LCC main characteristics

- operating environment as basis for LCU applications
- provides solutions to requirements common to all applications
- simplifies the use of the LCU (all LCUs having the same appearance)
- simplifies the design of the LCU application (basic software already provided, does not have to be reinvented for each LCU, all LCUs having the same basic software interface)
- increases the reliability (basic software well debugged)
- simplifies the integration of subsystems (basic software interfaces known and well tested)
- simplifies the maintenance

LCC characteristics

- LCC is to a large extent compatible with CCS both programmatic interface and the internal structures.
- provides a complete interface with CCS on the workstation

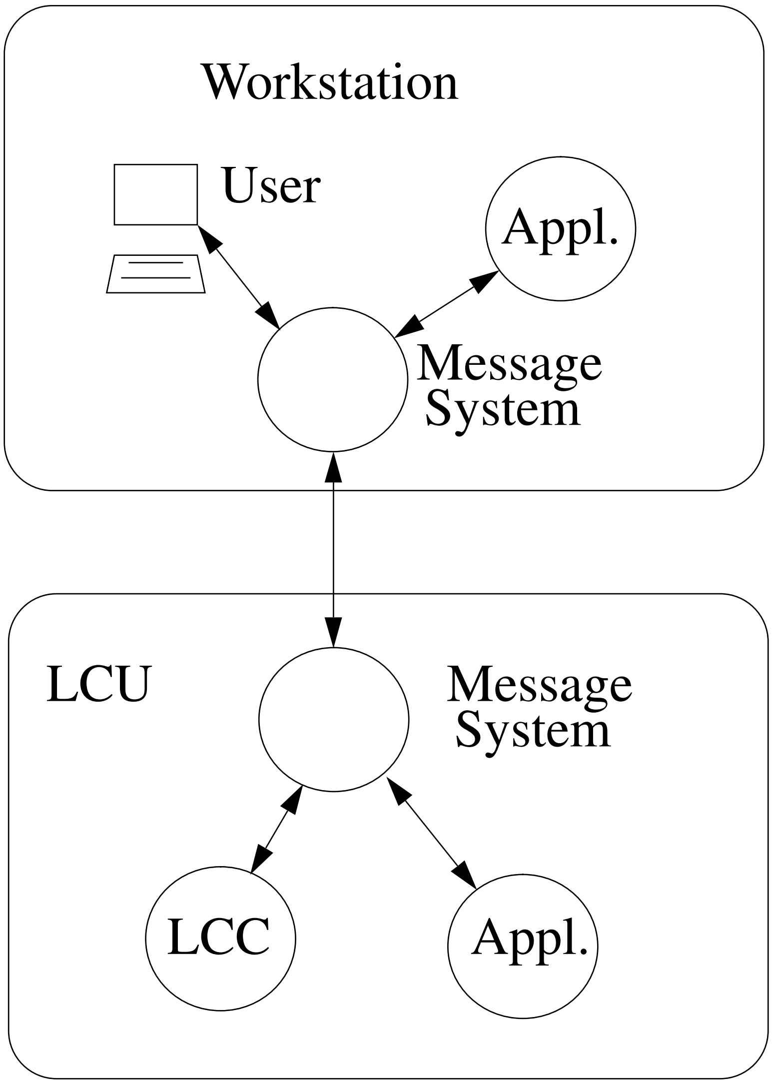


Figure 1: LCU Common Software

LCU Common Software

The main components of LCC are:

- Local database
- Interfaces with CCS
    - message system
    - logging system
    - error system
    - scan system
    - access control
- I/O system
- Time services
- Command Interpreter
- Start-up and shut-down procedures
- Simulation
- Drivers for ESO standards boards
- Motor Control module
- Engineering test tools

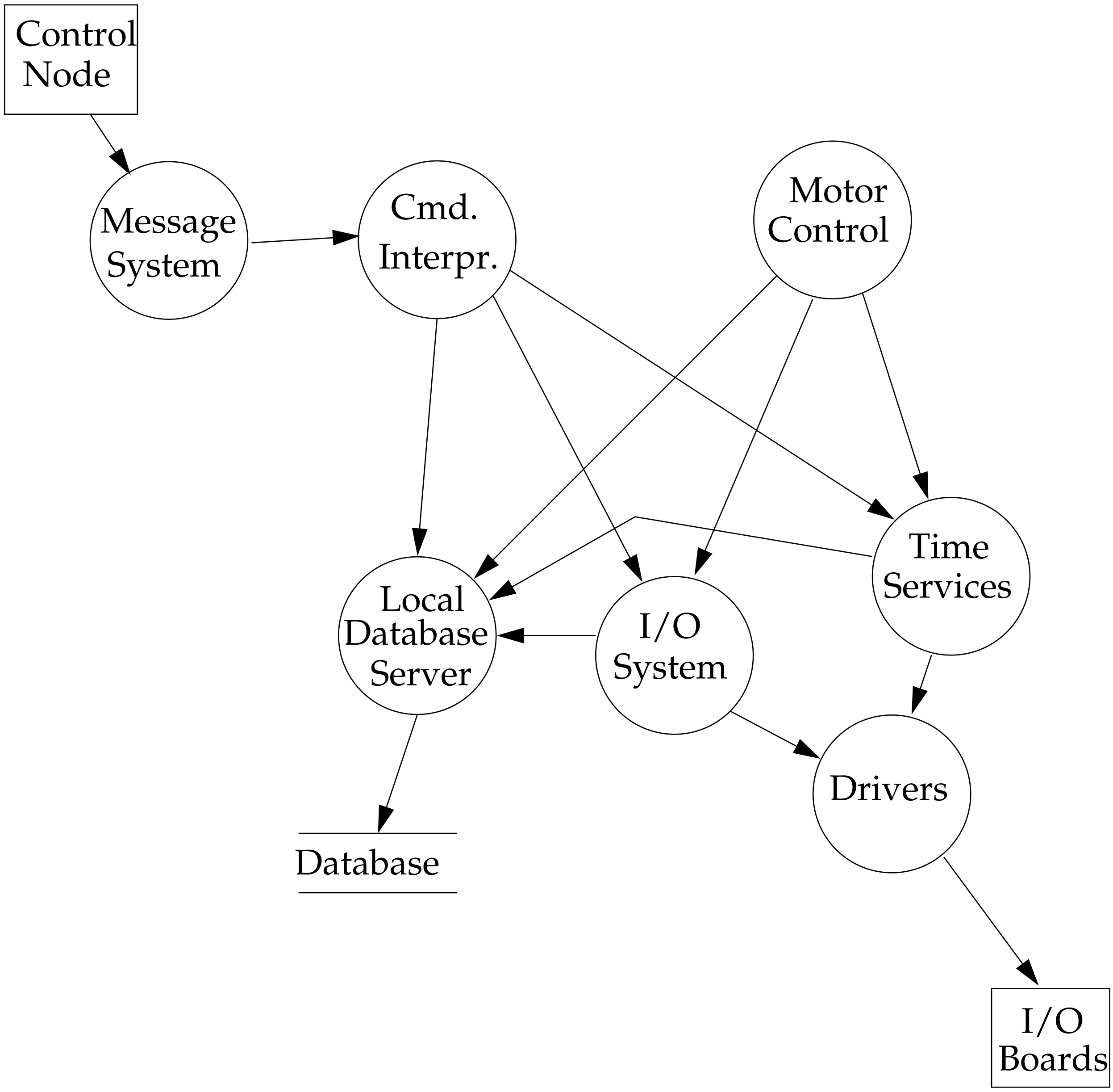


Figure 2: LCU Common Software -LCU part

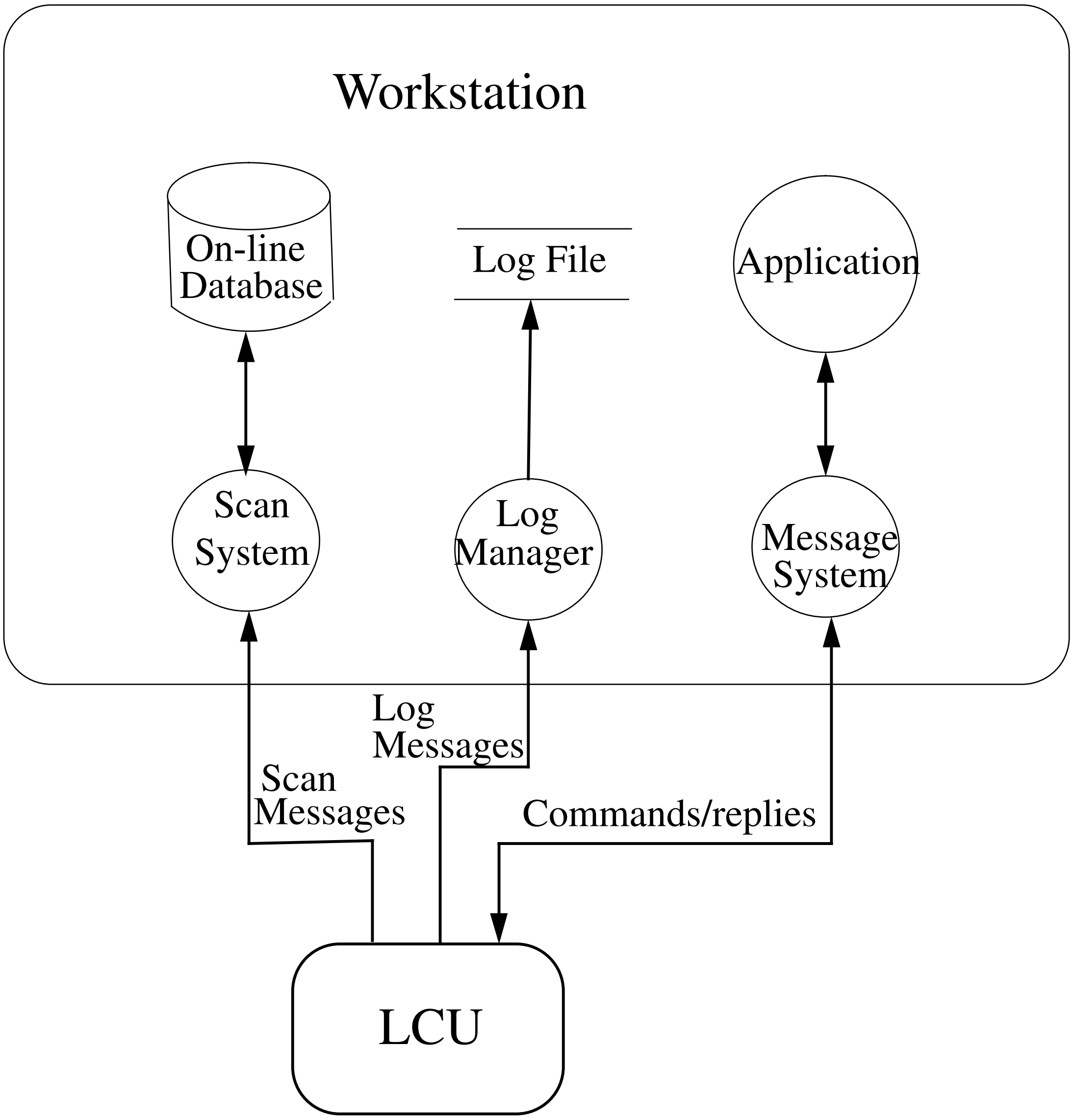


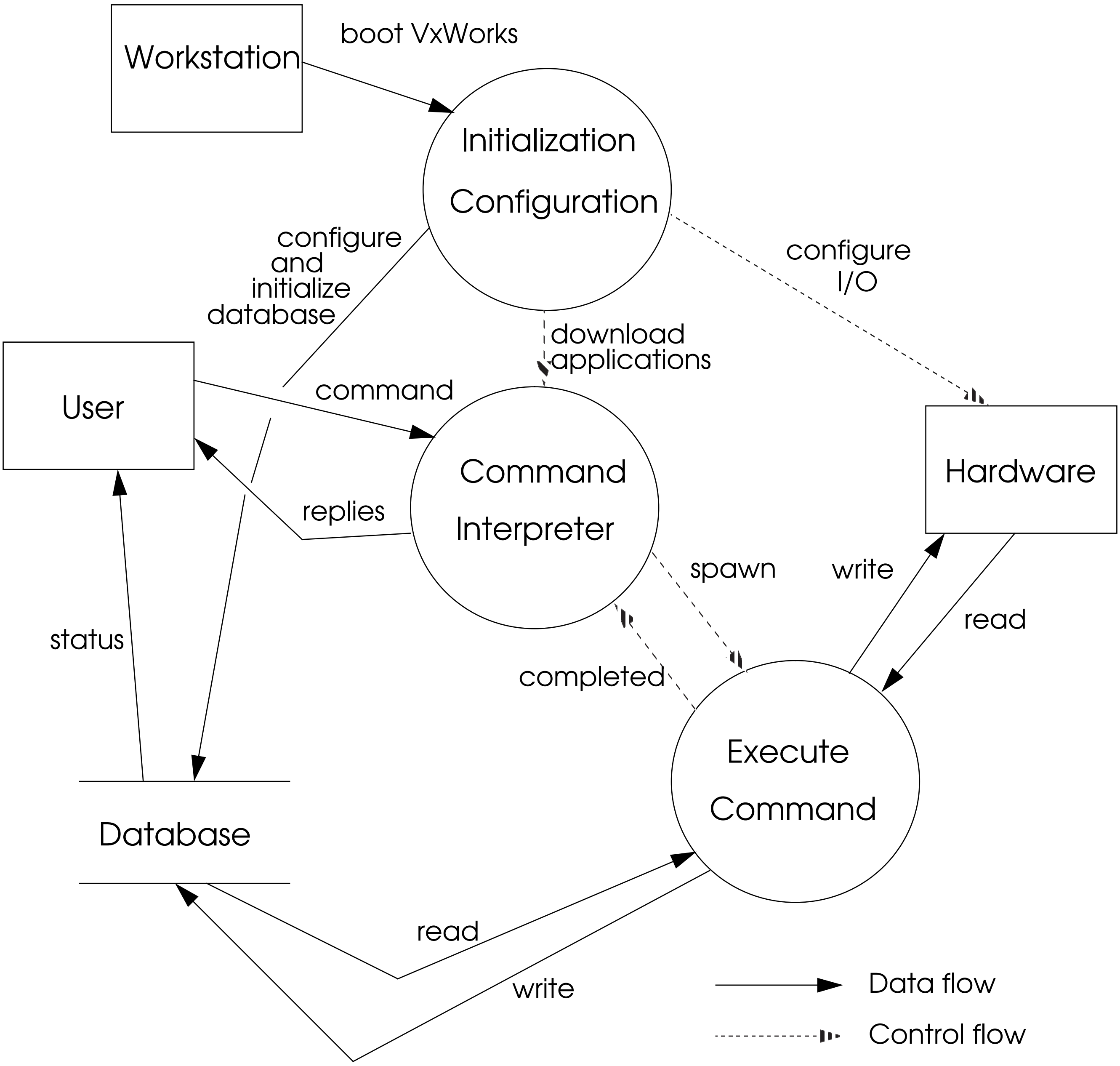
Figure 3: Data flow between CCS and LCC

LCC Software

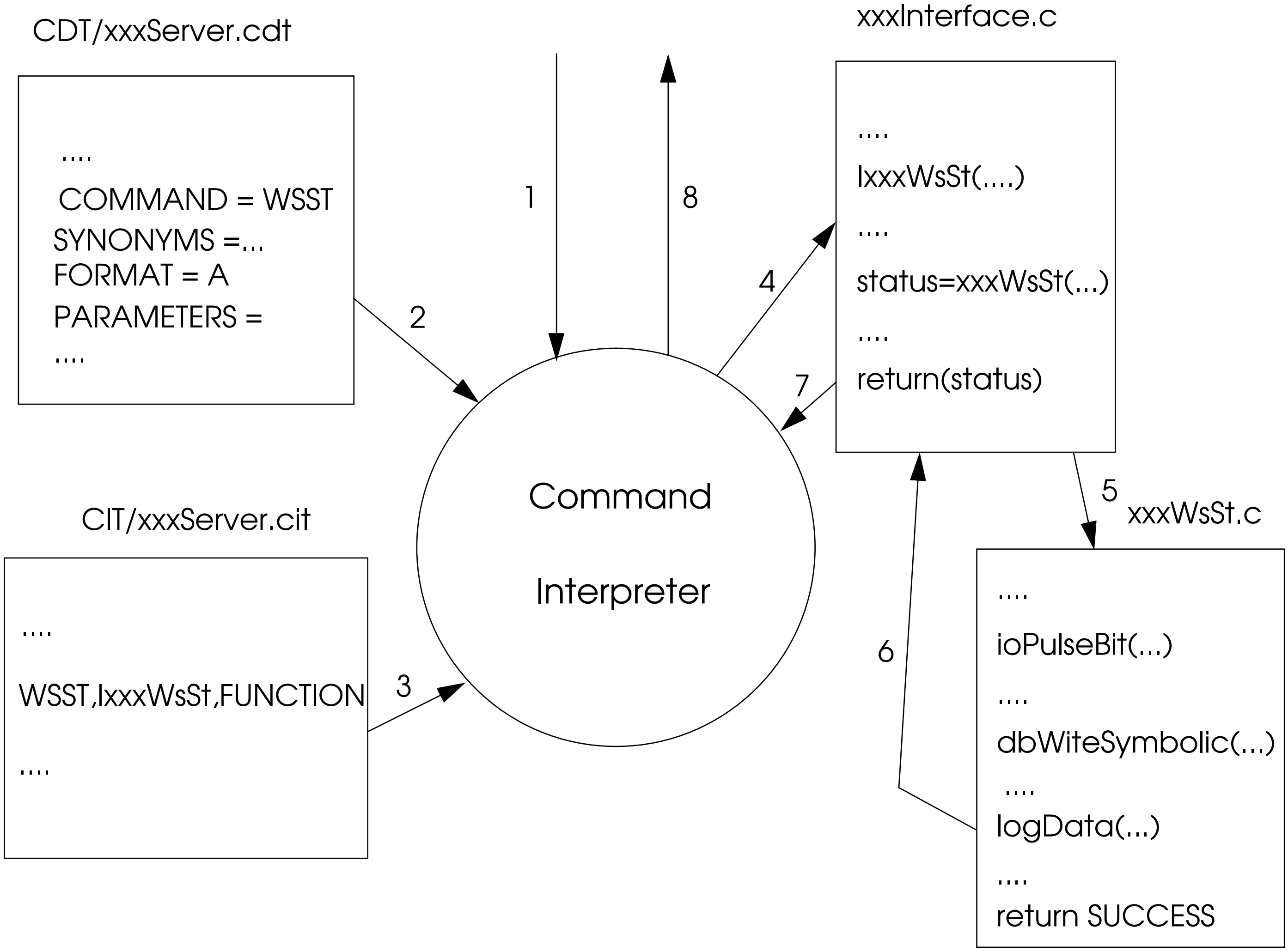
ESO developments added to LCC kernel:

– LCU logging task	lculog
– driver common libraries	lcudrv
– LCU boot support module	lcuboot
– digital I/O driver	acro
– analog I/O driver	aio
– transputer interface board	b016
– encoder interface driver	ikon
– serial interface driver	iser
– ethernet interface	net01
– reflective memory	rmn
– time interface driver	tim
– LCU Q-server	lqs
– Q-server emulator	qsemu
– Servo amplifier driver	ampl
– Maccon motor contr.	mcon
– Motor Control Module	mcm
– engineering test software	inducers
– motor test software	motei
– configuration software	vcc

Command Interpreter



VLT Software - LCU Common Software (LCC)



1. Command Interpreter receives command message with command WSST
2. Existence and syntax is checked in command definition table
3. Command Interpreter Table is read to find out what to do for this command
4. Command Interpreter calls function lxxxWsSt as a function
5. lxxxWsSt calls xxxWsSt
6. xxWsSt access hardware, database and makes a log entry. Returns SUCCESS
7. lxxxWsSt returns SUCCESS to the command interpreter
8. Command Interpreter sends a reply to the originator

Guidelines for VLTApplication Software

The guidelines are intended for developers of VLT application software being

- External contractors
- Cooperating institutes
- Internal developers

To get a homogeneous VLT software ESO has set up a number of rules how to develop LCU software:

- Use and storage of data
- Communication
- Standard commands for applications
- Standard driver concept
- Failure mode operation including
    - failure modes
    - error reporting
    - recovery
    - safety

Guidelines (continued)

Maintenance concept

- monitoring
- self-test
    - HW checking
    - move functions
    - complete test
- simulation
    - device level
    - LCU level
    - WS simulation
- tuning
- preventive maintenance
- corrective maintenance

Standard Commands

Mandatory standards commands on the LCU:

INIT
ONLINE
STANDBY
OFF
STATE
STATUS
STOP
SELFTEST
TEST
SIMULAT
STOPSIM
VERBOSE
VERSION
EXIT
KILL
BRAKE

Standard States

State / Switching function	Loaded	Stand-By	On-Line
INIT	x		
STANDBY			
ONLINE			
STOP	x	x	x
OFF			

Application Guidelines

- decomposition of an application
- use of command interpreter
- states, state transitions
- standard commands
- start-up
- communication
- database
- ECCS - EVH
- error handling
- logging
- signals
- time handling
- scan system
- simulation
- configuration
- debugging tools

Guidelines concept

- Provides examples how to use basic concepts
- Gives a deeper explanation than the user manuals
- Provides concepts to be followed in the design

More examples are also found in the user manuals
Templates provided with the VLT software. Example citmp

Porting to PowerPC

- first include file “vxWorks.h”
- memory alignment, avoid casting
- database table mapping
- interrupts (INUM_TO_IVEC)
- POSIX compliance
- global symbol table
- task priorities
- compiler CPU dependency

More in LCC relase notes of OCT98-release

The LCU Task

- Start from “citmp”
- Change name to “lcumod”
- Create default environment “lted”
- Add modules “too” and “lcumod” + process “lcumodServer”
- Add “LCUMOD.db” in “DATABASE.db”
- Remake db
- Modify devicesFile (from template)

Test application

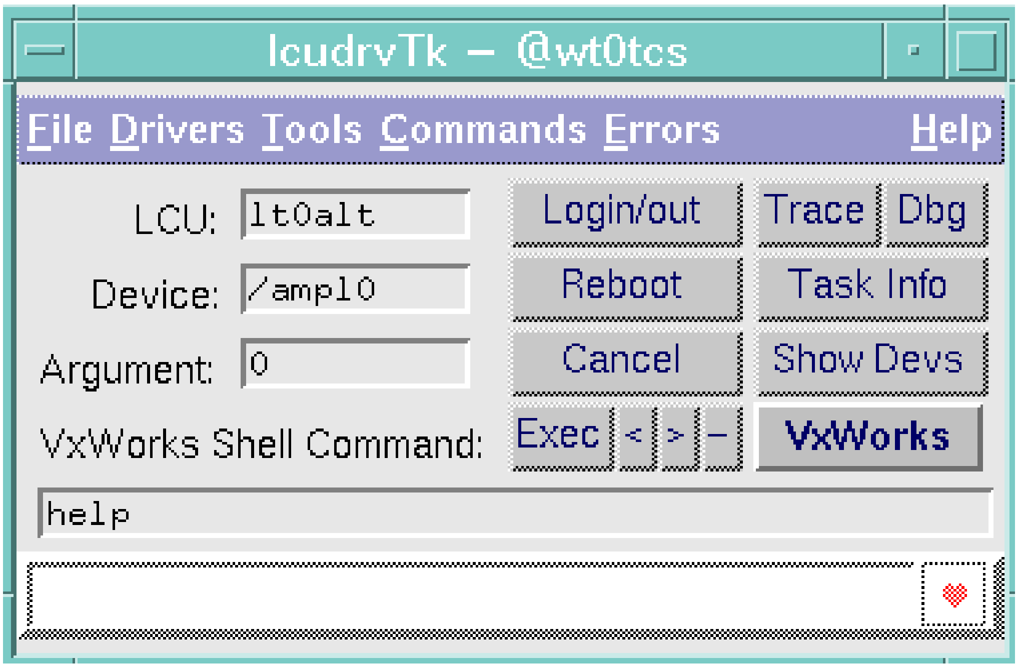
- Send standard commands (STATE, INIT, SETDB, GETDB)
- Monitor database (PARAMS:DBLCUMOD.dbValue)

Add Application

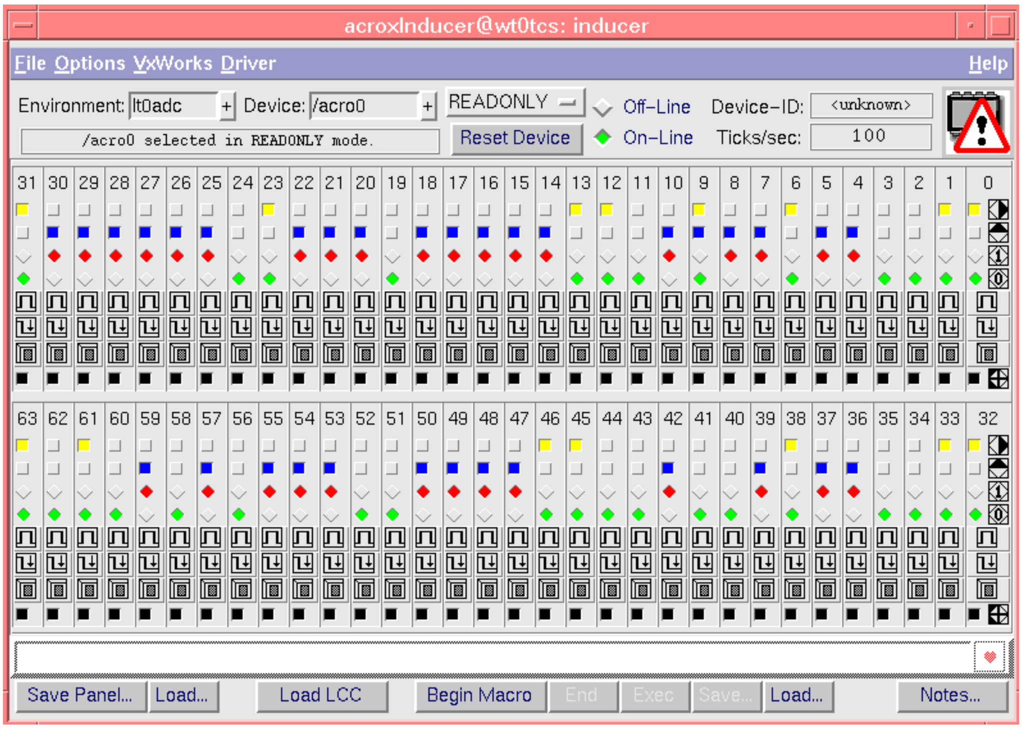
- Write new functions, lcumodMoveMotor.c, lcumodGetStatus.c
- Add commands MOVEABS, STATUS to CDT and CIT file
- Add interface functions

Engineering Panels

lcudrvTk



acroxInducer



aioxInducer