Operating Instructions OAGLinux V2.2

All technical information concerning our products has been attained from past experience and should serve to assist the user. None of the details given in our documents assure that a product has special characteristics. Excepted from this are product features which we in particular cases individually guaranty in writing. We reserve the right to make amendments brought about as a result of technical advances.

Table of Contents

1 Overview

The CompactPCI CNC System D500, as well as the operating stations E2000 and E3000[1], make use of the operating system OAGLinux, which supports a broad spectrum of functions and protocols, that are important nowadays. It is dedicated for use in industrial real time applications. Together with the application and the associated data, the operation system is stored on a CompactFlash Card, that is inserted into a designated slot available in the CNC and the operating station. The configuration of the operating system, respectively the adaptation to its operational area, is made by a single user-friendly file. By use of licenses issued by OPTRONIC AG, system and application functions are unlocked. Application software, licenses, configuration data, and system adjustments are loaded into the system by use of diskettes. The system can be integrated into a 10 or 100Mbit/s TCP/IP Ethernet network and can be monitored remotely over a telephone connection by use of an analog modem.

2 Supported Communication Protocols

For communication over Ethernet and (for remote maintenance purposes) modem, various possibilities are available:

- Web Server: With a common Web Browser, data can be fetched from, or transmitted into the system.
- SMB Server: Integration of the system as SMB File Server into a Microsoft Windows network. As usual between Microsoft Windows machines, the SMB protocol is used.
- NFS Server: Integration of the system as NFS File Server into a network. NFS is supported for example by UNIX and Linux.
- FTP Server: Data can be fetched from, or transmitted into the system by use of a FTP Client.
- FTP Client: Data can be transmitted automatically into another system, on which a FTP Server runs.
- Email Client: Emails with possibly attachment can be transferred automatically to a SMTP Mail Server, that posts the email.

Customized special solutions with proprietary protocols are possible, too.
A printer can be attached to the system over the parallel interface or over the network. As printer languages, PostScript and HP PCL are used. Additionally, it is possible to print pure text. The following protocols for accessing a network printer are supported by standard:

- SMB: Usually for printers, which are accessible over a Microsoft Windows computer.
- TCP: For example for printers, which receive data over a TCP port.
- BSD: For example for printers, which are attached to a UNIX computer.

If the system is equipped with a modem, it can be maintained remotely over the telephone. Depending on the application, the connection is made by use of proprietary software, or by a telecommunications connection, how it can be established with a Microsoft Windows computer by standard.
Existing plants with D100 and D300 hardware, which use a E2000 or E3000 operating station for visualization, can also profit from a majority of the communication possibilities by updating the operating station to an OAGLinux based system. In many cases, it is also possible to enhance other OPTRONIC parts and even other vendors' systems, by equipping them with an operating station E2000 or E3000 (possible without flat screen unit).

---

[1] For the sake of simplicity, from now on the term 'system' is used for the CNC D500, as well as for the operating stations E2000.

3 CompactFlash Card

3.1 Handling of the Card

The CompactFlash Card serves as storage medium for the operating system, the application software, as well as for various configuration and application data. It is inserted into the Slot intended in the system for it (see Illustration 1 and Illustration 2). In order to avoid impurities as far as possible, the slot is covered by a plate that is fixed with two screws (see Illustration 4 and Illustration 3).

CompactFlash Slot

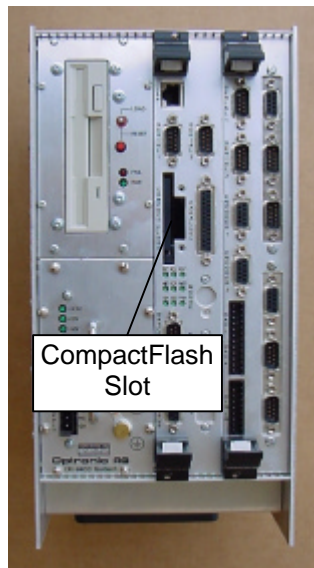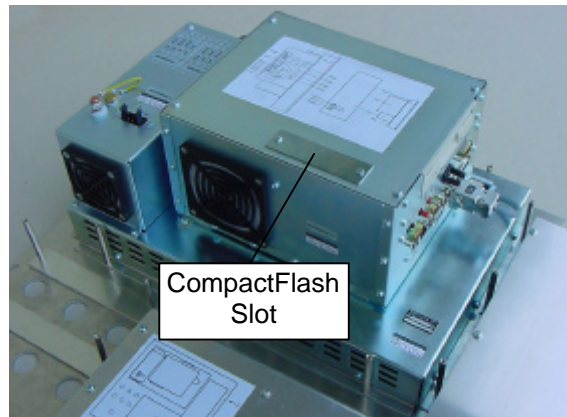Illustration 1: Frontal view D500

CompactFlash Slot

Illustration 2: Rear view operating station

Illustration 3: D500 CompactFlash Slot with inserted CompactFlash Card and removed cover plate

Illustration 4: CompactFlash Slot of an operating station with inserted CompactFlash Card and removed cover plate

Please note that the card may be removed only from the slot, if the system is switched off!

3.2 Data Areas of a Card

A CompactFlash Card with installed OAGLinux operating system offers four data areas to the user and the application. Each area is dedicated to a certain purpose:

Name of the data area	Intended purpose
config	Area for system configuration data and licenses.
app	Area where the application software is stored.
vdata	Area for application data that gets lost when power is turned off.
nvdata	Area for application data that must be maintained also when power is turned off.

All areas are administered by the system. The user can access the areas **config** and **app** by hand. as described in chapter 3.3. **vdata** and **nvdata** are used by the application, for example in order to buffer data or to supply data to the user. It can be accessed over the network and, if necessary, over the telephone line, if the system is equipped with a modem for remote maintenance. In principle, all communication protocols presented in chapter 2 can be used therefore.

3.3 Changing the System Configuration or Loading New Application Software on a Card

By use of a standard PC with Microsoft Windows operating system, and a CompactFlash Card Reader or a free PC Card Slot (mostly available with laptops), the system configuration can be changed, or new licenses and application software can be loaded. Please proceed as follows:

- Power-down the system.
- Remove the cover plate in front of the CompactFlash Card slot of the system.
- Take the CompactFlash Card out of the system by pressing the eject lever. The eject lever is located in the slot beside the card.
- Insert the CompactFlash Card into the intended slot in your PC.
    - Access by use of a PC Card slot:
        - Order a CompactFlash to PC-Card Adapter from OPTRONIC AG (see Illustration 5).
        - Insert the CompactFlash Card into the CompactFlash to PC-Card Adapter (refer to Illustration 6).
        - Insert the CompactFlash to PC-Card Adapter, together with the CompactFlash Card, into a free PC Card slot of your PC or laptop.
        - Wait until your PC or laptop has detected the card and shows the readable contents.

Illustration 5: CompactFlash Card and CompactFlash to PC Card Adapter

Illustration 6: CompactFlash to PC-Card Adapter with inserted CompactFlash Card

- Access by use of a CompactFlash Card Reader:
    - Read the part of the manual of the CompactFlash Card Reader that informs you how a CompactFlash Card must be logged on and off.
    - Insert the CompactFlash Card into the CompactFlash Card Reader.
    - Log on the card to the operating system by use of the CompactFlash Card Reader software. This is maybe done automatically.
- Two drives with the labels **OAG_CONFIG** (data area **config**) and **OAG_APP** (data area **app**) should be added (refer to chapter 3.2). If your computer has only added one drive and you use a USB CompactFlash Card Reader, log off the USB device from the system an log it on again afterwards with already inserted CompactFlash Card. Now, both drives should be present.
- Apply the necessary changes. The drive with the label **OAG_CONFIG** may contains two files: The file **key.txt**, that contains licenses, and the file **config.txt**, which holds system configuration data. The drive **OAG_APP** contains the application. You can modify the system configuration or the license file by removing, replacing, or editing[2] the files **config.txt** and **key.txt**.

  In many cases, you can load a new application by first removing all files from drive **OAG_APP** and unpacking the archive **<target_name>app.tgz** you got from OPTRONIC AG (for example with *WinZip* (www.winzip.com)) to the root directory of the drive, afterwards. **<target_name>** stands for the name of the system, and can be therefore **d500**, **e2000** or **e3000**.

  However, in some systems, the application does not only consist of only one archive, but of two or even several with the names **<target_name>ap*.tgz**. The star is a wildcard for a number or the letter '**n**'. In this case, archives must be unpacked in sequence, beginning with number 1. The last one must be **<target_name>apn.tgz**.

  Further,it can be necessary to load or change additional files, which is done during the standard software update by use of scripts, as described in chapter 4. In this case, do not load new software in the way mentioned above. You can detect this case, if one of the files **<target_name>cmd.txt**, **<target_name>cmd.tgz** or **<target_name>run.sh** is part of the software update data.

  If you possess two CompactFlash Cards and if you want to copy the application from one to the other, then it is in most cases sufficient to simply copy the complete contents of the drive labeled **OAG_APP** from the source to the destination card. Sometimes this is maybe not sufficient, but you cannot recognize this circumstance by yourself. Therefore, please contact OPTRONIC AG if you want to copy an application onto several cards in this way.
- Log off the PC or CompactFlash Card, as soon as you have made all changes (please consult the Windows or CompactFlash Card Reader manual).
- Remove the PC- or CompactFlash Card from the slot in your PC.
- Insert the CompactFlash Card into the intended slot in the system.
- Reattach the cover plate in front of the CompactFlash Card slot of the system.

Please acquire all CompactFlash Cards, you want to use with OPTRONIC systems, always at OP-TRONIC AG! Large quality differences between the individual manufacturers exist, and even manufacturers of industrial-grade cards produce others for the mass-market that are qualitatively worse!
The operating system files stored on the CompactFlash Card are hidden for the user. Normally, operating system adjustments are accomplished only by OPTRONIC AG.
New CompactFlash cards with already installed operating system and prepared data areas can be ordered from OPTRONIC AG at any time.

---

[2] In Microsoft Windows, normally **edit.com** and **notepad.exe** are available.

4 Handling and Software Maintenance

The basic handling and software maintenance of the system is done by use of the reset button, the reset/load switch, the red LED with label **FAIL,** and the green LED with label **RUN**. The reset/load switch has two positions with labels **LOAD** and **RESET.** LED **RUN** and LED **FAIL** each can be in one of the following four conditions:
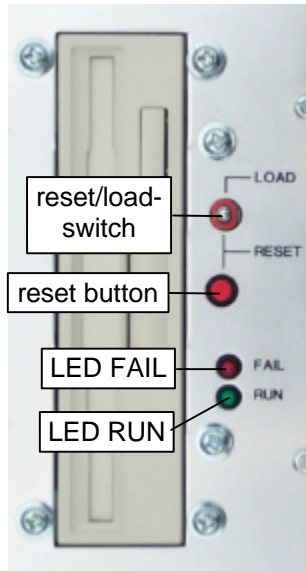
- turned off
- turned on
- fast flashing: 250ms on and 250ms off
- slow flashing: 250ms on and 750ms off

All control elements are located near the diskette drive. They serve to:
- Restart the system.
- Load new application software.
- Load a new license file.
- Load new system configuration data.
- Load new application configuration data.
- Execute system adjustments.
- Determine, if the operating system and the application could be started correctly.
- Control basic system functions.
- Inquire system status information.
- Execute system tests.

Illustration 8: Control elements of the CNC D500
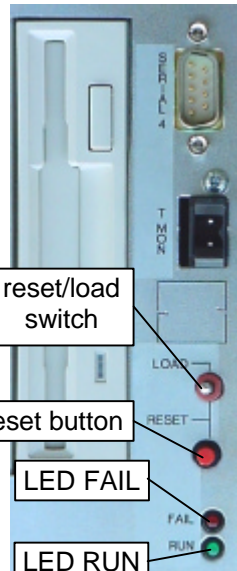
Illustration 7: Control elements of the operating stations E2000 and E3000

4.1 System Cold Start

As soon as the system is connected to the power supply, the operating system is loaded and - if available - the application started. The start-up process lasts about one minute.

4.2 Restarting the System (Reset/Warm Reboot)

A restart (respectively a reset or a warm reboot) of the system can be enforced without interrupting the power supply by shifting the reset/load switch to the position **RESET** and holding the reset button at the same time. It lasts about one minute until the system is restarted.

4.3 Error-Free System Start

An error-free system start is roughly partitioned into the following sections:
1. The system is connected to the power supply (cold start), or restarted, like it is described in chapter 4.2.
2. The hardware is initialized and the first phase of the operating system start-up is executed. During this phase, LED **FAIL** are LED **RUN** are both turned off.
3. After about half a minute, LED **FAIL** and LED **RUN** are turned on simultaneously. The second phase of the operating system start-up has begun.
4. After approximately ten seconds, it is changed over to start-up phase three. LED **FAIL** and LED **RUN** are turned off.
5. After about ten more seconds, the operating system is started up. Now, the application will be started. The turned on LED **RUN** indicates that the operating system and the application are started correctly. If no application is loaded, or if the application cannot be started, LED **FAIL** is turned on instead of LED **RUN**.

4.4 Loading new Application Software, Licenses and Configuration Data, Executing System Adjustments and System Tests, Inquiring System Status Information

Loading new application software, licenses and configuration data, executing system adjustments and system tests, and/or inquiring system status information is done by use of the diskette drive. Insert the MS-DOS formatted diskette containing the necessary data into the diskette drive and start the system as described in chapter 4.1 or 4.2. After about half a minute, LED **RUN** and LED **FAIL** are turned on simultaneously for about ten seconds. Now, you have time to move and hold the reset/load switch into position **LOAD**, until LED **FAIL** is turned off and LED **RUN** begins to flash fast. Release the reset/load switch. Data is read from the diskette and the necessary actions are carried out. If it comes to an error, LED **FAIL** begins to flash fast and LED **RUN** is turned off. If everything runs off error free, the application is started, if available. The turned on LED **RUN** indicates then that the application was started error free. If LED **FAIL** is turned on instead, either no application is loaded, or it could not be started.

If a new application that is distributed over several diskettes must be loaded, the system requests for a diskette change by fast flashing LEDs **RUN** and **FAIL**. As soon as the new diskette is inserted, the reset/load switch must be shifted and held into position **LOAD** until LED **FAIL** turns off and LED **RUN** flashes fast again, which can last not longer than one second. Reading in is continued thereafter.

Depending on the type of data loaded, the system maybe restarts automatically at the end. This is for example the case, if new system configuration data is read in.

The output of executed system tests and inquired system status information is stored on the inserted diskette. Therefore, make sure that the diskette is not write protected.

The operating station offers the possibility to load new application software, licenses and configuration data, execute system adjustments and system tests and/or inquiry system status information by use of the application. The screen and the keyboard for the dialogue with the user are used.

4.5 Erroneous System Configuration Data, Loading New System Configuration Data

During the first phase of the operating system start-up, if the system detects that the system configuration data is incorrect, LED **RUN** and LED **FAIL** begin to flash slowly. The system waits until a diskette with new configuration data is inserted and the reset/load switch is shifted into position **LOAD**. The switch must be held into this position until the system tries to read from the inserted diskette, which can last not longer than one second. If no new configuration data could be loaded, or if the new data falls through a rudimentary check, the system will be stopped. This condition is indicated by a turned off LED **RUN** and a slowly flashing LED **FAIL**. If new configuration data is loaded correctly, the system restarts automatically.

4.6 Controlling Fundamental System Functions

A started system can be examined at any time for the correct working of fundamental system functions. Therefore, the reset/load switch must be shifted and held into position **LOAD**, until LED **RUN** starts flashing fast. LED **FAIL** stays dark, or will be turned off. It can last up to five seconds, until the system recognizes the into position **LOAD** shifted reset/load switch. No longer than one second after the reset/load switch is released, LEDs **RUN** and **FAIL** will shine the same way as before the switch was operated.

4.7 Inquiring System Status Information and Executing System Tests in the Running System

Chapter 4.4 describes how system status information can be inquired and system tests can be executed during operating system start-up. This is also possible when the system is already started up and runs in normal operation. Therefore, the MS-DOS formatted diskette containing the necessary data must be inserted into the diskette drive, and the user must go on as described in chapter 4.6. As soon as LED **RUN** flashes fast, the system loads the relevant data from the diskette, executes the demanded tests, and inquires the requested system status information. Now, release the reset/load switch. The output is stored in the inserted diskette. However, this is only possible if the disk is not write protected.

If it comes to an error during the execution, LED **FAIL** begins to flash fast additionally to LED **RUN**. This error signaling must be acknowledged by again shifting and holding the reset/load switch into position **LOAD,** until LED **FAIL** turns off.

Independently of whether everything ran off correctly or not, no longer than one second after the reset/load switch is released, LEDs **RUN** and **FAIL** will shine the same way as before the inquiry of system status information and/or executing of system test was started.

4.8 Overview of the System Status Indication by Means of LED FAIL and LED RUN

The system reports its state by use of LED **FAIL** and LED **RUN**. The following table lists all states that can arise and can be signaled during error-free operation:

LED RUN	LED FAIL	Meaning
off	off	The system is in the first or third phase of the start-up process.
on	on	The system is in the second phase of the start-up process. The user may initiates the loading of new data from diskette by shifting the reset/load switch into position **LOAD**.
fast flashing	off	The system loads new application software, licenses and configuration data, executes system adjustments and system tests, and/or inquires system status information.
alternating fast flashing	alternating fast flashing	While new application software is loaded, the system requests the user to change the diskette, and to shift the reset/load switch into position **LOAD** for confirmation.
on	off	The application was started error-free.

The following error states are signaled by use of LED **FAIL** and LED **RUN**:

LED RUN	LED FAIL	Meaning
off	on	No application is loaded, or the application could not be started error-free.
synchronous slow flashing	synchronous slow flashing	The system configuration data is erroneous. The system waits until a diskette containing new system configuration data is inserted and the reset/load switch is shifted into position **LOAD**.
synchronous fast flashing	synchronous fast flashing	The execution of system tests and/or the inquiry of system status information while the system is running has failed.
off	slow flashing	The system configuration data is erroneous and it was not possible to load correct data. The system has been stopped.
off	fast flashing	Loading new application software, licenses and configuration data, executing system adjustments and system tests and/or inquiring system status information during system start has failed.

5 Files that can be Loaded by use of a Diskette

As described in chapter 4, loading new application software, licenses and configuration data, executing system adjustments and system tests, and/or inquiring system status information is done by use of the diskette drive. The therefore used diskette must be MS-DOS formatted. The files loaded automatically by the system must have special predefined names, consisting always of the system name (`<target_name>`), three additional letters to define the purpose (`<file_purpose>`), and a three letter extension (`<extension>`), that is separated by a point (`.`):

`<target_name><file_purpose>.<extension>`

Please take the meaning of the individual parts from the following tables and subchapters.

`<target_name>`	System
`d500`	CNC D500.
`e2000`	Operating station E2000.
`e3000`	Operating station E3000.

`<extension>`	File type
`sig`	Signature file.
`tgz`	Archive.
`txt`	Text file, that can be viewed and modified with any text editor.
`sh`	Script, that can be executed by the system.

`<file_purpose>` `.<extension>`	Usage
`pwd.txt`	The file contains the system password, which is maybe checked by the system at the beginning.
`app.tgz`	The file is an archive, that contains a complete application software.
`app.sig`	The file is a signature file, that protects the archive with the same name up to the extension (`<target_name>app.tgz`).
`ap[1-9].tgz`	The file is an archive, that contains a part, except the last one, of an application software.
`ap[1-9].sig`	The file is a signature file, that protects the archive with the same name up to the extension (`<target_name>ap[1-9].tgz`).
`apn.tgz`	The file is an archive, that contains the last part of an application software
`apn.sig`	The file is a signature file, that protects the archive with the same name up to the extension (`<target_name>apn.tgz`).
`key.txt`	The file contains one or more licenses.
`cfg.txt`	The file is a system configuration file.
`par.txt`	The file is an application configuration file.
`cmd.txt`	The file contains a script in a high-level language especially developed for OAGLinux, which can be executed on the system. It is used to perform system adjustments, execute system tests, and inquire system status information.
`cmd.tgz`	The archive contains the script file `<target_name>cmd.txt`, and maybe additional files used by the script.
`cmd.sig`	The file is a signature file, that protects an archive with the same name up to the extension (`<target_name>cmd.txt` or `<target_name>cmd.tgz`).
`run.sh`	The file contains a script in a language used by system, that can be executed on the system.
`run.sig`	The file is a signature file, that protects an archive with the same name up to the extension (`<target_name>run.sh`).

Examples:
- A file, that contains one or several licenses for an E3000 system must be named **`e3000key.txt`**.
- An archive, that contains a complete application for a D500 system must be named **`d500app.tgz`**, the associated signature file **`d500app.sig`**.

5.1 Password File

The password file **`<target_name>pwd.txt`** must contain the correct system password. Depending on the system configuration and the system state, the password is checked at the beginning. If it is wrong, the following operation, for example loading new application software, is not executed.

The password stored in the file is not checked, if the system is an operating station and if the operation is started by use of the visualization software (refer to chapter 4.4).

5.2 Signature File

A signature file is used to guarantee, that another file, it is associated to, is unaltered with high probability and is therefore correct. The file name of the signature file differs from the file, which is protected by it, only by the file extension, which is always **`sig`**. Please refer to the following example:

file to protect:	**`example.dat`**
associated signature file:	**`example.sig`**

All signatures are generated only by OPTRONIC AG. If a signature is checked by the system or not depends on the actual system configuration.

5.3 Archives

All archives have the file extension **`tgz`**. They contain one or several compressed files and are generated by use of the programs *GNU zip* and *GNU tar*. For example, you can view and unpack an archive by use of the program *WinZip* (refer to www.winzip.com).

5.4 Application Software

OPTRONIC AG delivers new application software in one or several archives, together with associated signature files. Depending on the system configuration, the signature files are maybe checked by the system. If the application consists of only one archive, the files are named as follows:

> **`<target_name>app.tgz`**   **`<target_name>app.sig`**

If the application consists of several archives, the file names are composed like this:

> **`<target_name>ap1.tgz`**   **`<target_name>ap1.sig`**
> **`<target_name>ap2.tgz`**   **`<target_name>ap2.sig`**
> **`...`**   **`...`**
> **`<target_name>apn.tgz`**   **`<target_name>apn.sig`**

With this method, the application can consist of up to 10 archives. The system reads in the archives starting at number 1. The last one is the archive with the letter '**`n`**' instead of a digit.

5.5 Licenses

By use of licenses issued by OPTRONIC AG, system and application functions can be enabled. You always receive licenses from OPTRONIC AG in a file with name `<target_name>key.txt`, which can be viewed by a common text editor. It is structured like follows:

```
# OPTRONIC AG Software License
TARGET  = "..."
SERIAL  = "..."
PRODUCT = "..."
DATE    = "..."
ID      = "..."
KEY     = "..."
```

A text defined by OPTRONIC AG is inserted between the quotation marks at the right side of the equal sign ('=') (symbolically represented by '"..."'). It is system and application specific. In the license file shown above, the last line contains the actual license key. If several license keys are available, each key is located in a separate line and the file looks like follows:

```
# OPTRONIC AG Software License
TARGET  = "..."
SERIAL  = "..."
PRODUCT = "..."
DATE    = "..."
ID      = "..."
KEY1    = "..."
KEY2    = "..."
...
KEYn    = "..."
```

The letter 'n' is a wildcard for the largest number available. For example, `KEY1` could be a system key, and `KEY2` an application key.

Except of comment lines starting with character '#', each line consists of an assignment of a value (character string) within quotation marks to a variable:

```
VARIABLE_NAME = "VALUE"
```

The variables have the following meanings:

Name	Meaning	Example
`TARGET`	System name.	`"D500"`
`SERIAL`	The serial number of the device.	`"12345"`
`PRODUCT`	The name of the product.	`"D512/34"`
`DATE`	License creation date and time.	`"25.04.2003"`
`ID`	System identification number.	`"1122-3344-5566-7788"`
`KEY`	License key.	`"1122-3344-5566-7788-99AA-BBCC-DDEE-FF00-1122-3344-5566-7788"`
`KEY1, ..., KEYn`	First license key to `n`-th license key.	`"1122-3344-5566-7788-99AA-BBCC-DDEE-FF00-1122-3344-5566-7788"`

An operating station may views the identification number of the system and - if available - the application key. If it is for example connected to a D500, it normally views also the identification number and the application key of the D500. A license key exactly works for one system. It cannot be transferred to another system. Variable `ID` contains the identification number of the system the license is generated for.

File **key.txt**, which is mentioned in chapter 3, is identical to the presented file **<target_name>key.txt**. Loading a new license file as described in chapter 4.4 overwrites file **key.txt** in data area **config** by **<target_name>key.txt**.

5.6 System Configuration Data

At delivery, the system is normally configured optimally for your needs. It can occur that OPTRONIC AG did or does not have all necessary information, or that an adjustment to a new environment becomes necessary. By use of file **<target_name>cfg.txt**, which is available for your system by OPTRONIC AG or which you can read out by use of the script *'Store Configuration'*, various settings can be changed by use of a common text editor. Each line of this file consists of an assignment of a value (number or string in quotation marks) to a variable:

```
VARIABLE_NAME="TEXT"
VARIABLE_NAME=NUMBER
```

File **config.txt**, which is mentioned in chapter 3, is identical to the presented file **<target_name>cfg.txt**. Loading a new system configuration file as described in chapter 4.4 overwrites file **config.txt** in data area **config** by **<target_name>cfg.txt**.

5.7 Application Configuration Data

Application configuration data are passed to the system by use of the file **<target_name>par.txt**. Structure and format of the file correspond to the file **<target_name>cfg.txt**, which is discussed in the previous section:

```
VARIABLE_NAME="TEXT"
VARIABLE_NAME=NUMBER
```

Loading application configuration data by use of the diskette drive is usually used as fast solution to pass data to the application, that cannot be entered somewhere else (e.g. over the visualization software or the Profibus).
An example is setting the Profibus station address of the D500, if no operating station is attached to the system and the operation is made completely by Profibus. In this case, the station address must be set, before data can be passed over the Profibus. The user can write the address into file **<target_name>par.txt**, which can be loaded by use of the floppy disk drive.

5.8 Scripts

Scripts are used in order to make system adjustments, to execute system tests, and to query system status information. A file which contains a script has one of the following names:

```
<target_name>cmd.txt
<target_name>cmd.tgz
<target_name>run.sh
```

<target_name>cmd.txt and **<target_name>cmd.tgz** contain scripts in a high-level language especially developed for OAGLinux, which always generate one or more files on the used diskette as result. Therefore make sure that the disk is not write protected and that as much as possible free storage is available. Beside the actual script file, the associated signature in the file **<target_name>cmd.sig** or **<target_name>run.sig**, as well as the password **file <target_name>pwd.txt** are maybe needed. In any case, **<target_name>cmd.txt** and **<target_name>cmd.tgz** produce the file **status.txt**, which contains information about the processing of the script. Usually, OPTRONIC delivers its systems together with some general scripts, that are normally stored on the *'Utility Diskette'*. Together with additional files that are necessary for execution, each script is stored in a separate folder.

The following table holds all necessary information for the available scripts:

Script name:	Modem Dial Number
Folder on *'Utility Diskette'*:	`scripts/hardware/modem/dialnum`
Purpose:	Dials the in the file `number.txt` given phone number for 20 seconds.
Supported Systems:	Systems equipped with modem.
Additionally needed files:	`number.txt`
Generated Files:	`status.txt`, `scripts.txt`

Script name:	Modem Test Tone
Folder on *'Utility Diskette'*:	`scripts/hardware/modem/testtone`
Purpose:	Checks for the dial tone for 10 seconds.
Supported Systems:	Systems equipped with modem.
Additionally needed files:	none
Generated Files:	`status.txt, modem.txt`

Script name:	Clear NVRAM
Folder on *'Utility Diskette'*:	`scripts/hardware/nvram/clear`
Purpose:	Clears the nonvolatile memory (NVRAM). All memory cells are set to 0.
Supported Systems:	D500
Additionally needed files:	none
Generated Files:	`status.txt`, `nvram.txt`

Script name:	Load NVRAM
Folder on *'Utility Diskette'*:	`scripts/hardware/nvram/load`
Purpose:	Loads the contents of file `nvram.dat` into the nonvolatile memory (NVRAM).
Supported Systems:	D500
Additionally needed files:	`nvram.dat`
Generated Files:	`status.txt`

Script name:	Store NVRAM
Folder on *'Utility Diskette'*:	`scripts/hardware/nvram/store`
Purpose:	Stores the contents of the nonvolatile memory (NVRAM) into the file `nvram.dat`.
Supported Systems:	D500
Additionally needed files:	none
Generated Files:	`status.txt`, `nvram.dat`

Script name:	Test NVRAM
Folder on *'Utility Diskette'*:	`scripts/hardware/nvram/test`
Purpose:	Tests the nonvolatile memory (NVRAM). The memory contents is not destroyed.
Supported Systems:	D500
Additionally needed files:	none
Generated Files:	`status.txt`

Script name:	Test OMI/OIPs 13Bit, Binary
Folder on *'Utility Diskette'*:	`scripts/hardware/omiips/test13b`
Purpose:	Tests the hardware located on the OMI500 and OIP500 modules for 60 seconds. Absolute encoder SSI interfaces are configured for using a 13 bit binary interface.
Supported Systems:	D500 with OMI500 and possibly OIP500 modules
Additionally needed files:	none
Generated Files:	`status.txt`, `oagio.txt`

Script name:	Test OMI/OIPs 13Bit, Gray
Folder on *'Utility Diskette'*:	**scripts/hardware/omiips/test13g**
Purpose:	Tests the hardware located on the OMI500 and OIP500 modules for 60 seconds. Absolute encoder SSI interfaces are configured for using a 13 bit gray coded interface.
Supported Systems:	D500 with OMI500 and possibly OIP500 modules
Additionally needed files:	none
Generated Files:	**status.txt**, **oagio.txt**

Script name:	Test OMI/OIPs 25Bit, Binary
Folder on *'Utility Diskette'*:	**scripts/hardware/omiips/test25b**
Purpose:	Tests the hardware located on the OMI500 and OIP500 modules for 60 seconds. Absolute encoder SSI interfaces are configured for using a 25 bit binary interface.
Supported Systems:	D500 with OMI500 and possibly OIP500 modules
Additionally needed files:	none
Generated Files:	**status.txt**, **oagio.txt**

Script name:	Test OMI/OIPs 25Bit, Gray
Folder on *'Utility Diskette'*:	**scripts/hardware/omiips/test25g**
Purpose:	Tests the hardware located on the OMI500 and OIP500 modules for 60 seconds. Absolute encoder SSI interfaces are configured for using a 25 bit gray coded interface.
Supported Systems:	D500 with OMI500 and possibly OIP500 modules
Additionally needed files:	none
Generated Files:	**status.txt**, **oagio.txt**

Script name:	Test OMI/OIPs Relays
Folder on *'Utility Diskette'*:	**scripts/hardware/relays/test**
Purpose:	Tests the relays on the OMI500 and OIP500 modules. They are opened and closed once per second five times.
Supported Systems:	D500 with OMI500 and possibly OIP500 modules
Additionally needed files:	none
Generated Files:	**status.txt**

Script name:	Test Watchdog
Folder on *'Utility Diskette'*:	**scripts/hardware/watchdog/test**
Purpose:	Tests the watchdog. The watchdog is flashing once per second for five seconds.
Supported Systems:	D500
Additionally needed files:	none
Generated Files:	**status.txt**

Script name:	Delete Application
Folder on *'Utility Diskette'*:	**scripts/system/delapp**
Purpose:	Removes the loaded application and all it's data, and reinitializes the data areas **app** and **nvdata** on the CompactFlash Card.
Supported Systems:	all
Additionally needed files:	none
Generated Files:	**status.txt**

Script name: Delete NVDATA
Folder on *'Utility Diskette'*: `scripts/system/delnvdat`
Purpose: Reinitializes the data area `nvdata` on the CompactFlash Card.
Supported Systems: all
Additionally needed files: none
Generated Files: `status.txt`

Script name: Delete System Configuration and Licenses
Folder on *'Utility Diskette'*: `scripts/system/delcfgky`
Purpose: Removes the files `config.txt` and `key.txt` and reinitializes the configuration area `config` on the CompactFlash Card.
Supported Systems: all
Additionally needed files: none
Generated Files: `status.txt`

Script name: Get System Status
Folder on *'Utility Diskette'*: `scripts/system/getstat`
Purpose: Retrieves system status information that can be used for problem analysis. Stores it into the file `system.txt`.
Supported Systems: all
Additionally needed files: none
Generated Files: `status.txt, system.txt`

Script name: Store Application
Folder on *'Utility Diskette'*: `scripts/system/storeapp`
Purpose: Stores the currently loaded application into the archive `app.tgz`.
Supported Systems: all
Additionally needed files: none
Generated Files: `status.txt`, `app.tgz`

Script name: Store Configuration
Folder on *'Utility Diskette'*: `scripts/system/storecfg`
Purpose: Stores the current system configuration data into the file `config.txt`.
Supported Systems: all
Additionally needed files: none
Generated Files: `status.txt`, `config.txt`

Script name: Store Licenses
Folder on *'Utility Diskette'*: `scripts/system/storekey`
Purpose: Stores the loaded licenses into the file `key.txt`.
Supported Systems: all
Additionally needed files: none
Generated Files: `status.txt`, `key.txt`

All these scripts can be executed only during system start up and not during normal operation. The structure and contents of the produced files are not described in this document. They are to a large extent self-explanatory. `system.txt` contains many system information, which usually can be evaluated only by OPTRONIC AG. The by script *'Store Application'* in the file `app.tgz` stored application cannot be loaded so easily back into the system, since sometimes additional steps could be necessary to load a new application. The file should be used only for purely informative purposes.
Please contact OPTRONIC AG, if you are not sure how a scripts must be handled!

6 By License Unlockable System Services

The following services can be unlocked for a specific system by use of licenses:

- Web Server
- SMB Server
- NFS Server
- FTP Server

- FTP Client
- Email Client
- Network Printer
- Remote Maintenance by modem

Normally, unlocking the service is not the only thing that must be done. For example, it is necessary to define type and structure of the data that should be exchanged by one of the above mentioned protocols, and the application has to produce and/or evaluate it.

7 Examples: Files Necessary for Software Maintenance

7.1 Loading new Application Software

7.1.1 Operating Station E2000 and E3000

Loading new application software is the same for both operating stations and is presented for the E3000.

Filename	Contents
`e3000app.tgz`	The actual application software.
`e3000app.sig`	Signature for the file `e3000app.tgz`.
`e3000run.sh`	Script used to load the image in file `logo.img`.
`e3000run.sig`	Signature for the file `e3000run.sh`.
`logo.img`	Image that should be viewed on the title page.
`e3000pwd.txt`	Password file, usually provided by OPTRONIC AG with the password '`176671`'. The password can be changed by use of the visualization software. The file is not needed, if the software update is made by use of the visualization software, since then the password must be entered by used of the keyboard.
`e3000key.txt`	File with new licenses. Only needed, if new options were acquired or a larger software change is to be carried out.
`e3000cfg.txt`	File with new system configuration data. Only needed, if fundamental adjustments must be accomplished.

7.1.2 CNC D500

Filename	Contents
`d500app.tgz`	The actual application software.
`d500app.sig`	Signature for the file `d500app.tgz`.
`d500key.txt`	File with new licenses. Only needed, if new options were acquired or a larger software change is to be carried out.
`d500cfg.txt`	File with new system configuration data. Only needed, if fundamental adjustments must be accomplished.

7.2 Loading System Configuration Data and Licenses

Loading system configuration data and licenses is the same for all systems and is presented for the CNC D500. It varies whether only one of the two files is necessary or whether both must be present.

Filename	Contents
`d500key.txt`	File with new licenses. Needed, if new options were acquired or a larger software change is to be carried out.
`d500cfg.txt`	File with new system configuration data. Needed, if fundamental adjustments must be accomplished.

7.3 Loading Application Configuration Data

Loading application configuration data is the same for all systems and is presented for the CNC D500.

Filename	Contents
`d500par.txt`	File with new application configuration data.

7.4 Executing System Tests and System Adjustments, Inquiring System Status Information

7.4.1 Operating Station E2000 and E3000

Executing system tests and system adjustments, and inquiring system status information is the same for both operating stations and is presented for the E3000. As an example, it should be determined, if the integrated modem is connected correctly to the telephone line. Therefore, a given number is dialed for 20 seconds by use of the script '*Modem Dial Number*'.

Filename	Contents
`e3000cmd.tgz`	Script archive.
`e3000cmd.sig`	Signature for the file `e3000cmd.tgz`.
`e3000pwd.txt`	Password file, usually provided by OPTRONIC AG with the password '`176671`'. The password can be changed by use of the visualization software. The file is necessary as the script is started by use of the reset button and the reset/load switch.
`number.txt`	File containing the phone number that should be dialed.

7.4.2 CNC D500

As an example, the script '*Get System Status*' should be executed to inquiry system status information for problem analysis.

Filename	Contents
`d500cmd.tgz`	Script archive.
`d500cmd.sig`	Signature for the file `d500cmd.tgz`.