CEC Continuing Education Center

DesignNews

Designing, Building and Coding Custom Raspberry Pi RP2040 Arduino Devices

Day 1:

RP2040 Arduino Basics

Sponsored by

DigiKey

informa markets

Webinar Logistics

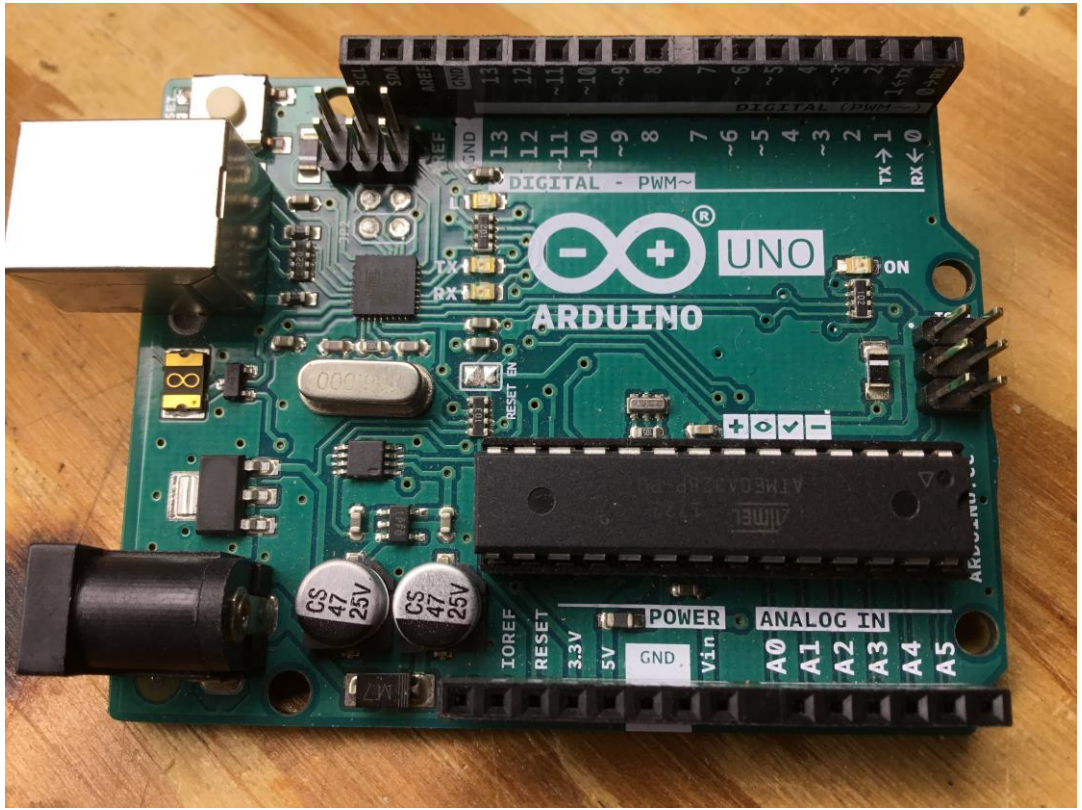
- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.

Fred Eady

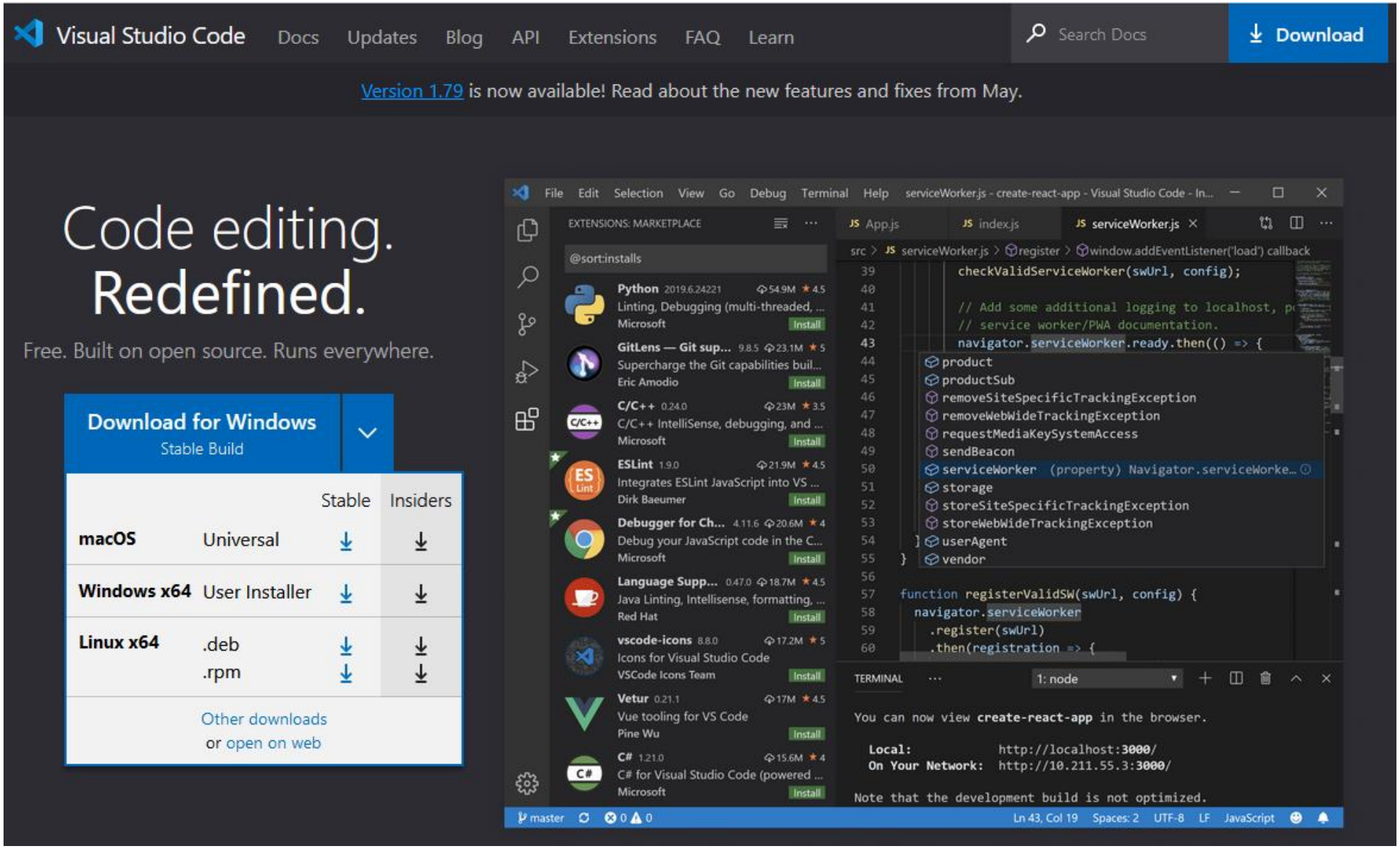
Visit ‘Lecturer Profile’ in your console for more details.

AGENDA

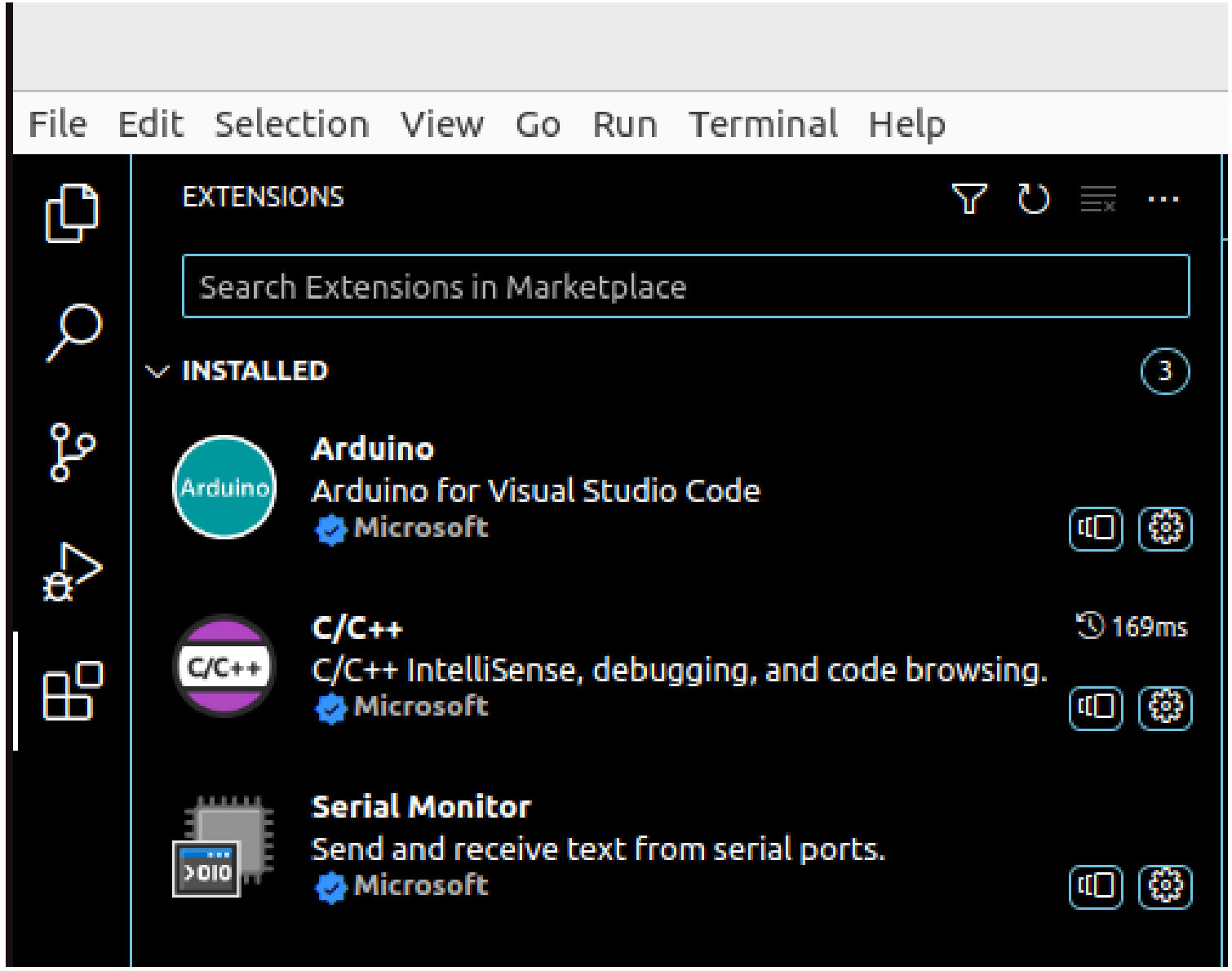
- RP2040 Arduino Toolchain Assembly
- RP2040 Arduino Target Hardware
- Sketch: wizFi360_connect.ino
- Repaired WizFi360 Arduino Library Sketches



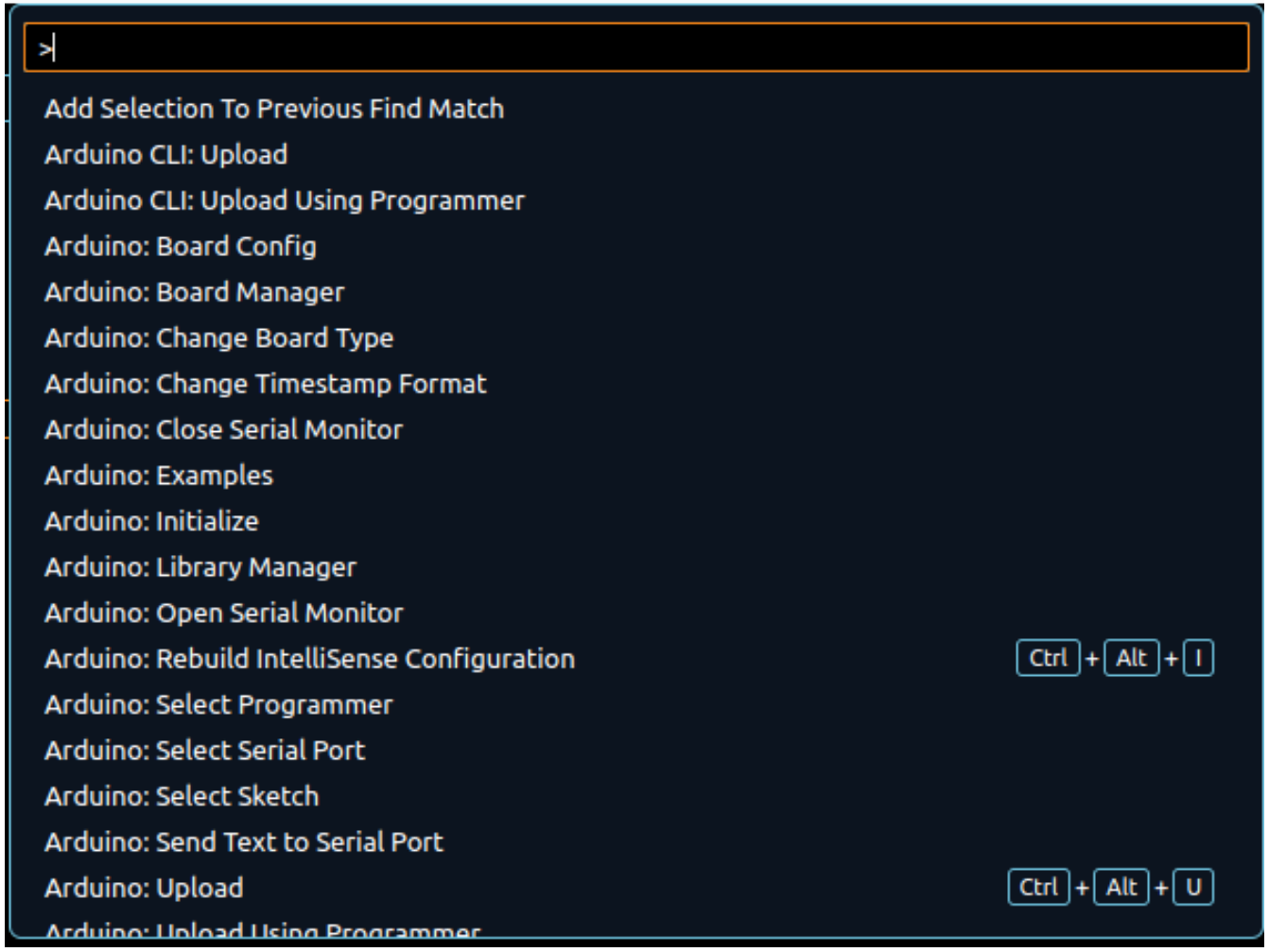
Visual Studio Code

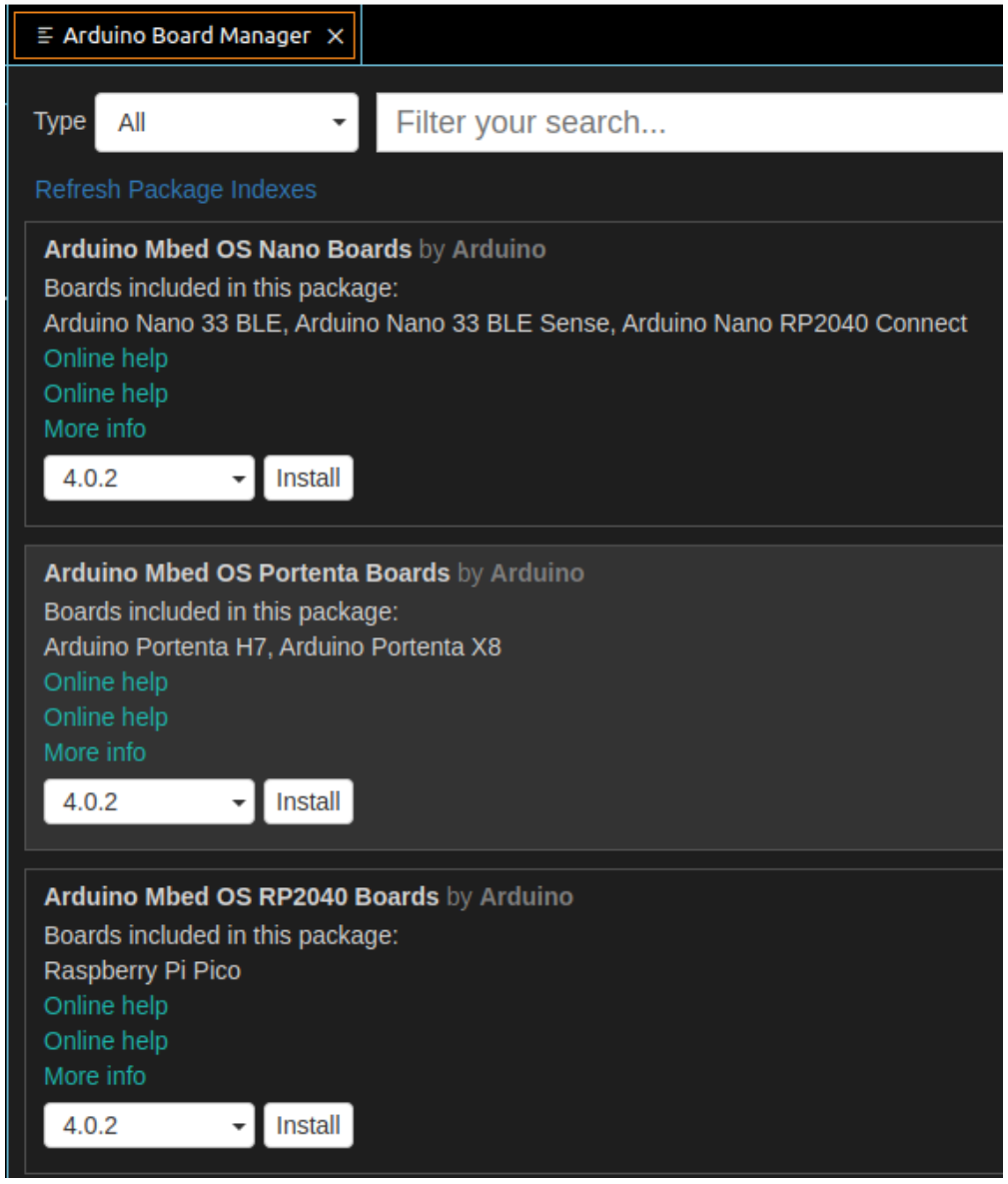


Arduino Extension for Visual Studio Code



Install RP2040-Based Arduino Boards





Install the WizFi360 Arduino Library

Getting Started with WizFi360 Arduino Library & Examples

This document will guide through a series of steps from configuring development environment to running Arduino library and examples using the WIZnet's Wi-Fi product - WizFi360.

- Development environment configuration
- Hardware requirements
- Examples
- Example testing
  - Using WizFi360-EVB-Shield & Arduino Mega 2560
  - Using WizFi360-EVB-Pico

Development environment configuration

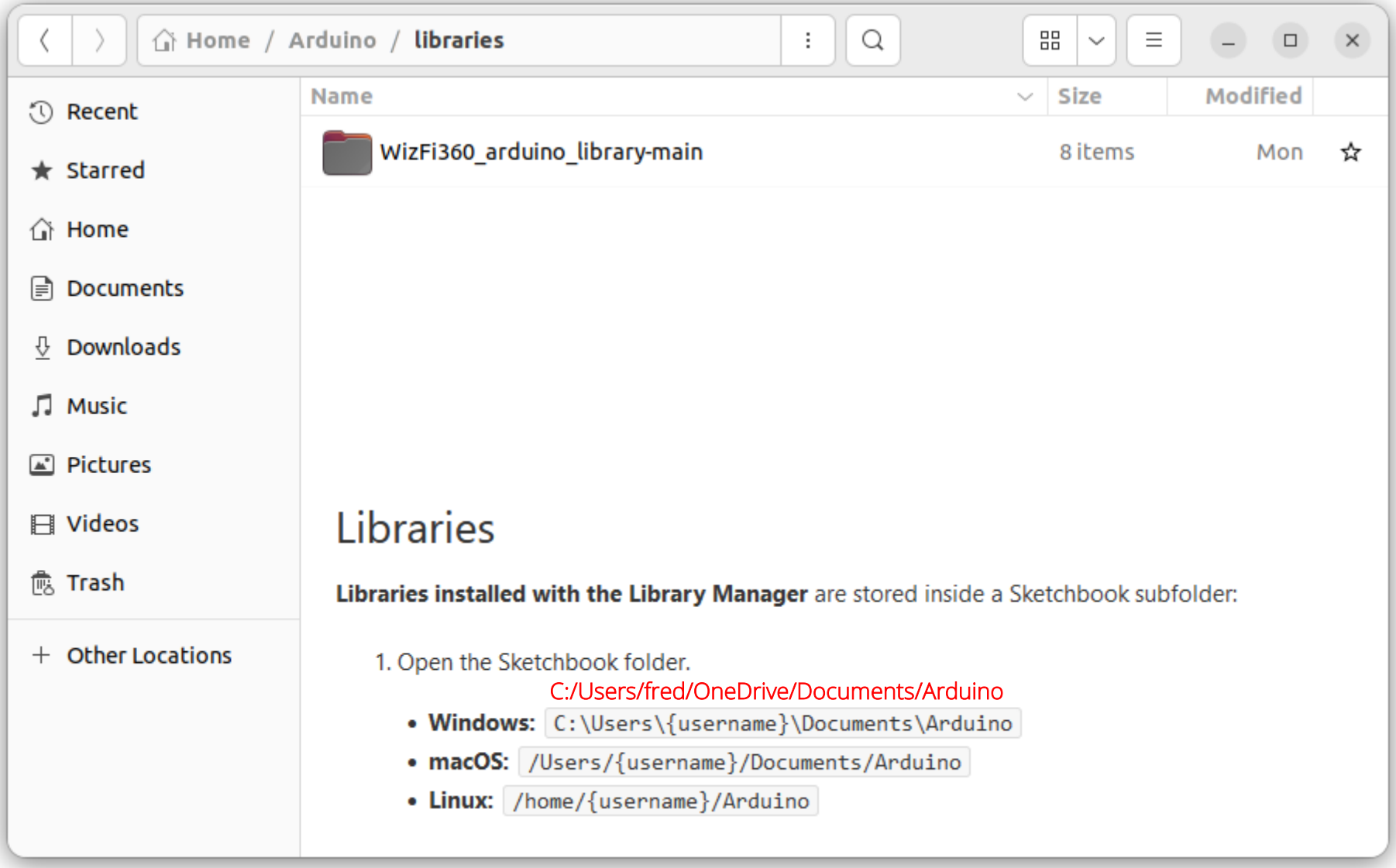
To test the WizFi360 Arduino library & examples, the development environment must be configured to use Arduino platform.

Please note that in this guide examples were configured and tested using **Arduino IDE** in **Windows** environment.

Before starting, download and install the Arduino IDE from the link below.

- Arduino IDE

Install the WizFi360 Arduino Library - Ubuntu

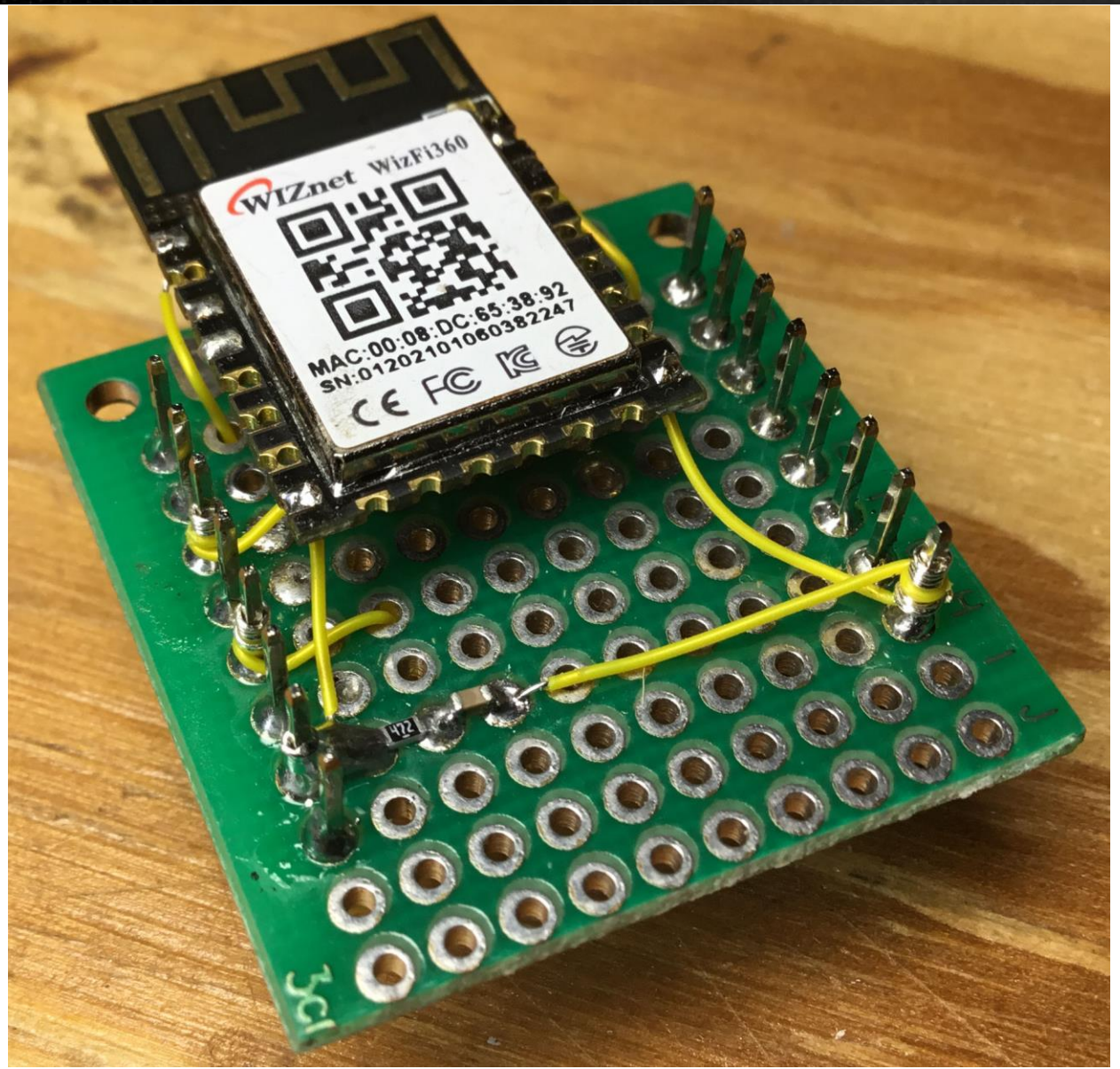


Install J-Link

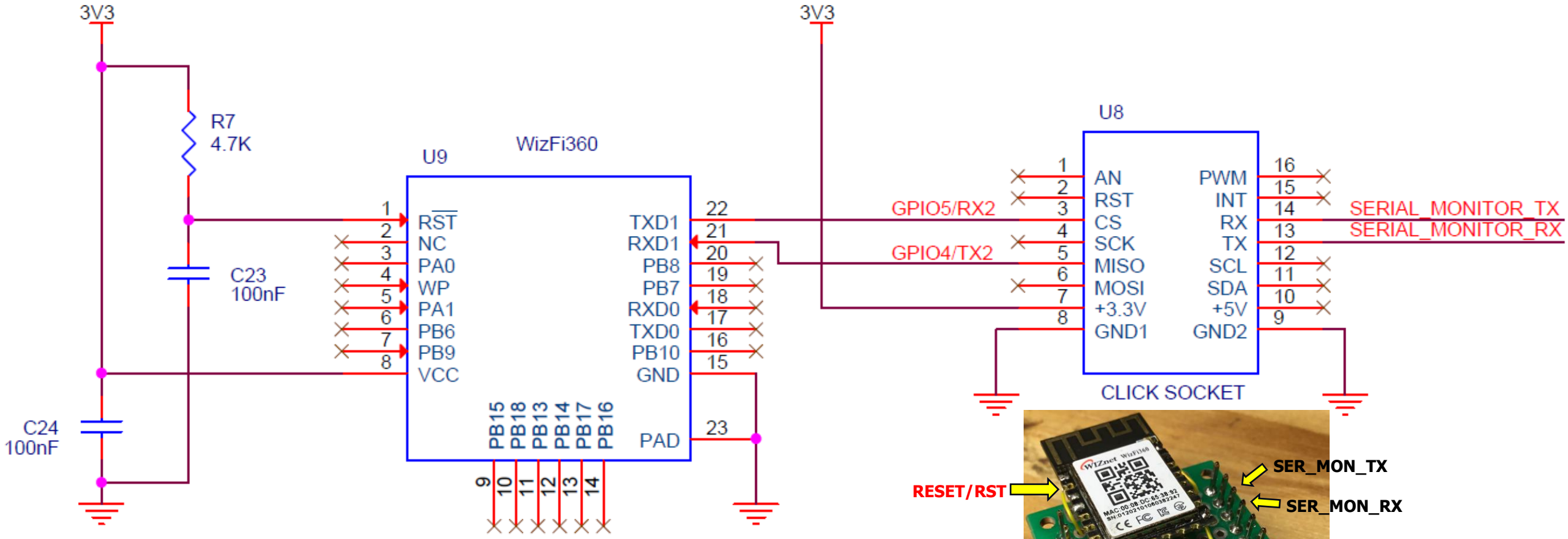
J-Link Software and Documentation Pack

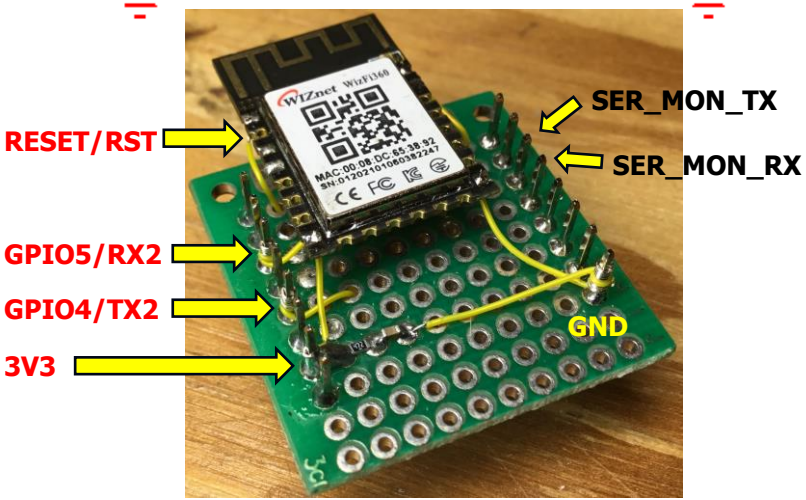
	Version	
J-Link Software and Documentation pack • All-in-one debugging solution • Can be downloaded and used free of charge by any owner of a SEGGER J-Link, J-Trace or Flasher model. Not all features of it may be available on all J-Link / J-Trace / Flasher models. • Updated frequently • Release Notes • More information	V7.88g [2023-06-13]	**Windows** 64-bit Installer 32-bit Installer **Windows ARM** 64-bit Installer **Linux** 64-bit DEB Installer 32-bit DEB Installer 64-bit RPM Installer 32-bit RPM Installer 64-bit TGZ Archive 32-bit TGZ Archive **Linux ARM** 64-bit DEB Installer 32-bit DEB Installer 64-bit TGZ Archive 32-bit TGZ Archive **macOS** 64-bit Installer 64-bit Apple M1 Installer Universal Installer

WizFi360 Pseudo *click*



WizFi360 Pseudo *click*





WizFi360 Pseudo *click* – GPIO Pin Assignments

Board platforms and cores

Board platforms installed with the Board Manager are stored inside the Arduino15 folder.

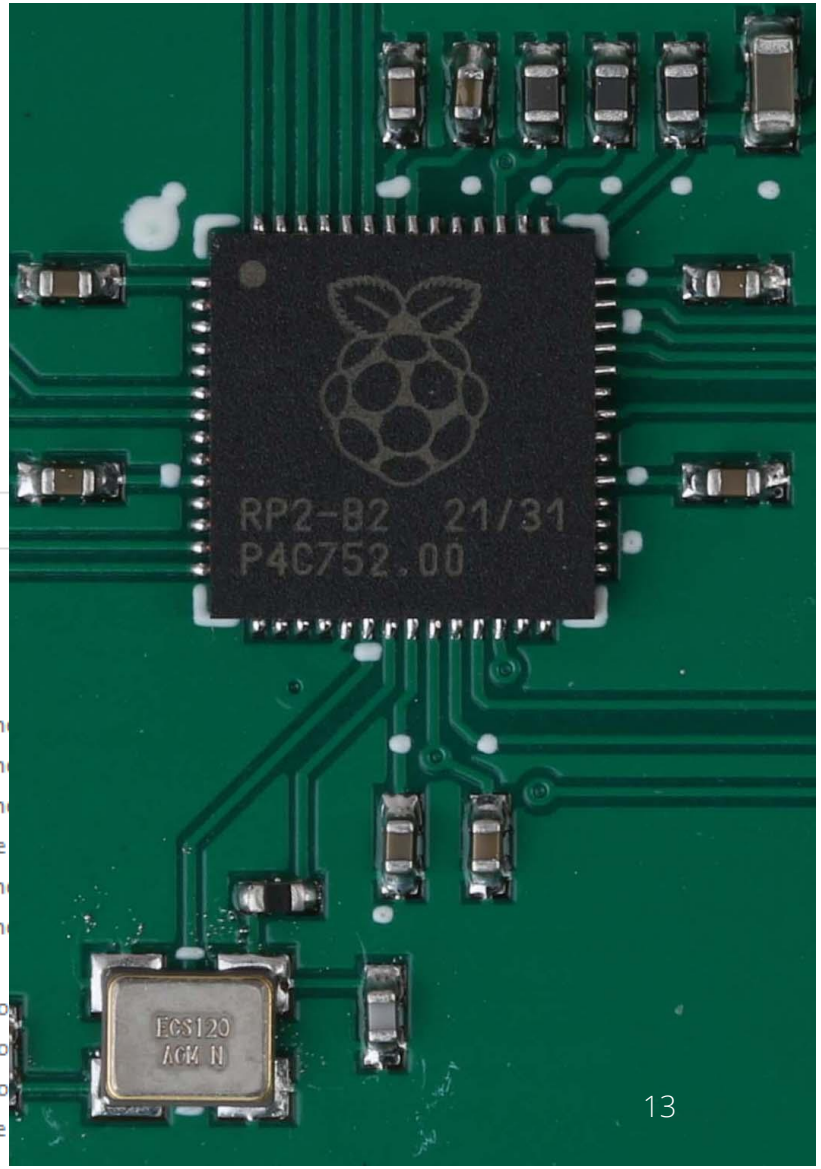
1. Open the Arduino15 folder.

- **Windows:** `C:\Users\{username}\AppData\Local\Arduino15`
- **macOS:** `/Users/{username}/Library/Arduino15`
- **Linux:** `home/{username}/.arduino15`

```
Arduino15
└── packages
    └── arduino
        └── hardware
            └── avr
                └── 1.8.4
                    ├── README.md
                    ├── boards.txt
                    ├── bootloaders
                    ├── cores
                    ├── extras
                    ├── firmwares
                    ├── installed.json
                    ├── libraries
                    ├── platform.txt
                    ├── programmers.txt
                    └── variants
```

« hardware › mbed_rp2040 › 4.0.2 › variants › RASPBERRY_PI_PICO ›

	Name	Date modified	Type
avr	conf	6/1/2023 12:01 PM	File folder
mbed_nano	libs	6/1/2023 12:01 PM	File folder
mbed_rp2040	cflags.txt	6/1/2023 12:01 PM	Text Docum
samd	cxxflags.txt	6/1/2023 12:01 PM	Text Docum
	defines.txt	6/1/2023 12:01 PM	Text Docum
	double_tap_usb_boot.cpp	6/1/2023 12:01 PM	C++ Source
	includes.txt	6/1/2023 12:01 PM	Text Docum
	ldflags.txt	6/1/2023 12:01 PM	Text Docum
	linker_script.ld	6/1/2023 12:01 PM	LD File
	mbed_config.h	6/1/2023 12:01 PM	C Header So
	pinmode_arduino.h	6/1/2023 12:01 PM	C Header So
	pins_arduino.h	7/10/2023 2:24 PM	C Header So
	variant.cpp	6/27/2023 9:26 AM	C++ Source

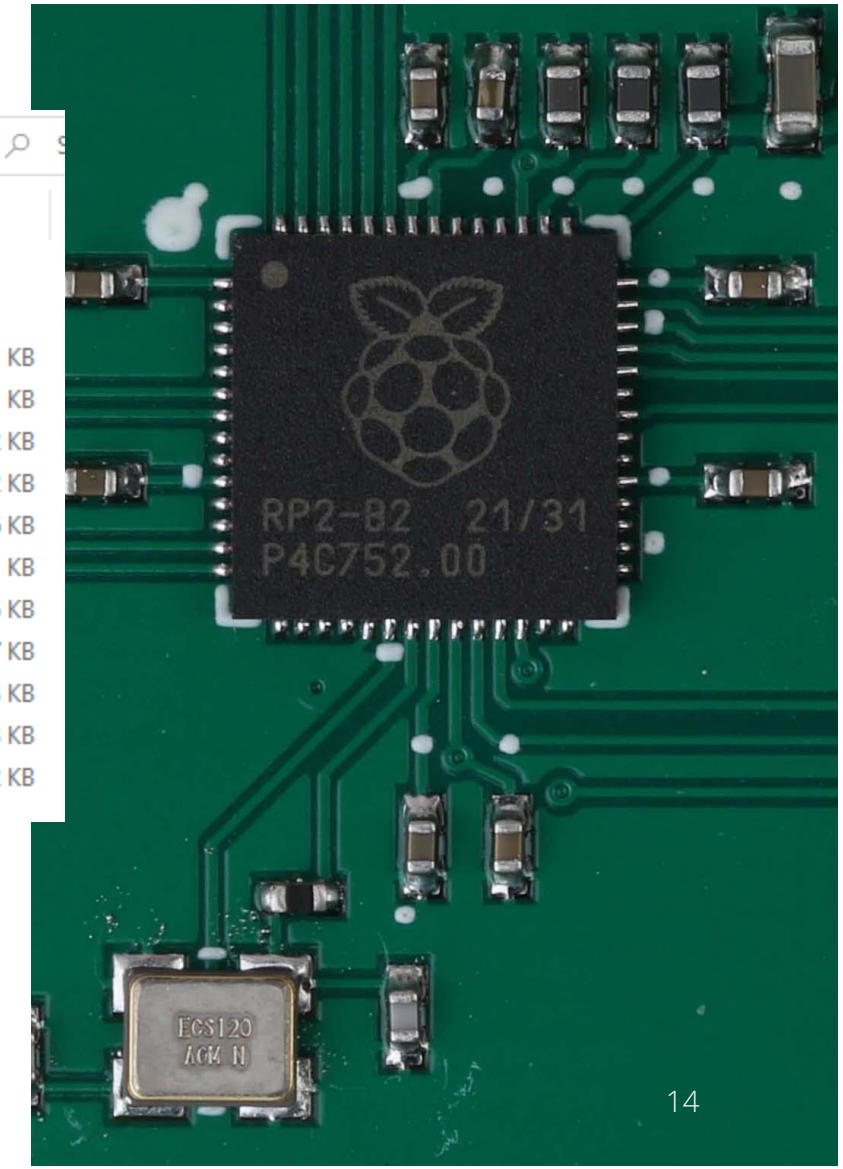


WizFi360 Pseudo *click* – GPIO Pins - variant.cpp

```
PinDescription g_APinDescription[] = {
  // D0 - D29
  { p0,  NULL, NULL, NULL },
  { p1,  NULL, NULL, NULL },
  { p2,  NULL, NULL, NULL },
  { p3,  NULL, NULL, NULL },
  { p4,  NULL, NULL, NULL },
  { p5,  NULL, NULL, NULL },
  { p6,  NULL, NULL, NULL },
  { p7,  NULL, NULL, NULL },
  { p8,  NULL, NULL, NULL },
  { p9,  NULL, NULL, NULL },
  { p10, NULL, NULL, NULL },
  { p11, NULL, NULL, NULL },
  { p12, NULL, NULL, NULL },
  { p13, NULL, NULL, NULL },
  { p14, NULL, NULL, NULL },
  { p15, NULL, NULL, NULL },
  { p16, NULL, NULL, NULL },
  { p17, NULL, NULL, NULL },
  { p18, NULL, NULL, NULL },
  { p19, NULL, NULL, NULL },
  { p20, NULL, NULL, NULL },
  { p21, NULL, NULL, NULL },
  { p22, NULL, NULL, NULL },
  { p23, NULL, NULL, NULL },
  { p24, NULL, NULL, NULL },
  { p25, NULL, NULL, NULL },
  { p26, NULL, NULL, NULL },
  { p27, NULL, NULL, NULL },
  { p28, NULL, NULL, NULL },
  { p29, NULL, NULL, NULL },
};
```

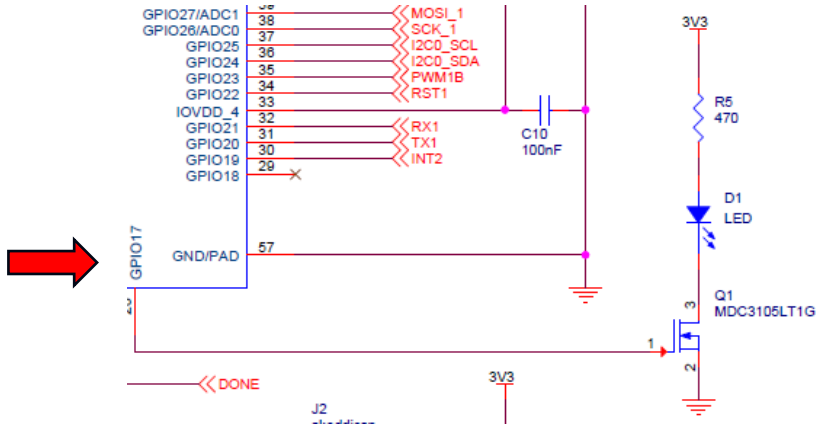
« hardware › mbed_rp2040 › 4.0.2 › variants › RASPBERRY_PI_PICO ›

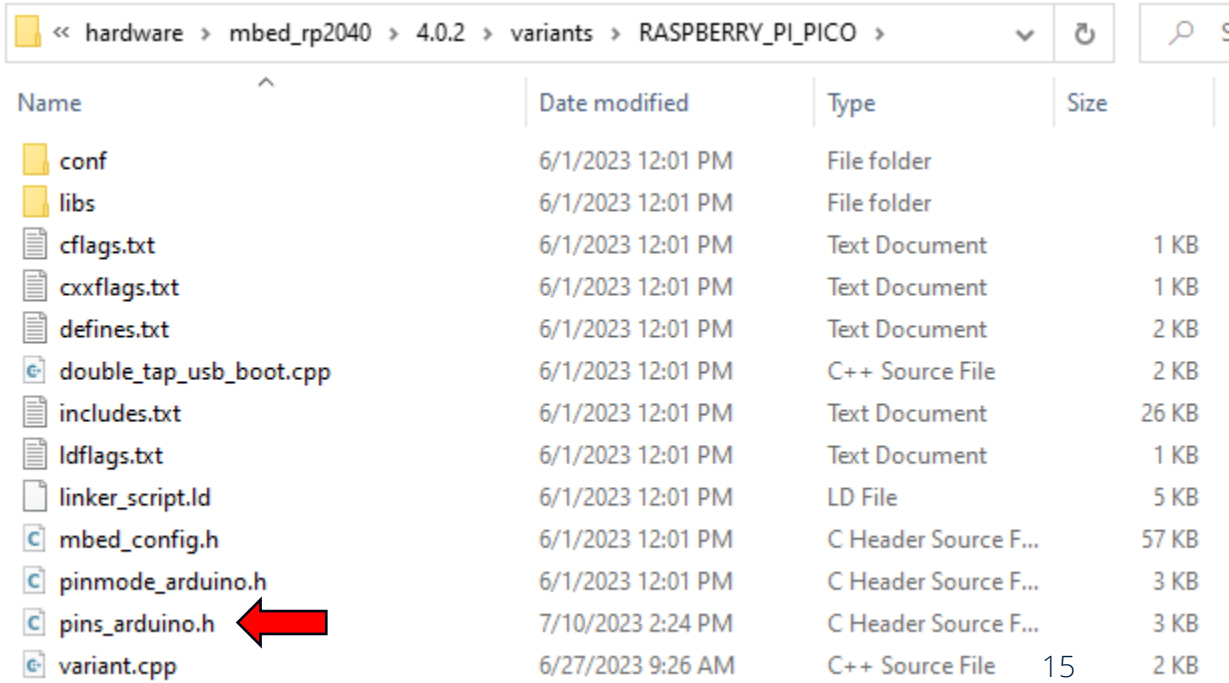
Name	Date modified	Type	Size
conf	6/1/2023 12:01 PM	File folder	
libs	6/1/2023 12:01 PM	File folder	
cflags.txt	6/1/2023 12:01 PM	Text Document	1 KB
cxxflags.txt	6/1/2023 12:01 PM	Text Document	1 KB
defines.txt	6/1/2023 12:01 PM	Text Document	2 KB
double_tap_usb_boot.cpp	6/1/2023 12:01 PM	C++ Source File	2 KB
includes.txt	6/1/2023 12:01 PM	Text Document	26 KB
ldflags.txt	6/1/2023 12:01 PM	Text Document	1 KB
linker_script.ld	6/1/2023 12:01 PM	LD File	5 KB
mbed_config.h	6/1/2023 12:01 PM	C Header Source F...	57 KB
pinmode_arduino.h	6/1/2023 12:01 PM	C Header Source F...	3 KB
pins_arduino.h	7/10/2023 2:24 PM	C Header Source F...	3 KB
variant.cpp	6/27/2023 9:26 AM	C++ Source File	2 KB



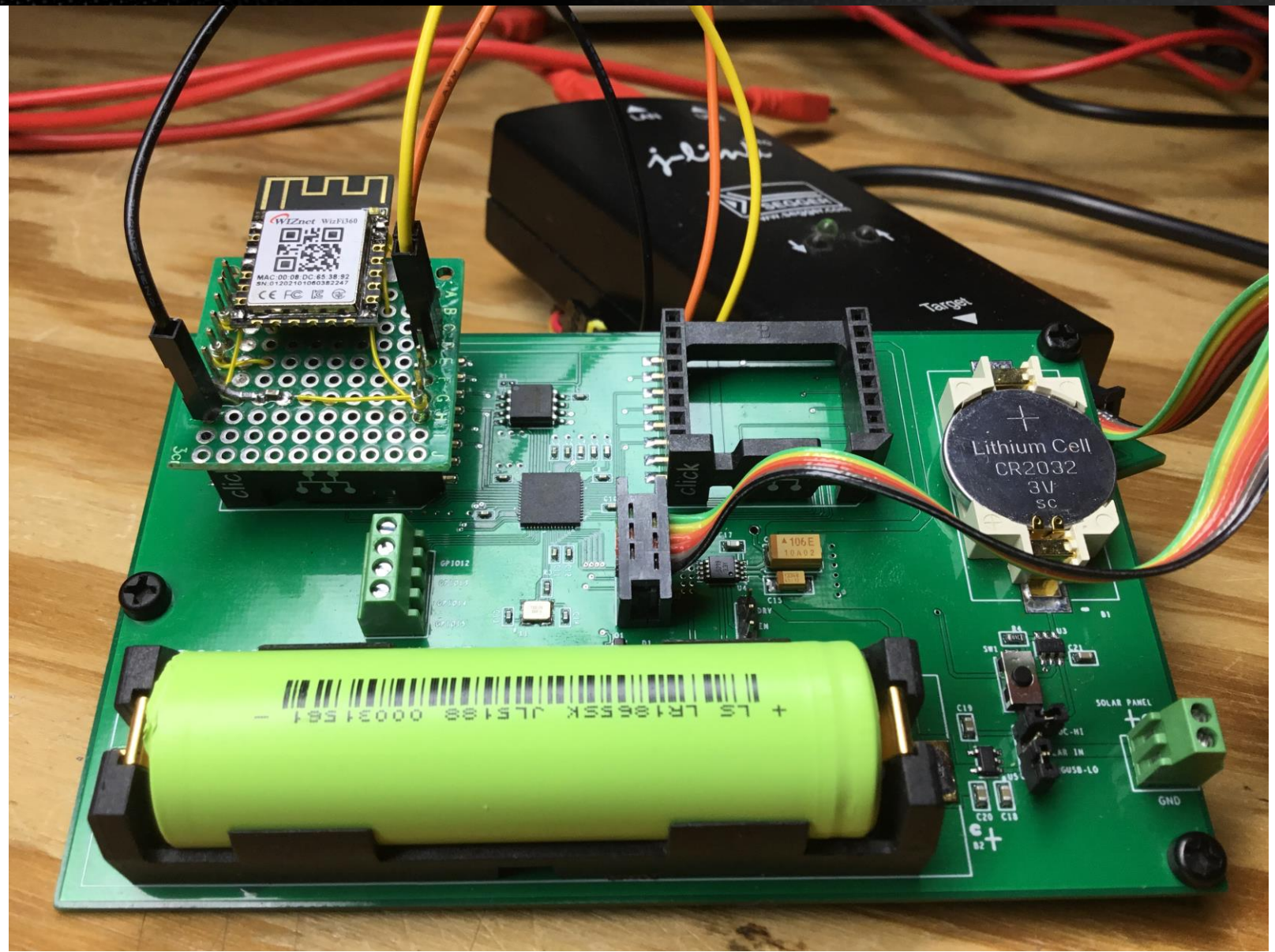
WizFi360 Pseudo *click* – GPIO Pins - pins_arduino.h

```
21  // LEDs
22  // ----
23  #define PIN_LED     (17u)
24  #define LED_BUILTIN PIN_LED
25
26  // Serial
27  #define PIN_SERIAL_TX (0ul)
28  #define PIN_SERIAL_RX (1ul)
29
30  #define SERIAL_HOWMANY      2
31  #define SERIAL1_TX          (digitalPinToPinName(PIN_SERIAL_TX))
32  #define SERIAL1_RX          (digitalPinToPinName(PIN_SERIAL_RX))
33  #define SERIAL2_TX          (digitalPinToPinName(4))
34  #define SERIAL2_RX          (digitalPinToPinName(5))
35
36  #define SERIAL_CDC          1
37  #define HAS_UNIQUE_ISERIAL_DESCRIPTOR
38  #define BOARD_VENDORID      0x2e8a
39  #define BOARD_PRODUCTID     0x00c0
40  #define BOARD_NAME          "RaspberryPi Pico"
41
42  #define SERIAL_PORT_USBVIRTUAL      SerialUSB
43  #define SERIAL_PORT_MONITOR         SerialUSB
44  #define SERIAL_PORT_HARDWARE        Serial1
45  #define SERIAL_PORT_HARDWARE_OPEN   Serial2
```

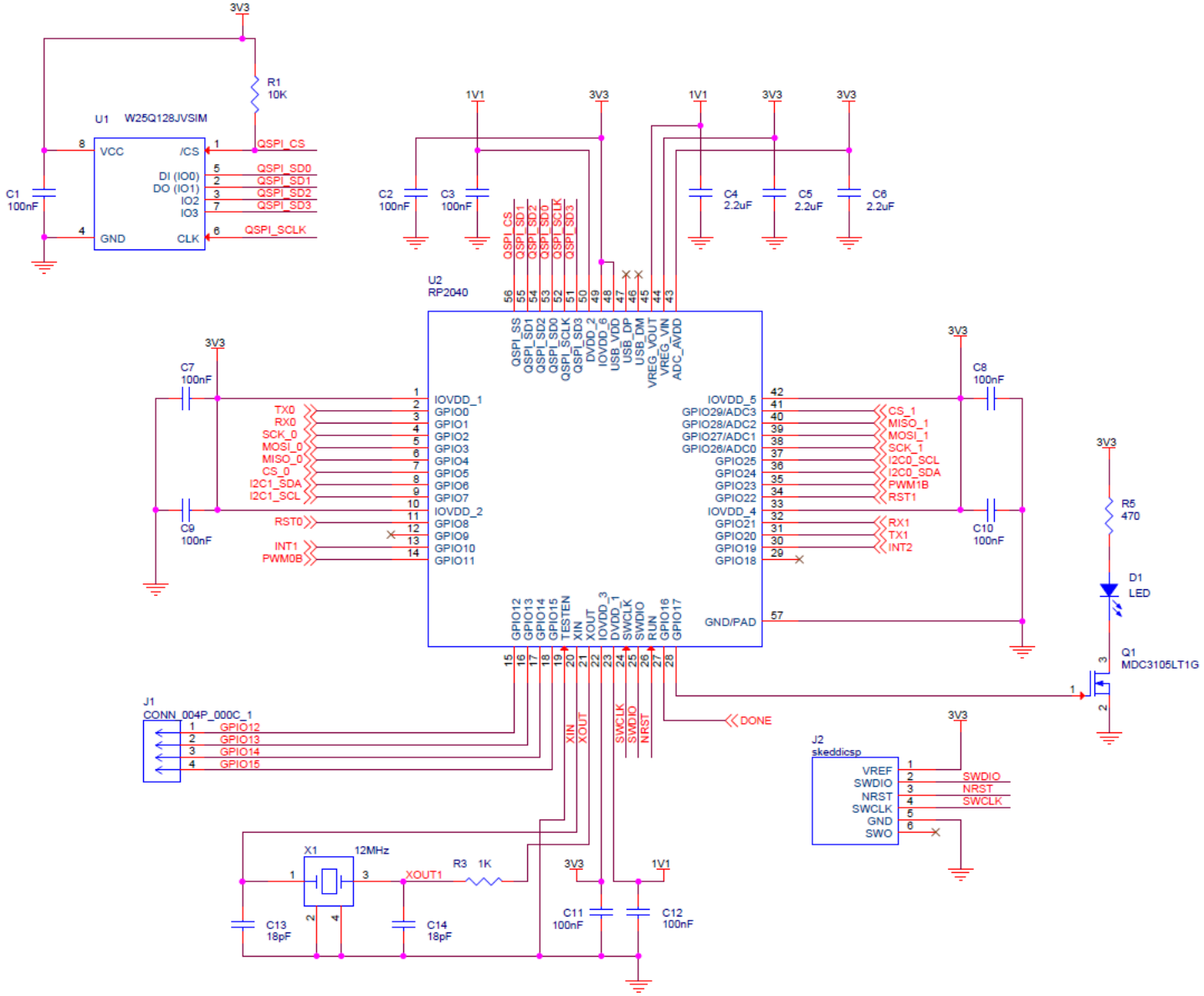




RP2040 Sensor Node



RP2040 Sensor Node



Insecticide – debug.h

```
24  // Change _WIZFILOGLEVEL_ to set tracing and logging verbosity
25  // 0: DISABLED: no logging
26  // 1: ERROR: errors
27  // 2: WARN: errors and warnings
28  // 3: INFO: errors, warnings and informational (default)
29  // 4: DEBUG: errors, warnings, informational and debug
30
31  #ifndef _WIZFILOGLEVEL_
32  #define _WIZFILOGLEVEL_ 0
33  #endif
34
35
36  #define LOGERROR(x)     if(_WIZFILOGLEVEL_>0) { Serial1.print("[WizFi360] "); Serial1.println(x); }
37  #define LOGERROR1(x,y) if(_WIZFILOGLEVEL_>2) { Serial1.print("[WizFi360] "); Serial1.print(x); Serial1.print(" "); Serial1.println(y); }
38  #define LOGWARN(x)      if(_WIZFILOGLEVEL_>1) { Serial1.print("[WizFi360] "); Serial1.println(x); }
39  #define LOGWARN1(x,y)  if(_WIZFILOGLEVEL_>2) { Serial1.print("[WizFi360] "); Serial1.print(x); Serial1.print(" "); Serial1.println(y); }
40  #define LOGINFO(x)      if(_WIZFILOGLEVEL_>2) { Serial1.print("[WizFi360] "); Serial1.println(x); }
41  #define LOGINFO1(x,y)  if(_WIZFILOGLEVEL_>2) { Serial1.print("[WizFi360] "); Serial1.print(x); Serial1.print(" "); Serial1.println(y); }
42
43  #define LOGDEBUG(x)       if(_WIZFILOGLEVEL_>3) { Serial1.println(x); }
44  #define LOGDEBUG0(x)      if(_WIZFILOGLEVEL_>3) { Serial1.print(x); }
45  #define LOGDEBUG1(x,y)    if(_WIZFILOGLEVEL_>3) { Serial1.print(x); Serial1.print(" "); Serial1.println(y); }
46  #define LOGDEBUG2(x,y,z) if(_WIZFILOGLEVEL_>3) { Serial1.print(x); Serial1.print(" "); Serial1.print(y); Serial1.print(" "); Serial1.println(z); }
47
48
49  #endif
```

RingBuffer Bug

```
utility/WizFi360Drv.h:315:9: error: reference to 'RingBuffer' is ambiguous
```

PROBLEM:

```
314     // the ring buffer is used to search the tags in the stream
315     static RingBuffer ringBuf;
```

FIX:

```
314     // the ring buffer is used to search the tags in the stream
315     static Ring_Buffer ringBuf;
```

Renamed RingBuffer.h to Ring_Buffer.h
Renamed RingBuffer class in Ring_Buffer.h to Ring_Buffer
Changed all references to RingBuffer to Ring_Buffer in the sketch files

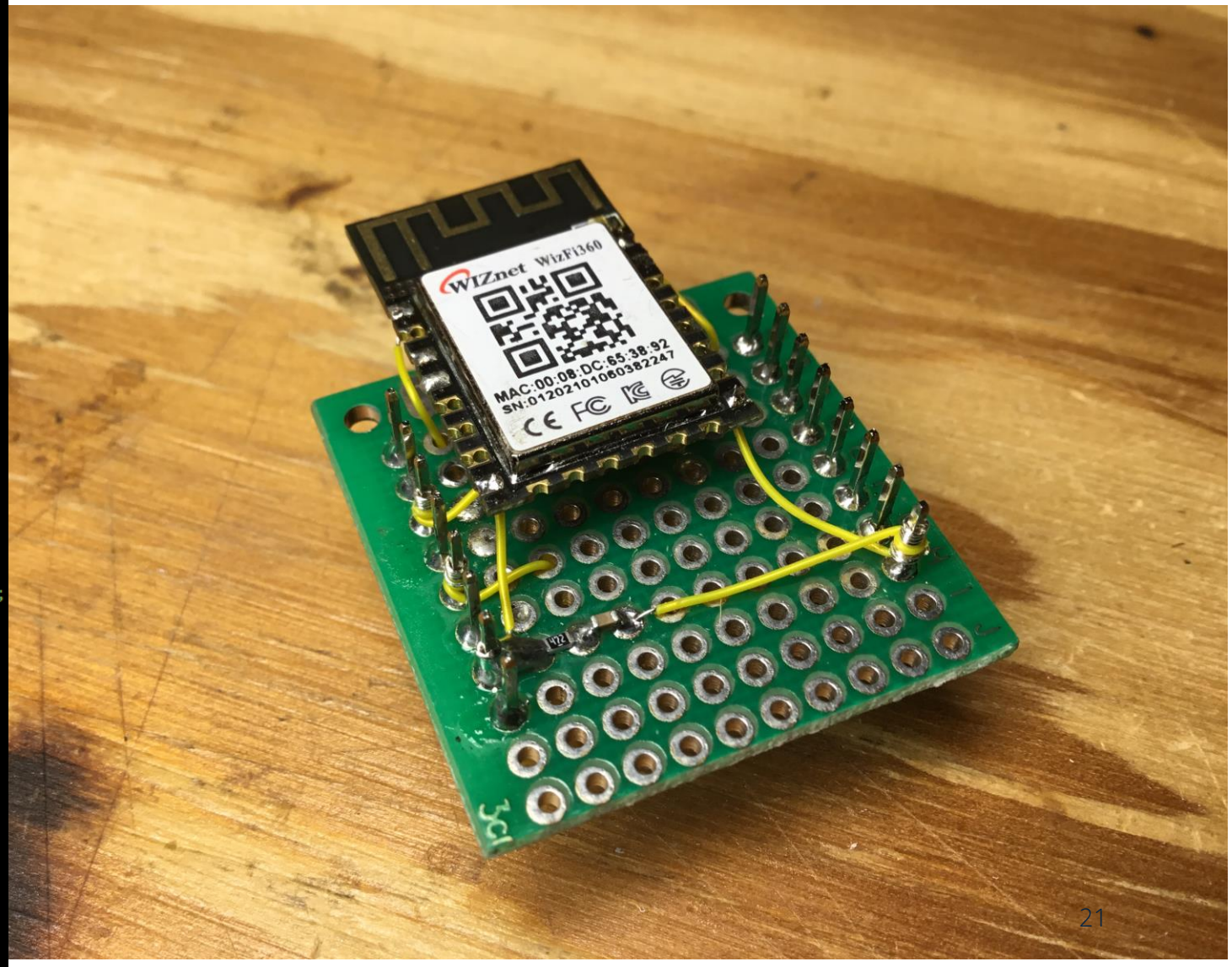
snprintf_P *"PROBLEM"*

```
1095  /*
1096  * Sends the AT command and returns the id of the TAG.
1097  * The additional arguments are formatted into the command using sprintf.
1098  * Return -1 if no tag is found.
1099  */
1100  int WizFi360Drv::sendCmd(const __FlashStringHelper* cmd, int timeout, ...)
1101  {
1102      char cmdBuf[CMD_BUFFER_SIZE];
1103
1104      va_list args;
1105      va_start (args, timeout);
1106      snprintf_P (cmdBuf, CMD_BUFFER_SIZE, (char*)cmd, args);
1107      va_end (args);
1108
1109      wizfi360EmptyBuf();
1110
1111      LOGDEBUG(F("-------------------------------------------"));
1112      LOGDEBUG1(F(">>"), cmdBuf);
1113
1114      wizfi360Serial->println(cmdBuf);
1115
1116      int idx = readUntil(timeout);
1117
1118      LOGDEBUG1(F("------------------------------------------- >"), idx);
1119      LOGDEBUG();
1120
1121      return idx;
1122  }
```

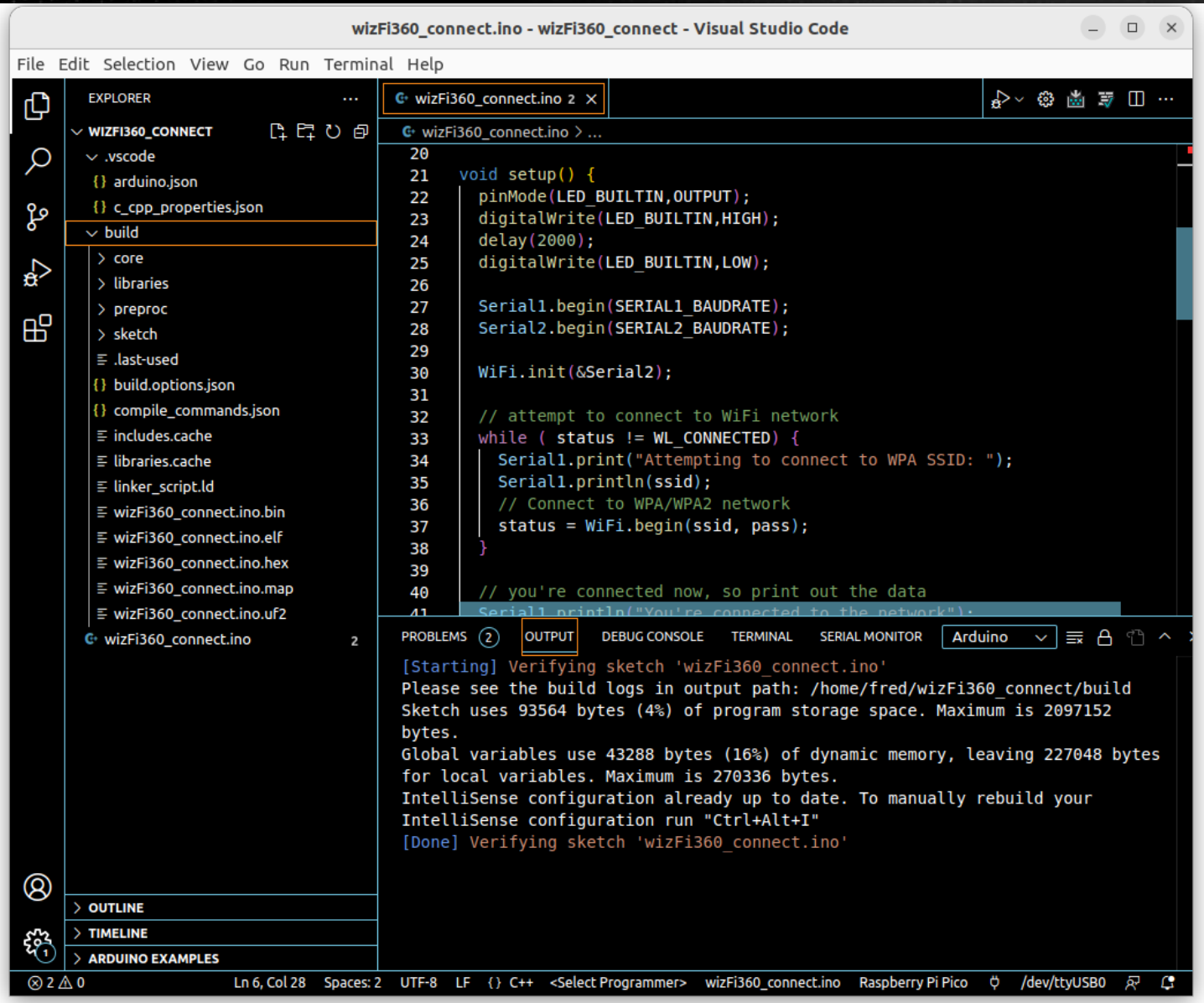
PROBLEM

snprintf_P *"FIX"*

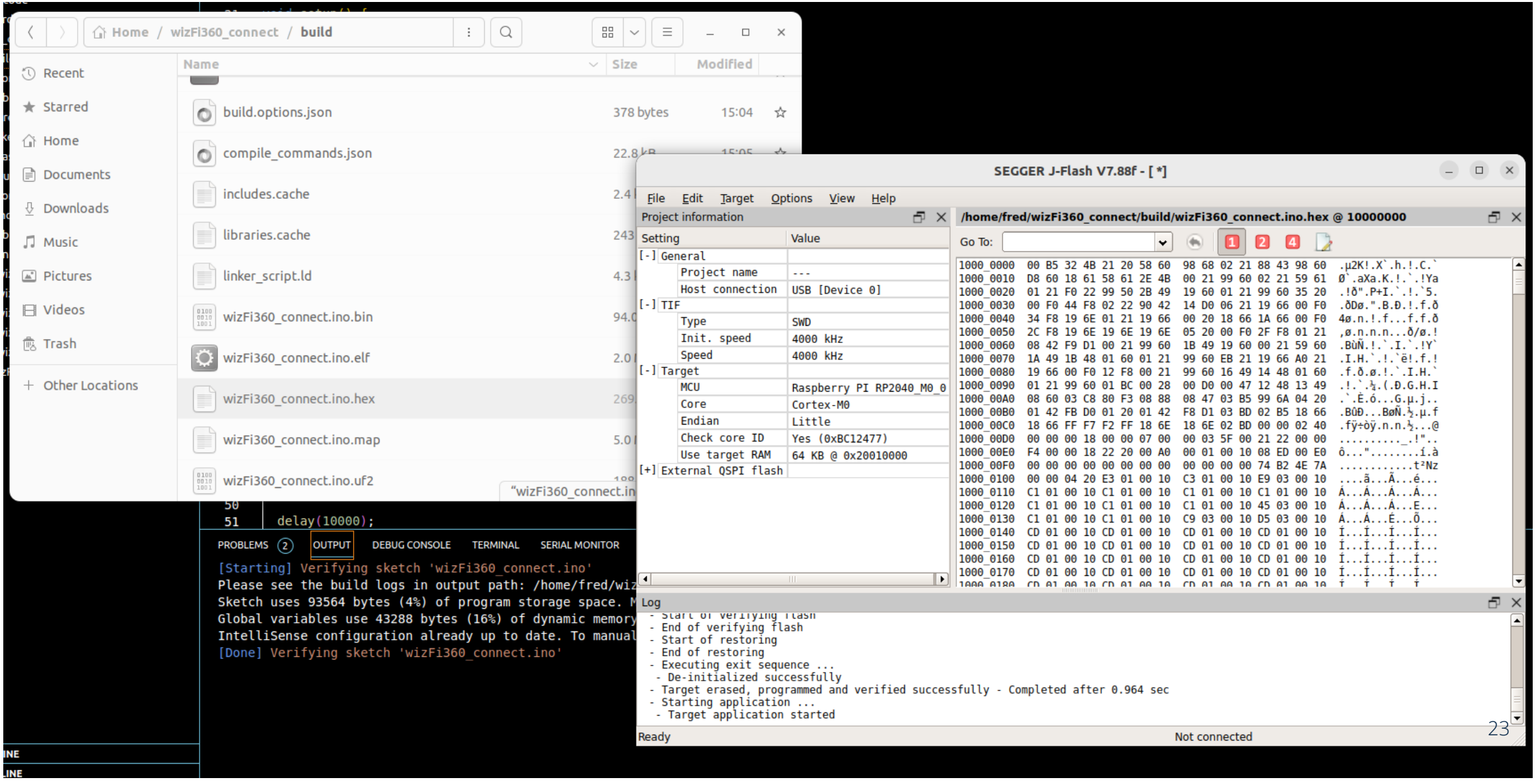
```
147  bool WizFi360Drv::wifiConnect(const char* ssid, const char* passphrase)
148  {
149      int timeout;
150      char cmdBuf[CMD_BUFFER_SIZE];
151
152      LOGDEBUG(F("> wifiConnect"));
153
154      // TODO
155      // Escape character syntax is needed if "SSID" or "password" contains
156      // any special characters (',', '"' and '/')
157      sprintf(cmdBuf,"AT+CWJAP_CUR=\"%s\",\"%s\"",ssid,passphrase);
158      timeout = 20000;
159
160      LOGDEBUG(F("-----------------------------------------------"));
161      LOGDEBUG1(F(">>"), cmdBuf);
162
163      wizfi360Serial->println(cmdBuf);
164
165      int idx = readUntil(timeout);
166
167      LOGDEBUG1(F("------------------------------------------- >"), idx);
168      LOGDEBUG();
169
170      // connect to access point, use CUR mode to avoid connection at boot
171      // int ret = sendCmd(F("AT+CWJAP_CUR=\"%s\",\"%s\""), 20000, ssid, passphrase);
172
173      if (idx==TAG_OK)
174      {
175          LOGINFO1(F("Connected to"), ssid);
176          return true;
177      }
178
179      LOGWARN1(F("Failed connecting to"), ssid);
180
181      // clean additional messages logged after the FAIL tag
182      delay(1000);
183      wizfi360EmptyBuf(false);
184
185      return false;
186  }
```



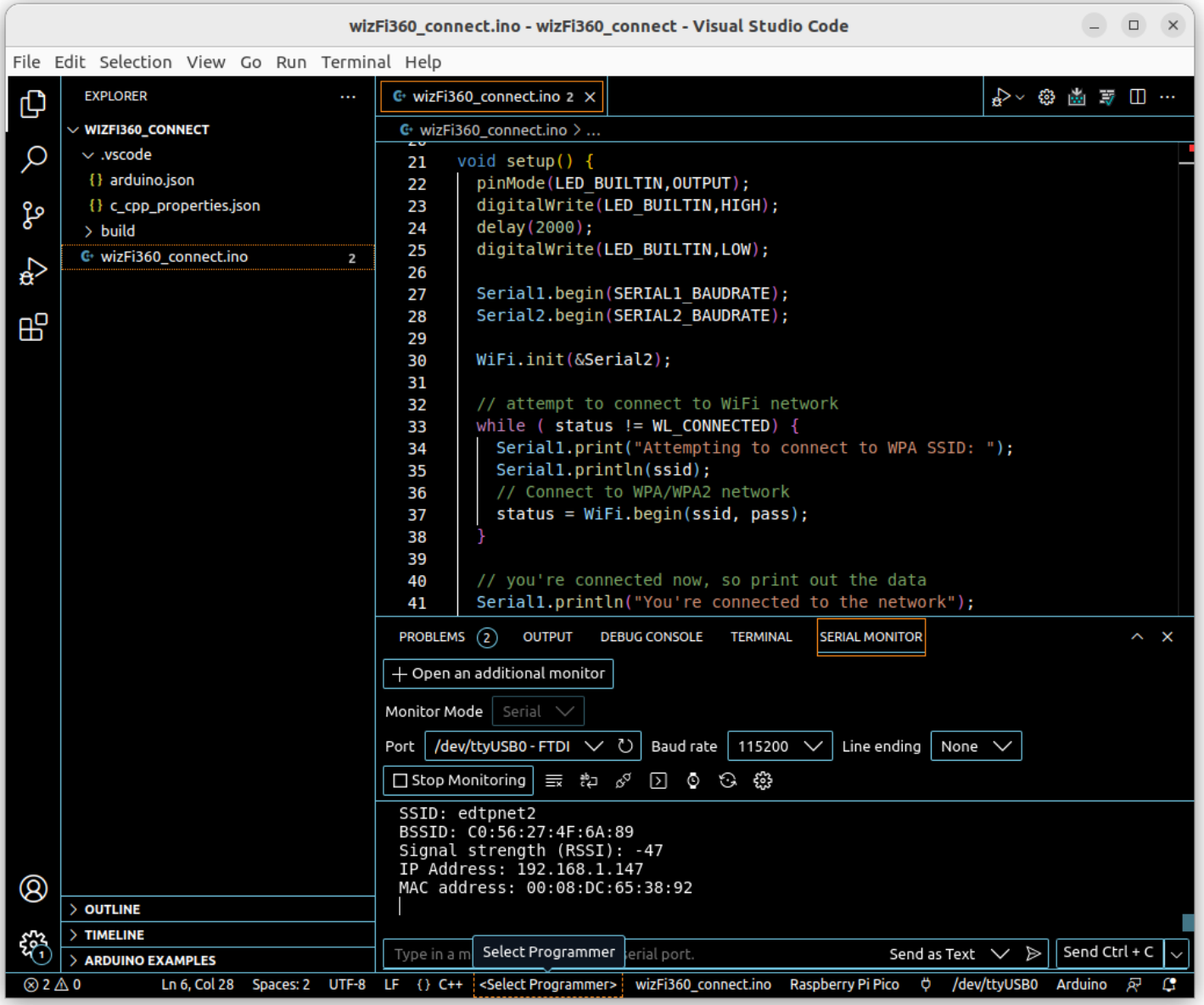
Build the Sketch



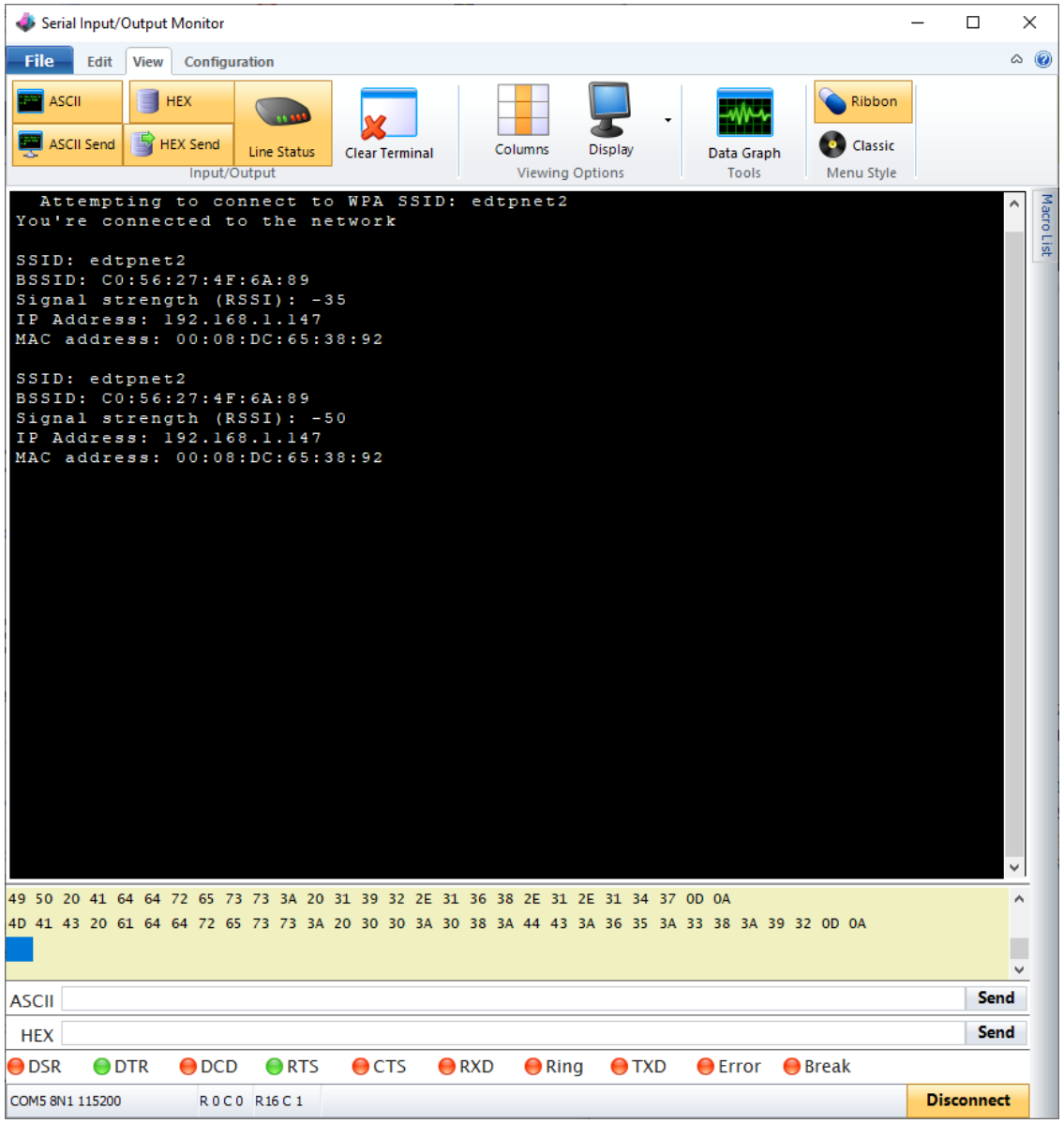
Upload the Sketch Using J-Link

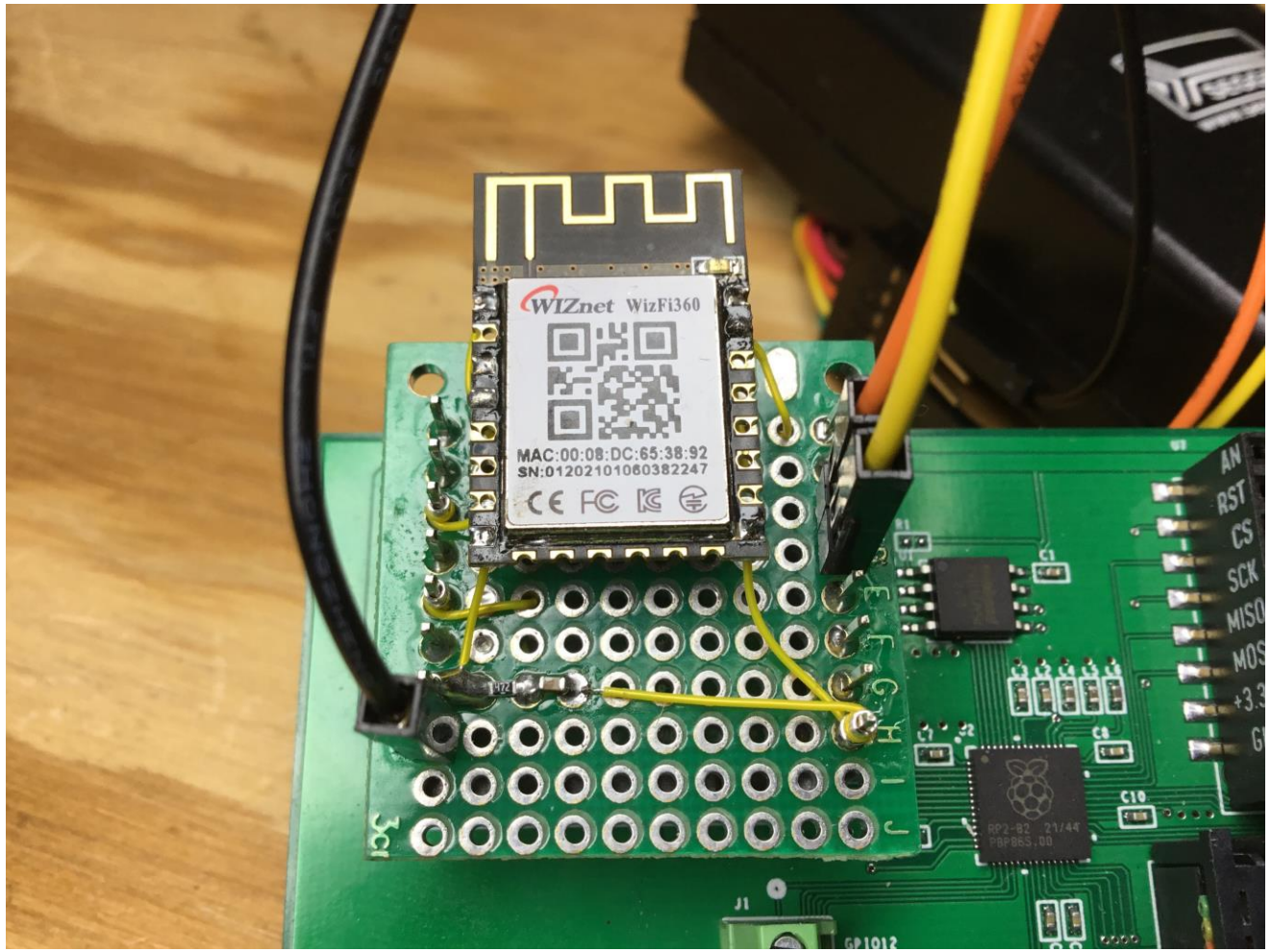


Run the Sketch



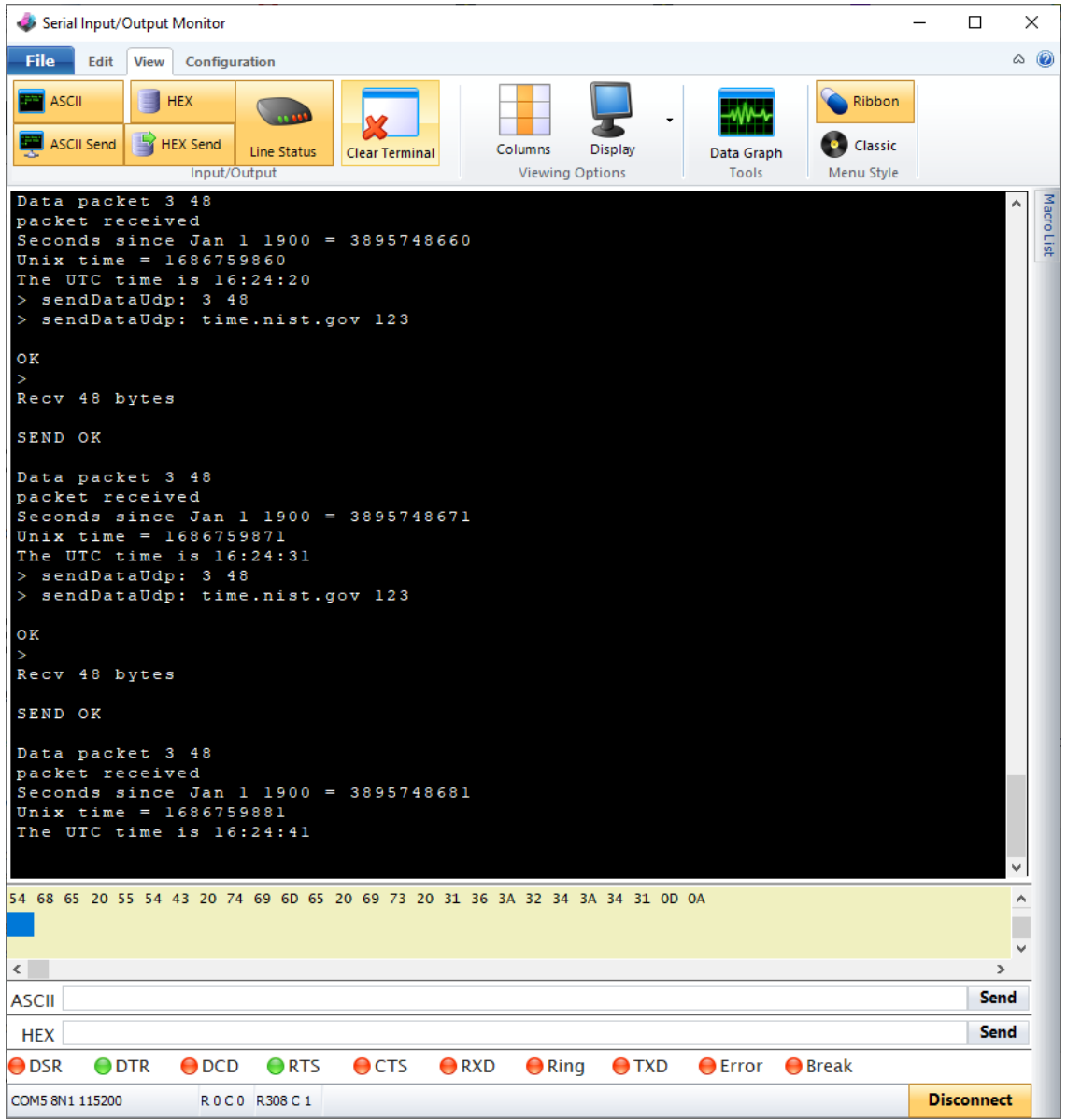
Run the Sketch



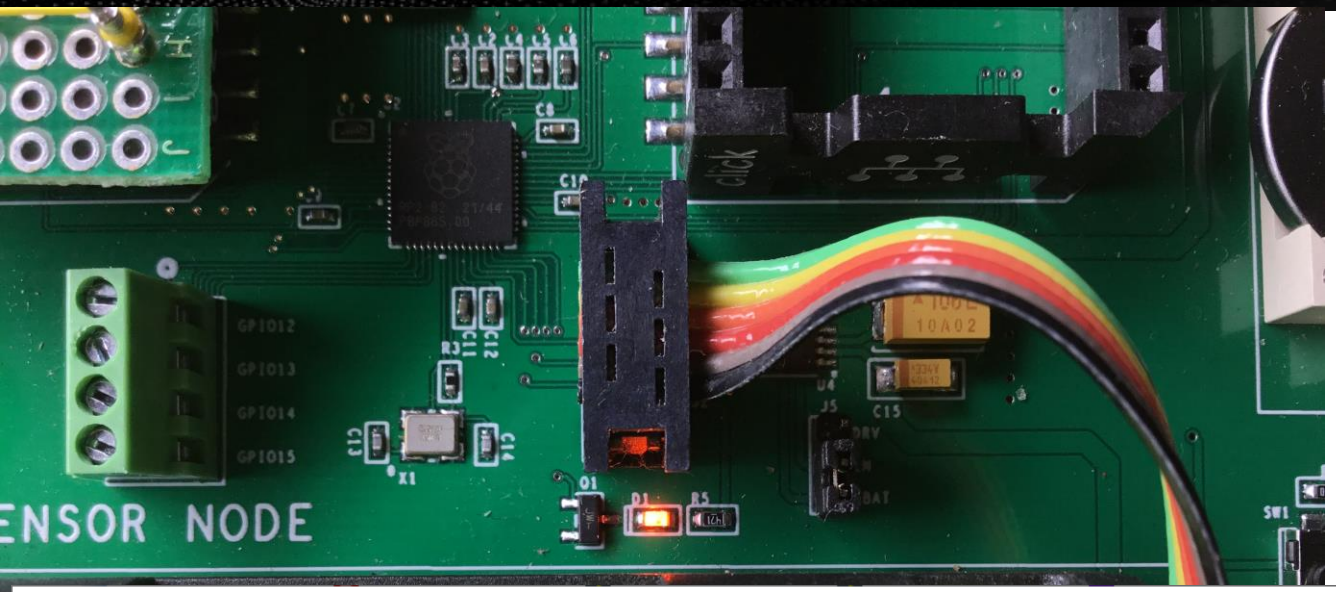


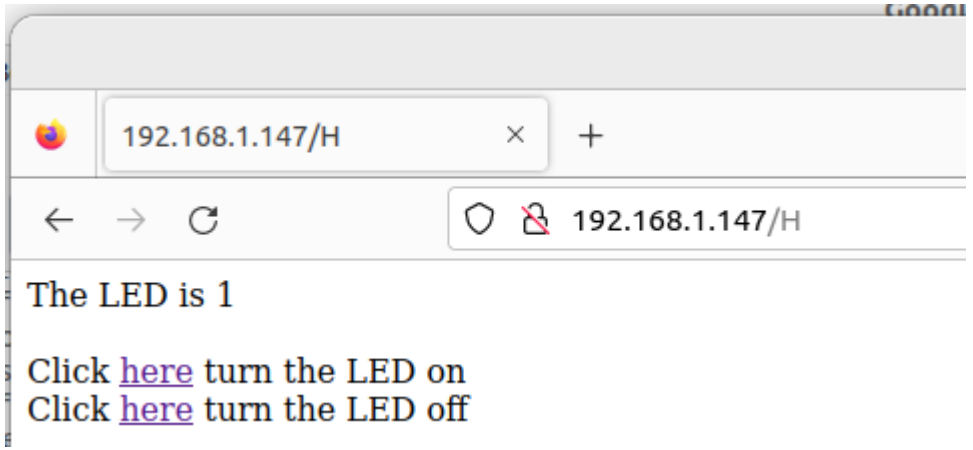
Network Scan

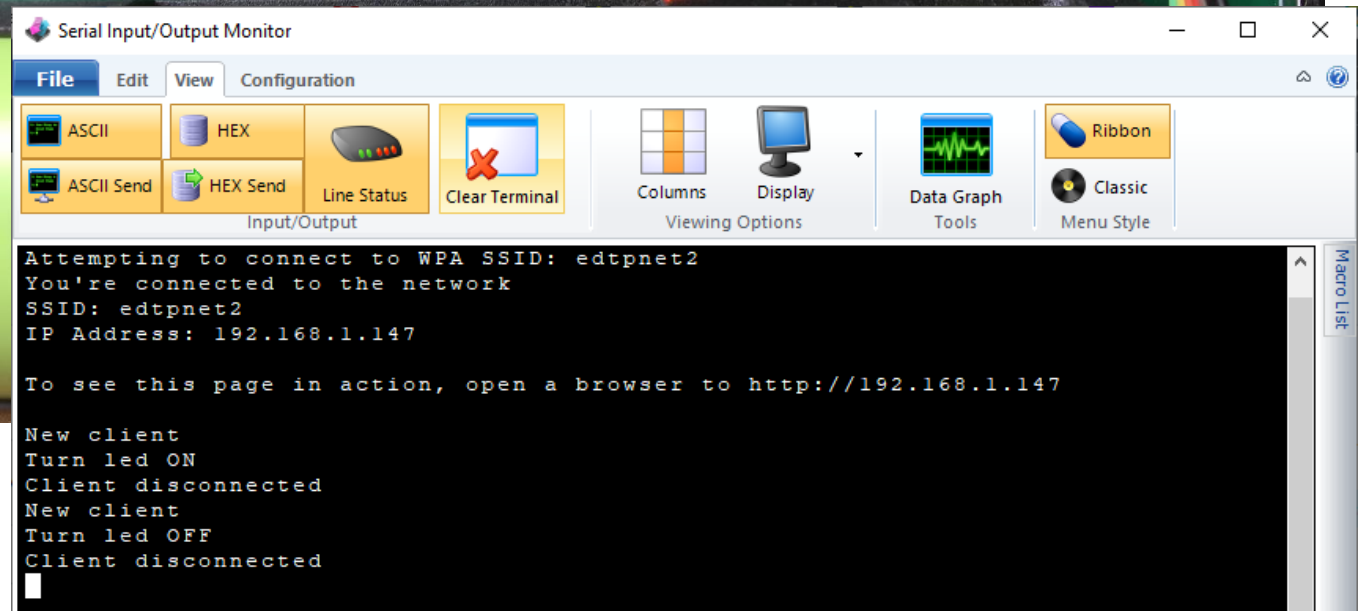
Network Time

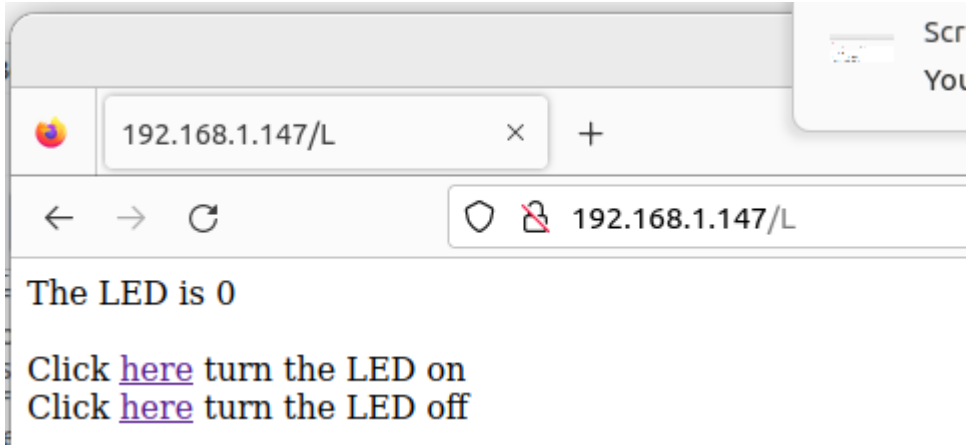


LED Control


ENSOR NODE


192.168.1.147/H
192.168.1.147/H
The LED is 1
Click here turn the LED on
Click here turn the LED off


Serial Input/Output Monitor
File
Edit
View
Configuration
ASCII
HEX
ASCII Send
HEX Send
Line Status
Clear Terminal
Input/Output
Columns
Display
Viewing Options
Data Graph
Tools
Ribbon
Classic
Menu Style
Attempting to connect to WPA SSID: edtpnet2
You're connected to the network
SSID: edtpnet2
IP Address: 192.168.1.147
To see this page in action, open a browser to http://192.168.1.147
New client
Turn led ON
Client disconnected
New client
Turn led OFF
Client disconnected
Macro List

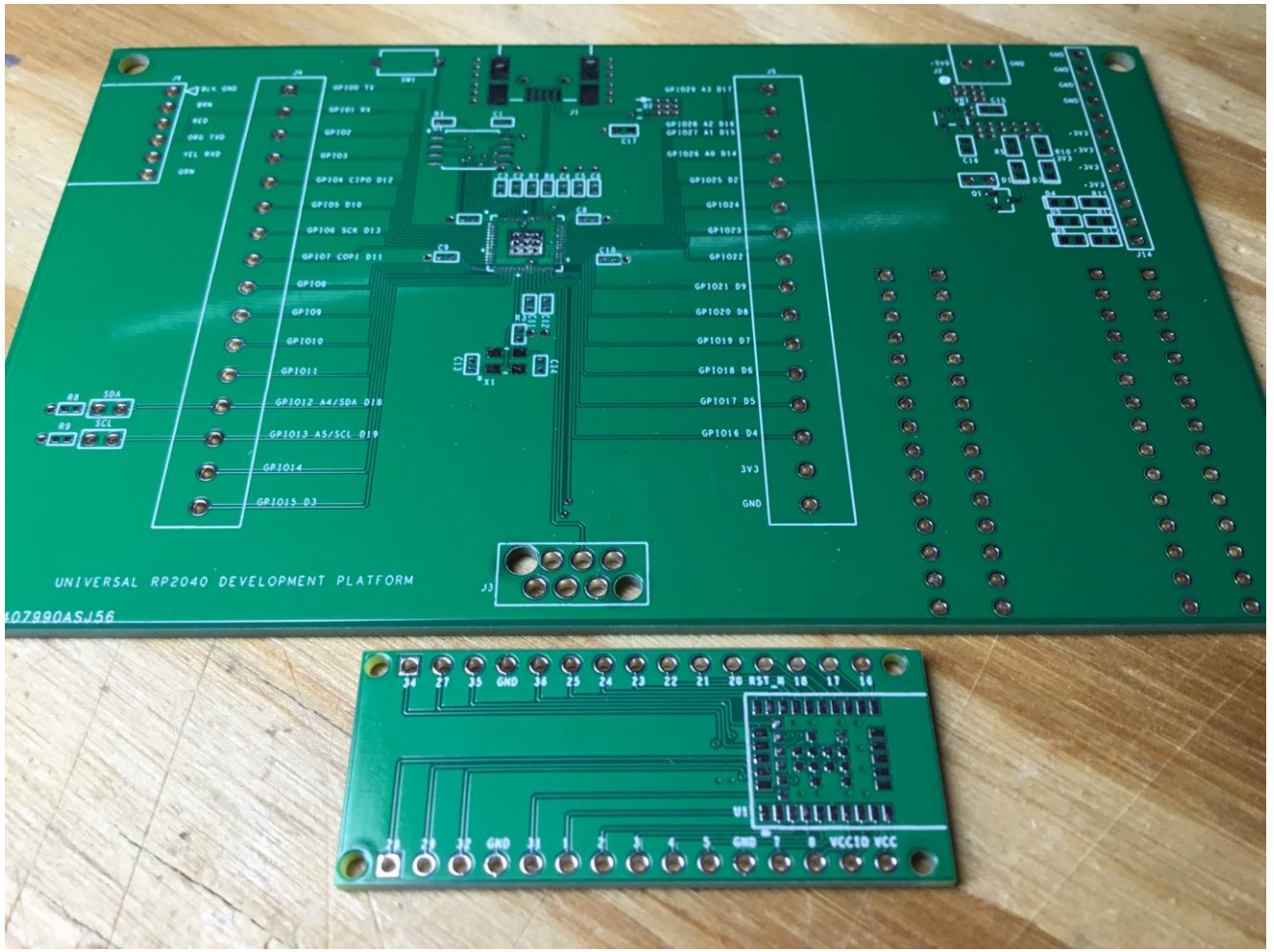

192.168.1.147/L
192.168.1.147/L
The LED is 0
Click here turn the LED on
Click here turn the LED off

Thank you for attending!!!

Please consider the resources below:

- arduino.cc
- wiznet.io
- raspberrypi.org

MORE TO COME..



CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

DigiKey

informa markets