Overview of the new KOSMA 3m driveprogram software:

H. Hafok, Institute for Radio Astronomy, University Bonn

The following document gives a overview of the driveprogram software for the KOSMA 3m telescope. We want to create a LINUX based solution for operating the KOSMA 3m telescope which gives the possibility to use main parts of the software at SOFIA.
So far at KOSMA we have a VAX based system to operate the facility using receivers at 230/345/690 GHz and a LINUX based system running on the AOS control computer (AOSOBS, Stutzki et al.). Both incorporate telescope control and data acquisition in one program. Due to the fact that AOSOBS is already based on LINUX with modularisation concerning telescope tracking and astronomical calculations and interprocess / interprogram communication this will be the bases for the new control program.

The new program will consist of independent LINUX programs directly started via the bash shell.

- Start up start telescope system
- set master parameters (sets master parameters (chopper throw, frames per side, darks etc.)
- set line (searches for line in line catalogue and directs name and freq to file structure)
- set object (see line)
- set offsets
- load measurement
- comb measurement
- sky-dip (performs sky dip measurement)
- Total Power scan (on-off measurement)
- dual-beam-switch measurement
- on-the-fly measurement (scans on strip at the sky and dumps data to HD)
- continuum line scan

commands could be accessed directly or via shell scripts, e.g. mapping is done via shell scripts or GUI which call appropriate commands.

Observing routines control telescope via server modules:

AOS-SERVER: reads commands and controls the AOS-hardware and dumps the output.

ASTRA-SERVER : reads the desired telescope positions and controls the telescope.
(Sky chopper, Telescope dome, Load / Tertiary mirror)

SYNTHESIZER-SERVER: if a load measurement is performed or a command is given the server sends the new calculated synthesizer frequency to the synthesizer.

NTP-SERVER Time synchronization via NTP via local source GPS.

DISPLAY-SERVER

OFFLINE-KALIBRATION

Communication:

Communication is performed via files, which are store locally or on NFS mounted devices. These files contain variables in ASCII-structures which are written and read using KOSMA I/O-lib (Graf et al.)

Example: Communication to telescope control

```
#include <unistd.h>
#include "KOSMA_file_io.h"

int main()
{
  struct {
    STRING_VAR      c_sourcename;
    INT_VAR         c_coordsys;
    INT_VAR         c_rotcsys;
    INT_VAR         c_offcsys;
    INT_VAR         c_scanduration;
    STRING_VAR      c_trueanglam;
    STRING_VAR      c_trueangazi;
    STRING_VAR      c_obsmodus;
    DOUBLE_VAR      c_equinox;
    DOUBLE_VAR      c_lam;
    DOUBLE_VAR      c_bet;
    DOUBLE_VAR      c_dlam;
    DOUBLE_VAR      c_dbet;
    DOUBLE_VAR      c_olam;
    DOUBLE_VAR      c_obet;
    DOUBLE_VAR      c_vlam;
    DOUBLE_VAR      c_vbet;
    DOUBLE_VAR      c_srcang;
    DOUBLE_VAR      c_pmalpha;
    DOUBLE_VAR      c_pmdelta;
    DOUBLE_VAR      c_parallax;
    DOUBLE_VAR      c_radvel;
    DOUBLE_VAR      c_chopamp;
    DOUBLE_VAR      c_chopfreq;
    DOUBLE_VAR      c_oa;
    DOUBLE_VAR      c_oe;
    DOUBLE_VAR      c_ovazm;
    DOUBLE_VAR      c_ovelv;
    INT_VAR         c_cookie;
  } cat;

  struct {
    DOUBLE_VAR      a_elvs;
    INT_VAR         a_inpos;
    INT_VAR         a_cookie;
  } astra;
  double newdata;

  ErrorLogfile( "/opt/aos/KOSMA_file_io/log/astra_dummy.log" );
  DebugLevel( FATAL - INFO );
  SetErrorFormat( "" );

  /* set standard config file name */
  StandardConfigFile( __FILE__ );

  ConfigureString (&cat.c_sourcename, "c_sourcename", "", "");
  ConfigureInt (&cat.c_coordsys, "c_coordsys", "", "");
  ConfigureInt (&cat.c_rotcsys, "c_rotcsys", "", "");
  ConfigureInt (&cat.c_offcsys, "c_offcsys", "", "");
  ConfigureInt (&cat.c_scanduration, "c_scanduration", "", "");
  ConfigureString (&cat.c_trueanglam, "c_trueanglam", "", "");
  ConfigureString (&cat.c_trueangazi, "c_trueangazi", "", "");
  ConfigureString (&cat.c_obsmodus, "c_obsmodus", "", "");
  ConfigureDouble (&cat.c_equinox, "c_equinox", "", "");
  ConfigureDouble (&cat.c_lam, "c_lam", "", "");
  ConfigureDouble (&cat.c_bet, "c_bet", "", "");
  ConfigureDouble (&cat.c_dlam, "c_dlam", "", "");
  ConfigureDouble (&cat.c_dbet, "c_dbet", "", "");
  ConfigureDouble (&cat.c_olam, "c_olam", "", "");
  ConfigureDouble (&cat.c_obet, "c_obet", "", "");
  ConfigureDouble (&cat.c_vlam, "c_vlam", "", "");
  ConfigureDouble (&cat.c_vbet, "c_vbet", "", "");
  ConfigureDouble (&cat.c_pmalpha, "c_pmalpha", "", "");
  ConfigureDouble (&cat.c_pmdelta, "c_pmdelta", "", "");
  ConfigureDouble (&cat.c_parallax, "c_parallax", "", "");
  ConfigureDouble (&cat.c_radvel, "c_radvel", "", "");
  ConfigureDouble (&cat.c_srcang, "c_srcang", "", "");
  ConfigureInt    (&cat.c_cookie, "c_cookie", "", "");
  ConfigureDouble (&cat.c_chopamp, "c_chopamp", "", "");
  ConfigureDouble (&cat.c_oa, "c_oa", "", "");
  ConfigureDouble (&cat.c_oe, "c_oe", "", "");
  ConfigureDouble (&cat.c_ovazm, "c_ovazm", "", "");
  ConfigureDouble (&cat.c_ovelv, "c_ovelv", "", "");

  ConfigureDouble (&astra.a_elvs, "a_elvs", "", "");
  ConfigureInt (&astra.a_inpos, "a_inpos", "", "");
  ConfigureInt (&astra.a_cookie, "a_cookie", "", "");

  astra.a_inpos.value = -1;
  astra.a_cookie.value = -1;
  ReadVariablesVarSet( &cat.c_cookie.name );
  astra.a_elvs.value  = -10.0;
  WriteVariablesVarSet( &astra.a_elvs.name );

  while ( TRUE )
  {
     astra.a_inpos.value = 0;
     newdata = ReadVariablesVarSet( &cat.c_cookie.name );
     if ( newdata > 0 ) {
/*              printf( "Obs_Id %d %s \n",cat.c_coookie.value,cat.c_sourcename.value); */
                usleep( 1000000 );
                astra.a_inpos.value = 0; /* loose track on old target*/
                astra.a_elvs.value       = 21.23456;
                WriteVariablesVarSet( &astra.a_inpos.name );
                usleep( 1000000 );
                astra.a_inpos.value = 1; /* track new target */
                astra.a_cookie.value = cat.c_cookie.value;
                astra.a_elvs.value       = 60.12345;
                WriteVariablesVarSet( &astra.a_inpos.name );
                printf(
     "Obs_ID %d: Source %s %f %f %d, map offs %f %f  ref offs %f %f %d\n",
      cat.c_cookie.value,cat.c_sourcename.value, cat.c_lam.value, cat.c_bet.value,
      cat.c_coordsys.value,
      cat.c_dlam.value, cat.c_dbet.value, cat.c_olam.value,cat.c_obet.value,
      cat.c_offcsys.value);
                printf(
     "   chop %f, rotator %f, rot_coord %d, Elev %f, astra_dummy on track %d\n",
      cat.c_chopamp.value,cat.c_srcang.value,cat.c_rotcsys.value,astra.a_elvs.value,astra.a_inpos.value);
     }
     usleep( 100000 );
  }
  exit( 0 );
}
```

Configuration file:

```
! Catalog
""              c_sourcename            %s
KOSMA_catalog.in    KOSMA_catalog.in    !Name of the observed source
1               c_coordsys%12d
KOSMA_catalog.in    KOSMA_catalog.in    !Coordinate System eg. 1=J2000, 9=B1950
1               c_rotcsys   %12d
KOSMA_catalog.in    KOSMA_catalog.in    !Coordinate System rotator
1               c_offcsys   %12d
KOSMA_catalog.in    KOSMA_catalog.in    !Coordinate System off position
0               c_scanduration          %12d
KOSMA_catalog.in    KOSMA_catalog.in    !Scan duration [s]
YES             c_trueanglam            %s
KOSMA_catalog.in    KOSMA_catalog.in    !True angle for lambda
YES             c_trueangazi            %s
KOSMA_catalog.in    KOSMA_catalog.in    !True angle for azimuth
O               c_obsmodus              %s
KOSMA_catalog.in    KOSMA_catalog.in    !Observe Mode (to be specified ...)
0               c_srcang    %12.8g
KOSMA_catalog.in    KOSMA_catalog.in    !anticlockwise angle of focal plane [degree]
2000.0          c_equinox %12.1f
KOSMA_catalog.in    KOSMA_catalog.in    !refer all coordinates to a specific equinox
0               c_lam                   %12.10g
KOSMA_catalog.in    KOSMA_catalog.in    !first coordinate [degree]
0               c_bet                   %12.10g
KOSMA_catalog.in    KOSMA_catalog.in    !second coordinate [degree]
0               c_vlam                  %12.10g
KOSMA_catalog.in    KOSMA_catalog.in    !first coordinate angle velocity [arcsec/s]
0               c_vbet                  %12.10g
KOSMA_catalog.in    KOSMA_catalog.in    !second coordinate angle velocity [arcsec/s]
0               c_dlam                  %12.10g
KOSMA_catalog.in    KOSMA_catalog.in    !first coordinate delta position [arcsec]
0               c_dbet                  %12.10g
KOSMA_catalog.in    KOSMA_catalog.in    !second coordinate delta position [arcsec]
0               c_olam                  %12.10g
KOSMA_catalog.in    KOSMA_catalog.in    !first coordinate off position [arcsec]
0               c_obet                  %12.10g
KOSMA_catalog.in    KOSMA_catalog.in    !second coordinate off position [arcsec]
0               c_pmalpha%12.8g
KOSMA_catalog.in    KOSMA_catalog.in    !proper motion [arcsec/yr]
0               c_pmdelta %12.8g
KOSMA_catalog.in    KOSMA_catalog.in    !proper motion [arcsec/yr]
0               c_parallax %12.8g
KOSMA_catalog.in    KOSMA_catalog.in    !Parallax [arcsec]
0               c_radvel    %12.8g
KOSMA_catalog.in    KOSMA_catalog.in    !Radial Velocity [km/s, +ve moving away]
0               c_chopamp               %12g
KOSMA_catalog.in    KOSMA_catalog.in    !Wobbler amplitude [arcsec]
0               c_oa                    %12.10g
KOSMA_catalog.in    KOSMA_catalog.in    !Offset azimuth [arcsec]
0               c_oe                    %12.10g
KOSMA_catalog.in    KOSMA_catalog.in    !Offset elevation [arcsec]
0               c_ovazm                 %12.10g
KOSMA_catalog.in    KOSMA_catalog.in    !Offset scan velocity [arcsec/s]
0               c_ovelv                 %12.10g
KOSMA_catalog.in    KOSMA_catalog.in    !Offset scan velocity [arcsec/s]
0               c_cookie    %12d
KOSMA_catalog.in    KOSMA_catalog.in    !Catalog Cookie
!
! Astra

0       a_inpos         %12d
NULL                KOSMA_astra.status      ! -1 not reachable, 0 not in position, 1 position reached
0       a_cookie        %12d
NULL                KOSMA_astra.status      ! Timestamp(from c_cookie)
0       a_elvs          %12.8g
NULL                KOSMA_astra.status      ! Soll Elevation Position [degree]
```