

Raspberry Pi programmieren mit Python

DAS INHALTS- VERZEICHNIS

» Hier geht's
direkt
zum Buch

Inhaltsverzeichnis

	Einleitung	13
	Lösungen und Programmcode zum Download	16
1	Begegnung mit Python	17
1.1	Was ist Python?	17
1.2	IDLE	18
1.2.1	IDLE-Shell	20
1.2.2	Hotkeys	21
1.3	Die IDLE-Shell als Taschenrechner	22
1.3.1	Operatoren und Terme	22
1.3.2	Zahlen	24
1.3.3	Mathematische Funktionen	28
1.4	Hilfe	32
1.5	Namen und Zuweisungen	33
1.5.1	Zuweisungen für mehrere Variablen	35
1.5.2	Rechnen mit Variablen in der Shell	36
1.5.3	Syntaxregeln für Bezeichner	36
1.5.4	Neue Namen für Funktionen und andere Objekte	37
1.5.5	Erweiterte Zuweisungen	38
1.6	Mit Python-Befehlen Geräte steuern	39
1.6.1	Projekt: Eine LED ein- und ausschalten	39
1.6.2	Das Modul gpiozero	41
1.6.3	Steuern mit Relais	42
1.6.4	Projekt: Eine Taschenlampe an- und ausschalten	44
1.6.5	Projekt: Steuerung eines LED-Strips mit MOSFETs	46
1.7	Aufgaben	49
2	Python-Skripte	53
2.1	Ein Skript mit IDLE erstellen	53
2.1.1	Ein neues Projekt starten	53
2.1.2	Programmtext eingeben	54
2.1.3	Das Skript ausführen	55
2.1.4	Shortcuts	55
2.2	Programme ausführen	55
2.2.1	Programm in der Konsole starten	55
2.2.2	Anklicken des Programmicons im File-Manager	56
2.3	Interaktive Programme – das EVA-Prinzip	59

2.3.1	Format mit Bedeutung – Aufbau eines Python- Programmtextes	60
2.3.2	Eingabe – die input()-Funktion	61
2.3.3	Verarbeitung – Umwandeln von Datentypen und Rechnen.	61
2.3.4	Ausgabe – die print()-Funktion	62
2.4	Programmverzweigungen	64
2.4.1	Einfache Bedingungen	64
2.4.2	Wie erkennt man eine gute Melone? Zusammengesetzte Bedingungen.	66
2.4.3	Einseitige Verzweigungen und Programmblöcke	67
2.4.4	Haben Sie Ihr Idealgewicht?	68
2.4.5	Eine Besonderheit von Python: Wahrheitswerte für Objekte	72
2.5	Bedingte Wiederholung – die while-Anweisung.	72
2.5.1	Projekt: Zahlenraten.	73
2.5.2	Have a break! Abbruch einer Schleife.	74
2.6	Projekte mit dem GPIO.	75
2.6.1	Blinklicht	75
2.6.2	Schalter	76
2.6.3	Zähler	78
2.6.4	Die Methode wait_for_press()	79
2.7	Projekt: Eine Alarmanlage.	79
2.7.1	Aufbau und Arbeitsweise der Alarmanlage	79
2.7.2	Programmierung	81
2.8	Aufgaben	82
3	Kollektionen: Mengen, Listen, Tupel und Dictionaries	85
3.1	Die Typhierarchie.	85
3.1.1	Reihenfolge der Elemente und Zugriff.	86
3.1.2	Änderbarkeit	86
3.2	Gemeinsame Operationen für Kollektionen	87
3.3	Kollektionen in Bedingungen.	88
3.3.1	Projekt: Kundenberatung.	89
3.3.2	Projekt: Sichere Kommunikation	89
3.4	Iteration – die for-Anweisung.	90
3.4.1	Verwendung von break	91
3.5	Sequenzen	92
3.5.1	Konkatenation und Vervielfältigung	92
3.5.2	Direkter Zugriff auf Elemente – Indizierung.	93
3.5.3	Slicing	93
3.5.4	Projekt: Lesbare Zufallspasswörter	94
3.6	Tupel.	96

3.7	Zeichenketten (Strings).....	97
3.7.1	Strings durch Bytestrings codieren.....	98
3.7.2	Formatieren	99
3.8	Listen	101
3.8.1	Listen sind Objekte und empfangen Botschaften	101
3.8.2	Klasse, Typ und Instanz.....	103
3.8.3	Kopie oder Alias?	103
3.8.4	Listenoperationen.....	104
3.8.5	Projekt: Zufallsnamen	106
3.8.6	Projekt: Telefonliste	107
3.8.7	Listen durch Comprehensions erzeugen	108
3.9	Zahlen in einer Folge – range()-Funktion	109
3.10	Projekt: Klopffzeichen	111
3.11	Mengen.....	114
3.12	Projekt: Zufallssounds	115
3.12.1	Wie kommen Töne aus dem Raspberry Pi?.....	115
3.12.2	Sounds mit PyGame	116
3.12.3	Programmierung	117
3.13	Dictionaries	119
3.13.1	Operationen für Dictionaries.....	120
3.13.2	Projekt: Morsen	121
3.14	Aufgaben	122
4	Funktionen.....	125
4.1	Aufruf von Funktionen	125
4.1.1	Unterschiedliche Anzahl von Argumenten	126
4.1.2	Positionsargumente und Schlüsselwort-Argumente.....	126
4.2	Definition von Funktionen	128
4.3	Funktionen in der IDLE-Shell testen	129
4.4	Docstrings	130
4.5	Veränderliche und unveränderliche Objekte als Parameter.....	131
4.6	Voreingestellte Parameterwerte	133
4.7	Beliebige Anzahl von Parametern	134
4.8	Die return-Anweisung unter der Lupe.....	135
4.9	Mehr Sicherheit! Vorbedingungen testen	136
4.10	Namensräume: Global und lokal	138
4.11	Rekursive Funktionen – die Hohe Schule der Algorithmik.....	141
4.11.1	Projekt: Rekursive Summe	141
4.11.2	Projekt: Quicksort.....	142
4.12	Funktionen per Knopfdruck aufrufen: Callback-Funktionen	143
4.12.1	Projekt: Klangmaschine.....	144
4.12.2	Projekt: Zeit schätzen.....	145
4.13	Aufgaben	147

5	 Fenster für den RPi – Grafische Benutzungsoberflächen	151
5.1	Wie macht man eine Benutzungsoberfläche?	151
5.2	Projekt: Die digitale Lostrommel	152
	5.2.1 Die Gestaltung der Widgets	154
	5.2.2 Das Layout-Management	156
5.3	Projekt: Farbmischer	158
5.4	Wer die Wahl hat, hat die Qual: Checkbutton und Radiobutton	160
5.5	Projekt: Automatische Urlaubsgrüße	160
5.6	Projekt: Digitaler Glückskeks	162
5.7	Viele Widgets schnell platziert: Das Grid-Layout	164
5.8	Projekt: 100 Farben	165
	5.8.1 Die professionelle Version	167
5.9	Aufgaben	170
6	 Daten finden, laden und speichern	173
6.1	Dateien	173
	6.1.1 Daten speichern	173
	6.1.2 Daten laden	174
6.2	Ein Blick hinter die Kulissen: Die SD-Karte	174
	6.2.1 Experiment 1: Wie viel Zeit wird zum Schreiben von 10 MB auf die SD-Karte benötigt?	175
	6.2.2 Experiment 2: Wie viel Zeit wird zum Lesen von 100 MB von der SD-Karte benötigt?	176
6.3	Datenstrukturen haltbar machen mit pickle	176
6.4	Versuch und Irrtum – Mehr Zuverlässigkeit durch try-Anweisungen	177
6.5	Projekt: Karteikasten	178
	6.5.1 Der Editor	179
	6.5.2 Der Presenter	182
6.6	Benutzungsoberfläche zum Laden und Speichern	185
	6.6.1 Dialogboxen	185
	6.6.2 Erweiterung des Editors für Karteikarten	187
	6.6.3 Erweiterung des Presenters	190
6.7	Daten aus dem Internet	192
6.8	Projekt: Goethe oder Schiller?	194
	6.8.1 Methoden der String-Objekte	194
	6.8.2 Programmierung	196
6.9	Daten finden mit regulären Ausdrücken	199
	6.9.1 Reguläre Ausdrücke	199
	6.9.2 Die Funktion findall()	201
	6.9.3 Projekt: Staumelder	202
	6.9.4 Programmierung	202
6.10	Aufgaben	206

7	Projekte mit Zeitfunktionen	209
7.1	Projekt: Fünf Sekunden stoppen und gewinnen.	209
7.2	Datum und Zeit im Überblick	211
7.3	Projekt: Digitaluhr.	212
	7.3.1 Woher bekommt der RPi die Zeit?	213
	7.3.2 Was ist ein Prozess?	213
	7.3.3 Vollbildmodus	215
	7.3.4 Event-Verarbeitung.	218
	7.3.5 Autostart	219
7.4	Projekt: Ein digitaler Bilderrahmen	219
	7.4.1 Zugriff auf das Dateisystem: Das Modul os	220
	7.4.2 Python Imaging Library (PIL)	221
	7.4.3 Die Programmierung	223
7.5	Projekt: Wahrnehmungstest.	225
	7.5.1 Die Programmierung	226
7.6	Aufgaben	230
8	Objektorientierte Programmierung	233
8.1	Überall Objekte	233
8.2	Klassen und Vererbung bei Python	235
	8.2.1 Einführendes Beispiel: Alphabet.	236
	8.2.2 Qualitätsmerkmal Änderbarkeit	239
	8.2.3 Vererbung	241
8.3	Projekt: Pong revisited	242
	8.3.1 Die Klasse Canvas.	244
	8.3.2 Aufbau des Projekts	249
	8.3.3 Die Tick-Metapher	250
	8.3.4 Die Programmierung	251
	8.3.5 Erweiterungen	256
8.4	Aufgaben	256
9	Sensortechnik	259
9.1	Was ist ein digitaler Temperatursensor?	259
9.2	Den DS1820 anschließen	260
9.3	Temperaturdaten lesen	261
	9.3.1 Temperaturdaten eines Sensors automatisch auswerten	263
9.4	Projekt: Ein digitales Thermometer mit mehreren Sensoren	264
	9.4.1 Ein Modul für die Messwerterfassung	265
	9.4.2 Die grafische Oberfläche	267
	9.4.3 Temperaturdaten per E-Mail senden	268
9.5	Projekt: Ein Temperaturplotter.	270
	9.5.1 Temperatur-Zeitdiagramme	270
	9.5.2 Programmierung	271

9.6	Spannung messen	275
9.6.1	Das SPI-Protokoll	277
9.6.2	Programmierung	278
9.7	Alkoholsensor.	280
9.7.1	Projekt: Achtung! Alkoholisches Getränk!	281
9.7.2	Projekt: Den Alkoholgehalt der Luft und von Flüssigkeiten messen	284
9.7.3	Wie kann man den Alkoholgehalt von Flüssigkeiten messen?	285
9.7.4	Messen und interpolieren	288
9.8	Projekte mit einem digitalen Lichtsensor	291
9.8.1	Ein Paket in einer virtuellen Umgebung installieren	292
9.8.2	Die Programmierung	293
9.8.3	Das Modul smbus	294
9.8.4	Mit dem Raspberry Pi Farben messen – Absorptionsspektrometer	298
9.9	Kohlendioxid-Sensor	304
9.9.1	Projekt: Datenlogger	305
9.9.2	Das Sensormodul kalibrieren	306
9.9.3	Projekt: Ein Experiment zur Diffusion von Gasen.	307
9.10	Mit Ultraschall Entfernungen messen	310
9.10.1	Die Schaltung	311
9.10.2	Projekt: Abstandsmessungen	312
9.10.3	Projekt: Fische – eine interaktive Animation mit Ultraschall-Sensor.	313
9.11	Aufgaben	318
9.12	Lösung des Rätsels.	320
10	Projekte mit der Kamera	321
10.1	Das Kameramodul anschließen	321
10.2	Die Kamerasoftware.	323
10.3	Projekt: Überwachungskamera – Livebild auf dem Bildschirm.	324
10.4	Projekt: Bewegung erfassen	326
10.5	Projekt: Gerichtete Bewegungen erfassen	330
10.5.1	Files verarbeiten mit subprocess und io	331
10.5.2	Die Programmierung	331
10.6	Projekt: Birnen oder Tomaten?	337
10.6.1	Magische Methoden – das Überladen von Operatoren	338
10.6.2	Programmierung	340
10.6.3	Weiterentwicklungen	343
10.7	Projekt: Fotos per E-Mail verschicken	344
10.8	Randbemerkung: Was darf man? Was soll man?	345
10.9	Aufgabe: Wie lang? Wie breit?	346

11	Webserver	349
11.1	Der RPi im lokalen Netz	349
	11.1.1 WLAN	349
	11.1.2 Virtual Network Computing (VNC)	350
11.2	Webserver	352
	11.2.1 Der Apache-Webserver	352
	11.2.2 Die eigene Startseite	353
11.3	Was ist los im Gartenteich?	355
	11.3.1 Projekt: Einfache Webcam mit statischer Webseite	355
11.4	Lösungen zu den Zwischenfragen	358
12	Erweiterungen: OLED und HAT	359
12.1	OLED-Display	359
12.2	Anschluss	359
12.3	Installation der SSD1306-Bibliothek	360
12.4	Auf dem Display Texte und Formen ausgeben	360
	12.4.1 Projekt: Uhrzeit	361
12.5	Mit ImageDraw zeichnen und Texte schreiben	362
	12.5.1 Grafiken zeichnen	363
	12.5.2 Schriftarten definieren	365
12.6	Projekt: Thermometer	366
12.7	Sense HAT	368
12.8	Die Klasse SenseHat	369
12.9	Grafische Ausgabe über die LED-Matrix	371
12.10	Die räumliche Orientierung des Sense HAT	372
	12.10.1 Projekt: Das Murmellabyrinth	373
12.11	Der Joystick	376
	12.11.1 InputEvent	377
	12.11.2 Die Klasse JoyStick	378
	12.11.3 Definition von Eventhandlern	378
	12.11.4 Auf Events warten und Events abfragen	380
12.12	Aufgaben	382
A	Den Raspberry Pi einrichten	383
B	Der GPIO	389
C	Autostart	393
D	Bau eines Fußschalters	395
E	Virtuelle Umgebungen	397
	Stichwortverzeichnis	401