IOC Redundancy: Integration and Tuning

EPICS Meeting – EPICS Core

DESY, April 27, 2007

Gongfa Liu, DESY / MKS-2

Outline

- **Design Goal**
- **System Overview**
- **Process and Interface of RMT**
- **Integrate PRRs into RMT with Driver IF**
- **Summary**

❖ ***abbr.***

- *RMT: Redundancy Monitor Task*
- *PRR: Primary Redundancy Resource*
- *Driver IF: Driver Interface*

Design Goal

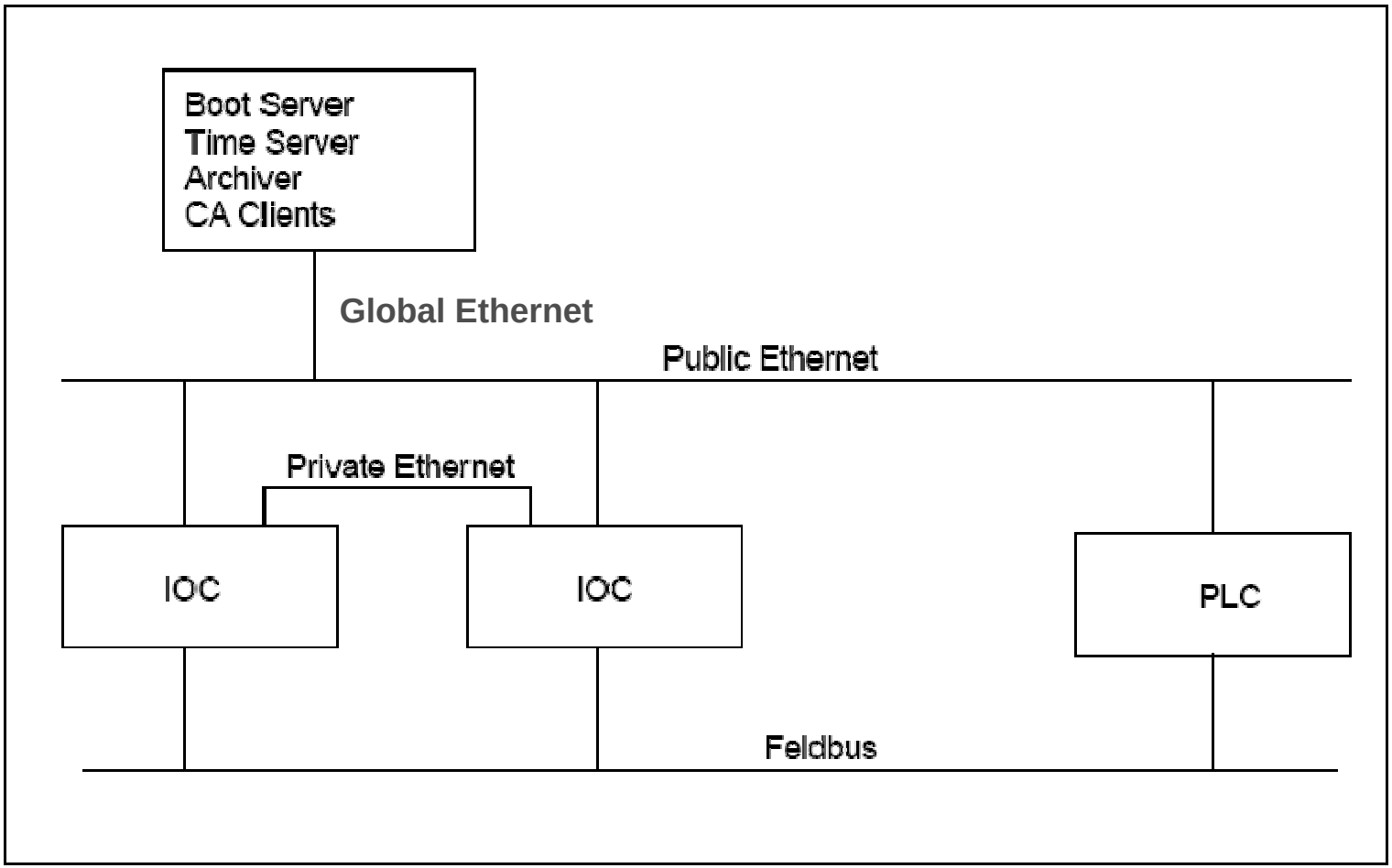
From the preamble of the design specification:

… last and most importantly one major design goal must be matched:

Any redundant implementation must make the system more reliable than the non redundant one. …

- Design Goal
- **System Overview**
- Process and Interface of RMT
- Integrate PRRs into RMT with Driver IF
- Summary

Hardware Layout of a Redundant IOC System



Software Components

- **RMT(Redundancy Monitor Task):**
  - Establish and maintain communications with the partner and the drivers.
  - With the information from these resources and information from the operator, it makes decisions about assuming or giving up control.
- **CCE(Continuous Control Executive):**
  - Keep the database of the primary and backup synchronized.
- **SNL Executive:**
  - Keep the state programs of the primary and backup synchronized.

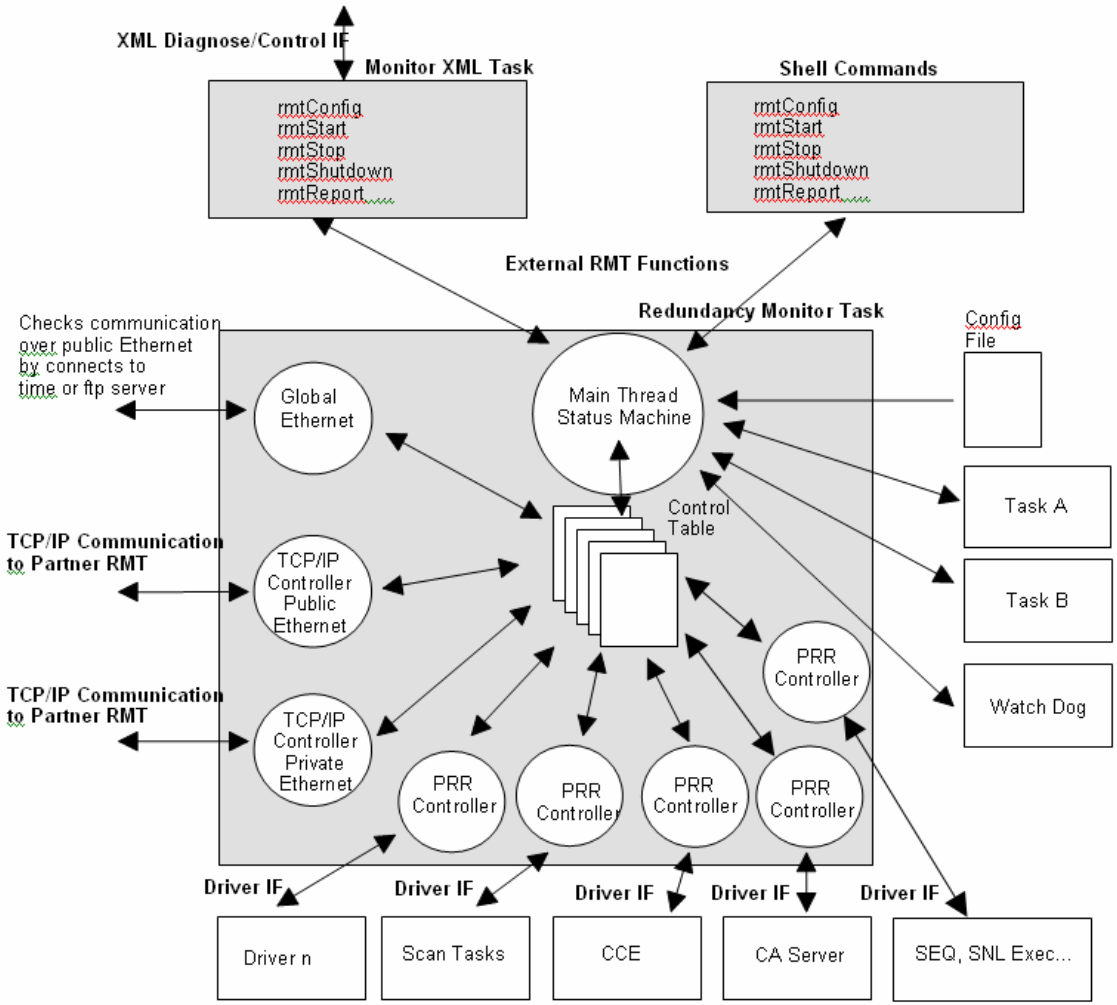
★ **RMT and CCE are subcontracted to two companies.**

Prototype System

- **Hardware:**
  2 SMA CompactPCI (CompactMAX CPU7.2 M)
- **Software:**
  vxWorks 5.5 + EPICS base 3.14.8.2 + seq 2.0.11

- Design Goal
- System Overview
- **Process and Interface of RMT**
- Integrate PRRs into RMT with Driver IF
- Summary

Process and Interface of RMT



List of monitored PRRs

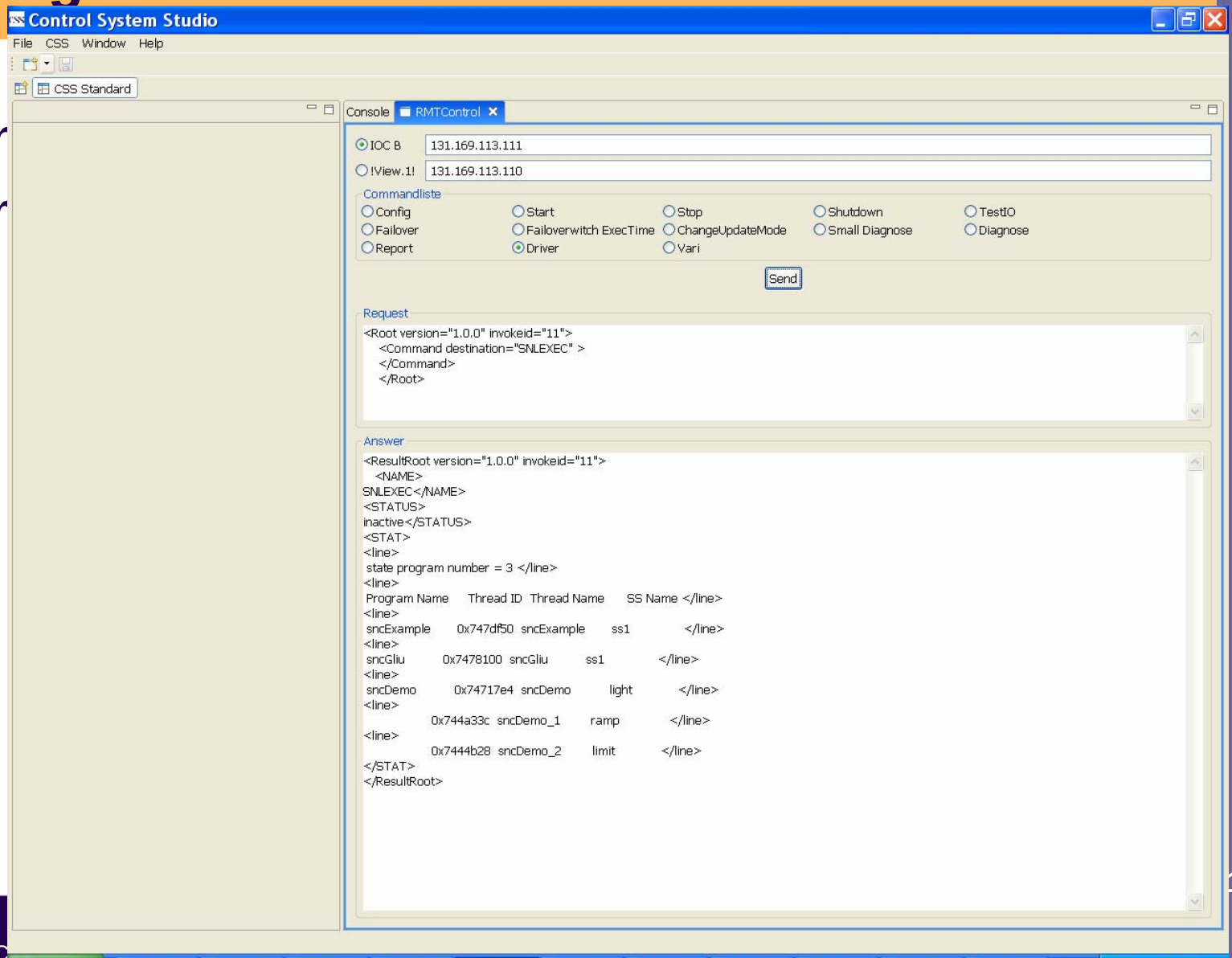
- Public, Private, Global Ethernet,
- System Tasks,
- Watchdog,
- Device Driver,
- CA Server,
- Scan tasks, CCE,
- SEQ, SNL Executive, ...

Driver IF (Driver Interface)

- It is an identical interface between PRR and RMT.
- It is implemented as functions defined in the PRR and callable by the RMT. The addresses of these functions is in an entry table of the PRR.
- It is designed by *Bernd Schoeneburg.*

XML Diagnosis

- It is ir
- It is ir



- Design Goal
- System Overview
- Process and Interface of RMT
- **Integrate PRRs into RMT with Driver IF**
- Summary

The Process of Integrating PRRs into RMT with Driver IF

1) Share a header file “rmtDrvIf.h” with RMT.
2) Define the private data structure for internal use.
3) Define the components of the entry table.
4) Register the PRR with the function rmtRegister().

List of registered PRRs

- drvTest
- Scan tasks
- SEQ
- CA Server
- CCE
- SNL Executive

drvTest/ Scan tasks/ SEQ

- **drvTest**
  - A test IO driver written by *Bernd Schoeneburg*.
  - Simulate the different behaviours of the IO driver with a parameter.
- **Scan tasks**
  - Register each periodic scan task.
  - When Slave, makes scan task inactive; otherwise active.
- **SEQ**
  - When Slave, makes sequencer inactive; otherwise active.

CA Server

- **Task:**
  - when Slave, destroys client connections.
  - when Master, accepts client connections.
- **Two types of CA Servers:**
  - RSRV: Server for IOCs and Soft IOCs.
  - CAS: Channel Access Server or Portable Server.
- **Spawned tasks at IOC**
  - CAS-TCP, CAS-UDP, CAS-beacon: at initialization
  - CAS-client, CAS-event: at a client connection
- **Implementation:**
  - When Slave, makes “CAS-TCP” , “CAS-beacon” and “CAS-UDP” sleeping with a flag, and deletes “CAS-client” and “CAS-event” (client->disconnent = TRUE); otherwise, makes these tasks normal.

CCE/SNL Executive

- **CCE**
  - Task: keep the database synchronized.
  - Construct internal data structures:

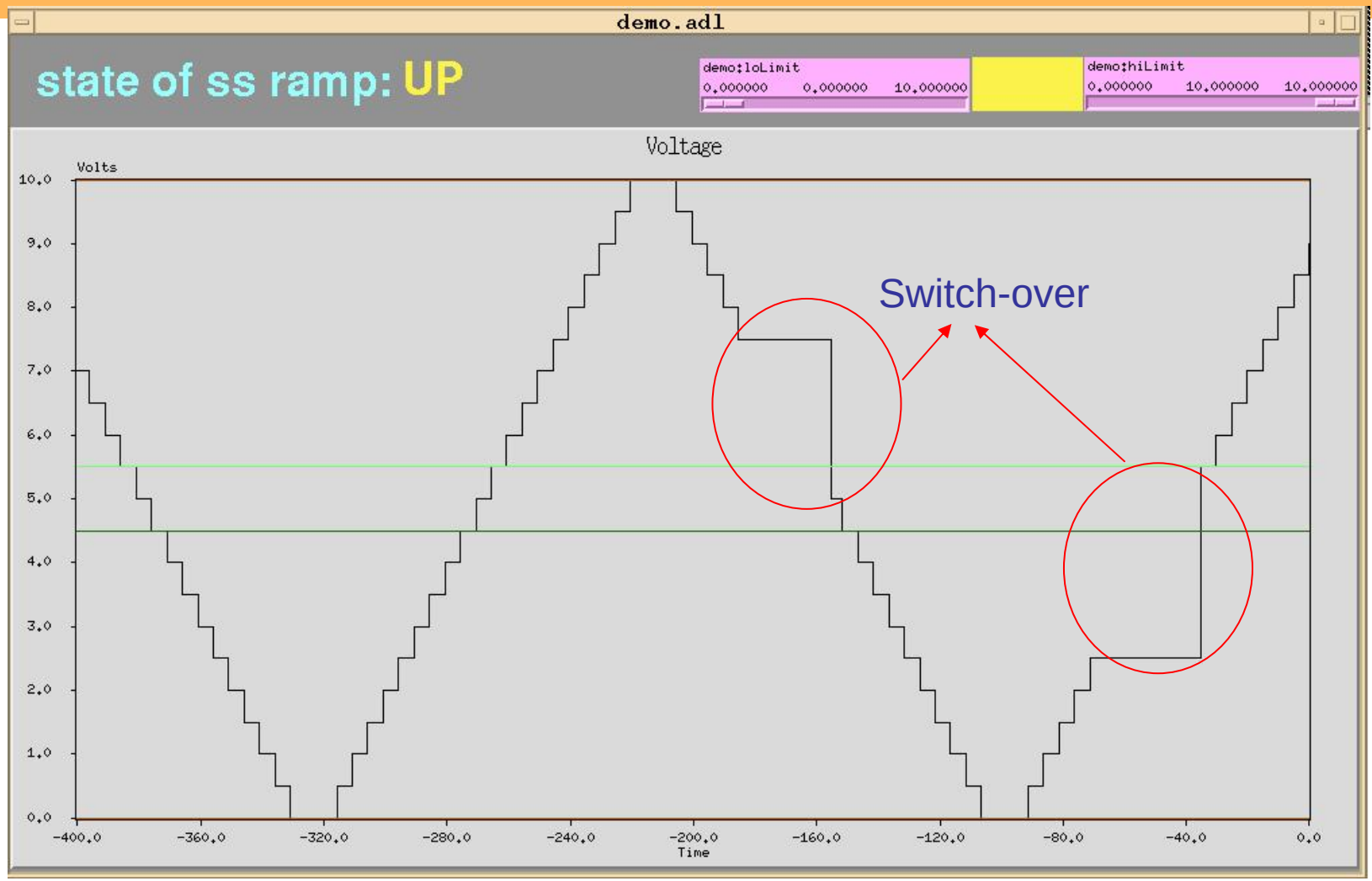
    Record Blocks, Field Blocks, Partner Record Blocks
  - Periodically synchronizes the database through the Private Ethernet.
- **SNL Executive**
  - Task: keep the state programs synchronized.
  - Construct the internal data structure.

    seqTraverseProg( ) is used to access the state program data
  - Periodically synchronizes the state programs through the Private Ethernet.

List of modified files concerning EPICS source code

1) **Scan tasks**
   - base/src/db/dbScan.c
2) **CA Server**
   - base/src/rsrv/caservertask.c
   - base/src/rsrv/camasgtask.c
   - base/src/rsrv/online_notify.c
   - base/src/rsrv/cast_server.c
3) **CCE**
   - base/src/dbStatic/dbBase.h
   - base/src/dbStatic/dbLexRoutines.c
   - base/src/db/dbAccess.c
   - base/src/rec/aiRecord.dbd
4) **SEQ**
   - seq-2.0.11/src/seq/*seq_task.c*

Test


demo.adl
state of ss ramp: UP
demo:loLimit
0.000000 0.000000 10.000000
demo:hiLimit
0.000000 10.000000 10.000000
Voltage
Volts
10.0
9.0
8.0
7.0
6.0
5.0
4.0
3.0
2.0
1.0
0.0
-400.0 -360.0 -320.0 -280.0 -240.0 -200.0 -160.0 -120.0 -80.0 -40.0 0.0
Time
Switch-over

- Design Goal
- System Overview
- Process and Interface of RMT
- Integrate PRRs into RMT with Driver IF
- **Summary**

Summary

- The prototype system of redundant IOC is integrated.
- The basic functions are realized.

Plans / Outlook

- SNL editor based on Eclipse – CDT (‘C’/ ‘C++’) development tool
- SNL debugger – make use of SLAC implementation

- Production in autumn 2007

Thank you!

The example of private data structure

```
typedef struct {
        BOOL            run;            /* driver is doing IO */
        BOOL            writeAccess;    /* driver has bus mastership */
        BOOL            testActive;     /* IO test is running */
        BOOL            inSync;         /* up-to-date with partner */
        BOOL            monitor;        /* continous updating of vars active */
        BOOL            updateBusy;     /* copying of data from partner is not finished */
        BOOL            tmo;            /* timeout of callback */
        BOOL            activeFlag;     /* flag to check if IO is running ok */
        int             instance;       /* instance Number given at start time */
        const char*     instanceName;
        RMTCALLBACK     pcallback;      /* address of rmt callback function */
        int             rmtId;          /* id number used by rmt callback */
        testResultType  testResult;
        errorType       error;
        int             tid;            /* task id of driver instance */
        WDOG_ID         wd;             /* watchdog to simulate test */
        WDOG_ID         wd2;            /* watchdog to simulate update */
        int             tickStart;      /* sys tick count when ioTest started */
        int             releaseTimeout; /* to hold time until releasing writeAccess after test */
} canPrivateType;
```

The entry table of Driver IF

```
typedef STATUS  (*RMTSUPFUN)();

typedef struct {
        const char          *type;
        const char          *instanceName;
        ushort               testTimeTypical;
        void                *pPrvt;
        RMTSUPFUN            pStart;
        RMTSUPFUN            pStop;
        RMTSUPFUN            pTestIO;
        RMTSUPFUN            pGetStatus;
        RMTSUPFUN            pShutdown;
        RMTSUPFUN            pGetInfo;
        RMTSUPFUN            pGetUpdate;
        RMTSUPFUN            pStartUpdate;
        RMTSUPFUN            pStopUpdate;
} rmtEntryTabType;
```

rmtInfo and rmtRegister()

```
typedef struct {
        const char      *partnerIPPrivate;
        short            preferredMaster;
} rmtInfoTabType;

STATUS  rmtRegister(rmtEntryTabType  *prmtEntryTab,
                    const rmtInfoTabType    **pprmtInfoTab);
```

Hardware Components

- **Two IOCs:**
  - One IOC is active (master)
  - The other is passive (slave)
- **Ethernet:**
  - Public Ethernet, Primary Link
  - Private Ethernet, Backup Link
  - Global Ethernet, monitor the higher-ranked network