

INSTITUT PRÍNCEP DE VIANA

XARXA DETECTORA D'INCENDIS FORESTALS

Treball de Recerca

Saida Li Sillo Pérez

2017-2018

Jordi Orts González

ÍNDEX

1. INTRODUCCIÓ	
1.1. Elecció del tema	3
1.2. Objectius	4
1.3. Introducció al projecte	4
2. DISSENY	
2.1. Diagrama de blocs	5
2.2. Esquema electrònic WeMos D1 mini	6
2.3. Elecció dels components	
2.3.1. Estudi dels components	7
2.3.2. Disseny i muntatge del mòdul de sensors	11
2.4. Estudi de les xarxes WiFi	12
2.5. La xarxa interna	
2.5.1. Introducció a Arduino	12
2.5.2. Difusió: Mètode de la patata calenta	13
2.5.3. Disseny de la xarxa interna	13
2.5.4. Muntatge de la xarxa	14
2.6. Codi	18
2.7. Disseny i creació de les peces 3D	
2.7.1. Introducció a OpenSCAD	26
2.7.2. Disseny tècnic	27
2.7.3. Codi	30
2.8. Planificació	33
2.9. Pressupost	35
3. CONCLUSIONS	
3.1. Comparativa amb els objectius inicials	37
3.2. Possibles millores del projecte	38
3.3. Valoració personal	39
4. BIBLIOGRAFIA	
4.1. Fonts consultades	40

5. ANNEXOS

A.	Impacte mediambiental	41
B.	Estat actual a Espanya	44
C.	Grans incendis forestals a Espanya	46
D.	Sensors de flama	48
E.	És perjudicial el WiFi?	50
F.	Gos guardià (Watchdog)	51
G.	Versions del codi del mòdul 1	52
H.	Versions del codi del mòdul 2	53
I.	Versions del codi del mòdul 3	58
J.	Versions de la carcassa del mòdul de sensors	69
K.	Video del funcionament del prototip	71

1. INTRODUCCIÓ

1.1. Elecció del tema

He escollit l'àmbit tecnològic per dur a terme el meu treball de recerca ja que és un projecte al qual hauré de dedicar una gran part del meu temps, i com que la tecnologia ha estat sempre la meva assignatura preferida no vaig dubtar ni un sol moment en escollir-la. A més, des del minut zero vaig tenir claríssim que volia dedicar aquest treball al tema medi ambiental ja que és molt important per mi preservar tant la natura com les espècies que hi viuen en ella i tenint en compte que al llarg d'aquest curs hem après conceptes bàsics del disseny amb la màquina 3D i la programació amb Arduino IDE, vaig pensar que profunditzar en aquests temes i consolidar una mica més els meus coneixements podria ésser interessant. És necessari mencionar l'afició i admiració a les plaques ESP del meu tutor i professor de tecnologia ja que sempre ens ha parlat molt bé d'elles, raó per la qual volia provar-les en profunditat i comprovar l'eficàcia que ell assegura.

Després de molt de temps estudiant les grans possibilitats que m'oferia l'àmbit de la tecnologia aplicada a l'ecologisme vaig creure que fer un petit robot encarregat de netejar la superfície de les platges seria una bona idea però finalment vaig pensar que seria un repte massa complicat. Així doncs, després d'informar-me sobre les majors catàstrofes que afecten a la natura i als éssers vius vaig descobrir que els incendis forestals són una de les principals causes de la destrucció de la vegetació en tot el món i que són els causants de que cada any es cremin milers d'hectàrees a Espanya i tenint en compte les polítiques mediambientals del govern actual i que Espanya segueix estant a dia d'avui en una de les últimes posicions del rànquing de la UE en quant a la inversió econòmica pel medi ambient vaig arribar finalment a una conclusió definitiva i a un possible projecte al qual tinc clar que vull dedicar el meu temps.

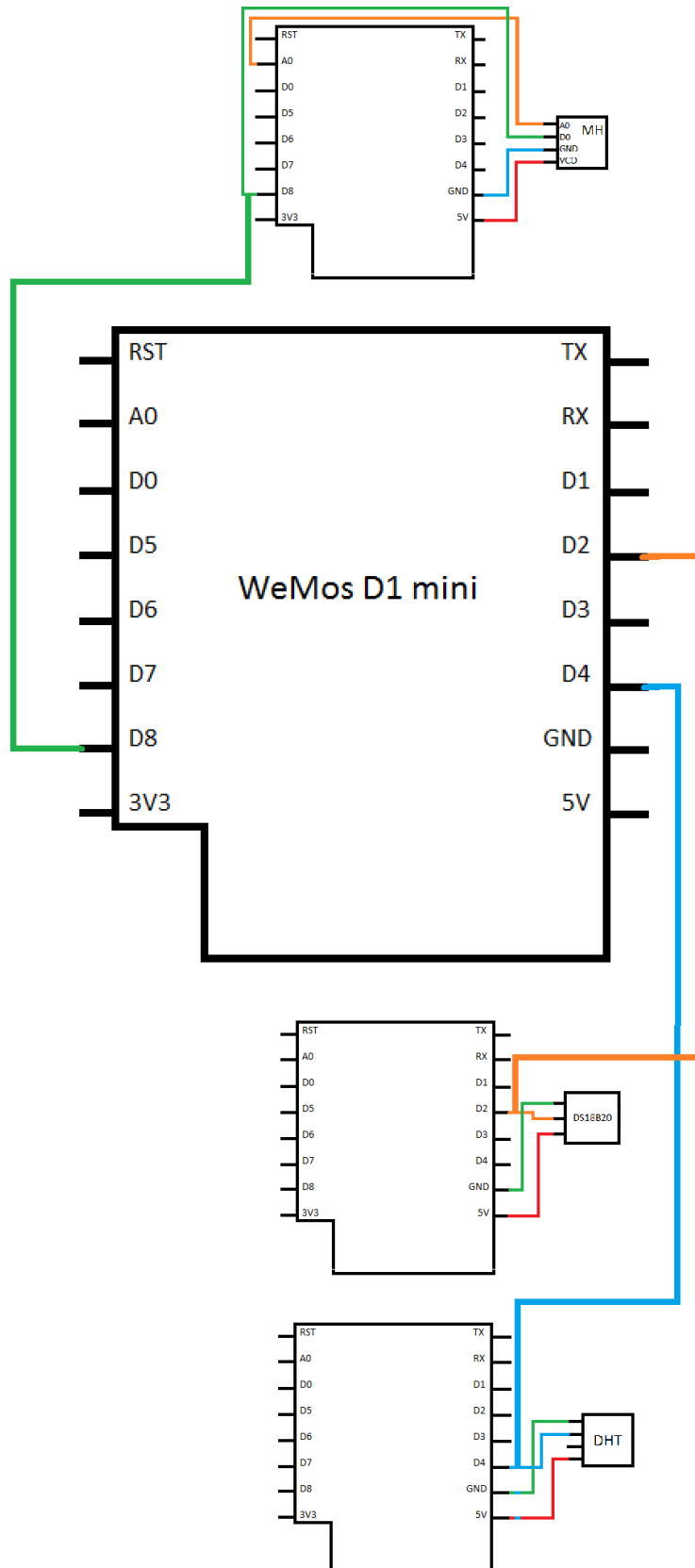
1.2. Objectius

Els principals objectius d'aquest treball són:

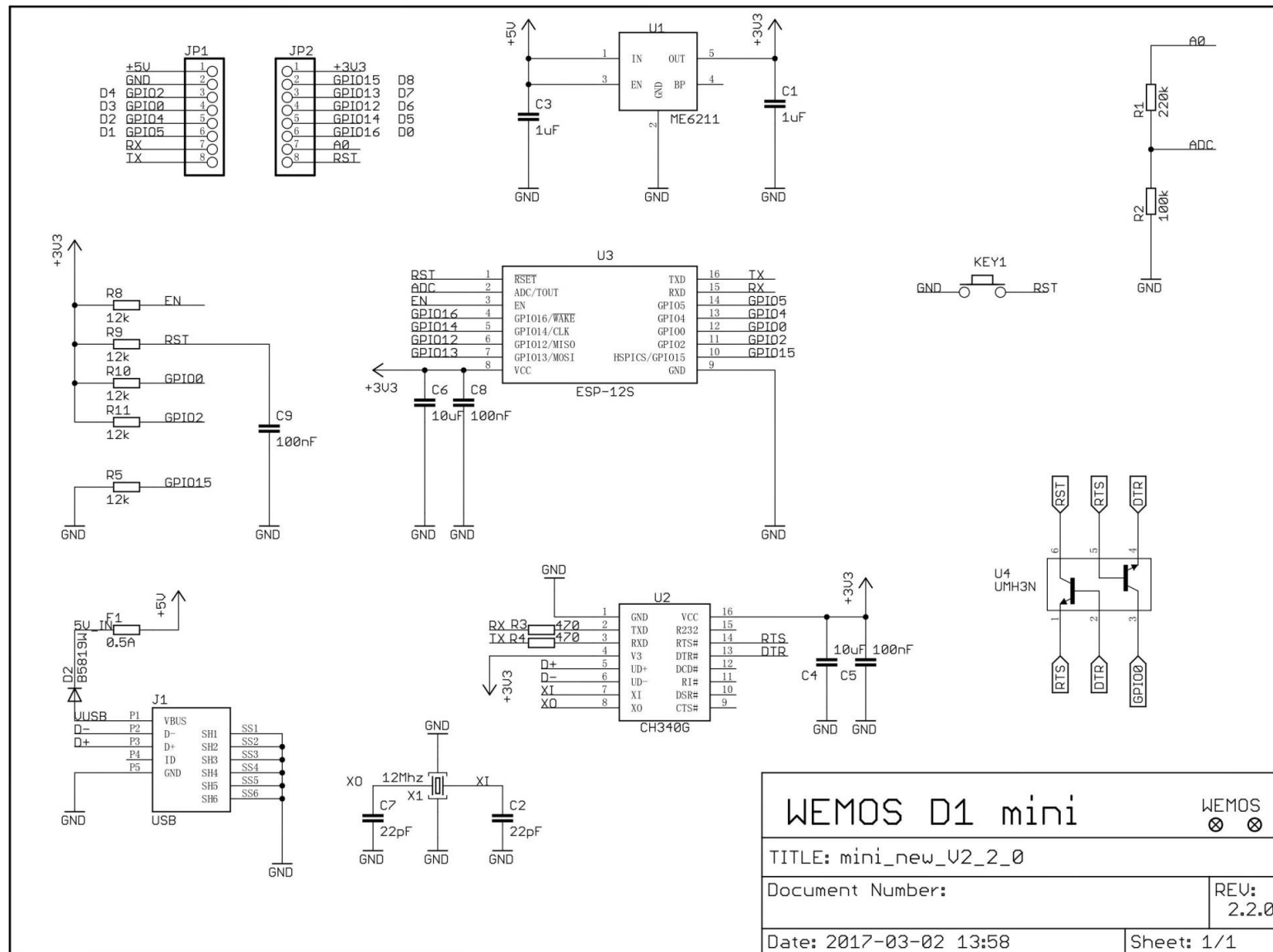
- Aprendre més en quant a les mesures de prevenció i extinció dels incendis forestals actuals, entendre'ls i jutjar per mi mateixa si aquestes són realment útils.
- Descobrir les raons per les quals a dia d'avui segueixen produint-se incendis de tals mesures als països desenvolupats
- Millorar les meves habilitats amb la programació Arduino i el disseny 3D
- Aprendre com funcionen les connexions sense cables i el WiFi intern
- Treure el màxim rendiment a les plaques ESP8266
- Crear un sistema automàtic de detecció d'incendis forestals`que no alterés l'ecosistema de manera notablement perjudicial pels éssers vius.
- Aconseguir unes conclusions totalment personals i basades en la meua experiència pròpia

1.3. Introducció al projecte

El projecte consta bàsicament d'un grup de sensors col·locats de manera que creen una xarxa connectada sense fils per la qual s'envia tota la informació necessària per localitzar els incendis forestals des dels seus inicis. És a dir, cadascun dels mòduls (sensors) té permanentment creat un punt d'accés, però la funció principal de cadascun és detectar l'augment de temperatura, la disminució de la humitat i la flama, i un cop detectada, es converteixen en els Clients dels mòduls del seu voltant, de manera que accedeixen als seus punts d'accés WiFi i envien la ubicació de l'incendi alhora que creen el seu propi punt d'accés al qual es podran connectar altres mòduls del voltant i així consecutivament fins que la informació arriba als punts claus de la xarxa on es troben els mòduls capaços no només de comunicar-se amb la xarxa interna del bosc sinó que permeten comunicar-se amb l'exterior, és a dir, enviar senyals als encarregats d'extingir el foc.

DISSENY**1.4. Diagrama de blocs**

1.5. Esquema electronic



1.6. Elecció dels components

Una de les parts més importants, per no dir la principal, per dur a terme el meu projecte va ser crear un llistat de necessitats, és a dir, seleccionar els components adequats, és per això que vaig haver de fer una recerca del ventall d'opcions i possibilitats que m'oferia ESP.

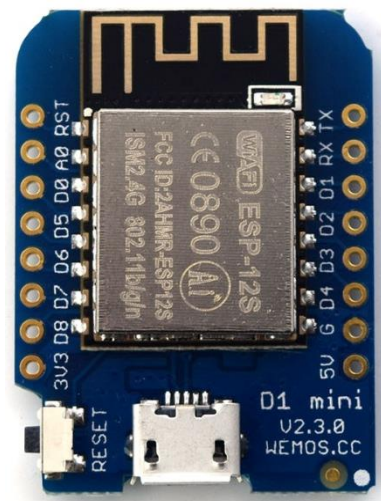
1.6.1. Estudi dels components

Des d'un primer moment vaig acordar amb el meu tutor que la manera de transmetre la informació seria via WIFI i que havia de ser un dispositiu que passés desapercebut a la naturalesa del bosc, és a dir, que no alterés en cap moment l'ecosistema en el qual s'instal·lés. Aquests dos punts ja dirigien molt el meu projecte.



Els xips ESP8266 han estat una autèntica revolució en el mercat dels WiFi econòmics i senzills. Es tracta, bàsicament d'un xip integrat amb connexió WiFi amb l'objectiu principal de donar accés a qualsevol microcontrolador a una xarxa amb un consum realment baix. Quan van aparèixer els primers ESP al 2014, el gran desavantatge

que presentaven era que tota la seva documentació estava escrita en xinès, però van començar a traduir-se poc a poc de manera que avui en dia, podem trobar moltíssima informació sobre aquests xips per aprendre a utilitzar-lo desde 0 i treure'n el màxim profit al gran ventall de possibilitats que ens ofereix. És per aquest motiu que vaig decidir sense dubtar en utilitzar-lo.



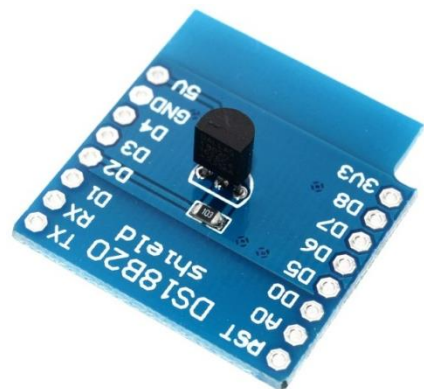
Com he mencionat, els mòduls havien de passar desapercebuts, una de les millor solucions per això és que els components siguin petits. Aquesta característica me la oferia WeMos D1 mini.

La placa WeMos D1 mini és una variant de la WeMos D1, podríem dir que és la seva germana petita, amb les mateixes possibilitats però un nombre menor de pins i sobretot una mida realment reduïda. És una placa governada per l'anteriorment presentat ESP8266 que s'encarrega del processament i el control de WiFi, el gran avantatge és que es pot programar amb l'entorn d'Arduino, i per tant és més senzill de programar, i és compatible amb qualsevol Shield de 3.3V

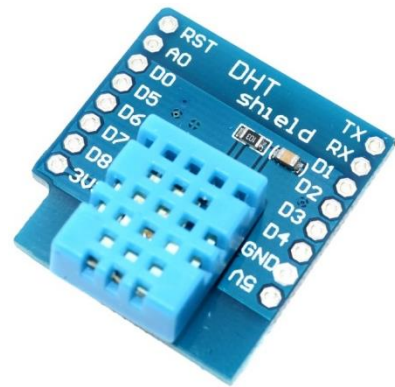
Un cop tenia la part principal, calia posar-me mans a la obra amb la resta dels components, que necessitaria per poder detectar un incendi, és per això que primer vaig fer un llistat de necessitats:

- El dispositiu ha de ser capaç de mesurar periòdicament la temperatura en la que es troba el bosc
- El dispositiu ha de saber quina és la humitat en tot moment
- El dispositiu ha de poder alimentar-se amb energia 100% verda
- El dispositiu ha de reconèixer el foc

Tot i tenir moltes opcions per mesurar temperatures. Jo vaig utilitzar la DS18B20 shield que és bàsicament una placa protagonitzada per un termòmetre electrònic que té una alta precisió ($\pm 0,5\text{ }^{\circ}\text{C}$) en un ampli rang (des de -10°C fins a $+85^{\circ}\text{C}$), converteix la temperatura a paraula digital de 12 bits en menys de 750 ms, pot alimentar-se desde la línia de dades, la franja de subministrament d'energia és d'entre 3V i 5.5V i és totalment compatible amb Wemos D1 mini



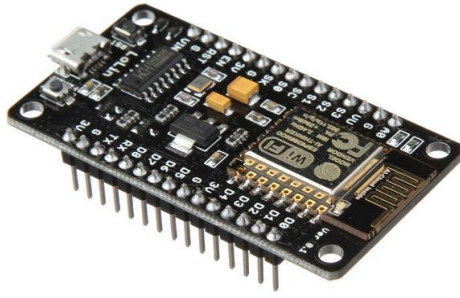
En quant a la mesura de la humitat vaig utilitzar un sensor molt semblant a un altre que ja havia utilitzat abans en un taller d'introducció a l'Arduino, l'escollit va ser el DHT shield que és un sensor digital de temperatura i humitat que utilitza el sensor DHT11. El seu rang de temperatura es de 0°C a 60°C i el d'humitat es del 20% al 90% amb una precisió de $\pm 5\%$ que és realment econòmic i com no, compatible amb Wemos D1 mini



Com a font d'alimentació vaig pensar en una placa fotovoltaica i és per això que vaig escollir la Battery shield on pensava connectar-la. Malgrat les meves intencions, aquest és un treball complex i el temps és limitat, per això, desenvoluparé aquesta part a l'apartat de possibles millores.

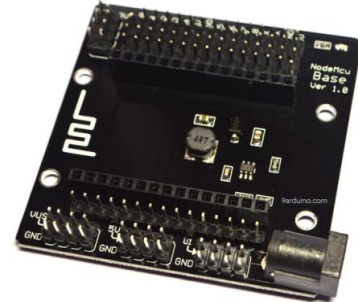
En la meua opinió, un dels components més interessants, del qual jo no coneixia existència, és sense dubte el KY-026 que és un sensor de flama òptic que permet detectar l'existència de combustió per la llum emesa, aquesta llum es detecta per un sensor òptic i es capturat per les entrades digitals i analògiques. És un component realment econòmic ja que la sensibilitat i fiabilitat d'aquests detectors no són suficients com per considerar-los autèntics dispositius de seguretat, de fet, pot ésser afectat inclús per la il·luminació interior, donant lloc a falsos positius però és molt interessant pel meu projecte ja que pot detectar, per exemple, la flama d'un encenedor. La millora d'aquest apartat és molt important de cara a una hipotètica reproducció, en parlo d'això a l'apartat de possibles millores.



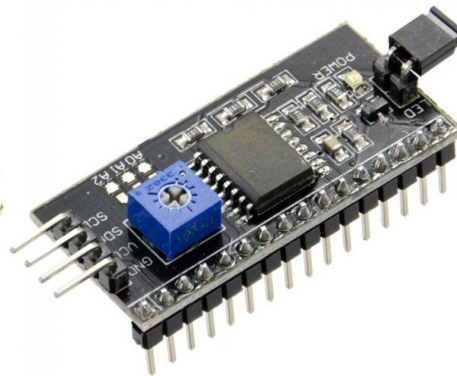
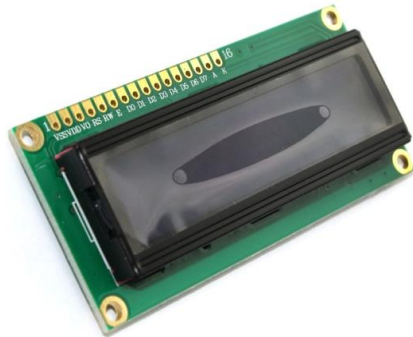


Per al receptor principal, el que enviarà les senyals a l'exterior tornaré a utilitzar el ESP8266, però aquest cop integrat al mòdul NodeMCU ja que permet programar el processador exactament com si fos un Arduino, el que em

facilita molt la feina ja que sempre he programat amb Arduino. Una altre avantatge és que tots els seus pins estan disponibles a l'exterior i inclou un connector mini USB per programar el xip intern. La variant NodeMCU que he utilitzat és la tercera generació del fabricant Lolin/WeMos.

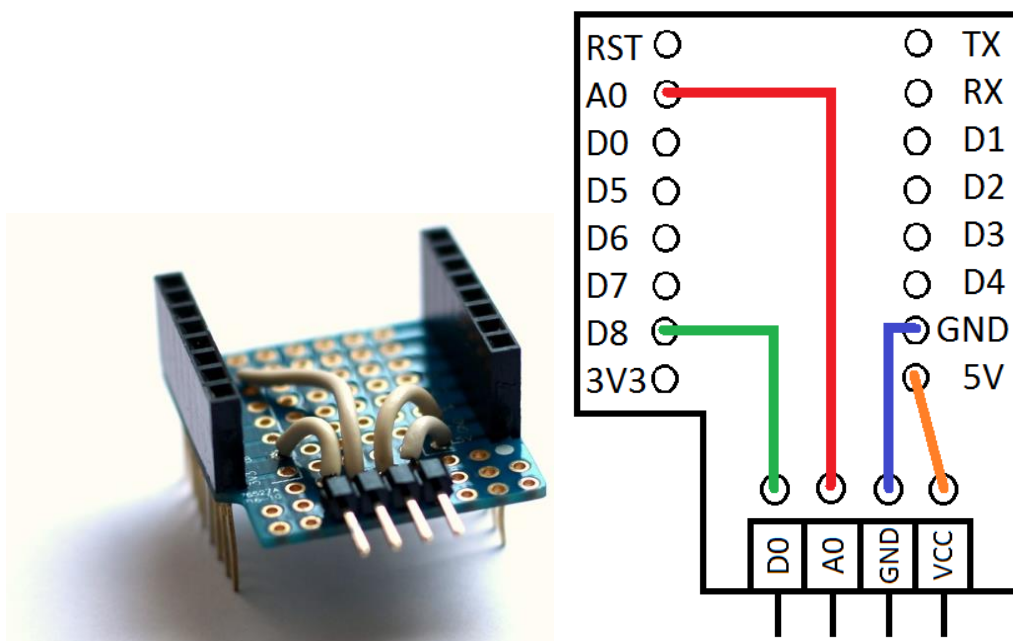


Per poder veure quin és el missatge que el receptor enviarà a l'exterior he utilitzat una pantalla LCD de 2x16 connectada a un mòdul conversor, que serveix per fer servir una LCD només amb dos fils mitjançant el protocol de comunicació I2C

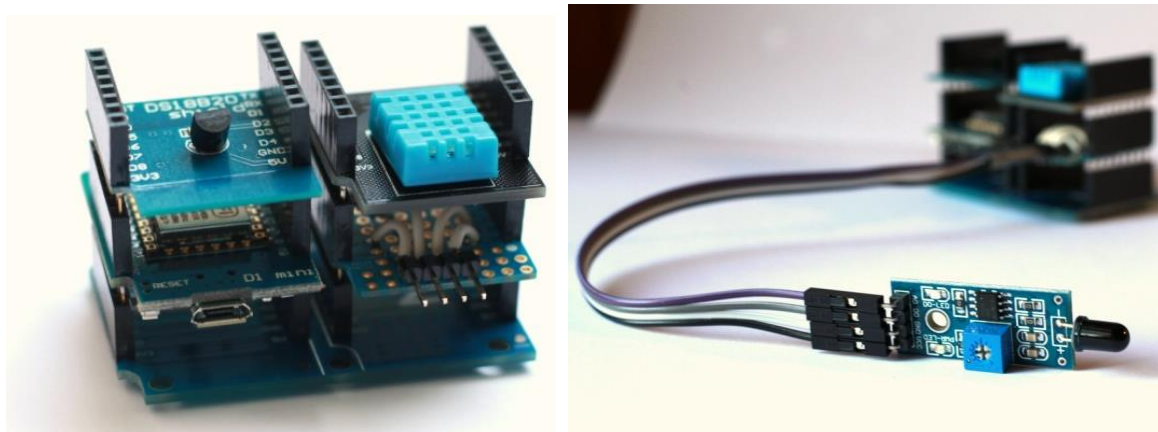


1.6.2. Disseny i muntatge del mòdul de sensors

Un cop havia escollit els components pels quals havien d'estar formats els meus mòduls principals, el dels sensors, vaig disposar-me a soldar la protoboard amb les connexions que pots veure al diagrama*.



Un cop tenia això enllestit, vaig soldar les respectives potes a cadascuna de les plaques, perquè poguessin apilar-se una sobre l'altre formant “una torre” amb l'objectiu d'ocupar el mínim espai, tal i com es mostra a la imatge* .



1.7. Estudi de les xarxes WiFi

Els dispositius que es connecten a la xarxa Wi-Fi s'anomenen estacions (STA). La connexió a Wi-Fi és proporcionada per un punt d'accés (AP), el qual actua com a centre per a una o més estacions (a l'ESP8266, el màxim d'estacions que poden connectar-se a un AP són 5).

Cada punt d'accés és reconegut per un SSID (Service Set Identifier), que essencialment serà el nom de la xarxa que seleccionem al connectar-nos amb un dispositiu (estació) al Wi-Fi.

El mòdul ESP8266 pot funcionar com una estació, per tant, podem connectar-lo a la xarxa Wi-Fi, però també pot funcionar com a soft-AP, per establir la seva pròpia xarxa Wi-Fi. Per tant, podem connectar altres estacions a aquest mòdul ESP. L'ESP8266 també pot operar en mode estació i en mode punt d'accés alhora. Això proporciona la possibilitat de construir xarxes.

Un cop creats els punts d'accés i les estacions, hem de saber que també existeix la funció de client, que permet connectar-nos al servidor per enviar i rebre informació i accedir a la funcionalitat que aquest proporciona.

1.8. La xarxa interna

1.8.1. Introducció a Arduino

Arduino IDE (Integrated Development Environment), és un programa informàtic compost per un conjunt d'eines de programació. Pot dedicar-se en exclusiva a un sol llenguatge de programació o bé pot utilitzar-se per diversos.

Un IDE és un entorn de programació que ha estat empaquetat com un programa d'aplicació; és a dir, que consisteix en un editor de codi, un compilador, un depurador i un constructor d'interfície gràfica (GUI). A més, en el cas d'Arduino incorpora les eines per carregar el programa ja compilat en la memòria flash del maquinari.

Els programes d'Arduino estan compostos per un sol fitxer amb extensió ".ino", encara que és possible organitzar-lo en diversos fitxers. El fitxer principal sempre ha d'estar en una carpeta amb el mateix nom que el fitxer.

1.8.2. Difusió: Mètode de la patata calenta

El mètode de difusió que he escollit podríem anomenar-lo “la patata calenta” com el seu nom indica, està basat en un joc de nens molt senzill el qual segurament tots hem jugat algun cop a la vida, però per qui no ho hagi fet, ho explicaré breument:

El joc consisteix en que un grup de persones col·locades de manera concreta (un cercle, per exemple) disposa d'una pilota. Una persona està fora del cercle i és qui comença el compte enrere. Mentrestant, la resta ha de passar-se la pilota, però no a qualsevol sinó als companys més propers (en el cas del cercle, les dues persones dels costats). Quan la persona que compta arriba al zero, qui tingui la pilota a les mans serà el perdedor.

Un cop hem entès el joc, és molt fàcil entendre el mètode de difusió: el moment en el qual la persona fora del cercle comença a comptar és el moment en el qual un dels mòduls amb sensors detecta un incendi, aquesta informació (la pilota) ha de començar a ser enviada ràpidament entre els mòduls de la xarxa (els jugadors) ja que, tot i haver dissenyat aquesta xarxa per detectar els incendis des del moment zero, hem de ser conscients de que es probable que les coses no siguin tan ràpides i precises com les havíem calculat i que per tant, en el moment en el qual un dels sensors detecta l'incendi, aquest ja estigui més avançat del que desitjaríem i aleshores cremi el mòdul abans de que aquest hagi pogut contactar amb l'exterior del bosc degut a que aquest és un procés més complicat que el de transmetre informació en una xarxa interna.

Per tant en el moment en el qual un dels sensors detecta un incendi, el primer que farà serà enviar aquesta informació als altres mòduls més propers i aquests a altres, això podem fer-ho de manera ràpida.

1.8.3. Disseny de la xarxa interna

La xarxa que jo he dissenyat consta de 3 mòduls, però el mateix disseny pot aplicar-se a possibles xarxes més grans. Com és una xarxa senzilla, consta només de dos d'aquests mòduls capaços de detectar incendis i per tant, només hi ha un receptor final, és a dir, que tota la informació ha d'acabar arribant al

mateix mòdul, el que, en el meu projecte, mostra en un display quin dels dos altres mòduls ha estat el detector de l'incendi, però com ja he dit, està pensat perquè sigui extrapolable a una xarxa més gran.

Per començar, tots tres mòduls creen una pròpia xarxa WiFi, per tant tots tres són punts d'accés (AP) en qualsevol moment, amb o sense incendi.

En el moment en el qual algun dels dos mòduls detecta un incendi, el que fa és convertir-se en un servidor amb la informació que enviarà, és a dir, l'existència de foc, el mòdul que l'ha detectat i com a conseqüència, la posició d'aquest. Un cop creada la funció de servidor, s'enviarà aquesta informació a tots els mòduls que trobi en l'anàlisi de xarxes (en el meu prototip, haurà de trobar dues xarxes) i quan la informació arribi al mòdul receptor, el qual és el final de la cadena, aquest la mostrarà en el display.

1.8.4. Muntatge de la xarxa

Per fer més senzill el treball de fer els codis, inicialment vaig utilitzar, en comptes de dos mòduls de sensors, dues plaques NodeMCU. Això em servia per poder crear la xarxa, sense haver de preocupar-me per la part dels sensors, que era en realitat, la més simple. Una altra raó per fer-ho d'aquesta manera és que una de les més importants són els sensors i per tant havia d'assegurar-me de que tot funcionava correctament, tal i com ho havia plantejat inicialment, perquè si muntava dos mòduls iguals i després no eren els definitius, havia de "llençar" dos mòduls, d'aquesta manera només en seria un.

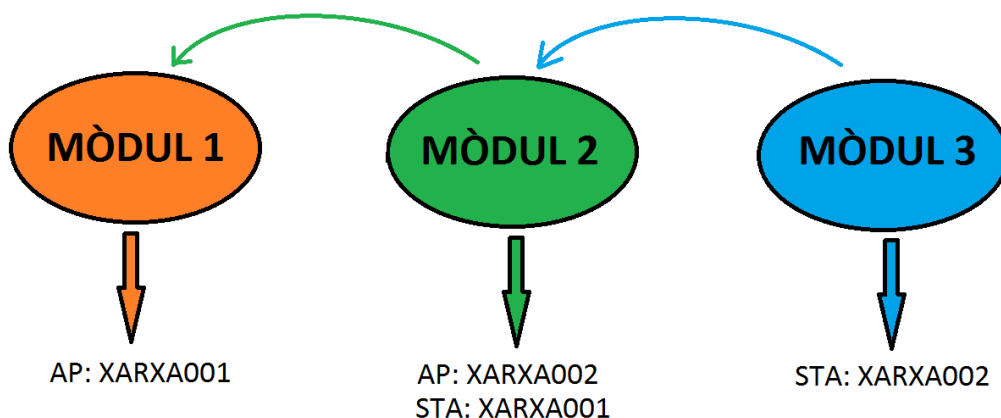
Per començar, vaig buscar exemples d'Arduino IDE poder entendre bé el funcionament dels Punts d'Accés i els clients. El primer exemple que vaig posar a prova va ser WiFiAccesPoint, aquest codi el que fa es crear una xarxa, la qual tu pots anomenar i posar contrasenya fàcilment i funciona com a servidor web, és a dir, en el moment en el qual des d'un altre dispositiu et connectes a aquesta xarxa creada, t'envia al servidor web, on pots posar la informació que desitgis. (versió 1, codi del MÒDUL1)

Un cop entès aquest codi i entesos els passos que duia a terme, vaig utilitzar un altre exemple d'Arduino IDE anomenat WifiClient, que simplement és un codi que

fa de Client, és a dir es connecta a un servidor i pot enviar-li informació. (versió 1, codi del MÒDUL3)

El següent objectiu era el de crear un programa que pogués ésser Client i Servidor alhora i crees una xarxa pròpia. Això vaig aconseguir-ho fent una barreja dels dos exemples anteriors, el problema va aparèixer amb les IP, ja que cada dispositiu ha de tenir una IP pròpia i aquesta venia determinada com 192.168.4.1 i jo no sabia com crear una nova IP per cada mòdul, això ho vaig solucionar incloent el SoftAPConf que et permetia canviar la configuració de la xarxa. (versió 1, codi del mòdul 2)

Del que en aquell inici disposava era d'una xarxa molt senzilla que tenia aquesta aparença:



Els tres programes inicials funcionaven correctament, connectant-se d'aquesta manera, però va aparèixer un problema que no esperava i és que les plaques tenen de fàbrica una protecció que anomenem gos guardià que bàsicament hi fa un RESET quan (Annex F) però això es pot solucionar fent que el codi estigui sempre en funcionament. Un cop solucionat, disposava d'una xarxa totalment estable.

El proper objectiu era fer enviar un missatge des del mòdul 3 al mòdul 1 dient quin és el seu nom, per això era necessari anomenar els mòduls d'una manera més formal, el nom que vaig escollir va ser: DETINC+ el número del mòdul. Un cop anomenats, apareix la part més complicada i a la vegada la més important de tot el codi.

En aquesta part el que havia de fer era que cadascun dels mòduls es reconegués a si mateix i un cop trobava una xarxa a la qual connectar-se s'hi connectés i envies al servidor un missatge en el qual aparegués el nom d'ell mateix. Aquest missatge seria enviat a través d'aquest mòdul al mòdul restant i per tant, es podria veure des de qualsevol monitor de qualsevol mòdul, qui ha estat el responsable de detectar l'incendi i enviar l'avís (versió 2, mòdul 1, 2, 3)

Seguidament vaig crear una nova versió del tercer mòdul que no enviés sempre el missatge sinó que mesurés la temperatura i si detectés, que aquesta pugés en algun moment, enviés el missatge. Per poder detectar la temperatura, vaig utilitzar un altre exemple d'Arduino anomenat Temperature Sensor DS18B20, un cop detectada hem d'incloure al codi la funció IF determinant primerament, quina és la temperatura a la qual comença la alerta, com el meu prototip no serà posat a prova amb un incendi real, he posat que la temperatura d'alerta és de 30°C). (versió 3, mòdul 3)

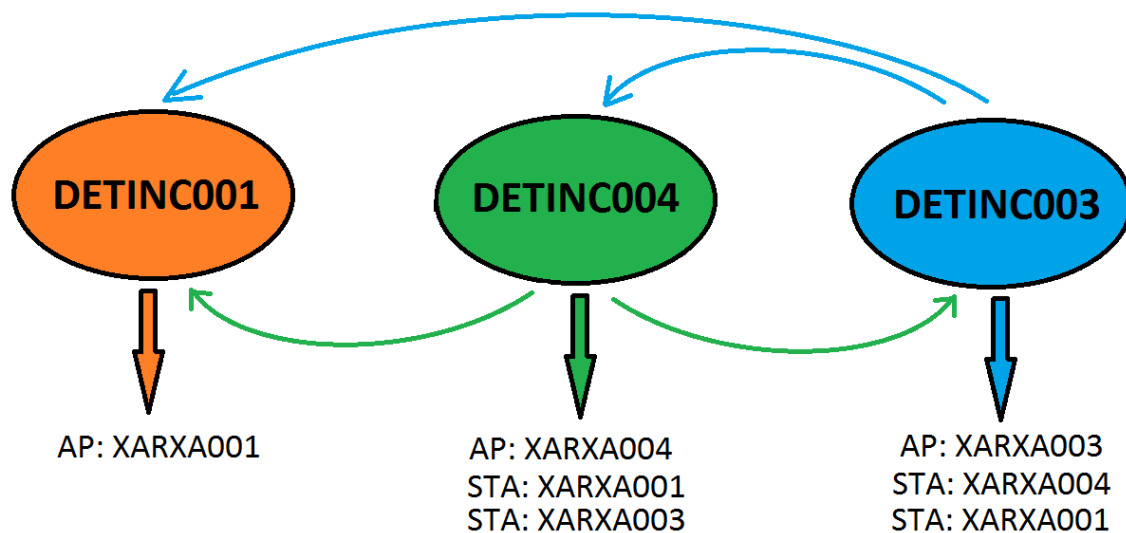
Un cop aconseguit això, havia de fer que el mòdul 3 fos client, servidor i mesures la temperatura alhora, per fer-ho vaig agafar part del codi del mòdul 2 (client i servidor alhora) i part del codi del mòdul 3 (mesura temperatura i envia missatge) i vaig crear un nou codi que realitza totes les tasques. (versió 4, mòdul 3)

Aquest últim codi ja era molt complex i feia les funcions bàsiques però havia de fer millores considerables. Una d'elles va ser que, en comptes de connectar-se a una xarxa anteriorment determinada al codi, realitzés un escaneig de les xarxes del voltant en descartés les que no formessin part de la xarxa interna, és a dir, que el seu SSID no comences per "XARXA". Per poder dur a terme aquesta actualització del codi, vaig fer-ho en dos parts: en la primera vaig utilitzar l'exemple d'Arduino WiFiScan per fer l'escaneig i vaig utilitzar un "IF" per descartar totes les xarxes excepte a la que havia de connectar-se, és a dir XARXA002 (versió 5, mòdul 3); en la segona part vaig canviar aquest IF perquè descartés totes les xarxes que no fossin de la xarxa interna utilitzant la funció startsWith. (versió 6, mòdul 3)

Ja tenia un codi que feia tot el que volia, però només amb un sensor de temperatura, per tant em calia incloure la resta de sensors importants (flama i

humitat). A la pàgina oficial d'Arduino vaig trobar un exemple per utilitzar el sensor de flama, però aquest notificava quan no hi havia flama, just el contrari del que jo volia, per tant vaig haver de modificar-lo perquè fes el que jo necessitava i vaig incloure-ho a l'última versió del mòdul 3. En quant al DHT, com ja tenia el DS18B20 (temperatura) que és molt més precís, vaig bloquejar la part del DHT que mesura la temperatura per només utilitzar-lo per mesurar humitat, a la pàgina d'Arduino també hi havia un codi d'exemple que t'ensenyava a utilitzar aquest sensor. (versió 7, mòdul 3)

Un cop tenia el codi finalitzat i funcional, vaig soldar un altre mòdul igual que el tercer al qual vaig compilar el mateix codi, canviant el nom per "DETINC004" i creant la xarxa "XARXA004". Aquest mòdul el vaig substituir pel mòdul 2, per tant, la xarxa definitiva té aquesta aparença:



Pots veure el funcionament del prototip de la xarxa a l'Annex K

1.9. Codi

MÒDUL 1

```
/* MODUL1
DISPLAY
Saida, treball de recerca, prova5 MODUL1
*/

#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3F,16,2);

const char *ssid = "XARXA001";           //Crea una xarxa anomenada XARXA001
const char *password = "treballderecerca"; //Crea contrasenya
const char *IP = "192.168.4.1";          //IP de la AP
const char *modul1orig = "DETINC001";    //Nom del mòdul DETINC001
IPAddress local_IP(192,168,4,1);
IPAddress gateway(192,168,4,1);
IPAddress subnet(255,255,255,0);
ESP8266WebServer server(80);

void handleRoot() {
  String origen = server.arg("T");
  server.send(200, "text/html", "<h1>Rebut missatge del client</h1>");
  Serial.print("missatge rebut de: ");      //Escriu rebut missatge de:
  Serial.println(origen);                   //Escriu el mòdul que ha detectat l'incendi
}

void setup() {
  delay(1000);
  Serial.begin(115200);
  Serial.println();
  delay(10);

  Serial.print("Configuring access point.."); //Es crea un AP
  WiFi.softAP(ssid, password);             //Contrasenya i ssid de la AP
```

```
WiFi.softAPConfig(local_IP, gateway, subnet);

IPAddress myIP = WiFi.softAPIP();
Serial.print("AP IP address: ");
Serial.println(myIP);
server.on("/", handleRoot);

server.begin();
Serial.println("HTTP server started");
delay(1);

lcd.init();           // initialize the lcd
lcd.init();
}

void loop() {
  delay(1);
  server.handleClient();
  delay(1);

  String origen = server.arg("T");
  server.send(200, "text/html", "<h1>Rebut missatge del client</h1>");
  lcd.backlight();
  lcd.setCursor(3,0);
  lcd.print("INCENDI");
  lcd.setCursor(2,1);
  lcd.print(origen);
}
```

MÒDUL 4

```

/* MÒDUL 4 */

#include <OneWire.h>
#include <DallasTemperature.h>
#include <ESP8266WiFi.h>
#define ONE_WIRE_BUS D2                //pota sensor temperatura
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
const int sensorPin = D8;              //pota sensor flama
#include "DHT.h"
#define DHTPIN D4                      // pota DHT
#define DHTTYPE DHT11                 // DHT 11

DHT dht (DHTPIN, DHTTYPE);
OneWire ourWire (ONE_WIRE_BUS);
DallasTemperature sensors (&ourWire);
const float Tincendi = 30;             //Temperatura d'incendi
const float Hincendi = 15;            //Humitat d'incendi
const char* modul4orig = "DETINCT004";

int value = 0;
const char* ssid = "XARXA002";        //Es conecta a XARXA002
const char* password = "treballderecerca"; //Crea la contrasenya
const char* host = "192.168.5.1";

const char *ssid4 = "XARXA004";       //Crea AP XARXA004
const char *password4= "treballderecerca"; //Crea la contrasenya
IPAddress local_IP (192,168,7,1);     //Canvi de IP, ara és 192.168.7.1
IPAddress gateway (192,168,7,1);
IPAddress subnet (255,255,255,0);
ESP8266WebServer server(80);

void handleRoot() {
    server.send (200, "text/html", "<h1>Rebut missatge del client</h1>");
    String origen = server.arg ("T");
    delay(5000);
    ++value;

```

```
Serial.println ("Connectant a");
Serial.print (ssid);

WiFiClient client;
const int httpPort = 80;
if (!client.connect(host, httpPort))
{
    Serial.println ("connexió fallida");
    return;
}

String url = String("/?T=") + String(origen);
Serial.println ("Requesting URL: ");
Serial.print (url);

// Envia la petició al servidor
client.print(String("GET ") + url + " HTTP/1.1\r\n" + "Host: " + host + "\r\n" +
    "Connection: close\r\n\r\n");
unsigned long timeout = millis();
while (client.available() == 0)
{
    if (millis() - timeout > 5000)
    {
        Serial.println(">>> Client Timeout!");
        client.stop();
        return;
    }
}
while (client.available())
{
    String line = client.readStringUntil('\r');
    Serial.print(line);
}
Serial.println();
Serial.println("finalitzant connexió");
}

void setup() {
    delay(1000);
    Serial.begin(115200);
```

```
Serial.println();
Serial.print("Configuring access point...");
WiFi.softAP(ssid4, password4);

IPAddress myIP = WiFi.softAPIP();
Serial.println(" IP del AP ");
Serial.print(myIP);
server.on("/", handleRoot);
server.begin();
Serial.println("servidor HTTP iniciat");
}

void loop() {
  server.handleClient();
  pinMode(sensorPin, INPUT);
  delay(1000);

  Serial.begin(115200);
  delay(900);
  Serial.println();
  Serial.println("INFORMACIÓ modul 4");

  sensors.begin(); //Enciende la libreria DallasTemperature

  dht.begin(); //Enciende libreria DHT
  float Hactual = dht.readHumidity();
  delay(30);
  Serial.print("-Humitat: ");
  Serial.print(Hactual);
  Serial.println();

  sensors.requestTemperatures(); // Send the command to get temperatures
  sensors.getTempCByIndex(0);
  delay(30);
  Serial.print("-Temperatura = ");
  Serial.print(sensors.getTempCByIndex(0));
  Serial.print(" Graus C");

  Serial.println();
  Delay (30);
}
```

```
Serial.print("-Flama? ");
Serial.print( digitalRead(sensorPin));
int flama = digitalRead(sensorPin);

const float Tactual = (sensors.getTempCByIndex(0));
if (Tactual >= Tincendi || flama == LOW || Hactual <= Hincendi) {
    Serial.begin(115200);
    WiFi.mode(WIFI_STA);
    WiFi.disconnect();
    delay(100);

    Serial.println("scan iniciat");
    int n = WiFi.scanNetworks();
    Serial.println("scan fet");
    if (n == 0)
        Serial.println(" cap xarxa trobada");
    else {
        Serial.print(n);
        Serial.println(" xarxes trobades:");
        for (int i = 0; i < n; ++i) {
            Serial.print(i + 1);
            Serial.print(": ");
            Serial.print(WiFi.SSID(i));
            Serial.println((WiFi.encryptionType(i) == ENC_TYPE_NONE)? " ":"*");
            delay(10);
        }
    }
    Serial.println("");
    for (int i = 0; i < n; ++i) {
        char* AP = "XARXA999";
        WiFi.SSID(i).toCharArray(AP,9);
        Serial.print(WiFi.SSID(i));

        if (WiFi.SSID(i).startsWith("XARXA")) {
            Serial.println();
            Serial.println(" xarxa interna");
            Serial.println();
            Serial.print("Connectant a ");
            Serial.println(AP);
            WiFi.begin(AP, password);
        }
    }
}
```



```

while (WiFi.status() != WL_CONNECTED)
{
    delay(500);
    Serial.print(".");
}

Serial.println("");
Serial.println("WiFi conectat");
Serial.println("Adreça IP ");
Serial.println(WiFi.localIP());
int value = 0 {
    delay(5000);
    ++value;
    WiFiClient client;
    const int httpPort = 80;
    if (!client.connect(host, httpPort)) {
        Serial.println("connexió fallada");
        Serial.println();
        return; }
    String url = String("/?T=") + String(modul4orig);
    Serial.print("Requesting URL: ");
    Serial.println(url);
    client.print(String("GET ") + url + " HTTP/1.1\r\n" +
        "Host: " + host + "\r\n" + "Connection: close\r\n\r\n");
    unsigned long timeout = millis();
    while (client.available() == 0) {
        if (millis() - timeout > 5000) {
            Serial.println(" Fi de la connexió ");
            client.stop();
            return;
        }
    }
    while(client.available()) {
        String line = client.readStringUntil('\r');
        Serial.print(line);
    }
    Serial.println();
    Serial.println("Connexió finalitzada");
}
}
}

```

MÒDUL 3

```

/* MÒDUL 3 */

#include <OneWire.h>
#include <DallasTemperature.h>
#include <ESP8266WiFi.h>
#define ONE_WIRE_BUS D2                //pota sensor temperatura
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
const int sensorPin = D8;              //pota sensor flama
#include "DHT.h"
#define DHTPIN D4                      // pota DHT
#define DHTTYPE DHT11                 // DHT 11

DHT dht (DHTPIN, DHTTYPE);
OneWire ourWire (ONE_WIRE_BUS);
DallasTemperature sensors (&ourWire);
const float Tincendi = 30;             //Temperatura d'incendi
const float Hincendi = 15;            //Humitat d'incendi
const char* modul4orig = "DETINCT003";

int value = 0;
const char* ssid = "XARXA002";         //Es conecta a XARXA002
const char* password = "treballderecerca"; //Crea la contrasenya
const char* host = "192.168.5.1";

const char *ssid2 = "XARXA003";        //Crea AP XARXA003
const char *password2 = "treballderecerca"; //Crea la contrasenya
IPAddress local_IP (192,168,6,1);      //Canvi de IP, ara és
192.168.6.1
IPAddress gateway (192,168,6,1);
IPAddress subnet (255,255,255,0);
ESP8266WebServer server(80);

```

(constants codi mòdul 3)

En el codi del mòdul 3, únicament canvien les constants perquè el cos del codi és exactament el mateix, omitint la inicialització del display.

1.10. Disseny i creació de les peces 3D

Per a cadascun dels mòduls necessitava una carcassa que em permetis protegir els components i alhora facilités la situació d'aquests.

La meua idea era que els mòduls puguin lligar-se a una branca de l'arbre de manera discreta, per alterar el menys possible l'ecosistema ja existent.

Vaig fer diverses proves de carcasses, pots trobar-les a l'Annex K

1.10.1. Introducció a OpenSCAD

Per poder dur a terme les peces 3D necessàries, he utilitzat el programa OpenSCAD ja que és el que havíem utilitzat anteriorment a l'aula i per tant, disposava d'una experiència prèvia. OpenSCAD és un software gratuït disponible per Linux/UNIX, MS Windows i MAC OS X.

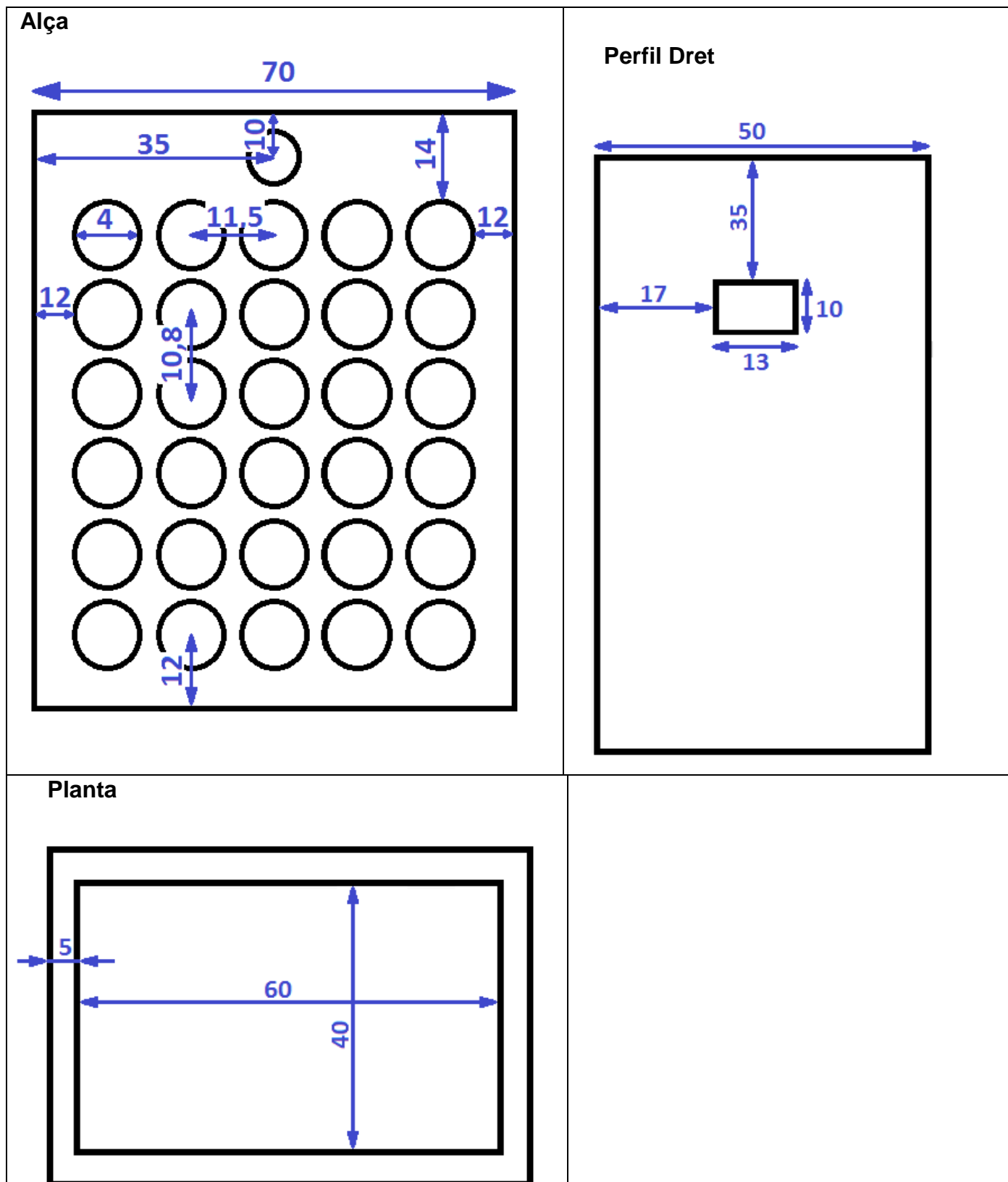
A diferència de la majoria de programaris gratuïts per crear models 3D, aquest no s'enfoca en els aspectes artístics del modelatge 3D sinó en els aspectes de CAD.

OpenSCAD no és un modelador interactiu sinó una mena de compilador 3D que llegeix en un arxiu *script* que descriu un objecte i el representa en un model 3D. Això ens ofereix un control total sobre el procés de modelatge i ens permet canviar fàcilment qualsevol pas en el procés de modelatge o fer dissenys que estan definits per paràmetres configurables.

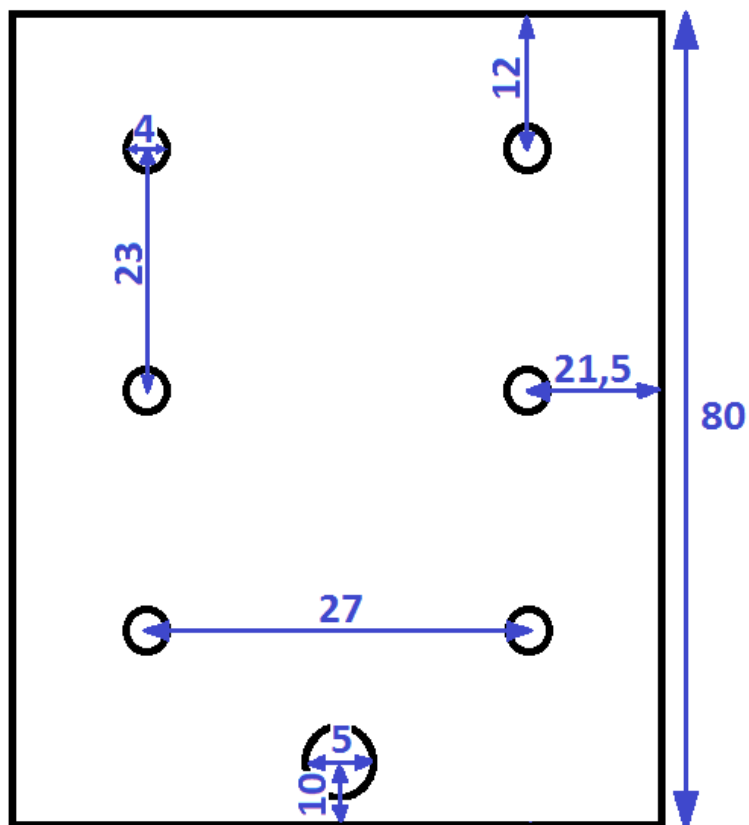
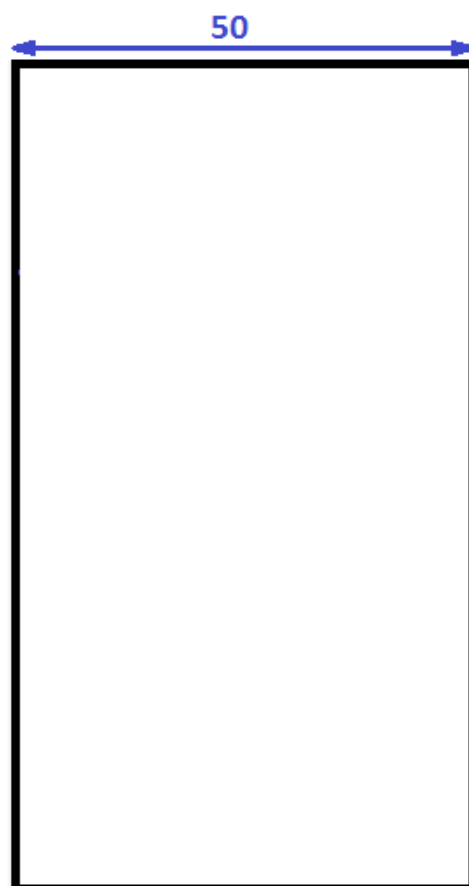
OpenSCAD ofereix dues tècniques principals de modelatge: primer, la Constructive Solid Geometry (també coneguda com CSG) que per met utilitzar figures geomètriques bàsiques i, en segon lloc, l'extrusió de contorns 2D. Com a format de format d'intercanvi de dades per a aquest 2D, s'utilitzen els arxius Autocad DXF a més, també és possible llegir paràmetres de disseny a partir d'arxius DXF i crear models 3D en els formats d'arxiu STL i OFF.

1.10.2. Disseny tècnic

COS

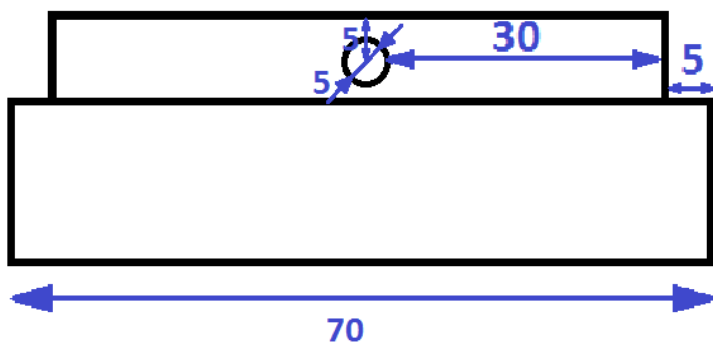


Degut a que aquesta peça no té cap cara igual, la he dividit en dos per poder mostrar les 6 cares.

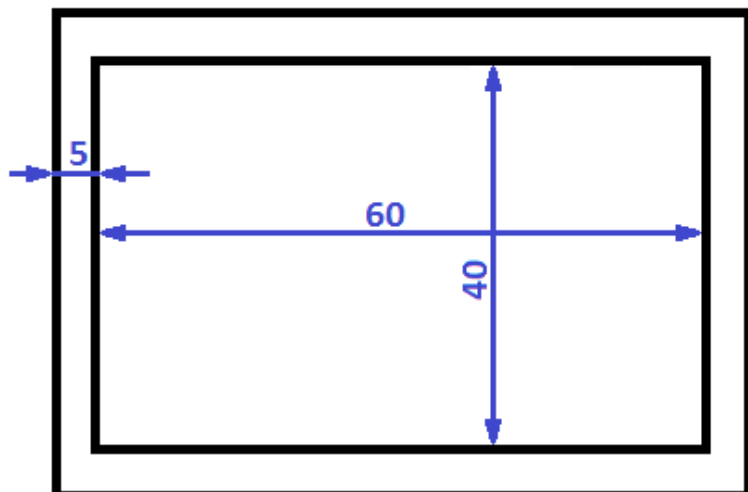
Planta**Alçat****Perfil esquerre**

TAPA

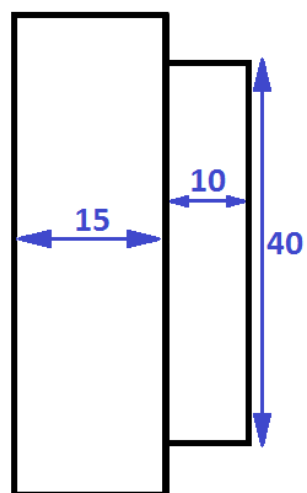
Alçat



Alçat



Perfil



1.10.3. Codi**COS**

```

difference(){
  //cos
  difference() {
    cube ([70,50,80]);
    translate ([5,5,5]) cube([60,40,90]);
  }

  translate ([21.5,8,12]) rotate ([90,0,0]) cylinder (h=10, r=2, $fn=48); //forat abaix esquerra
  translate ([21.5,8,35]) rotate ([90,0,0]) cylinder (h=10, r=2, $fn=48); //forat centre esquerra
  translate ([21.5,8,58]) rotate ([90,0,0]) cylinder (h=10, r=2, $fn=48); //forat amunt esquerra
  translate ([48.5,8,12]) rotate ([90,0,0]) cylinder (h=10, r=2, $fn=48); //forat abaix dreta
  translate ([48.5,8,35]) rotate ([90,0,0]) cylinder (h=10, r=2, $fn=48); //forat centre dreta
  translate ([48.5,8,58]) rotate ([90,0,0]) cylinder (h=10, r=2, $fn=48); //forat amunt dreta

  translate ([35,60,75]) rotate ([90,0,0]) cylinder (h=70, r=2.5, $fn=48); //forat ganxo

  //reixeta
  translate ([12,54,0])
  for (i = [12,22.8,33.6,44.4,55.2,66]) {
    translate([0, 0, i])
    rotate ([90,0,0]) cylinder (h=12, r=4, $fn=48);
  }

  translate ([23.5,54,0])
  for (i = [12,22.8,33.6,44.4,55.2,66]) {
    translate([0, 0, i])
    rotate ([90,0,0]) cylinder (h=12, r=4, $fn=48);
  }
}

```

```

translate ([35,54,0])
  for (i = [12,22.8,33.6,44.4,55.2,66]) {
    translate([0, 0, i])
    rotate ([90,0,0]) cylinder (h=12, r=4, $fn=48);
  }

translate ([46.5,54,0])
  for (i = [12,22.8,33.6,44.4,55.2,66]) {
    translate([0, 0, i])
    rotate ([90,0,0]) cylinder (h=12, r=4, $fn=48);
  }

translate ([58,54,0])
  for (i = [12,22.8,33.6,44.4,55.2,66]) {
    translate([0, 0, i])
    rotate ([90,0,0]) cylinder (h=12, r=4, $fn=48);
  }

translate ([65,17,45]) cube([10,13,10]); //forat carregador
}

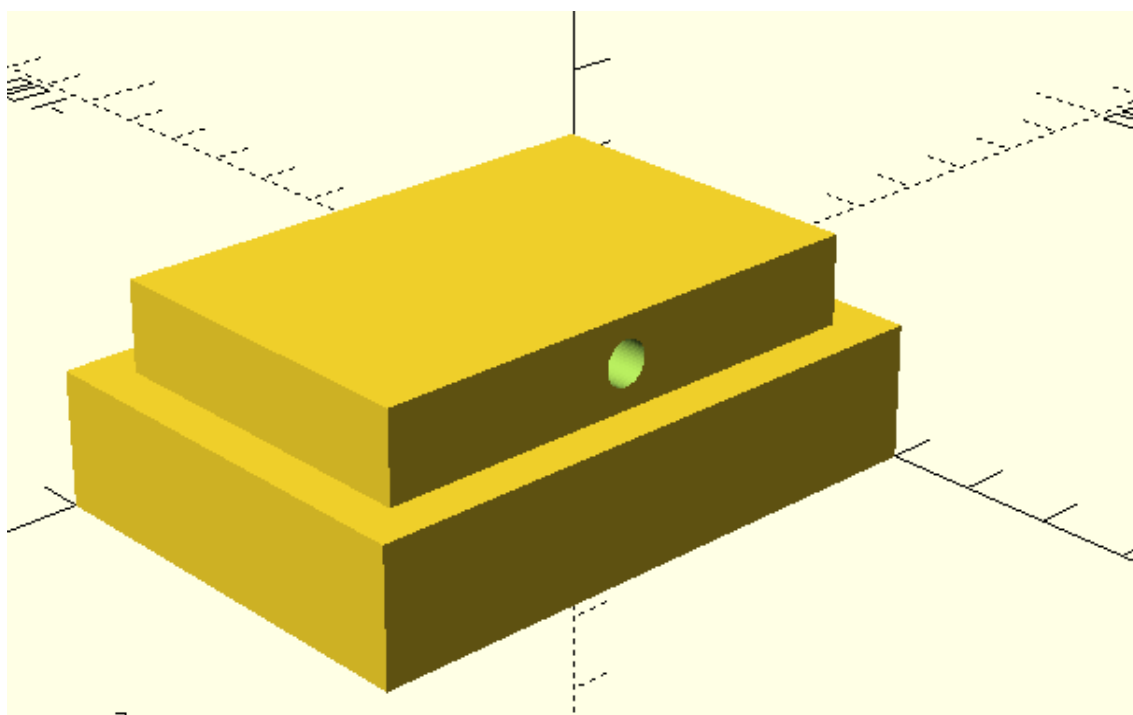
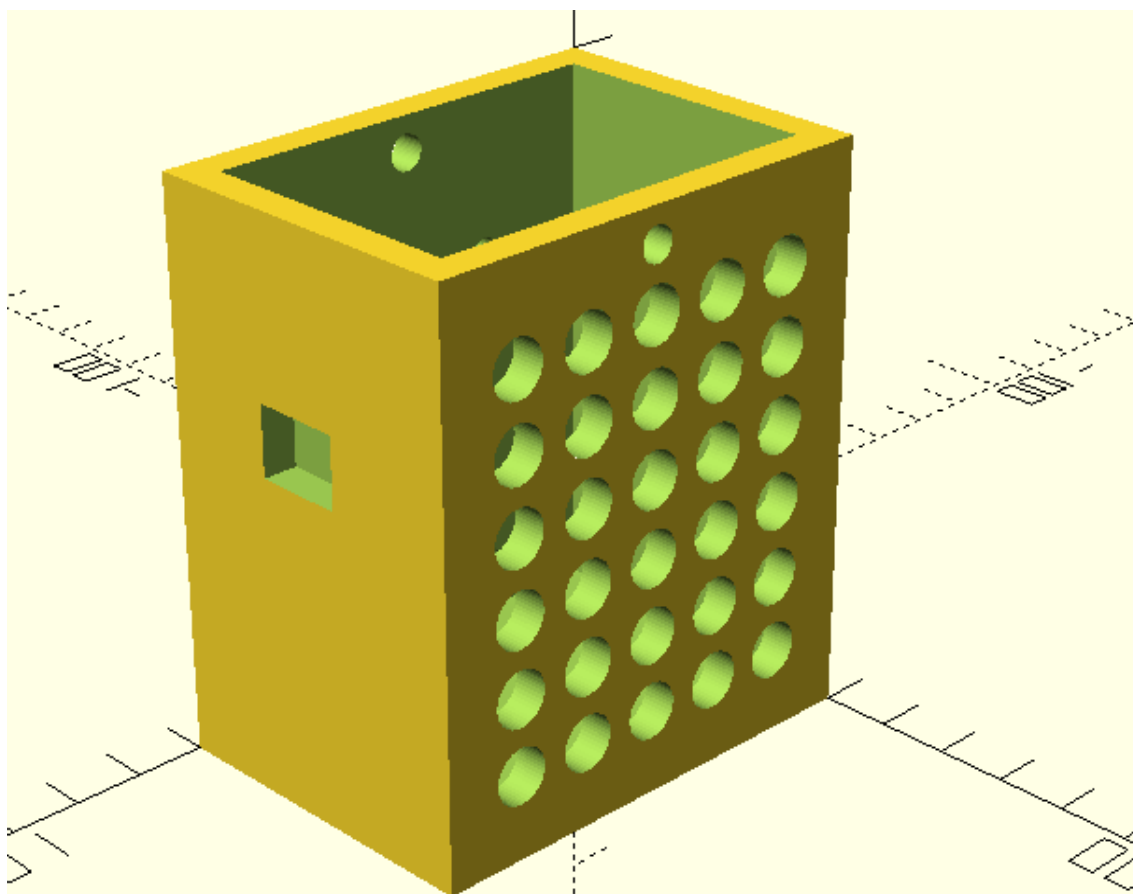
```

TAPA

```

union() {
  cube ([70,50,15]);
  translate ([5,5,15])
    difference () {
      cube ([60,40,10]);
      translate ([30,50,5]) rotate ([90,0,0]) cylinder (h=50, r=2.5, $fn=48);
    }
}

```

1.11. Planificació

	Operacions	Materials	Eines	Temps
Estudi	Recerca de mètodes de transmissió de dades	-	Ordinador	1:00h
	Estudi dels components	-	Ordinador	1:30h
	Estudi de la disposició dels mòduls	-	Ordinador	1:30h
Muntatge	Soldatge dels mòduls de sensors (1 i 4)	Estany Fusta per protegir la superfície de treball Wemos D1 mini DS18B20 shield DHT shield KY-026 Cables Dupond F-F 24 pins mascle femella	Tisores Soldador d'estany Pinces Alicates de punta plana	2:00h
Codis	Comprovació del funcionament dels sensors	DS18B20 shield DHT shield KY-026 Cable MicroUSB-USB	Ordinador Arduino IDE	1:00h
	Codi del mòdul 1	Mòdul NodeMCU Cable MicroUSB-USB Pantall LCD Bateria portàtil	Ordinador Arduino IDE	2:00h
	Codi del mòdul 2	Mòdul Node MCU Cable MicroUSB-USB Bateria portàtil	Ordinador Arduino IDE	3:00h
	Codi del mòdul 3	Wemos D1 mini DS18B20 shield	Ordinador Arduino IDE	6:00h

		DHT shield KY-026 Cables Dupond F-F Bateria portàtil		
	Codi del mòdul 4	Wemos D1 mini DS18B20 shield DHT shield KY-026 Cables Dupond F-F Bateria portàtil	Ordinador Arduino IDE	00:20h
	Modificacions per la millora	Wemos D1 mini DS18B20 shield DHT shield KY-026 Cables Dupond F-F Mòdul NodeMCU Pantall LCD Cable MicroUSB-USB	Ordinador Arduino IDE	1:30h
Disseny 3D	Mesura de les mides dels mòduls	Mòdul 1 totalment muntat Mòdul 3 totalment muntat	Peu de rei Paper i bolígraf	00:30h
	Esborrany del disseny	-	Paper Escaire Regla Llapis Goma	2:00h
	Disseny carcassa sensors	-	Ordinador Programari OpenSCAD	1:40h
	Disseny carcassa receptor	-	Ordinador OpenSCAD	1:00h

	Modificacions per la millora	-	Ordinador OpenSCAD	1:00h
Total de temps en hores				25:50h

1.12. Pressupost

Component/Material	Proveïdor	Quantitat	Preu de compra	Preu total
WeMos D1 mini	Aliexpress	2	3€	6€
DHT shield	Aliexpress	2	1€	2€
KY-026	Aliexpress	2	1€	2€
DS18B20	Aliexpress	2	2€	4€
NodeMCU	Aliexpress	1	5€	5€
Pantalla LCD	Aliexpress	1	2€	2€
Font d'alimentació 6V 2A DC	Basar	2	10€	20€
Bobina d'estany de soldadura	Onda Radio	1	9,35€	9,35€
Connector de 8 pins mascle-femella	Aliexpress	24	-	-
Dual shield base WeMos D1 mini	Aliexpress	2	0,79€	1,58€

Protoboard shield	Aliexpress	2	0,40€	0,80€
Cables Dupond F-F	DealExtreme	12	5,37€/50u	1,29€
Conector de 4 pins M-M 90º	Aliexpress	2	-	-
Cable USB- microUSB	Fnac	2	5€	10€
Plàstic PLA				
Cable de connexió	Taller	15cm	-	-
Preu total				64,02€

En aquest pressupost no ve inclòs el preu de la mà d'obra, després d'haver fet la planificació i haver calculat el temps aproximat que ha suposat dur a terme aquest prototip, podríem fer l'estimació del preu total si l'haguéssim encarregat a professionals:

Mà d'obra	Temps total	Preu de la mà d'obra	Preu dels materials
55€/h	26:00h	1430€	64,02€

El preu del prototip és de 1494,02€ però a això hauríem de sumar-li els impostos per fer-ho de manera totalment legal:

IVA (21%)	Preu total
313,74€	1807,76€

2. CONCLUSIONS

2.1. Comparativa amb els objectius inicials

He pogut assolir la gran majoria dels objectius que m'havia proposat inicialment:

He après moltíssim en quant a les mesures de prevenció i extinció dels incendis. La gran realitat és que existeixen moltíssims mètodes tant de detecció com prevenció però en quant a l'extinció seguim, essent per lo general, molt tradicionals. Tots els mètodes de detecció, que és en el que jo m'he centrat més profundament, són eficaços però els mètodes més habituals que són les torres de vigilància, tot i haver millorat al llarg d'aquests anys d'avenços tecnològics en la meva opinió s'estan quedant enrere respecte tots els altres mètodes, a més la construcció d'aquestes grandíssimes estructures alteren, conseqüentment, tot l'ecosistema en el qual s'instal·la. Per aquesta i moltíssimes més raons que podràs trobar a l'annex B, sincerament crec que en un futur el millor és que aquest tipus de vigilància acabi erradicant-se i substituint-se per les noves generacions de detectors com ho poden ser els drons o les xarxes de sensors. Així doncs per acabar amb aquesta massacre de la vegetació autòctona del nostre país necessitem no només seguir investigant i innovant, sinó finançar aquestes recerques.

Un altre objectiu de gran importància va ser el de millorar les meves habilitats amb la programació amb Arduino i el disseny 3D. Els dos programaris que he utilitzat al llarg d'aquest temps per dur a terme el meu treball ja els coneixia d'anys anteriors, però realment mai havia programat jo mateixa un codi ni havia pensat en com fer el disseny d'una peça per després imprimir-la, havia fet alguns dissenys però no pensats per mi. Degut a que la part de més rellevància en aquest projecte és la de programació, és aquesta en la que he dedicat més temps i per tant és el que he millorat notablement, abans no podia llegir un codi però ara mateix podria fer-ho sense grans dificultats. En quant a la part del disseny 3D, tot i haver-hi dedicat molt temps, vaig anar amb peus de plom per no espatllar tot el que havia fet durant el temps anterior, raó per la qual vaig utilitzar la meva experiència prèvia. Així que realment en aquesta part no considero haver fet una millora notable respecte el que ja sabia.

Si algú em preguntés quina és la part més complicada d'aquest treball diria sense dubtar-ho que és entendre com funcionen les connexions sense cables i particularment la connexió via WiFi, ja que tot i no ser difícil en quant al disseny ni al procés, la part més densa està aquí i per tant, sense tenir cap tipus de coneixement previ, entendre tot el que una xarxa implica és realment dur perquè és informació pura que ningú t'explica en una classe sinó que has de buscar i acabar entenent per tu mateixa. Realment d'aquest apartat em sento molt orgullosa de mi mateixa ja que he pogut dur-ho a terme i acabar entenent-ho, per mi això ja és un èxit.

En quant a la part més ecologista d'aquest projecte: l'objectiu de no alterar l'ecosistema de manera notablement perjudicial per els éssers vius, podríem dir que no la he pogut assolir completament. La part de la no alteració de l'ecosistema sí que he pogut dur-la a terme ja que els mòduls són realment petits i discrets, per tant no creen un gran impacte tot i que sí ho fa degut a que qualsevol canvi d'un ecosistema per petit que sigui, l'altera en major o menor mesura. D'altra banda, que la xarxa no sigui perjudicial pels éssers vius no és tan clar. Tot i que no està demostrat que la exposició als camps electromagnètics afecti a la salut, i hi ha molts debats sobre si el WiFi és o no perjudicial per la salut, en parlo més detalladament a l'Annex E.

L'únic punt que no he aconseguit ha estat el de treure el màxim rendiment a les ESP8266 ja que aquesta és una placa molt complexa amb moltíssimes funcions i aleshores aconseguir treure-li tot el seu rendiment possible és molt complicat tenint en compte que els meus coneixements sobre programació són molt bàsics i el temps que se'ns dona per realitzar aquest treball és limitat.

2.2. Possibles millores del projecte

La millora més important, segons el meu punt de vista, és aconseguir alimentar els mòduls amb energia verda ja que aquesta segueix la ètica del treball i degut a que aquests no consumeixen molta energia, podríem fer-ho amb una font d'alimentació reduïda, així com una petita placa fotovoltaica.

Aconseguir instal·lar sensors més potents i com a conseqüència, amb una fiabilitat més alta, seria una millora realment notòria, que faria del prototip, un dispositiu realment aplicable a un bosc ja que reduiria en gran mesura les falses alarmes. Pots veure més informació en quant a sensors de flama a l'Annex D.

Una altra gran millora seria la de canviar les carcasses dels mòduls per unes altres més robustes i com a proposta, podria tenir una doble funcionalitat. És a dir, no només la de protegir i penjar els mòduls sinó que també podria dur a terme una funció d'ajuda als animals, com per exemple les aus. Seria realment fàcil realitzar una caseta per a ocells on ells poguessin anar tranquil·lament a veure aigua o menjar, però que aquesta construcció tingués un doble departament destinat a protegir els mòduls.

2.3. Valoració personal

En quant a la vessant més ecologista d'aquest projecte i per tant, de la meua personalitat, ja que tot aquest treball és un reflex de mi i de les meves preocupacions mediambientals, ha estat un plaer fer una recerca sobre l'impacte dels incendis forestals, les causes d'aquests, la fiabilitat dels dispositius de detecció i el més important, el percentatge de responsabilitat que tenim els humans en aquestes catàstrofes. Tot i no ser aquesta la part fonamental del meu treball, personalment és la més important de totes, perquè al cap i a la fi, és la raó per la qual vaig voler dedicar tant temps a un detector d'incendis. Em semblava mediambientalment necessari intentar aportar el meu granet de sorra o si més no, estar informada de les causes de la situació que estem vivint no només a nivell estatal sinó mundial.

Passant ara a la part essencial del treball, la que ha posat nom al títol d'aquesta memòria: la xarxa detectora d'incendis forestals.

Estic realment orgullosa de tot el treball, des de les coses que no han acabat sent com imaginava o com havien de ser, fins a les que han funcionat, que són les que m'han anat donant grans alegries i una mica d'esperança en el futur de la xarxa, quan començava a perdre-la, durant tot aquest temps d'implicació.

Estic molt satisfeta de com a acabat sent el treball però sobretot de mi mateixa per haver estat capaç de fer tot això jo sola i haver anat, poc a poc, aconseguint els meus objectius i ampliant els meus coneixements.

3. BIBLIOGRAFIA

3.1. Fonts consultades

[COM01] ESP8266. <https://programarfacil.com/podcast/esp8266-wifi-coste-arduino/>
<https://aprendiendoarduino.wordpress.com/2017/09/12/que-es-esp8266/>
<https://www.prometec.net/wemos-d1-esp8266-wifi/>

[COM02] Wemos D1 mini. <http://hobbycomponents.com/shields/868-wemos-d1-mini-dht-temphum-shield>
<https://www.connectedcities.com.ph/products/ds18b20-shield-for-wemos-d1-mini>

[COM03] Sensor de flama <https://www.luisllamas.es/detector-llama-arduino/>

[COM03] NodeMcu <https://www.prometec.net/nodemcu-arduino-ide/>

[COM03] LCD <http://www.electrohobby.org/conversor-lcd-i2c-arduino/>

[PRG01] OpenSCAD. <http://www.openscad.org/>

[PRG02] Arduino IDE. <https://www.arduino.cc/en/main/software>

[EST01] Impacte mediambiental
<http://archivos.greenpeace.org/espana/Global/espana/report/other/incendiosforestales-que-per.pdf>
http://www.abc.es/sociedad/abci-80-por-ciento-incendios-forestales-estan-provocados-hombre-y-30-por-ciento-descuidos-201606162138_noticia.html
<https://civio.es/espana-en-llamas/2016/11/24/quien-detecta-los-incendios/>

[EST02] Majors incendis forestals.

http://cadenaser.com/programa/2017/06/24/hora_14_fin_de_semana/1498332703_202531.html

<http://www.who.int/mediacentre/factsheets/fs193/es/>

[EST03] Xarxes WiFi. <http://www.arduino-esp8266.readthedocs.io/>

[REL01] WiFi. <http://www.who.int/mediacentre/factsheets/fs193/es/>

[REL02] Gos Guardià.

[https://es.wikipedia.org/wiki/Perro_guardi%C3%A1n_\(electr%C3%B3nica\)](https://es.wikipedia.org/wiki/Perro_guardi%C3%A1n_(electr%C3%B3nica))

[MIL01] Sensors de Flama.

<https://www.notifier.es/documentacion/notifier/manualesobs/MNDT690.pdf>

4. ANNEXOS

A. Impacte mediambiental

Els boscs gestionats de manera sostenible ens ofereixen a tots els éssers vius una gran i renovable font de recursos i serveis ambientals, així doncs, ens aporten aliments, combustible, medicaments, materials de construcció i fins i tot valors culturals i estètics.

Una petita part dels incendis forestals es deuen a causes naturals, però el 80% dels incendis són causats pels homes. D'aquest percentatge, el 70% són provocats de manera conscient i l'altre 30%, són descuits. A més de la pèrdua de vides humanes i dels enormes danys materials que provoquen els focs, els danys ecològics presenten una llarga llista de conseqüències, molts cops irreparables, de les quals no en som plenament conscients.

IMPACTE SOBRE LA VEGETACIÓ

En els moments inicials després de l'incendi comencen a recuperar-se les espècies dotades de mecanismes de resistència al foc, així com les plantes herbàcies que es veuen afavorides pel momentani augment de la fertilitat i per l'absència de competidors. Posteriorment s'inicia la regeneració de les espècies abans existents que van desplaçant a les espècies oportunistes, dominants en el moment inicial. No obstant això, la recurrència dels incendis en un mateix lloc afavoreix que les espècies millor adaptades al foc esdevinguin dominants i que la vegetació instal·lada sigui cada vegada més piròfita.

IMPACTE SOBRE LA FAUNA

La fauna es qui pateix més directament els efectes negatius dels incendis des de els seus inicis, de manera que es veu afectada pel foc, per l'ona de calor i pels processos d'asfíxia. La destrucció dels ecosistemes i de les cadenes tròfiques dificulta la regeneració de la fauna preexistent abans de l'incendi, fundamentalment per l'absència d'estrat vegetal que porti aliment i per les condicions extremes que presenta el sòl. Les poblacions faunístiques es veuen afectades per la pèrdua de l'hàbitat, de recursos tròfics, de llocs de nidificació o refugi i en general per la dràstica pèrdua de la qualitat dels ecosistemes. Per una altra banda, la repetició d'incendis en zones determinades, provoca l'abandonament definitiu de les espècies autòctones

EFFECTES SOBRE ELS BALANÇOS HIDROLÒGICS

Es redueix la coberta vegetal que evita el vessament d'aigua superficial, i que permetia el retorn d'aquesta a l'atmosfera. D'altra banda, l'incendi incrementa la impermeabilitat del sòl, reduint així la humidificació d'aquest. La infiltració de l'aigua de pluja es veu dificultada per la destrucció de l'estructura superficial del sòl. Com a conseqüència d'aquests processos, augmenta el vessament superficial, duplicant els valors habituals. Amb les primeres pluges després de l'incendi s'originen les majors esorrenties d'aigua superficial, fins a un 20% de la precipitació, quan el normal és que no superi el 5% de la pluja caiguda.

IMPACTE SOBRE LA QUALITAT DE L'AIGUA

Les aigües que drenen ràpidament el sòl cremat, presenten un elevat potencial erosiu. El material en suspensió que porten, així com els nutrients en dissolució, afecten a les aigües contaminant-les. A més, la posterior sedimentació d'aquests materials va acumulant-se al fons dels rius, embassaments i ries

IMPACTE SOBRE LA QUALITAT DE L'ATMÒSFERA

Els incendis alliberen quantitats de CO₂ molt inferiors a les quantitats emeses per la combustió de combustibles fòssils, també són alliberats compostos nitrogenats i diversos hidrocarburs. Els compostos més contaminants s'alliberen quan la combustió és incompleta, quan el front de foc és molt ràpid o en les combustions fumejants posteriors a l'incendi. Les partícules de carbó i les cendres són un altre dels problemes atmosfèrics que causen els incendis.

EROSIÓ DEL SÒL

Després de l'incendi, la cobertura normal del sòl queda dràsticament reduïda, apareixent un sòl cobert per cendres i restes calcinades que desapareixen ràpidament amb el vent i les primeres pluges. L'erosió en els terrenys incendiats apareix fonamentalment amb les primeres pluges. Després de la combustió, la matèria orgànica vegetal es pot mineralitzar o passar a enriquir momentàniament el sòl en forma de nutrients. No obstant això, una bona part dels elements nutritius es perden en l'atmosfera, contaminant-la a causa de la volatilització. Altres nutrients es perden dissolts en les aigües d'escorrentia.

IMPACTE SOBRE EL PAISATGE

Amb els incendis forestals es destrueixen els ecosistemes i es malmeten tots els múltiples usos de les muntanyes i el paisatge: naturals, ecològics i productius, així com els referents a l'oci. Tants són els impactes ecològics dels incendis que, a més de ser impossible quantificar-se econòmicament, és difícil analitzar fins a les últimes conseqüències.

B. Estat actual a Espanya

VIGILÀNCIA TERRESTRE FIXA

Els llocs de vigilància terrestre fixa poden ser torres, casetes o refugis, que apareixen distribuïts per tot el territori, especialment en àrees on els recursos forestals posseeixen un alt valor o en zones d'alt risc d'incendis. Per a una ubicació ràpida i fiable del focus detectat des d'un lloc de vigilància s'han desenvolupat sistemes informàtics basats en la utilització de fotografies panoràmiques, com ara VIGIS i Ubi-foc. En la detecció automàtica d'incendis s'utilitzen càmeres de vigilància d'infrarojos a Andalusia i a Castella i Lleó.



VIGILÀNCIA TERRESTRE MÒBIL

Les patrulles de vigilància mòbil en vehicle tot terreny faciliten la detecció de focus registrats a les zones no cobertes per la xarxa de vigilància fixa. A més, els vehicles de vigilància, dotats d'un petit dipòsit d'aigua, permeten un primer atac al foc.



Les patrulles terrestres s'empren també en la Comunitat Valenciana, Aragó i Castella i Lleó per a localitzar possibles

incendis per llamps, a partir de les coordenades d'impactes facilitades per l'Agència Estatal de Meteorologia.

VIGILÀNCIA AÈRIA

La vigilància aèria es sol desenvolupar mitjançant helicòpters, avions de coordinació i observació i avions de petita càrrega, que permeten un primer atac després de la detecció. A més, aquestes aeronaus poden comptar amb avenços tecnològics com GPS, càmeres de vídeo i infrarojos.

Actualment els drons també s'han implementat en la detecció d'incendis. Són útils a l'hora de detectar incendis ja que treballen en hores nocturnes i que les seves càmeres de visió nocturna i d'infrarojos permeten detectar qualsevol presència humana o animal amb definicions espectaculars.



SATÈL·LITS ARTIFICIALS

Encara que aquests tenen un elevat cost permeten abastar extenses àrees de terreny, i a més poden realitzar altres funcions que permeten amortitzar el seu cost. Normalment segueixen òrbites geostacionàries, però també poden seguir altres òrbites que permetin realitzar un rastreig global. Aconsegueixen detectar la majoria de focs, exceptuant aquells que queden ocults sota els núvols.

XARXES DE SENSORS

Aquest sistema de detecció d'incendis es basa en la combinació de diferents tipus de sensors, connectats per mitjà d'una xarxa sense fils que envia les dades obtinguts a una consola central de monitoratge.

De moment no hi ha cap xarxa de sensors desplegada en funcionament, tot i que s'està treballant en aquest sistema de detecció.

C. Grans incendis forestals a Espanya

Gairebé 50.000 incendis forestals en l'última dècada a Espanya han convertit el foc en un dels problemes ambientals més importants.

Al 1994, va començar l'onada d'incendis devastadors a Espanya amb un total de 437.635 hectàrees cremades i 19.263 sinistres que van causar la mort de 33 persones.

La comunitat autònoma d'Extremadura va ser una de les més afectades durant el 2003, en total es van cremar 44.102 hectàrees aquell estiu.

Un foc provocat en 2005 per uns excursionistes mentre encenien una barbacoa va cremar més de 13.000 hectàrees a Guadalajara. En aquest incendi van morir 11 treballadors d'una brigada en ser atrapats pel foc.

Al 2006, Galícia va ser la comunitat més afectada pels incendis. Es van cremar un total de 95.947 hectàrees. Només el 10 d'agost es van arribar a comptabilitzar 281 incendis.

Al 2007, les Illes Canàries van patir un dels seus pitjors anys amb un total de 35.758 hectàrees forestals cremades.

El 2012, el foc va arrasar 13.000 hectàrees a la comarca de l'Empordà (Girona) que van provocar la mort de 4 persones. Els incendis van ser especialment violents també a València, amb més de 45.000 hectàrees cremades. Aquest mateix any, un incendi va acabar amb part d'un bosc de Laurisilva al Parc Nacional de la Gomera que costarà dècades a regenerar causa de l'antiguitat de l'espècie a la zona.

No tots els grans incendis es produeixen a l'estiu. Al 2014, els primers grans incendis forestals, amb més de 500 hectàrees calcinades, es van produir al

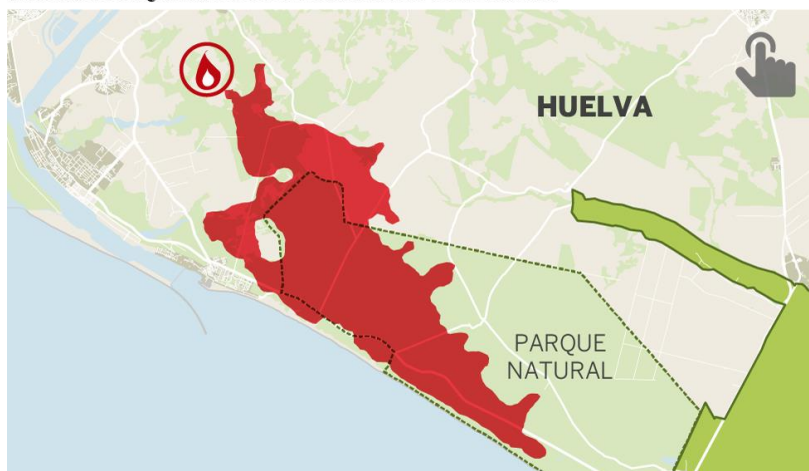
començament de la primavera. La Serra de Gador a la província d'Almeria va ser l'incendi més devastador de l'any amb 3.400 hectàrees calcinades.

Al 2015, coincidint amb l'entrada en vigor de la Llei 21/2015 que enduria les penes en la relació als incendis forestals, es va produir un repunt dels incendis. Un any especialment dur per a Andalusia amb incendis que van acabar amb gairebé 12.000 hectàrees, la majoria calcinades en l'incendi de Quesada, a la província de Jaén, va arrasar 9.806 hectàrees.

Al 2016 el foc va deixar una ferida profunda en moltes regions. A l'agost el foc va devorar el 7% de la Palma, emportant-se la vida d'un agent forestal. I es va acarnissar especialment amb la Comunitat Valenciana, amb 4 grans incendis, i amb Galícia, amb 9 grans incendis i més de 20.000 hectàrees afectades.

Sense anar molt més lluny, aquest mateix estiu tots nosaltres hem vist milers de notícies sobre de dos dels incendis més importants que ha viscut Espanya des que jo mateixa puc recordar: l'incendi al parc natural de Doñana que va acabar amb 8486 hectàrees i el gran incendi a Galícia i Astúries que va acabar amb més d'11500 hectàrees

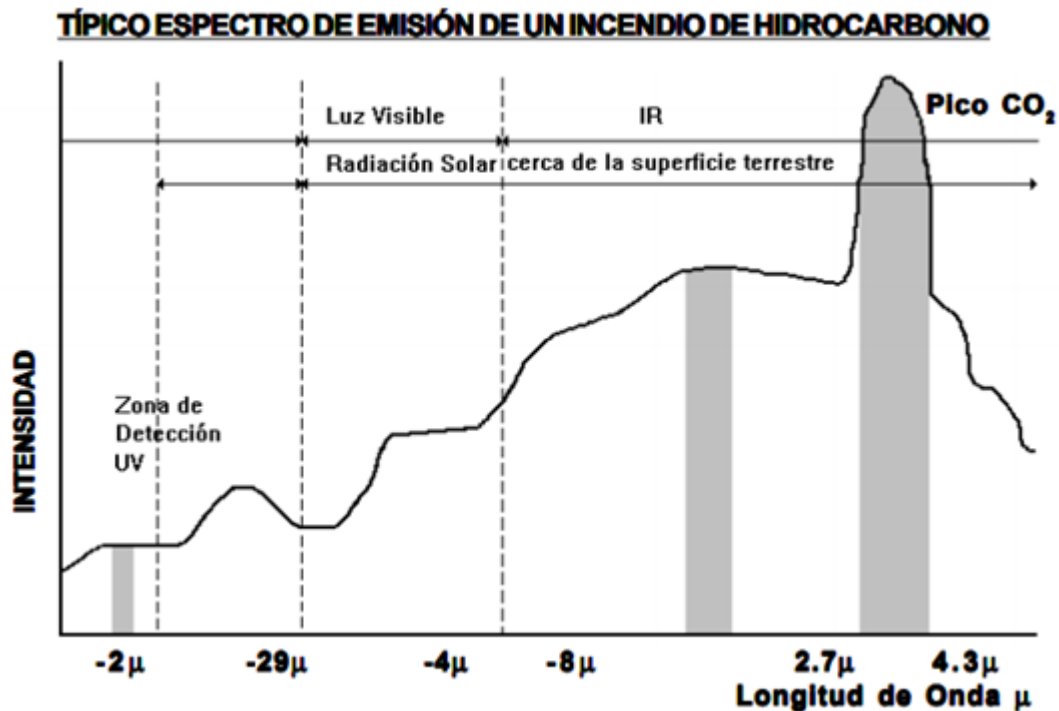
SUPERFICIE QUEMADA EN EL INCENDIO DE DOÑANA



Fuente: Comisión Europea, Junta de Andalucía. EL PAÍS

D. Sensors de flama

L'energia que irradia la flama d'un foc és primordial per la detecció d'aquest. El 30%-40% d'aquesta energia es dissipa en forma de radiació electromagnètica en diverses línies espectrals com l'ultra violeta (UV) i les infraroges (IR).



Com el model de radiació espectral del foc és únic, permet línies espectrals diferents, les quals s'utilitzen per equips de detecció. Normalment els detectors de foc utilitzen sensors òptics que registren les radiacions de les longituds d'ona seleccionades. Les senyals captades pel sensor passen per un procés predeterminat que inclou d'anàlisi de la freqüència de la flama, la comprovació entre les senyals d'energia, la correlació matemàtica entre les senyals, tècniques comparatives (AND o OR) o la correlació per memoritzar els anàlisis espectrals tot i que de vegades que els equips de detecció n'inclouen més d'una i per tant seran més fiables en quant a la sensibilitat. Els principals detectors òptics són els següents:

DETECTORS DE FLAMA UV

La banda espectral UV, es absorbida pel medi que l'envolta com l'aire, el fum, la pols, gasos i diverses matèries orgàniques.

Els detectors basats en aquesta tecnologia detecten el foc amb una extrema rapidesa (3-4 milisegons) degut a la alta radiació d'energia UV emesa per focs i explosions en els seus inicis.

Malauradament, si es produeixen radiacions UV que no s'absorbeixen pel medi, generen falses alarmes.

DETECTORS DE FLAMA IR

En molts incendis es produeixen radiacions infraroges. La temperatura de la flama emet un patró d'espectre específic fàcil de reconèixer però aquests detectors IR són víctimes de falses alarmes causades per la radiació de cossos calents com escalfadors, làmpades, etc.

DETECTORS DE FLAMA UV/IR

La tecnologia de doble espectre UV/IR utilitza un sensor UV i un IR. El propi sensor UV ja és un bon detector de les flames, però tal i com he mencionat anteriorment, generen falses alarmes, per prevenir-les s'incorpora el canal IR. Junts constitueixen un detector fiable per la majoria d'aplicacions a mitja distància. Però com tot, té els seus inconvenients: en ambients industrials ambdós sensors creen falses alarmes degudes a situacions en les que hi ha presents falsos estímuls de foc, com per exemple la llum solar.

DETECTORS DE FLAMA IR/IR

Una altra tecnologia de doble espectre és la que combina dues bandes espectrals en la banda IR. La base d'aquest anàlisi és "L'espectre diferencial" en el que s'analitzen dues línies espectrals: una és emesa pel foc mentre l'altre es emesa de forma dèbil pels voltants. Aquest tipus de detectors IR detecten la radiació en els dos canals i processa les senyals. Tot i la seva gran eficàcia,

aquests detectors pateixen atenuació atmosfèrica, especialment en aplicacions de llarga distància

DETECTORS DE TRIPLE IR

Aquest detector utilitza una combinació de tres sensor IR de banda extremadament estreta. Un avarca la banda espectral d'emissió de Co₂ i els altres dos cobreixen diferents bandes espectrals especialment seleccionades. El triple IR no causa falses alarmes a ninguna altra font de radiació continua que no sigui foc incloent la il·luminació i altres fonts de radiació de cossos calents.

E. És perjudicial el WiFi?

En els últims vint anys s'han realitzat un gran nombre d'estudis per determinar si l'exposició a les ones electromagnètiques dels dispositius sense fils presenten riscos per a la salut.

Efectes a curt termini

La principal conseqüència de la interacció amb l'energia radioelèctrica és l'escalfament dels teixits. En el cas de les freqüències utilitzades pels telèfons mòbils, la major part de l'energia és absorbida per la pell i altres teixits superficials, de manera que l'augment de temperatura en el cervell o en altres òrgans del cos és insignificant.

En diversos estudis s'han investigat els efectes dels camps de radiofreqüència en l'activitat elèctrica cerebral, la funció cognitiva, la son, el ritme cardíac i la pressió arterial. Fins a la data, aquests estudis semblen indicar que no hi ha proves de que l'exposició a camps de radiofreqüència (de nivell inferior als que provoquen l'escalfament dels teixits) tinguin efectes perjudicials per a la salut.

Efectes a llarg termini

Les investigacions epidemiològiques per analitzar els possibles riscos a llarg termini derivats de l'exposició a les radiofreqüències s'han centrat sobretot en trobar una relació entre els tumors cerebrals i l'ús de telèfons mòbils. No obstant això, atès que nombrosos tipus de càncer no són detectables fins molts anys després del contacte amb el causant del tumor i l'ús dels telèfons mòbils no es

va generalitzar fins a principis del decenni de 1990, a dia d'avui en els estudis epidemiològics només poden analitzar els tipus de càncer que es manifesten en un termini més breu. Tanmateix, els resultats d'estudis realitzats amb animals coincideixen que l'exposició a llarg termini a camps de radiofreqüències no augmenta el risc de contraure càncer.

Tot i que fins a la data no s'ha confirmat que l'ús de les xarxes sense fils tinguin efectes perjudicials per a la salut, La OMS determina i promou periòdicament les prioritats d'investigació relatives als camps de radiofreqüència i la salut.

F. Gos guardià

El Watchdog o gos guardià, en català és un mecanisme de seguretat que provoca un reset del sistema en cas que aquest s'hagi bloquejat. Va començar a utilitzar-se àmpliament a principi dels anys 80.

Consisteix en un temporitzador que anirà contínuament fent un compte enrere, inicialment amb un valor relativament alt. Quan aquest comptador arribi a zero, es reiniciarà el sistema, així que s'ha de dissenyar una subrutina en el programa de manera que refresqui o reiniciar el gos guardià abans que provoqui el reset. Si el programa falla o es bloqueja, en no actualitzar el comptador del gos guardià al seu valor d'inici, aquest arribarà fins a zero i es reiniciarà el sistema. Això fa que controlem l'execució del programa. Si la CPU, processador o microcontrolador entra en un bucle d'instruccions d'una manera infinita el temporitzador genera una interrupció generalment pel terminal de Reset fent que el sistema torni a reiniciar des del principi.

G. Versions del codi del mòdul 1

```
/* VERSIO 1 MODUL 1 */
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>

/* Set these to your desired credentials. */
const char *ssid = "WifiSaida";
const char *password = "treballderecerca";

ESP8266WebServer server(80);
void handleRoot() {
  server.send(200, "text/html", "<h1>hola Saida</h1>");
}
void setup() {
  delay(1000);
  Serial.begin(115200);
  Serial.println();
  Serial.print("Configuring access point...");
  /* You can remove the password parameter if you want the AP to be open. */
  WiFi.softAP(ssid, password);
  IPAddress myIP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
  Serial.println(myIP);
  server.on("/", handleRoot);
  server.begin();
  Serial.println("HTTP server started");
}

void loop() {
  server.handleClient();
}
```

H. Versions del codi del mòdul 2

```
/* VERSIÓ 1 MODUL 2 */

#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>

const char *ssid2 = "WifiSaida2";          //Crea un punt d'accés Wifi anomenat
WifiSaida2
const char *password2 = "treballderecerca"; //Crea la contrasenya
IPAddress local_IP(192,168,5,1);           //Canvi de IP, ara és 192.168.5.1
IPAddress gateway(192,168,4,9);
IPAddress subnet(255,255,255,0);
ESP8266WebServer server(80);
void handleRoot() {
    server.send(200, "text/html", "<h1>WifiSaida2 funciona correctament</h1>");
    //Envia un missatge WifiSaida2 funciona correctament
}

const char* ssid    = "WifiSaida";          //Es connecta a un punt d'accés Wifi
anomenat WifiSaida
const char* password = "treballderecerca"; //Contrasenya del WifiSaida
const char* host = "192.168.4.1";

void setup() {
    delay(1000);
    Serial.begin(115200);
    delay(10);

    Serial.println();
    Serial.print("Configuring access point...");
    WiFi.softAP(ssid2, password2);
    WiFi.softAPConfig(local_IP, gateway, subnet);

    IPAddress myIP = WiFi.softAPIP();
    Serial.print("AP IP address: ");
```

```
const int httpPort = 80;
if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    return;
}

// We now create a URI for the request
String url = "/";
Serial.print("Requesting URL: ");
Serial.println(url);

// This will send the request to the server
client.print(String("GET ") + url + " HTTP/1.1\r\n" +
    "Host: " + host + "\r\n" +
    "Connection: close\r\n\r\n");
unsigned long timeout = millis();
while (client.available() == 0) {
    if (millis() - timeout > 5000) {
        Serial.println(">>> Client Timeout !");
        client.stop();
        return;
    }
}

// Read all the lines of the reply from server and print them to Serial
while(client.available()){
    String line = client.readStringUntil('\r');
    Serial.print(line);
}

Serial.println();

Serial.println("closing connection");
}
```

```
/* VERSIÓ 2 MODUL 2 */

#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>

const char *ssid2 = "XARXA002";           //Crea un punt d'accés Wifi anomenat
XARXA002
const char *password2 = "treballderecerca"; //Crea la contrasenya
const char *modul2orig = "DETINC002";      //Nom del mòdul DETINC002
IPAddress local_IP(192,168,5,1);           //Canvi de IP, ara és 192.168.5.1
IPAddress gateway(192,168,5,1);
IPAddress subnet(255,255,255,0);
ESP8266WebServer server(80);

int value = 0;
const char* ssid = "XARXA001";           //Es connecta a un punt d'accés Wifi
anomenat XARXA001
const char* password = "treballderecerca"; //Contrasenya de XARXA001
const char* host = "192.168.4.1";

void handleRoot() {
  server.send(200, "text/html", "<h1>Rebut missatge del client</h1>");
  String origen = server.arg("T");
  Serial.print("missatge rebut de:");
  Serial.println(origen);
  delay(5000);
  ++value;
  Serial.print("connecting to ");
  Serial.println(host);
  // Use WiFiClient class to create TCP connections
  WiFiClient client;
  const int httpPort = 80;
  if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    return;
  }
}
```



```
// We now create a URI for the request
String url = String("/") + String(origen);

Serial.print("Requesting URL: ");
Serial.println(url);

// This will send the request to the server
client.print(String("GET ") + url + " HTTP/1.1\r\n" +
    "Host: " + host + "\r\n" +
    "Connection: close\r\n\r\n");
unsigned long timeout = millis();
while (client.available() == 0) {
    if (millis() - timeout > 5000) {
        Serial.println(">>> Client Timeout !");
        client.stop();
        return;
    }
}

// Read all the lines of the reply from server and print them to Serial
while(client.available()){
    String line = client.readStringUntil('\r');
    Serial.print(line);
}

Serial.println();
Serial.println("closing connection");
}

void setup() {
    delay(1000);
    Serial.begin(115200);
    delay(10);

    Serial.println();
    Serial.print("Configuring access point...");
    WiFi.softAP(ssid2, password2);
    WiFi.softAPConfig(local_IP, gateway, subnet);

    IPAddress myIP = WiFi.softAPIP();
    Serial.print("AP IP address: ");
```

```
Serial.println(myIP);
server.on("/", handleRoot);
server.begin();
Serial.println("HTTP server started");
delay(1);

Serial.println();           //Comença per connectar-se a la xarxa WifiSaida
Serial.println();
Serial.print("Connecting to ");
Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}

Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}

void loop() {
  delay(1);
  server.handleClient();
  delay(1);
```

I. Versions del codi del mòdul 3

```
/*VERSIÓ 1 MÒDUL 3 */

#include <ESP8266WiFi.h>

const char* ssid    = "WifiSaida2";
const char* password = "treballderecerca";
const char* host = "192.168.5.1";

void setup() {
  Serial.begin(115200);
  delay(10);

  // We start by connecting to a WiFi network

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

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

int value = 0;
++value;

Serial.print("connecting to ");
Serial.println(host);

// Use WiFiClient class to create TCP connections
```

```
void loop() {
  delay(5000);
  ++value;
  Serial.print("connecting to ");
  Serial.println(host);

  // Use WiFiClient class to create TCP connections
  WiFiClient client;
  const int httpPort = 80;
  if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    return;
  }
  // We now create a URI for the request
  String url = "/";
  Serial.print("Requesting URL: ");
  Serial.println(url);

  // This will send the request to the server
  client.print(String("GET ") + url + " HTTP/1.1\r\n" +
    "Host: " + host + "\r\n" +
    "Connection: close\r\n\r\n");
  unsigned long timeout = millis();
  while (client.available() == 0) {
    if (millis() - timeout > 5000) {
      Serial.println(">>> Client Timeout !");
      client.stop();
      return;
    }
  }

  // Read all the lines of the reply from server and print them to Serial
  while(client.available()){
    String line = client.readStringUntil('\r');
    Serial.print(line);
  }
  Serial.println();
  Serial.println("closing connection");
}
```

```
/* VERSIÓ 2 MÒDUL3
*/
#include <OneWire.h>
#include <DallasTemperature.h>
#include <ESP8266WiFi.h>
#define ONE_WIRE_BUS D2

OneWire ourWire(ONE_WIRE_BUS);
DallasTemperature sensors(&ourWire);
const float Tincendi = 30;          //Temperatura a la qual es connectarà a la
XARXA002
const char* modul3orig = "DETINCT003";    //Nom del modul DETINC003
const char* ssid    = "XARXA002";        //Es connecta a XARXA002
const char* password = "treballderecerca"; //Contrasenya de XARXA002
const char* host = "192.168.5.1";        //IP de XARXA002

void setup() {
  delay(1000);
  Serial.begin(115200);
  Serial.println("YourDuino.com: Temperature Sensor Test Program");
  Serial.println("Temperatura modul 3");
  delay(1000);

  sensors.begin(); //Enciende la libreria DallasTemperature
  delay(10);
}

void loop() {
  Serial.println();
  Serial.print("Mesurant temperatura");
  sensors.requestTemperatures(); // Send the command to get temperatures
  Serial.println(" FET ");

  Serial.print("Temperatura = ");
  Serial.print(sensors.getTempCByIndex(0)); //Calcula la temperatura
  Serial.println(" Graus C");
  delay(5000);
}
```

```
const float Tactual = (sensors.getTempCByIndex(0)); //Tactual es la temperatura
anteriorment calculada

if (Tactual >= Tincendi) //Si la temperatura actual es major o igual a la
temperatura d'incendi, es conecta a la XARXA002
{
    Serial.begin(115200);
    delay(10);

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

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
    int value = 0;

    { delay(5000);
    ++value;

    Serial.print("connecting to ");
    Serial.println(host);

    // Use WiFiClient class to create TCP connections
    WiFiClient client;
    const int httpPort = 80;
    if (!client.connect(host, httpPort)) {
        Serial.println("connection failed");
        return;
    }
}
```

```
// We now create a URI for the request
String url = String("/?T=") + String(modul3orig);

Serial.print("Requesting URL: ");
Serial.println(url);

// This will send the request to the server
client.print(String("GET ") + url + " HTTP/1.1\r\n" +
    "Host: " + host + "\r\n" +
    "Connection: close\r\n\r\n");
unsigned long timeout = millis();
while (client.available() == 0) {
    if (millis() - timeout > 5000) {
        Serial.println(">>> Client Timeout !");
        client.stop();
        return;
    }
}

// Read all the lines of the reply from server and print them to Serial
while(client.available()){
    String line = client.readStringUntil('\r');
    Serial.print(line);
}

Serial.println();
Serial.println("closing connection");
}
}
} Serial.println("closing connection");
}
```

```
/*VERSIO 3 MÒDUL 3 */

#include <OneWire.h>
#include <DallasTemperature.h>
#include <ESP8266WiFi.h>
#define ONE_WIRE_BUS D2
#include <WiFiClient.h>
#include <ESP8266WebServer.h>

OneWire ourWire(ONE_WIRE_BUS);
DallasTemperature sensors(&ourWire);
const float Tincendi = 30;
const char* modul3orig = "DETINCT003";

int value = 0;
const char* ssid = "XARXA002"; //Es conecta a un punt d'accés Wifi
anomenat XARXA002
const char* password = "treballderecerca"; //Crea la contrasenya
const char* host = "192.168.5.1";

const char *ssid2 = "XARXA003"; //Crea un punt d'accés Wifi anomenat
XARXA003
const char *password2 = "treballderecerca"; //Crea la contrasenya
IPAddress local_IP(192,168,6,1); //Canvi de IP, ara és 192.168.6.1
IPAddress gateway(192,168,6,1);
IPAddress subnet(255,255,255,0);
ESP8266WebServer server(80);

void handleRoot() {
  server.send(200, "text/html", "<h1>Rebut missatge del client</h1>");
  String origen = server.arg("T");

  delay(5000);
  ++value;

  Serial.println("Connectant a");
  Serial.print(ssid);

  // Use WiFiClient class to create TCP connections
```



```
Serial.print(ssid);

// Use WiFiClient class to create TCP connections
WiFiClient client;
const int httpPort = 80;
if (!client.connect(host, httpPort)) {
  Serial.println("connexió fallida");
  return;
}

// We now create a URI for the request
String url = String("/?T=") + String(origen);

Serial.println("Requesting URL: ");
Serial.print(url);

// This will send the request to the server
client.print(String("GET ") + url + " HTTP/1.1\r\n" +
  "Host: " + host + "\r\n" +
  "Connection: close\r\n\r\n");
unsigned long timeout = millis();
while (client.available() == 0) {
  if (millis() - timeout > 5000) {
    Serial.println(">>> Client Timeout !");
    client.stop();
    return;
  }
}

// Read all the lines of the reply from server and print them to Serial
while(client.available()){
  String line = client.readStringUntil('\r');
  Serial.print(line);
}

Serial.println();
Serial.println("finalitzant connexió");
}
```

```
void setup() {
  Serial.begin(115200);
  delay(10);

  Serial.println("Configurant el AP ");
  WiFi.softAP(ssid2, password2);
  WiFi.softAPConfig(local_IP, gateway, subnet);

  IPAddress myIP = WiFi.softAPIP();
  Serial.println("  IP del AP ");
  Serial.print(myIP);
  server.on("/", handleRoot);
  server.begin();
  Serial.println("servidor HTTP iniciat");
  delay(1000);
}

void loop() {
  delay(1);
  server.handleClient();
  delay(1);

  Serial.begin(115200);
  Serial.println();
  Serial.println("Temperatura del modul 3");
  delay(1000);

  sensors.begin(); //Enciende la libreria DallasTemperature
  delay(10);

  Serial.println();
  Serial.print("Mesurant temperatura");
  sensors.requestTemperatures(); // Send the command to get temperatures
  Serial.println(" FET ");

  Serial.print("Temperatura = ");
  Serial.print(sensors.getTempCByIndex(0));
  Serial.println(" Graus C");
```

```
delay(5000);

const float Tactual = (sensors.getTempCByIndex(0));

if (Tactual >= Tincendi)
{
    Serial.begin(115200);

    // Set WiFi to station mode and disconnect from an AP if it was previously connected
    WiFi.mode(WIFI_STA);
    WiFi.disconnect();
    delay(100);

    Serial.println("scan iniciat");

    // WiFi.scanNetworks will return the number of networks found
    int n = WiFi.scanNetworks();
    Serial.println("scan fet");
    if (n == 0)
        Serial.println(" cap xarxa trobada");
    else
    {
        Serial.print(n);
        Serial.println(" xarxes trobades:");
        for (int i = 0; i < n; ++i)
        {
            // Print SSID and RSSI for each network found
            Serial.print(i + 1);
            Serial.print(": ");
            Serial.print(WiFi.SSID(i));
            Serial.println((WiFi.encryptionType(i) == ENC_TYPE_NONE)? " ":"*");
            delay(10);
        }
    }
    Serial.println("");
    for (int i = 0; i < n; ++i) {
        String AP = WiFi.SSID(i);
        Serial.print(WiFi.SSID(i));
        if (WiFi.SSID(i).startsWith("XARXA")) {
```

```
Serial.println(" xarxa interna");
Serial.println();
    Serial.println();
Serial.print("Connectant a ");
Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}

Serial.println("");
Serial.println("WiFi conectat");
Serial.println("Adreça IP ");
Serial.println(WiFi.localIP());
    int value = 0;

    { delay(5000);
    ++value;

// Use WiFiClient class to create TCP connections
WiFiClient client;
const int httpPort = 80;
if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    return;
}

// We now create a URI for the request
String url = String("/?T=") + String(modul3orig);

Serial.print("Requesting URL: ");
Serial.println(url);

// This will send the request to the server
client.print(String("GET ") + url + " HTTP/1.1\r\n" +
    "Host: " + host + "\r\n" +
```

```
        "Connection: close\r\n\r\n");
    unsigned long timeout = millis();
    while (client.available() == 0) {
        if (millis() - timeout > 5000) {
            Serial.println(">>> Client Timeout !");
            client.stop();
            return;
        }
    }

    // Read all the lines of the reply from server and print them to Serial
    while(client.available()){
        String line = client.readStringUntil('\r');
        Serial.print(line);
    }

    Serial.println();
    Serial.println("Connexió finalitzada");
    }
}}}
    else {
        Serial.println (" xarxa externa");
        Serial.begin(115200);
        delay(10);

    }
}}
```

J. Versions de la carcassa del mòdul de sensors

```
//cos
difference(){
  difference() {
    cube ([70,50,80]);
    translate ([5,5,5]) cube([60,40,90]);
  }
  translate ([20.5,8,12]) rotate ([90,0,0]) cylinder (h=10, r=2, $fn=48); //forat abaix esquerre

  translate ([20.5,8,35]) rotate ([90,0,0]) cylinder (h=10, r=2, $fn=48); //forat centre esquerre

  translate ([20.5,8,58]) rotate ([90,0,0]) cylinder (h=10, r=2, $fn=48); //forat amunt esquerre

  translate ([47.5,8,12]) rotate ([90,0,0]) cylinder (h=10, r=2, $fn=48); //forat abaix dreta

  translate ([47.5,8,35]) rotate ([90,0,0]) cylinder (h=10, r=2, $fn=48); //forat centre dreta

  translate ([47.5,8,58]) rotate ([90,0,0]) cylinder (h=10, r=2, $fn=48); //forat amunt dreta

  translate ([35,60,75]) rotate ([90,0,0]) cylinder (h=70, r=2.5, $fn=48); //forat ganxo

//reixeta
translate ([8,43,0])
for (i = [8,18.8,29.6,40.4,51.2,62]){
  translate([0, 0, i])
  cube([8,8,8]);
}

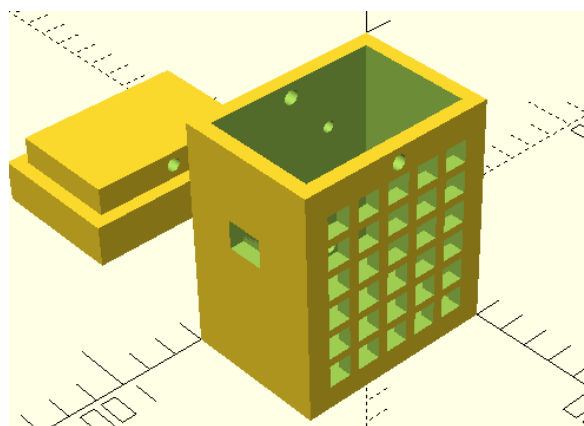
translate ([19.5,43,0])
for (i = [8,18.8,29.6,40.4,51.2,62]){
  translate([0, 0, i])
  cube(size = 8, center = false);
}

translate ([31,43,0])
for (i = [8,18.8,29.6,40.4,51.2,62]){
  translate([0, 0, i])
```

```

    cube(size = 8, center = false);
}
translate ([42.5,43,0])
for (i = [8,18.8,29.6,40.4,51.2,62]){
    translate([0, 0, i])
    cube(size = 8, center = false);
}
translate ([54,43,0])
for (i = [8,18.8,29.6,40.4,51.2,62]){
    translate([0, 0, i])
    cube(size = 8, center = false);
}
translate ([65,17,45]) cube([10,13,10]); //forat carregador
}
//tapa1
translate ([0,-100,0])
union() {
    cube ([70,50,15]);
    translate ([5,5,15])
    difference (){
        cube ([60,40,10]);
        translate ([30,50,5]) rotate ([90,0,0]) cylinder (h=50, r=2.5, $fn=48);
    }
}
}

```



K. Video del funcionament del prototip

Vull finalitzar aquesta memòria adjuntant una prova en vídeo del funcionament del prototip.

Podeu trobar-la a YouTube cercant: *Xarxa detectora d'incendis forestals*.

Copiant aquest enllaç al navegador:

https://www.youtube.com/watch?v=w_xozqMINRI

Escanejant el codi QR:

