

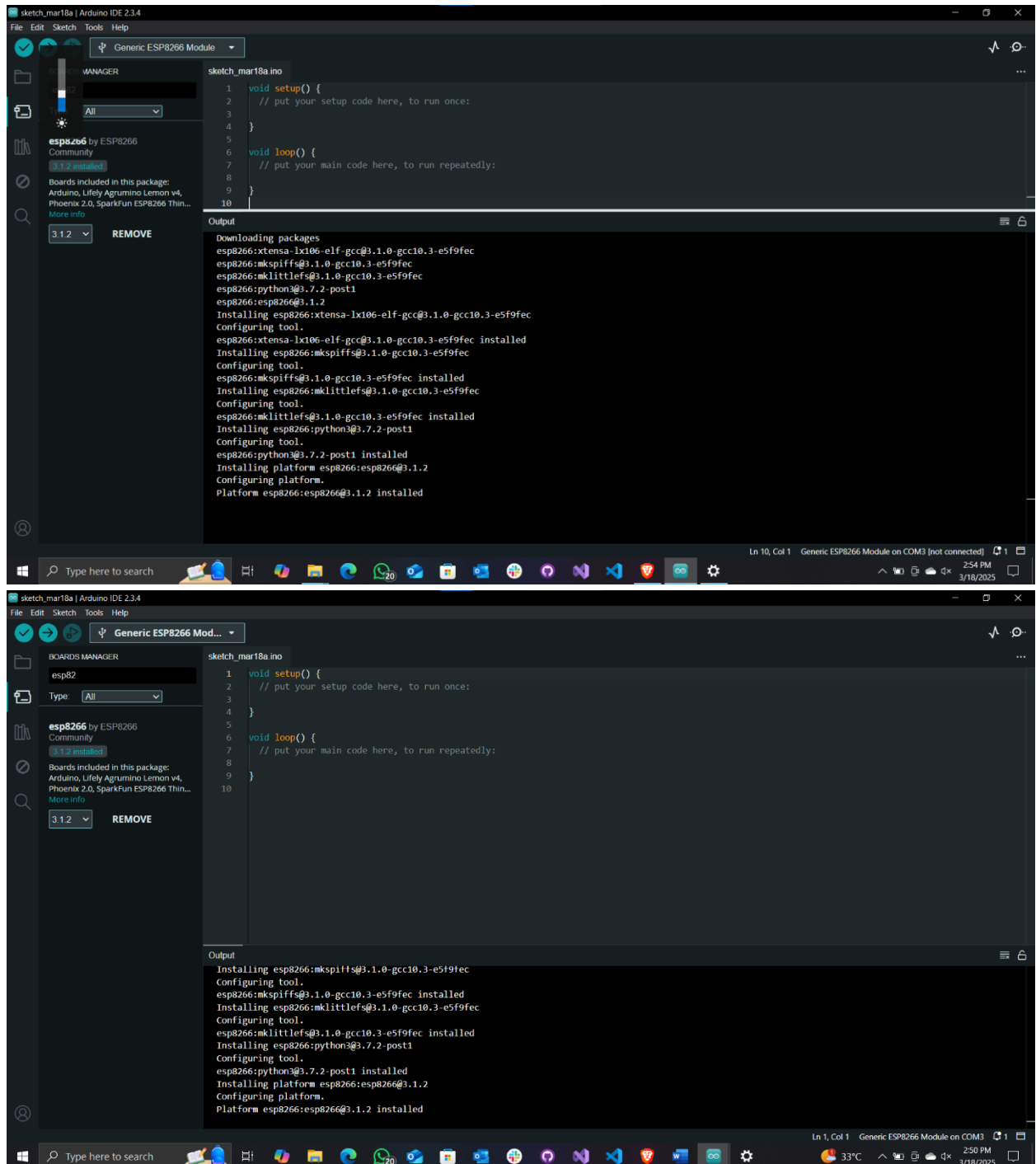
PLYMOUTH INDEX: 10953552

NSBM INDEX: 30040

USERNAME: Thisuri Gamage

1.

- This is a picture of downloading esp8266 package.



- Uploading the test code to scan the wifi networks that are available.

The image displays two screenshots of the Arduino IDE 2.3.4 interface, showing the process of scanning for WiFi networks using an ESP8266 module.

Top Screenshot: The code in `WiFiScan.ino` is shown. It includes the `ESP8266WiFi.h` library and sets the module to station mode. The `setup()` function initializes the serial port at 115200 baud and prints a message. The `loop()` function scans for networks and prints the results. The Serial Monitor shows the output of the scan, listing several networks including NISM-Staff, NISM-Student, NISM-Guest, Guest-Students, and D-Signage.

```

7  #include <ESP8266WiFi.h>
8
9  void setup() {
10   Serial.begin(115200);
11   Serial.println(F("ESP8266 Wifi scan example"));
12
13   // Set Wifi to station mode
14   WiFi.mode(WIFI_STA);
15
16   // Disconnect from an AP if it was previously connected
17   WiFi.disconnect();
18   delay(100);
19 }
20
21 void loop() {
22   String ssid;
23   int32_t rssi;
24   uint8_t encryptionType;
25   uint8_t *bssid;
26   int32_t channel;
27   bool hidden;
28   int scanResult;
29
30   Serial.println(F("Starting Wifi scan..."));
31
32   scanResult = WiFi.scanNetworks(false, true);
33
34   if (scanResult == 0) {
35     Serial.println(F("No networks found"));
36   } else if (scanResult > 0) {
37     Serial.printf(PSTR("%d networks found:\n"), scanResult);
38
39     // Print unsorted scan results
40     for (int i = 0; i < scanResult; i++) {
41       WiFi.getNetworkInfo(i, ssid, encryptionType, rssi, bssid, channel, hidden);
42
43       // get extra info
44       const bss_info *bssInfo = WiFi.getScanInfoByIndex(i);
45       String phyMode;
46       const char *wps = "";
47       if (bssInfo) {
48         phyMode.reserve(12);
49         phyMode = r("802.11");
50         String slash;
51         if (bssInfo->phy_11b) {
52           phyMode += "b";
53           slash = '/';
54         }
55         if (bssInfo->phy_11g) {
56           phyMode += slash + "g";
57           slash = '/';
58         }
59         if (bssInfo->phy_11n) {
60           phyMode += slash + "n";
61         }
62       }
63     }
64   }
65 }

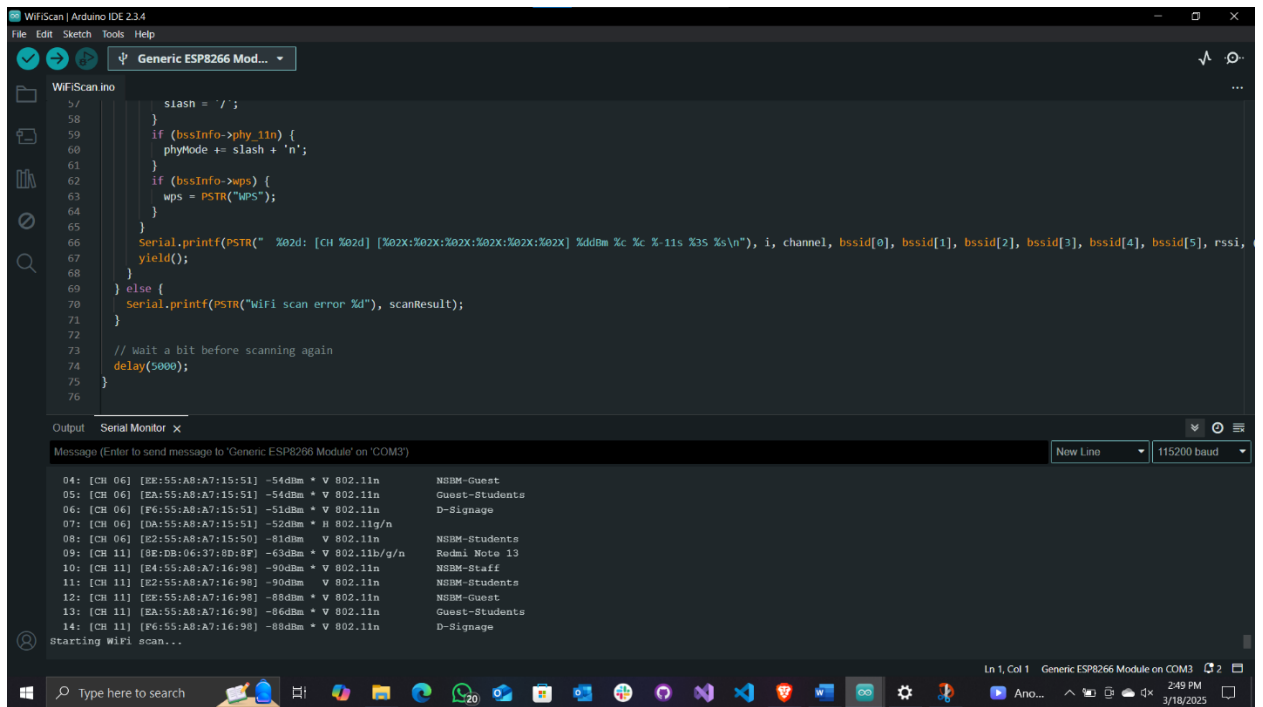
```

Bottom Screenshot: The code is updated to include the `WiFi.scanNetworks()` function call. The Serial Monitor shows the output of the scan, listing several networks including /slt_fiber and Redmi Note 13.

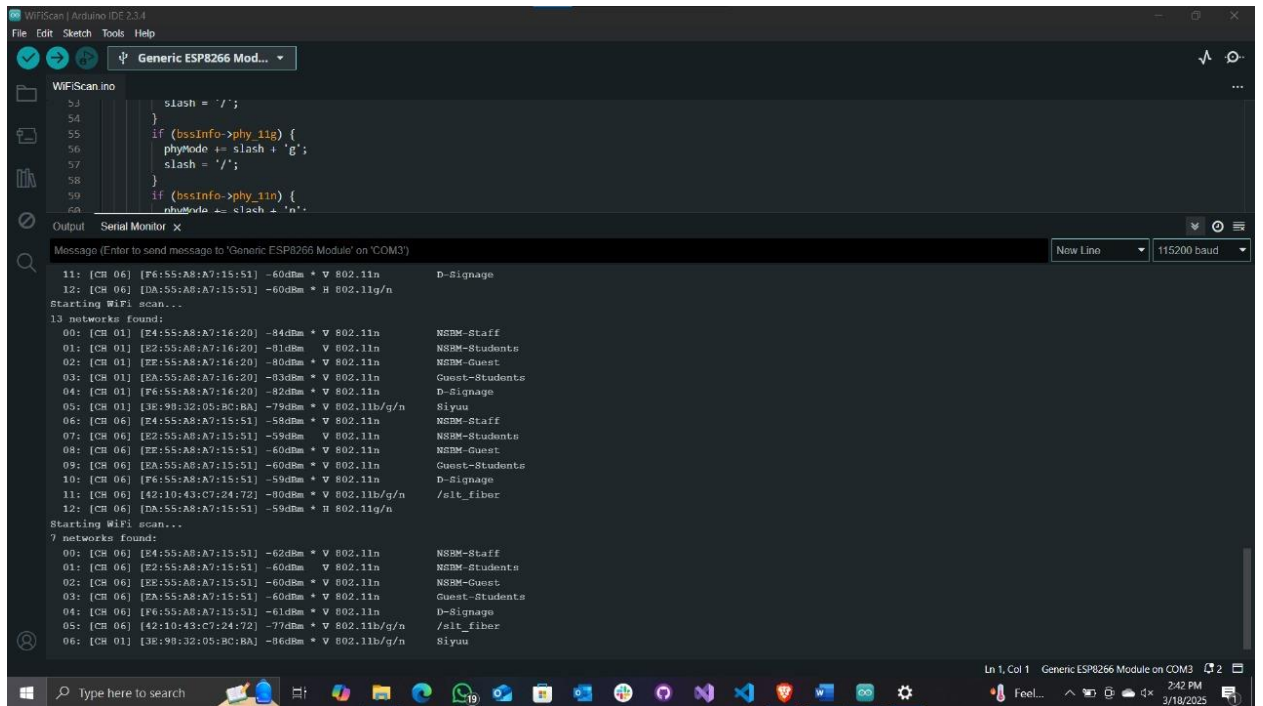
```

32   scanResult = WiFi.scanNetworks(false, true);
33
34   if (scanResult == 0) {
35     Serial.println(F("No networks found"));
36   } else if (scanResult > 0) {
37     Serial.printf(PSTR("%d networks found:\n"), scanResult);
38
39     // Print unsorted scan results
40     for (int i = 0; i < scanResult; i++) {
41       WiFi.getNetworkInfo(i, ssid, encryptionType, rssi, bssid, channel, hidden);
42
43       // get extra info
44       const bss_info *bssInfo = WiFi.getScanInfoByIndex(i);
45       String phyMode;
46       const char *wps = "";
47       if (bssInfo) {
48         phyMode.reserve(12);
49         phyMode = r("802.11");
50         String slash;
51         if (bssInfo->phy_11b) {
52           phyMode += "b";
53           slash = '/';
54         }
55         if (bssInfo->phy_11g) {
56           phyMode += slash + "g";
57           slash = '/';
58         }
59         if (bssInfo->phy_11n) {
60           phyMode += slash + "n";
61         }
62       }
63     }
64   }
65 }

```



2. This is a Screenshot that shows all the wifi networks at the moment.



- As you can see there are,

NSBM-Staff

NSBM-Students

NSBM-Guest

Guest-Students

D-Signage

/slt_fiber

Siyuu

WIFI networks are available right now.

3. Possible challenges we could face:

- Some systems fail to identify CH340 drivers which control the Arduino board because users need to install the driver manually.
- The selection of an incompatible board (when using Generic ESP8266 in place of Arduino UNO) leads to upload failure.
- The board needs a proper COM port selection to connect with the computer otherwise it will not transmit signals.
- The process of uploading firmware to ESP8266 becomes unsuccessful when users use incorrect wiring or defective USB cables.
- Reading the output from the Serial Monitor becomes impossible when the baud rate setting is not exact.

Gladly, I didn't run into any difficulties but I had to download cp2102 driver because it didn't get detected at first.