

Auf einen Blick

1	Der Mikrocontroller ESP32 und seine Funktionalitäten	19
2	Die Stromversorgung	39
3	Ihre Werkstatt	51
4	Entwicklungsumgebungen	81
5	Programmierung mit C und C++	145
6	Peripherie und Funktionen des ESP32	253
7	Externe Schnittstellen	325
8	Externe Kommunikations- und Steuerungsmöglichkeiten	399
9	Projektideen	513

Inhalt

Einleitung	13
1 Der Mikrocontroller ESP32 und seine Funktionalitäten	19
1.1 Die ESP32-SoCs	19
1.2 Die Module	22
1.2.1 Die WROOM-, SOLO-, WROVER- und PICO-Module	22
1.3 Die Boards	25
1.4 Das ESP32-Dev-KitC V4	26
1.4.1 Das Pinout	28
1.4.2 Funktionen der GPIO-Pins	29
1.4.3 Höhere Eingangsspannung an GPIOs	35
1.5 ESP32-Dev-KitC V4 – Erstinbetriebnahme am PC	37
1.5.1 Windows	37
1.5.2 Linux	38
2 Die Stromversorgung	39
2.1 Betriebszustände und typische Stromverbräuche	39
2.2 Anschlüsse zur Stromversorgung	41
2.3 Grundschema der Stromversorgung	42
2.4 Spannungsquellen	43
2.5 Spannungsregler	46
2.5.1 Lineare Spannungsregler	46
2.5.2 Schaltregler	48

3	Ihre Werkstatt	51
3.1	Breadboard & Co.	51
3.1.1	Das Breadboard	51
3.1.2	Jumperkabel	52
3.1.3	Leiterplatine	53
3.2	Hilfreiche Werkzeuge	55
3.3	Fritzing	58
3.3.1	Was ist Fritzing?	58
3.3.2	Fritzing starten	58
3.3.3	In der Steckplattenansicht arbeiten	60
3.3.4	In der Schaltplanansicht arbeiten	66
3.3.5	In der Leiterplattenansicht arbeiten	68
3.3.6	Bauteile importieren	70
3.3.7	Bauteile ändern und selbst erstellen	71
3.4	Löttechnik	71
3.4.1	Bauteile auf Platinen löten	72
3.4.2	Bauteile entlöten	73
3.4.3	SMD löten	74
3.4.4	ESP32-Modul löten	75
4	Entwicklungsumgebungen	81
4.1	Der Weg von lesbaren Anweisungen zum Maschinencode	81
4.2	Das Betriebssystem der Entwicklungsumgebung	84
4.3	esptool.py	85
4.3.1	Installation unter Windows	86
4.3.2	Installation unter Linux	90
4.3.3	Ausführbare Versionen von esptool	90
4.3.4	esptool einsetzen	90
4.4	Entwicklungsumgebungen in der Übersicht	93
4.4.1	Das Espressif IDF	94
4.4.2	Die Arduino-IDE auf dem PC	108
4.4.3	Der Arduino-Web-Editor	126
4.4.4	Eclipse	126
4.4.5	MicroPython	127
4.4.6	Lua	134

4.5	Partitionstabelle festlegen	141
4.6	Resümee	143
5	Programmierung mit C und C++	145
5.1	Programmdesign für den Mikrocontroller	145
5.1.1	»Normale« C-/C++-Programme	147
5.1.2	Sketche	148
5.1.3	Erweiterte Programmstruktur	148
5.2	Syntax und Semantik in der Programmiersprache	149
5.3	Zeichen, Kommentare und Begrifflichkeiten	150
5.3.1	Zeichen und ihre Bedeutung	150
5.3.2	Kommentare	152
5.3.3	Begrifflichkeiten	152
5.4	Daten und Datentypen	153
5.5	Variablen	158
5.5.1	Deklaration und Definition	158
5.5.2	Initialisierung und Wertezuweisung	159
5.5.3	Gültigkeitsbereiche von Variablen	159
5.5.4	Typqualifizierer	160
5.5.5	Der Variablenmodifikator PROGMEM	161
5.6	Ausdrücke und Operatoren	162
5.6.1	Arithmetische Operatoren	162
5.6.2	Zuweisungsoperatoren	162
5.6.3	Bedingte Operatoren (Vergleiche)	163
5.6.4	Aussagenlogik	165
5.6.5	Bitoperatoren	166
5.6.6	Typenumwandlung	169
5.6.7	Der Operator sizeof	170
5.6.8	Fehlerquellen	171
5.7	Zeiger (Pointer)	171
5.8	Arrays und Strukturen	175
5.8.1	Arrays (Tabellen)	175
5.8.2	struct (Strukturen)	177
5.8.3	unions	178
5.8.4	enum	178
5.8.5	typedef	179

5.9 Kontrollstrukturen	179
5.9.1 if-Bedingung	179
5.9.2 if-else	181
5.9.3 switch ... case	182
5.9.4 while-Schleife	184
5.9.5 do-while-Schleife	185
5.9.6 for-Schleife	186
5.10 Funktionen	188
5.10.1 Die Syntax einer Funktion	189
5.10.2 Funktionsdeklaration und Funktionsdefinition	190
5.10.3 Funktionsaufruf	192
5.10.4 Übergabe von Parametern	194
5.10.5 Rücksprung	199
5.10.6 Rückgabe von Ergebnissen	200
5.11 Funktionen nutzen: Ihr Werkzeugkasten	202
5.11.1 Zeitfunktionen	202
5.11.2 Mathematische und trigonometrische Funktionen	203
5.11.3 Zufallszahlen	206
5.12 Klassen und objektorientierte Programmierung	208
5.13 Die String()-Klasse	216
5.14 Serielle Aus- und Eingabe	225
5.15 Exkurs: Tasks und Threads	231
5.15.1 Die FreeRTOS-Task-API	232
5.15.2 Threads programmieren	233
5.16 Präprozessordirektiven	234
5.16.1 Die Direktive #define	234
5.16.2 Die Direktive #include	235
5.16.3 #ifdef-/#endif-Direktive	236
5.17 Prinzipien des Softwareentwurfs	237
5.18 Fehlermanagement	237
5.18.1 Einfache Methoden	238
5.18.2 Aufwändigere Methoden	240
5.18.3 Professionelle Methoden	248
5.18.4 Datenblätter	248
5.19 JSON – das Datenformat für den Datenaustausch	248

6 Peripherie und Funktionen des ESP32	253
6.1 Der interne Hall-Sensor	253
6.2 Digitale Aus-/Eingänge	254
6.2.1 Digitale Ausgänge	255
6.2.2 Digitale Eingänge	257
6.3 Analoge Aus-/Eingänge	263
6.3.1 Analoge Ausgänge	263
6.3.2 Analoge Eingänge	264
6.4 Das ESP32-Touch-Sensor-System	275
6.5 DHT11/DHT22	281
6.6 433-MHz-Steuerung	285
6.7 LED-Control und Pulsweitenmodulation	291
6.7.1 Pulsweitenmodulation	291
6.7.2 Die ESP32-LED-Steuereinheit	292
6.8 Impulszähler	297
6.9 Universaltimer, Watchdogtimer	301
6.9.1 Universaltimer	301
6.9.2 Watchdogtimer	303
6.10 Energie-(Spar-)Modi	305
6.11 Daten auf dem ESP32 speichern	314
6.12 FreeRTOS-Task-Funktionen	317
6.13 Thread-Funktionalität nutzen	321
7 Externe Schnittstellen	325
7.1 SPI	325
7.1.1 RFID mit RFID-RC522	326
7.1.2 SD-Karte	333
7.1.3 TFT-Display	339
7.2 I²C	344
7.3 Bluetooth	348
7.3.1 Bluetooth Classic	349
7.3.2 Bluetooth Low Energy	351

7.4	WLAN/WiFi	366
7.4.1	ESP32 als AP	367
7.4.2	ESP32 als »Station«	372
7.4.3	ESP32 mit ESP-Now	380
7.4.4	ESP32-Mesh	388
7.5	UART	389
7.5.1	Die UART-Schnittstellen des ESP32	390
7.5.2	Ein UART-Echo	391
7.5.3	ESP32-Modul flashen	393
7.6	OTA	394
8	Externe Kommunikations- und Steuerungsmöglichkeiten	399
8.1	Das JSON-Datenaustauschformat in der Praxis	400
8.2	Apps	405
8.2.1	Blynk	405
8.2.2	MIT App Inventor	417
8.3	Messenger	430
8.3.1	E-Mail-Versand	430
8.3.2	Pushbullet	438
8.3.3	Alexa	446
8.3.4	Telegram, Twitter und WhatsApp	450
8.4	Das MQTT-Protokoll	450
8.5	Cloud-IoT-Plattformen	458
8.5.1	Cayenne	458
8.5.2	Adafruit IO	465
8.5.3	IFTTT	473
8.6	Lokale Anwendungen nutzen	484
8.6.1	Node-RED	485
8.6.2	ioBroker	499

9	Projektideen	513
9.1	The Evil Dice	513
9.2	Die Maker-Uhr	519
9.3	Das Dateisystem einmal anders	539
9.4	Solar-Repeater	549
9.5	Als die Bilder laufen lernten	553
9.5.1	Schwarz-Weiß-Bild anzeigen	553
9.5.2	RGB-Bild anzeigen	557
9.5.3	TFT-Slide-Show	559
9.6	IoT-Barometer	566
9.6.1	Der Sensor BMP280	567
9.6.2	Das I ² C-OLED-Display SSD1306	571
9.6.3	Beide Module an einem gemeinsamen I ² C-Bus	578
9.6.4	Beide Module an getrennten I ² C-Kanälen	581
9.7	Eine kleine Schaltzentrale	582
	Anhang	603
	Index	607