



**DesignNews**

## Designing, Building and Coding Custom Raspberry Pi RP2040 Arduino Devices

### Day 1: RP2040 Arduino Basics

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.

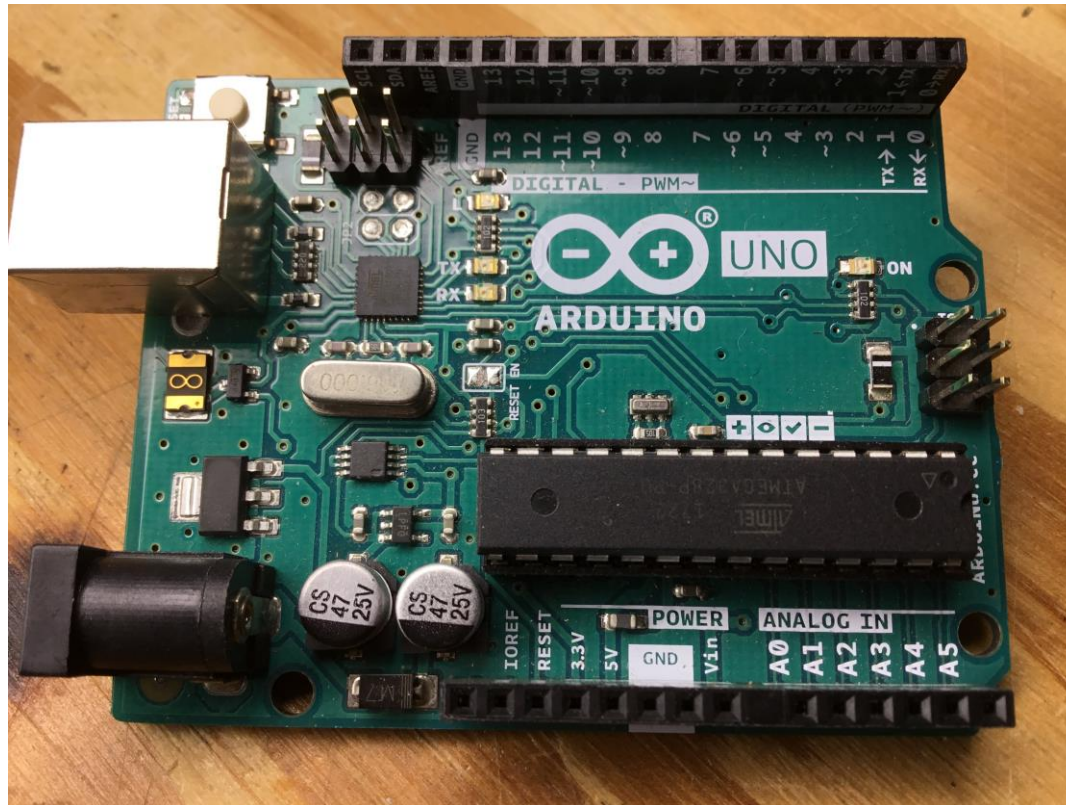


## Fred Eady

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

# AGENDA

- **RP2040 Arduino Toolchain Assembly**
- **RP2040 Arduino Target Hardware**
- **Sketch: wizFi360\_connect.ino**
- **Repaired WizFi360 Arduino Library Sketches**



## Visual Studio Code

Visual Studio Code Docs Updates Blog API Extensions FAQ Learn  [Download](#)

[Version 1.79](#) is now available! Read about the new features and fixes from May.

# Code editing. Redefined.

Free. Built on open source. Runs everywhere.

Download for Windows		Stable Build	
		Stable	Insiders
macOS	Universal	<a href="#">↓</a>	<a href="#">↓</a>
Windows x64	User Installer	<a href="#">↓</a>	<a href="#">↓</a>
Linux x64	.deb	<a href="#">↓</a>	<a href="#">↓</a>
	.rpm	<a href="#">↓</a>	<a href="#">↓</a>
<a href="#">Other downloads</a> or <a href="#">open on web</a>			

The screenshot shows the Visual Studio Code interface. On the left, the Extensions Marketplace is open, displaying a list of installed and available extensions. The main editor area shows a JavaScript file named 'serviceWorker.js' with code for registering a service worker. A dropdown menu is visible over the code, listing various browser APIs and properties. At the bottom, a terminal window shows the output of a 'node' command, indicating that the application is running on localhost:3000.

**EXTENSIONS: MARKETPLACE**

- @sortinstalls**
- Python** 2019.6.24221 54.9M ★ 4.5  
Linting, Debugging (multi-threaded, ...  
Microsoft [Install](#)
- GitLens — Git sup...** 9.8.5 23.1M ★ 5  
Supercharge the Git capabilities buil...  
Eric Amodio [Install](#)
- C/C++** 0.24.0 23M ★ 3.5  
C/C++ IntelliSense, debugging, and ...  
Microsoft [Install](#)
- ESLint** 1.9.0 21.9M ★ 4.5  
Integrates ESLint JavaScript into VS...  
Dirk Baeumer [Install](#)
- Debugger for Ch...** 4.11.6 20.6M ★ 4  
Debug your JavaScript code in the C...  
Microsoft [Install](#)
- Language Supp...** 0.47.0 18.7M ★ 4.5  
Java Linting, Intellisense, formatting, ...  
Red Hat [Install](#)
- vscode-icons** 8.8.0 17.2M ★ 5  
Icons for Visual Studio Code  
VSCode Icons Team [Install](#)
- Vetur** 0.21.1 17M ★ 4.5  
Vue tooling for VS Code  
Pine Wu [Install](#)
- C#** 1.21.0 15.6M ★ 4  
C# for Visual Studio Code (powered ...  
Microsoft [Install](#)

**Code Editor:** JS serviceWorker.js

```

src > JS serviceWorker.js > register > window.addEventListener('load') callback
39
40
41   checkValidServiceWorker(swUrl, config);
42   // Add some additional logging to localhost, p
43   // service worker/PWA documentation.
44   navigator.serviceWorker.ready.then(() => {
45     product
46     productSub
47     removeSiteSpecificTrackingException
48     removeWebWideTrackingException
49     requestMediaKeySystemAccess
50     sendBeacon
51     serviceWorker (property) Navigator.serviceWorke...
52     storage
53     storeSiteSpecificTrackingException
54     storeWebWideTrackingException
55     userAgent
56     vendor
57   })
58   function registerValidSW(swUrl, config) {
59     navigator.serviceWorker
60     .register(swUrl)
61     .then(registration => {

```

**Terminal:** 1: node

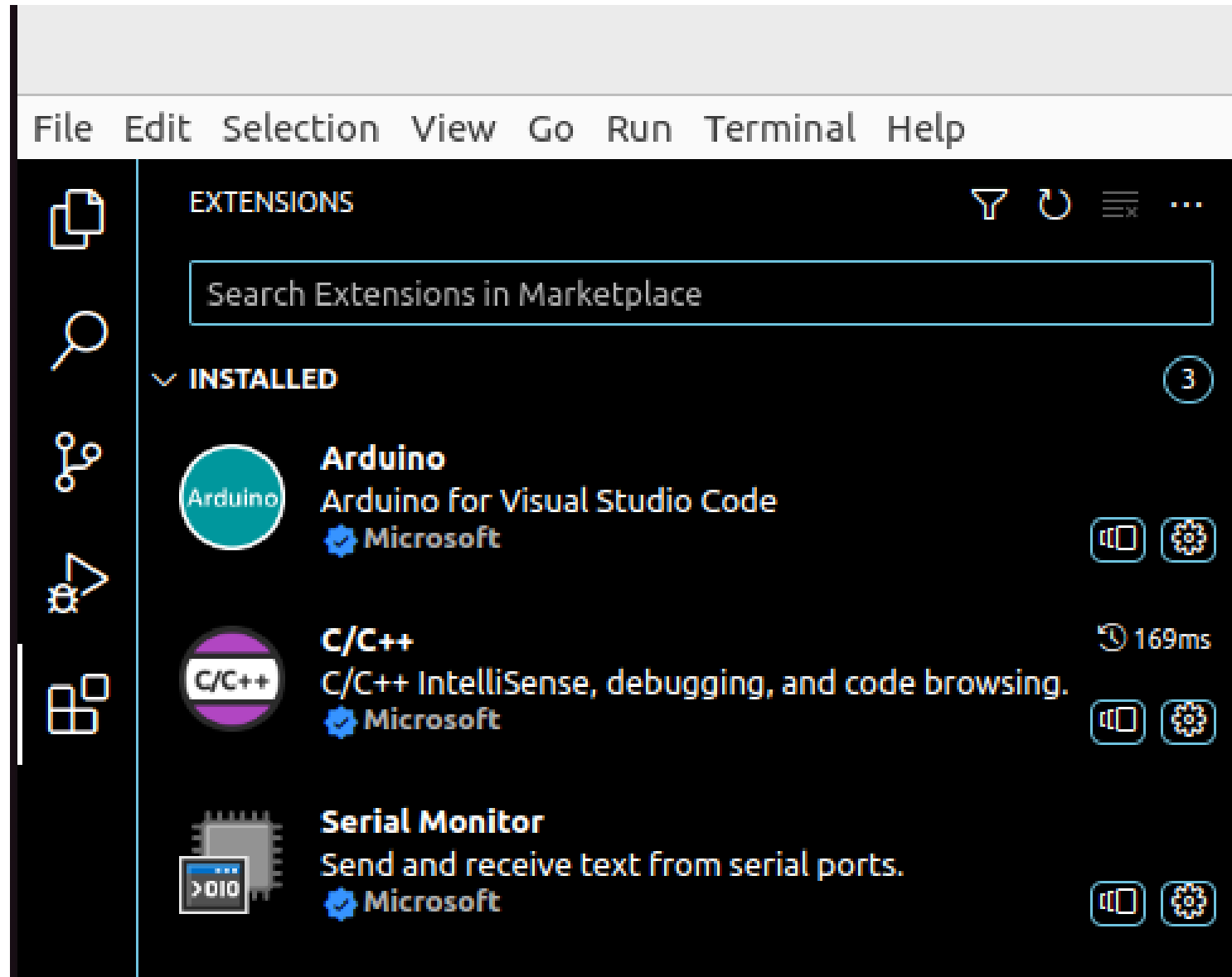
You can now view **create-react-app** in the browser.

Local: http://localhost:3000/  
On Your Network: http://10.211.55.3:3000/

Note that the development build is not optimized.

Ln 43, Col 19 Spaces: 2 UTF-8 LF JavaScript

## Arduino Extension for Visual Studio Code



# Install RP2040-Based Arduino Boards

&gt;

Add Selection To Previous Find Match

Arduino CLI: Upload

Arduino CLI: Upload Using Programmer

Arduino: Board Config

Arduino: Board Manager

Arduino: Change Board Type

Arduino: Change Timestamp Format

Arduino: Close Serial Monitor

Arduino: Examples

Arduino: Initialize

Arduino: Library Manager

Arduino: Open Serial Monitor

Arduino: Rebuild IntelliSense Configuration Ctrl + Alt + I

Arduino: Select Programmer

Arduino: Select Serial Port

Arduino: Select Sketch

Arduino: Send Text to Serial Port

Arduino: Upload Ctrl + Alt + U

Arduino: Upload Using Programmer

Arduino Board Manager

Type

All

Filter your search...

Refresh Package Indexes

**Arduino Mbed OS Nano Boards** by Arduino

Boards included in this package:

Arduino Nano 33 BLE, Arduino Nano 33 BLE Sense, Arduino Nano RP2040 Connect

[Online help](#)[Online help](#)[More info](#)

4.0.2

Install

**Arduino Mbed OS Portenta Boards** by Arduino

Boards included in this package:

Arduino Portenta H7, Arduino Portenta X8

[Online help](#)[Online help](#)[More info](#)

4.0.2

Install

**Arduino Mbed OS RP2040 Boards** by Arduino

Boards included in this package:

Raspberry Pi Pico

[Online help](#)[Online help](#)[More info](#)

4.0.2

Install

## 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.

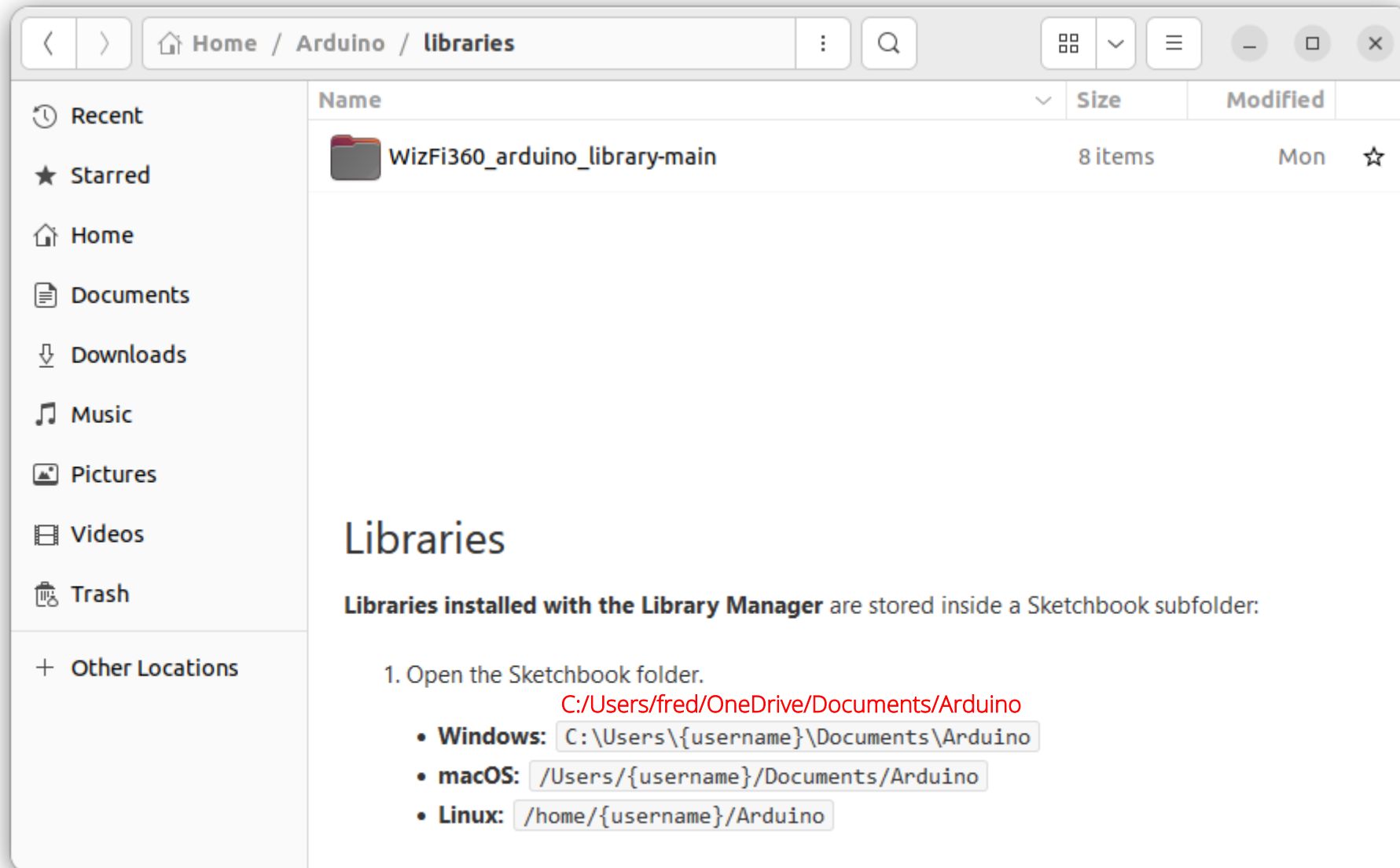
Please note that in this guide examples were configured and tested using **Arduino IDE** in **Windows** environment.

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

## 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](#)

Version

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

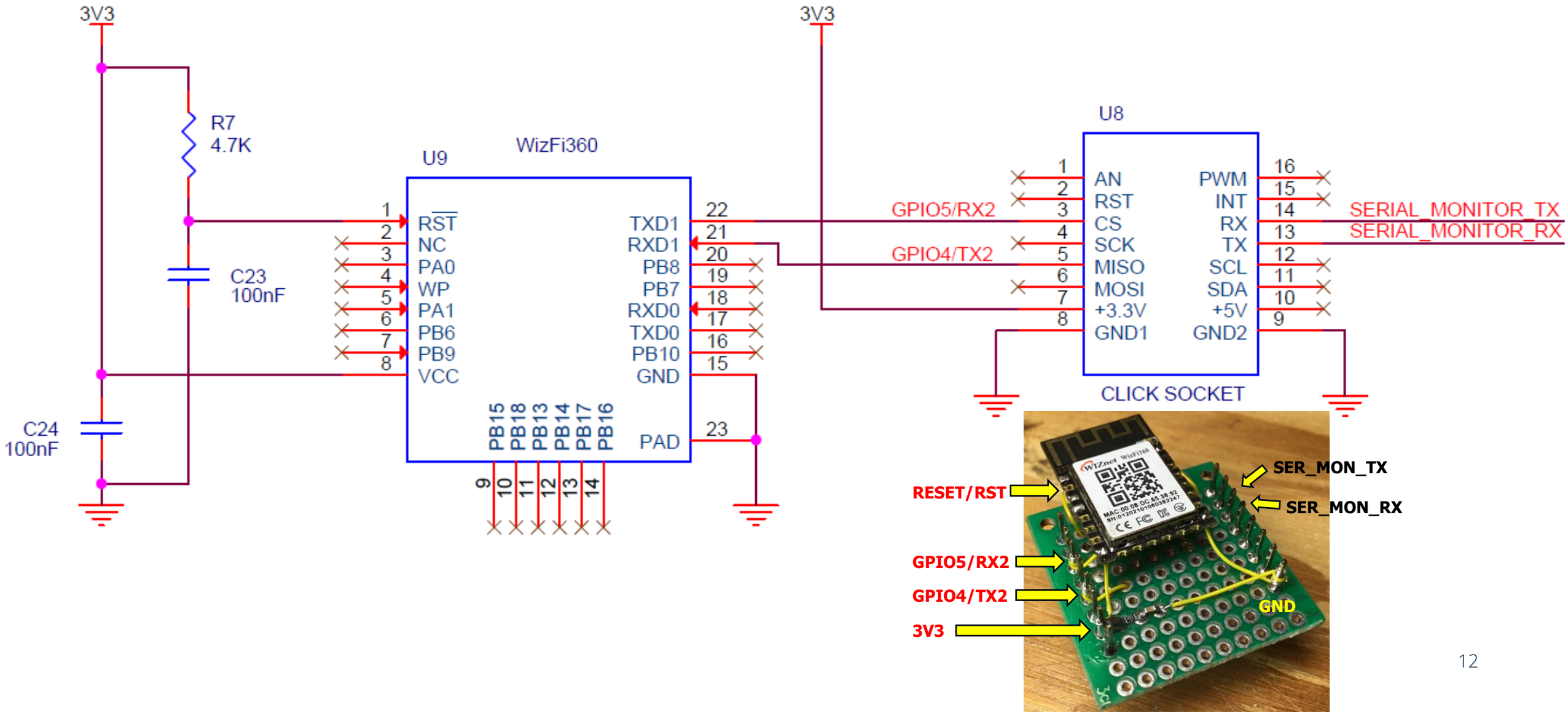
[64-bit DEB Installer](#) [32-bit DEB Installer](#) [64-bit TGZ Archive](#) [32-bit TGZ Archive](#)

## macOS

[64-bit Installer](#) [64-bit Apple M1 Installer](#) [Universal Installer](#)

**WizFi360 Pseudo *click***

**WizFi360 Pseudo *click***



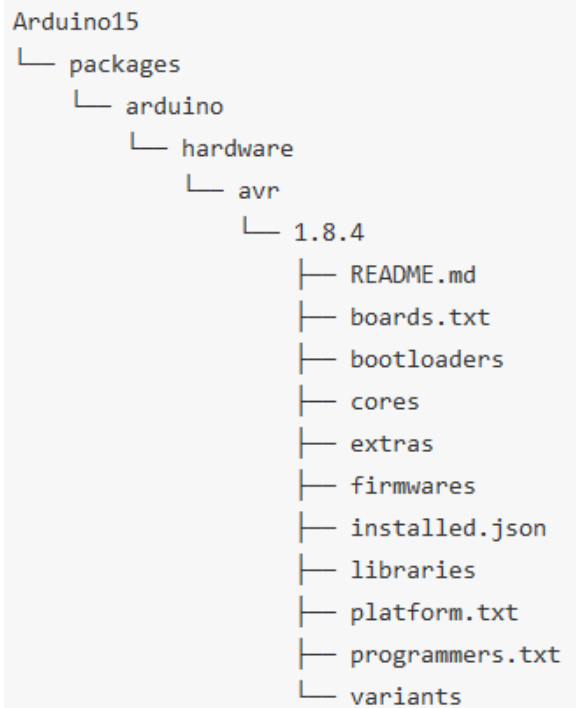
## WizFi360 Pseudo *click* – GPIO Pin Assignments

### Board platforms and cores

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

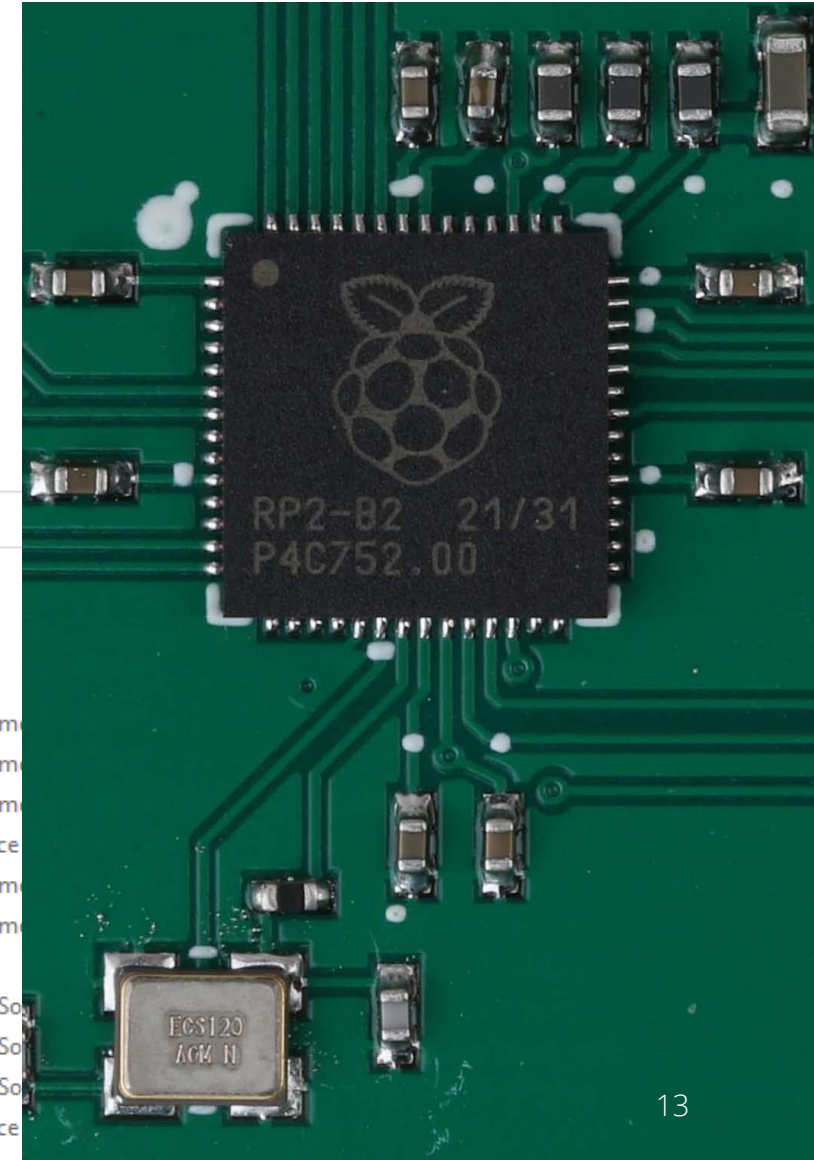
1. Open the Arduino15 folder.

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



hardware > mbed\_rp2040 > 4.0.2 > variants > RASPBERRY\_PI\_PICO >

Name	Date modified	Type
avr	6/1/2023 12:01 PM	File folder
conf	6/1/2023 12:01 PM	File folder
lib3	6/1/2023 12:01 PM	File folder
mbd_nano	6/1/2023 12:01 PM	File folder
mbd_rp2040	6/1/2023 12:01 PM	File folder
samd	6/1/2023 12:01 PM	File folder
cflags.txt	6/1/2023 12:01 PM	Text Document
cxxflags.txt	6/1/2023 12:01 PM	Text Document
defines.txt	6/1/2023 12:01 PM	Text Document
double_tap_usb_boot.cpp	6/1/2023 12:01 PM	C++ Source Code
includes.txt	6/1/2023 12:01 PM	Text Document
ldflags.txt	6/1/2023 12:01 PM	Text Document
linker_script.ld	6/1/2023 12:01 PM	LD File
mbed_config.h	6/1/2023 12:01 PM	C Header Source File
pinmode_arduino.h	6/1/2023 12:01 PM	C Header Source File
pins_arduino.h	7/10/2023 2:24 PM	C Header Source File
variant.cpp	6/27/2023 9:26 AM	C++ Source Code

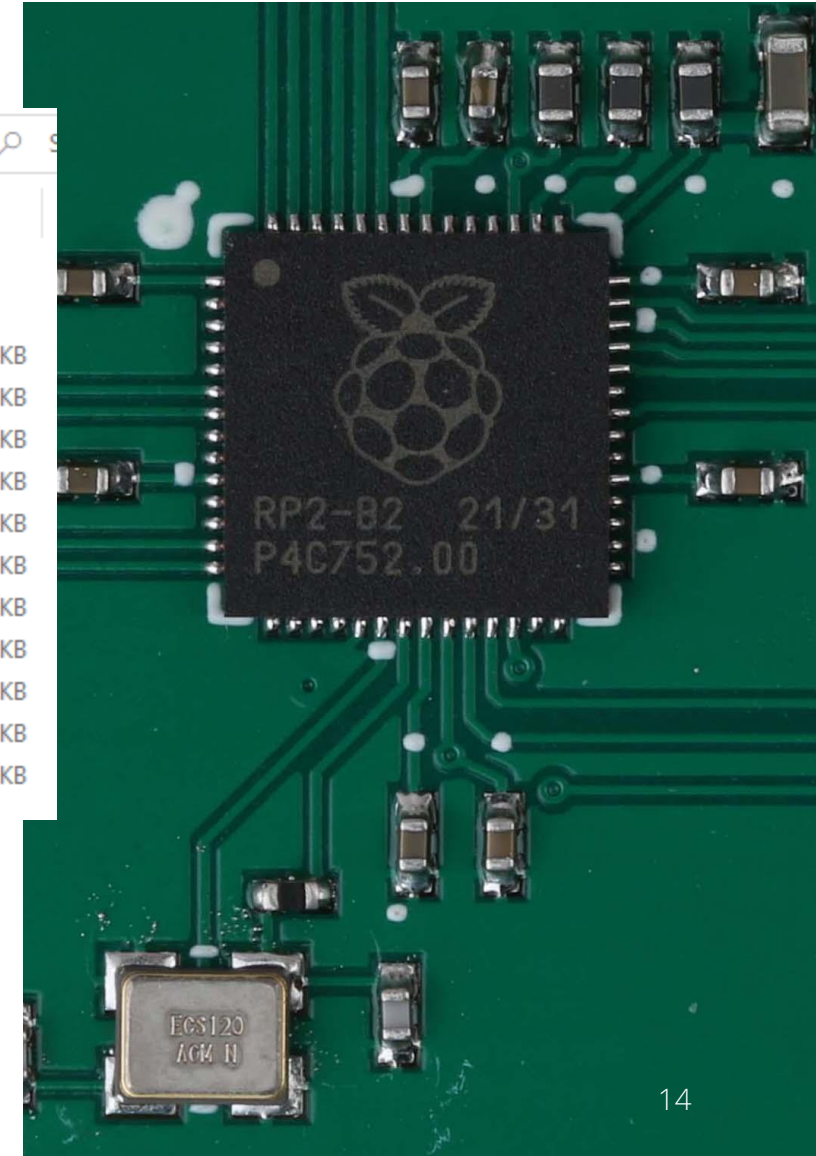


WizFi360 Pseudo *click* – GPIO Pins - *variant.cpp*

```
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 },  
};
```

File Explorer view showing the directory structure for the RASPBERRY\_PI\_PICO variant. The path is hardware > mbed\_rp2040 > 4.0.2 > variants > RASPBERRY\_PI\_PICO. The file variant.cpp is highlighted with a red arrow.

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

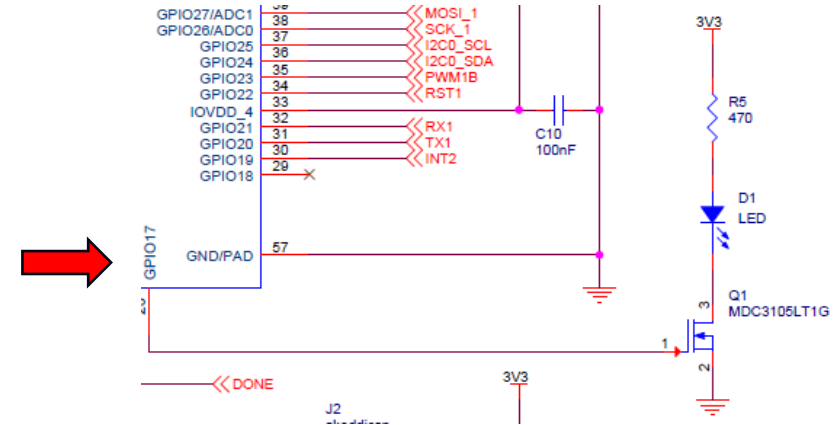


WizFi360 Pseudo *click* – GPIO Pins - pins\_arduino.h

```

21 // LEDs
22 // ----
23 #define PIN_LED      (17u)
24 #define LED_BUILTIN PIN_LED
25
26 // Serial
27 #define PIN_SERIAL_TX (0u)
28 #define PIN_SERIAL_RX (1u)
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

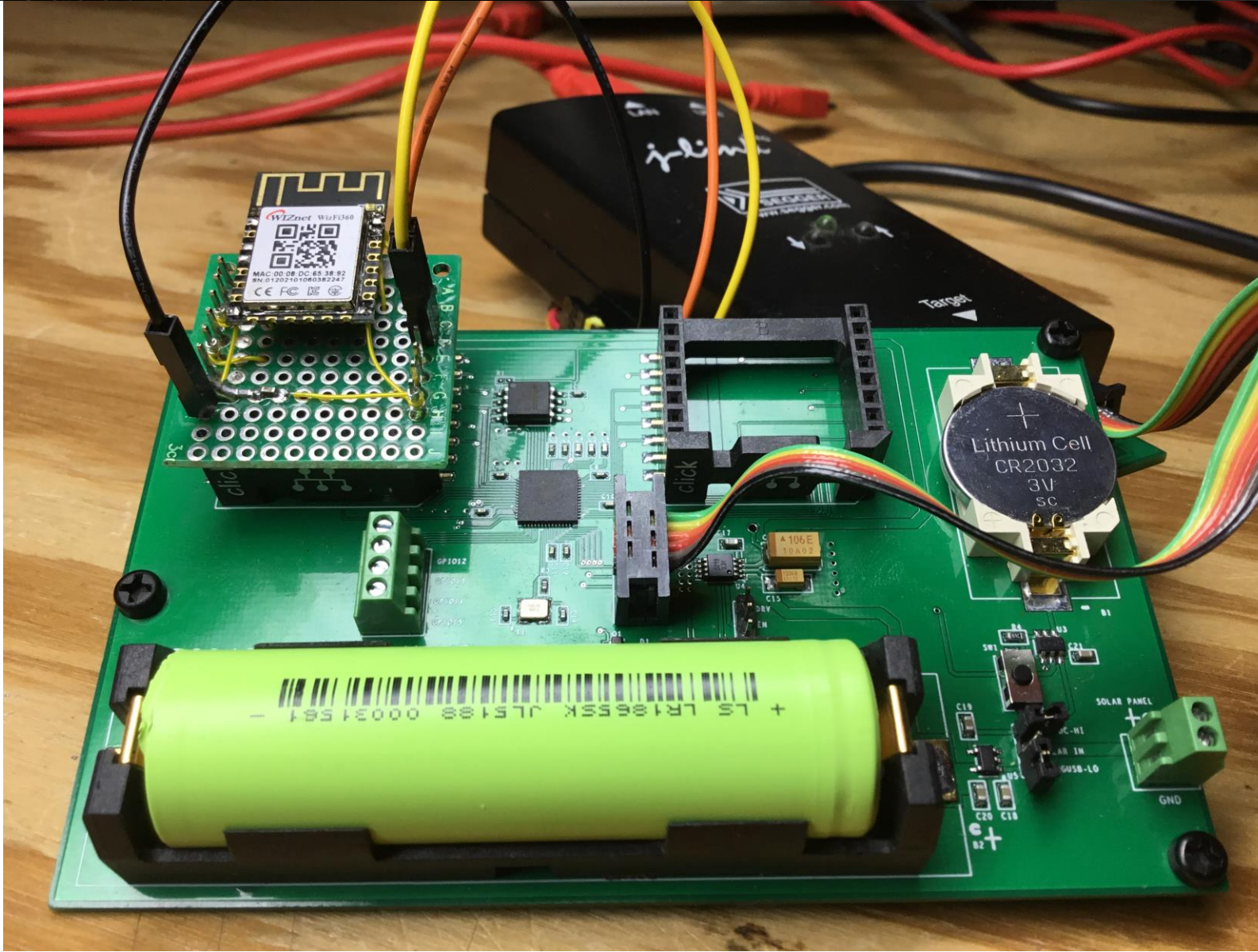
```



hardware > mbed\_rp2040 > 4.0.2 > variants > RASPBERRY\_PI\_PICO >

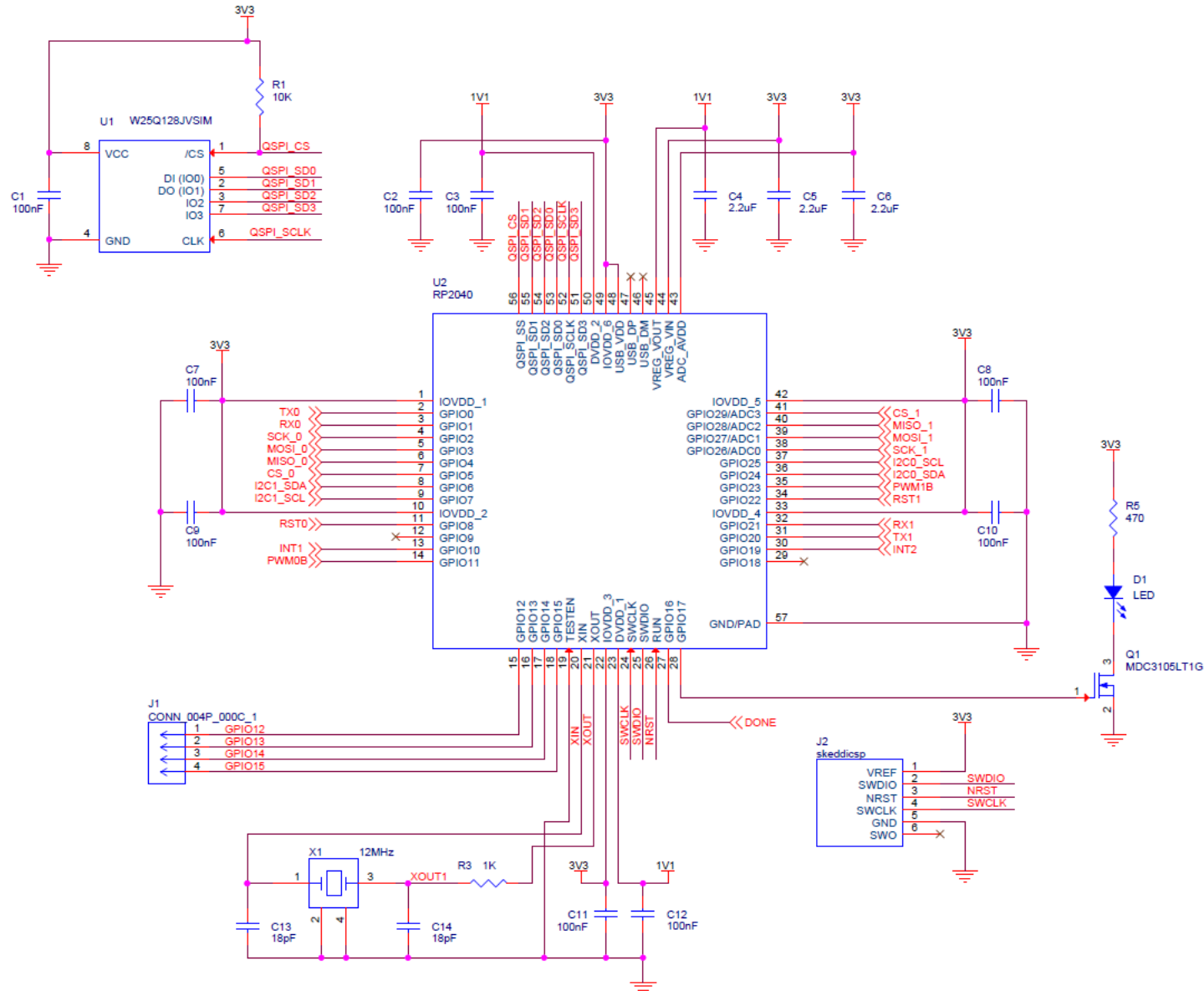
Name	Date modified	Type	Size
conf	6/1/2023 12:01 PM	File folder	
libs	6/1/2023 12:01 PM	File folder	
cflags.txt	6/1/2023 12:01 PM	Text Document	1 KB
cxxflags.txt	6/1/2023 12:01 PM	Text Document	1 KB
defines.txt	6/1/2023 12:01 PM	Text Document	2 KB
double_tap_usb_boot.cpp	6/1/2023 12:01 PM	C++ Source 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	15 2 KB

# 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 ('', '\', '\n' 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\"", ssid, passphrase), 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

```
wizFi360_connect.ino - wizFi360_connect - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
WIZFI360_CONNECT
  .vscode
  arduino.json
  c_cpp_properties.json
  build
    core
    libraries
    preproc
    sketch
    .last-used
  build.options.json
  compile_commands.json
  includes.cache
  libraries.cache
  linker_script.ld
  wizFi360_connect.ino.bin
  wizFi360_connect.ino.elf
  wizFi360_connect.ino.hex
  wizFi360_connect.ino.map
  wizFi360_connect.ino.uf2
wizFi360_connect.ino
2

wizFi360_connect.ino > ...
20
21 void setup() {
22   pinMode(LED_BUILTIN,OUTPUT);
23   digitalWrite(LED_BUILTIN,HIGH);
24   delay(2000);
25   digitalWrite(LED_BUILTIN,LOW);
26
27   Serial1.begin(SERIAL1_BAUDRATE);
28   Serial2.begin(SERIAL2_BAUDRATE);
29
30   WiFi.init(&Serial2);
31
32   // attempt to connect to WiFi network
33   while ( status != WL_CONNECTED) {
34     Serial1.print("Attempting to connect to WPA SSID: ");
35     Serial1.println(ssid);
36     // Connect to WPA/WPA2 network
37     status = WiFi.begin(ssid, pass);
38   }
39
40   // you're connected now, so print out the data
41   Serial1.println("You're connected to the network");

```

OUTPUT

```
[Starting] Verifying sketch 'wizFi360_connect.ino'
Please see the build logs in output path: /home/fred/wizFi360_connect/build
Sketch uses 93564 bytes (4%) of program storage space. Maximum is 2097152
bytes.
Global variables use 43288 bytes (16%) of dynamic memory, leaving 227048 bytes
for local variables. Maximum is 270336 bytes.
IntelliSense configuration already up to date. To manually rebuild your
IntelliSense configuration run "Ctrl+Alt+I"
[Done] Verifying sketch 'wizFi360_connect.ino'
```

Ln 6, Col 28 Spaces: 2 UTF-8 LF {} C++ <Select Programmer> wizFi360\_connect.ino Raspberry Pi Pico /dev/ttyUSB0

## Upload the Sketch Using J-Link

The screenshot displays the IDE interface during the upload of a sketch to a Raspberry Pi RP2040 using J-Link. The IDE window shows the sketch code, the build directory, and the J-Flash tool interface.

**IDE Window:**

- File Explorer: `Home / wizFi360_connect / build`
- Files: `build.options.json` (378 bytes, 15:04), `compile_commands.json` (22.8 kB, 15:05), `includes.cache` (2.4 kB), `libraries.cache` (2.4 kB), `linker_script.ld` (4.3 kB), `wizFi360_connect.ino.bin` (94.0 kB), `wizFi360_connect.ino.elf` (2.0 kB), `wizFi360_connect.ino.hex` (269 kB), `wizFi360_connect.ino.map` (5.0 kB), `wizFi360_connect.ino.uf2` (100 kB).
- Code Editor: Line 50: `50`, Line 51: `51 delay(10000);`
- Output Console: 

```
[Starting] Verifying sketch 'wizFi360_connect.ino'
Please see the build logs in output path: /home/fred/wizFi360_connect/build/
Sketch uses 93564 bytes (4%) of program storage space. Maximum allowed is 2048000 bytes.
Global variables use 43288 bytes (16%) of dynamic memory, leaving 161512 bytes (64%) free.
IntelliSense configuration already up to date. To manually update, use the Ctrl+Shift+P command and type "Arduino: Update IntelliSense configuration"
[Done] Verifying sketch 'wizFi360_connect.ino'
```

**SEGGER J-Flash V7.88f - [\*]**

Project information: `/home/fred/wizFi360_connect/build/wizFi360_connect.ino.hex @ 10000000`

Setting Value

- [-] General
  - Project name: ---
  - Host connection: USB [Device 0]
- [-] TIF
  - Type: SWD
  - Init. speed: 4000 kHz
  - Speed: 4000 kHz
- [-] Target
  - MCU: Raspberry PI RP2040\_M0\_0
  - Core: Cortex-M0
  - Endian: Little
  - Check core ID: Yes (0xBC12477)
  - Use target RAM: 64 KB @ 0x20010000
- [+] External QSPI flash

Log

- Start of verifying flash
- End of verifying flash
- Start of restoring
- End of restoring
- Executing exit sequence ...
- De-initialized successfully
- Target erased, programmed and verified successfully - Completed after 0.964 sec
- Starting application ...
- Target application started

Ready

Not connected

23

## Run the Sketch

```
wizFi360_connect.ino - wizFi360_connect - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
WIZFI360_CONNECT
  .vscode
  {} arduino.json
  {} c_cpp_properties.json
  > build
  wizFi360_connect.ino 2
wizFi360_connect.ino > ...
21 void setup() {
22   pinMode(LED_BUILTIN,OUTPUT);
23   digitalWrite(LED_BUILTIN,HIGH);
24   delay(2000);
25   digitalWrite(LED_BUILTIN,LOW);
26
27   Serial1.begin(SERIAL1_BAUDRATE);
28   Serial2.begin(SERIAL2_BAUDRATE);
29
30   WiFi.init(&Serial2);
31
32   // attempt to connect to WiFi network
33   while ( status != WL_CONNECTED) {
34     Serial1.print("Attempting to connect to WPA SSID: ");
35     Serial1.println(ssid);
36     // Connect to WPA/WPA2 network
37     status = WiFi.begin(ssid, pass);
38   }
39
40   // you're connected now, so print out the data
41   Serial1.println("You're connected to the network");
```

PROBLEMS (2) OUTPUT DEBUG CONSOLE TERMINAL SERIAL MONITOR

+ Open an additional monitor

Monitor Mode Serial

Port /dev/ttyUSB0 - FTDI Baud rate 115200 Line ending None

Stop Monitoring

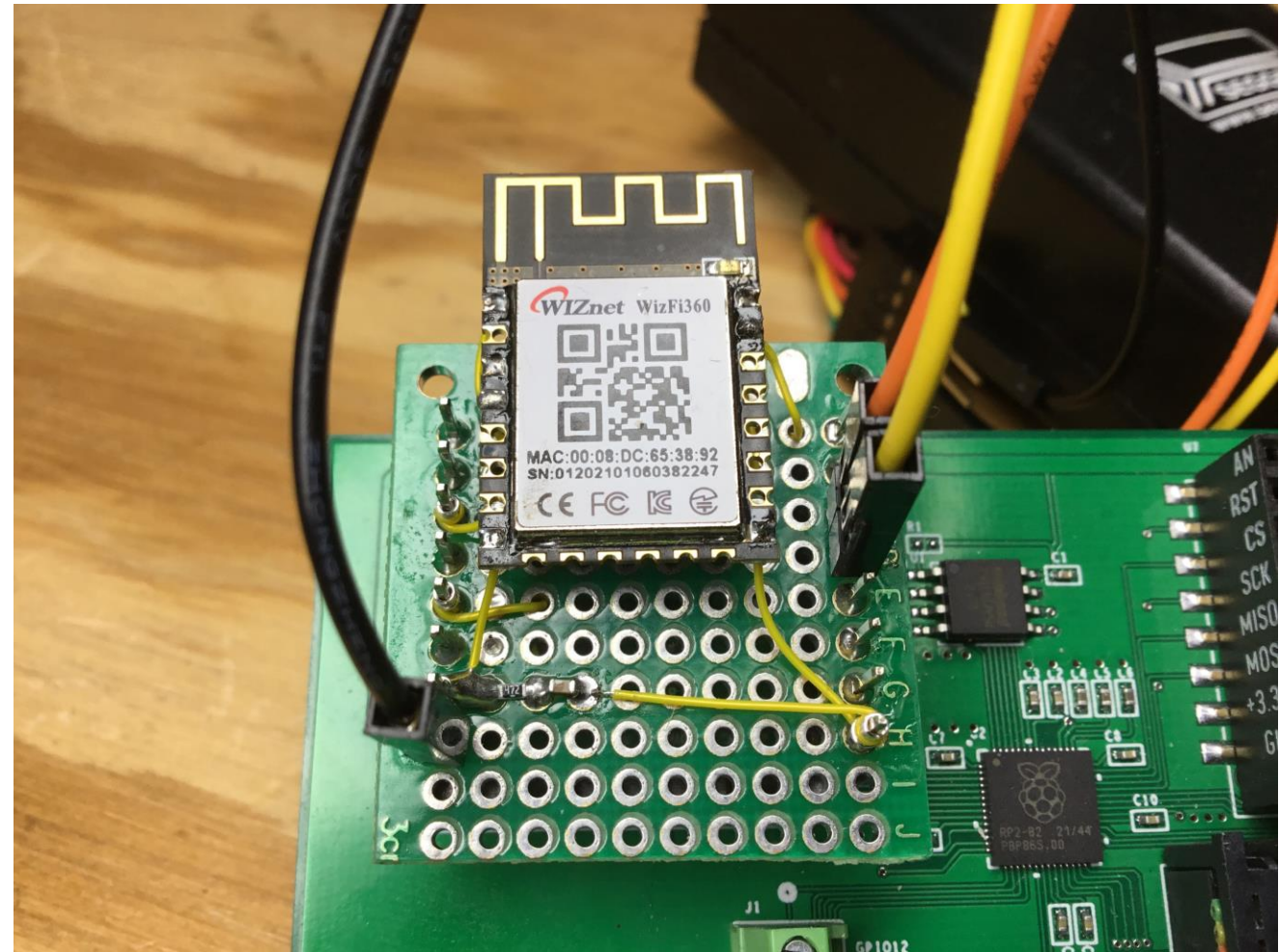
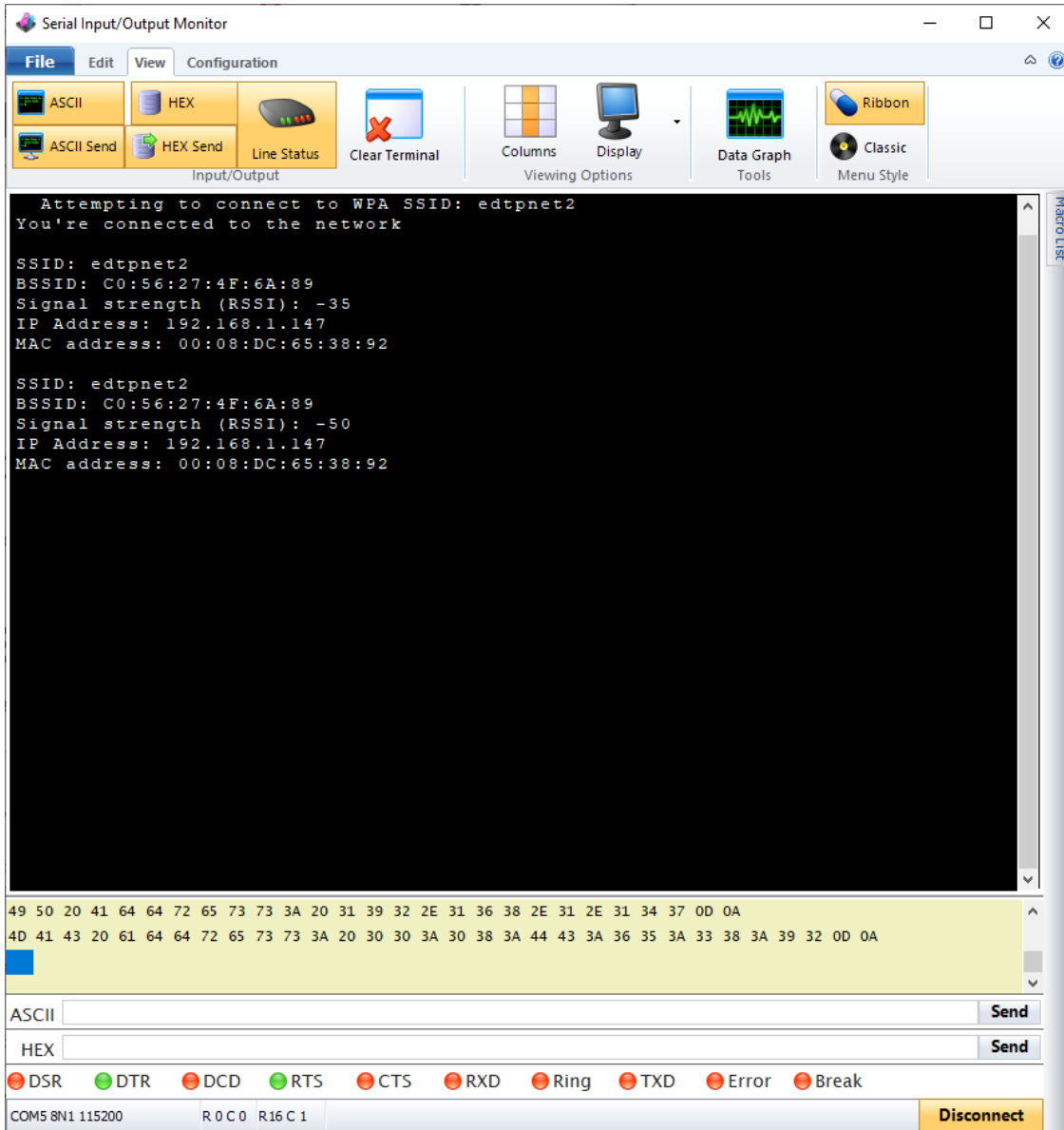
SSID: edtpnet2  
BSSID: C0:56:27:4F:6A:89  
Signal strength (RSSI): -47  
IP Address: 192.168.1.147  
MAC address: 00:08:DC:65:38:92

Type in a m Select Programmer serial port. Send as Text Send Ctrl + C

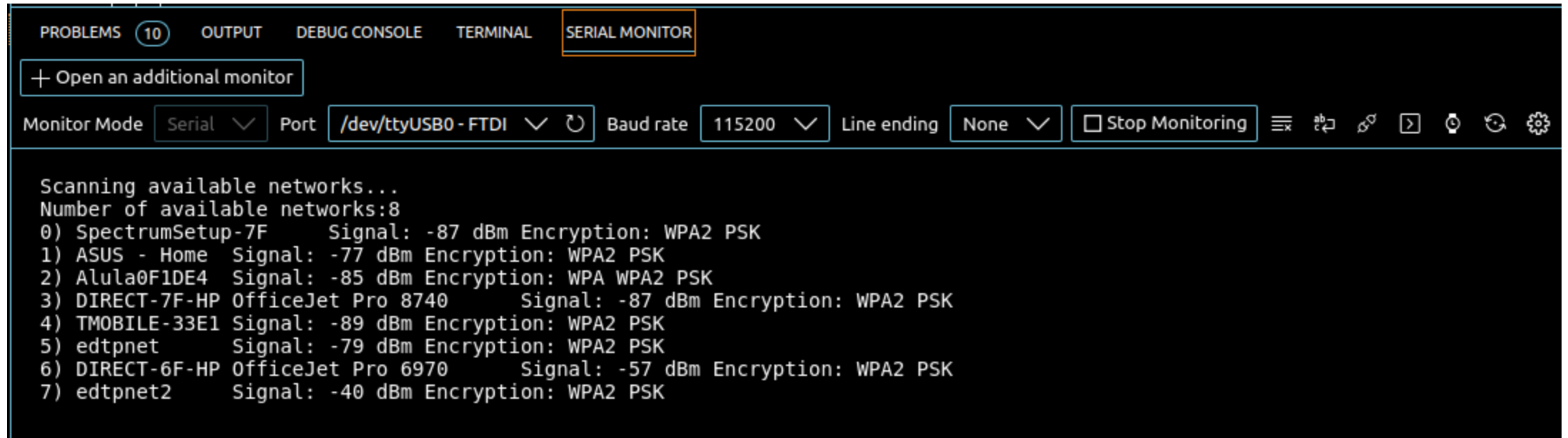
Ln 6, Col 28 Spaces: 2 UTF-8 LF {} C++ <Select Programmer> wizFi360\_connect.ino Raspberry Pi Pico /dev/ttyUSB0 Arduino



# Run the Sketch



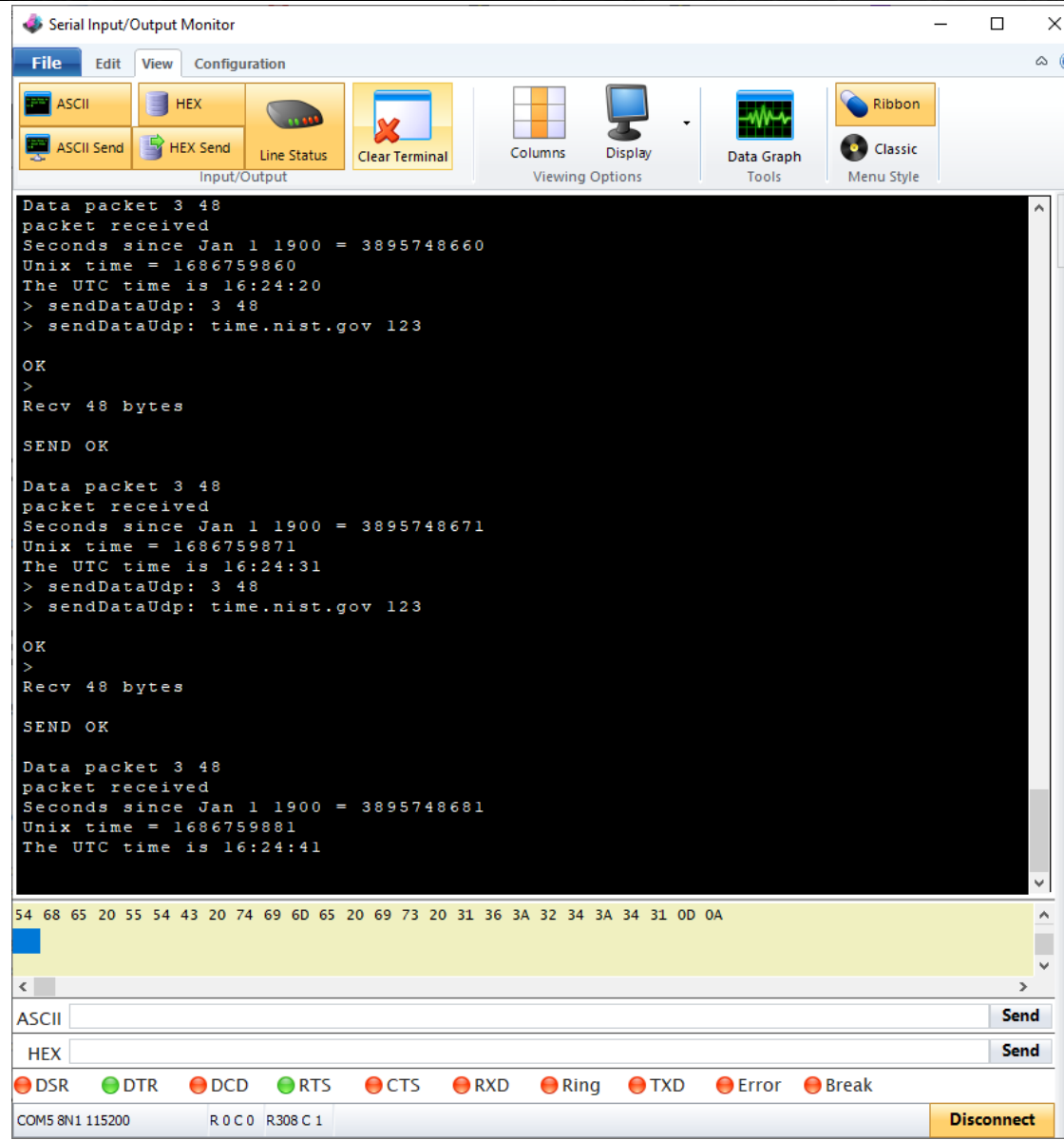
## Network Scan



The screenshot shows the Serial Monitor window in an IDE. The 'SERIAL MONITOR' tab is active. The interface includes a toolbar with a '+ Open an additional monitor' button, a 'Monitor Mode' dropdown set to 'Serial', a 'Port' dropdown set to '/dev/ttyUSB0 - FTDI', a 'Baud rate' dropdown set to '115200', and a 'Line ending' dropdown set to 'None'. There is also a 'Stop Monitoring' button and several utility icons. The main area displays the following text:

```
Scanning available networks...
Number of available networks:8
0) SpectrumSetup-7F      Signal: -87 dBm Encryption: WPA2 PSK
1) ASUS - Home           Signal: -77 dBm Encryption: WPA2 PSK
2) Alula0F1DE4           Signal: -85 dBm Encryption: WPA WPA2 PSK
3) DIRECT-7F-HP OfficeJet Pro 8740      Signal: -87 dBm Encryption: WPA2 PSK
4) TMOBILE-33E1          Signal: -89 dBm Encryption: WPA2 PSK
5) edtpnet               Signal: -79 dBm Encryption: WPA2 PSK
6) DIRECT-6F-HP OfficeJet Pro 6970      Signal: -57 dBm Encryption: WPA2 PSK
7) edtpnet2              Signal: -40 dBm Encryption: WPA2 PSK
```

## Network Time



The screenshot shows the Serial Input/Output Monitor window with the following content:

```
Serial Input/Output Monitor
File Edit View Configuration
ASCII HEX Line Status Clear Terminal Columns Display Data Graph Ribbon Classic
ASCII Send HEX Send Input/Output Viewing Options Tools Menu Style

Data packet 3 48
packet received
Seconds since Jan 1 1900 = 3895748660
Unix time = 1686759860
The UTC time is 16:24:20
> sendDataUdp: 3 48
> sendDataUdp: time.nist.gov 123

OK
>
Recv 48 bytes

SEND OK

Data packet 3 48
packet received
Seconds since Jan 1 1900 = 3895748671
Unix time = 1686759871
The UTC time is 16:24:31
> sendDataUdp: 3 48
> sendDataUdp: time.nist.gov 123

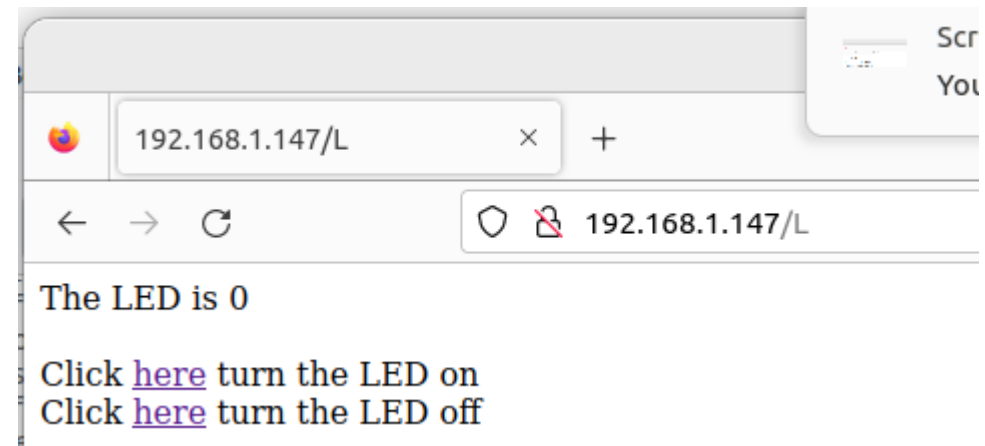
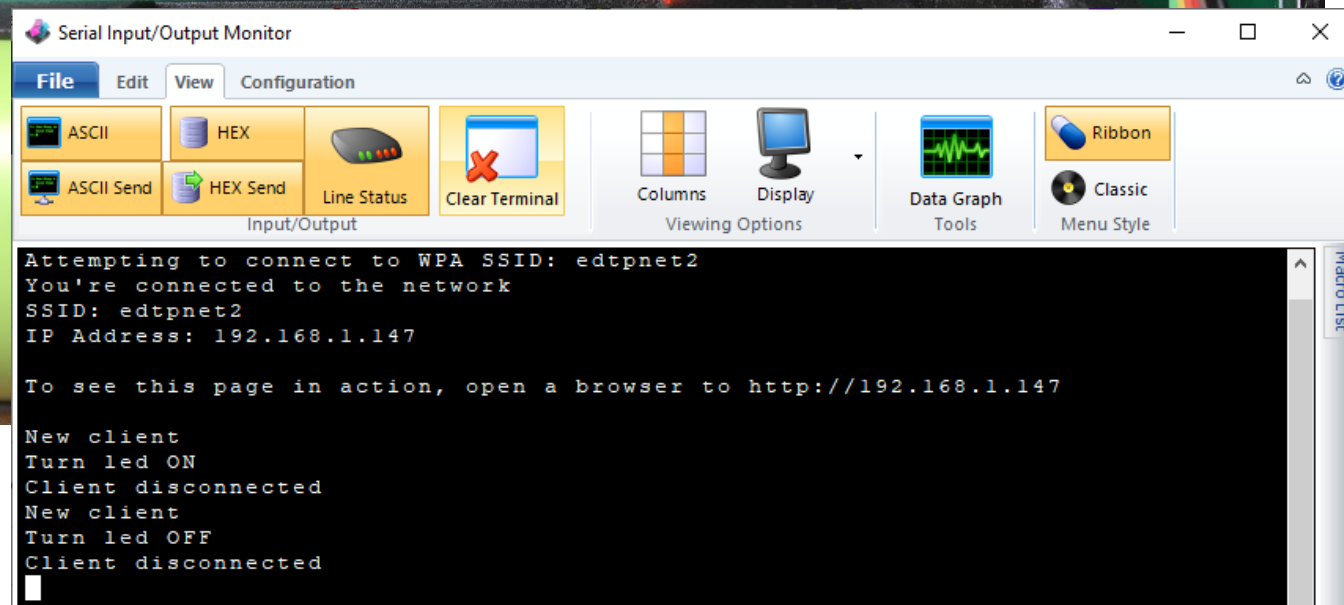
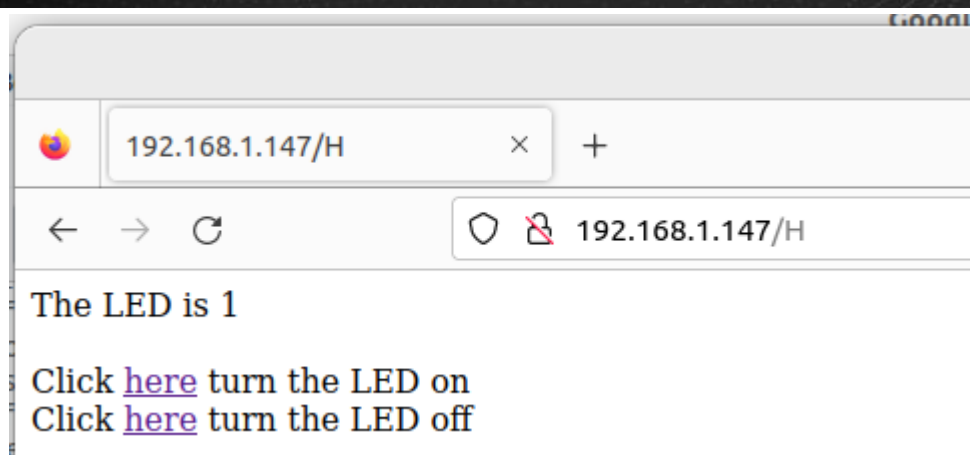
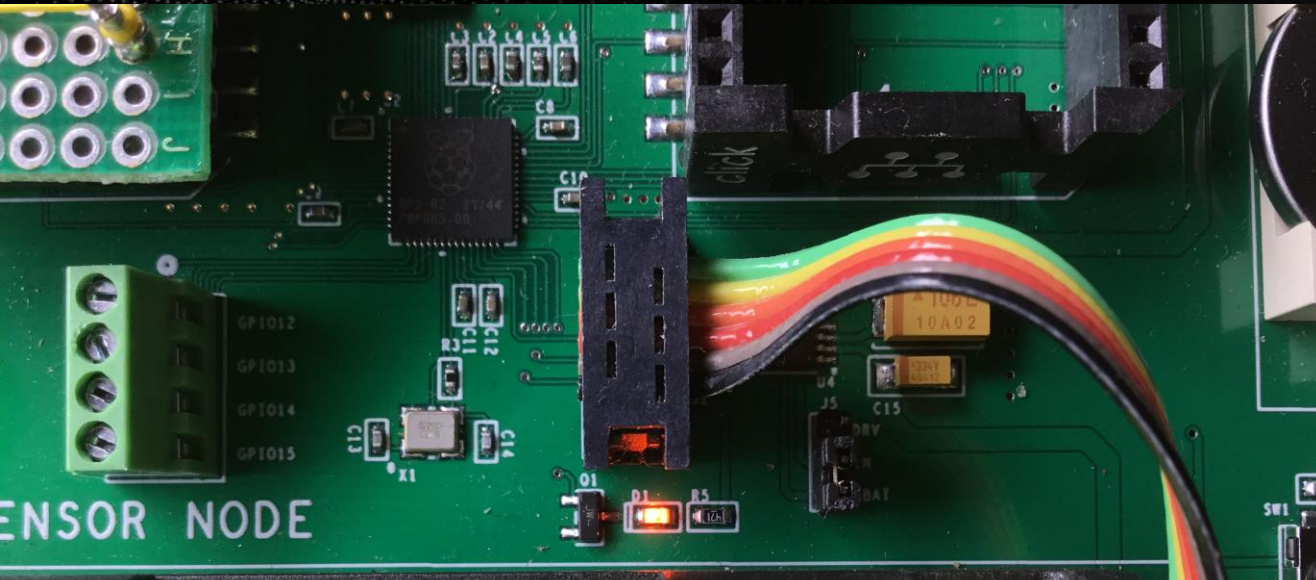
OK
>
Recv 48 bytes

SEND OK

Data packet 3 48
packet received
Seconds since Jan 1 1900 = 3895748681
Unix time = 1686759881
The UTC time is 16:24:41

54 68 65 20 55 54 43 20 74 69 6D 65 20 69 73 20 31 36 3A 32 34 3A 34 31 0D 0A
ASCII [ ] Send
HEX [ ] Send
DSR DTR DCD RTS CTS RXD Ring TXD Error Break
COM5 8N1 115200 R 0 C 0 R308 C 1 Disconnect
```

## LED Control

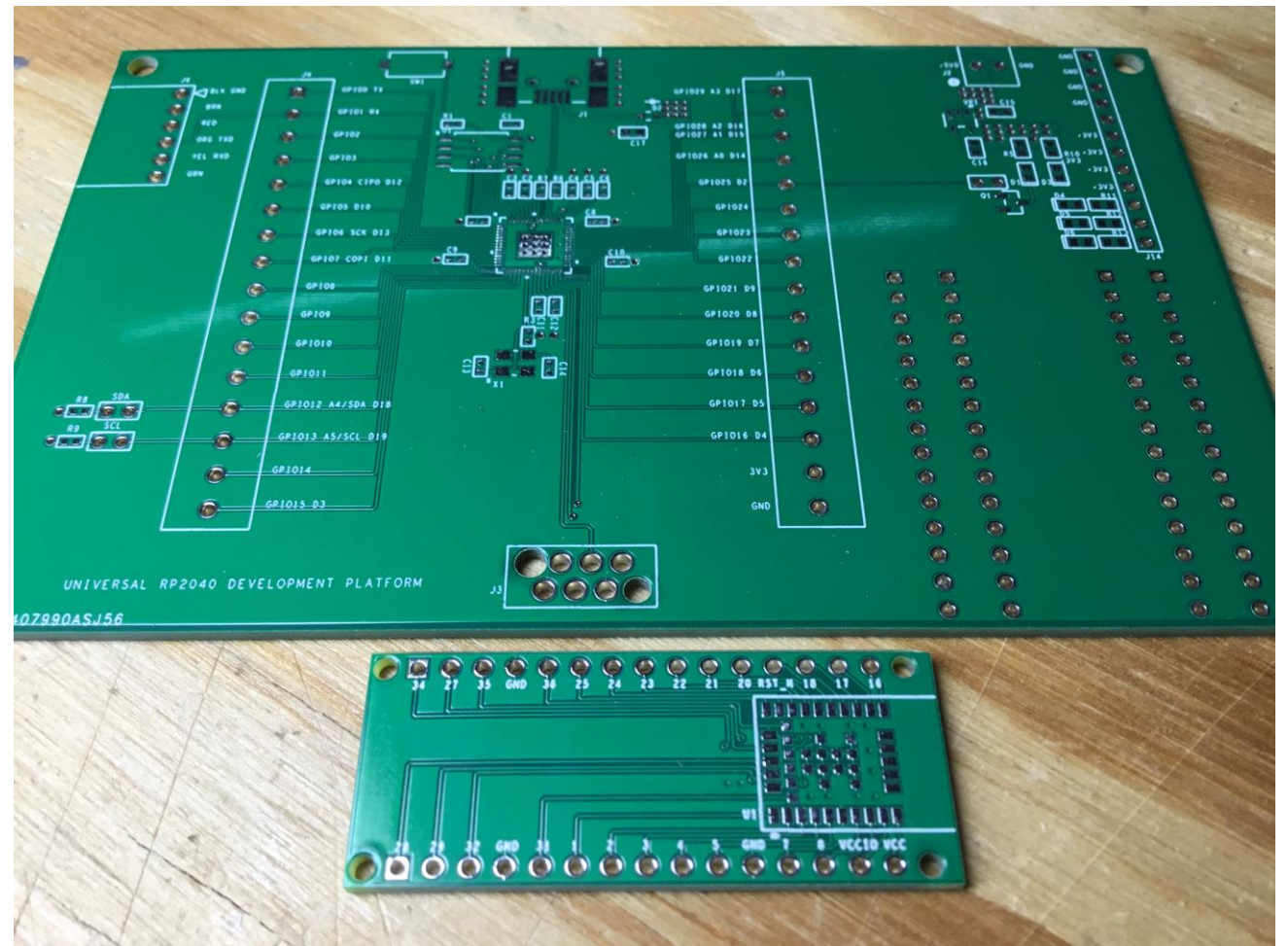


# Thank you for attending!!!

Please consider the resources below:

- [arduino.cc](https://arduino.cc)
- [wiznet.io](https://wiznet.io)
- [raspberrypi.org](https://raspberrypi.org)

MORE TO COME..





Thank You

Sponsored by

