



UNIVERSITAT^{DE}
BARCELONA

Pràctica 2: Capa física

Antoni Tuduri & Alejandro Guzman

10 octubre 2022

ÍNDICE

1	Introducció	3
1	Objectius	3
2	Materials i metodologia	4
1	Instal·lació	4
2	Verificació de funcionament	5
2.1	Blink test	5
2.2	Hello world	6
3	Anàlisi de la potència rebuda	6
3.1	Codi implementat	7
3.2	Obtenció dels valors RSSI experimentals	9
3.3	Construcció d'una taula comparativa	9
3.4	Atenuació del senyal	10
4	Connexió a una xarxa WiFi	10
3	Resultats	11
1	Obtenció dels valors RSSI experimentals	11
2	Obtenció del paràmetre η de l'equació de Friis	12
3	Atenuació del senyal	14
4	Conclusions	15

1. INTRODUCCIÓ

En aquesta pràctica treballarem més sobre la capa física de la torre OSI i del protocol TCP/IP, la qual és la capa que es troba al nivell més inferior tal i com es pot observar en la figura 1.1 corresponent al model OSI.

Amb aquest objectiu realitzarem diversos experiments amb un dispositiu hardware anomenat mota, amb el qual mesurarem la potència rebuda per part del transmissor d'una xarxa WiFi com per exemple a la que ens connectem quan ens trobem a casa, de fet aquesta xarxa és la que utilitzarem per realitzar les mesures pertinents.

Un cop obtinguts els valors experimentals, analitzarem la diferència entre aquests i els teòrics basant-nos en les lliçons explicades a teoria sobre l'equació de Friis i els seus components, especialment el valor de η .

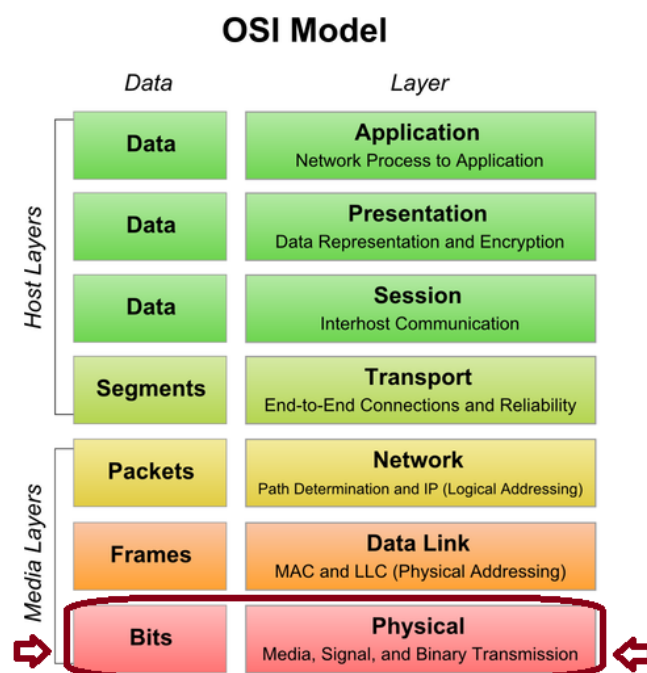


Figura 1.1: Model OSI i TCP/IP

1. Objectius

Els principals objectius de la segona pràctica de l'assignatura són:

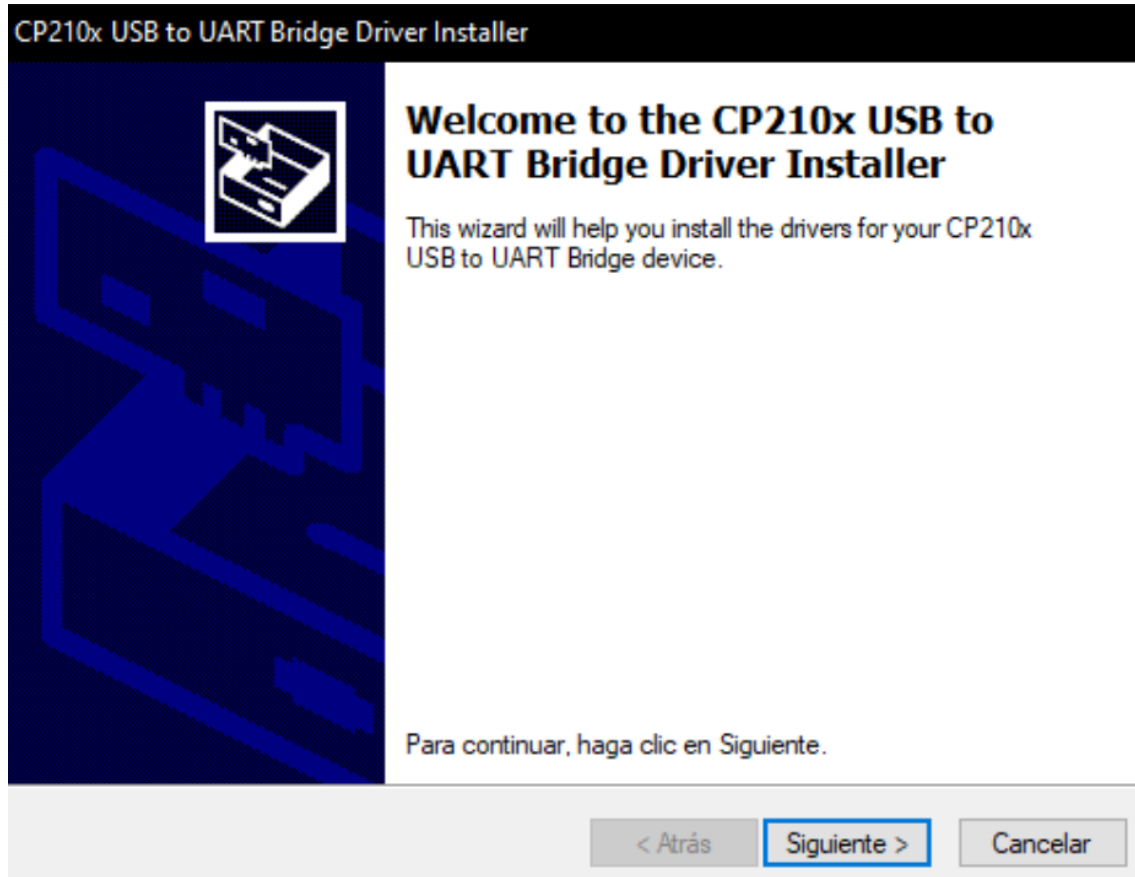
- Millorar la programació en Arduino
- Aprendre sobre les funcionalitats del dispositiu hardware proporcionat pel professorat.
- Valorar i analitzar la potència d'una xarxa wifi.

2. MATERIALS I METODOLOGIA

1. Instal·lació

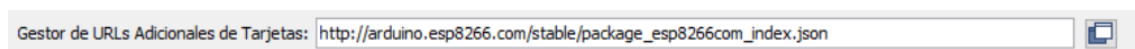
En aquest primer apartat ens dedicam en la instal·lació de tot lo necessari per poder fer la pràctica adequadament.

És necessari la instal·lació d'uns controladors en l'ordinador per crear una comunicació entre la UART del dispositiu encastat i l'USB que la uneix amb l'ordinador. El controlador corresponent serà el CP2104, el qual hem descarregat des de la pàgina web oficial.



És necessària la instal·lació d'un paquet de suport de placa per fer que aquest element de maquinari funcioni en el sistema operatiu de l'ordinador, fent les tasques d'inicialització dels components de la mota i l'execució del seu programa d'arrencada.

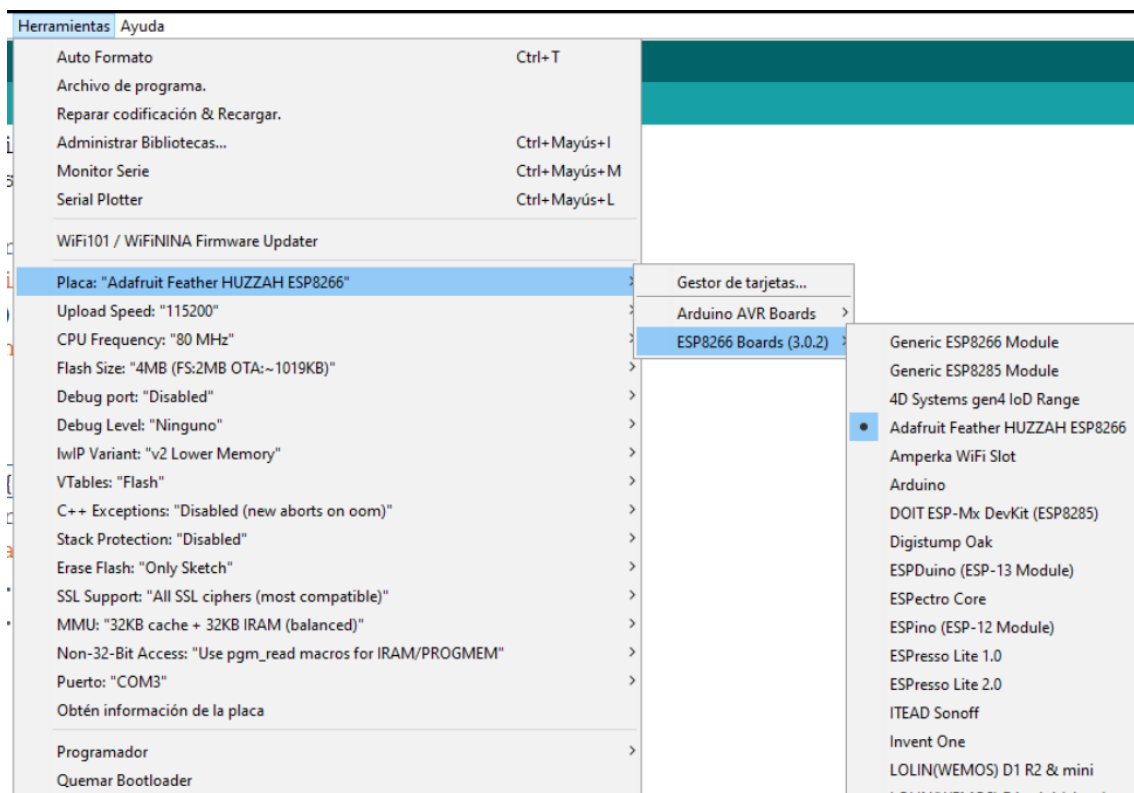
Aquest paquet es pot instal·lar copiant l'enllaç referent al document .json de configuració d'aquest mateix paquet en Preferències ->Gestor de URLs Adicionals de Targetes, dins de l'IDE d'Arduino, tal i com es mostra a la següent figura.



El següent pas és dirigir-nos a Eines ->Placa Arduino Uno ->Gestor de Targetes i instal·lar el paquet ESP8266.



Instal·larem aquest paquet i reiniciarem l'IDE de l'arduino per tal de assegurar que tot el que hem instal·lat s'implementi correctament dins del programa. Finalment seleccionarem la placa.



2. Verificació de funcionament

Abans de començar a realitzar els experiments que se'ns proposen en aquesta pràctica, hem de comprovar que efectivament la mota funciona correctament.

Amb aquest objectiu executem dos programes molt simples però efectius per saber si el dispositiu hardware funciona correctament.

2.1. Blink test

Carreguem el primer programa a l'Arduino i l'executem.

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);                     // wait for a second
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);                     // wait for a second
}
```

Figura 2.1: Codi del programa blink_test.ino

Aquest programa haurà d'encendre una llum de manera intermitent en la mota. En executar aquest codi podem comprovar com realment fa el que ha de fer, per la qual cosa aquesta prova ha estat completada amb èxit.

2.2. Hello world

El segon programa a carregar serà el típic Hello World, el codi del qual es mostra a continuació.

```
String incomingData;
boolean TransmisioCompleta = false;

void setup() {
  Serial.begin(9600);
  delay(1000);
  Serial.println("Hello world");
}

void loop() {
  if (Serial.available()) {
    Serial.print(" > ");
    Serial.println(Serial.readString());
  }
}
```

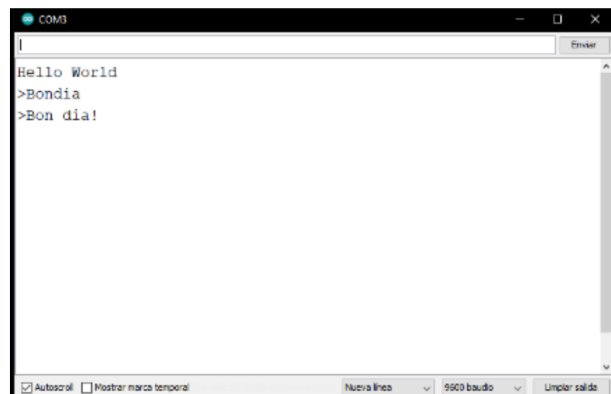


Figura 2.2: Codi del programa hello_world.ino

Tal com podem veure el programa fa el que s'esperava, el qual es imprimir per pantalla "Hello World", i altres missatges configurats per nosaltres, per la qual cosa ja tenim la mota configurada correctament.

3. Anàlisi de la potència rebuda

En aquesta part de la pràctica programarem la mota perquè aquesta pugui escanejar les xarxes WiFi pròximes i donar-nos informació sobre la seva RSSI, que consisteix en un indicador de la potència del senyal rebut. Aquesta magnitud es mesura en dBm i té una escala de valors enterament negatius, reflectint aquests una major pèrdua d'energia com més gran és el número negatiu, fins a arribar a -90 dBm que és el llindar.

En una escala de 0 a -80 RSSI:

- 0: senyal ideal, difícil d'aconseguir en la pràctica.
- -40 a -60: senyal idònia amb tasses de transferència estables.
- -60: enllaç bo ; ajustant la transmissió (Tx) es pot aconseguir una connexió estable al 80
- -70: enllaç mitjà-baix; es una senyal mes o menys bona tot i que poden patir problemes amb pluja i vents.
- -80: és la senyal mínima acceptable per establir la connexió; poden succeir caigudes que es tradueixen en talls de comunicació (perduda de trucades, perduda de dades), missatges SMS corruptes (illegibles), etc.

3.1. Codi implementat

A continuació implementarem un codi en Arduino per a poder escanejar les xarxes WiFi pròximes i donar-nos informació sobre la seva RSSI.

S'ha dividit el codi en dues clares funcions, la primera (setup) realitzant una configuració inicial i la segona (loop) encarregada d'analitzar continuament les xarxes de l'entorn.

De manera més explícita:

En la funció setup() es fa la configuració inicial de la mota, on establim la velocitat de transmissió a 9600 bauds amb la funció Serial.begin(9600). També establim el mode de connexió, que en aquest cas és en mode client o estació amb la funció Wifi.mode(WIFI_STA).

Després ens desconnectem del Wifi amb la funció Wifi.disconnect(). Tenint tot això configurat ens esperem 2000 ns amb la funció delay(2000).

En la funció loop() es on s'especifica el bucle principal de funcionament del programa. Aquí es fa un escaneig de totes les xarxes Wifi que hi ha en aquest moment i retorna el nombre de xarxes trobades, gràcies a la funció Wifi.scanNetworks(). En cas de no existir cap es mostra un missatge d'error, i, en el cas contrari s'itera sobre totes les xarxes trobades, mostrant per a cadascuna d'elles el seu identificador (SSID) i el seu RSSI i, si té xifratge, mostra "*". La finalitat d'implementar l'anteriorment esmentat programa és la de registrar els valors de RSSI que tenim en diversos punts del lloc en el qual realitzem la pràctica.

En la figura 2.3 es mostra el codi la funcionalitat del qual acabem de detallar.

```
#include "ESP8266WiFi.h"

void setup() {

    Serial.begin(9600);

    WiFi.mode(WIFI_STA);
    WiFi.disconnect();

    delay(2000);
    Serial.println("Setup done");

}

void loop() {

    Serial.println("Scan start");
    int n = WiFi.scanNetworks();
    Serial.println("Scan done");

    if (n == 0) Serial.println("no networks found");

    else {

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

            Serial.print(i + 1);
            Serial.print(WiFi.SSID(i));
            Serial.print(" ");
            Serial.print(WiFi.RSSI(i));
            Serial.print(" ");
            Serial.print((WiFi.encryptionType(i) == ENC_TYPE_NONE) ? " " :
" *");

            delay(10);

        }

    }

    Serial.println("*****");
    delay(5000);
}
```

Figura 2.3: Codi per mostrar les xarxes WiFi

3.2. Obtenció dels valors RSSI experimentals

L'objectiu principal en aquest punt de la pràctica és analitzar la diferència entre el RSSI en el lloc més pròxim possible al transmissor de WiFi i diverses posicions que s'allunyen gradualment d'aquest primer punt per tal de realitzar mesuraments per a analitzar la variació del valor de RSSI.

Aquest valor podem preveure per l'equació de Friis que disminuirà el seu valor si ens allunyem del transmissor WiFi. Es pot intuir directament observant la pròpia equació, on la distància de mesura (d) és inversament proporcional a la potència obtinguda (P_{RX}):

$$P_{RX} = P_{TX} \cdot G_{TX} \cdot G_{RX} \cdot \left(\frac{\lambda}{4 \cdot \pi \cdot d} \right)^2 \cdot \eta \quad (2.1)$$

En cas de que en un punt concret de l'espai on mesurem el RSSI donés un valor extremadament baix, s'hauria de plantejar la possibilitat d'instal·lar un repetidor de senyal amb l'objectiu d'augmentar-la en aquell punt (anomenat punt cec) i així augmentar la potència rebuda en punts on potser és més difícil de fer arribar la senyal des de el transmissor.

Un dels problemes que ens trobem és que la potència de transmissió es troba en mW segons RSSI, però volem el valor de la potència en dBm, per tant hem de fer una conversió d'unitats de tal manera que la fórmula 2.1 expressada en dBm, segons la definició de decibel

$$P_{TX[dBm]} = 10 \cdot \log_{10} \left(\frac{P_{TX[mW]}}{1mW} \right) = 10 \cdot \log_{10}(P_{TX[mW]}) \quad (2.2)$$

queda definida de la següent manera: $P_{RX[dBm]} = P_{TX} + G_{TX} + G_{RX} + 20 \cdot \log_{10} \left(\left(\frac{\lambda}{4 \cdot \pi \cdot d} \right)^2 \cdot \eta \right)$

on la longitud d'ona (λ) es pot definir com

$$\lambda = \frac{c}{f_c} \quad (2.3)$$

Unificant aquestes tres fórmules obtenim la potència de transmissió en dBm d'una manera més directa utilitzant la potència de transmissió en mW tal que

$$P_{RX[dBm]} = P_{TX} + G_{TX} + G_{RX} + 10 \cdot \log_{10}(P_{TX[mW]}) + 20 \cdot \log_{10} \left(\left(\frac{c}{4 \cdot \pi \cdot d \cdot f_c} \right)^2 \cdot \eta \right) \quad (2.4)$$

on definim les constants $P_{TX[mW]} = 1mW$, $G_{RX} = G_{TX} = 1$, $C = 3 \cdot 10^8$ m/s, $f_c = 2,4 \cdot 10^9$ Hz de forma que la fórmula 2.5 queda definida només en funció de la distància, cosa que simplificarà els càlculs a posteriori.

3.3. Construcció d'una taula comparativa

Tenint els valors del RSSI a diverses distàncies del transmissor de WiFi procedirem a col·locar-los en una taula, en la qual especificarem la distància a la qual hem pres la mesura, la mesura en si i la potència de recepció teòrica (amb $\eta = 1$), ja que desconexem el valor de η , per tant eliminem aquesta variable de la fórmula.

A partir dels resultats obtinguts, comparant el valor de la potència de recepció real i la teòrica, procedim a fer una gràfica comparativa de les dues variables mitjançant una regressió per a obtenir el factor de rendiment real.

3.4. Atenuació del senyal

Finalment realitzarem una gràfica per a visualitzar el nivell d'atenuació de la senyal en funció de la distància que separa el punt on prenem la mesura i el transmissor WiFi.

4. Connexió a una xarxa WiFi

El següent pas a realitzar és connectar-nos a una xarxa WiFi preseleccionada, dins de totes les disponibles.

Amb aquest simple objectiu l'enunciat proporciona el codi mostrat en les figures 2.2 i 2.3, amb el qual ens permet ingressar a una web introduint les credencials necessàries (SSID i contrasenya) i connectar-nos a una xarxa WiFi.

```
void connectToWifi(void) {  
  
    if (WiFi.status() != WL_CONNECTED) {  
  
        WiFi.begin(c_ssid, c_pwd);  
  
        while (WiFi.status() != WL_CONNECTED) {  
            delay(1000);  
            Serial.println("Connecting...");  
            contador++;  
  
            if (contador == 10) break;  
        }  
        Serial.println(WiFi.localIP());  
    }  
}
```

Figura 2.4: Codi del mètode connectToWifi

```

void wifiNetworkSelection(void) {

    if (WiFi.status() != WL_CONNECTED) {

        Serial.println("SSID? > ");

        while (!Serial.available());
        ssid = Serial.readString();
        Serial.print("SSID selected > ");
        Serial.println(ssid);
        delay(1000);
        Serial.print("Password? > ");
        while (!Serial.available());
        password = Serial.readString();
        Serial.print("PWD > ");
        Serial.println(password);
        delay(1000);
    }
}

```

Figura 2.5: Codi del mètode WifiNetworkSelection

3. RESULTATS

En aquesta secció es mostren els resultats dels experiments realitzats durant aquesta pràctica, detallats en l'apartat 2 de la mateixa.

1. Obtenció dels valors RSSI experimentals

A continuació es mostra la taula dels valors RSSI obtinguts, junt amb la distància on es trobàvem del transmissor i el valor RSSI que s'hauria de donar segons l'equació de Friis.

Distància	RSSI mesurat [dBm]	RSSI mesurat [mW]	P_{RX} amb $(\eta = 1)$ (dBm)
1	-42	$6,309 \cdot 10^{-5}$	-40.04
2	-50	10^{-5}	-46.06
3	-54	$3,981 \cdot 10^{-6}$	-49.58
4	-57	$1,995 \cdot 10^{-6}$	-52.08
5	-58	$1,584 \cdot 10^{-6}$	-54.02
6	-60	10^{-6}	-55.60
7	-64	$3,981 \cdot 10^{-6}$	-56.94
8	-70	10^{-7}	-58.11
9	-74	$3,162 \cdot 10^{-8}$	-59.13

Cuadro 3.1: Valors RSSI obtinguts

2. Obtenció del paràmetre η de l'equació de Friis

El següent pas després d'obtenir un conjunt de valors experimentals del RSSI de la nostra xarxa, llavors posem com a objectiu obtenir un valor aproximat de η conegut com el valor de rendiment.

Si ens fixem, tots els termes de la igualtat

$$P_{RX} = P_{TX} + G_{TX} + G_{RX} + 20 \cdot \log_{10} \left(\left(\frac{\lambda}{4 \cdot \pi \cdot d} \right)^2 \cdot \eta \right) \quad (3.1)$$

els hem calculat experimentalment, i per tant podem deixar η en funció d'una constant que anomenem ω , quedant una equació de la forma

$$P_{RX} = \omega \cdot \eta \quad (3.2)$$

Finalment, construïm una taula de valors de ω segons la distància (figura 3.2).

Distància	RSSI mesurat [mW]	Valor de ω
1	$6,309 \cdot 10^{-5}$	$9,89465 \cdot 10^{-5}$
2	10^5	$2,47366 \cdot 10^{-5}$
3	$3,981 \cdot 10^{-6}$	$1,09941 \cdot 10^{-5}$
4	$1,995 \cdot 10^{-6}$	$6,18416 \cdot 10^{-6}$
5	$1,584 \cdot 10^{-6}$	$3,95786 \cdot 10^{-6}$
6	10^6	$2,74851 \cdot 10^{-6}$
7	$3,981 \cdot 10^{-7}$	$2,01932 \cdot 10^{-6}$
8	10^7	$1,54604 \cdot 10^{-6}$
9	$3,162 \cdot 10^8$	$1,22156 \cdot 10^{-6}$

Cuadro 3.2: Valors RSSI y ω segons la distància

Com coneixem els valors experimentals de PRX i de ω , només fa falta fer una regressió lineal de PRX en funció de ω . Primer realitzarem una gràfica amb els valors experimentals de PRX i els d' ω per observar visualment la diferència tal i com es pot veure en la figura 3.1 i finalment aplicarem una regressió lineal per tal d'obtenir el valor de η .

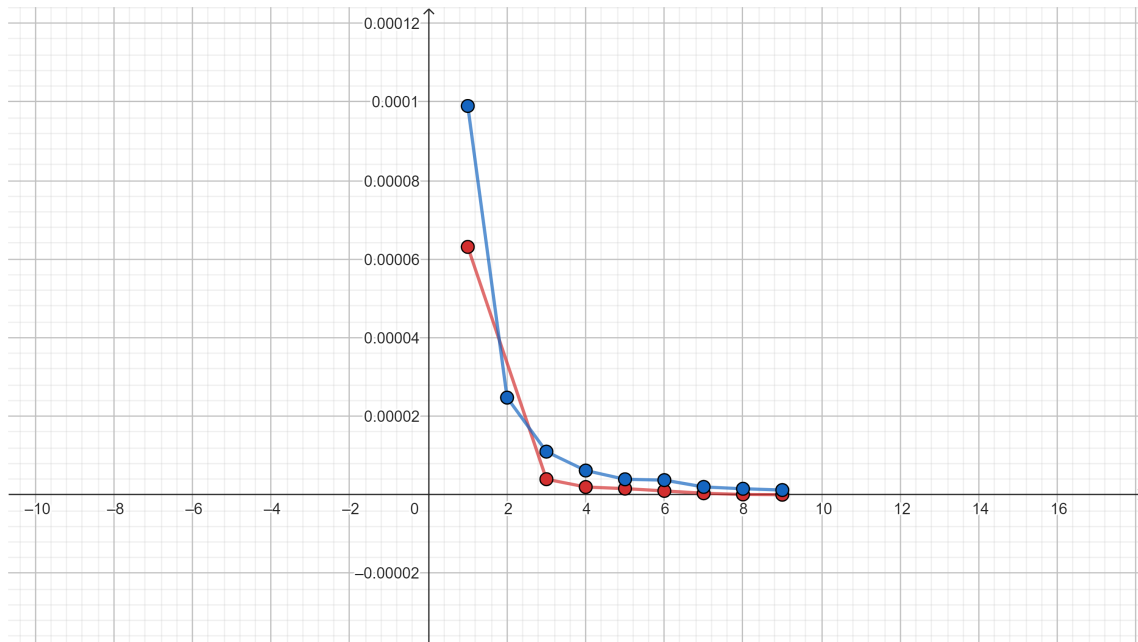


Figura 3.1: Valors de RSSI experimental (vermell) i PRX teòrica (blau) segons distància

Cal comentar que per realitzar la regressió és important tenir cura amb les unitats i en el nostre cas passarem totes les potències a mW. Llavors ara fem un gràfic de la regressió lineal dels valors de ω amb el resultat mostrat en la figura 3.1

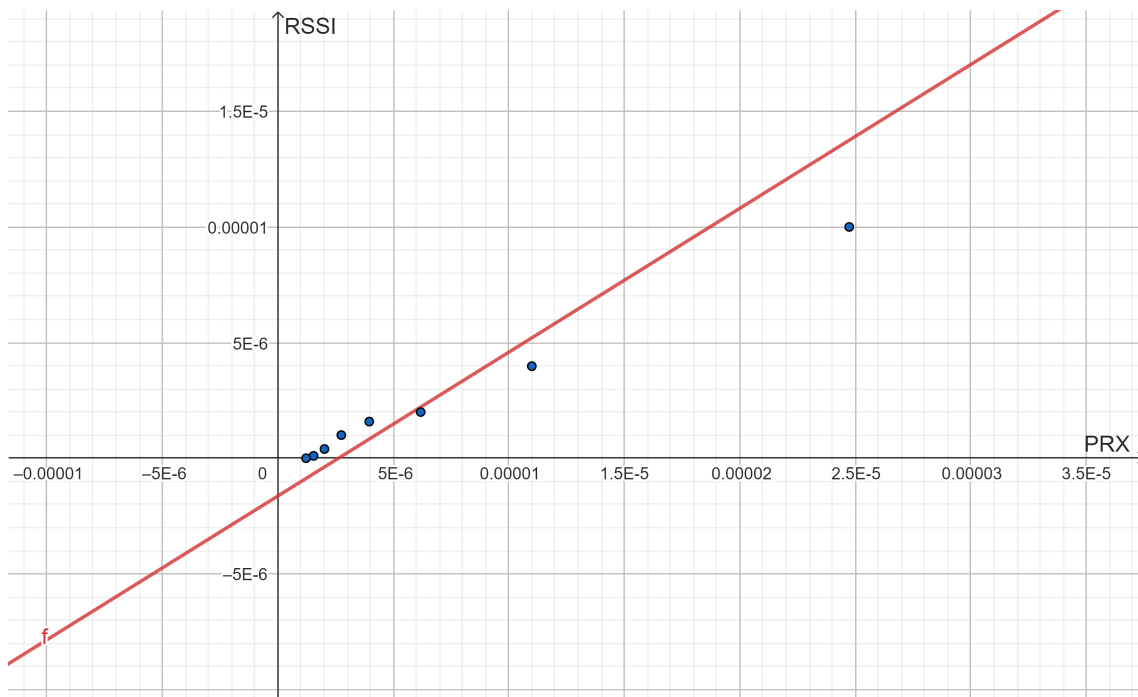


Figura 3.2: Regressió dels valors RSSI en funció de ω

La funció resultat és $f(\omega) = -1,634 \cdot 10^{-6} + 0,622 \cdot \omega$, per tant per tal d'obtenir una funció de la forma $f(x) = a \cdot x$ sense terme independent, suposem que $\eta = 0,622$ amb un error de la magnitud del terme independent degut a que l'eliminem de l'equació, per tant quedaria $\eta = 0,622 \pm 10^{-5}$

3. Atenuació del senyal

Recordem que la fórmula de la atenuació del senyal en un medi no físic és

$$Atenuació = 10 \cdot \log_{10} \left(\frac{P_{entrada}}{P_{sortida}} \right) [dB] \quad (3.3)$$

on $P_{entrada} [mW]$ és la potència calculada experimentalment i $P_{sortida}$ és una potència designada com $P_{TX} = 1 \text{ mW}$.

Llavors calculem l'atenuació del senyal en cada punt que hem calculat el RSSI mostrant-la en una taula i, posteriorment, realitzant una gràfica d'aquest augment d'atenuació a mesura que ens allunyem del transmissor.

Distància [m]	RSSI mesurat [mW]	Atenuació [dB]
1	$6,309 \cdot 10^{-5}$	-42
2	10^5	-50
3	$3,981 \cdot 10^{-6}$	-54
4	$1,995 \cdot 10^{-6}$	-57
5	$1,584 \cdot 10^{-6}$	-58
6	10^6	-60
7	$3,981 \cdot 10^{-7}$	-64
8	10^7	-70
9	$3,162 \cdot 10^8$	-74

Cuadro 3.3: Valors RSSI y ω segons la distància

Com és evident, si la potència de sortida és 1 mW llavors $Atenuació = P_{RX} = \text{RSSI}$ obtingut [dBm].

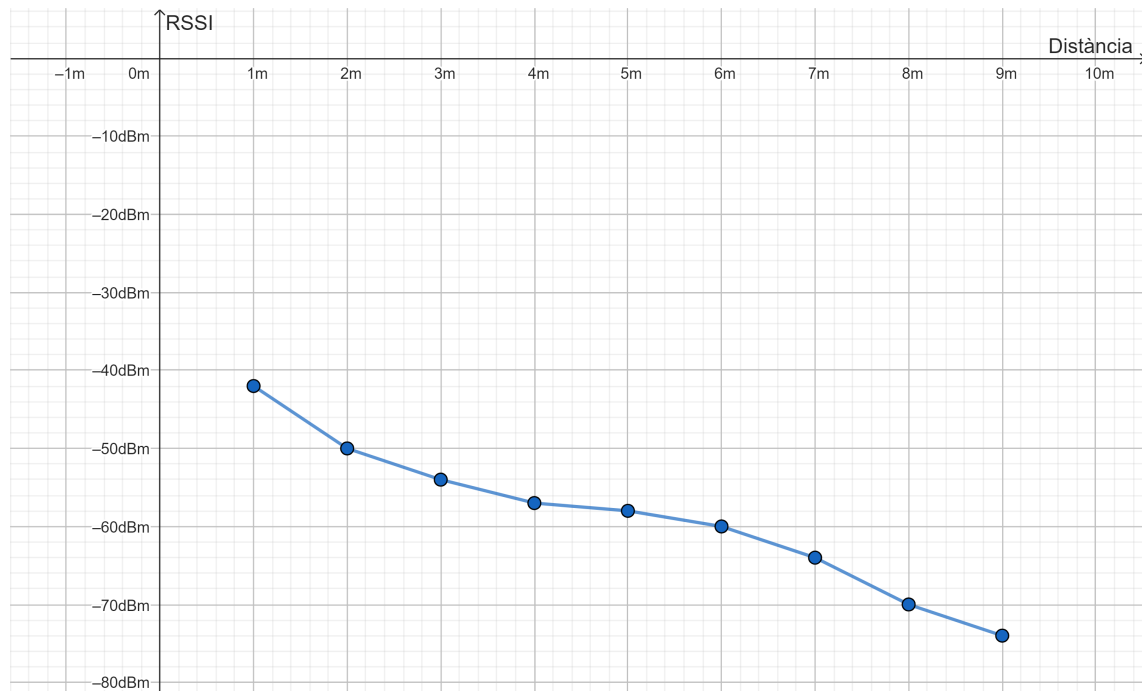


Figura 3.3: Atenuació del senyal en funció de la distància

4. CONCLUSIONS

La segona pràctica de la assignatura ens ha semblat molt interessant, ja que hem après, no només a desenvolupar més els coneixements que teníem sobre Arduino, sinó a utilitzar una eina hardware per a mesurar la potència de la xarxa WiFi de casa, lo qual ens sembla una cosa molt útil que es pot extrapolar a una situació quotidiana on volem comprovar com de bo és el WiFi de la nostra casa (o de qualsevol lloc) en algun lloc per tal de detectar punts cecs i actuar per millorar la connectivitat de la xarxa, com per exemple amb repetidors de senyal.