

Pràctica 2. Comunicacions de dades. La capa Física

Objectius de la Pràctica

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 o a d'altres equips i avaluar d'aquesta forma les característiques de la capa física. Feu servir un ordinador portàtil per instal·lar l'IDE de **Arduino** sempre que sigui possible. La programació d'aquests dispositius (a partir d'ara Mota) es realitzarà en llenguatge C, amb l'entorn de treball que proporciona el propi **Arduino** (<https://www.arduino.cc/en/Main/Software>) o si ho preferiu podeu instal·lar-vos un pluguin de Eclipse (<http://www.baeyens.it/eclipse/>).

Les pràctiques amb **Arduino** es faran servir en diverses sessions. Es recomana als alumnes estudiar-se detalladament els manuals que podeu trobar a la web de Arduino, Huzzah i tota la informació proporcionada a classe i referent a aquesta mena de dispositius.

Primers passos amb les Motes

Abans de començar la sessió es farà una introducció al funcionament d'aquest tipus de dispositius, i amb aquest guió s'anirà guiant l'alumne en els primers passos. Obriu l'ordinador i escolliu la opció de Windows com a Sistema Operatiu.

El dispositiu amb el que anem a treballar és de molt baix consum. Està basat en un microcontrol·lador ATMEL AVR de 8 bits com a centre de gestió i control de la placa. El sistema de comunicacions es basa en un ESP8266, un xip Wi-Fi de baix cost que té integrada la pila TCP/IP completa i capacitat de MCU amb la que interactua amb el microprocessador ATMEL.

La figura 1 mostra la mota amb la qual treballarem durant aquestes sessions de pràctiques.



Figura 1. Dispositiu huzzah ESP8266-Wi-Fi

El dispositiu es programa mitjançant un connector microUSB que es connecta per una banda a la placa i per l'altra directament a l'ordinador, a través de un port USB. Aquest cable també serveix per alimentar el sistema a 5V des de l'ordinador.

Configuració del dispositiu

Podeu trobar com configurar la mota al següent enllaç:

<https://learn.adafruit.com/adafruit-feather-huzzah-esp8266/using-arduino-ide>

Els passos ha seguir són els següents:

1.- Instal·leu-vos el driver CP2104 al vostre ordinador. Als ordinadors de l'aula no caldria fer-ho. Caldrà descarregar-se el següent zip:

<https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers?tab=downloads>

i) Descomprimiu el fitxer i, sobre l'arxiu silabser.inf, cliqueu amb el botó dret i trieu l'opció instal·lar

2.- Cliqueu a *Arxius* -> *Preferències* -> *Gestor de URLs addicionals de targetes* i afegiu la línia:

http://arduino.esp8266.com/stable/package_esp8266com_index.json

3.- Seleccioneu *Eines* -> *Placa xxx* -> *Gestor de targetes* i carregueu els drivers *ESP8266 by ESP8266 Community*. Després caldrà reiniciar el IDE d'Arduino

4.- Obriu el IDE amb el que programareu el dispositiu. Configureu el tipus de dispositiu amb el que treballareu i el port a través del qual accediu a aquest. Feu un petit test del sistema.

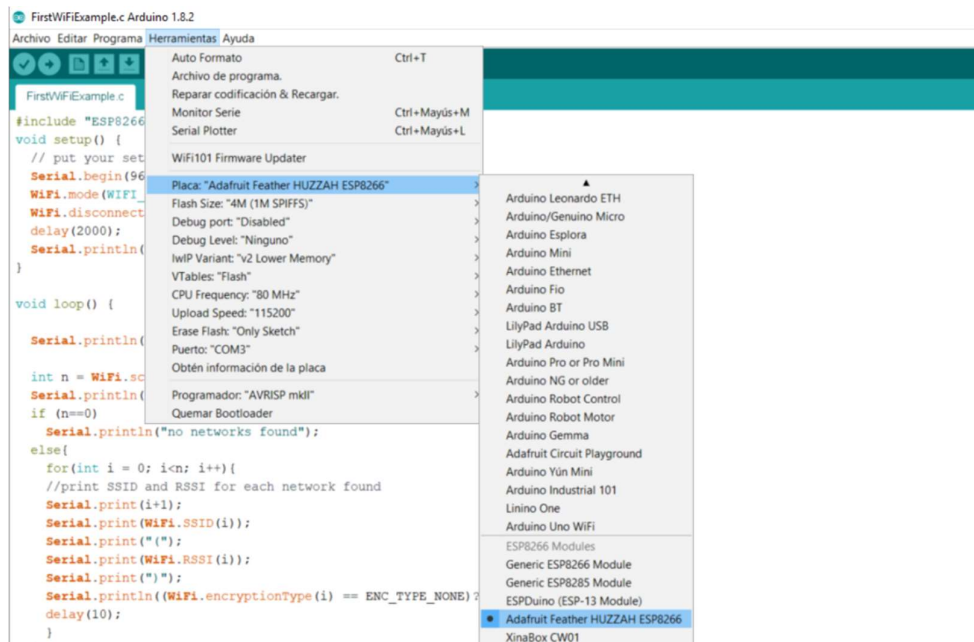


Figura 2

5.- Seleccioneu *Eines* -> *Port* i seleccioneu el port associat a les comunicacions amb el dispositiu.

6.- Testegeu com ha quedat tot plegat carregant el programa "*Blink test*" analitzeu el codi d'aquest programa. Tot seguit carreguem el codi (Figura 3) per validar la instal·lació del IDE. Ha de permetre parlar amb la mota des de el terminal que proporciona el propi Arduino IDE. En cas de problemes consulteu el professor



```
Archivo  Editar  Programa  Herramientas  Ayuda

HelloWorld

//#include "ESP8266WiFi.h"

String incomingData;
boolean TransmisioCompleta = false;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600); // open the serial port. 9600 b/s
  delay(1000);
  Serial.println("Hello World");
}

void loop() {
  // put your main code here, to run repeatedly:
  if(Serial.available()){
    Serial.print(" > ");
    Serial.println(Serial.readString());
    //serialEvent();
  }
}
```

Figura 3

Per tal de comunicar-vos amb la mota, cliqueu a *Eines* i seleccioneu *Monitor Sèrie*.

Anàlisis de la potència rebuda.

L'indicador de força del senyal rebut (*Received Signal Strength Indicator*) o RSSI és una escala de referència (en relació a 1 mW o 0dBm) per mesurar el nivell de potència dels senyals rebuts per un dispositiu en xarxes sense fils. L'escala té al valor 0 com a punt central, representa 0 RSSI o 0dBm. Tot i que teòricament pot donar-se el cas de mesurar valors positius, generalment l'escala s'expressa dins els valors negatius, indicant que quant més gran sigui el nombre negatiu, més gran és la pèrdua de senyal. El RSSI indica la intensitat rebuda, no la qualitat del senyal, ja que aquesta última es determina contrastant la intensitat del senyal respecte la relació senyal soroll (E_b/N_0)

Tot seguit comencem a operar amb la capa física. Genereu el següent codi en el vostre IDE:

```
#include "ESP8266WiFi.h"

void setup() {
  // put your setup code here, to run once:
  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++){
      //print SSID and RSSI for each network found
      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);
}
```

Figura 4

Què fa aquest codi? Expliqueu detalladament el que veieu.

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

Treball de camp

- 1.- Identifiqueu la xarxa WiFi de casa o del lloc on esteu fent la pràctica.
- 2.- Col·loqueu-vos el més a prop possible del transmissor WiFi on esteu connectats. Mesureu el RSSI
- 3.- Identifiqueu els punts on hi ha més connexió i els punts on hi ha menys connexió. Feu una taula explicativa on es vegi la variació RSSI - distància
- 4.- Identifiqueu si hi ha punts cecs.

(No es tracta de fer un mapa de casa vostra) Doneu una informació més genèrica. Com solucionaríeu el problema de punts cecs?). La relació més simple entre potència de recepció i distàncies ve donada per l'equació de *Friis*. Aquesta equació en format lineal s'expressa de la següent forma:

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

on η varia entre 0 i 1, $G_{TX} = G_{RX} = 1$, la freqüència de portadora $f = 2.4$ GHz, d és la distància en metres i $P_{TX} = 1$ mWatt. Programeu les motes amb el programa descrit anteriorment. Amb el metro, mesureu la distància en funció de la variació del RSSI treballant a la potència màxima (la que hi ha per defecte). Trebal·leu en línia recta en la mesura del que sigui possible per eliminar altres efectes secundaris. Tingueu en compte que en el punt límit de no recepció la potència de recepció serà aproximadament de -90 dBm. Determineu empíricament el mínim valor de RSSI a partir del qual ja no hi ha detecció. **Feu una taula amb tres columnes: Distància, RSSI obtinguda i la potència de recepció calculada a partir de l'equació anterior.** Inicialment, podeu considerar el factor $\eta = 1$. Feu la representació de la potència teòrica obtinguda en funció de la distància mesurada i poseu els punts del RSSI. A partir de l'expressió teòrica i les dades experimentals, feu també la regressió per tal d'obtenir el millor valor possible de η . Recordeu que η varia entre 0 i 1. **Tenint en compte tot això, expliqueu el que surt.** Representeu l'atenuació del senyal en funció de la distància a partir dels valors mesurats. La figura 5 mostra les especificacions tècniques del mòdul ESP que proporciona connectivitat WiFi al nostre dispositiu.

Product Specifications	
Module Model	ESP-12
Package	SMD-16
Size	24 x 16 x 3 (+/- 0.2)mm
SPI Flash	Default 32Mbit
Interface	UART/ HSPI/ IIC/ PWM/ ADC
IO Port	9
UART Baudrate	Support 300~460800 bps, Default 115200 bps
Frequency Range	2412~2484MHz
Antenna	PCB antenna, 2 dBi
Transmit Power	802.11b: 16 +/- dBm (@11Mbps) 802.11g: 14 +/-2 dBm (@54Mbps) 802.11n: 13 +/- dBm (@HT20, MCS7)
Receiving Sensitivity	CCK, 1 Mbps: -90dBm CCK, 11Mbps: -85dBm 6 Mbps (1/2 BPSK): -88dB 54 Mbps (3/4 64-QAM): -70dBm
Power (Typical Values)	Continuous Transmission=> Avg: ~71mA, Peak: 300mA Modem Sleep: ~20mA Light Sleep: ~2mA Deep Sleep: ~0.02mA
Security	WEP/WPA-PSK/WPA2-PSK
Power Supply	Voltage 3.0V~3.6V, Current>300mA
Operating Temperature	-20°C~85°C
Storage Environment	-40°C~90°C, <90%RH
Weight	0.45g
Certification	FCC, CE

Figura 5

(Ajuda per fer l'ajust demanat) Aprofitant els vostres coneixements en Python, podeu fer servir la llibreria:

```
from scipy.optimize import curve_fit
```

Una altra opció interessant pot ser:

```
from lmfit import minimize, Parameters, Parameter, report_fit
```

Finalment ens connectarem amb els dispositius a una xarxa WiFi. Obriu una WiFi des del vostre mòbil. Amb el codi anterior mireu de detectar-la. Un cop detectada, connecteu-vos a la xarxa. Podeu fer servir el següents codis.

1.- Codis per seleccionar una xarxa i connectar-se:

```
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());
    }
}

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 6

La funció *WiFi.begin(char_wifi, char_password)* funciona amb punters tipus char. En aquest codi que us passem la captura es guarda en un string. Podeu passar de String a char amb el següent codi:

```
void strTochar(String txt, char* c){
  if(WiFi.status() != WL_CONNECTED){
    int len = txt.length();
    Serial.println(len);
    Serial.println("*****");
    c = (char*)malloc(len);
    for(int i = 0; i<len; i++){
      c[i] = txt[i];
      Serial.print(c[i]);
    }
  }
}
```

Figura 7

Entregueu tots els codis que heu fet servir en aquesta pràctica comentats, així com les qüestions que es demanen explicades detalladament. El següent enllaç trobareu les diferents funcions que podeu fer servir amb un arduino com el de pràctiques.

<https://www.arduino.cc/reference/en/>