

Treball de Fi de Grau

Grau en Enginyeria en Tecnologies Industrials

Prototip d'una paperera intel·ligent

MEMÒRIA

Autor: Xavier Estadella Valls

Director/s: Lluís Solano

Convocatòria: Juny 2016



**Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona**



Resum

Actualment el terme *Big Data*, concepte que fa referència a l'emmagatzematge de grans quantitats de dades i als procediments per analitzar-los, està present entre nosaltres. Tant per temes empresarials com les xarxes socials o gran consum, com per temes d'esport tant en àmbit professional com aficionat i sobretot molt important per temes d'investigació. La creació d'aparells amb sensors o sistemes de mesura esta en augment i permeten la recopilació dades que després s'analitzen per a arribar a conclusions o trobar certs patrons de comportament en les coses.

En aquest treball es vol crear un prototipus d'una paperera que permeti la recollida d'una sèrie de dades sobre aquesta, per així poder veure l'estat en el que es troba en cada moment i per a poder analitzar les dades, extreure conclusions i/o trobar certs patrons. Bàsicament en aquest prototipus es controlarà tant la temperatura i humitat del interior com el tant per cent del volum que es troba ocupat. Aquest control es realitza mitjançant l'ús de sensors i una placa electrònica.

A més es requereix d'un sistema al darrere de la paperera que permeti l'emmagatzematge d'aquestes dades per a una posterior visualització i anàlisis. Aquest aspecte també es dissenya durant el treball, concretament es crea un servidor amb una Base de Dades per a poder disposar totes les mesures. Aquest servidor també funciona com a web per a visualitzar l'estat actual de la paperera i les dades dipositades.

Un cop realitzat el disseny del prototip i comprovar la seva funcionalitat, s'ha fet un estudi econòmic sobre el cost del treball i de l'impacte ambiental que aquest ha suposat. Destacar que aquest compleix amb la seva finalitat i que l'estructura global (paperera més servidor) s'ha pensat per a poder contenir més papereres i, per tant, poder escalar aquest treball al control de tot un conjunt de papereres de ,per exemple, un aeroport, centre comercial, ciutat ...

L'escriptura d'aquest treball s'ha pensat per a que qualsevol persona pugui reproduir la totalitat del prototipus, tant la part de la paperera com la del servidor. I ,per tant, pugui aplicar canvis o millores adaptant-lo al seu gust.

Sumari

RESUM	1
SUMARI	3
1. INTRODUCCIÓ	5
1.1. Objectius del treball	5
1.2. Abast del treball	5
2. ARQUITECTURA DEL PROTOTIPUS	6
2.1. Estructura global	6
2.2. Paperera	7
2.2.1. Arduino	7
2.2.1.1. Que és Arduino?	7
2.2.1.2. Arduino UNO	8
2.2.1.3. Arduino Yún	9
2.2.2. Raspberry	12
2.2.2.1. Que és Raspberry Pi?	12
2.2.2.2. Raspberry Pi 3	13
2.2.3. Elecció de la placa	14
2.2.4. Usuaris i connexions	17
2.3. Servidor	18
2.3.1. Servidor Apache	18
2.3.2. LAMP	19
2.3.3. Usuaris i connexions	19
3. ARQUITECTURA DE LA PAPERERA	21
3.1. Elements	21
3.1.1. Sensor de distancia d'ultrasons HC-SR04	21
3.1.2. Sensor d'humitat i temperatura DHT22	22
3.2. Posada a punt	23
3.2.1. Configuració Arduino Yún	23
3.2.1.1. Instal·lació Software Arduino IDE	24
3.2.1.2. Afegir llibreria del sensor de temperatura DHT22	27
3.2.1.3. Connectar la placa Arduino Yún a la xarxa	28
3.2.1.4. Actualització del OpenWrt-Yun Image	30

3.2.1.5. Expandir el espai en disc de la placa	32
3.2.2. Connexions.....	35
3.3. Localització dels sensors	37
3.4. Codi.....	38
3.4.1. Estructura del subcodi.....	38
3.4.2. Estructura del codi principal	43
3.5. Flux d'informació.....	47
4. ARQUITECTURA DEL SERVIDOR	48
4.1. Posada a punt.....	48
4.2. Estructura BBDD.....	51
4.2.1. Taula de la BBDD	51
4.2.2. Creació de la BBDD i taula	52
4.3. Fluxos d'informació	56
4.4. Mòduls	56
4.4.1. Add.php	56
4.4.2. Connexio.php.....	57
4.4.3. Index.php	57
4.4.4. Goto.php	58
4.4.5. Temperatura.php	58
4.4.6. Humitat.php	59
4.4.7. Volum.php.....	59
5. PLANIFICACIÓ I COSTOS DEL TREBALL	61
5.1. Planificació.....	61
5.1.1. Fase prèvia	61
5.1.2. Fase de disseny.....	61
5.1.3. Fase final	61
5.2. Costos associats al treball	64
6. IMPACTE AMBIENTAL	65
CONCLUSIONS	67
AGRAÏMENTS	69
BIBLIOGRAFIA	70
Referències bibliogràfiques	70
Bibliografia complementària	72

1. Introducció

1.1. Objectius del treball

L'objectiu principal d'aquest treball de fi de grau és el disseny d'un prototipus d'una paperera amb tota l'arquitectura que això comporta que sigui capaç de recollir una sèrie de dades sobre el seu estat i les enviï a un servidor per a poder emmagatzemar-les, visualitzar-les i analitzar-les. Això vol dir:

- Disseny del sistema de recollida de dades de la paperera amb una placa electrònica.
- Disseny del sistema d'emmagatzematge i visualització de les dades (servidor).
- Disseny de l'arquitectura que englobi ambdós sistemes.
- Document que serveixi per a reproduir el prototipus dissenyat.

1.2. Abast del treball

L'abast d'aquest treball és la creació d'un prototipus funcional de paperera que permeti la recollida de la temperatura i humitat interior i el volum ocupat. Això comporta el disseny de la recollida de dades, és a dir, elecció de la placa electrònica, cerca dels sensors, localització d'aquests en la paperera, programació de la placa...

Per altre banda també cal pensar en el disseny del sistema d'emmagatzematge i visualització de les dades que la paperera es capaç de mesurar i enviar. És a dir, creació d'un servidor, la base de dades que inclourà, el disseny de tots els mòduls que permetran la visualització de les dades...

Tot això ha estat pensat y dissenyat per a que puguin incorporar més papereres a la infraestructura i així fer un control de tot un conjunt de papereres com, per exemple, les un aeroport, centre comercial, ciutat... A continuació es mostren certs punts els quals no s'han realitzat però aquest treball pretén ser l'inici per a arribar a confeccionar-los.

- Disseny de la estructura de base de dades per a incorporar les dades de més papereres.
- Adequar la part web del servidor que s'encarrega de visualitzar l'estat de la paperera per a la incorporació de més.
- Realitzar proves amb més d'una paperera, comprovar que la infraestructura del servidor aguanta.
- Realitzar proves amb diferents papereres per a comprovar la funcionalitat del sistema de recollida de dades.

2. Arquitectura del prototipus

En aquest apartat es parlarà sobre l'arquitectura del prototipus, és a dir, els diferents elements que el constitueixen i la seva finalitat dins del prototipus, els usuaris de cadascun, com interaccionaran i els fluxos d'informació que existeixen entre ells. Un cop explicat l'arquitectura del prototipus en alt nivell, s'entrarà amb més detall en els elements explicant la seva arquitectura interna, posada a punt, etc.

Tot i que en aquest treball només s'usa una paperera, l'arquitectura del prototipus s'ha pensat per a que aquest nombre pugui augmentar. És a dir, que moltes papereres enviïn les dades al servidor i l'usuari pugui tractar aquestes. S'haurien de realitzar una sèrie de modificacions a la pagina web del servidor, ja que la finalitat d'aquest treball és controlar només una.

2.1. Estructura global

En la següent figura es mostra la estructura global de tots els elements i les seves connexions.

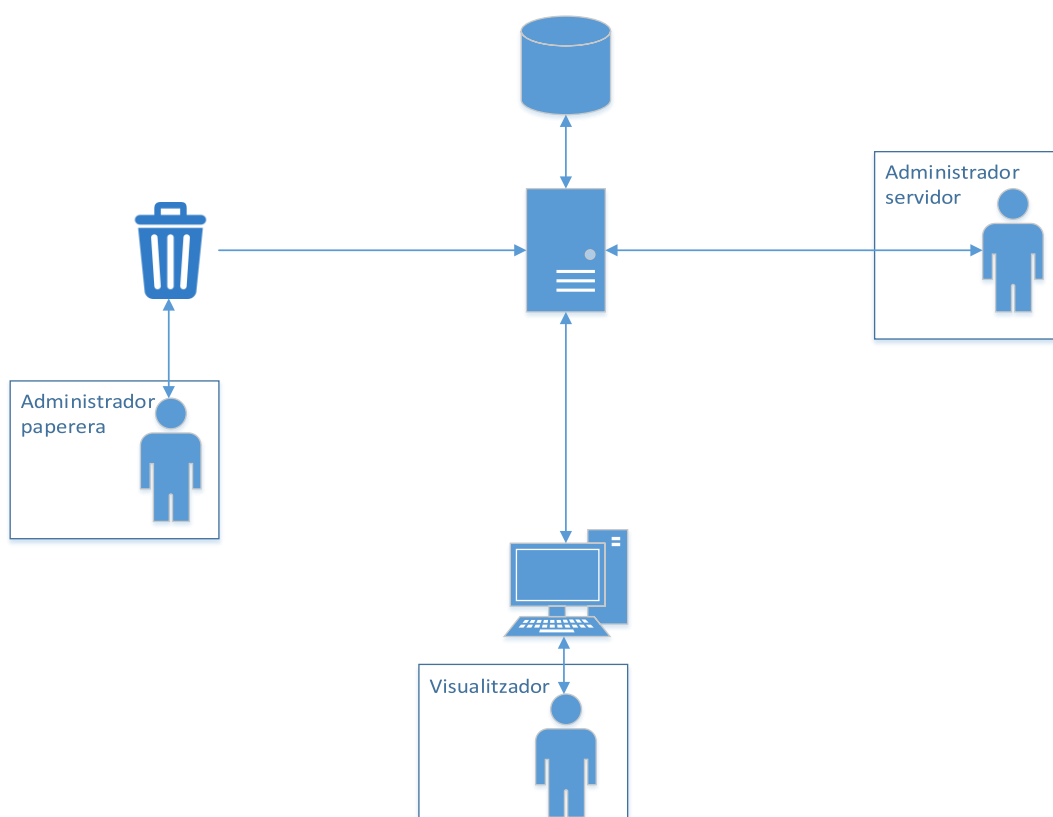


Figura 1 – Estructura global

2.2. Paperera

Aquest element té la funció de captar l'estat de la paperera i enviar-la al servidor. En aquest apartat es parlarà només del conjunt paperera, sense entrar en detall dels elements interns. Sobre aquests es parla més endavant i amb detall al apartat 3.

Per aconseguir això es precisa de l'ús d'una placa microcontroladora que tingui connexió a internet. S'ha pensat en una sèrie de plaques electròniques que podrien complir el objectiu, a continuació es parlarà una mica sobre aquestes i després es decidirà quina serà l'adequada per usar-la al treball.

2.2.1. Arduino

2.2.1.1. Que és Arduino?

Arduino és una plataforma de creació de prototips Open-Source basada en hardware i software easy-to-use¹. Es tracta d'un microcontrolador que permet llegir una sèrie d'inputs – com per exemple la temperatura d'un sensor o l'accionament d'un boto – i convertir-lo en un output – activar una alarma, encendre un LED o fins i tot publicar alguna cosa online. Pots dir-li a la teva placa que ha de fer mitjançant una sèrie d'instruccions, aquestes han d'estar programades mitjançant el llenguatge de programació de l'Arduino, basat en *Wiring*² i enviades a la placa mitjançant el Software Arduino (IDE), basat en *Processing*³.



Figura 2 - Logo Arduino

Durant els anys Arduino ha estat el cervell de milers de projectes, des de objectes quotidians fins a instruments científics complexos. Una comunitat mundial de makers – estudiants, aficionats, artistes, programadors i professionals – s'han reunit al voltant d'aquesta plataforma Open-Source, les seves contribucions han afegit una increïble quantitat de coneixement accessible per a tothom que pot ser de gran ajuda tant a principiants com a experts[1].

¹ Easy-to-use: fàcil d'usar. Terme que serveix per a referir-se que alguna cosa es de fàcil ús.

² Wiring: Infraestructura Open-Source de programació per a microcontroladors. Més informació a la referència bibliogràfica complementaria [1].

³ Processing: Software flexible d'edició de codi. Més informació a la referència bibliogràfica complementaria [2].

Arduino va néixer a Ivrea Interaction Design Institute – a Itàlia[2] – com a una eina fàcil per al prototipatge ràpid, enfocada a estudiants sense experiència en electrònica i programació. Tan aviat com va aconseguir una comunitat més amplia, la placa Arduino va començar a canviar i adaptar-se a les noves necessitats i reptes, diferenciant-se de les simples plaques de 8-bits i adaptant-la per a aplicacions IoT⁴, wearables⁵ i impressió 3D. Totes les plaques Arduino són completament Open-Source, permetent als usuaris crear de forma independent i que puguin adaptar-les a les seves necessitats particulars. El programari també és Open-Source i esta creixent a través de les contribucions dels usuaris a tot el món.

Tot i que hi ha un número elevat de diferents plaques Arduino, a continuació es parlarà només de dues, que són amb les que s'ha treballat en aquest treball de fi de grau. Aquestes són la placa Arduino UNO i la Arduino Yún.

2.2.1.2. Arduino UNO

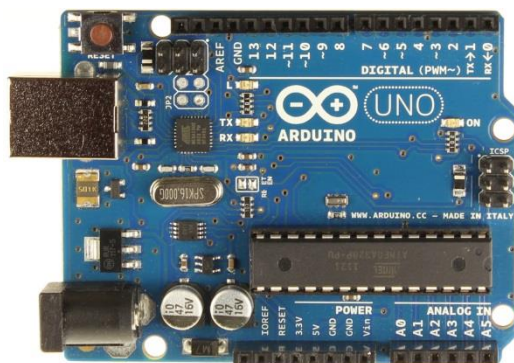


Figura 3 - Placa Arduino UNO

Arduino UNO és una placa electrònica basada en ATmega328P. Té 14 pins digitals que poden treballar de inputs o outputs – 6 d'aquests a més a més com a PWM outputs – , 6 inputs analògics, un cristall de quars de 16MHz, una connexió USB, un connector d'alimentació, connexió ICSP⁶ i botó de reinici [3].

⁴ IoT: Internet of Things. Xarxa d'objectes de la vida quotidiana connectats entre sí [4].

⁵ Wearables: Dispositiu electrònic que es porta a sobre, sota o inclòs a la roba i que està sempre encès [5].

⁶ ICSP: In Circuit Serial Programming. Pins que serveixen per a programar el BootLoader, és a dir, amb aquests podem carregar els programes directament sense ajuda de cap programa extern [6].

En la següent taula es mostren les especificacions tècniques d'aquesta placa:

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Taula 1 - Especificacions tècniques Arduino UNO

Es tracta de la placa més utilitzada y la primera que es va treure al mercat. és una placa molt robusta i la millor opció per endinsar-se aquest món de l'electrònica i la programació.

2.2.1.3. Arduino Yún

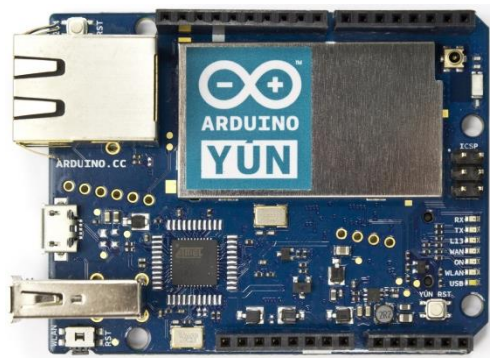


Figura 4 - Placa Arduino Yún

Arduino Yún és una placa electrònica basada en ATmega32u4 i Atheros AR9331. El processador Atheros suporta la distribució Linux basada en OpenWrt⁷ anomenada OpenWrt-Yun. La placa suporta Ethernet i WiFi, té un port USB-A, una ranura per targeta micro-SD, 20 pins digitals input/output (7 d'aquests poden ser utilitzats com a outputs de PWM i 12 com a inputs analògics), un cristall de quars de 16 MHz, una connexió micro USB, connexió ICSP i 3 botons de reinici.

Aquesta placa es distingeix de les altres en que es pot comunicar amb la distribució Linux que porta la placa, oferint un ordinador en xarxa de gran abast per usar-lo amb l'Arduino. A més dels comandaments de Linux, pots escriure els teus propis scripts de Shell i Python per a fer projectes més complexos.

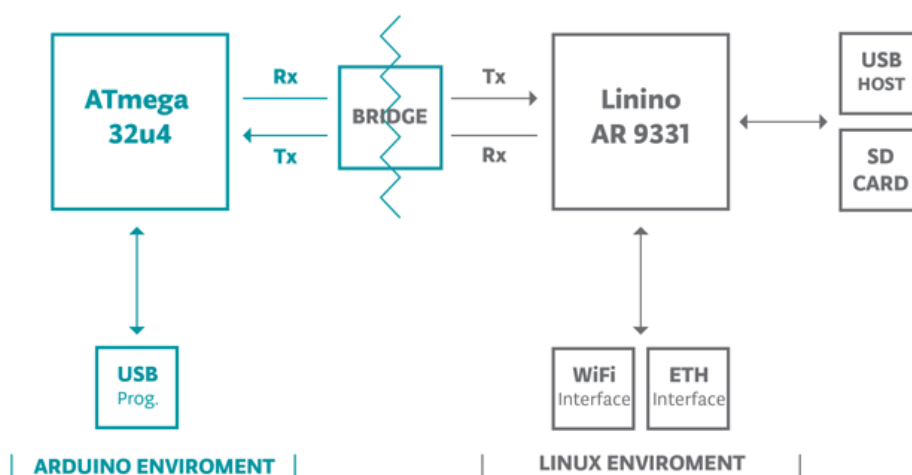


Figura 5 - Comunicació Arduino-Linux

La llibreria Bridge facilita la comunicació entre els dos processadors, donant la habilitat de poder executar scripts de Shell des de codi Arduino, comunicar-se amb networks, rebre informació del processador AR9331. Gracies a aquesta comunicació podem transmetre informació des de el entorn Arduino al entorn Linux i viceversa [7].

Es tracta d'una placa més completa que l'Arduino UNO, amb la qual es poden desenvolupar projectes més complexos. Com disposa de dos processadors, a continuació es mostren les especificacions tècniques d'ambdós processadors:

⁷ OpenWrt: firmware basat en la distribució de Linux principalment usat en routers [8].

Microcontroller	ATmega32U4
Operating Voltage	5V
Input Voltage	5
Digital I/O Pins	20
PWM Channels	7
Analog Input Pins	12
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (of which 4 KB used by bootloader)
SRAM	2.5 KB
EEPROM	1 KB
Clock Speed	16 MHz

Taula 2 - Especificacions tècniques entorn Arduino

Processor	Atheros AR9331
Architecture	MIPS @400MHz
Operating Voltage	3.3V
Ethernet	IEEE 802.3 10/100Mbit/s
WiFi	IEEE 802.11b/g/n
USB Type-A	2.0 Host
Card Reader	Micro-SD only
RAM	64 MB DDR2
Flash Memory	16 MB
SRAM	2.5 KB
EEPROM	1 KB
Clock Speed	16 MHz
PoE compatible 802.3af card support	See <i>Power</i>

Taula 3 - Especificacions tècniques entorn Linux

2.2.2. Raspberry

2.2.2.1. Que és Raspberry Pi?

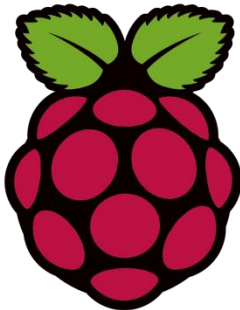


Figura 6 - Logo Raspberry Pi

Raspberry Pi és un ordinador de baix cost, de la mida d'una targeta de crèdit, que es connecta a un televisor o un monitor d'ordinador i utilitza un teclat i ratolí estàndard. És un petit dispositiu que permet a persones de totes les edats explorar la computació i aprendre a programar codi Scratch⁸ i Python⁹. Es capaç de fer tot el que s'espera d'un ordinador convencional, des de la navegació per Internet i reproducció de vídeo d'alta definició, a l'ús de fulls de càlcul, processador de textos i fins i tot jugar a jocs.

Desenvolupat al Regne Unit per la Fundació Raspberry Pi¹⁰ amb l'objectiu d'estimular l'ensenyança de las ciències de la computació a les escoles. La Raspberry Pi té la capacitat d'interactuar amb el món exterior i actualment s'ha utilitzat en una ampla varietat de projectes digitals, des de maquines de música i controladors de continguts infantils a la xarxa per a pares fins a estacions meteorològiques.

Es tracta d'un producte amb propietat registrada però d'ús lliure. Això permet un control de la plataforma però alhora permetent el seu ús lliure tant a nivell educatiu com particular. Pel que fa el software si que és Open-Source sent el seu sistema operatiu una versió adaptada de Debian¹¹, denominada RaspBian, encara que permet altres sistemes operatius com Windows 10 [9] [10].

Ara es mostrarà breument informació de la última placa Raspberry Pi que ha sortit al mercat, la Raspberry Pi 3.

⁸ Scratch: Llenguatge de programació enfocada per a nens desenvolupat per el Institut Tecnològic de Massachusetts (MIT) [11].

⁹ Python: Llenguatge de programació d'alt nivell Open-Source [12].

¹⁰ Fundació Raspberry Pi: Fundació creada al 2009 per a la promoció de l'estudi de las ciències de la computació a les escoles i responsable de crear la Raspberry Pi [13].

¹¹ Debian: Sistema operatiu Open-Source per als ordinadors.

2.2.2.2. Raspberry Pi 3

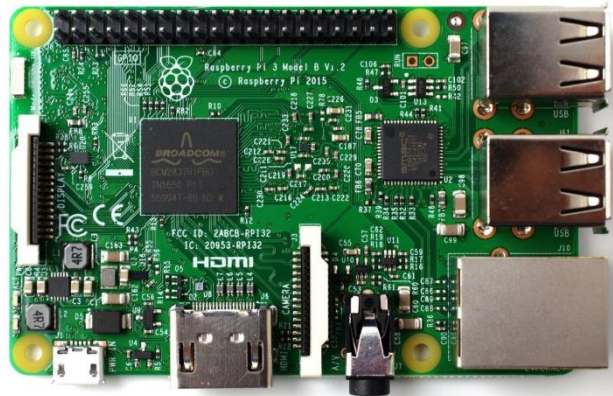


Figura 7 - Raspberry Pi 3

Raspberry Pi 3 és la tercera generació de Raspberry Pi. Reemplaçant l'anterior model (Raspberry Pi 2) al febrer del 2016. És el mini ordinador amb més prestacions de la gamma i totalment recomanable per al desenvolupament de nous projectes i per a l'estimulació de l'ensenyança de las ciències de computació a les escoles.

Desenvolupat per estimular i ajudar en l'ensenyança de la programació i la informàtica, també és un fantàstic punt de partida per el desenvolupament de projectes de IoT¹². El seu baix cost i la naturalesa plug-and-play¹³ de la Raspberry Pi la fa accessible a tothom i disposa de nombroses opcions de connectivitat. Eina perfecte per a experimentar, per utilitzar-la com a ordinador d'escriptori, centre multimèdia, servidor o fins i tot dispositiu de vigilància / seguretat dins de casa [14].

A continuació es mostren les especificacions tècniques de la Raspberry Pi 3:

¹² IoT: Ídem referencia número 5.

¹³ Plug-and-play: terme que es refereix a un dispositiu informàtic, el qual al ser connectat a l'ordinador ja funciona, sense la necessitat de configurar-lo mitjançant softwares específics.

Board	Raspberry Pi 3 Model B
Processor	Broadcom BCM2837
CPU Core	Quadcore ARM Cortex-A53, 64Bit
Clock Speed	1.2GHz (Roughly 50% faster than Pi2)
RAM	1 GB
GPU	400 MHz VideoCore IV®
Network Connectivity	1 x 10 / 100 Ethernet (RJ45 Port)
Wireless Connectivity	802.11n wireless LAN (WiFi) and Bluetooth 4.1
USB Ports	4 x USB 2.0
GPIOs	2 x 20 Pin Header
Camera Interface	15-pin MIPI
Display Interface	DSI 15 Pin / HDMI Out / Composite RCA
Power Supply (Current Capacity)	2.5 A

Taula 4 - Especificacions tècniques Raspberry Pi 3

2.2.3. Elecció de la placa

En aquest apartat s'estudiarà quina és la millor placa per a la realització d'aquest prototipus. S'utilitzarà un mètode après en l'assignatura Gestió de projectes, cursada en aquest grau, amb el qual serveix d'ajuda en la presa de decisions dins dels projectes de prototipatge. És la **matriu de decisió [17]**.

L'ús de matriu de decisió és molt simple conceptualment i es poden dividir en dos tipus: les matrius de filtrat i les d'avaluació. Les primeres (filtrat) s'usen en general per eliminar aquelles alternatives que no requereixen major consideració, mentre que les segones (selecció) serveixen per a escollir una.

En ambdós casos cal definir una sèrie de criteris basats en la necessitat del treball que seran amb els quals podrem decidir la nostra selecció idònia. Per exemple, el cost sempre ha d'aparèixer com a criteri. La matriu es construeix sempre d'igual forma per als dos tipus, amb els conceptes generalment en columnes i els criteris en files.

En el nostre cas s'ha estudiat la decisió de quina placa usar per al desenvolupament del prototipus, s'usarà directament la matriu de selecció doncs només hi ha tres opcions possibles i la matriu de filtrat no tindria sentit usar-la. Les 3 possibles opcions són :

- Arduino UNO
- Arduino Yún
- Raspberry Pi 3

Per a realitzar la matriu de selecció, a part d'escollir els criteris se'ls ha d'imposar un pes segons la rellevància que tindran aquests sobre el treball. A continuació es defineixen els criteris que s'han usat per a realitzar la matriu de selecció, el pes dins del prototipus i el perquè.

Criteri 1: Connexió a Internet sense fils

Un dels aspectes claus d'aquest projecte, és la necessitat de connexió a Internet per a que la paperera reporti el seu estat al servidor. El pes sobre el projecte serà d'un 30%, ja que és un criteri molt important i sense aquest no es podria dur a terme el prototipus.

Criteri 2: Compatibilitat amb sensors

Si es vol mesurar dades del món real i tractar-les es necessari l'ús de sensors per a extreure aquests. Per tant, es necessari que la nostra placa sigui compatible amb els sensors del mercat. El pes sobre el projecte serà d'un 30% equiparable al criteri de la connexió a Internet.

Criteri 3: Cost

Criteri fonamental, ja que es vol aconseguir un prototip funcional amb els mínims costos possibles associats. El pes sobre el projecte serà d'un 25%, ja que és un dels aspectes claus.

Criteri 4: Especificacions tècniques

Es busca una placa amb unes prestacions bones, per a no tenir mancances alhora de realitzar el projecte, i que aquesta es comporti de manera fluida i sense entrebancs. El pes sobre el projecte serà d'un 10%.

Criteri 5: Dimensions

La placa hauria de ser lo suficientment petita com per a incorporar-la en una paperera i que aquesta no es veies afectada d'una manera dràstica. Per això cal considerar la mida de la placa doncs serà el component més gran dins del nostre prototip. El pes sobre el projecte serà d'un 5%.

La matriu de selecció resulta la següent:

	Arduino UNO	Arduino Yún	Raspberry Pi 3	
Criteri 1: Connexió a Internet sense fils	3	5	5	
Criteri 2: Compatibilitat amb sensors	5	5	3	
Criteri 3: Cost	5	4	3	
Criteri 4: Especificacions tècniques	3	4	5	
Criteri 5: Dimensions	5	4	3	
	4,2	4,6	3,8	TOTAL

Taula 5 - Matriu de selecció

La puntuació es de 1 per a un incompliment total del criteri i 5 per a un total compliment del mateix. A continuació es comentarà de manera breu el perquè de les puntuacions escollides en cada cas.

Sobre el Criteri 1 (Connexió a Internet sense fils), tant l'Arduino Yún com la Raspberry Pi 3 obtenen la màxima puntuació, ja que ambdues posseeixen aquesta característica. Per a que l'Arduino Uno tingui connexió a Internet sense fils, es precisa d'uns mòduls complementaris a aquest que li permetin obtenir aquesta característica, per això la puntuació de 3 en aquesta placa.

Sobre el Criteri 2 (Compatibilitat amb sensors), ambdues plaques Arduino tenen una gamma molt amplia de sensors d'arrere cosa que les fa molt atractives per aquest projecte i per això tenen la màxima puntuació. En canvi la Raspberry Pi no té aquesta gamma de sensors tan amplia com les plaques Arduino i molts s'han de configurar prèviament per a que funcionin correctament a la Raspberry Pi, per això no obté la màxima puntuació.

Sobre el Criteri 3 (Cost), el cost més baix correspon a la de l'Arduino Uno amb un preu de 20,57 € obtenint per tant la màxima puntuació en aquest criteri, després ve la Raspberry Pi 3 amb un preu de 50,22 € i finalment el Arduino Yún amb un preu de 75,02 €. Cal però destacar que la placa Arduino Uno sola no tindria el requisit imprescindible de connexió a Internet sense fils, i la compra dels mòduls apart per a proporcionar-li aquest requisit suposaria un augment del preu a casi uns 60 €. Es per això que no li donem les puntuacions més baixes a les altres plaques tot i doblar i inclús triplicar el cost de la primera.

Sobre el Criteri 4 (Especificacions tècniques), pel que respecte al tema de les especificacions tècniques, sense cap dubte la Raspberry Pi és la que té les més potents per això té la màxima puntuació, li segueix la placa Yún i finalment la UNO. No obté aquesta última la mínima puntuació, ja que tot i tenir menors prestacions podria arribar a suportar el projecte.

Sobre el Criteri 5 (Dimensions), totes tres plaques són de dimensions bastant reduïdes, però la de menors dimensions i per tant la que obté la màxima puntuació es la UNO, seguida de la Yún i després la Raspberry Pi.

La placa més adequada per tant resulta la Arduino Yún, la més puntuada usant la matriu de selecció, doncs serà la que s'usarà en tot moment per a desenvolupar aquest projecte.

2.2.4. Usuaris i connexions

La paperera té un únic usuari, anomenat administrador, que s'encarrega del manteniment d'aquesta. Referint-se a manteniment, s'entén a que aquest es entenedor de la tecnologia Arduino i per tant sap accedir al codi, realitzar modificacions i solucionar els problemes que presenti. També és l'encarregat de la posada a punt de la paperera.

La paperera té dues connexions:

- Una unidireccional amb el servidor. La qual mitjançant un POST¹⁴ s'envien les dades registrades pels sensors al servidor i aquest les diposita a la base de dades. Si tots els elements del mapa de connexions estan a la mateixa xarxa, el POST es realitza a la IP privada del servidor, d'aquesta manera la paperera es connecta directament al servidor. D'altra banda si les papereres no es troben a la mateixa xarxa que el servidor, el POST es realitza a la IP pública del servidor ja que aleshores la connexió passa a través dels routers de les diferents xarxes.

¹⁴ POST: mètode http per a enviar dades i poder-les processar per alguna finalitat [18].

- Una bidireccional amb l'usuari administrador. La qual mitjançant aquesta l'administrador pot accedir a la paperera, realitzar el manteniment i la posada a punt. La connexió al codi es du a terme amb el software Arduino IDE, sí es vol accedir a la part Linux, mitjançant un browser et pots connectar a aquesta i configurar-la al teu gust. Es parlarà sobre la posada a punt de la paperera en l'apartat 3.2.

2.3. Servidor

Aquest element té com a finalitat emmagatzemar les dades enviades per la paperera i també permetre la visualització d'aquestes. En aquest apartat es parlarà només del conjunt servidor, sobre l'arquitectura interna del servidor es parla més endavant i amb detall al apartat 4.

Per a la creació del servidor s'ha pensat en un molt famós anomenat Apache2. Com que ha de disposar de una base de dades i una part web aquest servidor compleix l'objectiu perquè existeix un model de conjunt de solucions anomenat LAMP que ho permet. A continuació es farà una petita introducció de que consisteixen.

2.3.1. Servidor Apache

Per a l'elaboració d'aquest treball es posarà en marxa un servidor Apache2, a continuació de manera informativa es parla sobre aquest servidor HTTP¹⁵.

Va ser llançat el 1995 i ha estat el servidor web més popular des de l'abril del 1996. És



Figura 8 - Logo Apache Server

Open-Source, sobretot usat per a les plataformes Unix, però també compatible amb Microsoft Windows i Macintosh entre d'altres. Aquest servidor es desenvolupa i mantingut per la seva comunitat d'usuaris sempre sota la supervisió de la Apache Software Foundation dins del projecte

HTTP Server (httpd) [15].

És tracta d'un servidor molt complet, el qual complementa el seu core amb diferents mòduls afegibles. S'utilitza principalment per a webs tant de contingut estàtic com dinàmic i és el component principal del popular paquet LAMP, que també s'usarà en aquest treball.

¹⁵ Servidor HTTP: és el mateix que servidor web.

2.3.2. LAMP

LAMP és un model de conjunt de solucions de serveis web, LAMP correspon al acrònim dels noms dels seus components [16]:

- Linux: com a sistema operatiu.
- Apache HTTP Server: com a servidor web.
- MySQL: com a sistema de gestió de base de dades.
- PHP: com a codi de programació.

Durant l'explicació d'aquest treball es parlarà sobre la configuració d'aquest conjunt de solucions de serveis web.

2.3.3. Usuaris i connexions

El servidor disposa de dos usuaris:

- Administrador: usuari que té coneixement de l'estructura del servidor y de la programació d'aquest (HTML, PHP, JavaScript, MySQL). S'encarrega del manteniment d'aquest i de la posada apunt.
- Visualitzador: usuari que mitjançant un sobretaula, portàtil, mòbil o tablet podrà accedir al servidor i visualitzar l'estat actual de la paperera. També podrà visualitzar gràficament per dies les diferents mesures dels sensors.

Existeixen 3 connexions al servidor:

- Una unidireccional amb la paperera. Connexió explicada al apartat 2.2.4.
- Una bidireccional amb l'usuari Visualitzador. La qual mitjançant aquesta l'usuari Visualitzador pot veure l'estat actual de la paperera i les diferents lectures dels sensors per dies, mostrades en gràfiques. Aquesta connexió a la web del servidor es realitza mitjançant un browser i usant la IP pública del servidor si l'usuari es troba a una xarxa diferent, o la IP privada si l'usuari es troba en la mateixa xarxa. Cal destacar que si es troba en la mateixa xarxa, també es pot accedir al servidor amb la IP pública però realitza un trajecte innecessari del router a internet i tornar al mateix router.

- Una bidireccional amb l'usuari Administrador. La qual mitjançant aquesta l'usuari Administrador pot accedir a l'estructura del servidor, realitzar la posada a punt, administrar la pagina web i la base de dades que conte el servidor. La connexió a aquest servidor es fa mitjançant un laptop, el qual treballa com a servidor. És l'única manera d'accedir al servidor en el nostre cas.

3. Arquitectura de la Paperera

En aquest apartat es vol explicar de una manera detallada l'arquitectura de la paperera: de quins elements esta composta, com és la posada a punt d'aquesta, el codi de programació que executa, i els fluxos d'informació interns entre els elements.

3.1. Elements

La paperera estarà formada per:

- 1 Placa Arduino Yún
- 3 Sensors de distancia per ultrasons
- 1 Sensor de temperatura i humitat
- 1 Resistència de 10kΩ
- 1 Protoboard (no necessària)
- Cables de connexió

A continuació es donarà una petita informació sobre els sensors utilitzats per a introduir-los:

3.1.1. Sensor de distancia d'ultrasons HC-SR04

Per l'elaboració d'aquest treball, s'usarà el sensor d'ultrasons HC-SR04¹⁶. Es tracta d'un sensor que es capaç de detectar objectes i calcular la distancia a la que es troben dins d'un rang entre 2 a 400 cm. El sensor funciona per ultrasons i conté tota la electrònica encarregada de fer la mesura. El seu ús és tan senzill com enviar un puls d'inici i mesurar el puls de retorn. Destaca per el seva mida, gran precisió i pel baix cost i consum que té.

¹⁶ Per una informació amb més detall, consultar la seva fulla d'especificacions. Correspon a l'Annex A.



Figura 9 - Sensor HC-SR04

El sensor disposa de 4 pins de connexió:

- VCC.
- Trig (Sortida de l'ultrasò).
- Echo (Recepció de l'ultrasò).
- GND.

Característiques del sensor:

- Dimensions: 4,5 x 2,0 x 1,5 cm.
- Tensió d'alimentació: 5 V.
- Freqüència de treball: 40KHz.
- Rang màxim: 400 cm.
- Rang mínim: 2 cm.
- Duració mínima del puls d'inici: $1 \cdot 10^{-5}$ s.

3.1.2. Sensor d'humitat i temperatura DHT22

Per l'elaboració d'aquest treball, s'usarà el sensor d'humitat i temperatura DHT22¹⁷. Aquest utilitza un sensor capacitiu d'humitat i un termistor per mesurar l'aire del seu voltant. Mostra les dades mitjançant una senyal digital, és de fàcil ús. L'únic inconvenient que presenta és que només pot prendre dades cada 2 segons, per tant s'ha de tenir en compte en el moment de programar-lo per a no obtenir valors erronis.

¹⁷ Per a més informació consultar el seu full d'especificacions. Correspon a l'Annex B.

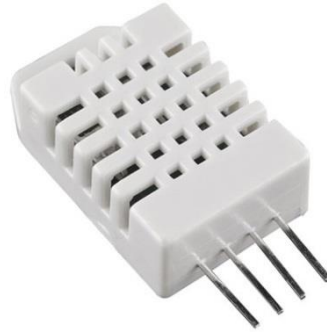


Figura 10 - Sensor DHT22

El sensor disposa de 4 pins de connexió:

- VCC.
- DATA.
- NC.
- GND.

Característiques del sensor:

- Tensió d'alimentació: 3,3 – 6 V.
- Rang de mesura (temperatura): -40°C a 80°C.
- Precisió (temperatura): $< \pm 5^{\circ}\text{C}$.
- Resolució (temperatura): 0,1 °C.
- Rang de mesura (humitat): 0% a 100% RH.
- Precisió (humitat): 2% RH.
- Resolució (humitat): 0,1% RH.
- Temps entre mesures: 2 s.

3.2. Posada a punt

A continuació es parlarà sobre la posada a punt de tots aquests elements, més concretament sobre la configuració de la placa Arduino Yún i les connexions entre els diferents elements.

3.2.1. Configuració Arduino Yún

En aquest apartat es vol mostrar els passos a seguir per a configurar el Arduino Yún i deixar-lo completament operatiu per a la realització del projecte, així com la instal·lació del software que permet afegir noves llibreries per a ser usades en el codi i enviar els scripts a la placa per a que els executi.

3.2.1.1. Instal·lació Software Arduino IDE

Primer de tot es començarà per explicar la instal·lació del Software Arduino IDE al ordinador, ja que posteriors passos es necessari l'ús d'aquest. Gracies a aquest es pot enviar els scripts a la placa, que aquesta els executi i així posar en marxa el projecte [7].



Figura 11 - Software Arduino IDE

1. Accedir a la següent web :

<http://www.arduino.cc/en/main/software>

2. Descarregar-se l'última versió del Arduino IDE disponible per al sistema operatiu pertinent. En aquest cas es descarregarà l'instal·lador per a Windows.

Download the Arduino Software



Figura 12 - Instal·ladors Software Arduino

3. Un cop descarregat el software Arduino IDE, realitzar la seva instal·lació fent un doble clic en la icona d'instal·lació.

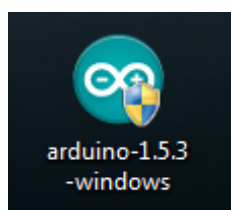


Figura 13 - Icona d'instal·lació del Arduino IDE

4. Llegir el acord de llicència i acceptar-lo.

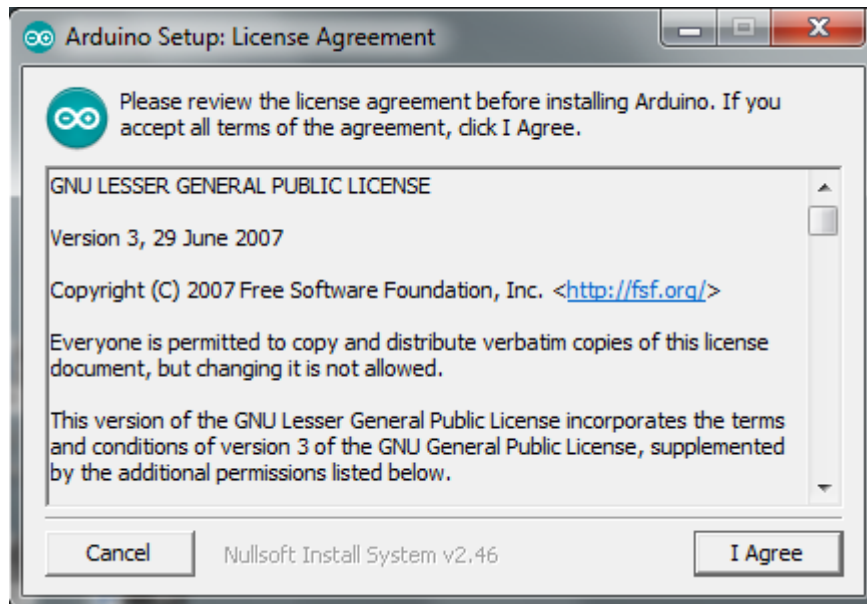


Figura 14 - Acord de llicència

5. Seleccionar els components a instal·lar. Per defecte estan inclosos tots que són els que es necessitaran per a la realització del treball.

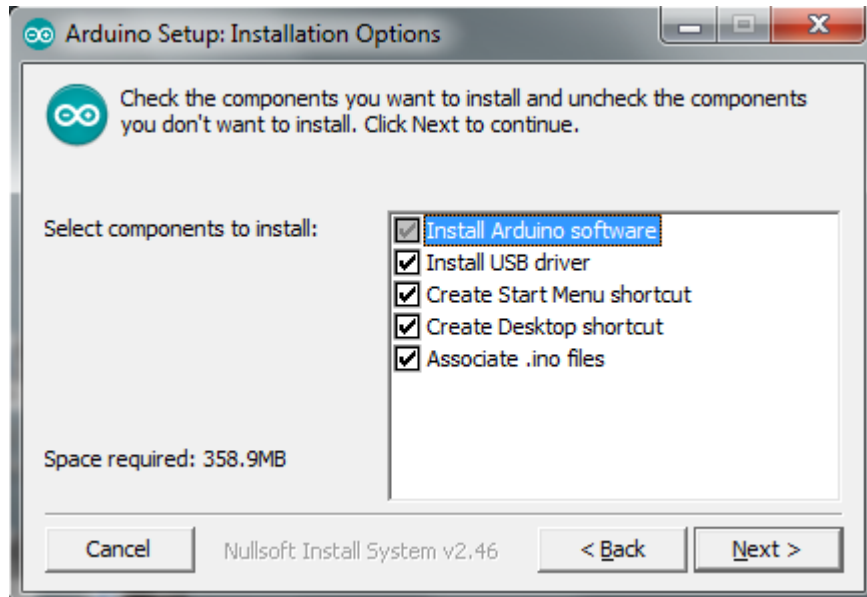


Figura 15 - Components a instal·lar

6. Selecciona la ubicación on es vol instal·lar el software.

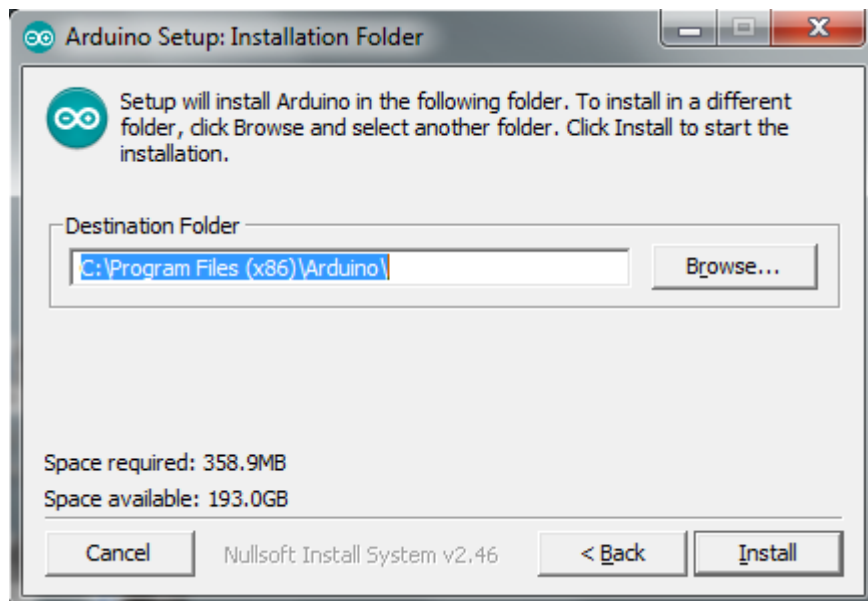


Figura 16 - Selecció de la ubicació

7. El instal·lador mostrarà l'estat del procés. Com que s'ha seleccionat que els divers també s'instal·lin apareixeran una sèrie de pop-ups¹⁸ que haurem de confirmar la seva instal·lació.

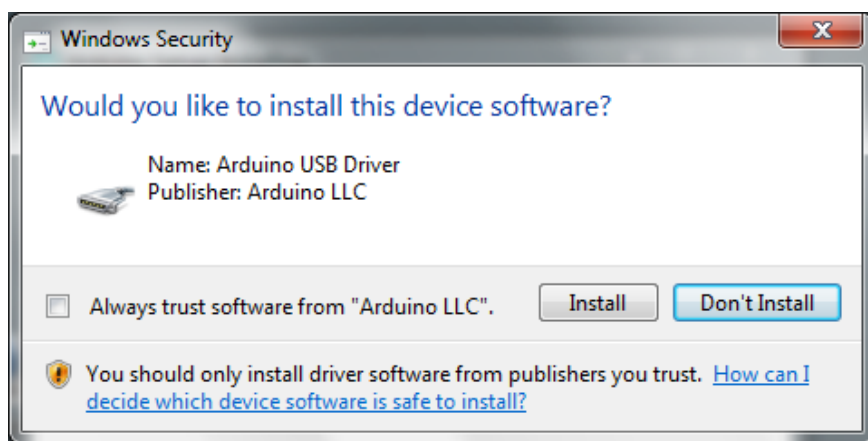


Figura 17 - Pop-up d'instal·lació dels drivers

8. Instal·lació completada.

¹⁸ Pop-up: Terme angles que significa finestra emergent. Usat en el llenguatge informàtic.

3.2.1.2. Afegir llibreria del sensor de temperatura DHT22

Existeix una llibreria de codi per al sensor de temperatura, aquesta cal descarregar-la i afegir-la al Software Arduino IDE per a poder usar-la en el codi ja que no es troba en les llibreries que venen per defecte amb el programa. Aquesta ens permet definir el sensor, encendre'l i calcular la temperatura i humitat. Ara s'exposaran els passos a seguir per afegir aquesta llibreria.

1. Descarregar-se la llibreria de la web següent:

<https://github.com/adafruit/DHT-sensor-library>

2. Descomprimir l'arxiu zip i anomenar la carpeta a "DHT".
3. Ubicar la carpeta descomprimida a la ruta següent:

/Documentos/Arduino/libraries

4. Un cop ubicada en la ruta esmentada ja es pot usar en els posteriors codis.

Aquest mateix procés serveix per afegir qualsevol altre llibreria que es necessites usar per al treball.

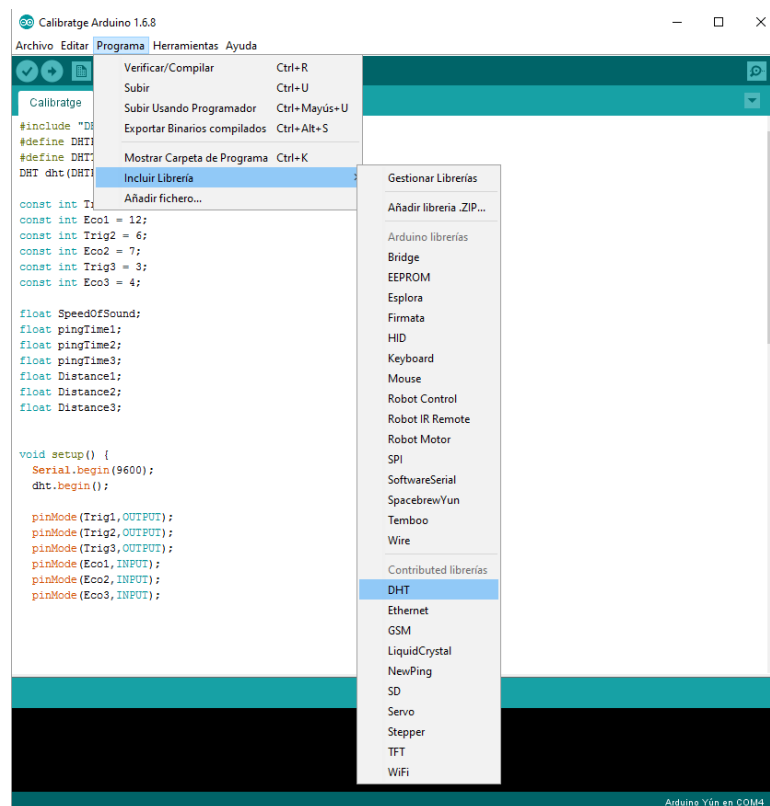


Figura 18 - Llibreria DHT instal·lada

3.2.1.3. Connectar la placa Arduino Yún a la xarxa

La placa Arduino Yún s'ha de trobar a la mateixa xarxa que el ordinador des de el qual es vol treballar, ja que serà el nexa d'unió per a enviar els scripts procedents del Software Arduino IDE [7].

El primer cop que es vol accedir a una xarxa, o cada cop que es vol canviar aquesta, s'ha de configurar per a que es connecti. La placa per si sola, si no es connecta a cap xarxa automàticament, ja sigui per que es el primer cop en connectar-la o per que no ha trobat la xarxa configurada anteriorment, crearà una xarxa WiFi anomenada *ArduinoYun-XXXXXXXXXXXX*.

Mitjançant aquesta es pot configurar a quina xarxa es connectarà la placa. Des de el ordinador, ens hem de connectar a la xarxa WiFi i mitjançant qualsevol browser accedir a la web <http://arduino.local>. Apareixerà la següent imatge:

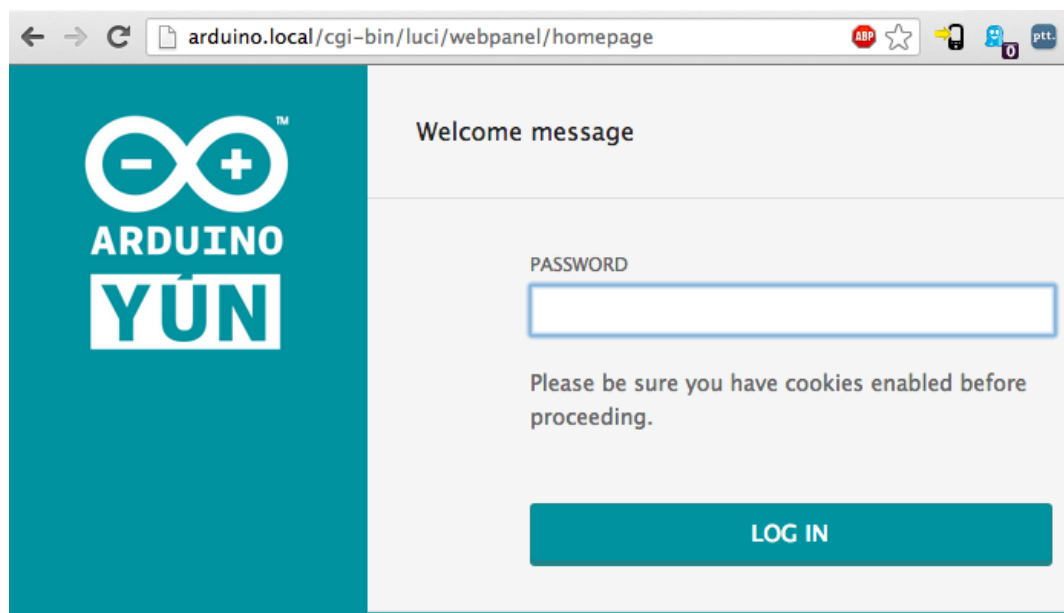


Figura 29 - Log In per a configurar les xarxes

Si es el primer cop que s'entra, la contrasenya es "arduino". Aleshores un cop accedit es prem el boto "CONFIGURE" com podeu observar en la figura següent i es procedeix a configurar la connexió desitjada i també alguns aspectes de la placa com el nom, contrasenya i la zona horària.

Figura 203 - Configuració xarxes Arduino Yún

Un cop realitzats els canvis pertinents i seleccionada la xarxa a la qual es vol connectar es prem el botó “CONFIGURE & RESTART” i la placa es reiniciarà, aplicarà els canvis realitzats i es connectarà a la xarxa que s’ha indicat.

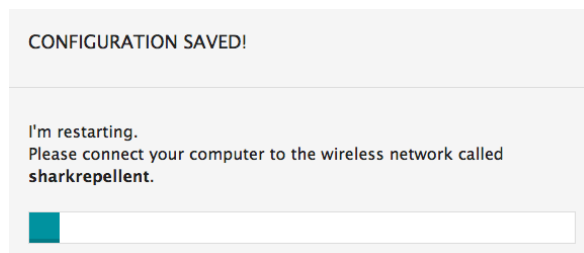


Figura 21 - Configure & Restart

3.2.1.4. Actualització del OpenWrt-Yun Image

Ara que ja es disposa del Software Arduino IDE instal·lat, es pot procedir a l'actualització del firmware de la placa, el OpenWrt-Yun Image. Per aquest pas es necessari l'ús d'una targeta microSD [19]:

1. Primer de tot descarregar-se el arxiu zip de la següent web:

<http://www.arduino.cc/en/main/software#toc8>



Figura 4 - Arxiu a descarregar

2. Descomprimir l'arxiu i moure la imatge binària a la targeta microSD. Aquesta ha de tenir un nom semblant al següent:

Openwrt-ar71xx-generic-yun-16M-squashfs-stupgrade.bin

3. Mitjançant el Software Arduino IDE, pujar el script YunSerialTerminal de la base d'exemples.

Archivo → Ejemplos → Bridge → YunSerialTerminal

4. Inserir la targeta microSD a la placa Arduino Yún un cop pujat el script.

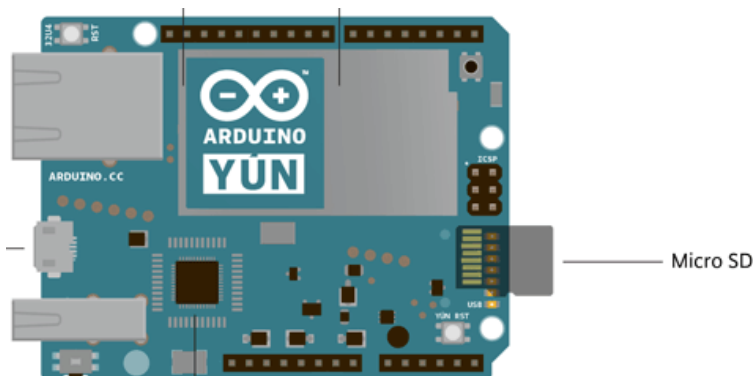


Figura 23 - Ranura microSD

5. Assegurar-se que la placa Arduino Yún es troba a la mateixa xarxa que l'ordinador.
6. Accedir a la placa des de un browser usant el link següent <http://arduino.local>. Introduir la contrasenya fixada en l'apartat 3.2.1.3, aleshores es mostrarà la següent pantalla:

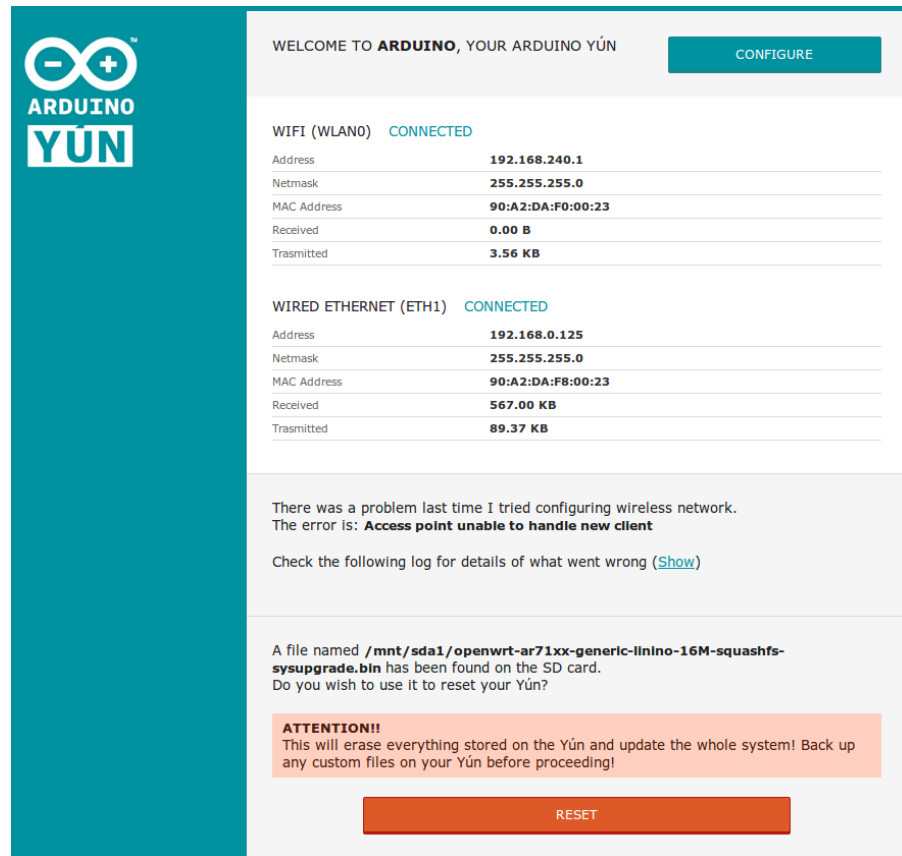


Figura 24- Pantalla de configuració Arduino Yún

7. Un cop es mostra la pantalla de la Figura 24, prémer el botó de "RESET" per a realitzar l'actualització de la imatge OpenWrt-Yun. Aquesta operació pot trigar uns minuts.
8. Quan finalitzi la actualització, s'ha de repetir el punt 3.2.1.3 un altre cop per a connectar-se a la xarxa.

3.2.1.5. Expandir el espai en disc de la placa

Aquest pas és opcional però recomanable, ja que el OpenWrt-Yun ocupa 9 dels 16MB disponibles de memòria flash de la placa. Això significa que es pot quedar fàcilment sense espai si instal·les software en aquest. Per tant explicarem com configurar una targeta microSD per a que esdevingui el nou sistema d'arxius per al OpenWrt-Yun [20].

Per a realitzar aquest procés, que no és molt complex però que requereix uns coneixements bàsics de Linux, els desenvolupadors de Arduino han automatitzat els passos a seguir i han fet un script. Per tant només cal inserir la targeta microSD, pujar el script a la placa, obrir el Serial Monitor del Software Arduino IDE i seguir els passos que s'indiquen.

Per a realitzar aquest procés, cal que la placa Arduino estigui connectada al ordinador mitjançant cable ja que d'altre manera no es podrà realitzar la connexió per Serial i per tant no funcionarà el Serial Monitor.

1. Primer de tot descarregar el script des de la següent pagina web:

<http://www.arduino.cc/en/tutorial/expandingyundiskspace>

2. Connectar la targeta microSD a la placa Arduino Yún. Cal destacar que la targeta microSD es formatarà, per tant es perdran tots els arxius que contingui.
3. Descomprimir l'arxiu i pujar el script a la placa Arduino Yún mitjançant el software Arduino IDE.
4. Obrir el Serial Monitor del Software Arduino IDE

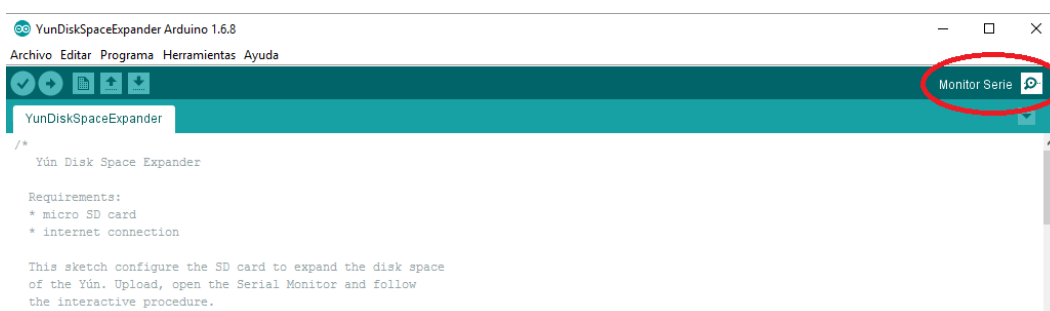


Figura 255 - Serial Monitor

5. Seleccionar del menú desplegable la opció Newline i escriure "yes" per acceptar la instal·lació.

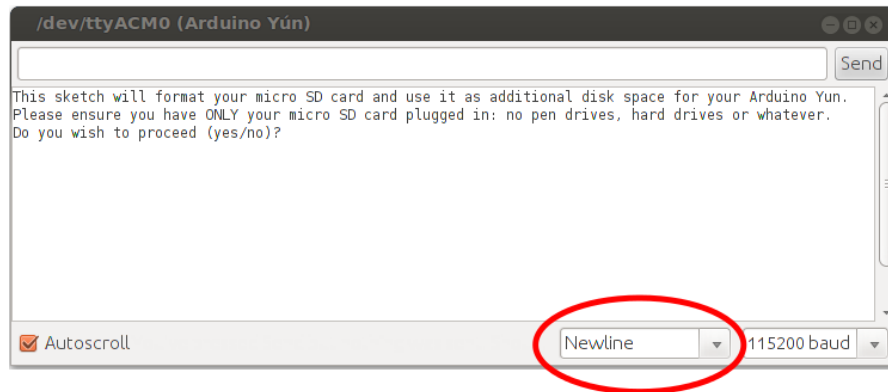


Figura 26 - Seleccionar Newline

6. Escriure "yes" per començar la instal·lació.

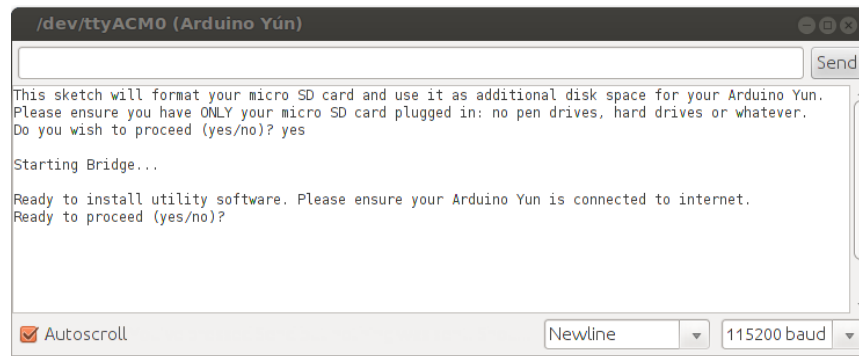


Figura 27 – Començar la instal·lació

7. Escriure "yes" per començar a formatar la targeta microSD.

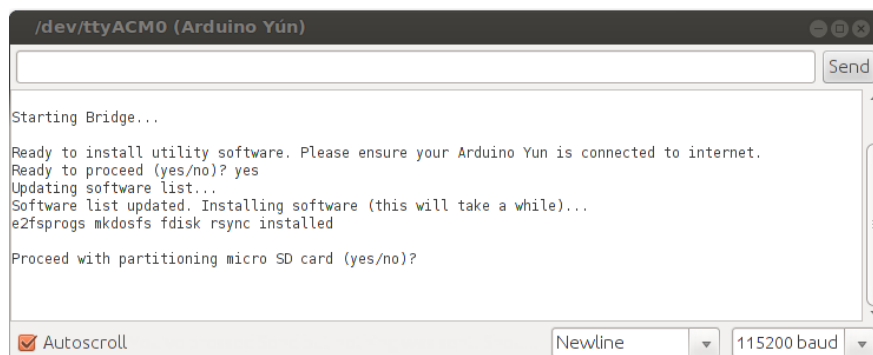
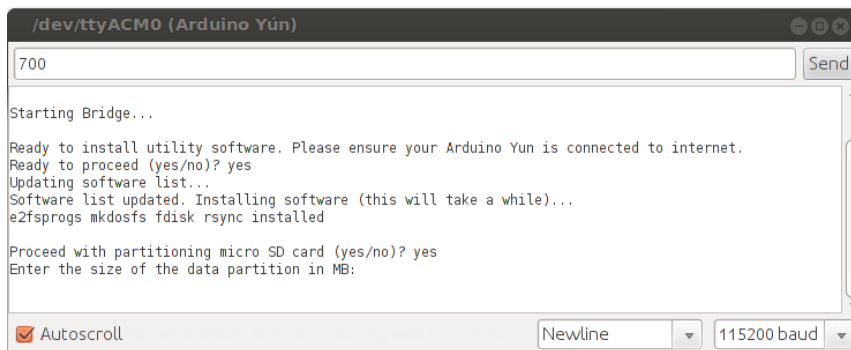


Figura 28 – Començar el format de la targeta

8. Especificar la mida de la primera part de la targeta microSD que volem utilitzar per a compartir arxius amb el nostre ordinador, la segona part estarà dedicada al sistema d'arxius del OpenWrt-Yun.



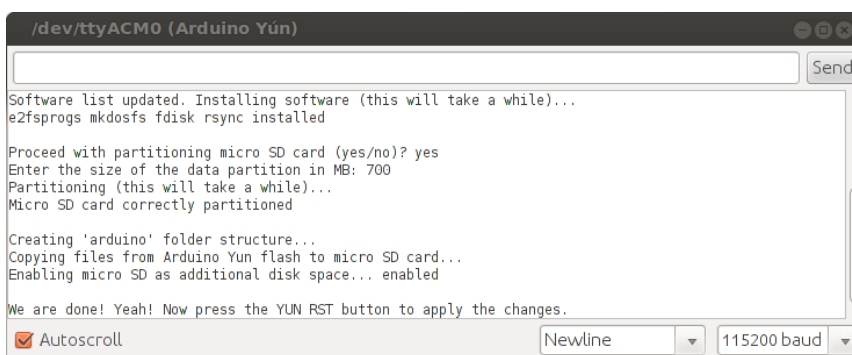
```

/dev/ttyACM0 (Arduino Yun)
700 Send
Starting Bridge...
Ready to install utility software. Please ensure your Arduino Yun is connected to internet.
Ready to proceed (yes/no)? yes
Updating software list...
Software list updated. Installing software (this will take a while)...
e2fsprogs mkdosfs fdisk rsync installed
Proceed with partitioning micro SD card (yes/no)? yes
Enter the size of the data partition in MB:

```

Figura 69 - Especificar mida particions

9. Esperar a que realitzi les particions i finalitzi el script. Aleshores s'haurà ampliat l'espai en disc de la placa.



```

/dev/ttyACM0 (Arduino Yun)
Software list updated. Installing software (this will take a while)...
e2fsprogs mkdosfs fdisk rsync installed
Proceed with partitioning micro SD card (yes/no)? yes
Enter the size of the data partition in MB: 700
Partitioning (this will take a while)...
Micro SD card correctly partitioned
Creating 'arduino' folder structure...
Copying files from Arduino Yun flash to micro SD card...
Enabling micro SD as additional disk space... enabled
We are done! Yeah! Now press the YUN RST button to apply the changes.

```

Figura 307 - Finalització de la expansió de l'espai en disc

El que s'ha realitzat amb aquest script es una partició de la targeta microSD en dues parts. La primera formatada en FAT32¹⁹, que pot llegir-ho qualsevol ordinador, s'usarà per compartir arxius entre l'Arduino i l'ordinador. La segona part és la que contindrà el sistema d'arxius del OpenWrt-Yun.

¹⁹ FAT32: sistema d'arxius per a la versió de 32 bits. S'utilitza en targetes de memòria i dispositius similars.

3.2.2. Connexions

En aquest apartat es centrarà en les connexions entre els diferents elements del prototipus. Tot i que s'han mencionat amb anterioritat els pins de connexió dels sensors en l'apartat 3.1, en aquest es tornaran a mencionar així com els de la placa Arduino Yún i es mostraran les connexions entre aquests.

Pel que respecta el sensor de distància d'ultrasons HC-SR04, els pins es mostren en la següent figura:

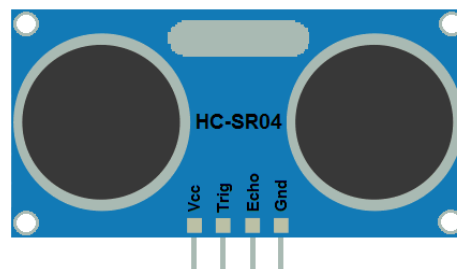


Figura 31 - Pins sensor HC-SR04

- El pin Vcc correspon al de tensió d'entrada i per tant segons la fulla d'especificacions tècniques ha d'anar connectat a 5V.
- El pin Trig correspon al d'activació de sortida d'ultrasons. Quan se li subministra un puls superior a 2.4V amb una amplada de $1 \cdot 10^{-5}$ s el mòdul s'activa i envia 8 ràfegues d'ultrasons a 40kHz. Es necessari l'ús dels pins PWM del Arduino per a controlar la amplada del puls subministrat.
- El pin Echo correspon al d'activació de rebuda d'ultrasons. Quan s'ha generat el puls al pin Trig i envii les 8 ràfegues d'ultrasons, cal activar aquest per a que estigui escoltant per a la rebuda de la senyal de tornada.
- El pin GND correspon al de tensió nul·la i per tant ha d'anar connectat a 0V (GND).

Si parlem del sensor de temperatura i humitat DHT22, els pins es mostren en la següent figura:

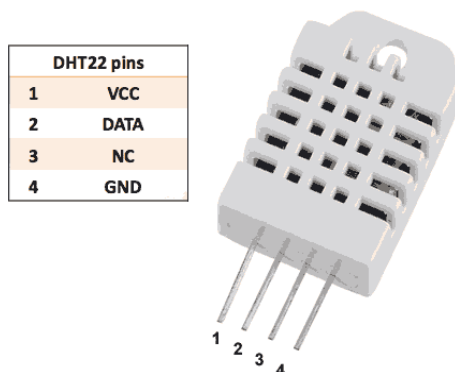


Figura 328 – Pins sensor DHT22

- El pin Vcc correspon al de tensió d’entrada i per tant segons la fulla d’especificacions tècniques ha d’anar connectat a un voltatge entre 3,3 i 5,5V.
- El pin DATA correspon al que envia les dades captades pel sensor a la placa. Per tant ha d’anar connectat a un pin de la placa Arduino que pugui funcionar com a Input.
- El pin NC no s’ha de connectar enlloc.
- El pin GND correspon al de tensió nul·la i per tant s’ha de connectar a 0V (GND).

Cal destacar que entre les connexions del pin Vcc i DATA, per a que el sensor funcioni cal connectar en paral·lel una resistència de 10kΩ. Aquest incís està detallat en la fulla d’especificacions del sensor. En la taula següent es mostren les connexions dels diferents pins dels sensors amb la placa seguit del mapa de connexions.

	Pins			Pins	
	Sensor	Arduino		Sensor	Arduino
Sensor HC-SR04 1	Vcc	5V	Sensor HC-SR04 3	Vcc	5V
	Trig	11		Trig	3
	Echo	12		Echo	4
	GND	GND		GND	GND
Sensor HC-SR04 2	Vcc	5V	Sensor DHT22	Vcc	5V
	Trig	6		DATA	2
	Echo	7		NC	-
	GND	GND		GND	GND

Taula 6 - Connexions entre sensors i la placa

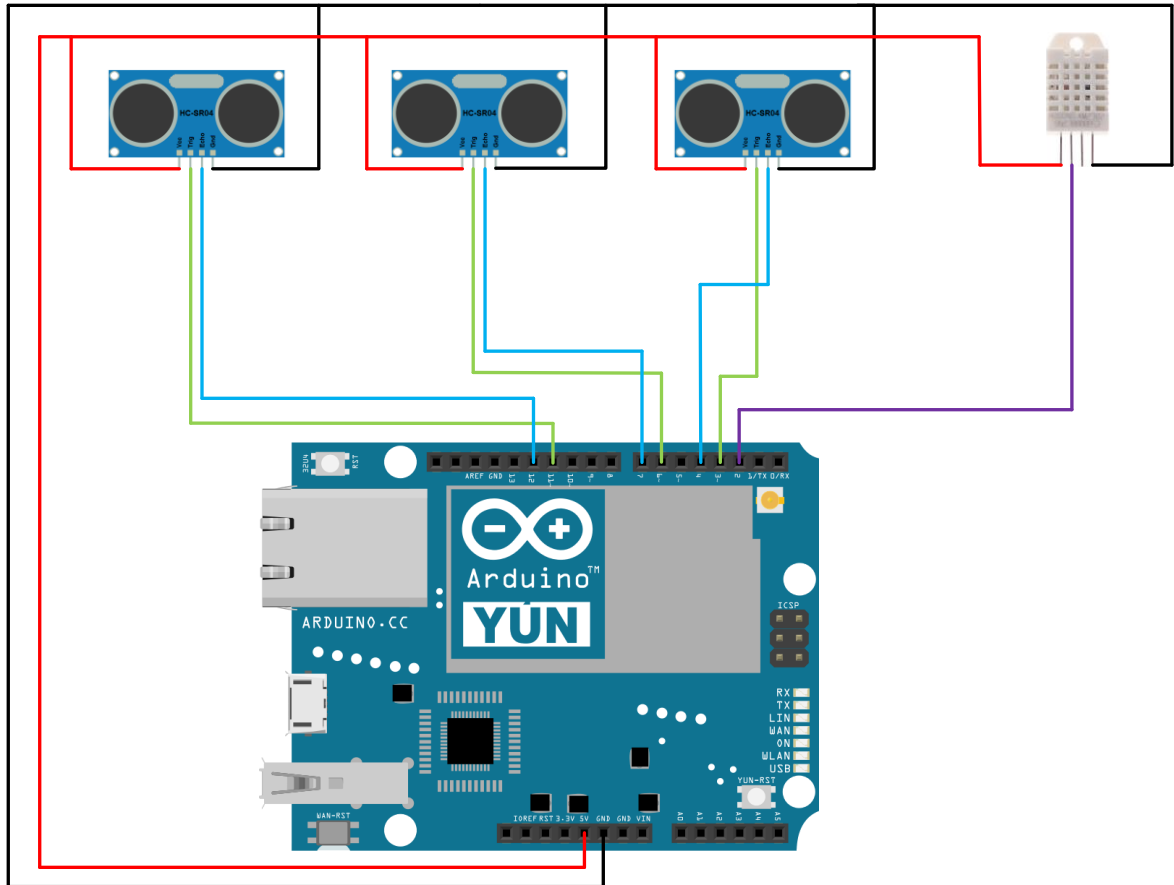


Figura 339 - Mapa de connexions

3.3. Localització dels sensors

Partint de que les proves s'han realitzat en una caixa rectangular, a continuació es mostrarà com s'han decidit col·locar els sensors en aquesta per a realitzar les mesures.

Pel que respecta al sensor de temperatura, aquest no presenta cap inconvenient alhora de decidir on col·locar-lo, ja que mesura la temperatura del interior de la paperera i la variació serà ínfima. Per tant es decideix posar-lo a un costat a la part superior de la paperera.

La col·locació dels sensors de distancia si que s'ha de prendre una decisió, ja que hi ha dues possibles maneres de col·locar-los. Una a la part lateral de la paperera disposant-los un a sobre del altre, per exemple a l'altura que representi el 25%, 50% i al 75% de la capacitat de la paperera i que quan la distancia que mesures fos diferent a una fixada significaria que la paperera estaria plena aquell tant per cent. L'altre és col·locant-los a la part superior de la paperera cobrint el màxim espai possible i que aquests mesurin la distancia fins que trobin un objecte.

Parlant del primer mètode només es disposaria de tantes mesures com sensors de distancia es possessin al lateral, en aquest cas 3. Un altre problema que es podria presentar és que algun objecte tapes més d'un sensor a la vegada i dones una falsa lectura de que esta plena.

Per altre banda col·locar els sensors a la part superior arreglem aquest problema ja que la mesura no es realitza de manera horitzontal sinó vertical i com usem varis sensors, si un té una lectura més elevada que l'altre per culpa d'un objecte, la mesura del volum ocupat total es realitza amb la mitjana de tots els sensors, per tant dona una lectura més acurada del estat d'aquesta.

Pels raonaments exposats es decideix col·locar els sensors de distancia a la part superior de la paperera.

3.4. Codi

En aquest apartat es parlarà del codi que executa la placa Arduino Yún, com esta estructurat i que realitza cada part. Però primer de tot també es parlarà sobre un subcodi que s'ha d'emprar abans de executar l'altre que serveix per a determinar uns certs valors usats al codi principal.

3.4.1. Estructura del subcodi

Aquest sistema pot adaptar-se a papereres de diferents mides, ja que els sensors de distancia HC-SR04 tenen un rang de mesura entre 2 i 400 cm. És aquí entra en joc aquest codi, que serveix per a donar-nos les distancies màximes que enregistren els sensors de distancia. Aquests es bolcaran al codi principal per a que realitzi correctament el càlcul de la capacitat a la que es troba la paperera.

El sensor de distancia d'ultrasons calcula el temps que triga el senyal d'alta freqüència que ha emes a retornar al sensor. Per tant el càlcul de la distancia no és directe, sinó que s'ha d'aplicar una formula:

$$D = \frac{t * V_{so}}{2} \cdot 100 \quad (Eq. 1)$$

On:

- D = Distancia mesurada (cm)
- t = Temps que triga el senyal en tornar (s)
- V_{so} = Velocitat del so (m/s)

Ja que es disposa d'un sensor de temperatura, s'ha decidit fer un càlcul més aproximat a la realitat, ja que la velocitat del so varia en funció de la temperatura i per tant podem agafar la velocitat del so adient a la temperatura del ambient. La velocitat del so varia de la següent manera:

$$V_{so} = 331,5 + 0,6 * T \quad (Eq. 2)$$

On:

- V_{so} = Velocitat del so (m/s)
- T = Temperatura (°C)

Com es necessari l'ús del sensor de temperatura, primer de tot es cridarà la llibreria que existeix per al sensor DHT22 (DHT.h) que prèviament s'ha afegit. A continuació es definirà on esta connectat el pin DATA del sensor a la placa i el tipus de sensor que s'està connectant. Finalment s'usa la funció *dht()* de la llibreria per a definir-lo i poder-lo cridar posteriorment.

```
#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT22;
DHT dht(DHTPIN, DHTTYPE);
```

Figura 3410 – Definició del sensor DHT22

Seguidament es defineixen les constants que s'usaran per a indicar on estan connectats els pins Trig i Echo dels sensors de distancia a la placa Arduino Yún. Aquests serviran per a indicar si han de treballar com a Inputs o Outputs.

```
const int Trig1 = 11;
const int Echo1 = 12;
const int Trig2 = 6;
const int Echo2 = 7;
const int Trig3 = 3;
const int Echo3 = 4;
```

Figura 3511 – Definició constants

Es segueix amb la definició de les variables que s'usaran durant el codi, aquestes no se les defineix com a *const* com a les anteriors, ja que no volem un valor fixe.

```
float SpeedOfSound;
float pingTime1;
float pingTime2;
float pingTime3;
float Distance1;
float Distance2;
float Distance3;
```

Figura 3612 - Definició variables

Els sensors HC-SR04 no calen definir-los d'igual manera com el de temperatura DHT22. Definint com treballaran les connexions Trig i Echo mitjançant els seus pins ja permet fer ús d'aquests. La definició d'aquestes connexions es produeix en el bloc *setup*.

En el bloc *setup* s'han d'incloure totes aquelles coses que només es vol que el codi les executi un cop abans d'entrar en el bloc *loop* que serà el que s'executi contínuament fins que la placa deixi d'estar connectada a la corrent. Per tant, en el bloc *setup* s'inclourà la definició de les connexions Trig i Echo, l'encesa del sensor DHT22 i també del Monitor Serial que servirà per a veure els valors que els sensors estan mesurant, aquests són els valors que més tard es bolcaran al codi principal.

```
void setup() {
  Serial.begin(9600);
  dht.begin();

  pinMode(Trig1, OUTPUT);
  pinMode(Trig2, OUTPUT);
  pinMode(Trig3, OUTPUT);
  pinMode(Eco1, INPUT);
  pinMode(Eco2, INPUT);
  pinMode(Eco3, INPUT);
}
```

Figura 3713 - Bloc Setup

El bloc *loop* cal explicar-lo per passes, ja que conté més codi que el *setup* i així s'entendrà millor que realitza. Un cop encès el sensor de temperatura DHT22 cal esperar uns 2 segons per a poder usar-lo, per això es comença amb un *delay*(retard) de 2 segons per a que la lectura d'aquest sigui correcte. Es segueix amb el càlcul de la velocitat del so usant la Eq.2 esmentada anteriorment, cal destacar l'ús de la funció *dht.readTemperature()* de la llibreria DHT que ens calcula la temperatura de manera immediata i ens la retorna en format *float*.

```
void loop() {  
    delay(2000);  
  
    SpeedOfSound = 331,5 + 0,6 * dht.readTemperature();
```

Figura 38 - Càlcul de la velocitat del so

A continuació es deixarà un *delay* de 1 segon per a començar a calcular les distàncies. El càlcul de les distàncies mitjançant els sensors es realitza de la següent manera:

1. Activar pin Trig amb un puls d'un voltatge superior a 2,4V i amplada mínima de $1 \cdot 10^{-5}$ s.
2. Emissió de 8 rafegues d'ultrasons a 40kHz.
3. Activació pin Eco.
4. Càlcul del temps d'espera de recepció de les rafegues.
5. Càlcul de la distància amb la Eq.1.

Entre els passos d'enviar el puls al pin Trig o l'activació del Eco es deixen una sèrie de *delays* per assegurar-nos que el pas anterior la placa Arduino la realitza amb seguretat i per tant el següent pas es pot executar.

```
delay(1000);  
  
digitalWrite(Trig1, LOW);  
delayMicroseconds(2000);  
digitalWrite(Trig1, HIGH);  
delayMicroseconds(15);  
digitalWrite(Trig1, LOW);  
delayMicroseconds(10);  
pingTime1 = pulseIn(Eco1, HIGH);  
pingTime1 = pingTime1/1000000;  
Distance1 = SpeedOfSound * pingTime1 * 100 / 2;
```

Figura 149 - Càlcul de la distància al sensor 1

El càlcul del temps que triguen les rafegues en tornar la treiem de la funcio *pulseIn()* que ens retorna en microsegons aquest temps, per això es necessari passar-ho a segons. També destacar que per al calcul de la distancia es multiplica per 100 per a obtenir el resultat en centimetres tot i que es totalment opcional. El codi que ve a continuació consisteix en repetir aquest proces per als dos altres sensors restants:

```

delay(1000);

digitalWrite(Trig2,LOW);
delayMicroseconds(2000);
digitalWrite(Trig2,HIGH);
delayMicroseconds(15);
digitalWrite(Trig2,LOW);
delayMicroseconds(10);
pingTime2 = pulseIn(Eco2,HIGH);
pingTime2 = pingTime2/1000000;
Distance2 = SpeedOfSound * pingTime2 * 100 / 2;

delay(1000);

digitalWrite(Trig3,LOW);
delayMicroseconds(2000);
digitalWrite(Trig3,HIGH);
delayMicroseconds(15);
digitalWrite(Trig3,LOW);
delayMicroseconds(10);
pingTime3 = pulseIn(Eco3,HIGH);
pingTime3 = pingTime3/1000000;
Distance3 = SpeedOfSound * pingTime3 * 100 / 2;

```

Figura 4015 - Càlcul de las distancies dels sensors 2 i 3

Finalment es mostraran aquests valors pel Monitor Serial i es deixarà que aquests s'estabilitzin i s'agafaran com a les variables *Dmax1*, *Dmax2* i *Dmax3* per al codi principal. Arrodonirem el valor al enter següent per a assegurar-nos que cap càlcul posterior ens porti a un volum ocupat inferior al 0%. És a dir, si el valor al qual el sensor 1 s'estabilitza es a 36,4 centímetres, s'agafarà 37 com a *Dmax1*.

```

Serial.println(String(Distance1) + " - " + String(Distance2) + " - " + String(Distance3));

```

Figura 41 - Mostrar valors calculats pel Monitor Serial

Es mostren els valors mitjançant el Monitor Serial de la següent manera:

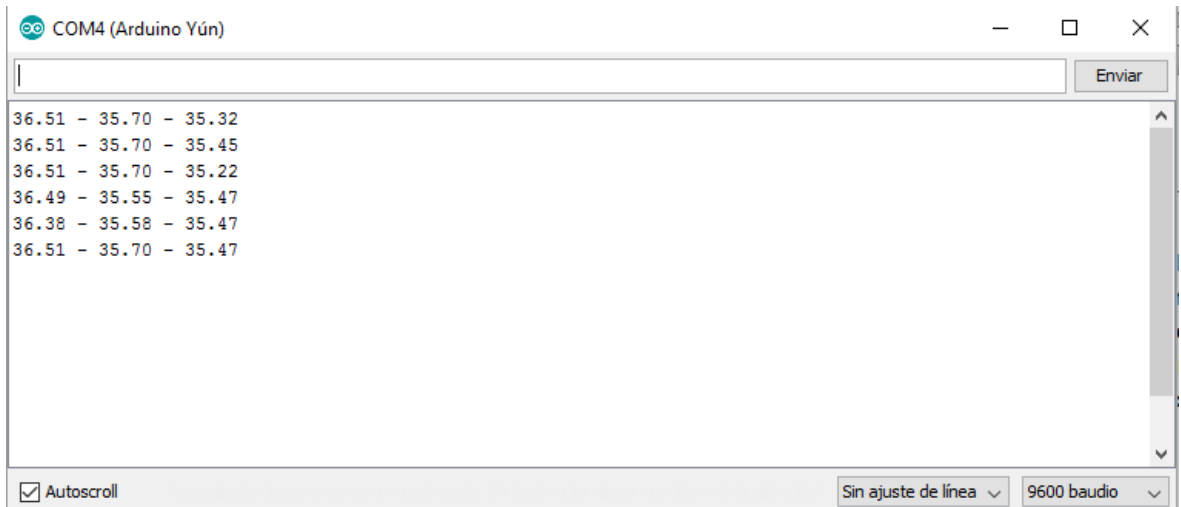


Figura 4216 - Lectura sensors mostrada pel Monitor Serial

3.4.2. Estructura del codi principal

En l'estructura del codi principal hi ha certs parts que és similar al del subcodi, ja que s'ha de definir el sensor de temperatura i els de distancia, el codi dels càlculs de les distancies es repeteix... Per tant les parts ja esmentades no s'explicaran en detall, aquestes es troben el punt anterior 7.4.1.

Com en el codi explicat al punt anterior, es comença cridant les llibreries que s'usaran durant tot el codi:

```
#include <Bridge.h>
#include <YunClient.h>
#include "DHT.h"
```

Figura 4317 - Cridar llibreries

A continuació es realitza la definició del sensor de temperatura DHT22 que és exactament igual al del apartat anterior.

```
#define DHTPIN 2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
```

Figura 4418 - Definició sensor DHT22

Ara es definirà la direcció IP del servidor on es volen enviar les lectures del sensors i també es definirà el client Yún per a usar-lo posteriorment.

```
IPAddress server(62,57,212,160);  
  
YunClient client;
```

Figura 4519 - Definició IP i client Yún

Com en el codi anterior, s'han de definir els pins del sensors de distància com les variables que s'usaran durant el codi. Destacar que apareixen dues variables noves: *Volum* que correspon al tant percent de volum ocupat a la paperera i el text *command* que correspon al text que s'envia al servidor com a POST.

```
const int Trig1 = 11;  
const int Eco1 = 12;  
const int Trig2 = 6;  
const int Eco2 = 7;  
const int Trig3 = 3;  
const int Eco3 = 4;  
  
float pingTime1;  
float pingTime2;  
float pingTime3;  
float Distance1;  
float Distance2;  
float Distance3;  
float Volum;  
float speedOfSound;  
String command;
```

Figura 4620 - Definició variables

A continuació es defineixen les constats que es bolcaran del codi anterior.

```
const int Dmax1 = 37;  
const int Dmax2 = 36;  
const int Dmax3 = 36;
```

Figura 4721 - Definició de les distàncies màximes

Un cop definit tot s'entra al bloc *setup* on l'únic diferent a destacar és que també s'inicialitza el *Bridge* que es la llibreria del Yún que permet fer la connexió entre la part del microcontrolador amb el sistema operatiu Linux. Gracies a aquesta connexió podem transferir les dades dels sensors a la part del Linux i enviar-les al servidor.

```
void setup() {
    Serial.begin(9600);
    Bridge.begin();
    dht.begin();

    pinMode(Trig1, OUTPUT);
    pinMode(Trig2, OUTPUT);
    pinMode(Trig3, OUTPUT);
    pinMode(Eco1, INPUT);
    pinMode(Eco2, INPUT);
    pinMode(Eco3, INPUT);
}
```

Figura 4822 - Bloc Setup

Després del bloc *setup* ve el *loop* on es mesurarà la temperatura, la humitat, la velocitat del so i les distàncies dels 3 sensors. També es realitzarà el POST al servidor amb les dades obtingudes. A continuació es mostra la lectura de la temperatura, humitat i el càlcul de la velocitat del so amb la Eq.2.

```
delay(2000);
String temp = String(dht.readTemperature());
String hum = String(dht.readHumidity());

speedOfSound = 331,5 + 0,6 * dht.readTemperature();
```

Figura 239 - Càlcul temperatura, humitat i velocitat del so

Un cop fet això es realitzen les mesures de les distàncies exactament igual que s'han fet en l'anterior codi, només es mostrarà el càlcul d'una dels tres.

```
delay(600);

delayMicroseconds(10);
digitalWrite(Trig1, LOW);
delayMicroseconds(2000);
digitalWrite(Trig1, HIGH);
delayMicroseconds(15);
digitalWrite(Trig1, LOW);
delayMicroseconds(10);
pingTime1 = pulseIn(Eco1, HIGH);
pingTime1 = pingTime1/1000000;
Distance1 = speedOfSound * pingTime1 * 100 / 2;
```

Figura 50 – Càlcul de la distància

S'ha decidit posar els sensors en la part superior de la paperera tal i com s'explica en l'apartat 3.3. Per tant el càlcul del volum ocupat es realitzarà de la següent manera:

$$V_{Rest} = \frac{D_{max} - D_{mes}}{D_{max}} \times 100 \quad (Eq. 3)$$

On:

- V_{Rest} correspon a el % de volum ocupat de la paperera.
- D_{max} correspon a la distancia màxima que enregistra el sensor, extremitat del subcodi.
- D_{mes} correspon a la distancia calculada pel sensor en la lectura.

Per tant el càlcul del volum ocupat amb les tres lectures es de:

$$V_{RestT} = \frac{V_{Rest1} + V_{Rest2} + V_{Rest3}}{3} \quad (Eq. 4)$$

On:

- V_{RestT} correspon a la mitjana.
- V_{Restx} corresponen als volums calculats amb la Eq. 3.

Per tant el codi resulta de la següent manera:

```
Volum = 100 * ((Dmax1 - Distance1)/Dmax1 + (Dmax2 - Distance2)/Dmax2 + (Dmax3 - Distance3)/Dmax3)/3;
```

Figura 51 - Càlcul del volum ocupat

Tenint tots els valors que es volen enviar, es formula el string *command* que serà el text que s'enviarà amb el POST i que tractarà el servidor. Aquest string es forma de la següent manera:

Variable_1=Valor_1&Variable_2=Valor_2&Variable_3=Valor_3&...

Aquesta es la estructura que el POST admet per a enviar variables. Entenent el caràcter & com a separador entre variables.

```
command = "Temperatura="+temp+"&Humitat="+hum+"&Distancia1="+String(Distance1)+"&Distancia2="+String(Distance2)+"&Distancia3="+String(Distance3)+"&Volum="+String(round(Volum));
```

Figura 5224 - String command

L'última part del codi és la comanda POST. Primer es mostrarà la imatge del codi i després s'explicarà que fa cada punt.

```

if (client.connect(server, 80)) {
  delay(2500);
  client.println("POST /add.php HTTP/1.1");
  client.println("Host: 62.57.212.160");
  client.print("Content-length:");
  client.println(command.length());
  client.println("Content-Type: application/x-www-form-urlencoded;");
  client.println();
  client.println(command);
}else{
  delay(1000);
}
if(client.connected()){
  client.stop();
}
delay(60000);

```

Figura 5325 - POST al servidor

Només es realitzarà el POST si el client Yún s'ha connectat sense problemes al servidor. La primera línia "POST /add.php HTTP/1.1" significa que es vol enviar un POST al mòdul *add.php* del servidor amb el protocol de transferència HTTP/1.1. A continuació s'indica la IP del servidor al qual s'està enviant la informació.

Seguidament s'indica la llargada y el tipus contingut enviat per a que el servidor sàpiga com tractar la informació que s'està transferint. Finalment s'envia el text amb les variables mesurades i els seus valors.

Si la connexió amb el servidor no s'efectua, el POST no s'envia, s'espera un segon i el bloc *loop* es torna a executar. Finalment es para el client Yún per a tancar la connexió si aquesta s'ha efectuat de manera correcte.

3.5. Flux d'informació

Per resumir el flux d'informació que hi ha a la paperera i que es segueix al codi s'ha volgut plasmar aquest en un diagrama de flux senzill com a resum de lo explicat anteriorment.

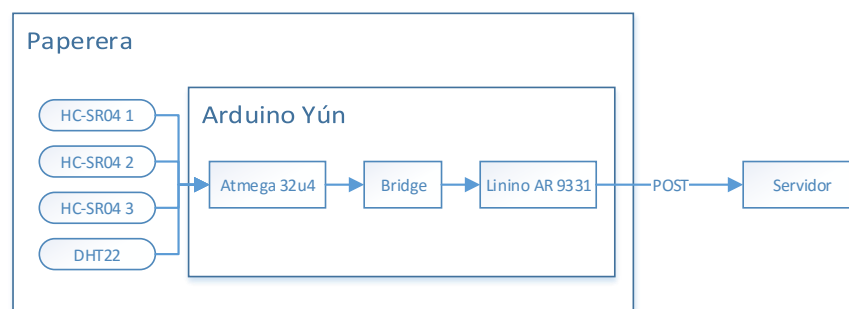


Figura 5426 - Diagrama de flux d'informació

4. Arquitectura del Servidor

En aquest apartat es vol explicar d'una manera detallada l'arquitectura del servidor: la posada a punt d'aquest, de quins mòduls està compost, que realitza cada mòdul, i els fluxos d'informació interns entre aquests.

4.1. Posada a punt

S'usarà un servidor HTTP molt conegut anomenat Apache i aquest se li afegiran una sèrie de components per a convertir-lo en el conegut conjunt de solucions web LAMP (Linux, Apache, MySQL, PHP). Per a aquest treball s'instal·larà el servidor en la versió xubuntu 14.04 LTS de 32 bits [21].

1. Primer de tot obrir la consola de Linux.
2. Fer una actualització del programa *apt-get* que s'usa en Linux per a la instal·lació de programes.

```
$ sudo apt-get update && sudo apt-get upgrade
```

3. Instal·lació del servidor Apache

```
$ sudo apt-get install apache2
```

4. Per a comprovar que la instal·lació ha estat un èxit accedirem a una de les següents webs i ens ha de sortir una pantalla com la de la Figura 55.

<http://127.0.0.1> o <http://localhost>

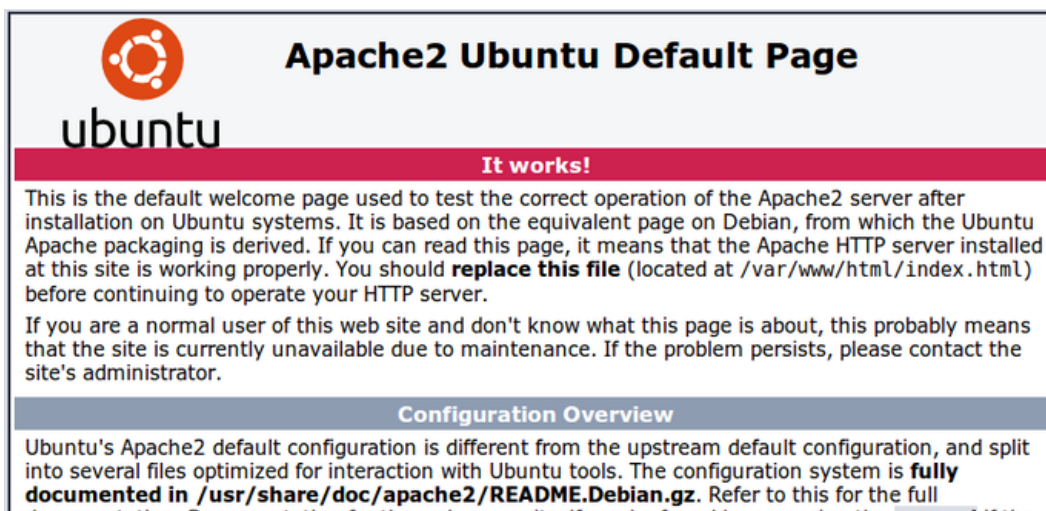


Figura 5527 - Comprovació instal·lació del Apache

5. Instal·lació de la base de dades MySQL. Durant el procés s'haurà d'introduir una contrasenya per al usuari aquesta és molt important perquè s'usarà per accedir a les bases de dades posteriorment.

```
$ sudo apt-get install mysql-server php5-mysql
```

6. Un cop instal·lat el MySQL s'han de realitzar una sèrie de configuracions. Primer de tot s'ha de crear la estructura on es guardarà la informació amb la següent comanda.

```
$ sudo mysql_install_db
```

7. A continuació s'han de eliminar algunes característiques de seguretat marcades per defecte. Això es realitza amb la comanda següent que executa un script on primer de tot el pregunta si vols canviar la contrasenya cosa que direm que no perquè ja s'ha fixat una al principi, les altres preguntes si es tecleja ENTER ja agafen els valors que es desitgen per al treball.

```
$ sudo mysql_secure_installation
```

8. Per comprovar que la instal·lació del MySQL ha estat un èxit s'escriurà la comanda següent i escrivint la contrasenya d'accés que s'ha fixat en el pas 5 quan la consola te la demani.

```
$ mysql -u root -p
```

9. Si tot funciona correcte et trobaràs al interior del MySQL això significa que la instal·lació ha estat un èxit. Per a sortir s'ha de teclejar *exit* i prémer la tecla ENTER.
10. Instal·lació del llenguatge de programació PHP. Aquest servirà per a poder executar scripts, connectar-se a la base de dades del MySQL i processar tot el contingut del servidor.

```
$ sudo apt-get install php5 libapache2-mod-php5 php5-mcrypt
```

11. Amb aquest pas la instal·lació del PHP ja estarà efectuada, però realitzarem un canvi per a que per defecte el servidor obri els mòduls anomenats *index.php* abans que els *index.html* com ve definit. Per a realitzar això cal accedir a un arxiu de configuració.

```
$ sudo nano /etc/apache2/mods-enabled/dir.conf
```

12. Això obrirà l'arxiu a la consola per editar-lo. S'haurà d'escriure *index.php* abans que *index.html*. Ha de quedar de la següent manera:

```
DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
```

13. Per a que els canvis es realitzin cal reiniciar el servidor.

```
$ sudo service apache2 restart
```

14. Per a comprovar que la instal·lació del llenguatge PHP ha estat un èxit es realitzarà el següent:

```
$ sudo nano /var/www/html/info.php
```


15. Escriure el següent codi:

```
<?php phpinfo(); ?>
```

16. Accedir a la web següent i comprovar que es mostra alguna cosa molt similar a la Figura 56.

<http://localhost/info.php>

PHP Version 5.5.9-1ubuntu4



System	Linux blah 3.13.0-24-generic #46-Ubuntu SMP Thu Apr 10 19:11:08 UTC 2014 x86_64
Build Date	Apr 9 2014 17:08:00
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d

Figura 5628 - Comprovació de la instal·lació del PHP

17. Instal·lació del conjunt LAMP completa.

18. Un complement addicional que no pertany al conjunt LAMP però que es necessari per a fer una gestió eficient del MySQL i les bases de dades del servidor és el programa *phpMyAdmin* aquest s'instal·la amb la següent comanda:

```
$ sudo apt-get install phpmyadmin
```

Al finalitzar la creació i configuració del servidor, s'ha tingut d'accedir al router de la xarxa on s'estava treballant i redirigir els ports del router per a que la gent pugui accedir al servidor des de fora. Aquest pas és totalment diferent depenen del router amb el qual es treballa per tant no s'ha volgut explicar el procediment sinó que cadascú busqui la manera de redirigir els ports a internet depenent del seu router.

S'ha cregut convenient explicar el procés de creació de la BBDD i la taula on s'aniran afegint el valors en el següent apartat que parla sobre la estructura de la BBDD.

4.2. Estructura BBDD

En aquest apartat es parlarà de la BBDD del servidor, com es gestiona, quins paràmetres conte i el perquè d'aquests. Per a realitzar la gestió de la BBDD s'usarà el programa instal·lat prèviament *phpMyAdmin*.

4.2.1. Taula de la BBDD

A continuació es mostraran els motius de perquè s'han escollit aquestes columnes en la creació de la taula en la BBDD.

- **ID**: Paràmetre bàsic i incremental que s'encarrega de donar la informació de quantes mesures s'han realitzat. Permet ordenar de manera senzilla les mesures per ordre d'entrada i també serveix per assegurar-se que mai hi hauran dues entrades iguals.
- **Dia**: Paràmetre que indica el dia en que s'ha realitzat la mesura, molt important ja que mitjançant aquest podem filtrar les mesures per dies i per tant fer les gràfiques de temperatura, humitat i volum ocupat. És un requisit indispensable per a la finalitat en que s'ha dissenyat la web.
- **Hora**: Paràmetre que indica la hora en que s'ha realitzat la mesura, important per a realitzar les gràfiques ja que correspon als valors del eix X. Sense aquests no podríem més tard fer una analítica de quan les papereres s'omplen més, hores on la temperatura és més elevada...

- **Temperatura**: Paràmetre que indica la temperatura mesurada, important per a realitzar la gràfica de temperatura ja que correspon als valors del eix Y. També serveix com a indicador de que la paperera pot estar cremant-se.
- **Humitat**: Paràmetre que indica la humitat mesurada, important per a realitzar la gràfica de humitat ja que correspon als valors del eix Y.
- **Distancia1**: Paràmetre que indica la distancia mesurada per un dels sensors HC-SR04, molt important per a detectar futures averies d'aquest sensor.
- **Distancia2**: Paràmetre que indica la distancia mesurada per un dels sensors HC-SR04, molt important per a detectar futures averies d'aquest sensor.
- **Distancia3**: Paràmetre que indica la distancia mesurada per un dels sensors HC-SR04, molt important per a detectar futures averies d'aquest sensor.
- **Volum**: Paràmetre que indica el volum ocupat de la paperera, important per a realitzar la gràfica de volum ocupat ja que correspon als valors del eix Y.

Bàsicament tots els paràmetres escollits s'han pensat per a que posteriorment es pugui fer un anàlisi de com varia l'estat de la paperera en funció de les hores del dia, de dies en concret... Tot i que els valors que corresponen a la distancia són valors que no són accessibles per al visualitzador, aquests serveixen per a detectar possibles averies i per tant identificar amb facilitat quin dels sensors no funciona correctament.

4.2.2. Creació de la BBDD i taula

Es parlarà de com usar el programa *phpMyAdmin* per a crear la BBDD que contindrà les dades de les nostres mesures que realitzarà la paperera.

1. Accedir al *phpMyAdmin* mitjançant la web següent:

<http://localhost/phpmyadmin/>

2. Introduir la contrasenya fixada prèviament en la instal·lació del MySQL.



Figura 57 - Inicia sessió al MySQL

3. Un cop iniciat sessió ens dirigirem a “Base de datos”.

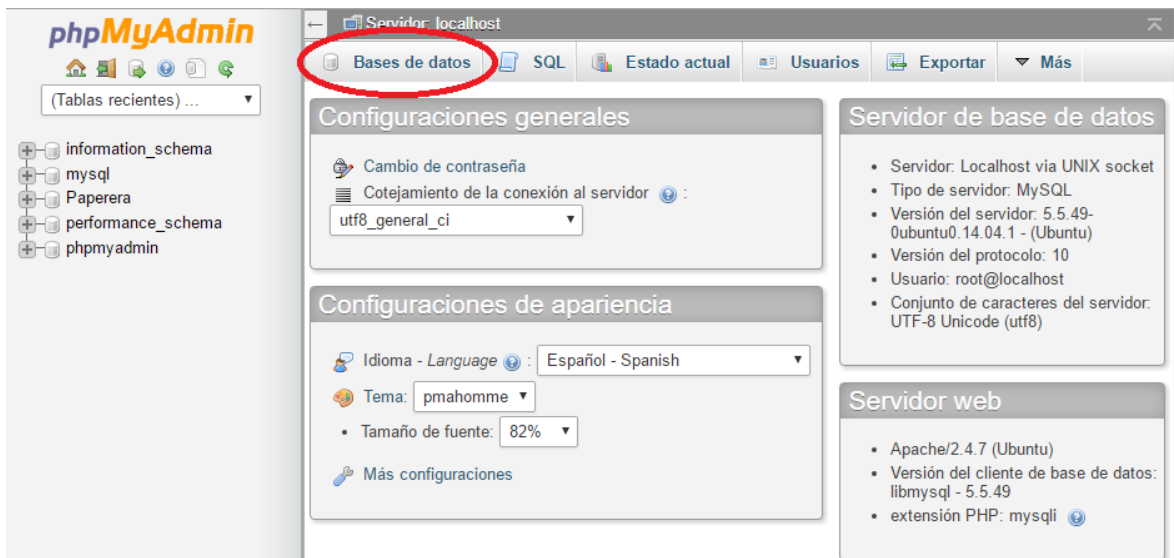


Figura 5829 - Pagina inicial phpMyAdmin

4. Crearem una BBDD, s'ha decidit anomenar-la “Paperera”.

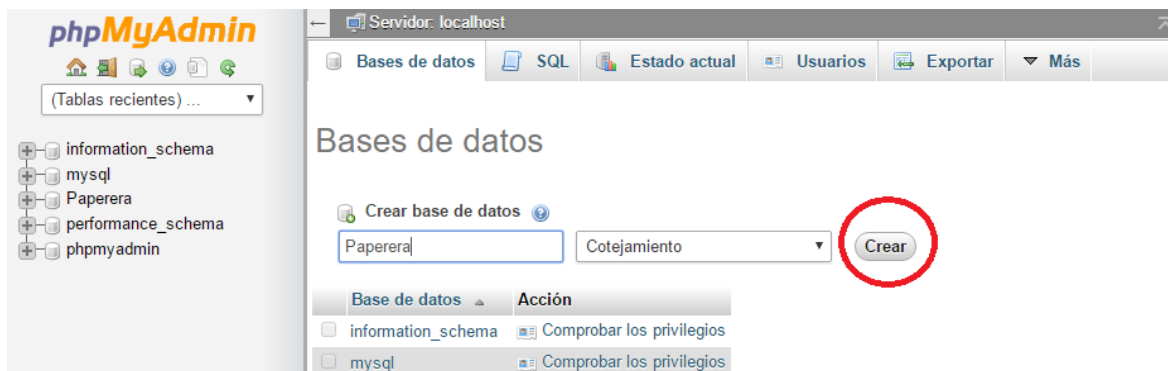


Figura 309 - Creació BBDD

5. Accedirem a la BBDD que s'ha creat amb el menú de la esquerra i crearem una taula al seu interior de 9 columnes. S'ha decidit anomenar-la "Sensors".

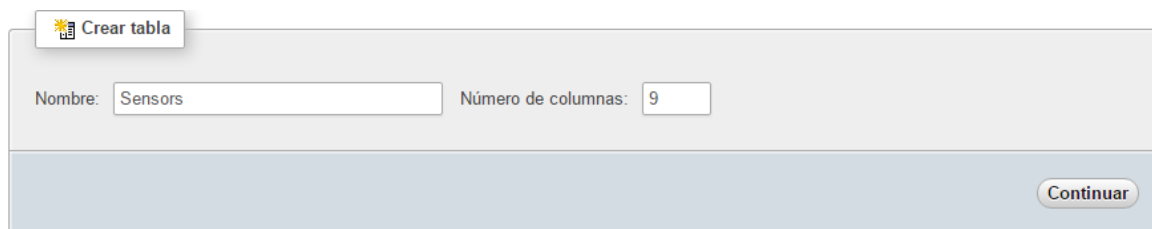


Figura 6031 - Creació taula

6. A continuació omplirem la taula que apareix amb els valors que es mostren en la Figura següent. Destacar que la primera columna *id* també cal marcar la casella *A_* de la taula, ja que volem que es comporti com a un comptador incremental que ens indiqui l'ordre d'entrada de les mesures.


Nombre	Tipo 
<input type="text" value="id"/>	<input type="text" value="INT"/>
<input type="text" value="Dia"/>	<input type="text" value="DATE"/>
<input type="text" value="Hora"/>	<input type="text" value="TIME"/>
<input type="text" value="Temperatura"/>	<input type="text" value="TEXT"/>
<input type="text" value="Humitat"/>	<input type="text" value="TEXT"/>
<input type="text" value="Distancia1"/>	<input type="text" value="TEXT"/>
<input type="text" value="Distancia2"/>	<input type="text" value="TEXT"/>
<input type="text" value="Distancia3"/>	<input type="text" value="TEXT"/>
<input type="text" value="Volum"/>	<input type="text" value="TEXT"/>

Figura 61 - Propietats de les columnes

7. Prémer el botó Guardar i es crearà la taula per a introduir els valors procedents de la paperera.

Cal destacar que en els noms fixats tant el de la BBDD (Paperera) com el de la taula (Sensors) cal tenir-los presents per que alhora de realitzar la connexió en els mòduls PHP del servidor s'han d'usar aquests mateixos noms.

4.3. Fluxos d'informació

Abans d'entrar a parlar sobre que fa cada mòdul dins del servidor, es mostrarà un mapa dels fluxos d'informació entre aquests per a quan s'expliqui la funcionalitat de cadascun sigui més fàcil d'entendre per al lector.

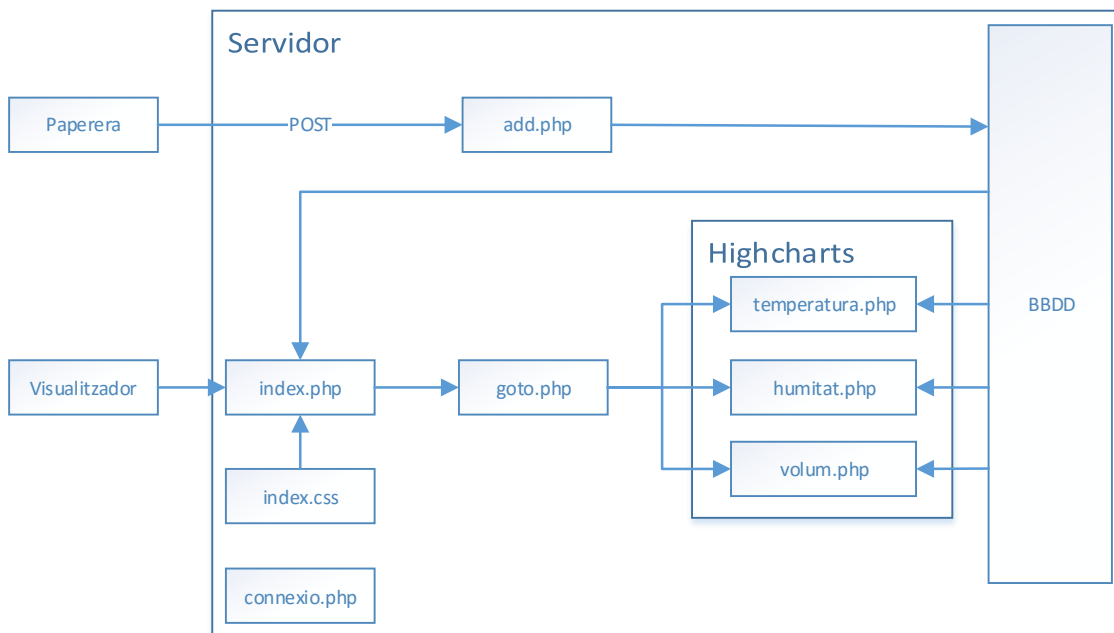


Figura 62 - Mapa de fluxos d'informació

Destacar que el usuari visualitzador accedirà al servidor pel mòdul *index.php*. D'altra banda l'usuari administrador serà l'encarregat de gestionar tots aquests mòduls i la base de dades que es troben dins la caixa Servidor. Per a mostrar les dades s'ha usat la llibreria Highcharts, la qual et permet mostrar les dades en format gràfica.

4.4. Mòduls

No es mostrarà el codi en detall si no que s'explicarà la funcionalitat de cadascun. Tot i així el codi estarà disponible als Annexos del treball.

4.4.1. Add.php

S'encarrega d'introduir a la BBDD les mesures que s'envien des de la paperera. Això ho aconseguix de la següent manera:

1. Connectant-se a la BBDD.
2. Agafant els valors de les variables enviades mitjançant el POST.
3. Introduir-les a taula creada dins la BBDD.

4.4.2. Connexio.php

S'encarrega de connectar-se a la BBDD. Aquest es crida per a altres mòduls al inici del codi si han de connectar-se a la BBDD. Els mòduls que l'usen són els següents:

- Index.php
- Temperatura.php
- Humitat.php
- Volum.php

4.4.3. Index.php

Mòdul que correspondrà a la pagina web inicial que el visualitzador usarà per veure l'estat de la paperera i accedir a les dades dels dies, aquesta tindrà el format següent:

Titol	
Estat de la paperera	Valors dels sensors
Última mesura	
Accedir a gràfiques	

Figura 6332 - Format index.php

En l'estat de la paperera es mostraran dos cercles:

- Un per la temperatura, que pot presentar el color verd o vermell, que indicarà d'una manera visual si la paperera es troba en perill d'incendi.
- Un per el volum ocupat, que pot presentar el color verd, taronja o vermell, que indicarà d'una manera visual si la paperera es troba ocupada per sota del 30%, entre el 30-80% o més del 80% respectivament.

Per altra banda, en la zona valors dels sensors es mostrarà l'última lectura d'aquests per si es vol conèixer en més detall l'estat. En la zona última mesura es mostrarà la data i la hora en que s'ha registrat les últimes dades. Finalment en la zona accedir a gràfiques es donarà la opció amb dos desplegable d'escollir veure les dades en format gràfica de tots els dies en que la paperera ha registrat dades i escollir si es vol visualitzar la temperatura, humitat o volum ocupat.

Cal destacar que per l'ús d'aquest mòdul es necessari usar JQuery per a poder fer efectiu l'script que permet canviar els cercles de color en funció d'una variable.

4.4.4. Goto.php

Mòdul que s'encarrega de redirigir-te a una gràfica o una altre (temperatura.php, humitat.php, volum.php) en funció del valor que vols visualitzar en una gràfica després de prémer el botó Mostrar en el mòdul *index.php*.

Bàsicament quan es prem el boto del mòdul *index.php*, s'envien tant el dia com la gràfica que vols visualitzar amb el mètode POST a aquest mòdul. Aleshores mitjançant una estructura de if, elseif i else es redirigeix cap al mòdul desitjat (temperatura.php, humitat.php, volum.php) amb la variable dia que servirà per a filtrar els valors de la BBDD a mostrar a les gràfiques.

En la manera de redirigir-te als mòduls que mostren les gràfiques és on s'envia la variable dia. Aquesta es enviada de la següent manera amb codi php:

```
header("Location: temperatura.php?Dia=$Dia")
```

D'aquesta manera el mòdul *temperatura.php* mitjançant un GET pot obtenir el valor de la variable Dia i filtrar els valors de la BBDD.

4.4.5. Temperatura.php

Mòdul que s'encarrega de mostrar-te en una gràfica els valors de temperatura enregistrats un cert dia per la paperera. Això ho fa bàsicament connectant-se a la BBDD usant el mòdul *connexio.php*, obtenint el valor de la variable Dia mitjançant un GET i gracies a la llibreria Highcharts que ens proporciona la estructura de les gràfiques podem mostrar els valors mesurats. També té un boto per a tornar a la pàgina d'inici.

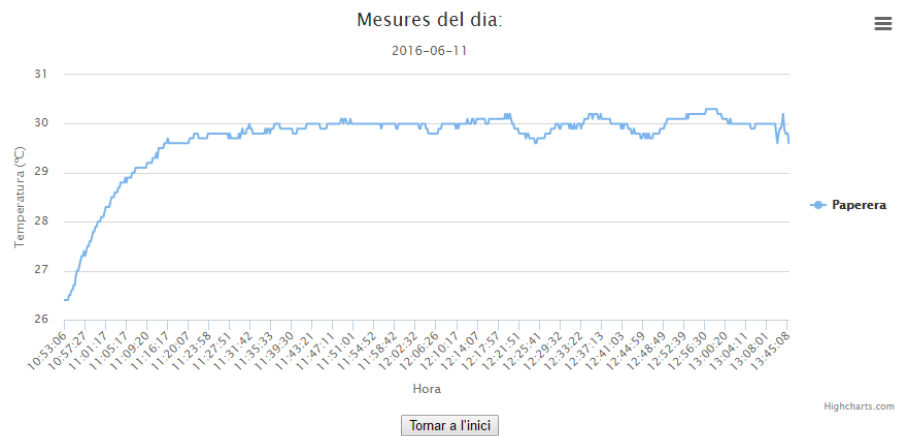


Figura 6433 - Exemple de gràfica de temperatura

4.4.6. Humitat.php

Mòdul que s'encarrega de mostrar-te en una gràfica els valors de la humitat enregistrats un cert dia per la paperera. Això ho fa bàsicament connectant-se a la BBDD usant el mòdul *connexio.php*, obtenint el valor de la variable Dia mitjançant un GET i gracies a la llibreria Highcharts que ens proporciona la estructura de les gràfiques podem mostrar els valors mesurats. També té un boto per a tornar a la pàgina d'inici.



Figura 6534 - Exemple de gràfica de humitat

4.4.7. Volum.php

Mòdul que s'encarrega de mostrar-te en una gràfica els valors del volum ocupat enregistrats un cert dia per la paperera. Això ho fa bàsicament connectant-se a la BBDD usant el mòdul *connexio.php*, obtenint el valor de la variable Dia mitjançant un GET i gracies a la llibreria Highcharts que ens proporciona la estructura de les gràfiques podem mostrar els valors mesurats. També té un boto per a tornar a la pàgina d'inici.



Figura 6635 - Exemple de gràfica de volum ocupat

Destacar que la llibreria Highcharts incorpora a les seves gràfiques un boto que permet descarregar-se la gràfica en format imatge PNG, JPG o SVG i en format document PDF. També permet l'enviament directe del gràfic a imprimir.

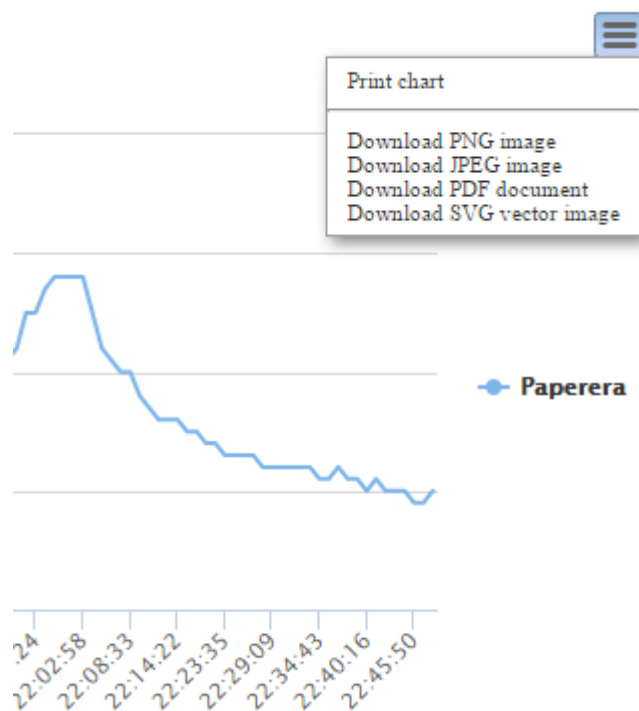


Figura 6736 - Funcionalitats Highcharts

5. Planificació i costos del treball

5.1. Planificació

La duració total del treball és de aproximadament 5 mesos i correspon a una dedicació total de 300 hores. La data d'inici del treball és el 13 de Febrer del 2016 i finalitza amb l'entrega programada del 20 al 23 de Juny del mateix any. La planificació del treball es divideix en 3 fases principals. Aquestes són la Fase prèvia, la Fase de disseny i la Fase final.

5.1.1. Fase prèvia

La primera fase del treball consisteix en plantejar el treball, definir els objectius d'aquest i l'abast que tindrà. Per això, es necessari la cerca d'informació dels principals components que s'haurien d'usar durant el treball, així com diferents opcions per a poder triar la més adequada per al treball.

5.1.2. Fase de disseny

En aquesta fase s'estudien les possibles solucions per a la confecció del prototipus i es tria la millor opció. Amb aquesta es comença el disseny de tot el prototipus, tant la part de la paperera com la del servidor. També es realitzen les proves per a comprovar que aquest funciona correctament.

5.1.3. Fase final

En aquesta fase un cop confeccionat tot el prototipus es procedeix a fer un estudi del cost del treball així com l'impacte ambiental que aquest ha suposat. També en aquesta fase s'inclou tota la estructuració i redacció de la memòria així com la preparació de la presentació del treball.

A continuació es mostra una taula amb les tasques realitzades, la seva duració i les dades de inici i finalització d'aquestes:

Id	Nom de la tasca	Inici	Final	Duració
0	FASE PREVIA			
1	Cerca d'informació	13/02/2016	01/03/2016	17 dies
2	Plantejament treball	13/02/2016	17/02/2016	4 dies
3	Definició dels objectius dels treball	17/02/2016	24/02/2016	7 dies
4	Definició del abast del abast	24/02/2016	01/03/2016	6 dies
5	FASE DE DISSENY			
6	Estudi opcions	01/03/2016	12/03/2016	11 dies
7	Tria de la millor opció	12/03/2016	20/03/2016	8 dies
8	Disseny arquitectura global	20/03/2016	30/03/2016	10 dies
9	Disseny arquitectura paperera	30/03/2016	26/04/2016	27 dies
10	Disseny arquitectura servidor	21/04/2016	25/05/2016	34 dies
11	Proves	30/03/2016	25/05/2016	56 dies
12	FASE FINAL			
13	Redacció memòria	03/04/2016	21/06/2016	79 dies
14	Estudi del cost del treball	25/05/2016	26/05/2016	1 dia
15	Estudi del impacte ambiental del treball	20/06/2016	21/06/2016	1 dia
16	Preparació presentació	23/06/2016	01/07/2016	8 dies

Taula 7 - Tasques

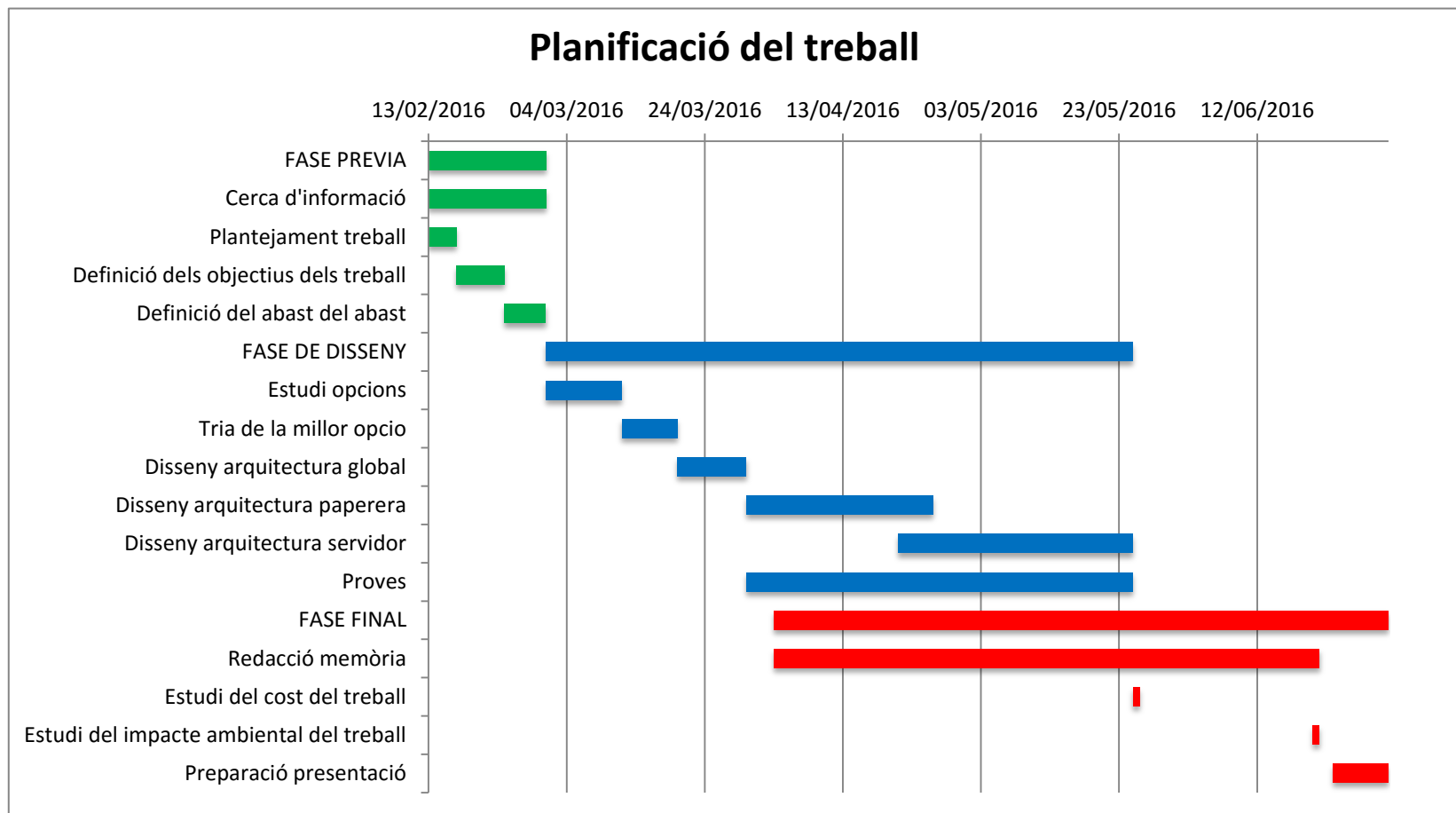


Figura 68 - Diagrama de Gantt de la planificació del treball

5.2. Costos associats al treball

En la realització del estudi dels costos del treball s'han considerat els costos associats a la investigació i desenvolupament del treball i el cost estimat del prototip incloent el de posada a punt. Aquest cost total s'ha desglossat en dues parts, la part de investigació i desenvolupament del treball, el del prototip i la seva posada a punt.

Item	Quantitat	Preu unitari (€/u)	Preu (€)
Capsa/paperera	1	3,50	3,50
Arduino Yún	1	75,02	75,02
Protoboard	1	6,04	6,04
Sensor HC-SR04	3	6,96	20,88
Sensor DHT22	1	14,52	14,52
Resistència 10kΩ	1	0,02	0,02
5m Cable	2	0,95	1,90
Connector CTO	20	0,06	1,20
Terminal connector	20	0,06	1,20
Servidor	1	200,00	200,00
		TOTAL	324,28 €

Taula 8- Costos prototip

Concepte	Quantitat	Preu	Import
Investigació i desenvolupament:			
• Enginyer Industrial	175h	40 €/h	7000 €
• Enginyer Tècnic	75h	30 €/h	2250 €
Prototip	1	324,28 €/u	324,28 €
Posada a punt del prototip	50h	30 €/h	1500 €
		TOTAL	11074,28 €

Taula 9 - Costos totals

6. Impacte ambiental

En aquest apartat es vol realitzar el càlcul del impacte ambiental que ha suposat l'elaboració d'aquest treball. Aquest s'ha realitzat fent un càlcul del consum elèctric total que aquest ha suposat, ja que l'elaboració de la major part d'aquest s'ha efectuat a una habitació amb llum, un ordinador de sobretaula, un portàtil que serveix com a servidor i també el consum de la placa.

S'ha desglossat el consum de cadascuna de les parts mencionades i s'ha fet el càlcul del consum elèctric màxim per hora per a poder calcular els grams de CO₂ emesos durant la realització del projecte:

Element	Potència
Llum	30W
Sobretaula	250 W
Monitor	30 W
Servidor	40 W
Arduino Yún	1 W
TOTAL	351 W

Taula 10 - Consum elèctric per component

Per tant existeix un consum màxim total de 351 W que és el mateix que 0,351 kW. Tenim un consum màxim per hora de 0,351 kWh, donat que pràcticament en la totalitat de la realització del treball s'han usat alhora tots els components. Utilitzant una dada de la Generalitat de Catalunya [22] sabem que el mix de producció bruta de la creació d'un kWh és de aproximadament 302 g CO₂/kWh. Per tant:

$$0,351 \text{ kWh} \cdot 302 \frac{\text{g CO}_2}{\text{kWh}} = 106 \text{ g CO}_2$$

Aquest ha estat el l'emissió de CO₂ per hora durant la confecció d'aquest treball.

Conclusions

Un cop fet el disseny i muntat tot el prototipus de la paperera i el servidor, es pot concloure que els objectius detallats en l'apartat 1 s'han complert. S'ha aconseguit muntar una paperera capaç de recollir les dades del seu estat mitjançant sensors i que aquesta les enviï a un servidor, tot això amb una placa microcontroladora Arduino Yún. També s'ha aconseguit dissenyar i configurar un servidor amb una base de dades on es dipositen les mesures dels sensors i amb una part web molt senzilla que permeti la visualització d'aquestes. A més a més pensat amb una arquitectura que permeti la incorporació d'altres papereres.

Cal destacar que al inici del treball només es posseïa d'un background bàsic de programació amb Python i per tant cap experiència amb la majoria de les tecnologies emprades:

- Arduino y la seva programació
- Servidors
- PHP
- HTML
- CSS
- MySQL
- JavaScript
- J-Query

De totes aquestes noves tecnologies, gracies a aquest treball, s'ha adquirit un cert coneixement molt enriquidor i amb un possible ús en el futur de la meva carrera professional. Aquest treball és un inici de molts següents passos a realitzar per a acabar-lo de millorar i poder-lo arribar a usar a gran escala. A continuació es mostren els punts a tractar a partir del desenvolupament d'aquest treball:

- Un dels objectius que es volia aconseguir amb aquest treball era deixar una arquitectura que permetés la incorporació de més papereres en aquesta i així poder arribar a controlar tota la infraestructura de papereres de per exemple un aeroport, centre comercial o fins i tot una ciutat. Per tant un dels punts de continuació d'aquest treball és incorporar més papereres i provar la infraestructura.
- En aquest treball la part web del servidor que permet la visualització de les dades ha estat dissenyada i pensada per a una paperera només. Això vol dir que amb la incorporació de noves papereres s'hauria de redissenyar aquesta part per a poder visualitzar l'estat de totes les papereres de la infraestructura així com a més de poder escollir quines dades visualitzar i de quin dia, també incorporar la opció de quina paperera volem veure les dades.
- Un altre punt a realitzar en un futur es provar el sistema de recollida i enviament de dades en altres tipus de paperera.

- Un altre possible punt de millora és el disseny d'una placa especifica que realitzi només lo necessari per a aquest treball, recollida de dades i enviament d'aquestes al servidor. Així es podrien minimitzar els costos de la placa i segurament la mida d'aquesta.
- Un aspecte que s'hauria de tractar també és la protecció de tota aquesta infraestructura contra possibles usuaris maliciosos. Tant en l'aspecte físic de protegir els elements de les papereres (sensors, placa...) com l'aspecte de la infraestructura del servidor per a que no pugui accedir-hi ningú i modificar les dades, eliminar-les...
- Millorar el sistema de avis si aquesta es crema ja que ara has d'estar pendent de refrescar la pagina web. Fer un petit script per a refrescar la pàgina on es mostra l'estat actual cada un cert temps.

Agraïments

I per últim, expressar la meva gratitud a aquelles persones que han fet possible la realització d'aquest treball. En primer lloc agrair al meu tutor del treball en Lluís Solano, per la atenció prestada i tots els consells i informació que m'ha facilitat.

Agrair també a tota la meva família i amics que m'han ajudat durant tot el treball, per el seus consells i correccions.

Bibliografia

Referències bibliogràfiques

- [1] *Arduino – Introduction* [en línia]. [Consulta: 4 abril de 2016] Disponible a: <https://www.arduino.cc/en/Guide/Introduction#>
- [2] *Interaction Design Institute Ivrea – Wikipedia* [en línia]. [Consulta: 4 abril de 2016] Disponible a: https://it.wikipedia.org/wiki/Interaction_Design_Institute_Ivrea
- [3] *Arduino – ArduinoBoardUno* [en línia]. [Consulta: 4 abril de 2016] Disponible a: <https://www.arduino.cc/en/Main/ArduinoBoardUno>
- [4] *Internet of things – Wikipedia, the free encyclopedia* [en línia]. [Consulta: 4 abril de 2016] Disponible a: https://en.wikipedia.org/wiki/Internet_of_things
- [5] *Wearable – Viquipèdia, l'enciclopèdia lliure* [en línia]. [Consulta: 6 abril de 2016] Disponible a: <https://ca.wikipedia.org/wiki/Wearable>
- [6] *¿Que es el ICSP? – Arduino | . : Vico Linker ^ : .* [en línia]. [Consulta: 6 abril 2016] Disponible a: <http://www.vicolinker.net/que-es-el-icsp-arduino/>
- [7] *Arduino – ArduinoYun* [en línia]. [Consulta: 7 abril 2016] Disponible a: <https://www.arduino.cc/en/Guide/ArduinoYun>
- [8] *OpenWrt* [en línia]. [Consulta: 6 abril 2016] Disponible a: <https://openwrt.org/>
- [9] *Raspberry Pi – Teach, Learn, and Make with Raspberry Pi* [en línia]. [Consulta: 10 abril 2016] Disponible a: <https://www.raspberrypi.org/>
- [10] *What is a Raspberry Pi?* [en línia]. [Consulta: 10 abril 2016] Disponible a: <https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>
- [11] *Scratch – Imagine, Program, Share* [en línia]. [Consulta: 10 abril 2016] Disponible a: <https://scratch.mit.edu/>
- [12] *Welcome to Python.org* [en línia]. [Consulta: 10 abril 2016] Disponible a: <https://www.python.org/>
- [13] *Raspberry Pi – Wikipedia, the free encyclopedia* [en línia]. [Consulta: 10 abril 2016] Disponible a: https://es.wikipedia.org/wiki/Raspberry_Pi

- [14] *Raspberry Pi 3 Model B – Raspberry Pi* [en línia]. [Consulta: 10 abril 2016] Disponible a: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- [15] *Apache HTTP Server – Wikipedia, the free encyclopedia* [en línia]. [Consulta: 12 abril 2016] Disponible a: https://en.wikipedia.org/wiki/Apache_HTTP_Server
- [16] *LAMP (software bundle) – Wikipedia, the free encyclopedia* [en línia]. [Consulta: 12 abril 2016] Disponible a: [https://en.wikipedia.org/wiki/LAMP_\(software_bundle\)](https://en.wikipedia.org/wiki/LAMP_(software_bundle))
- [17] ETSEIB-UPC. Dpt. De Projectes d'Enginyeria. *Tema 3. DISEÑO TÉCNICO Y ANÁLISIS DE ALTERNATIVAS*. Barcelona.
- [18] *HTTP Methods GET vs POST* [en línia]. [Consulta: 16 abril 2016] Disponible a: http://www.w3schools.com/tags/ref_httpmethods.asp
- [19] *Arduino – YunSysupgrade* [en línia]. [Consulta: 16 abril 2016] Disponible a: <https://www.arduino.cc/en/Tutorial/YunSysupgrade>
- [20] *Arduino – ExpandingYunDiskSpace* [en línia]. [Consulta: 16 abril 2016] Disponible a: <https://www.arduino.cc/en/Tutorial/ExpandingYunDiskSpace>
- [21] *Cómo instalar LAMP en Ubuntu 14.LTS, paso a paso – styde.net* [en línia]. [Consulta: 18 abril 2016] Disponible a: <https://styde.net/como-instalar-lamp-en-ubuntu-linux/>
- [22] *Factor de emisión asociado a la energía eléctrica: el mix eléctrico. Inicio. Generalitat de Catalunya* [en línia]. [Consulta: 20 juny 2016] Disponible a: http://canviclimatic.gencat.cat/es/reduex_emissions/factors_demissio_associats_a_len_ergia/

Bibliografia complementària

- [1] *Wiring* [en línia]. [Data de consulta: 4 abril 2016] Disponible a: <http://wiring.org.co/>
- [2] *Processing.org* [en línia]. [Data de consulta: 4 abril 2016] Disponible a: <https://processing.org/>
- [3] *PHP 5 Tutorial* [en línia]. [Data de consulta: 5 maig 2016] Disponible a: <http://www.w3schools.com/php/>
- [4] *HTML Tutorial* [en línia]. [Data de consulta: 7 maig 2016] Disponible a: <http://www.w3schools.com/html/default.asp>
- [5] *CSS Tutorial* [en línia]. [Data de consulta: 8 maig 2016] Disponible a: <http://www.w3schools.com/css/default.asp>
- [6] *JQuery Tutorial* [en línia]. [Data de consulta: 12 maig 2016] Disponible a: <http://www.w3schools.com/jquery/default.asp>

