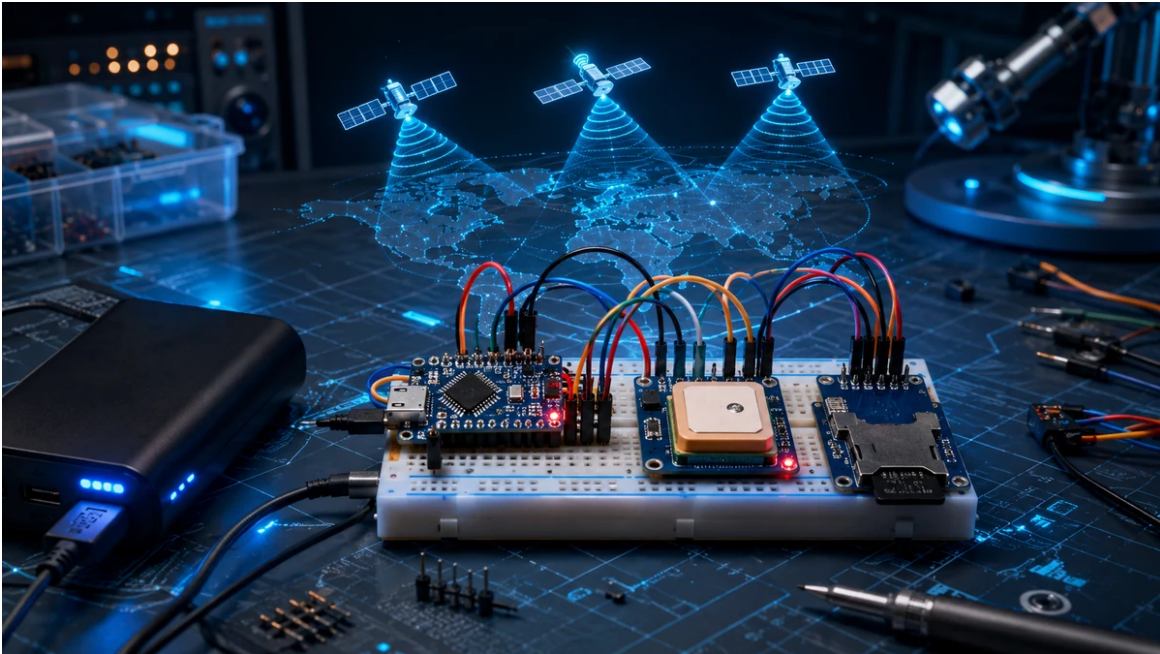


Arduino GPS Tracker Tutorial

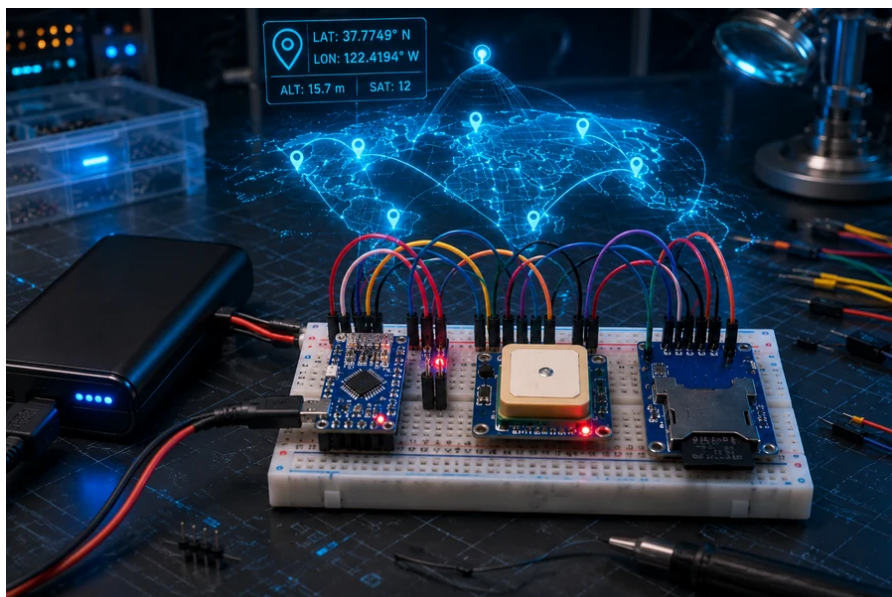
Serial GPS Data, Location Logging, and Portable Power



Build a portable Arduino GPS tracker that reads satellite data, logs latitude and longitude to a MicroSD card, and runs from battery power. This guide speaks directly to the reader: you will wire the modules, load the code, test GPS lock outdoors, and open the finished CSV log on your computer.

What You Will Build

You are building a compact GPS data logger. When powered on, the GPS module listens for satellite signals. The Arduino reads the GPS serial stream, extracts useful location data, and saves it as a CSV file on the MicroSD card.

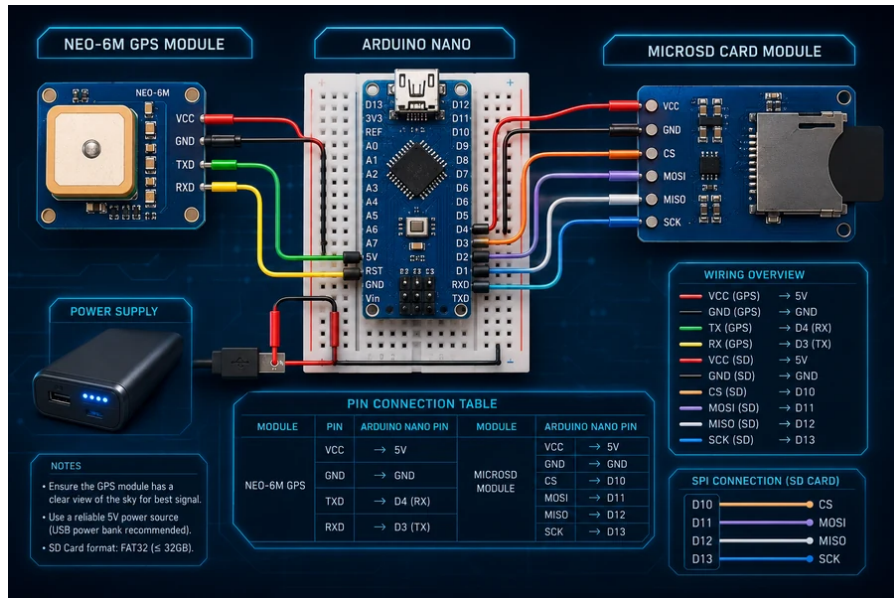


Parts List

Recommended parts: Arduino Nano or Uno, NEO-6M GPS module, MicroSD card module, 8GB to 32GB MicroSD card formatted as FAT32, jumper wires, breadboard or proto board, USB power bank, switch, and enclosure.

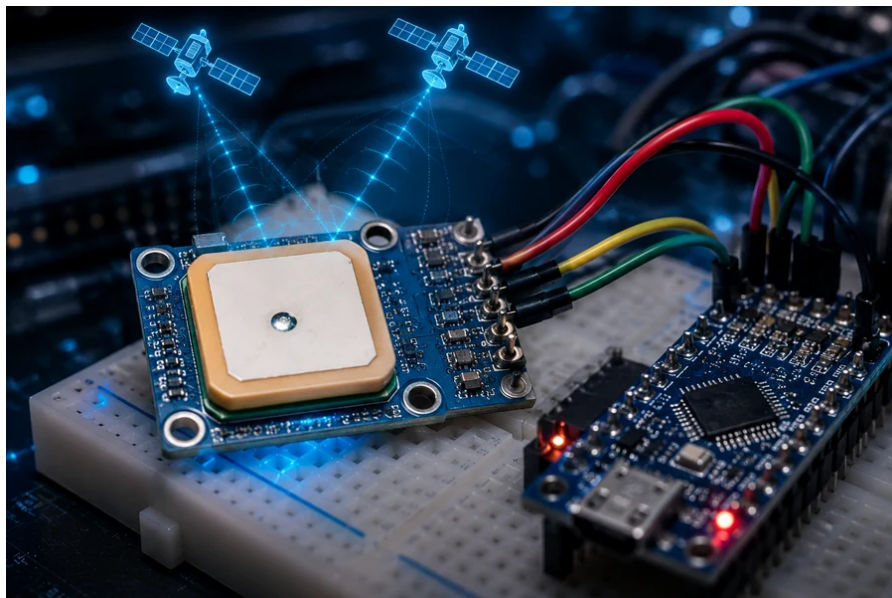
Wiring Overview

Connect GPS VCC to 5V, GPS GND to GND, GPS TX to D4, GPS RX to D3. Connect MicroSD VCC to 5V, GND to GND, CS to D10, MOSI to D11, MISO to D12, and SCK to D13.



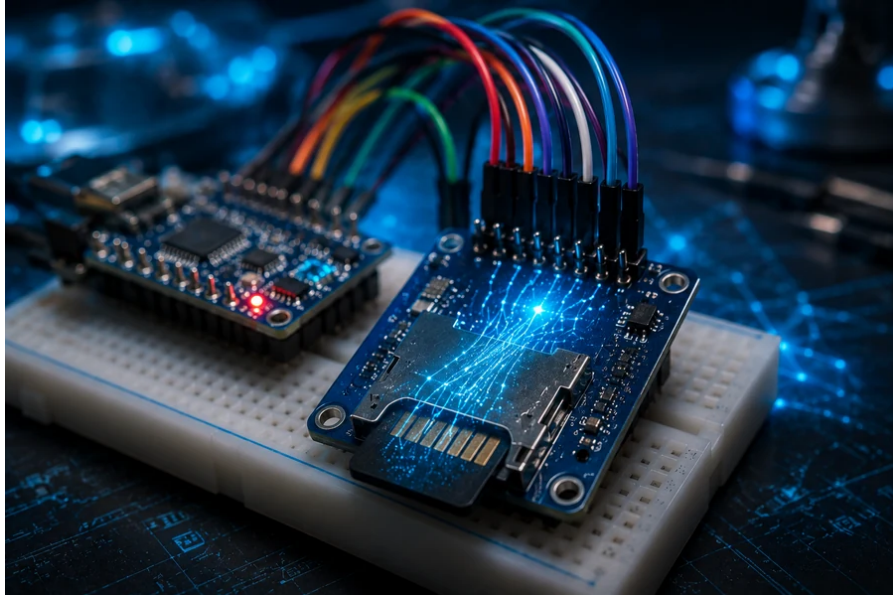
GPS Module Close-Up

The NEO-6M module sends serial NMEA GPS data. For best results, place the ceramic antenna face-up with a clear view of the sky.



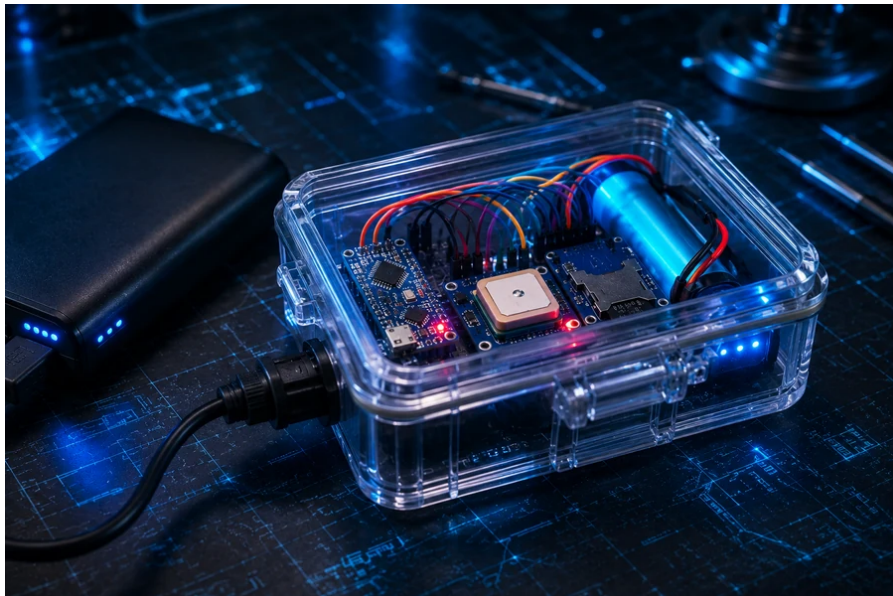
MicroSD Logger

The MicroSD module saves your route into gpslog.csv. Open the file later in a spreadsheet to review coordinates, altitude, speed, and satellite count.



Portable Power

Start with a USB power bank for stable 5V power. If you move to lithium batteries, use a protected charging module and do not wire raw cells directly into the project.



Outdoor Testing

GPS modules work best outdoors. A cold start may take several minutes. Wait for valid latitude and longitude values before trusting the log.



CSV Log Review

After testing, remove the MicroSD card and open gpslog.csv. Each row should contain date, time, latitude, longitude, altitude, speed, and satellite count.



Core Arduino Code

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include <SPI.h>
#include <SD.h>

TinyGPSPlus gps;
SoftwareSerial gpsSerial(4, 3); // GPS TX to D4, GPS RX to D3
const int chipSelect = 10;
File gpsFile;

void setup() {
  Serial.begin(9600);
  gpsSerial.begin(9600);
}
```

```

if (!SD.begin(chipSelect)) {
  Serial.println("SD card failed.");
  return;
}
gpsFile = SD.open("gpslog.csv", FILE_WRITE);
if (gpsFile) {
  gpsFile.println("Date,Time,Latitude,Longitude,Altitude_m,Speed_kmph,Satellites");
  gpsFile.close();
}
}

void loop() {
  while (gpsSerial.available() > 0) {
    gps.encode(gpsSerial.read());
    if (gps.location.isUpdated()) {
      gpsFile = SD.open("gpslog.csv", FILE_WRITE);
      if (gpsFile) {
        gpsFile.print(gps.location.lat(), 6);
        gpsFile.print(",");
        gpsFile.println(gps.location.lng(), 6);
        gpsFile.close();
      }
    }
  }
}
}

```

Fixed Video Section

The updated HTML uses these YouTube embed IDs: ffcg7lcqQ84 for NEO-6M setup, 9-v5hNuHFq4 for GPS logging to SD, and IEbFpwqCFJg for a complete Arduino GPS tracker walkthrough.

Troubleshooting

| Problem | Fix |
|--------------------|---|
| No coordinates | Test outdoors and check GPS TX to Arduino D4. |
| SD failed | Format FAT32 and check CS/MOSI/MISO/SCK wiring. |
| Random serial text | Set Serial Monitor to 9600 baud. |
| Invalid data | Wait for satellite lock. |
| Battery resets | Use a stable USB power bank. |