

mkc

MKC Michels & Kleberhoff Computer GmbH

Vohwinkeler Str. 58, D-42329 Wuppertal

Tel.: ++49 (0)202 27317 0 Fax: ++49 (0)202 27317 49

Internet: <http://www.mkc-gmbh.de>



Software Handbuch

Linux Systeme basierend auf eNetMaxi

Hinweise:

Die Informationen in diesem Handbuch wurden sorgfältig zusammengestellt und überprüft. Dieses Handbuch wird stetig auf dem aktuellen Zustand gehalten. Jedoch wird von MKC keine Gewähr für fehlerhafte Informationen übernommen.

MKC behält sich das Recht vor, jederzeit ohne weitere Ankündigung technische Änderungen zur Verbesserung der Zuverlässigkeit, der Funktion oder des Designs der Produkte und Überarbeitungen des Handbuchs durchzuführen. Änderungen des Handbuchs zwischen 2 Ausgaben werden im Text nicht markiert.

Das Datum einer Ausgabe bezieht sich auf das Handbuch. Dieses muss nicht mit dem Datum der Änderung der Hardware oder Software übereinstimmen. Bei der Versionsgeschichte wird der Grund für die Handbuch Änderungen genannt.

MKC übernimmt keine Haftung für die Anwendung des hier beschriebenen Produktes. MKC übernimmt weiterhin keine Haftung für Schäden oder Folgeschäden, die durch Verwendung dieses Produktes entstehen. Diese Haftungseinschränkung bezieht sich sowohl auf jeden direkten Abnehmer sowie auf alle seine Kunden und alle Anwender des Produktes.

Es gelten ausschließlich die in diesem Dokument gemachten Zusagen über die Anwendbarkeit des hier beschriebenen Produktes.

Kommentare:

Kommentare oder Korrekturen jedweder Art sind dem Autor jederzeit willkommen. Senden Sie diese bitte an:

**MKC Michels & Kleberhoff Computer GmbH
Vohwinkeler Str. 58
42329 Wuppertal**

oder

info@mkc-gmbh.de

Handbuch Versionen

Änderungen im Handbuch werden durch eine Erhöhung der Ausgabennummer angezeigt. Handbücher, deren Ausgabe durch einen Buchstaben gekennzeichnet ist, sind vorläufige Handbücher und stimmen möglicherweise noch nicht vollständig mit dem endgültigen Produkt überein. Die erste Ausgabe, die nicht mehr als vorläufig anzusehen ist, beginnt mit der Nummerierung „1“.

Handbuch Versionen			
Ausgabe	Änderungen	Datum	
A	Initial, Übernahme von Handbuch für eNetVario Rev. 1	02.07.19	
B	Korrekturen, Ergänzungen	14.08.19	
C	Überarbeitungen	12.09.19	
1	Erste Ausgabe	14.10.19	
1A	Korrekturen	17.06.20	
2	Debian 10 „Buster“	16.08.21	

Inhaltsverzeichnis

1 EINLEITUNG.....	7
1.1 Hinweise zu Angaben in diesem Handbuch.....	7
2 ALLGEMEIN.....	9
3 BOOTLOADER.....	11
3.1 Zusammensetzung.....	11
3.2 Boot Partition.....	11
3.3 Ablauf.....	11
3.4 SPI-Flash	12
4 LINUX.....	13
4.1 Rootfs Partition.....	13
4.2 Kernel und Treiber.....	13
4.2.1 Kernel-Treiber.....	13
4.2.2 Treiber für das eNetMaxi-Modul.....	14
4.2.3 MTD Treiber.....	15
4.2.4 Ein Beispiel-Treiber für die Trägerkarte.....	16
4.3 Debian Konfiguration.....	17
4.3.1 Benutzer.....	17
4.3.2 Dienste	17
4.3.3 Netzwerk.....	17
4.3.4 Paket-Management.....	18
5 AUSLIEFERZUSTAND.....	19
5.1 Beispiel für ein eNetMaxi auf MKC1502-Trägerkarte:.....	19

Liste der Abbildungen

Abbildung 1: System.....	9
Abbildung 2: Partitionen [eMMC/uSD].....	9
Abbildung 3: Bootprozess.....	11
Abbildung 4: Flash-Aufbau.....	15

Liste der Tabellen

Tabelle 1: Definitionen.....	8
Tabelle 2: SPI-Flash Startadressen.....	12
Tabelle 3: Schutzfunktionen.....	12
Tabelle 4: DTS.....	13
Tabelle 5: Modul Treiber, Übersicht.....	14
Tabelle 6: Extension Treiber, Übersicht.....	16

1 Einleitung

Willkommen beim MKC Software Handbuch für das eNetMaxi Modul. Diese Anleitung gibt Ihnen einen schnellen Einstieg in unser Debian Linux basierendes Betriebssystem.

Für das Verständnis dieser Dokumentation sind allgemeine Linux und Programmierkenntnisse zwingend erforderlich. Aus diesem Grund wird angenommen, dass die Nutzer mit einem Embedded Linux vertraut sind. Wir helfen Ihnen gerne, dieses Wissen zu erwerben. Wenn Sie Interesse an Schulungen haben, oder sich Unterstützung wünschen, kontaktieren Sie uns bitte.

Dieser Leitfaden zeigt Ihnen, wie Sie mit der Arbeit mit Ihrem Zielsystem starten können. Es wird gezeigt, wie Sie eine Netzwerkverbindung einrichten und ihre Anwendungen auf dem Zielsystem bereitstellen können. Die Beispiele in diesem Handbuch basieren auf spezifischer Hardware; wenn nicht anders erwähnt, funktionieren diese auf allen unterstützten MKC Geräten mit unserem angepasstem Debian-Linux Betriebssystem.

1.1 Hinweise zu Angaben in diesem Handbuch

Haftungsausschluss

Dieses Dokument enthält Links zu externen Websites Dritter, auf deren Inhalte wir keinen Einfluss haben. Deshalb können wir für diese fremden Inhalte auch keine Gewähr übernehmen. Für die Inhalte der verlinkten Seiten ist stets der jeweilige Anbieter oder Betreiber der Seiten verantwortlich. Die verlinkten Seiten wurden zum Zeitpunkt der Verlinkung auf mögliche Rechtsverstöße überprüft. Rechtswidrige Inhalte waren zum Zeitpunkt der Verlinkung nicht erkennbar.

Eine permanente inhaltliche Kontrolle der verlinkten Seiten ist jedoch ohne konkrete Anhaltspunkte einer Rechtsverletzung nicht zumutbar. Bei Bekanntwerden von Rechtsverletzungen werden wir derartige Links umgehend entfernen.

Zahlenangaben

Hexadezimale Zahlen werden in diesem Handbuch durch ein vorangestelltes Dollarzeichen „\$“ gekennzeichnet. Andere geläufige Schreibweisen für Hexadezimale Zahlen sind z.B. durch den Präfix „0x“ oder den Suffix „h“ in der Literatur angegeben. Sie werden hier nur der Vollständigkeit halber erwähnt.

Um die Lesbarkeit von langen hexadezimalen Zahlen zu verbessern, werden diese von rechts durch einen Punkt in 4er Gruppen unterteilt. Eine mathematische Bedeutung liegt diesem Punkt nicht zugrunde.

Lieferversionen

Die unten angegebenen Lieferversionen sind zur Zeit verfügbar. Damit ist nicht zugesagt, dass alle diese Versionen weiterhin lieferbar bleiben. MKC behält sich das Recht vor, die Produktion dieser Hardware oder Software aus technischen Gründen ohne vorherige Ankündigung einzustellen.

Vorläufige Angaben

In dieser Handbuchversion sind mehrere Kapitel noch vorläufig, diese Stellen sind mit dem Textzusatz '*TDB: ...*' an den entsprechenden Stellen gekennzeichnet.

Definitionen

Abkürzung/Def.	Beschreibung
DTB	Device tree (binär)
DTS	Device tree (ASCII)
MLO	Secondary program loader. Lädt und Startet das „u-boot“ .
MTD	Memory technology device; Linux Bezeichnung für NAND/NOR Flash
uSD	microSD Karte
TBD	Funktion ist zur Zeit noch nicht definiert.

Tabelle 1: Definitionen

2 Allgemein

Ein vollständiges System für ein eNetMaxi Modul besteht aus einem Bootloader, einem Linux-Kernel und einer Linux-Distribution (Debian). In diesem Dokument werden alle relevanten Teile beschrieben.

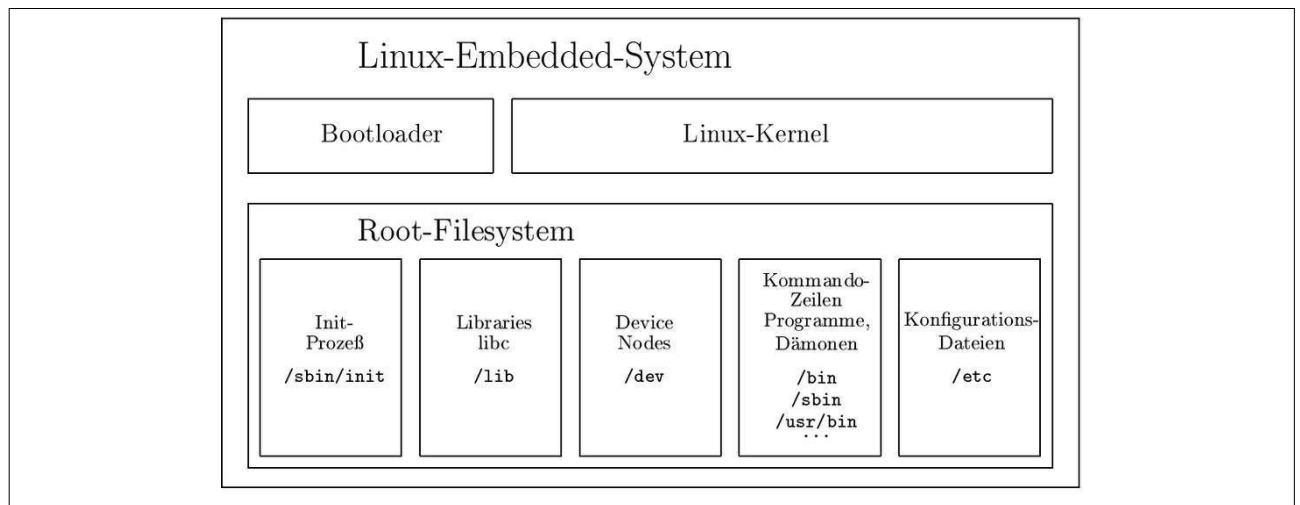


Abbildung 1: System

Als Bootloader verwenden wir „u-boot“ von DENX mit Erweiterungen von Texas Instruments (TI) und MKC. Die aktuell eingesetzte Version ist u-boot-2016.03. Die vollständige Beschreibung befindet sich innerhalb der Quellen; die jeweils aktuelle Dokumentation ist bei DENX im Internet zu finden.

Als Betriebssystem setzen wir Debian ein; eine der meistbenutzten Linux Distributionen. Im Internet findet man sehr gute und detaillierte Informationen auf www.debian.org oder wiki.debian.org.

Um das Arbeiten mit dem eNetMaxi Modul so einfach als möglich zu gestalten, empfehlen wir für den Entwicklungsrechner (Host PC) ebenfalls ein Debian Linux oder Derivat einzusetzen; z.B. in einer virtuellen Maschine. Einen Leitfaden über die Einrichtung und Konfiguration einer solchen virtuellen Maschine finden Sie auf unserer Homepage.

Um ein System auf dem eNetMaxi zu installieren bzw. auszuführen, werden unterschiedliche Speichermedien eingesetzt. Spezifische technische Informationen der Speichermedien können Sie den technischen Handbüchern des Moduls und/oder der Trägerkarte entnehmen. In einem lauffähigem System können die Informationen per Software direkt ausgelesen werden.

Auf dem Modul selbst ist ein eMMC FLASH (≥ 4 GB) und ein serielles SPI-FLASH (≥ 1 MB) vorhanden. Auf der Trägerkarte für das eNetMaxi Modul sollte eine Fassung für eine uSD-Karte zur Verfügung stehen. Als Speichermedien für das System wird entweder eine uSD-Karte oder das eMMC-Flash auf dem eNetmaxi eingesetzt. Auf einem Medium sind mindestens die zwei Partitionen (Boot und Rootfs) vorhanden

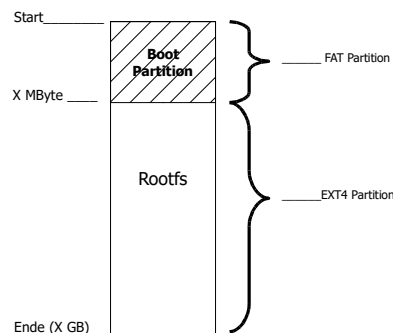


Abbildung 2: Partitionen [eMMC/uSD]

3 Bootloader

Es wird ein auf das eNetMaxi Modul angepasster Bootloader (u-boot mit MLO) verwendet um den Kernel zu laden. Der Kernel lädt das Linux-Betriebssystem und weitere Treiber während der Initialisierung. Benötigte Konfigurationsdateien befinden sich in den Linux üblichen Verzeichnissen und Dateien. Die Initialisierung wird in einem Debian Linux über „systemd“ realisiert.

3.1 Zusammensetzung

Der Bootloader besteht aus zwei Dateien: MLO und u-boot. Der MLO fungiert als „Loader“ des eigentlichen und vollständigen Bootloader „u-boot“. Der MLO wird aus den Quellen des „u-boot“ generiert. In der Namens-Konvention von DENX wird der MLO auch als SPL bezeichnet. Der MLO wird benötigt um die initiale Konfigurationen von Hardware Komponenten (CPU, RAM, ...) vorzunehmen.

3.2 Boot Partition

Die FAT Partition der uSD enthält eine Boot-Partition mit MLO, u-boot und der Konfigurationsdatei uEnv.txt.

Die FAT Partition des eMMC enthält Kopien der MLO und u-boot Dateien. Diese dienen nur als Sicherheitskopie. Sie werden zum booten nicht benötigt. Die Originaldateien, die zum booten benutzt werden, befinden sich im SPI-Flash.

3.3 Ablauf

Ausgehend vom ROM Code der CPU wird ein System über mehrere Stufen geladen und gestartet. Es sind zwei Medien als Systemdatenträger vorgesehen: eine uSD Karte und das eMMC Flash. Die uSD ist immer höher priorisiert. Somit wird – sofern sich ein MLO auf der uSD befindet – dieser geladen und ausgeführt, der Systemstart erfolgt vollständig von der uSD. Ist das vollständige Betriebssystem auf dem eMMC gespeichert (und keine bootfähige uSD eingelegt), wird der komplette Bootloader aus dem SPI-Flash des eNetMaxi Moduls gelesen und ausgeführt. Der Bootloader ist im Auslieferungszustand des Moduls in das SPI-Flash programmiert und dieses Flash wird standardmäßig mit gesetztem Schreibschutz ausgeliefert. Der MLO steht somit stets zur Verfügung.

Nach erfolgreicher Ausführung des Bootloader wird als letzter Schritt das Linux Betriebssystem gestartet bzw. der Linux Kernel (auf der uSD oder eMMC) geladen und ausgeführt.

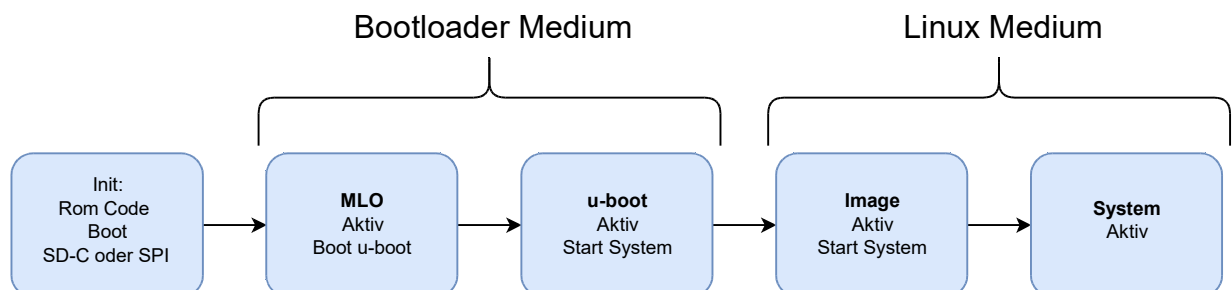


Abbildung 3: Bootprozess

Medien:

- Bootloader Medium: uSD Karte oder integrierter SPI-Flash Baustein auf dem Modul
- Linux Medium: uSD Karte oder integrierter Datenträger eMMC auf dem Modul

3.4 SPI-Flash

Auf dem SPI-Flash speichern wir den Bootloader und gerätespezifische Firmware-Daten. Die Startadressen sind in Tabelle 2. aufgeführt und orientieren sich an den physikalischen Sektorgrenzen.

Name	Startadresse	Bemerkung
MLO	0x0000	
u-boot	0x10000	
u-boot (Environment)	0x70000	Fix auf 4 KiB gesetzt
Firmware (MKC)	0x80000	Als MTD Partition nutzbar mit FS: JFFS2

Tabelle 2: SPI-Flash Startadressen

Die Speicherkapazität des Flash-Bausteins ist dem Technischen Handbuch zum eNetMaxi Modul zu entnehmen.

Hinweis:

Bei Drucklegung wurde ein Micron M25PE80 (8MBit) eingesetzt.

Beim Start des u-boot aus dem SPI-Flash ist immer eine der folgende Funktionen aktiviert um den Bootloader im laufenden Betrieb zu schützen.

Funktionen	
Art	Beschreibung
Schreibschutz (WP)	Der gesamte Flash-Baustein ist nicht beschreibbar; aufheben des Schreibschutzes für Update/Upgrade-Zwecke möglich.
WP + Sperre	Der Schreibschutz kann zur Laufzeit nicht aufgehoben werden. Ist diese Sperre gesetzt, so kann der Schreibschutz nur durch einen Hardware-Umbau aufgehoben werden. Weiter Hinweise hierzu entnehmen Sie bitte dem 'Technischen Handbuch eNetMaxi.

Tabelle 3: Schutzfunktionen

4 Linux

4.1 Rootfs Partition

Diese Partition enthält das Linux Betriebssystem und ist mit EXT4 formatiert. Das EXT4 Journaling ist im Standardmodus auf „ordered mode“ gesetzt. Für komplettes Journaling ist der Modus „journal mode“ zu wählen; siehe in der EXT4 Dokumentation, in den Kernel Quellen unter /Documentation/ext4.txt.

Hinweis:

Es wird empfohlen das System immer mittels der Kommandos „shutdown“ oder „halt“ sicher herunterzufahren.

4.2 Kernel und Treiber

Die Kernkomponenten des eNetMaxi Moduls sind eine ARM CPU und weitere wichtige Bauteile (PMIC, RAM, ...), diese werden in aktuellen Kernel-Versionen über das „Device Tree“-Modell in einer oder mehreren ASCII-Dateien (DTS) beschrieben. Diese Beschreibung wird in eine einzige Binärdatei kompiliert und während der Start/Boot Phase des Kernels geladen. Bei einem Gesamtsystem bestehend aus einem eNetMaxi-Modul und einer Trägerkarte (Extension) gibt es eine – die Gesamt-Hardware definierende – Beschreibungs-Datei (DTS). Weiterhin existieren Treiberdateien für die einzelnen Hardwarekomponenten des Moduls und Trägerkarte.

DTS Beschreibungen	
eNetMaxi Modul	Im Verzeichnis /boot/dtbs auf der Rootfspartition wird ein symbolischer Link „enetmaxi.dtb“ gesetzt.
Link-Ziel	name_der_trägerkarte_[Variante].dtb Variante: z.B. STD für Standard.

Tabelle 4: DTS

Basis ist der Linux Kernel in der Version 4.1.13 von Texas Instruments mit Erweiterungen für unsere Hardware. Dieser Stand wurde auf die Version 4.19.59 aktualisiert.

4.2.1 Kernel-Treiber

In einem Linux Betriebssystem werden eine Reihe von Treibern eingesetzt; diese sind nachladbar oder 'build-in' im Kernel verankert. MKC Treiber werden als nachladbare Treiber ausgelegt und während des Bootprozesses geladen.

4.2.2 Treiber für das eNetMaxi-Modul

Allgemeiner Aufbau eines modularen Linux Treiber von MKC:

- Zustands-Abfrage/Kontrolle über Einträge im SYSFS
- Steuerung über Startparameter des Treibers

Der Treiber „mkcmod“ dient der Steuerung des eNetMaxi-Moduls und stellt die Funktionen zur Unterstützung der Modulhardware bereit. Für den Fall, dass der Inhalt des SPI-Flashes für ein Update/Upgrade der MKC Firmware überschrieben werden muss, kann der Modul-Treiber in einem besonderen Start-Modus gestartet werden. Hiermit wird der Schreibschutz des Flashes aufgehoben.

Zustands-Abfrage/Kontrolle unter /sys/class/mkcmodul/	
Bezeichnung	Beschreibung
enetmaxi_state	Status des Treibers
config_in	Lesen des /CONFIG_IN Pins
rst_out	Lesen/Schreiben des RST_OUT Pins
ti_macX	Ausgabe der hardwareseitig fest codierten MAC-Adressen aus den CPU-Registern
mod_hw_rev, mod_hw_var	Ausgabe der hardwareseitig fest codierten Variablen Revision und Variante des Moduls

Tabelle 5: Modul Treiber, Übersicht

Beispiele:

Der folgende Aufrufe liefern die Variablen für den Hardware-Stand des Moduls:

```
'cat /sys/class/mkcmod/mod_hw_rev'           Liefert Wert des Modul-Revision.
'cat /sys/class/mkcmod/mod_hw_var'         Liefert Wert des Modul-Variante.
```

Der folgende Aufruf liefert den Status des CONFIG_IN Pins:

```
'cat /sys/class/mkcmod/config_in'         Liefert Wert des CONFIG_IN Pins.
```

Der folgende Aufruf startet den Treiber manuell:

```
'modprobe mkcmod mode=0' auch 'modprobe mkcmod'
(wenn mode nicht angegeben ist wird der default mode=0 gesetzt)
Startet im Standardmodus.
Es ist ein weiterer Modus definiert.
(mode=4096 → Update/Upgrade, Schreibschutz aufgehoben, MTD-Treiber)
```

Der folgende Aufruf liefert den Treiber Status:

```
'cat /sys/class/mkcmod/enetmaxi_state'    Liefert den „mode“ Wert, im Standardmodus : 0
```

Hinweis:

Der Treiber wird im Betrieb automatisch geladen und im Standardmodus mit aktiviertem Schreibschutz gestartet.

4.2.3 MTD Treiber

Für das SPI-Flash wird der Linux MTD Treiber eingesetzt. Die prinzipielle Belegung kann der Abbildung 4 entnommen werden.

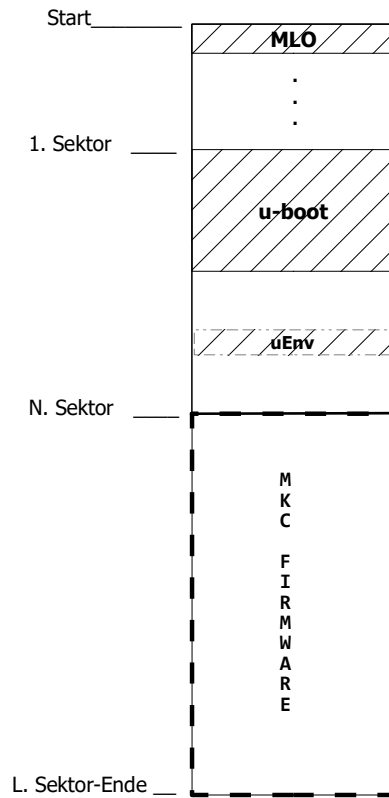


Abbildung 4: Flash-Aufbau

Auf dem MKC-Geräten wird der Schreibschutz bereits im Bootloader gesetzt. Der Standard Linux Treiber besitzt keine Funktionen um den Schreibschutz zu erkennen oder zu ändern.

4.2.4 Ein Beispiel-Treiber für die Trägerkarte

Wie bei dem eNetMaxi Modul werden auch für das Gerät spezifische Treiber benötigt. Es gibt mindestens einen Treiber für die Trägerkarte (Extension). Werden weitere Treiber benötigt, werden diese nach folgender Konvention benannt:

- mkcext_[FUNKTIONSGRUPPE].

Das Gerät eNetVario in der Variante STD benötigt keine zusätzlichen Extension-Treiber.

Der Treiber „mkcext“ dient der Steuerung der LEDs auf der Trägerkarte MKC1502 (Extension für das Gerät eNetVario):

Zustands-Abfrage/Kontrolle unter /sys/class/mkcext/	
Bezeichnung	Beschreibung
enetvario_state	Status des Treibers
led[Nr]_[Farbe]	Lesen/Schreiben des LED Zustandes

Tabelle 6: Extension Treiber, Übersicht

Der folgende Aufruf startet den Treiber manuell:

```
'modprobe mkcext' oder 'modprobe mkcext mode=0'
```

Startet im Standardmodus. Es sind weitere Modi definiert, siehe Kapitel 4.2.2.

Der folgende Aufruf liefert den Treiber Status:

```
'cat /sys/class/mkcext/enetvario_state'
```

Liefert den „mode“ Wert, im Standardmodus : 0

Hinweis:

Der Treiber wird im Betrieb automatisch geladen und im Standardmodus aktiviert.

Für die Steuerung der vier LED's [rt:rot | gr:grün] werden im Verzeichnis '/sys/class/mkcext/' Variablen angelegt.

Der folgende Aufruf liefert den LED 1 Zustand:

```
'cat /sys/class/mkcext/led1_gr'
```

Liefert Wert der LED 1 [gr] : 0 oder 1

Der folgende Aufruf schreibt den LED 1 Zustand:

```
'echo 1 > /sys/class/mkcext/led1_rt'
```

Setzt Wert der LED 1 [rt] : 1,
→ LED 1 leuchtet rot

Der folgende Aufruf liefert den LED PWR Zustand:

```
'cat /sys/class/mkcext/sys0_gr'
```

Liefert Wert der LED PWR [gr] : 0 oder 1

Hinweis:

Die LED PWR hat die Funktion [SYSOK]. Aus diesem Grunde wurde auf der Trägerkarte MKC1502 die Hardware so realisiert, dass eine Priorisierung der angezeigten Zustände vorgenommen wird. Wird **LED PWR [gr]** eingeschaltet, wird die **LED PWR [rt]** automatisch ausgeschaltet, analog dazu wird die **LED PWR [rt]** automatisch eingeschaltet, wenn die **LED PWR [gr]** ausgeschaltet ist.

4.3 Debian Konfiguration

Nachdem Start des Linux Kernels wird die installierte Rootfs-Partition aktiviert und das Betriebssystem mit der entsprechenden Konfiguration/Parametrierung gestartet. Die Konfiguration und Parametrierung des Betriebssystems wird wie im Linux üblich über den INIT Prozess (systemd) vorgenommen. Der Auslieferungszustand ist im Anhang beschrieben. Es wird ein Debian Linux in Version 10.10 (Buster) genutzt.

4.3.1 Benutzer

Auf allen von MKC ausgelieferten Linuxsystemen für das eNetMaxi-Modul ist standardmäßig der Linux Administrator 'root' eingerichtet. Das zugehörige Passwort lautet 'root'.

4.3.2 Dienste

Je nachdem, auf welcher Trägerkarte das eNetMaxi-Modul ausgeliefert wird, müssen beim Systemstart entsprechende spezifische Treiber/Programme gestartet werden. Die notwendigen Scripte sind im Verzeichnis `/etc/init.d` oder als ein systemd Service Unit installiert. Weiterhin können diese Scripte als Vorlage dienen, um eigene Dienste oder Programme bei Start/Stop des Systems zu aktivieren. Weitere Hinweise zu diesen Startdateien finden sich im gleichen Verzeichnis (Dateien README und skeleton).

Die systemd Service Units befinden sich im Verzeichnis `/lib/systemd/system`.

Detaillierte Informationen über den Aufbau finden Sie z.B. auf https://wiki.ubuntuusers.de/systemd/Service_Units/

4.3.3 Netzwerk

Permanente Änderung der Konfiguration des Netzwerks kann über das installierte Tool *connman* vorgenommen werden
Temporäre Änderung (bis zum Neustart des Betriebssystems) kann über *ifconfig* vorgenommen werden.

Der folgende Aufruf zeigt den aktuellen Zustand :

```
$SHELLPROMPT ifconfig
```

 Ausgabe der Informationen

Der folgende Aufruf konfiguriert die Schnittstelle eth0 mit einer IP Adresse und Netmask (temporär):

```
$SHELLPROMPT ifconfig eth0 192.168.15.22/24
```

Der folgende Aufruf startet den Interaktiven Modus:

```
$SHELLPROMPT connmanctl
```

 Ausgabe: **connmanctl**>

Der folgende Aufruf zeigt die verfügbaren Schnittstellen :

```
connmanctl> services
```

 Ausgabe der aktiven Schnittstellen

Der folgende Aufruf zeigt die aktuelle Konfiguration :

```
connmanctl> services ethernet_[MAC-Adresse]_cable
```

 Ausgabe der Konfiguration

Der folgende Aufruf konfiguriert eine Schnittstelle mit einer IP Adresse, Netmask und Gateway (permanent):

```
connmanctl> config ethernet_[MAC-Adresse]_cable --ipv4 manual 192.168.15.22 255.255.255.0 0.0.0.0
```

Der folgende Aufruf zeigt die verfügbaren Befehle :

```
connmanctl> help
```

 Ausgabe der Hilfe-Informationen

4.3.4 Paket-Management

Im Auslieferungszustand ist die Konfiguration für das Paket-Management-Tool (apt) auf einen Zugriff zu den offiziellen Debian Servern (ftp2.debian.org) voreingestellt. MKC Komponenten wie der Kernel, Kernel-Bibliotheken und der Bootloader stehen nicht unter Kontrolle des Paket-Managements.

5 Auslieferungszustand

Im Auslieferungszustand wird ein getestetes Modul ohne Bootloader / Linux geliefert.

Die Konfiguration und Parametrierung des zugehörige Betriebssystems ist abhängig von der eingesetzten Trägerkarte und wird mit dieser auf eine bootfähigen uSD-Karte geliefert. Weiterhin sind auf dieser uSD-Karte alle notwendigen Daten, Scripte und Programme vorhanden, um den Bootloader, Linux und Anwendungen auf dem modulinternen SPI-Flash und eMMC zu speichern.

5.1 Beispiel für ein eNetMaxi auf MKC1502-Trägerkarte:

Das Auslieferungssystem auf der bootfähigen uSD-Karte für ein eNetMaxi-Modul, welches auf einer MKC1502-Trägerkarte arbeitet, installiert auf dem SPI-Flash/eMMC das Linux-System mit folgenden Einstellungen:

Bootloader	MKC-u-boot im SPI-Flash
Linux	MKC-Linux (Kernel 4.19.59, Debian 10.10, MKC-Treiber) im eMMC
Programme	MKC Anpassungen und Anwendungen (Verzeichnisse /opt/mkc-extras/... und /opt/mkc/...)
Benutzer	'root' mit dem Kennwort 'root'
IP-Adressen	192.168.015.22 [ETH 0] und DHCP [ETH 1] Es ist kein Router/Gateway eingetragen, das Netzwerk wird im Auto Negotiation Modus betrieben.

Aktivierte Server

SFTP, SSH, HTTP(S), Node-RED, MQTT-Broker

Debian Paket Server

Der Debian Paket Server für Programm-Pakete und Updates ist in der Datei „/etc/sources.list“ sind auf die offiziellen Debian Server (ftp2.debian.org) voreingestellt.

Serielle Schnittstellen

Die erste serielle Schnittstelle [UART 0] ist als Konsole definiert mit den Parametern:

115200 Baud, 8 Bit, keine Parität, 1 Stopbit, keine Flussteuerung

Die zweite serielle Schnittstelle [UART 1] ist frei programmierbar und kann über die üblichen Linux Programme (für den Systemstart) konfiguriert werden. Im Auslieferungszustand erfolgt keine Parametrierung der Schnittstelle.

USB Schnittstellen

Die USB Schnittstellen sind hardwareseitig auf Host [USB0] und Slave [USB1] festgelegt. Im Auslieferungszustand sind beide Schnittstellen inaktiv und können über die üblichen Linux Programme aktiviert werden. Im Auslieferungszustand erfolgt keine Parametrierung der Schnittstellen.