USER-MANUAL

MANUAL
RS232 communication
TamCom®

Tamson Instruments bv

P.M. Tamson Instruments bv
van 't Hoffstraat 12
2665 JL Bleiswijk
THE NETHERLANDS

T. 31 (0) 10 522 43 73
F. 31 (0) 10 521 19 41
www.tamson.com
info@tamson.com

VAT: NL 80 66 34 984 B01
Bank account no.: Rabobank 16 01 00 046
Chamber of commerce 27 16 95 41
IBAN Code: NL95 RABO 0160100046

1 SAFETY AND WARNINGS

Make sure before installing or operating the equipment to read and understand all instructions and safety precautions listed in this manual. If there are any questions concerning the operation of the equipment or about the information given in this manual please contact your local dealer or our sales department first.

Performance of installation, operation, or maintenance other than those described in this manual may result in a hazardous situation and may void the manufacturer's warranty.

Never operate equipment that is not correctly installed. Unqualified personnel must not operate the equipment. Avoid damage to the equipment, or its accessories, caused by incorrect operation.

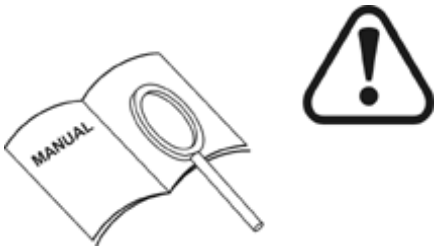
Important:
- When performing service, maintenance or moving the apparatus, always disconnect the apparatus at the main's socket,
- Proper skilled and trained personnel are only allowed to operate this equipment,
- Take notice of warning labels and never remove them,
- Refer service and repairs to qualified technician,
- If a problem persists, call your supplier or Tamson Instruments b.v.

2 WARRANTY

Tamson Instruments b.v. warrants that all their manufactured equipment is free from defects in material and workmanship, preventing the machine from normal operation. Tamson Instruments b.v does not warranty that the equipment is fit for any other use than stated in this manual. The manufacturer can only be held responsible for the security, reliability and performance of the equipment, when operated in accordance with the operating instructions, extensions, adjustments, changes and/or if repair is performed by Tamson Instruments b.v. or authorized persons only. This warranty is limited to one year from the date of invoicing. All equipment and materials are subject to standard production tolerances and variations.

3 PRECAUTIONS AND HAZARDS

Before attempting to operate the bath read all parts of this manual carefully to insure smooth operation and avoid damage to the equipment or its accessories.



READ CAREFULLY

If problem persists, call your supplier or Tamson Instruments bv. Never operate the equipment if not correctly installed. The equipment must be operated only by qualified personnel. Avoid damage to the equipment or its accessories through incorrect operation.

4 SERIAL COMMUNICATION

Equipment is standard equipped with RS232 communication. In some occasions the RS232 interface is optional. Using RS232 the controller can be controlled remotely using the Tamson software or a serial terminal or your own software.

Tamson apparatus can be connected to RS232 using a Modbus protocol or standard ASCII as interfacing language.

The Tamcom software can do following:

- Logging data into a file, (CSV)
- Programming a Set Point curve via simple data in a file
- Display process value and set point temperature in a graph
- Actual values
- Change set point temperature
- Show Process value
- Set Offset
- Set PID values

4.1 protocol

The Tamson equipment uses two types of controllers. The oldes controller uses the Modbus protocol. The most recent controller uses a standard ASCII interface.

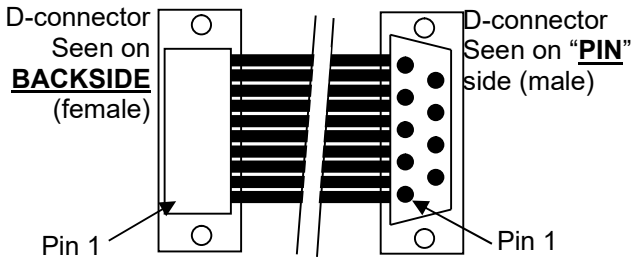
The Modbus is a more complex protocol however offers error check. It uses a device and parameter address and multiple apparatus can be operated on single COM-port

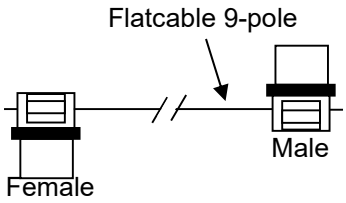
- Device addresses can hook up to 255 apparatus communicating on one single COM-port
- Parameter addresses supplying read or write up to 65536 parameters per device/apparatus

Please note that MODBUS counts addresses officially from 0 .. 65535. Some programs, like the mentioned Modpoll.exe later in this chapter, or interfaces start counting from 1. This means specific addresses move up with 1.

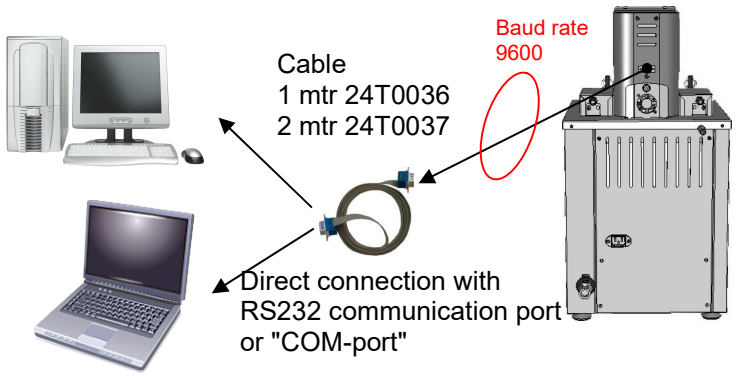
4.2 Connecting apparatus

Between computer and apparatus use following cable. Cable length up to 4 meters has been tested. Longer cables are possible but RFI intereference may lead to communication problems.

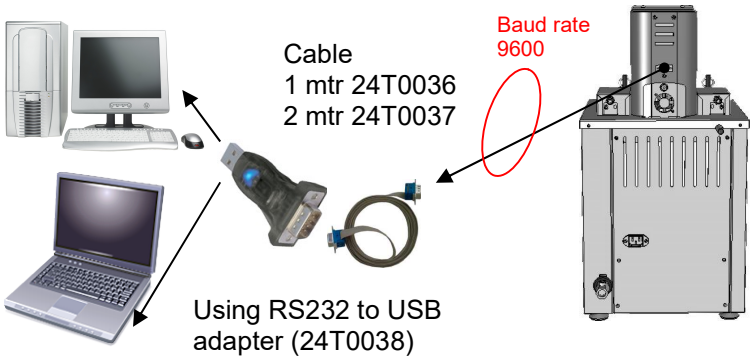




4.2.1 RS232(apparatus) to COM(computer)



4.2.2 RS232(apparatus) to USB adapter(computer)



5 TESTING MODBUS COMMUNICATION

In order to test PC, cable and apparatus the small program called MODPOLL can be used to verify. MODPOLL.EXE is a freeware program to be found at our website www.tamson.com in the download section or at www.modbusdriver.com

MODPOLL.EXE must be started from a DOS box and is command line operated. The DOS box can be found in Windows under “Start”, “Programs”, “Accessories” or ran from C:\windows32\cmd.exe. It also is possible under “Start”, Run and enter CMD and press “OK”

Following starts the communication in a DOS box
in DOS command center:
Example (case incentive)
MODPOLL -m rtu -a 1 -r 402 -c 10 com1

-m rtu
Modbus RTU protocol (default)

-a #
Slave address (1-255 for RTU/ASCII, 1 is default)

-r# start reference (1-65536, 100 is default)

-c # number of values to poll. I.e. a value of 10 will display first address and following 9 values of succeeding addresses.

-com1
Serial port COM1, COM2 etc

Please note MODPOLL.EXE starts counting addresses from "1" instead of the official "0". This means the actual parameter address is one higher. I.e. to read process value(601) use **602** for the address:

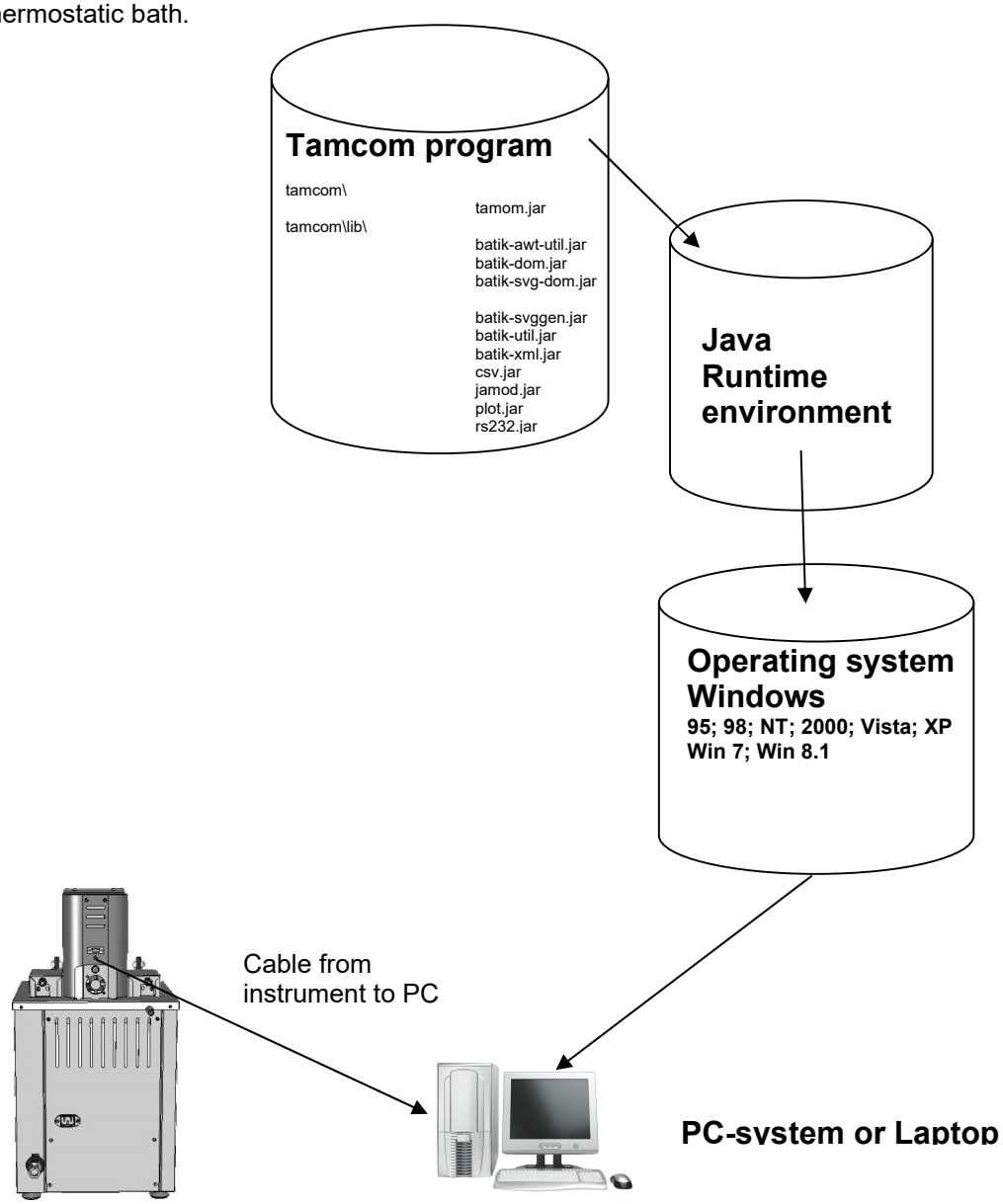
modpoll -m rtu -a 1 -r 602 -c 1 com1

6 USING THE COMPUTER FOR COMMUNICATION (USB OR RS232)

The TamCom® program needs a Java environment to run in. Both program file as the Java environment are supplied on disk in a full directory. This directory has to be copied into a specific folder. The program is than started by selecting the "tamcom.jar" file. This specific file loads the Java environment and starts the TamCom program.

Tamcom does not install any files in a registry or whatsoever. Simply removing the files from disk removes the complete program from your system. The program code is written in Java and can be used to run under all versions of Windows.

The serial interface and the program TamCom can be used to generate a set point curve or log data from the thermostatic bath.



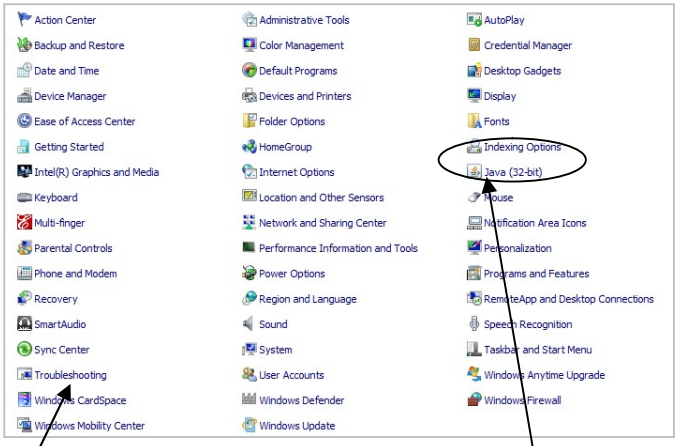
Check the Java-runtime environment

The Tamcom program is written in Java. This means the program can be used on any Windows platform as long as the Java Runtime environment is installed.

There are two ways to check wether the runtime environment is installed:
1) From the configuration menu
2) From the Java website

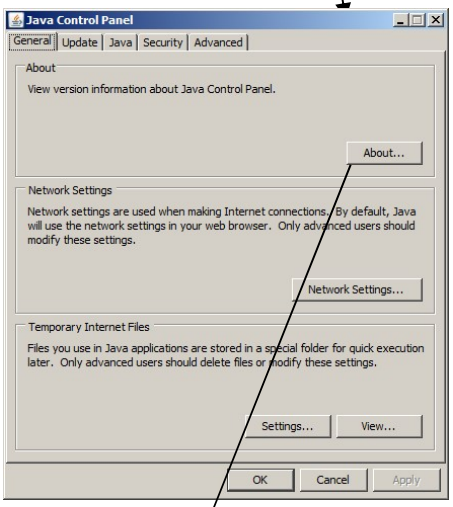
Check the Java RUN-time, the confiruration menu(1)

Java NOT installed



Java 32 bits installed

Java installed
double click



Tamcom uses the 32-bits library for USB and Comport. Java 64 bits gives problems with these libraries at this moment. **Therefore it is advised to use Java 32 bits**.

Check the Java RUN-time, the java website (2)
Another option is to visit the website at www.java.com. Look for the link "verify Installation" or “DO I have Java”. If the runtime is already installed the site will tell you what version you have.

Check if Java is installed or your version

Press the button

A message follows: Detecting Java on your computer.

If Java's not present install, or use the button "free Java Download" on www.java.com
So if there's no Java or you get a message that your version is too old, proceed

Accept the license agreement
Java will download a file jxpinstall.exe
if the file doesn't execute, double click the file
Install and follow the instructions.

Remember the Java Runtime is "free-ware" so no costs!

6.1 COM-port or USB

Nowadays on a modern PC the 9-pole SUB-D COM port has been replaced by USB
To connect hardware onto the USB you need a either a PCI-card with COM port or a so called USB-converter.

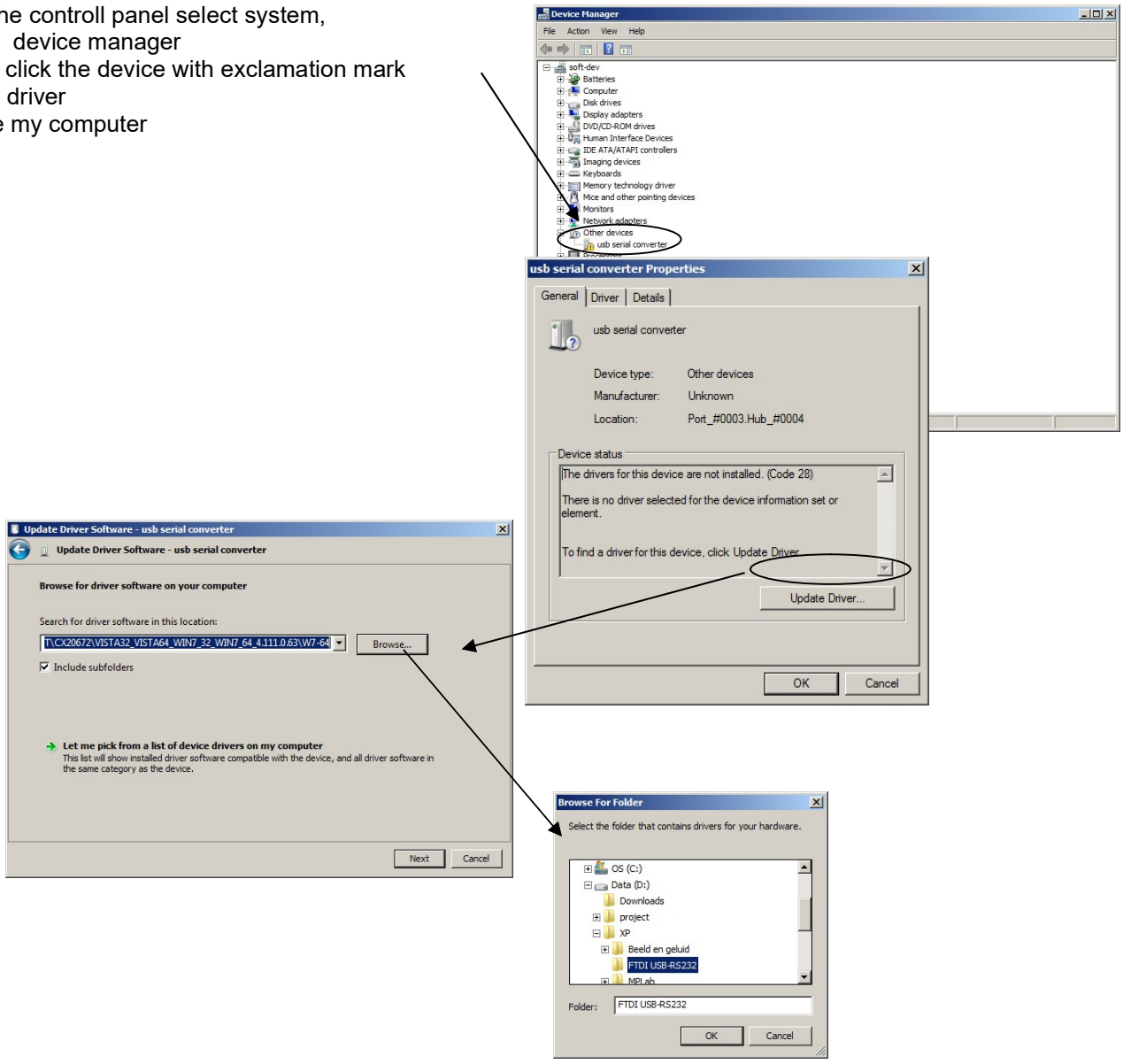
9-pole SUB-d RS232  to USB

Converter

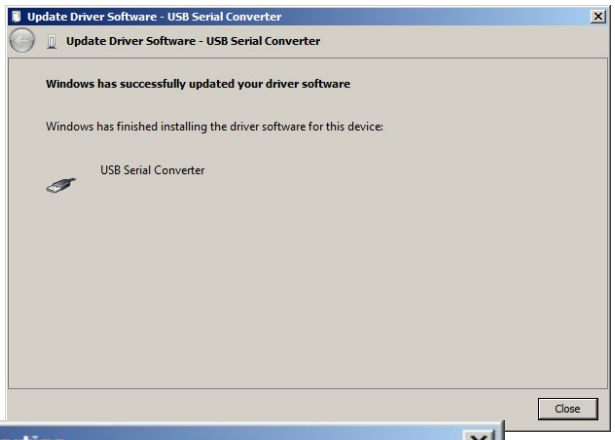
When the converter is plugged into the PC it can ask for a disk. If so, install the disk and follow the install procedure.

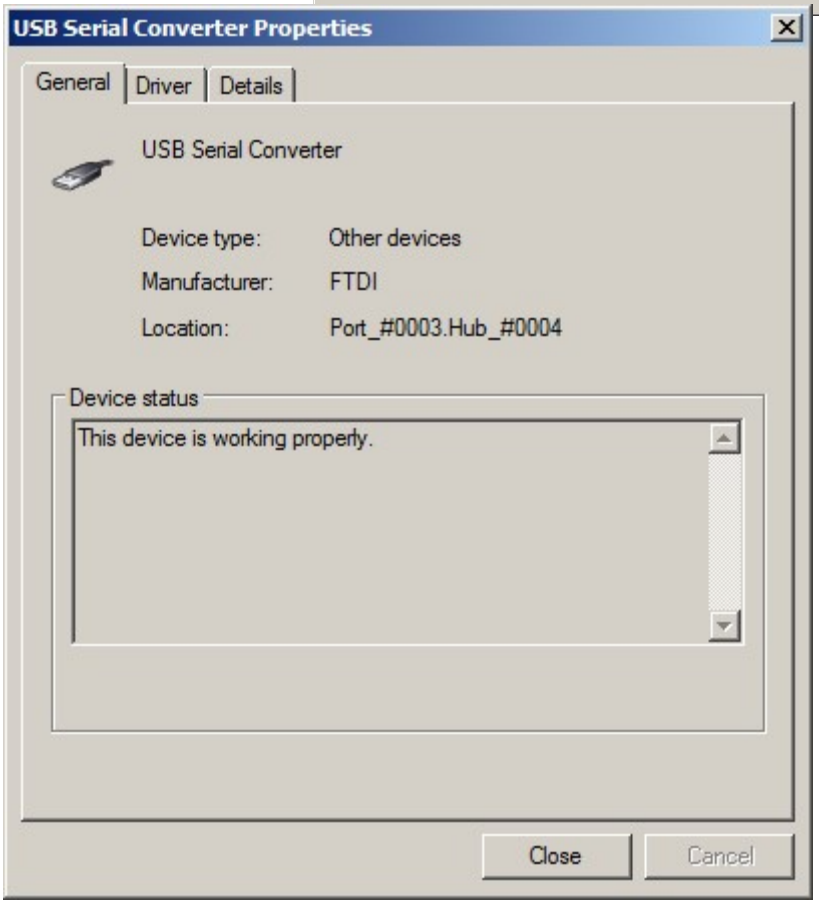
Sometimes already a driver is installed for a COM to USB converter. A lot of equipment use these converters without the user knowing it. It need specific installation.

From the controll panel select system,
device manager
double click the device with exclamation mark
update driver
browse my computer

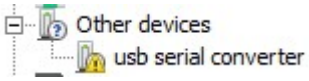



Windows Security
Would you like to install this device software?
Name: CDM Driver Package
Publisher: Future Technology Devices International ...
Always trust software from "Future Technology Devices International ...".
Install
Don't Install
You should only install driver software from publishers you trust. How can I decide which device software is safe to install?

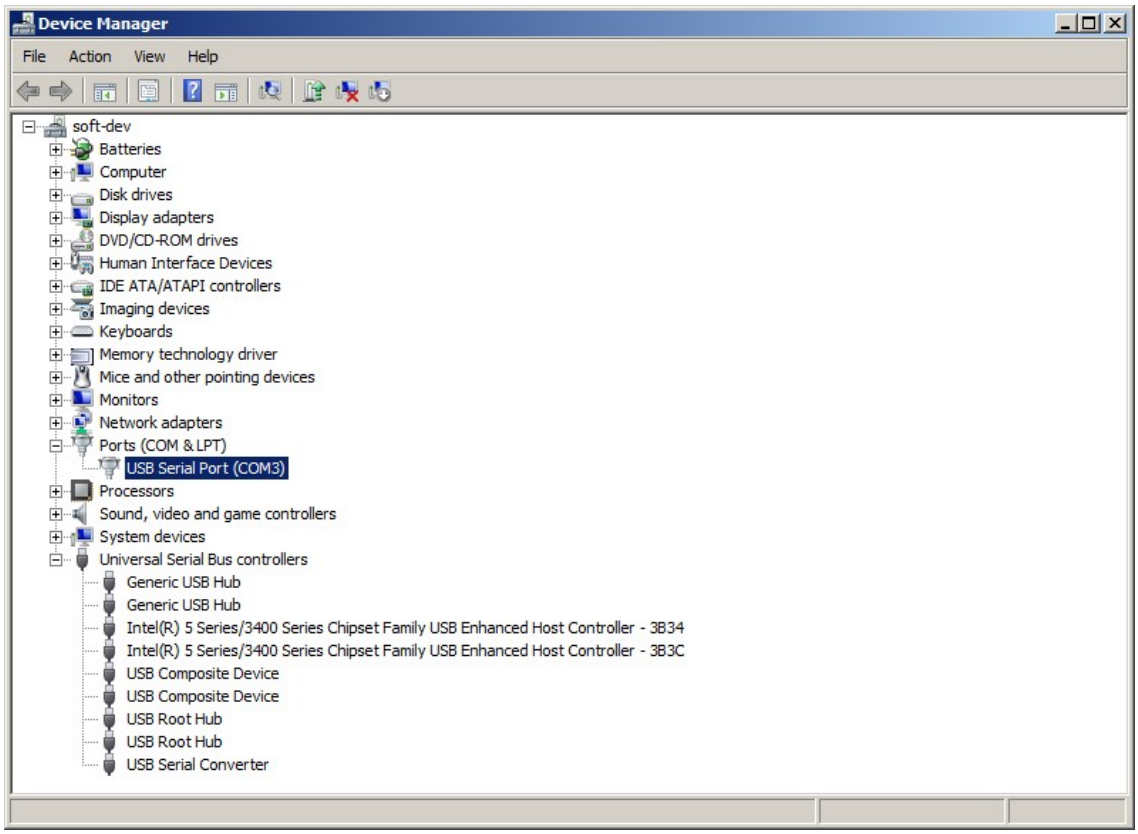

Update Driver Software - USB Serial Converter
Update Driver Software - USB Serial Converter
Windows has successfully updated your driver software
Windows has finished installing the driver software for this device:
USB Serial Converter
Close


USB Serial Converter Properties
General
Driver
Details
USB Serial Converter
Device type: Other devices
Manufacturer: FTDI
Location: Port_#0003.Hub_#0004
Device status
This device is working properly.
Close
Cancel

The exclamationmark must disappear in the "device manager window".



If not, install the device driver for a second time (this is normal). After the second time the exclamation mark must have disappeared.



6.2 Run TamCom®

Tamcom comes in a zip file with the following structure

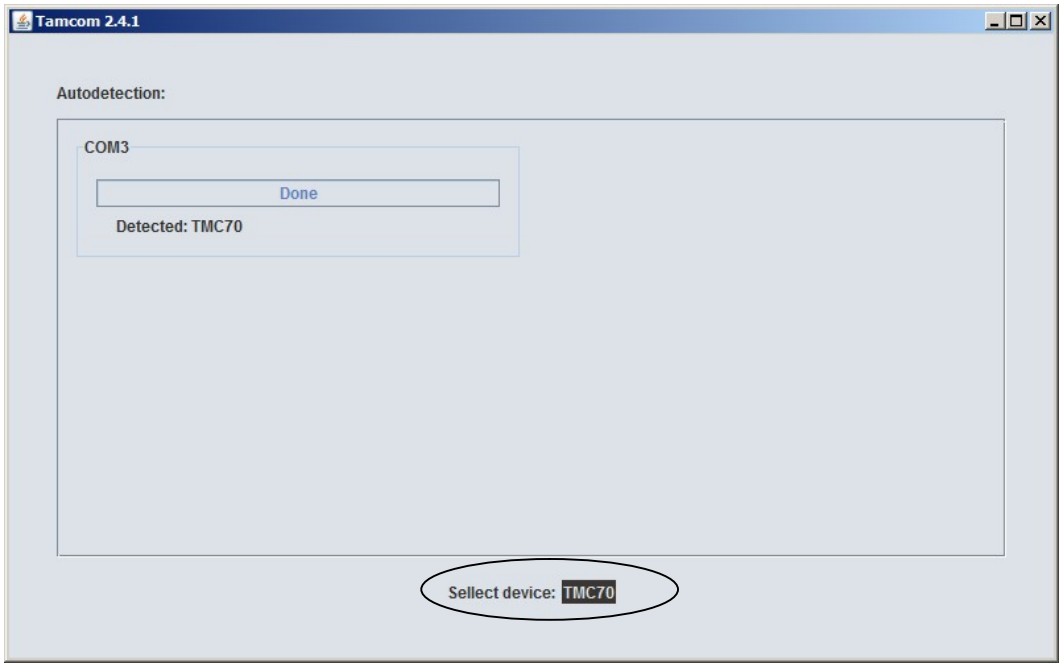
```
tamcom.jar
lib\
          batik-awt-util.jar
          batik-dom.jar
          batik-svg-dom.jar
          batik-svggen.jar
          batik-util.jar
          batik-xml.jar
          csv.jar
          jamod.jar
          plot.jar
          rs232.jar
```

On the windows PC make a Map i.e. C:\Program Files\Tamcom
Copy the tamcom.jar in this map

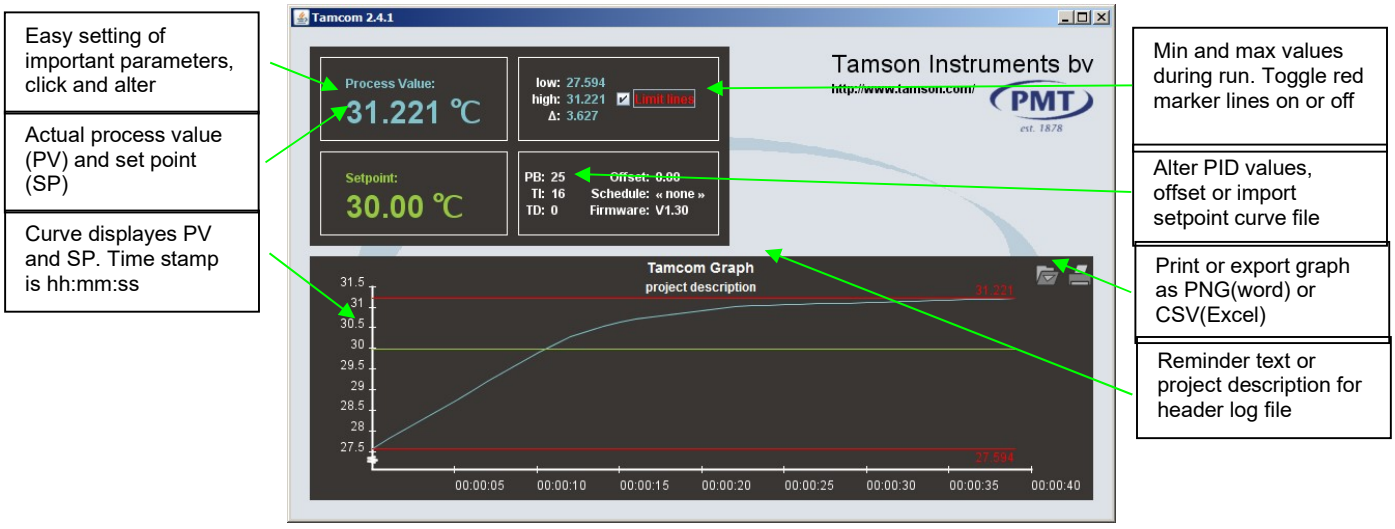
Make a map which is called lib
Copy the files " batik-awt-util.jar" .. " rs232.jar" to this sub map

doubbleclick tamcom.jar to run tamcom
In order to make a quick start on your desktop, rightclick the tamcom.jar file and choose "send to" and "Desktop".

Note:
In order to execute a "JAR" file, the java runtime needs to be installed
We have chosen for this type of installation, instead of using a wizard, to support ALL windows platforms thus also giving maximum backward compatibility also for old PC-systems.

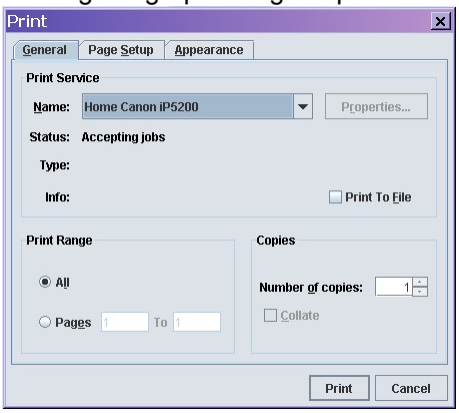


The serial ports are automatically found. In this example COM3. When using an RS232 to USB converter TamCom will find and display this device without problem. Now press scan and Tamcom starts querying the serial or USB ports to look for any connected Tamson products. If the appropriate device (displayed as TMC70 or K1S) is found select it and Tamcom will start to build a temperature graph. The screen after a few minutes of logging is shown below:

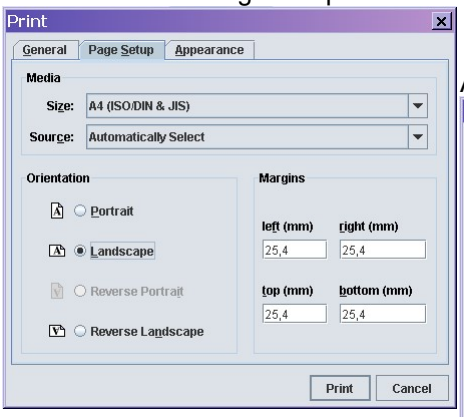


6.3 Printing

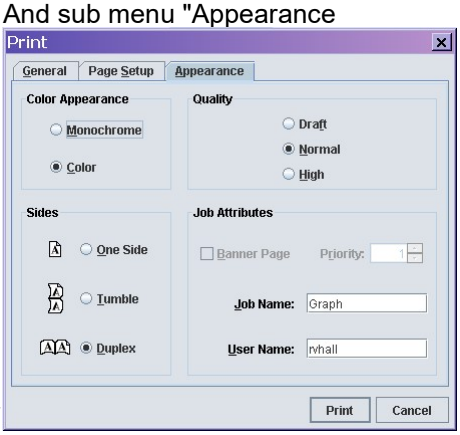
Printing the graph using the print button(bottom left):



And submenu "Page setup"



And sub menu "Appearance



Data export

Apart from printing the graph, data can be exported as vector image (SVG) or a compressed format and accepted by Microsoft Word.

The CSV(comma separated value) can be used to export data to a spread sheet.

```
Time,Setpoint,Proces Value
00:00,10,"23,863"
00:00:01,10,"23,878"
00:00:02,10,"23,889"
00:00:03,10,"23,892"
etc.
```

6.4 Zoom function:

Use your left mouse button to select a specific part of the graph. Keep the button pressed and move button to the left or right. When released this part of the graph is zoomed in. Click on the magnifying glass to return to the full graph.

When one saves the zoomed area to CSV, the actual data of this selected part is exported.

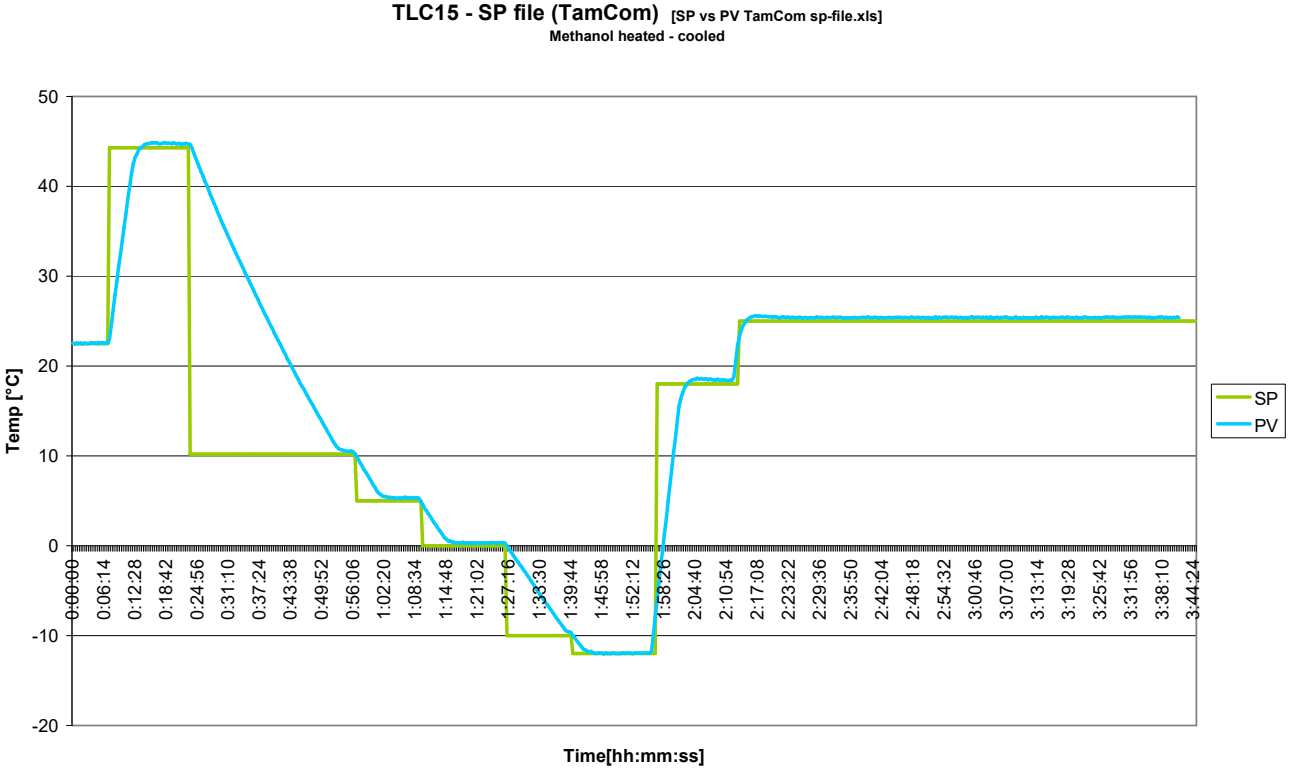
Tamcom can use following format to import set point temperature values versus time:

00:00:10,28
00:02:30,29
00:10:50,30
00:20:20,40
01:00:30,50

Save this data as CSV (Comma Separated Value). It will do following assuming reading on the controller is in degree Celsius:

After activating the file within 10 seconds the set point temperature will be set to 28 °C
After two minutes and 30 seconds SP will be set to 29°C
After ten minutes and 50 seconds SP will be set to 30°C
After twenty minutes and 20 seconds SP will be set to 40°C
After one hour and zero minutes and 30 seconds SP will be set to 50°C
Temperature will stay at this value.

Results retrieved when using a similar set point file and TLC15 cooler-circulator can be following:



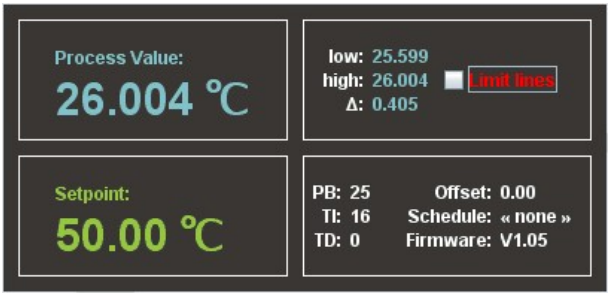
NB Depending on your windows setting the decimal point can be written as dot (".") or as shown in the example above as a comma (","). The setting can be altered in your configuration menu. (Start, Settings, Control pane Regional and Language Options, Tab "Regional Options", Tab "Numbers". However if you alter the setting in the configuration menu, it will be applied to all windows programs like i.e. Excel.

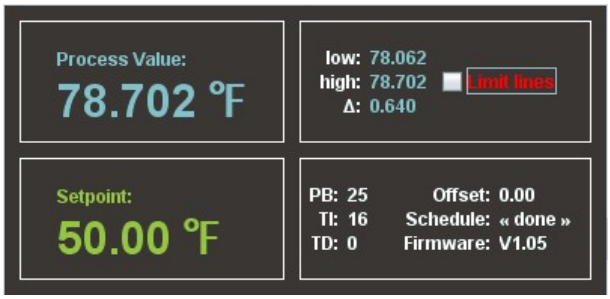
After 24 hrs you can restart with ie. 00:10 30,10 or,
00:00,5.00
05:00,3.20
10:00,6.40
15:00,7.50
20:00,4.20
00:00,2.10
05:00,5.40
10:00,7.30

6.5 Readout

Tamcom detects your controller setting at start up. If the readout on the display is in degree F, Tamcom will display degree F. If the readout is in degree C Tamcom will display degree C. Devices detected as TMC70 have the possibility to set the reading on the display of the Tamson bath. Be careful, Tamcom only detects the readout during its start up. If the readout is changed during operation the displayed value in Tamcom is not altered!

Reading in both degree C and degree F:





6.6 Hardware limits Eeprom

When setting a value, i.e. a new set point value, these values are stored in an Eeprom.

All operative and configuration parameters (like the control set point) are stored in EEPROM and guaranteed over one million erase/write cycles.

7 OPERATING THE RS232 DIRECTLY

As discussed earlier there are two protocolls:
Standard RS232
Sending plain characters
Modbus protocol
Using industry Modbus-standard as check for safe parameter read/write

7.1 RS232 Commands

General settings:
Baudrate can be selected from apparatus menu
4800

Data setting is:
8 data bits
1 stop bit
parity none
Commands overview. Notation
commands are place between brackets "[ ]"
values are placed between "< >" signs
(Do not use these brackets when sending commands.)

[ST] returns wether bath is stable or not. State 1 = stable, 0 not stable. Limits are set with parameters StableLimitHigh and StableLimitLow defined by ASTM D445. So when PV is in the region of 100°C>PV<0°C StableLimitHigh is defined as 0.05. When the average PV deviation (calculated from a sliding window of 255 seconds) is smaller than StableLimitHigh, the ST command will return stable(1).

When PV is 0°<=PV<=100°C and the average PV (calculated from a sliding window of 255 seconds) is smaller than StableLimitLow, the ST command will return stable(1)

[RS] Restarts (same as power off-on)
[RA] Reads raw ADC value, returns 3 bytes binary data, LSB first. This returns the last raw ADC reading. Note that this is updated every 200mS, so reading at a faster rate may return duplicate values. Maximum reading is 0x7FFFFF. This command is used for factory calibration/test.
[CT] **<enter>** Returns the offset value.
[CT] **<value> <enter>** Sets the offset value.

[TC] Sets/displays time constant in seconds
[PL] Power limit in seconds.
[BH] Returns boost heater on/off state, 0 or 1
[BH<n>] sets boost heater state 0/1
[CS] Control state – returns state of control algorithm
bits 4..7 represent the main control stages, bits 0..3 are sub-states within each main state
$00 : control algorithm disabled – allows external PC to take control of heaters
$10 : startup hold-off started
$11 : startup holdoff in progress
$12 : startup delay finished, decide to boost or cool
$20 : boosting
$30 : Waiting for cool-down
$40 : Start of ratio process
$41 : Ratio process in progress
$42 : waiting for PV<SP-tuneoffset after ratio

$50 : Start of successive approximation process
$51 : Sucessive approximation stabilisation delay
$52 : Successive approximation in progress
$53 : end of sucessuve approximation stage
$60 : Proportional mode (subject to change - other values to be defined)

[CS<n>] Sets state of control algorithm

[HP] Returns current heater power percentage
HP<nn> : sets heater percentage to <nn> - this command is only useful if control algorithm suspended with CS command.
[AC] Returns 1 when a new temperature value is available, 0 if no new has been measured since the last PV or PVH command. Cleared by PV or PVH command. Used for external PC control synchronisation.
PV Returns Proces Value with unit, no decimals. 23C or 74F
[PVF] returns temp as follows :
Use PVF to return a decimal value with 7 decimal digits, e.g.
12.345342C, works for degrees Centigrade, and degrees Fahrenheid. Only first three are significant. When reading is in degree Fahrenheit character F indicates the unit i.e. 73.3961328F

When temperature is negative, minus sign appears i.e. -1.6955468F

[PP] returns proportinal band value. Proportional band is 100/gain
[PI] returns integrating value
[PD] returns derivative value

PID range and set:

Pb = setting 1..999
I = 0..999
D = 0..999

[PP] Followed by value, or space and value, will set proportional band. I.E. PP25 wil set Proportional band to 25.
[PI] Followed by value, or space and value, will set integrating value. I.E. PI16 will set Integratin g value to 16.
[PD] Followed by value , or space and value, will set integrating value. I.E. PD2 will set Derivative value to 2.
The TMC70 contains 4 independent PID sets. Each set can be enabled using the TMC70.EXE program and can have its own name i.e “water”, “silicone oil” or “methanol”. However when the command PP, PI, or PD is used to set a value, automatically set 4 is used. **So PID set4 is reserved for RS232 operation and is activated when approached from RS232.**

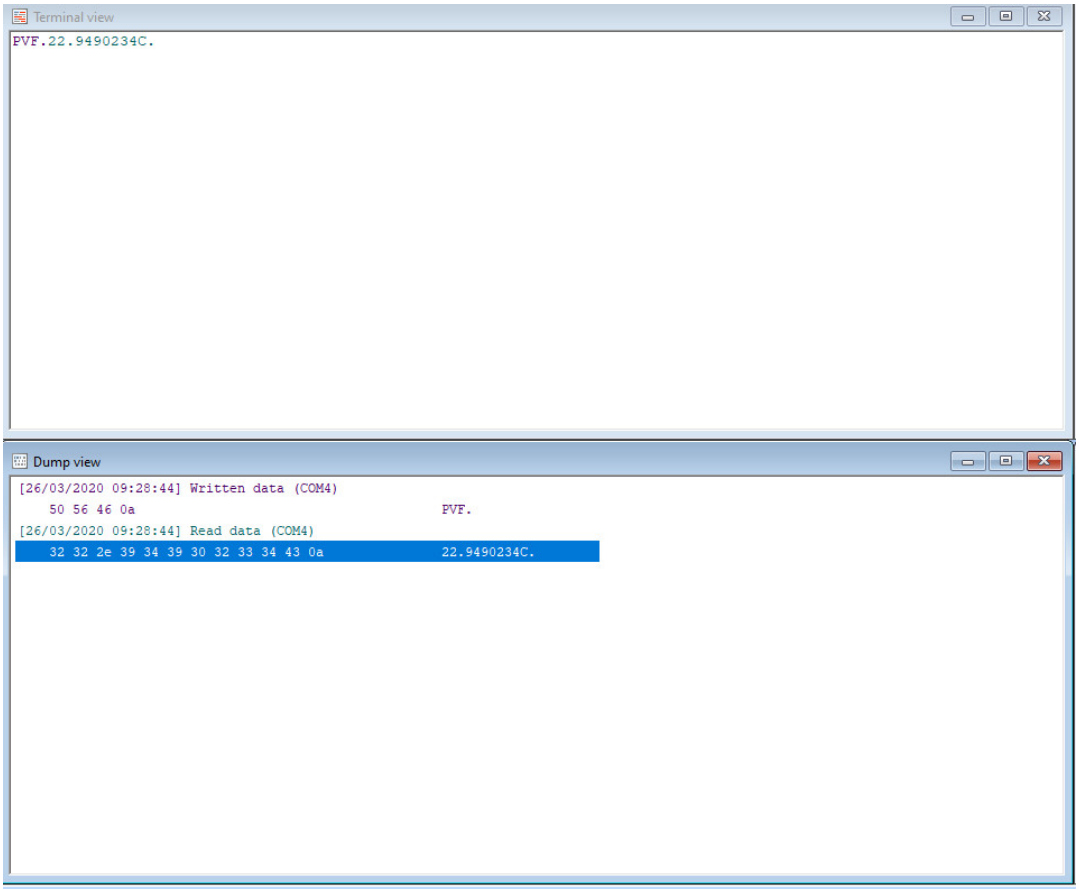
[SP] Set point temperature
SP, Set point temperature is returned i.e. 20.22
SP22.50 sets set point temperature to 20.50°C
When controller displays temperature in °F, SP20.50 sets the set point to 20.50°F
SP22.555 results in error code "10"

8 Some practice Monitoring the RS232

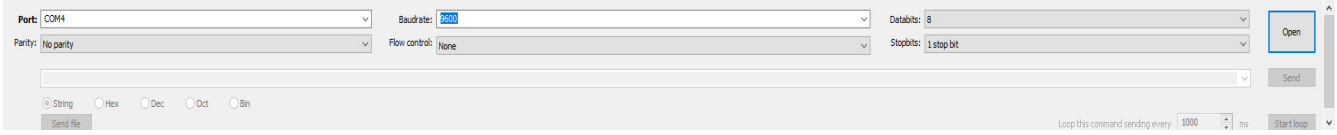
When using editors or self - programming to communicate with COM ports, linefeed and enter commands sometimes form a problem.

Some tools use plain ASCII to send, others automatically complete with a Line feed (<LF>, &h0A) or Enter (&H0D) and some do both. The end with a <LF> and <Enter>

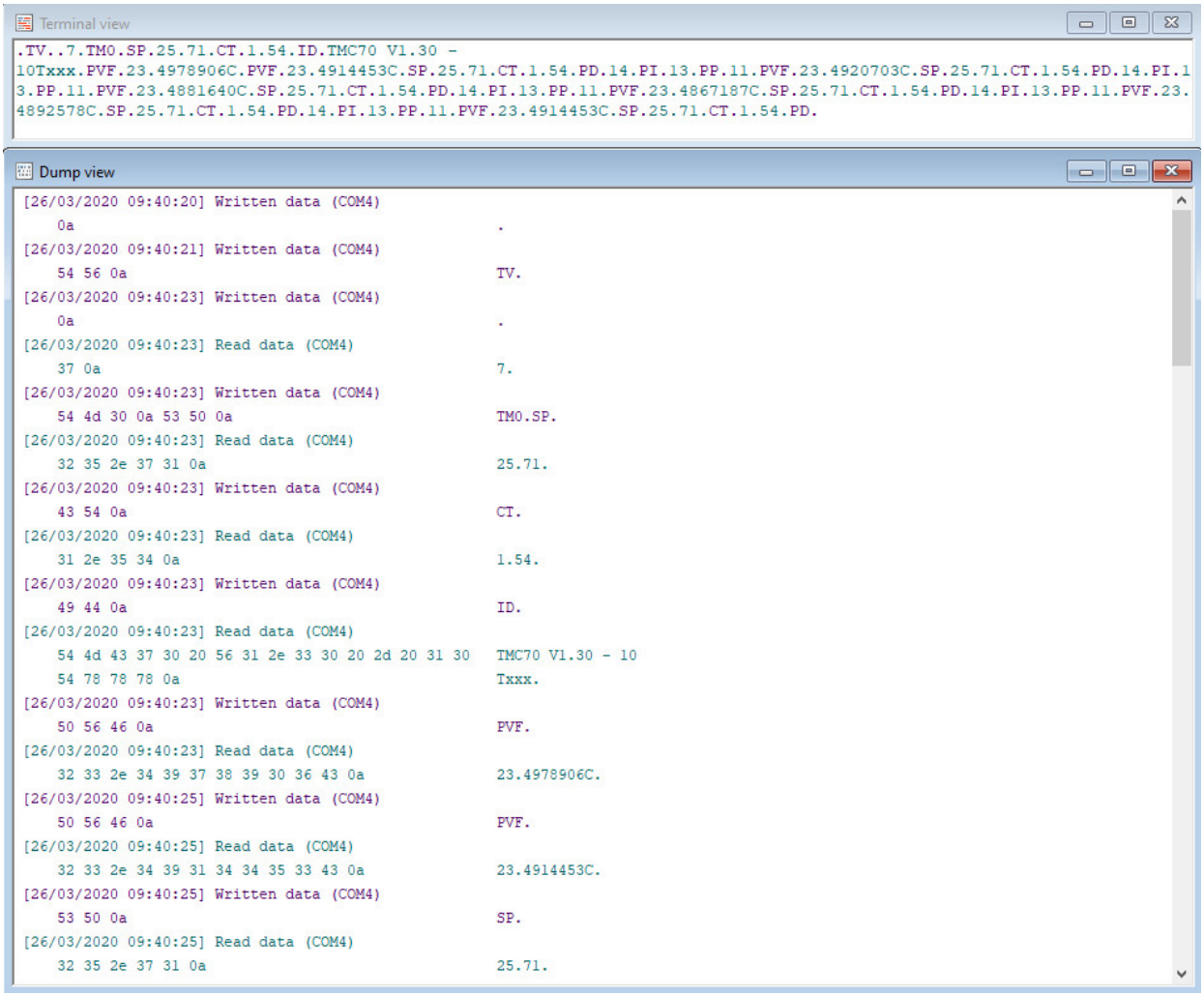
The Tamson controller needs a <LF> to process the sent command:



"Serial Port Monitor" is used to communicate and the window below explains all settings I think. In this case a RS232 to USB FTDI converter is used on port 4



Using this program (you have to pay for it), you can also listen to the port. So if the TAMCON program is connected to the Bath, you can see all communication and “Crack all commands”. To to so press "Open" button and change it to "Close". Port "COM4" wil no longer be kept occupied by "Serial Monitor" and you can start Tamcon. The log however will be visible in the program "Serial Monitor":


Terminal view
.TV..7.TM0.SP.25.71.CT.1.54.ID.TMC70 V1.30 -
10Txxx.PVF.23.4978906C.PVF.23.4914453C.SP.25.71.CT.1.54.PD.14.PI.13.PP.11.PVF.23.4920703C.SP.25.71.CT.1.54.PD.14.PI.13.PP.11.PVF.23.4881640C.SP.25.71.CT.1.54.PD.14.PI.13.PP.11.PVF.23.4867187C.SP.25.71.CT.1.54.PD.14.PI.13.PP.11.PVF.23.4892578C.SP.25.71.CT.1.54.PD.14.PI.13.PP.11.PVF.23.4914453C.SP.25.71.CT.1.54.PD.
Dump view
[26/03/2020 09:40:20] Written data (COM4)
0a .
[26/03/2020 09:40:21] Written data (COM4)
54 56 0a TV.
[26/03/2020 09:40:23] Written data (COM4)
0a .
[26/03/2020 09:40:23] Read data (COM4)
37 0a 7.
[26/03/2020 09:40:23] Written data (COM4)
54 4d 30 0a 53 50 0a TM0.SP.
[26/03/2020 09:40:23] Read data (COM4)
32 35 2e 37 31 0a 25.71.
[26/03/2020 09:40:23] Written data (COM4)
43 54 0a CT.
[26/03/2020 09:40:23] Read data (COM4)
31 2e 35 34 0a 1.54.
[26/03/2020 09:40:23] Written data (COM4)
49 44 0a ID.
[26/03/2020 09:40:23] Read data (COM4)
54 4d 43 37 30 20 56 31 2e 33 30 20 2d 20 31 30 TMC70 V1.30 - 10
54 78 78 78 0a Txxx.
[26/03/2020 09:40:23] Written data (COM4)
50 56 46 0a PVF.
[26/03/2020 09:40:23] Read data (COM4)
32 33 2e 34 39 37 38 39 30 36 43 0a 23.4978906C.
[26/03/2020 09:40:25] Written data (COM4)
50 56 46 0a PVF.
[26/03/2020 09:40:25] Read data (COM4)
32 33 2e 34 39 31 34 34 35 33 43 0a 23.4914453C.
[26/03/2020 09:40:25] Written data (COM4)
53 50 0a SP.
[26/03/2020 09:40:25] Read data (COM4)
32 35 2e 37 31 0a 25.71.


Serial Port Monitor 6
Version 6.0
Build 6.0.215
Registered to:
SPM6-SPRO00938
Registration code:
XXXXXX-H7AH7M-BVCV44-R6H2ZM-0EJPZ9-XXXXXX
License:
Professional edition - Single license
ELTIMA SOFTWARE
HOMEPAGE
ORDER FULL VERSION
SUPPORT E-MAIL: support@eltima.com
+49 696 773 3496
+1 360 312 7638
+44 131 208 3240

9 Modbus

The protocol uses the RTU (Remote terminal unit) mode of transmission.
RTU is a binary method with byte format composed as follows:

Serial data for device in CPI mode
(The CPI mode can be forced by keyboard or connecting CPI)
Address = 1
Baud rate = 9600
Byte format = 1 start bit, 8 data bit without parity,1 stop bit

9.1.1 Communication procedure

Only the master unit can initiate the communication; the slave units can transmit only after a query has been received from the master.
The general format for the transmission from master to slave is the following:

RANGE	BYTE
Slave address	1
Function code	1
Data	n
Error check (CRC-16) (low byte)	1
Error check (CRC-16) (high byte)	1

The slave detects the start of a query frame when the delay time between two characters is greater than 3.5 T.U. (Time Unit = Time necessary to transmit one character).

9.1.2 Error check (CRC-16 Cyclical Redundancy Check)

The CRC-16 value is calculated by the transmitting device. This value is appended to the message. The receiving device recalculates a CRC-16 and compares the calculated value to the received value. The two values must be equal.
The CRC-16 is started by first pre-loading a 16-bit register to all 1's. Then a process begins of applying successive the bytes of the message to the current contents of the register.
Only the eight bits of data in each character are used for generating the CRC-16. Start and stop bits, and the parity bit if one is used, do not apply to the CRC-16.
During generation of the CRC-16, each byte is exclusive OR-ed with the register contents. Then the result is shifted to the right, with a zero filled into the most significant bit (MSB) position. If the LSB was a 1, the register is then exclusive OR-ed with a preset, fixed value. If the LSB was a 0, no exclusive OR takes place.
This process is repeated until eight shifts have been performed. After the last shift, the next byte is exclusive OR-ed with the register's current value, and the process repeats for eight more shifts as described above. The final contents of the register, after all the characters of the message have been applied, are the CRC-16 value.

A procedure for generating a CRC-16 is:

1) Load a 16-bit register (CRC-16 register) with FFFFh (all 1's).

2) Exclusive OR the first byte of the message with the low byte of the CRC-16

register. Put the result in the CRC-16 register.

3) Shift the CRC-16 register one bit to the right (toward the LSB), zero-filling the MSB. Extract and examine the LSB.

4) (If the LSB was 0): Repeat Step 3 (another shift).
(If the LSB was 1): Exclusive OR the CRC-16 register with the polynomial value A001h (1010 0000 0000 0001b).

5) Repeat Steps 3 and 4 until 8 shifts have been performed. When this is done, a complete byte will have been processed.

6) Repeat Steps 2 through 5 for the next byte of the message.
Continue doing this until all bytes have been processed.

7) The final contents of the CRC-16 register is the CRC-16 value.

When the CRC-16 (16 bytes) is transmitted in the message, the low byte will be transmitted first, followed by the high byte.

9.1.3 An example of a C language function performing CRC generation:

```
/* ------------------------------------------------------------------
crc_16         calculate the crc_16 error check field

Input parameters:
  buffer: string to calculate CRC
  length: bytes number of the string

This function returns the CRC value.
------------------------------------------------------------------ */
unsigned int crc_16 (unsigned char *buffer, unsigned int length)
{
    unsigned int i, j, temp_bit, temp_int, crc;
    crc = 0xFFFF;
    for ( i = 0; i < length; i++ ) {
        temp_int = (unsigned char) *buffer++;
        crc ^= temp_int;
        for ( j = 0; j < 8; j++ ) {
            temp_bit = crc & 0x0001;
            crc >>= 1;
            if ( temp_bit != 0 )
                crc ^= 0xA001;
        }
    }
    return (crc);
}
```

9.1.4 An example of a Basic language function performing CRC generation:

```
Function CRC(message$) as long
'' CRC runs cyclic Redundancy Check Algorithm on input message$
'' Returns value of 16 bit CRC after completion and
'' always adds 2 crc bytes to message
'' returns 0 if incoming message has correct CRC
```

```
'' Must use double word for CRC and decimal constants
crc16& = 65535
FOR c% = 1 to LEN(message$)
crc16& = crc16& XOR ASC(MID$(message$, c%, 1))
FOR bit% = 1 to 8
IF crc16& MOD 2 THEN
crc16& = (crc16& \ 2) XOR 40961
ELSE
crc16& = crc16& \ 2
END IF
NEXT BIT%
NEXT c%
crch% = CRC16& \ 256: crcl% = CRC16& MOD 256
message$ = message$ + CHR$(crcl%) + CHR$(crch%)
CRC = CRC16&
END FUNCTION CRC
```

Note

The numerical values present in this text are expressed as:
Binary values if they are followed by b
Decimal values if they are not followed by any letter
Hexadecimal values if they are followed by h

9.1.5 Function codes

Codes 3 and 4: Words reading

These function codes are used by the master unit to read a consecutive group of words (16 bit) which contain the value of the variable of the slave unit.
The master can require a maximum of 10 words at a time.

Request from master to slave	
Range	Byte
Slave address (1-255)	1
Function code (03-04)	1
Word starting address (high byte)	1
Word starting address (low byte)	1
Number of word (high byte)	1
Number of word (low byte)	1
Error check (CRC-16) (low byte)	1
Error check (CRC-16) (high byte)	1

Reply from slave to master	
Range	Byte
Slave address (1-255)	1
Function code (03-04)	1
Byte count (n)	1
Data	n
Error check (CRC-16) (low byte)	1
Error check (CRC-16) (high byte)	1

The "Data" field contains the requested words in the following format: high bytes of the first word, low byte of the first word, high byte of the second word, and so on.
The "Data" field contains 8000h for not implemented addresses or for information not relevant in the actual device configuration.

Example:
Ask to slave the value of 3 words (3h) starting from word 178 (B2h)

Request from master to slave	
Range	Byte
Slave address	FFh
Function code	03h
Word starting address (high byte)	00h
Word starting address (low byte)	B2h
Number of words (high byte)	00h
Number of words (low byte)	03h
Error check (CRC-16) (low byte)	B0h
Error check (CRC-16) (high byte)	32h

Reply from slave to master	
Range	Byte
Slave address	FFh
Function code	03h
Byte count	06h
Data	FFh
Data	9Ch
Data	80h
Data	00h
Data	05h
Data	5Ah
Error check (CRC-16) (low byte)	xxh
Error check (CRC-16) (high byte)	xxh

The 6 bytes in "Data" field (FFh, 9Ch, 80h, 00h, 05h, 5Ah) are 3 words whose meaning is:

word 178 value = -100 (FF9Ch)
word 179 value = not implemented or not relevant (8000h)
word 180 value = 1370 (55Ah)

Code 6: Single word writing

By using this command, the master unit can change the value of one word (16 bit) of the slave unit.

Command from master to slave	
Range	Byte
Slave address (0*-255)	1
Function code (06)	1
Word address (high byte)	1
Word address (low byte)	1
Data	2
Error check (CRC-16) (low byte)	1
Error check (CRC-16) (high byte)	1

Reply from slave to master	
Range	Byte
Slave address (0*-255)	1
Function code (06)	1
Word address (high byte)	1
Word address (low byte)	1
Data	2
Error check (CRC-16) (low byte)	1
Error check (CRC-16) (high byte)	1

* To use the address 0, see note 9 (“Broadcast” address) in the “Notes” section.

The 8000h value, present in the "data" field, should be considered as a don’t care value, that is, the value present in the device at this address will not be modified.

Example:
Set word 2006 (7D6h) of the slave with value 1250 (4E2h)

Command from master to slave	
Range	Byte
Slave address	FFh
Function code	06h
Word address (high byte)	07h
Word address (low byte)	D6h
Data	04h
Data	E2h
Error check (CRC-16) (low byte)	FEh
Error check (CRC-16) (high byte)	11h

Reply from slave to master	
Range	Byte
Slave address	FFh
Function code	06h
Word address (high byte)	07h
Word address (low byte)	D6h
Data	04h
Data	E2h
Error check(CRC-16) (low byte)	FEh
Error check (CRC-16) (high byte)	11h

Code 8: Diagnostic

By using this command, the master unit can check the communication system to Slaves.

Request from master to slave	
Range	Byte
Slave address (1-255)	1
Function code (08)	1
Sub-function (high byte)	1
Sub-function (low byte)	1
Data	2
Error check (CRC-16) (low byte)	1
Error check (CRC-16) (high byte)	1

Reply from slave to master	
Range	Byte
Slave address (1-255)	1
Function code (08)	1
Sub-function (high byte)	1
Sub-function (low byte)	1
Data	2
Error check (CRC-16) (low byte)	1
Error check (CRC-16) (high byte)	1

The Sub-function code will not be processed by Slave, any code is accept.
The Sub-function code and data passed in the request is returned (looped back) in the slave replay. The entire replay message is identical to the request

Example:

Request from master to slave	
Range	Byte
Slave address	FFh
Function code	08h
Sub-function (high byte)	00h
Sub-function (low byte)	00h
Data	55h
Data	AAh
Error check (CRC-16) (low byte)	4Ah
Error check (CRC-16) (high byte)	FAh

Reply from slave to master	
Range	Byte
Slave address	FFh
Function code	08h
Sub-function (high byte)	00h
Sub-function (low byte)	00h
Data	55h
Data	AAh
Error check (CRC-16) (low byte)	4Ah
Error check (CRC-16) (high byte)	FAh

Code 16: Multiple words writing

This function code is used by the master unit to write a consecutive group of words.
The master unit can change a maximum of 10 words at a time.

Command from master to slave	
Range	Byte
Slave address (0*-255)	1
Function code (16)	1
Word starting address (high byte)	1
Word starting address (low byte)	1
Number of words (high byte)	1
Number of words (low byte)	1
Byte counter	1
Data	n
Error check (CRC-16) (low byte)	1
Error check (CRC-16) (high byte)	1

Reply from slave to master	
Range	Byte
Slave address (0*-255)	1
Function code (16)	1
Word starting address (high byte)	1
Word starting address (low byte)	1
Number of words (high byte)	1
Number of words (low byte)	1
Error check (CRC-16) (low byte)	1
Error check (CRC-16) (high byte)	1

* To use the address 0, see note 9 (“Broadcast” address) in the “Notes” section.

The data imposed for read only words will be ignored.
The command will be processed starting from the first word and it will be executed or not executed depending on the actual device status.
At the first error found, the command will be aborted and the slave will answer with an error.
The 8000h value, present in the "data" field, should be considered as a don’t care value, this is, the value present in the device at this address will not be modified.

Example:
Set words 1301 (515h), 1302 (516h), 1303 (517h) of the slave with 300 (12Ch), don’t care (8000h) and 200 (C8h) values.

Command from master to slave	
Range	Byte
Slave address	FFh
Function code	10h
Word starting address(high byte)	05h
Word starting address (low byte)	15h
Number of words (high byte)	00h
Number of words (low byte)	03h
Byte counter	06h
Data	01h
Data	2Ch
Data	80h
Data	00h
Data	00h
Data	C8h
Error check (CRC-16) (low byte)	08h
Error check (CRC-16) (high byte)	F7h

Reply from slave to master	
Range	Byte
Slave address	FFh
Function code	10h
Word starting address (high byte)	05h
Word starting address (low byte)	15h
Number of words (high byte)	00h
Number of words (low byte)	03h
Error check (CRC-16) (low byte)	xxh
Error check (CRC-16) (high byte)	xxh

9.1.6 Error reply

If the "error check" is wrong or the function code is not implemented or a buffer over flows has been received, the slave does not send any reply to the master.
If other errors are detected in the request or command frame, or the slave cannot reply with the requested values or it cannot accept the requested sets because it is in error condition, the slave replies by forcing at "1" the bit 7 of the "Function code" byte followed by an error code.

Error reply (from slave to master):

RANGE	BYTE
Slave address	1
Function code (+80h)	1
Error code	1
Error check (CRC-16) (low byte)	1
Error check (CRC-16) (high byte)	1

List of error codes:

ERROR Nr.	DESCRIPTION
2	Illegal data address
3	Illegal data value
9	Illegal number of data required
10	The word indicated cannot be modified

Note

1. **Device communication parameter**

   When connected with CPI or forced in CPI mode by keyboard, the device is always set with the following parameters:

Address	1
Baud rate	9600
Bits	8
Parity	None

   In other case the device is set in according to P97-P98-P99 parameters

2. **Words format**
   All the parameters are represented by words.
   Every time the information transfer is performed by using 2 bytes (1 word of 16 bits), the first byte transmitted is the most significant one. For the negative numbers the "two complement" format is used.
3. **Reply time**

The slave will start to send a reply from 3.5 T.U. to 700 ms after the end of the incoming frame detected by counting the received bytes.

4. Decimal digits

The decimal point that may be present in the value is ignored.

Example:
The value 204.6 is transmitted as 2046 (07FEh)
The value -12.50 is transmitted as -1250 (FB1Eh)

5. Device status

When the factory com cable is inserted into device connector the device go automatically into checked configuration mode.

6. Parameters reading and writing mode.

Two different mode to accept data are allowed:

a) **Checked mode**
<u>This is the unique method available in operative mode</u>
This method must be advised like the normal method of reading and writing.
The flow chart in the following pages show the rules applied by the device in reading and writing mode.

In writing mode if the data is accepted, it is stored into EEPROM and the device change, if required, the content of the other parameters correlated with itself.
(I.e.: If the input type (P1) is changed, the device changes, if needed, the value of set point limits, the range values, the alarm threshold).
Therefore, after each change, the user program must reread the current value of all parameters or at least those that they could be changed.
This mode is selected by the word 153.

b) **Unchecked mode**
The flow chart in the following pages show the rules applied by the device in reading and writing mode.
In writing mode no test is done on the incoming data and no change is performed on correlated parameters.
This mode is selected by the word 153.

This mode must be used only with data formerly checked <u>and only in Configuration mode.</u>

It is faster than the precedent, it don't require a defined sequence of download and it could be used for example for download the formerly file, built and saved by the configuration program.

7. Test of the parameters.

The correctness of the stores data could be done on application request

Sending a request of reading the content of word 152, it is ordered to the device to effect a check of correctness of the all parameters.
If the result of the check is correct the content of the word will be 0 otherwise it will be equal to the modbus address of the first wrong parameter found.

The execution of these test results important:

- At starting time, when the device is connected to the factory com, to verify that the received data don't require corrections.
- At the end of an editing session in not protected mode, for audit the correctness of the data.

8. Read / write access permissions

The access permissions are stated for each parameter in the description tables by means of two columns named “read” and “write” according the following meaning:

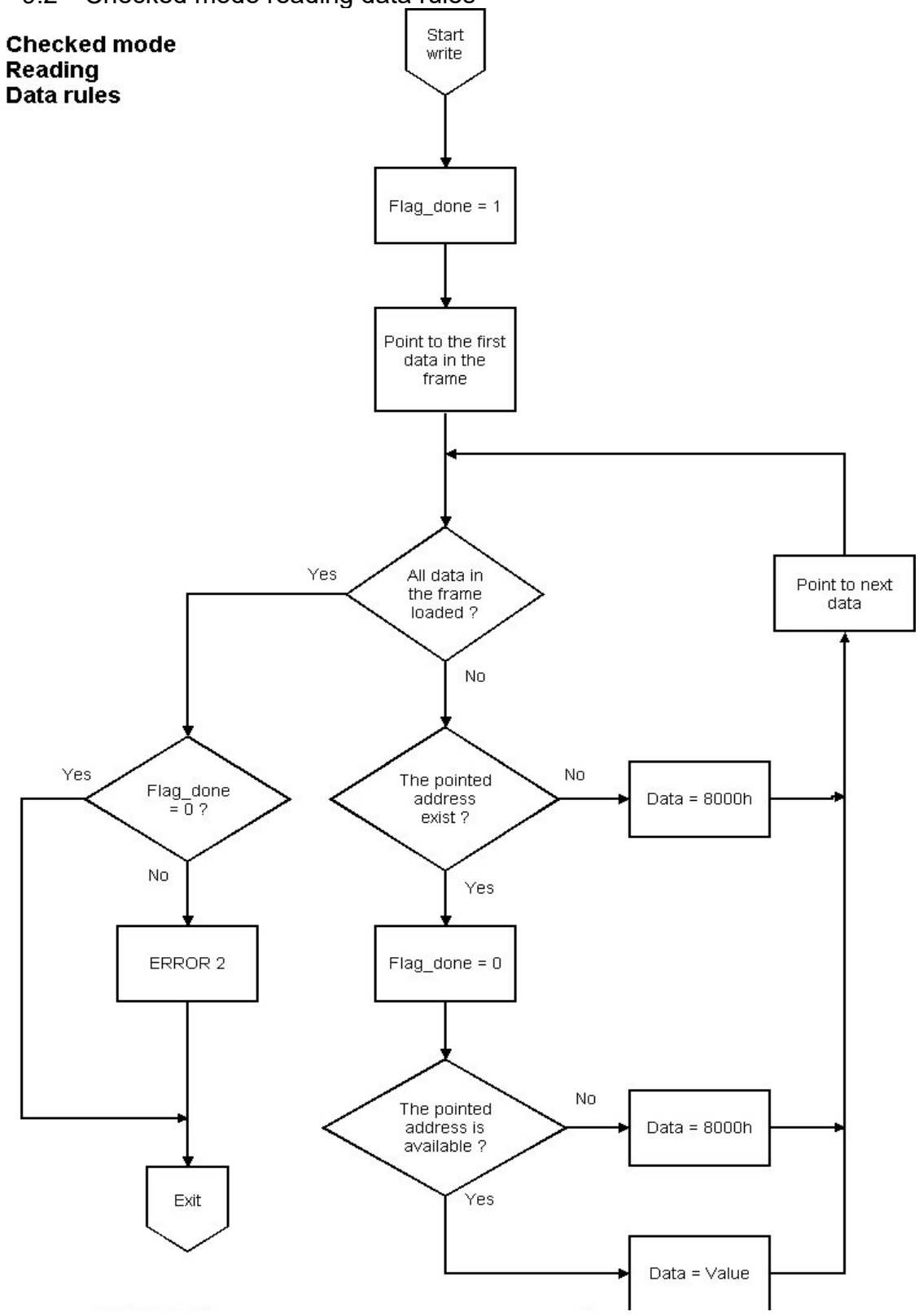
O transaction allowed in operative mode
C transaction allowed in configuration mode
L transaction allowed in calibration mode
F transaction allowed in factory test mode

9. Broadcast" address

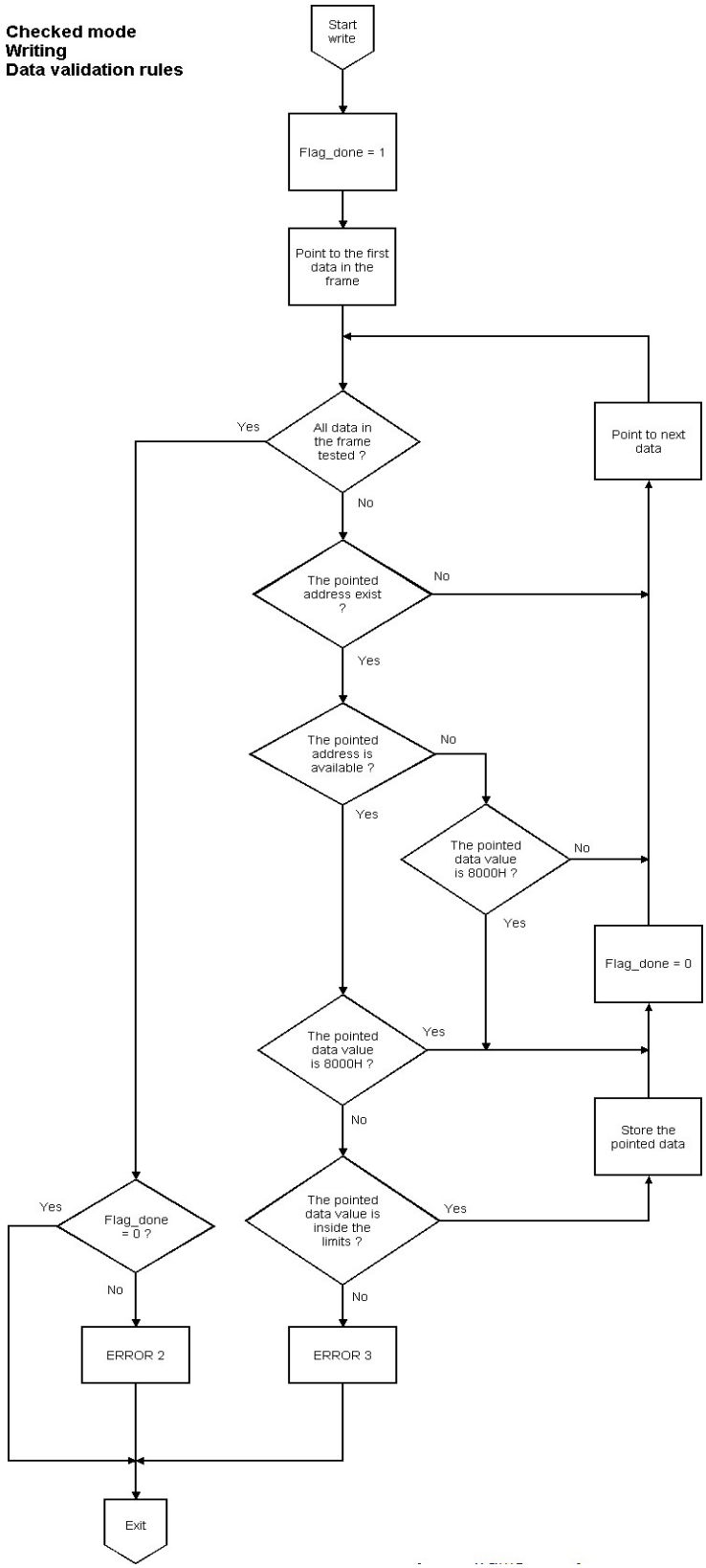
When using the writing codes (6 and 16) the slave address 0 is permitted:

In this case all the slaves connected accept the command but do not give any reply.

9.2 Checked mode reading data rules



9.3 Checking mode writing data validation rules



10 Used Modbus ADDRESSES

For the TAMCOM program following MODBUS addresses were used:
In case of setting up your own communication, the controller uses the Modbus standard for communication.
Following addresses can be used:

register 117 - read device class: throws exception if not equal to 419
register 119 - read serial number
register 153 - write 0: operative mode

getProcessValue()
register 601 - read process value

getSetPoint()
register 401 - read setpoint

setSetPoint()
register 401 - write setpoint

getOffset()
register 424 - read offset

setOffset()
register 424 - write offset

syncTempUnit()
register 201 - read: 7 means Celsius, 15 means Fahrenheit

getPb()
register 406 - read Pb

setPb()
register 163 - write 1: enable PID
register 159 - write 0: disable SMART
register 406 - write P

getI()
register 408 - read I

setI()
register 163 - write 1: enable PID
register 159 - write 0: disable SMART
register 408 - write I

getD()
register 409 - read D

setD()
register 163 - write 1: enable PID
register 159 - write 0: disable SMART
register 409 - write D

The following table offers a detailed overview of addresses available.

MODBUS ADDRESS	DESCRIPTION	MNEM CODE	READ	WRITE
117	**Firmware Device class** Availability: Always Value: 419 for CONTROLLER_S01		OCLF	
119	**Firmware revision** Availability: Always Value: Nr. of firmware revision		OCLF	
120	**Manufacturer trade mark** Availability: Always Value: 50 (32h)		OCLF	
121	**Device identification code** Availability: Always. Value: 1092 Nr. of firmware revision x 100 + identification code 92 for CONTROLLER_S01		OCLF	

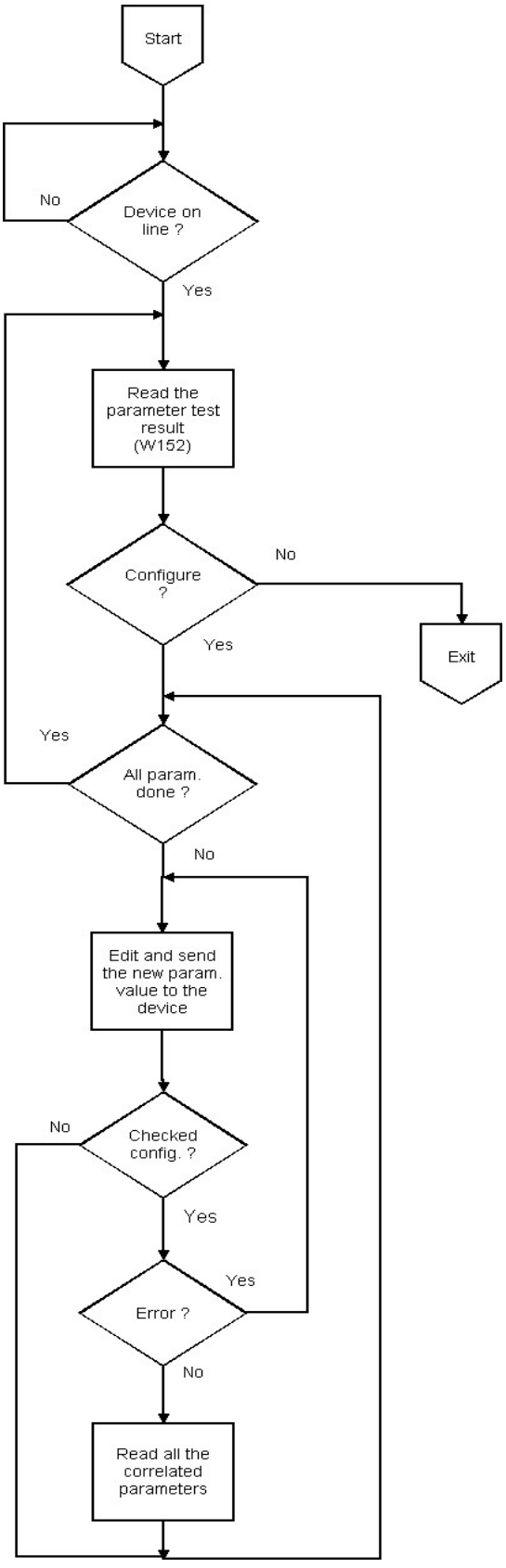
MODBUS ADDRESS	DESCRIPTION	MNEM CODE	READ	WRITE
153	**Device status** Availability: Always Range: 0 = Operative 1 = Input Calibration 2 = Checked Configuration 3 = Unchecked Configuration 4 = Factory test Note: When in operative mode only choices 2, 3 are allowed. After answer at choice 0 the device is reset and restarts using communication parameters set at P97-P98-P99 The choice 0 will not be accepted if CPI device is connected. See the mode description in the previous chapter.		OCLF	OCLF
156	**Smart Status** Availability: **If** **P15** **<>** **0** **and** **Pb**		OC	OC

	> **0** Range: 0 = Off 1 = On Note: If in Operative mode the set is not allowed if control disabled If set in Configuration, this status will be activated only when return in Operative mode.			
158	**SMART Option Status** Availability: Always Range: 0 = Option disabled 1 = Option enabled		O C L F	

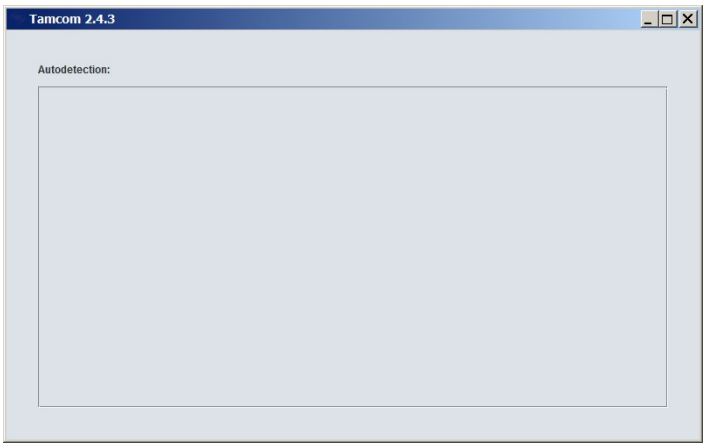
MODBUS ADDRESS	DESCRIPTION	MNEM CODE	READ	WRITE
201	**Input type and range value**	**("P1")**		
	Availability: Always Range : 7 = Rtd Pt 100 (- 90.0 .. 250.0°C 15 = Rtd Pt 100 (- 130.0 .. 482.0°F)		O C	C
215	**Smart function** Availability: If option is available. Otherwise it is forced to 0 Range: 0 = SMART function disabled 1 = SMART function may be enabled(TUNE+ADAPTIVE) 2 = SMART function may be enabled(TUNE only)	("**P15**")	O C	C
401	**Read / Write Set point** Availability: Always. Range: rL / rH Decimal figure: 1	("**SP1**")	O C	O C
406	**Proportional band** Availability: Always. Range : 10 – 1000 if P5 <> 5 15 – 1000 if P5 = 5 0 if ON/OFF control. Decimal figure: 1 Note: In operative mode the writing is not allowed if SMART is ON	("**Pb**")	O C	O C
408	**Integral time (in seconds)** Availability: Only if Pb > 0 Range: 1 / 1200 7FFFh = Integral action excluded Decimal figure:	("**ti**")	O C	O C

	0 Note: In operative mode the writing is not allowed if SMART is ON			
409	**Derivative time (in seconds)** Availability: Only if Pb > 0 Range: 0 / 600 Decimal figure: 0 Note: In operative mode the writing is not allowed if SMART is ON	("**td**")	O C	O C
424	**System calibration – "b" Coefficient** Availability: Always Range: -50 / 50 Decimal figures: 1 Note: This value can be modified by Auto (2 Pt) Calibration and from modbus word at Address 237	("**OFSt**")	O C	O C
601	**Measure value** Availability: Always Decimal figure: 2		O	
614	**Heating Power Output Value** Availability: Always Decimal figure: 0		O	

10.1 Suggested setting flow chart



11 Error solving



☹:
Display shows a blue screen
😐:
The computer has no serial ports
☺:
Install the USB to Serial adapter

☹:
A serial port is discovered but there's no communication
😐:
Bad wiring
Wrong Java edition
☺:
Check wiring and connectors
Check the Java version in the configuration screen

- 64 bit version gives problems.
- 32-bit version works without problems from release 6

12 DISCLAIMER

Information given herein is offered in good faith as accurate, but without guarantee. Conditions of use and suitability of the product for particular uses are beyond our control; all risks of use of the product are therefore assumed by the user and

We expressly disclaim all warranties of every kind and nature, including warranties of merchantability and fitness for a particular purpose in respect to the use or suitability of the product.

Nothing is intended as a recommendation for uses which infringe valid patents or as extending license under valid patents. Appropriate warnings and safe handling procedures should be provided to handlers and users. Alteration of this document is strictly prohibited. Except to the extent required by law, republication or retransmission of this document, in whole or in part, is not permitted. Tamson Instruments bv assume no responsibility for accuracy of information unless the document is the most current available. Tamson Instruments neither represent nor warrant that the format, content or product formulas contained in this document comply with the laws of any other country except the Netherlands.