

Auf einen Blick

1	Der Mikrocontroller ESP32	19
2	Die Stromversorgung	43
3	Ihre Werkstatt	55
4	Entwicklungsumgebungen	85
5	Programmierung mit C und C++	171
6	ESP32-Programming inside	287
7	Peripherie und Funktionen des ESP32	363
8	Externe Schnittstellen	427
9	Externe Kommunikations- und Steuerungsmöglichkeiten	511
10	Projektideen	619

Inhalt

Einleitung	13
1 Der Mikrocontroller ESP32	19
1.1 Ein kurzer Rückblick in die Entwicklung von Mikrocontrollern	19
1.2 Die ESP32-SoCs	20
1.3 Die Module WROOM, SOLO, WROVER und PICO	26
1.4 Die Boards	29
1.5 Das ESP32-Dev-KitC V4	30
1.5.1 Das Pinout	32
1.5.2 Höhere Eingangsspannung an GPIOs	38
1.6 Das ESP32-Dev-KitC V4 – Erstinbetriebnahme am PC	41
1.6.1 Windows	41
1.6.2 Linux	42
2 Die Stromversorgung	43
2.1 Betriebszustände und typische Stromverbräuche	43
2.2 Anschlüsse zur Stromversorgung	45
2.3 Grundschemata der Stromversorgung	46
2.4 Spannungsquellen	47
2.5 Spannungsregler	50
2.5.1 Lineare Spannungsregler	50
2.5.2 Schaltregler	52
3 Ihre Werkstatt	55
3.1 Breadboard & Co.	55
3.1.1 Das Breadboard	55
3.1.2 Jumperkabel	56
3.1.3 Die Leiterplatte	57

3.2	Hilfreiche Werkzeuge	59
3.3	Fritzing	62
3.3.1	Was ist Fritzing?	62
3.3.2	Fritzing starten	62
3.3.3	In der Steckplattenansicht arbeiten	64
3.3.4	In der Schaltplanansicht arbeiten	70
3.3.5	In der Leiterplattenansicht arbeiten	72
3.3.6	Bauteile importieren	73
3.3.7	Bauteile ändern und selbst erstellen	75
3.4	Löttechnik	75
3.4.1	Bauteile auf Platinen löten	75
3.4.2	Bauteile entlöten	77
3.4.3	SMD löten	78
3.4.4	Ein ESP32-Modul löten	79

4 Entwicklungsumgebungen

4.1	Der Weg von lesbaren Anweisungen zum Maschinencode	85
4.2	Das Betriebssystem der Entwicklungsumgebung	88
4.3	esptool.py	89
4.3.1	Installation unter Windows	89
4.3.2	Installation unter Linux	94
4.3.3	Ausführbare Versionen von esptool	94
4.3.4	esptool einsetzen	94
4.4	Entwicklungsumgebungen in der Übersicht	97
4.4.1	Das Espressif IDF (ESP-IDF)	98
4.4.2	Die Arduino IDE auf dem PC	114
4.4.3	Der Arduino-Web-Editor	134
4.4.4	Eclipse	135
4.4.5	MicroPython	135
4.4.6	Lua	143
4.4.7	Rust	148
4.4.8	Visual Studio Code mit der Erweiterung PlatformIO	149
4.4.9	Visual Studio Code mit der Erweiterung ESP-IDF	160
4.4.10	Online-Konfiguratoren	164
4.4.11	Programme mit KI erstellen	165
4.5	Die Partitionstabelle festlegen	167
4.6	Fazit	169

5	Programmierung mit C und C++	171
5.1	Programmdesign für den Mikrocontroller	171
5.1.1	»Normale« C-/C++-Programme	173
5.1.2	Sketche	174
5.1.3	Erweiterte Programmstruktur	174
5.2	Syntax und Semantik in der Programmiersprache	175
5.3	Zeichen, Kommentare und Begriffe	176
5.3.1	Zeichen und ihre Bedeutung	176
5.3.2	Kommentare	178
5.3.3	Weitere wichtige Begriffe	178
5.4	Daten und Datentypen	179
5.5	Variablen	183
5.5.1	Deklaration und Definition	184
5.5.2	Initialisierung und Wertzuweisung	184
5.5.3	Gültigkeitsbereiche von Variablen	185
5.5.4	Typqualifizierer	186
5.5.5	Der Variablenmodifikator PROGMEM	187
5.6	Ausdrücke und Operatoren	187
5.6.1	Arithmetische Operatoren	188
5.6.2	Zuweisungsoperatoren	188
5.6.3	Bedingte Operatoren (Vergleiche)	189
5.6.4	Aussagenlogik	191
5.6.5	Bitoperatoren	192
5.6.6	Typumwandlung	195
5.6.7	Der Operator sizeof	197
5.6.8	Fehlerquellen	197
5.7	Zeiger (Pointer)	197
5.8	Arrays und Strukturen	201
5.8.1	Arrays (Tabellen)	201
5.8.2	struct (Strukturen)	203
5.8.3	union	204
5.8.4	enum	205
5.8.5	typedef	205
5.9	Kontrollstrukturen	205
5.9.1	Die if-Bedingung	206
5.9.2	if-else	207
5.9.3	switch ... case	209

5.9.4	Die while-Schleife	210
5.9.5	Die do-while-Schleife	212
5.9.6	Die for-Schleife	212
5.10	Funktionen	215
5.10.1	Die Syntax einer Funktion	216
5.10.2	Funktionsdeklaration und Funktionsdefinition	217
5.10.3	Der Funktionsaufruf	219
5.10.4	Übergabe von Parametern	221
5.10.5	Rücksprung	226
5.10.6	Rückgabe von Ergebnissen	227
5.11	Funktionen nutzen: Ihr Werkzeugkasten	229
5.11.1	Zeitfunktionen	229
5.11.2	Mathematische und trigonometrische Funktionen	230
5.11.3	Zufallszahlen	233
5.12	Klassen und objektorientierte Programmierung	235
5.12.1	Deklariere eine Beispielklasse klasseKreis	237
5.12.2	Funktionen für eine Klasse einrichten	237
5.12.3	Mit Konstruktoren arbeiten	238
5.12.4	Die Zugriffsspezifizierer einsetzen	239
5.12.5	Klassen in der Arduino IDE	242
5.13	Die String()-Klasse	243
5.14	Serielle Aus- und Eingabe	252
5.15	Vorgriff: Tasks und Threads	258
5.16	Präprozessor Direktiven	259
5.16.1	Die Direktive #define	259
5.16.2	Die Direktive #include	261
5.16.3	Die #ifdef- und die #endif-Direktive	261
5.17	Prinzipien des Softwareentwurfs	262
5.18	Fehlermanagement	263
5.18.1	Einfache Methoden	263
5.18.2	Aufwendigere Methoden	265
5.18.3	Professionelle Methoden	273
5.18.4	Datenblätter	282
5.19	JSON – das Datenformat für den Datenaustausch	283

6 ESP32-Programmierung inside

287

6.1	ULP-Programmierung	287
6.2	FreeRTOS-Tasks verwalten	316
6.3	FreeRTOS-Task-Synchronisation	327
6.3.1	Mutexe und Semaphore	330
6.3.2	Event-Groups	338
6.3.3	Task-Notifications	342
6.3.4	Queues	343
6.3.5	Message-Buffer und Stream-Buffer	345
6.3.6	Ring-Buffer	353
6.4	Threads nutzen	358

7 Peripherie und Funktionen des ESP32

363

7.1	Der interne Hall-Sensor	363
7.2	Digitale Aus-/Eingänge	364
7.2.1	Digitale Ausgänge	365
7.2.2	Digitale Eingänge	367
7.3	Analoge Aus-/Eingänge	373
7.3.1	Analoge Ausgänge	373
7.3.2	Analoge Eingänge	374
7.4	Das ESP32-Touch-Sensor-System	385
7.5	DHT11 und DHT22	390
7.6	433-MHz-Steuerung	395
7.7	LED-Control und Pulsweitenmodulation	400
7.7.1	Pulsweitenmodulation	400
7.7.2	Die ESP32-LED-Steuereinheit	401
7.8	Impulszähler	407
7.9	Universal-Timer, Watchdog-Timer	410
7.9.1	Universal-Timer	411
7.9.2	Watchdog-Timer	413
7.10	Energie-(Spar-)Modi	415
7.11	Daten auf dem ESP32 speichern	423

8

Externe Schnittstellen

427

8.1	SPI	427
8.1.1	RFID mit RFID-RC522	428
8.1.2	SD-Karte	435
8.1.3	TFT-Display	441
8.2	I ² C	445
8.3	I2S	449
8.4	Bluetooth	460
8.4.1	Bluetooth Classic	460
8.4.2	Bluetooth Low Energy (BLE)	462
8.5	WLAN/Wi-Fi	477
8.5.1	ESP32 als AP	478
8.5.2	ESP32 als »Station«	483
8.5.3	ESP32 mit ESP-Now	490
8.5.4	ESP32-Mesh	499
8.6	UART	500
8.6.1	Die UART-Schnittstellen des ESP32	502
8.6.2	Ein UART-Echo	503
8.6.3	ESP32-Modul flashen	505
8.7	OTA	506

9

Externe Kommunikations- und Steuerungsmöglichkeiten

511

9.1	Das JSON-Datenaustauschformat in der Praxis	512
9.2	Apps	517
9.2.1	Blynk (2.0)	517
9.2.2	MIT App Inventor	526
9.3	Messenger	538
9.3.1	E-Mail-Versand	538
9.3.2	Pushbullet	547
9.3.3	Alexa	555
9.3.4	Telegram und WhatsApp	558
9.4	Das MQTT-Protokoll	558

9.5	Cloud-IoT-Plattformen	566
9.5.1	Cayenne	567
9.5.2	Adafruit IO	572
9.5.3	IFTTT	581
9.6	Lokale Anwendungen nutzen	592
9.6.1	Node-RED	593
9.6.2	ioBroker	606

10 Projektideen 619

10.1	The Evil Dice	619
10.2	Die Maker-Uhr	625
10.3	Das Dateisystem einmal anders	645
10.4	Solar-Repeater	655
10.5	Als die Bilder laufen lernten	659
10.5.1	Ein Schwarz-Weiß-Bild anzeigen	659
10.5.2	Ein RGB-Bild anzeigen	663
10.5.3	Eine TFT-Slide-Show	665
10.6	IoT-Barometer	672
10.6.1	Der Sensor BMP280	673
10.6.2	Das I ² C-OLED-Display SSD1306	677
10.6.3	Beide Module an einem gemeinsamen I ² C-Bus	684
10.6.4	Beide Module an getrennten I ² C-Kanälen	686
10.7	Eine kleine Schaltzentrale	688
10.8	Den ESP32 mit Alexa steuern	706
10.9	Mit Text To Speech und ChatGPT experimentieren	715
10.9.1	Text To Speech	715
10.9.2	ChatGPT	720

A Anhang 725

A.1	Sensoren	725
A.2	Fritzing-Parts	726
A.3	Verwendete Software und Softwareversionen	727
A.4	RTC-MUX-Pin-Liste	728
Index		731