

## Raspberry Pi und easyident



Hier wird beschrieben wie sie unsere **easyident** Transponder Leser mit Raspberry Pi verbinden. Folgende Leser können verwendet werden.



125 KHz 800-0003



LEGIC Prime 800-0002



Legic Advant 800-0006



Mifare Classic 800-0001



Mifare DESFire 800-0004

Inhaltsverzeichnis:

## **Inhalt**

Raspberry Pi und easyident.....	1
Raspberry Pi (Raspbian) Update durchführen.....	3
Easyident Leser am Raspberry Pi anschließen.....	4
Schaltplan Beschreibung:.....	5
Schaltplan:.....	6
Raspberry Pi vorbereiten.....	7
UART Anmeldung:.....	8
easyidentReader.py Software.....	9

## Raspberry Pi (Raspbian) Update durchführen.

Die hier beschriebene Update Prozedur,  
ist für das Betriebssystem Raspbian.

Mit dieser Prozedur können sie ihren Raspberry Pi auf den neuesten Stand bringen.  
Für die Anwendung easyident Transponder Leser ist das Update nicht unbedingt erforderlich!

Ihr Raspberry Pi muss dazu am Netzwerk angeschlossen sein!  
Alle einzugebenden Kommandos, dieser Beschreibung, sind in blau geschrieben.

Starten sie das Terminalprogramm **LXTerminal**.

Eingabe→ **sudo apt-get update** und **<Enter>** Taste  
Raspberry Pi liest nun diverse Paketlisten vom Internet und zeigt sie an.

Eingabe→ **sudo apt-get install** und **<Enter>** Taste  
Raspberry Pi wird aktualisiert.

Eingabe→ **sudo apt-get dist-upgrade** und **<Enter>** Taste  
Raspberry Pi macht ein komplettes Distributionsupdate  
Auf die Frage möchten sie fortfahren antworten sie mit **j** und **<Enter>** Taste  
Ihr System ist nun auf den aktuellen Stand.

Das Programm für easyident Leser wurde in der Programmiersprache Python3 geschrieben. Wenn sie ihre Python3 Programmiersprache updaten wollen, so können sie dies wie folgt durchführen.

Eingabe-→ **sudo apt-get update** und **<Enter>** Taste  
Raspberry Pi liest nun diverse Paketlisten vom Internet und zeigt sie an.

Eingabe→ **sudo apt-get install python3-dev** und **<Enter>** Taste  
Die Programmiersprache Python3 wird aktualisiert.

Ihr System ist nun auf dem aktuellen Stand.

## Easyident Leser am Raspberry Pi anschließen.

Wir, die **FS-Fertigungsservice**, haben für den Raspberry Pi eine kleine Leiterplatte entwickelt, damit sie unsere Transponder Leser am Raspberry Pi anschließen können. Alle unsere Tests wurden mit dem Raspberry Pi **Model B+** durchgeführt. Unsere Leiterplatte kann aber auch für das **Model B** benutzt werden. Die Leiterplatte kann direkt, auf den 40. poligen GPIO-Stecker, aufgesteckt werden.



Für den Anschluss unserer Platine verwenden wir nur die Pins 2, 4, 6, 8, 10 und 12 vom GPIO-Stecker des Raspberry Pi. Alle anderen Pins stehen weiterhin für ihre anderen Anwendungen zur Verfügung.

Die Pins 13 bis 40 sind auf unserer Platine ausgespart, so dass sie frei zugänglich sind. Die Pins 1, 3, 5, 7, 9 und 11 werden von unserer Platine verdeckt. Stehen aber für ihre Anwendung zur Verfügung.

Unser Transponder Leser ist über die RS232 Schnittstelle am GPIO-Port angeschlossen. Zusätzlich haben wir noch ein Umschaltrelais auf unserem Board.

Damit kann man dann eine Türöffner Funktion oder ähnliches programmieren.

### Anschlussbelegung vom GPIO-Port:

Pin-02	= +5V	(Stromversorgung)
Pin-04	= +5V	(Stromversorgung)
Pin-06	= GND	(Masse)
Pin-08	= TXD0	(RS232 Ausgang)
Pin-10	= RXD0	(RS232 Eingang)
Pin-12	= GPIO-18	(Relais)



## Schaltplan Beschreibung:

Der Schaltplan ist so ausgelegt, das die Serielle Schnittstelle auch als V24 mit einem 9 poligen D-SUB Stecker verwendet werden kann. Diese Funktion ist für den easyident Leser jedoch nicht erforderlich. Der easyident Leser ist direkt über die TTL-Leitungen mit dem Raspberry Pi verbunden. Auf dem Board ist ein 3 poliger Steckjumper. Der muss je nach Anwendung zwischen 1+2 oder 2+3 gesteckt werden. Für den easyident Leser muss er auf 1+2 gesteckt werden.

Ich habe auch noch ein kleines Schaltrelais auf der Leiterplatte angebracht. Es ist dafür gedacht, dass man damit das Türöffner Relais einer Eingangstür ansteuern kann. Somit kann man mit dieser Schaltung einen elektronischen Türöffner mit Transponder Technologie aufbauen.

### **Achtung:**

Das verwendete Relais in dieser Schaltung darf maximal mit 30 Volt und 1 Ampere belastet werden.

Der EAGLE Schaltplan mit Layout ist auf meiner Website zum kostenlosen Download bereitgestellt.



## Raspberry Pi vorbereiten.

**Bevor sie die Serielle Schnittstelle vom Raspberry Pi benutzen können müssen sie folgende Prozeduren durchführen.**

Der Raspberry Pi (B) + (B+) bietet auf der 40 poligen Stiftleiste auch eine Serielle Schnittstelle an. Im auslieferungszustand wird diese Schnittstelle jedoch als Kernel-Konsole (System-Schnittstelle) verwendet. Der Raspberry Pi sendet beim Booten Startmeldungen an diese Schnittstelle. Schließt man dort ein Terminal bzw. PC mit Terminal Programm an, sieht man diese Meldungen. Bevor also diese Schnittstelle anderweitig benutzt werden kann, muss diese Funktion ausgeschaltet werden.

### Kernel-Konsole:

Welche Geräte der Kernel nutzt, wird in der Kernel-Kommandozeile festgelegt, die gewöhnlich in der Datei `cmdline.txt` auf der Bootpartition eingerichtet wird. (Diese Partition wird im Linux als `/boot` angezeigt.) Diese Datei, deren gesamter Inhalt in einer einzigen Zeile steht, enthält eine Reihe von Elementen, die den Startvorgang des Kernels bestimmen. Dazu gehört auch die Angabe der Geräte, die als Konsolen verwendet werden. Der folgende Auszug zeigt ein gewöhnliches Beispiel.

```
pi@raspberrypi ~ $ cat /boot/cmdline.txt
dwc_otg.lpm_enable=0 console=ttyAMA0,115200
kgboc=ttyAMA0,115200 console=tty1
root=/dev/mmcblk0p2 rootfstype=ext4
elevator=deadline rootwait
```

In diesem Beispiel sind 2 Konsolen festgelegt.

```
console=ttyAMA0,115200
```

und

```
console=tty1
```

Die erste verwendet den UART (mit dem Gerätenamen `ttyAMA0`) und setzt dessen Geschwindigkeit auf 115200 Baud. Dieser Eintrag muss aus der Datei `cmdline.txt` entfernt werden. Bearbeiten sie diese Datei mit

```
sudo editor /boot/cmdline.txt
```

Entfernen sie den Eintrag `console=ttyAMA0,115200`

Speichern sie die Datei wieder ab.

## UART Anmeldung:

Der Kernel sucht beim booten alle Geräte die auf der Hardware zur Verfügung stehen und führt dann als erstes das Programm `/bin/init` im Wurzeldateisystem aus. Was dieses Programm im Einzelnen tut steht in der Datei `/etc/inittab`.

Die Datei `inittab` muss abgeändert werden.

Starten sie den Texteditor wie folgt.

### `sudo editor /etc/inittab`

Am Ende der Datei finden sie folgende Zeilen.

```
#Spawn a getty on Raspberry Pi serial line  
T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

Damit wird über den seriellen Port `/dev/ttyAMA0` (also den UART) gelesen und geschrieben und seine Geschwindigkeit auf 15200 Baud festgelegt.

Um diese Nutzung des UART zu unterbinden, müssen sie diese Zeile entfernen oder auskommentieren (fügen sie dazu einfach ein `#` am Anfang der Zeile ein, die mit `T0:23:...` beginnt) Beispiel:

```
#Spawn a getty on Raspberry Pi serial line  
#T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

Speicher sie die Datei wieder ab.

Nun steht nichts mehr im Wege den UART für unsere Anwendung zu benutzen.

## easyidentReader.py Software

Für meine Anwendung **easyident-Rasperry** habe ich eine kleine Leiterplatte entwickelt. Eine kurze Programmroutine (easyidentReader.py) die ich mit Python3 geschrieben habe, kann vom Internet herunter geladen werden. Speicher sie diese Datei in ihrem Raspberry Pi im Directory „pi“

Starten sie dann im Raspberry Pi das Programm **LXTerminal**. Das LXTerminal Fenster wird geöffnet.

Geben sie nun **sudo idle3** ein. (Das Programm idle3 muss als Administrator geöffnet werden, daher muss idle3 mit den Zusatz sudo geöffnet werden.)

Das Fenster Python Shell wird geöffnet. Wählen sie dann „File“ und „open“ aus. Öffnen sie den File **easyidentReader.py**

In einem weiteren Fenster wird dann das Programm easyidentReader.py angezeigt. Mit der „F5“ Taste können sie das Programm starten.

Diese Software initialisiert den UART mit 19200 Baud und einen Pin vom GPIO-Port für die Ansteuerung des Relais. Wenn sie einem meiner Leser angeschlossen haben und einen Transponder an den Leser halten so erfolgt eine Ausgabe auf dem Monitor.

**Transponder nicht berechtigt -> (ID-Nummer) Datum und Uhrzeit.**

Damit können sie schon mal die ID-Nummer ihres Transponders sehen.

Im Programm gibt es eine Liste mit dem Namen „**Transponder**“.

Ersetzen Sie dort die eingegebenen Transponder durch ihre eigenen Transponder ID-Nummern. (Bitte nur Großbuchstagen verwenden)

```
Transponder = [„0F006F2E27“,           # 1. Transponder
                „0F0179688B“,         # 2. Transponder
                „01098B179C“,         # 3. Transponder u.s.w.
                ]                     # Abschluß
```

Wenn sie jetzt das Programm starten, zieht das Türöffner Relais für die programmierte Zeit (1,5 Sekunden) an. Ein weiterer Ausbau der Software easyidentReader.py ist von unserer Seite nicht geplant. Diese Aufgabe überlassen wir ihrer Fantasie.

Viel Spaß beim Experimentieren.

Josef Rohrmüller

**FS-Fertigungsservice**

[www.easyident.de](http://www.easyident.de)

[Josef.Rohrmueller@FSJR.de](mailto:Josef.Rohrmueller@FSJR.de)