

```

#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <WiFiUdp.h>

#include <AzureIoTHub.h>
#include <AzureIoTProtocol_MQTT.h>
#include <AzureIoTUtility.h>

#include "config.h"

static bool messagePending = false;
static bool messageSending = true;

static char *connectionString;
static char *ssid;
static char *pass;

static int interval = INTERVAL;

void blinkLED()
{
    digitalWrite(LED_PIN, HIGH);
    delay(500);
    digitalWrite(LED_PIN, LOW);
}

void initWifi()
{
    // Attempt to connect to Wifi network:
    Serial.printf("Attempting to connect to SSID: %s.\r\n", ssid);

    // Connect to WPA/WPA2 network. Change this line if using open or WEP network:
    WiFi.begin(ssid, pass);
    while (WiFi.status() != WL_CONNECTED)
    {
        // Get Mac Address and show it.
        // WiFi.macAddress(mac) save the mac address into a six length array, but the endian may
        be different. The huzzah board should
        // start from mac[0] to mac[5], but some other kinds of board run in the oppsite direction.
        uint8_t mac[6];
        WiFi.macAddress(mac);
    }
}

```

```

        Serial.printf("You device with MAC address %02x:%02x:%02x:%02x:%02x:%02x connects
to %s failed! Waiting 10 seconds to retry.\r\n",
        mac[0], mac[1], mac[2], mac[3], mac[4], mac[5], ssid);
        WiFi.begin(ssid, pass);
        delay(10000);
    }
    Serial.printf("Connected to wifi %s.\r\n", ssid);
}

```

```

void initTime()
{
    time_t epochTime;
    configTime(0, 0, "pool.ntp.org", "time.nist.gov");

    while (true)
    {
        epochTime = time(NULL);

        if (epochTime == 0)
        {
            Serial.println("Fetching NTP epoch time failed! Waiting 2 seconds to retry.");
            delay(2000);
        }
        else
        {
            Serial.printf("Fetched NTP epoch time is: %lu.\r\n", epochTime);
            break;
        }
    }
}

```

```

static IOTHUB_CLIENT_LL_HANDLE iotHubClientHandle;
void setup()
{
    pinMode(LED_PIN, OUTPUT);

    initSerial();
    delay(2000);
    readCredentials();

    initWifi();
    initTime();
    initSensor();
}

```

```

/*
 * AzureIoT Hub library remove AzureIoT HubClient class in 1.0.34, so we remove the code
below to avoid
 * compile error
 */

// initIoT HubClient();
IoT HubClientHandle = IoT HubClient_LL_CreateFromConnectionString(connectionString,
MQTT_Protocol);
if (IoT HubClientHandle == NULL)
{
    Serial.println("Failed on IoT HubClient_LL_CreateFromConnectionString.");
    while (1);
}

IoT HubClient_LL_SetOption(IoTHubClientHandle, "product_info",
"HappyPath_AdafruitFeatherHuzzah-C");
IoT HubClient_LL_SetMessageCallback(IoTHubClientHandle, receiveMessageCallback,
NULL);
IoT HubClient_LL_SetDeviceMethodCallback(IoTHubClientHandle, deviceMethodCallback,
NULL);
IoT HubClient_LL_SetDeviceTwinCallback(IoTHubClientHandle, twinCallback, NULL);
}

static int messageCount = 1;
void loop()
{
    if (!messagePending && messageSending)
    {
        char messagePayload[MESSAGE_MAX_LEN];
        bool temperatureAlert = readMessage(messageCount, messagePayload);
        sendMessage(IoTHubClientHandle, messagePayload, temperatureAlert);
        messageCount++;
        delay(interval);
    }
    IoT HubClient_LL_DoWork(IoTHubClientHandle);
    delay(10);
}

```