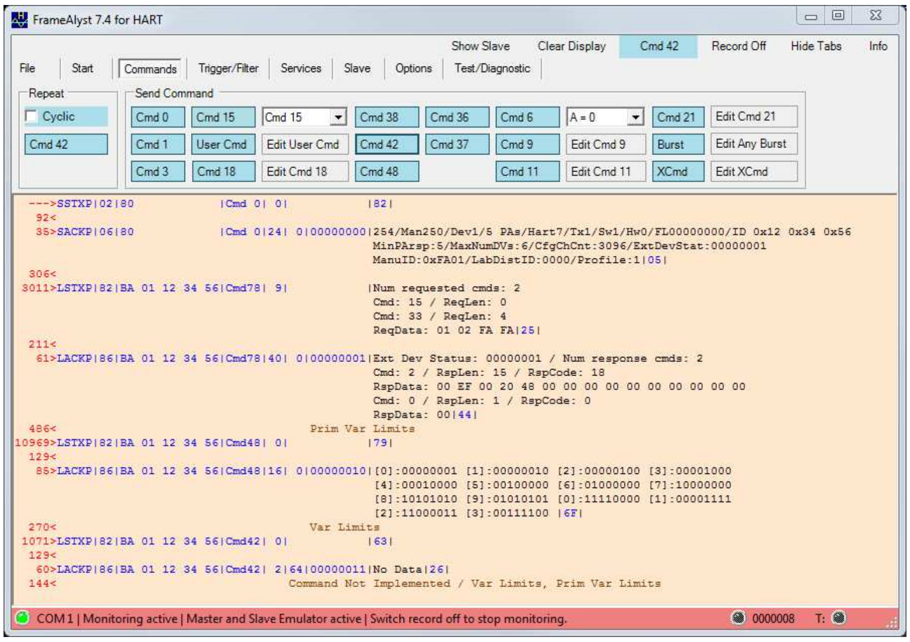
HartTools

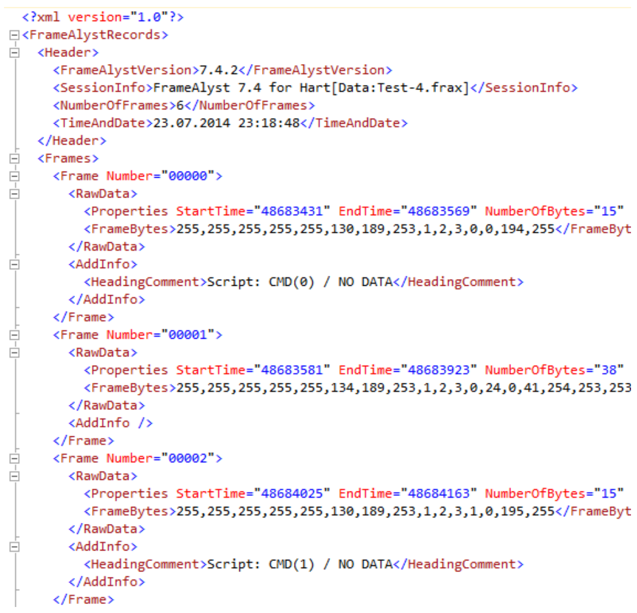
FrameAlyst 7.4
Software Documentation

Revision: 7.4.2
Date: 27.04.2015



```
 --->SSTXP|02|80           |Cmd  0|  0|           |82|
   92<
   35>SACKP|06|80          |Cmd  0|24|  0|00000000|254/Man250/Dev1/5 PAs/Hart7/Tx1/Sw1/Hw0/FL00000000/ID 0x12 0x34 0x56
                                                   MinPArsp:5/MaxNumDVs:6/CfgChCnt:3096/ExtDevStat:00000001
                                                   ManuID:0xFA01/LabDistID:0000/Profile:1|05|
  306<
 3011>LSTXP|82|BA 01 12 34 56|Cmd78|  9|          |Num requested cmds: 2
                                                   Cmd: 15 / ReqLen: 0
                                                   Cmd: 33 / ReqLen: 4
                                                   ReqData: 01 02 FA FA|25|
  211<
   61>LACKP|86|BA 01 12 34 56|Cmd78|40|  0|00000001|Ext Dev Status: 00000001 / Num response cmds: 2
                                                   Cmd: 2 / RspLen: 15 / RspCode: 18
                                                   RspData: 00 EF 00 20 48 00 00 00 00 00 00 00 00 00 00
                                                   Cmd: 0 / RspLen: 1 / RspCode: 0
                                                   RspData: 00|44|
  486<                                     Prim Var Limits
10969>LSTXP|82|BA 01 12 34 56|Cmd48|  0|          |79|
  129<
   85>LACKP|86|BA 01 12 34 56|Cmd48|16|  0|00000010|[0]:00000001 [1]:00000010 [2]:00000100 [3]:00001000
                                                   [4]:00010000 [5]:00100000 [6]:01000000 [7]:10000000
                                                   [8]:10101010 [9]:01010101 [0]:11110000 [1]:00001111
                                                   [2]:11000011 [3]:00111100 |6F|
  270<                                     Var Limits
 1071>LSTXP|82|BA 01 12 34 56|Cmd42|  0|          |63|
  129<
   60>LACKP|86|BA 01 12 34 56|Cmd42|  2|64|00000011|No Data|26|
  144<                                  Command Not Implemented / Var Limits, Prim Var Limits
```

COM 1 | Monitoring active | Master and Slave Emulator active | Switch record off to stop monitoring. 0000008 T:



```xml
<?xml version="1.0"?>
<FrameAlystRecords>
  <Header>
    <FrameAlystVersion>7.4.2</FrameAlystVersion>
    <SessionInfo>FrameAlyst 7.4 for Hart[Data:Test-4.frax]</SessionInfo>
    <NumberOfFrames>6</NumberOfFrames>
    <TimeAndDate>23.07.2014 23:18:48</TimeAndDate>
  </Header>
  <Frames>
    <Frame Number="00000">
      <RawData>
        <Properties StartTime="48683431" EndTime="48683569" NumberOfBytes="15"
        <FrameBytes>255,255,255,255,255,130,189,253,1,2,3,0,0,194,255</FrameByt
      </RawData>
      <AddInfo>
        <HeadingComment>Script: CMD(0) / NO DATA</HeadingComment>
      </AddInfo>
    </Frame>
    <Frame Number="00001">
      <RawData>
        <Properties StartTime="48683581" EndTime="48683923" NumberOfBytes="38"
        <FrameBytes>255,255,255,255,255,134,189,253,1,2,3,0,24,0,41,254,253,253
      </RawData>
      <AddInfo />
    </Frame>
    <Frame Number="00002">
      <RawData>
        <Properties StartTime="48684025" EndTime="48684163" NumberOfBytes="15"
        <FrameBytes>255,255,255,255,255,130,189,253,1,2,3,1,0,195,255</FrameByt
      </RawData>
      <AddInfo>
        <HeadingComment>Script: CMD(1) / NO DATA</HeadingComment>
      </AddInfo>
    </Frame>
```

Borst Automation
Neue Reihe 33
DE-27472 Cuxhaven
GERMANY

Fon: +49 (0)4721 6985100
Fax: +49 (0)6432 6985102

http://borst-automation.com

info@borst-automation.de

Borst Automation
Embedded Solutions

Contents

Introduction

When the development of FrameAlyst was started it was mainly targeted to simply monitoring Hart frames to detect errors in the device implementation.

Later the tool was expanded to use the HartDLL for the emulation of a master function.

In the recent years also a slave emulations was introduced. While in the latest implementation either a slave or a master emulation was available today the new FrameAlyst is supporting both functionalities at a time.

☞ User Version

There are two versions of FrameAlyst available. The user version is providing only master functionality and is used to debug Hart installations for process automation.

☞ Developer Version

The developer version is providing much more function than the user version, such as a slave emulation, sending special frames and commands, trigger functions, filtering and scripting.

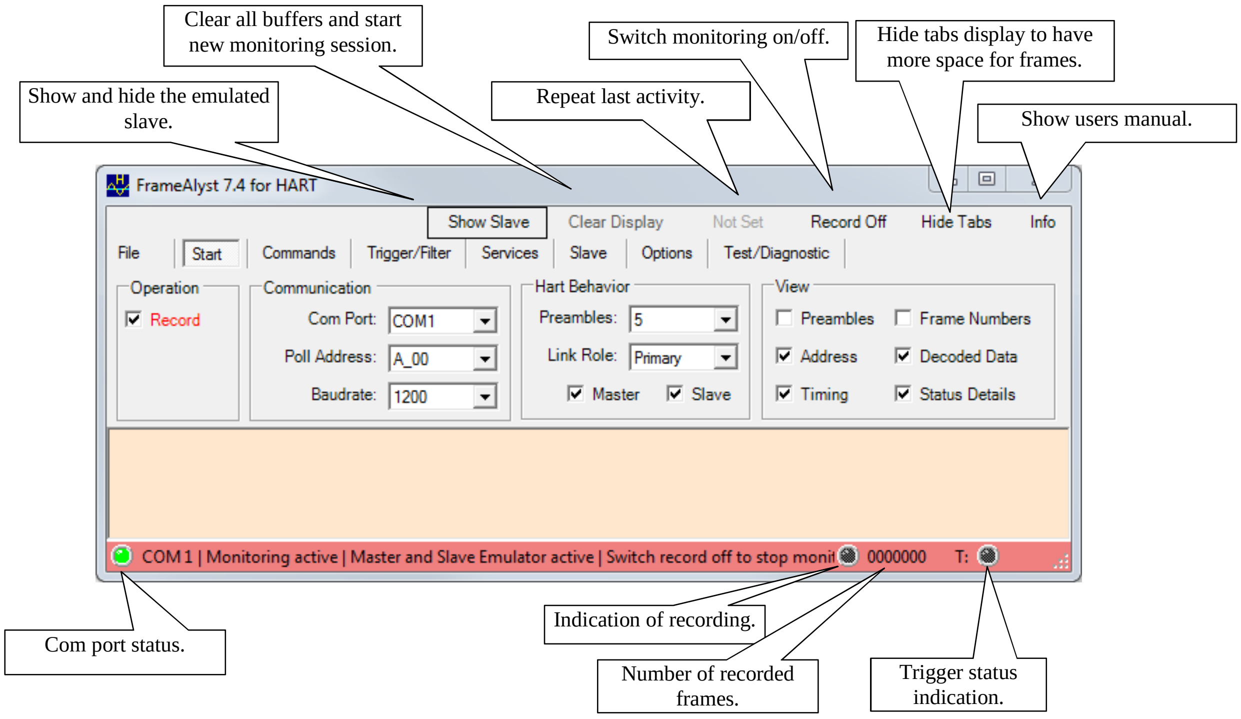
Functions Overview

The main features which are supported by FrameAlyst are the following.

- Full support of Hart 7.4
- Master emulation
- Slave emulation
- Slave DLL interface
- Trigger functions
- Filter functions
- Scripting
- Command data decoding
- Storing recorded data
- Test and diagnostic functions
- Integrated services
- Coding and Decoding
- Data syntax editor
- Data logging in xml-format

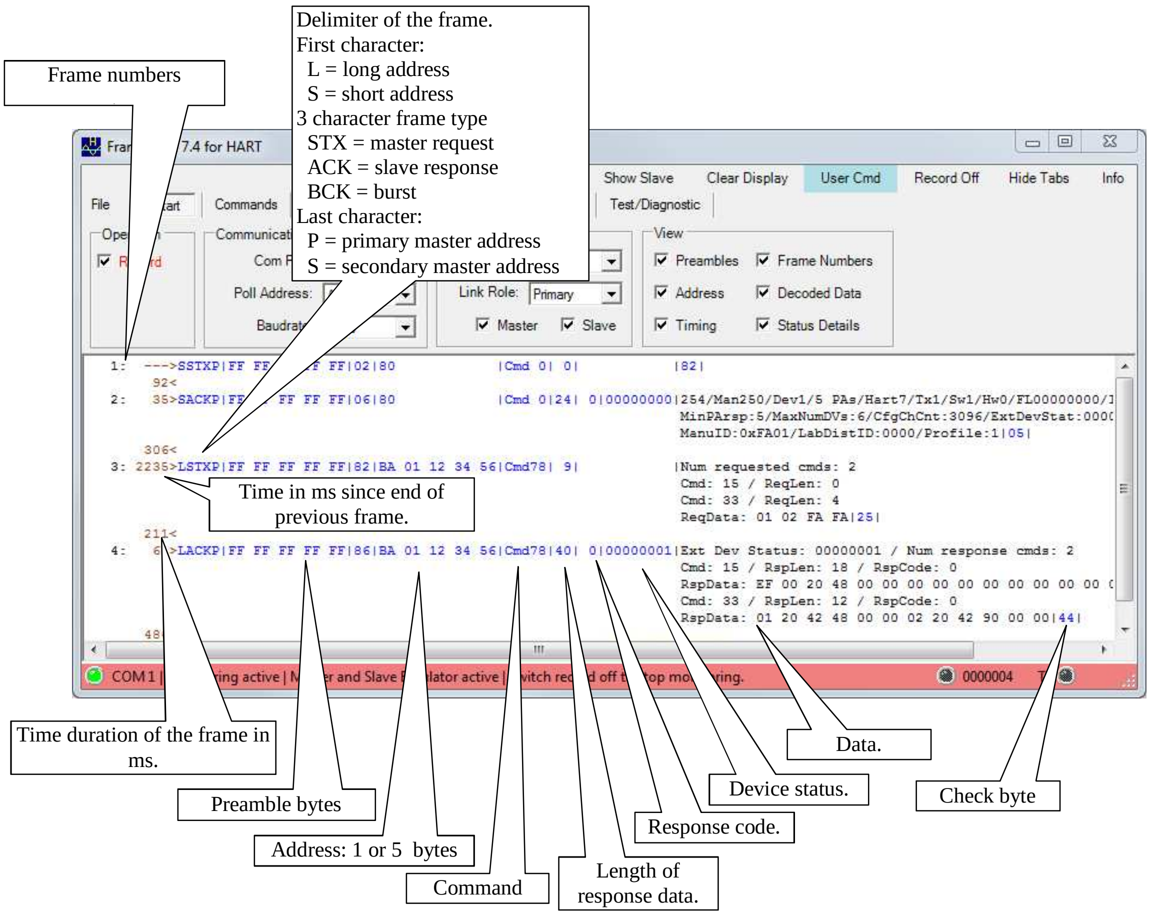
Operation Overview

The handling of FrameAlyst is based on tabs rather than menus.



COM 1 | Monitoring active | Master and Slave Emulator active | Switch record off to stop monit 0000000 T:

Display Items (Frames)

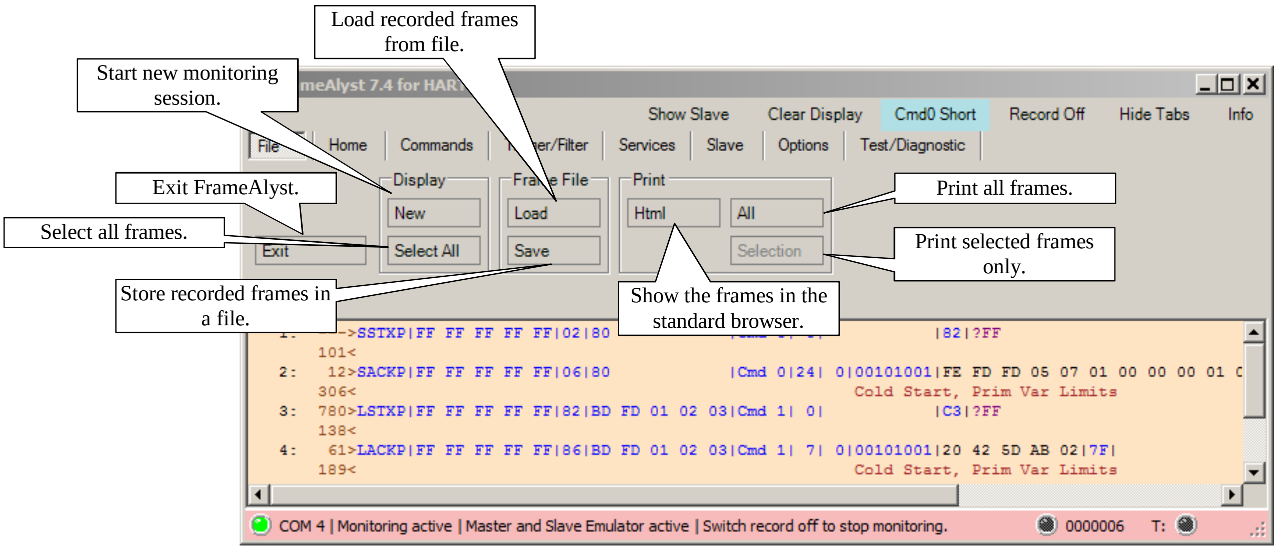


```
 1:   --->SSTXP|FF FF    F FF|02|80          |Cmd 0| 0|           |82|
       92<
 2:    35>SACKP|FF    FF FF FF|06|80          |Cmd 0|24| 0|00000000|254/Man250/Dev1/5 PAs/Hart7/Tx1/Sw1/Hw0/FL00000000/I
                                                                  MinPArsp:5/MaxNumDVs:6/CfgChCnt:3096/ExtDevStat:000
                                                                  ManuID:0xFA01/LabDistID:0000/Profile:1|05|

      306<
 3: 2235>LSTXP|FF FF FF FF FF|82|BA 01 12 34 56|Cmd78| 9|          |Num requested cmds: 2
                                                                  Cmd: 15 / ReqLen: 0
                                                                  Cmd: 33 / ReqLen: 4
                                                                  ReqData: 01 02 FA FA|25|
      211<
 4:     6>LACKP|FF FF FF FF FF|86|BA 01 12 34 56|Cmd78|40| 0|00000001|Ext Dev Status: 00000001 / Num response cmds: 2
                                                                  Cmd: 15 / RspLen: 18 / RspCode: 0
                                                                  RspData: EF 00 20 48 00 00 00 00 00 00 00 00 00 00 0
                                                                  Cmd: 33 / RspLen: 12 / RspCode: 0
                                                                  RspData: 01 20 42 48 00 00 02 20 42 90 00 00|44|
      48
```

COM 1 | ring active | M   er and Slave E   lator active | witch rec   d off t   top mo   ring. 0000004 T

Functions and Settings

File

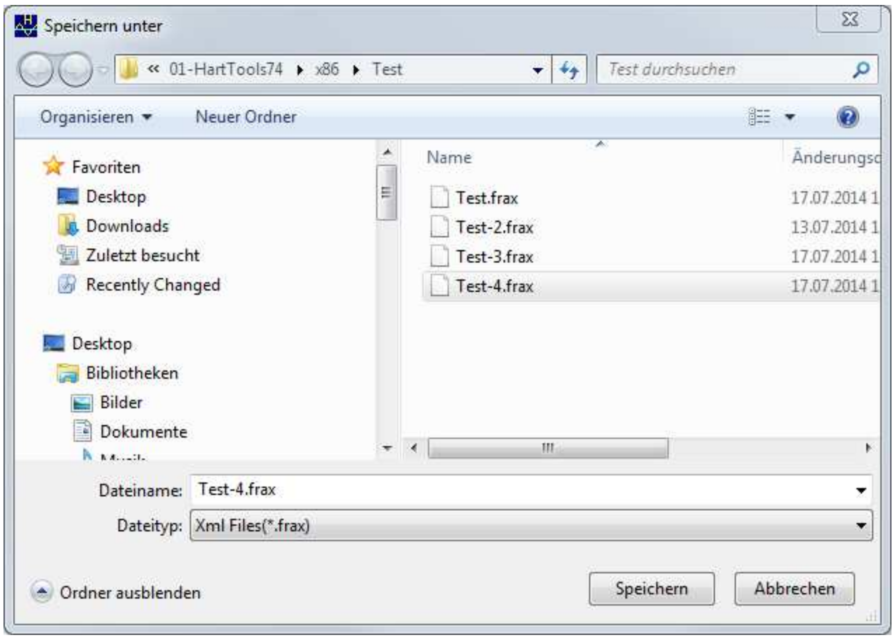


```
1:    ->SSTXP|FF FF FF FF FF|02|80                           |82|?FF
    101<
2:   12>SACKP|FF FF FF FF FF|06|80            |Cmd 0|24| 0|00101001|FE FD FD 05 07 01 00 00 00 01 0
    306<                                                  Cold Start, Prim Var Limits
3:  780>LSTXP|FF FF FF FF FF|82|BD FD 01 02 03|Cmd 1| 0|             |C3|?FF
    138<
4:   61>LACKP|FF FF FF FF FF|86|BD FD 01 02 03|Cmd 1| 7| 0|00101001|20 42 5D AB 02|7F|
    189<                                                  Cold Start, Prim Var Limits
```

COM 4 | Monitoring active | Master and Slave Emulator active | Switch record off to stop monitoring. 0000006 T:

The frames are still stored in the format which was used in the past. However when saving the frame data you may also select an xml format or html format.

Store in Xml and Html Format



Name	Änderungsd
Test.frax	17.07.2014 1
Test-2.frax	13.07.2014 1
Test-3.frax	17.07.2014 1
Test-4.frax	17.07.2014 1

Dateiname: Test-4.frax

Dateityp: Xml Files(*.frax)

If you select the file extension .frax, the frames will be strored in xml format.

Alternativly you may also choose an html format as a documentation of the debug session.

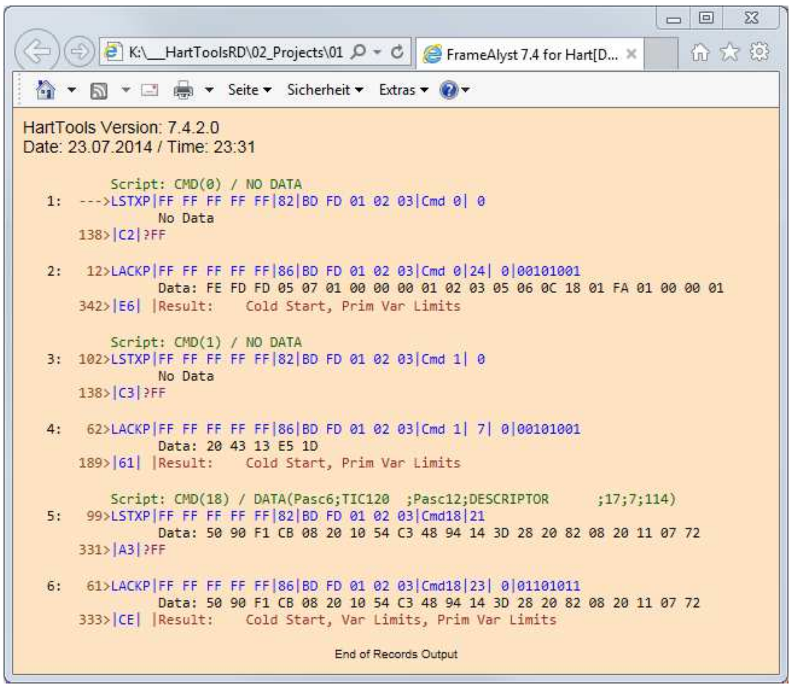
An example of an xml output is shown on the following page.

Xml Format Example

```xml
<?xml version="1.0"?>
<FrameAlystRecords>
  <Header>
    <FrameAlystVersion>7.4.2</FrameAlystVersion>
    <SessionInfo>FrameAlyst 7.4 for Hart[Data:Test-4.frax]</SessionInfo>
    <NumberOfFrames>6</NumberOfFrames>
    <TimeAndDate>23.07.2014 23:18:48</TimeAndDate>
  </Header>
  <Frames>
    <Frame Number="00000">
      <RawData>
        <Properties StartTime="48683431" EndTime="48683569" NumberOfBytes="15" WasGapTimeOut="False" ClientTxFlag="True" IsValidFrame="True" />
        <FrameBytes>255,255,255,255,255,130,189,253,1,2,3,0,0,194,255</FrameBytes>
      </RawData>
      <AddInfo>
        <HeadingComment>Script: CMD(0) / NO DATA</HeadingComment>
      </AddInfo>
    </Frame>
    <Frame Number="00001">
      <RawData>
        <Properties StartTime="48683581" EndTime="48683923" NumberOfBytes="38" WasGapTimeOut="False" ClientTxFlag="False" IsValidFrame="True" />
        <FrameBytes>255,255,255,255,255,134,189,253,1,2,3,0,24,0,41,254,253,253,5,7,1,0,0,0,1,2,3,5,6,12,24,1,250,1,0,0,1,230</FrameBytes>
      </RawData>
      <AddInfo />
    </Frame>
    <Frame Number="00002">
      <RawData>
        <Properties StartTime="48684025" EndTime="48684163" NumberOfBytes="15" WasGapTimeOut="False" ClientTxFlag="True" IsValidFrame="True" />
        <FrameBytes>255,255,255,255,255,130,189,253,1,2,3,1,0,195,255</FrameBytes>
      </RawData>
      <AddInfo>
        <HeadingComment>Script: CMD(1) / NO DATA</HeadingComment>
      </AddInfo>
    </Frame>
```

Regarding Html format you may either store the records in an Html file or click 'html' in the print functions. The print function for 'html' is opening your standard browser directly to display the frames.

Html Output Example

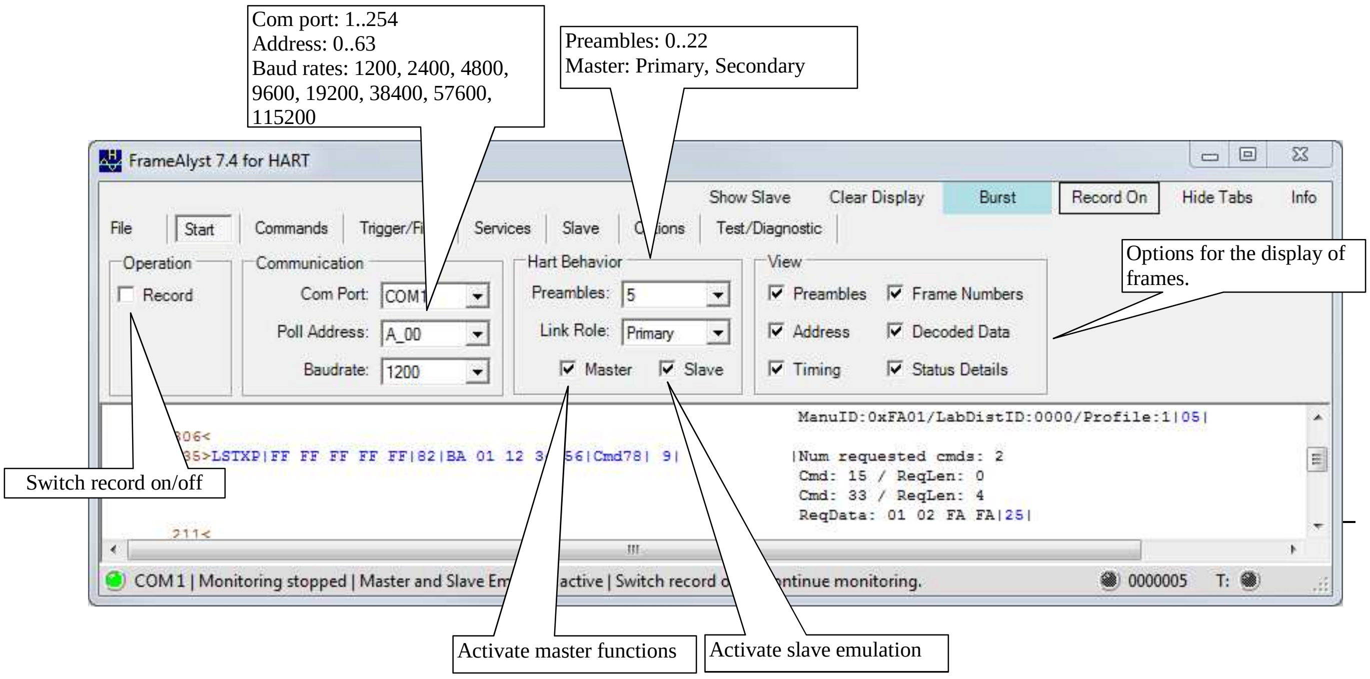


HartTools Version: 7.4.2.0
Date: 23.07.2014 / Time: 23:31

```
        Script: CMD(0) / NO DATA
  1:  --->LSTXP|FF FF FF FF FF|82|BD FD 01 02 03|Cmd 0| 0
              No Data
      138>|C2|?FF

  2:   12>LACKP|FF FF FF FF FF|86|BD FD 01 02 03|Cmd 0|24| 0|00101001
              Data: FE FD FD 05 07 01 00 00 00 01 02 03 05 06 0C 18 01 FA 01 00 00 01
      342>|E6| |Result:    Cold Start, Prim Var Limits

        Script: CMD(1) / NO DATA
  3:  102>LSTXP|FF FF FF FF FF|82|BD FD 01 02 03|Cmd 1| 0
              No Data
      138>|C3|?FF

  4:   62>LACKP|FF FF FF FF FF|86|BD FD 01 02 03|Cmd 1| 7| 0|00101001
              Data: 20 43 13 E5 1D
      189>|61| |Result:    Cold Start, Prim Var Limits

        Script: CMD(18) / DATA(Pasc6;TIC120  ;Pasc12;DESCRIPTOR     ;17;7;114)
  5:   99>LSTXP|FF FF FF FF FF|82|BD FD 01 02 03|Cmd18|21
              Data: 50 90 F1 CB 08 20 10 54 C3 48 94 14 3D 28 20 82 08 20 11 07 72
      331>|A3|?FF

  6:   61>LACKP|FF FF FF FF FF|86|BD FD 01 02 03|Cmd18|23| 0|01101011
              Data: 50 90 F1 CB 08 20 10 54 C3 48 94 14 3D 28 20 82 08 20 11 07 72
      333>|CE| |Result:    Cold Start, Var Limits, Prim Var Limits
```

End of Records Output

Start



Com port: 1..254
Address: 0..63
Baud rates: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200

Preambles: 0..22
Master: Primary, Secondary

Options for the display of frames.

Switch record on/off

Activate master functions

Activate slave emulation

Operation: ☐ Record

Communication:
Com Port: COM1
Poll Address: A_00
Baudrate: 1200

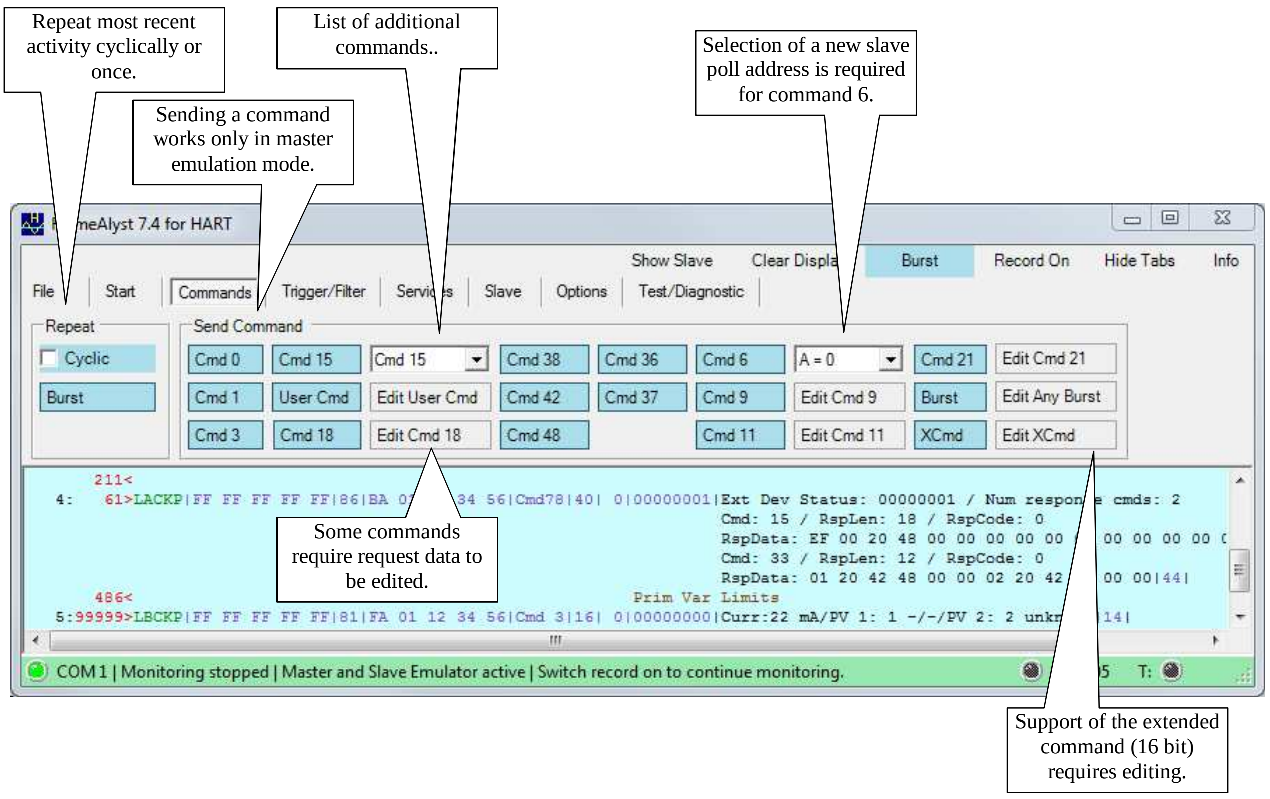
Hart Behavior:
Preambles: 5
Link Role: Primary
☑ Master ☑ Slave

View:
☑ Preambles ☑ Frame Numbers
☑ Address ☑ Decoded Data
☑ Timing ☑ Status Details

```
 06<                                                     ManuID:0xFA01/LabDistID:0000/Profile:1|05|
 35>LSTXP|FF FF FF FF FF|82|BA 01 12 3  56|Cmd78| 9|    |Num requested cmds: 2
                                                         Cmd: 15 / ReqLen: 0
                                                         Cmd: 33 / ReqLen: 4
                                                         ReqData: 01 02 FA FA|25|
 211<
```

COM 1 | Monitoring stopped | Master and Slave Em active | Switch record o ntinue monitoring. 0000005 T:

Hart Commands


Repeat most recent activity cyclically or once.

List of additional commands..

Selection of a new slave poll address is required for command 6.

Sending a command works only in master emulation mode.

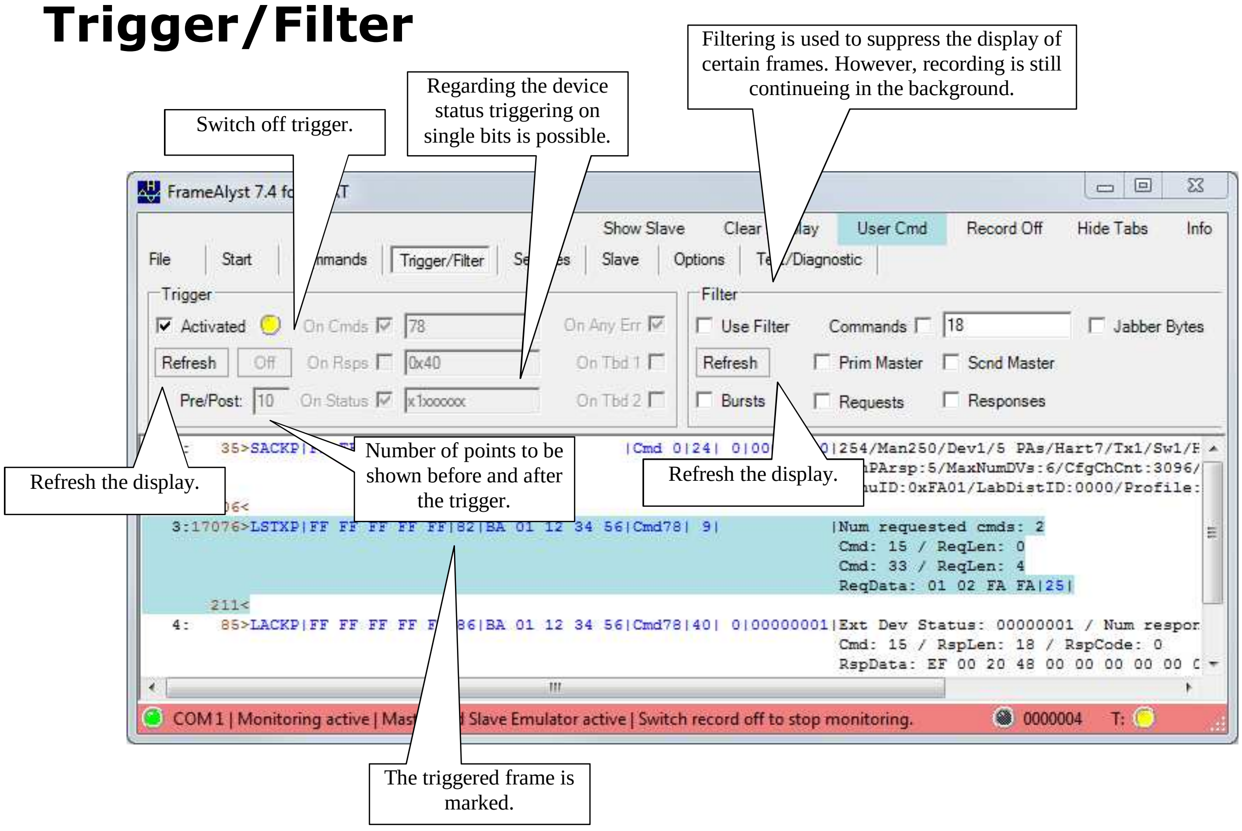
Some commands require request data to be edited.

Support of the extended command (16 bit) requires editing.

```
    211<
4:   61>LACKP|FF FF FF FF FF|86|BA 0       34 56|Cmd78|40|  0|00000001|Ext Dev Status: 00000001 / Num respon e cmds: 2
                                                                      Cmd: 15 / RspLen: 18 / RspCode: 0
                                                                      RspData: EF 00 20 48 00 00 00 00 00      00 00 00 00 0
                                                                      Cmd: 33 / RspLen: 12 / RspCode: 0
                                                                      RspData: 01 20 42 48 00 00 02 20 42      00 00|44|
    486<                                                        Prim Var Limits
5:99999>LBCKP|FF FF FF FF FF|81|FA 01 12 34 56|Cmd  3|16|  0|00000000|Curr:22 mA/PV 1: 1 -/-/PV 2: 2 unkr      |14|
```

COM 1 | Monitoring stopped | Master and Slave Emulator active | Switch record on to continue monitoring.

Trigger/Filter


Switch off trigger.

Regarding the device status triggering on single bits is possible.

Filtering is used to suppress the display of certain frames. However, recording is still continueing in the background.

Refresh the display.

Number of points to be shown before and after the trigger.

Refresh the display.

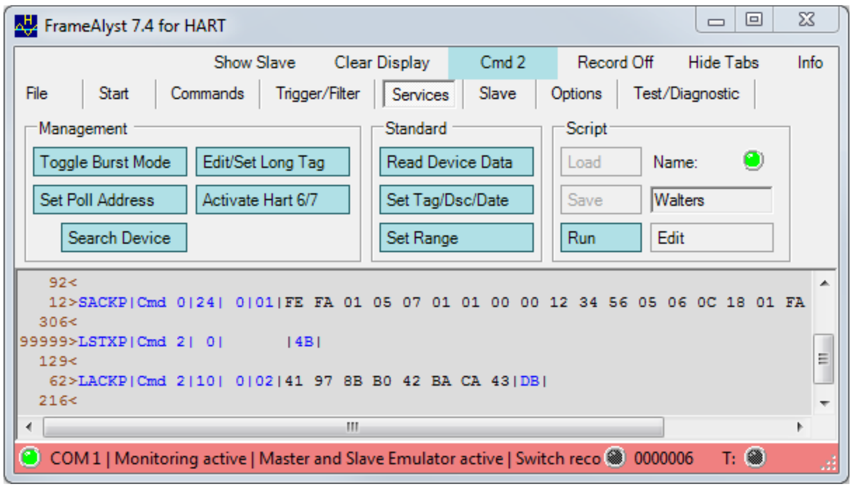
The triggered frame is marked.

```
      35>SACKP|                 |Cmd  0|24|  0|00      0|254/Man250/Dev1/5 PAs/Hart7/Tx1/Sw1/F
                                                         PArsp:5/MaxNumDVs:6/CfgChCnt:3096/
    06<                                                  uID:0xFA01/LabDistID:0000/Profile:
  3:17076>LSTXP|FF FF FF FF FF|82|BA 01 12 34 56|Cmd78|  9|         |Num requested cmds: 2
                                                                      Cmd: 15 / ReqLen: 0
                                                                      Cmd: 33 / ReqLen: 4
                                                                      ReqData: 01 02 FA FA|25|
      211<
  4:   85>LACKP|FF FF FF FF F  86|BA 01 12 34 56|Cmd78|40|  0|00000001|Ext Dev Status: 0000001 / Num respor
                                                                      Cmd: 15 / RspLen: 18 / RspCode: 0
                                                                      RspData: EF 00 20 48 00 00 00 00 0
```

COM 1 | Monitoring active | Mast   Slave Emulator active | Switch record off to stop monitoring. 0000004

Services

Services are some more complex functions as only sending a command.



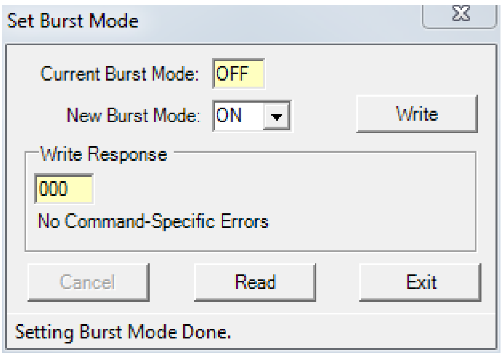
Name: Walters

```
   92<
   12>SACKP|Cmd 0|24| 0|01|FE FA 01 05 07 01 01 00 00 12 34 56 05 06 0C 18 01 FA
  306<
99999>LSTXP|Cmd 2| 0|      |4B|
  129<
   62>LACKP|Cmd 2|10| 0|02|41 97 8B B0 42 BA CA 43|DB|
  216<
```

COM 1 | Monitoring active | Master and Slave Emulator active | Switch reco 0000006 T:

The services are only working if the FrameAlyst is using the master emulation.

Toggle Burst Mode



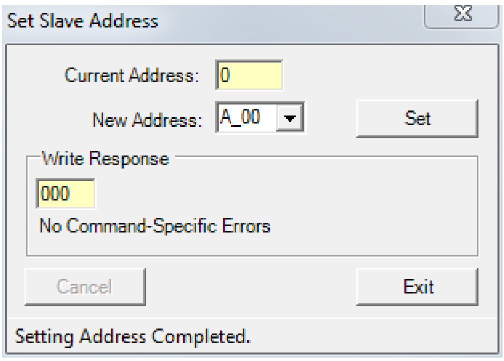
Current Burst Mode: OFF
New Burst Mode: ON

000
No Command-Specific Errors

Setting Burst Mode Done.

This service is handling command 109.

Set Poll Address



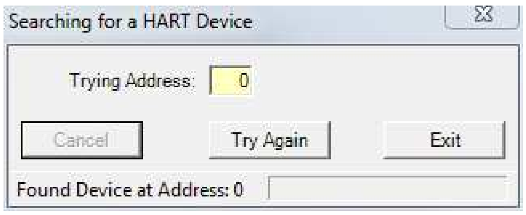
Current Address: 0
New Address: A_00

000
No Command-Specific Errors

Setting Address Completed.

Set slave poll address is handling command 6. Note: Hart5 is only supporting addresses 0..15 while Hart 7 has a range of 0..63.

Search Device



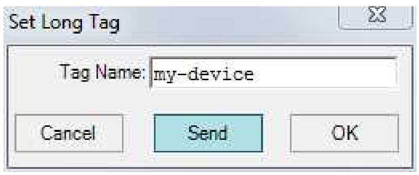
Searching for a HART Device

Trying Address: 0

Found Device at Address: 0

This service searches for slaves in a range of poll addresses from 0 to 63.

Edit/Set Long Tag



Set Long Tag

Tag Name: my-device

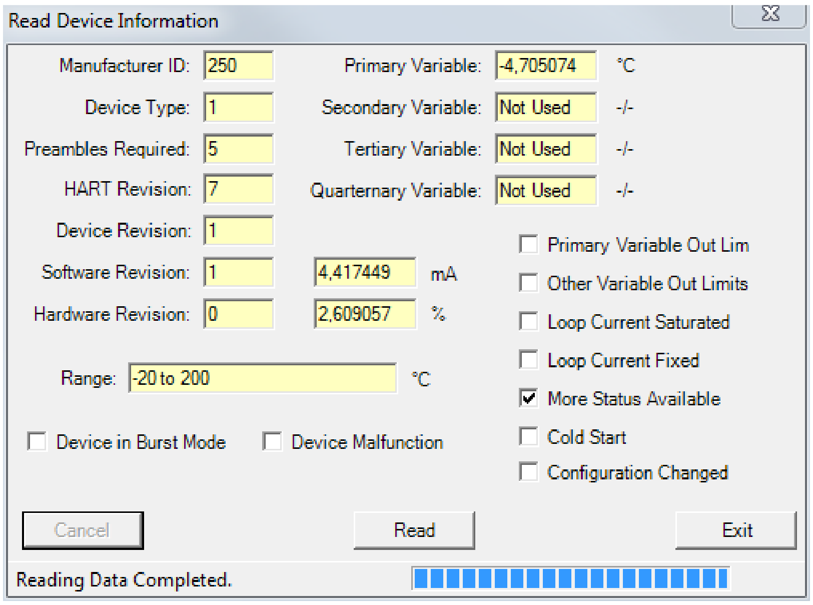
The long tag is an iso latin-1 string of a length of a maximum of 32 characters. If it contains less than 32 characters it is terminated by 0x00.

Activate Hart 6/7

There is no form provided which is used to realize this mean.

The service is using commands 7 and 6 to signal the slave device that a Hart 6/7 host is connected.

Read Device Data



Read Device Information

Manufacturer ID:	250	Primary Variable:	-4,705074	°C
Device Type:	1	Secondary Variable:	Not Used	-/-
Preambles Required:	5	Tertiary Variable:	Not Used	-/-
HART Revision:	7	Quarternary Variable:	Not Used	-/-
Device Revision:	1			
Software Revision:	1	4,417449	mA	
Hardware Revision:	0	2,609057	%	

Range: -20 to 200 °C

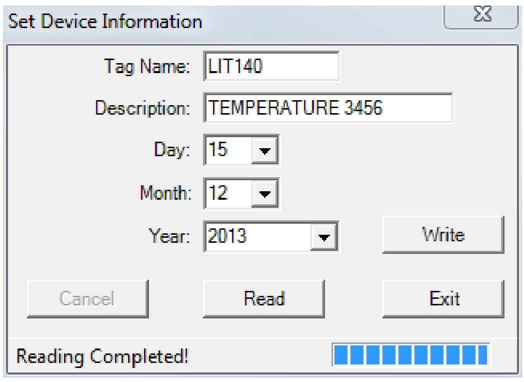
☐ Device in Burst Mode ☐ Device Malfunction

☐ Primary Variable Out Lim
☐ Other Variable Out Limits
☐ Loop Current Saturated
☐ Loop Current Fixed
☑ More Status Available
☐ Cold Start
☐ Configuration Changed

Reading Data Completed.

This service is reading the main information from a device.

Set Tag/Dsc/Date



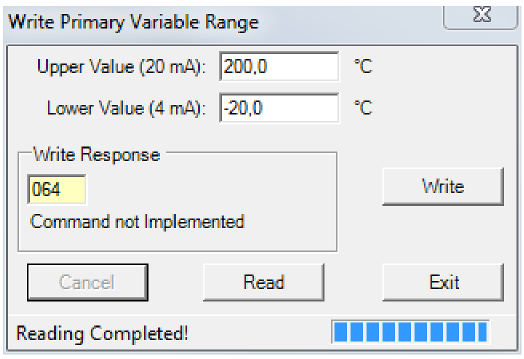
Set Device Information

Tag Name: LIT140
Description: TEMPERATURE 3456
Day: 15
Month: 12
Year: 2013

Reading Completed!

This application is setting the short tag, the descriptor and the date.

Set Range



Write Primary Variable Range

Upper Value (20 mA): 200,0 °C
Lower Value (4 mA): -20,0 °C

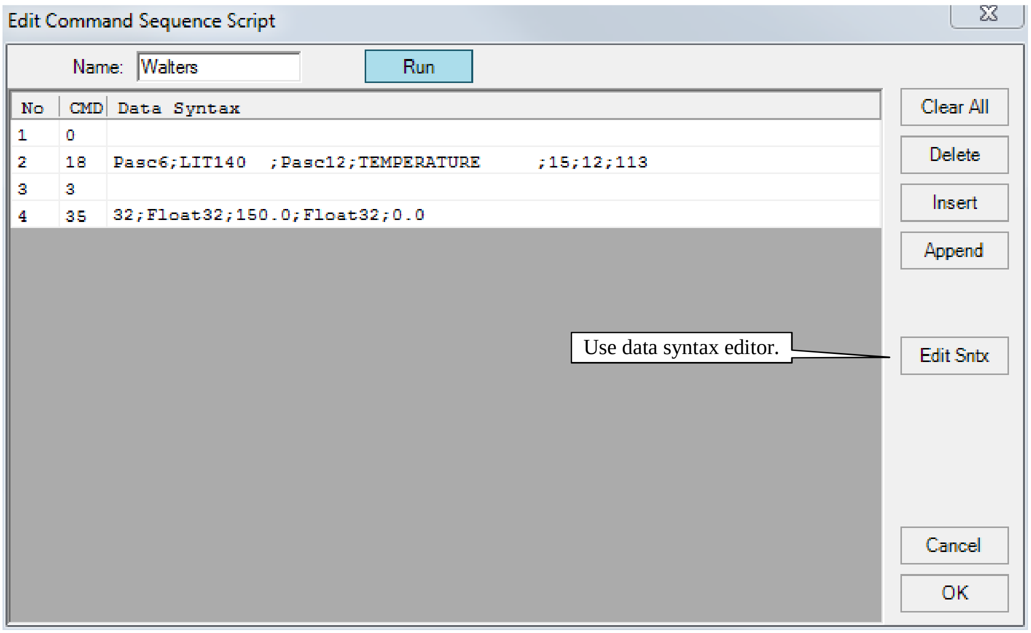
Write Response
064
Command not Implemented

Reading Completed!

The service is trying to write the upper and the lower range value of the primary variable of a device.

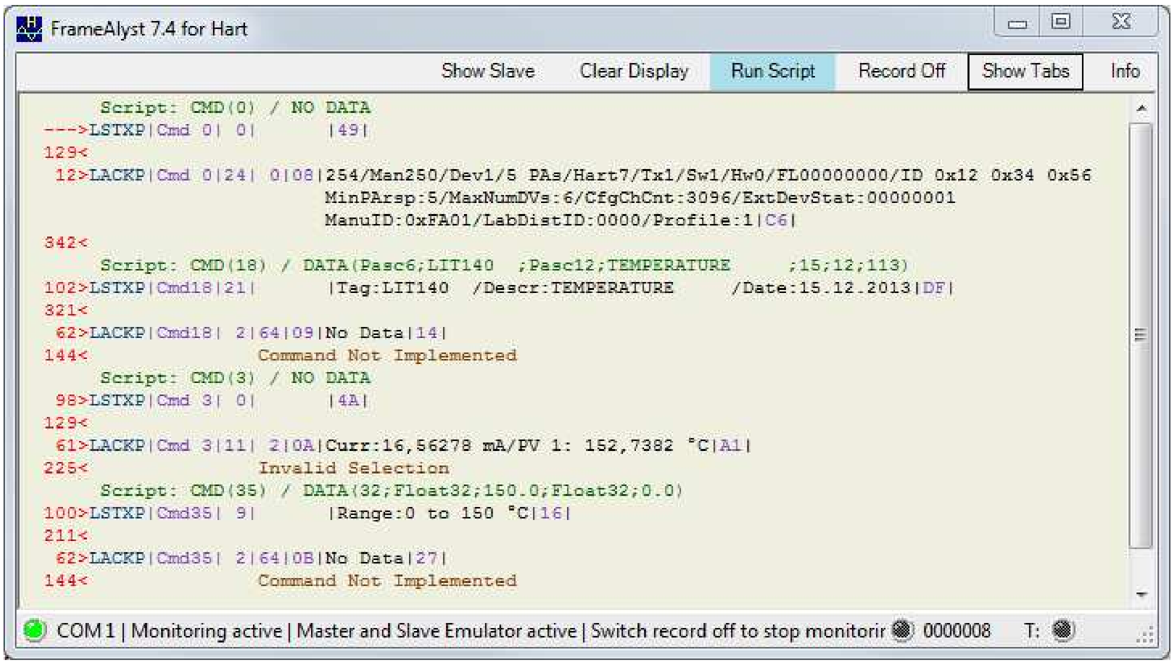
Edit/Run Scripts

Running scripts is simply sending a series of commands with or without request data.



No	CMD	Data Syntax
1	0	
2	18	Pasc6;LIT140 ;Pasc12;TEMPERATURE ;15;12;113
3	3	
4	35	32;Float32;150.0;Float32;0.0

The example above is sending the commands 0, 18, 3 and 35.



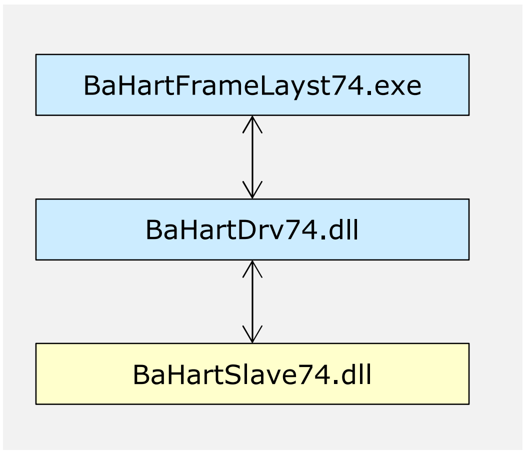
```
      Script: CMD(0) / NO DATA
--->LSTXP|Cmd 0| 0|       |49|
129<
 12>LACKP|Cmd 0|24| 0|08|254/Man250/Dev1/5 PAs/Hart7/Tx1/Sw1/Hw0/FL00000000/ID 0x12 0x34 0x56
                        MinPArsp:5/MaxNumDVs:6/CfgChCnt:3096/ExtDevStat:00000001
                        ManuID:0xFA01/LabDistID:0000/Profile:1|C6|
342<
      Script: CMD(18) / DATA(Pasc6;LIT140  ;Pasc12;TEMPERATURE     ;15;12;113)
102>LSTXP|Cmd18|21|       |Tag:LIT140  /Descr:TEMPERATURE     /Date:15.12.2013|DF|
321<
 62>LACKP|Cmd18| 2|64|09|No Data|14|
144<              Command Not Implemented
      Script: CMD(3) / NO DATA
 98>LSTXP|Cmd 3| 0|       |4A|
129<
 61>LACKP|Cmd 3|11| 2|0A|Curr:16,56278 mA/PV 1: 152,7382 °C|A1|
225<              Invalid Selection
      Script: CMD(35) / DATA(32;Float32;150.0;Float32;0.0)
100>LSTXP|Cmd35| 9|       |Range:0 to 150 °C|16|
211<
 62>LACKP|Cmd35| 2|64|0B|No Data|27|
144<              Command Not Implemented
```

COM 1 | Monitoring active | Master and Slave Emulator active | Switch record off to stop monitorir 0000008 T:

The script may be stored in a file and be loaded from a file. The active script is always stored in the settings of the software and automatically reloaded after the start of FrameAlyst.

If command 255 is specified in the script, the data will be sent as is not formatted as a Hart frame.

Slave



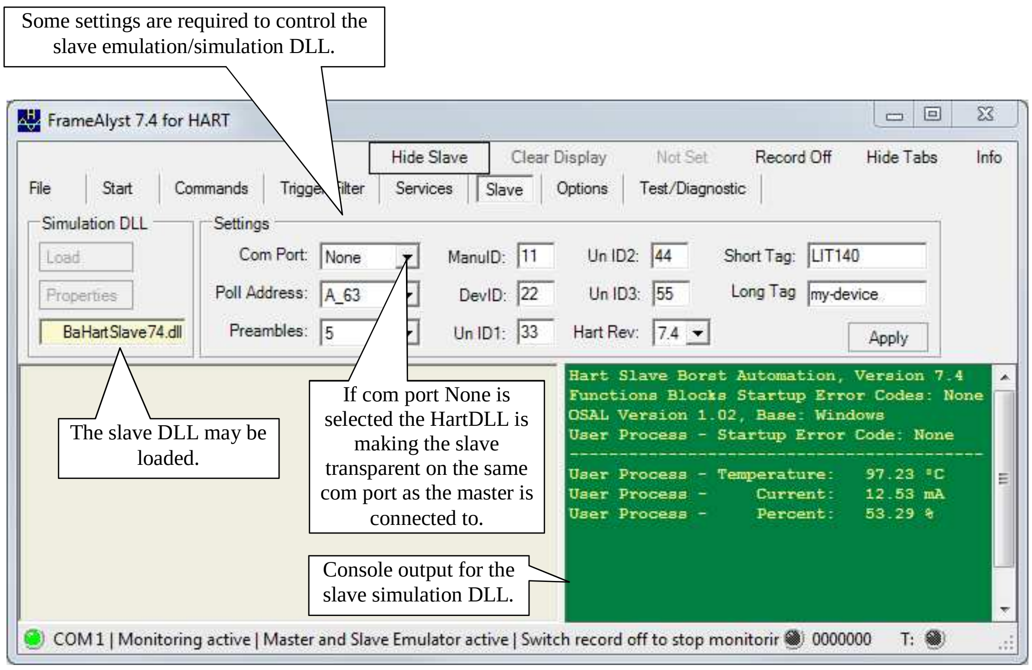
The HartDLL provides a standard interface for DLL access which is used by the FrameAlyst for the slave emulation.

The HartDLL is loading a specified DLL containing the slave simulation. As a default the HartDLL is loading BaHartSlave.dll.

With other words: the user may provide his own DLL for a slave emulation.

Figure 1: Slave Emulation Architecture

The slave may be configured through FrameAlyst.



Some settings are required to control the slave emulation/simulation DLL.

BaHartSlave74.dll

Com Port: None
Poll Address: A_63
Preambles: 5
ManuID: 11
DevID: 22
Un ID1: 33
Un ID2: 44
Un ID3: 55
Hart Rev: 7.4
Short Tag: LIT140
Long Tag: my-device

The slave DLL may be loaded.

If com port None is selected the HartDLL is making the slave transparent on the same com port as the master is connected to.

Console output for the slave simulation DLL.

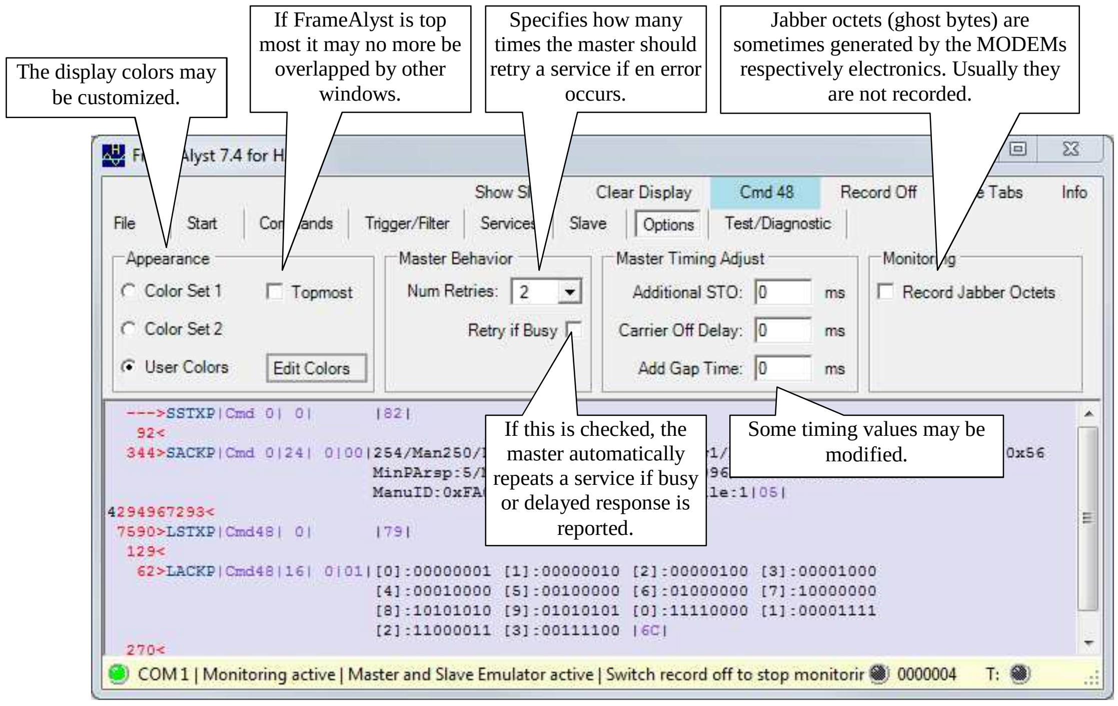
```
Hart Slave Borst Automation, Version 7.4
Functions Blocks Startup Error Codes: None
OSAL Version 1.02, Base: Windows
User Process - Startup Error Code: None
--------------------------------------------
User Process - Temperature:   97.23 °C
User Process -     Current:   12.53 mA
User Process -     Percent:   53.29 %
```

COM 1 | Monitoring active | Master and Slave Emulator active | Switch record off to stop monitorir 0000000 T:

The slave interface of the HartDLL allows the developer to write most of the slave software by using a simulation hosted by the HartDLL and the FrameAlyst.

Because the slave can be made transparent through the com port it can be tested also in a multidrop environment as well as with various Hart hosts.

Options



The display colors may be customized.

If FrameAlyst is top most it may no more be overlapped by other windows.

Specifies how many times the master should retry a service if en error occurs.

Jabber octets (ghost bytes) are sometimes generated by the MODEMs respectively electronics. Usually they are not recorded.

If this is checked, the master automatically repeats a service if busy or delayed response is reported.

Some timing values may be modified.

Appearance: Color Set 1 / Color Set 2 / [✓ User Colors]; Topmost ☐

Master Behavior: Num Retries: 2; Retry if Busy ☐

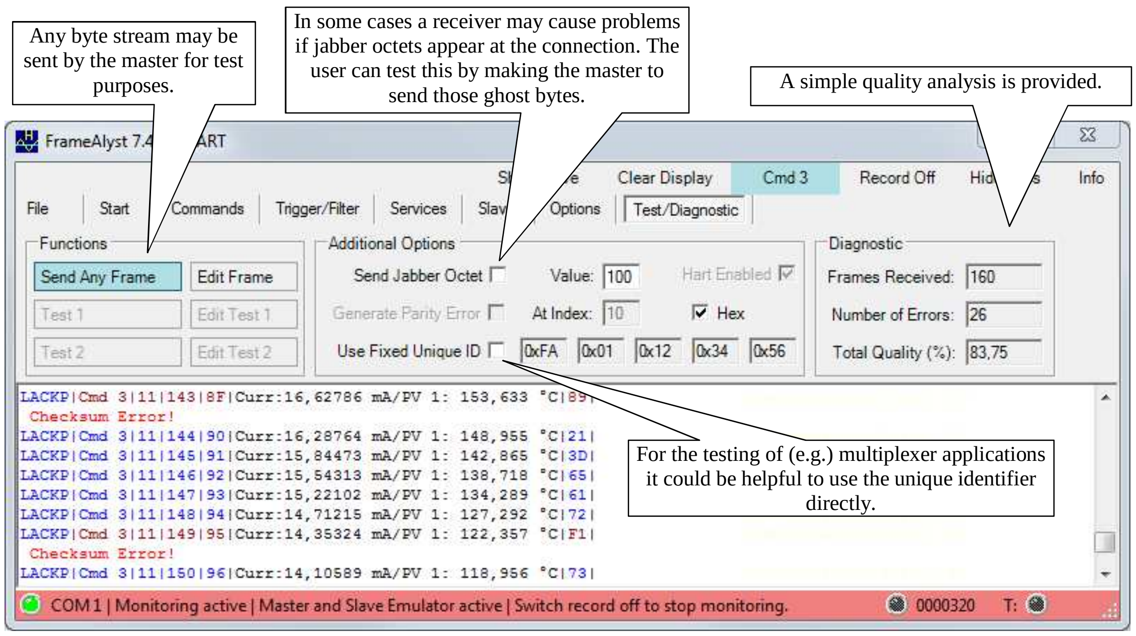
Master Timing Adjust: Additional STO: 0 ms; Carrier Off Delay: 0 ms; Add Gap Time: 0 ms

Monitoring: Record Jabber Octets ☐

```
 --->SSTXP|Cmd  0|  0|      |82|
   92<
 344>SACKP|Cmd  0|24|  0|00|254/Man250/                       1/                    0x56
                            MinPArsp:5/                       96
                            ManuID:0xFA                       le:1|05|
4294967293<
 7590>LSTXP|Cmd48|  0|      |79|
  129<
   62>LACKP|Cmd48|16|  0|01|[0]:00000001 [1]:00000010 [2]:00000100 [3]:00001000
                            [4]:00010000 [5]:00100000 [6]:01000000 [7]:10000000
                            [8]:10101010 [9]:01010101 [0]:11110000 [1]:00001111
                            [2]:11000011 [3]:00111100 |6C|
  270<
```

COM 1 | Monitoring active | Master and Slave Emulator active | Switch record off to stop monitorir 0000004 T:

Test/Diagnostic



Any byte stream may be sent by the master for test purposes.

In some cases a receiver may cause problems if jabber octets appear at the connection. The user can test this by making the master to send those ghost bytes.

A simple quality analysis is provided.

For the testing of (e.g.) multiplexer applications it could be helpful to use the unique identifier directly.

Additional Options: Send Jabber Octet ☐ Value: 100; Hart Enabled ✓; Generate Parity Error ☐ At Index: 10; Hex ✓; Use Fixed Unique ID ☐ 0xFA 0x01 0x12 0x34 0x56

Diagnostic: Frames Received: 160; Number of Errors: 26; Total Quality (%): 83,75

```
LACKP|Cmd 3|11|143|8F|Curr:16,62786 mA/PV 1: 153,633 °C|89|
 Checksum Error!
LACKP|Cmd 3|11|144|90|Curr:16,28764 mA/PV 1: 148,955 °C|21|
LACKP|Cmd 3|11|145|91|Curr:15,84473 mA/PV 1: 142,865 °C|3D|
LACKP|Cmd 3|11|146|92|Curr:15,54313 mA/PV 1: 138,718 °C|65|
LACKP|Cmd 3|11|147|93|Curr:15,22102 mA/PV 1: 134,289 °C|61|
LACKP|Cmd 3|11|148|94|Curr:14,71215 mA/PV 1: 127,292 °C|72|
LACKP|Cmd 3|11|149|95|Curr:14,35324 mA/PV 1: 122,357 °C|F1|
 Checksum Error!
LACKP|Cmd 3|11|150|96|Curr:14,10589 mA/PV 1: 118,956 °C|73|
```

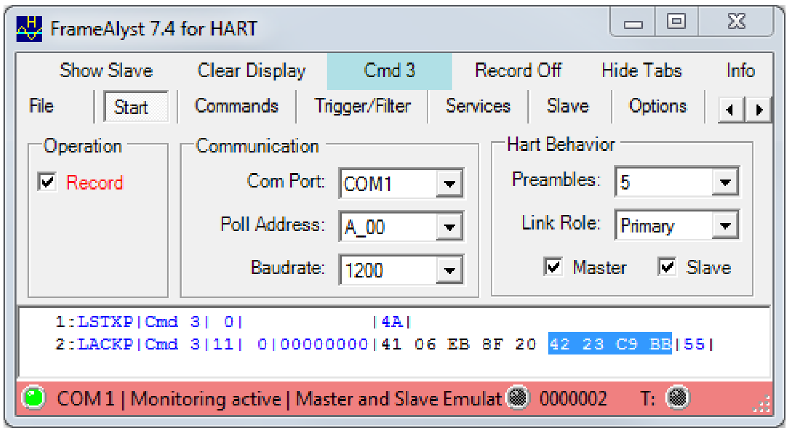
COM 1 | Monitoring active | Master and Slave Emulator active | Switch record off to stop monitoring. 0000320 T:

The above display was generated by using the filter for the suppression of requests and by error injection into the slave simulation.

Additional Details

Decoding Data in a Frame

For decoding data in a frame the data to be decoded has to be marked.



```
   1:LSTXP|Cmd  3|  0|                |4A|
   2:LACKP|Cmd  3|11|  0|00000000|41  06  EB  8F  20  42  23  C9  BB|55|
```

COM 1 | Monitoring active | Master and Slave Emulat 0000002 T:

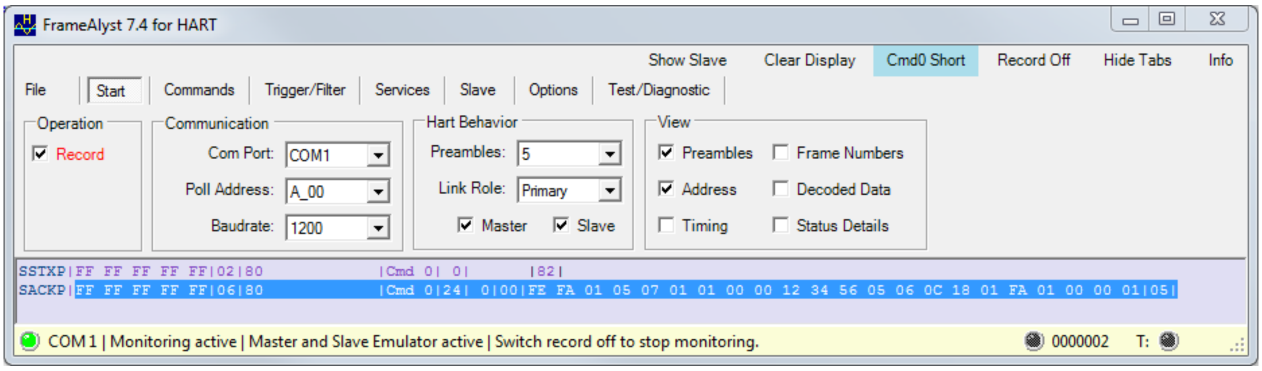
By using the right mouse button a context menu will be displayed.

```
Integer
Float
HartUnit
PackedASCII
Text
Binary
----------
Copy to AnyFrame
```

Select the decoding of your choice and the value will be displayed in a tool tip.

Copy to SendAnyFrame

Sometimes it could be helpful to copy a frame to the send any frame function to modify and send it.

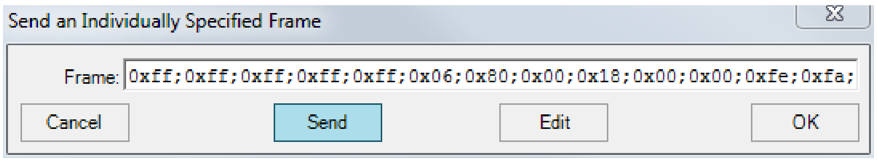


```
SSTXP|FF  FF  FF  FF  FF|02|80              |Cmd  0|  0|          |82|
SACKP|FF  FF  FF  FF  FF|06|80              |Cmd  0|24|  0|00|FE  FA  01  05  07  01  01  00  00  12  34  56  05  06  0C  18  01  FA  01  00  00  01|05|
```

COM 1 | Monitoring active | Master and Slave Emulator active | Switch record off to stop monitoring. 0000002 T:

Select the whole frame, click the right mouse button and click 'Copy to AnyFrame' in the context menu.

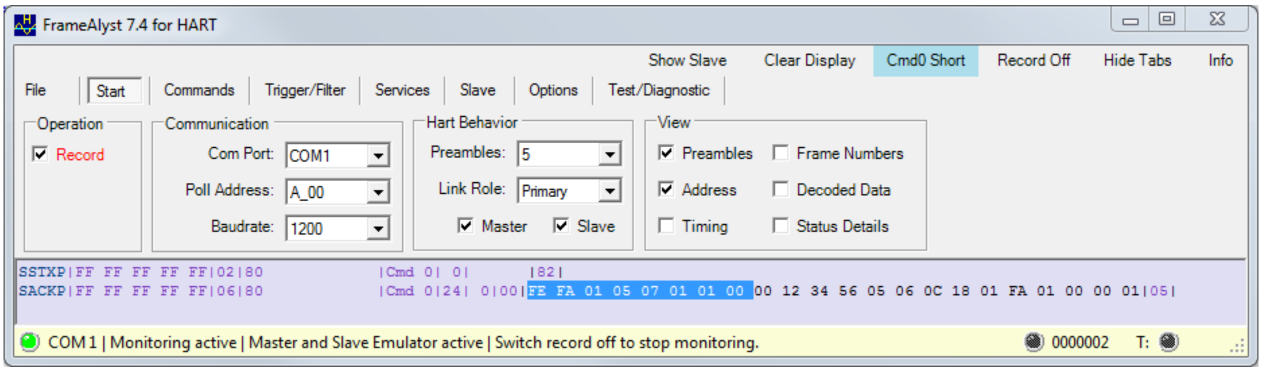
The data will be copied to this function and the edit any frame window will open.



Send an Individually Specified Frame

Frame: `0xff;0xff;0xff;0xff;0xff;0x06;0x80;0x00;0x18;0x00;0x00;0xfe;0xfa;`

It is also possible to copy only a part of the data.



FrameAlyst 7.4 for HART

Operation: Record ✓

Communication: Com Port: COM1; Poll Address: A_00; Baudrate: 1200

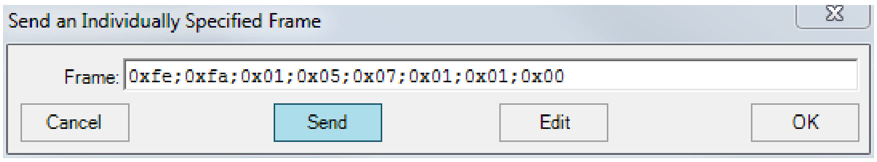
Hart Behavior: Preambles: 5; Link Role: Primary; Master ✓; Slave ✓

View: Preambles ✓; Frame Numbers ☐; Address ✓; Decoded Data ☐; Timing ☐; Status Details ☐

```
SSTXP|FF FF FF FF FF|02|80          |Cmd  0|  0|       |82|
SACKP|FF FF FF FF FF|06|80          |Cmd  0|24|  0|00|FE FA 01 05 07 01 01 00 00 12 34 56 05 06 0C 18 01 FA 01 00 00 01|05|
```

COM 1 | Monitoring active | Master and Slave Emulator active | Switch record off to stop monitoring.   0000002   T:

It will appear as is in the any frame editing function.



Send an Individually Specified Frame

Frame: `0xfe;0xfa;0x01;0x05;0x07;0x01;0x01;0x00`

Copy Bytes to the Clipboard

The same functionality as shown allows also to copy data bytes to the Windows clipboard by selecting 'Bytes to ClipBoard' in the context menu.

Editing Data Syntax

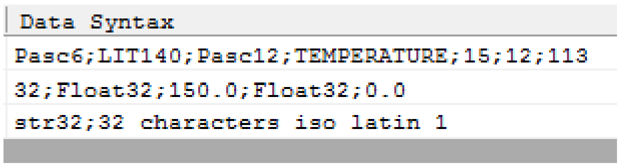
Data syntax allows to easily specify a stream of bytes to be send.

Prefix	Type	Example	Comment
None	Decimal or Hexadecimal	24; 0x18	The software will determine the required length
dec8, dec16, dec24, dec32	Decimal number	dec16; 1011	
bin8, bin16, bin24, bin32	Binary number	bin8; 10001101	
hex8, hex 16, hex24, hex32	Hexadecimal number	hex16; fa13	
float32	Single precision	float32; 1.34	
float64	Double precision	float64; 1.11e+48	

pca6, pca12, pca24	Packed ascii	pca6;LITT1400	pca6 = 8 characters
			pca12 = 16 characters
			pca24 = 32 characters

str8, str16, str32	Fixed length string	str32;my-device	Resulting byte array will be filled by 0s

All items the prefix and the data lement are separated by a colon ';'.

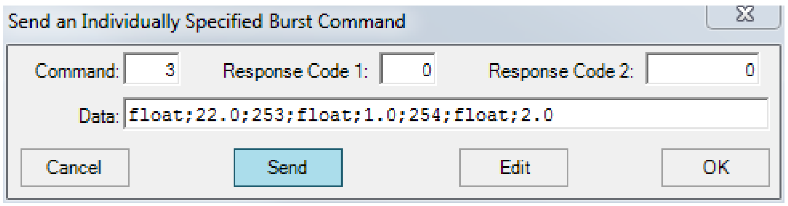


Data Syntax
Pasc6;LIT140;Pasc12;TEMPERATURE;15;12;113
32;Float32;150.0;Float32;0.0
str32;32 characters iso latin 1

A few examples are shown above

However, it could be much easier to do this by the data syntax editor.

When editing a command that requires data to be specified

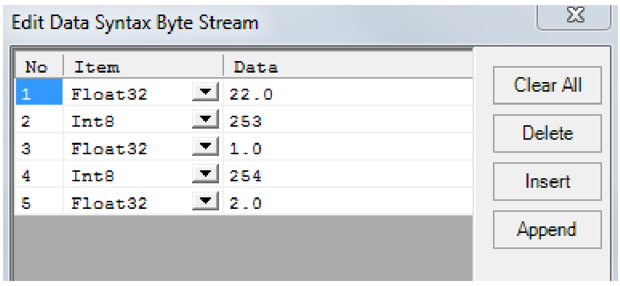


Send an Individually Specified Burst Command

Command: 3 Response Code 1: 0 Response Code 2: 0

Data: float;22.0;253;float;1.0;254;float;2.0

the data syntax editor will open on a click of the edit button.

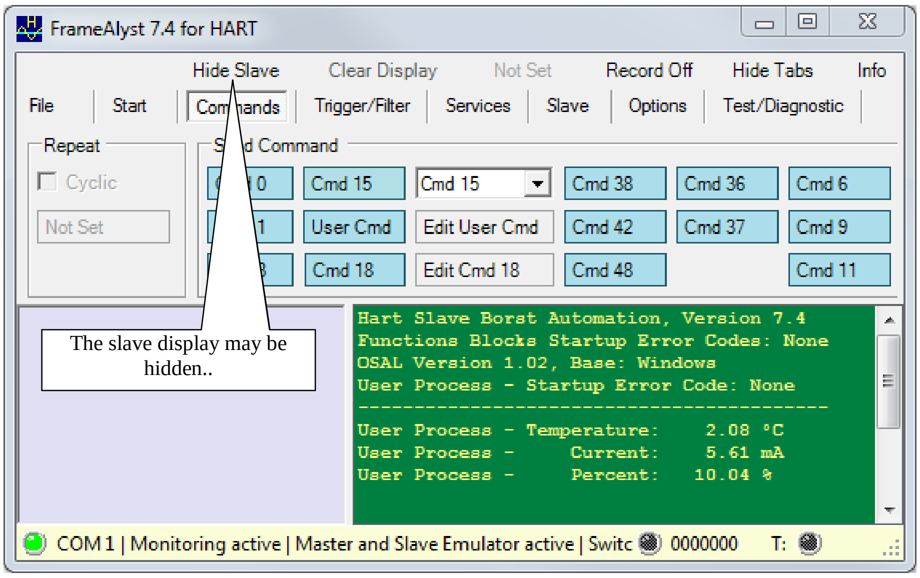


Edit Data Syntax Byte Stream

No	Item	Data
1	Float32	22.0
2	Int8	253
3	Float32	1.0
4	Int8	254
5	Float32	2.0

Displaying the Slave Emulation

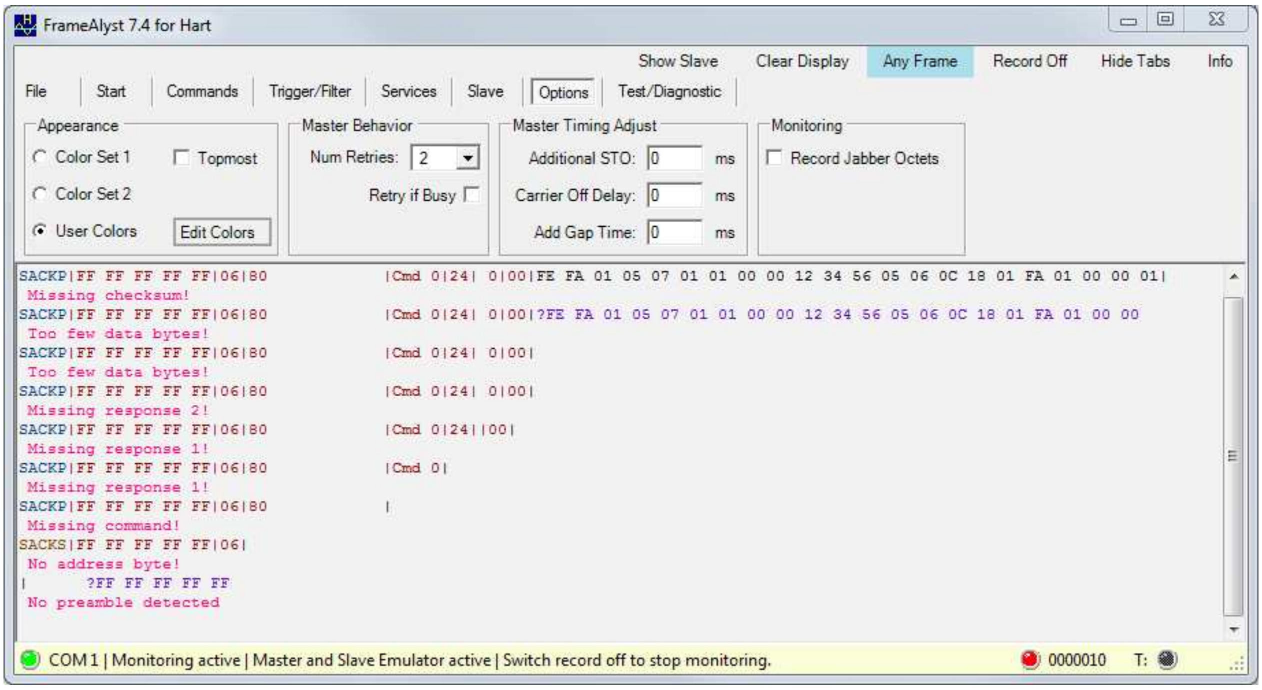
If the slave emulation is active, FrameAlyst provides a callback to the slave simulation which is used by this software for printing text with the printf function in the C libraries.



```
Hart Slave Borst Automation, Version 7.4
Functions Blocks Startup Error Codes: None
OSAL Version 1.02, Base: Windows
User Process - Startup Error Code: None
------------------------------------------
User Process - Temperature:    2.08 °C
User Process -     Current:    5.61 mA
User Process -     Percent:   10.04 %
```

COM 1 | Monitoring active | Master and Slave Emulator active | Switc 0000000 T:

Handling of Erroneous Frames



Color Set 1 | Topmost | Color Set 2 | [✓ User Colors]

Num Retries: 2 | Retry if Busy

Additional STO: 0 ms | Carrier Off Delay: 0 ms | Add Gap Time: 0 ms

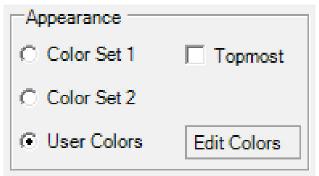
Record Jabber Octets

```
SACKP|FF FF FF FF FF|06|80              |Cmd 0|24| 0|00|FE FA 01 05 07 01 01 00 00 12 34 56 05 06 0C 18 01 FA 01 00 00 01|
 Missing checksum!
SACKP|FF FF FF FF FF|06|80              |Cmd 0|24| 0|00|?FE FA 01 05 07 01 01 00 00 12 34 56 05 06 0C 18 01 FA 01 00 00
 Too few data bytes!
SACKP|FF FF FF FF FF|06|80              |Cmd 0|24| 0|00|
 Too few data bytes!
SACKP|FF FF FF FF FF|06|80              |Cmd 0|24| 0|00|
 Missing response 2!
SACKP|FF FF FF FF FF|06|80              |Cmd 0|24||00|
 Missing response 1!
SACKP|FF FF FF FF FF|06|80              |Cmd 0|
 Missing response 1!
SACKP|FF FF FF FF FF|06|80              |
 Missing command!
SACKS|FF FF FF FF FF|06|
 No address byte!
|      ?FF FF FF FF FF
 No preamble detected
```

COM 1 | Monitoring active | Master and Slave Emulator active | Switch record off to stop monitoring. 0000010 T:

FrameAlyst is displaying the results while trying to read frames.

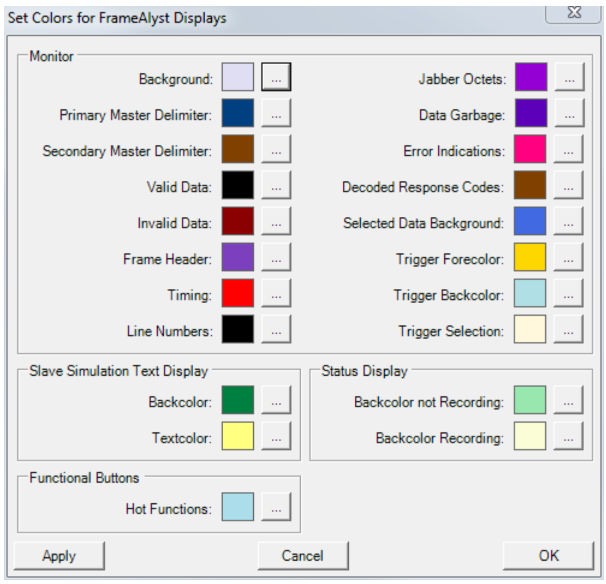
Setting Custom Colors


Appearance
Color Set 1 | Topmost
Color Set 2
[✓ User Colors] | Edit Colors

The tab Options is providing User Colors.

The user colors can be edited by clicking the button 'Edit Colors'.

The color editing form is shown in the following.


Set Colors for FrameAlyst Displays

Monitor

Background:	⬜	Jabber Octets:	🟪
Primary Master Delimiter:	🟦	Data Garbage:	🟪
Secondary Master Delimiter:	🟫	Error Indications:	🟥
Valid Data:	⬛	Decoded Response Codes:	🟫
Invalid Data:	🟥	Selected Data Background:	🟦
Frame Header:	🟪	Trigger Forecolor:	🟨
Timing:	🟥	Trigger Backcolor:	🟦
Line Numbers:	⬛	Trigger Selection:	⬜

Slave Simulation Text Display

Backcolor:	🟩
Textcolor:	🟨

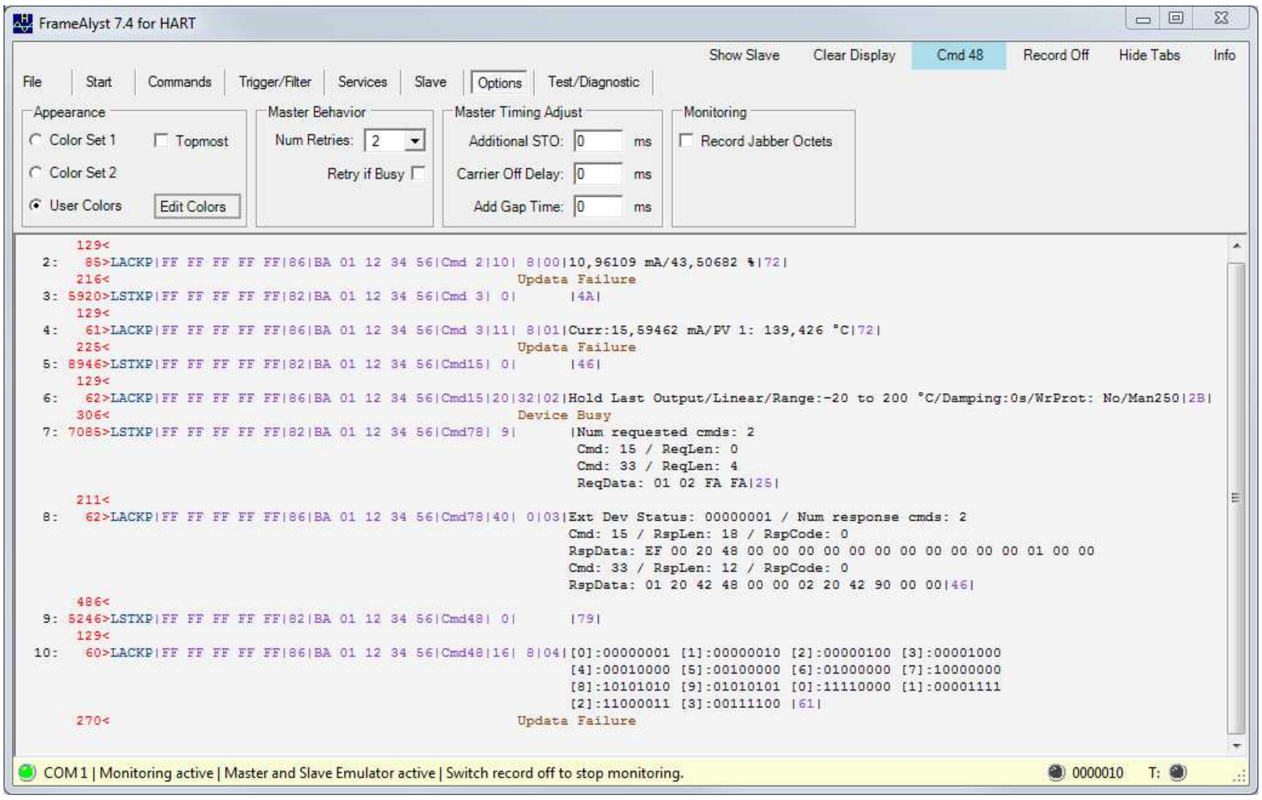
Status Display

Backcolor not Recording:	🟩
Backcolor Recording:	⬜

Functional Buttons

Hot Functions:	🟦

Frame Display Examples



Appearance: Color Set 1, Color Set 2, [✓ User Colors], Topmost, Edit Colors
Master Behavior: Num Retries: 2; Retry if Busy
Master Timing Adjust: Additional STO: 0 ms; Carrier Off Delay: 0 ms; Add Gap Time: 0 ms
Monitoring: Record Jabber Octets

```
     129<
 2:   85>LACKP|FF FF FF FF FF|86|BA 01 12 34 56|Cmd 2|10| 8|00|10,96109 mA/43,50682 %|72|
     216<                                             Updata Failure
 3: 5920>LSTXP|FF FF FF FF FF|82|BA 01 12 34 56|Cmd 3| 0|      |4A|
     129<
 4:   61>LACKP|FF FF FF FF FF|86|BA 01 12 34 56|Cmd 3|11| 8|01|Curr:15,59462 mA/PV 1: 139,426 °C|72|
     225<                                             Updata Failure
 5: 8946>LSTXP|FF FF FF FF FF|82|BA 01 12 34 56|Cmd15| 0|      |46|
     129<
 6:   62>LACKP|FF FF FF FF FF|86|BA 01 12 34 56|Cmd15|20|32|02|Hold Last Output/Linear/Range:-20 to 200 °C/Damping:0s/WrProt: No/Man250|2B|
     306<                                             Device Busy
 7: 7085>LSTXP|FF FF FF FF FF|82|BA 01 12 34 56|Cmd78| 9|      |Num requested cmds: 2
                                                               Cmd: 15 / ReqLen: 0
                                                               Cmd: 33 / ReqLen: 4
                                                               ReqData: 01 02 FA FA|25|

     211<
 8:   62>LACKP|FF FF FF FF FF|86|BA 01 12 34 56|Cmd78|40| 0|03|Ext Dev Status: 00000001 / Num response cmds: 2
                                                              Cmd: 15 / RspLen: 18 / RspCode: 0
                                                              RspData: EF 00 20 48 00 00 00 00 00 00 00 00 00 00 01 00 00
                                                              Cmd: 33 / RspLen: 12 / RspCode: 0
                                                              RspData: 01 20 42 48 00 00 02 20 42 90 00 00|46|

     486<
 9: 5246>LSTXP|FF FF FF FF FF|82|BA 01 12 34 56|Cmd48| 0|      |79|
     129<
10:   60>LACKP|FF FF FF FF FF|86|BA 01 12 34 56|Cmd48|16| 8|04|[0]:00000001 [1]:00000010 [2]:00000100 [3]:00001000
                                                              [4]:00010000 [5]:00100000 [6]:01000000 [7]:10000000
                                                              [8]:10101010 [9]:01010101 [0]:11110000 [1]:00001111
                                                              [2]:11000011 [3]:00111100 |61|
     270<                                             Updata Failure
```

COM 1 | Monitoring active | Master and Slave Emulator active | Switch record off to stop monitoring. 0000010 T:

FrameAlyst is decoding data of some commands.

Hart at a Glance

Frame Coding

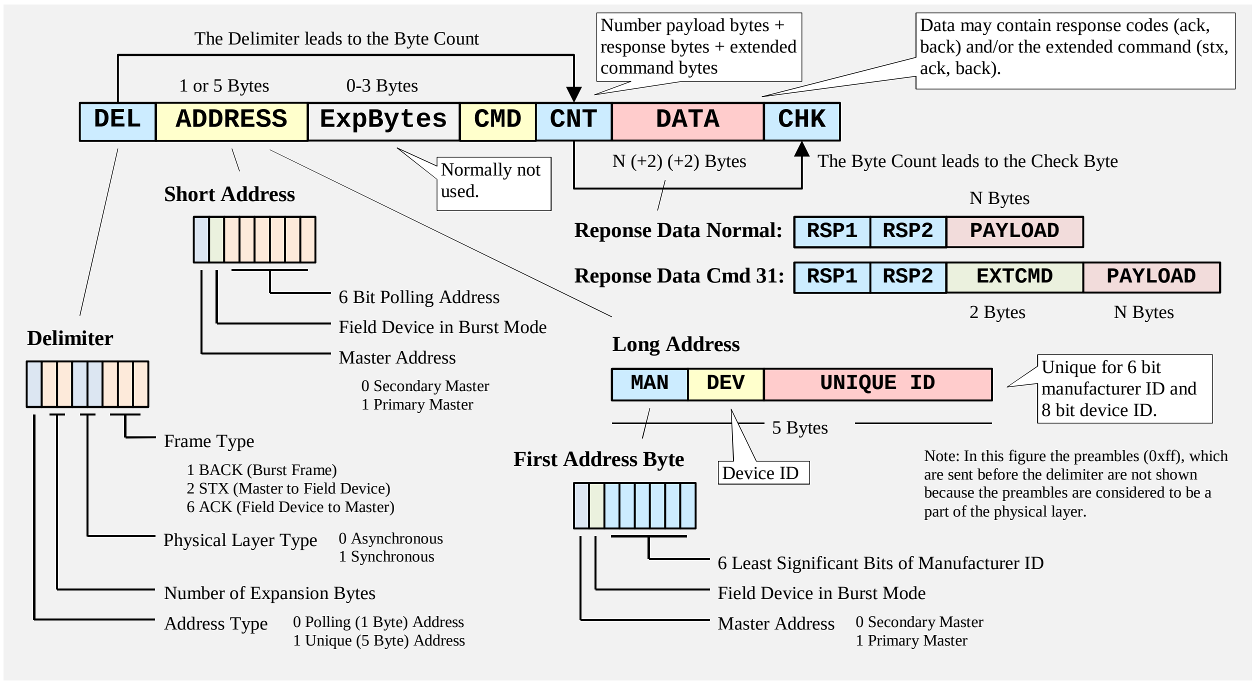

The Delimiter leads to the Byte Count
1 or 5 Bytes
0-3 Bytes
Number payload bytes + response bytes + extended command bytes
Data may contain response codes (ack, back) and/or the extended command (stx, ack, back).
DEL
ADDRESS
ExpBytes
CMD
CNT
DATA
CHK
Normally not used.
N (+2) (+2) Bytes
The Byte Count leads to the Check Byte
Short Address
N Bytes
Reponse Data Normal:
RSP1
RSP2
PAYLOAD
Reponse Data Cmd 31:
RSP1
RSP2
EXTCMD
PAYLOAD
6 Bit Polling Address
Field Device in Burst Mode
Master Address
0 Secondary Master
1 Primary Master
2 Bytes
N Bytes
Delimiter
Long Address
MAN
DEV
UNIQUE ID
Unique for 6 bit manufacturer ID and 8 bit device ID.
5 Bytes
Frame Type
1 BACK (Burst Frame)
2 STX (Master to Field Device)
6 ACK (Field Device to Master)
First Address Byte
Device ID
Note: In this figure the preambles (0xff), which are sent before the delimiter are not shown because the preambles are considered to be a part of the physical layer.
Physical Layer Type
0 Asynchronous
1 Synchronous
6 Least Significant Bits of Manufacturer ID
Number of Expansion Bytes
Field Device in Burst Mode
Address Type
0 Polling (1 Byte) Address
1 Unique (5 Byte) Address
Master Address
0 Secondary Master
1 Primary Master

Figure 2: The Basic Coding of a Hart Frame

The figure above is giving an overview of the coding of a Hart frame. Usually Hart services are composed of a request (stx) by the master followed the response (ack) of a slave. Bursts (back) are frames looking like a response (including response codes) but sent by the slave without any request. The slave is sending these frames in burst mode within defined time slots following the rules of the protocol specification. In fact Hart is a token passing protocol which allows also the slave to be a token holder and send burst frames.

The following chapter is showing a list of Hart commands which are used very often. The list is showing the major differences between Hart 5.3, Hart 6 and Hart 7.4.

New items in Hart 6 are marked with yellow color while new items of Hart 7.4 are marked by blue color.

However, the following is not replacing any specification and is not showing the details which are needed for an implementation. The details has to be taken from the Hart specifications which are provided by the Hart Communication Foundation (http://de.hartcomm.org/).
That the listed commands are most commonly used is not the opinion of the HCF but the opinion of the author of this document.

Commonly Used Commands

No	Title	Request Data			Response Data		
Universal							
00	**Read Unique Identifier**	None			0	int8	254
					1		Manufacturer ID
					2		Short device ID
					3		Number preambles request
					4		Hart revision
					5		Device revision
					6		Software revision
					7		Hw rev and signaling code
					8		Flags
					9	int24	DevUniqueID
					12	int8	Number preambles response
					13		Maximum number device variables
					14	int16	Configuration change counter
					16	int8	Extended device status
					17	int16	Extended manufacturer code
					19		Extended label distributor code
					21	int8	Device profile
01	**Read Primary Variable**	None			0	int8	PV Units
					1	float	Primary variable
02	**Read Current and Percent of Range**	None			0	float	Current
					1	float	Percent of range
03	**Read Current and Dyn. Variables**	None			0	float	Current
					4	int8	PV1 units code
					5	float	PV1 value
					9	int8	PV2 units code
					10	float	PV2 value
					14	int8	PV3 units code
					15	float	PV3 value
					19	int8	PV4 units code
					20	float	PV4 value
06	**Write Polling Address**	0	int8	Polling Address	0	int8	PV Units
		1	int8	Loop current mode	1	int8	Loop current mode
07	**Read Loop Configuration**	None			0	int8	Polling address
					1		Loop current mode
08	**Read Dyn. Vars Classification**	None			0	int8	PV1 classification
					1		PV2 classification
					2		PV3 classification
					3		PV4 classification

No	Title	Request Data			Response Data		
Universal							
09	**Read Device Variables with Status**	0	int8	Slot0: Device variable code	0	int8	Extended device status
		1		Slot1: Device variable code	1		Slot0: Device variable properties
		2		Slot2: Device variable code	1	int8	Device variable code
		3		Slot3: Device variable code	2		Device variable classification
		4	int8	Slot4: Device variable code	3		Device variable units code
		5		Slot5: Device variable code	4	float	Device variable value
		6		Slot6: Device variable code	8	int8	Device variable status
		7		Slot7: Device variable code	9	struct	Slot1: Device variable properties
					17		Slot2: Device variable properties
					25		Slot3: Device variable properties
					33	struct	Slot4: Device variable properties
					41		Slot5: Device variable properties
					49		Slot6: Device variable properties
					57		Slot7: Device variable properties
					65	time	Time stamp slot0
11	**Read Unique ID by Short Tag**	0	pac6	Tag name (packed ascii) 6 bytes = 8 characters	Same as command 0 read unique identifier		
12	**Read Message**	None			0	pac24	Message (packed ascii) 24 bytes = 32 characters
13	**Read Tag, Descriptor, Date**	None			0	pac6	Short tag (packed ascii) 6 bytes = 8 characters
					6	pac12	Descriptor (packed ascii) 12 bytes = 16 characters
					18	int8	Day
					19		Month
					20		Year (offset to 1900)
14	**Read Primary Variable Transducer Information**	None			0	int24	Transducer serial number
					3	int8	Units code
					4	float	Upper transducer limit
					8		Lower transducer limit
					12		Minimum span
15	**Read Device Information**	None			0	int8	Alarm selection code
					1		Transfer function code
					2		Units code
					3	float	PV upper range value (for 20 mA)
					7		PV lower range value (for 4 mA)
					11		PV damping value
					15	int8	Write protect code
					16		Reserved, must be set to 250
					17		PV analog channel flags
16	**Read Ass. Num**	None			0	int24	Final assembly number
17	**Write Message**	Same as response command 12			Same as response command 12		
18	**Write Tag, Descriptor, Date**	Same as response command 13			Same as response command 13		
19	**Write Ass. Num**	Same as response command 16			Same as response command 16		
20	**Read Long Tag**	None			0	str32	Long tag: 32 ISO Latin-1 characters
21	**Read Unique ID by Long Tag**	0	str32	Long tag: 32 ISO Latin-1 characters	Same as command 0 read unique identifier		
22	**Write Long Tag**	Same as response command 20			Same as response command 20		

No	Title	Request Data			Response Data		
Universal / Common Practice							
38	**Reset Config Changed Flag**	None			None		
		0	int16	Configuration change counter	0	int16	Configuration change counter
48	**Read Additional Device Status**	None					
		0	int8[5]	Transmitter specific status	0	int8[5]	Transmitter specific status
					6	int8[2]	Operating mode
		6	int8	Extended device status	6	int8	Extended device status
		7		Device operating mode	7		Device operating mode
					8	int8[3]	Analog output status
		8	int8	Standard status 0	8	int8	Standard status 0
		9		Standard status 1	9		Standard status 1
		10		Analog channel saturated	10		Analog channel saturated
					11	int8[3]	Analog output fixed
		11	int8	Standard status 2	11	int8	Standard status 2
		12		Standard status 3	12		Standard status 3
		13		Analog channel fixed	13		Analog channel fixed
					14	int8[3]	Transmitter specific status
		14	int8[10]	Transmitter specific status	14	int8[10]	Transmitter specific status
Common Practice							
33	**Read Device Variables**	0	int8	Slot0: Device variable code	0	Slot0: Device variable properties	
		1		Slot1: Device variable code	0	int8	Device variable code
		2		Slot2: Device variable code	1		Device variable units code
		3		Slot3: Device variable code	2	float	Device variable value
					6	struct	Slot1: Device variable properties
					12		Slot2: Device variable properties
					18		Slot3: Device variable properties
34	**Write Prim. Var. Damping**	0	float	PV 1 damping value	0	float	PV 1 damping value
35	**Write Prim. Var. Range Values**	0	int8	Units code	0	int8	Units code
		1	float	Upper range value	1	float	Upper range value
		5		Lower range value	5		Lower range value
36	**Set Prim. Var. Upper Range**	None			None		
37	**Set Prim. Var. Lower Range**	None			None		
40	**Enter/Exit Fixed Current**	0	float	Current value	0	float	Actual current value
42	**Device Reset**	None			None		
43	**Set Primary Variable Zero**	None			None		
44	**Write Prim. Var. Units**	0	int8	PV 1 units code	0	int8	PV 1 units code
45	**Trim Prim. Var. Current Zero**	0	float	Measured current value	0	float	Actual current value
46	**Trim Prim. Var. Current Gain**	0	float	Measured current value	0	float	Actual current value
50	**Read Dynamic Variable Assignments**	None			0	int8	PV 1 variable code
					1		PV 2 variable code
					2		PV 3 variable code
					3		PV 4 variable code

No	Title	Request Data			Response Data		
Common Practice							
51	**Write Dynamic Variable Assignments**	0	int8	PV 1 variable code	0	int8	PV 1 variable code
		1		PV 2 variable code	1		PV 2 variable code
		2		PV 3 variable code	2		PV 3 variable code
		3		PV 4 variable code	3		PV 4 variable code
54	**Read Device Variable Information**	0	int8	Device variable code	0	int8	Device variable code
					1	int24	Sensor serial number
					4	int8	Units code
					5	float	Variable upper limit
					9		Variable lower limit
					13		Variable damping
					17		Variable minimum span
					21	int8	Variable classification
					22		Variable family
					23	time	Acquisition period
					27	bin8	Variable properties
71	**Lock Device**	0	int8	Lock code	0	int8	Lock code
76	**Read Lock State**	None			0	int8	Lock status
78	**Read Aggregated Commands**	0	int8	Number of commands requested	0	int8	Extended device status
		1	str[]	Array of command requests struct { int16 command int8 byteCount int8[] requestData }	1	int8	Number of commands requested
					2	str[]	Array of command responses struct { int16 command int8 byteCount int8 responseCode int8[] responseData }
79[1]	**Write Device Variable**	0	int8	Device Variable Code	0	int8	Device Variable Code
		1		DV command code	1		DV command code
		2		DV units code	2		DV units code
		3	float	DV value	3	float	DV value
		7	int8	DV status	7	int8	DV status
103	**Write Burst Period**	0	int8	Burst message	0	int8	Burst message
		1	time	Update period	1	time	Update period
		5		Maximum update period	5		Maximum update period
104	**Write Burst Trigger**	0	int8	Burst message	0	int8	Burst message
		1		Trigger mode selection code	1		Trigger mode selection code
		2		Device variable classification for trigger level	2		Device variable classification for trigger level
		3		Units code	3		Units code
		4	float	Trigger level	4	float	Trigger level

[1] Used to simulate the value of a device variable

No	Title	Request Data			Response Data		
Common Practice							
105	**Read Burst Mode Configuration**	None			0	int8	Burst mode control code
					1	int8	Burst command number
					2	int8	Burst command slot 0
					3	int8	Burst command slot 1
					4	int8	Burst command slot 2
					5	int8	Burst command slot 3
		0	int8	Burst message	0	int8	Burst mode control code
					1		0x1f (31) command expansion
					2		DV code slot0
					3		DV code slot1
					4		DV code slot2
					5		DV code slot3
					6		DV code slot4
					7		DV code slot5
					8		DV code slot6
					9		DV code slot7
					10		Burst message
					11		Maximum number of burst messages
					12	int16	Extended command number
					14	time	Update time
					18		Maximum update time
					22	int8	Burst trigger mode code
					23		DV classification for trigger value
					24		Units code
					25	float	trigger value
106	**Flush Delayed Responses**	None			None		
107	**Write Burst Device Variables**	0	int8	DV code slot 0	0	int8	DV code slot 0
		1		DV code slot 1	1		DV code slot 1
		2		DV code slot 2	2		DV code slot 2
		3		DV code slot 3	3		DV code slot 3
		4	int8	DV code slot 4	4	int8	DV code slot 4
		5		DV code slot 5	5		DV code slot 5
		6		DV code slot 6	6		DV code slot 6
		7		DV code slot 7	7		DV code slot 7
		8		Burst message	8		Burst message
108	**Write Burst Mode Command**	0	int8	Command number for the burst response	0	int8	Command number of the burst response
109	**Burst Mode Control**	0	int8	Burst mode control code	0	int8	Burst mode control code
113	**Catch Device Variable**	0	int8	Destination DV code	0	int8	Destination DV code
		1		Capture mode code	1		Capture mode code
		2		Source slave manufacturer ID	2	int8[5]	Source slave address
		3		Source slave device type			
		2	int16	Source slave expanded device type			
		4	int8[3]	Source slave device ID			
		7	int8	Source command number	7	int8	Source command number
		8		Source slot number	8		Source slot number
		9	float	Shed time for this mapping	9	float	Shed time for this mapping
		7	int8	0x1f (31) command expansion	7	int8	0x1f (31) command expansion
		8		Source slot number	8		Source slot number
		9	float	Shed time for this mapping	9	float	Shed time for this mapping
		13	int16	Ext source command number	13	int16	Ext source command number

No	Title	Request Data			Response Data		
Common Practice							
114	**Read Caught Device Variable**	0	int8	Destination DV code	0	int8	Destination DV code
					1		Capture mode code
					2	int8[5]	Source slave address
					7	int8	Source command number
					8		Source slot number
					9	float	Shed time for this mapping
					7	int8	0x1f (31) command expansion
					8		Source slot number
					9	float	Shed time for this mapping
					13	int16	Ext source command number
523	**Read Condensed Status Mapping Array**	0	int8	Starting index status map	0	int8	Actual starting index
		1		Number of entries to read	1		Number of entries returned
					2	int4[]	Status map codes array
524	**Write Condensed Status Mapping Array**	0	int8	Starting index status map	0	int8	Actual starting index
		1		Number of entries to write	1		Number of entries returned
		2	int4[]	Status map codes array	2	int4[]	Status map codes array
525	**Reset Condensed Status Map**	None			None		
526	**Write Status Simulation Mode**	0	int8	Status simulation mode	0	int8	Status simulation mode
527	**Simulate Status Bit**	0	int8	Status bit index	0	int8	Status bit index
		1		Status bit value	1		Status bit value

Device Status

As response code 1 is command specific it is documented together with the command specifications. However response code 2 is of general nature and contains 8 bit flags with the following meaning.

Flag Number / Meaning	Description
Bit #7 Field Device Malfunction	The device has detected a hardware error or failure. Further information may be available through the Read Additional Transmitter Status Command, #48.
Bit #6 Configuration Changed	A write or set command has been executed.
Bit #5 Cold Start	Power has been removed and reapplied resulting in the reinstallations of the setup information. The first command to recognize this condition will automatically reset this flag. This flag may also be set following a Master Reset or a Self Test.
Bit #4 More Status Available	More status information is available than can be returned in the Field Device Status. Command #48, Read Additional Status Information, will provide this additional status information.
Bit #3 Primary Variable Analog Output Fixed	The analog and digital analog outputs for the Primary Variable are held at the requested value. They will not respond to the applied process.
Bit #2 Primary Variable Analog Output Saturated	The analog and digital analog outputs for the Primary Variable are beyond their limits and no longer represent the true applied process.
Bit #1 Non Primary Variable Out of Limits	The process applied to a sensor, other than that of the Primary Variable, is beyond the operating limits of the device. The Read Additional Transmitter Status Command, #48, may be required to identify the variable.
Bit #0 Primary Variable Out of Limits	The process applied to the sensor for the Primary Variable is beyond the operating limits of the device.

Appendix

FrameAlyst Versions

Function	Standard	Developer
Recording of HART frames	•	•
Save/Load recorded data	•	•
Print recorded data	•	•
Decode data	•	•
Standard commands	•	•
Configuration of display colors	•	•
Standard services	•	•
Send any command (user command)	•	•
Edit data syntax	•	•
Slave emulation		•
Send any frame		•
Send burst command		•
Edit and run scripts		•
Trigger functions		•
Filter functions		•
Send extended command		•