

Black:Anni

Konzept

Ich möchte gerne ein Motorrad Steuergerät entwickeln, das für meine Kawa mit Vergaser funktioniert.

Ziel ist es Motorkenndaten zu sammeln.

Und wichtigster Aspekt: Code generieren mit praktischen Anwendungen!

Neben Kosten sind auch die Integration der Elektronik Bauteile in Wasser und Stoßfesten Baugruppen eine Herausforderung!

Sensorik wird alles wild was geht zusammen getragen!

Zugriff auf die Daten wird ein Frontend im Browser werden.

Fortschritt

- Bestellungen

wir haben folgende Komponenten bestellt

GPS TRACKING: ★★★★

- Multi-band L1/L5 (NEO-M9N)
- Präzise Position
- Geschwindigkeit
- Heading/Richtung
- Altitude (GPS)

MOTION TRACKING: ★★★★★ (KILLER!)

-
- PROF
- Lean Angle bis 64°! 🏍
 - G-Force (Beschleunigung)
 - Wheelie/Stoppie Detection
 - Cornering Speed
 - Vibration Analysis
 - Compass Heading

UMWELT-DATEN: ★★

- Altitude (BME280 + GPS Fusion) ⛲
- Barometric Pressure
- Temperature
- Humidity

FAHRZEUG-DATEN:

- Tacho Signal (Hall Sensor)
- 4× Analog Inputs (ADS1115)
- Battery Voltage (INA260)

- Battery Current (INA260)
- Power Consumption (Watt)
- Fuel Gauge (MAX17048)

DISPLAY & OUTPUT:

- 8-LED Bargraph (Speed/RPM)
- 1.3" OLED (Live Data)
- 64GB SD-Karte (Logging)
- RTC Timestamps

DAS FEHLT NUR:

 Tire Pressure (TPMS)

→ Kommt später! 

Software Struktur

```

md
BikeLogger/
|—— BikeLogger.ino (Main + Setup)
|—— config.h (Alle Pins, I2C Adressen, Konstanten)
|—— sensors/
|   |—— gps_module.h/.cpp (NEO-M9N)
|   |—— imu_module.h/.cpp (ICM-20948 - LEAN ANGLE!)
|   |—— bme280_module.h/.cpp (Umwelt)
|   |—— ina260_module.h/.cpp (Power)
|   |—— ads1115_module.h/.cpp (Analog Inputs)
|   |—— rtc_module.h/.cpp (DS3231)
|—— display/
|   |—— oled_display.h/.cpp (SSD1306)
|   |—— led_bargraph.h/.cpp (74HC595)
|—— storage/
|   |—— sd_logger.h/.cpp (64GB SD-Card)
|—— utils/
|   |—— data_structures.h (Structs für Daten)
|   |—— calculations.h (Lean Angle, G-Force)

```

—
PROF