Karlsruher Institut für Technologie

ISS-ANKA, KIT Campus North, Karlsruhe
Germany

THE ANKA B-FIELD TEST FACILITY CONTROL SYSTEM, BASED ON A SPEC MACRO PACKAGE ENHANCED SETUP

Karlheinz Cerff, Thomas Spangenberg, Wolfgang Mexner, Institut for Synchrotron Radiation, (ISS)-ANKA, Karlsruhe Institut of Technology, (KIT)-Campus North, Germany

The ANKA B-field test facility provides users with a flexible tool to investigate magnetic field distributions of different setups of coils or permanent magnets, optimal sensor types, geometrical alignments of probes and the possibility to change the independent physical stimuli to generate and alter magnetic field distributions[1]. From the point of Software development it is taken as an example of a straight-forward device implementation with a recently introduced type of macro based 'building block system' for devices in SPEC, [2]. This macro package provides the C-like SPEC with an object orientated framework with a namespace and class concept to represent the power supplies of different brands, probe positioning devices and measurement amplifiers.

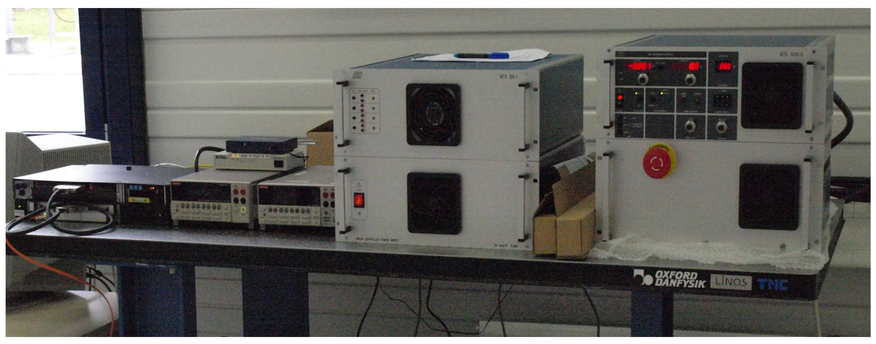

Fig.1, B-Test Facility Devices ready for shipment, from left to right: motor controller and amplifier, GPIB/TCP gateway, two digital multi-meters and the two power supplies.

B-TEST FACILITY DEVICE DRIVERS

The program code which has to be written are the device driver macros for power supplies "fug" and "fugbig" and the digital multi-meter with the connected Hall-probes.
The functions can be grouped in :

- internal functions, like socket functions to set/get specific hardware register values, to reset or initialize devices, to address sub device and functions to process data strings received.
- functions for data synchronisation with the pre-defined standard values in associative arrays or with the ongoing values of the hardware device of interest,
- functions to set /get device parameters by calling external measurement devices used at ANKA-beam-lines
- functions, which are 'built in' SPEC, here used for the linear motor drive with encoders to position Hall-probes.

FUNCTION-MAPPING

A set of 'standard-' or user functions' for communication is generated automatically by the macro package. The bulky type of driver functions with long argument lists is mapped to a set of more user friendly functions. The functions have the general form:

def blxx_function (value, argument) '{
<return> driver function ("devicename **.** property")
}
The simplest user functions don't have arguments, for example a simple 'blct'-call, gives the outputs of all parameters of the assembly of power supplies, DMMs, and Hall-probes of the B-Test facility:
A generic example for function mapping, will be the 'setcurrentrate' user function for 8-fold power supply 'fug', device No 3, with a I-current rate of 0.2A/sec.The user function call is ,

blset_fug_currentrate3(0.2)

#mapping to the device driver function :

FUG_setcurrentrate3("fug", 0.2, 3).

#This calls the SPEC socket functions of the driver to write an appropriate value to the hardware register, subadress 3 of the power supply "fug", using command reference given in [5]:

def FUG_setcurrentrate3(device,quiet,value,) '{__
FUG_setcurrentrate(device,quiet,value,2)
}'

 # call of __internal driver function :
def __FUG_setcurrentrate(device,quiet,value,devnr) '{

#which type of power supply ?:
if (FUG[device]["$fugtype"]=="FUG-NTV 100") {

 # write external inputs for ps with devnr=2+1 to' value':
value = NumberInput ("current rate", FUG [device] [sprintf ("setcurrentrate%i",devnr+1)] ,0, 1, quiet, value);

 # call subdevice 3, addressing, convention, s. command reference [5]

*__FUG_sendcommand(device,sprintf ("%s>S%iR %g\n",sprintf("#%i",int(devnr/2)),devnr-2*int(devnr/2), value));*

 # The __internal function uses the basic 'built in'
SPEC socket_put function:

def __FUG_sendcommand(device,command) '{
.
sock_put (FUG ["device"] ["$adress"],command);
}
}'
 # 'value' gets the formatted readback from subdev. 3:

value = __FUG_splitanswer(__FUG_readback (device));

 # __internal function calls basic sock_get function:

def __FUG_readback(device) '{ local tmp;
tmp=sock_get(FUG[device]["$adress"]); .
}'
 # updates appropriate element of global array FUG with current read back value:

FUG[device][sprintf("setcurrentrate%i",devnr+1)]=__FUG_readcurrentrate (device,quiet ? 0 :1,devnr); }

BUILDING THE B-TEST FACILITY DEVICE MODEL

The power supplies are modelled by:
status, ramping behaviour, address, type, set/get/ voltages, I-currents, I-current rates. The devices are abstracted as sets of object variables in the associative arrays "FUG" and "KEITHLEY", generated by init functions, s. Fig.2:
devn . property = "value" structure:
FUG["fug"]["$active"] = 0
FUG["fug"]["$adress"] = "192.168.4.4:23"
FUG["fug"]["$fugtype"] = "FUG-NTV 100"
FUG["fug"]["$maxcurrent"] = 10
.

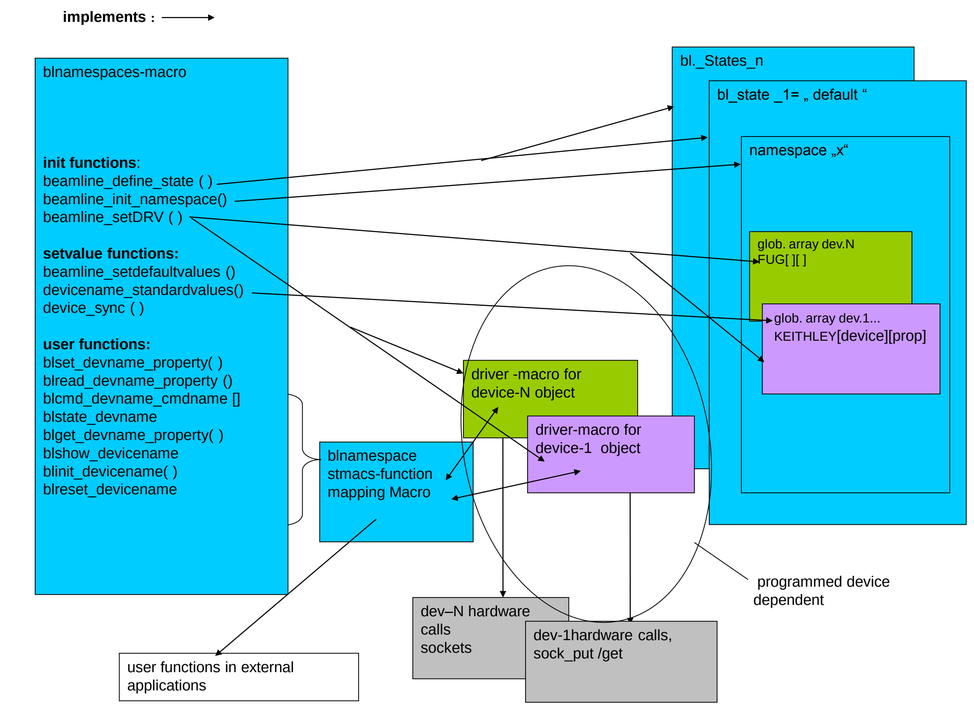

Fig. 2 Macro Package generated structures (blue), driver software to be programmed (green and cyan), SPEC-built in functions (grey), interface function calls (white), arrows mark 'implemented by function'.

Benefit

Two FUG devices, representing nine power supply 'objects' can be accessed by 11 (for the main power supply) and 73 (for the corrector power supply) standard-function calls obeying the naming rules introduced by the macro package.
Up to fifteen Hall-probes have to be addressed by 255 standard function calls for the Hall-probes plus three functions for the K2770.
The advantages using the object oriented approach is clearly visible, there is no need to write, a set of 84 nearly identical conventional SPEC-functions for power supplies and additional 255 functions to handle the output, in addition existing ANKA-beam line driver modules for motors can be used.

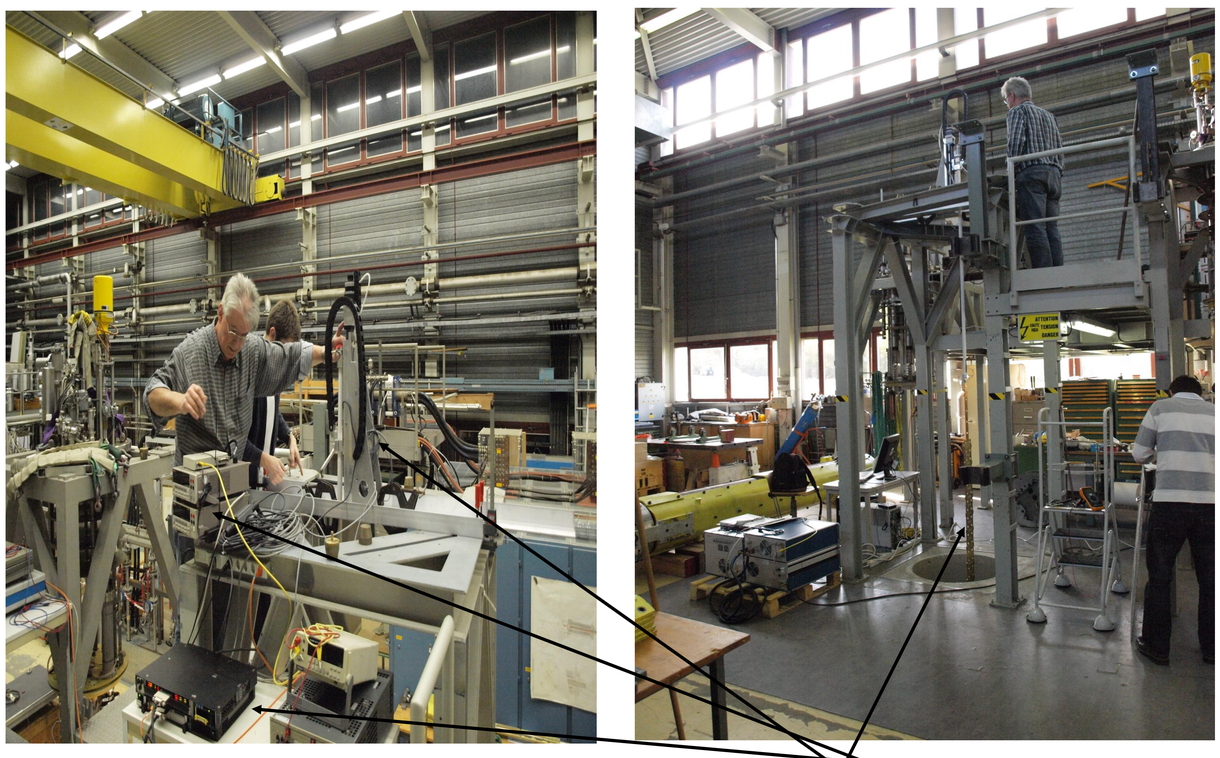

Fig. 3 Setup of B-Test Facility at CERN,TE-MCS site, Prevessin, October 2009. Right: motor, DMMs and linear motion controller for Hall-probes at test site. Left: cryogenics test bed for superconducting coils, the plate suspended in the duct supports the array of 15 Hall-probes which are moved vertically in the cryostat containing the superconducting coil (super conducting coil for new ANKA-insertion device was delivered later in 2009 directly from manufacturer).

CONCLUSION

The object oriented implementation, by use of existing beam line software modules, make the procedure straightforward since only the missing drivers for power supplies, digital multi-meters and the raw data evaluation algorithm, have to be introduced. But synergy proceeds, the FUG will be the power supplies of future insertion devices at ANKA, so the Software modul written to control its devices can easily be ported to the control system [4] of the next ANKA superconducting undulator.

REFERENCES

[1] *CASPER*- A magnetic measurement facility for superconducting undulators, E. Mashkina et al *2008 J. Phys.*: Conf. Ser. 97 012020
[2] www. certif. com, software SPEC
[3] Macro Package based Enhancement of SPEC controlled Experimental Setups, T. Spangenberg, K. Cerff, W. Mexner, Proceedings of PCaPAC2010, Canadian Light Source, Saskatoon, Canada, October 2010
[4] A modular control system based on ACS for present and future ANKA insertion devices, K. Cerff, W. Mexner, T. Spangenberg, M. Hagelstein, Proceedings of PCaPAC2008, Ljubljana, Slovenia, October 2008
[5] FUG power-supply, Probus, ADDAT30, command reference, V2.13