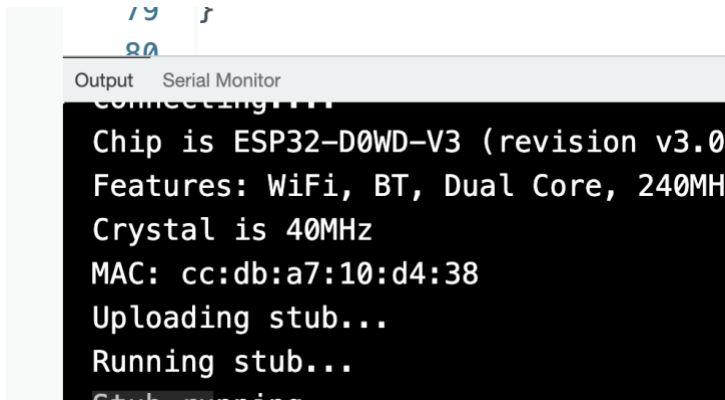


CSC 5930 9030 Spring 2025 Week 5 Assignment

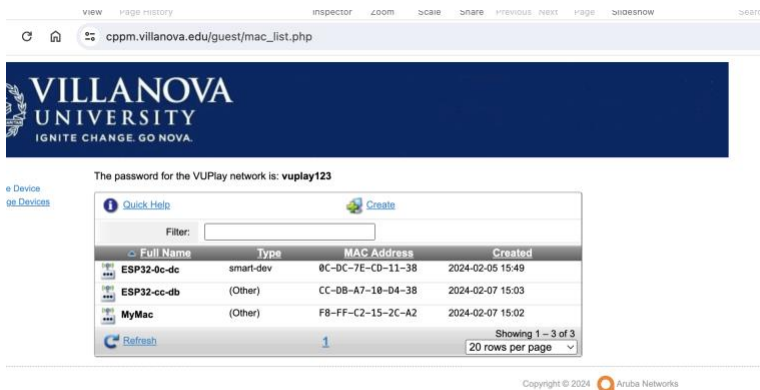
VUPlay registration. WiFi Examples completed. Submit screenshots showing items from assignment completed:

VUPlay registration:

EEK MAC address value

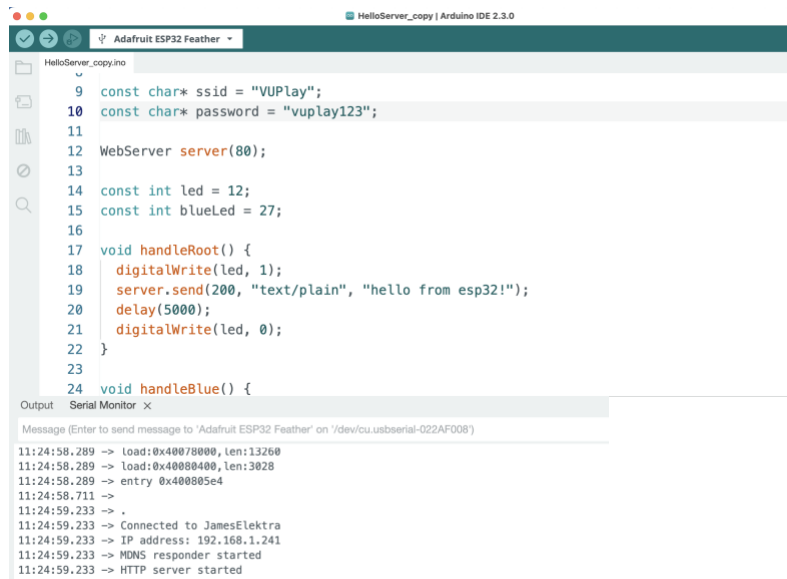


ESP32 registered



WiFi Examples:

Hello Web Server in IDE



```
9  const char* ssid = "VUPlay";
10 const char* password = "vuplay123";
11
12 WebServer server(80);
13
14 const int led = 12;
15 const int blueLed = 27;
16
17 void handleRoot() {
18   digitalWrite(led, 1);
19   server.send(200, "text/plain", "hello from esp32!");
20   delay(5000);
21   digitalWrite(led, 0);
22 }
23
24 void handleBlue() {
```

Output Serial Monitor x

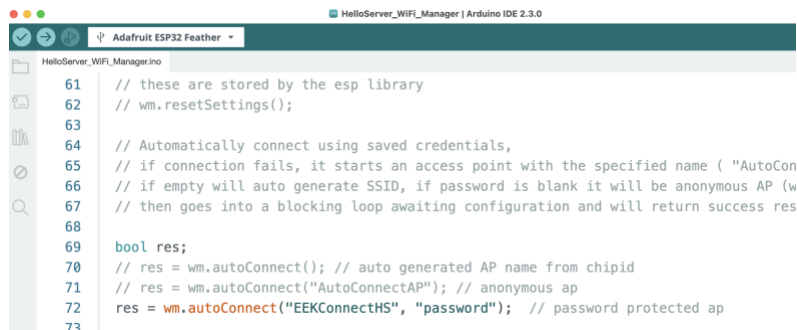
Message (Enter to send message to 'Adafruit ESP32 Feather' on '/dev/cu.usbserial-022AF008')

```
11:24:58.289 -> Load:0x40078000, len:13260
11:24:58.289 -> Load:0x40080400, len:3028
11:24:58.289 -> entry 0x400805e4
11:24:58.711 -> .
11:24:59.233 -> .
11:24:59.233 -> Connected to JamesElektra
11:24:59.233 -> IP address: 192.168.1.241
11:24:59.233 -> MDNS responder started
11:24:59.233 -> HTTP server started
```

Hello Web Server URL in Browser



WiFi Manager Hello Server in IDE



```
61 // these are stored by the esp library
62 // wm.resetSettings();
63
64 // Automatically connect using saved credentials,
65 // if connection fails, it starts an access point with the specified name ( "AutoCon
66 // if empty will auto generate SSID, if password is blank it will be anonymous AP (w
67 // then goes into a blocking loop awaiting configuration and will return success res
68
69 bool res;
70 // res = wm.autoConnect(); // auto generated AP name from chipid
71 // res = wm.autoConnect("AutoConnectAP"); // anonymous ap
72 res = wm.autoConnect("EEKConnectHS", "password"); // password protected ap
73
```

WiFi Manager Hello Server Serial Monitor

```
66 // An empty wait will generate 500ms, if password is blank it will be done
67 // then goes into a blocking loop awaiting configuration and will return
68
69 bool res;
70 // res = wm.autoConnect(); // auto generated AP name from chipid
71 // res = wm.autoConnect("AutoConnectAP"); // anonymous ap
72 res = wm.autoConnect("EEKConnectHS", "password"); // password protected
```

Output Serial Monitor x

Message (Enter to send message to 'Adafruit ESP32 Feather' on '/dev/cu.usbserial-022AF008')

```
11:30:10.211 -> entry 0x400005e4
11:30:10.669 -> *wm:AutoConnect
11:30:10.669 -> *wm:Connecting to SAVED AP: JamesElektra
11:30:11.158 -> *wm:connectTimeout not set, ESP waitForConnectResult...
11:30:11.288 -> *wm:AutoConnect: SUCCESS
11:30:11.288 -> *wm:STA IP Address: 192.168.1.241
11:30:11.288 -> connected...yeey :)
11:30:11.288 -> MDNS responder started
11:30:11.288 -> HTTP server started
```

SimpleTime with OLED and WM in IDE

SimpleTime_OLED_WM | Arduino IDE 2.3.0

SimpleTime_OLED_WM.ino

```
97 * A more convenient approach to handle timezones with daylightwiset
98 * would be to specify a environmnet variable with TimeZone definition including daylight adj
99 * A list of rules for your zone could be obtained from https://github.com/esp8266/Arduino/blob/master/libraries/Time/keywords.txt
100 */
101 configTzTime(time_zone, ntpServer1, ntpServer2);
102
103 //connect to WiFi
104 //WiFiManager, Local initialization. Once its business is done, there is no need to keep it around
105 WiFiManager wm;
106 // reset settings - wipe stored credentials for testing
107 // these are stored by the esp library
108 // wm.resetSettings();
```

Output Serial Monitor x

Message (Enter to send message to 'Adafruit ESP32 Feather' on '/dev/cu.usbserial-022AF008')

```
11:35:04.587 -> *wm:Connecting to SAVED AP: JamesElektra
11:35:05.079 -> *wm:connectTimeout not set, ESP waitForConnectResult...
11:35:05.176 -> *wm:AutoConnect: SUCCESS
11:35:05.176 -> *wm:STA IP Address: 192.168.1.241
11:35:05.176 -> connected...yeey :)
11:35:09.556 -> Got time adjustment from NTP!
11:35:09.602 -> Tuesday, February 13 2024 11:35:09
11:35:10.210 -> Tuesday, February 13 2024 11:35:10
11:35:15.245 -> Tuesday, February 13 2024 11:35:15
```

OLED Photo SimpleTime with OLED and WM



IDE of UDP Server with OLED and LEDs

```
UDP_Server_OLED_LEDs | Arduino IDE 2.3.0
Adafruit ESP32 Feather
UDP_Server_OLED_LEDs.ino
16
17 char packetBuffer[255];
18
19 unsigned int localPort = 8889;
20
21 const char *ssid = "EEK-CCDB";
22 const char *password = "EEK-TEST";
23
24 Adafruit_SSD1306 display = Adafruit_SSD1306(128, 32, &Wire);
25
26 void toggle_led(int ledToToggle) {
27 // Toggle the state of the LED pin (write the NOT of the current state to the LED pin)
```

Serial Monitor output UDP Server with OLED and LEDs

```
11 #define Bottom_Red 26
12 #define Yellow 25
```

Output Serial Monitor x

Message (Enter to send message to 'Adafruit ESP32 Feather' on '/dev/cu.usbserial-022AF008')

```
11:40:26.970 -> mode:DI0, clock div:1
11:40:26.970 -> load:0x3fff0030, len:1184
11:40:26.970 -> load:0x40078000, len:13260
11:40:26.970 -> load:0x40080400, len:3028
11:40:26.970 -> entry 0x400805e4
11:40:27.384 ->
11:40:27.384 -> Serial Connected
11:40:29.391 -> ESP Board MAC Address: CC:DB:A7:10:D4:38
11:40:29.391 -> Soft-AP IP address = 192.168.4.1
```

Wed Feb 12 10:46

Wi-Fi ☒


Personal Hotspot

iPhone14Pro 5G

Known Networks

- Arlington
- ESP32_APBAB4
- JamesElektra

Dual AP STA Password Prompt



The Wi-Fi network "ESP32_APBAB4" requires a WPA2 password.

You can also access this Wi-Fi network by sharing the password from a nearby iPhone, iPad, or Mac which has connected to this network and has you in their contacts.

Password:

☒ Show password

?CancelJoin

Dual AP STA Serial Monitor

```
18 Serial. // In Print
19 WiFi.softAPConfig(IP_ADDR, IP_ADDR, IP_ADDR);
20 Serial.print("[+] AP Created with IP Gateway ");
21 Serial.println(WiFi.softAPIP()); /*Printing the AP IP address*/
22
23 }
24 void loop() {}
```

Output Serial Monitor X

Message (Enter to send message to 'Adafruit ESP32 Feather' on '/dev/cu.usbserial-02650333')

10:44:26.620 -> configip: 0, SPIWP:0xee
10:44:26.620 -> clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
10:44:26.620 -> mode:DIO, clock div:1
10:44:26.620 -> load:0x3fff0030,len:4688
10:44:26.621 -> load:0x40078000,len:15460
10:44:26.621 -> ho 0 tail 12 room 4
10:44:26.621 -> load:0x40080400,len:4
10:44:26.621 -> load:0x40080404,len:3196
10:44:26.621 -> entry 0x400805a4
10:44:27.033 ->
10:44:27.033 -> [*] Connecting to WiFi Network
10:44:27.033 ->
10:44:33.342 -> [+] Connected to WiFi network with local IP : 192.168.1.177
10:44:33.342 ->
10:44:33.342 -> [*] Creating ESP32 AP
10:44:33.342 -> [+] AP Created with IP Gateway 192.168.4.1

PacketSender view for UDP Server with OLED and LEDs

Packet Sender - IPs: 192.168.1.221, 192.168.2.1, 192.168.135.1, 192.168.4.2, fe80::1%lo0, fe80::aede:48ff:fe00:1122%en25, fe80:...

Name APHello

ASCII hello

HEX 68 65 6c 6c 6f

Address 192.168.4.1 Port 8889 Resend Delay 0 ☒ UDP

Search Saved Packets... ☐ Persistent TCP

| | Send | Name | Resend | To Address | To Port | Method | ASCII | Hex |
|---|--|---------|--------|-------------|---------|--------|-------|----------------|
| 1 | <input checked="" type="button" value="Send"/> | APblue | 0 | 192.168.4.1 | 8889 | UDP | blue | 62 6c 75 65 |
| 2 | <input checked="" type="button" value="Send"/> | APHello | 0 | 192.168.4.1 | 8889 | UDP | hello | 68 65 6c 6c 6f |
| 3 | <input checked="" type="button" value="Send"/> | APtime | 0 | 192.168.4.1 | 8889 | UDP | time | 74 69 6d 65 |

IDE with SimpleTime with Dual WiFi modes

```
SimpleTime_copy_APsta.ino
104 int len = Udp.read(packetBuffer, 255);
105 if (len > 0) packetBuffer[len] = 0;
106 Serial.print("Received(IP/Size/Data): ");
107 Serial.print(Udp.remoteIP());
108 Serial.print(" / ");
109 Serial.print(packetSize);
110 Serial.print(" / ");
111 Serial.println(packetBuffer);
112
113 String command = String((char*)packetBuffer);
114 if (command.indexOf("time") >= 0) {
115     printLocalTime();
116 }
117
118 Udp.beginPacket(Udp.remoteIP(), Udp.remotePort());
119 Udp.printf("received: ");
120 Udp.printf(packetBuffer);
121 Udp.endPacket();
122 }

76 //connect to WiFi AP
77 WiFi.mode(WIFI_AP_STA); /*ESP32 Access point configured*/
78 Serial.println("\n[*] Creating ESP32 AP");
79 WiFi.softAP(APssid, APpassword); /*Configuring ESP32 access point SSID and password*/
80 Serial.print("[+] AP Created with IP Gateway ");
81 Serial.println(WiFi.softAPIP()); /*Printing the AP IP address*/
82
```

Serial Monitor of SimpleTime with Dual WiFi modes

Output Serial Monitor X

Message (Enter to send message to 'Adafruit ESP32 Feather' on '/dev/cu.usbserial-022AF008')

```
11:55:51.508 -> [*] Creating ESP32 AP
11:55:51.508 -> [+] AP Created with IP Gateway 192.168.4.1
11:55:51.508 -> Connecting to JamesElektra . CONNECTED
11:55:52.023 -> ESP Board MAC Address: CC:DB:A7:10:D4:38
11:55:52.023 -> Soft-AP IP address = 192.168.4.1
11:55:54.681 -> Got time adjustment from NTP!
11:55:54.681 -> Tuesday, February 13 2024 11:55:54
11:56:55.079 -> Received(IP/Size/Data): 192.168.4.5 / 4 / time
11:56:55.079 -> Tuesday, February 13 2024 11:56:55
```

Packet Sender for SimpleTime with Dual WiFi modes

Packet Sender - IPs: 192.168.1.221, 192.168.2.1, 192.168.135.1, 192.168.4.2, fe80::1%lo0, fe80::aede:48ff:fe00:1122%en25, fe80::...

Name

ASCII

HEX

Address Port Resend Delay

Search Saved Packets... ☐ Persistent TCP

| | Send | Name | Resend | To Address | To Port | Method | ASCII | Hex |
|---|-------------------------------------|---------|--------|--------------|---------|--------|---------|----------------------|
| 1 | <input type="button" value="Send"/> | APblue | 0 | 192.168.4.1 | 8889 | UDP | blue | 62 6c 75 65 |
| 2 | <input type="button" value="Send"/> | APhello | 0 | 192.168.4.1 | 8889 | UDP | hello | 68 65 6c 6c 6f |
| 3 | <input type="button" value="Send"/> | APtime | 0 | 192.168.4.1 | 8889 | UDP | time | 74 69 6d 65 |
| 4 | <input type="button" value="Send"/> | back | 0 | 192.168.10.1 | 8889 | UDP | back 40 | 62 61 63 6b 20 34 30 |

IDE with Full UDP Server and OLED with Dual WiFi

```
UDP_Server_OLED_LEDs_APsta_WM.ino
Adafruit ESP32 Feather
114 WiFiManager wm;
115 // reset settings - wipe stored credentials for testing
116 // these are stored by the esp library
117 // wm.resetSettings();
118 bool res;
119 // res = wm.autoConnect(); // auto generated AP name from chipid
120 // res = wm.autoConnect("AutoConnectAP"); // anonymous ap
121 res = wm.autoConnect(ssid, password); // password protected ap
122
123 if (!res) {
124   Serial.println("Failed to connect");
125   // ESP.restart();
126 } else {
127   //if you get here you have connected to the WiFi
128   Serial.println("connected...yeey :)");
129   Serial.print("\n[+] Connected to WiFi network with local IP : ");
130   Serial.println(WiFi.localIP()); /*Printing IP address of Connected network*/
131 }
132
133
```

Serial Monitor of Full UDP Server and OLED with Dual WiFi

```
13 #define Top_Red 12
14 #define Blue 27
15 #define Top_Green 33

Output Serial Monitor x
Message (Enter to send message to 'Adafruit ESP32 Feather' on '/dev/cu.usbserial-022AF008')
12:08:12.545 -> *wm:AutoConnect: SUCCESS
12:08:12.545 -> *wm:STA IP Address: 192.168.1.241
12:08:12.545 -> connected...yeey :)
12:08:12.545 ->
12:08:12.545 -> [+] Connected to WiFi network with local IP : 192.168.1.241
12:08:12.545 ->
12:08:12.545 -> [*] Creating ESP32 AP
12:08:12.545 -> [+] AP Created with IP Gateway 192.168.4.1
12:08:14.569 -> ESP Board MAC Address: CC:DB:A7:10:D4:38
12:08:14.569 -> Soft-AP IP address = 192.168.4.1
12:08:27.222 -> Got time adjustment from NTP!
12:08:27.263 -> Tuesday, February 13 2024 12:08:27
```


PacketSender for Full UDP Server and OLED with Dual WiFi

● ● ● Packet Sender - IPs: 192.168.1.221, 192.168.2.1, 192.168.135.1, 192.168.4.2, fe80::1%lo0, fe80::aede:48ff:fe00:1122%en25, fe80::...

Name APblue

ASCII blue

HEX 62 6c 75 65

Address 192.168.4.1

Port 8889

Resend Delay 0

UDP

Send

Save

Search Saved Packets...

Delete Saved Packet

Persistent TCP

| | Send | Name | Resend | To Address | To Port | Method | ASCII | Hex |
|---|------|---------|--------|-------------|---------|--------|-------|----------------|
| 1 | Send | APblue | 0 | 192.168.4.1 | 8889 | UDP | blue | 62 6c 75 65 |
| 2 | Send | APhello | 0 | 192.168.4.1 | 8889 | UDP | hello | 68 65 6c 6c 6f |
| 3 | Send | APtime | 0 | 192.168.4.1 | 8889 | UDP | time | 74 69 6d 65 |

Photo of EEK for Full UDP Server and OLED with Dual WiFi

