

# Questions étudiants GdP

## Challenge 1 Datasheet

```
char s[10];

void setup() {
  Serial.begin(9600);
}

// Connecter GND sur GND
// D4 (TXD1) du broadcaster sur notre RX (RXD0) (Transmitter vers Receiver), qui
// est l'entrée RX par défaut lorsqu'on initialise un Serial

void loop() {
  Serial.readBytes(s,10);
  Serial.print("Received:");
  Serial.println(s);
  delay(1000);
}
```

- Que veut dire RX/TX? Ou quelle est leur utilisation?
  - Receiver/Transmitter
- Expliquer comment marche votre code. (Si le code qu'ils ont est minimaliste)
- Si j'avais voulu lire dans mon moniteur série avec un baudrate de 115200, qu'est ce que vous auriez du changer dans votre code?

### S'ils n'ont pas réussi le challenge 1:

- Comment savoir a quel pin correspond TXD1? A quel pin ça correspond?
- Quelle connexion de pins aurait été nécessaire si vous aviez voulu récupérer un envoi de donnée sur TXD1?

## Challenge 2 PWM

```
// GND to GND
// A0 to bottom of potentiometer
// TX to RX

const int analogInPin = A0;

void setup() {
  Serial.begin(9600);
}

void loop() {
  int value = analogRead(analogInPin);
  Serial.println("1#" + String(value) + "/" );
  delay(200);
}
```

- Est ce que tous les pins peuvent être utilisés avec `analogRead` ?
  - Non, A0 est le seul a pouvoir lire en analogique, et il y a 9 autres pin qui peuvent envoyer en PWM (D1-D8+RSV)
- Expliquer comment marche votre code. (Si le code qu'ils ont est minimaliste)
- Quel est l'intervalle de la valeur qu'on pouvait récupérer avec `analogRead` ?
- Si le broadcaster attendait une valeur Serial avec un baudrate de 115200, qu'est ce que vous auriez du changer dans votre code?

## Challenge 3 Wifi

```
const char* ssid = "*****";
const char* password = "*****";

String url = "*****";

void setup() {
  WiFi.begin(ssid, password);
  Serial.begin(9600);
  Serial.println("Connecting");
  while(WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("Connected to WiFi network with IP Address: ");
  Serial.println(WiFi.localIP()); // Expliquer cette ligne 0
}

void loop() {
  delay(5000);
  if(WiFi.status() == WL_CONNECTED){ // Expliquer cette ligne 1
    WiFiClient client;
    HTTPClient http;

    http.begin(client, url.c_str()); // Expliquer cette ligne 2
    String httpRequestBody="4#4v17lk";
    int httpStatusCode = http.POST(httpRequestBody); // Expliquer cette
    ligne 3
    if(httpStatusCode > 0){
      String response = http.getString(); //Get the response to the request
      Serial.println(httpStatusCode); //Print return code
      Serial.println(response); //Print request answer
    } else {
      Serial.print("Error on sending POST: ");
      Serial.println(httpStatusCode);

      http.end();
    }
  }
  else {
    Serial.println("WiFi Disconnected");
  }
}
```

```
}
```

- Expliquer le code
- Expliquez cette ligne s'ils l'ont dans leur code (Voir sur le wu les lignes intéressantes a demander)
  - 0: print l'IP du nodeMCU assigné par la wifi (enfin par le DHCP du router pour être précis)
  - 1: vérification que l'objet global Wifi (la librairie agit comme un objet) est toujours connecté
  - 2: paramétrage du client http avec le client wifi + le nom de domaine que l'ont veut request
  - 3: envoie de la requête HTTP, récupération du status code (200 si bon, 404,500,501 sinon)
- Si j'avais voulu lire dans mon moniteur série avec un baudrate de 115200, qu'est ce que vous auriez du changer dans votre code?

## Challenge 4 MQTT Sub

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

const char* ssid = "*****";
const char* password = "****";

WiFiClient wifiClient;
PubSubClient mqttClient(wifiClient);
char *mqttServer = "*****";
int mqttPort = 1883;

void setupMQTT() {
    mqttClient.setServer(mqttServer, mqttPort);
    // set the callback function
    mqttClient.setCallback(callback);
}

void setup_wifi() {

    delay(10);

    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);

    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.print("Connected to WiFi network with IP Address: ");
```

```

    Serial.println(WiFi.localIP());
}

void setup() {

    Serial.begin(9600);
    setup_wifi();
    setupMQTT();
}

void reconnect() {
    Serial.println("Connecting to MQTT Broker...");
    while (!mqttClient.connected()) {
        Serial.println("Reconnecting to MQTT Broker..");
        String clientId = "ESP32Client-";
        clientId += String(random(0xffff), HEX);

        if (mqttClient.connect(clientId.c_str(), "group1", "FjeNSdBV")) {
            Serial.println("Connected.");
            mqttClient.subscribe("challenge4/1/token");
        }

    }
}

void loop() {
    if (!mqttClient.connected())
        reconnect();
    mqttClient.loop();
}

void callback(char* topic, byte* payload, unsigned int length) {
    Serial.print("Callback - ");
    Serial.print("Message:");
    for (int i = 0; i < length; i++) {
        Serial.print((char)payload[i]);
    }
}

```

- A quoi sert la fonction callback? (Quand est ce qu'elle est appelée? Ou est elle appelée?)
  - Elle est appelé après avoir subscribe à un topic et que quelqu'un publish sur ce topic, elle n'est pas appelé mais assignée avec le `mqttClient.setCallback(callback);`
- Si mon serveur avait tourné sur gdp2.devinci.fr et sur le port 8883, qu'aurais-je du changer?
  - `mqttClient.setServer(mqttServer, mqttPort);`
- S'ils ont une fonction reconnect :
  - A quoi sert-elle? (Quand est ce qu'elle est appelée? Ou est elle appelée?)

Sinon:

- Que se passe-t-il quand vous êtes déconnecté du brocker MQTT? Comment y remédier? (spoiler: une fonction reconnect et )
- A quoi sert la ligne `mqttClient.loop();`?
  - A attendre après un subscribe

#### S'ils n'ont pas réussi challenge 4:

- Expliquer votre code, pourquoi ça ne marche pas?

## Challenge 5

```
// DHT11 setup
#include "DHT.h"
#define DHTPIN 2      // D4
#define DHTTYPE DHT11 // DHT 11
DHT dht(DHTPIN, DHTTYPE);

// wifi and mqtt setup
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
const char* ssid = "****";
const char* password = "*****";
WiFiClient wifiClient;
PubSubClient mqttClient(wifiClient);
char *mqttServer = "****";
int mqttPort = 1883;

const char* mqtt_user = "admin";
const char* mqtt_password = "*****";

const char* temp_topic = "challenge5/0/temp";
const char* hum_topic = "challenge5/0/hum";

// For wifi and mqtt see chall 4

void setup() {

  Serial.begin(9600);
  setup_wifi();
  setupMQTT();
  Serial.println(F("DHTxx test!"));

  dht.begin();
}

void reconnect() {
  Serial.println("Connecting to MQTT Broker...");
  while (!mqttClient.connected()) {
    Serial.println("Reconnecting to MQTT Broker..");
    String clientId = "ESP32Client-";
    clientId += String(random(0xffff), HEX);

    if (mqttClient.connect(clientId.c_str(), mqtt_user, mqtt_password)) {
      Serial.println("Connected.");
      mqttClient.subscribe("challenge4/1/token");
    }
  }
}
```

```

void loop() {
    if (!mqttClient.connected())
        reconnect();
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    if (isnan(h) || isnan(t) ) {
        Serial.println(F("Failed to read from DHT sensor!"));
        return;
    }
    mqttClient.publish(temp_topic, String(t).c_str());
    Serial.printf("Publishing on topic %s at QoS 1, packetId: %i ", temp_topic);
    Serial.printf("Message: %.2f \n", t);

    mqttClient.publish(hum_topic, String(h).c_str());
    Serial.printf("Publishing on topic %s at QoS 1, packetId: %i ", hum_topic);
    Serial.printf("Message: %.2f \n", h);
    delay(15000);
}

void callback(char* topic, byte* payload, unsigned int length) {
    Serial.print("Callback - ");
    Serial.print("Message:");
    for (int i = 0; i < length; i++) {
        Serial.print((char)payload[i]);
    }
}

```

- Expliquez votre code (Focus sur la partie loop avec récupération des données et publish)

#### S'ils n'ont pas réussi le challenge:

- Quel est le problème?