CEC Continuing Education Center

DesignNews

Designing, Building and Coding Custom Raspberry Pi RP2040 Arduino Devices

Day 3:

The RP2040, Arduino and BLE

Sponsored by

DigiKey

informa markets

Webinar Logistics

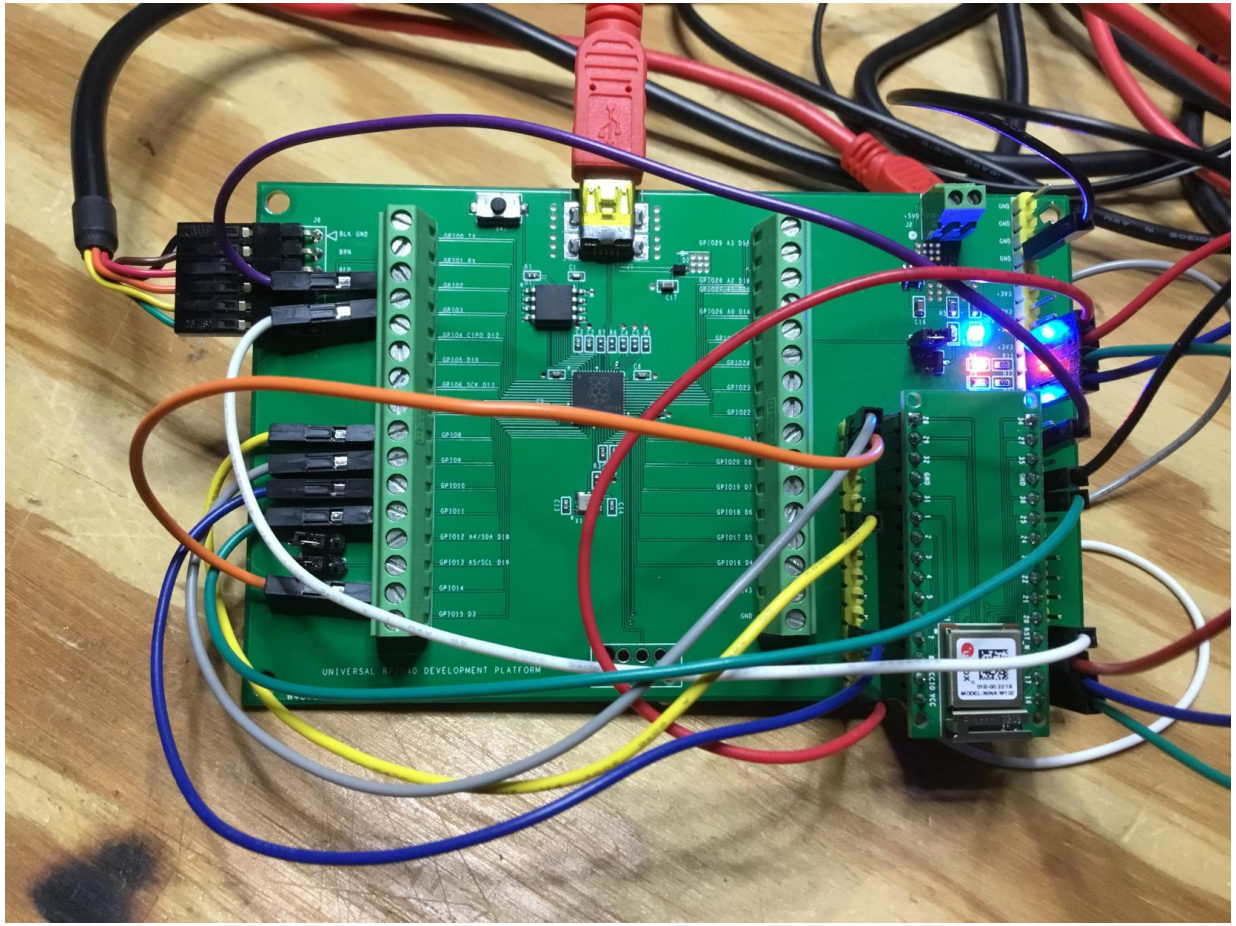
- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.

Fred Eady

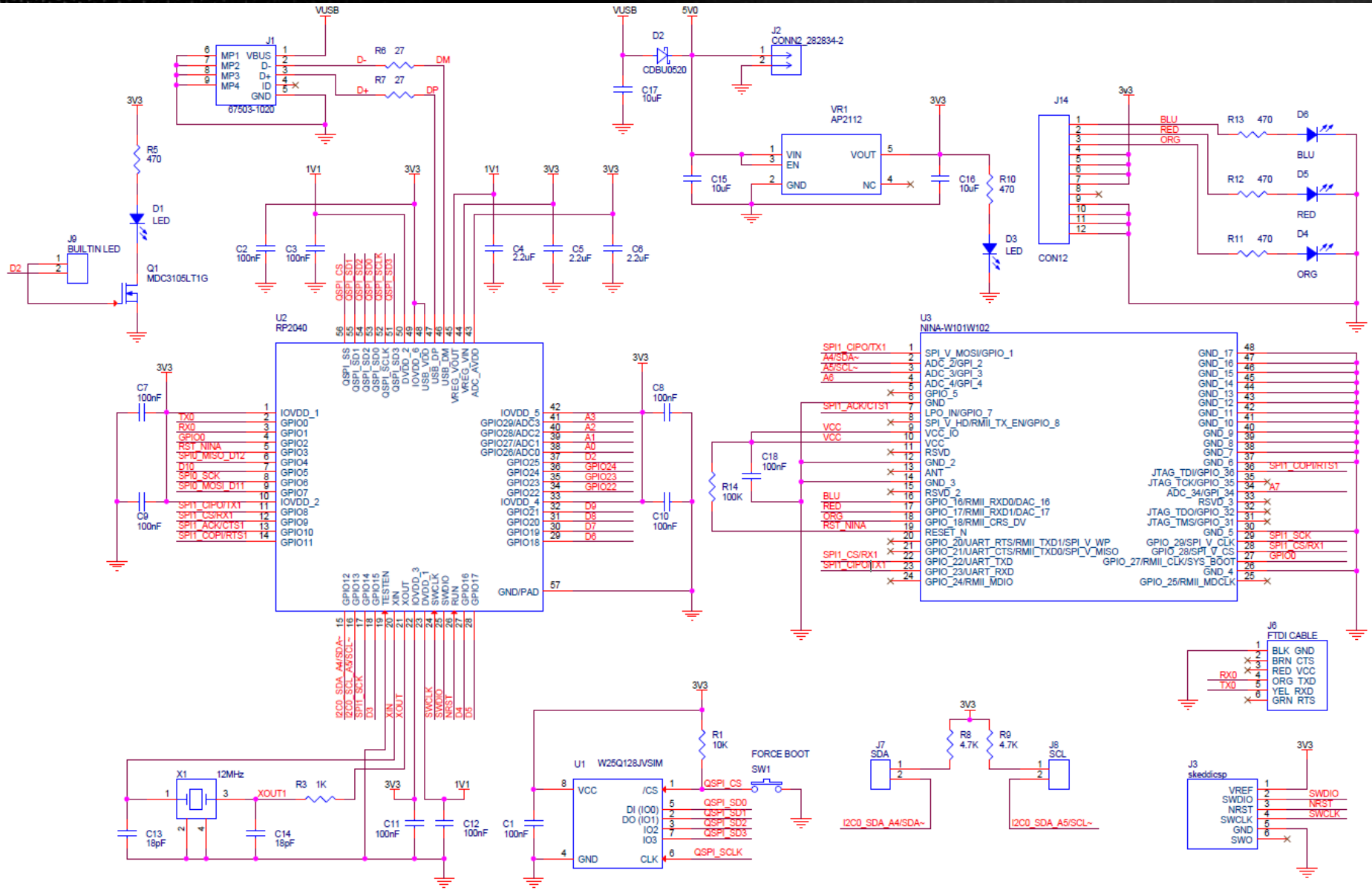
Visit ‘Lecturer Profile’ in your console for more details.

AGENDA

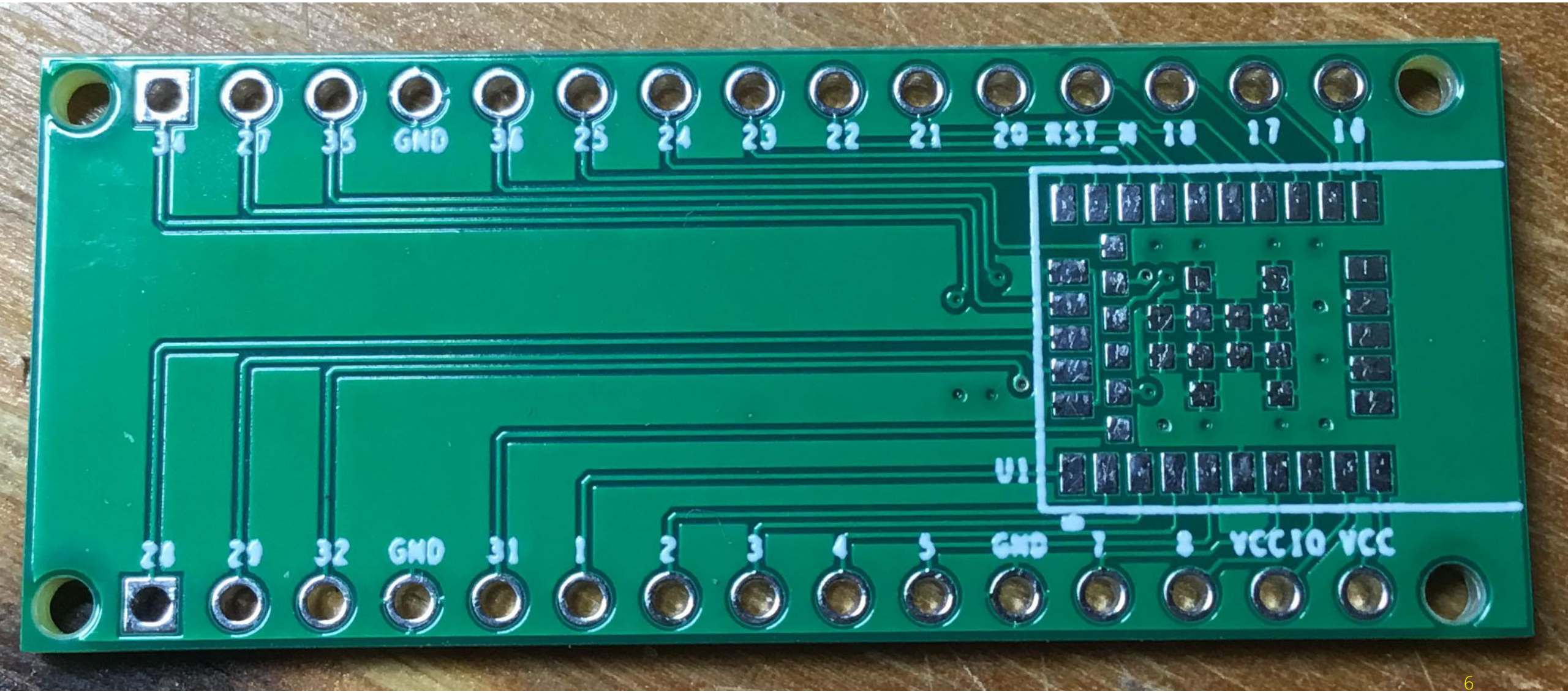
- RP2040 Arduino BLE Hardware
- u-blox Frankenstein
- Arduino BLE On Your Phone



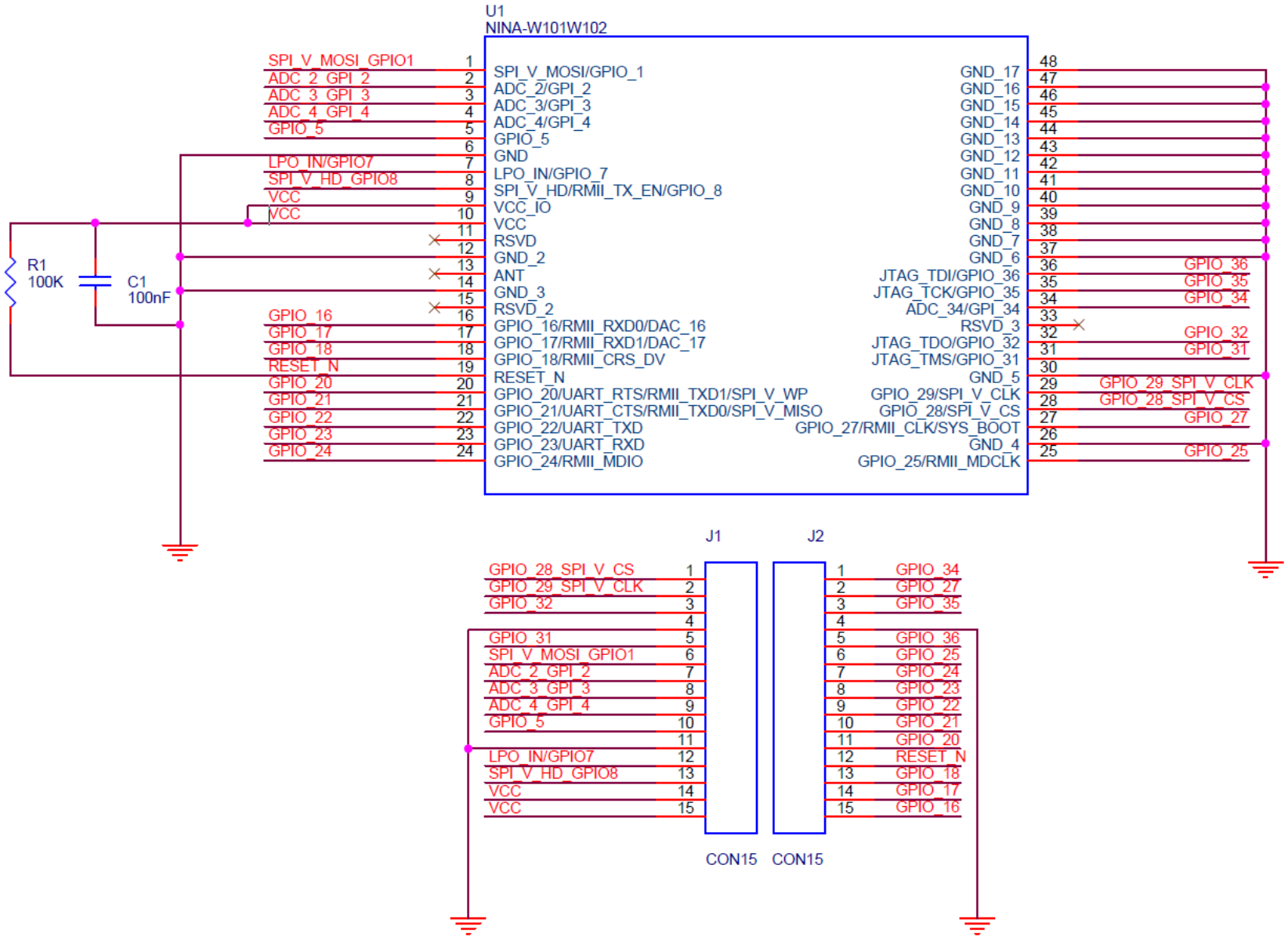
RP2040 Arduino BLE Hardware Design



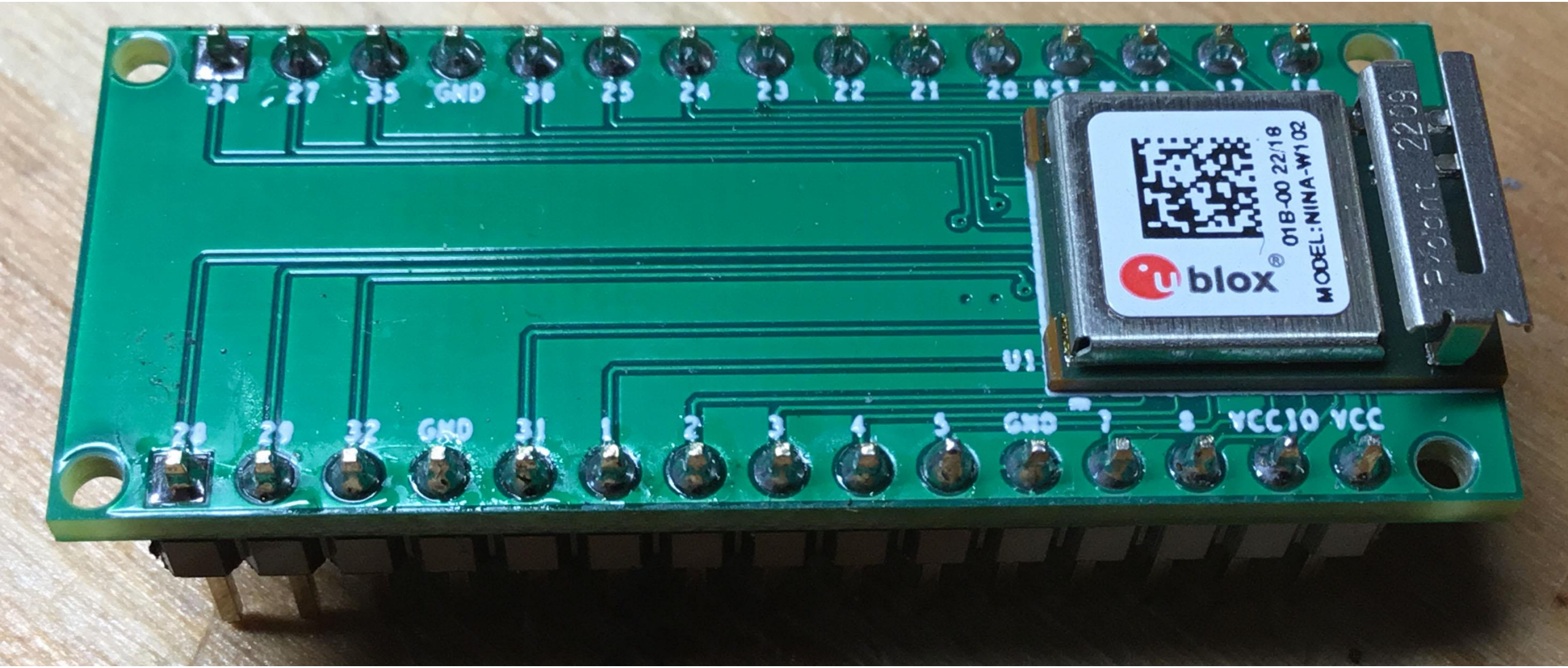
RP2040 Arduino BLE Hardware Design



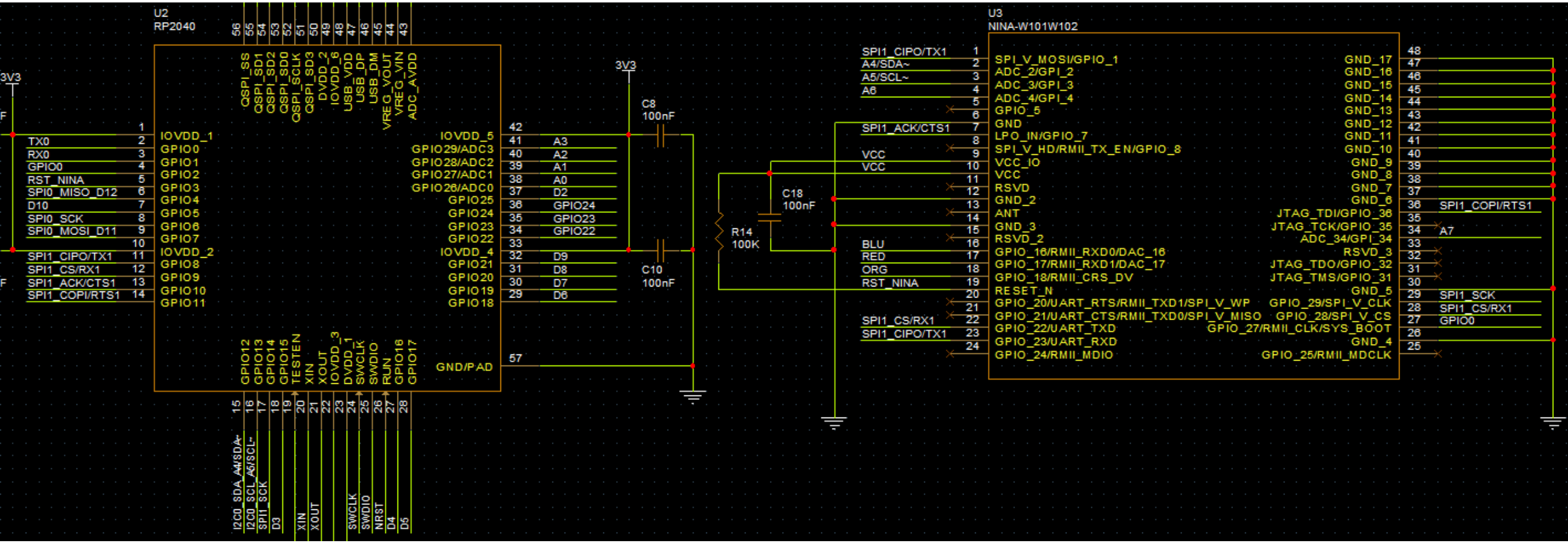
RP2040 Arduino BLE Hardware Design



RP2040 Arduino BLE Hardware Design



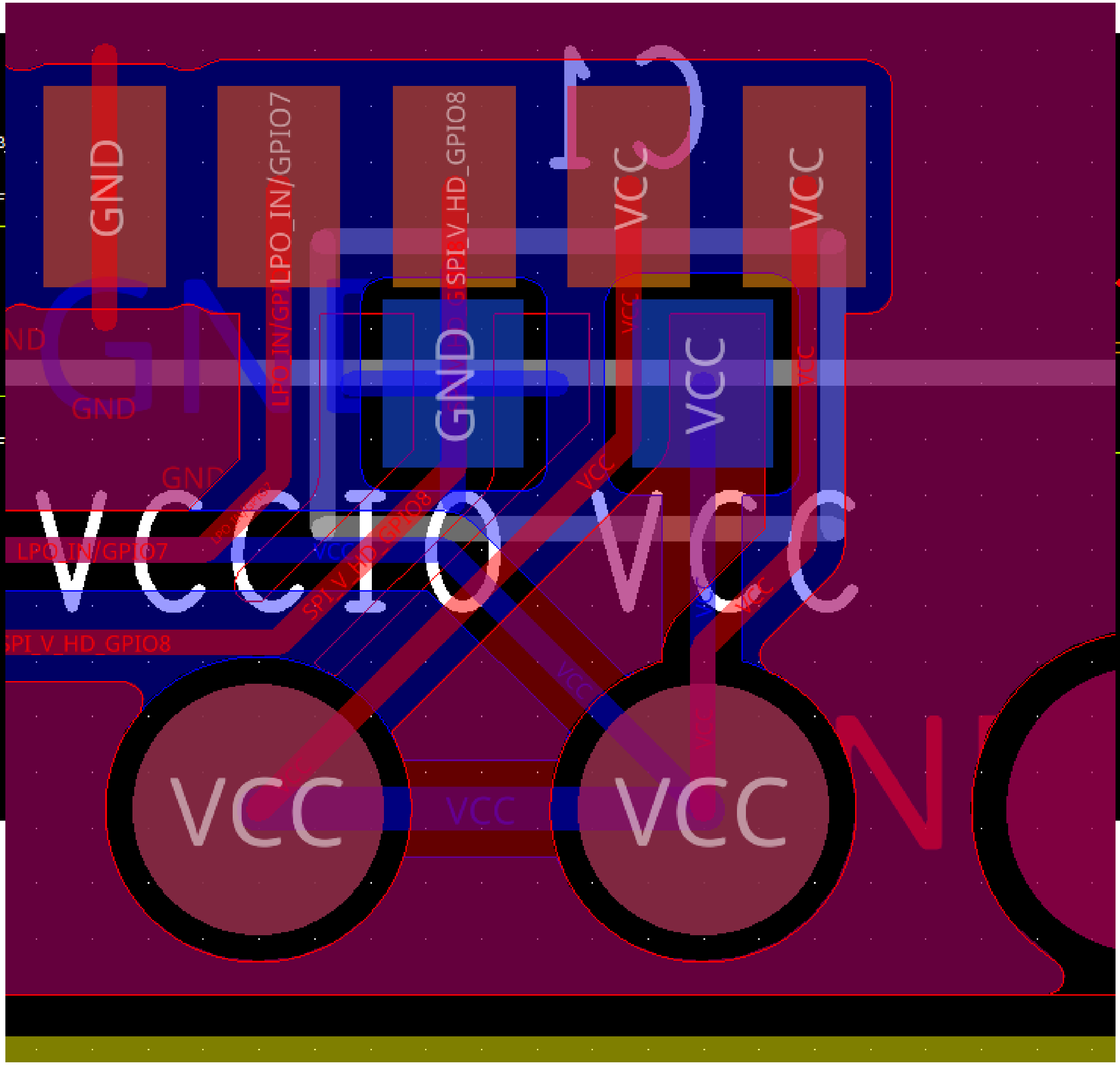
RP2040 Arduino BLE Hardware Design

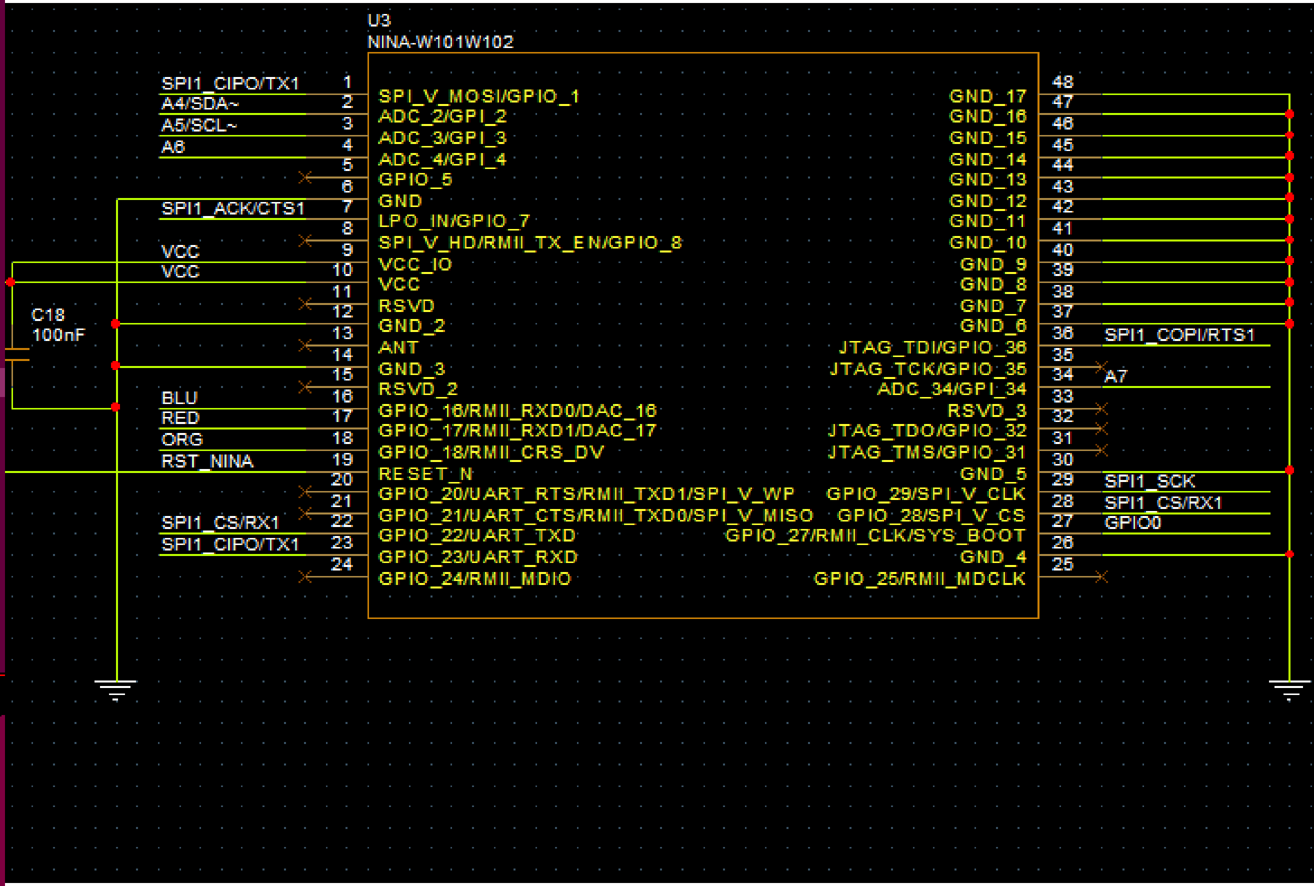


CEC Continuing Education Center

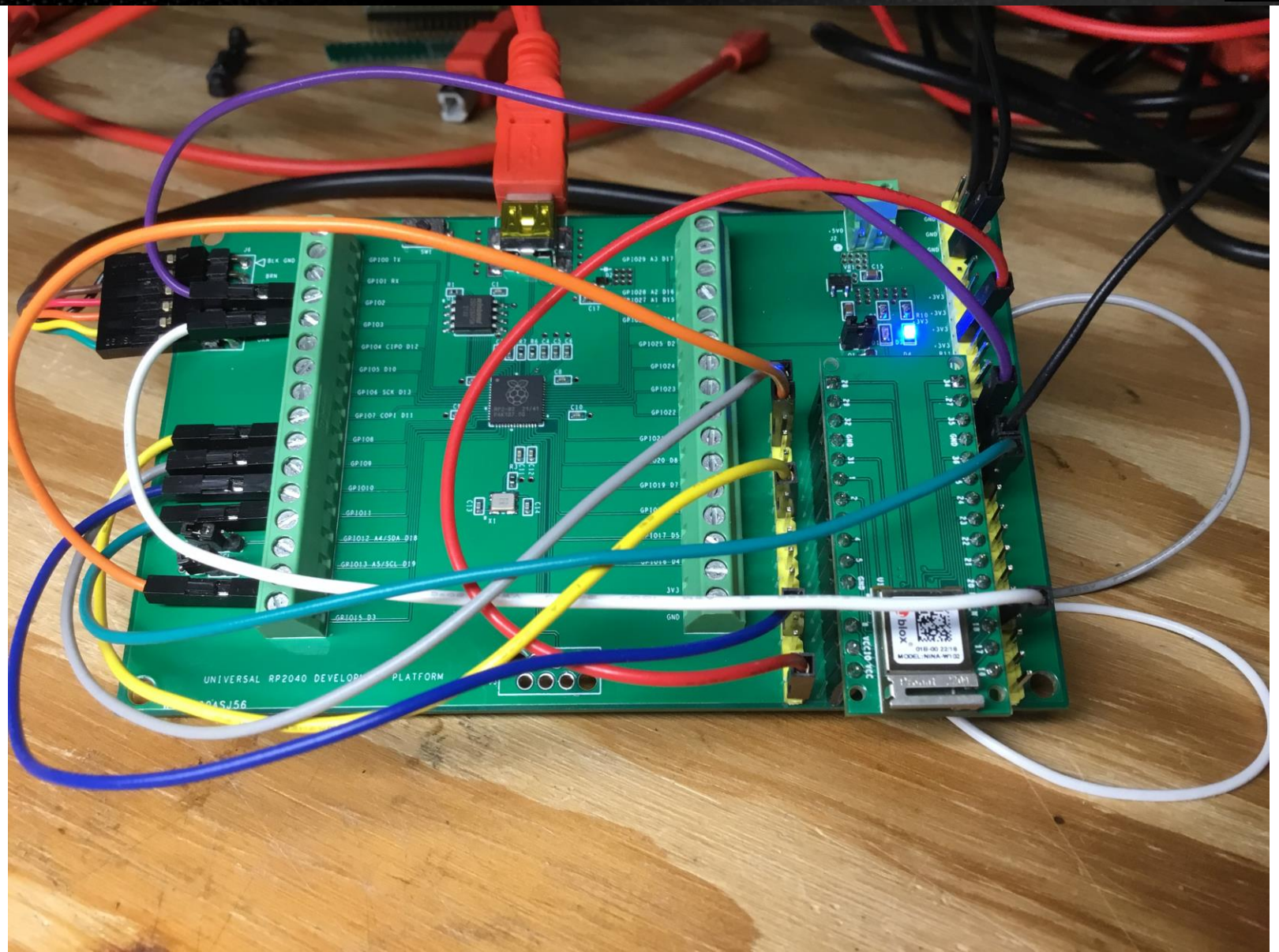
DigiKey

RP2040 Arduino BLE Hardware Design

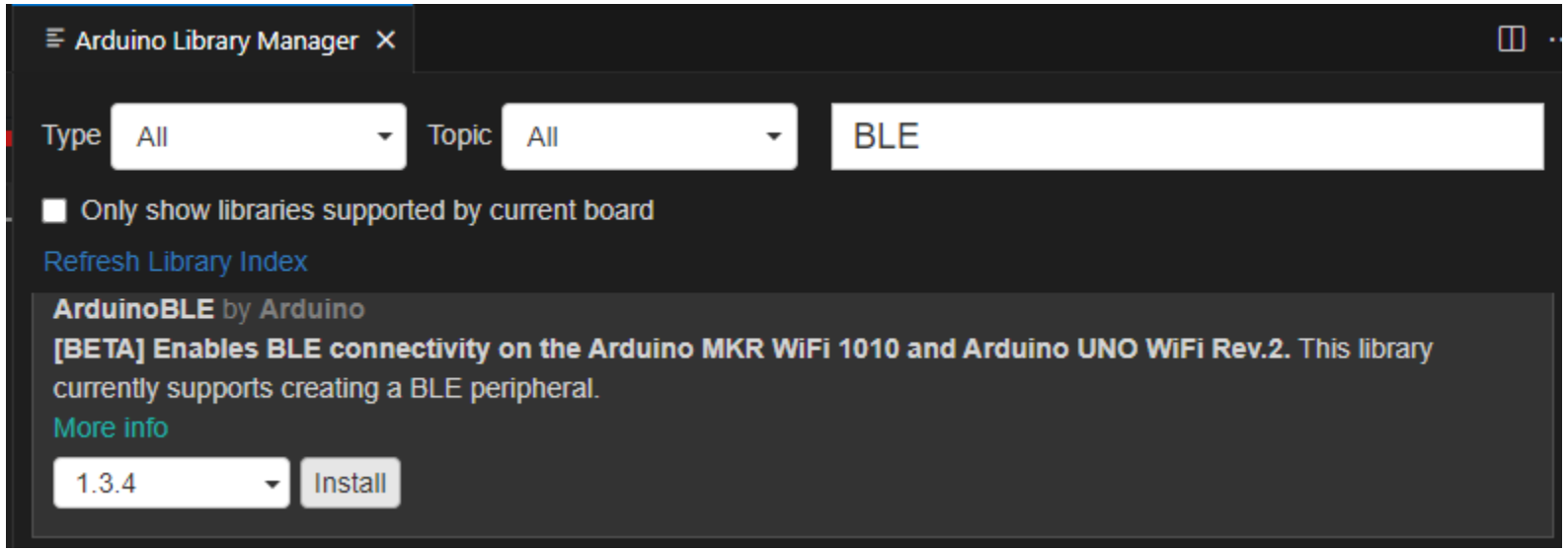


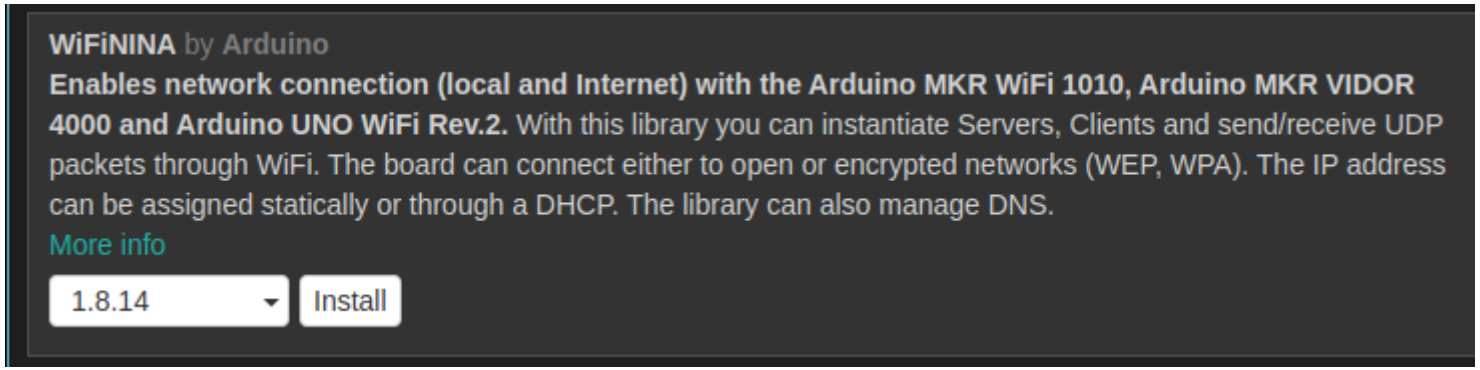


RP2040 Arduino BLE Hardware Design

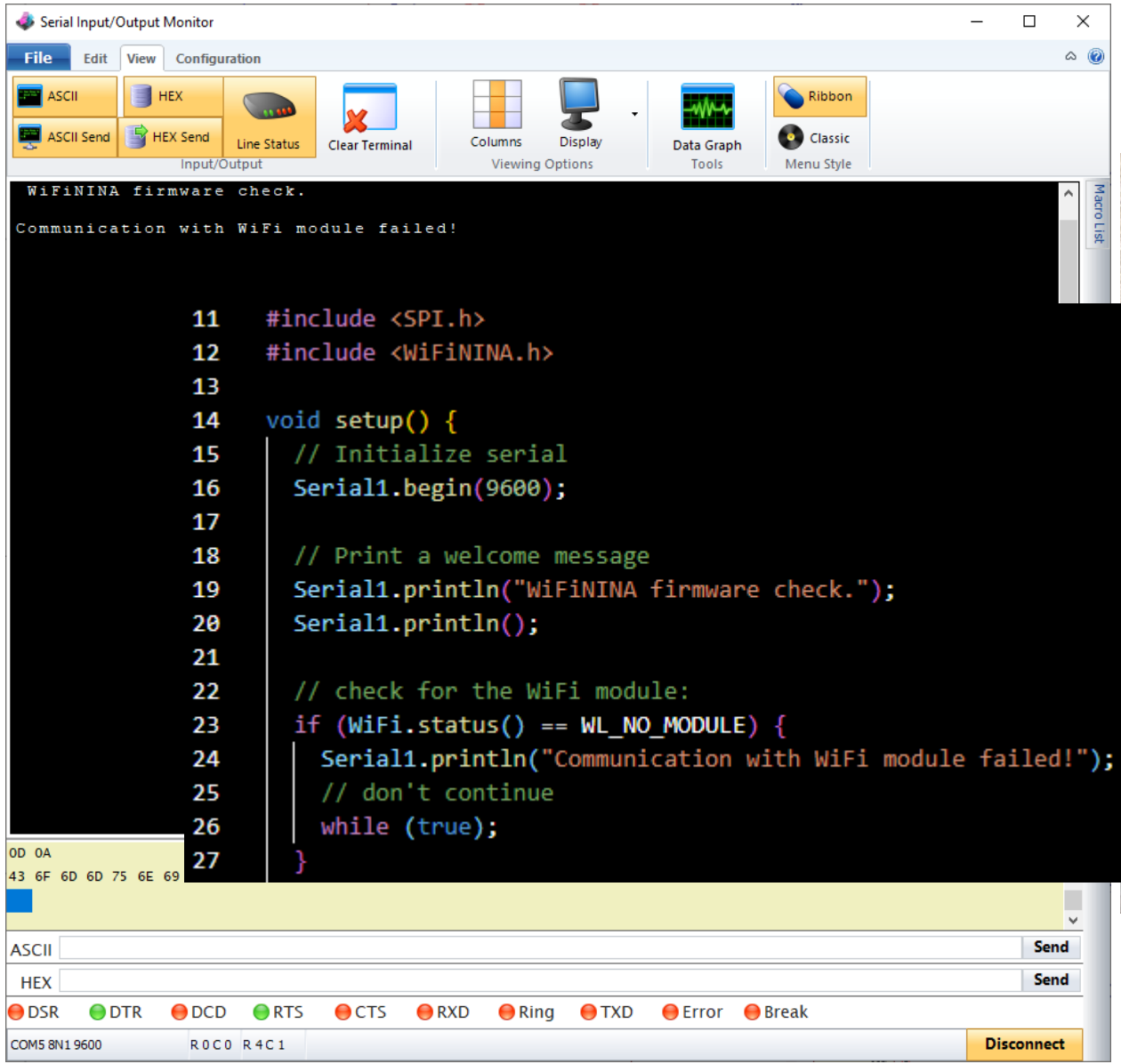


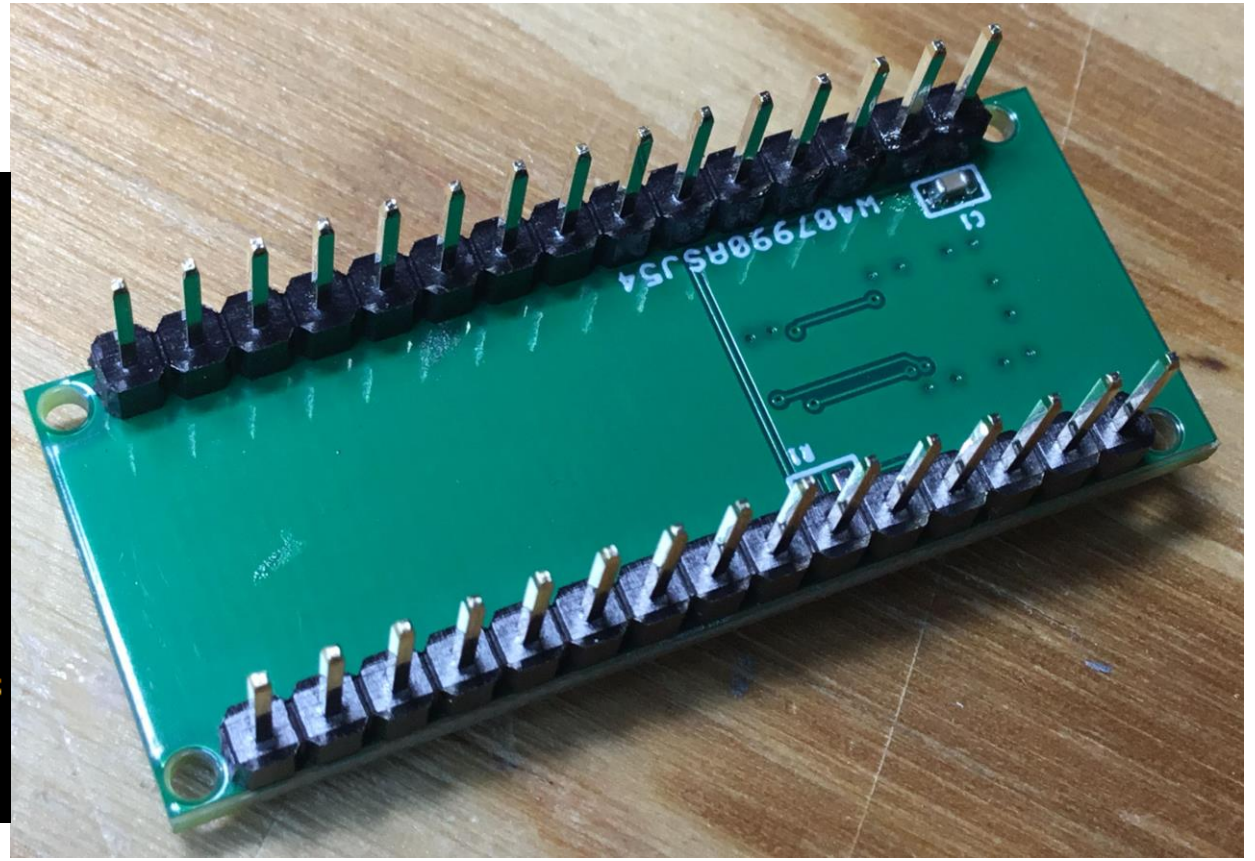
Load the Arduino BLE and Wi-Fi Libraries



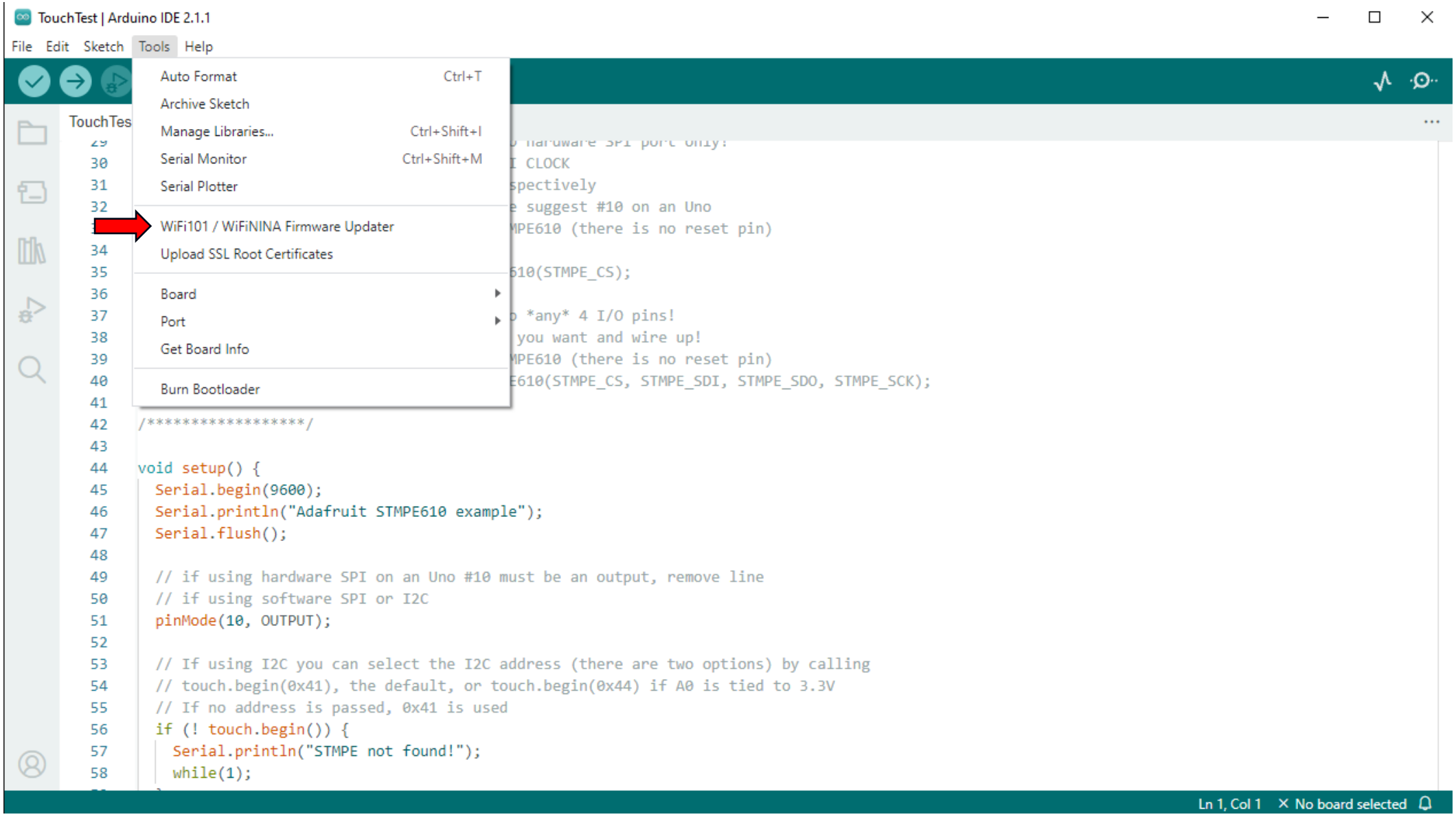


Pins Up - The Monster Is Dead

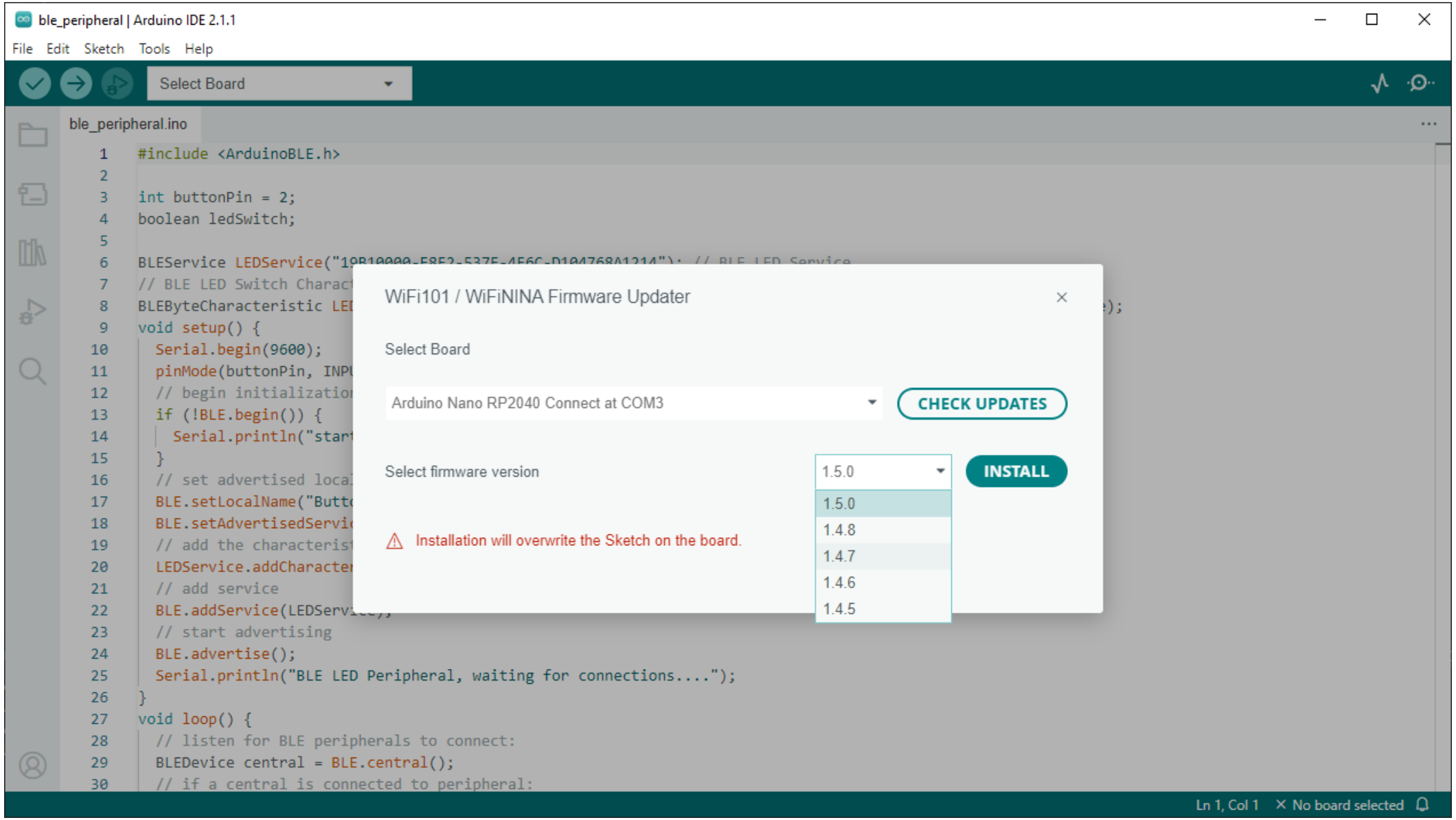




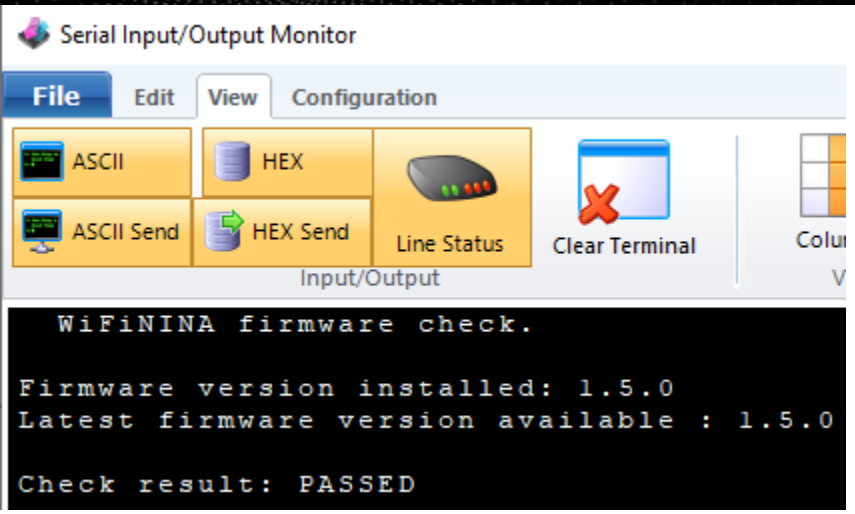
Give Life to the Monster

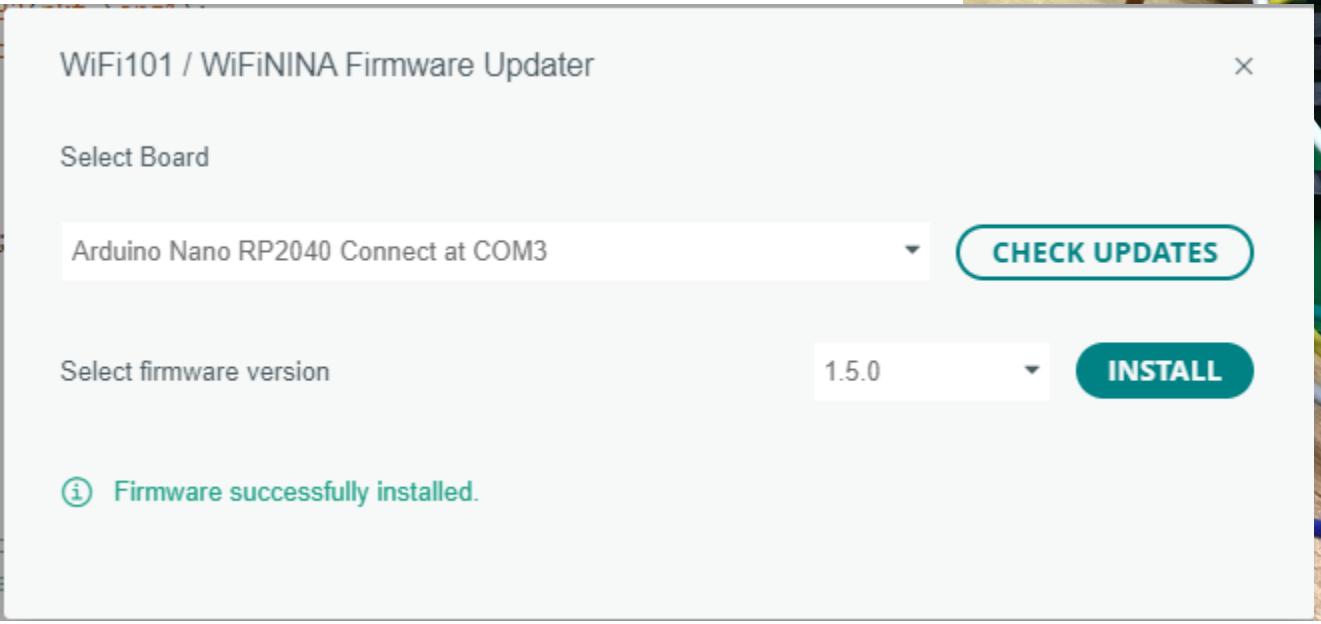


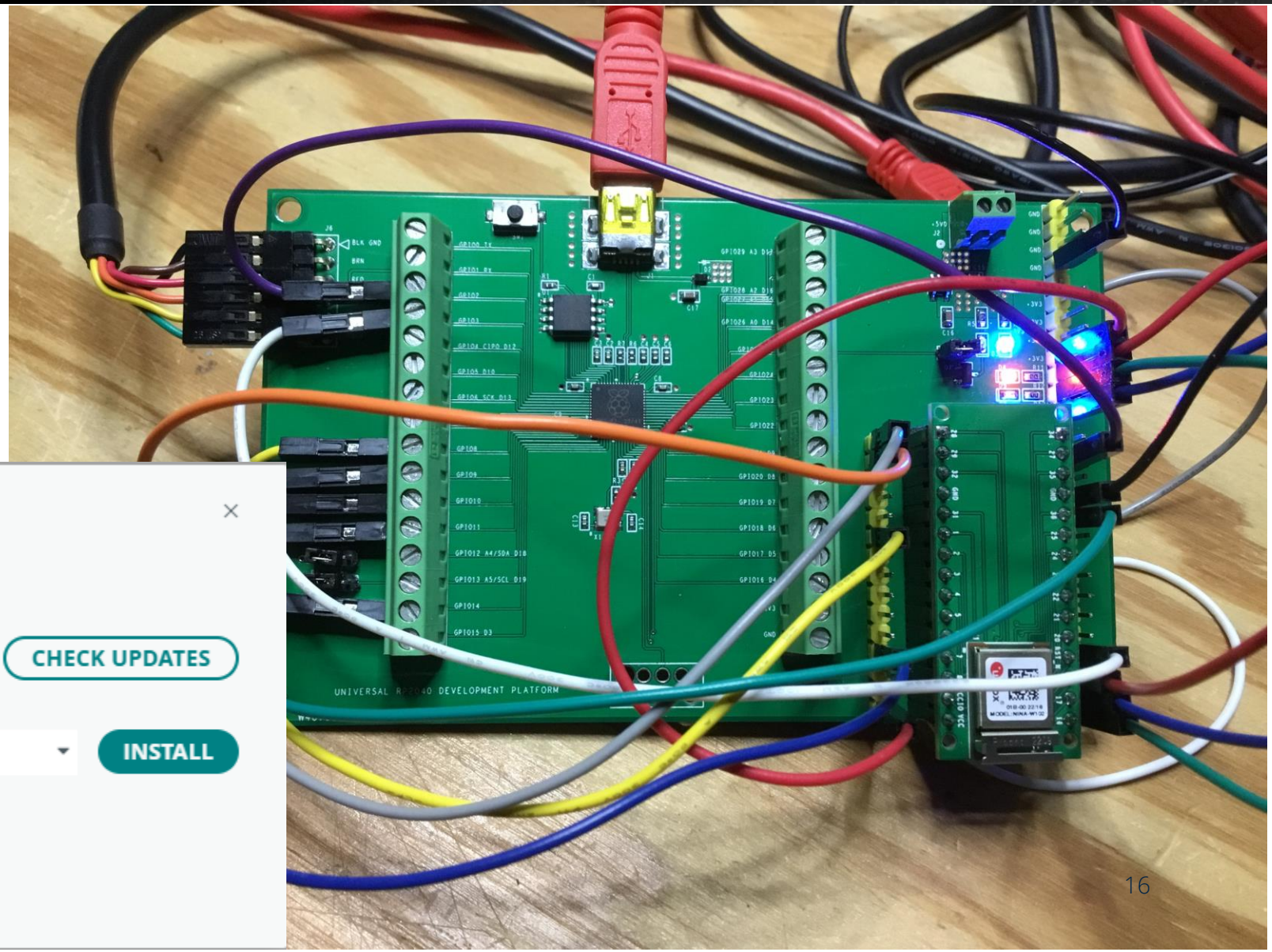
Give Life to the Monster



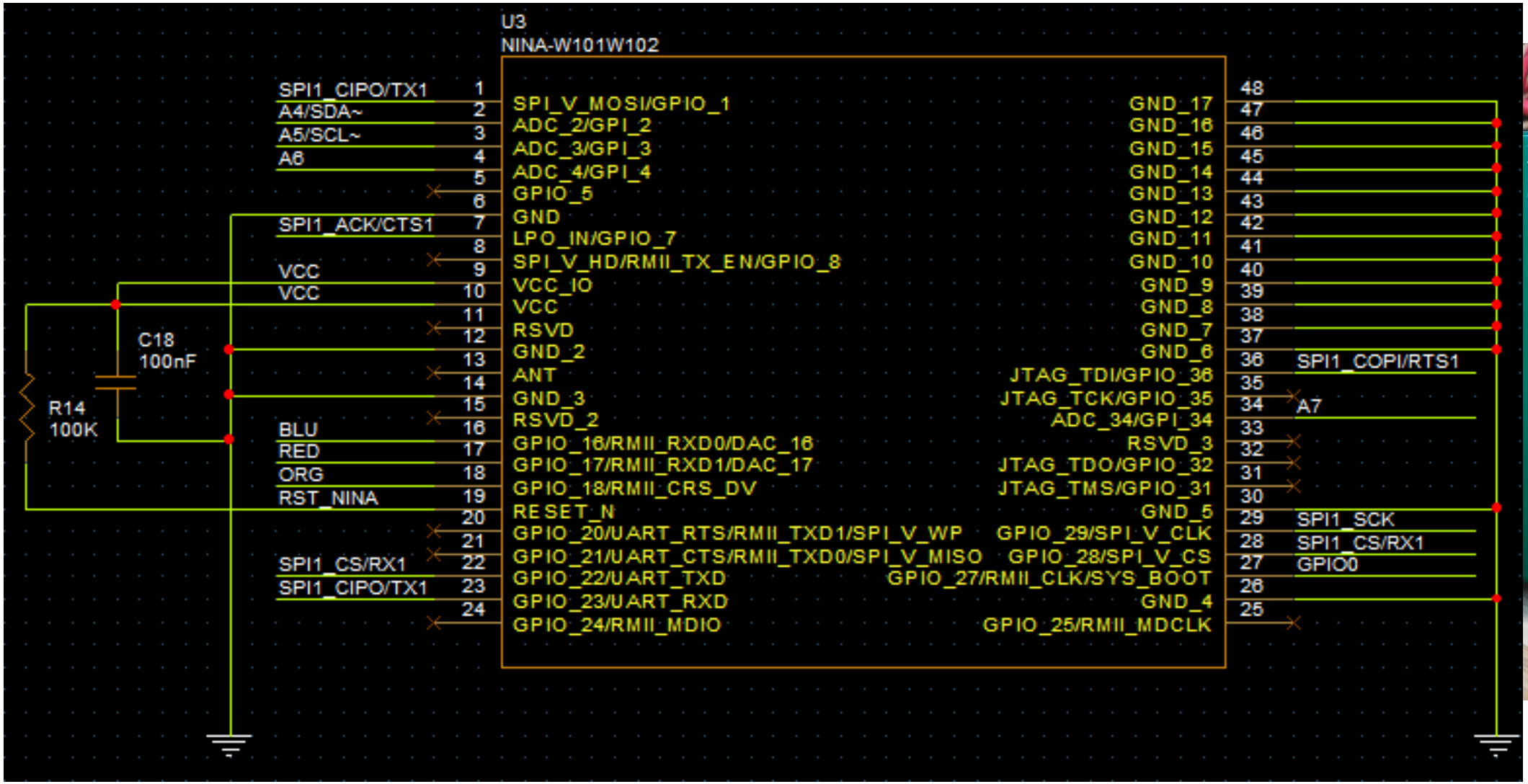
Give Life to the Monster

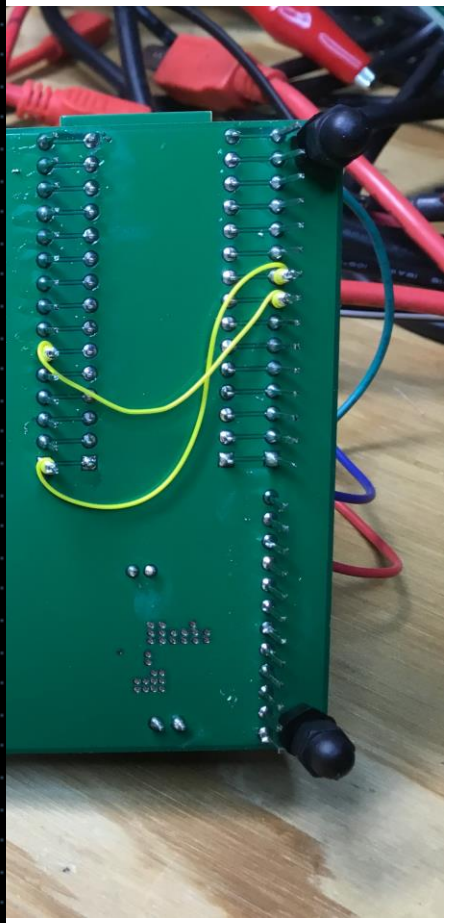






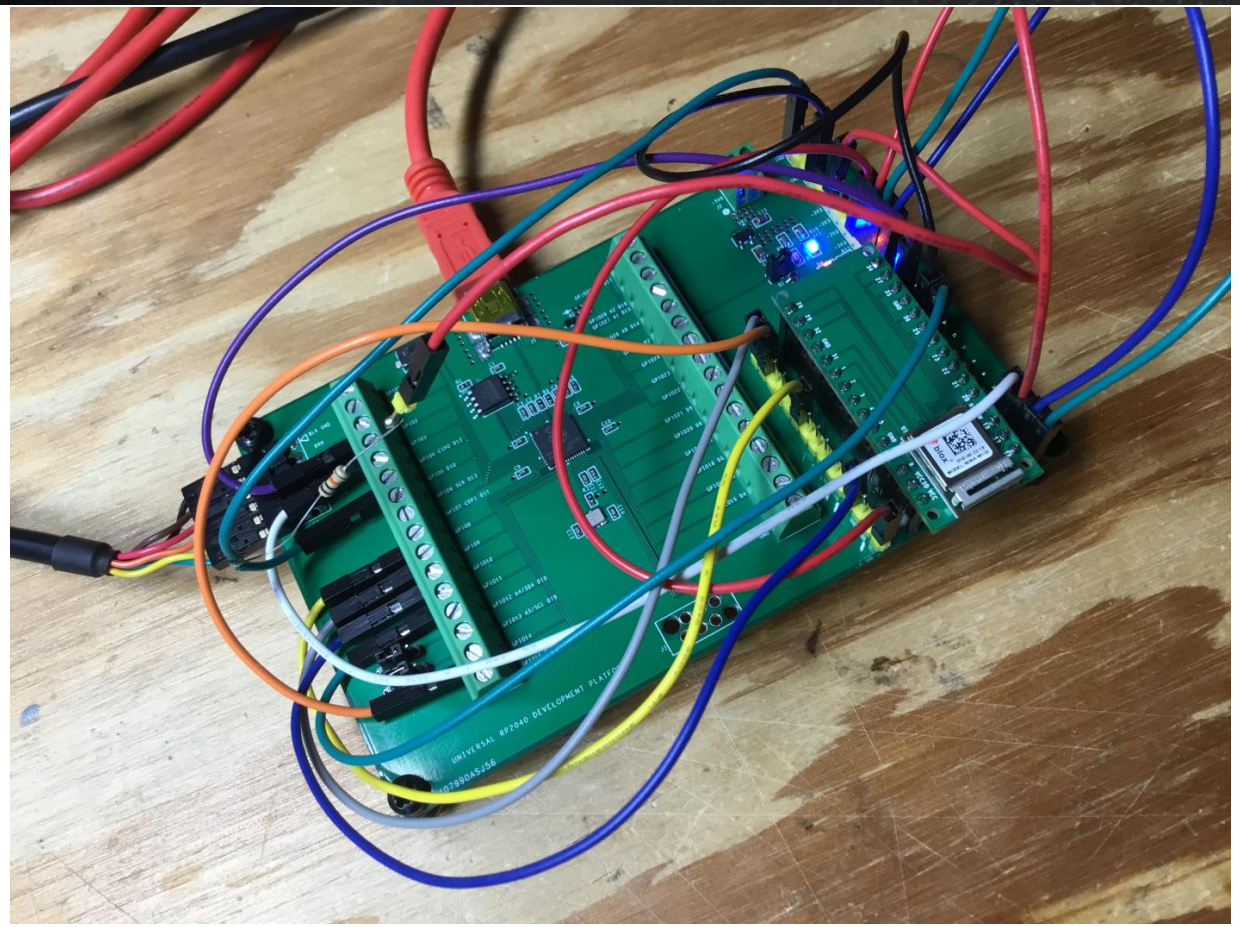
Arduino BLE Setup





Arduino BLE Setup

```
#include <ArduinoBLE.h>

const int ledPin = LED_BUILTIN; // set ledPin to on-board LED (D2 - RP2050 GPIO25)
const int buttonPin = 10;       // set buttonPin to digital pin 10 (D10 - RP2050 GPIO5)

BLEService ledService("19B10010-E8F2-537E-4F6C-D104768A1214"); // create service
// create switch characteristic and allow remote device to read and write
BLEByteCharacteristic ledCharacteristic("19B10011-E8F2-537E-4F6C-D104768A1214", BLERead | BLEWrite | BLENotify);
// create button characteristic and allow remote device to get notifications
BLEByteCharacteristic buttonCharacteristic("19B10012-E8F2-537E-4F6C-D104768A1214", BLERead | BLENotify);

void setup() {
  Serial1.begin(9600);
  while (!Serial1);

  pinMode(ledPin, OUTPUT);    // configure the LED pin as an output
  pinMode(buttonPin, INPUT);  // configure the button pin as an input

  // BLE initialization
  if (!BLE.begin()) {
    Serial1.println("Starting Bluetooth® - Low Energy module failed!");
    while (1);
  }

  // set the local name peripheral advertises
  BLE.setLocalName("BTNLED");
  // set the UUID for the service this peripheral advertises:
  BLE.setAdvertisedService(ledService);

  // add the characteristics to the service
  ledService.addCharacteristic(ledCharacteristic);
  ledService.addCharacteristic(buttonCharacteristic);

  // add the service
  BLE.addService(ledService);

  ledCharacteristic.writeValue(0);
  buttonCharacteristic.writeValue(0);

  // start advertising
  BLE.advertise();

  Serial1.println("Bluetooth® device active, waiting for connections...");
}
```

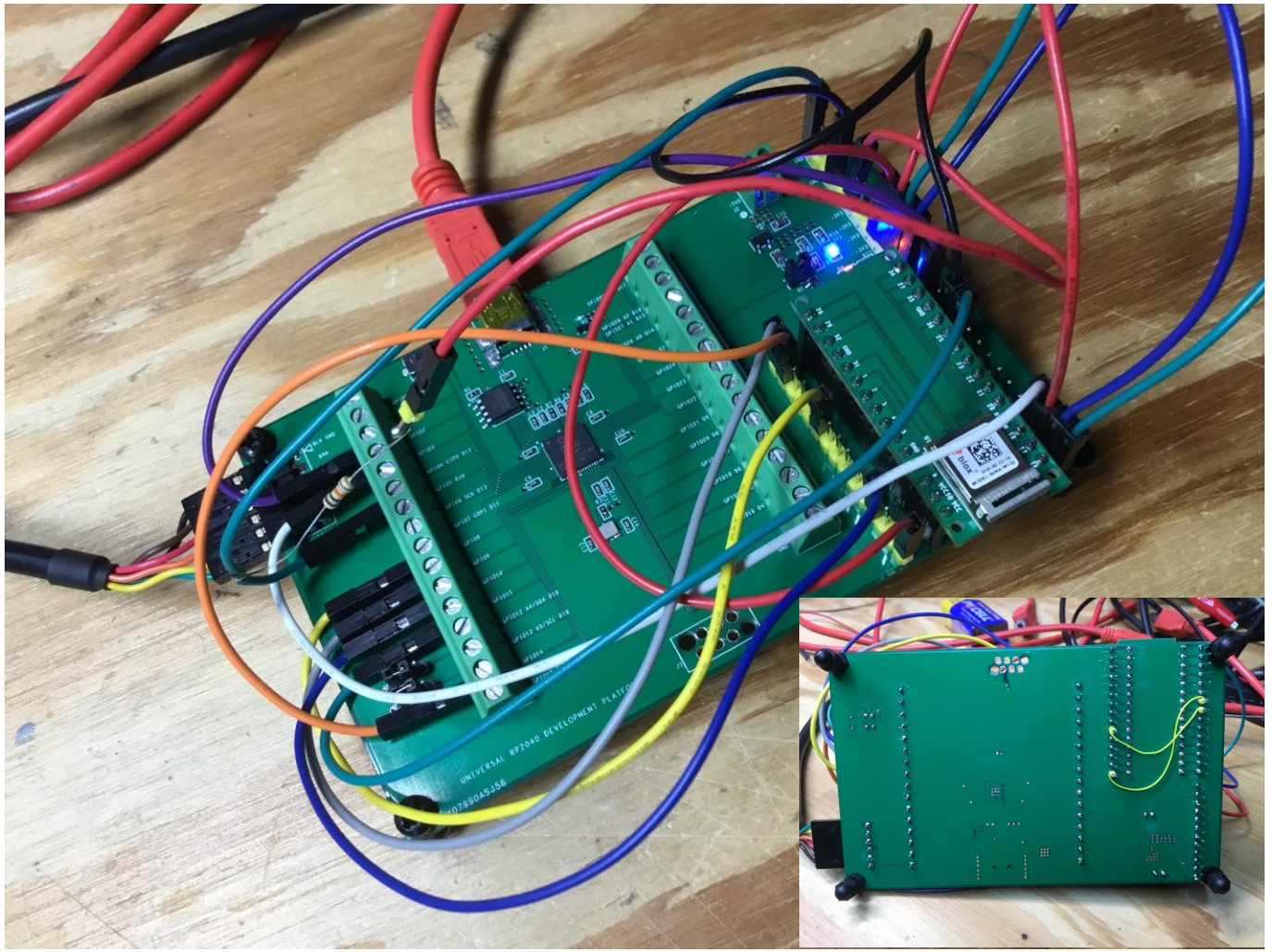


```
// D8 - D13
{ p20,        NULL, NULL, NULL },    // D8
{ p21,        NULL, NULL, NULL },    // D9
{ p5,         NULL, NULL, NULL },    // D10
{ p7,         NULL, NULL, NULL },    // D11 / SPITX
{ p4,         NULL, NULL, NULL },    // D12 / SPIRX
{ p6,         NULL, NULL, NULL },    // D13 / SPICLK / LEDB
```

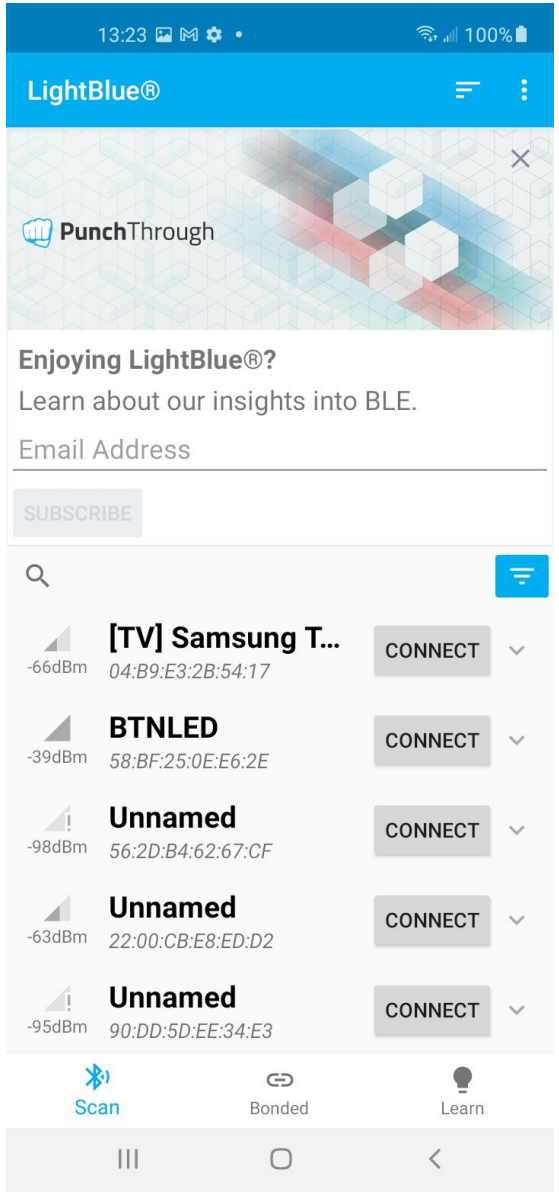
buttonPin

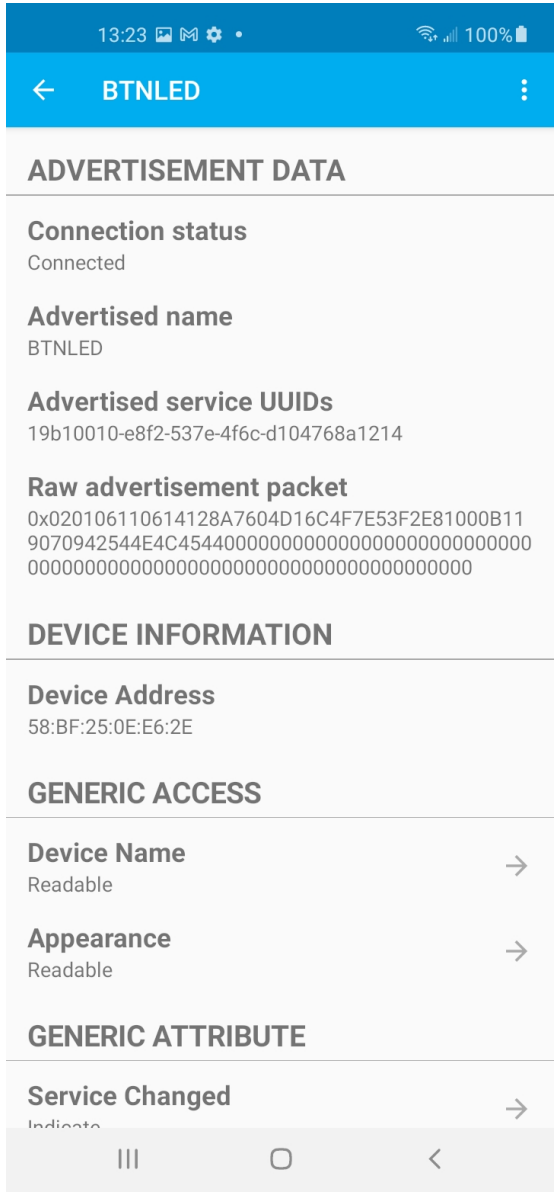
Arduino BLE Loop

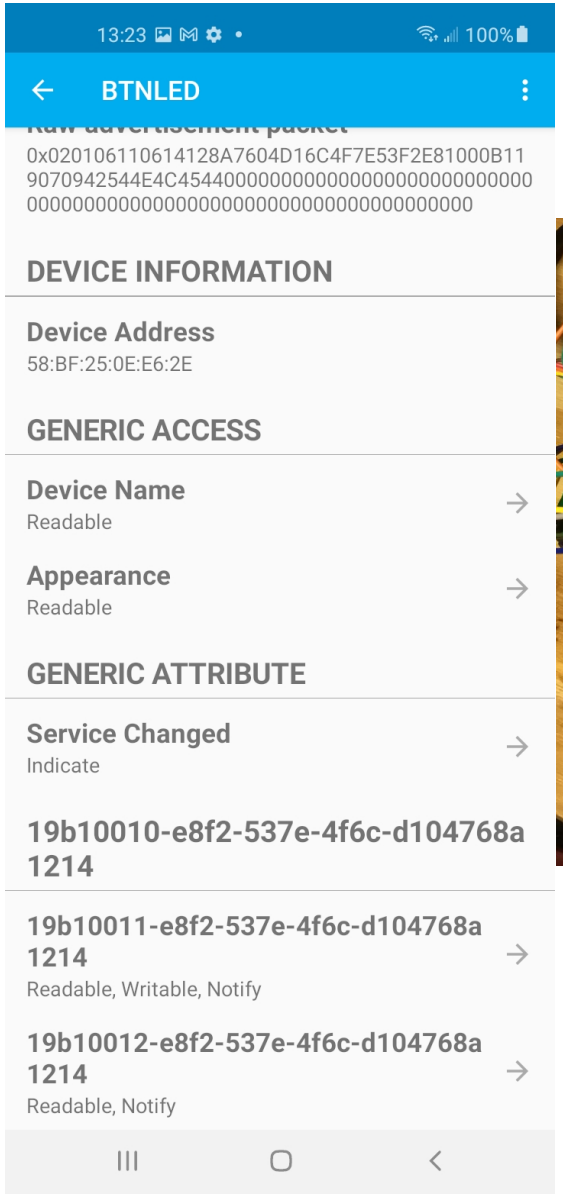
```
65  void loop() {
66    // Poll for Bluetooth® Low Energy events
67    BLE.poll();
68
69    // read the current button pin state
70    char buttonValue = digitalRead(buttonPin);
71
72    // has the value changed since the last read
73    bool buttonChanged = (buttonCharacteristic.value() != buttonValue);
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  }
```

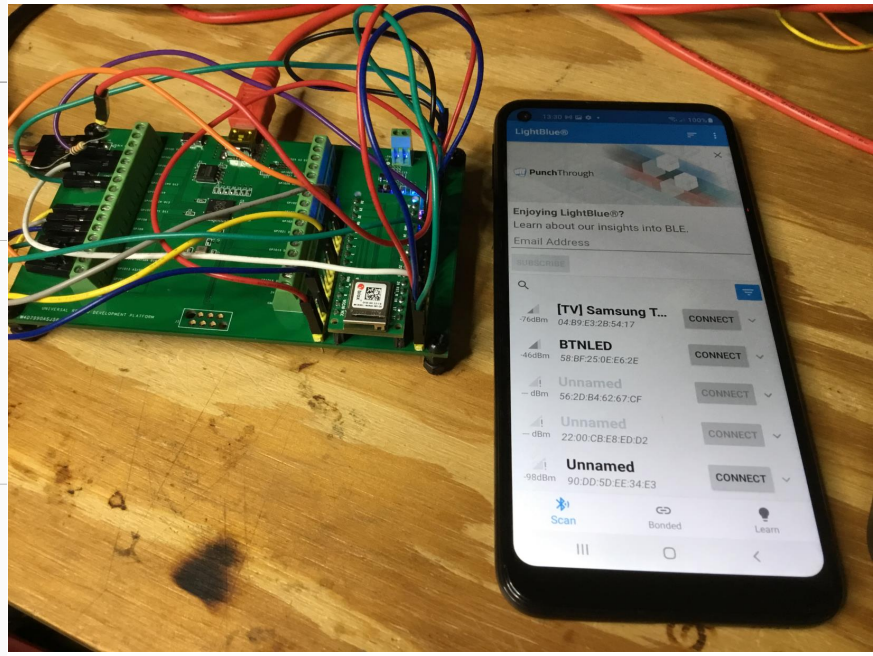


Arduino BLE First Contact

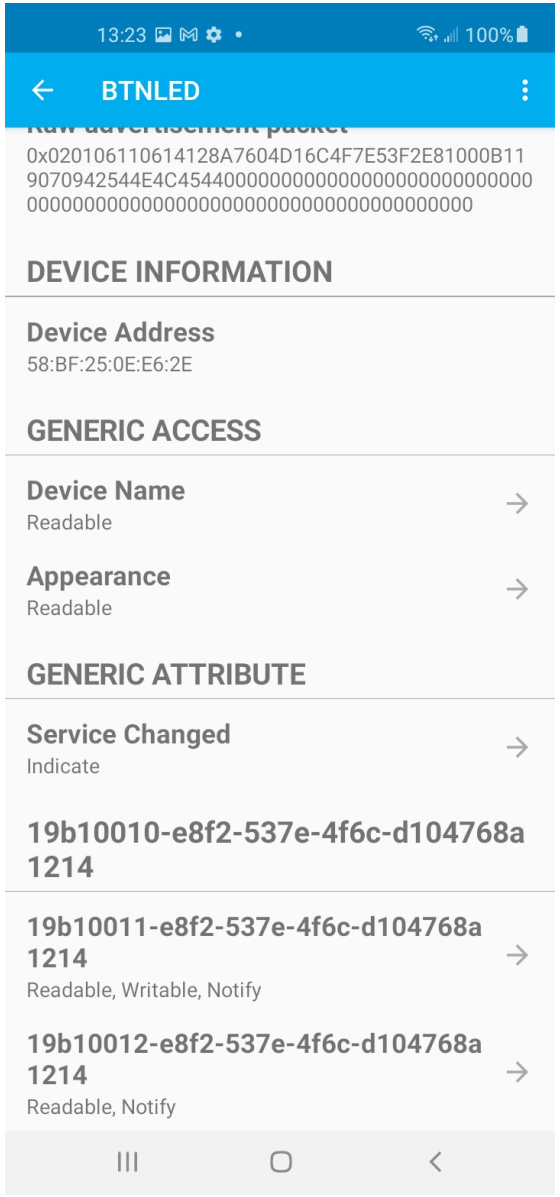


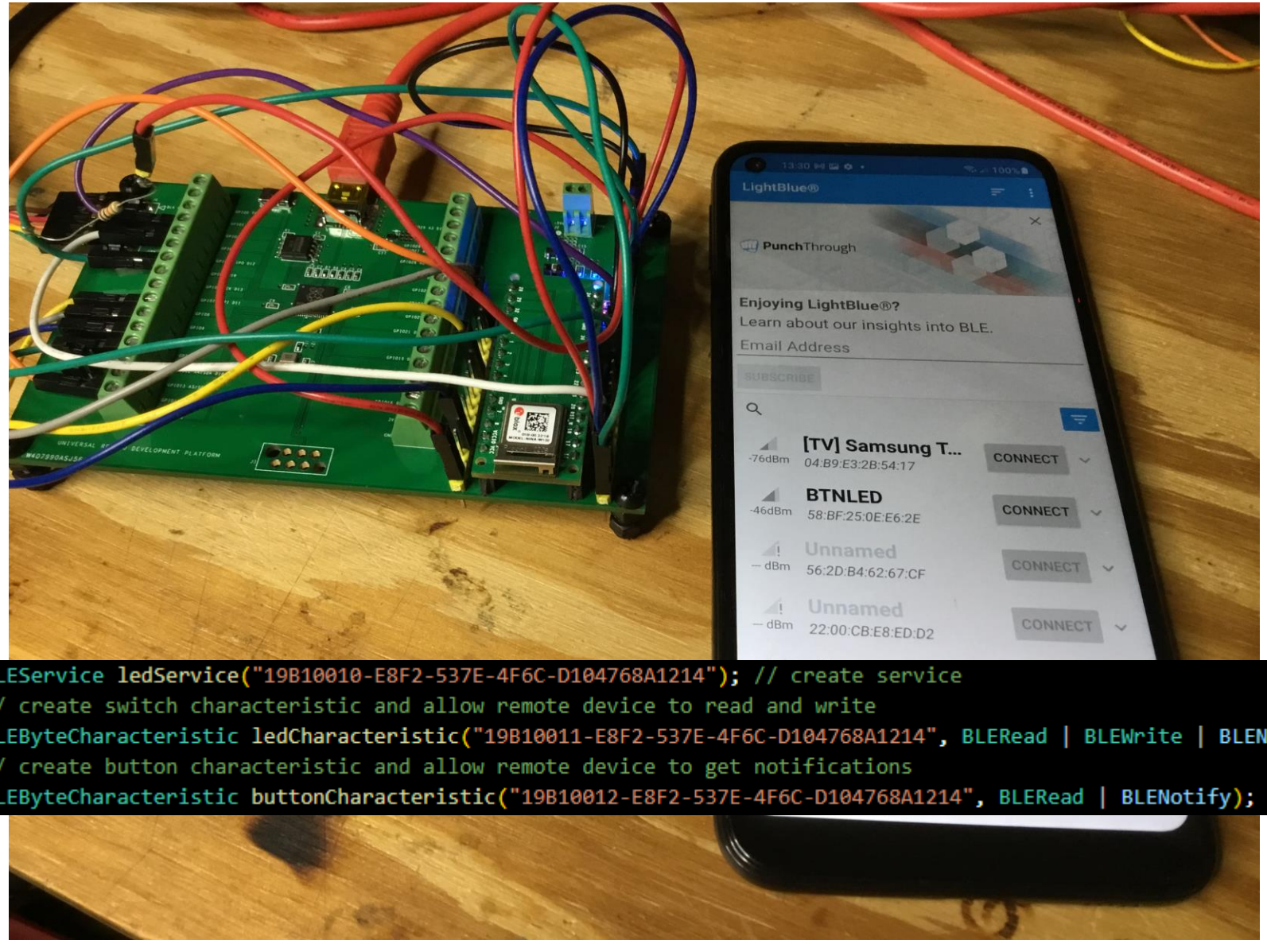






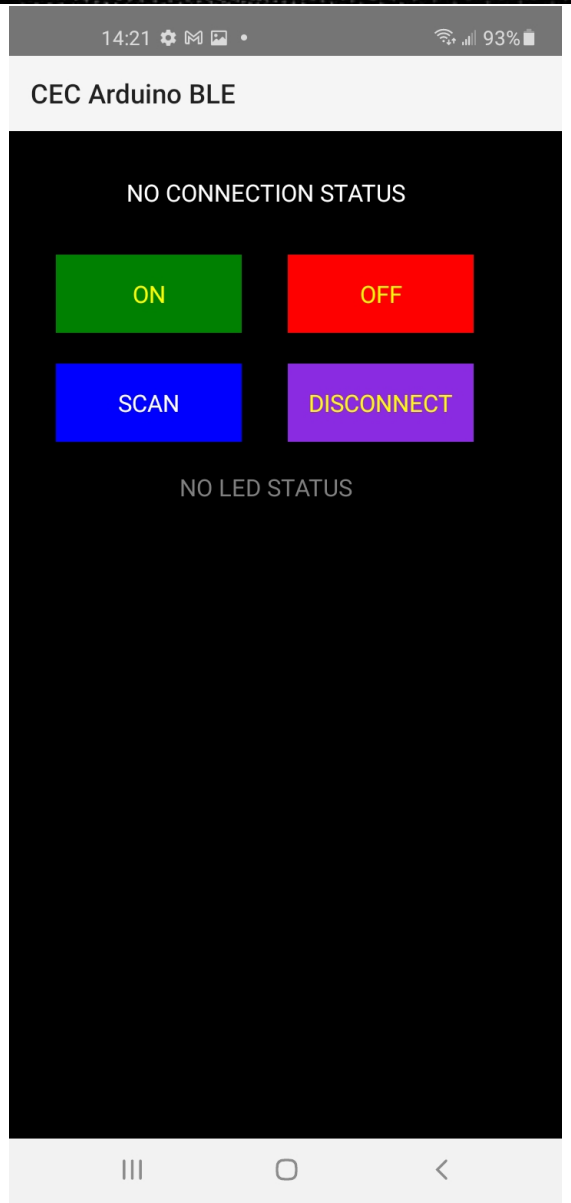
Arduino BLE First Contact

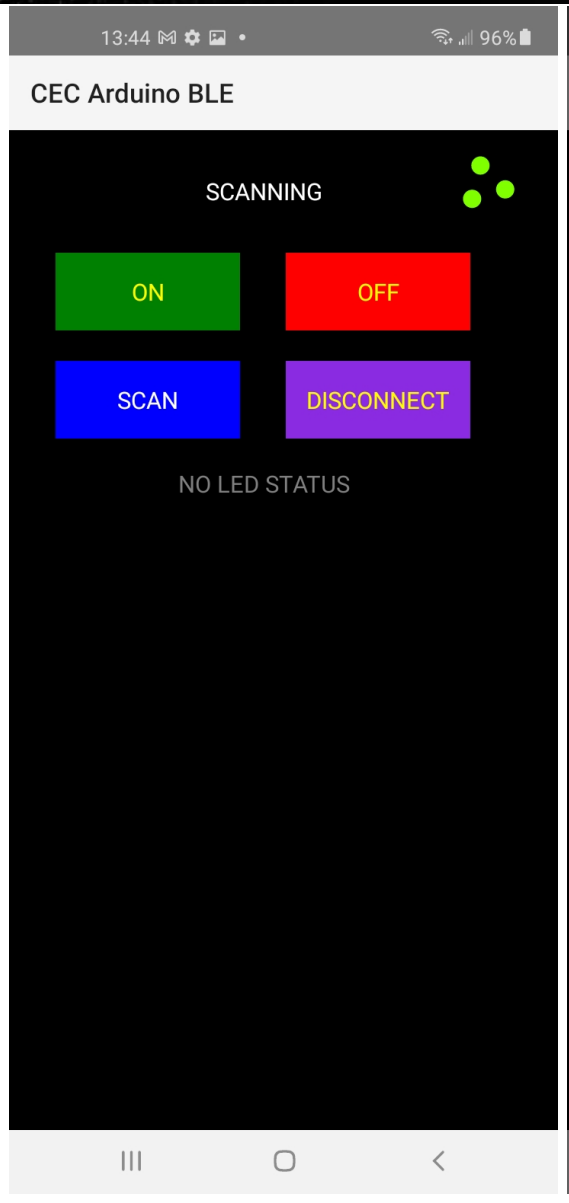


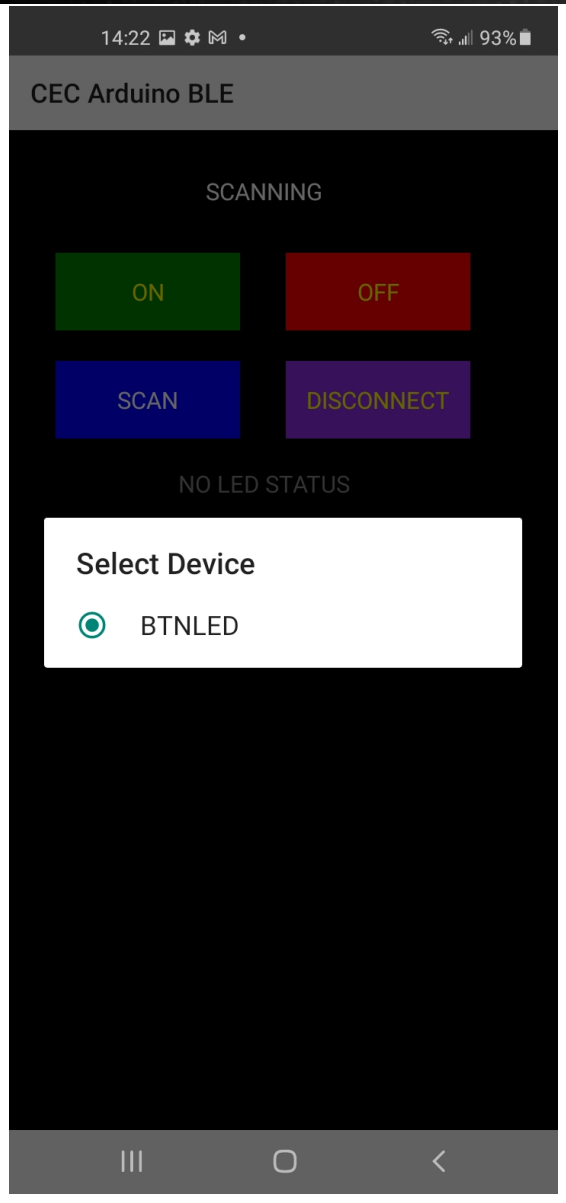


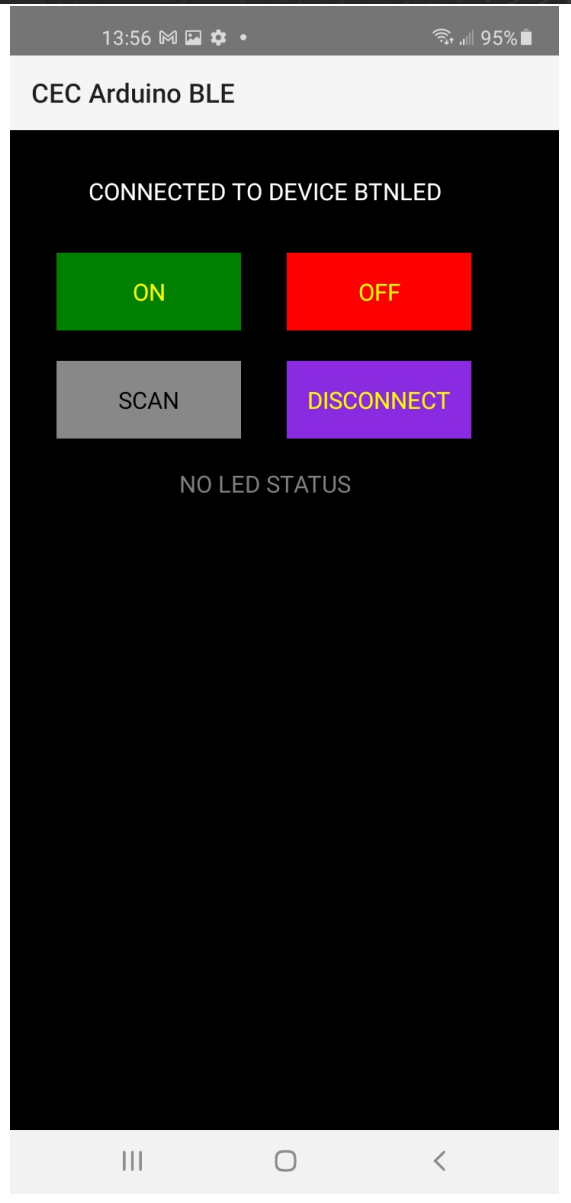

```
25  BLEService ledService("19B10010-E8F2-537E-4F6C-D104768A1214"); // create service
26  // create switch characteristic and allow remote device to read and write
27  BLEByteCharacteristic ledCharacteristic("19B10011-E8F2-537E-4F6C-D104768A1214", BLERead | BLEWrite | BLENotify);
28  // create button characteristic and allow remote device to get notifications
29  BLEByteCharacteristic buttonCharacteristic("19B10012-E8F2-537E-4F6C-D104768A1214", BLERead | BLENotify);
```

Arduino/B4A BLE Remote Control App

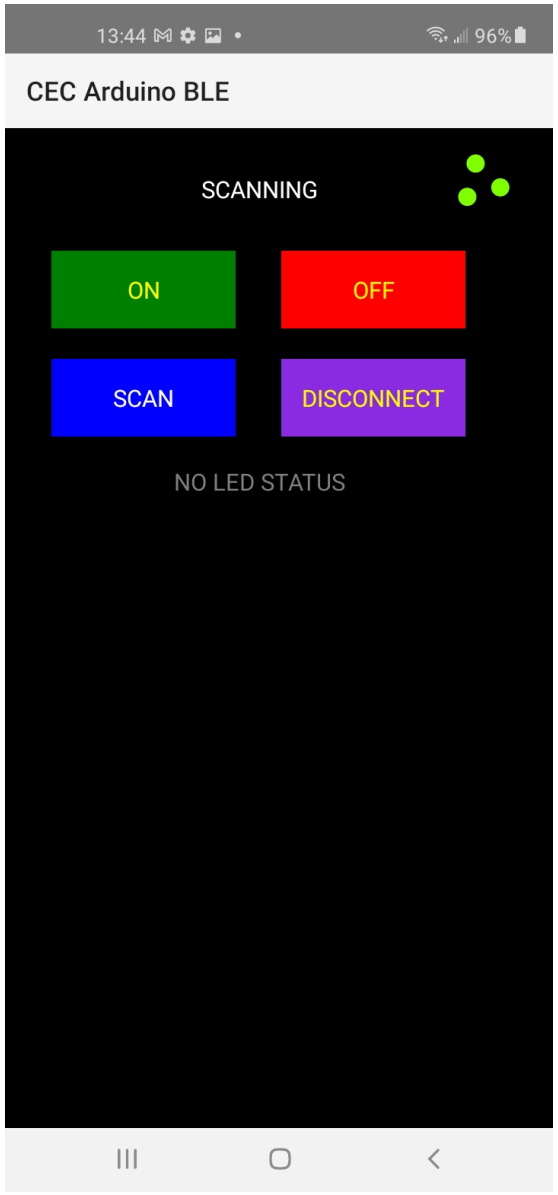


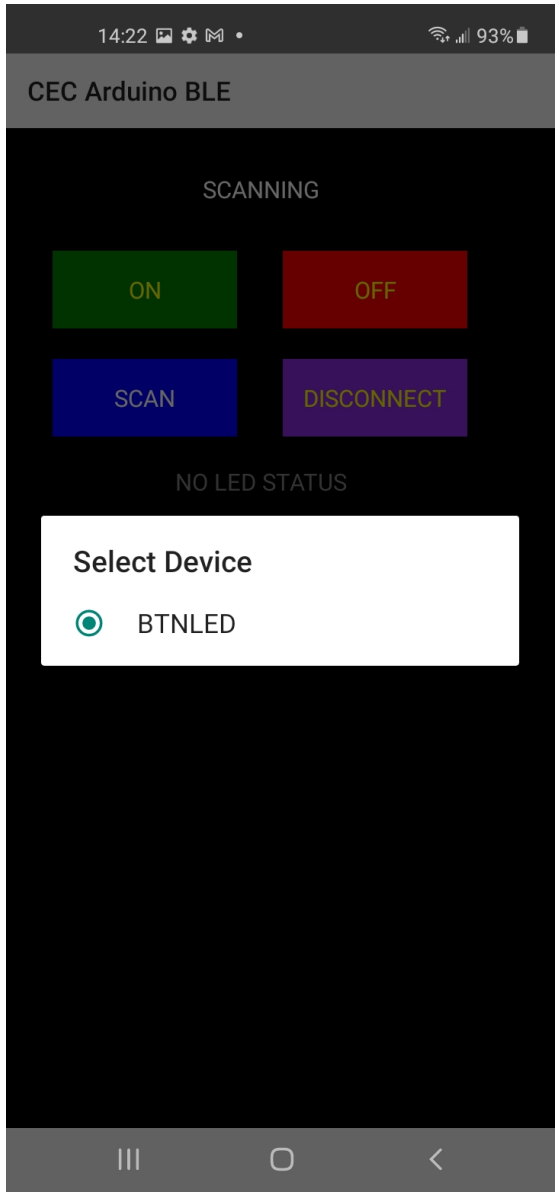


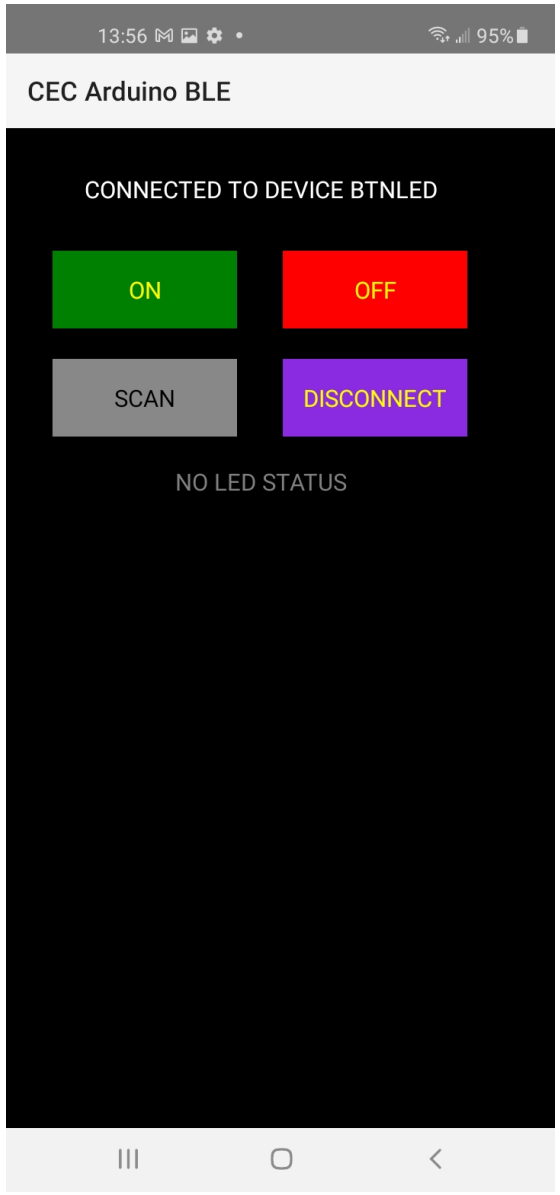




Arduino/B4A BLE Remote Control App

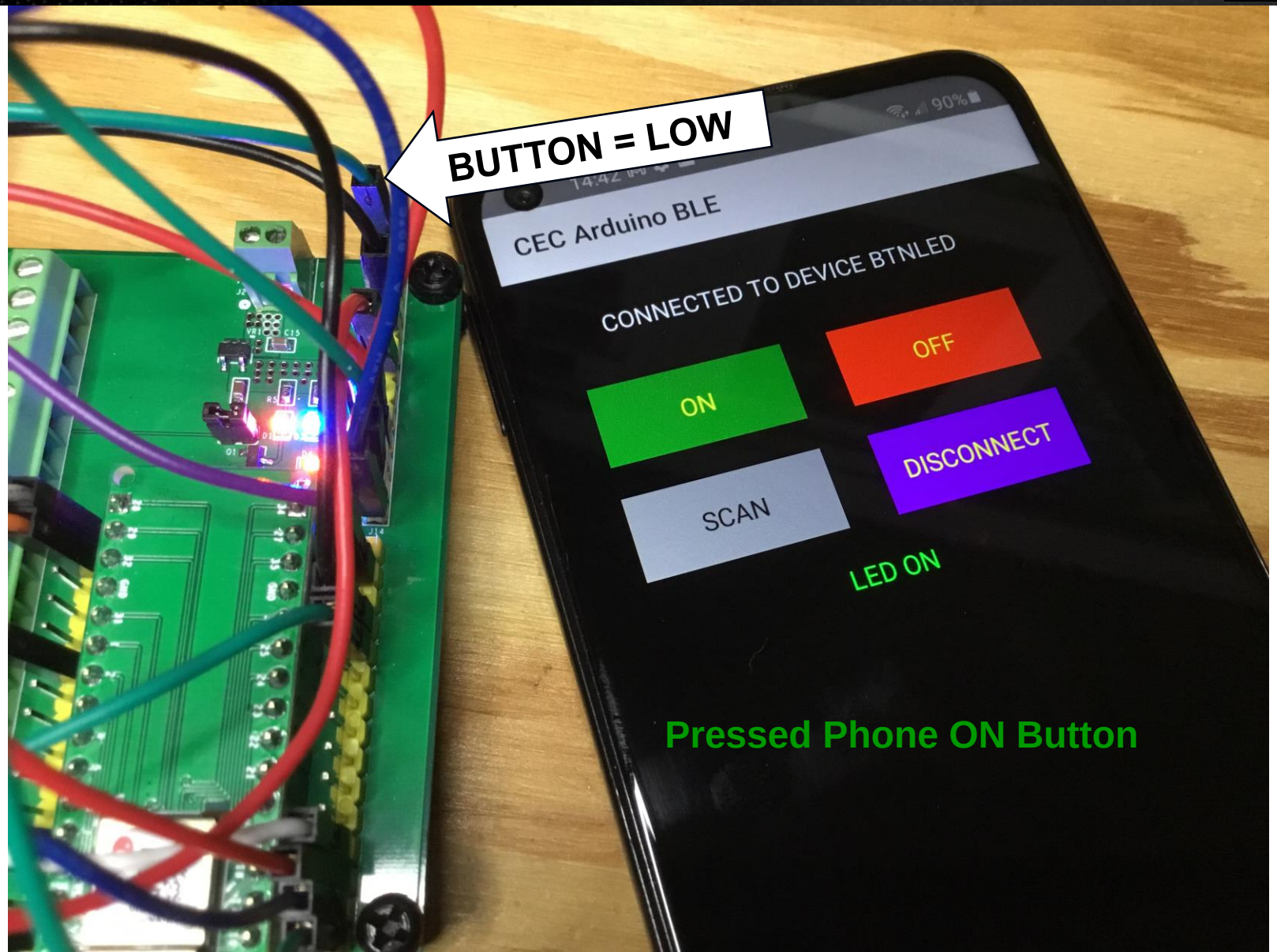




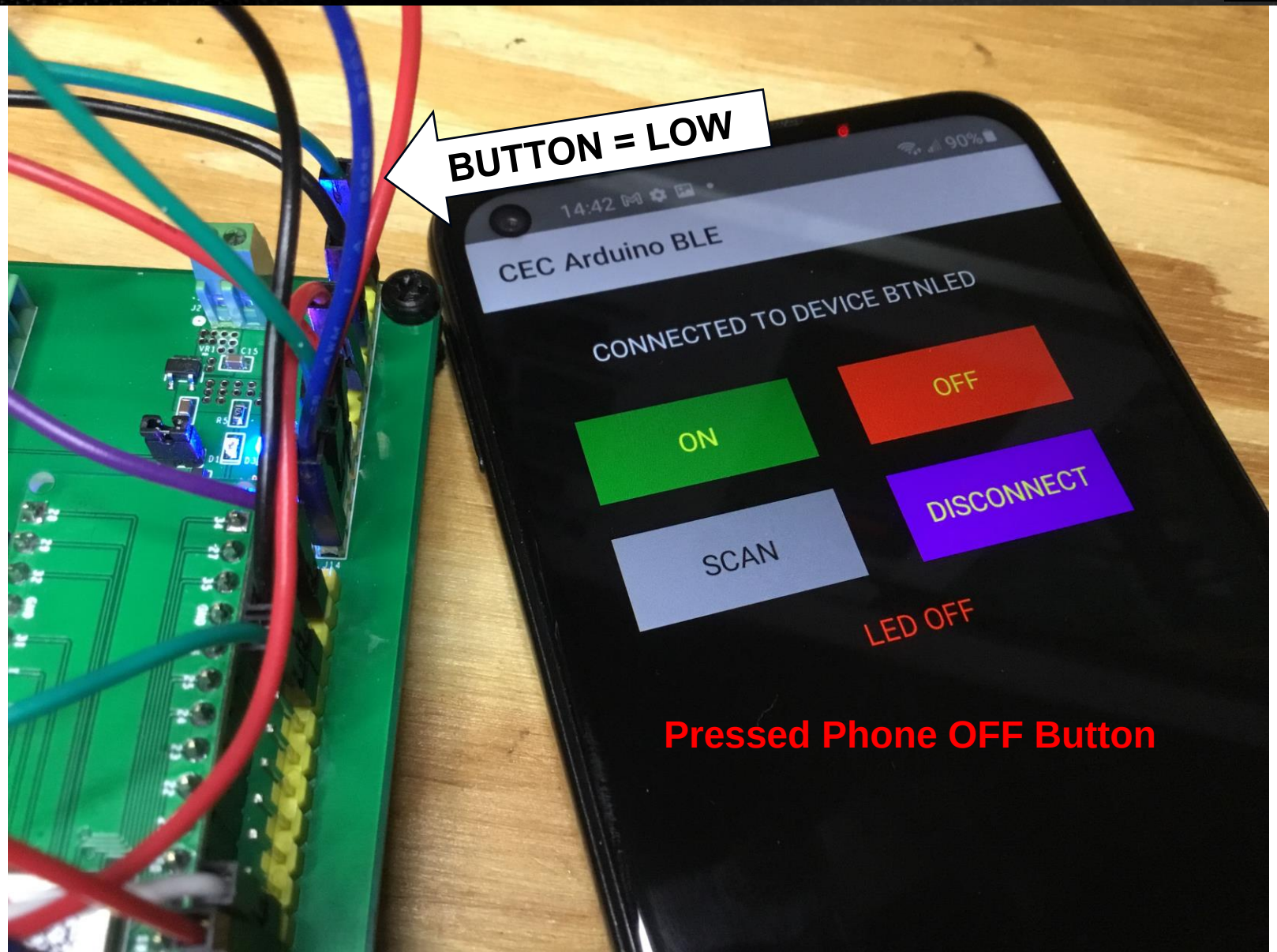



```
Device found:
Device found: [TV] Samsung TU700D 43 TV
Device found: BTNLED
Device found:
Device found:
Device found:
Device found:
Device found:
Device found:
Connect to: 58:BF:25:0E:E6:2E
Discovering services.
Connected
Setting descriptor. Success = true
writing descriptor: true
UUID: 00001800-0000-1000-8000-00805f9b34fb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

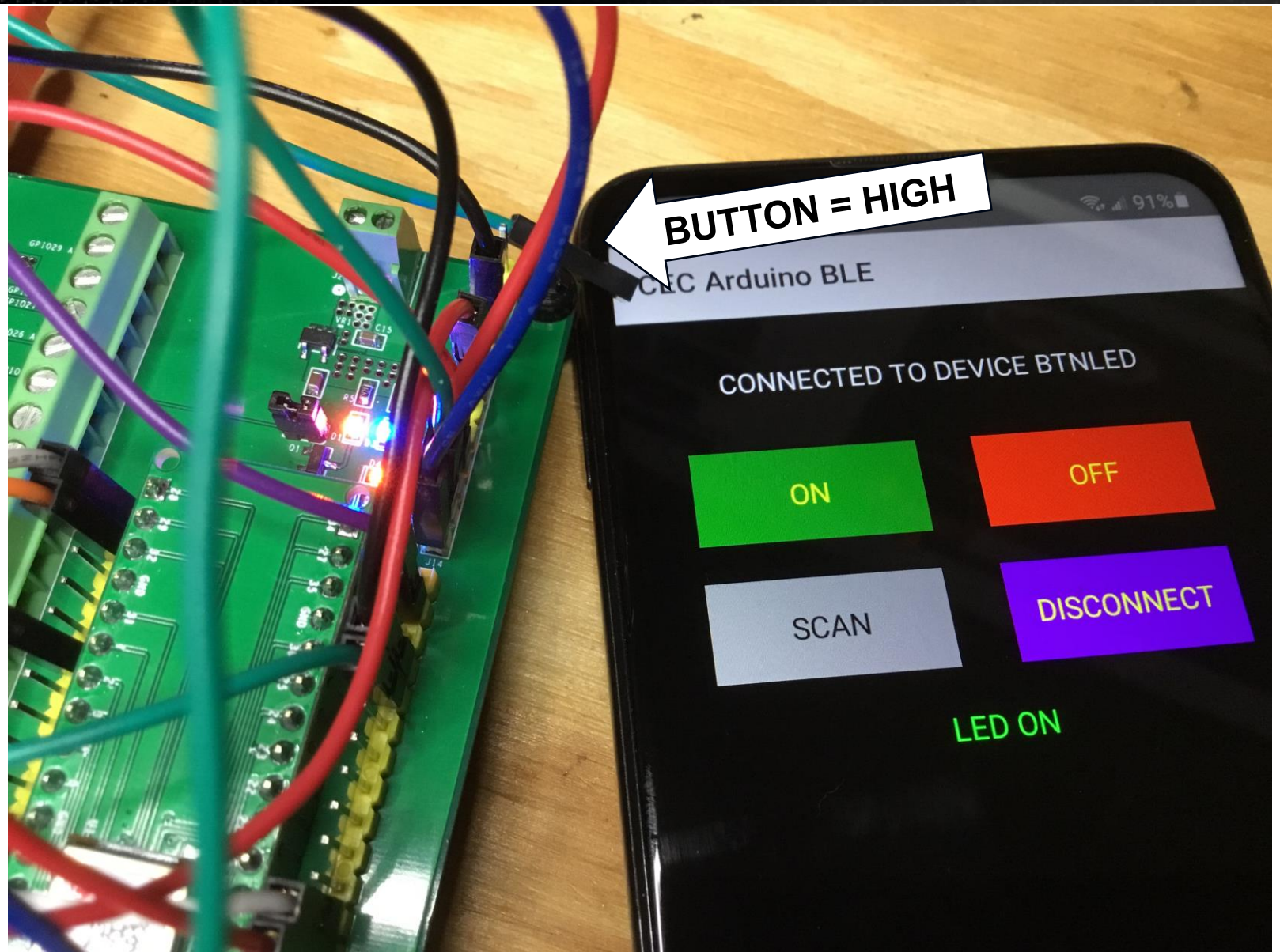
Arduino/B4A BLE Remote Control App



Arduino/B4A BLE Remote Control App

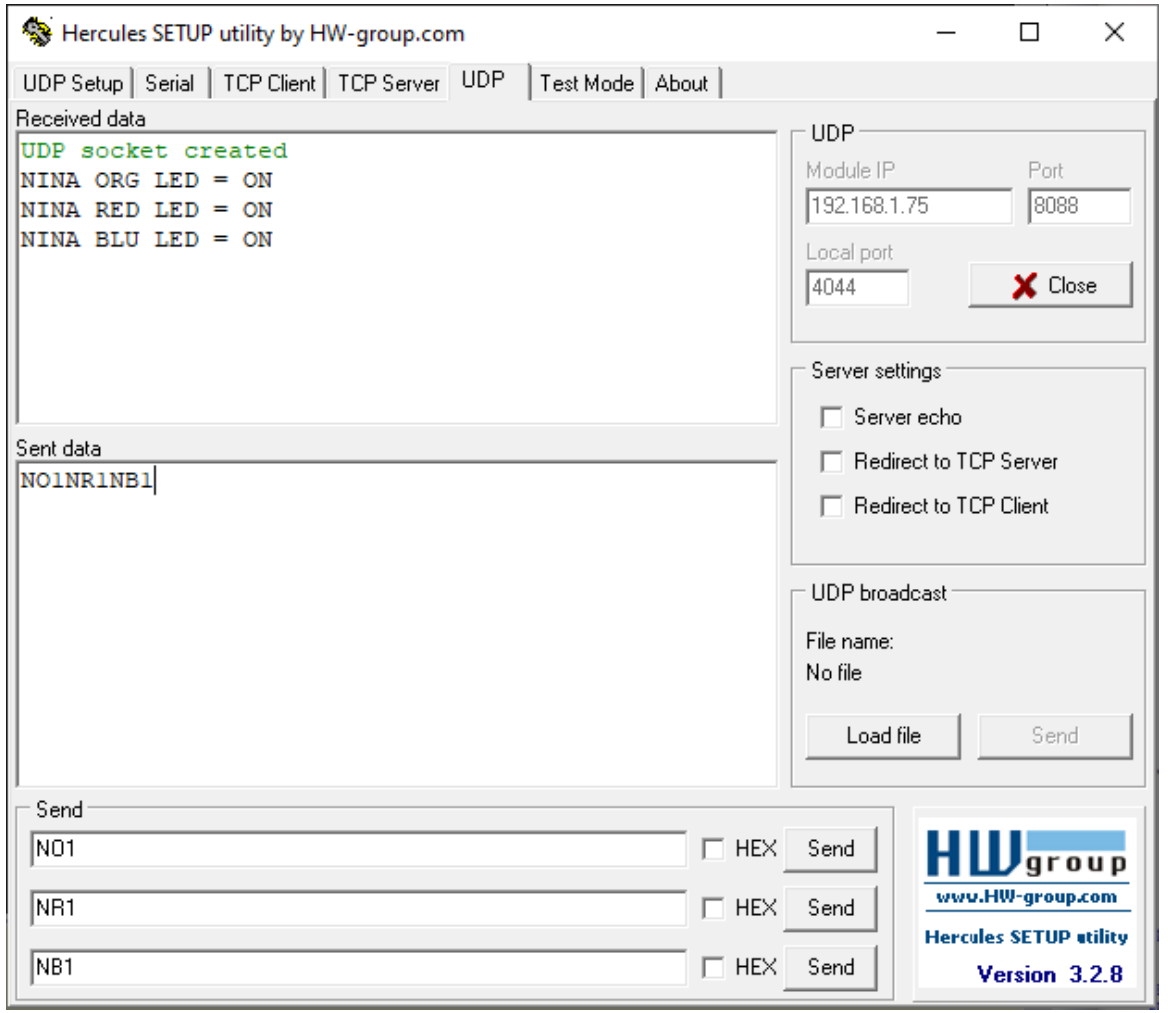


Arduino/B4A BLE Remote Control App



Thank you for attending!!!

MORE TO COME..



Please consider the resources below:

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

CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

DigiKey

informa markets