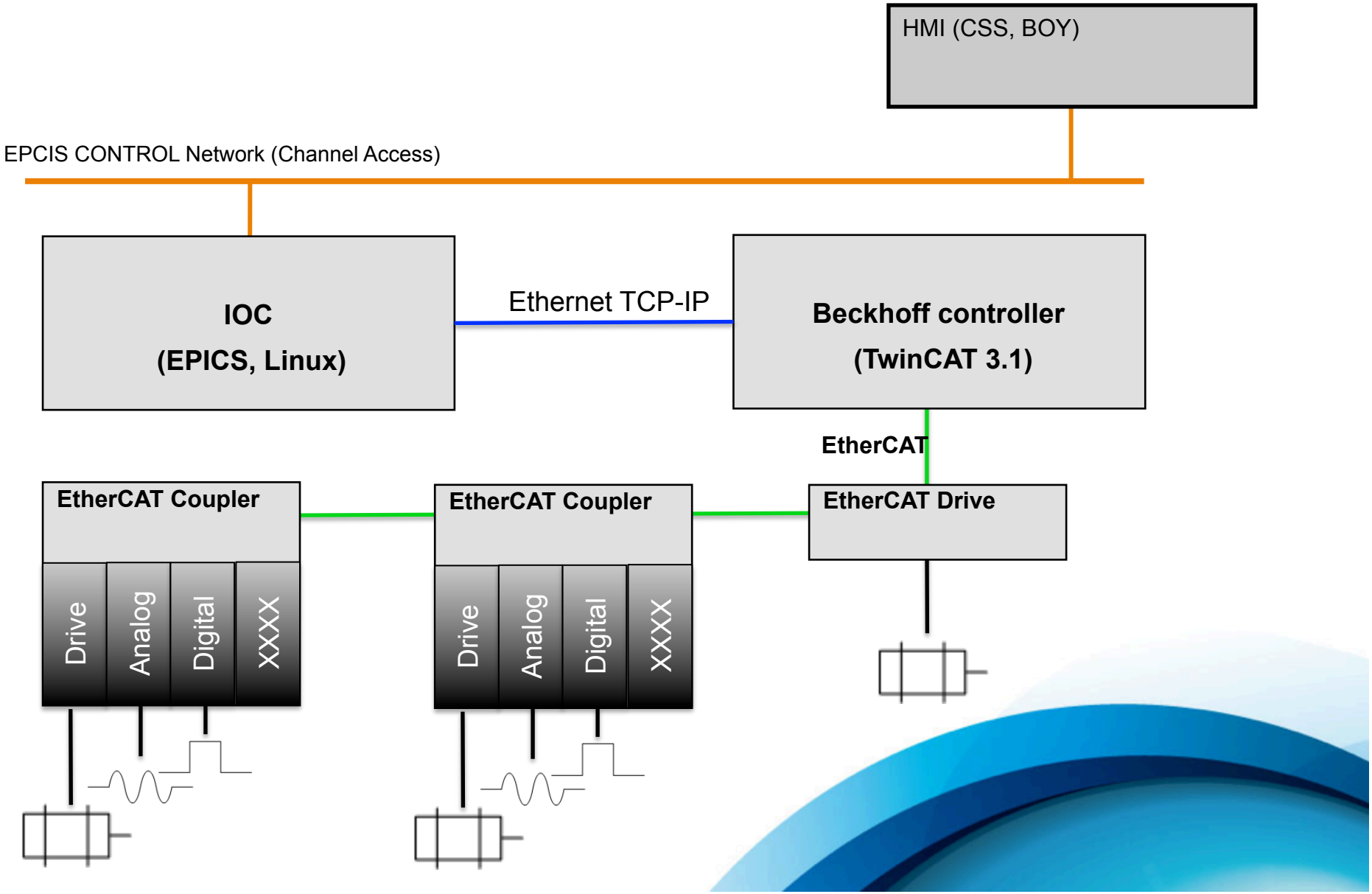
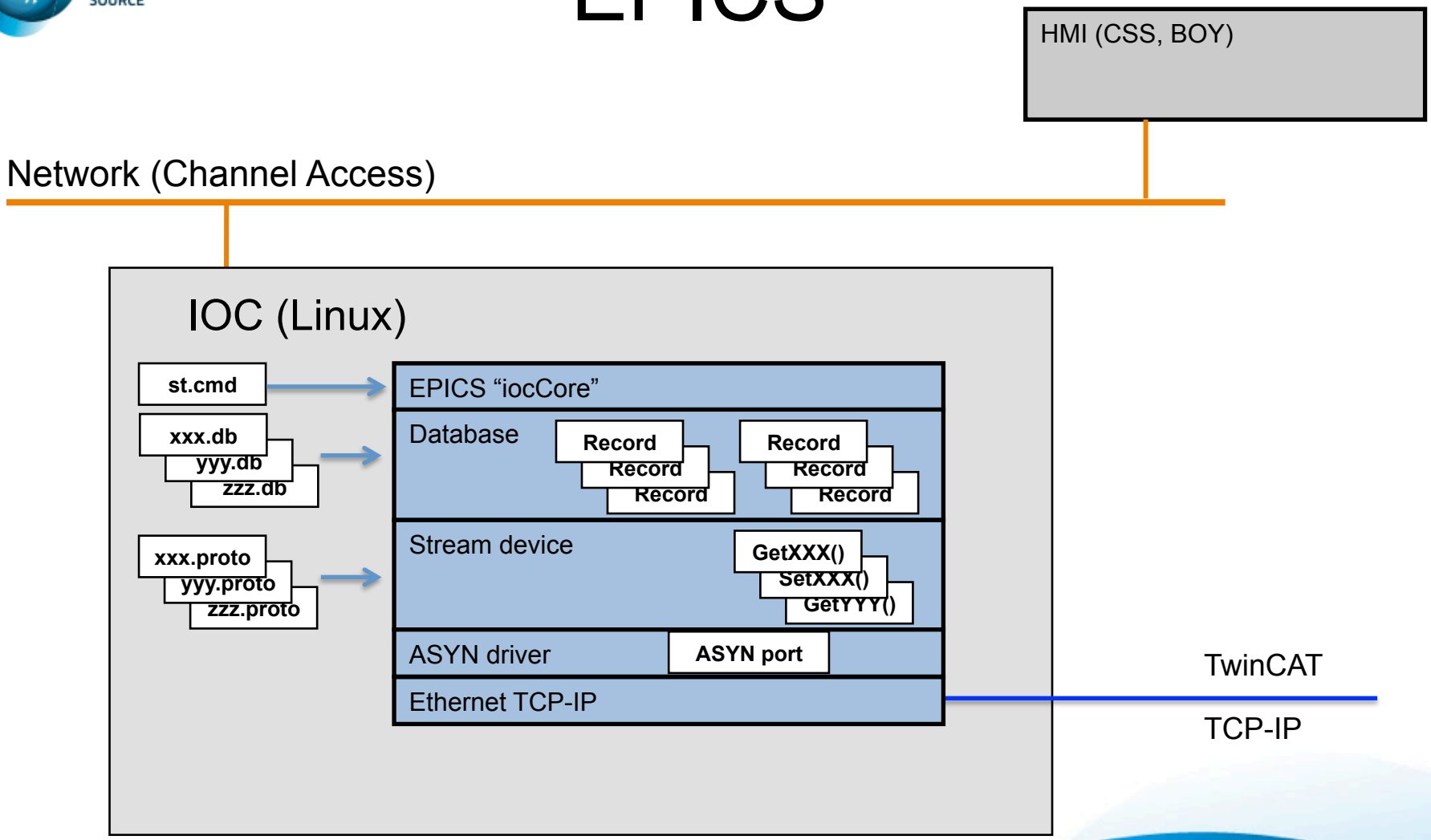
Beckhoff TwinCAT 3 Integration into EPICS

Anders Sandström
ESS MCAG 2015-06-26

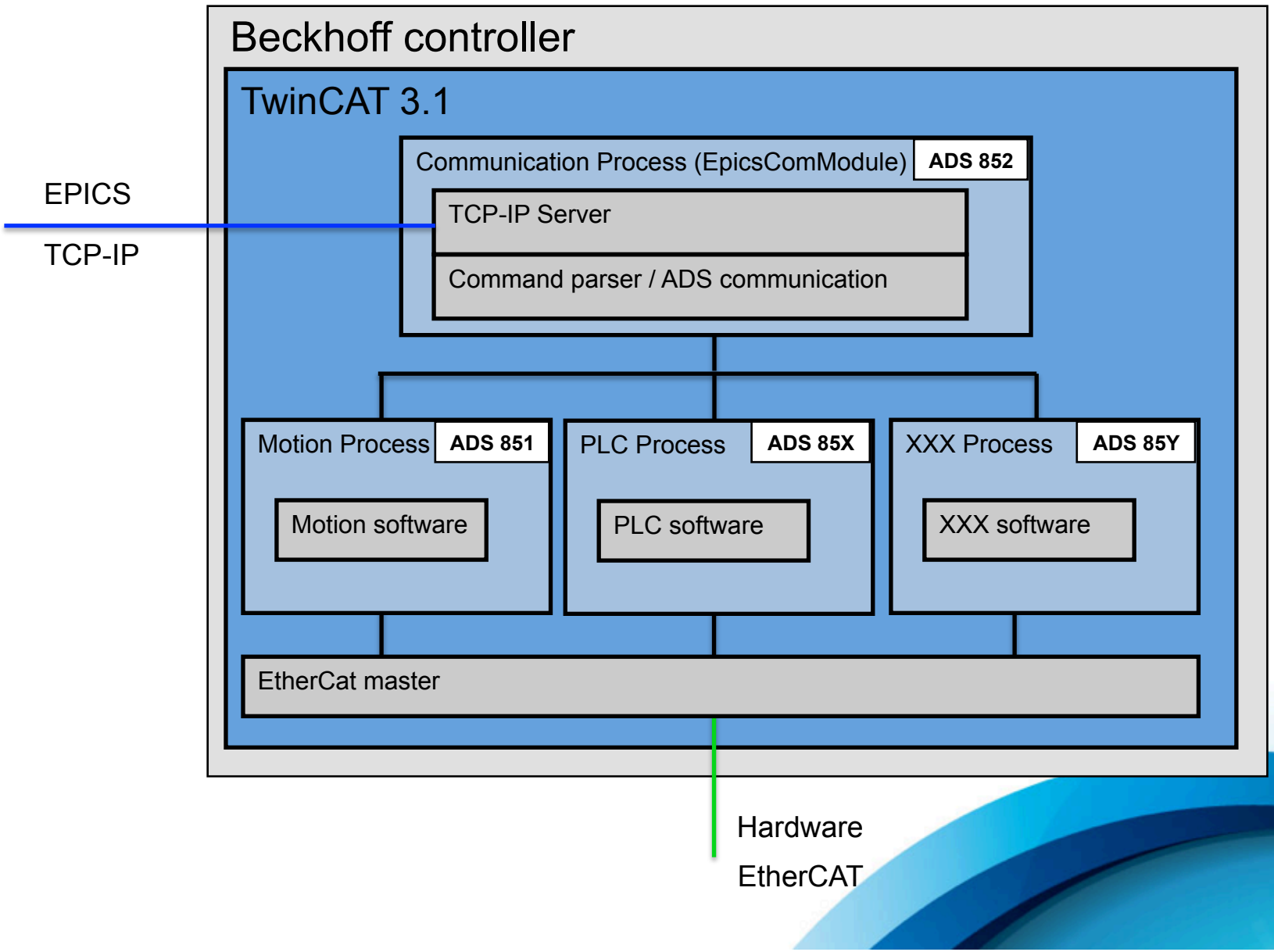
Overview


HMI (CSS, BOY)
EPCIS CONTROL Network (Channel Access)
IOC
(EPICS, Linux)
Ethernet TCP-IP
Beckhoff controller
(TwinCAT 3.1)
EtherCAT
EtherCAT Coupler
Drive
Analog
Digital
XXXX
EtherCAT Coupler
Drive
Analog
Digital
XXXX
EtherCAT Drive

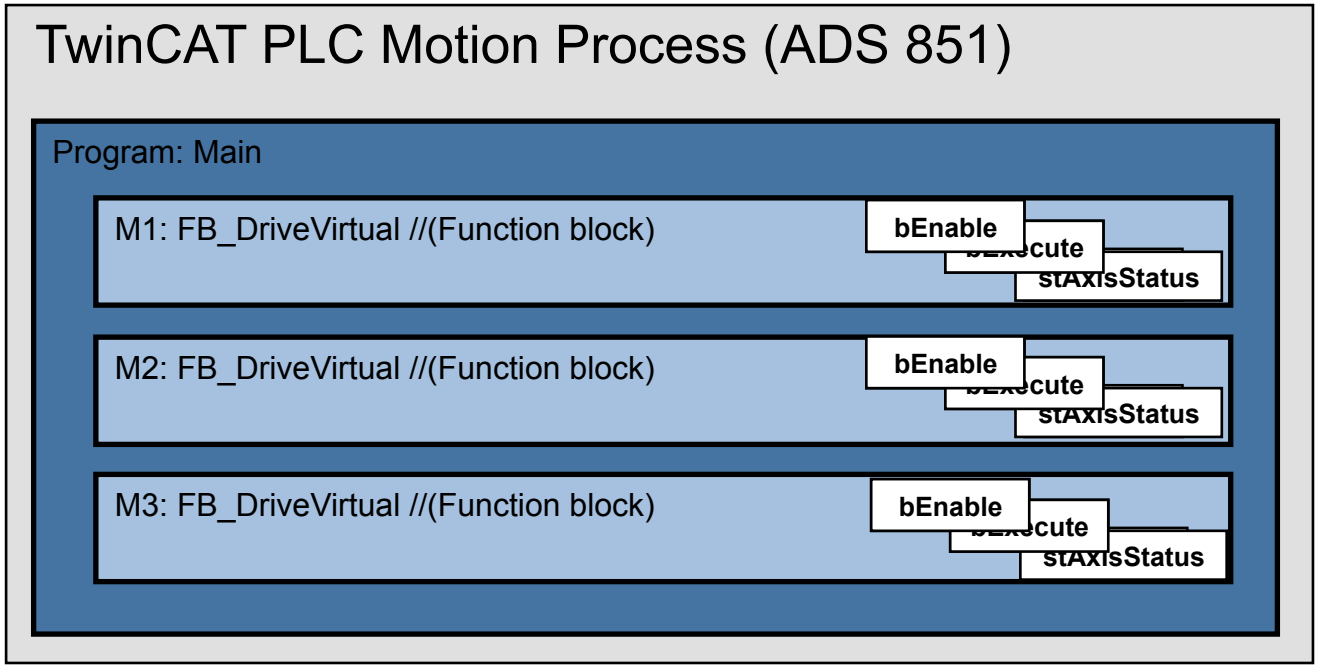
EPICS


HMI (CSS, BOY)
Network (Channel Access)
IOC (Linux)
st.cmd
xxx.db
yyy.db
zzz.db
xxx.proto
yyy.proto
zzz.proto
EPICS “iocCore”
Database
Record
Record
Record
Record
Record
Record
Stream device
GetXXX()
SetXXX()
GetYYY()
ASYN driver
ASYN port
Ethernet TCP-IP
TwinCAT
TCP-IP

TwinCAT controller



Motion Process (FB_DriveVirtual = control one axis)



FB_DriveVirtual supports: Jog, constant speed, absolute position, relative position, homing, …….

Protocol General

- Frame terminator = In (linefeed = ascii 10)
- Commands separator = “;”
- An option may be supplied before each command separated with”/”
- Commands can be stacked

Example:

“**Option1#**Command1;Command2;**Option3#**Command 3;
”

Protocol Write

Writing a value:

- "Symbolic name in PLC=value;"
- Acknowledgement:
  - "OK;" if successful
  - Error code if not successful

Example:

"Main.M1.fPosition=100; Main.M1.fVelocity=1000;
"

Will return: "OK;OK;
"

Protocol Read

Reading a value:

- "Symbolic name in PLC?;"
- Acknowledgement:
  - Value if successful
  - Error code if not successful

Example:

"Main.M1.fPosition?; Main.M1.fVelocity?;
"

Will return: "100;1000;
"

Protocol Option ADSPORT

ADSPORT:

- Sets ADS-port temporary for the command
- ADS-Port is default set to 851 which normally is the first PLC

Example (accessing two different modules in TwinCAT):

“ADSPORT=851/Main.M1.fPosition=100;ADSPORT=852/Main.iCounter?;
”

Supported native data types

	TwinCat data type	Size [bits]	Array	Comment
1	ADST_BIT	1	Yes	Bit
2	ADST_SINT8	8	Yes	Short Integer.
3	ADST_UINT8	8	Yes	Byte
4	ADST_INT16	16	Yes	Integer
5	ADST_UINT16	16	Yes	Unsigned Integer
6	ADST_INT32	32	Yes	Double Integer
7	ADST_UINT32	32	Yes	Unsigned Integer
8	ADST_INT64	64	Yes	Long Integer
9	ADST_UINT64	64	Yes	Unsigned Long Integer
10	ADST_REAL32	32	Yes	Real
11	ADST_REAL64	64	Yes	Float
12	ADST_STRING	8*length	No	String (arrays of strings not supported)

Supported custom data types
DUT_AxisStatus

DUT_AxisStatus:
The structure DUT_AxisStatus includes the most important information available for one axis and can be **read** with one single command.

Restrictions:
Only implemented for reading purpose.
Arrays of this structure is not supported (since EPICS don't have support for arrays of structures).

Example:
"Main.M1.stAxisStatus?"

Will return:
"Main.M1.stAxisStatus= 1 ,0, 1, 2, …….;
"

```
DUT_AxisStatus_v0_01:
    bEnable: BOOL;
    bReset: BOOL;
    bExecute: BOOL;
    nCommand: UINT;
    nCmdData: UINT;
    fVelocity: LREAL;
    fPosition: LREAL;
    fAcceleration: LREAL;
    fDeceleration: LREAL;
    bJogFwd: BOOL;
    bJogBwd: BOOL;
    bLimitFwd: BOOL;
    bLimitBwd: BOOL;
    fOverride: LREAL:=100;
    bHomeSensor: BOOL;
    bEnabled: BOOL;
    bError: BOOL;
    nErrorId: UDINT;
    fActVelocity: LREAL;
    fActPosition: LREAL;
    fActDiff: LREAL;
    bHomed:BOOL;
    bBusy:BOOL;
```