Mini project documentation for **"Real-Time Temperature and Humidity Logger using Arduino"** including:

- Project Overview
- Block Diagram
- Circuit Diagram
- Components Required
- Arduino Code
- Data Logging Setup (using SD card or Serial Monitor)
- Optional: RTC integration
- Future Scope

Project Title:

Real-Time Temperature and Humidity Logger using Arduino UNO

Objective

To design a low-cost system that monitors and logs real-time temperature and humidity using Arduino and DHT11/DHT22 sensor, storing data on an SD card or displaying on a Serial Monitor.

Block Diagram

```
+------------+          +------------+          +------------+
|  DHT11 /   | -------> |  Arduino   | -------> | SD Card    |
|  DHT22     |          |  Uno/Nano  |          | Module     |
+------------+          +------------+          +------------+
                                    |
                                    v
                             +--------------+
                             | Serial Monitor|
                             +--------------+
```

Circuit Diagram Description

Components and Connections:

Component	Arduino Pin	Notes
DHT11 / DHT22	Digital Pin 2	Connect VCC and GND
SD Card Module	CS -> Pin 10	MOSI -> Pin 11, MISO -> 12

Component	Arduino Pin	Notes
	SCK -> Pin 13	Connect 3.3V or 5V per module
RTC Module *(Optional)*	SDA -> A4, SCL -> A5	For timestamping

---

Hardware Components

1. Arduino Uno/Nano
2. DHT11 or DHT22 Sensor
3. SD Card Module + Micro SD Card
4. RTC Module (DS3231) *(Optional)*
5. Jumper Wires
6. Breadboard
7. USB Cable

Arduino Code

Version 1: Logging to SD Card

```
#include <DHT.h>
#include <SPI.h>
#include <SD.h>

#define DHTPIN 2
#define DHTTYPE DHT11   // Change to DHT22 if using that

DHT dht(DHTPIN, DHTTYPE);
const int chipSelect = 10;

File dataFile;

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

  Serial.print("Initializing SD card...");
  if (!SD.begin(chipSelect)) {
    Serial.println("Card failed or not present.");
    return;
  }
  Serial.println("Card initialized.");

  // Create file if not exists
  dataFile = SD.open("datalog.csv", FILE_WRITE);
  if (dataFile) {
    dataFile.println("Temperature (°C),Humidity (%)");
    dataFile.close();
  }
}

void loop() {
  float temp = dht.readTemperature();
  float hum = dht.readHumidity();

  if (isnan(temp) || isnan(hum)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  String dataString = String(temp) + "," + String(hum);

  // Print to Serial
  Serial.println(dataString);

  // Log to SD card
  dataFile = SD.open("datalog.csv", FILE_WRITE);
  if (dataFile) {
    dataFile.println(dataString);
    dataFile.close();
  } else {
    Serial.println("Error opening datalog.csv");
  }

  delay(5000);  // Log every 5 seconds
}
```

Optional: Add RTC for Timestamped Logging

```
cpp
CopyEdit
#include <RTClib.h>
RTC_DS3231 rtc;

void setup() {
  rtc.begin();
  if (rtc.lostPower()) {
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  }
}
```

Inside the `loop()`, use:

```
cpp
CopyEdit
DateTime now = rtc.now();
String timeStamp = String(now.year()) + "/" +
                   String(now.month()) + "/" +
                   String(now.day()) + " " +
                   String(now.hour()) + ":" +
                   String(now.minute()) + ":" +
                   String(now.second());

String dataString = timeStamp + "," + String(temp) + "," + String(hum);
```

Expected Output (CSV Log Format)

```
Temperature (°C),Humidity (%)
24.00, 50.00
24.10, 49.80
...
```

If timestamped:

```
Timestamp,Temperature (°C),Humidity (%)
2025/06/14 12:30:10,24.00,50.00
2025/06/14 12:30:15,24.10,49.80
```

Future Scope

- IoT Integration: Send data to cloud (e.g., via ESP8266)
- GUI Dashboard (Python, Processing)
- SMS/email alerts on threshold breach
- Battery-operated weather station

Project Folder Structure

```
TemperatureLogger/
│
├── Arduino_Code/
│   └── DHT_SDLogger.ino
│
├── circuit_diagram.png
├── README.md
└── sample_output.csv
```

circuit diagram and PCB layout for your **Real-Time Temperature and Humidity Logger using Arduino**, featuring the DHT11 sensor, SD card module, and optional DS3231 RTC module.

Circuit Diagram

Here's a Fritzing-based circuit diagram illustrating the connections:

Connections:

- **DHT11 Sensor:**
  - VCC → 5V (Arduino)
  - GND → GND (Arduino)
  - DATA → Pin 2 (Arduino)
- **SD Card Module:**
  - VCC → 5V (Arduino)
  - GND → GND (Arduino)
  - CS → Pin 10 (Arduino)
  - MOSI → Pin 11 (Arduino)
  - MISO → Pin 12 (Arduino)
  - SCK → Pin 13 (Arduino)

- **RTC Module (Optional):**
  - VCC → 5V (Arduino)
  - GND → GND (Arduino)
  - SDA → A4 (Arduino)
  - SCL → A5 (Arduino)

Note: Ensure your SD card is formatted to FAT32 before use.

PCB Layout

For a more compact and professional setup, you can design a custom PCB. Here's a sample PCB layout for the project:

Design Notes:

- **Software Used:** EasyEDA (browser-based PCB design tool)
- **Components Included:**
  - Arduino Uno footprint
  - DHT11 sensor
  - SD card module
  - DS3231 RTC module (optional)
- **Features:**
  - Compact design for easy enclosure
  - Clear labeling for components and connections
  - Power and ground planes for stable operation