

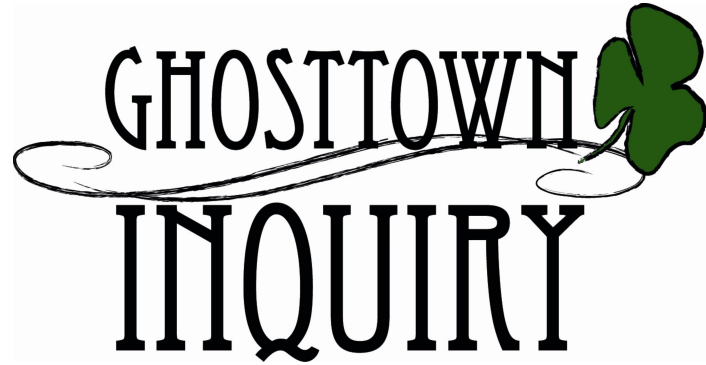


Songtexte einfach anzeigen!

Wieso ein Teleprompter für Musiker?



www.ghosttown-company.de



www.ghosttown-inquiry.de

Notenpulte/Notenständer/iPads

- Sind immer wackelig
- Sind windanfällig
- Verdecken die Sicht auf den Künstler/Bühne für Publikum und Fotografen
- kleines Display, aus Entfernung nicht mehr lesbar
- Abhängig von Akkulaufzeiten
- Zum Umblättern muss man eine Hand frei haben
- Blick des Musikers weggerichtet vom Publikum

Herkömmliche Lösungen: iPad, Notenständer, Notenpult



Anforderungen

An die Hardware:

- Robuste Verpackung für Transport
Tourtauglichkeit
- Einfaches Set-Up
- Gute Ablesbarkeit unter allen Bühnenbedingungen (Licht, Nebel etc.)
- Betrieb auch auf "gut geheizten" Bühnen
- Unabhängigkeit von Akkulaufzeiten, WLAN und Lichtverhältnissen
- Umschalten in Echtzeit

An die Software:

- Einfache Bedienung
Einschalten und Loslegen
- Robustheit
Stecker ziehen statt Herunterfahren
- Zuverlässigkeit
 - Keine Sorge um wackelnde Kontakte (SD-Karte/USB-Stick)
 - Keine Softwareupdates im Betrieb
 - Unabhängigkeit von WLAN, Internet & Co.

Konzept

- Raspberry Pi steuert gewöhnlichen Monitor an
- Einbau in Bühnengehäuse
- Songtexte von USB-Stick
- Bedienung über robusten Fußschalter

Von der Monitorbox zur *Monitorbox*



Monitor

230V
Kaltgerätebuchse

USB-Hub-
Netzteil

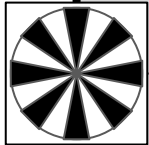
USB-Hub

Externe
USB-Buchse

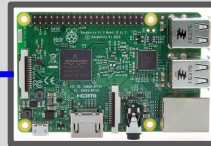
Fußschalter

Externe
USB-Buchse

USB-Stick
Songtexte



Gehäuselüfter





Anforderungen

An die Hardware:

- Robuste Verpackung für Transport
Tourtauglichkeit
- Einfaches Set-Up
- Gute Ablesbarkeit unter allen Bühnenbedingungen (Licht, Nebel etc.)
- Betrieb auch auf "gut geheizten" Bühnen
- Unabhängigkeit von Akkulaufzeiten, WLAN und Lichtverhältnissen
- Umschalten in Echtzeit

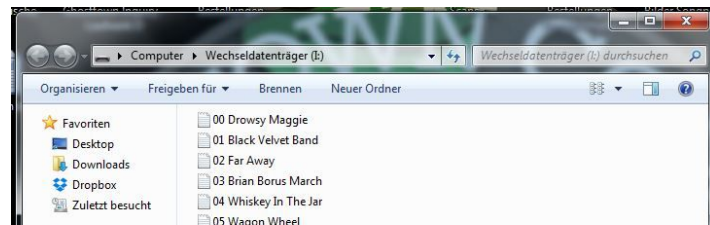
An die Software:

- Einfache Bedienung
Einschalten und Loslegen
- Robustheit
Stecker ziehen statt Herunterfahren
- Zuverlässigkeit
 - Keine Sorge um wackelnde Kontakte (SD-Karte/USB-Stick)
 - Keine Softwareupdates im Betrieb
 - Unabhängigkeit von WLAN, Internet & Co.

Songtexte



- Textdateien auf USB-Stick
- Bildschirmseiten getrennt mit "---"
- Einfache HTML-Formatierungen möglich



CAPO 2 4x Em G C C

Im wandering around see the faces passing by
Of hollow ghosts in ancient times
They're stuck into a mess of black and white
Praying for forgiveness of their crimes

2x Em G C C

In their feverdreams they see nothing like a wall
Of black burned concrete shattered on the ground
No chance to break out and forget about it all
They're running down the roads to the port of call

Bridge 2x D Em C G

You ain't got nothing left to lose and to leave behind
So come onboard and get your oneway ticket signed

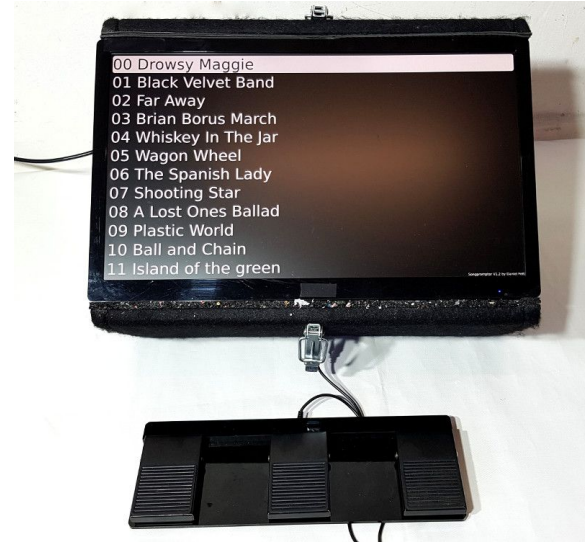
Chorus D D C Em

Were sailing far away on the seas of life 4x

...

User Interface

- Steuerung ausschließlich über Fußschalter
- Zwei Ansichten:
 - Hauptmenü
 - Songtextanzeige (auf Seiten aufgeteilt)
- Vollständig in QML implementiert



QML

- Teil des Qt-Frameworks
- Sprache und Engine für deklarativ beschriebene User-Interfaces
- Sieht aus wie eine Mischung aus CSS und Javascript

QML: Hello World

```
import QtQuick 2.0

Rectangle {
    id: page
    width: 320
    height: 480
    color: "lightgray"

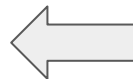
    Text {
        id: helloText
        text: "Hello world!"
        y: 30
        anchors.horizontalCenter: page.horizontalCenter
        font.pointSize: 24
        font.bold: true
    }
}
```



Hello world!

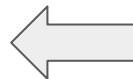
QML: JavaScript

```
Rectangle {  
    color: mousearea.pressed ? "steelblue" : "lightsteelblue"  
}
```



Property binding + JS

```
MouseArea {  
    id: mousearea  
    anchors.fill: parent  
  
    onPressed: {  
        // arbitrary JavaScript expression  
        label.text = "I am Pressed!"  
    }  
    onReleased: {  
        // arbitrary JavaScript expression  
        label.text = "Click Me!"  
    }  
}
```



Event handler + JS

Zuverlässigkeit

Zuverlässiges System durch Read-Only-Filesystem gebaut mit Buildroot



Buildroot

Making Embedded Linux Easy

[LEARN MORE](#) [DOWNLOAD](#)

Buildroot is a simple, efficient and easy-to-use tool to generate embedded Linux systems through cross-compilation.



Can handle everything

Cross-compilation toolchain, root filesystem generation, kernel image compilation and bootloader compilation.



Is very easy

Thanks to its kernel-like menuconfig, gconfig and xconfig configuration interfaces, building a basic system with Buildroot is easy and typically takes 15-30 minutes.

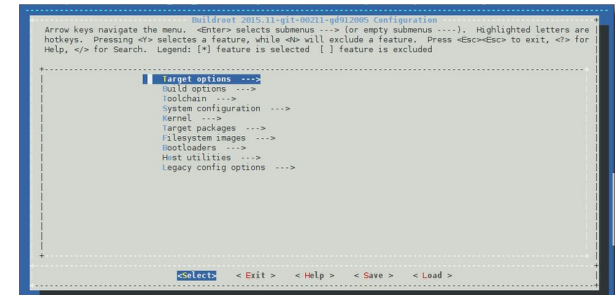


Supports hundreds of packages

X.org stack, Gtk3, Qt 5, GStreamer, Webkit, Kodi, a large number of network-related and system-related utilities are supported.

Buildroot

- Kompiliert eine vollständige Linux-Distribution aus dem Source-Code
 - Kernel
 - Userland, über Pakete individuell zuschneidbar
 - Bootloader
 - Erstellt das Dateisystemimage zum Kopieren auf die SD-Karte
- Cross-Compilation: PC kompiliert Raspberry-Pi-Betriebssystem
- Konfigurierbar über menuconfig (wie der Linux-Kernel)
- Kann auch Read-Only-Filesystems erstellen



Buildroot-Umgebung für den Songprompter

Beispiel Songprompter:

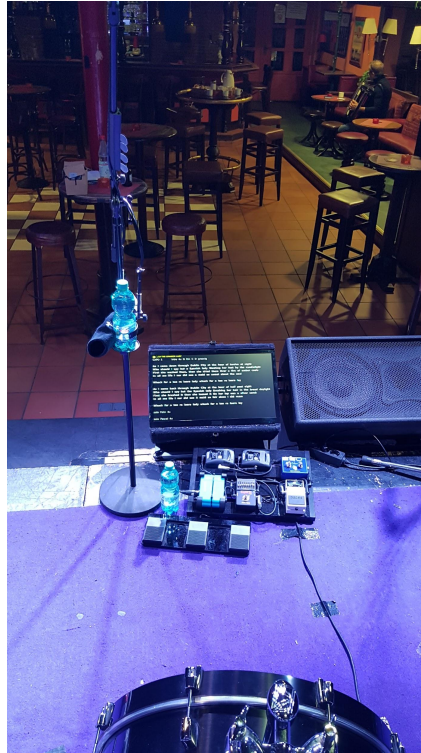
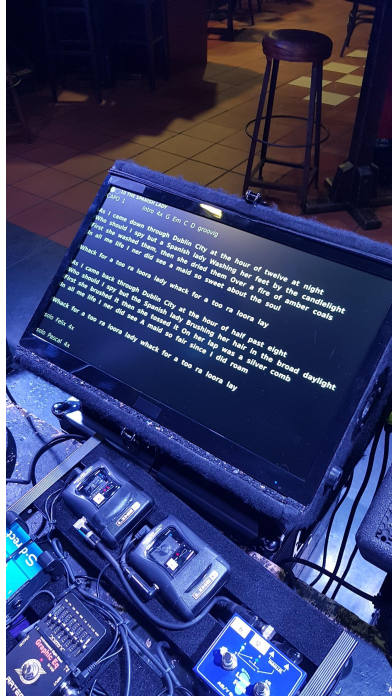
- Python
- Qt mit QML
- Kein X-Server (Anzeige direkt über Framebuffer)

Songprompter-Dateien direkt aus dem GIT-Repository.

Imagegröße: **90 MB**
(ohne Optimierungen)

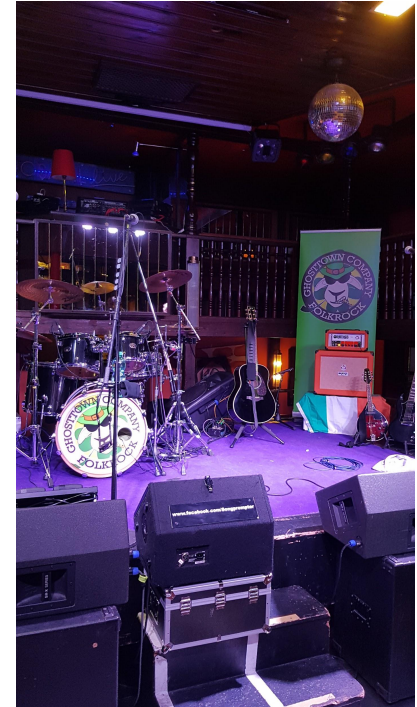
Live im Einsatz

100% lesbar



Blick des Musikers nach vorne gerichtet

Unauffällig für Publikum und Fotografen



Vielen Dank!

Zum Anfassen und Ausprobieren in der
Projektausstellung

