

# Informatik

## Programmierung:

Die Programmierung des Lernkoffers kann je nach Schulumgebung, Vorerfahrung der Schülerinnen und Schüler sowie verfügbarer Zeit im Unterricht auf drei unterschiedlichen Niveaus erfolgen. Jedes Niveau verfolgt ein eigenes Lernziel und adressiert unterschiedliche Kompetenzbereiche.

### 1. Plug-and-Play – Einstieg ohne Programmierkenntnisse

**Zielgruppe:**

Schülerinnen und Schüler ohne Vorerfahrung in der Programmierung.

**Ausstattung:**

Der Lernkoffer ist vollständig vorinstalliert. Beim Einschalten durchläuft der Bordcomputer automatisch einen Startprozess, bei dem alle Sensoren initialisiert und die Datenerfassung gestartet wird.

**Mögliche Aufgabenstellungen:**

- Beobachtung des Systemverhaltens: Welche Daten werden erfasst?
- Interpretation von Sensordaten im Webinterface (z. B. Beschleunigung, Höhe, Spannung)
- Exportieren und Auswerten der CSV-Daten am PC (z. B. in Excel, LibreOffice oder Python)
- Verständnisfragen: Was ist ein Sensor? Wie funktioniert ein Mikrocontroller?
- Physikalische Einordnung der Werte (z. B. Luftdruck ↔ Höhe)

**Lernziele:**

- Erste Begegnung mit digitalen Messsystemen
- Grundverständnis für Datenverarbeitung und Datenauswertung
- Förderung der Medienkompetenz ohne Einstieg in Programmiersyntax

### 2. Programmierung mit Unterstützung – Geführtes Lernen mit PlatformIO

**Zielgruppe:**

Schülerinnen und Schüler mit ersten Programmiererfahrungen oder technikaffine Gruppen.

**Ausstattung:**

Rechner mit Visual Studio Code (VS Code) und der PlatformIO-Erweiterung. Das Projekt wird bereitgestellt und kann geöffnet, angepasst und wieder auf den ESP32 hochgeladen werden.

**Typische Inhalte und Aufgaben:**

- Methoden anpassen: z. B. triggerParachute() nur bei bestimmtem Höhenabfall
- Sensorwerte filtern oder glätten (Moving Average, Schwellenwert)
- Neue Webinterface-Befehle implementieren (z. B. „Start Aufnahme“)
- LED oder Display-Ausgabe hinzufügen
- Logik für Messungen ändern (z. B. kürzere Intervalle, Schwellenwerte dynamisch setzen)

**Lernziele:**

- Verstehen von Methoden, Variablen und Bedingungen in C++
- Umgang mit Entwicklungsumgebungen (VS Code + PlatformIO)
- Erweitern bestehender Programme um eigene Ideen
- Teamarbeit beim Entwickeln und Testen

### 3. Selbst Programmieren – Freies Projekt für Fortgeschrittene

**Zielgruppe:**

Leistungsstarke oder besonders motivierte Schülerinnen und Schüler mit umfangreicher Vorerfahrung oder im Rahmen von Wahlkursen / Projekttagen.

**Ausstattung:**

Beliebige Entwicklungsumgebung (PlatformIO, Arduino IDE oder andere), Internetzugang zur Recherche.

**Anforderungen:**

- Selbständige Initialisierung der Sensoren über I2C/SPI
- Nutzung und Einbindung passender Bibliotheken (z. B. Adafruit, SparkFun)
- Strukturierung eines eigenen Programms: setup(), loop(), Klassenbildung
- Logging auf SD-Karte, Webinterface, OTA-Update (je nach Zielsetzung)

**Mögliche Projekte:**

- Raketenflug vollständig selbst messen und analysieren
- Dashboards in Echtzeit gestalten
- Datenauswertung mit Python oder in der Cloud
- Kombination mehrerer Sensoren (z. B. BME280 + ICM20948)

**Lernziele:**

- Selbstorganisiertes und forschendes Lernen
- Anwendung fortgeschrittener Programmierkonzepte
- Verstehen von Kommunikationsprotokollen und Datenformaten
- Problemlösestrategien und Debugging

### Hinweis für Lehrkräfte:

Alle drei Stufen können auch parallel in einer Lerngruppe stattfinden. Die Lernkoffer bieten genug Flexibilität, um unterschiedliche Anforderungsniveaus zu bedienen und individuelle Lernpfade zu ermöglichen.

Hier ein Tutorial, wie man den Fallschirm richtig faltet, dass er zuverlässig aufgeht.

<https://www.youtube.com/shorts/kGTpw6yevmc?feature=share>



Eine genaue Startanleitung ist in „Plug-and-Play“ beschrieben. Ansonsten gibt es eine kurze Videoanleitung unter:

[https://www.youtube.com/watch?v=\\_mfvzam6YKM&ab\\_channel=RaketfuedRockets](https://www.youtube.com/watch?v=_mfvzam6YKM&ab_channel=RaketfuedRockets)