

## AI Activity Log

### Why ChatGPT?

I chose ChatGPT because it's the tool I am most familiar with and had very stable results in previous projects including Arduino and circuit design

Note: I didn't add every prompt I used. I only added the ones that led me to the current state of the project because there are a lot of arbitrary questions.

Date: 5/8/25

Project Phase: Week 1: Sensor selection, basic temperature reading circuit

AI Tool Used: ChatGPT

Prompt Type: Design Help/ Code review

Original Prompt:

[https://www.youtube.com/watch?v=nssXWNYdGT8&ab\\_channel=HowToMechatronics](https://www.youtube.com/watch?v=nssXWNYdGT8&ab_channel=HowToMechatronics)  
see this video and make me a schematic that I can import into easyeda use the dht22  
version and modify his steps instead of using arduino mega i am going to use arduino uno

Quality: Excellent

What Worked: gave me exact code and wiring diagram

What Didn't Work: no issues

Follow-up Actions: wanted to try simulating the circuit in mind before going out to buy the components

Date: 5/8/25

Project Phase: Week 1: Sensor selection, basic temperature reading circuit

AI Tool Used: ChatGPT

Prompt Type: Question

Original Prompt: can i do simulation on easyeda using a temperature detection system with arduino and dht22

Quality: Excellent

What Worked: told me it doesn't work which was accurate

What Didn't Work: no issues

Follow-up Actions: I realized although accurate the answers were very general, so I started a new chat but this time I uploaded a picture of the project instruction

Date: 5/8/25

Project Phase: Week 1: Sensor selection, basic temperature reading circuit

AI Tool Used: ChatGPT

Prompt Type: Design Help

Original Prompt:

**Project 1: CoolTrack Pro - Smart Temperature Monitor**

**Project Overview**

Build a dual-sensor temperature logging system specifically designed for refrigerator diagnostics. The device monitors both freezer and refrigerator compartments, logs temperature data over time, and helps identify cooling system problems.

**Why This Project**

- Directly useful for your father's service business
- Teaches fundamental sensor interfacing and data logging
- Clear success criteria with immediate practical value
- Perfect introduction to diagnostic equipment design

**Learning Objectives**

- Temperature sensor calibration and interfacing
- Data logging techniques and storage methods
- Understanding refrigeration system operation
- Basic diagnostic equipment design principles
- PCB design and assembly using EASYEDA

**Key Features to Implement**

- Dual temperature sensor inputs (freezer/refrigerator)
- LCD display showing current temperatures
- Min/max temperature recording
- Data export to CSV files for analysis
- Low battery indicator and power management
- Compact, portable design for field use

**Weekly Breakdown**

- Week 1: Sensor selection, basic temperature reading circuit
- Week 2: Display integration and user interface
- Week 3: Data logging and storage implementation
- Week 4: Enclosure design, testing, and documentation

help me design this project currently i am in week 1 i took inspiration from this video  
:[https://www.youtube.com/watch?v=RkxqKmEyWRI&t=55s&ab\\_channel=RetroBuiltGames](https://www.youtube.com/watch?v=RkxqKmEyWRI&t=55s&ab_channel=RetroBuiltGames)

Quality: Excellent

What Worked: It had better understanding of the objective and gave me personalized results. It enhanced the idea I already had by suggesting a more suitable sensor (DS18B20)

## Step-by-Step Tasks

### 1. Select Sensor Type:

Sensor	Pros	Cons	
DHT22	Easy to use, gives temp & humidity	Slower updates (~2 sec), bulkier	
DS18B20	Accurate, compact, can be waterproof	Only temp, needs pull-up resistor	

 Recommendation: Use DS18B20 ×2 for accurate dual-zone temperature monitoring.

What Didn't Work: no issues

Follow-up Actions: started on the schematic design

Date: 6/5/25

Project Phase:

AI Tool Used: Week 1: Sensor selection, basic temperature reading circuit

AI Tool Used: ChatGPT

Prompt Type: Design Help

Original Prompt:

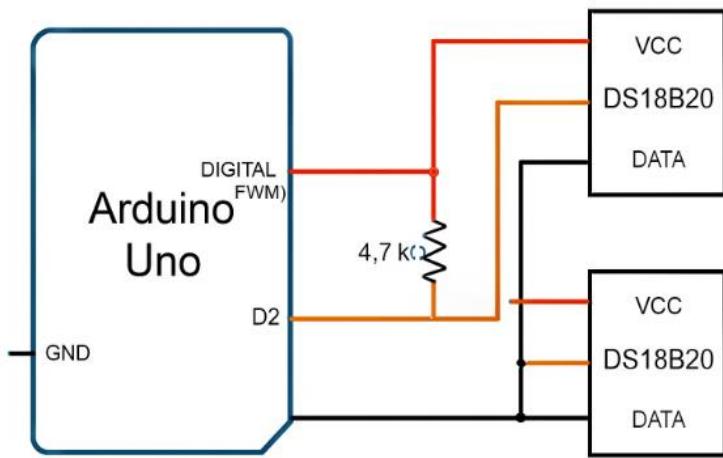
Let me know and I can draw the schematic or help you simulate it!



 proceed with DS18B20

Quality: Good

What Worked: it gave me a correct description to the connections even when prompted to generate the diagram it gave me a correct one



**What Didn't Work:** after further inspection I realized the design although right wasn't ideal for me because putting the 2 sensors on the same data line might cause future difficulties while trouble shooting

**Follow-up Actions:** did the schematic as suggested in EasyEDA (at that point I was still convinced with the design)

Date: 6/8/25

AI Tool Used: Week 1: Sensor selection, basic temperature reading circuit

AI Tool Used: ChatGPT

Prompt Type: Question/ Design help

Original Prompt: in the code why did you decide to use 1 digital pin for both sensors

Quality: Excellent

What Worked: it gave me logical reasons why it chose to do so

---

### Advantages of One Data Pin (1-Wire bus)

- Saves Arduino pins — you'll free up D3 for other features (like an LCD or buzzer later).
  - Less wiring — both sensors share the same VCC, GND, and Data.
  - Scales easily — you can add a 3rd, 4th, or more sensors without using more pins.
  - Library support is built for this — DallasTemperature works perfectly with multiple sensors on one line.
  - Only one pull-up resistor needed — cleaner circuit.
- 

### Downsides (but minor)

- If the data wire fails, both sensors stop working (instead of just one).
  - Slightly more code complexity if you want to identify each sensor by unique address (which we've already set up).
- 

### For your case:

Since this is a **fridge/freezer monitor** and we already have the **unique address code**, the one-pin approach makes a lot of sense.

It will simplify your EasyEDA schematic and breadboard wiring, and the Arduino will handle reading both sensors just fine.

What Didn't Work: no issues

Follow-up Actions: even though it had a lot of convincing arguments I chose to keep each sensor on its own data pin for now (I might change it later)

Date: 8/8/25

AI Tool Used: Week 1: Sensor selection, basic temperature reading circuit

AI Tool Used: ChatGPT

Prompt Type: Code Review/Design Help

Original Prompt: yes, I connected the wires and downloaded the libraries no walk me through the process

Code generated:

```
#include <OneWire.h>
#include <DallasTemperature.h>
```

```
// Assign pins for each sensor
```

```
#define FRIDGE_PIN 2
```

```
#define FREEZER_PIN 3
```

```
// Create OneWire instances for each pin
```

```
OneWire oneWireFridge(FRIDGE_PIN);
```

```
OneWire oneWireFreezer(FREEZER_PIN);
```

```
// Pass the OneWire references to DallasTemperature
```

```
DallasTemperature fridgeSensor(&oneWireFridge);
```

```
DallasTemperature freezerSensor(&oneWireFreezer);

void setup() {
  Serial.begin(9600);

  // Start both sensors
  fridgeSensor.begin();
  freezerSensor.begin();
}

void loop() {
  // Request temperature readings
  fridgeSensor.requestTemperatures();
  freezerSensor.requestTemperatures();

  // Read temperatures in Celsius
  float tempFridge = fridgeSensor.getTempCByIndex(0);
  float tempFreezer = freezerSensor.getTempCByIndex(0);

  // Display in Serial Monitor
  Serial.print("Fridge Temp: ");
  Serial.print(tempFridge);
  Serial.println(" °C");

  Serial.print("Freezer Temp: ");
  Serial.print(tempFreezer);
  Serial.println(" °C");

  Serial.println("-----");
  delay(2000);
}
```

---

### Step 3 – Upload and Test

1. Connect Arduino to your laptop with the USB cable.
2. In Arduino IDE:
  - Select Tools → Board → Arduino Uno (or Nano if you're using that).
  - Select Tools → Port and pick the one for your Arduino.
3. Paste the code above into a new sketch.
4. Click Upload.
5. Open Serial Monitor (Tools → Serial Monitor) and set baud rate to 9600.

You should see something like:

```
markdown                                ⌂ Copy ⌂ Edit  
Fridge Temp: 4.25 °C  
Freezer Temp: -17.00 °C
```

---

### Step 4 – Troubleshooting if needed

- If you see -127.00 °C → the Arduino can't detect the sensor → check wiring, pull-up resistor, and pin number.
- If you see random numbers → likely the resistor isn't connected correctly between data pin and 5V.

Quality: Excellent

What Worked: gave me exact instructions and it worked

What Didn't Work: no issues

Follow-up Actions: no follow up actions

Date: 9/8/25

Project Phase: Week 1: Sensor selection, basic temperature reading circuit

AI Tool Used: ChatGPT

Prompt Type: Design Help

Original Prompt: everything works fine now i need to do manual show me the usual format

Quality: Excellent

What Worked: gave me the exact format i was looking for

Answer:

## 典型项目手册格式

### 1. 封面页

- 项目标题
- 你的名字
- 日期
- 机构 / 课程信息
- 教师姓名（如果适用）

## **2. Table of Contents**

- Auto-generated if using Word / Google Docs

## **3. Introduction**

- Purpose of the project
- Brief background (why it's needed)
- Project goals and scope

## **4. System Overview**

- Short explanation of how the system works
- Block diagram of the system

## **5. Components & Materials**

- List of all parts used (Arduino, DS18B20, breadboard, resistor, etc.)
- Table with part names, quantities, specifications, images

## **6. Circuit Design**

- Schematic diagram (from EasyEDA)
- Explanation of the wiring (which pin connects where)
- Pull-up resistor explanation

## **7. Software**

- Arduino code listing
- Explanation of how the code works (main functions, pin assignments)
- Required libraries and how to install them

## **8. Assembly Instructions**

- Step-by-step wiring instructions (with photos or diagrams)
- Notes on sensor placement (fridge vs freezer)
- Testing procedure

## **9. Operation**

- How to power the system
- How to read the output on the Serial Monitor (and later on LCD)
- Normal temperature ranges for fridge/freezer

## **10. Troubleshooting**

- Common issues and fixes (e.g., “-127°C” = bad connection)
- How to reset the Arduino

## **11. Conclusion**

- Summary of what the project achieves
- Possible future improvements (LCD, alarms, logging, etc.)

## **12. References**

- Links to datasheets (DS18B20, Arduino board)
- Tutorials or videos used as inspiration

What Didn't Work: no issues

Follow-up Actions: creating logo

Date: 9/8/25

Project Phase: Week 1: Sensor selection, basic temperature reading circuit

AI Tool Used: ChatGPT

Prompt Type: Design Help

Original Prompt: CoolTrack Pro - Smart Temperature Monitor create a logo suitable for the manual

Quality: good

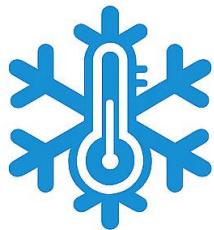
What Worked: gave me a good logo

Date



What Didn't Work: background was the wrong color due to me failing to give it enough information about the format

Follow-up Actions: corrected the color



# CoolTrack Pro

Smart Temperature Monitor

Date: 9/8/25

Project Phase: Week 1: Sensor selection, basic temperature reading circuit

AI Tool Used: ChatGPT

Prompt Type: Design Help

Original Prompt: make me a sensor selection grid dht11 sensor dht22 and DS18B20 with options and pros and cons/ give me a point wise description of the schematic we made

Quality: good

What Worked: gave me great description to everything in a short form

What Didn't Work: again, some transferring issues due to the wrong format and me failing to give it a more accurate prompt

Follow-up Actions: included everything in the Manual (most of the AI generated answers can be seen there)

Date: 14/8/25

Project Phase: Week 2: Display integration and user interface

AI Tool Used: ChatGPT

Prompt Type: Help

Original Prompt: we are in week 2 I have the lcd and i2c alter the code and tell me how to display the data on the lcd

Quality: Excellent

What worked: since I have uploaded all the past work it knew exactly what to do gave me code and wiring right away

What didn't work: nothing

Follow-up Actions: made sure the given code worked with my hardware after the AI advised me to

### 3 Important Notes

- **I<sup>2</sup>C Address:**
  - Most modules use `0x27`, some use `0x3F`.
  - If the LCD doesn't display anything, run an **I<sup>2</sup>C scanner** sketch to find the correct address.

Date: 14/8/25

Project Phase: Week 2: Display integration and user interface

AI Tool Used: ChatGPT

Prompt Type: Code review

Original Prompt: it's a generic 16x2 lcd with normal i2c

Quality: Excellent

What worked: after specifying my lcd and i2c type it confirmed past generated code was correct

What didn't work: nothing

Follow-up Actions: ran the code and did the wiring as described and it worked right away

16/8/25

Project Phase: Week 2: Display integration and user interface

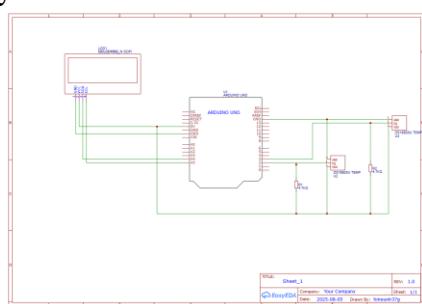
AI Tool Used: ChatGPT

Prompt Type: Design Help

Original Prompt: we agreed on the code we will use now make a report for me for week 2 in this same format as this report

<https://1drv.ms/w/c/41124573f611a022/Eep9RcTMZPlPjrc4CUG07YkB7Yijk7TVmVz8>

iGz9n\_7mqA?e=ZauK2d as an addition to week one here is the schematic for week 2 that you can add too



Quality: poor

What worked: the objective description was great

What didn't work: the format was all wrong he converted every section (Assembly, Schematic ... etc.) into paragraphs and was totally different than the layout AI use in my manual

Follow up: decided to do that part without AI

Date: 18/8/25

Project Phase: Week 3: Data logging and storage implementation

AI Tool Used: ChatGPT

Prompt Type: Design Help

Original Prompt: let's jump to week 3 i got the sd card arduino module help me complete the task

AI Response Quality: Excellent

What Worked: Gave me the right code and wiring right away even explained how it worked

What Didn't Work: Nothing

Follow up action: connected everything and it worked right away

Date: 18/8/25

Project Phase: Week 3: Data logging and storage implementation

AI Tool Used: ChatGPT

Prompt Type: Design Help

Original Prompt: add time tracking too so i we can turn data into a graph

AI Response Quality: Excellent

What Worked: Gave me the right code and wiring right away even explained how it worked and gave me a few RTC options

What Didn't work: Nothing

Follow up action: considered SD card size next

Date: 18/8/25

Project Phase: Week 3: Data logging and storage implementation

AI Tool Used: ChatGPT

Prompt Type: Design Help

Original Prompt: is an 8 GB SD card enough

Ai Response quality: Excellent

What worked: answered the question clearly even gave me an approximation in days in which the SD card can store data before it gets full

What didn't work: nothing

Follow up action went and bought the components applied the design and it worked fine

Date: 18/8/25

Project Phase: Week 3: Data logging and storage implementation

AI Tool Used: ChatGPT

Prompt Type: Design Help

Original Prompt: i need you to do the schematic description like you did for the past schematics explain also what miso and mosi sck cs sda and scl does (Attached Manual)

AI response Quality: good

What worked: since I uploaded the manual it had a clear reference this time and generated the correct info, I asked it to generate

What didn't work: format isn't always like the one in the manual have to do small fixes especially after copying and pasting

Follow up action: added everything into manual

Date: 18/8/25

Project Phase: Week 3: Data logging and storage implementation

AI Tool Used: ChatGPT

Prompt Type: Design Help

Original Prompt: can you adjust the assembly and execution part on the report i uploaded to week 3 of the project with this code #include <Wire.h> #include

```
<LiquidCrystal_I2C.h> #include <OneWire.h> #include <DallasTemperature.h>
#include <RTClib.h> #include <SD.h> // ===== Pin Assignments
===== #define FRIDGE_PIN 2 // DS18B20 (Fridge) #define
FREEZER_PIN 3 // DS18B20 (Freezer) #define SD_CS 10 // Chip Select for SD card //
===== Sensor Objects ===== OneWire
oneWireFridge(FRIDGE_PIN); OneWire oneWireFreezer(FREEZER_PIN);
DallasTemperature fridgeSensor(&oneWireFridge); DallasTemperature
freezerSensor(&oneWireFreezer); LiquidCrystal_I2C lcd(0x27, 16, 2); // LCD (I2C
address 0x27) RTC_DS3231 rtc; File logfile; // ===== Setup
===== void setup() { Serial.begin(9600); // Start temperature sensors
fridgeSensor.begin(); freezerSensor.begin(); // Start LCD lcd.init(); lcd.backlight();
lcd.clear(); lcd.setCursor(0, 0); lcd.print("CoolTrack Pro"); delay(2000); lcd.clear(); //
Start RTC if (!rtc.begin()) { Serial.println("RTC not found!"); lcd.print("RTC Fail");
while (1); } if (rtc.lostPower()) { rtc.adjust(DateTime(F(__DATE__), F(__TIME__))); //
Set to compile time } // Start SD card if (!SD.begin(SD_CS)) { Serial.println("SD Card
init failed!"); lcd.print("SD Fail"); while (1); } Serial.println("SD Card Ready.");
lcd.print("SD Ready"); delay(1000); lcd.clear(); // Prepare log file header logfile =
SD.open("temp_log.csv", FILE_WRITE); if (logfile) {
logfile.println("Date,Time,Fridge_C,Freezer_C"); logfile.close(); } } //
===== Main Loop ===== void loop() { // ---- Time
---- DateTime now = rtc.now(); // ---- Temperatures ----
fridgeSensor.requestTemperatures(); freezerSensor.requestTemperatures(); float
tempFridge = fridgeSensor.getTempCByIndex(0); float tempFreezer =
freezerSensor.getTempCByIndex(0); // ---- Serial Monitor ----
```

```
Serial.print(now.timestamp(DateTime::TIMESTAMP_DATE)); Serial.print(" ");
Serial.print(now.timestamp(DateTime::TIMESTAMP_TIME)); Serial.print(" | Fridge: ");
Serial.print(tempFridge); Serial.print(" °C | Freezer: "); Serial.print(tempFreezer);
Serial.println(" °C"); // ----- LCD Output ----- lcd.setCursor(0, 0); lcd.print("Fridge:");
lcd.print(tempFridge, 1); lcd.print((char)223); lcd.print("C "); // padding to clear leftovers
lcd.setCursor(0, 1); lcd.print("Freezer:"); lcd.print(tempFreezer, 1); lcd.print((char)223);
lcd.print("C "); // ----- Log to SD ----- logFile = SD.open("temp_log.csv",
FILE_WRITE); if (logFile) { logFile.print(now.year()); logFile.print("-");
logFile.print(now.month()); logFile.print("-"); logFile.print(now.day()); logFile.print(",");
logFile.print(now.hour()); logFile.print(":"); logFile.print(now.minute());
logFile.print(":"); logFile.print(now.second()); logFile.print(",");
logFile.print(tempFridge, 1); logFile.print(","); logFile.println(tempFreezer, 1);
logFile.close(); Serial.println("Data logged."); } else { Serial.println("Error opening log
file!"); } delay(2000); // update every 2 sec }
```

AI response Quality: good

What worked: gave me the description in the same wording style of the past manual

What didn't work: wasn't the same text format had to rearrange everything

Follow up action: added everything into manual