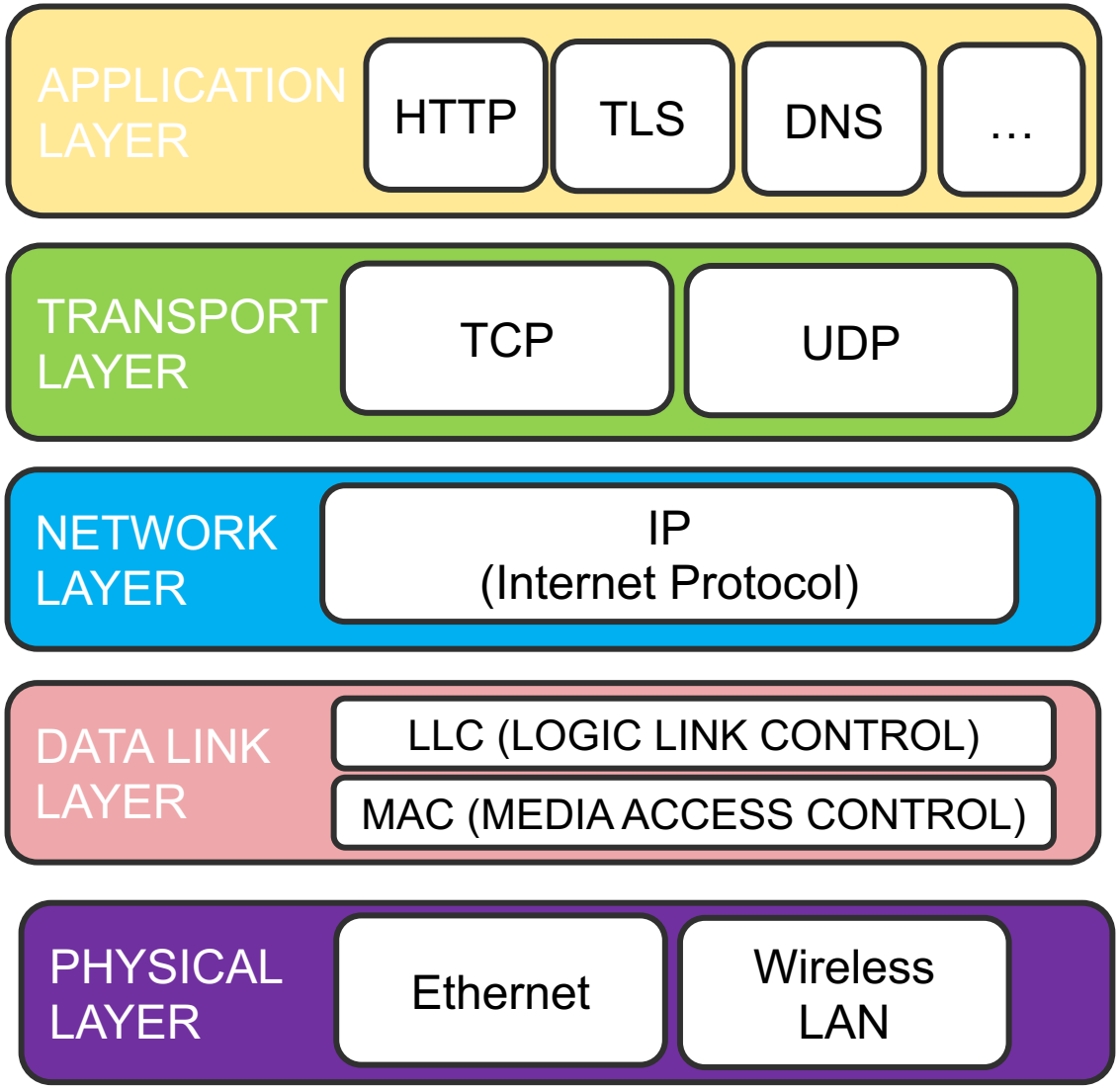
PAT 451/551 INTERACTIVE MEDIA DESIGN I

ARDUINO WIFI

INTERNET PROTOCOLS



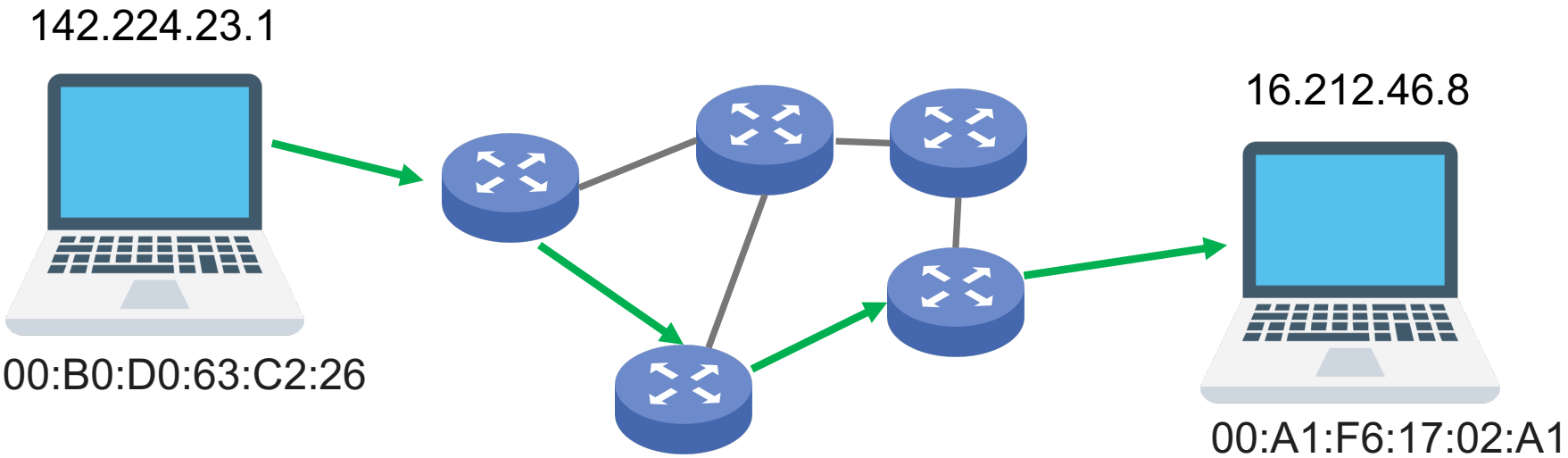
INTERNET ADDRESSING

IP Addresses	142.224.23.1	16.212.46.8	74.195.212.87
			
MAC Addresses	00:B0:D0:63:C2:26	00:A1:F6:17:02:A1	00:F8:A1:82:2A:15

IP Addresses: Typically assigned by the router (usually via DHCP)

MAC Addresses: “burned-in” – permanently attached to the hardware

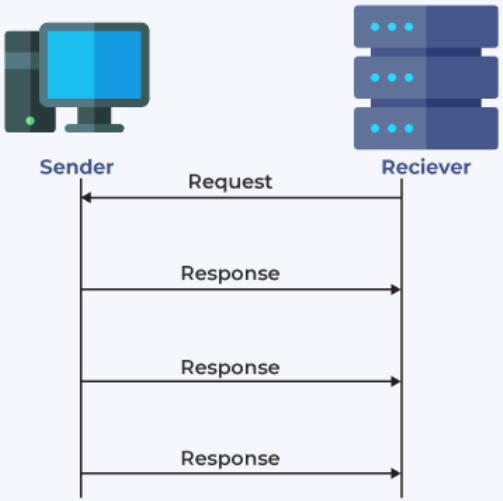
INTERNET ROUTING



Internet routing protocol

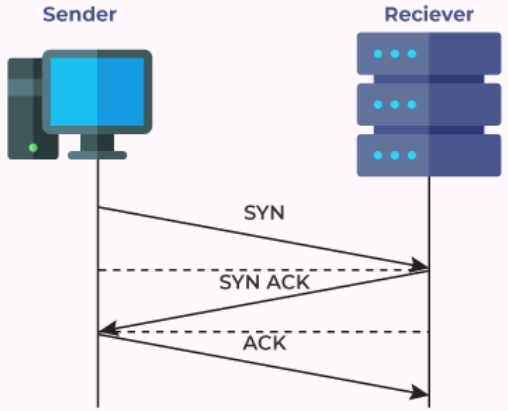
UDP: User Datagram Protocol

- No Connection
- Faster
- Less reliable, packets may be lost – “Best Effort”
- No error correction or acknowledgements
- Lightweight, less overhead
- Streaming, gaming, VOIP, DNS,
- Broadcast, Multicast



TCP: Transmission Control Protocol

- Establish a connection before data is delivered
- Guaranteed delivery, in order
- More overhead, requires ‘handshake’ and acknowledgment
- Slower
- Error correction
- HTTP, email, file transfer



OPENSOUND CONTROL (OSC)

"*a data transport specification (an encoding) for realtime message communication among applications and hardware…*

originally designed as a highly accurate, low latency, lightweight, and flexible method of communication for use in realtime musical performance "

In other words: A way to package data for timely communication between devices. An alternative to MIDI.

Some references:
https://itp.nyu.edu/networks/explanations/open-sound-control/
https://ccrma.stanford.edu/groups/osc/index.html

OPENSOUND CONTROL (OSC)

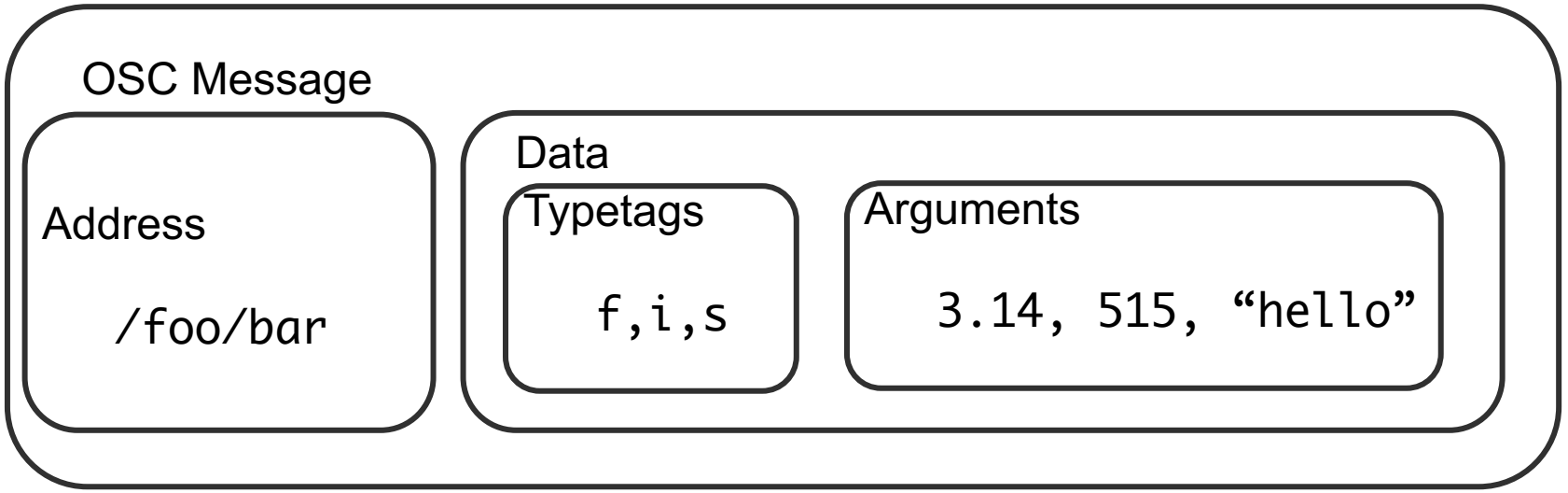
Key features

- Real-time message communication
- Human-readable
- User-defined hierarchical namespace
- Lightweight
  - minimal overhead
- Flexible
  - arbitrary names, data types, message size
- Transport-independent
  - Can be sent over UDP, TCP, Serial, etc.

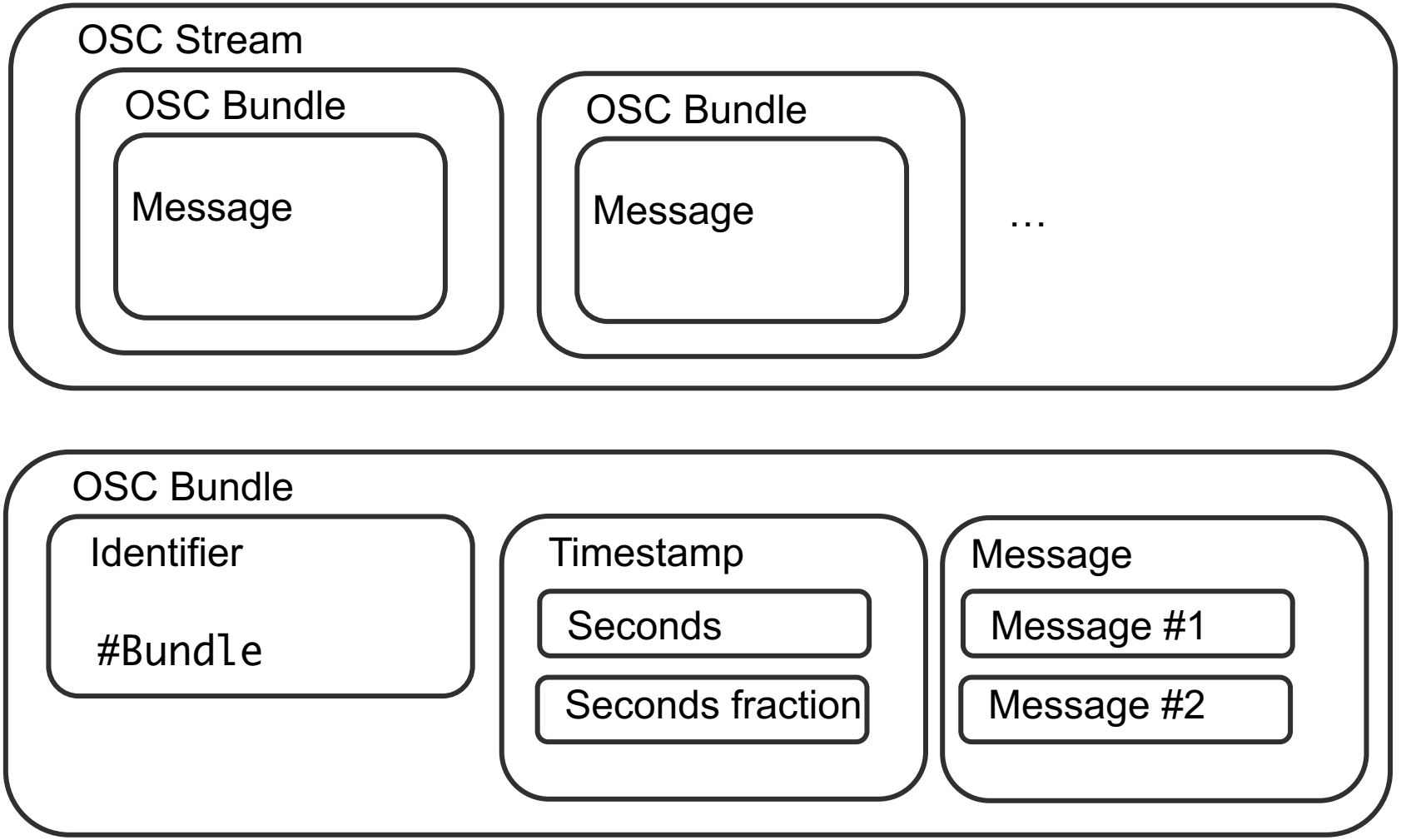
OSC MESSAGES

`/this/is/an/arbitrary/message` `3.14 515 “hello”`

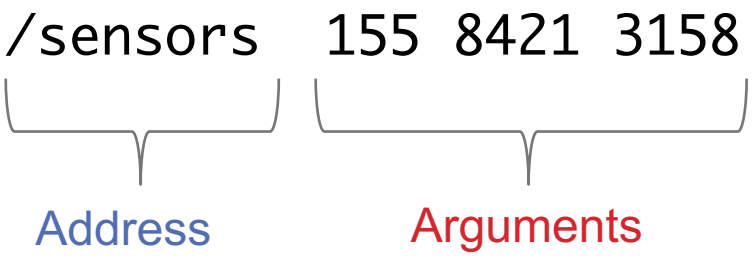
Address — Arguments



OSC BUNDLES


OSC Stream
OSC Bundle
Message
OSC Bundle
Message
…
OSC Bundle
Identifier
#Bundle
Timestamp
Seconds
Seconds fraction
Message
Message #1
Message #2

OSC MESSAGE EXAMPLES



Other possibilities

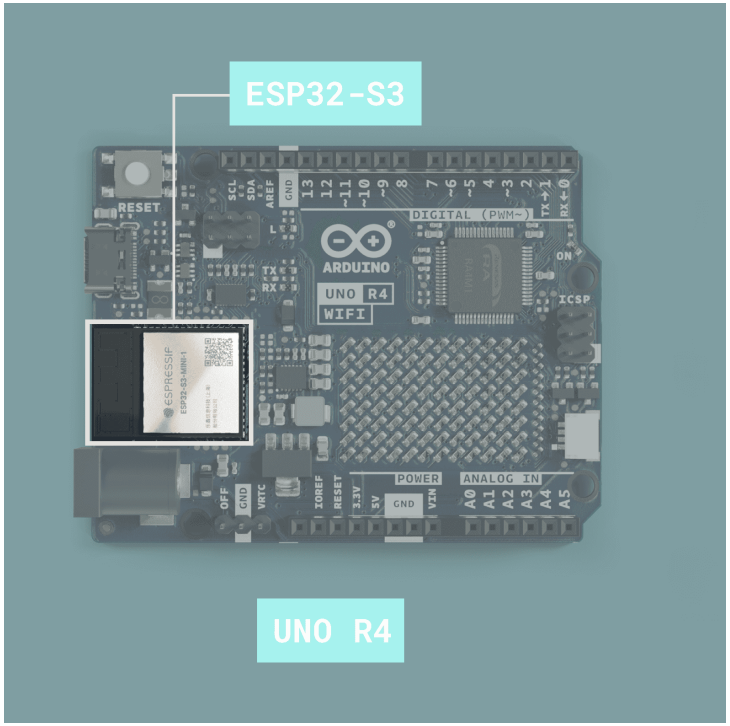
```
/sensor/fsr  155
/sensor/flex  8421
/sensor/pot 3158
/sensor/tof 60.54

/button/1 “on”
/button/2 “off”
```

```
/synth/saw/freq  440.0
/synth/saw/amp  0.9
```

ARDUINO WIFI

- Arduino has a second, ESP32-S3 microcontroller on board.
- ESP32 has Wi-Fi and Bluetooth capabilities
- It can be programmed directly, but isn't generally necessary
- More info here: https://docs.arduino.cc/tutorials/uno-r4-wifi/cheat-sheet#esp32-s3-mini-1-n8
- We can access the WiFi features using built-in WiFiS3 library



ARDUINO ← → MAX OSC

- Install OSC library (by Adrian Freed) via Arduino Package Manager
- Include WiFiS3 and OSCMessage libraries

```
#include <WiFiS3.h>
#include <OSCMessage.h>
```

- arduino_secrets.h file contains a dedicated wifi network and password in Davis

```
#include "arduino_secrets.h"
```

- If you are working at home, you can change these to your home network. You don't need to include the file in assignment submissions.
- Depending on your home network security settings, UDP data may be blocked

IP ADDRESSES

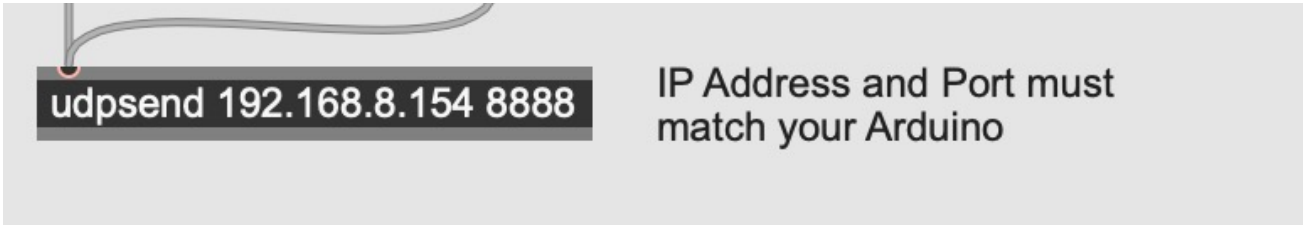
- Our `arduinowifi` WiFi router assigns your Arduino a dynamic IP address via DHCP
- It is NOT connected to the internet, so you must connect your computer to that network as well.
- Your Iarduino's P address will probably be something like 192.168.8.154
- The last 3 digits are most likely to change.
- Our example program prints the IP address to the Serial Port once it successfully connects to the router

UDP PORTS

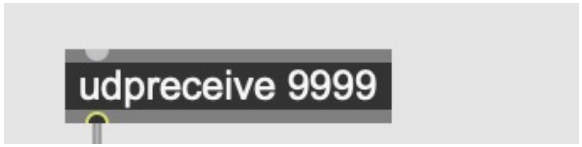
- UDP requires a **port** number
  - A number embedded in outgoing messages that helps get data to the right applications
  - Application protocols use designated ports
  - How your Spotify streams don't end up in your web browser
  - Sender sends to a destination port; receiver listens for messages on that port
- In Arduino, set `localPort` to the port you want to receive data FROM your computer
- In Arduino, set `remotePort` to the port number you will listen on in Max

ARDUINO ← → MAX OSC

- Use udpsend object with IP address of your Arduino and the port number you assigned to `localPort`



- Use udpreceive with the port number you assigned to `remotePort`



- Arduino program can get your computer's IP address when you first send it data

THERE IS A LOT MORE YOU CAN DO WITH YOUR ARDUINO'S WIFI!

See:

https://docs.arduino.cc/tutorials/uno-r4-wifi/wifi-examples

https://github.com/arduino/ArduinoCore-renesas/tree/main/libraries/WiFiS3/examples