



Bedienungsanleitung OAGLinux V3



Optronik AG

Unteregger Strasse 53
CH-9403 Goldach
www.optronic.ch

Alle technischen Informationen zu Erzeugnissen unseres Unternehmens sind Erfahrungswerte, die der Orientierung der Anwender dienen sollen. Keine der in unseren Dokumenten gemachten Angaben sichern einem Produkt spezielle Eigenschaften zu. Davon ausgenommen sind Produkteigenschaften, die wir in Einzelfällen schriftlich und individuell garantieren. Durch den technischen Fortschritt bedingte Änderungen behalten wir uns vor.

Es ist nicht gestattet, dieses Dokument oder Teile davon zu vervielfältigen. Die Verwendung dieses Dokuments für andere als persönliche Zwecke ohne Genehmigung der Firma OPTRONIC AG ist nach dem Urheberrechtsgesetz verboten.

Einige Bezeichnungen, die in diesem Dokument erwähnt werden, sind eingetragene Warenzeichen und sollten als solche betrachtet werden.



Inhaltsverzeichnis

1 Übersicht	5
2 Unterstützte Kommunikationsprotokolle	5
3 CompactFlash Karte	6
3.1 Handhabung der Karte	6
3.2 Datenbereiche einer Karte	7
3.3 Systemkonfiguration einer Karte ändern und neue Applikationssoftware darauf installieren	7
4 Bedienung und Softwarewartung	9
4.1 Kaltstart des Systems	10
4.2 System neu starten (Reset/Warmstart)	10
4.3 Fehlerfreier Systemstart	10
4.4 Einspielen neuer Applikationssoftware, Lizenzen und Konfigurationsdaten, Ausführen von Systemanpassungen und Systemtests, Abfragen von Systemstatusinformationen	10
4.5 Fehlerhafte System-Konfigurationsdaten, Einspielen neuer System-Konfigurationsdaten	11
4.6 Kontrollieren von grundlegenden Systemfunktionen	11
4.7 Abfragen von Systemstatusinformationen und Ausführen von Systemtests im laufenden System	11
4.8 Übersicht über die Systemzustandsanzeige mittels LED FAIL und LED RUN	12
5 Über USB-Stick einspielbare Dateien und deren Format	13
5.1 Passwortdatei	14
5.2 Signaturen	14
5.3 Archive	14
5.4 Applikationssoftware	14
5.5 Lizenzen	15
5.6 System-Konfigurationsdaten	16
5.7 Applikations-Konfigurationsdaten	16
5.8 Skripte	16
6 Per Lizenz freischaltbare Systemdienste	20
7 Beispiele: Zur Softwarewartung notwendige Dateien	20
7.1 Einspielen neuer Applikationssoftware	20
7.1.1 Bedienstationen E2000 und E3000	20
7.1.2 CNC D500	20
7.2 Einspielen von System-Konfigurationsdaten und Lizenzen	21

7.3 Einspielen von Applikations-Konfigurationsdaten	21
7.4 Ausführen von Systemtests und Systemanpassungen, Abfragen von Systemstatusinformationen	21
7.4.1 Bedienstationen E2000 und E3000	21
7.4.2 CNC D500	21

1 Übersicht

Im CompactPCI CNC-System D500, sowie in den Bedienstationen E2000 und E3000¹, kommt das Betriebssystem OAGLinux zum Einsatz, welches ein breites Spektrum von heute wichtigen Funktionen und Protokollen unterstützt, dabei aber speziell für den Einsatz in industriellen Echtzeitapplikationen ausgelegt ist. Zusammen mit der Applikation und den zugehörigen Daten ist das Betriebssystem auf einer CompactFlash Card abgespeichert, die in den dafür an der CNC und der Bedienstation vorgesehenen Slot eingeschoben ist. Die Konfiguration des Betriebssystems, beziehungsweise die Anpassung des Systems an sein Einsatzgebiet, erfolgt über eine einzige benutzerfreundliche Datei. Mittels von OPTRONIC AG vergebener Lizenzen werden System- und Applikationsfunktionen freigeschaltet. Applikationssoftware, Lizenzen, Konfigurationsdaten und Systemanpassungen werden mittels USB-Stick ins System geladen. Das System kann in ein 10 oder 100Mbit/s TCP/IP Ethernet-Netzwerk integriert und mittels einer Telefonverbindung über ein analoges Modem ferngewartet werden.

2 Unterstützte Kommunikationsprotokolle

Für die Kommunikation über Ethernet und (für Fernwartungszwecke) über Modem stehen diverse Möglichkeiten zur Verfügung:

- **Web-Server:** Mit einem üblichen Web-Browser können Daten abgerufen oder ins System übertragen werden.
- **SMB-Server:** Integration des Systems als SMB-Fileserver in ein Microsoft Windows Netzwerk. Wie zwischen Microsoft Windows Maschinen üblich, wird dazu das SMB-Protokoll verwendet.
- **NFS-Server:** Integration des Systems als NFS-Fileserver in ein Netzwerk. NFS wird beispielsweise von UNIX und Linux unterstützt.
- **FTP-Server:** Daten können mittels eines FTP-Clients abgerufen oder ins System übertragen werden.
- **FTP-Client:** Daten können automatisch in ein anderes System, auf dem ein FTP-Server läuft, übertragen werden.
- **Email-Client:** Emails mit eventuell als Anlage beigefügten Daten können automatisch an einen SMTP Mailserver übertragen werden, der dann die Email verschickt.

Kundenspezifische Speziallösungen mit proprietären Protokollen sind ebenso möglich.

An das System kann ein Drucker über die parallele Schnittstelle oder über das Netzwerk angeschlossen werden. Als Druckersprachen kommen PostScript und HP PCL zum Einsatz. Daneben besteht die Möglichkeit, reinen Text zu drucken. Folgende Protokolle zum Ansprechen eines Netzwerkdruckers werden standardmässig unterstützt:

- **SMB:** In der Regel für Drucker, die über einen Microsoft Windows Rechner ansprechbar sind.
- **TCP:** Zum Beispiel für Drucker, die Daten über einen TCP Port empfangen.
- **BSD:** Zum Beispiel für Drucker, die an einem UNIX-Rechner angeschlossen sind.

Ist das System mit einem Modem ausgestattet, kann es über das Telefonnetz ferngewartet werden. Die Einwahl erfolgt dabei je nach Anwendung mittels proprietärer Software, oder über eine DFÜ-Verbindung, wie sie mit jedem Microsoft Windows Rechner standardmässig aufgebaut werden kann. Bereits existierende Anlagen mit D100 und D300 Hardware, welche zur Visualisierung eine E2000 oder E3000 Bedienstation verwenden, können durch Aufrüsten der Bedienstation auf ein OAGLinux basiertes System ebenfalls von einem Grossteil der Kommunikationsmöglichkeiten profitieren. In vielen Fällen können auch andere OPTRONIC Anlagen und sogar Fremdsysteme erweitert werden, indem sie mit einer Bedienstation E2000 oder E3000 (eventuell ohne Flachbildschirmbaugruppe) ausgestattet werden.

¹ Der Einfachheit halber wird in der Folge für die CNC D500 sowie für die Bedienstationen E2000 und E3000 der Begriff 'System' verwendet.

3 CompactFlash Karte

3.1 Handhabung der Karte

Die CompactFlash Card dient als Speichermedium für das Betriebssystem, die Applikationssoftware, sowie für diverse Konfigurations- und Applikationsdaten. Sie wird in den im System dafür vorgesehenen Slot eingeschoben (siehe Abbildung 1 und Abbildung 2). Um Verunreinigungen soweit als möglich auszuschliessen, ist der Slot mittels einer durch zwei Schrauben befestigten Platte abgedeckt (siehe Abbildung 1 und Abbildung 2).

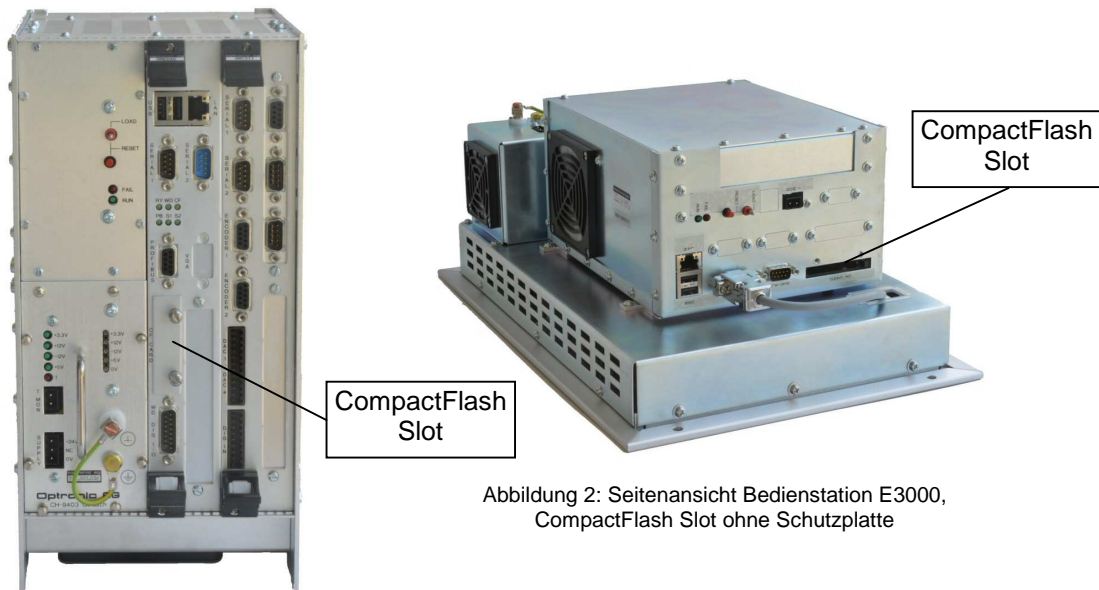


Abbildung 1: Frontansicht D500,
CompactFlash Slot mit aufgesetzter
Schutzplatte

Abbildung 2: Seitenansicht Bedienstation E3000,
CompactFlash Slot ohne Schutzplatte

Beachten Sie bitte unbedingt, dass die Karte nur aus dem Slot entfernt werden darf, wenn das System ausgeschaltet ist!

3.2 Datenbereiche einer Karte

Eine CompactFlash Card mit installiertem Betriebssystem OAGLinux stellt dem Anwender und der Applikation insgesamt vier Datenbereiche zur Verfügung. Jeder Bereich ist für einen bestimmten Zweck reserviert:

Name des Datenbereichs	Verwendungszweck
config	Bereich für System-Konfigurationsdaten und Lizenzen.
app	Bereich in dem die Applikationssoftware abgelegt ist.
vdata	Bereich für Applikationsdaten, die nicht über Netz-Aus gehalten werden müssen.
nvddata	Bereich für Applikationsdaten, die über Netz-Aus gehalten werden müssen.

Alle Bereiche werden vom System verwaltet. Auf die Bereiche **config** und **app** kann der Anwender „von Hand“ wie in Kapitel 3.3 beschrieben zugreifen. **vdata** und **nvddata** werden eventuell von der Applikation verwendet, um beispielsweise Daten zwischenspeichern, oder dem Anwender zur Verfügung zu stellen. Darauf zugegriffen werden kann über das Netzwerk und gegebenenfalls über die Telefonleitung, wenn das System mit einem Modem zur Fernwartung ausgerüstet ist. Dabei können prinzipiell alle in Kapitel 2 vorgestellten Kommunikationsprotokolle zum Einsatz kommen.

3.3 Systemkonfiguration einer Karte ändern und neue Applikationssoftware darauf installieren

Über einen handelsüblichen PC mit Microsoft Windows Betriebssystem und einem CompactFlash Card Reader oder einem freien PC-Card Slot (bei Laptops meistens vorhanden) kann die Systemkonfiguration geändert, oder neue Lizenzen und Applikationssoftware aufgespielt werden. Gehen Sie dazu bitte folgendermassen vor:

- Schalten Sie das System aus.
- Entfernen Sie die Abdeckung vor dem CompactFlash Card Slot im System.
- Nehmen Sie die CompactFlash Card aus dem System, indem Sie auf den Auswurfhebel drücken. Der Auswurfhebel befindet sich im Slot neben der Karte.
- Schieben Sie die CompactFlash Card in den dafür vorgesehenen Slot in Ihrem PC ein.
 - Zugriff über einen PC-Card Slot:
 - Beziehen Sie über OPTRONIC AG einen CompactFlash to PC-Card Adapter (siehe Abbildung 3).
 - Stecken Sie die CompactFlash Karte in den CompactFlash to PC-Card Adapter (siehe Abbildung 4).
 - Schieben Sie den CompactFlash to PC-Card Adapter mit der eingesteckten CompactFlash Card in einen freien PC-Card Slot Ihres PCs oder Laptops.
 - Warten Sie, bis Ihr PC oder Laptop die Karte erkennt und den für Sie lesbaren Inhalt anzeigt.



Abbildung 3: CompactFlash Card und CompactFlash to PC-Card Adapter



Abbildung 4: CompactFlash to PC-Card Adapter mit eingeschobener CompactFlash Card

- Zugriff über einen CompactFlash Card Reader:
 - Informieren Sie sich in der Anleitung zum CompactFlash Card Reader darüber, wie eine CompactFlash Karte an- und abgemeldet werden muss.
 - Stecken Sie die CompactFlash Karte in den CompactFlash Card Reader.
 - Melden Sie die Karte über die CompactFlash Card Reader Software beim Betriebssystem an. Eventuell funktioniert dies automatisch.
- Es sollten nun zwei neue Laufwerke mit den Bezeichnungen **OAG_CONFIG** (Datenbereich **config**) und **OAG_APP** (Datenbereich **app**) hinzugefügt worden sein (siehe Kapitel 3.2). Falls Ihr Computer nur ein neues Laufwerk hinzugefügt hat und Sie einen USB CompactFlash Card Reader verwenden, melden Sie das USB Gerät beim System ab und danach bei eingesteckter CompactFlash Karte wieder an. Danach sollten beide Laufwerke vorhanden sein.
- Führen Sie nun die gewünschten Änderungen durch. Das Laufwerk mit dem Namen **OAG_CONFIG** kann zwei Dateien enthalten: Die Datei **key.txt**, die Lizenzen enthält, sowie die Datei **config.txt** für System-Konfigurationsdaten. Das Laufwerk **OAG_APP** enthält die Applikation. Sie können Änderungen an der Systemkonfiguration oder der Lizenzdatei vornehmen, indem Sie die Dateien **config.txt** und **key.txt** nach belieben löschen, austauschen oder mittels eines Texteditors² bearbeiten.

In vielen Fällen können Sie eine neue Applikation einspielen, indem Sie zuerst alle Dateien vom Laufwerk mit der Bezeichnung **OAG_APP** löschen und danach das von OPTRONIC AG erhaltenen Archiv **<target_name>app.tgz** (beispielsweise mittels *WinZip* (www.winzip.com)) ins Grundverzeichnis des Laufwerks entpacken. **<target_name>** steht dabei für den Namen des Systems, kann also demzufolge **d500**, **e2000** oder **e3000** sein.

Bei manchen Systemen besteht die Applikation jedoch nicht nur aus einem einzigen Archiv, sondern aus zwei oder sogar mehreren mit den Namen **<target_name>ap*.tgz**, wobei der Stern für eine Zahl oder den Buchstaben 'n' steht. In diesem Fall müssen die Archive der Reihe nach, beginnend bei Nummer 1 entpackt werden. Als letztes muss **<target_name>apn.tgz** genommen werden.

Weiter kann es notwendig sein, zusätzliche Dateien aufzuspielen oder zu ändern, was beim normalen Softwareupdate, wie er in Kapitel 4 beschrieben ist, über Skripte durchgeführt wird. Führen Sie in diesem Fall das Einspielen neuer Software bitte nicht auf dem gerade vorgestellten Weg durch. Sie erkennen diesen Fall daran, dass sich bei den Daten für den Softwareupdate mindestens eine Datei mit dem Namen **<target_name>cmd.txt**, **<target_name>cmd.tgz** oder **<target_name>run.sh** befindet.

Besitzen Sie zwei CompactFlash Cards und wollen Sie die Applikation von der einen auf die andere kopieren, so genügt es in den meisten Fällen, einfach den kompletten Inhalt des Laufwerks mit der Bezeichnung **OAG_APP** von der einen Karte auf die andere zu kopieren. Unter Umständen kann es aber sein, dass dies nicht ausreicht. Sie können diesen Umstand aber nicht selber erkennen. Kontaktieren Sie daher bitte OPTRONIC AG, falls Sie eine Applikation auf diesem Weg auf mehrere Karten kopieren möchten.

- Melden Sie in Windows die PC- oder CompactFlash Card ab, sobald Sie alle Änderungen vorgenommen haben. (Konsultieren Sie dazu bitte die Anleitung zu Windows oder zum CompactFlash Card Reader.)
- Entfernen Sie die PC- oder CompactFlash Card aus dem Slot in Ihrem PC.
- Schieben Sie die CompactFlash Card wieder in den im System dafür vorgesehenen Slot ein.
- Bringen Sie die Abdeckung vor dem CompactFlash Card Slot im System wieder an.

Erwerben Sie bitte alle CompactFlash Cards, die Sie mit OPTRONIC Systemen verwenden wollen, immer bei OPTRONIC AG! Es existieren grosse Qualitätsunterschiede zwischen den einzelnen Herstellern und selbst Hersteller industrietauglicher Karten produzieren daneben qualitativ schlechtere für den Massenmarkt!

Die Betriebssystemdateien sind für den Anwender versteckt auf der CompactFlash Card untergebracht. Anpassungen am Betriebssystem werden normalerweise nur von OPTRONIC AG durchgeführt.

Neue CompactFlash Karten mit bereits installiertem Betriebssystem und vorbereiteten Datenbereichen können jederzeit bei OPTRONIC AG bezogen werden.

² Unter Microsoft Windows stehen normalerweise **edit.com** und **notepad.exe** zur Verfügung.

4 Bedienung und Softwarewartung

Die grundlegende Bedienung und Softwarewartung des Systems geschieht mittels der Reset-Taste, dem Reset/Load-Schalter, sowie der roten LED mit der Beschriftung **FAIL** und der mit **RUN** bezeichneten grünen LED.

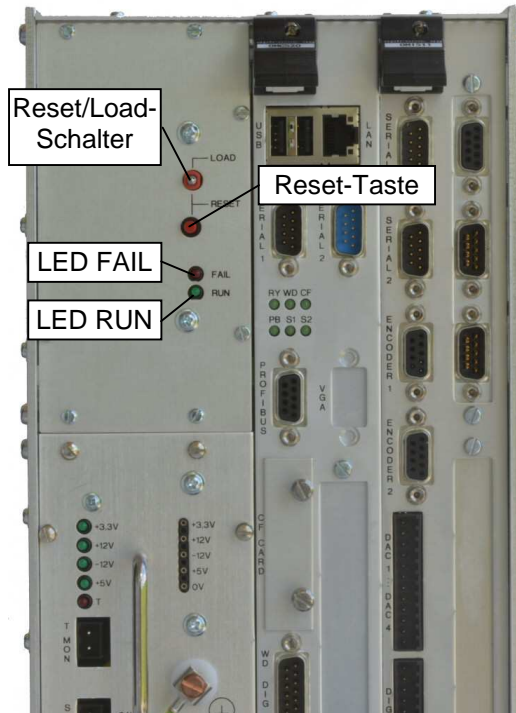


Abbildung 6: Bedienelemente der CNC D500

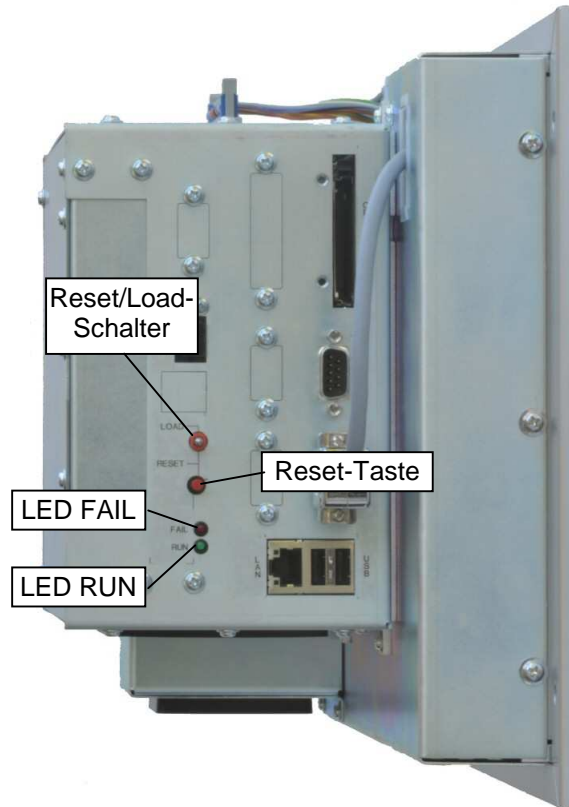


Abbildung 5: Bedienelemente der Bedienstationen E3000

Der Reset/Load-Schalter bietet zwei Stellungen mit den Bezeichnungen **LOAD** und **RESET**. LED **RUN** und LED **FAIL** können einen der folgenden vier Zustände einnehmen:

- ausgeschaltet
- dauerhaft eingeschaltet
- schnelles Blinken: 250ms ein- und danach 250ms ausgeschaltet
- langsames Blinken: 250ms ein- und danach 750ms ausgeschaltet

Die Bedienelemente dienen zum:

- Neustarten des Systems.
- Einspielen neuer Applikationssoftware.
- Einspielen einer neuen Lizenzdatei.
- Einspielen von neuen System-Konfigurations-daten.
- Einlesen von neuen Applikations-Konfigurations-daten.
- Ausführen von Systemanpassungen.
- Feststellen, ob Betriebssystem und Applikation korrekt gestartet wurden.
- Kontrollieren von grundlegenden Systemfunktionen.
- Abfragen von Systemstatusinformationen.
- Ausführen von Systemtests.

4.1 Kaltstart des Systems

Sobald das System an die Stromversorgung angeschlossen ist, wird das Betriebssystem geladen und - sofern vorhanden - die Applikation gestartet. Der Aufstartvorgang dauert etwa eine Minute.

4.2 System neu starten (Reset/Warmstart)

Ein Neustart (bzw. Reset oder Warmstart) des Systems kann ohne Unterbruch der Stromversorgung erzwungen werden, indem der Reset/Load-Schalter in die Position **RESET** geschoben und dabei die Reset-Taste gedrückt wird. Es dauert etwa eine Minute, bis das System neu gestartet ist.

4.3 Fehlerfreier Systemstart

Ein fehlerfreier Systemstart lässt sich grob in die folgenden Abschnitte unterteilen:

1. Das System wird an die Stromversorgung angeschlossen (Kaltstart), oder wie in Kapitel 4.2 beschrieben neu gestartet.
2. Die Hardware wird initialisiert und die erste Phase vom Betriebssystem-Aufstartvorgang ausgeführt. LED **FAIL** und LED **RUN** sind während dieser Phase ausgeschaltet.
3. Sobald LED **FAIL** und LED **RUN** gleichzeitig zu leuchten beginnen hat die zweite Phase des Betriebssystem-Aufstartvorgangs begonnen. Während dieser Phase können neue Daten eingespielt, Systemanpassungen und Systemtests durchgeführt, oder Systemstatusinformationen abgefragt werden.
4. Wenige Sekunden später wird in Aufstart-Phase drei übergegangen. LED **FAIL** und LED **RUN** werden wieder gelöscht.
5. Der Aufstartvorgang des Betriebssystems wird abgeschlossen und die Applikation gestartet. Die leuchtende LED **RUN** zeigt an, dass das Betriebssystem und die Applikation ordnungsgemäss gestartet wurden. Ist keine Applikation geladen oder kann die Applikation nicht gestartet werden, leuchtet LED **FAIL** anstelle von LED **RUN**.

4.4 Einspielen neuer Applikationssoftware, Lizenzen und Konfigurationsdaten, Ausführen von Systemanpassungen und Systemtests, Abfragen von Systemstatusinformationen

Das Einspielen neuer Applikationssoftware, Lizenzen und Konfigurationsdaten, das Ausführen von Systemanpassungen und Systemtests und/oder das Abfragen von Systemstatusinformationen geschieht während des Systemstarts über einen USB-Stick. Starten Sie das System zuerst wie in Kapitel 4.1 oder 4.2 beschrieben neu. Schliessen Sie den USB-Stick an. Nach ungefähr einer halben Minute beginnen LED **RUN** und LED **FAIL** gleichzeitig für einige Sekunden zu leuchten. Bewegen Sie nun den Reset/Load-Schalter in die Position **LOAD** und halten Sie ihn dort, bis die LED **FAIL** aufhört zu leuchten und die LED **RUN** schnell zu blinken beginnt. Lassen Sie nun den Reset/Load-Schalter los. Die Daten werden vom USB Stick gelesen und die notwendigen Aktionen ausgeführt. Kommt es zu einem Fehler, beginnt LED **FAIL** schnell zu blinken und LED **RUN** erlischt. Läuft alles fehlerfrei ab, wird - sofern vorhanden - die Applikation gestartet. Die leuchtende LED **RUN** zeigt dann an, dass die Applikation fehlerfrei gestartet wurde. Leuchtet statt dessen LED **FAIL**, so ist entweder keine Applikation geladen, oder sie konnte nicht gestartet werden.

Ist eine neu einzuspielende Applikation auf mehrere USB-Sticks aufgeteilt, fordert das System durch abwechslungsweises schnelles Blinken der LEDs **RUN** und **FAIL** zum Wechseln des USB-Sticks auf. Sobald der neue USB-Stick angeschlossen ist, muss der Reset/Load-Schalter in die Position **LOAD** bewegt und solange dort gehalten werden, bis LED **FAIL** erlischt und LED **RUN** wieder schnell blinkt. Das Einlesen wird danach fortgesetzt.

Je nach Art der neu eingespielten Daten kann es sein, dass das System am Ende eigenständig einen Neustart ausführt. Dies ist beispielsweise der Fall, wenn neue System-Konfigurationsdaten eingelesen wurden.

Die Ausgaben von durchgeführten Systemtests und die abgefragten Systemstatusinformationen werden auf dem angeschlossenen USB-Stick abgespeichert. Dies ist allerdings nur möglich, wenn der USB-Stick nicht schreibgeschützt ist.

Die Bedienstation bietet die Möglichkeit, das Einspielen neuer Applikationssoftware, Lizenzen und Konfigurationsdaten, das Ausführen von Systemanpassungen und Systemtests und/oder das Abfragen von Systemstatusinformationen über die Applikation vorzunehmen. Dabei werden der Bildschirm und die Tastatur für den Dialog mit dem Anwender verwendet.

4.5 Fehlerhafte System-Konfigurationsdaten, Einspielen neuer System-Konfigurationsdaten

Wird während der ersten Phase des Betriebssystem-Aufstartvorgangs festgestellt, dass die System-Konfigurationsdaten fehlerhaft sind, beginnen LED **RUN** und LED **FAIL** langsam zu blinken. Das System wartet, bis ein USB-Stick mit neuen Konfigurationsdaten angeschlossen und der Reset/Load-Schalter in die Position **LOAD** bewegt wird. Der Schalter muss solange in dieser Position gehalten werden, bis das System versucht, vom angeschlossenen USB-Stick zu lesen. Dies kann maximal eine Sekunde dauern. Können keine neuen Konfigurationsdaten eingelesen werden oder fallen die neuen Daten bei einem rudimentären Check durch, wird das System angehalten. Dieser Zustand wird durch eine ausgeschaltete LED **RUN** und eine langsam blinkende LED **FAIL** angezeigt. Konnten neue Konfigurationsdaten geladen werden, leitet das System automatisch einen Neustart ein.

4.6 Kontrollieren von grundlegenden Systemfunktionen

Ein aufgestartetes System kann jederzeit auf die korrekte Funktionsweise grundlegender Systemfunktionen überprüft werden. Dazu muss lediglich der Reset/Load-Schalter in die Position **LOAD** bewegt und solange dort gehalten werden, bis die LED **RUN** schnell zu blinken beginnt. Die LED **FAIL** bleibt dabei dunkel oder wird ausgeschaltet. Es kann bis zu five Sekunden dauern, bis das System den in die Position **LOAD** bewegten Reset/Load-Schalter erkennt. Spätestens eine Sekunde nachdem der Reset/Load-Schalter losgelassen wurde, leuchten die LEDs **RUN** und **FAIL** wieder genau so, wie bevor der Schalter betätigt wurde.

4.7 Abfragen von Systemstatusinformationen und Ausführen von Systemtests im laufenden System

In Kapitel 4.4 wurde beschrieben, wie während dem Betriebssystem-Aufstartvorgang Systemstatusinformationen abgefragt und Systemtests durchgeführt werden können. Im normalen Betrieb, also wenn das System bereits aufgestartet ist, ist dies ebenfalls möglich. Dazu muss der mit den notwendigen Daten beschriebene USB-Stick ans System angeschlossen und wie in Kapitel 4.6 beschrieben vorgegangen werden. Sobald die LED **RUN** schnell blinkt, lädt das System die notwendigen Daten vom USB-Stick, führt die angeforderten Tests aus und ermittelt die verlangten Systemstatusinformationen. Lassen Sie nun den Reset/Load-Schalter los. Die Ausgaben werden auf dem angeschlossenen USB-Stick abgespeichert. Dies ist allerdings nur möglich, wenn der USB-Stick nicht schreibgeschützt ist.

Kommt es während der Ausführung zu einem Fehler, beginnt die LED **FAIL** zusätzlich zur LED **RUN** schnell zu blinken. Die Fehleranzeige muss quittiert werden, indem der Reset/Load-Schalter wiederum in die Position **LOAD** bewegt und solange dort gehalten wird, bis die LED **FAIL** erlischt.

Unabhängig davon, ob alles korrekt abgelaufen ist oder nicht, leuchten die LEDs **RUN** und **FAIL** spätestens eine Sekunde nachdem der Reset/Load-Schalter losgelassen wurde wieder genau so, wie bevor die Abfrage von Systemstatusinformationen beziehungsweise das Ausführen von Systemtests gestartet wurde.

4.8 Übersicht über die Systemzustandsanzeige mittels LED FAIL und LED RUN

Der Systemzustand wird mittels LED **FAIL** und LED **RUN** mitgeteilt. Folgende Tabelle enthält Zustände, die während dem fehlerfreien Betrieb auftreten und angezeigt werden können:

LED RUN	LED FAIL	Bedeutung
aus	aus	System ist in der ersten oder dritten Phase des Aufstartvorgangs.
ein	ein	System ist in der zweiten Phase des Aufstartvorgangs. Der Benutzer hat die Möglichkeit, dem System durch Umlegen des Reset/Load-Schalters in die Position LOAD mitzuteilen, dass neue Daten geladen werden sollen.
schnelles Blinken	aus	Das System lädt neue Applikationssoftware, Lizenzen und Konfigurationsdaten, führt Systemanpassungen und Systemtests aus und/oder fragt Systemstatusinformationen ab.
abwechselndes schnelles Blinken	abwechselndes schnelles Blinken	Das System fordert den Anwender während dem Einspielen neuer Applikationssoftware dazu auf, den USB-Stick zu wechseln und als Bestätigung den Reset/Load-Schalter in die Position LOAD zu bewegen.
ein	aus	Die Applikation wurde fehlerfrei gestartet.

Folgende Fehlerzustände werden durch die LED **FAIL** und LED **RUN** angezeigt:

LED RUN	LED FAIL	Bedeutung
aus	ein	Es ist keine Applikation geladen oder die Applikation konnte nicht fehlerfrei gestartet werden.
synchrones langsames Blinken	synchrones langsames Blinken	System-Konfigurationsdaten sind fehlerhaft. Das System wartet auf das Anschliessen eines USB-Sticks mit neuen System-Konfigurationsdaten und auf das Umlegen des <i>Reset/Load-Schalters</i> in die Position LOAD .
synchrones schnelles Blinken	synchrones schnelles Blinken	Das Ausführen von Systemtests und/oder das Abfragen von Systemstatusinformationen im laufenden System ist fehlgeschlagen.
aus	langsames Blinken	System-Konfigurationsdaten sind fehlerhaft und es konnten keine neuen geladen werden. Das System wurde angehalten.
aus	schnelles Blinken	Das Laden neuer Applikationssoftware, Lizenzen und Konfigurationsdaten, das Ausführen von Systemanpassungen und Systemtests und/oder das Abfragen von Systemstatusinformationen während dem Systemstart ist fehlgeschlagen.

5 Über USB-Stick einspielbare Dateien und deren Format

Wie in Kapitel 4 beschrieben, geschieht das Einspielen neuer Applikationssoftware, Lizenzen und Konfigurationsdaten, das Ausführen von Systemanpassungen und Systemtests und/oder das Abfragen von Systemstatusinformationen über die USB Schnittstelle und einem USB-Stick. Der USB-Stick muss mit dem Dateisystem FAT bzw. exFAT formatiert sein. Die vom System automatisch einzulesenden Dateien müssen spezielle, vordefinierte Namen haben, die immer aus dem Systemnamen (<target_name>), drei weiteren Zeichen zur Festlegung des Verwendungszwecks (<file_purpose>) und einer drei Zeichen langen, durch einen Punkt (.) abgetrennten Erweiterung (<extension>) aufgebaut sein müssen:

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

Die Bedeutung der einzelnen Teile entnehmen Sie bitte den folgenden Tabellen und Unterkapiteln.

<target_name>	System
d500	CNC D500.
e2000	Bedienstation E2000.
e3000	Bedienstation E3000.

<extension>	Dateityp
sig	Signaturdatei.
tgz	Archiv.
txt	Textdatei, die mit einem beliebigen Texteditor eingesehen und verändert werden kann.
sh	Durch das System ausführbares Skript.

<file_purpose>	Verwendungszweck
.<extension>	
pwd.txt	Die Datei enthält das Systempasswort, welches gegebenenfalls vom System zu Beginn auf seine Richtigkeit hin kontrolliert wird.
app.tgz	Die Datei ist ein Archiv, das eine komplette Applikationssoftware enthält.
app.sig	Die Datei ist eine Signaturdatei, die das Archiv mit dem bis auf die Erweiterung identischen Namen (<target_name>app.tgz) schützt.
ap[1-9].tgz	Die Datei ist ein Archiv, das einen beliebigen Teil ausser den letzten einer Applikationssoftware enthält.
ap[1-9].sig	Die Datei ist eine Signaturdatei, die das Archiv mit dem bis auf die Erweiterung identischen Namen (<target_name>ap[1-9].tgz) schützt.
apn.tgz	Die Datei ist ein Archiv, das den letzten Teil einer Applikationssoftware enthält.
apn.sig	Die Datei ist eine Signaturdatei, die das Archiv mit dem bis auf die Erweiterung identischen Namen (<target_name>apn.tgz) schützt.
key.txt	Die Datei enthält eine oder mehrere Lizenzen.
cfg.txt	Die Datei ist eine System-Konfigurationsdatei.
par.txt	Die Datei ist eine Applikations-Konfigurationsdatei.
cmd.txt	Die Datei enthält ein Skript in einer speziell für OAGLinux entwickelten Hochsprache, das auf dem System ausgeführt werden soll. Es wird dazu verwendet, um Systemanpassungen vorzunehmen, Systemtests auszuführen und um Systemstatusinformationen abzufragen.
cmd.tgz	Das Archiv enthält die Skriptdatei <target_name>cmd.txt, sowie eventuell weitere Dateien die vom Skript verwendet werden.
cmd.sig	Die Datei ist eine Signaturdatei, die ein Skript oder ein Archiv mit dem bis auf die Erweiterung identischen Namen (<target_name>cmd.txt oder <target_name>cmd.tgz) schützt.
run.sh	Die Datei enthält ein Skript in der vom System verwendeten Skriptsprache, das auf dem System ausgeführt werden soll.

run.sig	Die Datei ist eine Signaturdatei, die ein Skript mit dem bis auf die Erweiterung identischen Namen (<code><target_name>run.sh</code>) schützt.
----------------	--

- Beispiele:
- Eine Datei, die eine oder mehrere Lizenzen für ein E3000-System enthält, muss somit den Namen `e3000key.txt` haben.
 - Ein Archiv, das eine komplette Applikation für ein D500-System enthält, muss `d500app.tgz` heissen, die zugehörige Signaturdatei `d500app.sig`.

5.1 Passwortdatei

Die Passwortdatei `<target_name>pwd.txt` muss das korrekte Systempasswort enthalten. Je nach Systemkonfiguration und Systemzustand wird das Passwort zu Beginn kontrolliert. Ist es falsch, werden anstehende Operationen wie beispielsweise das Einspielen neuer Applikationssoftware nicht durchgeführt.

Das Passwort in der Datei wird nicht überprüft, falls es sich beim System um eine Bedienstation handelt und der Vorgang über die Visualisierungssoftware gestartet wurde (siehe Kapitel 4.4).

5.2 Signaturen

Eine Signaturdatei dient dazu um sicherzustellen, dass eine andere von ihr zu schützende Datei mit hoher Wahrscheinlichkeit unverfälscht und demzufolge korrekt ist. Der Dateiname der Signaturdatei unterscheidet sich von der durch sie zu schützenden Datei lediglich durch die Dateierweiterung, die immer `sig` lautet. Folgendes Beispiel soll den Zusammenhang verdeutlichen:

zu schützende Datei: `example.dat`
zugehörige Signaturdatei: `example.sig`

Signaturen können nur von OPTRONIC AG erstellt werden. Ob das System eine Signatur prüft oder nicht, hängt von der aktuellen Systemkonfiguration ab.

5.3 Archive

Ein Archiv weist immer die Erweiterung `tgz` auf. Es enthält eine oder mehrere Dateien die komprimiert sind. Es wurde mittels der Programme *GNU zip* und *GNU tar* erzeugt. Sie können jedes Archiv beispielsweise mit Hilfe des Programms *WinZip* (siehe www.winzip.com) einsehen und entpacken.

5.4 Applikationssoftware

Neue Applikationssoftware wird von OPTRONIC AG in einem oder mehreren Archiven und zugehörigen Signaturdateien geliefert. Je nach Systemkonfiguration werden die Signaturdateien durch das System geprüft oder nicht. Besteht die Applikation nur aus einem Archiv, so sind die Dateien folgendermassen benannt:

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

Besteht die Applikation aus mehreren Archiven, werden die Dateinamen folgendermassen gebildet:

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

Auf diese Weise kann eine Applikation aus bis zu 10 Archiven bestehen. Das System liest die Archive beginnend bei Nummer 1 ein und nimmt als letztes das Archiv mit dem Buchstaben 'n' anstelle einer Ziffer.

5.5 Lizenzen

Mittels von OPTRONIC AG vergebener Lizenzen werden System- und Applikationsfunktionen freigeschaltet. Lizenzen erhalten Sie von OPTRONIC AG immer in einer Datei mit dem Namen `<target_name>key.txt`. Diese kann mittels eines üblichen Texteditors eingesehen werden. Sie ist folgendermassen aufgebaut:

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

Innerhalb der Anführungszeichen auf der rechten Seite neben dem Gleichheitszeichen (=) steht ein von OPTRONIC AG vergebener Text in Anführungszeichen (symbolisch durch `"... "` dargestellt), der spezifisch für das System und die freizuschaltende Applikation ist. Die letzte Zeile enthält in der oben dargestellten Lizenzdatei den eigentlichen Lizenzschlüssel. Sind mehrere Lizenzschlüssel vorhanden, dann ist pro Zeile jeweils ein Schlüssel untergebracht und die Datei sieht folgendermassen aus:

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

Der Buchstabe 'n' ist dabei ein Platzhalter für die grösste vorkommende Nummer. Beispielsweise könnte `KEY1` ein Systemschlüssel und `KEY2` ein Applikationsschlüssel sein. Mit Ausnahme der mit dem Zeichen '#' beginnenden Kommentarzeile besteht jede Zeile aus Zuweisungen von Werten (Zeichenketten) in Anführungszeichen zu Variablen:

```
VARIABLENNAME = "WERT"
```

Die Variablen haben folgende Bedeutung:

Name	Bedeutung	Beispielswert
TARGET	Der Name des Systems.	"D500"
SERIAL	Die Seriennummer des Geräts.	"12345"
PRODUCT	Der Name des Produkts.	"D512/34"
DATE	Datum an dem die Lizenz erstellt wurde.	"25.04.2003"
ID	Identifikationsnummer des Systems.	"1122-3344-5566-7788"
KEY	Lizenzschlüssel.	"1122-3344-5566-7788-99AA-BBCC-DDEE-FF00-1122-3344-5566-7788"
KEY1, ..., KEYn	Erster Lizenzschlüssel bis n-ter Lizenzschlüssel.	"1122-3344-5566-7788-99AA-BBCC-DDEE-FF00-1122-3344-5566-7788"

Eine Bedienstation kann die Identifikationsnummer des Systems und - sofern vorhanden - den Applikationsschlüssel anzeigen. Ist sie beispielsweise an eine D500 angeschlossen, so zeigt sie normalerweise auch deren Identifikationsnummer und Applikationsschlüssel an. Ein Lizenzschlüssel funktioniert genau für ein System. Er kann nicht auf ein anderes System übertragen werden. Sie können die Identifikationsnummer des Systems, für das die Lizenz erstellt wurde, der Variable `ID` entnehmen.

Die in Kapitel 3 erwähnte Datei `key.txt` ist identisch mit der hier vorgestellten Datei `<target_name>key.txt`. Beim wie in Kapitel 4.4 beschriebenen Laden einer neuen Lizenzdatei wird die im Datenbereich `config` abgelegte Datei `key.txt` durch `<target_name>key.txt` ersetzt.

5.6 System-Konfigurationsdaten

Bei Auslieferung ist das System normalerweise optimal für Ihre Bedürfnisse konfiguriert. Es kann vorkommen, dass OPTRONIC AG nicht im Besitz aller notwendigen Informationen war beziehungsweise ist, oder dass eine Anpassung an eine neue Umgebung notwendig wird. Über die bei OPTRONIC AG für Ihr System erhältliche, oder über das Skript 'Store Configuration' aus dem System auslesbare Datei mit dem Namen `<target_name>cfg.txt` können mittels eines üblichen Texteditors diverse Einstellungen geändert werden. Jede Zeile dieser Datei besteht aus einer Zuweisung von einem Wert (Zahl oder Text in Anführungszeichen) zu einer Variable:

```
VARIABLENNAME="TEXT"  
VARIABLENNAME=ZAHL
```

Die in Kapitel 3 erwähnte Datei `config.txt` ist identisch mit der hier vorgestellten Datei `<target_name>cfg.txt`. Beim wie in Kapitel 4.4 beschriebenen Laden wird die im Datenbereich `config` abgelegte Datei `config.txt` durch `<target_name>cfg.txt` ersetzt.

5.7 Applikations-Konfigurationsdaten

Applikations-Konfigurationsdaten werden dem System in der Datei `<target_name>par.txt` übergeben. Aufbau und Format der Datei entsprechen der im vorherigen Abschnitt besprochenen Datei `<target_name>cfg.txt`:

```
VARIABLENNAME="TEXT"  
VARIABLENNAME=ZAHL
```

Das Laden von Applikations-Konfigurationsdaten über einen USB-Stick wird in der Regel als einfache Möglichkeit verwendet, um Daten an die Applikation zu übergeben, die nicht anderweitig (z.B. über die Visualisierungssoftware oder den Profibus) eingegeben werden können.

Ein Beispiel dafür ist das Setzen der Profibus-Stationsadresse der D500, wenn keine Bedienstation an das System angeschlossen ist und die Bedienung vollständig über den Profibus erfolgt. Die Stationsadresse muss in diesem Fall gesetzt sein, bevor über den Profibus kommuniziert werden kann. Der Anwender kann die Adresse in die Datei `<target_name>par.txt` eingetragen, die Datei auf einen USB-Stick kopieren und sie danach ins System laden.

5.8 Skripte

Skripte werden dazu verwendet, um Systemanpassungen vorzunehmen, Systemtests auszuführen und um Systemstatusinformationen abzufragen. Eine Datei, welche ein Skript enthält, hat einen der folgenden Namen:

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

`<target_name>cmd.txt` und `<target_name>cmd.tgz` enthalten Skripte in einer speziellen für OAGLinux entwickelten Hochsprache, die als Resultat immer eine oder mehrere Dateien auf dem verwendeten USB-Stick erstellen. Achten Sie daher darauf, dass der USB-Stick ausreichend freien Speicherplatz bietet. Benötigt werden neben der eigentlichen Skript-Datei unter Umständen die zugehörige Signatur in der Datei `<target_name>cmd.sig` oder `<target_name>run.sig`, sowie die Passwortdatei `<target_name>pwd.txt`. Erzeugt wird von `<target_name>cmd.txt` und `<target_name>cmd.tgz` auf jeden Fall die Datei `status.txt`, die Informationen über die Abarbeitung des Skripts beinhaltet.

In der Regel bekommen Sie zusammen mit Ihrem System einige allgemeine auf Ihrem System ausführbare Skripte mitgeliefert. Jedes Skript ist dabei zusammen mit den ebenfalls benötigten Dateien in einem separaten Ordner untergebracht.



Der folgenden Tabelle können Sie alle notwendigen Informationen über die vorhandenen Skripte entnehmen:

Skriptname: Modem Dial Number
Ordner: **scripts/hardware/modem/dialnum**
Funktion: Wählt für 20 Sekunden die in der Datei **number.txt** vorgegebene Nummer.
unterstützte Systeme: Systeme mit Modem
zusätzlich benötigte Dateien: **number.txt**
erzeugte Dateien: **status.txt, scripts.txt**

Skriptname: Modem Test Tone
Ordner: **scripts/hardware/modem/testtone**
Funktion: Überprüft 10 Sekunden lang, ob das Modem das Freizeichen erkennt.
unterstützte Systeme: Systeme mit Modem
zusätzlich benötigte Dateien: keine
erzeugte Dateien: **status.txt, modem.txt**

Skriptname: Clear NVRAM
Ordner: **scripts/hardware/nvram/clear**
Funktion: Setzt alle Speicherzellen im nichtflüchtigen Speicher (NVRAM) auf 0.
unterstützte Systeme: D500
zusätzlich benötigte Dateien: keine
erzeugte Dateien: **status.txt, nvram.txt**

Skriptname: Load NVRAM
Ordner: **scripts/hardware/nvram/load**
Funktion: Lädt den Inhalt der Datei **nvram.dat** in den nichtflüchtigen Speicher (NVRAM).
unterstützte Systeme: D500
zusätzlich benötigte Dateien: **nvram.dat**
erzeugte Dateien: **status.txt**

Skriptname: Store NVRAM
Ordner: **scripts/hardware/nvram/store**
Funktion: Speichert den Inhalt des nichtflüchtigen Speichers (NVRAM) in die Datei **nvram.dat** ab.
unterstützte Systeme: D500
zusätzlich benötigte Dateien: keine
erzeugte Dateien: **status.txt, nvram.dat**

Skriptname: Test NVRAM
Ordner: **scripts/hardware/nvram/test**
Funktion: Testet den nichtflüchtigen Speicher (NVRAM) auf Fehler. Der Inhalt des Speichers wird nicht zerstört.
unterstützte Systeme: D500
zusätzlich benötigte Dateien: keine
erzeugte Dateien: **status.txt**

Skriptname: Test OMI/OIPs 13Bit, Binary
Ordner: **scripts/hardware/omiips/test13b**
Funktion: Testet die Hardware auf dem Einschub OMI500 und den OIP500-Modulen. Die Absolutgeber-SSI-Schnittstellen sind alle für 13Bit-Betrieb und binäre Übertragungsart konfiguriert.
unterstützte Systeme: D500 mit OMI500 und eventuell OIP500 Module
zusätzlich benötigte Dateien: keine
erzeugte Dateien: **status.txt, oagio.txt**



- Skriptname:** Test OMI/OIPs 13Bit, Gray
Ordner: `scripts/hardware/omiips/test13g`
Funktion: Testet die Hardware auf dem Einschub OMI500 und den OIP500-Modulen. Die Absolutgeber-SSI-Schnittstellen sind alle für 13Bit-Betrieb und gray-codierte Übertragungsart konfiguriert.
unterstützte Systeme: D500 mit OMI500 und eventuell OIP500 Module
zusätzlich benötigte Dateien: keine
erzeugte Dateien: `status.txt, oagio.txt`
- Skriptname:** Test OMI/OIPs 25Bit, Binary
Ordner: `scripts/hardware/omiips/test25b`
Funktion: Testet die Hardware auf dem Einschub OMI500 und den OIP500-Modulen. Die Absolutgeber-SSI-Schnittstellen sind alle für 25Bit-Betrieb und binäre Übertragungsart konfiguriert.
unterstützte Systeme: D500 mit OMI500 und eventuell OIP500 Module
zusätzlich benötigte Dateien: keine
erzeugte Dateien: `status.txt, oagio.txt`
- Skriptname:** Test OMI/OIPs 25Bit, Gray
Ordner: `scripts/hardware/omiips/test25g`
Funktion: Testet die Hardware auf dem Einschub OMI500 und den OIP500-Modulen. Die Absolutgeber-SSI-Schnittstellen sind alle für 25Bit-Betrieb und gray-codierte Übertragungsart konfiguriert.
unterstützte Systeme: D500 mit OMI500 und eventuell OIP500 Module
zusätzlich benötigte Dateien: keine
erzeugte Dateien: `status.txt, oagio.txt`
- Skriptname:** Test OMI/OIPs Relays
Ordner: `scripts/hardware/relays/test`
Funktion: Schliesst und öffnet die Relays auf dem Einschub OMI500 und den OIP500-Modulen im Sekundentakt je fünf mal.
unterstützte Systeme: D500 mit OMI500 und eventuell OIP500 Module
zusätzlich benötigte Dateien: keine
erzeugte Dateien: `status.txt`
- Skriptname:** Test Watchdog
Ordner: `scripts/hardware/watchdog/test`
Funktion: Lässt den Watchdog während 5 Sekunden pro Sekunde einmal kurz aufleuchten.
unterstützte Systeme: D500
zusätzlich benötigte Dateien: keine
erzeugte Dateien: `status.txt`
- Skriptname:** Delete Application
Ordner: `scripts/system/delapp`
Funktion: Löscht die auf dem System geladene Applikation mit allen ihren Daten und re-initialisiert die Datenbereiche `app` und `nvdata` auf der CompactFlash Card.
unterstützte Systeme: alle
zusätzlich benötigte Dateien: keine
erzeugte Dateien: `status.txt`
- Skriptname:** Delete NVDATA
Ordner: `scripts/system/delnvdat`
Funktion: Re-initialisiert den Datenbereich `nvdata` auf der CompactFlash Card.
unterstützte Systeme: alle
zusätzlich benötigte Dateien: keine
erzeugte Dateien: `status.txt`



Skriptname: Delete System Configuration and Licenses
Ordner: **scripts/system/delcfgky**
Funktion: Löscht die beiden Dateien **config.txt** und **key.txt** und re-initialisiert den Datenbereich **config** auf der CompactFlash Card.
unterstützte Systeme: alle
zusätzlich benötigte Dateien: keine
erzeugte Dateien: **status.txt**

Skriptname: Get System Status
Ordner: **scripts/system/getstat**
Funktion: Ermittelt eine grosse Menge an Systemstatusinformationen die zur Problemanalyse herbeigezogen werden können und speichert sie in der Datei **system.txt**.
unterstützte Systeme: alle
zusätzlich benötigte Dateien: keine
erzeugte Dateien: **status.txt, system.txt**

Skriptname: Store Application
Ordner: **scripts/system/storeapp**
Funktion: Speichert die auf dem System momentan installierte Applikation im Archiv **app.tgz**.
unterstützte Systeme: alle
zusätzlich benötigte Dateien: keine
erzeugte Dateien: **status.txt, app.tgz**

Skriptname: Store Configuration
Ordner: **scripts/system/storecfg**
Funktion: Speichert die System-Konfigurationsdaten in der Datei **config.txt**.
unterstützte Systeme: alle
zusätzlich benötigte Dateien: keine
erzeugte Dateien: **status.txt, config.txt**

Skriptname: Store Licenses
Ordner: **scripts/system/storekey**
Funktion: Speichert die geladenen Lizenzen in der Datei **key.txt**.
unterstützte Systeme: alle
zusätzlich benötigte Dateien: keine
erzeugte Dateien: **status.txt, key.txt**

Alle diese Skripte können nur während dem Systemstart und nicht im Betrieb ausgeführt werden. Der Aufbau und Inhalt der erzeugten Dateien wird in diesem Dokument nicht erläutert. Sie sind grösstenteils selbst erklärend. **system.txt** enthält viele Systeminformationen, die in der Regel nur von OPTRONIC AG ausgewertet werden können. Die über das Skript 'Store Application' in der Datei **app.tgz** abgespeicherte Applikation darf nicht ohne weiteres wieder ins System zurück geladen werden, da beim Aufspielen einer neuen Applikation unter Umständen zusätzliche Schritte notwendig sein könnten. Die Datei sollte nur für rein informative Zwecke verwendet werden.
Halten Sie bei Unsicherheiten im Umgang mit Skripten bitte unbedingt Rücksprache mit OPTRONIC AG!



6 Per Lizenz freischaltbare Systemdienste

Folgende Dienste können per Lizenz für ein System freigeschaltet werden:

- Web-Server
- SMB-Server
- NFS-Server
- FTP-Server
- FTP-Client
- Email-Client
- Netzwerkdrucker
- Fernwartung über Modem

Meistens ist es allerdings nicht alleine mit dem Freischalten der Funktion getan. Beispielsweise müssen die über eines der obigen Protokolle auszutauschenden Daten in der Regel definiert und von der Applikation generiert beziehungsweise ausgewertet werden.

7 Beispiele: Zur Softwarewartung notwendige Dateien

7.1 Einspielen neuer Applikationssoftware

7.1.1 Bedienstationen E2000 und E3000

Das Einspielen neuer Applikationssoftware ist für beide Bedienstationen gleich und wird anhand der E3000 vorgestellt.

Dateiname	Inhalt
e3000app.tgz	Die eigentliche Applikationssoftware.
e3000app.sig	Signatur zu Datei e3000app.tgz.
e3000run.sh	Skript zum Laden des Bildes in der Datei logo.img.
e3000run.sig	Signatur zu Datei e3000run.sh.
logo.img	Auf der Titelseite angezeigtes Bild.
e3000pwd.txt	Passwortdatei, die üblicherweise von OPTRONIC AG mit dem Passwort '176671' ausgeliefert wird. Das Passwort kann über die Visualisierungssoftware geändert werden. Die Datei wird nicht benötigt, wenn der Softwareupdate über die Visualisierungssoftware erfolgt, da dann das Passwort über die Tastatur eingegeben werden muss.
e3000key.txt	Datei mit neuen Lizenzen. Wird nur benötigt, wenn neue Optionen erworben worden sind oder ein grösserer Softwarewechsel vollzogen werden soll.
e3000cfg.txt	Datei mit neuer Systemkonfiguration. Wird nur benötigt, wenn grundlegende Anpassungen durchgeführt werden müssen.

7.1.2 CNC D500

Dateiname	Inhalt
d500app.tgz	Die eigentliche Applikationssoftware.
d500app.sig	Signatur zu Datei d500app.tgz.
d500key.txt	Datei mit neuen Lizenzen. Wird nur benötigt, wenn neue Optionen erworben worden sind oder ein grösserer Softwarewechsel vollzogen werden soll.
d500cfg.txt	Datei mit neuer Systemkonfiguration. Wird nur benötigt, wenn grundlegende Anpassungen durchgeführt werden müssen.



7.2 Einspielen von System-Konfigurationsdaten und Lizenzen

Das Einspielen von System-Konfigurationsdaten und Lizenzen ist bei allen Systemen gleich und wird anhand der CNC D500 vorgestellt. Es ist von Fall zu Fall verschieden, ob nur eine der beiden Dateien notwendig ist oder ob beide vorhanden sein müssen.

Dateiname	Inhalt
d500key.txt	Datei mit neuen Lizenzen. Wird benötigt, wenn neue Optionen erworben worden sind oder ein grösserer Softwarewechsel vollzogen werden soll.
d500cfg.txt	Datei mit neuer Systemkonfiguration. Wird benötigt, wenn grundlegende Anpassungen durchgeführt werden müssen.

7.3 Einspielen von Applikations-Konfigurationsdaten

Das Einspielen von Applikations-Konfigurationsdaten ist bei allen Systemen gleich und wird anhand der CNC D500 vorgestellt.

Dateiname	Inhalt
d500par.txt	Datei mit neuer Applikations-Konfiguration.

7.4 Ausführen von Systemtests und Systemanpassungen, Abfragen von Systemstatusinformationen

7.4.1 Bedienstationen E2000 und E3000

Das Ausführen von Systemtests und Systemanpassungen, sowie das Abfragen von Systemstatusinformationen ist für beide Bedienstationen gleich und wird anhand der E3000 vorgestellt. Als Beispiel soll überprüft werden, ob das integrierte Modem korrekt mit dem Telefonnetz verbunden ist. Dazu wird über das Skript 'Modem Dial Number' für 20 Sekunden eine vorgegebene Telefonnummer angewählt.

Dateiname	Inhalt
e3000cmd.tgz	Skript-Archiv.
e3000cmd.sig	Signatur zu Datei e3000cmd.tgz.
e3000pwd.txt	Passwortdatei, die üblicherweise von OPTRONIC AG mit dem Passwort '176671' ausgeliefert wird. Das Passwort kann über die Visualisierungssoftware geändert werden. Die Datei wird benötigt, da das Skript über die Reset-Taste und den Reset/Load-Schalter gestartet werden muss.
number.txt	Datei, welche die zu wählende Nummer enthalten muss.

7.4.2 CNC D500

Als Beispiel sollen zur Problemanalyse mit dem Skript 'Get System Status' Systemstatusinformationen abgefragt werden.

Dateiname	Inhalt
d500cmd.tgz	Skript-Archiv.
d500cmd.sig	Signatur zu Datei d500cmd.tgz.