```
#include "Arduino.h"
#include <EMailSender.h>
#include <ESP8266HTTPClient.h>
#include <ESP8266httpUpdate.h>
#include "EspMQTTClient.h"

#define VERSION "v23.04.1802"
#define HOST "mailAlert"
const char* urlBase = "http://192.168.1.208:5000/update";

float RatioFactor = 1.00;
int BAT = A0;
const byte relayPin = D6;
EMailSender emailSend("donbrew7@gmail.com", "tqsodgxyikevcurx");
EspMQTTClient client(
  "Dhome2",
  "TiffanyZ7",
  "192.168.1.208",  // MQTT Broker server ip
  "",   // Can be omitted if not needed
  "",   // Can be omitted if not needed
  "mailAlert",     // Client name that uniquely identify your device
  1883              // The MQTT port, default to 1883. this line can be omitted
);

void checkForUpdates()
{
  WiFiClient espClient;
  String checkUrl = String( urlBase);
  checkUrl.concat( "?ver=" + String(VERSION) );
  checkUrl.concat( "&dev=" + String(HOST) );
  Serial.println("INFO: Checking for updates at URL: " + String( checkUrl ) );
  t_httpUpdate_return ret = ESPhttpUpdate.update(espClient, checkUrl );
  switch (ret) {
    default:
    case HTTP_UPDATE_FAILED:
      Serial.println("ERROR: HTTP_UPDATE_FAILD Error (" +
      String(ESPhttpUpdate.getLastError()) + "): " +
      ESPhttpUpdate.getLastErrorString().c_str());
      break;
    case HTTP_UPDATE_NO_UPDATES:
      Serial.println("INFO: HTTP_UPDATE_NO_UPDATES");
      break;
    case HTTP_UPDATE_OK:
      Serial.println("INFO status: HTTP_UPDATE_OK");
      break;
  }
}

void onConnectionEstablished()
{
  ////////////////////////////////////Battery Voltage////////////////////////////////////
  //with 100K resistor for 4.2V
  float Tvoltage = 0.0;
  float Vvalue = 0.0, Rvalue = 0.0;
  for (unsigned int i = 0; i < 10; i++) {
    Vvalue = Vvalue + analogRead(BAT);     //Read analog Voltage
    delay(50);                              //ADC stable
  }
  Vvalue = (float)Vvalue / 10;         //Find average of 10 values
  Rvalue = (float)(Vvalue / 1023.0) * 4.2 ; //Convert Voltage in 4.2v factor
  Tvoltage = Rvalue * RatioFactor;      //Find original voltage by multiplying with
  factor
   char _volt[8];                                 // Buffer big enough for 7-character
   float
  dtostrf(Tvoltage, 3, 2, _volt);               // Leave room for too large numbers!

  delay(50);
  String(version) = "  Mail Version =  " + String(VERSION);
  String(volts) = "   Mail Voltage =  " + String(Tvoltage);
```

```
    checkForUpdates();
    client.publish("mailStatus", "delivered");
    client.publish("mailVolt", _volt);
    client.publish("mailversion", VERSION);
    EMailSender::EMailMessage message;
    message.subject = "Mail Delivered";
    message.message = "Mail Delivered" + String(version) + String(volts);
    EMailSender::Response resp = emailSend.send("donbrew9@gmail.com", message);
    delay(2000);
    Serial.println("Sending status: ");
    Serial.println(resp.status);
    Serial.println(resp.code);
    Serial.println(resp.desc);
    delay(1000);
    Serial.println("Going to Sleep");
    digitalWrite(relayPin, HIGH); //turn off
  }

  void setup()
  {
    Serial.begin(115200);
    pinMode(relayPin, OUTPUT);
    digitalWrite(relayPin, LOW);
  }

  void loop()
  {
    client.loop();
  }
```