

Sketch, Variablen und Serial Monitor

Die Struktur eines Arduino-ESP32-Programms lesen, veraendern und systematisch testen.

MERKEN

setup() laeuft nach Start oder Reset genau einmal. loop() wird danach fortlaufend wiederholt. Ein Semikolon beendet viele Anweisungen; geschweifte Klammern bilden Bloecke.

GRUNDSTRUKTUR

```
const int LED_PIN = 18;

void setup() {
  Serial.begin(115200);
  pinMode(LED_PIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_PIN, HIGH);
  delay(500);
  digitalWrite(LED_PIN, LOW);
  delay(500);
}
```

LESEREGEL

```
const int LED_PIN = 18;
// Typ + Name + Wert

pinMode(LED_PIN, OUTPUT);
// Funktion(Argumente);

if (wert > 500) {
  // nur wenn wahr
} else {
  // sonst
}
```

Baustein	Beispiel	Bedeutung
Datentyp	int, bool, float, uint32_t	Legt fest, welche Werte und welcher Speicherbereich verwendet werden.
Variable	int messwert = 0;	Benannter Speicher. Den Namen sprechend waehlen.
Konstante	const int PIN = 18;	Wert soll waehrend des Programms nicht geaendert werden.
Bedingung	if (wert >= 500)	Fuehrt einen Block nur aus, wenn der Vergleich wahr ist.
Funktion	void ledAus() { ... }	Fasst wiederkehrende Schritte unter einem Namen zusammen.
Bibliothek	#include <WiFi.h>	Stellt fertige Funktionen bereit; Dokumentation und passende Version beachten.

TYPISCH

= weist einen Wert zu. == vergleicht zwei Werte. Serial.begin(115200) und die Einstellung im Serial Monitor muessen identisch sein.

Sketch, Variablen und Serial Monitor

Die Struktur eines Arduino-ESP32-Programms lesen, veraendern und systematisch testen.

MESSWERT PRUEFEN UND LESBAR AUSGEBEN

```
const int SENSOR_PIN = 34;

void setup() {
  Serial.begin(115200);
}

void loop() {
  int messwert = analogRead(SENSOR_PIN);

  Serial.print("ADC: ");
  Serial.println(messwert);

  if (messwert > 2000) {
    Serial.println("Grenzwert ueberschritten");
  }

  delay(250);
}
```

Erst Rohwert anzeigen, dann Bedingungen ergaenzen. So bleibt die Fehlersuche nachvollziehbar.

Meldung/Problem	Bedeutung	Pruefung
expected ';'	Semikolon fehlt oft in der Zeile davor	Zeile und vorherige Zeile pruefen
not declared	Name unbekannt oder anders geschrieben	Gross-/Kleinschreibung und Geltungsbereich
expected '}'	Block wurde nicht geschlossen	Klammerpaare und Einrueckung pruefen
Upload ok, nichts sichtbar	Ausgabe fehlt oder Monitor falsch	Serial.begin, println und Baudrate
Programm reagiert spaet	Lange delay()-Pausen blockieren	Kurze Tests; spaeter millis() verwenden

KOMPILIEREN

- Syntax wird geprueft
- Ersten Fehler zuerst beheben
- Danach erneut kompilieren

HOCHLADEN

- Richtiges Board
- Richtiger Port
- Upload-Ausgabe lesen

BEOBACHTEN

- Werte beschriften
- Nur relevante Ausgaben
- Fehler schrittweise eingrenzen

PRAXIS

Neue Funktionen einzeln erproben. Funktioniert der Minimaltest, den naechsten Baustein ergaenzen. Kommentare erklaren das Warum - nicht jede offensichtliche Zeile.