# Arduino UNO R4 Minima with TF Card Reader Module and TF Card for Data Logging

## Description

In this project, we will learn how to interface an Arduino Uno R4 Minima with a TF (MicroSD) card reader module to perform data logging. We'll utilize a TF card to store data recorded by sensors connected to the Arduino. This project is particularly useful for applications such as environmental monitoring, data acquisition, and logging sensor data for analysis.

## How-To Guide

### 1. Gather Components

* 1 x Arduino Uno R4 Minima
* 1 x Breadboard
* 1 x TF Card Reader Module (MicroSD Card Module)
* 1 x Raindrop Sensor
* 1 x TF Card (MicroSD Card，additional purchase required)
* 15 x Jumper wires

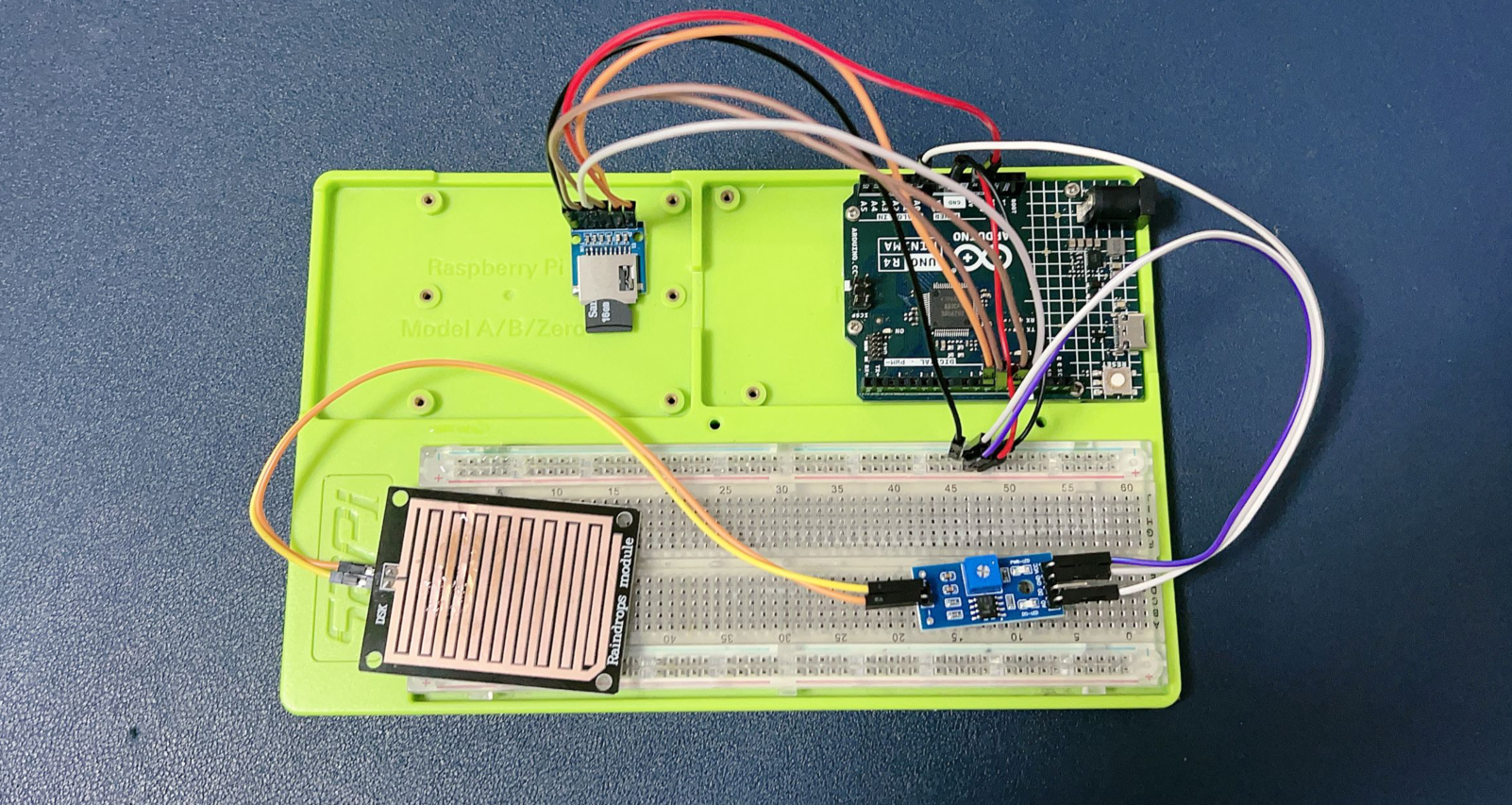
### 2. Assemble the Circuit

* Connect the TF card reader module to the Arduino Uno R4 Minima using jumper wires.
* Insert the TF card into the TF card reader module.
* Ensure proper connections between the TF card reader module and the Arduino Uno R4 Minima.

#### Wiring Diagram

|  |  |
| --- | --- |
| Arduino UNO R4 Minima | TF Card Reader Module |
| 3.3V | 3V3 |
| D10 | CS |
| D11 | MOSI |
| D13 | CLK |
| D12 | MISO |
| GND | GND |

|  |  |
| --- | --- |
| Arduino UNO R4 Minima | Rain Drop Module |
| 5V | VCC |
| GND | GND |
| NC-Not Connect | DO |
| A0 | AO |



### 3. Write the Arduino Code

Open the Arduino IDE on your computer and create a new sketch, It is a simple code to read and write the TF card.

#include <SD.h>

// Define chip select pin for the TF card reader module

const int chipSelectPin = 10;

void setup() {

    Serial.begin(9600);  // Initialize serial communication

    pinMode(chipSelectPin, OUTPUT);  // Set chip select pin as output

    // Initialize TF card

    if (!SD.begin(chipSelectPin)) {

        Serial.println("TF card initialization failed!");

        return;

    }

    Serial.println("TF card initialized.");

}

void loop() {

    // Record data to TF card

    File dataFile = SD.open("data.txt", FILE\_WRITE);

    if (dataFile) {

        dataFile.println("Hello, TF card!");  // Write data to TF card

        dataFile.close();  // Close the file

        Serial.println("Data recorded to TF card.");

    } else {

        Serial.println("Error opening data.txt");

    }

    delay(1000);  // Delay for stability

    // Read data from TF card

    dataFile = SD.open("data.txt");

    if (dataFile) {

        Serial.println("Reading data.txt:");

        while (dataFile.available()) {

            Serial.write(dataFile.read());  // Read data from TF card and print to serial monitor

        }

        dataFile.close();  // Close the file

    } else {

        Serial.println("Error opening data.txt");

    }

    delay(5000);  // Delay for stability

}

### 4. Understand the Code

The code initializes the SD library for TF card communication and sets the chip select pin for the TF card reader module.

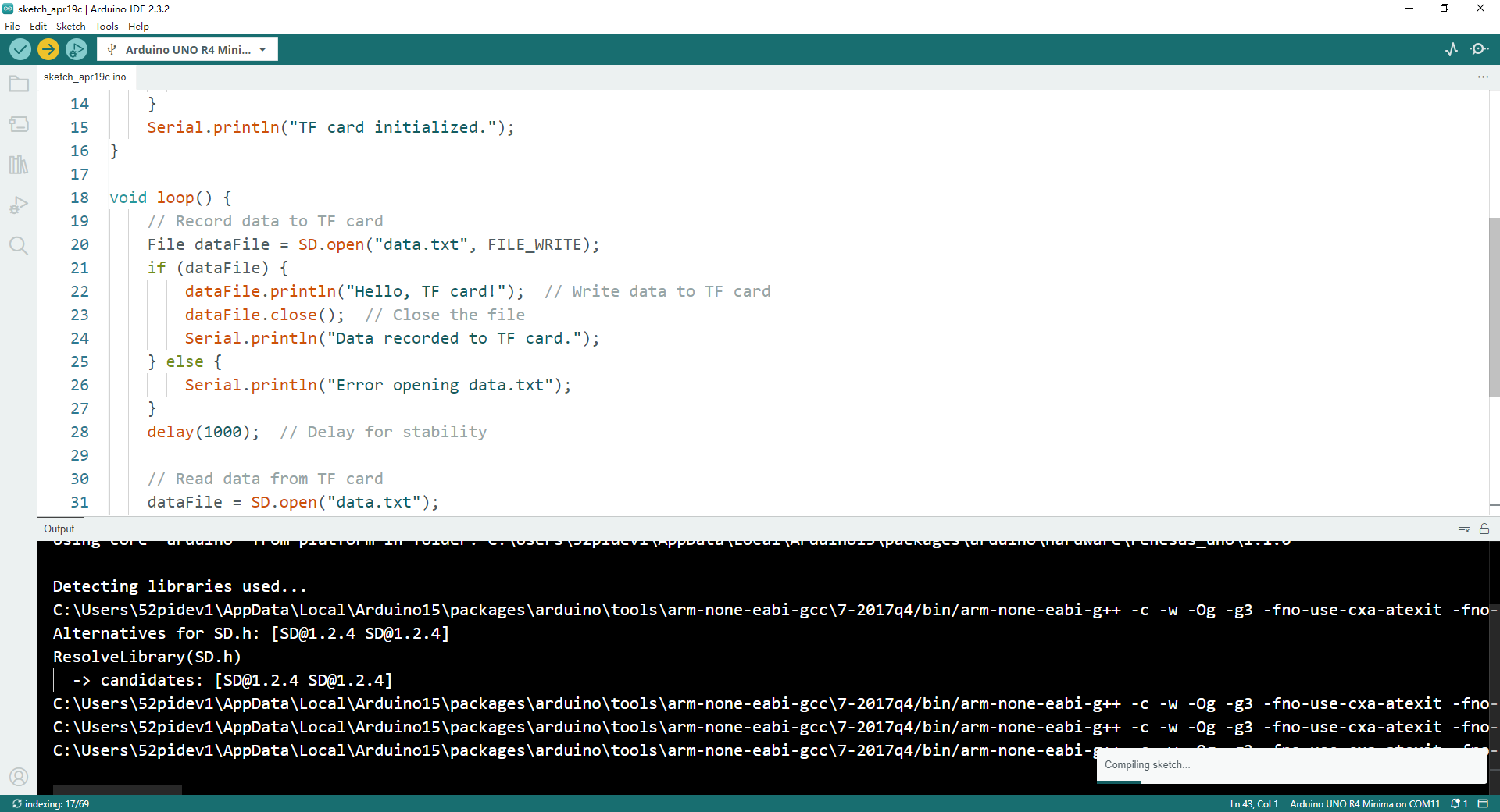
In the **setup()** function, it initializes serial communication and the TF card. If the TF card initialization fails, an error message is printed.

In the **loop()**, it records data to the TF card and reads data from the TF card.

Data is recorded to the file "data.txt" on the TF card, and then the file is read and printed to the serial monitor.

### 5. Test the Project

Upload the code to your Arduino Uno R4 Minima.



Open the serial monitor to observe the data recording and reading processes.



### 6. Experiment and Learn

Modify the code to record different sensor data to the TF card.

Implement timestamps for recorded data to track the time of data acquisition.

Explore error-checking mechanisms and data formatting techniques for robust data logging.

By following these steps, you can successfully interface a TF card reader module with the Arduino Uno R4 Minima to perform data logging tasks, enabling storage and retrieval of sensor data on a TF card for various applications.

### 7. Make a rain-drop data logger

Open the Arduino IDE and create a new sketch. Then, use the following code:

#include <SD.h>

// Define chip select pin for the TF card reader module

const int chipSelectPin = 10;

const int raindropSensorPin = A0;

void setup() {

    Serial.begin(9600);  // Initialize serial communication

    pinMode(chipSelectPin, OUTPUT);  // Set chip select pin as output

    // Initialize TF card

    if (!SD.begin(chipSelectPin)) {

        Serial.println("TF card initialization failed!");

        return;

    }

    Serial.println("TF card initialized.");

}

void loop() {

    // Read data from raindrop sensor

    int raindropValue = analogRead(raindropSensorPin);

    // Record data to TF card

    File dataFile = SD.open("newdata.txt", FILE\_WRITE);

    if (dataFile) {

        dataFile.print(millis());  // Record current time

        dataFile.print(", ");

        dataFile.println(raindropValue);  // Record raindrop sensor value

        dataFile.close();  // Close the file

        Serial.println("Data recorded to TF card.");

    } else {

        Serial.println("Error opening newdata.txt");

    }

    delay(1000);  // Delay for stability

}

### 8. Understand the Code

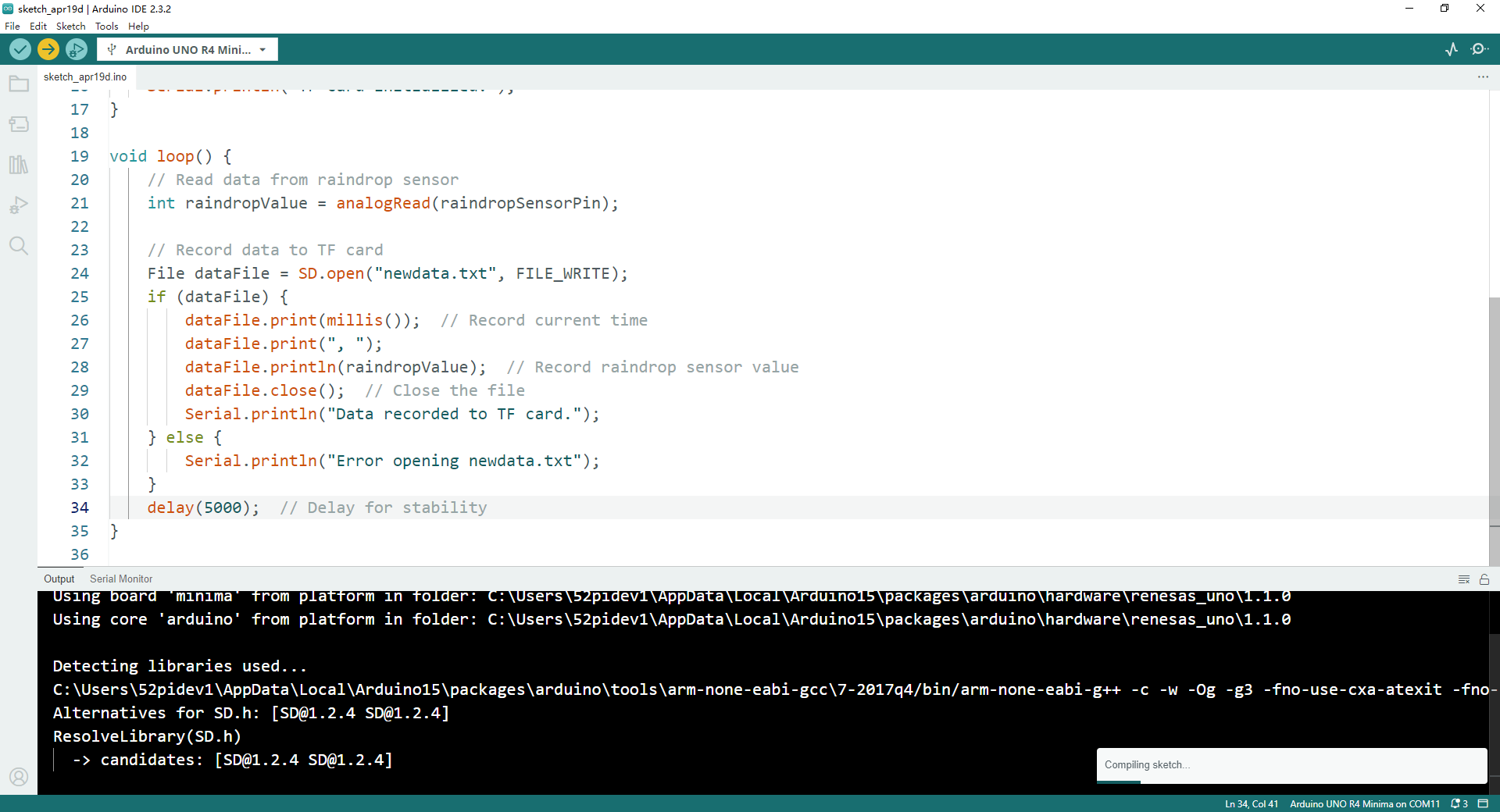
This code is similar to the previous project but reads data from the raindrop sensor instead of writing a fixed message.

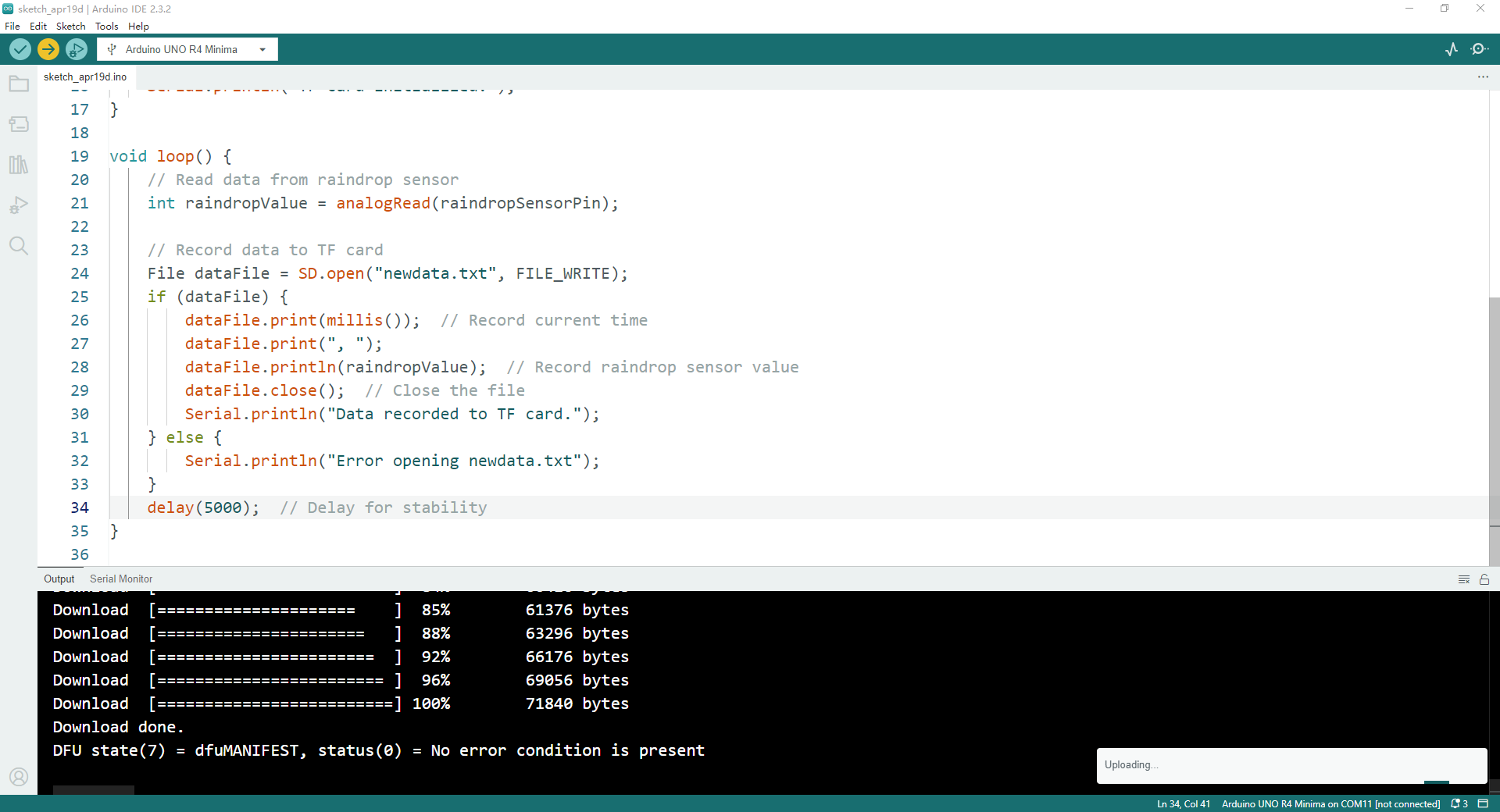
The raindrop sensor value is read using analogRead and stored along with the current time in milliseconds.

Data is logged to a file named "newdata.txt" on the TF card.

### 10. Test the Project

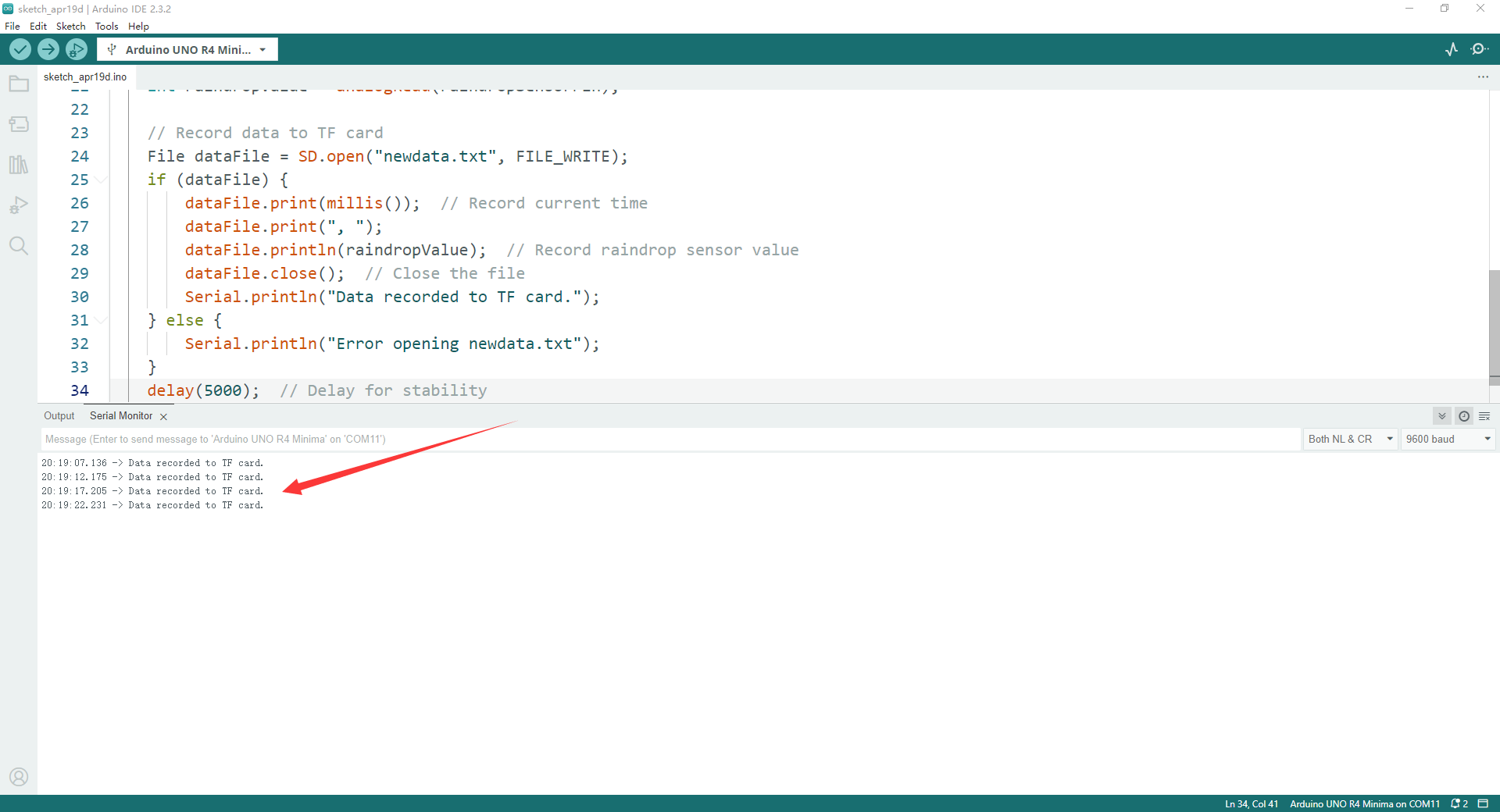
Upload the code to your Arduino Uno R4 Minima.





Ensure that the TF card is inserted into the TF card reader module.

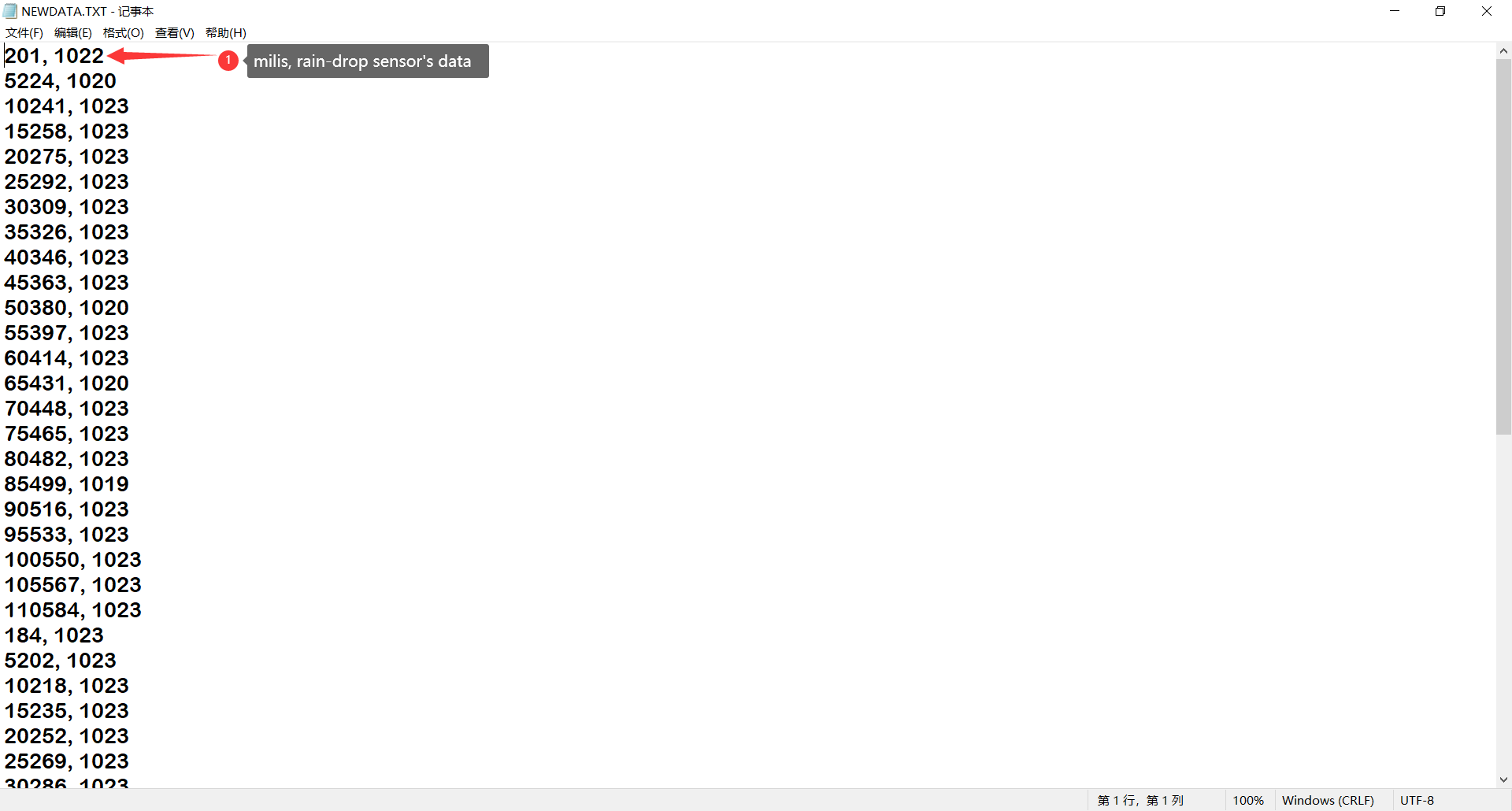
Open the serial monitor to observe the data logging process.



After that, unplug the USB-C cable and remove the TF card from the TF card reader, and insert it back to your Computer with card reader, and open the file manager：



Open it.



### 11. Experiment and Learn

Modify the code to include additional sensor data or timestamps for more comprehensive logging.

Implement error-checking mechanisms to handle cases where the TF card is not initialized or the file cannot be opened.

Explore ways to optimize data storage and retrieval for efficient data logging.

By following these steps, you can create a raindrop data logger using the Arduino Uno R4 Minima and a TF card reader module, enabling you to record and analyze raindrop sensor data over time.