



DesignNews

Designing, Building and Coding Custom Raspberry Pi RP2040 Arduino Devices

Day 3:

The RP2040, Arduino and BLE

Sponsored by

DigiKey



Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.

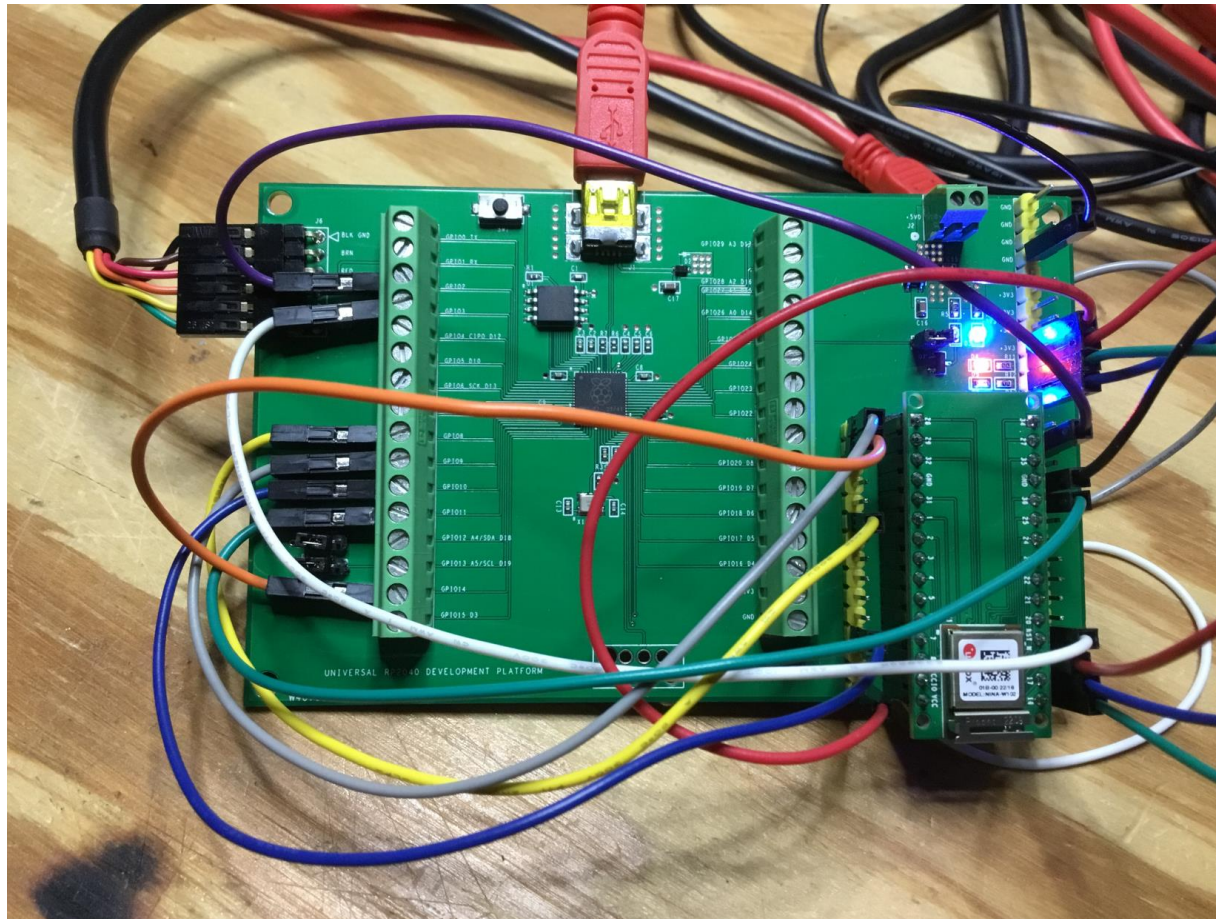


Fred Eady

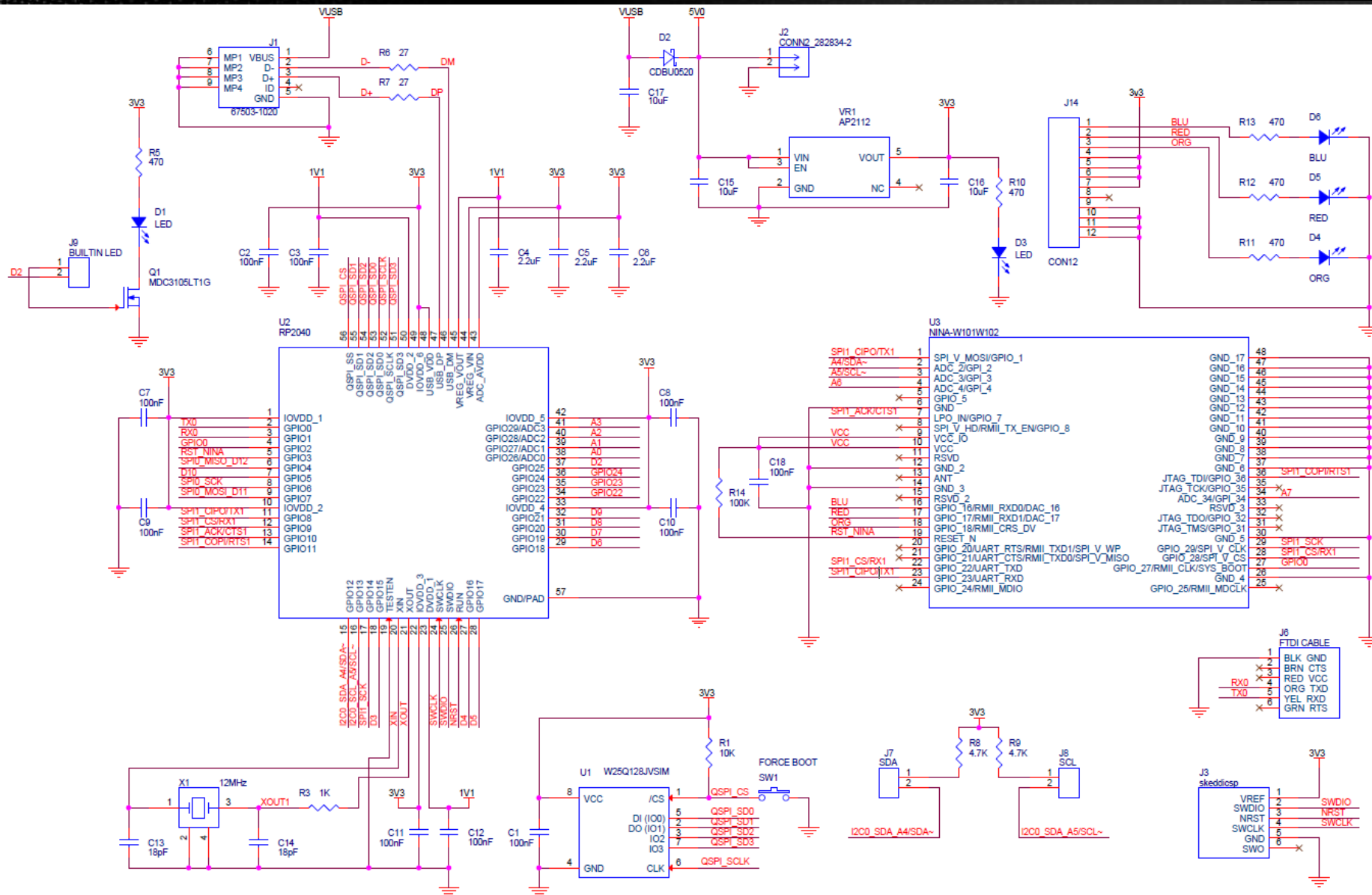
Visit 'Lecturer Profile' in your console for more details.

AGENDA

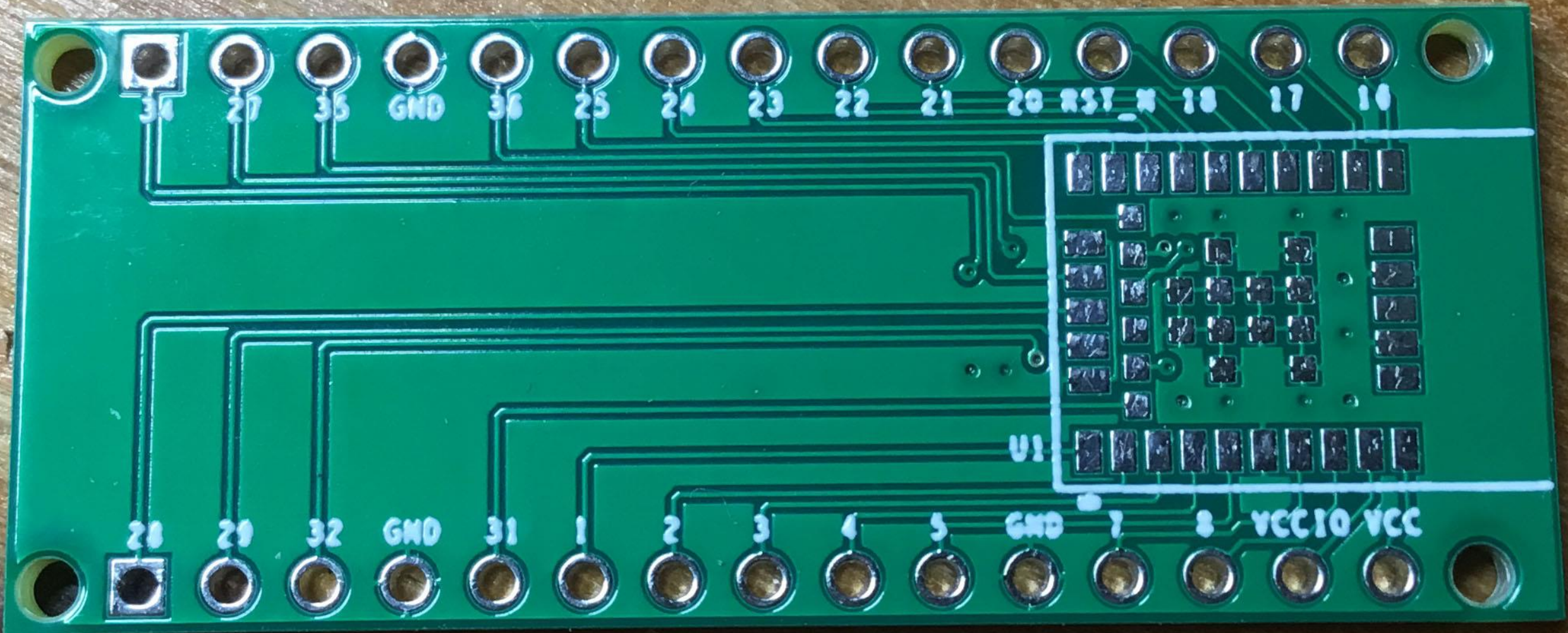
- **RP2040 Arduino BLE Hardware**
- **u-blox Frankenstein**
- **Arduino BLE On Your Phone**



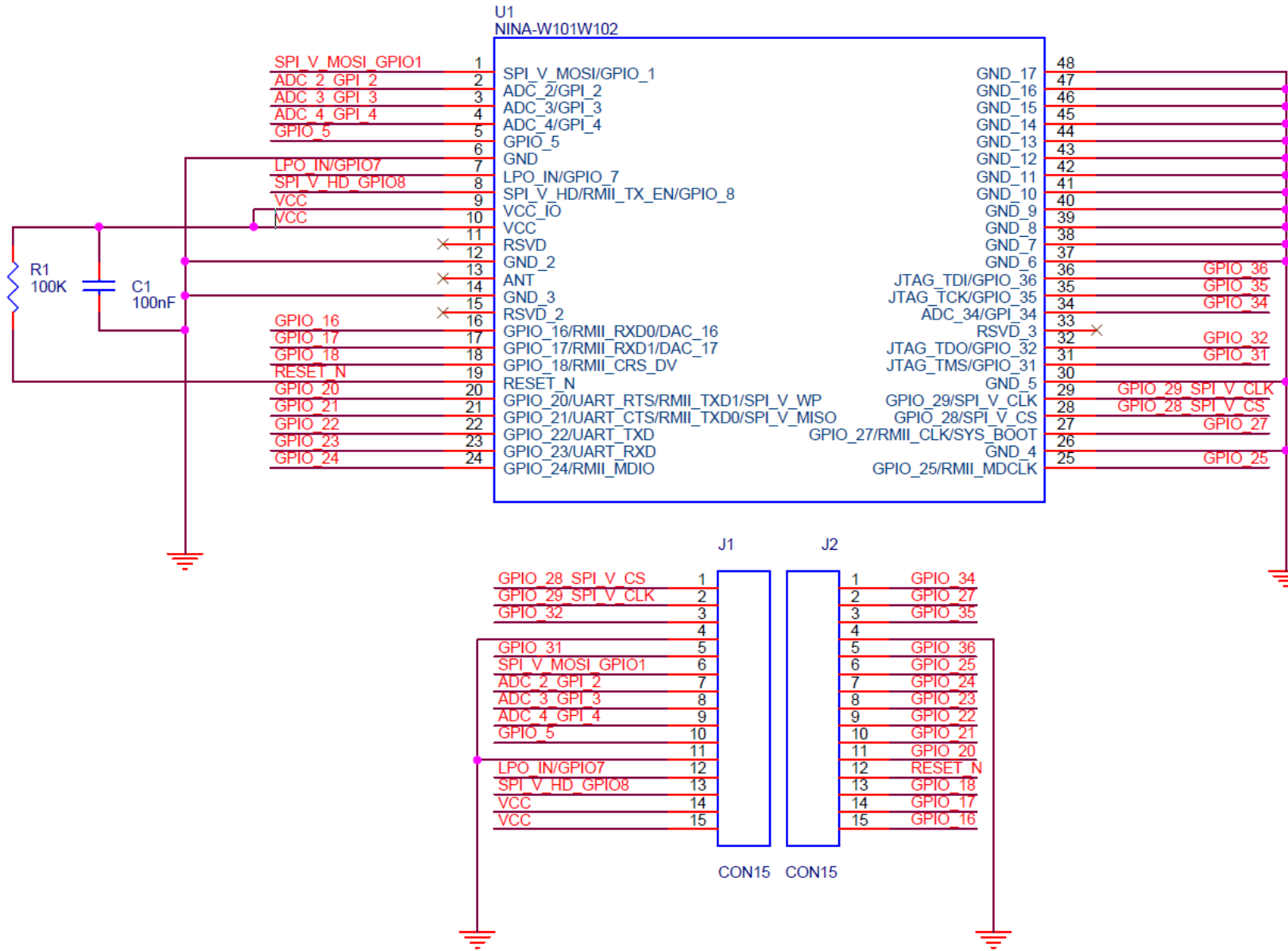
RP2040 Arduino BLE Hardware Design



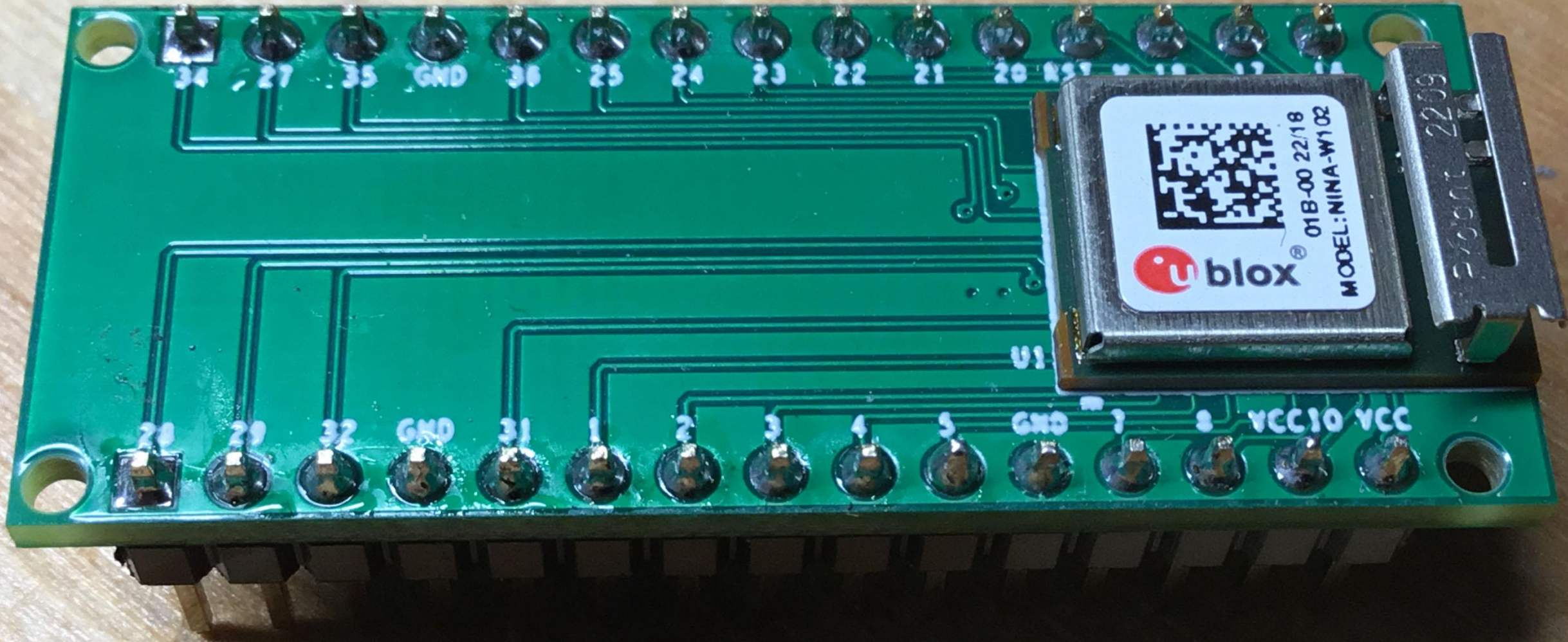
RP2040 Arduino BLE Hardware Design



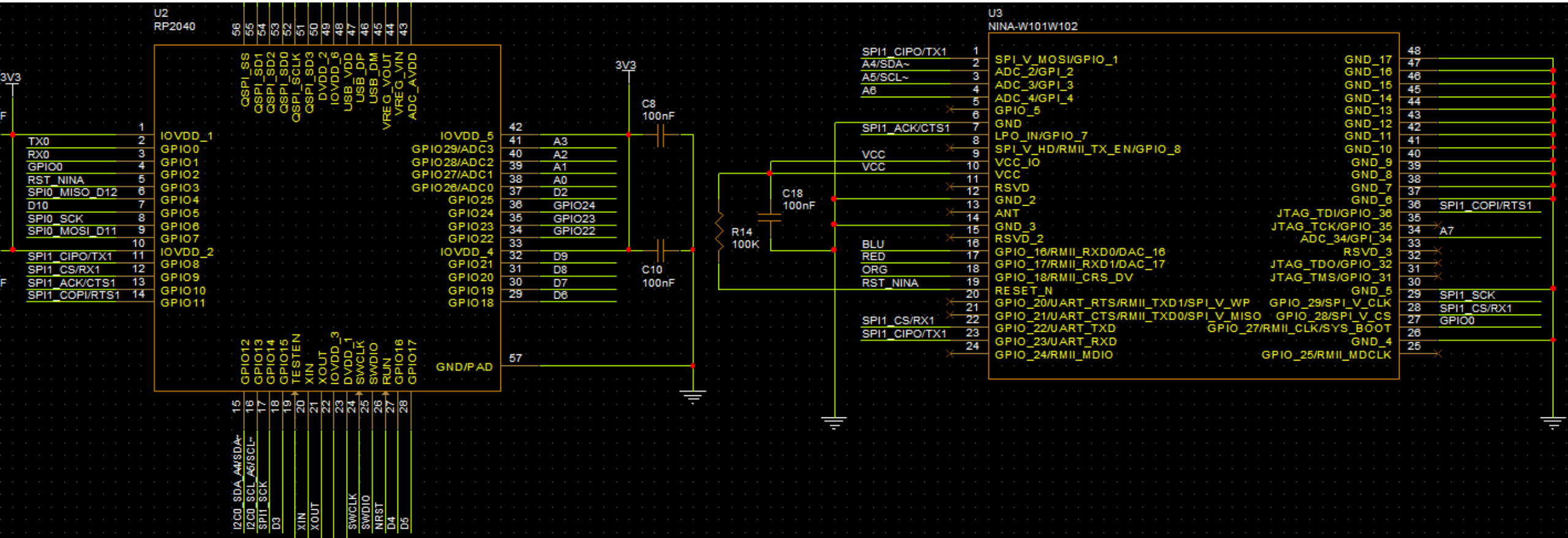
RP2040 Arduino BLE Hardware Design



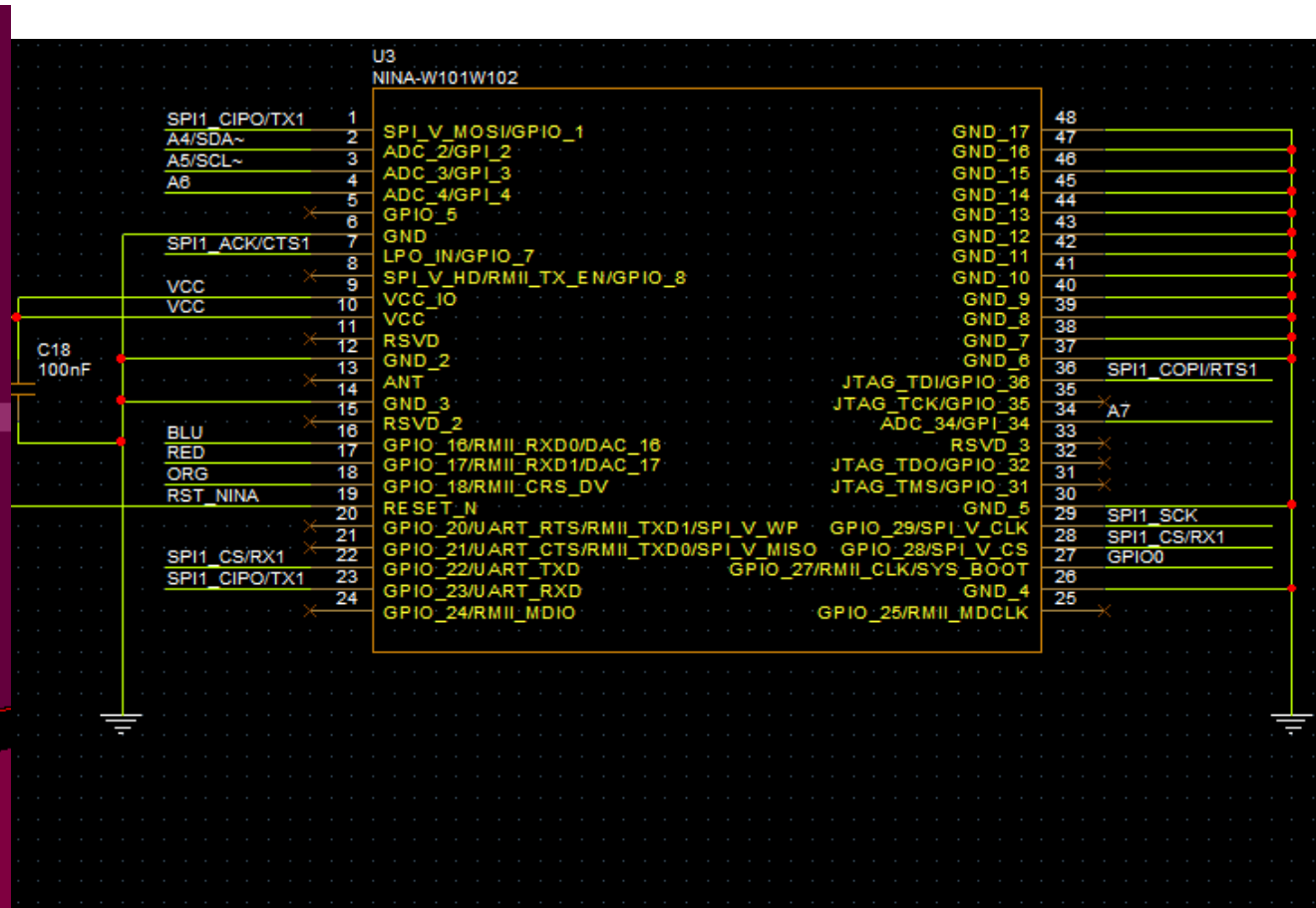
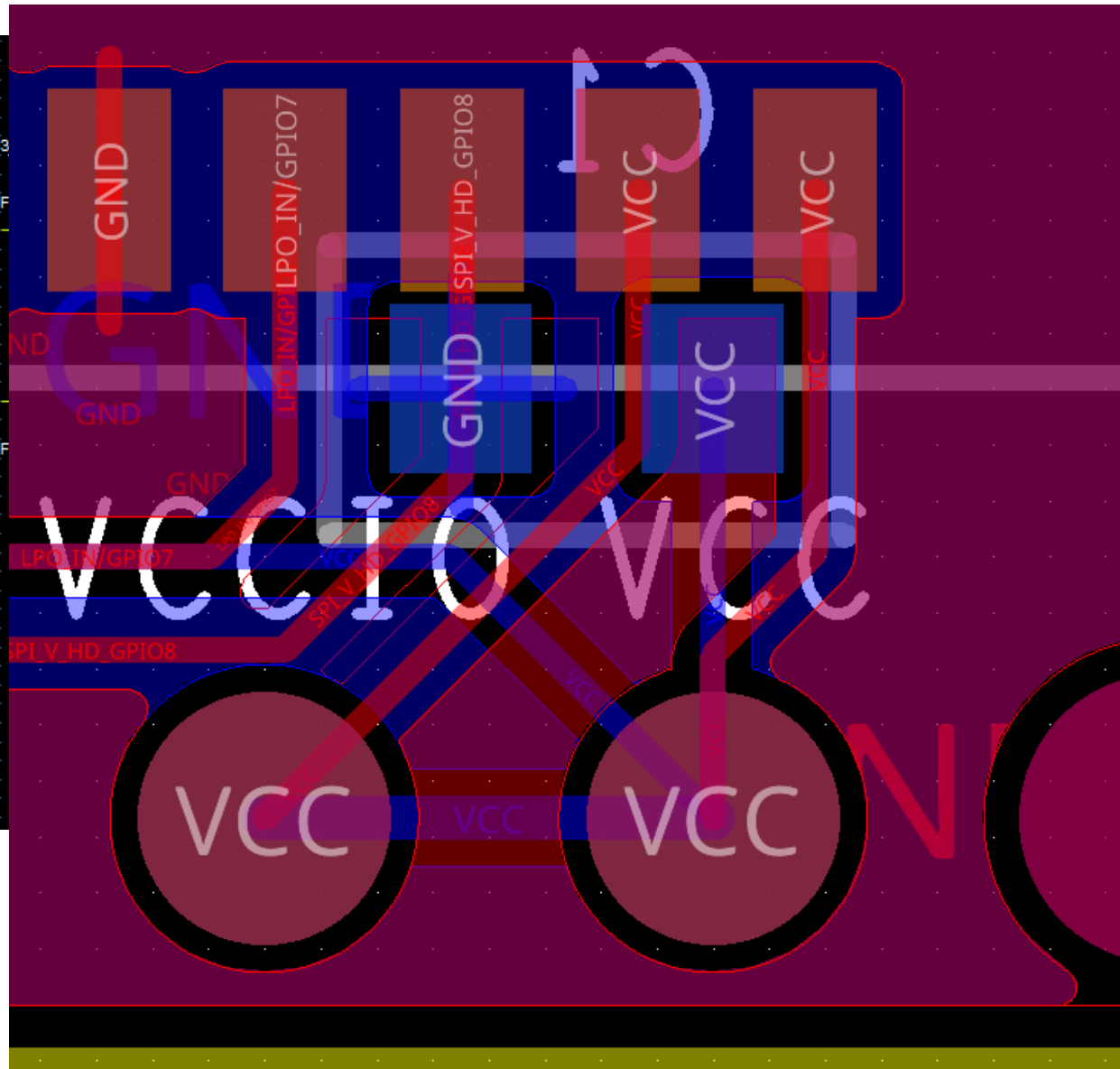
RP2040 Arduino BLE Hardware Design



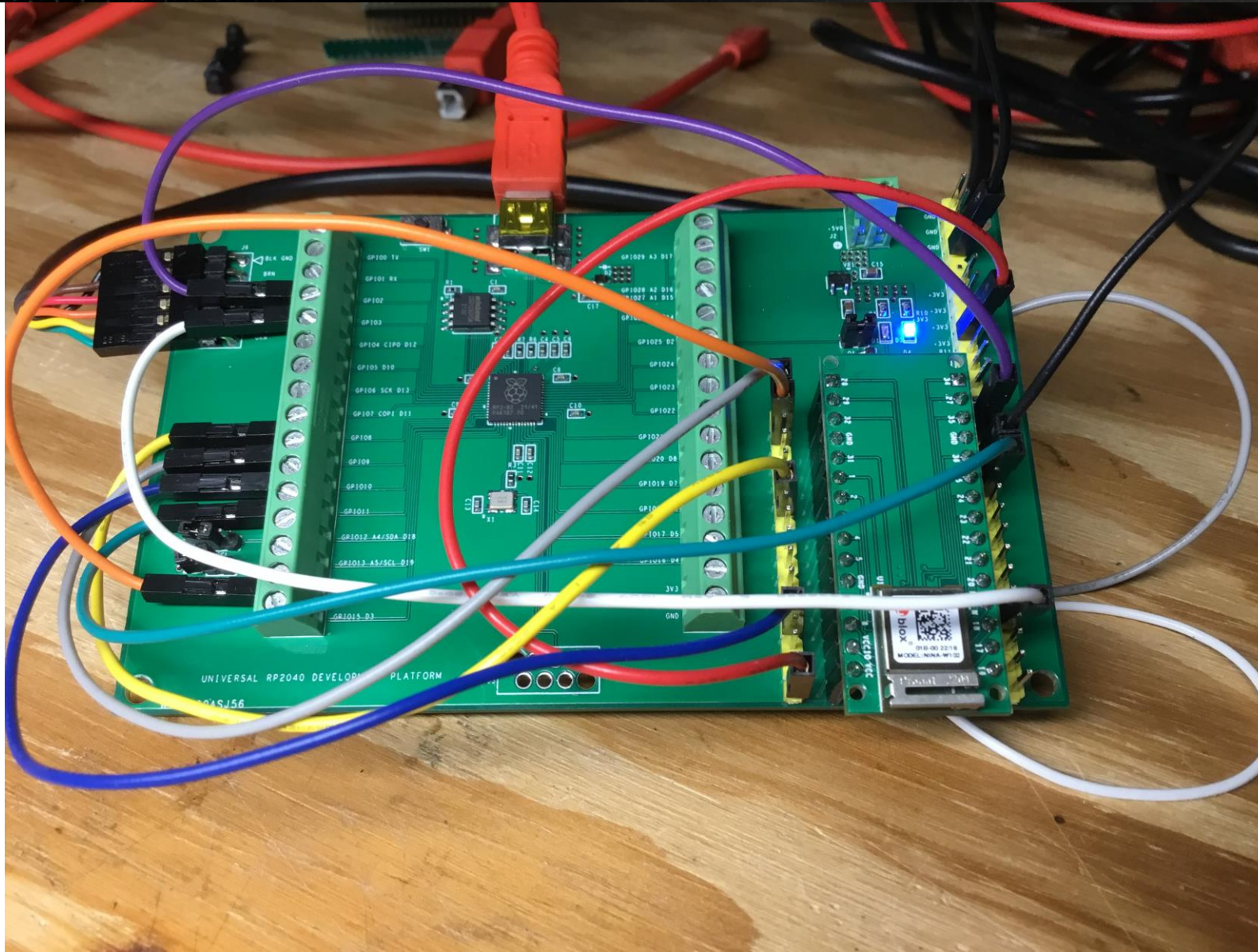
RP2040 Arduino BLE Hardware Design



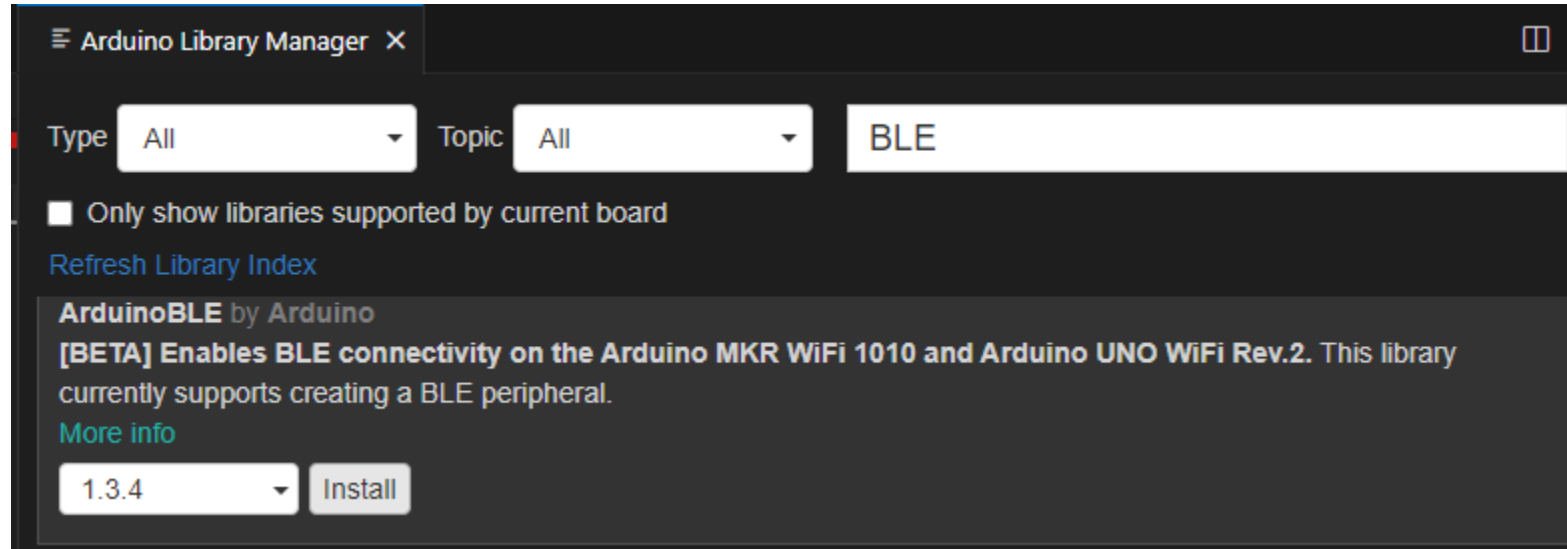
RP2040 Arduino BLE Hardware Design



RP2040 Arduino BLE Hardware Design



Load the Arduino BLE and Wi-Fi Libraries



Arduino Library Manager

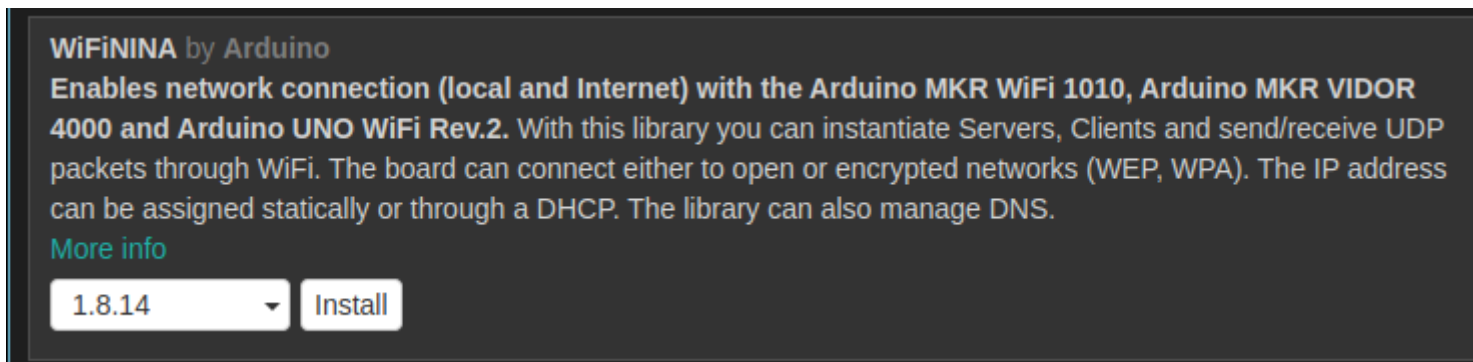
Type: All Topic: All BLE

Only show libraries supported by current board

[Refresh Library Index](#)

ArduinoBLE by Arduino
[BETA] Enables BLE connectivity on the Arduino MKR WiFi 1010 and Arduino UNO WiFi Rev.2. This library currently supports creating a BLE peripheral.
[More info](#)

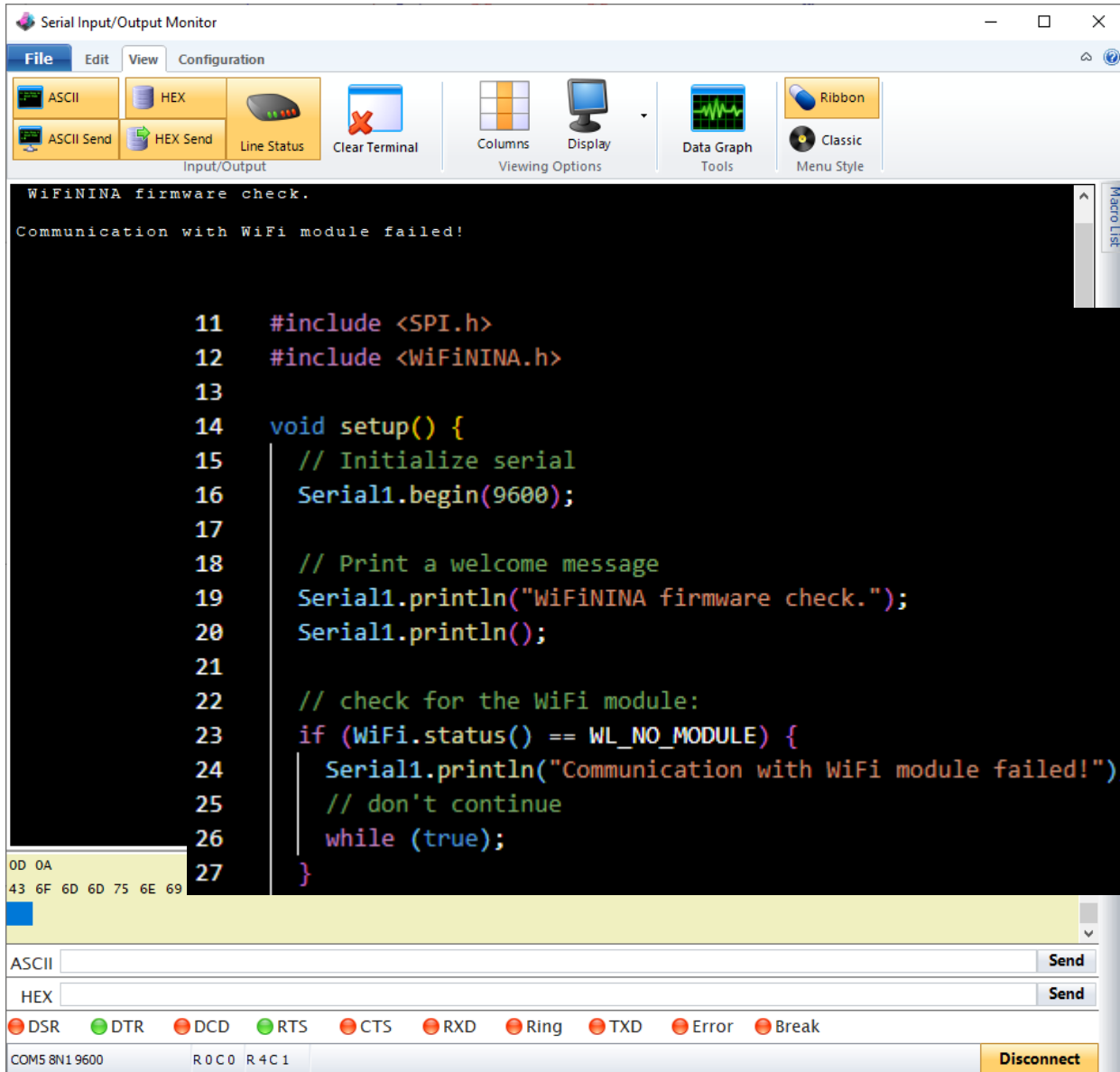
1.3.4 Install



WiFiNINA by Arduino
Enables network connection (local and Internet) with the Arduino MKR WiFi 1010, Arduino MKR VIDOR 4000 and Arduino UNO WiFi Rev.2. With this library you can instantiate Servers, Clients and send/receive UDP packets through WiFi. The board can connect either to open or encrypted networks (WEP, WPA). The IP address can be assigned statically or through a DHCP. The library can also manage DNS.
[More info](#)

1.8.14 Install

Pins Up - The Monster Is Dead



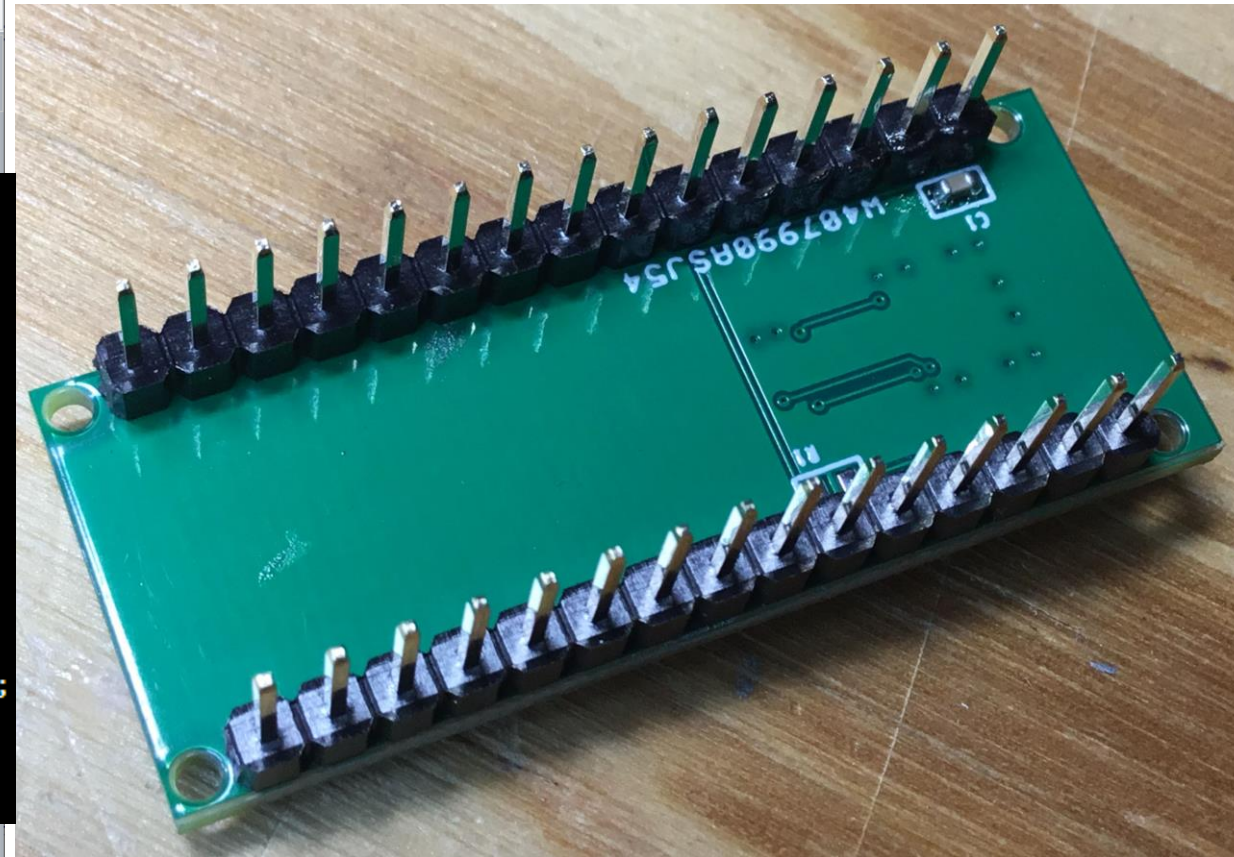
The screenshot shows the Serial Input/Output Monitor window. The output area displays the following text:

```
WiFiNINA firmware check.  
Communication with WiFi module failed!
```

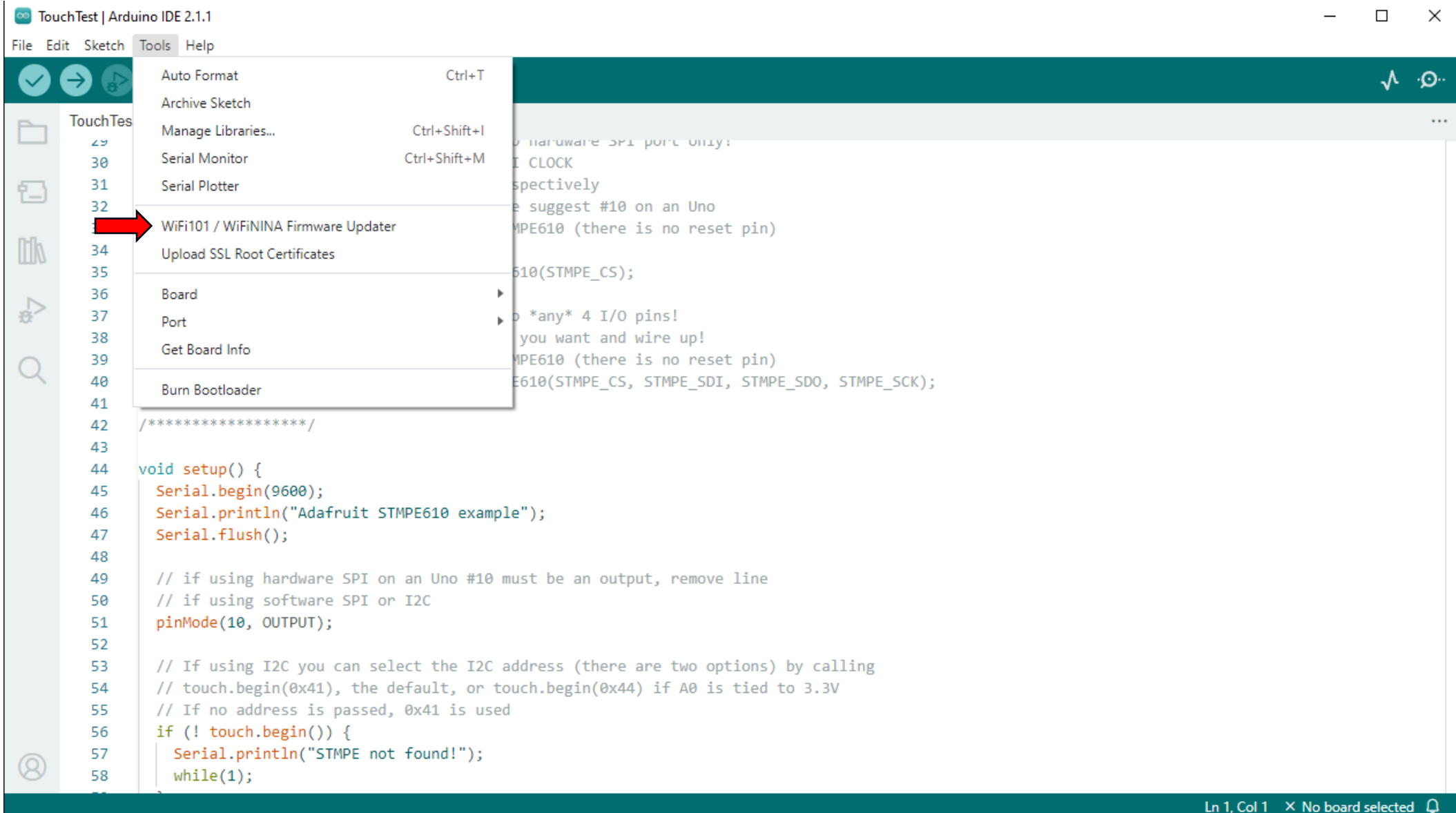
The code area shows the following C++ code:

```
11 #include <SPI.h>  
12 #include <WiFiNINA.h>  
13  
14 void setup() {  
15     // Initialize serial  
16     Serial1.begin(9600);  
17  
18     // Print a welcome message  
19     Serial1.println("WiFiNINA firmware check.");  
20     Serial1.println();  
21  
22     // check for the WiFi module:  
23     if (WiFi.status() == WL_NO_MODULE) {  
24         Serial1.println("Communication with WiFi module failed!");  
25         // don't continue  
26         while (true);  
27     }
```

The bottom of the window shows the configuration for the serial port: COM5 8N1 9600, R 0 C 0 R 4 C 1, and a Disconnect button.



Give Life to the Monster



The screenshot shows the Arduino IDE 2.1.1 interface. The 'Tools' menu is open, and the 'WiFi101 / WiFinINA Firmware Updater' option is highlighted with a red arrow. The background shows a code editor with the following code:

```
42 /*****  
43  
44 void setup() {  
45   Serial.begin(9600);  
46   Serial.println("Adafruit STMPE610 example");  
47   Serial.flush();  
48  
49   // if using hardware SPI on an Uno #10 must be an output, remove line  
50   // if using software SPI or I2C  
51   pinMode(10, OUTPUT);  
52  
53   // If using I2C you can select the I2C address (there are two options) by calling  
54   // touch.begin(0x41), the default, or touch.begin(0x44) if A0 is tied to 3.3V  
55   // If no address is passed, 0x41 is used  
56   if (! touch.begin()) {  
57     Serial.println("STMPE not found!");  
58     while(1);  
59   }
```

Give Life to the Monster

ble_peripheral | Arduino IDE 2.1.1

File Edit Sketch Tools Help

Select Board

```
ble_peripheral.ino
1 #include <ArduinoBLE.h>
2
3 int buttonPin = 2;
4 boolean ledSwitch;
5
6 BLEService LEDService("10B10000-F8E2-537E-4F6C-D104768A1214"); // BLE LED Service
7 // BLE LED Switch Characteristic
8 BLEByteCharacteristic LEDCharacteristic;
9 void setup() {
10   Serial.begin(9600);
11   pinMode(buttonPin, INPUT);
12   // begin initialization
13   if (!BLE.begin()) {
14     Serial.println("start");
15   }
16   // set advertised local name
17   BLE.setLocalName("Button");
18   BLE.setAdvertisedService(LEDService);
19   // add the characteristic
20   LEDService.addCharacteristic(LEDCharacteristic);
21   // add service
22   BLE.addService(LEDService);
23   // start advertising
24   BLE.advertise();
25   Serial.println("BLE LED Peripheral, waiting for connections...");
26 }
27 void loop() {
28   // listen for BLE peripherals to connect:
29   BLEDevice central = BLE.central();
30   // if a central is connected to peripheral:
```

WiFi101 / Wi-FiNINA Firmware Updater

Select Board

Arduino Nano RP2040 Connect at COM3 **CHECK UPDATES**

Select firmware version

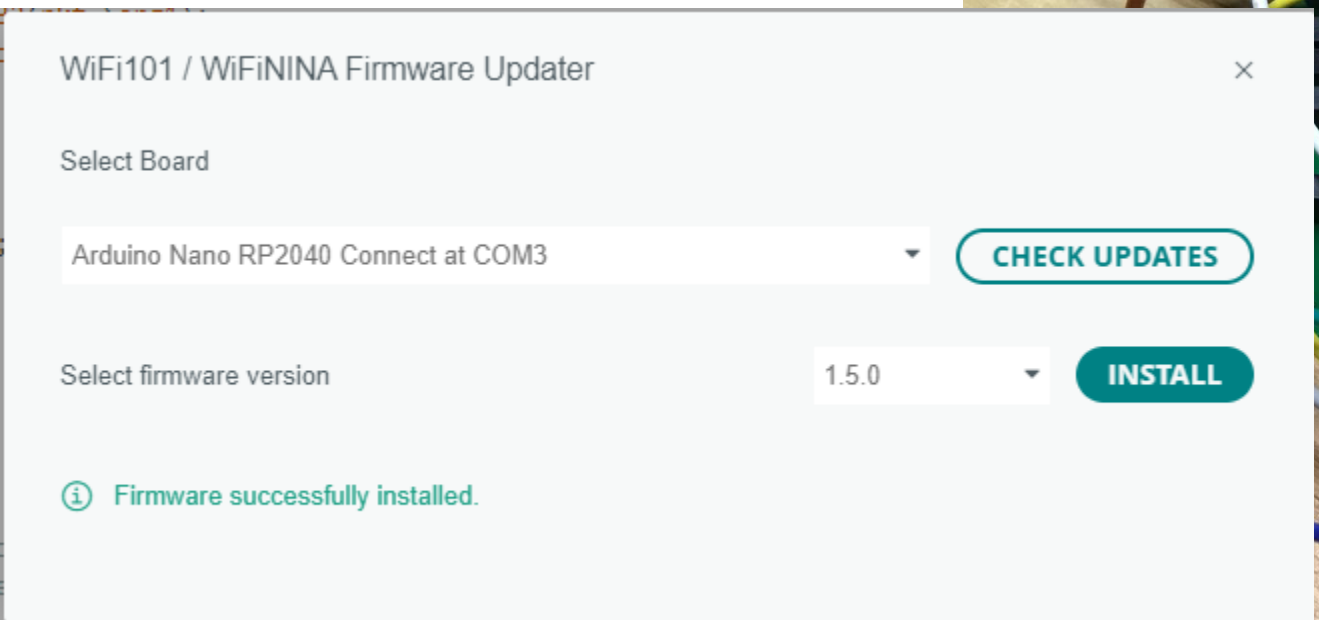
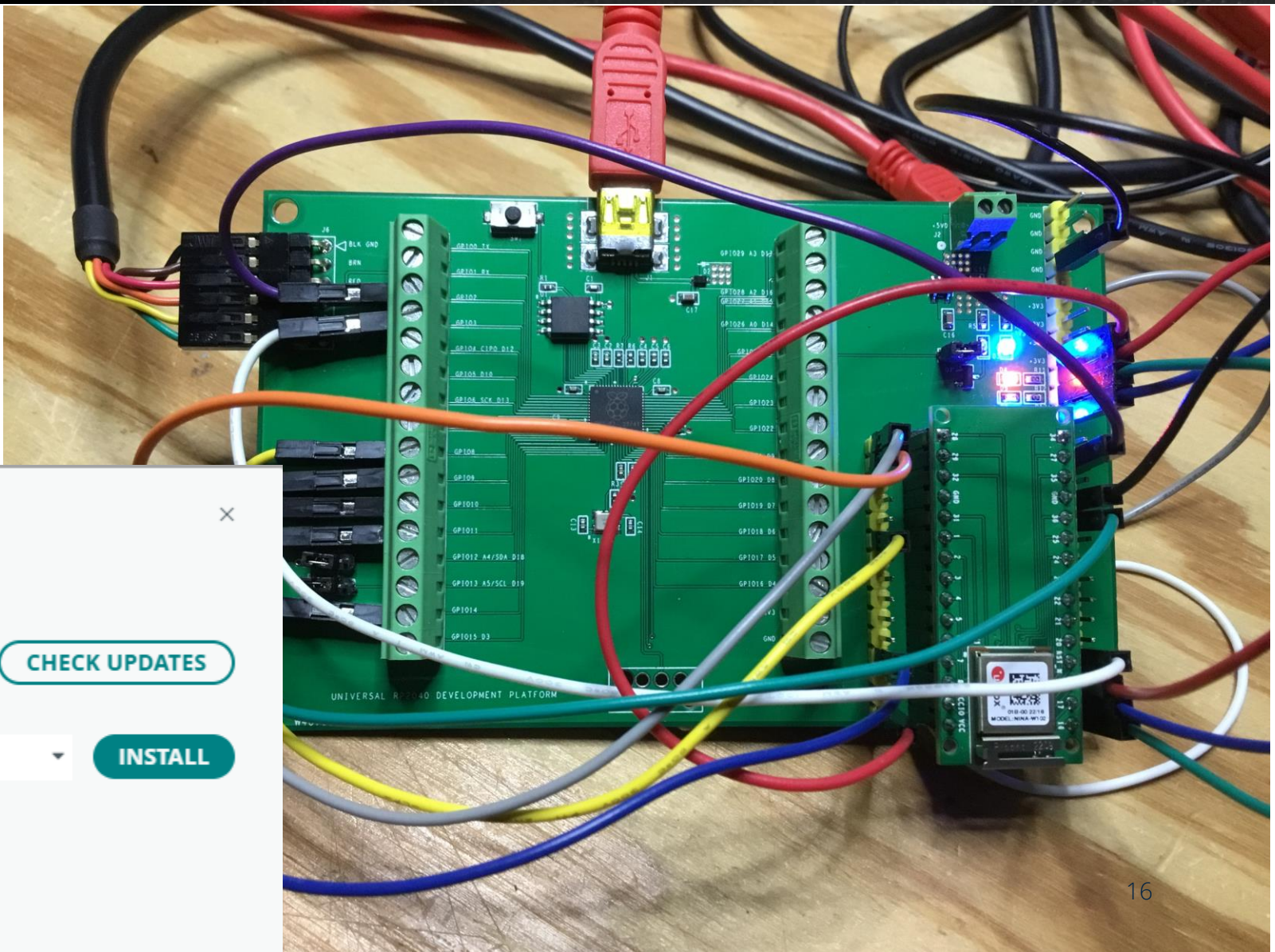
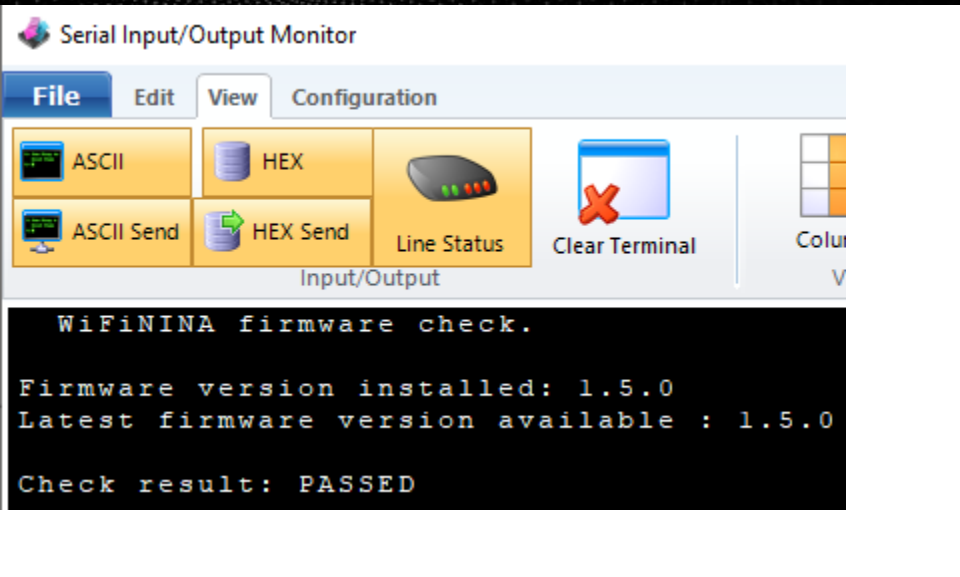
- 1.5.0
- 1.5.0
- 1.4.8
- 1.4.7
- 1.4.6
- 1.4.5

INSTALL

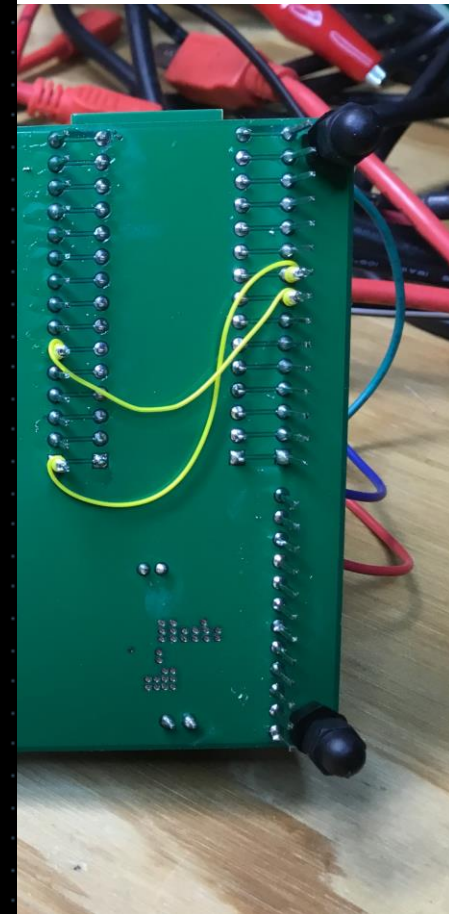
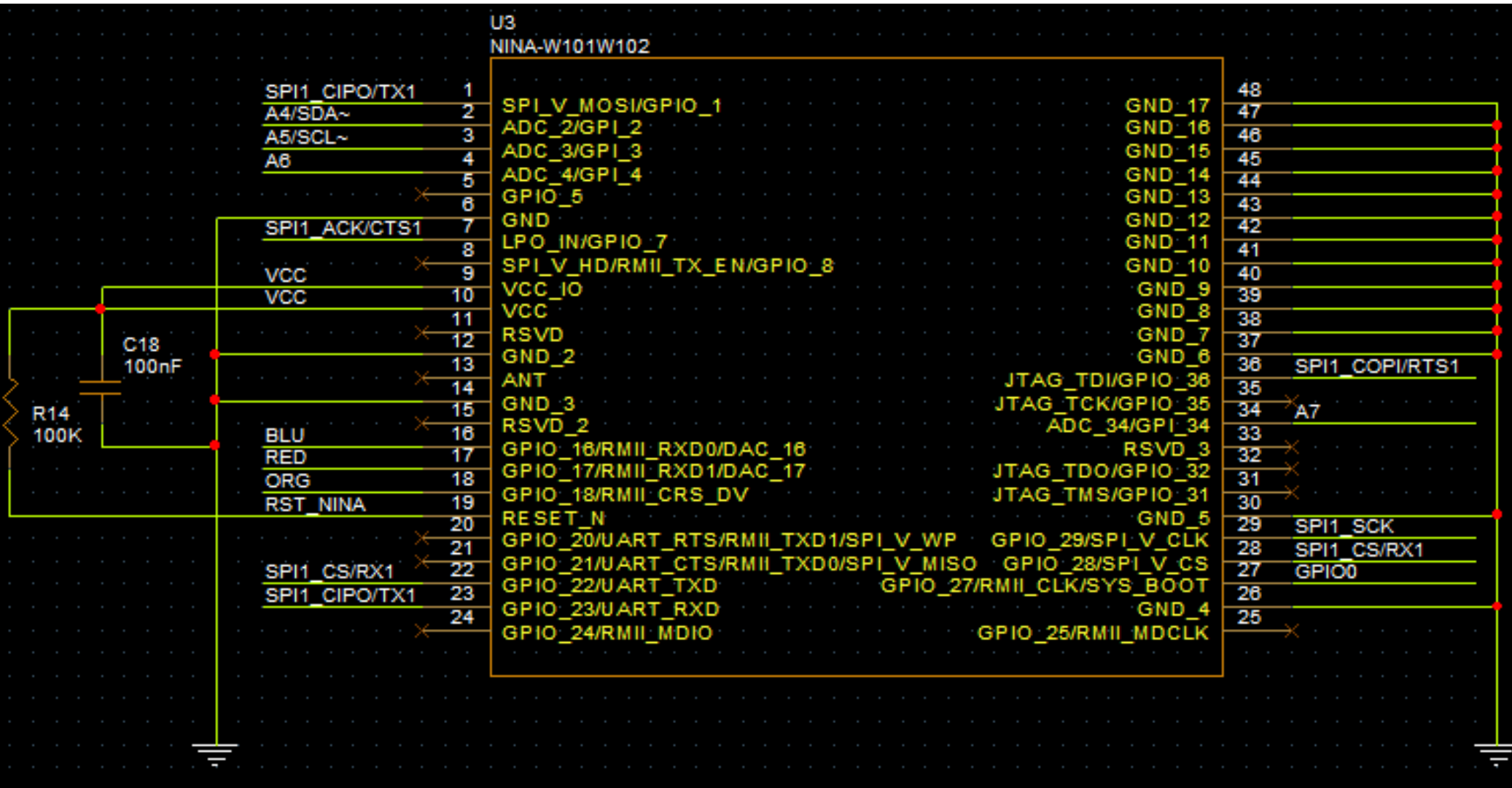
⚠ Installation will overwrite the Sketch on the board.

Ln 1, Col 1 × No board selected

Give Life to the Monster



Arduino BLE Setup

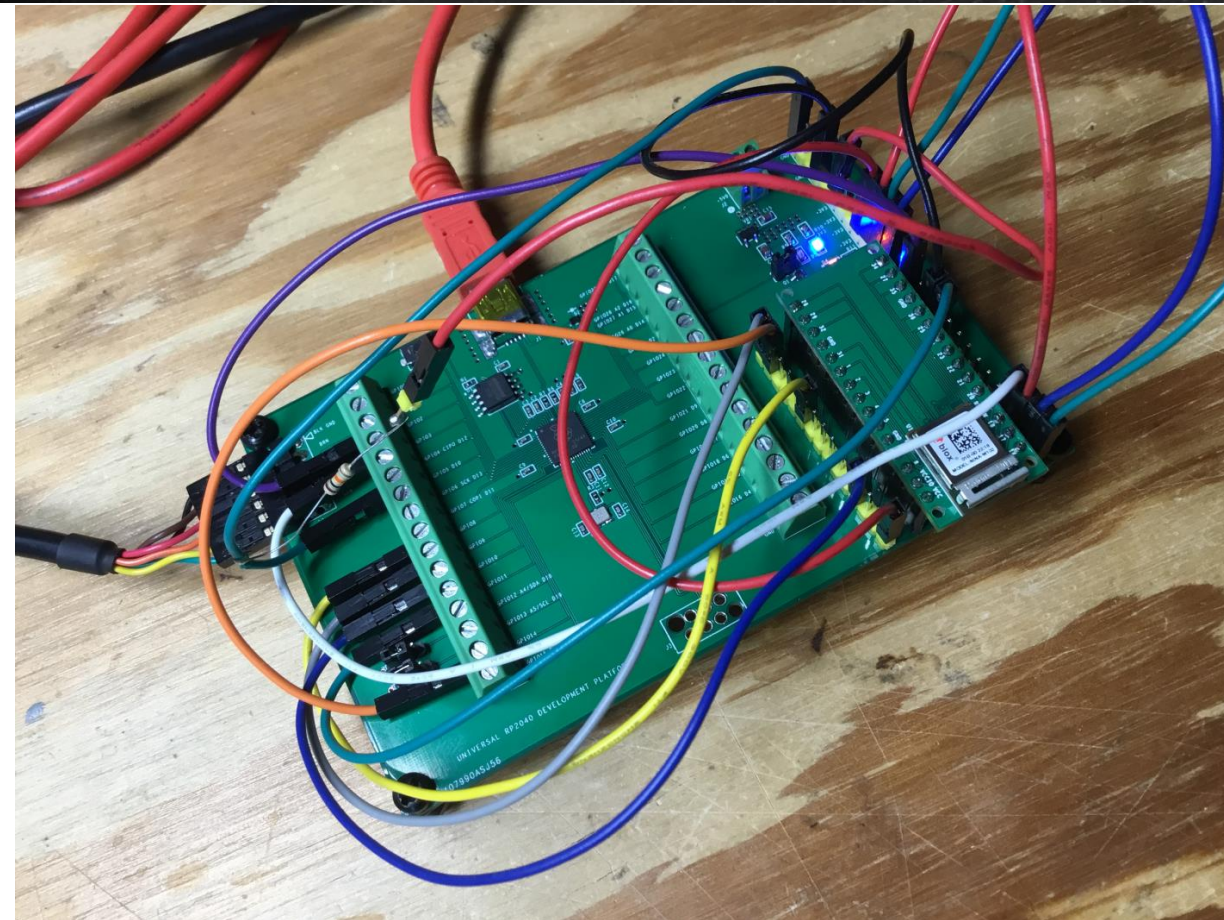


Arduino BLE Setup

```

20 #include <ArduinoBLE.h>
21
22 const int ledPin = LED_BUILTIN; // set ledPin to on-board LED (D2 - RP2050 GPIO25)
23 const int buttonPin = 10;      // set buttonPin to digital pin 10 (D10 - RP2050 GPIO5)
24
25 BLEService ledService("19B10010-E8F2-537E-4F6C-D104768A1214"); // create service
26 // create switch characteristic and allow remote device to read and write
27 BLEByteCharacteristic ledCharacteristic("19B10011-E8F2-537E-4F6C-D104768A1214", BLERead | BLEWrite | BLENotify);
28 // create button characteristic and allow remote device to get notifications
29 BLEByteCharacteristic buttonCharacteristic("19B10012-E8F2-537E-4F6C-D104768A1214", BLERead | BLENotify);
30
31 void setup() {
32   Serial1.begin(9600);
33   while (!Serial1);
34
35   pinMode(ledPin, OUTPUT); // configure the LED pin as an output
36   pinMode(buttonPin, INPUT); // configure the button pin as an input
37
38   // BLE initialization
39   if (!BLE.begin()) {
40     Serial1.println("Starting Bluetooth® - Low Energy module failed!");
41     while (1);
42   }
43
44   // set the local name peripheral advertises
45   BLE.setLocalName("BTNLED");
46   // set the UUID for the service this peripheral advertises:
47   BLE.setAdvertisedService(ledService);
48
49   // add the characteristics to the service
50   ledService.addCharacteristic(ledCharacteristic);
51   ledService.addCharacteristic(buttonCharacteristic);
52
53   // add the service
54   BLE.addService(ledService);
55
56   ledCharacteristic.writeValue(0);
57   buttonCharacteristic.writeValue(0);
58
59   // start advertising
60   BLE.advertise();
61
62   Serial1.println("Bluetooth® device active, waiting for connections...");
63 }

```



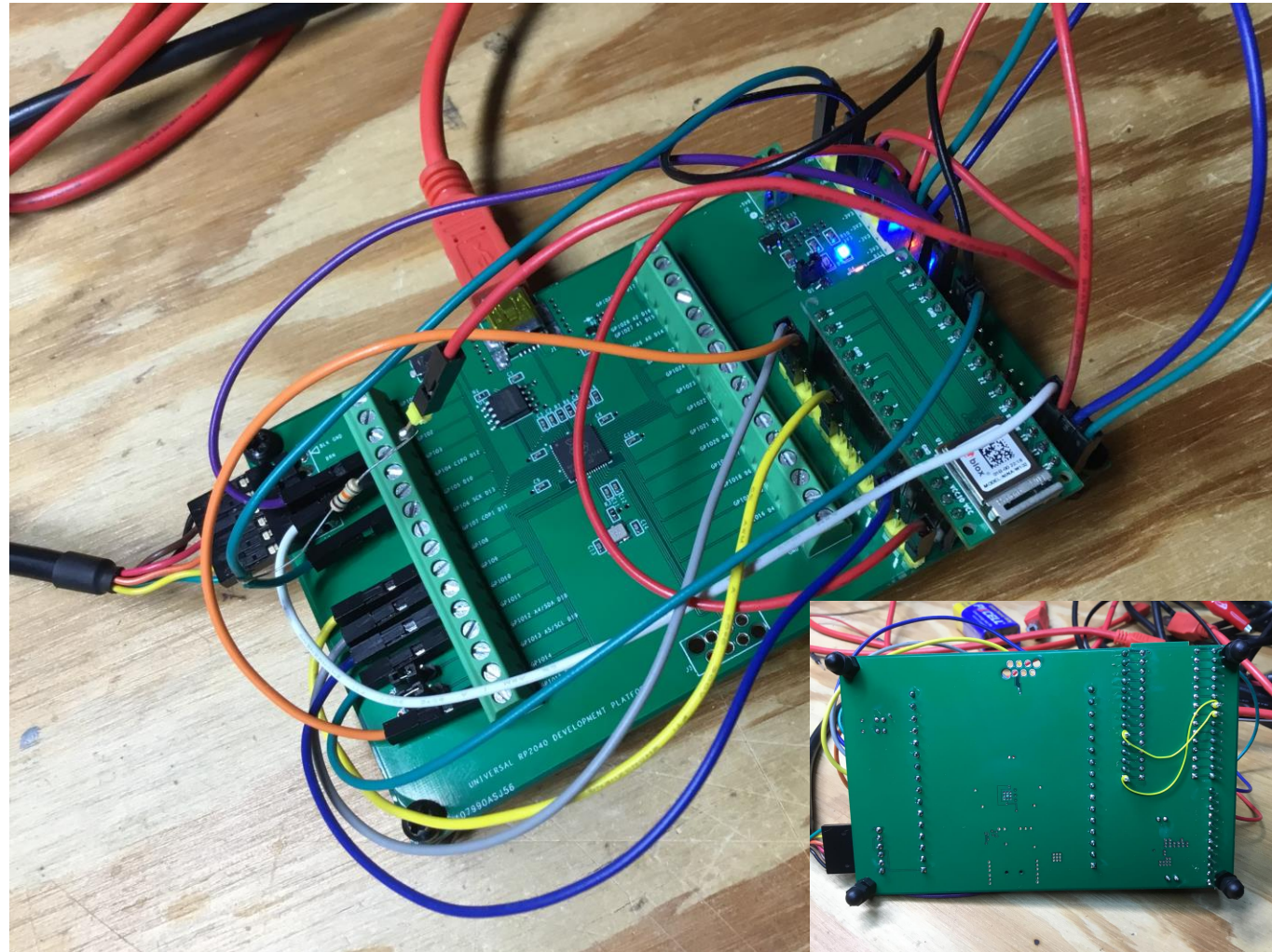
```

22 // D8 - D13
23 { p20, NULL, NULL, NULL }, // D8
24 { p21, NULL, NULL, NULL }, // D9
25 { p5, NULL, NULL, NULL }, // D10 ← buttonPin
26 { p7, NULL, NULL, NULL }, // D11 / SPITX
27 { p4, NULL, NULL, NULL }, // D12 / SPIRX
28 { p6, NULL, NULL, NULL }, // D13 / SPICLK / LEDB

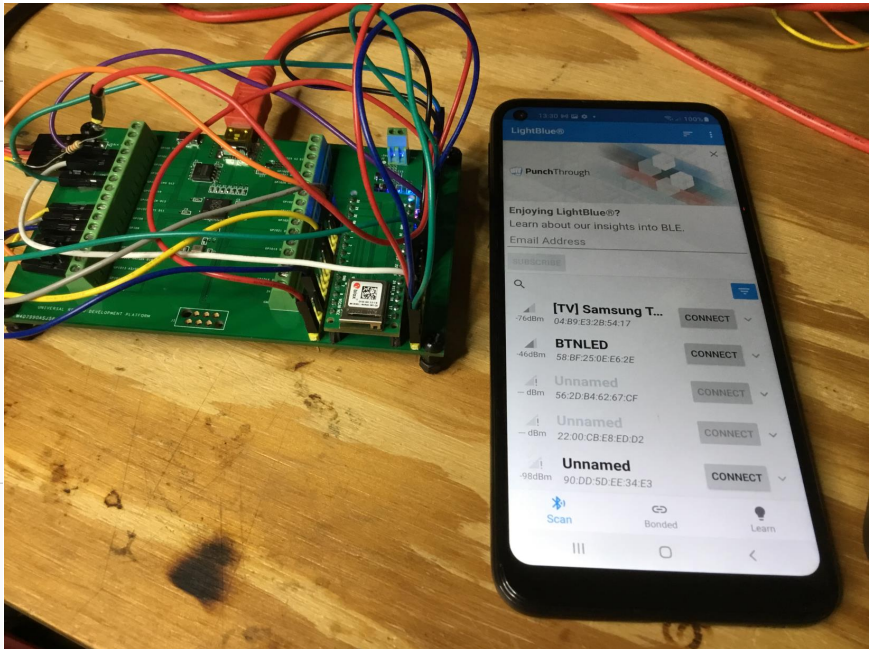
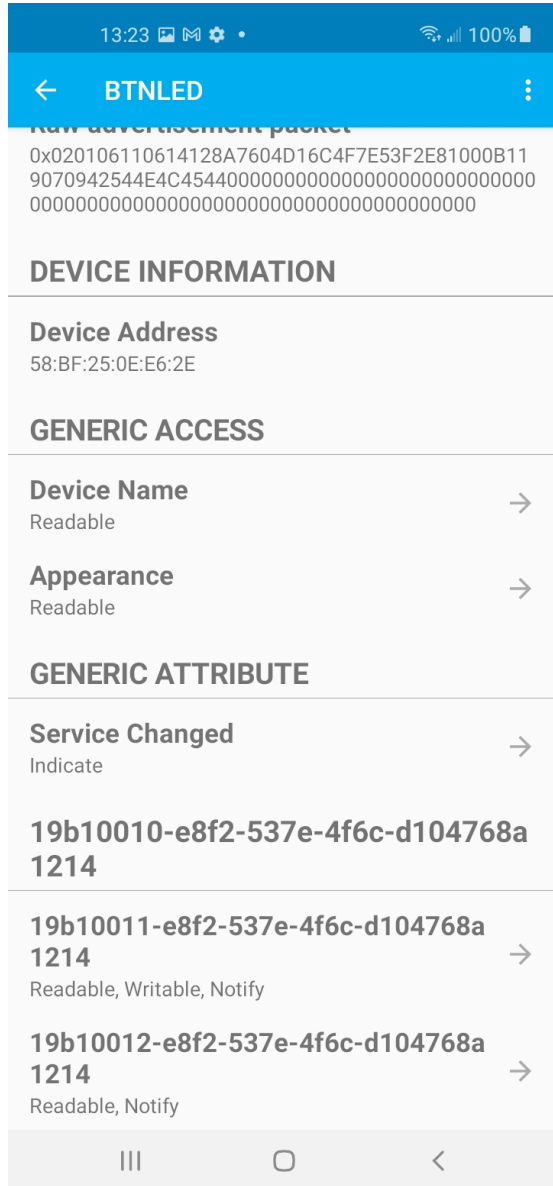
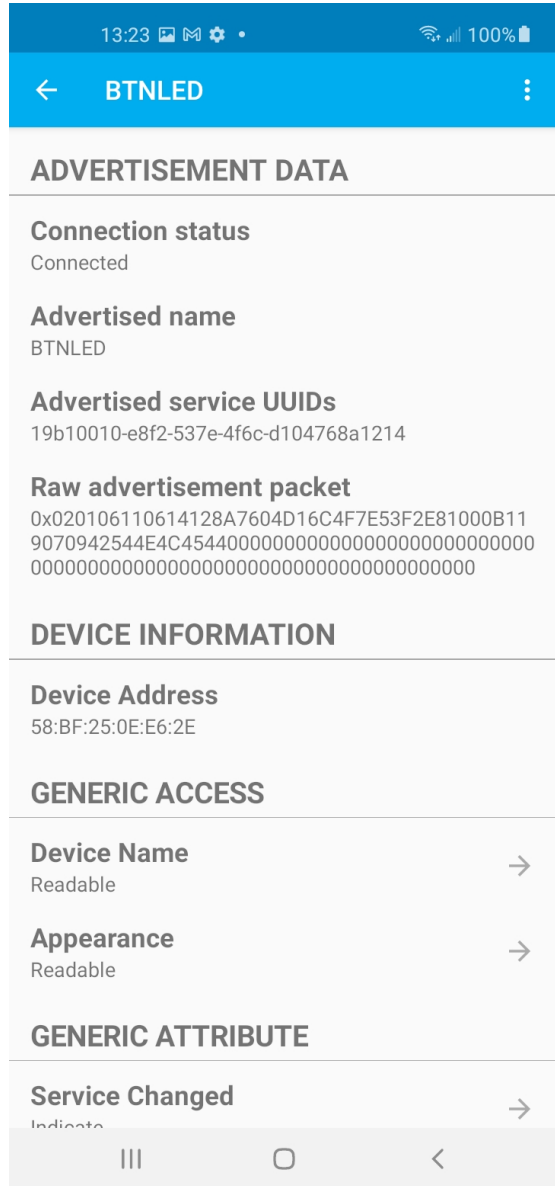
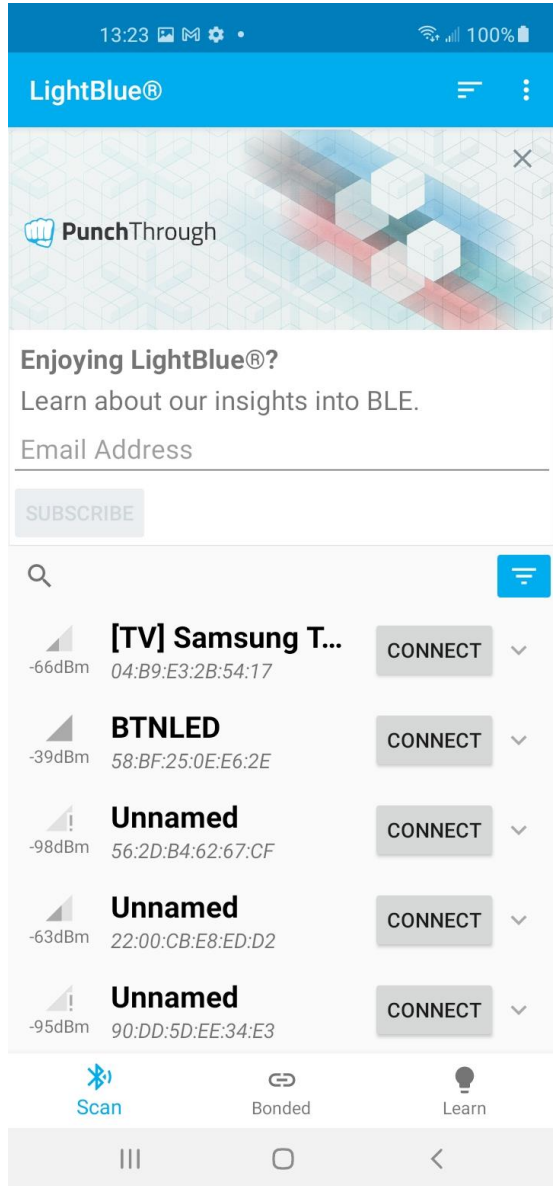
```

Arduino BLE Loop

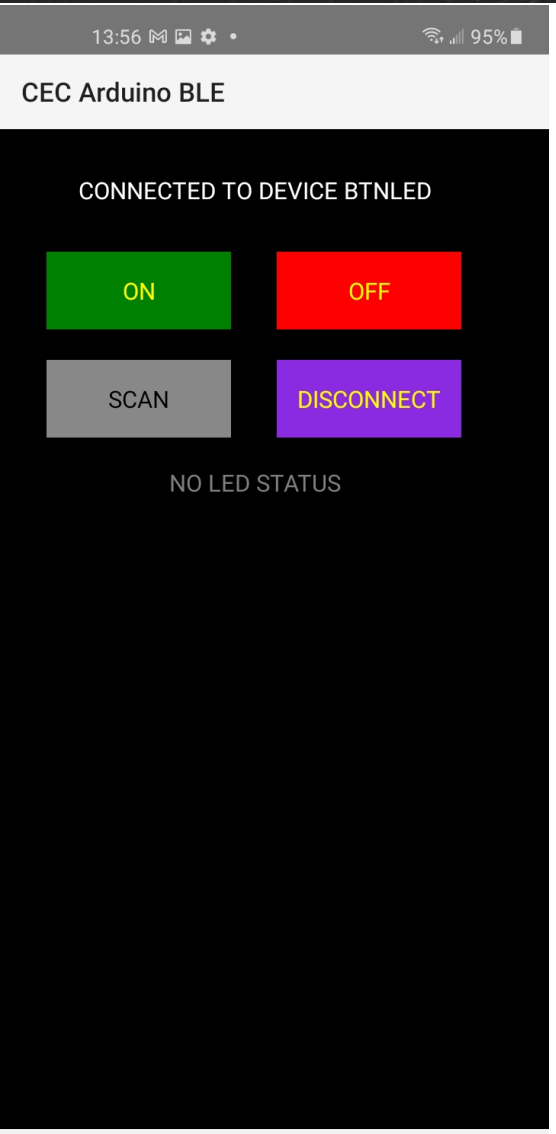
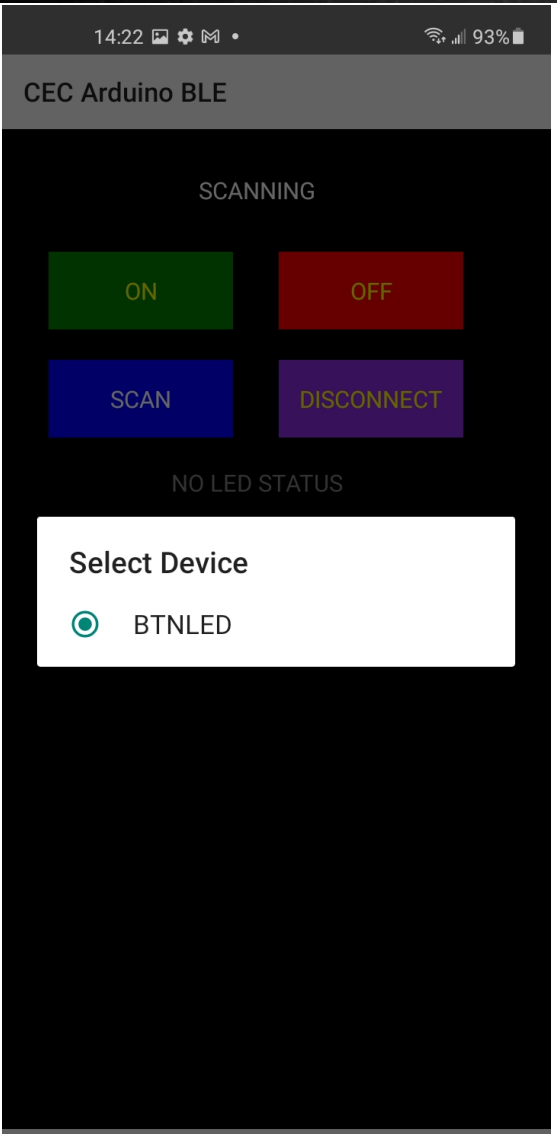
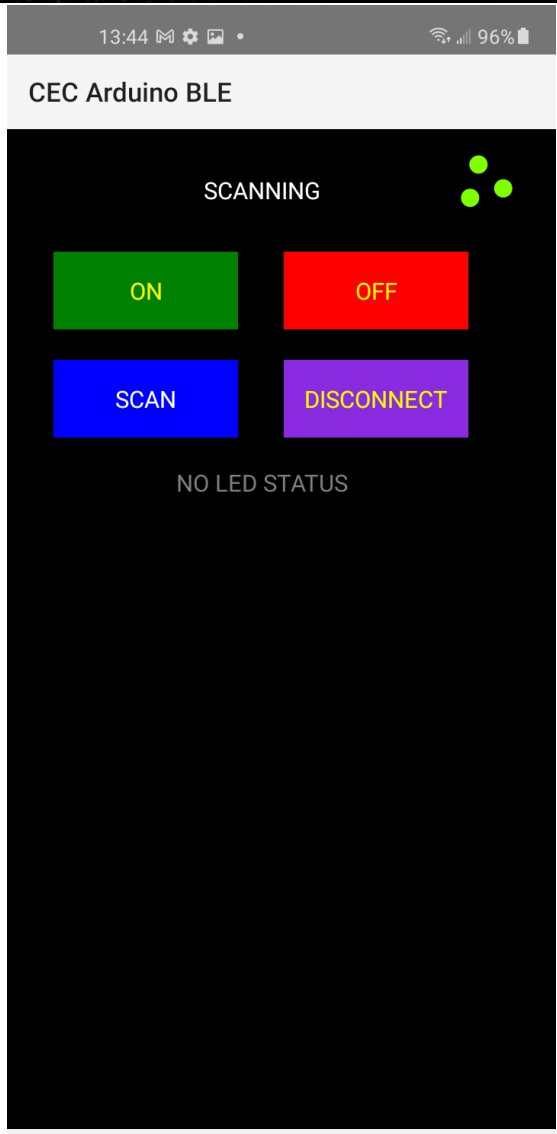
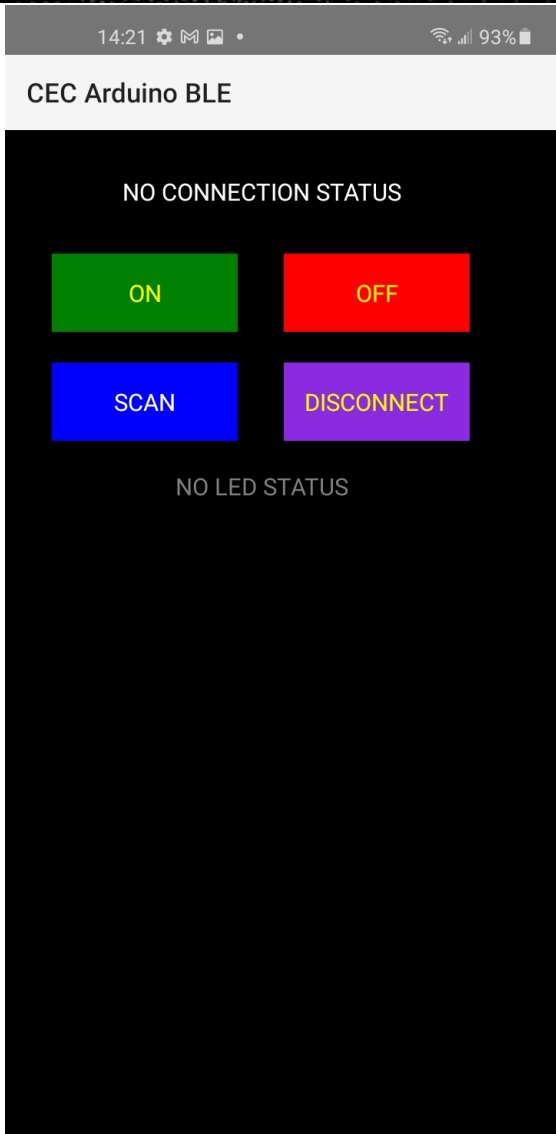
```
65 void loop() {
66     // Poll for Bluetooth® Low Energy events
67     BLE.poll();
68
69     // read the current button pin state
70     char buttonValue = digitalRead(buttonPin);
71
72     // has the value changed since the last read
73     bool buttonChanged = (buttonCharacteristic.value() != buttonValue);
74
75     if (buttonChanged) {
76         // button state changed, update characteristics
77         ledCharacteristic.writeValue(buttonValue);
78         buttonCharacteristic.writeValue(buttonValue);
79     }
80
81     if (ledCharacteristic.written() || buttonChanged) {
82         // central has written to characteristic
83         // or button state has changed..Update LED
84         if (ledCharacteristic.value()) {
85             Serial1.println("LED on");
86             digitalWrite(ledPin, HIGH);
87         } else {
88             Serial1.println("LED off");
89             digitalWrite(ledPin, LOW);
90         }
91     }
92 }
```



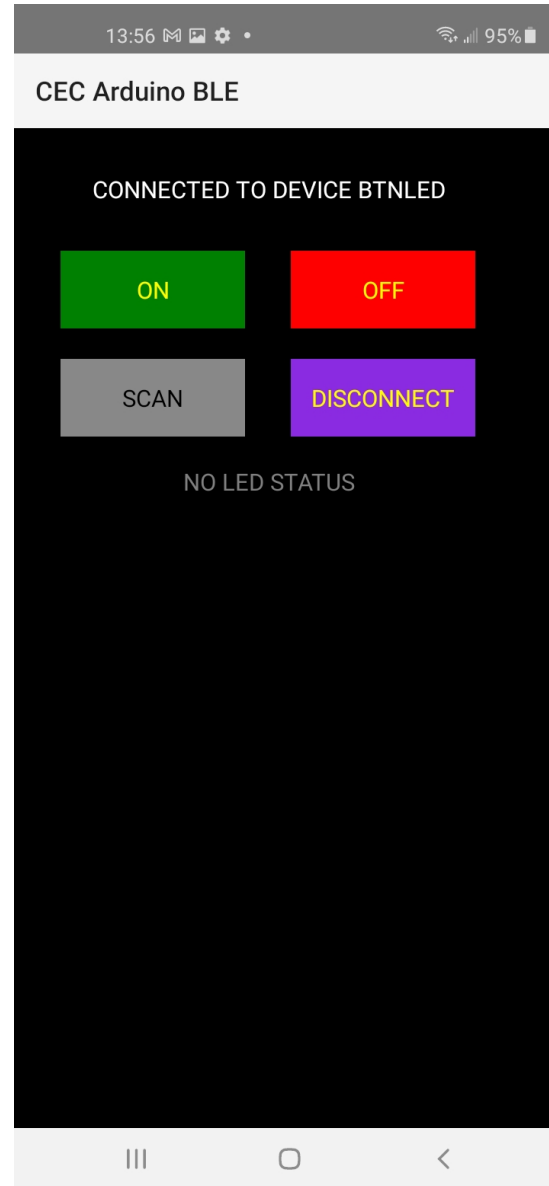
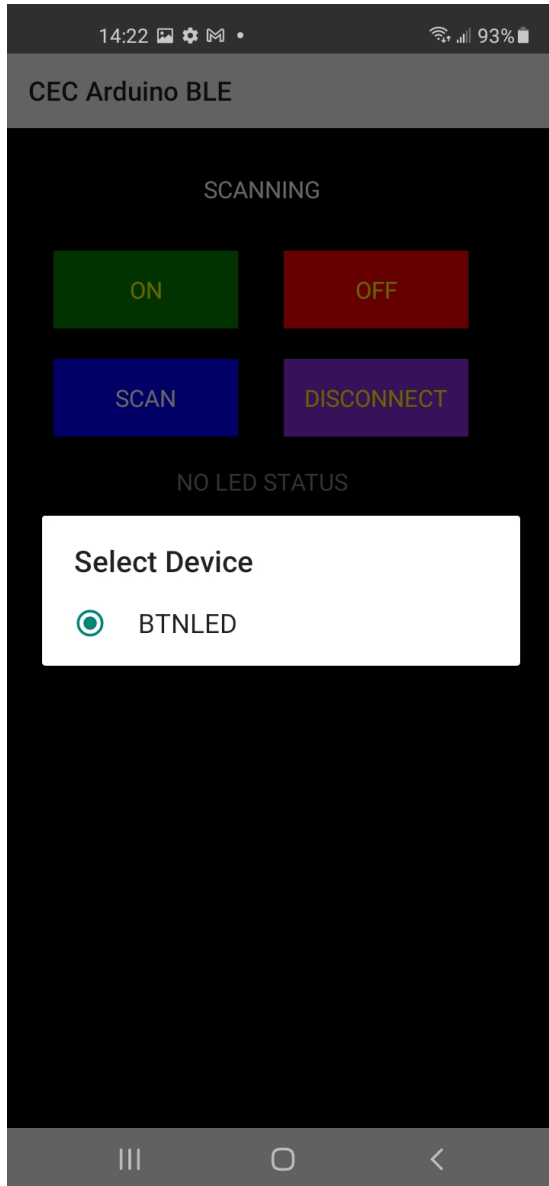
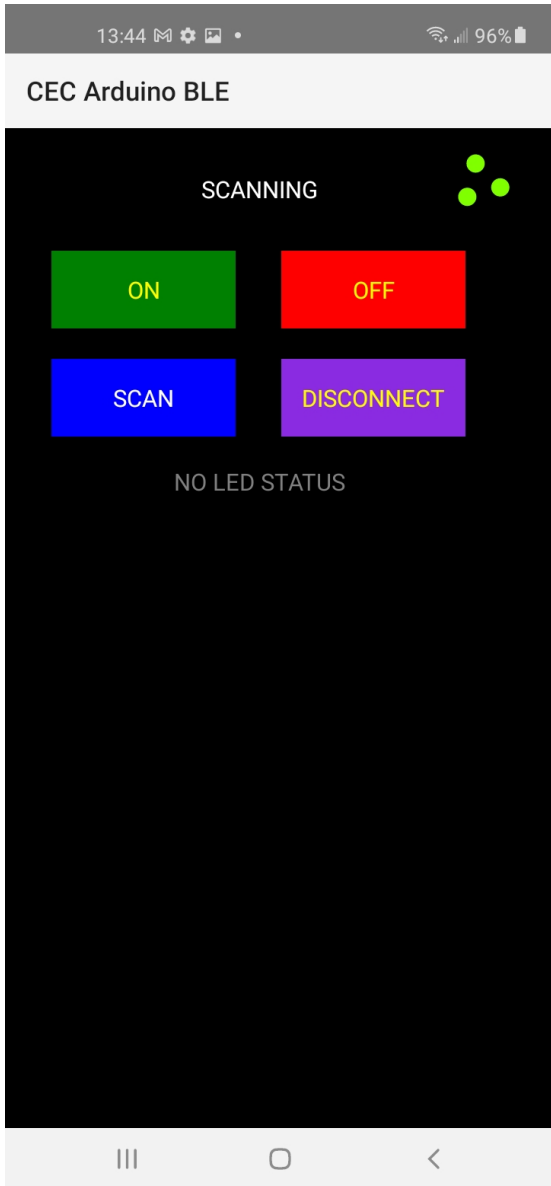
Arduino BLE First Contact



Arduino/B4A BLE Remote Control App



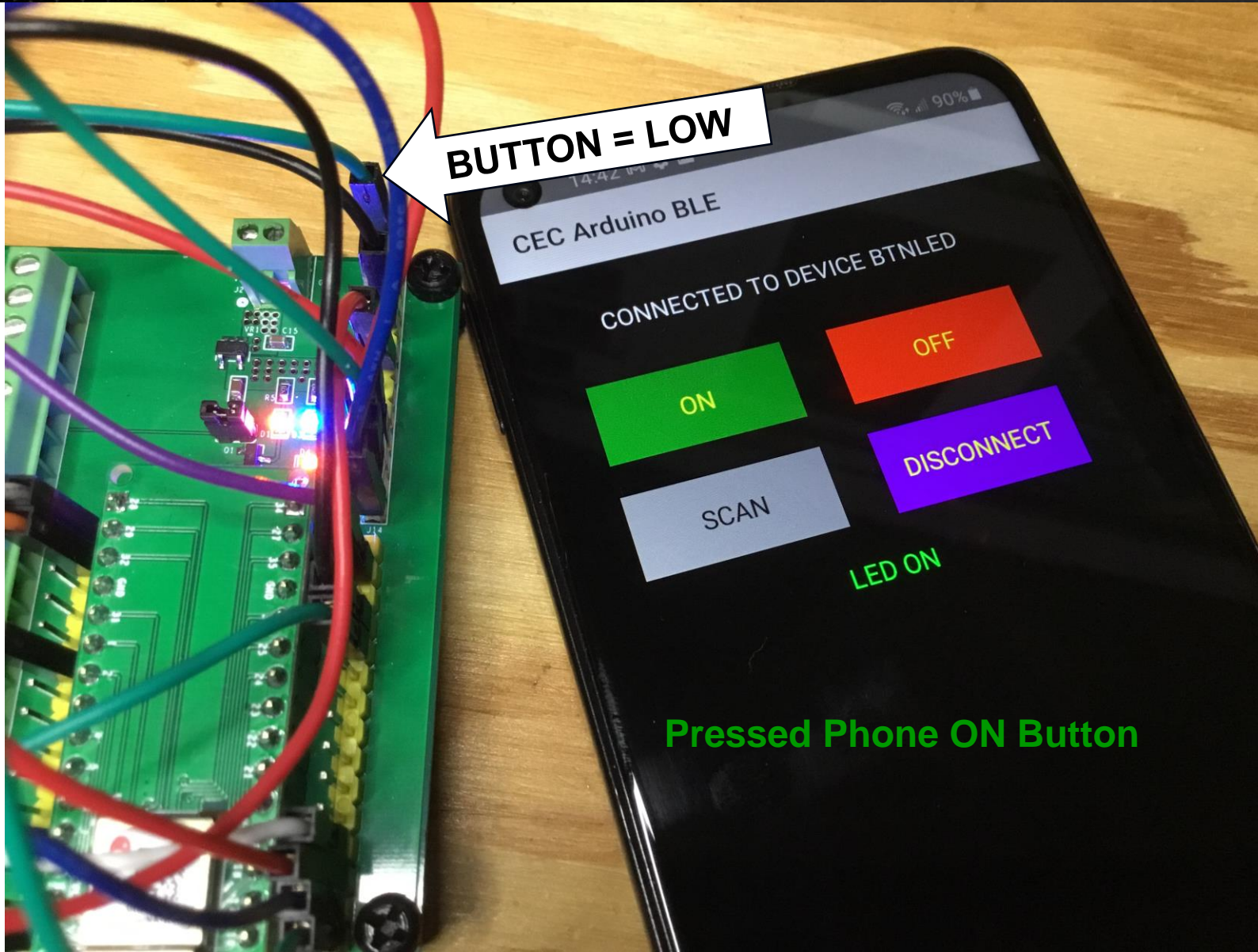
Arduino/B4A BLE Remote Control App



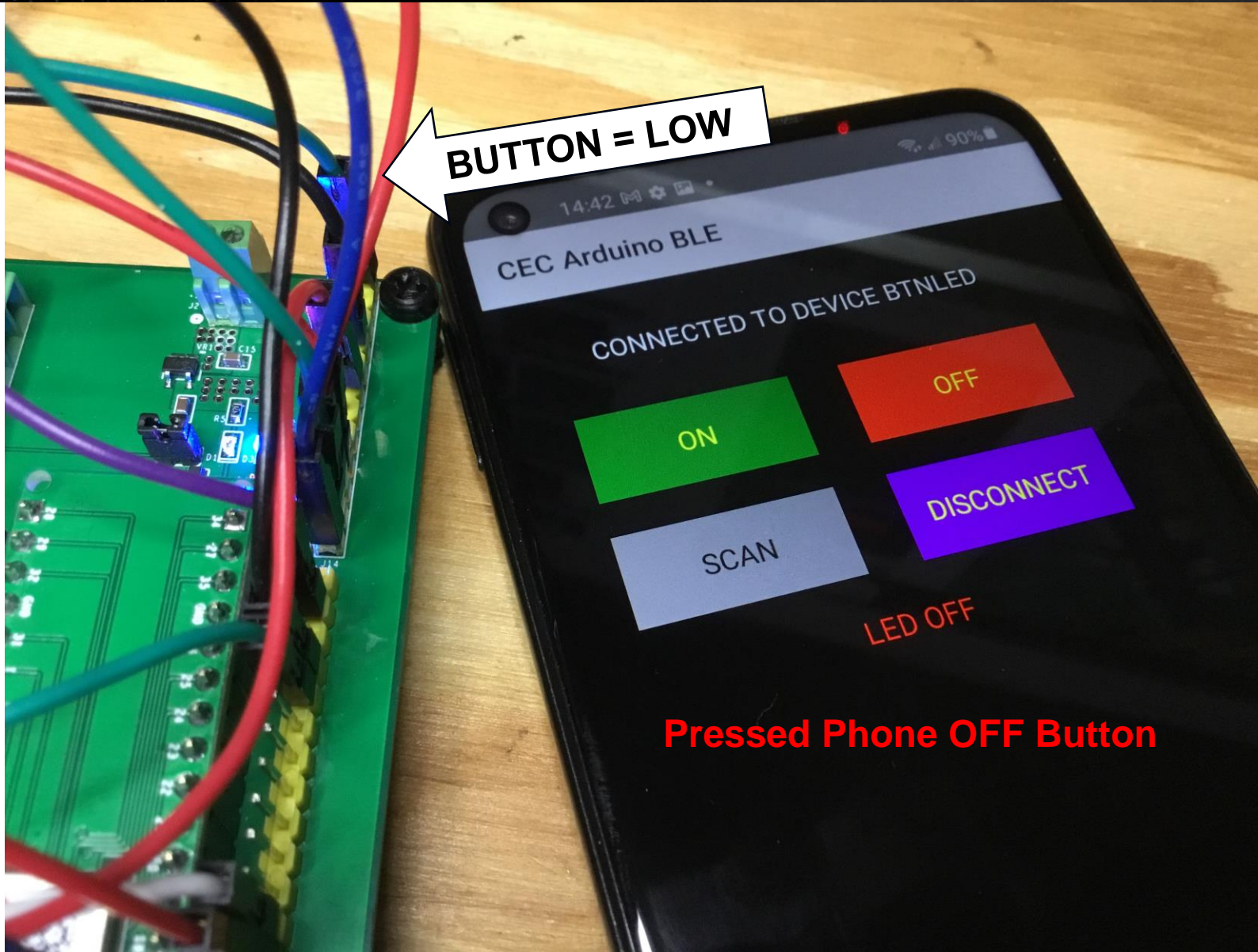
```

➤ Device found:
➤ Device found: [TV] Samsung TU700D 43 TV
➤ Device found: BTNLED
➤ Device found:
➤ Device found:
➤ Device found:
➤ Device found:
➤ Device found:
➤ Device found:
➤ Device found:
➤ Connect to: 58:BF:25:0E:E6:2E
Discovering services.
➤ Connected
Setting descriptor. Success = true
writing descriptor: true
➤ UUID: 00001800-0000-1000-8000-00805f9b34fb
➤ UUID: 00001801-0000-1000-8000-00805f9b34fb
➤ UUID: 19b10010-e8f2-537e-4f6c-d104768a1214
    
```

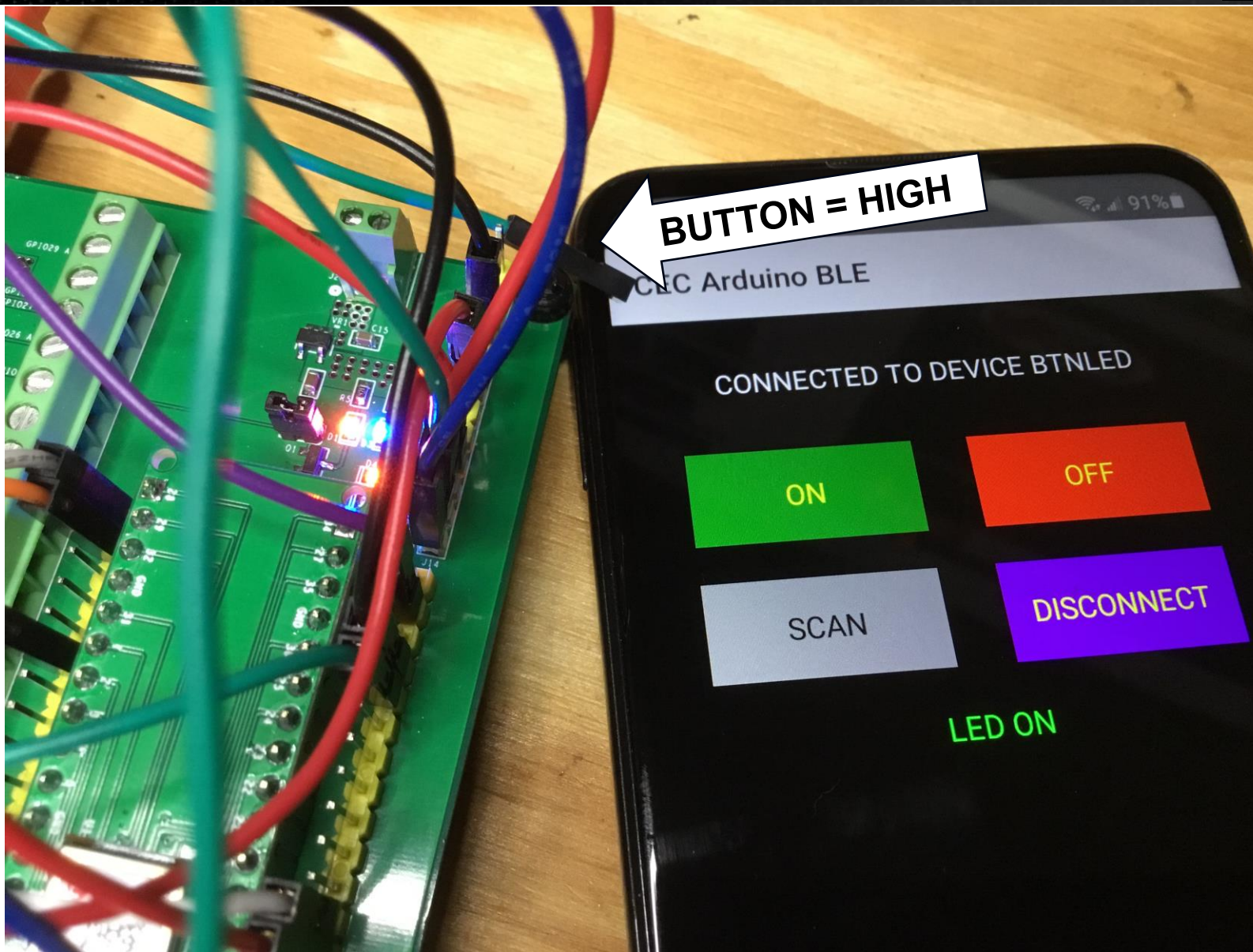
Arduino/B4A BLE Remote Control App



Arduino/B4A BLE Remote Control App



Arduino/B4A BLE Remote Control App

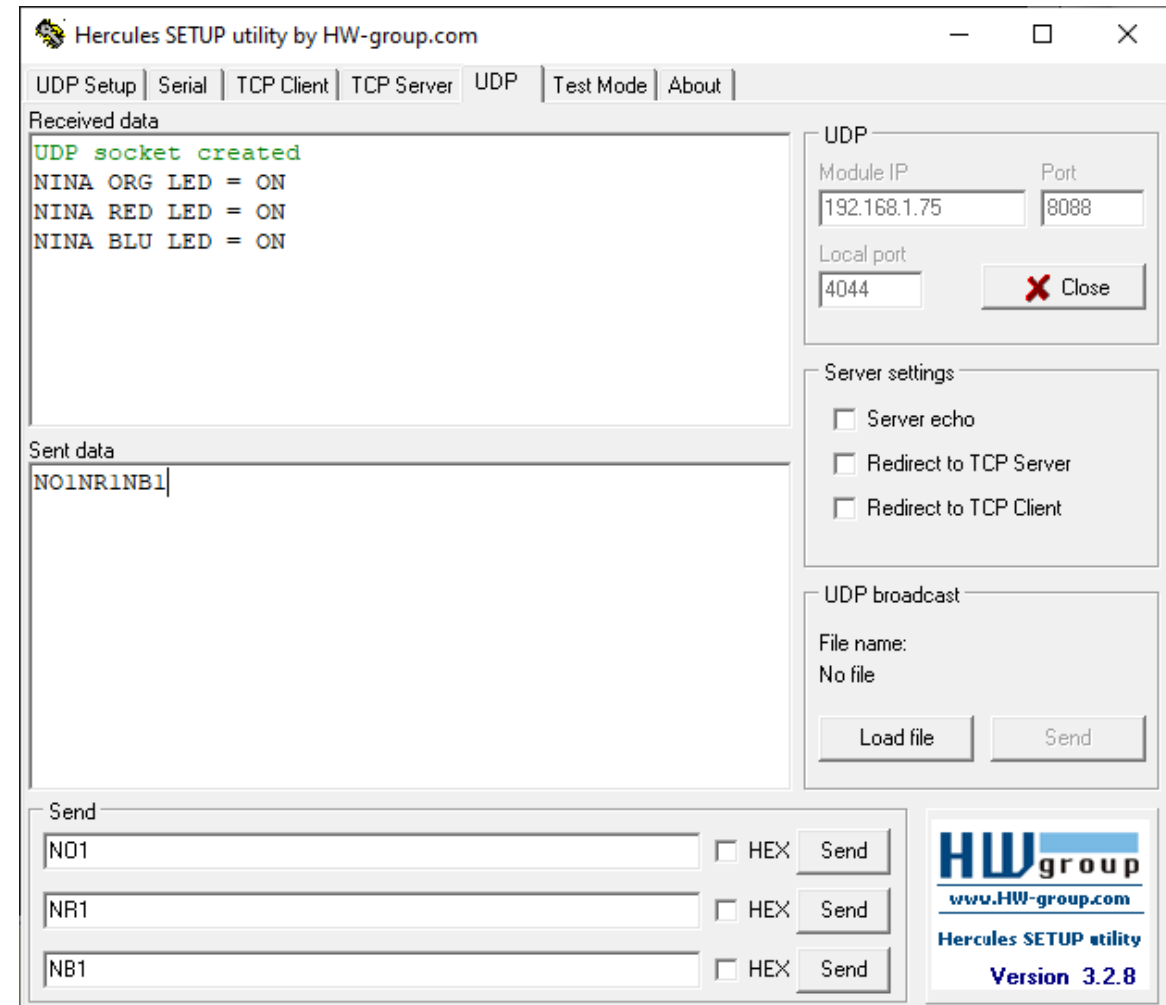


Thank you for attending!!!

Please consider the resources below:

- arduino.cc
- b4x.com
- raspberrypi.org
- u-blox.com

MORE TO COME..





Thank You

Sponsored by

