Druck

# ADTS542F
# ADTS552F/553F/554F
# ADTSTOUCH
# ADTSTOUCH-ER

## Air Data Test Set

Communications Manual


druck.com

# Introduction

This technical manual provides communication instructions for the Druck ADTS5xxF Air Data Test Systems 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 DK0428 (Hand Terminal) and DK0467 (Controller) version V06.00.00 onwards.

### 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[1] 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.

### Associated Documents

| Document | Title |
|---|---|
| K0553 | ADTS5xxF User Manual |

### Abbreviations

This manual uses the following abbreviations. They are the same in the singular and plural.

| Abbreviation | Description |
|---|---|
| A | Ampere |
| abs | Absolute |
| ADTS | Air data test system |
| ALT | Altitude |
| ARINC | Aeronautical Radio Incorporated |
| ASCII | American Standard Code for Information Interchange |

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.

| Abbreviation | Description |
|---|---|
| ATE | Automatic test equipment |
| CAS | Calibrated airspeed |
| ft | Foot |
| g | Gauge |
| hPa | Hecto Pascal |
| 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 |
| m | Meter |
| mA | Milliampere |
| max | Maximum |
| mbar | Millibar |
| min | Minute or minimum |
| mm | Millimeter |
| mmHg | Millimeter of mercury |
| No. | Number |
| Pa | Pascal |
| PC | Personnel computer |
| Ps | Pressure static |
| psi | Pounds per square inch |
| Pt | Pressure Total (Pitot) |
| Qc | Differential pressure |
| SCPI | Standard commands for programmable instruments |
| TAS | True airspeed |
| TPM | Test program manager (Druck software package) |
| USB | Universal Serial Bus |
| +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 and others. 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. |
| 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. |

| Item | Description |
|---|---|
| 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

# Tables

# Figures

# 1. Introduction

The USB interface provides remote control of the ADTS from Automatic Test Equipment (ATE), or a PC using standardized commands.

SCPI (pronounced “skippy”) stands for Standard Commands for Programmable Instruments. Created so that all instruments with a SCPI facility can communicate using the same commands. The number of instruments complying with the standard continually grows, although there are only a few pressure instruments.

The ADTS5XXF units implement the full SCPI command set with the exception of expressions and units suffixes. Although the SCPI standard defines all the ADTS5XXF commands at the first, or ‘root’ level, many of the functions necessary have not been allowed for in the standard. The lower levels of commands were created using the appropriate rules contained within the standard.

This manual gives information for each command where it specifically relates to the SCPI implementation of the command.

For a good introduction to the subject, refer to 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).

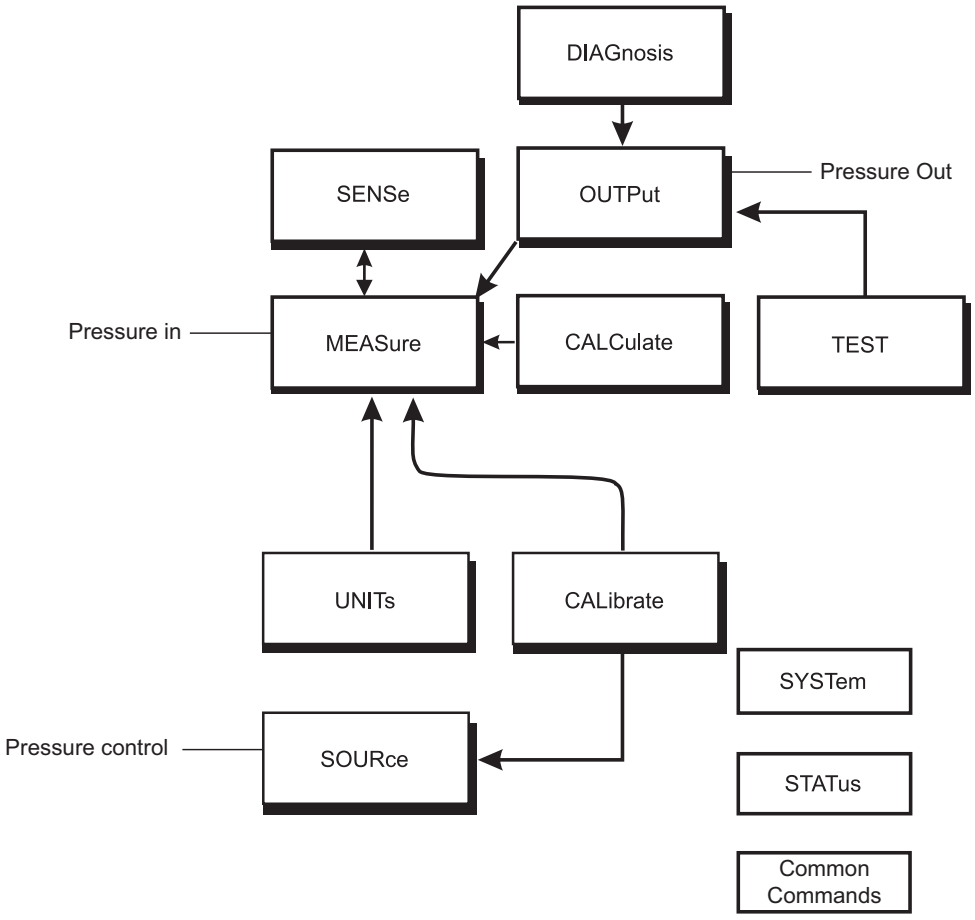



**Figure 1-1: System Model**

# Chapter 1. Introduction

# 2. Electrical Connection

**CAUTION** **The USB connections are not waterproof. Only use the USB connections in dry conditions.**


**Figure 2-1: USB Sockets on the ADTSTOUCH**

Using a suitable cable, connect either of the USB sockets (1 or 2) on the ADTSTOUCH to a suitable PC or ATE. Both sockets work for SCPI communications.

# Chapter 2. Electrical Connection

# 3. Menu Option Selection for SCPI


**Figure 3-1: Connections Options**

To configure SCPI communications, select **TOOLS** > **CONNECTIONS** on the ADTSTOUCH Dashboard.

To select SCPI communication for either USB-A or USB-Mini connections, select the '**Communication Mode**'. This is only available for one port at a time.

## 3.1 USB-A SCPI Communication


**Figure 3-2: USB-A Connections and Communication Settings**

When you set connections for USB-A as '**Communications Mode**', the USB-A will become a Virtual COM Port (VCP) for serial communication using SCPI. A **COMMUNICATIONS SETTINGS** menu becomes available. This configures serial port settings over the USB-A, including baud rate, parity and flow control.


**Figure 3-3: Status Bar Showing USB-A Icon**

The Status Bar shows a USB-A icon, showing that SCPI communication over USB-A is available. See Figure 3-3.

To confirm communication, send *IDN? Command over the serial connection.

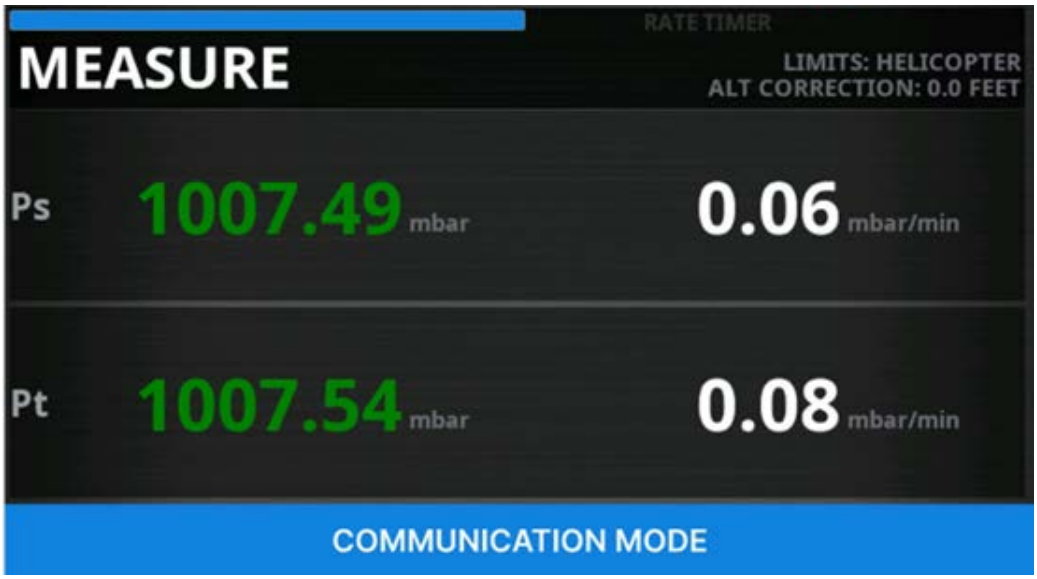



**Figure 3-4: Measure Screen with COMMUNICATION MODE Banner**

On correct communication, the ADTSTOUCH shows the **COMMUNICATION MODE** banner and pitot static screen.

## 3.2 USB-Mini SCPI Communication

When you select connections for USB-Mini as '**Communications Mode**', the USB-Mini will configure as a USB Port, for communication using SCPI, accessing via the device VID and PID.

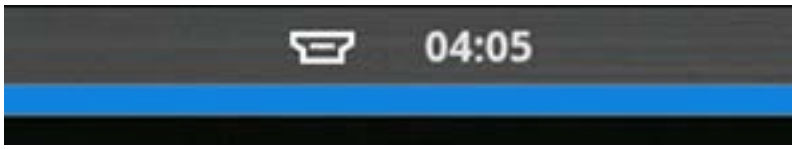



**Figure 3-5: Status Bar Showing USB-Mini Icon**

The Status Bar will show a USB-Mini icon, showing that SCPI communication over USB-Mini is available.

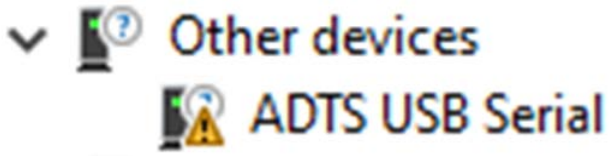



**Figure 3-6: PC Device Manager - Device Not Recognized**

When connected to the PC, the ADTS HT should show as 'ADTS USB Serial'. If the PC does not recognize the device, you may need to install the WinUsb Device Driver.

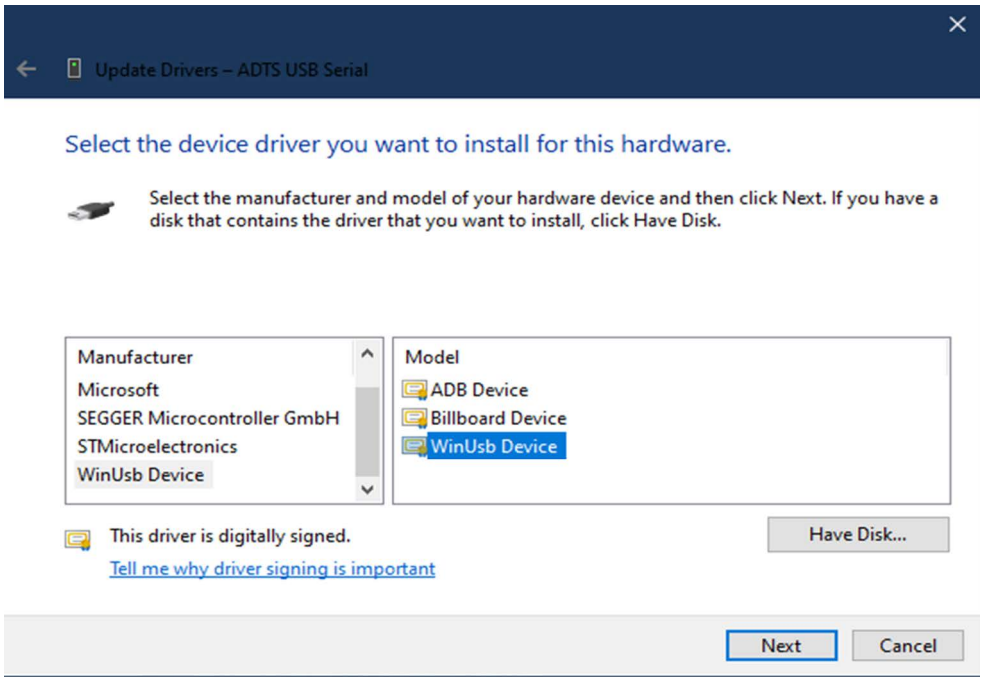



**Figure 3-7: Typical PC Update Drivers Dialog**

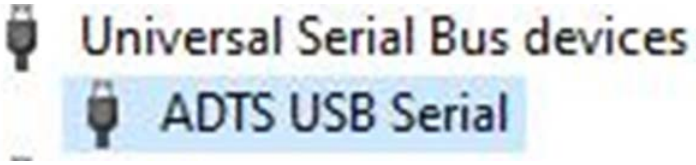



**Figure 3-8: PC Device Manager - Device Set Correctly**

When correctly set up, the device should show under ‘Universal Serial Bus devices’ in the PC Device Manager. See Figure 3-8.

These details are necessary to configure communication to the ADTS USB Serial Device:

- VID = 0x14f0 #Vendor ID
- PID = 0x001C #Product ID

To confirm communication, send *IDN? Command over the USB connection.

On correct communication, the ADTSTOUCH shows the **COMMUNICATION MODE** banner and pitot static screen. See Figure 3-4.

# Chapter 3. Menu Option Selection for SCPI

# 4. Remote/Local Instrument Control

| SCPI Command | Details |
|---|---|
| GTL | GO TO LOCAL<br>Takes the Hand Terminal out of local lockout mode. |
| LLO | LOCAL LOCKOUT<br>Locks the Hand Terminal out of local mode. The Hand Terminal will no longer be in local mode and cannot be operated from the front panel. |
| LOC | LOCAL<br>Puts the Hand Terminal into local mode. The Hand Terminal will no longer be in communications mode and can be operated from the front panel. |
| REM | REMOTE<br>Sets the Hand Terminal in Communications mode. The Hand Terminal will no longer be in local mode and cannot be operated from the front panel. |

Any command received over the SCPI interface puts the ADTS into REMOTE mode. In REMOTE mode, the screen shows a ‘Communications Mode’ banner and is locked. If selecting the banner when in REMOTE, the ADTS returns to LOCAL. If the command GTL is sent to the ADTS, the ADTS returns to LOCAL and the banner disappears. If the command LLO is sent to the ADTS, the banner will not respond when selected. This leaves the ADTS locked in REMOTE until power-down. When the ADTS is in REMOTE, the banner 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.

## 4.1 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.

# Chapter 4. Remote/Local Instrument Control

# 5. Command Syntax and Example Program

## 5.1 Commands to ADTS

The general syntax for a complete message is shown below:

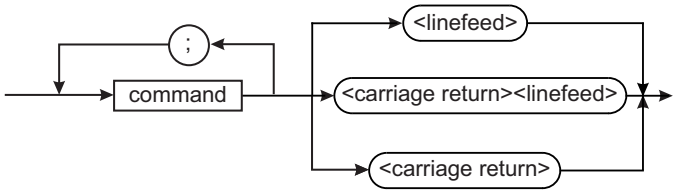



**Figure 5-1: General Command Syntax**

When sending more than one command in the same message, separate each command with a semicolon.

To send the same message twice, for example: MEAS:PRES PS and MEAS:PRES PT you must remove the repeated head of the message as follows MEAS PRES? PS;PRES? PT.

One of the three terminators shown must be used to end the message. For example, either <linefeed> (ASCII char decimal 10), <carriage return> (ASCCI char decimal 13) or <newline><carriage return> as configured in the communications setting menu of the ADTS.

This image shows the syntax for each command:

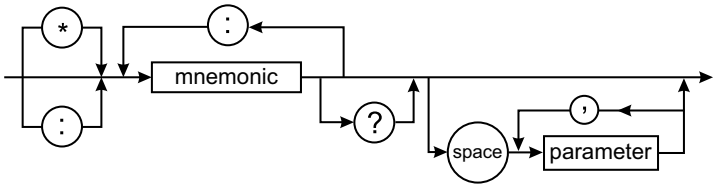



**Figure 5-2: Command Syntax**

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

Example: *ESE

You must use a colon to separate each command word or mnemonic.

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

## 5.2 Command Response from ADTS

This image shows the syntax for response messages:

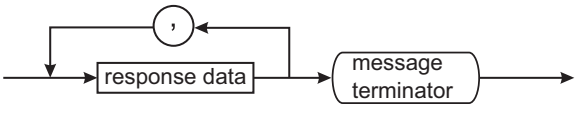



**Figure 5-3: Response Syntax**

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

**Note:** The message terminator used is configured in the communications setting menu of the ADTS.

## 5.3 SCPI Data Types

Various types of data can be sent to the ADTS as parameters or sent out from the ADTS as response data.

### 5.3.1 Numeric

Numeric data can 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.

### 5.3.2 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 ADTS5XXF 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

### 5.3.3 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).

#### 5.3.3.1 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

## 5.4 Example Program

The following program example is written in PYTHON. It assumes connection by COM3 and shows the use of various commands and responses.

### 5.4.1 Example Program Overview

1. Initialize the USB interface.
2. Initialize the ADTS5XXF 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 gained using a function that sends out the query command and returns the response. An error check is done after obtaining the response.

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

### 5.4.2 Example Program - written in Python

```
import time, datetime
import serial

print(str(datetime.datetime.now()) + "- Leak Test Start")
# Configure the serial port (adjust parameters as needed)
ser = serial.Serial(
    port='COM10',       # Replace with your actual serial port
    baudrate=115200,    # Match the instrument's baud rate
    bytesize=serial.EIGHTBITS,
    parity=serial.PARITY_NONE,
    stopbits=serial.STOPBITS_ONE,
    timeout=2           # Timeout for read operations)
```

```
def send_command(command):
    ser.write((command + '\n').encode())
    time.sleep(0.5)  # Short delay to allow processing
    ser.write("SYST:ERR?\n".encode())
    time.sleep(0.5)
    error = ser.readline().decode().strip()
    if error and float(error.split(',')[0]) != 0:
        print(f"Error reported when sending '{command}'")
        print(error)

def get_response(command):
    ser.reset_input_buffer()   # Avoid answering the question you previously asked
    ser.write((command + '\n').encode())
    time.sleep(0.5)
    response = ser.readline().decode().strip()
    ser.write("SYST:ERR?\n".encode())
    time.sleep(0.5)
    error = ser.readline().decode().strip()
    if error and float(error.split(',')[0]) != 0:
        print(f"Error reported when sending '{command}'")
        print(error)
    return response

# Initial setup
send_command("*CLS")
print("Error check reports -", get_response("SYST:ERR?"))

# Set units and control mode
send_command("UNITS:PRESSURE mbar")
send_command("SOURCE:STATE control")
time.sleep(10)

# Enter rate and pressure setpoints
send_command("SOURCE:RATE PS,100;RATE QC,100")
send_command("SOURCE:PRES PS,800;PRES QC,200")
```

```
# Wait for pressures to stabilize
while True:
    status = get_response("STAT:OPER:CON?")
    print(str(datetime.datetime.now()) + " -Going to setpoint")
    try:                    #test if status message can be cast to INT
        int(status)
    except:
        status = 0
    if int(status) & 2:
        break
    time.sleep(5)

# Go to measure mode
send_command("SOUR:STAT MEASURE")

# Program timing and start leak rate measurement
send_command("SENSE:TRATE:WAIT 1,0")
send_command("SENSE:TRATE:TIME 0,30")
send_command("SENSE:TRATE:START")

# Wait for timing to complete
while True:
    status = get_response("SENSE:TRATE?")
    print(str(datetime.datetime.now()) + " -" + str(status))
    if status.startswith("TIMED"):
        break
    time.sleep(5)

# Get and display leak rate
leak_rate = get_response("MEAS:TRATE? ps")
print("Ps leak rate (mbar/min) =", leak_rate)

# Get and display final pressure
final_pressure = get_response("MEAS:PRES? ps")
print("Final Ps pressure (mbar) =", final_pressure)

# Return to control mode and go to ground
send_command("SOURCE:STATE control")
time.sleep(3)
send_command("SOUR:GTGR")
```

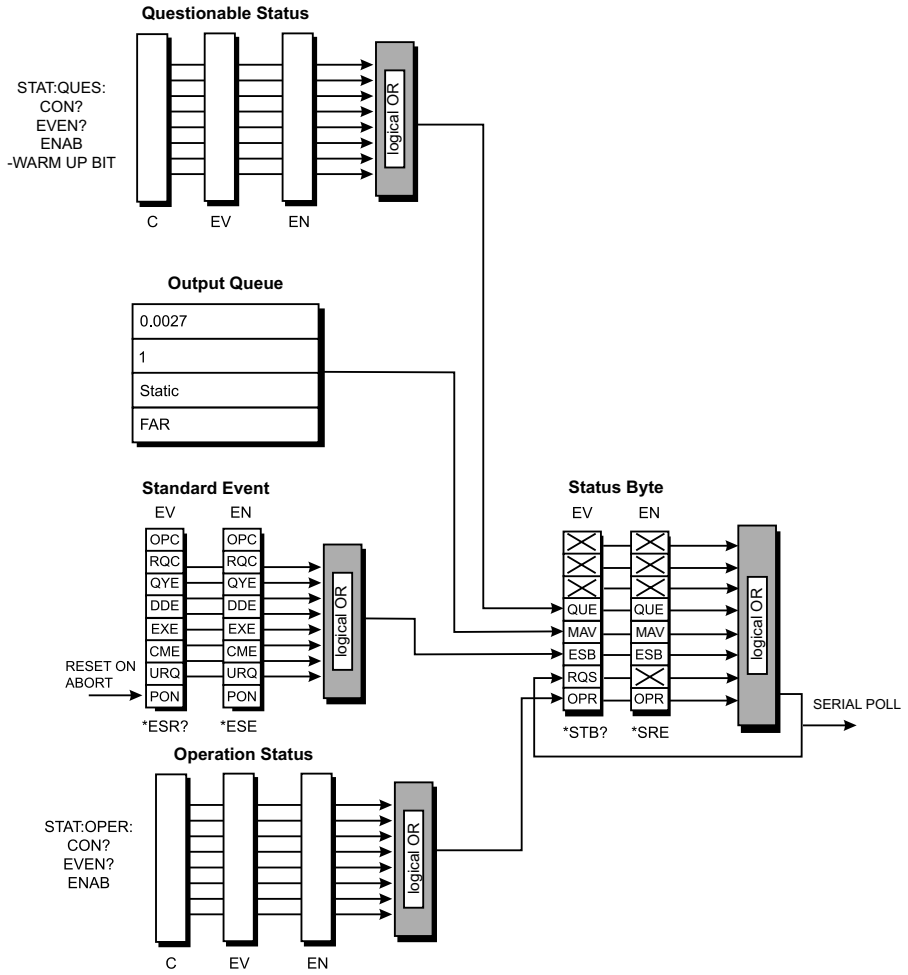
```
# Wait until at ground
while True:
    status = int(get_response("STAT:OPER:CON?"))
    print(str(datetime.datetime.now()) + " -Going to Ground")
    try:
        int(status)
    except:
        status = 0
    if status & 4:
        break

print("******TEST RESULT******")
print("Ps leak rate (mbar/min) =", leak_rate)
print("Final Ps pressure (mbar) =", final_pressure)
print("END OF TEST")
```

# 6. Status System

The following diagram shows the status system as implemented in the ADTS:

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

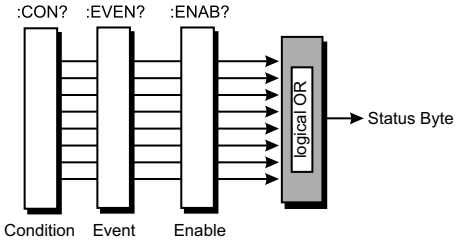



C Condition - variable values
EV Latched values
EN Bit mask

**Figure 6-1: Status System**

## 6.1 Questionable Status Group

The questionable data group are 16 bit registers read through the STAT:QUES commands. Read the event register to clear it. Clear the event and enable registers using 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 can be set through the STAT:QUES:ENAB command so that only selected questionable events cause the QUE bit to set.

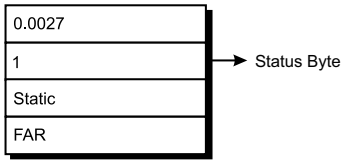
The questionable condition register is defined as follows:

**Table 6-1: Questionable Condition Register**

| Bit | Description |
|---|---|
| 0-6 | Reserved |
| 7 | Controller 1, Auto Zero in progress (ADTS553/ADTS554 only) |
| 8 | Controller 0, Auto Zero in progress |
| 9 | Warm Up. Set during thermal warm up of one or both controllers. |
| 10-15 | Reserved |

## 6.2 Output Queue

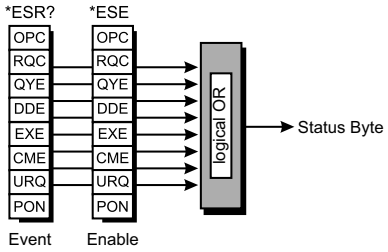
The output queue is a text readable data queue. The queue is cleared by reading all elements in it or by the *CLS command.




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.

## 6.3 Standard Event Group

The standard event group are 8 bit registers. 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 can 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:

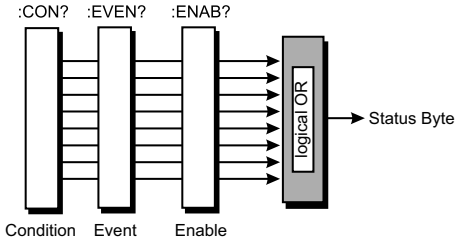
**Table 6-2: Standard Event Register**

| 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. |

See “Error Numbers” on page 95 for the individual error codes.

## 6.4 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 can be set through the STAT:OPER:ENAB command so that only selected standard operation events cause the OPR bit to set. Upon setting the enable register mask, the status will reflect upon the next change in condition.

The condition register is defined as follows:

**Table 6-3: Operation Status Register**

| Bit | Data |
|---|---|
| 0 | Reserved - returns 0. |
| 1 | Stable at aim value* |
| 2 | Safe at ground. |
| 3 | Ramping** |
| 4 | Ps2 at set-point and in control mode*** |

**Table 6-3: Operation Status Register**

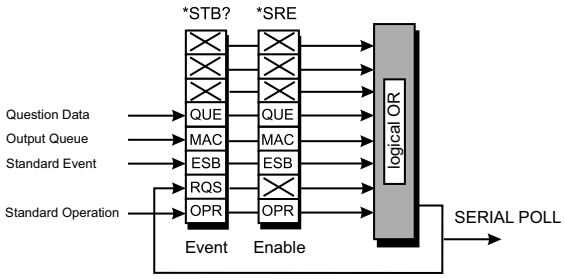
| Bit | Data |
|---|---|
| 5 | Ps2 ramping and achieving rate |
| 6 | Pt2 at set-point and in control mode*** |
| 7 | Pt2 ramping and achieving rate |
| 8 | Ps1 at set-point and in control mode*** |
| 9 | Ps1 ramping and achieving rate |
| 10 | Pt1 at set-point and in control mode*** |
| 11 | Pt1 ramping and achieving rate |
| 12 | Reserved - returns 0. |
| 13 | Reserved - returns 0. |
| 14 | Perform Calibrate - Set if calibrate in progress. |
| 15 | Calibrate Enable - Set if calibrate enabled. |

Bits highlighted by asterisks (*) are set after other bits have become set, determined by the amount of channels on the unit.

| | | Channels on the unit | After these Bits Set |
|---|---|---|---|
| * | Bit 1 is set after 15 seconds (to give pressure time to settle) after other bits become set, depending on the number of channels on the unit. | 2 | 8 and 10 |
| | | 3 | 4, 8 and 10 |
| | | 4 | 4, 6, 8 and 10 |
| ** | Bit 3 is set to one after other bits become set depending on the number of channels on the unit. | 2 | 9 and 11 |
| | | 3 | 5, 9 and 11 |
| | | 4 | 5, 7, 9 and 11 |
| *** | Set immediately when the aims set-points are reached. | | |

## 6.5 Status Byte Group

The status byte group are 8 bit registers that are read by the SCPI status 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 (for example, 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 can 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 6-4: 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. |

# Chapter 6. Status System

# 7. Command and Query Summary

The following is a summary of all the SCPI commands and queries that apply to the ADTS.

## 7.1 Command Structure

Some of the commands are enabled at specific times and conditions. Most can be enabled at any time.

## 7.2 Passwords

Some commands are password protected, for example, 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.
- 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.

## 7.3 Summary of Commands

| Command | Password | Parameters | Comments |
|---|---|---|---|
| CALCulate | | | |
| :ACORrection | N | <value> | |
| :ACORrection? | N | | |
| :AZERo | N | <state> | |
| :AZERo? | N | | |
| :LIMit | N | | |
| :MAXimum | | | |
| :PRESsure? | N | <parameter> | |
| :RATE? | N | <parameter> | |
| :MINimum | | | |
| :PRESsure? | N | <parameter> | |
| :PTTemp | N | <value> | |
| :PTTemp? | N | | |
| CALibration | | | |
| :ABORt | Y | | |
| :CHECk | | | |
| :CHANnel | Y | <channel> | |
| :ENDPs | Y | | |
| :MAIN | | | |
| :ACCept | Y | <state> | |

| Command | Password | Parameters | Comments |
|---|---|---|---|
| :CHANnel | Y | <channel> | |
| :RESult? | N | <slope> <zero> | |
| :VALue | Y | <value> | |
| MEASure | | | |
| :PRESsure? | N | <parameter> | |
| :RATE? | N | <parameter> | |
| :TRATe? | N | <parameter> | |
| OUTPut | | | |
| :ISOLation | | | |
| :STATe | N | <state> | |
| :STATe? | N | | |
| SENSe | | | |
| :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 | | | |
| :GTGRound | N | | |
| :GTGRound? | N | | |
| :LIMit | | | |
| :AIRCraft | Y | <limits> | |
| :AIRCraft? | Y | | |
| :CUSTom | | | |
| :ADD | Y | <custom limit name>, <default limits> | |
| :DEL | Y | <custom limit name> | |
| :LIST? | N | | |
| :MAXimum | | | |
| :PRESsure | Y | <parameter>, <value> | |
| :PRESsure? | N | <parameter> | |
| :RATE | Y | <custom limit name>, <parameter>, <value> | |
| :RATE? | N | <custom limit name>, <parameter> | |
| :MINimum | | | |

| Command | Password | Parameters | Comments |
|---|---|---|---|
| :PRESsure | Y | <custom limit name>, <parameter>, <value> | |
| :PRESsure? | N | <custom limit name>, <parameter> | |
| :MODE | | | |
| :ALEak | N | <state> | |
| :ALEak? | N | | |
| :ARATe | N | <state> | |
| :ARATe? | N | | |
| :CHANnel | Y | | |
| :CHANnel? | 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 | | |
| :EVENt? | N | | |
| :QUEStionable | | | |
| :CONdition? | N | | |
| :ENABle | N | <data> | |
| :ENABle? | N | | |
| :EVENt? | N | | |
| SYSTem | | | |
| :DATE | N | <yyyy>, <mm>, <dd> | } 2000+, 1 to 12, 1 to 31 |
| :DATE? | N | | |
| :ERRor? | N | <error number>, <err text> | |
| :PASSword | | | |
| :CDISable | N | <password> | |
| :CENable | N | <password> | |
| :NEW | N | <current>, <new> | |
| :STATe? | N | | |

| Command | Password | Parameters | Comments |
|---|---|---|---|
| :PRES | N | | |
| :TIME | N | <hh>, <mm>, <ss> | } 0 to 23, 0 to 59, 0 to 59 |
| :TIME? | N | | |
| :VERSion? | N | <version> | |
| UNITs | | | |
| :AERonautical | N | <units> | |
| :AERonautical? | N | | |
| :PRESsure | N | <units> | |
| :PRESsure? | N | | |
| :TEMPerature | N | <units> | |
| :TEMPerature? | N | | |
| *CLS | N | | |
| *ESE | N | <data> | |
| *ESE? | N | | |
| *ESR? | N | <data> | |
| *ID2? | N | <Manufacturer>, <Model>, <Serial No.>, <Software Version> | |
| *IDN? | N | <Manufacturer>, <Model>, <Serial No.>, <Software Version> | |
| *OPC | N | | |
| *OPC? | N | | |
| *OPT? | N | <options list> | |
| *RST | N | | |
| *SRE | N | <data> | |
| *SRE? | N | | |
| *STB? | N | <data> | |
| *TST? | N | <data> | |
| *WAI | N | | |

# 8. Command Reference

## 8.1 Introduction

This section describes each command and query in detail including parameters used and response data returned. The headings show the general short form command.

| Command | Details |
|---|---|
| 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. |
| 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 | Details |
|---|---|
| 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. |
| 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. |

## 8.2 CALCulate

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

### 8.2.1 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. |

### 8.2.2 CALC:AZER

**Command**

| | |
|---|---|
| Syntax: | CALCulate:AZERo <state> |
| Parameter Type: | Boolean |
| Parameter Range: | ON \| OFF |
| Units: | |
| Password | No |
| 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: | |
| Function: | To query the auto zero state. |
| Conditions: | None. |

### 8.2.3 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 instruments pressure limits of the ADTS. (These are the maximum limits that can be set by SOUR:LIM:MAX:PRES). |
| Conditions: | None. |

### 8.2.4 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 ADTS.<br>(These are the maximum values that can be set by SOUR:LIM:MAX:RATE) |
| Conditions: | None |

### 8.2.5 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 ADTS. (These are the minimum limits that can be set by SOUR:LIM:MIN:PRES). |
| Conditions: | None |

### 8.2.6 CALC:PTT

**Command**

| | |
|---|---|
| Syntax: | CALCulate:PTTemp <value> |
| Parameter Type: | REAL |
| Parameter Range: | See user manual |
| Units: | As set by UNIT:TEMP |
| Password: | No |
| Function: | To program the Pt temperature for true airspeed |
| Conditions: | None. |

**Query**

| | |
|---|---|
| Syntax: | CALCulate:PTTemp? |
| Parameter Type: | |
| Parameter Range: | |
| Returned Data: | <value> |
| Returned Data Type: | REAL |
| Returned Data Range: | See user manual. |

| | |
|---|---|
| Units: | As set by UNIT:TEMP |
| Function: | To query the Pt temperature for true airspeed |
| Conditions: | None. |

## 8.3 CALibration

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

### 8.3.1 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: | |

### 8.3.2 CAL:CHEC:CHAN

**Command**

| | |
|---|---|
| Syntax: | CALibration:CHECk:CHANnel <channel> |
| Parameter Type: | DISCRETE |
| Parameter Range: | PS \| PS1 \| PS2 \| PT \| PT1 \| PT2 \| PSPT |
| Units: | |
| Password | Yes |
| Function: | Starts calibration check facility for selected channel(s). |
| Conditions: | CAL:ABOR must be called to end calibration Check Mode |

**Query**

| | |
|---|---|
| Syntax: | N/A |
| Parameter Type: | |
| Parameter Range: | |
| Returned Data: | |
| Returned Data Type: | |
| Returned Data Range: | |
| Units: | |
| Function: | |
| Conditions: | |

### 8.3.3 CAL:CHEC:ENDP

**Command**

| | |
|---|---|
| Syntax: | CALibration:CHECk:ENDPs |
| Parameter Type: | None |
| Parameter Range: | |
| Units: | |
| Password | Yes |
| Function: | No function |
| Conditions: | None |

**Query**

| | |
|---|---|
| Syntax: | N/A |
| Parameter Type: | |
| Parameter Range: | |
| Returned Data: | |
| Returned Data Type: | |
| Returned Data Range: | |
| Units: | |
| Function: | |
| Conditions: | |

### 8.3.4 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.<br>No - rejects calibration. No adjustment is made. |
| Conditions: | Must be in Calibration Mode and complete all required calibration points. |

**Query**

| | |
|---|---|
| Syntax: | N/A |
| Parameter Type: | |
| Parameter Range: | |
| Returned Data: | |
| Returned Data Type: | |
| Returned Data Range: | |
| Units: | |
| Function: | |
| Conditions: | |

### 8.3.5 CAL:MAIN:CHAN

**Command**

| | |
|---|---|
| Syntax: | CALibration:MAIN:CHANnel <channel> |
| Parameter Type: | DISCRETE |
| Parameter Range: | PS \| PS1 \| PS2 \| PT \| PT1 \| PT2 \| PSPT |
| Units: | |
| Password | Yes |
| Function: | Starts main calibration 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: | |

### 8.3.6 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: | 1, 0 |
| Returned Data Type: | REAL,REAL |
| Returned Data Range: | 1, 0 |
| Units: | N/A |
| Password | Yes |
| Function: | Required for ADT405 procedure, default returns 1, 0 for ADTS500 series, not functional. |
| Conditions: | None |

### 8.3.7 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: | |

## 8.4 MEASure

This group of query commands returns measured values.

### 8.4.1 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 \| IN \| OUT \| PS \| PT \| QC<br>ALT1 \| CAS1 \| TAS1 \| MACH1 \| PS1 \| PT1 \| QC1<br>ALT2 \| CAS2 \| TAS2 \| MACH2 \| PS2 \| PT2 \| QC2 |
| 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: | * (-287) EPR\|IN\|OUT option must be enabled to read EPR measurement. Access Error<br>*(-221) Must be at ground to enter EPR mode. Settings conflict error |
| Conditions (ADTS542F): | Only Channel 1 parameters available else<br>*(-220) Error code |
| Conditions (ADTS552F): | Only Channel 1 parameters available else<br>*(-220) Error code |
| Conditions (ADTS553F): | Channel 2 only ALT2 and PS2 available else<br>*(-220) Error code |
| Conditions (ADTS554F): | All parameters valid. |

### 8.4.2 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 \| ALT1 \| ALT2 \| CAS \| CAS1 \| CAS2 \| EPR \| IN \| OUT \| PS\| PS1 \| PS2 \| PT \| PT1 \| PT2 \| QC \| QC1 \| QC2 \| MACH \| MACH1 \| MACH2 \| TAS \| TAS1 \| TAS2 |
| 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).<br>* (-287) EPR\|IN\|OUT option must be enabled to read EPR measurement. Access Error<br>*(-221) Must be at ground to enter EPR mode. Settings conflict error |
| Conditions (ADTS542F): | Only Channel 1 parameters available else<br>*(-220) Error code |
| Conditions (ADTS552F): | Only Channel 1 parameters available else<br>*(-220) Error code |
| Conditions (ADTS553F): | Channel 2 only ALT2 and PS2 available else<br>*(-220) Error code |
| Conditions (ADTS554F): | All parameters valid. |

### 8.4.3 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 \| ALT1 \| ALT2 \| CAS \| CAS1 \| CAS2 \| EPR \| IN \| OUT \| PS\| PS1 \| PS2 \| PT \| PT1 \| PT2 \| QC \| QC1 \| QC2 \| MACH \| MACH1 \| MACH2 \| TAS \| TAS1 \| TAS2 |
| 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 |

## 8.5 OUTPut

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

### 8.5.1 OUTP:ISOL:STAT

**Command**

| | |
|---|---|
| Syntax: | OUTPut:ISOLation:STATe<state> |
| Parameter Type: | DISCRETE |
| Parameter Range: | OPEN \| CLOSE |
| Units: | None |
| Password: | No |
| Function: | To set all the output isolation valves as ON/OFF. |
| Conditions: | Default is OPEN |
| | 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 |

## 8.6 SENSe

This group of commands control rate timing.

### 8.6.1 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\|TESTING\|END |
| Units: | |
| Function: | To query the current state of Leak Test Timing. |
| Conditions: | None |

### 8.6.2 SENS:TRAT:RES

**Command**

| | |
|---|---|
| Syntax: | SENSe:TRATe:RESet |
| Parameter Type: | None |
| Parameter Range: | |
| Units: | |
| Password: | No |
| Function: | To cancel the running leak test timing. |
| Conditions: | None |

**Query**

| | |
|---|---|
| Syntax: | N/A |
| Parameter Type: | |
| Parameter Range: | |
| Returned Data: | |
| Returned Data Type: | |
| Returned Data Range: | |
| Units: | |
| Function: | |
| Conditions: | |

### 8.6.3 SENS:TRAT:STAR

**Command**

| | |
|---|---|
| Syntax: | SENSe:TRATe:STARt |
| Parameter Type: | None |
| Parameter Range: | |
| Units: | |
| Password: | No |
| Function: | To start the leak rate timing. |
| Conditions: | None |

**Query**

| | |
|---|---|
| Syntax: | N/A |
| Parameter Type: | |
| Parameter Range: | |
| Returned Data: | |
| Returned Data Type: | |
| Returned Data Range: | |
| Units: | |
| Function: | |
| Conditions: | |

### 8.6.4 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. |

### 8.6.5 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 |

## 8.7 SOURce

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

### 8.7.1 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 (-221). |

**Query**

| | |
|---|---|
| Syntax: | SOURce:GTGRound? |
| Parameter Type: | None |
| Parameter Range: | |
| Returned Data: | <state> |
| Returned Data Type: | BOOLEAN |
| Returned Data Range: | 0 \| 1 |
| Units: | |
| Function: | Returns:<br>1 - The last SOUR:GTGR has reached ground and there has been no subsequent change to Control mode.<br>0 - For all other conditions. |
| Conditions: | None |

### 8.7.2 SOUR:LIM:AIRC

**Command**

| | |
|---|---|
| Syntax: | SOURce:LIMit:AIRCraft <limit name> |
| Parameter Type: | String |
| Parameter Range: | "MAX ADTS" \| "MAX AERO" \| "FIXED WING" \| "HELICOPTER" \| "ADSPROBE" \| "<Custom Set Name>" |
| Units: | |
| Password: | Yes |
| Function: | To select the aircraft limits set to be used. |
| Conditions: | |
| Conditions (ADTS542F): | ADSPROBE limit set not available. |
| Conditions (ADTS552F): | ADSPROBE limit set not available. |
| Conditions (ADTS553F): | Limits can only be changed in leak-measure mode. |
| Conditions (ADTS554F): | ADSPROBE limit set not available. |

**Query**

| | |
|---|---|
| Syntax: | SOURce:LIMit:AIRCraft? |
| Parameter Type: | |
| Parameter Range: | |
| Returned Data: | <limit name> |
| Returned Data Type: | String |
| Returned Data Range: | |
| Units: | |
| Password: | Yes |
| Function: | To Query which aircraft limit set is currently selected |
| Conditions: | |
| Conditions (ADTS542F): | ADSPROBE limit set not available. |
| Conditions (ADTS552F): | ADSPROBE limit set not available. |
| Conditions (ADTS553F): | |
| Conditions (ADTS554F): | ADSPROBE limit set not available. |

### 8.7.3 SOUR:LIM:AIRC:CUST:ADD

**Command**

| | |
|---|---|
| Syntax: | SOURce:LIMit:AIRCraft:CUSTom:ADD < custom limit name>, <default limits> |
| Parameter Type: | String, String |
| Parameter Range: | "MAX ADTS" \| "MAX AERO" \| "FIXED WING" \| "HELICOPTER" \| "ADSPROBE" |
| Units: | |
| Password: | Yes |
| Function: | To create a new editable custom limit set. Based on a fixed limit set selected. |
| Conditions: | Only if number of custom limit sets created is less than 5. |
| Conditions (ADTS542F): | ADSPROBE limit set not available. |
| Conditions (ADTS552F): | ADSPROBE limit set not available. |
| Conditions (ADTS553F): | |
| Conditions (ADTS554F): | ADSPROBE limit set not available. |

**Query**

| | |
|---|---|
| Syntax: | N/A |
| Parameter Type: | |
| Parameter Range: | |
| Returned Data: | |
| Returned Data Type: | |
| Returned Data Range: | |
| Units: | |
| Function: | |
| Conditions: | |

### 8.7.4 SOUR:LIM:AIRC:CUST:DEL

**Command**

| | |
|---|---|
| Syntax: | SOURce:LIMit:AIRCraft:CUSTom:DELete <custom limit name> |
| Parameter Type: | String |
| Parameter Range: | String |
| Units: | |
| Password: | Yes |
| Function: | To create a new editable custom limit set. |
| Conditions: | Only if custom limits set has been created. |

**Query**

| | |
|---|---|
| Syntax: | N/A |
| Parameter Type: | |
| Parameter Range: | |
| Returned Data: | |
| Returned Data Type: | |
| Returned Data Range: | |
| Units: | |
| Function: | |
| Conditions: | |

### 8.7.5 SOUR:LIM:AIRC:LIST

**Command**

| | |
|---|---|
| Syntax: | N/A |
| Parameter Type: | |
| Parameter Range: | |
| Units: | |
| Password: | |
| Function: | |
| Conditions: | |

**Query**

| | |
|---|---|
| Syntax: | SOURce:LIMit:AIRCraft:LIST? |
| Parameter Type: | |
| Parameter Range: | |
| Returned Data: | < limit set list > |
| Returned Data Type: | String |
| Returned Data Range: | |
| Units: | |
| Function: | To view the list of all available ADTS limit sets. |
| Conditions: | |
| Conditions (ADTS542F): | ADSPROBE limit set will not be present in the list |
| Conditions (ADTS552F): | ADSPROBE limit set will not be present in the list |
| Conditions (ADTS553F): | ADSPROBE limit set will be present in the list |
| Conditions (ADTS554F): | ADSPROBE limit set will not be present in the list |

### 8.7.6 SOUR:LIM:MAX:PRES

**Command**

| | |
|---|---|
| Syntax: | SOURce:LIMit:MAXimum:PRESsure <custom limit name>, <parameter>, <value> |
| Parameter Type: | String, Discrete, Real |
| Parameter Range: | ALT \| ALT1 \| ALT2\| CAS\| CAS1 \| CAS2 \| PS\| PS1 \| PS2 \| QC \| QC1 \| QC2 \| MACH \| MACH1 \| MACH2 |
| | ADTS maximum limits (CALC:LIM:MAX:PRES?) |
| Units: | As set by UNIT:AER or UNIT:PRES |
| Password: | Yes |
| Function: | Sets the maximum pressure limit for the named custom limit set. |
| Conditions: | Only if custom limits set has been created. |
| Conditions (ADTS553F): | Limits can only be changed in leak-measure mode. |

**Query**

| | |
|---|---|
| Syntax: | SOURce:LIMit:MAXimum:PRESsure?<custom limit name>,<parameter> |
| Parameter Type: | String, Discrete |
| Parameter Range: | ALT \| ALT1 \| ALT2 \| CAS\| CAS1 \| CAS2 \| PS\| PS1 \| PS2 \| QC \| QC1 \| QC2 \| MACH \| MACH1 \| MACH2 |
| Returned Data: | < max pressure> |
| Returned Data Type: | REAL |
| Returned Data Range: | ADTS maximum limits. |
| Units: | As set by UNIT:AER or UNIT:PRES |
| Function: | Gets the maximum pressure limit for the named custom limit set. |
| Conditions: | Only if custom limits set has been created. |

### 8.7.7 SOUR:LIM:MAX:RATE

**Command**

| | |
|---|---|
| Syntax: | SOURce:LIMit:MAXimum:RATE <custom limit name>, <parameter>, <value> |
| Parameter Type: | String, Discrete, Real |
| Parameter Range: | PS \| PS1 \| PS2 \| QC \| QC1 \| QC2 \| ROC \| ROC1 \| ROC2 |
| | ADTS maximum limits (CALC:LIM:MAX:RATE?) |
| Units: | As set by UNIT:AER or UNIT:PRES |
| Password: | Yes |
| Function: | Sets the maximum rate limit for the named custom limit set. |
| Conditions: | Only if custom limits set has been created. |
| Conditions (ADTS553F): | Limits can only be changed in leak-measure mode. |

**Query**

| | |
|---|---|
| Syntax: | SOURce:LIMit:MAXimum:RATE? <custom limit name>, <parameter> |
| Parameter Type: | String, Discrete |
| Parameter Range: | PS \| PS1 \| PS2 \| QC \| QC1 \| QC2 \| ROC \| ROC1 \| ROC2 |
| Returned Data: | < max rate > |
| Returned Data Type: | REAL |
| Returned Data Range: | ADTS maximum limits |
| Units: | As set by UNIT:AER or UNIT:PRES |
| Function: | Gets the maximum rate limit for the named custom limit set. |
| Conditions: | Only if custom limits set has been created. |

### 8.7.8 SOUR:LIM:MIN:PRES

**Command**

| | |
|---|---|
| Syntax: | SOURce:LIMits:MINimum:PRESsure <custom limit name>, <parameter>, <value> |
| Parameter Type: | String, Discrete, Real |
| Parameter Range: | ALT \| ALT1 \| ALT2 \| CAS\| CAS1 \| CAS2 \|PS \| PS1 \| PS2 \| QC \| QC1 \| QC2 |
| | ADTS minimum limits (CALC:LIM:MIN:PRES?) |
| Units: | As set by UNIT:AER or UNIT:PRES |
| Password: | Yes |
| Function: | Sets the minimum pressure limit for the named custom limit set. |
| Conditions: | Only if custom limits set has been created |
| Conditions (ADTS553F): | Limits can only be changed in leak-measure mode. |

**Query**

| | |
|---|---|
| Syntax: | SOURce:LIMits:MINimum:PRESsure?<custom limit name>,<parameter> |
| Parameter Type: | String, Discrete |
| Parameter Range: | ALT \| ALT1 \| ALT2 \| CAS\| CAS1 \| CAS2 \| PS \| PS1 \| PS2 \| QC \| QC1 \| QC2 |
| Returned Data: | < min pressure > |
| Returned Data Type: | REAL |
| Returned Data Range: | ADTS limits |
| Units: | As set by UNIT:AER or UNIT:PRES |
| Function: | Gets the minimum pressure limit for the named custom limit set |
| Conditions: | Only if custom limits set has been created. |

### 8.7.9 SOUR:MODE:ALE

**Command**

| | |
|---|---|
| Syntax: | SOURce:MODE:ALEak <state> |
| Parameter Type: | BOOLEAN |
| Parameter Range: | ON \| OFF |
| Units: | |
| Password: | No |
| Function: | To Switch 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 auto leak recovery. |
| Conditions: | None |

### 8.7.10 SOUR:MODE:ARAT

**Command**

| | |
|---|---|
| Syntax: | SOURce:ARATe <state> |
| Parameter Type: | BOOLEAN |
| Parameter Range: | ON \| OFF |
| Units: | |
| Password: | No |
| Function: | Sets the automatic airspeed rate setting. ON adjusts the Rate of CAS 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 |

### 8.7.11 SOUR:MODE:CHAN

**Command**

| | |
|---|---|
| Syntax: | SOURce:MODE:CHANnel |
| Parameter Type: | Discrete, Discrete |
| Parameter Range: | PS \| PS1 \| PS2 \| PT \| PT1 \| PT2,<br>CONTROL \| MEASURE |
| Units: | |
| Password: | Yes |
| Function: | To program Ps/Pt only mode on or off on each channel. |
| Conditions: | Must be in measure mode. |

**Query**

| | |
|---|---|
| Syntax: | SOURce:MODE:CHANnel? |
| Parameter Type: | Discrete |
| Parameter Range: | PS \| PS1 \| PS2 \| PT \| PT1 \| PT2 |
| Returned Data: | <state> |
| Returned Data Type: | Discrete |
| Returned Data Range: | CONTROL \| MEASURE |
| Units: | |
| Function: | To query state of Ps/Pt only mode on each channel. |
| Conditions: | |

### 8.7.12 SOUR:PRES

**Command**

| | |
|---|---|
| Syntax: | SOURce:PRESsure <parameter>,<aim> |
| Parameter Type: | DISCRETE,REAL |
| Parameter Range: | ALT \| CAS \| TAS \| MACH \| EPR \| IN \| OUT \| PS \| PT \| QC<br>ALT1 \| CAS1 \| TAS1 \| MACH1 \| PS1 \| PT1 \| QC1<br>ALT2 \| CAS2 \| TAS2 \| MACH2 \| PS2 \| PT2 \| QC2<br><br>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.<br>(-221)<br>* (-220) EPR\|IN\|OUT option must be enabled to read EPR measurement. Access Error.<br>*(-221) Must be at ground to enter EPR mode. Settings conflict error. |
| Conditions (ADTS542F): | Only Channel 1 parameters available else *(-220) Error code. |
| Conditions (ADTS552F): | Only Channel 1 parameters available else *(-220) Error code. |
| Conditions (ADTS553F): | Channel 2 only ALT2 and PS2 available else *(-220) Error code. |
| Conditions (ADTS554F): | All parameters valid |

**Query**

| | |
|---|---|
| Syntax: | SOURce:PRESsure? <parameter> |
| Parameter Type: | DISCRETE |
| Parameter Range: | ALT \| CAS \| TAS \| MACH \| EPR \| IN \| OUT \| PS \| PT \| QC<br>ALT1 \| CAS1 \| TAS1 \| MACH1 \| PS1 \| PT1 \| QC1<br>ALT2 \| CAS2 \| TAS2 \| MACH2 \| PS2 \| PT2 \| QC2 |
| Returned Data: | <aim> |
| Returned Data Type: | REAL |
| Returned Data Range: | See CALC:LIM:MAX:PRES? and CALC:LIM:MIN:PRES? |
| 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 current limits<br>* (--220) EPR\|IN\|OUT option must be enabled to read EPR measurement. Access Error.<br>*(-221) Must be at ground to enter EPR mode. Settings conflict error. |
| Conditions (ADTS542F): | Only Channel 1 parameters available else *(-220) Error code. |

Conditions (ADTS552F): Only Channel 1 parameters available else *(-220) Error code.
Conditions (ADTS553F): Channel 2 only ALT2 and PS2 available else *(-220) Error code.
Conditions (ADTS554F): All parameters valid

### 8.7.13 SOUR:RATE

**Command**

| | |
|---|---|
| Syntax: | SOURce:RATE <parameter>,<aim> |
| Parameter Type: | DISCRETE,REAL |
| Parameter Range: | ALT \| CAS \| TAS \| MACH \| EPR \| IN \| OUT \| PS \| PT \| QC |
| | ALT1 \| CAS1 \| TAS1 \| MACH1 \| PS1 \| PT1 \| QC1 |
| | ALT2 \| CAS2 \| TAS2 \| MACH2 \| PS2 \| PT2 \| QC2 |
| | See CALC:LIM:MAX:RATE? for upper limit; lower limit is 0. |
| Units: | As set by UNIT:AER or UNIT:PRES as appropriate. |
| Password: | No |
| Function: | To command the rate of change of the selected parameter to a new rate aim value. |
| Conditions: | Controllers must be ON before using this command. (-221). |
| | * (-220) EPR\|IN\|OUT option must be enabled to read EPR measurement. Access Error. |
| | *(-221) Must be at ground to enter EPR mode. Settings conflict error. |
| Conditions (ADTS542F): | Only Channel 1 parameters available else *(-220) Error code. |
| Conditions (ADTS552F): | Only Channel 1 parameters available else *(-220) Error code. |
| Conditions (ADTS553F): | Channel 2 only ALT2 and PS2 available else *(-220) Error code. |
| Conditions (ADTS554F): | All parameters valid |

**Query**

| | |
|---|---|
| Syntax: | SOURce:RATE? <parameter> |
| Parameter Type: | DISCRETE |
| Parameter Range: | ALT \| CAS \| TAS \| MACH \| EPR \| IN \| OUT \| PS \| PT \| QC |
| | ALT1 \| CAS1 \| TAS1 \| MACH1 \| PS1 \| PT1 \| QC1 |
| | ALT2 \| CAS2 \| TAS2 \| MACH2 \| PS2 \| PT2 \| QC2 |
| Returned Data: | <aim> |
| Returned Data Type: | REAL |
| Returned Data Range: | See CALC:LIM:MAX:PRES? and CALC:LIM:MIN:PRES? |
| Units: | As set by UNIT:AER or UNIT:PRES as appropriate. |
| Function: | To query the last rate aim command. |
| Conditions: | See CALC:LIM:MAX:RATE? for maximum limits. |
| | *(-220) EPR\|IN\|OUT option must be enabled to read EPR measurement. Access Error. |
| | *(-221) Must be at ground to enter EPR mode. Settings conflict error. |
| Conditions (ADTS542F): | Only Channel 1 parameters available else *(-220) Error code. |
| Conditions (ADTS552F): | Only Channel 1 parameters available else *(-220) Error code. |
| Conditions (ADTS553F): | Channel 2 only ALT2 and PS2 available else *(-220) Error code. |
| Conditions (ADTS554F): | All parameters valid |

### 8.7.14 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. (-220)<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>**Note:** 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 |

## 8.8 STATus

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

### 8.8.1 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: | 0-65535 |
| Units: | |
| Function: | To query the status of the ADTS; this data is unlatched. Table 6-3 shows the function of each bit. |
| Conditions: | None |

### 8.8.2 STAT:OPER:ENAB

**Command**

| | |
|---|---|
| Syntax: | STATus:OPERation:ENABle <data> |
| Parameter Type: | INTEGER |
| Parameter Range: | 0-65535 |
| 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.<br>Table 6-3 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: | 0-65535 |
| Units: | |
| Function: | To query the state of the Operation Event register. |
| Conditions: | None |

### 8.8.3 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: | 0-65535 |
| 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 6-3 shows the details of each bit. |
| Conditions: | None |

### 8.8.4 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: | 0-65535 |
| Units: | |
| Function: | To query the Questionable Data Condition register; this data is unlatched<br>Bit 0-7: Reserved<br>Bit 8: Cal (Auto zero) in progress<br>Bit 9: Warm-up. Bit 9 set to 1 during the warm-up period following power-on. After the warm-up period bit 9 changes to 0.<br>Bit 10-15: Reserved |
| Conditions: | None |

### 8.8.5 STAT:QUES:ENAB

**Command**

| | |
|---|---|
| Syntax: | STATus:QUEStionable:ENABle <data> |
| Parameter Type: | INTEGER |
| Parameter Range: | 0-65535 |
| 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. |
| Conditions: | None. |

**Query**

| | |
|---|---|
| Syntax: | STATus:QUEStionable:ENABle? |
| Parameter Type: | None |
| Parameter Range: | |
| Returned Data: | <data> |
| Returned Data Type: | INTEGER |
| Returned Data Range: | 0-65535 |
| Units: | |
| Function: | To query the state of the Questionable Data Enable register. |
| Conditions: | None |

### 8.8.6 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: | 0-65535 |
| Units: | |
| Function: | To query the Questionable Data register. This data is latched and will be cleared by this query or by *CLS. |
| Conditions: | None |

## 8.9 SYSTem

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

### 8.9.1 SYST:DATE

**Command**

| | |
|---|---|
| Syntax: | SYSTem:DATE <yyyy>,<mm>,<dd> |
| Parameter Type: | INTEGER, INTEGER, INTEGER |
| Parameter Range: | 2000->, 1 to 12, 1 to 31 |
| Units: | <yyyy>,<mm>,<dd> |
| Password: | - |
| Function: | To set the system date. |
| Conditions: | None |

**Query**

| | |
|---|---|
| Syntax: | SYSTem:DATE? |
| Parameter Type: | None |
| Parameter Range: | |
| Returned Data: | data |
| Returned Data Type: | INTEGER, INTEGER, INTEGER |
| Returned Data Range: | 2000->, 1-12, 1-31 |
| Units: | <yyyy>,<mm>,<dd> |
| Function: | To query system date. |
| Conditions: | None |

### 8.9.2 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.<br>**Note:** <err text> is returned in “double quotes”. |
| Conditions: | None |

### 8.9.3 SYST:PASS:CDIS

**Command**

| | |
|---|---|
| Syntax: | SYSTem:PASSword:CDISable <Password> |
| Parameter Type: | String |
| Parameter Range: | |
| Units: | |
| Password: | No |
| Function: | Disables some commands with a password (see command summary). |
| Conditions: | Using an invalid password causes an error -221.<br>Password contains a maximum of 32 characters.<br>Passwords are case sensitive. |

**Query**

| | |
|---|---|
| Syntax: | N/A |
| Parameter Type: | |
| Parameter Range: | |
| Returned Data: | |
| Returned Data Type: | |
| Returned Data Range: | |
| Units: | |
| Function: | |
| Conditions: | |

### 8.9.4 SYST:PASS:CEN

**Command**

| | |
|---|---|
| Syntax: | SYSTem:PASSword:CENable <Password> |
| Parameter Type: | String |
| Parameter Range: | |
| Units: | |
| Password: | No |
| Function: | Enables commands normally protected by passwords. |
| Conditions: | Using an invalid password causes an error -221.<br>Password contains a maximum of 32 characters.<br>Passwords are case sensitive. |

**Query**

| | |
|---|---|
| Syntax: | N/A |
| Parameter Type: | |
| Parameter Range: | |
| Returned Data: | |
| Returned Data Type: | |
| Returned Data Range: | |
| Units: | |
| Function: | |
| Conditions: | |

### 8.9.5 SYST:PASS:NEW

**Command**

| | |
|---|---|
| Syntax: | SYSTem:PASSword:NEW <current>,<new> |
| Parameter Type: | String (current), String (new) |
| 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.<br>**Note:** Both current and new passwords are case sensitive.<br>**Note:** The method to reset the SCPI password from a previously set value which has been forgotten. To set a new password with the OLD password value in the command set to "DEFAULT" e,g.: SYST:PASS:NEW "DEFAULT","NEWDEFAULT" |
| 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: | |

### 8.9.6 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: | State |
| Returned Data Type: | INTEGER |
| Returned Data Range: | 0 \| 1 |
| Units: | |
| Function: | Returns 0 when password disables protected commands (password locks out commands).<br>Returns 1 when password enables protected commands. |
| Conditions: | None |

### 8.9.7 SYST:PRES

**Command**

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

**Query**

| | |
|---|---|
| Syntax: | N/A |
| Parameter Type: | |
| Parameter Range: | |
| Returned Data: | |
| Returned Data Type: | |
| Returned Data Range: | |
| Units: | |
| Function: | |
| Conditions: | |

### 8.9.8 SYST:TIME

**Command**

| | |
|---|---|
| Syntax: | SYSTem:TIME <hh>,<mm>,<ss> |
| Parameter Type: | Integer (hh), Integer (mm), Integer (ss) |
| Parameter Range: | 0 to 23, 0 to 59, 0 to 59 |
| Units: | |
| Password: | No |
| Function: | To set the system time. |
| Conditions: | None |

**Query**

| | |
|---|---|
| Syntax: | SYSTem:TIME? |
| Parameter Type: | None |
| Parameter Range: | |
| Returned Data: | time <hh>,<mm>,<ss> |
| Returned Data Type: | Integer (hh), Integer (mm), Integer (ss) |
| Returned Data Range: | 0 to 23, 0 to 59, 0 to 59 |
| Units: | |
| Function: | To query system time. |
| Conditions: | None |

### 8.9.9 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: | 1995 |
| Units: | |
| Function: | To query SCPI standard version. |
| Conditions: | None |

## 8.10 UNITs

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

### 8.10.1 UNIT:AER

**Command**

| | |
|---|---|
| Syntax: | UNITs:AERonautical <units> |
| Parameter Type: | DISCRETE |
| Parameter Range: | FTKNTS\|FTMPH\|MKNTS\|MKPH |
| 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. |
| Conditions: | None |

**Query**

| | |
|---|---|
| Syntax: | UNITs:AERonautical? |
| Parameter Type: | None |
| Parameter Range: | |
| Returned Data: | <units> |
| Returned Data Type: | DISCRETE |
| Returned Data Range: | FTKNTS\|FTMPH\|MKNTS\|MKPH |
| Units: | |
| Function: | To query the SCPI aeronautical units. |

| Aeronautical Units | Selection |
|---|---|
| FTKNTS | Selects feet and knots (feet/minute) |
| FTMPH | Selects feet and miles per hour |
| MKPH | Selects meters and kilometers per hour (meters/minute) |
| MKPH (M/min) | Selects meters and kilometers per hour (meters/minute) |

### 8.10.2 UNIT:PRES

**Command**

| | |
|---|---|
| Syntax: | UNITs:PRESsure <units> |
| Parameter Type: | DISCRETE |
| Parameter Range: | BAR \| CMH2O20 \| CMH2O4 \| CMHG \| FTH2O20 \| FTH2O4 \| FTH2O60 \| HPA \| INH2O20 \| INH2O4 \| INH2O60F \| INHG \| KGCM2 \| KPA \| LBFT2 \| MBAR \| MH2O20 \| MH2O4 \| MHG \| MMH2O20 \| MMH2O4 \| MMHG \| MPA \| PA \| PSI \| TORR \| ATM |
| Units: | |
| Password: | No |
| Function: | To set the pressure units used for SCPI commands and queries. |
| Conditions: | None |

**Query**

| | |
|---|---|
| Syntax: | UNITs:PRESsure? |
| Parameter Type: | None |
| Parameter Range: | |
| Returned Data: | <units> |
| Returned Data Type: | DISCRETE |
| Returned Data Range: | BAR \| CMH2O20 \| CMH2O4 \| CMHG \| FTH2O20 \| FTH2O4 \| FTH2O60 \| HPA \| INH2O20 \| INH2O4 \| INH2O60F \| INHG \| KGCM2 \| KPA \| LBFT2 \| MBAR \| MH2O20 \| MH2O4 \| MHG \| MMH2O20 \| MMH2O4 \| MMHG \| MPA \| PA \| PSI \| TORR \| ATM |
| Units: | |
| Function: | To query the SCPI pressure units. |
| Conditions: | None |

### 8.10.3 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 |

## 8.11 *Standard Commands

The commands starting with * are SCPI standard commands.

### 8.11.1 *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: | |

### 8.11.2 *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 6-2 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 |

### 8.11.3 *ESR

**Command**

| | |
|---|---|
| Syntax: | N/A |
| Parameter Type: | |
| Parameter Range: | |
| Units: | |
| Password: | |

**Query**

| | |
|---|---|
| Syntax: | *ESR? |
| Parameter Type: | None |
| Parameter Range: | |
| Returned Data: | <data> |
| Returned Data Type: | INTEGER |
| Returned Data Range: | 0-255 |
| Units: | |
| Function: | To query the value of the Standard Event Register.<br>The data is latched and will be cleared by this query.<br>Table 6-2 shows the function of each bit. |
| Conditions: | None |

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

### 8.11.4 *ID2?

**Command**

| | |
|---|---|
| Syntax: | N/A |
| Parameter Type: | |
| Parameter Range: | |
| Units: | |
| Password: | |
| Function: | |
| Conditions: | |

**Query**

| | |
|---|---|
| Syntax: | *ID2? |
| Parameter Type: | None |
| Parameter Range: | |
| Returned Data: | <Manufacturer>, <Model>, <Serial No.>, <Software Version> |
| Returned Data Type: | STRING |
| Returned Data Range: | |
| Units: | |
| Function: | To query identification of the hand terminal |
| Conditions: | None |
| Example: | |

### 8.11.5 *IDN

**Command**

| | |
|---|---|
| Syntax: | N/A |
| Parameter Type: | |
| Parameter Range: | |
| Units: | |
| Password: | |
| Function: | |
| Conditions: | |

**Query**

| | |
|---|---|
| Syntax: | *IDN? |
| Parameter Type: | None |
| Parameter Range: | |
| Returned Data: | <Manufacturer>, <Model>, <Serial No.>, <Software Version> |
| Returned Data Type: | STRING |
| Returned Data Range: | |
| Units: | |
| Function: | To query identification of ADTS Controller device. |
| Conditions: | None |

### 8.11.6 *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 |

### 8.11.7 *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, Integer, Integer, Integer |
| Returned Data Range: | 0 or 1 |
| Units: | |
| Function: | Returns option information, this is a sequence of INTEGER values as follows:<br>EPR option enabled, Bluetooth option enabled, EALT option enabled, ADSB option enabled, FL280 option enabled. |
| Conditions: | None |

### 8.11.8 *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 | |

### 8.11.9 *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 6-4 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. |

### 8.11.10 *STB

**Command**

| | |
|---|---|
| Syntax: | N/A |
| Parameter Type: | |
| Parameter Range: | |
| Units: | |
| Password: | |
| Function: | |
| Conditions: | |

**Query**

| | |
|---|---|
| Syntax: | *STB? |
| 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 Status Byte.<br>Table 6-4 shows the function of each bit that represents each status byte. |
| Conditions: | None |

### 8.11.11 *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: | This command, although recognized by the system, is ignored. Returns 1 |
| Conditions: | |

### 8.11.12 *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

# 9. Error Numbers

Standard SCPI errors use error numbers, also called error codes. Negative error numbers are used for standard SCPI errors and events. Positive error numbers are reserved for device-specific errors. Each error number is followed by a message describing the error. When an error is detected, it is held in the error queue. When SYST:ERR? is sent, any error in the error queue sets the error bit in the event status register (*ESR).

**Note:** System errors in the range of +200 to +299 will only be put in the error/event queue and do not set any bits in the event status register.

See “Standard Event Group” on page 18 for more details.

## 9.1 Error Numbers -199 to -100 (Command Errors)

**Table 9-1: Command Errors**

| Error Number | Error Message |
|---|---|
| -100 | -100,“Command error” |
| -102 | -102,“Syntax error” |
| -104 | -104,“Data type error” |
| -109 | -109,“Missing parameter” |
| -112 | -112,“Program mnemonic too long” |
| -113 | -113,“Undefined header” |
| -114 | -114,“Header suffix out of range” |
| -120 | -120,“Numeric data error” |
| -121 | -121,“Invalid character in number” |
| -131 | -131,“Invalid suffix” |
| -140 | -140,“Character data error” |
| -144 | -144,“Character data too long” |
| -160 | -160,“Block Data Error” |
| -161 | -161,“Invalid Block Data” |

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

**Table 9-2: Execution Errors**

| Error Number | Error Message |
|---|---|
| -200 | -200,“Execution error” |
| -203 | -203,“Command protected” |
| -220 | -220,“Parameter error” |
| -221 | -221,“Settings conflict” |
| -222 | -222,“Data out of range” |
| -223 | -223,“Too much data” |
| -224 | -224,“Illegal parameter value” |
| -240 | -240,“Hardware error” |
| -241 | -241,“Hardware missing” |
| -255 | -255, “Directory full” |
| -287 | -287, “Option not enabled” |

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

**Table 9-3: Device Specific Errors**

| Error Number | Error Message |
|---|---|
| -300 | -300,“Device - specific error” |
| -310 | -310,“System error” |
| -311 | -311,“Memory Error” |
| -350 | -350,“Queue overflow” |

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

**Table 9-4: Query Errors**

| Error Number | Error Message |
|---|---|
| -400 | -400,“Query Error” |
| -440 | -410,“Query UNTERMINATED after indefinite response" |

## 9.5 Error Numbers -599 to -500 (Power on Event)

**Table 9-5: Power on Event**

| Error Number | Error Message |
|---|---|
| -500 | -500,“Power On” |

## 9.6 Error Numbers +200 to +299 (Device Specific Errors)

**Table 9-6: Device Specific Errors**

| Error Number | Error Message |
|---|---|
| 201 | 201, “Query only” |
| 202 | 202, “No Query Allowed” |
| 203 | 203, “Incorrect Password” |
| 204 | 204, “Constant not allowed in STATUS subsystem” |
| 207 | 207, “Enumerated value not in union” |
| 208 | 208, “Illegal number of parameters” |
| 210 | 210, “Run out of memory handle” |
| 211 | 211, “Unit not matched” |
| 212 | 212, “Unit not required” |
| 213 | 213, “Ps Capped” |
| 214 | 214, “Pt Capped” |
| 215 | 215, “Qc Capped” |
| 216 | 216, “Alt Capped” |

**Table 9-6: Device Specific Errors**

| Error Number | Error Message |
|---|---|
| 217 | 217, “MACH Capped” |
| 218 | 218, “CAS Capped” |
| 219 | 219, “EPR In Capped” |
| 220 | 220, “EPR Out Capped” |
| 221 | 221, “EPR Ratio Capped” |
| 222 | 222, “Ps Rate Capped” |
| 223 | 223, “Pt Rate Capped” |
| 224 | 224, “Qc Rate Capped” |
| 225 | 225, “Rate of Climb Capped” |
| 226 | 226, “CAS Rate Capped” |
| 227 | 227, “EPR In Rate Capped” |
| 228 | 228, “EPR Out Rate Capped” |
| 229 | 229, “EPR Ratio Capped” |
| 230 | 230, “MACH Rate Capped” |

# 10. Calibration

## 10.1 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.

## 10.2 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. SYST:PASS:CEN <password>

   Send the calibration password, usually "DEFAULT"

3. SOUR:LIM:AIRC "MAX AERO"

   Set the limits to Max Aero

4. UNITS:PRES mbar

   Set the ADTS to mbar pressure units for calibration.

5. SOUR:STAT ON

   Turn on the controllers for internal pressure generation, if an external pressure source is being used leave the controllers off (SOUR:STAT OFF).

6. CAL:MAIN:CHAN xx

   Enter main calibration mode for channel xx (PS, PS1, PS2, PT, PT1 or PT2).

7. SOUR:RATE xx, <rate>

   Set the rate for the channel.

8. SOUR:PRES xx, <aim>

   Apply the minimum adjustment point for the channel, as detailed in the user manual.

   Once the set point is reached wait for at least 1 minute for the pressure to stabilize.

9. CAL:MAIN:VAL <value>

   Enter pressure indicated by the pressure standard.

   Repeat steps 6 to 8. At least two different pressure points must be entered for the ADTS to calculate the slope and zero values, although we recommend using at least five different pressure points.

10. CAL:MAIN:ACC YES

    This takes two seconds to accept the changes. Also, it stores the current date (set in step 1) and backs up the new calibration data.

11. We recommend now disabling the calibration password using SYST:PASS:CDIS <password>, the unit is returned to ground conditions SOUR:GTGR and local mode is returned to the hand terminal LOC.

12. **Power cycle** the unit after calibration before continuing with testing.

## 10.3 ALL Channel Calibration

Dual channel calibration configures the ADTS5xxF so that all channels connect internally to PT1. By connecting to PT1 and applying 3 pressure points it is possible to calibrate all channels at the same time with only an external pressure reference. The ADTS will internally generate the necessary set point pressures.

Using this mode can leave the ADTS5xxF with trapped internal pressure, so it is important to return to between 92 & 1130 mbar before finishing the 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.
2. SYST:PASS:CEN <password>

   Send the calibration password, usually "DEFAULT"
3. SOUR:LIM:AIRC <maximum limits>

   Set the limits to "MAX AERO" for ADTS542F or "MAX ADTS" for all ADTS55xF
4. UNITS:PRES mbar

   Set the ADTS to mbar pressure units for calibration.
5. SOUR:STAT ON

   Turn on the controllers for internal pressure generation, if an external pressure source is being used leave the controllers off SOUR:STAT OFF
6. CLA:MAIN:CHAN PSPT

   Enter main calibration mode for all channel calibration.
7. SOUR:RATE PT, <rate>

   Set the rate for the channel.
8. SOURCE:PRES PT, <minimum pressure>

   Apply the minimum adjustment point for all channels, as detailed in the user manual.

   Once the set point is reached wait for at least 1 minute for the pressure to stabilize.
9. CAL:MAIN:VAL <measured reading>

   Send the measured pressure from the reference at this calibration point.
10. SOURCE:PRES PT, <maximum Ps pressure>

    Apply the maximum adjustment point for the Ps channels, as detailed in section the user manual.

    Once the set point is reached wait for at least 1 minute for the pressure to stabilize.
11. CAL:MAIN:VAL <measured reading>

    Send the measured pressure from the reference at this calibration point.
12. SOURCE:PRES PT, <maximum Pt pressure>

    Apply the maximum adjustment point for the Pt channels, as detailed in the user manual.

    Once the set point is reached wait for at least 1 minute for the pressure to stabilize.
13. CAL:MAIN:VAL <measured reading>

    Send the measured pressure from the reference at this calibration point.
14. SOURCE:PRES PT,1013

    Before completing the calibration, return the unit to normal operating range for all channels. This is between 92 mbar and 1130 mbar.

15. CAL:MAIN:ACC <YES or NO>

    Choose to accept the calibration or not.

    This takes two seconds to accept the changes. Also, it stores the current date (set in step 1) and backs up the new calibration data.

16. We recommend now disabling the calibration password using SYST:PASS:CDIS <password>, the unit is returned to ground conditions SOUR:GTGR and local mode is returned to the hand terminal LOC.
17. **Power cycle** the unit after calibration before continuing with testing.

# Chapter 10. Calibration

## Office Locations

https://druck.com/contact

## Services and Support Locations

https://druck.com/service



189M2378 Revision - | English

Druck

druck.com