Conditions: None

### Query Syntax

#### N/A

Parameter Type:

Parameter Range:

Returned Data:

Returned Data Type:

Returned Data Range:

Units:

Function:

Conditions:

---

## DIAG:SWIT:ERR

### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

### Query Syntax

#### DIAGnostic:SWITcher:ERRor?

Parameter Type:           None

Parameter Range:

Returned Data:           <Error number>, <Error text>

Returned Data Type:      STRING

Returned Data Range:

Units:

Function:                 Query the oldest logged LSU error. Each query returns next logged error, until error 0 is returned. <Error number> may contain a hexadecimal code.

Power up or DIAG:SWIT:ERR:RES resets the pointer to start of error list.

Conditions:               LSU software version V2.10+

**Note:** LSU error log is non-volatile. LSU must be present, see \*OPT?

## DIAG:SWIT:ERR:CLE

### Command Syntax

#### DIAGnostic:SWITcher:ERRor:CLEar

Parameter Type: None

Parameter Range:

Units:

Password: No

Function: Clears all logged errors from the LSU

Conditions: None

**Note:** Make sure all errors have been recorded before this command. LSU must be present, see \*OPT?

### Query Syntax

**N/A**

Parameter Type:

Parameter Range:

Returned Data:

Returned Data Type:

Returned Data Range:

Units:

Function:

Conditions:



---

## DIAG:SWIT:ERR:RES

### Command Syntax

#### DIAGnostic:SWITcher:ERRor:RESet

Parameter Type: None

Parameter Range:

Units:

Password: No

Function: Resets pointer to oldest error used by DIAG:SWIT:ERR?

Conditions: None

**Note:** LSU must be present, see \*OPT?

### Query Syntax

**N/A**

Parameter Type:

Parameter Range:

Returned Data:

Returned Data Type:

Returned Data Range:

Units:

Function:

Conditions:

## MEASure

This group of query commands returns measured values.

### MEAS:AENC

#### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

#### Query Syntax

#### MEASure:AENCoder?

Parameter Type:           None

Parameter Range:

Returned Data:           <value>

Returned Data Type:      REAL

Returned Data Range:     As ADTS

Units:                    As set by UNIT:AER

Function:                 To query altimeter encoder measured pressure

Conditions:               Altimeter encoder option must be fitted

---

## MEAS:PRES

### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

### Query Syntax

#### **MEASure:PRESSure? <parameter>**

Parameter Type: DISCRETE

Parameter Range: ALT | CAS | TAS | MACH | EPR | PS | PT | QC | ARINCALT |  
ARINCCAS | ARINCEPR | ARINCPS | ARINCPT | ARINCQC |  
ARINCTAS | ARINCMACH

Returned Data: <value>

Returned Data Type: REAL

Returned Data Range: As ADTS

Units: As set by UNIT:AER or UNIT:PRES as appropriate.

Function: To query measured pressure.

Conditions: Reads ARINC parameters, ARINC 429 option must be fitted, see  
\*OPT?

## MEAS:QFE

### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

### Query Syntax

**MEASure:QFE?**

Parameter Type:           None

Parameter Range:

Returned Data:           <value>

Returned Data Type:      REAL

Returned Data Range:    As ADTS

Units:                    As set by UNIT:PRES

Function:                 To query QFE measured at power-up.

Conditions:              None

---

## MEAS:QNH

### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

### Query Syntax

**MEASure:QNH?**

Parameter Type: None

Parameter Range:

Returned Data: <value>

Returned Data Type: REAL

Returned Data Range: As ADTS

Units: As set by UNIT:PRES

Function: To query QNH (corrected from QFE measured at power-up)

Conditions: None

## MEAS:RATE

### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

### Query Syntax

#### **MEASure:RATE? <parameter>**

Parameter Type: DISCRETE

Parameter Range: ALT | CAS | TAS | MACH | EPR | PS | PT | QC | ARINCALT |  
ARINCCAS | ARINCEPR | ARINCPS | ARINCPT | ARINCQC |  
ARINCTAS | ARINCMACH

Returned Data: <value>

Returned Data Type: REAL

Returned Data Range: As ADTS

Units: As set by UNIT:AER or UNIT:PRES as appropriate.

Function: To query measured instant rate of change of specified parameter.

Conditions: Rate timing must be off (see SENS:TRAT). To read ARINC parameters, ARINC 429 option must be fitted, see \*OPT?

---

## MEAS:TRAT

### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

### Query Syntax

**MEASure:TRATe? <parameter>**

Parameter Type: DISCRETE

Parameter Range: ALT | CAS | EPR | PS | PT | QC | MACH

Returned Data: <value>

Returned Data Type: REAL

Returned Data Range: As ADTS

Units: As set by UNIT:AER or UNIT:PRES as appropriate.

Function: To query timed rate of change of specified parameter.

Conditions: Rate timing must be TIMED, see SENS:TRAT

# OUTPput

This group of commands control and queries the output valve states. These are the ADTS isolation (OUTPUT) valves and the, optional, LSU.

## OUTP:ALL

### Command Syntax

#### OUTPut:ALL<state>

|                  |                               |
|------------------|-------------------------------|
| Parameter Type:  | DISCRETE                      |
| Parameter Range: | OPEN   CLOSE                  |
| Units:           | None                          |
| Password:        | No                            |
| Function:        | Sets all valves on the LSU.   |
| Conditions:      | LSU must be fitted, see *OPT? |

### Query Syntax

#### N/A

|                      |
|----------------------|
| Parameter Type:      |
| Parameter Range:     |
| Returned Data:       |
| Returned Data Type:  |
| Returned Data Range: |
| Units:               |
| Function:            |
| Conditions:          |



---

## OUTP:CHAN

### Command Syntax

#### OUTPut:CHANnel<source>,<state>

|                  |  |
|------------------|--|
| Parameter Type:  | DISCRETE,DISCRETE  |
| Parameter Range: | INTERNAL   STATIC   PITOT<br>OPEN   CLOSE                    |
| Units:           | None   |
| Password:        | No   |
| Function:        | Sets the state of all outputs connected to specified source. |
| Conditions:      | LSU must be fitted, see *OPT?                                |

### Query Syntax

#### N/A

|                      |  |
|----------------------|--|
| Parameter Type:      |  |
| Parameter Range:     |  |
| Returned Data:       |  |
| Returned Data Type:  |  |
| Returned Data Range: |  |
| Units:               |  |
| Function:            |  |
| Conditions:          |  |

## OUTP:ISOL:STAT

### Command Syntax

#### OUTPut:ISOLation:STATe<state>

|                  |   |
|------------------|---|
| Parameter Type:  | DISCRETE  |
| Parameter Range: | OPEN   CLOSE  |
| Units:           | None  |
| Password:        | No  |
| Function:        | Sets the state of ADTS isolation (output) valves which may be used to isolate ADTS from its output (including the optional LSU). This would be required for leak testing. |
| Conditions:      | Default is OPEN<br>The system must be "At Ground" see STAT:OPER:CON?  |

### Query Syntax

#### OUTPut:ISOLation:STATe?

|                      |                                    |
|----------------------|------------------------------------|
| Parameter Type:      | None                               |
| Parameter Range:     |                                    |
| Returned Data:       | <state>                            |
| Returned Data Type:  | DISCRETE                           |
| Returned Data Range: | OPEN   CLOSE                       |
| Units:               | None                               |
| Function:            | Returns state of isolation valves. |
| Conditions:          | None                               |

---

## OUTP:RES

### Command Syntax

#### OUTPut:RESet

|                  |  |
|------------------|--|
| Parameter Type:  | None                                   |
| Parameter Range: |  |
| Units:           | None                                   |
| Password:        | No                                     |
| Function:        | Sets all LSU valves to power on state. |
| Conditions:      | LSU must be fitted, see *OPT?          |

### Query Syntax

#### N/A

|                      |  |
|----------------------|--|
| Parameter Type:      |  |
| Parameter Range:     |  |
| Returned Data:       |  |
| Returned Data Type:  |  |
| Returned Data Range: |  |
| Units:               |  |
| Function:            |  |
| Conditions:          |  |

## OUTP:VALV:CONF

### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Password:

Units:

Function:

Conditions:

### Query Syntax

**OUTPut:VALVe:CONFIguration?<valve>**

Parameter Type: INTEGER

Parameter Range: 1 to OUTP:VALV:MAX?

Returned Data: <source>

Returned Data Type: STRING

Returned Data Range: INTERNAL | STATIC | PITOT | \*\*\*NOT FITTED\*\*\*

Units:

Function: Returns the channel connected to the specific LSU valve.

Conditions: LSU is fitted, see \*OPT?

Valve must be fitted, see OUTP:VALV:CONF?

---

## OUTP:VALV:ERR

### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

### Query Syntax

**OUTPut:VALVe:ERRor?<valve>**

Parameter Type: INTEGER

Parameter Range: 1 to DIAG:VALV:MAX?

Returned Data: <error flag>

Returned Data Type: INTEGER

Returned Data Range: 0 to 1

Units: None

Function: Returns 0 if valve is working correctly, 1 if feed back indicates a possible error.

Conditions: LSU must be fitted, see \*OPT?

Valve must be fitted, see OUTP:VALV:CONF?

## OUTP:VALV:MAX

### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

### Query Syntax

#### **OUTPut:VALVe:MAXimum?**

Parameter Type:           None

Parameter Range:

Returned Data:           <max>

Returned Data Type:      INTEGER

Returned Data Range:    0 to 24

Units:                    None

Function:                 Returns the number of output valves connected to the LSU.

Conditions:               LSU must be fitted, see \*OPT?

---

## OUTP:VALV:NAME

### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

### Query Syntax

**OUTPut:VALVe:NAME?<valve>**

Parameter Type: INTEGER

Parameter Range: 1 to OUTP:VALV:MAX?

Returned Data: <name>

Returned Data Type: STRING

Returned Data Range:

Units: None

Function: Returns the name of a specific valve.

Conditions: LSU must be fitted, see \*OPT?  
Valve must be fitted, see OUTP:VALV:CONF?

## OUTP:VALV:PRES

### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

### Query Syntax

#### **OUTPut:VALVe:PRESSure?<valve>**

Parameter Type: INTEGER

Parameter Range: 1 to OUTP:VALV:MAX?

Returned Data: <pressure>

Returned Data Type: REAL

Returned Data Range: Range of ADTS

Units: Set by UNIT:PRES

Function: Returns the pressure at the output side of the valve, or "Unknown" if the valve was closed at power-up.

Conditions: LSU must be fitted, see \*OPT?  
Valve must be fitted, see OUTP:VALV:CONF?



---

## OUTP:VALV:STAT

### Command Syntax

#### **OUTPut:VALVe:STATe<valve>,<state>**

|                  |                                       |
|------------------|---------------------------------------|
| Parameter Type:  | INTEGER,DISCRETE                      |
| Parameter Range: | 1 to OUTP:VALV:MAX?                   |
| Units:           | OPEN   CLOSE                          |
| Password:        | No                                    |
| Function:        | Sets the state of specified LSU valve |
| Conditions:      | Default is OPEN                       |

### Query Syntax

#### **OUTPut:VALVe:STATe?<valve>**

|                      |  |
|----------------------|--|
| Parameter Type:      | INTEGER  |
| Parameter Range:     | 0 to OUTP:VALV:MAX?  |
| Returned Data:       | <state>  |
| Returned Data Type:  | DISCRETE   |
| Returned Data Range: | OPEN   CLOSE   |
| Units:               | None   |
| Function:            | Returns state of specified valve.  |
| Conditions:          | LSU must be fitted, see *OPT?<br>Valve must be fitted, see OUTP:VALV:CONF? |

## SENSe

This group of commands is used to control rate timing.

### SENS:TRAT

#### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

#### Query Syntax

**SENSe:TRATe?**

Parameter Type:           None

Parameter Range:

Returned Data:           <state>

Returned Data Type:      DISCRETE

Returned Data Range:    OFF | WAITING | TIMING | TIMED

Units:

Function:                 To query the state of rate timing.

Conditions:               None

---

## SENS:TRAT:RES

### Command Syntax

#### **SENSe:TRATe:RESet**

Parameter Type: None

Parameter Range:

Units:

Password: No

Function: To cancel rate timing and return to instant rate display.

Conditions: None

### Query Syntax

#### **N/A**

Parameter Type:

Parameter Range:

Returned Data:

Returned Data Type:

Returned Data Range:

Units:

Function:

Conditions:

## **SENS:TRAT:STAR**

### **Command Syntax**

#### **SENSe:TRATe:STARt**

Parameter Type: None

Parameter Range:

Units:

Password: No

Function: To start rate timing.

Conditions: None

### **Query Syntax**

#### **N/A**

Parameter Type:

Parameter Range:

Returned Data:

Returned Data Type:

Returned Data Range:

Units:

Function:

Conditions:

---

## SENS:TRAT:TIME

### Command Syntax

#### **SENSe:TRATe:TIME <min>,<sec>**

|                  |   |
|------------------|---|
| Parameter Type:  | INTEGER,INTEGER   |
| Parameter Range: | 0 to 59,0 to 59   |
| Units:           | minutes,seconds   |
| Password:        | No  |
| Function:        | To program time period for rate timing. Default 1 minute. |
| Conditions:      | None  |

### Query Syntax

#### **SENSe:TRATe:TIME?**

|                      |   |
|----------------------|---|
| Parameter Type:      | None  |
| Parameter Range:     |   |
| Returned Data:       | <min>,<sec>                                 |
| Returned Data Type:  | INTEGER,INTEGER                             |
| Returned Data Range: | 0 to 59,0 to 59                             |
| Units:               | minutes,seconds                             |
| Function:            | To query remaining time during rate timing. |
| Conditions:          | None.                                       |

## **SENS:TRAT:WAIT**

### **Command Syntax**

#### **SENSe:TRATe:WAIT <min>,<sec>**

|                  |  |
|------------------|--|
| Parameter Type:  | INTEGER,INTEGER  |
| Parameter Range: | 0 to 59,0 to 59  |
| Units:           | minutes,seconds  |
| Password:        | No   |
| Function:        | To program wait time for rate timing. Default 5 minutes. |
| Conditions:      | None   |

### **Query Syntax**

#### **SENSe:TRATe:WAIT?**

|                      |  |
|----------------------|--|
| Parameter Type:      | None   |
| Parameter Range:     |  |
| Returned Data:       | <min>,<sec>                                      |
| Returned Data Type:  | INTEGER,INTEGER                                  |
| Returned Data Range: | 0 to 59,0 to 59                                  |
| Units:               | minutes,seconds                                  |
| Function:            | To query remaining wait time during rate timing. |
| Conditions:          | None   |

---

## SOURce

This group of commands controls the pressure controller state, aim and limit values.

### SOUR:ARIN:SPEE

#### Command Syntax

#### **SOURce:ARINc:SPEEd <parameter>**

Parameter Type: DISCRETE

Parameter Range: HIGH | LOW

Units:

Password: Yes

Function: Sets speed of ARINC 429 transmit function:  
High is 100 K  
Low is 12.5 K

Conditions: None

#### Query Syntax

#### **SOUR:ARINc:SPEEd?**

Parameter Type:

Parameter Range:

Returned Data:

Returned Data Type: DISCRETE

Returned Data Range: HIGH | LOW

Units:

Function: To query ARINC 429 option transmit speed.

Conditions: None

## SOUR:ARIN:TRAN

### Command Syntax

#### SOURce:ARINc:TRANsmit <parameter>

|                  |  |
|------------------|--|
| Parameter Type:  | DISCRETE   |
| Parameter Range: | ON   OFF   |
| Units:           |  |
| Password:        | Yes  |
| Function:        | Sets the ARINC transmit mode:<br>ON sends measured data.<br>OFF does not send measured data. |
| Conditions:      | None   |

### Query Syntax

#### SOUR:ARINc:TRANsmit?

|                      |  |
|----------------------|--|
| Parameter Type:      |  |
| Parameter Range:     |  |
| Returned Data:       | <mode>                                   |
| Returned Data Type:  | DISCRETE                                 |
| Returned Data Range: | ON   OFF                                 |
| Units:               |  |
| Function:            | To query ARINC 429 option transmit mode. |
| Conditions:          | None                                     |



---

## SOUR:GTGR

### Command Syntax

#### SOURce:GTGRound

Parameter Type: None

Parameter Range:

Units:

Password: No

Function: To return to ADTS to ground pressure and go to last measurement made. Rates must be set first.

Conditions: Must be in CONTROL mode.

**Note:** To check when the system is at ground pressure use the STAT:OPER:CON? command or SOUR:GTGR?. When at ground pressure, the protection valves open even if they are closed; see OUTP:PROT:STATE.

### Query Syntax

#### SOURce:GTGRound?

Parameter Type: None

Parameter Range:

Returned Data: <state>

Returned Data Type: BOOLEAN

Returned Data Range: 0 | 1

Units:

Function: Returns:  
1 - The last SOUR:GTGR has reached ground and there has been no subsequent change to Control mode.  
0 - For all other conditions.

Conditions: None

## SOUR:LIM:ARINC

### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

### Query Syntax

#### **SOURce:LIMit:ARINC?**

Parameter Type:           None

Parameter Range:

Returned Data:           <state>

Returned Data Type:      DISCRETE

Returned Data Range:    TRUE | FALSE

Units:

Function:                 To query the ARINC 565 limits enabled function.

**Note:** ARINC limits set by SOUR:LIM:AIRC ARINC.

Conditions:               None

---

## SOUR:LIM:AIRC

### Command Syntax

#### **SOURce:LIMit:AIRCraft <limits>**

Parameter Type:

Parameter Range: STANDARD | MAX | ARINC

Units:

Password: Yes

Function: Load STANDARD, ARINC 565 or MAXIMUM limits into SCPI limits.

Conditions: Limits can only be changed in leak-measure mode.

### Query Syntax

**N/A**

Parameter Type:

Parameter Range:

Returned Data:

Returned Data Type:

Returned Data Range:

Units:

Function:

Conditions:

## SOUR:LIM:MACH

### Command Syntax

#### **SOURce:LIMit:MACH <parameter>**

Parameter Type: DISCRETE

Parameter Range: ON | OFF

Units:

Password: Yes

Function: Sets the Mach limit disable function. When enabling MACH after it has been disabled, Mach limit is set to maximum valid Mach limit for the air data test system build.

Conditions: Limits can only be changed in leak-measure mode.

### Query Syntax

#### **SOURce:LIMit:MACH?**

Parameter Type: None

Parameter Range:

Returned Data: <state>

Returned Data Type: DISCRETE

Returned Data Range: ON | OFF

Units:

Function: To query the Mach disable function.

Conditions: None.

## SOUR:LIM:MAX:PRES

### Command Syntax

#### **SOURce:LiMit:MAXimum:PRESSure <parameter>,<value>**

|                  |  |
|------------------|--|
| Parameter Type:  | DISCRETE   |
| Parameter Range: | ALT   CAS   Ps   Qc   MACH<br>ADTS maximum limits (CALC:LIM:MAX:PRES?) |
| Units:           | As set by UNIT:AER or UNIT:PRES  |
| Password:        | Yes  |
| Function:        | Sets the maximum pressure limit.                                       |

|             |   |
|-------------|---|
| Conditions: | Limits can only be changed in leak-measure mode.<br><b>Note:</b> Changing limit values removes ARINC 565 enabled limits if these are set. |
|-------------|---|

### Query Syntax

#### **SOURce:LiMit:MAXimum:PRESSure?<parameter>**

|                      |   |
|----------------------|---|
| Parameter Type:      | None  |
| Parameter Range:     | ALT   CAS   Ps   Qc   MACH   TAS   Pt   EPR   |
| Returned Data:       | <state>   |
| Returned Data Type:  | REAL  |
| Returned Data Range: | ADTS limits.  |
| Units:               | As set by UNIT:AER or UNIT:PRES   |
| Function:            | To query the maximum pressure limits.<br><b>Note:</b> SOUR:LIM:MAX:PRES? Mach returns 0.0 when the Mach limit is not enabled. |
| Conditions:          | None  |

## SOUR:LIM:MAX:RATE

### Command Syntax

#### **SOURce:LIMit:MAXimum:RATE <parameter>,<value>**

|                  |   |
|------------------|---|
| Parameter Type:  | DISCRETE  |
| Parameter Range: | ALT   CAS   Ps   Qc<br>ADTS maximum limits (see CALC:LIM:MAX:RATE?)   |
| Units:           | As set by UNIT:AER or UNIT:PRES   |
| Password:        | Yes   |
| Function:        | Sets the maximum rate limit.  |
| Conditions:      | Limits can only be changed in leak-measure mode.<br><b>Note:</b> Changing limit values removes ARINC 565 enabled limits if these are set. |

### Query Syntax

#### **SOURce:LIMit:MAXimum:RATE?<parameter>**

|                      |                                      |
|----------------------|--------------------------------------|
| Parameter Type:      | None                                 |
| Parameter Range:     | ALT   CAS   EPR   Ps   Qc   TAS   Pt |
| Returned Data:       | <state>                              |
| Returned Data Type:  | REAL                                 |
| Returned Data Range: | ADTS limits                          |
| Units:               | As set by UNIT:AER or UNIT:PRES      |
| Function:            | To query the maximum rate limits.    |
| Conditions:          | None                                 |

---

## SOUR:LIM:MIN:PRES

### Command Syntax

#### **SOURce:LIMits:MINimum:PRESSure <parameter>,<value>**

|                  |   |
|------------------|---|
| Parameter Type:  | DISCRETE  |
| Parameter Range: | ALT   CAS   Ps   Qc<br>ADTS minimum limits (see CALC:LIM:MIN:PRES?)   |
| Units:           | As set by UNIT:AER or UNIT:PRES   |
| Password:        | Yes   |
| Function:        | Sets the minimum pressure limit.  |
| Conditions:      | Limits can only be changed in leak-measure mode.<br><b>Note:</b> Changing limit values removes ARINC 565 enabled limits if these are set. |

### Query Syntax

#### **SOURce:LIMits:MINimum:PRESSure?<parameter>**

|                      |   |
|----------------------|---|
| Parameter Type:      | None  |
| Parameter Range:     | ALT   CAS   Ps   Qc   TAS   TAS   Pt   MACH |
| Returned Data:       | <state>                                     |
| Returned Data Type:  | REAL  |
| Returned Data Range: | ADTS limits                                 |
| Units:               | As set by UNIT:AER or UNIT:PRES             |
| Function:            | To query the minimum pressure limits.       |
| Conditions:          | None  |

## **SOUR:MODE:ALE**

### **Command Syntax**

#### **SOURce:MODE:ALEak <state>**

|                  |  |
|------------------|--|
| Parameter Type:  | BOOLEAN                                  |
| Parameter Range: | ON   OFF                                 |
| Units:           |  |
| Password:        | Yes                                      |
| Function:        | To program auto leak recovery on or off. |
| Conditions:      | None                                     |

### **Query Syntax**

#### **SOURce:MODE:ALEak?**

|                      |                                       |
|----------------------|---------------------------------------|
| Parameter Type:      | None                                  |
| Parameter Range:     |                                       |
| Returned Data:       | <state>                               |
| Returned Data Type:  | BOOLEAN                               |
| Returned Data Range: | ON   OFF                              |
| Units:               |                                       |
| Function:            | To query state of auto leak recovery. |
| Conditions:          | None                                  |



---

## SOUR:MODE:ALIM

### Command Syntax

#### **SOURce:MODE:ALIMit <state>**

|                  |   |
|------------------|---|
| Parameter Type:  | BOOLEAN                                   |
| Parameter Range: | ON   OFF                                  |
| Units:           |   |
| Password:        | Yes                                       |
| Function:        | To program auto limit recovery on or off. |
| Conditions:      | None                                      |

### Query Syntax

#### **SOURce:MODE:ALIMit?**

|                      |  |
|----------------------|--|
| Parameter Type:      | None                                   |
| Parameter Range:     |  |
| Returned Data:       | <state>                                |
| Returned Data Type:  | BOOLEAN                                |
| Returned Data Range: | ON   OFF                               |
| Units:               |  |
| Function:            | To query state of auto limit recovery. |
| Conditions:          | None                                   |

## **SOUR:MODE:ARAT**

### **Command Syntax**

#### **SOURce:ARATe <state>**

Parameter Type:            BOOLEAN

Parameter Range:          ON | OFF

Units:

Password:                  Yes

Function:                  Sets the automatic airspeed rate setting. ON adjusts the Rate CAS, when ALTITUDE has not got to the set-point, so that both ALTITUDE and CAS get to their set-points at the same time.

Conditions:                None

### **Query Syntax**

#### **SOUR:MODE:ARATe?**

Parameter Type:            None

Parameter Range:

Returned Data:            <state>

Returned Data Type:        BOOLEAN

Returned Data Range:      ON | OFF

Function:                  To query the state of the AUTO Rate setting.

Conditions:                None

---

## SOUR:MODE:PSON

### Command Syntax

#### **SOURce:MODE:PSOnly <state>**

|                  |                                    |
|------------------|------------------------------------|
| Parameter Type:  | BOOLEAN                            |
| Parameter Range: | ON   OFF                           |
| Units:           |                                    |
| Password:        | Yes                                |
| Function:        | To program Ps only mode on or off. |
| Conditions:      | Must be in measure mode.           |

### Query Syntax

#### **SOURce:MODE:PSOnly?**

|                      |                                 |
|----------------------|---------------------------------|
| Parameter Type:      | None                            |
| Parameter Range:     |                                 |
| Returned Data:       | <state>                         |
| Returned Data Type:  | BOOLEAN                         |
| Returned Data Range: | ON   OFF                        |
| Units:               |                                 |
| Function:            | To query state of Ps only mode. |
| Conditions:          | None                            |

## **SOUR:MODE:PTON**

### **Command Syntax**

#### **SOURce:MODE:PTOnly <state>**

|                  |                                    |
|------------------|------------------------------------|
| Parameter Type:  | BOOLEAN                            |
| Parameter Range: | ON   OFF                           |
| Units:           |                                    |
| Password:        | Yes                                |
| Function:        | To program Pt only mode on or off. |
| Conditions:      | Must be in measure mode.           |

### **Query Syntax**

#### **SOURce:MODE:PTOnly?**

|                      |                                 |
|----------------------|---------------------------------|
| Parameter Type:      | None                            |
| Parameter Range:     |                                 |
| Returned Data:       | <state>                         |
| Returned Data Type:  | BOOLEAN                         |
| Returned Data Range: | ON   OFF                        |
| Units:               |                                 |
| Function:            | To query state of Pt only mode. |
| Conditions:          | None                            |

---

## SOUR:PRES

### Command Syntax

#### **SOURce:PRESSure <parameter>, <aim>**

|                  |  |
|------------------|--|
| Parameter Type:  | DISCRETE,REAL  |
| Parameter Range: | See CALC:LIM:MAX:PRES? and CALC:LIM:MIN:PRES?  |
| Units:           | As set by UNIT:AER or UNIT:PRES as appropriate.  |
| Password:        | No   |
| Function:        | To command the selected parameter to a new aim value.  |
| Conditions:      | Controllers must be ON before using this command. Refer to STAT:OPER:CON? for further details. |

### Query Syntax

#### **SOURce:PRESSure? <parameter>**

|                      |  |
|----------------------|--|
| Parameter Type:      | DISCRETE   |
| Parameter Range:     | ALT   CAS   TAS   MACH   EPR   PS   PT   QC                        |
| Returned Data:       | <aim>  |
| Returned Data Type:  | REAL   |
| Returned Data Range: | As for command   |
| Units:               | As set by UNIT:AER or UNIT:PRES as appropriate.                    |
| Function:            | To query the last aim value commanded.                             |
| Conditions:          | See CALC:LIM:MAX:PRES? and CALC:LIM:MIN:PRES? for currents limits. |

## SOUR:RATE

### Command Syntax

#### **SOURce:RATE <parameter>,<aim>**

|                  |   |
|------------------|---|
| Parameter Type:  | DISCRETE,REAL   |
| Parameter Range: | See CALC:LIM:MAX:RATE? for upper limit; lower limit is 0.                   |
| Units:           | Rate of change of units as set by either UNIT:AER or UNIT:PRES.             |
| Password:        | No  |
| Function:        | To command the rate of change of the selected parameter to a new aim value. |
| Conditions:      | Must be in control mode.  |

### Query Syntax

#### **SOURce:RATE? <parameter>**

|                      |   |
|----------------------|---|
| Parameter Type:      | DISCRETE  |
| Parameter Range:     | ALT   CAS   EPR   PS   PT   QC   TAS   MACH                     |
| Returned Data:       | <aim>   |
| Returned Data Type:  | REAL  |
| Returned Data Range: | As for command.   |
| Units:               | Rate of change of units as set by either UNIT:AER or UNIT:PRES. |
| Function:            | To query the last rate aim command.                             |
| Conditions:          | See CALC:LIM:MAX:RATE? for maximum limits.                      |

## SOUR:STAT

### Command Syntax

#### SOURce:STATe <state>

|                  |   |
|------------------|---|
| Parameter Type:  | DISCRETE  |
| Parameter Range: | CONTROL   MEASURE   ON   OFF   HOLD   RELEASE   |
| Units:           |   |
| Password:        | No  |
| Function:        | To change the mode of the pressure controllers.<br>CONTROL or ON to go to control mode.<br>MEASURE or OFF to go to measure mode.<br>HOLD to go to hold mode.<br>RELEASE to release hold mode.   |
| Conditions:      | Must be in CONTROL to go to HOLD mode.<br><b>Note:</b> It takes a few seconds to turn the controllers ON or OFF. To check controller operating conditions, use the STAT:OPER:CON? command. This returns data showing when Ps and Pt are at set-point i.e. control established. Refer to STAT:OPER:CON for further details. Controllers must be on before the SOUR:PRES command can be used. |

### Query Syntax

#### SOURce:STATe?

|                      |  |
|----------------------|--|
| Parameter Type:      | None   |
| Parameter Range:     |  |
| Returned Data:       | <state>  |
| Returned Data Type:  | DISCRETE   |
| Returned Data Range: | ON   OFF   HOLD  |
| Units:               |  |
| Function:            | To query state of pressure controllers.<br>HOLD means CONTROL 'AND' HOLD |
| Conditions:          | None   |

## STATUS

This group of commands monitor and control the status bytes that show the status of the ADTS.

### STAT:OPER:CON

#### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password: Yes

Function:

Conditions: None

#### Query Syntax

### STATus:OPERation:CONdition?

Parameter Type: None

Parameter Range:

Returned Data: <status>

Returned Data Type: INTEGER

Returned Data Range: 0-32767

Units:

Function: To query the status of the ADTS; this data is unlatched.

Table 7-3 shows the function of each bit.

Conditions: None



---

## STAT:OPER:ENAB

### Command Syntax

#### **STATus:OPERation:ENABLE <data>**

Parameter Type: INTEGER

Parameter Range: 0-32767

Units:

Password: No

Function: To set the Operation Enable register. Each bit in the Operation Enable register allows a corresponding bit in the Operation Event register to set OPR (bit 7) of the status register.

Table 7-3 shows the function of each bit.

Conditions: None

### Query Syntax

#### **STATus:OPERation:ENABLE?**

Parameter Type: None

Parameter Range:

Returned Data: <data>

Returned Data Type: INTEGER

Returned Data Range: 0-32767

Units:

Function: To query the state of the Operation Event register.

Conditions: None

## STAT:OPER:EVEN

### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

### Query Syntax

#### **STATus:OPERation:EVENT?**

Parameter Type:           None

Parameter Range:

Returned Data:           <status>

Returned Data Type:      INTEGER

Returned Data Range:    0-32767

Units:

Function:                 To query the status of the Standard Operations Event register. The data is latched and will be cleared by this query or by \*CLS. Table 7-3 shows the details of each bit.

Conditions:               None

---

## STAT:QUES:CON

### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

### Query Syntax

#### **STATus:QUEStionable:CONDition?**

Parameter Type:           None

Parameter Range:

Returned Data:           <status>

Returned Data Type:      INTEGER

Returned Data Range:     0-32767

Units:

Function:                 To query the Questionable Data Condition register; this data is unlatched. Table 7-3 shows the function of each bit.

Conditions:               None

## STAT:QUES:ENAB

### Command Syntax

#### STATus:QUESTionable:ENABle <data>

Parameter Type: INTEGER

Parameter Range: 0-32767

Units:

Password: No

Function: To set the Questionable Data Enable register. Each bit in the Questionable Data Enable register allows a corresponding bit in the Questionable Data Event register to set QUE (bit 3) of the status register. Table 7-1 shows the function of each bit.

Conditions: None.

### Query Syntax

#### STATus:QUESTionable:ENABle?

Parameter Type: None

Parameter Range:

Returned Data: <data>

Returned Data Type: INTEGER

Returned Data Range: 0-32767

Units:

Function: To query the state of the Questionable Data Enable register.

Conditions: None

---

## STAT:QUES:EVEN

### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

### Query Syntax

#### **STATus:QUEStionable:EVENT?**

Parameter Type:           None

Parameter Range:

Returned Data:           <status>

Returned Data Type:      INTEGER

Returned Data Range:     0-32767

Units:

Function:                 To query the Questionable Data register. This data is latched and will be cleared by this query or by \*CLS. Table 7-1 shows the function of each bit.

Conditions:               None

## SYSTEM

This group of commands controls system functions such as data, time and errors.

### SYST:DATE

#### Command Syntax

#### **SYSTEM:DATE <yyyy>,<mm>,<dd>**

Parameter Type: INTEGER,INTEGER,INTEGER

Parameter Range: 1993->, 1 to 12, 1 to 31

Units:

Password: Yes

Function: To set the system date.

Conditions: None

#### Query Syntax

#### **SYSTEM:DATE?**

Parameter Type: None

Parameter Range:

Returned Data: <yyyy>,<mm>,<dd>

Returned Data Type: INTEGER,INTEGER,INTEGER

Returned Data Range: 1993->, 1-12, 1-31

Units:

Function: To query system date.

Conditions: None

---

## SYST:ERR

### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

### Query Syntax

#### **SYSTEM:ERRor?**

Parameter Type: None

Parameter Range:

Returned Data: <err number>, <err text>

Returned Data Type: INTEGER, STRING

Returned Data Range: -32767 to +32767, string maximum length 256 bytes.

Units:

Function: Returns oldest SCPI error and then deletes that error. Use the query repeatedly to get all errors until 0, "No error" is returned.

**Note:** <err text> is returned in "double quotes".

Conditions: None

## **SYST:PASS:CDIS**

### **Command Syntax**

#### **SYSTem:PASSword:CDISable <Password>**

Parameter Type:

Parameter Range:

Units:

Password: No

Function: Disables some commands with a password (see command summary).

Conditions: Using an invalid password causes an error -221.

Password contains a maximum of 32 characters.

Passwords are case sensitive.

### **Query Syntax**

**N/A**

Parameter Type:

Parameter Range:

Returned Data:

Returned Data Type:

Returned Data Range:

Units:

Function:

Conditions:



---

## SYST:PASS:CEN

### Command Syntax

#### **SYSTEM:PASSWORD:CENable <Password>**

Parameter Type:

Parameter Range:

Units:

Password: No

Function: Enables commands normally protected by passwords.

Conditions: Using an invalid password causes an error -221.

Password contains a maximum of 32 characters.

Passwords are case sensitive.

### Query Syntax

**N/A**

Parameter Type:

Parameter Range:

Returned Data:

Returned Data Type:

Returned Data Range:

Units:

Function:

Conditions:

## SYST:PASS:NEW

### Command Syntax

#### **SYSTem:PASSword:NEW <current>,<new>**

Parameter Type: String

Parameter Range: Any string up to 32 characters long, containing letters 'A' to 'Z' and numbers '0' to '9'.

Units:

Password: No

Function: Changes the password from current to new.

**Note:** Both current and new passwords are case sensitive.  
Conditions: If <current> is not the correct current password this error causes an error -221.

### Query Syntax

**N/A**

Parameter Type:

Parameter Range:

Returned Data:

Returned Data Type:

Returned Data Range:

Units:

Function:

Conditions:

---

## SYST:PASS:STAT

### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

### Query Syntax

**SYSTEM:PASSWORD:STATE?**

Parameter Type:

Parameter Range:

Returned Data:

Returned Data Type: INTEGER

Returned Data Range: 0 | 1

Units:

Function: Returns 0 when password disables protected commands (password locks out commands).

Returns 1 when password enables protected commands.

Conditions: None

## **SYST:PRES**

### **Command Syntax**

#### **SYSTem:PRESet**

Parameter Type: None

Parameter Range:

Units:

Password: Yes

Function: Causes system to go into abort sequence.

Conditions: This should be the last command in a sequence sent to the unit, further commands are ignored.

### **Query Syntax**

#### **N/A**

Parameter Type:

Parameter Range:

Returned Data:

Returned Data Type:

Returned Data Range:

Units:

Function:

Conditions:

---

## SYST:TIME

### Command Syntax

#### **SYSTEM:TIME <hh>,<mm>,<ss>**

Parameter Type: INTEGER,INTEGER,INTEGER

Parameter Range: 0 to 23, 0 to 59, 0 to 59

Units:

Password: Yes

Function: To set the system time.

Conditions: None

### Query Syntax

#### **SYSTEM:TIME?**

Parameter Type: None

Parameter Range:

Returned Data: <hh>,<mm>,<ss>

Returned Data Type: INTEGER,INTEGER,INTEGER

Returned Data Range: 0 to 23, 0 to 59, 0 to 59

Units:

Function: To query system time.

Conditions: None

## SYST:VERS

### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

### Query Syntax

#### **SYSTem:VERSion?**

Parameter Type:           None

Parameter Range:

Returned Data:           <version>

Returned Data Type:      REAL

Returned Data Range:     1992.0

Units:

Function:                  To query SCPI standard version.

Conditions:               None

## TEST

This group of commands sets the self-test functions of the LSU.

### TEST:SWIT:ACT

#### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

#### Query Syntax

### TEST:SWITcher:ACTive?

Parameter Type: None

Parameter Range:

Returned Data: <state>

Returned Data Type: INTEGER

Returned Data Range: 0 | 1

Units:

Function: Performs an active self-test on the LSU.

'0' indicates a fail.

'1' indicates a pass.

Conditions LSU must be fitted, see \*OPT?. The system must be at ground and pressures 'not locked in'.

**Note:** The self-test changes the state of each valve. The recommended procedure follows:

|                |                                      |
|----------------|--------------------------------------|
| *CLS           | Clear output buffer.                 |
| *SRE 16        | Request serial poll on output queue. |
| TEST:SWIT:ACT? | Request switcher self-test.          |
|                | Wait for serial poll.                |
|                | Read output queue.                   |

## TEST:SWIT:PASS

### Command Syntax

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

### Query Syntax

#### TEST:SWITcher:PASSive?

Parameter Type:           None

Parameter Range:

Returned Data:           <data>

Returned Data Type:      INTEGER

Returned Data Range:     0 | 1

Units:

Function:                       Performs a passive self-test on the LSU.

                                  '0' indicates a fail.

                                  '1' indicates a pass.

Conditions                    LSU must be fitted, see \*OPT?

**Note:** The self-test changes the state of each valve. The recommended procedure follows:

\*CLS                            Clear output buffer.

\*SRE 16                        Request serial poll on output queue.

TEST:SWIT:PASS               Request switcher self-test.

                                  Wait for serial poll.

                                  Read output queue.



## UNITs

This group of commands is used for setting the units of measurement.

### UNIT:AER

#### Command Syntax

#### UNITs:AERonautical <units>

|                  |   |
|------------------|---|
| Parameter Type:  | DISCRETE  |
| Parameter Range: | FTKNTS   MKPH   MKPH (M/MIN)   MKPH (M/S)   MKPH (HM/MIN)   |
| Units:           |   |
| Password:        | No  |
| Function:        | To set the aeronautical units used for SCPI commands and queries. The display only shows these units when a SOUR:PRES or SOUR:RATE command is sent. |

| Unit          | Selection  |
|---------------|--|
| FTKNTS        | Selects feet and knots (feet/minute)                         |
| MKPH          | Selects metres and kilometres per hour (metres/minute)       |
| MKPH (M/min)  | Selects metres and kilometres per hour (metres/minute)       |
| MKPH (M/s)    | Selects metres and kilometres per hour (metres/second)       |
| MKPH (hM/min) | Selects metres and kilometres per hour (hecto metres/minute) |

Conditions: None

#### Query Syntax

#### UNITs:AERonautical?

|                      |  |
|----------------------|--|
| Parameter Type:      | None   |
| Parameter Range:     |  |
| Returned Data:       | <units>  |
| Returned Data Type:  | DISCRETE   |
| Returned Data Range: | FTKNTS   MKPH   MKPH (M/S)   MKPH (HM/MIN)   MKPH (M/MIN)   MKPH (M/S) |
| Units:               |  |
| Function:            | To query the SCPI aeronautical units.                                  |

**Note:** Units MKPH are the same as MKPH (M/MIN) and are included for backwards compatibility only, units in (brackets) refer to ROC.

## UNIT:PRES

### Command Syntax

#### UNITs:PRESSure <units>

|                  |   |
|------------------|---|
| Parameter Type:  | DISCRETE  |
| Parameter Range: | MBAR   INH2O4   INH2O20   INHG   MMHG   PA   HPA   PSI  <br>INH2O60F   KGCM2   %FS   MMH2O4   |
| Units:           |   |
| Password:        | No  |
| Function:        | To set the pressure units used for SCPI commands and queries.<br>The display only shows these units when a SOUR:PRES or<br>SOUR:RATE command is sent. |
| Conditions:      | None  |

### Query Syntax

#### UNITs:PRESSure?

|                      |  |
|----------------------|--|
| Parameter Type:      | None   |
| Parameter Range:     |  |
| Returned Data:       | <units>  |
| Returned Data Type:  | DISCRETE   |
| Returned Data Range: | MBAR   INH2O4   INH2O20   INHG   MMHG   KPA   HPA   PSI  <br>INH2O60F   KGCM2   %FS   MMH2O4 |
| Units:               |  |
| Function:            | To query the SCPI pressure units.  |
| Conditions:          | None   |

---

## UNIT:TEMP

### Command Syntax

#### **UNITs:TEMPerature <units>**

Parameter Type: DISCRETE

Parameter Range: C | F | CEL | FAR

Units:

Password: No

Function: To set the temperature units used by SCPI commands and queries.

Conditions: None

### Query Syntax

#### **UNITs:TEMPerature?**

Parameter Type: None

Parameter Range:

Returned Data: <units>

Returned Data Type: DISCRETE

Returned Data Range: C | F

Units:

Function: To query SCPI temperature units.

Conditions: None

## **\*Standard Commands**

The commands starting with \* are SCPI standard commands.

### **\*CLS**

#### **Command Syntax**

##### **\*CLS**

Parameter Type: None

Parameter Range:

Units:

Password: No

Function: Clears status data structures (clearing registers, output queue, error queue and closes vent valves).

Conditions: None

#### **Query Syntax**

##### **N/A**

Parameter Type:

Parameter Range:

Returned Data:

Returned Data Type:

Returned Data Range:

Units:

Function:

Conditions:

---

## **\*DTM**

### **Command Syntax**

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

### **Query Syntax**

**\*DTM?**

Parameter Type:           None

Parameter Range:

Returned Data:           'DOW MMM DD hh:mm:ss YYYY'

Returned Data Type:      STRING

Returned Data Range:

Units:

Function:                 To query the current system date and time.

Conditions:               None

Example:                  'Wed Oct 21 11:03:36 2015'

### **\*ESE**

#### **Command Syntax**

##### **\*ESE <data>**

|                  |  |
|------------------|--|
| Parameter Type:  | INTEGER  |
| Parameter Range: | 0-255  |
| Units:           |  |
| Password:        | No   |
| Function:        | To set the Standard Event Enable register. Each bit in the Standard Event Enable register allows a corresponding bit in the Standard Event register to set ESB (bit 5) of the status register. Table 7-2 shows the function of each bit. |
| Conditions:      | None   |

#### **Query Syntax**

##### **\*ESE?**

|                      |   |
|----------------------|---|
| Parameter Type:      | None  |
| Parameter Range:     |   |
| Returned Data:       | <data>  |
| Returned Data Type:  | INTEGER   |
| Returned Data Range: | 0-255   |
| Units:               |   |
| Function:            | To query the state of the Standard Event Enable register. |
| Conditions:          | None  |

---

**\*ESR****Command Syntax****N/A**

Parameter Type:

Parameter Range:

Units:

Password:

**Query Syntax****\*ESR?**

Parameter Type: None

Parameter Range:

Returned Data: &lt;data&gt;

Returned Data Type: INTEGER

Returned Data Range: 0-255

Units:

Function: To query the value of the Standard Event register.  
The data is latched and will be cleared by this query.  
Table 7-2 shows the function of each bit.

Conditions: None

**Note:** Set when the power-up sequence has been completed.

## **\*ID2?**

### **Command Syntax**

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

### **Query Syntax**

**\*ID2?**

Parameter Type:           None

Parameter Range:

Returned Data:           <CARDTYPE>, <Software Version>

Returned Data Type:      STRING

Returned Data Range:

Units:

Function:                 To query identification of the communication card.

Conditions:               DK 415 software only.

Example:                 'ADTSCOMMS, 1.04/0'



---

**\*IDN****Command Syntax****N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

**Query Syntax****\*IDN?**

Parameter Type: None

Parameter Range:

Returned Data: &lt;Manufacturer&gt;, &lt;Model&gt;, &lt;Serial No.&gt;, &lt;Software Version&gt;

Returned Data Type: STRING

Returned Data Range:

Units:

Function: To query identification of IEEE device.

Conditions: None

## **\*NET**

### **Command Syntax**

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

### **Query Syntax**

**\*NET?**

Parameter Type:           None

Parameter Range:

Returned Data:           IP=<n.n.n.n>

Returned Data Type:      STRING

Returned Data Range:

Units:

Function:                 To query the IP address of the communications card.

Conditions:               DK 415 software only.

Example:                 'IP=192.168.2.3'

---

## \*OPC

### Command Syntax

#### \*OPC

|                  |  |
|------------------|--|
| Parameter Type:  | None   |
| Parameter Range: |  |
| Units:           |  |
| Password:        | No   |
| Function:        | This command, although recognized by the system, is ignored. |
| Conditions:      | None.  |

### Query Syntax

#### \*OPC?

|                      |  |
|----------------------|--|
| Parameter Type:      | None   |
| Parameter Range:     |  |
| Returned Data:       | <date>   |
| Returned Data Type:  | INTEGER  |
| Returned Data Range: | 0  |
| Units:               |  |
| Function:            | This query causes a '0' to be immediately written into the output queue. |
| Conditions:          | None   |

### **\*OPT**

#### **Command Syntax**

**N/A**

Parameter Type:

Parameter Range:

Units:

Function:

Conditions:

#### **Query Syntax**

**\*OPT?**

Parameter Type:           None

Parameter Range:

Returned Data:           <options list>

Returned Data Type:     INTEGER,INTEGER, ...

Returned Data Range:    0 or 1, 0 or 1

Units:

Function:                 Returns option information, this is a sequence of INTEGER values as follows:

\*LSU present (0 = no, 1 = yes).

\*ARINC 429 option PCB present (0 = no, 1 = yes).

Conditions:              Later versions of this unit may include new options, adding more information with a maximum string length 256 characters.

---

## \*RST

### Command Syntax

#### \*RST

Parameter Type: None

Parameter Range:

Units:

Password: No

Function: This command, although recognized by the system, has no function and is ignored.

Conditions:

### Query Syntax

#### N/A

Parameter Type:

Parameter Range:

Returned Data:

Returned Data Type:

Returned Data Range:

Units:

Function:

Conditions

### **\*SRE**

#### **Command Syntax**

##### **\*SRE <data>**

Parameter Type: INTEGER

Parameter Range: 0-255

Units:

Password No

Function:

To set the Service Request Enable register. Each bit in the Service Request Enable register enables a corresponding bit in the Status Byte (bit 6). Bit 6 is always ignored. Table 7-4 shows the function of each bit.

Conditions: None

#### **Query Syntax**

##### **\*SRE?**

Parameter Type: None

Parameter Range:

Returned Data: <data>

Returned Data Type: INTEGER

Returned Data Range: 0-255

Units:

Function: To query the state of the Service Request Enable register.

Conditions: Bit 6 is always 0.

---

**\*STB****Command Syntax****N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

**Query Syntax****\*STB?**

Parameter Type: None

Parameter Range:

Returned Data: &lt;data&gt;

Returned Data Type: INTEGER

Returned Data Range: 0-255

Units:

Function: To query the state of the Status Byte.

Table 7-4 shows the function of each bit that represents each status byte.

Conditions: None

### **\*TST**

#### **Command Syntax**

**N/A**

Parameter Type:

Parameter Range:

Units:

Password:

Function:

Conditions:

#### **Query Syntax**

**\*TST?**

Parameter Type: None

Parameter Range:

Returned Data: <data>

Returned Data Type: INTEGER

Returned Data Range: 0 or 1

Units:

Function: Performs a self-test of the ADTS. A self-test is performed as at power-on. When complete, returns: Pass = 1 or FAIL = 0. The self-test operation takes approximately 60 seconds but, depending on configuration, may take longer.

Conditions: The ADTS405MK2 must be at ground before performing self-test.



---

## \*TTO

### Command Syntax

#### \*TTO <data>

|                  |  |
|------------------|--|
| Parameter Type:  | INTEGER  |
| Parameter Range: | 0-4294967295   |
| Units:           | milliseconds   |
| Password:        |  |
| Function:        | Set maximum time the GPIB interface will remain in talk mode if no data becomes available to send. Responds with 'nothing to say' if this period expires. Should be longer than the maximum query response time. (Works around a problem with FGPA implementation). A value of '0' means no timeout. |

Conditions:

### Query Syntax

#### \*TTO?

|                      |                                       |
|----------------------|---------------------------------------|
| Parameter Type:      | None.                                 |
| Parameter Range:     |                                       |
| Returned Data:       |                                       |
| Returned Data Type:  | INTEGER                               |
| Returned Data Range: | 0-4294967295                          |
| Units:               |                                       |
| Function:            | To query the current timeout setting. |
| Conditions:          | DK 415 software only                  |
| Example:             | '50000'                               |

### **\*WAI**

#### **Command Syntax**

##### **\*WAI**

Parameter Type: None

Parameter Range:

Units:

Password: No

Function: This command, although recognized by the system, is ignored.

Conditions:

#### **Query Syntax**

##### **N/A**

Parameter Type:

Parameter Range:

Returned Data:

Returned Data Type:

Returned Data Range:

Units:

Function:

Conditions

## 10. Errors

Negative error numbers are used for standard SCPI errors. Positive error numbers are reserved for device specific errors but are not used for the ADTS405MK2. Following the error number, a message describes the error. A bit is set in the event status register for different error number ranges, i.e. error number -299 to -200 set bit 4.

### 10.1 Error Numbers -499 to -400 (Query Errors)

Set the events status register bit 2 with error messages as follows:

**Table 10-1: Query Errors**

| Error Code | Error Message                            |
|------------|--|
| -410,      | -410, "Query INTERRUPTED"                |
| -410,      | -410, "Query INTERRUPTED; Bus timeout"   |
| -400,      | -400, "Query Error; Output buffer full"  |
| -400,      | -400, "Query Error; Output buffer empty" |

### 10.2 Error Numbers -399 to -300 (Device Specific Errors)

Set the events status register bit 3 with error messages as follows:

**Table 10-2: Device Specific Errors**

| Error Code | Error Message   |
|------------|---|
| -363       | -363, "Input buffer overrun"                                      |
| -360       | -360, "Communications error; LSU timeout"                         |
| -310       | -310, "System error; Internal error - wrong integer type"         |
| -310       | -310, "System error; Internal error - no matching discrete value" |

### 10.3 Error Numbers -299 to -200 (Execution Errors)

Set the events status register bit 4 with error messages as follows:

**Table 10-3: Execution Errors**

| Error Code | Error Message   |
|------------|---|
| -241       | -241, "Hardware missing"                                      |
| -240       | -240, "Hardware error, Altimeter encoder"                     |
| -224       | -224, "Data out of range; Output not fitted"                  |
| -224       | -224, "Illegal parameter value; Time Period Not Active"       |
| -224       | -224, "Illegal parameter value; Rate parameter not available" |
| -224       | -224, "Illegal parameter value; Wait Period Not Active"       |
| -222       | -222, "Data out of range"                                     |
| -222       | -222, "Data out of range; Invalid Time Period"                |
| -222       | -222, "Data out of range; Invalid date"                       |
| -222       | -222, "Data out of range; Invalid Wait Period"                |
| -222       | -222, "Data out of range; Invalid time"                       |

**Table 10-3: Execution Errors**

| <b>Error Code</b> | <b>Error Message</b>   |
|-------------------|--|
| -222              | -222, "Data out of range; Beyond ADTS programmed limits"     |
| -222              | -222, "Data out of range; Beyond programmed measured values" |
| -222              | -222, "Data out of range; Beyond ADTS programmed limits"     |
| -222              | -222, "Data out of range; Beyond current set-point"          |
| -221              | -221, "Settings conflict; Rate has not been timed"           |
| -221              | -221, "Settings conflict; Only timed rates available"        |
| -221              | -221, "Settings conflict; Must not be controlling"           |
| -221              | -221, "Settings conflict; Must be controlling"               |
| -221              | -221, "Settings conflict; Must be in Measure mode"           |
| -221              | -221, "Settings conflict; Invalid password"                  |
| -203              | -203, "Command protected"                                    |
| -200              | -200, "Execution error; Not at ground"                       |
| -200              | -200, "Execution error; Locked ports"                        |
| -200              | -200, "Execution error; Cal disabled on status panel"        |

**10.4 Error Numbers -199 to -100 (Command Errors)**

Set the events status register bit 5 with error messages as follows:

**Table 10-4: Command Errors**

| <b>Error Code</b> | <b>Error Message</b>  |
|-------------------|---|
| -124              | -124, "Too many digits; Too many mantissa digits"                     |
| -123              | -123, "Exponent to large"   |
| -120              | -120, "Numeric data error; Digits expected"                           |
| -113              | -113, "Undefined header; Unknown command"                             |
| -110              | -110, "Command Header Error; Insufficient characters"                 |
| -109              | -109, "Missing parameter; String expected"                            |
| -109              | -109, "Missing parameter; Discrete expected"                          |
| -109              | -109, "Missing parameter; Comma expected"                             |
| -108              | -108, "Parameter not allowed; Too many parameters"                    |
| -108              | -108, "Parameter not allowed"   |
| -104              | -104, "Data type error; Integer value between -128 and +127 expected" |
| -104              | -104, "Data type error; Integer value between -127 and +127 expected" |
| -104              | -104, "Data type error; Integer value between 0 and 255 expected"     |
| -104              | -104, "Data type error; Integer value between 0 and 65535 expected"   |
| -101              | -101, "Invalid character; Command terminator expected"                |
| -100              | -100, "Command error; Parameter not recognised"                       |

## 11. Calibration

### 11.1 Calibration using SCPI

The following routines are example programs which could be used to calibrate either Ps, Pt or both channels of the ADTS through the SCPI interface. An internal pressure source has been used.

### 11.2 Single Channel Calibration (PS or PT)

**Note:** In the following procedure, channel xx refers to either Ps or Pt as appropriate.

1. SYST:DATE?  
Before calibration, check the system date. The date will be stored when the calibration has been completed. If the returned string is incorrect, change it using the command SYST:DATE yyyy,mm,dd.
2. SOUR:STAT ON  
Turn on the controllers.
3. CAL:MAIN:CHAN xx  
Enter main calibration mode for channel xx (PS or PT).
4. UNITS:PRES <units>  
Select the units used for calibration.
5. SOUR:RATE xx, <rate>  
Set the rate for the channel.
6. SOUR:PRES xx, <aim>  
Send the channel to a known pressure.
7. Wait for the channel to get to the set-point; this can be checked by reading the "Control Status Byte" via STAT:OPER:CON? the returned bit pattern indicates when channel xx is at set-point. To find out when the channel has got to the set-point, set up a serial poll to interrupt the controller or poll the status bytes.
8. Wait at least one minute, after getting to the set-point, for the pressures to stabilize.
9. CAL:MAIN:VAL <value>  
Enter pressure indicated by the pressure standard.
10. Repeat steps 6 to 9. At least two different pressure points must be entered for the ADTS to calculate the slope and zero values, although it is recommended that at least five different pressure points are used.
11. CAL:MAIN:RES?  
This causes the ADTS to calculate and return the slope and zero changes from the last calibration.  
**Note:** The reply can take several seconds to calculate.
12. CAL:MAIN:ACC YES  
This takes two seconds to accept the changes returned in step 11. Also, the current date (set in step 1) is stored and the new calibration data backed up.

### 11.3 Dual Channel Calibration

1. SYST:DATE?  
Before calibration, check the system date. The date will be stored when the calibration has been completed. If the returned string is incorrect, change it using the command SYST:DATE yyyy,mm,dd.

## Chapter 11. Calibration

---

2. SOUR:STAT ON  
Turn on the controllers.
3. CAL:MAIN:CHAN PSPT  
Enter main calibration mode.
4. UNITS:PRES <units>  
Select the units used for calibration.
5. SOUR:RATE PS, <rate>  
Set the rate for the PS channel.
6. SOUR:PRES PS, <aim>  
Send the channels to a known pressure.
7. Wait for the channel to get to the set-point; this can be checked by reading the “Control Status Byte” via STAT:OPER:CON?
8. Wait at least one minute, after getting to the set-point, for the pressures to stabilize.
9. CAL:MAIN:VAL <value>  
Enter pressure indicated by the pressure standard.
10. Repeat steps 6 to 9. At least two different pressure points must be entered for the ADTS to calculate the slope and zero values, although it is recommended that at least five different pressure points are used.
11. CAL:MAIN:RES?  
This causes the ADTS to calculate and return the slope and zero for the PS channel.  
**Note:** These values can take several seconds to calculate.
12. CAL:MAIN:ACC YES  
This takes two seconds to accept the PS changes returned in step 11.
13. SOUR:RATE PT, <rate>  
Set the rate for the PT channel.
14. SOUR:PRES PT, <aim>  
Send the Pitot channel to a known pressure.
15. Wait for the channel to get to the set-point; this can be checked by reading the “Control Status Byte” via STAT:OPER:CON?
16. Wait at least one minute, after getting to the set-point, for the pressures to stabilize.
17. CAL:MAIN:VAL <value>  
Enter pressure indicated by the pressure standard.
18. Repeat steps 14 to 17. It is recommended that at least three additional different pressure points are used on the Pitot channel.
19. CAL:MAIN:RES?  
This causes the ADTS to calculate and return the slope and zero for the Pt channel.  
**Note:** These values can take several seconds to calculate.
20. CAL:MAIN:ACC YES  
This takes two seconds to accept the Pt changes returned in step 11. The current date (set in step 1) is stored and the calibration for both channels is backed up.



## Office Locations



<https://druck.com/contact>

## Services and Support Locations



<https://druck.com/service>