



UNIVERSITAT^{DE}
BARCELONA

Pràctica 2: Comunicacions de dades. La Capa Física

Noah Márquez & Jan Morales

30 octubre 2022

ÍNDIX

1	Introducció	3
1	Objectius	3
2	Configuració del dispositiu	5
1	Instal·lació	5
2	Verificació de funcionament	6
2.1	Blink test	6
2.2	Comunicació amb la Mota	7
3	Anàlisi de la potència rebuda	8
3.1	Codi implementat	8
3.2	Obtenció dels valors RSSI experimentals	11
3.3	Construcció d'una taula comparativa	12
3.4	Atenuació del senyal	12
4	Connexió a una xarxa Wi-Fi	12
3	Resultats obtinguts	13
1	Obtenció dels valors RSSI experimentals	13
2	Obtenció del paràmetre η de l'equació de Friis	15
3	Atenuació del senyal	18
4	Conclusions	20
5	Annexos	21
1	Codi exercici 1	21
2	Codi exercici 2	22

1 INTRODUCCIÓ

En aquesta pràctica treballarem més sobre la capa física de la torre del model OSI i del protocol TCP/IP. La capa física 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.

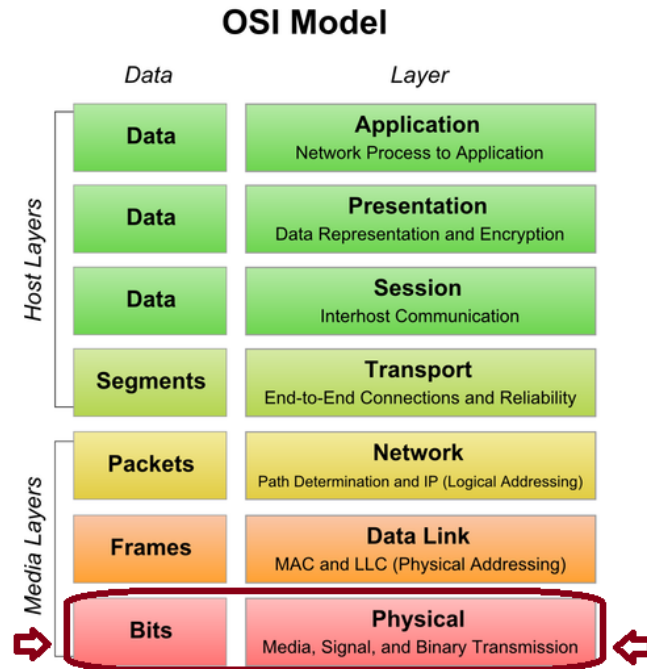


Figura 1.1: Model OSI i TCP/IP

Llavors, el principal objectiu d'aquesta pràctica és conèixer i aplicar els paràmetres principals associats a la capa física:

- Ample de banda i velocitat de transmissió.
- Cobertura de la xarxa i atenuació.
- Interferències associades a altres xarxes i soroll (Relació senyal/soroll).

Per assolir aquests objectius, haurem de programar un dispositiu integrat basat en **Arduino**, per tal que transmeti una sèrie de dades i camps de control al PC, i avaluar d'aquesta forma les característiques de la capa física.

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 Wi-Fi 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 .

1 Objectius

Podem llistar uns objectius primordials per aquesta segona pràctica:

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

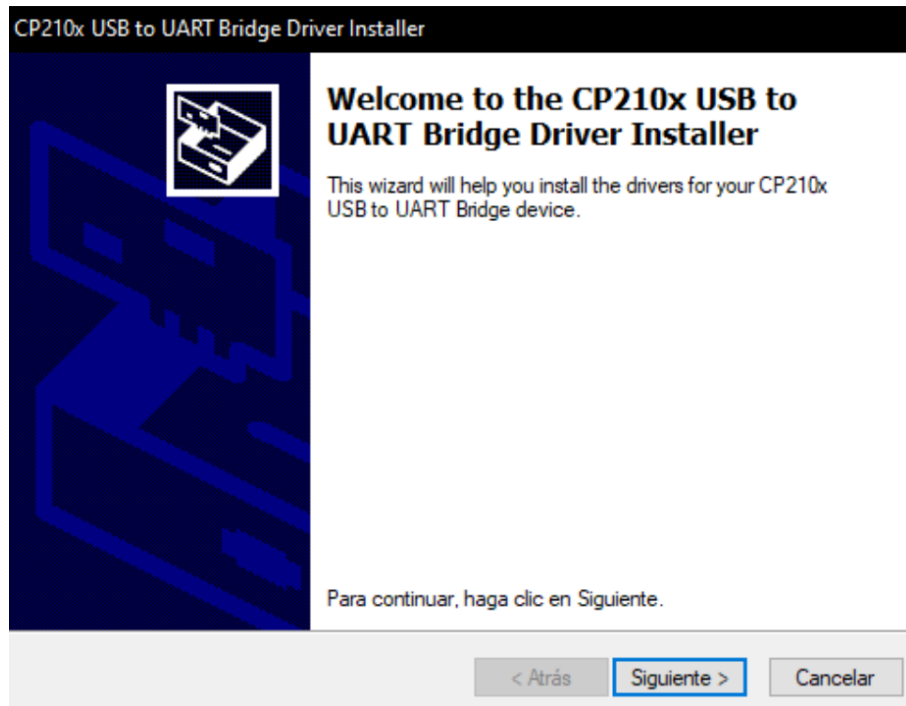
Tots els codis que fem servir durant la pràctica s'adjuntaran en format d'imatge per tal de facilitar la lectura mentre es llegeix el document. A més, també s'adjuntaran a l'annex en format de text.

2 CONFIGURACIÓ DEL DISPOSITIU

1 Instal·lació

En aquest primer apartat ens dediquem a la instal·lació de tot el necessari per poder fer la pràctica adequadament.

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



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

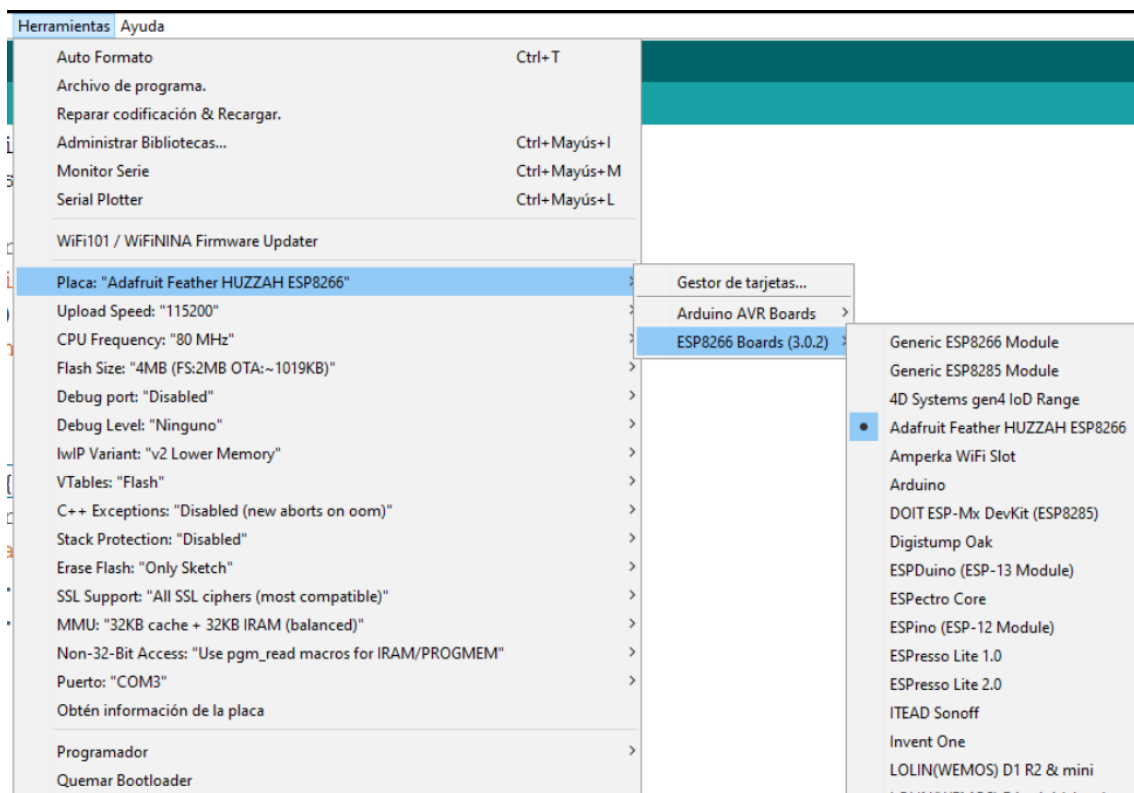
Aquest paquet es pot instal·lar copiant l'enllaç referent al document .json de configuració d'aquest mateix paquet en Arxiu -> Preferències -> Gestor de URLs Addicionals de Targetes, dins de l'IDE d'**Arduino**, tal i com es mostra a la següent figura.



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 amb la que treballarem i el port a associat a les comunicacions amb el dispositiu.



2 Verificació de funcionament

Abans de començar a realitzar els experiments que se'ns proposen en aquesta pràctica, hem de comprovar que la Mota funcioni 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 ha d'encendre una llum de manera intermitent en la Mota. En executar aquest codi podem comprovar com realment fa el que s'esperava.

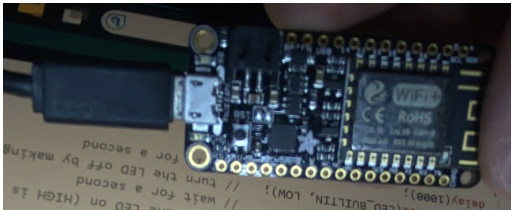


Figura 2.2: Llum apagat

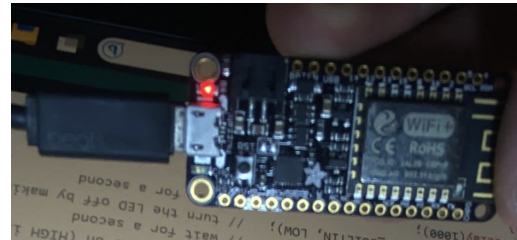


Figura 2.3: Llum encès

2.2 Comunicació amb la Mota

El segon programa és el que se'ns proporciona al document de la pràctica 2. Aquest codi ens ha de permetre parlar amb la mota des del terminal que proporciona el propi *Arduino IDE*. Per tal de comunicar-nos amb la Mota, hem de fer click a *Eines* i seleccionar *Monitor Sèrie*.

```
sketch_oct16a.ino ●
6 void setup() {
7   // put your setup code here, to run once:
8   Serial.begin(9600);
9   delay(1000);
10  Serial.println("Hello World");
11 }
12
13 void loop() {
14   // put your main code here, to run repeatedly:
15   if(Serial.available()){
16     Serial.println(" > ");
17     Serial.println(Serial.readString());
18   }
19 }
```

Output Serial Monitor x

Message (Enter to send message to 'Adafruit Feather HUZZAH ESP8266' on 'COM6')

>

Xarxes!!!

Figura 2.4: Codi del programa per comunicar-nos amb la Mota

Tal com podem veure el programa fa el que s'esperava, imprimir per pantalla el missatge que hem introduït.

3 Anàlisi de la potència rebuda

En aquesta part de la pràctica programarem la Mota perquè aquesta pugui escanejar les xarxes Wi-Fi 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, fins a arribar a *-80 dBm*, que és el llindar.

En una escala de 0 a *-80 RSSI*:

- **0**: senyal ideal, difícil d'aconseguir a 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 més 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 Wi-Fi pròximes i obtenir informació sobre la seva *RSSI*.

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

De manera més detallada:

- 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 Wi-Fi amb la funció *WiFi.disconnect()*. Tenint tot això configurat ens esperem 2000 ns amb la funció *delay(2000)*, abans d'imprimir el missatge *Setup done*.

- En la funció *loop()* és on s'especifica el bucle principal de funcionament del programa. Aquí es fa un escaneig de totes les xarxes Wi-Fi que hi ha en aquest moment aprop 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 ("*no networks found*"), 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 (contrasenya), mostra "*". La finalitat d'implementar aquest programa és la de registrar els valors de *RSSI* que tenim en diversos punts del lloc en el que realitzem la pràctica.

En la següent figura mostrem el codi que 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 2.5: Codi per analitzar la potència rebuda

Què fa aquest codi? Expliqueu detalladament el que veieu.

Anteriorment ja hem explicat el funcionament del codi, però ara ho fem després d'analitzar el que ens retorna el programa.

L'ESP8266 busca xarxes Wi-Fi properes i després imprimeix el *SSID* (nom) de cada xarxa, la seva *RSSI* (força de senyal), i si està assegurada amb xifratge (contrasenya). Quan comença, l'ESP8266 estableix el mode Wi-Fi i es desconnecta de qualsevol connexió anterior. En el bucle continua escanejant xarxes Wi-Fi cada 5 segons. La sortida representa el resultat d'escanejar les xarxes Wi-Fi properes a la placa. Cada línia representa una xarxa amb un número d'ordre.

És a dir, es mostra el nom (*SSID*) de la xarxa, per exemple, "wifi.ub.edu", "eduroam", "UBaux", etc. i un valor entre parèntesi que representa la força del senyal en *dBm* (decibels respecte al miliwatt). Valors més propers a zero indiquen una senyal més forta. Les xarxes protegides amb contrasenya, com hem comentat, tenen un asterisc.

```

Scan start          11wifi.ub.edu(-75)
Scan done           12wifi.ub.edu(-55)
1wifi.ub.edu(-76)    13eduroam(-55)*
2eduroam(-77)*       14UBaux(-55)*
3UBaux(-78)*         15UBaux(-72)*
4ESP8266(-67)*       16MOVISTAR_05F2(-80)*
5eduroam(-73)*       17wifi.ub.edu(-91)
6UBaux(-76)*         18wifi.ub.edu(-52)
7motaWiFi(-54)       19eduroam(-52)*
8wifi.ub.edu(-81)    20UBaux(-53)*
9eduroam(-79)*       21UBaux(-73)*
10UBaux(-79)*        22wifi.ub.edu(-74)

```

Figura 2.6: Output rebut del codi a l'analitzar les xarxes properes

Què és la *RSSI*? Expliqueu detalladament perquè podeu fer servir la *RSSI* i com es relaciona amb la qualitat del senyal.

El *RSSI* és un indicador que representa el nivell de potència que rep una antena. Sovint s'utilitza per estimar la qualitat del senyal en xarxes sense fil, per donar una idea aproximada de la força del senyal rebut. L'*RSSI* s'expressa generalment en decibels en relació amb un mil·liwatt (*dBm*). Un valor *RSSI* més alt (més proper a 0 *dBm*) indica un senyal rebut més fort, implicant una qualitat d'enllaç potencialment millor. Per contra, un valor *RSSI* més baix (més lluny de 0 *dBm*, p. ex. -75 *dBm*) indica un senyal més feble, el qual podria conduir a una connexió més lenta. És important recalcar que aquesta mesura pot ser afectada per factors ambientals com parets o obstruccions.

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 Wi-Fi (que serà el punt d'accés al nostre mòbil), i llavors en diverses posicions que s'allunyen gradualment d'aquest primer punt, per tal de realitzar mesures per 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 Wi-Fi. 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 el senyal des del 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 queda definida de la següent manera:

$$P_{RX_{dBm}} = P_{TX_{dBm}} + G_{TX_{dB}} + G_{RX_{dB}} - 20 \log_{10} \left(\frac{\lambda}{4\pi d} \right) \cdot \eta \quad (2.2)$$

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

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

i on definim les constants:

1. $P_{TX[mW]} = 1mW$
2. $G_{RX} = G_{TX} = 1$
3. $C = 3 \cdot 10^8$ m/s
4. $f_c = 2.4 \cdot 10^9$ Hz (2.4 GHz)

de forma que la fórmula (2.5) queda definida només en funció de la distància, i això simplificarà els càlculs.

3.3 Construcció d'una taula comparativa

Una vegada tenim els valors del RSSI a diverses distàncies del transmissor de Wi-Fi procedirem a col·locar-los en una taula, en la qual especificarem la distància a la qual hem pres la mesura. La mesura i la potència de recepció serà teòrica (amb $\eta = 1$), ja que desconexim el valor d' η , per tant obviem 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 del senyal en funció de la distància que separa el punt on prenem la mesura i el transmissor Wi-Fi.

4 Connexió a una xarxa Wi-Fi

Per fer tot el descrit anteriorment, necessitem primer connectar-nos a una xarxa Wi-Fi preseleccionada, dins de totes les disponibles. Nosaltres ens connectarem al punt d'accés del nostre mòbil.

Per aquest simple objectiu, se'ns proporciona el codi mostrat en les següents figures, amb els que podrem ingressar a una xarxa Wi-Fi introduint les credencials necessàries (SSID i contrasenya).

```

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 2.7: 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 2.8: Codi del mètode *wifiNetworkSelection*

```

Setup done
Scan start
Scan done
1ADAMO-F034(-76)*
2PRINKMFP2:0196(-83)
3ADAMO-F034(-76)*
4Domotica(-91)*
5iPhone de Jan(-42)*
6MOVISTAR_A898(-67)*
7ADAMO-57CF(-75)*
8MiFibra-300E(-89)*
9MOVISTAR_3840(-87)*
*****
SSID? > Password? > SSID: iPhone de Jan
PWD: ██████████
Connecting...
Connecting...
Connecting...
Connecting...
Connected successfully!
IP Address: 172.20.10.2

```

Figura 2.9: Realitzant la connexió al nostre punt d'accés

3 RESULTATS OBTINGUTS

En aquesta secció es mostren els resultats dels experiments realitzats durant el treball de camp, detallats en l'apartat 3.2 en endavant.

1 Obtenció dels valors RSSI experimentals

A continuació es mostra la taula dels valors *RSSI* obtinguts, junt amb la distància on ens trobàvem del transmissor i el valor *RSSI* que hauria de donar segons l'equació de *Friis* (les equacions utilitzades s'han donat anteriorment). Algunes de les proves les hem fet amb una paret entremig, per veure els diferents valors.

Distància [m]	RSSI mesurat [dBm]	RSSI mesurat [mW]	P_{RX} amb ($\eta = 1$) (dBm)
0.01	-31	$7.943 \cdot 10^{-4}$	-0.045
1	-42	$6.309 \cdot 10^{-5}$	-40.05
2	-50	10^{-5}	-46.07
3	-54	$3.981 \cdot 10^{-6}$	-49.59
4	-57	$1.995 \cdot 10^{-6}$	-52.08
5	-58	$1.584 \cdot 10^{-6}$	-54.03
5 (paret)	-64	$3.981 \cdot 10^{-7}$	-54.03
6	-60	10^{-6}	-55.61
7	-64	$3.981 \cdot 10^{-7}$	-56.95
8	-70	10^{-7}	-58.11
9	-74	$3.162 \cdot 10^{-8}$	-59.13
10	-78	$1.584 \cdot 10^{-8}$	-60.05
10 (paret)	-80	10^{-8}	-60.05

Taula 3.1: Valors RSSI obtinguts

Per tal de passar de *dBm* a *mW* ho hem fet amb la següent fórmula:

$$P(mW) = 10^{\frac{P(dBm)}{10}} \quad (3.1)$$

Els valors de les files amb paret no modifiquen P_{RX} , ja que la fórmula de Friis no considera les obstruccions com les parets. Hauríem d'incorporar la pèrdua de penetració de la paret per a una predicció més precisa. Els valors donats només representen un escenari d'espai lliure.

En una distància superior a 10 metres amb parets entremig només detectava a vegades la xarxa amb una potència ínfima. És en aquests punts, en els punts cecs, on caldria situar repetidors de senyal per tal d'arribar a tots els racons necessaris amb una potencia decent.

A continuació mostrem alguns dels outputs obtinguts:

Scan start	Scan start
Scan done	Scan done
1PRINKMFP2:0196(-65)	1PRINKMFP2:0196(-68)
2ADAMO-F034(-66)*	2ADAMO-F034(-66)*
3ADAMO-615F(-84)*	3ADAMO-615F(-84)*
4ADAMO-F034(-85)*	4MiFibra-300E(-81)*
5Salón.v,(-78)	5ADAMO-F034(-85)*
6iPhone de Jan(-31)*	6Salón.v,(-79)
	7MOVISTAR_A898(-39)*
	8ADAMO-57CF(-54)*
	9iPhone de Jan(-57)*
(a) 0.01 m	(b) 4 m
Scan start	Scan start
Scan done	Scan done
1PRINKMFP2:0196(-51)	1PRINKMFP2:0196(-55)
2ADAMO-F034(-76)*	2ADAMO-F034(-71)*
3ADAMO-F034(-91)*	3ADAMO-F034(-70)*
4MiFibra-6654-24G(-93)*	4ADAMO-615F(-88)*
5ADAMO-615F(-92)*	5MOVISTAR_1111(-92)*
6Salón.v,(-58)	6MiFibra-300E(-89)*
7iPhone de Jan(-64)*	7iPhone de Jan(-70)*
(c) 5 m (paret)	(d) 8 m

Figura 3.1: Potències rebudes amb el mòbil a diferents distàncies de la placa

2 Obtenció del paràmetre η de l'equació de Friis

El següent pas després d'obtenir un conjunt de valors experimentals d'*RSSI* de la nostra xarxa, és obtenir un valor aproximat d' η , conegut com el valor de rendiment.

Si ens fixem, tots els termes de la igualtat

$$P_{RX_{dBm}} = P_{TX_{dBm}} + G_{TX_{dB}} + G_{RX_{dB}} - 20 \log_{10} \left(\frac{\lambda}{4\pi d} \right) \cdot \eta \quad (3.2)$$

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

$$P_{RX} = \omega \cdot \eta \quad (3.3)$$

ω és el resultat de calcular tots els termes a l'esquerra de η . Per exemple, per una distància d'1 metre:

$$\begin{aligned} 1. \quad P_{RX} &= 0 + 0 + 0 - 20 \cdot \log_{10} \left(\frac{0.125}{4 \cdot \pi \cdot 1} \right) \cdot \eta \\ 2. \quad P_{RX} &= -40.05 \cdot \eta \\ 3. \quad 10^{\frac{-40.05}{10}} &\approx 9.89465 \cdot 10^{-5} = \omega \end{aligned} \quad (3.4)$$

Finalment, construïm una taula de valors de ω segons la distància:

Distància	RSSI mesurat [mW]	Valor de ω
0.01	$7.943 \cdot 10^{-4}$	$9.89465 \cdot 10^{-1}$
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	$1.584 \cdot 10^{-6}$	$3.95786 \cdot 10^{-6}$
5 (paret)	$3.981 \cdot 10^{-7}$	$3.95786 \cdot 10^{-6}$
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	$3.162 \cdot 10^{-8}$	$1.22156 \cdot 10^{-6}$
10	$1.584 \cdot 10^{-8}$	$9.89465 \cdot 10^{-7}$
10 (paret)	10^{-8}	$9.89465 \cdot 10^{-7}$

Taula 3.2: Valors RSSI y ω segons la distància

Com coneixem els valors experimentals de P_{RX} i de ω , només fa falta fer una regressió lineal de P_{RX} en funció d' ω . Primer realitzarem una gràfica amb els valors experimentals de P_{RX} i els d' ω , per observar visualment la diferència. Finalment aplicarem una regressió lineal per tal d'obtenir el valor d' η .

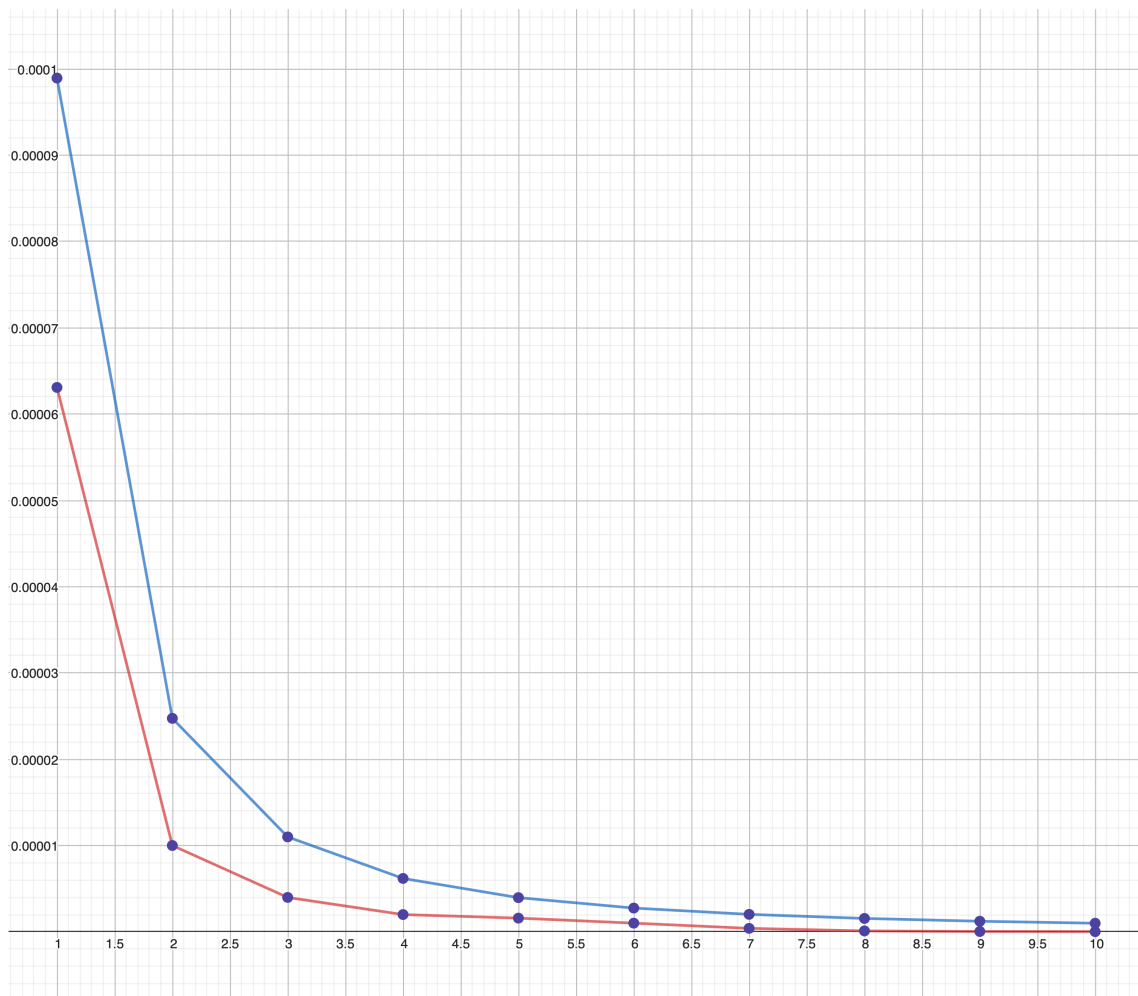


Figura 3.2: Valors d' $RSSI$ experimental (vermell) i P_{RX} teòrica (blau) segons distància (eix x)

Cal comentar que per realitzar la regressió és important tenir cura amb les unitats, és per això que hem convertit totes les potències a mW . A més, per tal de visualitzar el gràfic correctament, hem obviat el valor per una distància de 0.01 m.

Ara mostrem un gràfic de la regressió lineal dels valors d' ω :

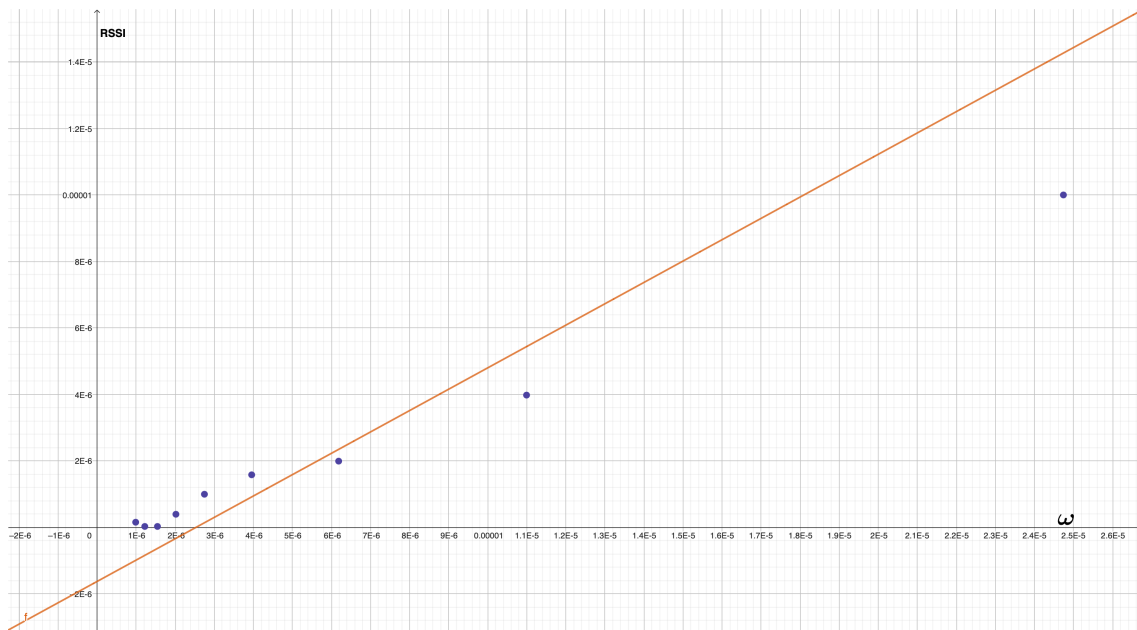


Figura 3.3: Regressió dels valors $RSSI$ en funció d' ω

La funció resultat és $f(\omega) = 1.623 \cdot 10^{-5} + 0.642 \cdot \omega$, per tant per tal d'obtenir una funció de la forma $f(x) = a \cdot x$ sense terme independent, suposem que $\eta = 0.642$ amb un error de la magnitud del terme independent degut a que l'eliminem de l'equació, per tant quedaria $\eta = 0.642 \pm 10^{-5}$.

$$\begin{aligned} f(x) &= \text{AjustePolinómico}(\text{EliminaIndefinidos}((x_1, y_1)), 1) \\ &= 0.6423689606341 x - 0.000001623253 \end{aligned}$$

Figura 3.4: $f(\omega)$ obtingut amb *GeoGebra*

3 Atenuació del senyal

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

$$\text{Atenuació} = 10 \cdot \log_{10} \left(\frac{P_{\text{entrada}}}{P_{\text{sortida}}} \right) [dB] \quad (3.5)$$

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

Llavors calculem l'atenuació del senyal en cada punt on hem calculat l' $RSSI$, 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]
0.01	$7.943 \cdot 10^{-4}$	-31
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
5 (paret)	$3.981 \cdot 10^{-7}$	-64
6	10^{-6}	-60
7	$3.981 \cdot 10^{-7}$	-64
8	10^{-7}	-70
9	$3.162 \cdot 10^{-8}$	-74
10	$1.584 \cdot 10^{-8}$	-78
10 (paret)	10^{-8}	-80

Taula 3.3: Valors RSSI i ω segons la distància

Com és evident, si la potència de sortida és 1 mW llavors: Atenuació = P_{RX} = RSSI obtingut [dBm]. A la figura següent obviem els valors amb parets entremig per claredat.

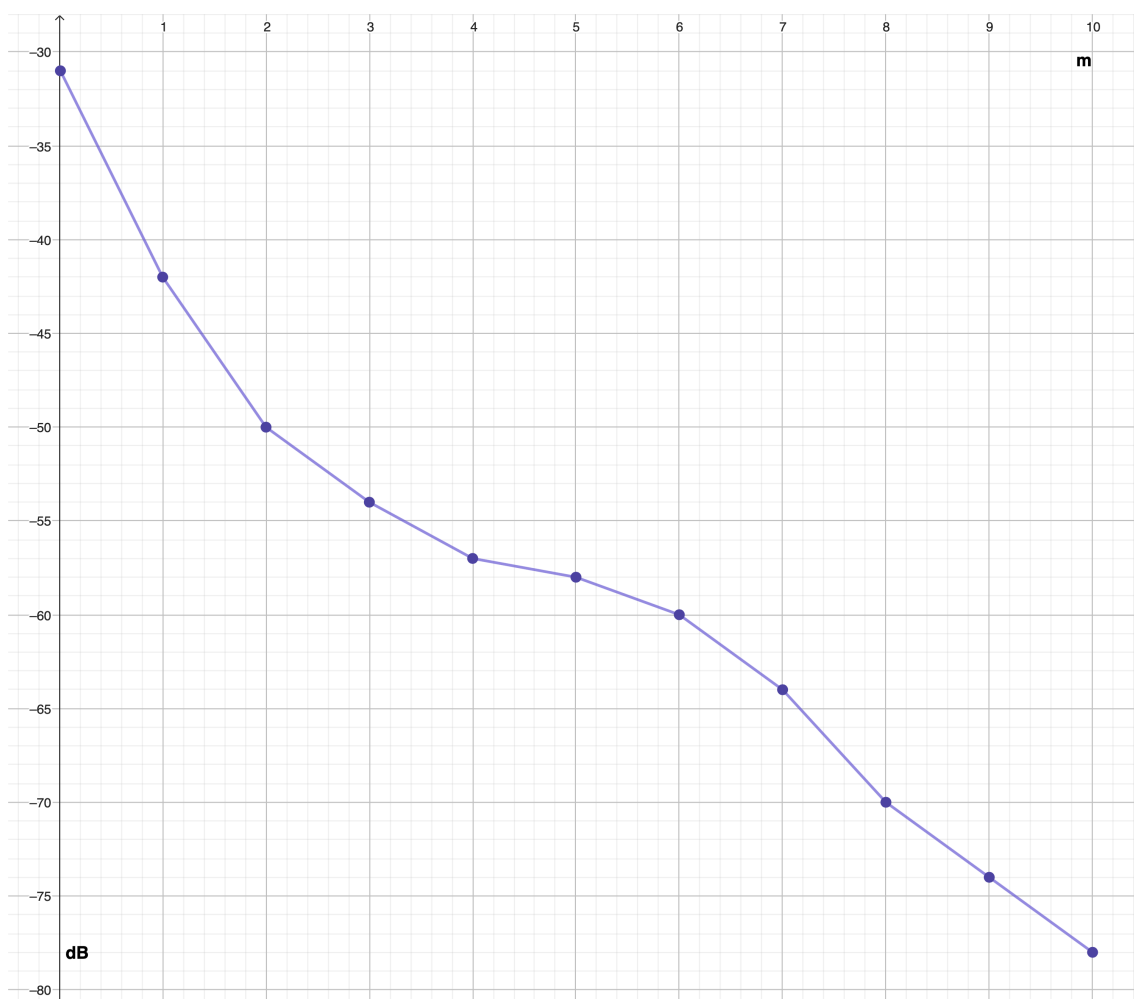


Figura 3.5: Atenuació del senyal en funció de la distància

4 CONCLUSIONS

Aquesta segona pràctica 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 rebuda de les xarxes Wi-Fi.

Ens sembla una cosa molt útil que es pot extrapolar a una situació quotidiana, on volguéssim comprovar com és el senyal Wi-Fi en diferents punts de la nostra casa (o de qualsevol altre lloc), com a detecció de punts cecs per decidir si és necessari o no utilitzar un repetidor de senyal, i actuar així per millorar la connectivitat de la xarxa.

5 ANNEXOS

1 Codi exercici 1

```
#include "ESP8266WiFi.h"

void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600); // Iniciem la comunicació serial a 9600
                        bps
    WiFi.mode(WIFI_STA); // Configurem el mode de Wi-Fi com a
                        estació i no com a punt d'accés
    WiFi.disconnect(); // Ens desconnectem de qualsevol altre
                        xarxa
    delay(2000); // Esperem dos segons
    Serial.println("Setup done"); // Escribim el missatge en el
                        serial
}

void loop() {
    // Aquest codi es repetirà amb un delay de 5 segons
    Serial.println("Scan start");

    int n = WiFi.scanNetworks(); // Escanegem les xarxes Wi-Fi
    Serial.println("Scan done");
    if(n==0) Serial.println("No networks found");
    else{
        // Per cada xarxa trobada escribim el seu SSID, el RSSI i
        // un asterisc si està protegida
        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.println((WiFi.encryptionType(i)==ENC_TYPE_NONE) ?
                " " : "*");
            delay(10);
        }
        Serial.println("**");
        // Wait a bit before scanning again
        delay(5000);
    }
}
```

2 Codi exercici 2

```
#include "ESP8266WiFi.h"

// Global Variables
char ssid[100];          // Emmagatzema l'SSID de la xarxa
char password[100];      // Emmagatzema la contrasenya de la
                          // xarxa
int contador = 0;        // Comptador per controlar el nombre d'
                          // intents de connexió

void setup() {
    Serial.begin(9600); // Inicia la comunicació serial a 9600
                        // bps
    WiFi.mode(WIFI_STA); // Configura el mode de la WiFi com a
                          // estació
    WiFi.disconnect(); // Es desconnecta de qualsevol xarxa WiFi
                        // anterior
    delay(2000);
    Serial.println("Setup done");
    scanNetworks(); // Cridem a la funció del exercici anterior
                  // que escaneja les xarxes wifi
    wifiNetworkSelection(); // Demana a l'usuari l'SSID i la
                          // contrasenya de la xarxa on es vol connectar
    connectToWiFi(); // Intenta connectar-se a la xarxa
}

void loop() {

}

void scanNetworks() {
    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.println((WiFi.encryptionType(i) == ENC_TYPE_NONE)
                          ? " " : "*");
            delay(10);
        }
    }
}
```

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

void connectToWiFi() {
    WiFi.begin(ssid, password); // Comença la connexió
                                utilitzant l'SSID i la contrasenya proporcionats

    while (WiFi.status() != WL_CONNECTED && contador < 10) { //
        Intentarem la connexió 10 cops
        delay(1000);
        Serial.println("Connecting...");
        contador++;
    }

    if (WiFi.status() == WL_CONNECTED) {
        Serial.println("Connected successfully!");
        Serial.print("IP Address: ");
        Serial.println(WiFi.localIP()); // Mostra l'adreça IP
                                         assignada
    } else {
        Serial.println("Failed to connect after 10 attempts.");
    }
}

void wifiNetworkSelection() {
    if (WiFi.status() != WL_CONNECTED) {
        Serial.print("SSID? > "); // Demana l'SSID
        while (!Serial.available());
        String ssidStr = Serial.readString();
        ssidStr.trim(); // Elimina espais en blanc innecessaris
        ssidStr.toCharArray(ssid, sizeof(ssid));

        Serial.print("Password? > "); // Demana l'SSID
        while (!Serial.available());
        String passwordStr = Serial.readString();
        passwordStr.trim(); // Elimina espais en blanc
                             innecessaris
        passwordStr.toCharArray(password, sizeof(password));

        Serial.print("SSID: ");
        Serial.println(ssid);
        Serial.print("PWD: ");
        Serial.println(password);
    }
}
```