



DesignNews

Arduino Pro Primer

Day 4:

Networking with the Arduino Giga R1 WiFi and Portenta H7

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.

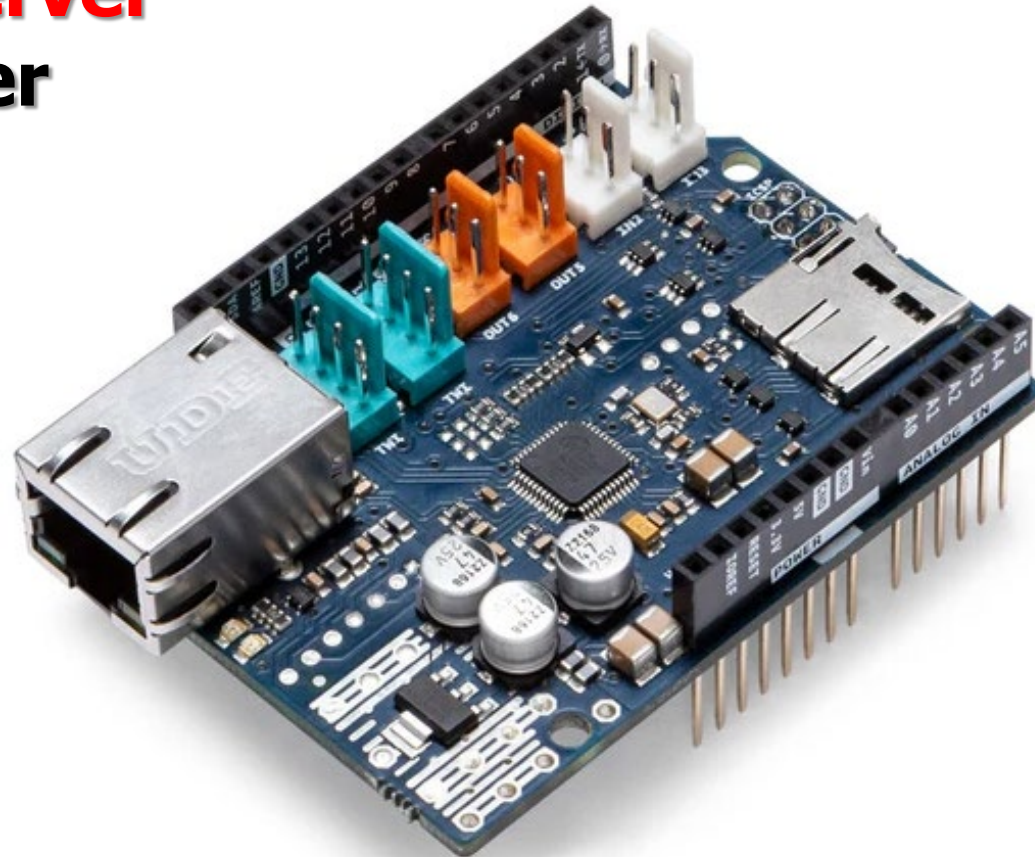


Fred Eady

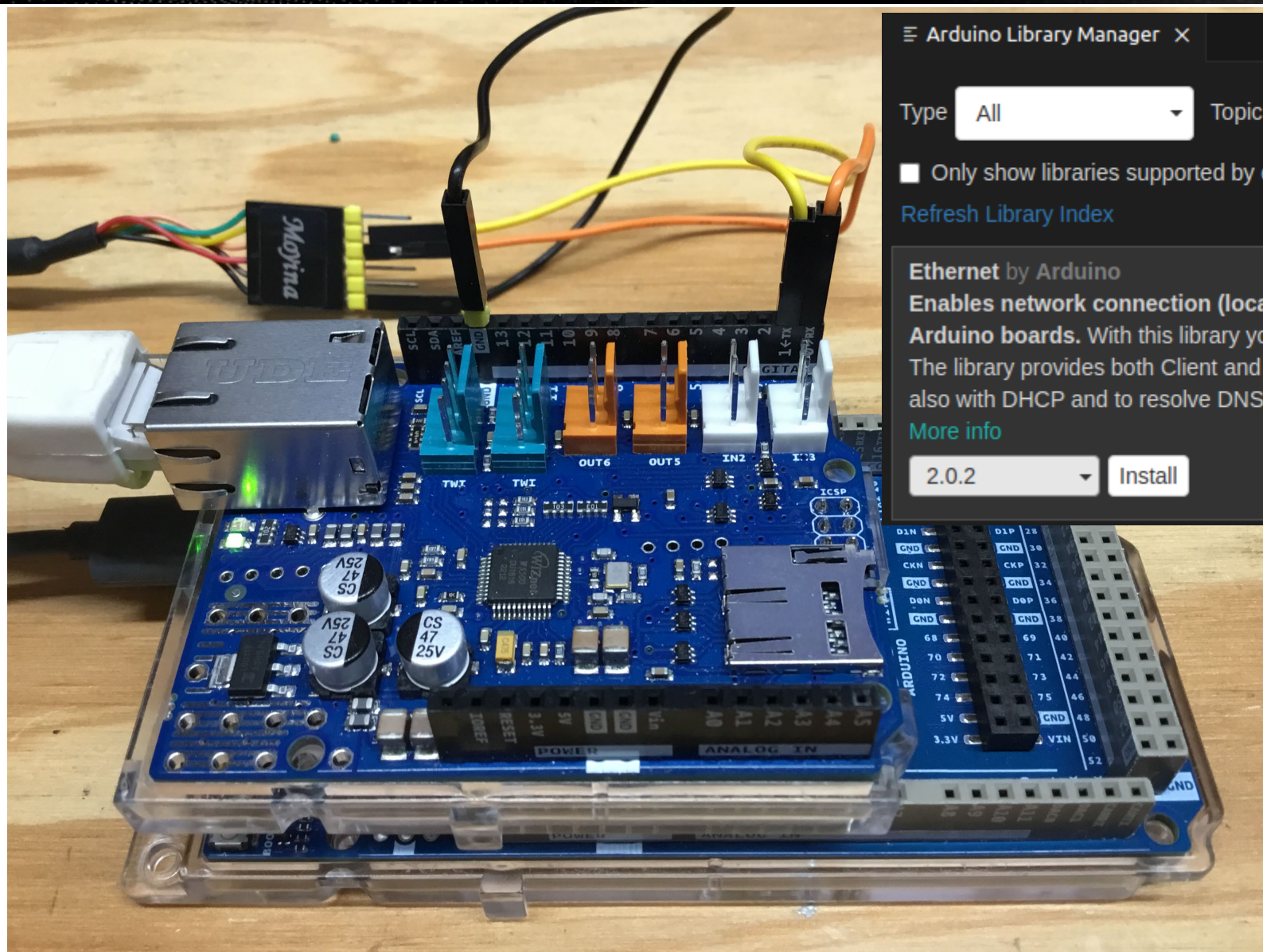
Visit 'Lecturer Profile' in your console for more details.

AGENDA

- **Ethernet Enable the Arduino Giga WiFi R1**
- **Ethernet Enable the Portenta H7**
- **Data and Command Transfer Via Ethernet**
- **TCP/IP Client-Server**
- **WiFi Client-Server**



Install the Ethernet Shield2 and Ethernet Library



☰ Arduino Library Manager ✕

Type Topic

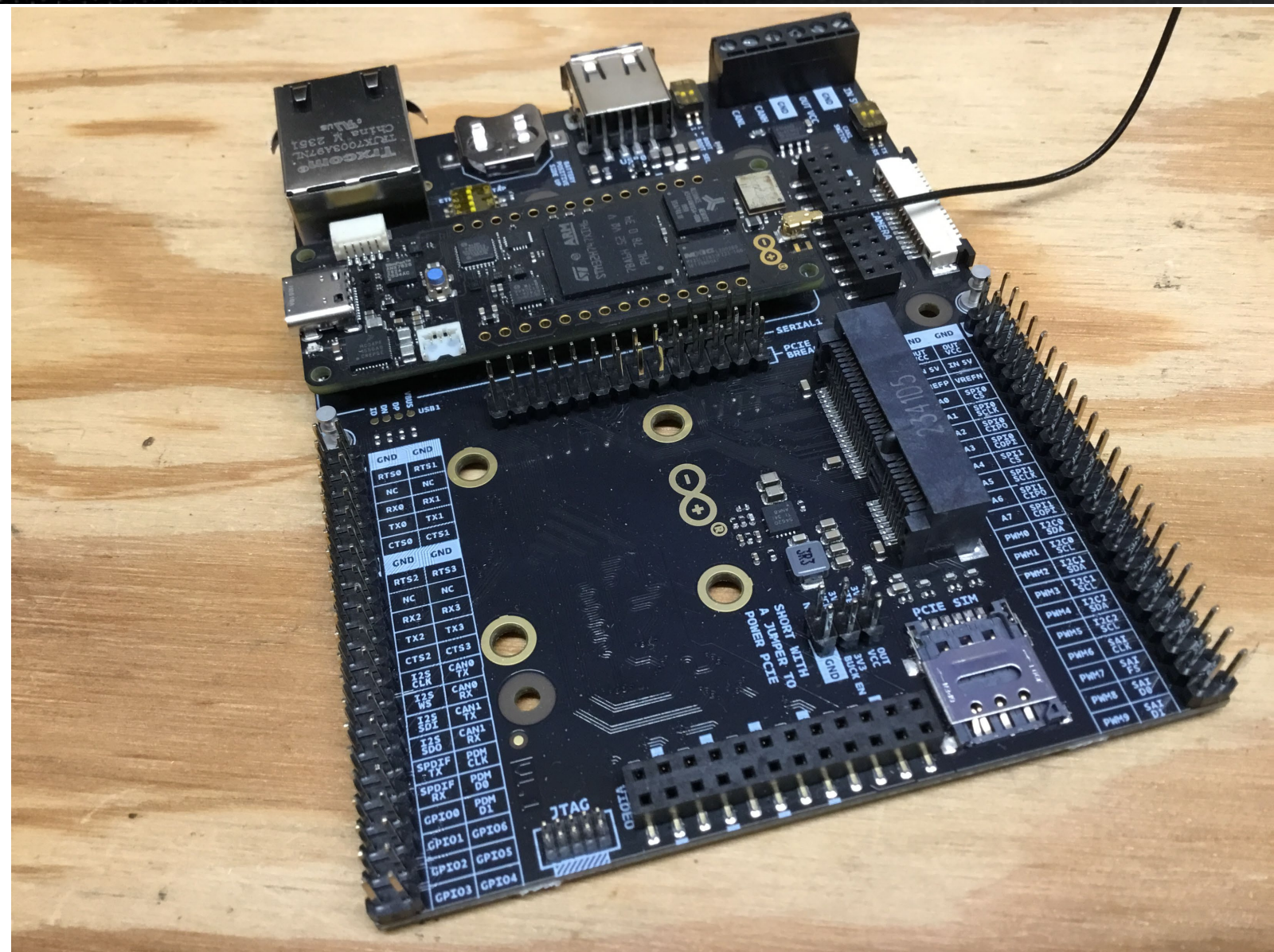
Only show libraries supported by current board

[Refresh Library Index](#)

Ethernet by Arduino
Enables network connection (local and Internet) using the Arduino Ethernet board or shield. For all Arduino boards. With this library you can use the Arduino Ethernet (shield or board) to connect to Internet. The library provides both Client and server functionalities. The library permits you to connect to a local network also with DHCP and to resolve DNS.

[More info](#)

Mount the Portenta H7 on the Mid Carrier



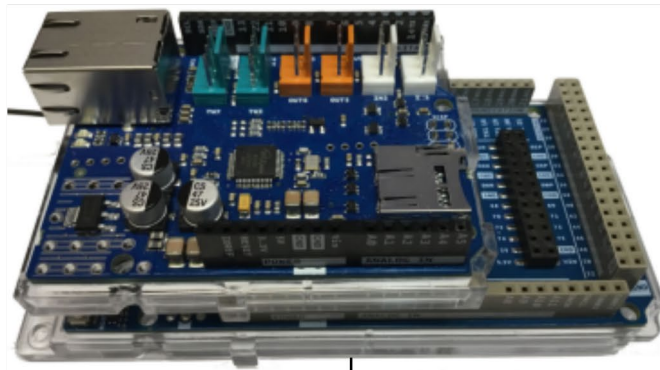
The Plan..

Arduino Giga R1 WiFi

IP Address: 192.168.1.49

MAC Address: 0xDE 0xAD 0xBE 0xEF 0xBA 0xDD

Port: 4044

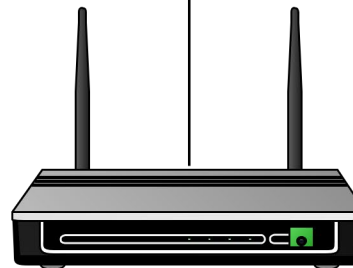
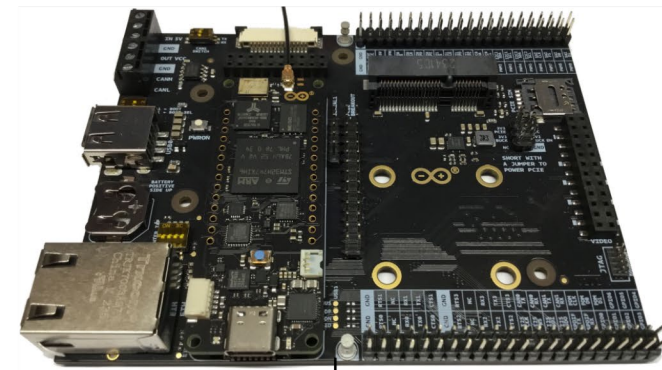


Portenta H7

IP Address: 192.168.1.50

MAC Address: 0xDE 0xAD 0xBE 0xEF 0xBA 0xBE

Port: 4023



TP-Link AXE5400

IP Address: 192.168.1.1

The Code – R1 and H7 Declarations and Definitions

```
1  /* UDP
2  /* Arduino Giga R1 WiFi
3  /* Written by: Fred Eady
4  /* Last Updated: 07/03/2024
5
6  #include <SPI.h>
7  #include <Ethernet.h>
8  #include <EthernetUdp.h>
9
10 String cmd;
11
12 #define port_R1 4044
13 #define port_H7 4023
14
15 IPAddress ip_R1(192,168,1,49);
16 IPAddress ip_H7(192,168,1,50);
17
18 byte mac_addr_R1[] = { 0xDE,0xAD,0xBE,0xEF,0xBA,0xDD};
19 byte mac_addr_H7[] = { 0xDE,0xAD,0xBE,0xEF,0xBA,0xBE};
20 char pktBufIN[128];
21 char pktBufOUT_A[] = "R1 Sent A command";
22 char pktBufOUT_B[] = "R1 Sent B command";
23 char pktBufOUT_C[] = "R1 Sent C command";
24
25 // Spawn an EthernetUDP instance
26 EthernetUDP udp_R1;
```

```
1  /* UDP
2  /* Portenta H7
3  /* Written by: Fred Eady
4  /* Last Updated: 07/03/2024
5
6  #include <PortentaEthernet.h>
7  #include <Ethernet.h>
8  #include <EthernetUdp.h>
9
10 String cmd;
11
12 #define port_R1 4044
13 #define port_H7 4023
14
15 IPAddress ip_R1(192,168,1,49);
16 IPAddress ip_H7(192,168,1,50);
17
18 byte mac_addr_R1[] = { 0xDE,0xAD,0xBE,0xEF,0xBA,0xDD};
19 byte mac_addr_H7[] = { 0xDE,0xAD,0xBE,0xEF,0xBA,0xBE};
20 char pktBufIN[128];
21 char pktBufOUT_A[] = "H7 Sent A command";
22 char pktBufOUT_B[] = "H7 Sent B command";
23 char pktBufOUT_C[] = "H7 Sent C command";
24
25 // Spawn an EthernetUDP instance
26 EthernetUDP udp_H7;
```

UDP (User Datagram Protocol)

The Code – R1 and H7 setup()

```
28 void setup() {
29   // Configure the SPI CS pin
30   Ethernet.init(10);
31   // Start Ethernet
32   Ethernet.begin(mac_addr_R1,ip_R1);
33   // Start UDP
34   udp_R1.begin(port_R1);
35
36   // Open serial COM port
37   Serial1.begin(115200);
38
39   // Check for attached Ethernet hardware
40   if (Ethernet.hardwareStatus() == EthernetNoHardware) {
41     Serial1.println("FATAL ERROR: Ethernet hardware was not found.");
42     while(1) {
43     }
44   }
45   // Check for Connected Ethernet Cable
46   while (Ethernet.linkStatus() == LinkOFF) {
47     Serial1.println("ERROR: Ethernet cable is not connected.");
48     delay(1000);
49   }
50   Serial1.println("Ethernet and UDP are UP!");
51 }
```

```
28 void setup() {
29   // Start Ethernet
30   Ethernet.begin(mac_addr_H7,ip_H7);
31   // Start UDP
32   udp_H7.begin(port_H7);
33
34   // Open serial COM port
35   Serial1.begin(115200);
36
37   // Check for attached Ethernet hardware
38   if (Ethernet.hardwareStatus() == EthernetNoHardware) {
39     Serial1.println("FATAL ERROR: Ethernet hardware was not found.");
40     while(1) {
41     }
42   }
43   // Check for Connected Ethernet Cable
44   while (Ethernet.linkStatus() == LinkOFF) {
45     Serial1.println("ERROR: Ethernet cable is not connected.");
46     delay(1000);
47   }
48   Serial1.println("Ethernet and UDP are UP!");
49 }
```

The Code – R1 and H7 loop()

```
53 void loop() {
54   // Wait for a command from the serial port
55   if(Serial1.available());
56   {
57     cmd = Serial1.readStringUntil('\n');
58     if(cmd.equals("A"))
59     {
60       udp_R1.beginPacket(ip_H7,port_H7);
61       udp_R1.write(pktBufOUT_A);
62       udp_R1.endPacket();
63     }
64     if(cmd.equals("B"))
65     {
66       udp_R1.beginPacket(ip_H7,port_H7);
67       udp_R1.write(pktBufOUT_B);
68       udp_R1.endPacket();
69     }
70     if(cmd.equals("C"))
71     {
72       udp_R1.beginPacket(ip_H7,port_H7);
73       udp_R1.write(pktBufOUT_C);
74       udp_R1.endPacket();
75     }
76   }
```

```
51 void loop() {
52   // Wait for a command from the serial port
53   if(Serial1.available());
54   {
55     cmd = Serial1.readStringUntil('\n');
56     if(cmd.equals("A"))
57     {
58       udp_H7.beginPacket(ip_R1,port_R1);
59       udp_H7.write(pktBufOUT_A);
60       udp_H7.endPacket();
61     }
62     if(cmd.equals("B"))
63     {
64       udp_H7.beginPacket(ip_R1,port_R1);
65       udp_H7.write(pktBufOUT_B);
66       udp_H7.endPacket();
67     }
68     if(cmd.equals("C"))
69     {
70       udp_H7.beginPacket(ip_R1,port_R1);
71       udp_H7.write(pktBufOUT_C);
72       udp_H7.endPacket();
73     }
74   }
```

UDP (User Datagram Protocol)

The Code – R1 and H7 loop()

```
78 // Get incoming UDP data and print it
79 int pktSize = udp_R1.parsePacket();
80 if(pktSize)
81 {
82     Serial1.print("Received UDP Packet From ");
83     IPAddress ipAddr_remote = udp_R1.remoteIP();
84     for(int i=0;i<4;i++)
85     {
86         Serial1.print(ipAddr_remote[i], DEC);
87         if(i<3)
88         {
89             Serial1.print(".");
90         }
91     }
92
93     Serial1.print(": Port ");
94     Serial1.println(udp_R1.remotePort());
95     Serial1.print("Packet Size = ");
96     Serial1.println(pktSize);
97
98     udp_R1.read(pktBufIN, pktSize);
99     Serial1.println("Packet Data:");
100    Serial1.print(pktBufIN);
101 }
102 }
```

```
76 // Get incoming UDP data and print it
77 int pktSize = udp_H7.parsePacket();
78 if(pktSize)
79 {
80     Serial1.print("Received UDP Packet From ");
81     IPAddress ipAddr_remote = udp_H7.remoteIP();
82     for(int i=0;i<4;i++)
83     {
84         Serial1.print(ipAddr_remote[i], DEC);
85         if(i<3)
86         {
87             Serial1.print(".");
88         }
89     }
90
91     Serial1.print(": Port ");
92     Serial1.println(udp_H7.remotePort());
93     Serial1.print("Packet Size = ");
94     Serial1.println(pktSize);
95
96     udp_H7.read(pktBufIN, pktSize);
97     Serial1.println("Packet Data:");
98     Serial1.print(pktBufIN);
99 }
100 }
```

UDP (User Datagram Protocol)

R1/H7 Interaction

```

Monitor Mode Serial View Mode Text Port /dev/ttyUSB0 - FTDI Baud rate 115200 Line ending LF
---- Opened the serial port /dev/ttyUSB0 ----
ERROR: Ethernet cable is not connected.
ERROR: Ethernet cable is not connected.
Ethernet and UDP are UP!
---- Sent utf8 encoded message: "A\n" ----
Received UDP Packet From 192.168.1.50: Port 4023
Packet Size = 17
Packet Data:
H7 Sent B command|

```

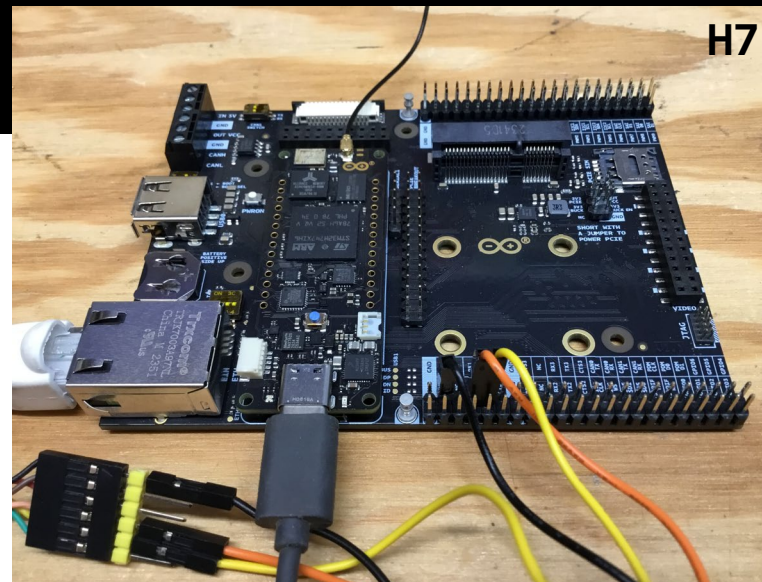
R1

```

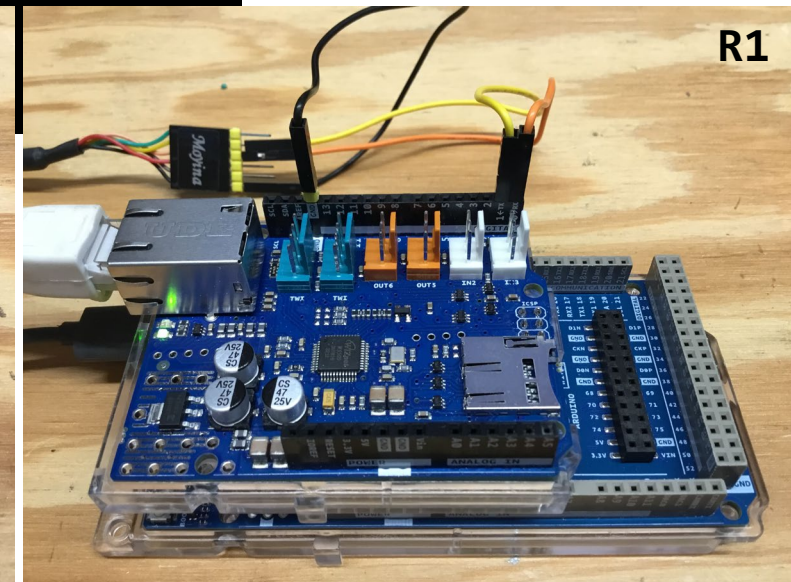
Monitor Mode Serial View Mode Text Port /dev/ttyUSB0 - FTDI Baud rate 115200 Line ending LF
---- Opened the serial port /dev/ttyUSB0 ----
Ethernet and UDP are UP!
Received UDP Packet From 192.168.1.49: Port 4044
Packet Size = 17
Packet Data:
R1 Sent A command
---- Sent utf8 encoded message: "B\n" ----
|

```

H7



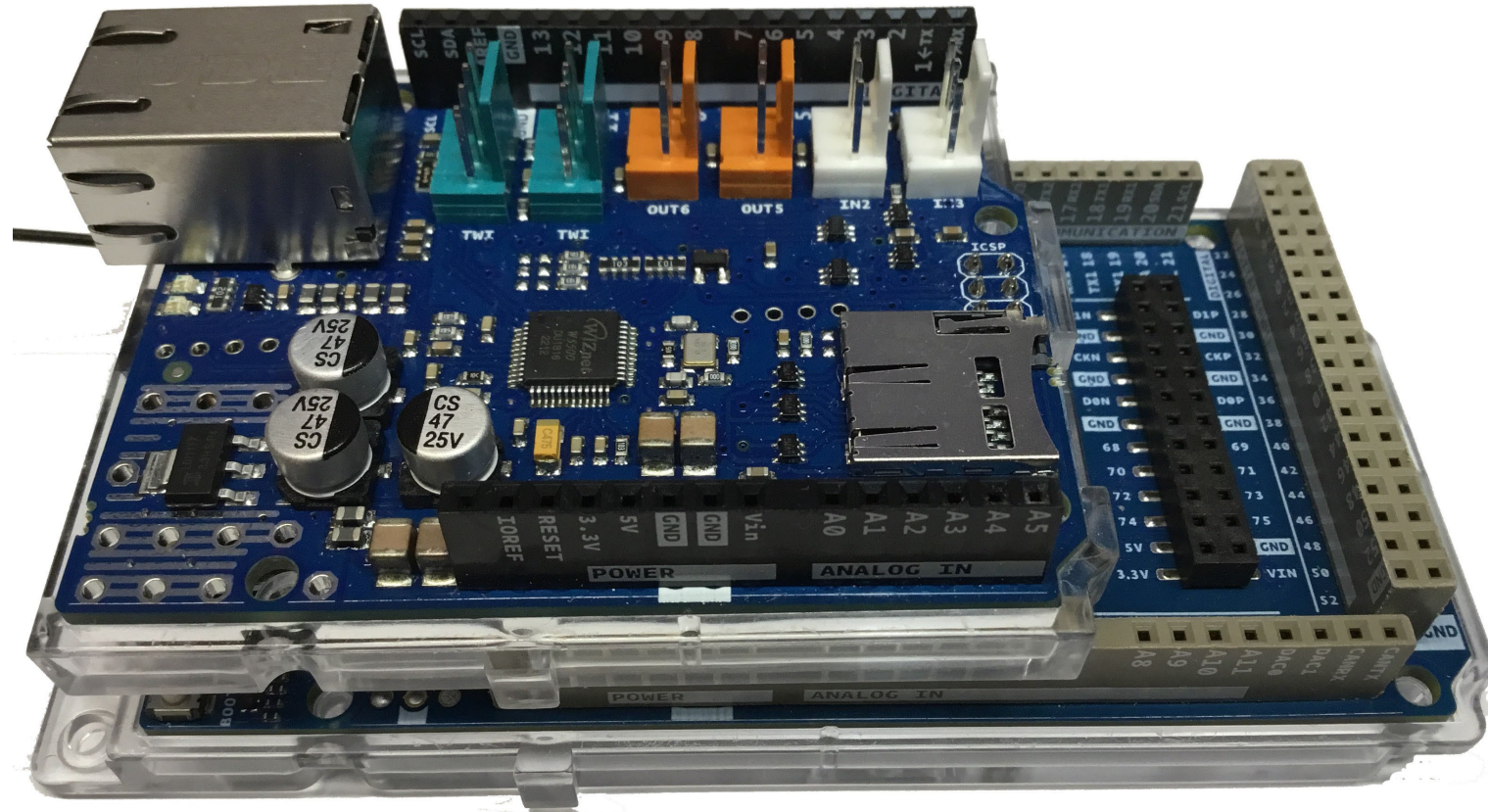
H7



R1

The Code – R1 MAC Address, Port and IP Definitions

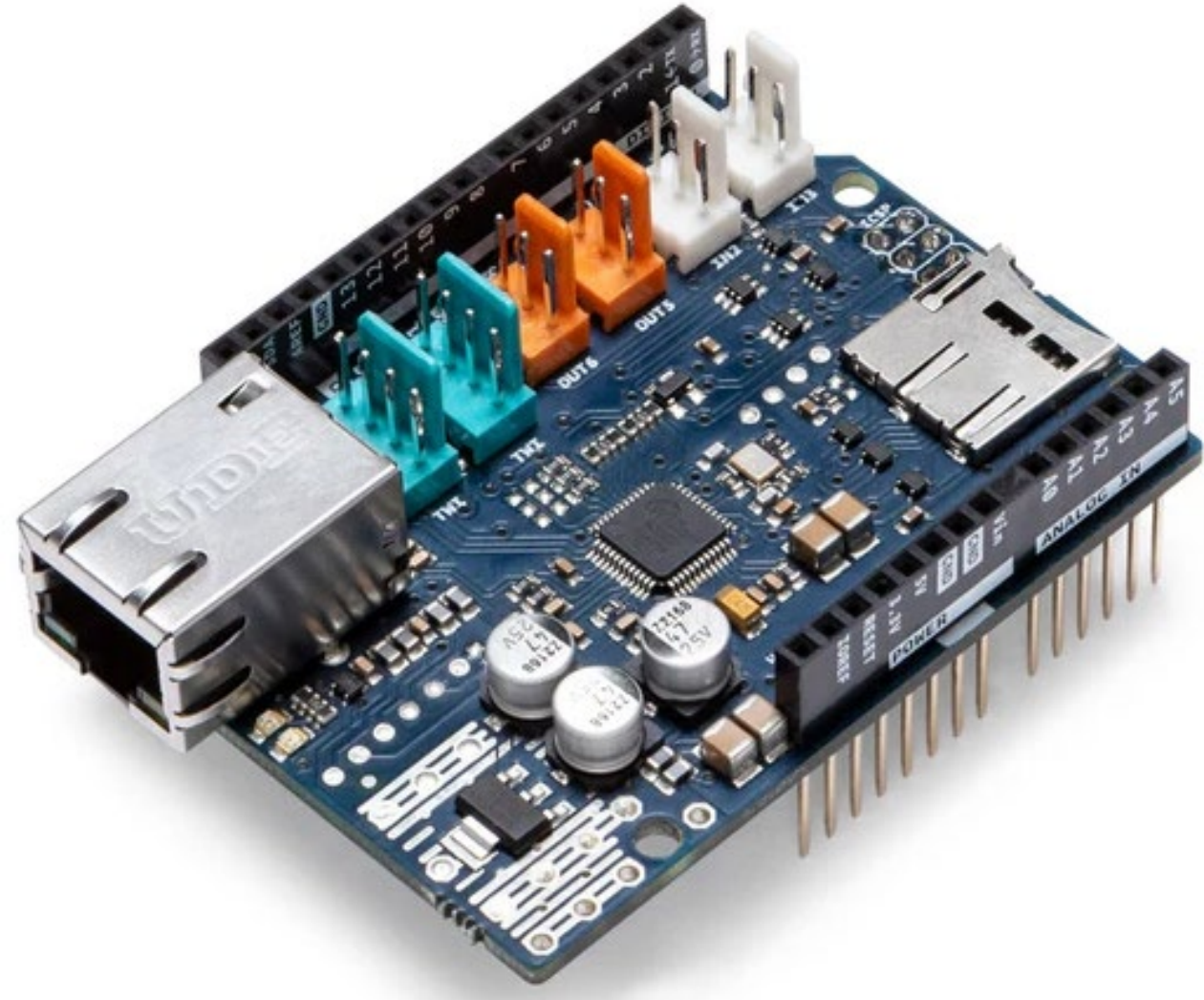
```
1  /* TCP CLIENT
2  /* ARDUINO GIGA R1 WIFI with Ethernet Shield2
3  /* Written by: Fred Eady
4  /* Last Updated: 07/04/2024
5
6  #include <SPI.h>
7  #include <Ethernet.h>
8
9  #define server_port 8088
10
11 // Ethernet Shield2 MAC Addr
12 byte client_mac_addr[] = {
13 | 0xBA, 0xDC, 0x0F, 0xFE, 0xEB, 0xED
14 };
15 // TCP Client IP Address
16 IPAddress client_ip(192, 168, 1, 50);
17
18 // TCP Server IP Address
19 IPAddress server_ip(192, 168, 1, 238);
20
21 // Spawn the tcp_client instance
22 EthernetClient tcp_client;
```



TCP/IP (Transmission Control Protocol/Internet Protocol)

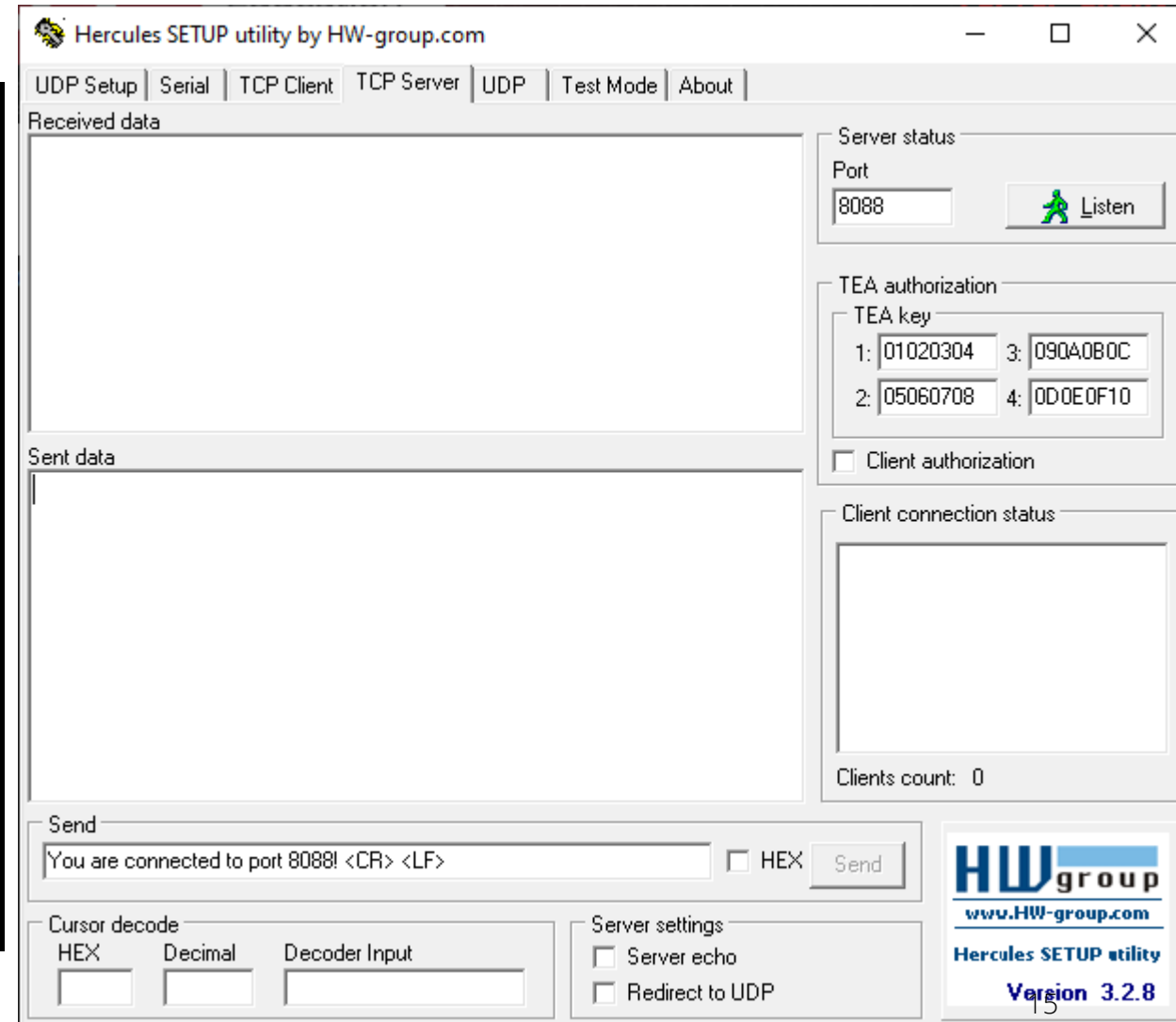
The Code – R1 Client setup()

```
24 void setup() {
25     // Use Ethernet.init(pin) to configure the CS pin
26     Ethernet.init(10); // Reference Ethernet Shield2 schematic
27     // Start the Ethernet connection
28     Ethernet.begin(client_mac_addr, client_ip);
29     // Open serial COM port
30     Serial1.begin(115200);
31
32     // Check for attached Ethernet hardware
33     if (Ethernet.hardwareStatus() == EthernetNoHardware) {
34         Serial1.println("FATAL ERROR: Ethernet shield was not found.");
35         while(1) {
36             }
37     }
38     while (Ethernet.linkStatus() == LinkOFF) {
39         Serial1.println("ERROR: Ethernet cable is not connected.");
40         delay(1000);
41     }
42
43     // Allow time for Ethernet shield to initialize
44     Serial1.println("Attempting to make a connection...");
45     // if we get a connection, report back via serial port
46     if (tcp_client.connect(server_ip, server_port)) {
47         Serial1.println("Connected to the server...");
48     }
49     else {
50         // if we didn't get a connection to the server
51         Serial1.println("Connection to the server failed");
52     }
53 }
```



The Code – R1 Client loop()

```
55 void loop() {
56     // Get incoming data from the TCP server and display it via Serial1
57     if (tcp_client.available()) {
58         char biteIn = tcp_client.read();
59         Serial1.print(biteIn);
60     }
61
62     // Send Serial1 COM port input data to TCP server
63     while (Serial1.available() > 0) {
64         char biteOut = Serial1.read();
65         if (tcp_client.connected()) {
66             tcp_client.print(biteOut);
67         }
68     }
69
70     // Stop the TCP client on TCP server disconnection
71     if (!tcp_client.connected()) {
72         Serial1.println();
73         Serial1.println("Disconnecting from the server.");
74         tcp_client.stop();
75         while(1) {
76         }
77     }
78 }
```



R1 TCP/IP Client-Server Communications

Hercules SETUP utility by HW-group.com

UDP Setup | Serial | TCP Client | TCP Server | UDP | Test Mode | About

Received data

```
TCP Client to TCP Server{0A}
```

Sent data

```
You are connected to port 8088! {0D} {0A}
```

Send

You are connected to port 8088! <CR> <LF> HEX

Cursor decode

HEX	Decimal	Decoder Input
7D	125	

Server status

Port: 8088

TEA authorization

TEA key

1: 01020304	3: 090A0B0C
2: 05060708	4: 0D0E0F10

Client authorization

Client connection status

```
1:54:54 PM: 192.168.1.50 Client co
```

Clients count: 0

HWgroup
www.HW-group.com
Hercules SETUP utility
Version 3.2.8

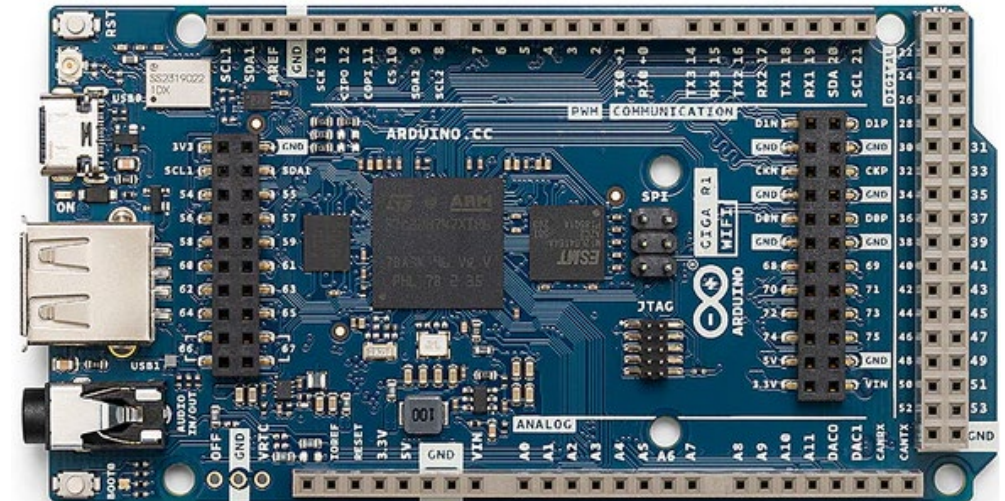
PROBLEMS 2 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS | SERIAL MONITOR

+ Open an additional monitor

Monitor Mode: Serial | View Mode: Text | Port: /dev/ttyUSB0 - FTDI

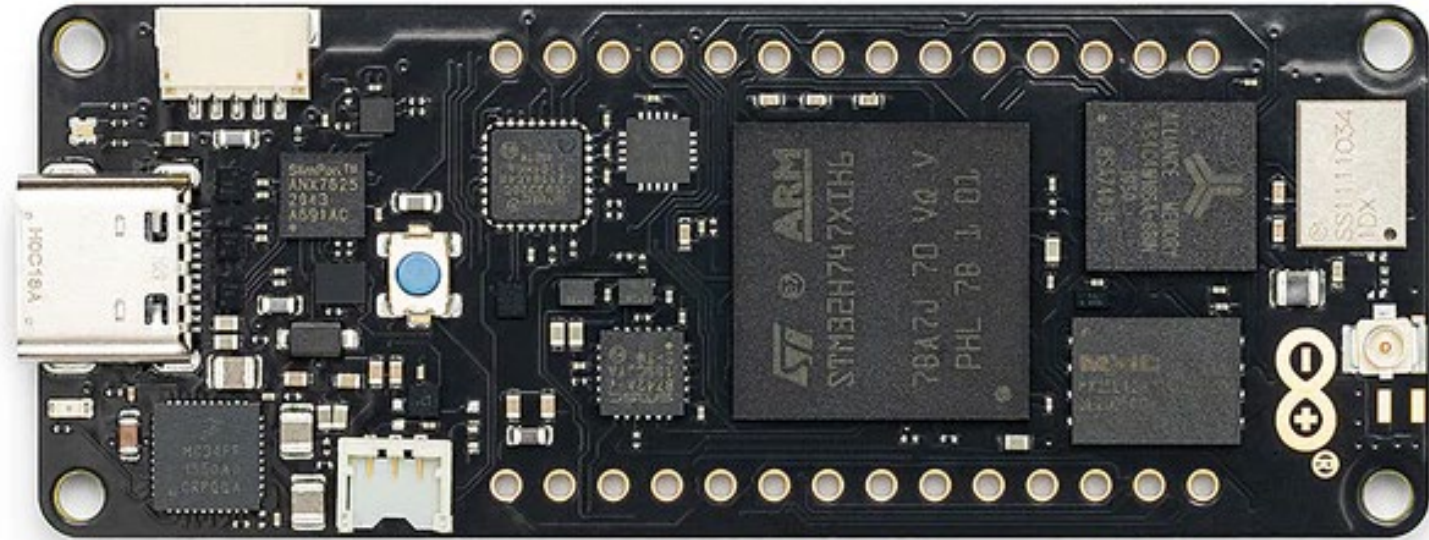
```
ERROR: Ethernet cable is not connected.
ERROR: Ethernet cable is not connected.
Attempting to make a connection...
Connected to the server...
---- Sent utf8 encoded message: "TCP Client to TCP Server\n" ----
You are connected to port 8088!

Disconnecting from the server.
```



The Code – H7 Variables, IP Addresses and Instances

```
1  /** WiFi
2  /** Portenta H7
3  /** Written by: Fred Eady
4  /** Last Updated: 07/04/2024
5
6  #include <WiFi.h>
7  #include "arduino_secrets.h"
8
9  char ssid[] = SECRET_WIFI_SSID; // really not so secret
10 char pass[] = SECRET_WIFI_PASS;
11 int status = WL_IDLE_STATUS;
12
13 IPAddress wifi_server(192,168,1,238);
14
15 // Spawn the h7_wifi_client instance
16 WiFiClient h7_wifi_client;
```



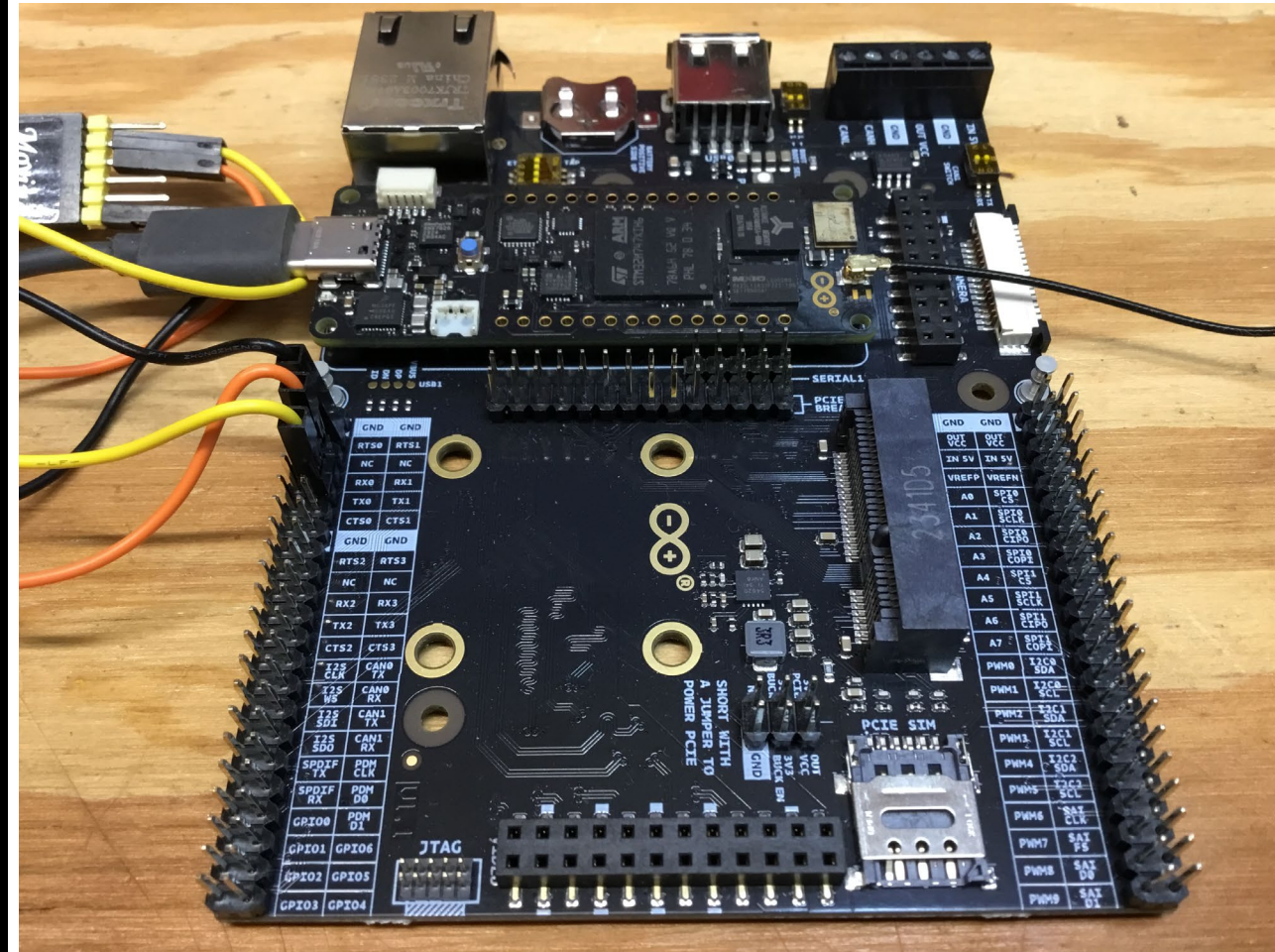
TCP/IP (Transmission Control Protocol/Internet Protocol)

The Code – H7 WiFi setup()

```

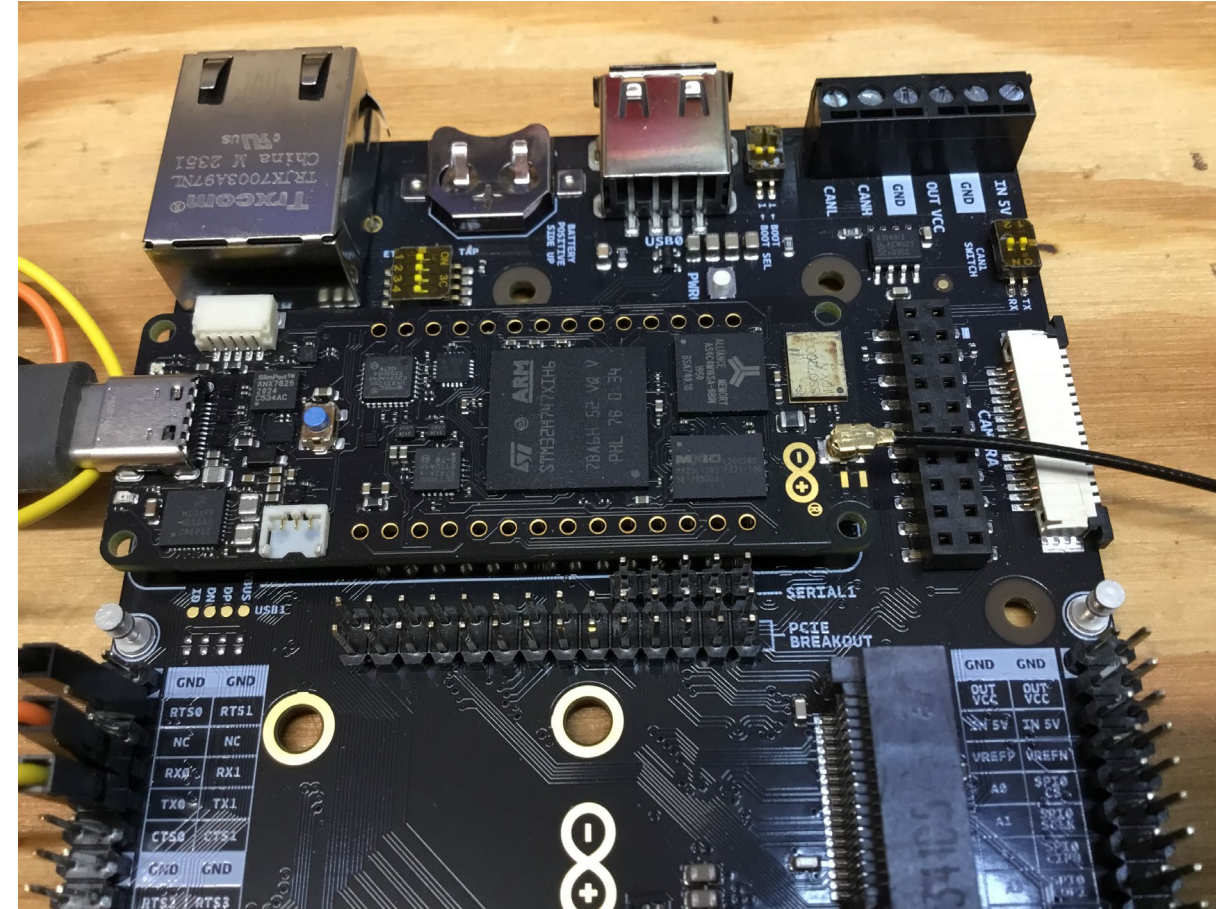
18 void setup() {
19     // Open Serial1 COM Port
20     Serial1.begin(115200);
21
22     // Check the WiFi hardware status
23     if (WiFi.status() == WL_NO_SHIELD) {
24         Serial1.println("FATAL ERROR: WiFi hardware did not respond.");
25         while (true);
26     }
27
28     // Attempt to connect to Wifi network
29     while (status != WL_CONNECTED) {
30         Serial1.print("Attempting to connect to SSID -> ");
31         Serial1.println(ssid);
32         status = WiFi.begin(ssid, pass);
33     }
34     Serial1.print("Connected to SSID -> ");
35     Serial1.println(WiFi.SSID());
36
37     // Display the Portenta H7's IP address
38     IPAddress ip = WiFi.localIP();
39     Serial1.print("IP Address -> ");
40     Serial1.println(ip);
41
42     // If we get a connection, report back via serial port
43     if (h7_wifi_client.connect(wifi_server, 8088)) {
44         Serial1.println("Connected to WiFi server");
45     }
46 }

```



The Code – H7 WiFi loop()

```
48 void loop() {
49   // Get incoming data from the WiFi server and display it via Serial1
50   while (h7_wifi_client.available()) {
51     char c = h7_wifi_client.read();
52     Serial1.write(c);
53   }
54
55   // Send Serial1 COM port input data to the WiFi server
56   while (Serial1.available() > 0) {
57     char biteout = Serial1.read();
58     if (h7_wifi_client.connected()) {
59       h7_wifi_client.print(biteout);
60     }
61   }
62
63   // Stop the WiFi client on WiFi server disconnection
64   if (!h7_wifi_client.connected()) {
65     Serial1.println();
66     Serial1.println("Disconnecting from WiFi server.");
67     h7_wifi_client.stop();
68     while (true);
69   }
70 }
```



H7 WiFi Client-Server Communications

Hercules SETUP utility by HW-group.com

UDP Setup | Serial | TCP Client | TCP Server | UDP | Test Mode | About

Received data

```
H7 WiFi Client{0A}
```

Sent data

```
You are connected to port 8088 via WiFi! {0D} {0A}
```

Send

You are connected to port 8088 via WiFi! <CR> <LF> HEX Send

Cursor decode

HEX	Decimal	Decoder Input
7D	125	

Server status

Port: 8088

TEA authorization

TEA key

1: 01020304	3: 090A0B0C
2: 05060708	4: 0D0E0F10

Client authorization

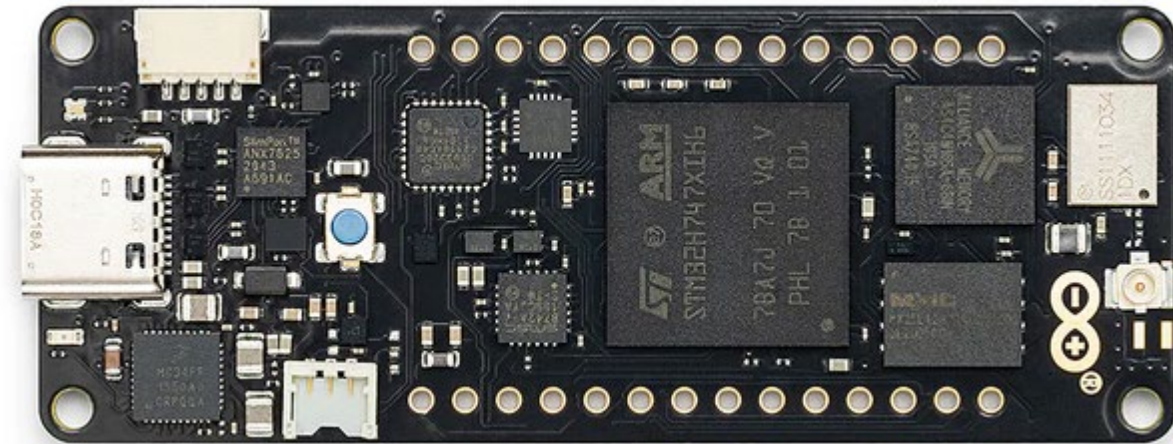
Client connection status

```
3:46:38 PM: 192.168.1.42 Client co
```

Clients count: 0

HWgroup
www.HW-group.com
Hercules SETUP utility
Version 3.2.8

```
Monitor Mode Serial View Mode Text Port /dev/ttyUSB0 - FTDI
Attempting to connect to SSID -> edtpnet2
Connected to SSID -> edtpnet2
IP Address -> 192.168.1.42
Connected to WiFi server
---- Sent utf8 encoded message: "H7 WiFi Client\n" ----
You are connected to port 8088 via WiFi!
```



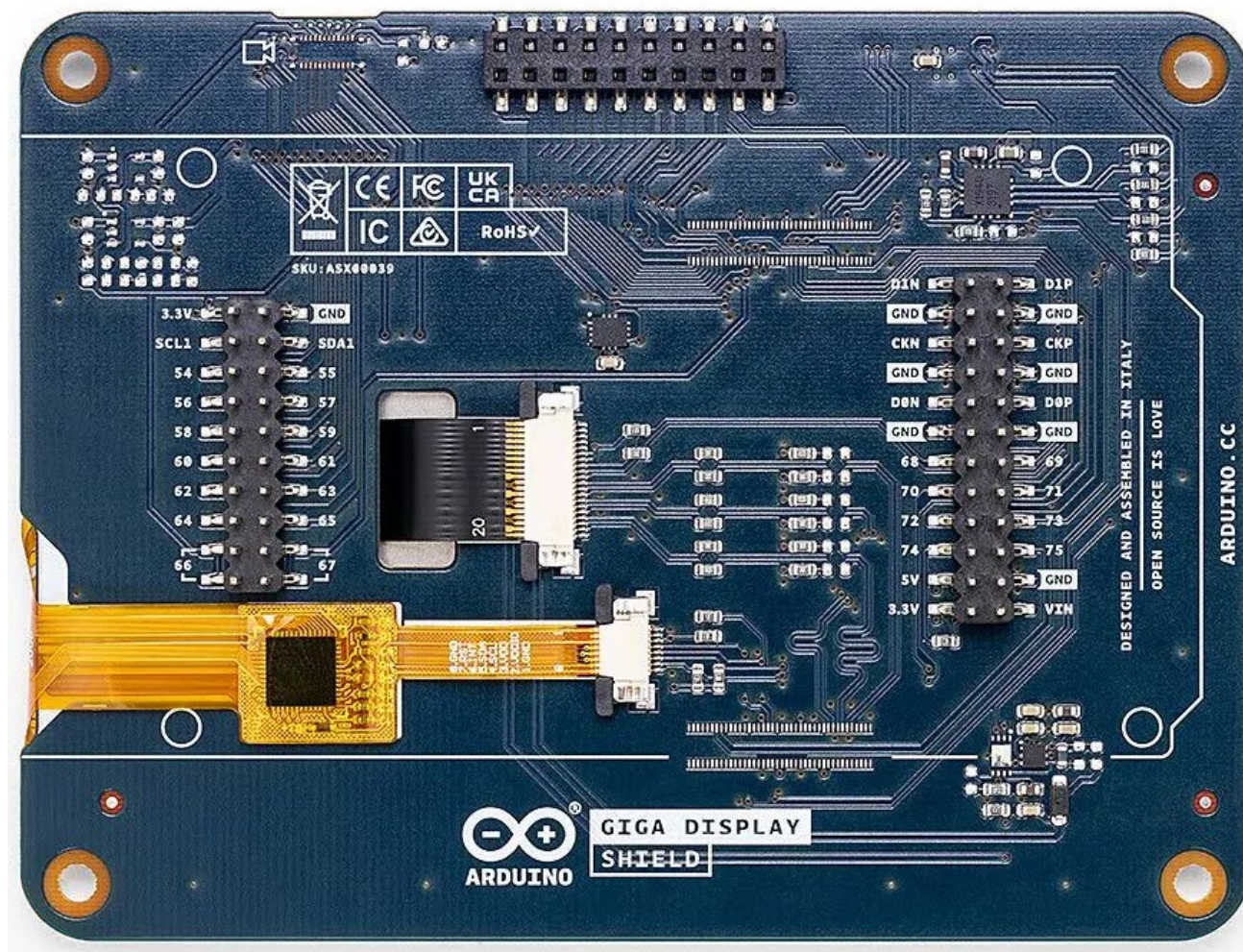
Next Time...

MORE TO COME..

Thank you for attending!!!

Please consider the resources below:

- [Today's Download Package](#)
- arduino.cc





Thank You

Sponsored by

