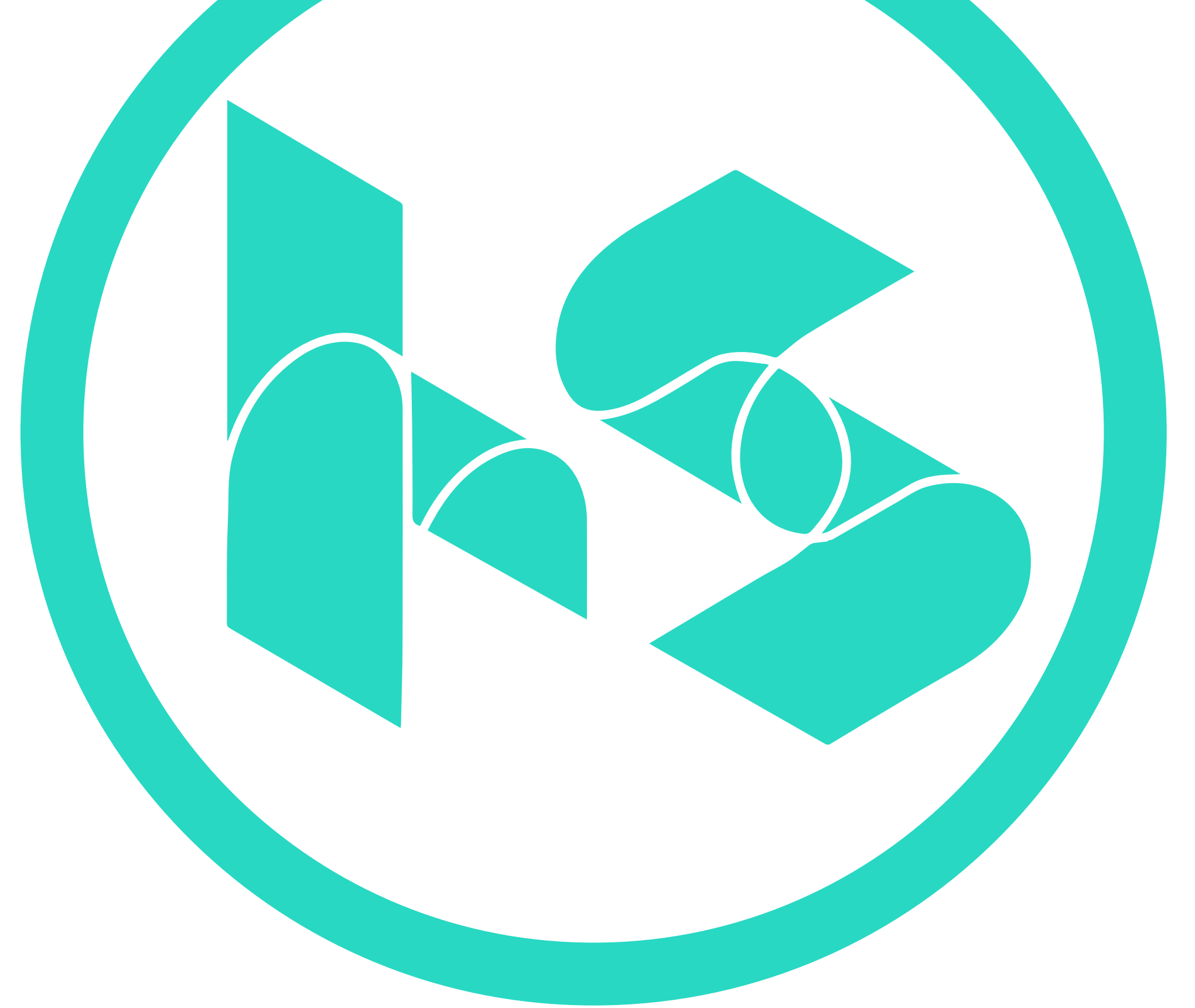
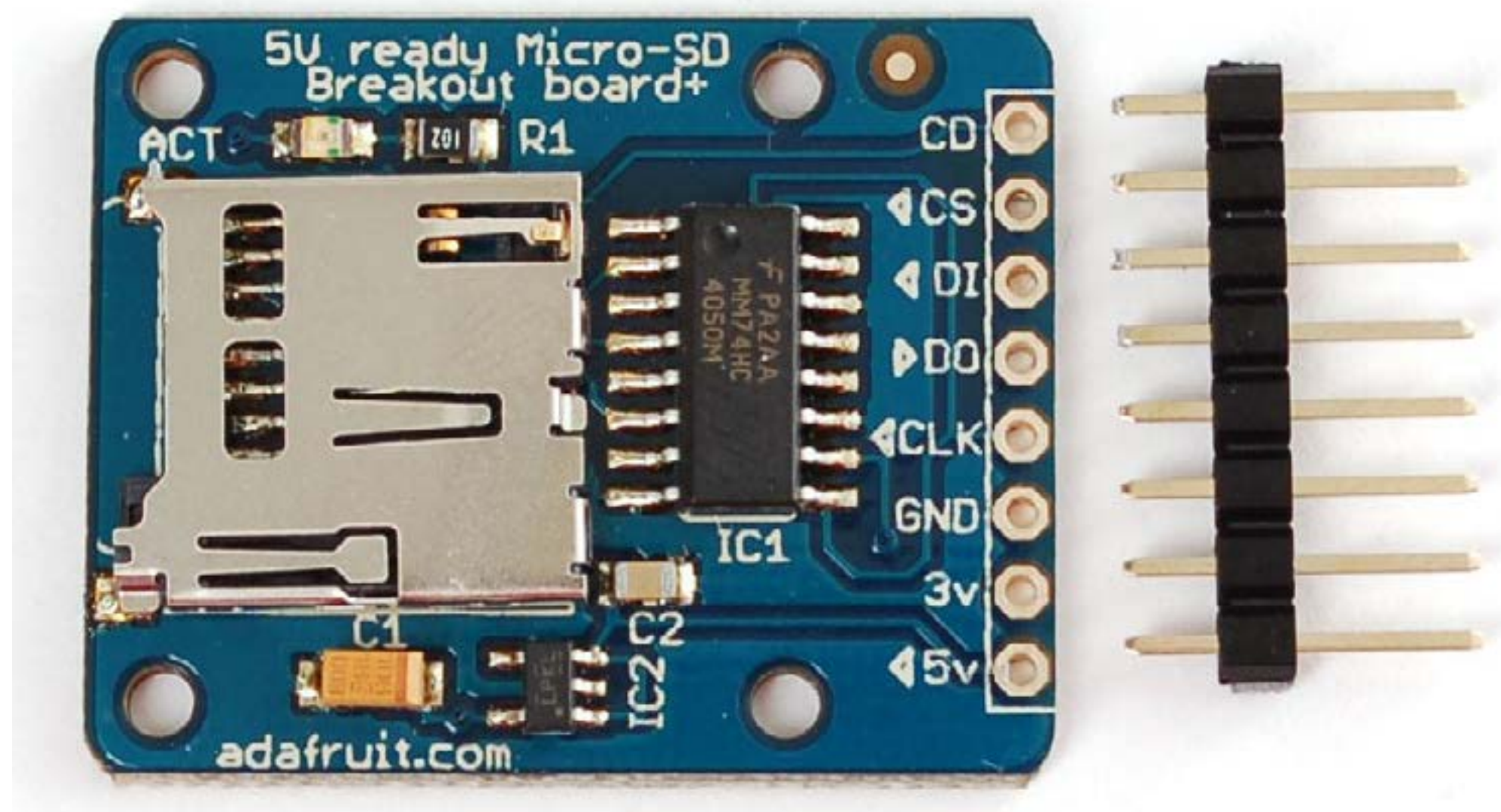


SDCARD MODULE



DOCUMENTATION

SD CARD MODULE

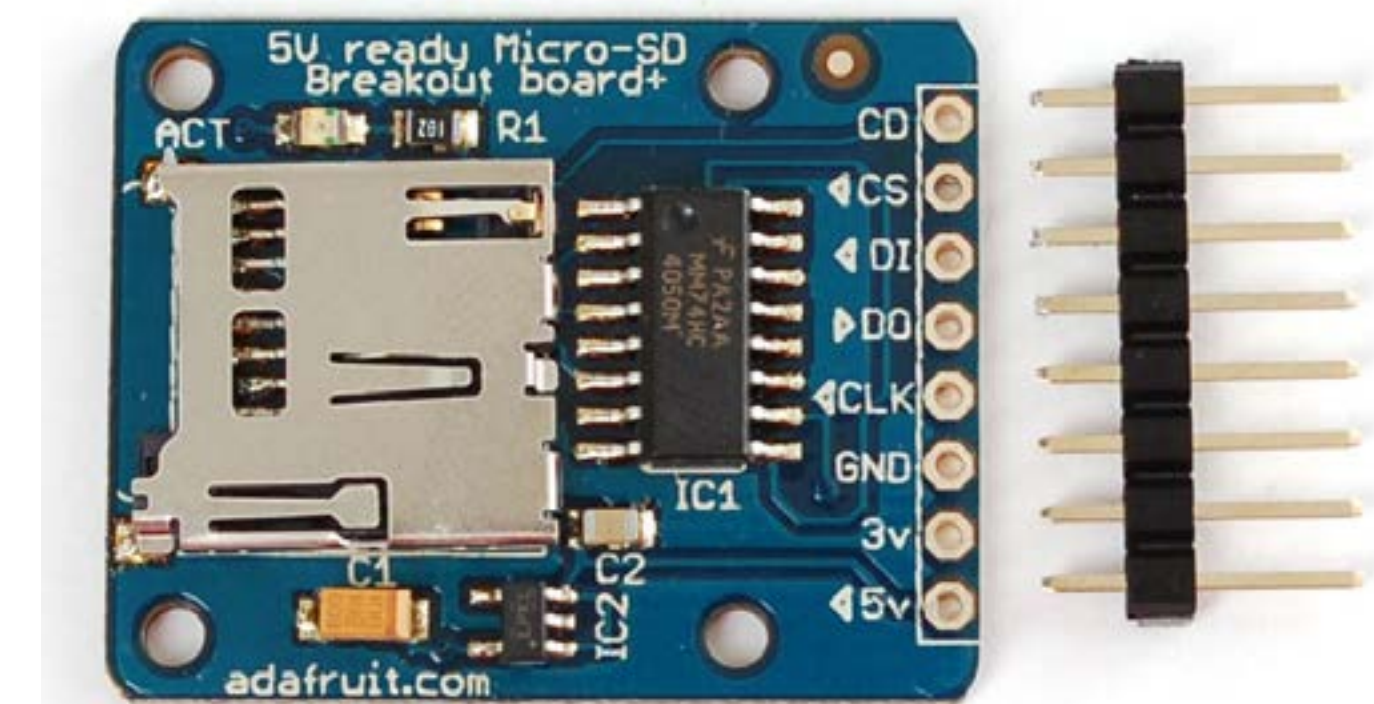
Most microcontrollers (such as the Arduino board) have extremely limited built-in storage. If your project involves any audio, video, graphics, data logging, etc., adding an SDCard module will be required.

The **SD card module** allows your microcontroller to communicate with an SD card via the SPI (Serial Peripheral Interface) protocol.

It can used to interact with a 3.3V or 5V microcontrollers.

MORE INFO:

[ADAFRUIT SD CARD MODULE TUTORIAL](#)

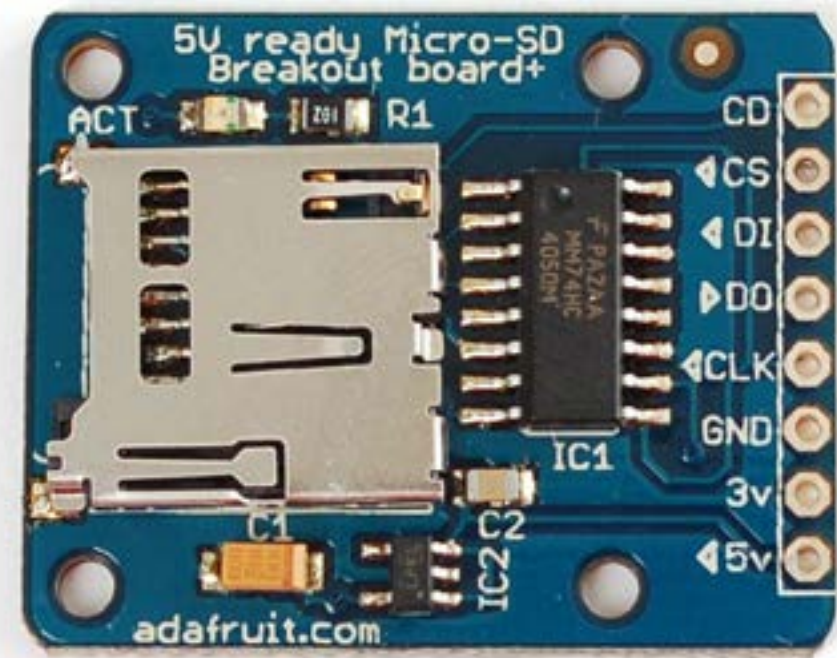


ACTIVITY

WRITING FILES TO AN SD CARD USING ARDUINO

WRITING FILES TO AN SD CARD USING ARDUINO

HARDWARE COMPONENTS



SD Card Module



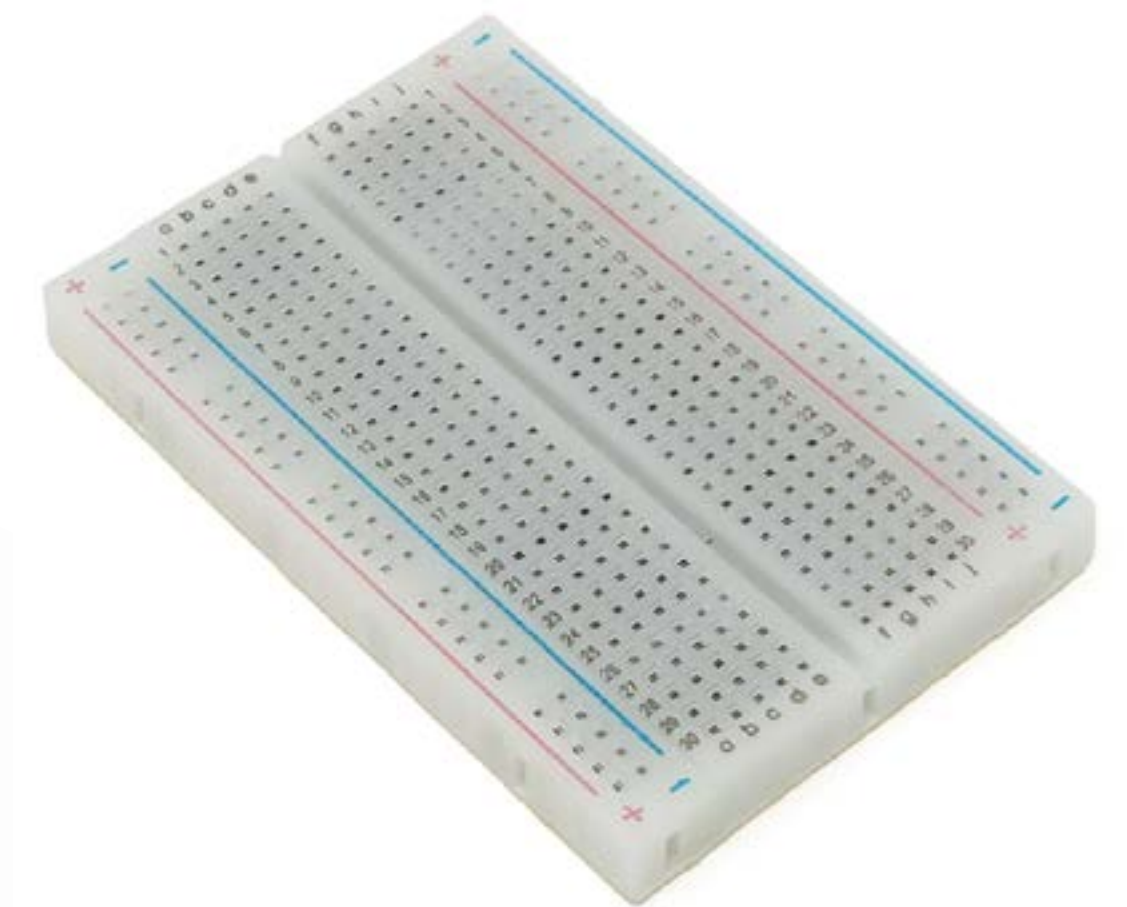
Micro SD Card



Arduino UNO
board



Jumpers



Breadboard

WRITING FILES TO AN SD CARD USING ARDUINO

WIRING

Connect the 5V pin to the 5V pin on the Arduino

Connect the GND pin to the GND pin on the Arduino

Connect CLK to pin 13 (pin 52 for Arduino MEGA)

Connect DO to pin 12 (pin 50 for Arduino MEGA)

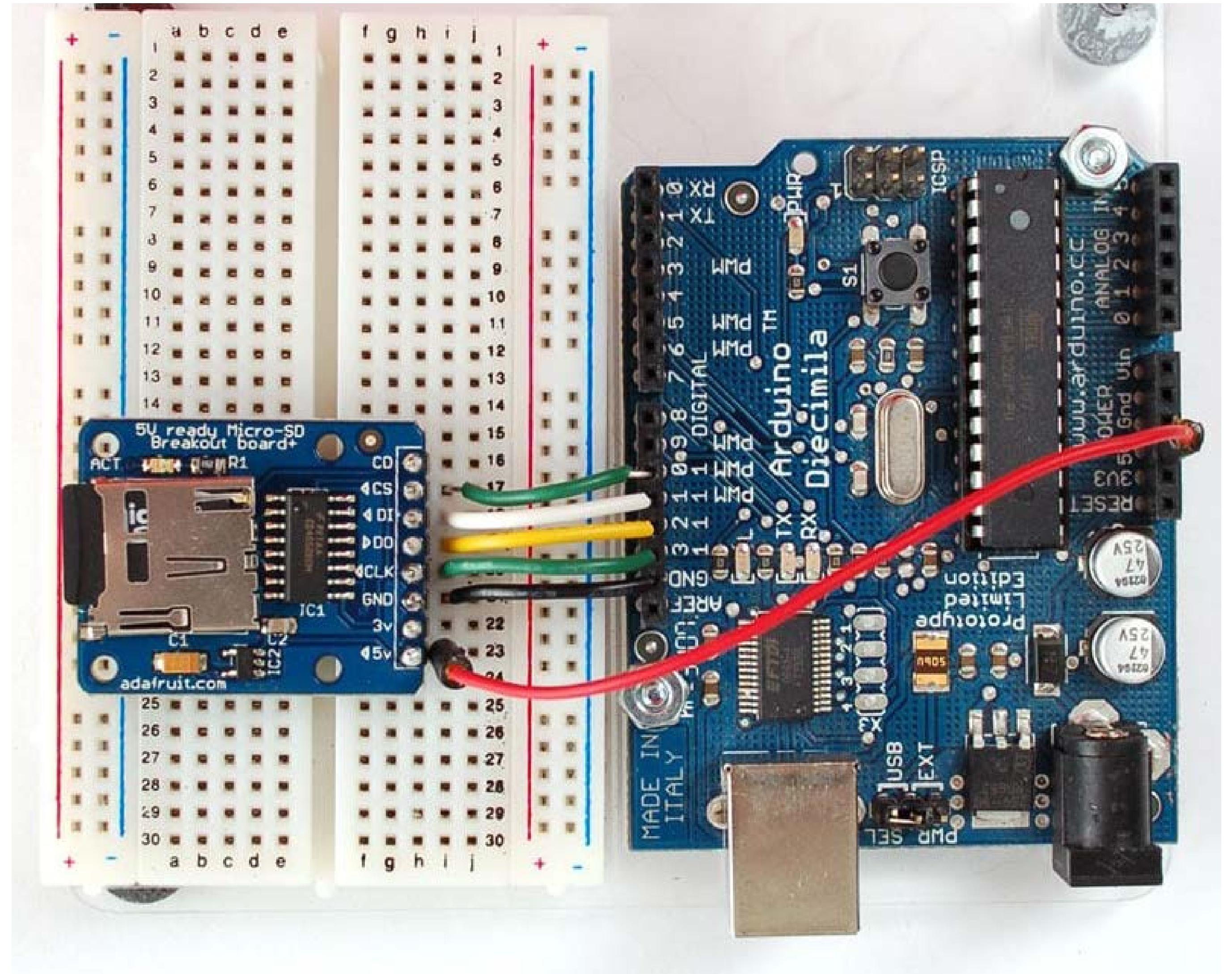
Connect DI to pin 11 (pin 51 for Arduino MEGA)

Connect CS to pin 10 (pin 53 for Arduino MEGA)

WARNING!

The digital pins in use are SPI, allowing fast communication with the SDCARD module.

[More info](#)



WRITING FILES TO AN SD CARD USING ARDUINO

CODING

The link below will direct you to a GitHub website.
Download the code from there.

[Link to download Arduino Code for SD Card Module - testing communication](#)

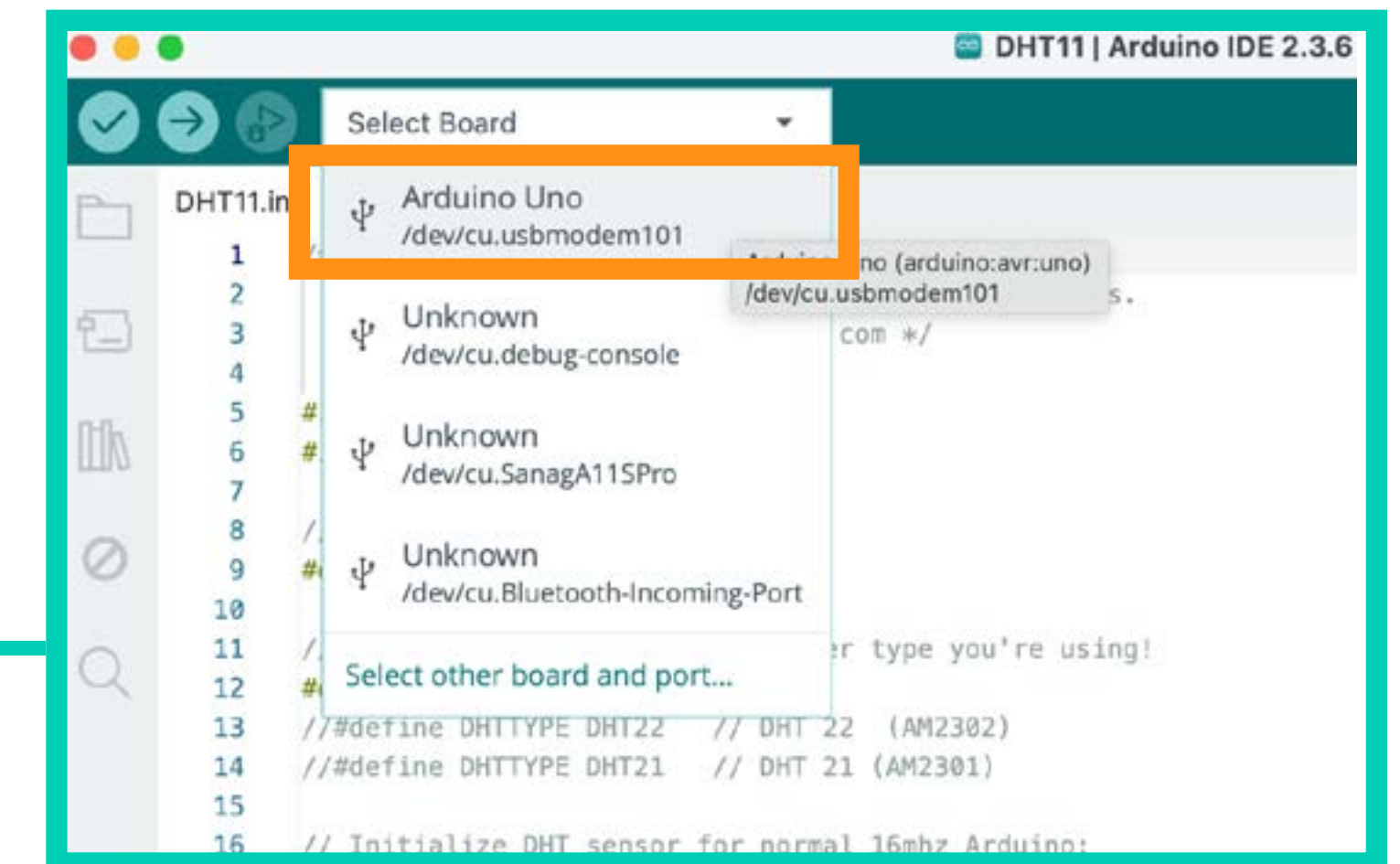


```
1  /* Arduino example code for DHT11, DHT22/AM2302
2     and DHT21/AM2301 temperature and humidity sensors.
3     More info: www.makerguides.com */
4
5  #include "Adafruit_Sensor.h"
6  #include "DHT.h"
7
8  // Set DHT pin:
9  #define DHTPIN 2
10
11 // Set DHT type, uncomment whatever type you're using!
12 #define DHTTYPE DHT11 // DHT 11
13 //#define DHTTYPE DHT22 // DHT 22 (AM2302)
14 //#define DHTTYPE DHT21 // DHT 21 (AM2301)
15
16 // Initialize DHT sensor for normal 16mhz Arduino:
17 DHT dht = DHT(DHTPIN, DHTTYPE);
18
19 void setup() {
20     Serial.begin(9600);
21
22     // Setup sensor:
```

WRITING FILES TO AN SD CARD USING ARDUINO

CODING: TEST COMMUNICATION

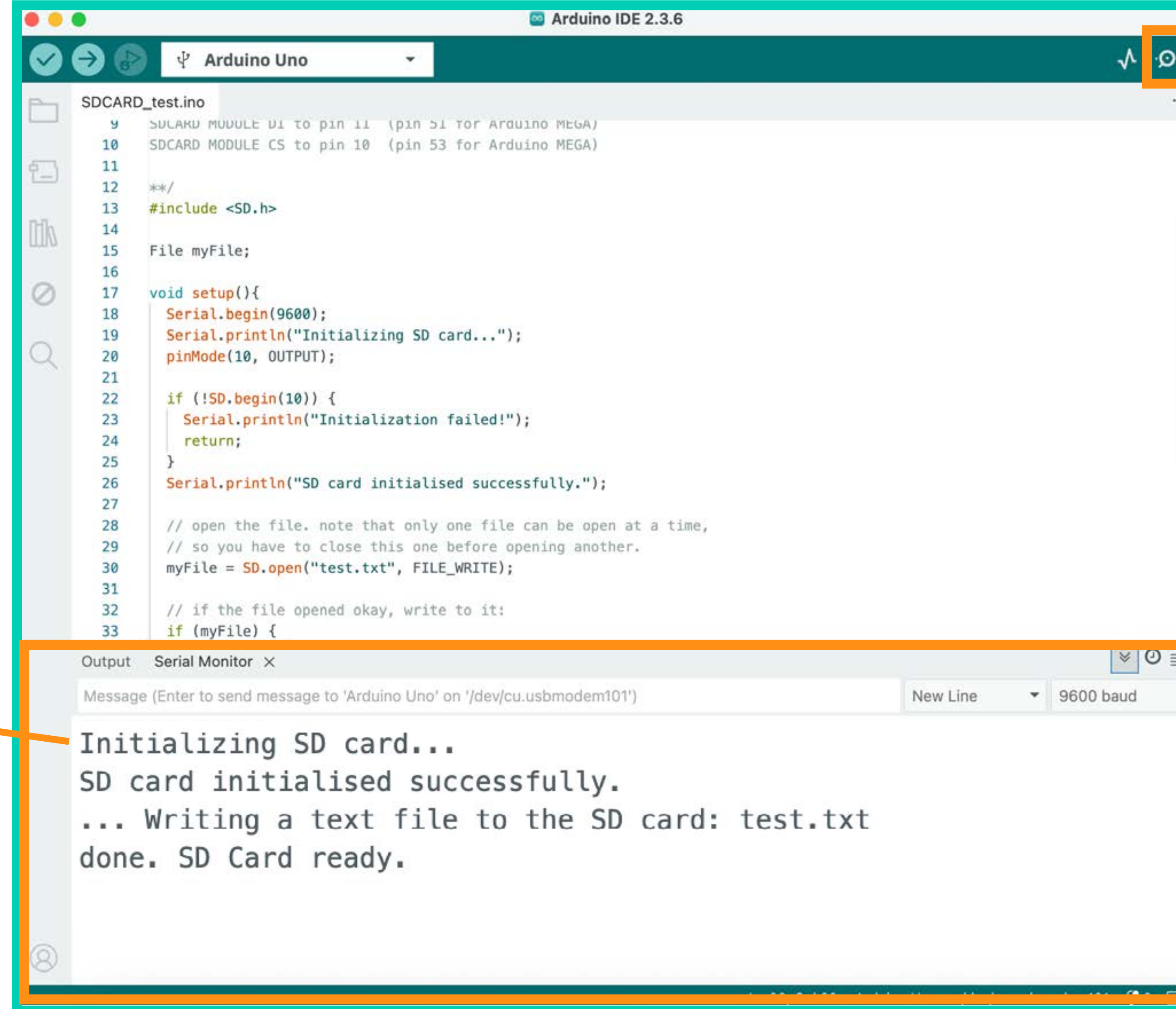
- Insert the SD card into the SD card module (if you haven't done it already)
- Plug The Arduino Board to your computer
- Open the Arduino file (.ino) that you previously downloaded
- Select the **Arduino UNO** Board
- Upload the code to the board



WRITING FILES TO AN SD CARD USING ARDUINO

Open the Serial Monitor to check if the communication between Arduino and the SD Card Module is working properly.

Your Serial Monitor should print the following message:



The screenshot shows the Arduino IDE 2.3.6 interface. The top toolbar has a button for the Serial Monitor, which is highlighted with an orange box. The main editor window displays the code for SDCARD_test.ino. The code includes comments for pin configurations, a header file inclusion, and a setup function that initializes the serial port and the SD card. The Serial Monitor window at the bottom shows the output of the code, which is also highlighted with an orange box. The output text is: "Initializing SD card...", "SD card initialised successfully.", "... Writing a text file to the SD card: test.txt", and "done. SD Card ready.".

```
SDCARD_test.ino
9  SDCARD MODULE DI to pin 11 (pin 51 for Arduino MEGA)
10 SDCARD MODULE CS to pin 10 (pin 53 for Arduino MEGA)
11
12 /**
13  *#include <SD.h>
14  *
15  *File myFile;
16  *
17  *void setup(){
18  *  Serial.begin(9600);
19  *  Serial.println("Initializing SD card...");
20  *  pinMode(10, OUTPUT);
21  *
22  *  if (!SD.begin(10)) {
23  *    Serial.println("Initialization failed!");
24  *    return;
25  *  }
26  *  Serial.println("SD card initialised successfully.");
27  *
28  *  // open the file. note that only one file can be open at a time,
29  *  // so you have to close this one before opening another.
30  *  myFile = SD.open("test.txt", FILE_WRITE);
31  *
32  *  // if the file opened okay, write to it:
33  *  if (myFile) {
```

Output Serial Monitor X

Message (Enter to send message to 'Arduino Uno' on '/dev/cu.usbmodem101')

New Line 9600 baud

Initializing SD card...
SD card initialised successfully.
... Writing a text file to the SD card: test.txt
done. SD Card ready.

Click to open
Serial Monitor.

Serial Monitor

WRITING FILES TO AN SD CARD USING ARDUINO

Done!

You have created a file titled 'TEXT.txt' written on your SD Card!

Now, remove the SD card from the module and insert it in your computer to check that your .txt file has been created properly. Let's open it and see what was written in it. It should look something like this:

A screenshot of a text editor window titled 'TEST.TXT'. The window has a light blue header bar with three colored window control buttons (red, yellow, green) on the left. The main area is white and contains 13 lines of text, each reading 'testing 1, 2, 3.'. A blue cursor is positioned at the end of the 8th line. The window is outlined with a thick teal border.

WRITING FILES TO AN SD CARD USING ARDUINO

PLAY AND TEST

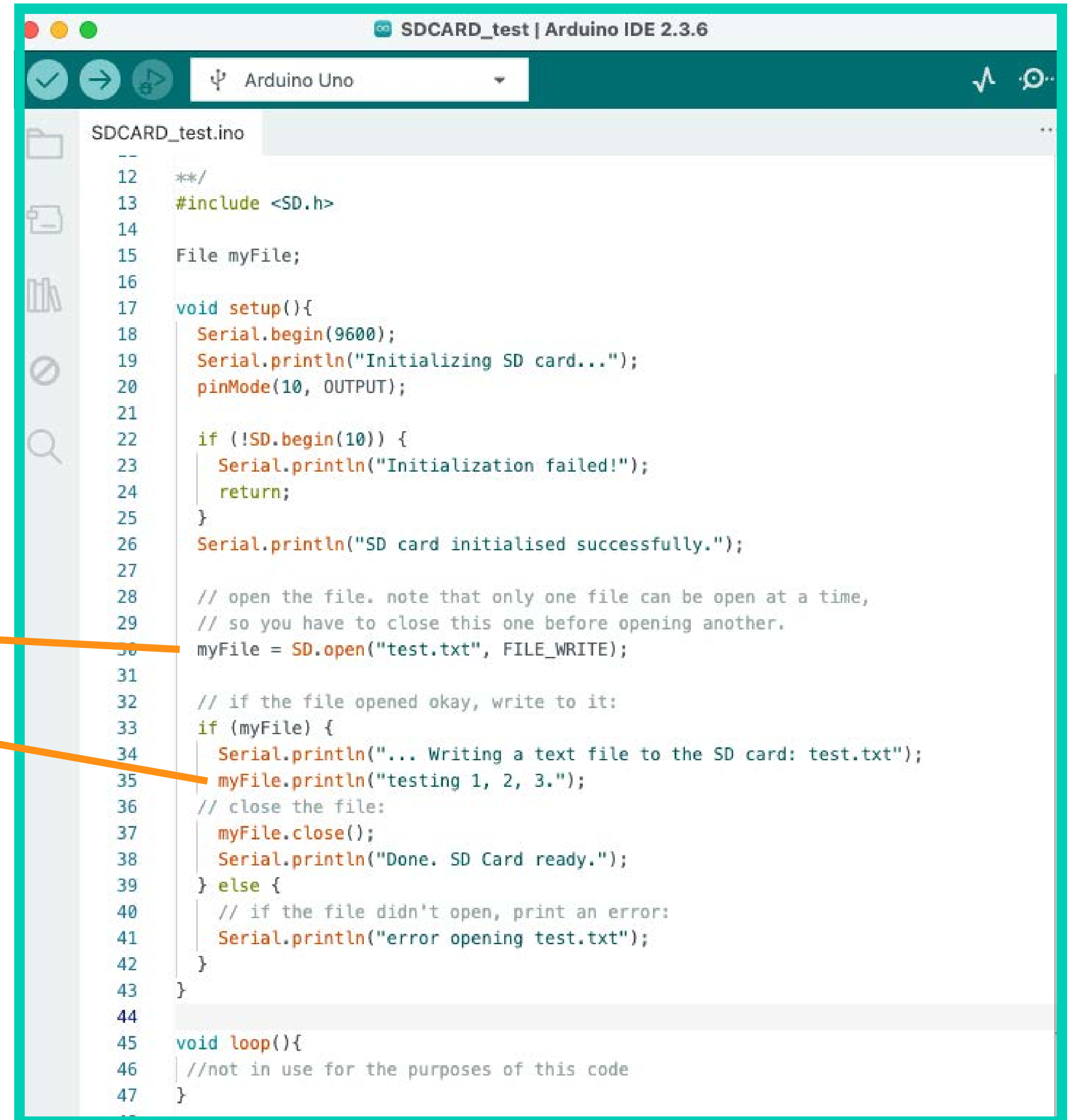
Insert the SD card back to the module.

Observe your code and :

- try changing the name of the .txt file.
 - try changing the content written in the .txt file
 - upload the new code to the Arduino Board.
- Has it worked?

*If you need help:

- research in internet
- approach a hackSpace technician



```
SDCARD_test | Arduino IDE 2.3.6
Arduino Uno

SDCARD_test.ino
--
12  /**
13  #include <SD.h>
14
15  File myFile;
16
17  void setup(){
18    Serial.begin(9600);
19    Serial.println("Initializing SD card...");
20    pinMode(10, OUTPUT);
21
22    if (!SD.begin(10)) {
23      Serial.println("Initialization failed!");
24      return;
25    }
26    Serial.println("SD card initialised successfully.");
27
28    // open the file. note that only one file can be open at a time,
29    // so you have to close this one before opening another.
30    myFile = SD.open("test.txt", FILE_WRITE);
31
32    // if the file opened okay, write to it:
33    if (myFile) {
34      Serial.println("... Writing a text file to the SD card: test.txt");
35      myFile.println("testing 1, 2, 3.");
36      // close the file:
37      myFile.close();
38      Serial.println("Done. SD Card ready.");
39    } else {
40      // if the file didn't open, print an error:
41      Serial.println("error opening test.txt");
42    }
43  }
44
45  void loop(){
46    //not in use for the purposes of this code
47  }
```


REFERENCE / SD CARD LIBRARY

ARDUINO DOCUMENTATION
SD CARD LIBRARY

<https://docs.arduino.cc/libraries/sd/>

