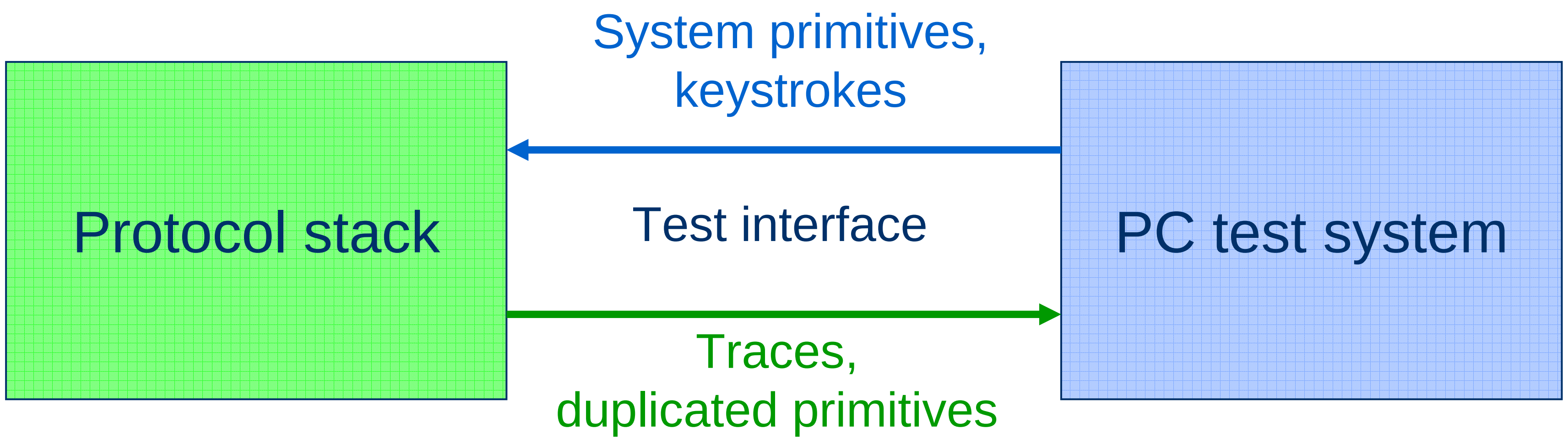
PCO

general concept

- Test interface approach:
  - ⇨ data interface between G23 protocol stack and a PC test system
  - ⇨ usually a standard serial cable, COM-ports on both ends



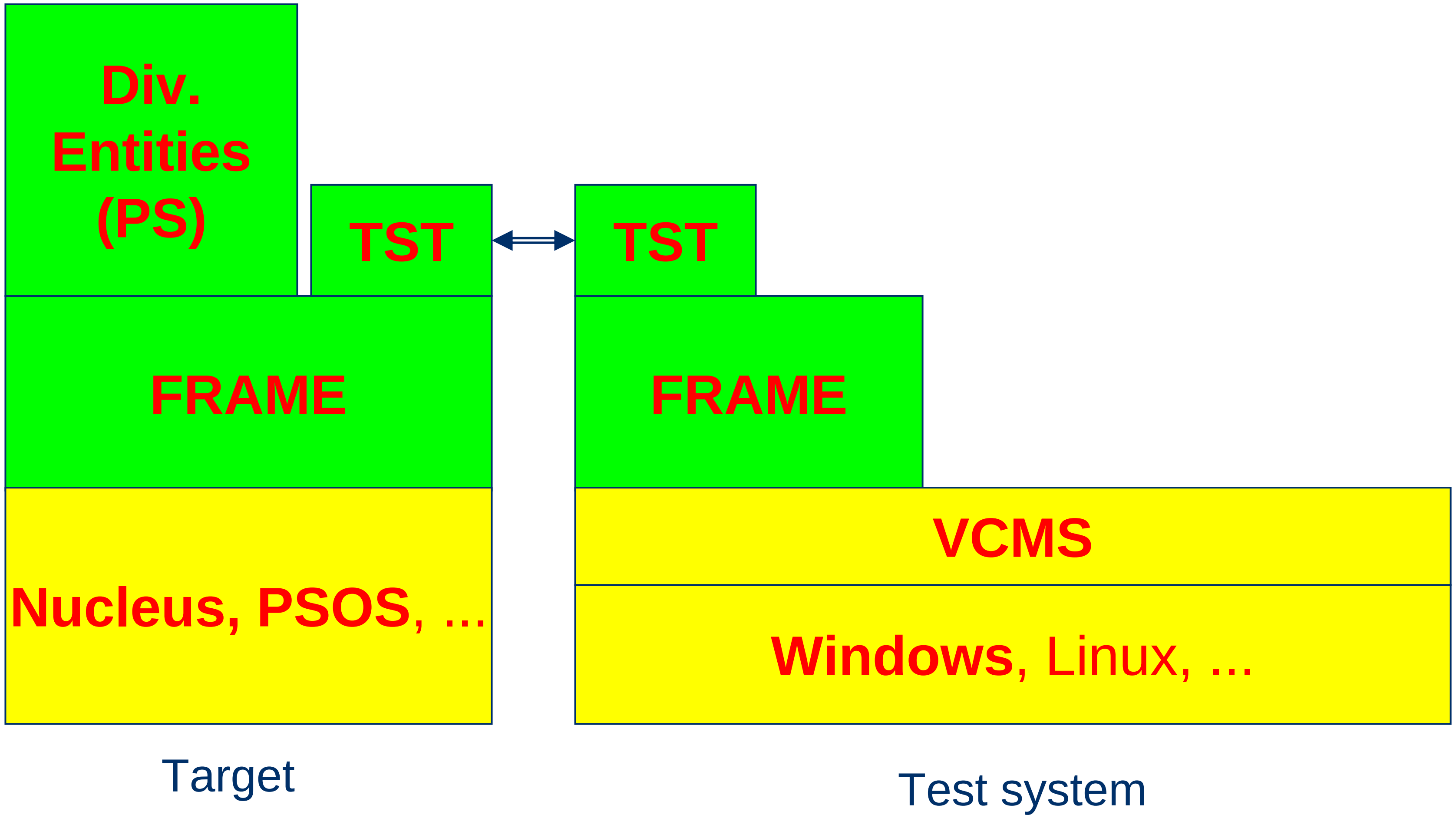
general concept

- On stack side:
  - ⇨ test interface entity included in the GPF-FRAME
  - ⇨ uses corresponding hardware driver for communication
- On PC test system side:
  - ⇨ test interface executable using the GPF-FRAME
    - ◆ connects via standard OS drivers
  - ⇨ PCO tools finally provide GUI-stack-access for testers

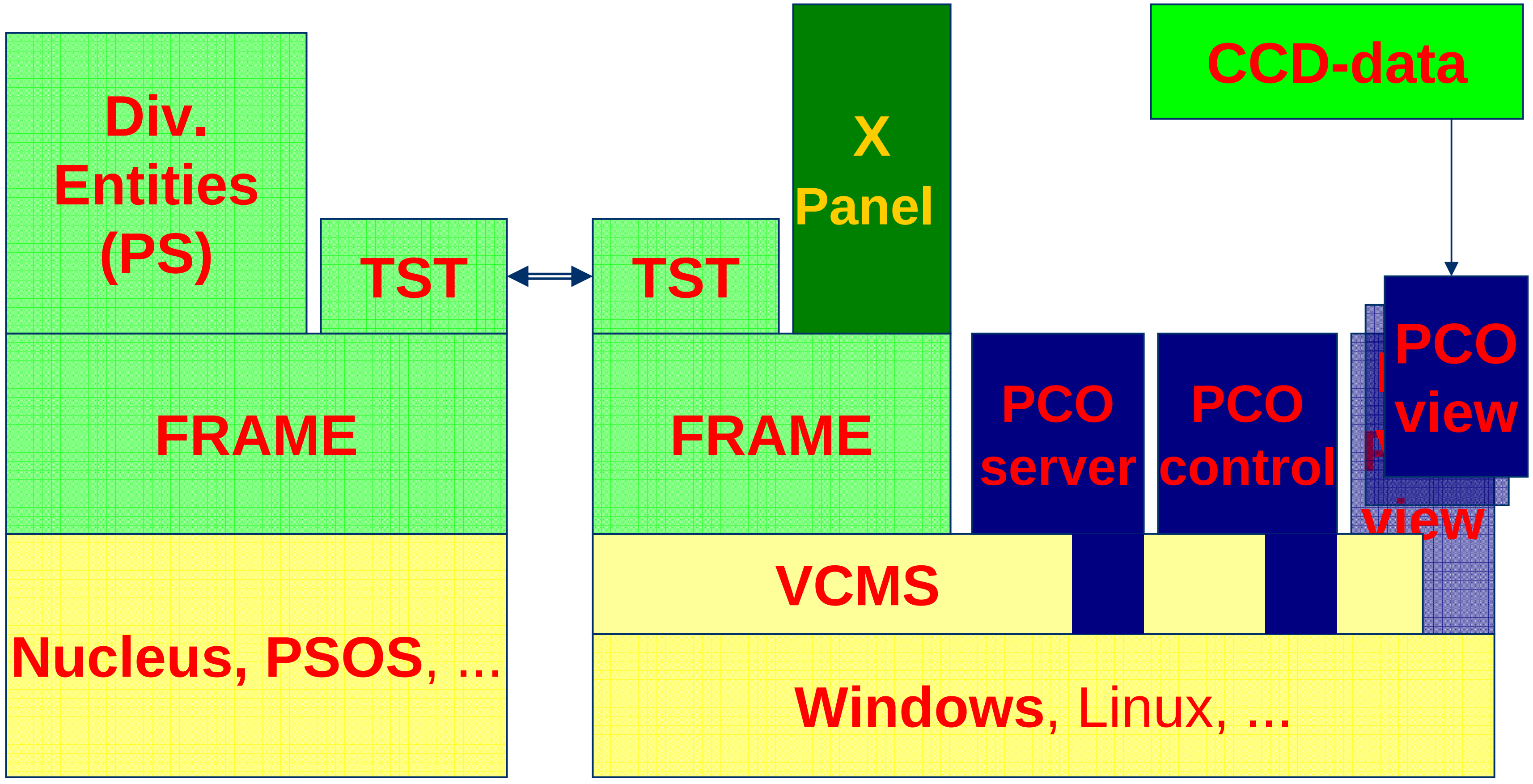
general concept

- xPanel - eXtended Panel:
  - ⇨ capable to display text & graphics output of mobile MMI
  - ⇨ mutable layout, easy to change
- PCO2 - Point of Control and Observation:
  - ⇨ filtered watching of traces and duplicated primitives
  - ⇨ intuitive configuration (traceclasses, routing) of protocol stack
  - ⇨ server, controller, extensible set of viewers

software layers



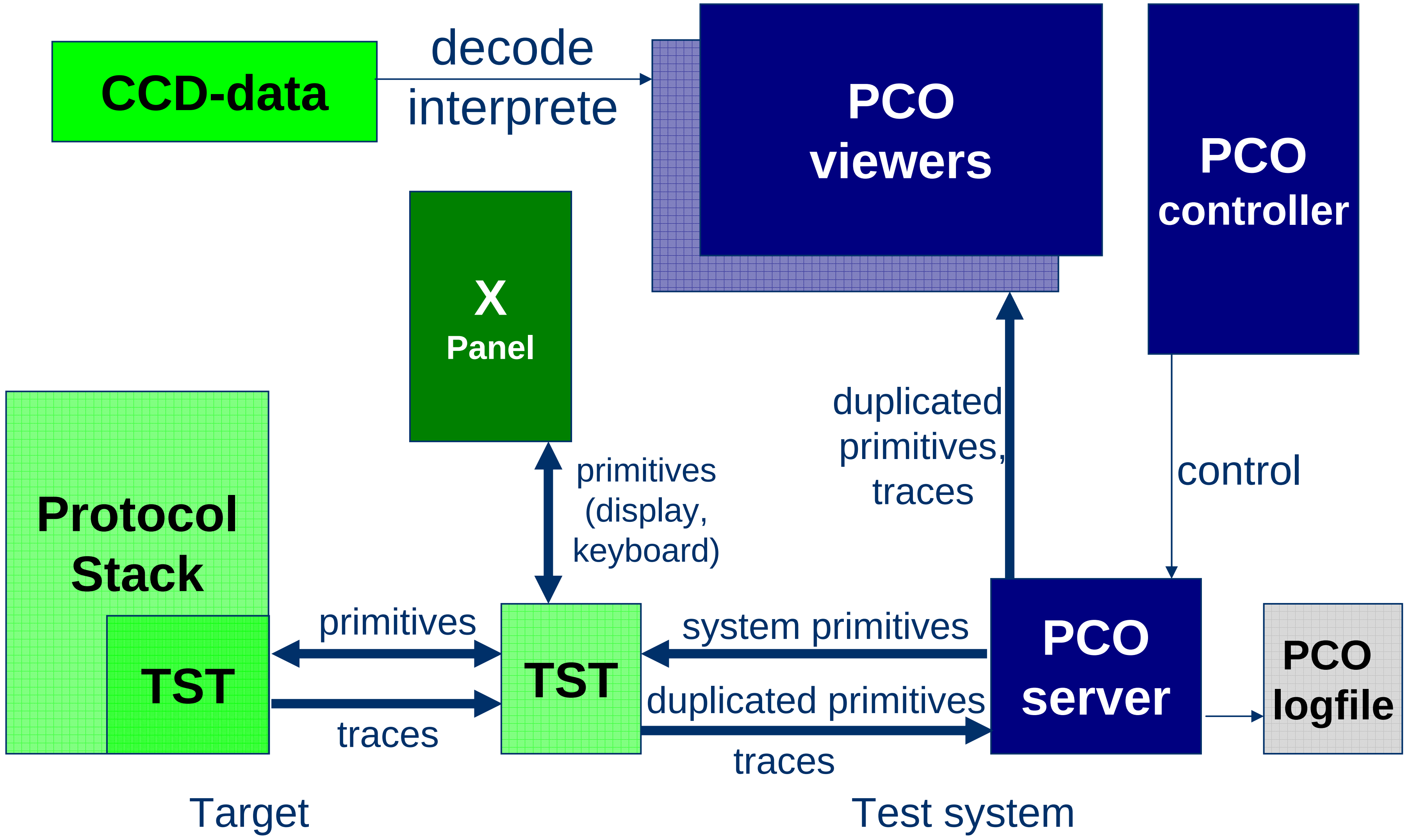
software layers



Target

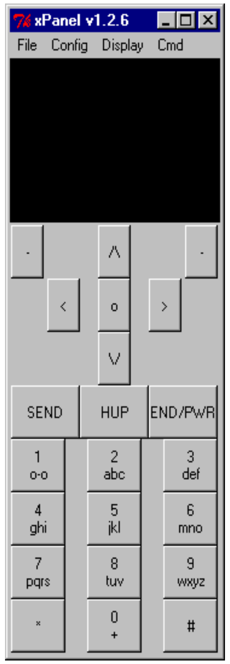
Test system

data flow

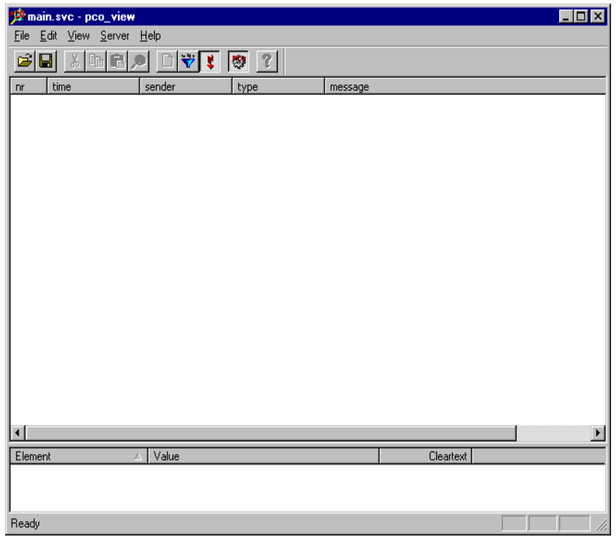

CCD-data
decode
interprete
PCO viewers
PCO controller
X Panel
primitives (display, keyboard)
duplicated primitives, traces
control
Protocol Stack
TST
primitives
traces
TST
system primitives
duplicated primitives
traces
PCO server
PCO logfile
Target
Test system

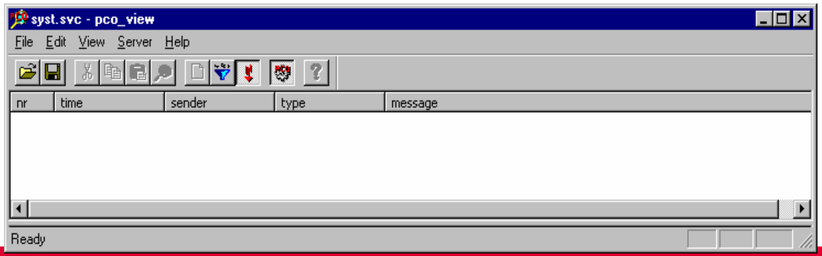
usage overview

- "Take off":
  - ⇨ starting of pco2.bat in the bin-directory (ClearCase: \gpf\BIN) …
  - ⇨ will per default result in such a scenario:

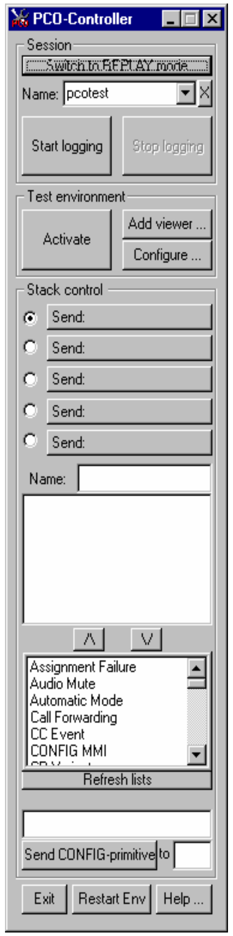


xPanel

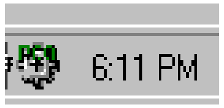




PCO-Viewers



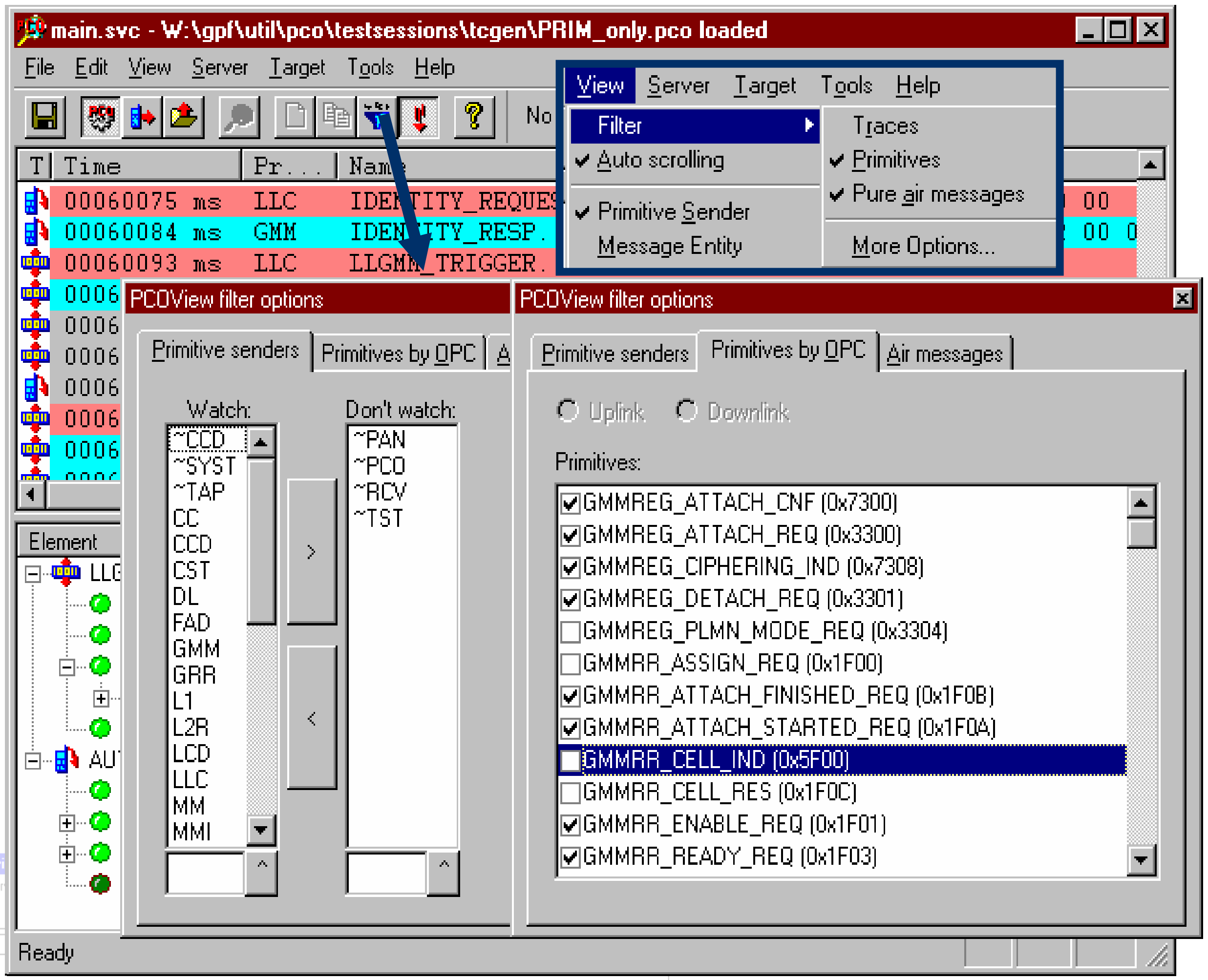
PCO-Controller



PCO-Server

usage overview

- The PCO-Viewer(s):
  - ⇨ watch traces of selected entities
    - ordered by time
    - distinguished by colors
  - ⇨ watch redirected primitives/messages
    - as hexdump
    - as structure
  - ⇨ filter by sender or OPC



main.svc - W:\gpf\util\pco\testsessions\tcgen\PRIM_only.pco loaded

File Edit View Server Target Tools Help

```
T  Time              Pr...  Name
   00060075 ms       LLC    IDENTITY_REQUES
   00060084 ms       GMM    IDENTITY_RESP
   00060093 ms       LLC    LLGMM_TRIGGER
   0006
   0006
   0006
   0006
   0006
   0006
```

Element: LLC, AU

Ready

View Server Target Tools Help

- Filter ▸ Traces / ✓ Primitives / ✓ Pure air messages / More Options...
- ✓ Auto scrolling
- ✓ Primitive Sender
- Message Entity

PCOView filter options — Primitive senders | Primitives by OPC | Air messages

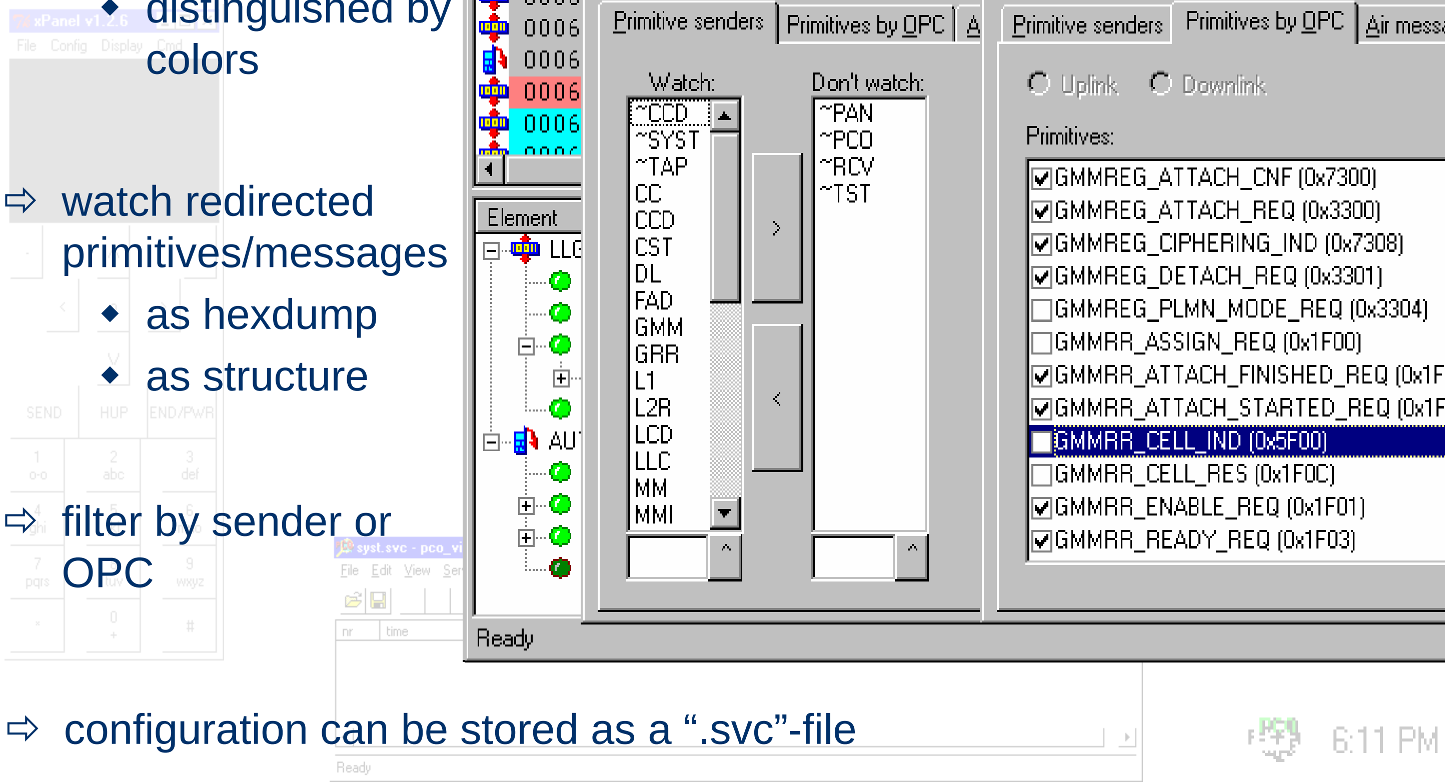
Watch:	Don't watch:
~CCD	~PAN
~SYST	~PCO
~TAP	~RCV
CC	~TST
CCD	
CST	
DL	
FAD	
GMM	
GRR	
L1	
L2R	
LCD	
LLC	
MM	
MMI	

PCOView filter options — Primitive senders | Primitives by OPC | Air messages

Uplink Downlink

Primitives:

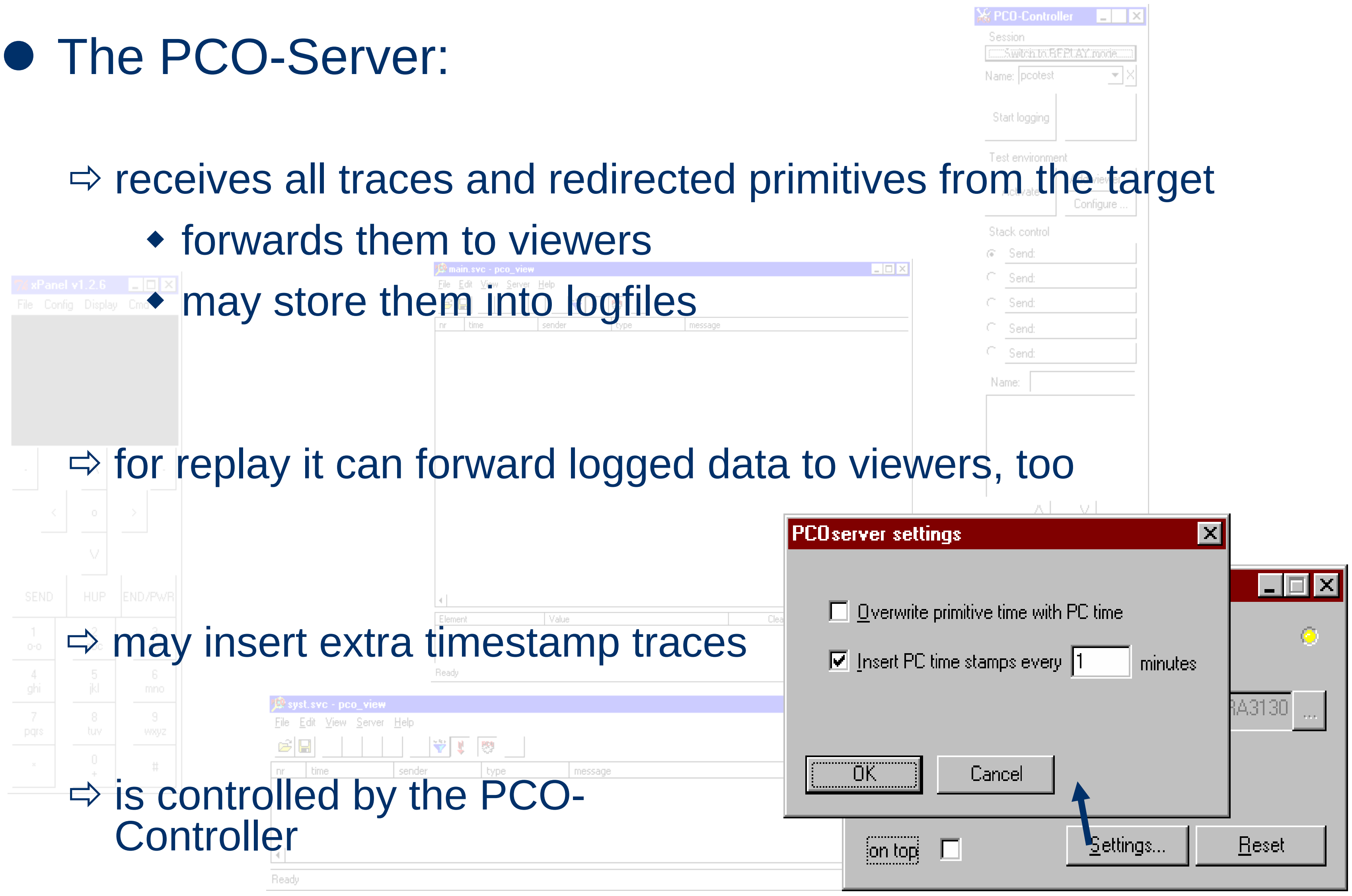
- [x] GMMREG_ATTACH_CNF (0x7300)
- [x] GMMREG_ATTACH_REQ (0x3300)
- [x] GMMREG_CIPHERING_IND (0x7308)
- [x] GMMREG_DETACH_REQ (0x3301)
- [ ] GMMREG_PLMN_MODE_REQ (0x3304)
- [ ] GMMRR_ASSIGN_REQ (0x1F00)
- [x] GMMRR_ATTACH_FINISHED_REQ (0x1F0B)
- [x] GMMRR_ATTACH_STARTED_REQ (0x1F0A)
- [ ] GMMRR_CELL_IND (0x5F00)
- [ ] GMMRR_CELL_RES (0x1F0C)
- [x] GMMRR_ENABLE_REQ (0x1F01)
- [x] GMMRR_READY_REQ (0x1F03)



- ⇨ configuration can be stored as a “.svc”-file

usage overview

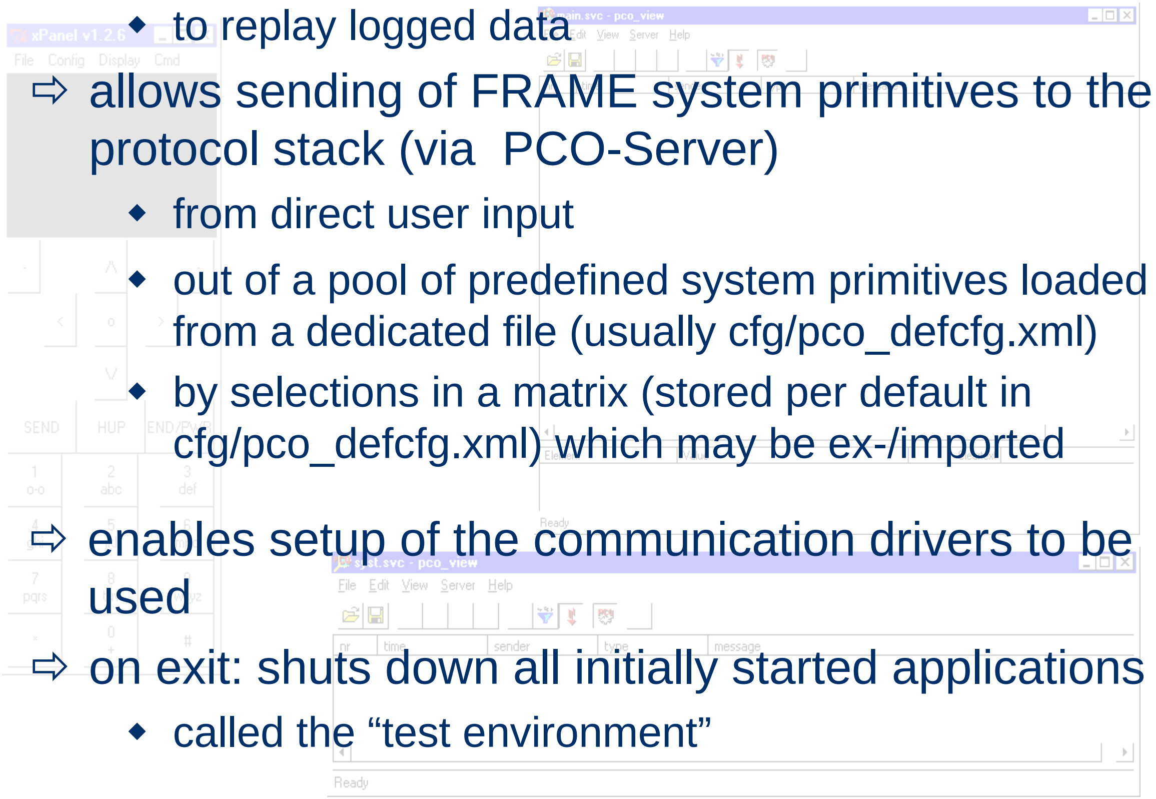
- The PCO-Server:
  - ⇨ receives all traces and redirected primitives from the target
    - ◆ forwards them to viewers
    - ◆ may store them into logfiles
  - ⇨ for replay it can forward logged data to viewers, too
  - ⇨ may insert extra timestamp traces
  - ⇨ is controlled by the PCO-Controller

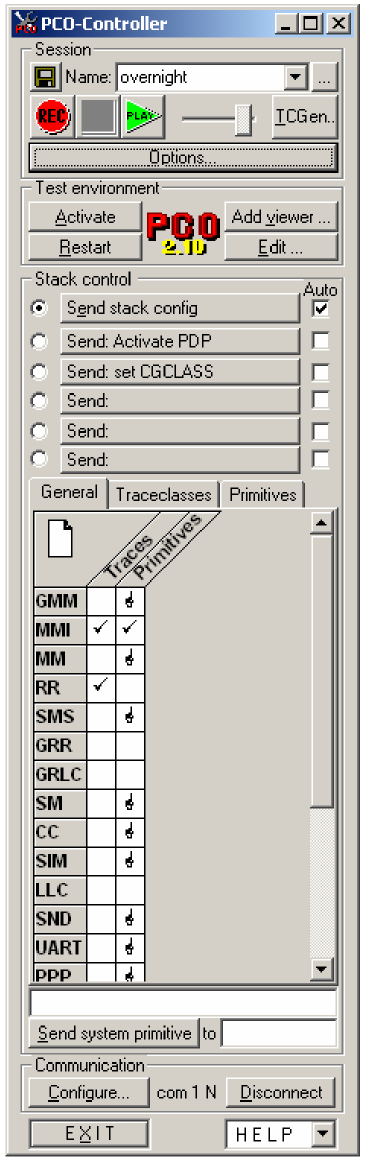

PCOserver settings

- [ ] Overwrite primitive time with PC time
- [✓] Insert PC time stamps every 1 minutes

usage overview

- The PCO-Controller:
  - ⇨ is actually executed by pco2.bat and starts a configurable set of other applications, like xPanel
  - ⇨ provides access to PCO-Server
    - to start logging of data
    - to replay logged data
  - ⇨ allows sending of FRAME system primitives to the protocol stack (via PCO-Server)
    - from direct user input
    - out of a pool of predefined system primitives loaded from a dedicated file (usually cfg/pco_defcfg.xml)
    - by selections in a matrix (stored per default in cfg/pco_defcfg.xml) which may be ex-/imported
  - ⇨ enables setup of the communication drivers to be used
  - ⇨ on exit: shuts down all initially started applications
    - called the “test environment”





Name: overnight

	Auto
● Send stack config	✓
○ Send: Activate PDP	
○ Send: set CGCLASS	
○ Send:	
○ Send:	
○ Send:	

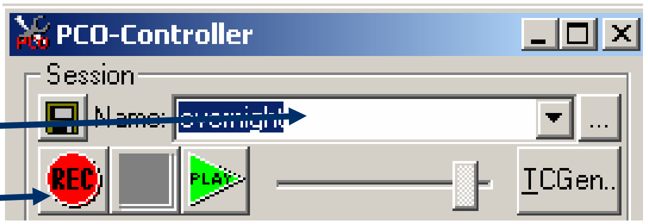
	Traces	Primitives
GMM		👆
MMI	✓	✓
MM		👆
RR	✓	
SMS		👆
GRR		
GRLC		
SM		👆
CC		👆
SIM		👆
LLC		
SND		👆
UART		👆
PPP		👆

Send system primitive to

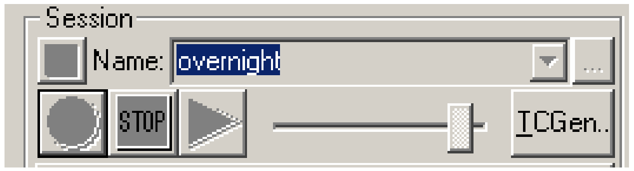
com 1 N

logging and replay

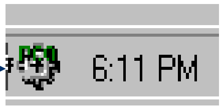
- Logging / Recording:
  - ⇨ specify name of test session
  - ⇨ start logging process



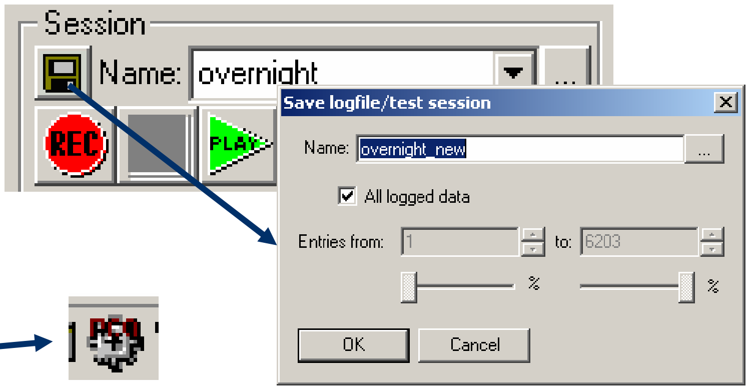
- ⇨ now every trace/primitive received via the test interface will be logged
  - ◆ Independent of any filter setting in a PCO-Viewer



- ⇨ PCO-Server appears green

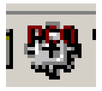


- ⇨ after pressing the “STOP” button …
  - ◆ a <session name>.pco file can be found in the current session dir of PCO-Server
  - ◆ a copy of (selected parts of) the logged session can be stored somewhere else (and, e.g., be sent to developers)



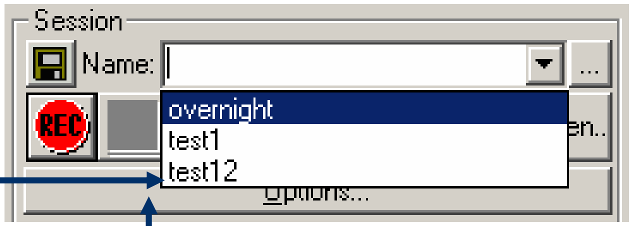
Name: overnight_new
☑ All logged data
Entries from: 1 to: 6203

- ⇨ PCO-Server appears red again

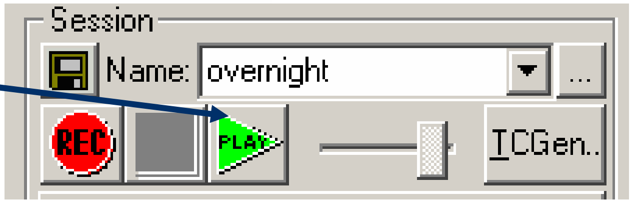


logging and replay

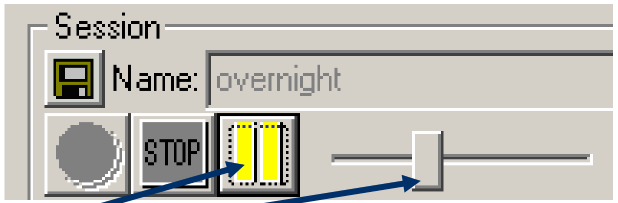
- Replay:
  - ⇨ select test session name
  - ⇨ or drag/drop .pco-file to PCO-Controller



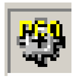
  - ⇨ press the „PLAY“ button



  - ⇨ now logged traces/primitives will be replayed in connected PCO-Viewers
    - depending on the individual filter settings
  - ⇨ pausing and repositioning are possible

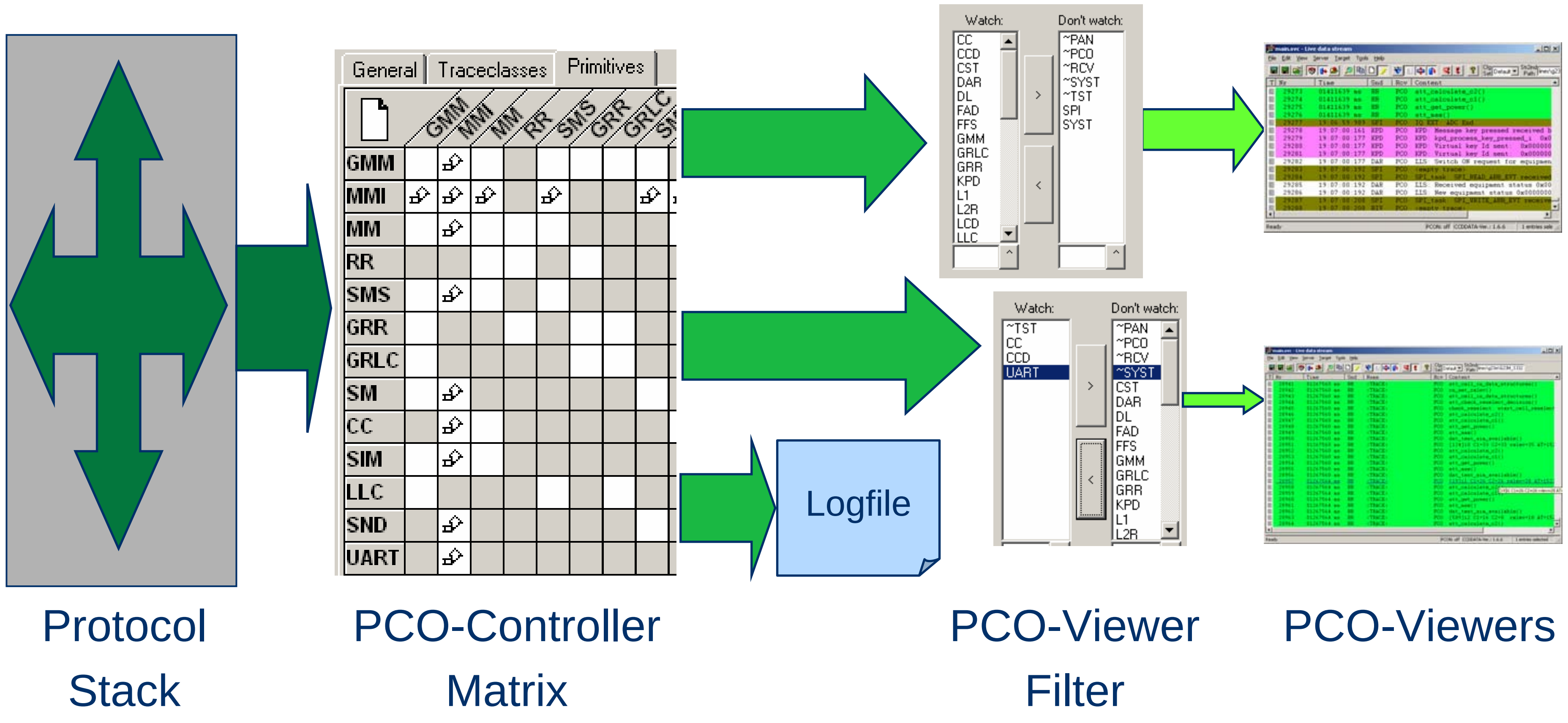


  - ⇨ PCO-Server appears yellow



filter setup

- Trace/primitive filtering is done in two stages:



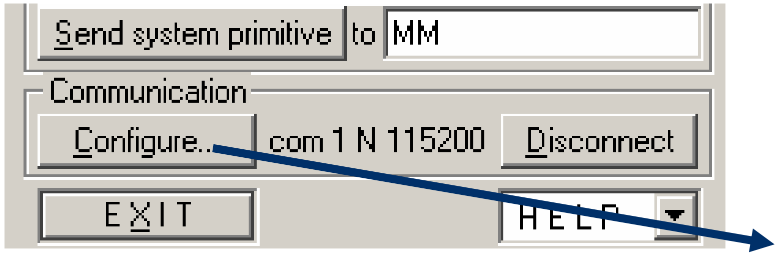
	GMM	MMI	MM	RR	SMS	GRR	GRLC	SM
GMM		✓						
MMI	✓	✓	✓		✓			✓
MM		✓						
RR								
SMS		✓						
GRR								
GRLC								
SM		✓						
CC		✓						
SIM		✓						
LLC								
SND		✓						
UART		✓						

Watch: CC, CCD, CST, DAR, DL, FAD, FFS, GMM, GRLC, GRR, KPD, L1, L2R, LCD, LLC
Don't watch: ~PAN, ~PCO, ~RCV, ~SYST, ~TST, SPI, SYST

Watch: ~TST, CC, CCD, UART
Don't watch: ~PAN, ~PCO, ~RCV, ~SYST, CST, DAR, DL, FAD, FFS, GMM, GRLC, GRR, KPD, L1, L2R

communication setup

PCO-Controller:

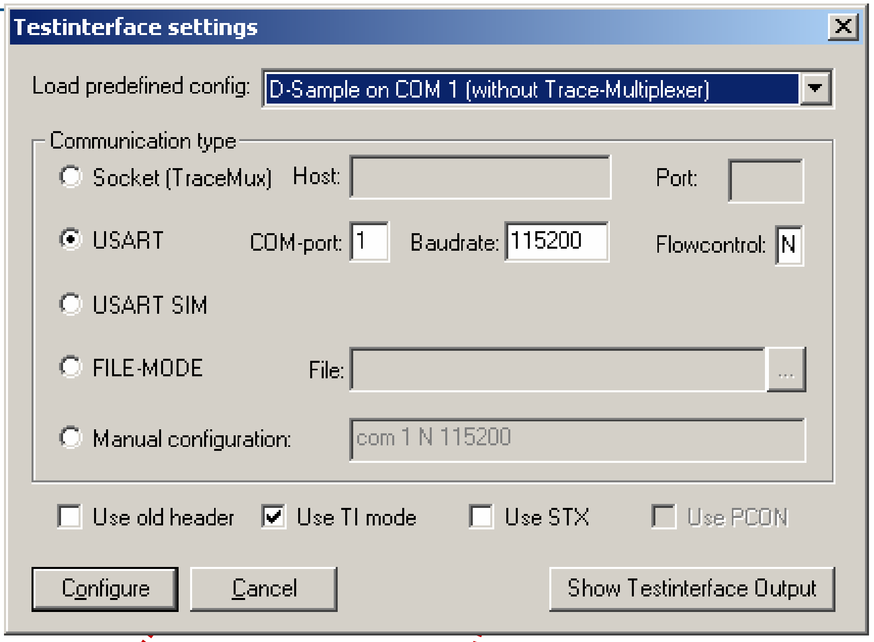


Send system primitive to MM

Communication

Configure... com 1 N 115200 Disconnect

EXIT HELP



Testinterface settings

Load predefined config: D-Sample on COM 1 (without Trace-Multiplexer)

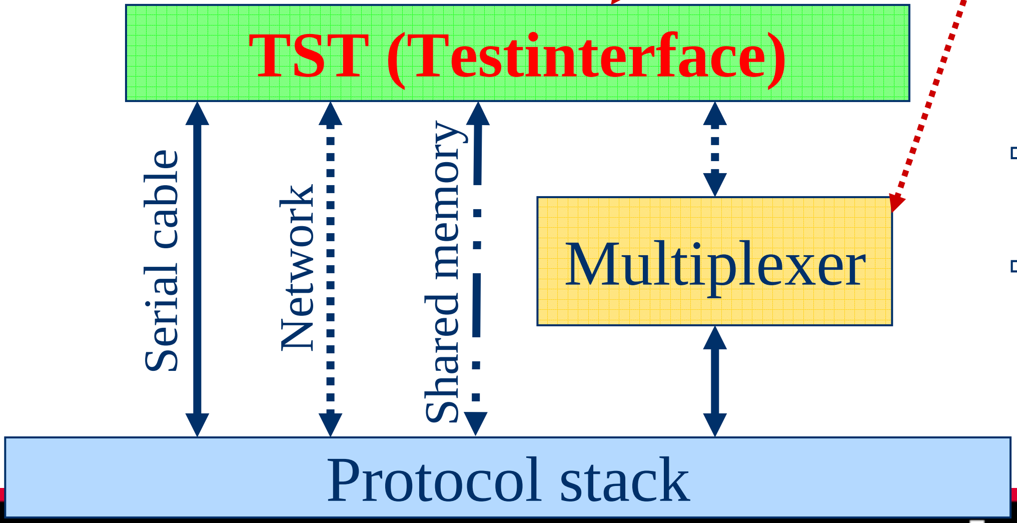
Communication type

- ○ Socket (TraceMux) Host: Port:
- ● USART COM-port: 1 Baudrate: 115200 Flowcontrol: N
- ○ USART SIM
- ○ FILE-MODE File:
- ○ Manual configuration: com 1 N 115200

☐ Use old header ☑ Use TI mode ☐ Use STX ☐ Use PCON

Configure Cancel Show Testinterface Output

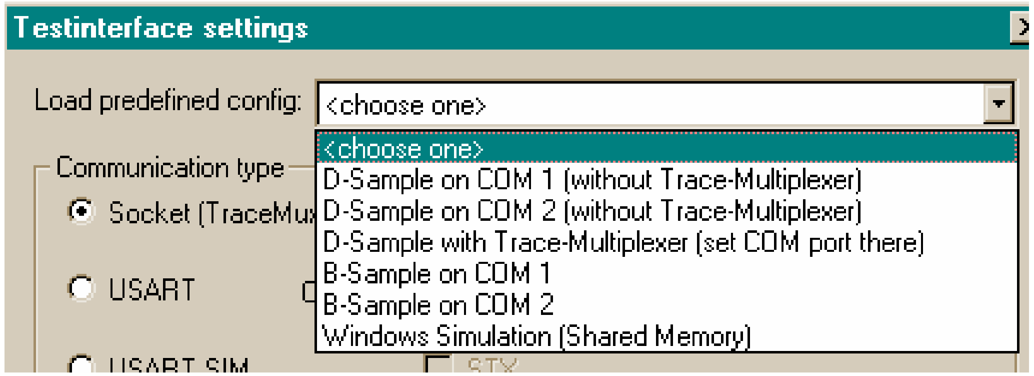
- ⇨ selection of mode
- ⇨ secification of individual parameters



- ⇨ configuration of test interface
- ⇨ evtl. start of extra tools

communication setup

⇨ for convenience several default configurations exist



Testinterface settings

Load predefined config: <choose one>

Communication type

● Socket (TraceMux

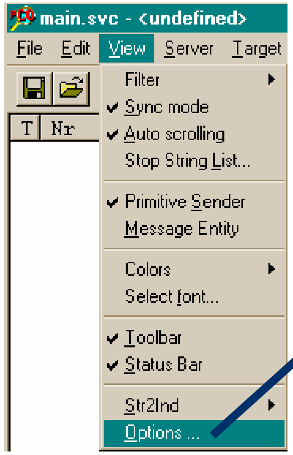
○ USART

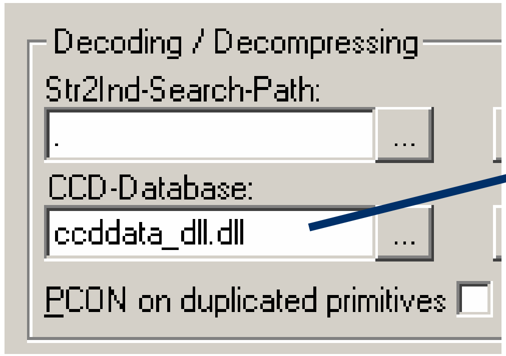
○ USART SIM

<choose one>
D-Sample on COM 1 (without Trace-Multiplexer)
D-Sample on COM 2 (without Trace-Multiplexer)
D-Sample with Trace-Multiplexer (set COM port there)
B-Sample on COM 1
B-Sample on COM 2
Windows Simulation (Shared Memory)

⇨ it has to be ensured that a matching ccddata-DLL is selected

PCO-Viewer:





Decoding / Decompressing

Str2Ind-Search-Path: .

CCD-Database: ccddata_dll.dll

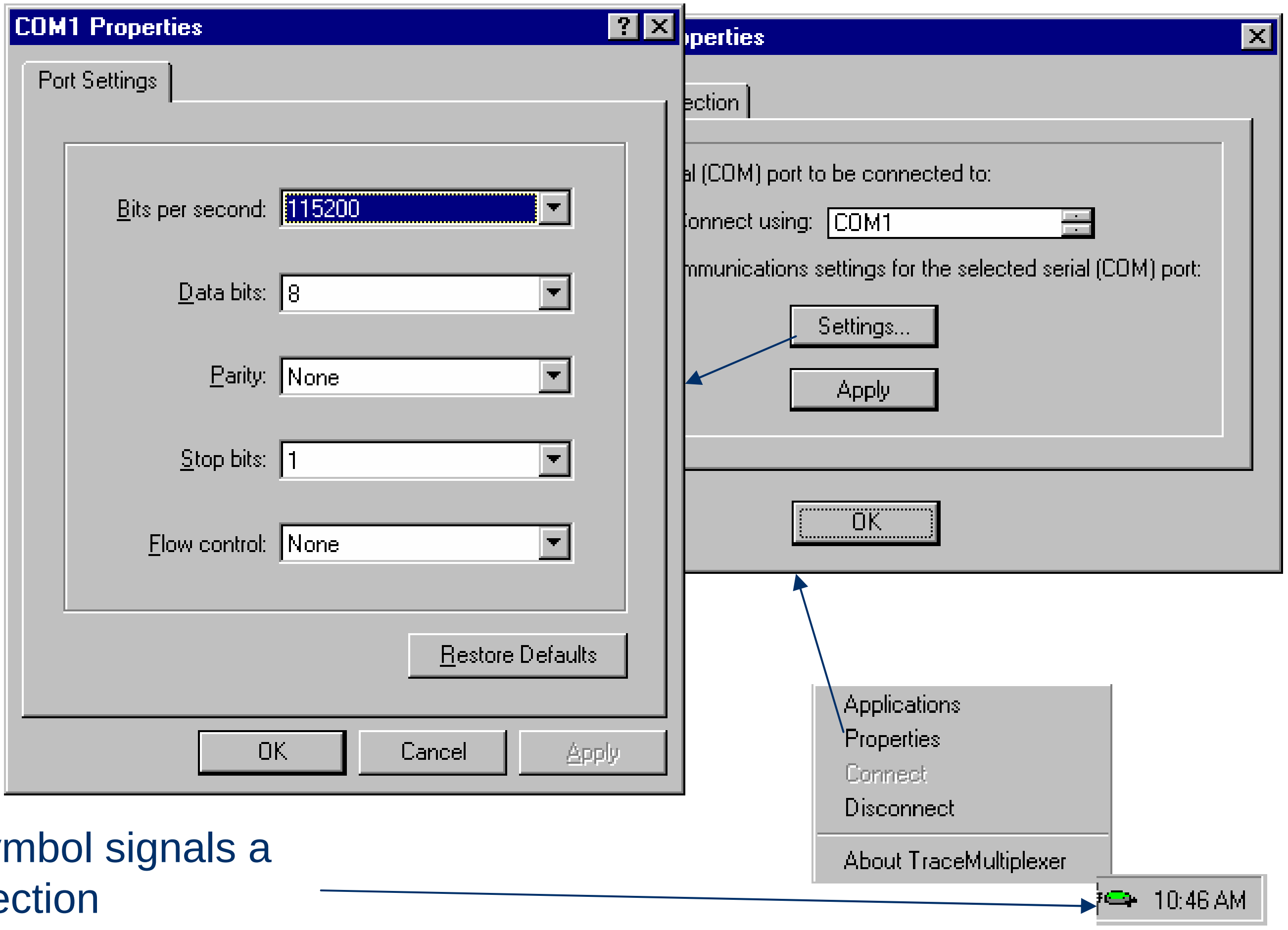
PCON on duplicated primitives ☐

- either a DLL which has been delivered together with the PS-image
- or a prebuild DLL from ...\ccddata\ (e.g. ccddata_G23M_333_S64.dll for older B-Sample-Releases)

communication setup

- Communication via TraceMultiplexer:
  - ⇨ if using the TraceMultiplexer for the first time it has to be configured:

1. choose COM port
2. select baudrate of 115200
3. disable flow control



COM1 Properties

Port Settings

Bits per second: 115200

Data bits: 8

Parity: None

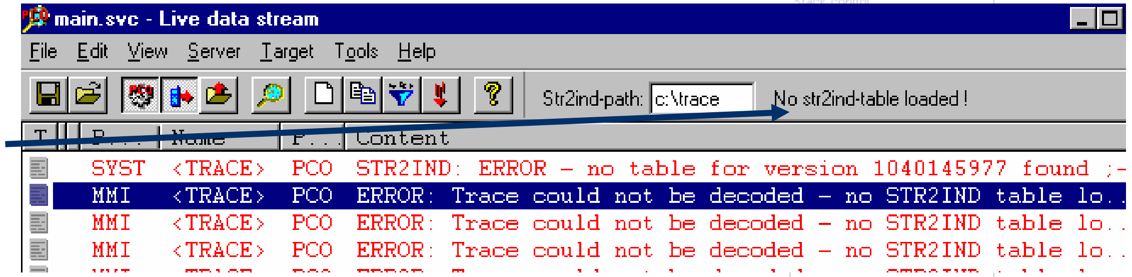
Stop bits: 1

Flow control: None

⇨ the green tray symbol signals a successful connection

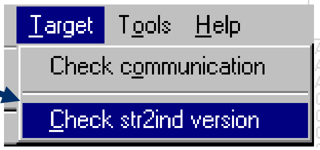
usage overview

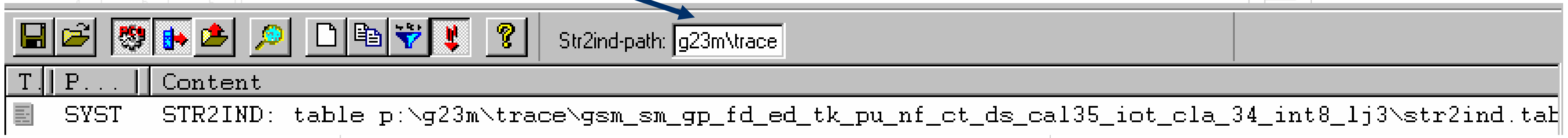
- Compressed Tracing with Str2Ind-Tables:
  - ⇨ for performance and memory reasons traces are compressed at compile time
  - ⇨ each PS build creates a str2ind-table containing [ID]->[Trace string] combinations
  - ⇨ initially no table for interpretation of Trace-IDs is loaded


main.svc - Live data stream
File Edit View Server Target Tools Help
Str2ind-path: c:\trace
No str2ind-table loaded !

```
T  P...  Name     P...  Content
   SYST  <TRACE>  PCO   STR2IND: ERROR - no table for version 1040145977 found ;-
   MMI   <TRACE>  PCO   ERROR: Trace could not be decoded - no STR2IND table lo..
   MMI   <TRACE>  PCO   ERROR: Trace could not be decoded - no STR2IND table lo..
   MMI   <TRACE>  PCO   ERROR: Trace could not be decoded - no STR2IND table lo..
```

  - ⇨ after reset or by explicit request
    - ◆ PS sends a version number
    - ◆ Viewer searches for matching .tab-file in specified directory structure



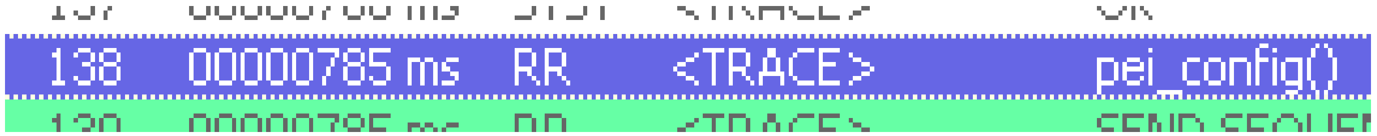


```
T  P...  Content
   SYST  STR2IND: table p:\g23m\trace\gsm_sm_gp_fd_ed_tk_pu_nf_ct_ds_cal35_iot_cla_34_int8_lj3\str2ind.tab
```

  - ⇨ after loading of the table all traces will be shown as expected

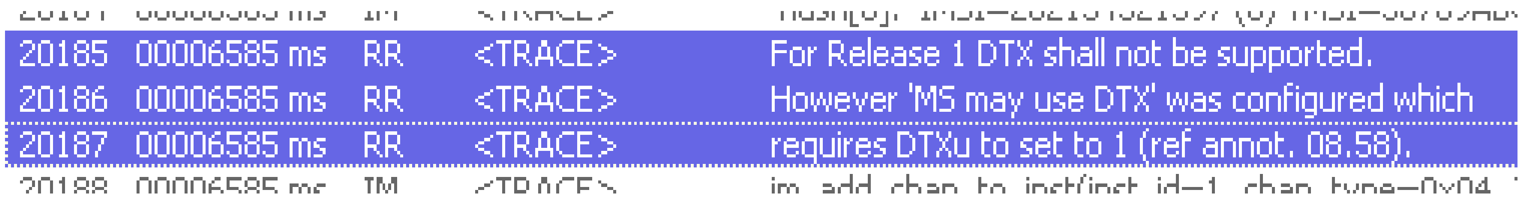
usage overview

- Types of traces shown by PCO-Viewer:
  - ⇨ Function traces …



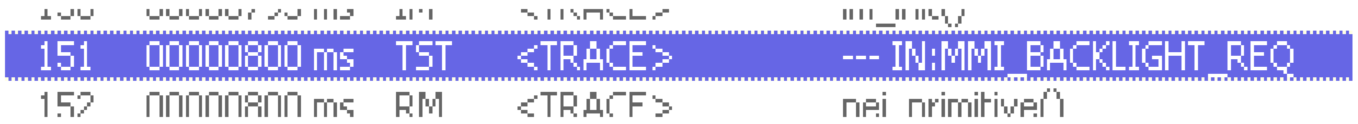
```
138   00000785 ms   RR   <TRACE>          pei_config()
```

  - ⇨ Event traces …



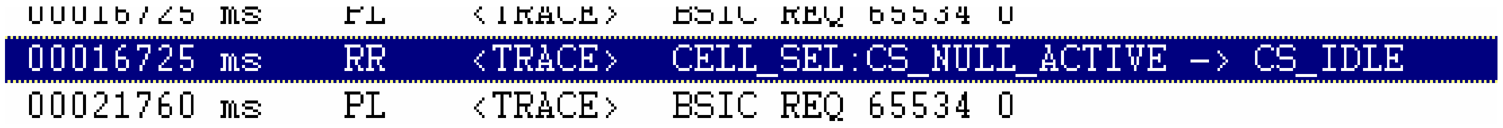
```
20185  00006585 ms  RR   <TRACE>    For Release 1 DTX shall not be supported.
20186  00006585 ms  RR   <TRACE>    However 'MS may use DTX' was configured which
20187  00006585 ms  RR   <TRACE>    requires DTXu to set to 1 (ref annot. 08.58).
```

  - ⇨ Primitive traces …



```
151   00000800 ms  TST   <TRACE>        --- IN:MMI_BACKLIGHT_REQ
152   00000800 ms  RM    <TRACE>        pei_primitive()
```

  - ⇨ State traces …

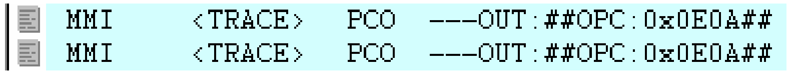


```
00016725 ms    PL    <TRACE>  BSIC REQ 65534 0
00016725 ms    RR    <TRACE>  CELL_SEL:CS_NULL_ACTIVE -> CS_IDLE
00021760 ms    PL    <TRACE>  BSIC REQ 65534 0
```

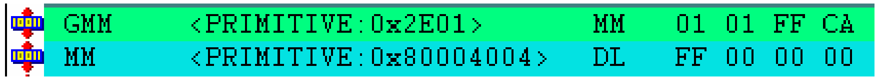
interpreting/decoding

- What is a ccddata-DLL used for (e.g. ccddata_dll.dll) ?
  - ⇨ contains information about all primitive and air message structures used in the corresponding protocol stack

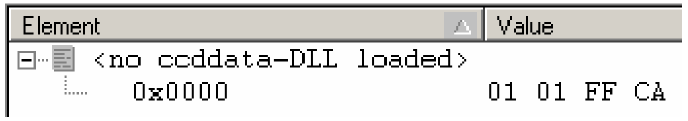
Without:



```
MMI     <TRACE>   PCO   ---OUT:##OPC:0x0E0A##
MMI     <TRACE>   PCO   ---OUT:##OPC:0x0E0A##
```

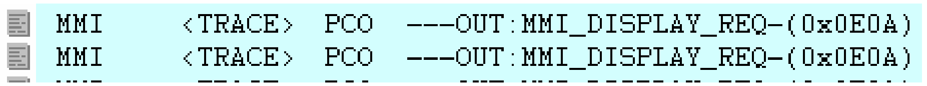


```
GMM     <PRIMITIVE:0x2E01>        MM    01 01 FF CA
MM      <PRIMITIVE:0x80004004>    DL    FF 00 00 00
```

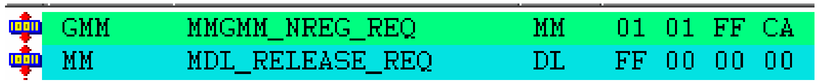


```
Element                          Value
⊟ <no ccddata-DLL loaded>
     0x0000                      01 01 FF CA
```

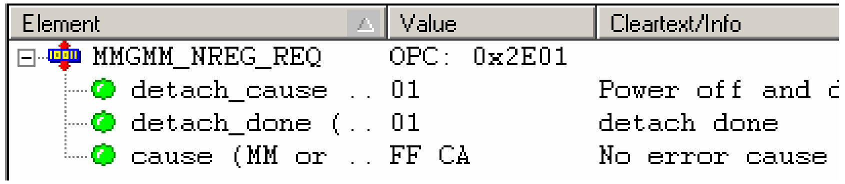
With a matching DLL:



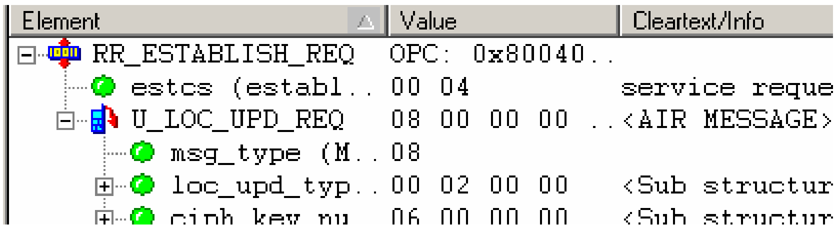
```
MMI     <TRACE>   PCO   ---OUT:MMI_DISPLAY_REQ-(0x0E0A)
MMI     <TRACE>   PCO   ---OUT:MMI_DISPLAY_REQ-(0x0E0A)
```



```
GMM     MMGMM_NREG_REQ       MM    01 01 FF CA
MM      MDL_RELEASE_REQ      DL    FF 00 00 00
```



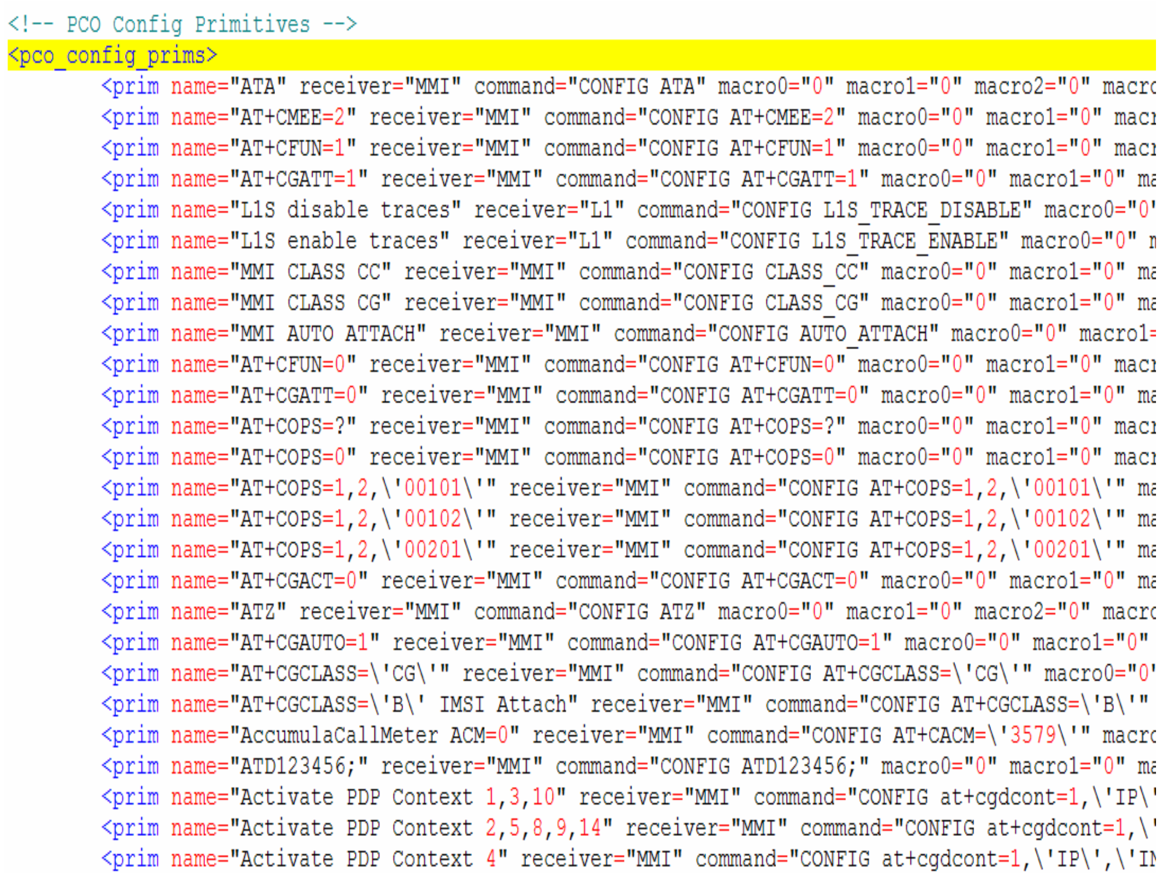
```
Element                    Value          Cleartext/Info
⊟ MMGMM_NREG_REQ           OPC: 0x2E01
    detach_cause ..        01             Power off and d
    detach_done (..        01             detach done
    cause (MM or ..        FF CA          No error cause
```



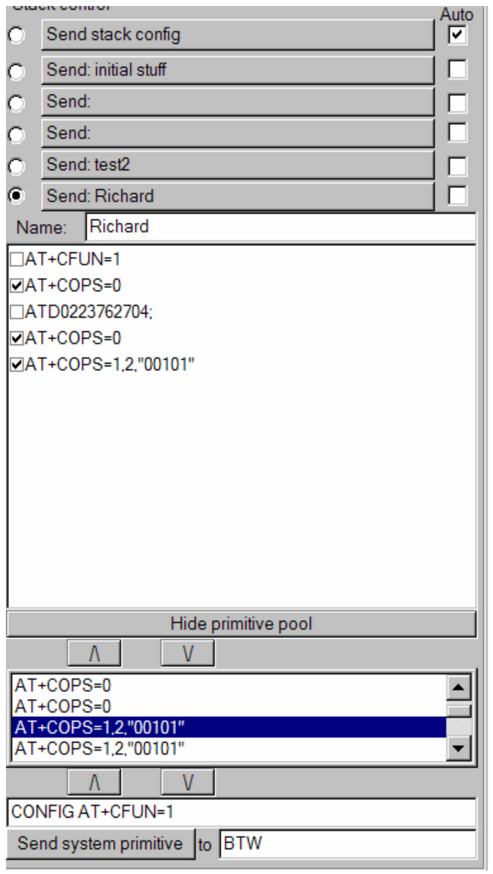
```
Element                    Value          Cleartext/Info
⊟ RR_ESTABLISH_REQ         OPC: 0x80040..
    estcs (establ..        00 04          service reque
  ⊟ U_LOC_UPD_REQ          08 00 00 00 ..<AIR MESSAGE>
      msg_type (M..        08
    ⊞ loc_upd_typ..        00 02 00 00    <Sub structur
    ⊞ ciph_key_nu          06 00 00 00    <Sub structur
```

configuration file

- cfg\pco_defcfg.xml
  - ⇨ Contains all PCO config primitives



```
<!-- PCO Config Primitives -->
<pco_config_prims>
        <prim name="ATA" receiver="MMI" command="CONFIG ATA" macro0="0" macro1="0" macro2="0" macr
        <prim name="AT+CMEE=2" receiver="MMI" command="CONFIG AT+CMEE=2" macro0="0" macro1="0" macr
        <prim name="AT+CFUN=1" receiver="MMI" command="CONFIG AT+CFUN=1" macro0="0" macro1="0" macr
        <prim name="AT+CGATT=1" receiver="MMI" command="CONFIG AT+CGATT=1" macro0="0" macro1="0" ma
        <prim name="L1S disable traces" receiver="L1" command="CONFIG L1S_TRACE_DISABLE" macro0="0'
        <prim name="L1S enable traces" receiver="L1" command="CONFIG L1S_TRACE_ENABLE" macro0="0" m
        <prim name="MMI CLASS CC" receiver="MMI" command="CONFIG CLASS_CC" macro0="0" macro1="0" ma
        <prim name="MMI CLASS CG" receiver="MMI" command="CONFIG CLASS_CG" macro0="0" macro1="0" ma
        <prim name="MMI AUTO ATTACH" receiver="MMI" command="CONFIG AUTO_ATTACH" macro0="0" macro1=
        <prim name="AT+CFUN=0" receiver="MMI" command="CONFIG AT+CFUN=0" macro0="0" macro1="0" macr
        <prim name="AT+CGATT=0" receiver="MMI" command="CONFIG AT+CGATT=0" macro0="0" macro1="0" ma
        <prim name="AT+COPS=?" receiver="MMI" command="CONFIG AT+COPS=?" macro0="0" macro1="0" macr
        <prim name="AT+COPS=0" receiver="MMI" command="CONFIG AT+COPS=0" macro0="0" macro1="0" macr
        <prim name="AT+COPS=1,2,\'00101\'" receiver="MMI" command="CONFIG AT+COPS=1,2,\'00101\'" ma
        <prim name="AT+COPS=1,2,\'00102\'" receiver="MMI" command="CONFIG AT+COPS=1,2,\'00102\'" ma
        <prim name="AT+COPS=1,2,\'00201\'" receiver="MMI" command="CONFIG AT+COPS=1,2,\'00201\'" ma
        <prim name="AT+CGACT=0" receiver="MMI" command="CONFIG AT+CGACT=0" macro0="0" macro1="0" ma
        <prim name="ATZ" receiver="MMI" command="CONFIG ATZ" macro0="0" macro1="0" macro2="0" macr
        <prim name="AT+CGAUTO=1" receiver="MMI" command="CONFIG AT+CGAUTO=1" macro0="0" macro1="0"
        <prim name="AT+CGCLASS=\'CG\'" receiver="MMI" command="CONFIG AT+CGCLASS=\'CG\'" macro0="0'
        <prim name="AT+CGCLASS=\'B\' IMSI Attach" receiver="MMI" command="CONFIG AT+CGCLASS=\'B\'"
        <prim name="AccumulaCallMeter ACM=0" receiver="MMI" command="CONFIG AT+CACM=\'3579\'" macr
        <prim name="ATD123456;" receiver="MMI" command="CONFIG ATD123456;" macro0="0" macro1="0" ma
        <prim name="Activate PDP Context 1,3,10" receiver="MMI" command="CONFIG at+cgdcont=1,\'IP\'
        <prim name="Activate PDP Context 2,5,8,9,14" receiver="MMI" command="CONFIG at+cgdcont=1,\'
        <prim name="Activate PDP Context 4" receiver="MMI" command="CONFIG at+cgdcont=1,\'IP\',\'IN
```



Name: Richard

☐ AT+CFUN=1
☑ AT+COPS=0
☐ ATD0223762704;
☑ AT+COPS=0
☑ AT+COPS=1,2,"00101"

AT+COPS=0
AT+COPS=0
AT+COPS=1,2,"00101"
AT+COPS=1,2,"00101"

CONFIG AT+CFUN=1

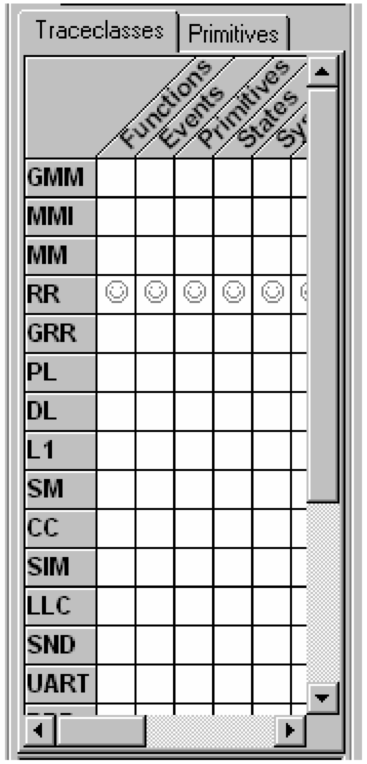
Send system primitive to BTW

configuration file

- ⇨ contains all “Matrix”-entries of the PCO-Controller
- ⇨ may be edited to e.g. change the entry-order

```
<stack>
  <entity name="GMM" traceclass="00000040"/ duplicates=""/>
  <entity name="MMI" traceclass="00000040"/ duplicates=""/>
  <entity name="MM" traceclass="00000040"/ duplicates=""/>
  <entity name="RR" traceclass="00000040"/ duplicates=""/>
  <entity name="GRR" traceclass="00000040"/ duplicates=""/>
  <entity name="SM" traceclass="00000040"/ duplicates=""/>
  <entity name="CC" traceclass="00000040"/ duplicates=""/>
  <entity name="SIM" traceclass="00000040"/ duplicates=""/>
  <entity name="LLC" traceclass="00000040"/ duplicates=""/>
  <entity name="SND" traceclass="00000040"/ duplicates=""/>
  <entity name="UART" traceclass="00000040"/ duplicates=""/>
  <entity name="PPP" traceclass="00000040"/ duplicates=""/>
  <entity name="SMS" traceclass="00000040"/ duplicates=""/>
  <entity name="SS" traceclass="00000040"/ duplicates=""/>
  <entity name="FAD" traceclass="00000040"/ duplicates=""/>
  <entity name="RLP" traceclass="00000040"/ duplicates=""/>
  <entity name="L2R" traceclass="00000040"/ duplicates=""/>
  <entity name="T30" traceclass="00000040"/ duplicates=""/>
  <entity name="L1" traceclass="00000040"/ duplicates=""/>
  <entity name="PL" traceclass="00000040"/ duplicates=""/>
  <entity name="DL" traceclass="00000040"/ duplicates=""/>
</stack>
```

→



	Functions	Events	Primitives	States	Sy
GMM					
MMI					
MM					
RR	☺	☺	☺	☺	☺
GRR					
PL					
DL					
L1					
SM					
CC					
SIM					
LLC					
SND					
UART					

important files

- Volatile files:
  - ⇨ have to be build together with the used protocol stack
    - **ccddata_dll.dll** (database with primitive symbols)
    - **str2ind.tab** (table with “ID <-> trace text” associations)