CEC Continuing Education Center

DesignNews

Designing, Building and Coding Custom Raspberry Pi RP2040 Arduino Devices

Day 4:

The RP2040, Arduino and Wi-Fi

Sponsored by

DigiKey

informa markets

Webinar Logistics

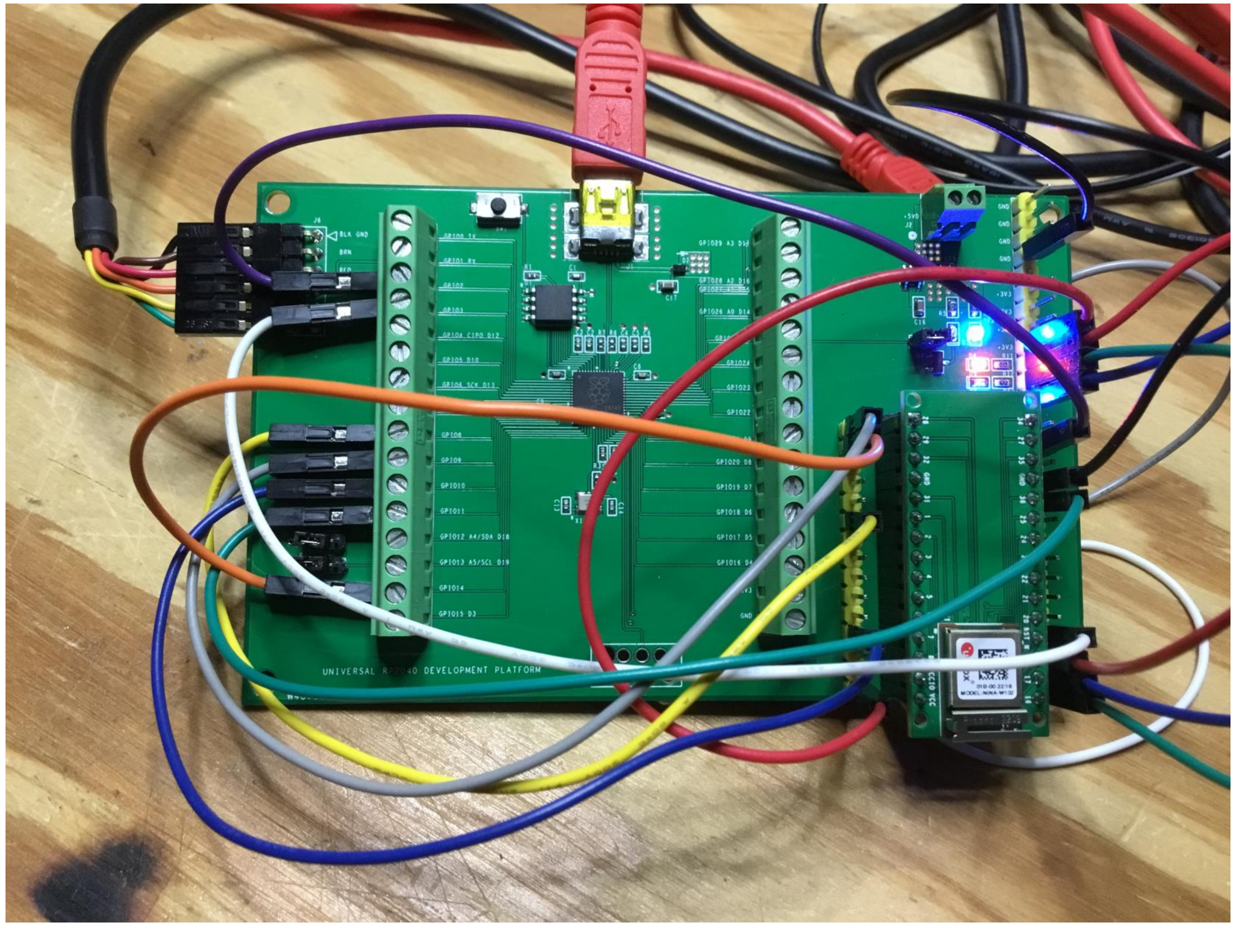
- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.

Fred Eady

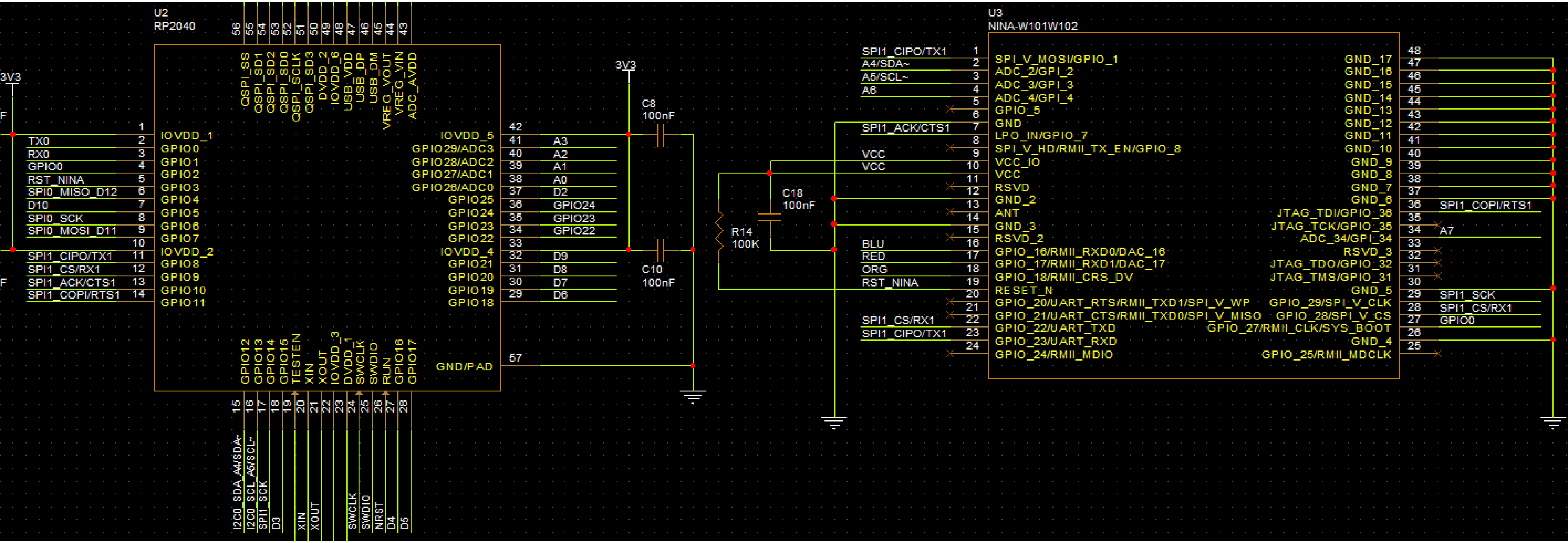
Visit ‘Lecturer Profile’ in your console for more details.

AGENDA

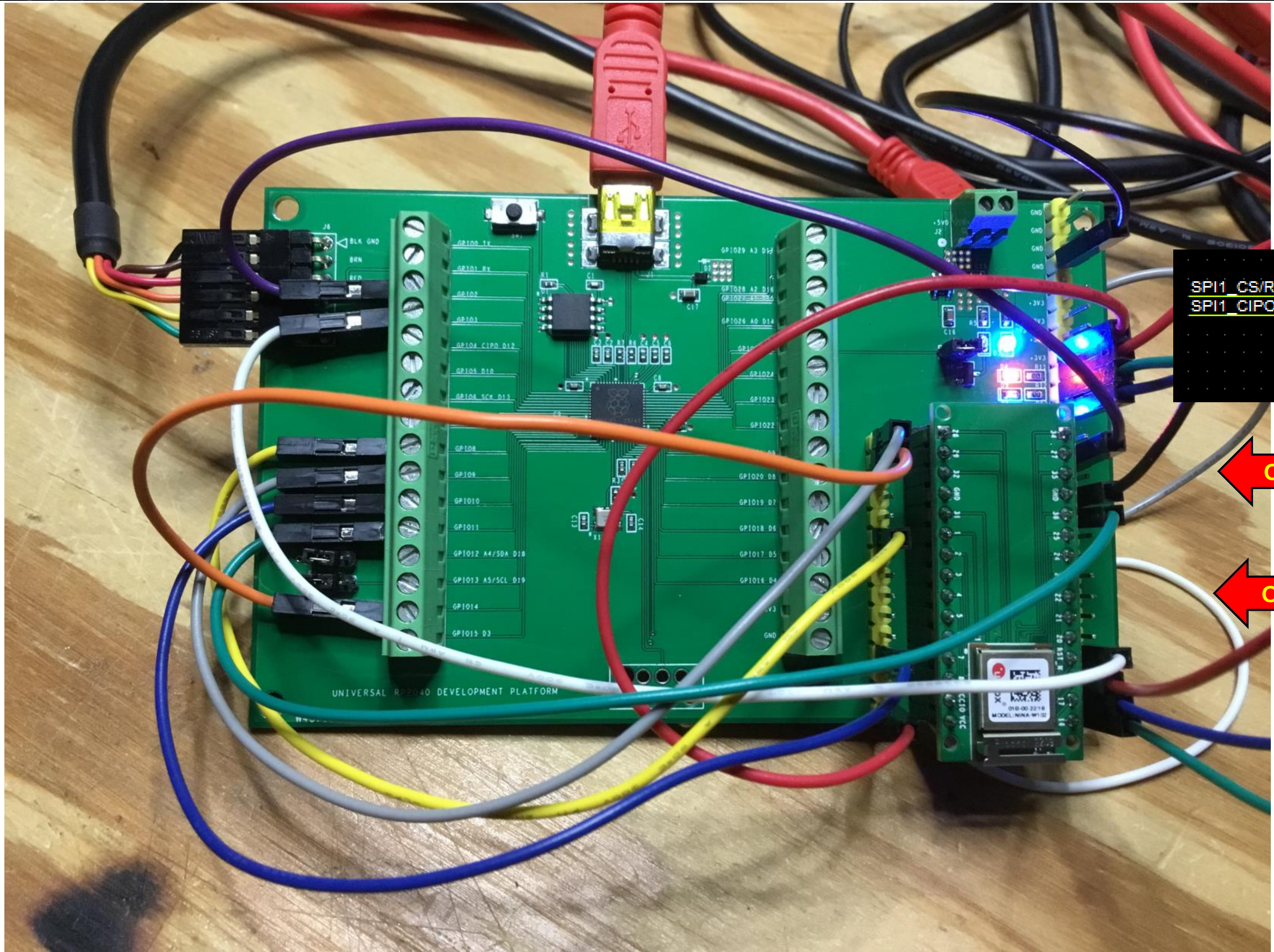
- RP2040 Arduino Wi-Fi Hardware
- u-blox Wi-Fi Powered by Arduino
- Wi-Fi Remote Control

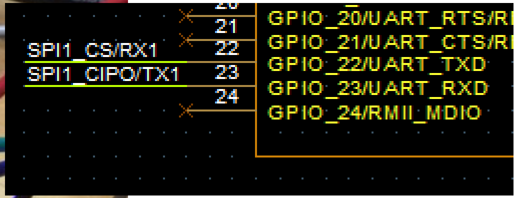


RP2040 Arduino Wi-Fi Hardware Design



RP2040 Arduino Wi-Fi Hardware Design





OPTIONAL FOR WI-FI

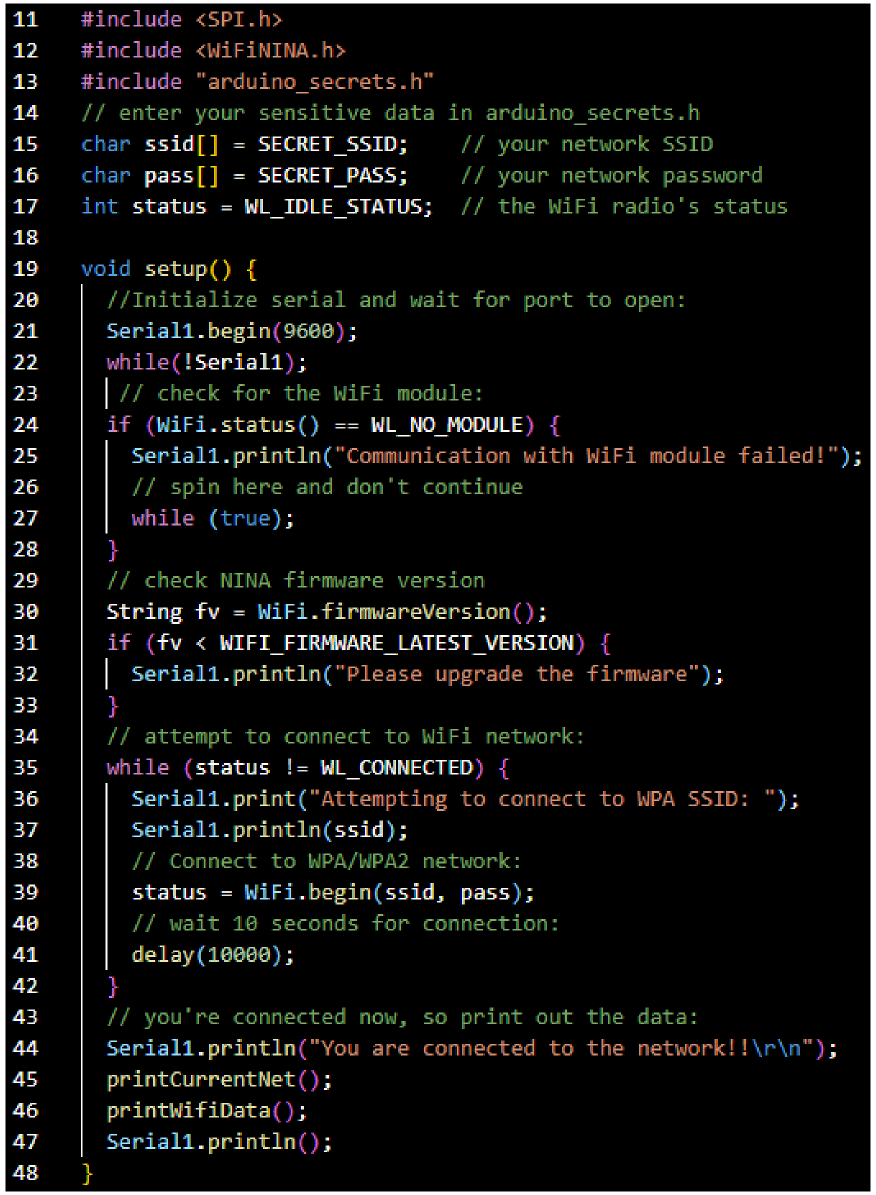
OPTIONAL FOR WI-FI

Designing, Building and Coding Custom Raspberry Pi RP2040 Arduino Devices

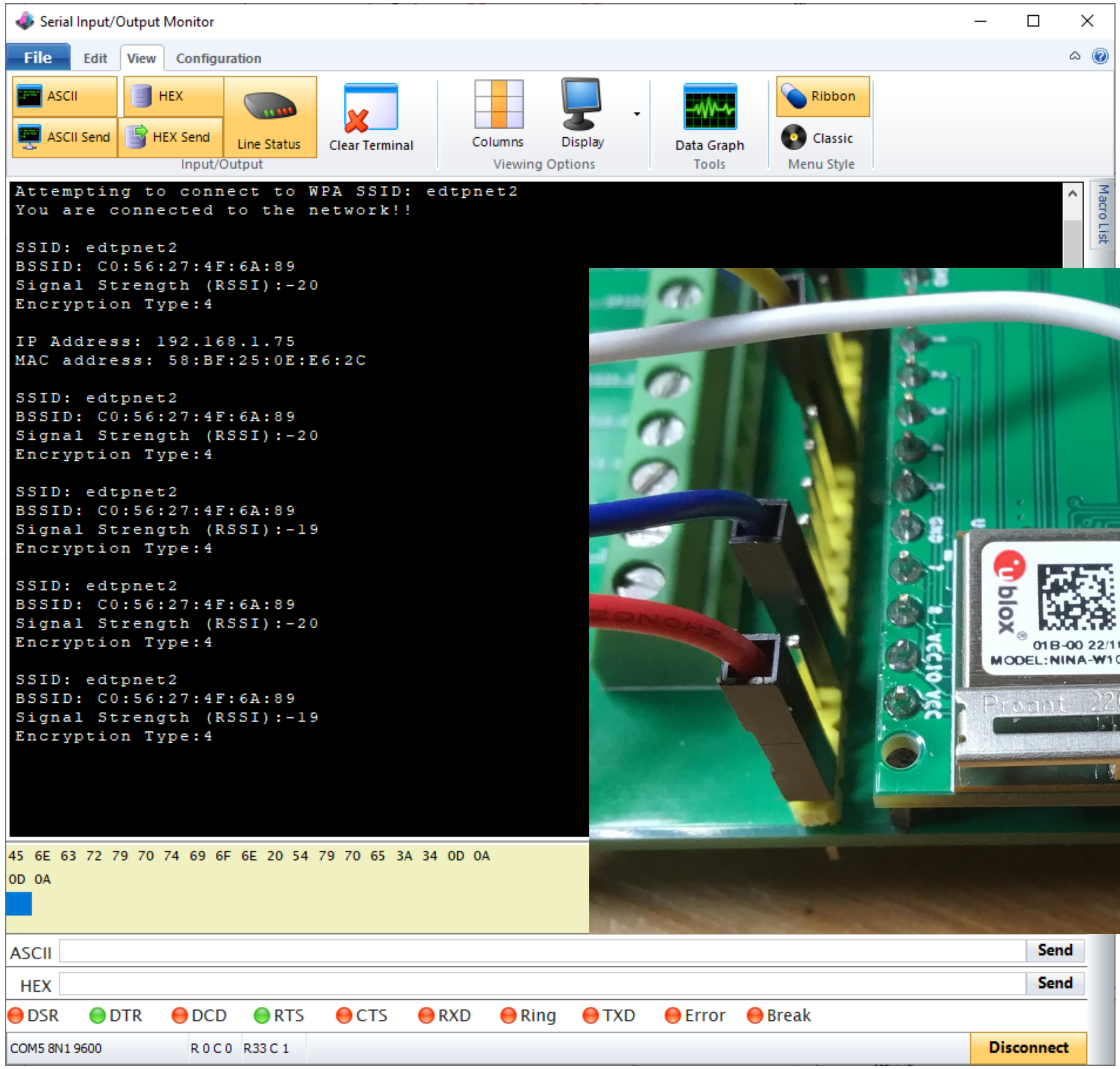
The RP2040, Arduino and Wi-Fi

u-blox Wi-Fi Powered by Arduino

Arduino Wi-Fi First Contact – setup()



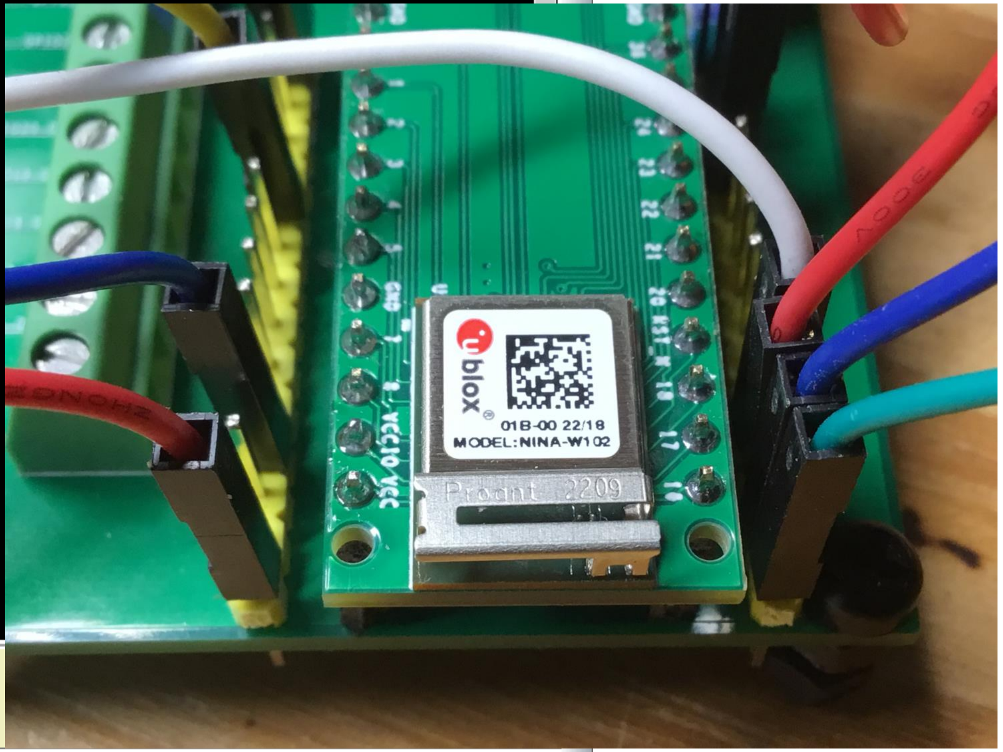
```
#include <SPI.h>
#include <WiFiNINA.h>
#include "arduino_secrets.h"
// enter your sensitive data in arduino_secrets.h
char ssid[] = SECRET_SSID;    // your network SSID
char pass[] = SECRET_PASS;    // your network password
int status = WL_IDLE_STATUS;  // the WiFi radio's status

void setup() {
  //Initialize serial and wait for port to open:
  Serial1.begin(9600);
  while(!Serial1);
   // check for the WiFi module:
  if (WiFi.status() == WL_NO_MODULE) {
    Serial1.println("Communication with WiFi module failed!");
    // spin here and don't continue
    while (true);
  }
  // check NINA firmware version
  String fv = WiFi.firmwareVersion();
  if (fv < WIFI_FIRMWARE_LATEST_VERSION) {
    Serial1.println("Please upgrade the firmware");
  }
  // attempt to connect to WiFi network:
  while (status != WL_CONNECTED) {
    Serial1.print("Attempting to connect to WPA SSID: ");
    Serial1.println(ssid);
    // Connect to WPA/WPA2 network:
    status = WiFi.begin(ssid, pass);
    // wait 10 seconds for connection:
    delay(10000);
  }
  // you're connected now, so print out the data:
  Serial1.println("You are connected to the network!!\r\n");
  printCurrentNet();
  printWifiData();
  Serial1.println();
}
```



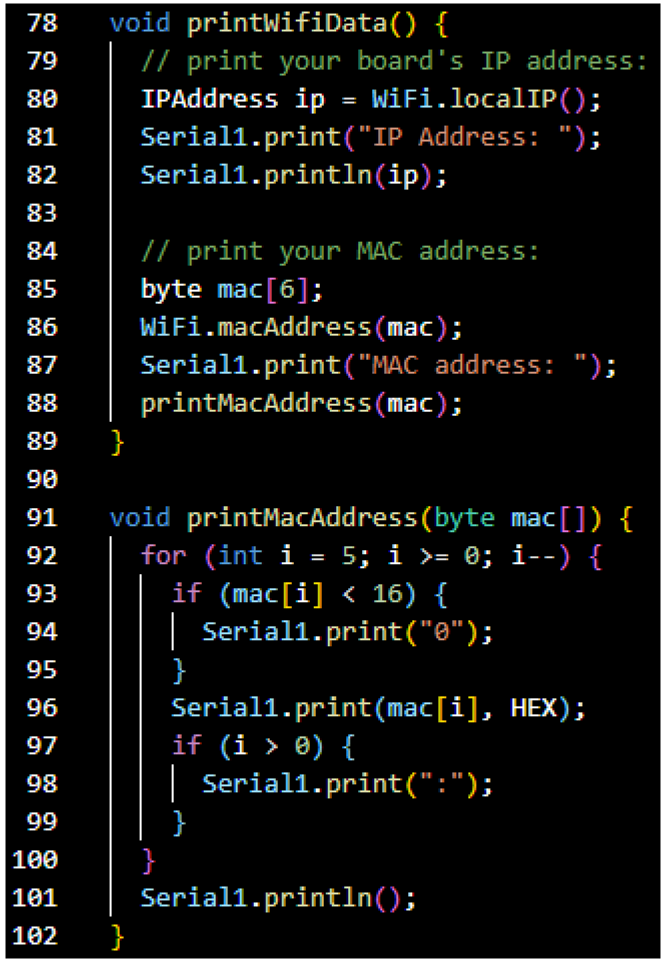
```
Attempting to connect to WPA SSID: edtpnet2
You are connected to the network!!

SSID: edtpnet2
BSSID: C0:56:27:4F:6A:89
Signal Strength (RSSI):-20
Encryption Type:4

IP Address: 192.168.1.75
MAC address: 58:BF:25:0E:E6:2C

SSID: edtpnet2
BSSID: C0:56:27:4F:6A:89
Signal Strength (RSSI):-20
Encryption Type:4

SSID: edtpnet2
BSSID: C0:56:27:4F:6A:89
Signal Strength (RSSI):-19
Encryption Type:4

SSID: edtpnet2
BSSID: C0:56:27:4F:6A:89
Signal Strength (RSSI):-20
Encryption Type:4

SSID: edtpnet2
BSSID: C0:56:27:4F:6A:89
Signal Strength (RSSI):-19
Encryption Type:4
```

```
45 6E 63 72 79 70 74 69 6F 6E 20 54 79 70 65 3A 34 0D 0A
0D 0A
```

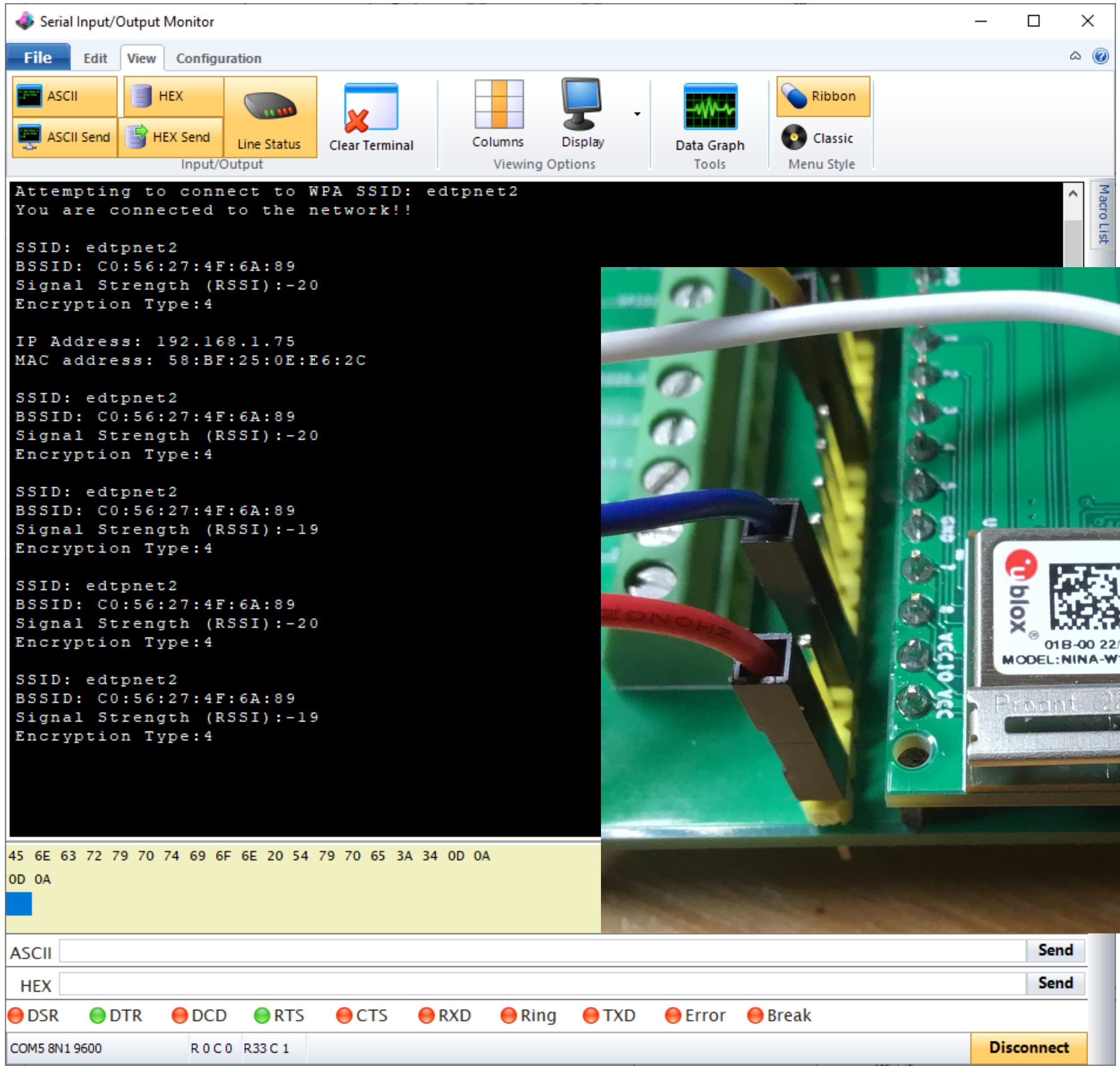
COM5 8N1 9600 R 0 C 0 R33 C 1



Arduino Wi-Fi First Contact – setup() functions

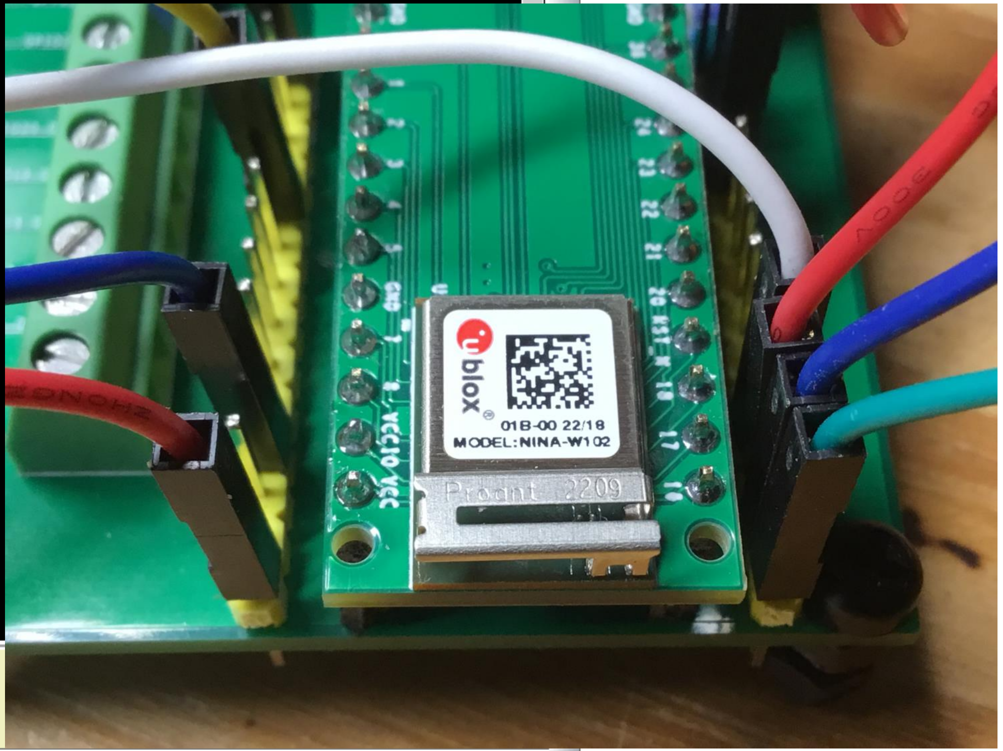


```
void printWifiData() {
    // print your board's IP address:
    IPAddress ip = WiFi.localIP();
    Serial1.print("IP Address: ");
    Serial1.println(ip);

    // print your MAC address:
    byte mac[6];
    WiFi.macAddress(mac);
    Serial1.print("MAC address: ");
    printMacAddress(mac);
}

void printMacAddress(byte mac[]) {
    for (int i = 5; i >= 0; i--) {
        if (mac[i] < 16) {
            Serial1.print("0");
        }
        Serial1.print(mac[i], HEX);
        if (i > 0) {
            Serial1.print(":");
        }
    }
    Serial1.println();
}
```

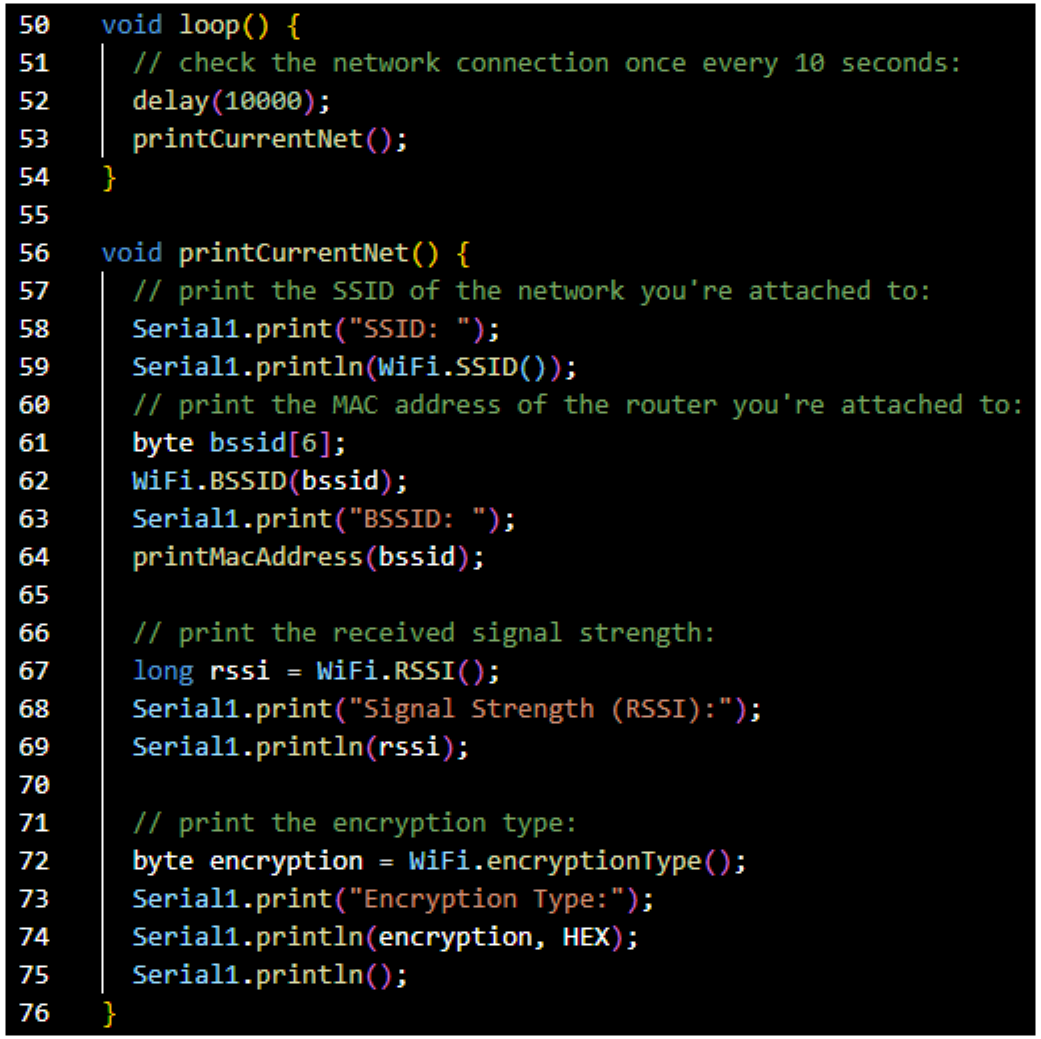


```
Attempting to connect to WPA SSID: edtpnet2
You are connected to the network!!

SSID: edtpnet2
BSSID: C0:56:27:4F:6A:89
Signal Strength (RSSI):-20
Encryption Type:4

IP Address: 192.168.1.75
MAC address: 58:BF:25:0E:E6:2C

SSID: edtpnet2
BSSID: C0:56:27:4F:6A:89
Signal Strength (RSSI):-20
Encryption Type:4

SSID: edtpnet2
BSSID: C0:56:27:4F:6A:89
Signal Strength (RSSI):-19
Encryption Type:4

SSID: edtpnet2
BSSID: C0:56:27:4F:6A:89
Signal Strength (RSSI):-20
Encryption Type:4

SSID: edtpnet2
BSSID: C0:56:27:4F:6A:89
Signal Strength (RSSI):-19
Encryption Type:4
```

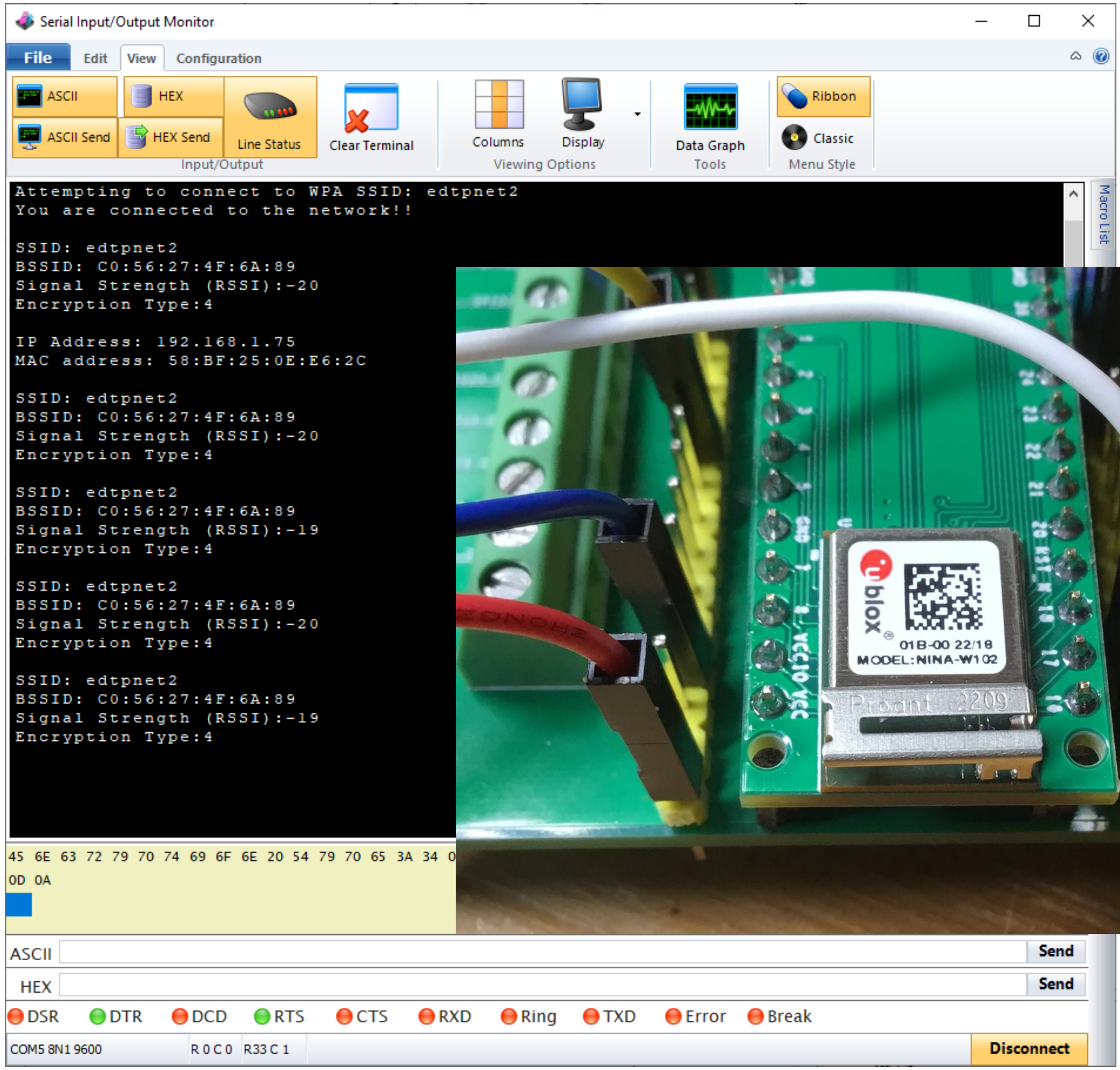
```
45 6E 63 72 79 70 74 69 6F 6E 20 54 79 70 65 3A 34 0D 0A
0D 0A
```



Arduino Wi-Fi First Contact – loop()



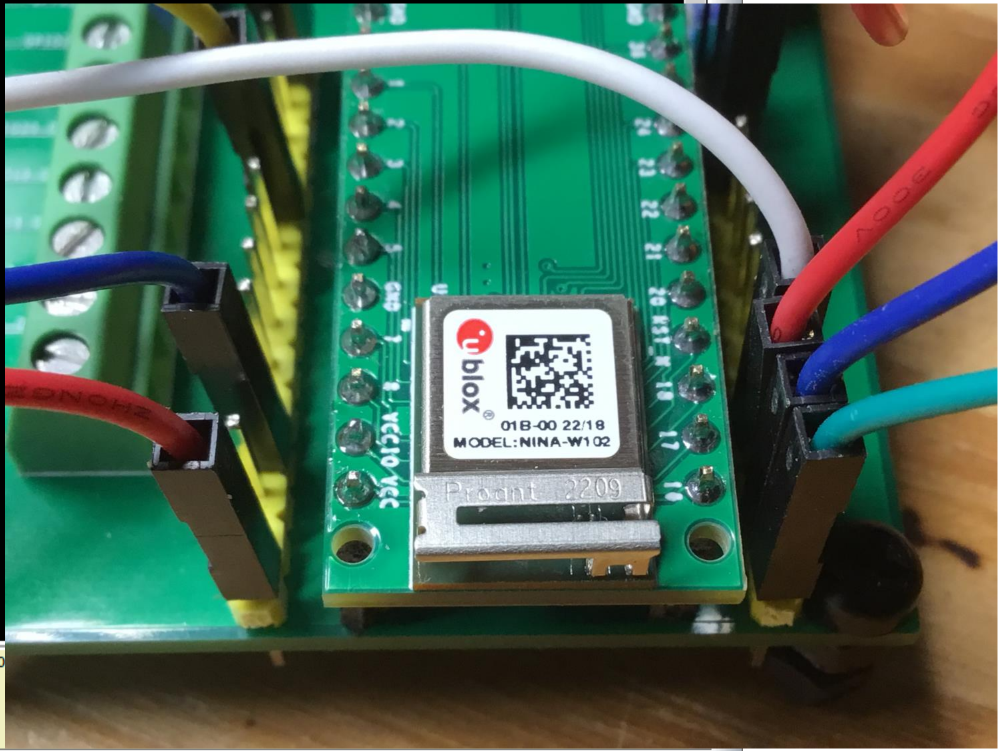
```
void loop() {
  // check the network connection once every 10 seconds:
  delay(10000);
  printCurrentNet();
}

void printCurrentNet() {
  // print the SSID of the network you're attached to:
  Serial1.print("SSID: ");
  Serial1.println(WiFi.SSID());
  // print the MAC address of the router you're attached to:
  byte bssid[6];
  WiFi.BSSID(bssid);
  Serial1.print("BSSID: ");
  printMacAddress(bssid);

  // print the received signal strength:
  long rssi = WiFi.RSSI();
  Serial1.print("Signal Strength (RSSI):");
  Serial1.println(rssi);

  // print the encryption type:
  byte encryption = WiFi.encryptionType();
  Serial1.print("Encryption Type:");
  Serial1.println(encryption, HEX);
  Serial1.println();
}
```



```
Attempting to connect to WPA SSID: edtpnet2
You are connected to the network!!

SSID: edtpnet2
BSSID: C0:56:27:4F:6A:89
Signal Strength (RSSI):-20
Encryption Type:4

IP Address: 192.168.1.75
MAC address: 58:BF:25:0E:E6:2C

SSID: edtpnet2
BSSID: C0:56:27:4F:6A:89
Signal Strength (RSSI):-20
Encryption Type:4

SSID: edtpnet2
BSSID: C0:56:27:4F:6A:89
Signal Strength (RSSI):-19
Encryption Type:4

SSID: edtpnet2
BSSID: C0:56:27:4F:6A:89
Signal Strength (RSSI):-20
Encryption Type:4

SSID: edtpnet2
BSSID: C0:56:27:4F:6A:89
Signal Strength (RSSI):-19
Encryption Type:4
```

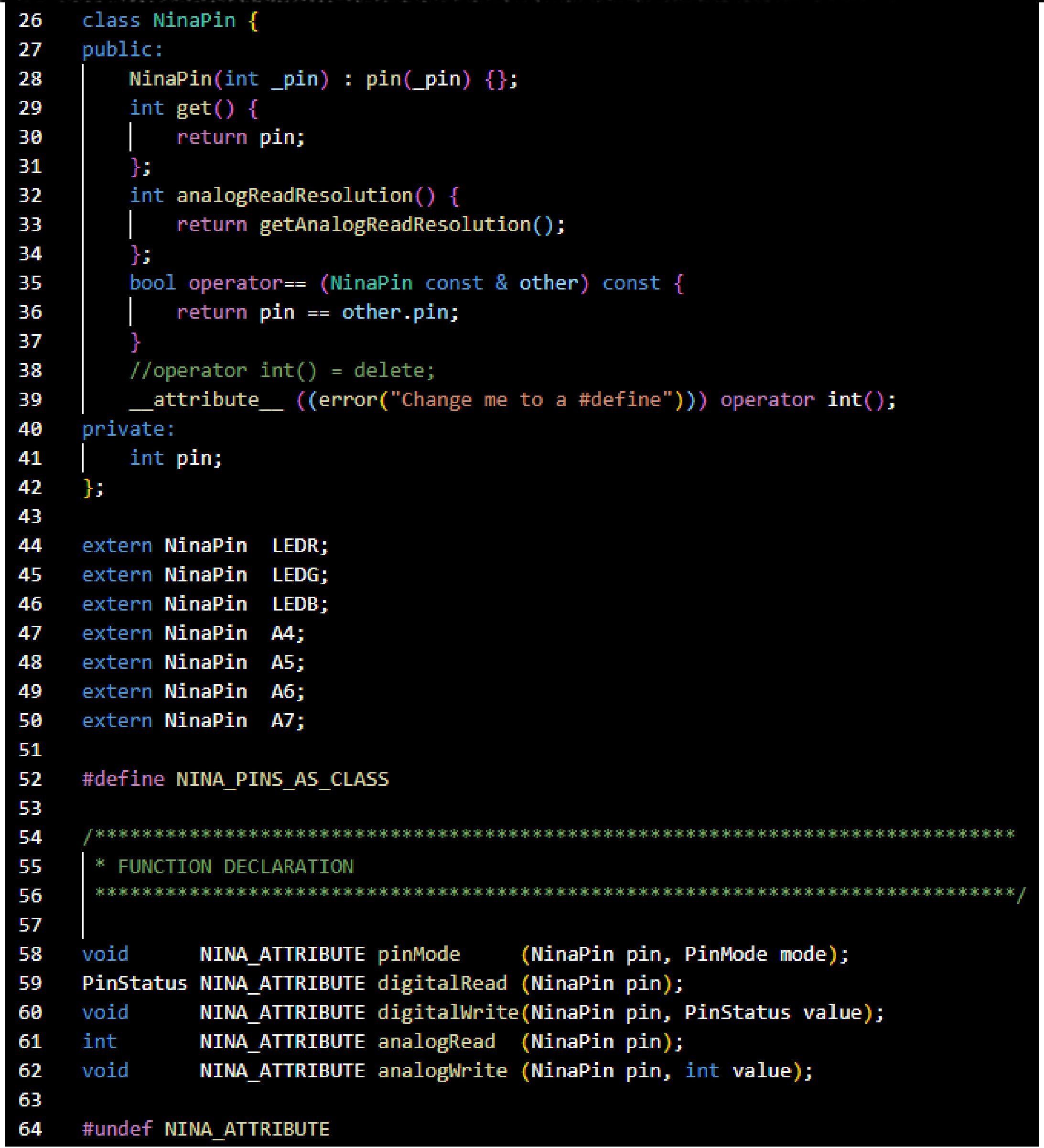
```
45 6E 63 72 79 70 74 69 6F 6E 20 54 79 70 65 3A 34 0
0D 0A
```

COM5 8N1 9600   R 0 C 0   R33 C 1

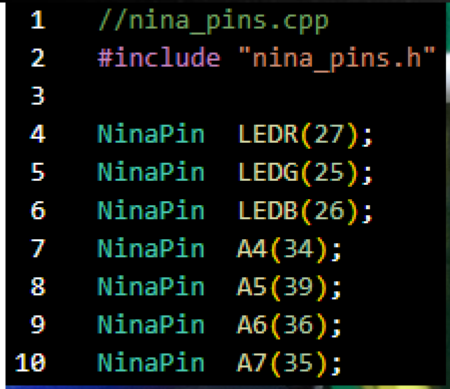


Designing, Building and Coding Custom Raspberry Pi RP2040 Arduino Devices
The RP2040, Arduino and Wi-Fi
Wi-FI Remote Control

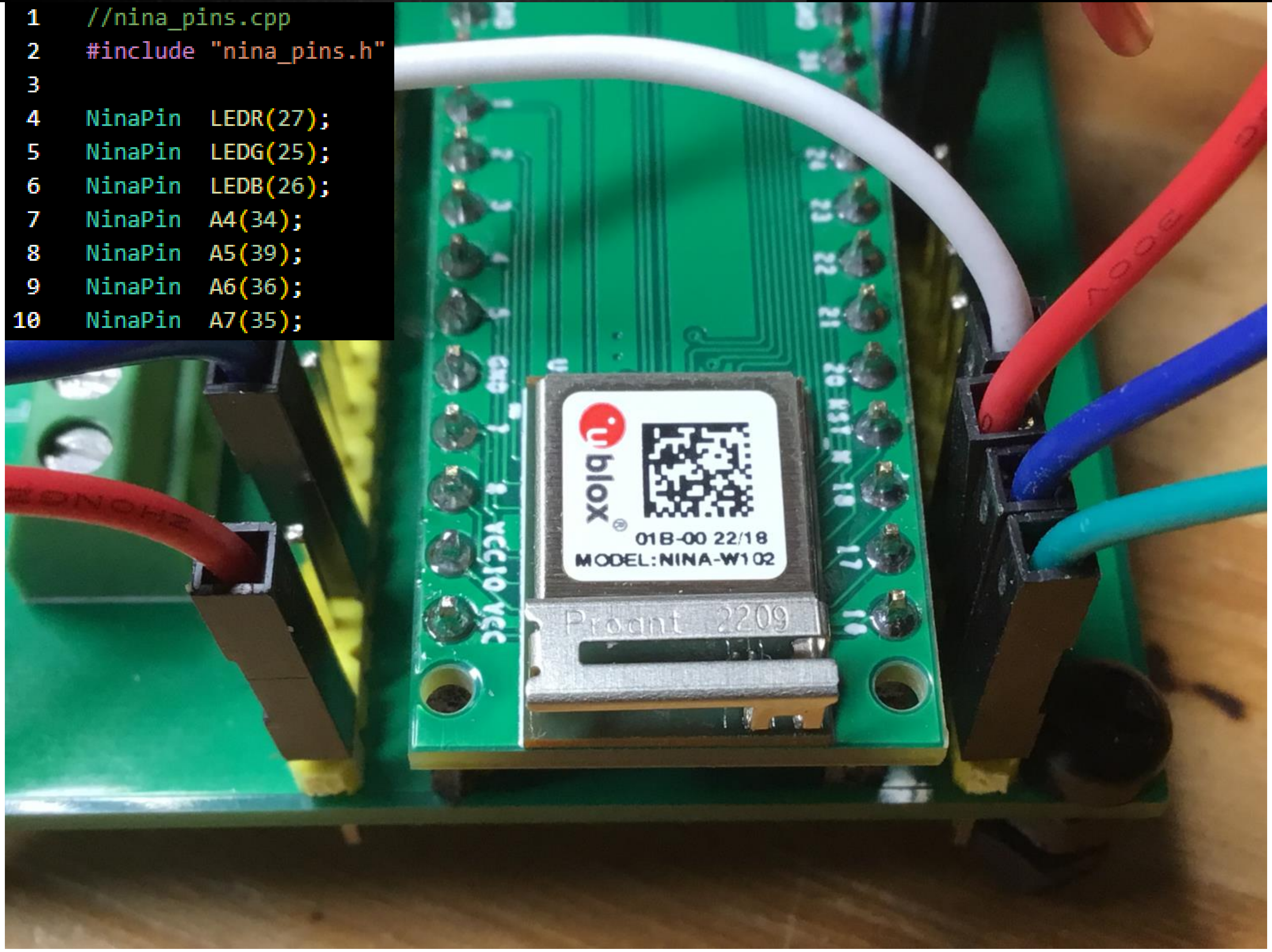
Driving NINA GPIO Pins – nina_pins.cpp/nina_pins.h



```
class NinaPin {
public:
    NinaPin(int _pin) : pin(_pin) {};
    int get() {
        return pin;
    };
    int analogReadResolution() {
        return getAnalogReadResolution();
    };
    bool operator== (NinaPin const & other) const {
        return pin == other.pin;
    }
    //operator int() = delete;
    __attribute__ ((error("Change me to a #define"))) operator int();
private:
    int pin;
};

extern NinaPin  LEDR;
extern NinaPin  LEDG;
extern NinaPin  LEDB;
extern NinaPin  A4;
extern NinaPin  A5;
extern NinaPin  A6;
extern NinaPin  A7;

#define NINA_PINS_AS_CLASS

/******************************************************************************
 * FUNCTION DECLARATION
 ******************************************************************************/

void       NINA_ATTRIBUTE pinMode     (NinaPin pin, PinMode mode);
PinStatus NINA_ATTRIBUTE digitalRead (NinaPin pin);
void       NINA_ATTRIBUTE digitalWrite(NinaPin pin, PinStatus value);
int        NINA_ATTRIBUTE analogRead  (NinaPin pin);
void       NINA_ATTRIBUTE analogWrite (NinaPin pin, int value);

#undef NINA_ATTRIBUTE
```



```
//nina_pins.cpp
#include "nina_pins.h"

NinaPin  LEDR(27);
NinaPin  LEDG(25);
NinaPin  LEDB(26);
NinaPin  A4(34);
NinaPin  A5(39);
NinaPin  A6(36);
NinaPin  A7(35);
```



CEC Continuing Education Center

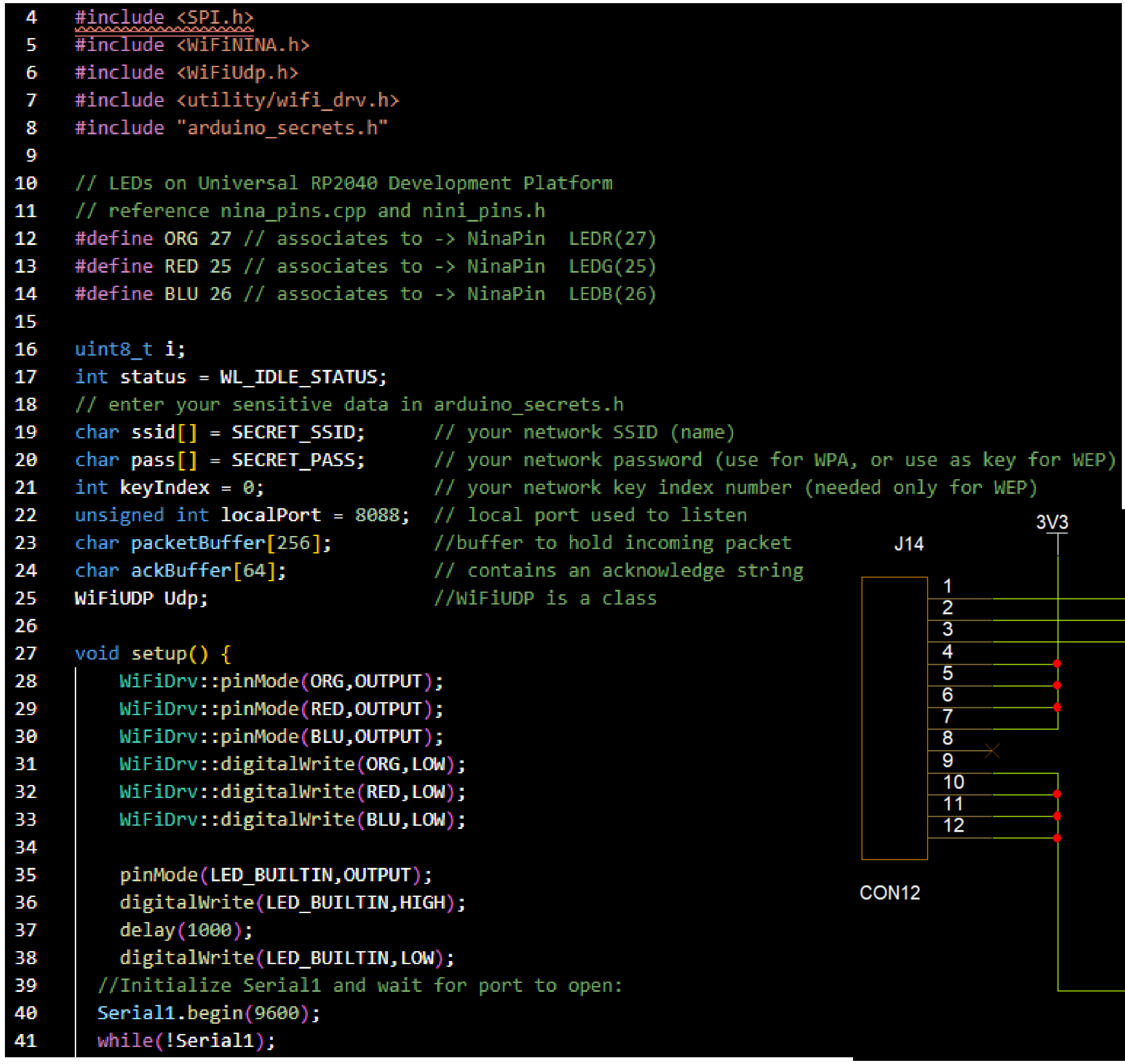
Designing, Building and Coding Custom Raspberry Pi RP2040 Arduino Devices

The RP2040, Arduino and Wi-Fi

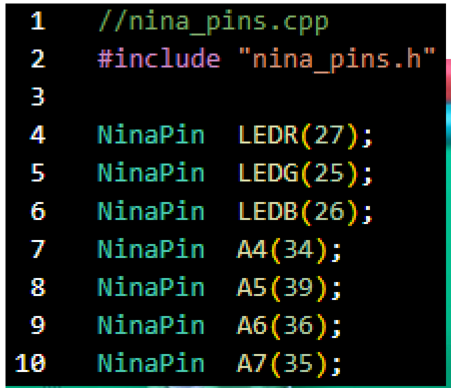
Wi-FI Remote Control

DigiKey

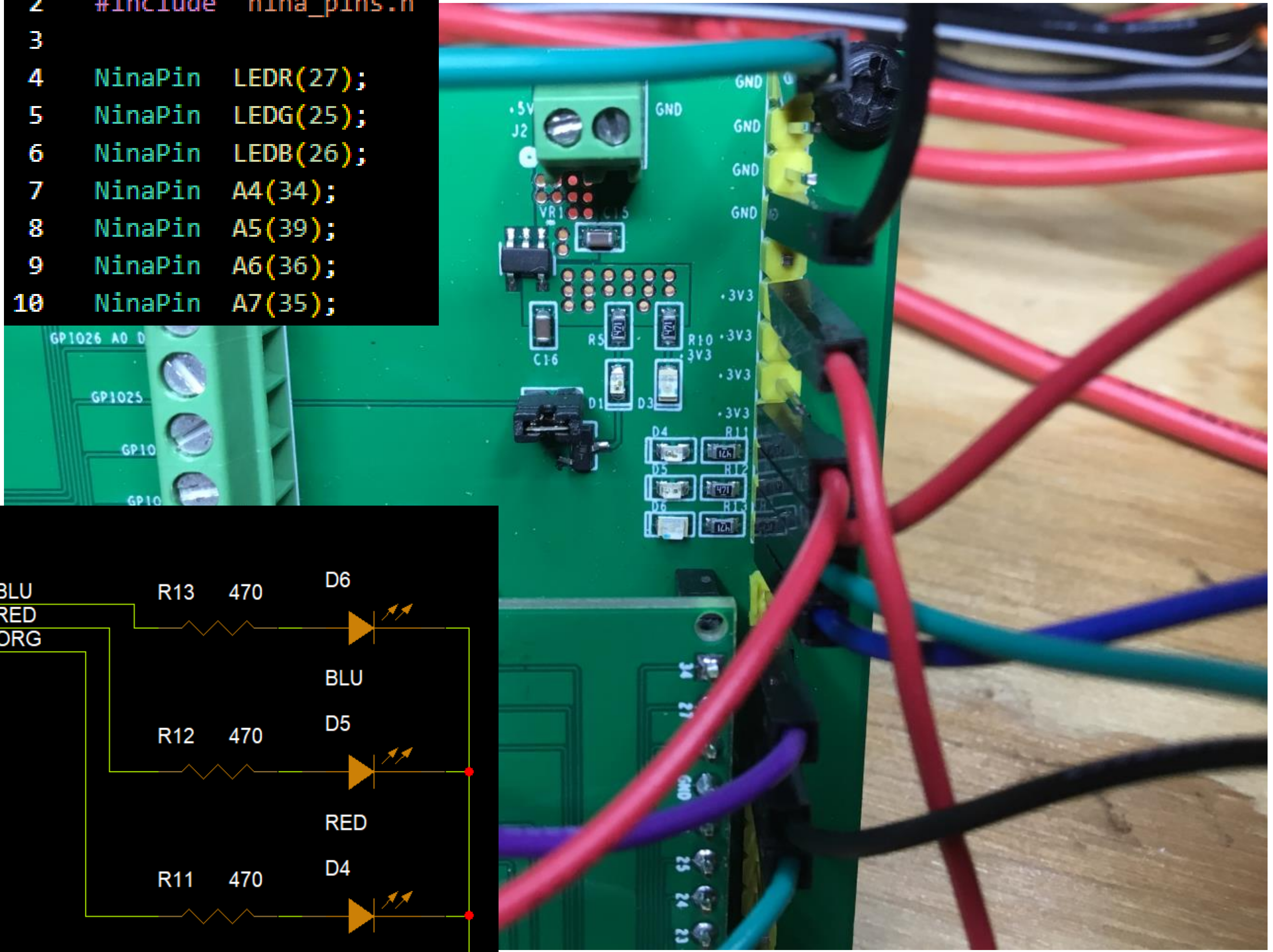
Driving NINA GPIO Pins

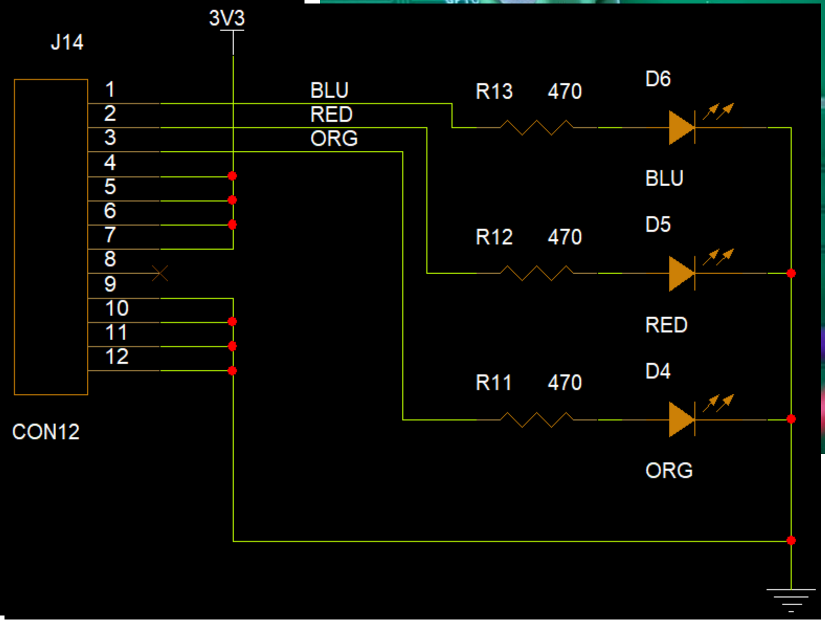


```
#include <SPI.h>
#include <WiFiNINA.h>
#include <WiFiUdp.h>
#include <utility/wifi_drv.h>
#include "arduino_secrets.h"

// LEDs on Universal RP2040 Development Platform
// reference nina_pins.cpp and nini_pins.h
#define ORG 27 // associates to -> NinaPin  LEDR(27)
#define RED 25 // associates to -> NinaPin  LEDG(25)
#define BLU 26 // associates to -> NinaPin  LEDB(26)

uint8_t i;
int status = WL_IDLE_STATUS;
// enter your sensitive data in arduino_secrets.h
char ssid[] = SECRET_SSID;      // your network SSID (name)
char pass[] = SECRET_PASS;      // your network password (use for WPA, or use as key for WEP)
int keyIndex = 0;               // your network key index number (needed only for WEP)
unsigned int localPort = 8088;  // local port used to listen
char packetBuffer[256];         //buffer to hold incoming packet
char ackBuffer[64];             // contains an acknowledge string
WiFiUDP Udp;                    //WiFiUDP is a class

void setup() {
    WiFiDrv::pinMode(ORG,OUTPUT);
    WiFiDrv::pinMode(RED,OUTPUT);
    WiFiDrv::pinMode(BLU,OUTPUT);
    WiFiDrv::digitalWrite(ORG,LOW);
    WiFiDrv::digitalWrite(RED,LOW);
    WiFiDrv::digitalWrite(BLU,LOW);

    pinMode(LED_BUILTIN,OUTPUT);
    digitalWrite(LED_BUILTIN,HIGH);
    delay(1000);
    digitalWrite(LED_BUILTIN,LOW);
  //Initialize Serial1 and wait for port to open:
  Serial1.begin(9600);
  while(!Serial1);
```

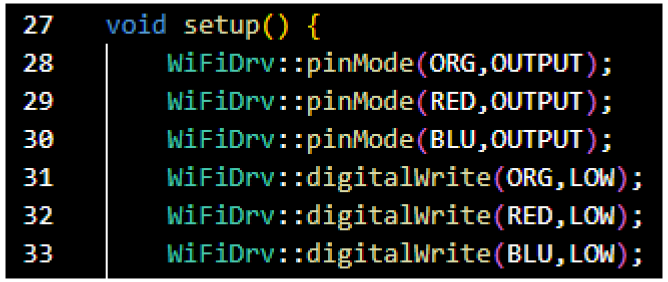


```
//nina_pins.cpp
#include "nina_pins.h"

NinaPin  LEDR(27);
NinaPin  LEDG(25);
NinaPin  LEDB(26);
NinaPin  A4(34);
NinaPin  A5(39);
NinaPin  A6(36);
NinaPin  A7(35);
```

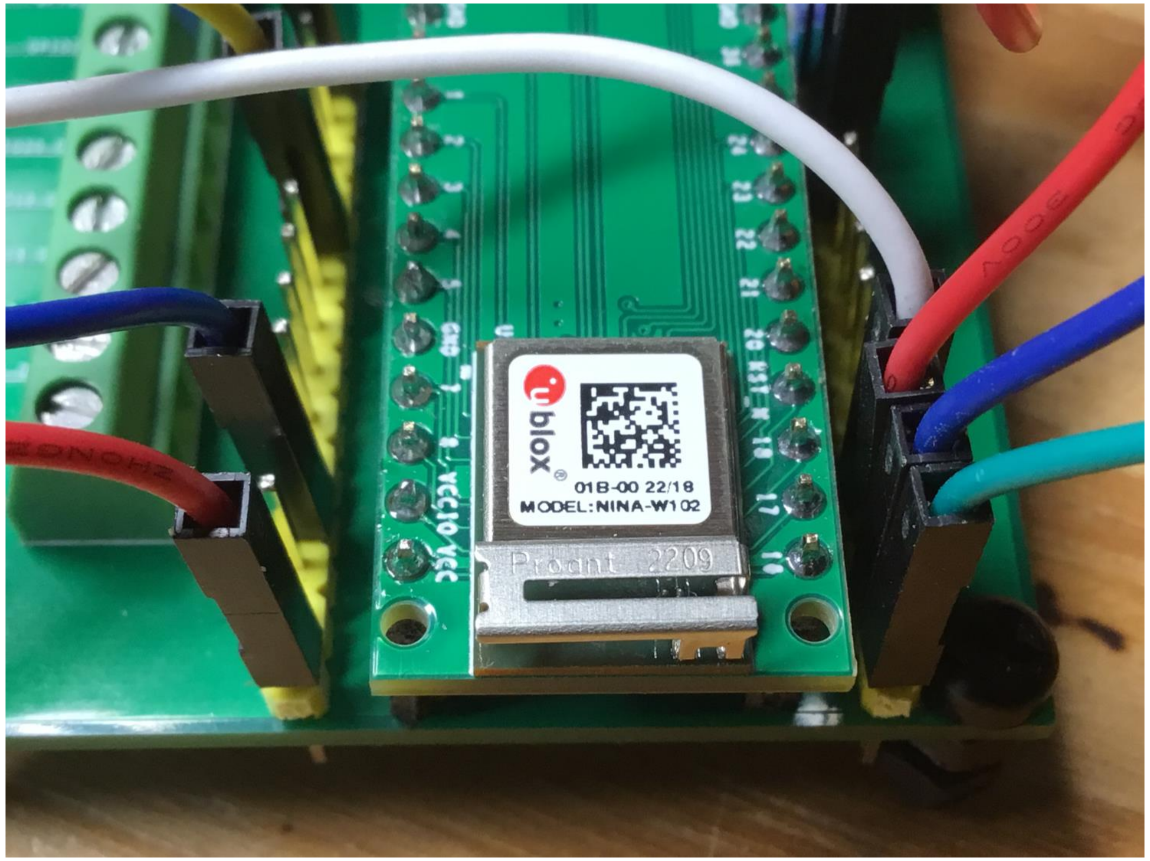


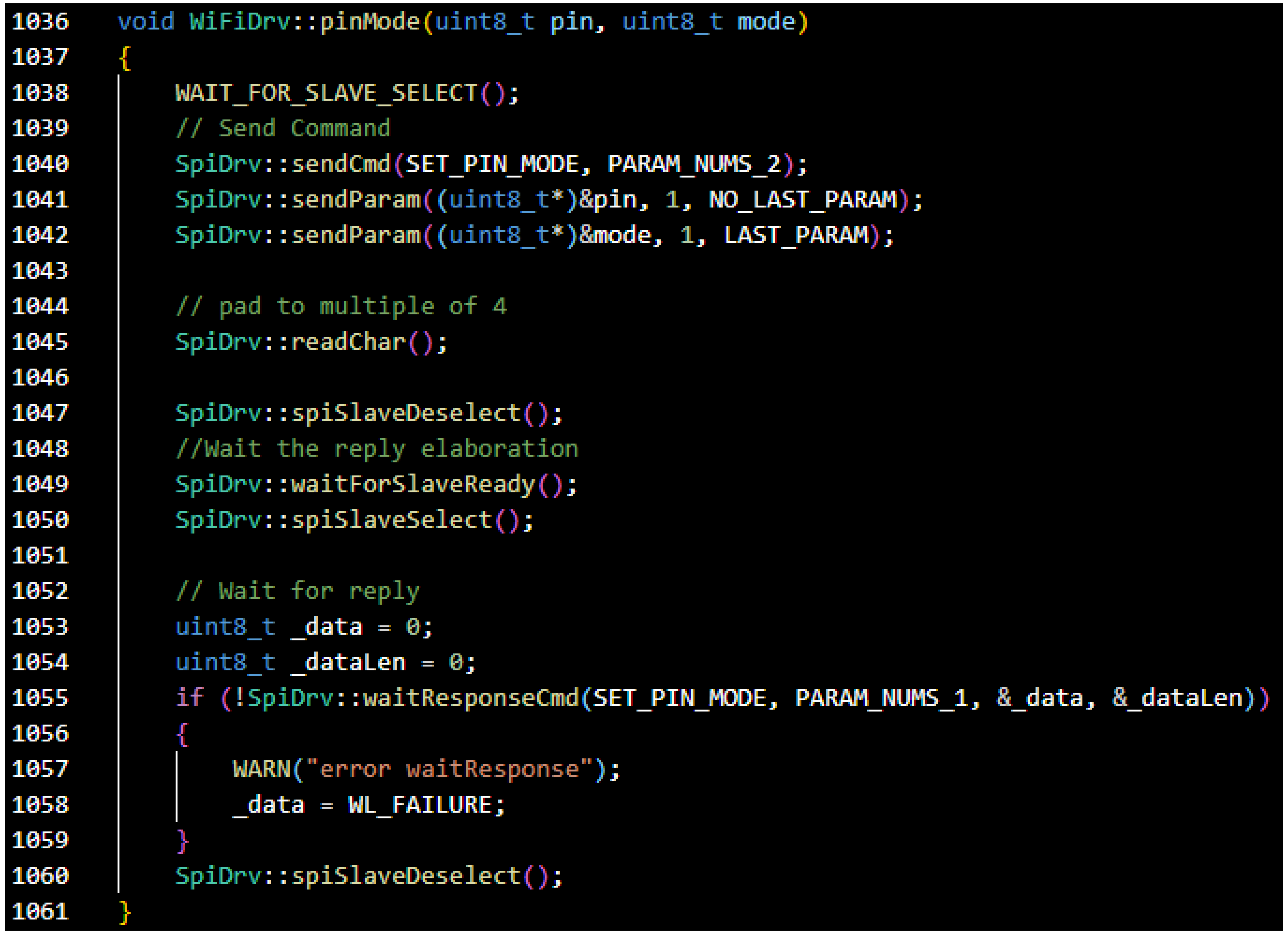


Driving NINA GPIO Pins



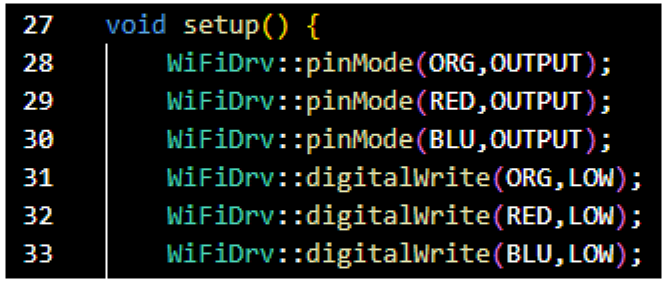
```
void setup() {
    WiFiDrv::pinMode(ORG,OUTPUT);
    WiFiDrv::pinMode(RED,OUTPUT);
    WiFiDrv::pinMode(BLU,OUTPUT);
    WiFiDrv::digitalWrite(ORG,LOW);
    WiFiDrv::digitalWrite(RED,LOW);
    WiFiDrv::digitalWrite(BLU,LOW);
```



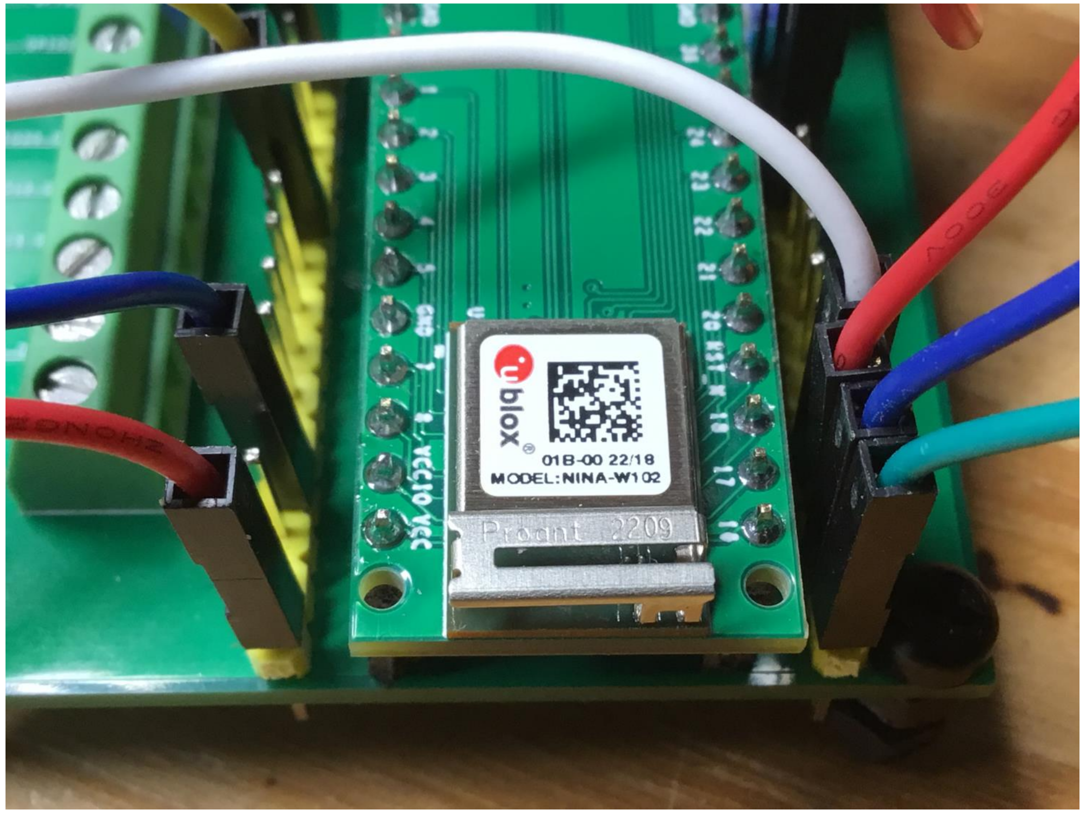


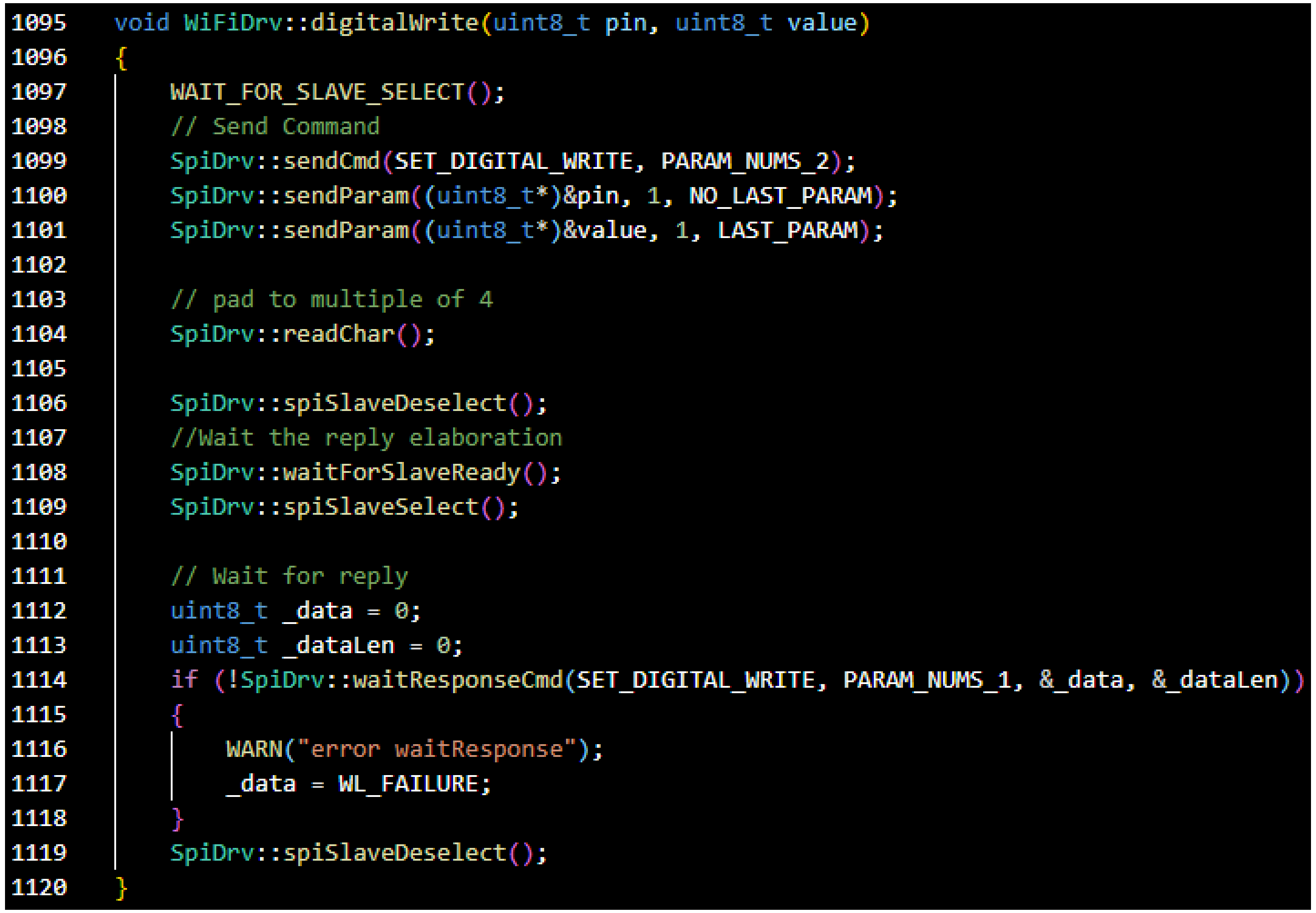
```
void WiFiDrv::pinMode(uint8_t pin, uint8_t mode)
{
    WAIT_FOR_SLAVE_SELECT();
    // Send Command
    SpiDrv::sendCmd(SET_PIN_MODE, PARAM_NUMS_2);
    SpiDrv::sendParam((uint8_t*)&pin, 1, NO_LAST_PARAM);
    SpiDrv::sendParam((uint8_t*)&mode, 1, LAST_PARAM);

    // pad to multiple of 4
    SpiDrv::readChar();

    SpiDrv::spiSlaveDeselect();
    //Wait the reply elaboration
    SpiDrv::waitForSlaveReady();
    SpiDrv::spiSlaveSelect();

    // Wait for reply
    uint8_t _data = 0;
    uint8_t _dataLen = 0;
    if (!SpiDrv::waitResponseCmd(SET_PIN_MODE, PARAM_NUMS_1, &_data, &_dataLen))
    {
        WARN("error waitResponse");
        _data = WL_FAILURE;
    }
    SpiDrv::spiSlaveDeselect();
}
```

Driving NINA GPIO Pins



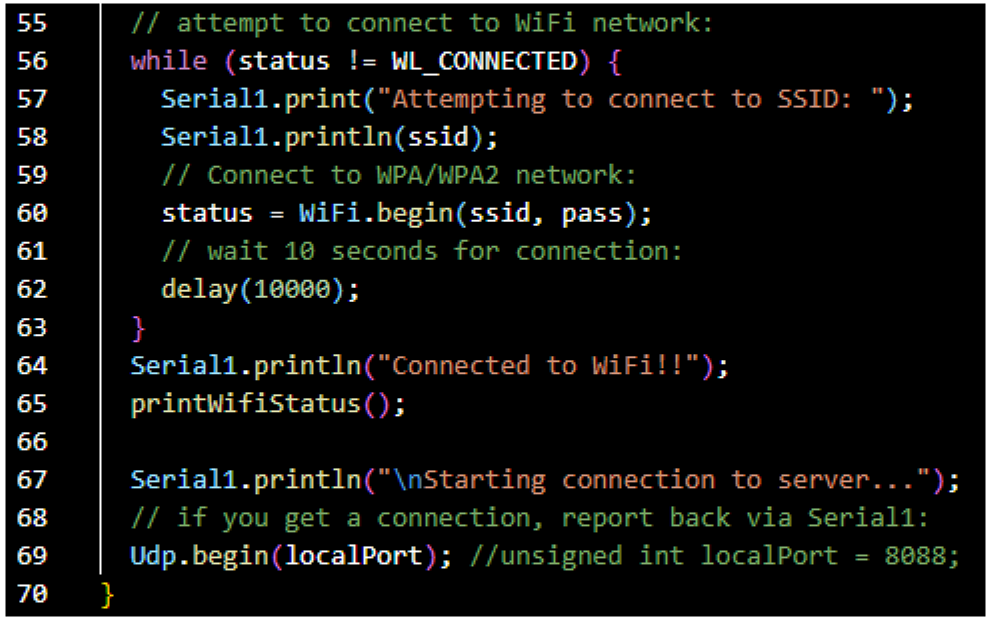
```
void setup() {
    WiFiDrv::pinMode(ORG,OUTPUT);
    WiFiDrv::pinMode(RED,OUTPUT);
    WiFiDrv::pinMode(BLU,OUTPUT);
    WiFiDrv::digitalWrite(ORG,LOW);
    WiFiDrv::digitalWrite(RED,LOW);
    WiFiDrv::digitalWrite(BLU,LOW);
```



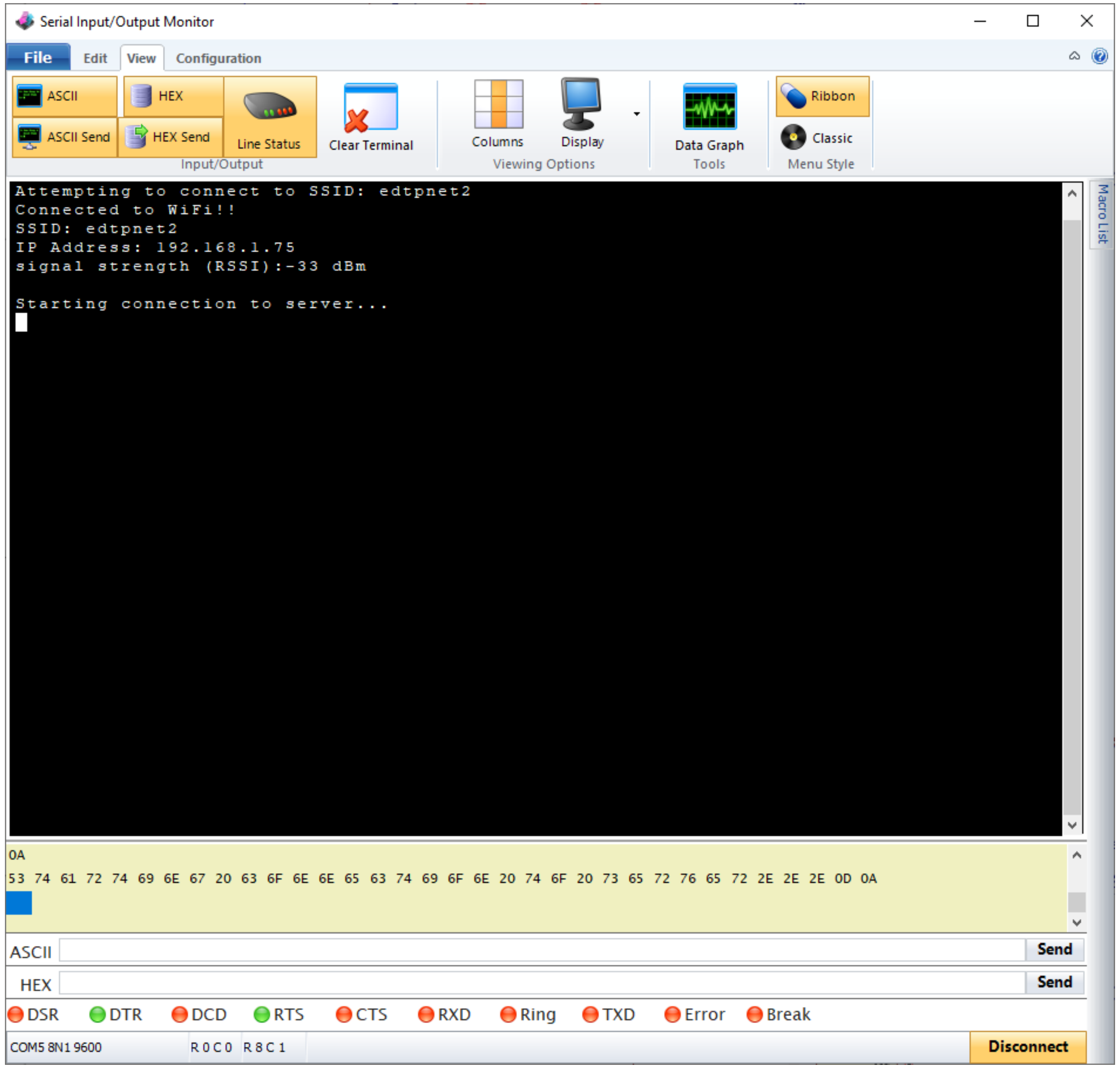


```
void WiFiDrv::digitalWrite(uint8_t pin, uint8_t value)
{
    WAIT_FOR_SLAVE_SELECT();
    // Send Command
    SpiDrv::sendCmd(SET_DIGITAL_WRITE, PARAM_NUMS_2);
    SpiDrv::sendParam((uint8_t*)&pin, 1, NO_LAST_PARAM);
    SpiDrv::sendParam((uint8_t*)&value, 1, LAST_PARAM);

    // pad to multiple of 4
    SpiDrv::readChar();

    SpiDrv::spiSlaveDeselect();
    //Wait the reply elaboration
    SpiDrv::waitForSlaveReady();
    SpiDrv::spiSlaveSelect();

    // Wait for reply
    uint8_t _data = 0;
    uint8_t _dataLen = 0;
    if (!SpiDrv::waitResponseCmd(SET_DIGITAL_WRITE, PARAM_NUMS_1, &_data, &_dataLen))
    {
        WARN("error waitResponse");
        _data = WL_FAILURE;
    }
    SpiDrv::spiSlaveDeselect();
}
```

Start and Listen on Local Port 8088



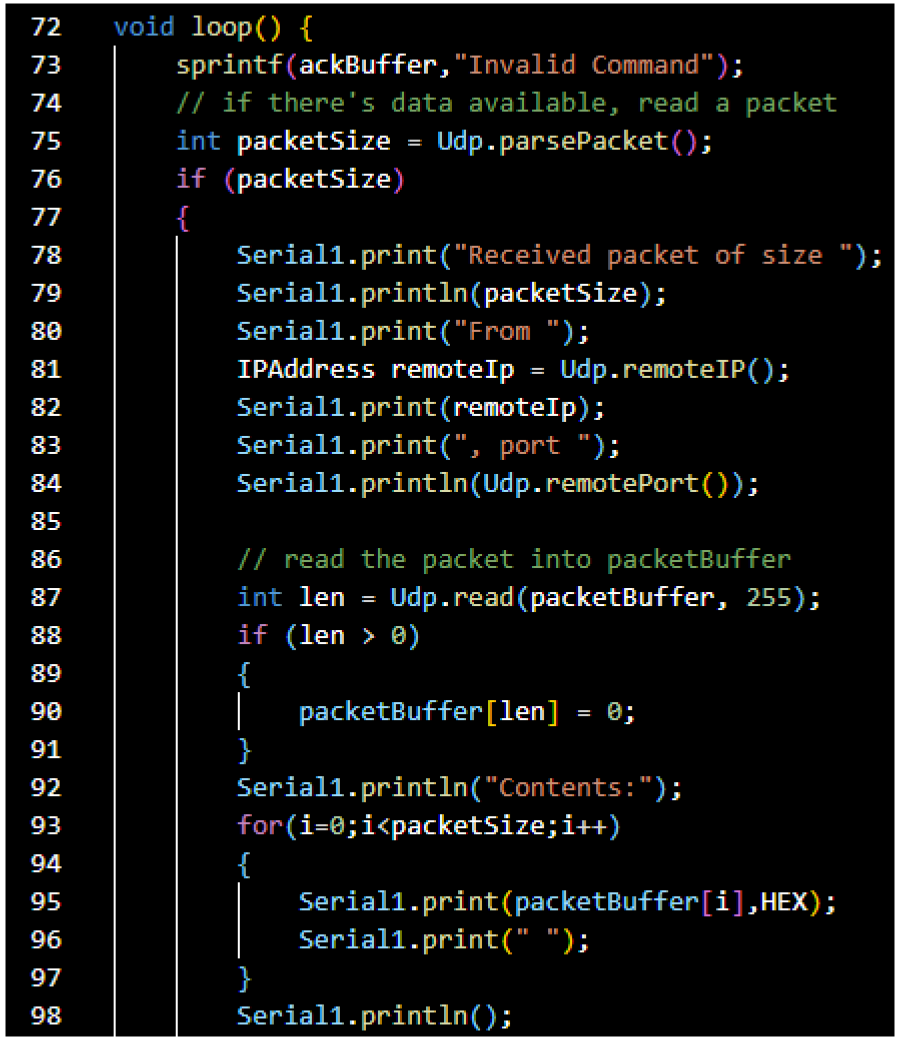
```
  // attempt to connect to WiFi network:
  while (status != WL_CONNECTED) {
    Serial1.print("Attempting to connect to SSID: ");
    Serial1.println(ssid);
    // Connect to WPA/WPA2 network:
    status = WiFi.begin(ssid, pass);
    // wait 10 seconds for connection:
    delay(10000);
  }
  Serial1.println("Connected to WiFi!!");
  printWifiStatus();

  Serial1.println("\nStarting connection to server...");
  // if you get a connection, report back via Serial1:
  Udp.begin(localPort); //unsigned int localPort = 8088;
}
```



```
Attempting to connect to SSID: edtpnet2
Connected to WiFi!!
SSID: edtpnet2
IP Address: 192.168.1.75
signal strength (RSSI):-33 dBm

Starting connection to server...
```

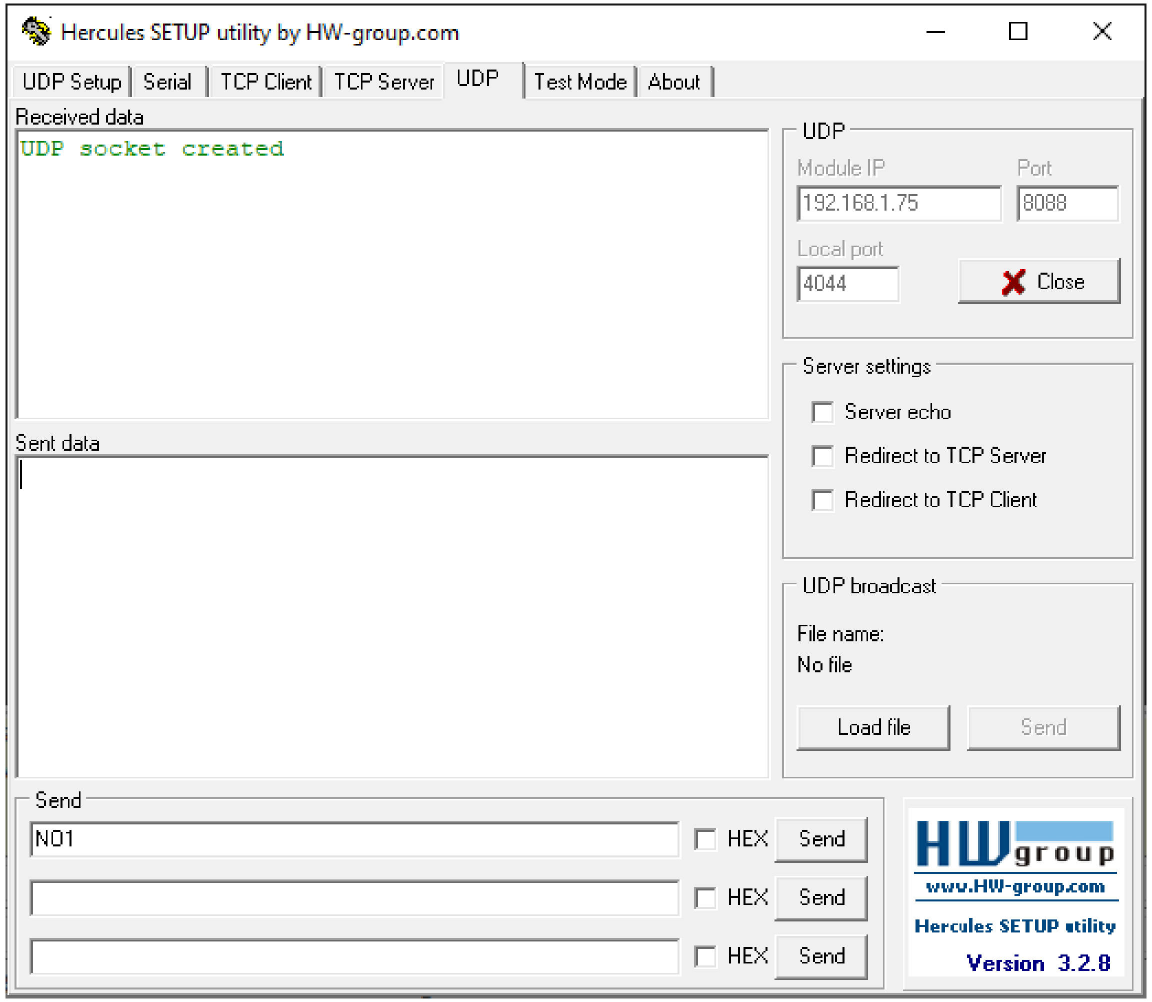
```
0A
53 74 61 72 74 69 6E 67 20 63 6F 6E 6E 65 63 74 69 6F 6E 20 74 6F 20 73 65 72 76 65 72 2E 2E 2E 0D 0A
```

Setup UDP Client/Server



```
void loop() {
    sprintf(ackBuffer,"Invalid Command");
    // if there's data available, read a packet
    int packetSize = Udp.parsePacket();
    if (packetSize)
    {
        Serial1.print("Received packet of size ");
        Serial1.println(packetSize);
        Serial1.print("From ");
        IPAddress remoteIp = Udp.remoteIP();
        Serial1.print(remoteIp);
        Serial1.print(", port ");
        Serial1.println(Udp.remotePort());

        // read the packet into packetBuffer
        int len = Udp.read(packetBuffer, 255);
        if (len > 0)
        {
            packetBuffer[len] = 0;
        }
        Serial1.println("Contents:");
        for(i=0;i<packetSize;i++)
        {
            Serial1.print(packetBuffer[i],HEX);
            Serial1.print(" ");
        }
        Serial1.println();
```

COMMAND FROM PC



Received data

```
UDP socket created
```

Sent data

UDP
Module IP: 192.168.1.75
Port: 8088
Local port: 4044

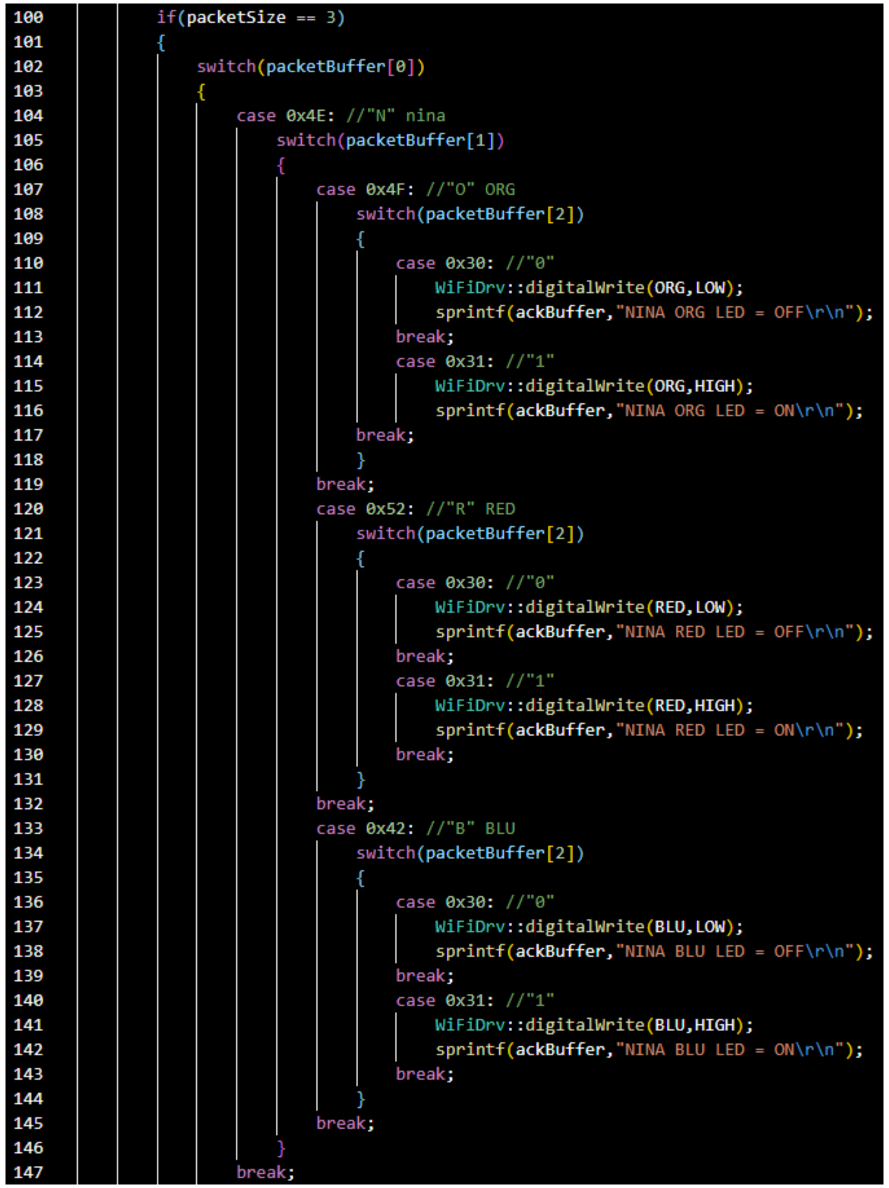
Server settings
- [ ] Server echo
- [ ] Redirect to TCP Server
- [ ] Redirect to TCP Client

UDP broadcast
File name:
No file

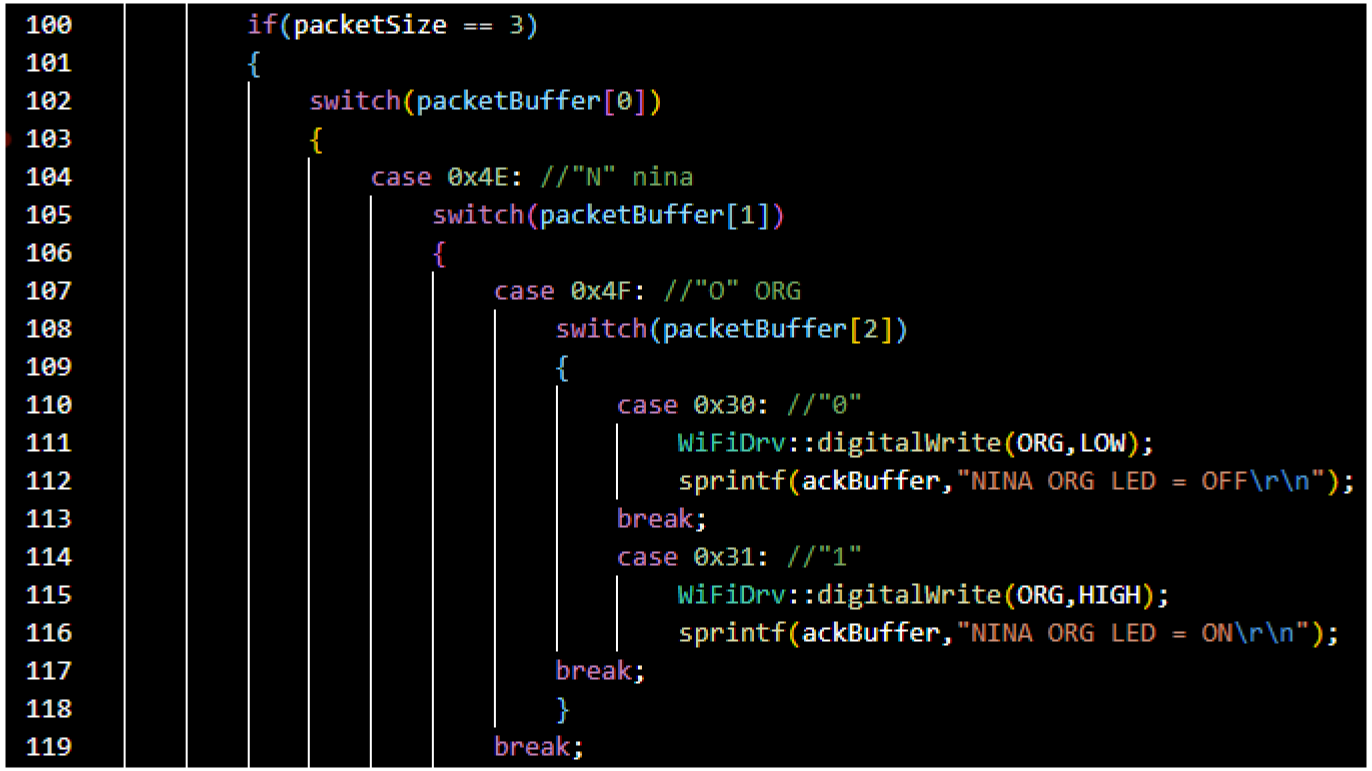
Send: NO1

ACKNOWLEDGEMENT FROM RP2040

Parse Received Command to Turn on the ORG LED



```
if(packetSize == 3)
{
    switch(packetBuffer[0])
    {
        case 0x4E: //"N" nina
            switch(packetBuffer[1])
            {
                case 0x4F: //"O" ORG
                    switch(packetBuffer[2])
                    {
                        case 0x30: //"0"
                            WiFiDrv::digitalWrite(ORG,LOW);
                            sprintf(ackBuffer,"NINA ORG LED = OFF\r\n");
                        break;
                        case 0x31: //"1"
                            WiFiDrv::digitalWrite(ORG,HIGH);
                            sprintf(ackBuffer,"NINA ORG LED = ON\r\n");
                    break;
                    }
                break;
                case 0x52: //"R" RED
                    switch(packetBuffer[2])
                    {
                        case 0x30: //"0"
                            WiFiDrv::digitalWrite(RED,LOW);
                            sprintf(ackBuffer,"NINA RED LED = OFF\r\n");
                        break;
                        case 0x31: //"1"
                            WiFiDrv::digitalWrite(RED,HIGH);
                            sprintf(ackBuffer,"NINA RED LED = ON\r\n");
                        break;
                    }
                break;
                case 0x42: //"B" BLU
                    switch(packetBuffer[2])
                    {
                        case 0x30: //"0"
                            WiFiDrv::digitalWrite(BLU,LOW);
                            sprintf(ackBuffer,"NINA BLU LED = OFF\r\n");
                        break;
                        case 0x31: //"1"
                            WiFiDrv::digitalWrite(BLU,HIGH);
                            sprintf(ackBuffer,"NINA BLU LED = ON\r\n");
                        break;
                    }
                break;
            }
        break;
```



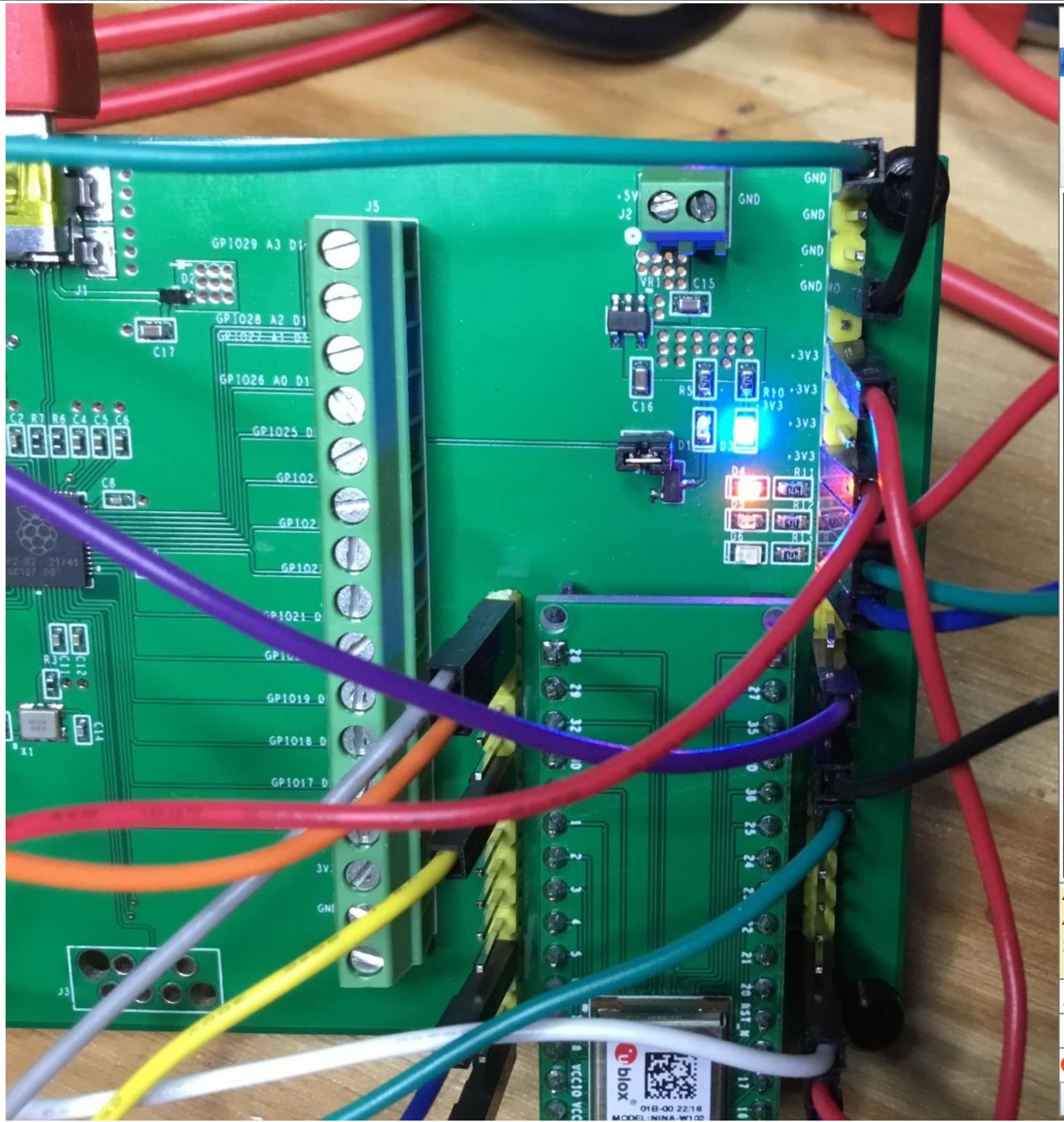
```
if(packetSize == 3)
{
    switch(packetBuffer[0])
    {
        case 0x4E: //"N" nina
            switch(packetBuffer[1])
            {
                case 0x4F: //"O" ORG
                    switch(packetBuffer[2])
                    {
                        case 0x30: //"0"
                            WiFiDrv::digitalWrite(ORG,LOW);
                            sprintf(ackBuffer,"NINA ORG LED = OFF\r\n");
                        break;
                        case 0x31: //"1"
                            WiFiDrv::digitalWrite(ORG,HIGH);
                            sprintf(ackBuffer,"NINA ORG LED = ON\r\n");
                    break;
                    }
                break;
```

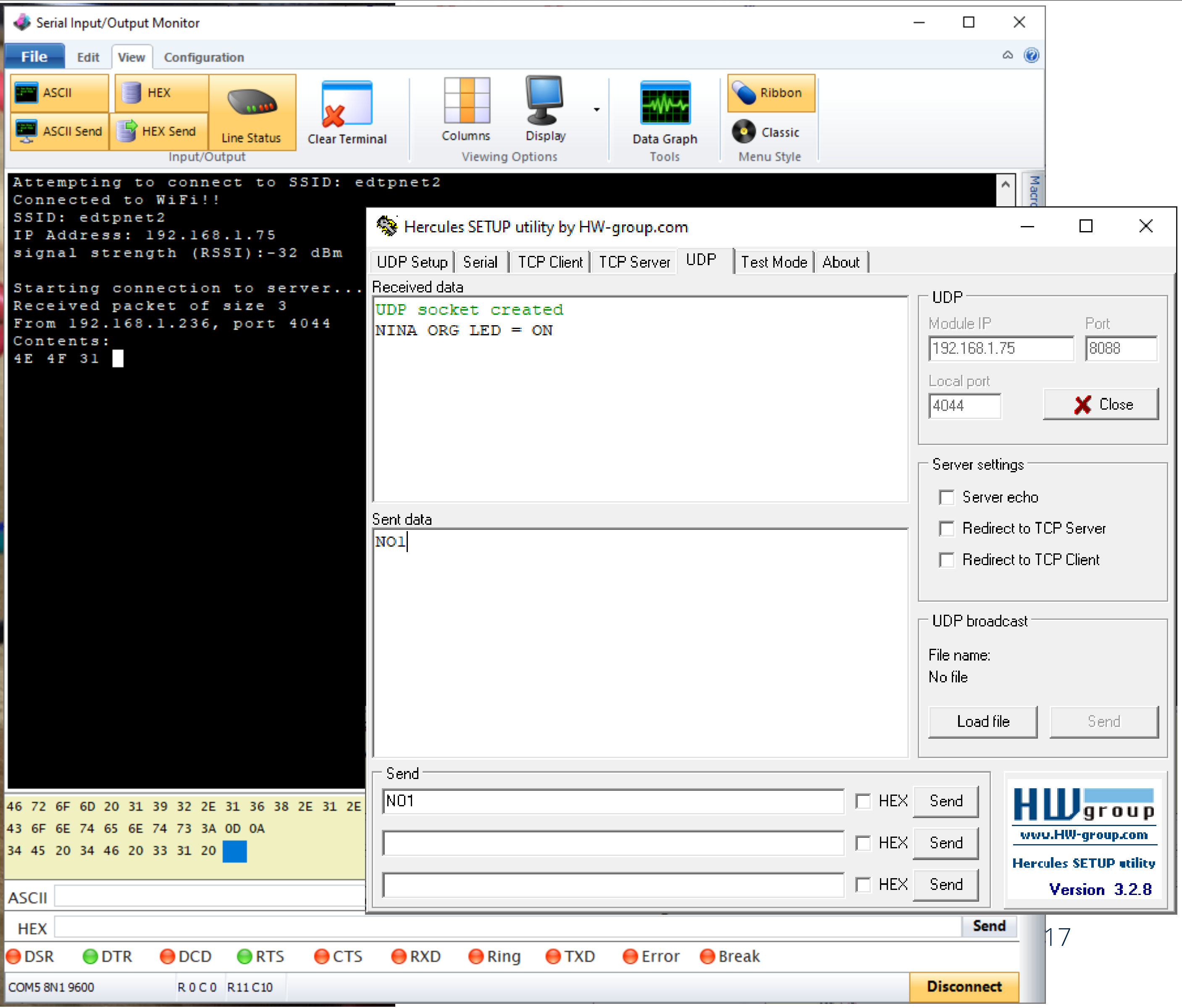
Designing, Building and Coding Custom Raspberry Pi RP2040 Arduino Devices

The RP2040, Arduino and Wi-Fi

Wi-FI Remote Control

Execute the Command and Turn on the ORG LED





```
Attempting to connect to SSID: edtpnet2
Connected to WiFi!!
SSID: edtpnet2
IP Address: 192.168.1.75
signal strength (RSSI):-32 dBm

Starting connection to server...
Received packet of size 3
From 192.168.1.236, port 4044
Contents:
4E 4F 31
```

```
46 72 6F 6D 20 31 39 32 2E 31 36 38 2E 31 2E
43 6F 6E 74 65 6E 74 73 3A 0D 0A
34 45 20 34 46 20 33 31 20
```

Received data

```
UDP socket created
NINA ORG LED = ON
```

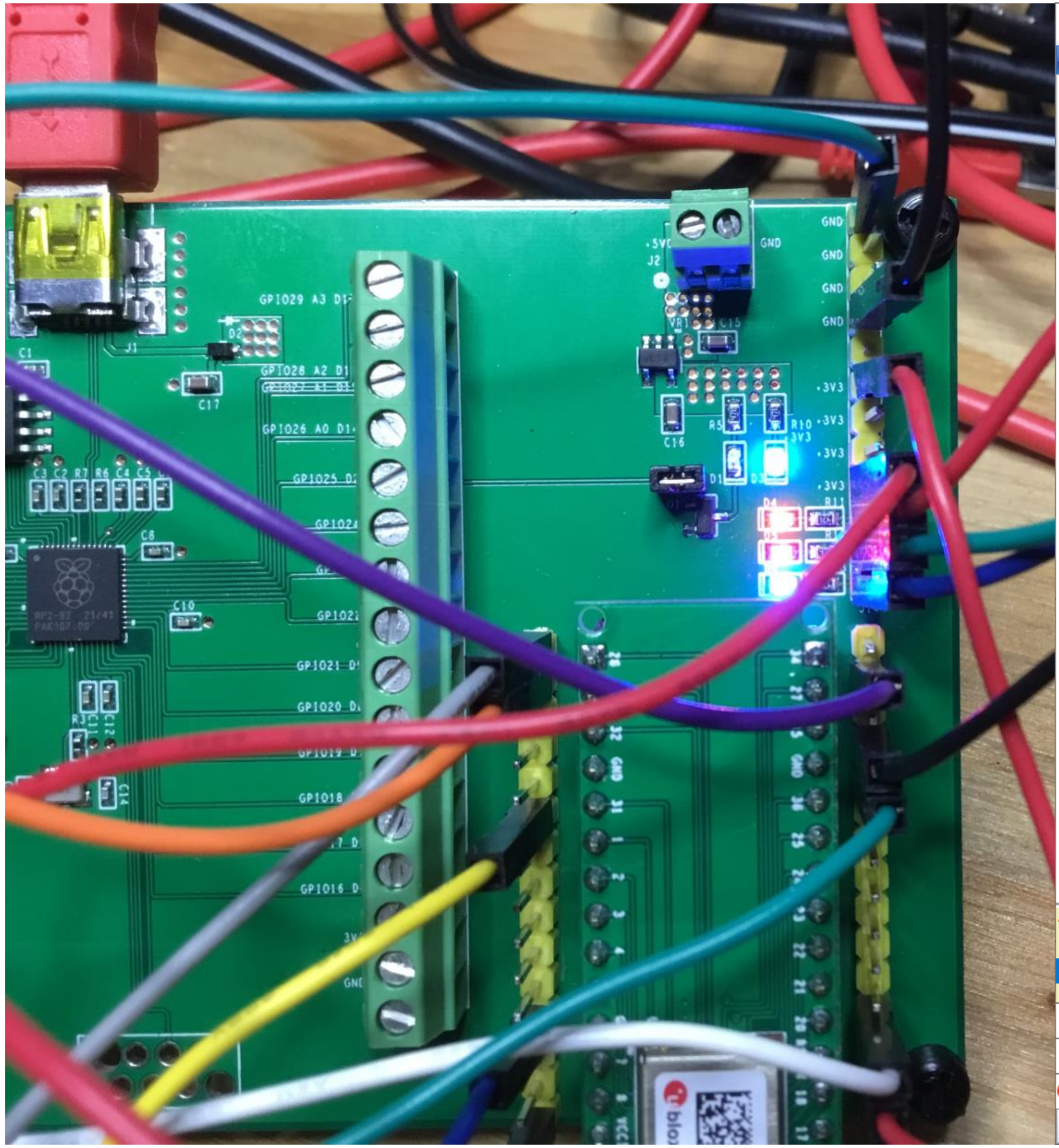
Sent data

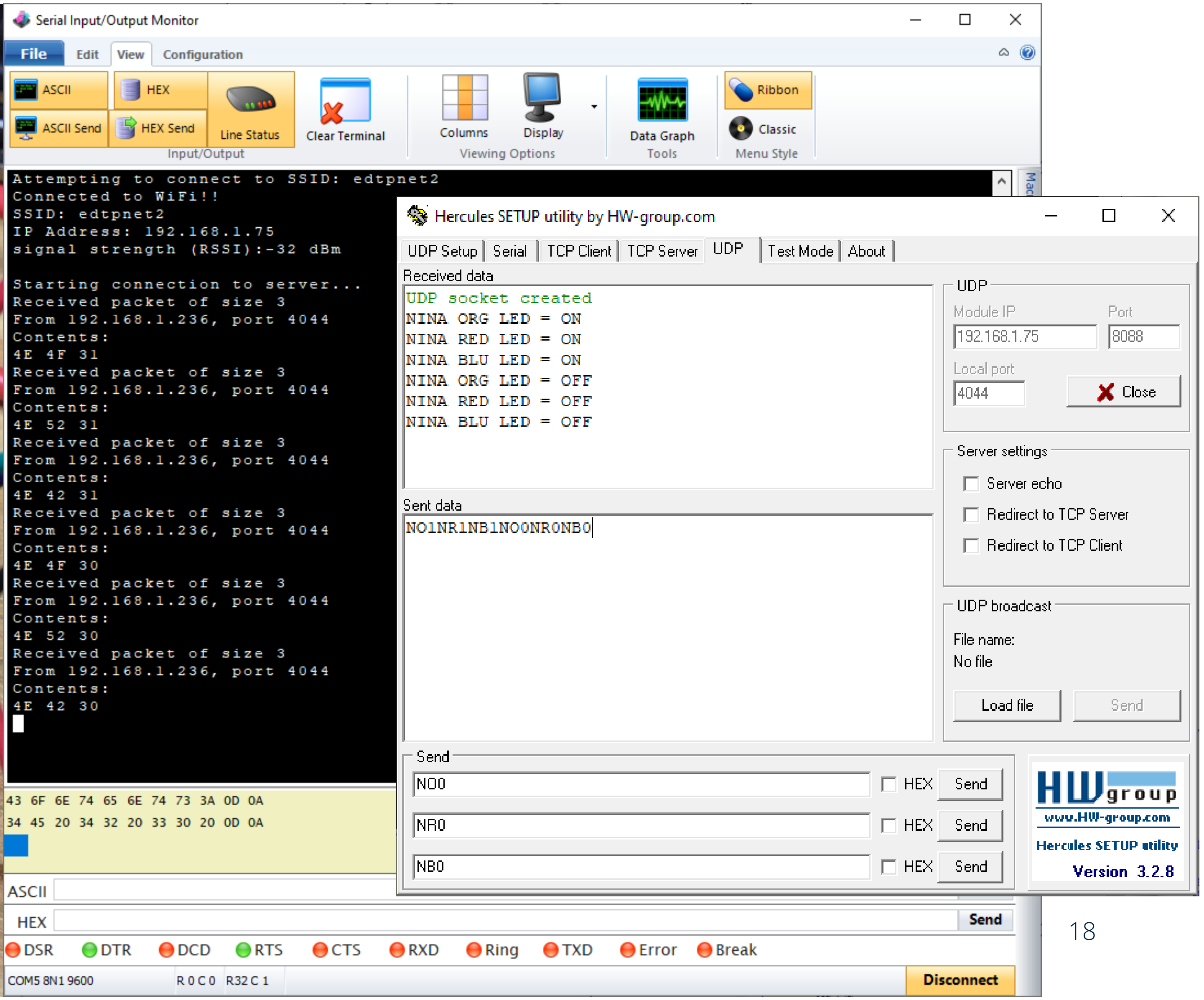
```
NO1
```

UDP
Module IP: 192.168.1.75
Port: 8088
Local port: 4044

Send: NO1

Execute the Commands to Turn on All NINA LEDs





```
Attempting to connect to SSID: edtpnet2
Connected to WiFi!!
SSID: edtpnet2
IP Address: 192.168.1.75
signal strength (RSSI):-32 dBm

Starting connection to server...
Received packet of size 3
From 192.168.1.236, port 4044
Contents:
4E 4F 31
Received packet of size 3
From 192.168.1.236, port 4044
Contents:
4E 52 31
Received packet of size 3
From 192.168.1.236, port 4044
Contents:
4E 42 31
Received packet of size 3
From 192.168.1.236, port 4044
Contents:
4E 4F 30
Received packet of size 3
From 192.168.1.236, port 4044
Contents:
4E 52 30
Received packet of size 3
From 192.168.1.236, port 4044
Contents:
4E 42 30
```

```
43 6F 6E 74 65 6E 74 73 3A 0D 0A
34 45 20 34 32 20 33 30 20 0D 0A
```

COM5 8N1 9600 R 0 C 0 R32 C 1

Received data

```
UDP socket created
NINA ORG LED = ON
NINA RED LED = ON
NINA BLU LED = ON
NINA ORG LED = OFF
NINA RED LED = OFF
NINA BLU LED = OFF
```

Sent data

```
NO1NR1NB1NO0NR0NB0
```

UDP
Module IP: 192.168.1.75
Port: 8088
Local port: 4044
Close

Server settings
- [ ] Server echo
- [ ] Redirect to TCP Server
- [ ] Redirect to TCP Client

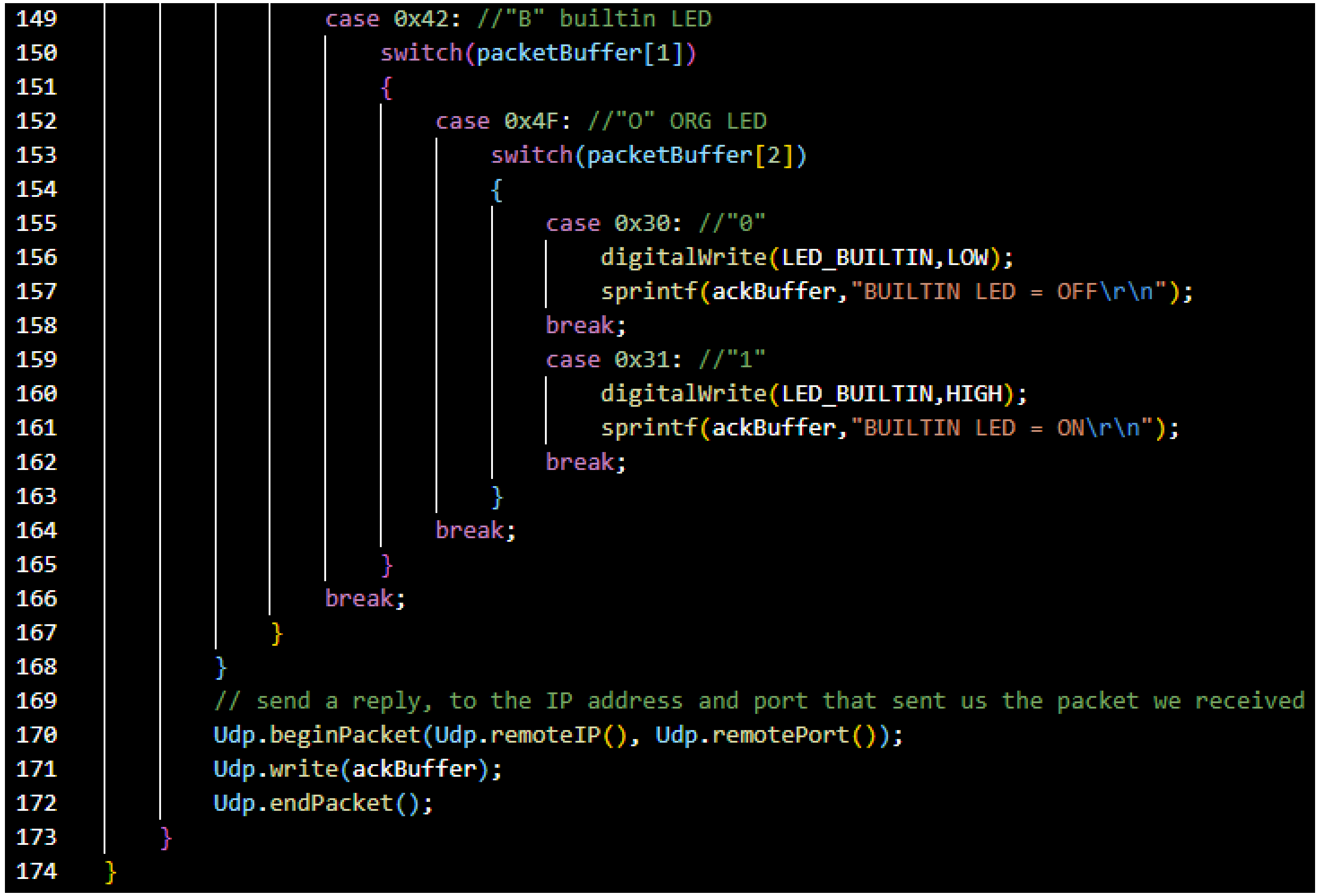
UDP broadcast
File name:
No file
Load file
Send

Send

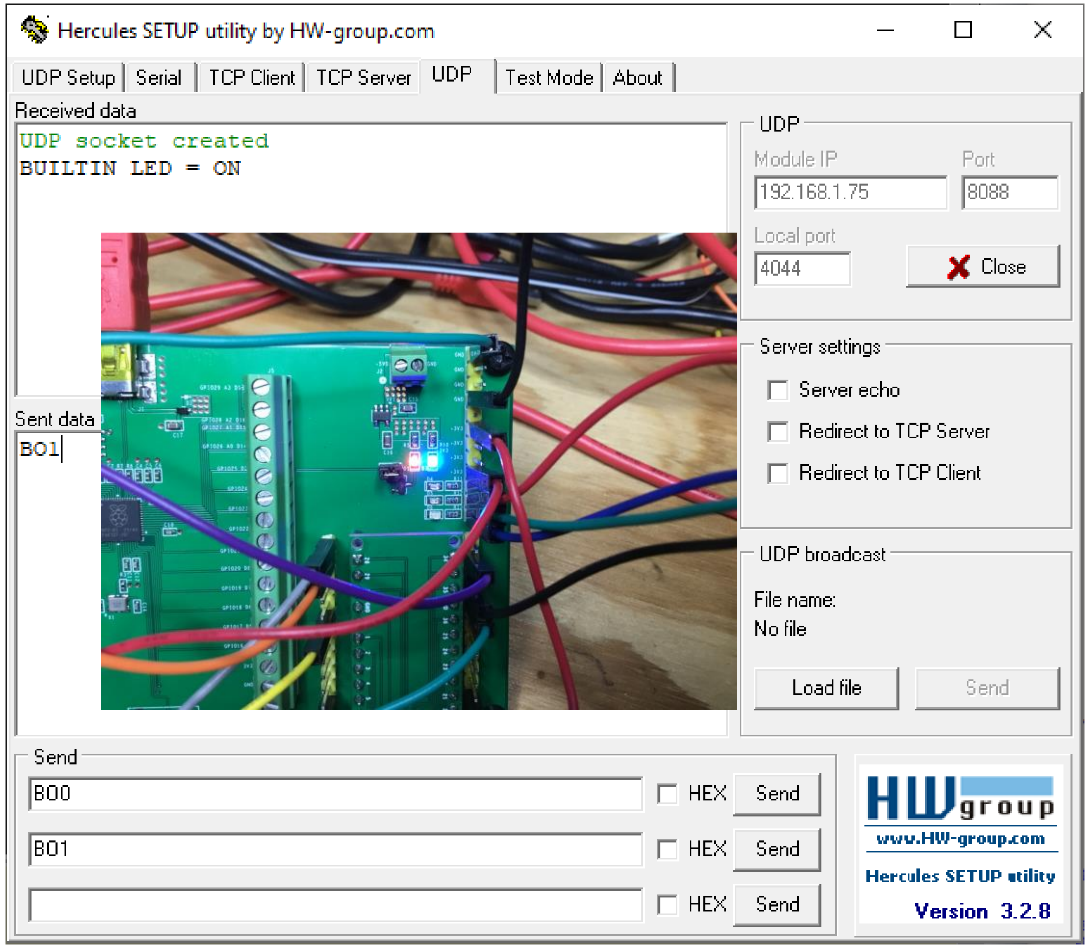
NO0	☐ HEX	Send
NR0	☐ HEX	Send
NB0	☐ HEX	Send

HW group
www.HW-group.com
Hercules SETUP utility
Version 3.2.8

Control the "BUILTIN" LED



```
                case 0x42: //"B" builtin LED
                    switch(packetBuffer[1])
                    {
                        case 0x4F: //"O" ORG LED
                            switch(packetBuffer[2])
                            {
                                case 0x30: //"0"
                                    digitalWrite(LED_BUILTIN,LOW);
                                    sprintf(ackBuffer,"BUILTIN LED = OFF\r\n");
                                break;
                                case 0x31: //"1"
                                    digitalWrite(LED_BUILTIN,HIGH);
                                    sprintf(ackBuffer,"BUILTIN LED = ON\r\n");
                                break;
                            }
                        break;
                    }
                break;
            }
        }
        // send a reply, to the IP address and port that sent us the packet we received
        Udp.beginPacket(Udp.remoteIP(), Udp.remotePort());
        Udp.write(ackBuffer);
        Udp.endPacket();
    }
}
```



Hercules SETUP utility by HW-group.com

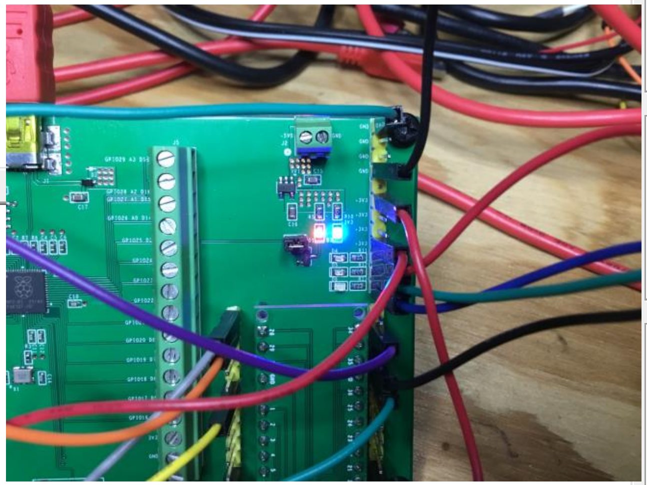
UDP Setup | Serial | TCP Client | TCP Server | UDP | Test Mode | About

Received data

```
UDP socket created
BUILTIN LED = ON
```

Sent data

```
B01
```



UDP
Module IP: 192.168.1.75
Port: 8088
Local port: 4044
Close

Server settings
☐ Server echo
☐ Redirect to TCP Server
☐ Redirect to TCP Client

UDP broadcast
File name:
No file
Load file | Send

Send
B00 ☐ HEX Send
B01 ☐ HEX Send
☐ HEX Send

HWgroup
www.HW-group.com
Hercules SETUP utility
Version 3.2.8

CEC Continuing Education Center

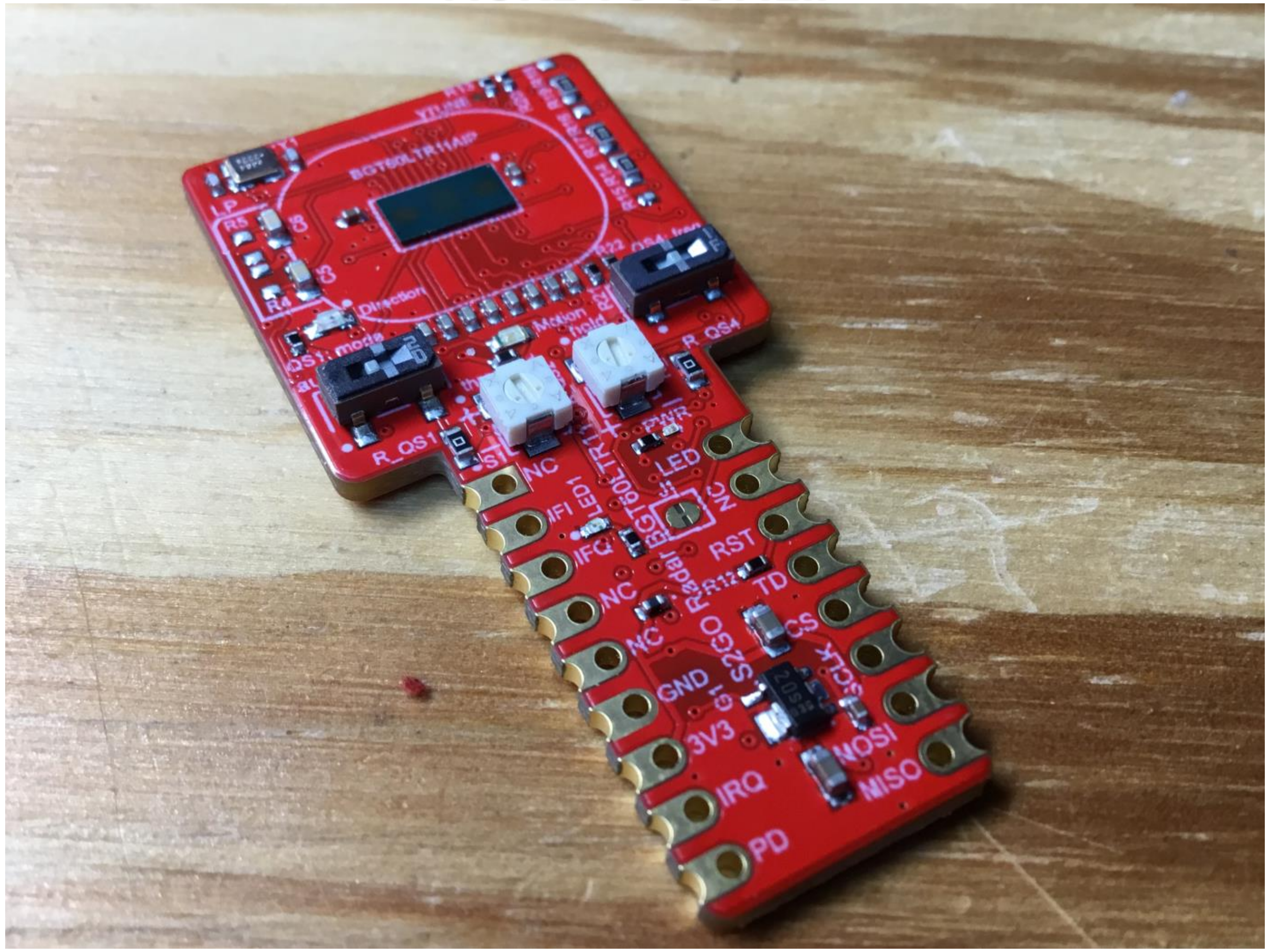
DigiKey

Thank you for attending!!!

Please consider the resources below:

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

MORE TO COME..



CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

DigiKey

informa markets