

リストA Cargo.toml

```
[dependencies]
esp-idf-sys = { version = "=0.32", features = ["binstart"] }
esp-idf-svc = { version="=0.45", features = ["experimental", "alloc"] }
embedded-svc = "0.24"
esp-idf-hal = "0.40.1"
log = "0.4"
anyhow = "1"
toml-cfg = "0.1"
```

リストB 設定ファイルcfg.toml

```
[wifi]
wifi_ssid = ""
wifi_psk = ""
```

リストC ビルド時のログ (抜粋) と解説

```
I (102) boot_comm: chip revision: 3, min. application chiESP-ROM:esp32c3-ap11-20210207
Build:Feb  7 2021
rst:0x15 (USB_UART_CHIP_RESET),boot:0xc (SPI_FAST_FLASH_BOOT)
```

中略

```
I (383) cpu_start: Starting scheduler.
I (394) pp: pp rom version: 9387209
# ESP-IDFのWi-Fiスタック初期化が始まります
I (394) net80211: net80211 rom version: 9387209
I (404) wifi:wifi driver task: 3fc9de64, prio:23, stack:6656, core=0
I (404) system_api: Base MAC address is not set
I (414) system_api: read default base MAC address from EFUSE
```

中略

Wi-Fiスタックの初期化が完了し、アクセス・ポイントのスキャンをします

```
I (504) esp_idf_svc::wifi: Driver initialized
I (504) esp_idf_svc::wifi: Event handlers registered
I (514) esp_idf_svc::wifi: Initialization complete
I (514) wifi::wifi: Wifi created, about to scan
I (524) esp_idf_svc::wifi: About to scan for access points
I (534) esp_idf_svc::wifi: Stopping
I (534) esp_idf_svc::wifi: Stop requested
I (534) esp_idf_svc::wifi: About to wait for status
I (544) esp_idf_svc::wifi: Waiting for status done - success
I (554) esp_idf_svc::wifi: Stopped
```

中略

見つかったアクセス・ポイント一覧が表示されます

```
I (2714) esp_idf_svc::wifi: Got wifi event: ScanDone
I (2714) esp_idf_svc::wifi: STA event ScanDone skipped
I (2764) esp_idf_svc::wifi: Found 16 access points
I (2764) esp_idf_svc::wifi: About to get info for found access points
I (2764) esp_idf_svc::wifi: Got info for 16 access points
```

アクセス・ポイント一覧

中略

ステーション・モードとして動き始めます

```
I (3174) esp_idf_svc::wifi: Wifi mode STA set
```

```
# 中略
```

```
# アクセス・ポイントに接続してDHCPでIPアドレスが割り振られています
```

```
I (3404) esp_idf_svc::wifi: Got wifi event: StaConnected
```

```
I (3404) esp_idf_svc::wifi: STA event StaConnected handled, set status: Status(Started(Connected(Waiting)), Stopped)
```

```
I (7394) esp_idf_svc::wifi: Got STA IP event: DhcpIpAssigned(DhcpIpAssignment { netif_handle: 0x3fca33bc, ip_settings: ClientSettings { ip: 192.168.100.34, subnet: Subnet { gateway: 192.168.100.1, mask: Mask(24) }, dns: None, secondary_dns: None }, ip_changed: true })
```

```
I (7404) esp_idf_svc::wifi: STA IP event DhcpIpAssigned(DhcpIpAssignment { netif_handle: 0x3fca33bc, ip_settings: ClientSettings { ip: 192.168.100.34, subnet: Subnet { gateway: 192.168.100.1, mask: Mask(24) }, dns: None, secondary_dns: None }, ip_changed: true }) handled, set status:
```

```
Status(Started(Connected(Done(ClientSettings { ip: 192.168.100.34, subnet: Subnet { gateway: 192.168.100.1, mask: Mask(24) }, dns: None, secondary_dns: None }))), Stopped)
```

```
I (7444) esp_netif_handlers: sta ip: 192.168.100.34, mask: 255.255.255.0, gw: 192.168.100.1
```

```
I (7454) esp_idf_svc::wifi: Waiting for status done - success
```

```
I (7464) esp_idf_svc::wifi: Providing status: Status(Started(Connected(Done(ClientSettings { ip: 192.168.100.34, subnet: Subnet { gateway: 192.168.100.1, mask: Mask(24) }, dns: None, secondary_dns: None }))), Stopped)
```

```
# 無事, アクセス・ポイントに接続できました
```

```
I (7484) wifi::wifi: Wifi connected
```

図A Wi-Fi接続成功のログ

```
info!("Wifi connected");
```

リストD 設定ファイルcfg.toml

```
toml
[std-net]
wifi_ssid = ""
wifi_psk = ""
http_server = "ホストPCのIPアドレス:8000"
```

図B simple-http-serverのインストールの確認

```
$ simple-http-server -h
Simple HTTP(s) Server 0.6.3
```

```
USAGE:
```

```
simple-http-server [FLAGS] [OPTIONS] [--] [root]
```

```
FLAGS:
```

```
--coep      Add "Cross-Origin-Embedder-Policy" HTTP header and set it to "require-corp"
--coop      Add "Cross-Origin-Opener-Policy" HTTP header and set it to "same-origin"
```

```
# 以下略...
```

図C HTTPサーバを起動する

```
# std-net/appディレクトリ
```

```
$ simple-http-server -i
```

```
Index: enabled, Cache: enabled, Cors: disabled, Coop: disabled, Coep: disabled, Range: enabled, Sort: enabled, Threads: 3
```

```
Upload: disabled, CSRF Token:
```

```
Auth: disabled, Compression: disabled
```

```
https: disabled, Cert: , Cert-Password:
```

```
TryFile404:
```

```
Address: http://0.0.0.0:8000
```

リストE index.html

```
html
<!DOCTYPE html>
<html>
  <body>
    Hello, ESP32-C3!
  </body>
</html>
```

図D プログラムの書き込みからHTTPサーバから受信までのログ

```
$ cargo espflash --monitor
# 中略...
I (7988) std_net::wifi: Wifi connected
I (9968) std_net: response:
HTTP/1.1 200 OK
Accept-Ranges: bytes
Content-Type: text/html
Content-Length: 79
Cache-Control: public, max-age=604800
Last-Modified: Sun, 27 Nov 2022 21:28:49 GMT
ETag: W/"4f-6383d691.0"
Date: Tue, 29 Nov 2022 03:44:39 GMT

<!DOCTYPE html>
<html>
  <body>
    Hello, ESP32-C3!
  </body>
</html>
```

リストF 必要な情報をCargo.tomlに追加する

```
attohttpc = { version="0.24", default-features = false, features = ["json"] }
serde_json = "1.0"
```

リストG partitions.csv

```
# Name, Type, SubType, Offset, Size, Flags
# Note: if you have increased the bootloader size, make sure to update the offsets to avoid overlap
nvs, data, nvs, , 0x6000,
phy_init, data, phy, , 0x1000,
factory, app, factory, , 3M,
```

リストH Cargo.tomlでパーティション・テーブルを書き込む設定

```
[package.metadata.espflash]
partition_table = "partitions.csv"
```