# 1.0 OVERVIEW

The MSCP driver is one of the intermediate levels of support for the CI-20. The driver is responsible for coordinating all access between PHYSIO and MSCP disks. It must communicate to the HSC50 disks using the SCA and MSCP protocols.

The MSCP driver is part of the PHYSIO system. This is necessary to interface the operating system with the MSCP disks. The MSCP driver will be a Kontroller in the PHYSIO system and will be the inferior to the KLIPA Controller.

This document describes the structure of the PHYMSC driver, the specific services it performs and its relationship to the other CI-related services and to the monitor in general.

## 1.1 References

1. Systems Communication Architecture, 20-Jul-84

   Rev 4 (Strecker)

2. Scampi Functional spec (Dunn)

3. TOPS-20 Coding Standard, 23-Mar-83 (Murphy)

4. Mass Storage Control Protocol Version 1.2 (Gardner)

5. PHYMSC Functional Specification (Mclean)

## 1.2 Purpose Of This Document

This document describes how PHYMSC implements the services described in the functional specification [5]. Ideally, this document is a template for the implementor to follow and therefore will serve as a "road map" to the code.

Portions of this document will appear as commentary in the source module so that maintainers will benefit from this design plan.

## 1.3 Driver Functions

The PHYMSC driver is a TOPS-20 device driver. It is generally a passive service that responds to requests from higher and lower level components and simply performs the device-specific operations required by the requests. The only operations it performs on its own are those of the poller. However, these operations are transparent to the rest of the monitor.

The driver is responsible for maintaining the interface between the MSCP disks and the PHYSIO system. That is the driver must locate all the remote disks when they come online and it must recognize their characteristics and create an interface to PHYSIO. It is also the responsibility of the driver to determine that those remote servers are functioning and to try and reload them if there is a failure.

The only active part of the driver is the polling operation and the process of opening initial connections to remote servers. All other data transfers, datagrams and messages are initiated by PHYSIO and SCA.

## 1.4 Driver Components And Data Structures

The Kontroller block (KDB). This is a communication area for PHYSIO. This block contains the specific data for each MSCP server to which PHYMSC has recognized. This block is identical to the standard PHYSIO KDB.

The Unit Data Block (UDB). This is a communication area for PHYSIO. This block contains the specific data for each disk unit which PHYSIO will recognize. This block is identical to the standard PHYSIO UDB.

The fundamental pieces of the driver are:

1. The poller. This is the code that attempts to recognize new servers when they appear on the CI and to detect

failures of these servers to respond. It is also expected to reload those servers if they consistently fail to respond to a Get Command Status request. The poller is called as a part of PHYSIO poller and therefore is periodically called by PHYKLP.

2. Interrupt service. This code is called by SCA in the form of SCA Callbacks. These are events that happen asynchronously and are usually a result of a previous request for service to SCA. The interrupt service routine is responsible for completion of I/O and the process of placing a disk online.

3. PHYSIO interface. These are a set of routines to service the Start I/O requests and other responsibilities of a Kontroller.

The remaining sections of this document will explore each of the pieces described in the overview as well as provide an operational description of PHYMSC.

## 2.0 LOCAL DATA DESCRIPTIONS

### 2.1 MSCCID

This table is used to keep track of the current Connect Id of each connection. This table has three states. Zero indicates an unused entry. Minus one indicates an entry that is no longer connected. Other values are the Connect Id of the current connection.

### 2.2 MSCOLD

This is a table of old Connect Id values. It is mainly for debugging purposes.

## 2.3 CIDATA

This is a table of the status of each entry in MSCCID. It contains the state of the connection during initalization and after initialization the status of the connection.

## 2.4 CICMST

Status of the oldest command for each connection. This is the status returned from the server. If it is the same on each polling cycle then the server is assumed to have died.

# 3.0 GLOBAL DATA USAGE

## 3.1 System Block

The system block is used to find the related QOR request blocks and to find the port and node numbers.

## 3.2 BHD/BSD

The BHD and BSD's are used to describe the I/O transfers at Start I/O.

## 3.3 QOR

The QOR list is used to permit Start I/O to closely control the state of requests to the HSC. This permits Start I/O to re-queue the requests when the drive becomes offline.

# 4.0 PHYMSC DESCRIPTION/OPERATION

## 4.1 PHYSIO Interface

PHYMSC is a device driver. As such it is called by PHYKLP to perform device polling, Start I/O and process interrupts.

PHYMSC creates the UDB and KDB tables that PHYSIO expects. These tables are in the same format as any other UDB and KDB in the PHYSIO system.

The poller is called from PHYKLP as a part of the PHYSIO poller.

## 4.2 Initialization

The PHYMSC driver is started by a call from PHYKLP. The initialization process starts by finding the configuration of each node on the CI. if a node of type KL or HSC is found then PHYMSC performs the following sequence.

1. Attempt to connect to the remote system. From here on the connect Initialization sequence is interrupt driven.

2. Loop back trying to find out the configuration of the next node.

Interrupt driven connect sequence:

1. Connect response available SCA Callback occurs. PHYMSC checks to see if the connect is accepted and then save the Connect Id if it is successful.

2. Connect Response available causes a request to Set Characteristics.

3. The SCA callback from the Set Characteristics permits the determination of the type of device on the remote server. If the remote device is not a 576 byte format disk it is ignored, otherwise, a Get Next Unit request is made to determine what units are on the HSC. The UDB is not created for other than 576 disk types. A BUGINF is created once for such an occurence.

4. When the SCA callback from the Get Next Unit occurs the disk characteristics are obtained. These are used to generate a UDB and the Disk is onlined with an ONLINE request.

5. When the SCA callback from the Online occurs the geometry of the disk is determined. The home blocks are flagged as needing to be checked and the initialization process is complete.

## 4.3 Poller

Poller functions:

1. When time and date become available to a new node a Set Characteristics is sent with time and date information to the remote system.

2. Get Command Status requests are made to the remote system for the oldest command on the queue. This is done to permit the determination of the state of the remote system.

3. Check offline disk drives and attempt to place them online as soon as possible. This is done because an Available message may become lost or may not happen for a long period of time.

4. Disconnect/Re-connect/Reload a remote HSC server that has failed to respond with the status of the oldest request.

## 4.4 Start I/O

Start I/O takes requests from PHYSIO. Start I/O then performs the following functions.

1. Validation of the function requested.

2. Allocation of QOR, BSD and BHD space.

3. Conversion of the addresses to physical block numbers.

4. Queueing of the request to SCA.

5. Move the request from the Position wait queue to the Transfer wait queue.

If a failure to be able to obtain a buffer or start a request to the remote server occurs then Start I/O leaves the request in the Position wait queue until a time when the I/O can be restarted by the Poller or a SCA callback indicating that credit is now available.

## 4.5 Unit Existence Check

This PHYSIO request causes PHYMSC to search thru the UDB entries and return +1 if no unit is found and +2 if a unit exists.

## 4.6 Error Recovery

Since all error recovery is done in the remote system PHYMSC returns +2 to PHYSIO to indicate that error recovery is complete.

All other PHYSIO Kontroller routines do not have any meaning in PHYMSC and they return +1 to indicate success.

## 4.7 SCA Interface

PHYMSC relies on SCA to send messages, receive messages and datagrams and provide information on node and port states.

SCA is responsible for providing message buffers sufficient for all the PHYMSC traffic on the CI. In particular, PHYMSC does not create any buffers for its own use.

## 5.0 INTERRUPT SERVICE

The interrupt service routines are all associated with SCA callbacks. Unused SCA callbacks just return since they shouldn't happen due to the fact that they are unused SCA features. The used callbacks are:

1. Datagram

This callback only happens for error logging. The error log information is taken directly from the SCA packet and entered in the SYSERR log. No translations are performed and the block type is SEC%EL.

2. Port Broke Connection

This callback causes a cleanup of the database and declares all the units on the specified ports to be offline.

3. Connect Response Available

This callback only occurs in response to the PHYMSC connect request and causes us to go to the next state in the initialization of a unit.

4. Node Came Online

This callback causes PHYMSC to attempt to initialize any units on the specified node.

5. Credit Available

When this callback occurs PHYMSC attempts to start any transfers that have been waiting due to lack of credit to the specified HSC.

6. Node Offline

This is the same as 2 since it is impossible to continue to talk to a node that has gone offline.

7. Message received

This is the mechanism with which PHYMSC is notified of an available unit and the status of any MSCP requests (end messages). When an available occurs PHYMSC proceeds to the initialization routine to attempt to online the unit. If the message was an End Message the type is determined and the specific service routine for initialization or I/O done. If the end Message was unsolicited a BUGCHK will occur.

8. All other SCA Callbacks are ignored and return +1.

## 6.0 ERRORS

Errors fall into two major categories. The first is the inability to obtain buffers for requests to the remote server. This failure will cause the current I/O requests to remain on the PHYSIO position wait queue until buffers become available for transmission to the remote server. The second failure is caused by data errors.

### 6.1 Data Errors

Device data errors are divided into two classifications to be mapped into the PHYSIO error returns. ST%MFT and ST%DAT are converted into IS.DAT which causes Bat Block allocation. All other errors are converted into device errors (IS.DVE). Both of these types of errors cause the user to receive an I/O error. Error logging is done on datagrams by request of the server as previously described.

### 6.2 Bat Blocks

The Bat Block allocation of PHYSIO is not altered for PHYMSC. When the remote servers do implement Bat Block replacement there should be no changes required for PHYMSC. Bat Block replacement is supposed to be invisible to the host. This means that if the server ever returns an error it was impossible for the server to do Bat Block replacement and we must perform Bat Block allocation in any case.

### 6.3 Server Failures/Hung Transfers

If a server fails it is assumed that the server will correct any problems by performing a dis-connect/re-connect sequence. The server is expected to completely re-initialize internal database information on dis-connect.

This means that the decision has been made not to terminate any transfers due to the length of time it takes for a transfer to complete to a remote server. As long Get Command Status shows progress on a command then it is assumed that the remote server is alive. If the server does not show progress then PHYMSC will dis-connect/re-connect if this still does not show progress PHYMSC will try to reload and start the remote server. It is assumed that this process should eventually work for all remote servers and therefore there should be no need to abort an I/O transfer.

## 6.4 Lack Of Credit

The lack of credit return from SCA is handled like any other of buffer allocation failure. If any message send request fails due to lack of credit PHYMSC postpones the request until more credit is available.

# 7.0 DUAL PORT SUPPORT

Dual Port disks are supported as provided for by the HSC50. The disks are onlined only to one port and the Hardware requires operator intervention switch ports on failure of the hardware by use of the Port select switches.

# 8.0 TAPE SERVICE AND RA60 SUPPORT

Currently we don't have any RA60 or Tape drives available. The code for RA60's is implemented but untested. The code for Tape service (TA78's) exists but is inaccessable and untested. The tape project should be the subject of another release and is only left in the PHYMSC driver for experimental purposes to test the KLIPA hardware as soon as possible.

## 9.0 CONFIGURATION SUPPORT

Currently we support 16 CI nodes and 24 units/node. This can be restricted further when Hardware engineering determines the maximum configuration that should be supported on the HSC50.