Projektarbeit: Elektronische IoT-Waage mit Temperatur-Logging und OLED-Display

Projektbeschreibung

Es soll eine elektronische IoT-Waage gebaut werden, welche periodisch (alle 60 Sekunden) das Gewicht der Last in kg misst. Zusätzlich soll die Temperatur der Last gemessen werden. Diese Messwerte sollen einerseits auf einem OLED-Display dargestellt und andererseits periodisch via WLAN in eine InfluxDB-Datenbank gespeichert werden. Die Werte können dann via Grafana als Zeitreihe visualisiert werden. Alle Projektdetails, Schaltpläne, Code und Dokumentation sollen öffentlich zur freien Verfügung auf GitHub veröffentlicht werden.

Grundsätzlicher Aufbau des Systems

1. Waage

Die Waage wird aus 4 Wägezellen aufgebaut, welche zwischen zwei 30 × 30 cm großen Aluminiumplatten liegen und in deren Ecken angebracht sind. Damit die Wägezellen auf der Aluminiumplatte befestigt werden können, muss dafür in MCAD SolidWorks für jede Wägezelle ein Befestigungsrahmen entworfen werden, welcher via 3D-Druck gefertigt werden soll.

Die 4 Wägezellen werden an einem HX711 24-Bit-ADC in einer Wheatstone'schen Messbrücke angeschlossen. Der HX711 wird über ein 2-Wire-Bitbang-Protokoll von einem ESP32-WROOM (Adafruit Feather V2) ausgelesen.

Spezifikationen:

- Messbereich: 0–100 kg

2. Temperaturmessung

Die Temperatur soll mit einem DS18B20-Sensor über das One-Wire-Protokoll periodisch vom ESP32-WROOM (Adafruit Feather V2) gemessen werden.

3. OLED-Display

Auf einem OLED-Display (SSD1306, 128×64 Pixel) sollen die aktuellen Messergebnisse dargestellt werden. Das OLED-Display wird via I^2C vom ESP32-WROOM (Adafruit Feather V2) angesteuert.

4. Mikrocontroller und Datenübertragung

Der ESP32-WROOM (Adafruit Feather V2) liest periodisch die Werte vom HX711 und DS18B20 aus, zeigt die Werte am Display an und sendet die Messwerte an die InfluxDB-Datenbank (die Datenbank mit IP-Adresse etc. wird zur Verfügung gestellt).

Stromversorgung: USB-Netzteil (5V)

Ziele

- Sensoren in Betrieb nehmen
- Mikrocontroller-Programmierung mit dem ESP32
- Kommunikationsprotokolle (I^2C, One-Wire, SPI) zu verstehen und anzuwenden
- Schaltpläne professionell in KiCad zu erstellen
- 3D-CAD-Modelle in SolidWorks zu entwerfen
- Versionskontrolle mit Git/GitHub anzuwenden
- Professionelle technische Dokumentation zu erstellen
- Messungen mit Oszilloskop und Logic Analyzer durchzuführen und zu interpretieren
- IoT-Datenübertragung (WLAN, InfluxDB) zu implementieren

Aufgabenbeschreibung

1. Projektzeitraum

- **Start:** Dezember 2025
- **Abgabe:** 31. Januar 2026

2. Recherche und Konzept

- Recherchiere die einzelnen Komponenten (HX711, DS18B20, ESP32, OLED SSD1306)
- Verstehen des Zusammenspiels der Komponenten im System
- **Dokumentation:** Die Haupt Dokumentation soll in einen Word File fest gehalten werden. Alle Notizen zur Recherche müssen handschriftlich in einem Notizbuch festgehalten werden.

3. Projektplanung

- Erstelle in einem Word-Dokument ein **Inhaltsverzeichnis** für die Projektdokumentation. Dieses Inhaltsverzeichnis stellt bereits den Aufbau des Projekts dar. Die Kapitel werden im Laufe des Projekts geschrieben.
- Erstelle eine **Projektplanung** (Roadmap) mit Zeitplanung und Meilensteinen, welche die einzelnen Design-Schritte vom Projektstart bis zur Abgabe am 31. Januar 2026 definiert.

4. CAD-Design (SolidWorks)

- Entwerfe einen Befestigungsrahmen für die einzelnen Wägezellen in SolidWorks
- Fertige den Rahmen via 3D-Druck
- Dokumentiere das CAD-Design mit Screenshots und technischen Zeichnungen

5. Schaltplan (KiCad)

- Zeichne die gesamte Elektronik sauber in einem KiCad-Schaltplan
- Beschrifte alle Komponenten und Anschlüsse
- Füge den Schaltplan in die Dokumentation ein

6. Stückliste (BOM)

- Erstelle eine vollständige Stückliste (Bill of Materials) als Excel-Datei
- Inkludiere: Bauteilbezeichnung, Anzahl, Händler, Bestellnummer, link, Preis
- Füge die BOM in die Dokumentation ein

7. Hardwareaufbau

- Baue die Komponenten auf einem Steckbrett oder einer Platine auf bzw. löte sie
- Achte auf eine saubere und professionelle Ausführung

8. Softwareentwicklung

a) Versionskontrolle

- Der gesamte Code muss zwingend mit Git-Versionskontrolle erstellt werden
- Erstelle ein GitHub-Repository für das Projekt
- Verwende aussagekräftige Commit-Messages

b) Entwicklungsschritte

Schritt 1: Basis-Funktionalität (ohne WLAN)

- Baue die Waage mit Temperatursensor und OLED-Display auf
- Der MCU soll die Sensoren alle 15 Sekunden auslesen
- Die Daten werden via OLED-Display angezeigt
- Verwende Arduino-Bibliotheken und -Dokumentation

Schritt 2: Protokollanalyse

- Nimm die I^2C-Bus-Kommunikation mit dem OLED via (USB-)Oszilloskop auf
- Dekodiere die Kommunikation mit einem Logic Analyzer
- Dokumentiere die Startsequenz der Kommunikation vom MCU zum OLED via Screenshot
- Erkläre die aufgenommenen Signale in der Dokumentation

Schritt 3: Kalibrierung der Waage

- Die Software muss eine Kalibrier Funktion haben, dies kann in einem separaten Kalibrier Code durchgeführt werden um die Kalibrierfaktoren zu finden. Kalibriere die Waage mit mindestens 3 bekannten Gewichten (z.B. 0 g, 500 g, 1000 g, 5000 g, 10000 g (Eimer Wasser etc.)
- Dokumentiere den Kalibrierungsprozess und die Kalibrierfaktoren
- Überprüfe die Messgenauigkeit nach der Kalibrierung

Schritt 4: WLAN-Integration

- Integriere das WLAN-Modul in den Code
- Verbinde den ESP32 mit der InfluxDB-Datenbank
- Die Code-Details und Datenbank-Credentials werden zur Verfügung gestellt
- Sende die Messwerte alle 60 Sekunden an die Datenbank

9. Dokumentation

Die Dokumentation soll folgende Kapitel umfassen (Mindestanforderung):

1. Einleitung und Projektübersicht
2. Zielsetzung
3. Systemarchitektur und Blockschaltbild
4. Komponentenbeschreibung
5. CAD-Design (SolidWorks)
6. Schaltplan (KiCad)
7. Stückliste (BOM)
8. Softwareentwicklung und Code-Beschreibung
9. Kalibrierung
10. Messungen und Tests (Oszilloskop, Logic Analyzer)
11. Inbetriebnahme und Resultate
12. Probleme und Lösungen
13. Fazit und Ausblick

10. GitHub-Veröffentlichung

- Veröffentliche das gesamte Projekt auf GitHub
- Inkludiere: Code, Schaltpläne (KiCad-Files), CAD-Dateien (STEP/STL), Dokumentation (PDF), BOM
- Erstelle eine aussagekräftige README.md-Datei

Bewertungskriterien

Die Arbeit wird nach folgenden Kriterien bewertet:

- **Funktionalität** (30%): Funktioniert die Waage wie gefordert?
- **Dokumentation** (25%): Ist die Dokumentation vollständig, verständlich und professionell?
- **Code-Qualität** (15%): Ist der Code sauber, kommentiert und wartbar?
- **Schaltplan/CAD** (15%): Sind die technischen Zeichnungen korrekt und professionell?
- **Eigenständigkeit** (10%): Wurde das Projekt selbstständig durchgeführt?
- **GitHub/Versionskontrolle** (5%): Wurde Git korrekt verwendet?

Verfügbare Ressourcen

- Arduino & Adafruit Bibliotheken: HX711, OneWire, DallasTemperature, Adafruit_SSD1306
- SolidWorks-Lizenz
- KiCad (Open Source)
- Oszilloskop und Logic Analyzer
- 3D-Drucker
- InfluxDB-Zugangsdaten (werden zur Verfügung gestellt)

Viel Erfolg bei der Umsetzung!