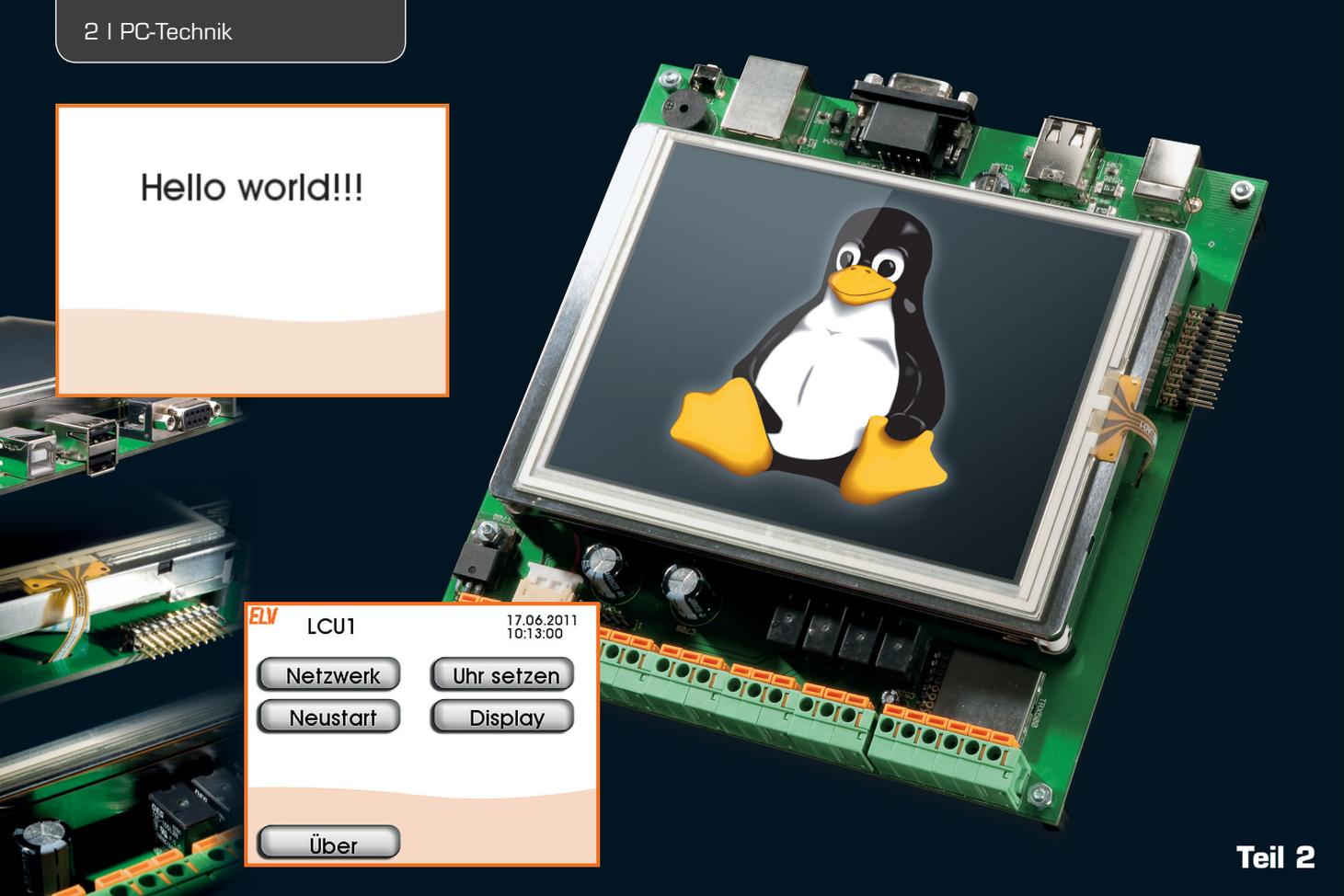




Hello world!!!

Teil 2

# LINUX-Control-Unit LCU1

## Benutzeroberfläche selbst erstellen

Mit diesem Online-Artikel, der die Oberflächen-Engine UI der LCU1 beschreibt, setzen wir den Bericht aus dem ELVjournal Oktober/November zur Beschreibung der LCU1 fort. Wir zeigen hier beispielhaft an einer einfachen Oberflächenseite, wie das XML-Format für die Oberflächenseiten aufgebaut ist.

### Beschreibung der Oberfläche

Jede Oberflächenseite besteht aus einer Liste von UI-Elementen. Diese werden in der Reihenfolge, in der sie in der XML-Datei vorgefunden werden, gezeichnet. Später angegebene UI-Elemente können also früher angegebene UI-Elemente überdecken. Da als Grafikelemente PNG-Dateien mit Transparenz verwendet werden, lassen sich dadurch auf einfache Weise nahtlose Oberflächenseiten erstellen.

### Unterstützte UI-Elemente

Die XML-Beschreibung der Oberflächenseiten unterstützt nur drei verschiedene Typen von UI-Elementen:

- Textelemente können Text in einer beliebigen Schriftart, die als TTF vorliegen muss, in einer beliebigen Größe anzeigen. Der Text kann zentriert oder linksbündig dargestellt werden.
- Grafikelemente können eine als PNG vorliegende Grafikdatei anzeigen. Es wird Transparenz unterstützt. Es lassen sich an einem Grafikelement mehrere Grafikdateien hinterlegen, zwischen denen im TCL-Code per Index umgeschaltet werden kann. Dadurch lassen sich z. B. Checkboxes einfach implementieren.
- Eine spezielle Art der Textelemente stellen DateTime-Elemente dar. Diese zeigen die aktuelle Zeit und das aktuelle Datum in einem benutzerdefinierbaren Format an. Die Darstellung wird automatisch aktualisiert.

### Ein einfaches Beispiel

Es soll beispielhaft anhand einer einfachen Oberflächenseite gezeigt werden, wie das XML-Format für die Oberflächenseiten aufgebaut ist.

Das folgende Beispiel befindet sich im Dateisystem der LCU unter:

```
/usr/share/ui/hello.xml
```

Um die Seite anzuzeigen, ist die UI-Engine wie folgt aufzurufen:

```
# uiengine -s hello
```

Sollte noch eine Instanz der UI-Engine laufen, muss diese zuvor mit

```
# killall uiengine
```

beendet werden. In [Bild 7](#) ist ein Screenshot der angezeigten Seite sowie der Quelltext der Seite zu sehen. Das Beispiel zeigt den grundlegenden Aufbau der Seitenbeschreibung. Im Folgenden werden die einzelnen Elemente der beschriebenen Seite näher erläutert.

```

<ui>
  <page id="hello" font="fonts/decker.ttf" font_height="30"
    onload="OnPageLoad">

    <elements>
      <image id="background" x="0" y="0"
        file="images/background.png" onclick="exit"/>
      <text id="text_1" x="160" y="50" height="30" center="true"/>
    </elements>

    <code><![CDATA[
      proc OnPageLoad { previousPage reason } {
        ui set text_1.text "Hello world!!!"
      }
    ]]></code>
  </page>
</ui>

```

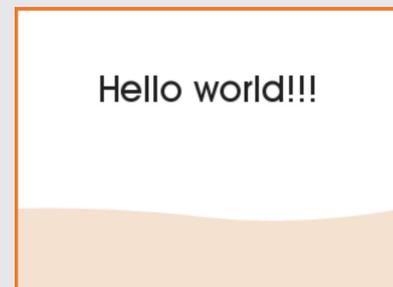


Bild 7: Der Quelltext des rechts als Screenshot gezeigten Einstiegsbeispiels

Eine komplette Beschreibung befindet sich im Quellcode der LCU-Firmware in [7].

```
<ui>
```

Das Root-Element ist immer `<ui>`

- `<page>`

Unterhalb des Root-Elementes folgen beliebig viele `<page>`-Elemente. Jedes `<page>`-Element beschreibt eine Oberflächenseite. Aus Gründen der Übersichtlichkeit wird empfohlen, abgesehen von Ausnahmefällen, nur eine Oberflächenseite pro XML-Datei zu beschreiben.

  - `id="hello"`

Das `<page>`-Element hat ein Attribut `id`, welches die in Verlinkungen verwendete ID der Seite angibt. Diese Seite hat also die ID "hello". Über diese ID und den Kommandozeilenparameter `-s` der UI-Engine kann die initial anzuzeigende Seite spezifiziert werden (siehe oben).
  - `font="fonts/decker.ttf"`

Hiermit wird die Grund-Schriftart für die Oberflächenseite gesetzt. Unabhängig davon kann jedes UI-Element seine eigene Schriftart verwenden. Alle Pfadangaben verstehen sich relativ zu dem Verzeichnis, in dem die XML-Dateien liegen. Dieser Verweis geht also auf die TTF-Datei `/usr/share/ui/fonts/decker.ttf`.
  - `font_height="30"`

Setzt die Default-Schriftgröße für die Oberflächenseite in Pixeln. Unabhängig davon kann jedes UI-Element seine eigene Schriftgröße verwenden.
  - `onload="OnPageLoad"`

Gibt an, dass bei jedem Anzeigen der Seite die TCL-Methode `OnPageLoad` aufgerufen wird.
- `<elements>`

Root-Element für die Auflistung der grafischen Elemente.
- `<image>`

Stellt eine PNG-Grafik dar, in diesem Fall eine displayfüllende Hintergrundgrafik.

  - `id="background"`

Definiert eine ID für die Grafik. Die Angabe ist optional. Über die

ID kann aus dem TCL-Code auf das Grafikelement zugegriffen werden, um es z. B. unsichtbar zu schalten oder zu verschieben.

- `x="0" y="0"`

Gibt die Position der Grafik an. In diesem Fall liegt die obere linke Ecke der Grafik in der oberen linken Ecke des Displays.
- `file="images/background.png"`

Gibt den Dateinamen der Grafikdatei an. Es handelt es sich um einen Pfad relativ zum Verzeichnis mit den XML-Dateien.
- `onclick="exit"`

Dieses Attribut gibt an, was passieren soll, wenn der Anwender innerhalb des Bereiches der Grafik auf den Touchscreen drückt. In diesem Fall wird dann die UI-Engine beendet. Es ließe sich hier z. B. auch ein Link auf eine andere Seite angeben oder eine TCL-Methode aufrufen.
- `<text>`

Ist ein Textelement, für das später per TCL der anzuzeigende Text gesetzt wird. Statischer Text ließe sich auch als Attribut „text“ direkt angeben.

  - `id="text_1"`

Definiert die ID für den Zugriff auf das Textelement.
  - `x="160" y="50"`

Gibt die Position des Textes an. Da auch `center="true"` gesetzt ist (siehe unten), bezieht sich die Position auf die horizontale Mitte der oberen Kante des Textes.
  - `height="30"`

Gibt die Höhe des Elementes in Pixeln an. Für Textelemente wird dadurch gleichzeitig die Schriftgröße angegeben.
  - `center="true"`

Gibt an, dass der Text zentriert angezeigt werden soll. Gleichzeitig wird dadurch der Referenzpunkt für die Positionierung in die Mitte der oberen Kante verschoben.

- `<code>`

Innerhalb des `<code>`-Elementes wird der TCL-Programmtext für das dynamische Verhalten der Seite angegeben. Um den Programmtext lesbar zu halten und keine XML-Zeichen durch Escape-Sequenzen darstellen zu müssen, empfiehlt es sich, diesen Bereich wie im Beispiel explizit als CDATA zu markieren.

- `proc OnPageLoad`

Dies ist die TCL-Methode, die oben als Handler für das Onload-Ereignis der Seite angegeben ist. Der Handler bekommt beim Aufruf zwei Parameter übergeben, die ID der vorherigen Seite und den Grund, warum die Seite angezeigt wird. Als Grund wird entweder „SHOW“ oder „BACK“ übergeben. Durch Prüfen der beiden Parameter kann der TCL-Code insbesondere bei Seiten mit untergeordneten Seiten korrekt reagieren. Der TCL-Code setzt die Eigenschaft „text“ des UI-Elementes `text_1` auf den Wert „Hello world!!!“.

### Kopieren der Beispielapplikation

Die Beispielapplikation ist als Grundlage für eigene Experimente gedacht. Damit das funktioniert, wollen wir zunächst eine Kopie der Beispielapplikation im beschreibbaren Bereich des Flash-Speichers anlegen. Die Originalapplikation befindet sich, wie bereits erwähnt, im nicht beschreibbaren Bereich unter `/usr/share/ui`. Wir legen jetzt eine Kopie im beschreibbaren Bereich unter `/usr/local/ui` an.

Das macht man am einfachsten an der Konsole:

```
# cp -R /usr/share/ui /usr/local/
```

Über „WinSCP“ kann diese Kopieroperation natürlich ebenfalls durchgeführt werden.

Die kopierte Applikation kann mit:

```
# uiengine -d /usr/local/ui
```

getestet werden. Der Parameter `-d` gibt dabei an, in welchem Verzeichnis sich die zu ladende Applikation befindet.

Als Nächstes richten wir das System so ein, dass beim Hochfahren automatisch die kopierte Applikation geladen wird. Dazu wird die Datei `/usr/local/etc/autostart` per WinSCP editiert. Sie sollte den in **Bild 8** gezeigten Inhalt haben.

### TCL-Befehle für die Oberflächensteuerung

Wie bereits erwähnt, integriert die UI-Engine einen Interpreter für die Sprache TCL. Der Befehlssatz von TCL wurde um spezielle Befehle für die UI-Engine erweitert. Eine ausführliche Dokumentation befindet sich in [5]. Hier wird in der Folge nur ein kurzer Überblick gegeben.

#### Befehle zur Manipulation von UI-Elementen

Es wurde ein neuer Befehl „ui“ hinzugefügt. Darüber lassen sich Eigenschaften von UI-Elementen setzen und abfragen. Der Befehl

```
ui set title.text "Seitentitel"
```

setzt die Eigenschaft `text` des Textelementes `title` auf der aktuellen Seite auf den Text „Seitentitel“. Für den Zugriff auf UI-Elemente anderer Seiten kann noch die ID der Seite angegeben werden:

```
ui set overview.title.text "Seitentitel"
```

Analog dazu können Eigenschaften abgefragt werden:

```
ui get title.text
```

www.elvjournal.de

```
AUTOSTART_APP=uiengine
AUTOSTART_ARGS="-d /usr/local/ui"
```

Bild 8: Die Einrichtung der Autostart-Datei für den Start mit der kopierten Beispielapplikation

Dies würde den String „Seitentitel“ zurückgeben. Eigenschaften der UI-Elemente, die sich hierüber setzen und abfragen lassen, umfassen Position und Größe, Sichtbarkeit, Texte und Farbe. Ebenfalls mit dem Befehl „ui“ lässt sich eine andere Oberflächenseite aufrufen. Der Befehl

```
ui load numpad
```

zeigt die Seite mit der ID „numpad“ an. Um eine Seite in der Seitenhistorie zurückzugehen, rufen Sie einfach den folgenden Befehl auf:

```
ui back
```

#### Befehle zur Steuerung der Hinterleuchtung

Die Hinterleuchtung wird mit dem neuen Befehl „backlight“ gesteuert. So setzt

```
backlight range 10 255
```

die minimale Helligkeit auf 10 und die maximale Helligkeit auf 255. Die aktuellen Einstellungen kann man abfragen mit

```
backlight range
```

Dieser Befehl liefert den Minimalwert und den Maximalwert in einer Liste mit zwei Elementen zurück. Die Verzögerungszeit für automatisches Herunterdimmen auf die minimale Helligkeit setzt

```
backlight time 120
```

auf 120 Sekunden. Ein Wert von 0 würde das automatische Herunterdimmen abschalten. Das Abfragen der aktuellen Einstellung erfolgt mit

```
backlight time
```



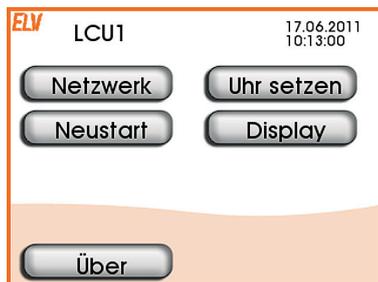
### Weitere Infos:

[7] Quellcode:  
`uiengine/doc/uiengine.pdf`

[8] Screenshots der UI-Engine:  
`<lcu-ip>/screen.cgi`

## Überblick über die Beispielapplikation

Wir wollen nun einen kurzen Überblick über die Seiten der Beispielapplikation geben. Sie ist als Grundlage für eigene Applikationen gedacht. Anhand des vorliegenden Quelltextes und der Dokumentation der UI-Engine können Sie die Implementierung nachvollziehen.



overview.xml Dies ist die Einstiegsseite, welche auf die anderen Seiten verlinkt.



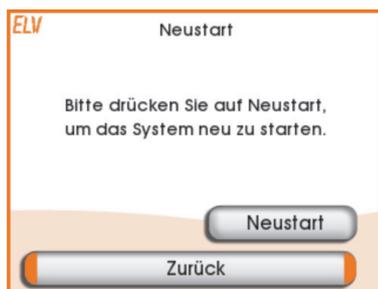
numpad.xml

Diese Seite stellt ein numerisches Eingabefeld mit Eingabepfung dar. Sie wird von vielen anderen Seiten zur Eingabe von Daten verwendet.



network.xml

Diese Seite erlaubt es, die Netzwerkeinstellungen der LCU vorzunehmen sowie die aktuell über DHCP erhaltene IP-Adresse anzuzeigen. Für die Eingabe der IP-Adresse wird *numpad.xml* verwendet. Interessant ist hier die Implementierung einer Checkbox durch das Hinterlegen von mehreren Grafikdateien an einem Grafikelement.



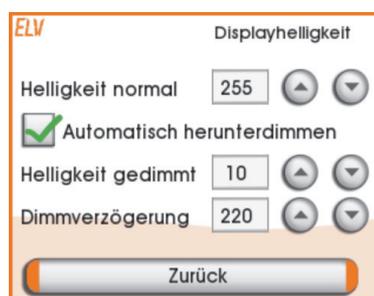
restart.xml

Erlaubt es, das System neu zu starten. Vor dem Neustart wird über *numpad.xml* ein Sicherheitscode abgefragt. Dieser lautet 26789.



setclock.xml

Über diese Seite kann die Systemzeit eingestellt werden.



backlight.xml

Über diese Seite kann die Hinterleuchtung des TFT-Displays kontrolliert werden. Die hier durchgeführten Einstellungen werden beim Verlassen der Seite in der Datei */usr/local/etc/ui.conf* gespeichert. Beim Start der UI-Engine werden die Einstellungen zur Hinterleuchtung aus dieser Datei eingelesen. Das geschieht im TCL-Code dieser Seite in der Methode *Backlight-Init()*. Diese wird in der letzten Zeile des TCL-Codes zu dieser Seite aufgerufen. Dadurch erfolgt dieser Aufruf beim Einlesen der Seitenbeschreibungen durch die UI-Engine.



info.xml

Diese Seite zeigt eine „About-Box“ an. Die angezeigten Informationen werden per TCL aus der Datei */usr/local/etc/info.txt* ausgelesen. Existiert diese Datei nicht, wird der oben zu sehende Standardtext angezeigt. Die Informationen zur Firmwareversion werden aus der im Build erzeugten Datei */etc/VERSION* ausgelesen. Noch ein Tipp: So schöne Screenshots vom TFT-Display der LCU kann man über einen Webbrowser machen. Dazu ist einfach nur die URL [8] aufzurufen.