HARDWARE SPECIFICATION

Overview

The proposed system uses multiple sensors interfaced with an ESP32 microcontroller to monitor critical freezer parameters. Each sensor is selected based on reliability, ease of integration, and suitability for real-time monitoring. This section describes the function of each sensor along with its pin configuration.

A) Cabinet Temperature Sensor

Purpose:
The cabinet temperature sensor monitors the internal freezer temperature to ensure it remains within the safe operating range.

Sensor Type: Digital temperature sensor (DS18B20)

Pin Configuration:

Sensor Pin	ESP32 Pin	Description
VCC	3.3V	Power supply
GND	GND	Ground
DATA	GPIO 4	Temperature data line

Operation:
The sensor provides accurate digital temperature readings, which are periodically read by the ESP32 and transmitted to the cloud for monitoring and anomaly detection.

B) Evaporator Temperature Sensor

Purpose:
This sensor measures the evaporator temperature to evaluate cooling efficiency and superheat behaviour.

Sensor Type:
Digital temperature sensor

Pin Configuration:

Sensor Pin	ESP32 Pin	Description
VCC	3.3V	Power supply
GND	GND	Ground

Sensor Pin	ESP32 Pin	Description
DATA	GPIO 5	Temperature data line

Operation:
Comparing cabinet and evaporator temperatures helps identify abnormal cooling performance and compressor inefficiencies.

C) Vibration Sensor (Compressor Health)

Purpose:
The vibration sensor monitors compressor vibrations to detect mechanical wear, imbalance, or abnormal operation.

Sensor Type:
Digital vibration sensor / SW-420 module

Pin Configuration:

Sensor Pin	ESP32 Pin	Description
VCC	3.3V	Power supply
GND	GND	Ground
DO	GPIO 27	Digital vibration output

Operation:
The sensor outputs a digital signal when vibration exceeds a predefined threshold, enabling real-time anomaly detection.

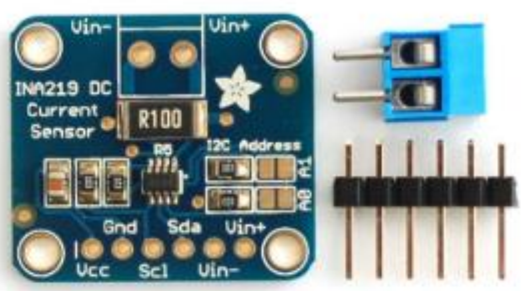
D)Power and Current Sensor (INA219)

Purpose:
The INA219 sensor measures voltage, current, and power consumption of the freezer system.

Sensor Type:
INA219 High-Side Current Sensor

Pin Configuration (I²C):

INA219 Pin	ESP32 Pin	Description
VCC	3.3V	Power supply
GND	GND	Ground
SDA	GPIO 21	I²C data



INA219 Pin	ESP32 Pin	Description
SCL	GPIO 22	I²C clock

Operation:
The sensor continuously measures power parameters, allowing detection of electrical faults and energy inefficiencies.

E) Humidity Sensor

Purpose:

The humidity sensor is used to monitor the **relative humidity level inside the freezer environment**. Abnormal humidity levels may indicate moisture ingress, improper sealing, frost formation, or frequent door opening, which can negatively affect product quality and freezer efficiency.

Sensor Type:

Digital Temperature and Humidity Sensor (DHT11 / DHT22)

Pin Configuration:

Sensor Pin	ESP32 Pin	Description
VCC	3.3V	Power supply
GND	GND	Ground
DATA	GPIO 26	Digital humidity data

Operation:

The humidity sensor contains a capacitive humidity sensing element that changes its capacitance based on the moisture content in the air. This variation is internally processed and transmitted as a digital signal through a single data line.

G) Door Status Sensor

Purpose:
The door sensor detects door open and close events to prevent temperature loss and energy wastage.

Sensor Type:
Magnetic reed switch

Pin Configuration:

Sensor Pin	ESP32 Pin	Description
Signal	GPIO 14	Door status input
GND	GND	Ground

Operation:
The ESP32 monitors the digital input state to determine door status and triggers alerts if the door remains open beyond a safe duration.

H) Electronic Expansion Valve (EEV)

Purpose

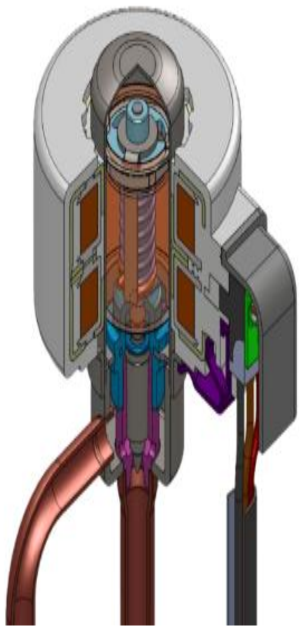
The Electronic Expansion Valve (EEV) regulates the **flow of refrigerant** entering the evaporator. Precise control of refrigerant flow is critical for maintaining optimal cooling performance, preventing liquid flood-back to the compressor, and improving system efficiency.

Working Principle

The EEV is a **stepper motor–controlled valve** whose opening is adjusted electronically based on system conditions. By increasing or decreasing the valve opening, the amount of refrigerant entering the evaporator is controlled.

In an intelligent refrigeration system, EEV control is typically based on:

- Evaporator temperature
- Suction pressure
- Calculated superheat
- Load conditions



Control Strategy in the Proposed System

In the proposed IoT freezer system, the EEV control logic is conceptually implemented. Since a physical EEV actuator was not directly driven, its operation is simulated using sensor data and software logic.

The system determines the required EEV position using:

- Cabinet temperature
- Evaporator temperature
- Estimated superheat values

I)Pressure Sensors

Purpose

Pressure sensors are used to monitor **refrigerant pressure** at critical points in the refrigeration cycle, particularly at the suction and discharge sides of the compressor. Pressure data is essential for evaluating system health and calculating superheat and subcooling.

Sensor Type

Industrial refrigeration pressure transducer (Analog output: 0–5 V or 4–20 mA)

Pin Configuration (Proposed)

Sensor Pin	**ESP32 Pin**	**Description**
VCC	External supply	Sensor power
GND	GND	Ground
OUT	GPIO 34 (ADC)	Analog pressure output

(Note: Pressure sensor values were simulated in the current implementation)

Working Principle

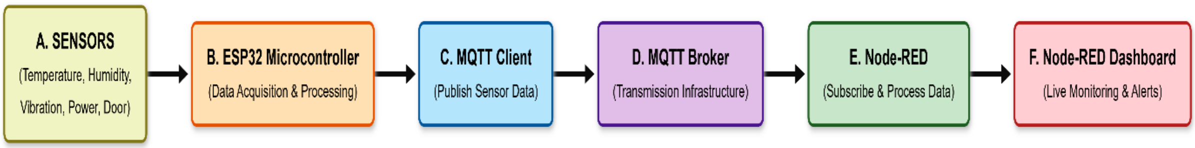
The pressure sensor converts refrigerant pressure into an electrical signal proportional to the applied pressure. The ESP32 reads this signal using its ADC and converts it into pressure values using calibration equations.

Role in the Proposed System

Pressure data is used for:

- Compressor health monitoring
- Detection of refrigerant leakage
- Calculation of superheat
    - Supporting intelligent EEV control

Hardware Architecture and Data Flow



Sensor Interface Layer

The sensing layer forms the physical input stage of the system. Multiple sensors are directly connected to the ESP32 microcontroller to monitor key freezer parameters such as temperature, humidity, vibration, power consumption, and door status.

Each sensor is powered using the ESP32 supply pins (3.3 V and GND) and connected through appropriate GPIO, ADC, or I²C interfaces depending on the sensor type. These sensors continuously sense environmental and operational conditions and provide raw electrical signals corresponding to the measured parameters.

This layer represents the direct hardware interaction with the freezer environment.

ESP32 Microcontroller (Edge Hardware Unit)

The ESP32 acts as the central hardware controller and edge processing unit of the system. All sensor outputs are physically wired to the ESP32, making it responsible for:

- Sensor signal acquisition through GPIO, ADC, and I²C pins
- Conversion of raw signals into usable digital values
- Basic preprocessing such as scaling, unit conversion, and validation

The ESP32 also provides built-in Wi-Fi capability, eliminating the need for an external communication module. This makes the hardware compact and suitable for embedded freezer applications.

MQTT Client (Embedded Communication Interface)

Within the ESP32 hardware, an MQTT client is implemented as part of the firmware. After sensor data is acquired and processed, the ESP32 packages the data into structured messages and publishes them to specific MQTT topics.

This stage represents the **hardware-to-network boundary**, where sensor data leaves the embedded device and enters the communication infrastructure.

MQTT Broker (Communication Backbone)

The MQTT broker functions as an intermediary communication node between the ESP32 hardware and the monitoring system. It does not perform hardware sensing or processing but ensures reliable message routing.

From a system architecture perspective, the broker decouples the embedded hardware (ESP32) from the visualization and monitoring platform, allowing scalable and asynchronous data transmission.

Node-RED (Processing and Control Interface)

Node-RED represents the backend control interface that receives data from the MQTT broker. It subscribes to the topics published by the ESP32 and processes the incoming data using flow-based logic.

Although Node-RED runs on a separate computing system (PC or server), it forms a logical extension of the hardware architecture by acting as the control and interpretation layer for the sensor data.

Node-RED Dashboard (Monitoring Output Interface)

The Node-RED Dashboard serves as the final output interface of the hardware system. It visually represents the real-time state of the freezer using gauges, indicators, and alert messages.

From a hardware flow perspective, this layer provides human–machine interaction (HMI), allowing users to observe the physical freezer conditions that were originally sensed by the hardware sensors.