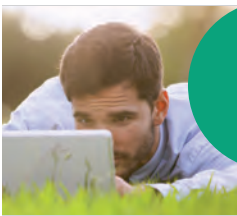




1

| | | |
|----------|---|-----------|
| 1 | Betriebssystem für mobilen Einsatz vorbereiten | 11 |
| 1.1 | USB-Geräte verwalten | 12 |
| 1.1.1 | USB-Geräte unterscheiden | 12 |
| 1.1.2 | Auf Ein-/Ausstecken reagieren | 15 |
| 1.2.1 | Automatische Dateisystemreparatur aktivieren | 17 |
| 1.2 | Daten sichern und retten..... | 17 |
| 1.2.2 | Störfaktor dmesg-Fehler: Dateisystemprüfung erzwingen..... | 18 |
| 1.2.3 | Probleme beim Start? – Bootverzeichnis reparieren | 19 |
| 1.2.4 | SD-Speicherkarte: nachträgliche Dateisystemreparatur..... | 20 |
| 1.2.5 | Risiko SD-Karte: automatische Sicherung per cron-Job | 23 |
| 1.3.1 | Automatisches Log-in: pi vom Start weg | 25 |
| 1.3 | Automatisiertes Hochfahren..... | 25 |
| 1.3.2 | Autostart nach dem Einschalten | 26 |
| 1.4.1 | Unnötige Pakete entfernen..... | 27 |
| 1.4.2 | Vorhandene Raspberry Pi-Pakete aktualisieren | 27 |
| 1.4 | Softwarepakete verwalten..... | 27 |
| 1.5 | GPIO-Zugriff mit Python | 28 |
| 1.5.1 | Bibliothek RPi.GPIO installieren..... | 28 |
| 1.5.2 | Praktischer und schneller GPIO-Zugriff mit der Wiring Pi-API..... | 30 |
| 1.6 | Drahtlose Kommunikation via USB und Bluetooth..... | 32 |
| 1.6.1 | USB-WLAN-Dongle im Einsatz..... | 32 |
| 1.6.2 | Statische IP-Adresse einrichten..... | 33 |
| 1.6.3 | Tastatur und Eingabegeräte für unterwegs..... | 34 |
| 1.7 | Kernel selbst kompilieren | 36 |



2

| | | |
|----------|---|-----------|
| 2 | Oberfläche per Touch bedienen..... | 43 |
| 2.1 | Display für unterwegs..... | 44 |
| 2.1.1 | Touch-LC-Display installieren | 44 |
| 2.1.2 | Der Touchbildschirm im Einsatz..... | 52 |
| 2.2.1 | Installation und Inbetriebnahme | 55 |
| 2.2 | Benutzeroberfläche im Eigenbau..... | 55 |
| 2.2.2 | Erste Schritte mit PyGame | 56 |
| 2.2.3 | Ausgabe eines Bildschirmhintergrunds mit Text | 57 |
| 2.2.4 | Interaktion mit dem Benutzer..... | 58 |
| 2.2.5 | Ausgabe der x,y-Koordinaten beim Klicken..... | 59 |
| 2.2.6 | Grafikausgabe und Interaktion | 60 |
| 2.2.7 | Grafische Benutzeroberfläche für ein Telefon | 61 |

| | | |
|----------|--|-----------|
| 3 | Orientierung mit GPS..... | 65 |
| 3.1 | GPS-Modul | 66 |
| 3.1.1 | GPS-Koordinaten und Entfernungen | 74 |

3.2 Navigationssystem im Selbstbau 82

3.2.1 Die ganze Welt auf dem Raspberry Pi 83

3.2.2 Touch-LC-Display im Einsatz: Navit-Parameter anpassen 91

4 Alternative Energiequellen 95

4.1 Stromalternativen im Auto 96

4.1.1 Auto, Wohnmobil & Co.: 12-V-Spannungsversorgung 96

4.1.2 USB-Anschlusskabel im Zigarettenanzünder 97

4.1.3 Netzteilalternative: Quetschstecker und Spannungswandler 98

4.2 Praktisch und sicher: USV für den Raspberry Pi 99

4.2.1 Pi USV in Betrieb nehmen 99

4.2.2 Ohne Strom nix los: Akkupack auswählen 100

4.2.3 Pi USV-Software in Betrieb nehmen 101

4.2.4 Status der Pi USV erkennen 102

4.2.5 Status der Pi USV mit Python auslesen 103

5 Raspberry Pi im Auto 107

5.1 Autodiagnose mit der OBD-II-Buchse 108

5.1.1 Anschluss gesucht: OBD-II-Adapter und Raspberry Pi 109

5.1.2 O'zapft is: Kraftfahrzeug mit Raspberry Pi auslesen 111

5.1.3 Kraftfahrzeuggedächtnis auslesen 117

5.1.4 Python-Zugriff auf die OBD-II-Daten 121

5.2 Dashcam-Einsatz: Kameramodul oder USB-Kamera? 125

5.3 HeadUp-Display mit dem Raspberry Pi 141

6 Mobilfunkprogrammierung 147

6.1.1 Neue Firmware: gefahrlos und schnell 149

6.1 Raspberry Pi als SMS-Alarmanlage 149

6.1.2 SIM900 GPRS/GSM-Modul in Betrieb nehmen 153

6.1.3 SIM900-Modem mit Python steuern 155

6.1.4 Mobilfunkmodem mit minicom-Konsole steuern 158

6.1.5 SMS-Nachrichten über minicom-Konsole 161

6.1.6 AT-Kommandos über Python 169

6.1.7 SMS-Versand mit Python 170

6.1.8 Auf den Spuren der NSA: Handyüberwachung per SMS 174

6.2 Standortinformationen ohne GPS: GSM-Hacking mit dem Raspberry Pi 175

6.2.1 Standortinformationen im Mobilfunknetz 176

6.2.2 Mobilfunkverbindung prüfen 177





| | | |
|-------|---|-----|
| 6.2.3 | LAC, CID & Co.: Standortdaten auslesen..... | 179 |
| 6.2.4 | Einfache Geolocation ohne GPS mit GSM-Standort..... | 181 |
| 6.2.5 | Standortdatenverarbeitung mit Python | 182 |
| 6.3 | GPS und GSM kombinieren: Alarm mit dem Raspberry Pi | 186 |
| 6.3.1 | Benachrichtigung per SMS | 186 |
| 6.3.2 | Aktuellen Standort abfragen per SMS..... | 189 |
| 6.4 | Einbau von Modemfunktionen: Autotelefon mit dem Pi..... | 195 |

| | | |
|-------|---|-----|
| 7 | Motorisierte Partypuppe | 199 |
| 7.1 | Inbetriebnahme der Bewegungserkennung per Infrarot..... | 200 |
| 7.1.1 | Shell-Skript für PIR-Bewegungsmelder | 202 |
| 7.2 | Atmen, Winken, Schlagen – Bewegungen mit Motor umsetzen | 203 |
| 7.3 | Hardware-PWM-Ausgang mit LED testen..... | 206 |
| 7.4 | Servoblaster-Treiber installieren | 208 |
| 7.5 | Servomotoren mit Servoblaster in Betrieb nehmen | 210 |
| 7.5.1 | Servomotor steuern mit Python..... | 211 |
| 7.5.2 | Achsen und Bewegung steuern | 212 |
| 7.6 | PIR-Bewegungsmelder im Einsatz..... | 216 |
| 7.7 | Alarmsignal und Geräusche abspielen | 217 |

Anhang

GPIO-Pins aller Modelle im Überblick:

| | |
|--|-----|
| A, B, A+, B+ und Raspberry Pi 2 Modell B | 220 |
|--|-----|

| | |
|------------|-----|
| Index..... | 222 |
|------------|-----|