



CPS_for_Industrial_Applications_Project_Batch_10

PROJECT NAME:DIGITAL WATER METERING SYSTEM

ROLL NUMBERS:

21781A0402 - A. LeelaKrishna

21781A0434 – B.Srinivasa

21781A0442 - CH Maddeiah

21781A0465 - G Sai Kumar

Project Github Link: https://github.com/Chandu12867/SVCET_Batch_10_cps

Aim of the Project:

The aim of this project is to develop a smart inventory management system that leverages hardware and software integration to track inventory movement, automate stock counting, and optimize storage space utilization. This system is designed to enhance productivity and reduce errors in small warehouses.

Problem Statement and Solution:

Problem Statement: Small warehouses face significant challenges in managing inventory efficiently due to manual processes and limited resources, leading to inefficiencies, errors, and suboptimal space utilization.

Solution: Design a compact, low-cost smart inventory management system that integrates a microcontroller with sensor modules for real-time inventory tracking. Develop software algorithms for accurate data interpretation and a user-friendly web interface for remote access.

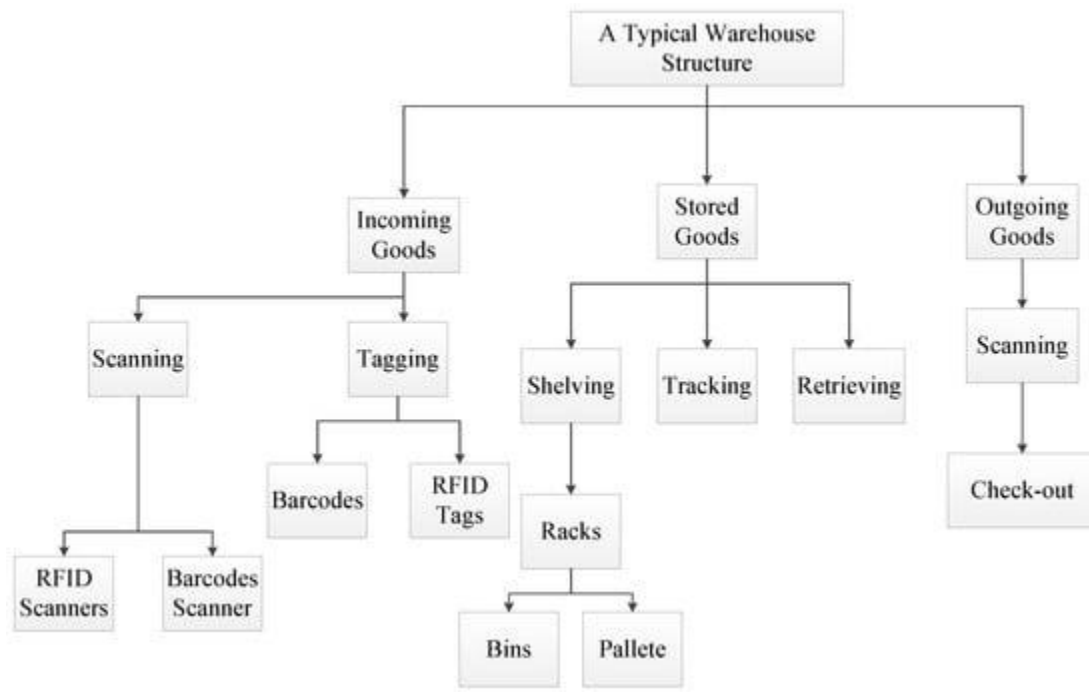
Project Design Specification:

- **Microcontroller:** ESP32 or Arduino
- **Sensor Module:** Ultrasonic sensor for distance measurement, load cell for weight measurement, and RFID for item identification
- **Connectivity:** Wi-Fi for real-time data transmission
- **Power Supply:** 5V DC adapter or battery pack
- **Web Interface:** HTML, CSS, JavaScript, and a backend server using Python (Flask/Django) or Node.js

Project Architecture:

1. **Hardware Layer:**
 - **Microcontroller (ESP32/Arduino):** Central processing unit for sensor data collection and transmission.
 - **Sensors:** Ultrasonic sensors to measure distance (stock levels), load cells to measure weight (inventory quantity), and RFID readers to identify items.
2. **Communication Layer:**
 - **Wi-Fi Module:** ESP32's built-in Wi-Fi or an external Wi-Fi module for Arduino to send data to the server.
3. **Server Layer:**
 - **Backend Server:** Python (Flask/Django) or Node.js server to handle incoming data and process it.
 - **Database:** MySQL or SQLite to store inventory data.
4. **Web Interface Layer:**
 - **Frontend:** HTML, CSS, JavaScript to display inventory data.

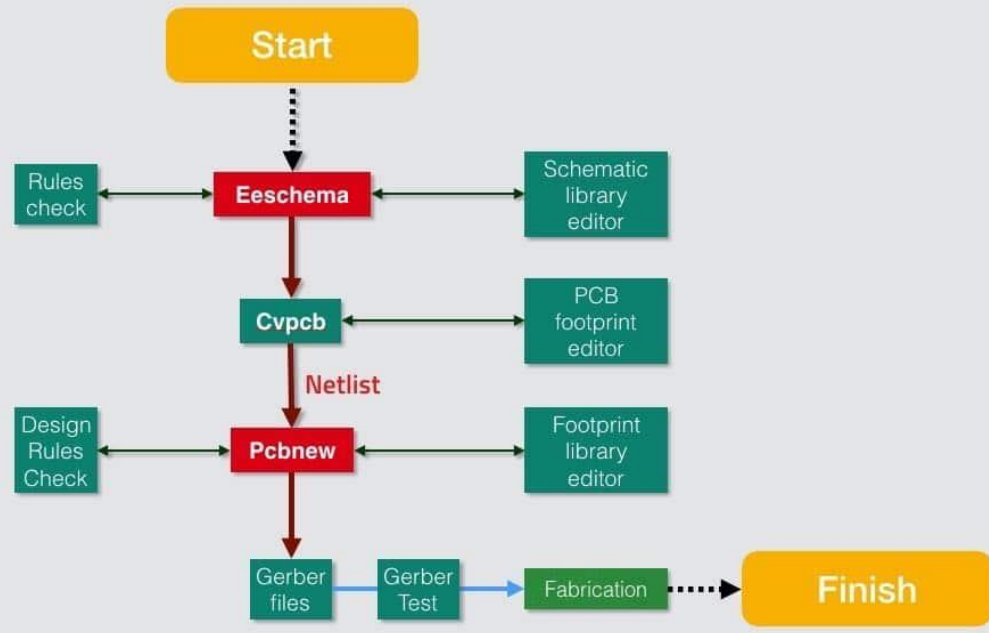
- **Backend API:** RESTful API to serve data to the frontend.



Flow Explanation:

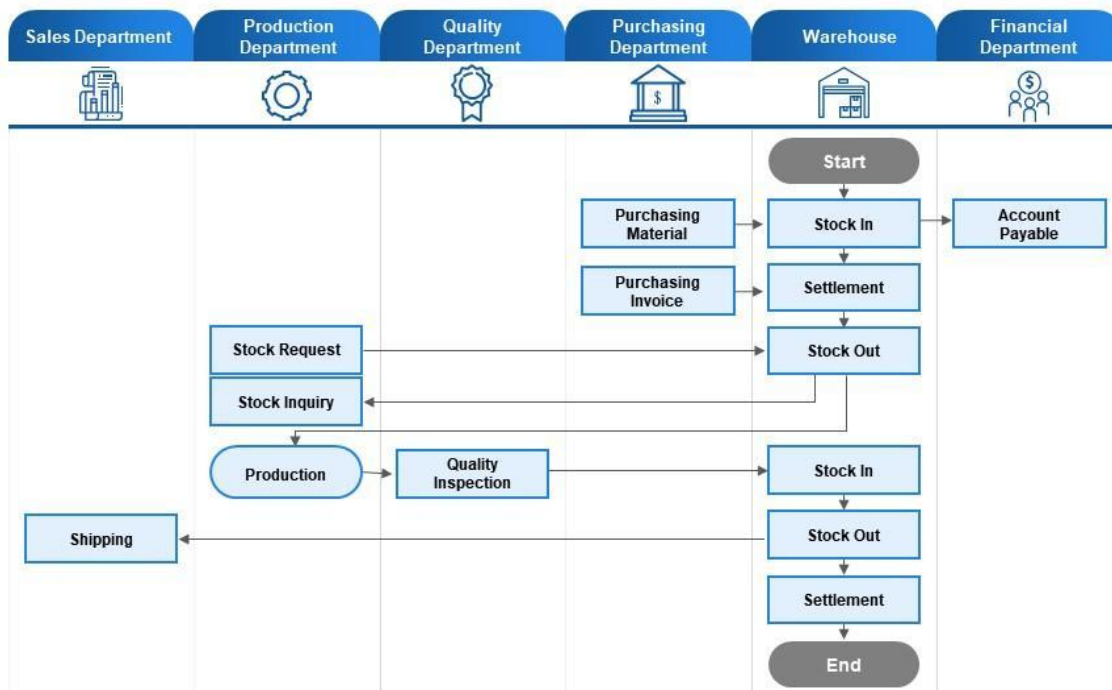
1. **Initialization:** System initializes and connects to Wi-Fi.
2. **Data Collection:** Sensors collect data on inventory (distance, weight, item ID).
3. **Data Processing:** Microcontroller processes the sensor data.
4. **Data Transmission:** Processed data is sent to the backend server via Wi-Fi.
5. **Data Storage:** Server stores the data in a database.
6. **User Interface:** Data is displayed on the web interface for user interaction and monitoring.

The Kicad design workflow



Process Flow Chart of Inventory Management

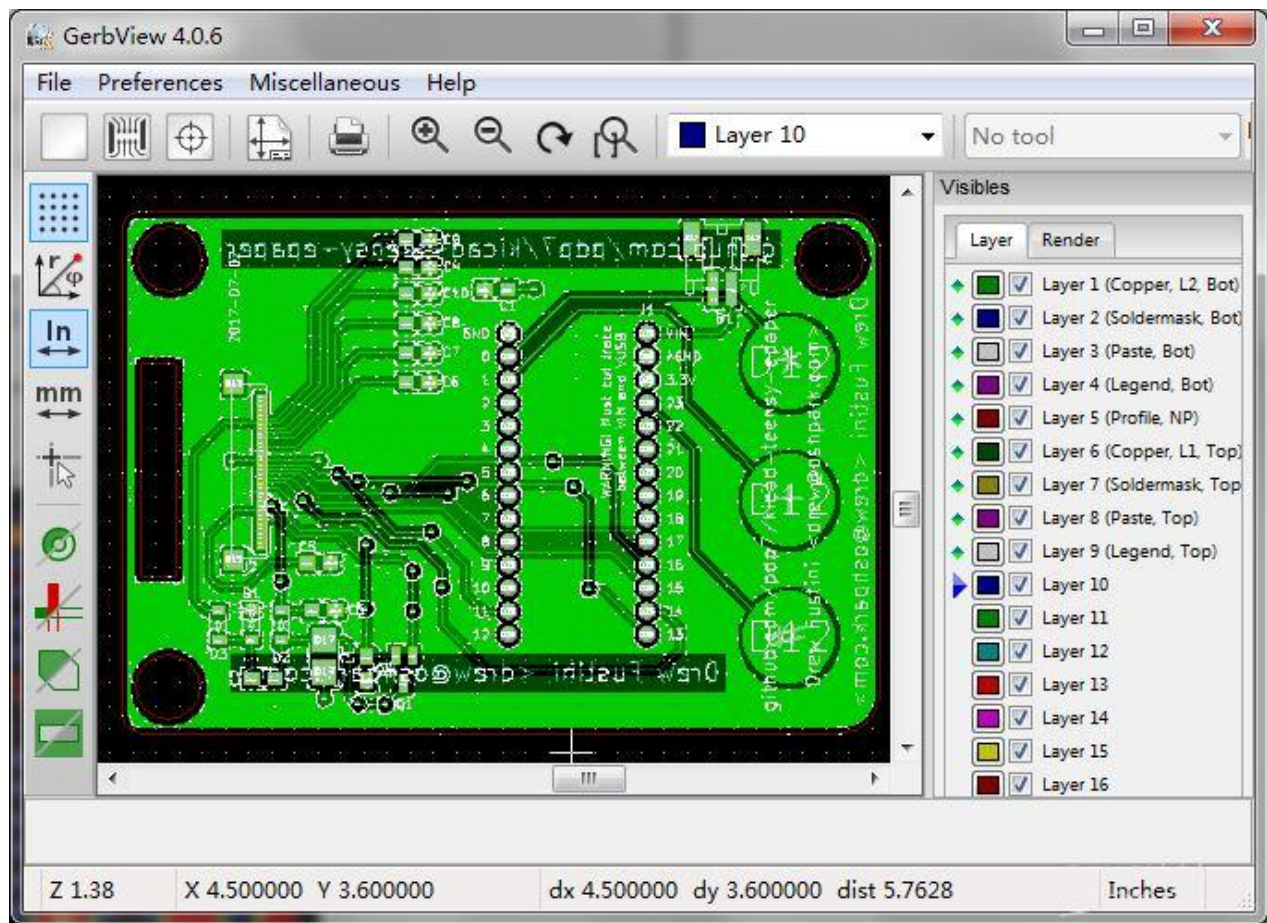
This slide is 100% editable. Adapt it to your need and capture your audience's attention.



Wiring Diagram:

- **Microcontroller (ESP32/Arduino):**

- Ultrasonic Sensor: Trig to GPIO, Echo to GPIO, VCC to 5V, GND to GND.
- Load Cell: Connected to HX711 module, DT and SCK to GPIO, VCC to 5V, GND to GND.
- RFID Reader: SDA to GPIO, SCK to GPIO, MOSI to GPIO, MISO to GPIO, IRQ to GPIO, GND to GND, RST to GPIO, 3.3V to 3.3V.



Components Working Principles/Functionality:

- **Microcontroller (ESP32/Arduino):** Executes the program, reads sensor data, and communicates with the server.
- **Ultrasonic Sensor:** Measures distance by emitting sound waves and calculating the time taken for the echo to return.
- **Load Cell:** Measures weight through strain gauge deformation, outputting an electrical signal proportional to the load.

- **RFID Reader:** Reads unique IDs from RFID tags attached to inventory items.

Project Outcome:

- **Efficient Inventory Management:** Real-time tracking of inventory levels and movement.
- **Error Reduction:** Automated stock counting reduces human errors.
- **Optimized Space Utilization:** Accurate measurement and data analysis for better storage management.
- **User-Friendly Interface:** Easy access to inventory data via a web interface.

