**EXPERIMENT 1**

// Example testing sketch for various DHT humidity/temperature sensors written by ladyada

// REQUIRES the following Arduino libraries:

// - DHT Sensor Library: https://github.com/adafruit/DHT-sensor-library

// - Adafruit Unified Sensor Lib: https://github.com/adafruit/Adafruit\_Sensor

#include "DHT.h"

#define DHTPIN 4 // Digital pin connected to the DHT sensor

// Feather HUZZAH ESP8266 note: use pins 3, 4, 5, 12, 13 or 14 --

// Pin 15 can work but DHT must be disconnected during program upload.

// Uncomment whatever type you're using!

#define DHTTYPE DHT11 // DHT 11

//#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321

//#define DHTTYPE DHT21 // DHT 21 (AM2301)

// Connect pin 1 (on the left) of the sensor to +5V

// NOTE: If using a board with 3.3V logic like an Arduino Due connect pin 1

// to 3.3V instead of 5V!

// Connect pin 2 of the sensor to whatever your DHTPIN is

// Connect pin 4 (on the right) of the sensor to GROUND

// Connect a 10K resistor from pin 2 (data) to pin 1 (power) of the sensor

// Initialize DHT sensor.

// Note that older versions of this library took an optional third parameter to

// tweak the timings for faster processors. This parameter is no longer needed

// as the current DHT reading algorithm adjusts itself to work on faster procs.

DHT dht(DHTPIN, DHTTYPE);

void setup() {

Serial.begin(115200);

Serial.println(F("DHTxx test!"));

dht.begin();

}

void loop() {

// Wait a few seconds between measurements.

delay(2000);

// Reading temperature or humidity takes about 250 milliseconds!

// Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)

float h = dht.readHumidity();

// Read temperature as Celsius (the default)

float t = dht.readTemperature();

// Read temperature as Fahrenheit (isFahrenheit = true)

float f = dht.readTemperature(true);

// Check if any reads failed and exit early (to try again).

if (isnan(h) || isnan(t) || isnan(f)) {

Serial.println(F("Failed to read from DHT sensor!"));

return;

}

// Compute heat index in Fahrenheit (the default)

float hif = dht.computeHeatIndex(f, h);

// Compute heat index in Celsius (isFahreheit = false)

float hic = dht.computeHeatIndex(t, h, false);

Serial.print(F("Humidity: "));

Serial.print(h);

Serial.print(F("% Temperature: "));

Serial.print(t);

Serial.print(F("°C "));

Serial.print(f);

Serial.print(F("°F Heat index: "));

Serial.print(hic);

Serial.print(F("°C "));

Serial.print(hif);

Serial.println(F("°F"));

}

**EXPERIMENT 2a**

/\*

\* This sketch demonstrates how to scan WiFi networks.

\* The API is almost the same as with the WiFi Shield library,

\* the most obvious difference being the different file you need to include:

\*/

#include "WiFi.h"

void setup()

{

Serial.begin(115200);

// Set WiFi to station mode and disconnect from an AP if it was previously connected

WiFi.mode(WIFI\_STA);

WiFi.disconnect();

delay(100);

Serial.println("Setup done");

}

void loop()

{

Serial.println("scan start");

// WiFi.scanNetworks will return the number of networks found

int n = WiFi.scanNetworks();

Serial.println("scan done");

if (n == 0) {

Serial.println("no networks found");

} else {

Serial.print(n);

Serial.println(" networks found");

for (int i = 0; i < n; ++i) {

// Print SSID and RSSI for each network found

Serial.print(i + 1);

Serial.print(": ");

Serial.print(WiFi.SSID(i));

Serial.print(" (");

Serial.print(WiFi.RSSI(i));

Serial.print(")");

Serial.println((WiFi.encryptionType(i) == WIFI\_AUTH\_OPEN)?" ":"\*");

delay(10);

}

}

Serial.println("");

// Wait a bit before scanning again

delay(5000);

}

**EXPERIMENT 2b**

#include <WiFi.h>

const char \*ssid = "IOT Lab";

const char \*passphrase = "012345678";

IPAddress local\_IP(192,168,4,22);

IPAddress gateway(192,168,4,9);

IPAddress subnet(255,255,255,0);

void setup()

{

Serial.begin(115200);

Serial.println();

Serial.print("Setting soft-AP configuration ... ");

Serial.println(WiFi.softAPConfig(local\_IP, gateway, subnet) ? "Ready" : "Failed!");

Serial.print("Setting soft-AP ... ");

Serial.println(WiFi.softAP(ssid,passphrase) ? "Ready" : "Failed!");

//WiFi.softAP(ssid);

//WiFi.softAP(ssid, passphrase, channel, ssdi\_hidden, max\_connection)

Serial.print("Soft-AP IP address = ");

Serial.println(WiFi.softAPIP());

}

void loop() {

Serial.print("[Server Connected] ");

Serial.println(WiFi.softAPIP());

delay(500);

}

**EXPERIMENT 2c**

//#ifndef \_\_CC3200R1M1RGC\_\_

// Do not include SPI for CC3200 LaunchPad

//#include <SPI.h>

//#endif

#include <WiFi.h>

char ssid[]="iot lab";

char password[]="12345678";

IPAddress ip;

IPAddress gateway;

void setup()

{

Serial.begin(115200); //Initialize Serial Port

//attempt to connect to wifi

Serial.print("Attempting to connect to Network named: ");

// print the network name (SSID);

Serial.println(ssid);

//Connect to WiFI

WiFi.begin(ssid, password);

//Wait untill wifi is connected

while ( WiFi.status() != WL\_CONNECTED)

{

// print dots while we wait to connect

Serial.print(".");

delay(300);

}

//If you are connected print the IP Address

Serial.println("\nYou're connected to the network");

//wait untill you get an IP address

while (WiFi.localIP() == INADDR\_NONE) {

// print dots while we wait for an ip addresss

Serial.print(".");

delay(300);

}

ip=WiFi.localIP();

Serial.println(ip);

gateway=WiFi.gatewayIP();

Serial.println("GATEWAY IP:");

Serial.println(gateway);

}

void loop()

{

// put your main code here, to run repeatedly:

}