

Visit **TRANSCAT**.com >

[sales@transcat.com](mailto:sales@transcat.com) | 800.828.1470

# Communication Protocol Manual JOFRA ATC

© Copyright 2007 AMETEK Denmark A/S



# Contents

---

<b>1</b>	<b>Introduction.....</b>	<b>6</b>
<b>2</b>	<b>Protocol.....</b>	<b>7</b>
	2.1 Variables.....	7
	2.2 Telegram structure.....	8
	2.3 Packing and unpacking of telegrams.....	9
	2.4 Error detection.....	9
	2.5 Establishment of connection.....	10
	2.6 Examples of the content of telegrams.....	10
<b>3</b>	<b>RS232 connection.....</b>	<b>11</b>
	3.1 Communication parameters.....	11
	3.2 Cable.....	11
<b>4</b>	<b>Telegrams.....</b>	<b>12</b>
	Telegram #1 – Log-on.....	12
	Telegram #2 – Log off.....	13
	Telegram #3 – Read temperature and input/output.....	13
	Telegram #4 – Write SET temperature.....	14
	Telegram #5 – Not applicable.....	14
	Telegram #6 – Not applicable.....	14
	Telegram #7 – Not applicable.....	14
	Telegram #8 – Not applicable.....	14
	Telegram #9 – Read serial number.....	14
	Telegram #10 – Not applicable.....	15
	Telegram #11 – Read calibration date (heat source).....	15
	Telegram #12 – Not applicable.....	15
	Telegram #13 – Read temperature unit and resolution.....	15
	Telegram #14 – Write temperature unit.....	16
	Telegram #15 – Write temperature resolution.....	16
	Telegram #16 – Set calibrator to remote mode.....	17
	Telegram #17 – Read maximum SET temperature.....	17
	Telegram #18 – Write maximum SET temperature.....	17
	Telegram #19 – Read slope rate.....	17
	Telegram #20 – Write slope rate.....	17
	Telegram #21 – Read stability time.....	18
	Telegram #22 – Write stability time.....	18
	Telegram #23 – Not applicable.....	19
	Telegram #24 – Not applicable.....	19
	Telegram #25 – Not applicable.....	19
	Telegram #26 – Not applicable.....	19

Telegram #27 – Read maximum temperature .....	19
Telegram #28 – Not applicable .....	19
Telegram #29 – Not applicable .....	19
Telegram #30 – Not applicable .....	19
Telegram #31 – Not applicable .....	19
Telegram #32 – Read status for work orders .....	19
Telegram #33 – Setup work order .....	20
Telegram #34 – Read work order data .....	26
Telegram #35 – Not applicable .....	32
Telegram #36 – Not applicable .....	32
Telegram #37 – Not applicable .....	32
Telegram #38 – Read clock .....	32
Telegram #39 – Write clock .....	33
Telegram #40 – Not applicable .....	33
Telegram #41 – Not applicable .....	33
Telegram #42 – Not applicable .....	33
Telegram #43 – Not applicable .....	33
Telegram #44 – Not applicable .....	33
Telegram #45 – Not applicable .....	33
Telegram #46 – Not applicable .....	33
Telegram #47 – Not applicable .....	33
Telegram #48 – Not applicable .....	34
Telegram #49 – Not applicable .....	34
Telegram #50 – Read temperature scaling .....	34
Telegram #51 – Write temperature scaling .....	34
Telegram #52 – Read cold-junction compensation .....	34
Telegram #53 – Write cold-junction compensation .....	35
Telegram #54 – Read sensor under test parameters .....	35
Telegram #55 – Write sensor under test parameters .....	37
Telegram #56 – Read reference sensor parameters .....	39
Telegram #57 – Write reference sensor parameters .....	39
Telegram #58 – Not applicable .....	40
Telegram #59 – Not applicable .....	40
Telegram #60 – Not applicable .....	40
Telegram #61 – Not applicable .....	40
Telegram #62 – Not applicable .....	40
Telegram #63 – Not applicable .....	40
Telegram #64 – Not applicable .....	40
Telegram #65 – Not applicable .....	40
Telegram #66 – Read external reference sensor data .....	40
Telegram #67 – Write external reference sensor data .....	41
Telegram #68 – Not applicable .....	42

Telegram #69 – Not applicable.....	42
Telegram #70 – Not applicable.....	42
Telegram #71 – Not applicable.....	42
Telegram #72 – Not applicable.....	42
Telegram #73 – Not applicable.....	42
Telegram #74 – Not applicable.....	42
Telegram #75 – Not applicable.....	42
Telegram #76 – Not applicable.....	42
Telegram #77 – Not applicable.....	42
Telegram #78 – Not applicable.....	42
Telegram #79 – Not applicable.....	42
Telegram #80 – Read calibration date for SUT inputs.....	42
Telegram #81 – Write calibration date for SUT inputs.....	43
Telegram #82 – Not applicable.....	43
Telegram #83 – Delete work order .....	43
Telegram #84 – Read test mode .....	43
Telegram #85 – Not applicable.....	44
Telegram #86 – Not applicable.....	44
Telegram #87 – Read slope rate status.....	44
Telegram #88 – Write slope rate status.....	44
Telegram #93 – Not applicable.....	44
Telegram #94 – Not applicable.....	44
Telegram #95 – Not applicable.....	44
Telegram #96 – Not applicable.....	44

# 1 Introduction

---

This document describes the serial communication between the PC and the Jofra ATC calibrators. The communication is based on telegrams complying with Ametek's ADK-protocol. The communication uses a serial RS232 connection.

## 2 Protocol

---

The ADK-protocol is based on the Master/Slave principle. The PC is always to be regarded as the master and the calibrator as the slave. This means that the calibrator never starts the communication for itself, but solely replies to demands from the PC.

When talking about writing and reading, the communication is seen from the PC's point of view, which means that writing is information from the PC to the calibrator, whereas reading is information from the calibrator to the PC.

### 2.1 Variables

All variables are stated in the format Big Indian, i.e. starting with the most significant variables followed by the less significant ones etc. (contrary to the Intel notation).

The following designations are used in this manual:

byte	1 byte unsigned integer, value 0 – 255.
char	1 byte unsigned integer, value 0 – 255.
word	2 bytes unsigned integer, value 0-65,535.
unsigned Int.	2 bytes unsigned integer, value 0-65,535.
integer	2 bytes signed integer, value –32,768 – 32,767.
long	4 bytes signed integer, value –2,147,483,648 – 2,147,483,647
unsign. long	4 bytes unsigned integer, value 0 – 4,294,967,295.
float	4 bytes floating point number, complying with the IEEC standard.
double	8 bytes floating point number, complying with the IEEC standard.
boolean	1 byte flag, value 0=False, 1=True.
string[X]	Array of X characters, occupying X+1 bytes. The last byte is always zero.

Numbers followed by the letter h (e.g. 07h) indicate a hexadecimal number.

## 2.2 Telegram structure

A telegram is constructed as a serial flow of bytes:

<b>Section:</b>	<b>Length:</b>	<b>Description:</b>
Telegram number	2 bytes (unsigned int)	States the telegram type
Data bytes	Depends on the telegram type.	If a telegram contains data, they are placed here.
CRC-checksum	2 bytes (unsigned int)	CRC-sum for the sections: Telegram type & data bytes.



## 2.3 Packing and unpacking of telegrams

Before transmitting the telegram, it is packed to ensure that the end of the telegram can be detected. The character EOT = 04h must always terminate the telegram, and therefore no other bytes in the telegram must have this value. To prevent this, a byte with the value 04h is translated into two characters 1Bh + FCh, and if the telegram contains a byte with the value 1Bh, this is translated into the two characters 1Bh + E5h. The telegram will thus consist of a number of bytes different from the value 04h and terminated with the value 04h.

## 2.4 Error detection

The error detection ensures that defective telegrams can be refused. A simple kind of CRC-checksum is used. At the start, the CRC-checksum of the type unsigned int has the value 0. The CRC sum does not include the last 3 bytes of a telegram. Neither is the EOT character included. The following routine is carried out for each telegram byte:

1. CRC-checksum = CRC-checksum Xor (telegram byte × 256)
2. To be repeated 8 times:
  - If the CRC-checksum > 7FFFh:  
CRC-checksum = (2 × CRC-checksum) Xor 8005h
  - If not:  
CRC-checksum = 2 × CRC-checksum

The following code example is a function written in C, generating the CRC-checksum:

```
unsigned int calcCrc(unsigned int crcSum, unsigned char
datachar)
{
    unsigned char count;

    crcSum = crcSum ^ (datachar * 256);

    for (count = 0; count <= 7; count++)
    {
        if (crcSum > 0x7FFF)
            crcSum = (crcSum * 2) ^ 0x8005;
        else
            crcSum = crcSum * 2;
    }
    return crcSum;
}
```

}

## 2.5 Establishment of connection

A telegram from the PC is answered by a telegram with the same telegram type, if necessary “empty” (i.e. without data bytes), to acknowledge the receipt or containing specific data. If a telegram is received with errors, it is entirely ignored by the recipient. The PC waits for at least 1 second after sending the telegram, until the time-out occurs and another transmission from the PC can be tried. After 3 attempts, the transmission is off and the connection is regarded as interrupted. Afterwards you may try to establish a new connection (by transmitting a log-on telegram).

## 2.6 Examples of the content of telegrams

In general a telegram consists of

<Telegram number (2 bytes)>, <Data bytes (x bytes)>, <CRC check sum (2 bytes)>, <04 (1 byte)>

As an illustration the log-on telegram is composed as

<01 (2 bytes)>, <CRC (2 bytes)>, <04 (1 byte)>

The log-on telegram has no data bytes. The CRC checksum is calculated as the CRC sum of the telegram number and the data bytes.

Note: If the composed telegram holds the byte value 04 hex, this byte is substituted by the escape sequences explained in section '2.3 Packing and unpacking of telegrams'. I.e. the finished telegram does not hold the value 04 hex, except the final byte, which is always 04 hex to indicate the end of the telegram.

Another example, telegram #6 - write coefficients

<06 (2 bytes)>, <12 bytes data>, <2 bytes CRC>, <04 (1 byte)>

## 3 RS232 connection

---

### 3.1 Communication parameters

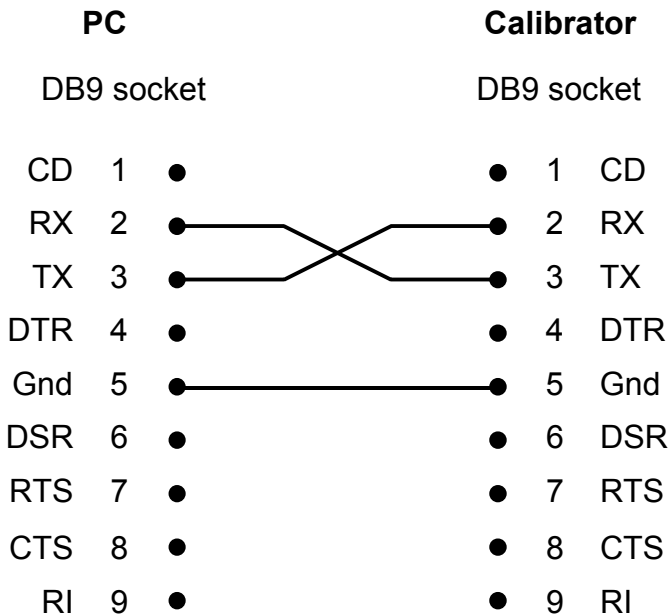
The serial communication is based on a standard RS232 communication. The communication parameters are specified as follows:

- 9600 baud
- 8 data bits
- No parity
- 1 stop bit

No hardware handshake is used.

### 3.2 Cable

The cable between the PC and the calibrator is designed in the following way:



## 4 Telegrams

---

The communication always starts with the PC setting the calibrator in communication mode (Log-on). At the same time any existing functions are interrupted. As long as the calibrator is in communication mode, the user interface is disconnected. This is shown by the text "REMOTE" in the display of the calibrator.

When the PC logs off, the user interface of the calibrator is reactivated.

### Telegram #1 – Log-on

Number: 1  
Subject: The connection between the PC and the calibrator is established.  
Reading/writing: Reading  
Comments: The returned data identify the instrument type, the protocol consulted and the version number of the software of the calibrator. The calibrators user interface is disabled.  
Data, PC → calibrator: -  
Data, PC ← calibrator: 6 bytes:  
Instrument type (unsigned int):  
3021: ATC-155A  
3022: ATC-320A  
3023: ATC-650A  
3024: ATC-156A  
3025: ATC-157A  
3026: ATC-125A  
3027: ATC-140A  
3028: ATC-250A  
3121: ATC-155B  
3122: ATC-320B  
3123: ATC-650B  
3124: ATC-156B  
3125: ATC-157B  
3126: ATC-125B  
3127: ATC-140B  
3128: ATC-250B

Protocol version (unsigned int): 101 (v1.01)  
Software version (unsigned int): 100 (v1.00)

Example: <Telegram No.(=01)>, <Instrument type (=3021)> , <Protocol version (=101)> , <Software version (=100)>

## Telegram #2 – Log off

Number: 2  
Subject: The communication is interrupted.  
Reading/writing: -  
Comments: The remote mode is closed and the user interface of the calibrator is activated. If the slope rate is active it is disabled.  
Data, PC → calibrator: -  
Data, PC ← calibrator: -

## Telegram #3 – Read temperature and input/output

Number: 3  
Subject: Temperature and input/output  
Reading/writing: Reading  
Data, PC → calibrator: -  
Data, PC ← calibrator: 33 bytes.  
SET temperature (Celsius) (float)  
READ temperature (Celsius) (float)  
TRUE temperature (Celsius) (float)  
SENSOR temperature (Celsius) (float)  
TRUE input (float) (ohm)  
SENSOR input (float) (mA, mV, V, ohm)  
SENSOR measure unit (byte)  
0: mA  
1: mV  
2: V  
3: ohm  
4: Switch test input (SENSOR input=0 means input is open, i.e. no value is valid)  
5: Manual input (Only temperature, SENSOR input value is not valid)

READ/TRUE stability (byte)

Not implemented. Reserved for future use.

SENSOR stability (byte)

Not implemented. Reserved for future use.

READ/TRUE (integer) stability time, i.e. the remaining time until stability is expected or the time consumed since stability was accomplished.

SENSOR (integer) stability time, i.e. the remaining time until stability is expected or the time consumed since stability was accomplished.

Switch input closed (boolean)

SYNC output active (boolean)

## **Telegram #4 – Write SET temperature**

Number: 4  
Subject: SET temperature  
Reading/writing: Writing  
Data, PC → calibrator: 4 bytes:  
SET temperature in °C (float)  
Data, PC ← calibrator: -

## **Telegram #5 – Not applicable**

## **Telegram #6 – Not applicable**

## **Telegram #7 – Not applicable**

## **Telegram #8 – Not applicable**

## **Telegram #9 – Read serial number**

Number: 9  
Subject: Serial number  
Reading/writing: Reading  
Data, PC → calibrator: -  
Data, PC ← calibrator: 13 bytes:

Serial number (string[12])

## **Telegram #10 – Not applicable**

## **Telegram #11 – Read calibration date (heat source)**

Number: 11  
Subject: Calibration date  
Reading/writing: Reading  
Data, PC → calibrator: -  
Data, PC ← calibrator: 4 bytes:  
Day, 1..31 (char)  
Month, 1..12 (char)  
Year, 1998..2025 (unsigned int)

## **Telegram #12 – Not applicable**

## **Telegram #13 – Read temperature unit and resolution**

Number: 13  
Subject: Temperature unit and resolution  
Reading/writing: Reading  
Data, PC → calibrator: -  
Data, PC ← calibrator: 5 bytes:  
Temperature unit (byte):  
0: °C  
1: °F  
2: K  
SET temperature resolution (byte):  
0: 1°  
1: 0,1°  
2: 0,01°  
READ temperature resolution (byte):  
0: 1°  
1: 0,1°  
2: 0,01°  
TRUE temperature resolution (byte):  
0: 1°  
1: 0,1°  
2: 0,01°

SENSOR temperature resolution (byte):

- 0: 1°
- 1: 0,1°
- 2: 0,01°

## Telegram #14 – Write temperature unit

Number: 14  
Subject: Temperature unit  
Reading/writing: Writing  
Data, PC → calibrator: 1 byte:  
Temperature unit (byte):  
0: °C  
1: °F  
2: K  
Data, PC ← calibrator: -

## Telegram #15 – Write temperature resolution

Number: 15  
Subject: Temperature resolution  
Reading/writing: Writing  
Data, PC → calibrator: 4 bytes:  
SET temperature resolution (byte):  
0: 1°  
1: 0,1°  
2: 0,01°  
READ temperature resolution (byte):  
0: 1°  
1: 0,1°  
2: 0,01°  
TRUE temperature resolution (byte):  
0: 1°  
1: 0,1°  
2: 0,01°  
SENSOR temperature resolution (byte):  
0: 1°  
1: 0,1°  
2: 0,01°  
Data, PC ← calibrator: -



## **Telegram #16 – Set calibrator to remote mode**

Number: 16  
Subject: Activate remote mode for calibrator  
Reading/writing: Writing  
Data, PC → calibrator: -  
Data, PC ← calibrator: -  
Comment: When the calibrator is not in remote mode it is only possible to use telegrams which read data. The calibrator ignores telegrams which write data.

## **Telegram #17 – Read maximum**

### **SET temperature**

Number: 17  
Subject: Maximum SET temperature permitted  
Reading/writing: Reading  
Data, PC → calibrator: -  
Data, PC ← calibrator: 4 bytes:  
Max. temperature in °C (float)

## **Telegram #18 – Write maximum SET temperature**

Number: 18  
Subject: Maximum SET temperature permitted  
Reading/writing: Writing  
Data, PC → calibrator: 4 bytes:  
Max. temperature in °C (float)  
Data, PC ← calibrator: -

## **Telegram #19 – Read slope rate**

Number: 19  
Subject: Slope rate  
Reading/writing: Reading  
Data, PC → calibrator: -  
Data, PC ← calibrator: 4 bytes:  
Slope rate in °C/min. (float)

## **Telegram #20 – Write slope rate**

Number: 20

Subject: Slope rate. If the slope rate is 0 the default (max) slope rate is used. To activate an on-line switch test first execute the SET command (telegram #4) then write the slope rate (telegram #20). The calibrator uses a negative slope rate if the SET temperature is lower than the actual temperature. (Valid from firmware ver. 1.08)

Reading/writing: Writing

Data, PC → calibrator: 4 bytes:  
Slope rate in °C/min. (float). The valid range is 0.1 – 9.9.

Data, PC ← calibrator: -

### Telegram #21 – Read stability time

Number: 21

Subject: Stability time

Reading/writing: Reading

Data, PC → calibrator: -

Data, PC ← calibrator: 15 bytes:  
 READ extended stability time in minutes (word)  
 TRUE stability time in minutes (word)  
 TRUE interval for temperature stability (Celsius) (float)  
 SENSOR stability time in minutes (word)  
 SENSOR interval for temperature stability (Celsius) (float)  
 SENSOR stability criteria active (boolean)

### Telegram #22 – Write stability time

Number: 22

Subject: Stability time

Reading/writing: Writing

Data, PC → calibrator: 15 bytes:  
 READ extended stability time in minutes (word)  
 TRUE stability time in minutes (word)  
 TRUE interval for temperature stability (Celsius) (float)  
 SENSOR stability time in minutes (word)

SENSOR interval for temperature stability  
(Celsius) (float)  
SENSOR stability criteria active (boolean)

Data, PC ← calibrator: -

### **Telegram #23 – Not applicable**

### **Telegram #24 – Not applicable**

### **Telegram #25 – Not applicable**

### **Telegram #26 – Not applicable**

### **Telegram #27 – Read maximum temperature**

Number: 27  
Subject: Maximum temperature  
Reading/writing: Reading  
Data, PC → calibrator: -  
Data, PC ← calibrator: 8 bytes:  
Max. temperature in °C (float)  
Min. temperature in °C (float)

### **Telegram #28 – Not applicable**

### **Telegram #29 – Not applicable**

### **Telegram #30 – Not applicable**

### **Telegram #31 – Not applicable**

### **Telegram #32 – Read status for work orders**

Number: 32  
Subject: Read the max. number of work orders,  
which can be loaded into the ATC.  
Reading/writing: Reading  
Data, PC → calibrator: 2 bytes:  
Work order number (word)

0: Calibrator returns the max number of work orders, which can be loaded into the calibrator.  
 1..20: Work order number (word)  
 Data, PC ← calibrator: 1 or 2 bytes:  
*Either* - space for work order number xx is occupied (boolean).  
*Or* - the maximum number of work orders, which can be loaded into the calibrator (word).

### Telegram #33 – Setup work order

Number: 33  
 Subject: Work order  
 Reading/writing: Writing  
 Data, PC ↔ calibrator: Number of bytes depends on the data set number (DataID).

Note: The format of this telegram is composed of the telegram identification number 33 followed by a set of data. Several different data sets are defined. A DataID number defines each data set. The work order number must be in the range 1..20.

Example: <Telegram No.(=33)>, <Work orderID(=13)>, <DataID(=03)>, <DTI Used(=00)>, <Ref. sensor(=01)>, <SUT(=00)>, <Temperature unit(=01)>

The defined data sets are as follows (identified by the DataID):

**ID**  
 Data, PC → calibrator: Work order number (integer)  
 DataID=0 (byte)  
 ID, string[30] (31 bytes)  
 AsFound (boolean)  
 AsLeft (boolean)

**Calibration notes**  
 Data, PC → calibrator: Work order number (integer)

DataID=1 (byte)  
NameOfCompany, string[80] (81 bytes)  
Notes, string[255] (256 bytes)

### **Sensor under test**

Data, PC → calibrator:

Work order number (integer)  
DataID=2 (byte)  
ID String[30] (31 bytes)  
Manufacturer, string[30] (31 bytes)  
Serial Number, string[30] (31 bytes)  
Type, string[30] (31 bytes)  
Last calibration date, string[10] (11 bytes)  
Last certificate number, string[30] (31 bytes)  
Pass/fail tolerance (%) (float)  
Pass/fail tolerance (temperature) (float)  
Min. temperature (Celsius) (float)  
Max. temperature (Celsius) (float)  
Tag No. 1, string[30] (31 bytes)  
Tag No. 2, string[30] (31 bytes)  
Tag No. 1 info, string[30] (31 bytes)  
Tag No. 2 info, string[30] (31 bytes)  
Certificate notes, string[30] (31 bytes)  
Notes, string[240] (241 bytes)

### **Scenario**

Data, PC → calibrator:

Work order number (integer)  
DataID=3 (byte)  
Manual heating source used (boolean)  
DTI used (boolean)  
Ref. sensor (byte)

- 0: Internal calibrator sensor
- 1: External sensor connected to calibrator
- 2: External sensor connected to DTI
- 3: External sensor read by operator

Sensor Under Test (byte)

- 0: Calibrator sensor input (RTD, TC, mA, V)
- 1: Switch test input (reserved for future use)
- 2: DTI is connected
- 3: Manual reading by operator

Temperature unit (byte)

- 0: Celsius
- 1: Fahrenheit
- 2: Kelvin

### **Calibrator**

Data, PC → calibrator:

Work order number (integer)

DataID=4 (byte)

ID, string[30] (31 bytes)

Manufacturer, string[30] (31 bytes)

Serial number, string[30] (31 bytes)

Type, string[30] (31 bytes)

Calibrator minimum temperature (Celsius)  
(float)

Calibrator maximum temperature (Celsius)  
(float)

User minimum temperature (Celsius) (float)

User maximum temperature (Celsius) (float)

Number of decimals (byte)

Mains voltage, string[11] (11 bytes)

Mains frequency, string[7] (7 bytes)

Uncertainty (float)

Certificate notes, string[30] (31 bytes)

Notes, string[240] (240 bytes)

### **Manual heating source**

Data, PC → calibrator:

Work order number (integer)

DataID=5 (byte)

ID, string[30] (31 bytes)

Manufacturer, string[30] (31 bytes)

Serial Number, string[30] (31 bytes)

Type, string[30] (31 bytes)

Minimum temperature (Celsius) (float)

Maximum temperature (Celsius) (float)

Number of decimals (byte)

Certificate notes, string[30] (31 bytes)

Notes, string[240] (240 bytes)

DataID=6 reserved for future use.

### **DTI**

Data, PC → calibrator:

Work order number (integer)

DataID=7 (byte)  
Serial number, string[30] (31 bytes)  
Certificate notes, string[30] (31 bytes)  
External reference sensor channel No.  
(0..2) (byte)  
Sensor under test channel No. (0..2) (byte)

### **External reference sensor**

Data, PC → calibrator:  
Work order number (integer)  
DataID=8 (byte)  
ID, string[30] (31 bytes)  
Manufacturer, string[30] (31 bytes)  
Serial Number, string[30] (31 bytes)  
Number of decimals (byte)  
Certificate notes, string[30] (31 bytes)  
Notes, string[240] (240 bytes)

### **Input from sensor under test**

Data, PC → calibrator:  
Work order number (integer)  
DataID=9 (byte)  
ID, string[30] (31 bytes)  
Manufacturer, string[30] (31 bytes)  
Serial number, string[30] (31 bytes)  
Uncertainty, string[60] (60 bytes)  
Type (byte)  
0: 0-4V  
1: 0-12V  
2: 4-20mA  
3: Pt50 M (Russian)  
4: Pt50 P (Russian)  
5: Pt100 IEC  
6: Pt100 Mill  
7: Pt100 M (Russian)  
8: Pt100 P (Russian)  
9: Pt500 IEC  
10: Pt1000 IEC  
11: TC E  
12: TC J  
13: TC K  
14: TC L  
15: TC N  
16: TC R

- 17: TC S
- 18: TC T
- 19: TC U
- 20: TC XK (Russian)
- 21: Pt 50 P6652-94
- 22: Pt 100 P6652-94
- 23: Manual reading (only work order)
- 24: Switch test (reserved for future use)
- 25: No sensor is selected. Display prints 'None'.
- 26: Cu50 (Chinese JJG229 std.)
- 27: Cu100 (Chinese JJG229 std.)
- 28: Pt10 IEC

Number of wires (byte) (Only used by Ptxx sensor types)

Convert to temperature (boolean)

Minimum temperature (Celsius) (float)

Maximum temperature (Celsius) (float)

Minimum measured temperature (Celsius) (float)

Maximum measured temperature (Celsius) (float)

Number of decimals (byte)

Certificate notes, string[30] (31 bytes)

Input text, string[60] (61 bytes)

Auto CJ (Cold Junction) compensation (boolean)

(Type 25-28 are valid from firmware ver. 1.22)

### **Calibration procedure in general**

Data, PC → calibrator:

Work order number (integer)

DataID=10 (byte)

ID, string[30] (31 bytes)

Number of temperature steps (1..20) (byte)

Use Sensor Under Test stability criteria (boolean)

Keep last measured temperature when finished (boolean)

Cooling temperature (Celsius) (float)

Use "Set follows True" correction (boolean)

Use slope (boolean)

Convert external reference sensor to temperature (boolean)



Data, PC → calibrator:

### **Calibration procedure, single step**

Work order number (integer)

DataID=11 (byte)

Line number (1..20) (byte)

SET temperature (Celsius) (float)

Slope rate (degree Celsius per minute), only used by switch test input (float) (reserved for future use)

Reference sensor stability time (minutes) (float)

Reference sensor stability tolerance (Celsius) (float)

Sensor Under Test stability time (minutes) (float)

Sensor Under Test stability tolerance (Celsius) (float)

Heating source (byte)

0: Calibrator

1: Manual heating source

## Telegram #34 – Read work order data

Number: 34  
Subject: Work order  
Reading/writing: Reading  
Data, PC ↔ Calibrator: Depends on work order ID

Note: The format of this telegram is composed of the telegram identification number 34 followed by a set of data. Several different data sets are defined. A DataID number defines each data set. The work order number must be in the range 1..20.

The defined data sets are as follows (identified by the DataID):

Data, PC → Calibrator: **ID**  
Work order number (integer)  
DataID=0 (byte)  
Line number (1..20), (byte), not used  
AsFound/AsLeft, (boolean), not used

Data, PC ← Calibrator: **ID**, string[30] (31 bytes)  
AsFound (boolean)  
AsLeft (boolean)  
Internal mode during run of work order (byte)  
0: Normal mode  
1: Simulation mode  
2: Service mode

### Calibration notes

Data, PC → Calibrator: Work order number (integer)  
DataID=1 (byte)  
Line number (1..20), (byte), not used  
AsFound/AsLeft, (boolean), not used

Data, PC ← Calibrator: NameOfCompany, string[80] (81 bytes)  
Notes, string[255] (255 bytes)

### Sensor under test

Data, PC → Calibrator: Work order number (integer)  
 DataID=2 (byte)  
 Line number (1..20), (byte), not used  
 AsFound/AsLeft, (boolean), not used

Data, PC ← Calibrator: ID String[30] (31 bytes)  
 Manufacturer, string[30] (31 bytes)  
 Serial Number, string[30] (31 bytes)  
 Type, string[30] (31 bytes)  
 Last calibration date, string[10] (11 bytes)  
 Last certificate number, string[30] (31 bytes)  
 Pass/fail tolerance (%) (float)  
 Pass/fail tolerance (temperature) (float)  
 Min. temperature (Celsius) (float)  
 Max. temperature (Celsius) (float)  
 Tag No. 1, string[30] (31 bytes)  
 Tag No. 2, string[30] (31 bytes)  
 Tag No. 1 info, string[30] (31 bytes)  
 Tag No. 2 info, string[30] (31 bytes)  
 Certificate notes, string[30] (31 bytes)  
 Notes, string[240] (241 bytes)

**Scenario**

Data, PC → Calibrator: Work order number (integer)  
 DataID=3 (byte)  
 Line number (1..20), (byte), not used  
 AsFound/AsLeft, (boolean), not used

Data, PC ← Calibrator: Manual heating source used (boolean)  
 DTI used (boolean)  
 Ref. sensor (byte)  
 0: Internal calibrator sensor  
 1: External sensor connected to calibrator  
 2: External sensor connected to DTI  
 3: External sensor read by operator  
 Sensor Under Test (byte)  
 0: Calibrator sensor input (RTD, TC, mA, V)  
 1: Switch test input (reserved for future use)  
 2: DTI is connected  
 3: Manual reading by operator  
 Temperature unit (byte)

- 0: Celsius
- 1: Fahrenheit
- 2: Kelvin

### **Calibrator**

Data, PC → Calibrator: Work order number (integer)  
DataID=4 (byte)  
Line number (1..20), (byte), not used  
AsFound/AsLeft, (boolean), not used

Data, PC ← Calibrator: ID, string[30] (31 bytes)  
Manufacturer, string[30] (31 bytes)  
Serial Number, string[30] (31 bytes)  
Type, string[30] (31 bytes)  
Calibrator minimum temperature (Celsius)  
(float)  
Calibrator maximum temperature (Celsius)  
(float)  
User minimum temperature (Celsius) (float)  
User maximum temperature (Celsius) (float)  
Number of decimals (byte)  
Mains voltage, string[11] (11 bytes)  
Mains frequency, string[7] (7 bytes)  
Uncertainty (float)  
Certificate notes, string[30] (31 bytes)  
Notes, string[240] (241 bytes)

### **Manual heating source**

Data, PC → Calibrator: Work order number (integer)  
DataID=5 (byte)  
Line number (1..20), (byte), not used  
AsFound/AsLeft, (boolean), not used

Data, PC ← Calibrator: ID, string[30] (31 bytes)  
Manufacturer, string[30] (31 bytes)  
Serial Number, string[30] (31 bytes)  
Type, string[30] (31 bytes)  
Minimum temperature (Celsius) (float)  
Maximum temperature (Celsius) (float)  
Number of decimals (byte)  
Certificate notes, string[30] (31 bytes)  
Notes, string[240] (241 bytes)

DataID=6 reserved for future use.

## **DTI**

Data, PC → Calibrator: Work order number (integer)  
DataID=7 (byte)  
Line number (1..20), (byte), not used  
AsFound/AsLeft, (boolean), not used

Data, PC ← Calibrator: Serial number, string[30] (31 bytes)  
Certificate notes, string[30] (31 bytes)  
External reference sensor channel No. (0..2) (byte)  
Sensor under test channel No. (0..2) (byte)

## **External reference sensor**

Data, PC → Calibrator: Work order number (integer)  
DataID=8 (byte)  
Line number (1..20), (byte), not used  
AsFound/AsLeft, (boolean), not used

Data, PC ← Calibrator: ID, string[30] (31 bytes)  
Manufacturer, string[30] (31 bytes)  
Serial Number, string[30] (31 bytes)  
Number of decimals (byte)  
Certificate notes, string[30] (31 bytes)  
Notes, string[240] (241 bytes)

## **Input from sensor under test**

Data, PC → Calibrator: Work order number (integer)  
DataID=9 (byte)  
Line number (1..20), (byte), not used  
AsFound/AsLeft, (boolean), not used

Data, PC ← Calibrator: ID, string[30] (31 bytes)  
Manufacturer, string[30] (31 bytes)  
Serial number, string[30] (31 bytes)  
Uncertainty, string[60] (61 bytes)  
Type (byte)  
0: 0-4V  
1: 0-12V  
2: 4-20mA  
3: Pt50 M (Russian)  
4: Pt50 P (Russian)  
5: Pt100 IEC  
6: Pt100 Mill  
7: Pt100 M (Russian)

- 8: Pt100 P (Russian)
- 9: Pt500 IEC
- 10: Pt1000 IEC
- 11: TC E
- 12: TC J
- 13: TC K
- 14: TC L
- 15: TC N
- 16: TC R
- 17: TC S
- 18: TC T
- 19: TC U
- 20: TC XK (Russian)
- 21: Pt 50 P6652-94
- 22: Pt 100 P6652-94
- 23: Manual reading (only work order)
- 24: Switch test (reserved for future use)
- 25: No sensor is selected. Display prints 'None'.
- 26: Cu50 (Chinese JJG229 std.)
- 27: Cu100 (Chinese JJG229 std.)
- 28: Pt10 IEC

Number of wires (byte) (Only used by Ptxx sensor types)

Convert to temperature (boolean)

Minimum temperature (Celsius) (float)

Maximum temperature (Celsius) (float)

Minimum input temperature (Celsius) (float)

Maximum input temperature (Celsius) (float)

Number of decimals (byte)

Certificate notes, string[30] (31 bytes)

Input text, string[60] (61 bytes)

Auto CJ (Cold Junction) compensation (boolean)

Value of compensation (float)

(Type 25-28 are valid from firmware ver. 1.22)

### **Calibration procedures in general**

Data, PC → Calibrator:

Work order number (integer)

DataID=10 (byte)

Line number (1..20), (byte), not used

AsFound/AsLeft, (boolean), not used

Data, PC ← Calibrator: ID, string[30] (31 bytes)  
Number of temperature steps (1..20) (byte)  
Use Sensor Under Test stability criteria (boolean)  
Keep last measured temperature when finished (boolean)  
Cooling temperature (Celsius) (float)  
Use "Set follows True" correction (boolean)  
Use slope (boolean)  
Convert reference sensor to temperature (boolean)

### **Calibration procedure, single step**

Data, PC → Calibrator: Work order number (integer)  
DataID=11 (byte)  
Line number (1..20), (byte)  
AsFound/AsLeft, (boolean), not used

Data, PC ← Calibrator: SET temperature (Celsius) (float)  
Slope rate (degree Celsius per minute), only used by switch test input (float)  
Reference sensor stability time (minutes) (float)  
Reference sensor stability time tolerance (Celsius) (float)  
Sensor Under Test stability time (minutes) (float)  
Sensor Under Test stability time tolerance (Celsius) (float)  
Heating source (byte)  
    0: Calibrator  
    1: Manual heating source

### **Calibration in general**

Data, PC → Calibrator: Work order number (integer)  
DataID=12 (byte)  
Line number (1..20), (byte), not used  
AsFound/AsLeft, (boolean)  
    0: AsFound  
    1: AsLeft

Data, PC ← Calibrator: Ambient temperature (float)  
Calibration time, begin, minutes (byte)  
Calibration time, begin, hours (byte)

Calibration time, begin, day (byte)  
Calibration time, begin, month (byte)  
Calibration time, begin, year (integer)  
Calibration time, end, minutes (byte)  
Calibration time, end, hours (byte)  
Calibration time, end, day (byte)  
Calibration time, end, month (byte)  
Calibration time, end, year (integer)

### **Calibration, single step**

Data, PC → Calibrator: Work order number (integer)  
DataID=13 (byte)  
Line number (1..20) (byte)  
AsFound/AsLeft, (boolean)  
    0: AsFound  
    1: AsLeft

Data, PC ← Calibrator: Reference sensor temperature (Celsius)  
(float)  
Sensor Under Test (Celsius, ohm, mV, V,  
mA) (float)  
Pass/Fail flag (boolean)

## **Telegram #35 – Not applicable**

## **Telegram #36 – Not applicable**

## **Telegram #37 – Not applicable**

## **Telegram #38 – Read clock**

Number: 38  
Subject: Date & time  
Reading/writing: Reading  
Data, PC → calibrator: -  
Data, PC ← calibrator: 8 bytes:  
Seconds (Byte) (0-59)  
Minutes (Byte) (0-59)  
Hours (Byte) (0-23)  
Day of the week (Byte) (1=Monday,  
2=Tuesday, ... , 7=Sunday)



Day (Byte) (1-31)  
Month (Byte) (1-12)  
Year (Word) (1998-2099)

### **Telegram #39 – Write clock**

Number: 39  
Subject: Date & time  
Reading/writing: Writing  
Data, PC → calibrator: 8 bytes:  
Seconds (Byte) (0-59)  
Minutes (Byte) (0-59)  
Hours (Byte) (0-23)  
Day of the week (Byte) (1=Monday,  
2=Tuesday, ... , 7=Sunday)  
Day (Byte) (1-31)  
Month (Byte) (1-12)  
Year (Word) (1998-2099)  
Data, PC ← calibrator: -

**Telegram #40 – Not applicable**

**Telegram #41 – Not applicable**

**Telegram #42 – Not applicable**

**Telegram #43 – Not applicable**

**Telegram #44 – Not applicable**

**Telegram #45 – Not applicable**

**Telegram #46 – Not applicable**

**Telegram #47 – Not applicable**

## Telegram #48 – Not applicable

## Telegram #49 – Not applicable

## Telegram #50 – Read temperature scaling

Number: 50  
Subject: Temperature scaling  
Reading/writing: Reading  
Data, PC → calibrator: 1 byte:  
Input type (Byte):  
0: 0-4V  
1: 0-12V  
2: 4-20mA  
Data, PC ← calibrator: 16 bytes:  
Input scaling, min. temperature in °C (Float)  
Input scaling, max. temperature in °C (Float)  
Input scaling, min. measuring value (Float)  
Input scaling, max. measuring value (Float)

## Telegram #51 – Write temperature scaling

Number: 51  
Subject: Temperature scaling  
Reading/writing: Writing  
Data, PC → calibrator: 17 bytes:  
Input type (Byte):  
0: 0-4V  
1: 0-12V  
2: 4-20mA  
Input scaling, min. temperature in °C (Float)  
Input scaling, max. temperature in °C (Float)  
Input scaling, min. measuring value (Float)  
Input scaling, max. measuring value (Float)

## Telegram #52 – Read cold-junction compensation

Number: 52  
Subject: Cold-junction compensation

Reading/writing: Reading  
Data, PC → calibrator: -  
Data, PC ← calibrator: 9 bytes:  
Automated compensation (boolean)  
Manual compensation value in °C (Float)  
Automated compensation value in °C (Float)

### **Telegram #53 – Write cold-junction compensation**

Number: 53  
Subject: Cold-junction compensation  
Reading/writing: Writing  
Data, PC → calibrator: 5 bytes  
Automated compensation (boolean)  
Manual compensation value in °C (Float)

### **Telegram #54 – Read sensor under test parameters**

Number: 54  
Subject: Sensor under test parameters  
Reading/writing: Reading  
Channel: PC ↔ calibrator  
Data, PC → calibrator: -  
Data, PC ← calibrator: 8 bytes:  
Type (Byte):  
0: 0-4V  
1: 0-12V  
2: 4-20mA  
3: Pt50 M (Russian)  
4: Pt50 P (Russian)  
5: Pt100 IEC  
6: Pt100 mill  
7: Pt100 M (Russian)  
8: Pt100 P (Russian)  
9: Pt500 IEC  
10: Pt1000 IEC  
11: TC E  
12: TC J  
13: TC K  
14: TC L

- 15: TC N
- 16: TC R
- 17: TC S
- 18: TC T
- 19: TC U
- 20: TC XK (Russian)
- 21: Pt 50 P6652-94
- 22: Pt 100 P6652-94
- 23: (reserved for future use)
- 24: Switch test (reserved for future use)
- 25: No sensor is selected. Display prints 'None'.
- 26: Cu50 (Chinese JJG229 std.)
- 27: Cu100 (Chinese JJG229 std.)
- 28: Pt10 IEC

Convert to temperature (boolean)  
RTD type wire number (Byte), 2 – 4  
Automated CJ compensation (boolean)  
Manual CJ compensation value (Float)  
(Type 25-28 are valid from firmware ver. 1.22)

## Telegram #55 – Write sensor under test parameters

Number: 55  
Subject: Sensor under test parameters  
Reading/writing: Writing  
Channel: PC ⇔ calibrator  
Data, PC → calibrator: 8 bytes:  
Type (Byte):  
0: 0-4V  
1: 0-12V  
2: 4-20mA  
3: Pt50 M (Russian)  
4: Pt50 P (Russian)  
5: Pt100 IEC  
6: Pt100 mill  
7: Pt100 M (Russian)  
8: Pt100 P (Russian)  
9: Pt500 IEC  
10: Pt1000 IEC  
11: TC E  
12: TC J  
13: TC K  
14: TC L  
15: TC N  
16: TC R  
17: TC S  
18: TC T  
19: TC U  
20: TC XK (Russian)  
21: Pt 50 P6652-94  
22: Pt 100 P6652-94  
23: (reserved for future use)  
24: Switch test (reserved for future use)  
25: No sensor is selected. Display prints 'None'.  
26: Cu50 (Chinese JJG229 std.)  
27: Cu100 (Chinese JJG229 std.)  
28: Pt10 IEC  
Convert to temperature (boolean)

RTD type wire number (Byte), 2 - 4, reading only at sensor types 3-10 and 21-22.  
Automated CJ compensation (boolean)  
Manual CJ compensation value (Float)  
(Type 25-28 are valid from firmware ver. 1.22)

Data, PC ← calibrator: -

## Telegram #56 – Read reference sensor parameters

Number: 56  
Subject: Reference sensor parameters  
Reading/writing: Reading  
Data, PC → calibrator: -  
Data, PC ← calibrator: 3 bytes:  
Internal/external reference sensor (bool):  
    0: False, Internal  
    1: True, External, connected to calibrator  
SET follows TRUE active (boolean)  
Convert reference sensor to temperature (boolean)

## Telegram #57 – Write reference sensor parameters

Number: 57  
Subject: Reference sensor parameters  
Reading/writing: Writing  
Channel: PC ↔ calibrator  
Data, PC → calibrator: 3 bytes:  
Internal/external reference sensor (boolean):  
    0: False, Internal  
    1: True, External, connected to calibrator  
SET follows TRUE active (boolean)  
Convert reference sensor to temperature (boolean)  
Data, PC ← calibrator: -

**Telegram #58 – Not applicable**

**Telegram #59 – Not applicable**

**Telegram #60 – Not applicable**

**Telegram #61 – Not applicable**

**Telegram #62 – Not applicable**

**Telegram #63 – Not applicable**

**Telegram #64 – Not applicable**

**Telegram #65 – Not applicable**

**Telegram #66 – Read external reference sensor data**

Number: 66  
Subject: External reference sensor data  
Reading/writing: Reading  
Data, PC → calibrator: -  
Data, PC ← calibrator: Up to 205 bytes:  
Number of temperature steps, 0 or 3-10 (Byte)  
0 temperature steps only occur at factory defaults, as the resistance/ temperature table is empty  
Serial number (String[12])  
A (Double)  
B (Double)  
C (Double)  
R<sub>0</sub> (Double)  
10 × resistance in Ω (Double)  
10 × temperature in °C (Double)  
Note for the resistance and temperature parameters: The number of values are from 0 to 10, depending on the parameter 'Number of temperature steps'. The resistor and



temperature parameters are combined in sets of (r1,t1) etc.

## **Telegram #67 – Write external reference sensor data**

Number: 67  
Subject: External reference sensor data  
Reading/writing: Writing  
Data, PC → calibrator: Up to 182 bytes:  
Number of temperature steps, 3-10 (Byte)  
Serial number (String[20])  
10 × resistance in  $\Omega$  (Double)  
10 × temperature in  $^{\circ}\text{C}$  (Double)  
Note, please ref. telegram #66 for the format of the resistance and temperature parameters.  
Data, PC ← calibrator: -

**Telegram #68 – Not applicable**

**Telegram #69 – Not applicable**

**Telegram #70 – Not applicable**

**Telegram #71 – Not applicable**

**Telegram #72 – Not applicable**

**Telegram #73 – Not applicable**

**Telegram #74 – Not applicable**

**Telegram #75 – Not applicable**

**Telegram #76 – Not applicable**

**Telegram #77 – Not applicable**

**Telegram #78 – Not applicable**

**Telegram #79 – Not applicable**

**Telegram #80 – Read calibration date for SUT inputs**

Number:	80
Subject:	Calibration date for sensor under test inputs
Reading/writing:	Reading
Data, PC → calibrator:	1 byte: Input, (Byte) , 0=mA, 1=TC, 2= V, 3=Ohm, 4=Ref
Data, PC ← calibrator:	4 bytes: Day, 1..31 (Byte) Month, 1..12 (Byte) Year, 1998..2099 (Word)

## Telegram #81 – Write calibration date for SUT inputs

Number: 81  
Subject: Calibration date for sensor under test inputs  
Reading/writing: Writing  
Data, PC → calibrator: 5 bytes:  
Input, (Byte) , 0=mA, 1=TC, 2= V, 3=Ohm,  
4=Ref  
Day, 1..31 (Byte)  
Month, 1..12 (Byte)  
Year, 1998..2099 (Word)  
Data, PC ← calibrator: -

## Telegram #82 – Not applicable

## Telegram #83 – Delete work order

Number: 83  
Subject: Delete work order  
Reading/writing: Writing  
Data, PC → calibrator: 2 bytes:  
Work order number, 1-20 (Integer)  
Data, PC ← calibrator: -

## Telegram #84 – Read test mode

Number: 84  
Subject: Test mode  
Reading/writing: Reading  
Data, PC → calibrator: -  
Data, PC ← calibrator: 2 byte:  
Test mode (Byte):  
0: Normal  
1: Simulation  
2: Service  
Internal status (Byte)  
0: Temperature setup.  
1: Switch test  
2: Auto step  
3: Work order

## Telegram #85 – Not applicable

## Telegram #86 – Not applicable

## Telegram #87 – Read slope rate status

Number: 87  
Subject: Slope rate status. Other slope rate than default is active.  
Reading/writing: Reading  
Data, PC → calibrator: -  
Data, PC ← calibrator: 1 bytes:  
Slope rate status (boolean)

## Telegram #88 – Write slope rate status

Number: 88 (Reserved for future use)  
Subject: Slope rate status. Pls. also ref. telegram #20.  
Reading/writing: Writing  
Data, PC → calibrator: 1 bytes:  
Slope rate status (boolean)  
Data, PC ← calibrator: -

## Telegram #93 – Not applicable

## Telegram #94 – Not applicable

## Telegram #95 – Not applicable

## Telegram #96 – Not applicable

Visit **TRANSCAT**.com >

[sales@transcat.com](mailto:sales@transcat.com) | 800.828.1470